UNIVERSITETI I EVROPËS JUGLINDORE
УНИВЕРЗИТЕТ НА ЈУГОИСТОЧНА ЕВРОПА
SOUTH EAST EUROPEAN UNIVERSITY

Functional software testing of a web application and analysis of requirements coverage

Mentor:
Assoc. Prof. Dr. Visar Shehu

Candidate:
Hana Rustemi, BSc

Contents

List of Figures

List of Tables

Abstract

Software testing is an important part of Software Development Life Cycle which encapsulates steps like designing, running and maintaining operations. It is a crucial step towards creating a valuable product that fulfills the requirements of the stakeholder. Testing both ensures that the initial requirements are fulfilled, and the product will be working causing no errors to the users. This thesis is analyzing the importance of software testing in SDLC and functional testing by analyzing the techniques, levels and methods of software development. Also, the related work has been researched and used for further conclusions and comparisons.

Methodology

Black-box test scenarios are designed to perform manually in an application and later test coverage metrics are calculated to evaluate the success of the testing process. In this paper process, time, human resource, advantage, disadvantage, limitations and usability differences of manual test and automation test processes are compared as well.

Keywords

Software testing, SDLC, STLC, Black-Box testing, White-Box testing, Manual Testing, Automated Testing.

1. Introduction

Testing has always had and will have a crucial role and importance in software development. Considering the fact that each field of our lives nowadays requires usage of a software, be it a web app for shopping, a desktop app for professional or educational use or be it a mobile app for using in our spare times, each of them in essence are software products that some developers have created and possibly are maintaining continuously. These products are required to work seamlessly, without errors and fast, pretty fast. Popular industrial research claims a rule named as *"15 seconds rule"* which is widely used to guide web developers to develop products to attract the visitors attention in less than 15 seconds. Doing so requires maximum effort and minimum errors. To be able to answer the needs of this dynamic world and to fill the ever-changing requirements several development cycles and approaches are being used.

With the increasing of available software in the world, a huge need for quality (software quality) has come in play which as we might guess is not quite easily achievable considering the fact that the requirements based on which this software are developed have become far than complex. Also, we should keep in mind that in these situations not only physical effort is spoken of but all these complex requirements also have a financial dimension, that when gone wrong can cost a fortune to the stakeholder companies or people and the only way of avoiding failures and generating successful and operational products lies on the field of software testing. Software testing even though sometimes is underrated because of constraints in time and cost, it is considered as one of the most important steps of Software Development Life Cycle. The whole process of testing is aimed at finding bugs which may result in smaller or bigger errors in the future.

Testing can be realized with the tester knowing the internal structure of the software or not knowing its structure at all. It is a process of evaluating the overall or detailed components of the system through manual or automated means to verify the functionality and requirements. All these elements lead to determine some attributes of a qualitative software like: usability, efficiency, reliability, security, capability, maintainability, compatibility and many more.

1.1.Purpose of thesis

Functional testing is mainly conducted to validate the software product opposed to given requirements for functionality and thereby contribute to its' usability.

In general, functional testing, recruits black-box testing methods to run several tests which do not consider the internal structure of the software product.

As the technology develops, the need for human interaction is decreasing in each field. Testing is one of them, where automated software outperforms people and human effort in any way, but yet there are situations were human involvement results in best accomplishments

These tests are conducted usually in automated environments that and they validate aspects like: Accessibility, Usability, User Interface and more. The paradox here is that testing manually these aspects is a time and cost consuming process but on the other hand using automated tools is costing much to the companies plus there is need for a specialist that knows how to use the automation tools as well as good programming language knowledge. In addition, using automation does not always mean that the process will work flawlessly just because it is done by computers and programs independently. It generally requires human input which in some organizational cases might confuse testers whether to use manual or automated methods. Another point that puts me in doubt about using automation tools in functional testing is that handling test scenarios in those environments is very much like doing the testing manually and in such cases, testers prefer to do the work manually with the conventional methods of collecting data and documenting results.

Hereby, this thesis aims to demonstrate the process of testing an application on its functionality with conventional black box method of creating test scenarios and their test cases and evaluate the Requirements coverage for these tests.

1.2 Hypotheses

H0. Automated test gives the best results in software testing.

H1. Manual Functional testing of a web application results in better error finding.

H2. Combining Manual and Automated test in functional testing enhances the requirement coverage.

1.3 Research Methodology

This thesis is conducted based on Primary Quantitative Method which includes Experimental Research that will address which testing method gives best results in terms of error finding and covering requirements.

The instruments to be used in this process will be use case scenarios for manual testing and automation tool.

2.0 Software Engineering

The term “Software” has been an immensely pronounced word in our lives through the past decades. According to Pressman and Maxin [1] it is a tool that “delivers the most important product of our time-Information”, because it has been playing a dual role by being both a product and a tool for delivering product. Independent of where it is located (mobile phone, tablet, embedded systems, pc, laptop) it transforms information that are gathered by multiple and independent sources.

It is not an easy task to obtain a software that is capable of transforming and serving information to users seamlessly considering the fact that the technology has developed itself and everyday new software products come along that simultaneously rises the expectations of the users. A vast percent of electronic devices includes a software, so does the manufacturing technology, also we have to mention that tasks like: banking, teaching, shopping, even entertainment market is based on a software.” The term 'software engineering' was suggested at conferences organized by NATO in 1968 and 1969 to discuss the 'software crisis'. The software crisis was the name given to the difficulties encountered in developing large, complex systems in the 1960s. It was proposed that the adoption of an engineering approach to software development would reduce the costs of software development and lead to more reliable software” https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/History/.

Software Engineering has been defined by to Pressman and Maxin [1] as a process along with a collection of methods and array of tools that result in building a high-quality software product. As a process, it enables professionals to escape the chaotic environment of development by following a defined path but also leaving them flexible to adapt their approach. Also, another definition is by IEEE” The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software"—IEEE Standard Glossary of Software Engineering Terminology”

2.1. Software Development Life Cycle

Software Development is a collection of processes that lead to creating a product based on the requirements, in some cases these processes are complex and interconnected steps which either are lined as an incremental array of steps or a cycle that iterates itself. Whatever process is undertaken, there are 4 major and fundamental steps that need to be inside a development process [24]:

- Software specification – which is usually called as Requirements Engineering, where usually the input from the customers is the key, which consequently generates the limits or constraints of the product by documenting the core functions of it.
- Software design and implementation – where the requirements are implemented or defined as data models, interfaces and structures.
- Software validation - usually called as Verification and Validation where Verification makes sure if the requirements have been implemented into functions, and Validation refers to the product being based on the given requirements. Usually testing is the most important component of the Validation process, which generally has three stages: Development testing, System Testing and Acceptance Testing.
- Software evolution- refers to the part of maintaining the product after development which in general is underrated and considered dull, but in fact a consistent maintenance of a product that has to undergo changes in correspondence to the needs and changes is what makes it valuable.

The above-mentioned processes are further combined together with other steps in so called Process Models or Process Paradigms which serve for defining the architectural framework of the steps taken to develop a certain product. Although they are defined and grouped in several types, they can be further extended in order to create other engineering processes.

2.1.1. Software Development Process Models

Although through the passing time the need and the strive for faster and perfect product has led to more models which today reach to a number of approximately 50 process models, there are some fundamental models that one way or another might be suitable to project. The following 8 models are considered as the widest used methods:

Fig.1.The waterfall model

Is the conventional development model which has been noted as a firm basis for the further coming development models.It is based on iterative and sequential approach where each step is a ground for the next step. This model is usually used in development of databases or operating systems or projects whose requirements won't be changing [1]. In contrary to this, mostly waterfall model is compared to the incremental model which is one of the best approaches to agile development in mainly e-commerce, where the user can interact with the product in its earliest stages because each increment of the product should possibly include a requirement of the customer and the customer gets to see a demonstration of the product up to some point.[24].Based on a research of The Standish Group in 2015 over 10.000 projects, its shown that the Waterfall model have a successful rate of 44% over Agile methods only in Small Sized Projects[26].

2.V-Model

The V-model (Vee Model) is an improved version of Waterfall model where the development process is broken down to components in top-down approach and for each component verification and validation is applied through compatible tests like acceptance, system, integration and unit testing, which consequently means that if you pass acceptance test- the requirements have been placed precisely, if you pass system test- the system architecture has been designed correctly, the same is valid for the integration and unit testing where its proven that module design and code are working properly. It is mostly chosen to be used in projects with restrictions in time and failure like airplane fleet management.

Software Product [[1]

F

i

g.2The V-Model

3. Incremental and Iterative Models

In the Incremental model, the aim is to create a Minimum Viable Product with core functions that reflect the requirements of the customer, and then based on the feedback and comments of the user the product can be added with the next features. We can say that, it goes through iterations and after each iteration the new requested feature is added.

Fig.3. The Incremental Model

However, in the Iterative model, the major requirements that cannot undergo a big change are stated, although through the iterations small modifications to requirements can be done. At the final stage, after all the requirements are fulfilled, the product is ready to be presented to the client.

Fig.3.1. The Iterative Model

4.The Agile Methods

In contrary to the conventional plan-driven model where strict project planning and rigorous processes, mainly focused on how the product should be developed rather than the development process itself including the testing, with no consideration on changing requirements that desperately need change in design, the agile software development process model is the most used development process in the World. According to [27] 71% of the companies have embraced an Agile approach in their development process. Agile method is implemented based on "The Agile Manifesto" [28], which is more considered as a philosophical approach to development rather than a methodology.

The general idea of this method is to observe and develop for a period and do these steps in a loop. Design and implementation are considered as main activity later combined with other steps like requirements gathering and testing [24]. Here the planning and observation takes more time due the fact that in these phases the possible mistakes are considered, which creates a solid base for decreasing the upcoming failures, in this way the product can easily transform to an MVP and then can be improved. It is usually fitting best in the projects which have constant changes in requirements while in development.

Fig.4 The Agile Method

4.1. Extreme Programming

Is the most widely used version of Agile method, which mainly is based on iterative approach but in 'extreme' limits. As a simple example, in extreme programming based on a requirement, a scenario is created, implemented as task units, test is planned ahead of development, then the code is integrated to the system with fully executable and successful tests which may lead to several developers creating versions, integrating and testing them in a very short period.

Extreme programming has generally 12 practices that may be grouped in 4 categories:

- Communication with customer and the group of programmers
- Application's specification and test cases are derived in collaboration with the customer
- A partner should be involved in programming
- Multiple testing of the base code

4.2 Scrum Method

In the scrum method, there are development phases of: requirement, analysis, design, evolution and delivery. All these phases need to happen in a process flow that is called as sprint. The process is carried out in a group of development activities:

Backlogs- business requirements of the client that prioritize the requirements.

Sprints-are blocks of tasks that need to be done in order to achieve a backlog that consists requirement (generally up to 30 days)

Meetings-daily held, short meetings where team members question what they have done since last meeting, the problems they have faced and what needs to be achieved until next meeting.

2.2. Software Testing in SDLC

Testing a product is generally seen as a process that will prove that it is error-free or all the client requirements have been fulfilled. It is usually considered as a dissuasive and a consuming process both psychologically and economically whose, in my opinion, sole aim should be finding errors in any circumstance. Not even a single product can be error-free, thus as much as consuming it might be, it needs to be incorporated in each type of SDLC.I am going to explain how testing is achieved in the following development models. But first we have to consider some testing principles that are defined in the book "The Art of Software testing" [2] in the following table:

Nr.	The Principle
1	The expected output is an important part of the test case
2	A programmer should avoid testing his/her code
3	An organization/company should avoid testing their own program
4	A testing process should analyze the results of the test in detail
5	Test cases should be prepared for both valid and invalid inputs
6	Testing should check both if a program does and doesn't do what it should not do
7	Test cases should be reusable
8	Do not create test cases that aim to prove the product to be error free
9	There is a proportional relation between errors that are found and those to be found
10	Testing is both a creative and challenging task

Table.1 Testing Principles

2.2.1. Testing in Waterfall Model

As mentioned above, Waterfall model is a strict and highly disciplined development model. It flows the process of development in a sequential way where each step comes after finalizing the previous one, and each step should be completed in order to pass to the next one. As a result of this, testing is carried out after each phase which in the advantage side helps provide a qualitative product. But the down side is that, because the testing happens in the final stages, the errors are detected late and their cost to the product are immense.

2.2.2.Testing in V Model

The V Model or Verification and Validation model, as explained above is not a linear approach to development but instead it is based on a verification principle where each phase is depending

on the previous step in order to proceed further. Here testing happens in each step and requires a parallel involvement of the developer and tester. System test cases are prepared based on requirements, Integration test cases are prepared based on High Level Design, and for the Low-Level Design the unit tests are prepared.

Testing in Iterative and Incremental Model

The iterative model aims to provide a working software in short and frequent intervals to the client and they get to test an actual version. The testing process is divided into two phases, the first phase consists of:

- Unit testing
- Component Testing
- Integration Testing

Which are usually conducted for each iteration before the product is completed as a whole, and then during the final phases of development where the team considers that the requirements delivered by the customer have been fulfilled, the product goes through the second phase of final testing which is made of:

- System Integration Testing
- Acceptance Testing [29]

2.2.3. Testing in Agile Model

As we know, agile development embraces a fast, iterative and incremental development including testing which generally provides user involvement and the changes happening in this phase are welcome for the general process. It is not a conventional or traditional model due the fact that the competition in the market has imposed a qualitative product that can be delivered immediately with the least possible errors. Because of this testing plays an important role in this development model. As in nature of agile development, also in the testing phase the involvement of the client is crucial in every stage of development stage. In addition to this, developers are also involved in the testing phase which makes testing process a collaborative one. For example:

- Initially developers generate unit tests ahead of coding actual units which will test their product
- Generally automated tests are used to enhance the fast production schedule.
- After a deploy, customers take the acceptance test to give their feedbacks on the product [2].

2.2.4. Testing in Extreme Programming

Compatible with the term “Extreme” the testing applied in XP is also considered as extreme therefore retestingor constant testing is practiced in this model. Laying on the foundations of Agile Development and similar to it, the tests are provided ahead of coding whose target is to find errors at the earliest. For this purpose, unit testing and acceptance testing are the principles in charge of testing process. Unit testing is done with the purpose of finding errors and consequently fixing those errors, and acceptance test is done by the clients who check the fulfillment of their requirements.

As we can see above, in each development model testing is included one way or another. In some models testing is done parallel to the development process in general, and in some models, testing is done after some phases have been implemented. In each situation, it shows us that testing is an important phase which feeds and enriches the product by making sure that there are possible or existing errors but they will be fixed and everything will work properly. Let’s not forget that testing is not conducted to assure the clients that there are no errors, it starts with the assumption that there are errors and we need to find them as soon as possible.

2.3. Software Testing

Testing is a process mainly conducted to find errors in the software, determine risks and help to get the best out of the software being produced.It usually serves to determine the qualiyt of the product and it is an important discipline of Software engineering due to the fact that it consumes approximately 50% of the devlopment process and efforts[9].All problems can be identified as bugs and these can be caused by anything from a mistake in the user requirements up to code errors.These bugs can drastically decrease the time and cost of the product which can end with multiple conflicts between customer and development team. At the end what matters is the quality of the product.Moreover, testing can also be seen as an important component of the Verification&Validation[1] which is mostly inherited as a framework with different activities that ensure a product, software or a system fulfills its initial requirements and that the final product is a qualitative result of these requirements.Today, as the requests for a qualitative software are rising, testing is becoming more important. The following are the ISO 9126 Software Quality Characteristics[7] which is an international standart for evaluating a software product regarding the following subjects:

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

where testability is lined under Maintainability subject of the standard. Not only for maintainability, but also with testing in the early stages of the development many mistakes and inadequacies can be detected which contributes to the quality of the product hence satisfy both the customer and development company.

Quality is a norm that corresponds to software being developed according to the requirements and it shows a degree of excellence if it conforms to some parameters which differentiate for the customer and developer.

When testing a software, the results of the tests tell us about errors, defects or any malfunction in it. Although this is a long process which contains many steps, levels and methods it serves to two goals:

1.To prove that the product fulfills the requirements of the customer

2.To track anomalies or cases where the product does not meet the customer's specifications

Through the history, testing has been primarily made to find errors, then it has been conducted to show that the software product had been built correctly, and in the past years employing testing in a destructive manner gives better results to the developers. The important thing to know is that no software is a total success and not any software can be tested in 100%.

2.3.1. The Importance Of Testing

What gave more importance to testing might be some failures in the past that have happened as a result of a software malfunction or a simple error in the code. The following are some major disasters that have happened in the past years, and the main reason has been considered as a lack of testing in their SDLC [31].

- Credential leak of Yahoo that exposed 500 million credentials in 2016
- Airbus A400M bug that had caused a crash in 2015
- Apple iPad in 2012 got on sale with a price of £49.99 instead of £650 in Tesco
- Ola India's largest taxi company, due to a glitch in their system provided free rides to programmers who infiltrated in their system

These are only a few examples that once again have proven the importance of testing in any kind of system, software, or product. As we can see the cost of these are not only ethical disasters that have damaged reputation of the related companies, at the same time they have had an impact in the material aspect as well. Usually the cost of fixing bugs rises proportionally as the stages of development progress further, which means that the earlier the bugs are discovered the minimal cost impact they will have on the company, for example a bug found in the early stages of development like requirements phase can cost nearly nothing; a bug in the coding process can take little to minimal effort to correct it with little interaction; a bug in the system testing may cause a delay of delivery of the product which is considered as tolerable but a bug on the system functioning may cause a disaster[3].

The advantages of software testing can be very different from the perspective of developers and customers but in general we can summarize them as the following:

1. Discovering new or existing errors
2. Proving the security of the product
3. Proving the quality of the product
4. Prevention from being sued for inadequacy by the customer
5. Proving that there are no errors in the product

Before proceeding with further testing information, we should clear the three main terms in testing, *fault, error* and *failure.* These terms are best described by Amman.P and Offutt.J in *"Introduction to Software Testing" [34]* as following:

- Fault-a defect in the software
- Error-An internal condition that comes as a consequence of a fault
- Failure-An external incorrect behavior that opposes the requirements

2.3.2. Foundations of software testing

Although the first statements about software testing date back to 50's to the time when FORTRAN language was developed, the genuine definitions for software testing are accepted as the book of Glenford Myers "*The Art of Software Testing"* was published in 1979[2]. But with radical and rapid changes both in computer hardware and software developed, it was a necessity to update and adapt the methodologies and definitions to the changing environment. If we have to explain how testing was developed in a timeline basis, then in the literature we can see the article of D. Gelperin and B. Hetzel [32] where they have divided the history of testing in different periods such as:

- The Debugging-Oriented Period (--1955) is the period when testing was based on computer hardware, it was not distinguished from debugging. Actually, the term bug comes from Ada Lovelace in 1843 where she described a problem with a program that Charles Babbage wrote, far later than that Thomas Edison in 1878 where he described a malfunction in his machine as a "bug". Correcting these bugs was mainly focused on hardware issues until 1950 when Alan Turing published an article where he was concerned about how a program can reveal intelligence, which consequently has brought up the concern whether a program satisfies its requirements. This led to creation of a test where an interrogator a program and a reference system developed testing.

- The Demonstration-Oriented Period (1957-1978) in timeline lasts between 1957 and 1978 when testing was developed to verify if it adhered to the specifications. Until this point, testing was referred as a process where you "checkout" if the program has bugs. Initially a step for definition of testing was taken by Charles L. Baker *(see Review of "Digital Computer Programming,1957")* when he proposed that a so called "checkout" needed a differentiation between testing and debugging and that it should bring clarity to two issues, first it should assure that the program works and it corresponds to a stated problem

and the second which is considered as the major purpose of testing is if the program satisfies the requirements. The sole aim of testing and debugging was to find, locate, and correct mistakes, but the need for making sure if a program runs correctly was an issue that needed further methodologies to be ensured.

- The Destruction-Oriented Period (1979-1982)- The destructive approach was applied from 1979 to 1982 whose main concern was to find as much possible errors at the earliest stages, even undiscovered errors. The testing in this period was so focused in finding errors that it had mostly ignored other aspects of providing and developing a successful product. In a way, it was in complete opposition of demonstration period where testing was done to assure requirements. In this point of view, substantial testing was done but the problem stood that the more destructive testing was done, the more errors were found hence debugging them took a lot of time, therefore this period lacked success and gave rise to fault detection methods that prospered in the following period of evaluation-oriented approach.

- The Evaluation-Oriented Period (1983-1987) - It was in 1983 when National Bureau of Standards published a guideline for developers, managers, maintainers and everyone included in software development process [33]. This guideline took the first step to a more convenient software development process. It was proposing a model called VV-T that was a combination of processes of verification, validation and testing to whom we are familiar at this point. The idea of this approach was to utilize different analysis and testing activities to provide evaluation of the product quality during its development life cycle to detect design, implementation and requirements faults. Based on this verification, validation and testing processes were adjusted to fit in each development cycle phases.

- The Prevention Oriented Period (1988- present)- As the name suggests, the main reason of testing was to prevent possible design, implementation and requirements faults. This was achieved by accentuating test planning, analysis and design, which opposed the evaluation approach based on analysis and review thus excluding testing process at later steps. The steps for testing have been described as: planning, design, construction, implementation and execution which until today have been present in any type of testing lifecycle.

2.3.3.Software Testing Life Cycle (STLC)

STLC refers to a set of activities that need to be done in order to verify that the product meets the requirements and in each organization testing is a separate but paradoxically inclusive act from SDLC.It is appropriate to say that although testing is a separate process which has its own phases, it corresponds to each cycle of the development process and each cycle has its own goals and results.We should keep in mind that each phase of the testing lifecycle has its input and

output criteria that define what requirements need to be completed before testing begins and what requirements need to be completed before testing finishes.

Testing life cycle has the following phases as depicted in the diagram:

Fig.5 Testing Life Cycle

Where:

1. Requirement analysis is the initial phase of testing in which developers and testers and optionally users discuss, analyze and gather user requirements against which the product will be tested. As a result of this requirements document is created.
2. Planning is the activity to predetermine the testing process, classify possible problems, define metrics and schedule of testing process. Also here the team determines which testing approach will be considered, what needs to be tested and what is opted out of testing.
3. Writing test cases is based on the requirements listed in the requirements document. Usually test cases and test data are the outputs of this step.
4. Designing is the phase where test document including plan, design and automation scenarios are usually the output. It
5. Running and executing the test cases happen in this phase which may include bug reporting and revision of test cases.
6. Reporting is the phase where the expected and the actual results of the tests are compared.
7. Final testing is the phase where all front and back-end test and manual or automated scripts are executed and if necessary updated.
8. Post implementation is the final phase of testing which usually ends with a defect report, provide strategies to ward off similar defects in the future [3].
9. This process can be modified by adding or removing some phases according to the needs, but this framework is the most general approach to testing lifecycle.

2.3.4. Software Testing Levels

Testing as we know is a process that begins with the developers and goes all the way to the user, there is no distinct line to define up to which point a user can test the product since errors might come up at the most unexpected time and way and in any level. The SDLC in itself is a layered structure of components that needs to be tested for each component so that the missing points can be fulfilled, and we can avoid repeating of the development stages. There are several testing methods suitable for each development stage, and we will initially define the four most important levels that are used in testing process:[3],[9],[12]

1. Unit Testing

2. Integration Testing
3. System Testing
4. Acceptance Testing

If we think of it as a testing hierarchy, the following would be suitable to describe these levels in the context of STLC:

Fig. 6 Levels of Testing

1. **Unit Testing** is the fundamental test level where each component of the software is tested including code, classess, objects and modules and this is done by decomposing each one of them separately. Testing in this level is done to verify the functionality and requirements fulfillment of the components. Usually, unit testing happens in the early stages of coding by the developers themselves, and errors are fixed as soon as they are found thereby avoiding further bugs in the software in higher levels of development.

2. **Integration Testing** is another nomenclature given to testing done when joining or combining components of a software, also integration testing can be done when integrating different software. The reason for conducting integration testing comes from the problems that possibly happen when developers finally combine several components of a software that they previously had tested individually but when combined together may cause errors. Testing is done by developers and it generally has sub-levels of testing such as:[15],[18]

- **Component Integration Testing** comes after component testing which simply tests the interaction between different components and aims to find any interface defects.
- **System Integration Testing** usually is conducted to test the behavior of the system as a complete integration of components from both hardware and software perspective which aims to verify the fulfillment of the requirements.

Integration testing may become more complicated based on the volume of the components that get integrated, thus segregation of errors may inevitably become impossible, to cope with this problem a popular approach is 'big-bang' integration testing that tests all components after they are integrated at the same instant. Also, there are top-down, bottom-up and incremental testing approaches in the context of Integration testing that are guided by the control flow based on the architectural structure.

3. **System Testing** is conducted on an entire, completed system and is usually a final test on the developer perspective which aims to verify that the resulting product meets its initial requirements including both functional and non-functional testing such as: load testing, performance testing, reliability, maintenance, security etc., and aims to find as many errors as possible. It generally begins with black-box methods that address testing functional-requirements, then white-box methods follow up to catch errors that may be missed in black-box methods.

4. **Acceptance Testing** is a non-functional testing level where the software gets to be tested by users themselves with the intention of confiding users that the system works properly rather than finding errors [8],[9].
   Along with these levels, usually the following are accounted as useful while testing a software to maintain its usability and confide to requirements:

5. **Regression Testing** is done after changes are made to the software, this ensures that although changes have been made, it does not cause any errors in the usage of the software.

6. **Alpha Testing** is an imitation of an acceptance testing process where the tester usually plays the role of the user and tries to identify any issues or bugs before the system gets to be released.

7. **Beta Testing** is similar to alpha testing but in this case the system is tested in a real environment by users before it is revealed to the clients, it may also be referred as external acceptance test.

8. **Functional Testing** is the last testing done by the user or independent tester to ensure that the system provides all the expected behavior.

LEVEL	TECHNIQUE	FOCUS	TESTER	Scope
UNIT	White-Box	Low Level, Code	Developer	Classes
INTEGRATION	White Box, Black Box	Low Level, High level	Developer	Single Class
SYSTEM	Black Box	Requirements Analysis	Tester	Whole System
ACCEPTANCE	Black Box	Requirements/Specifications	End-User	Whole System
REGRESSION	White Box, Black Box	High Level, Documentation	Developer, Tester	Any
ALPHA	Black Box	High Level	Tester	Any
BETA	Black Box	High Level	Customer, Tester	Whole system
FUNCTIONAL	Black Box	High Level	Tester	Whole System

Table. 2 Levels of testing

2.3.5. Software Testing Approaches

Aside from testing levels that mostly deal with how the testing process should be conducted based on the software development process in a hierarchical structure of a system which gives us a general view of what needs to be done in which development phase. To better understand the testing done in these development phases, we need to explain which approaches are embraced when trying to test a software product independent of its focus level.

In different sources approaches are referred as techniques or categories but because, the process in these tests is mostly based on an execution attitude that has many types and categories, I will refer to them as approaches that are combinations of several types of testing.

These approaches are called as Static and Dynamic, where in Static testing the program is not executed but rather source code is tested with respect to program analysis, error handling, functional requirements and overall code design. However, in Dynamic testing the software should be compiled and ran and be tested for its quality under real circumstances. Mainly inputs are given and their outputs are tested, generally test automation is employed here.

Fig 7. Dynamic And Static Approaches and The Corresponding Methods

As shown in the above graph, the dynamic approach is mainly concerned with *Validation* process meaning that it tests if the product meets its requirements and the static approach deals with verification to check if the product has been built based on the specified requirements.

In this paper, black-box methods will be emphasized but for the sake of comparison I will as well explain the main aspects of white-box testing, and compare their advantages and disadvantages.

Black Box Testing

Black box testing has an implicit role in the whole process of software testing because it tests the overall functionality of the software by validating its components. It is mainly done on the basis of requirements so that errors related to them can be traced easily. It is done in the perspective of the user or a tester and inputs/outputs are checked in a finished product. The product is precepted as a black box meaning that the tester or the user has no knowledge of the internal structure of the product (code) and deals with how the product is visible to the user and the quality it offers to them.

Fig.8 Black Box Testing Method

Black box testing method usually uncovers errors that are related to interface, performance, initializing and terminating different processes and functions that do not exist or are incorrect based on the requirements [12]. Black-box method can be conducted in every phase of development cycle and the participation of the tester to the project should begin since the requirements phase. This method is used in a variety of levels including Unit, System, Integration and Acceptance but has proven efficiency in System (if requirements are met), Integration (if the system/components stick to use cases) and Performance (if load or performance evaluation is requested in use case) testing best. Although there have evolved multiple types of this method mainly the following are accepted as best ways to implement testing in a black-box perspective of a product:

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Tables
- Orthogonal Array
- Cause-Effect Graph [15]

Advantages and Disadvantages of Black Box testing

Even though the usage of the black-box testing methods is broadly defined as to in which level, situation, purpose and many other circumstances that they might be used, there are still some points where these methods may cause issues when being used. Below are listed some of

advantages of using black-box testing method opposing the disadvantages, thus a tester should have in mind when deciding their testing methodology [35].

Some advantages:

- The tester is independent of the code, so there is no pre-requisite to know or understand the underlying code (neither programming or implementation)
- Testers and developers can work independently
- Testing emphasizes in the users' point of view, which provides a complementary help to find the errors that may be missed from developers
- Fast and effective in finding errors
- No technical knowledge required

Some disadvantages:

- It doesn't exactly uncover all related errors to the product, thus some deeper errors or bugs should be subject to other testing types
- Only a given set of inputs are tested, it is impossible to test all possible input cases
- Testing open or broad (not strictly defined) requirements or even creating test cases for them is a hard piece of work.
- A big number of back-end features are not tested

White Box Testing

White-box testing is conducted by developers who know the internal structure and the code of the system. In these methods, the components or entities of the system are tested for their errors from the perspective of the developer. It is necessary for these methods to be conducted by the developers or someone that knows the internal logic of the system because in that way every possible branch of the code gets tested and its crucial for them to know how the system is supposed to function and in that way, they can compare if its intentions are met. Because it deals with the code mainly and its logical implementation, its intended to be used in Unit, Integration and System testing, although there are several phases that it can suit like requirements, design etc. Generally, they serve for two purposes: structural testing and coverage analysis, and reveal errors related to syntax, data flow and different conditions. Finally, in white-box testing an anticipated outcome is expected via a given input [2].

Advantages and disadvantages in White-box testing

Some Advantages are as listed:

- Unseen errors are easier to uncover
- Coding skills of developers are enhanced
- Contributes to code optimization
- Dead points of code can be eliminated
- Code errors are uncovered in earlier stages due checking the whole code systematically

And some disadvantages:

- After integration the test methods may become inefficient
- It generally is expensive
- Test cases are volatile to the changes in the implementation
- Testing of all conditions and paths is time consuming

2.3.6. Manual vs. Automated Testing

It is a long debate whether manual or automated tests shall be used in testing process. Although automation tests show better performance in time, cost and efficiency there are cases where manual tests are ought to be used due the fact that tester/user involvement is higher in the whole process of developing test cases and running them which and this involvement contributes in complex scenarios by decreasing the risk of false negatives. The intention is to let them work where they suit best: automated tests in less complex and tedious scenarios and manual tests in complex scenarios.

Manual tests

When testing a software manually testers mimic random users where they use the software step by step and see if the real users get what they should from it and if software works properly, as it is intended to. In these tests, testers get a vision of the system and the feedback is immediate, if there are errors, they are visible. They are not expensive since testers use their intuition and can identify bugs. Also, manual testing is flexible and if there are small changes in the back end the tester can perform testing without the need to change the test case. On the other hand, human eye can be deceiving thus manual testing is prone to mistakes and since they are not recorded, in the cases where repetitive testing is needed it becomes a tiresome process to repeat the test cases over and over.

Automated tests

Since testing has gained a popularity in the previous years, it is used in each and every part of software development, this means that test cases are run repeatedly during development and deploy and it is a long-term process which requires automation to save time and cost.

Automated tests bring cost-effectiveness to the whole testing process by optimizing human resources. They are written in a programming language/script and they require less or no human involvement and these scripts are repeated until a given condition. The results are documented and delivered to the development team which consider the need for changes based on errors found. At the end test scripts can be run repeatedly in an infinite number depending on the circumstances.

Even though automated and manual tests seem diverse in the way how they operate and it amenably seems that automated tests are efficient in most ways, there are situations which show the significance of using manual testing as a part of testing process.

Below, I depict a table which shows best usage of both these methodologies that ensure the optimization of testing process:

Manual Test	Automated Test
Exploratory Testing	Regression Testing
AD-Hoc Testing	Performance Testing
Usability Testing	Iterative Functional Testing
Functional Testing	

Table 3. Manual and Automated test usage

We should be aware that, even though we can define usage spectrum of manual and automated tests there is no universal and no strict method of testing a product because each project has its own dynamics, deadlines, designs, structure and development models, thus combining manual and automated methods results in a better testing strategy but also this requires a highly qualified QA team that elaborates up to which point the product will be tested manually and when it becomes insufficient the rest of the process gets automated and this enables a high rate of testing coverage.

Selenium IDE

This thesis includes usage of Selenium IDE which is a test automation tools that saves time and workload when working in long term projects which consequently increases productivity. It handles main goals of test automation respectively automating ongoing functional testing, reduce workload and save time, but when speaking of complex tests, it has shown that manual testing is more efficient.

Explaining a little more, Selenium IDE can test whether a web site or application works properly for given functionalities like: login, search, add to cart, edit, delete etc. It is a user-friendly application that allows the user to record a test case with its steps on the web site or application then can playback the steps and the user can save these cases to reuse them later in case of need.

Selenium ID is an extension to Chrome and Firefox and as such it can easily be downloaded.

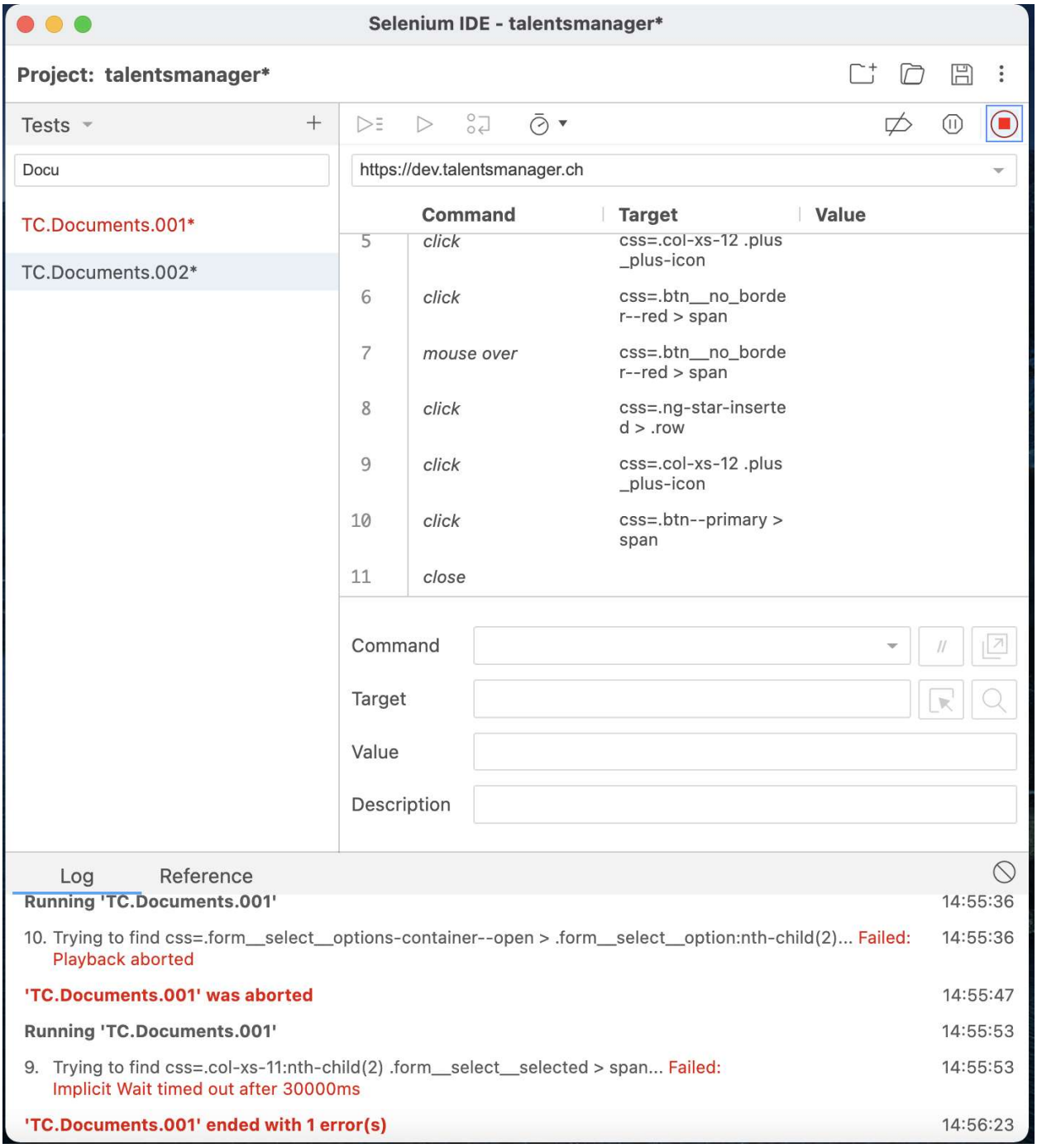


Fig.10 Selenium IDE

- **Projects** is the place where we can see and rename a project
- **Tests** menu locates all the test cases that are created by the user which also can be changed to show the **Test Suites** where corresponding test cases can be added.
- **Create Project, Open Existing Project and Save Project** buttons are on the right edge
- **Record** button is clicked prior to executing test steps which are then recorded
- **Pause** is used to stop recording steps and continue if necessary
- **Play From Beginning** is used to play a recorded test case

- **Play** is used to play recorded steps at any time
- **Step by step play** is used to play each step of the case individually
- **Playback speed** can be arranged based on the needs.
- **Command** is the place where Selenese commands are written based on actions performed by user, it can either be filled manually by the user or the recording catches them. There are many commands including: open, click, type, assert, verify, check, etc.
- **Target** is the element which is being tested on the page
- **Value** is the given value to a selected target
- **Log** is the place where information about running tests is displayed, it displays whether a test is successful or failing.
- **Reference** is the documentation of the commands used.

3. Test Process and Methods

Functional testing refers to what the product actually does. Generally, these are defined in a requirement or functional specification, and testing of these specifications and functional testing may be conducted in any test level (unit, component, etc). Testing functionality of a product can be based on the requirements or business where test cases are derived based on the requirements specification or business processes that most likely lead to use cases.[35].

The main method for conducting functional testing of a software is through black-box methods that are suitable for the case and the framework is consisting of ISO 9126 quality characteristics (mentioned in the above sections) namely Functionality characteristics consisting of:

- Suitability.
- Accuracy.
- Interoperability.
- Security.
- Functional compliance.

This research demonstrates the process of testing a web application that is developed to serve as a recruitment application in which companies/users create their profiles to employ people, maintain and cooperate with their employees. On the other hand, employees can use the system to coordinate with their employers and co-workers.

The application is developed on an agile development model which has several releases that integrate multiple iterations where features, requests or defects are maintained and organized. Each release contains new features and enhancements that need to be tested continuously, most of which are functional enhancements or defect improvements.

The url of the application is https://dev.talentsmanager.ch. Basically the system works on tasks or Todo's which can be appointed to different employees that may include documents and all these are conducted on a calendar basis. The application is still on development phase thus the testing conducted is also accounted as an alpha testing and the author is accounted as an end user that has no knowledge of the code and tests the application on the base of functional requirements that form the basis of test scenarios and cases.

The test cases are derived from 55 requirements which are formulated based on documentations that have been published by development team and several stakeholders of the project on GitLab.

3.1 Modules of Application

The application is a dynamic web page, consisting of several functionalities to enable the user maintain their employees, and it consists mainly of these modules:

- Dashboard
- Personal
- Documents
- Calendar
- Notifications
- Todos
- Settings

If we were to shortly explain the functionalities of these modules, it would be best to provide screenshots as follows to depict the application with its general features. Consequently, the Dashboard is the place which is displayed in 5 divisions that gives a dynamic vision to the user at the same time it provides a fast access to other functionalities like Todos, Contacts, Events, Team and Activities example:

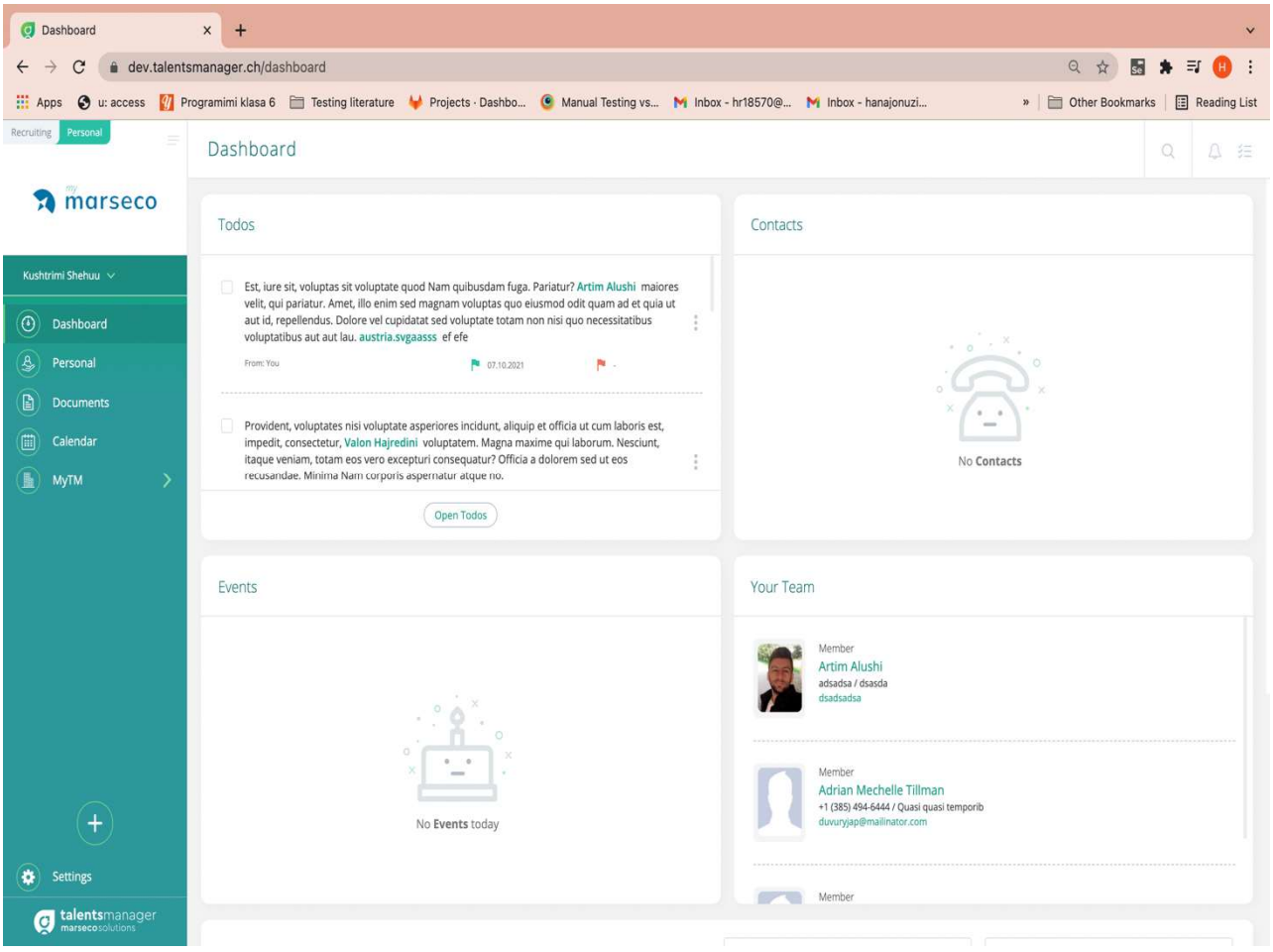


Fig.9 Talentsmanager Dashboard view

-Following the Dashboard, Personal components' main feature is to allow the user add a new employee besides it displays a populated list of employees which might be searched throguh statuses active and inactive or with details of location, division etc, eg.:

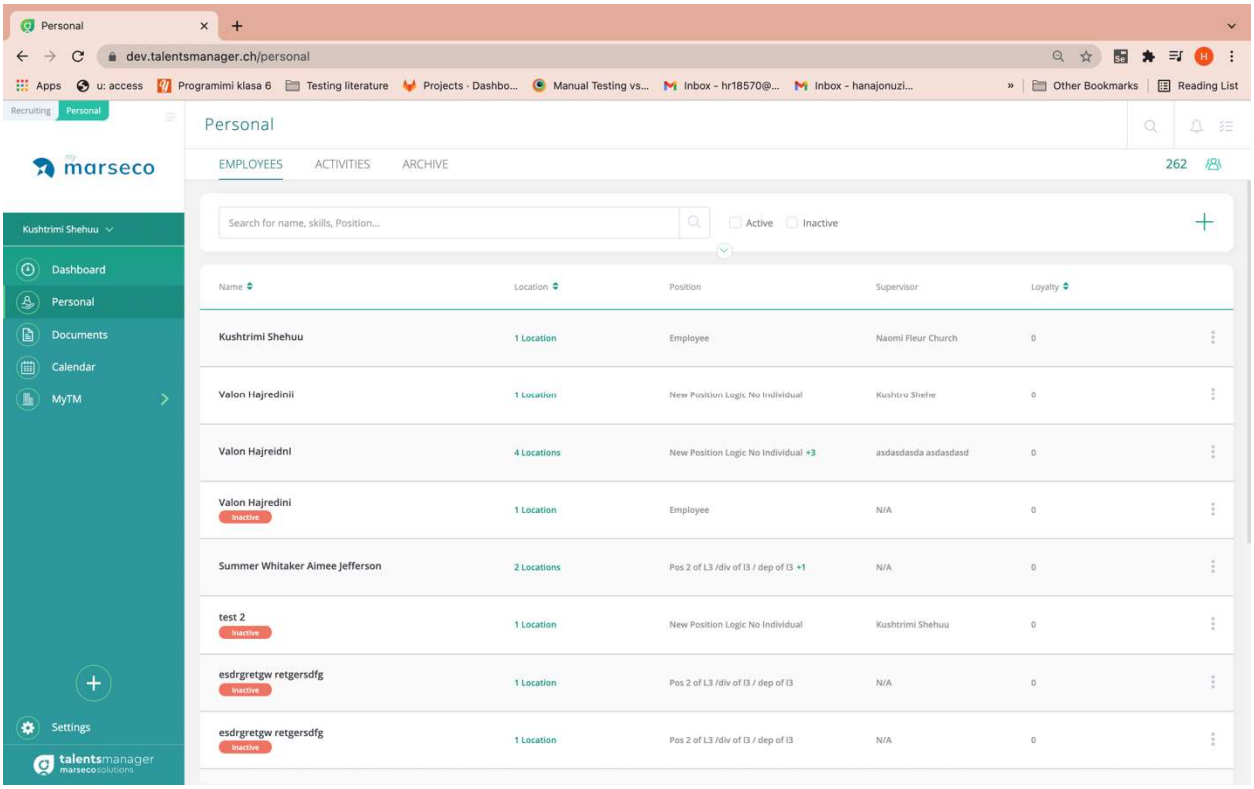


Fig.10 Talenstmanager Personal module

The Documents component stores and displays documents that a user or employee might need to provide for their duty. The user can add a document by dragging and dropping or uploading to from their device, attribute them as confidential or search through the already uploaded documents. Along with this, activities of different types like document deletion, document creation etc. of the users are also displayed in this component.

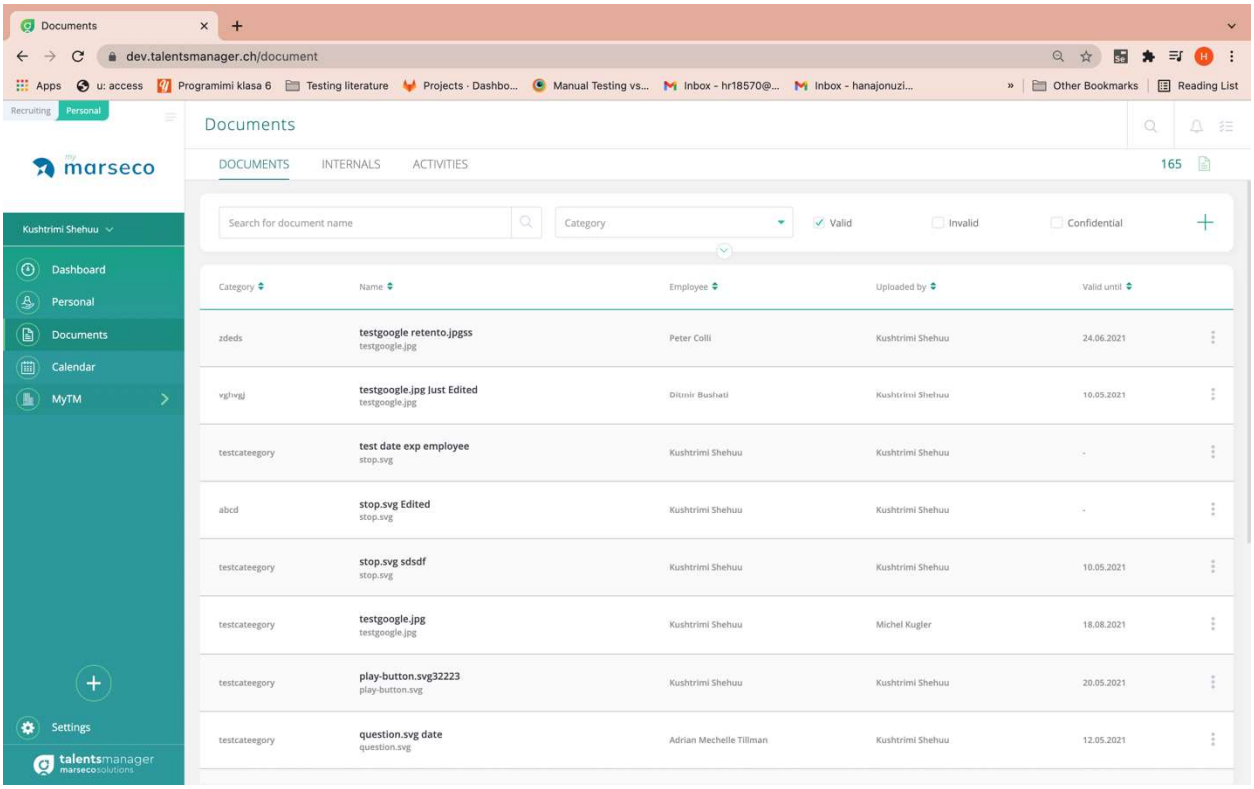


Fig.11. Talentsmanager Documents module

Calendar module is a general calendar that displays events related to the user where they can delete or edit an event.

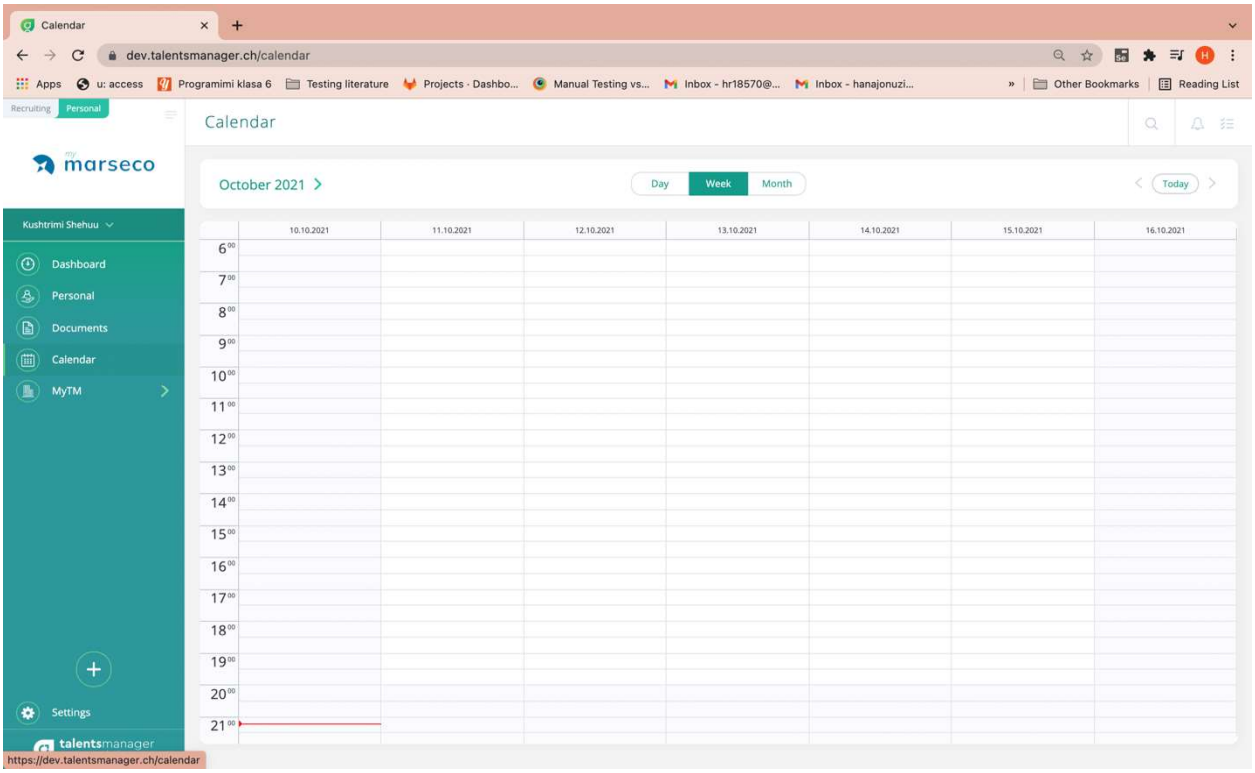


Fig.12. Talentsmanager Calendar module

3.2 Requirements

The application is designed to provide a solution for administration and organization of employees that in a broader term refers to a big infrastructure that can be hired by several companies to monitor their organization. In this context, the requirements of such a system can vary from tens to hundreds, and a lot of these requirements may be crucial requirements that are likely to appear as core features of the application or contrary to this many requirements are likely to remain as enhancements to the application that are not a priority neither to the development nor the stakeholders themselves.

For this thesis, the requirements were captured from documents that were published in GitLab and they are documented in a requirements traceability matrix that serve as basis for the developed test scenarios which in latter parts of the paper will be compared to evaluate how much automated or manual tests are covering them.

The requirements traceability matrix (RTM) is is a document that is widely used to generate a relation table between requirements of the stakeholder and the test cases that verify and validate those requirements. A single requirement may demand multiple test cases that need to be run in order to verify a feature or functionality.

The requirements are usually generated from stakeholders through several interviews, project documentations or informal discussions which is usually known as Requirements Elicitation that is an extensive phase of software development.

Beside verification and validation of requirements, RTM is also a practical way of tracking requirements through the development lifecyce and through connecting it with test cases it allows the dveleopment team not to miss any requirement or feature that has been asked for by the customer.

The requirements of this application can be found in Appendix A.

3.3. Methods

This part will present several testing scenarios that are designed according to stakeholder requirements and goals for the application in general.

The applications functionalities are tested against different scenarios and test scenarios belong to above mentioned modules of the application.

The first step is the manual black box testing that is run in Google Chrome web browser, and the following test cases depict several functionalities.

On the following chapter, the manual test cases are automated using Selenium IDE.

TS refers to Test Scenario and TC refers to Test Cases

3.3.1 Modules and their manual testing scenarios

1.The first module to be tested is the Login module:

***TS001.*User logs in to the system with correct username and password**.

Expected results: The user is able to log in to the system with a correct combination of username and password, or gets prompted when either or both of username or password is wrong.

Explanation: The user opens the page of Talentsmanager and tries to log in with a previously given username and password, if both are correct the user successfully logs in, if not the user gets prompted that the user is not found or the password is incorrect.

Created test cases:

TC.Login.001

TC.Login.002

TC.Login.003

TC.Login.004

TC.Login.005

TC.Login.006

TC.Login.007

See Appendix B.

***2.*Log out functionality:**

***TS002.*User logs out of the system when clicking on logout button**.

Expected result: The user is able to logout from the system just by clicking on the logout button and the session is closed afterwards.

Explanation: The user finds and clicks the logout button, then opens another tab to check if the session is closed.

Created test cases:

TC.Logout.001

TC.Logout.002

See Appendix C.

3.Search Box in Dashboard

***TS003.* Using the search box In the dashboard of the application to search for a keyword.**

Expected result: After entering a keyword on the search box a list containing related keyword expands under search box.

Explanation: User enters a keyword and checks if the search box functions properly to show results based on searched keyword.

Created test cases:

TC.Dashboard.001

TC.Dashboard.002

TC.Dashboard.003

TC.Dashboard.004

TC.Dashboard.005

TC.Dashboard.006

See Appendix D.

4.Notifications module in Dashboard

***TS004.C*hecking the functionality of Notifications in the dashboard**

Expected result: The system should display notifications and their settings should be editable by the user based on their preferences.

Explanation: The user clicks on notifications, checks if displayed notifications are related to events or todos and if notification settings are editable.

Created test cases:

TC.Notifications.001

TC.Notifications.002

TC.Notifications.003

TC.Notifications.004

TC.Notifications.005

TC.Notifications.006

TC.Notifications.007

TC.Notifications.008

TC.Notifications.009

TC.Notifications.010

TC.Notifications.011

TC.Notifications.012

TC.Notifications.013

TC.Notifications.014

TC.Notifications.015

TC.Notifications.016

TC.Notifications.017

TC.Notifications.018

TC.Notifications.019

TC.Notifications.020

See Appendix E.

5.Todos module in Dashboard

***TS005.*Create, edit and delete a todo from the menu displayed on the dashboard as Todos.**

Expected result: The user should be able to see their Todos as a list on dashboard, and if anyone of those is clicked the user can see them in a separate window. There the user can add, edit or delete a todo.

Explanation: The user scrolls in the Todos appearing in the dashboard, they can see the todos details in a new window, if they wish they should be able to edit or delete a todo by rightcliking on them. A new todo is possible to be created in the menu as well.

Test cases:

TC.TodoDash.001

TC.TodoDash.002

TC.TodoDash.003

TC.TodoDash.004

TC.TodoDash.005

TC.TodoDash.006

TC.TodoDash.007

See Appendix F.

6.Todo Overview page

***TS006.*Testing TODO component for displaying events (Todos) related to employees.**

Expected results: The Todo Overview page should display todos and user can either search for a todo or filter the todos based on options displayed.

Explanation: User opens Todos overview page and types keywords to see if related results will appear, also checks if the filtering options affect the results.

Test cases:

TC.TodoOver.001

TC.TodoOver.002

TC.TodoOver.003

TC.TodoOver.004

TC.TodoOver.005

TC.TodoOver.006

TC.TodoOver.007

TC.TodoOver.008

TC.TodoOver.009

See Appendix G.

7.Activities module of Dashboard

***TS007.*Checking the Activities component for displaying activities related to employees.**

Expected results: The activities component of dashboard should display all activities of employees, and they can be filtered based on categories and locations.

Explanation: User scrolls dashboard page to see Activities component, filters the displayed list of activities with different categories or location options displayed as dropdown menus.

Test cases:

TC.ActivitiesDash.001

TC.ActivitiesDash.002

TC.ActivitiesDash.003

TC.ActivitiesDash.004

TC.ActivitiesDash.005

TC.ActivitiesDash.006

See Appendix H.

8.Activities Overview Module

TS008: Displaying the activities details based on selected option from dropdown.

Expected results: The activities of all employees should be visible to the user based on a selected option from dropdown menu.

Explanation: User clicks on Open Activities and it should take the user to main page of Activities where they are displayed on a list, and the user can filter them based on a selected option from dropdown menu.

Test cases:

TC.ActivitiesOver.001

TC.ActivitiesOver.002

TC.ActivitiesOver.003

TC.ActivitiesOver.004

TC.ActivitiesOver.005

See Appendix I.

9.Personal Module

TS009: **Adding a new employee in Personal module, and checking the displayed information about employees, activities, archives.**

Expected results: User should be able to add a new employee with their information, and see their employees' information, activities and archives.

Explanation: User clicks on Personal tab on the left sidebar, and is able to add a new employee or search for employees, their activities or archives.

Test cases:

TC.Personal.001

TC.Personal.002

TC.Personal.003

TC.Personal.004

TC.Personal.005

TC.Personal.006

TC.Personal.007

TC.Personal.008

TC.Personal.009

TC.Personal.010

TC.Personal.011

Tc.Personal.012

See Appendix J.

10.Documents Module

TS010.Uploading a document to the Documents module of the application.

Expected results: The user should be able to add a new document to the system by either dragging and dropping or uploading via file browser, also should be able to search through documents.

Explanation: User clicks on Documents tab on left side bar, and can search on existing documents or add a new document by clicking on (+) button.

Test cases:

TC.Documentsl.001

TC.Documents.002

See Appendix K.

11.Calendar Module

TS011.Testing events displayed in the calendar.

Expected results: The events related to employees should be displayed in day, week and month view based on selected year.

Explanation: User clicks on Calendar tab on left sidebar, checks if the calendar has appropriate view modes (day, week, month) and if selected year changes events displayed on calendar.

Test cases:

TC.Calendar.001

TC.Calendar.002

See Appendix L

12.User interface

TS012.Testing interface elements in Dashboard and User Settings.

Expected results: The dashboard page should contain 5 components, a left sidebar should contain 4 tabs and settings is located there. The interface should comply to requested parameters.

Explanation: User checks if dashboard has required components, if there's a left sidebar and if it contains tabs and settings.

Test cases:

TC.Elements.001

TC.Elements.002

Tc.Elements.003

TC.Elements.004

TC.Elements.005

TC.Elements.006

See Appendix M

13.Settings Module

TS013.Adding an searching: a location, divison, role or document category in Settings page.

Expected results: User should be able to edit their administration settings by searching for a location, division, role or document category or adding one of them.

Explanation: User clicks on Administration settings and adds a new location, division, role or category for their employees. The user can also search for the above-mentioned elements with any keyword.

Test cases:

TC.Settings.001

TC.Settings.002

TC.Settings.003

Tc.Settings.004

Tc.Settings.005

Tc.Settings.006

Tc.Settings.007

Tc.Settings.008

Tc.Settings.009

Tc.Settings.010

Tc.Settings.011

Tc.Settings.012

Tc.Settings.013

Tc.Settings.014

Tc.Settings.015

See Appendix N

3.3.2 Automated test scripts

Above mentioned test scenarios and their test cases are automated in Selenium IDE with the same steps that are used in manual test cases. The tests are recorded as the user performs the steps from test cases and the recordings are re-run to check the functionalities.

The main idea for using Selenium IDE is because functional testing in this application is done in development environment which can be accounted as an alpha testing where the application gets tested by a random user without backend knowledge, thus record and play feature of Selenium IDE allows the user to perform simple steps (click, scroll, type a keyword etc.) to achieve a result which consequently serve as desired or required features/functionalities of the application.

The following are some of the automated test scripts which have been run in Selenium IDE:

- ***TS001.*** **User logs in to the system with correct username and password**

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestLoginTS001():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCLogin001(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()

  def test_tCLogin002(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("hanarustemi@hotmail.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.CSS_SELECTOR, ".align-items-center").click()

  def test_tCLogin003(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("12345678")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".align-items-center")
    assert len(elements) > 0

  def test_tCLogin004(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.NAME, "password").send_keys(Keys.ENTER)

  def test_tCLogin005(self):
    self.driver.get("https://dev.talentsmanager.ch/login")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()

  def test_tCLogin006(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()

  def test_tCLogin007(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("hanarustemi@hotmail.com")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()
```

***TS007.*Checking the Activities component for displaying activities related to employees.**

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestDashboardTS007():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCActivitiesDash002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1440, 791)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    WebDriverWait(self.driver, 30).until(expected_conditions.presence_of_element_located((By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header")))
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    assert self.driver.find_element(By.LINK_TEXT, "Todo").text == "Profile"

  def test_tCActivitiesDash003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1440, 791)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(4)").click()
    assert self.driver.find_element(By.LINK_TEXT, "Todo").text == "Todo"

  def test_tCActivitiesDash004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select__selected > span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tCActivitiesDash005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    assert self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(7) > .row").text == "Location"

  def test_tCActivitiesDash006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    assert self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(7) > .row").text == "Location"

  def test_tcActivitiesDash001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.execute_script("window.scrollTo(0,0)")
```

The created test scenarios in Selenium Ide can be exported as either Javascript or Python files and might be reused in different situations or when changes to the system are deployed, since Selenium Ide can be used for regression testing of an application these test scenarios can be used by the development team if necessary.

Further test scripts are available on Appendix C.

4. Results and Conclusion

Results

The testing process for https://dev.talentsmanager.ch/ as mentioned above has been conducted in two approaches:

- Manual
- Automated

for which, black box method has been used.

1.The manual and automated test results can be seen separately for each test scenario as PASS or FAIL:

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS001	TC.Login.001	Pass	Pass
	TC.Login.002	Pass	Pass
	TC.Login.003	Pass	Pass
	TC.Login.004	Pass	Pass
	TC.Login.005	Pass	Pass
	TC.Login.006	Pass	Pass
	TC.Login.007	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass /Fail	Pass/Fail
TS002	TC.Logout.001	Pass	Pass
	TC.Logout.002	Fail	Fail

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS003	TC.Dashboard.001	Pass	Pass
	TC.Dashboard.002	Pass	Pass
	TC.Dashboard.003	Fail	Fail
	TC.Dashboard.004	Pass	Pass
	TC.Dashboard.005	Fail	Fail
	TC.Dashboard.006	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS004	TC.Notifications.001	Fail	Fail
	TC.Notifications.002	Pass	Pass
	TC.Notifications.003	Pass	Fail

	TC.Notifications.004	Pass	Fail
	TC.Notifications.005	Pass	Fail
	TC.Notifications.006	Pass	Fail
	TC.Notifications.007	Pass	Fail
	TC.Notifications.008	Pass	Fail
	TC.Notifications.009	Pass	Fail
	TC.Notifications.010	Pass	Fail
	TC.Notifications.011	Pass	Fail
	TC.Notifications.012	Pass	Fail
	TC.Notifications.013	Pass	Fail
	TC.Notifications.014	Pass	Fail
	TC.Notifications.015	Pass	Fail
	TC.Notifications.016	Pass	Fail
	TC.Notifications.017	Pass	Fail
	TC.Notifications.018	Pass	Fail
	TC.Notifications.019	Pass	Fail

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS005	TC.TodoDash.001	Pass	Pass
	TC.TodoDash.002	Pass	Pass
	TC.TodoDash.003	Fail	Fail
	TC.TodoDash.004	Fail	Fail
	TC.TodoDash.005	Pass	Fail
	TC.TodoDash.006	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS006	TC.TodoOver.001	Pass	Pass
	TC.TodoOver.002	Pass	Pass
	TC.TodoOver.003	Pass	Fail
	TC.TodoOver.004	Pass	Pass
	TC.TodoOver.005	Pass	Pass
	TC.TodoOver.006	Pass	Pass
	TC.TodoOver.007	Pass	Pass
	TC.TodoOver.008	Fail	Fail
	TC.TodoOver.009	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS007	TC.ActivitiesDash.001	Pass	Pass

	TC.ActivitiesDash.002	Fail	Fail
	TC.ActivitiesDash.003	Pass	Fail
	TC.ActivitiesDash.004	Pass	Pass
	TC.ActivitiesDash.005	Fail	Fail
	TC.ActivitiesDash.006	Fail	Fail

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS008	TC.ActivitiesOver.001	Pass	Pass
	TC.ActivitiesOver.002	Fail	Fail
	TC.ActivitiesOver.003	Pass	Pass
	TC.ActivitiesOver.004	Pass	Pass
	TC.ActivitiesOver.005	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS009	TC.Personal.001	Pass	Pass
	TC.Personal.002	Pass	Pass
	TC.Personal.003	Pass	Pass
	TC.Personal.004	Pass	Pass
	TC.Personal.005	Pass	Pass
	TC.Personal.006	Pass	Pass
	TC.Personal.007	Pass	Pass
	TC.Personal.008	Fail	Fail
	TC.Personal.009	Pass	Pass
	TC.Personal.010	Pass	Pass
	TC.Personal.011	Pass	Pass
	Tc.Personal.012	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS010	TC.Documentsl.001	Pass	Fail
	TC.Documents.002	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS011	TC.Calendar.001	Pass	Pass
	TC.Calendar.002	Pass	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS012	TC.Elements.001	Pass	Pass
	TC.Elements.002	Pass	Pass
	Tc.Elements.003	Pass	Pass
	TC.Elements.004	Pass	Pass
	TC.Elements.005	Fail	Fail
	TC.Elements.006	Fail	Pass

Test Scenario	Test Case	Manual Testing	Automated Testing
		Pass/Fail	Pass/Fail
TS013	TC.Settings.001	Pass	Pass
	TC.Settings.002	Pass	Pass
	TC.Settings.003	Pass	Pass
	Tc.Settings.004	Fail	Pass
	Tc.Settings.005	Pass	Pass
	Tc.Settings.006	Pass	Pass
	Tc.Settings.007	Pass	Pass
	Tc.Settings.008	Pass	Pass
	Tc.Settings.009	Pass	Pass
	Tc.Settings.010	Pass	Pass
	Tc.Settings.011	Pass	Pass
	Tc.Settings.012	Pass	Pass
	Tc.Settings.013	Pass	Pass
	Tc.Settings.014	Pass	Pass
	Tc.Settings.015	Pass	Pass

In total, 13 Test Scenarios and 97 Test Cases have been created and run, and results are:

In manual testing:

69 test cases are recorded as Pass which means that the regarding functionality is working properly and the process is successful.

28 test cases have Failed which means that the desired functionality is not working properly and fails to fulfill a requirement.

In percentage, 71.13% of test cases have Passed and 28.87 % have Failed the manual testing process.

In automated testing:

63 test cases were automated successfully which means that the required functionality is working properly.

34 test cases have Failed from which 10 failed because IDE could not detect Interface errors such as: ADD TODO label should be EDIT TODO, some of search results did not match the query, verification alerts could not be detected. 17 of the test cases failed because the application did not allow interaction of page with Selenium IDE.

In percentage 64.95% of test cases have Passed and 35.05% have Failed the automated testing process.

Requirements Coverage

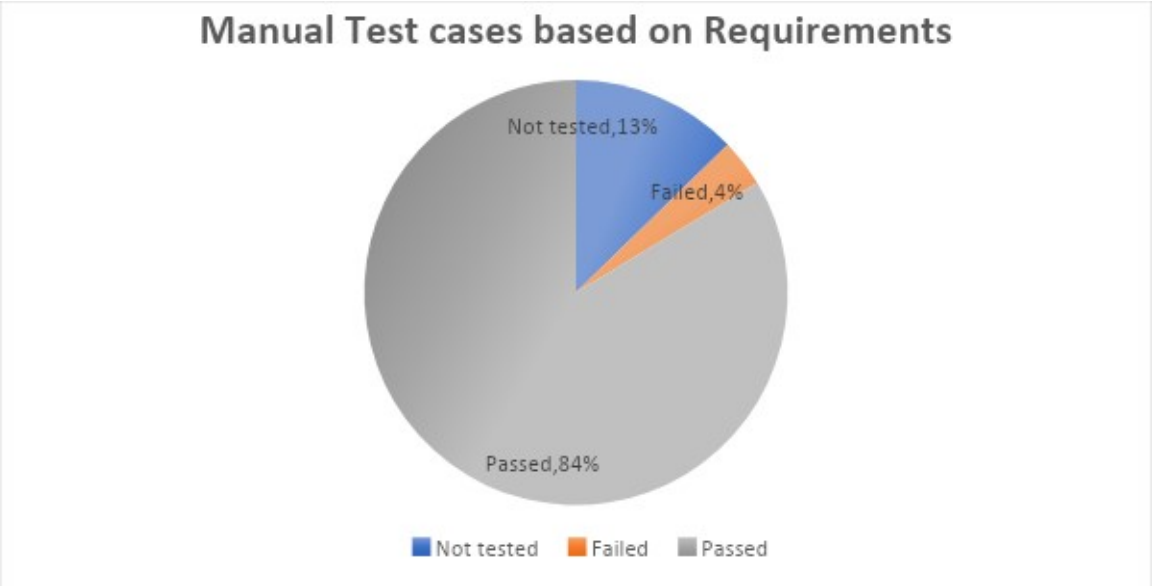
The requirements for this research have been documented and shown in the Appendix A, where 55 requirements in total have been saved.

Based on the requirements, initially manual test cases have been derived and then these manual tests have been automated.

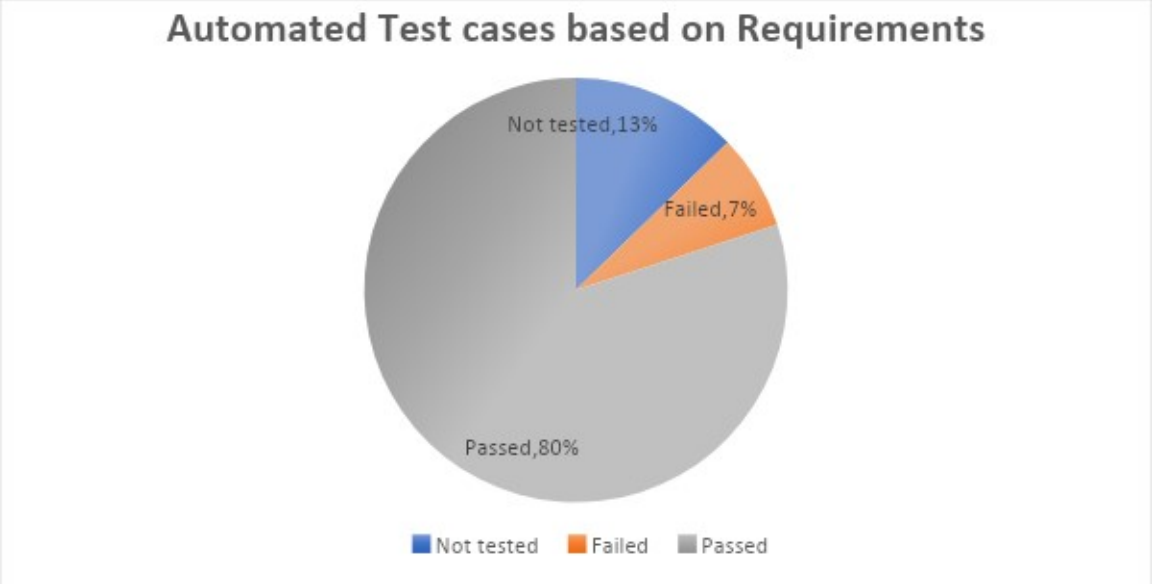
As for requirements coverage, it is an important metric that shows the tester whether the test suites are compatible to the requirement or if the test case really tests the desired requirement, because in many situations the tester can extend the test cases to excess which not just surpasses the requirements but results in more time spent without need. Thus, requirements coverage assures that the designed test cases comply with requirements that are put forward.

Based on the results of manual and automated tests, we can state that:

For 55 requirements 7 of requirements did not have an adequate manual test case and could not be tested, 2 tests Failed and 46 test cases passed.



This chart displays the percentage of requirements that are covered from manual test cases which are shown as Passed and Failed and the not tested part shows that 13% of the requirements were not covered by the test cases and 3 requirements failed manual testing.



This chart displays the percentage of requirements that are covered from automated test cases which are shown as Passed and Failed and the not tested part shows that 13% of the requirements were not covered by the test cases and 7% of the requirements failed automated testing.

Comparison of results:

Manual functional testing is a testing strategy where tester evaluates requirements, creates test cases, runs them and records their results to check if they meet the requirements of the stakeholders. In this situation several mistakes can be done because a functionality might

change during the time the tester tests it, a requirement might be missed from the tester, or a small mistake on any step of test case can result in errors not being seen from the tester.

On the other hand, automation process requires similarly requirements evaluation, creating test cases but the magic of automation happens afterwards where after carefully creating test cases user can run them and can easily detect any mistakes if a step of test case is missing or wrong.

The main difference between these approaches is that, manual testing is conducted on testers lead which includes instinct and critical eye which cannot be expected from automation tools and functional testing in some cases can include some complex steps which generally cannot be handled by automation tools.

In general, results have shown us that, functional testing when conducted with manual test cases, takes approximately 4 hours of work per day, for 5 days meaning 20 hours in total, and both automation of these cases and running them takes 3 hours of work per day for 3 days in total 9 hours.

28.87 % of manual tests have failed because of missing functionalities and 35.05% of automated tests have failed where nearly half of the failed test cases could not be conducted properly by automation tool because of above mentioned reasons (Results section), thus if we consider this, we can say that error finding rate of automation testing drops in half (approximately 17.02%).

Also, there are some errors that are missed by automation tools but discovered while manual testing, they are as follow:

ID	PROCESS	Note:
TC.Logout.001	LOGOUT	The Logout button is only visible when clicking on tab where users' name appears!
TC.Dashboard.003	SEARCH	Only displays seacrhed items if one of the related results is clicked
TC.Notifications.008	NOTIFICATIONS	The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.009	NOTIFICATIONS	The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.

TC.TodoDash.004	TODO	the page name remains add todo, when editing a todo, re editing doesn't change the name of todo!
TC.TodoOver.005	Todos Overview	Buttons are toggled by defautl
TC.TodoOver.006	Todos Overview	Buttons are toggled by defautl
TC.ActivitiesDash.002	ACTVITIES	Some of results do not have the Profile tag
TC.ActivitiesDash.005	ACTVITIES	The results don't change
TC.ActivitiesOver.002	Activities Overview	Some of Results do not contain Profie tag
TC.Personal.008	PERSONAL	MOST OF THE EMPLOYEES ARE NOT DISPLAYED WHEN ALL OPTIONS ARE SELECTED
TC.Elements.006	SETTINGS	Settings tab in left side remains open

Conclusion

When comparing manual and automated tests, it would be irrational to ignore the opportunities and pros of automated testing where the user simply can interact with a web application as if they were naturally navigating it and record these interactions within a short time period, at the same time the documentation and tracking of the cases is easily done through automation tools, but the problem stands that in functional testing some requirements need more complex test cases which inevitably require human interaction to detect different errors that might be missed from automation tools, also as seen on the above table, some of the automated test cases for the corresponding processes have resulted as PASS but they have not been able to discover the errors that are discovered by manual tests.

As for the Hypotheses of this research, we can state the following conclusions about them:

H0 stating that automated test gives best results in software testing can be partially true since automation only results in faster testing and allows user to reuse their test cases both for the development team and the tester but yet it requires test cases being prepared ahead and in our case, it did not fulfill the requirements being tested.

H1 stating that manual functional testing of a web application results in better error finding cannot be proven true since it is a slow process when compared to automation and the test cases are only written in informal language so it is no use for developers to test their applications on back end, but aside these manual testing gave better error finding results compared to automation tests.

Thus, the H3 can be proved right where we state that combining both manual and automated tests for functional testing can give better results by: enhancing time spent in test case running and ensuring that interface functionality errors or logic errors are captured by human interaction.

References

1. Pressman, S., Roger, Maxim, Bruce R., *Software Engineering A practitioner's approach*, Eighth Edition.

2. Myers, Glenford J., 1946-The art of software testing / Glenford J. Myers, Corey Sandler, Tom Badgett. — 3rd ed.

3. Chintakayala, Padmini ,2019, *Begginers guide to Software testing*

4. Strazdiņa, L., Arnicane, V., Arnicans, G., Bičevskis, J., Borzovs, J., Kuļešovs, I.,

What Software Test Approaches, Methods, and Techniques are Actually Used in Software Industry,

5. Padayachee, Indira & Kotzé, Paula & Van, A & Van der Merwe, Alta. (2010*). ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems.*

6. M. A. Jamil, M. Arif, N. S. A. Abubakar and A. Ahmad, "Software Testing Techniques: A Literature Review," 2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M), 2016, pp. 177-182, doi: 10.1109/ICT4M.2016.045.

7. Buenaflor, L., ISO 9126 Software Quality Characteristics https://medium.com/@leanardbuenaflor/iso-9126-software-quality-characteristics-a25a26e7d046

8. Farooq, S. *Gap between academia and industry: a case of empirical evaluation of three software testing methods*. Int J Syst Assur Eng Manag 10, 1487–1504 (2019). https://doi-org.uaccess.univie.ac.at/10.1007/s13198-019-00899-2

9. Ghuman, S.S. (2015). *Software Testing Techniques.*

10. Khan, Mohd. (2010*). Different Forms of Software Testing Techniques for Finding Errors.* International Journal of Computer Science Issues. 7.

11. Kumari, K. (2014*). Software Testing for Finding Errors*. International Journal of Scientific Research and Education

12. Saini, G., & Rai, K. (2013). *An Analysis on Objectives, Importance and Types of Software Testing.* IJCSMC, Vol. 2, Issue. 9, September 2013, pg.18 – 23

13. Sethi, M. (2017). *A REVIEW PAPER ON LEVELS, TYPES & TECHNIQUES IN SOFTWARE TESTING*. International Journal of Advanced Research in Computer Science, 8(7), 269-271. doi: https://doi.org/10.26483/ijarcs.v8i7.4236

14. Sawant, A., Bari, P.H., & Chawan, P. (2012). *Software Testing Techniques and Strategies.* International Journal of Engineering Research and Applications (IJERA. ISSN: 2248-9622 www.ijera.com.Vol. 2, Issue 3, May-Jun 2012, pp.980-986

15. Umar, Mubarak Albarka (2020): *A Study of Software Testing: Categories, Levels, Techniques, and Type*s. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.12578714.v1

16. Rosenberg, Linda H., Phd Theodore F. Hammer, Huffman, Lenore L. *Requirements, Testing, and Metrics.* In 15th Annual Pacific Northwest Software Quality Conference},1998

17 Whalen, Michael & Rajan, Ajitha & Heimdahl, Mats & Miller, Steven. (2006*). Coverage metrics for requirements-based testing*. 2006. 25-36. 10.1145/1146238.1146242.

18. Nalbant, E.- *The importance of testing in software development life cycle and the implementation of a test automation*. 2020-11-08 19:32:53

19. Kuday, G. *SOFTWARE TEST PROCESS IN SOFTWARE ENGINEERIN. 2015, published 2020-02-29.*

20. Iren.G, *REVIEW OF SOFTWARE DEVELOPMENT AND TESTING PROCESSES*. 2020--11-08

21. Hanoglu, E., *A TESTABILITY ANALYSIS METHOD BASED ON INTERNAL ATTRIBUTES AND METRICS FOR OPEN-SOURCE SOFTWARE PROJECTS*,2019.

22. The Power of Black Box Testing, https://applitools.com/black-box-testing-5-tools-used-by-the-pros/#test-auto

23. Eriksson, U., The A-to-Z Guide to the Software Testing Process https://reqtest.com/testing-blog/the-a-to-z-guide-to-the-software-testing-process/

24. Sommerville, I. (2019). *Software engineering, tenth edition*. Harlow: Pearson Education.

25 Shiklo, B.,8 *Software Development Models: Sliced, Diced and Organized in Charts* https://www.scnsoft.com/blog/software-development-models

26. The Standish Group Chaos Report https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

27. Survey Project Management Institute (PMI) PMI's Pulse of the Profession https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

28. Beck K., Beedle M., Van Bennekum A., Cockburn A., Cunningham W., Fowler M, Grenning J., Highsmith J., Hunt A, Jeffries R., Kern J., Marick B.,. Martin R.C., Mellor S., Schwaber K., Sutherland J., Thomas D. *Manifesto for Agile Software Development* https://agilemanifesto.org/

29. Choodi , Sh., S., *Testing in Iterative Product Development Environment*. 6th Annual International Software Testing Conference 2006

30. Rajan A., Whalen M., Staats M., Heimdahl M.P.E. (2008) *Requirements Coverage as an Adequacy Measure for Conformance Testing.* In: Liu S., Maibaum T., Araki K. (eds) Formal Methods and Software Engineering. ICFEM 2008. Lecture Notes in Computer Science, vol 5256. Springer, Berlin, Heidelberg. https://doi-org.uaccess.univie.ac.at/10.1007/978-3-540-88194-0_8

31. Cigniti, *37 Epic Software Failures that Mandate the Need for Adequate Software Testing* https://www.cigniti.com/blog/37-software-failures-inadequate-software-testing/

32. Gelperin, David & Hetzel, Bill. (1988). *The growth of software testing*. Communications of the ACM. 31. 687-695. 10.1145/62959.62965.

33 The National Bureau of Standards *Guideline for lifecycle validation, Verification, and testing of computer software.* https://www.govinfo.gov/content/pkg/GOVPUB-C13-886cadfdbff83e49dfdba09a89cd49cf/pdf/GOVPUB-C13 886cadfdbff83e49dfdba09a89cd49cf.pdf

34. Ammann, P., & Offutt, J. (2017). *Introduction to software testing.*

35 Graham, D., Black, R., Veenendaal, E.., & International Software Testing Qualifications Board. (2020*). Foundations of software testing: ISTQB certification.*

Appendices

Appendix A – Traceability Matrix

Traceability Matrix

Project Name: TALENTSMANAGER
Project Number:
Project Manager:

Purpose: To manage requirements throughout the software development lifecycle. The Traceabilty Matrix ensures that requirements are captured in the design, implemented in the code, and verified by testing.

Use Case	Requirement Number	Requirement Description	Status	Test Case	Design Component / Module	Comments
	1	The user should be able to login with a given username and password.	done	TC.Login.001, TC.Login.002	LOGIN	
	2	The system should prompt if either username or password is wrong	done	TC.Login.002	LOGIN	
	3	The system should allow the user to change their password by clicking on "forgot your password.	done	TC.Login.003	LOGIN	
	4	Forgot your password should send a verification e mail to users e mail address.		TC.Login.005, TC.Login.006, TC.Login.007	LOGIN	
	5	The user should be able to logout by clicking on log out button.		Tc.Logout.001	LOGOUT	
	6	The system should end the session if the tab is closed.		Tc.Logout.002		
	7	The dashboard should be divided in 5 main parts: Todos, Contacts, Events, Your Team, Activities.		Tc.Elements.001	DASHBOARD	
	8	There should be.a navigation bar for accessing Personal, Documents, Calendar, My Team.		Tc.Elements.002	DASHBOARD	
	9	The user should be able to create a todo or create an employee.		TS	DASHBOARD, PERSONAL	
	10	The user should have a settings page for their Personal Settings, Company Settings and Administration.		Tc.Elements.003	DASHBOARD	
	11	There should be a search bar for general search in the page.		Tc.Elements.004	DASHBOARD	
	12	The user should be notified for Todos if they are complete, past, or surpassed.			TODO	
	13	Todo link in notifications should be redirected to the todo mentioned in the notification		Tc.Notifications.001	TODO	
	14	The system should work on a calendar basis.		Tc.Calendar.001	TODO	
	15	The system should display each event, holiday, todo in a calendar.		Tc.Calendar.001	CALENDAR	
	16	public holidays are displayed in the calendar and event map accordingly.		Tc.Calendar.001	CALENDAR	
	17	The system should visualize holidays based on the Public Holiday configuration,		Tc.Calendar.001	CALENDAR	
	18	The system should display calendar on three bases: day, week, month, year.		Tc.Calendar.002	CALENDAR	
	19	System should display Profile of the employee with related tabs: Profile,Details,Contacs, Activities in the top nav bar..		Tc.Personal.012	EMPLOYEE PROFILE	
	20	-Profile of an employee should include information abot their allocation, experience,documents,absences,shifts and notes.		Tc.Personal.012	EMPLOYEE PROFILE	
	21	-Allocation in the Profile tab should allow the user to edit the employees allocations.		Tc.Personal.012	EMPLOYEE PROFILE	
	22	-Experience in the Profile tab should display employees' driver licences, language skills, overall skills.professional experience,education and further training.		Tc.Personal.012	EMPLOYEE PROFILE	
	23	-Documents in the Profile tab should display employees' related documnts.		Tc.Personal.012	EMPLOYEE PROFILE	
	24	-Notes in the Profile tab should allow the user to enter notes for the employee.		Tc.Personal.012	EMPLOYEE PROFILE	
	25	-Activities in the Profile tab should display employees' activities.		Tc.Personal.012	EMPLOYEE PROFILE	

Traceability Matrix

Project Name: _____
Project Number: _____
Project Manager: _____

Purpose: To manage requirements throughout the software development lifecycle. The Traceabilty Matrix ensures that requirements are captured in the design, implemented in the code, and verified by testing.

Use Case	Requirement Number	Requirement Description	Status	Test Case	Design Component / Module	Comments
	26	-The user should be able to add new employee with their details about personal information, contact information,working days,drivers's licences, languages, skills, professional experience and education.		Tc.Personal.010	EMPLOYEE PROFILE	
	27	-The employee is plannable during his working days.Absences, holidays.sickness/accidents are visualized in plan.		Tc.Personal.011	EMPLOYEE PROFILE	
	28	-The entire handling of ID, initials, public holidays,views, conflicts etc. is managed via the settings.		Tc.Personal.012	EMPLOYEE PROFILE	
	29	-The id. And initial handling is determined via the location and used on basis of Einsatzplanung configuration (settings).		Tc.Personal.013	EMPLOYEE PROFILE	
	30	The system should be able to save documents created by the user.		TC.Documents.001	DOCUMENTS	
	31	The system should allow the user to create documents.		TC.Documents.001	DOCUMENTS	
	32	The system should allow the user to search and find documents.		Tc.Documents.002	DOCUMENTS	
	33	The system should display activities related to documents (create, delete, edit etc)		Tc.Documents.002	DOCUMENTS	
	34	If an employee is archived, no more notifications must be sent to this user and any other user about this employee.			NOTIFICATIONS	
	35	The system should allow the user to edit their Personal Settings, Company Settings and Administration Settings.		Tc.Settings.001	USER SETTINGS	
	36	The system should be available in two languages (English and German) and this can be selected in Personal Settings.		Tc.Settings.002	USER SETTINGS	
	37	The user should be able to change their password in Personal Settings.		Tc.Settings.003	PERSONAL SETTINGS	
	38	System should allow user to set their Notification preferences in Personal Settings.		TS004	PERSONAL SETTINGS	
	39	The sysem should let users integrate their Google or Office calendar to talentsmanager.		Tc.Settings.004	PERSONAL SETTINGS	
	40	The system should display divisions, departments, positions and contacts to the user.		TC.Settings.008	DIVISION	
	41	The system should let the user add or search divisions, departments, positions, contacts based on their needs.		Tc.Settings.005 Tc.Settings.006 Tc.Settings.007 Tc.Settings.008 Tc.Settings.009 Tc.Settings.010 Tc.Settings.011	DIVISION	
	42	The user should be able to define roles and permissions to their employees.		Tc.Settings.012	EMPLOYEE	
	43	The user should be able to search through roles and permissions.		Tc.Settings.013	EMPLOYEE	
	44	The system should let the user search for document categories.		Tc.Settings.014	CATEGORIES	
	45	The system should let the user create a new document category.		Tc.Settings.015	CATEGORIES	
	46	-the user should be able to add a new location		Tc.Settings.007	LOCATION	
	47	-the id field on the screen should load along with the screen and cannot be changed bz the user.		Tc.Settings.007	LOCATION	
	48	-public holidays tab should be next to location tab.		Tc.Settings.007	LOCATION	
	49	-Add location page should have a local navigation map integrated.		Tc.Settings.007	LOCAITON	
	50	-Public holidays tab should have two options Undefined and Defined,		Tc.Settings.007	LOCATION	
	51	-In the years dropdown you should go back from current year to 2050.		Tc.Settings.007	LOCATION	
	52	-The user should be able to select a country and a province if necessary.		Tc.Settings.007	LOCATION	
	53	-There should be a list showing holidays with date, day, designation, day off, annual.		-	HOLI\	
	54	-The user should be able to add a personal public holiday.		-	HOL	
	55	-There should be a context menu where the user can edit or remove a holiday.		-	HOLIDAY	
	56	The system should display a separate tab where the user can see their company, profile, documents and team or contacts.		-	COMPANY	

Appendix B – Test Scenarios

TEST SCENARIO TS001					
'PROJECT NAME	TALENTSMANAGER		DATE		
EST SCENARIO DESCRIPTIO	A user can log in to the system with correct username/password, and cannot log with wrong username and/or password		REQUIREMENTS		1,2,3,4,5
MODULE	LOGIN		TEST TYPE		
GOAL	The system should work properly with verified accounts (login and password)				
TEST SETUP					
DEPENDENCIES					
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA
TC.Login.001	LOGIN	Enter Url in browser Type username Type Password Click log in	The user should log in successfully	PASS	kinurabi@yahoo.com 123456
TC.Login.002	LOGIN	Enter Url in browser Type username Type Password Click log in	The user cannot login with wrong username and correct password	PASS	hanajonuzi@gmail.com 123456
TC.Login.003	LOGIN	Enter Url in browser Type username Type Password Click log in	The user cannot login with correct username and wrong password and display a message "The login credentials (username or password) are not correct"	PASS	kinurabi@yahoo.com 12345678
TC.Login.004	LOGIN	Enter Url in browser Type username Type Password Press ENTER	Enter key should function as login button	PASS	/
TC.Login.005	LOGIN	Enter Url in browser Click Forgot Password	Forgot password button should take user to a next page for writing their email to reset their password	PASS	/
TC.Login.006	LOGIN	Enter Url in browser Click Forgot Password Enter a valid e mail address Click reset password	Writing a valid email address should send a password reset link to the corresponding e mail	PASS	kinurabi@yahoo.com
TC.Login.007	LOGIN	Enter Url in browser Click Forgot Password Enter a invalid e mail address Click reset password	Writing an invalid email address should send a password reset link to the corresponding e mail	PASS	hanajonuzii@gmail.com
	π				

TEST SCENARIO TS002						
'PROJECT NAME	TALENTSMANAGER		DATE			
TEST SCENARIO DESCRIPTION	Testing the systems' logout functionality		REQUIREMENTS	5,6		
MODULE			TEST TYPE			
GOAL	The system should logout the user when clicked on Logout button and close the session					
TEST SETUP						
DEPENDENCIES						
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Note:
TC.Logout.001	LOGOUT	Click on Logout Button	The user should be logged out of their account	PASS		The Logout button is only visible when clicking on tab where users' name appears!
TC.Logout.002	LOGOUT	Click on Close the tab Open a new tab Type the Url of the page	The session should be closed if the tab is closed	FAIL		

TEST SCENARIO TS003						
'PROJECT NAME	TALENTSMANAGER		DATE			
TEST SCENARIO DESCRIPTION	TESTING THE SYSTEMS DASHBOARD FUNCTIONALITY		REQUIREMENTS	11		
MODULE	DASHBOARD-SEARCH		TEST TYPE			
GOAL	The system should allow the user to search for any keyword, personnel, document etc.					
TEST SETUP						
DEPENDENCIES						
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	
TC.Dashboard.001	SEARCH	Look for Search Box	There should be a search box in the top of the page	PASS		
TC.Dashboard.002	SEARCH	Click on search box over Dashboard Type a name	The search box should display results matching the keyword?	PASS	VALON	
TC.Dashboard.003	SEARCH	Click on search box over Dashboard Check if it displays last 5 searches	When clicked on empty search box ,itshould display last 5 searches.	FAIL		Only displays seacrhed items if one of the related results is clicked
TC.Dashboard.004	SEARCH	Click on search box over Dashboard Type a name Click on of the names in the results	When clicking a result from a search (a user profile) it should take the user to the corresponding employees page.	PASS	VALON	
TC.Dashboard.005	SEARCH	Click on search box over Dashboard Type a name Press ENTER on keyboard	When typing a keyword on Search box, pressing ENTER should process the search and keep the results page open	FAIL	KUSHTRIM	
TC.Dashboard.006	SEARCH	Click on search box over Dashboard Type a name Click cancel/cross button	When typing a name on searchbox, clicking cancel/x button should abort the search.	PASS	KUSHTRIM	

TEST SCENARIO TS004

'PROJECT NAME	TALENTSMANAGER	DATE	
TEST SCENARIO DESCRIPTION	TESTING THE SYSTEMS DASHBOARD FUNCTIONALITY	REQUIREMENTS	12,13
MODULE	DASHBOARD-NOTIFICATIONS	TEST TYPE	
GOAL	The system should display notifications and their settings should be editable by the user based on their preferences.		
TEST SETUP			
DEPENDENCIES			

ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Notes!
TC.Notifications.001	NOTIFICATIONS	Click on Notifications Click on a single notification	When clicking a notification for a TODO, it should take the user to the Todos Overview page.	FAIL		
TC.Notifications.002	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots	Clicking on Three horizontal dots for settings should open the settings page for Notifications	PASS		NOTIFICATIONS TAB REMAINS OPEN!
TC.Notifications.003	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on Expand icon for TODO	Clcking on the expand Arrow of TODO should open notification settings for TODO	PASS		
TC.Notifications.004	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on Expand icon for TODO Click Activate all inApp	Clicking Activate all inApp radiobutton should automatically check every All inApp radiobuttons below	PASS		
TC.Notifications.005	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on Expand icon for TODO Click Activate all email	Clicking Activate all email radiobutton should automatically check every email radiobuttons below	PASS		
TC.Notifications.006	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for TODO Click Activate all push notifications	Clicking Activate all push notifications radiobutton should automatically check every All push radiobuttons below	PASS		
TC.Notifications.007	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Documents	Clicking the expand arrow of Documents should open the notification settings for Documents	PASS		
TC.Notifications.008	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Documents Click Activate all inApp	Clicking Activate all inApp radiobutton should automatically check every All inApp radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.009	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Documents Click Activate all email	Clicking Activate all email radiobutton should automatically check every email radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are
TC.Notifications.010	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Documents Click Activate all push notifications	Clicking Activate all push notifications radiobutton should automatically check every All push radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.011	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Profile	Clicking the expand arrow of Profile should open the notification settings for Profile	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.012	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Profile Click Activate all inApp	Clicking Activate all inApp radiobutton should automatically check every All inApp radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.013	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Profile Click Activate all email	Clicking Activate all email radiobutton should automatically check every email radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.014	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Profile Click Activate all push notifications	Clicking Activate all push notifications radiobutton should automatically check every All push radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.015	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes	Clicking the expand arrow of Notes should open the notification settings for Notes	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.016	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes Click Activate all inApp	Clicking Activate all inApp radiobutton should automatically check every All inApp radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.017	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes Click Activate all email	Clicking Activate all email radiobutton should automatically check every email radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.018	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes Click Activate all push notifications	Clicking Activate all push notifications radiobutton should automatically check every All push radiobuttons below	PASS		The Activate all inApp,Activate all email and Activate all push notifications is not checked but the related radiobuttons below are checked randomly.
TC.Notifications.019	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes Click Activate all push notifications Click Save	Clicking Save after changing notification settings should save the changes and notify the user	PASS		
TC.Notifications.020	NOTIFICATIONS	Click on Notifications Click on Three horizontal dots Click on expand icon for Notes Click Activate all push notifications Click Cancel	Clicking Cancel should abort the changes	PASS		

TEST SCENARIO TS005

'PROJECT NAME	TALENTSMANAGER	DATE	
TEST SCENARIO DESCRIPTION	Testing the TODO component for displaying events related to employees.	REQUIREMENTS	12,13,14
MODULE	DASHBOARD-TODO	TEST TYPE	
GOAL	The Todo Module should appear as a component of the dashboard, and the todo module should display events related to employees and users.		
TEST SETUP			
DEPENDENCIES			

ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Note:
TC.TodoDash.001	TODO	Scroll the bar to see the Todo list	The Todos list should show the Todos starting from today and backwards	PASS		
TC.TodoDash.002	TODO	Click on Open Todos	Clicking on Open Todos should open the Todos page with a calendar like a sidebar from right	PASS		
TC.TodoDash.003	TODO	Click on Open Todos Click on a date from calendar	Clicking on a date from the calendar in Todo page, should display todos related to that date	FAIL	CLICK ON 23.09.2021	
TC.TodoDash.004	TODO	Clcik on Open Todos Click on ellipsis near a todo Click edit	Clicking on ellipsis near a todo then EDIT should open the editing page under todos.	FAIL		the page name remains add todo, when editing a todo, re editing doesn't change the name of todo!
TC.TodoDash.005	TODO	Click on Open Todos Click on three dots(ellipsis) near a todo Click delete	Clcking on ellipsis near a todo then Delete should delete the todo	PASS		
TC.TodoDash.006	TODO	Click on Open Todos Click on plus sign in the bottom Type a word Select a date Click Save	The user should be able to add a todo by writing a keyword, selecting a date and saving it.	PASS		
TC.TodoDash.007	TODO	Click on Open Todos Click on the arrow on todos bar	The arrow in the todos bar should take the user to TODO Overview page	PASS		

TEST SCENARIO TS006

'PROJECT NAME	TALENTSMANAGER	DATE	
TEST SCENARIO DESCRIPTION	Testing the TODO component for displaying events related to employees.	REQUIREMENTS	
MODULE	Todos Overview	TEST TYPE	
GOAL	The overview page should display todos and the user can either search for a todo or filter the todos based on options.		
TEST SETUP			
DEPENDENCIES	TS005		

ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Note:
TC.TodoOver.001	Todos Overview	Click on Open Todos Click on the arrow on todos bar	The arrow in the todos bar should take the user to TODO Overview page	PASS		
TC.TodoOver.002	Todos Overview	Click on Search Type a keyword in lowercase	Typing a keyword on search bar should display results related to the lowercase keyword	PASS	test	
TC.TodoOver.003	Todos Overview	Click on Search Type a keyword in uppercase	Typing a keyword on search bar should display results related to the UPPERCASE keyword	PASS	TEST	The app doesn't support uppercase text
TC.TodoOver.004	Todos Overview	Toggle the My Todos option	Toggling My Todos should display only Todos related to the user itself	PASS		
TC.TodoOver.005	Todos Overview	Select Open radio button	Selecting Open radiobutton should display only open todos	PASS		Buttons are toggled by defautl
TC.TodoOver.006	Todos Overview	Select All radio button	Selecting All radiobutton should display All todos	PASS		Buttons are toggled by defautl
TC.TodoOver.007	Todos Overview	Select Done radiobutton	Selecting Done radiobutton should display only done todos	PASS		
TC.TodoOver.008	Todos Overview	Click arrow to expand the menu Select a date From and To	Selecting a date from and to should display todos between that time range	FAIL	some of todos displayed are out of range	
TC.TodoOver.009	Todos Overview	Click arrow to expand the menu Select a date From and To Select Include in time range Start Date and End date Select Type All Select From You Select Assigned to Other than me	Selecting the optionns from dropdown menus should display results based on preferences			
		'The language in date picker remains German				

TEST SCENARIO TS007						
`PROJECT NAME	TALENTSMANAGER			DATE		
TEST SCENARIO DESCRIPTION	Testing the ACTIVITIES component for displaying activities related to employees.			REQUIREMENTS		
MODULE	ACTIVITIES-DASHBOARD			TEST TYPE		
GOAL	The Activities part of Dashboard should display the activities related to the user with options for filtering					
TEST SETUP						
DEPENDENCIES						
ID	PROCESS	STEP	EXPECTED RESULT		PASS/FAIL	INPUT DATA	Note:
TC.ActivitiesDash.001	ACTVITIES	Scroll the bar	The activities should be dipslayed by default from latest to oldest		PASS		The Logout button is only visible when clicking on tab where users' name appears!
TC.ActivitiesDash.002	ACTVITIES	Select Search- Profile on dropdown menu	The results should be filtered by Profile activities		FAIL		Some of results do not have the Profile tag
TC.ActivitiesDash.003	ACTVITIES	Select Search- Todo on dropdown menu	The results should be filtered by Todo activities		PASS	TEST	
TC.ActivitiesDash.004	ACTVITIES	Select Search - Categorie All Select Location- All	The results should be filtered including All categories and All locations		PASS		
TC.ActivitiesDash.005	ACTVITIES	Select Search - Categorie All Select Location- 1234567	The results should be filtered by All categories and selected Location		FAIL		The results don't change
TC.ActivitiesDash.006	ACTVITIES	Select Search - Categorie All Select Location- bb	The results should be filtered by All categories and selected Location		FAIL		The results don't change

TEST SCENARIO TS008						
`PROJECT NAME	TALENTSMANAGER			DATE		
TEST SCENARIO DESCRIPTION	Testing the ACTIVITIES component for displaying activities related to employees.			REQUIREMENTS		
MODULE	Activities Overview			TEST TYPE		
GOAL	The activities page should display user activities in detail					
TEST SETUP						
DEPENDENCIES	TS007, TS009					
ID	PROCESS	STEP	EXPECTED RESULT		PASS/FAIL	INPUT DATA	Note:
TC.ActivitiesOver.001	Activities Overview	Click on Open Activities in Dashboard	Clicking on Open Activities should take the user to Personal Page/ Activities tab where all activities are displayed		PASS		
TC.ActivitiesOver.002	Activities Overview	Select Profile from dropdown	The results should be filtered by Profile activities		FAIL		Some of Results do not contain Profie tag
TC.ActivitiesOver.003	Activities Overview	Select Todo from dropdown	The results should be filtered by Todo activities		PASS		
TC.ActivitiesOver.004	Activities Overview	Select Termination from dropwon	The results should be filtered by Terminated actvities		Pass		
TC.ActivitiesOver.005	Activities Overview	Select Company from dropdown	The results should be filtered by Company activities		Pass		

TEST SCENARIO TS009						
`PROJECT NAME	TALENTSMANAGER			DATE		
TEST SCENARIO DESCRIPTION	Testing PERSONAL module of the system for information regarding a users' employees, activities and archives			REQUIREMENTS	20,21,22,23,24,25,26,27,28,29	
MODULE	PERSONAL			TEST TYPE		
GOAL	The Personal Page should display Personal information regarding Employee, Activities and Archive and allow the user to add a new employee					
TEST SETUP						
DEPENDENCIES						
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Note:
TC.Personal.001	PERSONAL	Click on Personal on left sidebar Click Employees Click on search bar Type a word/name	The employee list should be sorted by the typed word/name	PASS		
TC.Personal.002	PERSONAL	CLick on Personal on left sidebar Click Employees Click on Name sorter	The employee list should be sorted from A to Z and vice versa	PASS		
TC.Personal.003	PERSONAL	CLick on Personal on left sidebar Click Employees Click on Location sorter	The employee list should be sorted from minimum locations to max.	PASS		
TC.Personal.004	PERSONAL	Click on Personal > Employees Toggle Active	The employee list should be sorted by active status	PASS		
TC.Personal.005	PERSONAL	Click on Personal > Employees Toggle Inactive	The employee list should be sorted by inactive status	PASS		
TC.Personal.006	PERSONAL	Click on Personal > Employees Toggle both active and Inactive	The list should either be sorted by acitve and inactive status	PASS		
TC.Personal.007	PERSONAL	Click on Personal > Employees Open dropdown Select an option from Location dropdown	The list should display employees related to selected relation	PASS		
TC.Personal.008	PERSONAL	Click on Personal > Employees Open dropdown Select ALL option from Location dropdown Select ALL option from Division dropdown Select ALL option from Department dropdown Select All option from Position	The list should include all Employees	FAIL		MOST OF THE EMPLOYEES ARE NOT DISPLAYED WHEN ALL OPTIONS ARE SELECTED
TC.Personal.009	PERSONAL	Click on Personal > Employees Open dropdown Type a name in search box Select no option from Location dropdown Select no option from Division dropdown Select no option from Department dropdown Select no option from Position	The list should be updated by the name typed	PASS	VALON	
TC.Personal.010	PERSONAL	Click on Personal > Employees Open dropdown Toggle Active Type a name in search box Select ALL option from Location dropdown Select ALL option from Division dropdown Select ALL option from Department dropdown Select ALL option from Position	The list should display all Employees within all Locations, Departments, Divisions and Positions.	PASS	VALON	
TC.Personal.011	PERSONAL	Click on Personal > Employees Click on + (plus) sign Fill the information Click Save	The system should add a new employee with regarding information about employee	Pass	first name:HANA Last Name: RUSTEMI Birthday: 25.09.2021 Nationality: Albanian Country: Macedonia Zip code:1200 Location: contact Test dardan Division: 1 Department: cannot select Position: employee Team Assignment: team member Group: Intern Type of Employment: Temporary Pensum: part time Entry Date:20.09.2021 Working days: tu,thu,fri	
Tc.Personal.012	PERSONAL	Click on Personal>Employees Click on an employee Check if the employee Profile contains: Profile infomraiton	ployee. All their related and registerd information should be displaye	Pass		

TEST SCENARIO TS010					
`PROJECT NAME	TALENTSMANAGER		DATE		
TEST SCENARIO DESCRIPTION	Testing DOCUMENTS module of the system for uploading a document		REQUIREMENTS	30.31.32.33	
MODULE	DOCUMENTS		TEST TYPE		
GOAL	The user should be able to add a document to the system				
TEST SETUP					
DEPENDENCIES					
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA
TC.Documentsl.001	DOCUMENTS	Click on Documents on left sidebar Click on (+) add sign Click on Open File Browser	The user shoukd be able to add a document to the sytem	PASS	
TC.Documents.002	DOCUMENTS	Click on Documents on left sidebar Click on search Type a keyword	The user should be able to search for documents	PASS	

TEST SCENARIO TS011						
`PROJECT NAME	TALENTSMANAGER			DATE		
TEST SCENARIO DESCRIPTION	Testing CALENDAR module of the system			REQUIREMENTS	14,15,16,17,18	
MODULE	CALENDAR			TEST TYPE		
GOAL	The Calendar should be displaying all events related to user					
TEST SETUP						
DEPENDENCIES						
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Not
TC.Calendar.001	CALENADAR	Click on Calendar in left nav bar Check if events are displayed	The system should display events on the calendar	PASS		
TC.Calendar.002	CALENADAR	Click on Calendar in left nav bar Check if it has Day view Week view	The system should display calendar on: days, weeks. Months and years.	PASS		

TEST SCENARIO TS012					
`PROJECT NAME	TALENTSMANAGER		DATE		
TEST SCENARIO DESCRIPTION	Testing the interface elements		REQUIREMENTS	7,8,	
MODULE	DASHBOARD		TEST TYPE		
GOAL					
TEST SETUP					
DEPENDENCIES					
ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA
TC.Elements.001	DASHBOARD	Check if the dashboard page contains: Todos Contacts Events Your Team Activities	The page should be divided in 5 components: Todos,Contacts, Events, Your team, Activities	PASS	
TC.Elements.002	DASHBOARD	Check if a left Nav. Bar exists with elements: Personal Documents Calendar My Team	The page should contain a nav bar in the left side with elements: Personal, Documents,Calendar My Team,	PASS	
Tc.Elements.003	DASHBOARD	Check if settings button is in left nav bar	User settings should be located in left nav bar	PASS	
TC.Elements.004	DASHBOARD	Check if a search bar exists in dashboard	The search bar should be located on the main page in dashbaord	PASS	
TC.Elements.005	DASHBOARD	Check if the page auto fits to resized screen	The page elements should fit into resized page	FAIL	
TC.Elements.006	SETTINGS	Click on settings Click on my settings Close the left nav bar	The left nav bar should be minimized when menu icon Is clicked	FAIL	SETTINGS BAR REMAINS OPEN

TEST SCENARIO TS012

PROJECT NAME	TALENTSMANAGER	DATE	
TEST SCENARIO DESCRIPTION	Testing the functionality of Settings	REQUIREMENTS	35,36,37,38,39
MODULE	SETTINGS	TEST TYPE	
GOAL			
TEST SETUP			
DEPENDENCIES			

ID	PROCESS	STEP	EXPECTED RESULT	PASS/FAIL	INPUT DATA	Note:
TC.Settings.001	SETTINGS	Click on left nav bar Click on settings Click on My Settings Check if personal, company and administration settings are visible.	The user should be able to see their personal settings, company settings and administration settings.	PASS		
TC.Settings.002	SETTINGS	Click on left nav bar Click on settings Click on My Settings Click on a Flag (GER or EN)	The system should let the user change the system language either to English or German	PASS		
TC.Settings.003	SETTINGS	Click on settings Click on My Settings Type the current password Type a new password Type confirm new password Click change password	The user should be able to change their password	PASS		
Tc.Settings.004	SETTINGS	Click on settings Click on Calendar Settings Click on Office 365 Calendar	The user should be able to synchronize their office 365 calendar with talentsmanager	FAIL		
Tc.Settings.005	SETTINGS	Click on settings Click on Location Click on search Type keyword	The search bar should display results based on keyword	PASS		
Tc.Settings.006	SETTINGS	Click on settings Click on Location Click on expand Select a type Select a headquarter Select a city Select a language Select a country	The results displayed below should be filtered based on selected items	PASS	headquarter: New location test City:tetovo language:english Country:Macedonia	
Tc.Settings.007	SETTINGS	Click on settings Click on (+) add Location Type name of location Select Date of establishment Select type of company Select Company language Select Status click Save	The user should be able to add a new location by adding related infomration and clicking save	PASS		
Tc.Settings.008	SETTINGS	Click on Settings Click on Organisation Click on Division tab Select a divison from dropdown	The Division list displayed should be sorted by the selected location	PASS		
Tc.Settings.009	SETTINGS	Click on Settings Click on Organisation Click on Division tab Click on (+) to add a new division Type a name of divison Select an available location Click Save	The user should be able to add a new division by filling related informaiton and clicking save.	PASS		
Tc.Settings.010	SETTINGS	Click on Settings Click on Organisation Click on Departments tab Select a location from dropdown	The Departments list displayed should be sorted by the selected location	PASS		
Tc.Settings.011	SETTINGS	Click on Settings Click on Organisation Click on Departments tab Click on (+) to add a new division Type a name of department Select an available location Click Save	The user should be able to add a new department by filling related informaiton and clicking save.	PASS		
Tc.Settings.012	SETTINGS	Click on Settings Click on Roles/Permissions Click on Basis tab Search for a Name Select a role from dropdown	The displayed list of roles should be filtered based on typed name and selected role	PASS	test	
Tc.Settings.013	SETTINGS	Click on Settings Click on Roles/Permissions Click on Basis tab Click on (+) to add a new Role Type a name of Role Select a role from dropdown Select a group Select an available location Click on Save	The user should be able to add a new role to the system	PASS		
Tc.Settings.014	SETTINGS	Click on Settings Click on Document Categories Type a category name to search	The results displayed below should be filtered based on typed keyword	PASS		
Tc.Settings.015	SETTINGS	Click on Document Categories Click on (+) sign Type a name of category choose a type	The user should be able to add a new category to the documents	PASS		

Appendix C – Test Scripts

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestLoginTS001():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCLogin001(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()

  def test_tCLogin002(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("hanarustemi@hotmail.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.CSS_SELECTOR, ".align-items-center").click()

  def test_tCLogin003(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1440, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("12345678")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".align-items-center")
    assert len(elements) > 0

  def test_tCLogin004(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.NAME, "password").send_keys(Keys.ENTER)

  def test_tCLogin005(self):
    self.driver.get("https://dev.talentsmanager.ch/login")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()

  def test_tCLogin006(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("kinurabi@yahoo.com")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()

  def test_tCLogin007(self):
    self.driver.get("https://dev.talentsmanager.ch/login?returnUrl=%2Fdashboard")
    self.driver.set_window_size(1158, 794)
    self.driver.find_element(By.LINK_TEXT, "Forgot password?").click()
    self.driver.find_element(By.NAME, "email").click()
    self.driver.find_element(By.NAME, "email").send_keys("hanarustemi@hotmail.com")
    self.driver.find_element(By.CSS_SELECTOR, "span").click()
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestLogoutTS002():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
      return set(wh_now).difference(set(wh_then)).pop()

  def test_tCLogout001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__user").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__user").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__user--arrow-down").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__user__logout__link--icon").click()

  def test_tCLogout002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.close()
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestDashboardTS003():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCDashboard002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search").click()
    self.driver.find_element(By.ID, "searchInput").click()
    self.driver.find_element(By.ID, "searchInput").send_keys("Valon ")

  def test_tCDashboard003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(786, 794)
    element = self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".ml-4 > app-svgicon")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".header__autocomplete__item:nth-child(2) .header__autocomplete__name").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".header__autocomplete__item:nth-child(2) .header__autocomplete__profile")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()

  def test_tCDashboard004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search > use").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--close")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.ID, "searchInput").send_keys("Valon")
    self.driver.find_element(By.CSS_SELECTOR, ".header__autocomplete__item:nth-child(1) .header__autocomplete__name > p").click()

  def test_tCDashboard005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search").click()
    self.driver.find_element(By.ID, "searchInput").send_keys("Valon")
    self.driver.find_element(By.CSS_SELECTOR, "#searchInput").send_keys(Keys.ENTER)
    self.driver.close()

  def test_tCDashboard006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item--search app-svgicon").click()
    self.driver.find_element(By.ID, "searchInput").send_keys(" hana")
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".header__nav__icon--close > use")
    assert len(elements) > 0
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--close").click()

  def test_tCDashboard001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 750)
    WebDriverWait(self.driver, 4).until(expected_conditions.visibility_of_element_located((By.CSS_SELECTOR, ".header__item--search")))
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".header__item--search")
    assert len(elements) > 0
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestDashboardTS004():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCNotifications001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1328, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".todo-created-notification__description:nth-child(3)").click()
    self.driver.close()
    assert self.driver.title == "Todos"

  def test_tCNotifications002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon > use").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon > use")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".cdk-overlay-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".header__page-name > .ng-star-inserted")
    assert len(elements) > 0

  def test_tCNotifications003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    self.driver.close()

  def test_tCNotifications020(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()

  def test_tCNotifications005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs use").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.close()

  def test_tCNotifications006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs use").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
    actions = ActionChains(self.driver)
```

```
  def test_tCNotifications004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs > app-
svgicon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".cdk-overlay-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon > use").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-
inserted:nth-child(2)")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-
child(2)").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.close()

  def test_tCNotifications007(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--
xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-
inserted:nth-child(2)")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-
child(2)").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()

  def test_tCNotifications008(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--
xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon > use").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon > use")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-
inserted:nth-child(2)")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-
child(2)").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.close()

  def test_tCNotifications009(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--
xs .header__nav__icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-
inserted:nth-child(2)")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-
child(2)").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.close()

  def test_tCNotifications010(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 757)
    element = self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--
```

```
def test_tCNotifications011(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 757)
  self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs > app-svgicon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()

def test_tCNotifications012(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 757)
  self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs use").click()
  self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
  self.driver.execute_script("window.scrollTo(0,0)")

def test_tCNotifications013(self):

def test_tCNotifications014(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 747)
  self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
  element = self.driver.find_element(By.ID, "#foa0isl8q9vp")
  if element.is_selected() != True: element.click()

def test_tCNotifications015(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 747)
  self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
  element = self.driver.find_element(By.ID, "#foa0isl8q9vp")
  if element.is_selected() != True: element.click()

def test_tCNotifications016(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 747)
  self.driver.find_element(By.CSS_SELECTOR, ".header__item:nth-child(1) > .d-none--xs .header__nav__icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".header-notify__header__icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-focus-indicator > .ng-star-inserted:nth-child(2)").click()
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTODOTS005():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCTodoDash001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(969, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".d-block:nth-child(1) > .to-do-list .to-do-list__title--box > .to-do-list__title").click()
    self.driver.execute_script("window.scrollTo(1,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.find_element(By.CSS_SELECTOR, ".heightFull .tmCard__header").click()

  def test_tCTodoDash002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".header__item__menu--todo .header-notify__header")
    assert len(elements) > 0

  def test_tCTodoDash003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".date:nth-child(1) .date").click()

  def test_tCTodoDash004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".d-block:nth-child(1) > .to-do-list-header .dropdown__icon").click()
    self.driver.find_element(By.XPATH, "//div[@id=\'mat-menu-panel-19\']/div/button[2]").click()
    assert self.driver.find_element(By.CSS_SELECTOR, ".container-fluid > .row > .col-xs-12").text == "Edit Todo"

  def test_tCTodoDash005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".d-block:nth-child(4) .to-do-list__action use").click()
    self.driver.find_element(By.CSS_SELECTOR, ".dropdown__item--danger").click()
    self.driver.find_element(By.CSS_SELECTOR, ".color--white > span").click()

  def test_tcTodoDash006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".plus_plus-icon > use").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".btn__no_border--red")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".cke_wysiwyg_div").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".cke_wysiwyg_div")
    self.driver.execute_script("if(arguments[0].contentEditable === 'true') {arguments[0].innerText = 'hanatest'}", element)
    self.driver.find_element(By.CSS_SELECTOR, ".datepicker--enabled .form__icon--calendar").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(1) > .mat-calendar-body-cell:nth-child(2) > .mat-calendar-body-cell-content").click()
    self.driver.find_element(By.CSS_SELECTOR, ".icon_uploadLabel").click()
    self.driver.find_element(By.CSS_SELECTOR, ".position-relative:nth-child(1) > .position-relative > .icon-container").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(3) > .mat-calendar-body-cell:nth-child(5) > .mat-calendar-body-cell-content").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn--primary").click()
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tcTodoDash007(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, "app-button > .d-flex").click()
    element = self.driver.find_element(By.CSS_SELECTOR, "app-button > .d-flex")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".icon--svg--todo-landing-icon").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".sub_header__text")
    assert len(elements) > 0
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTodosOverviewTS006():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tcTodoOver001(self):
    self.driver.get("https://dev.talentsmanager.ch/todo/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".btn > span")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".icon--svg--todo-landing-icon").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".sub_header__container")
    assert len(elements) > 0

  def test_tCTodoOver009(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(905, 747)
    self.driver.find_element(By.XPATH, "//main[@id=\'content\']/app-todo/app-sub-header/div/div[2]/div/app-scroll-pagination-wrapper/div/div/app-card/div/div/div/div/label/app-svgicon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-9:nth-child(1) .form__icon--calendar:nth-child(1)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".fill--primary--light--3").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-9:nth-child(2) app-svgicon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(3) > .mat-calendar-body-cell:nth-child(6) > .mat-calendar-body-cell-content").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(4) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option--selected").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".cdk-overlay-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(6) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(5) .form__select").click()

  def test_tCTodoOver008(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".accordion__label__box--bottom").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-9:nth-child(1) .icon-container").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(1) > .mat-calendar-body-cell:nth-child(2) > .mat-calendar-body-cell-content").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-9:nth-child(2) .icon-container").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(3) > .mat-calendar-body-cell:nth-child(5) > .mat-calendar-body-cell-content").click()

  def test_tCTodoOver007(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".form__radio-group:nth-child(2) .form__radio-button").click()
    WebDriverWait(self.driver, 30).until(expected_conditions.presence_of_element_located((By.CSS_SELECTOR, ".d-block:nth-child(2) > .to-do-list")))
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".d-block:nth-child(2) .to-do-list__checkbox:nth-child(1) span:nth-child(1)")
    assert len(elements) > 0

  def test_tCTodoOver006(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".form__radio-group:nth-child(3) .form__radio-button").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tCTodoOver005(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".form__radio-group:nth-child(2) .form__radio-button").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".d-block:nth-child(1) .to-do-list__checkbox:nth-child(1) span:nth-child(1)")
    assert len(elements) > 0

  def test_tCTodoOver004(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".mt-0 > .form__checkbox-group svg").click()
    WebDriverWait(self.driver, 30).until(expected_conditions.visibility_of_element_located((By.CSS_SELECTOR, ".d-block:nth-child(1) .col-xs-12 .col-xs-12:nth-child(2)")))
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".d-block:nth-child(1) .col-xs-12 .col-xs-12:nth-child(2)")
    assert len(elements) > 0

  def test_tCTodoOver003(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.NAME, "todoSearch").click()
    self.driver.find_element(By.NAME, "todoSearch").send_keys("TEST")
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".mt-5 .pl-7")
    assert len(elements) > 0

  def test_tCTodoOver002(self):
    self.driver.get("https://dev.talentsmanager.ch/todo")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.NAME, "todoSearch").click()
    self.driver.find_element(By.NAME, "todoSearch").send_keys("test")
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".mt-5 .pl-7")
    assert len(elements) > 0
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestDashboardTS007():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCActivitiesDash002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1440, 791)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    WebDriverWait(self.driver, 30).until(expected_conditions.presence_of_element_located((By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header")))
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    assert self.driver.find_element(By.LINK_TEXT, "Todo").text == "Profile"

  def test_tCActivitiesDash003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1440, 791)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(4)").click()
    assert self.driver.find_element(By.LINK_TEXT, "Todo").text == "Todo"

  def test_tCActivitiesDash004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select__selected > span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(1) > .form__select__option__label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tCActivitiesDash005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    assert self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(7) > .row").text == "Location"

  def test_tCActivitiesDash006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-sm-6:nth-child(2) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    assert self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(7) > .row").text == "Location"

  def test_tcActivitiesDash001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.execute_script("window.scrollTo(0,0)")
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestActivitiesOverviewTS008():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCActivitiesOver001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .tmCard__header").click()
    self.driver.find_element(By.LINK_TEXT, "Open Activities").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    assert self.driver.find_element(By.CSS_SELECTOR, ".header__page-name > .ng-star-inserted").text == "Personal"

  def test_tCActivitiesOver002(self):
    self.driver.get("https://dev.talentsmanager.ch/personal/activities")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, "#multipleSelectf1qj7kmopooq").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(2) .form__checkbox-label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tCActivitiesOver003(self):
    self.driver.get("https://dev.talentsmanager.ch/personal/activities")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.ID, "multipleSelectf9buxbflshw").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(4) .form__checkbox-label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()

  def test_tCActivitiesOver004(self):
    self.driver.get("https://dev.talentsmanager.ch/personal/activities")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.ID, "multipleSelectfo5j0mbc3qwe").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(8) .form__checkbox-label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()

  def test_tCActivitiesOver005(self):
    self.driver.get("https://dev.talentsmanager.ch/personal/activities")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.ID, "multipleSelectfcrr3q28bd1").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(7) .form__checkbox-label").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestPersonalTS009():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCPersonal001(self):
    self.driver.get("https://dev.talentsmanager.ch//dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__menu").click()
    self.driver.find_element(By.XPATH, "//div[@id=\'app\']/app-side-nav/div/div[3]/div/nav/ul/li[2]/a/span[2]").click()
    element = self.driver.find_element(By.XPATH, "//div[@id=\'app\']/app-side-nav/div/div[3]/div/nav/ul/li[2]/a/span[2]")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(1) > .sub__nav__link").click()
    self.driver.find_element(By.NAME, "search").click()
    self.driver.find_element(By.NAME, "search").send_keys("Valon")

  def test_tCPersonal002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1440, 799)
    element = self.driver.find_element(By.LINK_TEXT, "Documents")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(2) > .sidenav__link--mobile-hide > .sidenav__link__text").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item__active > .sidenav__link__text")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".pl-7 .table__header__icon").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.execute_script("window.scrollTo(0,0)")

  def test_tCPersonal004(self):
    self.driver.get("https://dev.talentsmanager.ch//dashboard")
    self.driver.set_window_size(1200, 747)
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(2) > .sidenav__link--mobile-hide")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__menu--icon").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(2) > .sidenav__link--mobile-hide").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item > .sidenav__item__active")
    actions = ActionChains(self.driver)
```

```
def test_tCPersonal011(self):
  self.driver.get("https://dev.talentsmanager.ch//dashboard")
  self.driver.set_window_size(1200, 747)
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__menu--icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(2) > .sidenav__link--mobile-hide > .sidenav__link__text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12 > a")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-lg-6:nth-child(1) .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(4) > .form__select__option__label > span").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-lg-6:nth-child(2) .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3) > .form__select__option__label > span").click()
  self.driver.find_element(By.NAME, "preName").click()
  self.driver.find_element(By.NAME, "preName").send_keys("Hana")
  self.driver.find_element(By.NAME, "name").click()
  self.driver.find_element(By.NAME, "name").send_keys("Rustemi")
  self.driver.find_element(By.CSS_SELECTOR, ".ng-dirty .form__icon--calendar").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".fill--primary--light--3")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).click_and_hold().perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".fill--primary--light--3")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".fill--primary--light--3")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).release().perform()
  self.driver.find_element(By.CSS_SELECTOR, ".fill--primary--light--3").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-arrow")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-arrow").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-previous-button")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-previous-button").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(1) > .mat-calendar-body-cell:nth-child(1) > .mat-calendar-body-cell-content").click()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(3) > .mat-calendar-body-cell:nth-child(3) > .mat-calendar-body-cell-content").click()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(4) > .mat-calendar-body-cell:nth-child(5) > .mat-calendar-body-cell-content").click()
  self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(1) > .col-md-4:nth-child(1) .form__select:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select--single").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) .col-md-12:nth-child(1) .form__select:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) .col-md-12:nth-child(1) .form__select:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) .col-md-12:nth-child(1) .form__select:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option--to-select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-md-2:nth-child(3) .position-relative app-svgicon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-previous-button")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-previous-button").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(2) > .mat-calendar-body-cell:nth-child(4) > .mat-calendar-body-cell-content").click()
  self.driver.find_element(By.CSS_SELECTOR, ".pb-6 > .row:nth-child(3)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".design-width-button > span").click()
  WebDriverWait(self.driver, 30).until(expected_conditions.visibility_of_element_located((By.CSS_SELECTOR, ".emp-profile__data")))
```

```
def test_tCPersonal008(self):
  self.driver.get("https://dev.talentsmanager.ch//dashboard")
  self.driver.set_window_size(1200, 747)
  self.driver.find_element(By.LINK_TEXT, "Personal").click()
  element = self.driver.find_element(By.LINK_TEXT, "Personal")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(1) > .sub__nav__link").click()
  self.driver.find_element(By.CSS_SELECTOR, ".accordion__label__box--bottom").click()
  self.driver.find_element(By.CSS_SELECTOR, "#multipleSelectf8j1b1q5rx5q").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) span").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectf8jfnushgo95").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) svg").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectfpkebba57ops").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectfabj4mlvlz7e").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()

def test_tCPersonal009(self):
  self.driver.get("https://dev.talentsmanager.ch//dashboard")
  self.driver.set_window_size(1034, 747)
  self.driver.find_element(By.LINK_TEXT, "Personal").click()
  element = self.driver.find_element(By.LINK_TEXT, "Personal")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.NAME, "search").click()
  self.driver.find_element(By.NAME, "search").send_keys("Valon")

def test_tCPersonal010(self):
  self.driver.get("https://dev.talentsmanager.ch//dashboard")
  self.driver.set_window_size(1080, 747)
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(2) > .sidenav__link--mobile-hide").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item > .sidenav__item__active")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".accordion__label__box--bottom").click()
  self.driver.find_element(By.CSS_SELECTOR, ".ml-3 svg").click()
  self.driver.find_element(By.ID, "multipleSelectfzj6gjrktb9").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectfwqnbmaqxcol").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectfxilxqavbb1").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.ID, "multipleSelectfsfryn5xm1j").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__options--open > .form__group__select__option:nth-child(1) span").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.NAME, "search").click()
  self.driver.find_element(By.NAME, "search").send_keys("Valon")
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTS010():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
      return set(wh_now).difference(set(wh_then)).pop()

  def test_tCDocuments001(self):
    self.driver.get("https://dev.talentsmanager.ch//dashboard")
    self.driver.set_window_size(1200, 747)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(3) > .sidenav__link--mobile-hide").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item > .sidenav__item__active")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12 .plus_plus-icon > use").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()
    self.driver.find_element(By.XPATH, "//*[@id=\"documents\"]/app-card[1]/div/div/div/div/div/div[2]/div/div/file-upload/div/document-upload/div/div[1]/app-button/button").send_keys("Users/hanajonuzi/Desktop/
test.docx")
    self.driver.find_element(By.XPATH, "//div[@id=\'edit-document\']/div/div[2]/div[2]/app-form-field/div/app-form-select/div/div/label/span/span").click()
    self.driver.find_element(By.XPATH, "//div[@id=\'edit-document\']/div/div[2]/div[2]/app-form-field/div/app-form-select/div/div/div/label[2]").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select:nth-child(2) > .default-option").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()

  def test_tCDocuments002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.LINK_TEXT, "Documents").click()
    element = self.driver.find_element(By.LINK_TEXT, "Documents")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.NAME, "searchTerm").click()
    self.driver.find_element(By.NAME, "searchTerm").send_keys("test")
    self.vars["window_handles"] = self.driver.window_handles
    self.driver.find_element(By.LINK_TEXT, "test date exp employee").click()
    self.vars["win4710"] = self.wait_for_window(2000)
    self.vars["root"] = self.driver.current_window_handle
    self.driver.switch_to.window(self.vars["win4710"])
    self.driver.close()
    self.driver.switch_to.window(self.vars["root"])
    self.vars["window_handles"] = self.driver.window_handles
    self.driver.find_element(By.LINK_TEXT, "test 2").click()
    self.vars["win1344"] = self.wait_for_window(2000)
    self.driver.switch_to.window(self.vars["win1344"])
    self.driver.close()
    self.driver.switch_to.window(self.vars["root"])
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTS011():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCCalendar001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.LINK_TEXT, "Calendar").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".first")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".calendar_event:nth-child(1) span").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()

  def test_tCCalendar002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__item:nth-child(4) > .sidenav__link > .sidenav__link__text").click()
    self.driver.execute_script("window.scrollTo(0,0)")
    self.driver.find_element(By.CSS_SELECTOR, ".first").click()
    self.driver.find_element(By.CSS_SELECTOR, ".calendar__btn-view:nth-child(2)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".last").click()
    self.driver.find_element(By.CSS_SELECTOR, ".align-content-center").click()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-next-button").click()
    self.driver.find_element(By.CSS_SELECTOR, ".mat-calendar-next-button").click()
    self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(2) > .mat-calendar-body-cell:nth-child(6) > .mat-calendar-body-cell-content").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()
    self.driver.close()
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTS012():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCElements001(self):
    assert self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(1) > .col-xs-12:nth-child(1) .tmCard__header span").text == "Todos"
    assert self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(1) > .col-xs-12:nth-child(2) .tmCard span").text == "Contacts"
    assert self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) > .col-xs-12:nth-child(1) span").text == "Events"
    assert self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) > .col-xs-12:nth-child(2) .tmCard__header span").text == "Your Team"
    assert self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(3) .h1 > span").text == "Activities"

  def test_tCElements002(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    assert self.driver.find_element(By.LINK_TEXT, "Dashboard").text == "Dashboard"
    assert self.driver.find_element(By.LINK_TEXT, "Personal").text == "Personal"
    assert self.driver.find_element(By.LINK_TEXT, "Documents").text == "Documents"
    assert self.driver.find_element(By.LINK_TEXT, "Calendar").text == "Calendar"
    assert self.driver.find_element(By.CSS_SELECTOR, ".d-block > .sidenav__link").text == "MyTM"

  def test_tCElements003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link:nth-child(1) > .d-flex").click()
    assert self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__header--title").text == "Settings"

  def test_tCElements004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".header__nav__icon--search").click()

  def test_tCElements005(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1024, 728)
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".row:nth-child(1) > .col-xs-12:nth-child(1) .tmCard__header span")
    assert len(elements) > 0

  def test_tCElements006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1161, 728)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link:nth-child(1) > .d-flex").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(1) .sidenav__navigation--settings__nav__item:nth-child(1) .settings--link-text").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__logo-box").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__menu--icon > use").click()
    elements = self.driver.find_elements(By.CSS_SELECTOR, ".sidenav__navigation--settings")
    assert len(elements) == 0
```

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestTS013():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_tCSettings001(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
    assert self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(1) > .sidenav__navigation--settings__nav-section--title").text == "Personal settings"
    assert self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) > .sidenav__navigation--settings__nav-section--title").text == "Company settings"
    assert self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(3) > .sidenav__navigation--settings__nav-section--title").text == "Administration"

  def test_tCSettings003(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(1) .sidenav__navigation--settings__nav__item:nth-child(1) .settings--link-text").click()
    self.driver.find_element(By.CSS_SELECTOR, ".circle-language:nth-child(2) use").click()
    self.driver.find_element(By.NAME, "current_password").click()
    self.driver.find_element(By.NAME, "current_password").send_keys("123456")
    self.driver.find_element(By.NAME, "password").click()
    self.driver.find_element(By.NAME, "password").send_keys("123456")
    self.driver.find_element(By.NAME, "confirm_password").click()
    self.driver.find_element(By.NAME, "confirm_password").send_keys("123456")
    self.driver.find_element(By.CSS_SELECTOR, ".btn > span").click()

  def test_tCSettings004(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link:nth-child(1) > .d-flex").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(1) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text").click()
    self.driver.find_element(By.CSS_SELECTOR, ".calendar__item:nth-child(5) span").click()
    self.driver.close()

  def test_tCSettings006(self):
    self.driver.get("https://dev.talentsmanager.ch/dashboard")
    self.driver.set_window_size(1200, 753)
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link--plus")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
    self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(1) .settings--link-text").click()
    element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
    self.driver.find_element(By.CSS_SELECTOR, ".accordion__arrow-icon__small > use").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(1) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".specific-height > .form__select__option:nth-child(6)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) .form__group__select:nth-child(2) > .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(4)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(4) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__group__select:nth-child(2) > .default-option").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(5)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) .form__group__select:nth-child(2) > .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) .form__group__select:nth-child(2) > .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) .form__group__select:nth-child(2) > .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(5)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) .form__group__select:nth-child(2) > .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(6)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(4) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(5) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(4)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(5) .form__select").click()
    self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
    self.driver.find_element(By.CSS_SELECTOR, ".d-flex:nth-child(1) > .ng-untouched:nth-child(2) svg").click()
    self.driver.find_element(By.CSS_SELECTOR, ".default-option").click()
    self.driver.find_element(By.CSS_SELECTOR, ".default-option").click()
```

```
def test_tCSettings007(self):
  self.driver.get("https://dev.talentsmanager.ch/settings/location/create")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.NAME, "name_of_company").click()
  self.driver.find_element(By.NAME, "name_of_company").send_keys("hanatesttt")
  self.driver.find_element(By.CSS_SELECTOR, ".icon-container > app-svgicon").click()
  self.driver.find_element(By.CSS_SELECTOR, ".ng-star-inserted:nth-child(3) > .mat-calendar-body-cell:nth-child(5) > .mat-calendar-body-cell-content").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(1) > .row > .col-xs-12:nth-child(1) .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(3)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(4) .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".row:nth-child(2) > .col-xs-12:nth-child(3) .form__select__selected:nth-child(1) > span:nth-child(1)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__options-container--open > .form__select__option:nth-child(2)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()

def test_tCSettings008(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link--mobile:nth-child(1)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.ID, "multipleSelectf871159v1egl").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(2) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.CSS_SELECTOR, ".table__body__row:nth-child(2) > .table__cell .color--primary").click()
  self.driver.find_element(By.ID, "backdrop01zgubp7meep").click()
  self.driver.find_element(By.CSS_SELECTOR, ".table__body__row:nth-child(1) > .table__cell .color--primary").click()

def test_tCSettings009(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  element = self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.ID, "f052y8unbxsg5").click()
  self.driver.find_element(By.ID, "f052y8unbxsg5").send_keys("htest")
  self.driver.find_element(By.ID, "multipleSelectf3ovxekdiw1j").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()
  self.driver.find_element(By.ID, "multipleSelectfg1wx2kot3pm").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()

def test_tCSettings010(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.LINK_TEXT, "Locations")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".sub__nav__item:nth-child(2) > .sub__nav__link").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sub__nav__item:nth-child(2) > .sub__nav__link")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, "#multipleSelectfau7snml802m").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
```

```
def test_tCSettings011(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__link:nth-child(1) > .d-flex").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav-section:nth-child(2) .sidenav__navigation--settings__nav__item:nth-child(2) .settings--link-text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  element = self.driver.find_element(By.LINK_TEXT, "Calendar Settings")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".sub__nav__item:nth-child(2) > .sub__nav__link").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sub__nav__item:nth-child(2) > .sub__nav__link")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon").click()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.ID, "fjy6a8jofzab").click()
  self.driver.find_element(By.ID, "fjy6a8jofzab").send_keys("testdept")
  self.driver.find_element(By.CSS_SELECTOR, "#multipleSelectfn3emad10ah > .form__group__select__options-selected").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()

def test_tCSettings012(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item:nth-child(3) .settings--link-text").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item__link--active > .settings--link-text")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  element = self.driver.find_element(By.CSS_SELECTOR, "body")
  actions = ActionChains(self.driver)
  actions.move_to_element(element, 0, 0).perform()
  self.driver.find_element(By.NAME, "search").click()
  self.driver.find_element(By.NAME, "search").send_keys("test")

def test_tCSettings013(self):
  self.driver.get("https://dev.talentsmanager.ch/dashboard")
  self.driver.set_window_size(1200, 753)
  self.driver.find_element(By.CSS_SELECTOR, ".w-100 .sidenav__link__text").click()
  self.driver.find_element(By.CSS_SELECTOR, ".sidenav__navigation--settings__nav__item:nth-child(3) .settings--link-text").click()
  self.driver.find_element(By.CSS_SELECTOR, "a .plus_plus-icon").click()
  self.driver.find_element(By.NAME, "name_of_role").click()
  self.driver.find_element(By.NAME, "name_of_role").send_keys("testestest")
  self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(2) > .ng-invalid .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__select__option--to-select:nth-child(1)").click()
  self.driver.find_element(By.CSS_SELECTOR, ".col-xs-12:nth-child(1) > .ng-invalid .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select--single > .form__select").click()
  self.driver.find_element(By.CSS_SELECTOR, "#multipleSelectfjmrzumvzmbr > .form__group__select__options-selected").click()
  self.driver.find_element(By.CSS_SELECTOR, ".form__group__select__option:nth-child(1) .form__checkbox-label").click()
  self.driver.find_element(By.CSS_SELECTOR, ".app-backdrop").click()
  element = self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span")
  actions = ActionChains(self.driver)
  actions.move_to_element(element).perform()
  self.driver.find_element(By.CSS_SELECTOR, ".btn--primary > span").click()
  self.driver.find_element(By.ID, "f2adjag9sr56").click()
  self.driver.find_element(By.ID, "f2adjag9sr56").send_keys("testestest")
```