SOFTWARE TEST PLAN: ContractMe

Approvals:

Approved By:	Signature	Date

Document Control

Name	STP ContractMe
Doc. Ref. No.	12345
Document Status	For approval
Date of Issue	30/12/2020

Change History

Doc. Version	Author	Date	Description / Change
1.0	Team	22/10/2020	Start writing the STP document

Distribution List

Name	Role
Benny Zend	Course lecturer
Rotem Reshef Goldshtein	Course instructor

1 Introduction

The purpose of this document is to describe the plan for testing our system.
This part of the project is very important because it will help us find all the problems and bugs in our code, which is essential.
we will check each function, by entering inputs, for each input we will verify if we get the output we expected.
we will verify that the code matches with the SRS, the Prototype, the Use case and the DFD's.

2 Scope

The tests are done on all the functions in the code.
We use our database that will have idle data so we can perform the tests - users database' hiring history database and work history database.
The tests will be divided among the group members, the method of testing will be by users function.
There are no test phases that are done by other teams, the tests are done only by the main tests team.
The deadline to submit the tests is until December 30, 2020.
The tests are done on "Visual Studio".

3 Test Plan Identifier and Document Change Control

Doc. Ref. No : 12345
Version 1.0 - December 30 , 2020

4 References

Document Reference & Version	Document Title / Description
M34 - Version 1.2	software requirements specifications (SRS)
S39 - Version 1.1	Prototype - Simulation of the system
J51 - Version 1.1	Use case - functionality image
B84 - Version 1.1	DFD 0 - Data flow diagram of General view
B85 - Version 1.1	DFD 1.1
B86 - Version 1.1	DFD 1.2
B87 - Version 1.1	DFD 1.4
B88 - Version 1.1	DFD 1.5
B98 - Version 1.1	DFD 1.7
B99 - Version 1.1	DFD 2.2.1
K12 - Version 1.0	Initiation Document
F16 - Version 1.0	cpp code

5 Test Items

Test Item Name	Test Item Version No.
ContractMe.cpp	1.5

5.1 Features to be Tested

Feature	Parent Component / System	Overview	relevance
1. Log-in	ContractMe cpp code	Checks connection with different inputs (valid and invalid) and checks for valid inputs that the connection is successful and for invalid inputs that the connection failed and ask for new inputs.	Relevant to all three users
2. sign-up	ContractMe cpp code	Check if the user name that the user entered is available . If it is taken we will get an error message and ask for a new username. If the username is not taken, the user is asked to enter a password and immediately afterwards it is observed that the user will register in the database and enter in his menu.	Relevant to the employer users
3. Search contractor	ContractMe cpp code	The employer will enter inputs in the search filters (can choose not to enter a figure) and we will expect to get all the contractors who are suitable for the search.	Relevant to the employer users
4. Book contractor	ContractMe cpp code	The test includes the selection of a wanted contractor and it is observed at the end of the test to see the hiring write	Relevant to the employer users

		in the database of work history and also in a database of hiring history.	
5. Statistic analysis	ContractMe cpp code	Check that the printed data matches the information in the system. We get the number of employers in the system, the number of contractors in the system and how many contractors were actually employed.	Relevant to the HR users. Uses two functions that need to be checked.
6. Hiring history	ContractMe cpp code	In this feature we will check that the entire hiring history of the employer is printed as written in the database. Through human resources we will see the hiring history of all existing employers in the system and through a specific employer we will see only a printout of all his hiring.	Relevant to the employer and HR users.
7. Work history	ContractMe cpp code	In this feature we will check that the entire work history of the contractors is printed as written in the database. Through human resources when we select a contractor we will see his workday history and through a specific contractor we will see a printout of all his work days only.	Relevant to the contractor and HR users.
8. Edit workday	ContractMe cpp code	In the test we will enter a date of a work day that we want to change and enter a change of hours. We expected to see at the end of the test a change in the hours of the same work day in the database.	Relevant to the HR users
9. Edit details	ContractMe cpp code	In the test, we will change the contractor's information through his user. You can	Relevant to the contractor users .

		change the area, salary and interests of the contractor (there is a choice of what you want to change, there is no obligation to change everything). At the end of the test we observed a change in the data of this contractor in the database.	
10. Add contractor	ContractMe cpp code	The test is to add a contractor to the database. We will enter the name of the contractor, username, password, area, salary and interests and these should be added to the database. At the end of the test we observed to see in the database the details of the new contractor added.	Relevant to the HR users.
11. Reporting work day	ContractMe cpp code	The test is done to check that reporting workday has indeed entered the database. In the test, we will enter a work day and hours and we expect to see at the end of the test that the work day has been added to the database.	Relevant to the contractor users.
12. Reporting vacation	ContractMe cpp code	The test is done to check that reporting vacation day has indeed entered the database. In the test, we will enter a vacation day and we expect to see at the end of the test that the vacation day is added to the database.	Relevant to the contractor users.
13. Technical support	ContractMe cpp code	The test is made to check that the complaint has been sent to the department that handles complaints.	Relevant to all three users

		During the test, we will enter a complaint and we observe that a message will be printed that the complaint was indeed received by the system.	
14. System logo	ContractMe cpp code	Checking the logo printing and .We will check that the logo is indeed printed when getting into the system. It is expected that when we enter the system , we will get the logo printed.	Relevant to all three users
15. Exit logo	ContractMe cpp code	Checking the exit logo printing and exit from the system. We will check that the logo is indeed printed when leaving the system. It is expected that when we click on exit, we will get the exit logo printed.	Relevant to all three users
16. Full inspection of the system- H.r	ContractMe cpp code	We will perform a full test of the system by using a HR worker, we will check that all the functions related to it actually work. It is expected to receive proper operation of all functions and information that is compatible with the database.	Relevant to the HR users.
17. Full inspection of the system - Employer	ContractMe cpp code	We will perform a full test of the system by using an employer user , we will check that all the functions related to it actually work. It is expected to receive proper operation of all functions and information that is compatible with the database.	Relevant to the employer users.
18.Full inspection of the system- Contractor	ContractMe cpp code	We will perform a full test of the system by using a contractor user , we will check that all the functions related	Relevant to the contractor users.

		to actually work. It is expected to receive proper operation of all functions and information that is compatible with the database.	
19. Integration test	ContractMe cpp code	We will check that the data that runs during the system also changes in the corresponding databases.	Relevant to all the functions.
20. Log- out	ContractMe cpp code	We will check that we can log out of the system and go to the main menu from all the users menu - human resources, contractor and employer. We will log in from each user and make sure to exit and reach the main menu.	Relevant to all the users.
21. Home button	ContractMe cpp code	The test is for all functions. We will check that at the end of each use of a certain function for each of the users by pressing Enter we will return to each user's menu.	Relevant to all the users.

5.2 Features not to be Tested

In the testing process we will not test the data types because we are assuming that the input is of the desired type.
In our project we made assumptions therefore we won't do tests on these parts.
We assume that the HR workers are already written in the user database therefore we won`t check the option that the HR user does not exist.

The system is a stand alone system therefore we don't need to do load tests.

6 Testing Risk Register

Risk ID No.	R1
Summary	Non-compliance with deadlines
Probability of Occurrence	Med

Customer Impact	The customer will not receive the product at the set time. High score.
Trigger	Delayed tests, malfunctions in unexpected situations
Mitigation Action	timely notification of potential problems or shift in the schedule and make daily meetings.
Contingency Action	Add more members to the tests team

Risk ID No.	R2
Summary	Unskilled staff and get wrong results.
Probability of Occurrence	Low
Customer Impact	Can cause unexpected errors when the user uses it . Med score.
Trigger	Hiring a team without proper knowledge and who have not taken a course on the subject.
Mitigation Action	Make the testers a knowledge test before hiring them.
Contingency Action	Teach them, test them more, add skilled new team members to help.

Risk ID No.	R3
Summary	The budget allocated is over
Probability of Occurrence	Med
Customer Impact	Not all tests will be performed and this can cause bugs in the system . Med score.
Trigger	If the tests are done at a slow pace it can cause the budget to end earlier than expected.
Mitigation Action	Set a larger budget for tests in advance, and make sure the test team works at the required pace.
Contingency Action	Increase the budget.

Risk ID No.	R4
Summary	Failure to cover all customer requirements in inspections and the time required

Probability of Occurrence	Low
Customer Impact	Not all tests will be performed and this can cause bugs in the system . Med score.
Trigger	Unexpected delays, an unexamined requirement.
Mitigation Action	Coordinating expectations with the client to cover the scenarios and defining criticality for the scripts.
Contingency Action	Ask for more time and present a new work plan

Risk ID No.	R5
Summary	Unexpected absence of some team members
Probability of Occurrence	Low
Customer Impact	May delay the testing process and submission time. Med score.
Trigger	Illness of team members, call to reserve, etc.
Mitigation Action	Risks like that can usually not be prevent from happening in advance, but it is possible, for example, to hire people who do not make reserves
Contingency Action	We will recruit professional people for emergencies in case few members need to be absent we will have new staff members to fill their place.

7 Test Approach (Strategy)

The tests types that we will do are the functional tests to verify the system activity and we will do that test in the start of the test process, the regression tests (if needed) that will be done when one function won't return the output we expected to get, then after a change of this function we will do regression tests ,and in the end , system tests that check the system fully and integration tests to check that the database are changed in the process like they should.
We plan to test the code in a method that will be divided into several parts.
In the first part - we will examine the functions of the main systems, including the login and exit function, the registration function, the menu adjustment function ect.
In the second part we will examine all the functions related to the contractor's menu.
In the third part we will examine the functions related to the menu of the human resources company
In the last part we will look at the functions related to the employer menu.

After the functional tests we will do the integration tests ,regression tests and the system tests.
The tests we perform include a comprehensive test of how the functions deal with different inputs.
We use predominantly requirements-based manual test scripts, the tests will be done by all members of the group, each one will get several functions and perform the tests on them, so there are some parts in the functionality tests that we will need to 'direct' the test, for example if we test the function of statistic analysis , we assume that the log- in of the HR worker to the system succeeded.
In order to perform the tests, we will uniformly use the template we received in advance (test case).

7.1 Test Tools-NONE

List all the tools required for testing. Such tools may include obvious ones like test management software, a defect management system, GUI automation tools, etc., but there may be less obvious tools required like project management tools, etc. If you have a remote contingent as part of the overall test team then consider any tools you will need in order to effectively communicate with them.

If commercial-off-the-shelf (COTS) tools are not suitable then you may need to develop own tools, harnesses and stubs. This is all work that requires to be costed, estimated and planned so include it here.

7.2 Test Data

For the tests we will use our database and all the information in it, such as log-in details of the HR worker for the first log-in and will get assistance from the database administrator .
The various tests will be performed by the project members maunaly and we will not be needing to reset the database after each test.
In the tests we use three databases that in them we will put idle data for having a basis for starting the tests.
One database will save the details of each user, the second database will save the workday of all the contractors and the last one will save the hiring history of all the employers in the system.

7.3 Test Environment

The tests performed on operating system Windows 10 on visual studio.

8 Personnel

Name	Role	Responsibility
Maayan Nadivi	Test Manager	Responsible for managing the test teams and monitoring the testing process.
David Abenhaim	Software Tester	Responsible for the test cases.
Yarden Hovav	Software Tester	Responsible for the test cases.
Benny Shalom	Software Tester	Responsible for the test cases.
Mikhail Diyachkov	DBA	Responsible for managing the database, take care of the design and construction of a database and set up for routine maintenance.

8.1 Training-NONE

Since you have reviewed the roles and responsibilities of test-related personnel it may have become apparent that there are skills gaps. If this is the case then you will need to decide the best way to fill these gaps. It may be that paired testing and mentoring between senior and junior staff is a suitable solution. However, training may sometimes need to be more formal and this becomes a separate task in its own right that will require planning, costing and incorporation into the project.

9 Management and Metrics

The manager of the testing is Maayan Nadivi and she will be responsible for the adherence to the schedule.
The test team will manage the different versions of software released into the testing phase.
We will do a daily test team meeting mainly to monitor the progress of the testing team .
Each one of the members of the test team accepts part of the functions and does the tests on the features he receives until the deadline .

9.1 Test Estimation and Schedule

Project phase	**Time**	**Deadline**
Functional tests	4 days	27/12/2020
Integration tests	4 days	27/12/2020
regression tests(if necessary)	1 day	28/12/2020
system tests	1 day	29/12/2020

The deadline is December 30, therefore the team members must finish the tests they get until December 28 to make sure that all the tests have been done properly.

9.2 Test Phase Entry and Exit Criteria

9.2.1 Integration Test Phase Entry Criteria and Acceptance Test Phase Entry Criteria

- Getting a proper budget
- Approval of the developer
- Suitable infrastructure and technology to perform the tests
- Code in working order
- Testers have been trained
- Schedule that can be possible

9.2.2 Integration Test Phase Exit Criteria and Acceptance Test Phase Exit Criteria

- 100% of the tests are executed
- *90*% of the tests pass
- Integration Test Report has been approved
- Less than *10* outstanding low severity issues
- Less than *5* outstanding medium severity issues
- 0 outstanding high severity issues

- Number of issues found did not exceed planned number by more than 25%

9.3 Suspension and Resumption Criteria

We have some criteria for stopping the test process and delaying it.
First, if we find critical bugs in the system we must stop the tests and continue only after the bug fix.

In addition, we will stop the tests if the features required by the customer have bugs, change of environment components or technology including different versions and if the client changes his requirements.

10 Test Deliverables

In our project we have some test deliverables.
First, the Test schedules detail the entire schedule of our testing process, how the testing process is conducted and how a test will be monitored.
In the test specification document is indicated which scenario will be tested in our program and how they will be tested.
In the test data document is indicated all the tests data for the specific tests that we will do. The tests data will be valid data to verify that functions produce expected results and unvalid data to check the functions ability to handle unusual, exceptional or unexpected inputs. In the document we describe the three databases that we use to do the tests.
In test execution results we present the tests results and the history of tests case record.
The test reports document describes the summary of all the test activities that we have done and the final test results of a test project. The stakeholders can evaluate the quality of the product being tested and make a decision on software release by this document.
In the lessons to be learned we describe our lessons from the project management and execution so that we can learn from them and get better the next time in our next project .
In the test risk register we indicate all the risks that we have in our tests process, how we plan to avoid them and how we deal with the risk if it happens.

11 Communication Plan

Name	**Role**	**Contact Details**
Yarden Hovav	Software Tester	Email: Hovav4321@gmail.com Office: 08-8888888 Mob: 052-55555555 Fax: 01671 987 654
Maayan Nadivi	Test Manager	Email: Maayan4321@gmail.com

		Office: 08-1111111 Mob: 052-55555575 Fax: 01671 985 654
David Abenhaim	Software Tester	Email: Office: 08-8888888 Mob: 052-55555556 Fax: 01671 987 655
Benny Shalom	Software Tester	Email: Office: 08-8888888 Mob: 052-55555557 Fax: 01671 987 657
Mikhail Diyachkov	DBA	Email: Office: 08-8888888 Mob: 052-55555558 Fax: 01671 987 651

Communication Aspect	**Purpose**
daily Test Team Meeting	Review immediate issues and plan tasks for the week ahead.

12 Glossary

HR - Human resource

DBA - Database Administrator