SOFTWARE TEST PLAN:

SCE CRM

Approvals:

|  |  |  |
| --- | --- | --- |
| Approved By: | Signature | Date |
|  |  |  |
|  |  |  |

Document Control

|  |  |
| --- | --- |
| Name | Software Test Plan |
| Doc. Ref. No. | 1.0 |
| Document Status | For Approval |
| Date of Issue | 23.12.18 |

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Doc. Version | Author | Date | Description / Change |
| 1.0 | Author name | 23.12.18 | Initial Release – Prototype Test Plan |
|  |  |  |  |

Distribution List

|  |  |
| --- | --- |
| Name | Role |
| Shani Levi | practitioner |
| Benny Zan | lecturer |

# Introduction

This document describes the plan for testing the SCE CRM System. This Test Plan document supports the following objectives:

Identify existing project information and the software that should be tested.

List the recommended test requirements (high level).

Recommend and describe the testing strategies to be employed.

Identify the required resources and provide an estimate of the test efforts.

List the deliverable elements of the test activities.

# Scope

This Test Plan describes the integration and system tests that will be conducted on the CRM.

The purpose is to test feasibility and performance of the selected architecture and to test system functionality and features.

The testing process should check in details the login feature, since it is a critically important function

The testing is contained by time limits; all the test should be completed by 06.01.2019, no test should be running afterward.

All the tests should be done on a computer that contains all the databases.

.

# Test Plan Identifier and Document Change Control

This is the first version

1.0

# References

|  |  |
| --- | --- |
| Document Reference & Version | Document Title / Description |
| 1.3 | **DFD**- A data flow diagram is a graphical representation of the "flow" of through an information system,  modelling its process aspects.  https://github.com/Matan28/Software-Basics/blob/master/DFD.pdf |
| 1.0 | **SRS**- A software requirements specification (SRS) is a description of a software system to be developed.  SRS establishes the basis for an agreement between customers on how the software product should function.  https://github.com/Matan28/Software-Basics/blob/master/SRS%20-%20SCR%20%20-%20final.doc |
| 1.2 | **Use Case diagram**- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.  https://github.com/Matan28/Software-Basics/blob/master/Use-Case%20diagram.pdf |
| 2.0 | **Prototype**- release of a product built to test a concept or process or to act as a thing to be replicated or learned from.  https://github.com/Matan28/Software-Basics/blob/master/improved%20prototype%20final.pptx |

# Test Items

Our software "CRM" is the only product to be tested.

The CRM is a ticket based program intended for collage use.

The program enables students and managers to generate events regarding all collage facilities.

|  |  |
| --- | --- |
| Test Item Name | Test Item Version No. |
| CRM | 1.0 |

## Features to be Tested

Log in – we'll want to make sure the log in will authorize only registered users IDs (students & managers) alike, the log in page should redirect all students to the students page and the managers to the managers page.

Profile- the profile page should only show individual students events with no duplications of the events.

Event numbers- the event numbers should never be duplicated while generated. (no two different events may have the same event number)

Events date – making sure each event is signed with the exact date and time.

Add new event – making sure all new events are being saved and duplicated events deleted.

Manager profile – each page on the managers profile will show the relevant events or info related to the page solely. (pending , closed, all events, reports).

Analytics – should be valid and represent the up to date analysis.

|  |  |  |
| --- | --- | --- |
| Feature | Parent Component / System | Overview |
| Log in | CRM | ID & password input  & appropriate redirection |
| Profile | CRM | All profile parameters |
| Add new event | CRM | Check a variety of inputs to check for duplications of events ,misdirected events or unsaved events |
| Manager profile | CRM | Check any possible bugs regarding for instance change of status . |
| Analysis | CRM | Does the reports represent the correct data while changing various event specifications |

## Features not to be Tested

Event details - are not to be tested, the managers will determine the relevance of an event.

However, duplicated events will not be permitted.

Event priority - is not to be tested (also viewed by the assigned manager).

Event status – will be changed only by the manager.

|  |  |
| --- | --- |
| Document Reference & Version | Document Title / Description |
| CRM 1.0 | Document title and brief description |

# Testing Risk Register

Software development is full of risk, and the testing phases are no exception. It is always wise to take an active lead in managing the risks facing us.

Our project is a CRM to SCE College. During the writing of the program we encountered many risks such as:

* We need to consider the risks involved in working with files, such as saving the files in the right places, making sure they contain all the necessary data, making sure the program reads from them correctly. For example, make sure that the program reads the user's password from the file. If the password is not correct, the program should notify you and make sure that the program does not collapse.
* We need to consider risks that related to the ticket, such as tickets will not deleted, stored in the database and updated according to the user's wishes.
* We need to make sure there are no risks associated with event statuses. We will make sure that the statuses are updated accordingly in the data structure and that each event really receives the appropriate status. For example, we make sure that a new status received in the system is really saved in a pending status and not in a different status.
* If the input is incorrect or empty, we need to make sure that the program informs it and then asks the user for new input and checks whether it is correct and does not collapse.

Since we run the system using Visual Studio, which is a third party product we have to test the system only using the Visual Studio, in addition the ability to test the system is limited to the said program. As there is a deadline to finish the system we will not renew the risk register.

In both creating and reviewing risks it is well worthwhile teaming with other testers and developers since they will often be vocal in expressing their own personal key concerns about the project.

# Test Approach (Strategy)

The testing will contain a step by step running the system for several times using different inputs each time in order to check the system.

The testing will be done manually by the members of the team, as a group.

All the functions that were written in the SRS document will be checked thoroughly, as equally important.

## Test Tools-NONE

Not currently relevant.

## Test Data

The tests require using 2 databases, one which has the existing users and their passwords, the other one is the database of the events.

None of the data from testing would be relevant in case the testing was done without the databases.

## Test Environment

The testing will require a running computer with pre-installed Visual Studio application in order to run the system, in addition the proper run of the system requires the database files (the database of users and the database of the events) to be present in the same folder in the Windows as the system that we are testing.

# Personnel

All of the testing personnel for this project are college students, no team member has some additional skills from previous projects, and no member of the team has a previous experience in testing the system, therefore it was decided that the team will do the entirety of the testing together as a group, without additional separation of the group into subdivision.

## Training-NONE

Not relevant, no member of the team is getting an additional training.

# Management and Metrics

The team had a main meeting during class hours for the initial, main system testing. the team will have additional daily meeting during the following weeks during college hours, the tests should be done by 05.01.2019

The testing will be managed by the same team member that has managed the entirety of the project.

As was agreed by the group, Shai Idan was the manager of the testing process.

The testing was done by the Matan Tal and Yarin Edri.

## Test Estimation and Schedule-

All the tests should be finished by 05.01.2019

Not currently relevant.

## Test Phase Entry and Exit Criteria

All the tests are done manually by the testers; the testing will continue till the 4.01.2019 or until the testers will reach a 100% integration of the test completed.

### Integration Test Phase Entry Criteria

The integration test scripts (expected output vs the actual output) and the planning of the tests will be checked by all the members of the team.

The issues that will be found by the test group will be fixed during the testing.

### Integration Test Phase Exit Criteria

the integration test phase will be finished either when 100% of the integration test are executed or by 05.01.2019 the first of the occasions.

### Acceptance Test Phase Entry Criteria

Since all the testing are done manually without additional programs , the acceptance test will be performed in the same time as the integration tests, in

Addition after finishing the integration test, the team will perform one last run of the entire system , to make sure that the system answers to all the requirements of the user.

### Acceptance Test Phase Exit Criteria

The acceptance test phase will be finished either when 100% of the acceptance test passed successfully or by 06.01.2019

## Suspension and Resumption Criteria

Not relevant , since the developing group has been given a deadline, without possible suspensions.

# Test Deliverables

The tests contains a manual checking of the system , also during the testing of the system we create a STP list , the list should contain all the tests which were run , with the relevant(most resent) result for each of the test .the tests should be finished by 05.01.2019

# Communication Plan

The team will communicate during the college hours (the hours in which all the team members are present in the same territory), and if necessary additional communication will be done through email.

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact Details |
| Kesem Even Hen | Team leader | [Kesemevenhen18@gmail.com](mailto:Kesemevenhen18@gmail.com) |
| Noy Nir | UI & UX | [Noynir100@gnail.com](mailto:Noynir100@gnail.com) |
| Elena Alexeenko | Code writers | [elenaalexceinko@gmail.com](mailto:elenaalexceinko@gmail.com) |
| Shai Idan | Coder | [Shaiidan@hotmail.com](mailto:Shaiidan@hotmail.com) |
| Yarin Edri | QA | [Yarin2@edri@gmail.com](mailto:Yarin2@edri@gmail.com) |
| Matan Tal | developer | [Mamash17@gmail.com](mailto:Mamash17@gmail.com) |
| David Ohana | Report editor | [Davidohana07@gmail.com](mailto:Davidohana07@gmail.com) |
| Adam Yahnin | Code writers | [Adamyahnin@gmail.com](mailto:Adamyahnin@gmail.com) |

|  |  |
| --- | --- |
| Communication Aspect | Purpose |
| Initial meeting | To run the system from start to finish.  Once-only. |
| Testers hangout | To check for the second time , to fix the existing errors. |
| Last check  Before the deadline (day or two before 06.01.2019) | To run the system one last time, to check if the system answers to all the initial requirements and it runs properly |

# Glossary

The term "event" - for each request / malfunction / optimization proposal is called an event. Using the event form you can enter the details for the event we are recording.