SOFTWARE TEST PLAN:

Order clothing online

Approvals:

Approved By:	Signature	Date
Tair Shriki	-	05/01/2020
Chaya levin	-	05/01/2020
Ruth Bracha Cohen	-	05/01/2020
Bulma Rosenfeld	-	05/01/2020

Document Control

Name	software test plan
Doc. Ref. No.	STP 1.1
Document Status	Final
Date of Issue	12/01/2020

Change History

Doc. Version	Author	Date	Description / Change
1.0	The whole group	02/01/2020	Beginning of testing
1.1	The whole group	19/01/2020	After fixing problems

Distribution List

Name	Role
benaya Zand	Lecturer
Our group	

1 Introduction

The purpose of this test plan is to test and validate the functionality of our ticketing system and ensure that we build the right system for the customer (according to the srs document) , by the test plan we can in orderly and systematically find all the error and bugs in minimum time and effort to build a better system.
The system stands by itself and is not connected to additional systems

2 Scope

Budget constraints-irrelevant .This software testing program is very important in relation to the whole project. Software testing is performed by team members. We had to do the software testing for a limited period of time.

Throughout the project, testing was done on the software. The software tests were recorded in the "test case" documents.

3 Test Plan Identifier and Document Change Control

In 02/01/2020, the Test Team start with first draft, this is 1.0 version.
In 05/01/2020, the Test Team updating references, this is 1.1 version.
In 12/01/2020, the Test Team expending the STP, this is 1.2 version.

4 References

During the program review, the following documents were reviewed:

- SRS - Adjust the requirements tested between program outputs and requirements
- Checking databases for all revenues and expenses requires access to the correct database (databases: ticket bases, managers bases, employees bases , customers bases)
- Architecture plans: DFD and Use-case
- Functional specifications are explained in DFD document

The test cases were also examined:

- ❏ Sign up function test case
- ❏ Log in function case test
- ❏ Previous ticket function test case
- ❏ Create a new ticket function test case
- ❏ Ticket information function test case
- ❏ Search by index function test case
- ❏ Ticket for today's function test case

Document Reference & Version	**Document Title / Description**
SRS	requirement document
DFD	checking requirement for specific function
Sign up test case	Testing requirements and function performance sign up test case
Login test case	Testing requirements and function performance log in test case
Previous ticket test case	Testing requirements and function performance Previous ticket test case
Create a new ticket test case	Testing requirements and function performance Create a new ticket test case
Ticket information test case	Testing requirements and function performance Ticket information test case
Search by index test case	Testing requirements and function performance Search by index test case

Ticket for today's test case	Testing requirements and function performance Ticket for today's test case

5 Test Items

As part of this test program, we will check the code. List of programs to check:

Test Item Name	Test Item Version No.
Register Test case	1.0
Login Test case	1.0
Previous ticket Test case	1.0
Create new ticket Test case	1.0
Ticket information Test case	1.0
Search by index Test case	1.0
Tickets for today Test case	1.0

5.1 Features to be Tested

there are not any more features that need to be tested.

Feature	Parent Component / System	Overview
Irrelevant	Irrelevant	Irrelevant

5.2 Features not to be Tested

The following features will not be tested directly by us prior to delivery of the final product. The development team does not have the resources (hardware, software, and personnel) to verify these limits.
•Response time (to a limited extent)
•Dataset size
•Hardware availability

6 Testing Risk Register

Risk ID No.	Risk_1
Summary	Not enough memory
Probability of Occurrence	High
Customer Impact	Med
Trigger	Growth of the system
Mitigation Action	Make an a dynamic allocation of the database, and use a bigger storage. Invest in better and wider foundation of storage

Contingency Action	Send an error message, or use an extra database meantime.

Risk ID No.	Risk_2
Summary	Track of History of tickets does not preserve
Probability of Occurrence	High
Customer Impact	Low
Trigger	Lack of thinking forward,lack of knowledge of the developers.
Mitigation Action	Train the developers to think of all the options. Use extra database to save all the information that required.
Contingency Action	Suspend the program and release a new version containing all relevant data

Risk ID No.	Risk_3
Summary	Breakdown of specification
Probability of Occurrence	High
Customer Impact	Med
Trigger	During the initial phases of integration and coding, requirements might conflict. Moreover, developers may find that even the specification is unclear or incomplete.
Mitigation Action	Deeper investment in writing SRS document before coding steps
Contingency Action	Writing the SRS document again in such a way that it would not allow the meaning to occur in the code

Risk ID No.	Risk_4
Summary	Low knowledge of the developers
Probability of Occurrence	Low
Customer Impact	High
Trigger	Lack of resources where team members can collaborate and share knowledge.
Mitigation Action	Train the developers to maximize their abilities and collaborate with other teams .
Contingency Action	Explore about the topic, use others people knowledge, collaborate with other teams .

Risk ID No.	Risk_5
Summary	Productivity issues

Probability of Occurrence	High
Customer Impact	High
Trigger	On projects involving long timelines, developers tend to take things easy to begin with. As a result, sometimes, they lose significant time to complete the project. Set a realistic schedule, and stick to it.
Mitigation Action	From the beginning of program the developers should have plan a schedule.
Contingency Action	Try to divide the work into a larger number of employees so that the coding task is eventually done properly. Secondly, it would be desirable and important to create a list of urgent tasks in the code task and act on it.

Risk ID No.	Risk_6
Summary	Programmatic Risks
Probability of Occurrence	High
Customer Impact	High
Trigger	changes in government policy, the obsolescence of software, changing customer product strategy and priority or other risks that cannot be controlled or estimated.
Mitigation Action	It is vital that development firms focus on strategic planning to mitigate such risks.
Contingency Action	Freeze the current version and release a new one that will support customer requirements after further focus on the plan and early and comprehensive planning.

Risk ID No.	Risk_7

Summary	Working across multiple sites, off-cities team members, remote-working
Probability of Occurrence	Low
Customer Impact	Low
Trigger	Due to the distance between team members, meetings between team members are rarely held.
Mitigation Action	At the beginning of the work the team members should schedule meetings which all team members agree on. Fortheremore, it is recommended to divide the work between team members so that they will be able to work independently and separately.
Contingency Action	Make non-face-to-face conversations by daily bases. Document the work in an organized manner.

Risk ID No.	Risk_8
Summary	Developer absenteeism during the project
Probability of Occurrence	Low
Customer Impact	Med
Trigger	Not get delivered on time
Mitigation Action	Make a division of labor in a way that leaves the possibility of absence. The division of work will allow the other developers to take alternate roles in the event of absence.
Contingency Action	Delegating roles. Placing reinforcement staff

7 Test Approach (Strategy)

Unit testing and component testing will be performed on the components as they are developed.

Integrations tests will be performed by both the component testers as well as the system testers.As the integration checking begins to include high level functionality, the tests being run will utilize significantly more manual testing.

7.1 Test Tools-NONE

7.2 Test Data

All tests will be performed by team members. These tests do not use helper functions such as "unit test" and the like.

The tests are performed on the whole system by steps and thus encompass the entire system.

7.3 Test Environment

The development team will perform tests by creating a separate environment, the final code will be changed after all the tests cases were examined.
The purpose of the test cases is to fix the bugs in order to bring the final product to be compatible with the STP requirements.
During the test cases , any bug that was found will be documented and the documentation will be repaired at the end of the test case.
Once all the runs and all bugs are fixed, the development / testing environment closes.

8 Personnel

Name	Role	Responsibility
Bluna Rosenfeld	Engineer, Test team	Creation and executing test cases.
Chaya Levin	Engineer,Test team	Creation and executing test cases. Division and manage test cases between the team members.Prepare the Software Test Plan
Ruth Bracha Cohen	Engineer,Test Manager,Test team	Creation and executing test cases. Manage the Testing and provide technical support to the Testing team.Make Review of Software Test Plan and test cases.
Tair Shriki	DBA,Engineer,Test team	Creation and executing test cases.Evaluating exit criteria.

8.1 Training-NONE

9 Management and Metrics

The test will be conducted by the test team. All the responsibility is on the test team, which is also the software development team. Every few days, a meeting was held between team members. There was a brainstorming session on how to continue the program.

9.1 Test Estimation and Schedule

Deadline is until December 12th, this file is meant to be finished within 3 weeks.

Number of week	Work to do
#1	● dividing the roles ● division of work ● plan the time ● start the STP file
#2	● Develop test cases ● Execute test
#3	● Finish the STP file ● Fix the Test cases

9.2 Test Phase Entry and Exit Criteria

As part of building the program, each member in the group is required to write and test the functions assigned to him.
An examination will be considered 'successful' if and only if it stands with the requirements previously written and planned in the SRS document.After each member of the team has corrected and checked the functions that he is responsible on , the program driver changes so that each member of the team will eventually run the tests with the same updated software version.

9.2.1 Integration Test Phase Entry Criteria

When the program was distributed among the team members, it was decided that on 20.12.2019 the writing of the program would end, by 12.01.2020 the program review phase would be completed in a detailed manner and divided into test cases.

9.2.2 Integration Test Phase Exit Criteria

As part of the final product release, the development team is required to perform tests as follows:
Each member of the team will perform a manual check on his part of the work.
If a discrepancy between the program output and the expected output is found , the bug will be fixed and the driver will be re-checked.
Also, extreme cases that could lead to the program collapse will be examined.
After testing the integration tests, the software tested will be available for the next test phase. After testing, there should be a minimum of low / medium / high hardware issues. After the integration tests, the

number of problems found did not exceed the number of problems planned in the first place.

9.2.3 Acceptance Test Phase Entry Criteria

No external testing tools were used, all tests were done manually by the team members.

9.2.4 Acceptance Test Phase Exit Criteria

Irrelevant

9.3 Suspension and Resumption Criteria

when the system reaches a case where it cannot function such as: Memory allocation problem - which causes runtime error, the program will stop and along with the tests. Only after the critical problem has been corrected will the tests be conducted.
In the case of a "small" problem, the tests that do not belong to the problem created will be run as usual, but the tests that depend on the same case will be pending until resolved.

10 Test Deliverables

The following activities must be completed:

- Test plan prepared.
- Functional specifications written and delivered to the testing team
- Environment should be ready for testing
- Perform the tests.

11 Communication Plan

Name	Role	Contact Details
Bnaya sand	Test Team Leader	Email: bnayasa@ac.sce.ac.il
Our group	Test Team	

Communication Aspect	Purpose
weekly Test Team Meeting.	Review immediate issues and plan tasks for week ahead.
weekly Test Team Meeting with lecturer.	Reception hours for further guidance.

12 Glossary

Term	Meaning
SRS	Software Requirements Specificity
DFD	Data Flow Diagram
STP	Software Test Plan