



TEST PLAN

for

Travel in good health system

Version 1.0 approved

Prepared by

Abhishek Gandhi | 19CS10031

Shashvat Gupta | 19CS30042

Sajal Chhamunya | 19CS10051

Instructors :

Prof. Sourangshu Bhattacharya

Prof. Abir Das

Omprakash Chakraborty

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

18 March 2021

Table of Content

Table of Content	2
1 TEST PLAN IDENTIFIER	3
2 REFERENCES	3
3 INTRODUCTION	3
4 SOFTWARE RISK ISSUES	3
5 FEATURES TO BE TESTED	4
Feature - 1: Database Management System	4
Feature - 2: Placing Orders	4
Feature - 3: User Interface	4
7 FEATURES NOT TO BE TESTED	4
8 APPROACH	5
8.1 Testing Levels	5
8.1.1 Unit Testing	5
8.1.2 System/Integration Testing	5
8.2 Test Tools	5
9 ITEM PASS/FAIL CRITERIA	5
10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS	6
11 TEST DELIVERABLES	6
12 ENVIRONMENTAL NEEDS	6
13 GLOSSARY	6

1 TEST PLAN IDENTIFIER

TGHM_MTP01.1

2 REFERENCES

- Software Requirement Specification for TGHM, team demonic debuggers.

3 INTRODUCTION

This is the Master Test Plan for the Travel in Good Health Management System project by team Demonic Debuggers. This plan's primary focus will be addressing the non-GUI testing component of the software.

The project will have three levels of testing, Unit, System/Integration and Acceptance. The details for each level are addressed in the approach section and will be further defined in the level specific plans.

4 SOFTWARE RISK ISSUES

Since the software will be finally hosted as a website, hence system incompatibility won't be an issue. Other possibilities for issues which are not within the control of the TGHM application and need to be checked for are:

- A. Database security and access must be defined and verified.
- B. Limit on the maximum numbers of simultaneous logins will be based on the hosting servers capabilities, and needs to be updated and checked regularly according to the peak traffic on the website.
- C. Government regulations and rules would be needed to cross check with.

5 FEATURES TO BE TESTED

Feature - 1: Database Management System

Testing with Python Automate:

- *Fetch_And_Display*: To interact with the database
- *Edit_And_Store*: To edit train routes, restaurants and their menus
- *Show_Nearby_Stations*: to show nearby restaurants from the Customer by putting its nearest station as the newly added station.
- *Get_Statistics*: to get statistics of all orders/stations/restaurants/Agents.
- *Remove*: to remove all stations, restaurants, menus, etc.

Feature - 2: Placing Orders

Testing with Python Automate:

- *Issue_Order*: to create an order which a customer has placed
- *Restaurant_Accepts*: to be called when the restaurant accepts the order
- *Restaurant_Declines*: to be called when the restaurant declines the order due to any issue.

Feature - 3: User Interface

Testing with Python Automate:

- *Wrong_UserName_OR_Password*: to be called when Username or Password don't match.
- *Right_UserName_AND_Password*: to be called when the Username and Password match.
- *Forgot_Password*: for when, a user forgets his/ her Password.

7 FEATURES NOT TO BE TESTED

In the software all features implemented by the developer team have been tested. Third party libraries and their functions which have been used are not tested.

8 APPROACH

8.1 Testing Levels

There are many different testing levels which help to check the behavior and performance for software testing. Here, the testing of the TGHM project has been divided into two parts, namely Unit testing and System/Integration testing.

8.1.1 Unit Testing

All the individual units or components of the software will be tested in the Unit Testing so as to validate that each unit of the software code performs as expected. This way errors in any particular class can be pinpointed easily and corresponding measures can then be taken to rectify those errors.

8.1.2 System/Integration Testing

System/Integration testing is carried out to verify the behavior of the complete system. It is done after all the critical defects in any individual class/function have been corrected and is therefore will be carried out after Unit testing. System testing is performed to evaluate the end-to-end system specifications and verify the interactions between the modules of the software system. Here, we will be testing all the features of the software together and verifying the correctness of the entire code as a single unit, unlike the unit testing where we have to isolate a section of code and test it. This will be our final test to verify that the product meets the specifications mentioned in the SRS.

8.2 Test Tools

Test Automation and code coverage analysis libraries will be used for testing the application. Specifically, *Unittest framework* provided by Python is used. This framework supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework.

9 ITEM PASS/FAIL CRITERIA

The test process will be completed once the initial set of unit testing and system/Integration testing has been completed.

If there is a System failure or a wrong output situation in any of the testing phases, then changes will be suggested and implemented and after rectifying those errors, testing will be restarted.

10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

Any system fatal error will require the suspension of testing and will require the rectification of these errors from the software development team.

Wrong outputs on corner cases are not immediate concerns for suspension of the testing period and can be rectified at the end of the testing.

Resumption of testing, if testing had been suspended would be after the software development team works on rectifying the errors found in the testing phase. The testing will have to start from the beginning again, to make sure that no scope for error is left.

11 TEST DELIVERABLES

- Master Testing Plan
- Test Suite
- Testing file
- Testing Compliance report

12 ENVIRONMENTAL NEEDS

No special hardware/software will be needed for testing of the application other than the requirements for running the software.

As the software will be hosted as a Website, a fairly stable Internet Connection would be required to run the software for a client. The software, if required, for testing and development purposes can also be run on a local server.

The testing data will be provided in a .txt file as cases, provided by Test Suite. Later on, a testing file will also be provided, which upon running can be used to generate the compliance testing report.

13 GLOSSARY

Abbreviations used:

TGHM - Travel in Good Health Management system