



TEST PLAN

Social Serving Food Delivery System

Prepared By

Tireless Whoopers

IIT Kharagpur

21.03.2022

Table of Contents

1. TEST PLAN IDENTIFIER.....	2
2. REFERENCES	2
3. INTRODUCTION	2
4. TEST ITEMS (FUNCTIONS).....	2
5. SOFTWARE RISK ISSUES	3
6. FEATURES TO BE TESTED	3
7. FEATURES NOT TO BE TESTED.....	5
8. APPROACH (STRATEGY).....	5
9. ITEM PASS/FAIL CRITERIA.....	6
10. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS.....	7
11. TEST DELIVERABLES	7
12. REMAINING TEST TASKS	7
13. ENVIRONMENTAL NEEDS	7
14. PLANNING RISKS AND CONTIGENCIES	7
15. GLOSSARY	8

1. TEST PLAN IDENTIFIER

SSFDS_1.0.1

2. REFERENCES

SRS document for Social Serving Food Delivery System

3. INTRODUCTION

The purpose of this Software Test Plan (STP) is to develop a test plan to access all functionality of the Social Serving Food Delivery System V 1.0.1. It includes all the possible tests that need to be done to ensure the correctness and reliability of the software to be implemented. The software should pass all the tests before it can be delivered.

4. TEST ITEMS (FUNCTIONS)

The following is the list of the items to be tested:

1. Restaurant

- Edit Profile
- Add Food Item
- Increase Food Amount
- Check Pending Order List
- Process Order
- Log Out

2. NGO

- Edit Profile
- Check Restaurant Food Items
- Select Item
- Change Amount
- Place Order
- Paying Option
- Log Out

3. Customer

- Edit Profile
- Check Restaurant Food Items
- Select Item
- Change Amount
- Place Order
- Donate To Restaurant
- Paying Option
- Log Out

5. SOFTWARE RISK ISSUES

1. Log In authentication must be verified and password should not be leaked
2. Information of all types of users (Restaurant authority, NGO, Customer) in Database should be kept safe. Any other external authority should not collect those. Information from database should not get deleted at any way. The system log in function may not work then.
3. Updation of npm (Node Package Module) can cause problems like the node modules would then need to be re installed. However this problem mainly applies to maintenance phase. As there would be no problem in the deployed version as long as the hosting server recognizes the npm version used and deploys accordingly.

6. FEATURES TO BE TESTED

1. Successful adding of new user through sign up option (Manually / Google)
2. Successful deletion of user (Restaurant, NGO, Customer) through deleteUser option
3. Successful logging in of user (Restaurant) through Log In option.

4. Clicking add item button to get a space to enter details of food item.
5. Clicking add button to finally add the item into the server such that it can be visible from customer's account.
6. Clicking Show Order to get the order list from different customer
7. Successfully process an order by the restaurant authority
8. After entering into profile, selecting edit profile option to update the initial details of the restaurant
9. Successfully log out from the account by clicking log out button
10. Successful logging in of NGO/Customer through Log In option
11. Showing nearest restaurants on the account homepage
12. Clicking over restaurants to get the food lists.
13. Clicking add food option to add food into cart
14. Clicking a plus/minus (+/-) symbol button to increase/decrease the food amount
15. Clicking place order option to get payment option window
16. Successfully paying through UPI (If Online option is selected)
17. After entering into profile, selecting edit profile option to update the initial details of the restaurant.
18. For customer, opening of a NGO listed window to select donation to a NGO.
19. Successfully log out from the account by clicking log out button.
20. Redirection to homepage after logging out for all the user.

7. FEATURES NOT TO BE TESTED

1. Efficiency and reliability of hosting server.
2. Network access and security
3. Functions of the Database like select and querying elements.

8. APPROACH (STRATEGY)

The following testing levels are present:

- Unit Level
- System Level

Unit Testing:

Each function is to be tested as mentioned in Section 4

Application Testing:

- Restaurant
 - To check a valid previous restaurant can log in or not
 - Check if it redirects to account home page or not
 - Requested orders from customers and NGOs are showing or not
 - If the edit profile is working properly or not
- NGO
 - To check a valid previous NGO can log in or not
 - Check if it redirects to account home page or not
 - Check if nearest restaurants list is showing or not
 - Check if we click on restaurant, it is directing into food item list of that restaurant or not

- Check add item to cart and increment item is working or not
- To check order is being placed or not
- For online payment option, checking of successful transaction
- Customer
 - To check a valid previous Customer can log in or not
 - Check if it redirects to account home page or not
 - Check if nearest restaurants list is showing or not
 - Check if we click on restaurant, it is directing into food item list of that restaurant or not
 - Check add item to cart and increment item is working or not
 - To check order is being placed or not
 - For online payment option, checking of successful transaction
 - To check if NGO list is showing or not while clicking option donation to NGO

The testing tools used are:

1. JavaScript / Node.JS: All tests will be done in JavaScript itself
2. Manual Testing: For frontend/ User Interface manual visual testing to be done

9. ITEM PASS/FAIL CRITERIA

The completion criteria for the plan are:

1. At Unit Level:

- All Unit Tests Passes
2. At the System Level:
 - All Application Tests Passed

10. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

1. If user does not provide the correct details or the details does not match with none from database, then the system suspends its test. Like, in this case user will not be able to log into account. System will redirect user to login page.

11. TEST DELIVERABLES

As a part of this test plan the following are delivered:

1. Unit test plan
2. Application test plan
3. Test Suite (With test cases and Outputs)

12. REMAINING TEST TASKS

None as such

13. ENVIRONMENTAL NEEDS

- Node installed
- Test data to be provided from database for backend testing and as manual inputs for User interface testing.

14. PLANNING RISKS AND CONTIGENCIES

- As we are hosting the server for free, it may possible if the number of users is more, there might be a shortage of space for data. Then, we have to buy some plans to get extra space.

- If any feature is not passing the tests even after significant effort, it could be dropped if it not a major feature.

15. GLOSSARY

UPI: Unified Payment Interface

UI: User Interface