



**School of Engineering And Applied Science, Ahmedabad
University**

Software Engineering

Under the guidance of
Prof. Doctor Khushru

Online Food Ordering System

List of team members

- | | |
|-------------------|-----------|
| 1. Saha Vishal | AU1741004 |
| 2. Dhruv Shah | AU1741008 |
| 3. Jay Patel | AU1741018 |
| 4. Mann Bilimora | AU1741021 |
| 5. Manav Shah | AU1741042 |
| 6. Jaydeep Modi | AU1741070 |
| 7. Shashwat Mehta | AU1741100 |

INDEX

Sr. No.	Title	Page No.
1.	Introduction	3
1.1.	Objectives	3
1.2.	Testing Strategies	3
1.3.	Scope	7
1.4.	Definition and Acronyms	7
2.	Features to be tested	7
3.	Features not to be tested	9
4.	Approach	9
5.	Pass / Fail criteria	10
6.	Testing Process	11
7.	Environmental requirements	12
8.	References	12

1. Introduction

Hungry Kya? - A simple and convenient way for customers to purchase food online, without having to go to the restaurant. This system is enabled by the internet – it is the internet that connects the restaurant or the food company on one hand, and the customer on other hand. Therefore, as per this system, the customer visits the restaurant's website, browses through the various food items, combos and cuisines available there and goes ahead and selects and purchases the items he or she needs. These items will then be delivered to the customer at his or her doorstep at the time they choose by a delivery person. Payments for such online orders can be made through debit cards, credit cards, cash or card on delivery, or even through digital wallets. This system for online food delivery is completely safe, secure and is a very popular method that is revolutionizing the way in which the food industry operates.

1.1. Objectives

- Testing the features.
- Testing the interfaces and integrations.
- Preparing an updated list which needs to happen post testing.

1.2. Testing Strategy

- All modules are evaluated as units according to a feature oriented engineering strategy.
- Testers include developers , team peers and college peers.
- Within it is specified the participants for different test designs.
- It also determines the concepts and methodology of the tests.
- We focused on the tests that impact the behavior of the system.
- Used test data that is close to that of production.

1.2.1. Unit Testing

1.2.1.1. Definition :-

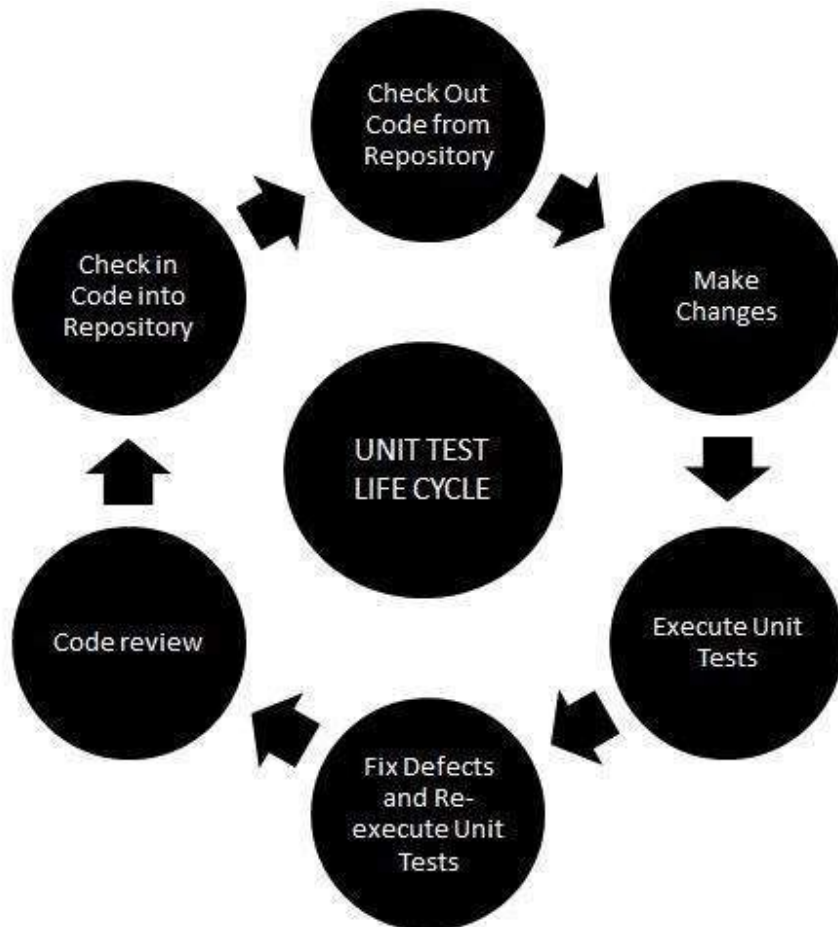
Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Unit testing consists of checking and testing the source code for various functions. To minimize redundancy the source code is checked. We strive to provide a minimum degree of comprehensiveness so that in need the user can register and access data

1.2.1.2. Participation :-

Unit Testing is conducted by the developers itself and team peers.

1.2.1.3. Methodology :-

Unit Test will be conducted by the developers of the features itself. The method will be conducted by Black Box Testing method. The test scripts will be executed by the developer of the feature or in some cases by project peers. We have decided on important features which impact the software working. is maintained according to simple state transitions, i.e, from login to exit. Testing occurs by both, writing test cases in code or fuzzy developer tests by firing up GUI and working ahead with the test case by inputting the data. Both are advantageous and help in debugging.



1. Photo reference

1.2.2. Component and System Integration Testing.

1.2.2.1. Definition :-

Integration Testing is the second level of testing performed after Unit Testing and before System Testing. Testing to expose crucial defects between the interactions of interfaces and internal integrated modules. This method is called component integration testing. Component Integration testing is a type of testing in which we integrate small small modules of code together and test them. These tests validate the performance and help in faster development when we validate integrated modules, we are able to integrate the complete system with minimum error.

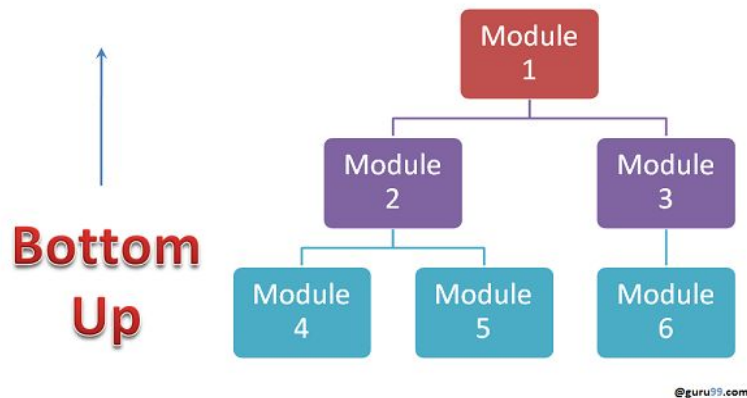
1.2.2.2. Participation :-

Integration testing is conducted by the test team and team peers depending upon modules which are being tested

1.2.2.3. Methodology :-

- **Bottom UP Approach**

Black Box testing is implemented as a second level of testing. We have selected to implement a **Bottom-Up** approach to implement integration testing. Once the modules are unit tested, they are integrated one by one, till all the modules are integrated, to check the combinational behavior, and validate whether the requirements are implemented correctly or not. Test team need to know that successful unit tests may not result in successful integration tests.



2. Photo reference

1.2.3. System Testing

1.2.3.1. Definition :-

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications

1.2.3.2. Participants :-

Testing was conducted independently of the group, by class peers not involved in the development phase.

1.2.3.3. Methodology :-

System testing generally validates that the system meets the requirements mentioned during the development and planning phase. Certain requirements include authorized updates, modularity, scalability, common platform, etc. These tests are conducted as a black box test. The independent tester will be unaware of the working of internal functions. The tester may attempt to find errors in categories such as incorrect or missing functions, interface errors, initialization or terminal errors

1.2.4. Acceptance Testing

1.2.4.1. Definition :-

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

1.2.4.2. Participants :-

Alpha testing is conducted by team peers and beta testing is conducted by class peers.

1.2.4.3. Methodology :-

The acceptance test is conducted as the last stage of testing before deployment. Alpha Testing is conducted as part of uncovering any missed errors or missing functions by peers in the organization itself and is generally a black-box testing method. Beta Testing is conducted outside the organization. They generally include customer testing and end-user testing. These users will test to validate that nontechnical requirements are

fulfilled and interfaces are integrated efficiently.

1.3. Scope

We are going to check all the models, views and controllers along with the connection with the database. Also all the features which are supposed to be checked and along with that the methodologies which are used to test the software are done. Along with that the security testing and the features which are not tested are also written with their uses and reasons.

1.4. Definitions and Acronyms

- **Project** – activities that will lead to the production of the Air Traffic Control System.
- **Client** – the person or organization for which this Air Traffic Control System is being built.
- **User** – the person or persons who will actually interact with the Air Traffic Control System.
- **Use case** – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.
- **Scenario** – one path through a use case.
- **Actor** – user or another software system that receives value from a use case.
- **Developer** – the person or organization developing the system, also sometimes called the supplier.
- **Stakeholder** – anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.

2. Features to be Tested

2.1. User Login

- User/customer enters login details i.e. Username/email and password
- Selects category of user according to login details
- This will guide the user to his/her personal dashboard.

2.2. Restaurant Selection -

- Customers should be given list of restaurants and order food from his desired restaurant
- Menus of restaurants should be available for customers to choose their food.

2.3. Add items

For restaurant id handler

- Restaurant id handlers enter their ID, password to log in.
- Restaurant id handlers can add new dishes in the menu by clicking on add new dish and entering valid dish details such as price, quantity etc.
- Changes should be reflected in the database after the update.

For customer

- Customer enters their ID, password to log in.
- Customers can add new dishes in the cart by clicking on add to cart and entering valid quantities.
- Final cart value should be displayed once the customer adds all the dishes he/she wants to order.

2.4. Delete items

For restaurant id handler

- Restaurant id handlers enter their ID, password to log in.
- Restaurant id handlers can remove existing dishes from the menu by clicking on remove this dish.
- Changes should be reflected in the database after the update.

For customer

- Customer enters their ID, password to log in.
- Customers can remove dishes from the cart by clicking on remove this dish from the cart.
- Final cart value should be displayed once the customer adds/removes all the dishes he/she wants to order.

2.5. Update Items

For restaurant id handler

- Restaurant id handlers can update the price of existing dishes.
- Once they update it, changes must be reflected in the database.

2.6. Cancel order

For customer

- Customer requests to cancel the order after the order is placed.

2.7. Cart total

- After the customer finalizes the food, the estimated bill should be directed to the customer for his/her food.
- Also respective prices of each item should be displayed according to the ordered quantity.

2.8. Payment Request

- After the customer finalizes the cart, he can pay for food in both ways - online payment and COD.
- Proper redirection to associated payment options

2.9. User Feedback

- Users are requested to give their opinion on food served by restaurants, services provided by delivery boy and any other feedback.

3. Features Not to be Tested

3.1. Menu

- Almost all restaurants provide a food menu along with decided prices so no need to test.

4. Approach

The scale to which the software may be handled and the intolerance of certain errors which may call for uncalled mishaps. As we have selected feature driven development as our guiding method, we have defined certain important modules which

are to be developed in iterations. Such modules can be tested separately for errors and performance before integrating such modules. Our approach consisted of the modules first tested by the team member who designed the same and thereafter by another team member and then modified the features according to feedback. These tests named unit tests consisted of testing code and debugging certain syntactical and logical errors. Specific features such as Login and Add New food to cart might not work efficiently when the database will scale to realistic numbers if not optimized. Certain features, being modulated, still work its product with integration, such as food order. Such features are to be tested for system design for efficient and valid system integration.

The testing team consisted of already participating team members and peer teams for certain specific tests. Unit and Integration testing were fulfilled by dividing the team in developer and non-developer for every feature and thorough tests were held by both teams as mentioned above. Certain non-technical reviews were recorded for the user interface and graphical interface. Graphics and aesthetics not important for a software in the field of work, the team focussed more on an errorless and banal system to be developed.

5. Pass / Fail Criteria

- If a certain user follows the guidelines of the system and does not make a mistake while executing one's own functionalities, then it can be depicted as a PASS criterion and the opposite of previously mentioned details results in a FAIL criterion.

5.1. Suspension Criteria

- If the user (either customer or restaurant admin) enters a invalid username or password the activity of logging into the account gets suspended with a pop up message reflecting that the user has entered invalid combination of username and password.
- When the user (customer) selects food from one restaurant and tries to add a new food item from a different restaurant then he gets notified that food from only one restaurant can be delivered at a time with activity suspension.
- When the user (customer) submits invalid payment details he gets a message that the payment details were invalid later the activity gets suspended.

5.2. Resumption Criteria

- After suspension from the login state, the system recreates the login state asking for correct login credentials.

- After selecting food from a particular restaurant, he can clear cart and move to that restaurant and buy food or stick to this one.
- Write correct payment details

5.3. Approval Criteria

- The user (Either customer or restaurant admin) has to enter the correct username and password to be able to access their respective functionalities.
- Restaurant admin has to match exact details of the food in order to edit or delete a particular food item from the list.
- While updating the status of a food available to either YES or NO, the restaurant admin has to select the food item that he/she intends to update.
- The customer has to provide correct payment details in order to check out with the selected items in the cart.

6. Testing Process

6.1. Test Deliverables

- The deliverables from the test processes are records from the project plan and minutes from the meetings. Deliverables can be defined from the test case log, in which the status of each test case changes to 'yes' or 'no' according to the results. The actual outcome column may also be viewed as deliverables for the test case documents.

6.2. Testing Tasks

- Duplicate classes are generated before unit testing each and every function of the project, so that they can be tested on another computer. A class is duplicated and passed into another computer where it is independently evaluated. That makes the testing process much simpler and easier to operate.

6.3. Responsibilities

- The roles were distributed evenly between the team. The people who were involved in software design were not interrupted to check the software's features. Testing was performed by team members as well as the people outside of the team. These outsiders checked the features they weren't designing so they could provide a clear analysis of the features reviewed. The documentation was distributed uniformly between the team members, as well as the coding task.

7. Environmental Requirements

7.1. Hardware

- No special specifications for Hardware. Software built on personal computers.

7.2. Software

- MySQL Workbench.
- Visual Code Studio
- Wampserver

7.3. Security

- The testing is done in one system, and the backup is taken in another system to prevent any malfunctions. The testing is also performed on another system to evaluate various other risks that may be encountered.

7.4. Tools

- MySQL Workbench: This software is used to store the data and understand the various queries which are written to check the situations. The tables and their connections are also checked.
- Visual Code Studio - This tool was used to develop javascript, php environments. Use of live servers was very helpful and autosave extension also keep us safe
- Wampserver - For running php servers and also helpful in updated mysql files

7.5. Risks and Assumptions

- The risk-based assessment is focused on checking a list of tasks objectives that may have issues with failure in them. We carry out checks on those factors. Once the concerns continue to evaporate, we look forward to the introduction of new concerns. Different risks, such as duplicating the data, retrieving null data, modifying the data , providing unauthorized functionality to a person with lesser authority.

8. References

- 8.1. <http://softwaretestingfundamentals.com/unit-testing/>
- 8.2. <https://www.softwaretestinghelp.com/black-box-testing/>

- 8.3. https://www.nws.noaa.gov/oh/hrl/developers_docs/General_Software_Standards.pdf
- 8.4. <https://hackr.io/blog/types-of-software-testing>