**Deliverable-3**

3.

Functionality from phase 2 was implemented in phase 1, which is Dashboard. Home page, Sign In and Sign out features were implemented. The sign up feature and the sign in, sign out functionality were not implemented for user in this iteration (were only implemented for admin). In the next iteration, the left over functionalities will be implemented.

Due to functional dependencies between features, we were a bit confused on which part to implement first. This was the challenge the team faced.

a. Requirements:

1. Login page:

* Admin can login into the website with their credentials.
* To login into the website the user need a username and password.
* Login page then checks if the credentials given are right or wrong.
* If the credentials are invalid, it will show the error.

1. Category:

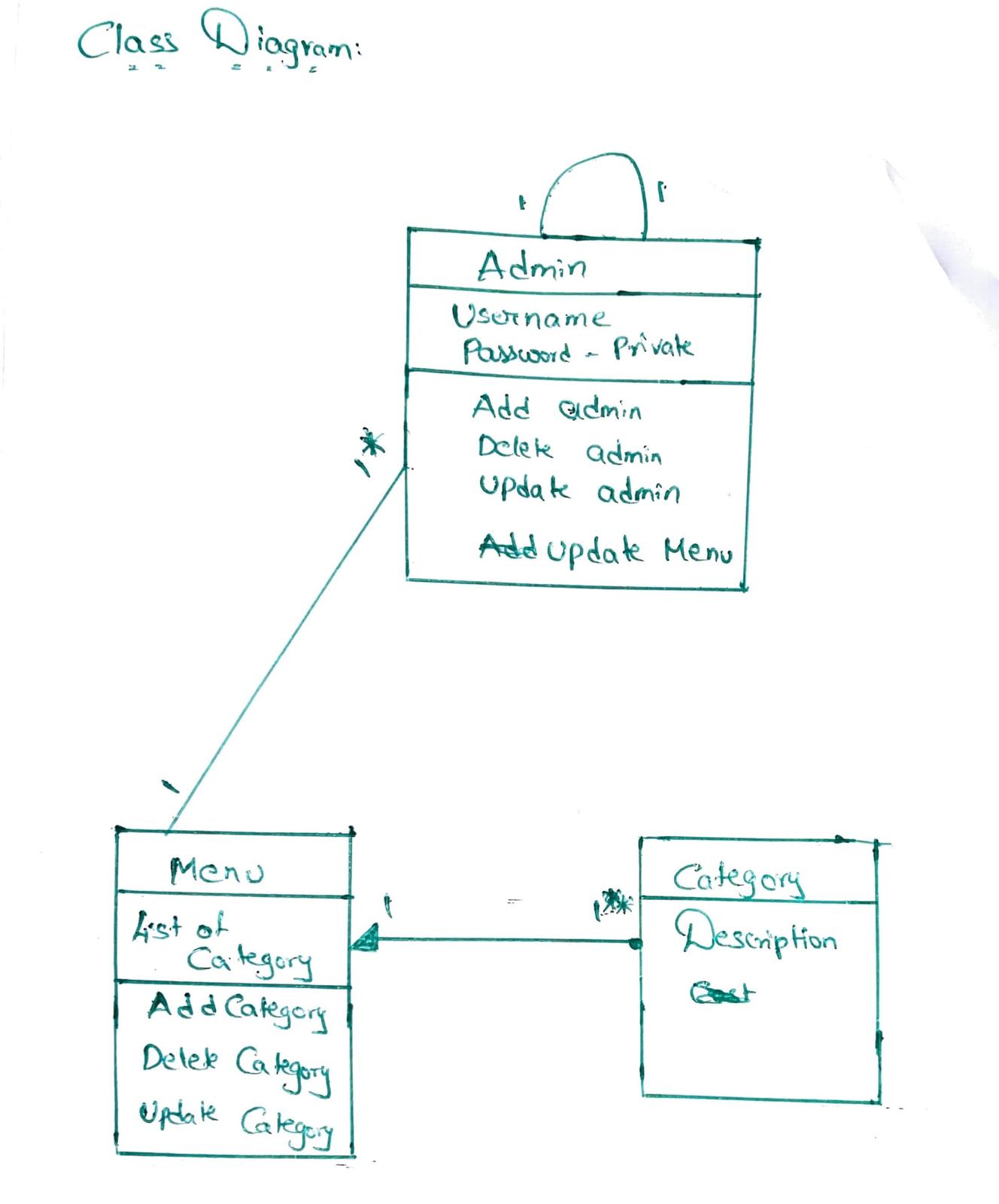
* The category page contains different food options like bowl, drinks, addons.
* The admin can add, delete or update the category.
* The category page contains the items with respect to the category they belong to.

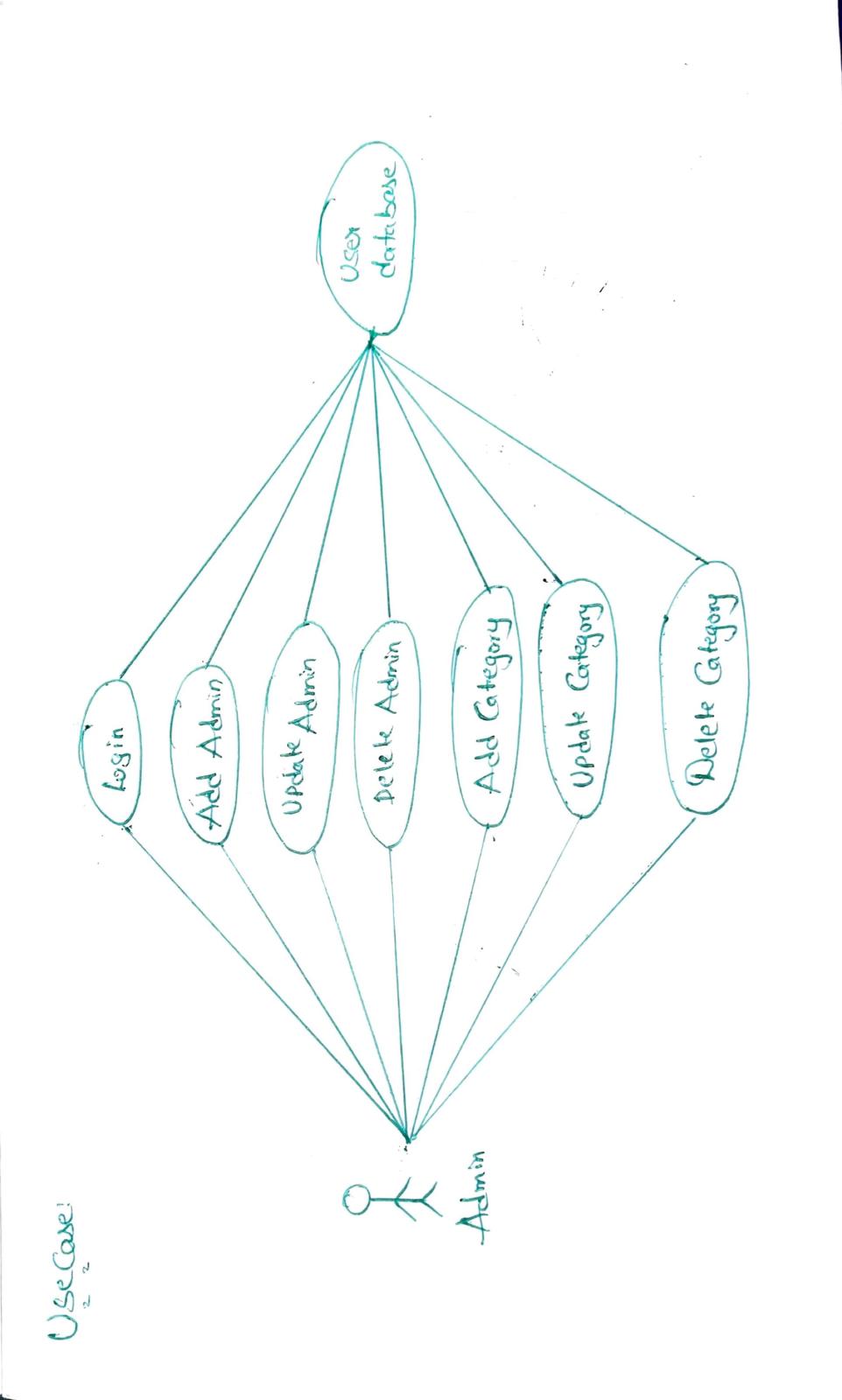
1. Home page:

* The homepage contains a dashboard which contains, tiles the represents categories, foods, total orders, revenue generated.
* The categories tiles show the number of categories present in the website.
* The food tile, show the total number of food items present in the menu as of date.
* Total orders tile shows the orders received today.
* Revenue Generated tile shows the total revenue generated today.

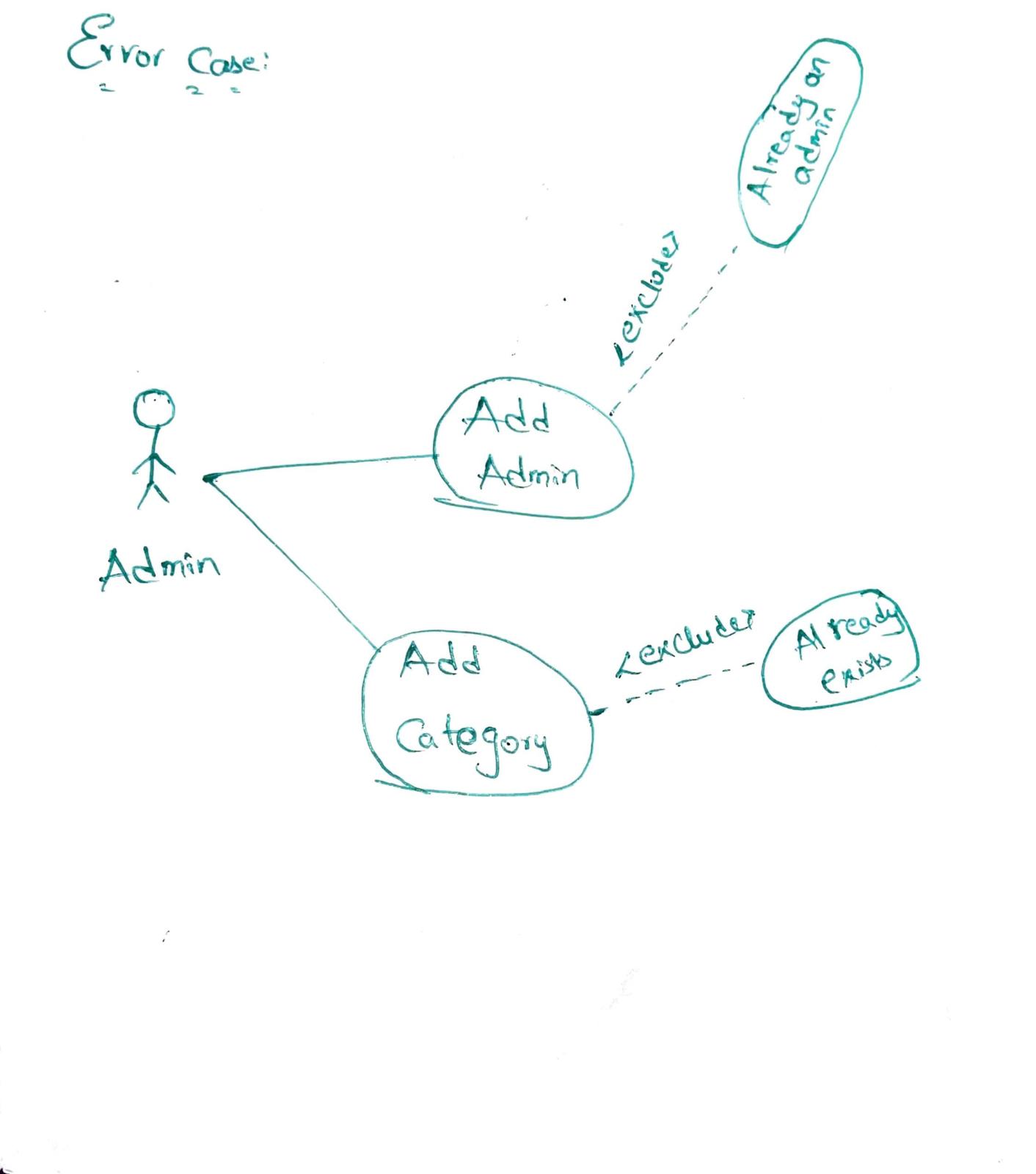
b. UML Diagrams:

1. Class Diagram:

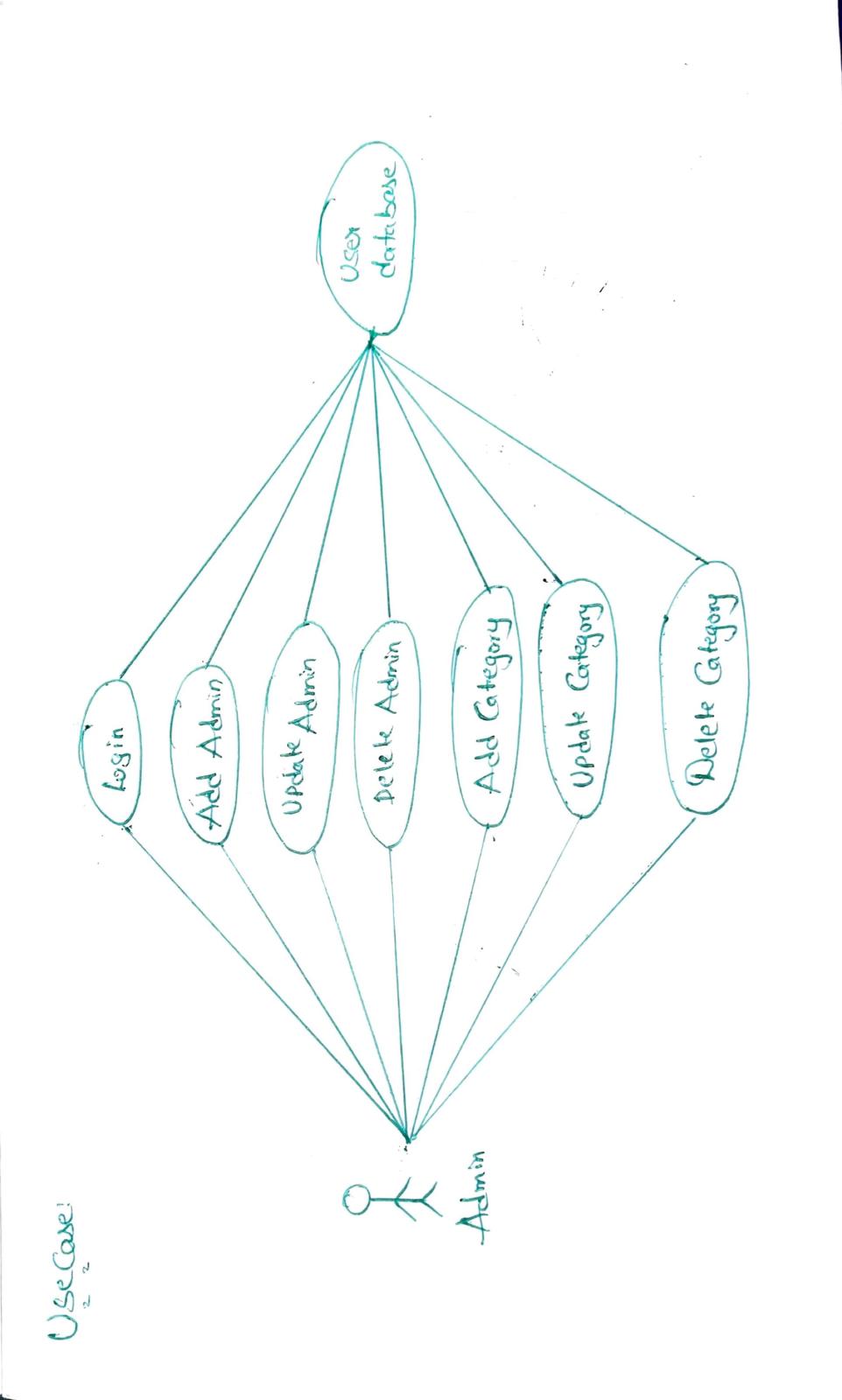


2. Use case Diagram: 

Error Case:



Sequence Diagram:



c. Test cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.no | Test case | Expected Results | Result | Remarks  (if fails) |
| 1. | Admin Login | Admin is able to login successfully. | Pass | If a valid admin with valid credentials is unable to login. |
| 2. | Creating a new admin | Existing Admin is able to add another admin for the system. | pass | Existing admin is unable to add a new admin. |
| 3. | Update an existing admin | Existing Admin should be able to update the details of the existing admins. | pass | Existing admin is unable to update the details of an existing admins. |
| 4. | Delete admin | Existing admin should be able to delete an existing admin completely from the system. | pass | Fails when the existing admin is unable to delete an existing admin. |
| 5. | Creating a new food category | Admin is able to add a new food category. | pass | Existing admin is unable to add a new food category. |
| 6. | Update an existing food category | Admin is able to update the existing food category. | Pass | Existing admin is unable to update the details of a food category. |
| 7. | Delete food category | Existing admin should be able to delete an existing food category. | Pass | Fails when the existing admin is unable to delete an existing food category. |
| 8. | Admin Logout | Admin should be able to logout successfully. | Pass | If admin is unable to logout from the system or the session doesn't terminate even after the admin logs out. |

d.

The bowl web application runs on the local host server we will be hosting the application on the xampp server and the web application is developed with php, html, Sql, css ,bootstrap .This web application allows the user to order the food from the following menu.

e. Instructions to run the project:

1) Any user or developer who wants to run the project in their system their system requirements should be:

a) Minimum Intel i3 configuration or higher

b) Ram: 4 GB or higher preferred

c) 5 GB disk space

2) First the user wants to install any IDE we prefer visual studio to write the code.

3) User wants to install the xampp server/ mamp for windows/ Mac respectively in their system.

4) The user wants to start the xampp server by starting their apache and myql in their xampp control panel.

5) The user wants to open PHP Myadmin and then create a database according to the connection file in the project. Then import the sql file to the database from the respective project folder.

6) Then the user can run the project by typing localhost/project\_folder\_name in their web browser.

Starting the Project in Detail:

* Download the project as zip.
* Extract the zip file.
* Place the extracted files in htdocs folder of xampp.
* Create a database in phpMyAdmin with the name “TheBowlDB”
* Import the “TheBowlDB.sql” into the database.
* Update the TheBowl1/config/constants.php file according to you database configurations.
* To start as the admin use the following link “localhost/thebowl1/admin/”
* To start as the user use the following link “localhost/thwbowl1/”

f. Feedbacks of the Peer review:

* How are you planning to carry the information changes made in one page to other page?

Actions taken:

* Using session in php we can handle this limitation and once the user logs in a session gets created and stays with the user until logout. So, mapping information to the user becomes easy and efficient.

g.

Functionality from phase 2 was implemented in phase 1, which is Dashboard. Home page, Sign In and Sign out features were implemented. The sign up feature and the sign in, sign out functionality were not implemented for user in this iteration (were only implemented for admin). In the next iteration, the left over functionalities will be implemented.

Due to functional dependencies between features, we were a bit confused on which part to implement first. This was the challenge the team faced.

h. Members Contribution:

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution  Description | Overall  Contribution (%) | Note  (if applicable) |
| 1. Siva Kumar Sai Pulavarthy | Core functionality  Skeleton | 100% |  |
| 2. Ganesh Bommisetty | Admin functionalities  Report: sequence diagram, class diagram | 100% |  |
| 3. Akhil Kata | Admin functionalities for category.  Report | 100% |  |
| 4.Shasidhar Jampana | Test cases | 100% |  |
| 5. Sandeep Gangullapattu | Report:  Usecase, instructions | 100% |  |
| 6. Satish Babu Nalajala | Report:  Feedbacks | 100% |  |