# ASSESSMENT 1

# eRestaurant Report

## Group 9

Contents

[Introduction 2](#_Toc67082723)

[Stakeholder Analysis 2](#_Toc67082724)

[Release Backlog 3](#_Toc67082725)

[Release & Iteration Plan 3](#_Toc67082726)

Use Case Model ………………………………………………………………………………………………………………………………….4

Non Functional Requirement………………………………………………………………………………………………………………..5

[Use Case Narratives 6](#_Toc67082727)

[Architecture Solution Options & Evaluation 11](#_Toc67082728)

[Architecture Solution Model Diagram 14](#_Toc67082729)

[Technology Stack, Proposed Solution 15](#_Toc67082730)

[Contribution Table 15](#_Toc67082731)

**Introduction**

The purpose of the project “eRestaurant Online System” is to exclusively provide a web application which is user friendly and reliable. The application has two modules one for administration and one for customer. The administration module allows the admin of to make changes to website such as updating the menu or managing customer invoices and the customer module allow customers to login into their to view available tables with date and time and making bookings.

## Stakeholder Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholder | Interest | Estimated Project Impact | Estimated Priority |
| Owner | Increase profits margin  Integration for future expansion  Better Accessibility | MED  HIGH  HIGH | 1 |
| Team Members | New tool to work with  Simplified booking process | LOW  HIGH | 2 |
| Customers | Ease of Access | LOW | 3 |

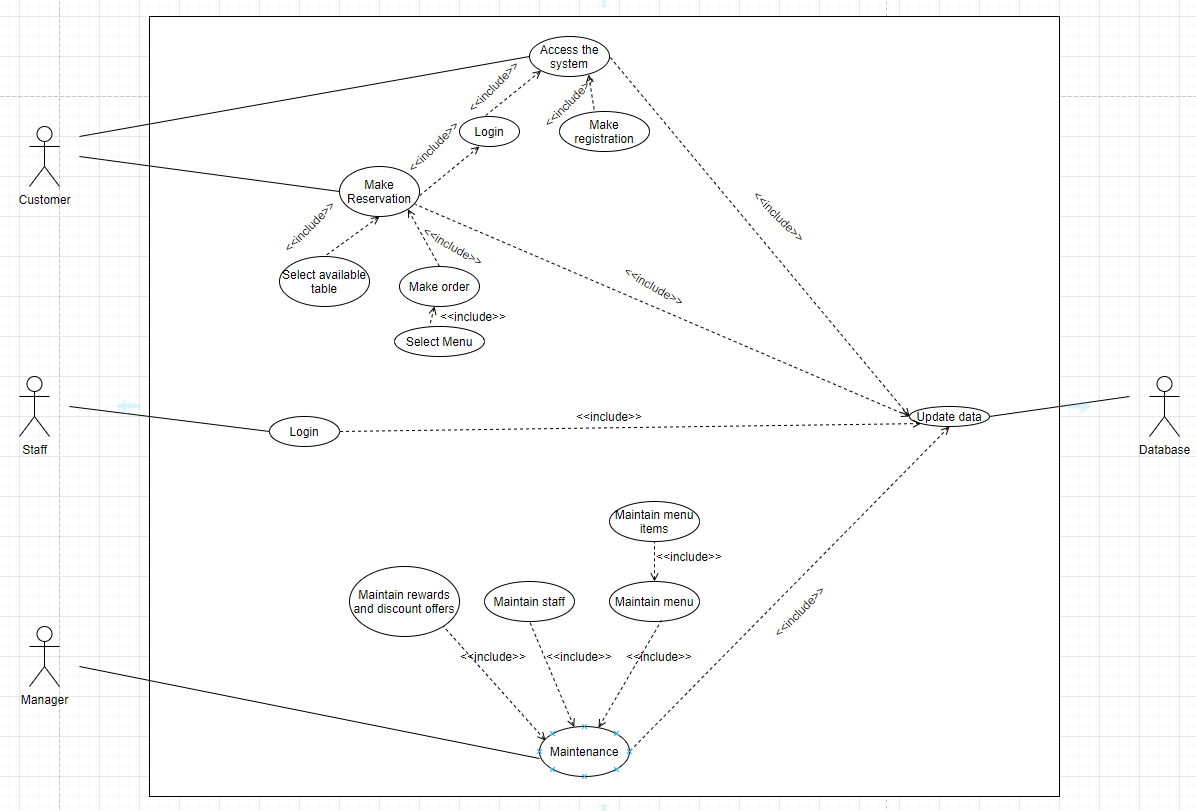
## Release Backlog

## Calendar Description automatically generated Release & Iteration Plan

Graphical user interface, text, application

Description automatically generated

**Use Case Model**



**Non-Functional Requirements**

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

**Compatibility**

* The program and website should be compatibility with all major browsers and operation systems such as Windows, IOS and Android.

**User Interface**

* The UI should be designed user friendly and need to consider the color blindness users.

**Security**

* Access should be controlled with account name and password.
* Only staffs can access the staff mode and other customer users cannot.
* Database should be secured, and loss of information should be prevented.

**Performance**

* The system should be up and running 24/7.
* The system should support at least 100 users using the online booking concurrently without any lag.

**Backup and Recovery**

* Backup is needed to prevent loss of data and database are down。

**System Maintenance**

* System maintenance should be conducted base on regular interval.
* It should be conducted on off-peak time.

## Use Case Narratives

**Use Case Name:** Register

|  |  |
| --- | --- |
| **Use Case ID** | U-101 |
| **User Story** | Customer wants to register their account. |
| **Goal** |  |
| **Priority** | High |
| **Actors** | Customer |
| **Pre-conditions** | Customer needs to have a valid email or phone number to register. |
| **Post-conditions** | Customer account will be created. |
| **Trigger** | Customer clicking register button on the website or app. |
| **Main Flow** | 1. Customer goes on restaurant website or app. 2. Customer clicks on register account. 3. System displays register profile page. 4. Customer enters first and second name. 5. Customer enters their email. 6. Customer enters their phone number. 7. Customer enters home address. 8. System sends validation email to customer. 9. Systems creates customer account. |
| **Exceptions** | 8.1 – If customer fail to validate their email address their account would not be created. |
| **Includes/Extends/Inherits** |  |
| **Supporting Information** |  |
| **Non-functional Requirements** | * As a customer I want to be able to register into my account from iOS, Android and Brower |

|  |  |
| --- | --- |
| **Alternate Flow 1** | 1. Customer goes on restaurant website or app. 2. Customer clicks on register account with google or Facebook. 3. System redirects customer to Facebook login page. |
| **Trigger** | Customer clicking on register with Facebook button. |
| **Step** | 1. Customer enters their login details with Facebook page. 2. Systems closes login page redirects customer to their account. |
| **Post-conditions** |  |
| **Exceptions** | * Customer unable to login into their Facebook account. |

**Use Case Name:** Login

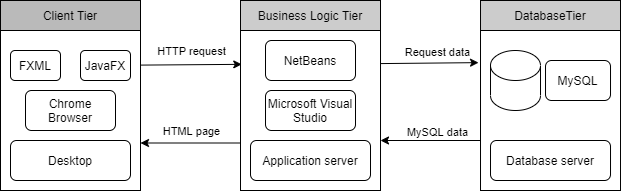
|  |  |
| --- | --- |
| **Use Case ID** | U-201 |
| **User Story** | Customer wants login into their account. |
| **Goal** | To make a booking or update their details. |
| **Priority** | High |
| **Actors** | Customer |
| **Pre-conditions** | Customer must an account. |
| **Post-conditions** | Customer gets access to their account. |
| **Trigger** | Customer Clicks on login button on website or app. |
| **Main Flow** | 1. Customer clicks on login button. 2. System displays login page. 3. Customer enters email and password. 4. Customer clicks on login button. 5. System verifies their account. 6. Customer gets access to their account. |
| **Exceptions** | 1. Customer enters wrong email. 2. Customer enters wrong password. 3. Customer doesn’t have an account. |
| **Includes/Extends/Inherits** |  |
| **Supporting Information** |  |
| **Non-functional Requirements** | * As a customer I want to be able to login into my account from iOS, Android and Brower |

|  |  |
| --- | --- |
| **Alternate Flow 1** | 1. Customer goes on restaurant website or app. 2. Customer clicks on login account with Facebook. 3. System redirects customer to Facebook login page. |
| **Trigger** | * Customer clicks on login with Facebook. |
| **Step** | * Customer is redirected to Facebook login page. * Facebook verifies their login details. * System redirects customer to their account. |
| **Post-conditions** |  |
| **Exceptions** | * Customer unable to login into their Facebook account. |

**Use Case Name:** Menu

|  |  |
| --- | --- |
| **Use Case ID** | U-401 |
| **User Story** | Customer wants to view the menu. |
| **Goal** |  |
| **Priority** | High |
| **Actors** | Customer |
| **Pre-conditions** | * Customer must have logged on (see login use case) |
| **Post-conditions** | * Customer views the menu. * Customer can filter the menu based on the ingredients. |
| **Trigger** | Customer clicks on menu button from their account |
| **Main Flow** | 1. Customer clicks on menu button. 2. Systems displays the menu page. 3. Customer filters their preferred ingredients. 4. Systems updates the menu according with customer’s preferred ingredients. |
| **Exceptions** | * Customer fails to login into their account (see login use case) |
| **Includes/Extends/Inherits** | * Includes use case 4.4 Menu use case. |
| **Supporting Information** |  |
| **Non-functional Requirements** | * As a customer I want to able print the menu |

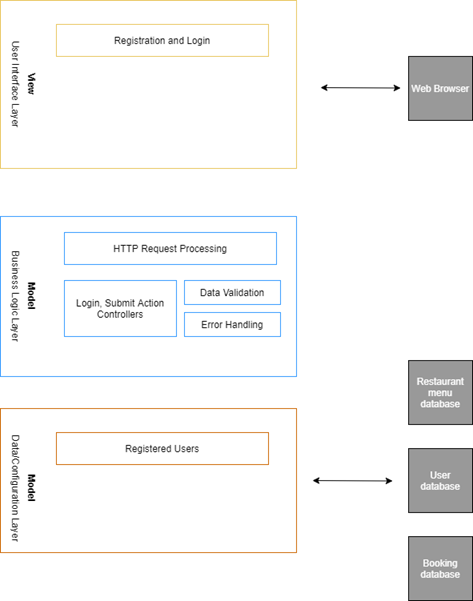
## Architecture Solution Options & Evaluation



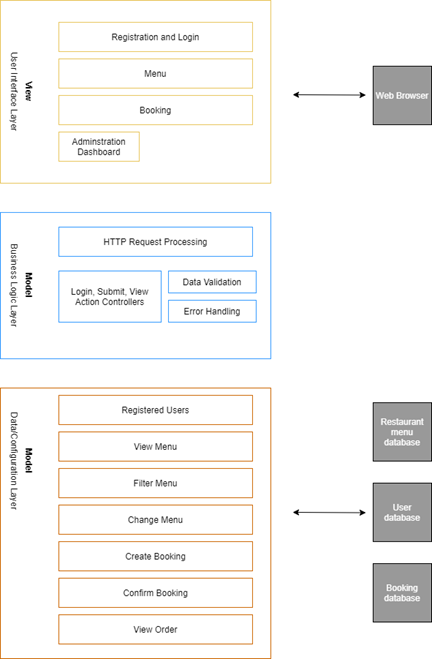
When considering an appropriate solution for the user interface (client tier), the software development teams skills and experience was a significant factor as it would greatly impact the quality of the system interface presented to customers. Thus, rather than using a language such as HTML which the development team has limited experience for, FXML (and its programming platform, JavaFX) was selected due to the proficiency the developers had with that language. Similarly, Netbeans was chosen as the programming environment due to its capacity to run FXML (unlike other Java environments such as BlueJ), as well as the development team’s knowledge and experience using the environment application. Netbeans was also selected due to its ability to synchronize with GitHub, which allows developers to efficiently access and share each other’s files during the project. For the database of the system, MySQL was chosen as it is a high performance, easy to use database management system that is also compatible with a variety of operating systems (such as windows, macOS and linux).

## Architecture Solution Model Diagram

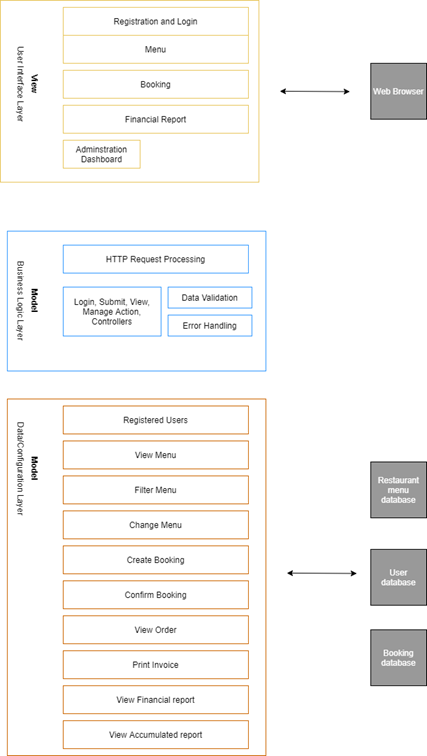
**Iteration 1**



**Release 1**



**Project 1**



## Technology Stack, Proposed Solution

The application is buildup of following tech stacks.

For website

* HTML is used for the structure of the website and its content.
* CSS is used for website style like its color, font, sizes and positioning.
* JavaScript is used to make the website work in the backend.

For iOS and Android

* Swift is used for iOS application.
* Java is used for android application.

For Server Side

* SQL is used for the database of application.

## Contribution Table

|  |  |
| --- | --- |
| **Student Number** | **Completed Tasks** |
| 13623905, Andie | Release Backlog, Release & Iteration, Stakeholder Analysis |
| 13924624, Chamod | Architecture Solution Options & Evaluation, Architecture Solution Model Diagram |
| 13421565, Zhihao | Use Case Model, Non-Functional Requirements |
| 14158226, Ahmad | Use Case Narratives, Technology Stack, Proposed Solution |
|  |  |
|  |  |