**Queue Project README**

**Team name: iQueue.**

**Student name: Low Jian Sheng and Brehmer Chan**

 Video Link : https://www.youtube.com/watch?v=TDSjSFMx9Gs

**Table of content:**

1: Overview of our project

* Scope of the project
* What it solves
* Why is it important?
* Features

2: Future plans for our project

**1.1 Scope of the project**

Our project aims to create an electronic queue system for patrons and stall owners in NUS.

Audience of project : Patrons and stall owners of NUS canteens.

Level of project: Project Gemini.

**1.2 What it solves**

It solves the issue of long queues and waiting time for food especially during peak hours. It also removes the hassle of having to physically queue and order your meals on campus. It also allows students to consume their meals on time and be able to manage their timetable more effectively.

We believe this will save a student's time spent queueing for food and allow them to stick to their tight schedules, ensuring they remain punctual for lectures as well as tutorials.

**1.3 Why is it important**

As a student in NUS, one may not have the time to queue for their food between their packed schedules. Many students hence find themselves skipping meals or missing out on delicious food due to long queues in the canteen. With this application, we aim to eliminate this hurdle between the students and the food that they crave for. We feel that there is a need to be addressed since it is quite common for students to miss their lunch as they rush for their classes or end up late for lectures due to long queues in canteens.

**1.4 Features/ user stories**

**1.4.1 IVLE login**

 Our application allows integration with student's IVLE account so store owners will be able to keep track of users who ordered their food.  This will save the hassle of creating yet another new account to use this application and having to remember another password. As a stall owner, this will allow me to ensure that users are held accountable for the food that they order and will minimise food wastage in the case the user did not collect their food after ordering.

**1.4.2 Browsing through canteens**

Our web-app showcases a basic canteen layout with stall tabs that users can toggle through. It also includes a dropdown button where users can browse through a variety of food sold by the individual stalls. Each food item is tagged with a basic description to allow users to have a better idea of what they are ordering.

**1.4.3 Cart function**

After browsing the food items, users can add their selected food item into an electronic "cart". They can then proceed to the cart page where they are able to change the quantity of the food item they ordered or remove them completed before making the finalised checkout.

**1.4.4 Electronic Queue System**

To eliminate the need for a physical queue system, users will be able to “join” in an electronic queue and order their food through the application.

**1.4.5 Scalability for different screen sizes**

Our application is scalable in terms of devices of differing screen sizes, showing the appropriate aspect ratio on all devices (phones, tablets, laptops, desktop).

**1.4.5 Database to store customer's orders**

All of the orders that user's checkout in the cart page will be stored into our database, and the store owners will GET the information from the database.

The following is list of user stories that we felt is important for our application to be successful.

* As a user, our app should be reliable in terms of accurately serving the correct type and number of food items.
* As a user, our app should be efficient in terms of serving time.
* As a user, our app should work on all devices.
* As a user, our app should have a simple and easy to use interface.
* As a stall owner, the app should ensure accountability for those who ordered the food.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Jian Sheng (Student A) | Brehmer (Student B) | Description |
| 090516 | 12 | 12 | Orbital Liftoff day 1 |
| 100516 | 12 | 12 | Orbital Liftoff day 2 |
| 160516 | 5 | 0 | Read up on HTML, CSS and basic JavaScript |
| 270516 | 0 | 5 | Read up on HTML, CSS and basic JavaScript |
| 280516 | 6 | 6 | Write up on README, discuss on project specifics. |
| 280516 | 4 | 0 | Learn about basic AngularJS directives and expressions. |
| 290516 | 5 | 0 | Continue learning about basic AngularJS routings, directives |
| 310516 | 5 | 0 | Learnt BootStrap for UI design |
| 060616 | 5 | 5 | Met up to discuss on UI design. Designed home page, navigation bar. |
| 070616 | 5.5 | 5.5 | Met up to discuss on UI design. Designed sign up page. 6 basic canteen pages with no design done. |
| 080616 | 8 | 6 | Fixed alignment and layout of signup page. Added angular directives into pages. Done routing of views. Migrated from sublime text to Web storm. Decided on Node.js for backend server development. |
| 090616 | 6 | 6 | Learnt Auth0 and re-read AngularJS. Imported angular and auth0 packages into the project. |
| 150616 | 6 | 6 | Improved design of main layout page and inclusion of 6 canteen pictures |
| 260616 | 0 | 6 | Created basic selection menu for The Terrace@Biz page without design |
| 270616 | 0 | 3 | Improved design of The Terrace@Biz page |
| 010716 | 0 | 5 | Learnt BootStrap for UI design |
| 020716 | 0 | 8 | Learnt more on javaScript and started on cart page |
| 090716 | 0 | 8 | Added basic cart functionalities |
| 100716 | 0 | 8 | Improved cart functionalities and cart page |
| 130716 | 0 | 8 | Finalised cart table and linked cart to other pages |
| 150716 | 0 | 3 | Added carousell layout to the mainpage of website |
| 150716 | 6 | 0 | Rerouted the views of the angular front end. |
| 160716 | 6 | 0 | Added in access controls for the routing. |
| 170716 | 5 | 5 | Learn about heroku server and node.js. |
| 180716 | 5 | 0 | Configured basic node.js server with express. |
| 190716 | 6 | 0 | Added in ajax calls to IVLE api. Tried to configure proxy server due to CORS error but failed. Using third party proxy server when fetching data from the ivle api. |
| 200716 | 8 | 8 | Connect our app to heroku server and finally hosted on the server. |
| 210716 | 5 | 0 | Further read up on node.js syntax and workings. |
| 220716 | 6 | 0 | Redesigned the default page (unlogged in) and the logged in page |
| 240716 | 7 | 0 | Added cart pages and orders pages together with store page for the store owners. |
| 250716 | 6 | 0 | Read up on mongoDB, mongoose and the other dependencies required to set up a database on node.js. |
| 270716 | 7 | 0 | Set up mongoDB with node.js. |
| 280716 | 0 | 8 | Worked on store page to display user’s orders and display cart table |
| 300716 | 5 | 0 | Set up mongoDB with node.js. Hosted database online with mlabs from heroku. |
| 010816 | 7 | 7 | Changed the way the order system works. |
| 020816 | 8 | 8 | Debugged the application. Write up log and prepare video. |
|  | 164.5 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total | 79.5 | 72.5 |  |

Reference:

[www.google.com](http://www.google.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

<https://docs.angularjs.org>

<http://adrianmejia.com/blog/2014/10/03/mean-stack-tutorial-mongodb-expressjs-angularjs-nodejs/>

<https://devcenter.heroku.com/articles/mean-apps-restful-api>

<https://www.sitepoint.com/creating-crud-app-minutes-angulars-resource/>

<http://mongoosejs.com/docs/>

<https://expressjs.com/en/4x/api.html>

<https://nodejs.org/en/docs/>

<https://www.youtube.com>