COMP4905 Project Proposal - WAITLESS

Jessica Tiberio
101085366
Fall 2021
Supervised by Lou Nel

Overview

The proposed project, WAITLESS, is an endeavor to create an interactive online menu where restaurants can upload their menu to the application, specifying ingredients within each dish. Diners will be able to scan a QR code to load the restaurant's menu, enter their dietary restrictions and receive recommendations for safe food and drinks to order. Ordering will be completed through the menu web application so that the restaurant chefs are able to receive orders digitally through the application's restaurant dashboard thereby eliminating the inevitable delay between receiving a menu and ordering food items.

Main Objectives

The main objective of this project is to develop a MEAN full stack web application that restaurants can easily adopt and implement as an efficient social distancing solution to slow and disorganized in person ordering processes. Additionally, WAITLESS menu will eliminate the ambiguity of meal ingredients so that those with dietary restrictions can instantly and clearly identify which dishes are safe.

Hardware & Software Requirements

This project should not require any specific hardware but will be developed on a Dell machine with Windows 10 OS. This application will be predominantly written in Javascript, with HTML and CSS. The project will be developed in accordance with the MEAN full stack framework; therefore, MongoDB will be used as the database, Express and Angular will be used in the back-end and front-end, respectively, and Node will be used as Javascript's runtime environment. The graphics editor, Figma, will be used to design user interface prototypes prior to programming. Additionally, Github will be used during development to backup files for version control.

Expected Deliverables

Upon completion, this project is expected to contain:
- WAITLESS web application containing four views (Registering Restaurant View, Restaurant Dashboard View, Edit Menu View and Patron View);
- Source code written in Javascript using MEAN stack framework;
- Documentation accompanying the source code, describing how to use the web application;
- Patron View will be accessible through QR codes that are uniquely generated for each restaurant's tables;

- Written report detailing the project, challenges and development experience, including test cases and testing data;

Schedule

Milestone Period (2021)		Milestones
Start Date (Monday)	End Date (Sunday)	
September 6th	September 19th	• Complete UI designs for 4 views in Figma • Set up programming environment • Preliminary research on QR codes, web sockets and MEAN stack framework
September 20th	October 3rd	• Complete Restaurant Register View (user interface, database, and programming logic)
October 4th	October 17th	• Complete Patron View (user interface, database, and programming logic)
October 18th	October 31st	• Complete Restaurant Dashboard and Edit Menu View (user interface, database, and programming logic)
November 1st	November 14th	• Complete web sockets to connect Patron View with Restaurant Dashboard
November 15th	November 28th	• Complete algorithm for automatically labelling menu dishes with dietary restrictions tags based on list of ingredients
November 29th	December 12th	• Complete QR codes and testing
December 13th	December 17th	• Complete application documentation, project report and submit the final project.

Not like Existing Course Assignments

This project is not like existing course assignments due to the complex scope of the endeavor. Most courses focus on one element of software development, such as databases, user interfaces or creating basic web applications. This application requires full stack development which demands utilizing skills from several previous courses in a much larger scope of work compared to that of course projects. Additionally, new technologies that are not currently taught at Carleton will be investigated. For example, utilizing Angular on the front-end, designing the user interface with Figma and incorporating QR codes. I also anticipate that working with web sockets may be an additional challenge.