

Software Project

**Sean Kelly**

**N00225867**

Software Project

e.g., Develop a PC or Sports shoe website

Year 2 2022-23

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

|  |  |
| --- | --- |
| GitHub | https://github.com/y2-SW-project/swproject23-Seankelly-Developer |
| Video | Link to your video file (MS Stream, YouTube) |

Table of Contents

[1 Introduction 1](#_Toc96009471)

[2 Business Concept 2](#_Toc96009472)

[2.1 Business Idea 2](#_Toc96009473)

[2.2 Business model 2](#_Toc96009474)

[2.3 Market Research 2](#_Toc96009475)

[2.4 Marketing/Advertising 2](#_Toc96009476)

[2.5 Suppliers 2](#_Toc96009477)

[2.6 Competitors 2](#_Toc96009478)

[2.7 Employees 2](#_Toc96009479)

[2.8 Environmental Impact 2](#_Toc96009480)

[3 Requirements 3](#_Toc96009481)

[3.1 Introduction 3](#_Toc96009482)

[3.2 Requirements gathering 3](#_Toc96009483)

[3.2.1 Similar applications 3](#_Toc96009484)

[3.2.2 Interviews 3](#_Toc96009485)

[3.3 Requirements modelling 3](#_Toc96009486)

[3.3.1 Functional requirements 3](#_Toc96009487)

[3.3.2 Non-functional requirements 3](#_Toc96009488)

[3.3.3 Use Case Diagrams 4](#_Toc96009489)

[3.4 Feasibility 4](#_Toc96009490)

[4 Web application Design 6](#_Toc96009491)

[4.1 Layout 6](#_Toc96009492)

[4.2 Interaction 6](#_Toc96009493)

[4.3 Colour schemes 6](#_Toc96009494)

[4.4 Font choices 6](#_Toc96009495)

[4.5 Wireframes 6](#_Toc96009496)

[5 Database Design 7](#_Toc96009497)

[5.1 Description 7](#_Toc96009498)

[5.2 Business Reporting Requirements 7](#_Toc96009499)

[5.3 Textual Representation of Data-Set 7](#_Toc96009500)

[5.4 Business Rules 8](#_Toc96009501)

[5.5 Entity Relationship Diagram 8](#_Toc96009502)

[5.6 Tables 9](#_Toc96009503)

[5.7 Database Dictionary 10](#_Toc96009504)

[6 System Design/ Architecture Overview 11](#_Toc96009505)

[6.1 Introduction 11](#_Toc96009506)

[6.2 Model View Controller 11](#_Toc96009507)

[6.3 User Authenticaion 11](#_Toc96009508)

[6.4 Routing 11](#_Toc96009509)

[6.5 Templating 11](#_Toc96009510)

[7 Testing 12](#_Toc96009511)

[7.1 Introduction 12](#_Toc96009512)

[7.2 Functional Testing 12](#_Toc96009513)

[7.2.1 Login/Registration 13](#_Toc96009514)

[7.2.2 Navigation 13](#_Toc96009515)

[7.2.3 Calculation 13](#_Toc96009516)

[7.2.4 CRUD 13](#_Toc96009517)

[7.2.5 Discussion of Functional Testing Results 14](#_Toc96009518)

[7.3 User Testing 14](#_Toc96009519)

[7.4 Conclusion 14](#_Toc96009520)

[8 Project Management 15](#_Toc96009521)

[8.1 Introduction 15](#_Toc96009522)

[8.2 Project Phases 15](#_Toc96009523)

[8.2.1 Requirements 15](#_Toc96009524)

[8.2.2 Design 15](#_Toc96009525)

[8.2.3 Implementation 15](#_Toc96009526)

[8.2.4 Testing 15](#_Toc96009527)

[8.3 SCRUM Methodology 15](#_Toc96009528)

[8.4 Project Management Tools 16](#_Toc96009529)

[8.4.1 Github Project 16](#_Toc96009530)

[8.4.2 GitHub 16](#_Toc96009531)

[9 Reflection 17](#_Toc96009532)

[9.1 Your views on the project 17](#_Toc96009533)

[9.2 How could the project could be developed further? 17](#_Toc96009534)

[9.3 Assessment of your learning. 17](#_Toc96009535)

[9.4 Completing a large software development project 17](#_Toc96009536)

[9.5 Technical skills 17](#_Toc96009537)

[9.6 Further competencies and skills 17](#_Toc96009538)

[10 References 18](#_Toc96009539)

# Introduction

Overall aim

Application area

Technologies

PHP, MySQL, Bootstrap, CSS, Vanilla

Tools

IDE, phpMyAdmin, Miro

Project management

GitHub

Business Concept

Requirements

Design

Implementation

Testing

Reflection

# Business Concept

## Business Idea

**Restaurant reservation booking system.** Although there are existing organizations that provide this service, I believe that there is a unmet need in the market with regards to finding a restaurant based on a meal you are searching for. Certain restaurants advertise themselves solely on the nationality of their cuisine and the customer must look through the internet to find out what specific meals they offer on their menu. My idea is that the customer can not only search for restaurants based on the type of restaurant they’re looking for, but also specify a meal-type and be displayed a list of restaurants offering that dish, or something similar. From there, the user is able to make a reservation. This eliminates the need to look up menu’s and reviews as the site will also display a star rating based on reviews imported from google reviews and the like. So, in conclusion, I am proposing a reservation system for restaurants with advanced search filters and the capability to create, change and delete reservations. Each user will have an account and their reservations will be saved to their account. The site will also include other features for convenience such as a general price range, map of the area with the restaurants location and more.

## Business model

The business would, over time build a regular customer base and restaurants can pay to be listed higher-up on the site or recommended to appropriate customers based on their past reservations. Advertising will account for any revenue the business may generate.

I believe restaurant goers will use this service to make reservations as all the relevant information that they would be looking for would all be in once place so it is a more efficient way of digesting information. Also, the convenience of being able to find restaurants offering specific dishes is beneficially in terms of taste, dietary requirements and ease-of-use in terms of searching for a desired location. The reservation service would also be completely complementary as revenue would be generated from the advertisements on the page.

## Market Research

The market consists of 15 million restaurants worldwide. The restaurant industry accounted for 1.2 trillion US Dollars in 2020 and has grown since then. Many people frequent these restaurants and surveys have learned that 4% of these people make reservations ahead of time. So the market is open to a service that provides easily digestible information, all in one place. Restaurants I believe would also benefit from using the site to advertise on as it will generate customers.

## Marketing/Advertising

Revenue generated for the site will be from restaurants that would like their advertisements to be pushed to appropriate users more frequently than a free advertisement.

Other advertisements would be industry-related and shown on the main pages.

Initially, the site will be loaded with data from local menus, restaurants and areas. Reviews can be imported from google reviews. The site will be advertised using facebook’s ad service. Facebook generates a pixel which can track user behavior and target appropriate people on their social media platforms. A budget of 500 per week for 4 weeks with the goal of achieving 2000 active users would be active during initial release.

## Suppliers

Facebook will be supplying the site with advertising, as well as TikTok after its major success in the social media industry. Restaurants will upload their own profiles to advertise their menu’s. Reviews can initially be imported from google reviews, yelp etc.

## Competitors

* **OpenTable (Irish Reservation Website)**

OpenTable is a site that provides customers with restaurants in order to make reservations. This site does show ratings and reviews as well as the ability to search based on location and cuisine. However, they are limited to Ireland whereas my idea is to cater for anywhere in the world as restaurants could upload their advertisements over time it could spread worldwide as well as nationwide. OpenTable also doesn’t provide results based on a specific meal or dietary requirement.

## Employees

Designer

Responsible for designing the website. Should document progress, test and deploy a functioning website.

Marketing director

Responsible for creating advertisements, Ad sets, reporting and monitoring results of each ad set.

Customer support director.

Communicates with customers, users and restaurants in order to ensure a satisfactory experience for all.

Network Security Engineer

In charge of finding and reporting on vulnerabilities and eradicating them. Ensuring that site follows GDPR standards and requirements for data protection.

## Environmental Impact

Average websites produce just over 1.7g of CO2 for every page view. Using a green hosting service can reduce this which will be taken into consideration.

Examples of services that consider the environmental impact of hosting a site include; GreenGeeks and Hostpapa. These providers focus on using renewable energy sources for their servers.

# Requirements

## Introduction

* Site should provide a platform for restaurants to upload their menu, location, ratings and opening hours.
* Provides a platform for customers to find local restaurants, to their search specifications and view the details provided by the restaurant and other users who have reviewed their experience.
* Allows user to search based on either rating, dietary requirements, menu items, location or type of cuisine.
* Allows customer to make a booking inquiry, which the business can then accept, decline or suggest an alternative time, which the customer may then accept or decline.
* Allows user to login and register and view their existing bookings.
* User can request to edit their reservation time/date or make a cancellation if required.
* Site links customer to a mailto link that allows them to contact the business directly should they have any questions.

## Requirements gathering

### Similar applications

|  |  |
| --- | --- |
| Site Name | OpenTable |
| Function | Search function with parameters date, time, amount of people, cuisine, location and specific restaurants |
| Advantage | * Easy to find, top of homepage * Can search based on several criteria * Can select time, date and amount of people. * Easily digestible results. |
| Disadvantage | * Can’t search based on multiple criteria in one query. * No way to quickly view menus * Not able to search based on desired meal or dietary requirements. |
| Screenshot | *Search bar on homepage*    *Results shown upon searching based on suggested criteria and pre-filled time/date etc.* |

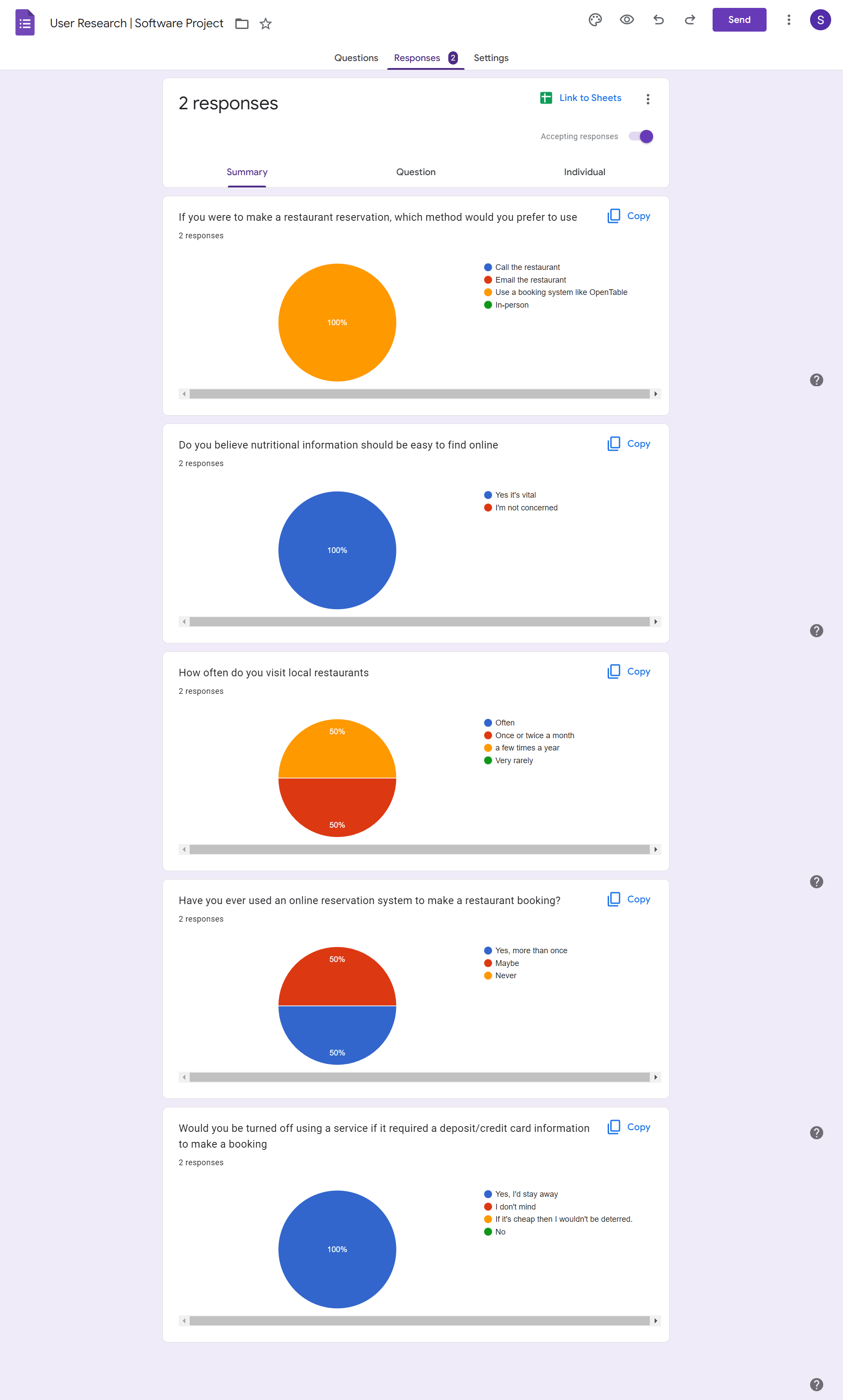
|  |  |
| --- | --- |
| Site Name | OpenTable |
| Function | Booking a timeslot |
| Advantage | * 5 minute timer to complete booking * Booking information clearly displayed on page header * Minimal personal information required |
| Disadvantage | * Credit card required * Sign-up required prior to accessing booking form |
| Screenshot | *Initial form*    *Credit card form – required for booking* |

|  |  |
| --- | --- |
| Site Name | OpenTable |
| Function | Featured advertisements layout |
| Advantage | * Clear card layout * Right scroll clearly shown * Available times (samples) shown on card * Imagery used as heading * Slogan/prompt shown. * Reviews shown |
| Disadvantage | * Hard to digest large amount of advertisements as this layout is used multiple times one after another on the homepage. |
| Screenshot |  |

### Interviews

<https://docs.google.com/forms/d/e/1FAIpQLScQ59DoZK6pT4sXs8TGsELZKm0p89xA096-z8DC-sIa_9wW7w/viewform?usp=sf_link>

I have linked the google form I’ve used above, the purpose of this survey is to find out more about what the user wants and what the market is like in terms of the online restaurant booking space. I have also shown the results of the survey below;



## Requirements modelling

### Functional requirements

* Create a booking with a restaurant for a specific time, time & amount of guests.
* Store user’s bookings on their account for later review, editing or cancellations.
* Search for restaurants based on criteria including; dietary requirements, location, menu items and ratings.
* View featured collections of restaurants without search criteria.
* Contact support with any queries about the service provided.

### Non-functional requirements

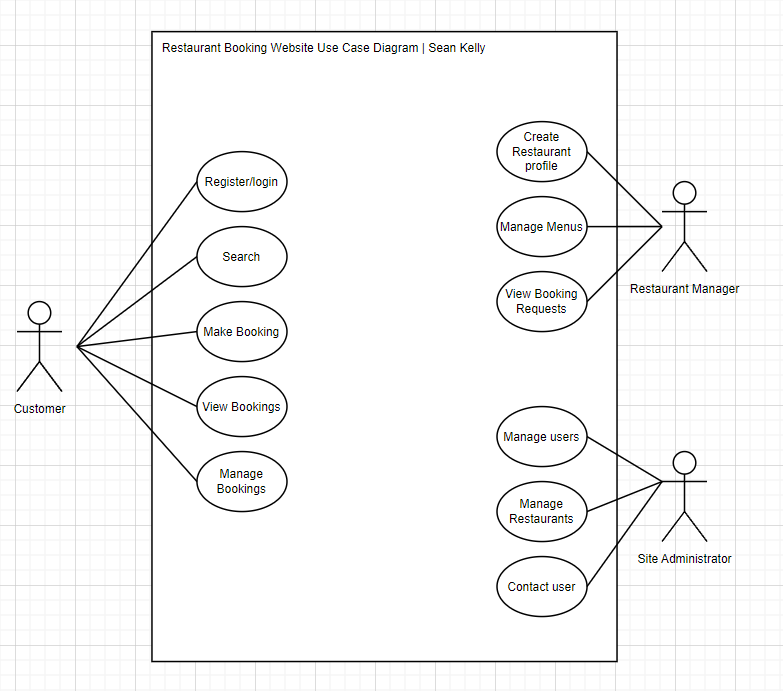
Below I have listed important non-functional requirements for consideration during the project’s lifecycle and also during the maintenance stage upon release of the site;

**Security**: This is important as data-protection needs to be focused on to comply with GDPR regulations and guidelines. Hackers can gain access to sensitive information regarding our users and if security is not at a reasonable level the company is acting irresponsibly toward it’s users.

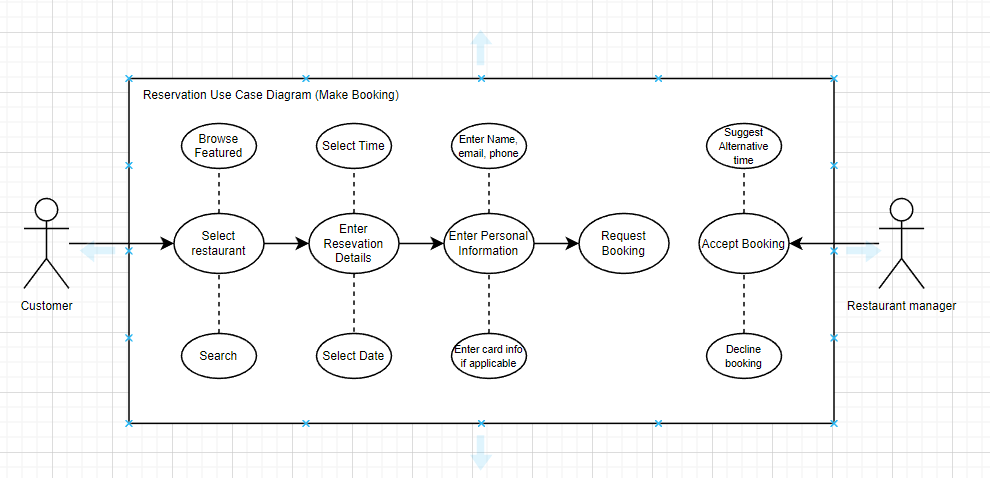
**Reliability:** Reliability needs to be considered in order to allow the user’s experience to be satisfactory and in order for them to be able to view their bookings and any other information provided by themselves into the system.

**Performance:** Speed tests using various hosts need to be carried out in order to gauge the average speed provided to our users in order to make sure that the user’s experience is satisfactory and data is never lost in transit.

### Use Case Diagrams



*Main Functions and Corresponding Actors*



*Making a Reservation*

## Feasibility

* Laravel will be used as a PHP web framework to allow for website development as well as a connection to a SQL Database.
* The site will be developed on a windows 11 PC using Visual Studio Code software as the source code editor.
* Hosting will be provided by a Renewable-energy focused organization which will generally be slightly more expensive but better in terms of a carbon footprint.
* Upon analyzing the requirements, software used and providers used I believe this project is feasible.

# Web application Design

## Layout

External sources & information

<https://m3.material.io/>

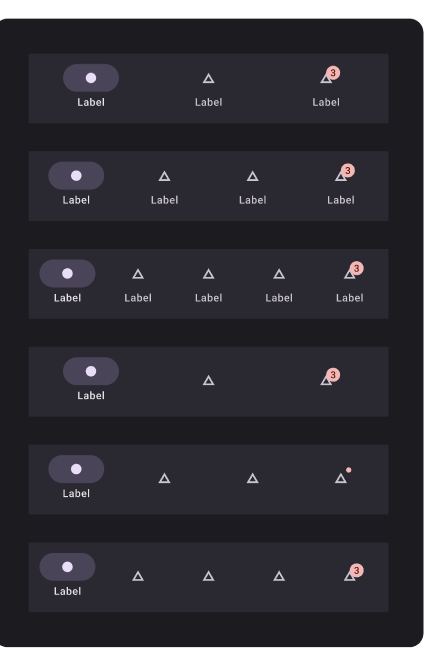
<https://laravel.com/>

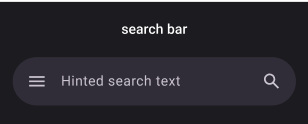
<https://www.apachefriends.org/>

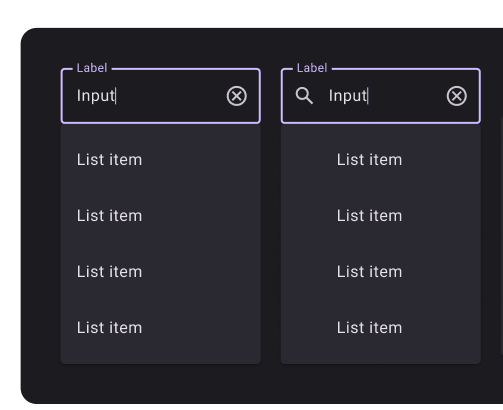
<https://getbootstrap.com/>

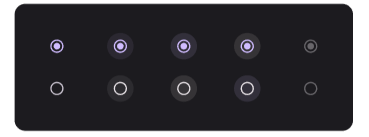
<https://nodejs.org/en/>

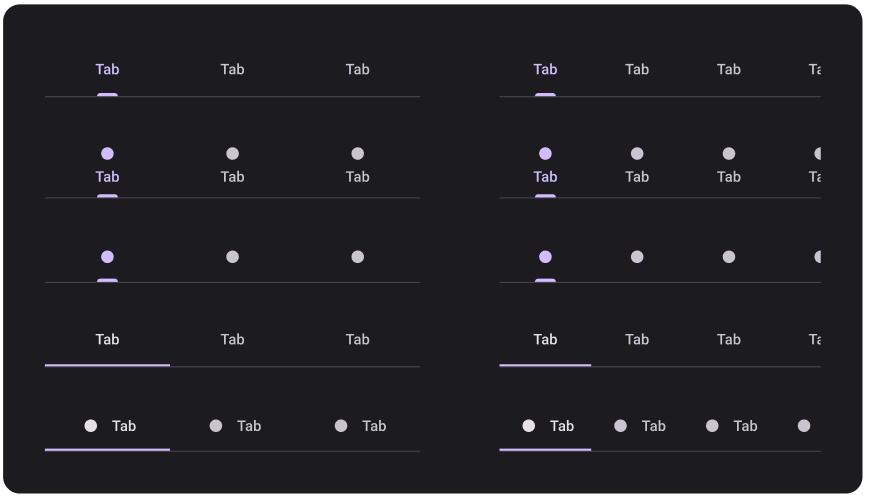
## Interaction

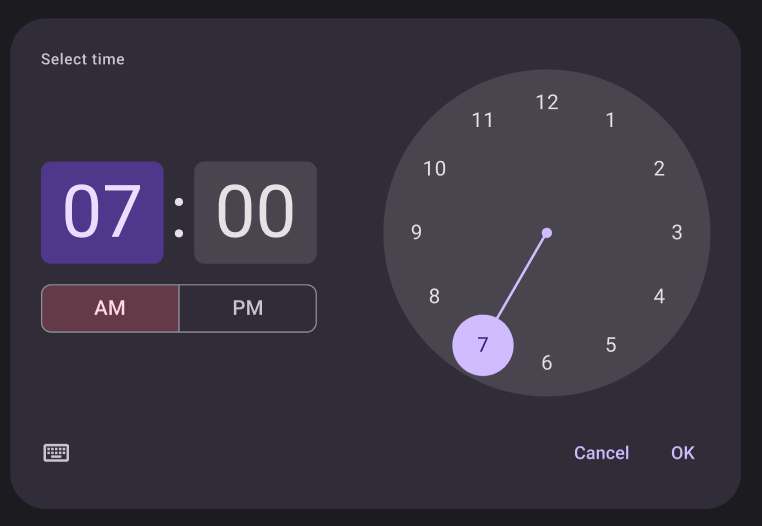


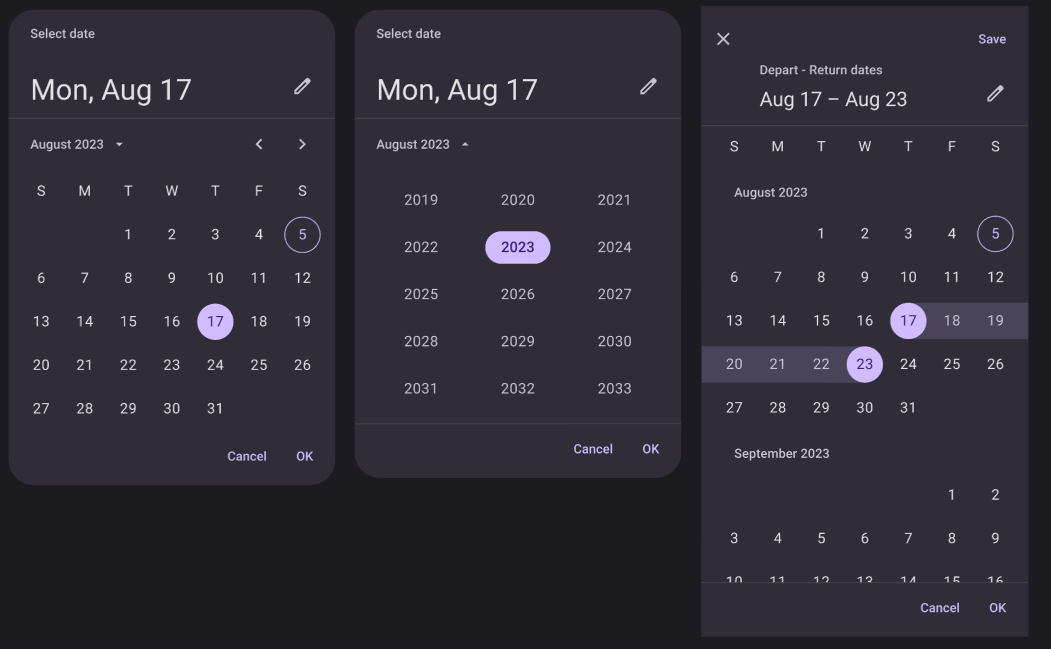




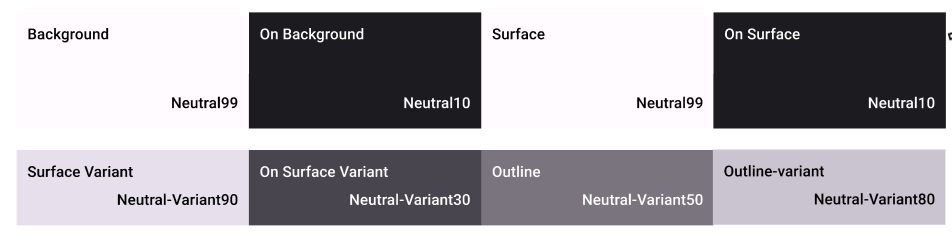




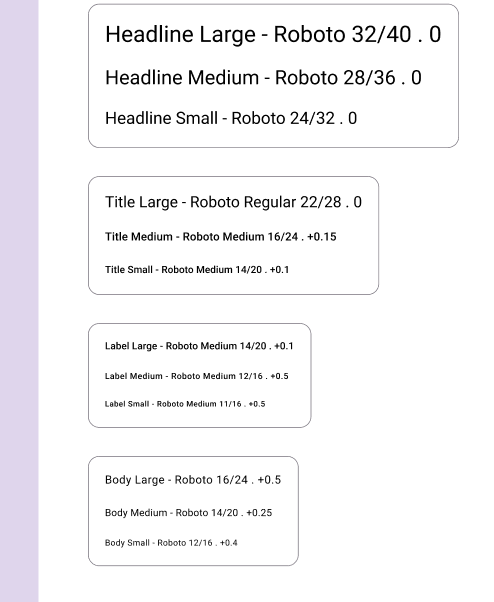




## Colour schemes



## Font choices



## Wireframes

Link to my figma;

<https://www.figma.com/file/TvmEyu1qVGWlqb0GxYkln6/Untitled?node-id=0%3A1&t=KrDp5ppW5bqHMGHN-0>

My figma includes high fidelity wireframes to outline the design plans for my site. I have used Google’s Material 3 design system and my design corresponds to the style of that system.

# Database Design

## Description

This report outlines the database design for a web application that enables customers to make restaurant reservations. The application will allow customers to search for available tables at their preferred restaurant, choose their preferred time and date, and make a reservation. The database will store information about restaurants, such as location, capacity, and opening hours. For each reservation, the database will store the customer's name, contact information, number of guests, time and date of the reservation, and any special requests. The system will also store information about tables, including table number, capacity, and availability. Additionally, the application will require customers to create an account to make a reservation, and the database will store their personal information, such as name, email, and phone number

## Business Reporting Requirements

1. Restaurant Managers need to be able to create, read, update, and delete: restaurants, tables, and menu items.
2. Users will need to be able to find all available tables at a restaurant ordered by their preferred time and date.
3. Users may want to find available tables at a restaurant by a specific time and date.
4. Users need to find restaurants by location and the location needs to be displayed on a Google Map.
5. Users may need to find restaurants by cuisine type.
6. Users need to find the table number for a specific reservation.
7. Users need to find the menu items for a specific restaurant.
8. Managers need to display a list of reservations for a specific restaurant.
9. Managers may need to find the list of customers who made reservations at a specific restaurant.
10. Users need to be able to view their past reservations and upcoming reservations.

## Textual Representation of Dataset

RESTAURANT:

id (PK)

name

location

cuisine\_type

opening\_hours

image\_id

TABLE:

id (PK)

restaurant\_id (FK)

table\_number

capacity

is\_available

RESERVATION:

id (PK)

table\_id (FK)

customer\_name

customer\_email

customer\_phone

number\_of\_guests

reservation\_time

special\_requests

CUSTOMER:

id (PK)

name

email

phone\_number

password

LOCATION:

id (PK)

address

city

state

zip\_code

latitude

longitude

CUISINE\_TYPE:

id (PK)

cuisine\_name

MENU\_ITEM:

id (PK)

restaurant\_id (FK)

item\_name

item\_description

item\_price

MANAGER:

id (PK)

restaurant\_id (FK)

employee\_id (FK)

RESERVATION\_HISTORY:

id (PK)

reservation\_id (FK)

customer\_id (FK)

is\_upcoming (boolean)

reservation\_date

## Business Rules

1. Only managers can create, read, update, and delete restaurants, tables, and menu items.
2. Users can search for available tables at a specific restaurant based on their preferred time and date.
3. Users can search for available tables at a specific restaurant by a specific time and date.
4. Users can search for restaurants by location, and the location should be displayed on a Google Map.
5. Users can search for restaurants by cuisine type.
6. Users can find the table number for a specific reservation.
7. Users can view the menu items for a specific restaurant.
8. Managers can display a list of reservations for a specific restaurant.
9. Managers can view the list of customers who made reservations at a specific restaurant.
10. Users can view their past and upcoming reservations.

## Entity Relationship Diagram

Substitute in here your ERD from draw.io



## Tables

Substitute in here your tables and the relationships between tables from draw.io in the format you used in DBMS with Mohammed.



## Database Dictionary’s

Table: Restaurant

| **Attribute** | **Data Type** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| id | int |  | Yes | PK |  |
| name | varchar | up to 100 characters | Yes |  |  |
| location | varchar | up to 100 characters | Yes |  |  |
| cuisine\_type | varchar | up to 100 characters | Yes |  |  |
| opening\_hours | varchar | up to 100 characters | No |  |  |
| image\_id | int |  | No |  |  |
|  |  |  |  |  |  |

Table: Table

| **Attribute** | **Data Type** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| id | int |  | Yes | PK |  |
| restaurant\_id | int |  | Yes | FK | Restaurant |
| table\_number | int |  | Yes |  |  |
| capacity | int |  | Yes |  |  |
| is\_available | boolean |  | No |  |  |

Table: Reservation

| **Attribute** | **Data Type** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| id | int |  | Yes | PK |  |
| table\_id | int |  | Yes | FK | Table |
| customer\_name | varchar | up to 100 characters | Yes |  |  |
| customer\_email | varchar | up to 100 characters | Yes |  |  |
| customer\_phone | varchar | up to 20 characters | Yes |  |  |
| number\_of\_guests | int |  | Yes |  |  |
| reservation\_time | datetime |  | Yes |  |  |
| special\_requests | varchar | up to 100 characters | No |  |  |

Table: Customer

| **Attribute** | **Data Type** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| id | int |  | Yes | PK |  |
| name | varchar | up to 100 characters | Yes |  |  |
| email | varchar | up to 100 characters | Yes |  |  |
| phone\_number | varchar | up to 20 characters | Yes |  |  |
| password | varchar | up to 100 characters | Yes |  |  |

Table: Location

| **Attribute** | **Data Type** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| id | int |  | Yes | PK |  |
| address | varchar | up to 100 characters | Yes |  |  |
| city | varchar | up to 100 characters | Yes |  |  |
| state | varchar | up to 100 characters | Yes |  |  |
| zip\_code | varchar | up to 20 characters | Yes |  |  |
| latitude | float |  | No |  |  |

Table Name: cuisine\_type

Attributes:

id (primary key)

cuisine\_name

Menu Item

Table Name: menu\_item

Attributes:

id (primary key)

restaurant\_id (foreign key to Restaurant)

item\_name

item\_description

item\_price

Manager

Table Name: manager

Attributes:

id (primary key)

restaurant\_id (foreign key to Restaurant)

employee\_id (foreign key to Employee)

# System Design/ Architecture Overview

* 1. Introduction

Laravel is a PHP web application framework that follows the Model-View-Controller (MVC) architecture pattern. It provides a range of built-in functionalities and features that make it easier for developers to build web applications quickly and efficiently.

Laravel's internal functionality can be divided into various components, such as the routing system, middleware, the database layer, the Eloquent ORM, the Blade templating engine, and more. The routing system allows developers to define application routes and map them to controller actions. Middleware provides a way to execute code before and after a request is processed. The database layer offers a simple and intuitive API for database interactions, while the Eloquent ORM provides an easy way to work with database records. The Blade templating engine offers a powerful way to build and organize application views.

Overall, Laravel's internal functionality is designed to help developers write clean, maintainable, and efficient code for web applications

* 1. Model View Controller

Model-View-Controller (MVC) is a design pattern commonly used in web applications. It separates the application into three interconnected parts: the Model, which manages the data and business logic, the View, which is responsible for rendering the user interface, and the Controller, which handles user input and updates the Model and View accordingly. This separation improves maintainability, scalability, and flexibility of the application.

* 1. User Authentication

User authentication is the process of verifying the identity of a user who wants to access a web application. In Laravel, user authentication involves confirming a user's credentials, such as their email and password, before granting them access to protected areas of the application.

Laravel provides built-in authentication features, including authentication controllers, middleware, and password reset functionality. Developers can easily implement these features into their Laravel applications, making it simple to authenticate users and manage user sessions.

Laravel's authentication system uses encryption and hashing to secure user passwords and prevent unauthorized access. Developers can also customize the authentication process to fit the specific needs of their application, such as implementing two-factor authentication or using social login providers.

* 1. Routing

In Laravel, routes are a way to define how a web application should respond to HTTP requests. Routes are defined in the routes/web.php file, and they associate a URI with a controller action. When a user visits a specific URL, Laravel's router matches the URL to a defined route, and then dispatches the associated controller method to handle the request.

For example, if you have a route defined for /users that maps to a UserController, when a user visits /users, the index() method in the UserController is called to handle the request.

* 1. Templating

Laravel's templating engine Blade, a powerful and intuitive tool for building dynamic web pages. It allows developers to write clean, concise templates using familiar syntax, while providing a range of useful features such as template inheritance, control structures, and easy integration with PHP code. Blade also includes features like automatic escaping of user input to prevent XSS attacks, and precompiled templates for faster performance. With its flexible and expressive syntax, Blade makes it easy to build complex, dynamic web applications with minimal hassle, making it a popular choice among Laravel developers.

Sequence Diagram under construction.

|  |  |
| --- | --- |
|  |  |

# Testing

* 1. Introduction

This chapter describes the testing that has been undertaken for the application. This chapter is presented in two sections:

1. Functional Testing
2. User Testing

Functional testing is a type of software testing whereby the system is tested against the functional requirements. The app is tested by looking to see if the actual output for a given input corresponds with the expected output. The tests should be based on the requirements for the app. The results of functional testing can indicate if a piece of software is functional and working, but not if the software is easy to use.

User testing looks to see if a piece of software is easy and intuitive for the user.

* 1. Functional Testing

This section describes the functional tests which were carried out on the app. These functional tests can be categorised as: (whatever is relevant to your app)

* Login/Registration
* Navigation
* Calculation
* CRUD

Functional testing generally uses a Black Box Testing technique which means that the internal logic of the system being tested is not of interest to the tester. The tester is only interested in whether the actual output agrees with the expected output.

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Discussion of Functional Testing Results

Describe the results from the tests. Address any functionality where unexpected behavior could not be debugged.

* 1. User Testing
  2. Conclusion

Discussion of test results.

# Project Management

## Introduction

This chapter describes how the project was managed. It shows the phases of the project, going from the project idea through the requirements gathering, the specification for the project, the design, implementation and testing phases for the project. It also discusses GitHub as a tool which assists in project management.

## Project Phases

In this section, describe each of the following project phases. Explain any issues which arose for each of the phases.

### Requirements

### Design

### Implementation

### Testing

Include a Gantt chart



## SCRUM Methodology (optional)

Sprints

## Project Management Tools

### GitHub Project

Description

Include screen shots

How it worked in practice

### GitHub

Description

How it is used

How it worked in practice

# Reflection

## Your views on the project

Describe how you feel the project went from your perspective.

## How could the project be developed further?

## Assessment of your learning.

Critically assess your learning. List what skills and competencies you have learned developed in this Continuous Assessment.

List which part of the project would need further development and itemize where you feel you have not satisfactorily completed the continuous assessment.

## Completing a large software development project

Describe what you have learnt from the project, from the point of view of completing a large software development project.

## Technical skills

Describe what you have learnt from the project, from a technical skills viewpoint.

## Further competencies and skills

Describe any extra competencies and skills that would help you with your development in the workplace.

# References

Add a list of references that you used to complete the project.

The Department of Technology and Psychology in IADT uses APA 7th referencing style.

Use alphabetical order for your references.

This site gives details about how to cite websites using APA:

https://www.wikihow.com/Cite-a-Website-in-APA

The following is a useful site for creating citations for APA for websites.

<http://www.citationmachine.net/apa/cite-a-website>

You can also use the Referencing tab within Microsoft Word to enter reference information manually. Word then creates an APA style reference.