# UNIVERSITY OF PLYMOUTH

# MAL2017

## Software Engineering 2

## Coursework 2

**BSSE2409163**

# Table of Contents

## Contents

# 1. Introduction

The Mother Pearl Restaurant app was designed with a focus on simplicity and ease of use, offering customers an efficient way to reserve tables for breakfast, lunch, or dinner. My primary goal was to create a seamless experience that reduce complexity by allowing users to effortlessly book their seats. By eliminating the need for direct interaction with staff during the reservation process, The reservation progress will become smooth and hassle-free experience.

We will be looking into the thought process behind the development. Here is what we will cover

- Background

The context and user scenarios for the app.

- Legal, Social, Ethical, and Professional (LSEP) considerations

Privacy and security issues we addressed during development.

- Design

Screenshots and a look at the system's structure.

- Implementation

Key development steps and technologies we used.

- Testing and Evaluation

Summary of user testing.

- Summary

Key features and lessons learned.

We also included links to helpful resources::

GitHub Repository

[mainassessment-TanggaPendek]

(https://github.com/Plymouth-University/mainassessment-TanggaPendek)

-YouTube Video

[Application Demo]

(https://youtu.be/LD_WL-jaY7c)

These resources provide access to the source code and a visual demonstration of the app.

# 2. Background

Mother Pearl Restaurant, a popular seaside restaurant, want to improve aims to its customer experience by introducing a mobile application for table reservations. The app allows customers to book tables for breakfast, lunch, or dinner. The app provides a simple reservation progress to reduce staff interaction which can become time consuming.

Users have the control of choosing their seats, type of meal and date to personalize their reservation. The seats provided are fixed but can be up to 10 people. Reservation management is also easier with the app when the customer can edit or cancel the reservation as soon as needed. Push notifications provided to ensure that user know if the reservation is created, changed, or even cancelled. This is possible due to personal account integration to make sure the customer knows the status of the reservation.

This app will be focused on customer across the world with its sleek UI and intuitive UX. Which will make it easier to use even for the old.

# 3. Legal, Social, Ethical, and Professional (LSEP) Considerations

The development of the Mother Pearl Restaurant Reservation App involved addressing key legal, social, ethical, and professional concerns to ensure it meets industry standards and user expectations.

**Legal Considerations**

1. **User Data Privacy**:
    - The app collects usernames and phone numbers as credentials. To comply with data protection regulations (e.g., GDPR or CCPA), user data is stored in Firebase.
    - Only the minimum information is collected to reduce risks.
2. **Data Retention**:
    - User data is kept only as long as needed for functionality. Once the reservation is completed, any non-essential data is deleted.
3. **Compliance with Restaurant Policies**:
    - The app adheres to the restaurant's policies for and maximum table size.

**Social Considerations**

1. **Accessibility**:
    - The app is designed with a clean, intuitive UI to make it accessible for a broad audience.
    - Visual elements such as different text sizes ensure important information is easy to notice.
2. **User Convenience**:
    - By eliminating the need for direct staff interaction, the app simplifies the reservation process.
3. **Notifications**:
    - Push notifications keep users updated on reservation statuses and changes.

**Ethical Considerations**

1. **Fair Access**:
    - The app ensures equal access to all users by displaying available tables.
2. **Data Usage**:
    - User data is strictly used for its intended only and is not shared with third parties.

**Professional Considerations**

1. **Security**:
    - Firebase provides secure authentication to protect against unauthorized access.
    - Sensitive operations, such as account creation and reservation management, are implemented with validation checks to prevent errors.
2. **Reliability**:
    - Efforts have been made to ensure that the app functions consistently, with error prevention mechanisms like input validation and try-catch blocks to handle unexpected crashes.

# 4. Design

The design of the Mother Pearl Restaurant mobile application prioritizes a clean, modern aesthetic with a focus on simplicity and ease of use. Key design choices were made to reflect the restaurant's seaside theme.

**User Interface (UI) Design**

The app's primary colour scheme features a soothing ocean-like hue, #006D77, to evoke feelings of calm and relaxation. The rounded, circular design elements were chosen for their soft, approachable nature. Additionally, a blurred beach background subtly reinforces the theme without overwhelming the user experience, making it both attractive and easy on the eyes.

Typography is used in the design to improve clarity and bring some visual guide to the important part of the data while maintaining good visual.

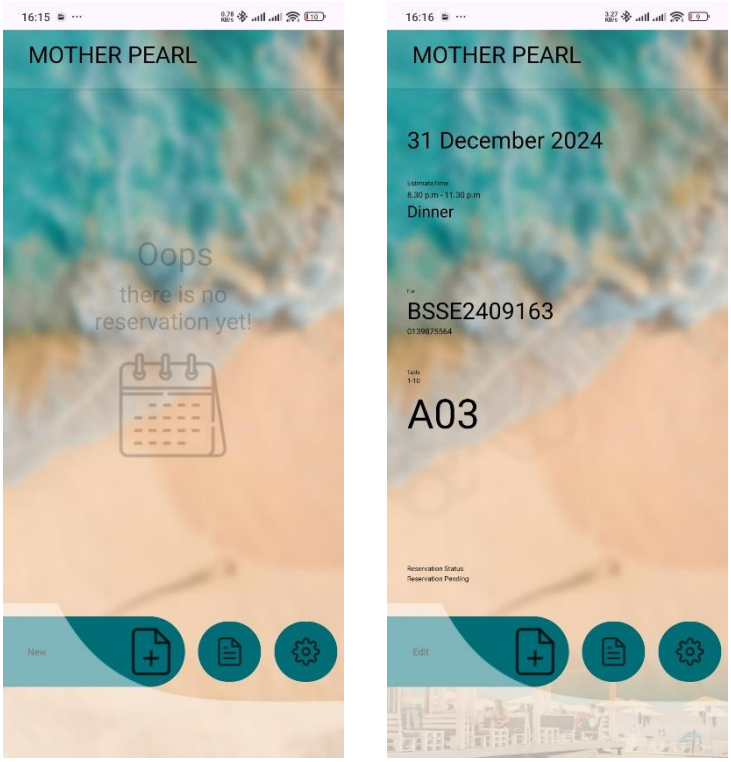


*Figure 1: Example of use of colour and typography*

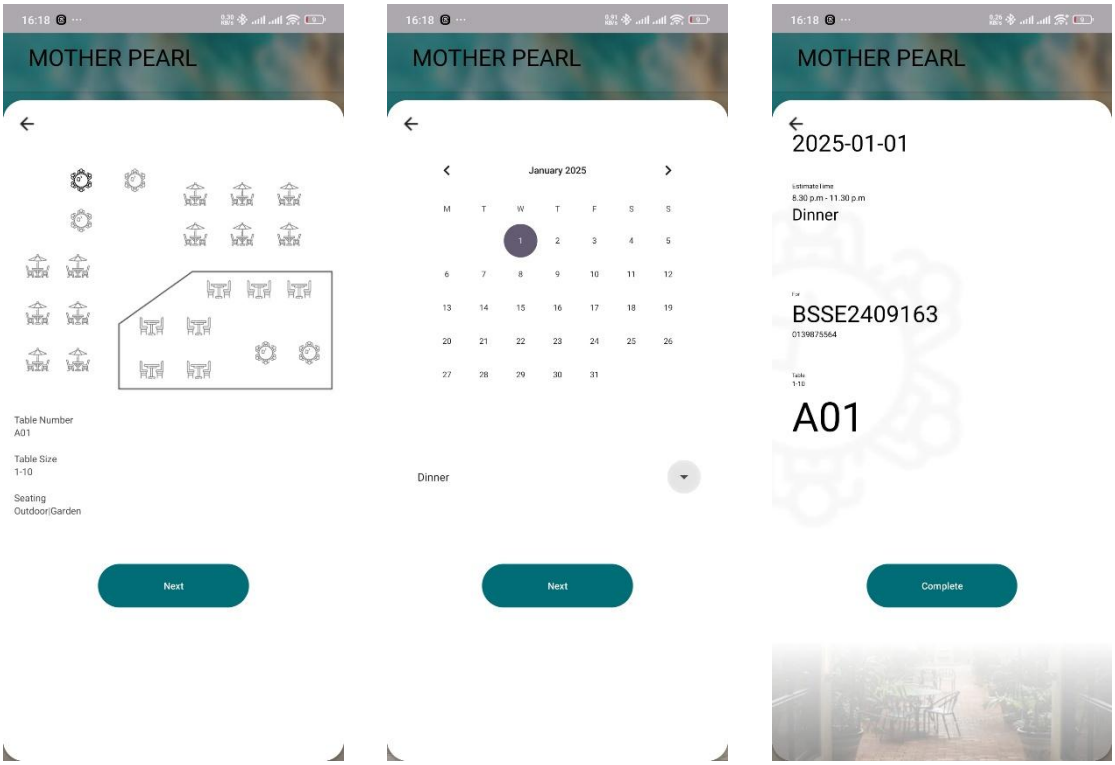


*Figure 2: Progress of creating a reservation*

**System Architecture**

The app uses a client-server architecture, where data is managed and accessed through two primary sources: a REST API and Firebase.

- **REST API:** The REST API handles the essential data operations, including fetching and inserting reservation information. This ensures that the app's core functionality remains stable and efficient.
- **Firebase:** Firebase is used to manage user authentication and additional data to help strengthen the core functionality. This integration enhances the app's ability to provide personalized experiences for each user, including managing their reservations securely.

**Design Patterns**

Singleton pattern was used to increase data consistency and easier synchronization between database.

# 5. Implementation

The development of the Mother Pearl Restaurant mobile application was carried out using Android Studio with Java as the programming language. This is due to its strong and solid foundation for building an efficient Android application. The core components of the app include Firebase for user authentication and additional data storage, and Retrofit2 for fetching the main reservation data from the server.

**Technologies Used**

- **Android Studio & Java:** These tools were used to build the application, providing a comprehensive IDE for coding and testing the app.
- **Firebase:** Firebase serves multiple purposes in the app. It handles user authentication, ensuring secure access to the app's features. Firebase also stores extra data.
- **Retrofit2:** Retrofit2 is used for fetching reservation-related data from the provided REST API. The app employs a **Singleton** pattern for fetching data, ensuring efficient and centralized management of API calls. Firebase also helps mitigate the API's limitations by enabling smooth integration with the server.

**Table Reservation Process**

The app's reservation feature is divided into three key stages, each designed to simplify the reservation process:

1. **Selecting Date and Meal Type:** Users choose a date for their reservation along with the type of meal. The meal type determines the time slots for reservation.
2. **Selecting Seat:** Users then choose their seating placement, which includes options for inside or outside seating. This step ensures that users can find a suitable spot based on their group size and seating preference.
3. **Confirmation:** Before the reservation is finalized, users confirm the details of their booking. The app inserts the reservation data into both Firebase and the REST API, ensuring synchronization across the system.

**Development Challenges**

During development, several challenges happened due to limitations in the provided REST API:

- **Reset Time**: The API's "reset time" created synchronization difficulties when integrating data with Firebase or other databases, leading to delays in updates.
- **Data Structure**: The API's data structure was not flexible enough to meet user requirements, making it impossible to rely solely on the API while delivering a fully functional app.
- **Limited Data Storage**: The API's attribute constraints resulted in insufficient data for designing the app's intended features. To address this, additional data was stored in Firebase to bridge the gap.
- **Shared Use**: The API was shared with other systems and introducing unnormalized data risking crashes or breaking functionality across both ends.

# 6. Testing and Evaluation

Testing for the Mother Pearl Restaurant mobile application was primarily conducted through manual testing. Given the limitations of automated testing knowledge, the testing process relied heavily on debugging techniques and error tracking.

**Testing Methods**

- **Brute Force Testing:** The application underwent extensive brute force testing to ensure that the flow of the reservation process was smooth and functional. By repeatedly interacting with the app, various bugs were identified and addressed.
- **Code Logging:** To help track down bugs and errors, logs were inserted before, during, and after key processes in the code. This allowed for easier identification of where problems were happening, and helped reduce search time.
- **Android Crash Reports:** When the app encountered crashes, Android's crash reporting system provided valuable error messages, helping pinpoint the exact line of code where the issue occurred. This provided a targeted approach to fixing bugs.

**Usability Heuristics**

- **Visibility of System Status:** Visibility was the top priority throughout the design and testing. The app was designed to ensure that users were always informed of the status of their reservation and any actions taken.
- **Consistency:** Consistency was ensured using Firebase, which synchronized data across the app. By using Firebase for user authentication and data storage, the app will be consistent across different user sessions.
- **Error Prevention:** Error prevention was addressed by implementing a combination of the manual testing methods mentioned above, as well as using try-catch blocks in the code to prevent crashes. This ensured that the app could handle unexpected input or failure points gracefully.

**Strengths and Weaknesses**

- **Strengths:** The app excels in its UI/UX design, offering a clean, intuitive interface. The navigation is user-friendly, and the design prioritizes clarity without overloading the user with excessive information.
- **Weaknesses:** The main weakness lies in the database syncing capabilities. The provided REST API has restrictive and reset-prone characteristics, which can sometimes result in delays or inconsistencies when syncing data between the app and the server. This could be addressed in the future with a more stable API or additional synchronization methods.

# 7. Summary

This report shows an overview of the development and implementation of the Mother Pearl Restaurant mobile application. The app was designed to simplify the reservation process for customers by allowing them to book tables for breakfast, lunch, or dinner, with seating preferences and meal type options. It incorporates features like account creation, reservation management, and push notifications to enhance the user experience.

The app's UI/UX design was a key strength, ensuring that users can navigate the reservation process with ease and clarity. Firebase was used to maintain consistency in user data and manage authentication, while Retrofit2 helped fetch the necessary reservation data. The app’s architecture was based on a client-server model, with the server provided via a REST API and Firebase integrated for additional functionality.

In terms of testing, manual methods such as brute force testing and error logging were used to identify and address bugs. The app was designed with usability in mind, particularly prioritizing visibility , consistency, and error prevention.

Despite its strengths, the app encountered limitations due to the REST API’s restrictive nature, which impacted database synchronization. However, the user experience remains solid, and improvements in the future could include resolving this issue and enhancing syncing mechanisms.

# Reference

Google. (n.d.) *Android Developers*. Available at: https://developer.android.com (Accessed: 18 December 2024).

Google. (n.d.) *Firebase Documentation*. Available at: https://firebase.google.com/docs (Accessed: 18 December 2024).

Square. (n.d.) *Retrofit: Type-safe HTTP client for Android and Java*. Available at: https://square.github.io/retrofit/ (Accessed: 18 December 2024)