

# **SMART DESK BOOKING SYSTEM**

**TEY MEI QUN**

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY**

**MULTIMEDIA UNIVERSITY**

**JULY 2024**

# **SMART DESK BOOKING SYSTEM**

**BY**

**TEY MEI QUN**

**THE PROJECT REPORT IS PREPARED FOR**

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY  
MULTIMEDIA UNIVERSITY  
IN PARTIAL FULFILLMENT  
FOR**

**BACHELOR OF COMPUTER SCIENCE  
B.I.T. (HONS) ARTIFICIAL INTELLIGENCE**

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY**

**MULTIMEDIA UNIVERSITY**

**JULY 2024**

© 2024 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED

Copyright of this report belongs to Universiti Telekom Sdn. Bhd as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialization policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

## **DECLARATION**

I hereby declare that the work have been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

A handwritten signature in black ink, appearing to read "TEY MEI QUN". It consists of several loops and curves.

---

TEY MEI QUN

Faculty of Information Science & Technology  
Multimedia University

Submission Date: 9/07/2024

Submission Time: 12:00PM

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to the wonderful people who supported the success of my project.

First and foremost, I am very grateful to my supervisor, the respected Professor , Dr. Ong Lee Yeng. Her tireless support, valuable guidance and infinite patience were the backbone of this project. I am truly blessed to have the privilege of learning from such a wise and inspiring supervisor.

To my loving parents and friends, your unwavering support, tireless help, and constant encouragement are the wind that keeps my wings flying. Your trust in me is the source of my strength. Without you by my side, I would not be able to complete this journey.

The success of this project is not only by mine alone. It's a demonstration to the collective effort, intelligence, and love that went into this entire project. Thank you ,from the bottom of my heart ,for being a part of this extraordinary journey.

## **ABSTRACT**

In nowadays world of dynamic workspaces, adaptability and flexibility have become key elements of the way of work. Flexible workspaces are reshaping human work environments by giving employees the ability to choose a workspace based on the unique needs. Especially in the wake of the COVID-19 pandemic, with companies mostly shifting from traditional working models to hybrid or work-from-home, so ensuring available workspace is crucial.

The Smart Desk Booking System project is a response to the growing need for enhanced flexibility and efficiency in office space management in company. The main purpose is to develop a multi-tasking web and mobile application that enables employee and administrators to simplify seat booking within the company.

Providing employee with a remote platform to reserve seats through an inborn seating map, allowing them to plan their workspace up to a week in advance. Additionally, the proposed project provides employee with the ability to register by using QR code.

For administrators, the system introduces an additional layer of control and convenience. Administrators can prioritize desk bookings on behalf of other employee, extending the booking period beyond the usual one-month limit. They can also manage desk assignments and availability efficiently.

## TABLE OF CONTENTS

<b>DECLARATION .....</b>	<b>IIII</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>IVV</b>
<b>Abstract .....</b>	<b>V</b>
<b>Table of contents.....</b>	<b>VI</b>
<b>List of tables .....</b>	<b>VIII</b>
<b>List of figures .....</b>	<b>IX</b>
<b>LIST OF ABBREVIATIONS/ SYMBOLS .....</b>	<b>XV</b>
<b>List of appendices .....</b>	<b>XVI</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 Overview .....	1
1.2 Problem Statement.....	2
1.3 Project Objectives.....	3
1.4 Project Scope .....	4
1.5 Report Organisation.....	4
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>7</b>
2.1 The smart desk system.....	7
2.2 QR Code .....	10
2.3 Comparison between Existing Systems.....	27
<b>CHAPTER 3 METHODOLOGY .....</b>	<b>29</b>
3.1 Introduction of Agile model .....	29
3.2 Gantt chart .....	31
3.3 Task List .....	32
3.4 Tools technique and programming language.....	33
3.5 Framework.....	37
3.6 Library .....	39
<b>CHAPTER 4 PROPOSED SYSTEM .....</b>	<b>41</b>
4.1 Use Case .....	41
4.2 Context Diagram.....	42
4.3 Data Flow Diagram .....	43
4.4 Flow chart diagram.....	44
4.5 Data Dictionary.....	49
4.6 Initial prototype .....	50
<b>CHAPTER 5 IMPLEMENTATION .....</b>	<b>59</b>
5.1 System Architecture .....	59

5.2	Admin Interface .....	63
5.3	Employee Interface.....	78
5.4	Challenges Encountered .....	89
5.5	Variation of the Project Objective .....	90
<b>CHAPTER 6 TESTING EVALUATION.....</b>	<b>92</b>	
6.1	Login.....	92
6.2	View and Manage Booking list .....	94
6.3	Customizing the date range for generating reports.....	99
6.4	Map management .....	100
6.5	Employee View Map .....	101
6.6	QR code .....	102
6.7	Case Study .....	104
6.8	Limitation .....	112
<b>CHAPTER 7 CONCLUSION .....</b>	<b>113</b>	
7.1	Summary.....	113
7.2	Future Works .....	114
<b>REFERENCES .....</b>	<b>115</b>	
<b>APPENDICES.....</b>	<b>118</b>	

## **LIST OF TABLES**

<b>Table 2.1: Contrast between 4 similar system and proposed system.....</b>	<b>28</b>
<b>Table 4.1: Entity Table.....</b>	<b>49</b>
<b>Table 5.1: Summary of Objective Variation Table .....</b>	<b>90</b>

## LIST OF FIGURES

<b>Figure 2.1: Evaluation of an office (Halim, N, 2021).....</b>	8
<b>Figure 2.2: Timeline phases of smart office (Hartman, V.,2023)......</b>	9
<b>Figure 2.3: QR Information's Format (Fajar et al., 2023) .....</b>	11
<b>Figure 2.4: QR basic structure (Ho, W. E et al., 2022) .....</b>	11
<b>Figure 2.5: LiquidSpace booking page (LiquidSpace, 2017) .....</b>	12
<b>Figure 2.6: LiquidSpace date available selection (LiquidSpace, 2017) .....</b>	13
<b>Figure 2.7: LiquidSpace view calendar page (LiquidSpace, 2017) .....</b>	13
<b>Figure 2.8: LiquidSpace application booking process (LiquidSpace, 2017) .....</b>	14
<b>Figure 2.9: LiquidSpace application booking confirmation (LiquidSpace, 2017) .....</b>	14
<b>Figure 2.10: LiquidSpace search query (LiquidSpace, 2017).....</b>	15
<b>Figure 2.11: Racquetvenues booking page (Racquetvenues, n.d.).....</b>	16
<b>Figure 2.12: Racquetvenues bottom of booking (Racquetvenues, n.d.) .....</b>	16
<b>Figure 2.13: Racquetvenues mobile view on website (Racquetvenues, n.d.).....</b>	17
<b>Figure 2.14: Racquetvenues booking detail (Racquetvenues, n.d.) .....</b>	17
<b>Figure 2.15: Racquetvenues desktop view and mobile view on the mobile phone (Racquetvenues, n.d.) .....</b>	18
<b>Figure 2.16: Racquetvenues registration error (Racquetvenues, n.d.).....</b>	19
<b>Figure 2.17: GSC Homepage (Golden Screen Cinemas, n.d) .....</b>	20
<b>Figure 2.18: GSC date selection(Golden Screen Cinemas, n.d) .....</b>	20
<b>Figure 2.19: GSC seat selection (GSC, n.d.).....</b>	21
<b>Figure 2.20: GSC ticket detail (GSC, n.d.).....</b>	21
<b>Figure 2.21: GSC booking process by mobile phone (GSC, n.d.) .....</b>	22
<b>Figure 2.22: GSC booking process by mobile phone(GSC, n.d.) .....</b>	22

<b>Figure 2.23: Red Bus seat selection (RedBus, n.d.) .....</b>	24
<b>Figure 3.1: Agile model flow (What is agile software development?, 2023) .....</b>	29
<b>Figure 3.2: Fyp 1&amp;2 report.....</b>	31
<b>Figure 3.3: Objectives and deliverables during Fyp 2 .....</b>	31
<b>Figure 3.4: Flutter (Flutter, n.d.) .....</b>	33
<b>Figure 3.5: VS code (Visual studio code, 2021).....</b>	36
<b>Figure 3.6: Android Studio (Android Mobile App Developer Tools, n.d.) .....</b>	36
<b>Figure 3.7: Architectural layers (Flutter Architectural Overview, n.d.) .....</b>	37
<b>Figure 4.1: Use case diagram.....</b>	41
<b>Figure 4.2:Context Diagram.....</b>	42
<b>Figure 4.3: Data Flow Diagram (level 0) .....</b>	43
<b>Figure 4.4: Flow chart of Smart Desk Booking System .....</b>	44
<b>Figure 4.5: Activity diagram for the seat allocation process .....</b>	45
<b>Figure 4.6: Flow chart for the time privilege process .....</b>	46
<b>Figure 4.7: Entity relationship diagram .....</b>	47
<b>Figure 4.8: Entity relationship diagram .....</b>	48
<b>Figure 4.9: Admin login page .....</b>	50
<b>Figure 4.10: Admin dashboard page .....</b>	51
<b>Figure 4.11: Employee page.....</b>	51
<b>Figure 4.12: Book desk page.....</b>	52
<b>Figure 4.13: Calendar pop up.....</b>	52
<b>Figure 4.14: Booking (view by list) page .....</b>	53
<b>Figure 4.15: Booking (view by map) page .....</b>	54
<b>Figure 4.16: Booking (view by map) page showing the booking desk location..</b>	54

<b>Figure 4.17: Employee scan QR code page .....</b>	55
<b>Figure 4.18: Employee booking page.....</b>	56
<b>Figure 4.19: Employee booking list page.....</b>	57
<b>Figure 4.20: Employee desk map page .....</b>	58
<b>Figure 5.1: System architecture .....</b>	59
<b>Figure 5.2: API set up in Firebase Function .....</b>	60
<b>Figure 5.3: Initial Firebase setting .....</b>	61
<b>Figure 5.4: Express Setup in JS file .....</b>	61
<b>Figure 5.5: Login page in website .....</b>	63
<b>Figure 5.6: Flowchart to login and book in the website.....</b>	64
<b>Figure 5.7: Admin dashboard page in website .....</b>	65
<b>Figure 5.8: Admin dashboard calendar in website.....</b>	66
<b>Figure 5.9: Admin generate report in website .....</b>	66
<b>Figure 5.10: Admin report download success in website .....</b>	67
<b>Figure 5.11: Admin report download success in website .....</b>	68
<b>Figure 5.12: Admin view booking list page in website .....</b>	69
<b>Figure 5.13: Admin add booking page in website .....</b>	69
<b>Figure 5.14: Admin add booking of show available employee in website .....</b>	70
<b>Figure 5.15: Admin view by map page in website .....</b>	70
<b>Figure 5.16: Admin view by map of desk detail in website .....</b>	71
<b>Figure 5.17: Admin employee page in website.....</b>	71
<b>Figure 5.18: Admin map management page in website .....</b>	72
<b>Figure 5.19: Admin block seat in map management page in website .....</b>	72
<b>Figure 5.20: Admin add desk page in website .....</b>	73

<b>Figure 5.21: Admin search function on map management page in website.....</b>	73
<b>Figure 5.22: Flow chart for Employee QR Login process .....</b>	74
<b>Figure 5.23: Admin add desk on map function in website .....</b>	75
<b>Figure 5.24: Admin add or swap desk on map function in website .....</b>	75
<b>Figure 5.25: Admin drag to change seat location in website .....</b>	76
<b>Figure 5.26: Admin successful to change seat location in website .....</b>	76
<b>Figure 5.27: Admin profile setting in website .....</b>	77
<b>Figure 5.28: Employee home screen in website .....</b>	78
<b>Figure 5.29: Employee successful booked desk in website .....</b>	78
<b>Figure 5.30: Employee successful check-in desk in website .....</b>	79
<b>Figure 5.31: Employee seat map screen in website .....</b>	79
<b>Figure 5.32: Employee change seat confirmation in website.....</b>	80
<b>Figure 5.33: Employee successful change seat in website .....</b>	80
<b>Figure 5.34: Employee setting profile page in website.....</b>	81
<b>Figure 5.35: Employee login page in app .....</b>	82
<b>Figure 5.36: Employee login process in app.....</b>	83
<b>Figure 5.37: Employee home page and booking process in app.....</b>	84
<b>Figure 5.38: Employee view seat map process in app .....</b>	85
<b>Figure 5.39: Employee checkin process in app .....</b>	86
<b>Figure 5.40: Employee check in successful process in app .....</b>	87
<b>Figure 5.41: Employee profile page in app .....</b>	88
<b>Figure 6.1: Login page with empty fields on website .....</b>	92
<b>Figure 6.2: Login page with wrong email format on website .....</b>	92
<b>Figure 6.3: Login page with empty field on mobile .....</b>	93

<b>Figure 6.4: Add booking page with empty field on website.....</b>	94
<b>Figure 6.5: Add booking page with one empty field on website.....</b>	94
<b>Figure 6.6: Added a booking four times on the website for the same booking date .....</b>	95
<b>Figure 6.7: Added a booking four times on the website for the same booking date .....</b>	96
<b>Figure 6.8: Delete confirmation alert on the website .....</b>	97
<b>Figure 6.9: Deleted successful on the website .....</b>	97
<b>Figure 6.10: Edit booking validation testing on the website.....</b>	98
<b>Figure 6.11: Edit booking save the same value on the website.....</b>	98
<b>Figure 6.12: Date range calendar for report on the website.....</b>	99
<b>Figure 6.13: Part of the downloaded report content .....</b>	99
<b>Figure 6.14: Map management alert cannot remove column.....</b>	100
<b>Figure 6.15: Map management alert cannot remove row.....</b>	100
<b>Figure 6.16: Selected the seat that under repair .....</b>	101
<b>Figure 6.17: Selected the seat that booked by others .....</b>	101
<b>Figure 6.18: QR API authentication test by Postman .....</b>	102
<b>Figure 6.19: Scan invalid QR code test.....</b>	102
<b>Figure 6.20: Scan expired QR code test.....</b>	103
<b>Figure 6.21: Machine learning model Hyper tuning .....</b>	104
<b>Figure 6.22: The result of the prediction .....</b>	104
<b>Figure 6.23: Flow chart of the random prediction .....</b>	105
<b>Figure 6.24: Desk return from the random predict by Postman .....</b>	106
<b>Figure 6.25: View booking on the map .....</b>	107
<b>Figure 6.26: Change desk location successfully .....</b>	107

<b>Figure 6.27: Updated desk location .....</b>	108
<b>Figure 6.28: Code for selecting available desks in system .....</b>	108
<b>Figure 6.30: Booking calendar on admin module.....</b>	109
<b>Figure 6.31: Application Login process via email password .....</b>	110
<b>Figure 6.32: Application Login process via QR code .....</b>	111
<b>Figure 6.33: Received QR code by email.....</b>	111

## **LIST OF ABBREVIATIONS/ SYMBOLS**

CSS	Cascading Style Sheets
DT	Decision Tree
FIST	Faculty of Information Science and Technology
FYP	Final Year Project
GB	Gradient Boosting
GSC	Golden Screen Cinemas
HTML	Hypertext Markup Language
KNN	K-Nearest Neighbors
LR	Logistic Regression
MMU	Multimedia University
NPM	Node Package Manager
PHP	Hypertext Preprocessor
QOE	Quality of Experience
QR Code	Quick-response code
RF	Random Forest
SDLC	Software Development Lifecycle
SVM	Support Vector Machine
VS Code	Visual Studio Code

## **LIST OF APPENDICES**

<b>Appendix A: Meeting Log.....</b>	118
-------------------------------------	-----

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Overview**

The Smart Desk Booking System represents a forward-thinking solution, tailored to meet the demands of modern office spaces. In an environment where adaptability and flexibility take central stage, the project aims to change the traditional model of fixed desk allocation. The system acknowledges the growing importance of ensuring safe and efficient workspaces, particularly in the post-pandemic era.

In this project, the workspace will be divided into workstations and Touch & Go stations. The Touch & Go workstation will be a quick-change and non-permanent workstation, allowing employees to focus more on their work. One of the main features of the project is the random allocation system, which provides employee with a dynamic workspace allocation model. The system will randomly assign desk seats to reservations, but when a booking is successfully made, it also allows employee to select their preferred location based on individual work requirements.

The attractive interface with built-in seating maps makes desk booking easier. Meanwhile, administrators can control and reserve desks for themselves or others at the same time to reduce duplicate work. In addition to these features, the system incorporates a QR code verification login process for the employee. Once the employee need to login the application, the employee must request the QR code from the system. While the system will send out the email with the QR code to employee.

The centralized system demonstrates the power of synchronizing the intended deliverables and enables the Smart Desk Booking System to showcase its capabilities, ready to redefine office management in the evolving modern workspace. It combines the flexibility of desk allocation with the security of QR code verification to provide a comprehensive solution for the changing needs of today's office environments.

## **1.2 Problem Statement**

### **1.2.1 Lack of hybrid platform**

The existing gap in the market is the absence of hybrid desk booking tools that integrated to the various needs of employees in both office-based and remote work settings. Current solutions typically focus on either in-office booking or entirely remote work management, leaving employees without a flexible platform to seamlessly switch between these work modes as needed. With the hybrid platform, it allows employees to book desks using their mobile phones for added convenience, while administrators can efficiently manage bookings through the website to create a seamless and employee-centric experience.

### **1.2.2 Limited Booking Flexibility for Administrators**

Traditional office management systems often lack features that enable administrators to effectively control and allocate desk space over time. The current seat reservation system is not necessary to provide smooth long-term bookings and optimal utilization of office resources. Administrators face a lack of tools to streamline the long-term assignment process. Therefore, solving this issue is critical to enhance workspace management strategies and provide administrators with the necessary tools to improve efficient desktop space allocation.

### **1.2.3 Inadequate Registration Verification**

The current registration process lacks the robust security required to safeguard sensitive data and ensure employee authentication. Traditional methods fall short of providing real-time verification, posing potential risks to the integrity of the system. Addressing this, the project aims to implement a real-time QR code verification system during registration, ensuring a swift, secure, and reliable method to authenticate employees and access the Smart Desk Booking System.

### **1.3 Project Objectives**

These project objectives are aimed to create a comprehensive smart desk booking system that meets the diverse needs of employees, administrators, and achieves the overall goal of promoting efficient, secure, and adaptable office space management. The detail is mentioned as below:

#### **To provide random seat allocation management via web and mobile applications.**

➤ The seat booking process can be simplified through a hybrid model of web and app. This approach is intended to replace current manual methods that are inefficient, impact space utilization, or overcrowd. Employees or administrators can easily book and reserve seats to manage their workspace conveniently and effortlessly. The streamlined process not only simplifies seat booking but also improves the overall efficiency of the system.

#### **To provide flexibility for administrators to book desks for longer periods of time beyond a month.**

➤ Admins managing seats for multiple employees may accidentally create duplicate reservations through repetitive processes. To address this un flexibility, implementing a system with real-time tracking and synchronization such as displaying a seat map to provide an overview of bookings status. This objective also aims to empower admins to strategically manage desk allocations for extended periods beyond one month, ensuring efficient and error-free seat management.

#### **To verify registration with QR code verification.**

➤ To ensure the registration process is safe and efficient, the project introduces QR code verification. This method aims to enhance security by verifying employee registration in real-time via QR codes. The immediacy of this verification process not only enhances the system's ability to protect against unauthorized access, but also provides employees with a fast and seamless onboarding experience. The incorporation of QR code verification adds an extra layer of protection, protecting sensitive information and ensuring a robust authentication process.

## **1.4 Project Scope**

The project's scope is mainly around the development and implementation of a smart desk booking system which designed to ingratiate to the evolving requirements of modern office spaces. The primary components of this project encompass the creation of a hybrid web and mobile application interface, providing employees and administrators with the tools to efficiently manage desk reservations.

The project focuses on fulfil the specific company needs. For take notice into the target audience, the project is designed to serve the requirements and functions relevant to the operations of that particular organization. The target employees of this system include administrators and ordinary employees within the company.

Administrators and employees, each plays an important role in effectively utilizing the systems developed within the company. Administrators have higher authority and are the key supervisors responsible for system management and configuration. The tasks include setting administrative controls, managing employee accounts, and ensuring the overall functionality of the platform. For employees, the only function they can do is to reserve seats and view the seat details.

## **1.5 Report Organisation**

Chapter 1 introduces and provides a comprehensive overview of the system proposed in this project. Part of this chapter is an outlining problem statement, highlighting the specific problem of the project aims to solve, followed by a clearly defined objective for describing the precise goals the research seeks to achieve. Additionally, it defines the scope of the project and clarifies the boundaries and the expected output which this project will operate.

Chapter 2 conducts an extensive literature review by reviewing relevant papers and studies that informed this project. This section not only reviews the existing literature but also provides a critical analysis of the overall area where the proposed project can contribute new insights. Furthermore, this chapter explores the background of the project, providing historical context and laying the groundwork for the chosen

research direction. A comparative analysis of existing systems is also included, providing a basis for understanding how the proposed project differentiates itself in the field.

Chapter 3 discusses the selected software development life cycle (SDLC) model and justifies its choice based on the specific requirements of the project. Future planning is explored, outlining expected milestones and timelines that may occur during the development process. Additionally, this chapter details the tool technology, programming languages, and frameworks chosen for implementation and provides justification for each choice.

Chapter 4 shifts the focus to the design phase by presenting a comprehensive list of diagrams that necessary to understand the proposed system architecture. These diagrams include use cases, context diagrams, data flow diagrams, flow chart, etc., which together provide a visual representation of the system structure and functionality. This chapter also describes an initial prototype of the system.

Chapter 5 provides a detailed implementation of the system, presenting its architecture and outlining the final output of the system interface. Screenshots and descriptions of the main interfaces are included to clearly demonstrate the user experience. This section explores how the system achieved its initial goals, showing changes and adjustments made during development.

Chapter 6 is the test evaluation, which will test each scenario that the system roles will face to show that it can also handle the validation. Detailed test cases and scenarios are documented showing the responses and adjustments of the system to ensure the system stable. The case studies will also be discussed to finally determine how the proposed system achieves the goals. Limitations will also be mentioned then.

Chapter 7 summary the conclusion of the main findings and contributions of this project. It reiterates the goals of the project and highlights how it will be achieved through the systematic approach outlined in the previous chapters. This section also

outlines potential areas for future work and discusses possible enhancements, additional features, and new capabilities that could be integrated into the system.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 The smart desk system**

The integration of smart desks with desk booking systems represents a key development in modern office furniture and workplace management. This innovation aligns with the broader development of smart office solutions, where technology and furniture design converge to enhance employee experience and optimize workspaces (Aryal, A et al., 2019). Equipped with features such as digital interfaces, wireless charging and health tracking capabilities, smart desks have become integral to creating dynamic and employee-centric office environments (Zhang et al., 2022).

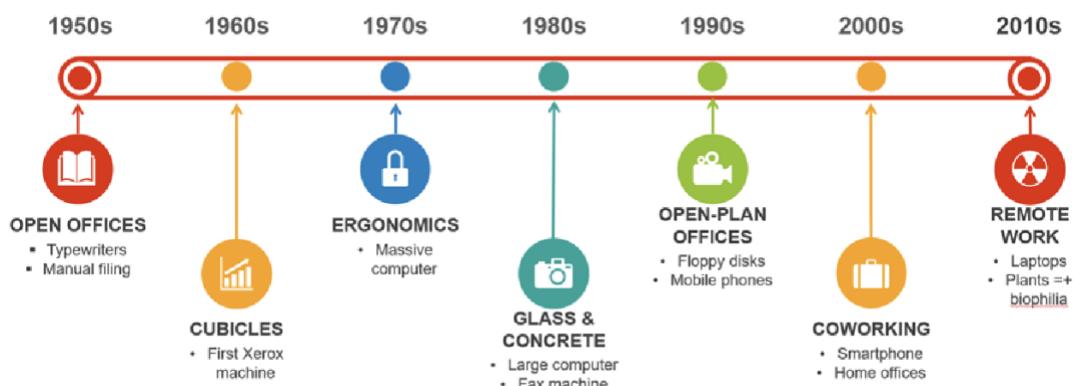
The phenomenon of hybrid work environments where employees can work remotely has gained significant traction in the contemporary corporate environment which makes it a new way called coworking (Fan, Y., & Jin, Y., 2020). In response to this shift, desk reservation systems have become important tools, enabling employees to reserve desks in advance and facilitating a seamless transition between remote and in-office work. The integration of intelligent functions in these systems has become a necessity for efficient management. A key feature is a graphical map that incorporates seating arrangements to provide employees with a visual representation of the office space.

The addition of a desk booking system further empowers employees, allowing them to choose a seat based on personal preference or the collaborative nature of the job. This flexibility not only helps create a more adaptable and efficient workspace, but also addresses contemporary considerations such as the need for social distancing protocols (Richardson, L. 2021). As offices continue to transform, the interconnection between the development of coworking spaces and the implementation of smart desk booking systems is clear (The History of Coworking, n.d.).

### 2.1.1 Evolution of smart Office Concept

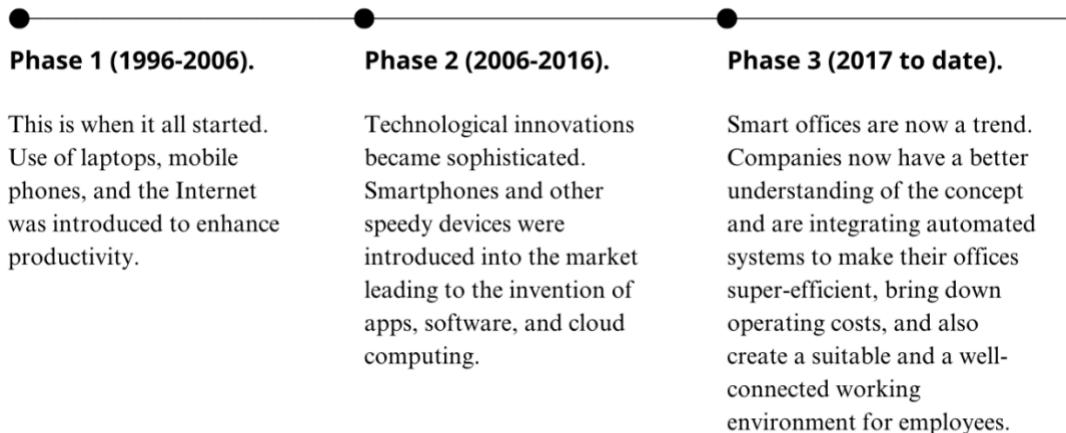
The office has experienced a substantial evolution over the years, progressing from a conventional workspace to a smart office (Howell, T., 2022). When discussing smart desks, it is inevitable to discuss the evolution of smart offices, as both innovations share common roots. The evolution of smart desks and smart offices are intertwined and evolving together, redefining the modern workplace. A smart office can be defined as a fully equipped office equipped with IoT devices such as various sensors connected to the Internet (Shinde et al., 2020). Figure 2.1 shows that since the 1950s, the office evaluation has changed in terms of technology adoption, workspace design, and employee well-being every 10 years. (Chen, 2020).

Office design concepts between 1950 and 1980 also explored open office layouts, modular furniture systems, and integrating technology into the workplace. The aim is to create a more flexible and adaptable work environment that promotes collaboration, communication, and efficiency. Additionally, the emphasis on office evaluation during this period highlighted the need to collect and analyse data on office activities, characteristics, and outcomes to inform design decisions and improve overall work performance.



**Figure 2.1: Evaluation of an office (Halim, N, 2021)**

The assessment of smart offices commenced in 1996 during the initial phase, marked by the introduction of technologies like mobile phones and the internet as shown in Figure 2.2. Progressing into the third phase until 2017, this evolution involved the integration of automated systems to enhance office efficiency, cut operating costs, and establish a conducive and interconnected workspace for employees. With the development of smart offices, the emergence of smart desks has become the focus of the third phase. Equipped with innovative features and connectivity, these desks represent the culmination of a comprehensive shift toward smarter and more adaptable office environments.



**Figure 2.2: Timeline phases of smart office (Hartman, V.,2023).**

### **2.1.2 Smart Desk Environment in Malaysia**

As per Malaysian standard regulations, employees usually work 48 hours per week. The results of the Malaysia's Healthiest Workplaces Survey 2019 reveal a worrying statistic, indicating that 51% of employees suffer from work-related stress. This highlights the importance of establishing an efficient office environment. For employees who spend most of their day in the office, creating an environment that is productive, enjoyable, relaxing and inviting is crucial (Halim, N,2021). Recognizing the common challenges of maintaining a healthy work-life balance, organizations must prioritize creating workspaces that are conducive to employee well-being and productivity.

As the world progress into a new era, the integration of smart solutions becomes increasingly essential, particularly in the workplace where employees dedicate a significant part of their time (Leong, Y. C., & Sharif, R., 2020). With a specific focus on smart desk systems, the design must evolve to meet the demands of this changing landscape. The significance of Quality of Experience (QoE) in smart space design is increasingly determines the future directions of design. (J.Zeng, et al.,2015). As employees make heavy use of some of these innovative office solutions, it is crucial to prioritize enhanced QOE to ensure that smart desk systems are not only technologically advanced but also seamlessly customized to the specific needs and preferences of Malaysian employees.

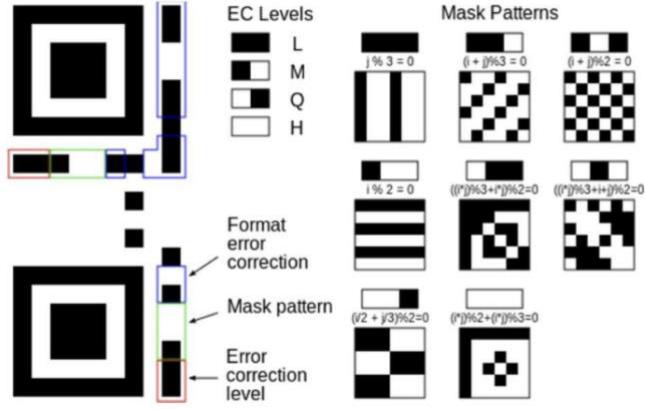
## **2.2 QR Code**

### **2.2.1 History of QR Code**

QR codes (quick response codes) have become a ubiquitous part of the modern world and have found their way into a wide range of applications and industries. The story of the QR code began in Japan in the mid-1990s, when it was developed by Denso Wave, a Toyota subsidiary (De Seta,2023). The main motivation behind their creation was to improve the efficiency of tracking auto parts during the manufacturing process.

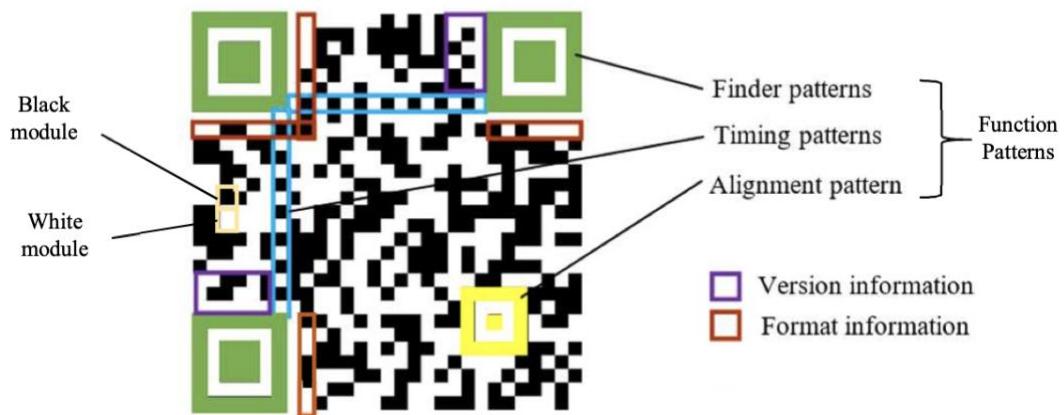
### **2.2.2 QR code verification**

By using QR codes, the authentication process becomes more streamlined, providing a convenient solution and contributing to the overall security environment of mobile-based transactions (Chow et al., 2016). There is many key structures of QR codes such as the version, data capacity, error correction levels, structure element, mask pattern, format information and quiet zone as shown as figure 2.3 (Fajar, M. et al., 2023). The specific structure used must depends on factors such as the amount of data to be stored and the level of error correction required. QR codes are designed to be versatile and adaptable, making them suitable for a wide range of applications across various industries.



**Figure 2.3: QR Information’s Format (Fajar et al., 2023)**

A QR code is a two-dimensional barcode that can be store data information then read by specialized QR reading devices such as mobile phone build in camera (Escobedo, P., et al., 2023). Humans may not feel much about the black and white modules shown in Figure 2.4 due to human perception, but it is an essential and crucial element for the decoding task (Ho, W. E et al., 2022) .The QR codes can be detected through functional modes and allow high-quality scanning in various positions to sustain the real world (H. Su et al., 2020). Therefore, due to the important role of functional patterns in QR codes, functional patterns should be remained when generating hybrid QR code blends.



**Figure 2.4: QR basic structure (Ho, W. E et al., 2022)**

### 2.2.3 Existing Systems

In fact, desk booking systems are not widely available off-the-shelf in the market and it usually require for customization. Due to this limitation, it is challenging to find existing systems. As an alternative, references and comparisons would be made by similar systems such as cinema seat selecting systems to represent similar functionality.

### 2.2.4 LiquidSpace

LiquidSpace is a platform designed to facilitate flexible workspace solutions (LiquidSpace, 2017). The site serves as a marketplace that connects individuals and businesses through flexible office space, meeting rooms and coworking spaces. Users can explore a variety of workspaces on demand or through membership to suit different needs and preferences. LiquidSpace also enables users to book or search nearby workspaces. Figure 2.5 show the interface which displays some detailed information about the workspace, including descriptions and images.

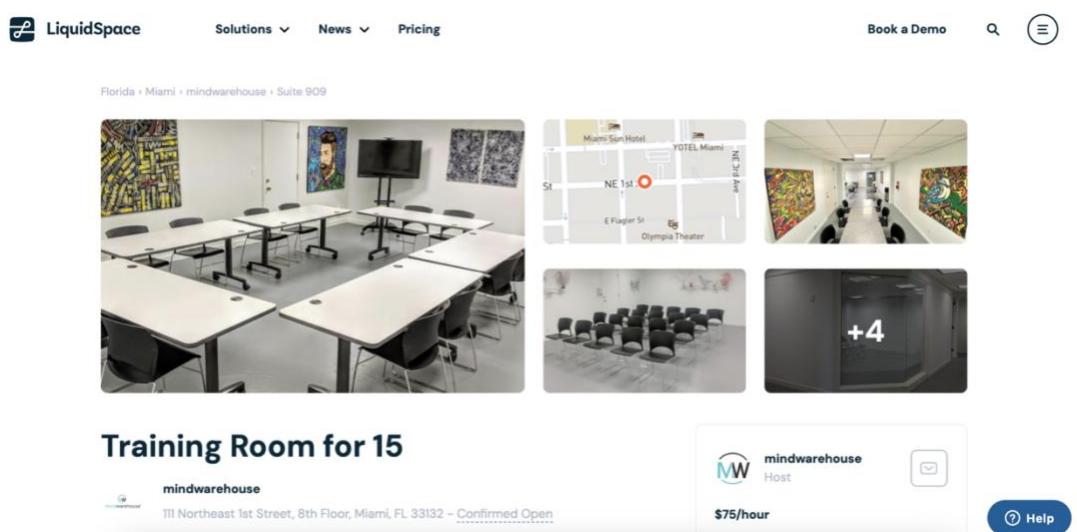
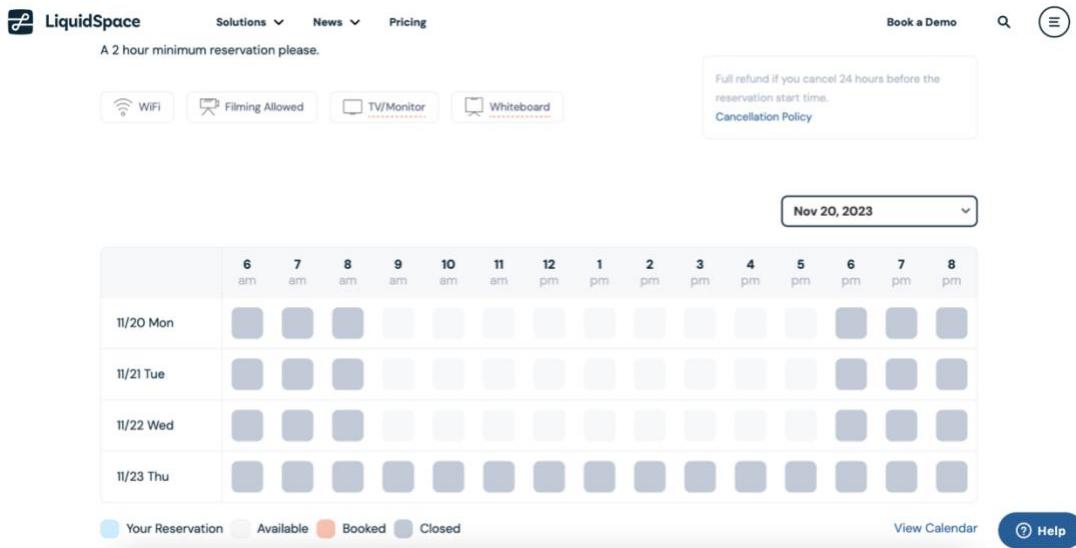


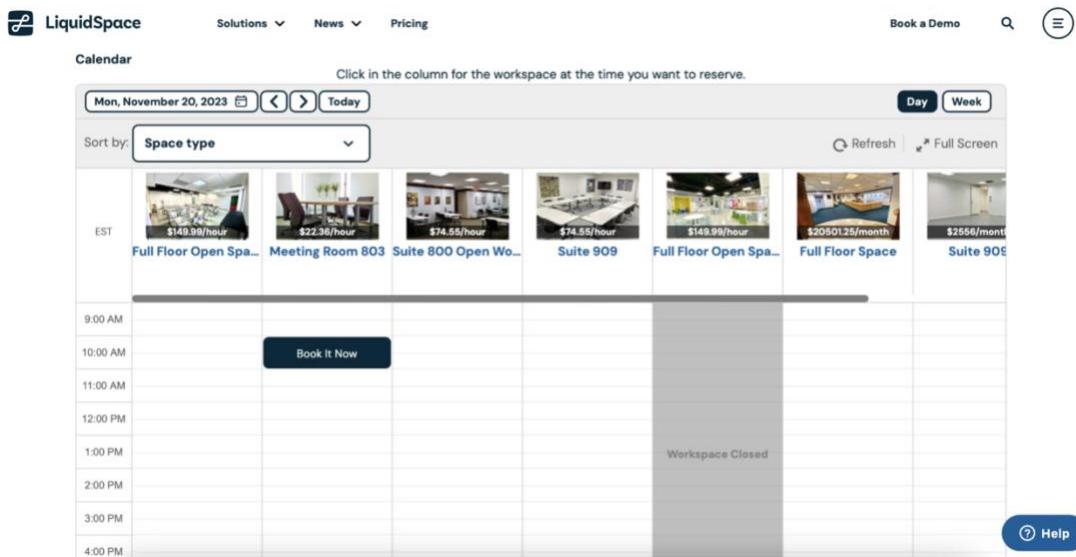
Figure 2.5: LiquidSpace booking page (LiquidSpace, 2017)

Figure 2.6 is a continuation of the previous Figure 2.5, showing the bottom interface. It provides a table which presenting the dates available for booking.



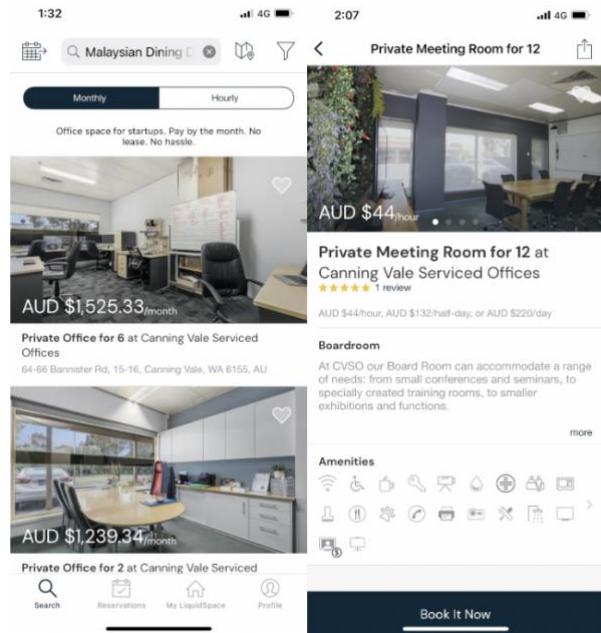
**Figure 2.6: LiquidSpace date available selection (LiquidSpace, 2017)**

While users click on the "View Calendar" option in the bottom right corner of Figure 2.6, it will redirect the users to a new page show in figure 2.7. This page allows users to explore various workspaces and make bookings for specific time slots. On the upper left side, users can choose the option to sort spaces by different types and select specific dates to enhance flexibility in the search and booking process.

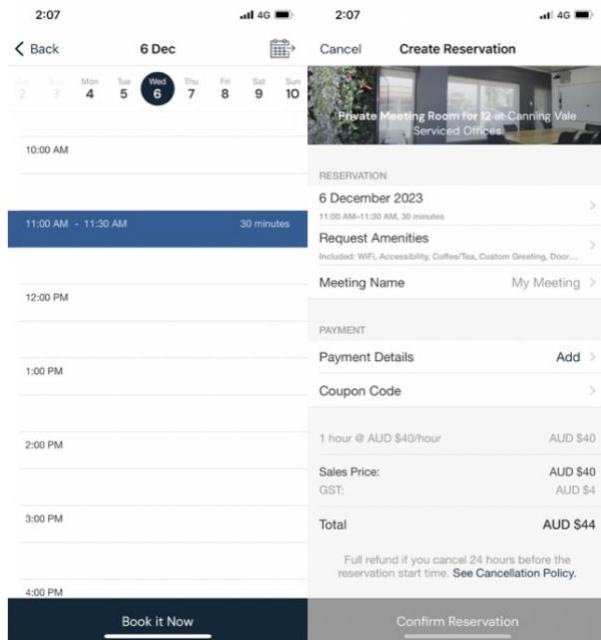


**Figure 2.7: LiquidSpace view calendar page (LiquidSpace, 2017)**

Figures 2.8 and 2.9 show the LiquidSpace application, which has similar Web functionality and flow. During the booking process, the app changes the room list display to vertical for consistent with the size of the mobile screen.



**Figure 2.8: LiquidSpace application booking process (LiquidSpace, 2017)**



**Figure 2.9: LiquidSpace application booking confirmation (LiquidSpace, 2017)**

Functionalities:

- a. Flexible booking: LiquidSpace offers a flexible booking system that allows users to find and book workspaces, conference rooms or private offices on a short-term basis.
- b. Diverse workspace options: Users can explore multiple workspace options based on their specific needs, including coworking spaces, private offices, and conference rooms.
- c. Real-time Availability: The platform provides real-time availability updates, ensuring users get accurate information about workspace availability and can book in a timely manner.
- d. Mobile App Integration: LiquidSpace is accessible via a mobile app, allowing users to book workspaces and receive notifications about bookings from anywhere.

Limitations:

- a. Limited Locations: The availability of workspaces on LiquidSpace may be limited to specific regions or cities, restricting access for users in other areas.
- b. Workspace Amenities: The range of amenities in listed workspaces can vary, and users might find limitations in terms of facilities provided by certain spaces.
- c. Lack of recommendations: Unable to directly target employee region to give specific area booking suggestion.
- d. Link Security Issues: When users perform searches on LiquidSpace, the platform may expose the search query in the link, as shown in Figure 2.10 potentially revealing sensitive information.



**Figure 2.10: LiquidSpace search query (LiquidSpace, 2017)**

## 2.2.5 Racquetvenues

Racquetvenues is an online service dedicated to booking badminton court which allowing user to book courts conveniently through the platform (Racquetvenues, n.d). Users can browse available courses through the website, select a specific date and time to complete the reservation process as shown in Figure 2.11.

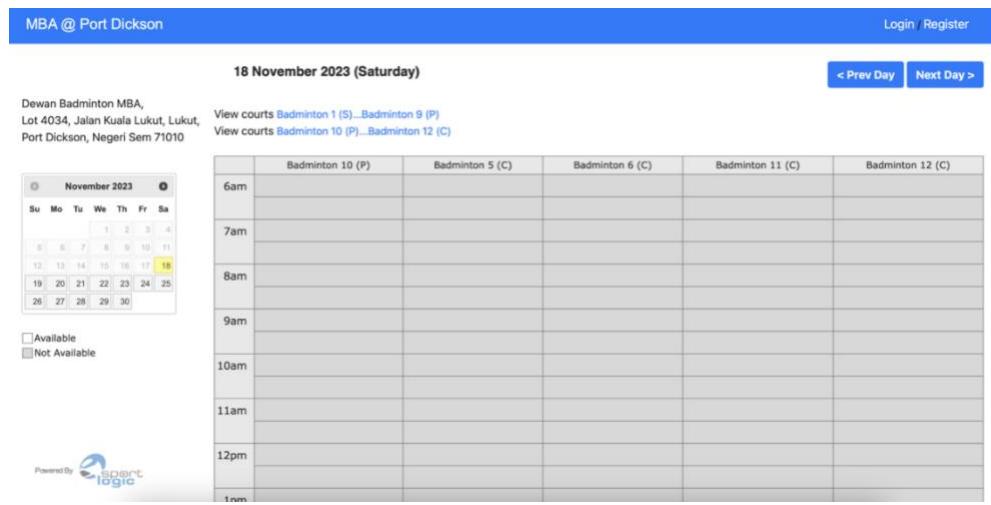
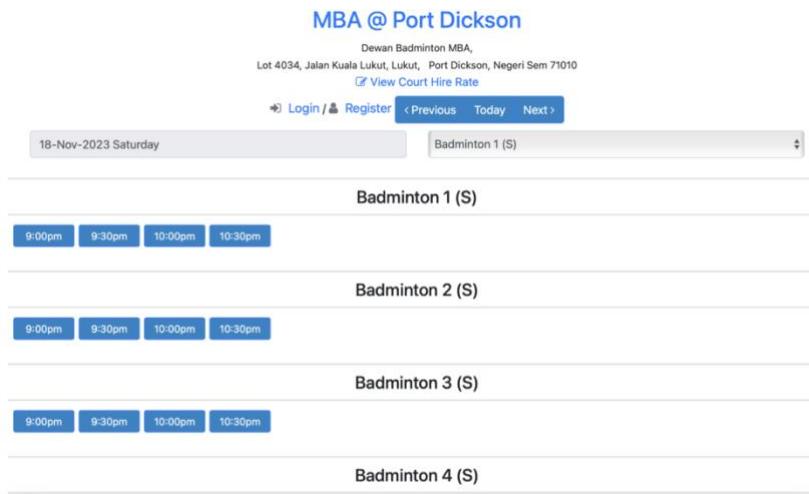


Figure 2.11: Racquetvenues booking page (Racquetvenues, n.d.)

Users also have the option to view detailed information about other available facilities and potentially access other features related to managing reservations. Figure 2.12 is a bottom view of Figure 2.11, by clicking 'Mobile View' it will retrieve the page layout as a mobile-friendly version as shown as Figure 2.13.



Figure 2.12: Racquetvenues bottom of booking (Racquetvenues, n.d.)



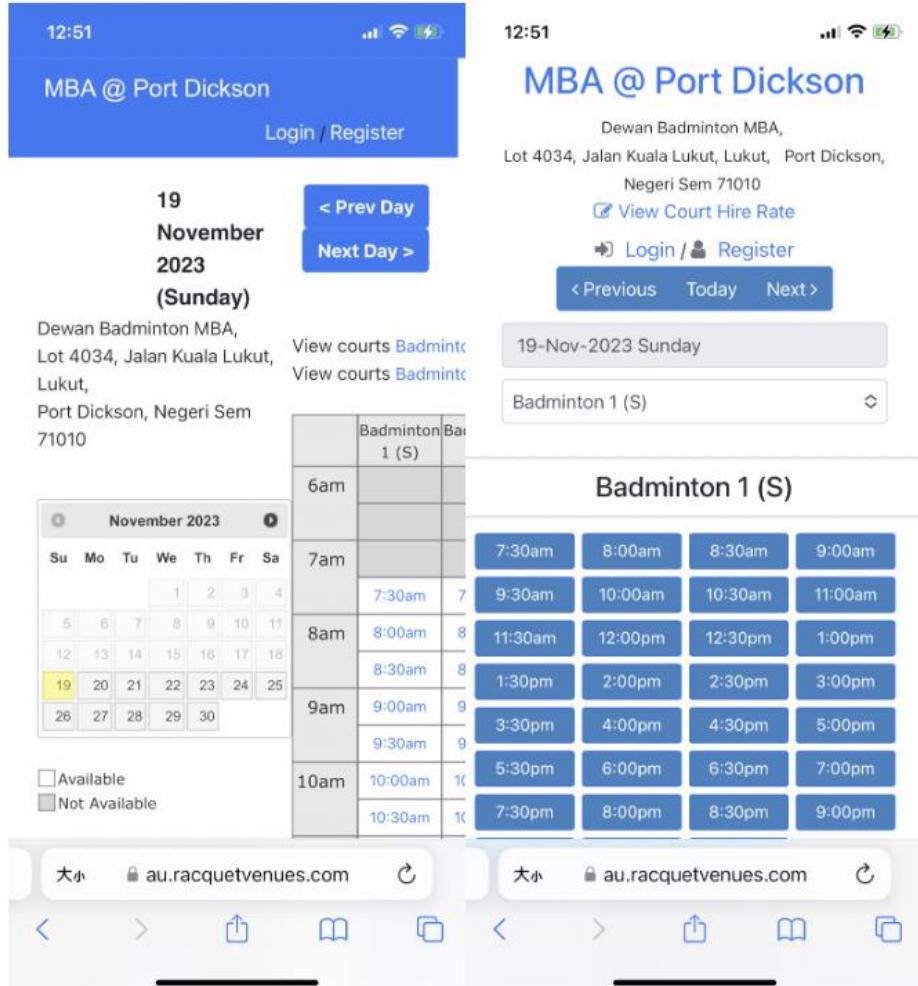
**Figure 2.13:** Racquetvenues mobile view on website (Racquetvenues, n.d.)

When the employee wants to make an appointment, by clicking on the desired time slot in the schedule, the appointment details will be displayed in Figure 2.14. Users can select the duration of the appointment via a drop-down list.

The booking detail form is for 'Badminton 2 (P) - MBA @ Bukit Puchong'. It includes fields for 'From' (8:30am Monday 4 Dec 2023) and 'To'. The 'To' field is a dropdown menu containing a list of time slots and their durations, starting with '9:30am (1 hour)' and ending with '8:30pm (12 hours)'. Other options include 10:00am (1.5 hours), 10:30am (2 hours), 11:00am (2.5 hours), 11:30am (3 hours), 12:00pm (3.5 hours), 12:30pm (4 hours), 1:00pm (4.5 hours), 1:30pm (5 hours), 2:00pm (5.5 hours), 2:30pm (6 hours), 3:00pm (6.5 hours), 3:30pm (7 hours), 4:00pm (7.5 hours), 4:30pm (8 hours), 5:00pm (8.5 hours), 5:30pm (9 hours), 6:00pm (9.5 hours), 6:30pm (10 hours), 7:00pm (10.5 hours), 7:30pm (11 hours), 8:00pm (11.5 hours), and 8:30pm (12 hours). The form also includes fields for 'Name', 'Contact Number', 'Email', 'Confirm Email', 'Membership Number (if any)', 'Promo Code (if any)', and 'Credit Voucher Number (if any)'. The entire form is presented on a white background with black text and blue highlights for required fields.

**Figure 2.14:** Racquetvenues booking detail (Racquetvenues, n.d.)

Figure 2.15 displays the mobile phone view, with the desktop view on the left-hand side and the mobile view on the right-hand side. When users access the website on a mobile phone, the layout may appear distorted. However, after clicking the 'Mobile View' button, the layout undergoes beautification than before.



**Figure 2.15: Racquetvenues desktop view and mobile view on the mobile phone (Racquetvenues, n.d.)**

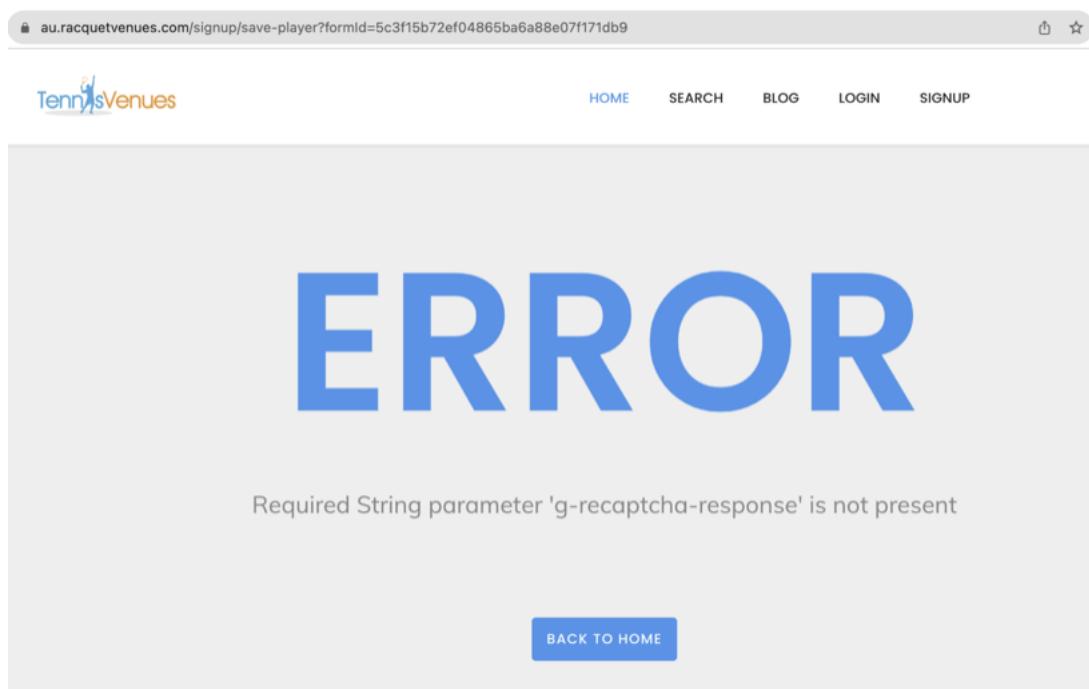
Functionalities:

- Venue Selection: Users can select specific venues, such as MBA Bukit Puchong badminton court, for booking sports or racquet facilities.
- Date and Time Selection: Users can choose their preferred date and time for the booking, offering flexibility based on their schedules.

- c. Reservation Confirmation: The platform likely generates a confirmation for successful reservations, providing users with booking details and relevant information.
- d. Employee Accounts: Users may have the option to create accounts, facilitating easier tracking of bookings, accessing booking history, and managing personal preferences.

Limitations:

- a. Limited Venue Selection: Venues available for booking may be limited to those listed on the platform and no photos are available. This limits employee choices based on location or facility preferences.
- b. Employee Interface: Depending on the design, the employee interface is too simple and appears cluttered when users open it on their phone.
- c. Unchangeable date: It may be difficult for users to change the time or date on the booking details page as the employee interface is not intuitive and comfortable enough.
- d. Technical Issues: Users may experience technical difficulties, such as website downtime or malfunctions, affecting their ability to book successfully. Figure 2.16 showing registration error issue.



**Figure 2.16: Racquetvenues registration error (Racquetvenues, n.d.)**

## 2.2.6 Golden Screen Cinemas (GSC)

Golden Screen Cinemas (GSC) is Malaysia's leading cinema chain, providing moviegoers with a comprehensive cinematic experience (Golden Screen Cinemas, n.d.). The GSC website often serves as an online platform for movie lovers to check showtimes, buy tickets, and explore upcoming movies. Users can also find information about movie theatre locations, promotions, and special events through the GSC website. Figure 2.157 shows the home page of GSC website, where users can choose the specific film. After selecting the film, the user can choose the date, location and showtime as shown in Figure 2.18.

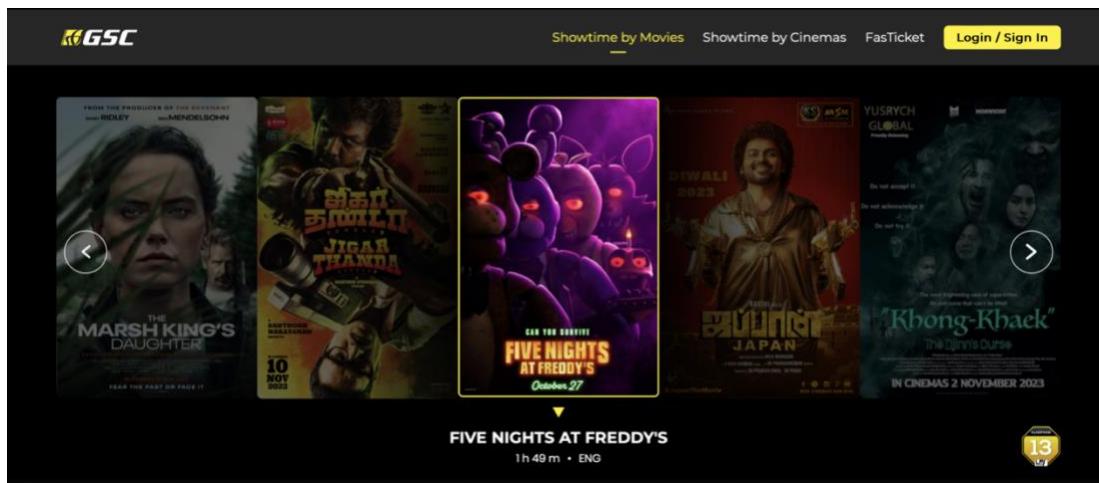


Figure 2.17: GSC Homepage (Golden Screen Cinemas, n.d)

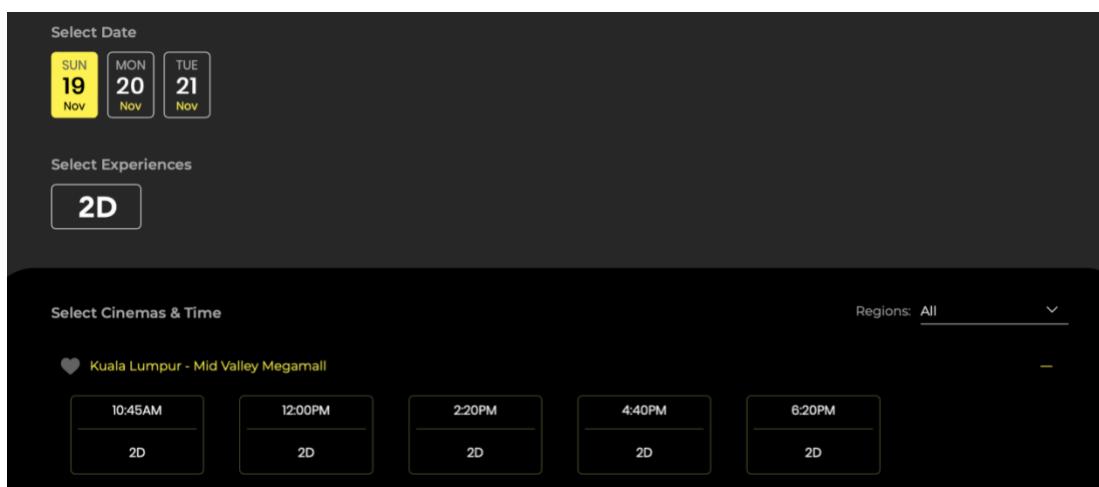


Figure 2.18: GSC date selection(Golden Screen Cinemas, n.d)

The employee will then select the seat via a graphical map as shown in Figure 2.19. The seat selection detail will appear in the bottom and the booking details will be displayed at the top. While employee done the payment, they can view their reservations details in the ‘My Ticket’ as shown in Figure 2.20.

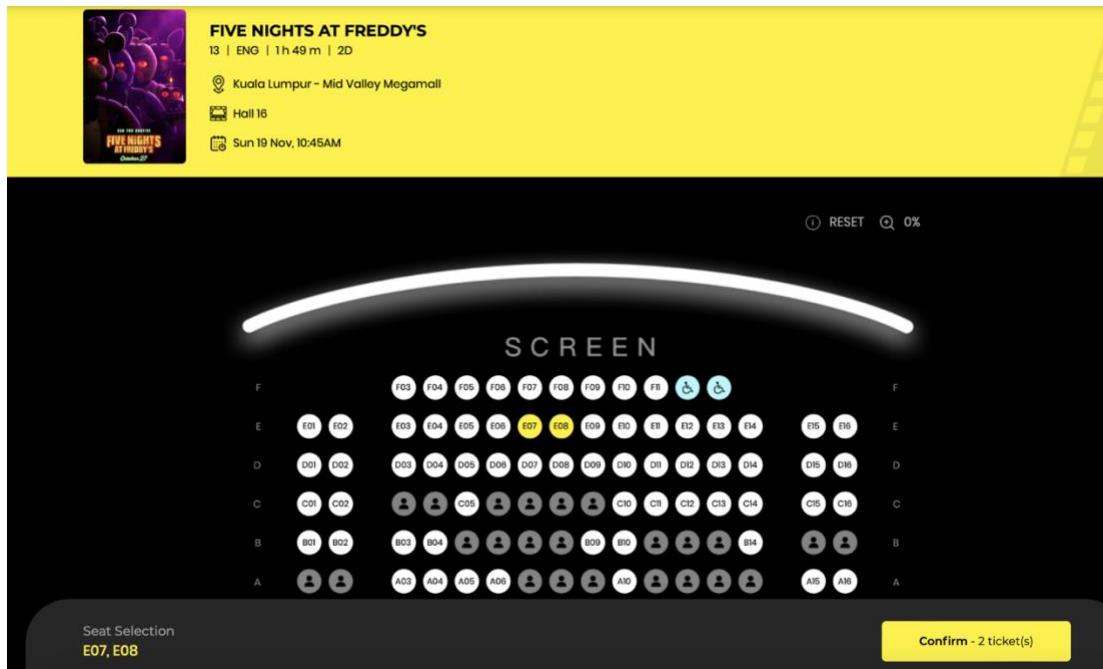


Figure 2.19: GSC seat selection (GSC, n.d.)

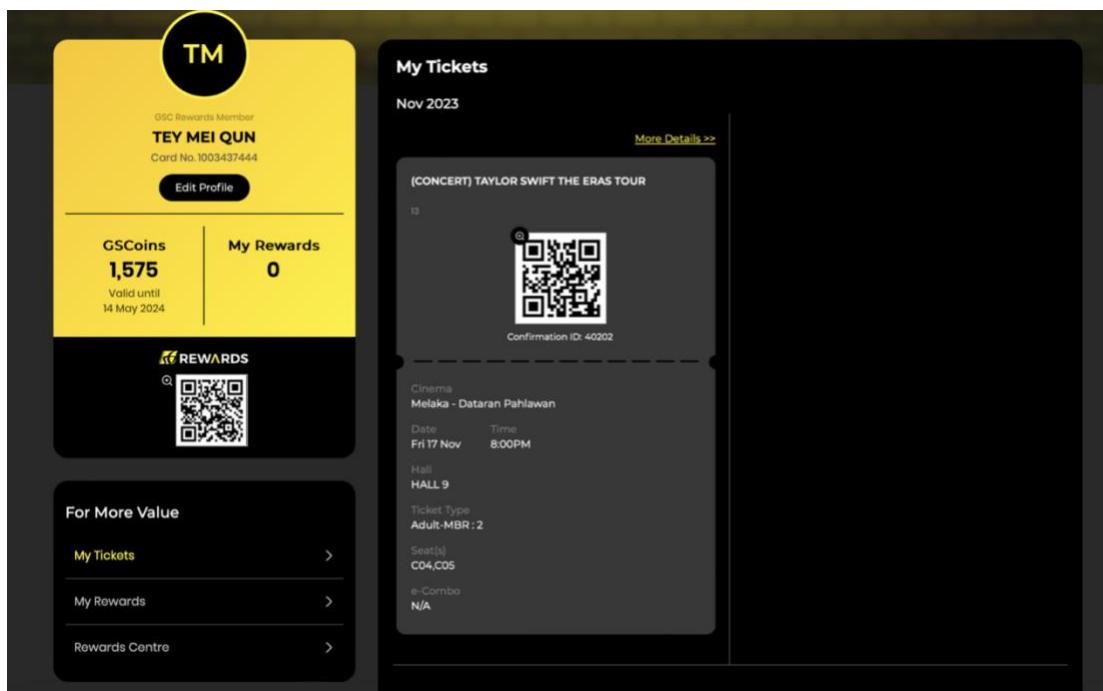


Figure 2.20: GSC ticket detail (GSC, n.d.)

GSC also offers a mobile application that 100% mirrors the functionality of its website. Figure 2.21 and Figure 2.22 illustrate the same prototype on both platforms.

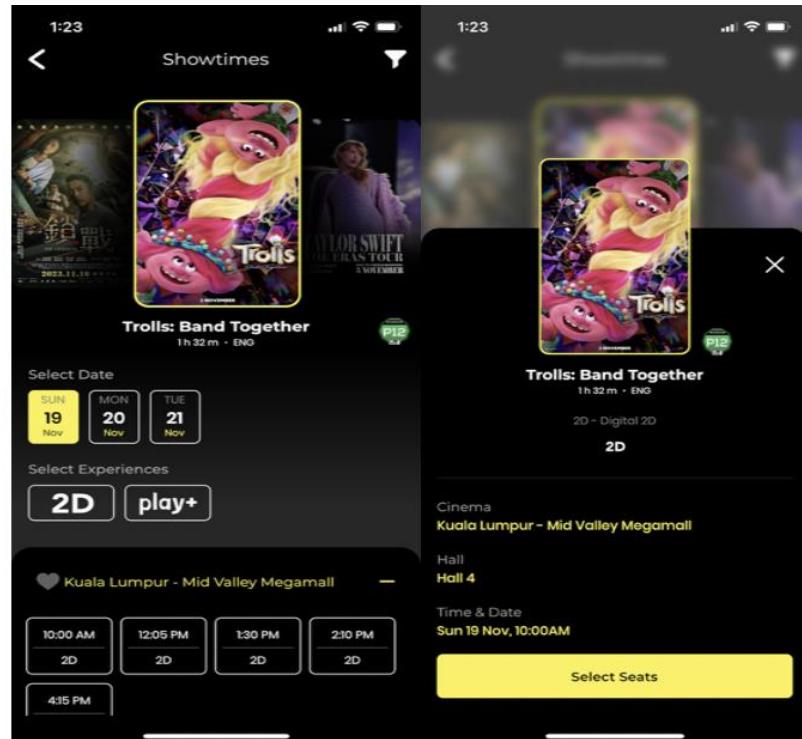


Figure 2.21: GSC booking process by mobile phone (GSC, n.d.)

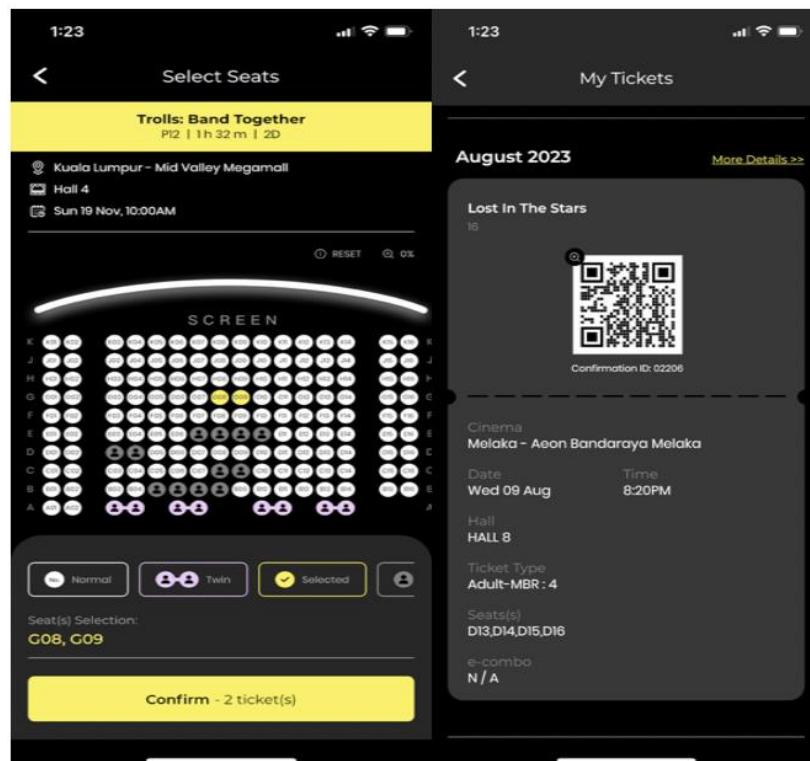


Figure 2.22: GSC booking process by mobile phone(GSC, n.d.)

**Functionalities:**

- a. Movie Selection: Users can browse and select a wide range of movies available at GSC Cinemas.
- b. Location-based searches: The Platform may offer location-based searches to enable users to find nearby GSC movie theaters based on their preferences.
- c. Showtime Information: GSC allows users to view the showtime schedule for each movie, including available time slots and cinema halls.
- d. Online booking: Users can book movie tickets online, providing a convenient and fast way to book seats in advance.
- e. Seat Selection: The Platform may provide a seat selection feature that allows users to select their preferred seats within the cinema lobby.

Golden Screen Cinemas (GSC) is Malaysia's leading cinema chain, providing

**Limitations:**

- a. Seat lock: If a reservation attempt fails, users may experience seat availability limitations as seats are temporarily locked once the employee initiates the reservation process. In this case, users may need to wait approximately 10 minutes before trying again to book previously unreleased seats.
- b. Lack of recommendation: Unable to retrieve the employee's location to provide location-based suggestions for booking.

## 2.2.7 Red Bus

Red Bus is an online platform for convenient bus ticket booking, providing travellers with a convenient and efficient way to plan and secure bus trips (RedBus, n.d.). Users can visit the Red Bus website or mobile app to search bus routes, view schedules, view available seats and book individual destinations. The platform often partners with different bus operators to provide a variety of routes and travel options.

Additionally, Red Bus offers features such as real-time tracking, seat selection, and secure payment options, thereby enhancing the overall experience for travellers seeking bus transportation services. Once the employee chooses the destination, date, and time, the page will look like Figure 2.23, displaying seat selection options following with a bus seat map.

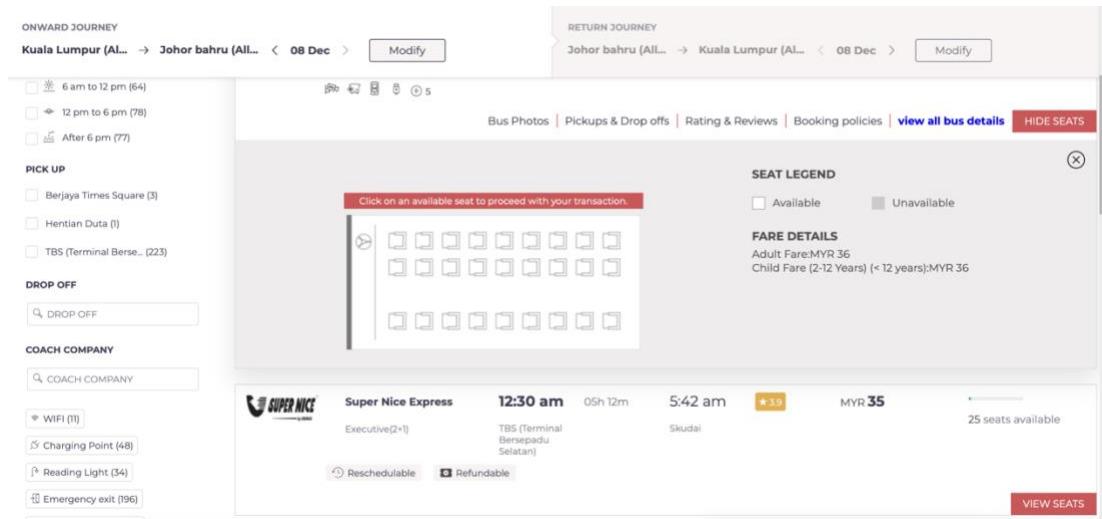
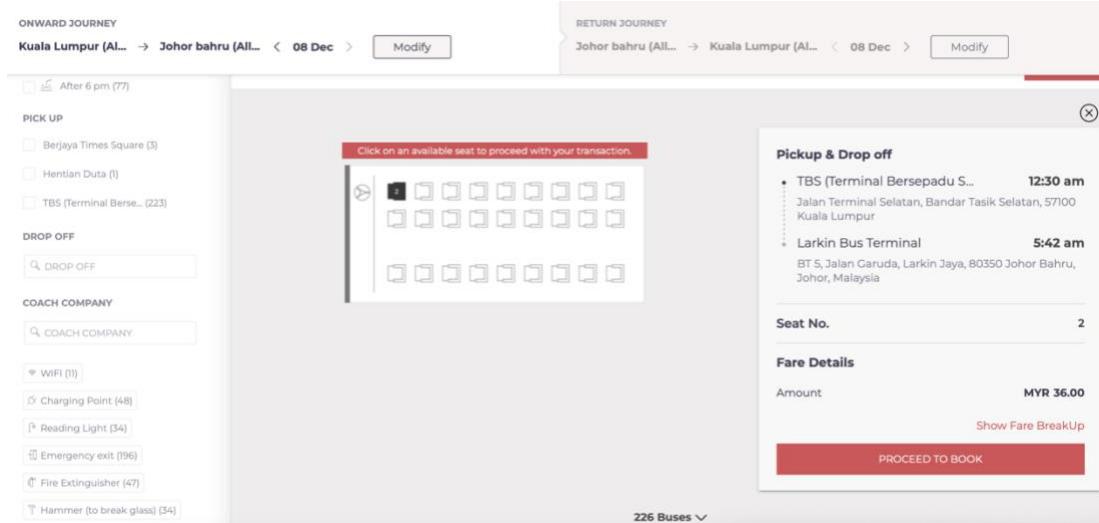


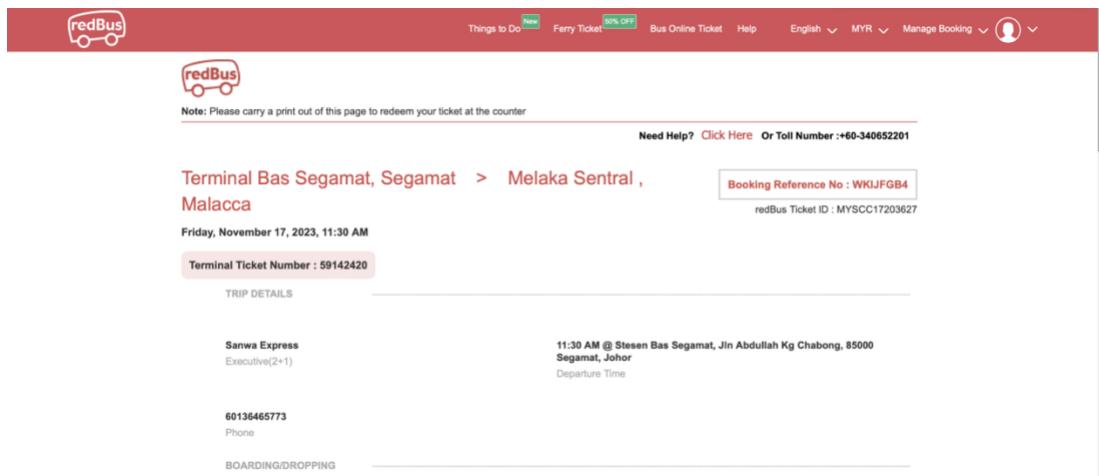
Figure 2.23: Red Bus seat selection (RedBus, n.d.)

After selecting a specific seat, the reservation details will be displayed on the left side of Figure 2.24. The selected seat is highlighted in black, while the available seats are shown in white and the occupied seats are shown in grey.



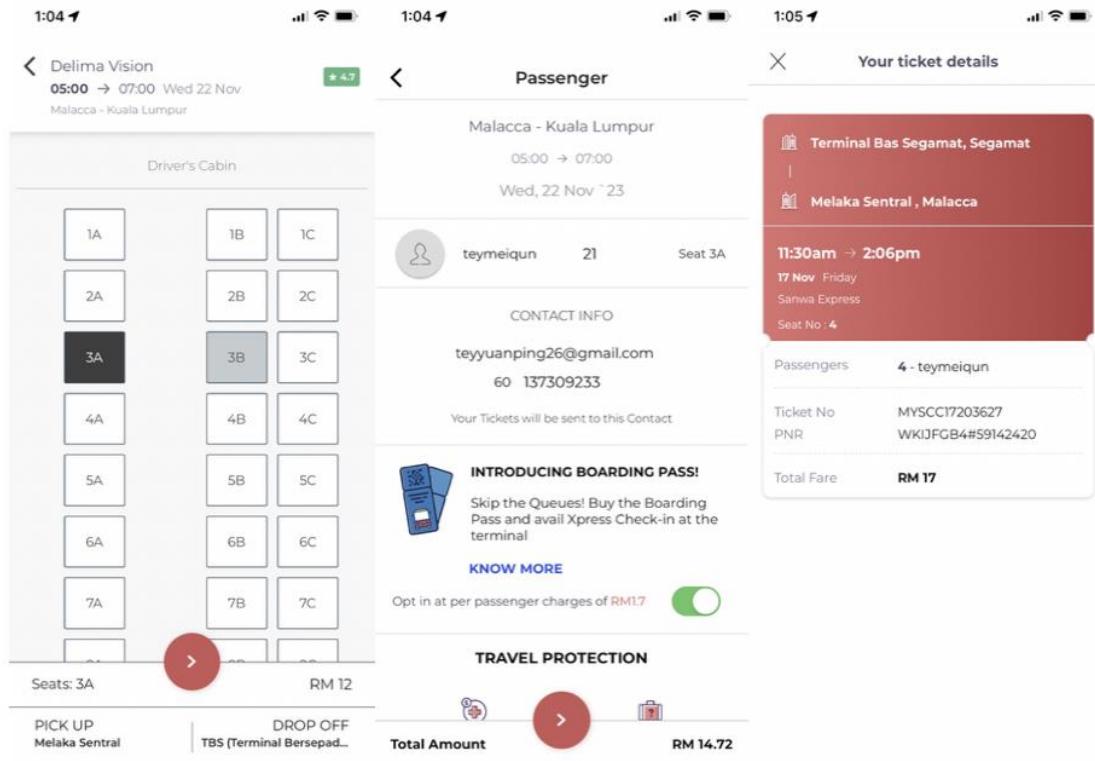
**Figure 2.24: Red Bus seat selection (RedBus, n.d.)**

Once the payment process is complete, the booking details will appear in Figure 2.25. This includes information such as the terminal ticket number, departure time, and booking reference number.



**Figure 2.25: Red Bus reservation details (RedBus, n.d.)**

Figure 2.26 shows the Red Bus application booking process, demonstrating similar functions and flows as the website. It includes seat selection, passenger information, and ticket details, providing a comprehensive view of the booking experience on the app.



**Figure 2.26: Red Bus application booking process (RedBus, n.d.)**

#### Functionalities:

- Bus Ticket Booking: RedBus allows users to search, compare and book bus tickets for a variety of routes and operators.
- Route information: Users can get comprehensive information about bus routes, including departure and arrival locations, schedules and durations.
- Seat Selection: Platforms often provide seat selection functionality that enables users to choose their preferred seat on the bus.
- Live Tracking: Some bus operators on RedBus may offer live tracking, allowing users to monitor the current location and status of their booked bus.

Limitations:

- a. Seat lock: If a RedBus booking fails, the Seat Lock feature temporarily reserves the selected seat. Users may face a short wait time (approximately 10 minutes) before attempting to rebook a temporarily locked seat.
- b. Lack of recommendation: Unable to provide a seat suggestion for booking.

### 2.3 Comparison between Existing Systems

Table 2.1 analyses various seat selecting systems to assess their similarities or differences in their main goal, functionalities, and overall impact. Seat maps are used to display the real-time availability of seats, showing which seats are already booked and which are still available to enhance the employee experience by enabling visual seat selection. The seating map only required by GSC, Red bus and the proposed system. The timetable functionality available on all platforms is crucial as it provides users with basic information about the available booking slots. However, for the proposed system, no time slots are provided as desks should be booked in full day slots, but it does provide the available timetable dates. The proposed system, accessible via a website and mobile app, follow by the GSC and Red Bus. This multi-platform presence ensures that users can seamlessly access its services, making it competitive in this regard.

Although the proposed system does not provide maximum flexibility in terms of time booking options, allowing users to book "unlimited". But the reason is to prevent interrupt caused by accidents and minimize hidden risks. Interestingly, the proposed system does not provide blocking seats functionality, but the reason is that no payments need to be processed, so blocking seats are useless. Lastly, only proposed system provides the QR code verification feature, while other sites omit this feature. This feature can make it simpler and keep the target users safe.

**Table 2.1: Contrast between 4 similar system and proposed system**

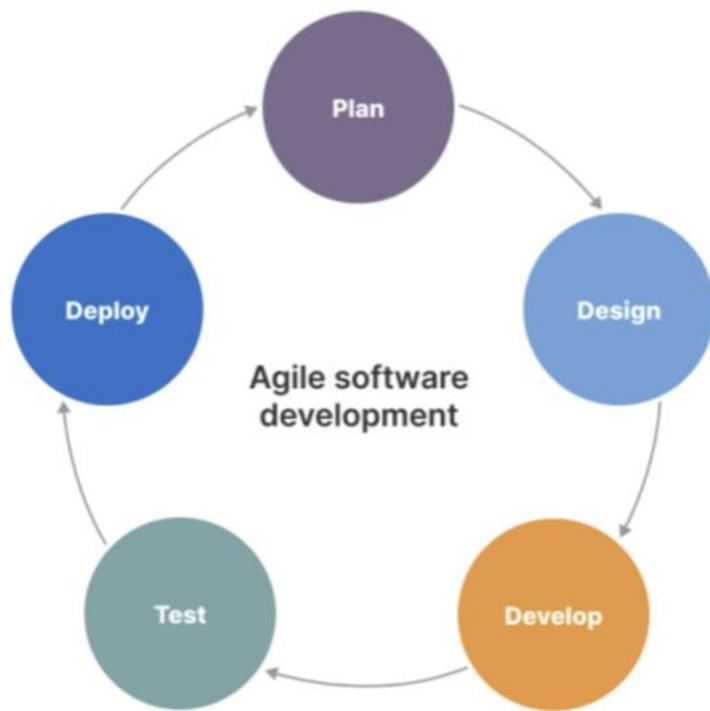
Aspect/ Feature	LiquidSpace	Racquetvenues	GSC	Red Bus	Proposed System
<b>Primary Domain</b>	Flexible Workspace Booking	Badminton Court Booking	Movie Ticket Booking	Bus Ticket Booking	Smart Desk Booking
<b>Platform Availability</b>	Website, Mobile App	Website	Website, Mobile App	Website, Mobile App	Website, Mobile App
<b>Seat map</b>	No	No	Yes	Yes	Yes
<b>Available time schedule</b>	Yes	Yes	Yes	Yes	Yes
<b>Time booking selection duration</b>	No limit	3 months	8 days	3 months	1 month
<b>seat blocking</b>	No	No	8 minutes	10 minutes	No
<b>Qr code verification</b>	No	No	No	No	Yes

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction of Agile model

The agile model is an iterative and flexible approach to software development. Agile focuses primarily on incremental progress, allowing teams to deliver functional products in small, manageable cycles that can be completed in just a few weeks. This approach allows for continuous feedback and adjustments throughout the development process, enhancing communication between cross-functional teams. Once the iteration at a time is planned, it will develop to customers first and there is no long-term plans were made. Agile models are widely accepted in the software development industry for their ability to adapt to changing requirements and efficiently deliver high-quality, customer-focused solutions.



**Figure 3.1: Agile model flow (What is agile software development?, 2023)**

The initial phase of the agile model is the requirements planning phase, which starts with a comprehensive proposal. The proposal outlines the core goals, objectives, and requirements of the project. The project is in collaboration with Continental IT Hub, Petaling Jaya. All of the project details and expected deliverables will be confirmed during collaborative discussions.

The second is the system design phase, which defines the system architecture, database structure, interface layout and component interaction. The use case diagrams, content diagrams, data flow diagrams and data dictionary tables will be drawn in detail to fully understand the functional requirements and data structure of the project. Then the prototyping process will then be initiated to ensure that the following project development is effectively aligned with its intended goals and with expected outcomes.

The development phase of a software project constitutes the actual coding and building of the software system. The template will be apply to streamline and simplify the interface design, ensuring that the interface is both intuitive and efficient. Then all the necessary functions and features will be created by following the deliverables that had been discuss.

After the development phase is completed, a review and feedback session will conduct. During this testing phase, all scenarios are tested and debugged to ensure that no errors occur. The feedback will then refine and improve the system again.

During the deployment phase, execution tests are released to ensure all system functionality is working flawlessly. Issues can then be debugged in a timely manner based on features developed and improvements released to end employees. Once the adjustment is complete, the deployment is run on the stakeholder servers and system performance is monitored so that any issues can be resolved promptly.

### 3.2 Gantt chart

Gantt charts are powerful visual tools that illustrate timelines and tasks for effective project management and coordination. Figure 3.2 illustrates the tasks specified for the Fyp 1 and Fyp 2 of the project reporting and accurately outline the timeline of the different phases in an agile project. At the same time, Figure 3.3 provides a comprehensive statement of all the objectives and deliverables to be achieved during Fyp 2.

	Project Stage	Activity	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
FYP 1 Report	Requirements Planning phase	Abstract & Introduction												
		Literature Review												
		Methodology												
FYP 2 Report	System Design phase	Interface design												
		Conclusion												
	Development phase	System implementation and development												
FYP 2 Report	Testing phase	Testing and implementation												
	Deployment phase	Deploy												
		Conclusion												

Figure 3.2: Fyp 1&2 report

	Activity	Deliverables	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
Objective 1	Check the website and application is functional.	The code is correct, no error & mistakes.The prototype is changing smoothly.												
	Check the flow of app and website .	The seat can be book successfully without error,then the booked seat will not be available for others.												
Objective 2	Check admin privileges.	Check everyone other then admin can not book the date among one month.												
	Check the admin can help others to book,	Admin can login website successfully,can book for themselves, and should also be allowed to book for other employee.												
Objective 3	Check the QR code can be generate unique .	The QR code can contains the necessary registration details.												
	Check the QR code can be scan and login.	User not only can scan and login through the app ,they can also login by using password.												

Figure 3.3: Objectives and deliverables during Fyp 2

### **3.3 Task List**

- 1. FYP 1**
  - 1.1. Project requirement planning**
    - 1.1.1. Defining project title and details.
    - 1.1.2. Identify the problem statement.
    - 1.1.3. Identify the project objectives.
    - 1.1.4. Identify the project scope.
  - 1.2. Literature review**
    - 1.2.1. Research about the smart desk system.
    - 1.2.2. Describe the evaluation of smart desk system.
    - 1.2.3. Discuss the environment in Malaysia.
    - 1.2.4. Analysis the current existing system on market relevant to the project.
    - 1.2.5. Compare the existing system with proposed system.
    - 1.2.6. Identify the potential challenges
  - 1.3. Methodology**
    - 1.3.1. Describe the selected SDLC model.
    - 1.3.2. Prepare Gannt chart
    - 1.3.3. Declare task distribution.
    - 1.3.4. Investigate various technologies or method that can used in systems.
    - 1.3.5. Define the framework and library.
  - 1.4. Interface design**
    - 1.4.1. Prepare Use case diagram.
    - 1.4.2. Prepare Context diagram.
    - 1.4.3. Prepare Data flow diagram.
    - 1.4.4. Draw Entity relationship diagram.
    - 1.4.5. Draw Data dictionary table.
    - 1.4.6. Clearly define the entire process of the system.
    - 1.4.7. Design initial interface prototype.
- 2. FYP 2**
  - 2.1. Development**
    - 2.1.1. Flutter environment setup.

- 2.1.2. Implement prototype.
- 2.1.3. Implement functionalities.
- 2.1.4. Discuss the objective variation .
- 2.2. Testing
  - 2.2.1. Test system performance under various conditions.
  - 2.2.2. Debug system error.
  - 2.2.3. Discuss the objective case study
- 2.3. Deployment
  - 2.3.1. Deploy system.
  - 2.3.2. Report conclusion

### **3.4 Tools technique and programming language**

A comprehensive set of tools, languages and technologies can ensure the development of powerful and efficient desk selection systems with cross-platform functionality and attractive interfaces. The goal of the project is to provide functional prototypes developed using programming languages and tools, with security and privacy measures in place to protect employee data and make development easier.

#### **3.4.1 Flutter**

Flutter is an open source UI toolkit developed by Google for building natively compiled applications. It provides a comprehensive framework for creating applications that run on mobile, web, and desktop platforms. Flutter allows developers to write code once and deploy it to multiple platforms, including iOS, Android, web, and desktop. Flutter provides a rich set of customizable widgets to create visually appealing and consistent employee interfaces.



**Figure 3.4: Flutter (Flutter, n.d.)**

### **3.4.2 Dart**

Dart is a programming language used in conjunction with Flutter to build high-performance native interfaces. It is optimized for building mobile, desktop, server, and web applications. Dart is an object-oriented language that allows developers to model real-world entities and sketch functionality in classes. Dart uses static typing to help catch errors during development and increase the code reliability.

### **3.4.3 HTML**

HTML, or Hypertext Markup Language, is the standard markup language for creating web pages and web applications. HTML is essential for creating the structure and content of web pages, while Flutter provides a modern framework for building cross-platform mobile and web applications. Both technologies have their own strengths and can be used together or separately to create rich, dynamic interfaces. Understanding HTML and Flutter allows developers to create comprehensive solutions that meet the needs of a variety of platforms and users.

### **3.4.4 MySQL**

MySQL is an open-source relational database management system (RDBMS) known for its reliability and performance. It is widely used to store and manage structured data. MySQL serves as the backend database of the channel selection system. It stores and manages data related to seat selection, employee profiles and other related information.

### **3.4.5 Widgets**

Flutter offers a rich library of widgets that play a fundamental role in building employee interfaces for mobile, web, and desktop applications. These widgets serve as the building blocks for creating visually appealing and highly interactive UIs (Tillu, 2023). Flutter widgets are divided into two main categories which is Material Design Widgets and Cupertino Widgets.

Material Design Widgets in Flutter are designed to follow Google's Material Design guidelines and provide a consistent and visually appealing design language. Although it is often associated with Android app development due to Google's Android operating system, Material Design Widgets are not limited to Android apps.

The Cupertino Widget in Flutter is primarily designed to mimic the look and behavior of iOS components. Cupertino Widgets help Flutter apps seamlessly integrate with the iOS ecosystem, ensuring a native-like experience for iOS employees. Although it's often used to build iOS-style interfaces, but it's not limited to iOS apps. Flutter provides the flexibility to mix and match widgets from the Material Design and Cupertino libraries to achieve the desired design on different platforms. UI/UX Design

### **3.4.6 JavaScript**

JavaScript is a versatile scripting language primarily used for client-side web development. It enables dynamic and interactive employee experiences in web browsers. JavaScript is used in client-side scripts to enhance the interactivity of web-based elements in the desktop selection system. It ensures a dynamic and responsive employee interface through features such as real-time updates and asynchronous communication.

### **3.4.7 Development Tools**

#### **a. Visual Studio Code (VS Code)**

VS Code is a popular code editor with excellent support for Dart and Flutter development. It provides extensions and plug-ins to enhance the development experience. With powerful Dart and Flutter extensions, VS Code provides intelligent code completion, debugging, and real-time code analysis to make the development process smoother and more efficient. VS Code also supports Flutter's rich ecosystem of packages and extensions, making it easy to integrate third-party libraries and tools.



**Figure 3.5: VS code (Visual studio code, 2021)**

#### **b. Android Studio**

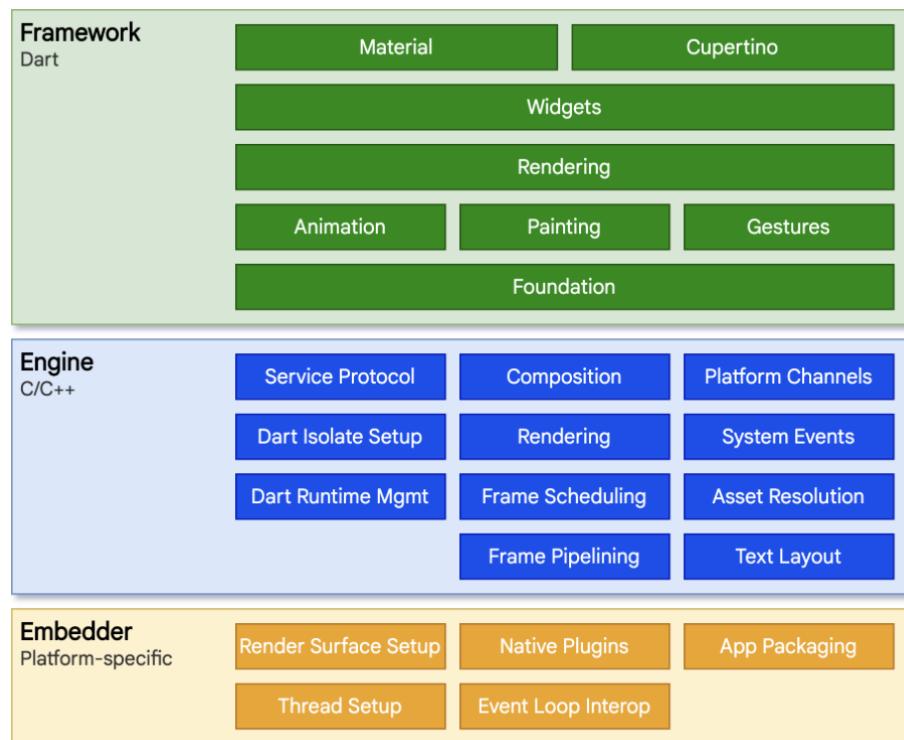
Android Studio is an integrated development environment (IDE) specifically for the Android platform. Android Studio is a powerful, feature-rich IDE developed by Google that is tailor-made for Android and Flutter development. It offers comprehensive Flutter support through plugins and extensions, making it a solid choice for building cross-platform apps. Android Studio provides an intuitive UI design editor for creating Flutter app layouts and simplifies the process of managing Android-specific configuration and dependencies. With its built-in emulator and device testing capabilities, developers can easily test and debug Flutter apps on Android devices and virtual machines.



**Figure 3.6: Android Studio (Android Mobile App Developer Tools, n.d.)**

### 3.5 Framework

In Flutter, the entire framework is mainly focused on front-end development. In fact, Flutter is a complete front-end UI framework for building employee interfaces across various platforms including mobile, web, and desktop. Flutter consists of three architectural layers: the framework, the engine, and the embedder (Your Guide to the Flutter Framework, 2021). The Flutter framework is a UI toolkit that helps developers create high-performance, high-fidelity applications. This rich library can be divided into multiple layers, such as basic classes, widgets, rendering, etc.



**Figure 3.7: Architectural layers (Flutter Architectural Overview, n.d.)**

### 3.5.1 Backend Frameworks

Firebase is a platform developed by Google for creating mobile and web applications. It provides a variety of tools and services to help developers build high-quality applications, including real-time database, authentication, analytics, cloud functions, and more. Firebase acts as a Backend as a Service (BaaS), allowing developers to focus on building the front end of their applications without having to worry about managing servers and infrastructure.

Key Features of Firebase:

- Realtime Database: A NoSQL cloud database that stores data and synchronizes it between clients in real time.
- Firestore: A flexible, scalable database for mobile, web, and server development.
- Authentication: An easy-to-use authentication service that supports email/password, phone authentication, and third-party providers like Google, Facebook, and Twitter.
- Cloud Functions: Run backend code in response to events triggered by Firebase functions and HTTPS requests.
- Cloud Storage: Store and serve user-generated content like photos and videos.
- Hosting: Provides fast, secure static hosting for web apps.



**Figure 3.8: Firebase** (Firebase, n.d.)

### **3.6 Library**

Flutter is a popular open source UI software development kit developed by Google. It can be used to develop mobile, web, and desktop applications from a single code base. These libraries are part of a large ecosystem that makes Flutter a versatile and powerful framework for building modern applications. Each library adds specific features to Flutter, helping developers to easily create feature-rich and high-performance applications.

#### **3.6.1 QR code generation**

The QR.Flutter library is a widely used Flutter package that simplifies the generation of QR codes in Flutter applications (Qr\_Flutter | Flutter Package, n.d.). With QR.Flutter it is easy to create QR codes from text data. One of its main advantages is its customization options, which allow the appearance of the QR code to be customized to preference. This includes setting the size, color scheme, and error correction level of the QR code. Whether it's a QR code for data sharing, authentication, or other purposes, QR.Flutter makes QR code generation a simple and efficient process that enhances your app's functionality and employee experience.

#### **3.6.2 QR code scanning**

The Barcode Scanner SDK library is a valuable asset for Flutter (Android Barcode Scanner Library Using Kotlin - Scanbot SDK, 2023). This library simplifies the process of capturing and decoding QR codes and barcodes using your device's camera. This library can scan the codes by various types of barcodes such as QR codes, barcodes, etc. Employees can scan QR codes and barcodes simply by activating the device's camera, and the library provides easy access to scanned data and related metadata.

#### **3.6.3 Maps and location services**

The Google Maps API provides a comprehensive mapping service with features such as geocoding, which converts addresses into geographic coordinates (Google\_Maps\_Flutter | Flutter Package, n.d.). It can also route and visualize maps. Location autocompletes for embedded maps, websites and apps for efficient location

searches. In addition to this, it can display maps in various layers, including streets, satellite imagery, and terrain.

#### 3.6.4 Interface design tools

Figma is a cloud-based design and prototyping tool that enables collaborative applications for real-time interface design. Collaboration for seamless teamwork, and interactive prototyping for testing employee flows.



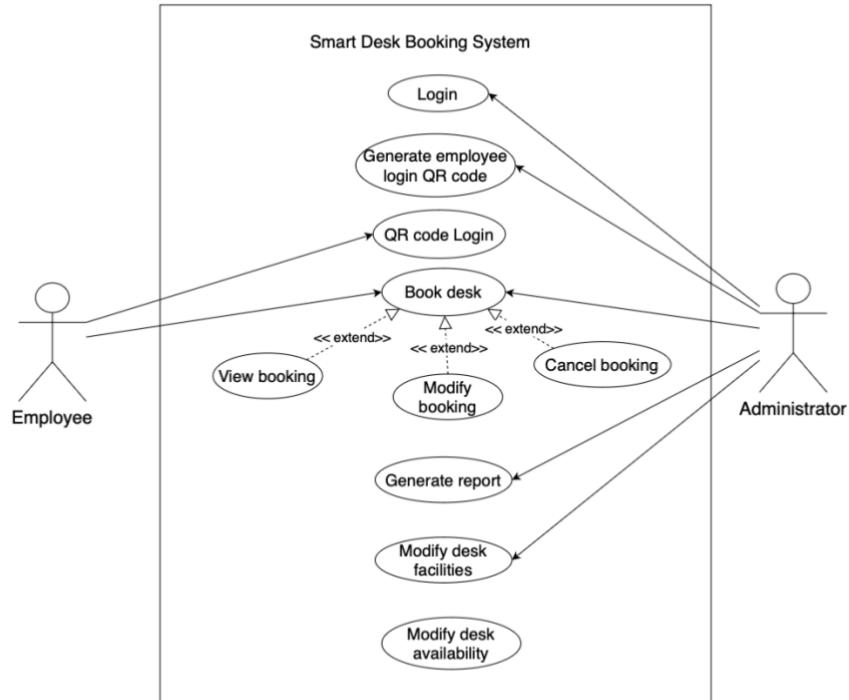
**Figure 3.9: Figma (Figma, n.d.)**

## CHAPTER 4

### PROPOSED SYSTEM

#### 4.1 Use Case

Use case diagrams show how the system responds to employee and administrators requests consisting of multiple use cases. In the diagram below, the participants are employee and administrator. The boxes (also called boundaries) contain all the use cases and how the data connects to each other through arrows. A "extend" relationship indicates when one use case is extended by another use case. For example, the view, cancel and modify desk function was extended by the book desk action.

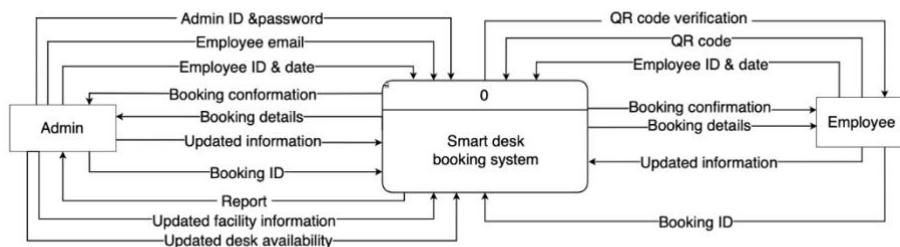


**Figure 4.1: Use case diagram**

## 4.2 Context Diagram

A context diagram (CD) also known as DFD level 0, shows the flow of information that changes through processes in a system. From the diagram below, both admin and employee have some of the same processes, so both parties will be working with the same data. For example, both admin and employee will get the booking details and booking confirmation.

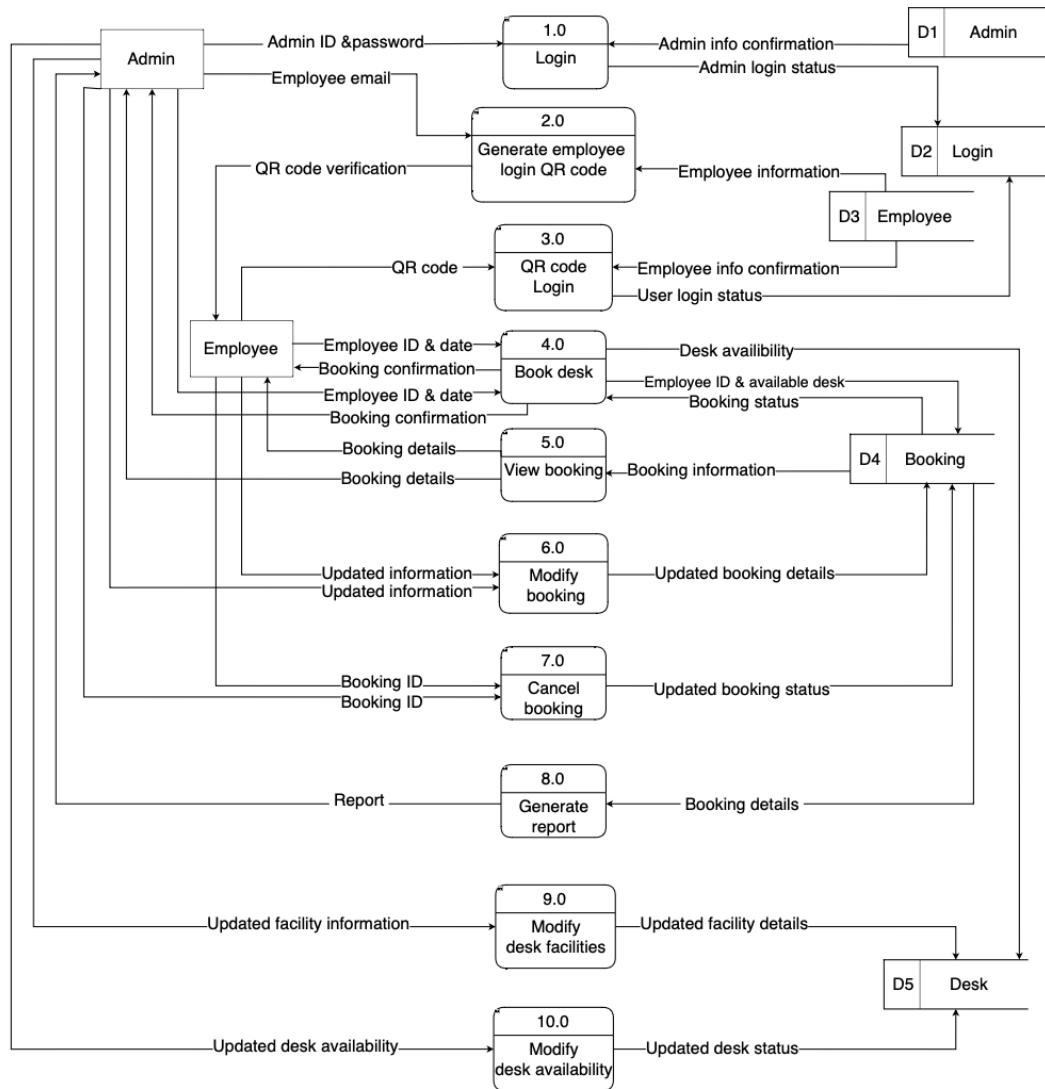
After admin login by admin ID and password, admin can generate QR code. The QR code verification will be send out through email to the employee. The employee will then scan the QR code to verify the identity authenticity for login the system. For booking a desk, the employee ID and date will be passed and stored in the system. Admin and employee can also view, modify and cancel the booking details, but admin have permission to generate the report. To manage the desk availability, admin could also update the desk detail such as the facility information and desk availability.



**Figure 4.2:Context Diagram**

### 4.3 Data Flow Diagram

Level 1 DFDs show the system's main processes, data flows, and data stores in greater detail than level 0 diagram. In the figure below, there are few processes, labelled "1.0" to "10.0" respectively. These processes show how the system processes data flows and transforms inputs into outputs.

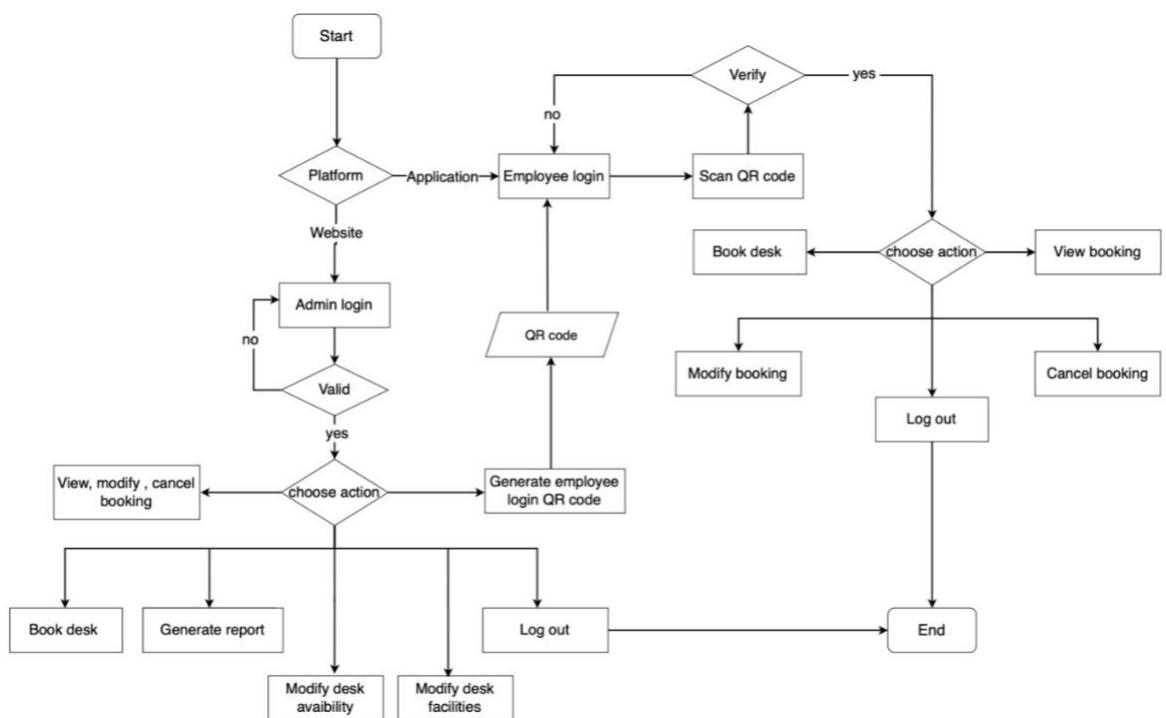


**Figure 4.3: Data Flow Diagram (level 0)**

#### 4.4 Flow chart diagram

Flow chart is a graphical representation that describes the dynamic aspects of a system, process, or workflow. It can display complex processes in the system in the form of diagrams, making it easier to understand.

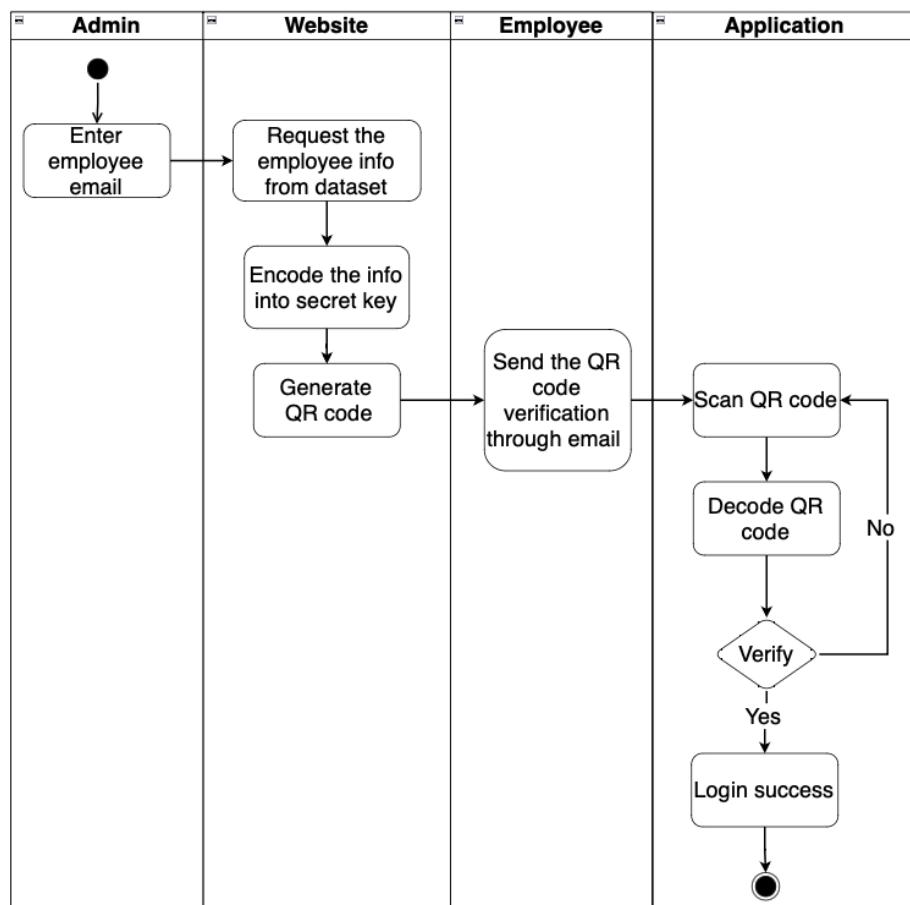
To start this project, the first step is to choose a platform. The proposed system offers two platform options which is application and website. If there is admin, it will be processed as admin login on the website, otherwise it will be retrieved as employee login through the application. When the admin login is valid, the admin can choose to generate a QR code for employee to log in, book desk, view, modify or cancel booking, generate reports, modify desk details and log out. When the employee is authenticated by logging in using the QR code, the employee can book the desk by selecting the date. After confirming the details, the employee will receive the booking detail. This booking process is identical for both administrators and employees but takes place on different platforms. Then employee could also be able to view, modify and cancel the bookings.



**Figure 4.4: Flow chart of Smart Desk Booking System**

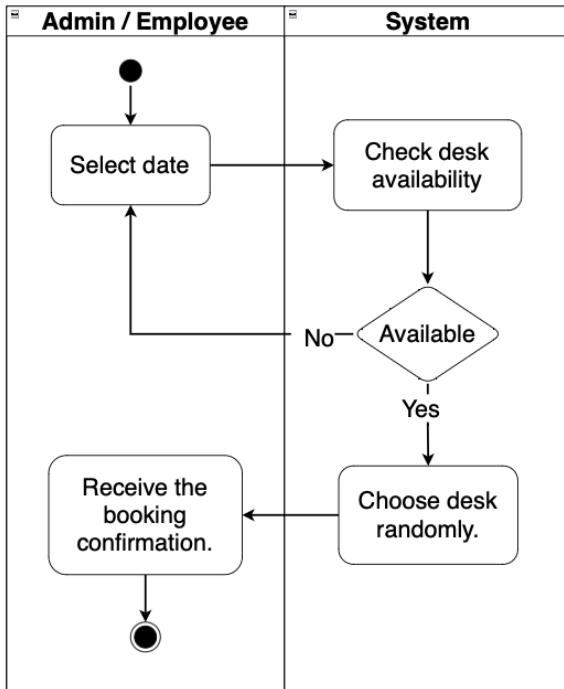
#### 4.4.1 Activity diagram for the QR code process

To access to the application, employees must initiate the login process through the admin by entering the email on the website. Subsequently, the system will retrieve the employee's information from the database and encode it into a secret key, which is then used to generate a QR code. An email containing a verification link and the QR code will be sent to the employee. To complete the login, the employee must open the application, scan the QR code, and the application will automatically decode it. It will then verify whether the QR code is valid or expired. If it is valid, the employee will successfully access to the application.



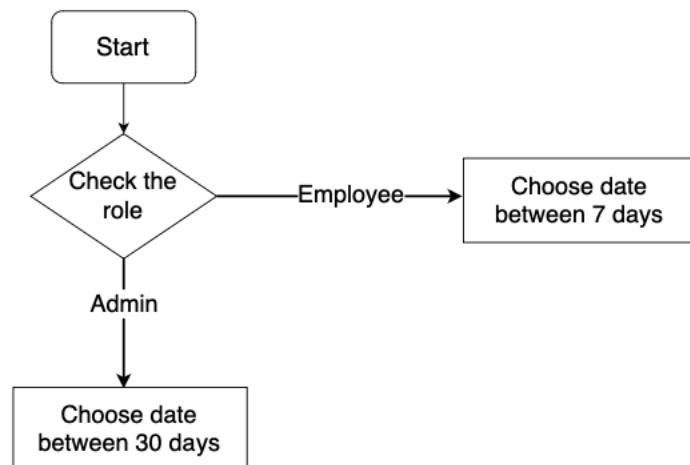
**Figure 4.5: Activity diagram for the seat allocation process**

The system will allocate the desk during the booking process. Both employees and admin must select a booking date first, and the system will subsequently verify the desk's availability for that day. If no desks are available on the selected date, the system will notify and prompt the notice to choose another date. However, if desk is still available, the system will randomly allocate a desk. A booking confirmation will be displayed once the booking is successfully completed.



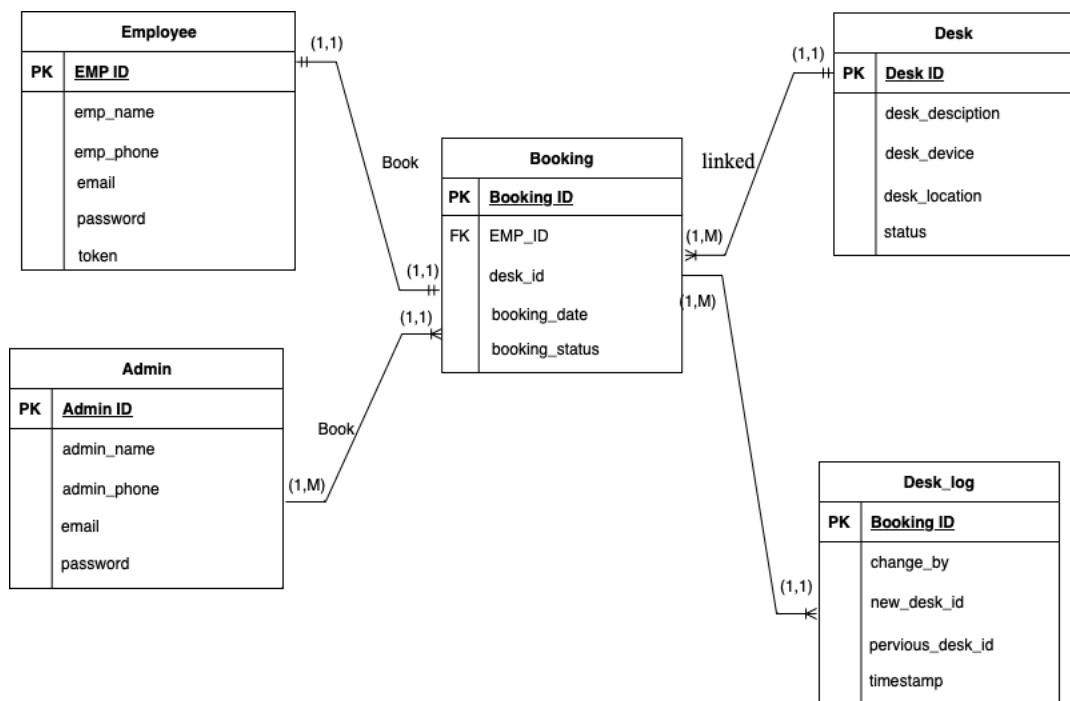
**Figure 4.6: Flow chart for the time privilege process**

The system will determine the booking date privileges based on the role. While different platforms have categorized admin and employee, there is still a need to verify roles rather than relying solely on the platform. If the role is admin, then the system will be offering 30 days range of selection dates. However, if the role is employee, then the booking will be limited to 7 days.



**Figure 4.7: Entity relationship diagram**

ERDs are important tools in database design and facilitate the visualization, planning, and communication of complex data relationships. The ERD diagram is shown below. There are 5 tables in the database, namely login table, employee table, admin table, booking table and desk table. The login table mainly stores login status and login timestamp. Basic information such as names, phone numbers, email addresses, and passwords are stored in the employee and admin tables by different role. The booking table will store booking details such as booking date, booking desk and employee ID. The desk table stores some information, such as how many devices or tables are available for use right now.



**Figure 4.8: Entity relationship diagram**

The business rules derived from the ERD provided can be summarized as follows:

- Employee can reserve multiple bookings and a booking can be book by only one employee.
- Admins can book multiple bookings, and a single booking can be book with multiple admins.
- Each booking is linked to one desk, and each desk can have multiple bookings over time.
- Each bookings could have multiple desk logs , but one desk logs was only owned by one bookings.

## 4.5 Data Dictionary

Table 4.1 consists of five tables, namely Login, Employee, Admin, Booking and Desk. This entity table is designed to store basic information about employees, admin and booking details. Once some information changes, each table will be affected by the foreign key related tables.

**Table 4.1: Entity Table**

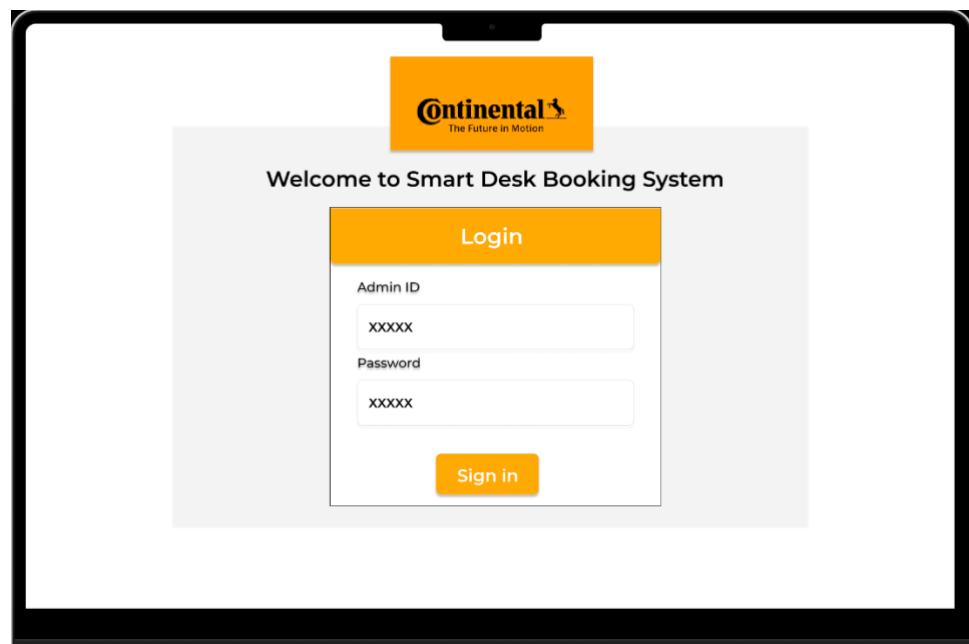
Table	Attribute Name	Content	Data Type	Null	Default	PK	FK	FK Reference Table
Employee	EMP_ID	Employee ID	STRING	No	None	Yes	No	
	emp_name	Employee name	STRING	No	None	No	No	
	emp_phone	Employee phone	STRING	No	None	No	No	
	email	Employee email	STRING	No	None	No	No	
	password	Employee password	STRING	No	None	No	No	
	token	Employee token	STRING	No	None	No	No	
Admin	Admin_ID	Admin ID	STRING	No	None	Yes	No	
	admin_name	Admin name	STRING	No	None	No	No	
	admin_phone	Admin phone	STRING	No	None	No	No	
	email	Admin email	STRING	No	None	No	No	
	password	Admin password	STRING	No	None	No	No	
Booking	Booking_ID	Booking ID	STRING	No	None	Yes	No	
	EMP_ID	Employee	STRING	No	None	No	Yes	Employee
	desk_id	Desk ID	STRING	No	None	No	Yes	Desk
	booking_date	Booking date	TIMESTAMPS	No	None	No	No	
	booking_status	Booking status	STRING	No	None	No	No	
Desk	Desk_ID	Desk ID	STRING	No	None	Yes	No	
	desk_description	Desk description	STRING	No	None	No	No	
	desk_device	Desk device	STRING	No	None	No	No	
	desk_location	Desk location	STRING	No	None	No	No	
	status	Desk status	STRING	No	None	No	No	
desk_log	Booking_ID	Booking ID	STRING	No	None	Yes	Yes	Booking
	change_by	Employee ID that make changes	STRING	No	None	No	No	
	new_desk_id	The new changes desk	STRING	No	None	No	No	
	pervious_desk_id	The pervious desk	STRING	No	None	No	No	
	timestamp	Timestamp	TIMESTAMPS	No	None	No	No	

#### **4.6 Initial prototype**

The smart desk system will be separate into two modules, one is admin module, and the other is employee module. The admin module will be used through the website platform while the employee will through the application.

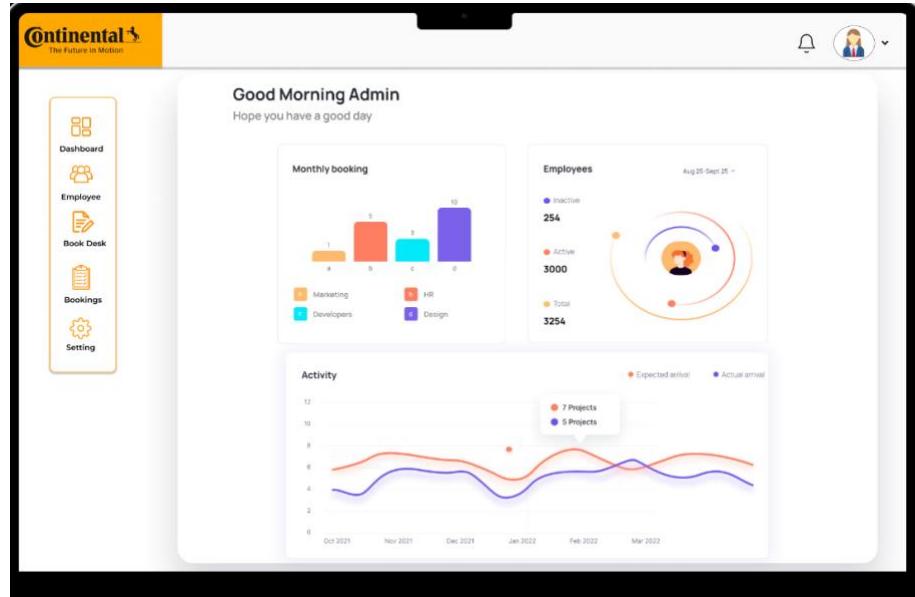
##### **Admin module (Website)**

The admin will start in the login page which shown as below. The admin need to login first by entering their admin ID and password.



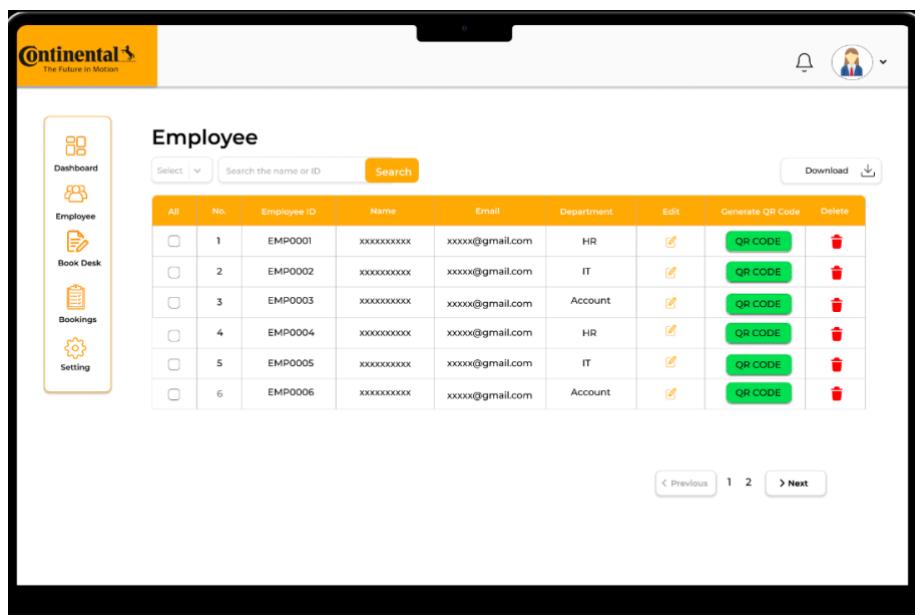
**Figure 4.9: Admin login page**

After successful login, it will be retrieved to the dashboard page. Then there is a navigation bar on the left to retrieve the corresponding page.



**Figure 4.10: Admin dashboard page**

While clicking the ‘employee’ on the navigation bar, the page will list out all the employees' details. The admin could also edit, delete and add new employees. Once the admin click the QR code button, an email will send to that employee with the login link and Qr code.



**Figure 4.11: Employee page**

While clicking the ‘Book Desk’ on the navigation bar, the page will display the booking details form. The admin needs to enter the employee ID and select the date to make the booking. Once clicking on the date, a calendar will pop up as shown as Figure 4.13, and it only allowing to book beyond 30 days.

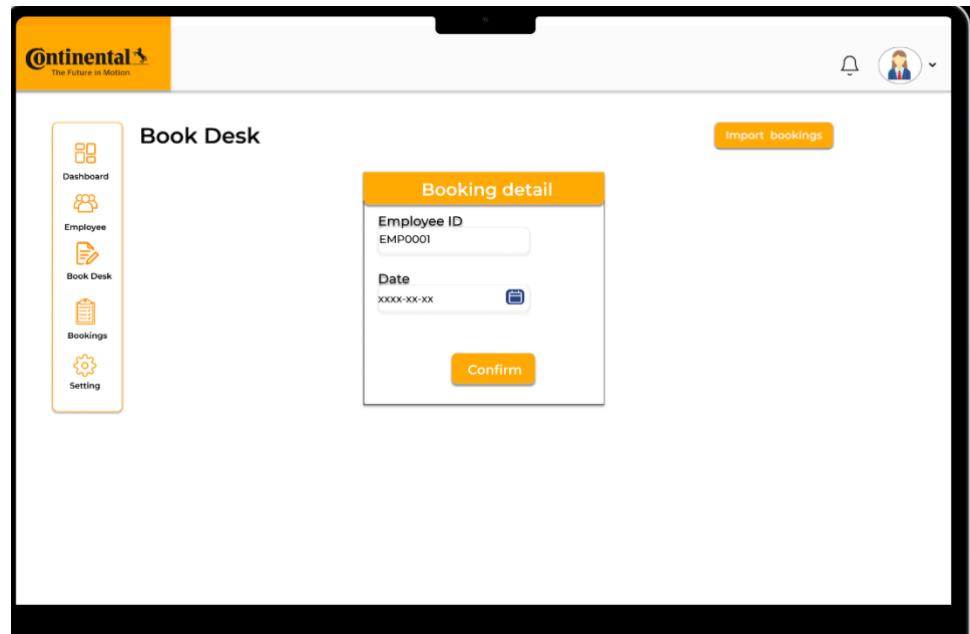


Figure 4.12: Book desk page

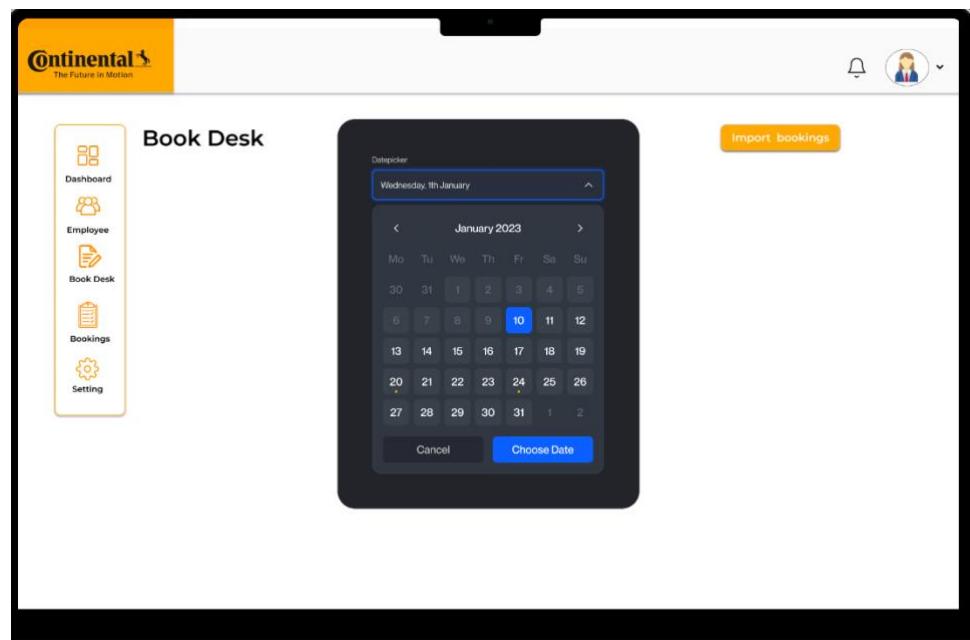


Figure 4.13: Calendar pop up

If the desk booking is successful, it will retrieve to the booking page and show all of the booking list. The booking can able to edit or delete. The report can also be generated by clicking on the ‘generate report’ button. A mapping view is also provided to see the seating by changing it to ‘view the map’ mode.

The screenshot displays the 'Bookings' section of the Continental application. On the left, there's a sidebar with icons for Dashboard, Employee, Book Desk, Bookings (which is selected and highlighted in orange), and Setting. The main area has tabs for 'View by list' (selected) and 'View by map'. Below that is a search bar with dropdowns for 'Select' and 'Search the name or ID', and a 'Search' button. To the right is a 'Generate report' button with a dropdown arrow. The central part is a table showing booking details:

No.	Employee ID	Date	Booking ID	Seat No.	Create Date	Edit	Delete
1	EMP0001	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		
2	EMP0002	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		
3	EMP0003	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		
4	EMP0004	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		
5	EMP0005	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		
6	EMP0006	xxxx-xx-xx	xxxxxxxxxx	XX	xxxx-xx-xx xxxxxxxxxx		

At the bottom, there are navigation buttons for 'Previous', '1 2', and 'Next'.

**Figure 4.14: Booking (view by list) page**

The available and not available seat will be marking as black and red. Admin can select the date to view in the right-hand side. Admin can click on the booking list to see the location of the desk that book by specify employee. Then the booking desk will mark as blue as shown in Figure 4.15.

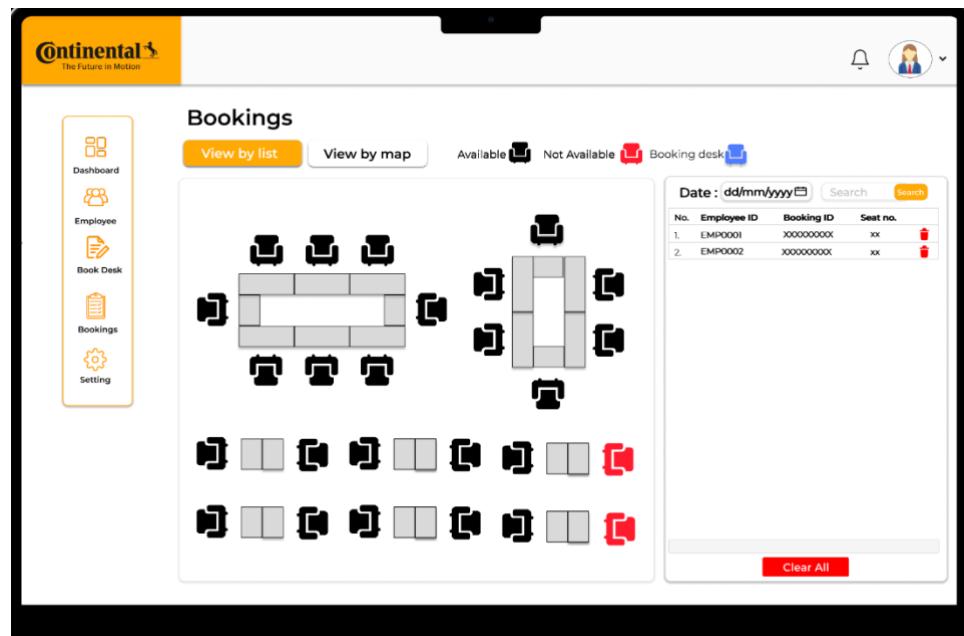


Figure 4.15: Booking (view by map) page

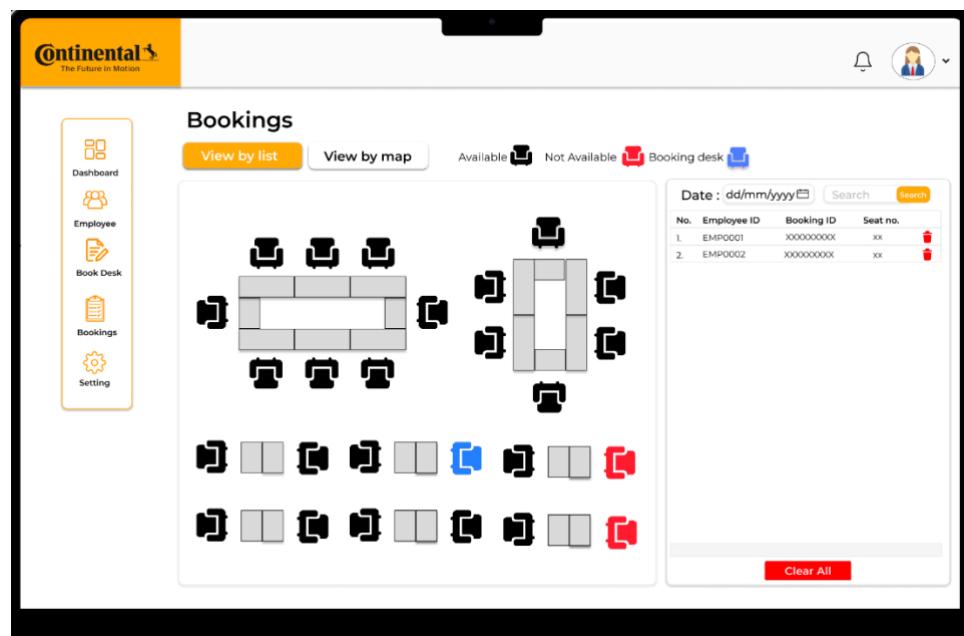
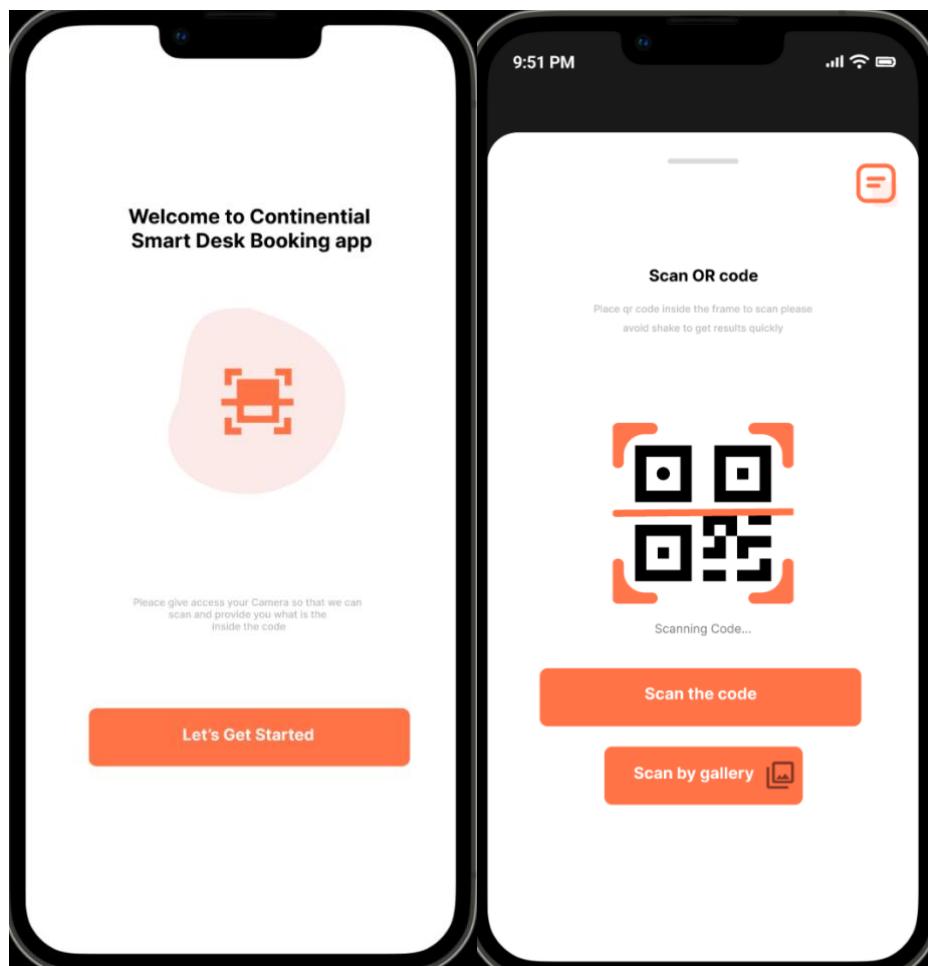


Figure 4.16: Booking (view by map) page showing the booking desk location

## Employee

Once the admin sends out the login email to the employee, the employee need click on the link provided and download the app. Employees can then scan the QR code in the email to log in.



**Figure 4.17: Employee scan QR code page**

After a successful login, employees can book a desk, view bookings, or view a map of desks. Once the employee chooses to book desk, the only thing that need to do is select the date only.

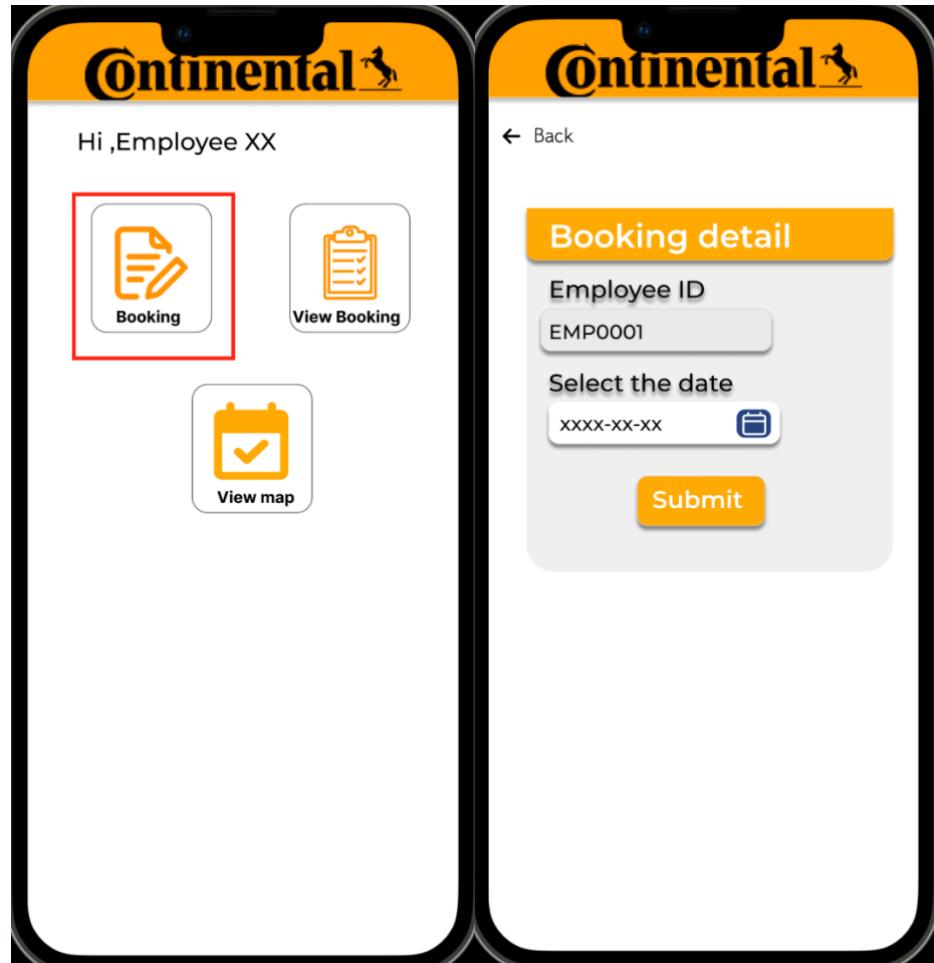
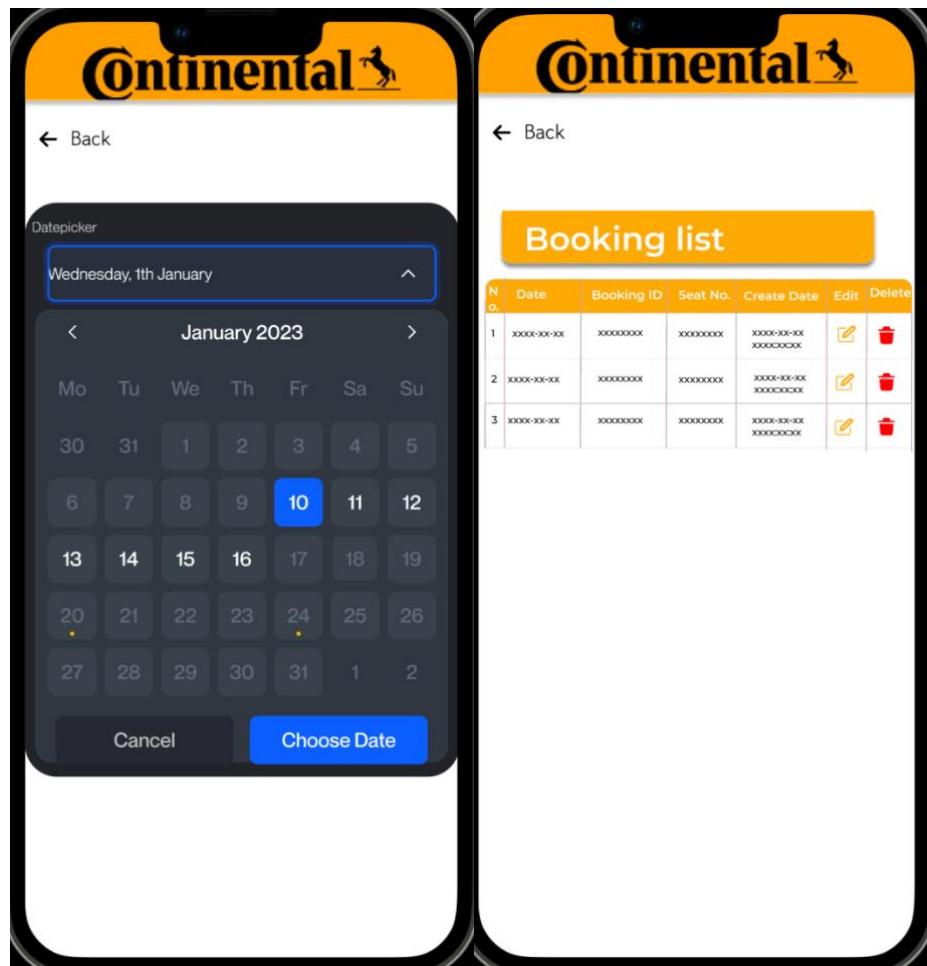


Figure 4.18: Employee booking page

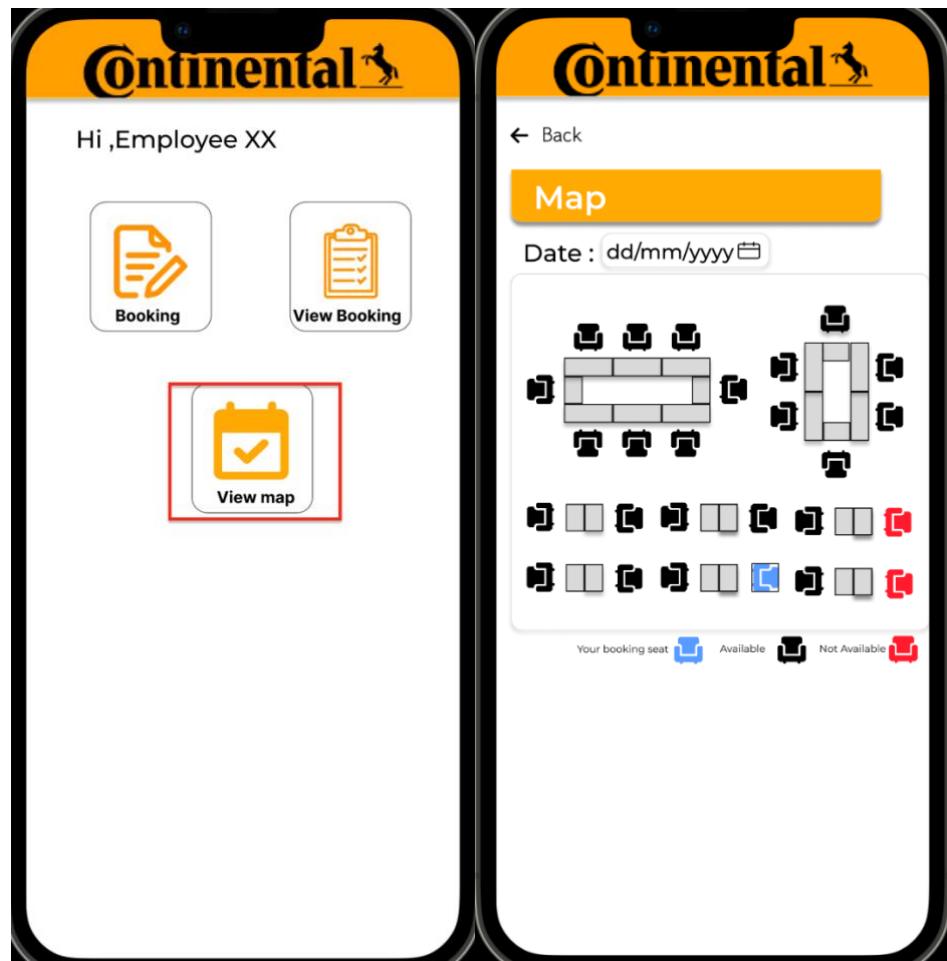
The calendar will prompt out too while selecting the date. When the booking is success, employee will be return to the booking list page to view the detail.

Employee can also edit or delete the booking. If the admin helps to book the desk, employee can also get updated booking right away on this page.



**Figure 4.19: Employee booking list page**

By clicking on the ‘view map’, the employee can see the available seats and the desk they booked. There will be displayed in blue color seat to indicate that the seat is reserved by an employee.



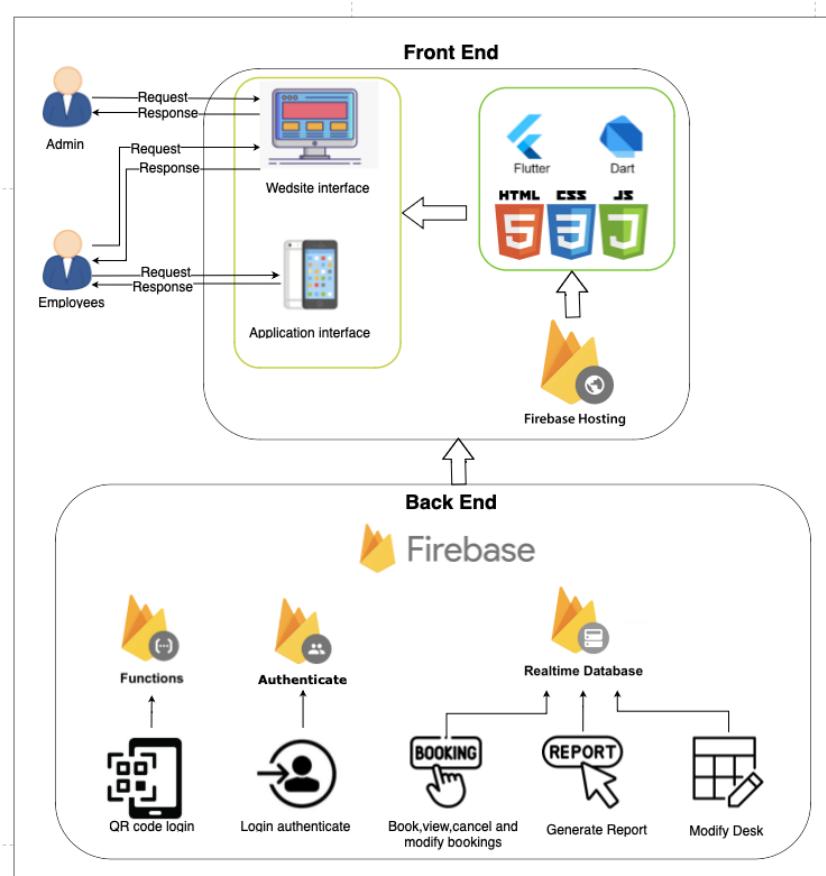
**Figure 4.20: Employee desk map page**

# CHAPTER 5

## IMPLEMENTATION

### 5.1 System Architecture

Figure 5.1 illustrates the system architecture of the proposed mobile and web application interface, divided into front-end and back-end components. Employees interact with the application through a unified interface developed for mobile and web platforms using Flutter and Dart. The backend is powered by Firebase for serverless functionality to handle business logic, Firestore for flexible and scalable real-time data storage, and Realtime Database for optimized data synchronization. The architecture ensures a seamless employee experience across different platforms while leveraging powerful backend services for efficient data management and operations.



**Figure 5.1: System architecture**

### **5.1.1 Front end**

Flutter is a great framework as described in chapter 3. It is the main motivation for the proposed system and in this project, it is shown in more depth how Flutter and Dart provide the basis for building powerful cross-platform applications on the front-end.

Firebase Hosting is primarily a front-end service, as it serves static assets and ensures that users can load and interact with your web app quickly and securely. However, it integrates well with back-end Firebase services, allowing for a full-stack development experience where the front-end provided by Firebase Hosting can interact with the various back-end services provided by Firebase.

Firebase Hosting helps in building a Flutter web or app and compiling it into static assets. These assets include HTML, CSS, JavaScript, and any other resources required for the app or web. This compilation is done using the flutter build web command and the output is placed in the build/web directory. Firebase Hosting also integrates seamlessly with other Firebase services, which will cover later in the backend.

### **5.1.2 Firebase Function**

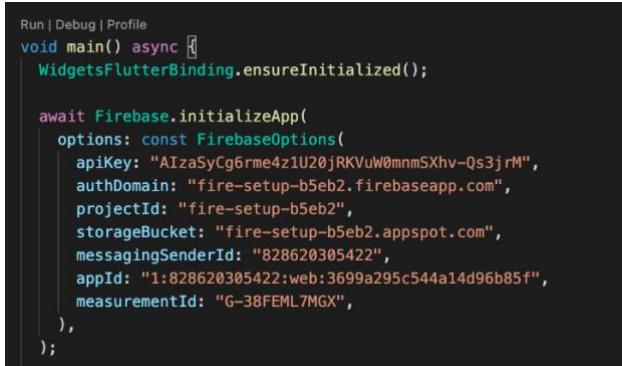
Firebase functions are serverless functions that run in response to events triggered by Firebase functions and HTTPS requests. Therefore, it is a good choice to handle the QR code login process in this system through the URL retrieval API generated after hosting. When the employee scans the QR code, a Firebase function is triggered to handle the login request and work with Firebase Authentication to verify the employee's identity.

Function	Trigger	Version	Requests (24 hours)	Min/max instances	Timeout
api us-central1	HTTP Request https://us-central1-firebase... ...	v1	2	0 / -	1 m

**Figure 5.2: API set up in Firebase Function**

### 5.1.3 Firebase JS API

The Firebase JS API is a powerful tool that simplifies the integration of various Firebase services with web applications, providing powerful authentication, real-time data management, back-end logic execution, and more. By using the Firebase JS API in this project, the project first initializes Firebase using the configuration details obtained from the Firebase console as shown as Figure 5.3.

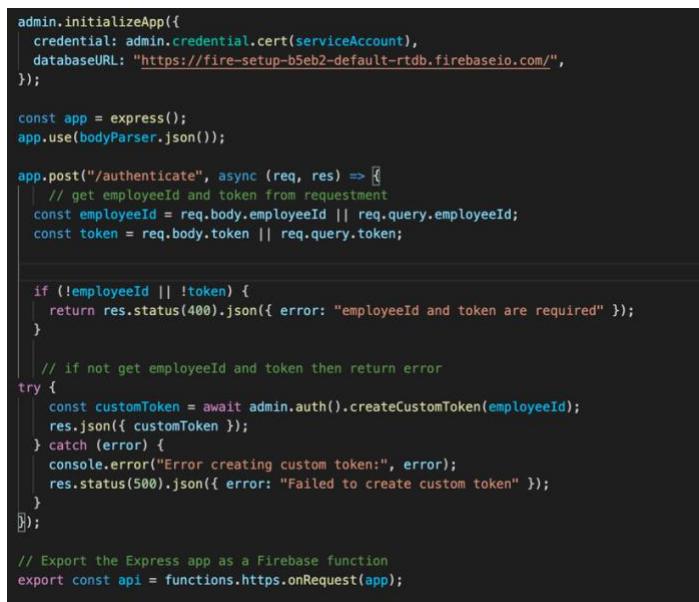


```
Run | Debug | Profile
void main() async {
    WidgetsFlutterBinding.ensureInitialized();

    await Firebase.initializeApp(
        options: const FirebaseOptions(
            apiKey: "AIzaSyCg6rme4z1U20jRKVuW0mmSXhv-Qs3jrM",
            authDomain: "fire-setup-b5eb2.firebaseio.com",
            projectId: "fire-setup-b5eb2",
            storageBucket: "fire-setup-b5eb2.appspot.com",
            messagingSenderId: "828620305422",
            appId: "1:828620305422:web:3699a295c544a14d96b85f",
            measurementId: "G-38FEML7MGX",
        ),
    );
}
```

Figure 5.3: Initial Firebase setting

Then the below code show how to sets up a JS file application that integrates with Firebase to provide an authentication endpoint. This endpoint validates incoming requests, creates a custom authentication token using the Firebase Admin SDK, and handles errors appropriately. The Express app is then deployed as a Firebase Cloud Function, allowing it to be triggered via HTTPS requests.



```
admin.initializeApp({
    credential: admin.credential.cert(serviceAccount),
    databaseURL: "https://fire-setup-b5eb2-default-rtdb.firebaseio.com/",
});

const app = express();
app.use(bodyParser.json());

app.post("/authenticate", async (req, res) => {
    // get employeeId and token from request
    const employeeId = req.body.employeeId || req.query.employeeId;
    const token = req.body.token || req.query.token;

    if (!employeeId || !token) {
        return res.status(400).json({ error: "employeeId and token are required" });
    }

    // if not get employeeId and token then return error
    try {
        const customToken = await admin.auth().createCustomToken(employeeId);
        res.json({ customToken });
    } catch (error) {
        console.error("Error creating custom token:", error);
        res.status(500).json({ error: "Failed to create custom token" });
    }
});

// Export the Express app as a Firebase function
export const api = functions.https.onRequest(app);
```

Figure 5.4: Express Setup in JS file

#### **5.1.4    Firebase Authentication**

Firebase Authentication verifies the credentials of employees and administrators during the sign-in process. Once authenticated, employees and administrators can access other parts of the system. It ensures the secure role-based access to system functionality.

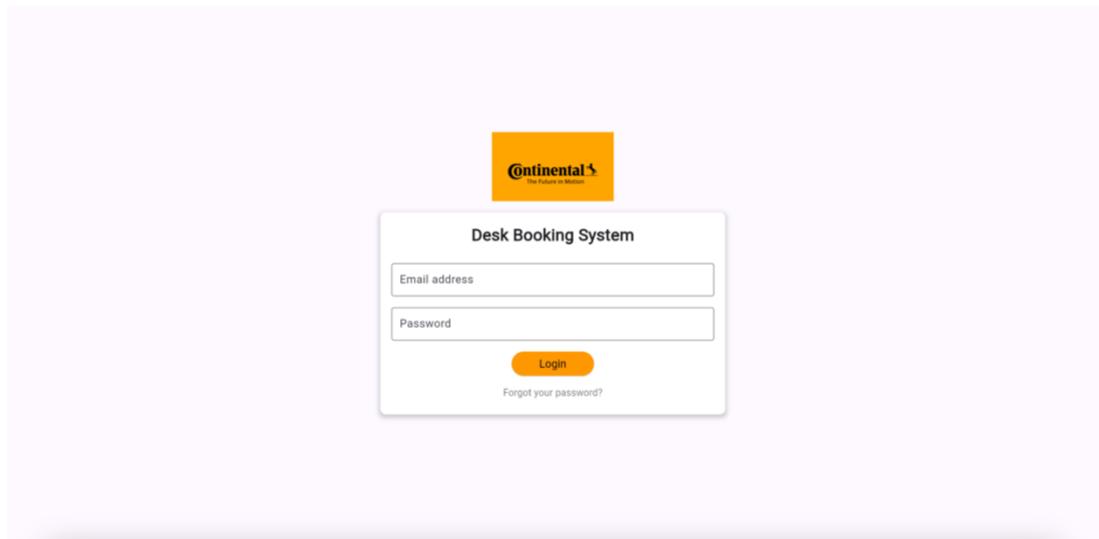
#### **5.1.5    Firebase Realtime Database**

Realtime Database ensures that any changes to the data are immediately reflected on all connected clients, providing real-time updates like Firestore. Realtime Database stores data in JSON trees. This hierarchy allows for easy nesting of data but can become complicated for deeply nested objects. Firebase Realtime Database is a cloud-hosted NoSQL database that stores data as JSON and synchronizes it in real time to every connected client.

Firestore and Firebase Realtime Database on the backend provide a powerful, real-time, and secure data management solution that enables seamless interaction and data synchronization for front-end applications built with Flutter and Dart. Authenticated employees and administrators can interact with the booking system to book, manage, cancel and view bookings in real-time using the real-time database. For administrators, it can retrieve data to generate reports and modify desk details and reflect the changes in the real-time database.

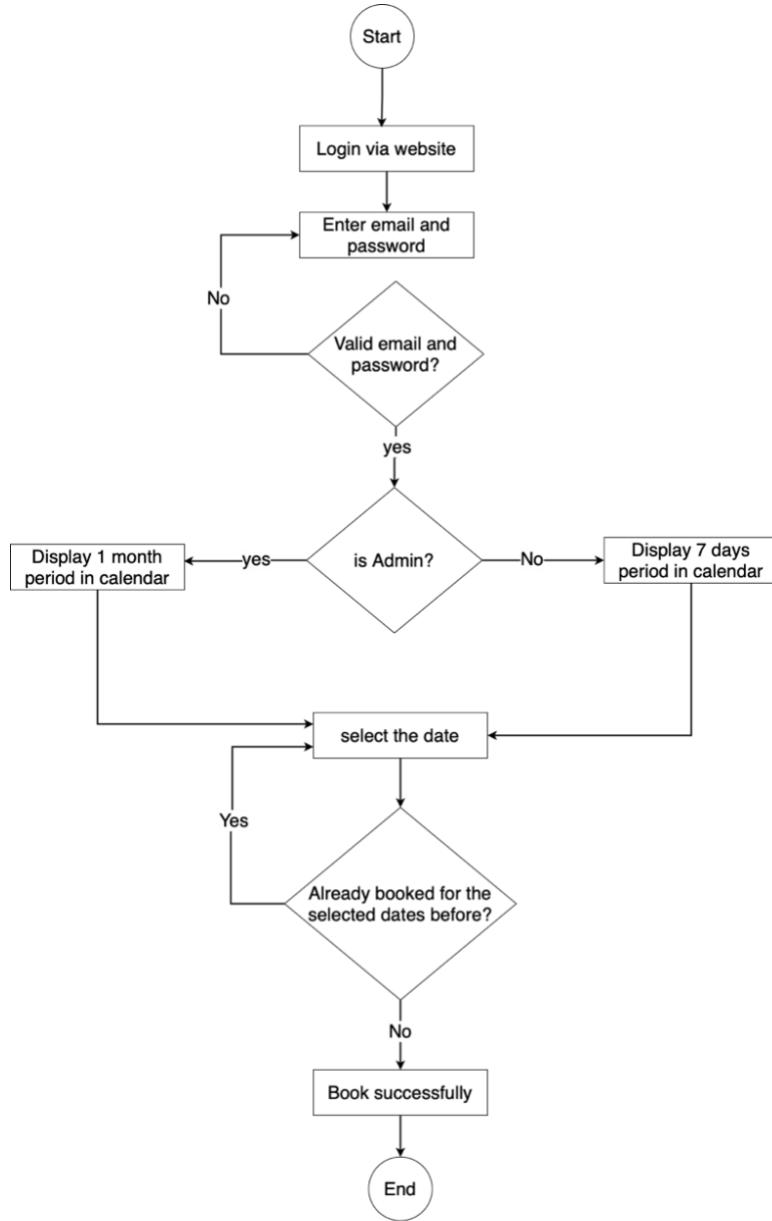
## 5.2 Admin Interface

In this proposed system, the admin module is mainly used on the website. The employees are also allowed to use the website to make bookings, therefore the login page requires an email and password, rather than another unique ID.



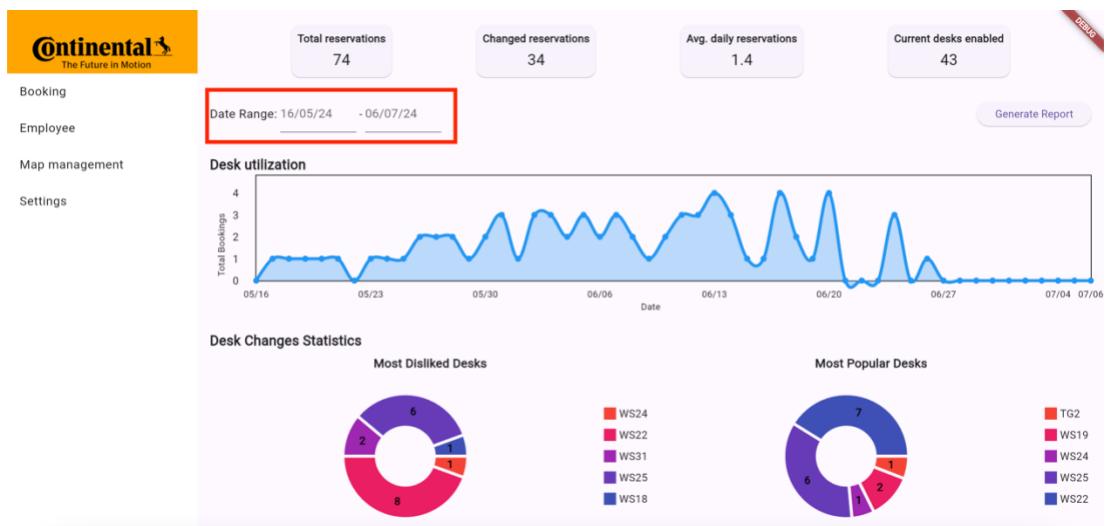
**Figure 5.5: Login page in website**

In this proposed system, the system flow for booking desks includes login verification, role-based schedule calendar display, and ensuring that there are no duplicate bookings for the same date. Administrators can have a longer booking period (1 month) compared to employees (7 days), thereby enhancing the flexibility and management of desk allocation. This process ensures that desk bookings are handled efficiently and fairly, with clear verification steps to prevent conflicts.



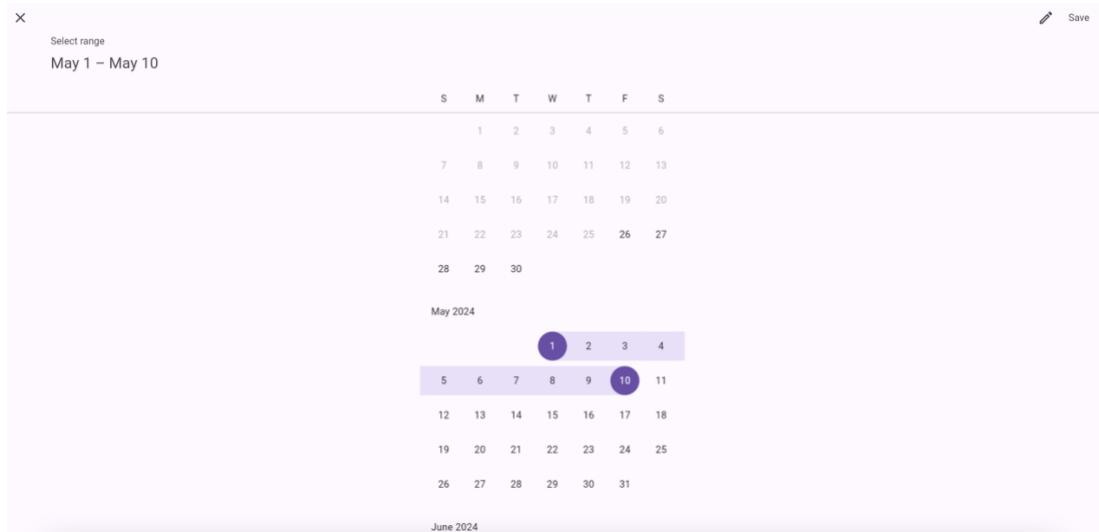
**Figure 5.6: Flowchart to login and book in the website**

Once the administrator lands on the home page, a dashboard showing some booking analytics is displayed. Four boxes are displayed at the top, listing total bookings, changed bookings, daily average bookings, and the number of desks was enable to use. The key charts included in the system are a column chart representing desk utilization and two pie charts highlighting desk seat location picker preference. Desk utilization shows the date range on the y-axis in the column chart, depending on the length of the custom date range. The other two pie charts are used to see which desk is the most "disliked" by employees, and employees are always willing to change from this "disliked" desk to a new desk. The remaining charts are the most popular and are replaced with new desks.



**Figure 5.7: Admin dashboard page in website**

Once click on the Data Range, a calendar will show out to let the administrators could custom the graph. The selected date is between the date of first booking until the current date.



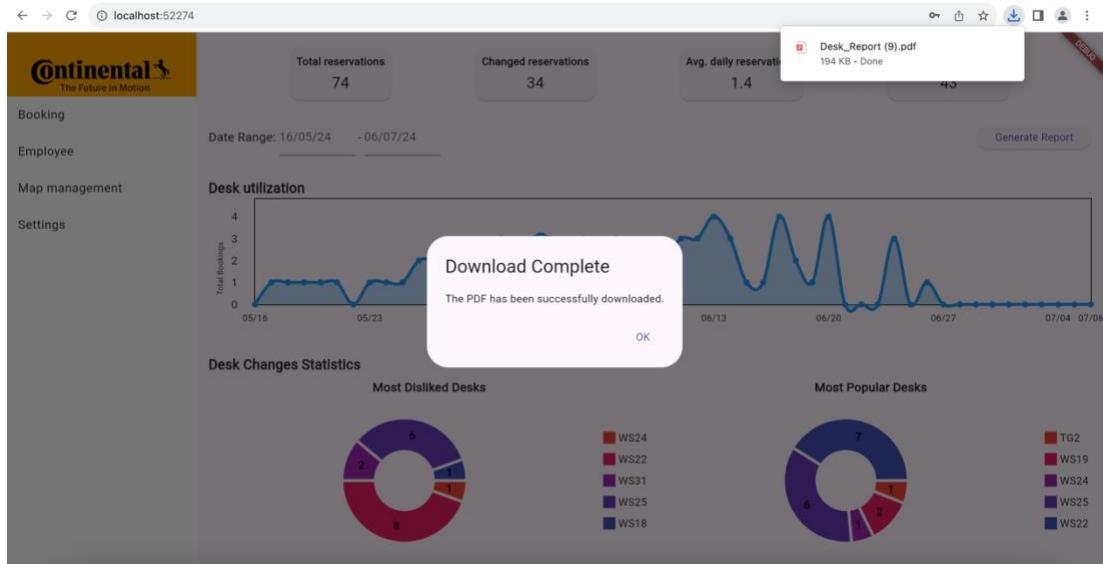
**Figure 5.8: Admin dashboard calendar in website**

When clicking on the generate report button, a PDF file will be downloaded. This PDF will display the same information that is shown on the dashboard.



**Figure 5.9: Admin generate report in website**

It will take few second to download the PDF file on the browser. Then it will show the alert message once the PDF file is download successful. Then the administrators could click on the PDF to check with the report.

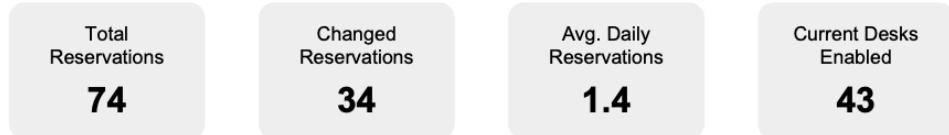


**Figure 5.10: Admin report download success in website**

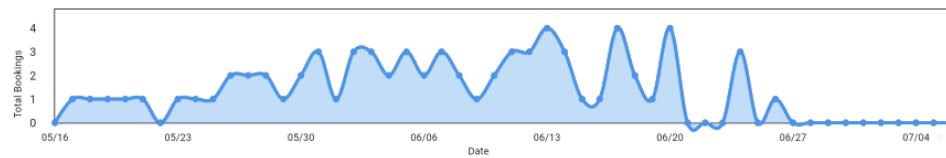
Figure 5.11 shows an overview of the PDF contents, exactly as in the dashboard. The only difference is that the title will show as Analysis Report and then the date range will display on the first line.

## Analysis Report

Date Range: 16/05/24 - 06/07/24



### Desk Utilization

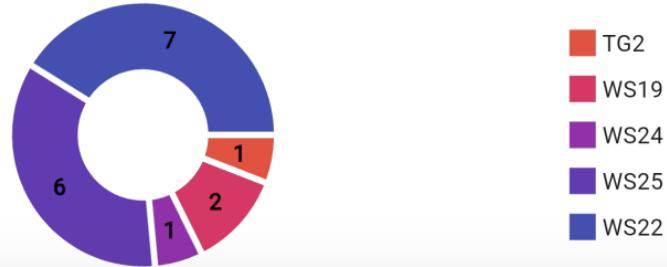


### Desk Change Statistics

Most Disliked Desks:



Most Popular Desks:



**Figure 5.11: Admin report download success in website**

When the administrator go to the booking page through the navigation bar, all of the booking detail will show out. Then administrator can choose view by list or view by map to had a better visualize or flexibility while manage the booking. In the view by list boking page, the employees could edit, delete and add new desk.

No.	Employee ID	Booking Date	Desk ID	Booking Status	Actions
1	EMP04	2024-06-16	WS8	scheduled	
2	EMP04	2024-06-15	WS1	scheduled	
3	EMP04	2024-06-14	TG8	scheduled	
4	EMP03	2024-06-14	WS28	scheduled	
5	EMP03	2024-06-13	WS13	scheduled	
6	EMP04	2024-06-13	WS12	checked in	
7	EMP06	2024-06-13	WS3	scheduled	
8	EMP05	2024-06-12	TG9	scheduled	
9	EMP03	2024-06-12	TG7	scheduled	
10	EMP04	2024-06-11	WS26	scheduled	
11	EMP05	2024-06-11	WS31	scheduled	
12	EMP03	2024-06-11	+ 2	scheduled	

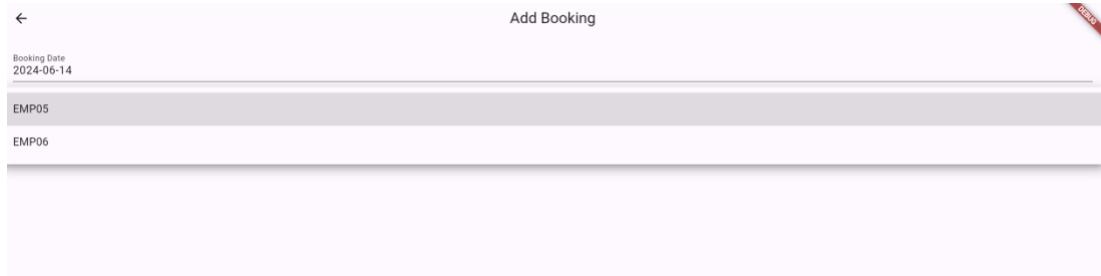
**Figure 5.12: Admin view booking list page in website**

For adding the new booking ,the administrator need to enter the date range which can book by one month in advance. Then the booking status will automatic save as scheduled once the booking generator .Then the desk ID was only allow to choose the current available desk and the employee ID was also require to fill up to make this booking success.

The screenshot shows the 'Add Booking' interface. It includes fields for 'Booking Date' and 'Employee ID', and a 'Save' button. A modal window titled 'Select date' is open, displaying a calendar for June 2024. The date 'Jun 13' is highlighted with a blue circle. The calendar grid shows the days of the week from Sunday to Saturday. Buttons for 'Cancel' and 'OK' are at the bottom right of the modal.

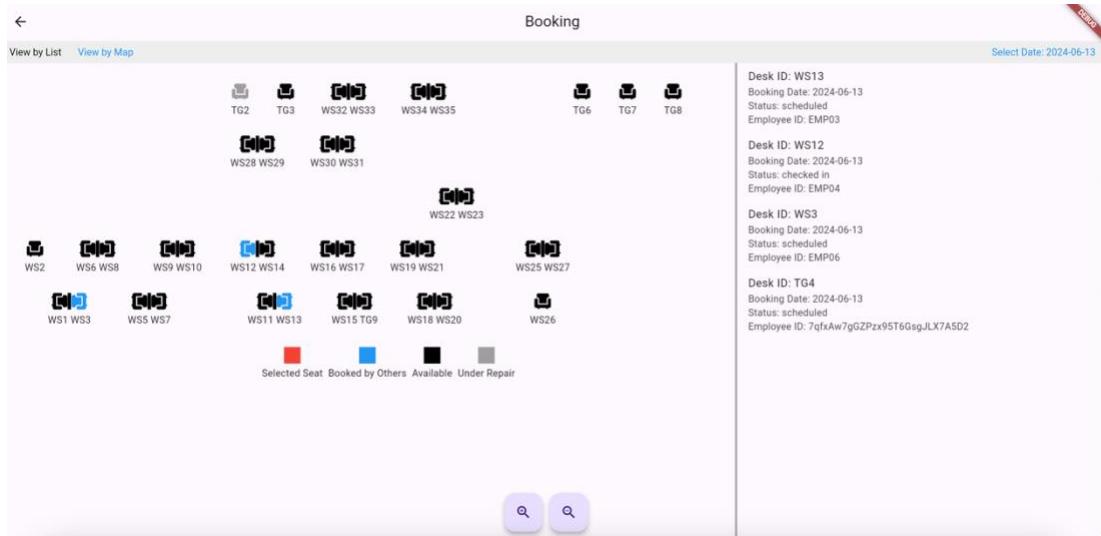
**Figure 5.13: Admin add booking page in website**

Once the date had been chosen, then the available employee will be list down by a drop-down list. It is to ensure that no duplicate booking will be make for the same employee in the same date.

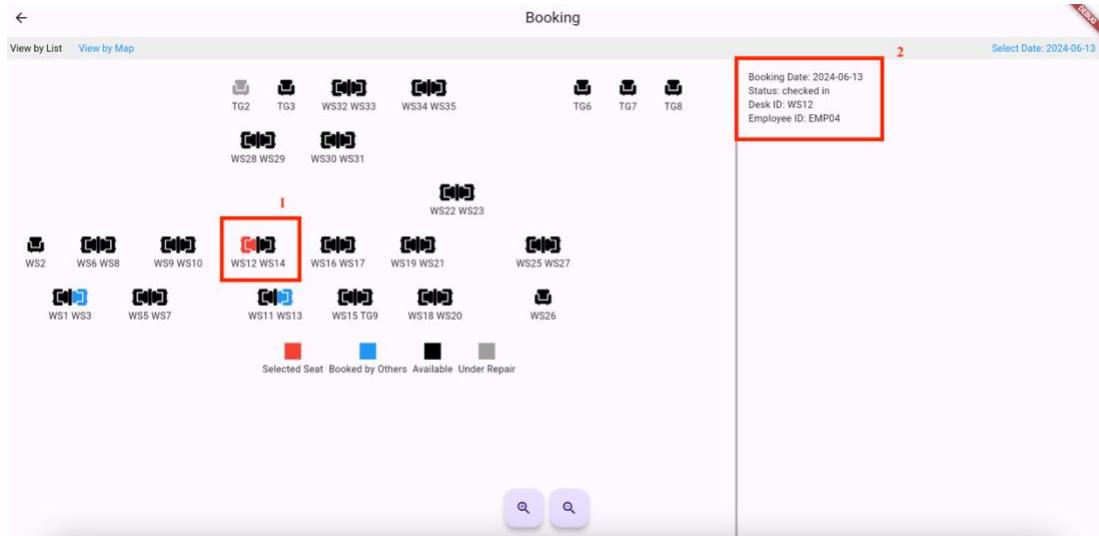


**Figure 5.14: Admin add booking of show available employee in website**

While the administrator click on the view by map ,the seat map will print out on the left side and the right side is the booking detail that booked by the selected date. Because the screen might be too small to see the whole map ,so the zoom in and out button is get ready below the map. The administrator also allows scrolling to the right side using the mouse touch pad when the right side of the map is invisible. Administrator can also select a desk on the map to view its booking details as shown in Figure 5.15.

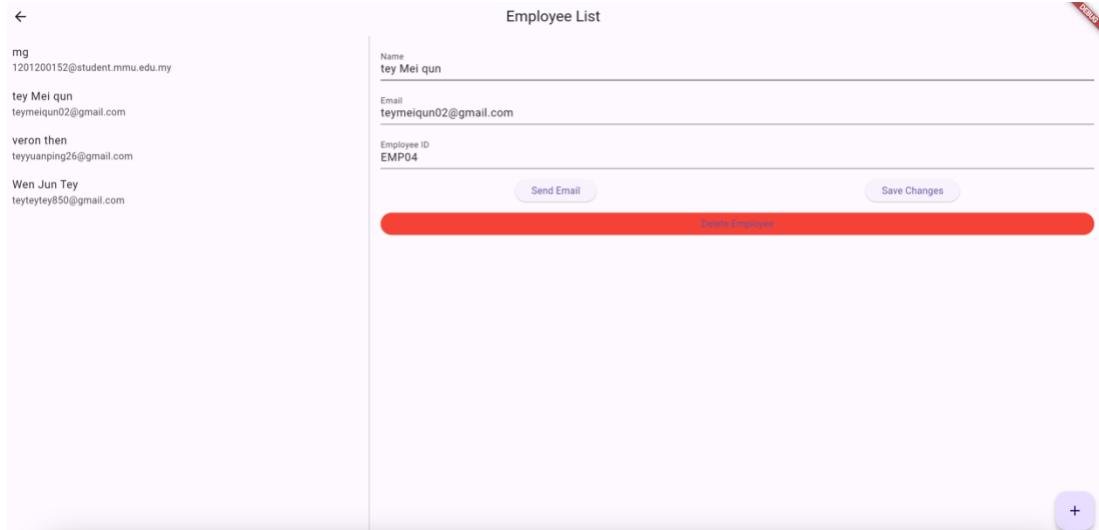


**Figure 5.15: Admin view by map page in website**



**Figure 5.16: Admin view by map of desk detail in website**

If the administrator choose to the employee in the navigation bar, the employee list will be show out. All of the employee name, email and employee ID will be show by a list, then administrator could able to edit, delete and add employee. The send email button will generate and send the QR code for the employee to login on the application.



**Figure 5.17: Admin employee page in website**

After entering the map management, the left side is the map and the right side is the desk details, as shown in Figure 5.18. After clicking the banned seat button in Figure 5.19, the color of the desk will turn grey, as shown in the map below, indicating that the desk is under maintenance.

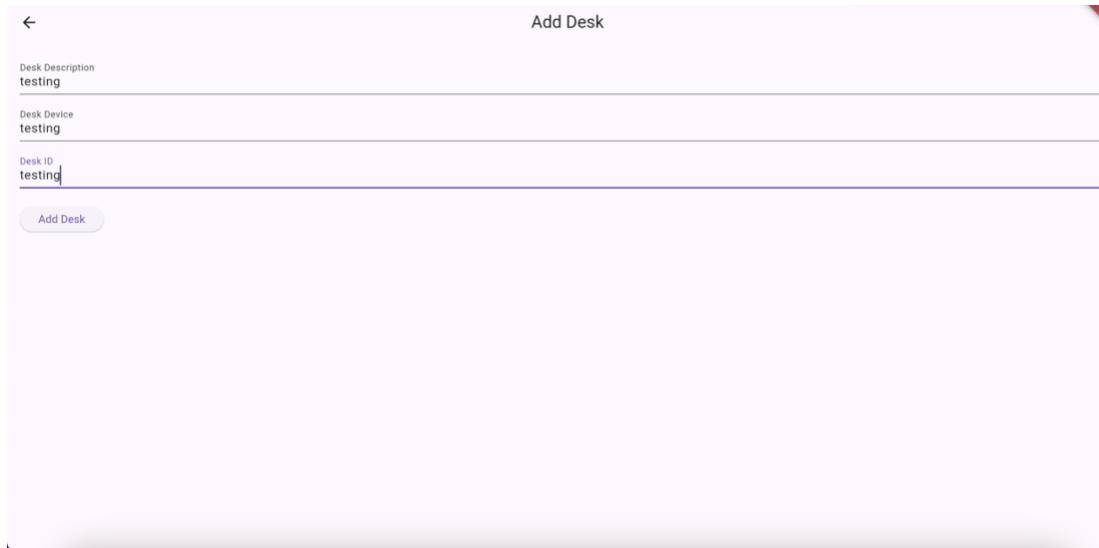
Desk ID	Device	ID	Status	Actions
WS2	Touch n Go	TG2	available	<input type="checkbox"/>
WS5 WS8	Touch n Go	TG3	available	<input type="checkbox"/>
WS9 WS10	Touch n Go	TG4	available	<input type="checkbox"/>
WS12 WS14	Touch n Go	TG5	available	<input type="checkbox"/>
WS16 WS17	Touch n Go	TG6	available	<input type="checkbox"/>
WS19 WS21	Touch n Go	TG7	available	<input type="checkbox"/>
WS22 WS23	Touch n Go	TG8	available	<input type="checkbox"/>
WS25 WS27				
WS1 WS3				
WS5 WS7				
WS11 WS13				
WS15 TG9				
WS18 WS20				
WS26				
WS24				

Figure 5.18: Admin map management page in website

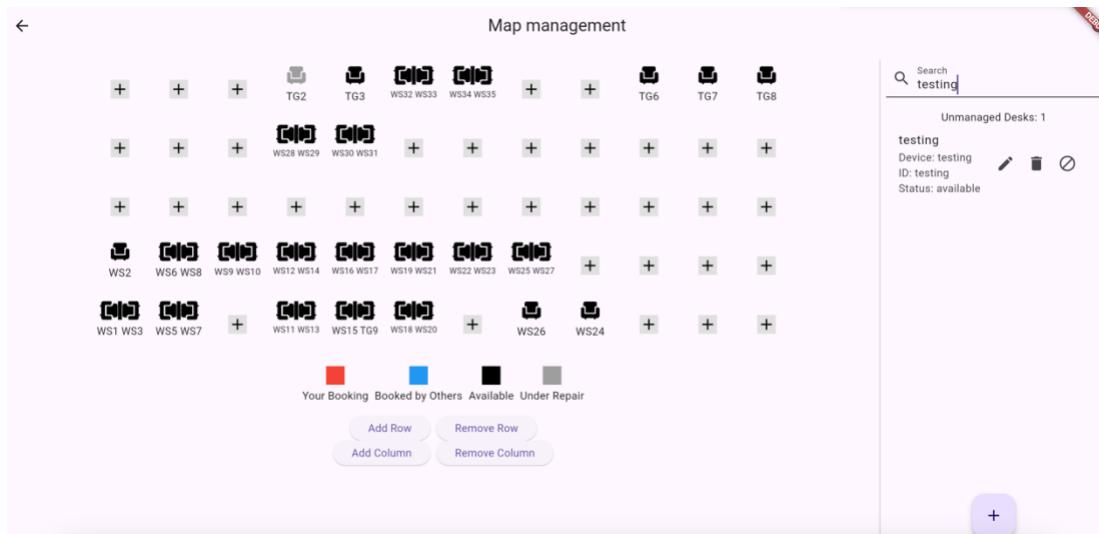
Desk ID	Device	ID	Status	Actions
WS2	Touch n Go	TG2	under repair	<input checked="" type="checkbox"/>
WS5 WS8	Touch n Go	TG3	available	<input type="checkbox"/>
WS9 WS10	Touch n Go	TG4	available	<input type="checkbox"/>
WS12 WS14	Touch n Go	TG5	under repair	<input checked="" type="checkbox"/>
WS16 WS17	Touch n Go	TG6	available	<input type="checkbox"/>
WS19 WS21	Touch n Go	TG7	available	<input type="checkbox"/>
WS22 WS23	Touch n Go	TG8	available	<input type="checkbox"/>
WS25 WS27				
WS1 WS3				
WS5 WS7				
WS11 WS13				
WS15 TG9				
WS18 WS20				
WS26				
WS24				

Figure 5.19: Admin block seat in map management page in website

After clicking the add desk button, the administrator needs to enter the desk description, device, and desk ID to add the desk. Then the administrator can search by looking up the desk's keywords to quickly have a review of the desk information like Figure 5.20.

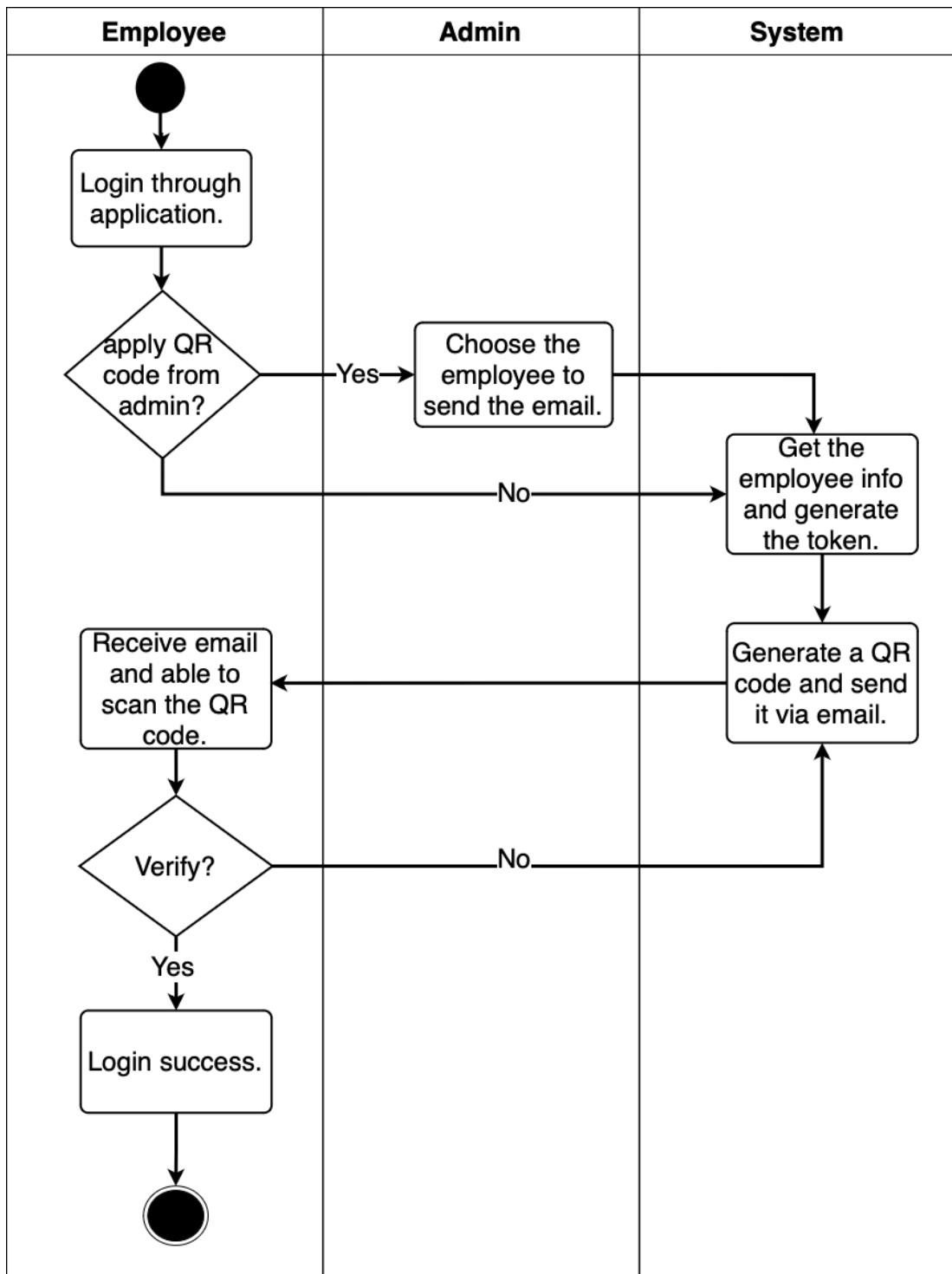


**Figure 5.20: Admin add desk page in website**



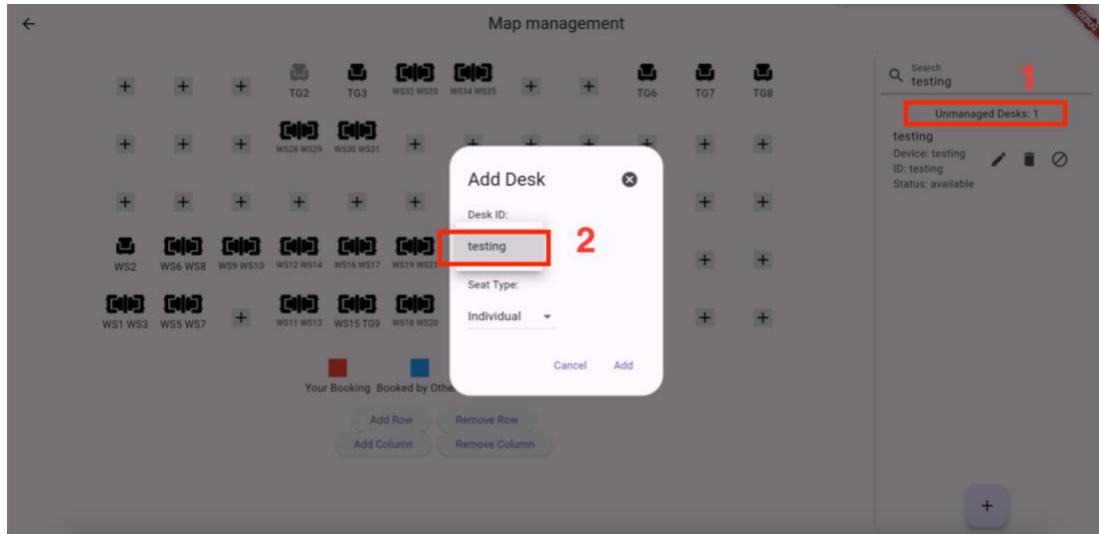
**Figure 5.21: Admin search function on map management page in website**

The employee can choose to obtain the QR code from the administrator or directly get from the system. It takes a while for the administrator to send an email to the employee, and then the alternative plan where the employee can get the QR code directly from the system is more time-saving.

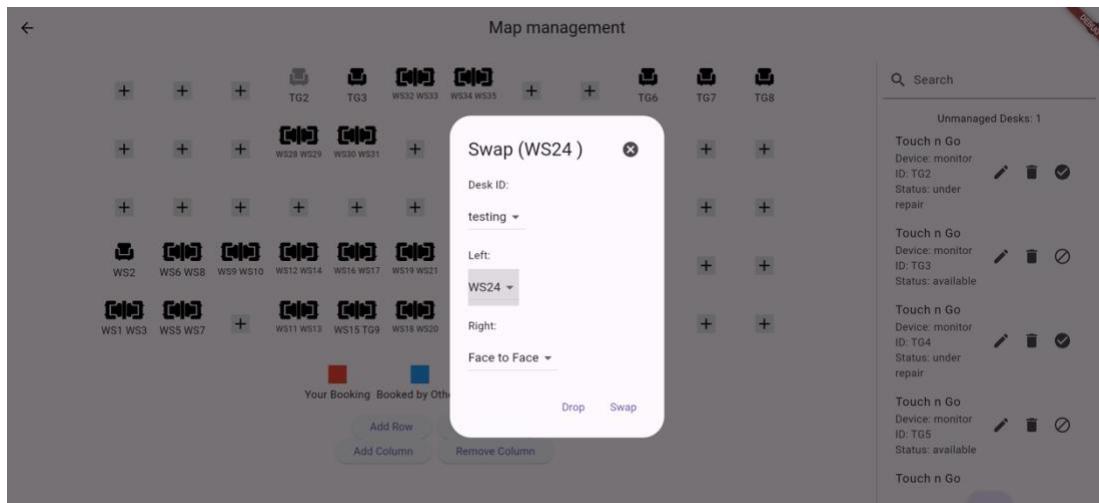


**Figure 5.22: Flow chart for Employee QR Login process**

Once the desk has been successfully added, the number of unmanaged desks will increase. The newly added desk should then be able to be arranged on the map by simply click the add icon on the map to place the seat at that location. There is two seat type could be selected, which is individual desk and face to face desk If the seat location is misplaced, the administrator can click back the desk location use the drop button to remove the desk in that location. The swap function also allows changing the desk ID of the selected location if other desks are still unmanaged into seat.

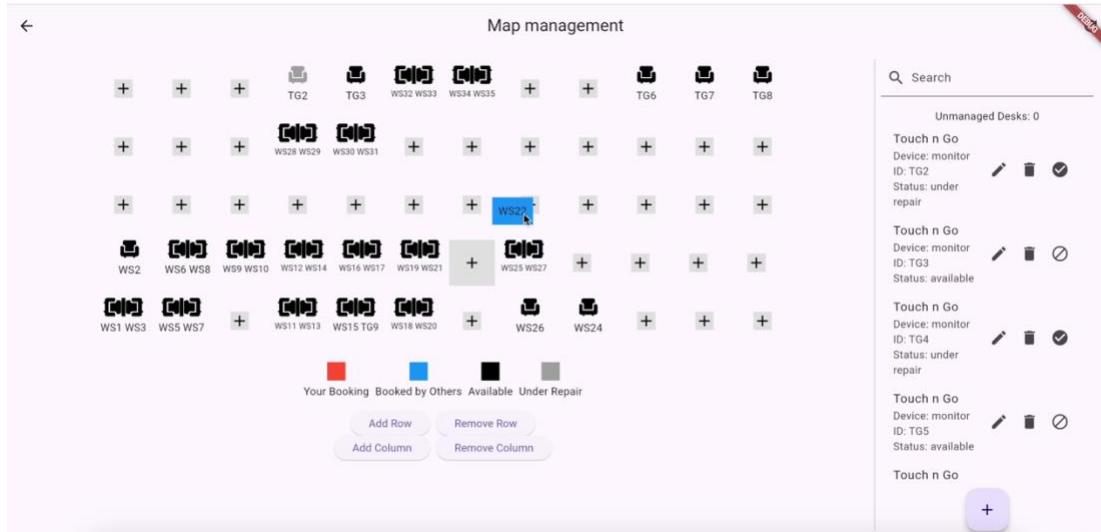


**Figure 5.23:** Admin add desk on map function in website

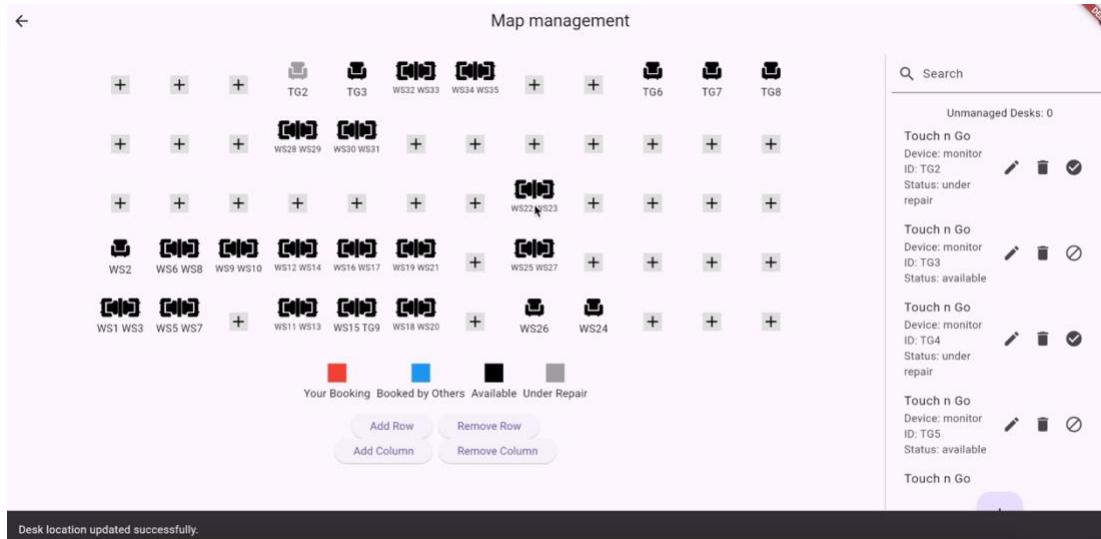


**Figure 5.24:** Admin add or swap desk on map function in website

When drag the desk ,it could be replace and update into other location .The Figure 5.25 was demonstrate the dragging action and the Figure 5.26 show as the result that the WS 22 and WS 23 updated the location successfully.

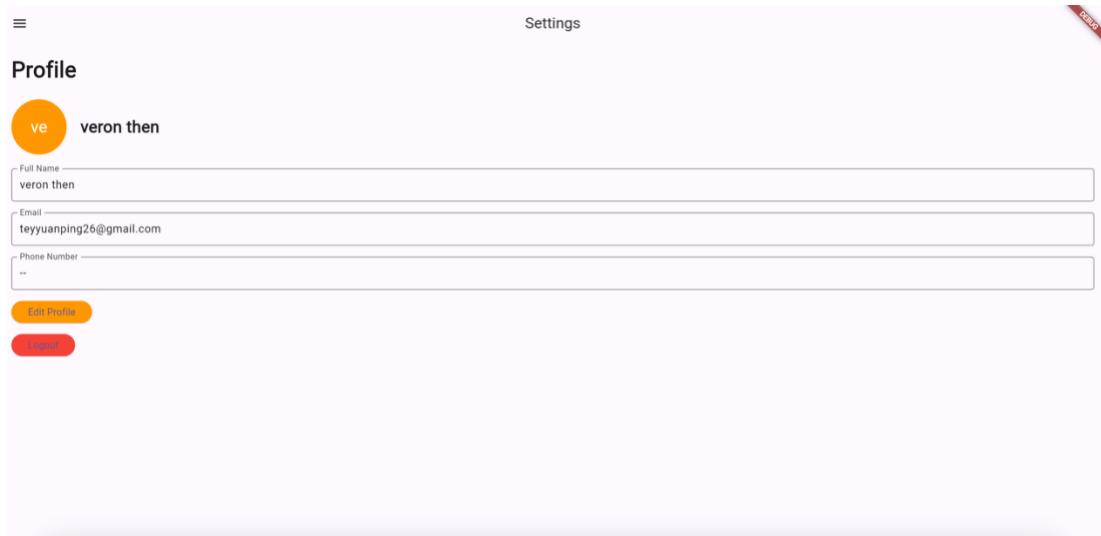


**Figure 5.25: Admin drag to change seat location in website**



**Figure 5.26: Admin successful to change seat location in website**

After entering the setting page, the admin profile will be shown. The admin name, email and phone number will be display. The edit profile function is allowed for this page and once click on log out button, the admin can exit from the system.



**Figure 5.27: Admin profile setting in website**

### 5.3 Employee Interface

In this proposed system, the employee module is mainly used on the application but employees also allow to use the website. Employees can directly schedule by click on the date list which only show as 7 days to fulfil the project requirement. While the booking make successfully as shown as Figure 5.29, the employee enable to unschedule the booking by click the unschedule button. Then when the mouse hovers over the listing, the desk ID will be displayed, or the employee can click on the listing to view a seating map.

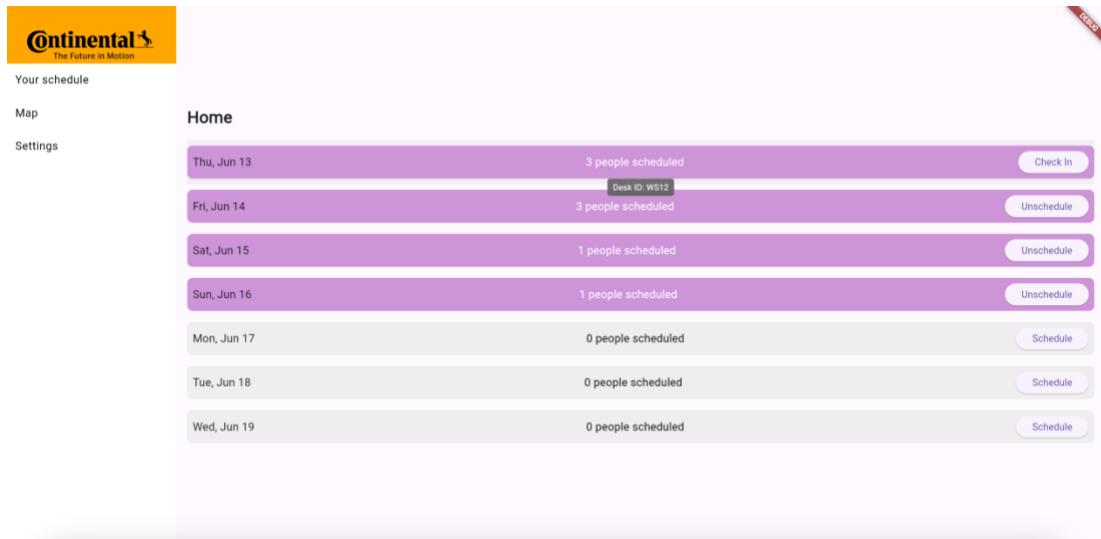


Figure 5.28: Employee home screen in website

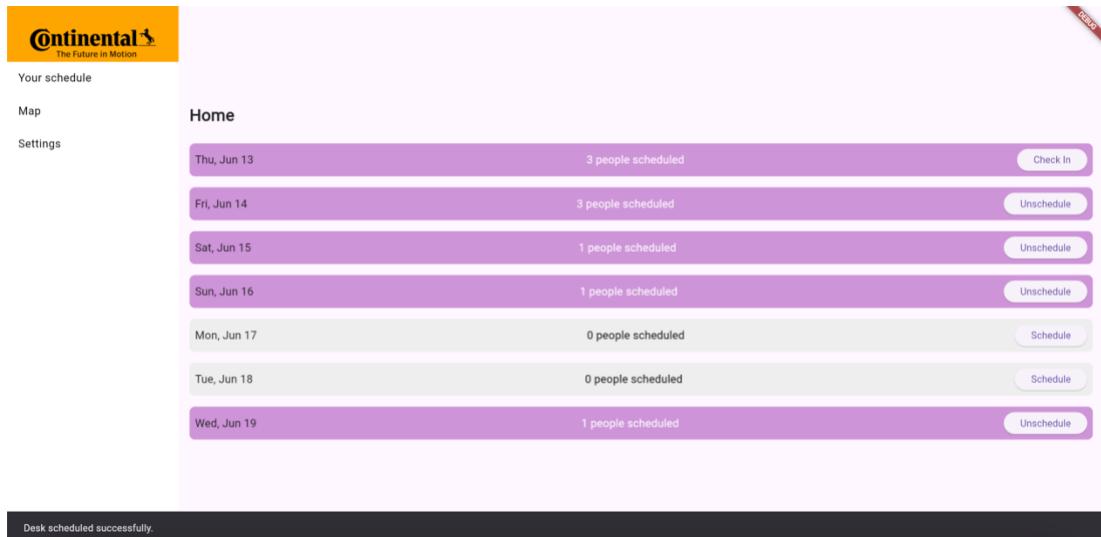
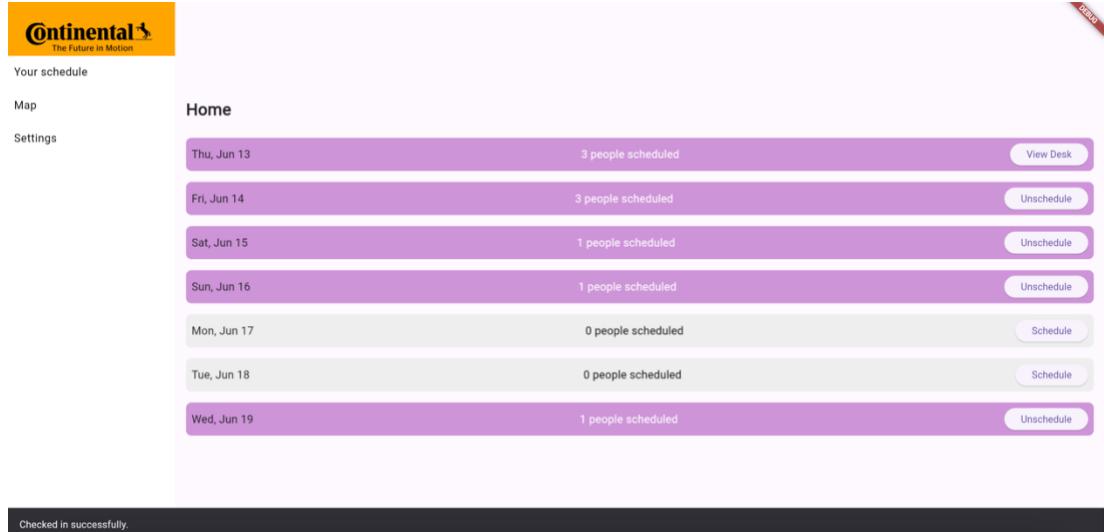


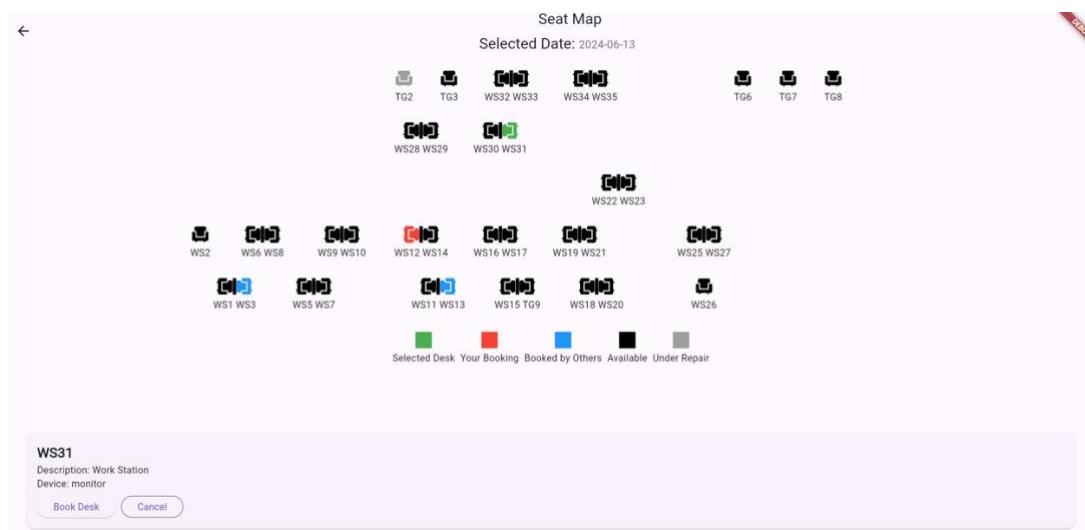
Figure 5.29: Employee successful booked desk in website

Once an employee schedule for today's date, it will ask the employee to check in upon arrival at the company. After check-in, the check in button will become the view desk, so the employee could also click on that or click on the map navigation to see the seat map.



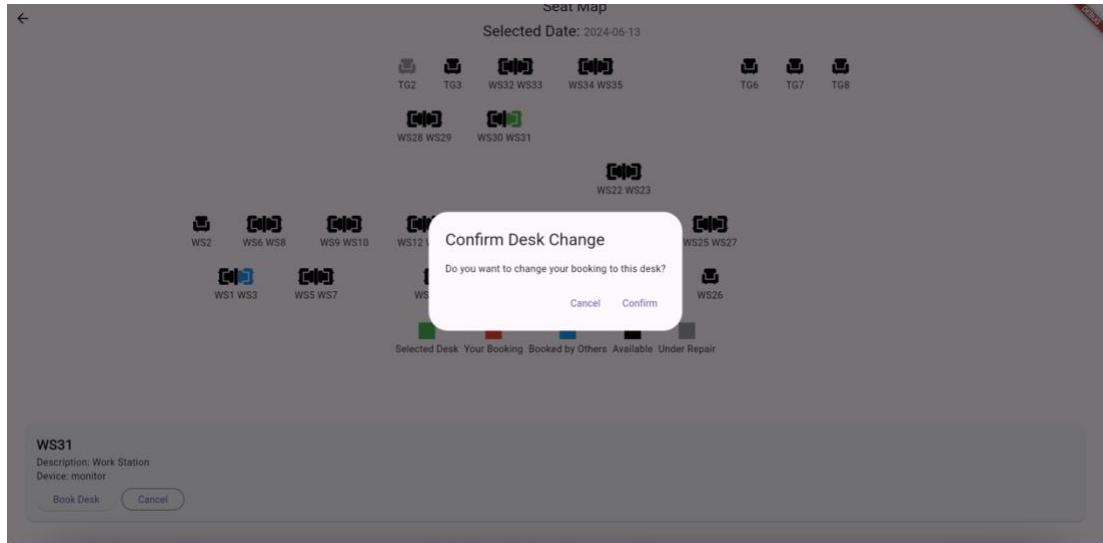
**Figure 5.30: Employee successful check-in desk in website**

When viewing the seat map, desks reserved by employees are shown in red, desks booked by others are shown in blue, and the currently selected desk is indicated in green to display the information below. Alternatively, employees can retrieve this map from the navigation bar by clicking on the map and change the selected date to see other date booking status.

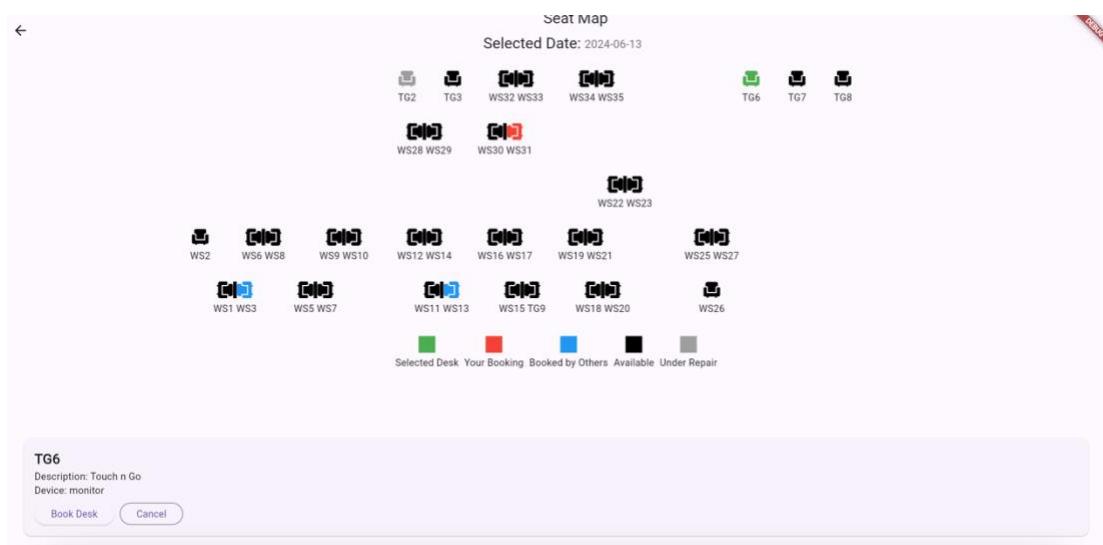


**Figure 5.31: Employee seat map screen in website**

Employees can change desks immediately by clicking the book desk button in the desk information text field. The progress show on the Figure 5.32 and Figure 5.33.

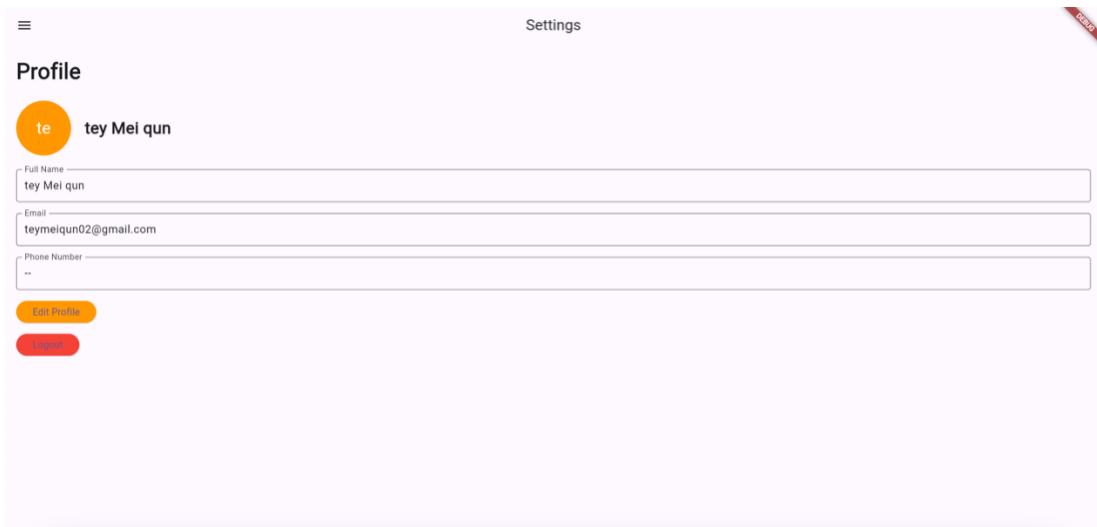


**Figure 5.32: Employee change seat confirmation in website**



**Figure 5.33: Employee successful change seat in website**

The employee profile settings page is generally the same as the administrator side. There will also display the employee's basic information like name, email and phone number. Employees can also exit from this page.



**Figure 5.34: Employee setting profile page in website**

### 5.3.1 Mobile application

The landed page for the application will consist of a welcome speech and an email address login request. If the email address is verified correctly, the employee can then choose the login method, whether it is via email password or QR code.

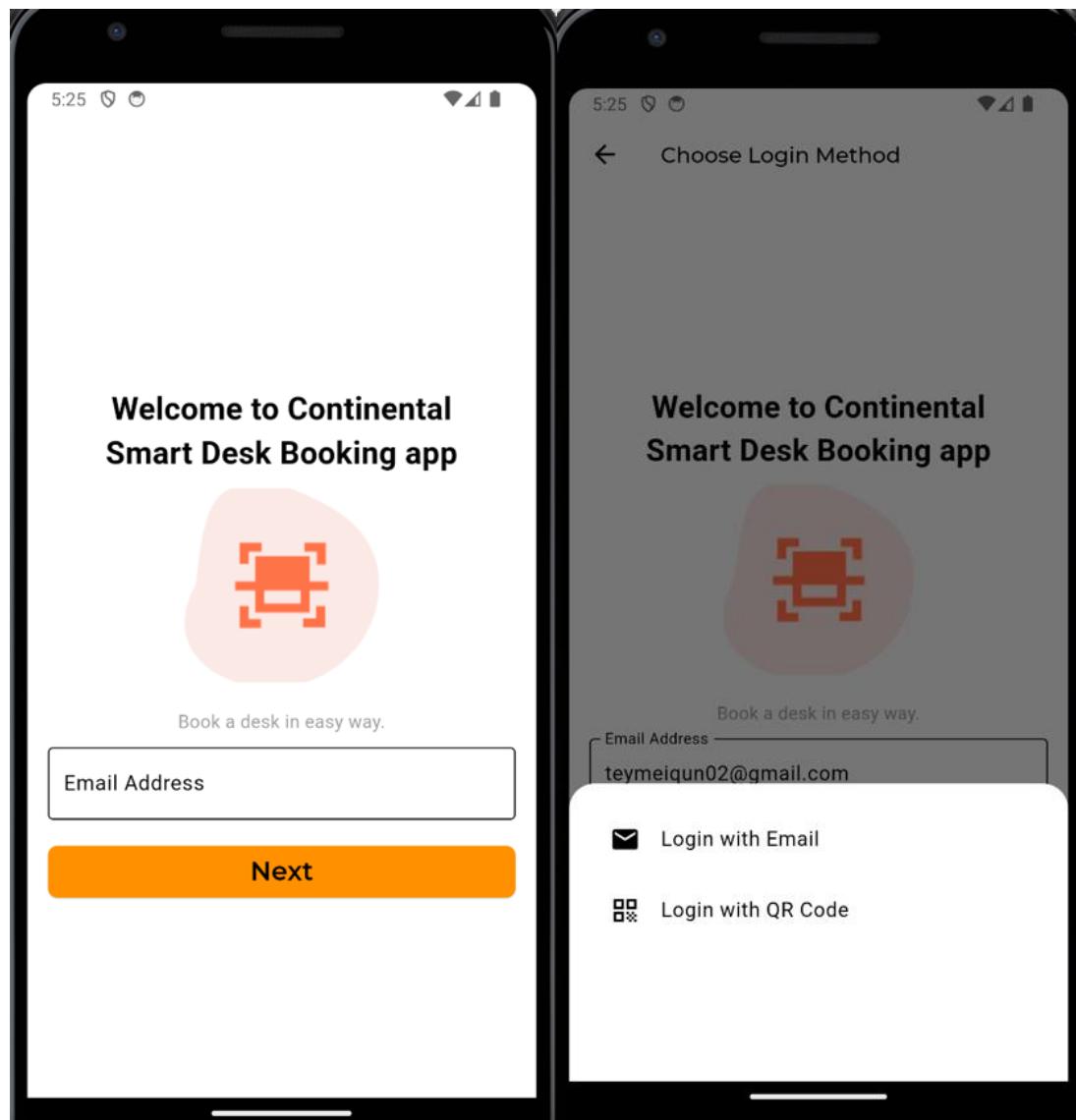
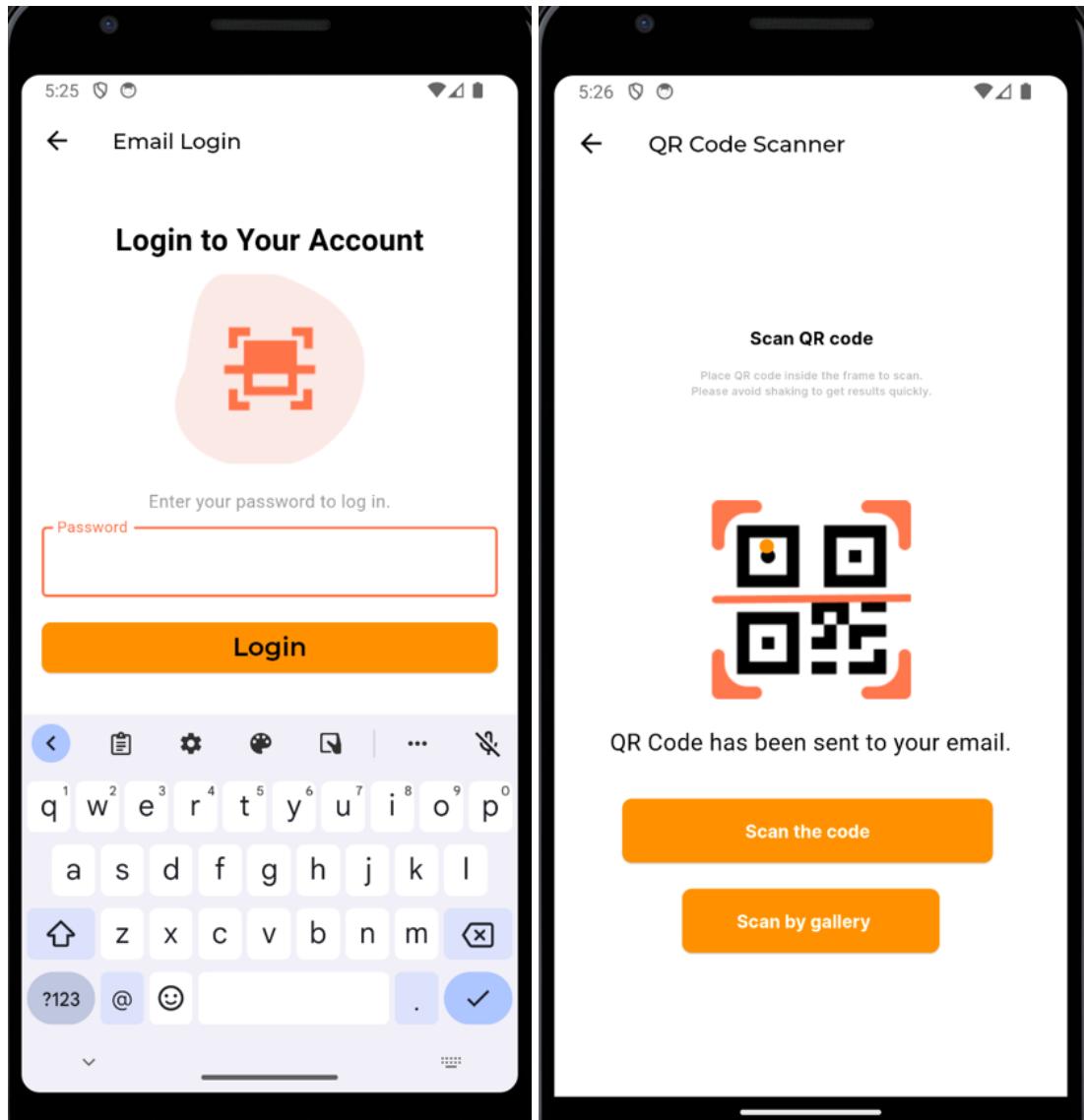


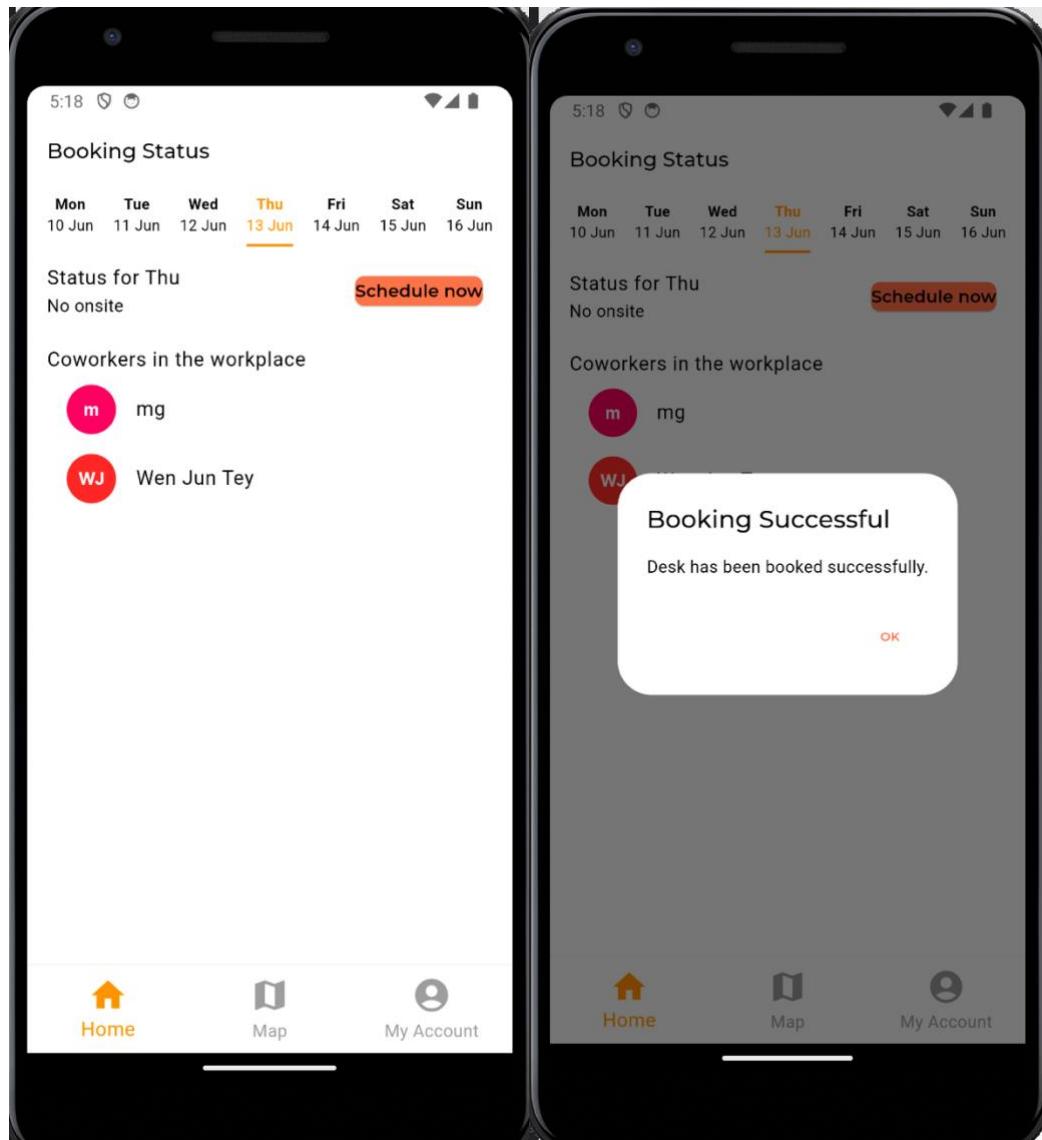
Figure 5.35: Employee login page in app

If the employee chooses to log in using an email password, then only the password need to enter for authentication. If they choose to log in using a QR code, the employee needs to wait for an email which containing a QR code, then the system will scan to decode and ask an API request to check if the QR code is valid.



**Figure 5.36: Employee login process in app**

Once the employee login successfully, it will retrieve to the home page as shown as below. The day and date of the week will be displayed in the top as a list. The employee can select and view the booking status. A list of other colleagues scheduled for the same day will also be displayed. Then the user could able to schedule now, the alert message will be show out when the booking is successful.



**Figure 5.37: Employee home page and booking process in app**

Then the employee could view the schedule by tapping on the button that appears, or click on the bottom navigation to navigate to the seat map screen. The seat map is synchronized with the website. Once the location of the desk is modified, all seat maps will be updated.

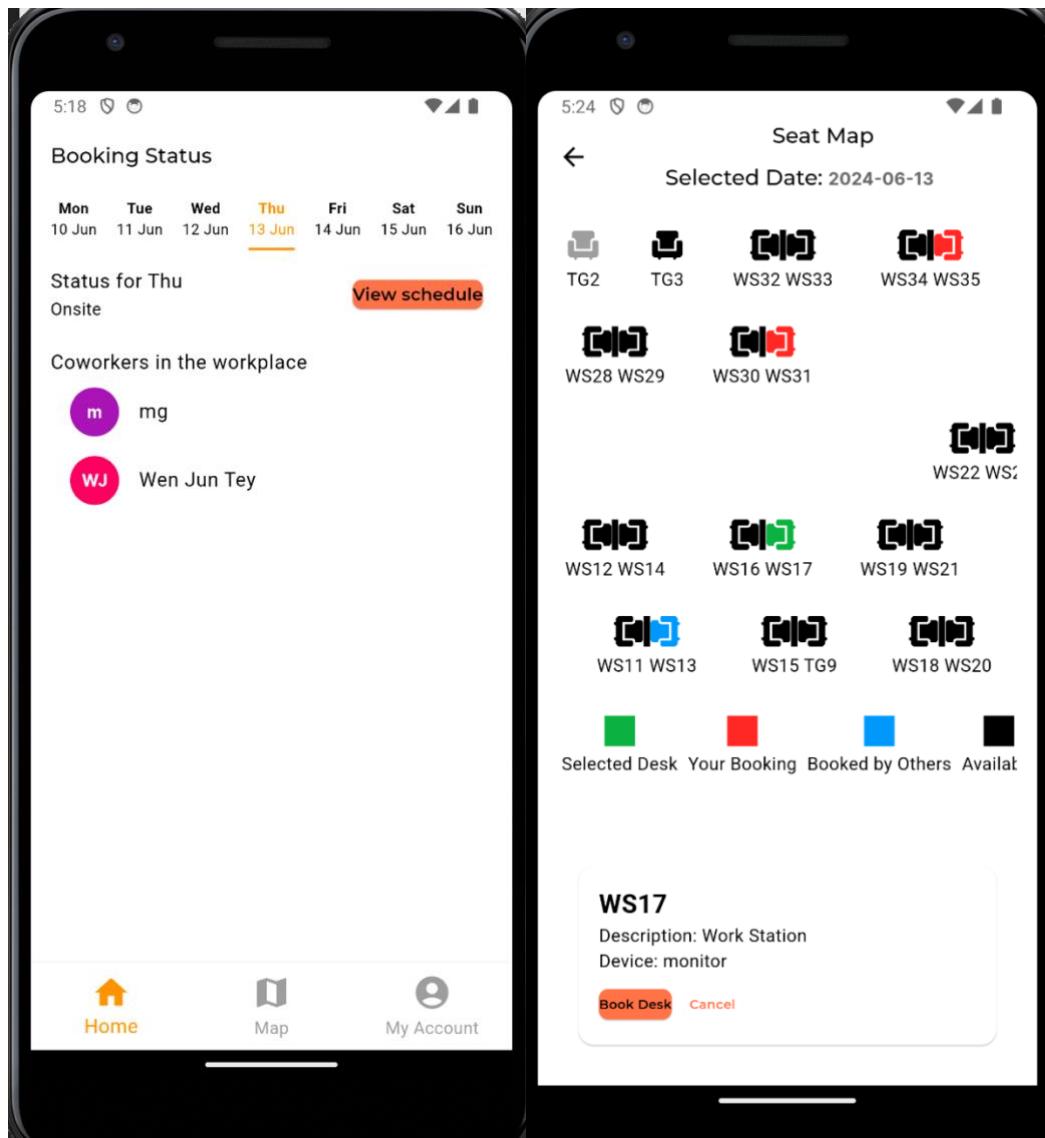
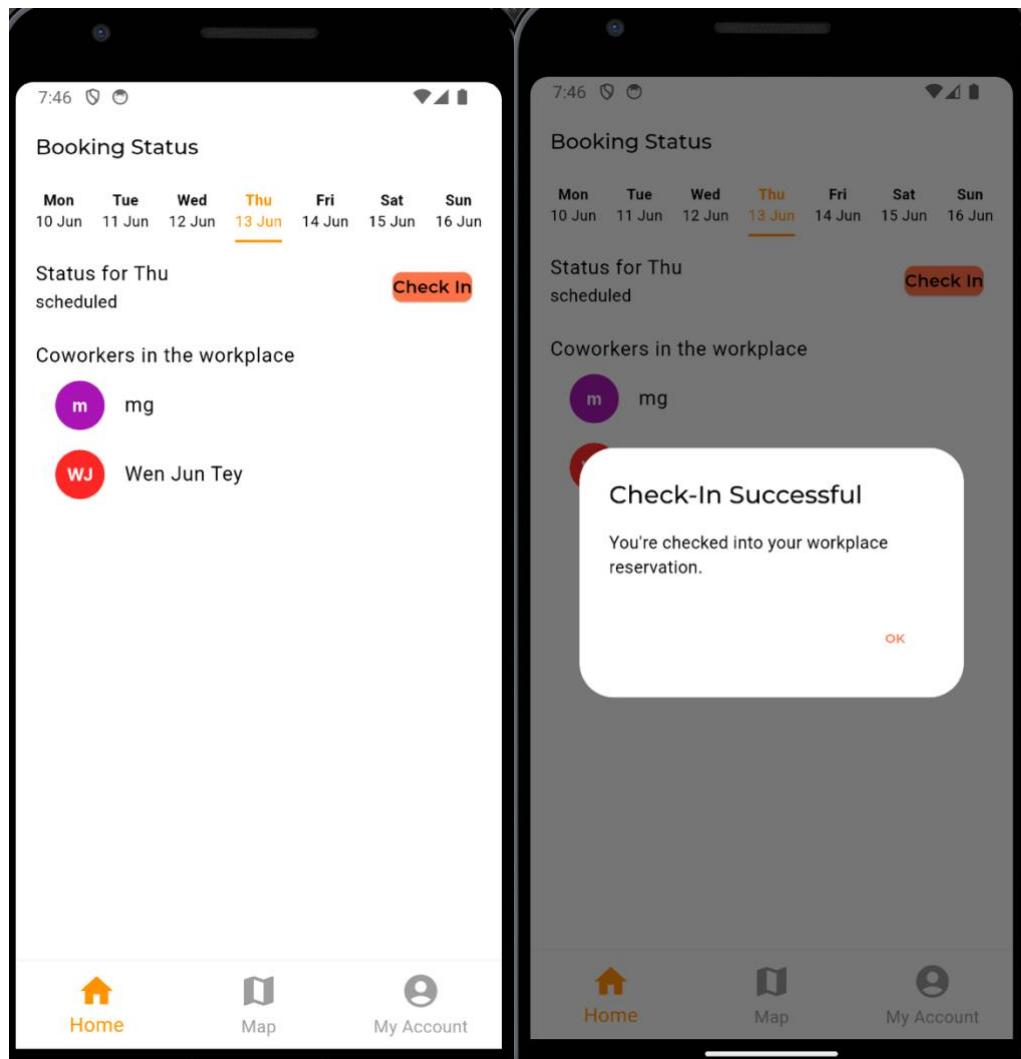


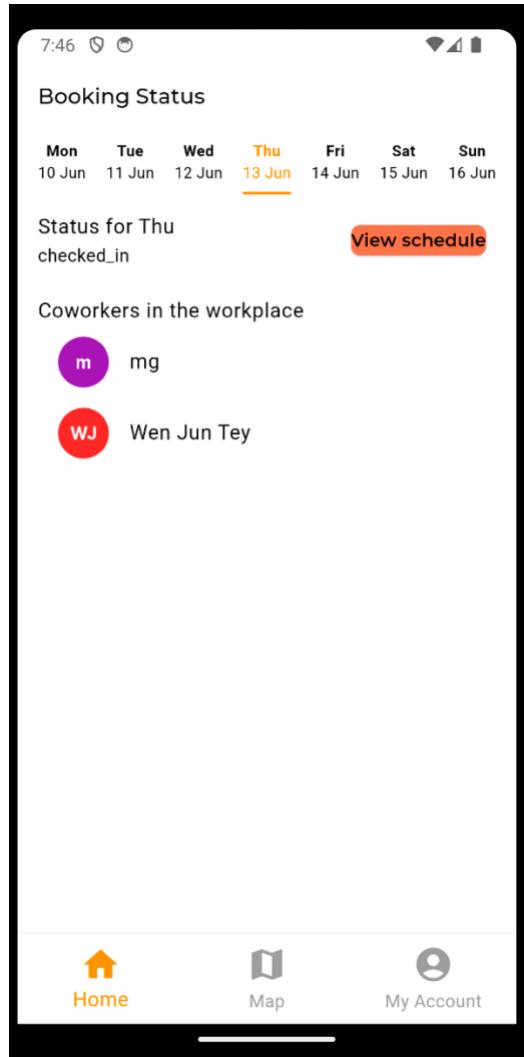
Figure 5.38: Employee view seat map process in app

When the employee was schedule by today, the system will ask to check in. This check in is to find out whether the employee is on site in the company or absent .While click on the check in button, it will prompt out that the check in is successful.



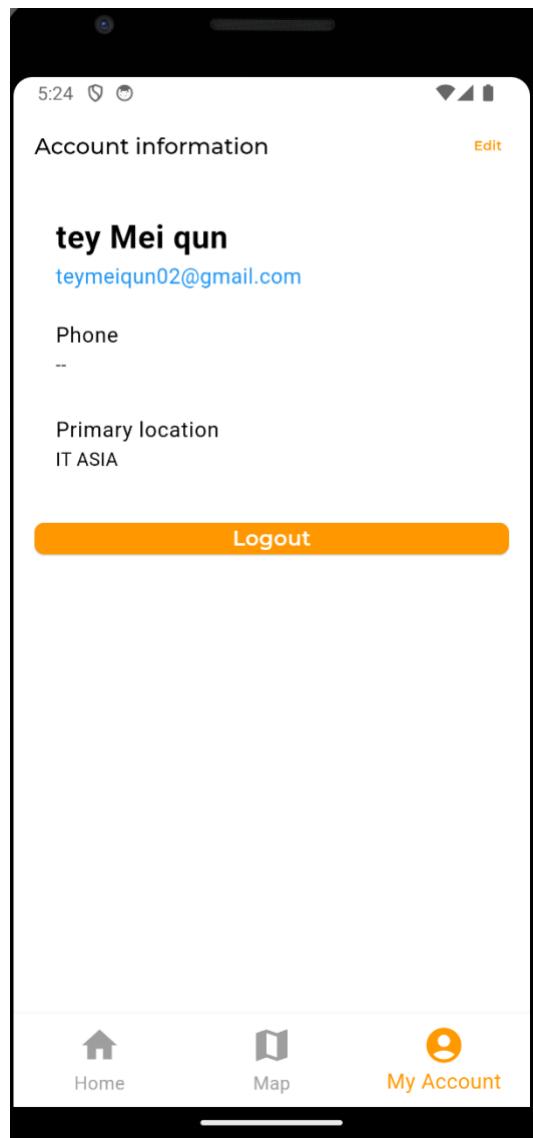
**Figure 5.39: Employee checkin process in app**

After check in, the status for the days will show as check in to prove that the employee is in the workspace already. Check in is to let the other party know that the employee has arrived at the company so that it is easier for managers to notice who is absent.



**Figure 5.40:** Employee check in successful process in app

Finally, employees can review or edit the information on this profile screen by navigating to the "My Account" bar. The log out button was also provided to exit.



**Figure 5.41:** Employee profile page in app

## **5.4 Challenges Encountered**

In the initial setup, the database was changed from AWS Amplify to the current Firebase due to connection failures on AWS. During the period of not connecting to the database, the main objectives are hard to integrate the mobile application and website. QR code verification has also stopped development for a long time which cause the whole system implementation progress is completely slower than the estimated time on the previously planned Gantt chart.

After solving the database connection problem, new API connection problems emerged. In firebase, the API must go through cloud functions, which means functions have to be written logically to handle a specific request. To achieve this, the tested and failed are repeated, the npm (Node package manager) and eslint were also initialized repeatedly to identify and fix problems in the JavaScript code.

The random selectors generated in the proposed system are used to predict the employee preference for choosing a desk. Python and Flutter (which uses Dart) are different programming languages with different runtime environments. Calling Python code directly from Dart is not straightforward. The final solution was to use a backend service written in Python and communicate with it through HTTP requests, which was achieved by creating a REST API using the Python web framework Flask.

## 5.5 Variation of the Project Objective

**Table 5.1: Summary of Objective Variation Table**

<b>Original Objective</b>	<b>Current Objective</b>
<p>To provide automated seat allocation management via web and mobile applications.</p> <ul style="list-style-type: none"> <li>The seat booking process can be streamlined through a hybrid model of web and app by automatic seat allocation to provide the preferred desks for employee.</li> </ul>	<p>To provide random seat allocation management via web and mobile applications.</p> <ul style="list-style-type: none"> <li>The seat booking process will through a random seat allocation which is driven by chance of available desks.</li> <li>The desk randomly picker will be done and follow by the employee preference for the random desk prediction.</li> </ul>
<p>To provide flexibility for administrators to book desks for longer periods of time beyond a month.</p> <ul style="list-style-type: none"> <li>Only administrators are allowed to manage desk allocations for extended periods beyond one month.</li> </ul>	<p>To provide flexibility for administrators to book desks for longer periods of time beyond a month and the ability to manage employees' bookings.</p> <ul style="list-style-type: none"> <li>Only administrators are allowed to manage desk allocations for extended periods beyond one month.</li> <li>Between one-month periods, administrators also able to help employee to make or manage the bookings.</li> </ul>
<p>To verify registration with QR code verification.</p> <ul style="list-style-type: none"> <li>To ensure a safe and efficient registration process, the</li> </ul>	<p>To verify registration with both QR code verification and email login.</p> <ul style="list-style-type: none"> <li>To ensure a safe and efficient registration process, the</li> </ul>

<p>proposed system uses QR code verification to log into the application.</p> <ul style="list-style-type: none"> <li>• The QR code will send by the administrators through the email for two-factor verification.</li> </ul>	<p>proposed system uses QR code verification to log into the application.</p> <ul style="list-style-type: none"> <li>• To provide a faster experience for the employees, the proposed system also allows employees to log in to the application through regular email login.</li> <li>• Employees can request a QR code directly from the system when they log in, rather than having to waiting the email to send from the admin.</li> </ul>
--	---

# CHAPTER 6

## TESTING EVALUATION

### 6.1 Login

The website is used by both employees and administrators, so only an email address and password are required as login identification. Figure 6.1 illustrates that the system does not permit logging in with any empty fields. Figure 6.2 shows that incorrect email formatting is also not eligible for acceptance.

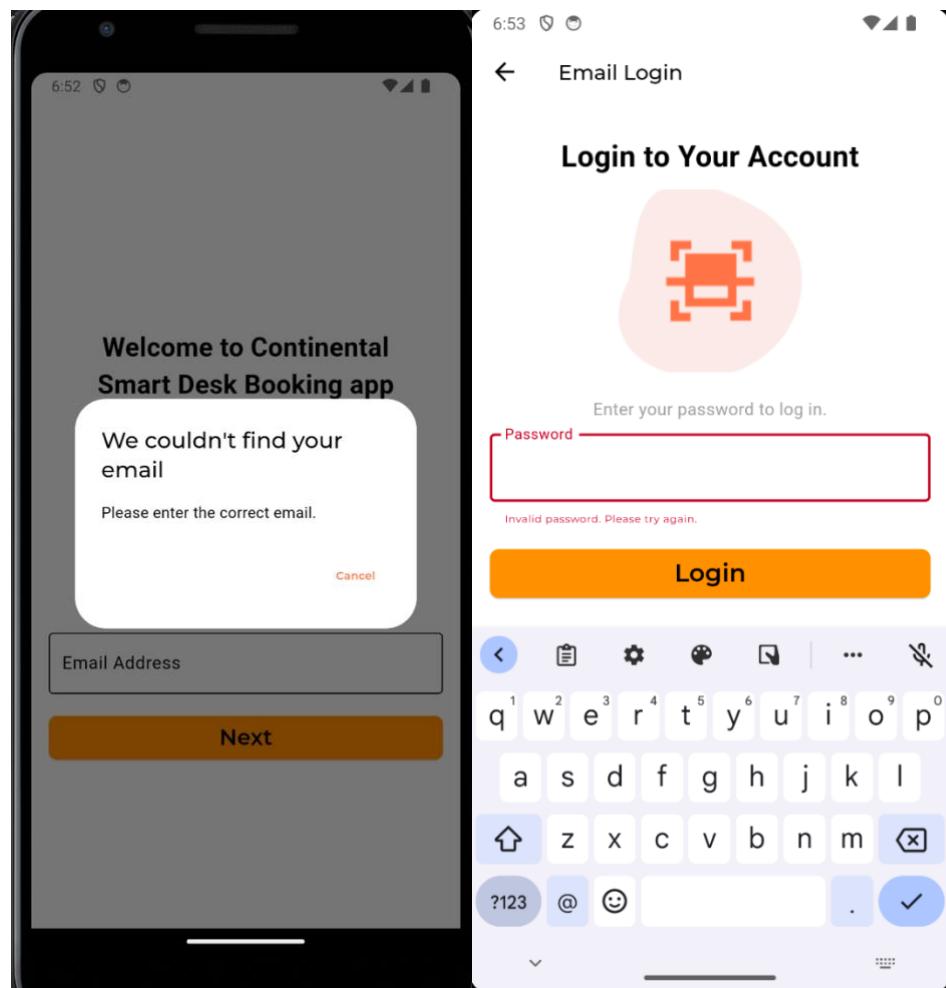
The screenshot shows the 'Desk Booking System' login page. At the top is the Continental logo. Below it is a yellow header bar with the text 'Desk Booking System'. The main form has two input fields: 'Email address' and 'Password', both of which are currently empty. Under each field is a red error message: 'Invalid email or password. Please try again.' Below the fields is a large orange 'Login' button. At the bottom of the form is a link 'Forgot your password?'

Figure 6.1: Login page with empty fields on website

The screenshot shows the 'Desk Booking System' login page. At the top is the Continental logo. Below it is a yellow header bar with the text 'Desk Booking System'. The main form has two input fields: 'Email address' and 'Password'. In the 'Email address' field, the user has typed 'tey', which is an invalid email format. In the 'Password' field, the user has typed '...', which is also invalid. Both fields have red outlines and red error messages below them: 'Invalid email or password. Please try again.'. Below the fields is a large orange 'Login' button. At the bottom of the form is a link 'Forgot your password?'

Figure 6.2: Login page with wrong email format on website

As shown in the figure6.3 below, the application will alerts the employee when the email and password entered is incorrect or empty. Only if the email address is entered correctly can let the employee proceed to the login screen to choose to log in by email or password. If the email is incorrect from the beginning, the employee will not be able to access the next screen.



**Figure 6.3: Login page with empty field on mobile**

## 6.2 View and Manage Booking list

Figure 6.4 is the administrator adding a booking form, showing that the form cannot be displayed by fields that have no data. Then it will issue an alert, please fill in all fields. Once the administrator fills in the only field for the booking date, it will also return an alert.

The screenshot shows a web-based booking form titled "Add Booking". At the top left is a back arrow icon, and at the top right is a red "X" button. Below the title are two input fields: "Booking Date" and "Employee ID", both of which are currently empty. A blue "Save" button is located below the Employee ID field. A black horizontal bar at the bottom of the form contains the text "Please fill all fields" in white.

**Figure 6.4: Add booking page with empty field on website**

This screenshot shows the same "Add Booking" page as Figure 6.4, but with one field populated. The "Booking Date" field now contains the value "2024-06-17". The "Employee ID" field remains empty. The blue "Save" button is still present. The black bar at the bottom of the form still displays the message "Please fill all fields".

**Figure 6.5: Add booking page with one empty field on website**

Figure 6.6 shows the same page for adding a booking, but with the same date, the 4 bookings was made. The test starts with the first booking made with the first employee ID "EMP03". The numbers shown in the figure represent the order of bookings from one to four. Therefore, once an employee has already made a booking for a certain day, that day will not be allowed and the employee ID will be displayed.

The figure consists of four vertically stacked screenshots of a web-based booking application. Each screenshot shows a form with a header 'Add Booking' and a back arrow. Below the header is a 'Booking Date' field set to '2024-06-17'. A dropdown menu lists employee IDs: EMP03, EMP04, EMP05, and EMP06. The employee ID selected in each screenshot is highlighted with a red border and a red number indicating the booking order: 1, 2, 3, and 4 respectively. In the bottom right corner of each screenshot, there is a 'Save' button.

**Figure 6.6: Added a booking four times on the website for the same booking date**

Figure 6.7 shows the view bookings list when view by list. In this page it could able to search by a selected date and also could search by other keyword to show the related bookings list.

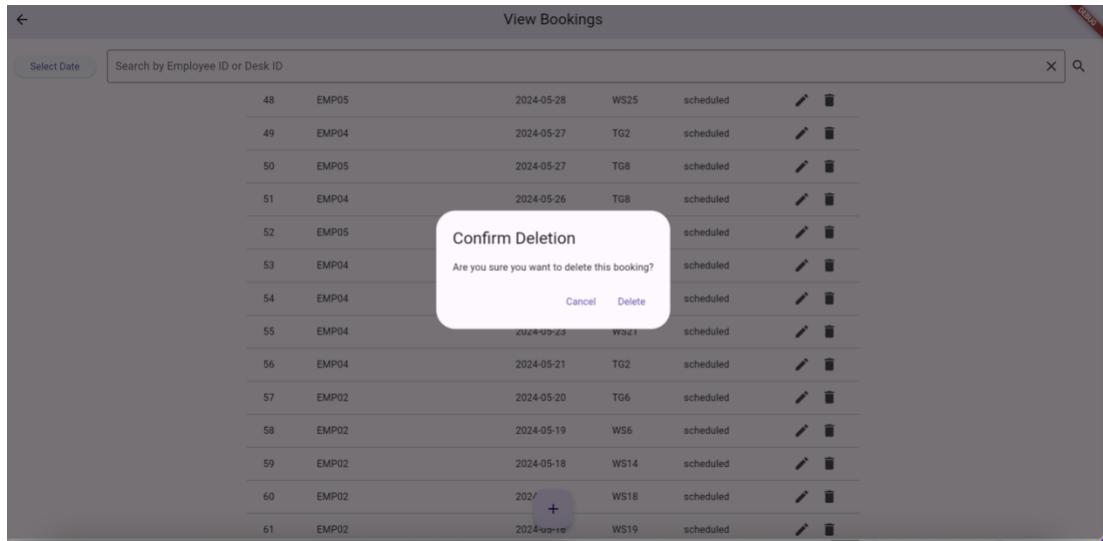
The screenshot displays a web application titled "View Bookings". At the top left is a back arrow icon, and at the top right is a red "REFRESH" button. Below the title is a search bar containing the placeholder "Search by Employee ID or Desk ID" with a clear "X" icon and a magnifying glass icon. To the right of the search bar is a date selector showing "2024-06-04". The main content area is a table with the following data:

No.	Employee ID	Booking Date	Desk ID	Booking Status	Actions
1	EMP05	2024-06-04	WS31	scheduled	
2	EMP03	2024-06-04	WS27	scheduled	

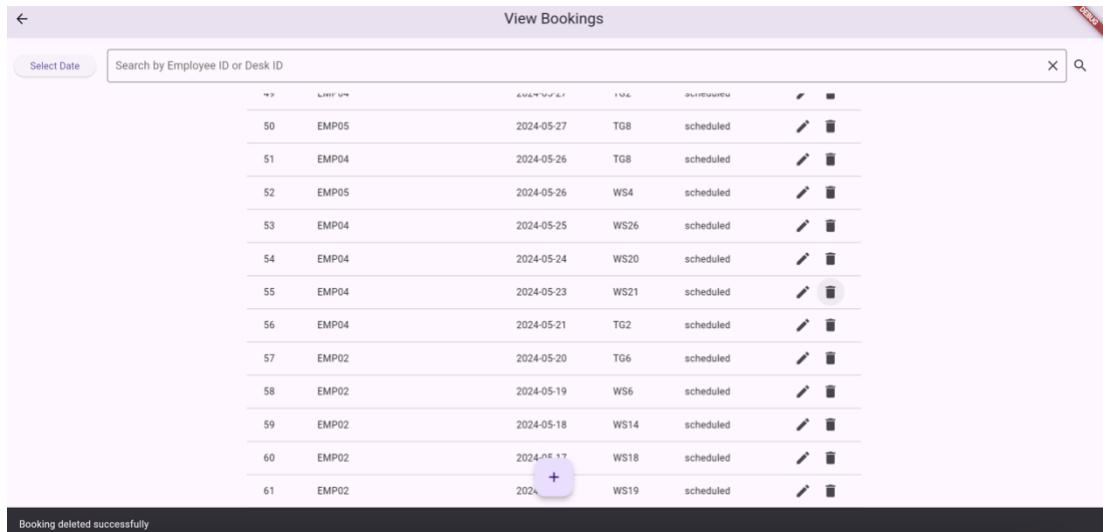
At the bottom center of the table is a small purple button with a white plus sign (+). The background of the page is light pink.

**Figure 6.7: Added a booking four times on the website for the same booking date**

Figure 6.8 shows the alert message of the delete function, which asking the administrator to confirm the deletion of the record. Once the deletion is confirmed, the bookings will be permanently deleted.



**Figure 6.8: Delete confirmation alert on the website**



**Figure 6.9: Deleted successful on the website**

Figure 6.10 shows four tests that test whether a null value exists on the booking text field and whether it can be changed on the edit page. It turns out that once there is a null value, it cannot be update the edit booking form. Figure 6.11 shows the original text field being submitted with a warning message stating that no changes have occurred.

The figure consists of four separate screenshots of an 'Edit Booking' form, each showing a different field that is empty or contains an invalid value, resulting in a validation error message.

- Screenshot 1:** Shows the 'Booking Date' field with the value '2024-06-04'. Below the form, a horizontal bar displays the error message 'Please fill all fields'.
- Screenshot 2:** Shows the 'Booking Status' field with the value 'scheduled'. Below the form, a horizontal bar displays the error message 'Please fill all fields'.
- Screenshot 3:** Shows the 'Desk ID' field with the value 'WS31'. Below the form, a horizontal bar displays the error message 'Please fill all fields'.
- Screenshot 4:** Shows the 'Employee ID' field with the value 'EMP05'. Below the form, a horizontal bar displays the error message 'Please fill all fields'.

Each screenshot includes a back arrow at the top left, a save button labeled 'Save Changes' at the bottom left, and a back arrow at the top right.

**Figure 6.10: Edit booking validation testing on the website**

This screenshot shows the 'Edit Booking' form with all fields populated with their original values: 'Booking Date' (2024-06-04), 'Booking Status' (scheduled), 'Desk ID' (WS31), and 'Employee ID' (EMP05). A save button labeled 'Save Changes' is visible at the bottom left.

Below the form, a horizontal bar displays the message 'No changes made'.

**Figure 6.11: Edit booking save the same value on the website**

### 6.3 Customizing the date range for generating reports

Figure 6.12 is the date range calendar on the dashboard which will impact to the desk utilization graph and also the report. The dates selected in the graph below start from May 23rd and end on May 31st. Once applied, the graph will be updated immediately. Figure 6.13 shows the graph according to the selected date.

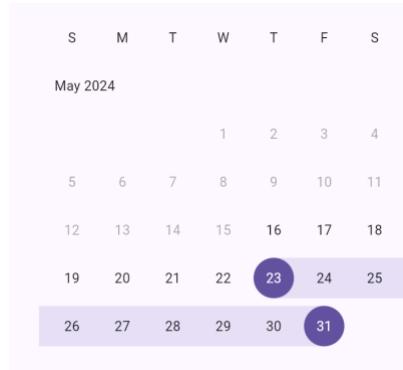


Figure 6.12: Date range calendar for report on the website

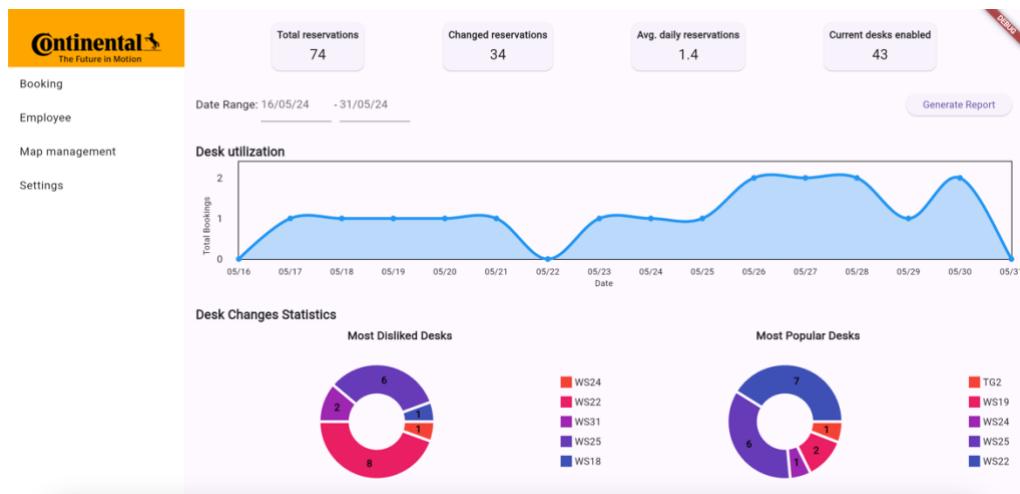


Figure 6.13: Dashboard page on the website

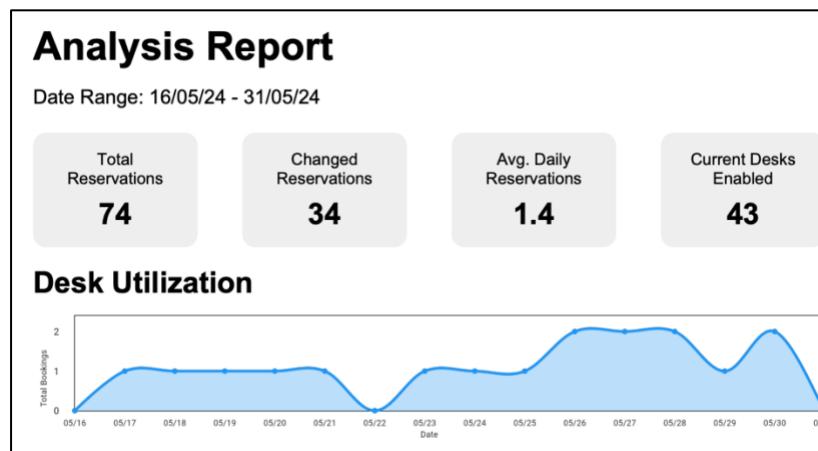


Figure 6.14: Part of the downloaded report content

## 6.4 Map management

Both Figure 6.14 and Figure 6.15 is showing the alert message that cannot delete rows or columns of existing tables. This is a limitation to ensure that the map can be displayed completely.

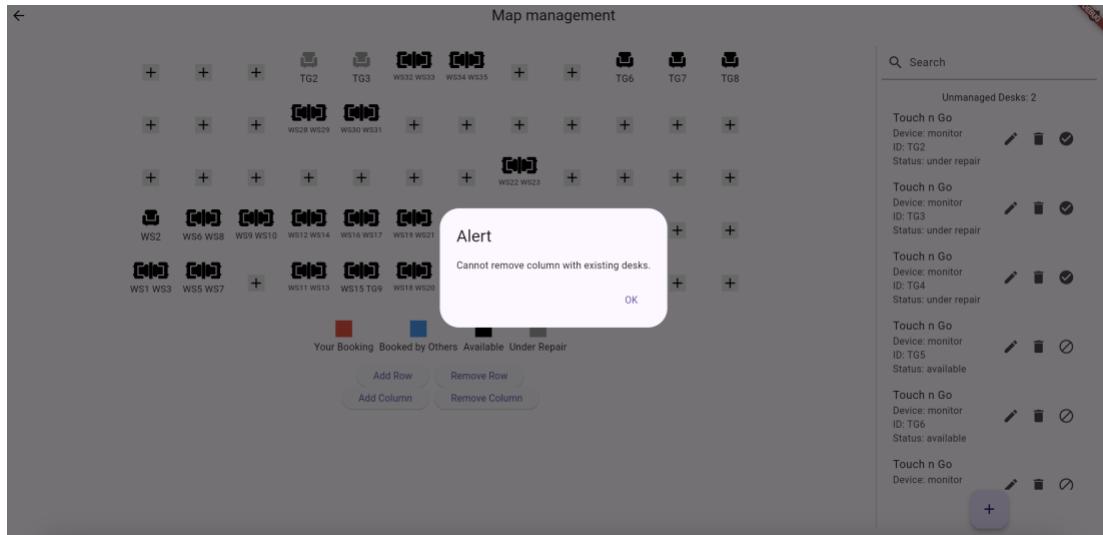


Figure 6.13: Map management alert cannot remove column

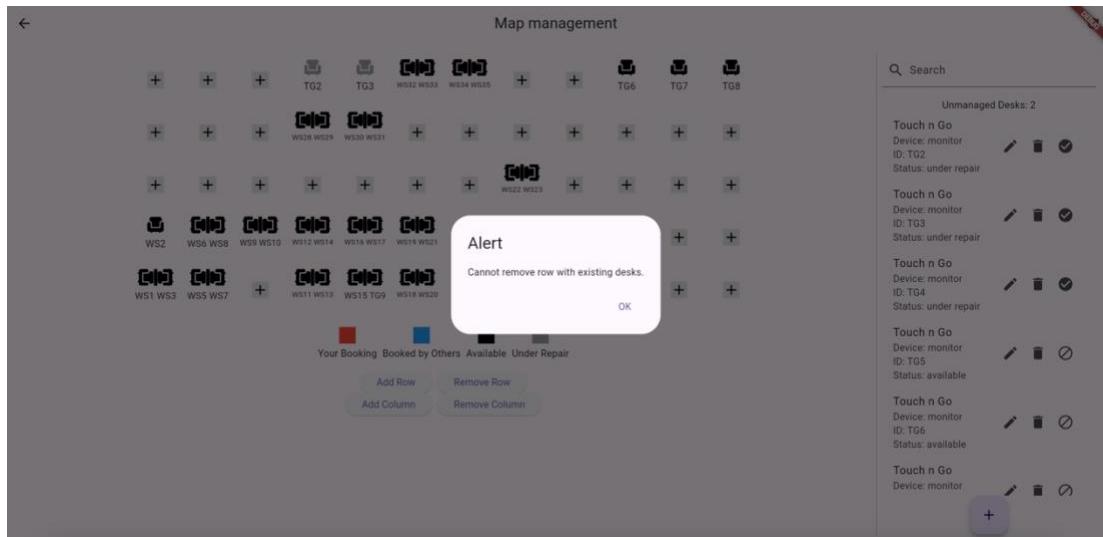
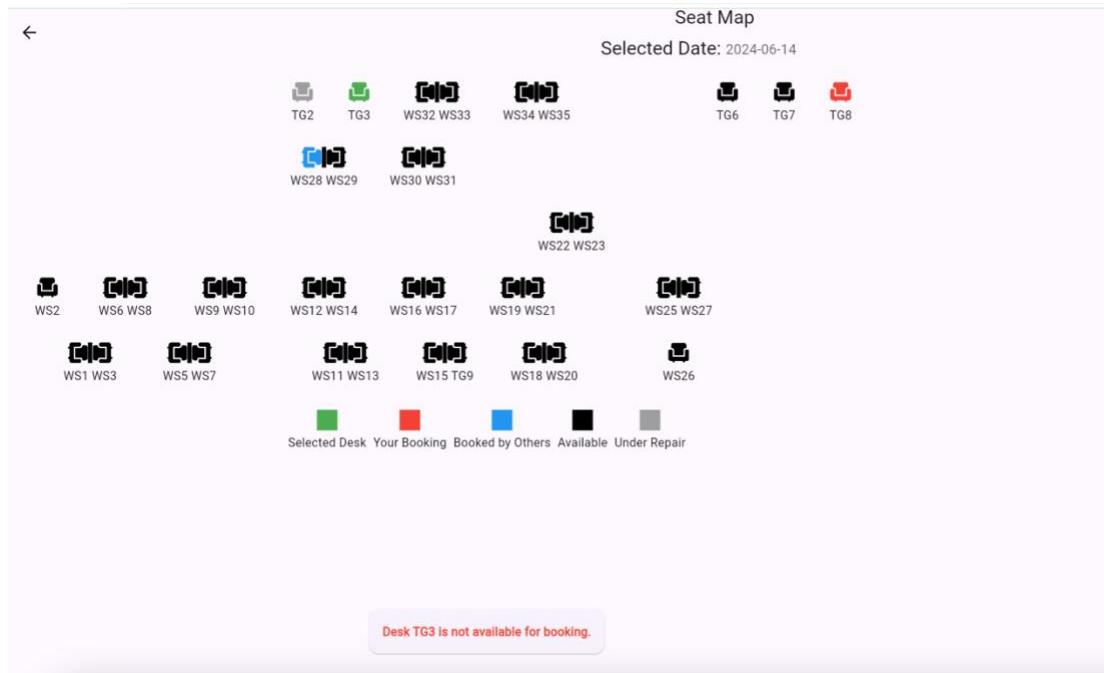


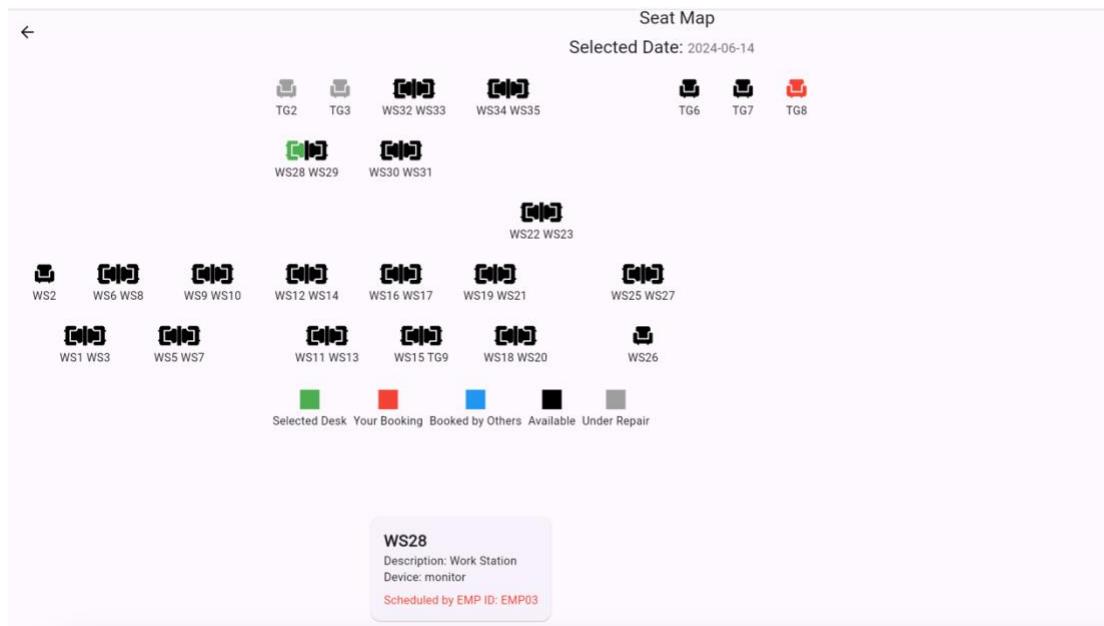
Figure 6.14: Map management alert cannot remove row

## 6.5 Employee View Map

There is a restriction that an employee cannot book a desk that is under repair and has been reserved by someone else. These seats are considered unreservable desks. So Figure 6.16 and Figure 6.17 show that in both cases that just discussed, when an employee selects that seat, it will directly show that the desk is unavailable.



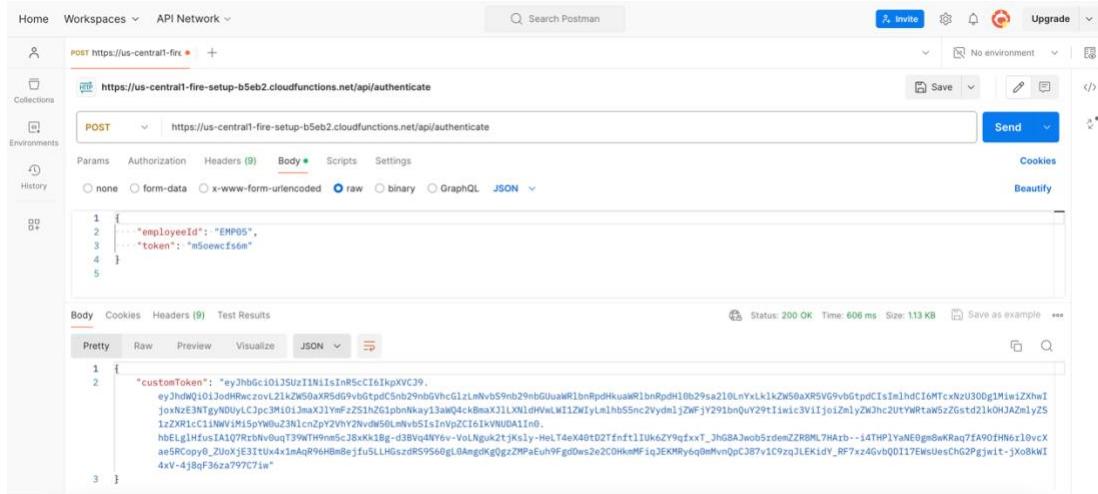
**Figure 6.15: Selected the seat that under repair**



**Figure 6.16: Selected the seat that booked by others**

## 6.6 QR code

Below shows an API testing process using Postman for a QR code verification function. The URL for the API endpoint being tested is on the URL that contain the QR code authentication. This API testing verifies if the provided employee ID and token are valid by returning a Custom Token upon successful authentication.



The screenshot shows the Postman interface with a POST request to <https://us-central1-fire-setup-b5eb2.cloudfunctions.net/api/authenticate>. The request body is set to raw JSON:

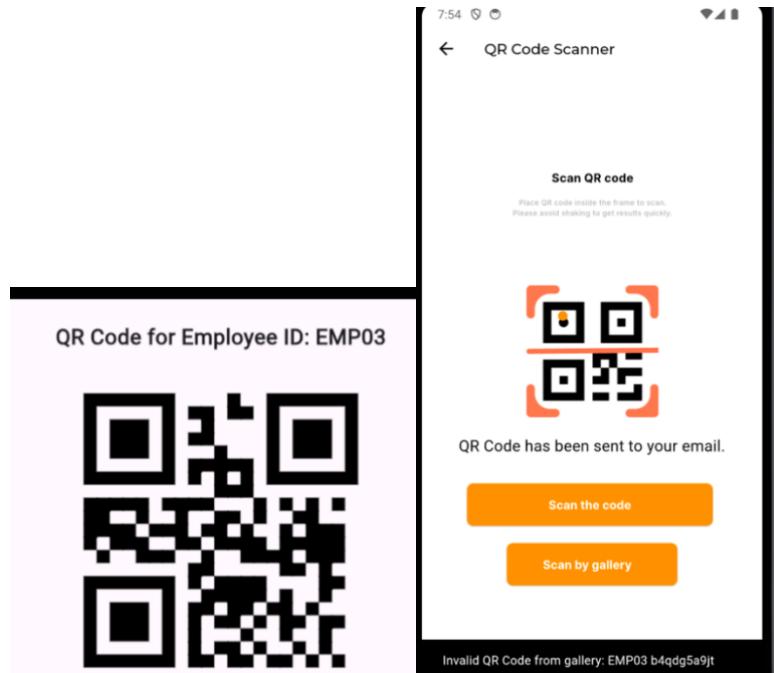
```
1 {
2   ...
3   "employeeId": "EMP05",
4   ...
5 }
```

The response body is also in raw JSON, showing a custom token:

```
1 {
2   "customToken": "eyJhbGciOiJFUzI1NltsTr8EcC16IhpXVCj9...
3   eyJhdW10I2d0Ricvazv211Zw59XR5d9v9b6tpSnb2nbGvcg121mkbV9n29nbGluak1bnRpdh10b29sa210lnYxLk1kZW50aXR5VG9v06tpcdIs1m1hdC16MTaxzu300g1Miw1ZKhwI
4   jox0zE3N7gyNDUyLCljpc3M1013maxX1Ynfz251nZC1pbhNkay13aWq4c18maX21lXN1dWvLW112WIy1mh55n2Yvd1nY29t1iwi3V1joiZmly2Zh2U1YRtaW5zZstd21kOJ3Azmly2Z
5   12zKR1c11MwV1M1SpYh0z2N1cn2pY2VhY2Nvds6LmVnB51s1nVpZC16IkwRUD1n0
6   h0ELg1Hfcs1A017RcbhVv0uqt39NTHmre5c3bXXk1Bg-d3Bvq4NyYev-VoLNguk2tjkG1y-HeLTdeX40tD27nf1tIU626ZY9qfx1xT_3hGBA3eob5demZzR0M7HAtv--i4THPlYnNE0g8wKraq7EA0fHNa10vCX
7   ae5RCopy_2UoXjE31tUx41m4qRf4hBmBejfus5LNGsZdRS9756dgl04engAGqgz2HrEuH9Fgd0ws2e2C0HkmRfiqJEKmRy6qMvnQpC287v1C9zq3LEK1dy_RF7xz4GvbQ0117EWsUesChG2Pggjw1t-jXo9AkI
8   4xV-4j8q136za797c71n"
```

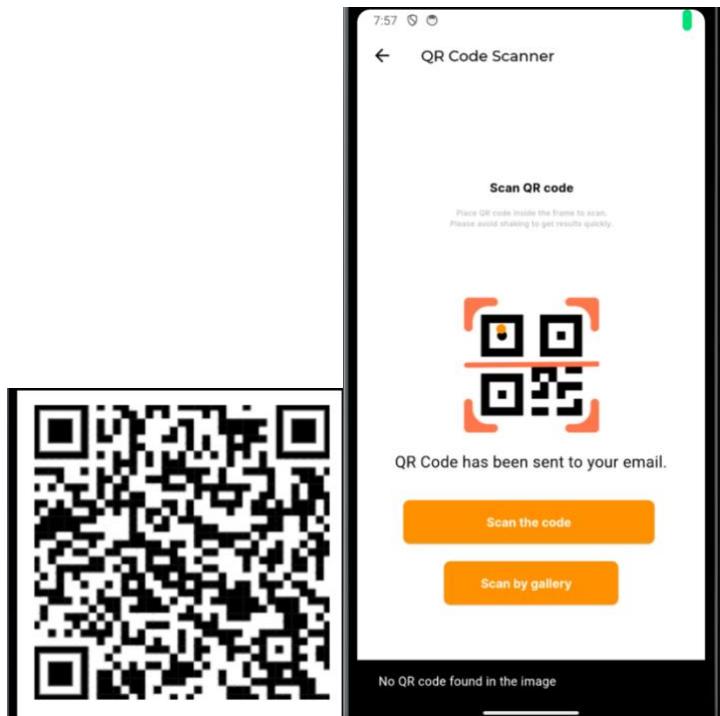
**Figure 6.17: QR API authentication test by Postman**

Another test was conducted by scanning a QR code that contained invalid data. The result show that the QR code is invalid, because it only included the employee ID and token. However, a valid QR code should contain the URL along with the employee ID and token.



**Figure 6.18: Scan invalid QR code test**

This test is occurs to see how the system will handle the expired QR code. As the output, the QR code can not found in the image means that the data inside the QR code is not correctly to authenticate with the employees.



**Figure 6.19: Scan expired QR code test**

## 6.7 Case Study

### 6.7.1 Case 1: Manage to book desk randomly

The external python code is call out using Flask by getting the three table from the firebase. The KNN, RF, SV, GB, LR, and DT were tested as machine learning models for predicting desk preference. In the below, the hyper tunings are being applying but most of the accuracy is in the middle range is believed to be due to the lack dataset to predict and train machine learning models.

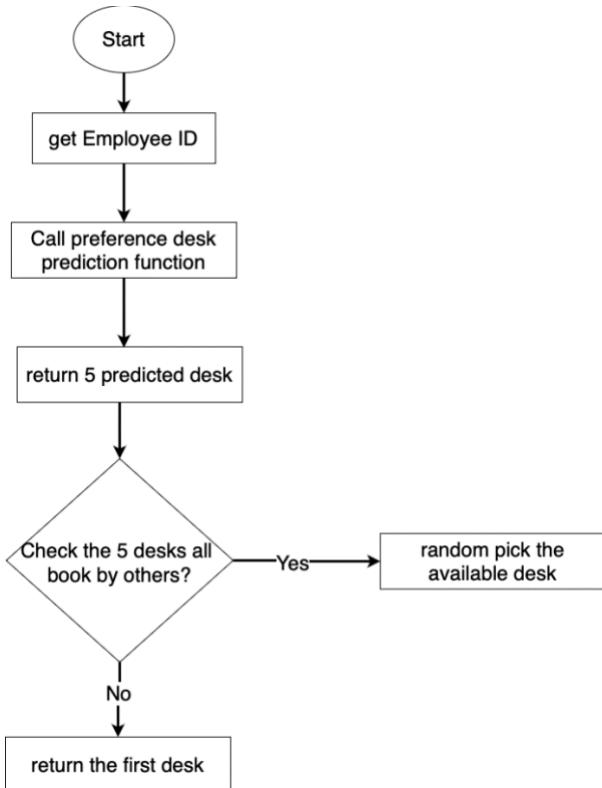
```
WEATHERING-SMITH
KNN - Best parameters: {'n_neighbors': 1}
KNN - Best cross-validated accuracy: 0.4069264069264069
RF - Best parameters: {'max_depth': None, 'n_estimators': 100}
RF - Best cross-validated accuracy: 0.45310245310245306
SVM - Best parameters: {'C': 10, 'gamma': 'auto'}
SVM - Best cross-validated accuracy: 0.40764790764790765
GB - Best parameters: {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 200}
GB - Best cross-validated accuracy: 0.5772005772005772
LR - Best parameters: {'C': 10}
LR - Best cross-validated accuracy: 0.4379509379509379
DT - Best parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
DT - Best cross-validated accuracy: 0.5476190476190476
```

Figure 6.20: Machine learning model Hyper tuning

```
KNN Accuracy: 0.5
RF Accuracy: 0.5
SVM Accuracy: 0.5
GB Accuracy: 0.5625
LR Accuracy: 0.5
DT Accuracy: 0.625
Predicted desk for EMP04: WS26
```

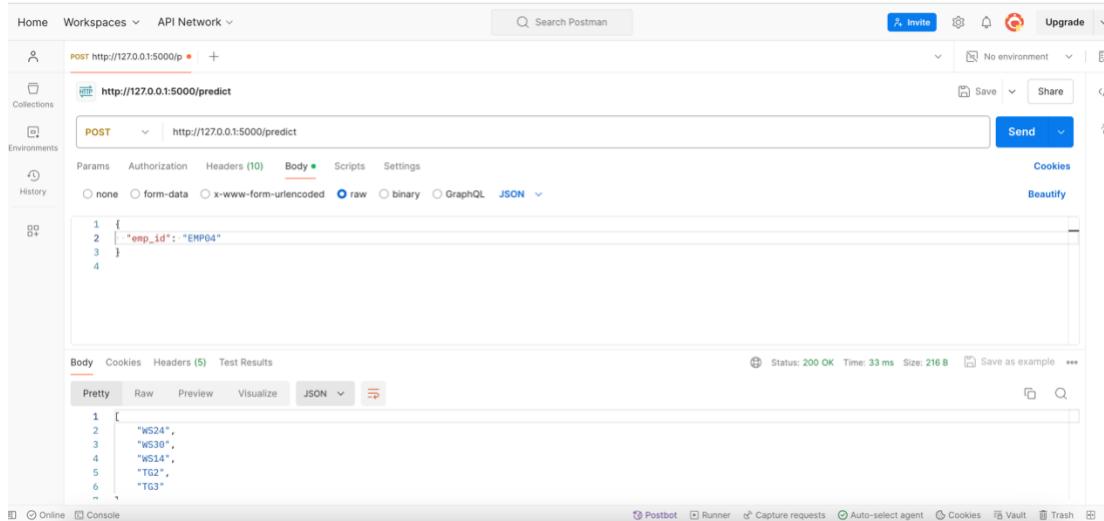
Figure 6.21: The result of the prediction

This flow chart illustrates how to work the random prediction. In the beginning the employee will try to book the desk and then the Employee ID is required to pass. Then it will call the prediction which had been train and then test by using the Employee ID to get and return the 5 best seats for employee. Then the system will start to check whether the 5 desks are available, if not then it will use the flutter build in math random to pick the desk; If yes, then the only one desk will be returned and able to make the booking.



**Figure 6.22: Flow chart of the random prediction**

Below shows an API testing process using Postman for a random desk prediction. This API testing show that the 5 desk can be return successfully by passing the only emp ID.



The screenshot displays the Postman application interface. At the top, there are navigation tabs: Home, Workspaces, API Network, and a search bar labeled "Search Postman". On the right side of the header, there are buttons for "Invite", "Upgrade", "No environment", "Save", "Share", and a "Send" button. The main workspace shows a POST request to "http://127.0.0.1:5000/predict". The "Body" tab is selected, showing the following JSON input:

```
1 {
2   "emp_id": "EMP04"
3 }
4
```

Below the body, the "Test Results" tab is visible, showing a status of "200 OK" with a response time of "33 ms" and a size of "216 B". The "Pretty" tab is selected in the preview area, displaying the JSON response:

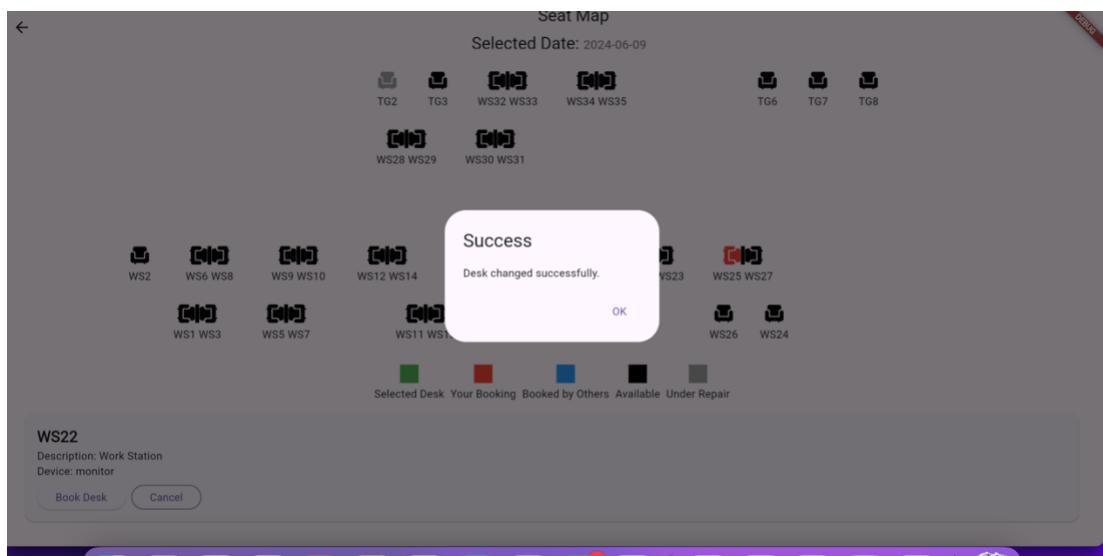
```
1 [
2   "WS24",
3   "WS30",
4   "WS14",
5   "TG2",
6   "TG3"
```

**Figure 6.23: Desk return from the random predict by Postman**

In the admin and employee modules, the system will first ensure the employee's booking status, while each employee is only allowed one booking per day. If the employee already has a schedule for the day, then the employee will not have the opportunity to make another booking on the same day. The process was tested and display in the Figure 6.25 until Figure 6.27 which show the result that the employee can only change the desk instead of schedule a new booking.



**Figure 6.24: View booking on the map**



**Figure 6.25: Change desk location successfully**



**Figure 6.26: Updated desk location**

Once a booking request is made, the system will randomly assign a desk from the pool of available desks, ensuring a fair and equitable allocation process. The system will check the availability of the desk at the time of booking and will only proceed if the desk is available. Desks that are under maintenance or have been booked by other users will be excluded from the pool to prevent conflicts.

```

Future<String> _getAvailableDeskId(DateTime date) async {
    List<String> scheduledDeskIds = [];
    // Get all booked desk IDs for the given date
    QuerySnapshot bookingSnapshot = await FirebaseFirestore.instance
        .collection('booking')
        .where('booking_date', isEqualTo: DateFormat('yyyy-MM-dd').format(date))
        .get();

    for (QueryDocumentSnapshot doc in bookingSnapshot.docs) {
        scheduledDeskIds.add(doc['desk_id']);
    }
    // Get all desk IDs and find the maintenance desk ID
    QuerySnapshot deskSnapshot =
        await FirebaseFirestore.instance.collection('desks').get();
    List allDeskIds = deskSnapshot.docs.map((doc) => doc['desk_id']).toList();

    String? maintenanceDeskId;
    for (var doc in deskSnapshot.docs) {
        if (doc['status'] != 'available') {
            maintenanceDeskId = doc['desk_id'];
            break;
        }
    }
    // Exclude scheduled and maintenance desk IDs from available desk IDs
    List availableDeskIds = allDeskIds
        .where((deskId) =>
            !scheduledDeskIds.contains(deskId) && deskId != maintenanceDeskId)
        .toList();

    if (availableDeskIds.isNotEmpty) {
        return availableDeskIds[Random().nextInt(availableDeskIds.length)];
    } else {
        return 'No available desk';
    }
}

```

**Figure 6.27: Code for selecting available desks in system**

### 6.7.2 Case 2: Prevent administrator can book for more than one month.

The system first detects the role at login to check whether the role is an administrator or an employee. If the role is identified as an employee, the booking calendar will only show availability for the next 7 days. Conversely, if the role is detected as an administrator, the calendar will be extended to show availability for the next 30 days.

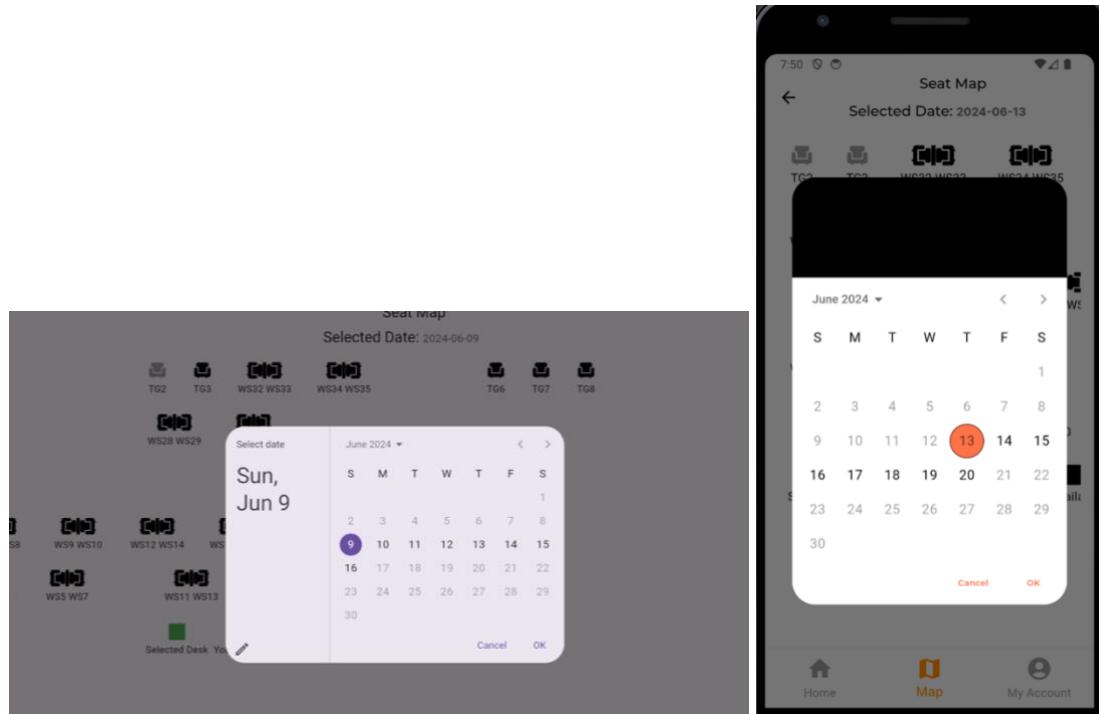


Figure 6.28: Booking calendar on employee module

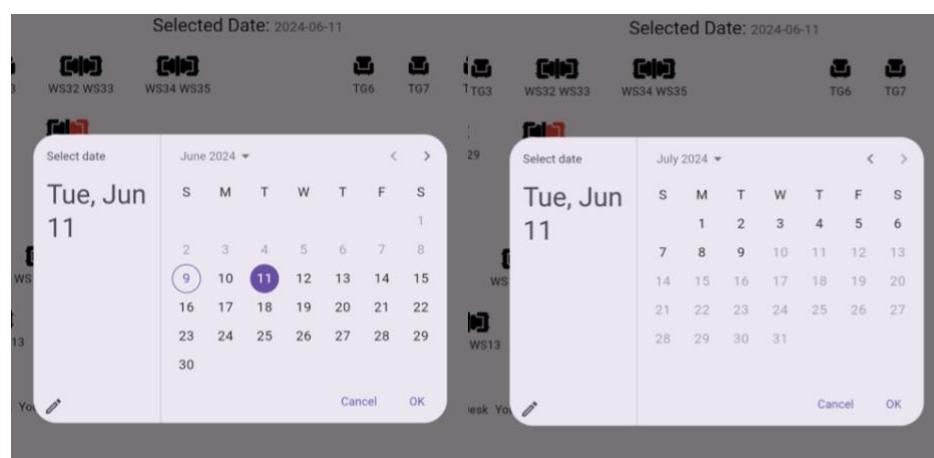
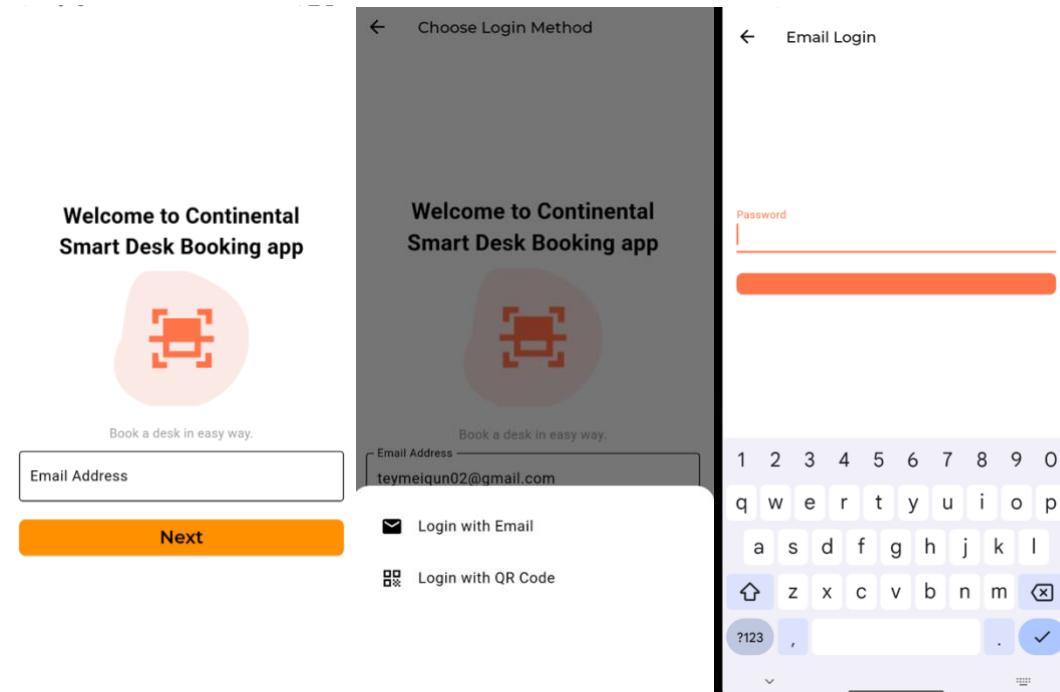


Figure 6.29: Booking calendar on admin module

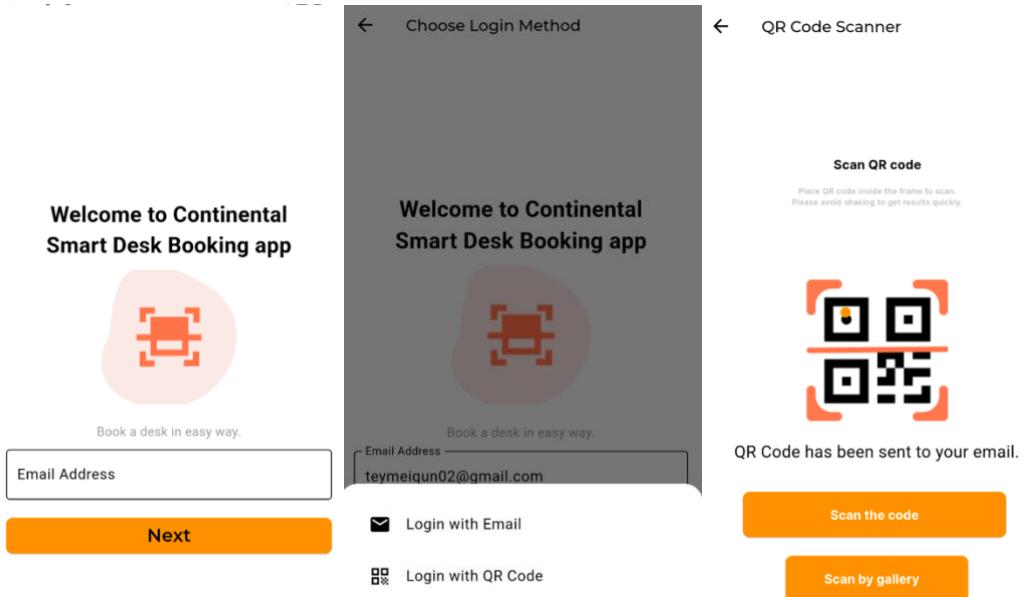
### 6.7.3 Case 3: QR Code login via application

The application will first ask the employee to enter the email address to check if the email exists in the database. If so, the employee can choose to log in via email password or QR code. If password login is selected, once the employee enters the correct password, then the login will be successfully made. The step was taken by 3 as shown in below.

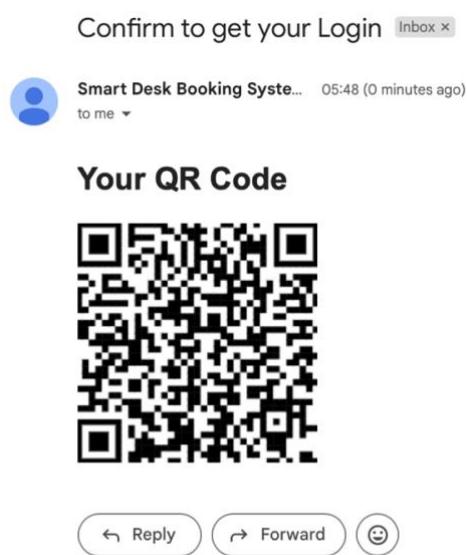


**Figure 6.30: Application Login process via email password**

If the employee chooses to log in via the QR code, the next page will be retrieved and it will say that the email was sent. The employee will then need to open the email to check it, then return to the app and scan to log in. So it already taken the step that more than the email password when comparing the efficiency.



**Figure 6.31: Application Login process via QR code**



**Figure 6.32: Received QR code by email**

## **6.8 Limitation**

The proposed system relies heavily on a stable internet connection for real-time updates and data synchronization. In areas with poor connectivity, system performance and reliability may suffer. The current use of Firebase presents compatibility issues that may necessitate custom development and ongoing maintenance due to its limited capacity. Firebase offers only 1 gibibyte for data storage, with daily limits of 50,000 document reads, 20,000 document writes, and 20,000 document deletes. The QR code verification process also presents significant security risks as it involves passing sensitive information such as tokens and employee IDs in the QR code.

# **CHAPTER 7**

## **CONCLUSION**

### **7.1 Summary**

In summary, this dual-platform system meets the needs of modern hybrid work environments by seamlessly integrating website and app interfaces and is designed for smart desk booking systems. It improves the efficiency of office operations for administrators and provides employees with flexible workspace booking through an easy-to-use application. The system is designed to be intuitive and accessible, ensuring that both tech-savvy and non-tech-savvy employees can navigate with ease.

Additionally, automatic seat assignment is a key feature that significantly contributes to the overall efficiency of a smart desk booking system. This seat assignment feature not only simplifies the process of assigning desks, but also provides multiple benefits that enhance employee experience and office operations. By automating the allocation process, the system reduces the possibility of human error, ensures optimal utilization of available workspace and supports social distancing measures where required. A graphical map interface combined with real-time updates forms the basis of an advanced seat assignment system. This visual representation allows employees to easily view available desks and make informed choices about their workspace.

On other hands, the QR code verification process implemented in the system adds additional security and convenience, especially for employee login and access control. This feature ensures that only authorized personnel can access certain areas, thereby enhancing overall security. QR code verification is fast and efficient, allowing employees to log in and access their assigned workspace without unnecessary delays. In addition, the integration of Firebase services provides powerful backend support, ensuring data is managed securely and efficiently, with real-time synchronization across all platforms.

The system's reliance on modern technologies like Firebase, Flutter, and Dart highlights its adaptability and forward-thinking design. These technologies not only ensure a smooth and responsive employee experience but are also easily updateable

and scalable as the organization's needs grow. Smart desk booking systems can handle a variety of office layouts and employee preferences, making them a versatile solution for dynamic work environments.

Lastly, this project not only enabled the author to master the Dart programming language and Flutter development skills, but also provided valuable insights into cross-platform application design. The experience gained by integrating various Firebase services, enabling real-time data updates, and ensuring secure authentication methods prepared the author for future challenges in software development. The project demonstrates the potential of well-designed software to transform traditional office operations and increase flexibility and efficiency in the modern workplace. Smart desk booking systems are an example of how technology can be used to create a more efficient, secure and adaptable work environment.

## 7.2 Future Works

To address the limitations of the proposed system, future work will focus on enhancing connectivity solutions, scalability, security, data management, and ongoing system optimization. First, developing offline capabilities and optimizing data transfer protocols will ensure normal use in areas with poor Internet connectivity, allowing users to perform basic functions offline and synchronize data when a stable connection is available. In addition, exploring hybrid cloud solutions and developing custom backend infrastructure will help alleviate storage limitations and provide more powerful data processing capabilities, allowing the system to scale effectively as the user base grows. Desk booking systems should also provide advanced analytics and reporting tools to track desk usage patterns, providing administrators with customizable reports to improve resource utilization. Therefore, it will be scalable, and performance optimized and will ensure that the system can effectively handle the growing number of employees and data volumes. These future works will ensure that the system remains a robust and secure solution, improving employee productivity and operational efficiency.

## REFERENCES

- Android Barcode Scanner Library using Kotlin - Scanbot SDK. (2023, September 13). Scanbot SDK. Retrieved January 13, 2024, from <https://scanbot.io/developer/android-barcode-scanner/>
- Android Mobile App Developer Tools – Android Developers. (n.d.). Android Developers. Retrieved January 13, 2024, from <https://developer.android.com/>
- Aryal, A., Becerik-Gerber, B., Anselmo, F., Roll, S. C., & Lucas, G. M. (2019). Smart desks to promote comfort, health, and productivity in offices: A vision for future workplaces. *Frontiers in Built Environment*, 76.
- Chen, G. (2020). Office Space Timeline: Past, Present, and Future [INFOGRAPHIC] | HubbleHQ.<https://hubblehq.com/blog/office-space-timeline-past-present-future-infographic>
- Chow, Y. W., Susilo, W., Yang, G., Au, M. H., & Wang, C. (2016). Authentication and transaction verification using QR codes with a mobile device. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 9th International Conference, SpaCCS 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings* 9 (pp. 437-451). Springer International Publishing.
- De Seta, G. (2023). QR code: The global making of an infrastructural gateway. *Global Media and China*, 20594364231183618.
- Escobedo, P., Ramos-Lorente, C. E., Ejaz, A., Erenas, M. M., Martínez-Olmos, A., Carvajal, M. A., ... & Palma, A. J. (2023). QRsens: Dual-purpose Quick Response code with built-in colorimetric sensors. *Sensors and Actuators B: Chemical*, 376, 133001.
- ESLint. (n.d.). ESLint. Retrieved January 13, 2024, from <https://eslint.org>
- Fajar, M., Gunawan, Y., Manaf, K., Abd Rahman, A. B., Perdana, R. S., & Mulyono, T. T. (2023, October). Web Login Authentication System Using QR-Code Scanner with Sha-1 Encryption Method. In *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)* (pp. 1-4). IEEE.
- Qr\_flutter | Flutter Package. (n.d.). Dart Packages. Retrieved January 13, 2024, from [https://pub.dev/packages/qr\\_flutter](https://pub.dev/packages/qr_flutter)
- Fan, Y., & Jin, Y. (2020). Research on the Development and Future Trend of Office Furniture Design. In *E3S Web of Conferences* (Vol. 179, p. 02075). E3S Web of Conferences.

Figma: The Collaborative Interface Design Tool. (n.d.). Figma. Retrieved January 13, 2024, from <https://www.figma.com/>

Flutter architectural overview. (n.d.). Flutter - Build Apps for Any Screen. Retrieved January 13, 2024, from <https://flutter.dev/>

Flutter Architectural Overview | Flutter. Retrieved January 13, 2024, from <https://docs.flutter.dev/resources/architectural-overview>

Golden Screen Cinemas - Malaysia's leading Cinema Online. Golden Screen Cinemas - Malaysia's Leading Cinema Online. (n.d.). Retrieved on December 3, 2023, from <https://www.gsc.com.my>

Google\_maps\_flutter | Flutter Package. (n.d.). Dart Packages. Retrieved January 13, 2024, from [https://pub.dev/packages/google\\_maps\\_flutter](https://pub.dev/packages/google_maps_flutter)

Halim, N., Awang, A. H., Ahmad, N., Jalil, N. A., Denan, Z., & Abdul Majid, N. H. (2021, July 30). Towards Green Office: A Systematic Literature Review on Smart Office Interior in Malaysia.

Hartman, V. (2023, July 7). What Is a Smart Office? History, benefits and hot desking apps. — Do You Want To Optimise Your Workspace? Do You Want to Optimise Your Workspace? <https://www.iotspace.co.uk/blog/2018/7/20/what-is-a-smart-office>

Ho, W. E., Ong, L. Y., & Leow, M. C. (2022, September). Blended QR Code for Digital Advertising. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)* (pp. 1-6). IEEE.

Howell, T. (2022). Coworking spaces: An overview and research agenda. *Research Policy*, 51(2), 104447.

Introduction to Node.js. (n.d.). Node.js. Retrieved January 13, 2024, from <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

Leong, Y. C., & Sharif, R. (2020). The potentials of new work environment for dynamic workforce in the 21st century of Malaysia context. *MAJ-Malaysia Architectural Journal*, 2(1), 33-37

LiquidSpace - Flex Space & Hybrid Workplace Software. (n.d.). Liquidspace.com. Retrieved December 3, 2023, from <https://liquidspace.com>

Racquetvenues (n.d.).MBA @ Bukit Puchong—TennisVenues.com.au. Retrieved on December 3, 2023, from <https://au.racquetvenues.com/venue/mba-bukit-puchong>

RedBus. (n.d.) Bus Tickets Online, Ferry Tickets | Malaysia's #1 Bus Booking Platform. redBus Malaysia. Retrieved January 20, 2024, from <https://www.redbus.my>

Richardson, L. (2021). Coordinating office space: Digital technologies and the platformization of work. *Environment and Planning D: Society and Space*, 39(2), 347-365.

Shinde, R. M., Deval, N. D., & Kadam, S. S. (2020). Smart office. 11(6), 1006–1009. What is agile software development? -AHA!. (2023, September 29). Retrieved December 3, 2023, from <https://www.aha.io/roadmapping/guide/agile/agile-software-development>

Sutheebanjard, P., & Premchaiswadi, W. (2010, November). QR-code generator. In 2010 Eighth International Conference on ICT and Knowledge Engineering (pp. 89-92). IEEE.

The History of Coworking. (n.d.). FreeOfficeFinder. Retrieved January 13, 2024, from <https://www.freeofficefinder.com/article/the-history-of-coworking>

Tillu, J. (2023, July 9). 4. What is Widget in flutter ? Let's clear the basics first. Medium. Retrieved January 13, 2024, from <https://medium.com/jay-tillu/4-what-is-widget-in-flutter-lets-clear-the-basics-first-82f501c8d0f0>

Visual studio code - code editing. redefined. (2021, November 3). Retrieved on January 31, 2023, <https://code.visualstudio.com/>

Your Guide to the Flutter Framework. (2021, May 18). Perfecto by Perforce. Retrieved January 13, 2024, from <https://www.perfecto.io/blog/what-is-flutter-framework>

Zeng, J., Yang, L. T., Ning, H., & Ma, J. (2015). A systematic methodology for augmenting quality of experience in smart space design. *IEEE Wireless Communications*, 22(4), 81-87.

Zhang, X., Zheng, P., Peng, T., He, Q., Lee, C. K., & Tang, R. (2022). Promoting employee health in smart office: A survey. *Advanced Engineering Informatics*, 51, 101518.

## APPENDICES

### Appendix A: Meeting Log



Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log

MEETING DATE: 21/2/2024	MEETING NO.:1
PROJECT ID: T823234	
<b>PROJECT TITLE :SMART DESK BOOKING SYSTEM</b>	
SESSION : TRIMESTER MAR/APRIL 2024	COMPANY SUPERVISOR :Chew,Phye Boon
STUDENT ID & Name:1201200152 Tey Mei Qun	FACULTY SUPERVISOR :Ong Lee-Yeng

All to be filled in by student

<b>1. WORK DONE</b> <ul style="list-style-type: none"><li>• Prepare a timetable plan for achieving the basic framework and functions of the three main objectives for FYP2.</li></ul>
<b>2. WORK TO BE DONE</b> <ul style="list-style-type: none"><li>• Follow the timeline schedule to develop the functionalities.</li></ul>
<b>3. PROBLEMS ENCOUNTERED</b> <ul style="list-style-type: none"><li>• The database AWS Amplify that selected previously cannot connect with the proposed system.</li></ul>
<b>4. COMMENTS</b>

A handwritten signature in black ink, appearing to read "Ong".

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
71800 Pahang, Malaysia

A handwritten signature in black ink, appearing to read "S".

Company Supervisor's  
Signature & Stamp

Faculty Supervisor's  
Signature & Stamp

Student's Signature

#### NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
3. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
4. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log**

<b>MEETING DATE:</b> 30/3/2024	<b>MEETING NO.:</b> 2
<b>PROJECT ID:</b> T823234	
<b>PROJECT TITLE :</b> SMART DESK BOOKING SYSTEM	
<b>SESSION :</b> TRIMESTER MAR/APRIL 2024	<b>COMPANY SUPERVISOR :</b> Chew,Phye Boon
<b>STUDENT ID &amp; Name:</b> 1201200152 Tey Mei Qun	<b>FACULTY SUPERVISOR :</b> Ong Lee-Yeng

All to be filled in by student

<b>1. WORK DONE</b>
<ul style="list-style-type: none"> <li>• Demonstrate initial system prototype.</li> <li>• Able to make the simple booking</li> </ul>
<b>2. WORK TO BE DONE</b>
<ul style="list-style-type: none"> <li>• Implement the interface of system .</li> <li>• Fix the problem that the emulator cannot scan QR code.</li> </ul>
<b>3. PROBLEMS ENCOUNTERED</b>
<ul style="list-style-type: none"> <li>• Fail to fulfil the objective 3 to verify the application by scanning the QR code on phone.</li> </ul>
<b>4. COMMENTS</b>

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama

Company Supervisor's  
Signature & Stamp

Faculty Supervisor's  
Signature & Stamp

Student's Signature

**NOTES:**

5. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
6. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
7. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
8. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log**

<b>MEETING DATE:</b> 17/4/2024	<b>MEETING NO.:</b> 3
<b>PROJECT ID:</b> T823234	
<b>PROJECT TITLE :</b> SMART DESK BOOKING SYSTEM	
<b>SESSION :</b> TRIMESTER MAR/APRIL 2024	<b>COMPANY SUPERVISOR :</b> Chew, Phye Boon
<b>STUDENT ID &amp; Name:</b> 1201200152 Tey Mei Qun	<b>FACULTY SUPERVISOR :</b> Ong Lee-Yeng

All to be filled in by student

<b>1. WORK DONE</b>
<ul style="list-style-type: none"> <li>• Demo the current system progress.</li> <li>• Rebuild the database framework with Firebase.</li> </ul>
<b>2. WORK TO BE DONE</b>
<ul style="list-style-type: none"> <li>• Write the FYP 2 report.</li> <li>• Illustrate and draw the system architecture for chapter 5.</li> </ul>
<b>3. PROBLEMS ENCOUNTERED</b>
<ul style="list-style-type: none"> <li>• Fail to fulfil the objective 3 to verify the application by scanning the QR code on phone.</li> </ul>
<b>4. COMMENTS</b>

Company Supervisor's  
Signature & Stamp

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
76450 Melaka, Malaysia

Student's Signature

**NOTES:**

9. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
10. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
11. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
12. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log**

<b>MEETING DATE:</b> 24/4/2024	<b>MEETING NO.:</b> 4
<b>PROJECT ID:</b> T823234	
<b>PROJECT TITLE :</b> SMART DESK BOOKING SYSTEM	
<b>SESSION :</b> TRIMESTER MAR/APRIL 2024	<b>COMPANY SUPERVISOR :</b> Chew, Phye Boon
<b>STUDENT ID &amp; Name:</b> 1201200152 Tey Mei Qun	<b>FACULTY SUPERVISOR :</b> Ong Lee-Yeng

All to be filled in by student

**1. WORK DONE**

- Done complete the report for Chapter 5 and illustrate the system architecture.

**2. WORK TO BE DONE**

- Revise the system architecture and write more details.
- Trying to enable the system to authenticate by passing a token in a QR code

**3. PROBLEMS ENCOUNTERED**

- Fail to fulfil the objective 3 to verify the application by scanning the QR code on phone.

**4. COMMENTS**

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Sciences and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
80000 Johor Bahru, Malaysia

Student's Signature

Company Supervisor's  
Signature & Stamp

Faculty Supervisor's  
Signature & Stamp

**NOTES:**

13. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
14. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
15. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
16. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log

<b>MEETING DATE:</b> 30/4/2024	<b>MEETING NO.:</b> 5
<b>PROJECT ID:</b> T823234	
<b>PROJECT TITLE :</b> SMART DESK BOOKING SYSTEM	
<b>SESSION :</b> TRIMESTER MAR/APRIL 2024	<b>COMPANY SUPERVISOR :</b> Chew, Phye Boon
<b>STUDENT ID &amp; Name:</b> 1201200152 Tey Mei Qun	<b>FACULTY SUPERVISOR :</b> Ong Lee-Yeng

All to be filled in by student

**1. WORK DONE**

- Done presenting the updated progress to the company and discuss the current challenges.

**2. WORK TO BE DONE**

- Implement the interface of the website ,suggested by supervisor to used the M3 Material.

**3. PROBLEMS ENCOUNTERED**

- Fail to fulfil the objective 3 to verify the application by scanning the QR code on phone.

**4. COMMENTS**

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
75450 Melaka, Malaysia

Company Supervisor's  
Signature & Stamp

Faculty Supervisor's  
Signature & Stamp

Student's Signature

**NOTES:**

17. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
18. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
19. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
20. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log**

<b>MEETING DATE:</b> 8/5/2024	<b>MEETING NO.:</b> 6
<b>PROJECT ID:</b> T823234	
<b>PROJECT TITLE :</b> SMART DESK BOOKING SYSTEM	
<b>SESSION :</b> TRIMESTER MAR/APRIL 2024	<b>COMPANY SUPERVISOR :</b> Chew, Phye Boon
<b>STUDENT ID &amp; Name:</b> 1201200152 Tey Mei Qun	<b>FACULTY SUPERVISOR :</b> Ong Lee-Yeng

All to be filled in by student

<b>1. WORK DONE</b>
<ul style="list-style-type: none"> <li>The application and website could integrate and run smoothly.</li> </ul>
<b>2. WORK TO BE DONE</b>
<ul style="list-style-type: none"> <li>Implement the switch seat function after the booking has been made successfully.</li> </ul>
<b>3. PROBLEMS ENCOUNTERED</b>
<ul style="list-style-type: none"> <li>The QR code cannot decode the passing token.</li> </ul>
<b>4. COMMENTS</b>

DR. ONG LEE YENG  
Lecturer  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
35600 Melaka, Malaysia

Company Supervisor's  
Signature & Stamp

Faculty Supervisor's  
Signature & Stamp

Student's Signature

**NOTES:**

21. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
22. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
23. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
24. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)

## APPENDICES

### Appendix B: Check list for Final Report Submission



### Faculty of Information Science and Technology (FIST) Checklist for Final Report Submission (To be filled in by Student)

#### STUDENT'S DETAILS

Project Code	T823234
Name	TEY MEI QUN
ID No	1201200152
Title of Thesis	SMART DESK BOOKING SYSTEM
Supervisor Name	DR.ONG LEE YENG

REPORT ARRANGEMENT	✓	Comments (if any differences)
1. Cover of The Final Report	✓	
2. Title Page of the Final Report	✓	
3 Copyright page of I Final Report	✓	
4. Declaration Page of Final report	✓	
5. Acknowledgement	✓	
6. Table of Contents	✓	
7. Abstract	✓	
8. List of Tables	✓	
9. List of Figures	✓	
10. List of Symbols	✓	
11. List of Appendices	✓	
12. Chapter 1: Introduction – objectives, scope	✓	
13. Chapter 2: Literature Review	✓	
14. Chapter 3: Methodology	✓	
15. Chapter 4: Proposed System	✓	
16. Chapter 5: Implementation	✓	
17. Chapter 6: Testing Evaluation	✓	
18. Chapter 7: Conclusion	✓	
19. References – APA style	✓	
20. Appendices	✓	
21. CD/ DVD and envelope as shown in Appendix K		
22. Attachment: FYP Meeting Logs (all) - 1 set	✓	
FORMAT OF REPORT	✓	Comments

1. Page Numbering	✓	
2. Font and Type Face	✓	
3. Font Cover	✓	
4. Tables and Figures	✓	
5. Spine Format	✓	
6. Comb Bind (For evaluation)		
7. Permanent Bind (After approval)		
8. Colour of the Front Cover		
9. Number of words > 10000 (Main content only)	✓	

Checked by


 A handwritten signature consisting of several loops and a horizontal line extending to the right.

Student's Signature & Date

13/6/2024