Project Management Plan

Voucher_Service

E-voucher application development using the incremental method

V1.0

Team: T16_01_Incremental

1124298	Sakshi Chandel
359101	Alastair Daivis
1147173	Decy Restyan Damayanti
970553	Setender Nandal
1114278	James Sinclair
758966	Xin Wei Ding

2. Executive Summary

Voucher_Service is a web-based voucher application being developed as a means to provide additional end-of-year bonuses for MYD employees in the form of vouchers which can be exchanged for goods and services from a variety of surrounding small businesses. This will help build employee satisfaction by rewarding staff for MYD's strong year. Voucher_Service will also support the participating small businesses surrounding MYD that have struggled due to the COVID-19 pandemic, strengthening their relationships with MYD, and encouraging future collaboration. The initial development of the application will demonstrate the viability of Voucher_Service, which if successful, could be opened to the public after further development. This could increase MYD's revenue through commissions on voucher sales.

The current project will follow the incremental methodology, delivering the project in two releases over one month. The first release, delivered 7 May 2021, will implement the main functionality of user registration, sign in, and voucher booking. The final release, delivered 24 May 2021, will add email confirmations for various stages of the voucher booking process, and allow users to modify their details.

The project budget is $0, as the project is being implemented by a team of University of Melbourne students. The team of six has a project manager, business owner (representing MYD), one technical specialist, a UX expert and two web developers. While they will not be paid for the development, the project team will benefit by gaining experience in software project management.

A key risk for the project is that estimates of student availability may be lower than estimated, resulting in unimplemented features. Accordingly, progress will be closely monitored against the project schedule.

The technology stack for this project will be React and Firebase. These were chosen for their rich feature set, ease of use, and simple deployment.

3. Table of Contents

4. Introduction

4.1 Purpose of document

The intent of this PMP is to document in detail the plan for the execution of the Voucher_Service project. The document will continue to be updated throughout the project and should be consulted as the definitive source of project information. The document is intended to be complemented by the full project schedule linked in Section 6. The project manager will be responsible for the maintenance of the PMP.

The PMP documents project information in Section 5. This includes stakeholders and an assessment of the value of the project to them. The project scope including in-scope and out-of-scope features is also provided. The SDLC chosen, the incremental method, and reasoning for its selection are detailed. The roles and responsibilities of the team are indicated, as well as the communication plan for stakeholders and the project team. Key risks and their management strategies are outlined. The possible tech stacks considered are compared, and a rationale for the chosen stack of Firebase/React is provided.

Section 6 of the PMP provides a breakdown of the project into a two-release project schedule.

4.2 Audience of document

This document is primarily intended for the project team, project manager, and MYD business owner Tianyi. Future development or maintenance efforts should also benefit from this document by understanding the rationale for decisions made.

4.3 Evolution of document

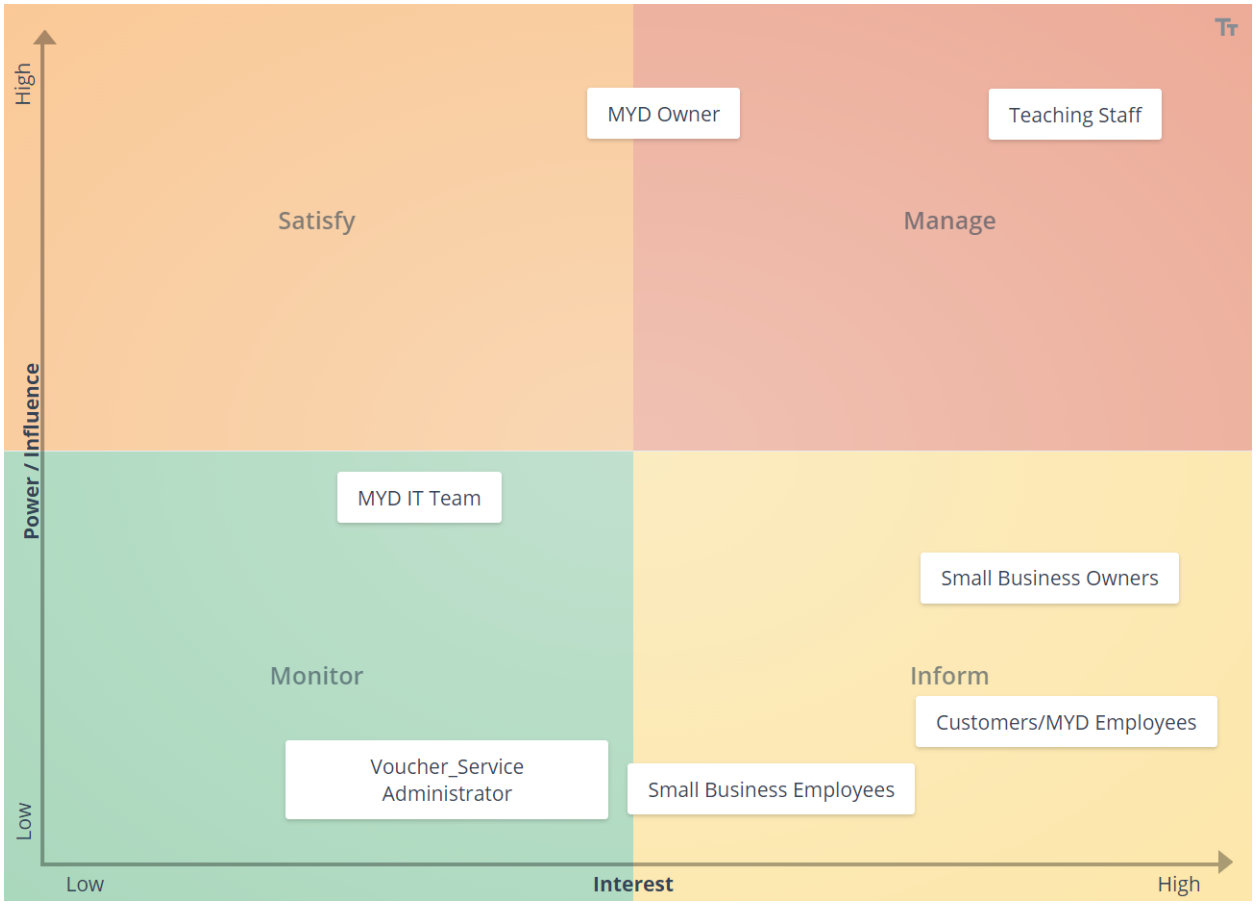
Version	Individual Responsible	Date created	Comments
v1.0	James Sinclair	23/04/2021	Compiled by project team

5. Project Information

5.1 Key Stakeholders

Voucher_Service has many stakeholders that are interested in the outcome of the service. A stakeholder register is shown below detailing and classifying each stakeholder. The stakeholder map below shows the interest/power of each stakeholder.

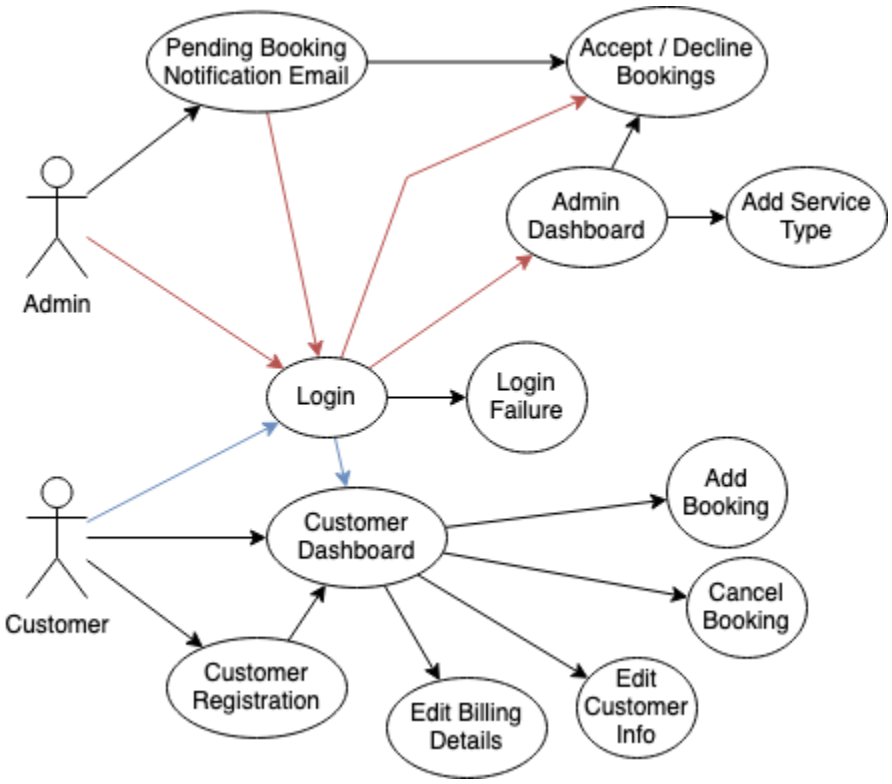
Stakeholder map



Stakeholder Register

Stakeholder	**Internal/ external**	**Role**	**Communication Channel**	**Level of influence**	**Frequency of communication**
Tianyi	Internal	Owner	Zoom meeting, Email	High	Weekly
Voucher_Service Administrator	Internal	Ability to confirm/cancel voucher bookings	Email	Low	Per release
MYD IT Team	Internal	User administration (setup and maintaining account), Maintaining system.	Email	Low-Med	Per release
Project Team	Internal	Developing website for the customers /users for voucher service	Zoom meeting	Medium	Daily
Small business owners	External	Shareholders.pay the owner whenever a coupon is redeemed	Email	Medium	As required
Customers (here MYD Employees)	External	Buy and redeem coupons from the website	Email notifications	Low	Per release
Small business staff members	External	Will need to check validity of vouchers	Face to face	Low	As required

5.2 In-scope Features



Use Case Diagram for Voucher_System

In brief, the features in scope are:

- Administrator features:
    - Log in as administrator
    - View all bookings (Administrator dashboard)
    - Accept / Decline bookings from dashboard
    - Accept / Decline bookings from pending booking notification email
    - Add new service types
- Customer features
    - Customer registration
    - Log in as customer
    - View customer bookings (customer dashboard)
    - Edit billing details
    - Edit customer information
    - Add new booking

        - Cancel pending booking
- Miscellaneous features:
    - Notifying Admin by email when a new booking is made
    - Notifying Customer by email when a booking is accepted or declined

Details on the in-scope features may be found in the Use Case documentation (see Appendix A).

5.3 Out-of-scope Features

The following features are defined as out-of-scope for the initial development of this project:

- Administrator features:
    - Remove or modify service types
    - Change administrator email and password
    - View bookings as calendar
    - Invoicing system
- Customer features:
    - View bookings as calendar
    - Delete profile
    - Change email and password
    - Voucher acquisition
    - Point balance tracking
    - Booking failure
- Service provider features
    - All service-provider interfaces are out of scope

5.4 Delivery approach / SDLC

☐ Waterfall ☑ **Incremental** ☐ Agile

Three SDLC methodologies were considered for the implementation of this project: waterfall, incremental, and agile. In this section, we assess the suitability of these methodologies with respect to project characteristics, and provide a justification for our choice of the incremental approach.

Requirements and change

This project's requirements are well defined and are well understood. The voucher booking service is not particularly novel, and the project's technical complexity is relatively low.
Given the short timeframe, and the clearly defined project scope, it does not seem like there is a large amount of room for the requirements to change significantly. This suggests that the rigidity of a formal methodology is unlikely to reduce the likelihood of project success.
The simplicity of the requirements suggests that a waterfall approach may be feasible, however any changes in requirements that do occur would have no time to be absorbed and the project modified accordingly. The incremental approach would allow core functionality to be demonstrated early in the project, and based on feedback from the business owner, changes could be made before final delivery. Any changes in requirements that do occur would have more opportunity to be incorporated using the incremental approach. The agile approach would be the most tolerant of change, however given the expectation of change is low, this does not make agile a preferred approach.

Communication and coordination

The team is spread across time zones, and has conflicting availability.
This poses challenges for the synchronous communication which is typically expected with an agile approach (e.g. daily standups). Formal methods appear to be a better choice, as a clear communication plan is produced and documented, with no particular expectation of synchronous communication. While the incremental approach will have a higher overhead than waterfall in terms of documentation, by keeping the number of releases small (i.e. two or three), this will be kept to a minimum.

Team experience

Most of the team members are relatively inexperienced with web development.
Using an agile approach may present a challenge as team members would be expected to self-organise, which could result in delays without clear direction. In contrast, a formal approach establishes clearly defined roles and responsibilities, which will be simpler to follow for the inexperienced project team. If an incremental approach was adopted, the team would be required to produce a working release part-way through the project, quickly building experience with a new toolset, while providing a buffer to adapt if the release does not meet requirements. Lacking a clearly defined point in the development lifecycle for an initial release, a Waterfall

lifecycle model would not provide a convenient point at which to assess the efficacy of the toolset and adjust development practices accordingly.

Chosen SDLC

The project characteristics suggest that a formal methodology would be best suited to the task. While agile is tolerant of change, little change is expected. Agile would also produce communication/coordination challenges, and may add schedule risk due to self-organising team structure.

The choice between waterfall and incremental approaches was made by considering that the chance of change in requirements is small but non-zero. The waterfall approach is highly rigid, and provides no opportunity to change course, while the incremental approach is able to adapt to a small amount of change in requirements between releases. The incremental approach also helps minimise the chance of having undelivered core features by delivering the most important functionality in early releases.

5.5 Business Value (Financial & Non-Financial Benefits)

The value of the Voucher_Service application to each stakeholder is detailed below.

MYD owner (Tianyi)

The project provides value to Tianyi by supporting employees through an additional bonus via the voucher provided, building team morale and loyalty. The project also supports the surrounding local businesses, which helps build relationships and encourages future collaborations. Besides, it demonstrates the viability of the voucher service, which could provide an additional source of revenue to MYD.

MYD Business

MYD Business gets goodwill benefits from business partners offering services through the application. This benefit may manifest as word-of-mouth advertising or continuing business relationships.

MYD Employees (Customer)

This project increases the flexibility and convenience for the MYD employees to redeem their vouchers. They have several options for redeeming vouchers and make bookings through the application without queuing at the particular merchant.

MYD Voucher_Service administrator

This project provides convenience by facilitating the system admin to manage voucher services.

MYD IT Team

This project does not provide clear benefits to the MYD IT team, and may create additional work for them. They will benefit from the system by receiving vouchers.

Participating small business owners

This project helps to revive small businesses that are struggling through a voucher system. A successful voucher project potentially supports future business growth.

Participating small business employees

This project helps strengthen small businesses, so their employees will help them keep their jobs.

Project team

This project allows the students team to gain experience and practice their knowledge in an actual company. They can sharpen their communication, managerial, and technical skills throughout project development. They also get a bonus voucher worth 100 points for each member.

5.6 Constraints

There is a hard deadline for the delivery of this project on 24 May 2021. This implies that any delays to the schedule will reduce project scope, i.e. unimplemented features. As students are developing the project, the budget is $0. The development team consists of 6 students with different courses and schedules, making it challenging to arrange times to meet for group discussions. This project also has significant communication challenges due to the geographic

location, COVID situation, internet connection, and schedule variation of each member. Several team members are involved in other projects outside of this subject, and so their availability for the project is much lower than a full-time worker. Assuming each person has approximately 0.25 availability on a team of 6, this gives the equivalent of 1.5 full-time workers.

6. Project Governance

6.1 Roles and Responsibilities

The roles and responsibilities of each member of the project team are outlined below.

Business Owner (Setender)

- Represents MYD to the project
- Provide guidance on priority of features to implement
- Provide feedback on system releases to ensure alignment
- Set up agreements with local businesses participating

Project Manager (James)

- Oversee execution of project: monitoring team members' progress against project plan
- Coordinate updates to PMP
- Monitor and maintain risk register
- Monitor and maintain communications across team, and assist in conflict resolution
- Coordinate with Business Owner to ensure alignment
- Provide project status updates to Business Owner

Senior User (Alastair)

- Engage in testing of system
- Coordinate with UX professional
- Provide feedback on experience with voucher system

Developers (Sakshi, Xin, Setender)

- Implement core functionality of system as directed by PM
- Receive guidance from SME when technical issues arise

Technical SME (Xin)

- Provide guidance on technical implementation details
- Guide high level design decisions

UX Professional (Decy)

- Research on potential user
- Research on similar applications
- Create design prototype
- Usability testing

6.2 Communication Plan

Communication with project stakeholders has been documented in the communication matrix below. This captures the plan for regular communication, as well as contingencies in case the above approaches fail.

Day to day team communication will be via WhatsApp. Weekly team meetings will be held on Zoom to review progress and address issues and help needed. Ad hoc meetings will be scheduled as needed for collaboration.

The escalation of team communications will follow a hierarchy from first to last of email, WhatsApp message, Zoom call. In the event that planned contingencies fail, a team member is unable to be contacted, or a conflict is unable to be resolved, university staff will be contacted for assistance.

Communication matrix

Communication	Goal	Format	Frequency	Owner	Audience	Importance
Project status report	Provides information about project progress, completed work, main issues and summary schedule.	Formal report via email	Weekly	Project manager	Tianyi (MYD owner)	high
	Inform all team members that the product will be released.	Release notes via email	Every release	Technical Subject Matter	Project Team	medium
Team meeting	Discuss each member's work progress from the previous meeting, what their next plans are, and find solutions if there are any obstacles or issues.	Virtual meeting (zoom) Meeting minutes available on google drive for the member who is unable to attend the meeting. *Contingency plan*: - Discussion via Whatsapp - Meeting minutes via email	Weekly	Project Manager	Project Team	high
	Discuss alternative solutions if urgent problems arise	Virtual meeting (zoom) Meeting minutes available on google drive for the member who unable to attend the meeting *Contingency plan*: - Discussion via Whatsapp - Meeting minutes via email	Ad Hoc	Project Manager	Project Team	high

Task progress updates	Notify team members about assigned tasks, and provide daily progress information	Discussion via Whatsapp and/or Email notification (from monday. com)	Daily	Project Manager	Project Team	high
Project review	Present and review project outcomes, collect feedback and discuss the nextactions	Meeting face to face *Contingency plan*: - Virtual meeting (zoom) - Discussion with instant message - Formal report via email	Every milestone	Project Manager	Project Team, Tianyi (MYD owner)	high
Post mortem meeting	Evaluate project outcomes, identify if there are any undeliverable features, and discuss what actions to take.	Virtual meeting (zoom) Meeting minutes available on google drive for the member who unable to attend the meeting *Contingency plan*: - Discussion via Whatsapp - Meeting minutes via email	At the end of the project	Project Manager	Project Team	high
Product release	Present project outcomes and discuss website or new feature hosting on MYD's local server.	Meeting face to face *Contingency plan*: - Virtual meeting (zoom) - Discussion with instant message - Formal report via email	Every release	Project Manager	Project team, MYD's IT division, Business owner	high
	Provide information to the customer (MYD employees) and MYD owner if the website ready to use or new features is launched	Information via email	Every release	Project Manager	MYD's employees , Business owner	low

6.3 Risk Management

A number of risks have been identified and analysed for the project. The risk impact analysis, and risk response analysis for the top 5 risks are presented in the tables below.

Impact	**High**	11	7, 16	
	Med		5	
	Low		3	
		Low	**Med**	**High**
		Likelihood		

Risk Matrix for top 5 project risks. Number listed is Risk ID.

Risk Impact Analysis

Risk ID	Risk Type	Description	Probability (Low 1, High 10)	Impact (Low 1, High 10)	Exposure	Justification
7	Project	Inaccurate estimate of student availability	5	8	40	Students undertake several subjects and may also work at the same time. Availability may be overestimated by the inexperienced project team, producing schedule impact and/or failure to meet specification.
16	Product	Email notification service unable to deliver emails and false spam detections.	5	7	35	Email is a key part of booking confirmation/cancellation for this application. The backend mail delivery may fail. Recipient mail service providers (gmail, outlook, telstra) may mark notification emails as spam.
5	Project	Participating businesses modify the type of goods and services they are willing to provide	5	5	25	Businesses often adjust the goods and services provided. Addition of different services is out of scope. A change in the types of goods and services may require modification to the data model and web interface, with impact to schedule for the update.
11	Product	MYD IT department rules prevent local deployment of web service.	2	10	20	The IT department may be unwilling to host the service (e.g. security concerns, lack of budget). The impact is high because the service would not be able to be deployed.
3	Product	Local business owner pulls out of voucher program	5	2	10	COVID has decimated local businesses, so there's no guarantee they'll survive to product deployment. This could mean bookings already made are unable to be honoured. It is beyond the current project scope to allow removal of services via the web interface.

Risk Register

Risk ID	Trigger	Owner	Response	Strategy	Resources Required
7	Release 1 does not have all of the planned functionality	PM	Mitigate - monitor progress per project schedule. Adjust project schedule between releases. May also need to reduce scope.	Mitigate	Project schedule tools to track progress and highlight shortfall early
16	Release 2 deployment shows emails being blocked	Technical SME	As initial release is intended to be hosted locally, MYD IT team can be handed the responsibility of ensuring email is able to pass through.	Transfer	Technical SME, PM should get buy-in on the transfer from Tianyi, and then inform MYD IT team
5	Business owner indicates they are modifying their goods/services	PM	Coordinate with participating business owners to find out early if they are planning to modify goods/services. Lock in business owners' goods/services with agreements, and indicate that changes may not be able to be supported in the project.	Mitigate	Tianyi, PM to ensure business owner agreements are in place
11	Final release is unable to be deployed by MYD	Technical SME	Coordinate with MYD IT team early in project to ensure that they are able to support the application	Mitigate	Technical SME to meet with MYD IT team
3	Local business indicates they no longer wish to participate in the program	PM	The MYD admin can reject/cancel vouchers for businesses that pull out.	Transfer	Get buy in from Tianyi on this approach. PM to notify MYD admin of the possibility and how to address it.

6.4 Technology

The software product is a web app that requires authentication, multiple users roles, a persistent storage and sending emails. The following language/technologies/frameworks were researched.

Wordpress (PHP, MySQL, Apache or Nginx)

Wordpress is popular for web development, with built-in user authentication and user role management. Its content management system makes it suitable to use as a blogging platform. Wordpress comes with a large selection of first and third party plugins. Woo-commerce is a plugin that offers the ability to convert Wordpress into a production ready ecommerce platform with ease. Voucher_services could benefit from its out of the box ecommerce features. However, Wordpress requires each developer to set up their own local development environments, such as MySQL, Apache or nginx server. Any changes made to the database schema would require every developer to synchronize their environment. Furthermore, as all developers work on the same codebase, code merges could introduce some issues and overheads. As no team member has expertise in PHP, and given the small team size, meeting project schedules could be difficult using this approach.

React + Firebase (Typescript, Firebase Firestore, Cloud Functions, Firebase Authentication)

Firebase offers authentication, NoSQL database, hosting and serverless functions out of the box. It enables developers to focus on coding rather than setting up development environments. Firebase offers a generous free tier, meaning developers can use these services without charges. Developers may choose to use live services directly or local emulators freely. Firebase's Cloud Function offers several benefits, such as scalability, code splitting, and built-in event triggers with other Firebase offerings such as Firestore. It allows developers to use a separate code base for backend logic therefore reduce code merging issues. It is also highly failure resilient by allowing code to run in parallel to ensure errors in a single function do not crash the entire system.

React is one of the most popular web frameworks. It has a rich ecosystem, a large amount of learning resources and ongoing support from communities as well as large corporations such as Facebook, Airbnb and Microsoft. Building user interfaces with React is declarative and simple, making it ideal for a team of students to learn. React is rendered on the client side eliminating

the need to set up a web server for this project. The flexibility of React allows developers to integrate with any other frameworks, making changes easy to implement.
However, only one team member has experience with React and Firebase.

Django

Django is a popular web framework, it is free to use and simple to deploy. It offers better flexibility compared to Wordpress as it does not require MySQL and can work with most databases. Django is Python based, a language all team members have experience with. Its abstraction and tools are easy to learn. Authentication and user role functionalities are out of the box.
However, the monolithic nature of Django makes it difficult to use for this project. All components are deployed together, producing a higher chance of some component crashing the whole site than other alternatives. Developers are required to understand the full system before they can contribute efficiently. Django's Object Relational Mapper could potentially create challenges when integrating with other systems. Furthermore, Django templates still require developers to write HTML and CSS but not many team members have substantial experience with HTML or CSS.

Spring Boot

Spring Boot is a popular web framework with a large amount of learning resources and community support. It works with most databases and allows easy integration with other Spring ecosystems. It offers many production ready features such as embedded server and health check. Managing dependencies are made easy with starter POMs.
However, Spring boot has a steep learning curve and only one member has experience with it. Without understanding of Spring's underlying system, developers may have difficulties with troubleshooting and modifications.

Project Stack

The React and Firebase combination has been chosen for this project, because in comparison to Wordpress, Django and Spring Boot, React and Firebase offer more features out of the box. With Firebase's hosting offering, deployment can be done via a single command. CDN, versioning and traffic routing is automatically configured for free. Other technologies are more time consuming to deploy. React utilizes Virtual DOM which greatly improves user experience by providing responsive user interactions and reducing loading times between pages. Other

frameworks such as Wordpress bundle large amounts of unused Javascript and CSS on every page heavily impacting page load speed. Firebase's Firestore database is the only database that does not require a server, and the React frontend is able to call the database directly. User interactions such as service booking and profile updates can be completed near instantly, which further enhances user experience. Other technologies require making API calls to the server first, then server to database, producing a longer delay. Firestore is a NoSQL database that is highly scalable without resource provisioning overhead, making it more future proof than other databases such as MySQL used by Wordpress. Firebase's Cloud Function is available in multiple languages, allowing all team members to choose a language they feel comfortable coding with and contribute to the backend logic. In contrast, Wordpress, Django and Spring boot, all require code to be written in a specific language. Cloud Function's micro service architecture allows deployment to be independent down to the function level. Therefore React and Firebase offer better availability than Django or Wordpress, as failure in any component will not bring down the entire system. Firebase Authentication integrates seamlessly with other Firebase products. It provides end to end identity solutions, freeing developers from writing any code for handling role based access control, such as adding services as an admin.

Overall, while other technologies/frameworks each have their own perks, the React and Firebase combination drastically outshines the others in terms of flexibility, development acceleration, availability, product quality and synergy with team members' skills.

6.5 Project Planning

The project will be delivered on two releases to be completed on 7 May 2021 and 21 May 2021 respectively. The first release will deliver the core functionality of user login and voucher bookings. The second release will extend the system by implementing features that use email integration, as well as features involving the update of user information and booking cancellation.

Project Schedule

The Project Manager is responsible for maintaining the project schedule and tracking team progress against it. The schedule will be tracked and updated weekly following status meetings to establish whether the project is progressing as planned. A larger update to the project schedule will take place between product releases as there will be a better understanding of team productivity.

The project schedule has been implemented using monday.com (see Gantt Chart). It is presented as a Gantt chart, showing the project timeline with dependencies and milestones.

Estimation of resources required for tasks was based on team consensus, and is expected to have a high uncertainty as student availability and productivity are not well known at project commencement. Task allocation (see Work breakdown structure and resource allocation) was then based on nominal roles, and adjusted for balance in workload across the team.

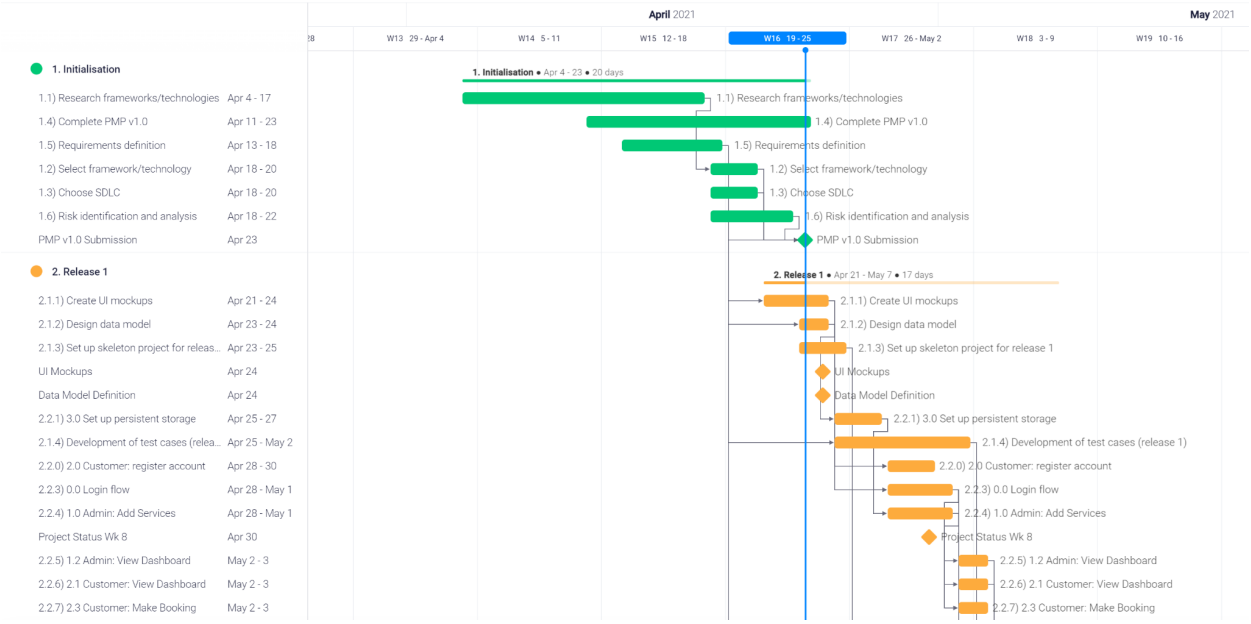
Project Schedule Artefacts

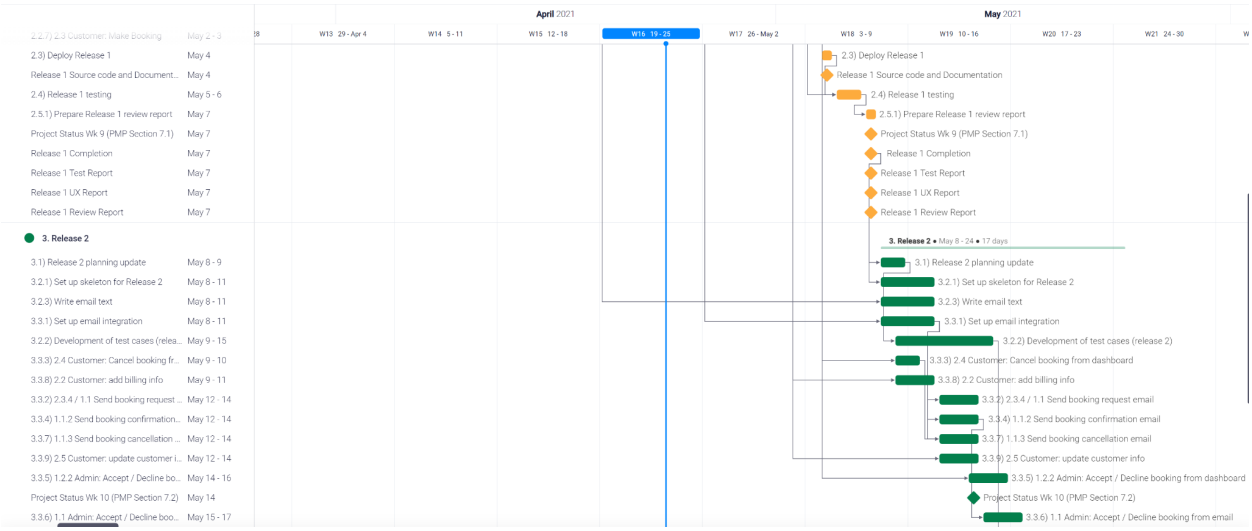
The definition of the project schedule is maintained at the following links:

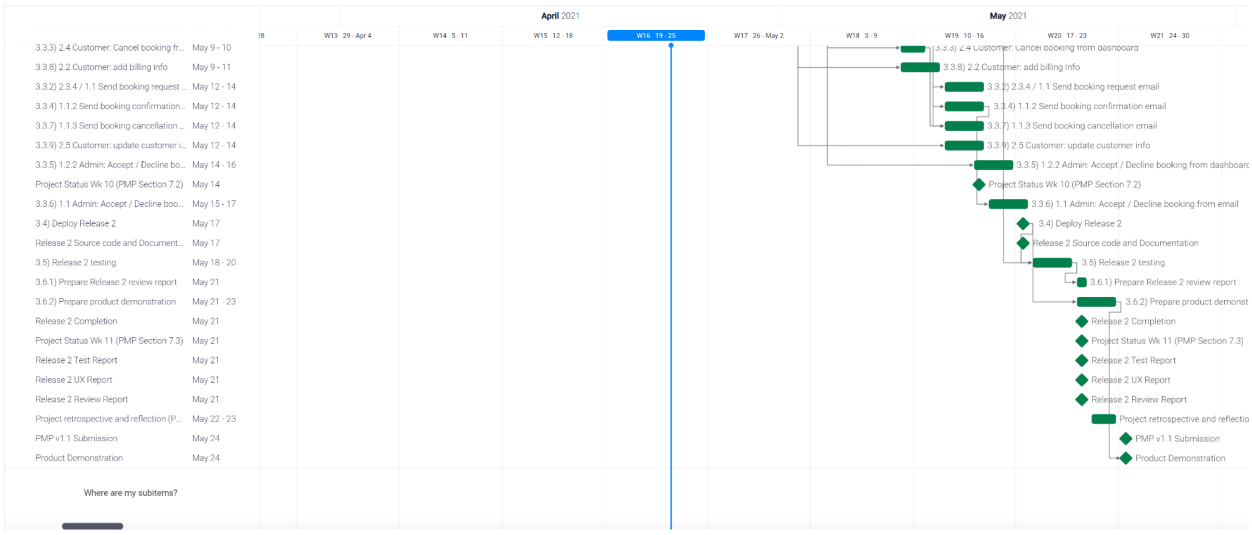
- Gantt Chart
- Work breakdown structure and resource allocation
- Timeline per team member

Gantt Chart

The definition of the Gantt chart is maintained in the Project Schedule. Screenshots are shown below for reference.


April 2021
May 2021
W13 29 - Apr 4
W14 5 - 11
W15 12 - 18
W16 19 - 25
W17 26 - May 2
W18 3 - 9
W19 10 - 16
1. Initialisation
1.1) Research frameworks/technologies Apr 4 - 17
1.4) Complete PMP v1.0 Apr 11 - 23
1.5) Requirements definition Apr 13 - 18
1.2) Select framework/technology Apr 18 - 20
1.3) Choose SDLC Apr 18 - 20
1.6) Risk identification and analysis Apr 18 - 22
PMP v1.0 Submission Apr 23
1. Initialisation • Apr 4 - 23 • 20 days
1.1) Research frameworks/technologies
1.4) Complete PMP v1.0
1.5) Requirements definition
1.2) Select framework/technology
1.3) Choose SDLC
1.6) Risk identification and analysis
PMP v1.0 Submission
2. Release 1
2.1.1) Create UI mockups Apr 21 - 24
2.1.2) Design data model Apr 23 - 24
2.1.3) Set up skeleton project for releas... Apr 23 - 25
UI Mockups Apr 24
Data Model Definition Apr 24
2.2.1) 3.0 Set up persistent storage Apr 25 - 27
2.1.4) Development of test cases (relea... Apr 25 - May 2
2.2.0) 2.0 Customer: register account Apr 28 - 30
2.2.3) 0.0 Login flow Apr 28 - May 1
2.2.4) 1.0 Admin: Add Services Apr 28 - May 1
Project Status Wk 8 Apr 30
2.2.5) 1.2 Admin: View Dashboard May 2 - 3
2.2.6) 2.1 Customer: View Dashboard May 2 - 3
2.2.7) 2.3 Customer: Make Booking May 2 - 3
2. Release 1 • Apr 21 - May 7 • 17 days
2.1.1) Create UI mockups
2.1.2) Design data model
2.1.3) Set up skeleton project for release 1
UI Mockups
Data Model Definition
2.2.1) 3.0 Set up persistent storage
2.1.4) Development of test cases (release 1)
2.2.0) 2.0 Customer: register account
2.2.3) 0.0 Login flow
2.2.4) 1.0 Admin: Add Services
Project Status Wk 8
2.2.5) 1.2 Admin: View Dashboard
2.2.6) 2.1 Customer: View Dashboard
2.2.7) 2.3 Customer: Make Booking


April 2021
May 2021
W13 29 - Apr 4
W14 5 - 11
W15 12 - 18
W16 19 - 25
W17 26 - May 2
W18 3 - 9
W19 10 - 16
W20 17 - 23
W21 24 - 30
2.2.7) 2.3 Customer: Make Booking May 2 - 3
2.3) Deploy Release 1 May 4
Release 1 Source code and Document... May 4
2.4) Release 1 testing May 5 - 6
2.5.1) Prepare Release 1 review report May 7
Project Status Wk 9 (PMP Section 7.1) May 7
Release 1 Completion May 7
Release 1 Test Report May 7
Release 1 UX Report May 7
Release 1 Review Report May 7
3. Release 2
3.1) Release 2 planning update May 8 - 9
3.2.1) Set up skeleton for Release 2 May 8 - 11
3.2.3) Write email text May 8 - 11
3.3.1) Set up email integration May 8 - 11
3.2.2) Development of test cases (relea... May 9 - 15
3.3.3) 2.4 Customer: Cancel booking fr... May 9 - 10
3.3.8) 2.2 Customer: add billing info May 9 - 11
3.3.2) 2.3.4 / 1.1 Send booking request ... May 12 - 14
3.3.4) 1.1.2 Send booking confirmation... May 12 - 14
3.3.7) 1.1.3 Send booking cancellation ... May 12 - 14
3.3.9) 2.5 Customer: update customer i... May 12 - 14
3.3.5) 1.2.2 Admin: Accept / Decline bo... May 14 - 16
Project Status Wk 10 (PMP Section 7.2) May 14
3.3.6) 1.1 Admin: Accept / Decline boo... May 15 - 17
2.3) Deploy Release 1
Release 1 Source code and Documentation
2.4) Release 1 testing
2.5.1) Prepare Release 1 review report
Project Status Wk 9 (PMP Section 7.1)
Release 1 Completion
Release 1 Test Report
Release 1 UX Report
Release 1 Review Report
3. Release 2 • May 8 - 24 • 17 days
3.1) Release 2 planning update
3.2.1) Set up skeleton for Release 2
3.2.3) Write email text
3.3.1) Set up email integration
3.2.2) Development of test cases (release 2)
3.3.3) 2.4 Customer: Cancel booking from dashboard
3.3.8) 2.2 Customer: add billing info
3.3.2) 2.3.4 / 1.1 Send booking request email
3.3.4) 1.1.2 Send booking confirmation email
3.3.7) 1.1.3 Send booking cancellation email
3.3.9) 2.5 Customer: update customer info
3.3.5) 1.2.2 Admin: Accept / Decline booking from dashboard
Project Status Wk 10 (PMP Section 7.2)
3.3.6) 1.1 Admin: Accept / Decline booking from email


April 2021
May 2021
W15 12 - 18
W16 19 - 25
W17 26 - May 2
W18 3 - 9
W19 10 - 16
W20 17 - 23
W21 24 - 30
3.3.3) 2.4 Customer: Cancel booking fr... May 9 - 10
3.3.8) 2.2 Customer: add billing info May 9 - 11
3.3.2) 2.3.4 / 1.1 Send booking request ... May 12 - 14
3.3.4) 1.1.2 Send booking confirmation... May 12 - 14
3.3.7) 1.1.3 Send booking cancellation ... May 12 - 14
3.3.9) 2.5 Customer: update customer i... May 12 - 14
3.3.5) 1.2.2 Admin: Accept / Decline bo... May 14 - 16
Project Status Wk 10 (PMP Section 7.2) May 14
3.3.6) 1.1 Admin: Accept / Decline boo... May 15 - 17
3.4) Deploy Release 2 May 17
Release 2 Source code and Document... May 17
3.5) Release 2 testing May 18 - 20
3.6.1) Prepare Release 2 review report May 21
3.6.2) Prepare product demonstration May 21 - 23
Release 2 Completion May 21
Project Status Wk 11 (PMP Section 7.3) May 21
Release 2 Test Report May 21
Release 2 UX Report May 21
Release 2 Review Report May 21
Project retrospective and reflection (P... May 22 - 23
PMP v1.1 Submission May 24
Product Demonstration May 24
Where are my subitems?
3.3.8) 2.2 Customer: add billing info
3.3.2) 2.3.4 / 1.1 Send booking request email
3.3.4) 1.1.2 Send booking confirmation email
3.3.7) 1.1.3 Send booking cancellation email
3.3.9) 2.5 Customer: update customer info
3.3.5) 1.2.2 Admin: Accept / Decline booking from dashboard
Project Status Wk 10 (PMP Section 7.2)
3.3.6) 1.1 Admin: Accept / Decline booking from email
3.4) Deploy Release 2
Release 2 Source code and Documentation
3.5) Release 2 testing
3.6.1) Prepare Release 2 review report
Release 2 Completion
Project Status Wk 11 (PMP Section 7.3)
Release 2 Test Report
Release 2 UX Report
Release 2 Review Report
PMP v1.1 Submission
Product Demonstration

Work-breakdown structure (WBS)

The definition of the WBS is maintained in the Project Schedule. It is replicated here for reference.

1. Initialisation
    - 1.1. Research frameworks/technologies
    - 1.2. Select framework/technology
    - 1.3. Choose SDLC
    - 1.4. Complete PMP v1.0
    - 1.5. Requirements definition
    - 1.6. Risk identification and analysis
2. Release 1
    - 2.1. Design
        - 2.1.1. Create UI mockups
        - 2.1.2. Design data model
        - 2.1.3. Set up skeleton project for release 1
        - 2.1.4. Development of test cases (release 1)
    - 2.2. Implementation*
        - 2.2.1. 3.0 Set up persistent storage
        - 2.2.2. 2.0 Customer: register account
        - 2.2.3. 0.0 Login flow
        - 2.2.4. 1.0 Admin: Add Services
        - 2.2.5. 1.2 Admin: View Dashboard
        - 2.2.6. 2.1 Customer: View Dashboard
        - 2.2.7. 2.3 Customer: Make Booking
    - 2.3. Deploy Release 1
    - 2.4. Release 1 testing
    - 2.5. Release 1 review
3. Release 2
    - 3.1. Release 2 planning update
    - 3.2. Design
        - 3.2.1. Set up skeleton for Release 2
        - 3.2.2. Development of test cases (release 2)
        - 3.2.3. Write email text
    - 3.3. Implementation*
        - 3.3.1. Set up email integration
        - 3.3.2. 2.3.4 / 1.1 Send booking request email
        - 3.3.3. 2.4 Customer: Cancel booking from dashboard
        - 3.3.4. 1.1.2 Send booking confirmation email
        - 3.3.5. 1.2.2 Admin: Accept / Decline booking from dashboard
        - 3.3.6. 1.1 Admin: Accept / Decline booking from email
        - 3.3.7. 1.1.3 Send booking cancellation email
        - 3.3.8. 2.2 Customer: add billing info

3.3.9. 2.5 Customer: update customer info
3.4. Deploy Release 2
3.5. Release 2 testing
3.6. Release 2 review
3.6.1. Prepare release 2 review report
3.6.2. Prepare product demonstration

*Implementation task descriptions beginning with numbers correspond to use cases. Refer to Use Case Documentation (see Appendix A) for further details. The corresponding task is the implementation of the functionality to fulfill the use case.

Project Artefacts

Key project artefacts include:

Artefact	**Owner**	**Description**
Project Management Plan	James	Document planning, control, execution of the project
Meeting minutes	James	Minutes of meetings, including agenda, discussion, action items
Project Schedule	James	Gantt chart showing breakdown of tasks, resource allocation, dependencies
Use Cases	Alastair	Detailed description of project requirements in the form of use cases
UI Mockups	Decy	Mockups of different UI elements
Data Model Definition	Xin	Definition of the data model for persistent storage
Release 1 Test Report	Alastair	Test results for Release 1
Release 1 UX Report	Decy	Assessment of UX of Release 1
Release 2 Test Report	Alastair	Test results for Release 2
Release 2 UX Report	Decy	Assessment of UX of Release 2
Release 1 Review Report	Setender	Confirm alignment of Release 1 to MYD goals
Release 2 Review Report	Setender	Confirm alignment of Release 2 to MYD goals
Project retrospective and reflection	James	Reflection on teamwork and project

Source code and documentation	Sakshi	Definition of application source code, and appropriate documentation to support future development and maintenance

Weekly milestones

The weekly milestones are maintained in the Project Schedule. They are replicated here for reference.

Week 7

- PMP v1.0 submission

Week 8

- UI Mockups
- Data Model Definition
- Project status

Week 9

- Release 1
- PMP 7.1 Project status
- Release 1 Test Report
- Release 1 UX Report
- Release 1 Review Report
- Release 1 Source code and documentation

Week 10

- PMP 7.2 Project status complete

Week 11

- Release 2 complete
- Release 2 Test Report
- Release 2 UX Report
- Release 2 Review Report
- Release 2 Source code and documentation
- PMP 7.3 Project status

Week 12

- Project Retrospective and Reflection
- Project demonstration
- PMP v1.1 submission

Appendix A: Use Cases

0.0 Login

In this section, “resource” may refer to any URL associated with the service, including:

- URLs used to serve web pages
- URLs used to access REST or database resources.

0.0.0 Login page

As a CUSTOMER or ADMIN of the service,
When I try to access any component of the service without being first authenticated,
I want to be presented with a login screen requiring me to enter my:

- email address, and
- password

This information should be encrypted in transit, and at rest (if it is ever stored in a persistent manner).

0.0.1 Login Failure

As a CUSTOMER or ADMIN of the service,
When I enter invalid credentials during a login interaction,
I want be notified of login failure, and not given access to the resource I was trying to access.

0.0.2 Login Success with invalid destination

As a CUSTOMER of the service,
When I enter invalid credentials during a login interaction,
And the original resource I was trying to access was an ADMIN-RESTRICTED resource,
I want to access an error page.

0.0.3 Login Success with valid destination

As a CUSTOMER of the service,
When I enter valid credentials during a login interaction,
And the original resource I was trying to access was NOT an ADMIN-RESTRICTED resource,
I want to access the resource I was trying to access.

1. ADMIN

1.0 CRUD service types

1.0.0 Add service types

As the ADMIN, I want to be able to add new service types to the list of service types accessible by vouchers.
The SERVICE TYPE information should include:

- Display name
- Point value (currently out of scope)
- Available delivery types

1.0.1 View service types

OUT OF SCOPE
As the ADMIN, I want to be able to view the list of service types accessible by vouchers.

1.0.2 Remove service types

OUT OF SCOPE

1.1 Bookings

1.1.0 Notification

As the ADMIN,
When a CUSTOMER makes a new PENDING BOOKING using the service,
I want to receive an email containing the following information:

- CUSTOMER name
- CUSTOMER phone number
- CUSTOMER email address
- PENDING BOOKING date and time
- PENDING BOOKING message included by the CUSTOMER

The email should also include:

- A button I can use to accept the PENDING BOOKING
- A button I can use to decline the PENDING BOOKING

At this stage in development, validating this information is OUT OF SCOPE (invalid emails / phone numbers are ok)

1.1.1 Accept / Decline buttons in notification email require login

As the ADMIN,
When I click the accept or decline button included in the email in Notification while not

logged in,
I want to be taken to a login page,
and on success taken through the accept / decline flow as described in:

- 1.1.2 (accept button), and
- 1.1.3 (decline button)

1.1.2 Accept button after login

As the ADMIN,
When I click the accept button included in the email in Notification while not logged in,
IF the PENDING BOOKING object still exists,

- An email confirming the booking should be sent to the email associated with the CUSTOMER who made the booking.
  The email should include:
  - The CUSTOMER Name, Phone number, and Email address
  - The PENDING BOOKING date, time, and delivery type
- The PENDING BOOKING should be replaced with a BOOKING object
- I should see a confirmation page showing the details of the booking that was accepted, using the same information as shown in the original email.

1.1.3 Decline button after login

As the ADMIN,
When I click the accept button included in the Notification email while logged in,
IF the PENDING BOOKING object still exists,

- An email stating that the booking has been declined should be sent to the email associated with the CUSTOMER who made the booking.
  The email should include:
  - The CUSTOMER Name, Phone number, and Email address
  - The PENDING BOOKING date, time, and delivery type
- The PENDING BOOKING should be deleted
- I should see a confirmation page showing the details of the booking that was cancelled, using the same information as shown in the original email.

1.1.4 Accept / Decline of cancelled booking

As the ADMIN,
When I click the accept button included in the Notification email while logged in,
IF the PENDING BOOKING object no longer exists,
I want to see a page informing me that the booking has already been cancelled.

1.2 Admin dashboard

1.2.1 Dashboard requires login

As the ADMIN,
When I try to access my dashboard while not logged in,
I want to be taken through the login flow,
And taken to the dashboard on success

1.2.2 Dashboard

As the Admin,
When I access my dashboard while logged in,
I want to be able to access a list of all PENDING BOOKING objects, with the following information for each:

- Customer name
- Date and time
- Service type
- Accept / Decline buttons (behaviour as described here)

2. CUSTOMER

2.0 Registration

2.0.0 Registration

As a CUSTOMER
I want to be able to register my profile with the system by accessing a web url
I expect to need to provide:

- My name
- My contact phone number
- My email address
- A password

After registration, I expect to be taken to the dashboard.
At this stage, phone number and email address verification are out of scope.

2.1 Dashboard

2.1.0 Dashboard requires login

As a CUSTOMER,
After registering my profile with the system,
When I try to access my dashboard while not logged in,

I want to be taken through the login flow

2.1.1

As a CUSTOMER,
When I access my dashboard while logged in, I want to be able to access

- A way to access the billing info flow
- A way to access the update customer info flow
- A way to access the add booking flow
- A list of future bookings, with a way to access the cancel booking flow for each

2.2 Add billing info

2.2.0 Add billing info requires login

As a CUSTOMER, after registering my profile
When I try to add my billing information while not already logged in
I want to be taken through the login flow,
and directed to the add billing info page on success

2.2.1 Add billing info page

As a CUSTOMER, after registering my profile
When I try to add my billing information while logged in,
I want to be able to enter my booking information in a form.
The information should include:

- The name to be included in any invoice sent to me
- The email address to send any invoices to

2.3 Add Booking

2.3.0 Add Booking requires login

As a CUSTOMER, after registering my profile,
When I try to access the add-booking flow while not logged in
I want to be taken through the login flow,
and directed to the add booking page on success.

2.3.1 Add Booking page

As a CUSTOMER, after registering my profile
When I try to access the add-booking flow while logged in
I want to be able to enter my desired booking information in a form
This information should include:

Service type (from a drop-down)

Delivery type (Delivery or Pick-up)
The date and time the delivery should take place
A message to associate with the booking, which may be blank

None of these values should have pre-filled content.

2.3.2 Booking submission button

As a CUSTOMER, after registering my profile
When I have entered booking information on the add booking page,
And that information includes at least:

- Service type,
- Delivery type,
- Date and time

A button should become available that, when clicked, will create a PENDING BOOKING using the information I have described

2.3.3 Booking submission button availability

As a CUSTOMER, after registering my profile
When I have entered booking information on the add booking page,
And that information does not include at least:

- Service type,
- Delivery type,
- Date and time

The booking submission button should be inactive or otherwise unavailable.

2.3.4 Pending booking creation notifies admin

This is the same as Admin: bookings

2.3.5 Booking failure - OUT OF SCOPE

As a CUSTOMER,
When I attempt to make a booking
If I have an insufficient number of points in my CUSTOMER INFORMATION
I want an error message to be displayed
AND the PENDING BOOKING to not be created
AND to points to be deducted from my CUSTOMER INFORMATION

2.3.6 Booking success - points balance - OUT OF SCOPE

As a CUSTOMER,
When I make a booking,
I want the appropriate point value to be deducted from my CUSTOMER INFORMATION, if possible

2.4 Cancel Booking

2.4.0 Cancel button shows a confirmation dialog

As a CUSTOMER,
When I click the “cancel booking” button on the dashboard
I want to be presented with a confirmation dialog requiring me to confirm my intention to cancel the booking.
The dialog should have two buttons (“confirm”, and “cancel”)
The default action should be “confirm”.

When the “cancel” button is clicked, the dialog should be closed.
When the “confirm” button is clicked:

1. The BOOKING or PENDING BOOKING object should be deleted
2. The dashboard should show some kind of confirmation that this has occurred

2.4.1 Cancelling a booking updates the point balance - OUT OF SCOPE

As a CUSTOMER,
When I confirm that I want to cancel a booking,
I want the POINT VALUE of the BOOKING or PENDING BOOKING to be added to my CUSTOMER INFORMATION

2.5 Update customer info

2.5.0 Update customer info flow

As a CUSTOMER,
When I click the “update profile” button on the dashboard,
I want to see a form containing fields for my CUSTOMER INFORMATION and BILLING INFORMATION
These fields should all be editable
A “submit” button should be available - as in registration, no validation is required.

2.5.1 Submit button

As a CUSTOMER,
When I click the “submit” button on the update info flow,
I want my details to be updated in the database,
And a confirmation of some kind to be displayed

3.0 Persistence

The following items need to be stored in persistent storage:

List of actors:

ADMIN
- Name
- Email

CUSTOMER
- CUSTOMER INFORMATION
- BILLING INFORMATION

List of types:

SERVICE TYPE
- Point value

PENDING BOOKING
- Service type (required)
- Delivery type (delivery / pick-up) (required)
- Booking date and time (required)
- Message
- (Point value? would help mitigate against price changes)

BOOKING
- Same fields as PENDING BOOKING

CUSTOMER INFORMATION
- Name
- Email
- Phone number
- Number of points

BILLING INFORMATION
- Biller name
- Biller email