TableMatter – REST API and mobile app

Project Plan

Project Development Team

Name	Position	email	phone
Jin Hoe Lim	Team Lead Software developer	101458985@student.swin.edu.au	042 041 0533
Lachlan O'Neill	Software Architect Software developer	101631391@student.swin.edu.au	0413 792 160
Lauren Stampoultzis	Project Manager Software developer	101634264@student.swin.edu.au	
Travis Wheeler	Front-end developer Software developer	100580065@student.swin.edu.au	
Trent Williams	UX Designer Software developer	101448465@student.swin.edu.au	0488 374 812
William Middlemiss	Software developer	101572681@student.swin.edu.au	0438 236 683

SWE40001, Software Engineering Project A, Semester 1 2020

CONTENTS

DOCUMENT CHANGE CONTROL

Version	Date	Authors	Summary of Changes
0.1	27/03/20	Jin, Lachlan, Lauren, Travis, Trent and Will	First draft of project plan.
0.2	28/03/20	Jin, Lachlan, Lauren, Travis, Trent and Will	Second draft of project plan. Changes based on feedback from Project Academic Supervisor.
1.0	31/03/20	Jin	Grammatical and spelling corrections. Added project timeline + Gantt Chart. Finished project budget section. Final version for client sign-off.

DOCUMENT SIGN OFF

Name	Position	Signature	Date
Jin Hoe Lim	Team Leader	Jin Hoe Lim	31/03/20
Lachlan O’Neill	Technical Lead	L.O’Neill	1/04/20
Lauren Stampoultzis	UI Designer UX Testing Lead	LStampoultzis	01/04/20
Travis Wheeler	Software Engineer	Twheeler	31/03/20
Trent Williams	Interface Design Lead	TWilliams	31/03/20
William Middlemiss	Software Architect	William Middlemiss	01/04/20

CLIENT SIGN OFF

Name	Position	Signature	Date
Organisation			

1. INTRODUCTION

This document is intended as a comprehensive overview of the project structure, intended outcomes, scope, risks and development details such as schedule, requirements, budget. This document should be referenced when required by the project stakeholders throughout the duration of the project. This document should be reviewed, and formally approved and signed by the client.

This document will assist the project development by defining a clear project scope and establishing correct procedure for the team to follow as well as Identifying key acceptance criteria for the product.

1.1. BACKGROUND

qThe driving force of this project is to create a restaurant table booking application. As most people own a smartphone, digital applications are crucial. As consumers are expecting to complete simple tasks and perform everything on their device, a need for creating this application was required. Instead of calling and making appointments and reservations, this application will hopefully allow easy and quick bookings and increase the amount of customers. [MORE INFO NEEDED]

1.2. KEY PROJECT PERSONNEL

The individuals who will contribute to the execution of this project are as follows:

1.2.1. CLIENT

Mahesh Samaradiwakara

1.2.2. OTHER STAKEHOLDERS

- Each team member for the project
  - Jin Hoe Lim - 101458985@student.swin.edu.au
  - Lachlan O’Neill - 101631391@student.swin.edu.au
  - Lauren Stampoultzis - 101634264@student.swin.edu.au
  - Travis Wheeler - 100580065@student.swin.edu.au
  - Trent Williams - 101448465@student.swin.edu.au
  - William Middlemiss - 101572681@student.swin.edu.au
- Each team member for other project (Tablematter Admin & Super Admin Dashboard)
- Project Academic Supervisor (Supervises and assists, where required, the project to support the student group and ensure the project is on track technically and organisationally)
  - Dr Sivachandran Chandrasekaran - schandrasekaran@swin.edu.au
- Software Engineering Project course coordinators

1.2.3. PROJECT SUPERVISOR, TEAM LEADER AND KEY PROJECT MEMBERS

- Sivachandran Chandrasekaran: Project Supervisor
- Jin Hoe Lim: Team Leader
- Lachlan O’Neill: Technical Lead
- Lauren Stampoultzis: UI Designer and UX Testing lead.
- Travis Wheeler: Software Engineer
- Trent Williams: UI / Interface designer

- William Middlemiss: Software Architect

2. TERMS OF REFERENCE

The goal of this project is to develop an application that will be used by a diverse range of users with the aim of finding, booking, reviewing and ordering from various restaurants, using a graphical mobile application. This is accompanied by the ability to define and request specific reservations within the restaurant that may involve a specific arrangement of tables, seats, food and other utilities.

2.1. OBJECTIVES

The key objectives of the TableMatter - REST API and mobile application project include:

1. Key functional requirements defined in the scope of the project are delivered to a MVP standard

2. Any developed functional requirements receive rigorous testing to ensure they function and integrate properly with the rest of the solution.

3. The delivered applications integrate with aspects of the Admin and Super Admin dashboard project.

4. The application runs in an efficient and effective manner and delivers a comfortable and enjoyable user experience.

2.2. SCOPE

The following items are within the TableMatter - REST API and mobile application project scope:

- Create a customer profile containing information including:
  - The users email
  - The users name
  - The users phone number
  - Visual aspects to the following components **(functionality not currently in scope)**:
    - change password
    - reset password
- Create custom searches for restaurants and meals that allows for:
  - Searching for restaurants based off ingredient
  - Searching based off dietary requirements (vegetarian etc)
  - Searching based off name
  - Searching based off location
- Pre-order meals
  - This includes both ordering for take away and within the restaurant.
- Book Tables, with the following options
  - This includes visual chair selector (with layout of restaurant)
  - Time/ Date select
  - Booking size (number of people)

    - Pre-book meals (see above)
- Payment integration
    - The product should support the potential for future implementation of:
        - Paypal
        - Mastercard
        - Visa
- Review the restaurant
    - Give comments up to 150 characters
    - Rating 1-5 stars
- View the average traffic a restaurant has at a given time
    - Average based on day of the week combined with hour of day
- Share their booking through third party apps/ software including:
    - WhatsApp
    - Messenger
    - Text/ SMS
    - Email

The TableMatter – REST API and mobile application **will not** accomplish the following:

- Functionality for the following aspects of the application
    - User password management (Reset/ Change)
- Saving the users payment information/ attaching it to their account.
- Create metrics on user interaction with different restaurants
- Create a user interface for restaurants to set up accounts for the user facing application
- Generate usage reports
- Implement AI
    - This includes to monitor peak times for the restaurants based off previous bookings through the application
    - Finding and banning/ blocking free credit abusers
- Deploy the software solutions for continued use after the end of the project.
- Provide a complete and deliverable version of the solution.

2.3. PROJECT START AND END DATES

Project Start Date:

4th of April, 2020 - Expected Project Plan sign-off date

Project End Date:

1st November, 2020 - Scheduled end of 2020 Swinburne Higher Education Teaching Period

2.4. CRITICAL SUCCESS FACTORS

The TableMatter - REST API and mobile app has multiple different KPIs that will be used to determine the overall success of the project including:

- Responsible, precise and effective quality assurance measures. This includes unit tests, integration testing and user testing.
- Completing functional requirements in a timely manner, and ensuring they are delivered by the appropriate time according to the project plan
- An appropriate understanding of which requirements are the most important and valuable to the final product, these should be given a higher priority and more attention
- Effective construction and implementation of the system architecture to ensure a robust and efficient product

2.5. ACCEPTANCE CRITERIA

The following acceptance criteria are defined for the project to determine the requirements to be met before which the project deliverables can be accepted:

- The key functional requirements in the scope of the project have been developed and function correctly in the final product
- The user interface design and user experience for the application allows for the user to easily and quickly complete perform the functions and features in the application.
- The TableMatter Admin and Super Admin Dashboard project has been taken into account and functionality exists to integrate the 2 projects

3. ESTABLISHMENT

3.1. PROJECT PROCESSES, PROCEDURES AND STANDARDS

- The team will undertake Agile software development methodologies, particularly the SCRUM process framework, for all stages of development. This process framework has been chosen due to the familiarity of the concept to all members of the team, as well as the proven advantages of Agile development allowing for effective and efficient progress. The SCRUM network was also recommended by the Client to provide flexibility and improve direct collaboration.

- The code base will be hosted on an online Git repository allowing for effective version control and remote access to all members of the team. It is planned for read-only access to be granted to the client during the development process.

    - A prototype application will be prioritised as one of the first milestones of the development process. This will allow features to be continually developed and added to the prototype for testing by the client throughout the duration of the project.

- This project will consist of frequent communications with the client throughout the development stages to ensure that expectations are understood and achieved. The team aims to deliver multiple prototypes and hold routine meetings with the client during the development process.

- The software delivered in this project will be coded in the React Native framework following the concepts of Object Oriented Programming focusing on functional components. Open source libraries will be implemented into the software.

3.2. PROJECT ENVIRONMENT

Due to the ongoing COVID-19 global pandemic, certain considerations have been made to the project environments to ensure that the project development continues with minimal impact.

- This project will be developed by all team members working remotely within their homes. Routine virtual meetings with the team, supervisor and client, and the usage of various instant communication mediums, such as Zoom and Skype, ensures that all participating stakeholders remain in frequent contact.

- All devices including computers used in the development of this project will be of private property of each team member. It is planned that development computers will run on the Windows 10 OS with the exception of computers involved in native iOS app development to run on MacOS.

- The following tools and platforms will be used by the team to develop the project. Note that the list is subject to change during project development. This entails that there will be relevant user accounts created/ used on the tools and platforms:
    - Slack
    - Google Drive
    - GitLab

  - Adobe Creative Cloud
  - Adobe XD
  - Amazon AWS

- The database involved in this project is expected to be a shared schema by the software developed by the TableMatter Admin and Super Admin Dashboard project team. Both projects aim to use MongoDB as the DBMS.

- An Apple Developer account will be required to be set up for testing and deployment of the native iOS application.

- A free tier Amazon AWS EC2 environment will be used to host the back-end API plus the MongoDB database. This micro instance will run on Windows 10 Server OS and be used throughout the duration of the project.

- The native mobile applications and REST API will be developed so that the deployment of the mobile application will require mobile devices running iOS or Android operating systems, and all relevant signings that permit the native applications to be published on their respective application marketplaces. The REST API will be delivered as a deployable package to be hosted on a machine or virtual instance such as Amazon AWS EC2.
  The deployment of the deliverable software is not included in the scope.

3.3. PROJECT TEAM SKILL DEVELOPMENT REQUIREMENTS

The project scope currently is expected to be completed within the existing skill sets of the team members. However, not all team members possess the same skills.

Though all required skills are present in the team in at least one member, the following skills will be developed amongst the team members throughout the development stages to ensure that the team has the capability to deliver the project. Note that the list may be subject to change throughout the development of the project:

- Navigating and using GitLab
- Using Node Packet Manager
- Understanding and applying React Native concepts
- Developing native mobile applications based on a single code base

4. DELIVERABLES, ACTIVITIES AND CAPITAL RESOURCES

4.1. DELIVERABLES

The deliverables for the TableMatter REST API and Mobile Application project are as follows:

1. The Software Requirements Specification (SRS) document
2. The Project Plan (this document)
3. Visual application prototype (Adobe XD file)
4. Native Android TableMatter application (APK file)
5. Native iOS TableMatter application (IPA file)
6. Deployable TableMatter REST API

More deliverables will be listed and defined as the project is further defined.

4.1.1. SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

The purpose of this document is to outline, in detail, the requirements surrounding the TableMatter MVP software. It will describe the purpose as well as features of the system, including the systems current requirements, architecture and interfaces. The intended audience for this document includes all members of the project team and the primary client.

Each revision of the SRS will be approved by the supervisor before being delivered to the client and signed off by both the team and the client.

4.1.2. PROJECT PLAN

This document. The purpose of the project plan is to establish all relevant details related to this project in a succinct and structured format. The intended audience for this document includes all members of the project team, the project academic supervisor, and the primary client.

Each revision of the project plan will be approved by the supervisor before being delivered to the client and signed off by both the team and the client.

4.1.3. MOBILE APPLICATION

The key deliverable of the TableMatter REST API and Mobile application project is to develop a MVP as a proof of concept for the client. The application will be developed based on the high level overview provided by the client, which is as follows:

1. Customer profile
2. Custom Search Restaurant/Meal
3. Meal ordering
4. Table booking - graphical representation of tables
5. Payment integration
6. Review restaurant
7. Rest API for mobile app
8. Show how busy is a restaurant at a given time
9. Booking sharing via 3rd party apps

The client has requested development of the MVP be conducted using the Scrum implementation of agile methodology. As such, iterations will be delivered based on the sprints determined by the team through the help of the supervisor and client.

4.1.4. DEPLOYABLE REST API

A deployable back-end server API using REST principles that will be used to create and modify resources within the mobile application. The REST API may be designed and developed as a collaborative effort with the project development team working on the TableMatter Admin and Super Admin Dashboards project.

4.2. ACTIVITIES

1. Project Scoping
    - Defining the Project Scope
    - Creation of Project Plan
    - Creation of SRS Document

2. Application Design
    - Creation of Visual Prototype (Application wireframe)
    - Writing User Stories
    - Defining software architecture
    - Establishing and set up tsproject development environments
    - Collaborative Application Design

3. Application Development
    - Development of native mobile applications
    - Development of back-end REST API
    - Development of application database
    - Collaborative Application Development

4. Application Testing
    - Application Unit Testing
    - Application flow testing on iOS and Android
    - User testing on iOS and Android
    - User Acceptance Testing on iOS and Android
    - Back-end REST API testing
    - Application extensions, fixes and changes

5. Final Delivery
    - Creation of technical and user guides
    - Handover of project deliverables

4.2.1. DOCUMENTATION

The SRS and Project Plan will be updated and revised as the project evolves through development and meetings among the team, with the supervisor, and with the client. This activity will be broken down into the following tasks:

- Update Documentation
  - The documentation will be updated to reflect any changes that arise due to changes proposed by the team, supervisor or client

- Submit Documentation for Approval
  - Major changes to documentation should be submitted to the supervisor for approval before submitting to the client to be signed off

- Submit Documentation to Client
  - Approved changes to the documentation will be sent to the client and either discussed for further changes or signed off by both the team and client

Additional documentation may be required and maintained as the project progresses and will be reflected within this document.

4.2.2. MEETINGS

Meetings will be held on a weekly basis and are as follows:

- Team Meetings
  - The team will meet amongst themselves to discuss the weekly progress and contribution of individual members. The team will also discuss the details of the client meeting to determine the tasks for the current or future sprint and if any changes are required for the project. This meeting is currently set to occur every Monday at 5pm.

- Supervisor Meetings
  - The team will meet with the supervisor to discuss the weekly progress and determine if there are any issues that need to be resolved. The supervisor meeting will help guide the team in an appropriate direction to fulfil the deliverables required of the project and client. This meeting is currently set to occur every Tuesday at 2:30pm.

- Client Meetings
  - The team will meet with the client to discuss the weekly progress and ensure expectations are being met. The client may offer feedback or changes to the project based on this progress or iterations submitted which can alter the current and future tasks for the team to complete. This meeting is currently set to occur every Sunday at 2:30pm.

The team will be routinely communicating via Slack to relay any important information or questions that occur during the week. Small discussions are also held after every meeting to examine meeting minutes and summaries.

4.2.3. SPRINT PLANNING

Each sprint planning session will define what is to be delivered during the sprint and how the work will be achieved. The sprints will be determined in conjunction with the supervisor and client. To successful plan for a sprint, the team must:

- Determine the objective of the sprint from the client
- Determine the length of the sprint with the client
  - This will help determine what tasks can be completed within this timeframe to achieve the objective
- Build a product backlog from the sprint objective and timebox
  - This backlog will be used to determine the development of the current or next iteration of the application / project
- Sprints will then be tested, re-adjusted as needed and marked as completed
- Other:
  - Sprints will be structured
  - Clear allocated team members to each task - clear responsibilities and deadlines
  - Regular updates to sprints

4.2.4. DEVELOPMENT

This is the main activity of the project and will be conducted via sprints. Each sprint will be determined by the details from sprint planning and will either achieve a new iteration of the application or progress upon an existing one. The tasks of each sprint may vary, however, they may be generalised into the following tasks:

- Feature Development
  - As listed in the Deliverables, there are many features that will be developed to form the mobile application. There may be more features or iterations of features developed as the project evolves and is further defined.
- UI/UX Design
  - The design for the mobile application, encompassing all the end user facing sections and features of the application. These designs will be created in Adobe XD and will be prototyped and iteratively designed to determine the final design for the application.
- Testing
  - The application will require testing to determine whether the functionality and design is working as intended and is appropriate for the project. Both the features and the design will require testing and this may occur through end-to-end testing, unit testing, user testing and so forth.

4.3. RESOURCES

The following software resources will be used to develop the project:

- React Native
  - React Native is the framework that will be used to develop the mobile application for TableMatter as per request of the client

- Node.js
  - Node.js will be used in conjunction with the Express framework in order to build the REST API endpoints.

- GitLab
  - GitLab will be used to manage the repository for the TableMatter REST API and Mobile application project. It also provides other services that may be used by the team such as issue tracking, continuous integration, continuous deployment and more. If such services are used, they will be listed and described.

- MongoDB
  - A MongoDB database will be required in order to achieve the MVP for the mobile application for TableMatter. This database will either be developed by the Admin and Super Admin Dashboard group or as a collaborative effort among both teams.

- Adobe XD
  - Adobe XD is the application that will be used to create the UI/UX designs of the mobile application as per request of the client.

- Amazon AWS EC2
  - A free-tier virtual machine will be hosted using Amazon AWS EC2. The purpose of the virtual machine is to host the REST API and database of the application for testing and development purposes.

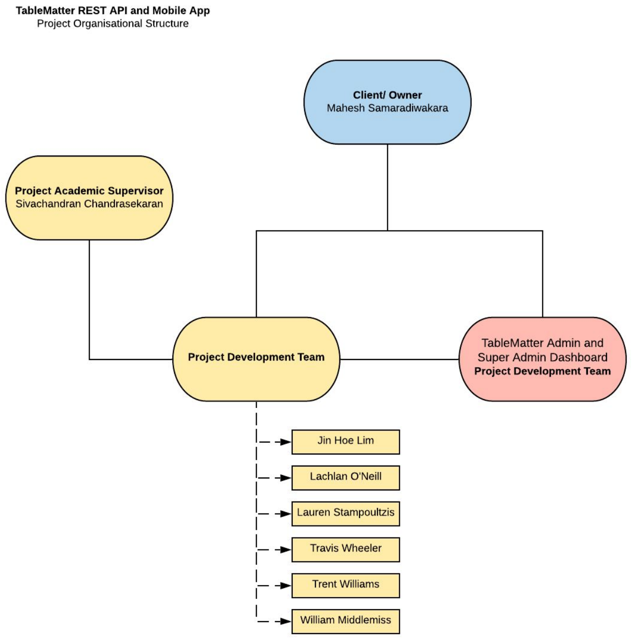
- Apple Developer Program subscription
  - An Apple Developer Program subscription will be required for testing the native mobile application on iOS devices. **The acquisition of said subscription is yet to be determined.**

Additional resources such as third party libraries and packages will be added as they are determined when development begins. The following hardware resources will be used to develop the project:

- Windows 10 and MacOS development computers.
  - These computers are expected to be private property of the team members. However, instances may arise where specific hardware is required for certain aspects of development.

- Android and iOS mobile test devices
  - Devices running Android and iOS will be required to test builds of the native applications. These devices are expected to be private property of the team members. However, instances may arise where specific hardware is required for certain aspects of development.

5. ORGANISATION AND STRUCTURE


TableMatter REST API and Mobile App
Project Organisational Structure
Client/ Owner
Mahesh Samaradiwakara
Project Academic Supervisor
Sivachandran Chandrasekaran
Project Development Team
TableMatter Admin and Super Admin Dashboard Project Development Team
Jin Hoe Lim
Lachlan O'Neill
Lauren Stampoultzis
Travis Wheeler
Trent Williams
William Middlemiss

The following group structures and roles are identified within this project.

1. Project Development Team
   - The project team consisting of Jin Hoe Lim, Lachlan O'Neill, Lauren Stampoultzis, Travis Wheeler, Trent Williams and William Middlemiss will be responsible for developing and delivering the project through all stages highlighted in this project plan. The Project Development Team will work closely with the client throughout the project development process, collaborate with the TableMatter Admin and Super Admin Dashboard's project development team, and report to the Project Academic Supervisor.

2. TableMatter Admin and Super Admin Dashboard Project Development Team
   - The project development team involved in the sister project will collaborate with the project development team of this project in certain areas. This project development team will share the same client, but will report to a different Project Academic Supervisor.

3. Client/ Owner
   - The client and owner of the project deliverables consist of Mahesh Samaradiwakara. The project development team will work closely with the client throughout the project development process to receive feedback on the project deliverables.

4. Project Academic Supervisor
   - Dr Sivachandran Chandrasekaran has been assigned the role of this project's academic supervisor. The Project Academic Supervisor will provide feedback and guidance to the Project Development team throughout the project development stages. The Project Academic Supervisor will hold authority over the Project Development Team over the duration of this project.

6. RISKS

Due to the context of TableMatter – REST API and mobile application as well as its primary goals, there are several key risks within the project, as well as there exist issues that all IT and web-based projects must account for.

Each of the risks currently facing this project can be subdivided into:

- **Market risks**
  1. One market risk that faces this project is the fact that user acceptance testing may be difficult to achieve, due to the nature of the project combined with the potential escalation of the COVID-19 response taken by governing bodies.
  2. Due to the nature of web technology, a minor risk of the project is that the technology used within the project may become obsolete by the completion or shortly after the completion of the project.

- **Security risks**
  3. As this project's scope includes the implementation of a payment system, one risk that will need to be considered is the impact of incorrect implementation leading to account information being accessible.
  4. Since the project will be capable of storing personal information about the user, incorrect implementation of the database and security pose a serious risk to the project's safety and success.

- **Financial risk**
  5. Due to the fact that this project is being completed with minimal to no budget, one risk that must be considered is how to proceed in the situation that certain tasks of the system are incompletable without paid solutions (from a package/ distribution level). This risk is currently very high due to the inclusion of payment system in the scope
  6. Due to the limited budget the project will be restricted to using free development tools.

- **Technological risks**
  7. One risk associated with the minimal budget is that most open source projects/ MIT licensed packages do not provide support with their usage. This means that certain issues within packages may be difficult to address in the situation that the project is dependent on a defective package.
  8. Due to the fact that the project is built upon open source packages, another risk that that should be considered is that the packages we use may be edited/ updated in a manor that may impact our projects ability to work
  9. Another risk associated with open source packages is that they may not be maintained or maintenance could be pulled from the package in the future.

- **Personnel/ Organisational risks**
  10. One major source of risk is the fact that the TableMatter product as a whole is subdivided across both this project as well as the **TableMatter - Admin and Super Admin Dashboards**, with some overlap in project tasks/scope requiring collaboration.
  11. Potential scope creep as the project progresses
  12. Potential miscommunication/ misunderstandings in the scope of the project as we progress into development.

- **Environmental/ Global risks**
  13. Another major risk that this project currently faces is the current, ongoing COVID-19 pandemic. COVID-19 has the potential to slow workflow causing setbacks in deliverables and poses a great threat to the ongoing health of members of both the project team and stakeholders.
  14. Due to restrictions on travel and meeting in person enacted to combat the COVID-19 pandemic, there are risks to the project's ability to communicate effectively with both stakeholders as well as with members of the project team.

Risks associated with this project (Risk Register).

Rank	Name / Description	Occurrence Probability (H/M/L/VL)	Severity (H/M/L/ VL)	Mitigation Strategy	Contingency
1	Risk of project members contracting COVID-19. - **RISK 13**	H	H	The project team shall abide by all government recommendations for reducing the spread of COVID-19 and reduce in person contact to minimal throughout the completion of this project	In the event a member of the project contracts COVID-19 each other healthy member shall be expected to split the individuals workload amongst each other for the affected period
2	Restricted communication as a result of COVID-19 - **RISK 14**	H	M	Multiple online platforms will be pursued for communication and document sharing including (google drive, slack and zoom) in order to supplement communication	In the case that these platforms go down or internet access is restricted. Team members are expected to maintain offline copies of documentation and work as well as should be able to communicate via phone/ text.

3	Inability to conduct some aspects of user testing due to restricted in person communication - **RISK 1**	H	M	In order to mitigate risks associated with minimal user testing, access to the repo/ alpha access to the development version of the application will be provided to the user for feedback in design.	In the case that the client is unable to access the repo or the development version of the app screenshots outlining the current ui design as well as functionality shall be used during meetings.
4	Miscommunication/ misunderstanding surrounding functional requirements - **RISK 12**	L	H	To mitigate the impact of misunderstanding functional requirements, the team shall discuss the understanding of the scope as well as get the client to sign off on the requirements prior to the beginning of development.	In the case that incorrect requirements get through to development or they are implemented incorrectly, weekly meetings with the client as well as their access to the development version will allow for the problems to be identified and given a priority for adjustment/ fixes based on their impact and the current schedule.
5	Flaws within implementation of payment gate - **RISK 3**	VL	H	In order to mitigate the risk associated with implementing the payment gate, a commercially accepted solution will be chosen and their outline of how to implement shall be followed	In the case that a security flaw is identified in the payment integration. The project will be deemed as not fit under any circumstance to be released (this should not be an issue due to the project being a PoC) until the issue is fixed, and within our flow a high importance issue shall be opened to fix it within either the current or next sprint.
6	Issues implementing features (payment gate) due to restricted budget. - **RISK 5**	M	M	In order to mitigate the risk of potentially having to use paid solutions, extensive research of our required stack and architecture shall be conducted prior to development in order to ensure that any potential paid requirements are brought to the customer early for either approval or Scope adjustment.	In the case that a paid requirement is identified during development an issue shall be opened to clarify its need within the team as well as with the client to ensure that the feature is of a high enough importance to the client to merit them spending money on it.

7	Miscommunication with other project team - **RISK 10**	M	M	Members from the other team shall be added to our communication frameworks such as our slack workspaces etc, so that we are not only able to communicate, but they are able to view our work. Further, they shall be provided view access to our private repo so that they can see what we have completed that they require.	In the case that communication between the two project teams breaks down, it is expected that the REST API written by our team is completed to our best knowledge of their needs as well as a wiki shall be produced in order to share how it works with the other team and so they may use it as they see fit.
8	Mismanagement of Scope creep - **RISK 11**	M	M	In order to ensure a clear understanding of the projects scope minimal changes shall occur to the corresponding sections of the document following the initial agreement of work	In the case that the client comes up with additional requirements/ scope that they want to add to the project, meetings will be conducted to both assess the requirements feasibility as well as the additional work it will require and our current ability to complete pre-existing requirements.
9	Project dependencies lose maintenance - **RISK 9**	L	M	In order to mitigate the chances of our dependencies losing maintenance only mature solutions with existing roadmaps will be utilised, such as React Native.	In the case that a project dependency becomes unmaintained during development. A meeting shall be held to determine whether the package is “complete enough” to remain within the code base or whether we must pursue alternative solutions. In the case that we must pursue an alternative solution the client shall be informed as well as the timeline and scope shall be reassessed.
10	Minimal support provided by open source/MIT packages - **RISK 7**	M	L	In order to mitigate potential issues involved with lack of support, packages with relatively large user bases will be pursued in order to ensure ample documentation and support exists.	In the case that a project dependency has issues that we are unable to resolve, alternative solutions shall be pursued. A meeting with the client will also be conducted in which the scope shall be reassessed.

11	Security issues caused by database/ encryption mis-implementation - **RISK 4**	VL	M	A standard approach to database security shall be utilised in order to ensure industry standards as well as commonly used encryption types shall be utilised to protect sensitive information	In the situation that there are security issues, an issue will be raised that must be addressed within either the current or next sprint as well as the project will be marked as not being fit for release project is technically not fit for release as it is a PoC).
12	Project dependency updates - **RISK 8**	L	L	To mitigate the risk of dependency updates affecting the projects workflow only mature dependencies shall be pursued with guidelines concerning the deprecation of code in order to ensure the project has ample time to make changes in the event code becomes deprecated.	In the situation that part of the project's code becomes deprecated a meeting will be created within the project team to assess the difficulty in updating our code. In the situation that this is impossible the client shall be informed of the potential delays and alternative solutions will be pursued.
13	Technology becoming obsolete - **RISK 2**	L	L	In order to ensure that the technology used within the project does not become obsolete within the next year only packages and external solutions that are mature and have a roadmap for maintenance spanning at least a year will be pursued for the project.	In the situation that the technology in the project becomes obsolete the project team will discuss the viability of pivoting to newer technology, however, due to the nature of this project “obsolete” will not mean much more as there being a newer way to complete some tasks the project tackles and will not have an impact on the overall maintainability of the codebase.

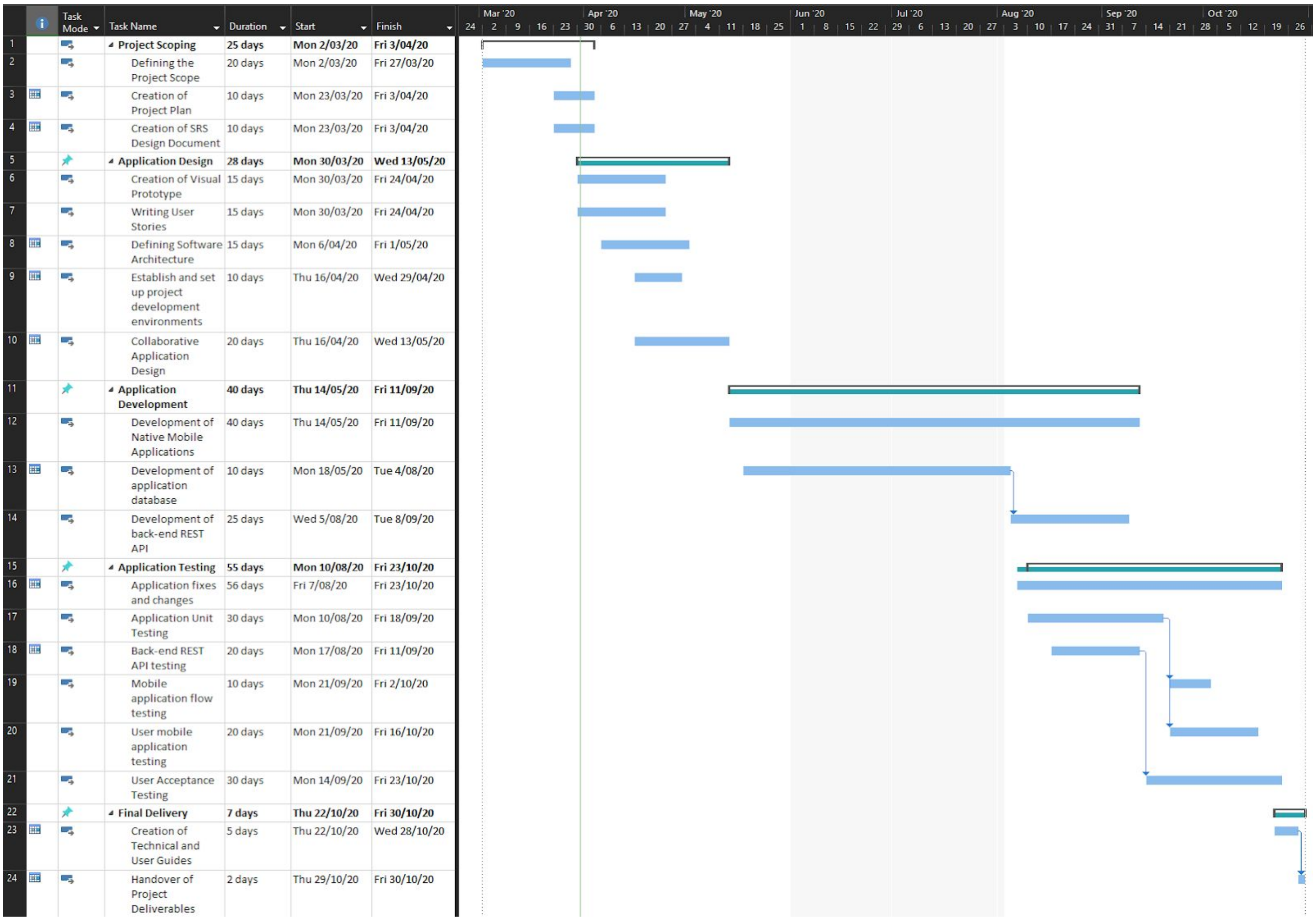
14	Limited to use of freeware - **RISK 6**	M	VL	Prior to the beginning of development research shall be conducted to ensure that all aspects of the projects development is completable with freeware. In the case that there are issues, the client shall be informed and either licenses for the software can be attained or adjustments to the scope can be made	In the situation that paid development software is not identified until development a meeting shall be arranged to assess the necessity of both the requirements the software addresses as well as the the software itself.

Table 2 Risks

		Impact			
		V. Low	**Low**	**Moderate**	**High**
Likelihood	**High**			• Restricted communication as a result of government response to COVID-19 • Inability to perform in person early user testing as a result of COVID-19	• Team members/ important stakeholders catching COVID-19
	Moderate	• Limited options in development tools due to freeware restriction.	• Issues addressing limits of MIT/ open source packages due to minimal support.	• Issues implementing certain aspects of the system (like payment portal) due to restricted budget. • Miscommunication/ mis-completion of tasks as a result of multiple projects working to create the TableMatter service. • Issues that may arise due to mismanagement of scope creep	•
	Low		• Technology becoming obsolete by the time of the project's completion • Issues caused by project dependencies being updated	• Issues concerning the fact that MIT/Open source projects may lose maintenance.	• Miscommunication/ misunderstanding of requirements implied by the scope.
	V. Low			• Issues to security caused by incorrect implementation of database and encryption.	• Issues to security due to incorrect implementation of payment gateway.

7. SCHEDULE

7.1. PROJECT TIMELINE AND GANTT CHART



	ℹ	Task Mode	Task Name	Duration	Start	Finish
1		➜	**◢ Project Scoping**	**25 days**	**Mon 2/03/20**	**Fri 3/04/20**
2		➜	Defining the Project Scope	20 days	Mon 2/03/20	Fri 27/03/20
3	▦	➜	Creation of Project Plan	10 days	Mon 23/03/20	Fri 3/04/20
4	▦	➜	Creation of SRS Design Document	10 days	Mon 23/03/20	Fri 3/04/20
5		📌	**◢ Application Design**	**28 days**	**Mon 30/03/20**	**Wed 13/05/20**
6		➜	Creation of Visual Prototype	15 days	Mon 30/03/20	Fri 24/04/20
7		➜	Writing User Stories	15 days	Mon 30/03/20	Fri 24/04/20
8	▦	➜	Defining Software Architecture	15 days	Mon 6/04/20	Fri 1/05/20
9	▦	➜	Establish and set up project development environments	10 days	Thu 16/04/20	Wed 29/04/20
10	▦	➜	Collaborative Application Design	20 days	Thu 16/04/20	Wed 13/05/20
11		📌	**◢ Application Development**	**40 days**	**Thu 14/05/20**	**Fri 11/09/20**
12		➜	Development of Native Mobile Applications	40 days	Thu 14/05/20	Fri 11/09/20
13	▦	➜	Development of application database	10 days	Mon 18/05/20	Tue 4/08/20
14		➜	Development of back-end REST API	25 days	Wed 5/08/20	Tue 8/09/20
15		📌	**◢ Application Testing**	**55 days**	**Mon 10/08/20**	**Fri 23/10/20**
16	▦	➜	Application fixes and changes	56 days	Fri 7/08/20	Fri 23/10/20
17		➜	Application Unit Testing	30 days	Mon 10/08/20	Fri 18/09/20
18	▦	➜	Back-end REST API testing	20 days	Mon 17/08/20	Fri 11/09/20
19		➜	Mobile application flow testing	10 days	Mon 21/09/20	Fri 2/10/20
20		➜	User mobile application testing	20 days	Mon 21/09/20	Fri 16/10/20
21		➜	User Acceptance Testing	30 days	Mon 14/09/20	Fri 23/10/20
22		📌	**◢ Final Delivery**	**7 days**	**Thu 22/10/20**	**Fri 30/10/20**
23	▦	➜	Creation of Technical and User Guides	5 days	Thu 22/10/20	Wed 28/10/20
24	▦	➜	Handover of Project Deliverables	2 days	Thu 29/10/20	Fri 30/10/20

Please see the external Microsoft Project file delivered with this document to view the project timeline in further detail

As the development stages are in-line with the team member's study load, please note the following non-working days as listed on the Swinburne Academic Calendar. All non-working days have been accounted for in the project timeline:

Semester 1 Mid-Semester/ Easter Break:	Thursday, 9 April - Wednesday, 15 April
ANZAC Day:	Saturday, 25 April
Semester Break/ Winter Break:	Sunday, 7 June - Monday, 3 August
Semester 2 Mid-Semester Break:	Sunday, 13 September - Sunday, 20 September

7.2. EXTERNAL DEPENDENCIES

1. **TableMatter Admin and Super Admin Dashboard Project Team**

   The development and completion of the project is dependent on the degree and effectiveness of collaboration with the TableMatter Admin and Super Admin Dashboard team.

7.3. ASSUMPTIONS

These are events that are expected to occur during the development of the project. Having all our assumptions listed is very important for the project as if they are not met it could potentially have a key project milestone missed or delay the project. All assumptions must be followed with any implications and actions if it is not met. Arriving at the schedule it is assumed that:

- Assumption: All team members have the required and relevant skills to complete their tasks
  - Implication: Delay in project/ code not functioning properly or not working
  - Action: Testing, Weekly meetings and communication tools (Slack) to explain why they are struggling or need help

- Assumption: COVID-19 will not interrupt the project's planned schedule
  - Implication: Unforeseen delays in project and work commitments
  - Action: Communication. Changing the project plan and timeline to cater for this

- Assumption: The team will have access to all the required resources/ software/ hardware
  - Implication: Production testing and development cannot continue. Milestones will not be met.
  - Action: Speak to Swinburne Supervisor/ Team Leader on alternative tools and measures that can take place.

8. BUDGET

As part of the Swinburne Software Engineering Project undergraduate unit, all projects are granted a budget of $400.00 for hardware purchases.

All team members will be developing the project as part of their course at Swinburne University of Technology. Hence, there is no payment or no labour costs associated with this project.

Equipment/Resource Budget

This table lists the identified equipment and resource costs of the project. Note that the table items and costs may be subject to change throughout the development stages of the project.

Name	Cost	Source of Funding
Apple Developer Program	$99 USD per seat per annum	*Unknown as of 31/03/2020*

Time Estimated to Complete Each Task

Activity	Task	Estimated hours needed (Shared)	Cumulative hours per activity
Project Scoping	*Defining the Project Scope*	*100*	100
	Creation of Project Plan	*50*	150
	Creation of SRS Document	*50*	**200**
Application Design	Creation of Visual Prototype	50	50
	Writing User Stories	*50*	*100*
	Defining Software Architecture	50	150
	Establish and set up project development environments	30	180
	Collaborative Application Design	40	**220**
Application Development	Development of Native Mobile Applications	100	100
	Development of application database	20	120
	Development of back-end REST API	100	220
	Collaborative development	100	**320**
Application Testing	Application Extensions, Fixes and Changes	200	200
	Application Unit Testing	80	280
	Mobile application flow testing	80	360
	User mobile application testing	20	380
	User acceptance testing	30	**410**
Final Delivery	Creation of Technical and User Guides	50	50
	Handover of Project Deliverables	6	**56**

9. REFERENCES

Drumond, C 2020, *What is Scrum?*, Atlassian, viewed 26 March 2020, <https://www.atlassian.com/agile/scrum>.

Project Management Assumptions 2018, March Limited, viewed 26 March 2020, <http://www.project-management-basics.com/project_management/project_management_015_Assumptions.shtml>.