FIT2101 Assignment 1

Project Inception

Tutorial Group: Thursday (11AM - 2PM)

Group No: Group 2

Group Name: <NAME NOT FOUND>

Prepared by

Name	Student ID
Anas Tarek Qumhiyeh	32985754
Lahiru Weliwitiya	31177654
Ong Zhen Ni	32842848
Tay Chean Hao	32620691
Tong Zhi Hao (Howard)	33547823
Yan Zimeng (Gina)	32973977

TABLE OF CONTENTS

Project vision

"Empowering Agile Excellence: Revolutionising Team Dynamics and Performance with our Cutting-Edge SCRUM Team Management System."

1. Introduction

An application that is tailored towards Agile Teams that utilise the SCRUM process model. This software will be designed to host multiple users for only one project. The feature to handle multiple projects simultaneously will be implemented in future implementations.

1.1. Team Organisation

The Team

1.1.1. List of team members

1. Anas Tarek Qumhiyeh (32985754)
2. Lahiru Weliwitiya (31177654)
3. Ong Zhen Ni (32842848)
4. Tay Chean Hao (32620691)
5. Tong Zhi Hao (33547823)
6. Yan Zimeng (32973977)

1.1.2. Team members' contact, roles and responsibilities

Member's Name	Contact	Roles	Responsibilities
Anas Tarek Qumhiyeh	011-12983246	Scrum Master	Oversees the whole project, delegate roles to team members; keeping everything updated
Tay Chean Hao	012-2899178	Product Owner/Technical Writer/Quality Assurance	First point of contact with the client, also documents and writes tests for code blocks
Lahiru Weliwitiya	016-8310794 / +94 70 3189806	UI/UX Designer / Backend Developer	Overseeing the development of the UI and the backend of the application.
Ong Zhen Ni	017-726 9528	Secretary, QA/Technical Writer, Database Handler	Responsible for meeting minutes, keep track with prototype and documentation
Tong Zhi Hao	011-33330895	Backend Developer	Creating and managing the behind-the-scenes functionality of the product.
Yan Zimeng	010-9167893	Backend Developer	Creating and managing the behind-the-scenes functionality of the product

The contact information for each team member, as well as his or her roles and responsibilities for the first sprint, are listed above.

The above section also provides the contact details for every team member along with their roles and duties during the initial sprint. According to the Scrum framework's recommendation, team members will take turns handling different roles and responsibilities during each sprint to ensure everyone comprehensively understands all aspects of the team's tasks.

1.1.3. Process Model Used

The process model implemented throughout the course of this project is the standard SCRUM model, where the team consists of a Product Owner (PO) and a Scrum Master in the team, although some slight modifications are made, where a SCRUM meeting will be held every 3-5 days (as opposed to having daily meetings) due to the commitments we have from other units in our semester (minor daily tasks can be communicated daily via direct messaging channels). Another detail about the process model is that it will consist of 3 sprints each lasting 2 weeks

1.2. Time and task tracking

1.2.1. Task Allocation

Tasks will be listed and distributed among team members every inception using a table as shown below. The table below shows how the team distributes tasks in this project plan.

Item	Task	Person(s) In Charge	Expected Completion date
1.1.3	Describe process model used and how it differs from Scrum	Anas	7/8/2022
1.2.2	Task and progress tracking	Zhen Ni	7/8/2022

1.2.3	Explain how the team will use github	Chean Hao	7/8/2022
1.2.4	Time and task tracking policies	Zhen Ni	7/8/2022
1.3	State and describe the Definition of Done agreed by the team	Chean Hao	8/8/2022
1.4	Vision statement of the project to be delivered	Chean Hao	13/8/2022
2.1	State and describe programming language and platform agreed to be used by the team	Whole Team	13/8/2022
2.2	Architectural decisions	Howard	13/8/2022
2.3	Framework	Gina	13/8/2022
3.1	Identify potential risks that may affect the team's progress and tabulate a risk register.	Lahiru	14/8/2022
3.2	Design wireframe for demo	Zhen Ni	12/8/2022
3.3	Review document quality	Chean Hao & Lahiru	15/8/2022

1.2.2. Tracking system

1.2.2.1. Teams method of keeping track of progress

The Secretary will make sure that the meetings will be held regularly and every meeting we will have a progress report/check, while also having our product backlog to help us identify what tasks need to be completed.

1.2.2.2. Teams method of managing backlogs

Product backlog will be managed under GitLab issue tracker, and will be updated every meeting if progress has been made.

1.2.2.3. Teams method of keeping track of time spent on project task

Monday.com will be used to help keep track of time spent on the project tasks.

1.2.3. Usage of git

How the team will use git: They'll be about 5 branches (including the main one).

They are:

1. Backend Dev Branch: The branch where backend dev work will occur.
2. UX Branch: The branch that holds the UI of the project.
3. Database Branch: The branch with the Database handling part of the project.
4. Integration Branch: The branch where integration will occur before being merged to the main branch.
5. The main branch will be reserved for the stable, integrated version.

1.2.4. Tracking policies

1.2.4.1. Expectations

1.2.4.1.1. Project Expectations

Overall expectations for the project should be well tempered to keep workload to a manageable amount.

1.2.4.1.2. Member Expectations

All members of the project should attend all the scheduled meetings throughout this semester, and if they can not attend then they should provide a proper reason why.

As well as doing enough work that by the end of the sprint the SCRUM master will go through all the commits pushed, and determine whether their work is deemed enough .

1.2.4.1.3.

Role	Expectations
Backend Developer	Handle the logic and structure of the project code
UX/UI Designer	Creates the GUI and makes it as user friendly to our client
Database Manager	Handles the database that the project will utilise
QA/Technical Writer	Develop test cases and provide documentation
Secretary	Makes meeting minutes and makes sure that team runs cordially, also they'll be the first point of contact for conflict resolution

1.2.4.1.4. Role Expectations

If any of these roles require assistance for their task, they can request help from members who have their role as their 2nd preference, or the SCRUM master.

1.2.4.2. Communication

1.2.4.2.1. Communication Medium

Medium of communication will mostly be via WhatsApp, although we will use Discord to schedule and commence our meetings, as well as to share documents with each other that are too big for WhatsApp to handle.

1.2.4.2.2. Communication Timelines

There will be a scheduled meeting every Saturday or Sunday based on preference of the team's schedule for their weekend.

1.2.4.2.3. Communication Code of Conduct

Respect among team members is of utmost importance, and any form of discrimination or harassment is not tolerated. Professionalism must always be upheld when discussing work matters.

1.2.4.3. Team Meetings

Scheduling of meetings will be done by the secretary, as well as jotting down the meeting minutes to help retain the information discussed throughout.

Involvement in the meetings will be quite simple, as each and every team member will have a partition of time in the meeting to talk and discuss about how their task is going and if they request any help.

1.2.4.4. Monitoring

As we'll be using GitLab for our version control, and Monday for project management. We'll be able to monitor the tasks via Monday, and monitor everyone's contributions via the commit log in GitLab.

1.3. Definition of done

After iteration is completed and has successfully gone through testing by the QAs, the Technical Writers will write documentation about the code written in the iteration. After one last testing to fully integrate it into the complete system, this would be our DoD.

2. Analysis of alternatives

2.1. Summary

This Analysis of Alternatives is a description of what this team has chosen for the following:

- Backend Programming Language
- Frontend Framework
- Database Management System (DBMS)
- Platform

And our choices for each were:

Decisions	Choices
Backend Programming Language	Python, JS and PHP
Frontend Framework (Presentation)	Bootstrap, Tailwind
Frontend Framework (Structure)	Vanilla JS, jQuery
Database Management System (DBMS)	Firebase Cloud, Oracle, MySQL, MongoDB
Platform	Web, Mobile App

We analysed each and every choice with a set of factors, and graded them from 0 - 5 (unless the factor is talking about a type -like SQL or NoSQL for example-). 0 being the least fulfilling of the factor and 5 being the most fulfilling of the factor. The factors at play were:

Backend Languages:

- Teams Expertise of the language
- Difficulty of learning the language
- Compatibility with Frontend
- Compatibility with Database Management Systems

Frontend Frameworks (Presentation):

- Difficulty of Implementation
- Ease of Integration

Frontend Frameworks (Structure):

- Difficulty of Implementation
- Difficulty of learning the language

Database Management System:

- SQL or NoSQL
- Ease of Use and Integration

Platform:

- Difficulty of building an app on the platform
- If needed to be maintained and/or updated

After comparing all our choices with the factors shown above. We decided to pick the following

Decisions	Choices
Backend Programming Language	JS
Frontend Framework (Presentation)	Bootstrap
Frontend Framework (Structure)	jQuery
Database Management System (DBMS)	Firebase Cloud
Platform	Web

We have chosen our backend programming language JS mainly because of its compatibility with the frontend. JS is naturally designed for programming a user-friendly interface compared to all other languages we are familiar with. Although we are more familiar with and satisfied with Python, it is not a good choice if we are considering making a program on the Web, which is why we have chosen JS as our final answer although it is not a very easy language to learn.

2.2. Terms of references

2.2.1. Backend Programming Language

The options we have as a team are Python, JavaScript, PHP. The factors we'll consider the following factors:

1) Teams expertise of the language
2) Difficulty of gaining Knowledge
3) Is compatible with Front-End Languages
4) Compatibility of the chosen framework with the Database Management System (DBMS)

2.2.2. Frontend Frameworks

The options we have are for 2 separate categories:

Presentation: Bootstrap, Tailwind

The factors we'll consider are:

1) Difficulty of implementation
2) Ease of Integration

Structure & Functionality: Vanilla JS, jQuery

The factors we'll consider are:

1) Difficulty of learning the language

2.2.3. Database Management System

The options we have are MongoDB, Firebase Cloud, MySQL and Oracle. The factors we'll consider are:

1) SQL or NoSQL
2) Ease of use and Integration

2.2.4. Platform

The type of platforms to build this application on are the following:

1) Mobile App
2) Web

The factors to consider are:

1) Difficulty of building an application with the platform
2) If needed to be maintained and/or updated

2.3. Body

2.3.1. Backend Programming Language

After considering the options, we graded each and all of the factors on a scale of 0 - 5. 0 being least fulfilling of the factor and 5 being the most no matter if the factor is positive or negative.

	Teams Expertise of the Language	**Difficulty of Learning the Language**	**Compatibility with Front-End Languages**	**Compatibility with Database Management System**
Python (Django/Flask)	3/5	Django = 3.5/5 Flask = 2.75/5	2/5	3/5
JavaScript	2.5/5	4/5	5/5	4/5
PHP (Laravel)	1/5	5/5	5/5	1.5/5

For Teams Expertise and Difficulty of Learning the Language, both of these gradings were based of the opinions of the team members who actually tried using said language

As for compatibility with Frontend, since Python is a derivation of C, so since python requires the help of frameworks like Django or Flask to actually be compatible with the backend, this is likely to make it slower (and harder to use), in comparison with Javascript or PHP in which we graded as 5/5 as it doesn't require any additional frameworks for support -although Laravel does make PHP more developer friendly.

As for DBMS Compatibility, Javascript and Python have quite an easy method to be able to connect to a DBMS (whether it's adding dependencies for Javascript, or creating a link object for Python). Yet PHP requires the use of external support either from APIs or Libraries.

2.3.2. Frontend Frameworks

Just like the section with Backend Languages, we'll rate each factor from 0 to 5.

	Difficulty of Implementation	**Ease of Integration**
Bootstrap	1/5	1/5
Tailwind	2/5	2/5

For both difficulty of implementation and ease of integration are mainly based on opinion. As it's easier to implement and integrate Bootstrap as implementation is just adding a class to an HTML element, whereas for Tailwind it'll generate an entire CSS File. For integration it still goes to Bootstrap as it requires the user to get its CDN and that's nearly all, whereas for Tailwind they'll have to install and configure it to their CSS file.

	Difficulty of learning the language
Vanilla JS	4/5
jQuery	3/5

Difficulty of learning the language is mainly opinion based as well, but the main reason is here's a code for a click listener on button, one is on vanilla JS and the other is jQuery.

jQuery:

```
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
        $('#button1').click(function() {
            alert("GeeksforGeeks");
        });
```

Vanilla JS:

```
        document.getElementById("button1")
            .addEventListener('click', function(){
                alert("GeeksforGeeks");
            });
```

Whilst Vanilla is easier to read, jQuery is more concise and is overall less clunky. And as more code gets written this will cause jQuery to be more readable compared to Vanilla JS. Obviously the team isn't forced to use jQuery and can use Vanilla if it's their preference.

2.3.3. Database Management System

Ranking the factors is similar to that of the last two, but for the factor of SQL or NoSQL will only have the possible answer of "SQL" or "NoSQL'.

	SQL or NoSQL	**Ease of use and integration**
MongoDB	NoSQL	3/5
Firebase Cloud	NoSQL	2/5
MySQL	SQL	2/5
Oracle	SQL	2/5

The factor of Ease of use and integration, is based on both opinion and fact. The opinion is from people who actually used DBMS services (i.e. our UI/UX/Backend Developer), but due to this team's very little experience with DBMS services, these factors are also graded from the opinion of those on the internet.

2.3.4. Platform

Factors will be graded from 0 to 5, whereas the factor for "If it needs to be maintained and/or updated" will only be a yes or no.

	Difficulty of building an app on the platform	If needed to be maintained and/or updated
Mobile App	4/5	Yes
Web	3/5	No

Difficulty of building the app is determined by opinion, especially the opinions of members who actually built mobile apps using Languages such as (Android Studios, Flutter, Swift, etc.)

2.4. Recommendations

2.4.1. Backend Programming Language

With all the factors considered, we decided to use Javascript due to its relatively easy setup and little need for external support for both frontend and DBMS. As for the team members that aren't well experienced in it, not only do we plan to give them less strenuous tasks. We also plan to provide them with courses and tutorials on how to use it

2.4.2. Frontend Frameworks

So for presentation we'll choose Bootstrap due to its simplicity and if any team members get confused, we'll help refer them to its documentation. For structure, we'll be using jQuery as opposed to Vanilla due to it being more concise and the scripts written usually are shorter compared to Vanilla JS (for example, the click listener function shown in the body). For team members who aren't well experienced or just don't have a preference for it, they are always free to use Vanilla JS as that won't greatly affect the efficiency of our product

2.4.3. Database Management System

And after considering the factors, we'll be using Firebase as our DBMS due to its very easy steps for configuration and use, as well as being one of the DBMS's that have actually been used by some of the members in this team. Firebase was also chosen as the preference of DBMS for our Database Manager (Howard/Zhi Hao)

2.4.4. Platform

We decided to use a Web Platform mainly because of everyone's preference against Mobile App Development as this team is very inexperienced with languages such as Android Studios, Flutter, etc. . Also because we heavily prefer the idea of not having to maintain or update the product after release

3. Risk Register

From what the team has observed through their own experiences, we have compiled a list of risks that have varying levels of impact and likelihood.

This list of risks will aid us in ensuring that things go about as planned, avoiding any potential delays and drops in software quality.

Attached together with this list is the monitoring strategy and the mitigation plan we came up with to deal with the risks, in terms of prevention and actually responding to it. ***For a more detailed version of this risk register, refer to the Appendix.***

Risk Description	Impact	Likelihood	Cause	Monitoring Strategy	Response	Risk Owner
Lack of team morale	High	Medium	Lack of professionalism and sufficient mental resilience	Scrum Master needs to keep morale up and take note of member's mental state	Insist that development can be finished if the team can just keep remain strong during these tough times	Anas (Scrum Master)
Lack of experience with SCRUM	Medium	High	Team never experienced SCRUM, or were used to other process models such as Waterfall or Spiral	Look up various SCRUM artifacts	Allocate time for the member to learn about the SCRUM artifact and adjust the sprint accordingly.	Anas (Scrum Master)
Health Issues	Medium	Medium	A team member responsible for one or more user stories in a particular sprint falls ill such that he/she requires rest for a time period that	Team members should try to keep their health at an acceptable level	Either distribute the user stories amongst the rest of the team members, or move them to a future sprint (if the user stories are of a lower	Anas (Scrum Master)

			lasts for more than one sprint.		priority)	
Team member is busy dealing with external circumstances	Medium	Medium	Team member lives in an unpredictable environment, which is susceptible to putting them into a situation that'll take their time	Communicate with team members frequently	Redistribute the workload among the team accordingly	Anas (Scrum Master)
Accidental branch deletion	High	Low	Carelessness when working with branches	Be careful when working on branches	Restoring the deleted branch using Git	Anas (Scrum Master)
Violation of integrity by pirating paid softwares	High	Low	Lack of funding and unwillingness to pay for premium softwares that make certain aspects of development easier	Constantly remind team members to not pirate software	Immediate deletion of the software and any products that were made with the help of it, and subsequent switching of	Anas (Scrum Master)
Scope Creep	Low	Medium	Uninformed changes made to the requirements by the client during a sprint.	Ensure that project development is set in-line with the project objectives	Recheck with client, to confirm whether the requirement is a valid one	Chean Hao (Product Owner)
Inadequate Client Involvement	Low	Medium	The client either has too little, or too many requirements, which the team isn't made aware of due to the client's lack of participation.	Ensure stakeholder remains engaged in the development	Attempt to organise further meetings/reshedule any prescheduled meetings with the client.	Chean Hao (Product Owner)
Poor Team Communication	Medium	Low	Lack of communication between team members responsible for user stories related to one feature of the software in a sprint can cause a	Continuously monitor team relations	Maintain consistent contact with teammates to ensure progress is maintained	Zhen Ni (Seceratary)

			disconnect.			
Inadequate Testing Time	Medium	Low	Allocating a lot of time to one particular task in a sprint so that the ssubsequent sprints are affected.	Keep track of task progress for each sprint	Work on test cases beforehand, and identify whether user story should be broken down if it takes longer than expected to complete	Chean Hao (Technical Writer)
Inadequate Testing Environment	Medium	Low	Limited availability of resources required for testing, such as devices with different specifications and screen sizes	Continuously monitor software environment	Use emulators to further test the software	Chean Hao (Technical Writer)
Security Concerns	Medium	Low	Inadequate testing increases the chances of vulnerabilities going undetected	Constantly produce test cases to cover for breaches	Create more testcases by considering all possible security threats	Lahiru (UI/UX/Backend Developer)
User Interface Bugs	Medium	Low	Failure to perform necessary amount of testing which allows for more bugs to go unnoticed	Constantly produce test cases to cover for UI bugs	Create more testcases catered towards ensuring User Interface works and looks as intended.	Lahiru (UI /UX/ Backend Developer)
API Instability	Medium	Low	Possible lack of rate throttling on the API side	Monitor API stabilities	Monitor further API changes	Howard (Backend Developer)
Unrealistic Expectations	Medium	Low	Taking on tasks that exceed current capacity	Discuss and ensure expectations are manageable at task assignment	Continuously manage team expectations and set realistic end goals	Anas (Scrum Master)
Goldplating	Medium	Low	Project team mistakenly views client's project as their own, thus using it to satisfy their own ambitions	Constantly look up on the features needed by the client during sprints	Confide with the client again, and determine whether additional feature is a desired feature. If not, remove it immediately.	Chean Hao (Product Owner)
Breach of Trust	Medium	Low	Team member	If team member	Immediately tell	Anas (Scrum

			doesn't want to tell the truth as it may result in getting the blame for a failure	was found to be lying, reassure them that nothing bad will happen if they tell the truth, and to ask them to tell the truth	the true state of the situation and discuss about potential responses	Master)
Tooling Issues	Low	Low	Pre-existing bugs on open-source tools/frameworks, that haven't been detected or fixed, yet, by contributors	Monitor the tooling versions, along with their patch notes to identify any existing bugs	Change the version of the tool that is used (to update it to the latest version)	Howard (Backend Developer)
Language Barriers	Low	Low	Lack of knowledge of the common language used by everyone in the team.	When a team member attempts to converse in a language other than the common language, remind them about it	Team members that aren't fluent in the team's decided common language can use other members that share a common language with them as a translator/interpreter	Anas (Scrum Master)
Unfamiliar Technologies	Low	Low	Team's unfamiliarity with using the specific technologies	Identify which technologies the team member is unfamiliar with, and learn about them.	Reevaluate the need of using the particular technology	Chean Hao (Technical Writer)

References

1. *How to connect Firebase database to Php scripts?* (2017, August 18). Stack Overflow. https://stackoverflow.com/questions/45367284/how-to-connect-firebase-database-to-php-scripts
2. Editor. (2019, October 15). Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others. *AltexSoft*. https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/
3. Add Firebase to your JavaScript project. (2023, August 20). *Firebase*. https://firebase.google.com/docs/web/setup
4. NamyaLG. (2022, January 6). Connecting Firebase to Python - TheLeanProgrammer - Medium. *Medium*. https://medium.com/theleanprogrammer/connecting-firebase-6102ef4eca08
5. *Installation - Tailwind CSS*. (2023, July 25). Tailwind CSS. https://tailwindcss.com/docs/installation
6. Murtaza, A. (2023). How to add Bootstrap to HTML (Step by Step Guide). *Creative Tim's Blog - Fully Coded Design Resources for Web Developers*. https://www.creative-tim.com/blog/web-design/add-bootstrap-html-guide/
7. Stevens, E. (2023, May 24). Mobile Apps vs Web Apps Compared: Which is Better? *CareerFoundry*. https://careerfoundry.com/en/blog/web-development/what-is-the-difference-between-a-mobile-app-and-a-web-app/#:~:text=Web%20apps%20need%20an%20active,Web%20apps%20will%20update%20themselves

Appendix

Full version of the risk register:

https://docs.google.com/spreadsheets/d/1H-CnUk84xfxIeaenEh-jJpL9cZ_ASxzU4pGUN4ChDd0/edit#gid=0

Appendix

Risk Register

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Likelihood	Impact	Status	Monitoring Strategy
R1	6	Lack of team morale	Lack of motivation to finish development due to reasons such as short timeframe and tight deadlines	Communication	Lack of professionalism and sufficient mental resilience	Factors such as overlapping assignmment deadlines closing in may lower team morale.	Insist that development can be finished if the team can just keep remain strong during these tough times	Anas (Scrum Master)	Medium	High	Closed	Scrum Master needs to keep morale up and take note of member's mental state
R2	6	Lack of experience with SCRUM	Due to teams lack of experience with the SCRUM process model, the project timeline may be slower compared to its ideal	Development & Communication	Team never experienced SCRUM, or were used to other process models such as Waterfall or Spiral	A team member encounters a SCRUM artifact that he/she did not know about, or wasn't briefed about.	Allocate time for the member to learn about the SCRUM artifact and adjust the sprint accordingly.	Anas (Scrum Master)	High	Medium	Closed	Look up various SCRUM artifacts
R3	4	Health Issues	One of the team members falls sick, or gets into an accident, leading to reduced productivity in the current sprint, and all subsequent sprints.	Health	A team member responsible for one or more user stories in a particular sprint falls ill such that he/she requires rest for a time period that lasts for more than one sprint.	* Serious illnesses such as viral infections * fatal accidents, such as vehicle crashes	Either distribute the user stories amongst the rest of the team members, or move them to a future sprint (if the user stories are of a lower priority)	Anas (Scrum Master)	Medium	Medium	Closed	Team members should try to keep their health at an acceptable level
R4	4	Team member is busy dealing with external circumstances	Due to external circumstances (familial issues, device gets corrupted and/or destroyed, etc.) Team member can't effectively do work in the mean time	Communication & Development	Team member lives in an unpredictable environment, which is susceptible to putting them into a situation that'll take their time	The unpredictable environment of one of the team members gives them a situation which requires their attention	Redistribute the workload among the team accordingly	Anas (Scrum Master)	Medium	Medium	Closed	Communicate with team members frequently
R5	3	Accidental branch deletion	Accidental deletion of branches that may cause setbacks for development	Security	Carelessness when working with branches	Pressing random buttons that lead to branch deletion	Restoring the deleted branch using Git	Anas (Scrum Master)	Low	High	Closed	Be careful when working on branches
R6	3	Violation of integrity by pirating paid softwares	Developers may choose to pirate software for a commercial project, failing to demonstrate integrity as according to the 'Engineers Australia Code of Ethics'	Tools	Lack of funding and unwillingness to pay for premium softwares that make certain aspects of development easier	Usage of illegally downloaded cracked versions of paid softwares	Immediate deletion of the software and any products that were made with the help of it, and subsequent switching of	Anas (Scrum Master)	Low	High	Closed	Constantly remind team members to not pirate software
R7	2	Scope Creep	Frequent changes in requirements can lead to scope expansion, affecting project timelines and resources	Scope	Uninformed changes made to the requirements by the client during a sprint.	Tasks that weren't part of the user stories at the beginning of the sprint is added	Recheck with client, to confirm whether the requirement is a valid one	Chean Hao (Product Owner)	Medium	Low	Closed	Ensure that project development is set in-line with the project objectives
R8	2	Inadequate Client Involvement	Lack of (or over) engagement of the client can lead to the misalignment of the overall project goals.	Communication	The client either has too little, or too many requirements, which the team isn't made aware of due to the client's lack of participation.	Client misses one or more meeting/s with the developers.	Attempt to organise further meetings/reshedule any prescheduled meetings with the client.	Chean Hao (Product Owner)	Medium	Low	Closed	Ensure stakeholder remains engaged in the development
R9	2	Poor Team Communication	Ineffective communication among team members may cause misunderstandings, inadvertedly leading to delays	Communication	Lack of communication between team members responsible for user stories related to one feature of the software in a sprint can cause a disconnect.	Establishing communication strategies that are ineffective during project inception.	Maintain consistent contact with teammates to ensure progress is maintained	Zhen Ni (Seceratary)	Low	Medium	Closed	Continuously monitor team relations
R10	2	Inadequate Testing Time	Inadequate time allocated for testing can lead to a product release with critical bugs	Time	Allocating a lot of time to one particular task in a sprint so that the ssubsequent sprints are affected.	A massive user story wasn't broken down to smaller user stories by the team, causing important features to be delayed in testing	Work on test cases beforehand, and identify whether user story should be broken down if it takes longer than expected to complete	Chean Hao (Technical Writer)	Low	Medium	Closed	Keep track of task progress for each sprint
R11	2	Inadequate Testing Environment	Limited testing environments can hamper comprehensive testing	Resources	Limited availability of resources required for testing, such as devices with different specifications and screen sizes	The requirement to test a particular feature cannot be fullfilled due to the lack of resources	Use emulators to further test the software	Chean Hao (Technical Writer)	Low	Medium	Closed	Continuously monitor software environment
R12	2	Security Concerns	Failing to address security vulnerabilities can lead to breaches (of client data), or other privacy violations.	Security	Inadequate testing increases the chances of vulnerabilities going undetected	QA of the software's seciurity was performed in a very lackadaiscal manner	Create more testcases by considering all possible security threats	Lahiru (UI/UX/Backend Developer)	Low	Medium	Closed	Constantly produce test cases to cover for breaches
R13	2	User Interface Bugs	User Interface (UI) bugs that weren't detected in the testing environment occurs, post production	Development	Failure to perform necessary amount of testing which allows for more bugs to go unnoticed	QA of the team tested the application UI and UX in a very lackadaiscal manner	Create more testcases catered towards ensuring User Interface works and looks as intended.	Lahiru (UI /UX/ Backend Developer)	Low	Medium	Closed	Constantly produce test cases to cover for UI bugs
R14	2	API Instability	Instabilities from using external APIs that change unexpectedly	Development	Possible lack of rate throttling on the API side	API downtime	Monitor further API changes	Howard (Backend Developer)	Low	Medium	Closed	Monitor API stabilities
R15	2	Unrealistic Expectations	Unrealistic expectations of team members, causing them to miss deadlines	Communication	Taking on tasks that exceed current capacity	Failure to manage expectations within the team	Continuously manage team expectations and set realistic end goals	Anas (Scrum Master)	Low	Medium	Closed	Discuss and ensure expectations are manageable at task assignment
R16	2	Goldplating	If client's requirements are very vague and the project team fills in additional details that the client never wanted. Thus delaying the project and wasting resources	Development	Project team mistakenly views client's project as their own, thus using it to satisfy their own ambitions	Addition of unwanted but seemingly useful features in the client's project	Confide with the client again, and determine whether additional feature is a desired feature. If not, remove it immediately.	Chean Hao (Product Owner)	Low	Medium	Closed	Constantly look up on the features needed by the client during sprints
R17	2	Breach of Trust	Developers break the trust with their stakeholders by either making false promises, or lying to other members of the team about their contributions and amount of progress. This will cause delays and break the chemistry of the team and stakeholders	Communication	Team member doesn't want to tell the truth as it may result in getting the blame for a failure	Lying about development progress to other team members, or lying to the client about software features	Immediately tell the true state of the situation and discuss about potential responses	Anas (Scrum Master)	Low	Medium	Closed	If team member was found to be lying, reassure them that nothing bad will happen if they tell the truth, and to ask them to tell the truth
R18	1	Tooling Issues	Inherent/unresolved bugs with different versons of development tools chosen.	Tools	Pre-existing bugs on open-source tools/frameworks, that haven't been detected or fixed, yet, by contributors	Team wasn't informed about potential bugs of the tool they plan to use, and proceed to use it in a way that can cause bugs with it	Change the version of the tool that is used (to update it to the latest version)	Howard (Backend Developer)	Low	Low	Closed	Monitor the tooling versions, along with their patch notes to identify any existing bugs
R19	1	Language Barriers	Cultural or language differences among team members can hinder communication.	Communication	Lack of knowledge of the common language used by everyone in the team.	Team member can't efficiently communicate with the language of the rest of the team's decided common language (For this team it's English)	Team members that aren't fluent in the team's decided common language can use other members that share a common language with them as a translator/interpreter	Anas (Scrum Master)	Low	Low	Closed	When a team member attempts to converse in a language other than the common language, remind them about it
R20	1	Unfamiliar Technologies	Using unfamiliar technologies could delay implementation	Tools	Team's unfamiliarity with using the specific technologies	Failure to learn the new technologies in time at a level required by the task	Reevaluate the need of using the particular technology	Chean Hao (Technical Writer)	Low	Low	Closed	Identify which technologies the team member is unfamiliar with, and learn about them.

The probability of a risk is taken as Low = 1, Medium = 2, High = 3

Impact ratings are taken as: Low = 1, Medium = 2, High = 3

The rank of a risk is taken as a scale from 1 - 10 **(highest to lowest)**

Risk Status = Not Achieved, In progress, Open, Closed

Risk Owners and their roles:
* Anas - Scrum Master
* Lahiru - UI/UX/Backend Developer
* Chean Hao - Product Owner (PO)
* Gina - Backend Dev
* Zhen Ni - Seceratary
* Howard - Database Manager