

# Final year project ICT 30001 2022

A Project plan

For

Internship management system

## Contents

Introduction .....	3
Project purpose.....	3
Project Description.....	3
Project Problem .....	3
Project Scope .....	3
Proposed Solution.....	4
Risk Management .....	5
KoST gap analysis .....	5
Work Estimation .....	5
Stakeholders .....	6
Risk Analysis .....	7
Human Resource Management .....	9
Organization structure .....	9
Roles and Responsibility .....	9
Functional requirements.....	10
Admin functionalities.....	10
Student and Industry Functionalities - .....	10
Non-Functional requirements.....	11
Performance and Quality.....	11
Security .....	12
Project Schedule .....	13
Gantt chart.....	13
Work breakdown structure.....	14
Resource breakdown structure .....	15
Communication management .....	16
Budget.....	19

## Introduction

### Project purpose

This is a final year project for the unit ICT30001 Information technology project, our team is assigned the responsibility of developing the Internship management system. Our responsibility of the project would be to design, document and develop the software for our client, which is Dr. Bibiana who is in charge of the internship program. Our supervisor is Mr. Sim Kwan Hua, who will guide us through the project.

### Project Description

Swinburne Sarawak is a university located in Kuching; they are a part of a larger branch in Melbourne. Recently they have created a new unit in their repertoire of courses that revolves around the students undertaking a 3-month internship placement. The staff in charge of running this unit, Dr Bibiana wants to create a system to help facilitate the course so that the process is easy on all parties involved.

### Project Problem

A large part of the Swinburne management occurs on their private databases and systems, including software like Canvas, which is a framework used by many universities around the world. Canvas is the main interaction between students and Swinburne teaching staff. The current active internship model is heavily reliant on manual interaction, where the staff have to engage with students and employers through emails. Managing this on a full scale will be extremely difficult as there's no way to measure and track every single student's internship placement. Swinburne cannot give access of their systems to the employers as it can be seen as a breach of security of the company and trying to find ways to facilitate them without interfering with sensitive information would be tedious.

### Project Scope

Dr. Bibiana has decided on wanting a platform where both the students and employers can interact with each other in real time. The software should allow both parties to sign up and log in, where the student fill their relevant information and that will be shown on a broad list filled with queries for the staff and employers to look through. The platform would also serve as the approval or rejection notification for both students and the staff. Aside from easing the management aspect of the internship program, the platform should be able to provide important analytical data through the usage of a queries or infographic data on a dashboard where the staff can use to improve the course for future semesters.

### Proposed Solution

Based off the initial client description we can conclude that the system design has a lot of very specific requirements which would require custom programming for the software. In general, the software should be a user management system, with admin tool functionality like CRUD (Create,Read,Update,Delete). Coding a desktop program with very detailed requirements with the current timeframe is a lot more difficult compared to a web application. For a website, the team can utilize various plugins and frameworks that are pre-designed for general user management. This would help with cutting down development time as we would only really need to focus on working on the exclusive features. Web applications are also more applicable and deployable compared to desktop applications because performance is less hardware dependent for web apps.

Therefore, our team proposes a web application software for client, similar to Swinburne's other external systems like "IMON". The website would have the functionalities of a regular user management portal, like a login page, user dashboard and registration. Extra features would then be added on top that layer based off the client's needs.

No	Features	Business value (1-10)	Sprint #
1	Query filters	8	Sprint #1
2	User database	9	Sprint #1
3	Report generation	9	Sprint #2
4	Approval and Rejection System	9	Sprint #2
5	Review System	9	Sprint #3
6	Infographic data	9	Sprint #3

The features above are what the team has hypothesized after consulting with the client. One of the critical aspects of the platform and the backbone would be the user database which consists of both student's information and employer's information. Query filters would then further compliment the database by allowing the users to sort through and look for what they want.

Within our solution our team would implement Agile development as our management style as it allows us to continuously adapt to the client's changing needs, by chopping large work loads into small modular tasks. This helps the team during the development phase and allows us to rearrange or reprioritize items in our sprints. Compared to a more traditional, abrupt changes would ultimately slow down the entire project. This approach is very crucial to our current project as the internship program is quite new and just has been rolled out recently and the requirements might change based of the first user data of manual semester.

## Risk Management

### KoST gap analysis

#### ***Knowledge***

As this project is for an entity that is beyond our knowledge, we would have to make a lot of estimations based off guesses. We believe our client does not have proper authority to share the technological structure of how Swinburne operates its software and database. This would prevent us from creating a database or software that would meld well with their pre-existing systems. We might also not confirm their staff's personal preferences so we would have to use existing usability guidelines for our designs to ensure operation of the software would be well adequate at the minimum.

#### ***Skills***

In order to produce a custom web application, the team would need to be experienced in web technologies like PHP, SQL, HTML and CSS. In addition, the team must be aware of frameworks like Laravel which used to make a lot of custom web applications with a functioning back end. If the team isn't fully adept in such skills, measures must be taken in place for example learning in spare time or substituting complicated coding with open-source plugins readily available in public repositories.

#### ***Technology***

An important factor to consider is that of if there are any alternatives readily available to the client for use. Off the shelf software are generally very reliable but at the cost of customizability of the tools it provides. For the client's case, there's many User management systems that focus on a web application framework, but most are very generic. One such instance is ManageEngine which is a very well-built web application made for management, however it lacks the specific custom requirements of the client. Through research our team concluded that there is currently no adequate software in the market. Therefore, solidifying our need to carry on with the project.

### Work Estimation

A major setback to most projects is the allocated timeframe and how much can be done within that time frame. Work efficiency is often overlooked and is never taken into account. In a normal environment a worker might have unforeseen distractions or problems in development.

Sometimes they might just out right slack with the work they are given. To avoid this kind of risk, the team has decided to apply estimation techniques to adequately assign work hours for each member.

Our current team of 3 members are currently expected to work at least 3 hours per day on the Project, totaling up to 15 hours per week. The expected weekly work hours from the team are 45 hours. With 12 weeks as the time frame, each individual member would work for 180 hours total, with a grand sum of 540 hours for the entire team. Now we need to assume that not all 540 hours are going to be used to their full extents or the effort put into those would be the same throughout. To get a more accurate estimation, we would need to use the triangular distribution method, also known as the 3-point estimation.

$$E \text{ (Estimated time)} = (O \text{ (Optimistic)} + M \text{ (Most Likely)} + P \text{ (Pessimistic)}) / 3$$

The team will assume that 540 hours is the most likely timeframe. In order to find the optimistic and pessimistic hours, the 70:40 rule will be applied where most teams are expected to work at 70% out of 100% this would serve as the pessimistic criteria, whereas 40% increase in efficiency would be the optimistic value.

$$E = (378 + 540 + 756) / 3$$

$$E = 558 \text{ Hours total}$$

Each team member should also be working at least 186 hours total. With this we can conclude that estimated time should account for any laziness or drop in efficiency, or just any other project hurdles that might interfere with development

### Stakeholders

No	Name	Position	Stake level
1	Dr. Bibiana Lim Chiu Yiong	Client	High
2	Sim Kwan Hua	Supervisor	Medium
3	Muhammad Arsalan Sohail	Project Manager	High
4	Ashfaq Ali Shagor	Project Member	High
5	Erik Sher Chuen Kiu	Project Member	High

It is essential to identify stakeholders early as you can gauge the stake levels and carry on the project adequately. For example, in depth conversations regarding the requirements of the platform would be better suited for Dr. Bibiana and Mr. Sim would be better suited for technical suggestions

## Risk Analysis

**Risk Impact information**

<b>Project Objective</b>	<b>Low</b>	<b>Medium</b>	<b>High</b>
<b>Cost</b>	Cost does not increase	5-10% increase in cost	More than 10% increase in cost
<b>Scope</b>	No notable change	Minor change in scope	Major change in scope
<b>Quality</b>	Barely reduced	Vital functionality intact	Severe reduction in quality
<b>Schedule</b>	Little to no delay	Slight delay	Major delay

**Technical**

<b>Risk event</b>	<b>Risk chance %</b>	<b>Risk Impact</b>	<b>Mitigation plan</b>
<b>Requirements too complicated</b>	Low	Medium	Postpone that feature for future release and focus on others
<b>Skill not adequate</b>	Low	Medium	Assign appropriate member's to their own strengths
<b>Slow Performance</b>	Low	High	Ensure optimized coding is implemented
<b>Confusing UI/UX</b>	Low	High	Implement UI design guidelines for better user experience

**External**

<b>Risk event</b>	<b>Risk chance %</b>	<b>Risk Impact</b>	<b>Mitigation plan</b>
<b>Change in requirements</b>	Medium	Medium	Ensure usage of Agile development to cope with change in requirements
<b>Client is busy for meetings</b>	Medium	Medium	Prepare future meetings in advance and prioritize to get the most out of each session

***Organizational***

<b>Risk event</b>	<b>Risk chance %</b>	<b>Risk Impact</b>	<b>Mitigation plan</b>
<b>Hardware failure</b>	Low	High	Make arrangements for back up hardware to be used by the team, for example the computer labs in university
<b>Internet outage</b>	Low	High	Ensure mobile data is ready as back up
<b>Lack of tools</b>	Low	Medium	Ensure that the team is well equipped with both good hardware and software

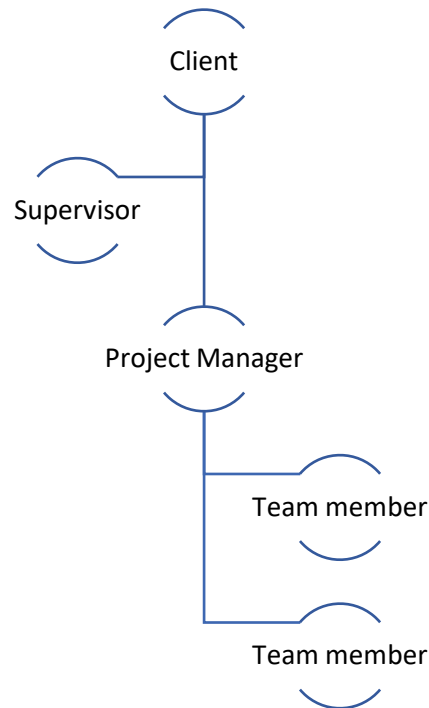
***Project management***

<b>Risk event</b>	<b>Risk chance %</b>	<b>Risk Impact</b>	<b>Mitigation plan</b>
<b>Team members absent for meetings</b>	Low	Medium	Prepare meeting minutes and send them to other communication channels so that members are caught up
<b>Team members are busy with other units</b>	High	High	Ensure all work is completed on time and workload is distributed evenly
<b>Team members are not communicating</b>	Low	Medium	Ensure that all team members are of good mental health and warmed up to each other
<b>Estimated hours are not enough</b>	Low	Medium	Ensure to use a concise estimation technique that take inefficiency into account
<b>Main channel of communication is down</b>	Low	Medium	Provide multiple means of communication



## Human Resource Management

### Organization structure



### Roles and Responsibility

Role	Responsibility
<b>Client</b>	<ul style="list-style-type: none"> <li>Define the project requirements</li> <li>Attend meetings</li> <li>Confirm progress on the project</li> <li>Approve final project</li> </ul>
<b>Supervisor</b>	<ul style="list-style-type: none"> <li>Ensure team completes the project</li> <li>Provide guidance to the team</li> </ul>
<b>Project Manager</b>	<ul style="list-style-type: none"> <li>Create a streamlined plan for the project</li> <li>Make sure team members are competent</li> <li>Make sure team members are punctual to meetings</li> <li>Make sure team members can complete the work given in time</li> </ul>
<b>Project Member</b>	<ul style="list-style-type: none"> <li>Accept the work that is to be given</li> <li>Assist in the development</li> <li>Assist in the project plan</li> <li>Design and document the system</li> <li>Attend every meeting</li> </ul>

## Functional requirements

After a few meetings with the client, the developers gather Functional requirements which are product features or functions that those developers must implement to enable users to accomplish their tasks. This list of functional requirements is not full scale and is mostly based on the estimations and data collected from the initial meetings with the client and hence this list would be updated

### Admin functionalities

- Admins must have a separate login system with a different domain for access
- infographic dashboard for admin in which it will display valuable information
- Database system by which the entire system will be created for data handling
- Queries which will generate results from the database
- A function that will generate reports
- Sign out option

### Student and Industry Functionalities -

- Sign up pages where they can register their information for login
- Log in system by which they will enter the system
- Search engine to view open students/industries
- Industry is getting a student list that will display all the students that applied for the industry
- Sorting feature that will allow the users to search with specific criteria
- A function by which the users can accept or reject offers/applications.
- A guideline page for students in which they will be instructed how to approve and reject offers
- CV upload feature for students in which they can upload their CV to the industries they applied to view
- Survey for students after they are done with the internship

## Non-Functional requirements

This is the list of all the non-functional requirements that describe the system's operation capabilities and constraints that enhance its functionality. This list is not the definitive version as more requirements might be needed after more meetings with the client.

### Performance and Quality

- Web page should load within 5 seconds
- Cannot exceed more than 5 mb page size
  - This includes logo image and other icons
  - The images and logos should have lossless compression and must have appropriate size
- Data request cannot have hang or time out means the data must undergo minimum operations

- Operating systems and their versions for users – Mac/windows

- Availability

Time required for a component to be fixed is 24 hours with a 75% chance

The system will be available 98% of the time when the system is online after enrolling of students start. The 2% unavailability risk isn't far too significant as after the application rejection and approval process end, users do not have many activities on the Application.

- The application's features and functionalities should support extensibility via NPM modules/packages.
- Network requirements for users-
  - Model-driven apps are designed to work best over networks that have the following elements:
  - Bandwidth greater than 50 KBps (400 kbps)
  - Latency under 150 ms
- Web application hosting specification

The following table lists the requirements for the Web Server Application, for any device capable of being a web server.

<b>Operating system</b>	Windows* Server, Linux*, or any operating system that can run as a webserver, capable of delivering HTML5 content, including JSON and MP4.
<b>Domain</b>	Swinburne.edu.my
<b>Processor</b>	Intel® Celeron® Processor 847, 1.10 GHz, or or higher.
<b>Storage</b>	Between 1.3 GB - 2.3 GB depending on the language version.
<b>RAM</b>	Minimum of 512 MB. The recommended amount can vary depending on number of users connected, number of websites hosted on the device, and other factors.
<b>Hard Disk</b>	3 GB of available hard-disk space for installation; additional free space is required during installation. You cannot install removable flash storage devices.

## Security

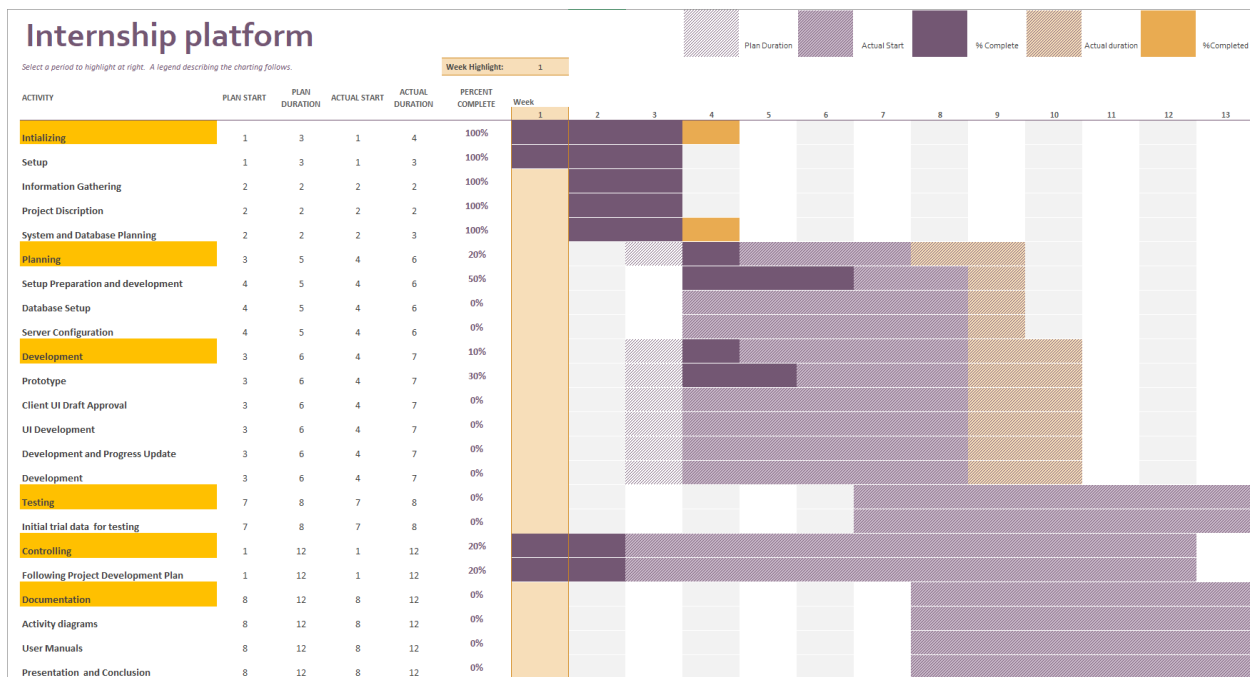
- The application's source code and logic must be backed up in a private online repository such as GitHub. So that the codes can be completely secured and can only be accessed by the servicing and developers.
- Use HTTPS protocol for code encryption. This protocol will ensure that the codes shown in website will be completely encrypted and will also provide internet security.
- Only admins can access the database via their log in. This means that, none of the other users will be able to view data and even manage that application information. This way the data will be completely secured and only be accessible for the administration.

- All passwords should be encrypted. During login, the password should not be visible and should appear in disguised format to prevent ghosting from anyone nearby.
- All input forms must contain input validation to prevent sql injections. Sql injections are sql queries placed inside the input fields in the application. The validations will prevent any other formats from being allowed for other php functions to process.
- Authentication of user login. This is to prevent any random user from accessing the system. It means that they will have to log in with a registered password and log in ID that is available in the system.

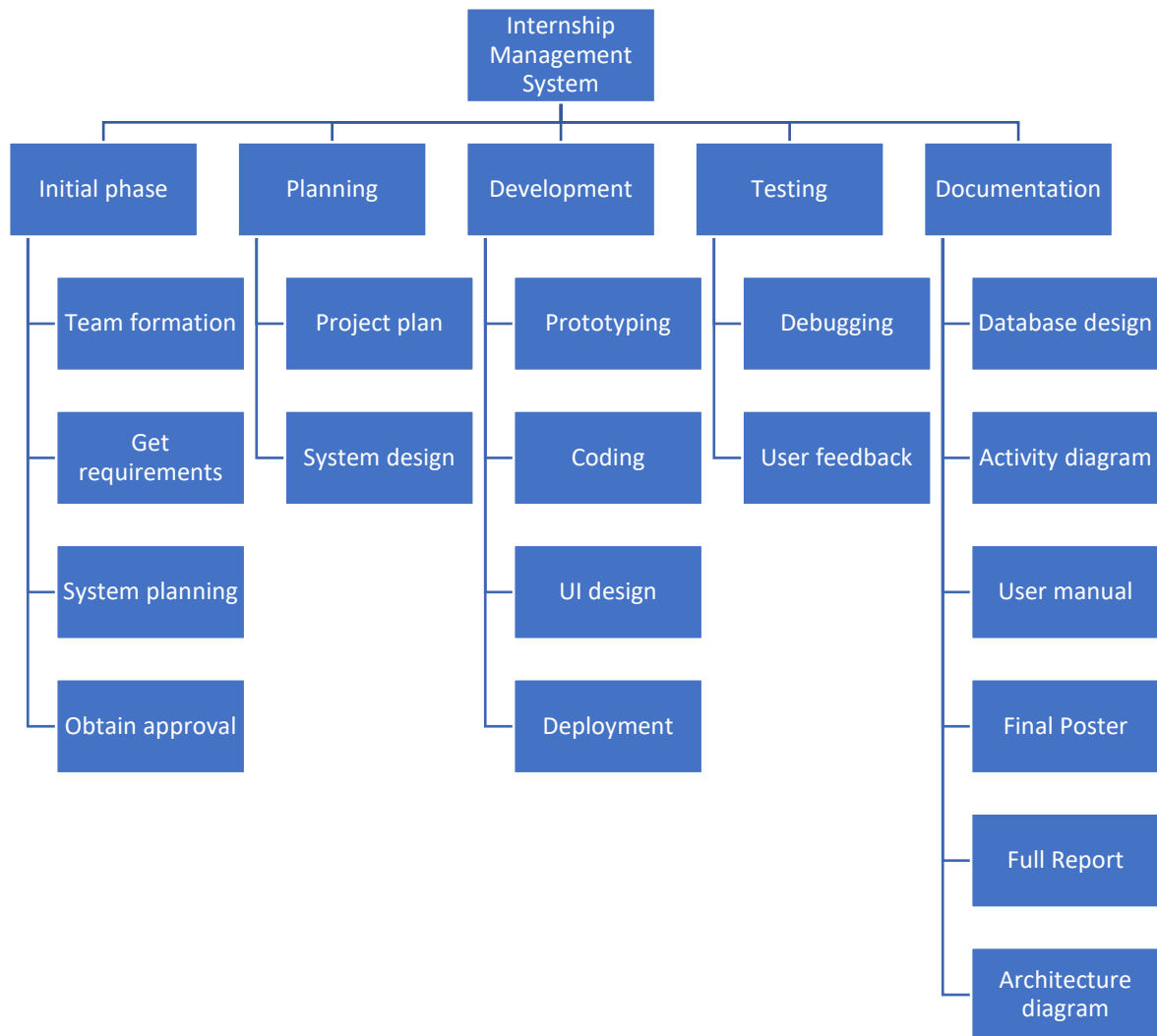
## Project Schedule

### Gantt chart

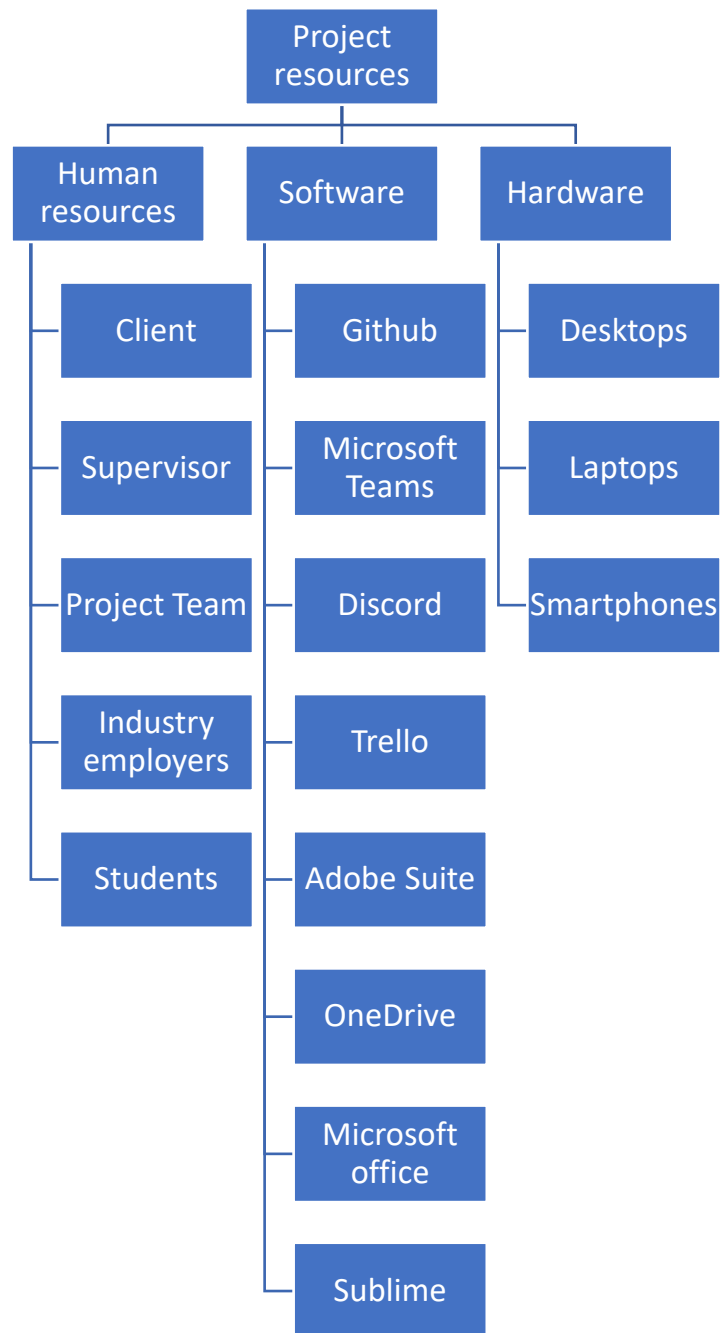
The Gantt chart allows our team to visualize our deadlines in order to meet them more effectively.



## Work breakdown structure



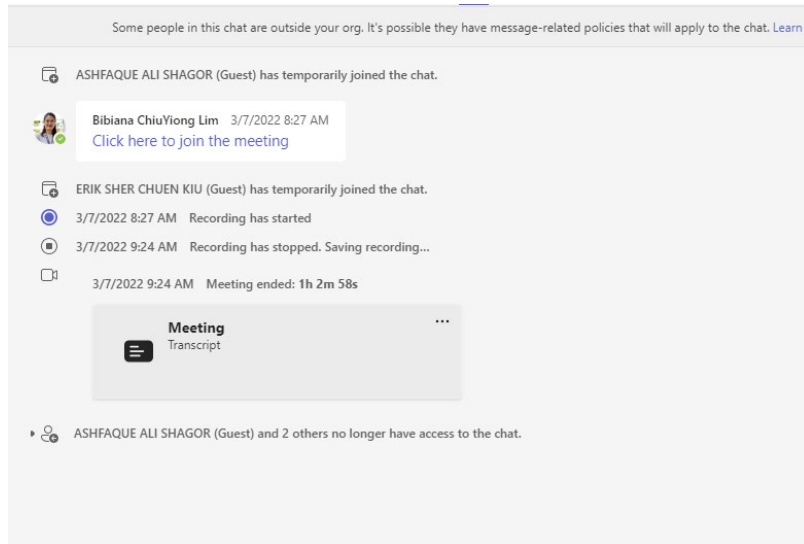
## Resource breakdown structure



## Communication management

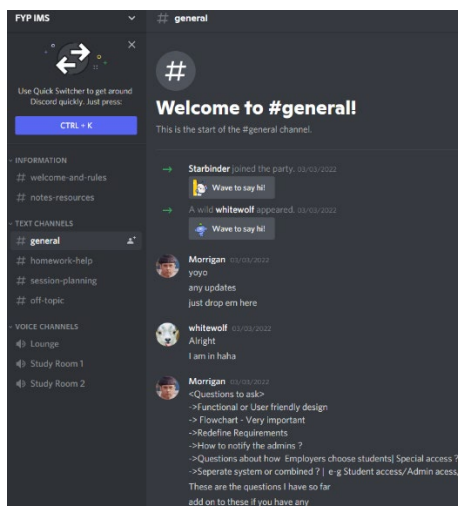
### Client meeting

Client meetings will be conducted via teams as it's proven that Microsoft teams are the most secure and available option for the clients to meet for product discussion. We will conduct client meetings at-max once a week.



### Developers meeting – Discord

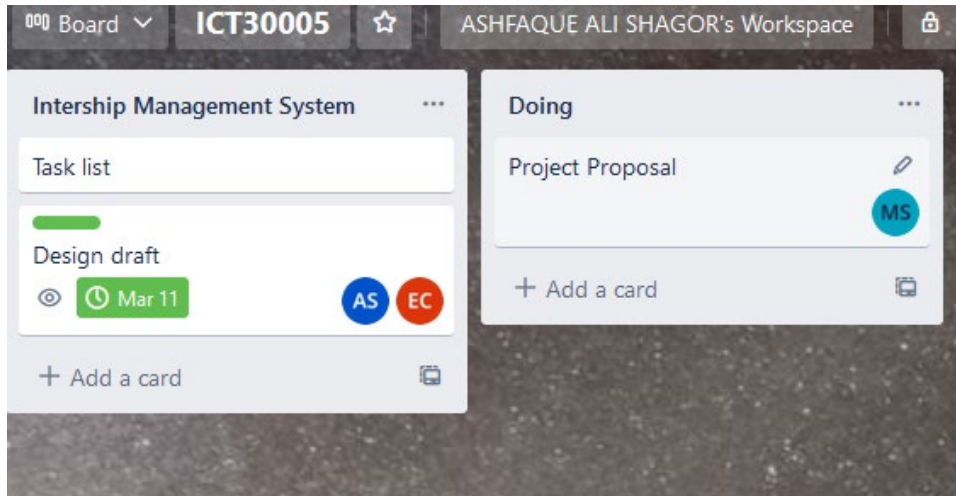
The reason the team chose discord for developers meeting is because the application provides wide range of functions and chat rooms in a single private server of our own. We can create multiple rooms with different names for specific purposes for the team to use such as file Sharing, documentation etc. Meeting will be conducted twice a week or any time the team calls a meeting .





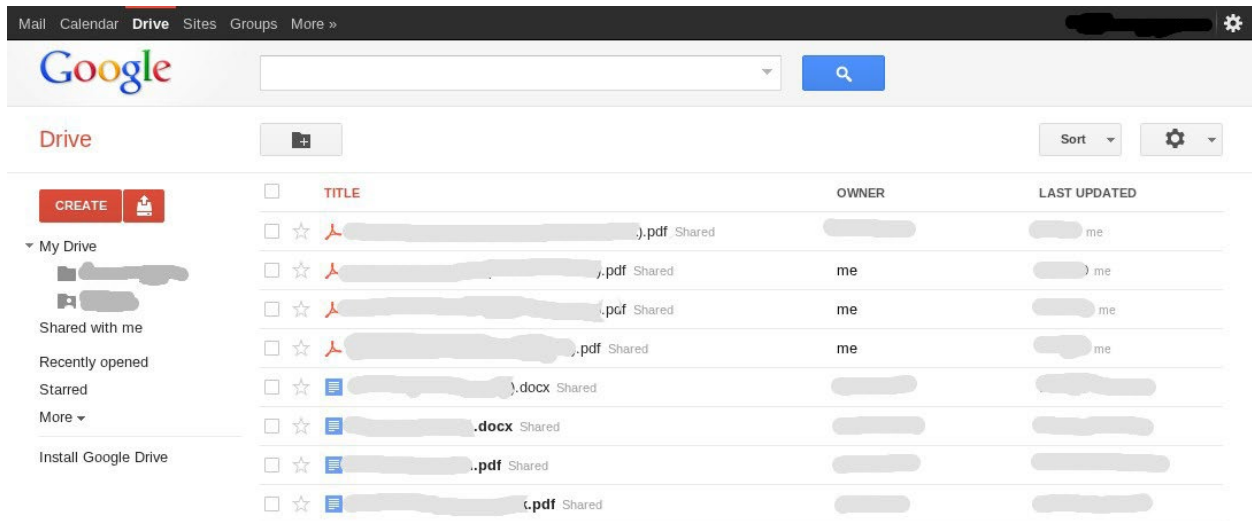
## Work/backlog tracking – Trello

Trello has been the best application to keep track of teamwork progress. It can specify how much time is needed to work and how much has been completed. This is updated every day during the project development.



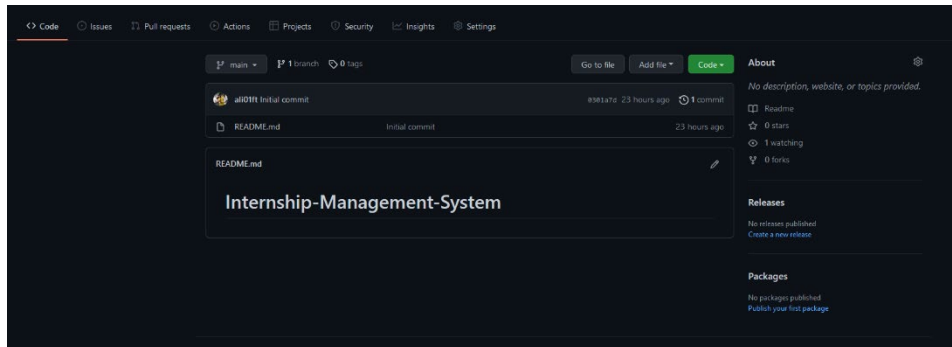
## Documents – OneDrive/google drive

The team chose OneDrive/google drive for its online applications such as word documents etc. for the team to access remotely from anywhere



## File sharing – GitHub

GitHub provides an online repository system in which the developers can access and share codes and place information on updates in a secure account.



## Project team directory

Name	Title	Email	Contact Number
<b>Dr. Bibiana Lim Chiu Yiong</b>	Client	blim@swinburne.edu.my	+60 82 260 704
<b>Sim Kwan Hua</b>	Supervisor	khsim@swinburne.edu.my	+60 82 260 694
<b>Muhammad Arsalan Sohail</b>	Project Manager	100074157@students.swinburne.edu.my	+60173363446
<b>Ashfaq Ali Shagor</b>	Team Member	101220618@students.swinburne.edu.my	+880 1726-593705
<b>Erik Sher Chuen Kiu</b>	Team Member	101232121@students.swinburne.edu.my	+6014-994 9739

## Budget

### Project Budget

3/19/2022

FYP

Expense	Category	Budget	Actual	Difference (\$)	Difference (%)
Word Press Hosting	Operating	\$ 5.00	\$ 2.95	\$ 2.05	41%
Github	Operating	\$ -	\$ -	\$ -	-
Discord	Operating	\$ -	\$ -	\$ -	-
Ms Team	Operating	\$ -	\$ -	\$ -	-
Trello	Operating	\$ -	\$ -	\$ -	-
Excel	Operating	\$ -	\$ -	\$ -	-
Hardware	Operating	\$ -	\$ -	\$ -	-
Microsoft Office	Operating	\$ -	\$ -	\$ -	-
Adobe	Operating	\$ -	\$ -	\$ -	-
Total Expenses		\$ 5.00	\$ 2.95	\$ 2.05	41.00%

Overall, most of the software and equipment have not cost anything, besides for hosting.