Project Management Plan

Daniel Yuen Hao Xian

Chen Wei Siang

Wong Chung Chieng

Yap Han Tiong Jason

David Levy Christianson

Project Vision

For teachers who strive for a fairer marking system towards individuals in group projects, our marking system is able to statistically measure and evaluate the productivity and contribution of each individual in a group project. Unlike most marking systems, our system is feasible of use and can be accessed via the internet.

Team Contact Info

Name	Contact info
Daniel Yuen Hao Xian	60103928029 dyue0003@student.monash.edu
Chen Wei Siang	60192855561 wche0065@student.monash.edu
Wong Chung Chieng	60109715900 cwon0040@student.monash.edu
Yap Han Tiong Jason	601110231068 hyap0009@student.monash.edu
David Levy Christianson	6281233450786 dlev0002@student.monash.edu

Roles and Responsibilities

Role	Names	Responsibilities
Scrum Master		Ensure the team follows the agile processes and practices that the team agreed they would use Clearing obstacles that would impede the team Ensuring a good relationship between the team and product owner Protecting the team from outside interruptions and distractions.

Developers	Chen Wei Siang Daniel Yuen Hao Xian David Levy Christianson Yap Han Tiong Jason	Researching, designing, implementing and managing software programs. Develop technical and functional specifications for projects. Plan, coordinate and execute project activities to ensure timely completion. Assist in determining time and cost estimates for assigned projects. Resolve project development issues in a timely manner. Manage project risks, and milestones. Ensure project deliverables meet the definition of done in the allocated time.
Product Owner	Wong Chung Chieng	Confirming and clarifying clients' request Reporting on team's progress Negotiating with client Responsible for deciding the content of the product backlog and priority of each backlog entry on the basis of business value/Return On Investment (ROI)

Team Process Model

As the members have other units to attend to and cannot commit to this project entirely, the typical Scrum model does not fit well with our schedules. However, we will follow the Scrum methodology as closely as possible to achieve the core value in this model.

All the Scrum ceremonies will proceed as in a typical Scrum but the intensity and the duration has been altered. We divide the meetings into two days; where we conduct standup, retrospective and sprint planning on Mondays and a second standup on Fridays.

A single sprint duration is set to a week. Therefore the sprint planning will also be carried out once per week. The planning will be carried out after the retrospective. The items in the product backlog will be considered and moved to the sprint backlog as the next sprint's items. Members will also be assigned to various tasks based on the ability of the member and familiarity towards the subject. This is done exactly as in a Scrum model.

For the daily standups, it is deemed infeasible for us to meet up every single day, so we decided to have standups twice a week, which is on Mondays and Fridays. This is planned as we are more likely to make significant progress over the weekends. The standup will be held by one of the members and is strictly limited to 15 minutes per standup. In a standup, a member is required to update on the new progress between the meetings. Any technical problems will be tackled and resolved after the standups. This approach is different from a Scrum because of the lower intensity.

Sprint review will be conducted weekly on each Friday before or after the lab class to inspect the most recently completed sprint. Code reviews will be done and members will discuss the codebase and make sure everything is integrated smoothly. Product demonstrations will mostly be conducted in the lab class, as that is the opportunity to meet up with the Client. Our approach is similar to a typical sprint review where it is carried out near the end of a sprint cycle.

Retrospective will be done on the Monday meeting or after the lab class if possible. That way we can discuss the inputs given by the tutor and improvise on the next sprint planning which will be carried out right after the Retrospective.

Definition of Done

Checklist:
1. Meet all client's requirements
2. Pass test harness
   - Unit test are done on each function
   - Acceptance criteria passes test cases
   - Integration testing performed
3. Non functional requirements: Has maintainability, Flexible to changes
4. No active bugs present
5. Documentation on the codebase is complete
   - Code is well commented
   - Any assisting documentation needed is complete

Scrum Ceremonies

1. Sprint Planning
   a. Sprint: 1 week duration
   b. Conducted every Monday on Discord
   c. Assess product backlog and decide on items to be moved to sprint backlog for the current sprint

2. Daily Scrum
   a. Conducted every Monday and Friday on Discord
   b. 15 minute discussion about progress of each team member towards their responsibilities

3. Sprint Review
   a. Conducted at the end of every sprint to discuss about the most recently completed sprint
   b. Review completed items in sprint backlog and organise if necessary
   c. Conduct product demo with Product Owner when needed

4. Retrospective
   a. Conducted before the next sprint is started (Monday)
   b. Provide suggestions to improve sprint planning based on experience
   c. Discuss strategies to improve our productivity in following sprints

Task Allocation

Team members are allocated tasks based on their skills and familiarity with each technical aspect required. Heavier tasks are given to more experienced team members while easier tasks of the same technical aspect are given to less experienced team members so that all members get to work on all aspects of the project and to prevent siloing. The Scrum Master and Product Owner will be given a lesser share of work.

Progress Tracking

The team will be using Trello to track the completion progress of each sprint. Tasks can be easily shown in the dashboard to show the progress so far of the current sprint with its in-built functionality such as label and checklists. It is also easier to keep track of which developers are working on which tasks and their progress.

Managing and Storing Backlog

The team will be using Trello to manage and store the backlogs of each sprint.

Tracking Time Spent

The team will be using GitLab to track the time spent on project tasks. Since the date and number of commits can be tracked, the time spent can also be calculated to prevent overspending on a trivial task and allowing the project to be completed on time.

Risk register

ID	Date Raised	Risk Description	Probability level	Impact level	Mitigation action	Monitoring
1	24/8	Estimation and scheduling risk -Software development are hard to estimate and schedule, especially with a group of inexperienced developers	High	High -Failing to meet the deadline	-Get the team more involved in planning and estimating. -Get early feedback from clients.	-Monitor each developer's progress and get feedback from them → To evaluate if more time and hands are needed
2	24/8	Requirements Inflation	Low	High -May cause the whole project to be rebuild from the ground	Constant involvement of client and developers->Inquire the requirements to be as clear as possible from the client	-Have frequent meeting between client and developers → Ensure that the requirements of the client are clear and the development is on track
3	24/8	Poor Task Management/Allocation	Medium	High -Failing to meet the deadline -May cause the whole project to be rebuild from the ground	-Make full use of the scrum structure and communicate well with each other on the task completed and in progress.	-Monitor each developer's progress and get feedback from them → To evaluate if more time and hands are needed -If developers are met with a difficult task, discuss it during sprint meetings
4	24/8	Technical	Low	High	-Make an analysis of	-Monitor each

		difficulties		-May need to change the software/ programming language and rebuild it from the ground	potential software to be develop and choose the most feasible software that can meet all the client's requirements	developer's progress and get feedback from them → If the progress seems slow, prompt the developer feedback and change software while still in early stages of development, if in later stages of development, advice developers to give up the minimum amount of requirements while be able to achieved most of the requirements set

Deliverable 2: Analysis of Alternatives

2.1 Programming Language

1. Programming Language Used: Javascript, Layout: HTML & CSS

2. Analysis Foundation & Criteria:

This project involves building a web-based application. To follow enlisted requirements, the language that will be used should fulfill this criteria of selection:

- The language must be supported by our targeted platform (describe the platform here pls)
- The language must have sufficient resources & libraries capable to support web development
- The language used should be efficient to use in it's time to be produced and executed
- Most of the team should have at least basic knowledge or experience of the language
- The language should have some tools / IDE that can be used by the team
- The language should not be costly to use
- Language that have certain benefit of use that is useful to our project should be prioritized

3. Analysis of Alternatives:
   1. Python
      a. Pros
         i. Python support integration and can be used in web development
         ii. Python have huge community and libraries including web development
         iii. Python have an acceptable time of execution
         iv. Most of the team have basic to extensive knowledge and experience with the language
         v. Python have many IDEs
         vi. Python is among the easiest and shortest to code language

      b. Cons
         i. Python is not native to web environment (cannot be run directly)
         ii. Python is not good option for memory intensive task
         iii. Python can't build high graphic 3D objects

   2. Java
      a. Pros

   i. Java support integration and can be used in web development

   ii. Java have huge community and libraries including web development

   iii. Most of the team have basic to intermediate knowledge and experience with the language

   iv. Java have many IDEs

   v. Java is among the most stable language

  b. Cons

   i. Huge java project can be slow to compile and execute

   ii. Java is considered complex language

   iii. Java have low performance in web programming

3. Javascript

  a. Pros

   i. Javascript integrate well with web development and with other core www language like html and css

   ii. Javascript can be developed rapidly and is executed fast in web-based platform

   iii. Most of the team have basic knowledge with the language

   iv. Javascript have many IDEs

   v. Javascript is purposely developed for web development, it has unique benefits such as Client-side execution and UI Interactivity

   vi. Javascript works well across multiple browsers and platforms

  b. Cons

   i. Javascript code is always visible

4. PHP

  a. Pros

   i. PHP is a server-scripting language and can be used in web development easily

   ii. Some of the team have basic knowledge of the language

   iii. PHP have many open-source IDEs

   iv. PHP is compatible cross browser platform

  b. Cons

   i. PHP is slow in performance

   ii. PHP is not adjusted for desktop application

4. Recommendation

By considering the criteria of selection, each possible choice of our programming language has their own pros and cons. However, javascript is the only language that enables ease of use in implementing UI interactivity. Enabling easy transfer of

information between the user, UI and the program itself will greatly help in enhancing the functionality, usability, and features of our project. To enable ease of use in implementing many functionality intended for this project, javascript will be used in trade of code visibility.

2.2 Platform
1. Platform targeted: desktop and mobile(Android and iOS)

2. Analysis Foundation & Criteria:
Project is to be built to be run on both PC and mobile platforms. The platform should follow the requirements:
- Able to be run on both desktop and mobile
- Accessible on any browser
- Accessible on any OS

3. Analysis of Alternatives:
1. Mobile (native app)
    a. Pros
        i. Able to produce cleaner, smoother UI using a native app
        ii. Smaller target devices would reduce compatibility issues between devices of similar OS
        iii. Does not require a browser to run
        iv. More accessibility to device functionality
        v. Does not need access to the Internet to be able to run
        vi. Uses Java which is familiar to all team members
    b. Cons
        i. Compatibility issues between different OS (Android and iOS)
        ii. Required to download the app to run
        iii. No cross-platform support (desktop and mobile)

2. Desktop and Mobile (web app)
    a. Pros
        i. Cross-platform support (desktop and mobile)
        ii. Only require a browser and an active Internet connection to run
        iii. Compatible with all browsers and OS
    b. Cons
        i. Require an internet connection to access
        ii. Slower and less efficient UI
        iii. Less device accessibility
        iv. Uses Javascript which is unfamiliar to most team members

4. Recommendation

Using a mobile native app would provide much more functionality given a better UI and access to device functionality but will limit the app's accessibility on different platforms. Using a web app will allow the app to be run from any device but reduce the functionality of the app. Since cross-platform support is an important functional requirement of the project, the project will be built as a web app with a trade-off of performance efficiency.

2.3 Communication tools

1. Communication tools: Zoom, Discord

2. Analysis Foundation & Criteria:

To ensure an efficient way of communication, communications tools used should follow the following criteria of selection:
- Have good capability to support live video-based communication
- Have good capability to support text-based communication
- Have good capability to support screen share feature
- Have good capability to support sharing and storing of links and small files
- Does not cost the team financial resources to use
- Fast and easy to use

3. Analysis of Alternatives:
1. Zoom

   (https://zoom.us/)

   Pros:
   i. Proper webcam interaction
   ii. Chat section for every session
   iii. Easy-to-use screen sharing function
   iv. Free to use

   Cons:
   i. Cannot support file sharing other than posting links
   ii. Requires entering ID to join every single time
   iii. Chat and anything sent is not saved for every meeting

2. Discord

   https://discord.com/

   Pros:
   i. Has webcam function
   ii. Has share screen feature
   iii. Support both text and video communication

- Chats are saved and can be reviewed back
  - iv. Supports long term file storage
  - v. Has open source bots and management tools to increase efficiency
  - vi. Some members uses it frequently and are well familiar with it
  - vii. Free to use

  Cons:
  - i. Unstable on Linux
  - ii. Some luxury functions are locked behind paywall
    - e.g. higher quality voice calls and hd screen sharing

3. Microsoft Teams
   (https://www.microsoft.com/en-au/microsoft-365/microsoft-teams/group-chat-software)
   Pros:
   - i. Multiple channel facilitates text-based discussion
   - ii. Easy to use, cloud based with fast processing power
   - iii. Chats are saved and can be reviewed back
   - iv. Can share and store files*
   - v. Free to use

   Cons:
   - i. *Microsoft teams have complex storing system for uploaded files
   - ii. Limited capability in facilitating webcam meeting
   - iii. Limited flexibility in using channels and managing access permissions

4. Slack:
   (https://slack.com/)
   Pros:
   - i. Multiple channels facilitates text-based discussion
   - ii. Have webcam feature
   - iii. Support screen-sharing
   - iv. Data is protected with high security
   - v. Usable in desktop and mobile applications
   - vi. Free to use

   Cons:
   - i. Conversations are deleted after 14 days
   - ii. Minimum file storage capacity

4. Recommendation

Microsoft Teams and Slack provide a more formal discussion and communication platform over Zoom and Discord. However, they are not chosen due to the limited control over the rooms and the overall unfamiliarity of those programs. Discord is

ultimately chosen over Zoom for it is easy to plan and initiate a meeting with chat recorded in the text channels.