**PROJECT PLAN**

**1.1 TEAM ORGANIZATION:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEAM MEMBERS** | **CONTACT NUMBER** | **EMAIL ID(@student.monash.edu)** | **ROLES** |
| Vibhas Kamal | 0451 621 080 | vkam0002 | Product Owner, Developer |
| Mike(Quang Nghiep Ly) | 0420 819 099 | qlyy0001 | Developer |
| Jack Beard | 0400 926 760 | jbea0002 | Developer |
| Isara(Tito) Arunanondchai | 0481 944 555 | iaru1 | Developer |
| Manvendra Singh | 0450 798 864 | msin0017 | Developer |

**1.1.1 Roles and Responsibilities:**

* Product owner:
  + Availability: 3 hours on Monday/Tuesday (beginning of each sprint)
  + Tasks: finalizes each sprint's product backlog before the Sprint meeting on Monday/Tuesday
  + The main responsibility of the product owner is to represent the interests of the client and act as the middleman between the team and the client
  + If any there is any confusion regarding the requirements and the team members are not able to resolve it, the product owner would be contacted and would be responsible to resolve the confusion.
* Team:
  + Availability for contact from 6pm - 11pm.
  + Submit code regularly (ie. should commit parts of tasks or tasks instead of finishing all allocated tasks then submit everything in one go)
  + Submitted work needs to meet the definition of done
  + Need to support one another in the case of technical uncertainty

**1.1.2 Team’s Process Model:**

* Weekly meeting held thrice (Monday and Thursday)
  + Purpose:

These meetings will be the Scrum meetings. It will include discussion of to-do work, assigning tasks, and possible re-allocation of tasks to team members (in case certain tasks turn out to be much easier or harder than expected).

* The sprint duration will alternate between between 3 days (Monday to Thursday) and 4 days (Thursday to Monday).
* Checking progress of assigned task checked in the last meeting.
  + Upon completion of every task assigned in one sprint, update product backlog.
* Sprint retrospective after every sprint.
* Formal consultation with client each Friday and contact through email (if needed) during the week.
* If the team members are facing any problems regarding the requirements, the team would be consulted. If the issue is still not resolved, the Product Owner would be contacted immediately. In case the Product Owner does not have the relevant information, the client will be contacted and the work on that requirement will be halted, and the developer working on that task would be asked to move on to the next task.
* Most of the communication within the team would be done via Facebook Messenger. If there is anything that needs to be clarified or conveyed immediately, the team members would use the contact numbers of the team members to contact the intended person.

**1.1.3 How the team’s process model differs from Scrum:**

* Group members do not have a daily meeting (Daily Scrum). Meetings are held twice a week.
* In addition to process review (sprint retrospectives), we will also run a technical review on the code produced within each sprint. This will help us to plan our next sprint.
* In Scrum, the team’s sprints are supposed be of the same duration. But due to the availability of team members in our team, our sprints would alternate between 3 days (Monday to Thursday) and 4 days (Thursday to Monday).
* Rotation of Scrum Master role through each Sprint

In the conventional Scrum process model, the person undertaking the responsibility of the Scrum master does not change. However, in order to allow all the team members to learn the Scrum process model properly, every team member would given the chance to be the Scrum master

**1.2 TIME AND TASK TRACKING:**

**1.2.1 Allocation of tasks to team members:**

* During the team meetings (scrum meetings), everyone will give their preferences regarding what task they want to do. After a discussion on the task allocation, each team member will be provided with tasks on the basis of the team member's abilities, interests, knowledge in that area.
* If there a task exists such that no team member wants to do, the team would meet together and try to figure it out together during one of the team meetings or will allocate a separate time which suits every team member’s availability..
* These tasks will then be formally allocated on Trello to the team members with an expected completion time.

**1.2.2 Tracking of time spent on project tasks:**

* The tasks will have an allocated time decided in the scrum meetings which will be mentioned on Trello.
* Google Spreadsheet will be used to keep track of the hours each member spent on doing the tasks related to this project (based on the assumption that no one would falsify information). This would be shared with the marker at the end of the project or the marker would be given access to the spreadsheet if required.
* If in case a team member is not able to complete the task in the allocated time, an extension would be given for that task based on what the other team members think about how much more time that task should be given.

**1.2.3 .Tracking of progress on your project:**

* We will use a web application, Trello, which will allow us to track personal and team progress on individual and group tasks.
  + The marker will be given access to Trello if required.
* In addition to the formal use of Trello, we will employ the casual strategy of communication through social media for instant delivery and retrieval of milestones related to the progress of our project.
* The sprint backlog is another artifact that can be used to track the amount of work done and remaining.
* Burndown charts may be drawn for a more general view of the total sprint and can raise alerts when the chart is not reaching 0 fast enough.

**1.2.4 Storing and managing backlogs**

* Vibhas (product owner) - Product Backlog/Sprint Backlog
* Remaining members of the team - Sprint Backlog
* The product and sprint backlog can be kept on separate folders inside a big “Backlog” folder on Google Drive (eg. sprint 1 stores product and sprint backlog of sprint 1, sprint 2 stores product and sprint backlog of sprint 2). This is done in the case we need previous Sprint’s data for progress monitoring, efforts/Sprint.
* Sprint backlogs are updated daily so team members know the current state of the sprint.
* The main reasoning behind the need for varying editions of the sprint backlog lies with the focus on agile development, whereby if the requirements of the project are susceptible to change, then the overall list of requirements can have tasks that the team has previously agreed on removed, or added with a newly refined definition.
* With the lack of flexibility in sprint backlogs, it proves incredibly important that these changes are made apparent before the sprint has begun.

**1.3 DEFINITION OF DONE/COMPLETE (CODE/DOCUMENTATION):**

For each of the features that are being implemented, the following points describe what ‘done’ means:

**1.3.1 Written code**

* Functional correctness
* Functions that take too long to run (>60s) are not acceptable[G5]
* Proper syntax and semantics used.
* Correct indentation
* Utilizes encapsulation (functions, classes) instead of copying and pasting of code in different sections
* Meaningful names (variables, constants, functions, classes, modules) which are self-descriptive
* Conventions to use:
  + Classes: underscore between words
  + Functions: lower camel case
  + Constants: all capital

**1.3.2 Documenting the code**

* Documenting of the code (which will include comments and function documentation) should be able to clearly explain what the code does)
* Documentation should only occur in areas of the code that exhibit non-intuitive behaviour. This is to keep the code easy to read.
* Parameters for functions (what they are and their type)
* Return value and type of functions
* There should be file headers describing what a file is doing.
* There should be comments before each block of codes and/or anywhere where the other team members might face problems
* Sufficient amount of documenting - 1 or 2 lines of explanation for each chunk of code
* Spell checked
* Documentation for every line is not needed

**1.3.3 Tested the code**

* Separate folder to store test cases - submitted alongside the functional code
* Test cases should cover a wide range of values
* Proper error and exception handling - Use of test cases, assertions, try/except clauses

**1.3.4 Submitted the code for review before merging**

* The code should be finalised >= 2 days before the deadline[G6] [G7] [G8] [G9]

**1.3.5 For other documents of any kind**

* Written the document
* Correctly formatted
* Spell checked
* Submitted for review by other team members (on an individual level) OR review of the document completed by all the team members (on a team level)

**1.4 PROJECT VISION:**

To deliver a high-quality software in terms of the documentation, functionality, ease of use and reusability, while fulfilling all the requirements specified by the client.