FIT3077
SPRINT 1: PROJECT INCEPTION

Done By: The Dev Dynasty
Shoumil Guha (32700660)
Rachit Bhatia (32638396)
Tan Jun Yu (32025130)

INDEX

1. Team Information

Document the following pieces of information related to your team.

■ Come up with your own <mark>personal team name</mark>. Your team name must be professional. The name of the team you belong to in Moodle (with the format <campus>_<workshop session>_<team number>) is not an acceptable team name for this task.

■ Your <mark>team photo</mark> must not be edited/photoshopped. All team members in an on-campus group must be present together physically at the time of taking the photo. For online groups, a Zoom team photo is accepted. Team photos via Zoom will not be accepted for on-campus groups.

Team Membership

■ Document the basic information of each team member - for example <mark>name and contact details.</mark>

■ List out what the <mark>technical and professional strengths</mark> of each member are.

■ Provide a <mark>fun fact</mark> about each member that not many people know about.

<mark>Team Schedule</mark>

■ Document your <mark>team's regular meeting schedule and regular work schedule</mark>.

■ Document how th<mark>e workload will be distributed and managed within your team.</mark>

Technology Stack and Justification

■ Document what programming languages, APIs, and technologies are you planning to use and how this maps to the team's current expertise, and which ones you anticipate needing support from your tutors with.

■ Justify your team's final choice of technologies that will be used.

Technology Stack: Java as programming language + JavaSwing or JavaFx for graphics

Justification: All team members are familiar with Java as an object oriented programming language

Team Name : The Dev Dynasty
Team Photo :

Team Members Contact Details

1. Shoumil Guha

   - **Phone**: 011-5640-9250
   - **Discord**: shoumil#3182
   - **Email**: sguh0003@student.monash.edu

---

2. Rachit Bhatia

   - **Phone**: 019-405-7032
   - **Discord**: racrage#2757
   - **Email**: rbha0031@student.monash.edu

---

3. Tan Jun Yu

   - **Phone**: 016-278-3792
   - **Discord**: junyu#7430
   - **Email**: jtan0245@student.monash.edu

Team Members Info

1. Shoumil Guha

   - **Technical and Professional Strengths :**
     - ➢ Well versed in object oriented programming and its design using Python and Java.
     - ➢ Experience in database management systems, web development, image processing.
     - ➢ Fast learner and starts work early which leads to no panicking close to deadlines.
     - ➢ Proficient in the agile working pattern due to past experience in Scrum projects

   - **Fun Fact :** I eat breakfast an hour before lunch just because I can.

2. Tan Jun Yu

    ○ **Technical and Professional Strengths :**
       ➢ A well-developed understanding on Object Oriented Programming principles which is vital for this assignment .
       ➢ Good collaboration skills by being able to work effectively with team members and actively participate in any group discussions for the best project outcome
       ➢ Familiar with the programming language we have chosen for this assignment that is Java being applied with Object Oriented Programming
       ➢ Experienced in the field of agile projects

    ○ **Fun Fact :** I dislike playing 9 Men's Morris and will not ever want to play or see this game again after this assignment

3. Rachit Bhatia

    ○ **Technical and Professional Strengths :**
       ➢ Very familiar with Object-Oriented Programming principles due to experience from prior projects.
       ➢ Equipped with some knowledge in UI development for iOS mobile applications
       ➢ Well-experienced with Java principles and semantics
       ➢ Proficient in team management
       ➢ A good team player with the ability to provide valuable input to every discussion while respecting everyone's opinions
       ➢ Well adapted to an agile team format

    ○ **Fun Fact :** I love to listen to music in all moods except when feeling overwhelmed or stressed

Team Schedule

❖ <u>Meeting Schedule</u>
   ➢ At the beginning of each sprint, the team will hold a sprint planning meeting to discuss and agree on the sprint goal, scope, and tasks to be completed. Task division for the sprint will be done in this meeting.

   ➢ A weekly standup meeting will be held to check on the team's progress to ensure that everything is on the right pace to lessen the chance of missing deadlines. Each team member will update one another on the progress of one's assigned tasks and make sure no one is left behind. This meeting will usually be on every Friday at 4pm. The timing may change subject to the team's convenience but the meeting cannot be cancelled.

   ➢ The frequency of these weekly standup meetings will change from once a week to 4 times a week when closer to the end of the sprint. When this change takes place will depend on the remaining amount of work for that particular sprint (frequency of meetings could increase from 7 to 10 days before the end of the sprint).

   ➢ After the end of each sprint (preferably the same or next day after a sprint ends), an additional sprint retrospective meeting will be held to discuss the shortcomings of the completed sprint and possible improvements for future sprints.

- ❖ <u>Work Schedule</u>
  - ➢ Every member is expected to spend at least 1-2 hours per day on the allocated work. This may increase based on the workload for the sprint.
  - ➢ The tasks given to each team member will be assigned an internal deadline considering the availability of the member and the difficulty of the task.
  - ➢ In every standup meeting conducted, the work done by each member will be discussed and reviewed. At the same time the team will look at any issues faced by a member and try to resolve them to smoothen the workflow.
  - ➢ It will be aimed to allocate roughly the same amount of workload to each sprint to ensure consistency in the workflow.

- ❖ <u>Workload Division and Management</u>
  - ➢ Every type of activity (writing user stories, coding, prototyping) is distributed equally
  - ➢ Tasks allocated to each member will cover all aspects of the project to prevent siloing.
  - ➢ Tasks will be created at the beginning of the sprint and will be managed through a Trello board. Each task's priority will also be recorded. All of this will be done during the sprint planning meeting.
  - ➢ Each member will update the status of task completion on the Trello board.
  - ➢ Team members should voluntarily choose tasks from different aspects to make sure everyone is comfortable with their workload. The final decision should be agreed upon by the other members.
  - ➢ Any task that might be harder to implement can be divided into smaller tasks and assigned to more than one team member.
  - ➢ Any severe difficulty encountered by a team member on the given tasks can be further discussed in a standup meeting and a decision whether to reassign the particular task to another member can be made.

Technology Stack and Justification

Definition and Purpose:

Chosen Programming Language: Java
  Pros:
  1. Strong structured language
  2. Industry standard for Object Oriented Programming
  3. Widespread support online as it has been around for decades
  4. Platform independent
  Cons:
  1. Slow
  Justification for Selection:

Discarded Alternative Programming Language: Python
  Pros:
  1. More unstructured with less rules to follow when programming
  2. Overwhelming support online
  Cons:

2. User Stories

Submit a list of user stories (e.g., 10 to 25 stories) that covers both the basic 9MM gameplay and the chosen advanced requirements specified above. A majority of the user stories are expected to be devoted to the basic requirements for the Basic prototype.
If your group consists of 4 members, your user stories must also cover the additional advanced requirement.

3. Basic Architecture

Design and draw a domain model that covers both the basic 9MM gameplay and the chosen advanced requirements specified above.
Provide detailed justifications for the domain model that you come up with, with a focus on the following aspects:
- Rationale for each chosen domain and their relationships (if any)
- Were there any design choices that you had to make while modelling the domain and WHY?
- Explain any assumptions you have made, as well as any other part of your domain model that you feel warrants a justification as to WHY you have modelled it that way.
If your group consists of 4 members, your domain model and justifications must also cover the additional advanced requirement.

4. Basic UI Design

Draw low-fidelity (low-fi) prototype drawings of the proposed user interface for the application. The low-fi prototypes need to demonstrate both the basic 9MM gameplay and the chosen advanced requirements specified above. The prototypes should cover all the key interaction scenarios, e.g. initial board, placing tokens, moving tokens, 'flying', forming a mill (win condition), and the advanced feature of your choice. This can be achieved in one large drawing space or across multiple pages. Avoid redundancy, i.e. do not create multiple prototypes for the same interaction. All drawings should be large and clear enough to understand and any writing should be legible. You may use pen and paper, or digital drawing tools.

If your group consists of 4 members, your lo-fi prototype drawings must also cover the additional advanced requirements.

2) ANALYSIS OF ALTERNATIVES

1.   Definition & Purpose

2.   Analysis for Programming Languages

2.1. Plan
{Insert Plan}

2.2.   Choice of Alternatives

2.2.1.   Alternative 1

	Backend	Frontend
Platform	Flask	Bootstrap
Programming Language	Python	JQuery, JavaScript, HTML, CSS

2.2.2.   Alternative 2

	Backend	Frontend
Platform	Node.js	PHP, Node.js
Programming Language	JavaScript	C, C++, JavaScript

2.3. Pros and Cons of Alternatives

2.3.1. Alternative 1 (Flask & Bootstrap)

Pros:
{Insert Pros}

Cons:
{Insert Cons}

2.3.2. Alternative 2 (Node.js & PHP)

Pros:
{Insert Pros}

Cons:
{Insert Cons}

2.4. Comparison of Alternatives
{Compare}

2.5. Conclusion

{Conclude}

3. Analysis for Product Platforms
{Same as Programming Languages}

3.1. Plan

3.2. Choice of Alternatives

3.2.1. Alternative 1:

3.2.2. Alternative 2:

3.3. Pros and Cons of Alternatives

3.3.1. Alternative 1

Pros:
Cons:

3.3.2. Alternative 2

Pros:
Cons:

3.4. Comparison of Alternatives