Table 1: Revision History

Date	**Developer(s)**	**Change**
2020-01-22	Justin Rosner, Justin Prez, Harshil Modi	Initial write-up of development plan
2020-03-22	Justin Rosner, Justin Prez, Harshil Modi	Revision1 of Development Plan

SE 3XA3: Development Plan
Mastermind

Team 204, Trident Inc.

Justin Prez, prezj
Justin Rosner, rosnej1
Harshil Modi, modih1

March 22, 2020

This document outlines the roles and responsibilities for each team member, the standard protocol for team meetings, communication, Git workflow, and coding style, as well as the details for our proof of concept, technology, project schedule, and project review. Our team is developing a new implementation of the classic 1970's game mastermind. Mastermind is a simple, code-breaking game in which the player (the codebreaker) must try and guess the pattern, in both order and color, within a certain number of turns.

1 Team Meeting Plan

Our team plans to meet weekly ~~during the designated lab hours (Tuesday and Wednesday 2:30-4:30pm)~~ from 4:30-6:00pm on Tuesdays and Wednesdays in study rooms on campus to discuss plans and progress for upcoming milestones. If we need to meet ~~to meet outside of lab hours to complete various tasks, the team is generally available to meet Tuesday, Wednesday, and Thursday from 4:30-6:30pm~~ additional times throughout the week to meet project deadlines, this will be communicated via Facebook Messenger, where a time and location will be specified. In person meetings generally should not exceed a length of 2 hours. Location will be determined prior to meeting, with plans made to book a room (at a location on McMaster campus) if necessary. ~~A meeting agenda~~ Meeting notes from each member of the team will be maintained throughout project. For each meeting the date, time, duration, members present, and decisions made will be recorded. Prior to meetings all team members must come prepared to discuss ~~the agenda~~ important topics. Each agenda topic will be lead by a team member, and realistic time will be estimated to spend on each topic at the meeting.

2 Team Communication Plan

Our primary form of communication will be Facebook Messenger. Our team has a group chat on this platform that allows us to communicate with each other instantly about project meetings, tasks and issues. Our project will be stored as a repository in GitLab. This will be the primary version control tool that will allow our team to work on the project simultaneously, while avoiding issues of merge conflicts. We will also use Google Drive to store auxiliary documents and older drafts.

3 Team Member Roles

Each team member has been designated roles to uphold for the project. These roles are subject to change throughout the course.

Justin Rosner	**Team Leader -** is responsible for setting internal deadlines of milestones, and ensuring that all technical work meets the set requirements. Expert on Git and version control.
Justin Prez	**Project Coordinator -** is responsible for organizing team meetings, maintaining the meeting agenda and assigning group tasks. Expert on LaTeX and documentation.
Harshil Modi	**Team Enforcer -** is responsible for editing and reviewing all work before submission, submitting milestones on time, and holds people accountable for assigned work. Expert on software development in Dart and Flutter.

4 Git Workflow Plan

To develop mastermind our group will be adopting a feature-branch workflow described in ”Git Feature Branch Workflow” [1]. This is an extension of the centralized workflow, where individual feature branches will be made for each feature to be implemented. This ensures that no branch is left detached from master for too long (making it hard to merge back in to master) and it ensures that the master branch never contains any broken code. The branches will be named according to the feature being worked on (i.e developmentplan).

Furthermore, this project will make use of git tags to identify the repo in a working condition at major development landmarks. This will allow our group to easily revert back to previous working conditions, and allow for an overview of the progression of the project as a whole.

5 Proof of Concept Demonstration Plan

Our proof of concept will involve creating a GUI which can identify different colours based on user input. Once the proof of concept is demonstrated, adding the logic of the game should be relatively easy. Deploying the mobile application on Android or ~~IOS~~ iOS will pose as one of the greatest challenges for our team, as we do not have experience in mobile development. The POC demo will be considered a success if we are able to get a working Android or iOS simulation running, as it shows that we were able to overcome the hurdle. Testing the validity of the game will also be a problem as sample tests are not available. However, sample tests will be created on the Dart unit testing library based on the rules of the game. In order to show that the app functions as intended, the app will be tested using a mobile application testing plan. Overall, by using the Dart unit testing library and the mobile application testing plan, we ~~can~~ can overcome the risks.

6 Technology

We intend to program the modules of our game using Dart and Flutter, and the application will be deployed using Google Firebase. Our team will be using Visual Studio as the primary IDE for project development. Flutter Test is a unit testing framework for Dart applications that we will use to validate the functionality of our game. Doxygen will be used for documentation and LaTeX will be used for reports.

7 Coding Style

Our team will aim to uphold the style guide standards outlined on Darts website: https://dart.dev/guides/language/effective-dart/style. We will also use a Dart Linter to flag programming errors, bugs, stylistic errors, and improve readibility of our code: https://github.com/dart-lang/linter.

8 Project Schedule

Our project schedule is detailed in the following link to our Gantt Chart.

9 Project Review

Overall, the development of Mastermind and the corresponding documents was very successful. At the start of the project the team had no experience designing or developing mobile applications, so to complete the design and development in a limited time frame is a great success. One of the most challenging aspects of development was coding the user interactions with the buttons. Specifically, our team had difficulties with routing the logic behind each of the button presses to occur upon the user pressing the button. This was a crucial component in development of the program as proper button presses are required to ensure that buttons can be placed properly on the game board.

Additionally, the development team struggled with directly translating the material from the MIS to Dart and Flutter code. The lack of this one-to-one translation is due to the mathematical nature of the MIS and the fact that Dart/Flutter mobile applications are inherently focused on user interfaces. All of this means that the Dart and Flutter code will have to be structured in a way that adheres to the unique way in which Flutter relays data to the screen of the Android/iOS device, and not structured based off the methods outlined in the MIS.

In the future, the team would modify the development plan to assign more specific roles to team members regarding who needs to implement certain sections of the MIS in Dart/Flutter Code. Before starting the implementation, the team thought that it was possible to have the application split into very modular sections that could be worked on individually. However, considerations for the implementation language were not taken into account at the time, as there was substantial overlap in implementing the different modules outlined in the MIS. An example of this is that in order to properly implement the buttons, a member of the development team not only had to figure out how to draw the buttons to the screen but also how and where to route the information upon a button press (which occurred in a separate module in the MIS). It would have made more sense to have one person deal with all aspects of the button, rather than having one person deal with drawing the button to the screen, and having another person deal with the information that occurred upon a button press.

Overall, the team worked very well together, and were able to overcome the difficulties of developing a mobile application for the first time.

References

[1] Atlassian: Git Feature Branch Workflow,
`https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow`