

# Software Development Plan: Revision 0

Group 2 - Genzter

Binu, Amit - binua - 400023175

Bengezi, Mohamed - bengezim - 400021279

Samarasinghe, Sachin - samarya - 001430998

September 29, 2017

Professor: Dr. Bokhari

Lab: L01

September 29, 2017

# Contents

<b>1</b>	<b>Revision History</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Purpose . . . . .	4
<b>3</b>	<b>Team Meeting Plan</b>	<b>5</b>
3.1	Details of Meetings . . . . .	5
3.2	Meeting Agendas . . . . .	5
<b>4</b>	<b>Team Communication Plan</b>	<b>5</b>
<b>5</b>	<b>Team Member Roles</b>	<b>6</b>
<b>6</b>	<b>Git Workflow Plan</b>	<b>7</b>
<b>7</b>	<b>Proof of Concept</b>	<b>8</b>
7.1	Technical Feasibility . . . . .	8
7.2	Identification of Risks . . . . .	8
7.3	Demonstration Plan . . . . .	9
<b>8</b>	<b>Technology</b>	<b>9</b>
<b>9</b>	<b>Coding Style</b>	<b>9</b>
<b>10</b>	<b>Project Schedule</b>	<b>9</b>

# 1 Revision History

Date	Version	Description	Author
29/SEP/17	0.0	Created development plan	Mohamed Bengezi

Table 1: Revision History

## **2 Introduction**

### **2.1 Purpose**

The purpose of this Software Development Plan is to compile all information that is imperative to the development and completion of the project. Included in this document are the team meeting plans, communication plans, member roles, proof of concept, among other documents. This document will continually be revisited and revised. As the project develops and matures, much of the information in this document will change, and as such, it will need continued maintenance.

## **3 Team Meeting Plan**

### **3.1 Details of Meetings**

Meetings will occur 3 times a week. Two of these meetings will take place during the lab section. The third meeting will take place on Wednesday's, at Thode library, after classes. This meeting will be used primarily for technical work, such as implementation and testing. During each meeting, members will rotate filling the position of Meeting Chair. the Meeting Chair will be in charge of writing the agenda for that meeting, which is discussed below.

### **3.2 Meeting Agendas**

As stated above, agendas will be completed by the meeting chair. Meeting agendas will be included in the Meeting Minutes, which will be completed during meetings. This is the [Meeting Minutes Template](#) we will be using. We will also include a summary of decisions made during the meeting, as well as who needs to do what next. The meetings will follow Harvard guidelines, which were discussed in class.

## **4 Team Communication Plan**

The team will primarily use a Facebook group chat to communicate, but will also exchange emails as well as phone numbers. The group chat will be used if any issues come up outside of meetings, or if any group member needs assistance with something non-urgent. For meetings, the Meeting Chair will use both the chat and cell numbers in order to ensure everyone is on the same page. Emails will be used to send/receive data regarding the project.

## 5 Team Member Roles

Team member roles will vary, but the following table can provide a general description of each member's current roles.

Person	Role
Mohamed Bengezi	<ul style="list-style-type: none"><li>- Documentation Lead</li><li>- Designer</li><li>- Developer</li><li>- Test Cases</li></ul>
Amit Binu	<ul style="list-style-type: none"><li>- Development Lead</li><li>- Designer</li><li>- Documentation assistance</li><li>- Test Cases</li></ul>
Sachin Samarasinghe	<ul style="list-style-type: none"><li>- Information/Research Lead</li><li>- Developer</li><li>- Code Review</li><li>- Test Cases</li></ul>

Table 2: Team Member Roles

## 6 Git Workflow Plan

A Git workflow plan will be used to develop the software in an organized and structured manner. A centralized Git repository will be used to store and keep track of development of the source code. There will be two main branches in the repository, “develop” and “master”. The “develop” branch will keep track of the code that is in development. Furthermore, the “develop” branch will contain sub branches that keep track of different features. Once a feature is completed, the specific sub branch that kept track of that feature will be merged to the “develop” branch. The “master” branch will be used to keep track of milestones. A milestone is a specific set of features and functions that software must have and perform. Milestones will be used throughout the development process. Once the code corresponding to a specific milestone is stable, it will be merged to the “master” branch. If any additional steps such as testing is required before the merge, they will be performed on a release branch which will be forked from the stable “develop” branch. Tags will be used to track significant commits. They will be used to label every milestone, significant development commits and features.

## **7 Proof of Concept**

### **7.1 Technical Feasibility**

This project is a feasible undertaking, because the idea in itself is a simple one. Although there will be a slight learning curve with respect to some of the technology involved, the implementation of the project is viable. All group members are technically equipped in regards to a majority of the technologies required. As for required hardware, there are many school resources available in addition to our own computers and laptops, all of which are well equipped to run the required software.

### **7.2 Identification of Risks**

In general, the implementation will mostly include incorporating user input with information retrieval from the JSON file that contains the course data. One part of the implementation that poses the biggest challenge is generating timetables that contain no conflicts, in an efficient manner. This is a challenge because it will be difficult to keep track of each time slot that is taken. With normal timetable generators, the user manually fills in the schedule, but with this one, the schedules will be generated automatically.

Testing will not be difficult. We will manually create a set of schedules that do not contain any conflicts, and use them as a benchmark against the generated set of schedules. This will be done repeatedly, with different courses selected for each set of tests.

As for required libraries and technologies, installation will not be difficult at all, seeing as all are free and work with the software environment that we have setup. Portability will not be a concern, seeing as the project will be hosted on the web, and so any user with access to internet will not have any issues using the project.



### 7.3 Demonstration Plan

For the demonstration plan, we will develop an algorithm that takes in two courses, and generates a set of viable (no conflicts) timetables. This will infer the viability of developing a similar algorithm for 5 or 6 courses.

## 8 Technology

The programming languages used will be HTML, JavaScript, and CSS. More specifically, we will be using Node.js primarily for the back-end of the project. As for the IDE, we will be working with JetBrains WebStorm, which is specifically designed for Node.js projects. The testing framework used will be [Mocha](#). Mocha is "a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun" [1]. In regards to documentation, all documentation will be written and compiled in L<sup>A</sup>T<sub>E</sub>X, and the resulting PDF will be used.

## 9 Coding Style

This project will be using the [Mozilla coding style](#).

## 10 Project Schedule

The project schedule will be included in the GanttProject file. It will be regularly updated, and include all milestones.

## References

- [1] Mocha Framework. *Mocha JS Homepage* <https://mochajs.org/>