

# COMP SCI 4ZP6: Development Plan HubListener

Authored By:  
Ahmad, Zed  
Japlan, Prakhar  
Oliveira, Pedro  
Selvathayabaran, Piranaven

The following official document is an agreed upon plan on how the COMP SCI 4ZP6 Capstone project "HubListener" will be carried out. It contains information regarding how the project will be managed, implemented, reviewed and any other critical details that need to be defined. The Development Plan is subject to change and any modifications will be noted in the Revision History Section.

## Contents

<b>1</b>	<b>Team Meeting Plan</b>	<b>2</b>
<b>2</b>	<b>Team Communication Plan</b>	<b>2</b>
<b>3</b>	<b>Team Member Roles</b>	<b>2</b>
<b>4</b>	<b>Git Workflow Plan</b>	<b>3</b>
<b>5</b>	<b>Proof of Concept Demonstration Plan</b>	<b>3</b>
<b>6</b>	<b>Technology</b>	<b>3</b>
<b>7</b>	<b>Coding Style</b>	<b>3</b>
<b>8</b>	<b>Project Schedule</b>	<b>4</b>
<b>9</b>	<b>Project Review</b>	<b>4</b>
<b>10</b>	<b>Revision History</b>	<b>5</b>

## 1 Team Meeting Plan

The team plans to meet every Monday, Tuesday & Thursday at the pre-defined times listed below. These dates and times are subject to change. The time-slot setup with the supervisor, Dr.Smith, is every Tuesday from 1:30PM - 2:30PM. A member of the team will email Dr.Smith with an agenda 24 hours prior to meeting.

Tentative Meeting Schedule		
Date	Time	Location
Monday	3:30PM - 4:30 PM	Receiving Room - ETB
Tuesday	1:30PM - 4:30PM	Dr.Smith's Office & Receiving Room - ETB
Thursday	2:30PM - 5:30PM	Receiving Room - ETB

## 2 Team Communication Plan

The team has decided on three lines of communication:

1. **Slack:** Utilized for everyday communications regarding tasks or quick questions. Channels will be created as necessary to organize relevant information & dialogue. Any questions, comments or concerns that do not belong on Github/Zenhub will be posted here. Any course updates, information from in-class sessions will be posted here as well.
2. **Github/Zenhub:** Utilized for communication on tasks related to the project implementation or the project management. Information relevant to all stakeholders and may be needed for future use should be posted here. Comments should be made on Github issues such that all parties are aware of details and rationales for decisions. Moving tasks from Backlog to Done will communicate progress with relevant stakeholders
3. **Email:** This is for any communications with Dr.Smith, Dr.Anand and any other third party. These are meant to be formal in nature.

## 3 Team Member Roles

**Scrum Master - Piranaven :** In charge of making sure Zenhub/Github board is up to date. Tasks are well formed and divided equally amongst agile team members. Responsible for speaking to internal and external stakeholders to ensure backlog is filled. Organize and manage sprint planning meetings.

**Note Taker - Prakhar:** Responsible for taking detailed notes during meetings with Dr.Smith. Also, in charge of taking meeting minutes during team

meetings scrums, standups, retrospectives, etc.

**Wiki Administrator - Zed:** Responsible for ensuring that the Wiki Pages on the Github Repo are up to date. Updating links, adding any necessary references and commuting with stakeholders to make sure it is meeting high quality standards.

**Repo Administrator - Pedro:** Responsible for ensuring that the Github Repo is up to date. In charge of making sure README.md's are accurate and verifyin any commits, merges, issues created are of standard.

**Developer - Piranaven, Prakhar, Zed, Pedro:** Responsible for abiding by the coding style rules listed below. Responsible for delivering high quality code which meet the requirements of the tasks created. Provide insight towards design and development decisions.

## 4 Git Workflow Plan

## 5 Proof of Concept Demonstration Plan

## 6 Technology

Programming Language : Currently Javascript

IDE: Visual Studio Code, Atom, Sublime , etc. There are no restrictions

Testing Framework : TBD

Documentation Generation : Honestly,a great question

## 7 Coding Style

Using W3Schools Coding Style as defined below:

1. All names start with a letter.
2. Use camelCase for identifier names (variables and functions).
3. Always put spaces around operators ( = + - \* / ), and after commas:
4. Always use 4 spaces for indentation of code blocks: I.E.

```
function toCelsius(fahrenheit) {  
    return (5 / 9) * (fahrenheit - 32);  
}
```

5. Always end a simple statement with a semicolon.
6. For complex (compond statements)
  - (a) Put the opening bracket at the end of the first line.

- (b) Use one space before the opening bracket.
  - (c) Put the closing bracket on a new line, without leading spaces.
  - (d) Do not end a complex statement with a semicolon.
- 7. Avoid lines longer than 80 characters.
  - 8. Global variables written in UPPERCASE
  - 9. Constants (like PI) written in UPPERCASE
  - 10. NO Shakespeare , just good, clean, meaningful comments.

## **8 Project Schedule**

Gantt chart  
Milestones

## **9 Project Review**

To be completed on second revision.

## 10 Revision History

Table 1: Revision History

Date	Developer(s)	Change
October 8th, 2018	Piranaven Selva	Make Foundation for Development Plan as per Issue #6