



Assignment 2

Project Management Plan

PRESENTED BY: TEAM 152

MONASH UNIVERSITY MALAYSIA
FACULTY OF ENGINEERING
ENG1003: ENGINEERING MOBILE APPLICATION

TEAM 152 PERSONNEL

NAME:	IDENTIFICATION NUMBER
SOH OWEN	29980534
JASON HAN ZHI KWANG	29274206
CHEAH CHING LIANG	29382289
NALAU NOBEL MOURA	29378524

TABLE OF CONTENTS

Introduction	1
1. Project Information	2
1.1. Background and Intended Use	2
1.2. Scope	2
1.2.1. Product Requirement	2
1.2.2. Process Requirements	2
1.2.3. Assumptions	2
1.2.4. Limitations	2
1.2.5. Deliverables	3
2. Information on Personnel.....	4
3. Decisions on Process.....	4
3.1. Practices and Processes	4
3.1.1. Processes	4
3.1.2. Tools Used	5
3.1.3. Repository	6
3.1.4. Documents Locations	6
4. Communication management	7
4.1. Team meetings	8
4.2. Git commit messages.....	9
4.3. E-mails	9
4.4. Communication Conduct.....	9
5. Risk Management	9
6. Class Diagram	10



INTRODUCTION

itGrowsOnYou has opened tender for parties to help them create a proof of concept of an application that can be used to make suggestions for farmers to help producing higher yields. The application is a system concept intended to help farmers in locations with extreme temperatures help them plan on what to grow based on recent temperature readings in order to maximize their produce. This documentation is intended as a guide to our project progress and give detailed explanations to the process of the development of the application.

1. PROJECT INFORMATION

1.1. BACKGROUND AND INTENDED USE

This project is intended to help farmers currently focused in Australia, to overcome low yield caused by not being able to anticipate the changes in temperature that affects their crops yield. It is crucial so that farmers can predict the upcoming weather, therefore plan the suitable choices of crops according to the temperatures predicted based on the past.

The application will require the location of the farm and the details of the crops that are going to be planted there. Using the location coordinates, the application will obtain its history of temperatures from the past 12 months. Farmers also will have to enter the best temperatures for growing each crop. Using these data, the application will suggest which crop that the farmer should plant on the time being.

During this development time, we assume that each location's seasons are at the same time of the year as Australia. We also assume that the temperature of a season is constant throughout the season.

1.2. SCOPE

1.2.1. PRODUCT REQUIREMENT

Develop an application which utilise existing location mapping and weather forecasting service to present user with historical temperature data and help users decide which crop to grow to produce the maximum yield possible.

1.2.2. PROCESS REQUIREMENTS

1. User adds desired locations.
2. User can view any of the location's past weather forecasts.
3. User adds all the details of the crops to be grown with its details of growing temperature.
4. The application will process all the crops' details and suggest which crops to grow at which time according to the crops' specified growing conditions.
5. User is able to delete any locations or crops at any point in time.

1.2.3. ASSUMPTIONS

- The application will be used only in Australia and will follow Australia's seasons.
- Temperature of a season is constant throughout the season.
- Locations accuracy is assumed accurate at city level.

1.2.4. LIMITATIONS

- Locations accuracy is only at suburb/city level. Any locations found that is less accurate than city level, it is considered an invalid position.

1.2.5. DELIVERABLES

The project is expected to be completed on 24 May 2019.

Deliverables	Description	Due date
Feature 1.1	Add an “Add Crop” button on the title bar, opens crop page.	04 May 2019
Feature 1.2	Add an “Add Location” button on the title bar, opens location page.	04 May 2019
Feature 1.3	Delete crop; When a user taps on a crop on main page, crop is deleted with a confirmation both on list and local storage	08 May 2019
Feature 1.4	View a location when tapped on main page, show temperature data for the location	11 May 2019
Feature 1.5	A button to delete a location on respective View location page for each location, when tapped, should remove the location and takes user back to main page.	11 May 2019
Feature 2.1	Add crop by taking in user inputs and create a new instance, which will be saved to local storage. Once a crop is added, takes user back to main page where the list shows the crops added.	04 May 2019
Feature 2.2	Create a Crop class in shared.js, includes instances: <code>_name</code> , <code>_season</code> , <code>safeTempRange</code> , <code>lowYieldOffset</code> , <code>tolerance</code> .	04 May 2019
Feature 3.1	User enters a name of a location, then search the location using MapQuest to obtain its coordinates, and display this information using MapBox on the addLocation.html page. If no valid location is found, treat this as an error and nothing should be added to the locations list.	12 May 2019
Feature 3.2	Direct user to location information when user tapped on one of the locations in the location list on main page. Show weather details for a given date. Also include a button to delete the location.	12 May 2019
Feature 3.3	Use Dark Sky API key to retrieve the weather information on coordinate obtained from feature 3.1 and user can choose any date within the past 12 months to show its weather details. Show the high and low temperature of the date chosen. Before the user choose a date, it should show the current date and its temperature data.	12 May 2019
Feature 3.4	Create a class (LocationWeatherCache) in shared.js to represent a set of historical data held persistently by the application.	12 May 2019
Feature 3.5	Getting data from cache and keep it up to date in local storage. First check if information is already in cache, only if the information is not already in cache should make aa API request from Dark Sky.	20 May 2019
Feature 3.6	Have a representation of dates that correspond to each season in Australia. Determine which crops stored in local storage are in season. Display these on the view page with their safe temperature range.	20 May 2019
Feature 3.7	Check high and low temperature of the day against the crops’ requirements. Determine which crop will produce high yield, which crop will survive but produce low yield, which crop will perish after n number of days, n is any integer.	20 May 2019
Feature 3.8	Display the state for each crop in season on the main page.	20 May 2019
Feature 4.1	Create code to store and retrieve forecasts and crops separately.	20 May 2019
Project Management Plan	Create the Project Management Plan According to the requirements.	26 May 2019
User Guide	Create a user guide for first time user to familiarize themselves with the application.	26 May 2019

2. INFORMATION ON PERSONNEL

Name:	Responsibility:	Email:	Contact number:
Owen Soh	Leader	osoh0001@student.monash.edu	+60 16-608 5228
Jason Han Zhi Kwang	Assistant Leader	jhan0013@student.monash.edu	+60 12-703 7366
Ching Liang	Brainstorm ideas	cche0086@student.monash.edu	+60 17-463 6259
Nalau Nobel Moura	Minute taker, scheduler	nnob0001@student.monash.edu	+60 14-992 4651

3. DECISIONS ON PROCESS

3.1. PRACTICES AND PROCESSES

3.1.1. PROCESSES

The project is divided into several parts, namely brainstorming, task division, development, and finishing parts. Brainstorming involves thinking of ideas that we would like to implement into the application itself, besides the features that are already required by the client. Task division involves allocating equal number of tasks to all personnel of the team. This also involves discussion among team personnel so that all task is given at ideal amount and personnel can do it efficiently. Next, development stage involves doing all the tasks allocated within the given deadline in order to meet client's expectations. Finally, after all the development stage is done, all personnel may add finishing touches to the elements of the application and recheck everything to make sure that all features are working as expected before the presentation to the client.

3.1.2. TOOLS USED

- **Asana**

Asana is a web and mobile application designed to help teams organize, track, and manage their work. In this project, the utilization of Asana allows easier task allocation and progress monitoring that results in a more effective and stable progress. Team personnel are reminded by the due date set in Asana, in addition, communication is also possible through comments.

- **Atom**

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub. Through Atom, all team personnel can view and edit the files related to the development of the project. Unfortunately, it does not allow for real-time coding, which may be able to speed up the development progress.

- **ENG1003 Engineering Mobile Application Server**

This is the server where the application will be tested on. It acts as a real-life situation for when the real application launches officially.

- **Dark Sky API**

The Dark Sky API is used to look up the weather anywhere around the globe, provided that the coordinate for the proposed location exists. In this project, it is used to look up the historical temperature of the past 12 months for a location that the user added, then the information is then used to predict the suitable crop to grow within a specific time of the year.

- **GitKraken**

It is used to exchange coding between team personnel by committing and pulling changes that was made. This way, team personnel can exchange code without any hassle. Also, it enables the team to track versions of each file exchanged, therefore allows the team to revert changes if required.

- **Google Chrome**

A browser was required to test the application after changes were made to the code. The use of Google Chrome also enables the team to use Chrome Developer Tools that allows for easier debugging for the application.

- **Google Drive**

As there are several files to be managed by the team, a storage was needed to store these files. A Shared Google Drive folder was created under the team leader, Owen, with the name “ENG1003 MOBILE APPS” for each member to utilise. This allows for backup files to be added, therefore would not cause a chaos in case of missing files.

- **Microsoft Word**

A word editing document is required by the team to enable the writing of Project Management Plan and User Guide for the application. Microsoft Word was chosen as the word editing software since it is possible to create a livelier document using its templates.

- **MapBox API**

The MapBox API is used to display the coordinates retrieved from the use of MapQuest API on the view location page of the application.

- **MapQuest API**

The MapQuest API is used to find the latitude and longitude coordinates of the location that will be entered by the user in the application.
































3.1.3. REPOSITORY

Folder name: 19-S1-T152

URL: <https://git.eng1003.monash/19-S1-T152/19-S1-T152.git>

3.1.4. DOCUMENTS LOCATIONS

...\19-S1-T152\assignment2\code

- css
 -  style.css
- images
 -  clear-day.png
 -  clear-night.png
 -  Climacons_license.md
 -  cloudy.png
 -  crop26.png
 -  crop26white.png
 -  delete.png
 -  fog.png
 -  icon.png
 -  loading.png
 -  map26.png
 -  map26white.png
 -  partly-cloudy-day.png
 -  partly-cloudy-night.png
 -  rain.png
 -  sleet.png
 -  snow.png
 -  sun.png
 -  view32.png
 -  wind.png
- js
 -  addCropPage.js
 -  addLocationPage.js
 -  dateExtensions.js
 -  mainPage.js
 -  shared.js
 -  viewLocationPage.js
-  addCrop.html
-  addLocation.html
-  index.html
-  viewLocation.html

4. COMMUNICATION MANAGEMENT

Communication name	Communication type	Audience	Purposes	Requirements	Strategies to make the communication effective
Project Management Plan	Internal document	New or existing team members	Communicate procedures and information to team members	“Closed” and “context independent”	<ul style="list-style-type: none"> Needs to fully explain how to use the application, including addressing any known issues.
Client presentation	External message	The client	To hand over the maintenance of the project and code	“Open” and “context-dependent”	<ul style="list-style-type: none"> Every team member must present coherently and equally Intention and context must be clearly explained: structure and functionality of application, design decisions
Interview with demonstrator	External message	Demonstrator	To explain and demonstrate individual understanding	“Open” and “context-dependent”	<ul style="list-style-type: none"> Every team member must be able to answer all questions asked individually, because assignment marks dependent on it
Asana communication	Internal message	Team members	To keep track of the current progress of the project.	“Open” and “context-dependent”	<ul style="list-style-type: none"> Every team member should update Asana about their progress as soon as possible so other members are able to know the progress. Every team member should check Asana daily to understand the current progress of the project.
Git commit messages	Internal message	Team members	To document the changes made to the code	“Closed” and “context-dependent”	<ul style="list-style-type: none"> Every team member should document a meaningful git message on the changes made for other team members to understand the changes easier
Meeting minutes	Internal document	Team members and team manager	To document the past and future progress of each meeting	“Closed” and “context-independent”	<ul style="list-style-type: none"> Content needs to be precise and concise to allow easy understanding of the outcome of the meeting
User guide	External document	The client and user	To guide and show the user how to use the Application	“Closed” and “context-independent”	<ul style="list-style-type: none"> The explanation on how to use the App should be easy to understand Explanations should include images to help the user to understand how to use the App better
E-mails	Internal message	Team members	To communicate between team members	“Open” and “context-dependent”	<ul style="list-style-type: none"> Every team member should reply as soon as possible to the emails Information being conveyed should be clear

4.1. TEAM MEETINGS

Meetings expect all team members to be present. Everyone will reflect on what they have been doing since the last meetings, other members may suggest on actions that can be done to further improve a member's work. Everyone will discuss on what to work on and set deadline for the next meeting.

4.1.1. MEETING MINUTES: SATURDAY, 04 MAY 2019

Members Present: ALL

Location: Discussion Room, Library, Monash Malaysia

Discussions:

- Tasks allocation between team members.

Actions:

- Feature 1

Tasks taken by Jason and Ching Liang, deadline: 11/05/2019

- Feature 2

Tasks taken by Owen and Nobel, deadline: 11/04/2019

4.1.2. MEETING MINUTES: SATURDAY, 11 MAY 2019

Members present: ALL

Location: Discussion Room, Library, Monash Malaysia

Discussions:

- Feature 1 and 2 progress
- Further tasks allocation

Actions:

- Feature 3

Tasks taken by Owen and Jason

- Documentation

Tasks taken by Nobel

4.1.3. MEETING MINUTES: SATURDAY, 18 MAY 2019

Members present: ALL

Location: Discussion Room, Library, Monash Malaysia

Discussions:

- Feature 3 progress
- Further tasks allocation

Actions:

- Feature 4

Tasks taken by Owen and Jason

- Documentation

Tasks taken by Nobel and Ching

4.2. GIT COMMIT MESSAGES

All Git commit messages should state clearly of all the additions or changes that has been made to a document within the repository. Please add a brief description to all changes if possible, to prevent confusion among other personnel. All messages should be presented in a professional and formal language, brief and concise.

4.3. E-MAILS

E-mails are allowed 24 hours a week. All e-mails should be written in a professional manner, without any errors and provide brief and concise information. It should be directed and distributed to all personnel involved within the topics discussed, to keep all them up-to-date. Attachments in e-mails should be checked for prior of sending to prevent confusion and prevent distribution of a virus.

4.4. COMMUNICATION CONDUCT

- All notices and broadcasts are sent through WhatsApp group “Team 152 Mobile Apps”.
- All changes in code are staged and committed through GitKraken.
- All tasks assignment and checklists are done in Asana.

5. RISK MANAGEMENT

Risk	Likelihood	Mitigation
Clash schedules	Very likely	Acknowledge earlier to Nobel, so that any important activity can be rescheduled without delaying the progress of the project
Coding difficulties	Likely	Acknowledge and seek help to other team members immediately before the deadline of the tasks allocated, so that tasks can be completed on time.
Members attitude	Unlikely	Monitor each other’s performance. Always try to assist others if one’s task is done earlier.

6. CLASS DIAGRAM

