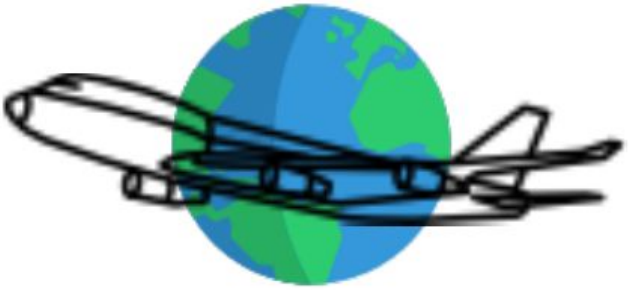
Trip Planner Project

Project Management Plan

Version 1.0.0



Jun Jie Ng, Grant Lu, Joseph Dal Bosco, Logithan Chandrakumar

TEAM 036

26 September 2020

Contents

1. Introduction and Purpose

This document, Project Management Plan (PMP), provides a comprehensive overview of the Team 036 development of the web application known as the Trip Planner. This PMP document delivers detailed information on the web application as well as a detailed description of the development processes including expected deadlines and presentations. The Trip Planner application is a web application that allows its users to plan their domestic travel while possessing functionalities such as accessing all their past and future trips the users had created. This trip planner web application is developed on behalf of Monash University and is aimed to be used by Monash University students for their domestic travel.

2. Project Information

2.1 Background and intended use

The Trip Planner application is developed by Team 036 on behalf of Monash University. The web application is introduced due to the current COVID-19 pandemic where international travels are forbidden. However, the federal government is easing the domestic travel restrictions, Monash University requested the development of an application that allows university students to use and to plan their domestic trips amid the international health outbreak. The client requested a user-friendly application where the user can plan trips within the country and be able to book connecting routes. Furthermore, the application needs to be capable to display trip details and cancel any future trips scheduled by the user. The Trip Planner application also requires to archive all previous trips as history for the user to revisit. The client also indicates the application should be capable to display all the available domestic routes in a selected country such that the user can easily plan the trips.

2.2 Scope

The development of the Trip Planner application is limited due to the following constraints. This includes the assumptions made in order for the development of the application to progress.

- Account details are stored locally instead of on a cloud due to the lack of knowledge to implement cloud storage within the code resulting in loss of data when accessing the application from a different device.
- App was assumed to be designed mainly for iPad Pro landscape resolution layout which means the layout on other devices such as iPhones and desktop will vary and may affect the overall user interface experience.
- The main style of the application is created using Material Design Lite (MDL) References therefore the style of the app will be limited in design choices. This is both due to the limited style variation provided by the MDL and the lack of knowledge in creating a unique layout for the application and use of other styling resources.
- The application will not handle any booking information to external travel agents as it is outside the scope of this project due to the lack of knowledge in transmitting information to third party sites.

2.3 Deliverables/due dates

The list of tasks below are to be completed at the end of the project and each task is dependent on all the tasks listed before it

Table 1. Deliverables and due dates

Task	Detail	Dependencies	Due Date
Project Requirements	Initial requirements of the project after gathering information from client/s. Sorted by features and each feature has its own user story and its associated acceptance criteria.	The transcript of the conversation with the client.	15th of September 2020
Project Design	Structure and overall view of the app created from project requirements. Design decisions are outlined, wireframes and storyboards are created for the app and class diagram for the app data is designed.	All the requirements for the project as requested by the client from the project requirements document.	6th of October 2020
Application Prototyping	Creating HTML and CSS files for the application to match design decisions from project design. Files are structured and named with HTML before using MDL reference to style the pages based on the wireframes of the app.	MDL design choices, color, font and layout outlined from the design document.	13th of October 2020
Application Implementation	Programming the prototype files to interact with one another for the application using Javascript. Utilise pair programming, follow Coding Standards and implementation based on requirements and design document.	Requires all files from prototype phase to be completed before Javascript can be implemented.	27th of October 2020
Application Test Plan	Testing of functionalities or features from the web application after implementation. Creating a test plan and filling out results of all tests and recording ways of fixing any errors that arise.	Requirements, design, prototype and implementation to be finished as the final application must be tested against the requirements document.	3rd of November 2020
Client presentation	Handover of the app to the client explaining the interface, functionalities, limitations as well as the design choices for the app.	All the previous tasks must be completed before presenting to the client.	10th of November 2020

3. Personnel/HR Management

Table 2: Personnel Details

Team Member	Contact Email	Responsibilities
Jun Jie Ng	jngg0073@student.monash.edu	***Information Gatherer*** Looks through Edstem forums and Moodle to gather information for the project. ***Project Planner*** Plans deadlines for project tasks and allocates workload to team members.
Grant Lu	gluu0006@student.monash.edu	***Communicator*** Responsible for contacting the Demonstrator for feedback on tasks through Edstem forums. ***Editor*** Formats layout of documents and tasks to improve readability and overall clarity.
Joseph Dal Bosco	jpdal4@student.monash.edu	***Proofreader*** Reads through all tasks written by the team and checks for spelling, grammar and text formality. ***Project Leader*** Coordinates the team and ensure team members are working on the task and are on schedule.
Logithan Chandrakumar	lcha0064@student.monash.edu	***Time Manager*** Manages the time and date of each meeting and makes sure all tasks are being completed on time. ***Meeting Coordinator*** Sets up Zoom each time to facilitate team meetings.

4. Decision on Processes

To facilitate document control and document storage, the team will be utilising a shared Google Drive. The assignment folder is to be contained within this Google Drive, which is where all documentation and reports will be stored. All documentation can be accessed and edited by all team members by utilizing the shared functionality of Google Docs.

For the purposes of coding this web application, a remote team git repository will be created using the web based repository ‘Git Lab’.This is to be used by all team members to access, and edit all HTML and JavaScript files. All team members are required to create a local repository on their own device. To ensure minimal conflicts when coding, it is required by all team members to pull the remote repository before beginning any coding session. The team members are also expected to notify the team when making changes to the code before and after the coding such that the conflicts are kept

to a minimum. When a team member completes a session of coding, they must stage, commit and push all changes made to the remote repository. All team members must be notified when any new push has been made to the repository. When team members are notified of a new push to the remote repository, they must pull the changes to their own local repository before they begin their next coding session to minimise the possibility of conflicts.

Testing and debugging of the web application is to be completed using Google Chrome and its built in debugging tool. HTML documents are to be opened in the Chrome browser and tested to check if functionality is as expected. When issues arise, the debugging tool should be utilised to step through the code and identify areas that need to be amended.

5. Communications Management

Communication is handled by the combination of Facebook Messenger, Zoom and Trello. Facebook Messenger is used to place zoom meeting times and a platform for team members to have a general meeting over text if something urgent comes up during the time of the project development. Trello is used to set all due dates and to keep the whole team on track. The expected response time over messenger is 30~60 minutes and for a zoom meeting it is 1~5 minutes. When the response time is exceeded, the group member who is not responding will be contacted on a different communication platform to check in on that group member because the increased response time could be for a multitude of reasons. Zoom meetings occur approximately once to twice per week for all members of the team to discuss and complete the relevant tasks for that week. It's also used to check on the progress of each team member and keep the team on track for the project. Before each Zoom meeting a meeting minutes document is created in the team Google Drive to plan the tasks expected to be finished during that meeting. At the end of the meeting, the meeting minutes document and Trello are updated to reflect the completion status of that task.

6. Quality Management

To ensure correct functionality of the web application and to make sure the quality of work completed is not compromised throughout the lifecycle of the project. All team members are expected to meet their respective quality assurance tasks and quality control procedures as outlined in Table 3.

Table 3: Quality Management Details

Step	Personnel Responsible	Quality Assurance Tasks	Quality Control Procedures
Review for Client's Requirements	All team members	Analyse the transcript from the interview with the client to determine requirements.	All team members should read and take notes of the transcript, ensuring all requirements are met.
		Development of requirements documentation and seek external feedback	All feedback should be considered and amendments made to requirements document.

Code Development	All team members	Code written meets all coding standards	Team members need to be aware of coding standards and fix them when any standards are not being met.
		Correct use of team repository	All team members are expected to pull the repository and notify the team before commencing work, and push the repository and notify the team at the completion of work.
Web Application Functionality Testing	All team members	Test all buttons on web pages	Check buttons on all web pages, ensure they work as expected when clicked.
	Jun Jie Ng & Grant Lu	Test account creation and login/logout functionality	Check the account creation page and attempt to create a new account. Then check that this account can successfully login and logout.
	Joseph Dal Bosco & Logithan Chandrakumar	Test scheduling of a new trip and the scheduled trips page	Attempt to schedule a new trip to check that the correct information is displayed on the web page and the map API. Then check that the scheduled trips page includes all scheduled trips.
	Grant Lu & Joseph Dal Bosco	Test trip history page	Check that the correct information is displayed for the currently logged in user.
	Jun Jie Ng & Logithan Chandrakumar	Test detailed trips page	Check all correct information is displayed on the web page and map API.
Report and Documentation standards	All team members	Ensuring correct grammar, punctuation, writing style and structure is used in all documentation	Thorough proofreading should be completed, and any obvious mistakes fixed immediately and any ambiguous errors consulted with other team members to determine what action is to be taken. Where possible, templates should be utilised for documentation.

7. Risk Management

This section outlines the various risks involved in the development process of the trip planner web application project. Team members are expected to be aware of the potential risks involved in the project and understand the strategies to prevent and/or resolve these occurrences. All project risks, likelihood and management strategies can be found in Table 4.

Table 4: Risk Management Details

Risk	Likelihood	Management strategy
Git conflict (Merge Conflict)	Medium	A management strategy could be to increase communication between the members in the Git server. Merge conflict happens when more than one group member working on the same project under the same git repository changes the same line of code in two different ways. This leads to Git not being able to tell which one is right, thus this risk was given a medium likelihood because Merge conflict can occur frequently even within a group with good communication.
File Loss (Fail to Save/Deleted Files)	Medium ~ High	A management strategy could be to continuously save code every 5~10 minutes on VsCode to reduce the risk of forgetting to save and losing 10 + minutes of work/code. The chance of this happening is Medium ~ High due to the fact that it is easy to get carried away while writing code and forgetting to save will be a likely outcome.
Fail to Backup	Low	A management strategy could be to make sure the backup process is completed with a steady internet connection and is backed up more than once.If any complications arise during the backup stage it will result in the loss of all of the work being backed up on the server. The chance of this happening is considered low due to the fact that complications arising exactly during the backup time is rare.
Failure to Communicate Coherently within the team	Low	A management strategy could be to make sure to have regular team meetings even if you have been allocated to do a certain part of a code by yourself. This will allow all team members to be briefed equally thus decreasing the chance of mistakes due to lack of communication.
Failure to assign team members to their strengths	Medium	A management strategy could be to have a team meeting and to set individuals of the team towards their strengths according to their preexisting skills. For example if one team member has past experience with HTML they can do the set tasks for the HTML part. If team members are assigned to their strengths, the overall end product will be of a higher quality. This risk has a likelihood of Medium, this is due to the fact that some team members may not speak up or contribute their strengths as easily in group tasks.
Failure to follow task requirements	Low~Medium	A management strategy for this risk could be to make sure every team member has read the task requirements coherently. This will reduce the risk of making major mistakes relating to the task transcripts and requirements which would lead on a fault end product. This risk has a likelihood of Low~Medium due to the fact that it is very easy to miss major points of the task requirements even if it is proofread by all team members.

Failure to Complete set Tasks on Time	Low	A management Strategy would be to increase communication between team members, and to have reminders on Trello and other communication methods. This would allow for tasks to be completed on time. This risk has a Low likelihood due to the fact that there are alot of communication tools that are being used to set reminders, thus it is a very low chance a team would miss tasks to be completed on time.