MONASH University

Assignment 2 Project Management Plan

ENG1003 - Team 44

- Linton Charles
- Khanh Le
- Ryan Verma
- Hashini Udugoda

Contents

Introduction and Purpose

For our project, our team is aiming to design a web application that allows users to book a domestic flight. We aim to implement an interactive map where the user can select a specific country for their travels, the departing airport and destination, as well as connecting airports. This app should include login and sign-up features and also allow the user to view their flight history.

This Project Management Plan document is divided into six main sections: Project Information, HR Management, Decision on Processes and Communications, Quality and Risk Management.

In the 'Project Information' section, some background information about our project is provided, including the aims and limitations of the web application. Our project deliverables and due dates are also summarised in a table, which outline the specific tasks to be completed for each week in order to meet the project deadline. Under 'Personnel/HR Management' the responsibilities of each teammate are detailed. Furthermore, information regarding our team's organisation has been given in the 'Decision on Processes' section. Here, the programmes and applications we have been using are explained in detail; how we use them organise our work and how often we utilise them.

The 'Communications Management' sub-section explains how our team has kept in contact, what applications we have used to do this as well as the expectations of each team mate with regards to communication and punctuality. Our quality control procedures for coding and team documents are outlined in 'Quality Management'. Finally, the 'Risk Management' section explains potential factors that may hinder our progress and jeopardize our project; the likelihood of these risks are evaluated and possible solutions are explored.

Project Information

Background and intended use

Many universities have exchange programs which allow students to experience and learn different cultures by studying abroad in a different country. Many students take advantage of this experience to visit different areas of the city and country. Although the COVID-19 pandemic has led to the closing of international borders, steps are taken by the federal government to allow domestic travel in an initiative to improve the nation's economy.

Monash University, which offers many exchange programs for students, has been sponsored by a flight booking company to create a web application to aid students who are planning their trips within the country. The aim of this project is to create this web application on behalf of Monash for students to use. As such, the target audience of the application will be university students on an international exchange.

The criteria of the application involves a user-friendly interface and information available instantaneously. In addition, there is a strong preference against the sight of tabs. The application should allow the user to plan a trip within the country and be able to book connecting routes. The user should also be able to see information about all available domestic routes in their selected country. Information about the trip, ability to cancel trips and earlier trips should also be available to view in the application. The application should allow all users to search and book a trip but the ability to save scheduled trips are for users who sign up to the web application. Once a trip has been scheduled, the trips will be booked by the flight booking company.

The target device for this web application will be the iPad Pro in landscape view although other devices and orientations are also taken into consideration.

Scope

The web application is limited to domestic routes within a country selected by the user. Transnational routes have been excluded due to the purpose of the application being to encourage students to experience the culture of the country in which they are undertaking an exchange. The user is also unable to change their username, password and email address after they have created an account as this has not been stated to be a requirement by the client. Users who are not logged in will have limited access to the web application but will still be able to book a trip as per the client's request. As a web application, the user is limited to viewing it with a web browser.

Deliverables/due dates

The ability to begin tasks is dependent on the completion of all tasks in the preceding week.

Starting Week Date	**Tasks**
14/09/2020 Week 7	*Completion and submit for feedback and marking* - Requirements Document - Project background - Application Features - User Stories *Completion* - Requirements Phase
21/09/2020 Mid-semester Week 1	*Complete draft and submit for feedback:* - Wireframes - Storyboard - Class Diagrams - Project Management Plan - Design Document

28/09/2020 Mid-semester Week 2	*Completion* - Design Phase - Wireframes - Storyboard - Class Diagrams - Project Management Plan - Quality and Communication Management. - Project Info and Personnel - Decisions on Processes - Risk Management - Prototyping Phase - HTML code - MDL design and layout - Auxiliary CSS
05/10/2020 Week 8	*Completion* - Implementation Phase - All Javascript code. - All Classes and Functions. - All software drafting.
12/10/2020 Week 9	*Completion* - Verification Phase - Test Plan - Comprehensive Code Review
19/10/2020 Week 10	*Completion* - Client Handover Phase - Presentation Slides - Transcripts - Rehearsals
26/10 Week 11	*Submission* - Full Assignment
3/11 Week 12	*Presentation* - Client Handover

Personnel Management

All team members share the responsibility of proofreading and formatting of documents. Design decisions are decided by all members and tasks distributed evenly among members on trello. Administrative roles such as taking meeting minutes and chairing the meeting are rotated every meeting.

The storyboard and class diagrams for the application was designed simultaneously by all members in order for instant feedback and discussion. Labelling of wireframes was divided into four equal parts (Fig. 1) and distributed to members. The following section details the allocation of tasks up to the current point of the project timeline:

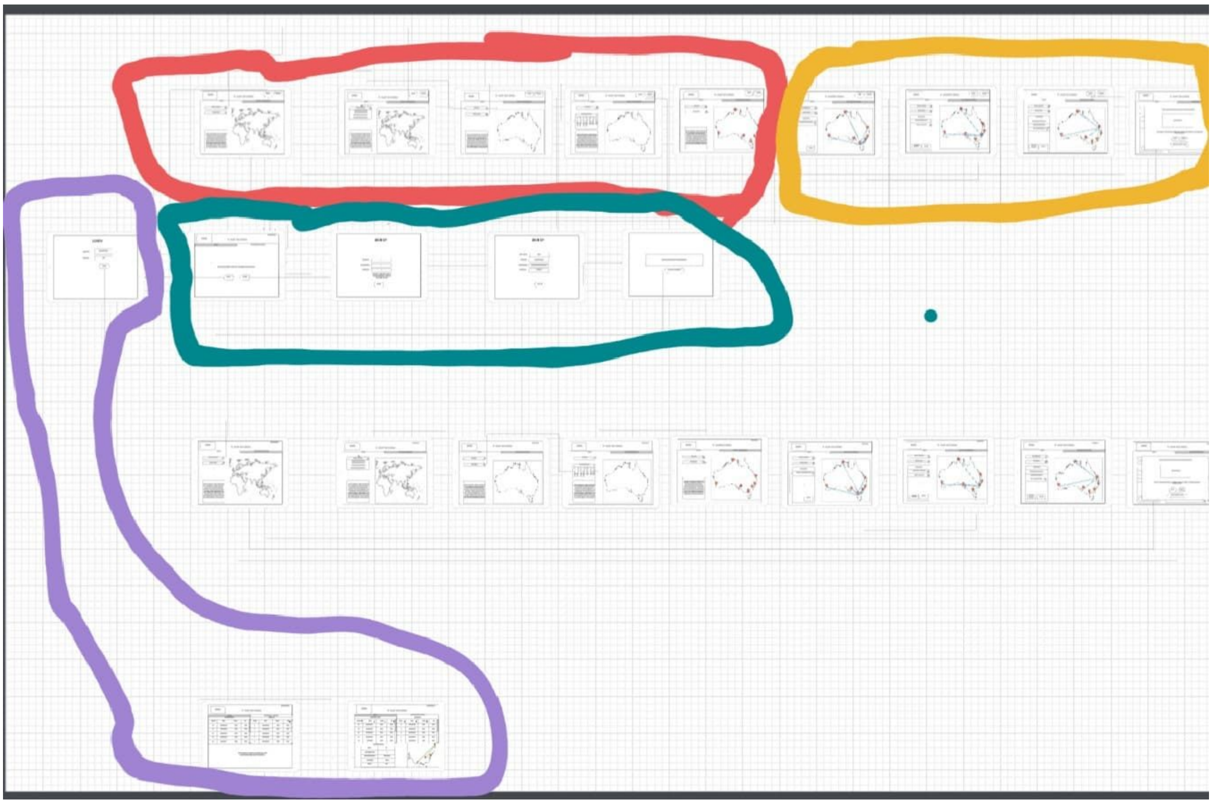


Fig. 1 - The allocation of Wireframes to each individual member

Linton Charles

Email

Requirements Document

- Scheduled flights display user stories
- Airport display & route selection user stories.
- Application Features

Design Document

- Collaboration on Storyboard.
- Red Wireframe Set (Fig. 1)

Project Management Plan

- Communications Management
- Quality Management

Hashini Udugoda

Email

Requirements Document

- Trip confirmation user stories.
- Country & Date selection user stories.
- Application Features

Design Document

- Collaboration on Storyboard.
- Yellow Wireframe Set (Fig. 1)

Project Management Plan

- Introduction and Purpose
- Decisions on Processes

Ryan Verma

Email

Requirements Document

- Flight History user stories
- Account Creation user stories
- Application Features

Design Document

- Collaboration on Storyboard.
- Purple Wireframe Set (Fig. 1)

Project Management Plan

- Risk management

Khanh Le

Email

Requirements Document

- Application Features
- Project Information
- Formatting

Design Document

- Collaboration on Storyboard.
- Green Wireframe Set (Fig. 1)

Project Management Plan

- Project Information
- Personnel Management

Decision on Processes

Our team has been using numerous programmes and applications to store or organise our work effectively.

Storing and editing work

The main tool our team has been using to collaborate and store our work on has been Google Drive. Most of our work, including the Requirement Document, Project Management Plan and Design Document) is stored in a shared google drive. This is useful because it allows team members to work on the documents together and make edits at the same time. New changes are instantly saved and there is minimal lag.

The platform 'Lucidchart' was used to work on our Wireframes, Storyboards and Class Diagrams. We decided to work on this together as a team to ensure that we were all on the same page with regards to the design of our app and all of its functionalities; it is essential that our team has the same vision for the app as this will prevent future conflicts when it comes to the prototyping and coding phases.

As we move onto the prototyping phase of our project, we will be using the programme 'Visual Studio Code'. As we will have to work on the code together, we will be storing and updating our work onto git. This will allow us to work collaboratively on the code for the Web app in an organised and orderly manner. The git-push and git-pull functionalities will help to prevent clashes and inconsistencies with our codes.

Task Allocation and Management

For the tracking and allocation of tasks, our team uses the web application 'Trello', which is a Kanban based project management tool. It allows us to create cards for each task, assign it to a list, allocate team member(s) to it and follow its completion. We have used the default and most popular three tier list system of To Do, Doing and Done in order to track the completion of tasks. Trello also allows the writing of detailed descriptions and the assigning of due dates which come in handy for project management as well.

The procedure we use to allocate tasks is to identify the most urgent and important phase/item for completion and split it into four equal components. Then we come up with brief titles for these components which become Trello cards in the To Do list. After this, each of us volunteer to pick a task they like most and then go ahead with its completion. The presumed procedure to move tasks to other lists is, when a member begins a task they move the appropriate card to the Doing list and when they complete it, they move it to the Done list. Flexible deadlines are agreed upon through Facebook messenger which usually aim to get far ahead of the hard deadlines allocated through Moodle by the University.

Communications Management

In order to facilitate satisfactory project completion, an effective communication management system being utilized is essential. The sections below detail the different processes, procedures and components of our communication management system along with the rationale behind our decision to choose them for this project.

Meeting Scheduling Procedures

Since this is a small project with a tight-knit team of 4 students, meetings are arranged on a semi-ad hoc basis. To elaborate, our team used this nifty web application called when2meet.com that allowed all of us to display the times we were free throughout the week and figure out when all of us were expected to be free to attend project meetings at the same time as seen in (Fig.2). These hours would then become the window in which meetings are supposed to be arranged unless an alternative time is agreed upon by all team members.

The De Facto system through which meetings are arranged consists of two methods. The first is at the end of the previous meeting when we talk to each other through the Zoom video conferencing software. Meetings in this method are arranged by brief, informal discussion and concurrence. Usually a team member proposes a meeting time and purpose then the rest either agree or object. When an objection is raised, an alternative meeting time is proposed and the process repeats. Since we have already discovered a general compatible window of free time using when2meet.com previously, this process is not as inefficient as it seems. In fact it is the simplest, most natural and most appropriate procedure that we have come up with for meeting scheduling. The second method is identical except for the use of Facebook messenger where we use written communication instead of verbal video conferencing.

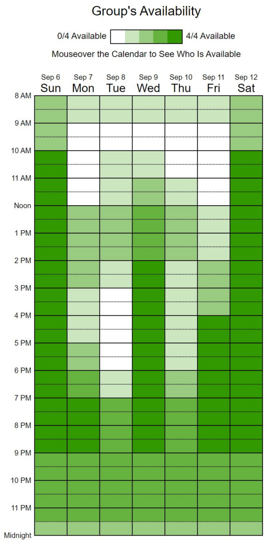


Fig. 2 - Discovering common availabilities (From when2meet.com)

Communication Platforms

We have decided to utilize relatively informal means of communication within our team due to their efficiency & ease and because of the fact that we are comfortable with it. To this end we primarily use two software platforms - Zoom video conferencing software and Facebook Messenger.

For communication that is outside meetings and practical classes, we use the group chat feature on Facebook messenger, an instant messaging application that allows us to communicate with all team members with brief informal written text. This platform also allows us to upload a wide variety of files and information including images, pdf files and links to web resources. Another feature of this platform is the ability to call a contact with video and audio but this feature is used less often by our team for lack of necessity reasons. The main benefit of Facebook messenger is its instant notification system which consists of vibration and noise when a message is received. This allows relatively instant communication which improves productivity and quality of life through time-saving and instant replies. But these text messages are often too time-consuming to write hence somewhat more inefficient than Zoom video conferencing.

We were introduced to Zoom through Monash University's use of it for conducting workshops and practical sessions. All of our team members immediately recognized the software's high quality and its potential for conducting virtual meetings that simulate live conference room meetings as close as possible with its real-time visual and audio communication features. This enables us to use the fastest and most natural means of communication - verbal discussion, in order to conduct scheduled meetings. Zoom also has the ability to mute microphones, turn off front cameras and mute the output volume (through the Windows volume mixer) which allows for a high degree of control to enhance comfort. The software also has the ability to create breakout rooms, view and control another person's screen while allowing the sharing of links and brief text messages through the chat service. All of these features make Zoom the most appropriate software platform through which to conduct meetings.

Response Times

There are two times that team members are expected to adhere to - 1 day for responding to Facebook messenger queries and 15 minutes for meeting lateness which was not notified by the late member to the team through group chat, either previously or within the response time. We have a 3 strike rule to deal with breaches of these response times. For each violation, the team member is reprimanded by the rest of them and after that member has exhausted all 3 strikes, the rest of the team should inform the demonstrator of their conduct.

Quality Management

Our team's quality assurance and management system consists of a 3-tier system of documentation editing & proofreading, review of software and incorporating demonstrator feedback into this project. Rather than assigning individual team members to quality assurance, we have decided to collectively take on this incredibly important responsibility. This was done in order to ensure a higher standard of review by combining the abilities of all available human capital.

Editing Documentation

The 1st component of this 3-tier system is the process of refining all documentation. An exhaustive list of documentation undertaken for this project in chronological order consists of the,

1. Requirements document.
2. Design document - including wireframes and UML diagrams.
3. This Project Management Plan you are viewing.
4. All documentation in software files which consists of,
    - File header documentation.
    - Function headers.
    - In-line comments.
5. Test Plan document.
6. Client presentation.

The process of editing and reviewing this documentation is relatively straightforward. After their completion each of us simply read all of the files and inform the team on any changes we'd like to make. After receiving unanimous approval, each of us implement those changes. That particular piece of documentation is finalized after either everyone is satisfied with it or when it requires urgent submission during exceptional circumstances. We will try to avoid the latter scenario as much as possible, by the way.

Code Review

The use of GitLab and VSCode's Git features enables us to efficiently modify code, receive updated code quickly and review & revert changes as necessary. Unlike large open source projects, we have decided not to mandate code review, by other members, for every commit or push with the exception of a merge conflict where it is necessary to determine what code to delete which may involve modifying or erasing another member's code. The following is the general procedure that is followed during code review.

1. Remove Syntax Errors.
2. Remove Logic Errors.
3. Ensure code follows Intended Behaviour.
4. Ensure coding standards are adhered to.
5. Ensure good software design (High Cohesion & Low Coupling).

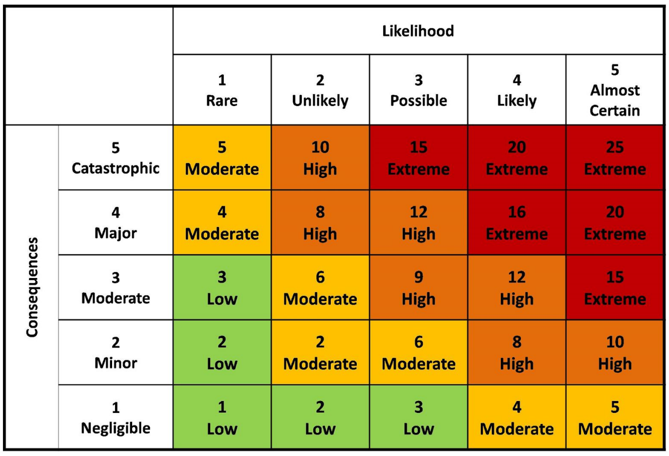
In this project, code review will initially be conducted individually. During the verification phase, the team will deploy a test plan alongside a final and comprehensive code review process which follows the above procedure as well, but more exhaustively.

Demonstrator Feedback

Finally the central component of the 3-tier system is demonstrator feedback. All of our team members are relatively unskilled compared to University staff hence we should rely on them for crucial feedback to our project. We intend on asking questions regularly, including questions that ask for confirmation, clarification, explanation and general feedback. We plan on implementing most, if not all of this feedback into our project. The stand up meetings will be a chance to receive regular feedback and surface level review of our work.

Risk Management

This project is being conducted by only 4 people. Due to the small nature of the team as well the workload required, a lot of risk is involved that could potentially bring the project to a halt, or worse, end up being incomplete. The three main threats to this project are Human, procedure or circumstantial threats. The image below (Fig. 3) shows the quantitative risk parameters used to assess these threats in the form of a matrix:



		Likelihood				
		1 Rare	2 Unlikely	3 Possible	4 Likely	5 Almost Certain
Consequences	5 Catastrophic	5 Moderate	10 High	15 Extreme	20 Extreme	25 Extreme
	4 Major	4 Moderate	8 High	12 High	16 Extreme	20 Extreme
	3 Moderate	3 Low	6 Moderate	9 High	12 High	15 Extreme
	2 Minor	2 Low	2 Moderate	6 Moderate	8 High	10 High
	1 Negligible	1 Low	2 Low	3 Low	4 Moderate	5 Moderate

Fig. 3 - Risk Matrix (From: hastam.co.uk)

Potential risks with associated contingency plans:

- Particular task member's work is not completed by the team's set deadline [Likelihood: possible (3), Consequences: Major (4)] risk score: 12

In order to mitigate this risk, the team member will need to let the other team members know as early as possible when and why he is unable to meet the deadline. This will let the other team members work that part of the project to ensure the deadline is still met and the team is able to follow the overall project deadlines.

- A team member is unable to work for the rest of the project due to unforeseen circumstances [Likelihood: unlikely (2), Consequences: Catastrophic (5)] risk score: 10

The chances of this happening are very unlikely, however this is not preventable. In order to reduce and keep damage to a minimum level, the rest of the team will take on this particular team member's workload equally and the set deadlines will change accordingly, but still be before the official project deadline.

- The project deadline is not met [Likelihood: unlikely (2), Consequences: Catastrophic (5)] risk score: 10

In order to mitigate the risk, the team will aim to finish the project much earlier than the set deadline and also have goals/expectations of work that needs to be completed each week. Trello is being used to note down the team's journey towards completion so everyone in the team is kept in the loop. Trello ensures that the project deadline will not be exceeded even if a team member falls ill or if the work associated with the project needs to be revised.

- The quality of the code is poor [Likelihood: likely (4), Consequences: Catastrophic (5)] risk score: 20

The chances of this are quite high as all team members only have 2 months of coding experience. In order to reduce the risk of this happening the following things will be done

- code reviews
- testing of all code
- use clear coding standards and guides

- Poor productivity among team members [Likelihood: possible (3), Consequences: Moderate (3)] risk score: 9

Many team meetings could potentially have poor productivity. In order to have good productivity in all team meetings

- all team meetings will have an achievable end goal that must be done before the end of the meeting
- set a good working culture among team members
- have a team leader who makes sure that the team is on task at all times
- have meetings be no longer than 2 hours so that productivity levels are constant throughout