Deliverable 2

Group work

Course code: COS60010
Tutor: Dr. Eureka Priyadarshani
Team members:
Lucas Qin
Natalie Huang (103698990)
Mona Ma (103699029)
Minji Kim (103510175)
Word count: 2805
Date: 15/09/2021

Table of Content

DELIVERABLE2

1.0 Introduction

P&NP is a small software development studio in Melbourne, Australia. The company is expanding rapidly, and most of its work is contract-based in areas like building small apps and automation scripts for external clients in different fields. As the employee base of the company is growing fast, P&NP finds it challenging to manage the project tasks and resources. However, P&NP is not satisfied with the existing products on the market. Hence, P&NP wishes to build a new app to help them manage the to-do list in allocating resources and monitor ongoing projects.

In this article, the priority requirements for the P&NP company will be discussed first. Secondly, the features we plan to implement to meet these requirements in the web application. Then we explain the technologies which are used to achieve the functions in the front end and back end. After that, the design of the web application is displayed, which includes a login and register page, to-do-list page, progress page, assign and request tasks page, and recourse page. Finally, we talk about the timeline of the whole project and analyze the risks we may meet and how we control these risks.

2.0 Features and Requirements

2.1 Feature plan on implementing

2.1.1 To-do list

The proposed to-do list design is shown in Figure 6. The to-do list is designed similar to the Kanban-style as P&NP is looking for a board interface in this style. On this board, the functions for adding new tasks and removing unnecessary tasks will be available to users.

There are three sections under the board. ‘To-do’ list, ‘In progress’ list, and ‘Done’ list. Users are also able to manage their tasks by moving different tasks in between each section.

When a task is assigned to different team members, they can also tag other team members under each task.

Page function: To-do list (Kanban style).

- Add new task
- Remove task
- Move each task to another section (From ‘In progress’ to ‘Done’)
- Tag team member

Page structure.

- Team board
    - To-do list
    - In progress
    - Done
- Individual board
    - To-do list
    - In progress
    - Done

2.1.2 Project progress

The proposed project progress page is good for the team to review the overall progress and performance visually. The page also provided a Gantt chart-style road map for the team to manage the progress easily. If the team is unsure about what project is due the

most recent, they can check it from the calendar which shows the due dates and any important event dates.

- Calendar
    - Check important event
    - Check important due dates
- Performance board
    - Check overall performance
    - Check to-do progress
- Team member board
    - Add new members
    - Remove member
    - View team member
- Road map
    - Adjust timeline
    - Change sprints
    - Represent a linear schedule

2.1.3 Project resources

The proposed project resources page is shown in Figure 9. The team can manage various resources associated with different tasks. Team members are allowed to upload resources, including images, links, and other files. They can also leave a note while uploading these resources. The project resources folders are sorted by task names. The team can find files documenting information related to projects and company protocols in individual task folders.

- Project resources
    - Key resources
    - Individual resources
- Upload new resource
    - Upload images
    - Upload files

Leave notes

2.1.4 Assign and request tasks

The proposed assigning task page is shown in Figure 8. The manager can assign tasks to team members by creating a new message. The team members can check their inbox messages from managers or other employees to respond to the assigned tasks.

- Assign tasks
- Check message
- Create new message

2.1.5 Notification bar

The proposed notification bar is located at the top-right of every web page. It will show the number of new messages from the team. By clicking the notification button, the user will be led to the assigning task page.

- New notification

2.1.6 Search bar

The proposed search bar is designed for the team member to search for any information within the whole web page.

- Search for information

2.2 Key technologies and libraries

2.2.1 Front end

Technologies used in UI design

Figma for Wireframe and Prototype

When we all finished the priorities of our product features that we needed to make and wrote down the user story. And then, in preparation for the users to actually use our web application, we created the user task process. Based on the user story and task process we finished the rough sketch step and moved on to the wireframe part.

We utilized Figma to create the wireframe and prototype. Overall, a consistent design was created. Each page of wireframes shows the essential functions of our final product. The prototype is designed to be readable and neat.

Technologies used in Front-end web development

HTML, CSS, JavaScript

While we have the prototype design of the web pages, HTML language is used in building the skeleton of these web pages. Before writing the HTML code, we firstly need to draw boxes around the content for the layout to determine the columns and boxes we have for each web page. After having the boxes and columns, we need to determine the main text elements such as headings, images, paragraphs, and links. Determining their styles, fonts, heading sizes.

When we have all the details available, now we are ready to start building. Firstly, we create the “*.html” file for each web page. Coding them refers to the boxes and columns we have, editing them for responsive design.

When the skeleton is ready, we are going to style the content using CSS language. To style the content, we create the “style.css” file separately. During the HTML process, we’ve already named different id and class for tags we want to add style to. We just simply add styles to these predefined tags. We may want to create separate CSS files for individual web pages, as some pages might be complex in styling them.

JavaScript is a programming language to be used on the client-side and server-side. That means in the front-end web page development, we need javascript to make web pages interactive. In our project, a typical function is the log-in and registration. Javascript can be used to check input data while the registration information in a correct format is essential. Bootstrap and React are proposed to be used in this project.

PHP

While JavaScript is used to check data on the client-side, PHP is a scripting language on the server-side. It doesn’t execute data within the browser as JavaScript does. It helps in sending the data which has been checked and approved by JavaScript to the server-side. In our project, the user might enter their personal information to register for the web page. We use JavaScript to make sure the information, for example, the first name, last name, and email address are valid. The PHP files will receive the information and send it to the server.

2.2.2 Back end

Database design

In this project, SQL is used to create and manage the database of the web application. First of all, we find out the entities in this situation as well as the relationships between them in the real world. After that, we decide what data will be needed in building this web application and what data we should save on the server-side. Finally, all attributes

from the entities have been chosen. The following is the E-R diagram of the database process.

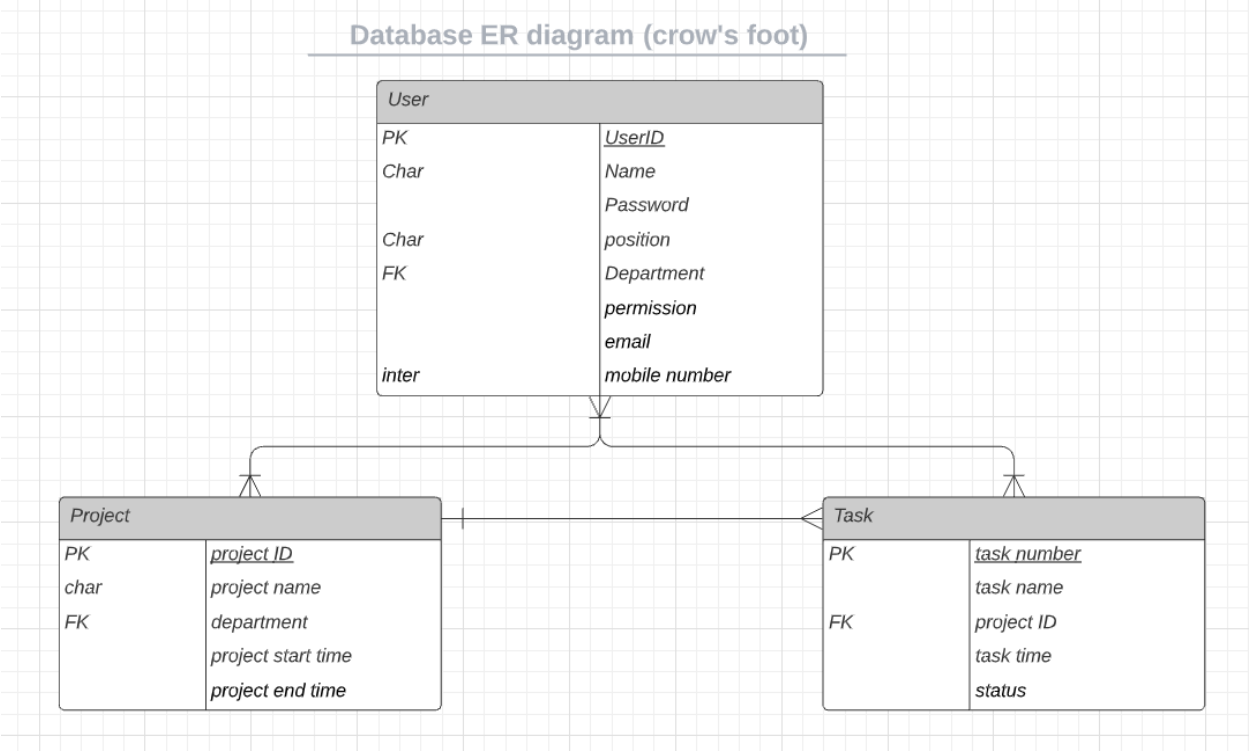

Figure 1. E-R diagram of the database

Design the structure of the database by SQL

In order to build this web application and achieve the purpose of all these functions, a database has been created with three tables to store basic information from employees and projects by MySQL. The first table includes the users' information, the second table consists of the project details, and the last table contains the tasks and who they are assigned to.

Management database

All these tables have their own primary keys and foreign keys, which ensure all data is independent and all three tables can be linked together. There are also some constraints for these tables, such as no null data when a user registers a new account to ensure all the

updates satisfy the database integrity. These constraints ensure the general operations of the database. It will help users check or update any information from the projects and themselves. After that, all users can use the application to find their own tasks and adjust them.

Connecting database and Embedded SQL

All the data from the front end needs to be sent to the server and saved in the server. The user should see parts of the database and update them. Therefore, the database should be connected by 'require DB.php' and 'DB::connect' in PHP which link the database to the front end [2]. We will also embed SQL into PHP to implement some complex operations, such as queries. When users edit the web page, such as changing one to-do task to a process one, the status of the task changes to the process. The database cannot read from users straightly, and the embedded SQL help sent the command from front end to back end. It will update and save the new data into the database. After that, it is sent back to the front end, and users will see what they changed and achieve the functions.

Some SQL will be embedded in python to achieve some complex selection and update of the database. At the same time, it also guarantees the integrity of the database.

Server-side technology stack

Python and Django

In this project, we are going to use Python and Django as a framework to build a server-side. To begin with, Python is a free open-source high-level programming language with great popularity and huge developer community support, it can be used in a range of applications including data visualization, machine learning, backend of a website, etc. It's easy for beginners to get started compared to other languages because it

uses many plain English words in its code. Python is also a database-friendly language which offers interfaces for all major DBMS systems.

Apart from the programming language itself, we also use Django to help build the server-side. Django is a free open-source framework based on Python. There are many websites that rely on the Django framework including Pinterest, NASA, Spotify, etc.

In this project, Django's defined server-side structure is shown in Fig 2. The basic idea is that there are 3 most important parts in this structure which are Model, Views, and Controller, and it is also known as the MVC framework. In this framework, the controller receives a request from client-side and validates and interacts with models, which is a database that stores and organizes data for a web application. After the controller retrieves the data user requests, it will proceed those data to views, and views will cognate all relevant HTML and CSS files and present the content to the browser.

To get started with Django, there is a process we are going to follow.

1. Install Django in a virtual environment;
2. Define model in medels.py then activate model in settings.py;
3. Create a superuser in admin.py and go to the admin interface via a browser.
4. Define URL pattern in url.py;
5. Edit views to connect with specific URL pattern in views.py;
6. Edit base template that applies for all pages in base.html;
7. Create URL, views, and templates for different topics;
8. Create forms.py to process users' type-in information, at the same time make the corresponding edit in urls.py, views.py;
9. Create a new application to manage user account in setting.py;
10. Define url.py, views.py, and HTML template for login, logout register page;
11. Apply base.html to the new application;

12. Apply Django decoration @login_required in views.py to restraint users from accessing another user's data;
13. Create and migrate database for all users;
14. Protect user's data by editing in views.py;

This process only shows a basic idea of how we are going to develop the to-do list application using the MVC framework, and any step in the list could be iterated and repeated to make the application functioning well.

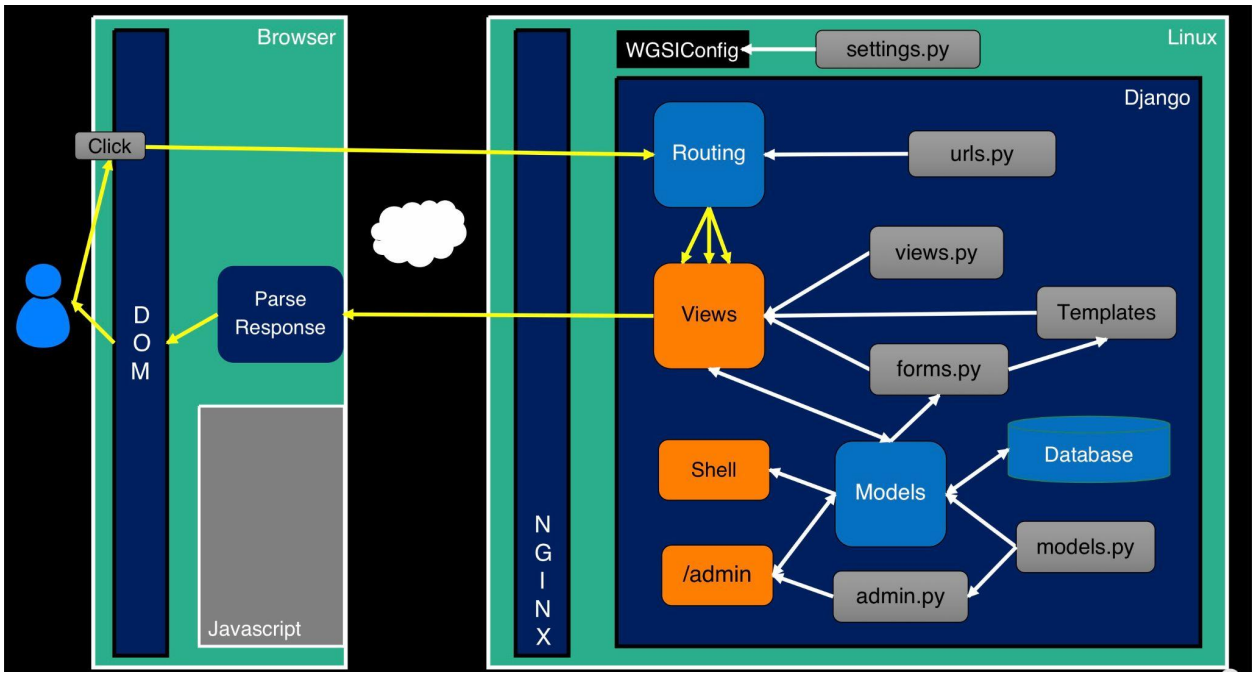

Figure 2. Workflow of back end [1]

2.3 Priority list of features

Table 1. Priority of features

Priority	Features
High Priority	**To do list (Add, Remove, Move task)**

	Project road map
	Assign & Request task form to team member
	Upload and store project files
Medium Priority	**Tagging team member**
	Project calendar, performance board
	Search bar
Low priority	**Team member board**
	Editing profile
	Notification of a request

3.0 Design

3.1 Wireframe of the first page

At the top of the first page, there is the P&NP logo and login and register buttons. At the center of the page, there is a company description and a login button.

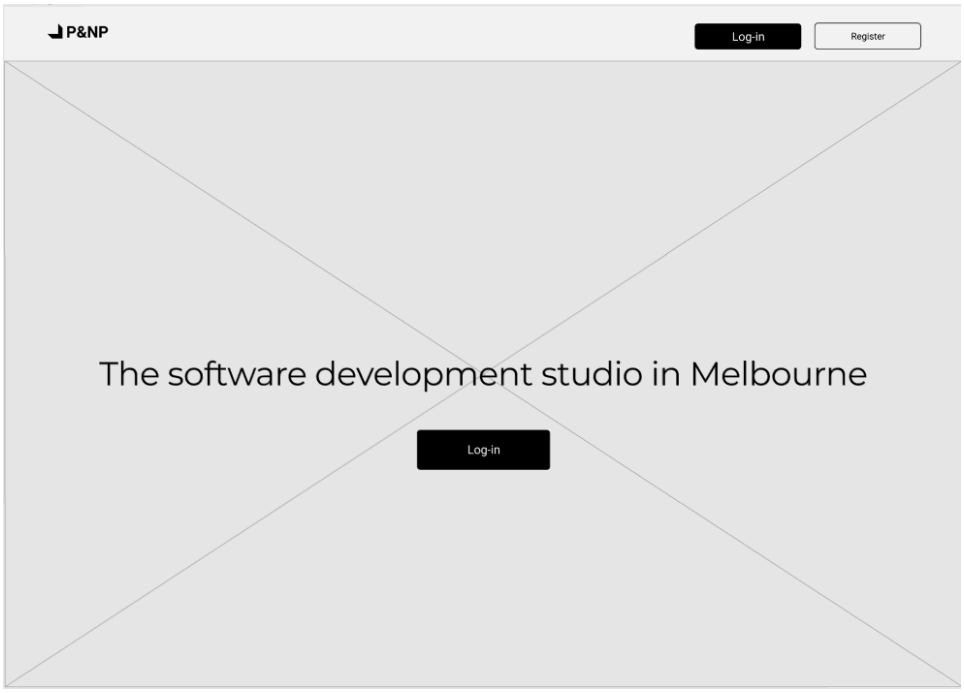

Figure 3. The first page of the design

3.2 Wireframe of the login page

When users click the login button, the form of entering the email and password is shown. After that, if it is a registered email and password, users can access the next page.

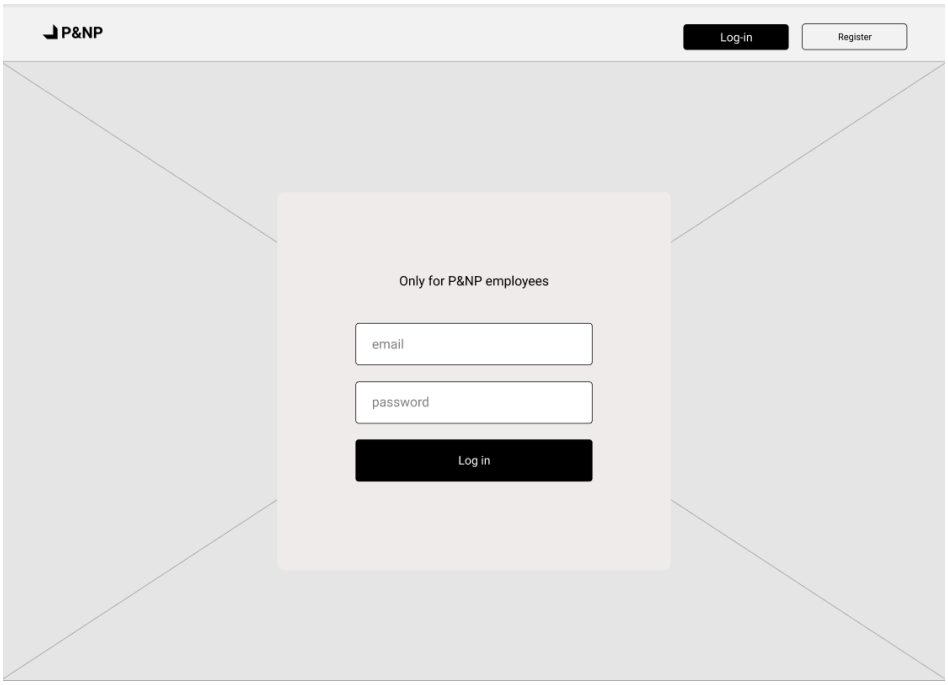

Figure 4. Login page of the design

3.3 Registration page

When the user clicks the Register button, the registration form page appears, and the user must insert the information, such as first name, last name, email, password, and position. After filling in all the information, the user will be able to register.

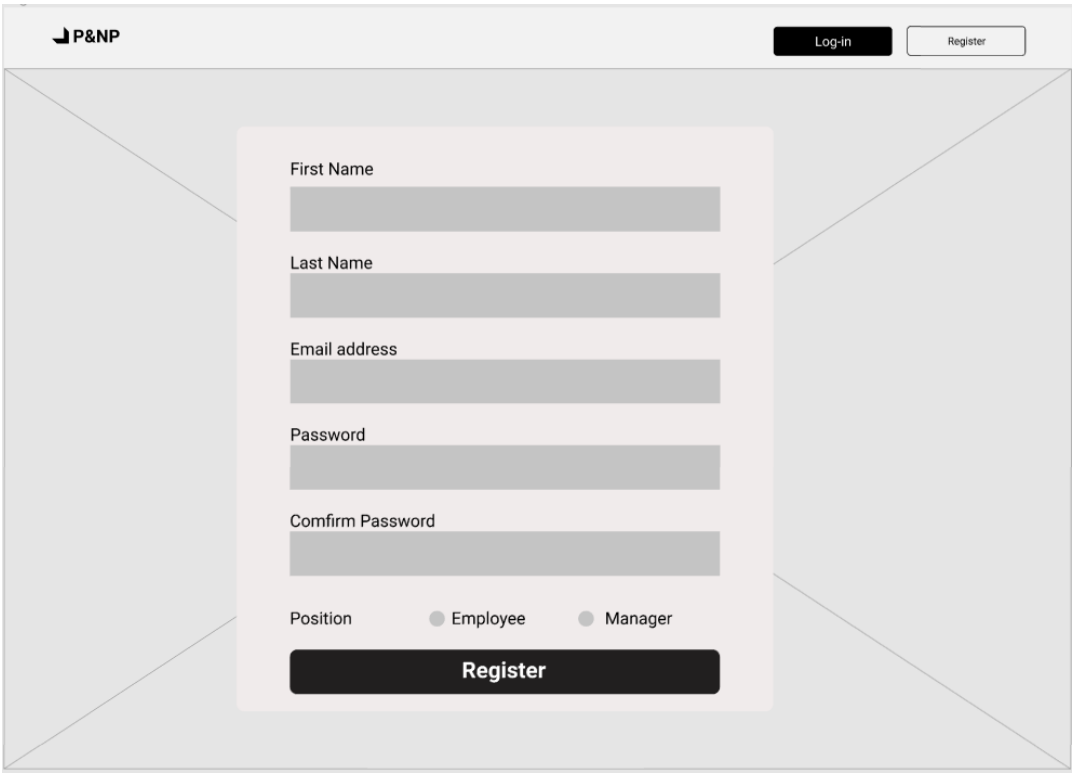

Figure 5. Registration page of the design

3.4 To-do list page

When successfully login, the user sees the to-do list page for the first time.

At the top of the page, there are the company logo, search bar, notification, and editing profile feature And there are menu and logout buttons on the left side of the page that is defaulted throughout the whole application.

The to-do list page is divided into three sections in a Kanban style. The ‘to-do list’, ‘In progress’ and ‘Done’ are shown on the left, center, and right respectively. On this page, the manager has the authority to add and remove the tasks.

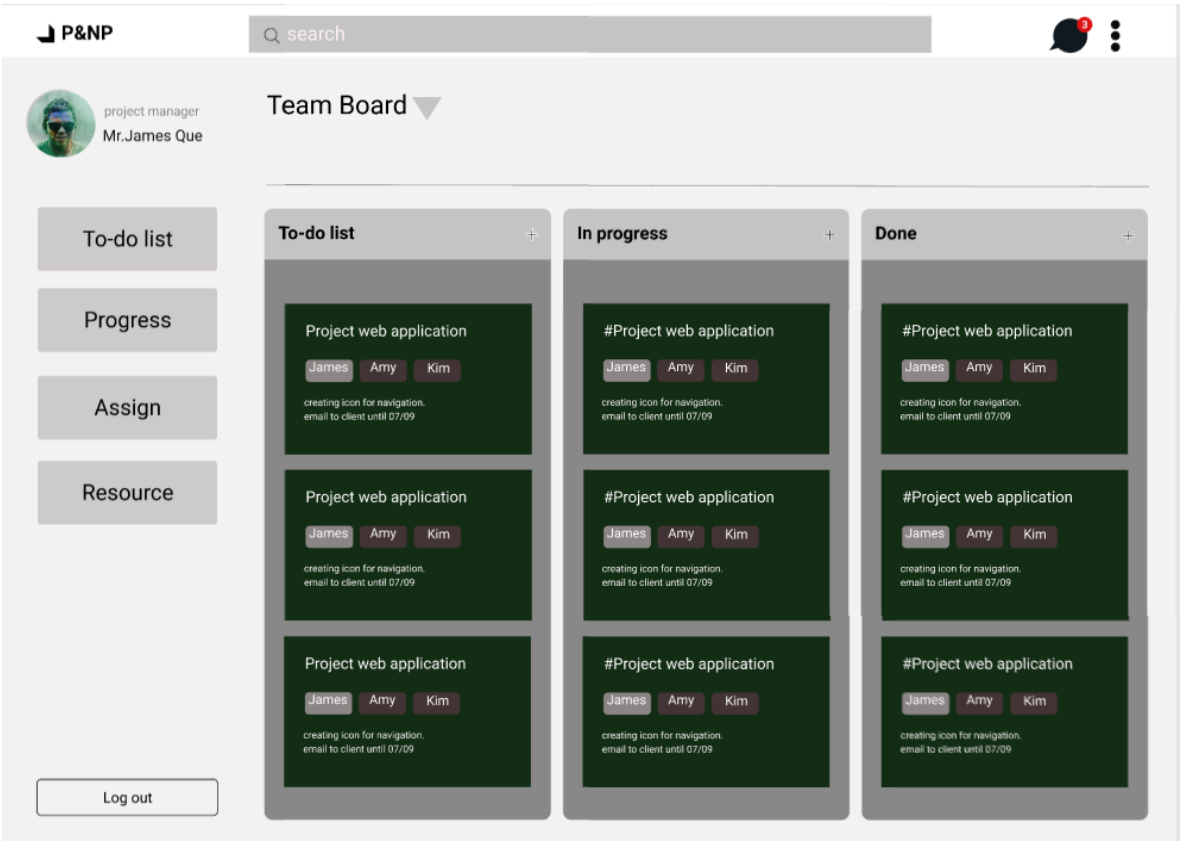

Figure 6. To-do list page of the design

3.5 Progress page

On this page, users can monitor the progress of each project with a calendar, project performance board, and roadmap. There is a team member board where users can check

the project members.

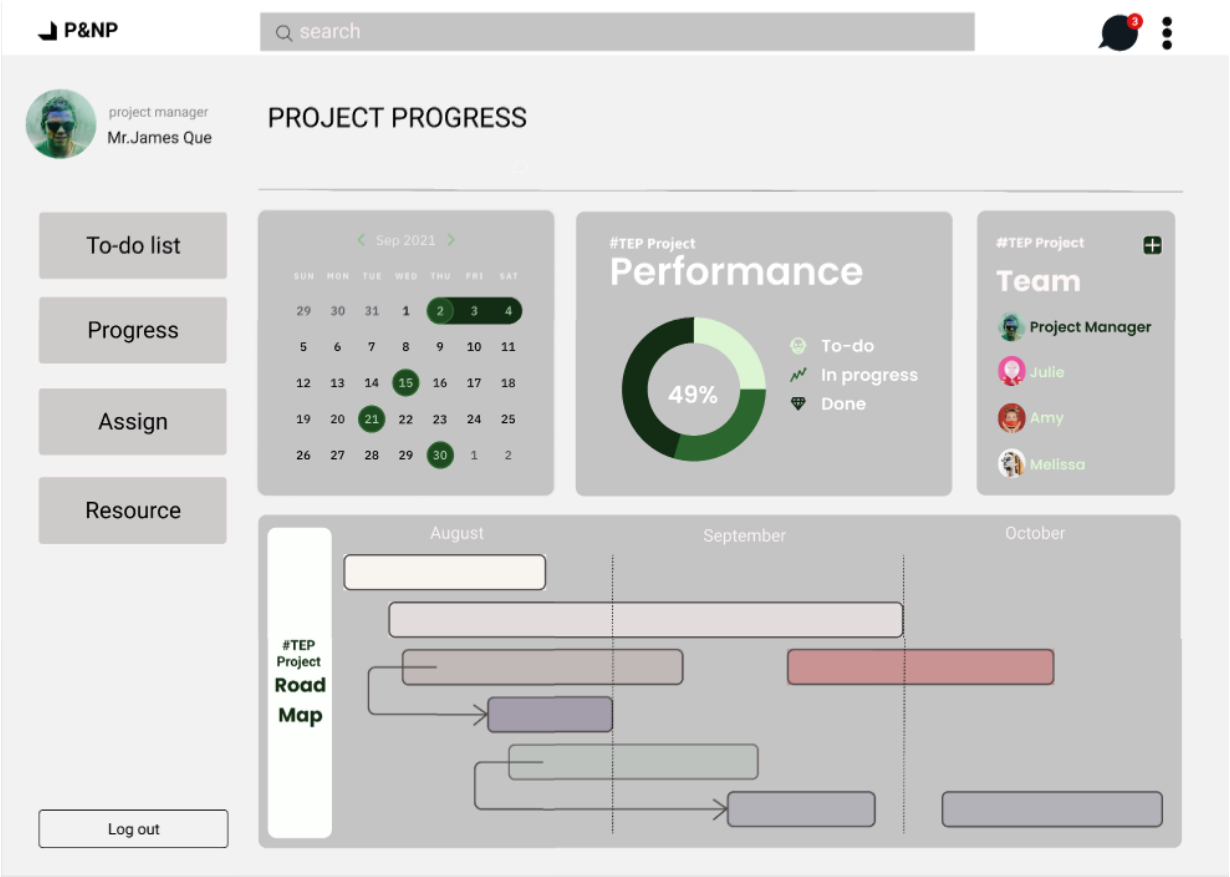

Figure 7. Progress page of the design

3.6 Assign and request task page

On the assigned page, users can request tasks to other team members using the form. Users are able to set the assignee, task priority, category, due date, and description of the task in the form. On the right side of the page, users can review the list of messages received.

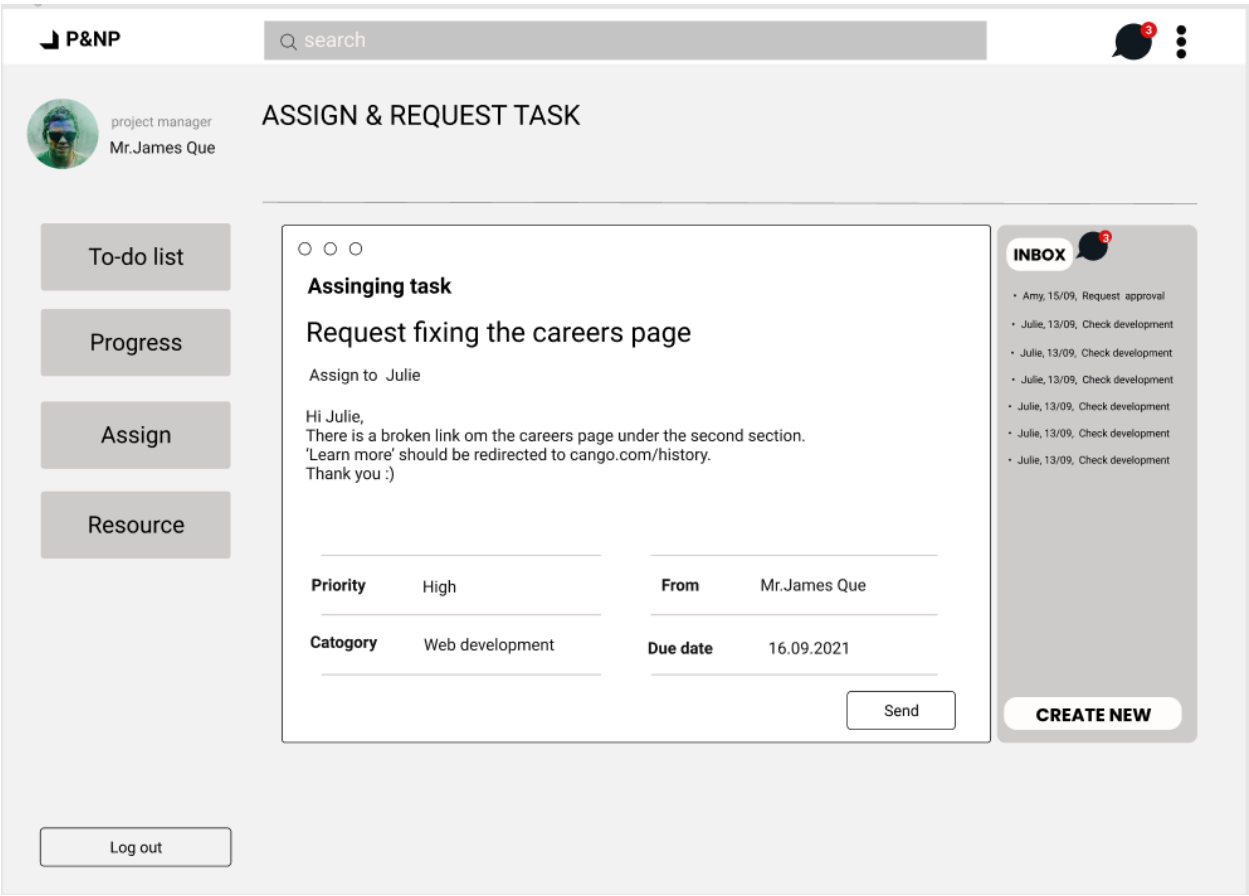

Figure 8. Assign and request task page of the design

3.7 Resource page

The resource page allows users to manage and upload project resources by dividing them for each task. When users upload new files, they must select a task folder.

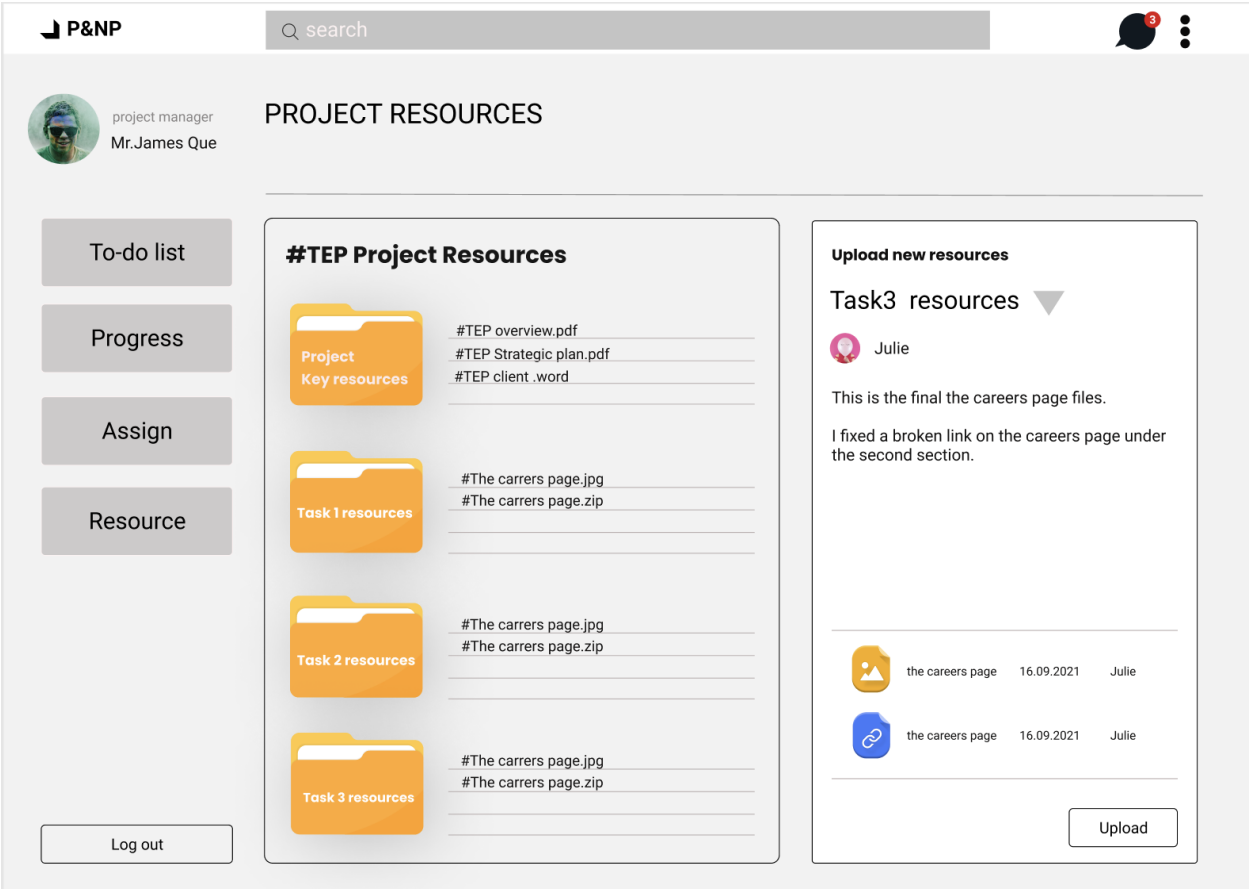

Figure 9. Resource page of the design

4.0 Timeline and Risk Control

4.1 Timeline

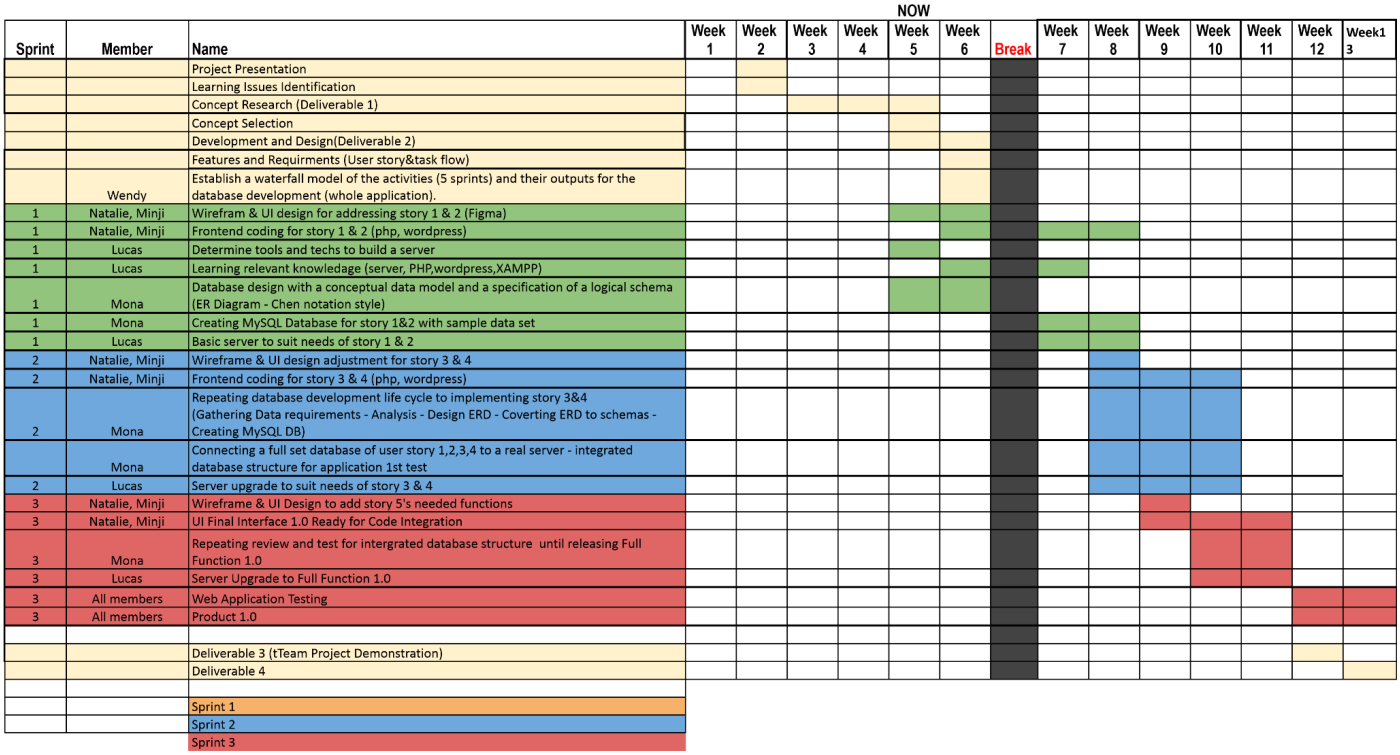

Figure 10. Gantt Chart of the project

4.2 Risk control

Table 2. Risks and solutions

Identify code	Risk rating	Risk Description	Downside	Solution
001	**low**	Taking more time than scheduled on one single task.	With one task lag behind schedule, other tasks' timelines might be compromised due to procrastination.	Keep track of the team progress by regular meetings, once identify possible delay signs on delivering the designated task, try to find out the root cause and seek advice on solving the issue.
002	**high**	Taking more time than scheduled on a sprint.	With one sprint lag behind schedule, other sprints' timelines	Evaluate the reasons that caused the delay on the project, and report to

			might be affected and it might lead to the whole project delay or failure.	stakeholders. Discuss future prevention strategy and finish the lesson learned document and split it to every team member.
003	**medium**	Lack of expertise in a specific technology.	There are many technologies required in building a web application, if one of the members lacks a certain level of knowledge in finishing a task on time, it could compromise a sprint's timeline.	Report project manager. Assign any team member with the required skills to help. Check knowledge management documents if any. Schedule time for learning or allocate resources for the task to minimize the risk.
004	**medium**	Team member conflict	There could be conflicts among team members and stakeholders if handled improperly, it could lead to mass delay on the project and have a bad impact on the team's motivation.	Follow communication protocol and get intervention asap.

*Risk rating definition: 1. Low: A risk that only affects one task, it will pose no threat to the sprint's schedule if handled properly; 2. Medium: A risk that affects more than 2 tasks and could compromise the sprint's schedule; 3. High: A risk that affects more than 3 tasks and could compromise the whole project's deliverable.

5.0 Summary

This article explained the web pages of the project we are going to build, and how we are going to build them. The proposed features to be applied in our web pages are discussed

in section 2.0, including the to-do list, progress page, assign task page, and resource page. The technology stacks and tools applied to implementing these features are presented as well. The proposed prototype interface design is shown in section 3.0. In addition, a detailed timeline of the whole design process and a detailed risk control plan are also discussed and presented visually in section 4.0.

Reference

[1] Severance, C. *Model View Controller (MVC)*. Charles Russell Severance, 2021.

[2] Elmasri, Ramez ; Navathe, Shamkant B. 2016. *Fundamentals of Database Systems* (Global Ed.). Pearson Education.