# Welcome to the BioLogic Project

## Project Overview

BioLogic is an educational tool designed to facilitate the teaching of scientific concepts to students. It enables students to answer questions by constructing answers using graphical Explanation Networks (ExNets), which consist of statements and logical connectors. This method enhances students' ability to express themselves clearly and precisely using grammatically correct English. Students can also validate the ExNets' meaning through an English language readout. A BioLogic Marking Assistant can expedite the grading process by identifying the similarity in a student's answer, which greatly reduces the workload on academics.

The key components of this project are the Editor.vue and Builder.vueflow tools. This project is an extension of a previous project where another team developed a BioLogic WebApp from a Python-based desktop application. The focus of semester 2's Project 1 is to improve and integrate these tools for use by students, tutors, and academics. The Editor.vue requires refinement, bug fixing, and feature additions, while the Builder.vueflow tool needs polishing and integration with the Editor.vue. The goal is to allow academics to create questions using Builder.vueflow, which can then be passed to Editor.vue for students to answer. Additionally, the Editor.vue is ideally capable of offering immediate feedback to students in a formative assessment mode. This feedback will be based on assessment information from the Builder.vueflow's rubric.

## Team Members

This project is carried out by a team from the University of Melbourne's 2023 Semester 2 COMP90082 Software Project course, consisting of five students and a supervisor.

| Team member name | Role | Email |
|---|---|---|
| Paul Calverley | Supervisor | paul.calverley@unimelb.edu.au |
| Yiyun Yang | Product Owner | yiyyang1@student.unimelb.edu.au |
| Steven Zhang | Scrum Master | stevenz1@student.unimelb.edu.au |
| Xingchen Han | Developer | xingchenh1@student.unimelb.edu.au |
| Jiuneng Zhang | Developer | jiunengz@student.unimelb.edu.au |
| Sihan Zhang | Developer | sihazhang1@student.unimelb.edu.au |

### Client

| Client name | Project | Email |
|---|---|---|
| Mike Murray | BioLogic | murraym@unimelb.edu.au |
| Ping | BioLogic | ping.charoenwet@unimelb.edu.au |

### Tools for Project Management

| Tool | Description | Links |
|---|---|---|
| Confluence | Document the details of the project and all created resources from Planning, Analysis and Design phases. | https://confluence.cis.unimelb.edu.au:8443/display/COMP900822023SM2BIBlueRing/Home |
| Trello | Visualise the project's progress and to estimate, prioritise and manage our tasks and sprints. | https://trello.com/b/b9bcSevn/bi-bluering |
| GitHub | Manage and modify the project's code and to generate releases that will be accessed by the clients. | https://github.com/COMP90082-2023-SM2/BI-Bluering |
| Slack | Instant messaging program used as the the primary communication channel. | |

### Recent space activity

**Yiyun YANG**

Welcome to the BioLogic Project updated less than a minute ago • view change

Requirements updated 9 minutes ago • view change

### Space contributors

- Yiyun YANG (less than a minute ago)
- Xingchen HAN (37 minutes ago)
- Steven Zhang (56 minutes ago)
- Sihan ZHANG (3 hours ago)
- Jiuneng Zhang (3 hours ago)
- ...

**Xingchen HAN**

Plan for next sprints updated 37 minutes ago • view change

**Steven Zhang**

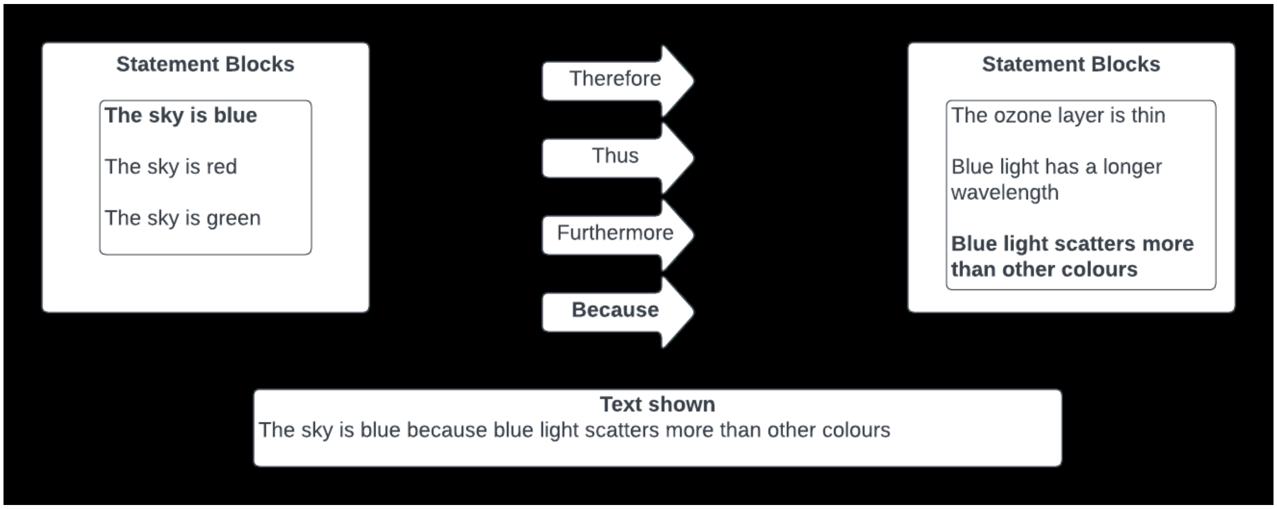
2023-08-11 Weekly supervisor meeting - Meeting notes updated about an hour ago • view change

Do/Be/Feel and Goal Models updated about an hour ago • view change

# Project Background

## Background Description

This project aims to further develop a web application that enhances student question answering and instructor assessment marking efficiency through the use of graphical Explanation Networks (ExNets). These networks consist of statement blocks and logical connectors, which can be combined to create coherent sentences. Statement blocks provide selectable statements for students' responses, while logical connectors, such as "furthermore" and "because", link these statements together. The below visual example helps illustrate the concept, highlighting multiple selectable statements within each block which are then connected by logical connectors. The bolded statements within each section specify the student's selections. As an example, a student might choose "The sky is blue" in the first block and "Because" as the logical connector, while the last block might contain "Blue light scatters more than other colours." This results in the sentence: "The sky is blue because blue light scatters more than other colours."


Statement Blocks
The sky is blue
The sky is red
The sky is green
Therefore
Thus
Furthermore
Because
Statement Blocks
The ozone layer is thin
Blue light has a longer wavelength
Blue light scatters more than other colours
Text shown
The sky is blue because blue light scatters more than other colours

Moreover, the application features an English language readout to help students validate their answers' meaning. This involves identifying the chosen statements and connecting words to ensure coherence. Once answers are selected, students can utilise a drag-and-drop mechanism to arrange these components within the user interface. This removes the need for manual typing which not only enhances response efficiency, but also ensures readability with no spelling or grammar errors. The final completed statement is displayed in a text box which allows students to check and make sense of their constructed answer.

The holistic goal of the BioLogic project is to enhance the teaching of scientific concepts by providing students with a structured and efficient way to express their understanding of the content. The integration of the BioLogic Marking Assistant streamlines the marking process for assessors. Ultimately, this project aims to enhance the learning experience for students, the teaching experience for tutors, and the efficiency of academic assessment.

## Motivation

After discussing with the client, the main motivation of this project is to focus on improving the web application's functionality by polishing certain aspects of the tool, debugging issues, and adding other helpful features. The client has confirmed the Builder is mostly finished and our team should focus on the Editor's features. Some of the main pain points of the application are bugs when dragging components, the collapsing of the overall statement is not smooth, students are running out of space in the user interface, and there's an oversall lack of user-friendliness within the tool. The Editor should also be able to talk to the server to contruct answers and questions, while also incorporating feedback from the Grader. There's also room for additional features in terms of accessibility such that issues with colour blindness and limited mobility, as well as including an option for a light and dark mode. Additionally, features which are nice-to-haves include having an undo/redo function, as well as sound feedback when students are interacting with the tool to provide some reinforcement of a "clicky" feeling. The Editor has two main usages which are in tutorial and exam settings. In tutorials, the immediate feedback to students is very helpful for their learning, particularly when the tutor-to-student ratio is not substantial. For exams, this would greatly help markers be more efficient in assessing students' answers. Near the end of this project, our team will strive to work together with Team RedBack to ensure the Grader communicates to the Editor and also provides changes of the rubric to the builder.

## Scope Outline

Ultimately, this project aims to achieve the below outlined goals:

1. Firstly, our highest priority is to improve the Web Editor (VueJS) so that it becomes a useable tool in tutorial settings. We aim to address graphical defects that interfere with the user experience, such as imprecise drag-and-drop of objects, and unexpected behaviours of elements when dropping them onto different areas. Ideally, Editor.vue would match the current Python version of the BioLogicEditor in terms of robustness, ease of use and features available.
2. Secondly, we aim to implement a new “Student Feedback” feature that can provide the student users with information about what is right and wrong in their current answer and display text-feedback provided by the tutor and academics. The actual feedback will be provided by the Grader. The Editor should be naïve about the rubric information that the Grader is using, such that the Grader might provide a “tick” or “mark” for a Statement and some text-feedback to display for a given Statement or Connector.

3. Thirdly, we will explore the integration options of the Web Editor to the backend server and the Grader. Furthermore, we will also review how to integrate the Editor into the university's Canvas system and provide guidelines for deploying the Editor in a big class of students.

## Team Roles

The development team is responsible for designing the web application, its implementation, testing the application, and ensuring quality assurance. The development team works closely with the product owner, scrum master, and client to ensure that the requirements and needs of the client's project are met. The Product Owner's primary focus is on maximising the value delivered by the development team by clearly defining requirements, maintaining the backlog, and clarifying the tasks needing to be completed by the developers. The product owner is responsible for creating and managing the timeline of the project, as well as tracking the progress of the development team and making necessary adjustments if the team faces any challenges. The Scrum Master focuses on helping the whole team continuously improve and optimise their processes by removing road blocks that hinder the team's progress. They also help the team work cooperatively in order to create a productive and collaborative environment, which also leads to effective communication between team members and stakeholders. Both the Product Owner and Scrum Master work closely together to ensure the successful delivery of the project.

# Do/Be/Feel and Goal Models

## Do/Be/Feel

**Student**

DO:

- Click, drag and drop answers' components onto the application's interface which is user friendly
- Edit and rearrange the statements' components without any issues
- Read and review their final answers
- Spend less time on formatting answers and more time on understanding the concepts

BE:

- More more efficient in completing questions for tutorials and exams
- Less stressed and overwhelmed with ensuring correct spelling and grammar
- More engaged and interested in the concepts being taught

FEEL:

- Relieved to utilise a tool which makes answering questions easier, as well as receiving immediate feedback
- Empowered to take on more challenging questions and learning more complex concepts

**Tutor**

DO:

- Upload the answer key and student work to the application's interface
- Quickly provide students with feedback
- Use the tool to teach concepts to a larger number of students in a shorter period of time

BE:

- More efficient in providing immediate feedback to students
- More capable in providing targeted feedback to help students improve

FEEL:

- Empowered to provide more timely and detailed feedback to students
- Satisfied to see students improve and succeed with the help of the tool

**Marker**

DO:

- Review and verify the automated marking results
- Quickly provide students with feedback and grades

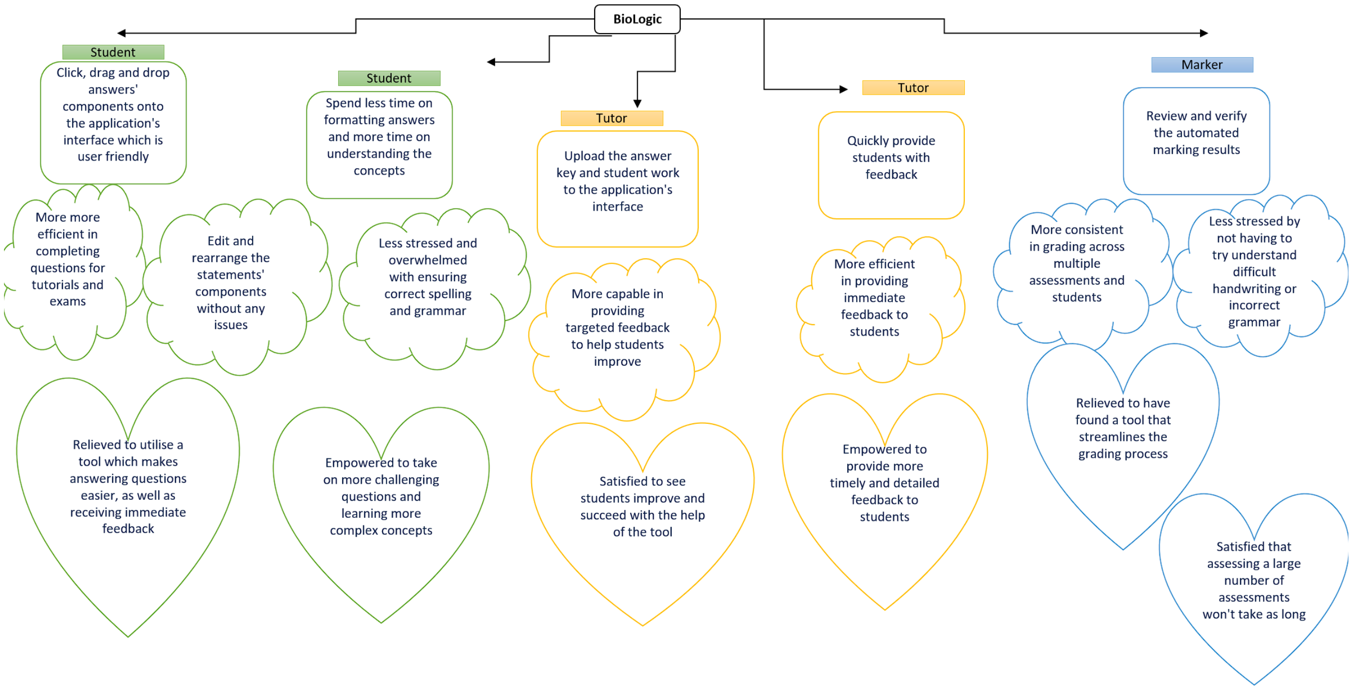
BE:

- More efficient in grading and providing feedback to students
- More consistent in grading across multiple assessments and students
- Less stressed by not having to try understand difficult handwriting or incorrect grammar

FEEL:

- Relieved to have found a tool that streamlines the grading process
- Satisfied that assessing a large number of assessments won't take as long

## Goal Model



BioLogic

Student

Click, drag and drop answers' components onto the application's interface which is user friendly

More more efficient in completing questions for tutorials and exams

Edit and rearrange the statements' components without any issues

Relieved to utilise a tool which makes answering questions easier, as well as receiving immediate feedback

Student

Spend less time on formatting answers and more time on understanding the concepts

Less stressed and overwhelmed with ensuring correct spelling and grammar

Empowered to take on more challenging questions and learning more complex concepts

Tutor

Upload the answer key and student work to the application's interface

More capable in providing targeted feedback to help students improve

Satisfied to see students improve and succeed with the help of the tool

Tutor

Quickly provide students with feedback

More efficient in providing immediate feedback to students

Empowered to provide more timely and detailed feedback to students

Marker

Review and verify the automated marking results

More consistent in grading across multiple assessments and students

Less stressed by not having to try understand difficult handwriting or incorrect grammar

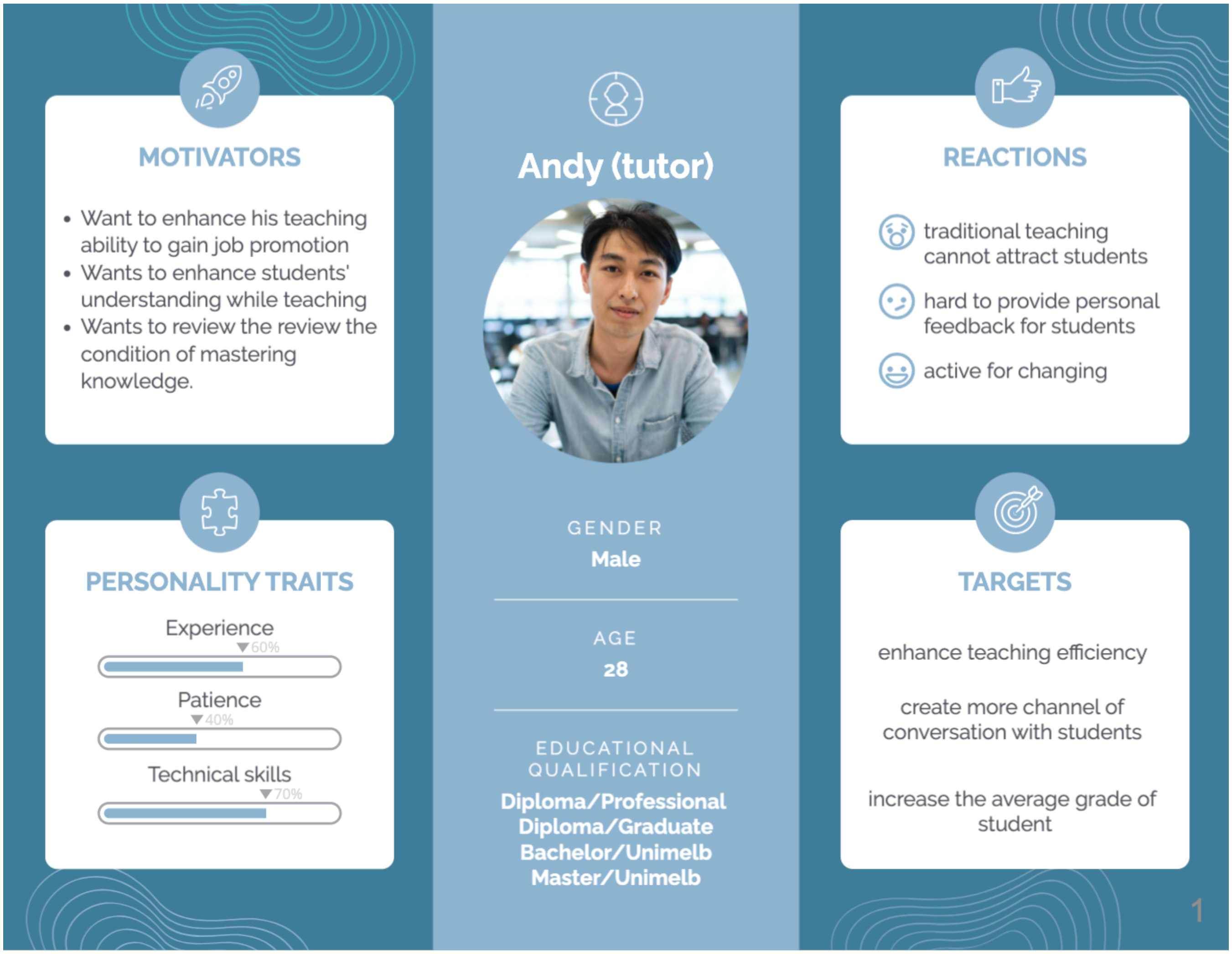
Relieved to have found a tool that streamlines the grading process

Satisfied that assessing a large number of assessments won't take as long

# Personas

We considered potential customers in three perspectives, students, tutors and markers , they have different aspects to view the new programs, we made three personas to ensure we could have a better understanding for user portraits.

Andy, who is the PhD student of Melbourne University and also works as a tutor in school, he is willing to work in Unimelb for future.



## Andy (tutor)

GENDER
**Male**

AGE
**28**

EDUCATIONAL QUALIFICATION
**Diploma/Professional**
**Diploma/Graduate**
**Bachelor/Unimelb**
**Master/Unimelb**

### MOTIVATORS

- Want to enhance his teaching ability to gain job promotion
- Wants to enhance students' understanding while teaching
- Wants to review the review the condition of mastering knowledge.

### PERSONALITY TRAITS

Experience ▼60%

Patience ▼40%

Technical skills ▼70%

### REACTIONS

traditional teaching cannot attract students

hard to provide personal feedback for students

active for changing

### TARGETS

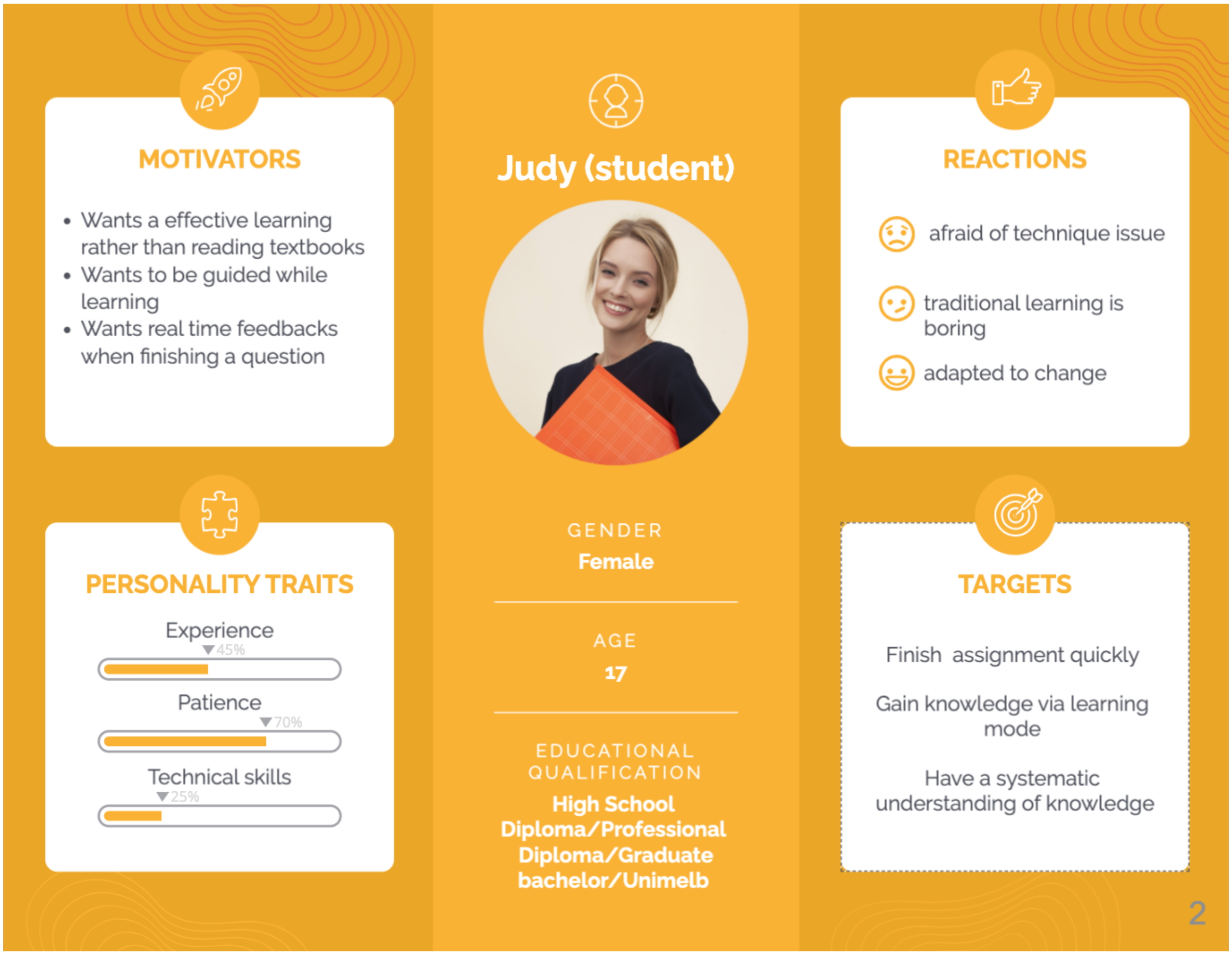
enhance teaching efficiency

create more channel of conversation with students

increase the average grade of student

1

Judy, she is the second year student of unimelb, she is willing to get H1 so that she could apply scholarship.



## MOTIVATORS

- Wants a effective learning rather than reading textbooks
- Wants to be guided while learning
- Wants real time feedbacks when finishing a question

## PERSONALITY TRAITS

Experience ▼45%

Patience ▼70%

Technical skills ▼25%

## Judy (student)

GENDER

**Female**

AGE

**17**

EDUCATIONAL QUALIFICATION

**High School Diploma/Professional Diploma/Graduate bachelor/Unimelb**

## REACTIONS

- afraid of technique issue
- traditional learning is boring
- adapted to change

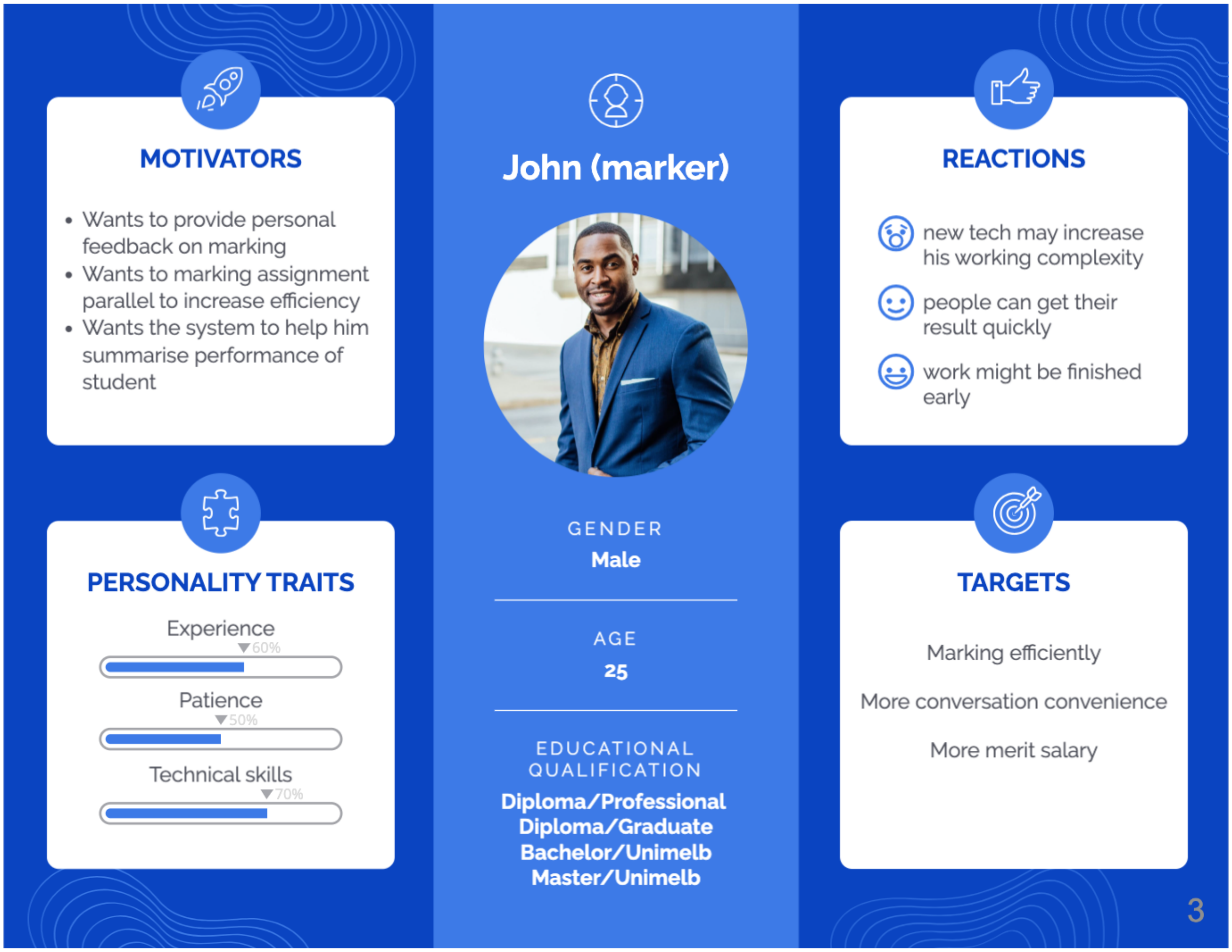
## TARGETS

Finish  assignment quickly

Gain knowledge via learning mode

Have a systematic understanding of knowledge

John, he is in charge of marking students work and giving feedback, also he just had a baby so that he is willing to earn more for his family



## John (marker)

GENDER
**Male**

AGE
**25**

EDUCATIONAL QUALIFICATION
**Diploma/Professional**
**Diploma/Graduate**
**Bachelor/Unimelb**
**Master/Unimelb**

### MOTIVATORS

- Wants to provide personal feedback on marking
- Wants to marking assignment parallel to increase efficiency
- Wants the system to help him summarise performance of student

### PERSONALITY TRAITS

| | |
|---|---|
| Experience | 60% |
| Patience | 50% |
| Technical skills | 70% |

### REACTIONS

😲 new tech may increase his working complexity

🙂 people can get their result quickly

😀 work might be finished early

### TARGETS

Marking efficiently

More conversation convenience

More merit salary

# Requirements

**Summary of Requirements**

This project aims to enhance the Web Editor (built with VueJS) to serve as an effective tool within tutorial settings. The primary objective is to align its performance, user-friendliness, and capabilities with the current Python version of the BioLogicEditor. This entails rectifying graphical anomalies that impede usability, such as addressing issues with imprecise drag-and-drop actions and unexpected behaviors of elements upon placement. Furthermore, the Web Editor should incorporate selected features from the Python version, such as the ability to present Student statements using radio-button formatting.

Additionally, a new "Student Feedback" feature should be implemented. This feature will furnish student users with real-time feedback on the correctness of their responses, along with textual guidance from the teacher. It's important to note that the actual feedback content will be provided by the Grader. The Editor itself should remain unaware of the rubric details employed by the Grader. For example, the Grader might provide positive and negative indicators for a Statement, as well as corresponding feedback text.

Furthermore, the project requires exploration of integration options for linking the web editor with the backend server and the grading system. The integration of the editor into a canvas system also needs to be considered, along with the development of deployment guidelines to facilitate its implementation in large student classes.

**In scope features**

1. Web Editor Enhancement for Tutorial Settings:

- Improving the Web Editor's usability to match the BioLogicEditor's Python version in terms of robustness, ease of use, and features.
- Addressing graphical defects that hinder usability, such as refining drag-and-drop functionality and rectifying unexpected behaviors of elements.

2. Feature Incorporation from Python Version:

- Implementing selected features from the Python version, such as presenting Student statements using a radio-button format.

3. Student Feedback Feature:

- Developing a "Student Feedback" feature to provide real-time feedback to students about the correctness of their answers supplied by the Grader.
- Ensuring the Web Editor remains unaware of specific rubric details used by the Grader.

4. Integration with Backend Server and Grader:

- Exploring integration options for connecting the Web Editor with the backend server and the grading system.

5. Canvas System Integration:

- Evaluating the integration of the Web Editor into a canvas system, ensuring compatibility and functionality.

6. Deployment Guidelines:

- Providing guidelines for deploying the Web Editor in large student classes, assisting in successful implementation.

**Out-of-Scope Features**

1. Complete Feature Parity with Python Version:

- Achieving complete feature parity with the BioLogicEditor's Python version may be beyond the scope of this project.

2. Advanced Grading Algorithm Development:

- Developing complex grading algorithms that go beyond basic "tick" and "mark" indicators may not be included in this project.

3. Integration with Third-Party Tools:

- Integrating the Web Editor with specific third-party tools or platforms beyond the backend server, grader, and canvas system is not within the current scope.

4. User Authentication and Security:

- Addressing user authentication, authorization, and security measures for the Web Editor may not be covered in this project.

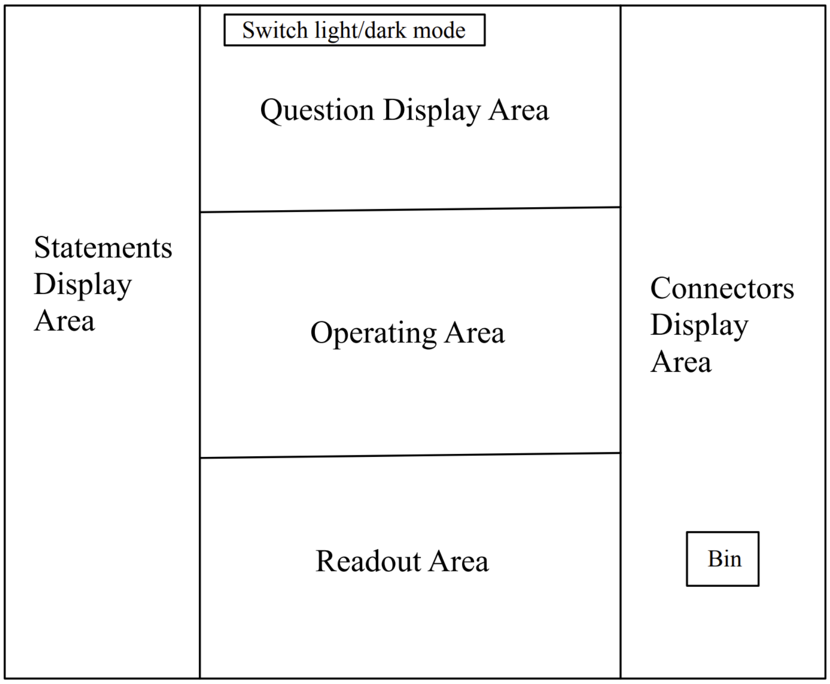
5. Extensive User Interface Overhaul:

- Making extensive changes to the user interface design or structure of the Web Editor may be considered out of scope.

6. Scalability Testing:

- Conducting extensive scalability testing for large-scale deployment might be considered out of scope for this project.

**Prototypes**



**User Stories**

| Epic | User Story ID | User Stories | MosCow<br>Priority | Size Estimation |
|---|---|---|---|---|
| Improve the Web Editor's usability | 1.1 | As a student, I want to have precise drag and drop of objects on Editor so that I can drag and drop elements to different areas. | Must Have | 4 |
| | 1.2 | As a student, I want to categorize different types of statements while suffering from color blindness so that I can build the answer correctly. | Should Have | 4 |
| | 1.3 | As a student, I want to switch between light mode and dark mode automatically and manually so that it is more comfortable and easier on the eyes. | Should Have | 3 |
| Align functionality of the Python version | 2.1 | As a student, I want to be able to view the selected statement in a radio-button format so that I could have a clear view of my answer. | Must Have | 4 |
| Display the Student Feedback | 3.1 | As a teacher, I want to display text-feedback on Editor so that the student can know what is right and wrong in their current answer. | Must Have | 4 |
| | 3.2 | As a student, I want to get feedback on answers while Tutorial setting so that I can know what’s right and what’s wrong. | Must Have | 3 |
| Integrate the editor into the backend server | 4.1 | As a teacher, I want to deploy the editor in a big class of students so that I can collect the answers within the class. | Must Have | 3 |
| | 4.2 | As a student, I want to get the same question from the Canvas system so that I can complete the tutorial/exam successfully. | Must Have | 3 |
| Product deployment | 5.1 | As a student/teacher, I want to receive comprehensive guidelines covering installation, configuration, and best practices. | Must Have | 4 |

* Some of the 'In scope features' and 'Out-of-scope features' are generated and developed from ChatGPT.

# Development Environment

**Introduction**

Welcome! This document provides instructions for setting up the project from scratch. It covers operating system and platform requirements, development tools and IDEs, programming languages and frameworks, as well as step-by-step instructions for setting up Vue and running the development server.

**Prerequisites:**

- **Operating System and Platform:** Supported platforms include Windows (Windows 10 and above), macOS (latest version), and Linux distributions (such as Ubuntu, Fedora, etc.).
- **Development Tools:** It is recommended to use an IDE or text editor suitable for Vue.js development, with Visual Studio Code being the recommended choice.
- **Web Browser Requirements:** Since Vue.js applications run in web browsers, make sure to list major supported web browsers such as Google Chrome, Microsoft Edge, etc.
- **Node.js Version:** Familiarity with the command-line interface for Node.js, with a version of 18 or higher.
- **Package Manager:** Ensure npm version 9 or higher. The contents of the GitHub repository will be utilized on your device.

**Project Collaboration Platforms:**

- **Confluence**: Create, share, and collaborate on project documents, design documents, requirements specifications, meeting notes, and more.

  https://confluence.cis.unimelb.edu.au:8443/display/COMP900822023SM2BIBlueRing/Home

- **Trello**: Create, manage, and track project tasks, user stories, issues, backlogs, and more.

  https://trello.com/b/b9bcSevn/bi-bluering

- **GitHub**: Used to host project code, collaborate on development, conduct code reviews and version control.

  https://github.com/COMP90082-2023-SM2/BI-Bluering

Please adhere to the specified prerequisites and project collaboration platform setup for proper execution.

**Setting up Vue and running the development server**

Here are the steps for installing Vue:

1. **Install Node.js:**js relies on the Node.js runtime. If you haven't done so already, download and install the appropriate version of Node.js for your operating system from the official Node.js website.

2. **Open the Command Terminal:** On Windows, you can use Command Prompt or PowerShell. On macOS and Linux, use the Terminal.

3. **Navigate to the Project Folder and Install Dependencies:** Follow these steps to navigate to your project folder and install the necessary packages.

a. Use the following command to navigate to the project folder:

```
cd C:\Your\Project\Here
```

Replace the path with your actual project path.

b. Install the required packages.

```
npm install
```

c. Verify successful installation:

```
npm fund
```

This command will confirm the installation status of required packages.

4. **Start the Development Server:** Execute the following command in the project folder to launch the development server:

```
npm run serve
```

will start a local development server, allowing you to view your Vue application in a browser.

5. Open the project folder with an editor and edit the Vue components and application logic in the src directory.

# Meeting notes

Create meeting note

## Incomplete tasks from meetings

### Task report

Looking good, no incomplete tasks.

## All meeting notes

| Title | Creator | Modified |
|---|---|---|
| 2023-08-11 Weekly supervisor meeting - Meeting notes | Steven Zhang | about an hour ago |
| 2023-08-04 Weekly supervisor meeting - Meeting notes | Steven Zhang | about 2 hours ago |
| 2023-08-10 First client meeting with combined teams - Meeting notes | Steven Zhang | about 2 hours ago |
| 2023-08-13 Team meeting - Meeting notes | Steven Zhang | about 2 hours ago |
| 2023-08-17 Team meeting - Meeting notes | Steven Zhang | about 2 hours ago |

# 2023-08-04 Weekly supervisor meeting - Meeting notes

## Date and Time

5:15pm Friday 04 Aug 2023 Week 2 of Semester

## Attendees

- Paul Calverley (Supervisor)
- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang

## Goals

- First meeting with project supervisor Paul Calverley
- For the team members and supervisor to meet each other for the first time over Zoom.
- To understand the basics of the project and next steps required for the project.

# Discussion items

| Notes |
|---|
| Not much documentation available at the moment |
| Need to plan first meeting with him, continue on slack for now, end of week now, probably no response from him until monday |
| Client to send us a recording of the software and how it works |
| Previous semester group worked on this project, 2nd topic is more developing, 1st is more polishing and debugging, |
| • Confluence workspace - no one has used before<br>https://confluence.cis.unimelb.edu.au:8443/display/COMP900822023SM2BIBlueRing/Home |
| Checklist provided by Lucy: need to get requirements from client to build up documentation for first sprint |

## Action items

☑ Paul will let the team know once the client responds on his availability.

☑ Meet with client next week, think about availability for next week

☑ Michael will be invited to the Slack workspace by Paul

# 2023-08-10 First client meeting with combined teams - Meeting notes

## Date and Time

1pm Thursday 10 Aug 2023 Week 3 of Semester

## Attendees

- Paul Calverley (Supervisor)
- Mike Murray (Client)
- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang
- Team RedBack

## Goals

- Introductions of the team members and confirmation that team BlueRing will work on Project 1 and team RedBack on Project 2
- Clarify the project scope and objectives, intended outcomes, and expectations: Which aspects of the Editor.vue and Builder.vueflow require polishing? What features need to be added to Editor.vuer? Are we required to integrate all three tools as part of this project?
- How do we get access to the code repositories and instructions for how to set up our development environment?
- Our team members have been invited to join the GitLab project but we're unable to accept the invitation as the student email in GitLab is different from the invitation email. Can invitations be resent to our other student email addresses?
- When will the next meeting be, and what are the expectations for it? What are our main action points before the next meeting?

## Discussion items

Meeting with Client

Talk about the roadmap of the project

We need to achieve a reasonable output and run

Builder is fairly finished, should focus on the editor features,

Bugs when you're dragging, better collapsing of the overall statement

Students are running out of space,

There is an undo/redo function, not a huge deal,

Need to be able to talk to the server, construct an answer/question from the server,

Make the editor user friendly

Incorporate feedback from the grader side of things

Two ways to use the editor, exam and tutorial, feedback in tutorial is very helpful,

Does python give audible sounds? Does web version have sounds? Students like the clicky feeling. Gives a bit of reinforcement, not a big deal. Open to what's a good UX/UI or not

accessibility : color blindness or limited mobility, change from light mode to dark mode,

The functionality is more important than aesthetics, it would take Michael too long to fix all the bugs,

Grab the Vuecode and put it in the grader shouldn't be too hard

Both teams will need to work together to make sure the grader communicates to the editor and provides changes of the rubric to the builder.

Michael just wants to run this in a web browser

# 2023-08-11 Weekly supervisor meeting - Meeting notes

## Date

5:15pm Friday 11 Aug 2023 Week 3 of Semester

## Attendees

- Paul Calverley (Supervisor)
- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Steven Zhang

## Discussion items

Designing user stories, write down the requirements,
Making prototype,
UI/UX
Deliverables
Inception stage:
Personas, user stories, requirements, UI design,
First run the program to identify the bugs,

"As a user of this app"

**Zoom Chat**

17:19:41 From Paul Calverley To Everyone:

Tasks

1. Define requirements and write user stories
2. UI design
3. Document the roles (SM and PO)
4. Learning frameworks
5. Personas
6. Dev environment setup
7. Plan for sprint 2
8. Design Confluence home page
9. Document the project background, goal and objectives, scope

Effort
8. 2 effort
9. 4 effort
3. 1 effort
10. Run the app to find bugs
6. 2 effort
10. 5 effort

Assigning responsibility

# 2023-08-13 Team meeting - Meeting notes

## Date and Time

9pm Sunday 13 Aug 2023 Week 3 of Semester

## Attendees

- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang

## Discussion items

Design Confluence home page 2hrs - Steven

1. Document the project background, goal and objectives, scope - 4hrs Steven

Document the roles (SM and PO) 1hr Steven

1. Dev environment setup - 2hrs Carol - Everyone
2. Run the app to find bugs Sihan - Everyone

Define requirements and write user stories - Jodie + Carol

UI design - Jodie + Carol

1. Personas - Stark
2. Plan for sprint 2 - Stark

Meeting minutes + notes - Steven

Learning frameworks - Everyone

Another team meeting on Thursday 17th Aug

# 2023-08-17 Team meeting - Meeting notes

## Date and Time

5pm Thursday 17 Aug 2023 Week 4 of Semester

## Attendees

- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang

## Discussion items

Carol to reach out to Ping to confirm login details for the program

Sihan to make page on confluence for development environment

Setting deadline for 12pm tmr to complete release tag

Clarify steps required for sprint 1 submission

# Plan for next sprints

18 AUG

**Sprint 1**
**Inspection & Design**

**Sprint 1**
**Inspection & Design**

- Background - Steven
  - background description ✓ - 2 hours
  - motivation ✓ - 1 hour
  - scope outline ✓ - 1 hour
- Personas - Xingchen
  - motivators ✓ - 2 hours
  - personality traits ✓ - 2 hours
  - reactions ✓ - 2 hours
  - targets ✓ - 2 hours
- User stories - Steven&Yiyun
  - epic identification ✓ - 2 hours
  - stories prioritization ✓ - 2 hours
- Development - Everyone
  - app testing ✓ - 2 hours
  - bugs finding ✓ - 2 hours
- Plan for sprint 2 - Xingchen
  - sprint identification ✓ - 2 hours
  - section identifcation ✓ - 2 hours

22 SEPT

**Sprint 2**
**Development**

- Design document - TBA
  - system review - 1.5 hours
  - requirement adjustment - 1 hour
  - UI adjustment - 0.5 hour
- Design documentation - TBA
  - design specification - 2 hours
  - target identification - 1 hour
- Product baklog -TBA
  - elements identification - 2 hours
  - tasks allocation - 2 hours
- Sprint review - TBA
  - client meeting - 0.5 hour
  - bugs testing - 0.5 hour
- Product demo -TBA
  - presentation - 0.5 hour
- Cyber security review - TBA
  - vulnerabilities identify - 2 hours
  - security evaluation - 2 hours

20 OCT

**Sprint 3**
**Development**

- Design document - TBA
  - system review - 1.5 hours
  - requirement adjustment - 1 hour
  - UI adjustment - 0.5 hour
- Design documentation - TBA
  - design specification - 2 hours
  - target identification - 1 hour
- Product baklog -TBA
  - elements identification - 2 hours
  - tasks allocation - 2 hours
- Sprint review - TBA
  - client meeting - 0.5 hour
  - bugs testing - 0.5 hour
- Product demo -TBA
  - presentation - 0.5 hour
- Cyber security review - TBA
  - vulnerabilities identify - 2 hours
  - security evaluation - 2 hours

4 NOV

**Sprint 4**
**Handover**

- Handover document - TBA
  - user manual handover - 1 hour
  - diagrams handover - 1 hour
  - UI adjustment - 0.5 hour
- Handover code - TBA
  - user acceptance test - 2 hours
  - performace test - 3 hours
  - code handover - 0.5 hour
- Final product demo -TBA
  - project exhibition - 2 hours