

1. Welcome to the BioLogic Project	2
1.1 Project Background	4
1.1.1 Do/Be/Feel and Goal Models	6
1.1.2 Personas	8
1.2 Requirements	11
1.2.1 User Stories	12
1.2.2 In-scope & Out-of-scope features	13
1.2.3 Prototypes	15
1.3 Task Tracking	16
1.3.1 Task Size Estimations	17
1.4 Cyber Security	18
1.5 Development Environment	20
1.6 Ethical Considerations	23
1.7 Sprint 2	25
1.7.1 Sprint 2 Planning - Meeting notes	26
1.7.2 Plan for next sprints	28
1.7.3 Code Review for Sprint 2	29
1.7.4 Sprint 2 Demonstration	30
1.7.5 Sprint 2 Review - Meeting notes	31
1.8 Sprint 3	32
1.8.1 Sprint 3 Planning - Meeting notes	33
1.8.2 Plan for Sprint 3	34
1.8.3 Code Review for Sprint 3	35
1.8.4 Sprint 3 Demonstration	36
1.8.5 Sprint 3 Review - Meeting notes	37
1.8.6 Sprint 3 Retrospective - Meeting notes	38
1.8.7 Final Presentation	39
1.9 Testing	40
1.10 Meeting Notes	41
1.10.1 Team Meetings	42
1.10.1.1 2023-08-13 Team meeting - Meeting notes	43
1.10.1.2 2023-08-17 Team meeting - Meeting notes	44
1.10.1.3 2023-08-23 Team meeting - Meeting notes	45
1.10.1.4 2023-08-28 Team meeting - Meeting notes	46
1.10.1.5 2023-10-12 Team meeting - Meeting notes	47
1.10.1.6 2023-10-13 Team meeting - Meeting notes	48
1.10.1.7 2023-10-16 Team meeting - Meeting notes	49
1.10.1.8 2023-10-19 Team meeting - Meeting notes	50
1.10.1.9 2023-10-20 Team meeting - Meeting notes	51
1.10.2 Supervisor Meetings	52
1.10.2.1 2023-08-04 Weekly supervisor meeting - Meeting notes	53
1.10.2.2 2023-08-11 Weekly supervisor meeting - Meeting notes	54
1.10.2.3 2023-08-18 Weekly supervisor meeting - Meeting notes	55
1.10.2.4 2023-08-25 Weekly supervisor meeting - Meeting notes	56
1.10.2.5 2023-09-01 Weekly supervisor meeting - Meeting notes	57
1.10.2.6 2023-09-08 Weekly supervisor meeting - Meeting notes	58
1.10.2.7 2023-09-15 Weekly supervisor meeting - Meeting notes	59
1.10.2.8 2023-09-22 Weekly supervisor meeting - Meeting notes	60
1.10.2.9 2023-10-06 Weekly supervisor meeting - Meeting notes	61
1.10.2.10 2023-10-13 Weekly supervisor meeting - Meeting notes	62
1.10.3 Client and Team RedBack Meetings	63
1.10.3.1 2023-08-10 First client meeting with combined teams - Meeting notes	64
1.10.3.2 2023-09-28 Meeting with Team RedBack - Meeting notes	65

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 Charoenwet	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-bluring
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-Bluring
Slack	Instant messaging program used as the the primary communication channel.	

Recent space activity



Steven Zhang

[Sprint 3 Demonstration](#) updated less than a minute ago
• [view change](#)

[Code Review for Sprint 3](#) updated 2 minutes ago • [view change](#)

Space contributors

- [Steven Zhang](#) (less than a minute ago)
- [SihaN ZHANG](#) (42 minutes ago)
- [Yiyun YANG](#) (3 hours ago)
- [Jiuneng Zhang](#) (6 hours ago)
- [Xingchen HAN](#) (18 hours ago)
- ...

Testing updated 2 minutes ago • [view change](#)

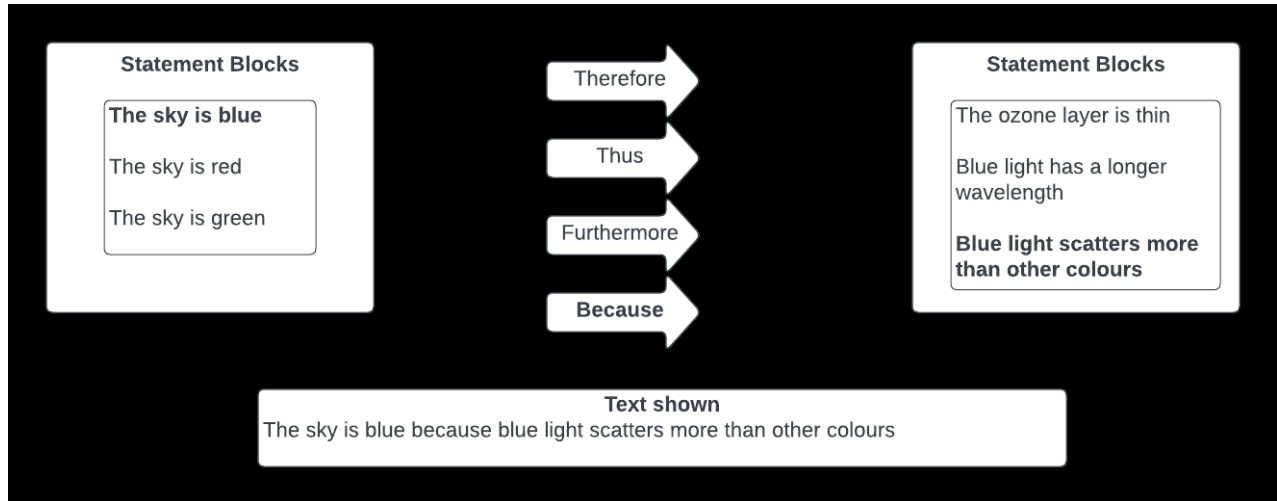
Ethical Considerations updated 2 minutes ago • [view change](#)

Cyber Security updated 2 minutes 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."



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 construct 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
- Create intellectually stimulating questions to challenge students appropriately

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

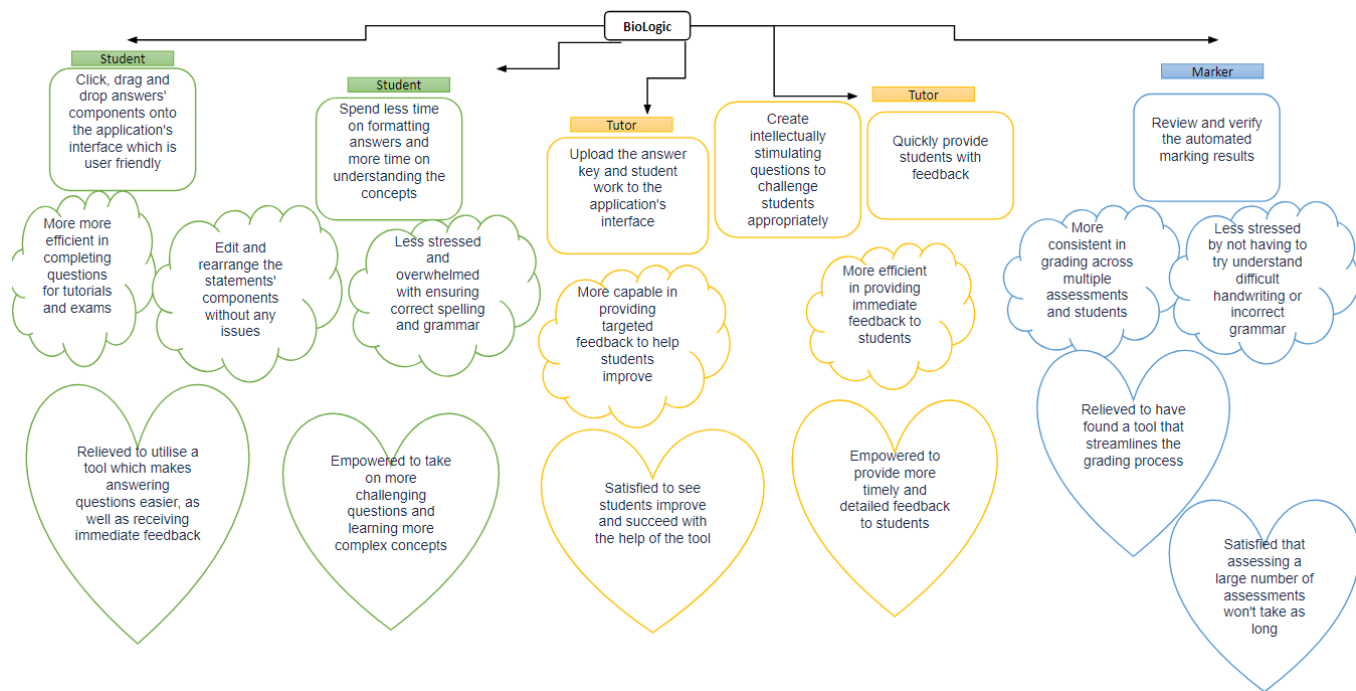
BE:

- 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



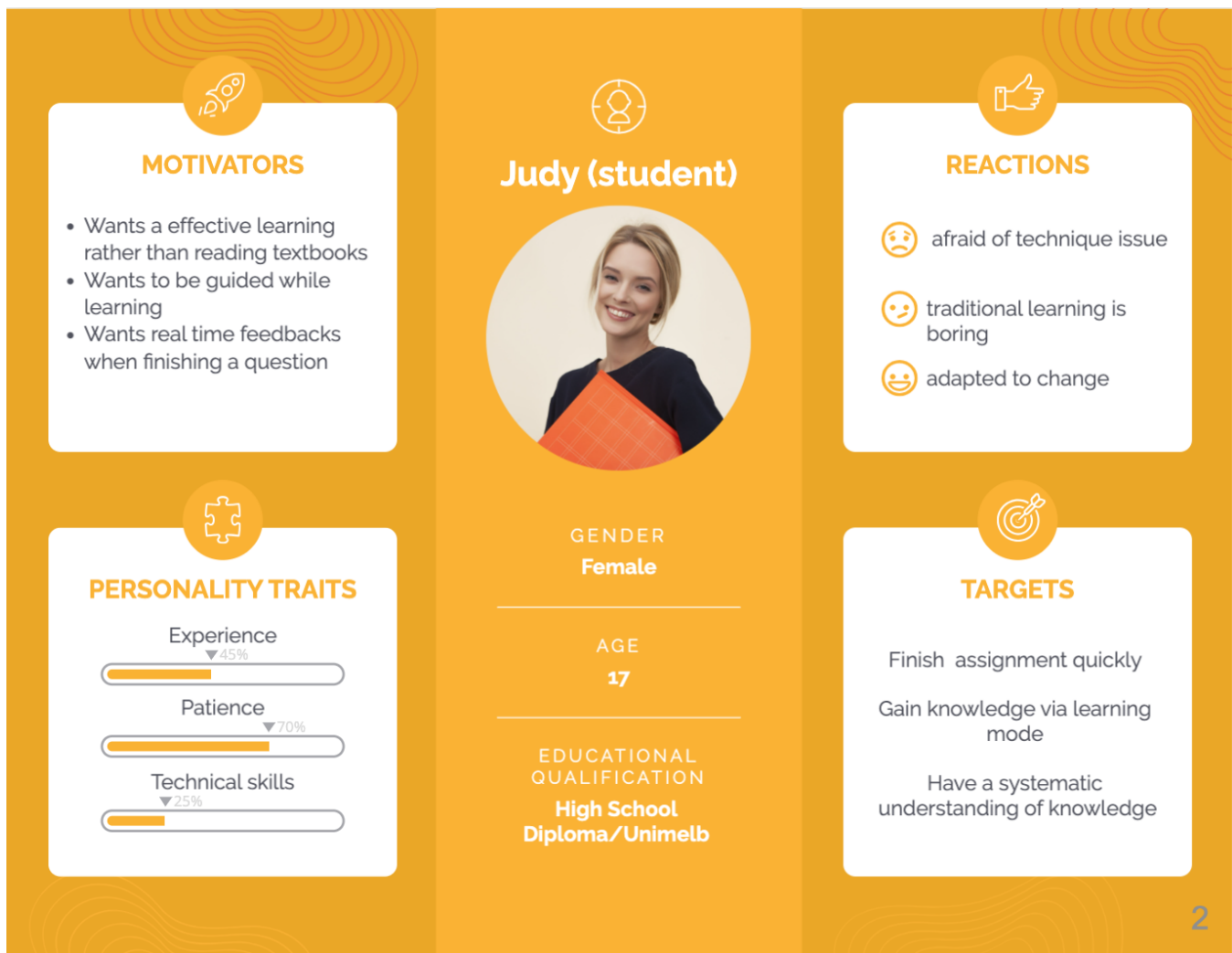
Personas

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.

(Students and teachers are major users of the system, and there are two categories of teachers which showing concern in different aspects , tutors more caring about the teaching use of the system, which is the effectiveness of knowledge representation from system. markers are teachers who using system to mark answers from student and they shows more caring on the abilities of answer marking and feedback giving rather than knowledge teaching using the system)

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.





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



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



John (marker)



GENDER
Male

AGE
25

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



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 and one-sentence 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.

User Stories



Epic	User Story ID	User Stories	MosC ow Priority	Size Estimation
Improve the Web Editor's usability	1.1	As a student, I want to have precise drag of objects on Editor so that I can precisely drag elements to different areas.	Should Have	Small (1-2 days)
	1.2	As a student, I want to have the ability to drop connector from list of connectors onto existed statements on Answer Area on Editor so that I can merge statements with connectors immediately.	Should Have	Medium (3-4days)
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	Medium (3-4days)
	2.2	As a student, I want to be able to view the selected statement in a one sentence format so that I could have a clear view of my answer.	Must Have	Medium (3-4days)
	2.3	As a student, I want to have a switch button in the statement box so that click the button and switch the format of the statement.	Must Have	Small (1-2 days)
Display the Student Feedback	3.1	As a student, I want to click the check button for my answer so that I can receive feed back to improve my answers.	Must Have	Small (1-2 days)
	3.2	As a student, I want to view immediate text-feedback besides my answer on Editor so that I can know what is right and wrong in the current answer.	Must Have	Large(6-12days)
Product deployment	4.1	As a student/teacher, I want to receive comprehensive guidelines covering installation, configuration, and best practices.	Must Have	Small (1-2 days)

In-scope & Out-of-scope features



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.

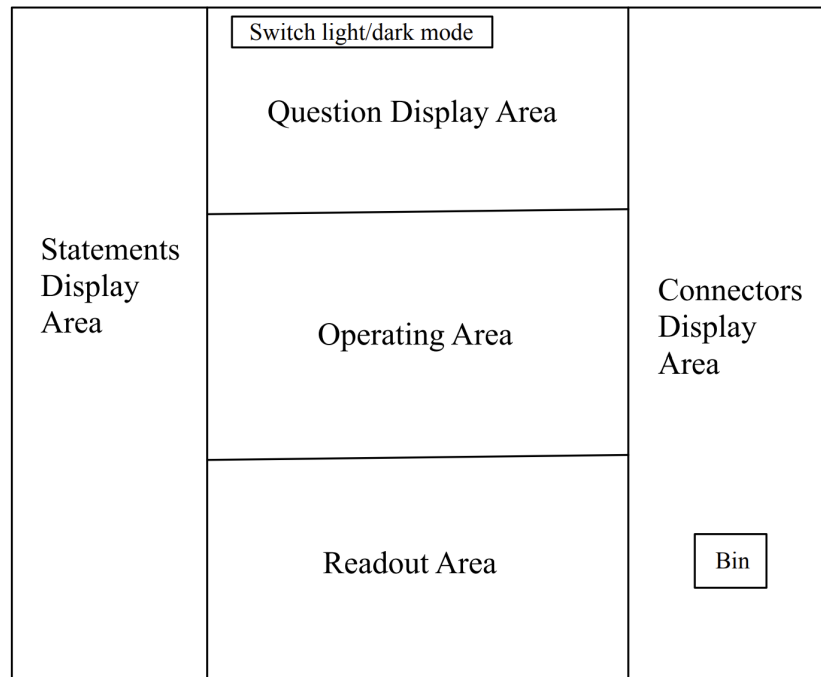


Out-of-Scope Features

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.

*'Out-of-scope features' are generated and reformed from ChatGPT.

Prototypes



Task Tracking

Product Backlog

Team's Trello board can be found at: <https://trello.com/b/b9bcSevn/bi-bluering>



Trello Documentation

This Trello board is designed to help our team manage our project tasks and keep track of our progress. The board is divided into three distinct sections, which are illustrated by the different coloured columns: **blue** for **Backlog**, **red** for **Doing & Testing**, and **green** for **Done**.

The Backlog column with a **blue** background contains all tasks in the user stories that are yet to be completed.

Each task contains a description when expanded which includes more details of the task. Moreover, we have tags which specify who worked on each task, which sprint it's related to, if it's a "must have" or a "should have", its due date, and size estimation. The different sizes are also colour-coded.

For more detailed information about each task, you can also refer to the corresponding user story on Confluence.

Some of the user stories are broken down into subtasks which are indicated by the numbers after 'US' (user story) such as US 2.1.

The **Doing**, **Testing** and **Done** columns are designed to help team members keep track of their progress.

The **Doing** column lists all the tasks that need to be completed in the current sprint, as well as the tasks that were not completed in the previous sprint. These tasks are decided during each sprint's planning session. Team members are allocated tasks from the **Backlog**, add their name to it and move it to the **Doing** column when they start working on it.

The **Doing** column elicits the current status of each member's assigned tasks. Some tasks also has a checklist or Acceptance Criteria shown under the description, which is used by developers to better comprehend the task and check their work before moving the task to the **Testing** column.

The **Testing** column is used to ensure that the task is on the right track and to let the team know that the code can be reviewed and tested. Once a developer finishes a task, they can move it to the **Testing** column so that a tester can review their work.

After an item has passed testing, it may be moved into the **Done** column.

Task Size Estimations



In the process for determining our task size estimations, we had our development team work together to decide on the best method to estimate the size of our broken down user stories.

After their discussion and with the confirmation of their decision with the product owner and scrum master, the size of the tasks were broken down into three main categories as per below:

Task Type	Size Estimation
Small	1-2 days
Medium	3-4 days
Large	5-6 days

Our team felt this breakdown of sizes created a clear distinction between the various task sizes while also matching the effort required for our user stories. Due to the difficulty of estimating the tasks down to the approximate hour, we found days to be more effective. One day in this case would be considered someone spending a regular full day of normal working hours on the task.

Cyber Security



Cybersecurity is often a comprehensive issue, and it is crucial to incorporate cybersecurity considerations into projects that span multiple layers throughout the application (both front-end and back-end). This application would likely contain sensitive information from students and faculty for this project and would likely be deployed within the university's Canvas system. The network security issues we may need to consider mainly cover the following aspects to ensure the confidentiality, integrity, and availability of data:

Authentication and access control:

Risks: Unauthorized access can lead to data leakage and misuse of sensitive information.

Mitigation: Implement strong authentication mechanisms. Using multi-factor authentication (MFA) is an effective way to increase security. Review and update your password policy regularly. Use session management best practices to ensure only authorized users have access to sensitive areas.

Data encryption:

Risk: Unencrypted sensitive data can be intercepted and used maliciously.

Mitigation: Data is encrypted at rest and in transit using industry-standard encryption algorithms such as AES and RSA. Avoid storing sensitive data on the front end and ensure it is encrypted if necessary. Use HTTPS to ensure data transmission is secure.

Data Privacy and Protection:

Risk: Data transmitted through unencrypted channels may be intercepted, compromising data privacy.

Mitigation: Ensure data is only transferred over HTTPS. Encrypt sensitive data during transmission to ensure your application complies with relevant data protection regulations such as GDPR or FERPA.

Input validation and defensive programming:

Risks: Insufficient validation can lead to SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) attacks.

Mitigation: Use input validation and output encoding to prevent these attacks. Implement a web application firewall (WAF) to filter and block malicious traffic.

Vulnerability Management:

Risk: Outdated dependencies may contain known vulnerabilities that can be exploited.

Mitigation: Regularly update dependencies and perform vulnerability scans. Implement a strong patch management process. Regularly monitor security bulletins and promptly address known vulnerabilities.

Monitor and respond:

Risk: Failure to monitor systems can result in delayed detection of security incidents.

Mitigation: Establish a security monitoring system to detect abnormal behavior and potential attacks in real time. Develop emergency response plans to minimize the impact of security incidents. Train your team regularly to effectively respond to security incidents.

The BioLogic tool utilises student answer data and questions data from academics, however the content is not highly sensitive as it just contains course material and de-identified student answers. However as this project progresses and later on when our team may work together with Team RedBack who works on the backend systems, we may need to be more careful in considering possibilities of data protection and access control.

Cooperation with Team RedBack:

As projects integrate with back-end systems, tighter data protection and access controls are required.

Need to plan how to mitigate the risk of cybersecurity issues

URL verification and verification: Ensures that the user's request targets a trusted URL and not a malicious site. It is a common best practice to use the HTTPS protocol to encrypt data transmission.

Input validation: When constructing params objects, ensure that management performs appropriate validation and sanitization of user-supplied data to prevent malicious input. Consider using an input validation library such as Joi or Yup.

Authentication: Information such as `student_id`, `client_id`, and `persistent_secret_key` are being used to ensure that this sensitive data is properly protected during transmission and storage. In particular, in a production environment, do not hardcode this information in client code.

Cross-site request forgery (CSRF) protection: To protect against CSRF attacks, administrators can include a CSRF token in the request and verify the validity of the token on the server side.

Error handling: In the try-catch block, the exception is caught, but more details are needed to better understand and log the error. Connect error messages with a backend logger to troubleshoot problems when they occur.

Key management: If keys are used in your code, ensure that key storage and transmission are secure. Keys should not be stored in the clear in code, but should be stored securely in environment variables or key management tools.

These cyber security measures will help protect the program from potential cyber threats and vulnerabilities, ensuring that student data and privacy are appropriately protected.

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 Development Platform:

- **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 docker and initiate client username

We need to have docker and client username set up before running the the server and frontend.

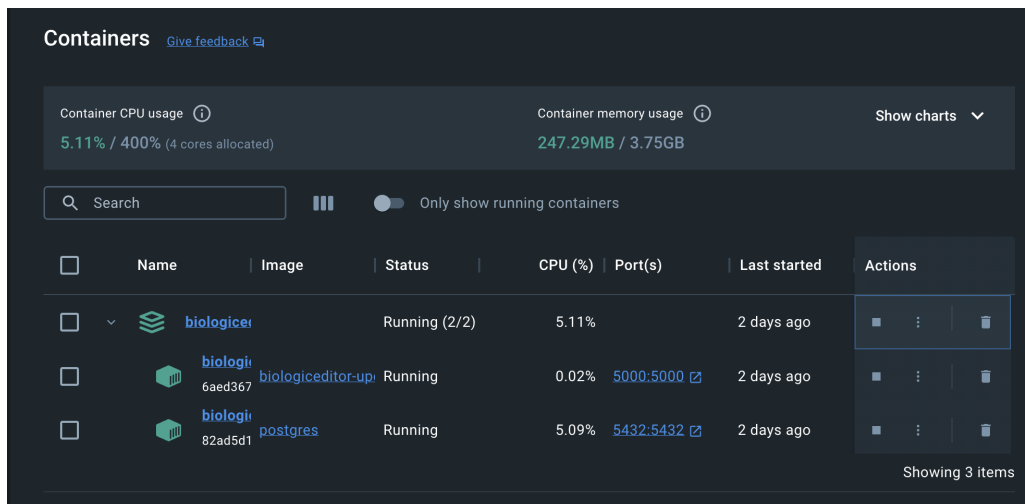
1. Set up docker

Make sure to install docker software and keep it open before running the command.

To run the server with **Docker**, use following commands in virtual environment.

```
bash ./start-server.sh
```

We should be able to see the containers in the docker after successfully installing.



2. Create admin

Run the following command:

```
docker exec biologic-web flask --app BiologicServer database create-admin --username="admin_username" --password="admin_password"
```

then we can open a page in the browser and enter localhost:5000/admin and login with admin_username and admin_password.

3. Pre-initiate client

We can find a file called `hashed_seed_client.txt` in the `BiologicDataService` folder, and we need to add a new file called `seed_client.txt` in the same folder

In the file `seed_client.txt`, we can input some dummy username such as `dummy_1` **before** running `pre_initiate` command.

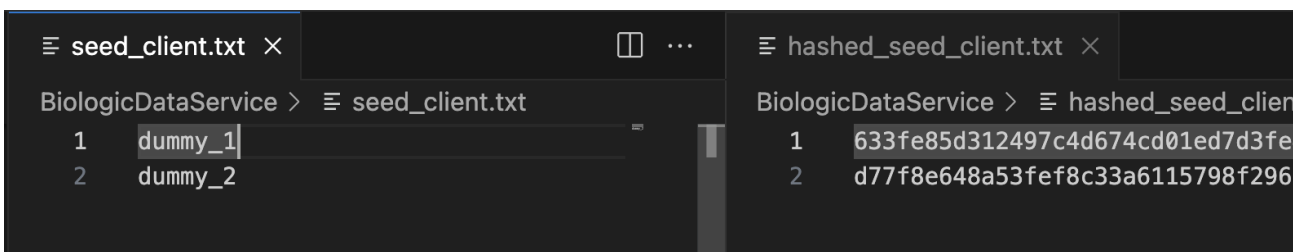
Run the following two commands:

```
flask --app BiologicServer database hash-seed-client
```

(if flask doesn't work, it could be replaced by 'docker exec biologic-web flask')

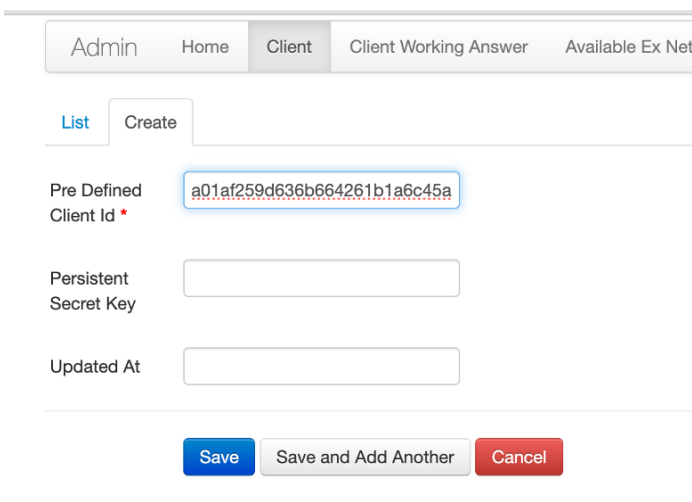
```
docker exec biologic-web flask --app BiologicServer database pre-initiate
```

The ID will be added to the backend after running the `pre-initiate` command as shown below, and we can login with the client username `dummy_1`.



4. Backup way

If login still fails after running step 4, we can manually add **hashed client** in the database in the link <http://localhost:5000/admin/client/>



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 dev
```

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.

Ethical Considerations



There are many aspects of ethical consideration which need to be considered during the development process:

Ethical Considerations and related possible assumptions for the BioLogic Educational Tool Project

1. Privacy and Data Protection:

- As the BioLogic tool will collect students' responses and possibly personal information, it is vital to ensure that this data is stored securely and in compliance with data protection regulations. Access to student data should be limited only to authorized personnel.
- Consider the implications of data breaches. Steps should be taken to ensure encrypted storage and secure transmission of data.
- **Secure Data Storage:** With the rise of cyber threats, it's imperative that data, especially sensitive student information, is stored securely. This might entail using advanced encryption methods, regular security audits, and implementing best practices for cybersecurity.
- **Data Collection Minimization:** Only collect data that's essential. Reducing the volume of personal data stored can minimize risks associated with potential breaches.

2. Transparency in Grading:

- The BioLogic Marking Assistant's methodology for grading and identifying similarity in students' answers should be transparent. This will help in ensuring that students and academics have faith in the system's fairness.
- If there's any machine learning or algorithmic decision-making involved in grading, steps should be taken to ensure the algorithm's decisions can be explained and understood.
- **Open Algorithms:** If grading is algorithm-based, the underlying principles of that algorithm should be transparent. This ensures that students and educators know how grades are determined and can trust the system.
- **Human Review:** Encourage educators to review grades, especially in edge cases or where there are discrepancies, to ensure the algorithm functions as intended.

3. Bias and Fairness:

- Care should be taken to ensure the BioLogic tool does not inadvertently introduce or perpetuate biases. For example, the way it interprets language should not disadvantage any group of students based on linguistic or cultural backgrounds.
- **Diverse Training Data:** If machine learning is involved, ensure the training data is diverse and representative. This can prevent unintentional biases from creeping in.
- **Continuous Monitoring:** Regularly monitor and test the system's outcomes to identify and rectify any inadvertent biases.

4. Accessibility:

- The tool should be accessible to all students, including those with disabilities. This may involve ensuring compatibility with screen readers, providing alternative input methods, or offering different modes of interaction.
- **Universal Design:** The design should consider the wide array of students' needs. This includes ensuring the tool is usable for people with varying physical and cognitive abilities.
- **Testing:** Regular testing with diverse user groups can help identify and address accessibility issues.

5. Dependence on Technology:

- Over-reliance on the tool for grading could lead to an erosion of human judgment in educational assessment. It's important to ensure that there's a balance, and human oversight remains a part of the grading process.
- **Human-in-the-loop:** Encourage a system where the tool aids human judgment rather than replacing it. This can be achieved by having periodic reviews by educators even if the tool is automated.
- **Training:** Educators should be trained on how to use the tool and also how to rely on their own judgment when necessary.

6. Feedback Accuracy:

- Given that the tool aims to offer immediate feedback, it is crucial to ensure this feedback is accurate and constructive. Misleading feedback could hamper a student's learning process.
- **Iterative Development:** Regularly update the feedback mechanisms based on actual student outcomes and educator reviews. This ensures feedback remains relevant and accurate.
- **Student-Educator Communication:** Encourage open lines of communication between students and educators. If a student feels the feedback is inaccurate, they should feel empowered to discuss it with their educator.

7. Intellectual Property:

- As this project is an extension of a previous one, ensure that there are clear understandings and agreements about the ownership and rights related to the software and any content produced.
- **Clear Licensing:** Clearly define the rights, responsibilities, and ownership of the software. If there are third-party components, ensure licenses are in order and there's no infringement.
- **Content Creation Rights:** If the tool aids in creating content, clarify who owns the rights to that content: the student, the institution, or the software creator.

8. Informed Consent:

- Before deploying the tool in a real-world classroom setting, both students and educators should be informed about how the tool works, its intended benefits, and any potential risks. They should provide their consent to use the tool, understanding these aspects fully.

- **Educational Sessions:** Host sessions where students and educators can learn about the tool, its capabilities, and limitations.
- **Transparent Documentation:** Offer detailed documentation, FAQs, or guidelines that can be referred to at any time. This helps users make an informed decision about using the tool.

By addressing these ethical considerations, the BioLogic project can uphold the highest standards of integrity and fairness, ensuring that the tool benefits students, educators, and the broader educational community.

Sprint 2

This folder contains all the relevant Sprint 2 artefacts.

Sprint 2 Planning - Meeting notes

Date

5:45pm Friday 25 Aug 2023 Week 5 of Semester

Attendees

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

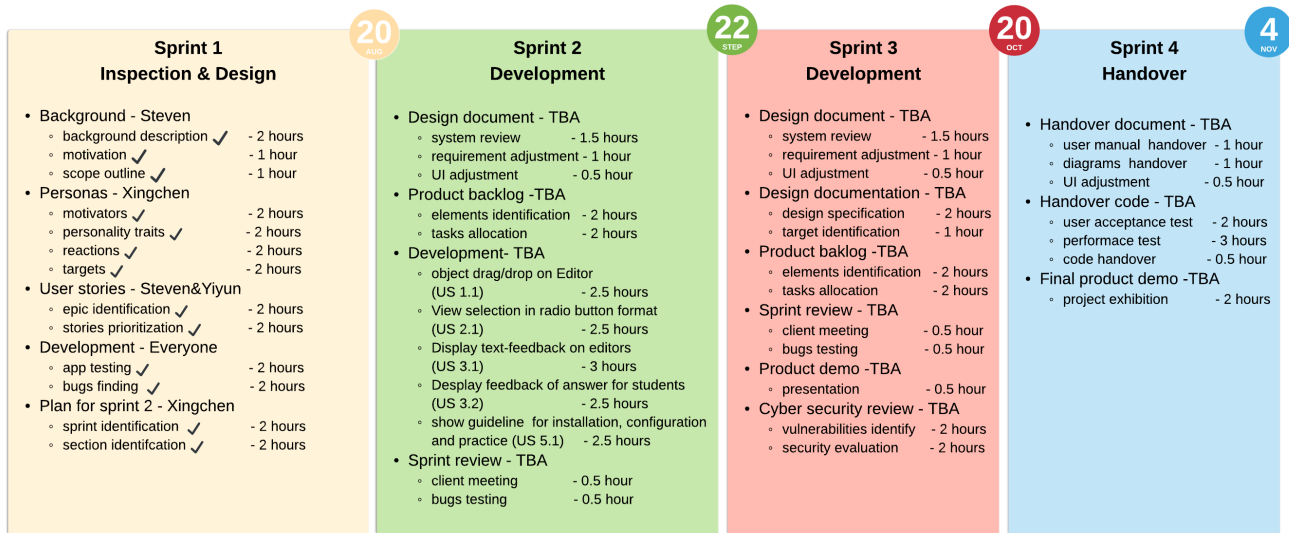
Discussion items

- Sprint planning
- User stories breakdown

Epic	User Story ID	User Stories	Breakdown of User Stories with more details	Mos Cow Priority
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.	Sprint 2 (Steven - Drag, Carol - Drop) The black box when hovering over the statement still shows when dragging the actual statement Unexpected behaviour of elements when dropping onto different areas Imprecise drag and drop. In the web version, you can only put the statement into the connector box, but not the other way around.	Must Have
	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.	Leave this user story to the end as it's not a must-have	Should Have
	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.	Leave this user story to the end as it's not a must-have	Should Have
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.	Sprint 2 Converting the statements and connectors into a single box which displays the statement for easier readability. There are three different formats for the radio button. (Stark, Jodie)	Must Have
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.	The answer would be derived from the Grader so this user story will be focused on in the 3rd sprint. Editor will just display what the Grader has, including a tick/mark for the statement and feedback text for the statement. Can look into testing with test data initially. We can build the UI first (Sprint 2), then display the actual data. (Sihan)	Must Have
	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.	The answer would be derived from the Grader so this user story will be focused on in the 3rd sprint.	Must Have
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.	Will need to ask the client to clarify this aspect. Looks like this user story isn't included in the client's requirements so don't need to focus on this one at this stage.	Should Have
	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.	To align the frontend and backend and database. This is not a must-have.	Should Have

Product deployment	5.1	As a student/teacher, I want to receive comprehensive guidelines covering installation, configuration, and best practices.	<p>Sprint 2 (Everyone does a bit)</p> <p>Documentation for the users. To help students/teachers launch the webapp so they can use it.</p>	Must Have
--------------------	-----	--	---	-----------

Plan for next sprints



Code Review for Sprint 2

A completed peer-to-peer code review for Sprint 2 using the recommended COMP90082 template which details how our team completed the code review and the relevant issues identified.

Please see the below spreadsheet attached which is also available on our GitHub at: <https://github.com/COMP90082-2023-SM2/BI-Bluering/tree/main/docs>



COMP90082_TMPLT...g Sprint 2.xlsx

Sprint 2 Demonstration

A short recorded video which demonstrates our development progress after Sprint 2.

https://www.youtube.com/watch?v=c99gLT0rCrQ&ab_channel=Carol

Sprint 2 Review - Meeting notes

Date

2pm Thursday 21 Sep 2023 Week 9 of Semester

Zoom Recording Link

https://drive.google.com/file/d/1qdL2sa9Yqlna7RwR7FQHKiE6zb5izmcl/view?usp=share_link

Attendees

- Mike Murray (Client)
- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang

Discussion items and details

- Talk about what we've achieved
 - We've finished working on the student feedback UI, and radio button
 - Still working on fixing bugs with dragging and dropping, and the statement convertor button (collapse statement)
- Demonstrate student feedback on UI
- Ask client for feedback on UI
 - Feedback should be more fine grained
 - Client's original idea was to have an instant feedback response which presents a tick or cross and a brief feedback message
- Demonstrate the radio button
- Ask client for feedback for radio button
 - Any feedback on if the radio button is vertical or horizontal, is horizontal too long?
 - Academic can choose the options for the dropdown
- Demonstrate the collapse
 - Pop up menu and radio buttons all collapse
 - Not a single line, have line wrap with multiple lines
- Wants undo button first
- Clarify plan for sprint 3
 - Do we want a button for the statement convertor or is double clicking to convert fine?
 - Best to be consistent when it comes to UI interactivity
 - What other features from the Python version do you want us to implement? Or is what we have enough for Sprint 3?
 - The current plan is sufficient

Sprint 3

This folder contains all the relevant Sprint 3 artefacts.

Sprint 3 Planning - Meeting notes

Date

9:30pm Friday 28 Sep 2023 Non-Teaching Week

Attendees

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

Discussion items

- Discussing which Sprint 3 tasks and confluence sections will be completed by who:
 - 1. Confluence - structure - Jiuneng
 - 2. Confluence - meeting minutes - Steven
scope of the project - Yiyun
technologies used in the project - Sihan
user stories - Yiyun
test cases - Jiuneng
 - 3. Sprint Planning and Sprint Review - Xingchen
 - 4. Task Tracking - product backlog and lower-level sprint backlog - Steven
 - 5. Ethical Considerations - Xingchen
 - 6. Code Review - Steven
 - 7. Product Demonstration - Yiyun
 - 8. Cyber Security - Sihan
-
- User stories breakdown and clearly specifying our size estimations and then we worked together as a team to clearly present our completed and uncompleted stories on our Trello board.

Epic	User Story ID	User Stories	MoSCoW Priority	Size Estimation
Improve the Web Editor's usability	1.1	As a student, I want to have precise drag of objects on Editor so that I can precisely drag elements to different areas.	Should Have	Small (1-2 days)
	1.2	As a student, I want to have the ability to drop connector from list of connectors onto existed statements on Answer Area on Editor so that I can merge statements with connectors immediately.	Should Have	Medium (3-4days)
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	Medium (3-4days)
	2.2	As a student, I want to be able to view the selected statement in a one sentence format so that I could have a clear view of my answer.	Must Have	Medium (3-4days)
	2.3	As a student, I want to have a switch button in the statement box so that click the button and switch the format of the statement.	Must Have	Small (1-2 days)
Display the Student Feedback	3.1	As a student, I want to click the check button for my answer so that I can receive feed back to improve my answers.	Must Have	Small (1-2 days)
	3.2	As a student, I want to view immediate text-feedback besides my answer on Editor so that I can know what is right and wrong in the current answer.	Must Have	Large(6-12days)
Product deployment	4.1	As a student/teacher, I want to receive comprehensive guidelines covering installation, configuration, and best practices.	Must Have	Small (1-2 days)

Plan for Sprint 3



The development plan for sprint 3:

1. Editor drag bug fixing and polishing (US 1.1)
2. Editor drop bug fixing and polishing (US 1.2)
3. Add a button for feedback viewing (US 3.1)
4. Display data for student feedback UI (US 3.2.1)
5. Integrate with grader and receive data from the API call (US 3.2.2)
6. Display the feedback from grader on the frontend (US 3.2.3)
7. Prepare documentation with comprehensive guidelines covering installation, configuration, and best practices (US 4.1)

Code Review for Sprint 3



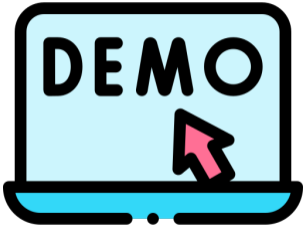
A completed peer-to-peer code review for Sprint 3 using the recommended COMP90082 template which details how our team completed the code review and the relevant issues identified.

Please see the below spreadsheet attached which is also available on our GitHub at: <https://github.com/COMP90082-2023-SM2/BI-Bluering/tree/main/docs>



COMP90082_TMPLT...g Sprint 3.xlsx

Sprint 3 Demonstration



A short recorded video which demonstrates our development progress after Sprint 3.

https://www.youtube.com/watch?v=vwvQAcCnGHk&ab_channel=Carol

Sprint 3 Review - Meeting notes

Date

4pm Friday 20 Oct 2023 Week 12 of Semester

Zoom Recording Link

<https://unimelb.zoom.us/rec/share/USsVWW7daNLRaYbkH06Gah53csWzXwxx7WIPxND0S7cVzXbpYUjR2H0cl02-qzll.wkRU3giUQMmKmwwq>
Passcode: sBS207#s

Attendees

- Mike Murray (Client)
- Xingchen Han
- Yiyun Yang
- Jiuneng Zhang
- Sihan Zhang
- Steven Zhang

Discussion items and details

- The client Mike started the meeting by asking how we were all feeling at the end of the development sprints
 - Overall our team is feeling pretty happy with how we went as we all put in effort until the very end to ensure we completed as many user stories as possible.
 - We completed most of the initial user stories which were required, although there were a few which couldn't be completed due to the level of difficulty and time constraints.
- We talked about what we had achieved and gave an overview of the features which had been completed
 - **Completed tasks/features:**
 - Bug fixed for dragging statements
 - Converting between formats for built statements - Drop-down, Radio Buttons & One Sentence (statement is collapsed)
 - Provide documentation for deployment
 - **Partially completed:**
 - Display feedback from the grader
 - Dropping connectors onto statements -
 - **Haven't started:**
 - Duplicate statements - Ran out of time to start this feature
- Reflected on what we couldn't fully complete and the struggles we encountered
 - Display feedback from the grader - We require more information on the backend data from the RedBack team, as right now we can only display raw feedback data for a specific scenario which lacks flexibility and usability.
 - The dropping of connector onto the statement was very difficult to implement due to technical difficulties and struggles understanding the implementation of the lengthy code segments.
 - We also didn't implement being able to change the orientation of the connectors, only the orientation change of the statements in radio button format was implemented.
 - Ideally we would want to have the ability to duplicate function for statements which future teams will be able to work on.
- Client provided feedback that what we have built and improved on marks a good foundation for what the next BioLogic project teams will work on.
- Confirm what actions to take before handover date of 3rd November
 - We asked the client what he will need from RedBack and us in terms of merging any code or combining our resources
 - Client has confirmed we can just provide them access to our repository
 - We will need to give access of our GitHub repository to Ping and Mike before the end of Sprint 4

Sprint 3 Retrospective - Meeting notes

Date

4:45pm Thursday 20 Oct 2023 Week 12 of Semester

Attendees

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

Discussion items and details



What Went Well

- Our team felt that our communication and teamwork had improved relative to Sprint 2, as we reached out to each other on a more regular basis to provide updates on our progress, to clarify any uncertainties, and to ask for feedback and assistance when required.
 - This helped us keep track of each others' progress and any project challenges and roadblocks we were facing.
- For this sprint, we completed the majority of our user stories which illustrates our effective task size estimations, task tracking, and time management efforts.
- Our team also took on the feedback we received from sprint 2 and made improvements to our Trello board, task tracking information, size estimation process, and overall Confluence documentation.
- There was a lot of efficient teamwork displayed during this sprint as most of us teamed up with someone else to tackle the development tasks via pair programming.
 - This way, one member can spectate someone who is driving the actual programming, while being able to offer guidance and assist with any conceptual understandings, debugging, and helpful suggestions.
 - We found this to be very beneficial for both parties as they were able to learn off each other, leading to faster results being produced.



What Did Not Go Well

- We did not coordinate well enough with Team RedBack to gain a stronger understanding of their backend system and the data they were going to feed through to our frontend.
 - If we communicated earlier and on a more frequent basis with Team RedBack, we may have been able to achieve stronger results for our frontend feedback system.
- Some of our testing was also lacking in this sprint as we left it till last minute due to the development tasks being completed during the last week before the deadline.
 - If we spent more time testing and reviewing the code, the code submitted may have been of higher quality.



What We Will Do

- Our team will continue to maintain our levels of quality communication and regular meetings to ensure the rest of the project is completed to an adequate standard.
- For sprint 3, we plan to ensure all the required artefacts are submitted and have been completed to a much better standard than sprint 2 after reviewing our previous works' feedback.
- For sprint 4, we plan to hand over all the deliverables to the client by giving them access to our repository.
 - We will ensure there is attached comprehensive documentation to help understand our resources and the changes we have made.
- We will improve our code review efforts to ensure our code is of the high quality before the final product and code submission.

Final Presentation

This page contains our final presentation recording as well as the PDF of our final slides submission for Sprint 3.

Zoom Recording Link

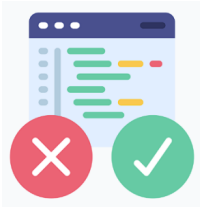
https://unimelb.zoom.us/rec/share/pdBmdOsStLdGwcue8Yy0_Jlv-g30KNxD0oWjMgzftLJHaGHcvuikZy96HTme-FvT.6Uk7vJeONwiMiWS2
Passcode: Pic#15#f

Final Presentation Slides



BI-BlueRing Sprin...tation Slides.pdf

Testing



Test case ID	Test case description	Test steps	Test data	Expected results	Actual results	Pass /fail
TC001	Check that precisely drag elements to different areas.	1. Drag and drop a statement into the answer area and check whether the tooltip is hidden during the process.	All statements	1. The tooltip is no longer displayed after dragging statements.	Same as the expected results.	Pass
TC002	Check that there is a switch button in the statement box and click the button to switch the format of the statement.	1. Drag and drop a statement into the answer area. 2. Check that there is a switch button on the left side of the statement box. 3. Click the button to check whether the statement format is switched.	Statement Root and Statement Student	1. There is a switch button in the statement box. 2. Click this button to switch the format of the statement.	Same as the expected results.	Pass
TC003	Check that the selected statement is in a radio-button format.	1. Select the options. 2. Click the switch button of the statement. 3. Check that the statement is in a radio-button format. 4. Click the switch button again to check whether the statement format is switched.	Statement Root and Statement Student	1. The statement is in a radio-button format. 2. The options can be selected.	Same as the expected results.	Pass
TC004	Check that the selected statement is in a one sentence format.	1. Select the options. 2. Click the switch button of the statement. 3. Check that the statement is in a one sentence format. 4. Click the switch button again to check whether the statement format is switched.	Statement Root and Statement Student	1. The statement is in a one sentence format. 2. The options can be selected.	Same as the expected results.	Pass
TC005	Check that the feedback can be immediately viewed on Editor.	1. Click the submit button to upload the answer to the grader. 2. Click the feedback button. 3. Display a pop-up window and get the feedback from the grader.	All statements and connectors	1. The answer is uploaded successfully. 2. The feedback is displayed besides the answer.	Same as the expected results.	Pass
TC006	Check that the development environment can be installed and configured according to comprehensive guidelines.	1. Setting up docker and initiate client username. 2. Setting up Vue and running the development server. 3. Run this project.	Project code	1. Run the command according to the guidelines. 2. View the Vue application in a browser.	Same as the expected results.	Pass

Meeting Notes

[Create meeting note](#)



All meeting notes

All our notes from team meeting, supervisor meetings, and client meetings are laid out under this parent page. Individual folders can be found in the sidebar.

Title	Creator	Modified
Sprint 2 Review - Meeting notes	Steven Zhang	about 2 hours ago
Sprint 2 Planning - Meeting notes	Steven Zhang	about 2 hours ago
2023-09-08 Weekly supervisor meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-09-01 Weekly supervisor meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-08-28 Team meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-08-25 Weekly supervisor meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-08-23 Team meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-08-18 Weekly supervisor meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-09-15 Weekly supervisor meeting - Meeting notes	Steven Zhang	22 Sep, 2023
2023-08-11 Weekly supervisor meeting - Meeting notes	Steven Zhang	18 Aug, 2023
2023-08-04 Weekly supervisor meeting - Meeting notes	Steven Zhang	18 Aug, 2023
2023-08-10 First client meeting with combined teams - Meeting notes	Steven Zhang	18 Aug, 2023
2023-08-13 Team meeting - Meeting notes	Steven Zhang	18 Aug, 2023
2023-08-17 Team meeting - Meeting notes	Steven Zhang	18 Aug, 2023

Team Meetings

All our notes from team meetings are accessible from under this folder on the sidebar. Please note that any Sprint ceremonies/meetings are kept under the relevant Sprint folder.

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

2023-08-23 Team meeting - Meeting notes

Date

9:30pm Wednesday 23 Aug 2023 Week 5 of Semester

Attendees

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

Discussion items

- Send through the entire confluence PDF to Mike on Slack for feedback
- Update confluence page
- Clarify Ping's responsibility
- Carol asked Ping a lot of questions on how to open the WebApp
- Running the App and testing for bugs - Everyone to do
- ReadME page for background page not complete
- For sprint 2 plan, we need a design documentation

2023-08-28 Team meeting - Meeting notes

Date

11am Monday 28 Aug 2023 Week 6 of Semester

Attendees

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

Discussion items

- Assigning tasks
- Reviewing Mike's
- Confirm when to have the sprint review meeting with client
- Sprint 2 due 22nd Sep

2023-10-12 Team meeting - Meeting notes

Date

7:30pm Thursday 12 Oct 2023 Week 11 of Semester

Attendees

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

Discussion items

- Reflecting on Sprint 2 feedback
- Task tasking:
 - Detail overall task tracking steps
- Size estimation
 - Use hours
 - How did we come up with these sizes?
- Trello board
 - Correction of names
 - Assign people to each user story card
 - Everyone will watch Mike's youtube video properly tonight and think about which user stories are important and achievable given the Sprint 3 deadline of 20th Oct midnight
 - Team will meet again after supervisor meeting tomorrow 13th Oct at 5:45pm to clarify user stories for Sprint 3 and what's actually achievable
- Final presentation: Need to do slides & demonstration link
 - Go through user stories
 - Order of presentation: Stark (project intro), Sihan (goal), Carol (demo), Jodie (user stories), Steven (handover & reflection)
 - Create google slides for presentation (to be done next week)
 - Also prepare script - each person to speak for around 1-2mins
 - Will also need to answers questions for 2 minutes at the end
- Reminder to complete self assessment and peer review by 13th Oct midnight

2023-10-13 Team meeting - Meeting notes

Date

6pm Friday 13 Oct 2023 Week 11 of Semester

Attendees

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

Discussion items

- Small (1-2 days), medium (3-4 days), large (5-6 days) size estimations
- Edit and fix Trello board as a team
 - Add in due dates and break down larger tasks
- Need to organise Sprint 3 review meeting time for Friday 20th Oct
 - Steven to reach out to Team RedBack and client to organise meeting time
- Aim to complete all user stories by wednesday 18th Oct
 - Spend thursday consolidating everything, improving confluence, and preparing for sprint review and final presentation
- Team meeting on Monday 16th Oct 5pm

2023-10-16 Team meeting - Meeting notes

Date

5pm Monday 16 Oct 2023 Week 12 of Semester

Attendees

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

Discussion items

- Sprint 3 Review meeting on Friday around 4PM
 - Need to confirm with the client
- Next team meeting on Wednesday 18th Oct 2PM

2023-10-19 Team meeting - Meeting notes

Date

4:30pm Thursday 19 Oct 2023 Week 12 of Semester

Attendees

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

Discussion items

- Confirmed which user stories had been completed and which ones still need some more finalising before merging with the main branch
- We will have a team meeting tomorrow to do a practice run through of our final presentation to make sure our presentation is 10 minutes long.

2023-10-20 Team meeting - Meeting notes

Date

3pm Friday 20 Oct 2023 Week 12 of Semester

Attendees

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

Discussion items and details

- Worked on the final presentation slides as a team and finalised our presentation preparation
- As a team, we did a practice run of our final presentation to ensure it was around 10 minutes long

Supervisor Meetings

All our notes from supervisor meetings are accessible from under this folder on the sidebar.

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,
<ul style="list-style-type: none">• Confluence workspace - no one has used before https://confluence.cis.unimelb.edu.au:8443/display/COMP900822023SM2BIBBlueRing/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

2023-08-11 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

2023-08-18 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 18 Aug 2023 Week 4 of Semester

Attendees

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

Discussion items

- Stand up meeting, what have we done? What do we need to do?
- Next step is sprint 2, start the development
- Yiyun reached out to Mike & Ping on how to login into the App
- Downloaded the code, set up the repo
- What are Ping's responsibilities? Where does her role start and end? Update client table, add role responsibility, advisory role,
- Add key to the goal model diagram so that it's clear what the symbols mean
- Ask the client to review and validate those personas? They should represent reality as much as possible.
- Can we please get Mike to review and accept the user stories?
- Confluence and Trello are project management tools, not needed to be included in the development environment, enough to include it on the homepage
- Try to structure confluence so it's easier for people to find
- Vue - how to learn or get started?
- <https://www.youtube.com/@WebDevSimplified/videos>
- Get Mike to review documentation: send github release zip file or just pdf, just want to get some feedback on the user stories particularly, note on confluence if client has reviewed or agreed on our documentation for quality assurance purposes

2023-08-25 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 25 Aug 2023 Week 5 of Semester

Attendees

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

Discussion items

- What have we done this week?
- Pretty slow week for me, just been getting to know the languages, raised the user stories with Mike for validation
- Do we have documentation for how to set up the App on our own computers?
- Why are we installing python and vue versions?
- We still need the python version/backend version so we can confirm both the python and vue versions.
- Can we change the README file?
- Apart from these steps, need to do these steps instead,
- Document our learnings and frustrations,
- Are we supposed to modify the Python version? How do we open the Python version?
- We need to clarify the user stories we're looking to complete for Sprint 2
- Backlog items don't need to be there, don't need sprint review, we need user stories on the backlog, technical tasks that need to be done, we only track for development effort on the Trello board predominantly only user stories, client cares about the user stories and features which have been completed
- We should break up the user stories into smaller user stories and technical tasks
- Need to lock in a meeting with client, sprint review meeting with the client in 3 weeks time
- Need feedback for the next sprint and how to plan
- Most of the user stories are independent
- Should focus on must haves first
- Everyone needs to have development environment set up

2023-09-01 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 01 Sep 2023 Week 6 of Semester

Attendees

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

Discussion items

- Discuss feedback from Sprint 1 and what needs to be worked on and improved from Sprint 1
- Put user stories into the Trello Board, consider using checklist, include member names on the Trello tasks
- The task size estimation should be relative to other tasks, the simplest task should be 1 unit
- For sprint planning, a big task should be broken down into smaller individual user stories
- Risk discussion: Product risk, process risk(technology skills), project risk(requirements)
- Difference between risk and an issue: Issue is something we need to fix, risk is something we have to consider as it might occur, security risk, operation risk, these risks should be documented and tabulated on Confluence, how to assess project risks (impact).
- Cybersecurity review to come later in the project
- Deliverable Sprint 2 is due week 9, sprint review can be before the end of sprint 2. Could ask this question on EdDiscussion. Reach out to the client to confirm the date for sprint review.
- Think of the project like a game - Can we sprint to the end and achieve the desired product with the lowest number of bugs and defects?
- Blockers column to be included in the Kanban Board

2023-09-08 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 08 Sep 2023 Week 7 of Semester

Attendees

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

Discussion items

- More details for each Trello card
- Create folder for team management - Trello workflow: Once we have finished developing the task, please move it to Code Review then testing, then move it to Done.
- What process did you follow to produce those estimates? Did you estimate, get a majority vote? Document the process for producing those estimates.
- How do we document testing? Extra section for testing.
- ChatGPT prompt: "I'm a software developer, what should we include in our test plan?"
- Please document the design architecture - create a separate document for architecture, which is a blueprint for the whole project, system block diagram. What are the software components which make the application? Backend? APIs? How do the modules fit together? Ask client for architecture diagrams. Ask the other team as we're both responsible for different types of architecture.
- Create separate Slack channel for our team and supervisor - don't include the client
- There should be a testing framework for Vue.Js library, similar to React
- <https://testing-library.com/docs/vue-testing-library/intro/>

2023-09-15 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 15 Sep 2023 Week 8 of Semester

Attendees

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

Discussion items

- Breakdown down the user story into subtasks
- Create checklists for each user story
- What is the actual behaviour of the bugs? Detail this into the Trello card.
- Talking through each Trello card and providing updates on progress.
- Send meeting invite to Mike Murray via Gmail & include the Zoom link
- Consider back to back meeting with the other RedBack team for Sprint Review
- Reach out to the other team members to arrange meeting time for Thurs 21st Sep 1pm

2023-09-22 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 22 Sep 2023 Week 9 of Semester

Attendees

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

Discussion items

- Plan a retrospective meeting:
- What we did well, what we didn't do well, what can we improve on for next time
- Not product related
- Do a test plan, which test cases pass and fail, so the next team knows what to work on
 - Test cycles, tests passed and failed

2023-10-06 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 06 Oct 2023 Week 10 of Semester

Attendees

- Paul Calverley (Supervisor)
- Jiuneng Zhang
- Steven Zhang

Discussion items

- We had a meeting with RedBack to clarify what their backend feedback data will look like
- We clarified user stories with Mike for Sprint 3
- Task tracking should be separate to size estimation
 - Documentation around process for when and how team moves tasks from doing, to testing, and to done
 - Task workflow tracking and estimation should be different things
- Scrum master to make sure Trello board plan is up to date and correct
 - Some names are not reflecting correctly
- Most likely will be getting feedback for Sprint 2 today or very soon
- Looking to do presentations for groups from 16th-20th Oct
 - 12-13 min presentation+2 min questions
 - Paul will send out a poll or message on Slack to arrange times for both groups to have back to back meetings with clients
- Need to add Sprint 2 retrospective notes on confluence
 - What worked well and what didn't work well
- Try to get all team members to join next week's supervisor meeting
- Discuss how to integrate two different teams' projects into the same repository
 - How will the projects merge back into the client's project?
 - Will it be a manual or automated process?
 - Paul will ask and confirm with RedBack on monday

2023-10-13 Weekly supervisor meeting - Meeting notes

Date

5:15pm Friday 13 Oct 2023 Week 11 of Semester

Attendees

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

Discussion items

- Need to fix the size estimations on Trello
- Consider creating a "blocked" section for tasks which have run into issues and roadblocks
- Go through the feedback for Sprint 2
- Need to have separate Sprint 3 review meeting with client before 20th Oct
 - Paul suggests having a joint sprint review meeting with Team RedBack and the client due to the dependencies involved from both teams
- Need to use RedBack's API information to link to their backend data
 - Need to clarify with RedBack how to use the API information they have sent through and what's the endpoint? What's the method? What's the data?

Client and Team RedBack Meetings

All our notes from client meetings are accessible from under this folder on the sidebar. Please note that any Sprint ceremonies/meetings are kept under the relevant Sprint folder.

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

2023-09-28 Meeting with Team RedBack - Meeting notes

Attendees

- Yiyun Yang
- Steven Zhang
- Team RedBack (Yash and Leon)

Discussion items

- Team RedBack is working on grader interface, displaying on student Xnet,
 - What changes we've made to our frontend user interface
- Team RedBack will send through code and information around their API and backend date by the end of next week