Grades Management System

for Instructors

Of the

Pharmacy Assistant Program

By

Syed Jawad Raza Baquar

Luan Dao

Ahmad Heshmati

Yuta Kikuchi

Qiao Mu Lei

Prepared for

School of Public Safety

Requested by

Alex Schmidt, Instructor of CPSY 301

Southern Alberta Institute of Technology

# Executive Summary

This report summarizes the development of a Pharmacy Assistant Grading System for the Southern Alberta Institute of Technology (SAIT). The system will solve the inefficiencies of the manual grading system that instructors are currently using.

The purpose of this document is to report the summary of the requirements gathering and analysis phase, serving as a high-level reference for future use.

Introduction: This document first mentions details of issues that instructors at SAIT face currently since they use physical grading sheets with specific formats in order to grade student’s labs. The project focuses on creating a solution for these inefficient grading sheets, accessible on mobile devices, with functions for exporting grades.

Methods**:** The team conducted interviews with the user to gather information about the current way to grade student’s labs, including grading rules, access requirements, and data export needs. Comprehensive analysis and understanding of the requirements are involved in this phase.

Use Case Diagrams and Descriptions**:** The report represents detailed use case diagrams for instructors. They can log in, manage blocks, create/edit weeks, create/edit grading sheet templates, grade students' labs, and export reports. The accompanying descriptions elaborate on the flow of each use case.

Class Diagram**:** A class diagram shows the system's structure, encapsulating essential data and functionalities. This provides a visual representation of how a variety of components interact with each other within the system.

**UI Design:** TheUI interface will present the workflow of our application and show how we will implement our use cases.

Conclusion**:** The report concludes by summarizing the project's goals and functionalities. It emphasizes the replacement of the manual grading process with a web-based solution, enabling instructors to create labs, grade student work, and export reports. The document highlights the readiness of the project to move into the development phase, offering an efficient solution for grading pharmacy assistant lab work.

In summary, this report outlines the problem, requirements, and proposed solution for the Pharmacy Assistant Grading System at SAIT. It provides a clear understanding of the project's scope, functionalities, and the team's readiness to proceed with the development phase.

[Executive Summary II](#_Toc152880418)

[Introduction: II](#_Toc152880419)

[Methods II](#_Toc152880420)

[Use Case Diagrams and Descriptions II](#_Toc152880421)

[Class Diagram II](#_Toc152880422)

[Conclusion II](#_Toc152880423)

[Table of Figures VI](#_Toc152880424)

[Introduction 1](#_Toc152880425)

[Purpose 1](#_Toc152880426)

[Background 1](#_Toc152880427)

[Scope 1](#_Toc152880428)

[Methods 1](#_Toc152880429)

[Preview 1](#_Toc152880430)

[Project Selection 2](#_Toc152880431)

[Use Case Diagram 3](#_Toc152880432)

[Instructor Use Cases 4](#_Toc152880433)

[Create blocks: 4](#_Toc152880434)

[Select block: 4](#_Toc152880435)

[Use Case 1: Create/select blocks 4](#_Toc152880436)

[Create week: 5](#_Toc152880437)

[Select week: 5](#_Toc152880438)

[Create/Select grading sheet template: 6](#_Toc152880439)

[Edit lab/grading sheet template: 6](#_Toc152880440)

[Grade lab for each student: 7](#_Toc152880441)

[Choose RX#: 8](#_Toc152880442)

[Edit Grading: 8](#_Toc152880443)

[Export Report: 9](#_Toc152880444)

[Add, remove and edit RX records: 9](#_Toc152880445)

[Create/Select the Lab: 10](#_Toc152880446)

[Data Storage and Persistence 18](#_Toc152880447)

[Real-time Database: 18](#_Toc152880448)

[Authentication: 18](#_Toc152880449)

[Scalability: 18](#_Toc152880450)

[Serverless Architecture: 18](#_Toc152880451)

[Integration: 18](#_Toc152880452)

[Documentation and Community Support: 18](#_Toc152880453)

[Creating the project: 19](#_Toc152880454)

[Setting up the database: 19](#_Toc152880455)

[Back-end Integration: 19](#_Toc152880456)

[Front-end Integration: 19](#_Toc152880457)

[File Storage: 19](#_Toc152880458)

[Testing and Deployment: 19](#_Toc152880459)

[Entity Diagram 20](#_Toc152880460)

[Class Diagram 21](#_Toc152880461)

[Create/Select Block: 22](#_Toc152880462)

[Create/Select Week: 23](#_Toc152880463)

[Create/Select Lab: 24](#_Toc152880464)

[Grade lab for each student: 25](#_Toc152880465)

[Activity diagram: 26](#_Toc152880466)

[State Machine diagram: 26](#_Toc152880467)

[System Architecture and Patterns: 27](#_Toc152880468)

[Frontend Layer (Client Side): 27](#_Toc152880469)

[User Interface (UI): 27](#_Toc152880470)

[Backend Layer (Server-Side with Next.js API Routes): 27](#_Toc152880471)

[API Layer: 27](#_Toc152880472)

[Authentication: 27](#_Toc152880473)

[Authorization: 27](#_Toc152880474)

[Database Layer (Firebase Firestore): 28](#_Toc152880475)

[Database Design: 28](#_Toc152880476)

[Data Access Layer: 28](#_Toc152880477)

[Integration Layer: 28](#_Toc152880478)

[Prototype 28](#_Toc152880479)

[Work Breakdown Structure and Gantt Chart 29](#_Toc152880480)

[LOC-Based Estimation Table 29](#_Toc152880481)

[PNR Reconciliation 30](#_Toc152880482)

[Division of Labor 30](#_Toc152880483)

[Gantt Chart 31](#_Toc152880484)

[[PDF Included separately] 31](#_Toc152880485)

[Conclusion 31](#_Toc152880486)

[Appendix A: Team Constitution A](#_Toc152880487)

[Team Constitution B](#_Toc152880488)

[Team Name: B](#_Toc152880489)

[Group members: B](#_Toc152880490)

[Team Goals C](#_Toc152880491)

[Intellectual Property Statement C](#_Toc152880492)

[Roles:    Leading the team: Ahmad Heshmati, Luan Dao C](#_Toc152880493)

[Editing the final document: Luan Dao C](#_Toc152880494)

[Formatting the final document: Syed Jawad Raza Baquar C](#_Toc152880495)

[Ensuring the document complies with IEEE format C](#_Toc152880496)

[Keeping the team on track with deadlines D](#_Toc152880497)

[Ensuring the document meets the report guidelines: D](#_Toc152880498)

[Division of labor D](#_Toc152880499)

[Norms and Expectations D](#_Toc152880500)

[Conflict Resolution Plans D](#_Toc152880501)

[Punishment for Offences F](#_Toc152880502)

[Contract Conclusion H](#_Toc152880503)

# Table of Figures

[Figure 1 Instructor Use Cases 3](#_Toc152865045)

[Figure 2 Home Page 12](#_Toc152865046)

[Figure 3 List of Weeks Page 14](#_Toc152865047)

[Figure 4 List of Labs Page 15](#_Toc152865048)

[Figure 5 Classlist page 16](#_Toc152865049)

[Figure 6 Marking page 17](#_Toc152865050)

[Figure 7 Notes/Comments section 18](#_Toc152865051)

[Figure 8 Entity Diagram 22](#_Toc152865052)

[Figure 9 Class-based model of the grading system 22](#_Toc152865053)

[Figure 10Sequence Diagram of Creating/Selecting a Block 23](#_Toc152865054)

[Figure 11 Sequence Diagram of Creating/Selecting a Week 24](#_Toc152865055)

[Figure 12 Sequence Diagram of Creating/Selecting a Lab 25](#_Toc152865056)

[Figure 13 Sequence Diagram of Grading a lab for a student 26](#_Toc152865057)

[Figure 14 Activity Diagram of Marking a Lab for a student 27](#_Toc152865058)

[Figure 15 State machine diagram of the system as it undergoes state changes during the process of marking/viewing a lab for one student. 27](#_Toc152865059)

# Introduction

This is the resultant work product from conducting the communication and planning phases for our Pharmacy Assistant Grading System.

## Purpose

This document was created to summarize the results of the requirements gathering and analysis done. It will serve as a high-level guide for us in the future should we need a reference.

## Background

Instructors currently use physical grading sheets with a specific format to grade students’ labs based on the criteria listed within. There are certain rules for how the grades are supposed to be tallied and whether a student achieved a satisfactory mark for that lab.

## Scope

We are only creating and implementing a solution to the issue of those specific physical grading sheets. The system will not allow for general marking of other types of assessments. The system should be accessible on mobile devices and the grades calculated by the system should be exported; however, the means by which these requirements will be achieved are to be determined.

## Methods

We conducted an interview with the customer and gathered the necessary information. Information pertaining to the existing system such as: what is currently used to record grades, how grades are calculated, how the proposed system should be accessed, and how the data generated by the system should be exported was attained.

## Preview

This report includes two use case diagrams depicting how the two primary users of the system will interact with it in most cases. Accompanying the use cases are their descriptions to elaborate upon them. Following the use case diagrams is a class diagram created from analyzing the use cases and extracting the pieces of information that are retained by the system and is necessary for operation and grouping them into classes. Attached within the appendix is also our Team Constitution.

# Project Selection

* Identify your project.

We are tasked with creating a pharmacy assistant lab grading system. Instructors will use the system to evaluate the lab work completed by students over several weeks. Certain business rules and requirements regarding the system’s functionality that reflect how instructors currently mark lab work will need to be considered. The system will be used to grade student labs and print reports to indicate a student’s progression through the labs withing their program.

* Name your project sponsor and, where applicable, the name of their organization.

Sheri Shahid, a representative of the instructors withing the Pharmacy Assistant program, is our project sponsor. The system will be implemented at the Southern Alberta Institute of Technology.

* Describe the sponsor’s current system, where applicable.

The instructors currently use grading sheets with a specific format to grade students’ labs based on the criteria listed within. Instructors need to follow certain rules with regards to grading. For instance, the students will fail the whole lab if they fail the first and second criteria. Such operations are currently done by the instructors manually and there is no automatic system.

* Detail your team’s proposed solution.

Our team’s proposed solution is a web-based system that enables users to utilize a variety of features to organize and mark students’ lab work. To achieve this primary business goal, the system will allow instructors to create labs, scenarios, and patients. The system will automatically create an appropriate number of grading sheets for class based on a class list provided and link them to the appropriate lab or week. Within these grade sheets, there are fields for instructors to complete when they evaluate a student’s work. Certain fields prompt for information to identify the scenario. When instructors open the grade sheets to begin marking, the system will automatically fill in these fields with information of the scenario associated with the lab and patient for that week. The remaining fields are criteria against which a student’s work is marked. They will also use the system to export the grade data that was created when they marked the lab work into either PDF’s or Excel spreadsheets. or managing students’ information and lab examinations, creating class lists and gradable components, and exporting the stored data for entry into Brightspace.

# Use Case Diagram

A diagram of a workflow

Description automatically generated

Figure Instructor Use Cases

# Instructor Use Cases

Create blocks:   
This use case shows that the instructor can create blocks of student(s) that they oversee. Marking will be done per week per lab based on the selected block.

Select block:   
After browsing the available blocks of students, the instructor can select one of them and continue working with the system.

## Use Case 1: Create/select blocks

Precondition: Users must be in the system.

Postcondition: A block that instructors with which instructors can work is available.

Limitation: Users cannot create a block without adding students.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. An instructor selects:   Create blocks (refer to alternate flow 1.1) | 1. Prompt users to add/remove students. Display a list of students if it already exists. |
| 1. Add/remove students (refer to error flow \*.1)     Continue when users make sure that students are added. | 1. Ask users to edit the name of the block. |
| 1. Edit the name and select “Create”. (refer to error flow \*.2) | 1. Create the block and save it. |

Alternate Flow 1.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Display existing blocks. |
| 1. Choose a block. | 1. A block is selected. |

Error Flow \*.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.1 If the same student is already added, system displays an error message |

Error Flow \*.2

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.2 Display an error message if the same name already exists in the system. |

Create week:   
Instructors can create a week in the dashboard based on their needs. These weeks will allow them to manage and organize the contents and labs chronologically and according to the timeline of the course.

### **Select week:**

The instructor can select a particular week to look at all the labs that were created for that week.

Precondition: A block is selected.

Postcondition: A week is available in the dashboard and is ready for the addition of a lab.

Limitation: Changes are discarded if the user interrupts the operation.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Display existing weeks |
| 1. An instructor selects:   Create week (refer to alternate flow 2.1) | 1. Prompt to enter the name of the week. |
| 1. Enter the name and select “Create”. (Error Flow \*.1) | 1. Create a week in the dashboard. |

Alternate Flow 2.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Select a week. | 1. A week is selected. |

Error Flow \*.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.1 Ask to choose a different name if the name already exists. |

Create/Select grading sheet template:   
It allows the instructors to create or select a grading sheet template which contains the marking criteria and passing threshold to use for a specific lab that they are creating. A default template is available for the instructors to select.

Precondition: An instructor already selected specific block and week.

Postcondition: A grading sheet template is created/selected for use within a certain lab.

Limitation: Changes are discarded if the user interrupts the operation.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Display all available grading sheets |
| 1. An instructor selects:   Create a grading sheet template(refer to alternate flow 2.1) | 1. Display grading sheets template that is created. |
| 1. Select create a grading sheet template. | 1. Prompt user for a name and marking criteria. |
| 1. User select create (refer to alternate flow 6.1) | 1. Create the grading sheet template. |

Alternate Flow 2.1

|  |  |
| --- | --- |
| **Actor Action** | **System response** |
| 1. Instructor selects one of the available grading sheet templates | 1. The selected grading sheet template is applied to the lab |

Alternate Flow 6.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 6.1.1 Select “Cancel” | 6.1.2 Redirect the user to the previous page. |

Edit lab/grading sheet template:   
After creating an instance of the lab, Instructors can edit its default values for the marking criteria and passing threshold and change it based on their needs.

Precondition: There is at least one lab and the user selected a specific lab.

Postcondition: A lab is edited.

Limitation:  Changes are discarded if the user interrupts the operation.

Maximum marks can be either 3 or 5.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. An instructor selects:   Edit lab (Alternate Flow 1.1) | 1. Display the edit screen |
| 1. The user edits a marking criteria for the lab. |  |
| 1. Click “Save” when editing finished (Error Flow \*.1) | 1. Save the change |

Alternate Flow 1.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1.1.1 Edit maximum marks | 1.1.2   Display options (either 3 or 5) |
| 1.1.3 Select maximum marks and click “Save” | 1.1.4 Save the change |

Error Flow \*.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.1  Display error message if user entered any invalid value. |

### **Grade lab for each student:**

After selecting a certain lab within a week, the instructor can then proceed to grade the lab for each individual student.

Precondition: An instructor selected a particular lab.

                       There is already a class list created.

Postcondition: Grading the lab for each individual student is completed.

Limitation: Changes are discarded if the user interrupts the operation.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Choose an individual student from the class list. | 1. Display a grading sheet for the student. |
| 1. Evaluate the student’s lab by selecting pass/fail and leave comments if needed. | 1. Update the preview of the grading sheet as user modifies |
| 1. Select “save” | 1. Save the grade and display a message |
| 1. Choose another student and start grading. (Alternate Flow 7.1) |  |

Alternate Flow 7.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 7.1.1 User can leave from grading when it is finished, or the user needs to leave in a halfway |  |

### **Choose RX#:**

To create a lab for a certain week, the instructor needs to choose the RX# (prescription) for that lab which contains the patient name and the drug name.

Precondition: RX# is added and exists in the system.

                      User selected a particular lab.

Postcondition: RX# is chosen.

Limitation: User can not choose RX# if there is not any RX record added.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Display RX# after a particular lab is selected. |
| 1. Choose RX# from a list. | 1. Set the RX# and enable users to start grading. |

### **Edit Grading:**

The instructor can edit the grading of each individual student by selecting the corresponding lab for the student.

Precondition: User selected a specific lab.

                      There is at least one graded lab that is saved in the system.

Postcondition: A selected lab is edited.

Limitation: Changes are discarded if the user interrupts the operation.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. An instructor selects:   Edit the lab | 1. Display an edit screen for the grading. |
| 1. Edit the grading by modifying pass/fail for each field and edit the comment if needed. | 1. Update the preview of the grading sheet as user modifies. |
| 1. Click Save. | 1. Save the changes. Display a confirmation message. |

### **Export Report:**

This use case extends select the lab use case. The instructor can export the report generated by the system for each graded lab and submit it on Brightspace or anywhere else as required.

Precondition: There is at least one grading for a lab that is completed in the system.

                        User selected a particular lab.

Postcondition: The report is exported.

Limitation: The report will be exported in PDF format.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. An instructor selects:   Print | 1. Display a preview of the report and a confirmation message. |
| 1. Selects Export | 1. Export the report and display a success message. |

Add, remove and edit RX records:   
Instructors can add, remove and edit RX records stored in the system.

Precondition: A lab is selected, or user have access to existing RX records.

Postcondition: RX record is successfully added, deleted, or edited.

Limitation:  If the user interrupts the work in progress, it will not be saved to maintain data consistency.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. An instructor selects:         Manage RX records | 1. Display existing RX records. |
| An instructor selects either:     1. Add RX record | 1. Ask the user to input number, name of patient and drug. |
| 1. Enter the information and click “Add”. | 1. The new RX record is added. (Alternate Flow 6.1) |
| -or-   1. Delete RX record | 1. Ask to select RX record that the user wants to delete. |
| 1. Select RX record. | 1. Display a confirmation message. |
| 1. Select “Yes”. (Alternate Flow 11.1) | 1. Delete the RX record. |
| -or-   1. Edit RX record | 1. Ask to select RX record to edit. |
| 1. Select RX record. | 1. Display an edit screen. |
| 1. Modify the RX record (number, patient name, or drug name), and select “Save”. | 1. Save the change.   (Error Flow \*.1) |

Alternate Flow 6.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 6.1.1 Display an error message if there is a missing information or the user entered any invalid value. |

Alternate Flow 11.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 11.1.1 Selects “No”. | 11.1.2 Redirect the user to the previous page. |

Error Flow \*.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.1 Display an error message if there is a missing information or the user entered any invalid value. Ask to enter valid information. |

### **Create/Select the Lab:**

After selecting the week, the instructor can then create and or select a particular lab to grade, view or export as needed.

Precondition: A block and a week has been created.

Postcondition: A lab will be selected for the instructor with which they will work.

Limitation: The user can choose the interrupt the process

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Display list of labs |
| Manager selects either:   * 2. Create lab (Alternate Flow 2.1) | 3. Redisplay list of labs |
| -or-   * 4. Select lab (Alternate Flow 4.1) | 5. Display actions (grade, view, export a report) for each student |
|  | 5 Display switchboard screen. |

Alternate Flow 1.1

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1.1.1 Prompt for name, RX#, grading sheet template. |
| 1.1.2 Enter required information (Refer to error flow \*.1) | 1.1.2 Create lab based off of the provided information |

Error Flow \*.1

At any time, if the user enters incorrect information or interrupts the operation, the system will exit the operation and display an error message.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | \*.1 Display error message. |

**UI design**

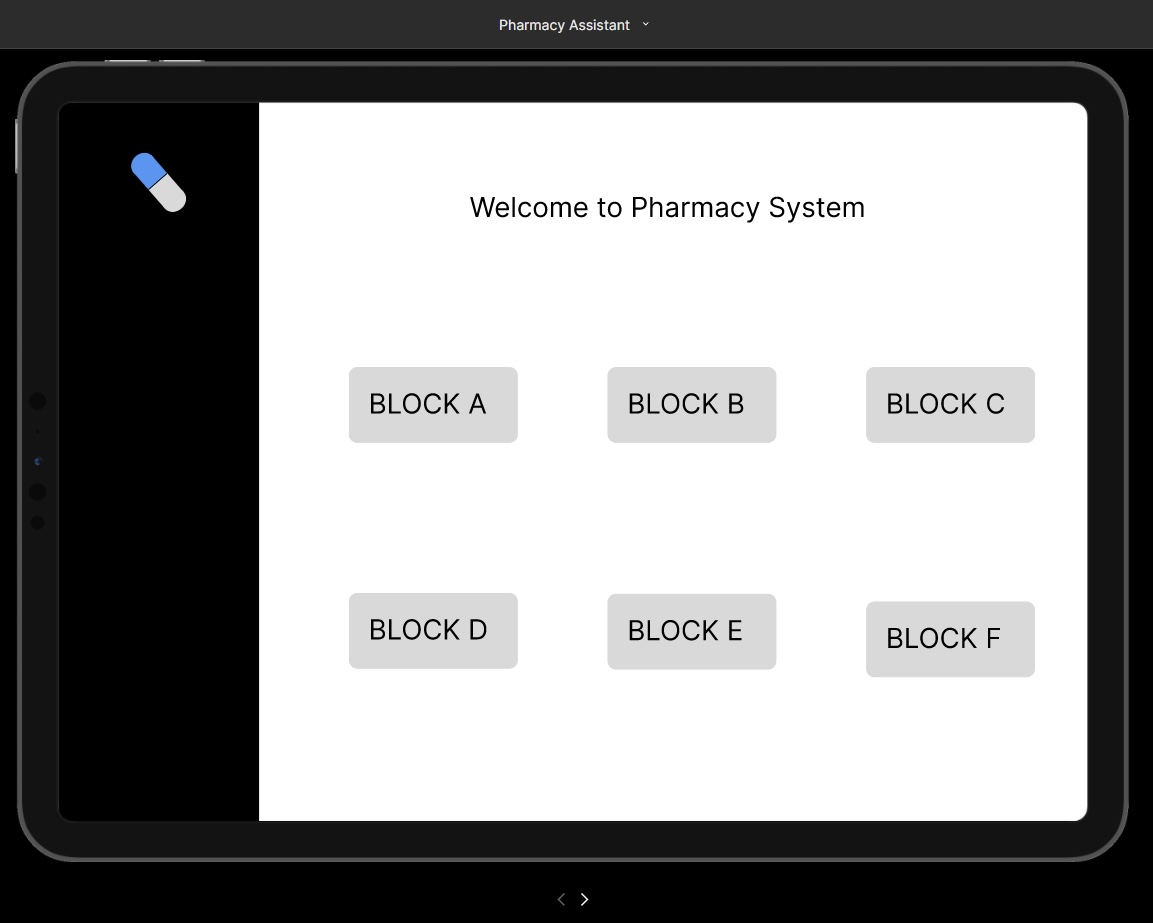


Figure Home Page

**Home page:**

All of the blocks are visible on the main page. To select a block, the instructor needs only click the block tag.

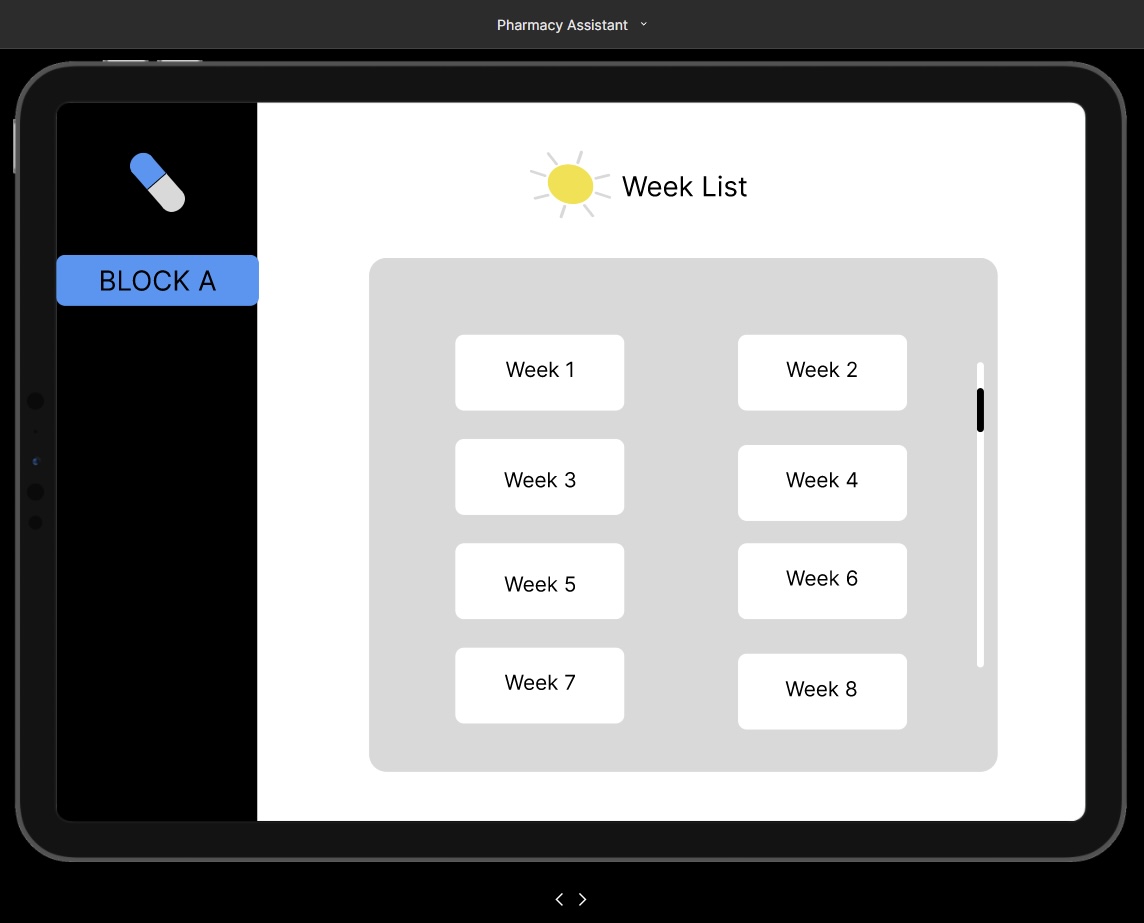


Figure List of Weeks Page

**Week List page:**

The block tag will show on the side bar after the instructor picks one, the week list page presents all the weeks, to select a week, the instructor needs click the week tag.

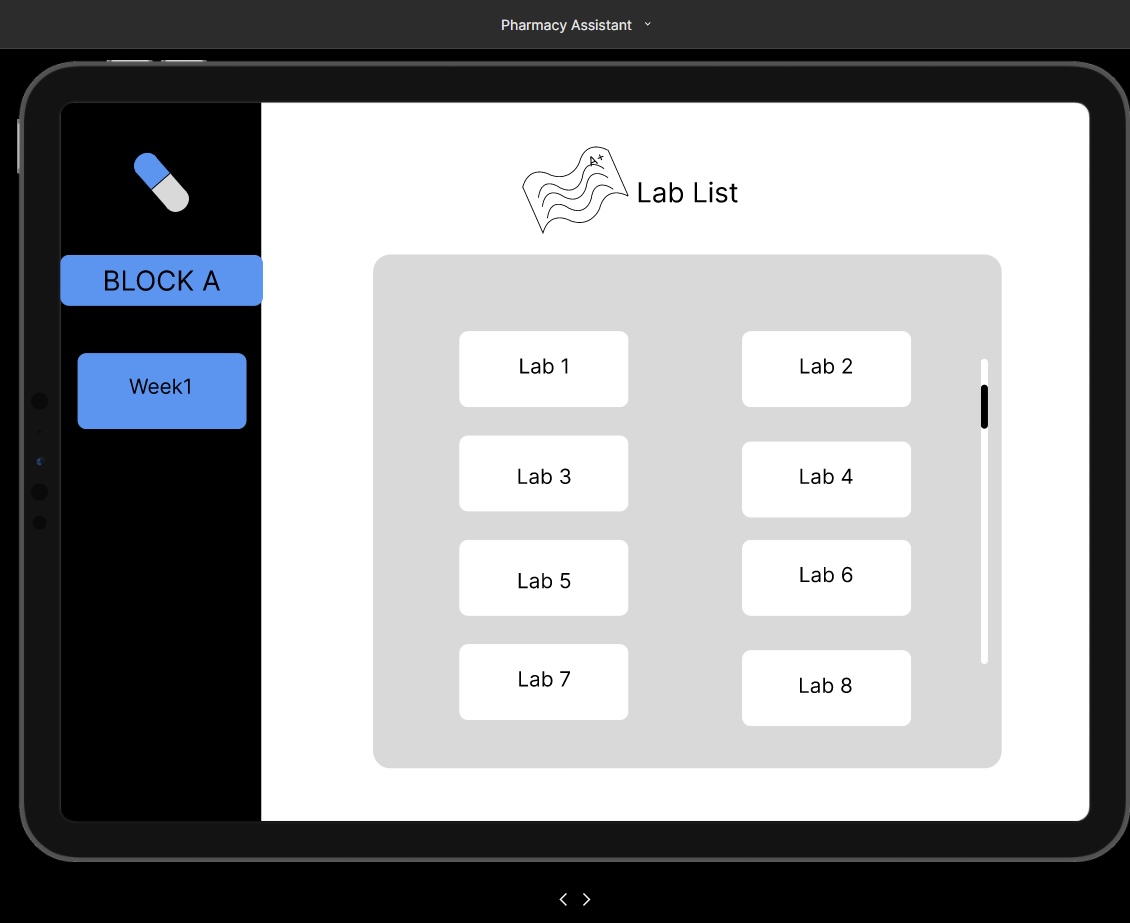


Figure List of Labs Page

**Lab List page:**

After the teacher selects a week, the week tag will appear on the sidebar. All the labs are displayed on the lab list page, and the instructor only needs to click the lab tag to select the lab.

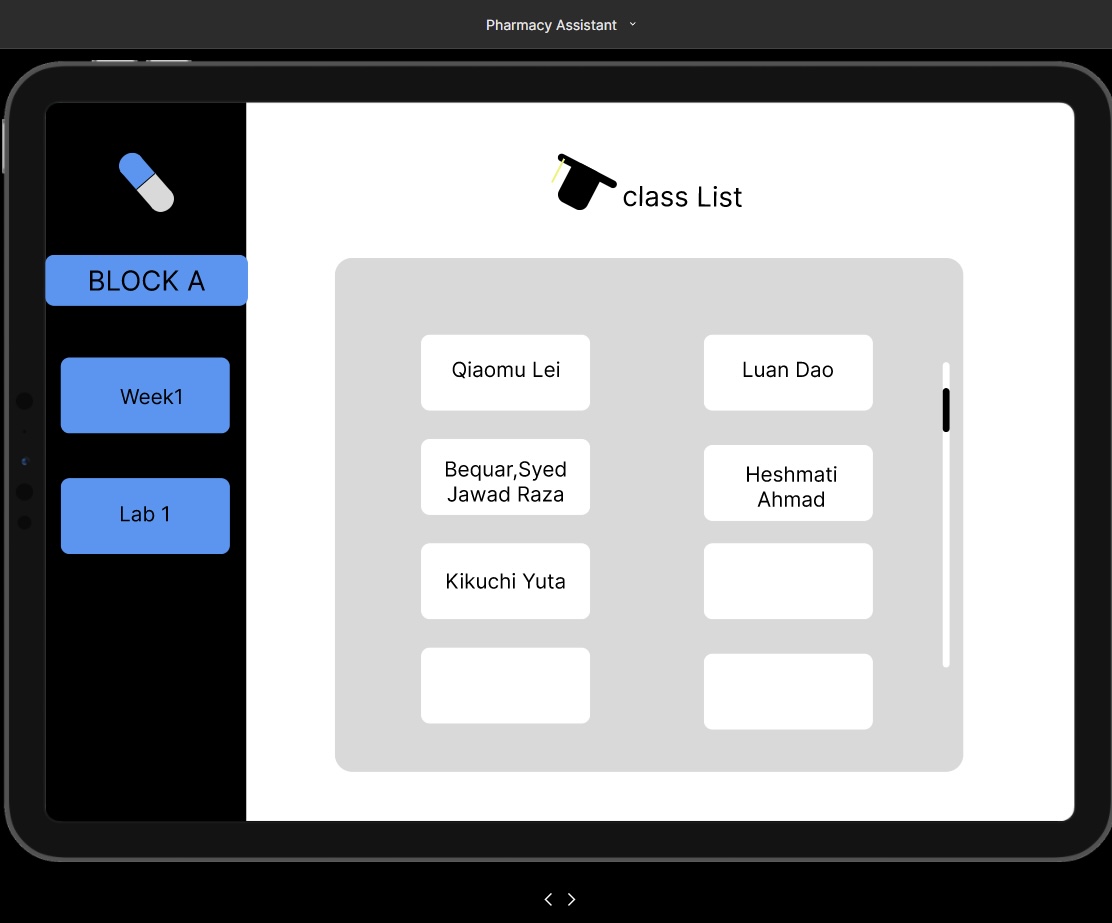


Figure Classlist page

**Class list page:**

After the teacher selects a lab, the lab tag will appear on the sidebar. All of the students' names are displayed on the class list page, and the instructor only needs to click the student tag to select the student.

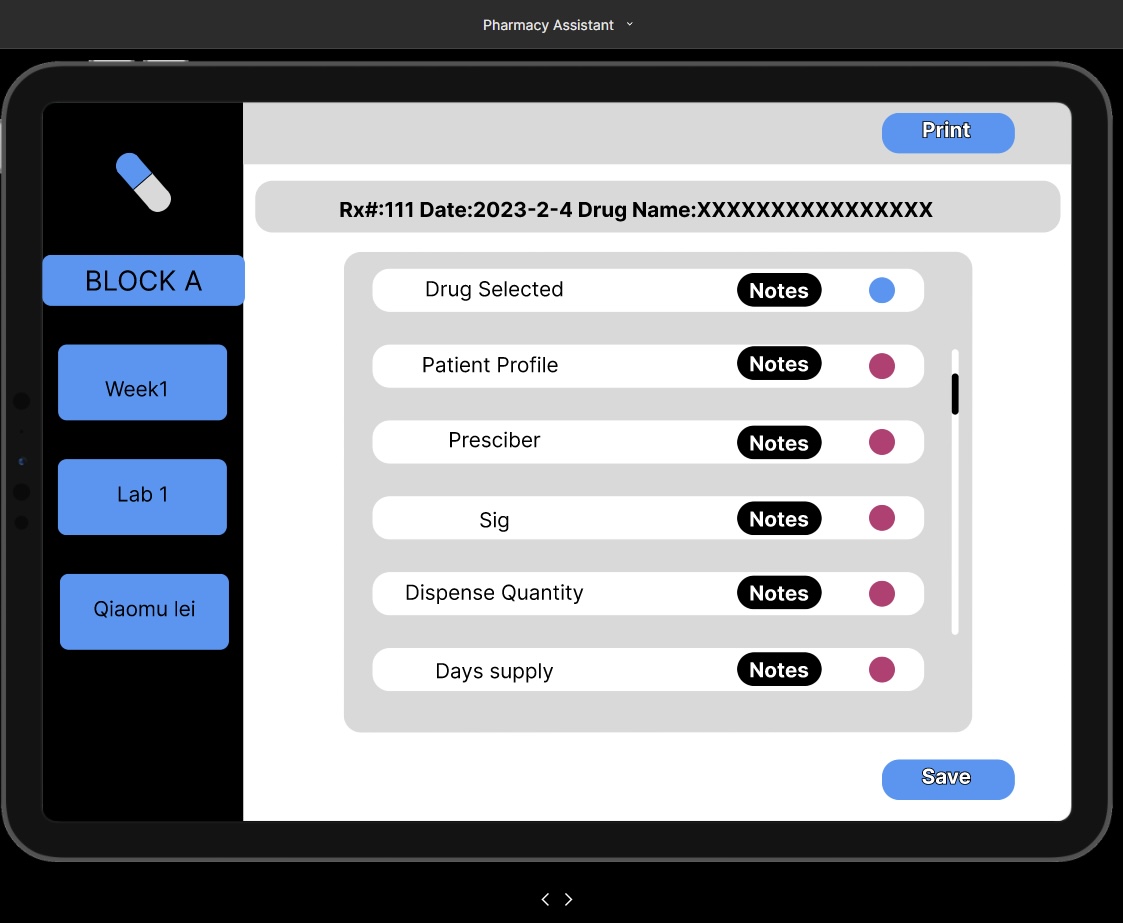


Figure Marking page

**Marking page:**

All the marking criteria are displayed on the marking criteria page, we design a small circle on the end of each criterion. The instructor needs to click the circle to mark each criterion, red indicates fail and blue indicates pass, instructor can click save button to save all the modification.

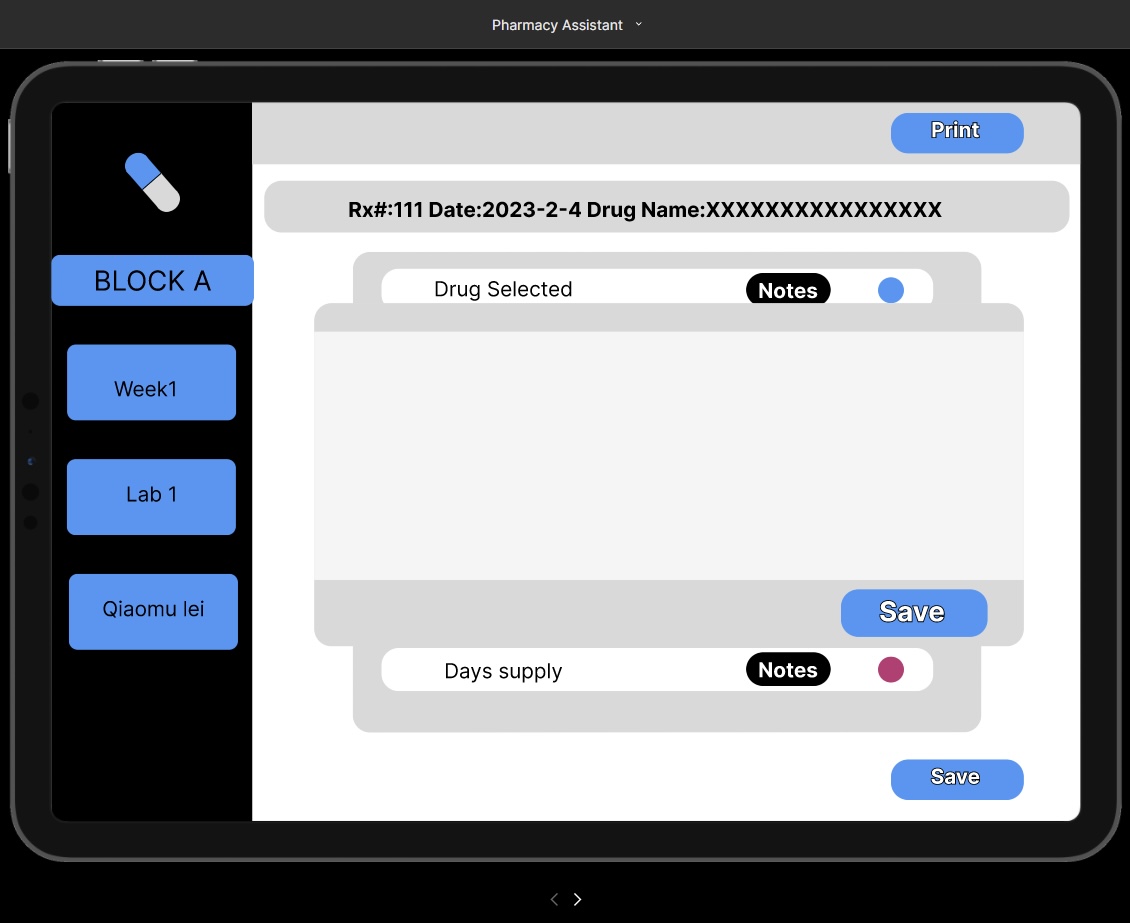


Figure Notes/Comments section

**Notes:**

To make comments for each criterion, the instructor needs to click the Notes button then the notes window will pop up, the instructor can save all the comments by clicking save button. After all the contents are saved the instructor can use the print button to print a report for each student.

**Additional information:**

The instructor simply needs to click the sidebar symbol to return to the main page. They can also click each tag to navigate back to the specific page.

# Data Storage and Persistence

Since our project is going to be developed as a web application, we decided to use a NoSQL database for implementing the data storage and persistence of our App. We have decided to go with Google’s Firebase as our database. Firebase offers a comprehensive suite of services for implementing and managing web-based applications. Listed below are some of the features that made us choose Firebase over other database solutions.

Real-time Database:

Firebase’s NoSQL database provides real-time data synchronization across multiple client devices. This feature is very useful for applications where multiple users are accessing and updating the data in the application simultaneously across multiple devices.

Authentication:

Firebase provides out of the box, ready to use authentication implementation using OAuth algorithm which can help us manage different users with varying degrees of permissions to read or edit the data seamlessly.

Scalability:

Firebase is built of scalability by its very design. It supports horizontal scalability instead of vertical scalability, meaning that if the user base grows, we do not need to buy bigger data servers, instead we can distribute the data load across multiple devices. This feature is perfect for today’s cloud driven environment where most applications are deployed using a cloud service.

Serverless Architecture:

Firebase comes equipped with serverless architecture design, essentially abstracting away sever management or any server related tasks from the developers, allowing us to focus on designing and building the core features and logic of our application.

Integration:

Firebase provides seamless integration with various front-end and back-end technologies such as React, Angular, Vue, JavaScript, Java, C++ etc.

Documentation and Community Support:

Firebase has extensive documentation and an active community which will enable us to find valuable resources and community support if we encounter any issues during development.

Below we provide step-by-step process that we will use for implementing our Firebase Database solution:

## Creating the project:

* + Go to Firebase console (<https://console.firebase.google.com/>) and click on Create a new project.
  + Configure authentication methods such as email/password or use of one the authentication service providers like Google or GitHub.

## Setting up the database:

* + Firebase uses Collection and Documents to store data in a JSON-like, tree hierarchical structure. We create the collections and documents for our project.
  + Implement database rules to secure our data, ensure the data is being stored in the correct format and to define who can read and write specific parts of our database.
  + We can also choose to create composite indexes to create complex queries for fetching the data if required.

## Back-end Integration:

* + Firebase provides SDKs to integrate with our server-side code for various programming languages, easily allowing us to perform basic CRUD operations.

## Front-end Integration:

* + Firebase SDKs also integrate seamlessly with various front-end technologies, allowing us to implement user authentication and interact with the real-time database.
  + Implement data binding and data synchronization so that changes made by one user are updated in real time for other users as well.

## File Storage:

* + Firebase also provides cloud storage for storing various files and documents such as images and PDFs. We can easily integrate it with our application for storing the graded labs of the students.

## Testing and Deployment:

* + We will be thoroughly testing our application for various database interactions and user authentications.
  + Deploy our front-end and back-end code and host in by using either the hosting services provided by Firebase or Vercel.

# Entity Diagram

A computer screen shot of a computer code

Description automatically generated

Figure Entity Diagram

# Class Diagram

A screenshot of a computer screen

Description automatically generated

Figure Class-based model of the grading system

**Sequence diagram**

## **Create/Select Block:**

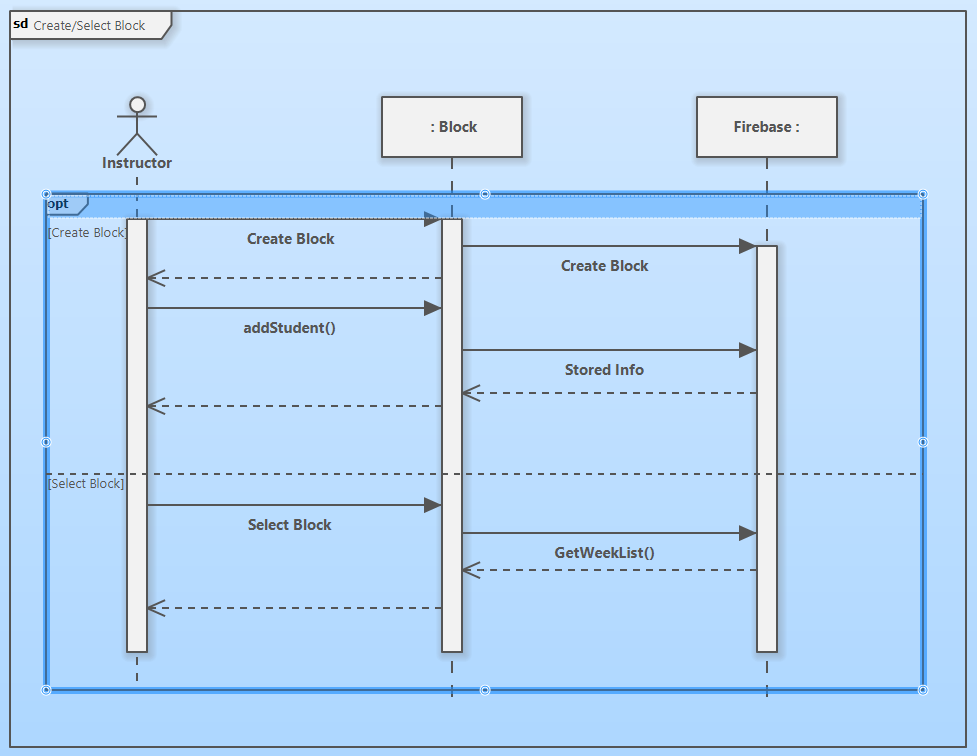


Figure Sequence Diagram of Creating/Selecting a Block

## **Create/Select Week:**

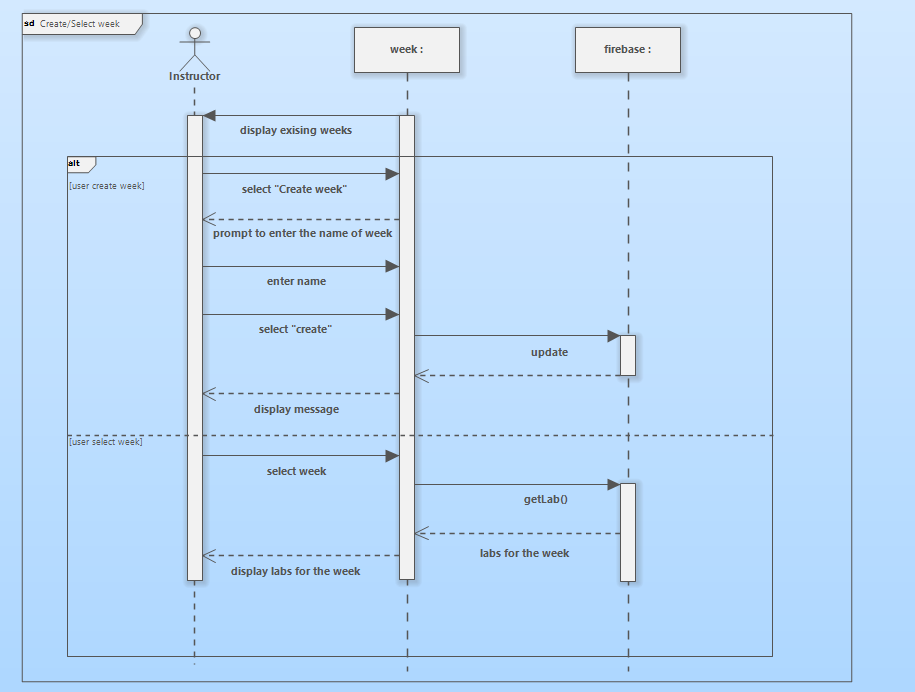


Figure Sequence Diagram of Creating/Selecting a Week

## Create/Select Lab:

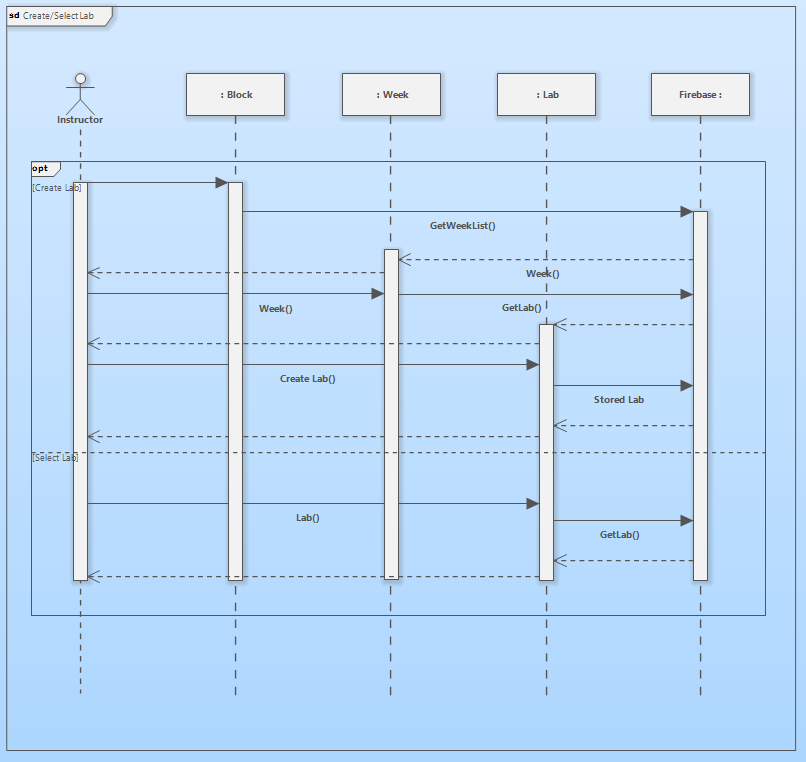


Figure Sequence Diagram of Creating/Selecting a Lab

## Grade lab for each student:

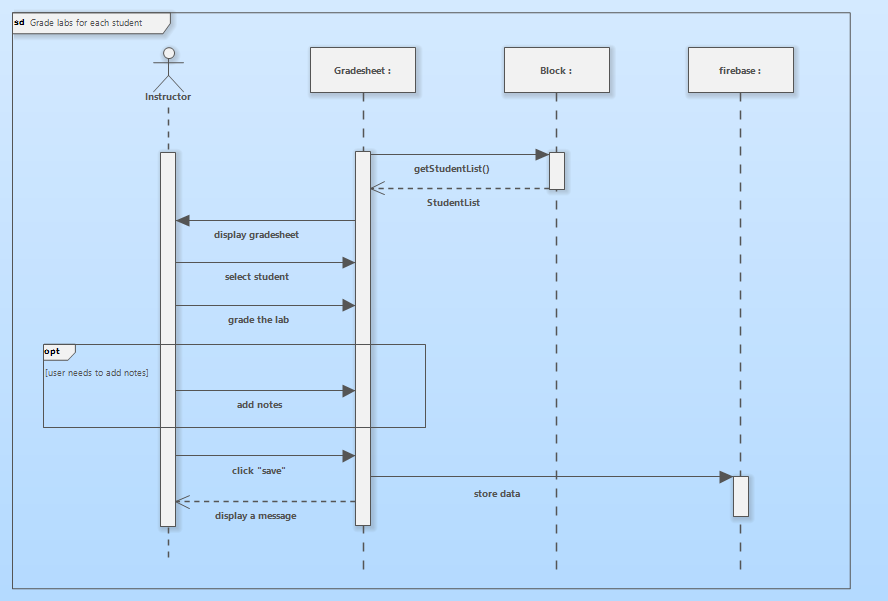


Figure Sequence Diagram of Grading a lab for a student

# **Activity diagram:**

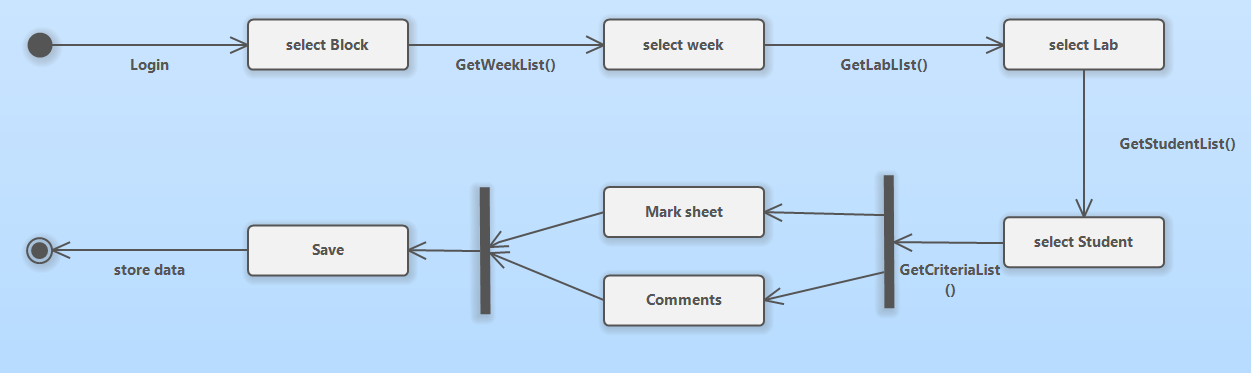


Figure Activity Diagram of Marking a Lab for a student

# **State Machine diagram:**

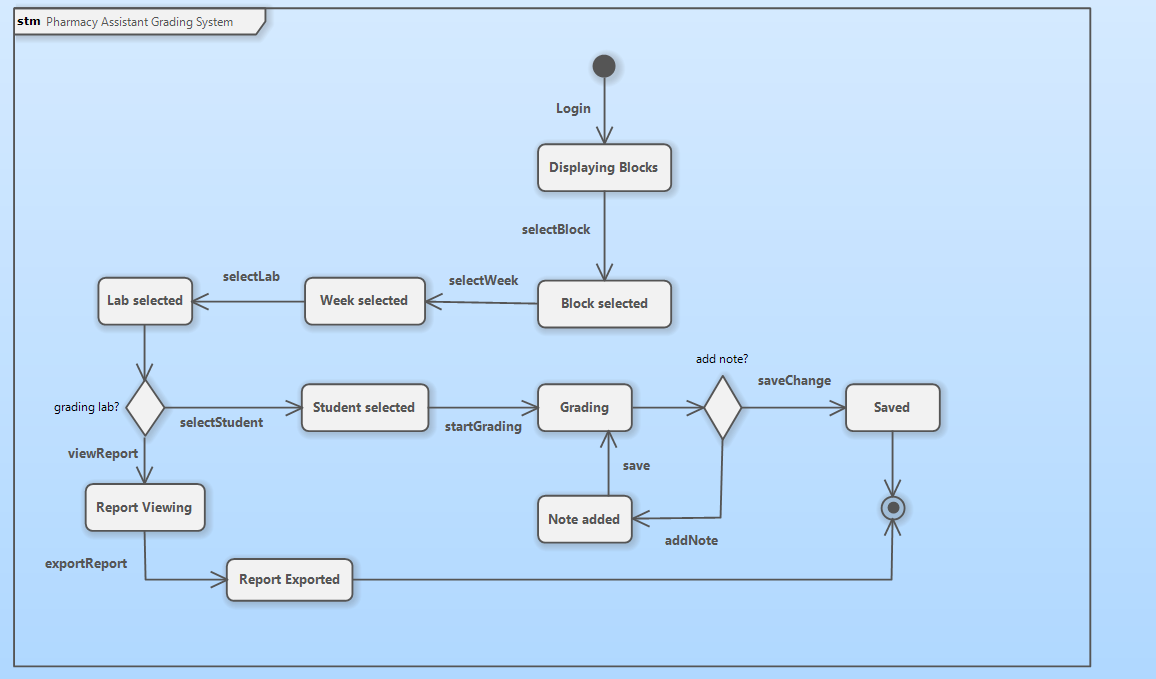


Figure State machine diagram of the system as it undergoes state changes during the process of marking/viewing a lab for one student.

System Architecture and Patterns:

System architectures and design patterns are both important concepts in software engineering. However, they function at different abstraction layers and serve distinct roles. In this section, the differences between these two concepts are explained, and the selected system architecture and potential design patterns that might be implemented in the system are discussed.

System architecture is considered the high-level structure of a software system. It provides a layout that defines the system's components, their interaction, and the environment in which the system operates. Moreover, it includes the hardware and software elements, and the relationships and interactions between them. System architecture is used to create a scalable and maintainable system that meets all business requirements.

In the context of system architecture, there are sometimes specific design problems that can be solved by recurring patterns. These patterns, known as design patterns, are reusable solutions to common problems in software design, and they do not address the overall system design. While system architecture defines the high-level structure of the software, design patterns address particular design problems and provide reusable solutions to these problems.

In this project, our group is going to solve the given problem by providing a web-based solution using Next.js, a full-stack framework built on top of React.

The system architecture for this solution includes five different layers:

## Frontend Layer (Client Side):

User Interface (UI): The UI of this web application is designed using React, leveraging Next.js for server-side rendering and route management.

## Backend Layer (Server-Side with Next.js API Routes):

API Layer: This layer provides the API routes within Next.js, which the frontend layer uses to communicate with the Firebase backend. The requests and responses between the frontend layer and the Firebase backend are handled by these routes.

Authentication: This layer is responsible for managing user access using Firebase Authentication.

Authorization: This layer defines the roles and permissions in the web application, ensuring that only authorized users can perform certain functions.

## Database Layer (Firebase Firestore):

Database Design: The Firestore database is modeled to store the needed data, including user accounts, grading templates, student blocks, labs, weeks, student records, and other necessary information.

Data Access Layer: This layer provides the functions to interact with the Firestore database.

## Integration Layer:

**Firebase Services:** All the additional Firebase services are utilized in this layer as needed.

**Infrastructure:**

**Deployment:** This Next.js web application will be deployed using Vercel.

**Potential Design Patterns:**

**Composite Pattern:** The Block class can be part of a composite pattern, as it might contain other blocks or elements.

**Factory Pattern:** The Lab class can be related to a factory pattern, as it might be responsible for creating instances of “Gradesheet” objects.

**Strategy Pattern:** The MarkingCriteria class could be part of a strategy pattern, as different instances of this class can be applied to a “Gradesheet”.

# Prototype

Github repository: <https://github.com/ahmadhmi/CPSY301-final-project-prototype>

Vercel live deployment: https://cpsy-301-final-project-prototype.vercel.app/

Explaining the structure of our JSON **data structure:**

Below we explain the structure of Data being used in our prototype:

We are using a Hardcoded JSON file in our program to fetch all the data we need to be displayed on the screen.

Our JSON file is an array of objects. Each object contains information about the various blocks that we have.

* Each object has three properties, namely, **blockName** , **listOfWeeks** and **students**.
* **blockName** is a string value containing the name of the block.
* **listOfWeeks** is an array of objects and each object contains information about the weeks for each block.
  + Each object has two properties, namely, **weekName** and **labs**.
  + **weekName** is a string value containing the name of the week.
  + **labs** is an array of string values that has the names of all the labs for that particular week.
* **students** is an array of string values that has the names of all the students in that particular block.

# **Work Breakdown Structure and Gantt Chart**

## **LOC-Based Estimation Table**

|  |  |
| --- | --- |
| Function | Estimated LOC |
| Non-functional, minimally viable user interface | 1000 |
| Client-side data processing | 500 |
| Server-side data processing | 1000 |
| Database management services | 1000 |
| Communication Helpers | 500 |
| Data presentation Helpers | 500 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Estimated lines of code | 4500 |

Assuming accuracy of the preceding:

5 people X 2.5 months = 12.5 person months (pm),

4500LOC / 12.5pm = 360 LOC/month (Necessary productivity)

## PNR Reconciliation

**t(o) = F x Effort ^ 0.33,**

**7.13 = 3.10 X 12.5^0.33,**

Optimal delivery time according to the PNR curve is 7.13 months

Team size = 12.5 / 7.13,

Team size = 2 people (1.75)

## **Division of Labor**

The roles listed below are assigned to the respective team members. These responsibilities are tentatively assigned and will be changed depending on member skills or project requirements. Members are allowed to assume more responsibility than assigned and are allowed to pursue and collaborate on other responsibilities with the other assignees if they fulfill their own responsibilities.

|  |  |
| --- | --- |
| Function |  |
| **Non-functional, minimally viable user interface** | Ahmad, Syed, Qiaomu, Yuta, John |
| **Client-side data processing**  Making the user interface functional and processing the information from forms to create the necessary objects for storage | Ahmad, Syed, Qiaomu, Yuta, John |
| **Server-side data processing**  Taking the information received from API calls and processing them for storage | Syed, John |
| **Database management services**  Creating helpers that interact with the database | Yuta, Ahmad, Qiaomu |
| **Communication Helpers**  Creating the APIs that call | Syed, John |
| **Data presentation Helpers**  Taking the information received from API calls on the front-end and displaying it to the user on the interface | Yuta, Qiaomu |
| **Authentication and Authorization** | Syed, John |
| **Database Management**  Creating and configuring the database | Yuta, Qiaomu, Ahmad |

## Gantt Chart

A screenshot of a computer

Description automatically generated

[PDF Included separately]

Conclusion  
  
**In conclusion, this document outlines the planning and communication phases for the Pharmacy Assistant Grading System.**

* The project aims to replace the manual grading process used by **the instructors of the Pharmacy Assistant program at the Southern Alberta Institute of Technology with a web-based solution.**
* The proposed system allows instructors to create labs, grade student work, and export reports.
* Administrators can manage instructor accounts and assignments.

The thorough analysis and use case descriptio**ns provide a clear understanding of system functionalities. The class diagram represents the structure of the system and their relationships, encapsulating the system's essential data and functionalities.**

The project is now ready to move into the developm**ent phase, offering an efficient solution for grading pharmacy assistant lab work.**

# Appendix A: Team Constitution

# Team Constitution

## Team Name:

We are Team rayJS.

## Group members:

**Luan “John” Dao**

John is a recent high school graduate who found coding fun and is now taking the Software Development course at SAIT (Southern Alberta Institute of Technology) to further pursue knowledge in that field. He has had a brief work history in IT (Information Technology) and is acquainted with the fundamentals of programming, JavaScript, CSS, and the SDLC. He is also well versed in writing, time management, and cooperation skills.

**Ahmad Heshmati**

Ahmad is a third semester student at SAIT taking Software Development Diploma program. He has a background in Mechanical Engineering, giving him the engineering and problem-solving skill set needed. Also, He is knowledgeable about software analysis and UML diagrams which can be beneficial. Moreover, his coding knowledge in programming languages and frameworks like Java, JavaScript, C# and React will come in handy. On top of that, he has been working in a management position in retail for 2 years, making him able to contribute to managing the project.

**Qiaomu Lei**

Qiaomu Lei is a third semester student in the Software development diploma at SAIT. He was working in the CNC industry and is now trying to start a new career in the IT industry. He has foundational knowledge in Java, JavaScript, C#, SQL, and CSS in addition to his skills in cooperation and management skills. He is passionate and motivated to contribute to this project.

**Syed Jawad Raza Baquar**

Syed is a current third semester student in the Software development diploma program at SAIT. After completing his bachelor's degree in industrial engineering and management he decided to join the software industry and worked as a software engineer for over 3 years before deciding to move to Canada to pursue his diploma. For this course, some of the skills that he can bring to the table are System analysis and UML designing, coding using ReactJS, NodeJS and Java, and his experience working as a productive member of a team.

**Yuta Kikuchi**   
Yuta is an alumnus of Meijigakuin University, having studied in the faculty of International Studies. He is currently pursuing software development at SAIT. He has fundamental knowledge of some Frond-end languages such as HTML/CSS and JavaScript and is familiar with CASE tools like Software Ideas Modeler to create UML diagrams. He desires, as a career after graduation, a job relevant to web development.

## Team Goals

# 

In the duration of this course, rayJS aims to develop a deeper understanding of the Software Development Life Cycle. We hope to familiarize ourselves with the analysis and design process that comes with every software project and the accompanying skills such as requirements engineering, selecting an appropriate process model, and gain a deeper familiarity with the five framework activities.

Over the course of this project, rayJS would like to gain experience in undertaking a small-scale, real-world software project, such as learning how to communicate effectively with clients, translating their needs to software solutions, and working effectively as a team.

## Intellectual Property Statement

The resulting application or work products of the project belongs to all members of Team rayJS and members are permitted to use them as fit for their respective portfolios. The application and work products are the property of Team rayJS and do not belong to the client.

# **Roles:**    **Leading the team: Ahmad Heshmati, Luan Dao**

As the team leaders, they are responsible for providing guidance, direction, and motivation to team members. Team leaders need to delegate tasks among the team members fairly and properly. Additionally, they are responsible for outlining the deadlines and the overall strategy for completing the tasks. 

### Editing the final document: Luan Dao

As the final document editor, they are responsible for ensuring the quality of the textual content of the course’s graded assessments and the accuracy, coherence, and fidelity to the rubric, of the contents of the graded assessments.

Formatting the final document: Syed Jawad Raza Baquar

They handle layout, typography, page elements, images, and formatting consistency for a professional and appealing presentation

Ensuring the document complies with IEEE format:Qiaomu Lei

Their role is to ensure that the final project document is visually polished, consistent, and follows formatting guidelines.

Keeping the team on track with deadlines:Ahmad Heshmati

They are responsible for setting clear deadlines, creating timelines, regularly communicating, prioritizing tasks, and monitoring the progress of the project.

Ensuring the document meets the report guidelines:Yuta Kikuchi

It involves going through the official guidelines for writing the reports and making sure the team’s final report adheres to the rules and best practices as stated in the guidelines.

## Division of labor

Other than the roles explicitly listed in the preceding section, all members are expected to contribute to any work that is being done. Some may take on more of a leadership role for certain activities, depending on their expertise, but members are expected to participate in any ongoing assignments, activities, tasks, and group-work periods.

## Norms and Expectations

**I.** Regarding meetings held outside of class time

There is no set schedule for meetings that are to be held outside of class time. Meetings are called as necessary for situations similar but not limited to:

1. An incomplete graded assessment that is due within the next two business days.

2. A sudden meeting with a stakeholder has been arranged.

3. A member is not responding to direct correspondence.

4. Any extraordinary circumstance that most team members deem urgently needing a solution.

Team meetings, held outside of class time, are implicitly understood to be an hour long and held on the Teams video-meeting service, unless otherwise noted.

**II.** Regarding communication between team members

Members are expected to put any official and noteworthy communication on the “Software Projects” Teams channel. Most standard communication should also be held there, and it is encouraged that informal messages be hosted on a third- party group chat.

**III.** Regarding file hosting

All files and work products produced as a direct or indirect result, especially those that are of great relevance to the larger project at hand, of any graded assessment are to be hosted in the files section of the “Software Projects” Teams channel.

## Conflict Resolution Plans

1. **A team member fails to attend meetings.**

As a rule, a notice must be sent to the Discord server beforehand which we use as our communication tool. Failing to attend meetings without valid excuses is considered a minor infraction. The members are going to have one-on-one conversations with the team leader. In case the event happens repetitively, a group meeting warning will be held, and we decide which degree of offense it conflicts with.

1. **A team member fails to meet deadlines.**

Since we work as a team, it is crucial that we do not fail to meet deadlines, so we set aside at least 24 hours before the due date to ensure that the team is able to complete the work in advance and ensure it meets the team’s requirements for the submission. If a team member is not able to submit their portion of the work to the team, it is a major infraction. If it is the first time, the member will be given a warning which is a first offense. For the repetition of the same incident, we will follow offense rules which are established in a later section.

1. **A team member is doing inadequate quality work.**

An essential aspect of this class is that every team member understands the course contents and they make this project a valuable learning opportunity. Before considering offense rules, we will aim to provide support in order to prevent leaving the members behind in terms of the team’s progress. For that reason, we will have a conversation with the members in a group meeting to become informed of the present circumstances. However, if there is no recognizable effort or progress towards improvement, it will be dealt with based on offense rules.

1. **A team member fails to communicate**

Failure in communication will be handled by offense rules which are stated in a later section.

1. **You have different levels of commitment to the project**

Although there are no obvious offense rules to cope with this situation, we are able to utilize a group meeting as an occasion for commitment alignment. After evaluating the degree of commitment to the project for each member, we will ensure the objectives of the project and adjust the level of commitment so that the imbalance is reduced.

1. **You have personality conflicts.**

We will attempt to improve communication to reveal issues and true thoughts behind the conflict by providing an opportunity for openly sharing opinions. Depending on the situation, a new set of communication rules possibly be established during a group meeting. If we find that the situation is incompatible with our guidelines for respectful behavior or if it becomes uncontrollable any further, we will address it in accordance with our offense rules and, when necessary, inform the instructor.

1. **A team member feels that they are doing more work than other members.**

As a team meeting is held regularly, we will clarify roles and responsibilities for each member in each phase of the project. Moreover, if a workload is not divided equally like in this case, we will discuss about apportion of the workload to relieve the members from the amount, and new charges of the responsibilities.

1. **A team member is doing less work than other members.**

In addition to reconsideration of task allocation, we will confirm whether the members have difficulty keeping up with a class, managing time, or anything that causes the situation. Ultimately, we assign the members to have additional work since there should be other members who are in charge of heavy workload.

## Punishment for Offences

Anyone GUILTY of committing misconduct as listed below and the action of the offender matches the corresponding descriptions will be reprimanded with the stated sanctions.

How offences will be CHARGED and how SANCTIONS will be APPLIED:

List of offences in descending order of severity and, therefore, precedence

1. Unreachable

2. Plagiarism

3. Disrespectful Behaviors

4. Failure to Communicate Progress

5. Failure to be on Task

6. Incomplete work by the given deadline

7. Using coercion or colluding to avoid responsibilities:

Only when the victim chooses to formally accuse the suspected offender will the offenses be charged against the offender.

Official accusations may be made at any time (including when the current response period has ended).

Any of the appointed team leaders will receive complaints, and they will launch an investigation.

All offenses of omission, i.e., negligence, (note: not commission) will be absolved from the offender if it is later discovered that it was not preventable (Acts of God or Force Majeure).

If the actions of the offender match two or more listed offences, the more severe sanction(s) will be applied, and a tally of the more severe offence will be recorded. e.g., Someone has plagiarized in the past and holds a record as a first-time offender in the plagiarism category. Now, if they are guilty of being disrespectful and plagiarism for the second time, a second order plagiarism sanction will now be applied to the offender, and they will be recorded as a second-time offender in the plagiarism category. Plagiarism holds precedence over disrespectful behavior.

1. **Unreachable**
2. First Offence: Warning
3. Second Offence: The offender is monitored and required to accomplish the task or resolve the issued delegated within the missed message
4. Third Offence: 20% Deduction off the assignment grade
5. Fourth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
6. **Plagiarism**
7. First Offence: Warning
8. Second Offence: Deduct 10% off of the current assignment and a discussion with the instructor about the incident in question.
9. Third Offence: Deduct 20% off of the current assignment and a discussion with the instructor about the incident in question and report all evidence to instructor.
10. Fourth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
11. **Disrespectful Behaviors**
12. Frist offence: WARNING MANDATORY APOLOGY to the victim of such abuse. Failure to apologize will result in a second offence.
13. Second Offence: WRITTEN WARNING: REMOVED FROM ANY DISCUSSION for the day on which the offence occurred and the following 2 meetings.
14. Third Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
15. **Failure To Communicate Progress**
16. Frist offence: WARNING, Review of Progress and Discuss with Offender to identify and provide solutions.
17. Second Offence: 5% Deduction off the assignment grade.
18. Third Offence: 10% Deduction off the assignment grade.
19. Fourth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
20. **Failure to be on Task (Leading discussions and/or project tasks into off task topics or implementation in the project. Not finishing one’s responsibilities by a reasonable or pre- arranged time)**
21. First Offence: Warning
22. Second Offence: Warning
23. Third Offence:  RESPONSIBILITIES DOUBLED relative to fellow members’ workload for the current assignment
24. Fourth Offence: 5% Deduction off the assignment grade.
25. Fifth Offence: 10% Deduction off the assignment grade.
26. sixth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
27. **Incomplete work by the given deadline**
28. First Offence: Warning
29. Second Offence: 5% Deduction off the assignment grade.
30. Third Offence: 10% Deduction off the assignment grade.
31. Fourth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.
32. **Using coercion or colluding to avoid responsibilities:**
33. First Offence: WARNING Offender must fulfill their own responsibilities and assume those of the victim.
34. Second Offence: Offender will have to do double the amount of work.
35. Third Offence: 10% Deduction off the assignment grade.
36. Fourth Offence: Team needs to vote to expel someone who is guilty with 4 out 5 majority.

## Contract Conclusion

In conclusion, we can say that our team rayJS has team members with diverse backgrounds and skills and strengths, but we are all dedicated towards a common goal to deepen our understanding of the software development life cycle and improving our skills as programmers and software engineers.

We have clearly established the roles and responsibilities for each team member to ensure efficient project management and have unambiguously stated that the intellectual property for anything produced by rayJS will be completely owned by rayJS and we can use it for any purpose in the future such as including it in our portfolios to showcase our skills.

To avoid any potential conflicts for the duration of the project, we have clearly outlined norms and expectations for various friction points such as attending team meetings, communication expectations and file hosting to name a few. We also have an elaborate and comprehensive conflict resolution plan which covers various scenarios like attendance, deadlines, work quality, communication issues, commitment levels, personality conflicts and workload resolution.

In the event of misconduct, we have clearly defined offense rules and corresponding punishments with the consequences getting more severe for repeated offences.

With our well-defined team roles and clearly structured communication channels and conflict resolution framework we are well positioned to succeed in our group project. Our diverse skill sets and commitment to learning and doing well in our project will help us to succeed in our endeavours.

However, potential conflicts might rise due to repeated violations and meeting the group expectations by some members. This might lead to disruption in our project’s progress and completion. Ensuring that all members remain engaged and committed towards the completion of the project will be essential to avoid such pitfalls.

We can say that team rayJS is well positioned to succeed in our software development journey with our robust constitution and shared commitment to success.

Ahmad Heshmati \_\_\_\_\_\_\_\_\_\_*Ahmad Heshmati*\_\_\_\_\_\_\_\_\_\_

Luan Dao \_\_\_\_\_\_\_\_\_\_Thieu Luan Dao\_\_\_\_\_\_\_\_\_\_

Yuta Kikuchi \_\_\_\_\_\_\_\_\_\_\_\_Yuta Kikuchi\_\_\_\_\_\_\_\_\_\_\_

Syed Jawad Raza Baquar \_\_\_\_\_\_*Syed Jawad Raza Baquar*\_\_\_\_\_\_\_  
Qiaomu Lei \_\_\_\_\_\_\_\_\_\_\_\_\_Qiaomu Lei \_\_\_\_\_\_\_\_\_\_\_\_\_