24/11/2023

Cheuk Chun Yu [sxdn64]  
Chun Suet Ivy Ngai [tkvl77]  
Dylan Chua Yi Xuan [fwdc83]  
Kwan Ling Megan Law [wthj54]  
Phichayapa Soranaraksopon [wfkw57]  
Raine De Asis [cfbn68]

A drawing of a building

Description automatically generatedA logo with black text

Description automatically generated

Academic Tracker

Requirements Specification for Elemore Hall School - Group 5

Table of Contents

[1 - Introduction 2](#_Toc150999038)

[1.1 - Overview and Justification 2](#_Toc150999039)

[1.1.1 Purpose 2](#_Toc150999040)

[1.2 - Project Scope 2](#_Toc150999041)

[1.2.1 - Problems 2](#_Toc150999042)

[1.2.2 - Solutions 2](#_Toc150999043)

[1.2.3 - Stakeholders and Users 2](#_Toc150999044)

[1.2.4 - Future Features 2](#_Toc150999045)

[1.3 - System Description 3](#_Toc150999046)

[2 - Solution Requirements 4](#_Toc150999047)

[2.1 - Functional requirements 4](#_Toc150999048)

[2.2 - Non-Functional requirements 8](#_Toc150999049)

[2.3 - Risks and Issues 10](#_Toc150999050)

[2.3.1 - Risks Table 10](#_Toc150999051)

[2.3.2 Risks Matrix 12](#_Toc150999052)

[3 - Project Development 13](#_Toc150999053)

[3.1 – Development approach 13](#_Toc150999054)

[3.1.1 Comparison with Other Methodologies 14](#_Toc150999055)

[3.1.2 Confirming Our Path Forward 14](#_Toc150999056)

[3.2 – Project Schedule 14](#_Toc150999057)

## 1 - Introduction

This document outlines the **Requirements Specification** for the Classroom Monitor project, organised into three sections. The **Introduction** provides a project overview, purpose, scope, and system description, including a comparative analysis of similar products. The **Solution Requirements** section details functional and non-functional requirements, along with potential risks. The closing section addresses **Project Development**, outlining the development approach and long-term project schedule.

### 1.1 - Overview and Justification

We are undertaking this project under the direction of **Mr. Michael Hunter**, who shall henceforth be referred to as ‘the client’. The client has tasked us with creating an **Online tracking system, the Classroom Monitor,** in which the system can generate gradebook for students by their academic performance and users can track the students’ gradebook online as their old version is going to expire soon and they want to add some new elements to the system. There are several key project goals:

1. Create a safe platform to save the information of the students.
2. Analyse students’ data and generate gradebooks according to the behaviour of the students.
3. Only allow people who have access to log in to the website.

#### 1.1.1 Purpose

**The website that we are planning to create would serve doubly as a way for students to track their academic progress in each different subject, and for the school to update grades and monitor the progress of each student over time and intervene when necessary.**

### 1.2 - Project Scope

#### 1.2.1 - Problems

Michael, our client, urgently needs a replacement for their current academic system, set to be discontinued by July 2024. They seek an all-encompassing academic tracking and monitoring system, retaining familiarity for users at Elemore Hall while incorporating specific features missing in their previous tracker.

#### 1.2.2 - Solutions

To meet Elemore Hall's academic tracking needs, we will develop a user-friendly web interface with key features like student profiles, grading, attendance tracking, and progress reporting. Additional functions include extracurricular activity tracking, tutoring services, and a parent/guardian portal. Prioritising data security and scalability, we'll integrate with educational tools, offering training, support, testing, and user feedback. Deployment, maintenance, and evaluation will align with set timelines and budgets.

#### 1.2.3 - Stakeholders and Users

Elemore Hall's academic tracking system caters to diverse stakeholders—students (elementary to higher education), parents, faculty, and administrators. Students and parents track performance and access resources, teachers and administrators manage academic tasks, and the IT department ensures system functionality and security. Elemore Hall defines requirements for this efficient academic tracking platform, benefiting all users.

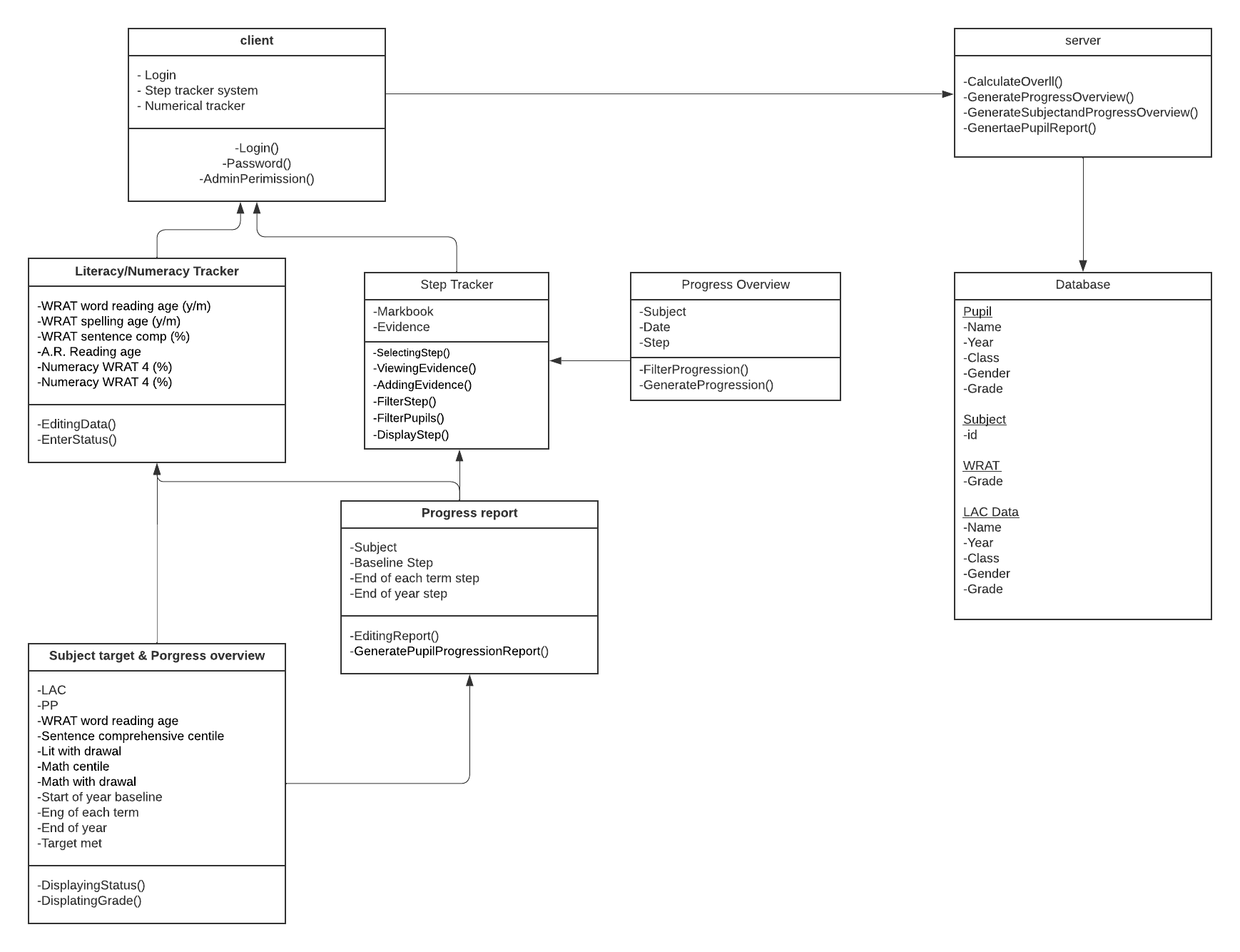
#### 1.2.4 - Future Features

Our academic tracking webpage offers secure authentication, a user-friendly interface with automated grades and real-time updates. It includes student portfolios, attendance tracking, and a communication platform for engagement. Analytics tools enable trend identification, mobile responsiveness, and Learning Management System integration. Customizable reporting, gamification, and feedback mechanisms ensure a user-centric approach. Integration with external data sources and accessibility features enhance functionality, guided by continuous stakeholder feedback for optimal satisfaction.

### 1.3 - System Description

## 2 - Solution Requirements

This is a UML Diagram to represent the flow of the system, the independent items in the system, and how they’re interconnected.



### 2.1 - Functional requirements

The functional requirements, categorized into 5 main features: Mark book, Progress Overview, Literacy/Numeracy Tracker, Pupil Progress Report, and Subject Target and Progress Overview, are shown below.

FR1 - Login

The user is asked to sign into their account using their school email and password on the homepage.

|  |  |
| --- | --- |
| ID, type, and title | FR1 Login - Confirm email and password |
| Description | The user shall enter their school email and password for accessing the software. |
| MuShCo and Priority | Must Have - High |
| Dependencies | N/A |
| Expected results | The entered email and password will be matched and confirmed by the database that they are valid.  The software navigates to the homepage. |
| Exception handling | An error will be displayed if the entered email and password are invalid.  An error will be displayed if the format of the email is invalid. |

FR2 - Admin Permission

The software provides a page for editing the level of permission of each user on a page that is only visible and accessible by the admin.

|  |  |
| --- | --- |
| ID, type, and title | FR2 Admin permission - Edit users' permissions |
| Description | A table view page that displays users' names, and level of permissions. Admin can add new users, remove existing users, and edit the level of permissions of users. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR1 |
| Expected results | The updated permissions shall grant each user access permission to different pages. |
| Exception handling | An error will be displayed if the admin tries to add an existing user to the system. |

FR3 - Mark Book

The software provides a grid page for users to view and edit the academic performance of students.

|  |  |
| --- | --- |
| ID, type, and title | FR3.1 Edit statement banks |
| Description | The users can add or remove statements which are task metrics to evaluate the performances of students. |
| MuShCo and Priority | Should Have - Medium |
| Dependencies | N/A |
| Expected results | The updated statement banks would be displayed on the main page of the Mark Book |
| Exception handling | If users did not save the results, the statement banks should not be updated. |

|  |  |
| --- | --- |
| ID, type, and title | FR3.2 Select steps |
| Description | The users can select the progression of each step according to the performance of the student. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3.1 |
| Expected results | The step tile will change on users’ click, changing its value and color indication of different step progressions S(ecure), D(eveloping), E(merging), N (ot yet). The steps shall initialize as N(ot yet). |
| Exception handling | The step will be disabled and displayed in grey if the student does not participate in it. |

|  |  |
| --- | --- |
| ID, type, and title | FR3.3 View evidence |
| Description | A tab to switch from viewing steps to viewing the number of evidence corresponding to each step. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3.2 |
| Expected results | The page shall have the same layout as the steps and change from displaying the step progressions to displaying the number of evidence to support the progressions that the student gained while maintaining the color of the step. |
| Exception handling | When the step is disabled, the corresponding evidence tile will also be disabled. |

|  |  |
| --- | --- |
| ID, type, and title | FR3.4 Add evidence log |
| Description | Add descriptions, files, and images as supporting evidence of the progression that students have. The log consists of the date, location, evidence type, and the evidence. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3.3 |
| Expected results | When an evidence log is added, the number on the corresponding tile shall update and the uploaded data shall be saved into the system. |
| Exception handling | If users did not upload the log, the evidence shall not be recorded. |

|  |  |
| --- | --- |
| ID, type, and title | FR3.5 Filter steps and pupils |
| Description | Users can use the search bar and filter menu to filter results using input or selecting specific groups, curricula, subjects, or other aspects. |
| MuShCo and Priority | Could Have - Low |
| Dependencies | N/A |
| Expected results | A filtered result of Mark Book will be displayed |
| Exception handling | A custom error message will be displayed if no data entries satisfy the filter |

|  |  |
| --- | --- |
| ID, type, and title | FR3.6 Display overall step description |
| Description | The overall step will be displayed after calculating each student’s step. Worked out by S=1 D=0.5 E=0.05 N=0.005 - the sum of values divided by the number of statements in each step x 100 – rounded up to the whole number. The number will be fitted into S+ to E- boundaries |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3.2 FR3.5 |
| Expected results | An accurate overall step is displayed after summing up all the steps that students have. |
| Exception handling | Overall step will display “N/A” if the student does not have any records of steps. |

FR4 - Progress Overview

The software provides a page that shows the progress overview of the specified group, subject, and time period.

|  |  |
| --- | --- |
| ID, type, and title | FR4.1 Filtering progression |
| Description | Users can use the drop-down menu to select subjects, time scroll to select a time period, and menu to select groups of students to analyze. |
| MuShCo and Priority | Must Have - High |
| Dependencies | N/A |
| Expected results | A parameter for generating the progress overview graph will be constructed. |
| Exception handling | A custom error message will be displayed if the parameters are out of range. |

|  |  |
| --- | --- |
| ID, type, and title | FR4.2 Generate progression overview |
| Description | A table, line graph, or bar chart will be generated based on the user’s choice displaying students' performance across time periods, which can be exported and shared as a pdf. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3 |
| Expected results | A table or chart will be displayed |
| Exception handling | A custom error message will be displayed if one of the necessary fields is missing. |

FR5 - Literacy/Numeracy Tracker

The software provides a grid page for users to view and edit the literacy and numeracy data of students.

|  |  |
| --- | --- |
| ID, type, and title | FR5.2 Literacy/Numeracy Tracker - Edit data |
| Description | Users can enter students’ Wide Range Achievement Test (WRAT) and reading age results in the grid table. |
| MuShCo and Priority | Must Have - High |
| Dependencies | N/A |
| Expected results | A grid table consisting of all students’ WRAT and reading age results within 5 years. |
| Exception handling | A custom error message will be displayed if the wrong type of data is entered. |

|  |  |
| --- | --- |
| ID, type, and title | FR5.1 Enter LAC / PP status |
| Description | Users can enter students’ LAC (looked-after children) PP (pupil premium) status by selecting and coloring the corresponding tiles. |
| MuShCo and Priority | Must Have - High |
| Dependencies | N/A |
| Expected results | Students’ LAC / PP status will be displayed by colored tiles in the overall. |
| Exception handling | A custom error message will be displayed if the wrong type of data is entered. |

FR6 - Pupil Progression Report

The software provides a page for displaying an annual progression report for each student.

|  |  |
| --- | --- |
| ID, type, and title | FR6 Generate Pupil Progression Report |
| Description | An annual report will be generated based on the selected student displaying all subject results, attendance, and end-of-year target steps. Users can enter other information in the report. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3 |
| Expected results | A report will be displayed |
| Exception handling | A custom error message will be displayed if necessary data is not entered by the user. |

FR7 Subject Target and Progress Overview

The software provides a page for displaying subject targets and progress overview data.

|  |  |
| --- | --- |
| ID, type, and title | FR7 displaying WRAT data and steps |
| Description | WRAT data and steps are fetched from the mark book and Literacy/Numeracy tracker and displayed. |
| MuShCo and Priority | Must Have - High |
| Dependencies | FR3 FR5 |
| Expected results | Accurate WRAT data and steps are fetched and displayed in the overview |
| Exception handling | A custom error message will be displayed if the data cannot be fetched properly. |

### 2.2 - Non-Functional requirements

The non-functional requirements are divided into five categories based on their influence on our program, including **Efficiency requirement, Usability requirement, security requirement, scalability requirement,** and **ethical requirement.**

NFR1 Efficiency requirements

Efficiency requirements specify and constrain the storage space limitation and processing time of the software.

|  |  |
| --- | --- |
| Type | NFR1.1 Storage space limitation |
| Metrics | Monitor and manage storage space consumption to stay within defined limits of 10GB. |
| Security | Restrict access to limited roles to ensure data security. |
| Constraints | Ensure data storage and access control align with data privacy regulations and security standards. |

|  |  |
| --- | --- |
| Type | NFR1.2 Processing time |
| Metrics | Measure the system's response time for various operations and monitor the delay or waiting time experienced during processing in the 20s. |
| Security | Use encryption for data at rest and in transit to protect sensitive information during processing |
| Constraints | Ensure that the processing time aligns with the available hardware resources and doesn't exceed system capabilities. |

NFR2 Usability requirements

Usability requirements specify and constrain the utility, flexibility, and robustness of the software.

|  |  |
| --- | --- |
| Type | NFR2.1 Utility and convenience |
| Metrics | The user can use it even if they don’t have IT skills |
| Security | None |
| Constraints | Users should find the navigator and other instructions simply accessible. |

|  |  |
| --- | --- |
| Type | NFR 2.2 Flexibility |
| Metrics | The user can use the software under OS and Windows |
| Security | The login process should be protected by the third-party verified software, for example, Microsoft Authenticator. |
| Constraints | It should be used to test and identify the user's operating system and to set all of the software to the appropriate level. |

|  |  |
| --- | --- |
| Type | NFR 2.3 Robustness |
| Metrics | The system follows ACID (Atomicity, Consistency, Isolation, Durability) properties. This ensures that data changes are not lost, and transactions are handled reliably. |
| Security | None |
| Constraints | Using a method or mechanism to catch errors related to data. |

NFR3 Security requirements

Security requirements specify and constrain data security and account security.

|  |  |
| --- | --- |
| Type | NFR 3.1 Data security |
| Metrics | Any third party should not be able to view private user data and not be able to access it. |
| Security | Ensure that only authorised personnel have access to the systems and tools used for data security. Implement access control policies and regularly review and update user permissions. |
| Constraints | Using third-party services or organisations are constrained by the security practices and measures of those third parties. |

|  |  |
| --- | --- |
| Type | NFR 3.2 Sign-in security |
| Metrics | Plug-in functions and software should protect the sign-in process. |
| Security | None |
| Constraints | limitation of password entering time, Confirmation through email when logging in |

NFR4 Scalability requirements

Scalability requirements specify and constrain support for large data.

|  |  |
| --- | --- |
| Type | NFR 4.1 supports large data |
| Metrics | The system is capable of potentially handling higher volumes, whether it be a greater number of users, increased throughput, or larger datasets. |
| Security | Managing and securing large datasets can be particularly challenging, as there may be increased risks associated with data breaches, data loss, and other security vulnerabilities. |
| Constraints | Ensuring that the system can scale seamlessly as data volumes grow is a constraint that requires careful planning and infrastructure design. |

NFR5 Ethical requirements

Ethical requirements specify and constrain data protection.

|  |  |
| --- | --- |
| Type | NFR 5.1 Data protection |
| Metrics | Collect the data with consent |
| Security | Only users holding the role “Customer advisor” or “Supervisor” can update the customer record |
| Constraints | Organisations must comply with various data protection laws and regulations |

### 2.3 - Risks and Issues

In this section, 5 potential risks and issues are assigned into three categories based on their influence on our development process, including the technical risks, client risks, and security risks.

#### 2.3.1 - Risks Table

|  |  |
| --- | --- |
| Type & Title | Technical Risks - Ineffective Time Management |
| Description | The time required to develop the website or finish documentation may be beyond initial time estimation. This may lead to failure in making deadlines or a subpar finished product. |
| Probability | Probable - High (16) |
| Mitigation Strategies | Team members should always be aware of their individual allocated tasks as well as the others. Any issues or problems beyond the scope of what was initially discussed when assigning tasks should be communicated with the team to find solutions as soon as possible. |

|  |  |
| --- | --- |
| ID, Type & Title | Client Risk - Accessibility and User Interface |
| Description | The Client has specifically tasked us to model the product over their previous academic tracker “Classroom Monitor”. The finished product may not be as the client envisioned and can lead to user dissatisfaction. |
| Probability | Unlikely - Very High (10) |
| Mitigation Strategies | Ensure constant communication with the client over design issues, conduct a small sample testing with students to test whether the website is user-friendly and intuitive and adjust based on feedback. |

|  |  |
| --- | --- |
| ID, Type & Title | Technical Risk - Cross-Browser and Mobile Phone Compatibility |
| Description | Different web browsers may interpret HTML, CSS, and JavaScript Codes differently, leading to inconsistency and incompatibility with the website display and functionality. This also applies to mobile phones with different resolutions and screen sizes. |
| Probability | Unlikely - High (8) |
| Mitigation Strategies | Use CSS frameworks or libraries that can handle cross-browser and cross-device compatibility and follow web standards and good practices to ensure no problem occurs. Alternatively, we can conduct testing over multiple browsers to address and identify issues. |

|  |  |
| --- | --- |
| ID, Type & Title | Technical Risk - Integration with Educational System |
| Description | Integrating with existing school academic systems and databases may pose challenges due to compatibility issues or data synchronisation. |
| Probability | Unlikely - Medium (6) |
| Mitigation Strategies | Communicate with Elemore Hall IT department and use standard protocols and APIs if available, and if possible, reformat the database to suit current needs. |

|  |  |
| --- | --- |
| ID, Type & Title | Security Risk - Data Breaches |
| Description | Risk of unauthorised access to databases, student information, and administrative information can lead to theft of information or manipulation. |
| Probability | Rare - High (4) |
| Mitigation Strategies | Implement strong authentication measures (e.g., MFA Microsoft Authenticator), authentication protocols, strong password complexity requirements, and regular password updates. |

|  |  |
| --- | --- |
| ID, Type & Title | Technical Risk - Later Updates/Modifications |
| Description | The client may decide to add or change certain aspects of the new academic tracker that may otherwise have not come up during the discussion. |
| Probability | Rare - Low (2) |
| Mitigation Strategies | Apply good coding practices to ensure readability and code quality, having with clear and descriptive comments and documentation can ensure that future changes can be done smoothly. |

### 2.3.2 Risks Matrix

| Probability/Impact (Score)\* | Very Low (1) | Low (2) | Medium (3) | High (4) | Very High (5) |
| --- | --- | --- | --- | --- | --- |
| Highly (5) |  |  |  |  |  |
| Probable (4) |  |  |  | Ineffective Time Management  (16) |  |
| Possible (3) |  |  |  |  |  |
| Unlikely (2) |  |  | Integration with Educational Systems  (6) | Cross-Browser and Mobile Compatibility  (8) | Accessibility and UI  (10) |
| Rare (1) |  | Later Updates/  Modifications  (2) |  | Data Breaches  (4) |  |

\*Score indicates risk class and is calculated by probability \* score

## 3 - Project Development

This section outlines our chosen development methodology and projects a detailed schedule for the Classroom Monitor project For Elemore Hall. We justify our approach, to ensure it aligns with the unique demands of our project. A structured timeline will be provided to effectively monitor our process towards project completion.

### 3.1 – Development approach

In the development of the Classroom Monitor, we have adopted the **Agile** methodology, particularly focusing on the **Scrum** framework within the software development lifecycle (SDLC). We have chosen Agile development due to its strong synergy with our project strategy, as we were inspired by the "Manifesto for Agile Software Development" formulated in 2001. We believe its four core values – prioritizing individuals and interactions, working software, customer collaboration, and responsiveness to change – are essential for effective software development. By implementing these principles through the Scrum framework, we aim for a flexible, collaborative development process that is focused on delivering a functional, user-centric software solution.

The approach we chose is based on a thorough evaluation of key factors including the project scope, team dynamics, the expectations set by our client Mr. Hunter, and our organizational capabilities, as detailed in the points below:

* **Flexibility of Product Scope:** Agile Scrum's iterative development and responsive planning are essential for adapting our tracking system to Mr. Hunter's dynamic requirements. The flexibility of Scrum enables us to rapidly adjust our development process, which is particularly beneficial when addressing the imminent obsolescence of current systems and incorporating new functionalities.
* **Leveraging Team Strengths**: The Scrum framework enhances our planning and task allocation by utilizing our team's diverse skills and preferences. Focusing on collaboration and self-organization, Scrum promotes regular sprints and weekly meetings. This structure enables us to effectively set priorities and evenly assign tasks based on individual member strengths, ensuring alignment with our project goals. This approach allows everyone to contribute optimally within their comfort zone, thus boosting productivity and team cohesion.
* **Collaborative Client Relationship**: The Scrum framework encourages a dynamic partnership with our client. Regular sprint reviews and continuous communication ensure that our development process remains in sync with the client’s vision, which is specifically customised to fulfil Elemore Hall's specific requirements. This ongoing approach is crucial for delivering a system that meets the client's requirements effectively and guarantees client satisfaction.
* **Ensuring Quality and Security**: As we are dealing with the data of sensitive students in special schools, those data demand a strong commitment to security and quality. Agile Scrum's emphasis on frequent testing and code reviews is an ideal fit with our commitment to maintaining high data protection standards, especially for the students' personal information. This approach guarantees that we deliver a monitoring platform that is both secure and reliable sustainably.
* **Responsiveness to User Needs:** Given that our target users include students, teachers, and support staff, our system must address diverse requirements and be user-friendly, because our clients and users are not software experts. Scrum's user-centric methodology is promoted for easier iterative development with feedback loops, it can make sure our system improvements are guided by user feedback. By actively involving users in usability testing sessions, we can customise it to meet their needs and preferences based on reviews.

#### 3.1.1 Comparison with Other Methodologies

* **Waterfall** - Rigidity: The rigidity of the linear and sequential phases of the Waterfall model is a major drawback for the Classroom Monitor project. Since we have a relative lack of experience with such projects, its fixed stages are inflexible and cannot adjust to changing client requirements. On the other hand, **Scrum** provides greater efficiency because of its regular sprint reviews and active client engagement, which enables us to effectively customise our work to meet Elemore Hall's unique requirements.
* **Spiral** - Complexity: The Spiral model with its emphasis on risk assessment, is overly complex for the Elemore Hall project since we only have a tight six-month timeline to complete. Considering the urgent need to replace the existing system, a model that involves extensive risk testing and potential delays is not suitable for us. We have chosen Scrum because it offers a simpler and more time-efficient approach, which is essential for addressing the particular requirements and quick adjustments needed for the Elemore Hall project.
* **XP** - challenging: Extreme Programming (XP) heavily emphasis on frequent releases and continuous customer feedback, which might be challenging given our project's scope and our team's experience level. The strict cooperation and quick adaptation required by XP may be excessive for a team that is still getting to know the basics of project management and software development. Instead, we chose Scrum because it provides a well-balanced framework with its time-boxed sprints and frequent reviews, better fitting the six-month timeline of our project.

#### 3.1.2 Confirming Our Path Forward

By using Agile Scrum as our development approach, we commit to applying the approach that reflects the dynamic and intricate nature of the Classroom Monitor project. This approach ensures that our work goes beyond simple software development, with the goal of developing a complete solution that consistently aligns with the school's requirements and desires to meet or exceed Mr. Hunter's expectations.

### 3.2 – Project Schedule

This section introduces the Project Schedule for the Classroom Monitor, is an important component of our project management methodology. It illustrated in a Gantt chart to outline tasks, durations, and dependencies for each project phase, ensuring we meet our development goals efficiently and on time. Utilizing the Scrum approach, we conduct daily scrums via WhatsApp, hold two-week sprint cycles, and organize monthly general reviews meeting with our client and team. Each sprint cycle involves planning meetings, coding, usability testing, and review sessions for continuous feedback and improvement. Our schedule can be adjusted as needed, considering holidays and other deadlines for module coursework.

A. Srivastava, S. Bhardwaj and S. Saraswat, "SCRUM model for agile methodology," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 864-869, doi: 10.1109/CCAA.2017.8229928.

Manifesto, Agile. (2001). *Manifesto for Agile software* development. Available online:   <https://agilemanifesto.org/iso/en/manifesto.html> (accessed on 4 November 2023).