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

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]

**Academic Tracker –Contents**

[1 - Introduction 2](#_Toc151087686)

[1.1 - Overview and Justification 2](#_Toc151087687)

[1.1.1 Purpose 2](#_Toc151087688)

[1.2 - Project Scope 2](#_Toc151087689)

[1.2.1 - Problems 2](#_Toc151087690)

[1.2.2 - Solutions 2](#_Toc151087691)

[1.2.3 - Stakeholders and Users 2](#_Toc151087692)

[1.2.4 - Future Features 2](#_Toc151087693)

[1.3 - System Description 3](#_Toc151087694)

[1.3.1 - System Design 3](#_Toc151087695)

[1.3.2- Existing Solutions 3](#_Toc151087696)

[1.3.3 Integration into Existing Systems 3](#_Toc151087697)

[2 - Solution Requirements 4](#_Toc151087698)

[2.1 - Functional requirements 5](#_Toc151087699)

[2.2 - Non-Functional requirements 9](#_Toc151087700)

[2.3 - Risks and Issues 11](#_Toc151087701)

[2.3.1 - Risks Table 11](#_Toc151087702)

[2.3.2 Risks Matrix 13](#_Toc151087703)

[3 - Project Development 14](#_Toc151087704)

[3.1 – Development Approach 14](#_Toc151087705)

[3.1.1 Comparison with Other Methodologies 14](#_Toc151087706)

[3.2 – Project Schedule 15](#_Toc151087707)

[4 - Reference 16](#_Toc151087708)

## 1 - Introduction

This document outlines the Requirements Specification for the Classroom Monitor project, structured into three sections. The Introduction provides an overview of the project, encompassing its purpose, scope, and a detailed system description, which includes an analysis comparing similar products. The Solution Requirements section elaborates on both functional and non-functional requirements, as well as potential risks. Finally, the Project Development section outlines our development approach with comparison to others and provides a detailed long-term project schedule.

### 1.1 - Overview and Justification

We are initiating this project under the guidance of **Mr. Michael Hunter**, who shall henceforth be referred to as ‘the client’. The client has commissioned us to develop an **Online tracking system**, named **the Classroom Monitor.** The system will enable to generate Mark Book for students based their academic performance and allow users to track the students’ Mark Book online. This development comes as the old version of the system is not aligning with their needs and requires new features especially for the special school. There are several key project goals:

1. Creating a secure online platform for storing student information.
2. Analyzing student data to generate Mark Book reflective of student behavior.
3. Ensuring that only authorized individuals have login access and editing permissions to the website.

#### 1.1.1 Purpose

The website we plan to create will serve a dual purpose: enabling students to track their academic progress in each subject, and allowing the school to update grades, monitor student progress over time, and intervene as necessary.

### 1.2 - Project Scope

#### 1.2.1 - Problems

Our client urgently requires a replacement for their current academic system, which is scheduled for discontinuation in July 2024. They seek an all-encompassing academic tracking and monitoring system, that maintains user familiarity at Elemore Hall, while adding specific special features for special school that absent in the previous tracker.

#### 1.2.2 - Solutions

To address Elemore Hall's needs, we will develop a user-friendly web interface featuring essential elements like student profiles, grading, attendance tracking, and progress reports. Additional functionalities will include extracurricular activity tracking, tutoring services, and a parent/guardian portal. Emphasizing data security and scalability, the system will be integrated with educational tools and supported by training, user support, testing, and user feedback. Deployment, maintenance and evaluation will occur according to set schedules and budgets.

#### 1.2.3 - Stakeholders and Users

The academic tracking system for Elemore Hall is designed for a diverse range of stakeholders—students (from elementary to higher education), parents, faculty, and administrators. This platform enables students and parents to monitor academic performance and access resources, while teachers and administrators manage academic task operations, as well as the IT department ensure the system’s functionality and security. Elemore Hall's specific requirements drive the development of this efficient tracking platform, aimed at benefiting all users.

#### 1.2.4 - Future Features

Our academic tracking webpage will include secure user authentication, a user-friendly interface with automated grading and real-time updates, individual student portfolios, comprehensive attendance tracking, and a communication platform for enhanced engagement. Advanced analytics tools will enable trend identification and mobile responsiveness. The system will support Learning Management System integration, customizable reports, gamification, and feedback channels for a user-centric design. Integration with external data and accessibility features will further improve functionality, continuously refined based on stakeholder feedback for optimal user satisfaction.

### 1.3 - System Description

#### 1.3.1 - System Design

The project we will create is an academic tracker website, similar to existing websites such as Classroom Monitor and Insight but tailored to meet the needs of the stakeholder, Elemore Hall. Our design prioritizes simplicity and ease of use. Below is a summary of the key features that the end system include:

* **Steps tracker**: This feature enables tracking of students’ current and past progression in 'Elemore steps'.
* Users (Teaching staff) will be able to view and update the Mark Book for each student in their subjects.
* Users will be able to add evidence new entries in a student’s Mark Book.
* **Progress overview:** Presented as a line graph or bar chart, that visually represents the progress of either an individual or a group within a selected timeframe in a specific subject
  + Users will be able to select the timeframe for this overview and choose between individual or class progress displays
* **Pupil progress report:** This report will highlight the level that each student is working at, at key points of the academic year, allowing teachers to track pupil growth term by term
* Users will be able to view the step that each student is on in each subject at the beginning and end of each term
* **Literacy/Numeracy tracker:** Thiswill contain data such as reading age, spelling age, etc.
  + Users will be able to input, modify, and view the levels of students
* **Target overview:** Thiswill allow targets to be set for students to meet at the end of their academic year
  + The overview will contain data on student’s learning level at the beginning of the year and the end of each term
  + The overview will state what each student’s target for the end of the year is and whether they have met it
* **Database Development:** We plan to build a database to hold student’s data and other relevant data

**System Design Details**

* **Frontend** - We plan to implement the front end of the website using JavaScript, CSS, and HTML. As the system is an academic tracker, we want the website to react quickly to user input/interaction - JavaScript will be good for this
* **Server side** (Backend) - We plan to implement the back end of the website using JavaScript
* **Database** - We will implement a database to store data, primarily about Elemore School students

#### 1.3.2- Existing Solutions

**Classroom Monitor**

Our client currently uses Classroom Monitor (CM) to keep track of student progress, for this reason, our planned system will have similarities to CM as the client would like to keep a lot of the functionality. CM has a Mark Book keeping track of STEPs, we plan to implement a similar Mark Book as the client has stated that the current format of the Mark Book is easy to read and understand. Key similar features to be implemented include adding evidence for each STEP, enabling views of school-wide, year group, and class data, and facilitating data set comparisons. While the client is satisfied with CM, they seek additional features like a separate literacy/numeracy tracker and more versatile progress overviews, including bar charts, line graphs, and comparative views of data sets.

**Educater**

The Educater assessment tracker has features that are similar to what we plan to implement in our system. In particular, it has a steps tracker with a similar visual format to the Elemore Steps that our client wants. Spreadsheets and files can be uploaded to their system which is useful for uploading test scores. While Educater offers numerous useful features like attendance reports, these exceed our project scope. Our client's requirements for the features are outlined in Section 1.3, we should aim to focus on implementing these features well and robustly and prevent scope creep.

#### 1.3.3 Integration into Existing Systems

We will need to find a way to pull the data the client currently has stored on Classroom Monitor into our database/system. It is likely we will have to do this using an API for data retrieval. The client also uses SIMS alongside CM currently so we will also need to set up a way to pull data from SIMS into our database/system, SIMS appears to have APIs to achieve this so we plan to retrieve student data using API keys.

## 2 - Solution Requirements

A diagram of a software company

Description automatically generated with medium confidenceThis is a UML Diagram to represent the flow of the system, the independent items in the system, and how they are interconnected.

### 2.1 - Functional requirements

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

**FR1 - Login**

Users must sign in to their account using their school email and password on the homepage.

|  |  |
| --- | --- |
| ID, Type and Title | FR1 Login - Confirm email and password |
| Description | Users must enter their school email and password to access the software. |
| MuShCo and Priority | Must-Have - High |
| Dependencies | N/A |
| Expected Results | Entered email and password will be matched and confirmed by the database if valid.  The software navigates to the homepage. |
| Exception Handling | Errors displayed for invalid email/password.  Errors displayed for incorrect input/email format. |

**FR2 - Admin Permission**

The software includes a page for editing the level of permission of different users, which accessible by admin only.

|  |  |
| --- | --- |
| ID, Type and Title | FR2 Admin permission - Edit users' permissions |
| Description | A table view page displaying users' names and permissions level. 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 reflect user access levels to different pages. |
| Exception Handling | Error displayed if an existing user is added to the system. |

**FR3 - Mark Book**

The software features a grid page for viewing and editing student academic performance.

|  |  |
| --- | --- |
| ID, Type and Title | FR3.1 Editing 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 | 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 Selecting 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 colour 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 students does not participate. |

|  |  |
| --- | --- |
| ID, Type and Title | FR3.3 Viewing 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 colour of the step. |
| Exception Handling | When the step is disabled, the corresponding evidence tile will also be disabled. |

|  |  |
| --- | --- |
| ID, Type and Title | FR3.4 Adding 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 Filtering 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 Displaying overall step descriptions |
| 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 displaying a page that shows the progress overview of the specified groups, subjects, and periods.

|  |  |
| --- | --- |
| ID, Type and Title | FR4.1 Filtering progression |
| Description | Users can use the drop-down menu to select subjects, time scroll to select a 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 for out-of-range parameters. |

|  |  |
| --- | --- |
| 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 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 for missing necessary fields. |

**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 for incorrect data entry. |

|  |  |
| --- | --- |
| 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 colouring the corresponding tiles. |
| MuShCo and Priority | Must-Have - High |
| Dependencies | N/A |
| Expected Results | Students’ LAC / PP status will be displayed by coloured tiles in the overall. |
| Exception Handling | A custom error message will be displayed for incorrect data entry. |

**FR6 - Pupil Progression Report**

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

|  |  |
| --- | --- |
| 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 | Comprehensive annual reports for students will be displayed. |
| Exception Handling | A custom error message will be displayed for missing necessary data. |

**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 can not 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, Usability, Security, Scalability,** 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 within 20 seconds. |
| Security | Use encryption for data at rest and in transit to protect sensitive information during processing. |
| Constraints | Ensure processing time is within the limits of available hardware and 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 | Ensure usability for non-IT skilled users. |
| Security | Not applicable. |
| Constraints | Navigator and other instructions should be easily accessible. |

|  |  |
| --- | --- |
| Type | NFR 2.2 Flexibility |
| Metrics | Operate on both OS and Windows systems. |
| Security | Secure login process with verified third-party software like 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 | Not applicable. |
| Constraints | Using a error-catching mechanisms for data-related issues. |

**NFR3 Security requirements**

Security requirements specify and constrain data security and account security.

|  |  |
| --- | --- |
| Type | NFR 3.1 Data Security |
| Metrics | Prevent any third party access to private user data. |
| Security | Ensure that only authorized personnel have access to systems and tools for data security. Implement and regularly review access control policies, updating user permissions as necessary. |
| Constraints | Using third-party services or organizations is constrained by the security practices and measures of those third parties. |

|  |  |
| --- | --- |
| Type | NFR 3.2 Sign-in Security |
| Metrics | Implement functions or software for protected the sign-in process. |
| Security | Not applicable. |
| Constraints | Include password entry limitations and email confirmation for logins. |

**NFR4 Scalability requirements**

Scalability requirements specify and constrain support for large data.

|  |  |
| --- | --- |
| Type | NFR 4.1 Support for 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 presents significant challenges, as it often involves heightened risks of data breaches, data loss, and various 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 data only with consent. |
| Security | Restrict data updates to specific roles like “Customer Advisor” or “Supervisor”. |
| 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 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 the 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 make adjustments 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 synchronization. |
| 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 | The 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 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 development methodology and provides a detailed schedule for the Classroom Monitor project at Elemore Hall. It is crucial to ensure that our approach aligns with the project's unique demands and guaranteeing an effective path to completion.".

### 3.1 – Development Approach

For the Classroom Monitor, we have adopted the **Agile** *(Manifesto for Agile Software Development)* methodology, particularly focusing on the **Scrum** *(Srivastava et al., 2017)* 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". We believe its four core values – prioritizing individuals and interactions, working software, customer collaboration, and responsiveness to change – are essential for effective software development. 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 customized 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 private data of students, 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 student's 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 customize it to meet their needs and preferences based on reviews.

#### 3.1.1 Comparison with Other Methodologies

* **Waterfall:** The rigidity of the linear and sequential phases of Waterfall model is a major drawback for the Classroom Monitor project. Since we are lack of experience with such projects, its fixed stages are inflexible and cannot adjust to changing client demand. Scrum provides greater efficiency because of its regular sprint reviews and active client engagement, which enables us to effectively customize our work to fulfil Elemore Hall’s need.
* **Spiral:** This model's complexity and risk assessment focus are impractical and overly complex for our six-month project timeline. 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 its simplicity and efficiency are more beneficial for addressing the particular requirements and quick adjustments needed for Elemore Hall project.
* **Extreme Programming (XP):** XP heavily emphasizes frequent releases and continuous customer feedback, which challenging for 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 flexible framework with its time-boxed sprints and frequent reviews, better fitting the six-month timeline of our project.

By using Agile Scrum, 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, to develop a complete solution that aligns with the school's requirements and meet or exceed Mr. Hunter's expectations.

### 3.2 – Project Schedule

This section introduces the Project Schedule for the Classroom Monitor, which is an important component of our project management methodology. It is 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, engage in bi-weekly sprint cycles, and organize monthly general review meetings 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.

The Gantt Chart for our project has been created using the TeamGantt application (app.teamgantt.com).

## 4 - Reference

1. *(No date) Manifesto for Agile Software Development. Available at: https://agilemanifesto.org/ (Accessed: 16 November 2023).*
2. *Srivastava, A., Bhardwaj, S. and Saraswat, S. (2017) ‘Scrum model for Agile methodology’, 2017 International Conference on Computing, Communication and Automation (ICCCA) [Preprint]. doi:10.1109/ccaa.2017.8229928.*