

SECURED STUDENT INFORMATION MANAGEMENT SYSTEM

HEMANILASHINII A/P THARMALINGAM

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

JUNE 2024

SECURED STUDENT INFORMATION MANAGEMENT SYSTEM

BY

HEMANILASHINII A/P THARMALINGAM

THE PROJECT REPORT IS PREPARED FOR

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR**

**BACHELOR OF INFORMATION TECHNOLOGY
B.I.T. (HONS) SECURITY TECHNOLOGY**

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY
MULTIMEDIA UNIVERSITY**

JUNE 2024

© 2024 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED

Copyright of this report belongs to Universiti Telekom Sdn. Bhd as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialization policy. No part of this publication may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.



Hemanilashinii A/P Tharmalingam

Faculty of Information Science & Technology
Multimedia University

Date: 13th June 2024

ACKNOWLEDGEMENT

Firstly, I would like to express my gratitude to MS. Pa Pa Min, my supervisor who helped me to complete the project within the stipulated time. I was able to communicate at any time regarding the project details and thanks for spending her time to answer my questions. The guidance and support from madam were helpful throughout the project development.

Besides that, I would like to thank my family, especially my mother for being there with me as moral support to deliver the project as good as possible.

Additionally, I would like to thank the Faculty of Information Science & Technology, Multimedia University for giving me the opportunity to conduct the final year project.

Finally, thanks to my friends who helped answer some of the doubts based on the project.

ABSTRACT

The student information management system plays an important role in educational institutions. This system is used to manage and organize valuable information related to students. The system addresses key aspects such as enrollment, academic records, financial details, subject management, and attendance class schedules, sending and receiving emails, creating and submitting assignments. The main users of this system are students, admin, and instructors. The system can be accessed based on their email address and the matching password whether the system brings to the instructor or student dashboard. Whereas for admin, they have their own URL to access all the necessary resources. The availability of the system must be user friendly, giving a convenient feeling to the users to use the system error free and easily navigate the functions and features. The aim of this project is to provide a holistic solution for an effective and secured student information management system with securing information by leveraging technology to streamline administrative processes, students and instructor's activities while contributing to the success and growth of educational institutions.

Table of Contents

DECLARATION.....	III
ACKNOWLEDGEMENT.....	IV
ABSTRACT.....	V
TABLE of CONTENTS.....	VI
LIST of TABLES.....	VIII
LIST of FIGURES.....	IX
LIST OF ABBREVIATION/SYMBOL.....	XII
LIST of APPENDICES.....	XIII
CHAPTER 1: INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Problem Statement.....	2
1.3 Project Objectives.....	3
1.4 Project Scope.....	4
1.5 Report Organization.....	5
CHAPTER 2: LITERATURE REVIEW.....	6
2.1 Overview.....	6
2.1.1 Key trends in Student Information Management System.....	7
2.1.2 Challenges in developing Student Information Management System.....	8
2.2 Web based approach.....	9
2.3 Existing Student Information Management System.....	10
2.4 Example interface of existing system.....	23
2.5 Summary.....	28
CHAPTER 3: METHODOLOGY.....	29
3.1 Overview.....	29
3.2 Chosen methodology.....	31
3.3 Hows does the waterfall methodology works.....	32
3.4 Requirements.....	35
3.5 Gantt chart.....	39
3.6 Summary.....	41

CHAPTER 4: IMPLEMENTATION.....	42
4.1 Overview.....	42
4.2 DFD, Flowchart, UML Case Diagram.....	43
4.3 HTTPS implementation.....	50
4.4 Password hashing implementation.....	54
4.5 Time-out session implementation.....	55
4.6 CSRF token implementation.....	56
4.7 Database implementation.....	58
4.8 Summary.....	58
CHAPTER 5: TESTING and DISCUSSION.....	59
5.1 Overview.....	59
5.2 Interface testing.....	59
5.3 Summary.....	77
CHAPTER 6: CONCLUSION.....	78
6.1 Overview.....	78
6.2 Conclusion.....	78
6.3 Limitation and recommendation for future works.....	80
REFERENCES.....	82
APPENDICES.....	84

List of Table

Table 2.3.11 Summary of existing systems.....	21
Table 3.3.1 Advantage and disadvantage of waterfall methodology.....	34
Table 3.5.1 Gantt chart for FYP1.....	39
Table 3.5.2 Gantt chart for FYP2.....	40

List of Figures

Figure 2.3.1 PowerSchool.....	11
Figure 2.3.2 Blackbaud.....	12
Figure 2.3.3 Skyward.....	13
Figure 2.3.4 Aeries.....	14
Figure 2.3.5 Rediker.....	15
Figure 2.3.6 Teachmint.....	16
Figure 2.3.7 QuickSchools.....	17
Figure 2.3.8 Wisenet.....	18
Figure 2.3.9 Gradelink.....	19
Figure 2.3.10 Orah.....	20
Figure 2.4.1 PowerSchool Interface.....	23
Figure 2.4.2 Blackbaud Interface.....	23
Figure 2.4.3 Skyward Interface.....	24
Figure 2.4.4 Aeries Interface.....	24
Figure 2.4.5 Rediker Interface.....	25
Figure 2.4.6 Teachmint Interface.....	25
Figure 2.4.7 QuickSchools Interface.....	26
Figure 2.4.8 Wisenet Interface.....	26
Figure 2.4.9 Gradelink Interface.....	27
Figure 2.4.10 Orah Interface.....	27
Figure 3.1.1 Process of SDLC.....	30
Figure 3.2.1 Waterfall methodology.....	31
Figure 3.4.1 VS Code.....	35
Figure 3.4.2 Python.....	36
Figure 3.4.3 Command.....	37
Figure 3.4.4 Device.....	38
Figure 4.2.1 DFD-Level 0.....	43
Figure 4.2.2 DFD-Level 1.....	44
Figure 4.2.3 Flowchart for entire system.....	45
Figure 4.2.4 Flowchart for student.....	46
Figure 4.2.5 Flowchart for admin.....	47

Figure 4.2.6 Flowchart for instructor.....	48
Figure 4.2.7 UML case diagram.....	49
Figure 4.3.1 Install chocolatey.....	50
Figure 4.3.2 Chocolatey verification.....	51
Figure 4.3.3 Mkcrt installation.....	51
Figure 4.3.4 Install of CA.....	52
Figure 4.3.5 CA for localhost.....	52
Figure 4.3.6 SSL certificate and key.....	53
Figure 4.3.7 SSL server running.....	53
Figure 4.4.1 Password hashing.....	54
Figure 4.5.1 Time-out session.....	55
Figure 4.4.1 HTTPS request headers.....	56
Figure 4.4.1 Cookies being sent.....	57
Figure 4.4.1 CSRF in the payload.....	57
Figure 5.2.1 Register page.....	59
Figure 5.2.2 Login page.....	60
Figure 5.2.3 Student dashboard.....	61
Figure 5.2.4 Courses.....	61
Figure 5.2.5 Instructors.....	62
Figure 5.2.6 Fees payment.....	62
Figure 5.2.7 Enrollment	63
Figure 5.2.8 Grade.....	63
Figure 5.2.9 Notification.....	64
Figure 5.2.10 Send email.....	64
Figure 5.2.11 Sent email.....	65
Figure 5.2.12 Class schedule.....	65
Figure 5.2.13 Instructor dashboard.....	66
Figure 5.2.14 Courses.....	66
Figure 5.2.15 Enrolled students.....	66
Figure 5.2.16 Assignment.....	67
Figure 5.2.17 Class schedule.....	68
Figure 5.2.18 Grade.....	68
Figure 5.2.19 Send email.....	69
Figure 5.2.20 Sent email.....	69
Figure 5.2.21 Make announcement.....	70
Figure 5.2.22 Mark attendance.....	70

Figure 5.2.23 Attendance record.....	71
Figure 5.2.24 Django admin.....	71
Figure 5.2.25 Courses.....	72
Figure 5.2.26 Add course.....	72
Figure 5.2.27 Enrolled students.....	73
Figure 5.2.28 Add enrollment.....	73
Figure 5.2.29 Add fees.....	74
Figure 5.2.30 Add instructor	74
Figure 5.2.31 Payment by students.....	75
Figure 5.2.32 Add schedule.....	75
Figure 5.2.33 Student lists.....	76

List of Abbreviations/Symbols

DFD	Data Flow Diagram
UML	Unified Modeling Language
SDLC	Software Development Life Cycle
FYP	Final Year Project
FIST	Faculty of Information Science & Technology
RBAC	Role Based Access Control
SSL	Secure Socket Layer
SQL	Structured Query Language
VPN	Virtual Private Network
RSA	Rivest, Shamir, Adleman Algorithm
TLS	Transport Layer Security
HTTPS	Hypertext Transfer Protocol Secure
LDAP	Lightweight Directory Access Protocol
SAML	Security Assertion Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
CA	Certificate Authority
PBKDF2	Password-Based Key Derivation Function 2
CSRF	Cross-Site Request Forgery

List of Appendices

Appendix A: Meeting logs.....84

Chapter 1

Introduction

1.1 Overview

The Student Information System is an application-based project which is a software-designed website to manage and organize student-related pieces of information in a single repository for the educational institution's purpose. This application is huge and usually includes types of elements to handle different aspects of student data, administration processes and academic staff in an education system. This system mainly focuses on students where they can access the enrollment page for registration of subjects and view the details about academic records to know their progress in studies. Students are also allowed to scan the attendance, view, and pay tuition fees on time. Apart from this the administrator manages the student user, financial details of a student, and maintains course information. Academic staff is one of the users of this system providing teaching and learning processes to the student and at the same time they are the main actors in collecting and gathering academic information after each course assessment and at the end giving the final output of the examination marks. A well-implemented system application enhances the interaction between the users and the system by providing valuable information for decision-making within educational institutions contributing to the efficient management of student data and other operations.

1.2 Problem Statement

In the current educational system, every institution faces difficulties in managing student information efficiently and effectively. Organizing multiple data for an entire institution such as student enrollment, academic records, financial details, and student personal information can consume a substantial amount of administrative time. The manual data entry process for gathering the data results in data redundancy, errors in academic records, and difficulties in keeping track of information on a real-time basis.

Moreover, some of the existing systems may lack user-friendly interface aspects for students, administration, and academic staff. This might affect the workflow of the student information system in enrolling students, hard-to-access relevant academic information and lack of interaction between the user and the system.

In addition, with the increasing reliance on networks or digital systems, the safety of sensitive student data has become a concern for users. Insufficient security measures may expose vulnerabilities such as data breaches, compromising the privacy and confidentiality of student records.

Furthermore, the lack of a single repository to collect and keep the data about students can result in delays in sharing important information between students, admin, and academic staff. At the same time, important announcements, updates, and notifications may not reach all the users in the supposed time.

1.3 Project Objectives

This project aims to enhance the Student Information System that approaches existing problems. By considering the proposed objectives, the enhancement works in the Student Information System will deliver a more efficient, secure, and user-friendly academic management environment within the educational institution.

To provide one stop resources

- Developing a single repository as a student information management system to accommodate all information about the students, academic staff, and administration. It is to view all the data in one place rather than separate places.

To deliver user friendly system

- Designing a user-friendly interface with responsive design to ensure accessibility. It will be easy for all users to get their job done without any errors occurring and can easily navigate through features and functions available in the student information management system.

To develop a system that prioritizes data safety

- Implementing a robust security feature in the Student Information System can ensure the security and privacy of sensitive student information. This helps the system to protect the data from breaches by anonymous users.

To deliver an automated process

- An automatic procedure rather than a manual process, it can reduce paperwork among the students, administration, and academic staff. Implementing a computerized process improves the convenience of using the system by reducing any paperwork and can save time eventually.

1.4 Project Scope

The project scope of the Student Information System outlines the boundaries of functionalities and features when developing the program for students, admin, and academic staff.

Starting the system must have user roles and permissions. It defines different roles such as student, academic staff, and administration. The specific role-based access helps to ensure only the specific user can use the respective page. On the other hand, security measures are important in incorporating security features such as role-based authentication and time out for the section to avoid unauthorized access. Safeguards sensitive data.

Apart from the security function and user roles and permission the features of the system are also equally important for a system. Here starting off with personal information of the user. The user is encouraged to add their details such as name, id, email address, phone number and other private data. Moreover, financial management is another feature in the student information management system. Students as a user can view and settle the tuition fees. Therefore, admin gets to know the financial details of a student.

Besides that, enrollment is an important feature in this system. Because students must enroll into the subjects for a particular time. Admin will manage the course information after the students are done with the enrollment. In terms of attendance students can mark the attendance according to the subject and the time and date will display in real time.

Furthermore, the academic records function can be used by both the students and academic staff. Students can view their final grades from examinations, quizzes, and assignments. Academic staff are responsible for assessing the work done by the students and providing the final marks. Meanwhile the course management is managed by the admin for the academic staff's teaching details. Finally, the academic staff can check the class details.

1.5 Report Organization

Before starting with the chapters this document contains table of contents, declaration, acknowledgement, abstract, references, meeting logs, and checklist for the report. This project is divided into seven separate sections namely, the introduction, literature review, methodology, proposed design, and finally the conclusion. The first chapter discusses the project's problem statement, objectives, scope and how the features work according to the user's permissions in developing the Student Information System.

The second chapter is a literature review that analyzes the existing student information system and summarizes the collected information and details. The methodology is the third chapter that discusses Software Development Life Cycle phases and the suitable method for the project. And talks about how the selected methods work and are carried out throughout the project. Apart from that this chapter also talks about the software and hardware requirements and finally, a Gantt chart is also presented in this chapter.

The fourth chapter is the proposed design. Here in this chapter a sample of system interfaces has been provided. Software representation of flowchart, DFD and UML case diagram has also been discussed here. The representation provides the visualization of how the data flows within the system for each action done by the users. The last chapter is 5 about the work done during Phase 1 whereas Phase 2 is about the plan that needs to be done to provide a final output of the Student Information System as a website.

Chapter 2

Literature Review

2.1 Overview

Student Information management systems play a vital role in educational institutions, serving as a singular location for storing, managing, and retrieving student data, including academic records, personal information, financial details, reports, and analytics. The effectiveness of SIMS significantly results in the productivity of administrative processes, the quality of student services, and ultimately, the overall success of educational commitments. In recent years, the student information system has evolved notably, embracing technological advancements, and accustoming to the changing needs of education. This literature review explores the current state of SIMS, highlighting key trends, challenges, and emerging strategies for increasing its effectiveness based on the impact.

2.1.1 Key Trends in Student Information Systems

Mobile Access and User-Friendly Interfaces

- The increasing ubiquity of mobile devices has ushered in the creation of mobile-optimized system interfaces. Institutions are creating responsive web designs and mobile applications that provide students, faculty, and staff convenient access to student information on the go.

Personalization and Self-Service

- The student information system incorporates personalization features, based on individual roles, permissions, and academic interests. This personalization strengthens the relevance and accessibility of information for different user groups such as staff, lecturers, and students.

2.1.2 Challenges of Developing Student Information Systems

Despite the advances, several challenges do exist, prompting the development of innovative strategies to address these issues.

Data Security and Privacy

- The increasing volume and sensitivity of user's data raises the importance of data security and privacy. Encryption, access controls, and regular security policies, are some of the security measures to safeguard information and comply with data privacy regulations should be implemented in the educational platforms.

User Experience and User Interfaces

- Effectively utilizing the full potential of the Student information system requires institutions to develop a website which is up to date to provide information to the users whenever and wherever they need. This can easily improve the usability of the website. To create a non-error design the developed kit needs to be designed, tested, and trained before releasing the student information management system by qualified developers.

Cloud Computing

- Most of the student information management system is based on cloud computing as it involves setting up an educational institution's data and IT resources to a cloud server. This can help make education more efficient and equitable administration for the education management.

2.2 Web-Based Approach

Every student information management system is created based on web-related technologies. Web-based student information systems offer various benefits, including accessibility, flexibility, and scalability. By encouraging web technologies, educational institutions can implement a comprehensive student data management system accessible from anywhere with an internet connection for the intended users. By implementing a web-based approach to create a student information management system, educational institutions can develop a dynamic and adaptable system.

Accessibility

- Easily accessible from anywhere with an internet connection, enabling student data access for different users within the system, such as students, staff, and lecturers.

Security

- Implement solid security measures to protect student information, including encryption and access control.

Enhanced user experience

- Provides a user-friendly interface with intuitive navigation and personalized features.

Availability

- Users can access the data or information in one place whenever they want.

2.3 Existing Student Information Management System

Before developing the system, I researched other existing systems. Conducting a study on another system provides numerous benefits, especially when considering adding any features that are not available in the original system or enhancing those errored features. Analyzing an existing system is an opportunity to identify the weaknesses and strengths of the functions that work. In terms of user feedback, the study can provide the user's thoughts based on satisfaction with using it. User feedback is important to understand the needs and requirements which can help to enhance the developing websites or applications. In conclusion, studying an existing system is a necessary step in the process of designing, developing, maintaining, and upgrading a system. This ensures providing user-friendly products with error-free websites and applications.



Figure 2.3.1: PowerSchool

PowerSchool was created in North America and it is a cloud-based software for K-12 education. This system acts as a bridge between students, teachers, parents, and the administration of an institution. The services of this system are delivered through Microsoft Azure. According to PowerSchool (2023), PowerSchool became a public company on July 28, 2021. Even though it is a public company, it offers a service for both K-12 private and public schools and districts. PowerSchool Student Information Cloud's purpose is to provide student information, enrollment management, and database management. This system maintains accurate and complete student data by ensuring the information is easily accessible and delivering precise details. Apart from the features, this system takes the secrecy and privacy of data importantly. Based on PowerSchool, Security (2023) they are using security operations center type 2 (SOC 2) to maintain the confidentiality, integrity, and availability of the students' information (CIA) which runs 24x7x365. To avoid any vulnerabilities threatening the system PowerSchool will do penetration testing. In PowerSchool, the administrators use a Lightweight Directory Access Protocol (LDAP) to securely manage password authentication for teachers, students, and parents. In terms of system security, the users can only access the site according to the time set, the system is automatically inactive after several minutes. The PowerSchool enables session cookies to span subdomains. This system security works for every user.



Figure 2.3.2: Blackbaud

According to Blackbaud (2023), this cloud-based software is designed for fundraising, nonprofit financial management, education, corporate social responsibility (CSR), and more. Among its offerings, Blackbaud provides Education Solutions for K-12 private and parochial schools, as well as higher education institutions. The services include enrollment management systems, learning management systems, and student information systems, which facilitate seamless data flow and a user-friendly experience for parents, students, faculty, and staff via single login and mobile access. Given its cloud-based nature, security is a primary concern in managing organizational data. The system upholds the CIA (Confidentiality, Integrity, Availability) information security standard and employs industry control frameworks like NIST CSF to mitigate cybersecurity risks. To safeguard customer data, Blackbaud utilizes SOC 1 Type 2 reports, which assess design and operational effectiveness. Furthermore, as stated by Blackbaud, Security (2023), they implement encryption mechanisms such as TLS 1.2, AES 256, RSA 1024, and other FIPS 140-2 compliant algorithms.



Figure 2.3.3: Skyward

According to Skyward (2023), Skyward has been serving K-12 education for over 40 years, offering software solutions for student management. Key features of Skyward include office and administration tools, classroom tools, family engagement, and student services. Office and administration features include scheduling, fee management, and curriculum management. Classroom tools encompass attendance tracking, behavior tracking, and a grade book. For family engagement, Skyward provides family access and a family mobile app. Student services include health services, college and career readiness, and behavior management. Any changes made in the scheduling site are updated in real-time. Parents can view a student's fee structure through pre-built reports in the system. According to Skyward's Privacy Policy (2023), the system employs role-based access control (RBAC) to ensure users access only features and data relevant to their roles, such as students and staff, thereby limiting unauthorized access to sensitive information. Skyward uses SSL (HTTPS) via Transport Layer Security (TLS) to encrypt communications, formerly known as Secure Socket Layer (SSL). Kerberos is implemented for authentication and authorization, providing a secure option. Additionally, dual-factor biometric credentials are used for data center access, with all entrances either alarmed or monitored, often utilizing RFID badges or biometric scanners. Based on Skyward (2023), the system restricts direct database access, allowing it only through VPN connections to cloud-hosting data centers. VPN connections are limited to authorized district system admin personnel to prevent unauthorized access.



Figure 2.3.4: Aeries

As stated by Aeries Software (2023), Aeries is a student information system with over 40 years of experience serving K-12 districts in California and Texas. The features offered include student information systems, enrollment, and mobile apps. The student information system supports solutions for student data, scheduling, and grade reporting. The enrollment feature supports multiple languages, allowing different language speakers to continue from where they left off with seamless integration and timesaving features. The mobile app, developed for students, teachers, and parents, provides access to information anytime and anywhere. Aeries integrates with partners like Microsoft, Google Classroom, Canvas, and Illuminate to enhance system functionality. According to Aeries Software, Cloud Hosted Services (2023), Aeries is built on Amazon Web Services, leveraging the platform's scalability, security, and reliability. Aeries Cloud Hosted Services provides a fully managed end-to-end platform with automatic scaling based on district size, ensuring faster performance. Data and information collected are monitored and accessed only by authorized personnel. In addition to being a web-based system, Aeries is also available on mobile devices.



Figure 2.3.5: Rediker

According to Rediker Software, they have been providing technology solutions to public, private, and international schools in over 120 countries for more than 44 years. Their services include gradebook, fund accounting, mobile apps for iOS and Android, report writing, notifications, attendance tracking, transcripts, ParentPlus and StudentPlus portals, discipline management, and e-portfolios. For attendance, Rediker integrates with the AdminPlus Mobile App to monitor both student and staff attendance. They also offer PlusPortals, a family portal that enables teachers to communicate and share information with parents and students simultaneously. The system supports online learning through dedicated websites for students. As mentioned by Rediker in their Integrations and Partnerships documentation, they collaborate with leading educational organizations such as FACTS, Frontline Education, and Google Classroom, and have partnered with Microsoft Azure Cloud to ensure seamless connections between web and mobile applications. According to Rediker's Privacy Policy, they automatically collect and log information such as IP addresses, accessed pages, features used on the website, and geographical locations when users visit the site. They also use analytics and advertising cookies. To protect data, Rediker configures firewalls for real-time scanning of incoming and outgoing traffic and employs IPS/IDS (Intrusion Prevention System/Intrusion Detection System) to inspect traffic and block threats. HTTPS (Hypertext Transfer Protocol Secure) ensures authentication and strong encryption while using the website.



Figure 2.3.6: Teachmint

Teachmint, a multinational company, has developed the Integrated School Platform (ISP), serving over 15 million users across many countries and available in 20+ languages, as stated by Teachmint (2023). This cloud-based system uses SaaS as its cloud delivery platform and offers various solutions, including learning management systems, fee management, student information systems, admission management, and exam planners. The student information system features include student profiles, directories, personal information, academic performance, and attendance details. This system consolidates all student data into a single interface via the SaaS cloud, facilitating easy monitoring of academic progress. Additionally, it helps administrators manage outstanding fees and access information about the facilities used by a student, all within a unified system. Unlike other systems, this one does not provide a separate portal for parents; instead, access to specific fields is determined by the administrators, as mentioned by Teachmint (2023). To become a user, a one-time password (OTP) is sent to a valid mobile number. Teachmint employs several security measures, including a Web Application Firewall (WAF), cloud protection, routine audits, data encryption, and Role-Based Access Control (RBAC). The WAF monitors and controls incoming web traffic, such as DOS attacks, and blocks malicious requests. Additionally, Dynamic Application Security Testing (DAST) is implemented to identify and address vulnerabilities or weaknesses that attackers could exploit to threaten the system.



Figure 2.3.7: QuickSchools

QuickSchools provides a comprehensive school management system with features that include managing student and teacher information, attendance, gradebook, transcripts, parent access, report cards, homework, school scheduling, and admissions. This system allows parents to log in and monitor their child's progress by tracking homework, attendance, exam records, and fee management. Beyond student information management, QuickSchools also offers teacher management features that assist teachers with class scheduling, creating new classes, and preparing reports on student achievements and progress. When teachers submit reports, the system automatically notifies students and parents. According to Azroy Kandan (2015), QuickSchools is hosted by Amazon Web Services (AWS) in Virginia and uses standard encryption protocols like SSL, ensuring that all passwords are encrypted, even QuickSchools cannot access them. As mentioned by Azroy Kandan (2022), QuickSchools employs 2-factor Authentication (2FA) via Google Authentication. The system collects automated information such as the IP address of the device used to log in, regional location, browser type, and operating system. Using cookies, the system serves advertisements on websites that may interest users based on their use of the QuickSchools website. Additionally, it supports Do Not Track (DNT) settings in users' web browsers to prevent the collection of information about the websites they visit.



Figure 2.3.8: Wisenet

According to Wisenet, an Australian software solution founded in 1997, it operates as a cloud-based Learning Relationship Management platform and serves a multinational client base. Colleges and educational institutions utilize Wisenet as a student management system to engage and inform students, manage administrative tasks such as CRM for education, and support online learning through Wisenet One. Key features of the student management system include timetables, enrollment, document management, and reporting. Their network security involves using firewalls with rule sets, Access Control Lists (ACLs), and configurations to restrict information transfer to specific system services. Wisenet applications are accessed via HTTPS, utilizing Secure Socket Layer (SSL) to protect against man-in-the-middle attacks, tampering, and message forgery. As stated by Wisenet (2023), their Application Programming Interface (API) links applications through an interface containing several Wisenet fields called ‘endpoints.’ HTTPS Basic Authentication (RFC 2617) via SSL is used to convey the user’s identity by providing a valid username and password. Wisenet also employs Two-Factor Authentication (2FA) to prevent unauthorized access, Single Sign-On (SSO) for multiple system access with a single set of credentials, and Role-Based Access Control (RBAC) to assign permissions based on user roles specified during registration. To store and manage user data, Wisenet utilizes AWS cloud services, keeping all data in a single repository.



Figure 2.3.9: Gradelink

Gradelink is a cloud-based student information system designed for easy navigation through its resources. It provides fully integrated tools to manage administrative tasks for educational institutions. Gradelink helps track various data such as student information, employee information, class schedules, attendance, grade books, medical records, and more. Its student information management system monitors student enrollment status, stores multiple contacts and emergency addresses, and allows data to be exported into Excel sheets. As stated by Gradelink (2023), all web and mobile connections are secured with SSL encryption to protect web traffic on the server. The collected data is archived and delivered as an encrypted SQL server backup file. Gradelink maintains a locked and secure data center repository in Irvine, California, accessible only to a limited number of authorized IT staff through remote server tools that are regularly logged and audited. In addition to the data center, Gradelink employs network protection protocols such as network segregation to separate high-risk elements from lower-risk networks, firewalls, router technologies, an intrusion detection system using log data, alert mechanisms, and secure VPN connectivity for all authorized personnel, as mentioned in Gradelink Security Overview (2023). These measures help prevent and mitigate the impact of malicious traffic and threats, ensuring the system's security. Gradelink also integrates seamlessly with Google Classroom, ClassLink, and Apple School Manager.



Figure 2.3.10: Orah

Orah, previously known as Boardingware until July 1st, 2021, as mentioned by Paul Organ (2021), has been designed to integrate with existing student information systems since 2014. According to Boardingware (2023), the company has served over 300 schools, 70,000+ students across 20 countries, and has been operational for over nine years. Orah offers features such as attendance tracking, coordination of student activities and schedules, student health monitoring, mood checks, and data safeguarding. Students and parents or guardians can view and sign up for these events via the mobile app or website. As detailed in the Boardingware Cloud Security Whitepaper (2023), Orah utilizes AWS as its cloud platform. As a SaaS provider, Boardingware handles cloud configurations, including user data, hosted information, firewall setups, and network traffic protection. In Orah, usernames and passwords are encrypted with an 11-factor cryptographic hash function. Authentication is server-side, returning a token over an SSL connection. These tokens, encrypted with 256-bit entropy and signed with an RSA signature, ensure that user credentials must be authenticated with each system access. Orah employs Amazon Virtual Private Cloud (Amazon VPC) with multiple security restrictions, assigning private (RFC 1918) IP addresses within the VPC to create an isolated network separated from other Amazon VPCs.

Table 2.3.11: Summary of existing system

Student Management System	Functions	Security Features
PowerSchool	Student Information Cloud Personalized Learning Cloud Educator Effectiveness Cloud	Service Organization Control 2 Lightweight Directory Access Control (LDAP) Time-Based Access Control
Blackbaud	Enrollment Management System Learning Management System Student Information System	Encryption Mechanism (TLS, RSA) Microsoft Azure Active Directory Security Assertion Markup Language
Skyward	Office and Administration Classroom Tools Family Engagement Student Services	Role-Based Access Control HTTPS RFID VPN
Aeries	Student Information System Enrollment Mobile Apps	Hashing System MD5, SHA1 Duo 2-Factor Authentication Google Authentication
Rediker	School-to-Home Communication Mobile Access Real-Time Reporting Online Learning	Intrusion Prevention System HTTPS Google Analytics
Teachmint	Learning Management System	One Time Password Web Application Firewall

	Fee Management Student Information System	Role-Based Access Control Dynamic Application Security Testing
QuickSchools	School Schedule Exam Record Fee Management Gradebook	Secure Socket Layer Encryption 2 Factor Authentication Cookies Do Not Track Setting
Wisenet	CRM for Education Student Management System Online Learning Platform	Access Control List 2 Factor Authentication Single Sign On Role-Based Access Control
Gradelink	Class Schedule Student Information Attendance Management	SSL SQL Server Backup VPN Network Segregation
Orah	Attendance Management Student Health Management Student Mood Check	SSL RSA Amazon Virtual Private Cloud TLS

2.4 Example of Existing System Interface



Student and Parent Sign In

Sign In Create Account

Username

Password

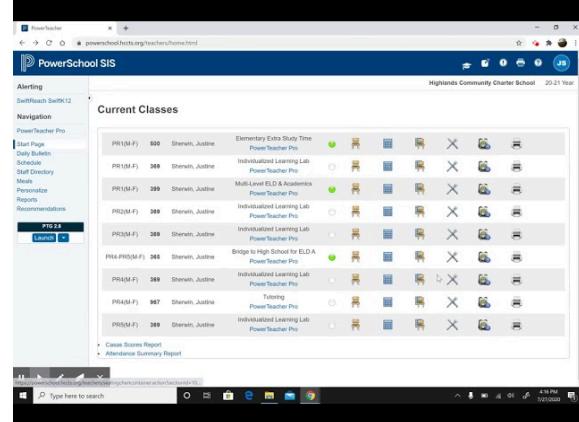
[Forgot Username or Password?](#)

Sign In

Copyright © 2005-2022 PowerSchool Group LLC and/or its affiliate(s). All rights reserved.
All trademarks are either owned or licensed by PowerSchool Group LLC and/or its affiliates.

[Privacy Policy](#)

Student and Parent Sign in Page



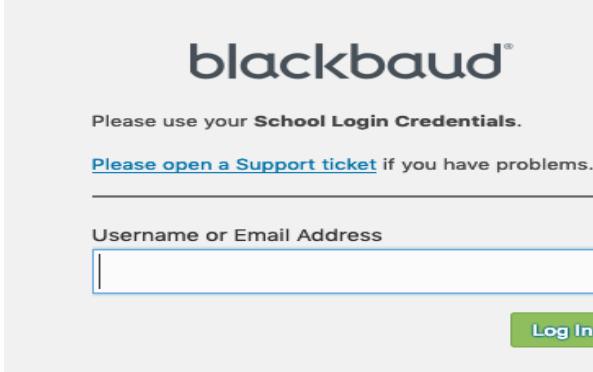
Current Classes

Class ID	Class Name	Teacher	Subject	Day	Start Time	End Time	Location	Status
PR1(M-F)	500	Sherwin, Justice	Elementary Extra Study Time					
PR1(M-F)	369	Sherwin, Justice	Individualized Learning Lab					
PR1(M-F)	369	Sherwin, Justice	Multi-Level ELA & Academics					
PR2(M-F)	369	Sherwin, Justice	Individualized Learning Lab					
PR3(M-F)	369	Sherwin, Justice	Individualized Learning Lab					
PR4 PR5(M-F)	365	Sherwin, Justice	Bridge to High School for ELA					
PR4(M-F)	369	Sherwin, Justice	Individualized Learning Lab					
PR4(M-F)	367	Sherwin, Justice	Tutoring					
PR5(M-F)	369	Sherwin, Justice	Individualized Learning Lab					

[Class Scores Report](#) [Attendance Summary Report](#)

Current Classes for Teacher Page

Figure 2.4.1: PowerSchool



blackbaud®

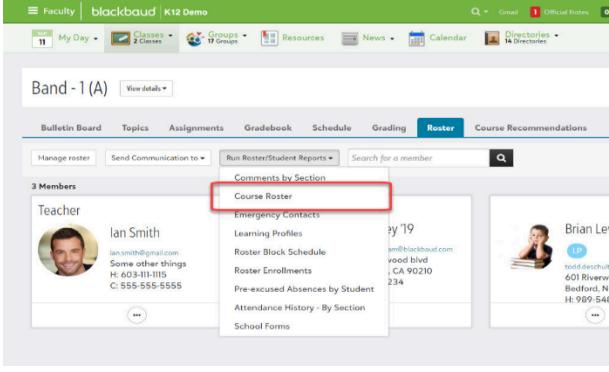
Please use your School Login Credentials.

[Please open a Support ticket if you have problems.](#)

Username or Email Address

Log In

Login Page



Faculty blackbaud K12 demo

Band - 1 (A) [View details](#)

My Day Classes Groups Resources News Calendar Directories

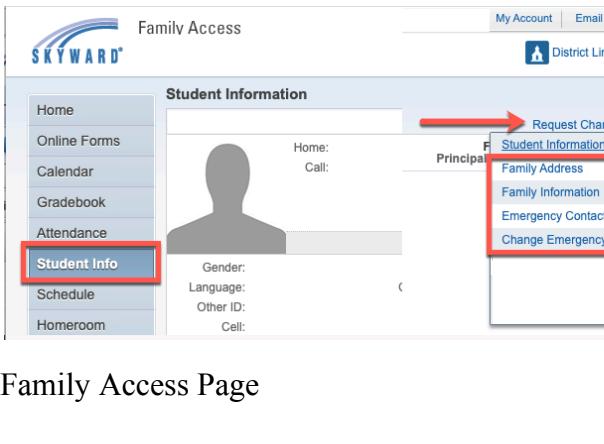
Bulletin Board Topics Assignments Gradebook Schedule Grading Roster Course Recommendations

3 Members

Teacher	Comments by Section
ian.smith@blackbaud.com Some other things H: 603-111-1115 C: 555-555-5555	Course Roster
Emergency Contacts Learning Profiles Roster Block Schedule Roster Enrollments Pre-excused Absences by Student Attendance History - By Section School Forms	ay '19 ian.smith@blackbaud.com wood blvd , CA 90210 234
Brian Lev brian.lev@blackbaud.com 601 Riverine Bedford, NH H: 989-546	

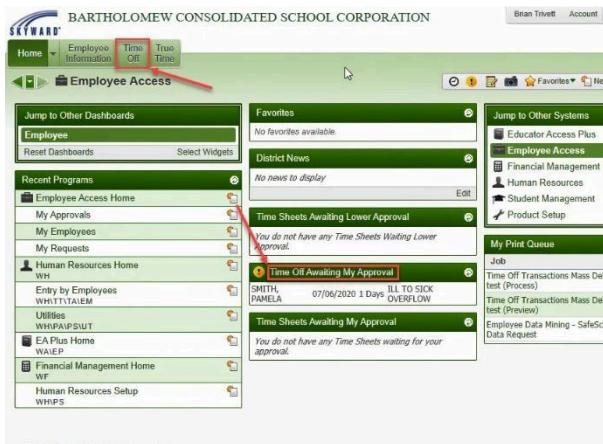
Teacher Portal Page

Figure 2.4.2: Blackbaud



Family Access Page

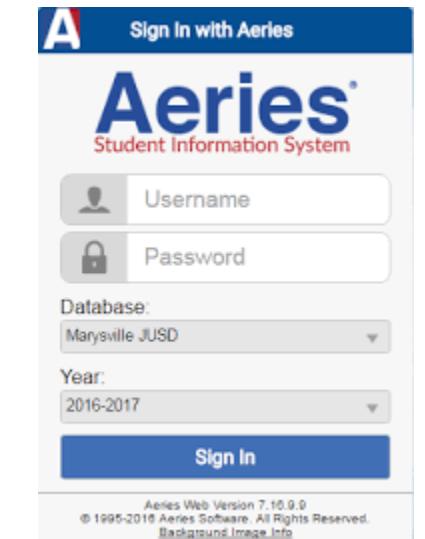
The Family Access page shows a sidebar with links like Home, Online Forms, Calendar, Gradebook, Attendance, Student Info (highlighted with a red box), Schedule, and Homeroom. The main area displays a student profile with fields for Home, Call, and Principal. A red arrow points from the 'Student Info' link in the sidebar to the 'Request Change' button in the right-hand sidebar.



Employee Access Page

The Employee Access page for Bartholomew Consolidated School Corporation shows various dashboard sections. A red box highlights the 'Time Off' and 'True Time' buttons in the top navigation bar. Other sections include Jump to Other Dashboards, Favorites, Recent Programs, and Time Sheets Awaiting Lower Approval.

Figure 2.4.3: Skyward



Login Page

The Aeries login page features a blue header with the text 'Sign In with Aeries'. It includes fields for 'Username' and 'Password', a dropdown for 'Database' set to 'Marysville JUSD', and a dropdown for 'Year' set to '2016-2017'. A large blue 'Sign In' button is at the bottom.



Teacher Portal for Attendance Summary

The Teacher Portal shows an 'Attendance Summary' section with counts for Enrolled (640), Absent (10), and Tardies (12). Below it is an 'Academic Plan Status Summary' table:

Grade Level	Approved	Pending Approval	Pending Submissions	No Action
-1	0	0	0	0
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0

Figure 2.4.4: Aeries

The image displays two side-by-side screenshots of educational software interfaces.

Left Side (Student Information Page):

- Header:** Rediker ACADEMY, Welcome Smith Family, Michele Smith.
- Michele's Progress:**

Subjects	Current Avg
Math	88%
English	93%
Religion	72%
Science	80%
Reading	88.5%
- Michele's Recent Scores:**

Description	Scores
Math - Quiz	88 / 100
English - Test	93 / 100
Religion - Quiz	72 / 100
Science - Quiz	80 / 100
Science - Homework	87 / 100
- Michele's Attendance:**

Date	Attendance
1/1/2013 to Today	4 Tardies 5 Excused Absences 2 Half Absence 2 Tardies 2 Half Absence 2 Tardies
- Michele's Schedule Today:**

Today, 14 February (A Rotation)
08:05-09:15 - Math
09:15-10:15 - Reading
10:20-11:20 - English
11:35-12:30 - Lunch
12:10-13:30 - Social Studies
13:35-14:45 - Religion
- Incidents:**

1/17/2012 - Verbal warning for inappropriate attire - served
1/3/2013 - 1 hr detention for misbehavior - served
2/20/2013 - 1 hr detention for late pass violation - not served

Right Side (TeacherPlus Web Gradebook):

- Header:** REDIKER ACADEMY, 0190/01 English 12 A, Kenneth Adams.
- Contact Details:**
 - E-Mail of contacts: Mother: Jeanette Adams, Father: Tom Adams
 - Phone numbers: (413)444-6207, (413)444-7093
 - Father: Tom Adams, Mr. and Mrs. Tom Adams, 241 Clemont Ave, Springfield, MA 01109, (413)444-6207, (413)444-7091, (413)555-7890
- Class Attendance:**
 - By Date: 04-11-2018, Type: ABSENCE
- Demographics:**
 - Mr. and Mrs. Tom Adams, 241 Clemont Ave, Springfield, MA 01109, (413)444-6207
- Class Attendance:**
 - Total Tardies: 0, Total Absences: 1
- Column Scores:**

Name	Score
Classroom...	81.00
Classroom...	66.00
Classroom...	90.00
Classroom...	57.00
- Score Analysis:**

Bar chart showing student scores across various subjects.

Student Information Page

Figure 2.4.5: Rediker

The image displays two side-by-side screenshots of the Teachmint platform.

Left Side (Login/Signup Page):

- Header:** Login / Signup as Tell Us About You.
- Options:**
 - Admin / Owner (selected, indicated by a checkmark)
 - Teacher
 - Student
- Next:** A blue button at the bottom.

Right Side (Fee Management Option Page):

- Header:** Teachmint, Fee Management.
- Text:** Keep track, reduce costs and eliminate administrative hassle.
- Benefits of modern fee management system:**
 - Lump sum payment solutions
 - Custom reports
 - Customized fee structure
 - Automated fee reminders
 - Custom fee receipts
 - Online fee collection

Login or Sign in Page

Fee Management Option Page

Figure 2.4.6: Teachmint

Student Updates

Events

Alvin DeSilva

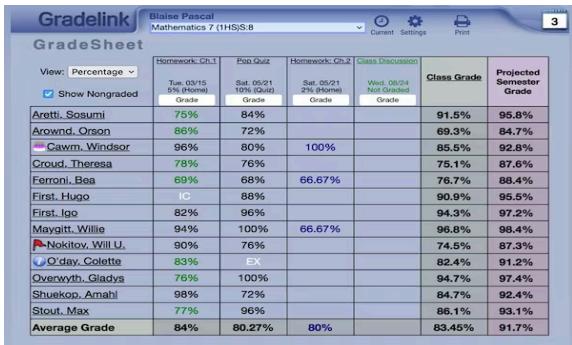
Student Information Page

Figure 2.4.7: QuickSchools

Student Enrollment Page

Login and Sign in Page

Figure 2.4.8: Wisenet

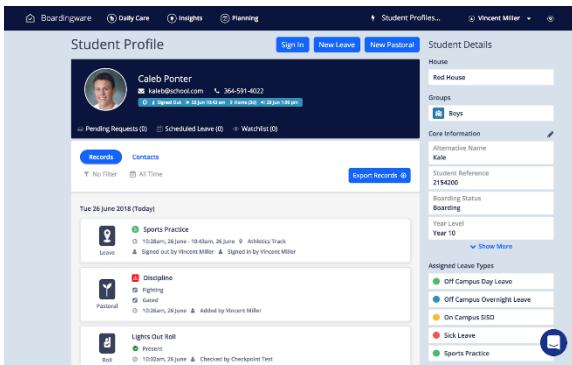


Gradesheet Page

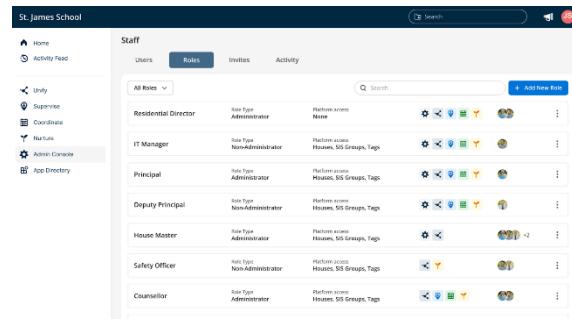


Academic Calendar Page

Figure 2.4.9: Gradelink



Student Profile Page



Staff Page

Figure 2.4.10: Orah

2.5 Summary

In summary, Chapter 2 introduces Student Information System Management and talks about the existing system. This system is a single repository to store all information related to students, such as education and finances. From the study of existing systems, I collected data about the software's functions and security features. For example, enrollment, exam records, role-based access, and HTTPS in terms of service and data protection respectively. Moreover, in Chapter 2, I have also attached some system interface examples. From this chapter we can know about the features and functions used by the existing system in a different way.

Chapter 3

Methodology

3.1 Overview

It is an important task to complete any research or project at each time. Software Development Life Cycle is a methodology useful to design and develop high-quality software after identifying cost-effectiveness and time efficiency. SDLC is used to improve software quality and allocate resources to complete projects. Carrying out robust testing procedures after developing an application will help to produce products with high performance and speed while being user-friendly and error-free. Distributing resources which includes money, assets, and people can give the development team the necessary support to achieve their project goals. Waterfall, spiral, agile, road, incremental and v-model are software development methodologies. Using suitable methodologies depends on the size or complexity of a project, the requirements of shareholders, available resources or budgets and the development team's skills. (LinkedIn, 2023). The discussion below will be based on suitable methodologies for applying for the Student Information Management System.

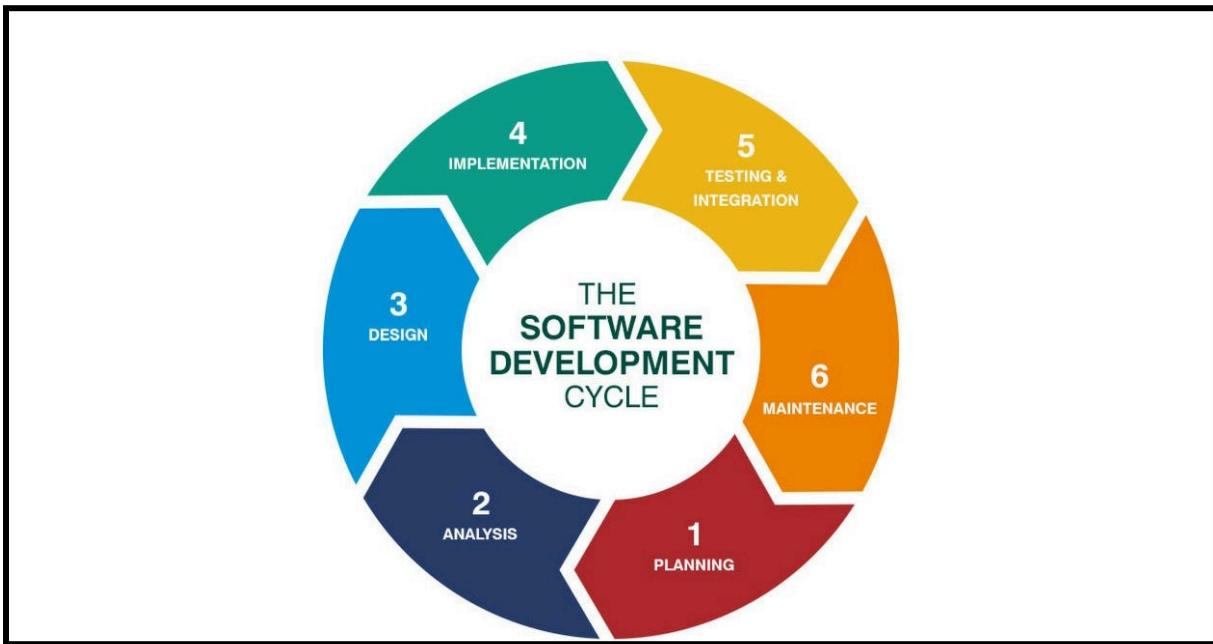


Figure 3.1.1: Process of SDLC

Based on Figure 1 Jilvan Pinheiro, (2018, April 12). Software Development Life Cycle shows the process of iterative development. Planning is the initial phase to identify the project's objectives, goals, and requirements of the stakeholders. Next is analyzing the gathered information from the planning phase. The design phase is to construct the wanted software according to the stakeholders' requirements with available resources. After designing the software, the implementation part comes in to start coding for the developers. The testing part is conducted to examine the developed product and whether it satisfies the goals or objectives of the system without any faults or errors. The deployment phase happens when the system is successfully developed and distributed to the aimed users to check for bugs in the system as a beta version. The last phase is the maintenance, once the software is fully released to the users the developing team must update or upgrade from time to time to encounter any problems that arise.

3.2 Chosen Methodology

For developing the Student Information System, I have chosen the waterfall methodology. The waterfall is a model where the following steps are dependent on the previously completed steps. This ensures that the requirements and functions are identified and completed. The waterfall method is known for being well-documented, enabling stakeholders to have a clear view and understanding of the functions of a project and it can be assured to meet the expectations or goals of the project. Using a waterfall in developing a Student Information System will be useful in beforehand planning and the sequential implementations with managing the budget and time.

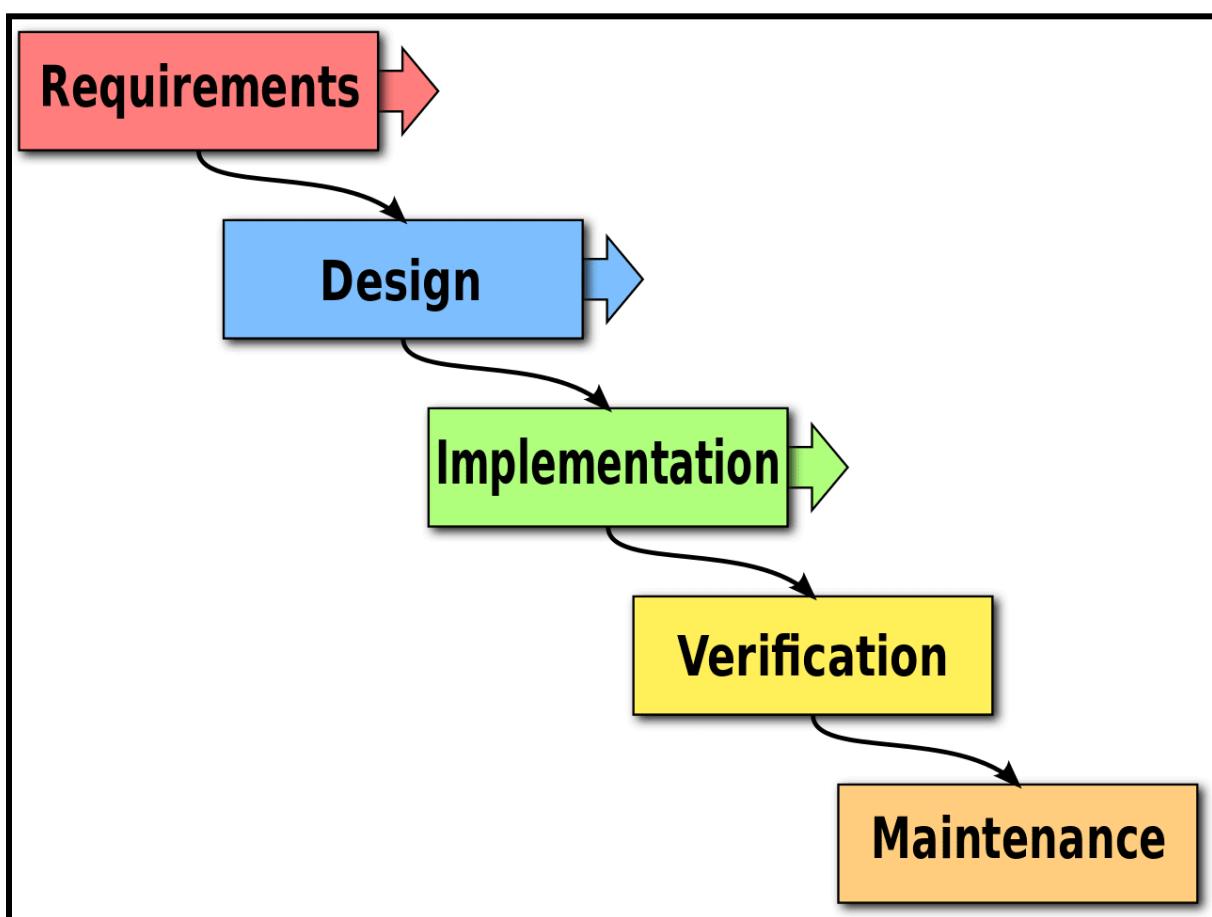


Figure 3.2.1: Waterfall Methodology

3.3 How Does the Waterfall Model Work

Phase 1: Planning

Phase 1 of the waterfall model is conducted to collect and document the information of the developing system in this case Student Information System. Establishing the objectives, problem statement and goal of the project are based on gathered information. Therefore phase 1 is important for the developers to pay attention to proceeding with the assignment.

Phase 2: Analysis

This phase is important to understand a system's requirements. Based on Harshuhi. (2020). What is the Waterfall Methodology or Model in SDLC? In the analysis phase the team will create flow charts, and UML diagrams and will document them to present and review the project with the stakeholders.

Phase 3: Design

In phase 3 once the flow charts and UML diagrams are explained and accepted by the stakeholders the system development is started. A report should be drawn out to explain clearly and easily to use guidelines while creating the model. The creators should be able to think like users in what they expect while using the software to come up with a user-friendly application. At this level, developers are required to create system interfaces such as a sign-in page, login page, and other features according to the system requirements. While designing the interfaces a

system flow also should be created which represents the interaction between the users and the system.

Phase 4: Implementation

Implementation is an important process of SDLC, therefore every step in this phase should be taken effectively to obtain the desired final output as a perfect Student Information System. The first step is writing code according to the designs and requirements using any code editor such as VS Code, Notepad++ and so on. While writing the code, algorithms, databases, and security features should be done simultaneously. Databases are written to store data collected from the Student Information System in one place to use it from one place. Security features must be created in any system to protect the information privacy of the users.

Phase 5: Testing

At the end, before deploying the final product, the system should be tested. This testing consists of detailed procedures on how and who will perform the processes, for example pen testing. Compilation of system testing with success and well-architecture Student Information System ready to deploy into the real world for the users with the awareness of stakeholders.

Table 3.3.1: Advantages and disadvantages of Waterfall Methodology

Advantage	Disadvantage
<ul style="list-style-type: none">• Simple and easy to use. <p>The details are documented straightforwardly after each task completion</p>	<ul style="list-style-type: none">• Long time. <p>Since the project's phase is created after a task is completed it takes more time to complete the entire report for a large system.</p>
<ul style="list-style-type: none">• Well-defined phase <p>The documents are divided into discrete phases such as requirement, planning, design, implementing and testing the system.</p>	<ul style="list-style-type: none">• It's not easy to make changes. <p>Once a phase is done it will be hard to make changes if needed without affecting the entire project's progress</p>

3.4 Requirements

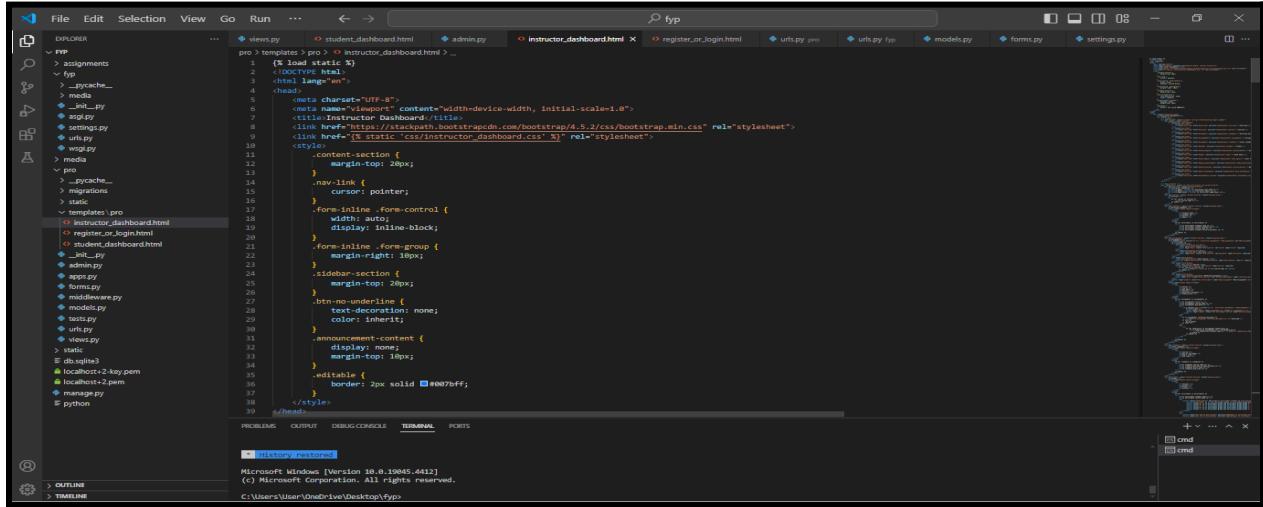


Figure 3.4.1: Visual Studio Code

Visual Studio Code, commonly known as VS code is a source-code editor. This is an open source provided by Microsoft to Windows 10 onwards, OS X 10.11 or later and Linux. This software supports multiple programming languages such as Python, HTML, Java, C++ and many more. VS code provides many features like debugging, auto-indentation, bracket matching, syntax highlighting, etc. Users can install the source from the official website independently. The latest version of this software is 1.85 which was released in November 2023. Download the VS Code installer for Windows. Once it is completed, run the installer by default and it is saved under C:\Users\Username\AppData\Local\Programs.



Figure 3.4.2: Python

There are several types of programming languages that have been used in creating Secured Student Information Management Systems such as Python, HTML, CSS and Java Script. Python is the backend language where the whole has been built upon using it whereas HTML, CSS and Java Script is for the frontend language where it is responsible for the layout or interface creation. Django is a Python high-level open-source web framework. The reason for using Django is the built-in functions such Django Admin, securely automatic hashing, checking password, provides cross-site request forgery protection and provides the support for enabling HTTPS. This high-level framework makes the system more secure. The reason for using frontend language is to create a user-friendly, easy to navigate together to develop structure, designs, and functionality of web server.

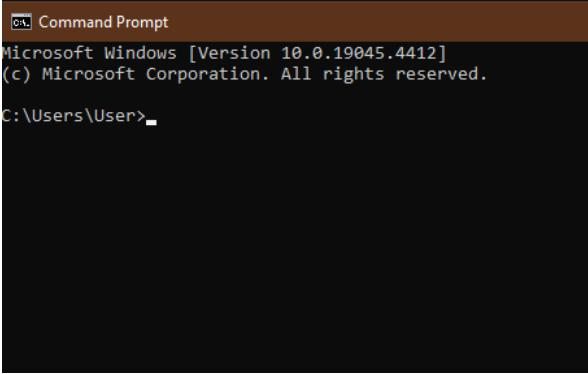
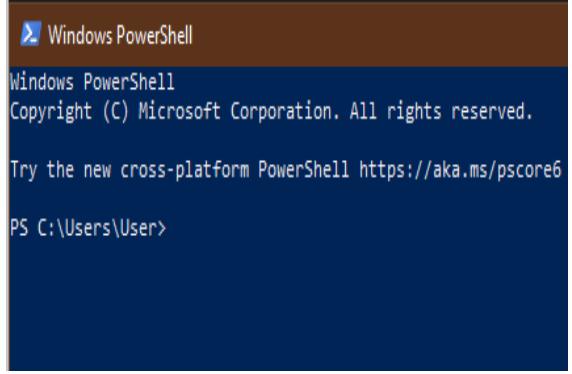
CMD	Windows PowerShell
 <pre>Command Prompt Microsoft Windows [Version 10.0.19045.4412] (c) Microsoft Corporation. All rights reserved. C:\Users\User></pre>	 <pre>Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Try the new cross-platform PowerShell https://aka.ms/pscore6 PS C:\Users\User></pre>

Figure 3.4.3: Command prompt

The command prompt (CMD) is a command-line interface to interact with the operating system and it does basic tasks in a simple and faster way but lacks in advanced functionality. CMD uses simple batch scripting, in tasks such as file manipulation, and simple system commands. Whereas in Windows PowerShell is a task automation and configuration management framework consisting of a command-line shell and an associated scripting language developed by Microsoft, for system administration and automation. PowerShell offers more advanced scripting access to .NET Framework, strong integration with Active Directory Exchange, Windows System and Azure. I used the Windows PowerShell for generating SSL certificate to move from HTTP to HTTPS.



Figure 3.4.4: Device

For this project, I will be using Lenovo IdeaPad with Windows 10 pro version 22H2. This edition was installed on 19/9/2022. This device holds 8.00 GB of Random Access Memory (RAM). Lenovo IdeaPad works on the x64-based processor. Solid state drives are faster than hard disk drives when accessing a large amount of storage in the device. To connect every system to the network NIC a network interface card is needed in a device. Last, hardware requirements for a system depend on the specific tasks and applications planned to use.

3.5 Gantt Chart

Table 3.5.1: Gantt Chart for FYP1

Activities/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
FYP Meeting														
Title finding and Confirming														
Chapter 2 - Literature Review														
Chapter 3 - Methodology														
Chapter 4 - Proposed Design														
Chapter 1 - Introduction														
Chapter 5 - Conclusion														
Finalizing Report														
Preparing for Presentation														
Presentation														

Table 3.5.2: Gantt Chart for FYP2

Activities/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Setting up the project														
Coding														
Testing and Evaluating														
Finalizing Report														
Preparing for Presentation														
Presentation														

As we talked about the planning methodology a Gantt chart is important to estimate how long the project takes to complete. This Gantt chart visualizes the time taken to complete each task. It was a 14-week duration. The rows of this Gantt chart show the tasks and the columns represent the week needed to complete the activities to be carried out in the first phase of the final year project. The bar between the row and columns is the duration taken to finish a task. This chart helps us to understand the progress of the project and be able to finish it in the given time.

There are two gantt chart figures one is FYP1 and FYP2. The timeline for FYP1 starts with confirming a title. After the title is being finalized, Secured Student Information Management System, the literature review is made based on the existing system. Then, evaluating which methodology is suitable for carrying out the project from the day one. For this project Waterfall Methodology have been chosen. Proposing a design on how does the system looks like or functions before developing it and considering all the important requirements for developing the system such as VS code, CMD, hardware requirements and others. Most importantly determining which type of programming language are going to be used in this case, I chose Python programming language, this is because, this language is easy to learn and use which is useful for creating web servers. The next two chapters are Introduction and the

Conclusion. Introduction is the part where project scope, project objectives, and problem statement. This chapter aims to identify the challenges by developing a secured student information system with user-friendly interface. And by identifying the challenges need to find out the project objectives and scope to further improve the system against the challenges. Last chapter is the conclusion. In here we conclude everything have done, researched, and implemented ideas in terms of developing efficient and effective Secured Student Information Management System.

The gantt chart for FYP2 shows the timeline for developing the system fully or partially based on the FYP1. First thing first starting with setting up the project like creating a directory in the device path. After starting the project, we need to perform coding to make all the ideas into functions and resources so that the Secured Student Information Management System is ready to test and evaluate the flow of the system. Once everything settled with the coding and not identifying errors or bugs during the testing then finally, writing a report to finalize every task done during the FYP2. Finally, prepare for the presentation in week 14.

3.6 Summary

In Chapter 3 we have walked through the SDLC methodologies suitable for developing a Student Information System. The chosen method is waterfall. Identifying a suitable methodology initially helps further the project without difficulties. We have also talked about requirements such as hardware in terms of devices, and software such as testing, and programming language. It is important to know about the requirements for smooth execution of the project and optimal user experience. Finally, the Gantt chart was presented in this chapter to understand the timeline and manage the tasks according to the time of completing phase 1 of FYP in creating a Student Information System.

Chapter 4

Implementation

4.1 Overview

The proposed design is a description done on the computer. Developing a well-performing system methodology is important, as discussed earlier in Chapter 3. Designing the program interfaces is important and needs to be done carefully while considering the requirements, objectives, and user expectations. Creating user-friendly software is easy for the users to use. A data flow diagram (DFD) must be constructed apart from interfaces. DFD is a design that explains or can visualize how the data flows through a system based on certain processes or procedures taken inside the program. Next is the unified modeling language UML case diagram to visualize the actors of the system and the actions. DFD focuses on illustrating how data moves and transforms between system operations while flow charts show the actions or decisions made by the users. Next is the UML Case Diagram, which identifies user interactions between users and the system.

4.2 Data Flow Diagram, Flow Chart, and UML

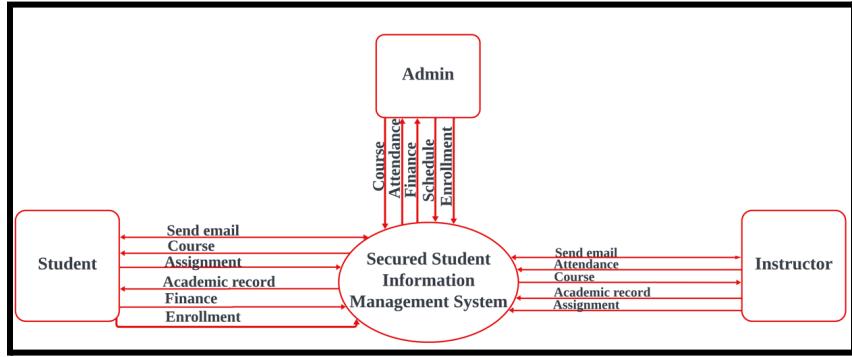


Figure 4.2.1: DFD - Level 0

This figure represents a “Secured Student Information Management System” and illustrates the interactions between different types of user as in (Admin, Student and Instructor) and the system.

Students interact with the system in various ways: they can send and receive emails via the system, access course information, view and manage assignments, and check their academic records to see the grades for the subjects they are registered in. Additionally, students can view and pay their fees through the finance section and enroll in available courses through the enrollment feature.

Instructors interact with the system by sending and receiving emails, recording and referring to student attendance records, and viewing course information. They can manage academic records by adding, updating, and deleting student grades. Additionally, instructors can create assignments and view both the assignments and their submissions by students.

Admins interact with the system by adding, updating, and editing course information and schedules for subjects. They can also view attendance records, enroll students from the admin site, and view the payment history for students.

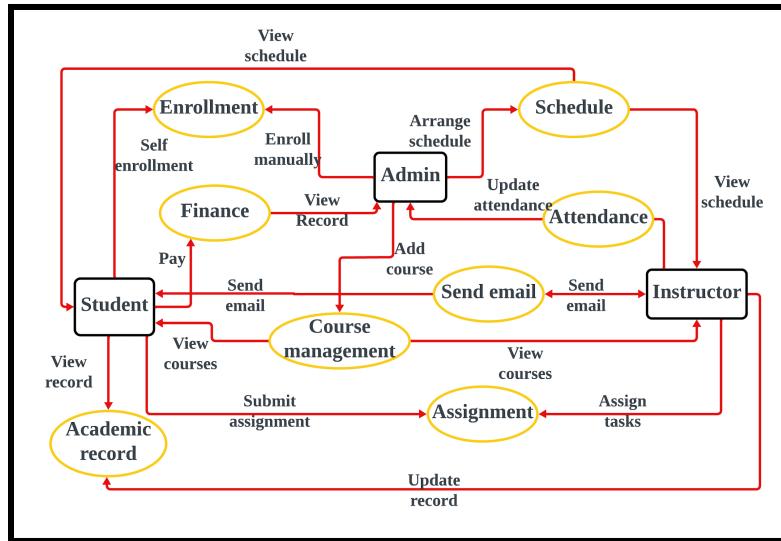


Figure 4.2.2: DFD level 1

DFD level 1 is the enhanced version of level 0 which explains external entities, giving a clear view of the entire system for the stakeholders.

The interaction between Admin and Students involves the Admin managing students' enrollments, schedules, attendance, and course management, all of which are accessible to students. Additionally, the Admin can view emails sent by students. Regarding the interaction between Admin and Instructors, the Admin arranges schedules, enrolls students, manages courses, and views attendance records, and can also view emails sent by instructors.

Communication between Instructors and Students includes the ability for both parties to send emails, with students submitting assignments and instructors creating them. Instructors can also add, update, and delete grades. Student self-services allow students to manage enrollments,

pay fees, view courses, submit assignments, and view their academic records. Instructors are responsible for updating attendance, assigning tasks, and viewing the courses they teach.

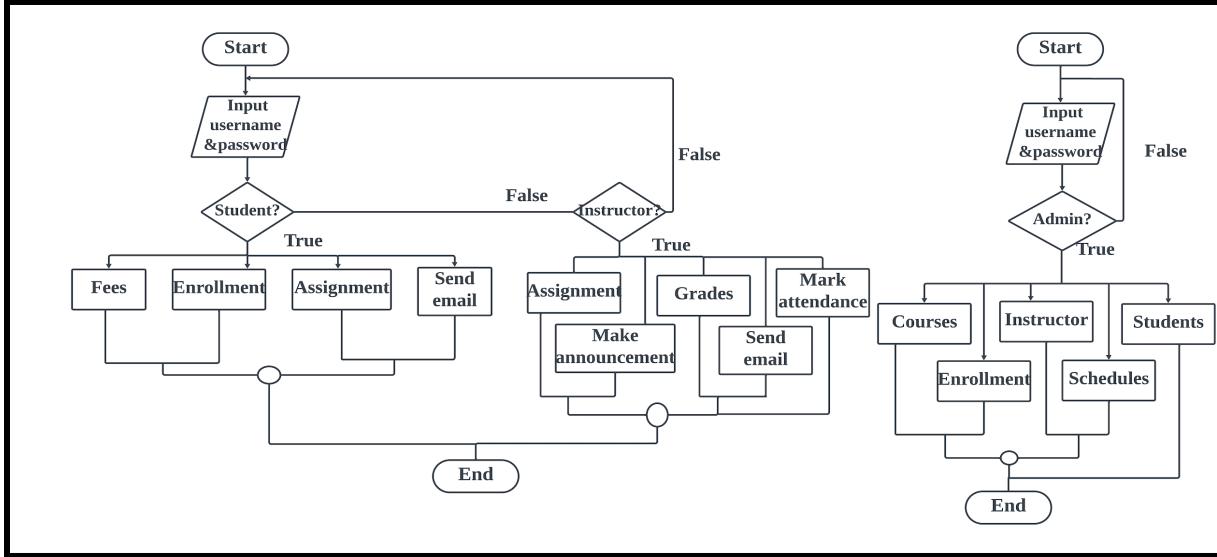


Figure 4.2.3: Flowchart for the entire system

This figure shows the flowchart for Secured Student Information Management System. Upon starting instructors and students are required to input their credentials such as password and username. The system first checks the database if the username and password are matching to begin the authentication process. During the register section the users (instructor or students) are required to enter the university address provided to them by the admin and together with username and password to save it to the database. The registration will be successful only if the university email is saved in the admin database if not the registration will be unsuccessful. Using the entered username and password during the registration the system will grant the authorisation for the users. The system passes the grant based on the keyword (stu and acad) in the email; for students (@stu.uni.edu), for instructors (@acad.uni.edu). So based on the keyword the student will bring to the student dashboard and instructor to the instructor dashboard. Once entered into the system the users can perform the actions that are available in the dashboards. If

the system fails to grant the access then the users have to check the entered password and username or it means that the user is certified to use the system.

As for the admin they are recognised as a super user. Admin does have their own url address. If the username and password are correct then the admin gets the permission to use the system. Admins, upon successful authentication, can manage courses, oversee student enrollments, handle instructor details, manage class schedules and student records and instructor records.

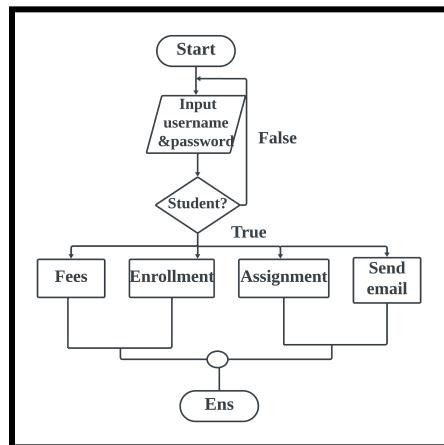


Figure 4.2.4: Flowchart for Students

This flowchart is for students. If the system recognizes the username and password entered as a student then they are allowed to make actions to pay fees according to the number of subjects and the amount for the subjects, enroll into subjects that are available, submit assignments, and send email to the instructors

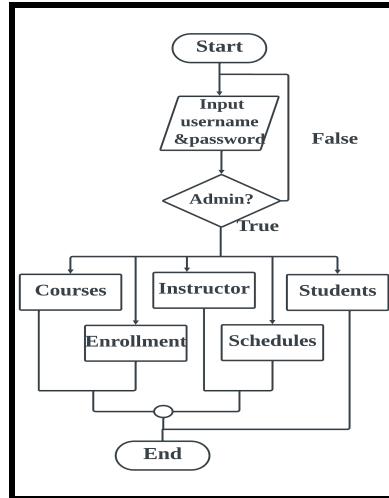


Figure 4.2.5: Flowchart for admin

This flowchart is for the admin to manage the students and instructors records. The admin can adjust the class schedule, add course details such as code, name, and fees. Admin can also add or delete and edit the students and instructor details if the students changed the major after some time and generates new student emails with new majors and random words or numbers. As for instructors, the admin can change the subject they have to teach in the future. Enrollment from the admin can be done for the students.

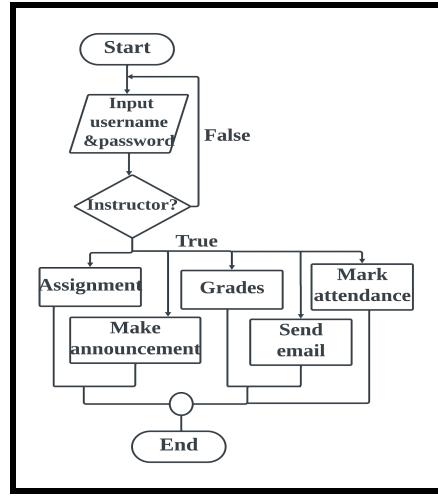


Figure 4.2.6: Flowchart for instructor

This flowchart is for the instructor to make actions according to the features. Instructors can create assignments for the students who are enrolled under them for specific subjects. Grades can be added, edited and updated. Instructors can mark the attendance, and send email to the students.

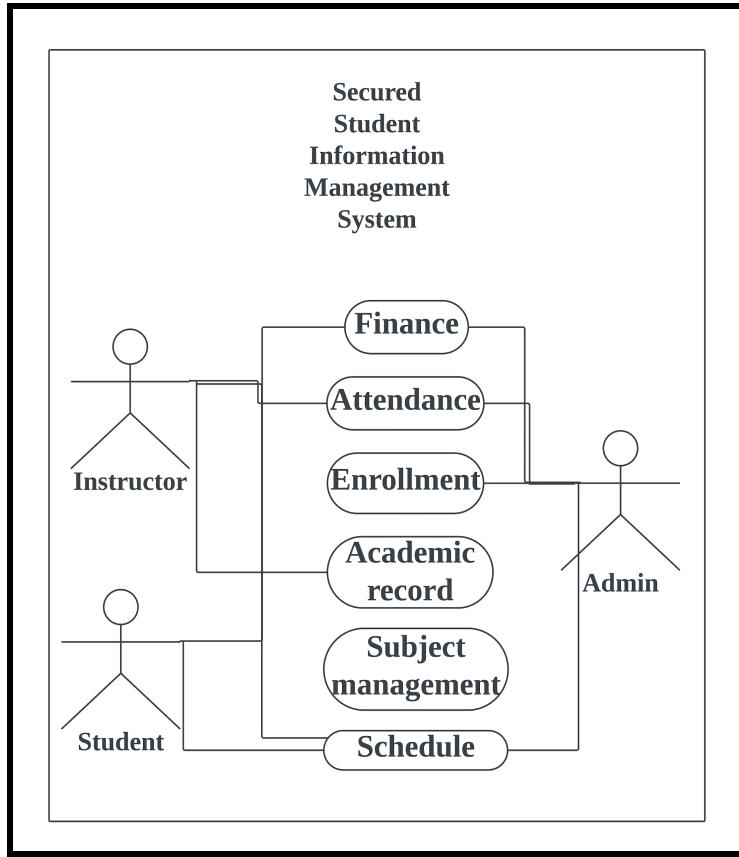


Figure 4.2.7: UML Case Diagram

The figure effectively showcases the central components of a Secured Student Information Management System and highlights the interactions each user role has with these components. Instructors focus on teaching-related activities, students on their academic and financial activities, and admin on overall system management and maintenance. This structure ensures a well-organized, role specific access to various functionalities within the SIMS, promoting efficiency and security.

4.3 HTTPS Implementation

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
WARNING: 'choco' was found at 'C:\ProgramData\chocolatey\bin\choco.exe'.
WARNING: An existing Chocolatey installation was detected. Installation will not continue. This script will not
overwrite existing installations.
If there is no Chocolatey installation at 'C:\ProgramData\chocolatey', delete the folder and attempt the
installation again.

Please use choco upgrade chocolatey to handle upgrades of Chocolatey itself.
If the existing installation is not functional or a prior installation did not complete, follow these steps:
- Backup the files at the path listed above so you can restore your previous installation if needed.
- Remove the existing installation manually.
- Rerun this installation script.
- Reinstall any packages previously installed, if needed (refer to the lib folder in the backup).

Once installation is completed, the backup folder is no longer needed and can be deleted.
```

Figure 4.3.1: Install Chocolatey

To use this Chocolatey we need to use Windows PowerShell instead of cmd. This is because the Chocolatey installation script is written for PowerShell, making it more reliable to run the command (**Set-ExecutionPolicy Bypass -Scope Process -Force**) which sets the execution policy to bypass restrictions for the script execution. This command is only for temporary use and only affects the current session. Once the bypass have done then need to run the code to install Chocolatey using (**[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]::Tls12;iex((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))**). This command is to ensure that the script uses the TLS 1.2 protocol for secure downloads.

```
PS C:\WINDOWS\system32> choco -v  
2.2.2
```

Figure 4.3.2: Chocolatey verification

After the installation is complete, I need to verify whether the Chocolatey installation is successful or not by running the command (choco -v). If the installation is successful then version of Chocolatey will be shown here it shows as 2.2.2

```
PS C:\WINDOWS\system32> choco install mkcert  
Chocolatey v2.2.2  
Installing the following packages:  
mkcert  
By installing, you accept licenses for the packages.  
mkcert v1.4.4 already installed.  
Use --force to reinstall, specify a version to install, or try upgrade.  
  
Chocolatey installed 0/1 packages.  
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).  
  
Warnings:  
- mkcert - mkcert v1.4.4 already installed.  
Use --force to reinstall, specify a version to install, or try upgrade.
```

Figure 4.3.3: mkcert installation

In the same command after finding out the version of Chocolatey then need to run the command (**choco install mkcert**). mkcert is a simple tool to make it easier to develop a system locally with HTTPS from HTTP by generating SSL certificates.

```
PS C:\WINDOWS\system32> mkcert -install  
The local CA is already installed in the system trust store! ☀
```

Figure 4.3.4: Install of CA

This command will generate and install the local CA in the system. CA is a digital certificate authority that is set up and trusted locally on a developer's machine. It is used to provide SSL/TLS certifications for local development to secure communications.

```
PS C:\WINDOWS\system32> mkcert localhost
```

Figure 4.3.5: CA for localhost

To make the SSL work in the django project, I need to install django extensions which support the local CA. The installation command is (pip install django-extensions). And then finally can run the development server with SSL certificate with (**python manage.py runserver_plus --cert-file localhost.pem --key-file localhost-key.pem**). Once having configured the server to use the generated certificates, then the locally developed system can run over HTTPS. This command generates two files 'localhost.pem' and 'localhost-key.pem'.

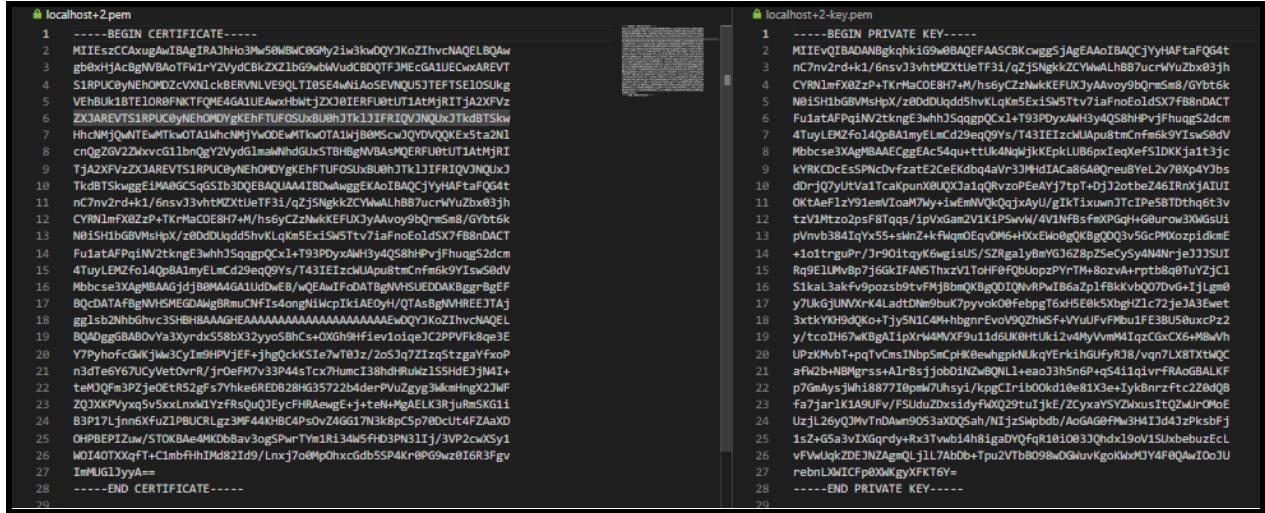


Figure 4.3.6: SSL certificate and the key

```
C:\Users\User\Desktop\fyp>python manage.py runsslserver --certificate "C:\Users\User\fyp\localhost+2.pem" --key "C:\Users\User\fyp\localhost+2-key.pem"
Watching for file changes with StatReloader
Validating models...

System check identified no issues (0 silenced).
June 09, 2024 - 22:19:43
Django version 3.0.6, using settings 'fyp.settings'
Starting development server at https://127.0.0.1:8000/
Using SSL certificate: C:\Users\User\fyp\localhost+2.pem
Using SSL key: C:\Users\User\fyp\localhost+2-key.pem
Quit the server with CTRL-BREAK.
```

Figure 4.3.7: SSL server running

This figure proves that the system server is successfully running with HTTPS enabled, using the SSL certificate and the responding key. (<https://127.0.0.1:8000/>) is the URL format to open the server in browsers. HTTPS indicates that the information is secured throughout the sessions while 127.0.0.1 indicates the localhost IP address. ‘8000’ indicates the default port number used by Django’s development server.

4.4 Password Hashing

Secured Student Information Management System is built based on Python Django high-level framework. Therefore, Django automatically uses a hashing algorithm for securely saving a user's password in a way it is not readable by anyone for security purposes. When the user creates a password during the registration session, Django's built-in authentication system hashes the password before saving it to the database and makes it encrypted. This ensures that the plaintext is converted into ciphertext by providing a higher-level of security from being compromised by a third party or hacker. By default the latest version of Django uses the PBKDF2 algorithm with a SHA256 and salt.

The screenshot shows the Django admin interface for changing a user. The top bar says 'Change user'. Below it, the username 'AI76d6@stu.uni.edu' is displayed. A form field for 'Username' contains 'AI76d6@stu.uni.edu' with the placeholder 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.' To the right of the form, the hashed password is shown: 'algorithm: pbkdf2_sha256 iterations: 720000 salt: M8H85D***** hash: 6CRXWE*****'. A note at the bottom states: 'Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.'

Figure 4.4.1: Password hashing

This is how the password hashing happens in the Django project. The hashed password and username saves in the admin site interface only in the database for future extractions.

4.5 Time-out Session Implementation

```
SESSION_EXPIRE_AT_BROWSER_CLOSE = True
SESSION_COOKIE_AGE = 300
SESSION_SAVE_EVERY_REQUEST = True
SESSION_ENGINE = 'django.contrib.sessions.backends.db'
```

Figure 4.5.1: Time-out session implementation

This session time-out durations and automatic expiration of URL can help prevent risks related to session hijacking. A user logs into the application, and a session is created with a lifetime of 5 minutes. Everytime the user performs an action the time resets the session expiration timer to another 5 minutes. If the user closes the running browser, the session cookie is deleted, and the session is effectively ended. At the same time if the user is inactive for 5 minutes by not performing any actions the session automatically expires and the user will news to login again.

4.6 CSRF protection

The CSRF token is used to protect web systems from Cross-Site Request Forgery attacks. The CSRF token generated when a user loads a page with a form, and includes it in the form. When the user submits the form, the CSRF token is sent along with the form data, then the system verifies the token with each form submitted. If the token with each form missing or not valid the server rejects the request to load the server.

```
Accept: text/css,*/*;q=0.1
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-GB,en;q=0.9,en-US;q=0.8
Connection: keep-alive
Cookie: csrfToken=eDKIDM6ogG0dLqtnXBOBAkyZ43fAGnUg; sessionId=l3affqegn8rt704swq8lmls2yoj0wri5
Host: 127.0.0.1:8000
Referer: https://127.0.0.1:8000/student_dashboard/17/
Sec-Ch-Ua: "Microsoft Edge";v="125", "Chromium";v="125", "Not A/Brand";v="24"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: style
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36 Edg/125.0.0.0
```

Figure 4.6.1: HTTPS request headers

This figure provides the HTTPS request headers for a network request, including the CSRF token and session ID in the cookies. This proves that the CSRF token is being generated and included in the request cookies.

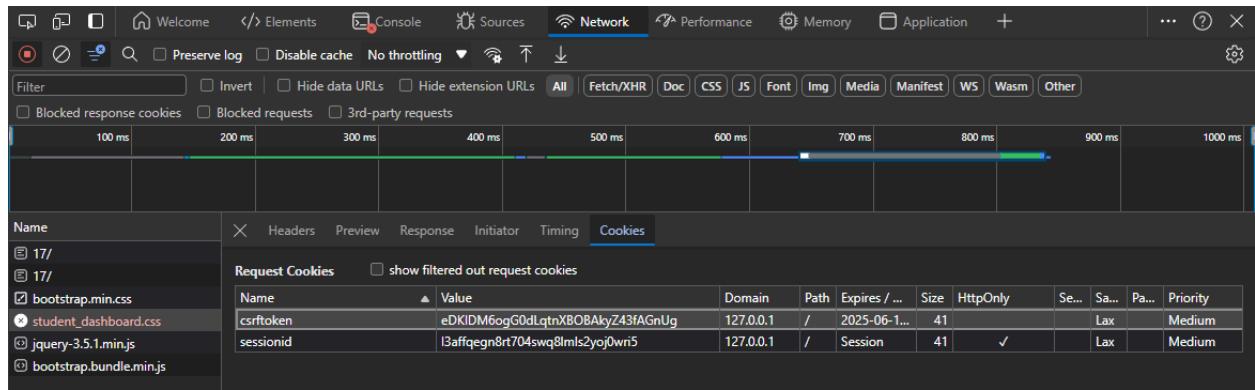


Figure 4.6.2: Cookies is being sent

This figure shows the cookies being sent with a network request, including the 'csrfmiddlewaretoken' cookie. This indicates that the CSRF token is being generated and sent by the server.

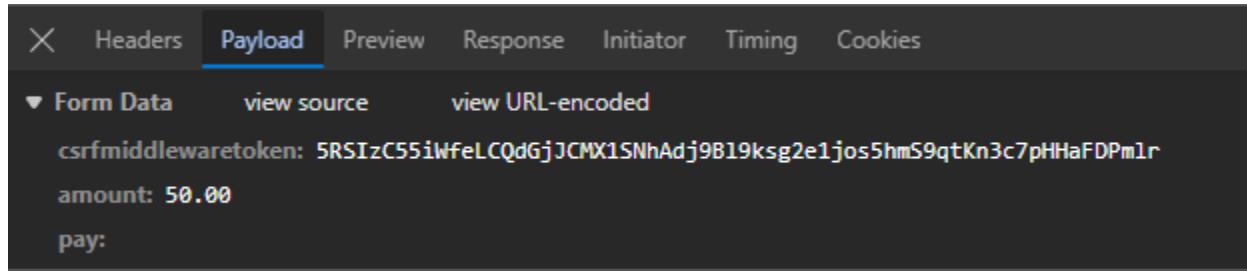


Figure 4.6.3: CSRF in the payload

The figure shows the form data payload in a network request, which includes the 'csrfmiddlewaretoken'. This indicates that the CSRF token is being correctly included in the form data when the form is submitted. This proves that the generation of CSRF tokens works perfectly.

4.7 Database

The Secured Student Information Management System runs in the SQLite3 database. Which is the default database for the Django project. This database is typically used for development because it is a light-weight and easy to configure and handle and fully self-contained. Every data or information is saved in one file inside the project directory called as db.sqlite3.

4.8 Summary

In summary, this chapter 4 Implementation discusses the implementation that has been done in terms of flow of the system, DFD and the UML case diagram with actors and actions can be made by them. Apart from that this chapter also discusses about how the HTTPS have implemented to the system, hows does the hashing passwords works, time-out session for the system to stay alive to prevent from session hijacking attacks, and the CSRF token generation and what does the generated token does to the system. Implementing more security measures in a system makes the server to be more secure and free from being hacked by outsiders or even from internal parties. Therefore after performing all these security functions, we were able to develop a Secured Student Information Management System making it safe for the users ro use it.

Chapter 5

Testing and Discussion

5.1 Overview

The chapter Testing and Discussion is an important section of the software development process, ensuring that the system functions works perfectly fine and according to the users actions. Testing is the process of evaluating a system or its components to determine whether it satisfies specified requirements. It involves executing the software to find any errors or missing steps or requirements and rectifying the problems.

5.2 Interface Testing

The image shows a registration form titled "Welcome" with the sub-instruction "Please login or register to continue". It features two buttons at the top: "Register" (highlighted in blue) and "Login". Below these are four input fields: "Username", "Email", "Password", and "Confirm Password", each with a corresponding text input box. At the bottom is a large blue "Register" button.

Figure 5.2.1: Register page

The figure shows the register page for instructor and student. Username is self entered input whereas email is provided by the admin upon registering to the system for the first time. If the entered email is not in the database then automatically the user cannot save the password and cannot login to the system. If the email is present in the database then the user can successfully enter the password. Then the password will be hashed before saving into the database.

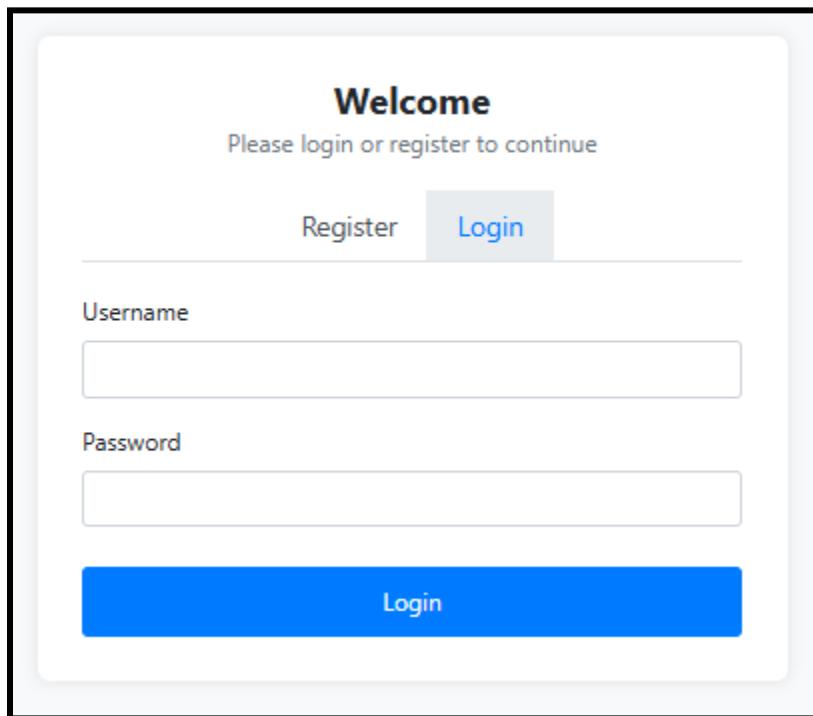


Figure 5.2.2: Login page

This image shows the login page with Username which is the email address and along with the Password. Once the password matches the username, the system will grant the authorization to use it. The authorization is given based on the RBAC which means Role Based Authentication Control. If the username contains the keyword 'stu' for example (AI76d6@stu.uni.edu) it will treat this user as a student therefore it is brought to the student dashboard. While, for an instructor if the username contains the word 'acad' for example (sam@acad.uni.edu) brings to the instructor dashboard.

Student Dashboard

[Profile](#)[Courses](#)[Instructors](#)[Fees](#)[Enrollment](#)[Grades](#)[Assignments](#)[Notifications](#)[Send Email](#)[Sent Emails](#)[Class Schedule](#)

Student Profile

Name: Hema
Email: AI76d6@stu.uni.edu
Student ID: AI76d6
Registration Date: June 5, 2024, 4:36 a.m.

Figure 5.2.3: Student dashboard

This page is for the student user. The contents in the ‘Profile’ section shows the name, email address and the student ID plus the registration date when a new student has been added to the database in the admin site. The sidebar shows the features that a student can access.

Courses

- Computer Security

Figure 5.2.4: Course

This section is for the list of courses the student has been enrolled into through admin or self enrollment.

Instructors

- Sam - sam@acad.uni.edu

Figure 5.2.5: Instructor

This screenshot shows the list of instructors, the student has been enrolled based on the subjects they have enrolled

Fees

Total Fees: 500.00

Total Paid: 300.00

Remaining Balance: 200.00

Amount

Pay

Show Payment History

Payment History

- June 7, 2024, 3:29 a.m. - 250.00
- June 9, 2024, 9:06 a.m. - 50.00

Figure 5.2.6: Fees Payment

In this section the student can view their total amount of fees, total paid and the remaining amount that needs to be paid. At the same time the student can make the payment and view their payment history for any further references. Once the payment has been done automatically will happen in the admin site to keep track of the students financial information.

The image shows a user interface for enrolling in a course. At the top, the word "Enrollment" is displayed in a large, bold, dark blue font. Below it, the text "Select Course" is shown in a smaller, dark blue font. A horizontal input field contains the text "Computer Security". At the bottom of the form is a blue rectangular button with the word "Enroll" in white.

Figure 5.2.7: Enrollment

This section is useful for students to enroll for subjects themselves and the update will be automatically done in the admin site. The student is not allowed to enroll in the subject registered under their name.

Grades	
Course	Grade
Computer Security	None

Figure 5.2.8: Grade

In this section, the student can view their grades updated by the instructor for a subject. The 'None' shows that the instructor has not updated the grade for the respective student.



Figure 5.2.9: Notifications

This figure shows the notification whenever an instructor sends an email and makes an announcement for a student. After reading the announcement the student can press the 'Mar as Read' button and after reading the student either can delete or keep it as long as they want.

A screenshot of a 'Send Email to Instructors' form. The title is at the top. Below it are three input fields: 'Recipient Email' containing 'sam@acad.uni.edu', 'Subject' containing 'Leave Permission', and 'Message' containing 'Good Day sir'. Each field has its label to its left.

Figure 5.2.10: Send email

In this page the student can send an email to a particular instructor by typing manually an instructor email address. After completing the typing the cursor must be placed inside the 'Message' box and then hit the 'Enter' key in the keyboard to successfully send the email if the

student wants to move to the new line, then the user must press the ‘Shift’ together with the ‘Enter’ key.

Sent Emails				
Recipient	Subject	Message	Sent At	Action
sam@acad.uni.edu	Leave Permission	Good Day sir	Jun 09, 2024, 08:55 a.m.	<button>Delete</button>

Figure 5.2.11: Sent email

This section shows the list of sent email with the recipient, subject, message and sent at field and plus with the ‘Delete’ button where the students can delete the sent email if they want to.

Class Schedule			
Course	Day of the Week	Start Time	End Time
Computer Security	Monday	5:11 a.m.	7:11 a.m.

Figure 5.2.12: Class schedule

The class schedule information is retrieved from the admin site. The class schedule appears if the student is enrolled into that certain class otherwise it would not appear in the class schedule section in the student dashboard.

Instructor Dashboard

Name: Sam

Email: sam@acad.uni.edu

Department: FIST

Figure 5.2.13: Instructor dashboard

This section shows the name, email address and the department they are teaching the students.

Courses

- Computer Security

Figure 5.2.14: Courses

This section shows the list of course names where an instructor has been added to a particular subject in the administration site.

Enrolled Students

Student Name	Student ID	Course	Email
Hema	AI76d6	Computer Security	AI76d6@stu.uni.edu

Figure 5.2.15: Enrolled students

This page is for the instructor to view the students enrolled under a specific instructor. This page includes the student's name, student id, course and the student email address.

Assignments

Title Due date mm/dd/yyyy

Description

Course Computer Security

Reference Document Choose File No file chosen

Title	Course	Due Date	Actions	Reference Document	Submissions
Flowchart	Computer Security	July 22, 2024, midnight	<input type="button" value="Delete"/>	<input type="button" value="Download"/>	• Hema - Download
Flowchart	Computer Security	June 24, 2024, midnight	<input type="button" value="Delete"/>	<input type="button" value="Download"/>	

Figure 5.2.16:Assignment

This section is for the instructor to give an assignment for the student and the instructor also can choose a file from their device for student's reference. Below the assignment assigning section, the instructor can view the submitted assignment and download the file from the student to assess it.

Class Schedule			
Course	Day of the Week	Start Time	End Time
Computer Security	Monday	5:11 a.m.	7:11 a.m.

Figure 5.2.17: Class schedule

This page has the course name, day of the week, start and end time. From this page, the instructor can view the timetable for their classes. The schedule is adjusted by the admin in the administration page.

Grades			
Student	Course	Grade	Actions
Hema	Computer Security	None	<button>Add</button> <button>Update</button> <button>Delete</button>

Figure 5.2.18: Grades

This Grades section is used for adding, updating and deleting the grades according to the student's name and course. There are six types of grade options which are A, B, C, D, F, and None. If the instructor adding grades for a student for the first time then the 'Add' button must be pressed, after that if the instructor would like to change the grades means need to press the 'Update' button and the 'Delete' button to delete the current grade and changes into 'None' word in the instructor dashboard while in the student dashboard does not show anything.

Send Email to Students

Recipient Email	
Subject	
Message	

Figure 5.2.19: Send email

This section is for instructors to send email to students. In the recipient email the instructor can manually enter the email address; for the email address they can get it from the ‘Notifications’ sections where an email is sent to the instructor. This send email to students is kind of replying back in another section not straight from the ‘Notification’ section.

Sent Emails				
Recipient	Subject	Message	Sent At	Action
AI76d6@stu.uni.edu	FYP	FINAL	Jun 09, 2024, 09:40 a.m.	Delete

Figure 5.2.20: Sent emails

This page is for the instructor’s reference for viewing with the sent email for students and the users can delete the emails. In this section the instructor can see the receiver's email address, subject, the content of message and the time when the message sent by the instructor.

Make Announcement

Title

Content

Make Announcement



Figure 5.2.21: Make announcement

This page allows the instructor to make announcements for all the students. When the announcement has been made, it will be sent to the student's 'Notifications' section where the students will be able to view the announcement.

Mark Attendance

Select Course

Select Students

Select Schedule

Date

Status

Mark Attendance



Figure 5.2.22: Mark attendance

This page is for instructors to perform attendance marking with the option of selecting course, selecting multiple students in one go, selecting schedule if the time needs to match with the course type, the current date of having the class and selecting the status between 'Present' and 'Absent'. If the student is in the class, then go for 'Present' if not 'Absent' and then finally press, 'Mark Attendance' button to save it to the 'Attendance Record' and automatically the attendance will be saved in the admin site.

Attendance Records				
Student	Course	Schedule	Date	Status
Hema	Computer Security	Monday - 5:11 a.m. to 7:11 a.m.	June 6, 2024	Present

Figure 5.2.23: Attendance record

This is where the student attendance is recorded in the instructor dashboard.

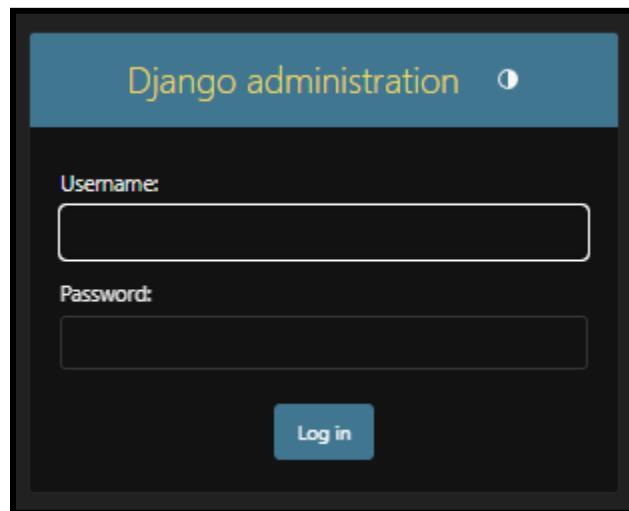


Figure 5.2.24: Django admin

This interface is used by the admin to get access to the admin site by entering the username and matching password.

Select course to change				ADD COURSE +
				FILTER
				<input type="checkbox"/> Show counts <input checked="" type="checkbox"/> By credit hours All 4
Action:	-----	Go	0 of 1 selected	
NAME	CODE	CREDIT HOURS	INSTRUCTOR	
Computer Security	CS2411	4	Sam	
1 course				

Figure 5.2.25: Course

In this page the admin can view the list of courses that have been added into the database.

Add course												
Code:	<input type="text"/>											
Name:	<input type="text"/>											
Credit hours:	<input type="text"/>											
Instructor:	-----	<input type="button" value="▼"/>	<input type="button" value="✎"/>	<input type="button" value="+"/>								
FEES <table border="1"> <thead> <tr> <th>AMOUNT</th> <th>DELETE?</th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="button" value="X"/></td> </tr> </tbody> </table> <input type="button" value="+ Add another Fee"/>					AMOUNT	DELETE?	<input type="text"/>	<input type="button" value="X"/>				
AMOUNT	DELETE?											
<input type="text"/>	<input type="button" value="X"/>											
SCHEDULES <table border="1"> <thead> <tr> <th>DAY OF WEEK</th> <th>START TIME</th> <th>END TIME</th> <th>DELETE?</th> </tr> </thead> <tbody> <tr> <td>-----</td> <td><input type="text"/> Now <input type="button" value="○"/></td> <td><input type="text"/> Now <input type="button" value="○"/></td> <td><input type="button" value="X"/></td> </tr> </tbody> </table> <input type="button" value="+ Add another Schedule"/>					DAY OF WEEK	START TIME	END TIME	DELETE?	-----	<input type="text"/> Now <input type="button" value="○"/>	<input type="text"/> Now <input type="button" value="○"/>	<input type="button" value="X"/>
DAY OF WEEK	START TIME	END TIME	DELETE?									
-----	<input type="text"/> Now <input type="button" value="○"/>	<input type="text"/> Now <input type="button" value="○"/>	<input type="button" value="X"/>									
<input type="button" value="SAVE"/> <input type="button" value="Save and add another"/> <input type="button" value="Save and continue editing"/>												

Figure 5.2.26: Add course

In this page the admin allowed to add new courses by filling up the Code, Name, Credit Hour, and Instructor. At the same time, the admin can add in the teaching amount for that course and adjust the scheduling. After creating the course, in the instructor dashboard, the instructor will see the name of the course they have been assigned and the schedule for that subject. Whereas in the student dashboard, if the student enrolled into newly created subject, then the

name of the course will be added into the course section and the amount of fees will be also update according to the amount have assigned for that subject by the admin and the student can view the schedule for enrolled subject which also adjusted by the admin.

The screenshot shows a dark-themed web interface titled "Select enrollment to change". At the top right is a button labeled "ADD ENROLLMENT +". Below it is a search bar with a magnifying glass icon and a "Search" button. Underneath is a table header with columns: "Action", "STUDENT", "COURSE", and "INSTRUCTOR". A dropdown menu under "Action" shows "-----". The "STUDENT" column lists "Hema". The "COURSE" column lists "Computer Security". The "INSTRUCTOR" column lists "Sam". Below the table, a message says "1 enrollment".

Figure 5.2.27: Enrolled students

This image shows the enrolled students detail such as student name, course have enrolled and the respective instructor for the enrolled subjects

The screenshot shows a dark-themed web interface titled "Add enrollment". It has three input fields: "Course:" with a dropdown menu, "Student:" with a dropdown menu, and "Instructor:" with a dropdown menu. Each field has a pencil icon, a green plus sign icon, and an eye icon. At the bottom are three buttons: a blue "SAVE" button, a light blue "Save and add another" button, and a light blue "Save and continue editing" button.

Figure 5.2.28: Add enrollment

In this section, the admin can enroll the student according to a certain subject and assign the instructor. Then the name of the instructor will appear in the student dashboard and the name of the student will appear in the instructor dashboard.

The screenshot shows a dark-themed web interface for adding fees. At the top left, it says "Add fee". Below that is a "Course:" dropdown menu with a small downward arrow icon. To the right of the dropdown are three icons: a pencil, a green plus sign, and an eye. Underneath the dropdown is a "Amount:" input field with a placeholder value of "0.00". At the bottom of the form are three buttons: "SAVE" (in white text), "Save and add another" (in white text), and "Save and continue editing" (in white text).

Figure 5.2.29: Add fees

This figure shows the add fee for available courses, if want to edit or change can perform the adjustment also. In this section, the admin can add the course amount if the fees have not been added during the add course section.

The screenshot shows a dark-themed web interface for adding instructors. At the top left, it says "Add instructor". Below that are three input fields: "Full name:" with an empty text input, "Email:" with an empty text input, and "Department:" with an empty text input. At the bottom of the form are three buttons: "SAVE" (in white text), "Save and add another" (in white text), and "Save and continue editing" (in white text).

Figure 5.2.30: Add instructors

In this figure, the admin is required to enter the instructor name, email with (acad.uni.edu) and the department they work in.

The screenshot shows a payment history interface. At the top, a search bar and a filter section are visible. The filter section includes options like 'Show counts', 'By date', 'Any date', 'Today', 'Past 7 days', 'This month', and 'This year'. Below the filter is a table with the following data:

STUDENT	AMOUNT	DATE	TRANSACTION ID
Hema	250.00	June 7, 2024, 11:29 a.m.	95e4ca74-68bf-4e73-ba25-dfa3727eca
Hema	50.00	June 9, 2024, 5:06 p.m.	4e9ba679-980e-4898-af49-59666271c2dd

At the bottom left, it says '2 payments'.

Figure 5.2.31: Payment by the students

The payment information reflects from the student site whenever the student pays the amount from the student dashboard. This image contains the student name, amount paid, date and the transaction id.

The screenshot shows an 'Add schedule' form. It has four main input fields: 'Course' (dropdown menu), 'Day of week' (dropdown menu), 'Start time' (text input with 'Now' button), and 'End time' (text input with 'Now' button). At the bottom, there are three buttons: 'SAVE', 'Save and add another', and 'Save and continue editing'.

Figure 5.2.32: Add schedule

The admin can adjust the schedule to the available courses only with specification of days of week, start time and end time. For the time the admin are advised to enter the time in (H:M:S)

A screenshot of a web-based student management system. The title bar says "Select student to change". On the right, there's a "FILTER" sidebar with sections for "Show counts", "By faculty" (with "All" and "FIST" options), and "By major" (with "All" and "AI" options). The main area shows a table with one student record:

Action	NAME	FACULTY	MAJOR	STUDENT ID	UNIVERSITY EMAIL	REGISTRATION DATE
<input checked="" type="checkbox"/>	Hema	FIST	AI	AI76d6	AI76d6@stu.uni.edu	June 5, 2024, 12:36 p.m.

Below the table, it says "1 student". At the top right, there's a "ADD STUDENT +"

Figure 5.2.33: Student lists

This section shows the student name, major, faculty, student id, university email which was generated randomly and the registration date.

5.3 Summary

In the chapter "Testing and Discussion," we show the results of performing several actions by the three main users: Admin, Student, and Instructor. Testing the system ensures that each and every requirement is met with the functions and features applied in the system. Testing ensures that new data does not corrupt or delete existing records. Apart from understanding the functions and features, it is crucial to know what the Secured Student Information Management System (SIMS) should do. Successful testing and discussion of the system prove that it runs without errors and addresses any debug problems that arise during development. The above discussion on testing the system demonstrates how every feature works accordingly. In conclusion, the Secured Student Information Management System is robust, secure, and effectively supports the flow of the system with everyone's actions.

Chapter 6

Conclusion

6.1 Overview

In conclusion, successfully implementing the Student Information System will significantly improve the efficiency and effectiveness of managing student-related information within an educational institution. The system streamlined administrative processes, and the accuracy of data, providing timely access to collected information through the system for students, administrators, and academic staff.

6.2 Conclusion

In conclusion, the development of the Secured Student Information Management System marks a significant importance in improving the access and management of student information. This system integrates with some of the features not only the information but such as creating assignments, sending email, submitting assignments, self enrollment and with robust security measures, ensuring that all users as in Admin, Students and the Instructors can interact with the system efficiently and securely.

The key achievements of this system is comprehensive user management for the admins who are controlling the information about every user in one server. Admins have the full authority over user creating, updating and deleting. While performing any new actions the system ensures that these operations do not interfere with existing records or without knowingly the data gets deleted, by maintaining the data integrity securely.

By using this system the students can effortlessly enroll in courses that are available in the database as a self-enrollment, submit assignments according to the assignment provided by the instructors, and keep track of the sent emails, notifications from instructors, grades and the fees payment as well. This is an enhanced experience for the student for using the system seamlessly.

Apart from the students and admin views, the instructors also can make use of this Secured Student Information Management System. Instructors benefit from the functions that allow them to post announcements, create assignments, mark attendance records, assign the grade for every student and instructors also can send email to students and keep track of the sent email for future references. This improves the communication and education between the instructors and the students.

Additionally we will look into the security measures that have been applied in the system. RBAC, role based access control is necessary in this system for students and instructors and the separate URL address for the admin site. The role based access control prevents unauthorized access to resources based on the roles of individual users within the system. This method helps to maintain the integrity and confidentiality of student information. CSRF, cross-site request forgery is another measure which is implemented in the system that ensures that all form submissions are secure and authenticated based on the generated CSRF token.

Moreover, password hashing. In this Secured Student Information Management System to save the entered password, the plaintext is encrypted by using the hashing algorithm PBKDF2 (Password-Based Key Derivation Function2) with a SHA256 and salt which makes the password difficult to be guessed by someone else apart from the user themselves. Hashing passwords is a one-way function, which means it cannot be reversed to obtain the original password. Lastly, the time-out session is also another security measure, where it prevents the authorized session from being hijacked by the unauthorized users which is called a session hijacking attack.

The Secured Student Information Management System might stand as a robust, secure and user-friendly platform that significantly efficiently improves the management of the student information. It supports the academic process to be more straightforward and easy to navigate for the users. By integrating strong security measures, the system ensures data CIA, confidentiality, integrity and availability for those who have the valid access to use the system.

6.3 Limitation and Recommendation for Future Works

The current implementation of the Secured Student Information Management System may face scalability issues. This is because this system runs in the SQLite3 database which is lightweight and serverless. This database particularly works for development, testing and for the small to medium size systems. In Django during the project development the SQLite3 is used as the default database. The database is designed to be fast, reliable, and fully self-contained. Since the student information management system deals with a lot of information and data the SQLite3 is not recommended because this database is designed for large number of dataset at the same time if the database handles up to some size, the efficiency of the system performance might degrade everytime an information is stored into and the system does not work seamlessly and effectively.

Therefore, given the scalability issues of SQLite3 in the Secured Student Information Management System, with the dataset might grow in the future, it is recommended to use the type of database that can handle large amount of informations such as the robust relational database management system (RDBMS). This is suitable when the system moves to the production state. Updating the database settings in the Django project to use (rdbms) such as PostgreSQL or MariaDB and performing the migrations to transfer the existing data to the new database. Apart from the suitable database, it is recommended to monitor and maintain the database regularly to decrease any loopholes that can be useful for attackers to make use of it. Implementing a backup strategy helps to prevent data loss in future and use the backup to solve the data corruption which can happen due to software bugs, hardware failures or any other issues

which can be unintentional and unexpected. These recommendations for future works not only enhance the reliability but also the security of the Secured Student Information Management System.

REFERENCES

PowerSchool. (2024, January 23). *PowerSchool K-12 Software & Cloud-Based Solutions*.

<https://www.powerschool.com/>

Blackbaud. (2024, January 5). *Student Information System (SIS) Software | BlackBaud*.

<https://www.blackbaud.com/products/student-information-system>

Skyward | A better SIS and ERP experience. (n.d.).

<https://www.skyward.com/>

Aeries SIS - Aeries Software. (n.d.).

<https://www.aeries.com/solutions/aeries-sis/>

Rediker Software Advantages and Products. (n.d.). [Video].

<https://www.rediker.com/>

Teachmint. (n.d.). *TeachMint : School LMS & ERP | Integrated School Platform (ISP)*.

<https://www.teachmint.com/en-my>

QuickSchools. (n.d.). *School management software - QuickSchools*.

<https://www.quickschools.com/>

Wisenet - a division of Adapt IT. (n.d.). *Student Management system and training applications from Wisenet.* Wisenet - a Division of Adapt IT.

<https://www.wisenet.co/>

Moffitt, P. (2023, September 21). *GradeLink Student Information System: Simplify and Succeed.*

Gradelink: Student Information System.

<https://www.gradelink.com/>

Orah: Improve Student Life | Attendance, Wellbeing & Communications. (n.d.).

<https://www.orah.com/>

Sad. (2013, March 3). *Advantages and Disadvantages of SDLC*. Eternal Sunshine of the IS Mind.

<https://eternalsunshineoftheismind.wordpress.com/2013/03/03/advantages-and-disadvantages-of-sdlc/>

20 Best Student Information Systems in 2024. (2023, October 10). *Research.com*.

https://research.com/software/best-student-information-systems-in-2022-key-features-of-the-top-student-information-systems#google_vignette

Team, A. (2024, January 19). *The most loved and hated programming languages according to developers* | *Appetiser*. Appetiser: Web and Mobile App Development Melbourne, Australia.

<https://appetiser.com.au/blog/the-most-loved-and-hated-programming-languages-according-to-developers/>

Sandamal, H. (2021, December 16). Software Development Life Cycle methods - Hansa Sandamal - Medium. *Medium*.

<https://hansasandamal.medium.com/software-development-life-cycle-methods-7ccd25d6e198>

SQLite. (2024, May, 23). What is SQLite

<https://sqlite.org/>

APPENDICES

Appendix A: Meeting Logs



Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log

MEETING DATE: 4/4/2024	MEETING NO.: 1
PROJECT ID: T831084	
PROJECT TITLE : Secured Student Information Management System	
TRIMESTER : Trimester 2 2023/2024	SUPERVISOR : Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashinii A/P Tharmalingam	CO-SUPERVISOR :

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> • Find and learn how does the Python language works • Get to know how the Django framework is used in the Python
2. WORK TO BE DONE
<ul style="list-style-type: none"> • Starting to create the system
3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> • Setting up and installing the Django into the device
4. COMMENTS



thema

Supervisor's Signature &
Stamp

Co-Supervisor's Signature &
Stamp (if any)

Student's Signature

NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
3. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
4. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log

MEETING DATE: 18/4/2024	MEETING NO.: 2
PROJECT ID: T831084	
PROJECT TITLE : Secured Student Information Management System	
TRIMESTER : Trimester 2 2023/2024	SUPERVISOR : Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashini A/P Tharmalingam	CO- SUPERVISOR :

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> Starting with creating users and saving it to the database Creating registration and login page
2. WORK TO BE DONE
<ul style="list-style-type: none"> Adding RBAC to the login page
3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> Creating database accurately and precisely
4. COMMENTS


Faculty of Information
Science and Technology

thema

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

5. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
6. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
7. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
8. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log

MEETING DATE: 2/5/2024	MEETING NO.: 3
PROJECT ID: T831084	
PROJECT TITLE: Secured Student Information Management System	
TRIMESTER: Trimester 2 2023/2024	SUPERVISOR: Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashinii A/P Tharmalingam	CO- SUPERVISOR:

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> • Implementing the role-based access control into the system
2. WORK TO BE DONE
<ul style="list-style-type: none"> • Creating the features each and every user can access to • Making the features to function

3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> • Accurately RBAC making to work based on the university email. • Recongnising the users based on the emails

4. COMMENTS



Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

thema

NOTES:

9. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
10. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
11. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
12. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)

Final Year Project Meeting Log

MEETING DATE: 23/5/2024	MEETING NO.: 4
PROJECT ID: T831084	
PROJECT TITLE: Secured Student Information Management System	
TRIMESTER: Trimester 2 2023/2024	SUPERVISOR: Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashini A/P Tharmalingam	CO-SUPERVISOR:

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> • Implementing the role-based access control into the system
2. WORK TO BE DONE
<ul style="list-style-type: none"> • Implementing the SSL certificate to the system • Creating the front-end languages
3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> • Integrating the front-end with Python
4. COMMENTS



Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

thema

NOTES:

13. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
 14. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
 15. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
 16. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.
- *: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)
- **: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log

MEETING DATE: 30/5/2024	MEETING NO.: 5
PROJECT ID: T831084	
PROJECT TITLE: Secured Student Information Management System	
TRIMESTER: Trimester 2 2023/2024	SUPERVISOR: Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashini A/P Tharmalingam	CO- SUPERVISOR:

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> • Done Implementing SSL
2. WORK TO BE DONE
<ul style="list-style-type: none"> • Creating the front-end languages
3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> • Integrating the front-end with Python
4. COMMENTS



thema

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

17. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
18. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
19. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
20. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)

Final Year Project Meeting Log

MEETING DATE: 6/6/2024	MEETING NO.: 6
PROJECT ID: T831084	
PROJECT TITLE: Secured Student Information Management System	
TRIMESTER: Trimester 2 2023/2024	SUPERVISOR: Ms. Pa Pa Min
STUDENT ID & Name: 1191101916 & Hemanilashinii A/P Tharmalingam	CO- SUPERVISOR:

All to be filled in by student

1. WORK DONE
<ul style="list-style-type: none"> • Done integrating front-end with Python
2. WORK TO BE DONE
<ul style="list-style-type: none"> • Implementing the SSL certificate to the system • Creating the front-end languages
3. PROBLEMS ENCOUNTERED
<ul style="list-style-type: none"> • Making the system from HTTP to HTTPS
4. COMMENTS


 Supervisor's Signature &
 Stamp Co-Supervisor's Signature
 & Stamp (if any) Student's Signature
thema

NOTES:

21. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
22. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
23. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
24. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)