**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**FACULTY OF PHYSICAL AND COMPUTATIONAL SCIENCE**

**DEPARTMENT OF COMPUTER SCIENCE**



**PROJECT TOPIC:**

**DESIGN AND IMPLEMENTATION OF ATTENDANCE TRACKING SOFTWARE FOR UNIVERSITY CAMPUS STUDY FACILITIES USING QR-CODE TECHNOLOGY**

**SUBMITTED BY:**

**ASAMANING REDOLF (9398419)**

**SUPERVISED BY:**

**Mr. J.K PANFORD**

**AUGUST 2023**

# DEDICATION

I would like to dedicate this work first to the lord for making it possible for me to reach this far in our academic laurels. Again, I dedicate this final year project to all those who have supported and inspired me throughout my journey as a computer science student. To my family, thank you for your unwavering love, encouragement, and belief in my abilities. Your constant support has been a source of strength for me, and I am deeply grateful for your presence in my life. To my friends and classmates, thank you for the late-night study sessions, the brainstorming sessions, and the camaraderie we shared. Your friendship and collaboration have made this project both enjoyable and rewarding. To my professors and mentors, thank you for your guidance, knowledge, and expertise. Your invaluable insights and feedback, have shaped my understanding of computer science and have played a crucial role in the successful completion of this project. This project would not have been possible without the contributions of the open-source community and all the developers who selflessly share their work, thank you for providing a wealth of resources and tools that have been instrumental in my project's development. Your contributions to the field of computer science have inspired me to push boundaries and strive for excellence.

Lastly, I dedicate this project to myself. It is a testament to my hard work, determination, and passion for computer science. I am proud of the skills I have acquired, the challenges I have overcome, and the growth I have experienced throughout my academic journey.

May this final year project serve as a stepping stone towards a fulfilling career in the field of computer science and as a reminder of the incredible support system, including the guidance of my supervisor, Mr. J.K Panford that has surrounded me.

# DECLARATION BY STUDENT

I declare, without any reservation, that I undertook this project known as “**ATTENDANCE TRACKING SOFTWARE FOR UNIVERSITY CAMPUS STUDY FACILITIES USING QR-CODE TECHNOLOGY”** on the **KNUST** campus, herein submitted under supervision.

Signed: …………………………. Date: ………………………………….

ASAMANING REDOLF

(9398419)

# DECLARATION BY SUPERVISOR

I declare that I have personally supervised these students in undertaking the study report herein and I confirm that this student has my permission to present it for assessment.

Signed: …………………………. Date: ………………………………….

Mr. J.K PANFORD

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project.

First and foremost, I would like to thank my supervisor, Mr. J.K Panford for his guidance, mentorship, and continuous support throughout the project. His expertise and valuable insights have been instrumental in shaping the direction and scope of this work. I would also like to extend my appreciation to the faculty members of the Computer Science department for providing a stimulating academic environment and equipping me with the necessary knowledge and skills to undertake this project. My deepest gratitude goes to my friends for their unwavering encouragement and understanding during the course of this project. Their belief in my abilities has been a constant source of motivation. I would like to acknowledge the contributions of my classmates and colleagues for their collaborative efforts, insightful discussions, and assistance during various stages of the project. Their feedback and suggestions have significantly improved the quality of my work.

Furthermore, I would like to express my gratitude to the open-source community for providing an abundance of resources and tools that have been essential in the development and implementation of this project.

Lastly, I would like to acknowledge the support and assistance provided by KNUST as a whole in facilitating the necessary resources, equipment, and infrastructure required for the successful completion of this project.

This project would not have been possible without the collective efforts and support of all those mentioned above. I am deeply thankful for their contributions, and I am confident that their guidance and encouragement will continue to shape my future endeavors in the field of computer science.

# ABSTRACT

Attendance tracking is an essential part of the education process, as it helps to ensure that students are regularly visiting their learning facilities and engaging in the material. However, traditional methods of attendance tracking, such as paper sign-in sheets or calling out names, can be time-consuming and prone to error.

To address these issues, I propose the development of an attendance tracking system software to be used at the various study facilities. This software will allow administrators of the facility to easily and accurately track student attendance in real-time, using a computer.

TABLE OF CONTENTS

[DEDICATION i](#_Toc143778959)

[DECLARATION BY STUDENT ii](#_Toc143778960)

[DECLARATION BY SUPERVISOR ii](#_Toc143778961)

[ABSTRACT iv](#_Toc143778962)

[LIST OF FIGURES vii](#_Toc143778963)

[LIST OF TABLES viii](#_Toc143778964)

[CHAPTER ONE 1](#_Toc143778965)

[INTRODUCTION 1](#_Toc143778966)

[1.1 BACKGROUND OF THE STUDY 1](#_Toc143778967)

[1.2 PROBLEM OF THE RESEARCH PROJECT 2](#_Toc143778968)

[1.3 PROJECT AIM 2](#_Toc143778969)

[1.4 PROJECT OBJECTIVES 3](#_Toc143778970)

[1.5 PROJECT RESEARCH QUESTIONS 4](#_Toc143778971)

[1.6 JUSTIFICATION OF THE RESEARCH PROJECT 4](#_Toc143778972)

[1.7 SIGNIFICANCE OF THE RESEARCH PROJECT 5](#_Toc143778973)

[1.8 SCOPE OF THE RESEARCH PROJECT 6](#_Toc143778974)

[1.9 OVERVIEW OF METHODOLOGY 7](#_Toc143778975)

[2.0 ORGANIZATION OF THE RESEARCH PROJECT 8](#_Toc143778976)

[CHAPTER TWO 9](#_Toc143778977)

[REVIEW OF SIMILAR SYSTEMS 9](#_Toc143778978)

[2.1 PROCESSES OF THE EXISTING SYSTEM 9](#_Toc143778979)

[2.2 PROPOSED SYSTEM 10](#_Toc143778980)

[2.3 ARCHITECTURE OF THE PROPOSED SYSTEM 11](#_Toc143778981)

[2.3.1 PRESENTATION LAYER 11](#_Toc143778982)

[2.3.2 BUSINESS LOGIC LAYER 11](#_Toc143778983)

[2.3.3 DATA ACCESS LAYER 12](#_Toc143778984)

[2.3.4 CACHE LAYER 12](#_Toc143778985)

[2.4 COMPONENTS DESIGN AND COMPONENTS DESCRIPTIONS 14](#_Toc143778986)

[2.4.1 KEY ASPECTS OF COMPONENT DESIGN IN SOFTWARE 14](#_Toc143778987)

[2.5 PROPOSED SOFTWARE FEATURES 21](#_Toc143778988)

[2.6 DEVELOPMENT TOOLS AND ENVIRONMENT 22](#_Toc143778989)

[3.4 FUNCTIONAL REQUIREMENTS 25](#_Toc143778990)

[3.4.1 FUNCTIONAL REQUIREMENTS OF THE SYSTEM 26](#_Toc143778991)

[3.5 NON-FUNCTIONAL REQUIREMENTS 27](#_Toc143778992)

[3.5.1 SOME NON-FUNCTIONAL REQUIREMENTS 27](#_Toc143778993)

[3.6 UML DIAGRAMS 29](#_Toc143778994)

[3.7 SECURITY CONCEPTS 32](#_Toc143778995)

[3.8 PROJECT METHODOLOGY 33](#_Toc143778996)

[3.8.1 SOME BENEFITS OF INCREMENTAL DESIGN APPROACH 35](#_Toc143778997)

[3.9 VARIOUS SOFTWARE PROCESSES 36](#_Toc143778998)

[3.10 CHOSEN MODEL AND JUSTIFICATION 37](#_Toc143778999)

[CHAPTER 4 39](#_Toc143779000)

[IMPLEMENTATION AND RESULTS 39](#_Toc143779001)

[4.0 OVERVIEW 39](#_Toc143779002)

[4.1 MAPPING LOGICAL DESIGN ONTO PHYSICAL PLATFORM 40](#_Toc143779003)

[4.2 CONSTRUCTION 41](#_Toc143779004)

[4.3 TESTING 49](#_Toc143779005)

[4.3.1 TOOLS USED IN TESTING 49](#_Toc143779006)

[4.3.2 REASONS FOR TESTING 50](#_Toc143779007)

[4.4 RESULTS 52](#_Toc143779008)

[4.4.1 RESULTS FOR TESTING 53](#_Toc143779009)

[CHAPTER 5 62](#_Toc143779010)

[FINDINGS AND CONCLUSION 62](#_Toc143779011)

[5.0 OVERVIEW 62](#_Toc143779012)

[5.1 FINDINGS 62](#_Toc143779013)

[5.2 CONCLUSION 63](#_Toc143779014)

[5.3 LIMITATIONS OF THE SYSTEM 64](#_Toc143779015)

[5.4 LESSON LEARNT 65](#_Toc143779016)

[5.5 RECOMMENDATIONS FOR FUTURE WORK 66](#_Toc143779017)

[5.6 RECOMMENDATIONS FOR COMMERCIALIZATION 66](#_Toc143779018)

[**REFERENCES** 68](#_Toc143779019)

# LIST OF FIGURES

[Figure 1 Layered architecture 13](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440791)

[Figure 2 System architecture 13](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440792)

[Figure 3 Component design 17](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440793)

[Figure 4 Class diagram 29](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440794)

[Figure 5 Database schema 30](#_Toc143440795)

[Figure 6 User registration sequence diagram 31](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440796)

[Figure 7 Student sequence diagram 31](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440797)

[Figure 8 User login sequence diagram 32](#_Toc143440798)

[Figure 9 Agile process model 36](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440799)

[Figure 10 Flowchart for the system 40](#_Toc143440800)

[Figure 11 Fronted designing 41](#_Toc143440801)

[Figure 12 Backend coding 41](#_Toc143440802)

[Figure 13 Launcher screen 42](#_Toc143440803)

[Figure 14 Login screen 42](#_Toc143440804)

[Figure 15 Dashboard 43](#_Toc143440805)

[Figure 16 Local report generation 43](#_Toc143440806)

[Figure 17 Central records report generation 44](#_Toc143440807)

[Figure 18 Retrieve records 44](#_Toc143440808)

[Figure 19 Push local records to server 45](#_Toc143440809)

[Figure 20 Account registration 45](#_Toc143440810)

[Figure 21 Current Logged in administrator 46](#_Toc143440811)

[Figure 22 Integrated surveillance configuration 46](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440812)

[Figure 23 Web server response frame 47](#_Toc143440813)

[Figure 24 Web request URL configuration 47](#_Toc143440814)

[Figure 26 Mailing of reports frame 48](#_Toc143440815)

[Figure 25 Integrated surveillance frame 48](file:///C:\Users\BTC%20OMEN\Documents\project\asamaning_redolf_9398419.docx#_Toc143440816)

# LIST OF TABLES

[Table 1 PyQt5 application module - Unit Testing 53](#_Toc143176864)

[Table 2 Student attendance taking - Unit Testing 54](#_Toc143176865)

[Table 3 Code generation, mailing, printing and searching - Unit Testing 54](#_Toc143176866)

[Table 4 Record retrieval and exporting - Unit Testing 56](#_Toc143176867)

[Table 5 Administrator account details updating - Unit Testing 59](#_Toc143176868)

[Table 6 Camera feed enhancement - Unit Testing 60](#_Toc143176869)

[Table 7 System configuration - Unit Testing 61](#_Toc143176870)

[Table 8 Report mailing - Unit Testing 61](#_Toc143176871)

# CHAPTER ONE

# INTRODUCTION

## 1.1 BACKGROUND OF THE STUDY

The university is a hub of learning where students and faculty come together to exchange ideas and knowledge. One of the critical resources that universities provide is study facilities, such as libraries, computer labs, and study halls. These facilities are essential for students to succeed in their academic pursuits, but their effective management is critical. An attendance tracking system can help in this regard by providing a comprehensive and accurate picture of facility usage, enabling university administrators to make data-driven decisions about resources and how to manage the facility. In this project background, we will explore the need for an innovative attendance tracking system in can be utilized in university study facilities and how the attendance data generated can help determine the usage of the facility.

Traditionally, they have relied on manual methods of tracking attendance, such as sign-in sheets. While these methods can provide some basic information about facility usage, they are often inaccurate, time-consuming, and inefficient. Manual tracking methods can also make it challenging to identify patterns in facility usage and make informed decisions about resource allocation. As such, there is a need for a more efficient and accurate system for tracking attendance in facilities.

An innovative attendance tracking system can provide a comprehensive and accurate picture of facility usage, allowing university administrators to make data-driven decisions about the facility. Such a system can help identify peak usage times and patterns in facility usage, enabling administrators to make more effective and informed decisions using data.

## 1.2 PROBLEM OF THE RESEARCH PROJECT

The problem statement for this project is to design and implement an innovative attendance tracking system for university campus study facilities that can provide accurate and comprehensive data on facility usage. The attendance tracking system will enable administrators to make data-driven decisions about resource allocation, ensuring that resources are allocated efficiently and effectively to support the needs of students and faculty.

In many cases, facility administrators have limited information about the usage of study areas, books, and computer labs. This can result in resources being underutilized or overburdened, leading to inefficient allocation and reduced access to resources for users. Furthermore, without accurate data on facility usage, administrators cannot make informed decisions about resource allocation, which can lead to inefficient and ineffective use of resources.

## 1.3 PROJECT AIM

The project aims to provide a reliable, efficient, and convenient way for students to sign in and out of study facilities, thereby enabling the university to monitor facility usage and improve access for all students. By implementing the system, students and faculty will have access to accurate information on facility usage, which will improve their overall management of the facility.

## 1.4 PROJECT OBJECTIVES

The objective of the project is to provide an efficient and accurate way of tracking the attendance of students in study facilities to monitor their usage and enhance their access to these resources. The system will provide a more accurate and efficient way of tracking facility usage, allowing facility administrators to make data-driven decisions about how to improve on the existing one.

The specific objectives of this project include:

* Design a user-friendly attendance tracking system that can be easily used to take attendance.
* To provide real-time updates on attendance data, making it easier for administrators to monitor and manage attendance records.
* To have a range of features and functionality to enhance the management and analysis of attendance data, including the ability to generate customized reports and charts to help visualize attendance trends and patterns.
* Develop a set of best practices and guidelines for implementing the attendance tracking system in other study facilities.

The pilot facility for this project will be the learning commons and some study facilities, which have a high volume of visitors and a diverse range of resources. The pilot study will run for two months, during which attendance data will be collected and analyzed. The project will be led by a team of IT professionals and library administrators, with support from university management.

## 1.5 PROJECT RESEARCH QUESTIONS

Having stated the purpose of the study, the following research questions were stated to guide the researcher.

* How can the data generated by the system be used to make data-driven decisions to improve efficiency and accuracy in academic study facilities?
* How does the implementation of a QR code-based attendance tracking system impact attendance tracking processes in student study facilities?
* What is the effect of QR code-based attendance tracking systems on student engagement and accountability in student study facilities?
* How does the adoption of QR code-based attendance tracking systems affect the workload of staff members in student study facilities?

## 1.6 JUSTIFICATION OF THE RESEARCH PROJECT

The justification for a research project on QR code-based attendance tracking systems in student study facilities is rooted in the potential benefits this technology can provide to both students and staff members.

Firstly, the implementation of a QR code-based attendance system can help to improve the accuracy and efficiency of attendance-tracking, reducing the potential for errors and fraud. This can save time for both students and staff members, allowing them to focus on other important tasks.

Secondly, a QR code-based attendance system can provide valuable data for decision-making. Real-time data can be used to track attendance trends, identify patterns, and make informed decisions about facility usage, student behavior, and areas for improvement.

Thirdly, this technology can help to increase student engagement and accountability. By providing students with a quick and easy way to track their attendance, they are more likely to attend regularly and take ownership of their learning.

Additionally, the implementation of a QR code-based attendance system can demonstrate the institution's commitment to innovation and modernization, helping to attract and retain students who value technology-driven solutions.

## 1.7 SIGNIFICANCE OF THE RESEARCH PROJECT

The implementation of a QR code-based attendance tracking system in student study facilities has significant potential benefits for both students and staff members. By using this technology, study facilities can improve their attendance-tracking processes while reducing the potential for errors and fraud.

One of the main advantages of a QR code-based attendance system in study facilities is that it can improve the accuracy and efficiency of attendance tracking. This system allows students to quickly scan their QR codes using their smartphones or other devices, eliminating the need for paper-based systems or manual data entry. This can save time for both students and staff members, allowing them to focus on other important tasks.

Moreover, the system provides real-time data that can be analyzed and used to track attendance trends, identify patterns, and make informed decisions. This can help staff members to better understand how students are using the facility, and identify areas for improvement or additional support.

Additionally, a QR code-based attendance system can help to increase student engagement and accountability. By providing students with a quick and easy way to track their attendance, they are more likely to attend regularly and take ownership of their learning.

## 1.8 SCOPE OF THE RESEARCH PROJECT

The project will focus on designing and implementing an attendance tracking system for university libraries and study facilities. The attendance tracking system will be designed to be user-friendly and easily integrated into existing university systems. The system will use the QR codes data collection method, among the various data collection techniques such as barcodes, RFID, or biometrics, to collect attendance data. The QR code is a great emerging technology that can be used to solve many problems. The chosen technology is fast in terms of speed, less costly, and easy to implement. The attendance data collected will be analyzed to identify patterns in facility usage and make informed decisions to improve upon existing ones and the creation of new academic study areas for students.

* Analysis of existing attendance tracking processes: The project will involve a comprehensive analysis of existing attendance tracking processes in student study facilities, identifying challenges and areas for improvement.
* ***Development of QR code-based attendance tracking system***: The project will focus on the development and implementation of a QR code-based attendance tracking system that addresses the identified challenges and provides an efficient and user-friendly solution.
* ***Testing and evaluation of the system***: The project will involve testing and evaluation of the QR code-based attendance tracking system, ensuring that it meets the needs of student study facilities and provides a reliable and accurate attendance tracking solution.
* ***Impact evaluation***: The project will assess the impact of the QR code-based attendance tracking system on attendance tracking processes, student engagement, and staff workload.

Evaluation of the attendance tracking system will be based on several key metrics. These metrics will include the accuracy of attendance data, the ease of use of the attendance tracking system, the effectiveness of the attendance tracking system in identifying patterns in facility usage.

## 1.9 OVERVIEW OF METHODOLOGY

The innovative attendance tracking system for the study area can be developed using an agile methodology. This approach emphasizes collaboration, flexibility, and iterative development. The project team works closely with stakeholders to understand their needs and requirements and to deliver a system that meets their needs. The project is divided into multiple phases, including planning, requirements gathering, design, development, testing, deployment, and maintenance. The project team uses agile practices such as daily stand-up meetings, sprint planning, and retrospectives to ensure continuous iteration and collaboration with stakeholders.

The agile approach allows for flexibility and responsiveness to changing requirements and priorities. The project team can adjust the scope, timeline, and budget as needed based on feedback from stakeholders and the evolving needs of the facility administrators. This approach also allows for early and frequent delivery of working software, which can help ensure that the system meets the needs of stakeholders and facilitates usage effectively and efficient.

## 2.0 ORGANIZATION OF THE RESEARCH PROJECT

Six components make up this thesis. The first is an introduction that outlines the context of the research issue, the need for a study, the research question, the purpose, any limitations, and the format of this thesis. Since the entirety of the conduct of this study was based on a systematic review of prior studies, the literature review section (section two) was left to offer concepts and definitions that strengthen the general understanding of the important topics covered in depth in the result and discussion section (section four and five). The methods part follows as the third section. This section provided a comprehensive explanation of the stages involved in conducting a systematic review of the literature. The final section of this thesis, titled "Conclusion," presents conclusions derived from the result and discussion sections along with potential suggestions for future research topics.

# CHAPTER TWO

# REVIEW OF SIMILAR SYSTEMS

## 2.1 PROCESSES OF THE EXISTING SYSTEM

The proposed attendance monitoring system software aims to address the limitations of these existing systems by providing a comprehensive and reliable solution for tracking and monitoring student attendance. It will have a range of features and functionality to enhance the management and analysis of attendance data, including the ability to generate customized reports and charts to help visualize attendance trends and patterns. It will also have the ability to integrate with other systems and software and will have robust security measures in place to protect against data breaches and other security threats. There have been several previous attendance monitoring systems developed for educational institutions, including:

* ***Manual attendance tracking***: This involves using paper-based or spreadsheet-based systems to track attendance manually. This can be time-consuming and prone to errors and does not provide real-time updates on attendance data.
* ***Attendance tracking software***: This type of software automates the attendance tracking process and provides real-time updates on attendance data. However, some systems may have limited reporting capabilities and may not be able to generate customized reports or charts to help visualize attendance trends and patterns.
* ***Biometric attendance systems***: These systems use biometric technologies, such as fingerprints or facial recognition, to track attendance. While these systems are accurate, they may be more expensive to implement and may not be suitable for all facilities.
* ***Mobile attendance tracking apps***: These apps allow facility administrators to record attendance using their mobile devices. However, they may not have the same level of functionality and reporting capabilities as more comprehensive attendance monitoring systems.

## 2.2 PROPOSED SYSTEM

The proposed attendance monitoring system software is designed to provide the facility with a comprehensive solution for tracking and monitoring the attendance of their students. It aims to improve the accuracy and efficiency of attendance tracking by automating the process and providing real-time updates on attendance data.

The software will allow administrators to record attendance using a computer and will provide real-time updates on attendance data. This will make it easier for administrators to monitor and manage attendance records and will help to ensure that attendance data is accurate and up-to-date. The software will also have the ability to generate attendance reports, which can be used by teachers and administrators to identify trends and patterns in attendance.

In addition to automating the attendance tracking process, the proposed system will also have a range of features and functionality to enhance the management and analysis of attendance data. This may include the ability to track attendance by program, and student, and to generate customized reports and charts to help visualize attendance trends and patterns.

## 2.3 ARCHITECTURE OF THE PROPOSED SYSTEM

Layered architecture is a software design pattern that organizes a system into distinct layers, each responsible for specific functionalities. It promotes modularity, scalability, and ease of maintenance. The addition of a cache layer enhances performance by storing frequently accessed data in a faster and more easily accessible location.

In a typical layered architecture, the system is divided into multiple layers, with each layer having a specific role and set of responsibilities. The common layers include the presentation layer, business logic layer, data access layer, and sometimes caching layer. Let's explore each layer and then discuss the inclusion of a cache layer.

### 2.3.1 PRESENTATION LAYER

The presentation layer is responsible for handling user interactions and presenting information to the users. It encompasses user interfaces, such as web pages, mobile apps, or desktop applications, and focuses on rendering the user interface and handling user input. This layer communicates with the underlying layers to retrieve or update data as needed.

### 2.3.2 BUSINESS LOGIC LAYER

The business logic layer contains the core functionality and rules of the application. It encapsulates the business processes and workflows, ensuring that the system behaves as intended. This layer implements the business rules, validations, calculations, and transformations required for the application's specific domain. It acts as an intermediary between the presentation layer and the data access layer.

### 2.3.3 DATA ACCESS LAYER

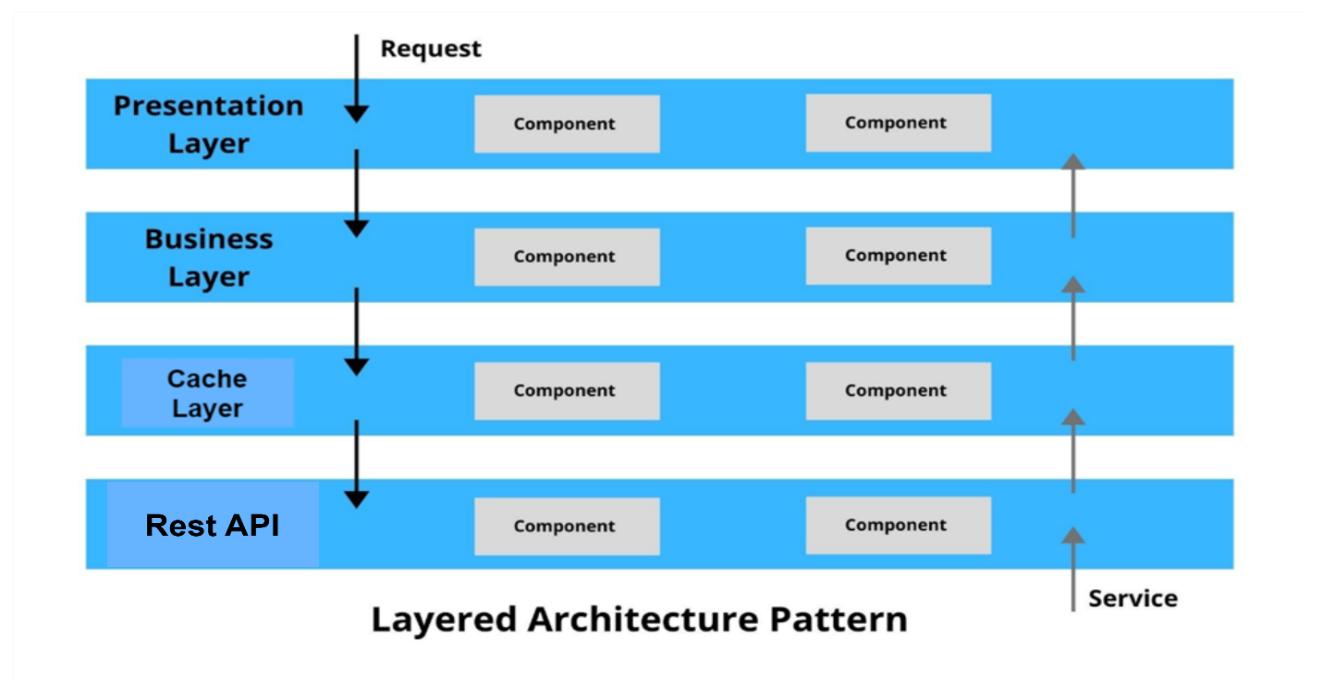
The data access layer handles the retrieval and persistence of data. It provides an abstraction over the data storage, such as a database, file system, or external service. This layer handles the creation, modification, deletion, and retrieval of data entities. It encapsulates the data access logic and shields the other layers from the underlying data storage details.

### 2.3.4 CACHE LAYER

The cache layer is an additional layer that sits between the data access layer and the underlying data storage. Its primary purpose is to improve the system's performance by storing frequently accessed data in a cache, which is a faster and easily accessible form of memory.

When data is requested from the data access layer, the cache layer checks if the requested data is already available in the cache. If it is, the data is retrieved from the cache directly, eliminating the need to access the underlying data storage. This significantly reduces the response time and improves the overall system performance. If the requested data is not found in the cache, the cache layer retrieves it from the underlying data storage and stores a copy in the cache for future use. The cache uses various strategies, such as expiration policies or eviction algorithms, to manage the data and ensure that it stays relevant and up to date.

By incorporating a cache layer into the architecture, you can reduce the load on the underlying data storage, minimize network latency, and improve the overall responsiveness of the system. Caching is particularly effective for read-heavy applications where data doesn't change frequently or can be temporarily stored.



##### Figure 1 Layered architecture

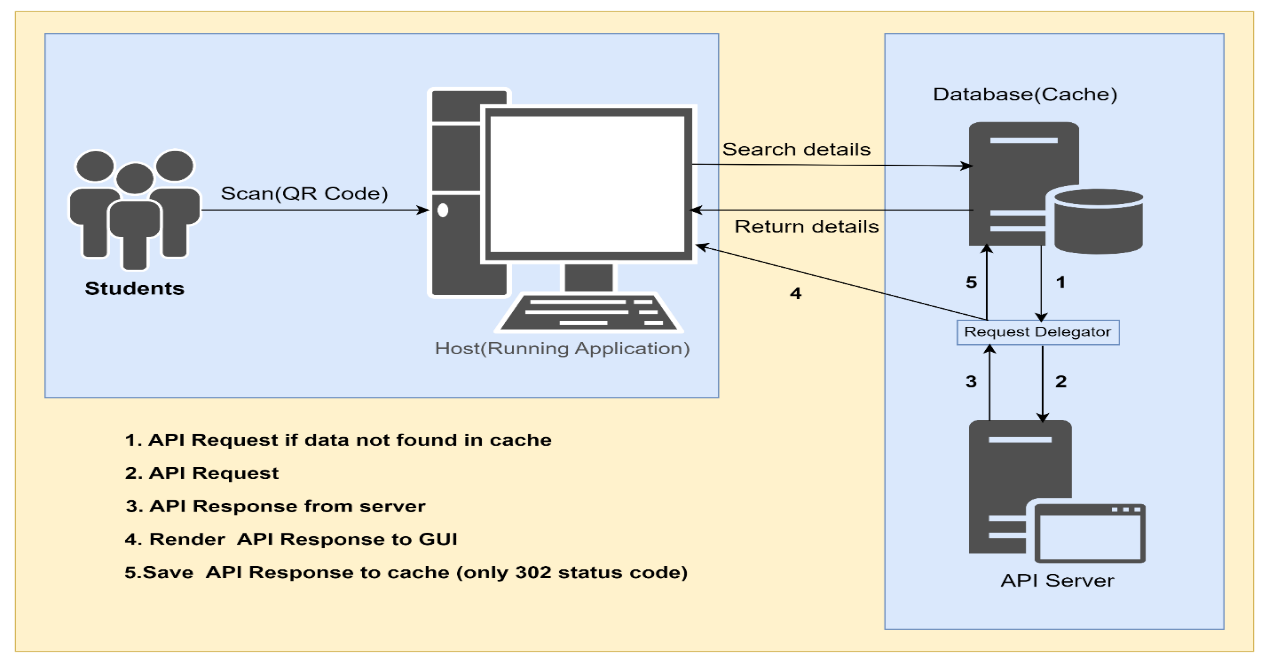


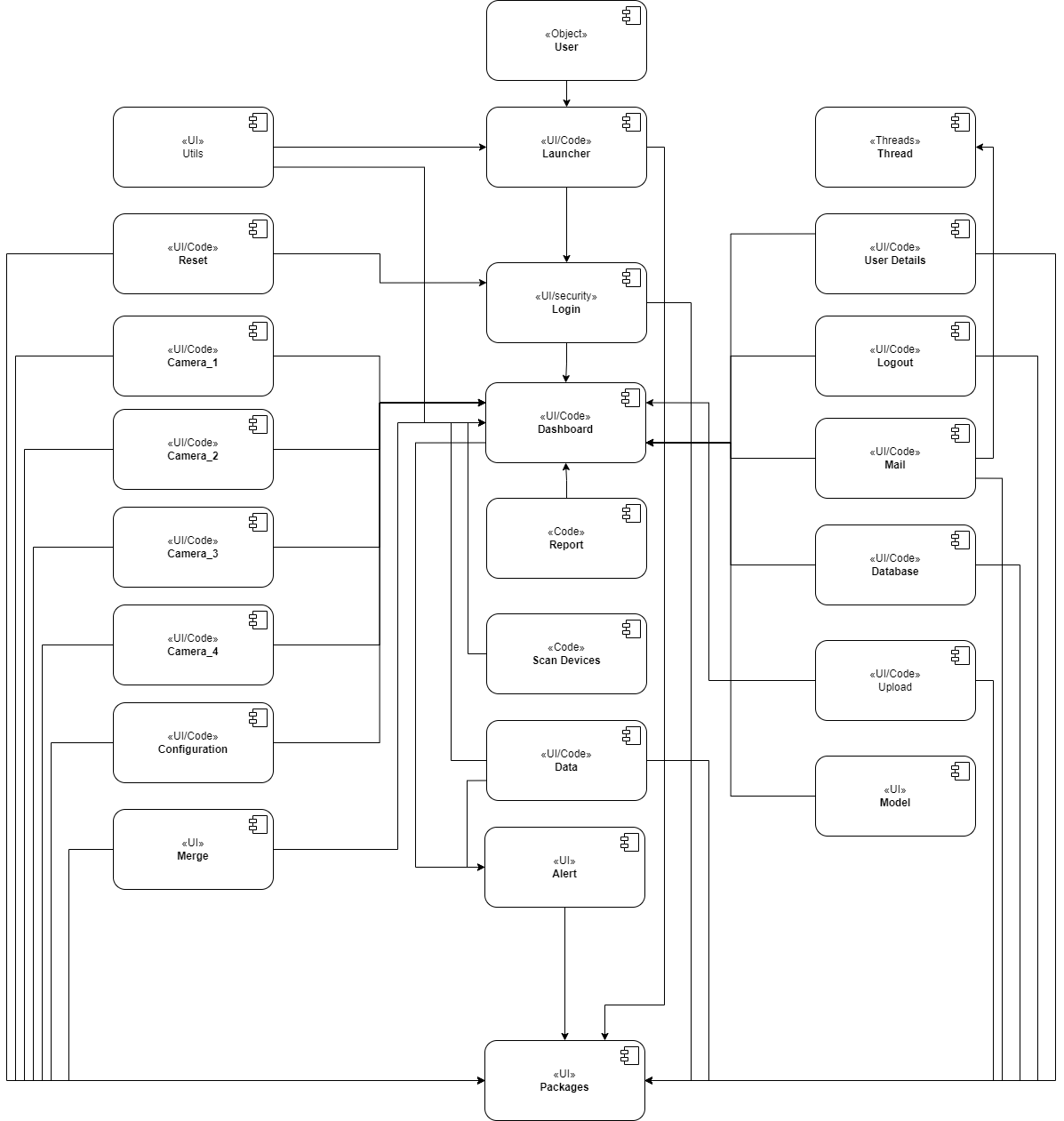
Figure 2 System architecture

## 2.4 COMPONENTS DESIGN AND COMPONENTS DESCRIPTIONS

The concept of component design in software refers to the process of breaking down a complex system into modular and reusable components. It involves identifying the different functional units of the system and designing each component to encapsulate a specific set of functionalities, data, and behavior. Component design focuses on creating cohesive and loosely coupled units that can be easily understood, developed, tested, and maintained.

### 2.4.1 KEY ASPECTS OF COMPONENT DESIGN IN SOFTWARE

* ***Modularity***: Components are self-contained units that encapsulate related functionality. They can be developed, tested, and maintained independently of each other. Modularity helps in managing complexity, promoting code reusability, and enhancing the system's overall organization.
* ***Encapsulation:*** Components encapsulate their internal workings, data, and behavior behind well-defined interfaces. This allows other components to interact with them through a clearly defined set of methods, properties, and events. Encapsulation provides abstraction and reduces the complexity of working with a component.
* ***Reusability*:**Components are designed to be reusable across different projects, systems, or even within the same system. Reusable components save development time and effort by leveraging existing functionality and reducing redundancy. They can be easily integrated into different contexts and scenarios, enhancing productivity.
* ***Cohesion***: Components should have high cohesion, meaning that they should encapsulate functionality that is closely related and aligned with a specific purpose. Components with high cohesion are focused, maintainable, and easier to understand. They should have a clear and well-defined responsibility.
* ***Loose Coupling***: Components should be loosely coupled, meaning they should have minimal dependencies on other components. Loose coupling promotes flexibility, and independence, and reduces the impact of changes. Components should communicate through well-defined interfaces, allowing them to evolve independently.
* ***Interface Design***: Components should define clear and well-documented interfaces that specify how other components or systems can interact with them. Interfaces provide a contract for communication, ensuring that components work together effectively and facilitating seamless integration.
* ***Testing and Validation***: Each component should be thoroughly tested and validated to ensure its correctness, reliability, and compliance with the specified requirements. Testing can include unit tests, integration tests, and component-level validations to verify the component's functionality in isolation and in collaboration with other components.
* ***Maintainability***: Component design aims to enhance the maintainability of the system by promoting modular, reusable, and well-structured code. Components can be updated, modified, or replaced without affecting the entire system. This makes it easier to troubleshoot issues, apply bug fixes, and introduce new features.



##### Figure 3 Component design

* **Launcher:** The first screen that appears when the user starts the application.
* **Login:** Authentication is a UI security that makes sure that users who access the platform are who they claim to be. This is shown after the launcher is done loading.
* **User Object:** The user object is people who interact with the system. Also known as final users or End users, often human consumers of a software product. In contrast to people who support or maintain the product, such as system administrators, database administrators, and computer technicians, the end user is not one of these users.
* **Dashboard:** The app's dashboard page gives you an overview of the activities that the user object can perform. It allows the user object to navigate through the system.
* **Utils:** This component contains SQL DLL and queries used to create tables when the application starts.
* **Report:** This component will be responsible for generating reports based on attendance data, such as daily attendance reports, monthly attendance reports, and custom attendance reports.
* **Scan Device:** This contains codes to help the system scan for active cameras dynamically.
* **Alert:** The alert component helps display some piece of information to users based on their actions. The design of this component was made reusable by providing an interface for dynamic message passing and calling.
* **Packaging:** This component contains the library import header files which are used by the other components. All library header files are nicely arranged in this component
* **Reset:** This helps the users to reset their credentials if forgotten.
* **Logout:** This component is called when a logged-in user wants to exit their current session. It performs the necessary operations for the data to be stored in the database.
* **Camera\_1:** This is responsible for communicating with the integrated surveillance system cameras. It contains all the necessary codes to start and stop the connected camera and even to enhance image capturing quality through the use of image processing.
* **Camera\_2:** This is responsible for communicating with the integrated surveillance system cameras. It contains all the necessary codes to start and stop the connected camera and even to enhance image capturing quality through the use of image processing.
* **Camera\_3:** This is responsible for communicating with the integrated surveillance system cameras. It contains all the necessary codes to start and stop the connected camera and even to enhance image capturing quality through the use of image processing.
* **Camera\_4:** This is responsible for communicating with the integrated surveillance system cameras. It contains all the necessary codes to start and stop the connected camera and even to enhance image capturing quality through the use of image processing.
* **Configuration:** This is responsible for configuring the integrated surveillance system cameras. This component is made available to users with elevated system privileges.
* **Merge:** For management to use the data generated by the application to make reports, the system needs to provide a way to consolidate data from different facilities in case the system is scaled up. This component provides an interface for the database connection and also helps push locally generated data to be partitioned and pushed to servers online.
* **Thread:** To make the system responsive during the long-running task, the system must be design in such a way that these tasks can be executed in the background allowing the user to still use the system. The thread component helps to achieve this goal. This component contains codes of different threading mechanisms and is been used by the most long executing tasks in the system.
* **Model:** This contains model classes to help capture data from users for storage. These are data classes or pydantic objects with validations.
* **Mail:** This component helps in mailing application data (generated reports) to interested stakeholders and sending the generated QR-Codes to the individual students. This component heavily depends on the thread component for sending emails as a background task.
* **Database/API:** For the application to achieve its purpose, data needs to be stored and retrieved from a data source. This component helps with the communication between the application and the data source or any REST API endpoints.

## 2.5 PROPOSED SOFTWARE FEATURES

* **User-Friendly Interface**: The system will have a user-friendly interface that will be easy to use for both administrators and users.
* **Student management**: The software will have student management functionality, allowing administrators to create and manage students’ details.
* **E-mailing data or reports**: The software will have the ability to send generated reports or exported data to facility managers.
* **Attendance history**: The software will maintain a record of attendance history, allowing teachers and administrators to track attendance over time.
* **Integration with other systems**: The software will have the ability to integrate with other systems and software, allowing for the sharing of attendance data with other stakeholders.
* **Integrated surveillance system**: The software will have surveillance system functionality, allowing administrators to sense what is happening within the facility.
* **Security**: The software will have robust security measures in place to protect against data breaches and other security threats, ensuring the integrity and confidentiality of attendance data.
* **Customized reports**: The software will have the ability to generate customized reports and charts to help visualize attendance trends and patterns. This will allow teachers and administrators to identify areas of concern and take corrective action.

## 2.6 DEVELOPMENT TOOLS AND ENVIRONMENT

* **Python PyQt5**: Python PyQt5 is a set of Python bindings for the Qt application framework. Qt is a popular cross-platform development framework that provides a comprehensive set of libraries and tools for creating graphical user interfaces (GUIs) and other applications. PyQt5 allows developers to use Qt functionalities within Python applications, enabling the creation of robust and visually appealing desktop applications.
* **Integrated development environment (IDE)**: An IDE is a software application that provides a range of tools and features for software development, including a text editor, debugging tools, and integration with version control systems. Visual Studio and PyCharm were used.
* **Database management system (DBMS)**: A DBMS is a software application that is used to manage and organize data, including storing, retrieving, and updating data. MySQL and SQLite were used.
* **Version control system**: A version control system is a software application that is used to track changes to software code and manage the development process. Git and Bitbucket were used.
* **Testing tools**: Testing tools are software applications that are used to test the functionality and performance of software, including unit testing tools, integration testing tools, and performance testing tools. The approach used here was manual testing and pytest library.

**CHAPTER 3**

**METHODOLOGY**

**3.0 OVERVIEW**

A process model in software engineering is a framework or set of guidelines for organizing and managing the development of software. It defines the steps and activities involved in the software development process, as well as the roles and responsibilities of the development team or the developer. The choice of a process model in software engineering depends on the specific needs and constraints of the project. The development team should choose a process model that is appropriate for the project's size, complexity, and requirements.

**3.1 REQUIREMENT SPECIFICATION**

Software Requirement Specification (SRS) is a formal report, that acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements. (*Software Engineering | Software Requirement Specifications - Javatpoint*, n.d.).

**3.2 STAKEHOLDERS OF THE SYSTEM**

The stakeholders of the proposed attendance tracking system software for students will likely include the following parties:

* Students: Students will be the primary users of the software and will be responsible for recording their attendance data.
* Administrators: Administrators will be responsible for managing and maintaining the software, including setting up user accounts, configuring the system, and ensuring that it is secure and reliable.
* School board: The school board will be responsible for providing oversight and support for the attendance tracking system, and may be involved in the decision-making process for the development and implementation of the software.
* IT staff: The IT staff will be responsible for maintaining and supporting the software, including troubleshooting any technical issues that may arise.

**3.3 REQUIREMENT ENGINEERING PROCESS**

Requirements engineering involves three key activities. These are discovering requirements by interacting with stakeholders (elicitation and analysis); converting these requirements into a standard form (specification); and checking that the requirements define the system that the customer wants (validation). However, in practice, requirements engineering is an iterative process in which the activities are interleaved.

## 3.4 FUNCTIONAL REQUIREMENTS

The functional requirements for a system describe what the system should do. Theserequirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements. When expressed as user requirements, functional requirements should be written in natural language so that system users and managers can understand them. Functional system requirements expand the user requirements and are written for system developers. They should describe the system functions, their inputs and outputs, and exceptions in detail. Functional system requirements vary from general requirements covering what the system should do to very specific requirements reflecting local ways of working or an organization’s existing systems.

### 3.4.1 FUNCTIONAL REQUIREMENTS OF THE SYSTEM

* ***Attendance tracking***: The software should have the ability to track and record attendance data for each student, including the dates and times of attendance and any absences.
* ***Customized reports***: The software should have the ability to generate customized reports and charts to visualize attendance trends and patterns, and to identify areas of concern.
* ***Integration with other systems***: The software should have the ability to integrate with other systems and software, such as student information systems, to allow for the sharing of attendance data with other stakeholders.
* ***User roles***: The software should have different user roles, including students, teachers, and administrators, each with different access rights and permissions.
* ***Data management***: The software should have the ability to manage and store attendance data, including the ability to import and export data.
* ***User support***: The software should include user support and training materials to help users understand how to use the software and troubleshoot any issues that may arise.

## 3.5 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. These non-functional requirements usually specify or constrain characteristics of the system as a whole. They may relate to emergent system properties such as reliability, response time, and memory use. Alternatively, they may define constraints on the system implementation, such as the capabilities of I/O devices or the data representations used in interfaces with other systems.

### 3.5.1 SOME NON-FUNCTIONAL REQUIREMENTS

* ***Usability***: The software should be easy to use and navigate, with a clear and intuitive interface. This is important because a user-friendly interface will make it easier for students, teachers, and administrators to use the software, which will help to improve the overall effectiveness of the attendance tracking system.
* ***Scalability***: The software should be able to handle a large number of users and a high volume of attendance data without experiencing performance issues. This is important because the attendance tracking system may be used by a large number of students and teachers, and it needs to be able to handle the associated workload without experiencing performance issues.
* ***Security***: The software should have robust security measures in place to protect against data breaches and other security threats, ensuring the integrity and confidentiality of attendance data. This is important because attendance data is sensitive and needs to be protected from unauthorized access or tampering.
* ***Reliability***: The software should be reliable and should have a high uptime, with minimal downtime or system failures. This is important because the attendance tracking system is a critical part of the facility’s operations, and it needs to be available and reliable at all times.
* ***Performance***: The software should have good performance, with fast response times and minimal delays when accessing or processing attendance data. This is important because delays or slow performance can impact the efficiency of the attendance tracking process and may lead to frustration among users.
* ***Maintainability***: The software should be easy to maintain and update, with a clear and well-documented codebase. This is important because it will help to ensure that the software is well-maintained and can be easily updated and improved over time.

## 3.6 UML DIAGRAMS

Figure 4 Class diagram

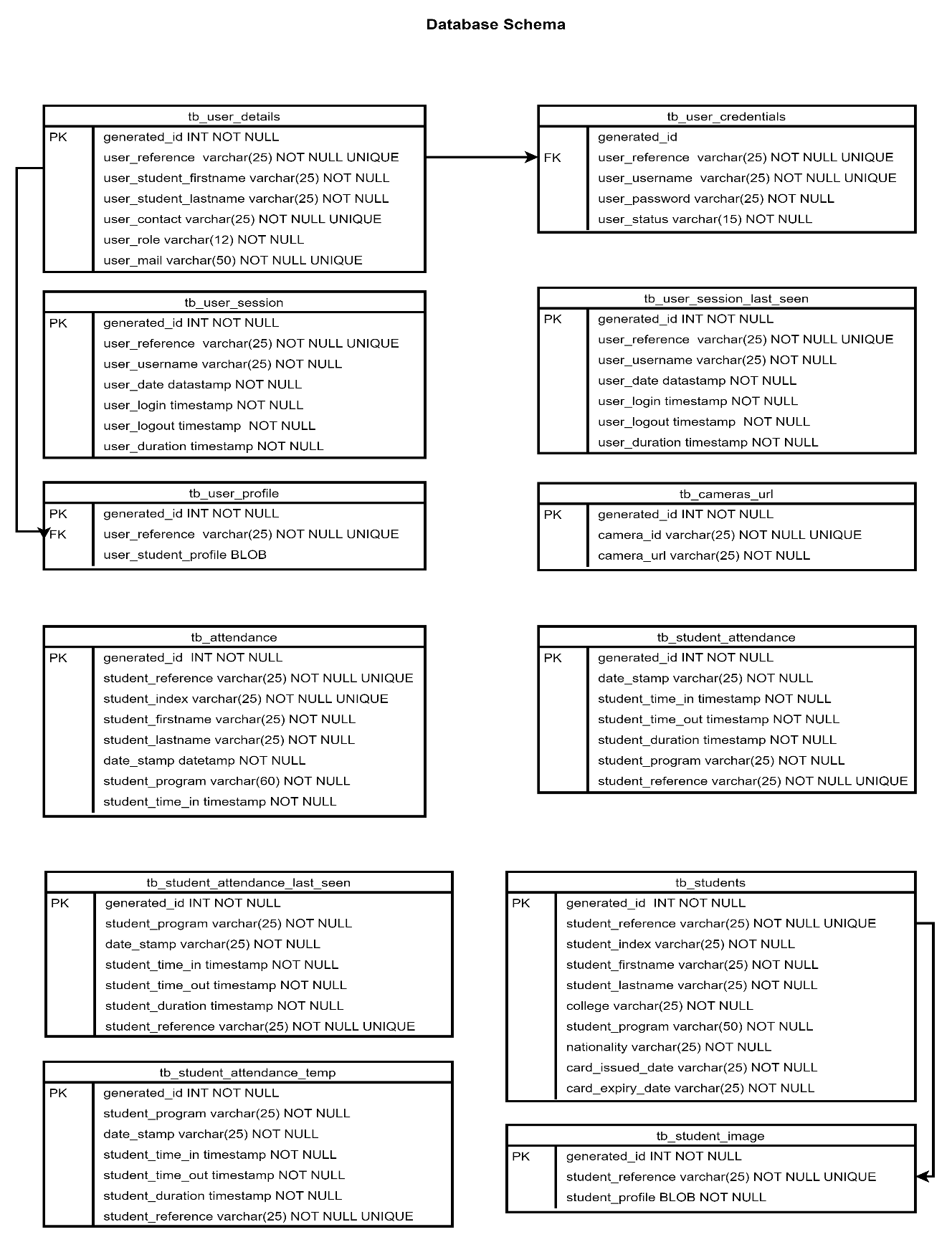
****

Figure 5 Database schema

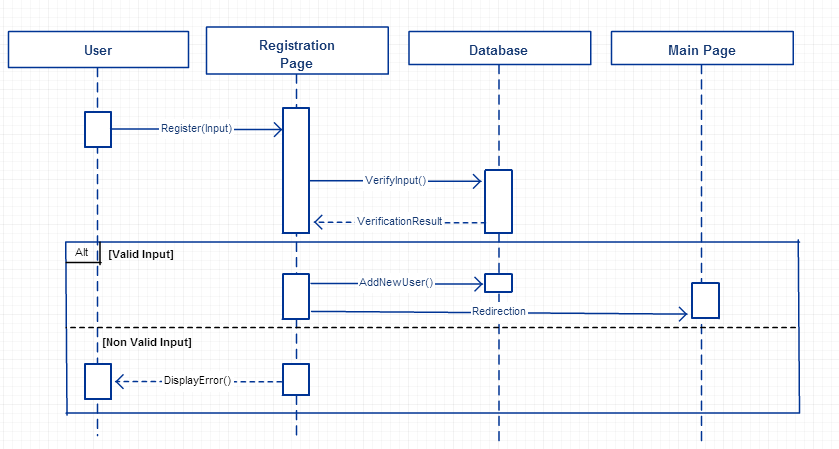
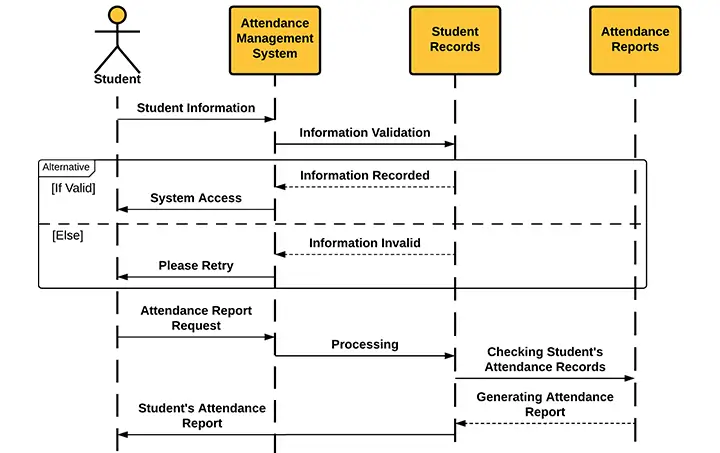


Figure 6 User registration sequence diagram

Figure 7 Student sequence diagram

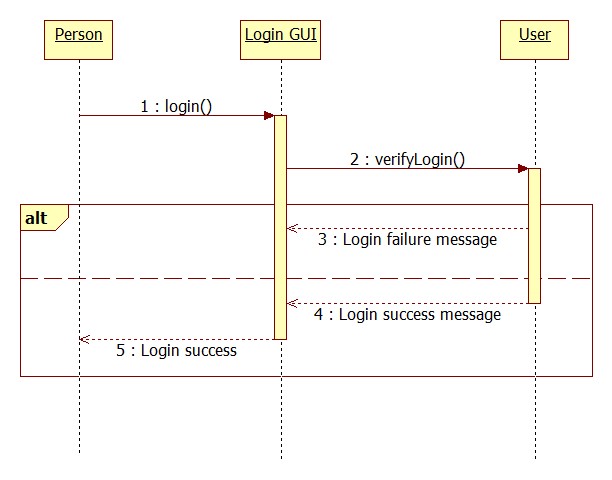


Figure 8 User login sequence diagram

## 3.7 SECURITY CONCEPTS

* Confidentiality: The term "confidentiality" refers to a set of standards and procedures, typically carried out in the context of confidentiality agreements that guarantee that the information is only accessible by specific parties or locations. This system allows only authenticated users and with specific roles to access system data.

## 3.8 PROJECT METHODOLOGY

A set of ideas and practices known as a project management methodology can help you structure your projects to ensure their success. (*Project Management Methodologies - Everything You Need to Know*, n.d.)

Examples of the various Project Methodologies include:

1. Waterfall methodology
2. Scrum methodology
3. Kanban methodology
4. Lean methodology
5. eXtreme Programming (XP) methodology
6. Scrumban methodology
7. Adaptive project framework (APF) methodology
8. Critical chain project management
9. Critical Path Methodology (Critical Path Analysis)
10. New product introduction (NPI)
11. Rapid application development (RAD) methodology
12. Agile methodology

The agile methodology was used in this project due to its unique features which make it different from the other methodologies. Agile values people, their relationships, and interactions through tools; offers collaboration throughout the development process; responds to change rather than following a set plan, and focuses on presenting functional software rather than documentation.

Unlike Plan driven, Agile is well equipped for the complexity and variability of development projects.

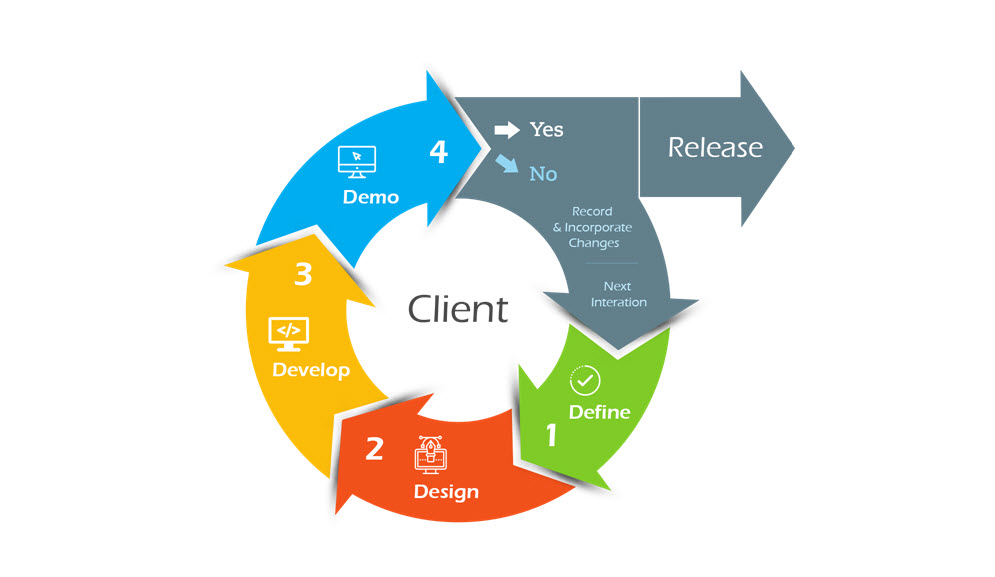
According to the agile approach, we develop in short sprints or iterations, each of which contains a defined duration and a list of tasks to be delivered, but in no specific order. During sprints, we work to deliver functional software or other tangible and verifiable results.

However, upon careful and close examination of the system requirement, the agile software process model was the most appropriate for the proposed system due to changing requirements or needs of stakeholders. Below are some listed steps involved in the agile approach.

* ***Define project scope and objectives***. The first step in the project will involve defining the scope and objectives of the attendance tracking system. This will involve working with stakeholders (e.g., students, teachers, administrators) to understand their needs and requirements, and identifying the key features and functionality that the system should include.
* ***Design and develop an initial increment***. Based on the defined scope and objectives, the team will design and develop the initial increment of the attendance tracking system. This will involve creating a high-level design for the system, identifying the technologies and frameworks that will be used, and developing the core functionality of the system.
* ***Test and deploy the initial increment***. Once the initial increment of the attendance tracking system has been developed, it will be tested to ensure that it meets all requirements and is ready for deployment. This may involve conducting unit testing, integration testing, and user acceptance testing. Once the testing is complete, the initial increment will be deployed to a small group of users for further testing and feedback.
* ***Gather feedback and refine the system***. Based on the feedback received from the initial group of users, the team will gather and analyze data to identify areas for improvement and refinement. This may involve making changes to the design or functionality of the system or adding new features.
* ***Develop and deploy additional increments***. The process of developing and deploying additional increments of the attendance tracking system will be repeated until the system is complete and all requirements have been met. Each increment will build on the previous one, adding new functionality and features to the system.
* ***Maintenance and support***. Once the attendance tracking system has been fully developed and deployed, the team will be responsible for providing ongoing maintenance and support to ensure that the system remains reliable and efficient. This may involve fixing bugs, adding new features, and providing technical support to users who encounter issues with the system.

### 3.8.1 SOME BENEFITS OF INCREMENTAL DESIGN APPROACH

* It allows the software to be developed and delivered more flexibly and iteratively, which can be helpful in situations where the requirements are not well-defined or are subject to change.
* It allows the team to get feedback from users and stakeholders at an early stage, which can help to identify and address issues before they become more difficult to fix.
* It allows the team to break the development process down into smaller, more manageable chunks, which can make it easier to estimate and plan the project



## 3.9 VARIOUS SOFTWARE PROCESSES

Figure 9 Agile process model

A software process is a collection of actions with a corresponding result that results in a software product. These tasks are mainly done by software engineers. All software processes share these four fundamental process steps. These processes include:

* ***Software specifications***: The software's capabilities and restrictions on operation must be specified. This procedure involves a thorough description of the software system that will be created, together with its functional and non-functional requirements.
* ***Software development***: It is necessary to create the software to satisfy the need. Designing, programming, documenting, testing, and bug fixing are all done during this phase.
* ***Software validation***: To make sure the program fulfills the customer's requirements; it must be validated. In this process, software products are evaluated to make sure they satisfy both the needs of end users and business requirements.
* ***Software evolution***: To adapt to changing customer needs, the software must also change. Software is first created, and then it is promptly updated for several reasons. (*Software Processes - Javatpoint*, n.d.)

## 3.10 CHOSEN MODEL AND JUSTIFICATION

Depending on the type and functionality of the proposed system, different types of software development process models are used for different types of software. The requirements were determined after a careful examination of the existing systems.

The approach used in system development is the incremental process model. The incremental development model is based on the idea of developing the first implementation, exposing it to user feedback, and developing it in multiple versions until a suitable system is developed. Specification, development, and validation activities are nested rather than separated, with quick comments on all activities. The model is divided into three simultaneous activities:

***Requirements elicitation***

The aims of the requirements elicitation process are to understand the work that stakeholders do and how they might use a new system to help support that work. During requirements elicitation, software engineers work with stakeholders to find out about the application domain, work activities, the services and system features that stakeholders want, the required performance of the system, hardware constraints, and so on.

***Requirements specification***

Requirements specification is the process of writing down the user and system requirements in a requirements document. Ideally, the user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent. In practice, this is almost impossible to achieve. Stakeholders interpret the requirements in different ways, and there are often inherent conflicts and inconsistencies in the requirements.

***Requirements validation***

Requirements validation is the process of checking that requirements define the system that the customer wants. It overlaps with elicitation and analysis, as it is concerned with finding problems with the requirements. Requirements validation is critically important because errors in a requirements document can lead to extensive rework costs when these problems are discovered during development or after the system is in service.

# CHAPTER 4

# IMPLEMENTATION AND RESULTS

## 4.0 OVERVIEW

The system implementation and documentation are the phase of system development that constitutes proof of the making of the new system. Planning the implementation of a system begins in the detailed design stage. This phase of the project will involve the installation and testing of a thoroughly structured program to make sure that the result of the new system is the anticipated one. Most system failures are due to inadequate conduct of the implementation phase. This phase is very important because the success of the project depends on it.

## 4.1 MAPPING LOGICAL DESIGN ONTO PHYSICAL PLATFORM

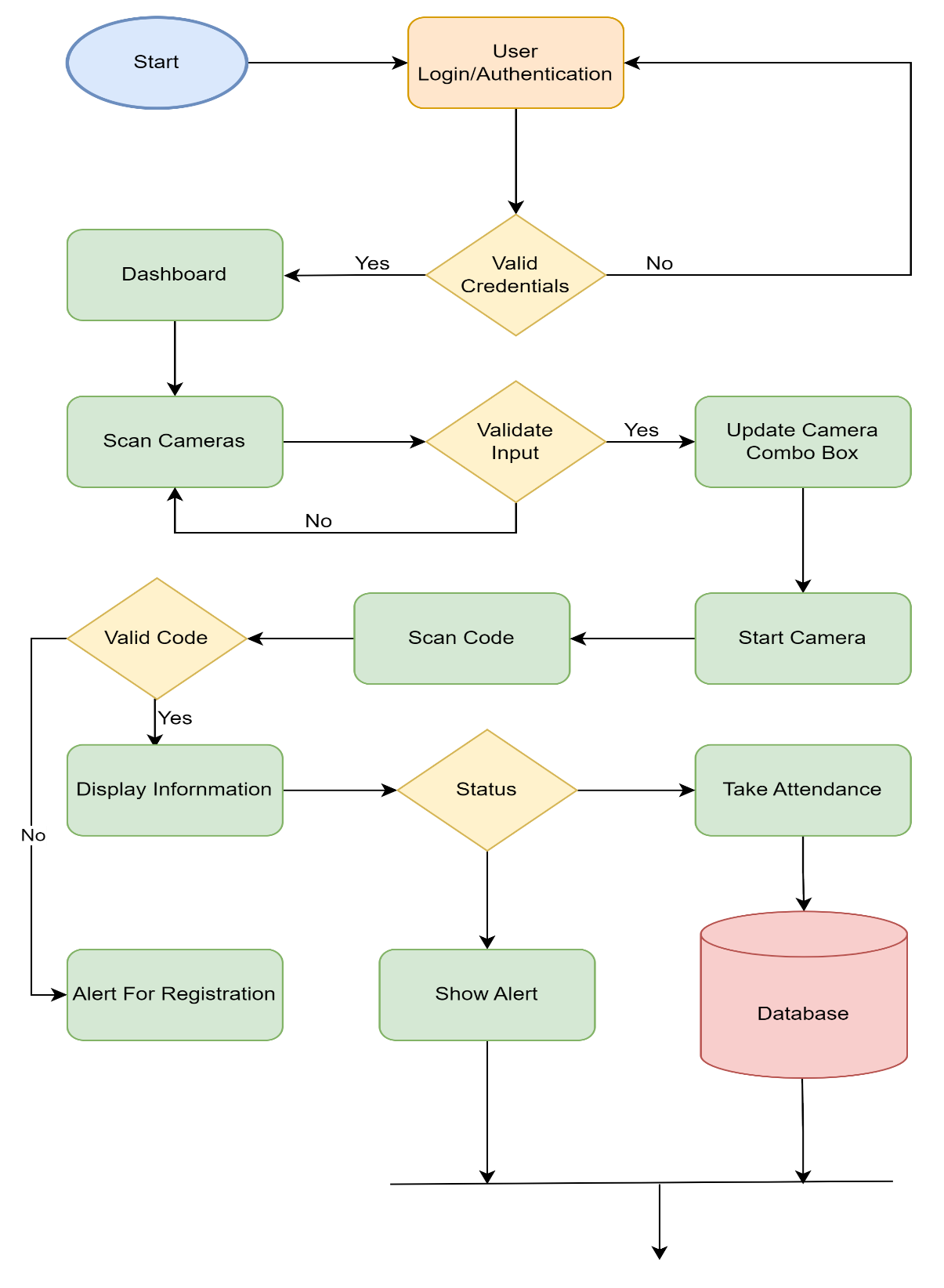
****

Figure 10 Flowchart for the system

## 4.2 CONSTRUCTION

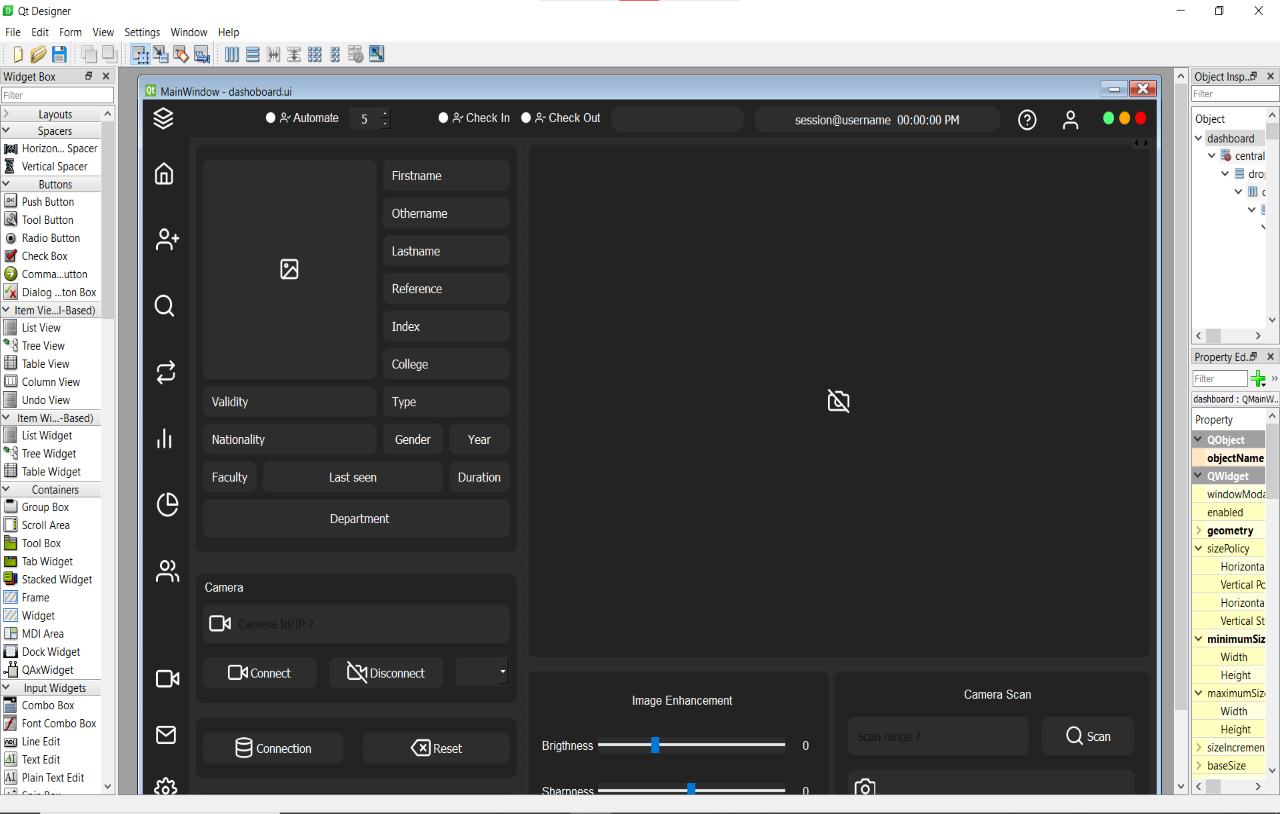
****

Figure 11 Fronted designing

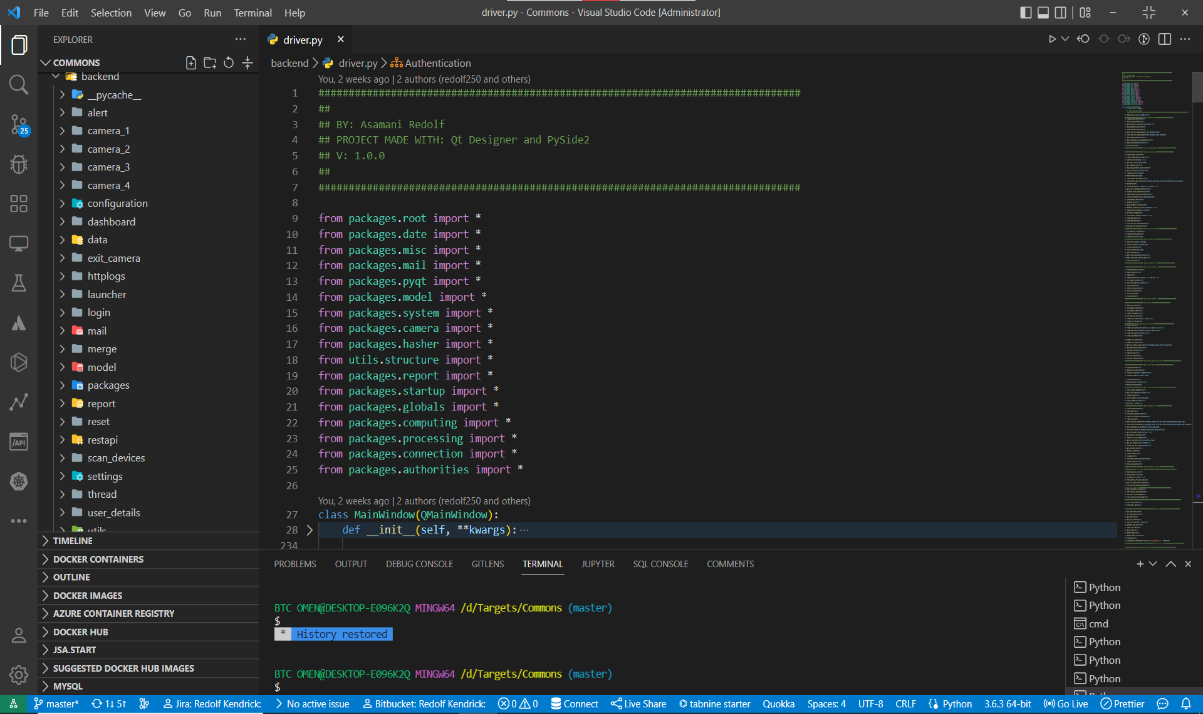
****

Figure 12 Backend coding



Figure 13 Launcher screen

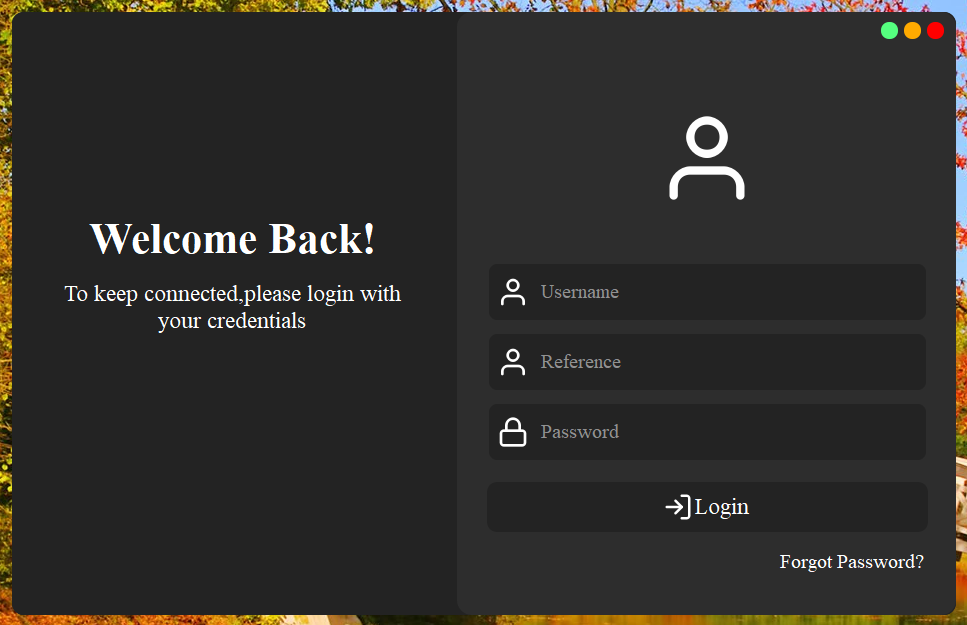


Figure 14 Login screen

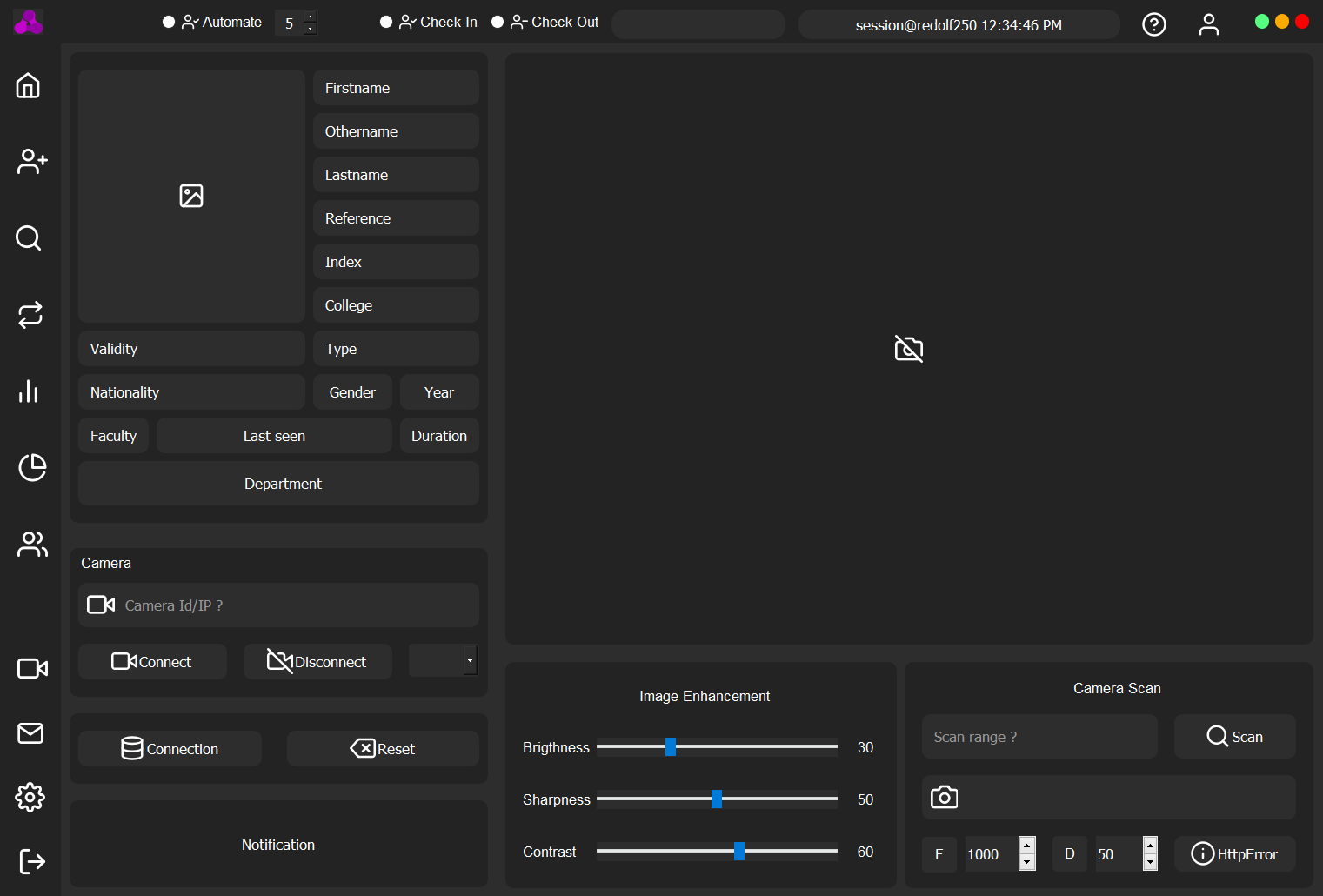
******

Figure 15 Dashboard

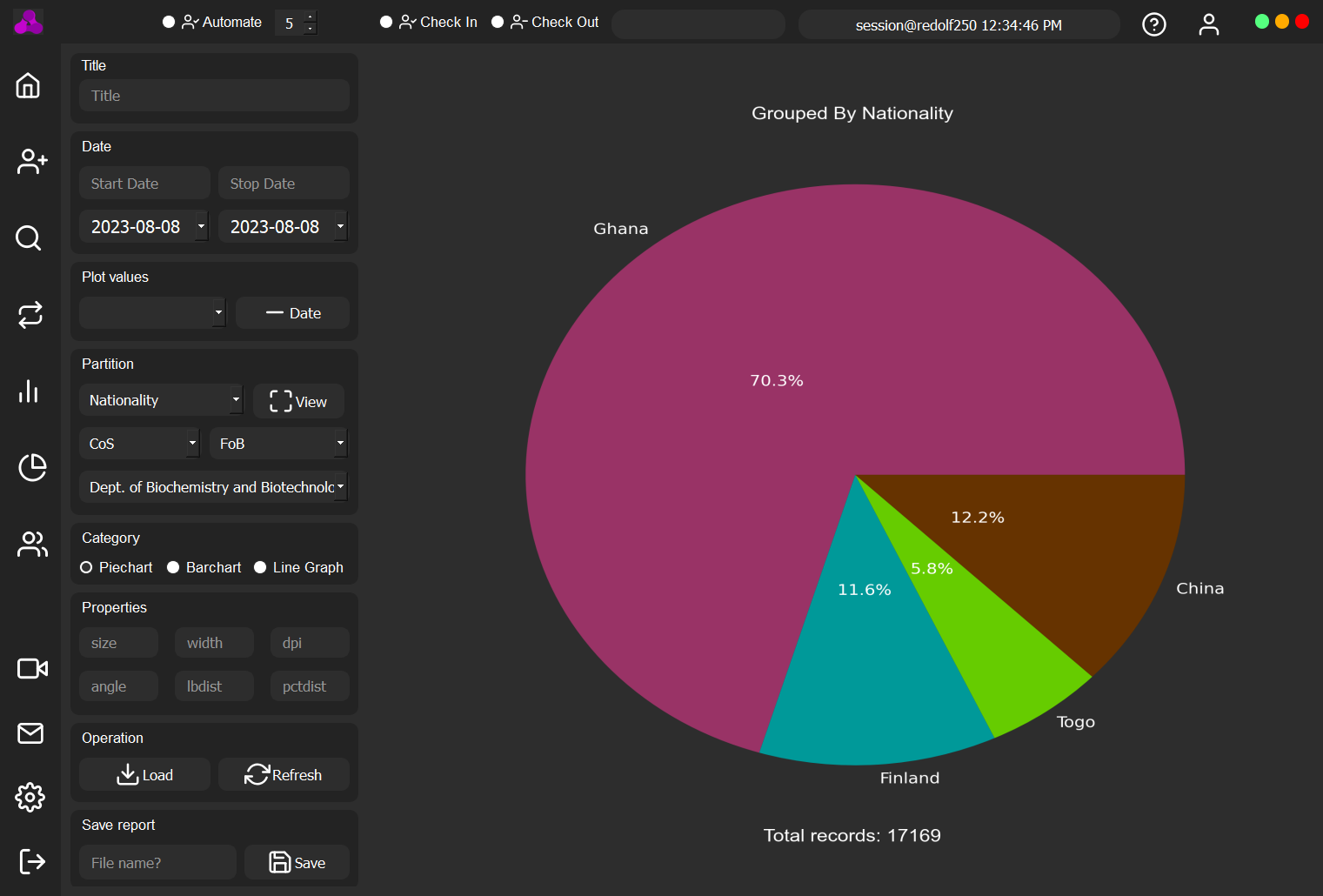


Figure 16 Local report generation

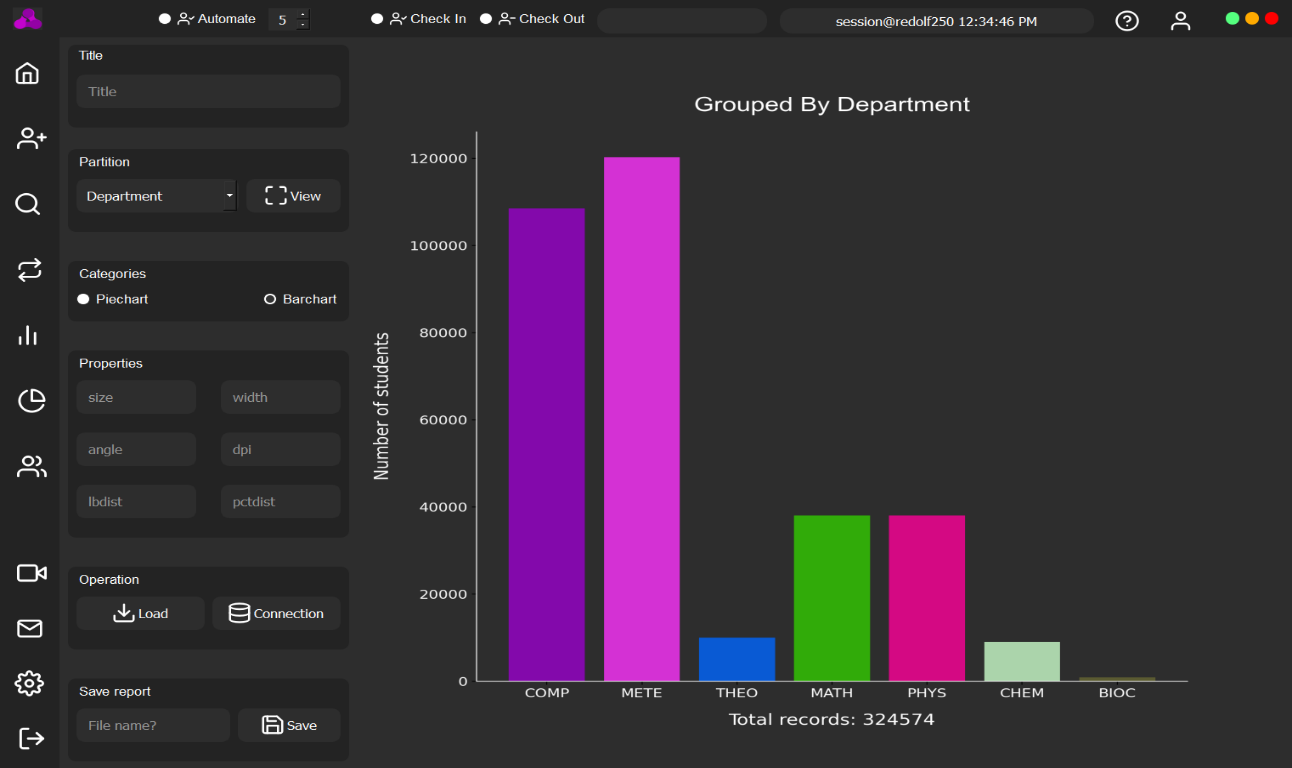


Figure 17 Central records report generation

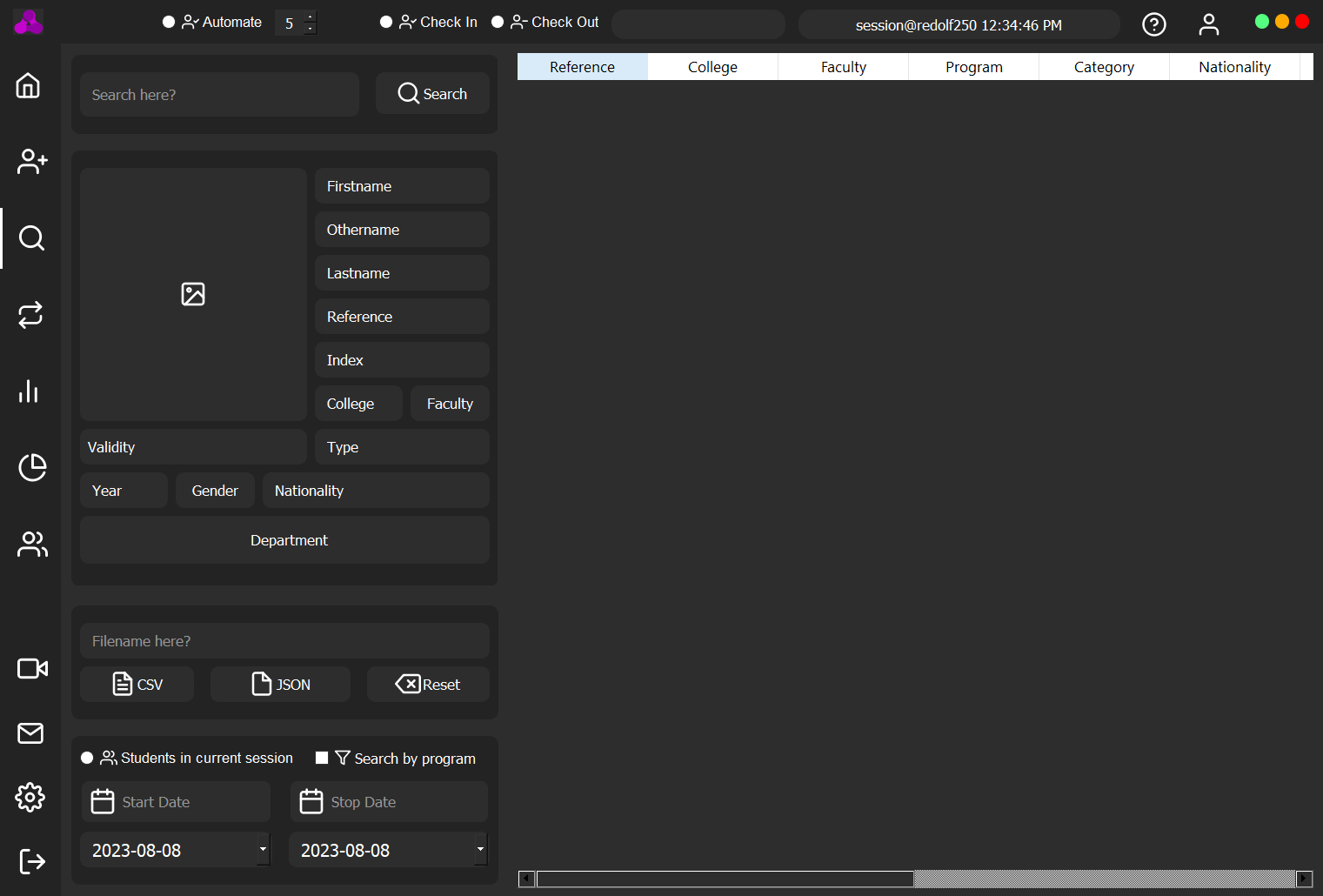


Figure 18 Retrieve records

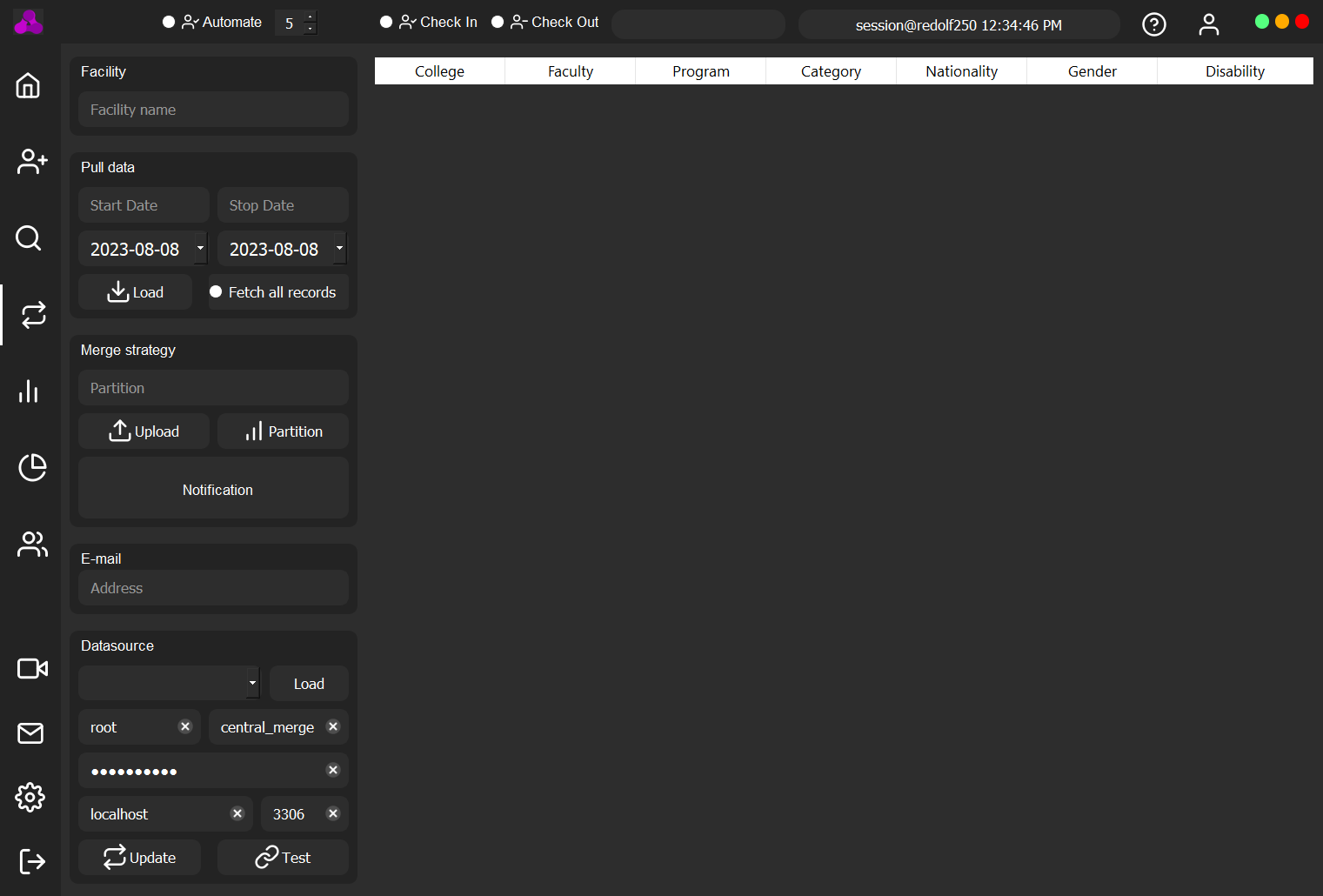
**

Figure 19 Push local records to server

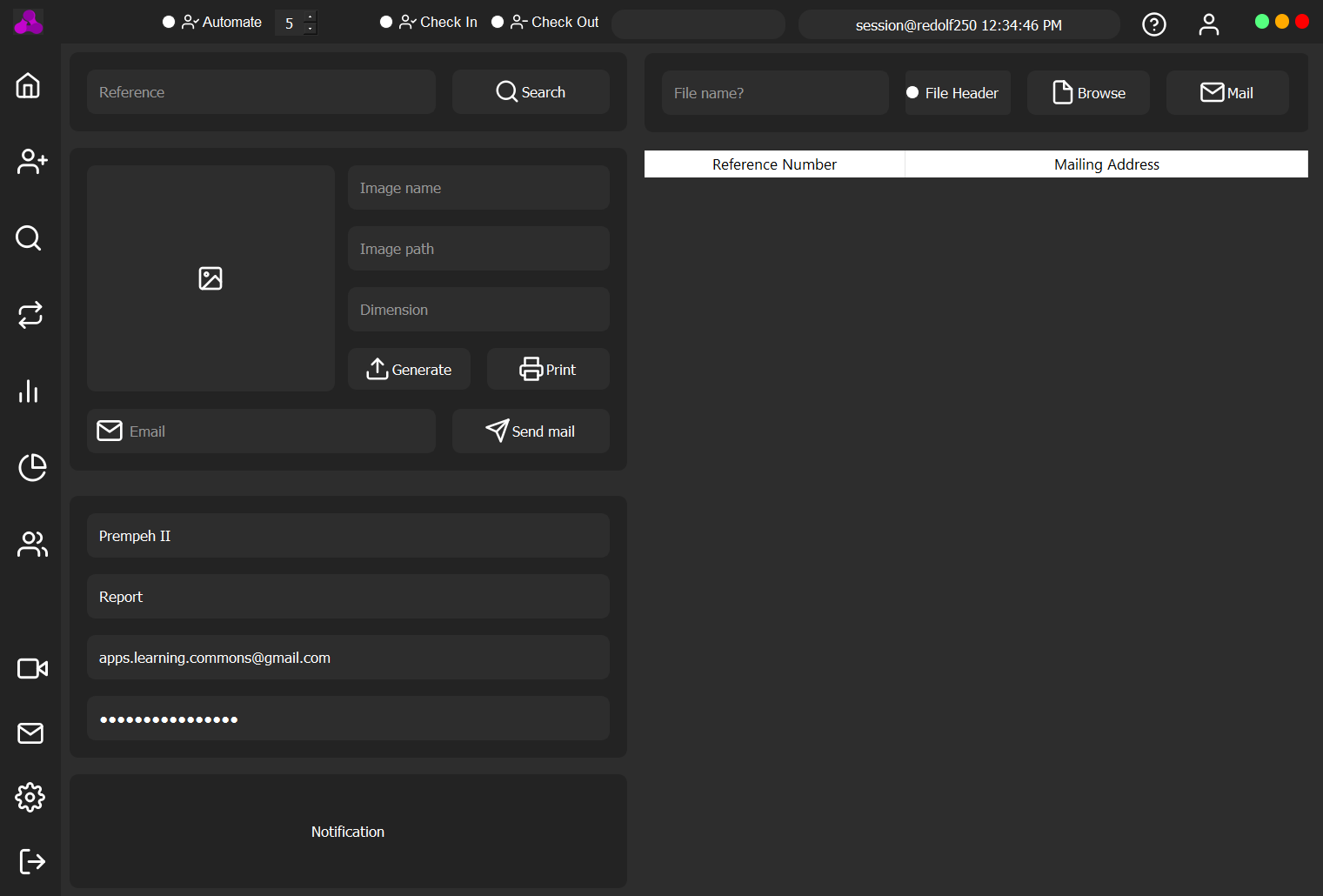
**

Figure 20 Account registration

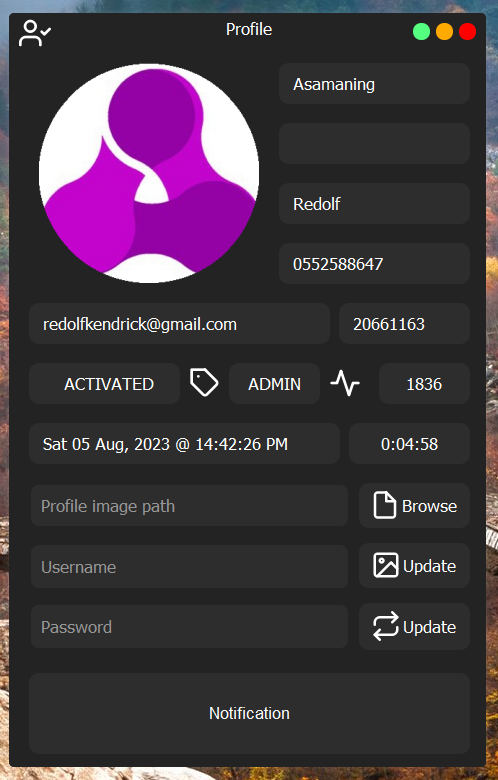
**

Figure 21 Current Logged in administrator

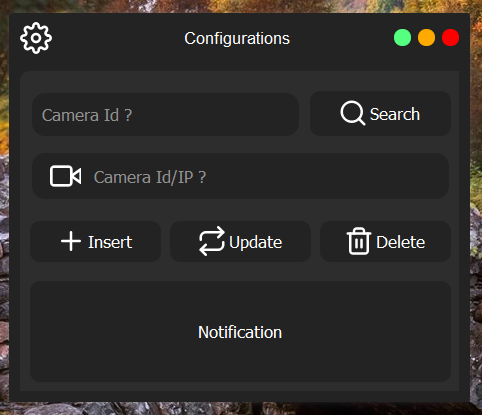
**

Figure 22 Integrated surveillance configuration

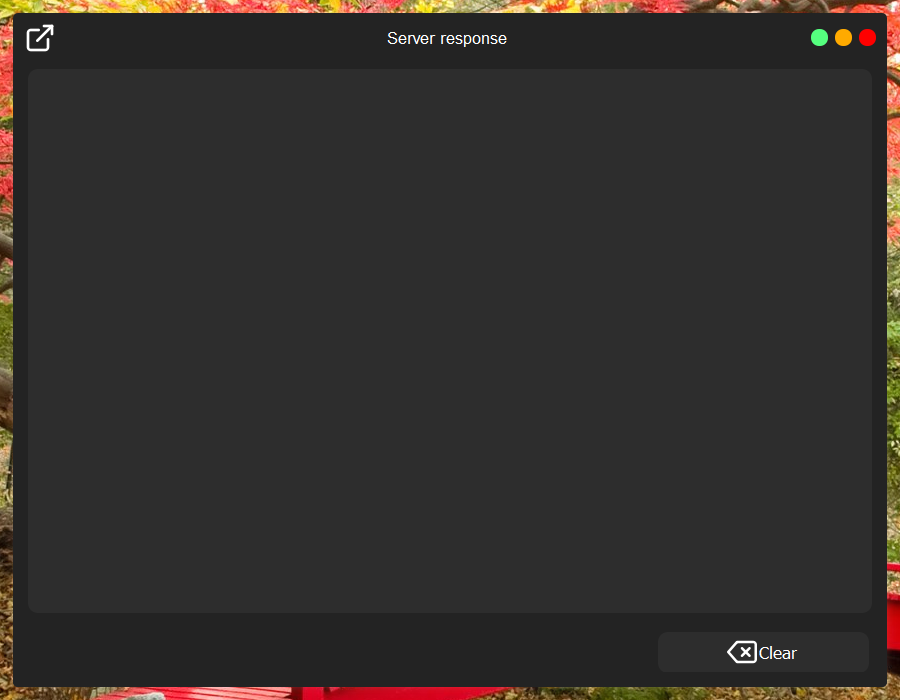
**

Figure 23 Web server response frame

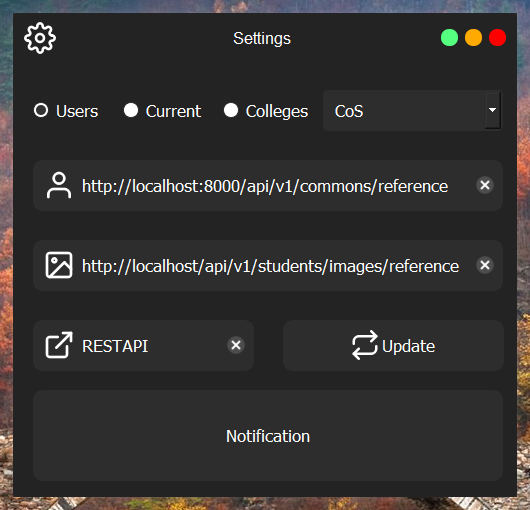
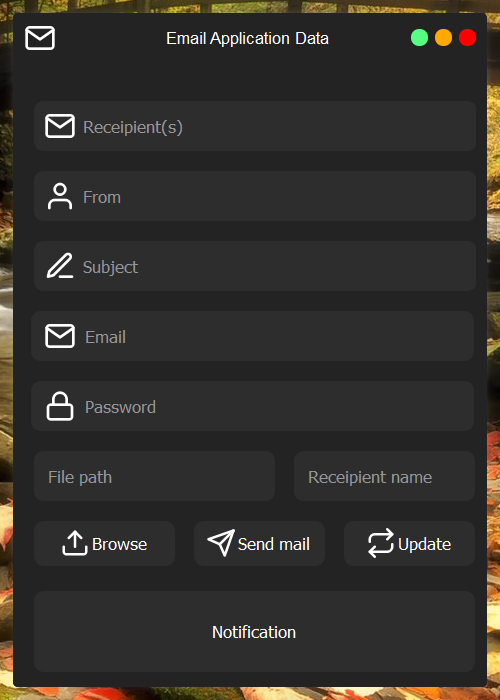
**

Figure 24 Web request URL configuration

**

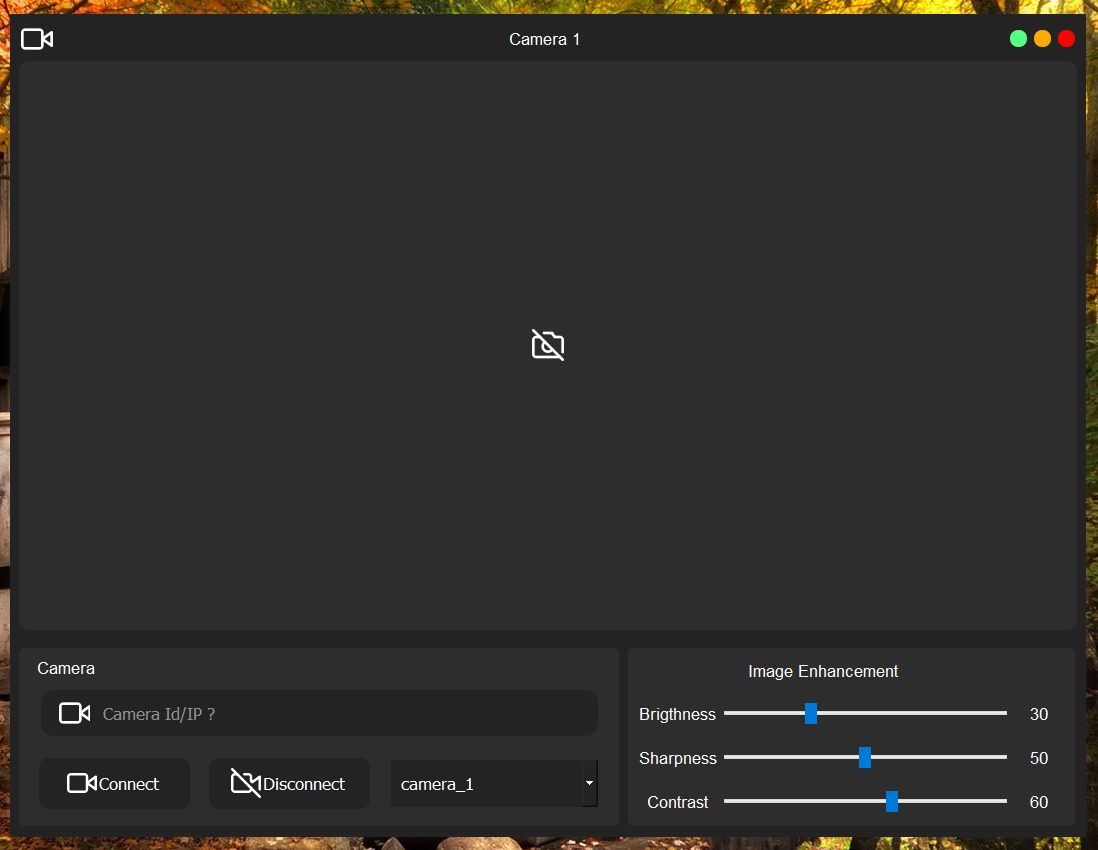
Figure 26 Mailing of reports frame

Figure 25 Integrated surveillance frame

## 4.3 TESTING

In the native desktop application, there are a few ways to test the components. Broadly, they divide into two categories:

* Running a complete app on a realistic desktop environment (also known as “end-to-end” tests).
* Rendering component trees in a simplified test environment and asserting their output.

This write-up section focuses on testing strategies for the first case. While full end-to-end tests can be very useful to prevent regressions to important workflows, such tests are not concerned with PyQt5 components in particular and are out of the scope of this section. The system was tested on Ubuntu 22.2 and Kali Linux 20.2 all running in virtual environments and Windows 10 Pro was further used for the final testing as it was the main development environment. The system worked perfectly well.

### 4.3.1 TOOLS USED IN TESTING

* **Unit-test (also known as PyUnit)**

Unit test is a built-in testing framework in Python, inspired by the Java testing framework JUnit. It provides a set of classes and methods for writing and running tests. Unit test allows developers to define test cases as classes that inherit from the unit test. Test Case class. Test methods within these classes are prefixed with the word "test" and can use a variety of assertion methods to verify expected behavior. Unit test provides various features such as test discovery, test fixtures, and test runners. It supports test automation, test isolation, and the ability to organize tests into test suites. With its extensive documentation and wide adoption, unit test is a reliable choice for testing Python code.

* **pytest**

pytest is a popular third-party testing framework for Python that provides a more concise and expressive way of writing tests compared to unit test. It aims to make testing easier and more readable by leveraging Python's features, such as function decorators and assertions. pytest can be used as a drop-in replacement for unit test and offers additional features and plugins. pytest discovers tests automatically, without requiring test classes or test case inheritance. It allows you to write test functions using plain Python functions, which reduces boilerplate code. pytest supports powerful assertions, fixtures for test setup and teardown, parameterized testing, and test coverage reporting.

Python (PyQt5) supports three different kinds of testing.

* **Unit test**: A unit test examines how a method or class behaves.
* **Widget test**: Without launching the app itself, a widget test validates the functionality of Flutter widgets.
* **End-to-end test**: Run the entire application during an integration test, often known as end-to-end testing or GUI testing.

### 4.3.2 REASONS FOR TESTING

* ***Unit testing***: Testing is a critical practice in software development that involves testing individual units of code to ensure their functionality and correctness. Here are five key reasons why unit testing is important.
* ***Early detection of defects***: Unit testing allows for the early detection of defects and bugs within individual units of code. By testing each unit in isolation, developers can identify and address issues at an early stage, before they become more complex and harder to fix. This helps in reducing the overall development time and cost.
* ***Improved code quality and maintainability***: Unit testing encourages developers to write modular and well-structured code. Writing testable code requires breaking down complex functionalities into smaller, testable units, leading to code that is easier to understand, maintain, and refactor. Unit tests act as a safety net, providing confidence that changes or additions to the codebase will not break existing functionality.
* ***Faster development iterations***: Unit tests enable developers to quickly verify the behavior of individual code units. When changes are made to the codebase, unit tests can be run to ensure that the modifications haven't introduced any unintended side effects. These speeds up the development process by reducing the need for manual testing and allowing for frequent iterations without sacrificing quality.
* ***Facilitates refactoring and regression testing***: Unit tests provide assurance during code refactoring. When refactoring code, developers can run unit tests to validate that the changes made have not introduced any regressions or functional defects. This allows developers to confidently make improvements and optimize code without the fear of breaking existing functionality.
* ***Promotes collaboration and code reusability***: Unit testing encourages collaboration among team members. Writing unit tests forces developers to think about how their code interacts with other units and dependencies. It promotes a modular and loosely coupled architecture, enabling easier integration and reusability of code components. Unit tests also serve as documentation, providing examples and usage scenarios for other developers who may need to work with the codebase in the future.

## 4.4 RESULTS

Following system implementation, system testing was carried out to evaluate the system's performance. System testing is a crucial step in the system development process. It functioned once the actual system or prototype has been constructed, or after development. The testing phase is very beneficial and significant since it might uncover hidden software bugs. User and system requirements are typically the foundation of testing criteria, which determines whether or not the system satisfies the requirements. System dependability is crucial for a system to function correctly for its end users, and testing can confirm this. Unit testing and module integration testing are the two components of system testing. A type of testing on each component in a big system is called unit testing. Unit testing is done on each module to ensure that it is fully functional before module integration. The procedure of combining all modules would be tested during the module integration testing. The ultimate system is complete after all modules can communicate with one another, and integration testing would test the entire system.

### 4.4.1 RESULTS FOR TESTING

Table 1 PyQt5 application module - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure the application launches successfully | Run the driver.py file to start application using the terminal | The application should run successfully without errors | Passed |
| 2 | To ensure that users(administrators) can login with required credentials | Login to iAttend and get authenticated | User should be able to login to access the system | Passed |
| 3. | To ensure the navigation buttons take a user to the required location  on the application | Toggle between each of the navigation tabs to switch from one page to the other | There should be a smooth transition between the pages in the application | Passed |
| 4 | To ensure the application can scan for active connected cameras | 1. User provides valid scan range. 2. User toggles the scan button. | The active camera combo box should display the all-active camera connected to it and as well display notification. | Passed |
| 5 | To ensure application can connect to camera over IP address | 1. User can provide a link to IP camera. 2. User can toggle the connect button. | The camera should open smoothly with any problem | Passed |

Table 2 Student attendance taking - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure students can clock in | Students scans valid QR code to clock in the application should  smoothly | The application should scan the code and retrieve student details and take the attendance | Passed |
| 2 | To ensure students can clock out | Students scans valid QR code to clock out the application should  Smoothly. | The application should scan the code and clock out the student | Passed |

Table 3 Code generation, mailing, printing and searching - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can generate codes for students | Students provide their reference numbers which administrators use to generate codes | The application should be able to generate the code base on the reference number provided | Passed |
| 2 | To ensure administrators can mail the generated codes to students’ mail box | 1. Students provides valid mail address. 2. Administrators also configure their mailing account credentials. 3. Toggling the send button should send the code. | This should be done without any errors from the application end with internet connection | Passed |
| 3 | To ensure administrators can print the soft copy of the code | 1. Administrators can either search for the code if it is already generated or generate new one based on the student reference number. 2. Toggling the print button should open the print dialog box showing list of available printers to choose from. | The application should be able to perform this functionality with ease without encountering problems | Passed |
| 4 | To ensure that administrators can search for generated codes | 1. Administrators enter student reference number into the search field provided. 2. Toggling the search button should retrieve the code if present or show alert message. | The application should be able to perform this functionality. | Passed |

Table 4 Record retrieval and exporting - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can fetch saved records | 1. Administrators can use some predefined strategies for data retrieval shipped with the application to perform such. 2. Upon toggling the search button, the data should be retrieved if present or show alert. | The application should be able to do this smoothly | Passed |
| 2 | To ensure administrators can save or export records | 1. Upon searching for records using predefined strategies. 2. Providing a file name in the filename field and toggling the export button should save the records in CSV or JSON format | The application should be able to do this smoothly | Passed |

#### Table 5-Report generation and exporting - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can generate reports | 1. Administrators can use some predefined strategies for report generation shipped with the application to perform such. 2. Upon toggling the load button, the data should be retrieved to generate the report. | The application should be able to do this smoothly | Passed |
| 2 | To ensure administrators can save reports | 1. Upon generating reports using predefined strategies. 2. Providing a file name in the filename field and toggling the export button should save the report in PDF format | The application should be able to do this smoothly | Passed |

#### Table 6- Push records from local server to online server- Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrator can export locally generated data to database hosted on a server | 1. Administrators can set some predefined strategies for retrieving and pushing locally generated data to hosted server. 2. Toggling the push button should execute this functionality | The application should perform this functionality smoothly | Passed |
| 2 | To ensure administrators of online hosted database are notified upon push records. | 1. Local database administrators can set mailing address for hosted database servers administrators mail address. 2. After pushing the records mails should be sent to the administrators. | The application should send the mail to the administrators if only the address is valid and internet connection is present. | Passed |

Table 5 Administrator account details updating - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can update their details | 1. Administrators upon providing their new details in the fields provided on the current login user dialog pane. 2. The system should validate the new details before updating. 3. Toggling the update button should ensure option 2. | The application should be able to update the details | Passed |
| 2 | To ensure administrators can reset passwords if forgotten | 1. Administrators provide their reference number and username. 2. Upon toggling the reset button from the forgot password dialog the old password should be updated to the new password if user is present in database and send mail to notify the user or alert user if not present | The application should be able to perform this smoothly | Passed |

Table 6 Camera feed enhancement - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can use integrated surveillance cameras | 1. Toggling the surveillance button opens four integrated dialogs for connecting to cameras. 2. Administrators can choose from a list of saved camera IP addresses or URL. 3. Toggling the connect button should display feeds from the camera and toggling the disconnect button should stop the camera. | The application should be able to connect to cameras if the IP/URL provided is valid | Passed |
| 2 | To ensure administrators can enhance integrated surveillance cameras feeds | Administrators can use the sliders provided below the dialog to enhance the quality of the feed | The application should be able to perform this smoothly | Passed |

Table 7 System configuration - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can configure the system | Administrators can use the settings dialog to change the default settings of some parts of the system | The application should to load the new configuration | Passed |

Table 8 Report mailing - Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| 1 | To ensure administrators can mail generated reports to stakeholders | 1. Toggling the mail dialog button should open the dialog. 2. Toggling the browse button should open system file picker dialog box to select file in CSV, JSON and PDF format. 3. Toggling the send button should send the mail. | The application should to send the mail provided the address is valid and internet connection is available | Passed |

# CHAPTER 5

# FINDINGS AND CONCLUSION

## 5.0 OVERVIEW

This chapter delves into the valuable lessons learned during the implementation and usage of the QR-code-based attendance tracking system for study facilities. It also addresses the limitations and challenges encountered throughout the process. This chapter aims to provide insights into the system's effectiveness, usability, and implications for future improvements.

## 5.1 FINDINGS

* ***Increased Attendance Compliance***: The implementation of a QR-code-based attendance tracking system resulted in a significant increase in attendance compliance among students and staff. The ease of scanning QR codes with their smartphones provided a convenient and user-friendly method for marking attendance, leading to more accurate and reliable data.
* ***Data Analysis for Student Performance***: The digital nature of the QR-code-based attendance tracking system facilitated the integration of attendance data with other academic performance metrics. Researchers and educators could analyze the correlation between attendance and academic performance, allowing for evidence-based interventions and support systems to be put in place for students who were consistently absent.
* ***Time Efficiency***: Compared to traditional manual attendance taking methods, the QR-code-based system proved to be much more time-efficient. Instructors and administrative staff reported that the process of scanning QR codes took only a few seconds per individual, resulting in saving time that could be allocated to more productive tasks.

## 5.2 CONCLUSION

In conclusion, the implementation of a QR-code-based attendance tracking system for study facilities has demonstrated numerous advantages over traditional manual methods. The findings presented here highlight the positive impact it has had on attendance compliance, time efficiency, real-time monitoring, reduction of proxy attendance, and data analysis for student performance.

By adopting this digital solution, study facilities have witnessed a significant increase in attendance compliance among students and staff. The simplicity and convenience of scanning QR codes with smartphones have encouraged greater participation, leading to more accurate and reliable attendance data.

Furthermore, the system's time efficiency has been commendable, enabling instructors and administrative staff to save valuable time that can now be redirected toward more productive tasks in education and management.

Real-time attendance monitoring has empowered educators and administrators with instant access to attendance data, allowing them to identify attendance patterns and intervene promptly to address any attendance-related issues.

Additionally, the integration of attendance data with other academic performance metrics has offered valuable insights into the correlation between attendance and student outcomes. This data-driven approach has enabled evidence-based interventions and support systems to be implemented, potentially improving academic performance and overall student success.

## 5.3 LIMITATIONS OF THE SYSTEM

* ***Dependency on Technology***: The QR-code system relies heavily on technology, such as smartphones and stable internet connectivity. In case of technical failures or network issues, the system's effectiveness may be compromised, leading to inaccuracies in attendance data and potential disruptions in the learning process.
* ***Exclusion of Non-Smartphone Users***: While smartphones are prevalent, some students or staff may not have access to such devices. Implementing a QR-code system may inadvertently exclude these individuals, leading to disparities in attendance monitoring and affecting the overall reliability of attendance data.
* ***Accessibility Challenges for Students with Disabilities***: QR-code scanning may present accessibility challenges for students with visual impairments or motor disabilities who may struggle to scan the codes independently. This could result in an incomplete representation of their attendance and potentially create inequities in tracking their participation.

## 5.4 LESSON LEARNT

* ***Technology Redundancy Planning***: The implementation process highlighted the importance of having contingency plans in place to deal with technological failures. Institutions should have backup measures, such as manual attendance sheets or alternative tracking methods, to ensure data accuracy and continuity of attendance monitoring during technical disruptions.
* ***Comprehensive Training and Support***: Proper training and ongoing support for faculty, staff, and students are vital for the successful adoption and use of the QR-code system. Institutions should invest in training sessions, resources, and assistance to ensure everyone is familiar with the system and can use it effectively.
* ***Continuous Evaluation and Improvement***: Implementing the QR-code system served as a reminder of the importance of continuous evaluation and improvement. Institutions should regularly assess the system's effectiveness, gather feedback from stakeholders, and make necessary adjustments to optimize its functionality and address any emerging challenges.
* ***Inclusivity and Accessibility Considerations***: To ensure inclusivity, institutions should explore alternative attendance-tracking methods for students who do not have smartphones or face accessibility challenges. Providing options like NFC tags or attendance kiosks can help accommodate diverse needs and ensure all students can participate in the attendance tracking process.

## 5.5 RECOMMENDATIONS FOR FUTURE WORK

* ***Integration of Biometric Authentication***: Consider integrating biometric authentication methods, such as fingerprint or facial recognition, as an additional layer of security for marking attendance. This can further enhance data accuracy and reduce the possibility of proxy attendance.
* ***Multi-Modal Attendance Tracking***: Create a multi-modal attendance tracking approach that combines QR codes with other technologies like RFID (Radio Frequency Identification) or NFC (Near Field Communication). This approach allows for flexibility and caters to students with diverse needs and preferences.
* ***User Experience (UX) Research***: Conduct regular user experience research and gather feedback from students, faculty, and administrative staff to continuously improve the system's usability and address any usability issues. Emphasize user-centered design principles to create an intuitive and engaging attendance tracking experience.
* ***Cross-Institutional Collaboration***: Foster collaborations between different study facilities to share best practices and experiences in implementing attendance tracking systems. Pooling resources and knowledge can lead to more innovative solutions and a better understanding of the impact of such systems on student success.

## 5.6 RECOMMENDATIONS FOR COMMERCIALIZATION

* ***Scalability and Customization***: Ensure the QR-code system is scalable and can accommodate a wide range of educational institutions, from small schools to large universities. Offer customization options to meet the unique requirements of different institutions, such as integration with existing student management systems or compatibility with various platforms and devices.
* ***User-friendly Interface and Support***: Emphasize the system's user-friendly interface to make it easy for educators, administrative staff, and students to navigate and use effectively. Provide comprehensive training and ongoing support to ensure smooth adoption and address any technical or user-related issues promptly.
* ***Pilot Programs and Testimonials***: Conduct pilot programs with a select group of educational institutions to gather feedback, iron out any potential issues, and showcase successful case studies. Collect testimonials and success stories from early adopters to build credibility and demonstrate the system's impact on attendance management and student outcomes.
* ***Market Research and Validation***: Before fully commercializing the QR-code-based attendance tracking system, conduct comprehensive market research to understand the target audience's needs, preferences, and pain points. Validate the demand for such a system in educational institutions, identifying potential customers, such as schools, colleges, universities, and corporate training centers.

# REFERENCES

*1. Documentation for Visual Studio Code. (n.d.). Documentation for Visual Studio Code. Retrieved September 29, 2022, from* [*https://code.visualstudio.com/docs*](https://code.visualstudio.com/docs)

*2. Software Processes - javatpoint. (n.d.). www.javatpoint.com. Retrieved September 29, 2022, from* [*https://www.javatpoint.com/software-processes*](https://www.javatpoint.com/software-processes)

*3. Software Processes in Software Engineering - GeeksforGeeks. (2019, September 4). GeeksforGeeks. Retrieved September 29, 2022, from* [*https://www.geeksforgeeks.org/software-processes-in-software-engineering/*](https://www.geeksforgeeks.org/software-processes-in-software-engineering/)

*4. Project Management Methodologies - Everything You Need to Know. (n.d.). Project Management Methodologies - Everything You Need to Know. Retrieved September 29, 2022, from* [*https://www.teamwork.com/project-management-guide/project-management-methodologies/*](https://www.teamwork.com/project-management-guide/project-management-methodologies/)

*5. Garousi, V., Borg, M. & Oivo, M. Practical relevance of software engineering research: synthesizing the community’s voice. Empir Software Eng* ***25****, 1687–1754 (2020).* [*https://doi.org/10.1007/s10664-020-09803-0/*](https://doi.org/10.1007/s10664-020-09803-0/)