



**COLLEGE OF BUSINESS AND ECONOMICS**

**SCHOOL OF BUSINESS**

**BIT DEPARTMENT**

**LEVEL 3**

**GROUP 1, Sub Group 6**

**GROUP MEMBERS**

<b>NAMES</b>	<b>REGISTRATION NUMBER</b>
<b>Alexis HAKIZIMANA</b>	<b>221013271</b>
<b>Hugues DENOTRADAME</b>	<b>221022786</b>
<b>Kevine Gatesi HAKIZIMANA</b>	<b>221001925</b>
<b>Boniphile IHIMBAZWE</b>	<b>221025205</b>
<b>Didacienne IKUZWE BYISHIMO</b>	<b>221014640</b>
<b>Betty HIRWA IGIRANEZA</b>	<b>221007588</b>
<b>Marie Merci Carine INGABIRE</b>	<b>221022884</b>

**COURSE: PROGRAMMING WITH CLIENT/SERVER TECHNOLOGY**

**CODE: BIT 3134**

**PROJECT PROPOSAL**

**ON**

**FOOD STORE MANAGEMENT SYSTEM**

**Submitted to: Lecturer Benjamin YABAGENGA**

**Submission date: 10-september-2023**

## **1. INTRODUCTION:**

### **1.1 Overview**

The Student Attendance Management System (SAMS) is a software application developed in Microsoft Visual Studio and integrated with a SQL Server database. It aims to simplify and streamline the process of tracking and managing student attendance in educational institutions. This documentation provides a comprehensive understanding of the SAMS project, its objectives, and its significance in modern education.

### **1.2 Objectives**

The primary objectives of the SAMS project are:

1. To automate the attendance recording process for teachers and students.
2. To provide real-time attendance data to Lectures and Students.
3. To reduce Lecture workload and improve efficiency in attendance management.

## **2. PROJECT SETUP AND DEVELOPMENT ENVIRONMENT**

### **2.1 Development Tools**

The Student Attendance Management System (SAMS) project was developed using a set of essential development tools:

#### **1. Microsoft Visual Studio: Version 2022**

Visual Studio was chosen as the primary Integrated Development Environment (IDE) for its robust features, excellent code editor, and seamless integration with .NET technologies.

#### **2. SQL Server Management Studio (SSMS):**

SSMS played a crucial role in managing the SQL Server database used by SAMS. It provided a user-friendly interface for database design, querying, and administration.

#### **3. .NET Framework: Version 4.8.1**

The project relies on the .NET Framework for building Windows applications. .NET Framework provides a stable and well-supported platform for developing desktop applications like SAMS.

#### **4. VB.NET Programming Language**

VB.NET was chosen as the primary programming language for its compatibility with .NET, strong typing, and extensive libraries. It's well-suited for building reliable and efficient Windows applications.

### **2.2 Project Setup**

**The setup process for the SAMS project involved several key steps:**

**Creating the Visual Studio Solution:**

We initiated the project by creating a new solution in Visual Studio, selecting the appropriate project template for a Windows Forms Application. This template provided the foundation for our desktop application.

**Database Integration with SQL Server:**

To store and manage attendance records and other data, we integrated SQL Server into our project. SQL Server provides a robust and scalable database management system that aligns well with .NET applications. We configured the necessary database connections within Visual Studio.

**Continuous Integration and Deployment (CI/CD):**

We established a CI/CD pipeline to automate the build, testing, and deployment processes. Tools like Jenkins or Azure DevOps were used to streamline these operations, ensuring that changes were regularly tested and deployed to the appropriate environments.

**Security Measures:**

Security was a top priority in our project setup. We implemented measures to secure the SQL Server database, including proper user authentication and authorization. Additionally, Visual Studio's built-in security features, such as code analysis, helped identify and address potential vulnerabilities in the codebase.

**Documentation and Knowledge Sharing:**

We created project documentation and shared it with the development team. This documentation included design documents, coding standards, and guidelines for collaboration. Knowledge sharing sessions were conducted to ensure that team members were proficient with the chosen tools and technologies.

**Testing and Debugging Tools:**

Visual Studio provides a suite of testing and debugging tools that greatly assisted in the development process. We configured unit tests using Visual Studio's built-in testing framework and used debugging features to identify and resolve issues in the code.

### Development Environment Reproducibility:

To ensure consistency across development, testing, and production environments, we adopted containerization technologies like Docker. This allowed us to package the application and its dependencies into containers, making it easier to reproduce the environment on different systems.

## 3. System Design in Microsoft Visual Studio and SQL Server

### 3.1 Visual Studio Solution Structure

The structure of the Student Attendance Management System (SAMS) project within Visual Studio was thoughtfully organized to enhance code maintainability and modularity. The solution consisted of multiple projects:

**SAMS.WindowsApp:** This project served as the main user interface for the application, built using Windows Forms. It contained forms for Lecture registration, Student Registration, Login, Course register/update/delete by Lecture, Enrollment key creation by lecture, attendance recording, and report generation.

**SAMS.Data:** The data access layer, responsible for communication with the SQL Server database. It included Entity Framework Core classes for mapping database tables to VB.NET objects, as well as repositories for data retrieval and manipulation.

**SAMS.Models:** A separate project for defining data models, including entities like Student, Teacher, and AttendanceRecord. These models were used across various layers of the application.

**SAMS.Services:** This project encapsulated the application's business logic and service layer. It handled tasks such as attendance calculations, report generation, and user authentication.

**SAMS.Tests:** A unit testing project where we created test cases to validate the functionality of critical components, including data access methods and business logic.

**SAMS.Utility:** A utility project containing shared helper classes and functions used throughout the application, such as date formatting and error handling. The solution structure promoted a clean separation of concerns and facilitated team collaboration by allowing developers to work on specific components independently.

The user interface (UI) design in SAMS aimed for simplicity and user-friendliness while ensuring efficient attendance management:

**Windows Forms:** Visual Studio's Windows Forms Application template was employed to create the desktop application's UI. This choice allowed us to build responsive and visually appealing forms.

**Attendance Recording Form:** The attendance recording form provided teachers with an intuitive interface to mark student attendance. It displayed class details, student lists, and date selection controls. Real-time feedback and error handling were integrated to enhance usability.

**Code-Behind and Event Handling:** The code-behind files for each form contained event handlers that responded to user actions. For example, clicking the "Record Attendance" button triggered the attendance recording logic, which interacted with the data access layer to update records in the SQL Server database.

**Data Binding:** Visual Studio's data binding features were employed to connect UI elements, such as DataGridViews, to data sources. This facilitated the display and manipulation of student and attendance data.

**Testing and Debugging:** Visual Studio's debugging tools were essential during the development phase. We used the Visual Studio Debugger to step through code, inspect variables, and identify and fix issues in real-time.

**Accessibility:** Accessibility features were implemented to ensure that the application could be used by individuals with disabilities. This included providing alternative text for images and using keyboard-friendly controls.

**Localization:** The application was designed with localization in mind, allowing for easy translation into multiple languages if needed. Resource files were used to store localized strings.

## 4. Implementation

### 4.1 Code Structure

The implementation of the Student Attendance Management System (SAMS) project in VB.NET followed a well-organized structure designed to enhance code readability, maintainability, and modularity. Key aspects of the code structure include:

**Solution Layout:** The project was organized within a Visual Studio solution consisting of multiple projects. These projects were categorized based on functionality and included:

**SAMS.WindowsApp:** The primary project responsible for the application's user interface.  
**SAMS.Data:** The data access layer handling interactions with the SQL Server database.

**SAMS.Models:** A project containing data model classes representing entities like students, teachers, and attendance records.

**SAMS.Services:** The business logic and service layer, containing functions for attendance calculations, report generation, and user authentication.

**SAMS.Utility:** A shared utility project housing helper classes and functions used throughout the application.

**Separation of Concerns:** We implemented a clear separation of concerns, ensuring that each component had a well-defined responsibility. For example, data access was isolated in the SAMS.Data project, and business logic resided in SAMS.Services.

**Modular Code:** The codebase was designed to be modular, with classes and functions organized logically. This structure allowed for easy maintenance and expansion as the project evolved.

**Error Handling:** Robust error handling mechanisms were implemented throughout the codebase to gracefully handle exceptions and provide meaningful error messages to users.

## 4.2 Database Interactions

VB.NET, in combination with Entity Framework or ADO.NET, allowed for seamless interactions with the SQL Server database. Key aspects of database interactions in the project include:

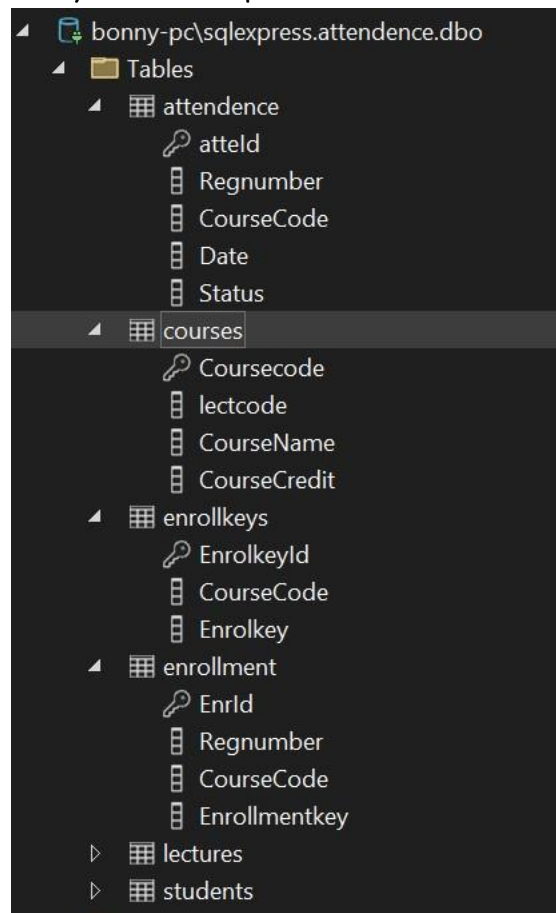
**Entity Framework Usage:** Entity Framework was used for data access, providing an Object-Relational Mapping (ORM) approach to interact with the database. It simplified tasks such as CRUD (Create, Read, Update, Delete) operations.

**Connection String Management:** Connection strings to the SQL Server database were stored securely in configuration files, ensuring that sensitive database credentials were not exposed in the source code.

**Entity Framework Migrations:** Entity Framework migrations were employed to manage database schema changes as the project evolved. This approach allowed for efficient database schema updates without manual intervention.

**Transaction Handling:** To ensure data consistency, transactions were used when multiple database operations needed to be performed as a single unit of work. This approach-maintained data integrity in critical scenarios.

#### Entity Relation Ship:



### 4.3 User Authentication and Role-Based Access

User authentication and role-based access control were key components of the implementation:

Authentication: Users were required to authenticate using their credentials (username and password) before accessing the system. VB.NET code verified user identities and managed login sessions.

Role-Based Access Control (RBAC): RBAC was implemented to assign specific roles (e.g., teacher, administrator, parent) to users. Access to certain features and data was restricted based on user roles.

**User Session Management:** VB.NET code-maintained user sessions and ensured that users remained authenticated during their interactions with the application.

## 5. Features and Functionality

### 5.1 Teacher Features

The details of how lecture uses this system is as follow.

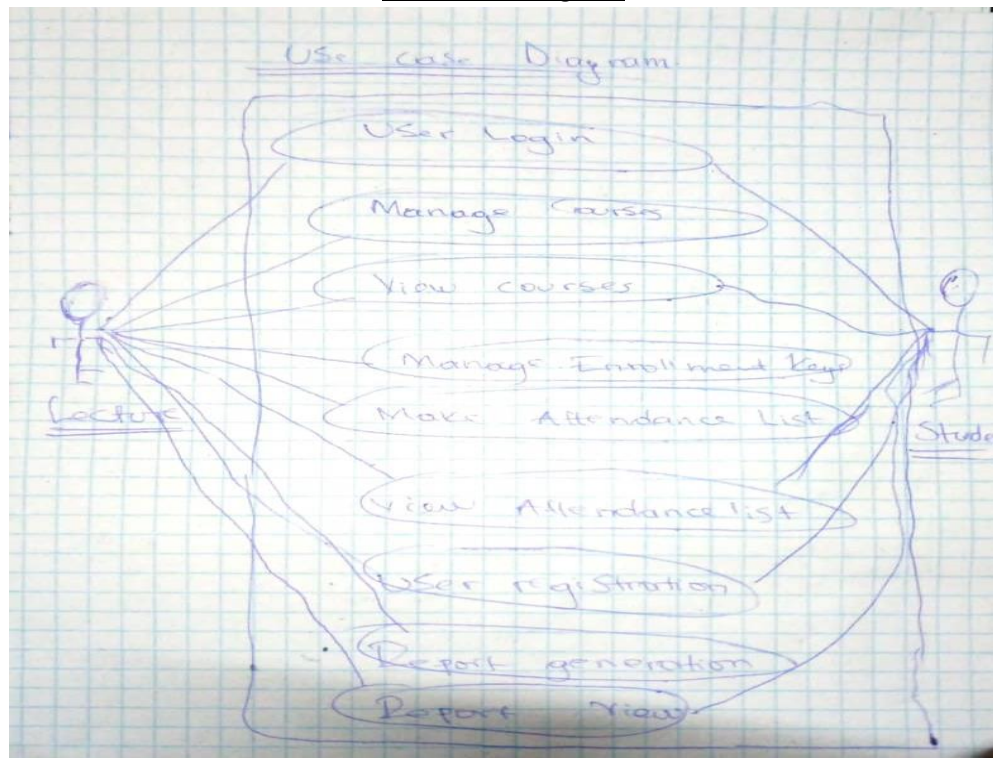
1. He or she has to register first: at the start-up of application Login form is loaded first, so lecture has to load registration page via the register as **Lecture link** and then login using registered username and password.
2. He or she also will have to register, view, update, and delete courses.
3. He or she will create course enrollment key and update or delete them.
4. He or she also will register courses, view them, update, and delete them.
5. He or she also will make attendance for students and generate attendance report.

### 5.2 Student Feature

1. Students will have the access to register to the system via the link register as **Student**.
2. Students also will have to login using registered username and password.
3. Students will have access to registered courses by viewing them only. No updating or deleting them.
4. Students also will have to enroll course using enrollment key given by their lecture.
5. He or she also will have access to the attendance report by viewing only. Note: Student will have option to search by his or her registration number, course code or dates depending on what he or she want to see.



## User Case Diagram



## 6. Testing and Quality Assurance

### 6.1 Testing Methodologies

**Integration Testing:** Integration tests were performed to validate the interactions between different components of the system. This ensured that data flowed seamlessly between the user interface, business logic, and data access layers. Integration tests covered scenarios such as attendance recording, report generation, and user authentication.

**User Acceptance Testing (UAT):** Prior to deployment, the system underwent UAT with end-users, including teachers, administrators, and parents. UAT aimed to validate that the system met user requirements and expectations. Feedback from UAT sessions was used to make final adjustments and refinements.

Load and Performance Testing: Load testing tools, such as Visual Studio Load Testing, were used to assess the system's performance under heavy loads. This testing ensured that SAMS could handle concurrent user requests without degrading performance.

## 6.2 Error Handling and Logging

Effective error handling and logging mechanisms were implemented to monitor and respond to issues:

**Exception Handling:** Exception handling was integrated throughout the codebase to gracefully handle unexpected errors and exceptions. Detailed error messages were provided to assist in troubleshooting.

**Centralized Logging:** A centralized logging system was employed to record application events, errors, and security-related incidents. Logs were stored securely and were accessible to authorized personnel for analysis and auditing purposes.

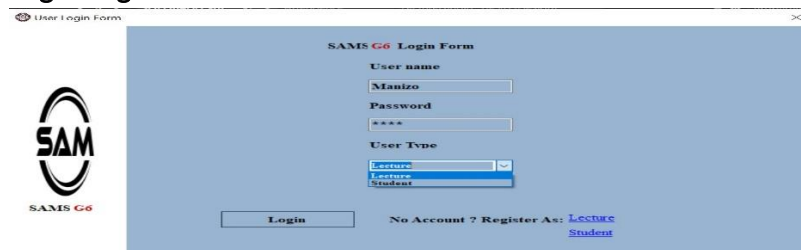
**Alerts and Notifications:** Automated alerts and notifications were set up to inform administrators and support teams about critical issues, ensuring rapid response and issue resolution.

## 7. User Documentation:

First let us describe how lecture will use this application. We started by lecture side because he is the key point for the application to complete intended activities. This means that without courses registration and enrollment key creation, student will not join the course and view his or her attendances.

Here are some of description of how lecture will use the application:

### 1. Login Page



The screenshot shows a web browser window titled "User Login Form". On the left is the SAMS G6 logo. The main content area is titled "SAMS G6 Login Form" and contains the following fields and options:

- User name:** A text input field containing the text "Manizo".
- Password:** A text input field containing four asterisks "\*\*\*\*".
- User Type:** A dropdown menu with "Lecture" selected. Other visible options are "Lecture" and "Student".
- Login:** A button at the bottom left.
- No Account ? Register As:** A link at the bottom right with "Lecture" and "Student" as sub-links.

- Both lecture and student will access this page for the first time. Once they do not have registered themselves, lecture will register via [Register as lecture link.](#) While once student wants to register will use [Register as Student link.](#)
- Once both have already registered, they can fill all fields and select user type and click on login button.
- After clicking on login button system will redirect them based on their user types to their landing pages.

## 2. Lecture Registration Page

The screenshot shows a web browser window titled 'Registration Form'. On the left is the SAMS G6 logo. The main content area is a blue form titled 'SAMS G6 Lecture Registration Form'. It has two columns of input fields: 'Lecture Code', 'Full Name', and 'Degree' on the left; 'Email', 'User Name', and 'Password' on the right. At the bottom, there is a 'Register' button and a link that says 'Already Have Account? Back To Login'.

- Here lecture will fill out all fields and click on register button to save his or her registration data. **Note:** he has to take in mind his or her username and password because will be used during login process.
- Also here, there is a link that will redirect him or her to login page when he do not want to register.

## 3. Lecture Landing Page

The screenshot shows a web browser window titled 'Lecture Landing Page'. The top navigation bar includes 'SAMS G6', 'Home', 'Students', 'Enrollments', 'Attendance', 'Reports', and 'Logout'. The main content area is divided into two sections. On the left, under 'Group Members', is a table with columns '#', 'Names', and 'Reg Number'. It lists seven members. On the right, under 'Add a Course', is a form with fields for 'Course Code', 'Course Name', 'Course Lecture', and 'Course Credit'. Below these fields are 'Add', 'Update', and 'Delete' buttons. Underneath the form is a 'List Of Courses' table with columns 'Coursecode', 'lectcode', 'CourseName', and 'CourseCredit'. It shows one course: 'Java020', 'Via012', 'Java Programming', and '7 Credits'. A 'Refresh' button is at the bottom of the list.

#	Names	Reg Number
1	Alexis HAKIZIMANA	221013271
2	Kevine Gatesi HAKIZIMANA	221001925
3	Boniphile IHIMBAZWE	221025205
4	Didacienne IKUZWE BYISHIMC	221014640
5	Betty HIRWA IGIRANEZA	221007588
6	Denotradame HYGUE	221022786
7	Marie Merci Carine INGABIRE	221022884

Coursecode	lectcode	CourseName	CourseCredit
Java020	Via012	Java Programming	7 Credits

- Now, login processes have done successfully. On this page, lecture can add, update, and delete course
- Also, he can refresh the page via refresh button.
- He has to use Student, enrollment, attendance, and report menus to carry out their related activities.
- Once he or she click on logout menu, the system will redirect him or her to the login page.

#### 4. Lecture Student Page

Regnumber	Fullname	Gender	Studyyear	Username	Password
221013271	Alexis HAKIZIMA...	Male	Year 3	Alexis	@221013271

- Here, lecture can Register student when it is needed for students who are not able to register his or herself.
- He will also view, update, and delete registered students
- For update, he or her will update student information using student registration number.
- For delete, he or she will just enter registration number for the student to be deleted.

#### 5. Lecture Enrollment Page

EnrolkeyId	CourseCode	Enrolke
4	Java020	JAVA_2

CourseCode	CourseName	Regnur
Java020	Java Programming	221013

- Firstly, lecture has to create enrollment key for his or her registered course.
- He can also update, and delete created enrollment key by using Course code to determine which course for enrollment key to be deleted
- Also on left side, lecture will have to view, delete, and search by student registration number for enrolled students

#### 6. Lecture Attendance Page

- Now, lecture can make, update, and delete attendance for enrolled students using right side.
  - Also, he can make search for attendance using left side by entering course code and student registration number.
  - also, he can refresh the page after search using refresh button.
7. Lecture Attendance Report View

- For Attendance report, Lecture can search for list of attendance using course code, and date between based on his or her desire.
- After search, he or she can refresh the page to view all data via refresh button.

Then now, let us take into account for the side of student. Here are some of direction on how student will use this application:

## 1. Login Page

- As we said, both students and lecture will use this form to login, so to login as student, you have to use your registered username and password and select user type which have to be student in this case.

## 2. Student Landing Page

StudentLandingPage

SAMS G6

Home Enrollments Attendance Logout

Search With Course Name

alex Search

List Of Courses Refresh List

Coursecode	lectcode	CourseName	CourseCredit
Java020	Via012	Java Programming	7 Credits

Please Keep Course Code To Enroll The Course!!!

- Here student has to search course name he or she wants to enroll and copy its course code and move on to the enrollment menu.
- He or she also has to refresh the list after search complete

### 3. Student Enrollment Page

StudentEnrollment

SAMS G6

Home Enrollments Attendance Logout

Search With Reg Number

Search

Reg Number

Course Code

Enrollment Key

Enroll Leav

List Of Enrolled Students Refresh List

Enrld	Regnumber	CourseCode	Enrollmentkey
5	221013271	Java020	JAVA_2023-2024

- Now, student will fill out all fields, his registration number, course code which he or she has copied on landing page, and enrollment key which has to be given by lecture.
- He or she also can leave the course he has enrolled by just entering Registration number and course code.

### 4. Student Attendance View Page

StudentAttendanceView

SAMS G6

Home Enrollments Attendance Logout

Search For Attendance Info

Course Code Reg Number

Search

Attendance List Refresh

CourseCode	CourseName	Regnumber	Fullname	Date	Status
Java020	Java Programming	221013271	Alexis HAKIZIMA...	09/09/2023	Yes

- Now, student will search his or her attendance just by entering course code and registration number
- After search complete also, student can refresh the list to view all list

### 8. Conclusion:

This system, built using Microsoft Visual Studio and integrated with a SQL Server database, will transformed the way educational institutions track and manage

student attendance. As we conclude this documentation, it is essential to reflect on the key accomplishments, the impact of SAMS, and the path forward.

### 8.1 Achievements:

- a. **Automation of Attendance Recording:** SAMS has successfully automated the attendance recording process, alleviating the burden on educators and administrators. With just a few clicks, teachers can mark attendance, saving valuable instructional time.
- b. **Real-Time Data Access:** Administrators and parents now have instant access to real-time attendance data through the intuitive user interface. This has empowered informed decision-making and facilitated timely interventions.
- c. **Efficiency and Accuracy:** By reducing manual record-keeping and the potential for human error, SAMS has significantly improved the efficiency and accuracy of attendance management. This has led to streamlined administrative processes and reduced administrative overhead.
- d. **User-Friendly Interface:** The user-friendly interface of SAMS, designed with responsiveness and accessibility in mind, has been well-received by teachers, students, parents, and administrators alike. It ensures that the system is accessible from various devices and meets the needs of a diverse user base.

### 8.2 Impact:

- a. **Enhanced Educational Experience:** SAMS has improved the educational experience by promoting regular attendance and engagement. Teachers can readily identify patterns of absenteeism and take proactive measures to support students.
- b. **Data-Driven Decision-Making:** Lecturer and student now have access to data-driven insights that inform policy decisions, resource allocation, and study strategies. Real-time attendance data has become a valuable tool for performance management.

**In conclusion,** the Student Attendance Management System (SAMS) stands as a testament to innovation and efficiency in educational technology. It has revolutionized attendance management in educational institutions, promoting regular attendance, data-driven decision-making, and enhanced engagement.