

BULE HORA UNIVERSITY

COLLEGE OF INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE

CLASS SCHEDULE AND COURSE DISTRIBUTION SYTEM FOR INFORMATICS COLLEGE

SUBMITTED TO DEPARTMENT OF SOFTWARE ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

Name Id

1. Ephrem Hirko……………………………………………………….…Ru1360/12
2. Barie Wakjira…………………………………………………………..Ru1351/12
3. Seblewongel Tsegaye…………………………………………....……..RU0181/12
4. Mamaru Alehegne……………………………………………………...Ru1851/12 Project AdvisorS: Mr. Getahun .F

Bule Hora University, Ethiopia

February 12, 2024

# DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Getahun F**.** , Mr. Gemada G. and Mr. Abinat and having the title Class Schedule and course distribution system is the sole contribution of Efrem Hirko, Barie Wakjira Seblewongel Tsegahe and Mamaru Alahegn.

No part of the project work has been reproduced illegally (copy and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if the violation of this declaration is proven.

Thursday, January 11, 2024

**Group Members: -**

**No Full name Signature**

1. Efrem Hirko \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Barie Wakjira \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Seblewongel Tsegahe \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Mamaru Alehegne \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# APPROVAL FORM

This is to confirm that the project report entitled Class schedule and course distribution system submitted to Bule Hora University, College of Informatics and Department of Software Engineering by Efrem Hirko, Barie Wakjira Seblewongel Tsagahe and Mamaru Alehegne to approved for submission.

Advisor name Signature Date

---------------------------- ---------------------------- ----------------------------

Department head name Signature Date

---------------------------- ---------------------------- ----------------------------

Examiner 1 name Signature Date

---------------------------- ---------------------------- ----------------------------

Examiner 2 name Signature Date

---------------------------- ---------------------------- ----------------------------

Examiner 3 name Signature Date

---------------------------- ---------------------------- ----------------------------

# Acknowledgement

We extend our gratitude to the Almighty God for continual guidance and blessings throughout our life journey and the completion of our education. Our heartfelt appreciation goes to our families for their unwavering support and belief in our abilities, serving as a constant source of motivation. Special thanks to Mr. Salamu Shirtawi, our former department head, and Mr. Ababe Bekele, our current department head, for their leadership and commitment to our academic development. We also acknowledge with gratitude the invaluable contributions of our project advisors, Mr. Getahun F., Mr. Gemada G., and Mr. Abinat, whose expertise and guidance have shaped the trajectory of our project. We appreciate the faculty and staff of the Software Engineering Department at Bule Hora University for fostering an environment of academic excellence. Our thanks extend to friends and colleagues for their encouragement. Generally, this project's success is attributed to the collaborative efforts and support of everyone mentioned, and each contribution has played a significant role in our academic and project journey. Thank you.

# EXECUTIVE SUMMARY

The proposed "Class Schedule and Course Distribution System for Informatics College" aims to address significant challenges faced by the Software Engineering Department at Bule Hora University related to manual class scheduling and course distribution. The current procedures have resulted in inefficient scheduling, limited accessibility to real-time information, and difficulties in course distribution management. Motivated by these challenges, our goal is to design and implement a unified web-based system that automates class schedules, optimizes course distribution, and contributes to the department's operational efficiency. The project's significance lies in its potential to streamline processes, prevent scheduling conflicts, balance workloads, and provide real-time visibility. The feasibility study indicates positive economic, technical, and operational feasibility. The proposed system's scope includes user-friendly registration interfaces, automatic class schedule generation, course distribution automation, and reporting features. The methodology follows Agile principles, utilizing technologies like Java, Spring Boot, Angular, and MySQL. The document details the existing system analysis, proposed system design, and system implementation. The success of the project is attributed to the collaborative efforts and support of various stakeholders, including our families, department heads, project advisors, and the entire faculty at the Software Engineering Department.

# Chapter One: Introduction

* 1. **Introduction**

In the lively world of Bule Hora University's College of Informatics, the rise of computers has brought many technical improvements, especially in education. Computers are everywhere, making teaching and learning easier. However, there is a challenge with how we organize classes and distribute courses in the College of Informatics. The current manual methods slow things down. Imagine how important a good system is for keeping track of student information during registration. Well, having an equally good system for managing courses and schedules is crucial. That is where our project comes in. We are creating a user-friendly web system, the Class Schedule and Course Distribution System (CSCDS), designed specifically for the College of Informatics at Bule Hora University.

This system will make things easier by automating the process of assigning courses to teachers and making sure it is fair and follows all the rules. The aim is to modernize how things work, save time, and reduce mistakes compared to the old-fashioned manual ways. Our system's main goal is to speed up how courses are assigned, make scheduling classes simple and overall, and make academic life better in the College of Informatics. With this project, we hope to contribute to the College of Informatics at Bule Hora University by bringing in better technology and improving how things are done.

* 1. **Organization Background**

Bule Hora University's College of Informatics stands as a cornerstone of excellence since its establishment. The objective of the college is to produce competent professionals in computer science, software engineering, IT, system analysis, and information science. The college offers a variety of undergraduate programs, including Computer Science, Information Technology, Software Engineering, Information Systems, and Information Science. Additionally, it provides Master's programs in Computer Science and Information Technology.

Our proposed CSCDS system aligns with the college's mission of equipping students with the necessary skills and knowledge to address individual and organizational challenges. By streamlining class schedules and course distribution, the system contributes to the college's objective of fostering growth and producing highly skilled graduates. Through collaboration with the college's departments, our project aims to enhance the academic experience and uphold the standards of excellence set by Bule Hora University's College of Informatics.

* 1. **Statement of the Problem**

The current manual procedures governing class scheduling and course distribution within our educational framework pose multifaceted challenges, including:

* **Inefficient Scheduling**: The manual creation of class schedules is time-consuming, error-prone, and frequently results in conflicts within the schedule.
* **Limited Accessibility:** Both students and educators encounter difficulties accessing real-time and accurate class schedule and course distribution information, leading to frequent miscommunications and disruptions in the learning process.
* **Course Distribution Challenges:** The absence of an automated mechanism for course distribution makes it challenging to effectively manage assignments and ensure an equitable distribution of workload among teachers.

In addressing these challenges, our proposed solution, the Class Schedule and Course Distribution System (CSCDSFIC), aims to streamline and modernize these processes, mitigating the inefficiencies posed by the current manual system.

* 1. **Objectives**
     1. **General Objective**

Develop an automated **Class Schedule and Course Distribution System (CSCDS)** for the Informatics College at Bule Hora University to enhance operational efficiency, improve transparency, and contribute to the overall academic excellence.

* + 1. **Specific Objectives**
* Gather information about the existing manual system.
* Analyze the gathered information and model the requirements for the proposed system.
* Design the proposed system based on the analyzed information and requirements.
* Develop a comprehensive system to automate class schedules, considering lecture hours, laboratory hours, and credit hours.
* Test and validate the system to ensure its functionality and effectiveness.
* Implement an automated system for course distribution.

By achieving these specific objectives within a single, integrated system, we aim to enhance the overall operational efficiency of the Departments within the informatics college, providing a cohesive solution to the identified challenges.

* 1. **Methodology**
     1. **Requirement Gathering Methods**

The Agile methodology, coupled with advanced technologies and hardware tools, will guide our approach to gathering requirements effectively for the Class Schedule and Course Distribution System (CSCDSFIC). We plan to utilize the following Agile-compatible methods:

* **Interviews:** Conduct one-on-one interviews with faculty, administrators, and students using collaborative platforms to gather in-depth insights into expectations and needs. Identify pain points and elicit specific system requirements.
* **Observations:** Utilize agile principles to observe existing manual processes directly. This approach ensures a firsthand understanding of workflows, allowing for more accurate requirement gathering.
* **Focus Group Discussions:** Organize Agile-focused focus group discussions using collaboration platforms with representatives from faculty and student groups. Facilitate interactive sessions for open dialogue and collaboration, fostering a collective understanding of system requirements.
  + 1. **Requirement Modeling (Agile-Oriented) with Integrated Technologies**

For Agile system modeling, we propose the utilization of the Object-Oriented (OO) approach with agile principles in mind. This will be seamlessly integrated with the specified technologies:

* **Complex Relationships:** Object-oriented modeling aligns with Agile by representing complex relationships naturally through classes and objects, implemented using Java and Spring Boot.
* **Modularity and Reusability:** The Agile-Oriented OO approach will leverage technologies such as Angular for front-end development, promoting modularity and code reusability.
* **User-Centric Design:** Object-oriented modeling resonates with Agile's user-centric design philosophy, ensuring an intuitive representation of system entities. This will be reflected in the HTML, CSS, and JavaScript used for the user interface.
* **Scalability:** The OO approach aligns with Agile scalability principles, allowing seamless integration of new features without disrupting existing components. This scalability will be supported by technologies like MySQL for database management.
  + 1. **Tools Used (Agile-Integrated) with Collaborative Technologies**

To support the agile development process and ensure effective collaboration, we will integrate the specified technologies:

* **Visual-paradigm (UML Tool):** to create Unified Modeling Language (UML) diagrams we will use tools like visual paradigm that ensuring an Agile-oriented representation of the system's structure.
* **Collaboration Platforms:** We will use **GitHub** for collaboration platforms that enable us co-working and getting iterative feedback sessions, integrated seamlessly with Git for version control.
* **Document Management Systems:** we will use Microsoft 2016 Docs for document organization of gathered information and writing the documentation of the project.
* **Frontend Development:** we will utilize JavaScript, HTML, and CSS for frontend development, following agile principles for a responsive and user-friendly interface.
* **Backend development:** we will utilize java programming languages for the algorithms.
* **Integrated Development Environments (IDEs):** We will utilize NetBeans IDEA or Eclipse for the development environment, ensuring an Agile-oriented and efficient coding experience.
* **Database:** we will use MySQL database software for storing and retrieving data.

**Implementation Environment (Hardware Tools)**

For the development and deployment of the CSAMCDS, the following hardware tools will be use

* **Computers:** Standard computers equipped with the necessary processing power and storage to support software development.
* **Flash Drives:** Portable storage devices for data backup, sharing, and transferring project-related files between team members.

By embracing Agile methodologies and seamlessly integrating these advanced technologies and hardware tools, we aim to ensure a dynamic, iterative, and efficient approach to requirement gathering, system modeling, and collaborative development, contributing to the success of the Agile-oriented CSCDSFIC project.

* 1. **Feasibility**
     1. **Economic Feasibility**

The implementation of the unified Class Schedule and Course Distribution System for the Departments within the college is economically feasible and presents significant potential benefits. Automating critical processes is expected to yield cost savings, minimize errors, and optimize resource utilization. While initial investments are required, the long-term economic gains and enhanced operational efficiency justify the project's feasibility. The system's streamlined efficiency is anticipated to contribute positively to the academic experience, attracting more students and fostering a conducive learning environment.

* + 1. **Technical Feasibility**

The technical feasibility of the project is robust, leveraging our acquired skills during the five-year software engineering journey. Proficiency in web development, database management, and system integration aligns with the project's requirements. The selected technologies (Java, Spring Boot, MySQL, HTML, CSS, and JavaScript) match our capabilities, and additional skills can be acquired during development. Technical support and learning resources are readily available, ensuring the project remains well within our capabilities.

* + 1. **Operational Feasibility**

The operational feasibility of implementing the unified system in the Departments within the informatics college is grounded in practicality and adaptability to the department's existing environment. The web-based nature facilitates seamless integration into daily operations with minimal disruptions. The user-friendly design ensures a smooth transition for faculty and students without extensive training. Streamlined processes, efficient resource allocation, and improved academic experiences make the operational integration highly practical, aligning with the department's goal of delivering high-quality education.

* + 1. **Time Feasibility**

The project's time feasibility is favorable, given the available timeframe for development. The skills and resources required are well within our capabilities, and the timeline aligns with the project's scope. A realistic schedule, coupled with efficient project management, ensures that the Class Schedule and Course Distribution System can be developed and implemented within the specified period.

* 1. **Project Scope**

The proposed Class and Course Distribution System for informatics college (CSCDSFIC) aims to cover comprehensive functionalities tailored to the specific needs of the Departments within the informatics college. The scope of the project encompasses the following key features:

**Teachers, Students, and Courses Registration**

* User-friendly registration interface for departments to register teachers, students, and courses.
* Teachers empowered to select courses based on their expertise, availability, and interest.
* Recording of teachers' qualifications, experience, workload, and availability within the system.
* Streamlined registration process for students to enroll in courses online.

**Class Schedule Generation**

* Automatic generation of class schedules based on course outline information, considering lecture hours, laboratory hours, and credit hours after course is distributed to the teachers availability of class room also should be considered.
* Manual adjustment capability for special cases or preferences.

**Course Distribution**

* Web-based automation of the course distribution process, accounting for teachers' experience, education level, interests, and field of study.
* Automatic generation of evaluation results based on predefined criteria, leading to the assignment of courses to teachers.
* Approval workflow involving department heads for final assessment and approval.

**Constraints and Rules**

* **Maximum Course Limit:** Teachers limited to a maximum of three courses per semester for a balanced workload.
* **Department Head/Coordinator Limit:** Teachers in administrative roles restricted to a maximum of 6 hours per week for teaching.
* **Batch Limitation:** Teachers cannot teach two courses in one batch in a semester to promote fairness and diversity.
* **HDP Program:** Teachers in the Higher Diploma Program limited to one course with reduced workload.
* **Student Advising:** Specific credit hour reductions for teachers advising bachelor's degree students to encourage student advising.
* **Maximum Credit Hour Limit:** Enforced limit of 12 hours per week for each teacher, with overload requiring additional approval.
* **Overload Compensation:** System calculates and tracks overload hours for compensation if a teacher exceeds the maximum credit hour limit.

**Calculation and Reporting:**

* The CSCDSFIC will dynamically calculate the number of courses and total credit hours assigned to each teacher.
  1. **Significance of the Project**

The proposed CSCDSFIC will play a pivotal role in elevating the Departments within the Informatics College, benefiting faculty members, and students. The project's significance encompasses various dimensions, as highlighted below:

1. **Enhanced Operational Efficiency**
   * The CSCDSFIC will streamline complex processes, such as class scheduling and course distribution, reducing manual errors and time-consuming tasks.
   * Automation of routine administrative tasks will free up faculty members' time for core teaching and research activities, ultimately enhancing departmental efficiency.
2. **Optimal Resource Utilization**
   * Automated class schedules and course distribution will ensure optimal use of teaching resources, preventing conflicts and promoting a balanced workload for faculty members.
   * Automation features will contribute to resource optimization, allowing the department to allocate teaching assignments based on expertise, interests, and availability.
3. **Improved User Satisfaction**
   * Faculty members will benefit from an intuitive and user-friendly system, empowering them to select courses aligned with their expertise, interests, and availability.
   * Students will enjoy streamlined course registration processes, access to real-time schedules, and enhanced communication channels within the system.
4. **Equitable Workload Distribution**
   * The CSCDSFIC incorporates rules and constraints for fair and equitable distribution of teaching assignments among faculty members.
   * Constraints, such as maximum course limits and workload considerations for administrative roles, contribute to a balanced workload, fostering a conducive teaching and learning environment.
5. **Real-Time Visibility and Monitoring**
   * The digitalized attendance system will provide real-time visibility into teaching and learning progress, enabling efficient attendance monitoring.
   * Department heads will have instant access to course distribution status, allowing timely review, evaluation, and approval of assignments.
6. **Empowering Decision-Making**
   * The CSCDSFIC will equip department heads and administrators with data-driven insights, facilitating informed decision-making for course distribution, faculty workload management, and resource allocation.
   * Automated evaluation processes will provide valuable information for decision-makers to optimize teaching assignments based on various criteria.
7. **Academic Excellence and Reputation**
   * Successful implementation of the project will contribute to the academic excellence of the Departments within the informatics college, attracting prospective students and faculty.
   * Improved operational efficiency, equitable workload distribution, and enhanced user satisfaction will positively influence the departments’ reputation, positioning it as a center of excellence in informatics college.

**Chapter Two: System Analysis**

1. **Introduction** 
   1. **Existing System**
      1. **Description Existing System**

The existing system description describes the current system of the organization as it is. This could be describing the activities they perform, how they handle information, the drawbacks of the system and users of the existing system

* + 1. **Business Rules**

Write the rules used by the organization currently. In the online banking case, this could be the interest rate allowed for saving account, the maximum amount of money that someone can withdraw at a time, interest rate for loan, etc. Generally, they are the rules by which the business is governed.

* 1. **New System (Proposed System)**
     1. **Functional Requirements**

Capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform or System that support user goals, tasks or activities. It describes the functional requirements of the system. What functionalities does the system have? Then proceed to show it using use case diagram under the next topic use case diagram.

* + 1. **Non-functional Requirements and Constraints**

Non-functional requirements are constraints and qualities of the system or describe the non-functional requirements of the system like security, performance, reliability, user interface etc.

* 1. **Use case Diagram**

Use case diagram show the functionality of your system using use case diagram and how the actors interact with the system. Also show use case reusability by including <<include>>, <<extend>>, and <<inherit>> relationships between use cases.

* + 1. **Use case Documentation/ Description**

This is step by step description of the actions performed by each use case. It should contain preconditions, post conditions, main course of action, and alternate course of action.

* 1. **Sequence Diagram**

Sequence diagrams should be drawn for each use case to show how different objects interact with each other to achieve the functionality of the use case.

* 1. **State chart Diagram**

Draw state chart diagram for each class showing the states the object (class) passes through in its life. This model the how objects change from one state to another and when the change happens.

**Activity Diagram**

Draw activity diagrams to show the operations/activities performed by use cases to achieve their functionality. Activity diagrams are drawn for each use case.

* 1. **CRC (Class Responsibility Collaboration) Diagram**

Identify the concepts and things that are important for the system and draw CRC card for them. This helps to identify the objects the system deals with and how they collaborate/interact with each other. This evolves into a class diagram of your system when you create a class diagram for the new system.

* 1. **Conceptual Modeling: Class Diagram**

Create a class diagram that will be the building block the system you will develop. Class diagrams should show the objects the system is comprised of and how they are interrelated.

* **Identifying Change Cases**

Change cases are used to describe potential modifications requirements to the system. You describe the potential change to your existing requirements, indicate the likeliness of that change occurring, and indicate the potential impact of that change.

* 1. **User Interface Prototyping**

Create a user interface prototype and include in the analysis document. This may help to gather more requirements from users or show how the system works. You can create the prototype using UI prototyping tools.

**Chapter Three: Design**

1. **Introduction**
   1. **Purpose and Goals of Design**

Explain the purpose and design goals of the system by using different criteria, like Performance, Dependability, end user…etc.

* + 1. **Overview of System Design**

Explain system design phase which provides a blueprint for implementation, testing and maintenance phases of the system.

* + 1. **Design Goal**

Explain design goals of the system represent the desired qualities or good-qualities of your system and providing a consistent set of criteria that must be considered when making design decisions.

* 1. **Sytem Architecture**
     1. **Current Software Architecture**

Try to illustrate the current software architecture of your system if any.

* + 1. **Proposed Software Architecture**
  1. **System Decomposition**
     1. **Subsystem Decomposition**

Subsystem diagram shows the service it provides or it accepts from other subsystems, and the coupling and the coherence between them.

* 1. **Class Collaboration Diagram**

A collaboration diagram shows an interaction organized around the objects in the interaction and their link to each other.

* 1. **Layering Class Model**

Layering class model shows how all classes such as system class, user interface class, and domain class are grouped and depicts their relationship.

* 1. **Component Diagram**

Systems may be built from components in component-based architecture. Component diagram shows how objects (classes) in your system are grouped together and form components. The components interact with each other either in giving service to other components or requesting service from other component.

* 1. **Deployment Diagram**

Deployment diagram show how the system is deployed on computers. In other words, it shows which component of the software is installed on which machine and how they communicate with each other if they are on different machines.

* 1. **Database Design (E-R diagram) for Relational Database**

Design the database of your system for persistent data storage using E-R diagram. Show the cardinality of entities clearly and how the E-R diagram is mapped to database tables.

* 1. **Persistence Modeling for Object Oriented Database**

If you use object-oriented databases for your system instead of relational databases, instead of designing E-R diagram, design persistence models. Show the mappings and the relations of the tables.

* 1. **Access Control and Security**

In the systems, different actors have access to different functionality and data. Define the access controls for your system.

* 1. **User Interface Prototype Development**