Ministry of Higher Education

LANDMARK METROPOLITAN UNIVERSITY INSTITUTE

“Training Productive Leaders”

School: COMPUTER ENGINEERING
Department: SOFTWARE ENGINEERING

TOPIC:
ANALYSIS AND DESIGN OF A WEB BASED STUDENT ID CARD MANAGEMENT SYSTEM FOR LANDMARK METROPOLITAN UNIVERSITY INSTITUTE

A Long Essay Submitted in Partial Fulfillment of the Requirements for the Award of a Bachelor of Technology (B-Tech) in Software engineering

Presented By:

KWA ADELLA MUNSHI

Registration Number:

LHI24SWE017

Supervisor

MR. KANG MODEST EKOME

Academic Year
2023/2024

CERTIFICATION

This is to certify that the long essay entitled: **“ANALYSIS AND DESIGN OF A WEB BASED ID CARD MANAGEMENT SYSTEM FOR LMUI**” submitted to the department of Software engineering, Faculty of Computer Engineering of Landmark Metropolitan University Institute, Buea in partial fulfillment of the requirements for the award of the Bachelors of Technology (B-TECH) Degree in Software Engineering is the original work of **KWA ADELLA MUNSHI (LHI24SWE017)** conducted under supervision.

Commented [EM1]: Bold the title
Bold your name and Mat

The thesis has been duly acknowledged and referenced.

Sign______________________________________ Date _______________________________

MR KANG MODEST EKOME

(Supervisor)

Sign______________________________________ Date _______________________________

MR KANG MODEST EKOME

(Head of Department)

DECLARATION

I declare that this Long Essay has not been submitted, in whole or in part, anywhere in application for any diploma or degree.

Name…………………………………………………………………………………………………..

Date ………………………………………………….Signature…………………………………...

DEDICATION

This work is dedicated to my father Mr.NJINGTI MOSES KWA, whose unwavering support, encouragement, and belief in my abilities have been a constant source of inspiration. His guidance and love have been instrumental in every step of my journey. Thank you for always being my pillar of strength.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my profound gratitude to my supervisor, MR. KANG MODEST EKOME, for his essential counsel, insightful input, and unwavering support throughout this journey. His knowledge and commitment have been crucial to this project's success.

I also extend my heartfelt thanks to my family for their unfailing love, support, and encouragement. Their endurance and confidence in me have served as a continual source of inspiration.

To my friends, I am deeply grateful for your support, inspiration, and encouragement throughout this project.

Lastly, I give thanks to God for giving me the courage, discernment, and endurance needed to complete this mission. Without His favor, this achievement would not have been possible.

ABSTRACT

The creation and deployment of a web-based system for the Landmark Metropolitan System to generate student ID cards is presented in this project. The principal aim is to facilitate students' creation of virtual ID cards in a simplified and effective manner, allowing administrators to approve and print them as needed. Students can use this system to log in, create accounts, fill out forms, and submit their information for the creation of ID cards by utilizing contemporary web technology. The suggested method seeks to ensure correctness and usability by substituting automated, user-friendly interface for conventional, prone to error manual methods.

The system uses an iterative waterfall development process to guarantee well-organized development and comprehensive testing at every turn. The creation of user accounts, form submission, data checking, and administrative approval are among the essential features. The

LIST OF ABBREVIATIONS

- **ID -** Identity Document
- **HTML -** Hypertext Markup Language
- **CSS -** Cascading Style Sheets
- **UML -** Unified Modeling Language
- **VSCod**e - Visual Studio Code
- **UI** - User Interface
- **DBMS** - Database Manage
- **HTTP -** Hypertext Transfer Protocol
- **HTTPS** - Hypertext Transfer Protocol Secure
- **API -** Application Programming Interface
- **IDE** - Integrated Development Environment

TABLE OF CONTENTS

Commented [EM2]: Font Type: Times new romans

LIST OF FIGURES.

Commented [EM3]:
Font Type: Times new romans
Remove empty pages.

CHAPTER ONE

GENERAL INTRODUCTION

1.1 INTRODUCTION.

Modernizing the process of providing and administering student identification cards in educational institutions is the goal of the Student ID Card Generator System project. Conventional approaches are frequently ineffective, prone to mistakes, and vulnerable to security lapses. These antiquated procedures may result in heavy administrative loads, a delay in the distribution of cards, and possible dangers from unapproved entry to school premises.

In this project, a web-based system that automates and simplifies the entire student ID card generation process is introduced. Via an easy-to-use web interface, administrators will be able to swiftly create ID cards, update and enter student information. By ensuring that all data is precisely recorded and safely kept, the system reduces the possibility of mistakes and improves the overall security of student identity.

The automated generating process enabled by the web-based system makes significant gains in speed, accuracy, and administrative efficiency, even if the final product is still a physically printed ID card.

The Student ID Card Generator System offers a comprehensive solution to the current problems with student ID card management by utilizing contemporary web technologies. Improved security, streamlined operations, and compliance with environmental objectives all add up to a safer and more effective learning environment. In addition to meeting urgent needs, this initiative lays the groundwork for upcoming technological developments in student services and school administration

1.2 Background of the Study.

In identifying the need for an environmentally friendly, safe, and more efficient solution, this project study suggests implementing a web-based system for creating student ID cards. This system uses web-based technology to automate every step of the process, from data entry to card creation, in contrast to traditional techniques that mostly rely on human participation.

Using only a few clicks, administrators will be able to design card, enter student information into the system, and start the card generation process using the suggested system. The system will guarantee the timeliness and accuracy of student information.

Administrators will have more flexibility, ease, and control over the card issuing process because the entire process will be facilitated through the online interface, even if the final output will be a physically printed ID card. Furthermore, by decreasing dependence on tangible resources and optimizing administrative procedures, the system is consistent with the organization's dedication to ecological sustainability.

1.2 Statement of the Problem.

The process of generating and maintaining student identification cards is beset with errors, inefficiencies, and security flaws in many educational institutions. Manual data entry is a common component of traditional approaches, which can be tedious and prone to human error. These antiquated procedures put the institution's overall security at risk by causing delays, misplaced or lost cards, and illegal access to campus buildings.

Moreover, the lack of integration with modern technology limits the ability to provide students with convenient access to ID cards that can be regenerated if missing, which can be used for a variety of on-campus services and events. This gap not only affects administrative efficiency but also detracts from the student experience, as students face unnecessary hurdles in accessing essential services.

1.4 Objectives of This Project.

1) Simplify Student Identification: Make it easier to recognize students during different school-related activities and events.
2) Boost Administrative Effectiveness: Lessen the paperwork involved in obtaining and maintaining student ID cards.
3) Encourage school branding by making sure that all student ID cards are professionally made and bear the school's branding.
4) Cut Costs: Keep the expenses related to creating and maintaining physical identification cards to a minimum.

5) Enable Data Integration: For easy data management and synchronization, integrate with the current school databases.

6) Boost Accountability: By utilizing ID cards for access and attendance tracking, you may help students and staff feel more accountable.

7) Encourage a Safe Environment: Limit access to the school grounds to only those who are permitted, in order to improve the general safety of the school environment.

8) Encourage Inclusivity: By using a standardized ID system, make sure that all students have an equal chance to access school services and resources.

9) Facilitate access control

1.4.1 Specific Objectives.

1) Provide a user-friendly interface: Provide administrators with an easy-to-use web-based interface to generate and distribute student ID cards.
2) Provide an automated system to create easy and quick IDs to identify Staff, students, visitors etc.
3) Working on this project enables me to put the technical know-how and understanding I've gained during my education to use. It offers a useful, hands-on experience that closes the knowledge gap between academic study and practical application.
4) Practicalize the use of waterfall methodology in developing the project.
5) By increasing the effectiveness of ID card administration, my proposal will have a direct impact on the day-to-day operations of my institution. This gift will help current and future faculty and staff members and serve as a concrete, enduring memorial to my time at the university.
6) Gain professional experience while putting my skills into practice

1.5 Significance of the Study.

1) It will make it easier to identify visitors, employees, and children, which will lower the risk of illegal entry and improve school safety overall.
2) Identification is improved by the use of standardized student ID cards.
3) Stops unwanted access to the school's property.
4) Contributes a safe and conducive that makes the classroom a safer place to learn.

5) Minimizes human mistake and the administrative load.
   Maintains current and accurate student records
6) Offers a contemporary and dependable method of identification.
7) Makes it easier to access school events, resources, and facilities.

1.6 Scope of the Project.

The scope of a project refers to the extent to which the project will be developed. This project intends to implement a web system for students to apply for, generate and manage ID cards. It will implement key features such as User Login, ID application form, Admin approval, integration with a database management system. Deliverables (which are tangible or intangible outputs produces as a result of completing a project) will include a functional web application, a user guide, an admin guide, a technical documentation.

1.7 Definition of Terms. Some key terms related to this web student ID card generation project include:

1) **ID card:** An identification card provided to students, containing personal information.
2) **Virtual card**: a digital version of the student ID card that can be accessed and viewed online.
3) **Web based application:** An application that is accessed through the web browser over a network such as the internet.
4) **User interface**: The environment you see when you launch your application**.**
5) **Approval process:** The steps involved in reviewing and authorizing the creation and printing of student ID cards.
6) **Login credentials:** The user information used by students and admins to access the system.
7) **Waterfall model**: A linear sequential approach to software development, where each phase must be completed before the next begins**.**
8) **Actor:** anyone/anything or another system that interacts with the system or the system initiates an interaction with them
9) **Admin:** a user with elevated privileges that manages the system
10) **Unified modelling language:** a standardized modelling language used to visualize the design of a system.

11) **System design**: the process of defining the architecture, components, and interfaces of the system to satisfy specified requirements**.**

12) **Life line:** a vertical dash line in a sequence diagram representing the existence of an object over time.

13) **Iteration:** a repetition of a process in software development allowing for review and refinement.

14) **Gantt chart:** a type of bar chart that represent a project schedule showing the start and end of each project task.

15) **Front end development:** The client-side development focusing on the user interface, and user experience of the application**.**

16) **Database:** An organized collection of structured data stored electronically in a database management system on computer for easy data management.

17) **Dataflow Diagram**: A graphical representation of data within a system showing how data is processed and transferred between parts of the system**.**

18) **Back-end development:** A server-side development focusing on database interaction, server logic, application functionality.

1.8 Organization of the Study.

The goal of this study is to develop and implement a web-based student ID card generator system for Landmark Metropolitan University. It is structured in a way that makes it easy to do a thorough analysis of the methods, findings, and implications of the research in a clear and systematic way. Each chapter is organized to address specific aspects of the project, ensuring a logical flow and comprehensive coverage of the topic.

Chapter 1: Overview

This chapter serves as an introduction to the study by summarizing the reasons why a web-based system for generating student ID cards is necessary, especially when it comes to educational institutions. The problem statement, study goals, research questions, significance, and scope are all included. This introductory chapter establishes the scene by outlining the goal and significance of the research while directing readers through the study's primary topic and anticipated contributions.

Chapter 2: Review of Literature

Thorough examination of earlier research on both conventional and contemporary ID card generating technologies is provided by the literature study. This chapter places the current topic within the larger academic debate, critically assesses previous research, and identifies research gaps. It provides a strong theoretical and empirical basis for the creation of the system that generates student ID cards.

Chapter 3: Methodology

The study design and development process for the web-based student ID card generation system are described in depth in this chapter. It explains the requirements analysis, system design, feasibility research, and waterfall approach implementation. The chapter provides a planned and methodical approach to development by outlining the exact steps involved in gathering requirements and designing the system.

Chapter 4: System Implementation and Testing.

The actual system implementation is covered in Chapter 4, which also describes the hardware and software tools utilized, how the system components are broken down, and the process of implementation. It also contains information about standards, testing, and validation methods used to make sure the system satisfies requirements and performs as intended.

Chapter 5: Findings, Conclusion and Recommendation.

The outcomes of the system's testing and deployment, including user comments, performance metrics, and system evaluation, are presented in this chapter. It analyzes the results and any problems that arose during testing and implementation, and it talks about how well the system met the project's objectives.

References A comprehensive list of all the sources used in the study, created in accordance with accepted citation guidelines to guarantee credibility and academic integrity.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter explores the literature on the creation of a web-based system for generating student ID cards, emphasizing the theoretical and practical elements that support the project. There are two primary divisions in the review: related concepts and related works. While the related works section assesses comparable systems and projects to gain knowledge and shape the suggested solution, the related concepts section covers key technologies and theories that serve as the project's foundation

Commented [EM4]: Paragraph spacing should be 1.5 like line spacing. Do this for all titles and sub titles

2.2.1 Review of Related Concepts or Conceptual Framework.

Web-based systems: These are programs that can be accessed via a network, like the internet or a company intranet. Central data administration, ease of maintenance, and cross-platform interoperability are just a few benefits of these systems. Notable technologies include JavaScript, HTML, CSS, and front-end and back-end development frameworks like React.js and Node.js, respectively.

Student ID card Systems: Systems for creating, organizing, and printing student ID cards are known as student ID card systems. Forms for entering data, image processing, and safe data storage are essential elements. In order to improve data accuracy and expedite the administration of student information, databases are frequently integrated with modern ID card systems.

Firebase: Google created the Firebase platform, which allows developers to create both online and mobile applications. It comes with a number of utilities, including Firebase Authentication for user management, Fire store for database administration, and Firebase Storage for file upload handling. Strong security features and real-time data syncing are offered by Firebase.

Authentication and Authorization: In web applications, it is essential to make sure users are who they say they are and have the right to carry out particular operations. To implement these capabilities securely, technologies like Firebase Authentication and JSON Web Tokens (JWT) are frequently employed.

Cloud Computing. Makes use of distant servers housed online for data processing, management, and archiving. Scalable storage and processing capacities for the ID card generation system are provided by cloud computing, which also provides flexibility, scalability, and lower infrastructure costs.

Data Encryption: Transforms data into a format that is coded to stop unwanted access. Backend data encryption can be achieved by using JavaScript libraries such as crypto-js or the crypto module in Node.js, which guarantee the confidentiality and integrity of sensitive data**.**

Image Processing: Image manipulation techniques, essential for working with ID photographs. JavaScript libraries such as Jimp and Sharp offer functions for resizing, cropping, and optimizing images to make sure they adhere to ID card requirements.

Responsive Web Design: makes sure web apps run well on a range of screens and gadgets. The user experience is improved on computers, tablets, and smartphones using flexible grids, layouts, pictures, and CSS media queries.

2.3 REVIEW OF RELATED WORKS

2.3.1 PROJECT NAME: CANVAS

AUTHOR: Instucture, Inc.

TECHNOLOGY USED:

- Open-Source Platform
- API and LTI Standards
- Customizable and Scalable Tools
- Mobile Accessibility
- Integration with Educational Technology Partners
- Part of the Instructure Learning Platform

STRENGTHS OF THE SYSTEM.

- User-Friendly Interface: All users may easily explore and utilize Canvas with ease.
- Integration Skills: It easily connects with a wide range of external programs, including Zoom (Instructure) and Google Classroom.

- Customization & Flexibility: A wide range of choices enables teachers to design courses that meet the needs of particular students (Instructure Community).
- Mobile Accessibility: According to the Instructure Community, learning is mobile-friendly thanks to specialized mobile apps.
- Strong analytics: Effective technologies monitor engagement and progress of students (Instructure).
- Community and Support: (Instructure) offers a sizable user base as well as extensive support resources.

LIMITATION.

- Steep Learning Curve: Canvas may be challenging for new users to get the hang of
- Inconsistent Browser Performance: The functionality of various web browsers can differ.
- Absence of Auto-Save Feature: Students' work is not automatically saved, putting their progress at risk.
- Complex Initial Setup: Because there aren't many teaching resources available, setting up classes might be difficult.
- Limited Offline Access: Mostly cloud-based, this system restricts use when not connected to the internet.

RECOMMENDATOIN FOR FUTURE WORK.

- Enhance User Onboarding: Provide thorough guides and training materials to help new users with the setup and learning process.
- Improve Browser Compatibility: To prevent access problems, make sure all major web browsers operate consistently.
- Add Auto-Save Feature: To avoid students losing their work, include an automatic save function.

2.3.2 PROJECT NAME: ID CARD GENERATOR SYSTEM

AUTHOR: Thompson Desjardins

TECHNOLOGY USED

In the development of an ID card generator system using PHP, there are several proposed methods and techniques that can be used to ensure the system is efficient, reliable, and secure. Some of these proposed methods are:

•**Object-Oriented Programming:** Object-oriented programming (OOP) is a programming paradigm that emphasizes the use of objects, which are instances of classes, to represent and manipulate data. OOP can be used in the development of an ID card generator system using PHP to improve the system's modularity, reusability, and maintainability.

In an ID card generator system, OOP can be used to represent different entities such as the user, the card, and the database. The following is an example of how OOP can be used in the development of an ID card generator system using PHP.

1**) User class.**

The User class represents a user of the system. It can have properties such as name, email, and password, and methods such as login and logout. The User class can be used for user authentication and authorization.

2) Card class.

The Card class represents an ID card. It can have properties such as name, photo, and ID number, and methods such as generate and print. The Card class can be used for generating and printing ID cards.

3) Database class

The Database class represents the database used by the system. It can have methods such as connect and query, which can be used for connecting to the database and executing SQL queries. The Database class can be used for storing and retrieving user and card data.

- Server-side Validation.
- Client-side Validation.
- Model-View-Controller (MVC) Architecture
- Image Processing.

STRENGTHS:

- OOP enhances modularity, allowing different parts of the system to be developed and tested independently.
- Because objects and classes can be reused across the program, code reusability is boosted.
- The system's well-organized structure makes it simpler to update and maintain.
- Validation on both the server and the client side aids in preserving data integrity and guards against typical security risks.
- User and card data may be efficiently and neatly stored and retrieved thanks to the Database class.
- Automatic handling of photographs for ID cards is made possible by built-in image processing techniques.

LIMITATIONS:

- Initial setup and setting can be difficult and necessitate a high level of technical proficiency.
- Requires substantial server resources for running PHP and handling database operations.
- Requires developers to have a good understanding of OOP
- If not properly implemented, server-side validation and session management can expose the system to security vulnerabilities.
- Server-side validation and session management can leave the system vulnerable to security flaws if they are not implemented correctly.

RECOMMENDATIONS FOR FUTURE WORK:

- Develop robust error handling and logging mechanisms to simplify debugging.
- Ensure that client-side validation is compatible with all major browsers.

2.3.3 PROJECT NAME: Design and Implementation of an Automated Students' ID Card Generating System, Study Guides, Projects, Research of Computer Science.

AUTHORS: Leo Simon Anfone

TECHNOLOGY USED:

- **HTML** stands for Hypertext Markup Language, which is used to organize online pages.
- **Cascading Style Sheets,** or CSS, are used to style web pages.
- **Python:** For implementing logic and handling backend processes.

- **Databases:** Used to store created ID card information and user details.
- **Excel File Handling:** Enable the input of large amounts of data via Excel files.
- **Web Application Framework:** The web application will probably be managed by a Python-based framework like Flask or Django.

STRENTHS OF THE SYSTEM.

- Users can easily generate ID cards without the need for technical skills thanks to the system's simple and intuitive design.
- By automating the ID card creation process, a substantial amount of time is saved over manual approaches.
- Provides a range of templates from which users can choose, guaranteeing that the ID cards satisfy particular needs and tastes.
- Enables the effective processing of huge datasets by supporting the uploading of data via Excel files.
- This reduces the possibility of human error by creating ID cards automatically depending on user input.
- Fit for a range of establishments, such as companies, educational institutions, and other organizations requiring ID cards.
- Web-based accessibility: Does not require local software installations and may be accessed from any location with internet connectivity.

LIMITATIONS:

- Dependency on the Internet: In order to access and function, a reliable internet connection is necessary.
- Restricted Offline Functionality: Requires internet access to operate.
- Potential vulnerabilities if data security is not properly safeguarded are a concern.
- Initial Setup Complexity: Technical know-how may be needed to set up the database and backend.
- Maintenance Requirements: To guarantee operation and security, regular upgrades and maintenance are required.
- Scalability Issues: To effectively manage extremely large datasets, further resources and

optimization may be needed.

- Limitations on Customization: Users may require particular customizations that are not offered right out of the box.
- Browser Compatibility: Inadequate testing may result in problems with specific web browsers.
- Integration Challenges: Having trouble integrating with databases and systems that are already in place.

RECOMMENDATIONS FRO FUTURE WORK:

- Including Extra Templates and Customization Features.
- Improved Management and Import of Data.
- Extraordinary Security Capabilities.

2.3.4 PROJECT NAME: Advanced QR Code Based Identity Card: A New Era for Generating Student ID card in Developing Countries

AUTHOR: Md. Sanaul Haque, Richard Dybowski

TECHNOLOGY USED

- Student information is encoded using QR Code technology for fast scanning.
- Image processing is the method used to take and process student photos.
- Database management systems are used to safely store student data.
- Web technologies include JavaScript, HTML, and CSS for front-end programming.
- Python is used for server-side scripting in back-end development.
- Tools that are open source are used in both development and deployment.
- Mobile technologies: use mobile phones to scan QR codes.
- Security technologies: Applied to guarantee confidentiality and integrity of data.
- Tools for data visualization: These could be used to create layouts for ID cards.
- Machine Learning (Optional): For more complex functions, such as photo validation by facial recognition.

STRENGTHS:

- Efficiency: Automation lessens the need for manual labor while creating ID cards.

- Integration of QR Codes: Offers instant access to student data.
- Accuracy: Less mistakes than with handwritten ID cards.
- Customization: Permits adding a photo and other student information.
- Features of a smart card include a photo, student information, and a QR code all combined into one card.
- Open Source: Allows a larger user base to access the software.
- Fast Verification: Student information can be quickly verified by scanning a QR code.
- Secure: Aids in thwarting illegal access and forgeries.
- Scalability: Suitable for big student enrollments.
- Possibility of Integration: Allows for data interchange with other educational systems.

LIMITATIONS:

- Technology Dependency: Needs a smartphone to be able to read QR codes.
- Initial Setup Complexity: Technical know-how may be needed to set up the system.
- Data Security Issues: The system's data must be protected from intrusions.
- QR Code Compatibility: In order to scan QR codes, a device must be compatible with them.
- Internet Dependency: Some functions might need to be connected to the internet.
- Updating and maintaining the system on a regular basis may be necessary.
- Printing Quality of QR Codes: Clear printing of QR codes is necessary for precise scanning.
- Cost of Implementation: For certain institutions, the initial setup expenditures could be substantial.
- User Training: To ensure efficient usage of the system, users may need to undergo training.
- Privacy Concerns: To prevent abuse, student data privacy needs to be properly handled.

RECOMMENDATIONS FRO FUTURE WORK:

- Provide a cloud-based solution that securely stores and manages student data.
- This would improve flexibility and scalability by making student information easily accessible from any location.
- Use data synchronization tools to make sure that data is consistent between devices and

locations.

2.3.5 PROJECT NAME: Electronic Student Identity Card Management System at the Poznan University of Technology

AUTHOR: Marek Gosławski

TECHNOLOGY USED:

- Distributed System Architecture: Used to manage data among several academic institutions.
- A centralized system for personalizing cards is the Inter-University Centre for Personalization (MCP).
- Plastic card printers: Electronic student identity cards are printed using specialized printers.
- Data exchange protocols: These provide communication between student information systems, ISELS, and SELS.
- Student Information System (SIS): Provider of student information for customized purposes.
- Database management systems are employed in the storage and administration of student data.
- Security protocols are put in place to guarantee the privacy and security of data.
- Card Lifecycle Management: Oversaw the electronic student identity cards' entire existence.

STRENGTHS OF THE SYSTEM:

- Standardization and uniformity in card customization are ensured by the centralized personalization center.
- Customization: Specifically designed solutions to meet the needs of every university.
- Automatic Data Retrieval: Uses the Student Information System to automatically retrieve student data.
- Flexibility in Integration: Manages varying degrees of computerization throughout universities.
- Card Lifecycle Management: Handles the electronic student identity card lifecycle effectively.

- Application Management: Various apps installed on the card are independently managed by this function.
- Data integrity: Assures accuracy and consistency of data when exchanging data.
- Resistance to Communication Breaks: Made to efficiently manage communication hiccups.
- Compliance with Security: Adheres to specific security guidelines for the issuance of electronic student ID cards.
- Scalability: The ability to grow and change to meet the demands of various colleges.

LIMITATIONS:

- Dependency on SELS: Real-time data updates may be limited if communication is solely started by SELS.
- Complicated Implementation: It could be difficult to integrate and guarantee compatibility with current systems.
- Limited Data Source: All data is sourced from the Student Information System at the nearby university.
- Possible Privacy Issues with Data: requires that sensitive student data be handled carefully.
- Risks of Communication Breaks: Interruptions in communication might have an impact on data synchronization.

RECOMMENDATIONS FOR FUTURE WORK:

- Boost the Student Information System (SIS) Integration.
- Put Mobile Application Integration into Practice.
- Strengthen Security Protocols.

Commented [EM5]:
Summarize all of this to max 4 pages

2.4 PROPOSED SOLUTION: The high-level organization and design concepts that specify how a system's constituent parts and subsystems communicate and function collectively are referred to as its architecture. It includes how the parts of the system are put together, how they work together, and the guiding ideas that have shaped it throughout time.

Proposed System Architecture for a Web-Based Student ID card Generator System.

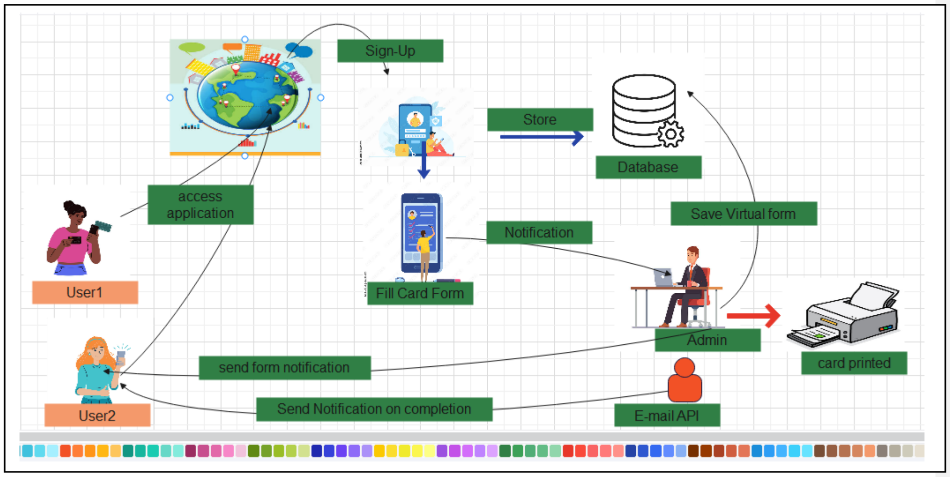


Figure 1: Architecture of proposed system.

1) Client-Side (Frontend).

a) Technologies : HTML, CSS, JavaScript.

b) Components.

- User Interface: Designed using java script to ensure a dynamic and responsive experience.
- Forms: For user registration, login, and ID card creation.
- Validation: Client-side validation to ensure data integrity before submission.
- Preview: ID card preview functionality before submission.

2) Server-Side (Backend).

Technologies: Node.js (JavaScript)

Components:

- Authentication: Handles user authentication and authorization
- Business Logic: Manages ID card creation, form processing, and approval workflows.

- Restful API: Exposes endpoints for client-side interaction and data retrieval.

3) Data Base

- Technology: Firebase Fire store
- Collections

a. Users: Preserves user data.
b. ID Cards: Keeps card information stored.
c. Stores administrator approvals and denials

4) Security Layer

- Technology: Firebase Authentication
- Data encryption: I will use java script (Node.js) to encrypt data at the backend.
- Token-based authentication with OAuth 2.0 provides secure user authentication.
- Authorization: Admin and user privileges are managed by role-based access control.

5) Notification Service.

- Technology: An E-mail API
- Components: Email Notifications: Sends account creation, submission status, and approval notifications.

6) **Admin Interface.**

- Firebase Authentication: For managing and authenticating users.
- Firebase Fire store: Used to store user information, ID card information, and other pertinent data.
- Firebase Storage: For keeping other files and photos that have been uploaded.
- Restful APIs: For front-end and back-end communication.

CHAPTER THREE

METHODOLOGY

3.2 INTRODUCTION

The approach used to create Landmark Metropolitan University's web-based student ID card generation system is described in this chapter. It offers a thorough explanation of the selected approaches, instruments, and strategies, together with a thorough description of the model that will be applied. In order to guarantee a reliable and user-friendly application, the chapter covers the system design, implementation, testing, and deployment procedures.

3.2.1 Research Methodology:

Iterative Water Fall Model: The technique selected for this project is a hybrid approach that combines the iterative feedback mechanisms of agile practices with the structured phases of the Waterfall model. Throughout the development process, this strategy permits flexibility and ongoing improvement while guaranteeing comprehensive planning and documentation.

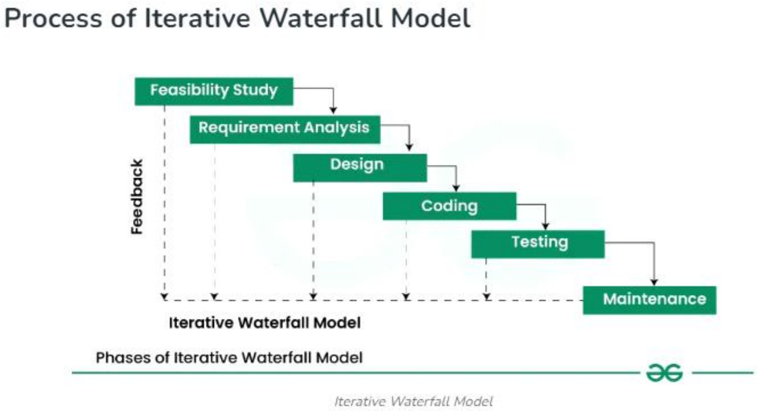
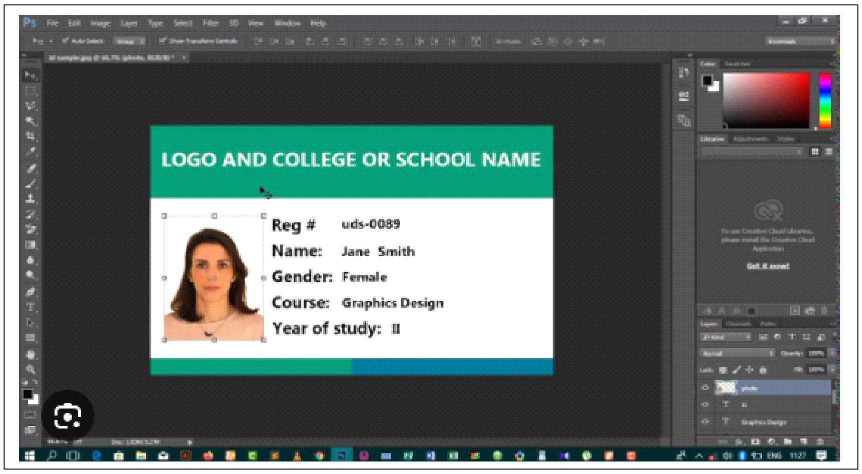

Process of Iterative Waterfall Model
Feasibility Study
Requirement Analysis
Design
Coding
Testing
Maintenance
Feedback
Iterative Waterfall Model
Phases of Iterative Waterfall Model
Iterative Waterfall Model

Figure 2: Iterative Waterfall Module. Source: internet

1) Feasibility Study:

Landmark Student-Id Card Generator.

Brief Description: Adobe Photoshop is used by Landmark Metropolitan University's current ID card generation system to create ID card templates. Local computers are used to upload and incorporate student photos into these designs. The cards are created, filled in with the required information and student images, and then printed out in hard copy. There is no system integration with external systems; this procedure is managed internally. Photoshop is used to create customizable, high-quality ID cards, but graphic design expertise and physical labor are needed for the procedure. The workflow is made simpler by the system's local setup, but it might be resource-intensive and contingent on the availability of trained staff.



LOGO AND COLLEGE OR SCHOOL NAME

Reg # uds-0089

Name: Jane Smith

Gender: Female

Course: Graphics Design

Year of study: II

Figure 3: study case user interface. Source Photoshop interface

Commented [EM6]: Image source???? Please include a source to all your images.

A) Detailed Workflow of the ID card Generation System.

1) Data collection and Preparation.

- Data collection: personal information about students, such as names, ID numbers, and other pertinent information are collected from a local storage.

- Data Storage: Keep this information locally on safe devices that only authorized users can access.

2) **Template Design.**

- Photoshop Setup: The Adobe Photoshop is launched and a new file with the right size for an ID card all created.
- Design elements: They go ahead to Include text areas for names, ID numbers, departments, and other pertinent information, as well as placeholders for student photographs and institution branding.

3) Image Upload and Insertion.

- Image Upload: the Adobe Photoshop is used to import student images from local computers.
- Insertion: the pictures are inserted into the appropriate spaces on the templates for ID cards.
- Alignment and Adjustment: the photographs are verified to be positioned correctly to fill the placeholders.

4) Customization with Student Details.

- Data entry: the appropriate text fields are appropriately filled on the templates by hand with each student's information, including name, ID number, department, etc.
- Review: the information entered is verified accurate and consistent.

5) Review and Approval.

- Quality Check: each ID card is checked to make sure all the information is accurate and the photographs are positioned correctly.
- Repairs: Before continuing, the Photoshop is used to make any necessary repairs.

6) Printing.

- Print Test: To ensure quality, a test print runned.
- Final Printing: hard copies of the finalized ID cards are printed.

7) Distribution.

- Sorting: The ID cards are arranged according to courses, departments, or other pertinent factors.

B) Security Measures.

- **User Authentication**. To limit access to authorized users, they utilize role-based access control
- **Backup and Disaster Recovery.** They made regular backups and store them safely so that data may be restored in an emergency.
- **System Security.** To stop malware assaults, they kept firewalls and antivirus programs up to date and updated software on a regular basis**.**
- **Data Security**. They also used strict access controls and password protection for sensitive data.

C) Challenges and Issues

1) Complexity in Design.

- For those who are not designers, creating ID card templates in Adobe Photoshop might be difficult and require graphic design knowledge.

2) Manual Data Entry.

- When entering student information by hand into ID card templates, particularly in big quantities, it can be laborious and error-prone.

3) Resource Intensive.

- Operating costs can go up with high-end technology and Adobe Photoshop software subscriptions.

4**) Integration Difficulties.**

- It could be difficult to integrate this system with other university information systems, which could cause problems with data synchronization.

3.3) Requirement Analysis (Tools and Materials used).

3.3.1 Hardware Requirement.

Developer Workstations:

- Hard Disk: At Least 500 GB and above.
- Processor: 1.90 GHZ
- RAM: At Least 8 GB.
- Mobile Phone: Android.

3.3.2. Software Requirement.

- Database: Firebase.
- Operating System: windows 10.
- Application Development Tool: Visual studio code (VSCode)
- Extensions: HTML, CSS, JavaScript.
- Repository Hosting Service: GitHub

3.4 System Modules.

1. User Management Module

a) User Registration.

- Permit users create accounts, including administrators and students.
- Verify user input and address mistakes.

b) User Login and Authentication.

- Secure login system with username/email and password.
- Implement password reset functionality.

c) User Roles and Permissions.

- Differentiate between student and administrator roles.
- Implement role-based access control (RBAC).

2) ID Card Management Module.

a) ID Card Creation

- Provide forms for students to enter personal information.
- Upload and handle profile pictures.

3) Approval and Review Module.

- Admin Dashboard: Give administrators a way to see and control requests for ID cards.
- Permit administrators to examine submitted requests for ID cards.
- Acceptance/Rejection: Grant or deny requests for ID cards.
- Use system notifications or email to inform students of the status.

4) Notification Module.

- Email Notifications: Send emails confirming registration, submission of ID cards, and approval or rejection.

5) Data Storage and Management Module.

- Database Design: Create a database structure to hold logs, ID card information, and user data.
- Use data encryption for sensitive information, such as passwords and personal information.

6) User Interface (UI) Module.

- Frontend Development: Create interfaces that are easy to use for all user roles.

7) Testing and Quality Assurance Module.

- Unit Testing: Create and execute unit tests for distinct parts.
- Integrity testing: Examine how well various modules work together.
- Conduct user acceptance testing (UAT) to make sure the system satisfies user needs.
- Performance testing: Evaluate how well the system performs under varying loads.

8) Deployment and Maintenance Module.

3.5 System Analysis.

Understanding and recording both functional and non-functional requirements, as well as defining system requirements and user needs, are all part of system analysis. These steps result in comprehensive specifications that will direct the development of the web-based student ID card generator system at Landmark Metropolitan University. Aspects of system analysis such as data flow, system architecture, non-functional requirements, and functional requirements will all be covered in this part.

3.5.1 Functional Requirement.

- The system must allow students and administrators to create accounts.
- Must verify registered users say, thorough 2 or 3 step verification into the system or through a better method.
- Allow users to log in and out securely.
- The system must let students fill out a virtual ID card by providing personal information.
- The system should enable administrators to review, approve, or reject ID card requests.
- Status Notifications: Notify students about the status of their ID card requests (submitted, approved, rejected)
- The system should securely store student information and ID card data.

3.5.2 Non- Functional Requirement.

1) Performance.

- Response Time: Ensure fast response times for most operations
- Scalability: Design the system to handle increasing loads without performance degradation.

2) Usability.

- User Interface: The UI should be intuitive and easy to use.
- Accessibility: The system should be accessible to all users

3) Reliability.

- Availability: The system should have high availability.
- Error Handling: The system should handle errors gracefully and provide meaningful error messages.

4) Security.

- Authentication and Authorization: Secure authentication and role-based access control.
- Data Protection: Encrypt sensitive data in transit and at rest.

5) Maintainability.

- Modularity: The system should be modular to facilitate easy maintenance and updates.
- Documentation: Comprehensive documentation for users and developers.

3.5.3 Cost Evaluation.

ITEM	**DESCRIPTION**	**ESTIMATE COST(FCFA)**
Laptop	High performance laptop for development and hosting	140,000
Visual Studio Code	Development environment (free software)	**0**
Firebase	Backend as a Service for database and hosting (free tier)	0
StarUML	Tool for creating UML diagrams (free software)	**0**
Internet	Daily internet cost for development (6 weeks, 500/day)	22,000
EdrawMax	Tool for drawing technical system architectures	**0**
Miscellaneous	Additional unforeseen expenses	5,000
TOTAL		**167,000**

Figure 4: cost evaluation table

3.5.4 PROJECT SCHEDULE.

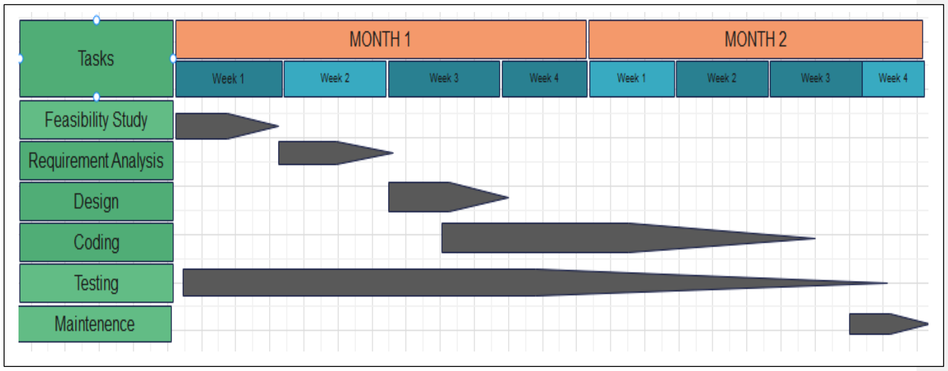


Figure 5: Project Schedule. Source EdrawMax

DESIGN

3.5.5 USE CASE ANALYSIS.

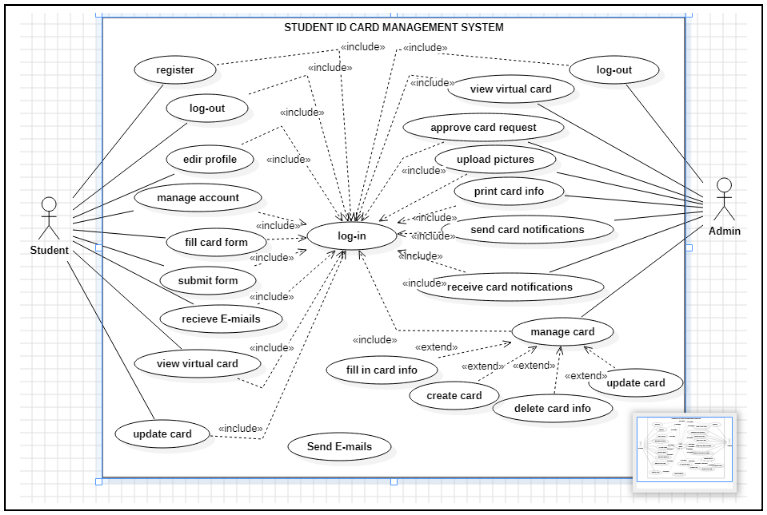


Figure 6: Use Case Diagram. Source starUML

3.5.6 SEQUENCE DIAGRAMS. A sequence diagram is a type of interaction diagram in Unified Modelling Language (UML) that shows the sequence of actions, messages, events, between different objects or components in a system over time.

a) Sign-up

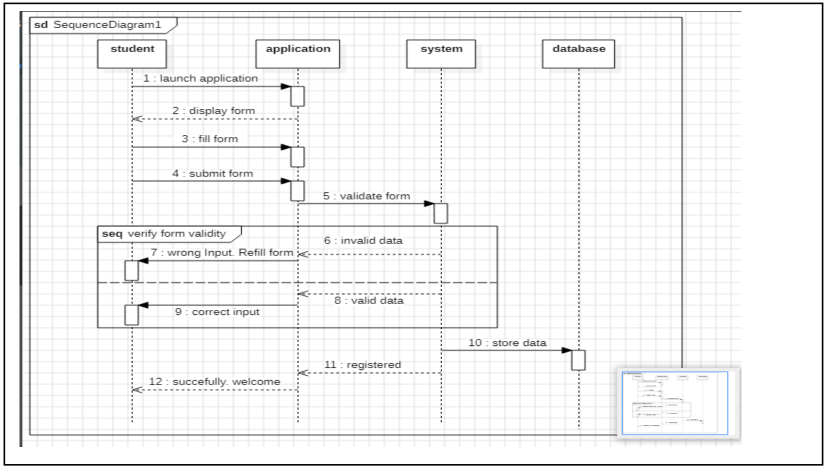


Figure 7: Student register sequence diagram. Source starUML

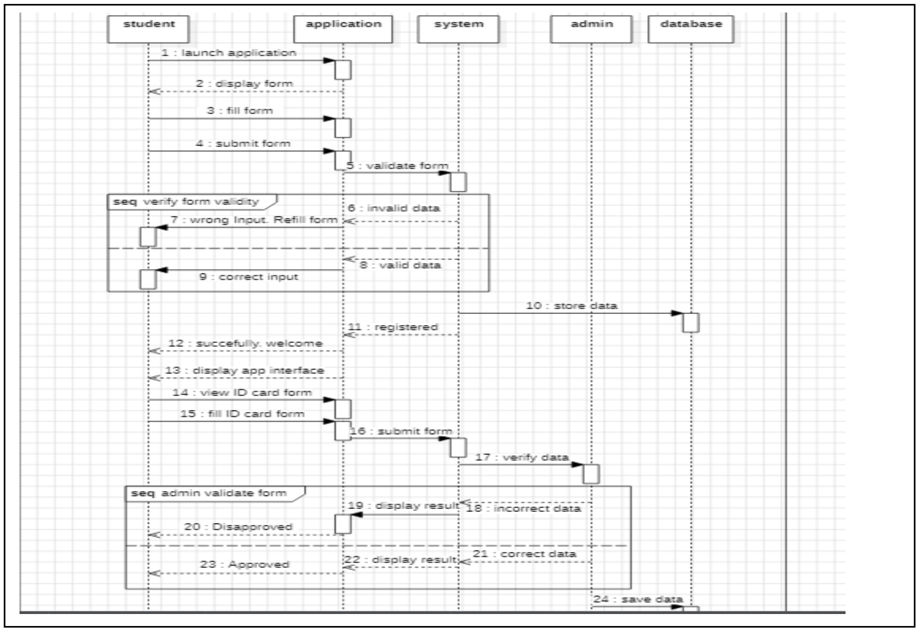


Figure 8: fill form sequence diagram. Source starUML

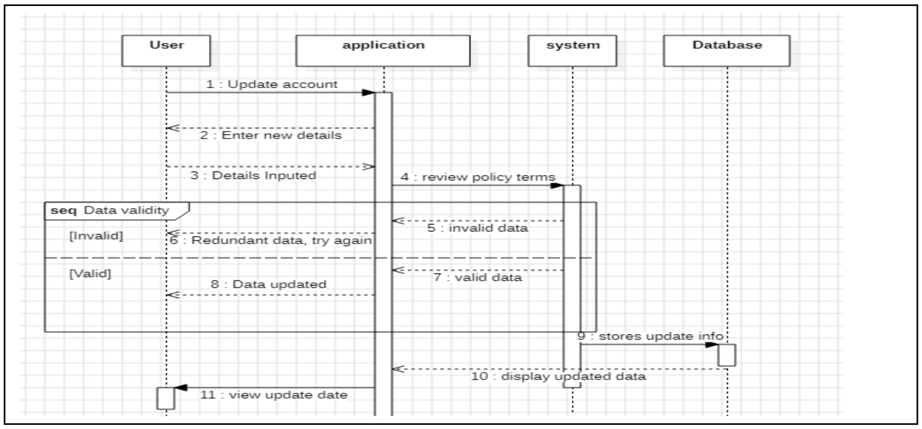


Figure 9: Update data sequence diagram. Source starUML

3.5.7 Flow Chart: A flowchart is a graphical representation of a workflow, system, or process that shows the flow of control from one step to the next using arrows to indicate different sorts of activities or stages.

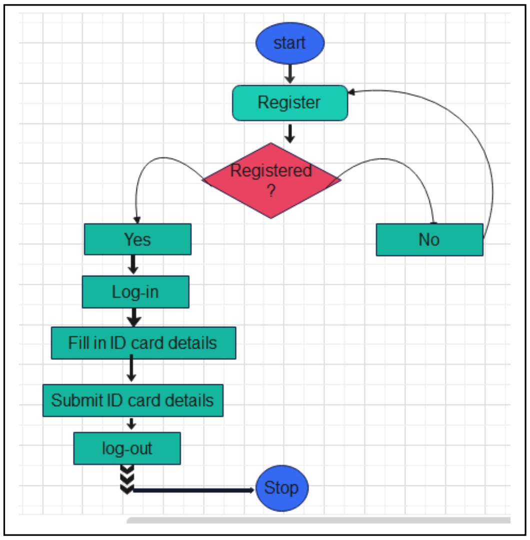


Figure 10: student flow chart. Source EdrawMax

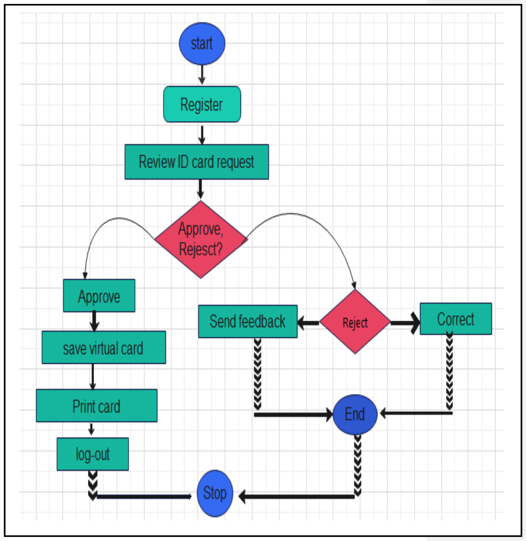


Figure 11: Admin flow chart

3.6 System Design.

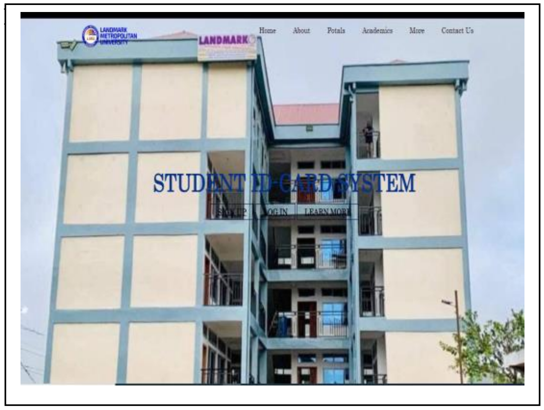


Figure 12: on-boarding. Source figma

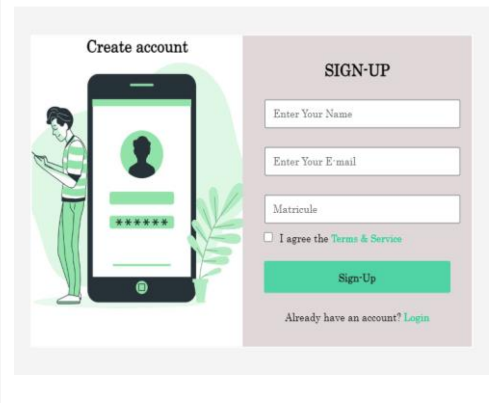
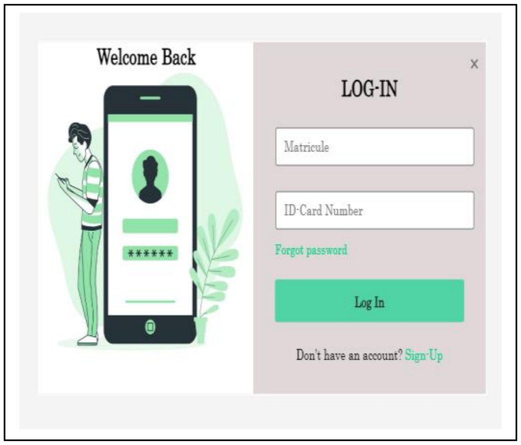


Figure 13: Create account. Source

figma



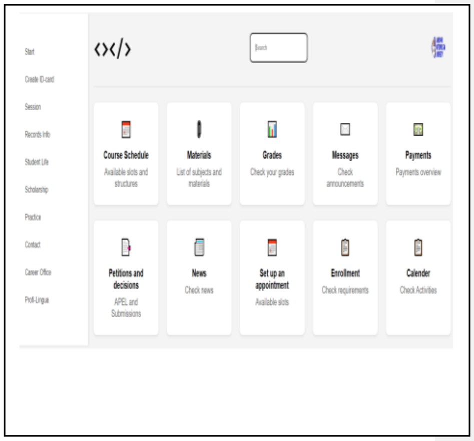


Figure 14: log-in. figma *Figure 15: student portal.*

Source figma

3.6.2 Class Diagram.

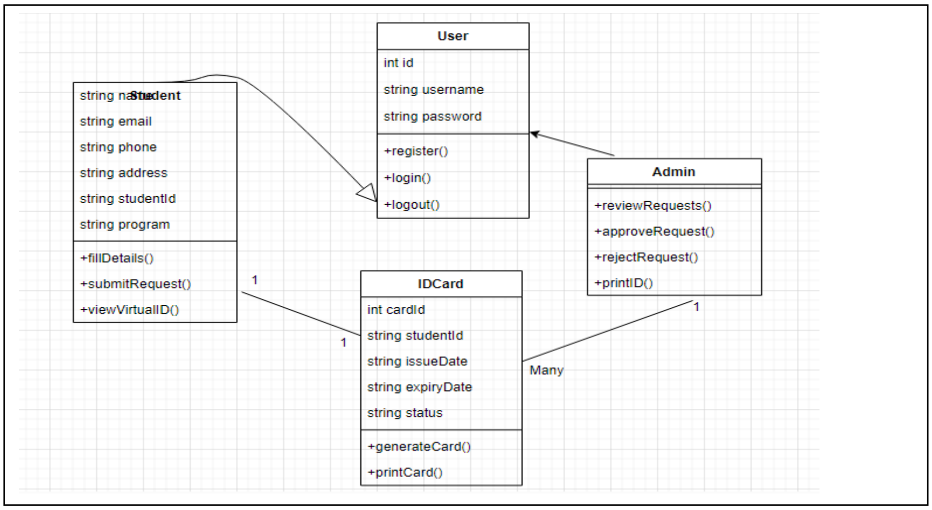


Figure 16: class diagram. Source starUML

3.6.4: Entity Relationship diagram

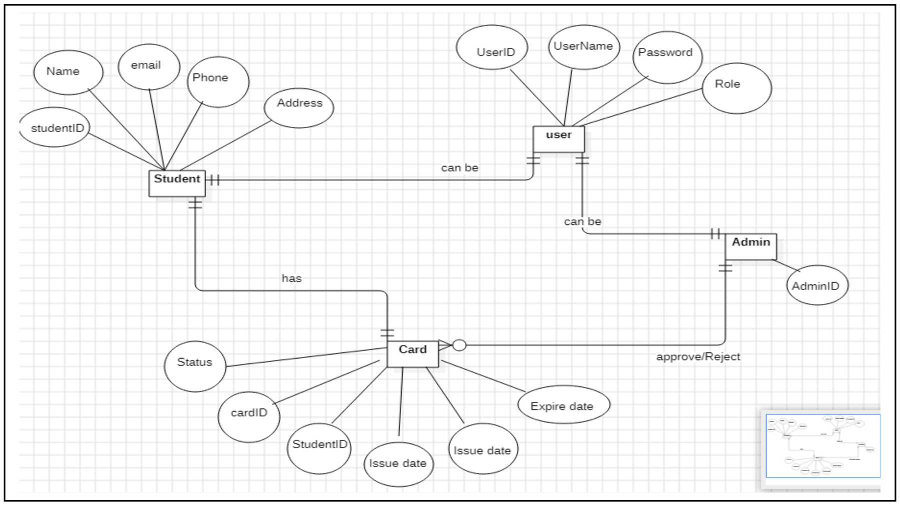


Figure 17: Entity Relationship. Source starUML

CHAPTER FOUR

IMPLIMENTATION, RESULTS, TESTING

4.1 Introduction

For the web-based student ID card generator system being developed for Landmark Metropolitan System, this chapter outlines the processes for system implementation and evaluation. It focuses on documenting the current progress in developing the system, including setting up the development environment, designing the system architecture, and creating the user interface. Since the development process is still in progress, it also describes the methodologies and approaches being used to ensure a reliable and functional system. It also discusses preliminary testing procedures and results to give a preliminary assessment of the system's performance and pinpoint any areas that need more work.

4.2 Implementation.

4.2.1 Development

Hardware Used

- High-performance laptop
  Processor: i5 – 3210m CPU
  RAM: 6144GM RAM
  Storage: 222GB

Software Used

- Development Tools: Visual Studio Code, Star UML, Chrome
- Programming Languages: HTML, CSS, and JavaScript.
- Databases: firebase database

4.2.2 System Architecture

Overall Architecture

Using HTML, CSS, and JavaScript, the overall architecture of the web-based student ID card generator system ensures a lightweight and efficient solution. The client-side components of the system consist of login and registration pages, an ID card creation form, and an admin dashboard. These features let students create accounts, enter their information to create virtual ID cards, and give administrators the ability to review, approve, and print these ID cards.

The system leverages Firebase for backend operations. Firebase Authentication manages user login, registration, and sessions. Firebase Fire store serves as the database for storing user information and ID card details. The ID Card Management Module processes the creation of ID cards The Admin Module provides functionalities for administrators to review and approve ID card requests.

Firebase Fire store handles data storage; it keeps track of user data, information about created ID cards, and administrative user data. CRUD (Create, Read, Update, and Delete) activities are carried out by client-side JavaScript in conjunction with Firebase.

User registration and login procedures, where client-side JavaScript interacts with Firebase Authentication for user management, ID card creation, where students fill out a client-side form, and admin approval, where administrators examine and handle ID card requests through a client-side dashboard, are examples of interactions between components. Using Firebase, HTML, CSS, and JavaScript, this architecture offers an integrated solution for creating and managing student ID cards while ensuring smooth communication and data flow throughout the system.

4.3 Results.

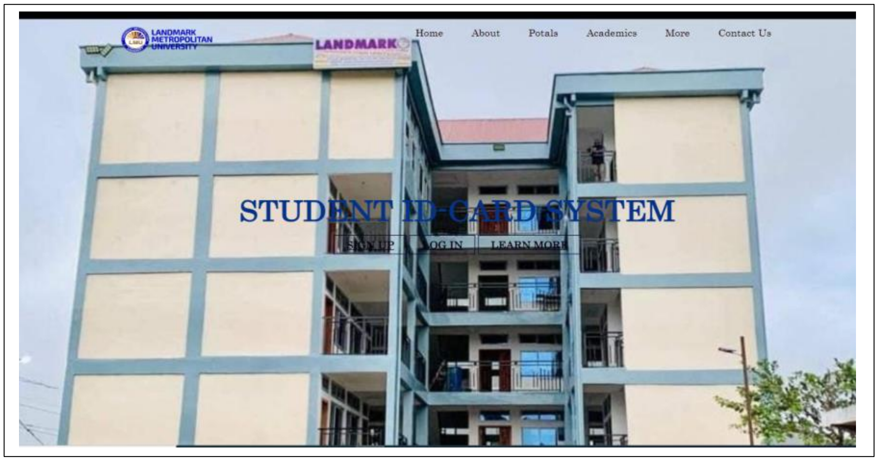
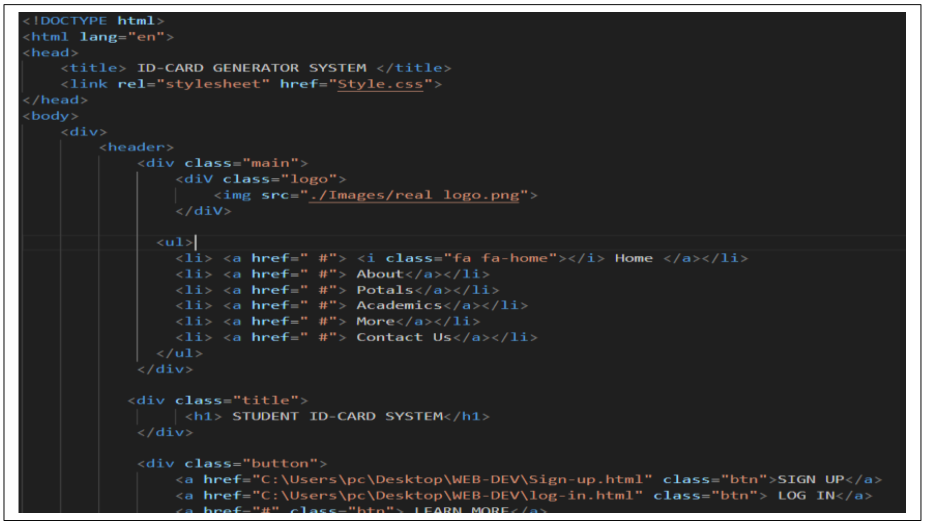


Figure 18: Onboarding page. Source VScode



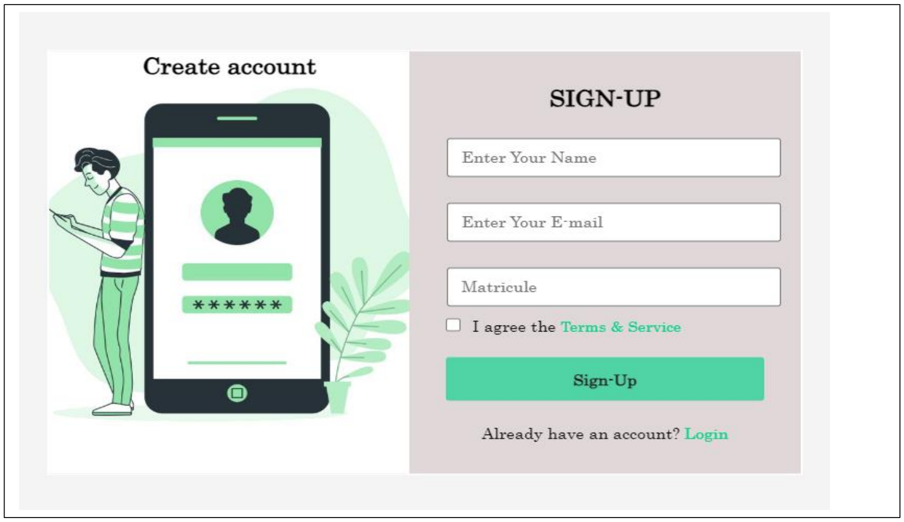
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title> ID-CARD GENERATOR SYSTEM </title>
    <link rel="stylesheet" href="Style.css">
</head>
<body>
    <div>
        <header>
            <div class="main">
                <diV class="logo">
                    <img src="./Images/real logo.png">
                </diV>

              <ul>
                <li> <a href=" #"> <i class="fa fa-home"></i> Home </a></li>
                <li> <a href=" #"> About</a></li>
                <li> <a href=" #"> Potals</a></li>
                <li> <a href=" #"> Academics</a></li>
                <li> <a href=" #"> More</a></li>
                <li> <a href=" #"> Contact Us</a></li>
              </ul>
            </div>

           <div class="title">
                 <h1> STUDENT ID-CARD SYSTEM</h1>
            </div>

            <div class="button">
                <a href="C:\Users\pc\Desktop\WEB-DEV\Sign-up.html" class="btn">SIGN UP</a>
                <a href="C:\Users\pc\Desktop\WEB-DEV\log-in.html" class="btn"> LOG IN</a>
                <a href="#" class="btn"> LEARN MORE</a>
```

***Figure 19: onboarding page code structure. Source** VScode*



Create account

SIGN-UP

Enter Your Name

Enter Your E-mail

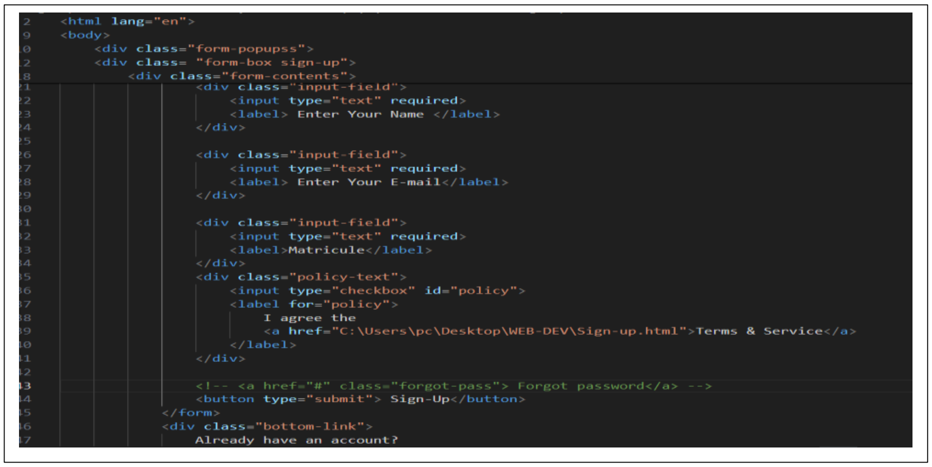
Matricule

□ I agree the Terms & Service

Sign-Up

Already have an account? Login

Figure 20: sign up page. Source VScode



```
<html lang="en">
<body>
    <div class="form-popupss">
    <div class= "form-box sign-up">
        <div class="form-contents">
                <div class="input-field">
                    <input type="text" required>
                    <label> Enter Your Name </label>
                </div>

                <div class="input-field">
                    <input type="text" required>
                    <label> Enter Your E-mail</label>
                </div>

                <div class="input-field">
                    <input type="text" required>
                    <label>Matricule</label>
                </div>
                <div class="policy-text">
                    <input type="checkbox" id="policy">
                    <label for="policy">
                        I agree the
                        <a href="C:\Users\pc\Desktop\WEB-DEV\Sign-up.html">Terms & Service</a>
                    </label>
                </div>

                <!-- <a href="#" class="forgot-pass"> Forgot password</a> -->
                <button type="submit"> Sign-Up</button>
            </form>
            <div class="bottom-link">
                Already have an account?
```

Figure 21: signup code structure. Source VScode

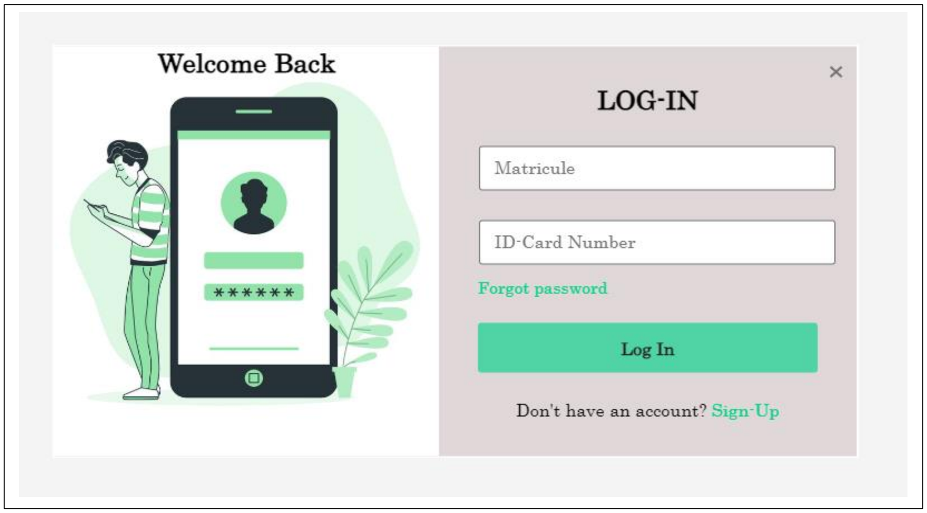
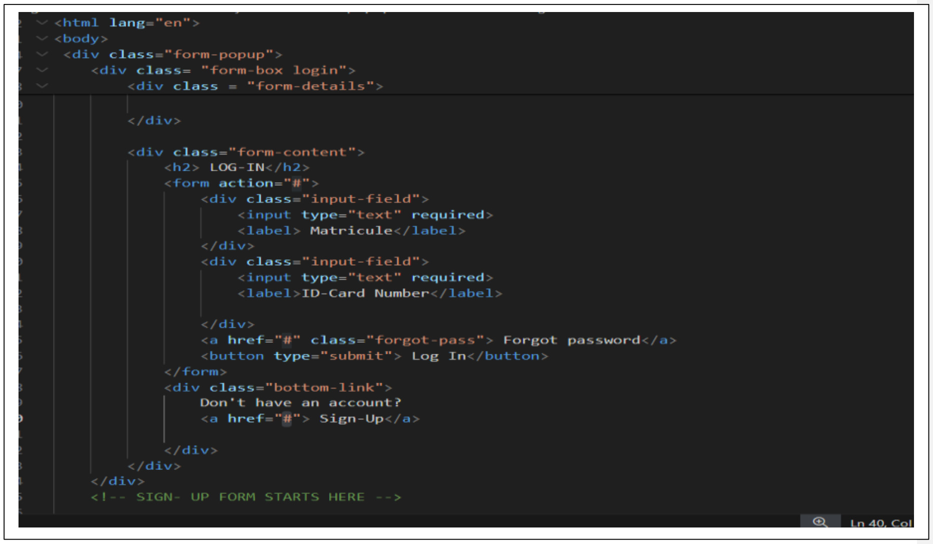


Figure 22: login page. Source VScode

The interface permits students to sign-up to create their accounts in order to the access the card fill form to generate their individual ID cards.



```
<html lang="en">
<body>
 <div class="form-popup">
    <div class= "form-box login">
        <div class = "form-details">

        </div>

        <div class="form-content">
            <h2> LOG-IN</h2>
            <form action="#">
                <div class="input-field">
                    <input type="text" required>
                    <label> Matricule</label>
                </div>
                <div class="input-field">
                    <input type="text" required>
                    <label>ID-Card Number</label>

                </div>
                <a href="#" class="forgot-pass"> Forgot password</a>
                <button type="submit"> Log In</button>
            </form>
            <div class="bottom-link">
                Don't have an account?
                <a href="#"> Sign-Up</a>

            </div>
        </div>
    </div>
    <!-- SIGN- UP FORM STARTS HERE -->
```

Figure 23: log in code structure. Source VScode

1) **The Registration Procedure.**

- **User Accesses Registration Page**: The user navigates to the registration page through the web browser.
- **Fill Registration Form:** The user enters their personal information as required by the form
- **Form Submission:** Upon completing the form, the user clicks the 'Sign-up button'
- **Validation:** The system checks for any errors or missing information. If everything is correct, the information is sent to Firebase Authentication.
- **Account Creation**: Firebase Authentication processes the information and creates a new user account.

- **Successful Registration:** The user is redirected to their individual portal pages with personalized information.

2) Login Process.

- **User Accesses Login Page:** The user navigates to the login page via a browser to the homepage or after registration.
- **Fill Login Form**: The user enters their registered email address and matricule.
- **Form Submission:** The user clicks the 'Login' button.
- **Validation**: The system verifies the credentials against Firebase Authentication.
- **Successful Login**: If the credentials are correct, the user is redirected to their dashboard.
- **Failed Login:** If the credentials are incorrect, the user is prompted with an error message and asked to try again or reset their password.

3) STUDENT PORTAL PAGE

Commented [EM7]: The student dashboard should simply capture the student id card sample. With 2 buttons UPDATE AND VIEW.

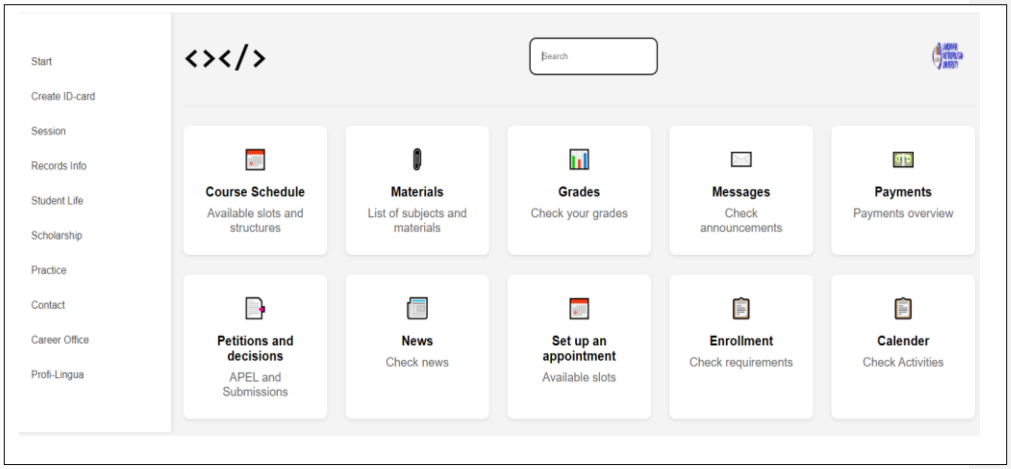
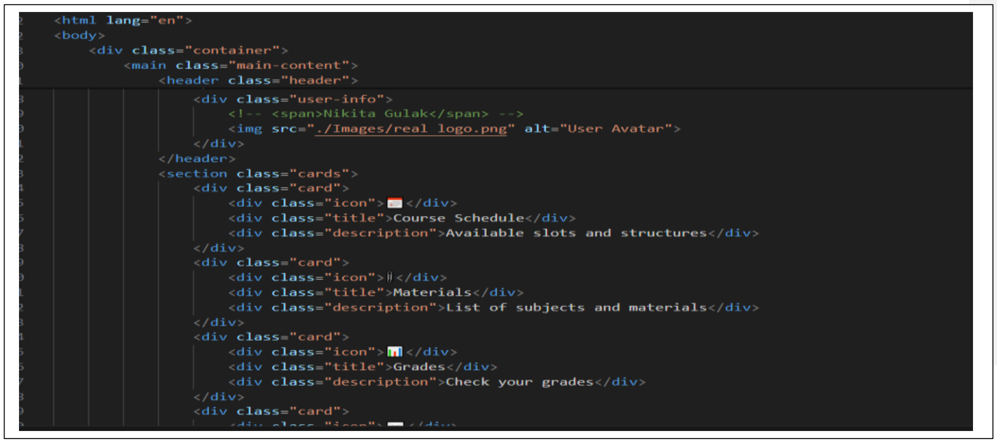


Figure 24: student portal page. Source VScode



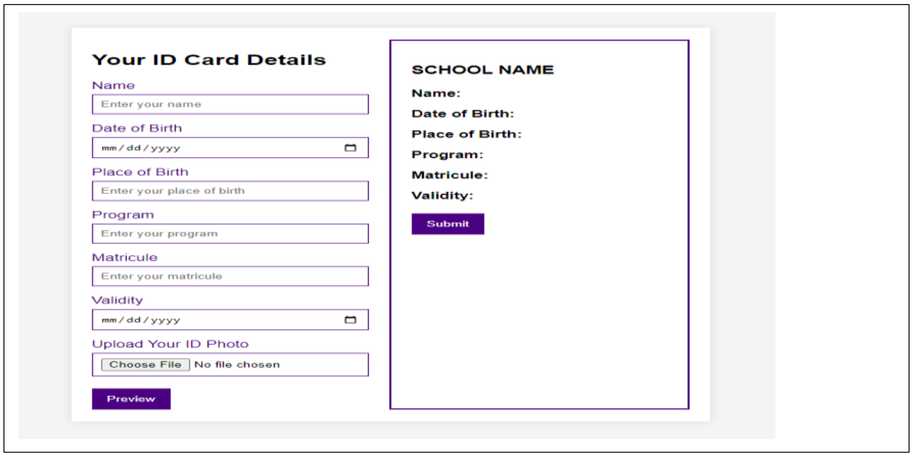
```
<html lang="en">
<body>
    <div class="container">
        <main class="main-content">
            <header class="header">
                <div class="user-info">
                    <!-- <span>Nikita Gulak</span> -->
                    <img src="./Images/real logo.png" alt="User Avatar">
                </div>
            </header>
            <section class="cards">
                <div class="card">
                    <div class="icon">📅</div>
                    <div class="title">Course Schedule</div>
                    <div class="description">Available slots and structures</div>
                </div>
                <div class="card">
                    <div class="icon">📎</div>
                    <div class="title">Materials</div>
                    <div class="description">List of subjects and materials</div>
                </div>
                <div class="card">
                    <div class="icon">📊</div>
                    <div class="title">Grades</div>
                    <div class="description">Check your grades</div>
                </div>
                <div class="card">
```

Figure 25: login code structure. ***Source VScode***

Card creation process.

- Student signs-up/logs-in into his or her portal.
- Student chooses option to create ID card.
- On tapping that option a form is displayed to be filled and submitted.

4) Card form data



Your ID Card Details

Name

Enter your name

Date of Birth

mm / dd / yyyy

Place of Birth

Enter your place of birth

Program

Enter your program

Matricule

Enter your matricule

Validity

mm / dd / yyyy

Upload Your ID Photo

Choose File No file chosen

Preview

SCHOOL NAME

Name:

Date of Birth:

Place of Birth:

Program:

Matricule:

Validity:

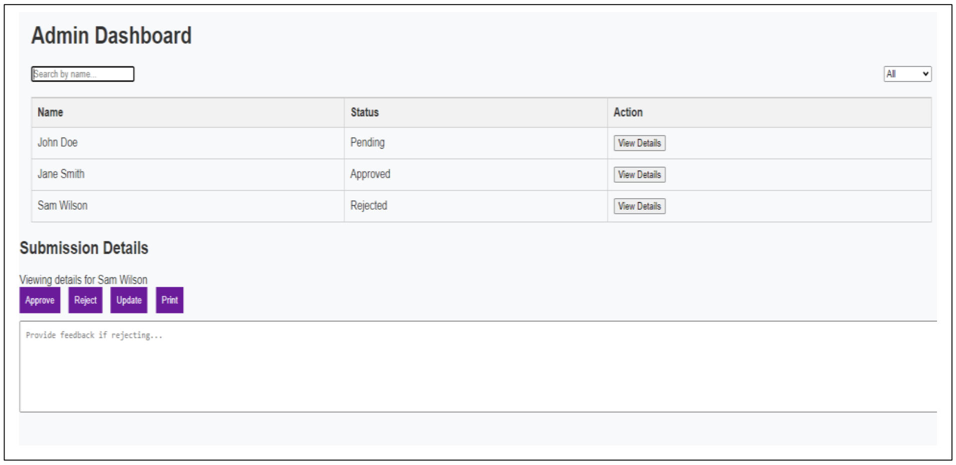
Submit

Figure 26: card form page

The provided HTML and CSS code creates a web page where students can enter their details to generate a preview of their student ID card. The form includes fields for name, date of birth, place of birth, program, matricule, and validity, along with an option to upload a photo. When the

"Preview" button is clicked, the entered details are displayed on a card preview. The "Submit" button is enabled once the preview is generated

5) Admin Dashboard



Admin Dashboard

Search by name...

All

Name	Status	Action
John Doe	Pending	View Details
Jane Smith	Approved	View Details
Sam Wilson	Rejected	View Details

Submission Details

Viewing details for Sam Wilson

Provide feedback if rejecting...

Figure 27: admin dashboard. Source VScode

Managing student ID card submissions falls under the purview of the administrator. This includes looking over each submission's information, approving or rejecting them, giving comments in the event that a submission is turned down, revising submissions as needed, and producing the ID cards that have been approved. The administrator searches and filters submissions by name or status using a dashboard. The administrator can access each submission for a thorough examination, enabling them to make defensible decisions. Prior to printing the actual ID cards, the administrator verifies the information's authenticity and accuracy.

4.4 System Testing Strategies

4.1 Unit Testing

Unit testing focuses on individual components or functions within the web-based student ID card generator system to ensure they work as expected. For this application, unit tests will verify that each part of the system, such as the form submission, data validation, and display functionalities, operates correctly in isolation. This includes checking if the form fields correctly capture input, if the preview function accurately displays the entered data, and if the data upload process works seamlessly.

1) **First Test Case1: Validation of Form Fields**

Test Title: Empty Fields Test

Goal: Make sure that form fields cannot be sent with empty values.

Prerequisites: The form is filled up.

Actions:

Fill out every field on the form with nothing.

Try submitting the form.

Anticipated Outcome: The attempt to submit the form is unsuccessful, and relevant error messages appear.

2) Test Case 2: Valid Data Entry

Test Name: Test Valid Data Entry

Objective: Ensure that valid data can be entered into the form fields.

Preconditions: The form is loaded.

Steps:

Enter valid data into all form fields (e.g., name, student ID, department, etc.).

Submit the form.

Expected Result: The form submission succeeds, and the data is correctly displayed in the preview section.

3) Test Case 3: Data Preview Update

Test Name: Test Data Preview Update

Objective: Ensure that the preview updates correctly when form data is entered or changed.

Preconditions: The form is loaded.

Steps:

Enter data into the form fields.

Observe the preview section.

Expected Result: The preview section updates in real-time to reflect the entered data.

4) Test Case 4: Admin Approve Functionality

Test Name: Test Admin Approve

Objective: Ensure that the admin can approve a submitted ID card request.

Preconditions: A submitted ID card request is present in the admin dashboard.

Steps:

Log in as an admin.

Select a submitted ID card request.

Click the "Approve" button.

Expected Result: The ID card request status is updated to "Approved," and the user is notified.

4.2 Integration Testing

Integration testing examines the interaction between different modules of the system to ensure they work together harmoniously. In this context, integration tests will validate the seamless interaction between the form submission component, the preview display, and the backend (Firebase) database. This involves testing the entire workflow from user input, data submission, and storage in the database to displaying the ID card details. These tests will help identify issues that arise from the interaction between modules.

1) Test Case 1: User Registration and Login Integration

Test Name: TestUserRegistrationLoginIntegration

Objective: Ensure that the user registration and login functionalities work seamlessly together.

Preconditions: The registration and login pages are accessible.

Steps:

Navigate to the registration page.

Register a new user with valid details.

Log out.

Navigate to the login page.

Log in with the newly registered user credentials.

Expected Result: The user is successfully registered and logged in, and redirected to the main dashboard.

2) Test Case 4: Data Preview and Form Interaction Integration

Test Name: TestDataPreviewFormInteractionIntegration

Objective: Ensure that data entered into the form updates the preview section in real-time.

Preconditions: The form page is loaded.

Steps:

Enter data into the form fields (e.g., name, student ID, department).

Observe the preview section.

Expected Result: The preview section updates in real-time to reflect the entered data.

4.3 System Testing

System testing evaluates the entire system's functionality and performance as a whole. This level of testing ensures that the web-based student ID card generator system meets all specified requirements. System tests will include end-to-end scenarios such as user registration, form submission, data preview, admin approval/rejection, and ID card printing. It will also check non-functional requirements like usability, performance, and security. This comprehensive testing phase ensures that the system is ready for deployment and use by actual users. Although this part of the testing phase is still not fully implemented

1) Test Case 4: Admin Approval Workflow

Test Name: TestAdminApprovalWorkflow

Objective: Verify the complete workflow of an admin approving a submitted form.

Preconditions: A submitted ID card request is present in the admin dashboard.

Steps:

Log in as an admin.

Select a submitted ID card request.

Click the "Approve" button.

Verify that the status of the request is updated to "Approved."

Log out and log in as the user who submitted the request.

Check the notification or status update.

Expected Result: The ID card request status is updated to "Approved," and the user is notified.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION.

5.1 Discussions.

The analysis of the results from the creation and application of the Student ID Card Management System is covered in detail in this section. It looks at the effectiveness of the system, user happiness, data correctness, and project-related difficulties. It also does a comparison between the new system and the manual method from before to pinpoint any improvements and gaps that still exist.

5.1.1 Analysis of Findings

Efficiency Improvements

The efficiency of the ID card issuance process was successfully tested and will significantly improve the Student ID Card Management System .Prior to the implementation of the system, administrative staff had to devote a significant amount of time and effort to a labor-intensive manual process. Many of these operations are automated by the new system, which makes data entry, processing, and card printing faster. In addition to expediting the issuance process, this automation has freed up staff time for other crucial administrative duties.

User Satisfaction

Testers taking part in the testing phase have provided extremely good early comments. People have complimented the online application process' simplicity and user-friendliness. It has been especially well-received since forms can be completed at any time, as this eliminates the need for in-person visits to administrative offices.

Data Accuracy

The improvement in distributing the burden of an excessive number of input actions from a single administrator to each student is one of the most significant benefits that were noted during testing. Because of this feature, there are a lot fewer errors, which results in more accurate and dependable records. The centralized database lowers the possibility of inconsistencies by guaranteeing that all data is readily available and consistent.

5.1.2 Challenges Faced

Security.

It is essential to make sure the system is safe from several kinds of attacks (such as SQL injection, XSS, and CSRF). Numerous possible vulnerabilities were found and most fixed during testing. To preserve the integrity of the system and safeguard sensitive student data, ongoing security evaluations are required.

Design and Architecture.

It was proven difficult to select the appropriate technological stack and to create a system that is manageable, scalable, and capable of operating in a variety of scenarios. Future additions and linkages with other educational systems must be supported by the architectural design. To achieve these objectives, the ongoing development process includes continual evaluation and potential redesigns.

Development Process

Maintaining high code quality, including readability, efficiency, and security, has been a significant focus with slight difficulties. During development I faced challenges related to version control and managing code conflicts.

5.1.3 Comparison with Previous Systems.

The new Student ID Card Management System exhibits significant gains over the prior manual procedure, according to the testing phase results. Students and staff frequently became frustrated with the manual approach due to its proneness to errors and delays. On the other hand, once completely implemented, the new system's real-time processing and updates should guarantee that ID cards are issued promptly and precisely. The reliability of student records during testing has already improved thanks to the automated data validation tools.

There is still room for development, though. Even if the new system solves a lot of the manual process's inefficiencies, it might still benefit from further features. A more seamless user experience might be achieved, for example, by linking the system with other school management software and mobile applications. Furthermore, as technology and user needs change, ongoing monitoring and modifications will be required to guarantee the system's security and efficacy.

5.2 Conclusion

To conclude, with increased efficiency and data accuracy compared to the prior manual procedure, the Student ID Card Management System is a major advance. Numerous manual procedures are automated by the system, which reduces administrative workload and saves time. Via the online application process and user-friendly interface.

In summary, the Student ID Card Management System's successful development and early testing point to a promising future for a more effective and user-friendly ID card issuance procedure. The system has the potential to significantly increase efficiency and user satisfaction and become an essential component of the school's administrative framework with further efforts to address issues and incorporate feedback.

5.3 Recommendations and Perspectives for Further Study

Based on the findings from the development and testing of the Student ID Card Management System, several recommendations can be made to further enhance the system and ensure its successful implementation and long-term effectiveness.

Enhanced Security Measures: Strong security measures must be put in place since securing sensitive student data is so important. To make sure the system's security is robust, regular security audits, penetration tests, and upgrades to fix vulnerabilities should be carried out.

Integration with Other Systems: Think about combining the system with mobile applications and school management software to get the most out of it. Through this integration, administrative procedures can be streamlined, data consistency can be enhanced, and user experience can be made smooth.

Performance Monitoring: Tracking the effectiveness and efficiency of the system can be facilitated by putting in place a strong performance monitoring system. Monitoring key performance indicators on a regular basis will reveal system performance and point up areas for development.

REFERENCES

Commented [EM8]: USE APA(MUST)

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology, 2*(5), 204-212.

Brooks, F. P. (1995). *The mythical man-month: Essays on software engineering* (20th Anniversary Edition). Addison-Wesley Professional.

Burrell, G., & Morgan, G. (2022). *Sociological paradigms and organizational analysis: Elements of the sociology of corporate life*. Routledge.

Cockburn, A. (2001). *Writing effective use cases*. Addison-Wesley.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approach*. Sage publications.

DeMarco, T., & Lister, T. (2013). *Peopleware: Productive projects and teams* (3rd ed.). Addison-Wesley.

Dennis, A., Wixom, B. H., & Roth, R. M. (2018). *Systems Analysis and Design* (6th ed.). Wiley.

Erickson, T., & Kellogg, W. A. (2000). Social translucence: An approach to designing systems that support social processes. *ACM Transactions on Computer-Human Interaction (TOCHI, 7*(1), 59-83. https://doi.org/10.1145/344949.345004

Firebase Documentation. (2023). *Firebase Authentication*. Retrieved from https://firebase.google.com/docs/auth

Firebase Documentation. (2023). *Firebase Realtime Database*. Retrieved from https://firebase.google.com/docs/database

Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley Professional.

Glesne, C. (2015). *Becoming Qualitative Researchers: An Introduction* (5th ed.). Pearson.

Hoffer, J. A., Ramesh, V., & Topi, H. (2013). *Modern systems analysis and design* (7th ed.). Pearson.

Johnson, D. W., & Johnson, F. P. (2017). *Joining together: Group theory and group skills* (12th ed.). Pearson.

Maguire, M., & Bevan, N. (2002). User requirements analysis. *ACM SIGCHI Proceedings of the 10th International Workshop on Internationalization of Products and Systems*, 133-152.

Mozilla Developer Network. (2023). *CSS*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS

Mozilla Developer Network. (2023). *HTML*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering*, 35-46.