# CERTIFICATION

# DEDICATION

# ACKNOWLEDGEMENT

# ABSTRACT

# LIST OF ABBREVIATIONS

Contents

[CERTIFICATION 1](#_Toc167253886)

[DEDICATION 2](#_Toc167253887)

[ACKNOWLEDGEMENT 3](#_Toc167253888)

[ABSTRACT 4](#_Toc167253889)

[LIST OF ABBREVIATIONS 5](#_Toc167253890)

[LIST OF FIGURES. 7](#_Toc167253891)

[*Figure 05……………………………….. badge pass identity manager software interface.* 7](#_Toc167253892)

[**CHAPTER ONE** 8](#_Toc167253893)

[GENERAL INTRODUCTION. 8](#_Toc167253894)

[**Background of the study.** 8](#_Toc167253895)

[CANVAS. 9](#_Toc167253896)

[KEY FEATURES. 10](#_Toc167253897)

[Database and Storage. 12](#_Toc167253898)

[2) **BadgePassID** 13](#_Toc167253899)

[**BADGEPASS ID MAKER SOFTWARE** 14](#_Toc167253900)

[**Key-Features.** 14](#_Toc167253901)

[***Figure 05. badge pass identity manager software interface.*** 14](#_Toc167253902)

[**System Architecture.** 14](#_Toc167253903)

[**OBJECTIVES OF THIS PROJECT.** 15](#_Toc167253904)

[**SPECIFIC OBJECTIVES.** 15](#_Toc167253905)

[**SIGNIFICANCE OF THE STUDY.** 17](#_Toc167253906)

# LIST OF FIGURES.

*Fig 01…………………………………………………………… Canvas Design software.*

*Fig 01………………………………………………………….. Canvas Design Interfaces.*

*Fig 03………………………………………………………..Badge-pass Design Interfaces.*

## *Figure 05……………………………….. badge pass identity manager software interface.*

## CHAPTER ONE

## 1.1 GENERAL 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, and alter card templates. 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.

**1.3 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 DEFFINITION 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 od 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 ORGANISATION 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**  
A 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**

**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

**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, Firestore 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:** Student ID card Application with SMS verification message

**SYSTEM:** Acase study of Gretsa University

**AUTHOR:** Mokaya Kebande Lameck

**TECHNOLOGY USED:**

To develop this project front-end tool as VISUAL STUDIO 2005, HTML: Hypertext Markup Language, Preprocessor CSS: Cascading Style Sheet and Java Script were used & back-end tool as relational database (MySQL) has been used, PHP: Hypertext Preprocessor.Other technologies included protocols (a protocol is a set rules that correlate procedures for formatting and processing data). they include HTTP: Hypertext Transfer Protocol. HTTPS: Hypertext Transfer Protocol.

**STRENGHTS OF THE SYSTEM:**

* increases security by using SMS to confirm student identity.
* uses JavaScript, HTML, CSS, and Visual Studio 2005 to create an interactive user interface.
* uses MySQL and PHP to create a dependable and expandable back-end system.
* guarantees accurate, non-redundant data storage in the MySQL database.
* facilitates the effective administration of big student datasets.
* use HTTP and HTTPS protocols to transmit data securely.
* reduces data loss and corruption with a strong RDBMS.
* makes it easier to manage and update student information.
* provides an additional degree of protection via SMS verification.
* ensures best practices by adhering to IEEE rules for software requirement specifications.

**LIMITATIONS:**

* Complex Setup: Configuration calls for technical know-how.
* Resource-intensive: Requires high-performance software and hardware.
* Limited Automation: There isn't any automatic verification or data entry.
* Dependency on Internet Connectivity: SMS and data transmission need a reliable internet connection.
* Issues with Scalability: Could have trouble handling a higher student load.
* User Training: To be used effectively, training is necessary.
* Data privacy concerns: The dangers of working with private information.
* Restricted Support for Digital IDs: Emphasizes paper ID cards.
* Lack of Integration: Does not easily integrate with other systems at the university.

**RECOMMENDATIONS FOR FUTURE WORK:**

* Automate data entry and verification procedures by implementing automation.
* Strengthen the Integration Connect to other information systems within the university.
* Presenting Digital Identity Provide ID cards in digital format.
* Offer Offline Functionality: Enable certain functions to work offline.

**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 areseveral 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 thatemphasizes the use of objects, which are instances of classes, to represent andmanipulate data. OOP can be used in the development of an ID card generatorsystem using PHP to improve the system's modularity,reusability,and maintainability.

In an ID card generator system, OOP can be used to represent different entitiessuch as the user, the card, and the database. The following is an example of howOOP 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 suchas name, email, and password, and methods such as login and logout. The Userclass 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 classcan be used for generating and printing ID cards.

**3) database class**

The Database class represents the database used by the system. It canhave methods such as connect and query, which can be used for connecting tothe database and executing SQL queries. The Database class can be used forstoring 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 cellphones 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 centralised personalisation centre.
* 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:**

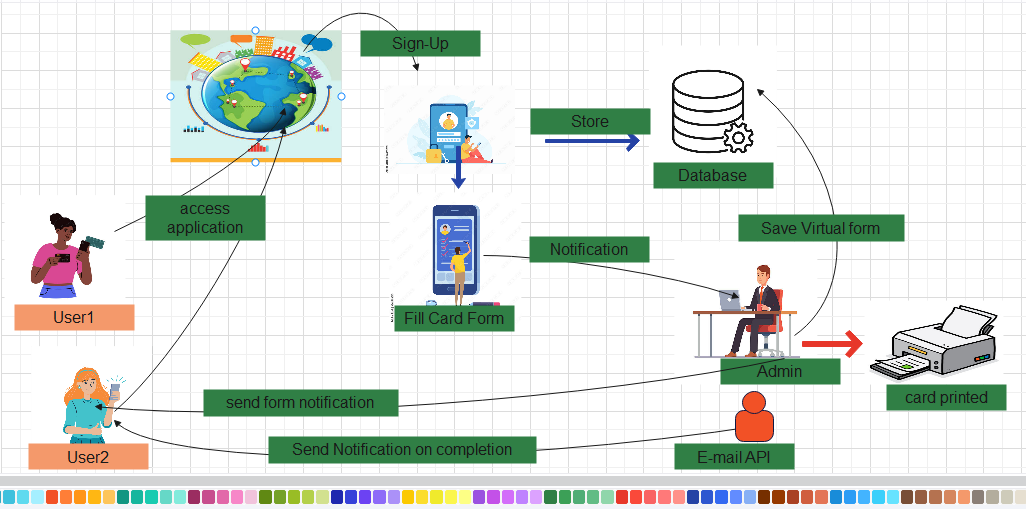
* Dependency on SELS: Real-time data updates may be limited if communication is solely started by SELS.
* Complicated Implementation: It could be difficult to integrate and guarantee compatibility with current systems.
* Limited Data Source: All data is sourced from the Student Information System at 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.

**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.**



**Fig 00: 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 Firestore
* Collections

1. Users: Preserves user data.
2. ID Cards: Keeps card information stored.
3. 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 Firestore: 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.

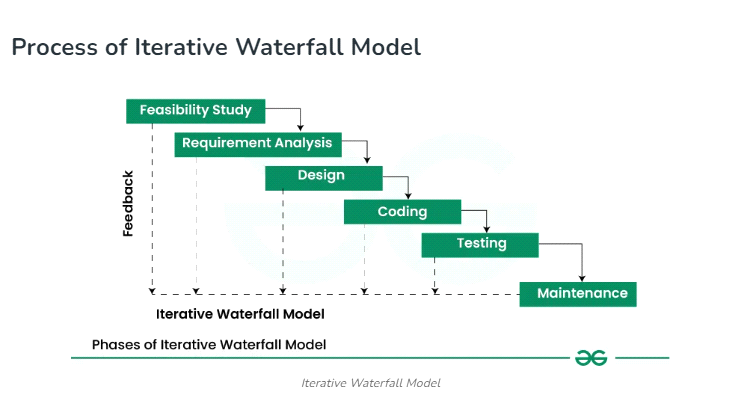
**CHAPTER THREE**

**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:**

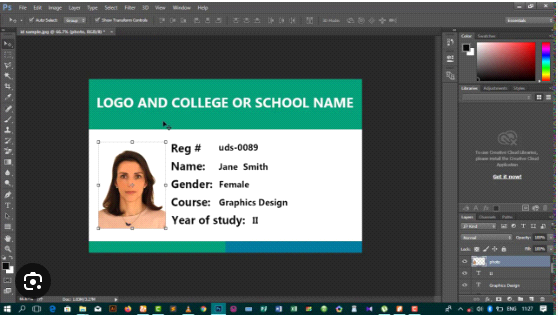
**Iterative Water Fall Model:** The technique selected for this project is a hybrid approach that combines the iterative feedback mechanisms of Agile practices with the structured phases of the Waterfall model. Throughout the development process, this strategy permits flexibility and ongoing improvement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



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 customisable, 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.



**A) Detailed Workflow of the ID Card Generation System.**

**1) Data colloection 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: the 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 appopraitely 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 cheched 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.**

* Hardware Cost: 1 high performance laptop/ 140.000frs
* Hosting: Free Hosting
* Software Cost:
* Development Tools:
* Visual Studio Code: free
* Star UML (Unified Modelling Language): free
* Developer cost: 1 full time developer
* Internet: 500 - 1,000frs.

**3.5.4 PROJECT SCHEDULE.**

|  |  |
| --- | --- |
| Weeks | Activities |
| Week1-2 | Project initiation and planning  Define Project Scope, Gather Requirements  Create project plan and Timeline, Set up project repository and tools. |
| Week3-4 | Design the system Architecture, Database Schema design. |
| Week5-7 | Develop the landing page, Implement Responsive design, Style components using CSS |
| Week8 | Set up backend server and database.  Integrate Frontend With backend API endpoints.  Develop Restful API for user management (registration, login), restaurant management.  Implement ordering Module.  Develop reservation system, Set up notifications for order updates and promotions. |

Fig. Project Schedule.

**3.5.5 USE CASE ANALYSIS.**

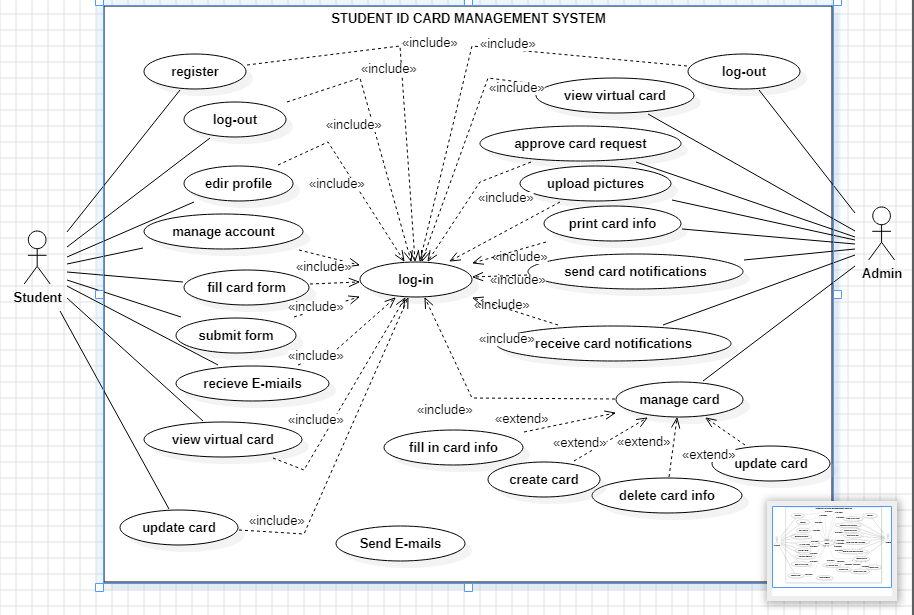


Figure.

**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**

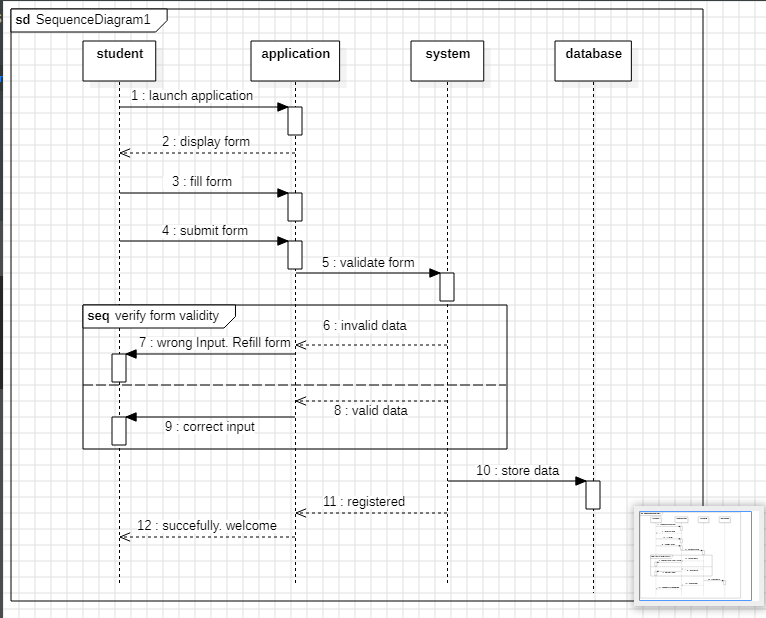


Fig.

Fig. sign-up sequence diagram.

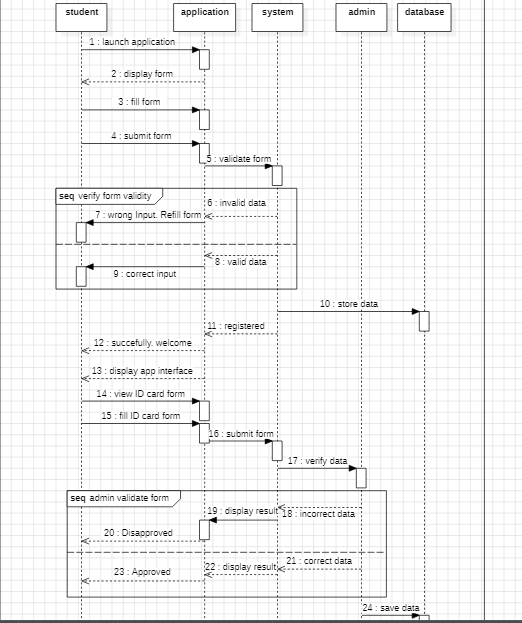


Figure. Fill form sequence diagram

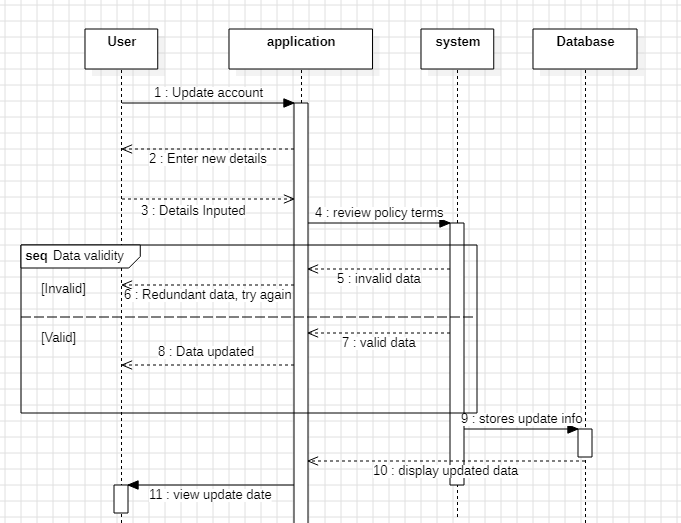
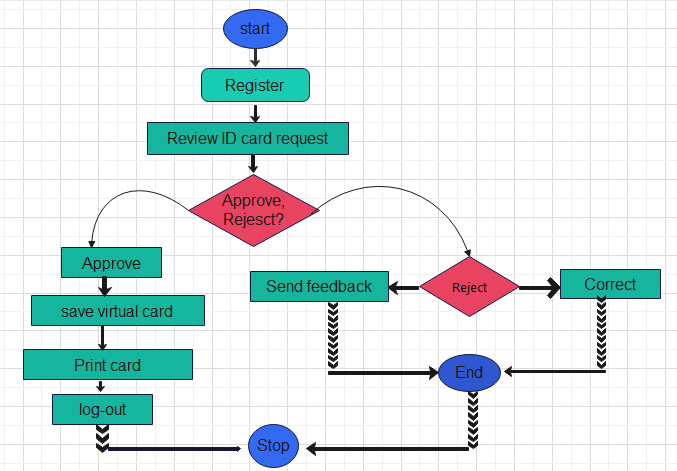
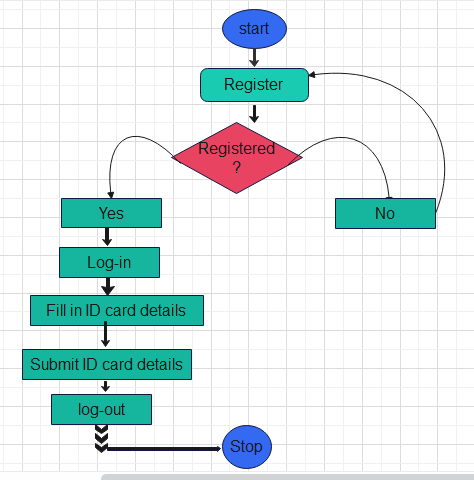


Fig. Update data sequence diagram.

**3.5.7 Activity diagram**. It is a type of UML diagram that represents the flow of activities or actions within a system or process. It is used to model the dynamic aspects of a system by showing the sequence of activities, decision points, and parallel processes.



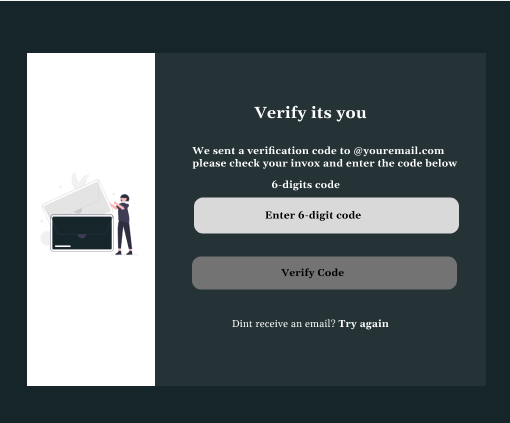


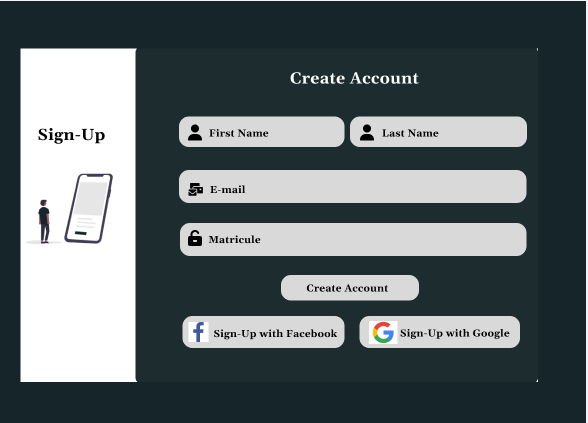
**Fig. Activity Diagram.**

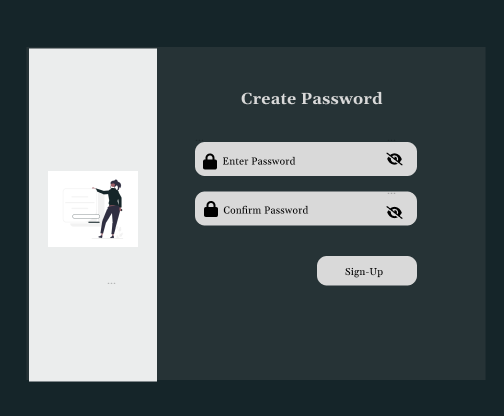
**3.6 System Design.**

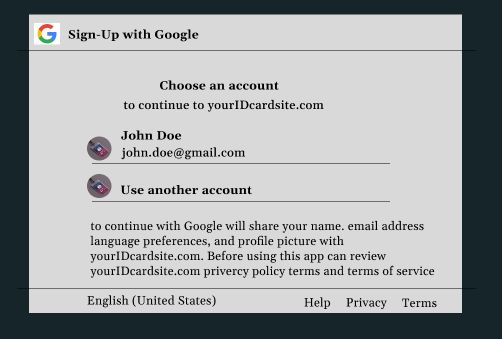


**Fig. H**

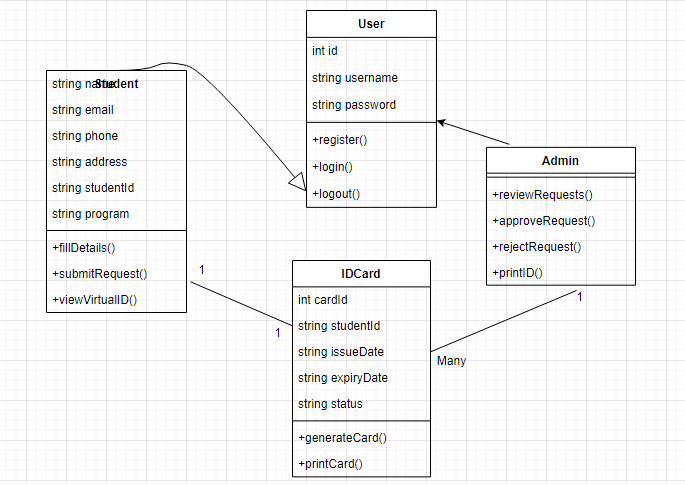








**3.6.2 Class Diagram.**



**Ffigure.drt**

**3.6.4: Entity Relationship diagram**

