DOI: http://dx.doi.org/10.26483/ijarcs.v15i5.7131

Volume 15, No. 5, September-October 2024

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

DEVELOPMENT OF AN EFFICIENT STUDENT HALL REGISTRATION AND EXEAT SYSTEM

Akande Oyebola, Adekola Olubukola, Ezenagu Okwuchukwu Divine, Adegbola Adesoji, Adewuyi Joseph Oluwaseyi, Wobo Michellle Eruchi, Ehindero Joseph Olamide
Computer Science Department
School of Computing, Babcock University,
Ilishan Remo, Ogun State Nigeria

Abstract: The current student's hall registration and exeat system in private owned Nigerian higher institutions uses manual or paper-based approach. This method is cumbersome, susceptible to mistakes, and poses security concerns. The aim of this project is to develop a web-based student hall registration and exeat system (SRES) that enhances student housing management by automating and streamlining the processes for registration and exeat processes using scrum process model. Thus, students can register and request for exeat online by submitting necessary documents. The request will be reviewed by authorized school administrator who will either accept or reject the exeat based on whether reasons and documents supplied are satisfactory or not. The SRES implements a comprehensive data management system, making physical records obsolete. By consolidating student information, the university gains efficient oversight of residency status, streamlining administrative processes and enhancing security. Moreover, the SRES fosters a better learning atmosphere by easing the hall registration procedures and minimizing delays in exeat approvals, allowing students to focus on their academics.

Keywords: scrum process model, student hall registration and exeat system, data management system, web-based, administrator

INTRODAUCTION

Student housing management in universities has changed over the years. Previously, universities offered limited or no on-campus accommodation, therefore students had no other choice but to live with family or friends in the surrounding area. But as the need for centralized student housing grew, universities built on-campus hostels and these hostels were managed manually. These methods, while functional, presented several limitations. Time-consuming tasks like room allocation, registration, and record-keeping were prone to human error [1]. Therefore, an effective student housing system was essential for a well-run university to make efficient the student registration processes and streamlined exeat (leave of absence) procedures.

The rise of computers in recent decades paved the way for web-based solutions for hostel management [2] and many universities adopted software solutions to streamline various aspects of hostel operations, including tasks like booking, scheduling, visitor management, registrations, checkouts, financial transactions, and record-keeping [3].

An organization or institution cannot be competitive or successful in today's business landscape without a computer system that makes operations efficient and easy to run. Therefore, it is evident that a computerized system is necessary in institutions and organizations to make operations efficient and improve the customer experience.

The development of a web-based student hall registration and exeat system serves as a foundational framework for efficiently managing hall registration at the beginning of each semester or session, as well as exiting activities within university campus halls of residence. Its primary objective is to facilitate seamless control and automation of various processes related to student accommodation, entry, and exit within the halls. Unlike general hostel management software that covers a wide range of hostel-related operations, this research specifically analyzes the processes involved in registering students and handling their requests to leave campus.

This work used scrum [4] software development methodology which is an iterative and incremental process model in Agile framework. Scrum emphasizes flexibility, transparency, and delivering functional software in short cycles (sprints) to allow for adjustments based on real-time feedback from stakeholders.

These following sections discussed review of related works, methodology including use of scrum process model, implementation and Gap Analysis.

REVIEW OF RELATED WORK

[5] outlined principles for designing an exeat system and provided insights into the application and approval process used within an institution. The author suggested using biometric verification to prevent unauthorized exits and emphasized the importance of biometric registration for all students. The system aimed to generate real-time estimates of the total number of students on and off campus. It proposed efficient exeat processing methods and a secure authorization process for individual exits. Additionally, student residence and exeat details were registered to facilitate campus exits.

[3], developed a web-based system for management of a hotel and its day-to-day activities. The technologies used to design and implement this system were: Microsoft Visual Studio, Visual Basic.net, and .NET framework Platform Architecture.

[6] developed an Online Hostel Management System to tackle the issues linked with manually storing hostel information. This system aimed to automate room allocation and aid hostel administrators in managing records of both current and past students. Developed using an Apache server, SQL for database management, PHP for programming, and

HTML or XML for the user interface, this system offered a digital solution for effective hostel management.

[7] developed a Students Exeat Monitoring System Using Fingerprint Biometric Authentication and Mobile Short Message Service. The system solved the problem of impersonation, inaccuracy and insecurity of paper-based exeat system used in majority of privately owned universities in Nigeria. The biometrics overcome the problem of impersonation while mobile short message sent to parents r guardians solved the problem of insecurity by making the guardians or parent know the current location of their ward.

III. METHODOLOGY

The following steps were followed to develop the SRES.

1) USER STORY

First all the users of the system were listed and what they are expecting the system will do or offer them were captured in a user story table. This is also called the functional requirements. Table I below is a sample of the user story.

Table 1: Sample of User Story

As a ...	I want to be able to ...	so that ....
student	register or sign up into the system	I can have access to functionalities of the system regarding student's
student	login into the system	I can register for hall, request for
student	view my registration status	I may know if my registration has
administrator	login	Use all the system functionalities
administrator	manage users	I may approve authorize users

2) USE-CASE DIAGRAM

The user story table was used as input into developing use case diagrams, for student's hall registration and student's exeat request.

Use case diagrams are fundamental tools in software development and system analysis, as described by [6]. These diagrams provide a visual representation of a system's functional requirements by focusing on the relationship between different users (actors) and what they do with the system (cases or goals). The goals are the action performed by each actor. For example, in figure I actor student can login into the system, can register for hall of residence, select room and check room availability. To draw the use case diagram, a table containing users and the expected actions they want to perform on the system is shown in Table II. The following figures I and II showed two use cases, one for the student hall registration and the other for the exeat system. The actors in student hall registration are: student and hall administrator. The goals relating with each actor are in the oval shaped cases.

Table II: Actors and Actions

Actor/User	Action
Student	Login, which includes verifying their password
	Register by selecting residence of choice
	Select Room and Block
	View registration status
	Request for exeat which includes weekly, or weekend
	Request for exeat which includes weekly, or weekend
	View exeat status
Hall Administrator	Login, which includes verifying their password.
	View student registration status
	Sign in and sign out students
	Manage student information
	View registered student status
	View Exeat requests
	Manage exeat requests by Approving or denying exeat

Figure I: Use Case for Students Hall Registration

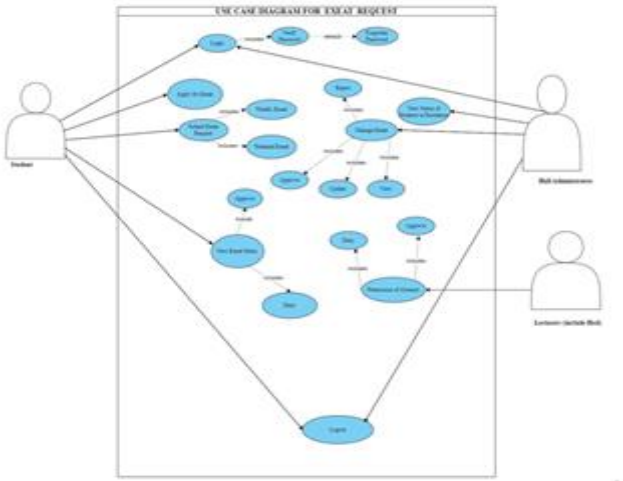

Figure II: Use case of Student's Exact Request

3) SEQUENCE DIAGRAM

Sequence diagrams, complementing use case diagrams, are essential tools in software development and system analysis [7]. These diagrams visually represent the interactions between objects in a system along a timeline, focusing on the message exchange [8]. Typically triggered by an actor (e.g., user) initiating an event, the sequence diagram illustrates how objects collaborate by exchanging messages in a specific order to complete a task [9]. This visual representation of message flows offers a clear understanding of object interactions and their sequence within a system. Figure III is the sequence diagram for student hall registration, while figure IV is the sequence diagram for the exeat request.

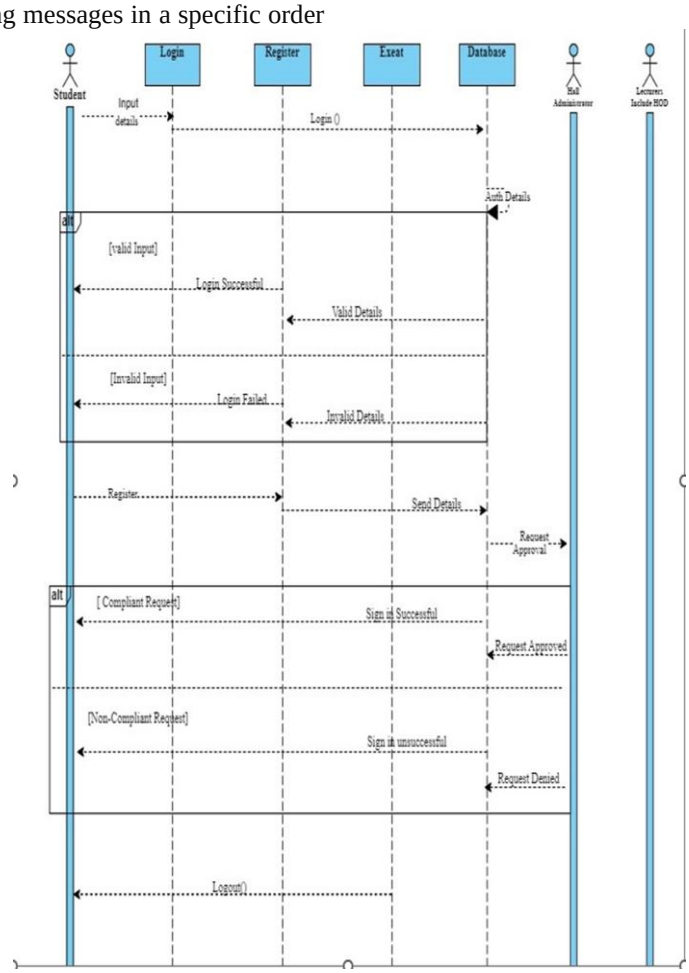

Figure III: Sequence Diagram for Student Hall Registration

In the Sequence diagram above students start the hall registration process by submitting a request with their login information. This information is checked to make sure it's correct. If the student logs in successfully, they can choose their preferred hall based on what's available. The request is then sent to the administrator for approval or rejection. Once the administrator approves, the system confirms that the student has been successfully registered and grants them access to the chosen hall.

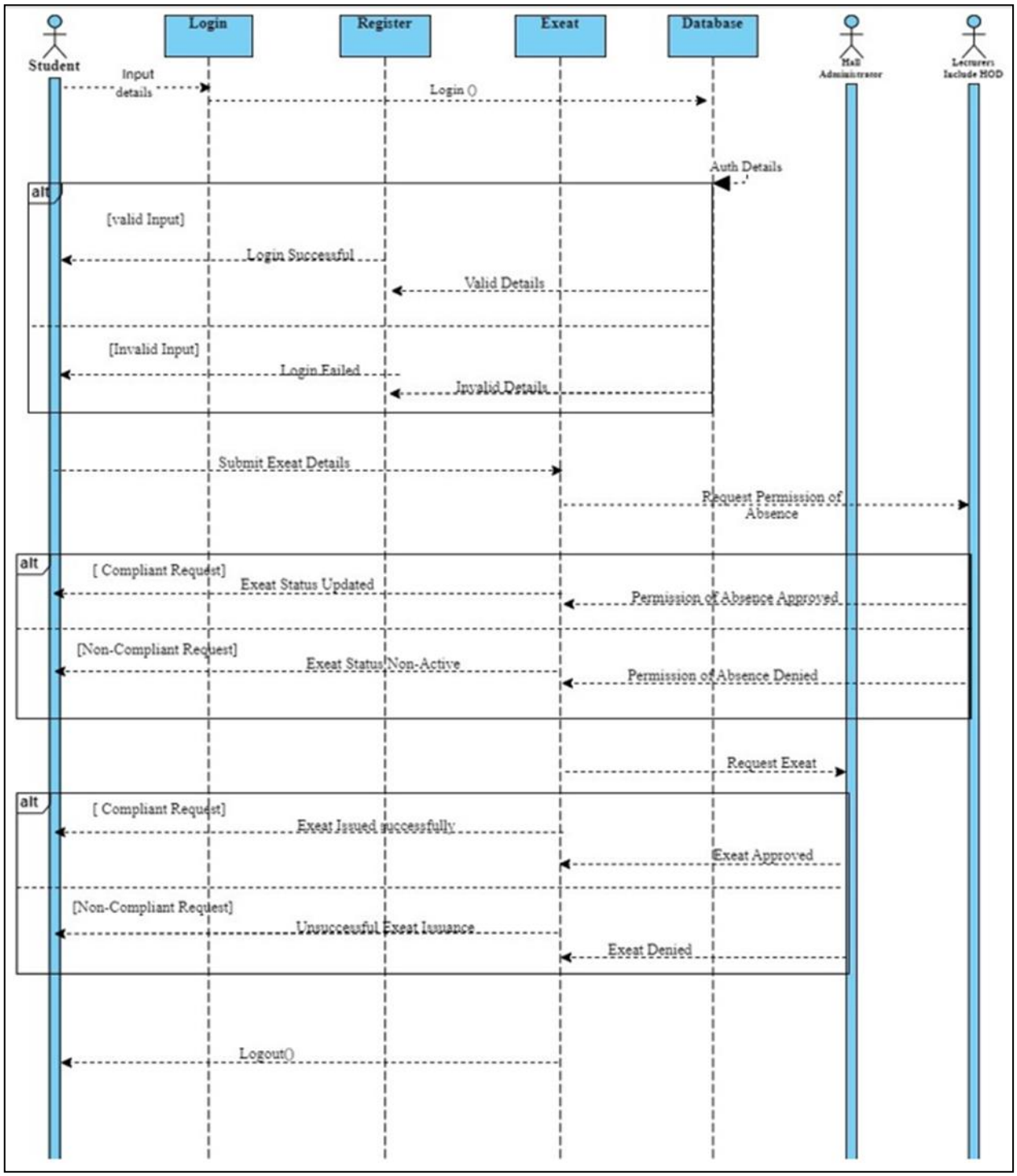

Figure IV: Sequence Diagram for The Exeat Request

4) ACTIVITY DIAGRAM

An activity diagram is a visual representation within Unified Modeling Language (UML) used to illustrate the flow of activities or workflows within a system or process. It is particularly useful for modeling the dynamic aspects of a system, focusing on the sequence of actions, decision points, parallel activities, and the overall flow of control. The activity diagram contains circular nodes (start and end point) which indicate the beginning and end of the flow of activities performed in the system.

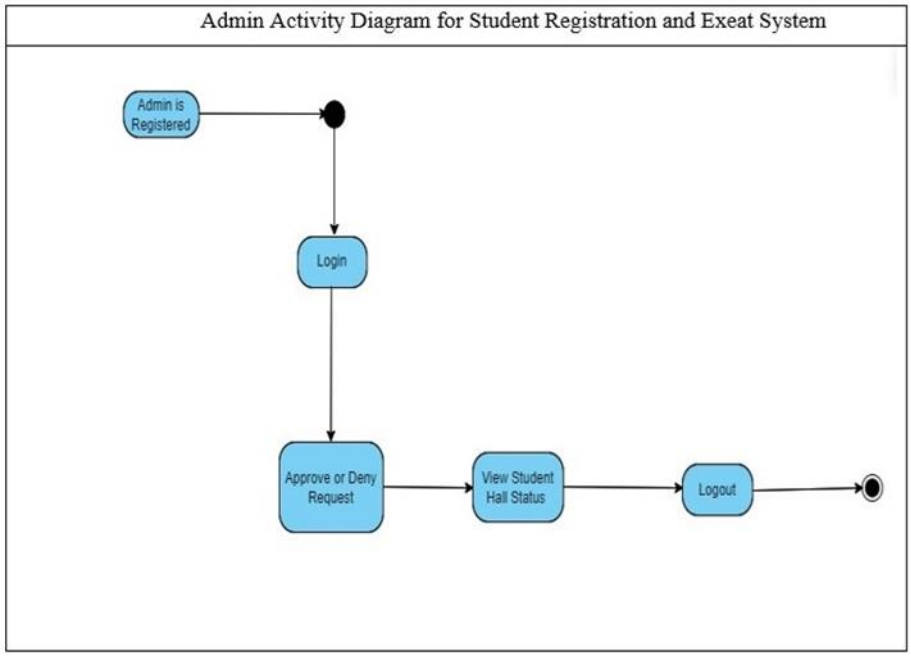

Figure V: Admin Activity Diagram

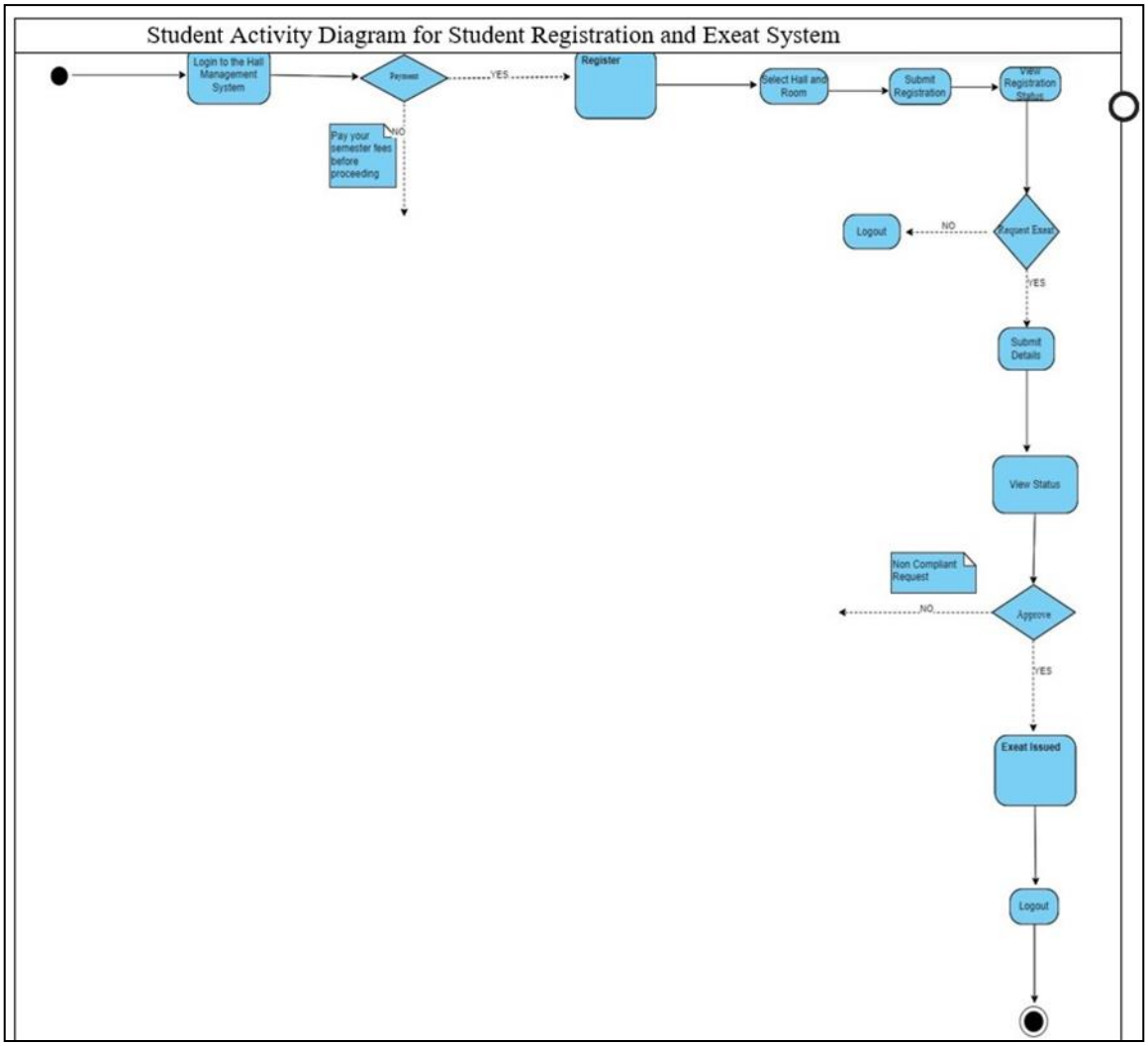

Figure VI: Student Activity Diagram

The activity diagram above describes the activities of the student which begins with the user logging in to the system. After login, the user proceeds to register by selecting a room and hall, followed by submitting the registration. Once registered, the user can view the registration status. At this point, the user can choose to log out; otherwise, they can

request an exeat. Upon exeat request, the user submits the necessary details and then views the status to determine if the request has been approved or marked as non-compliant. Figure V is the admin activity diagram and figure VI is the student activity diagram.

5) ENTITY RELATIONSHIP DIAGRAM

Entity-Relationship Diagrams (ERDs) serve as a graphical representation utilized for modeling a database's structure [10]. They visually showcase the entities present in a system or database, their attributes, and the connections between them. Entities represent real-world objects or concepts with associated data, depicted as rectangles in the ERD [11]. Attributes define specific properties of these entities and are encapsulated within the entity rectangles. Relationships are illustrated by lines connecting entities, illustrating how they are related or interconnected [12].

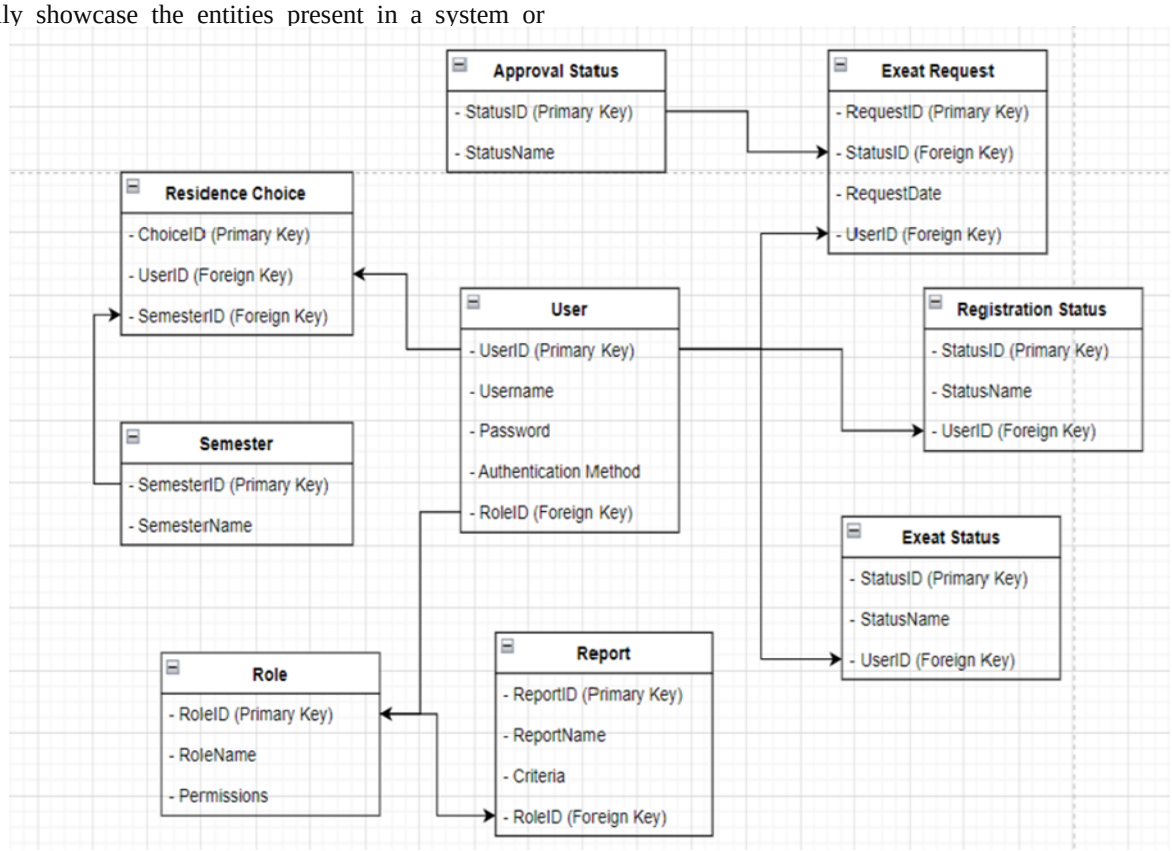

Figure VII: Entity Relationship Diagram for SRES

The ER diagram presents a system that includes entities like users, their roles, semesters, residence options, registration and approval statuses, requests, and reports. Each user has a unique identity and can have specific roles, reside in certain places, make requests, and create reports. Roles come with specific permissions or access levels. Semesters are linked to residence options, and registration statuses show the current state of users related to housing. Requests made by users regarding leaving campus (Exeat requests) are connected to approval statuses. The diagram shows how these entities and their features are linked, giving a detailed picture of how the system organizes and connects its data.

VI. DESIGN AND DEVELOPMENT TOOLS

The following tools were used for the development of the student registration and exeat system:

a) HTML: HTML, or Hypertext Markup Language, is a fundamental building block of web development. It helps define the layout and display of content on websites. As a fundamental part of web development, HTML was used to display the contents on the webpage such as registration forms, student information, tables, navigation links, and interactive elements such as buttons whilst complementing CSS and JavaScript to create visually appealing and functional online experiences.

b) CSS: In the student registration and exeat system, CSS (Cascading Style Sheets) plays a vital role in enhancing the visual presentation and layout of web pages by adding colors, fonts, layouts, and other appearance details for HTML elements. CSS helps make responsive and attractive designs that look good on different devices and screen sizes. It also keeps the content and design separate, making it easier to manage code and make changes to web development projects.

c) JAVASCRIPT: JavaScript is programming language that's often used to add interactivity and movement to websites. The SRES required JavaScript to manage the behavior and appearance of elements within the system's interface, and communicate with servers seamlessly, all without requiring page refreshes.

d) MONGODB: MongoDB is a widely utilized NoSQL database due to its adaptability, scalability, and user-friendliness. Using MongoDB, the system can effectively handle unstructured or semi-structured data, which is common in education systems where student information has varying formats and content. MongoDB stores data in JSON-like documents, allowing for flexible data models that support agile development and continuous improvements to the system. Additionally, it offers high availability, horizontal scalability through sharding for managing increased data

volumes, and advanced capabilities for complex queries, indexing, and aggregation pipelines.

e) NODE JS/EXPRESS: Node.js and Express.js together create a powerful framework for building the backend of web applications. Node.js allows the execution of JavaScript code on the server, making it a runtime environment for backend programming. Express.js, a lightweight Node.js web framework, makes it easy to handle HTTP requests and responses, route traffic, and use middleware. This combination proves beneficial for building and maintaining the backend infrastructure of systems like student registration and exeat management, ensuring streamlined operations and effective handling of web-based functionalities.

V. IMPLEMENTATION AND RESULTS

This part involves the conversion of the system specifications into a working web application, from the software design to development. The SRES consists of 3 parts, they are: the students view, the administrator views and the lecturer's view.

A. The student views:

This is the graphical user interface made for the students to interact with the system to achieve their expectations from the system. This part involves the main interaction done by the students and the graphical user interfaces GUI that help the student navigate through the system.

1) Student Login Page: This is where the student logs in to access the system with their unique details and secure password to access the student dashboard. Figure III is the student login page.

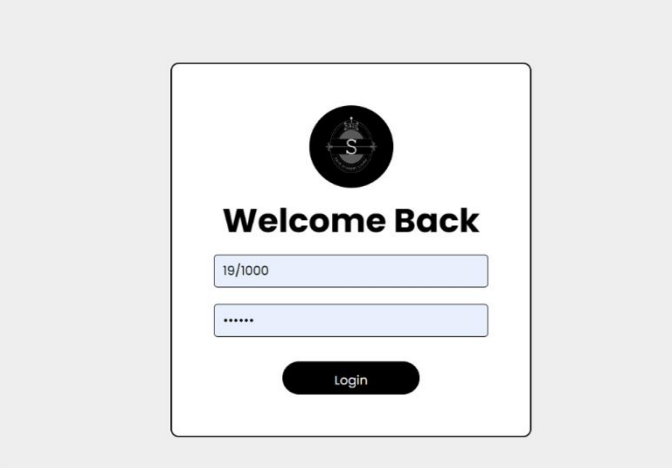

Figure VIII: Login page for Student Registration and Exeat System

2) Student Dashboard: This is the students first view when their login details meet the authentication requirements, it displays basic information and notifications the student is to know. Figure IV shows the student dashboard.

Figure IX: Student Dashboard for Student Registration and Exeat System

3) Student Registration page: This view shows the available halls based on the filtered type and it allows the user to insert the room and block they want to occupy. Figure V below shows the hall registration page. This is where student can choose the type of hall which can be classic, premium or regular depending on financial ability.

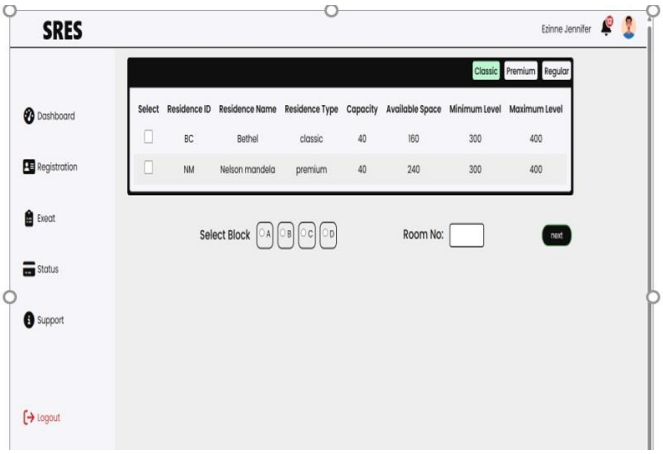

Figure X: Student Hall Registration Page

4) Student Exeat Page: This view allows the student to request for exeat between specific dates. The options given could be weekly or weekend exeat request and a description showing the rules for exeat request. Figure VI shows the exeat request page.

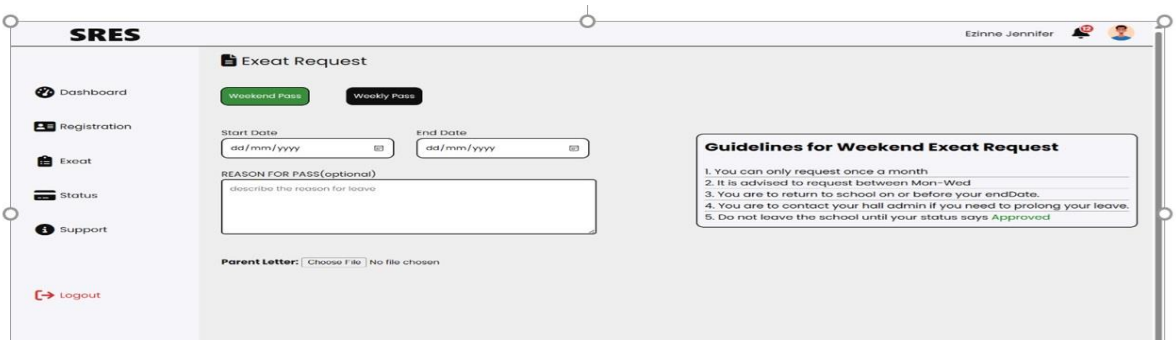

Figure XI: Weekday Exeat Request Page

5) Student Status Page: This view displays student registration and exeat request status and shows if the student is active or not in the hall.

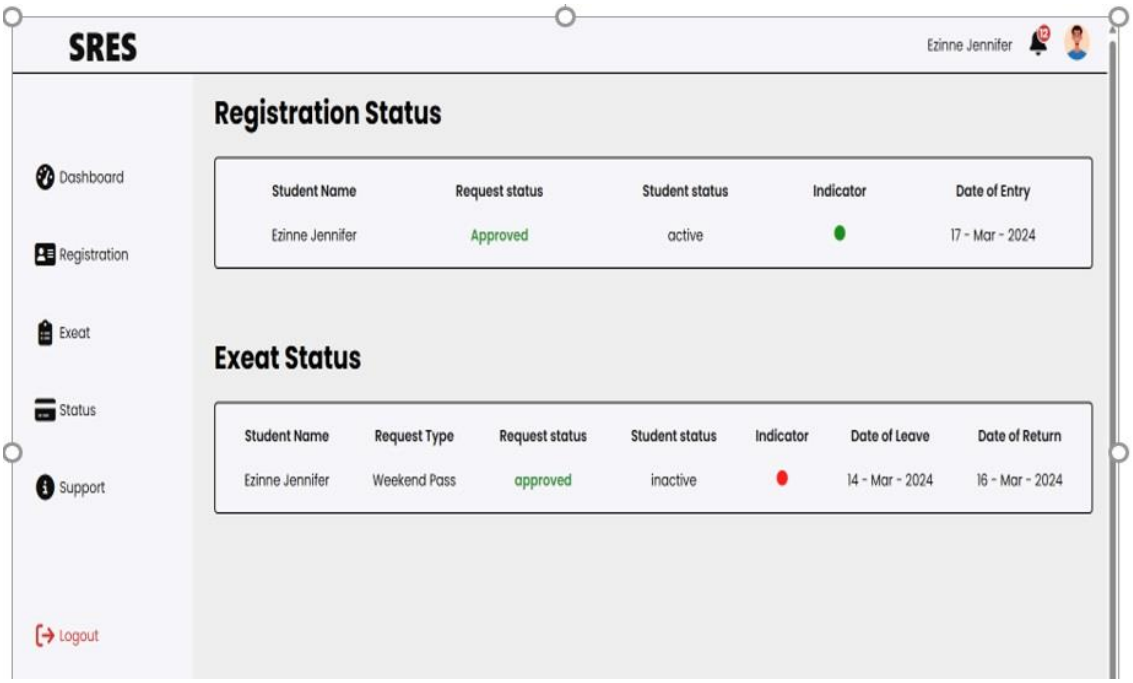

Figure XII: Student Status Page

B. The Admin views

This part involves the main interaction done by the students and the graphical user interfaces GUI that help the admin navigate through the system.

1) The Admin Student Registration page

This view shows the registered students awaiting to be signed in by the admin. It contains a search bar that allows easy retrieval of student information based on the student ID provided. Figure VIII is the admin student sign-in page figure IX is the admin search filter page.

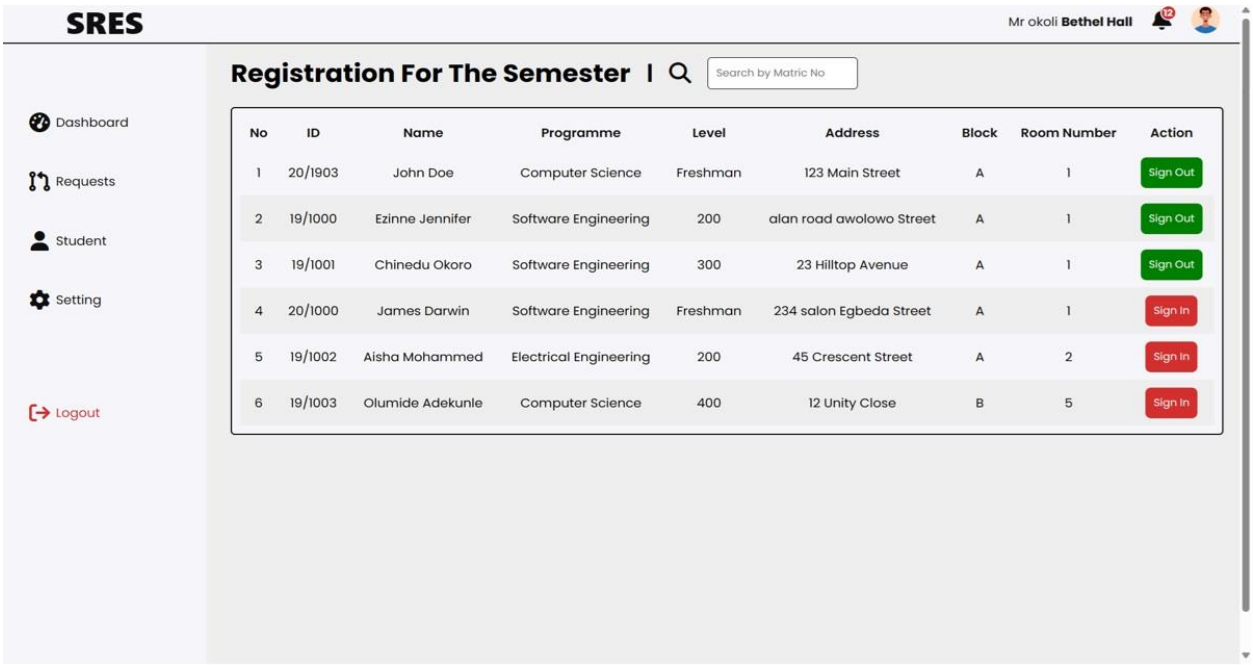

Figure XII: Admin Student Sign- in Page

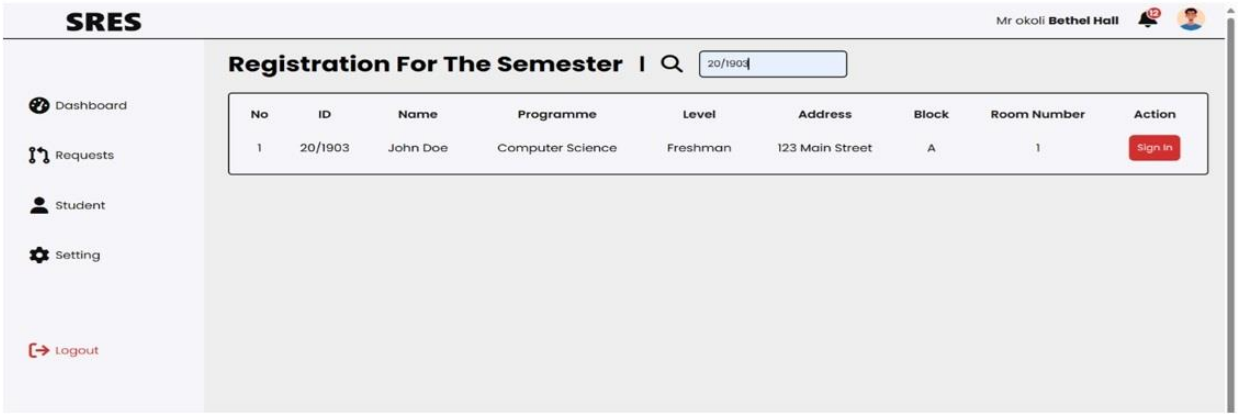

Figure XIII: Admin Search Filter Page

C. The Admin Exeat Request page

This view shows the students who have requested for exeat permission, the admin can view the student information and approve or reject the exeat request. Figure X is the admin exeat request view page.

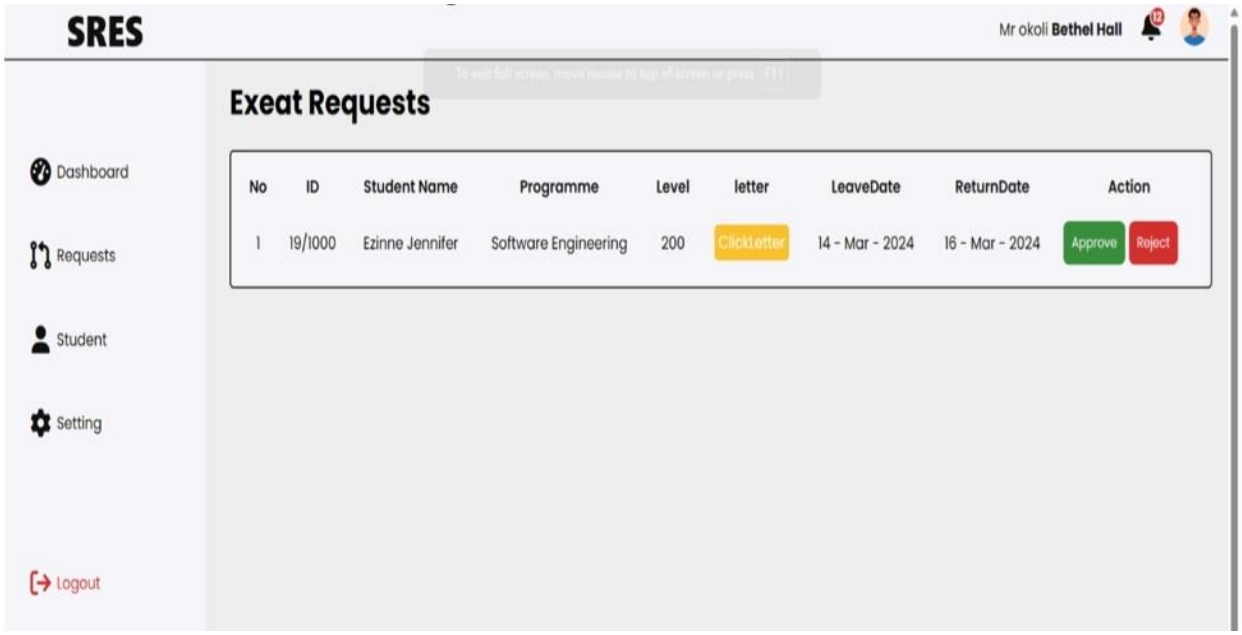

Figure XIV: Admin Exeat Request View Page

VI. SUMMARY AND CONCLUSION

Student registration and exeat processes has been made easier to run and manage thereby reducing administrative work with this project's goal. It aimed to make it easier and faster for students to register and to request leave. The most important parts of this system were put together allowing users to seamlessly navigate through the system and complete various registration and exeat operations.

This new web application is a major turn in managing student data for universities. It fixes the problems with old student registration and exeat request systems. It's a solution for the complicated process of getting permission for students to leave campus. Users will have no problems using this system, and it will get rid of all the annoying steps that usually come with student registration and exeat requests.

VII. RECOMMENDATIONS

A QR Code or Near Field Technology (NFC) can be implemented for easy check-in and access control. Feedback and support channels can also be implemented for users to submit feedback, suggestions, or support requests. Actively solicit user input to identify areas for improvement and address any issues or concerns promptly.

REFERENCES

[1] Bennett, S., McRobb, P., & Farmer, R. (2000). Object-Oriented Systems Analysis and Design Using UML (2nd ed.). Addison-Wesley Longman, Inc.
[2] Codd, E. F. (1970). A relational model of data for large shared banks of data. Communications of the ACM, 13(6), 377-387.
[3] Date, C. J. (2000). An introduction to database systems (Vol. 8). Addison-Wesley Longman, Inc.
[4] Elmasri, R., & Navathe, S. B. (2000). Fundamentals of database systems (5th ed.). Addison-Wesley Longman, Inc.

[5] Johnstone, D. B. (1999). *Student housing: Policy and practice* (Vol. 14). Taylor & Francis.

[6] Kendall, K. E., & Kendall, J. E. (2019). Systems Analysis and Design (10th ed.). Pearson Education Limited.

[7] Larman, C. (2004). Applying UML and patterns: An introduction to object-oriented analysis and design (3rd ed.). Addison-Wesley Professional. Chapter 8.

[8] Noraziah, A. H., Azila, H., & Adnan, S. (2021). Hostel management system: A review. *International Journal of Advanced Science and Technology*, 149(10), 118-127.

[9] Onuiri, E., Odukoya, A., Yadeka, C., & Nzei, M. (2014). Design and implementation of an electronic exeat system. International Journal of Scientific and Engineering Research, 5(1). Retrieve from: https://www.researchgate.net/publication/305426003_Design_and_Implementation_of_an_Electronic_Exeat_System

[10] Schwaber, K., & Sutherland, J. (2017). The Scrum guide: The rules of the game. Scrum Guides.

[11] Suriya, S., Sundaram, G. M., Abhishek, R., & Vignesh, A. A. (2021). Online hostel management system using hybridized techniques of random forest algorithm and long short-term memory. *Advances in Machine Learning and Computational Intelligence* (pp. 207-218). Springer

[12] Wirfs-Brock, R., Wilkerson, B., & Wiener, L. (2000). Designing Object-Oriented Software: Applying UML and Patterns (2nd ed.). Prentice Hall PTR.