Abstract

This is the design document for the software that the team is designing. It gives a brief summary on how the whole software works.

SOFTWARE DESIGN DOCUMENT

# Table of Contents

[Table of Contents 1](#_Toc120020131)

Group members' Information ……………………………………………………………………………………………………………2

Introduction ………………………………………………………………………………………………………………………………………3

Design Consideration ………………………………………………………………………………………………………………………..4

Architecture ………………………………………………………………………………………………………………………………………5

Use Case ……………………………………………………………………………………………………………………………………………6

Component Design ……………………………………………………………………………………………………………………………8

Software Interface Design ……………………………………………………………………………………………………………….11

Basic Folders for Class Diagrams ……………………………………………………………………………………………………..15

Group members:

Mpumelelo Dlamini - 202002852

Sakhile Dlamini - 202003084

Bathandwa Mavuso - 202003846

Mandisa Nkumane - 202004102

Luyanda Zwane - 202002531

1. **Introduction**

The project is to create a residential billing system for all in campus students of the University of Eswatini. The system should be able to make campus application for residence available for every student, provide accurate financial record pertaining students bills, safely store data for future access and be accessed by authorized personal only.

This design document presents the designs used or intended to be used in implementing the project.

* 1. **Purpose**

The purpose of this document is to present a detailed description of the designs of the Residential Billing System, created for the University of Eswatini. Firstly, this document is intended for the programmer of the group, to use the designs as guideline to implement the project. Equally, this document is also for the group instructor, Mr E.Dube, as it fulfils one of the requirements of the project. Lastly, this document could be of use to the group when they want to upgrade or modify the present design of the billing system.

* 1. **Scope**

This document gives a detailed description of the software architecture of the billing system. It also displays some of the use cases that had been transformed into sequential and activity diagrams. The class diagrams show how the programmer would implement the specific models.

* 1. **References**

The user of this Software Design Documentation may need the following documents for references:

IEEE Standard 1016-1998, IEEE Recommended Practices for Software Requirements Specification, IEEE Computer Society, 1998.

Team 1, 2010, Software Design Documentation, UUIS, Last Modified: Apr. 29, 2010.

* 1. **Overview**

This document is designed according to the standards for Software Design Documentation explained in CSC392: Practices in Software Engineering 1, Lecture 8: Object-Oriented Design and IEEE Practice for Software Design Documentation. Section 3-5, contain discussions of the design for the projects with diagrams.

1. **Design Considerations** 
   1. **Assumptions**

The user of the billing system is aware of the basic operations of a computer and web pages.

The user also understands the standard terms used for the operation.

* 1. **Constraints**

The system is built to be accessible only through the university’s website. The system is implemented using Java and JSP technologies.

* 1. **System Environment**

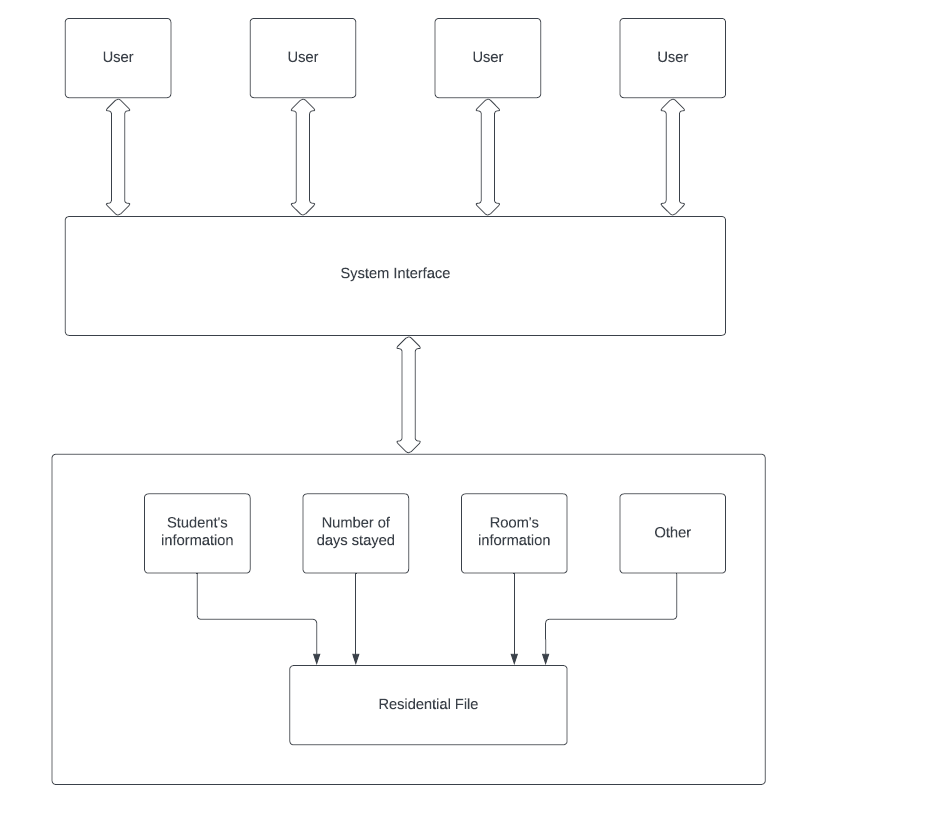
The web based billing system is designed to work on all operating systems. The system is accessible through any desktop, laptop and cellphone connected to the internet. It is accessible at all times.

* 1. **Design Methodology**

The system is designed with flexibility for further development and/or modification. The system is divided into manageable processes that are grouped to sub-modules and modules that are built with abstraction.

1. **ARCHITECTURE** 
   1. System Design

Below is a block diagram that shows the principal parts of the system and their interactions.



**4. Use Cases**

4.1. Actors

* + 1. Public Users: these are the Students that will often use this system for different actions. These users are essential for the system since the system will be mostly utilized by them.
    2. System Under Design: this is the Residential Billing System being created. This actor represents the system and the actions that it takes.
    3. **Administrative User**: The Administrative User is a user who administers the system byoverseeing accounts creation and administration.

4.2. List of Use Cases

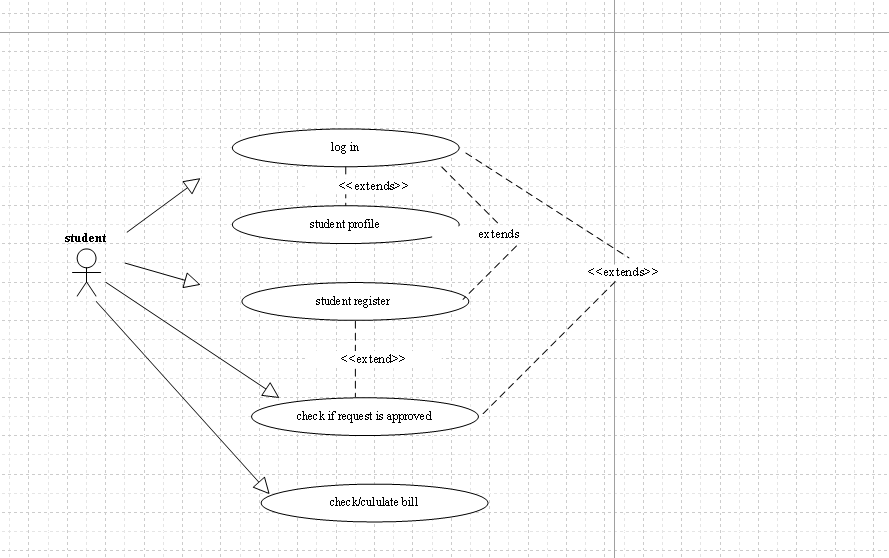
4.2.1 Student Use Case

* Log In
* Student Profile
* Student Register
* Check if register is approved
* Check/calculate bill

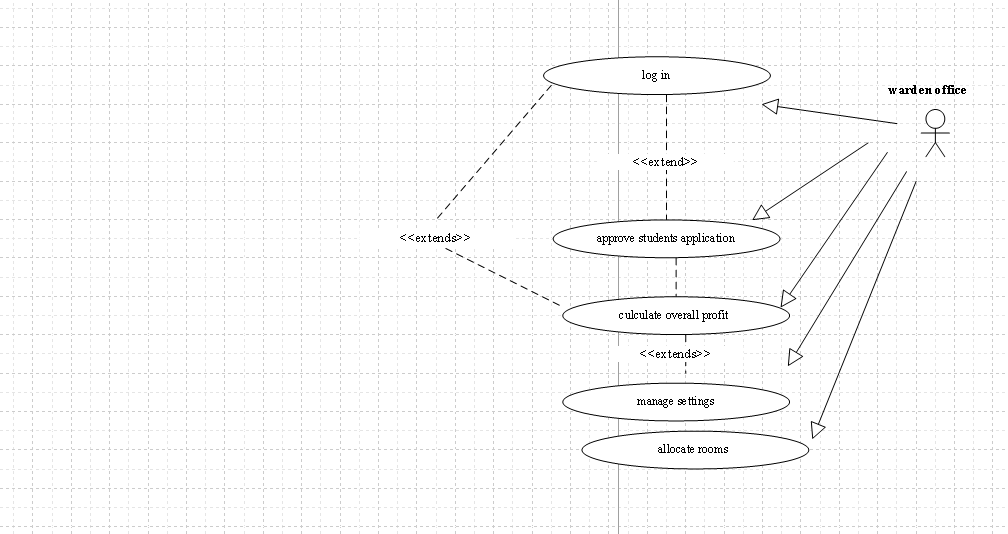
4.2.2 Warden Use Case

* Log In
* Approve students’ application
* Calculate overall profit
* Allocate rooms
* Manage settings
  1. Use Case Diagrams

* + 1. Student Use Case Diagram



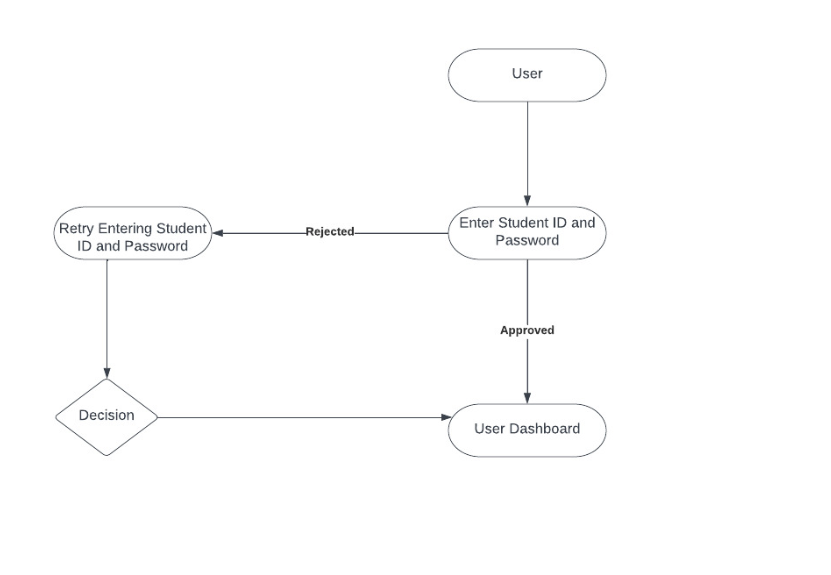
* + 1. Administrative Use Case Diagram



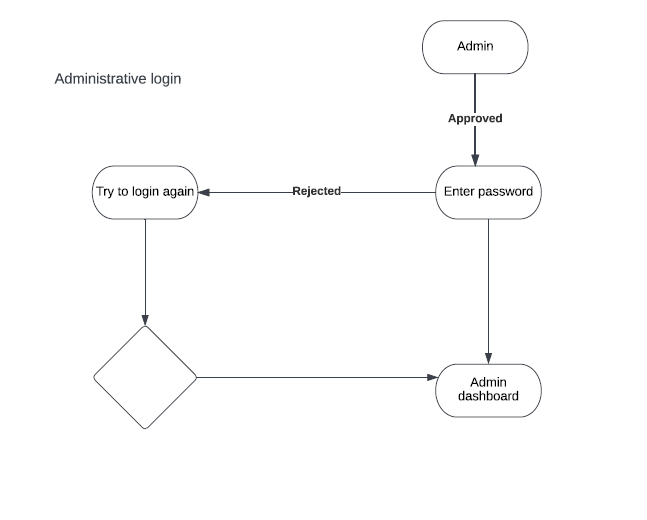
**5. Component design**

**5.1 Login**

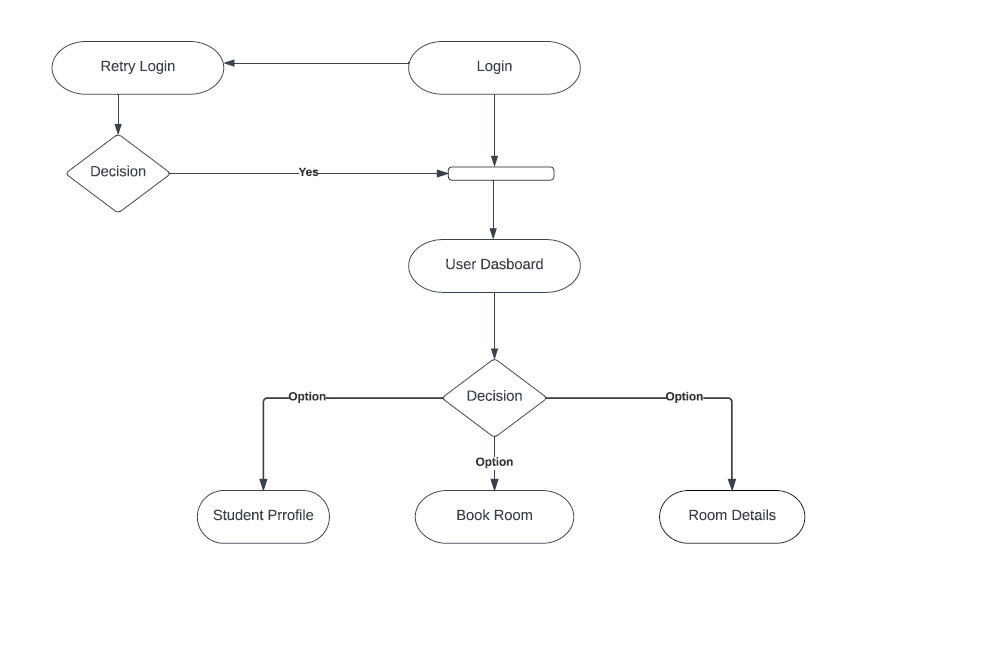
The diagram below shows the flow chart for logging in.



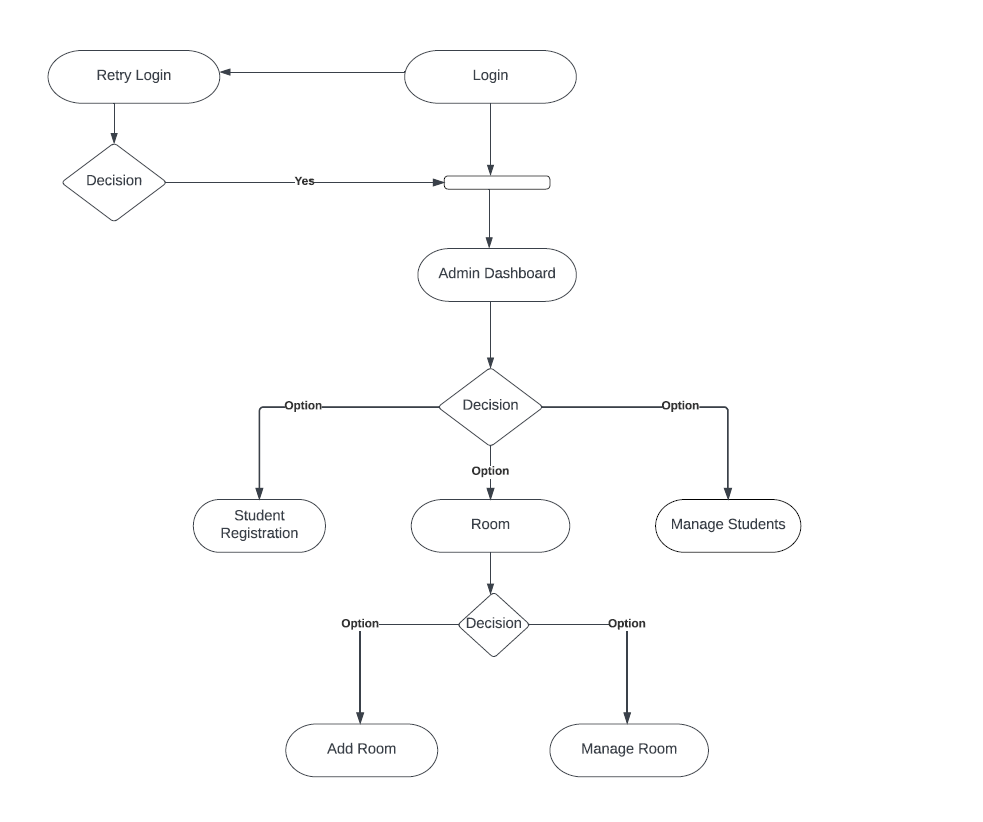
Administrative Login



User's Dashboard page



Administrator dashboard



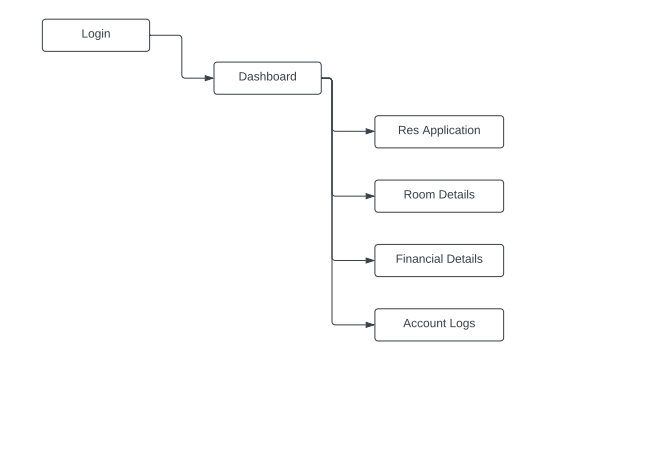
1. **Software Interface Design**

**6.1 User interface design**

UI is designed according to the UI design principles.

* + - The structure principle: UI is organized in such a way that related things are combined together and unrelated things are separated.
    - The simplicity principle: It is easy to follow the provided interface. In the case of a mistake, system will display an error massage.
    - The visibility principle: All system’s functions are available through the UI. It does not overwhelm the users with too many alternatives.
    - The feedback principle: Through the system of messages, the design keeps users informed of actions, errors, or exceptions.
    - The reuse principle: In design, same names were used to perform the same operations with different objects in order to reduce ambiguity.
  1. **System pages in a tree**

The system’s web pages are presented in a tree form. From logging in, the user can reach the main page. From the Dashboard page, the user can access the following pages: “Res application”, “Room details”, “Financial details” and “Account logs”. All these pages cover necessary functionality of the system. It easy to navigate between these pages.



The system web pages for the Administrator in a tree form. From logging in the administrator can reach the main page. From the Dashboard page, administrator can access the following pages: “Recent Applications”, “Manage Students”, “Manage Rooms”, “Settings” and “User Access Logs”. All of these pages cover necessary functionality of the system.

Diagram

Description automatically generated

* 1. **Description**

“**Dashboard**” contains a list of the main system functionality. After the “dashboard” page changes and guide the user on how to work with system.

“**Res Application**” displays a form that students need to fill in order to obtain a room. The form is easy to understand and fill.

“**Room Details**” displays the room details such as the block the room is located in and what type is the room, such as is it a single or double.

“**Financial Details**” displays all the financial records corresponding with each student account. Shows even the dates of when these financial transactions.

“**Account Logs**” displays the times at which a student log in and how many times the student logged in.

“**Manage Students”** displays all the students that are registered for rooms in the school. Also allows the Administrator to make some edits.

“**Manage Rooms**” displays all the rooms that are found in the school. Also, Administrator can tell which room is free or not.

“**Settings**” system function that allows the Administrator to be able to manipulate the billing function, like when should the system start to bill students and when to stop.

“**User Access Log**” system function that display all devices that were used to access a certain account.

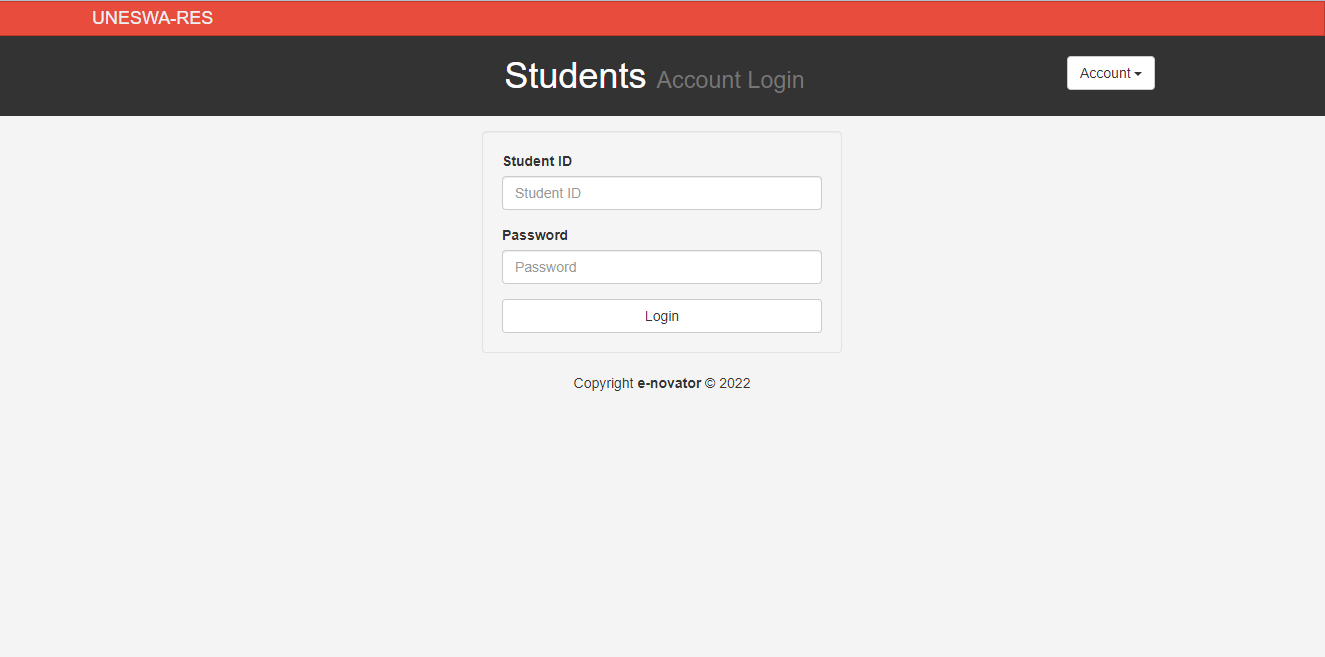
* 1. **User Interface**

Used GUI components are menus, submenus, buttons, text boxes, check boxes, down drop, links, lists and tables. The only means of accessing the entire database, by all users, is through this UI.

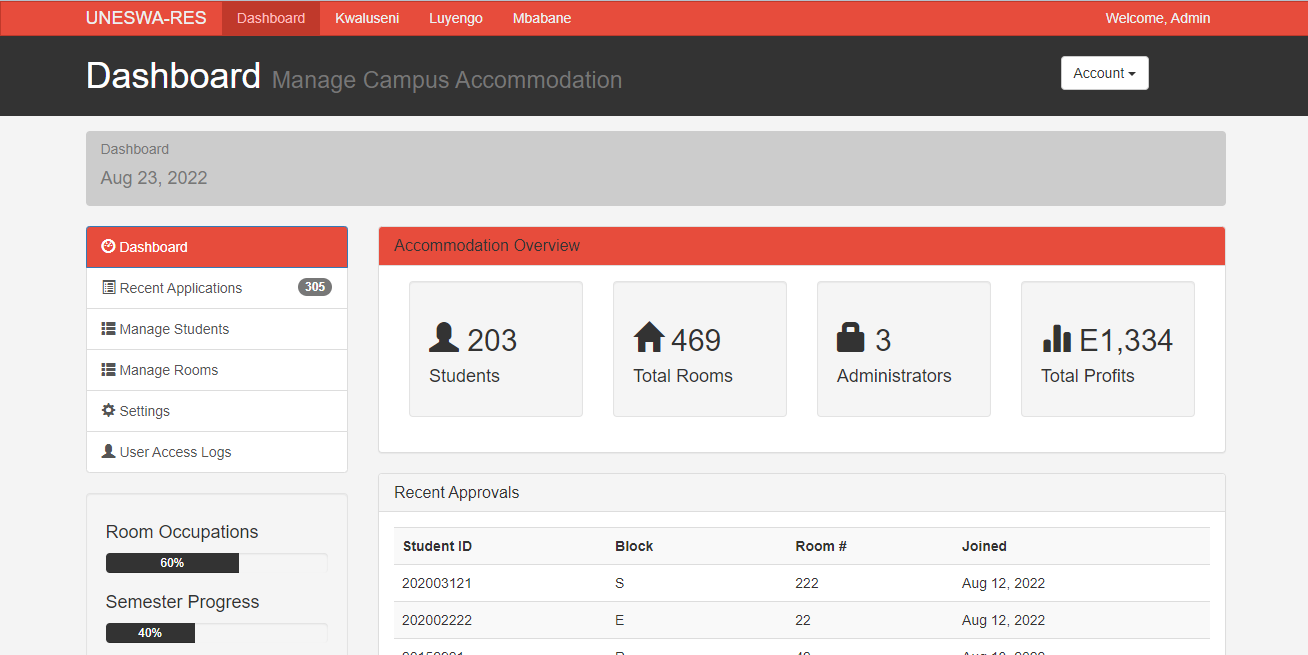
* + 1. **Screen Images**

Some examples of UI are presented below:

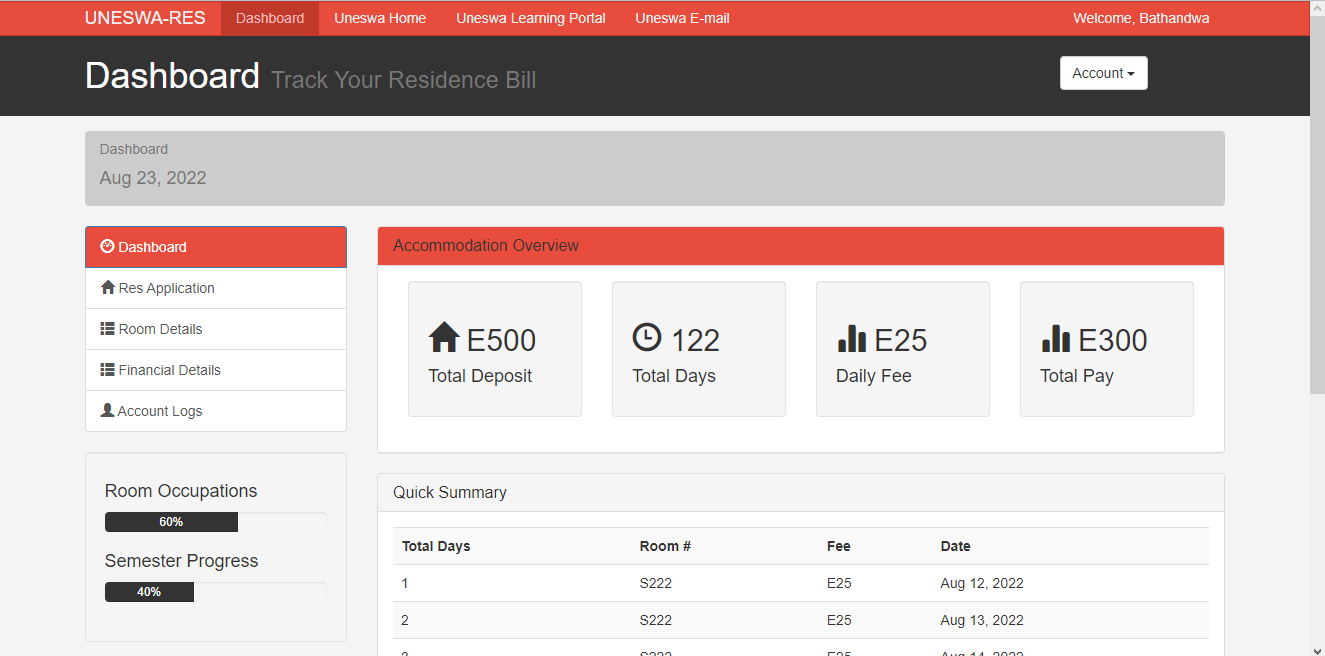
Log in page for students



Administrator’s Dashboard page



Students’ dashboard



* 1. **Module Interface Design**

Module design maintains MVC (model-view-controller) architecture. View is a UI. Through UI user input data which goes to Controller. Controller transfers data into Model. If data is incorrect Model shows error Message. Otherwise, it processes the request, prepares the result and sends it to the controller. Finally, controller transfer generated code into View. The user views the results.

1. **Basic folders for class diagram**

