# 



**-PROJECT ITT-**

**Documentation**

|  |  |
| --- | --- |
| **Authors** | **C2108G2\_Team 05** |
| **1. Thai Huu Tuan(L)**  **2. Ha Dinh Lam**  **3. Nguyen Van Lap**  **4. Nguyen Hung Man** |
| **Date** | **20/10/2022** |
| **Instructor** | **Teacher To Hoang Anh** |

Hanoi, 20/10/2022

**Index**

[**1.**](#_30j0zll) **Problem Definition 3**

[1.1.](#_1fob9te) Problem Abstraction 3

[1.2.](#_3znysh7) The Current System 3

[1.3.](#_2et92p0) The Proposed System 3

[1.3.1.](#_tyjcwt) Boundaries of the System 3

[1.3.2.](#_3dy6vkm) Hardware and Software Requirements 3

[**2.**](#_1t3h5sf) **Customer Requirements Specification 3**

[2.1.](#_4d34og8) Users of the System 3

[2.2.](#_2s8eyo1) System functions 3

[**3.**](#_17dp8vu) **System Designs 3**

[3.1.](#_3rdcrjn) Entity Relationship Diagram 3

[3.2.](#_26in1rg) Database Design 3

[3.3.](#_35nkun2) System fucntions design 4

[**4.**](#_1ksv4uv) **Task sheet 4**

[**5.**](#_44sinio) **Validation Checklists 4**

# Problem Definition

## Problem Abstraction

The application of information technology to management is being well received and widespread, along with that, the dormitories also need a new management application to help simplify the work of replacing the current system, is now outdated .

## The Current System

Dorm Manager

The management of students in the dormitory of the current system has many shortcomings, for example, it is difficult to statistic the current status of the dormitory, it is difficult to divide the management for many people because of difficulties. in information synchronization, etc.

## The Proposed System

App will be created based on the following requirements.

1. Give general information of the current dormitory system.
2. Buttons can go to another page.
3. Display the list of students, rooms, bills, employees clearly.
4. Details will be given when the user selects the item in the list.

5. The admin account has the right to manage subordinate staff accounts

1. Provides the ability to recover the password when forgot pasword
2. Function to search for students by name, filter bills by month
3. Prevent deleting rooms when there are students.
4. Automatically alert to student when have bill to pay or bill is paid
5. Allow student to send feedback to manager
6. Allow manager to send announcement to room

## Boundaries of the System

Student, user scope

Students living in faraway dormitories have a high demand for service

The audience can be every household, company, hospital, student, student

## Hardware and Software Requirements

**Hardware**

* A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better
* 64 Megabytes of RAM or better

**Software**

* Code editor
* Internet connection

# Customer Requirements Specification

1. A computer with internet connection
2. A mysql database sever

## Users of the System

There is a special application and system user list created when provisioning the system. Special system users created for support and integration scenarios. Application users are created during system provisioning to manage setup and configuration. Application users can also be used to perform ancillary services, and their data access is managed by a specially designated security role. These security roles are managed by the system and cannot be modified. View other predefined security roles in the system.

Most of these users are hidden from user view, but they can be found using the Advanced Find feature on the Users panel. Do not delete or modify these users, including changing or reassigning security roles.

## System functions

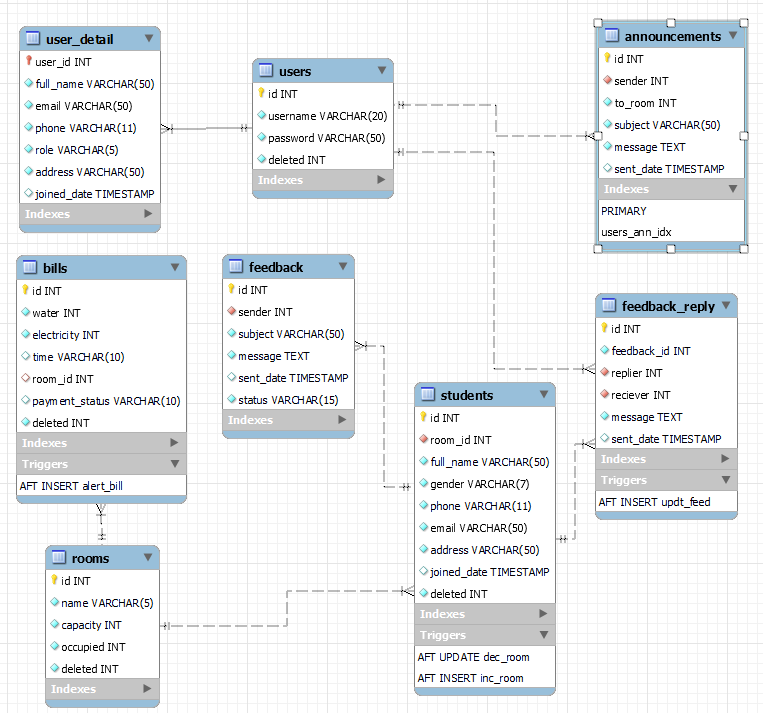
1. Login, logout, reset password, change password, add account
2. Add room, student, bill
3. Search student
4. Filter bill by month
5. Edit room, bill, account and student information

# System Designs

## Entity Relationship Diagram

Entity-relationship diagram, which represents the relationships of a set of entities stored in a database. Diagrams assist in explaining the logical structure of the database. includes many specialized symbols, and its meaning makes this pattern unique.

## Database Design



* Table detail

# Users

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT |  |
| username | VARCHAR (20) | Username used to login |
| password | VARCHAR (20) | password |
| deleted | INT | (1 or 0) deletion status |

# User\_detail

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| user\_id | INT | Id of user |
| full\_name | VARCHAR | User’s full name |
| email | VARCHAR | User email |
| phone | VARCHAR | Phone number |
| role | VARCHAR | Account type (Admin/ Staff) |
| address | VARCHAR | User’s address |
| joined\_date | TIMESTAMP | Register date |

# Rooms

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Room id |
| name | VARCHAR | Room label |
| capacity | INT | Room capacity |
| occupied | INT | Occupied slot |
| deleted | INT | (1 or 0) deletion status |

# Bills

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Bill id |
| water | INT | Water will |
| electricity | INT | Electricity bill |
| time | VARCHAR | Time of bill affects |
| room\_id | INT | Room id |
| payment\_status | VARCHAR | Paid or Not paid |
| deleted | INT | (1 or 0) deletion status |

# Students

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Student id |
| room\_id | INT | Room id that student live in |
| full\_name | VARCHAR | Student’s full name |
| gender | VARCHAR | Gender |
| phone | VARCHAR | Phone number |
| email | VARCHAR | Email |
| address | VARCHAR | Address |
| joined\_date | TIMESTAMP | Time join |
| deleted | INT | (1 or 0) deletion status |

# Feedback

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Feedback id |
| sender\_id | INT | Student’s id the sent the feedback |
| subject | VARCHAR | Feedback subject |
| message | VARCHAR | Feedback content |
| sent\_date | TIME\_STAMP | Time sent feedback |
| status | VARCHAR | Status (replies/not replied) |

# Feedback\_reply

|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Reply feedback id |
| feedback\_id | INT | Feedback id is being replied |
| replier | INT | Manager’s id the reply the feedback |
| reciever | INT | Feedback sender |
| message | VARCHAR | Content |
| sent\_date | TIME\_STAMP | Replied time |

# announcements

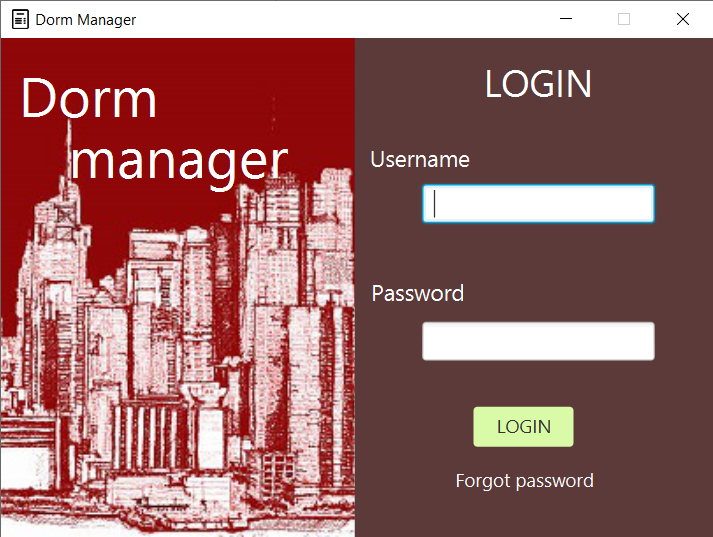
|  |  |  |
| --- | --- | --- |
| Key | Data Type | Descripion |
| id | INT | Announcement id |
| sender | INT | Manager’s id posted |
| to\_room | INT | Room id to recieved |
| subject | VARCHAR | Subject |
| message | VARCHAR | Content |
| sent\_date | TIME\_STAMP | Send time |

## Sitemap

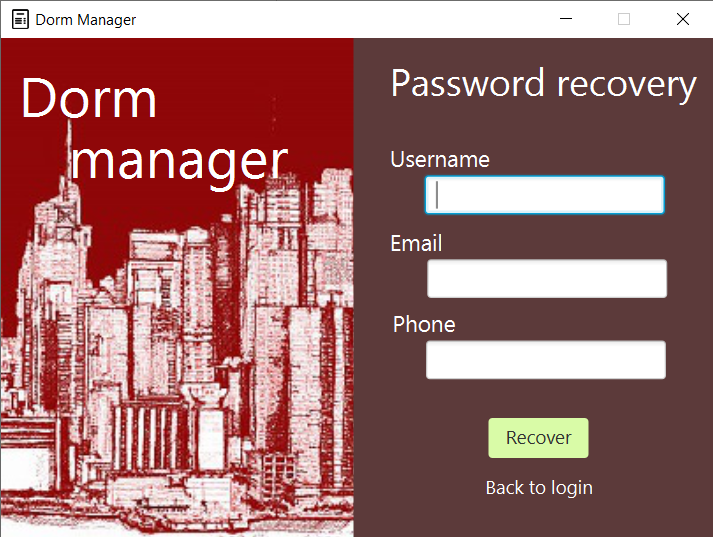
usually a hierarchical list of pages, often organized by topic or by the structure of the organization to which it belongs - as is the case in many websites designed solely to provide information about the organization. function instead of transactional business.

## System functions

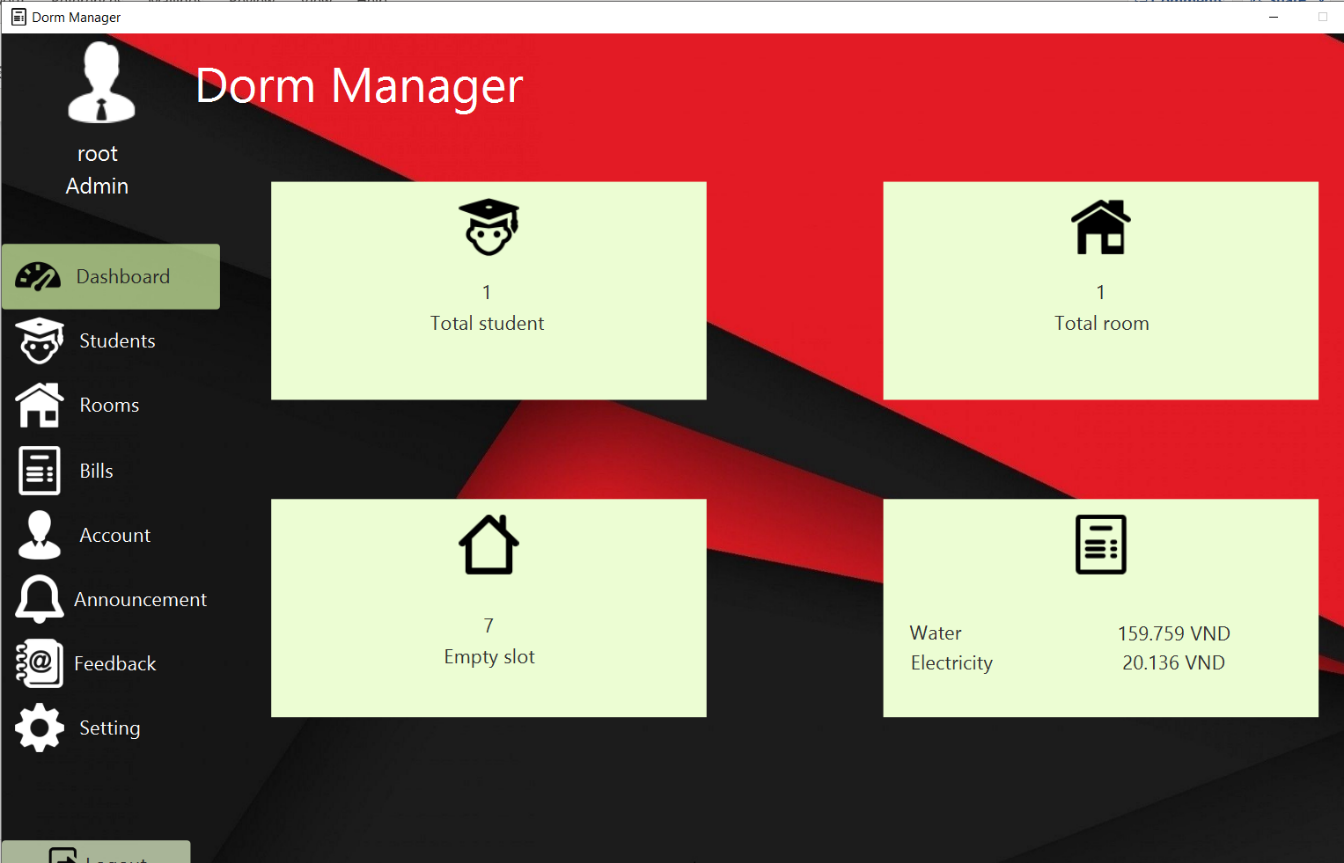
1. Login



1. Recover password



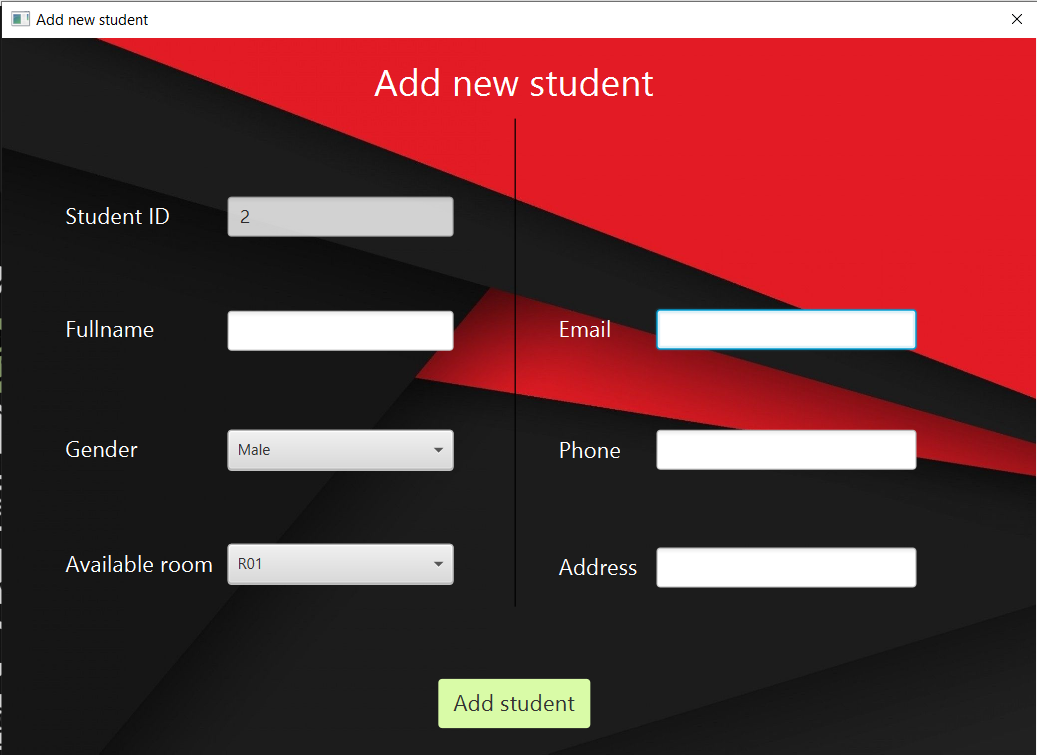
1. Dashboard



1. Manage student

# 

1. Add student



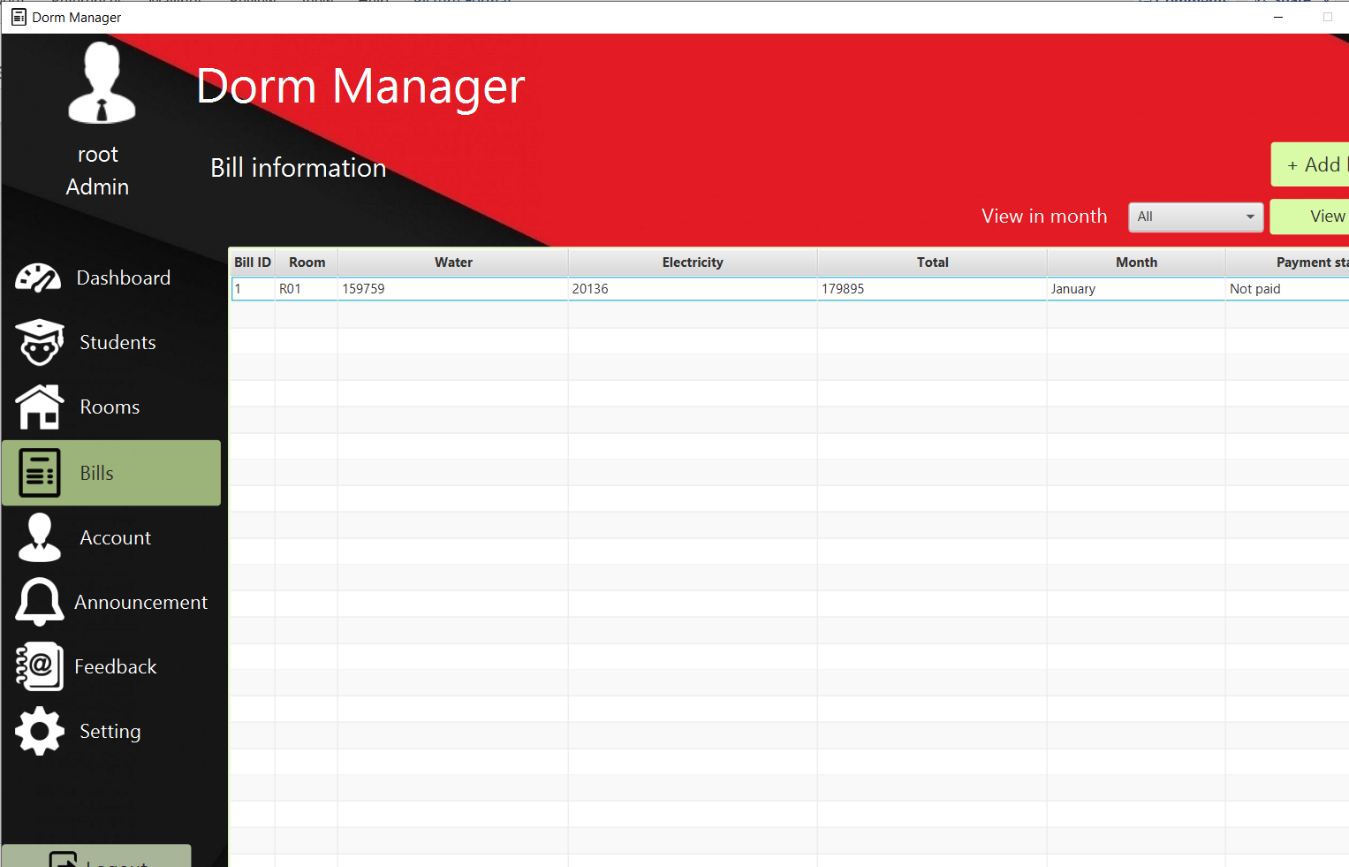
1. Manage room

# 

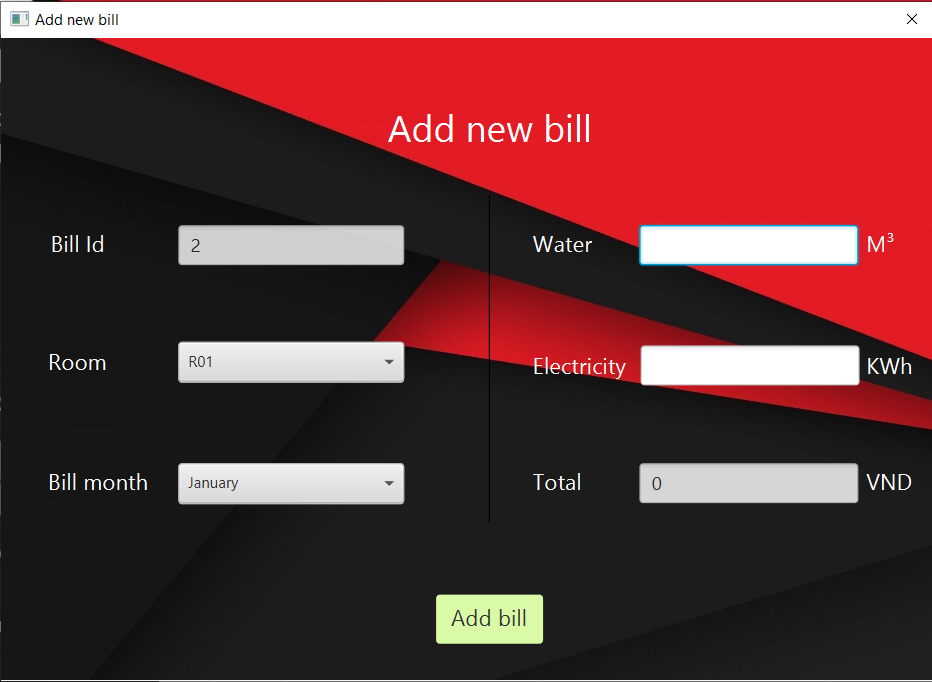
1. Add room

# 

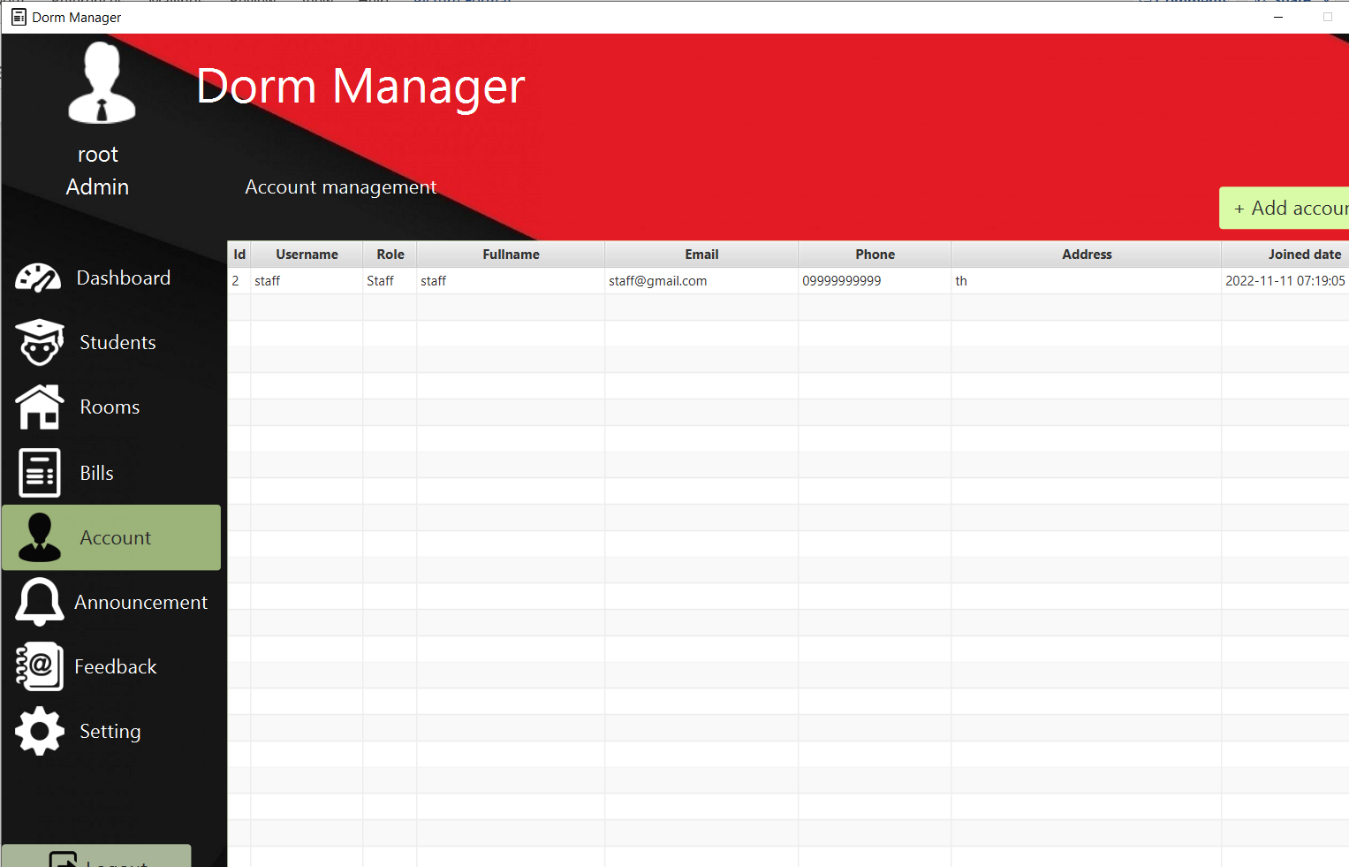
1. Manage bill



1. Add bill



1. Manage account



1. Add new manager account

# 

1. Manage announcement

Graphical user interface, application, table, Excel

Description automatically generated

1. Add announcement manually

Graphical user interface, website

Description automatically generated

1. Manage feedback

Graphical user interface

Description automatically generated

1. View and answer feedback

Graphical user interface

Description automatically generated

1. Setting (update account information)

# 

## Algorithms

1. Login, logout, change password and recover password

Diagram

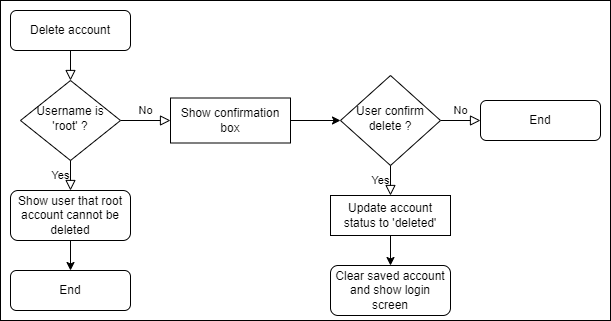
Description automatically generated

1. Add student, room, bill, account

Diagram

Description automatically generated

1. Delete account



1. Update account information

Diagram

Description automatically generated

# Task sheet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | Work | Start date | End date | Completion percentage |
| Thái Hữu Tuấn | Code, database design, fix bug | 6/10/2022 | 8/11/2022 | 95% |
| Hà Đình Lâm | UI design, document | 13/10/2022 | 20/10/2022 | 80% |
| Nguyễn Văn Lập | UI design, document | 13/10/2022 | 20/10/2022 | 75% |
| Nguyễn Hùng Mạnh | UI design, PowerPoint, test | 13/10/2022 | 20/10/2022 | 80% |

# Validation Checklists

|  |  |
| --- | --- |
| Test | Worked or resolved? |
| Login |  |
| Reset password |  |
| Change password |  |
| Dashboard |  |
| Students list |  |
| Add, search, view, modify, delete student |  |
| Room list |  |
| Add, view, modify, delete room |  |
| Bill list |  |
| Add, search, view, modify, delete bill |  |
| Account list |  |
| Add, view, modify, delete account |  |
| Setting (edit personal information) |  |
| Delete personal account |  |