***Uwe software management system hall Report***

***By***

***Sandra Sahnoune 21039366***

***Ala eddine Ballalou 21003026***

***Hamza Addad***

**Table of contents:**

* **Abstract**
* **Section 1: Specifications**

1. **Objective**
2. **Use case**
3. **Scenario**

* **Section 2: Plan of tests**

1. **Application of Agile practices**

* **Section 3: Conception**

1. **Software architecture**

* **Section 4: Final Tests**
* **Section 5: Documentation**

1. **Guide of Gui uses**

* **Abstract:**

This paper reports on the implementation of the UWE hall management system software developed by our team in order to design advanced software using good practices of project management methodology and concepts of object-oriented programming.

The software consists of creating a comfortable environment for the UWE staff in order to facilitate the managing of the university accommodation. Through a graphical user interface (Gui) using Tkinter as python framework and sqlite3 as database the users manage the supervision of the halls and keep track of all the new changes it undergoes.

* **Section 1: Specifications**

1. **Objective:**

The purpose of this software implementation is to solve the problem of managing a large amount of data the fastest, efficient and easiest way possible, eventually using graphical user interfaces features and agile concepts in order to build and develop an advanced software efficiently and consistent for future aims.

1. **Use Case:**

The software is designed for 3 users: admin, manager and warden, each of them having different roles, they can view and make changes of all the data related and limited to their responsibilities.

They interact with the Gui interface directly and all the changes made will be saved in the database.

The manager and the admin can access, delete and update all the data while the warden is only responsible for the cleaning.

1. **Scenario:**

The users are going to interact with the window interface, they need to login choosing the appropriate role and entering their authentication details, following this, another widget will appear displaying all the data saved in database. Depending on the user’s role the updating features will differ from them.

* **Section 2: Plan of tests**

|  |  |
| --- | --- |
| **Tests** | **Techniques** |
| Login window contains 3 authenticators: role, username, password | RBAC identifies in the easiest way who logged in the system and protect data more efficiently. |
| Set the window widget at a specific format | It saves the size and the way the data are displayed which looks better. |
| Display all the data in one widget | It gives a global view for the users to check all the data in one place and easy use of the software |
| Set only 2 command buttons: update and delete | It makes the software easy to use and clear for the users |
| Display the same data view for all the users | Because that all the users have the same access to view all the data the only exceptions are related to the updates. |

1. **Application of Agile practices:**

* **Strategy planning:** As a team of 3 developers, the main goal is to build strong software in an efficient and easy way during a short period of time. The software is designed to overcome the divert issues that face the manager and the warden to keep records of the general UWE accommodation system. The interface should be easy and clear to read in order to facilitate the interaction between the users and the GUI.
* **Continuous team iterations:** the software is developed using a good technique and efficient algorithms in order to be flexible for the different environment impacts and valid for all times and places and even future purposes.
* **Team communication:** communication and good understanding of each other is the key to a successful teamwork. The work has been divided into 3 main parts: the interface design using tkinter , the backend using python programming for the dynamic aspect of object-oriented and the database connection using the CRUD operations for system storage.

As everyone knows his job the communication was applied using web tool repository: Github, thus we can exchange our codes and ideas and see the changes made by others, all dealing with the same python file, this reviewing and editing technique make the project be processed in a faster way and improve ideas in order to solve problems efficiently.

* **Simplicity:** “Simple things are the best”, this quote has been one of the most coming to our minds, the software has been designed to users who may or may not have any previous knowledge or use of any IT tool, the design is soft and nice, the display of the data is wide clear and easy to read, and the updates are basic to enter and save.

Creativity does not mean complicated but innovative, and good software is usually designed in a straightforward method in order to be clear and understandable by other IT maintainers.

* **Section 3: Conception**

1. **Software Architecture:**

In order to be able to easily add new resolution methods and possibly new interfaces, the program was designed in such a way as to be modular.

There are 2 classes User() and Hall(), and some other structures such as function, in this case the delete Lease, button, update data are function, all of them connected to the database.

All the widgets including login window are setting up and implemented using Tkinter and other tools for GUI design and getting data as well as all the button command , update and delete.

Select function is implemented to get a Treeview graphical control of element while updating the system in a hierarchical view of data.

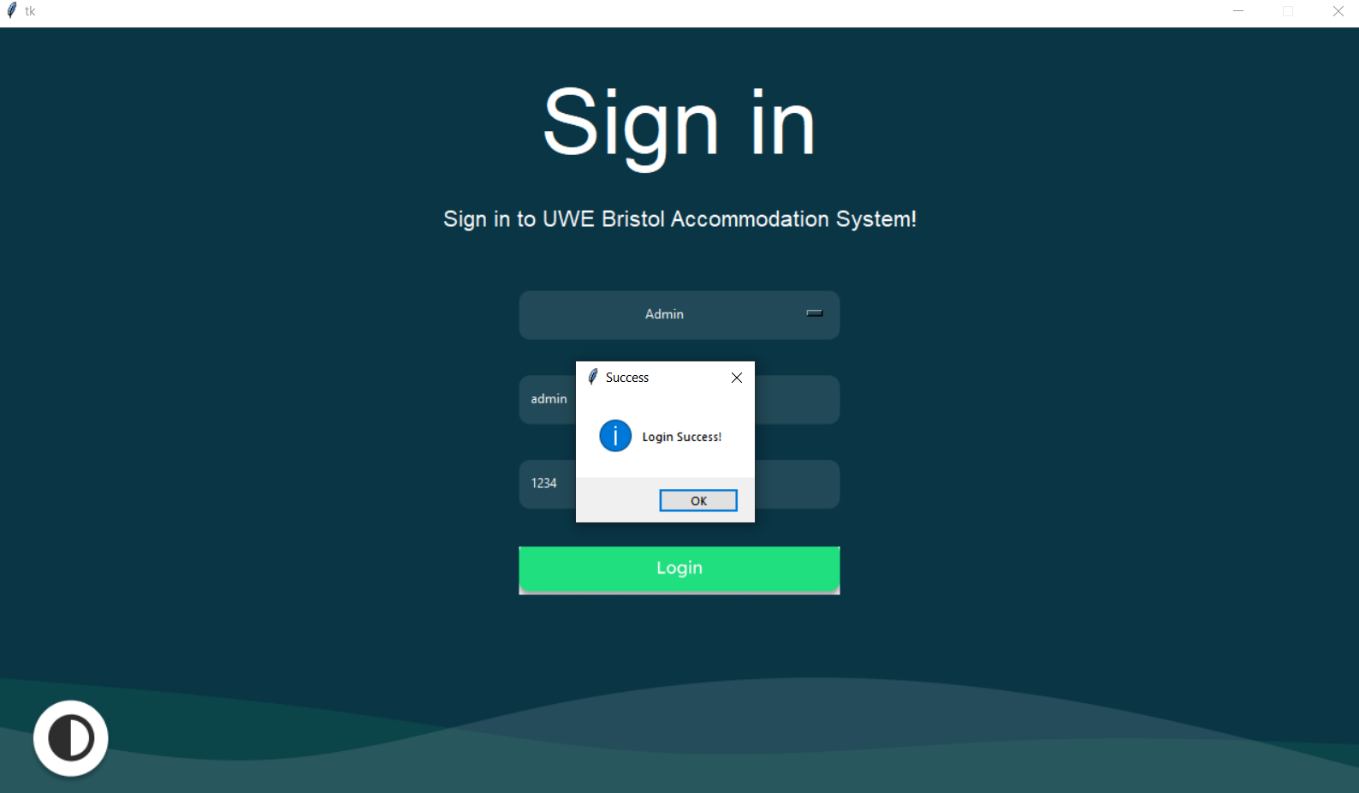
The admin and manager have the same responsibilities, thus only the warden must have limited editting, in this case the function update() which allows only the manager/admin to have a full access, while the warden will not.

The system is designed as when updating the data must be full or the user will get an error message mentioning that all the information must be entered, Room status is updating automatically, when a student is registered or the room is not ready it will be unavailable and vice-versa.

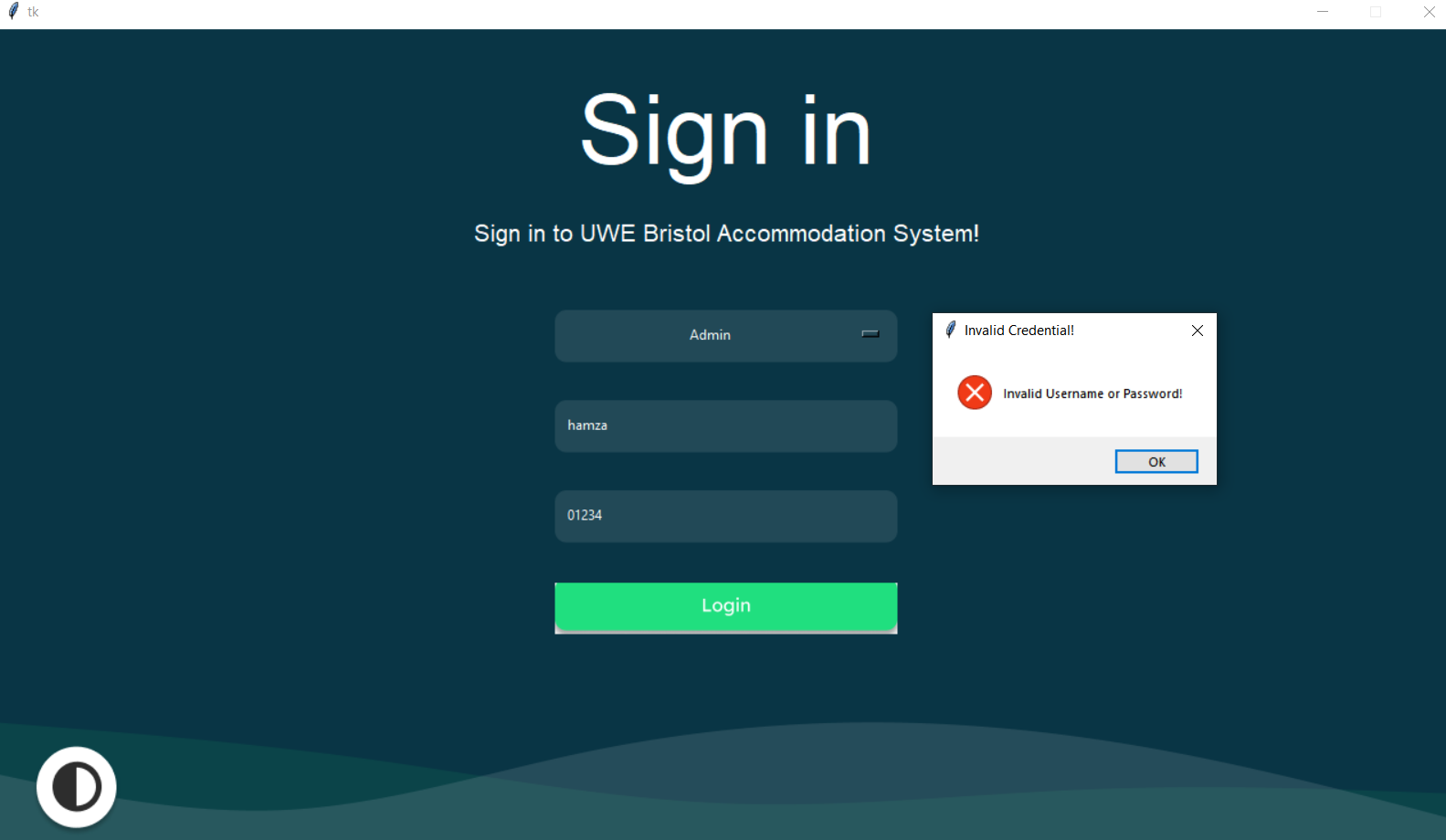
* **Section 4: Final tests**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Purpose** | **Expected result** |
| Login | Successful login | Successful message box display when press “ok” the window appears |
| Wrong Login authentication entered | Users must enter correct credentials | Error message displays |
| Entering data to create Lease | User Fill all the boxes to update | Error message display correctly when boxes are not fully filled |
| Room status dirty or offline | Add lease | Appropriate error message display to warn that the room must be clean to add a lease |

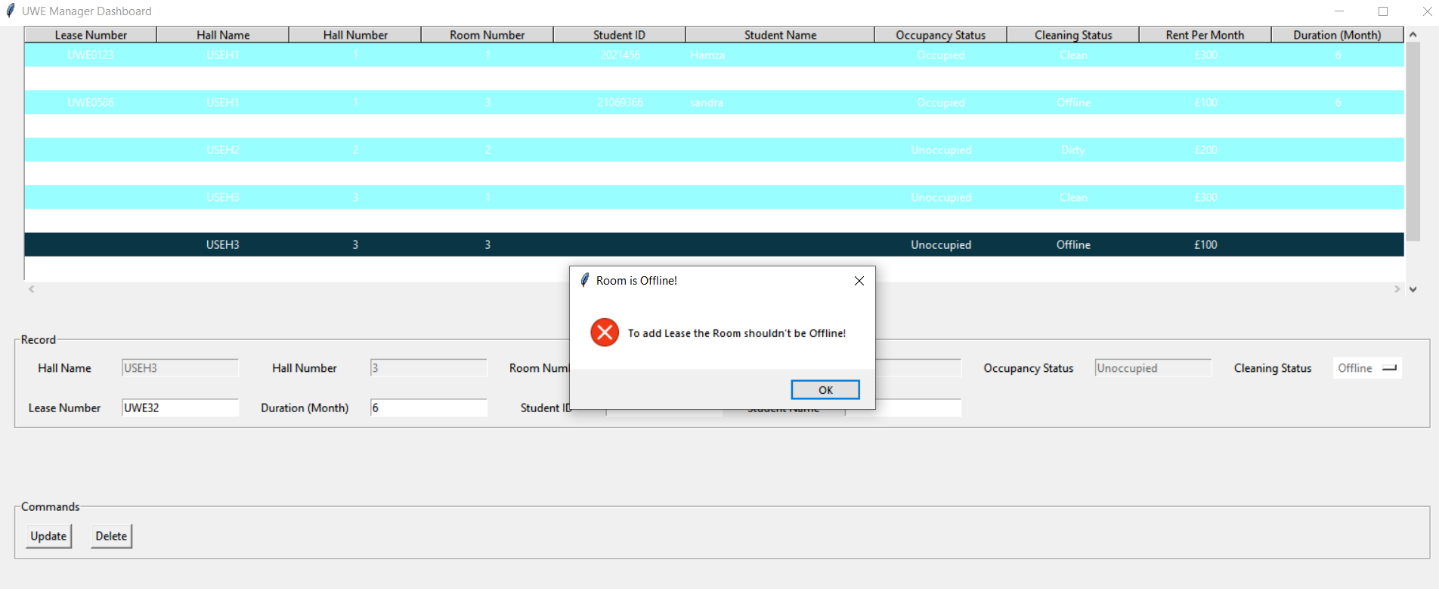
**Login successful:**



**Login error:**

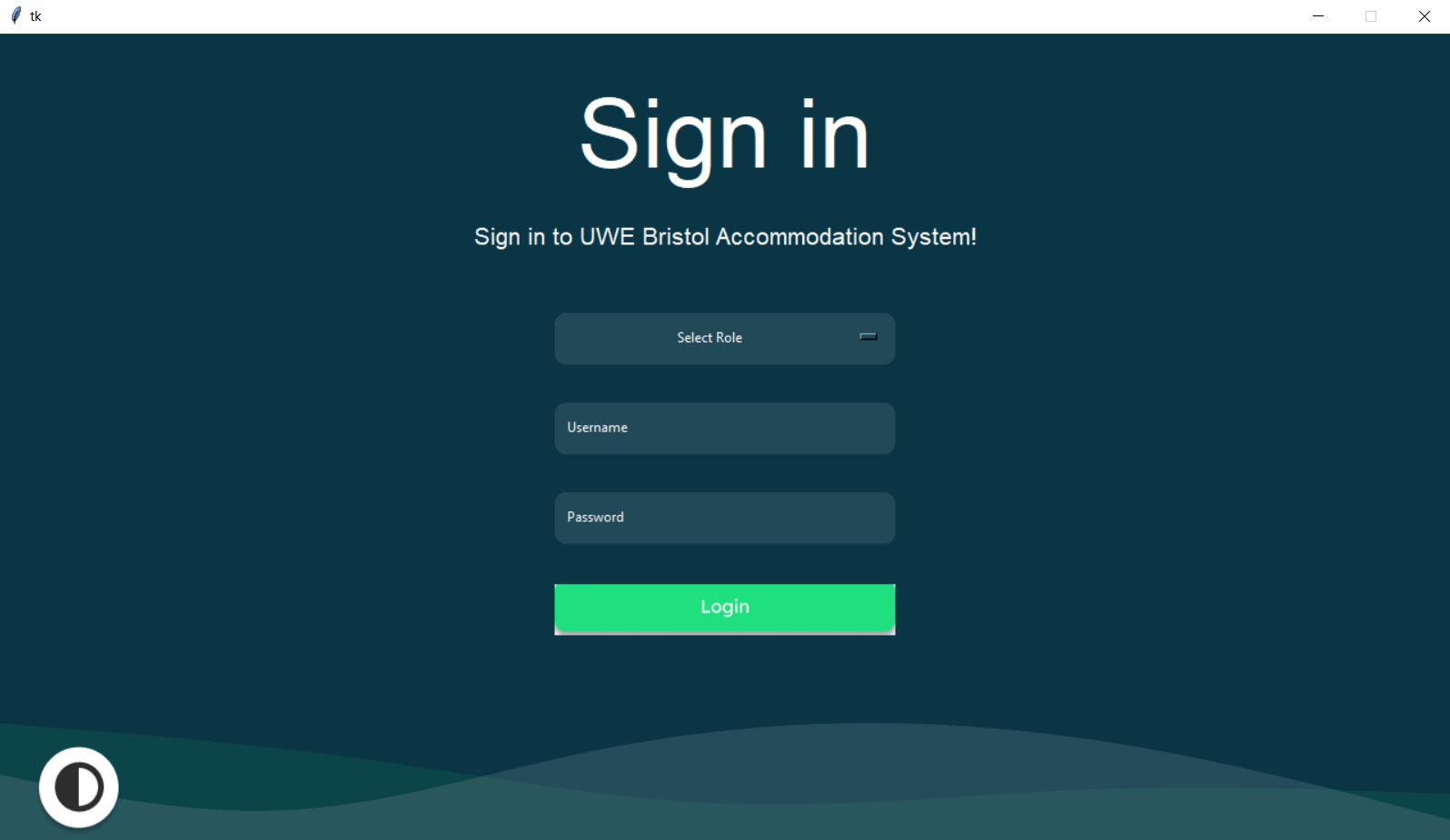


**Update Error:**

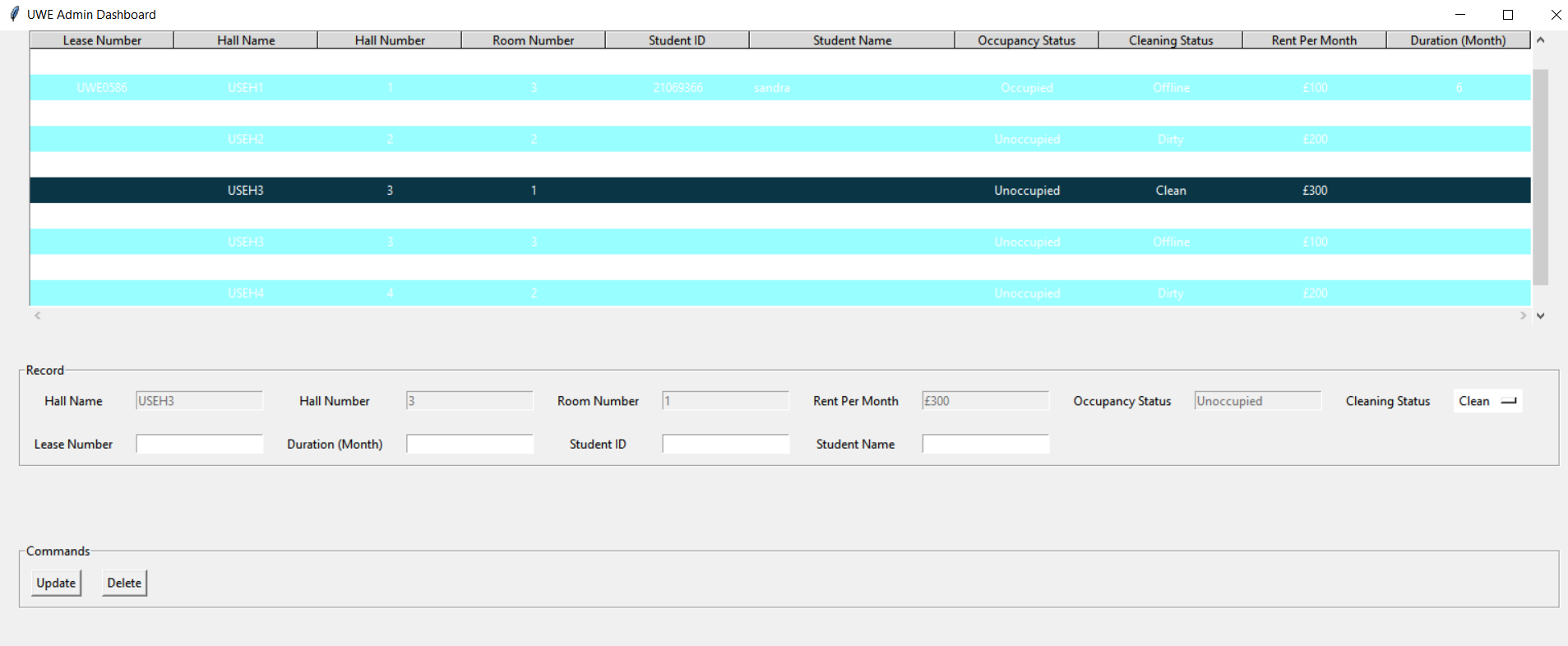


* **Section 5: Documentation**

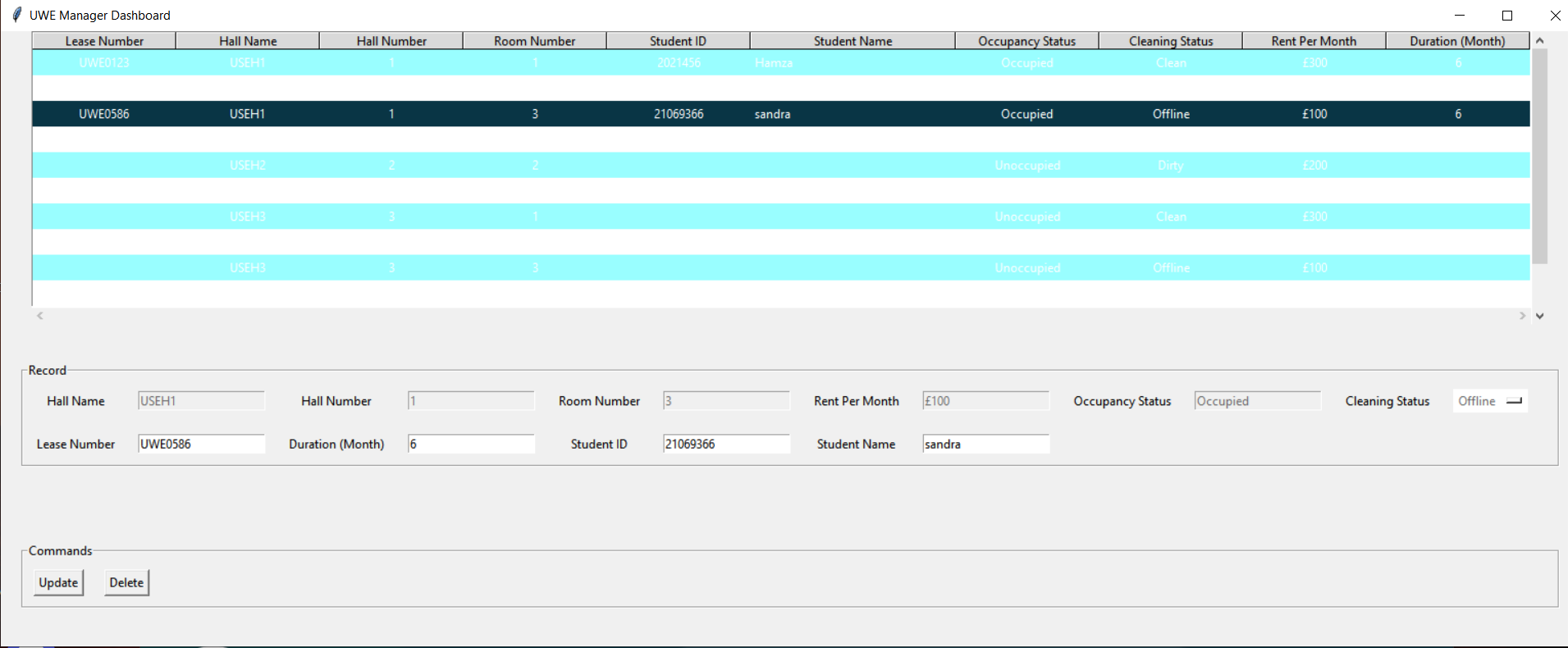
**Login page :**



**Admin view :**



**Manager view :**



**Warden view:**

