**BIT302**

**Software Engineering**

****

**ASSIGNMENT 2**

**Design & Test Document**

**“Web-based Information System for MicroHousing System in Kuala Lumpur”**

**Team Leader:**

**Luh Wulandari Maharani**

**E1700873 / 170030401**

**luhwulandari@gmail.com**

**Member:**

**Rivaldo Bagus Soepardhy**

**E1700882 / 170030400**

**aldobagus@hotmail.co.id**

Table of Content

[Introduction 1](#_Toc35627455)

[Use Case Diagram 2](#_Toc35627456)

[Architectural Design 3](#_Toc35627457)

[Class Diagram Design 4](#_Toc35627458)

[Entity Relationship Diagram (ERD) 5](#_Toc35627459)

[Sitemap 6](#_Toc35627460)

[Wireframe (General UI Design) 7](#_Toc35627461)

[System Sequence Diagram + Contracts 8](#_Toc35627462)

[Database Design 21](#_Toc35627463)

[Iteration 1 22](#_Toc35627464)

[Use Case 22](#_Toc35627465)

[Test Objectives 23](#_Toc35627466)

[Test Plan 24](#_Toc35627467)

[Unit Testing 24](#_Toc35627468)

[Integration Testing 25](#_Toc35627469)

[System Testing 25](#_Toc35627470)

[Test Analysis Report 26](#_Toc35627471)

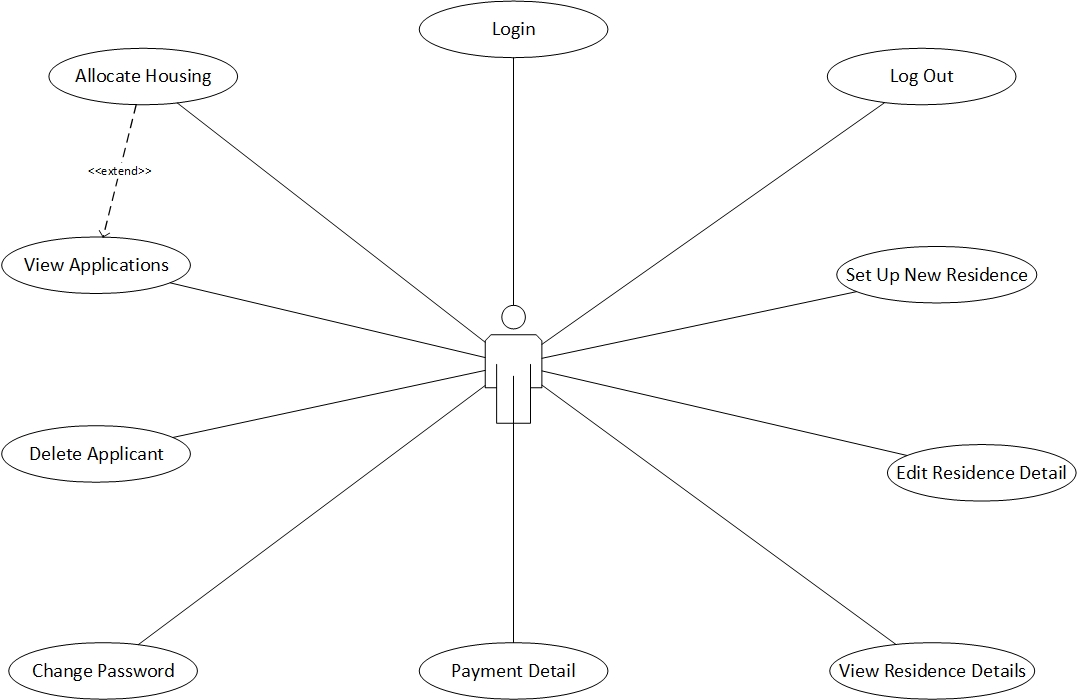
[Updated Gantt Chart, Trello Board, GitHub 27](#_Toc35627472)

Introduction

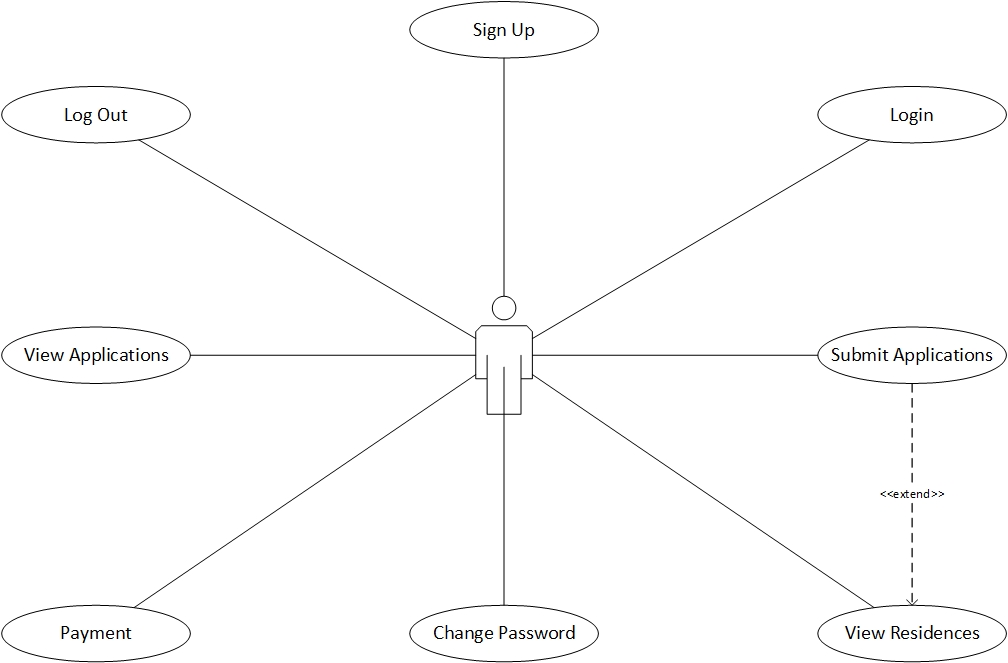
Every software development must have a clear and understandable requirement or specification to ensure developers can keep developing software within the boundary of requirement, while also, developer can adjust to what users need with the software, and developers can optimize the software according to users’ need.

Previously, we already created the general requirement of what kind tools and how our web-based information system will be developed. In this document, we will give you the design specification and other deliverables regarding to the project, such as SSD, ERD, Database Design, Wireframe, etc.

Use Case Diagram

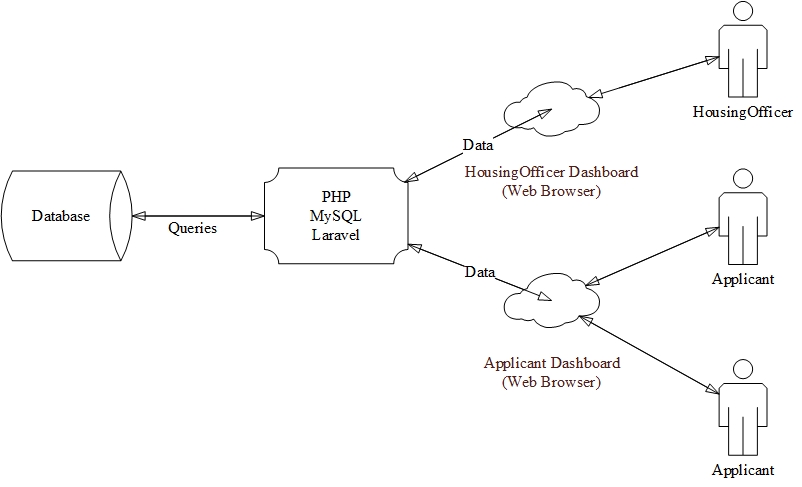


*Use Case Diagram with Actor HousingOfficer*

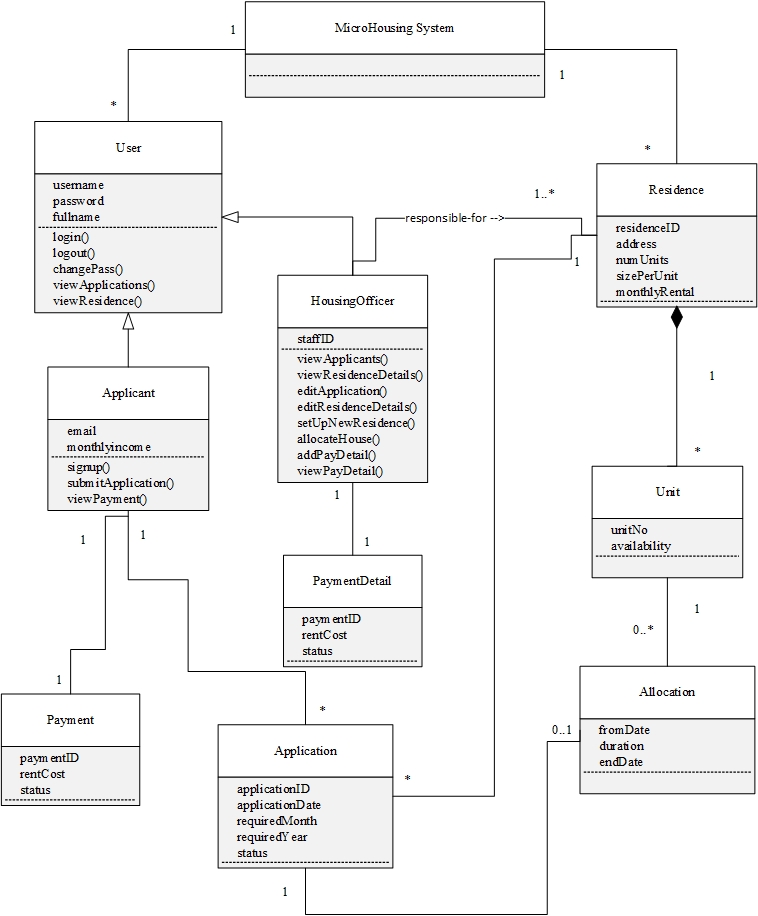


*Use Case Diagram with Actor Applicant*

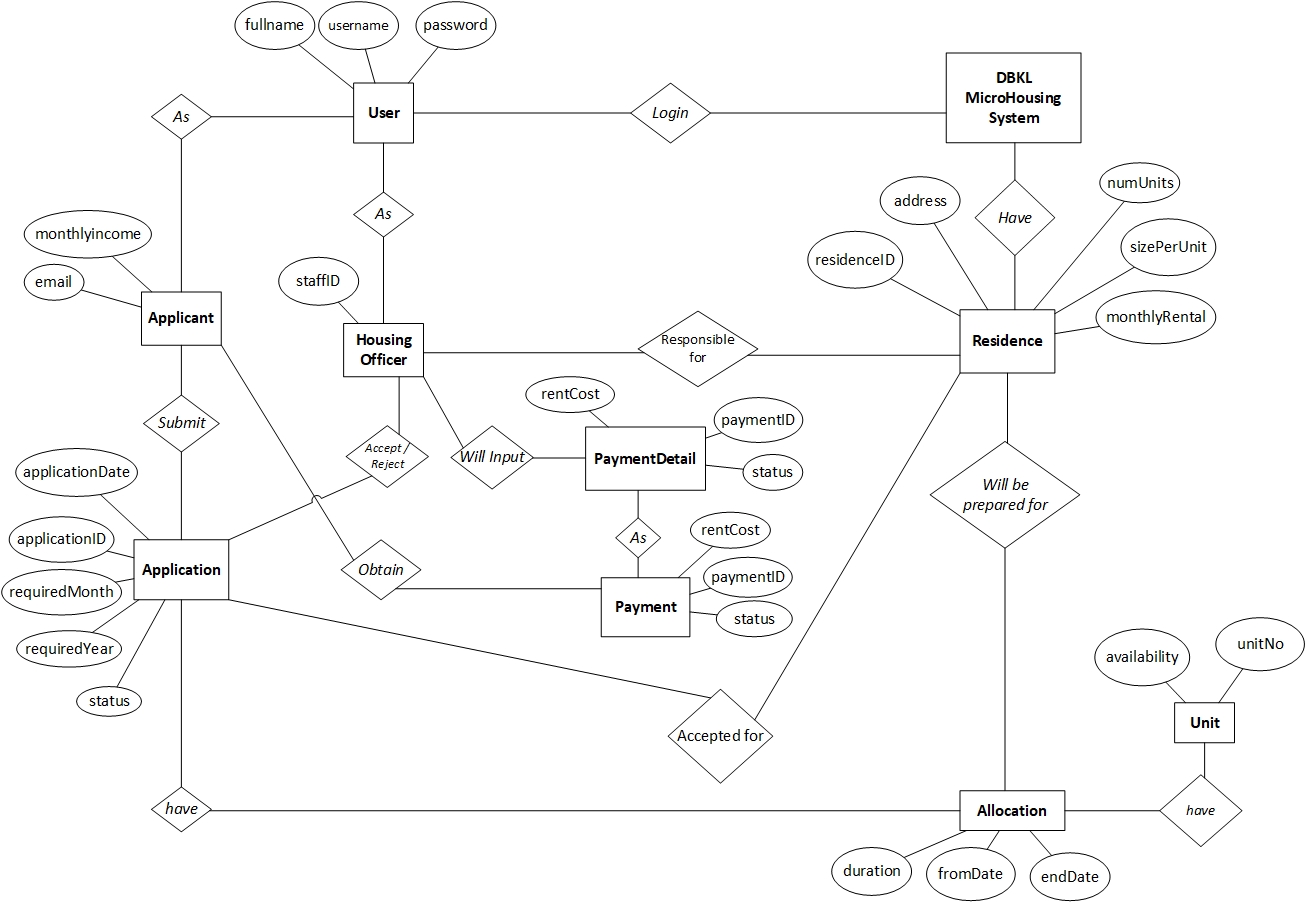
Architectural Design



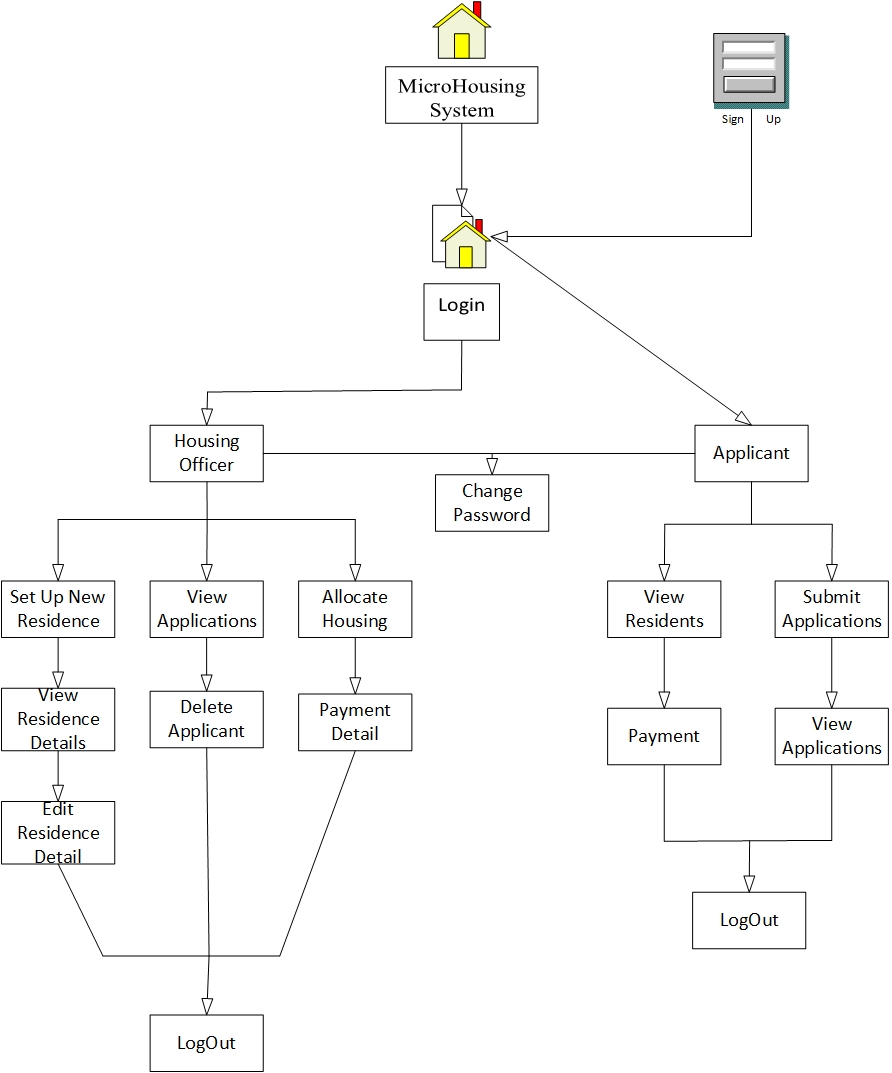
Class Diagram Design



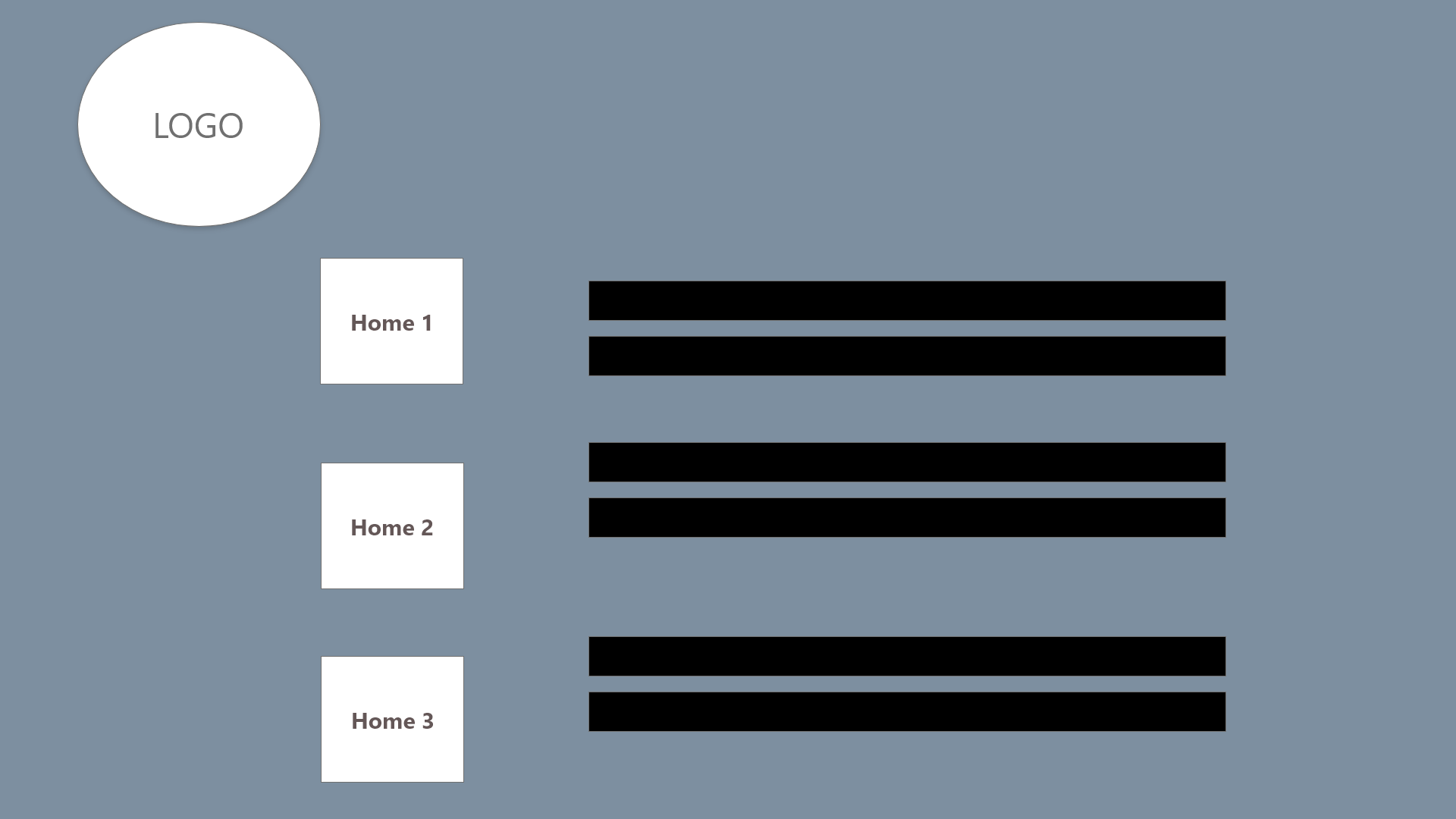
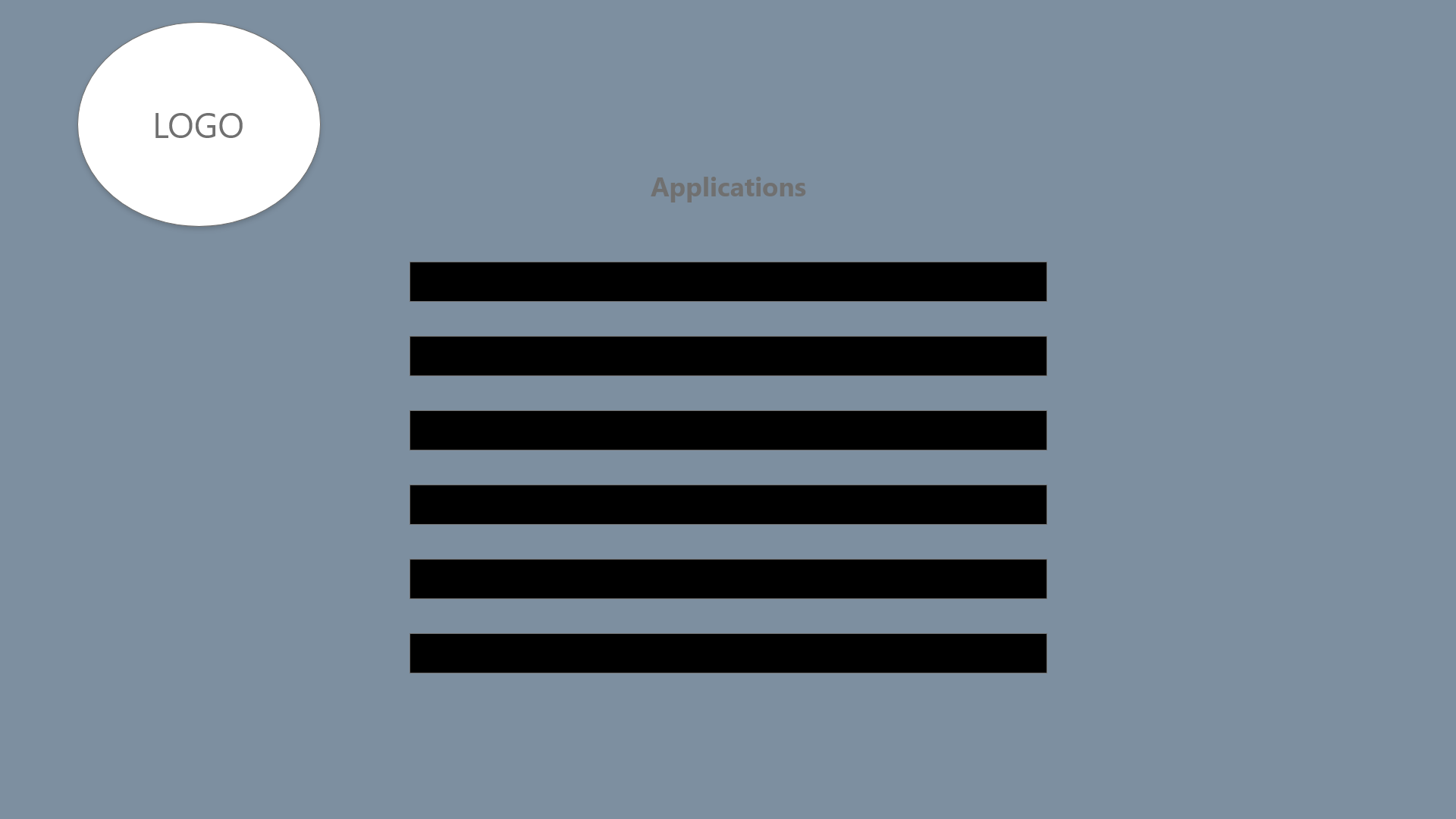
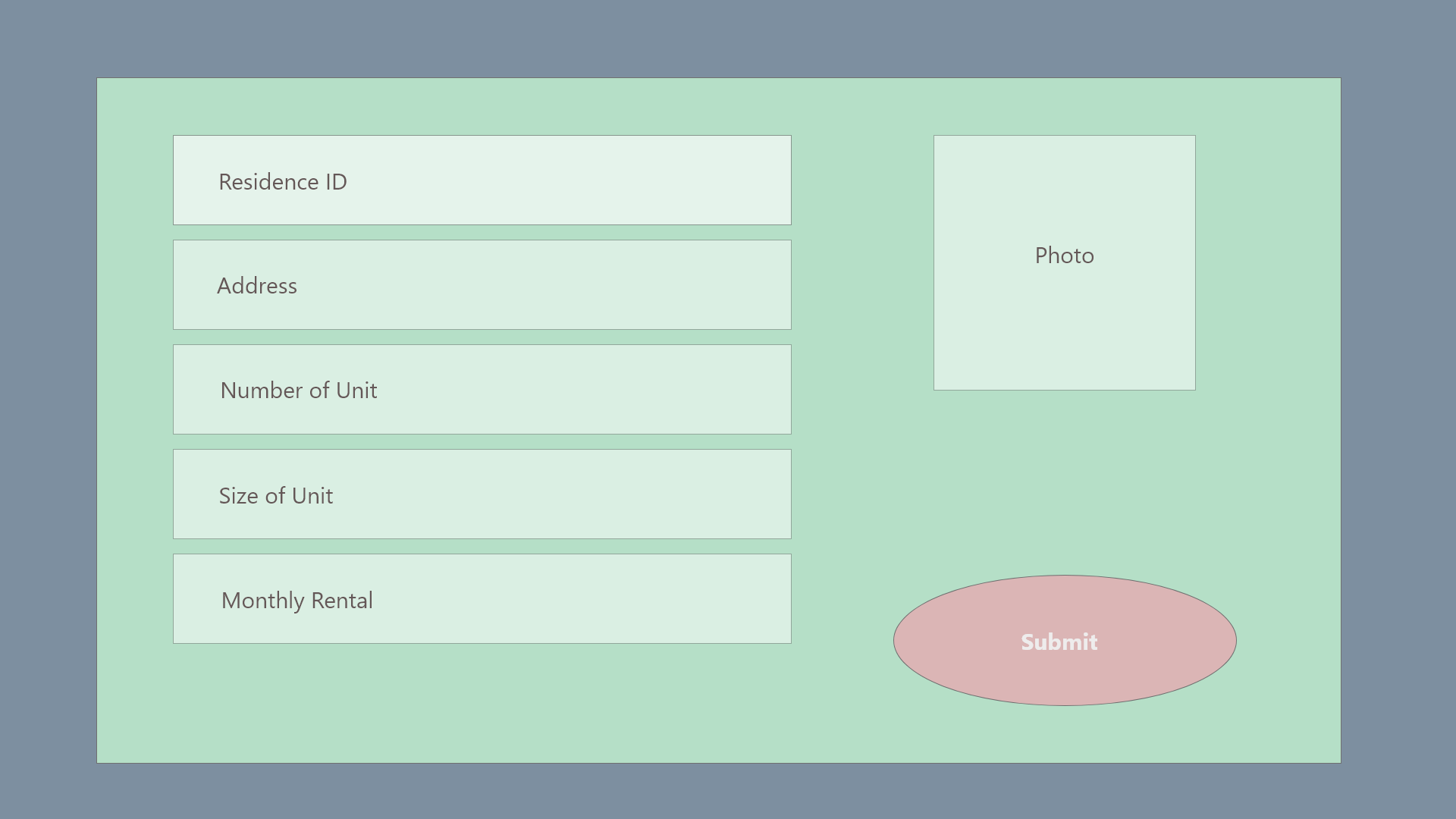
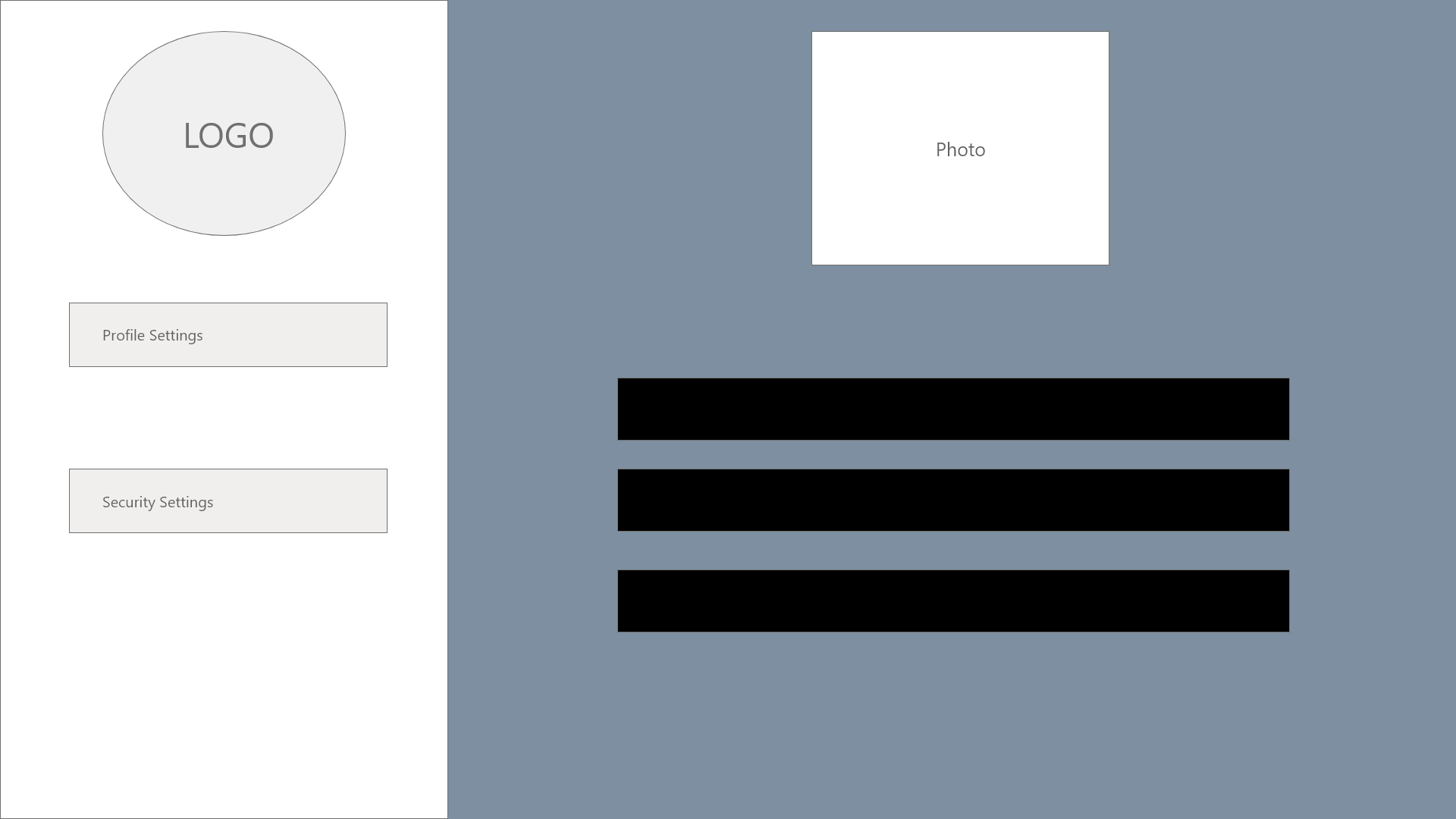
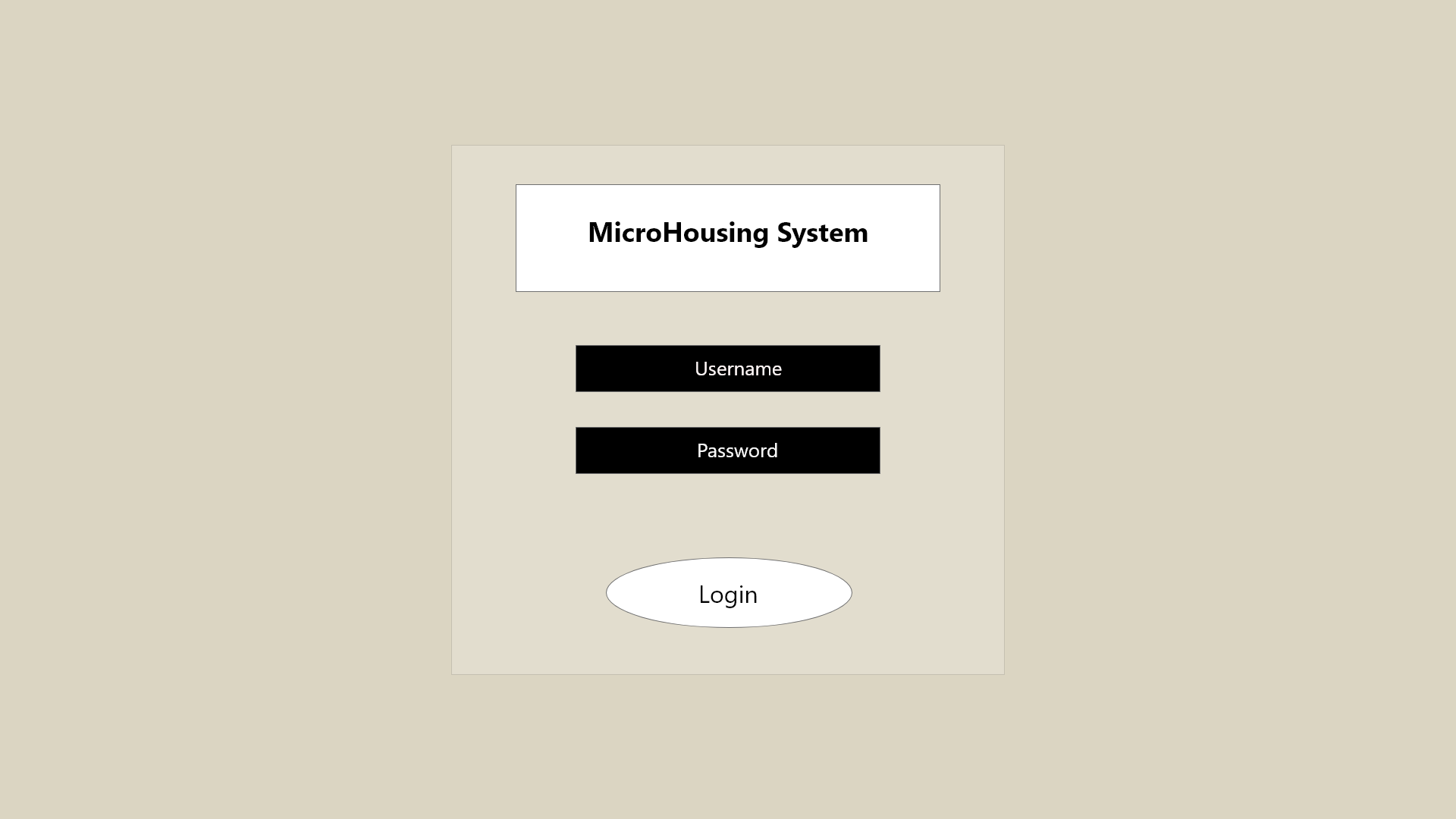
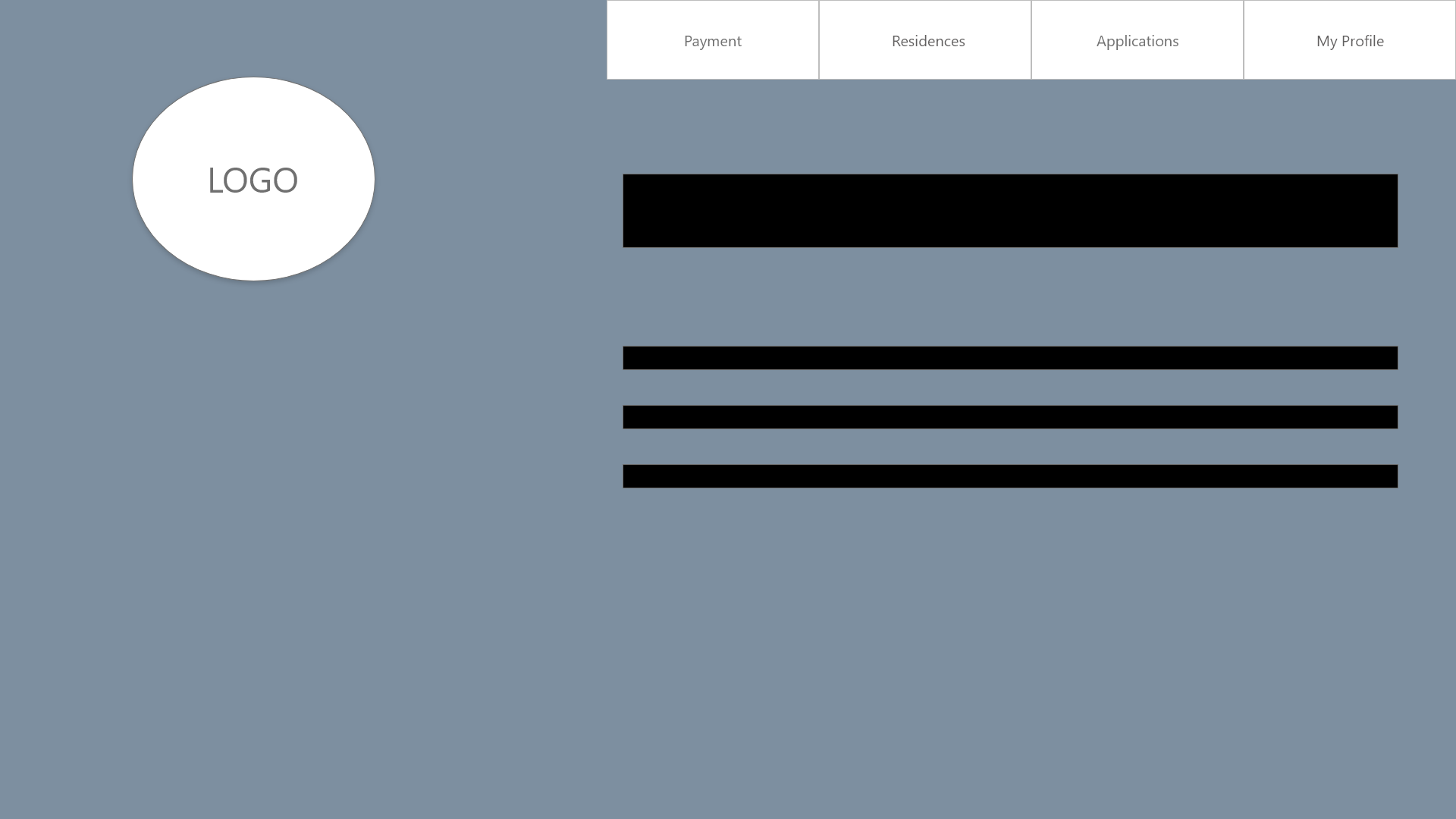
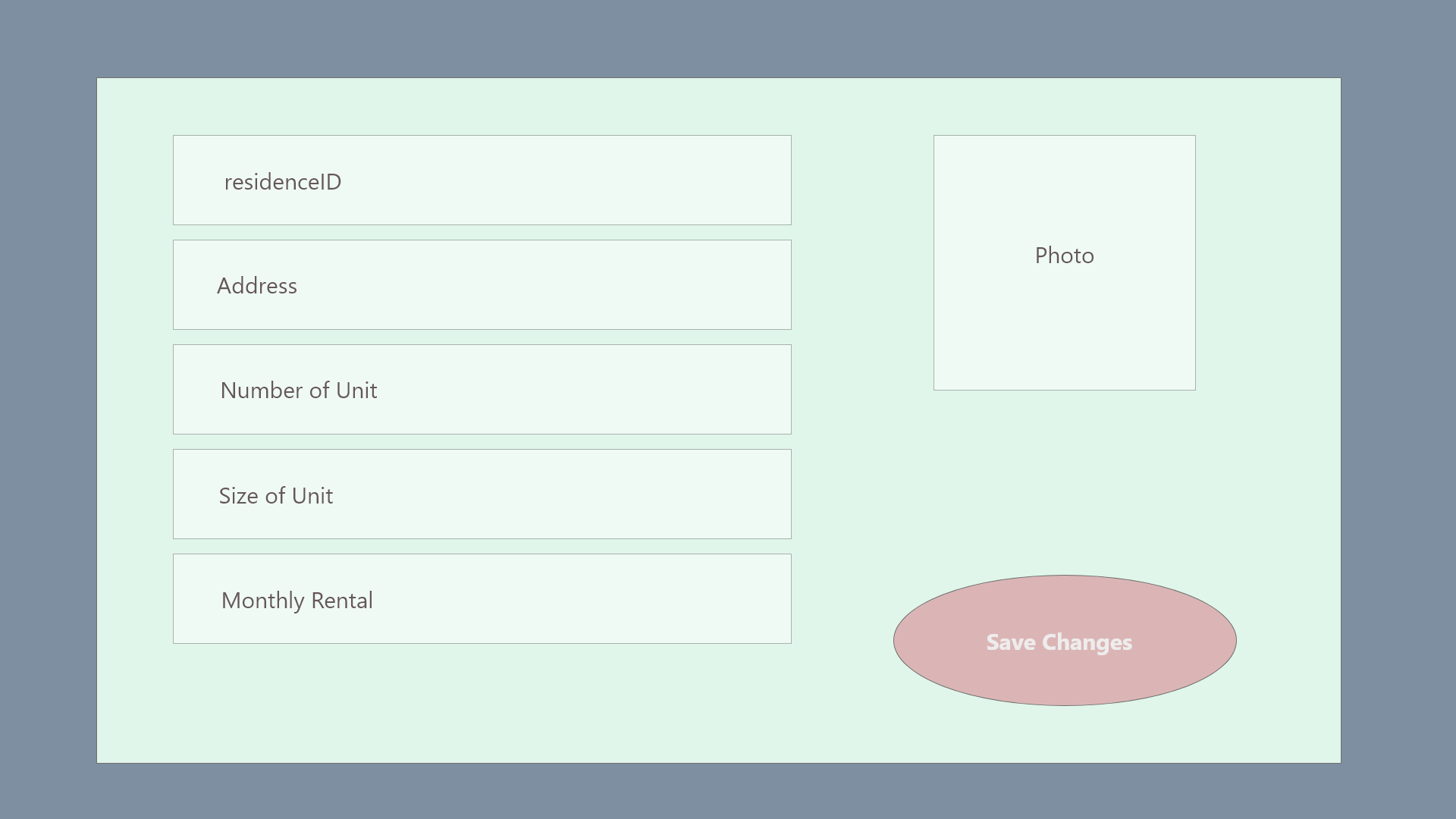
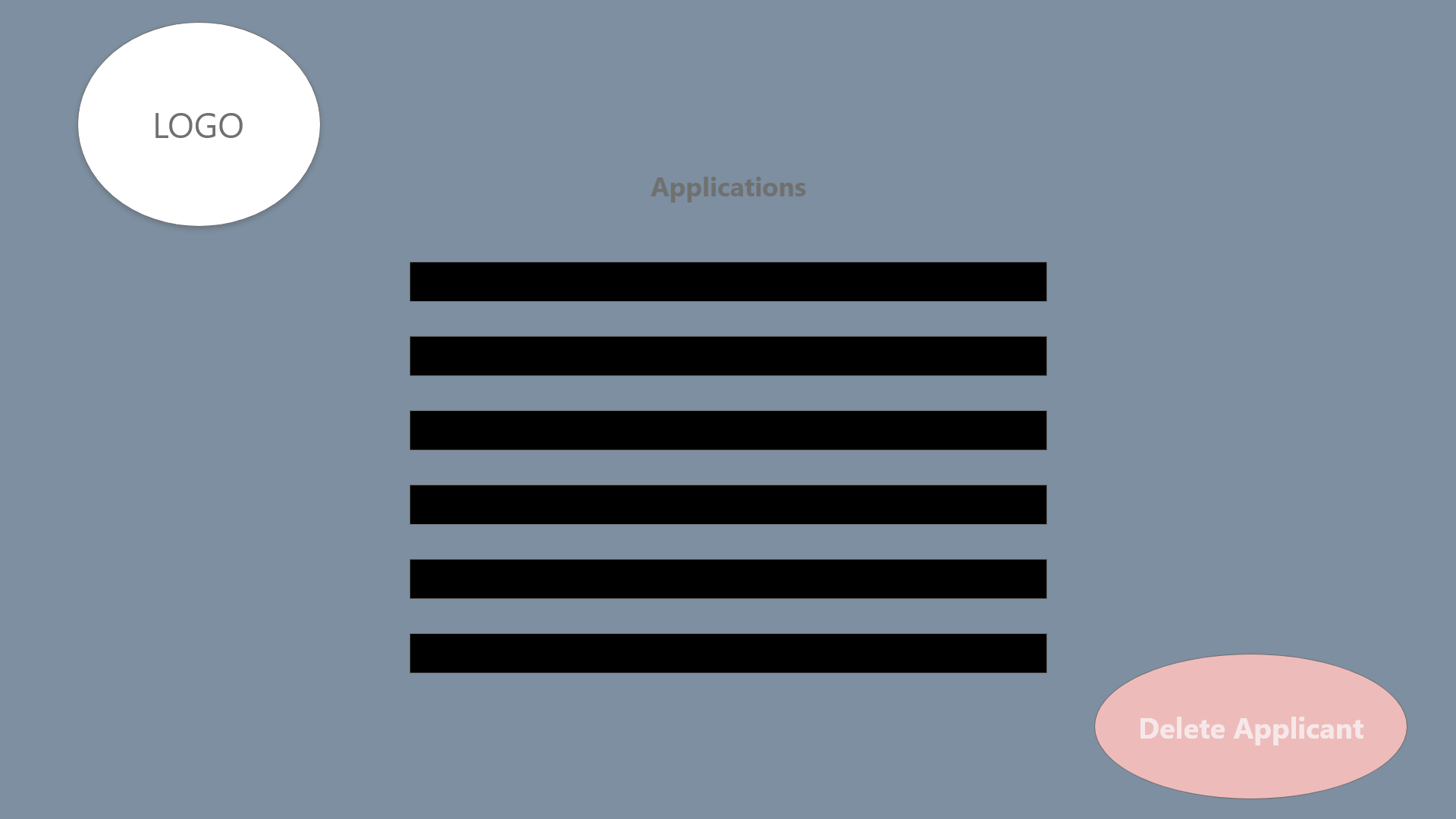
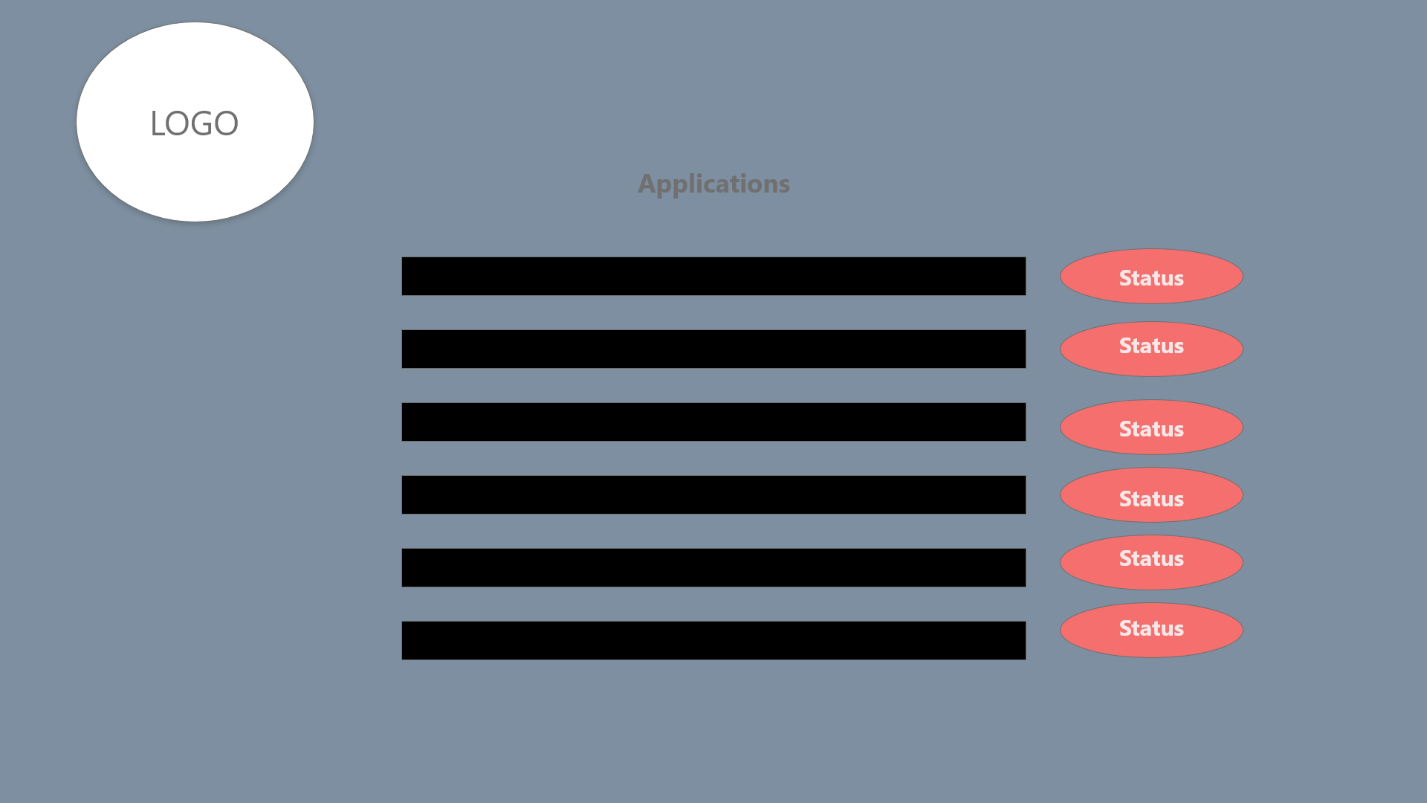
Entity Relationship Diagram (ERD)

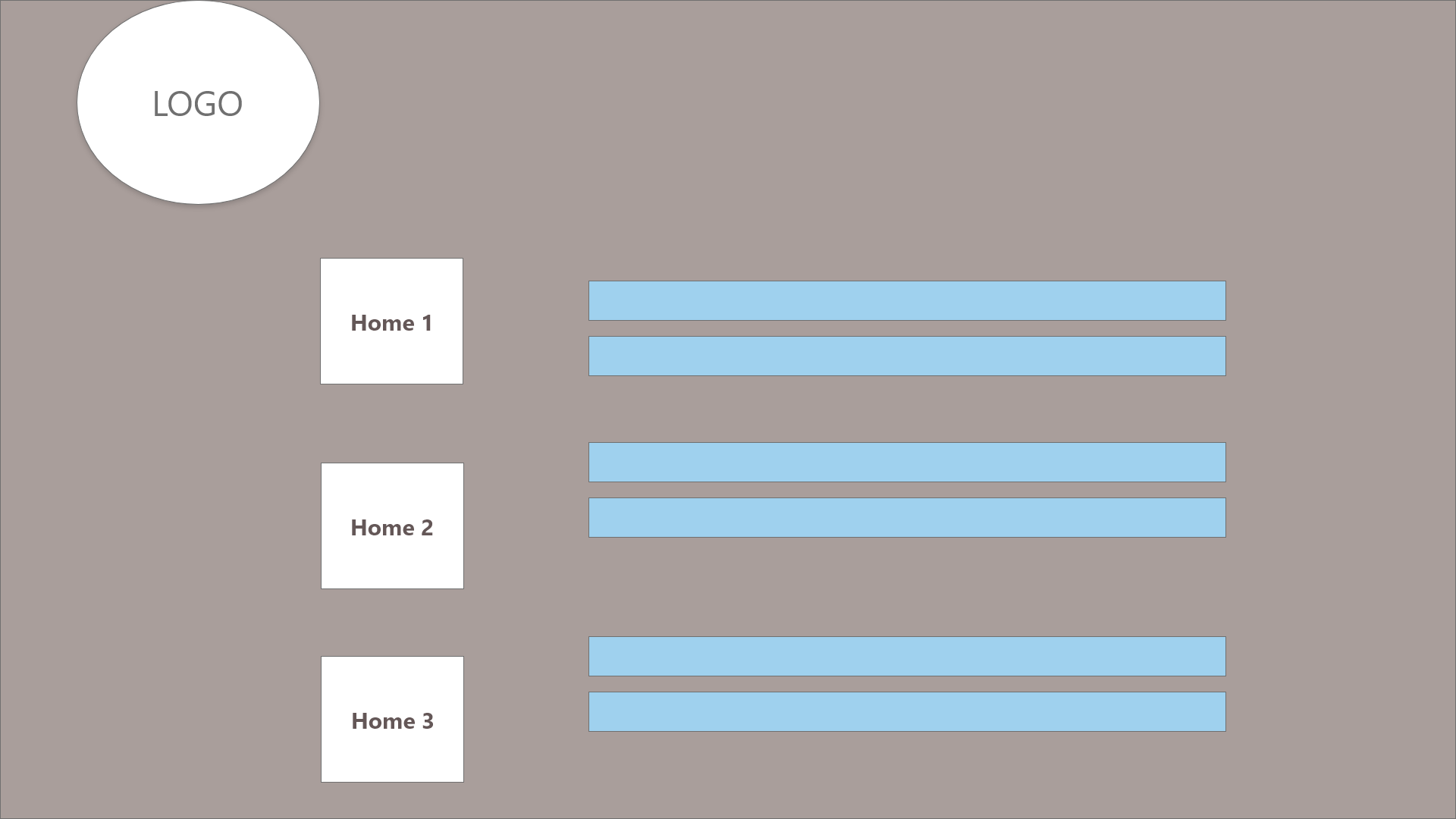
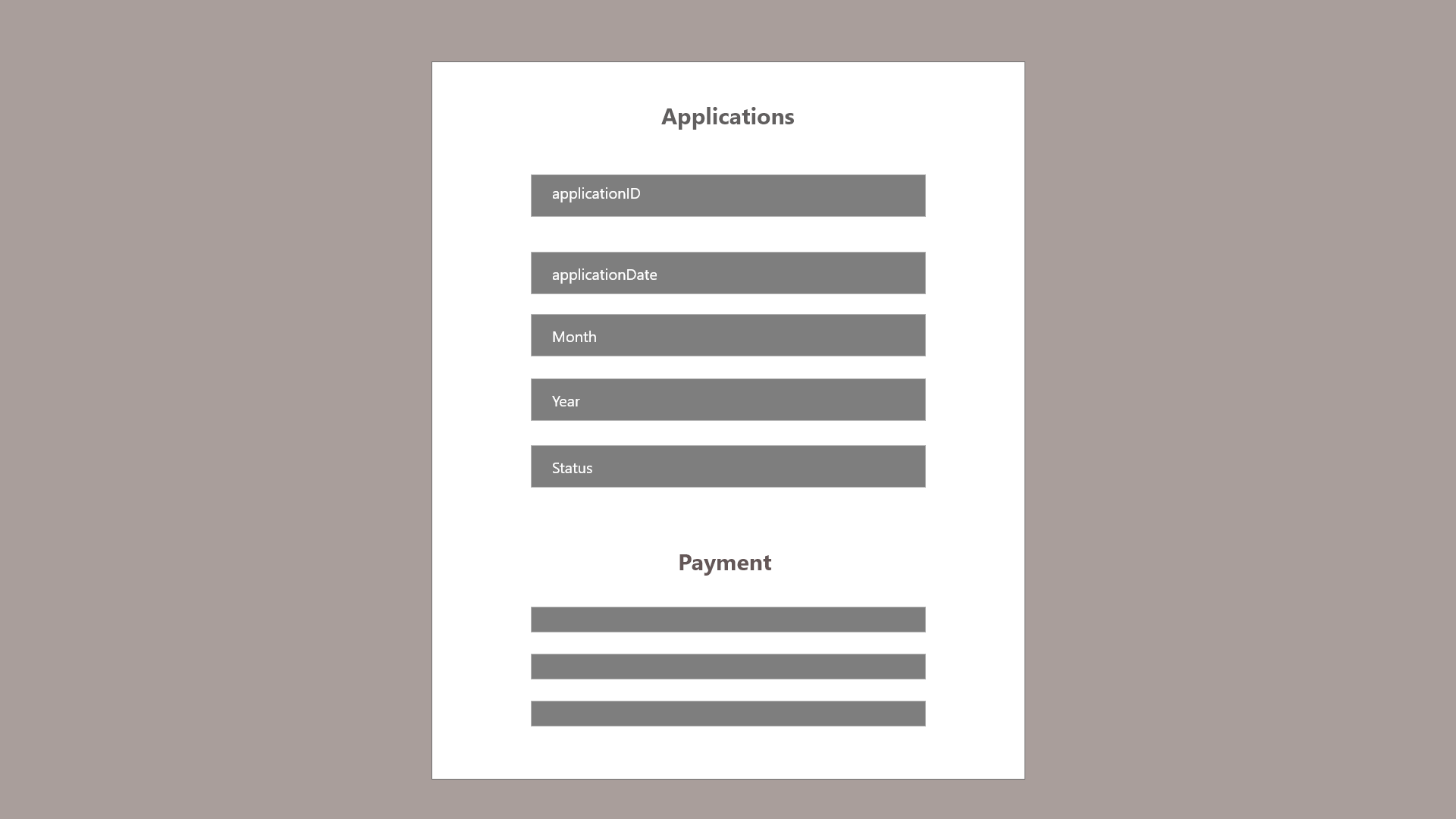
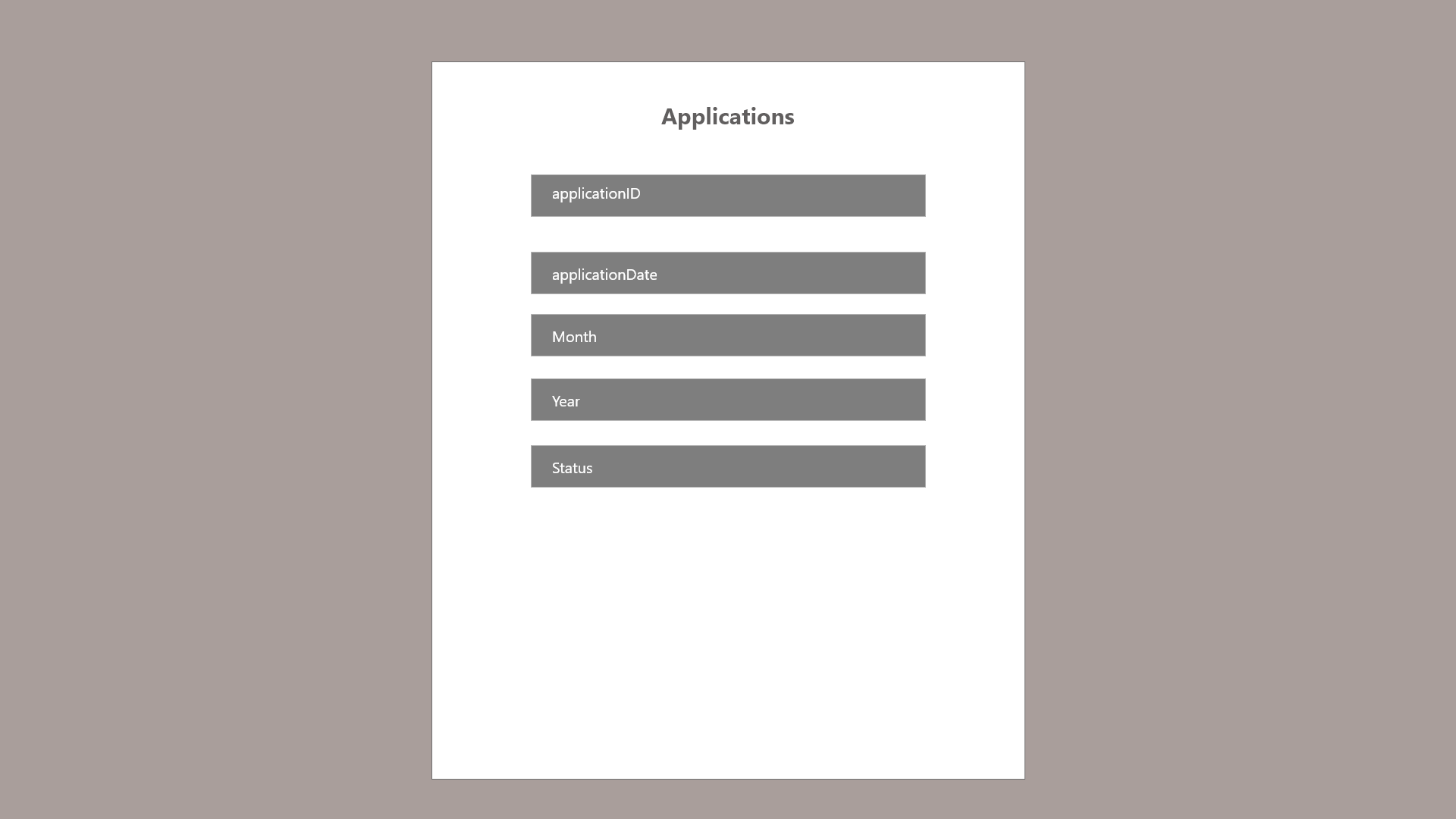
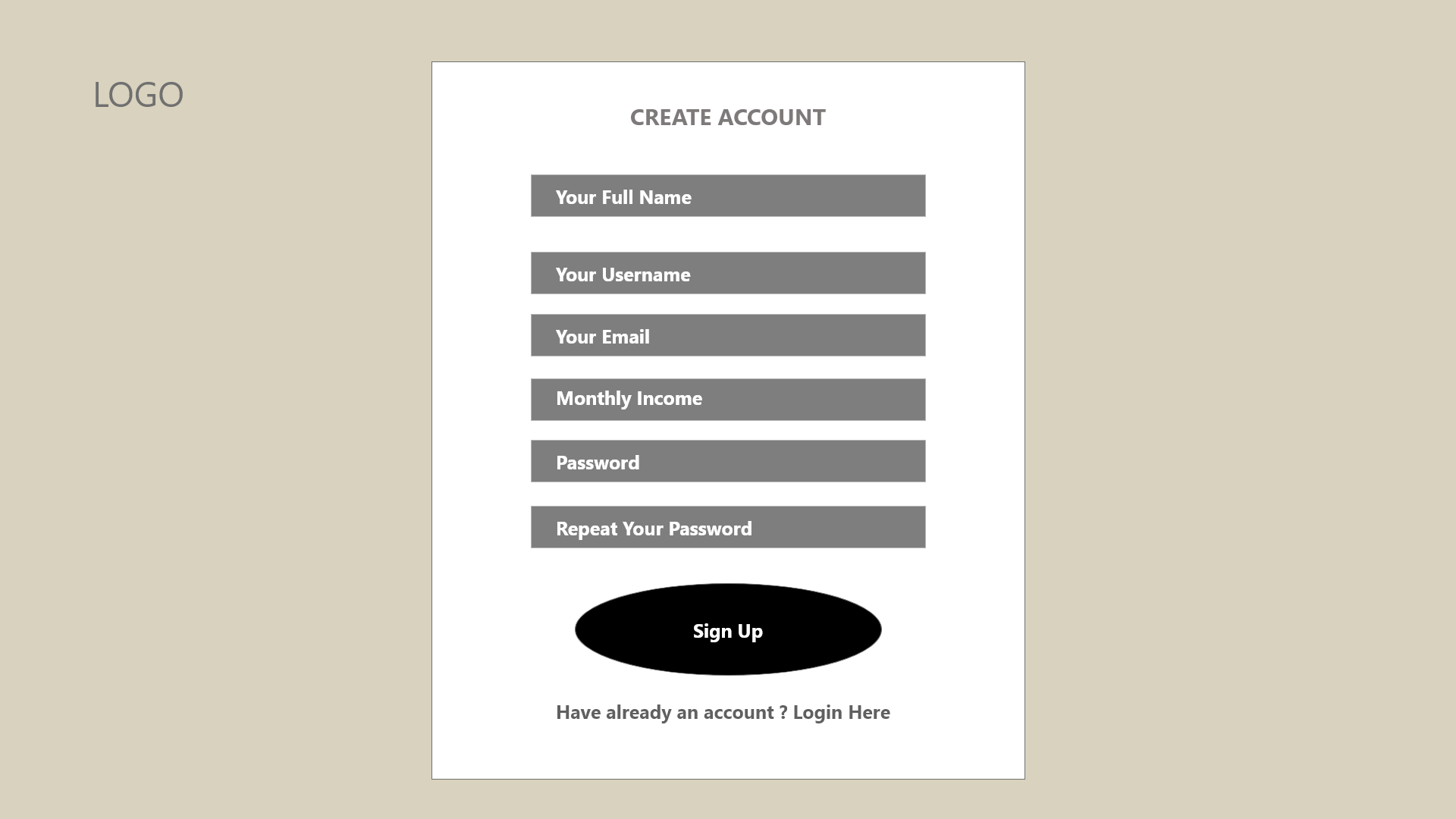
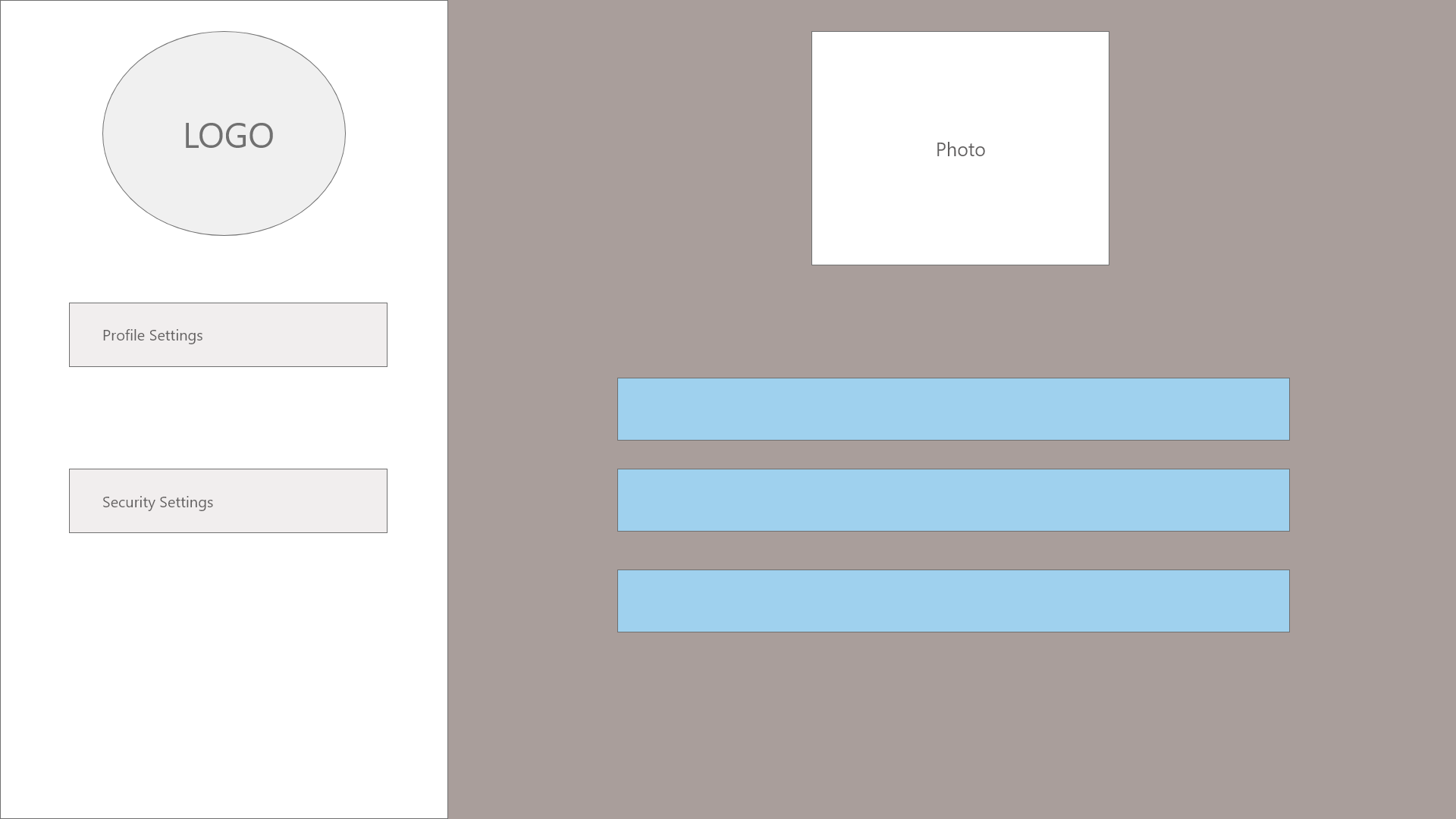
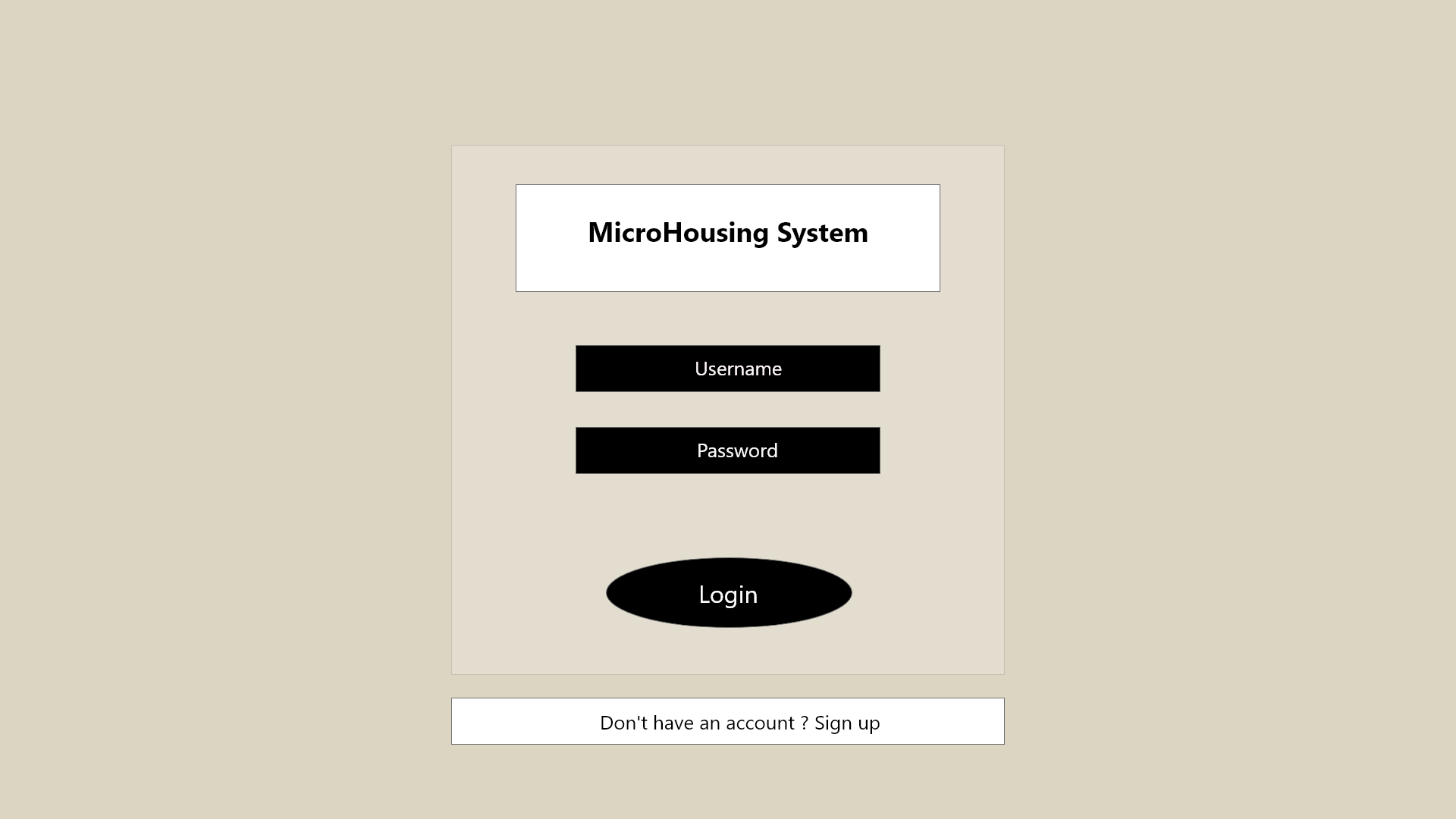
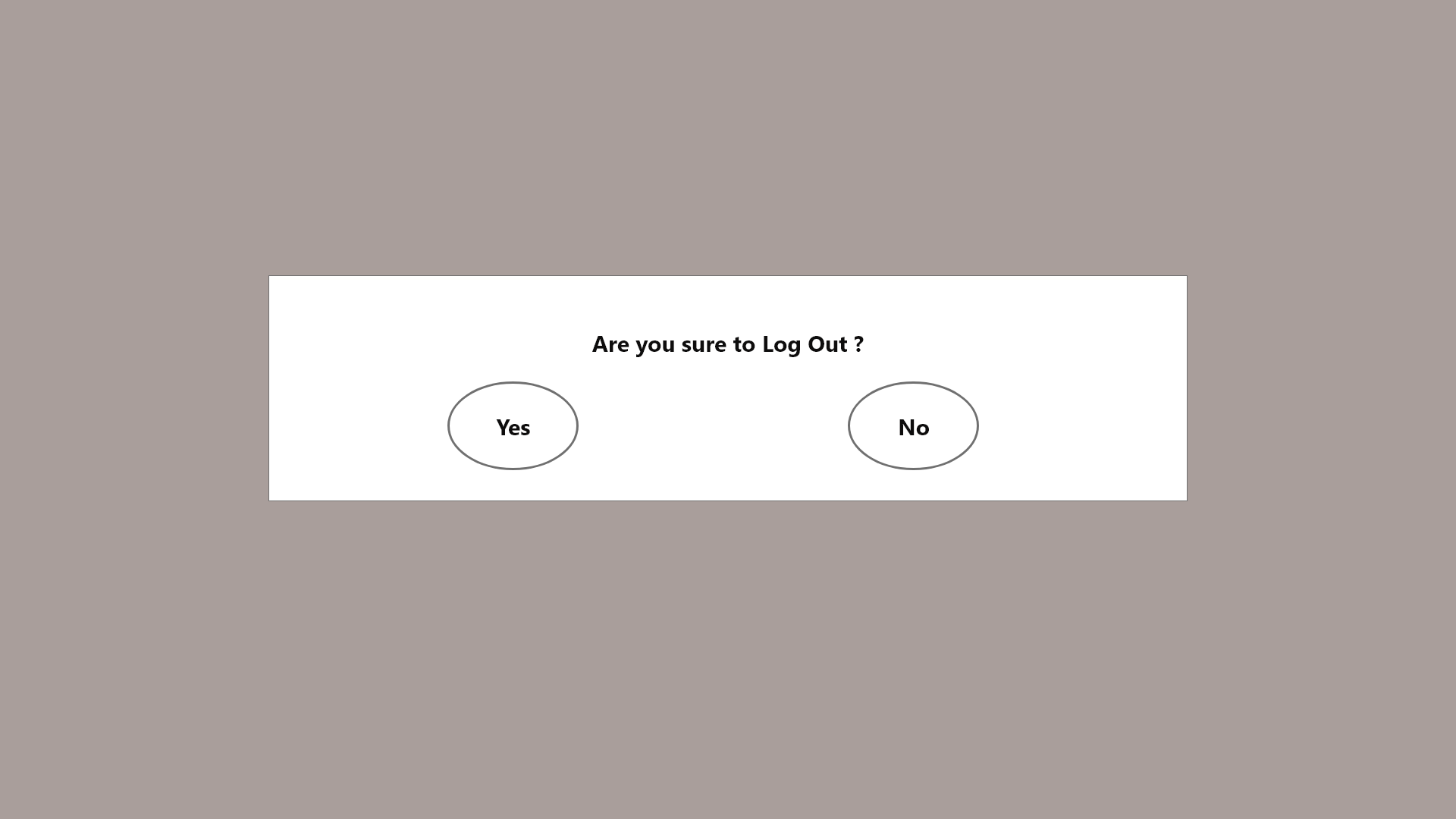
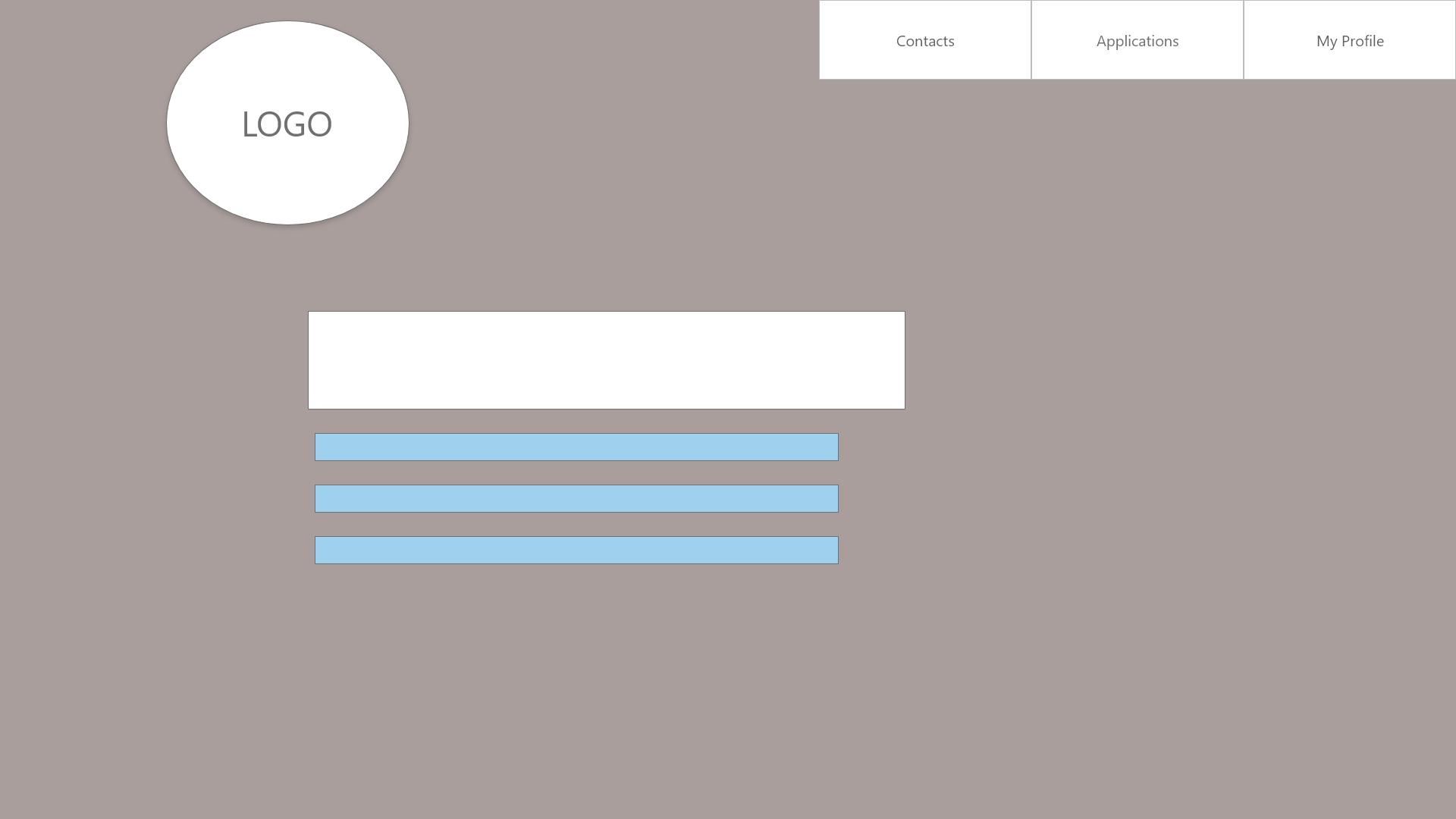
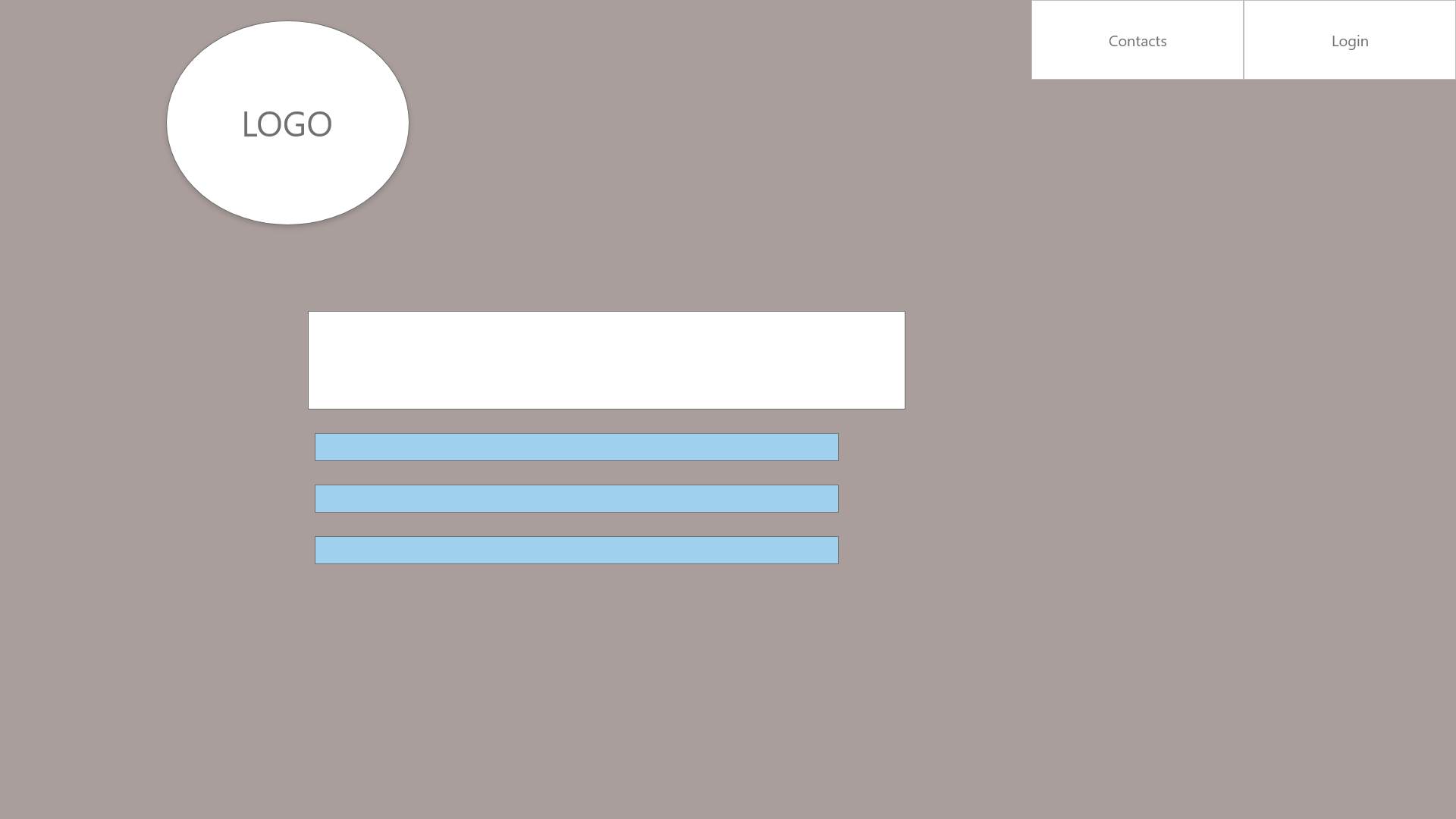


Sitemap



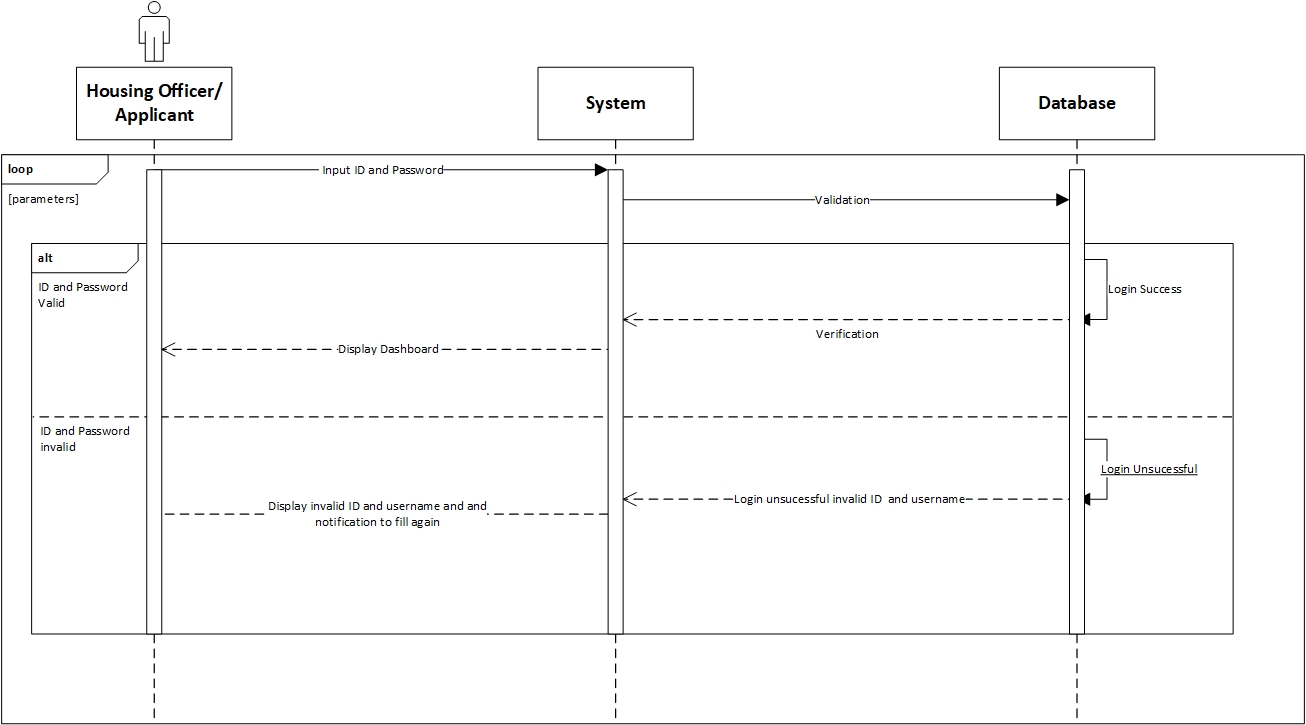
Wireframe



****

System Sequence Diagram + Contracts

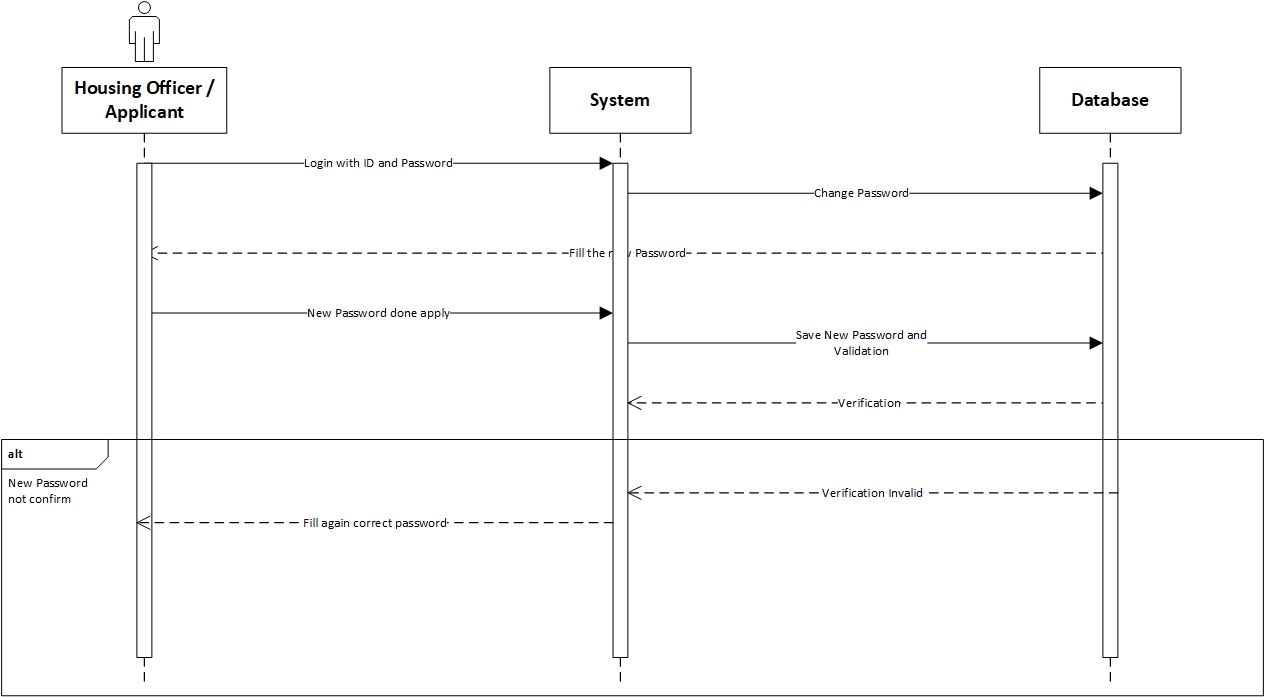
1. **Login**



**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Login** |
| Operation | Login with ID and username |
| Responsible | To access the main page |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |

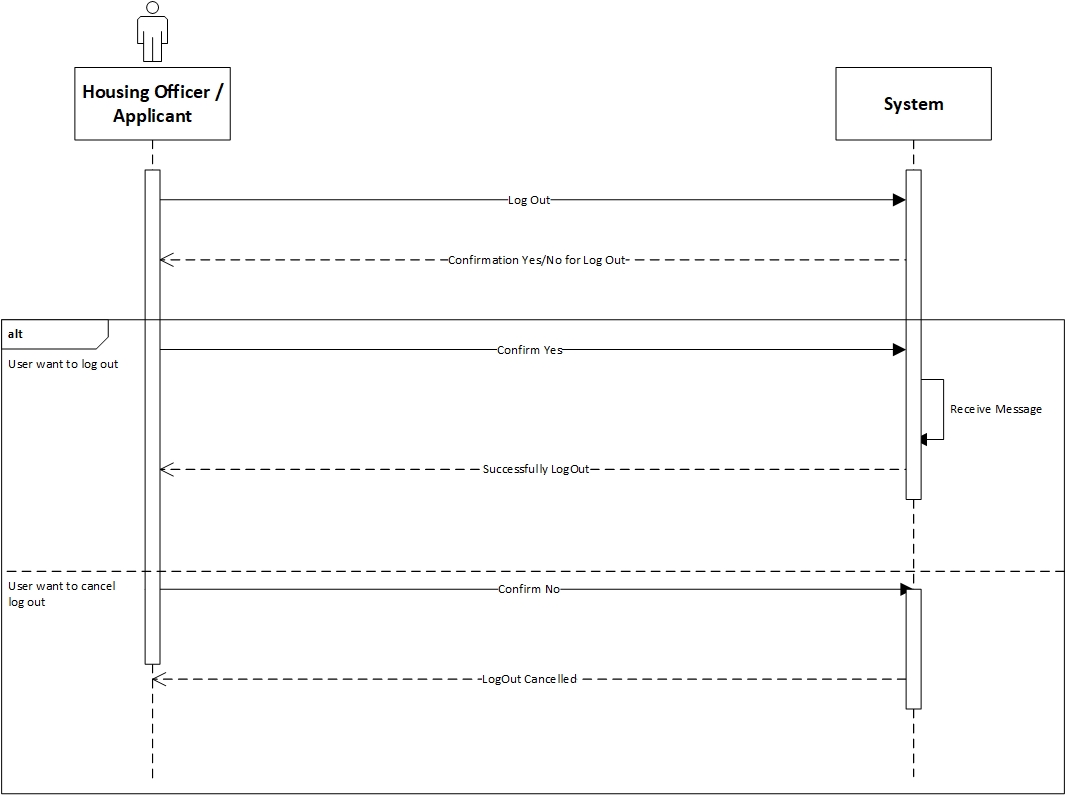
1. **Change Password**



**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Change Password** |
| Operation | Enter current password |
| Responsible | To change the password |
| Pre-conditions | The new password must be available |
| Post-conditions | Successfully changed password |
| **Cross References** | **Change Password** |
| Operation | Enter current password |
| Responsible | To change the password |
| Pre-conditions | The user must be enter the new password again to confirm |
| Post-conditions | Fill again the form with the new password |

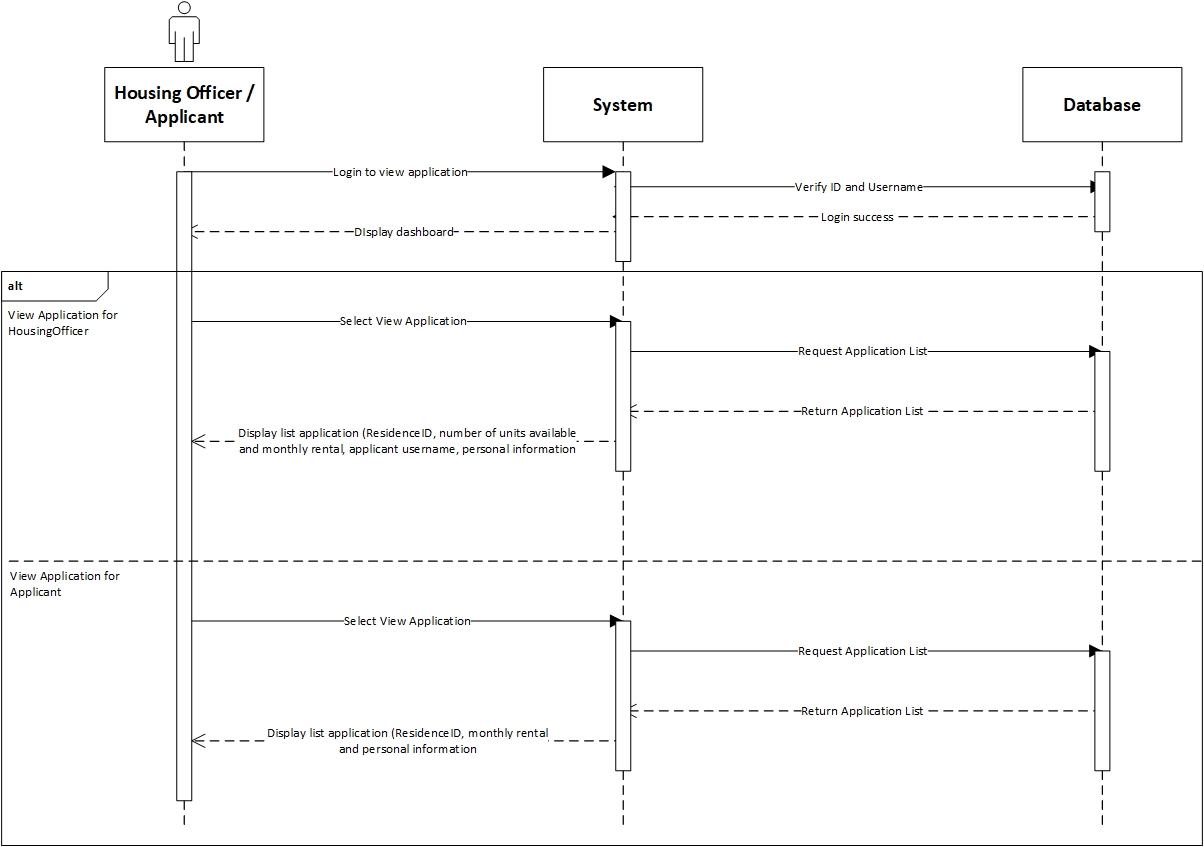
1. **Log Out**



**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Log Out** |
| Operation | User want to log out |
| Responsible | To log out from the system |
| Pre-conditions | The user accepts that they want to log out |
| Post-conditions | Successfully log out |
| **Cross References** | **Log Out** |
| Operation | Cancel to log out |
| Responsible | To cancel log out from the system |
| Pre-conditions | The user cancel log out by click the “No” option |
| Post-conditions | Log out cancelled |

1. **View Applications**



**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **View Applications** |
| Operation | Login with username and password |
| Responsible | To get access to view the applications |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **View Applications** |
| Operation | View Application in Housing Officer |
| Responsible | To view a list for the Residence that Housing Officer is responsible |
| Pre-conditions | The application object must be available |
| Post-conditions | The list of application with status for the Residence that the Housing Officer is responsible, showing the residence ID, number of units available, monthly rental, application username and personal information |

|  |  |
| --- | --- |
| **Cross References** | **View Applications** |
| Operation | View Application in Applicant |
| Responsible | To view a list of application that have been made for applicant |
| Pre-conditions | The application object must be available |
| Post-conditions | The list of application that have been made by the applicant, showing the residence ID, monthly rental and personal information |

1. **View Residences**

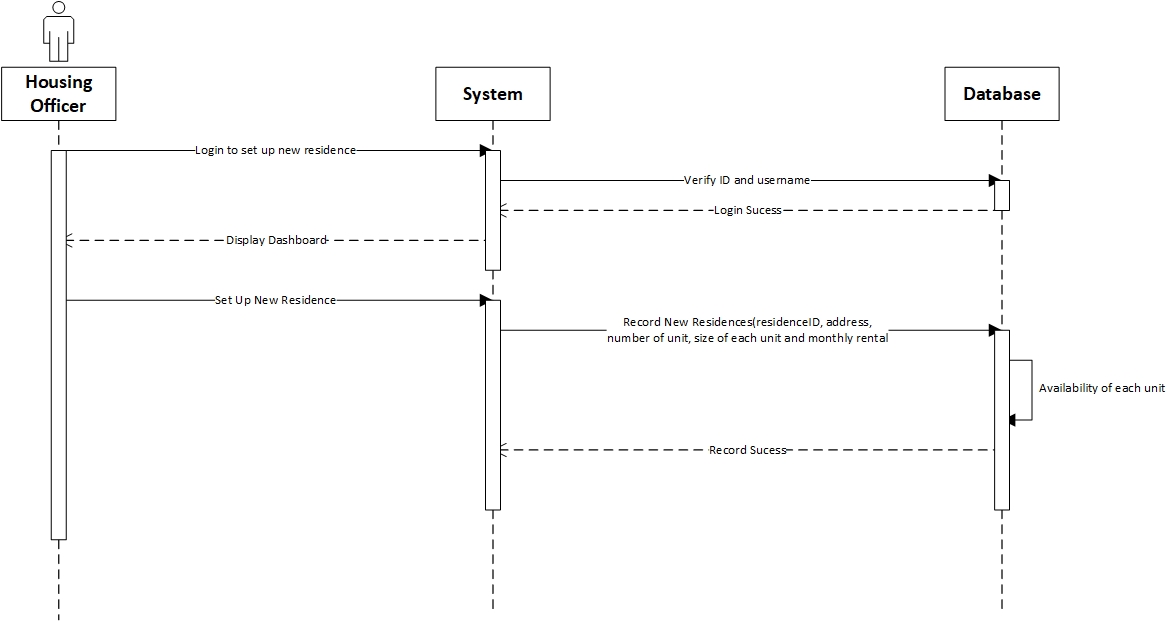


**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **View Residences** |
| Operation | Login with ID and username |
| Responsible | To access the main page |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |

|  |  |
| --- | --- |
| **Cross References** | **View Residences** |
| Operation | Click view all residences |
| Responsible | To get all list of the residences |
| Pre-conditions | The residence object must be available |
| Post-conditions | Success to display all residences of all residences |
| **Cross References** | **View Residences** |
| Operation | Select the residence to view |
| Responsible | To get information about selected residences |
| Pre-conditions | The residence ID must be available |
| Post-conditions | Success to display detail information about the selected residences |

1. **Set Up New Residences**

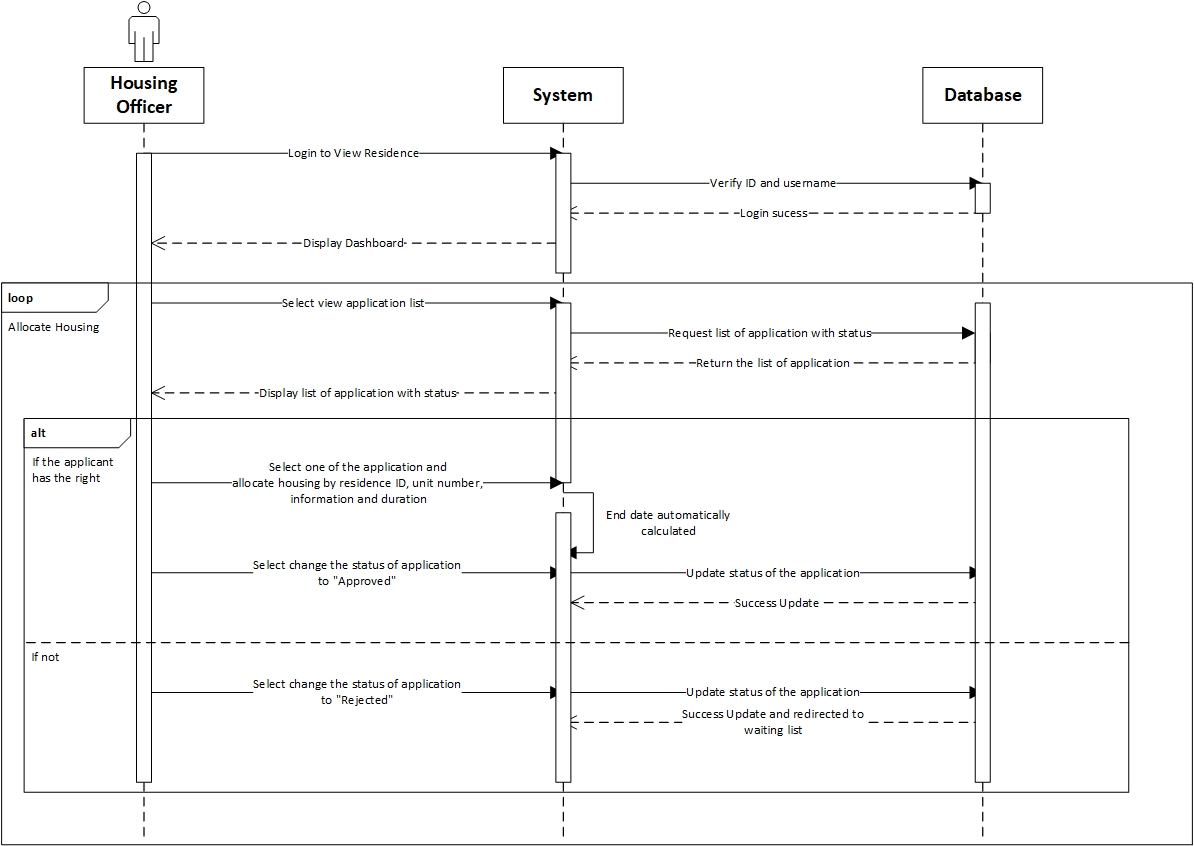


**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Set Up New Residences** |
| Operation | Login with ID and username |
| Responsible | To access the main page |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |

|  |  |
| --- | --- |
| **Cross References** | **Set Up New Residences** |
| Operation | Set up new residences |
| Responsible | To set up the new residences by input the residenceID, address, number of units available, size of each unit and monthly rental |
| Pre-conditions | Object residenceID, address, number of units available, size of each unit and monthly rental must be available |
| Post-conditions | New residences was successfully added to the system |

1. **Allocate Housing**

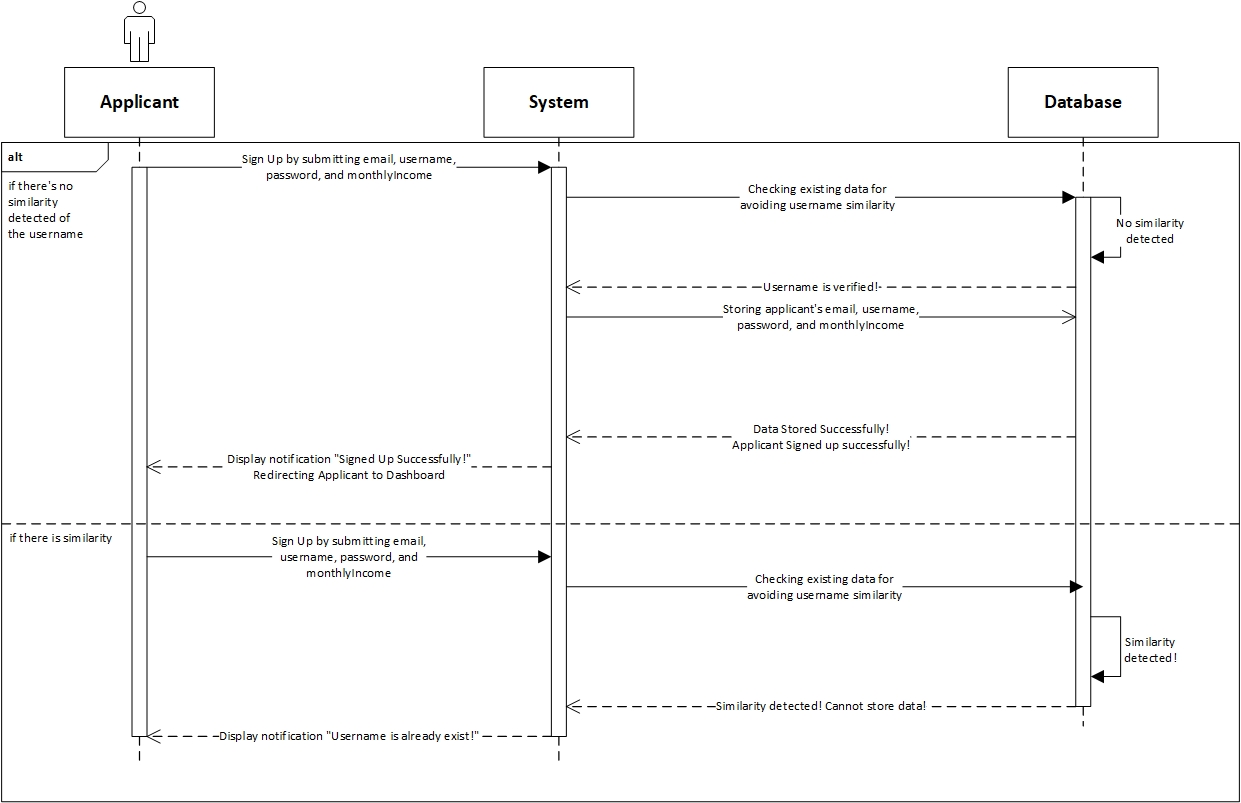


**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Allocate Housing** |
| Operation | Login with ID and username |
| Responsible | To access the main page |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |

|  |  |
| --- | --- |
| **Cross References** | **Allocate Housing** |
| Operation | Select view application list |
| Responsible | To get the list of all application |
| Pre-conditions | The application object must be available |
| Post-conditions | Success displayed list of all application with status |
| **Cross References** | **Allocate Housing** |
| Operation | Select one of the application and allocate the housing |
| Responsible | To allocate the housing for an application |
| Pre-conditions | Object residenceID, unit number, from date and duration must be available |
| Post-conditions | Success create allocation object based on data input |
| **Cross References** | **Allocate Housing** |
| Operation | Select to changed status of to be “Approved” |
| Responsible | To change the status to “Approved” |
| Pre-conditions | Application object must be available |
| Post-conditions | Success changed the application status |
| **Cross References** | **Allocate Housing** |
| Operation | Select to changed status of to be “Rejected” |
| Responsible | To change the status to “Rejected” |
| Pre-conditions | Application object must be available |
| Post-conditions | Success changed the application status and redirected to waiting list |

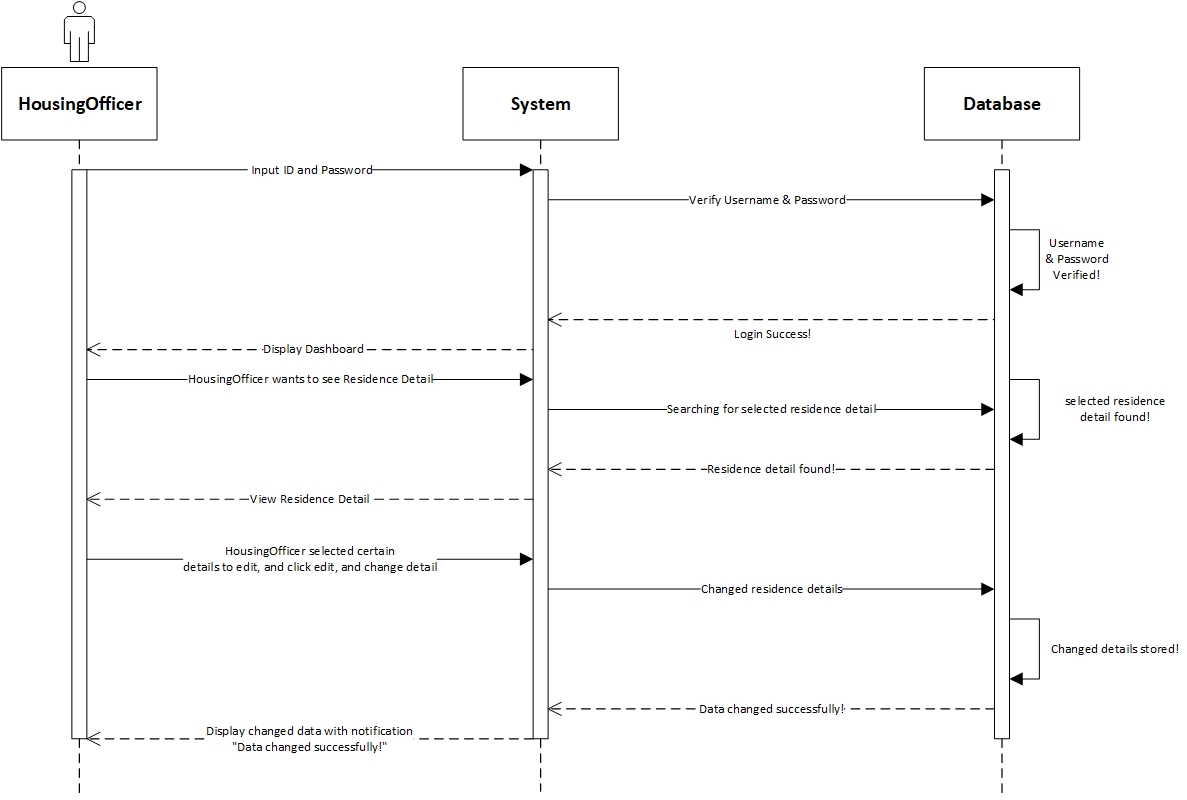
1. **Sign Up Applicant**



**Prepared by: Rivaldo Bagus Soepardhy**

|  |  |
| --- | --- |
| **Cross References** | **Sign up Applicant** |
| Operation | Sign up with email, username, password, and monthlyIncome |
| Responsible | To be registered in system and able to access the main page |
| Pre-conditions | * Email must be available * Username must be available * Password must be available * MonthlyIncome must be available |
| Post-conditions | * Username is verified (no similarity detected) * Registered in System * Display dashboard |
| **Cross References** | **Sign up Applicant** |
| Operation | Sign up with email, username, password, and monthlyIncome |
| Responsible | To be registered in system and able to access the main page |
| Pre-conditions | * Email must be available * Username must be available * Password must be available * MonthlyIncome must be available |
| Post-conditions | * Username is not verified (already existed in system) * Applicant must re-enter new username. |

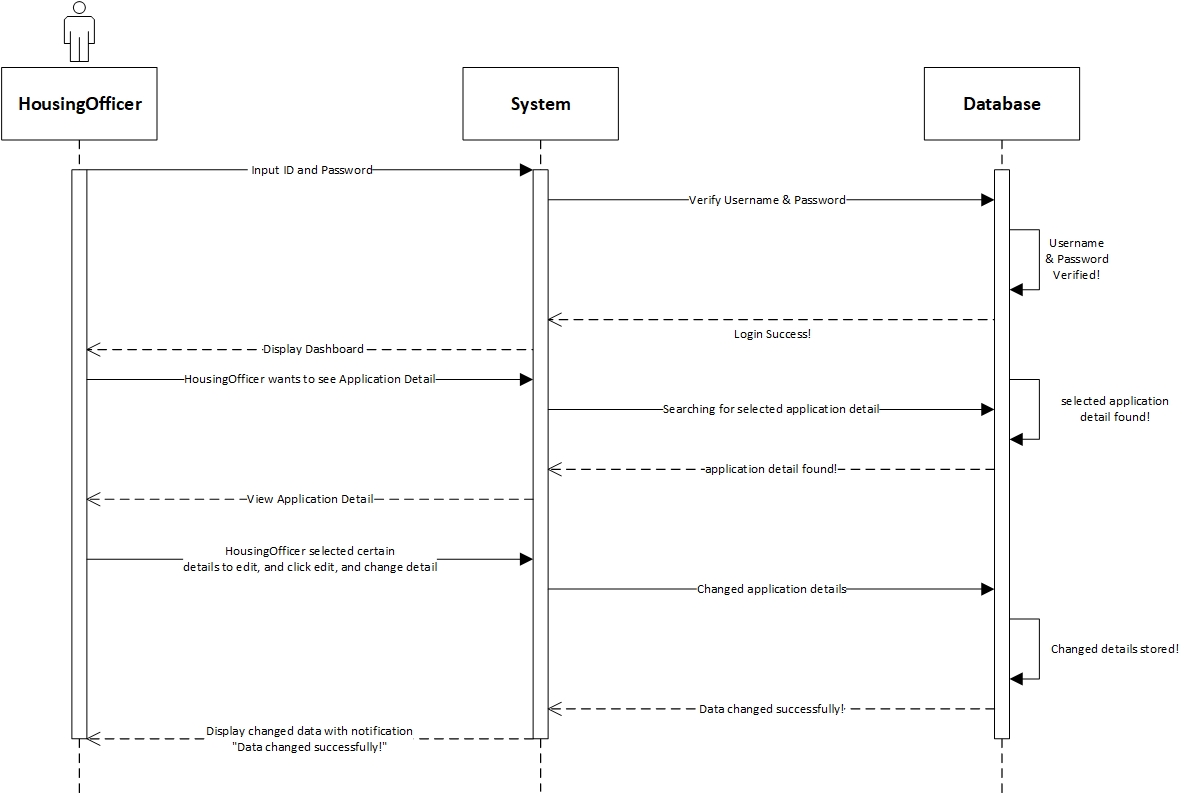
1. **Edit Residence Detail**



**Prepared by: Rivaldo Bagus Soepardhy**

|  |  |
| --- | --- |
| **Cross References** | **Edit Residence Detail** |
| Operation | Login with username and password |
| Responsible | To get access to view the applications |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **Edit Residence Detail** |
| Operation | Edit residenceID, address, numUnits, sizePerUnit, monthlyRental |
| Responsible | To change certain Residence Detail |
| Pre-conditions | We will need **one of** or **all of** the details below:   * Email must be available * Username must be available * Password must be available * MonthlyIncome must be available |
| Post-conditions | Residence Detail changed successfully |

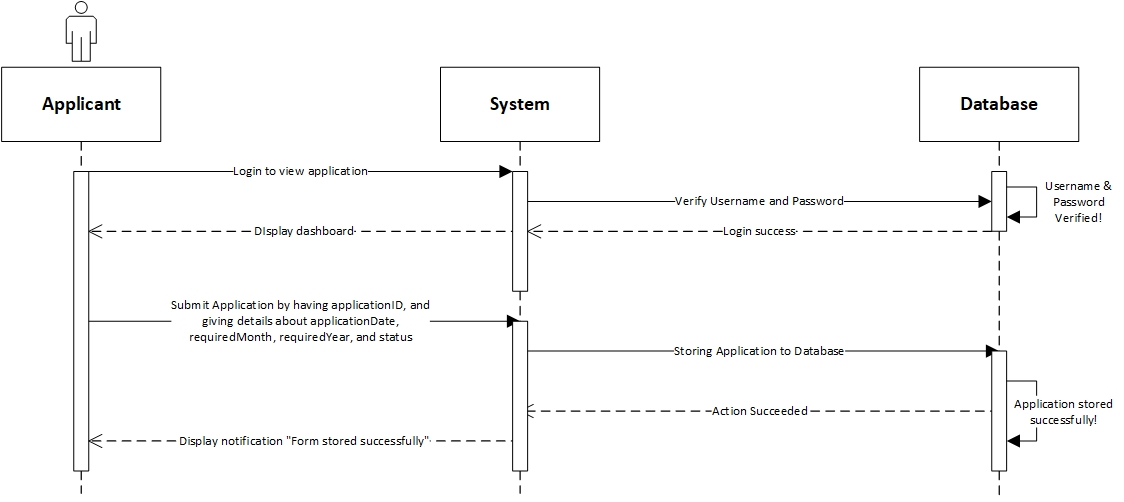
1. **Edit Applications**

****

**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Edit Applications** |
| Operation | Login with username and password |
| Responsible | To get access to view the applications |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **Edit Applications Detail** |
| Operation | Edit status, payment and payment status |
| Responsible | To change certain Applications Detail |
| Pre-conditions | We will need **one of** or **all of** the details below:   * applicantID must be available * residenceID must be available * requiredMonth must be available * requiredYear must be available |
| Post-conditions | Applications changed successfully |

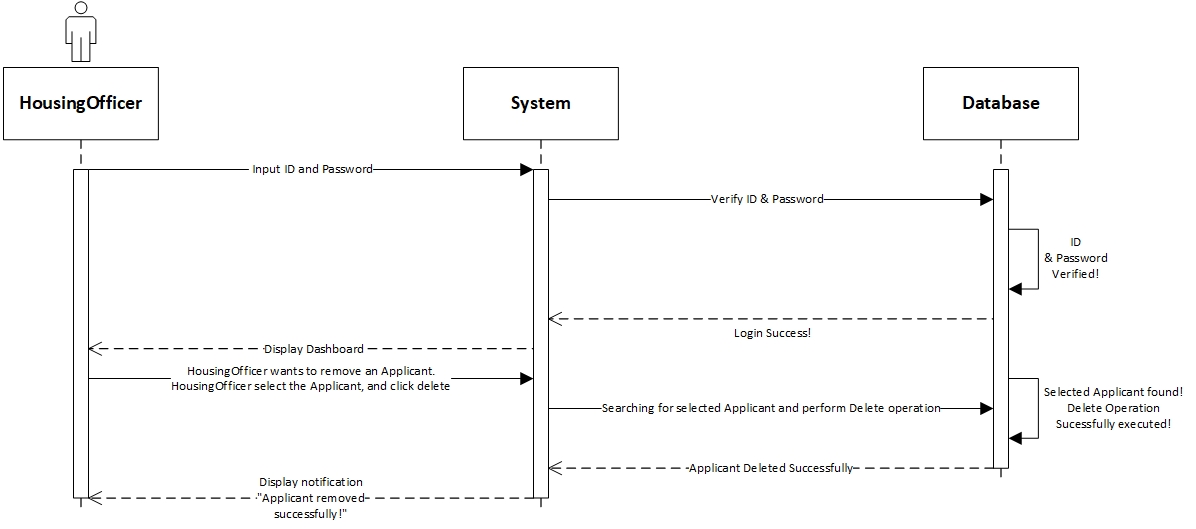
1. **Submit Application**



**Prepared by: Rivaldo Bagus Soepardhy**

|  |  |
| --- | --- |
| **Cross References** | **Submit Application** |
| Operation | Login with username and password |
| Responsible | To get access to view the applications |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **Submit Application** |
| Operation | View Residence & Select Residence |
| Responsible | To submit applicant’s application to the database. |
| Pre-conditions | * Applicant is already logged in to the system * Applicant has already determine the desired residence |
| Post-conditions | User will view the list of the residence, and then select the desired residence. |
| **Cross References** | **Submit Application** |
| Operation | Submit applicationID, applicationDate, requiredMonth, requiredYear, and status |
| Responsible | To submit applicant’s application to the database |
| Pre-conditions | Applicant has already selected the residence. We will need **all of** the details below:   * applicationID will be given by system automatically * applicationDate must be available * requiredMonth must be available * requiredYear must be available * status will be given by system automatically |
| Post-conditions | Application will be stored successfully, and will be on waiting list for the approval & arrangement by HousingOfficer. |

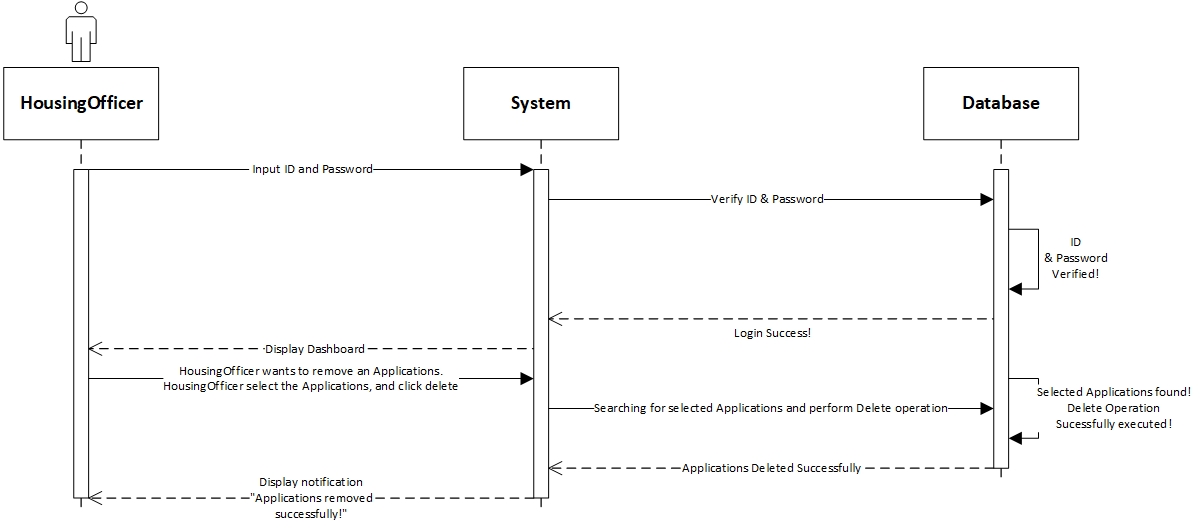
1. **Delete Applicant**



**Prepared by: Rivaldo Bagus Soepardhy**

|  |  |
| --- | --- |
| **Cross References** | **Delete Applicant** |
| Operation | Login with ID and username |
| Responsible | To access dashboard, and delete applicant |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **Delete Applicant** |
| Operation | Delete Applicant |
| Responsible | To remove an applicant |
| Pre-conditions | The applicant object must be available, and the applicant is no longer in place (checkout) |
| Post-conditions | Applicant object will be removed successfully |

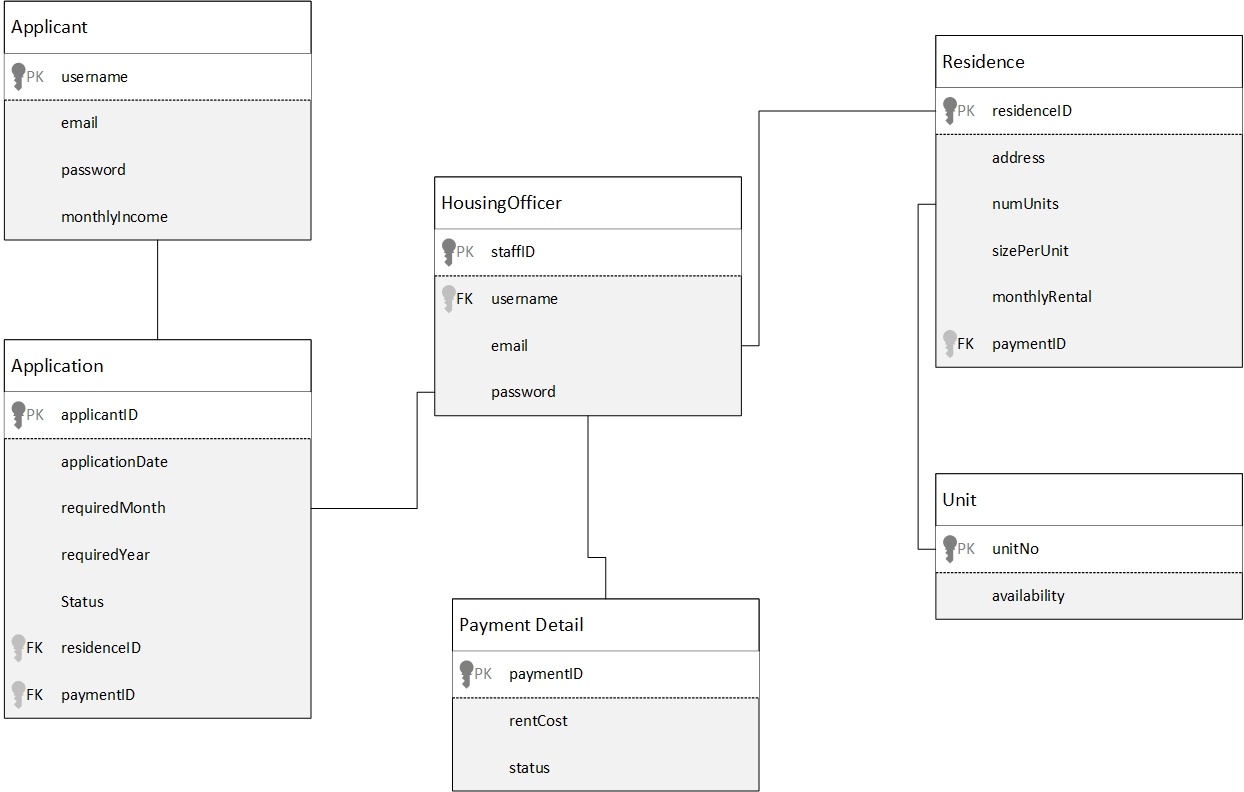
1. **Delete Applications**

****

**Prepared by: Luh Wulandari Maharani**

|  |  |
| --- | --- |
| **Cross References** | **Delete Applications** |
| Operation | Login with ID and username |
| Responsible | To access dashboard, and delete applicant |
| Pre-conditions | * Username must be available * Password must be available |
| Post-conditions | * Username is matched * Password must be match based on user’s password input * Display dashboard |
| **Cross References** | **Delete Applications** |
| Operation | Delete Applications |
| Responsible | To remove an applications |
| Pre-conditions | The applications object must be available, and the applications is no longer in place (checkout) |
| Post-conditions | Applications object will be removed successfully |

Database Design

****

Iteration 1

Use Case

|  |  |
| --- | --- |
| Use Case | Developed By |
| Login | Wulan |
| Sign Up | Wulan |
| Change Password | Wulan |
| Log Out | Wulan |
| View Application | Wulan |
| View Residence Details | Wulan |
| Set Up New Residence | Wulan |
| Edit Residence Detail | Wulan |
| Submit Application | Aldo |
| View Applications | Aldo |
| View Residences | Aldo |

Test Objectives

System developers are human that are not free from mistakes. Errors can present in the codes and design during development phase. Thus, the main purpose of testing is to find defects that may get created while developing the software. Testing is important to build confidence in the software quality before handing out the software to the clients. It is an essential stage to make sure requirements are met and all functions are performing well, in order to maintain clients’ reliability and satisfaction. Testing is also important to prevent any faults occur in the future, because it will be harder and more expensive to fix.

For our school information system, our first test objective is to make sure login validation works and password data is successfully encrypted with md5 encryption. Next, we want to test if the application (web pages) have the same view across different browsers. From the admin perspective, he/she should be able to add, edit and delete data in the system such as student, teacher, etc. Meanwhile, the student should be able to view data only related to that student, and able to send message to admin. We also have to test if all the links are working properly and directing to the right pages.

Tempel tempel pang ada template

Test Plan

Masih rancu, nanti tak copas dulu

Unit Testing

Register Applicant

This test will test the Applicant in creating a new account that will be used to login to the system. After logging in the Applicant will immediately be directed to the Applicant's home menu.

2. Login Applicant

In this test will show the Applicant in logging into the system by entering the username and password that was made before. 21

3. SAS Admin

This test will test the SAS Admin home menu which has the function to direct the page to Maintain Qualification and University Register.

4. Maintain Qualification

Pada testing ini akan mengetes fungsi add qualification yang digunakan untuk menambahkan qualification

The data input on the add qualification was successfully added to the database and the data contained in the table was in accordance with the database. 22

5. University Admin

In this test will test the University Admin home menu which has a function to direct the page to the Record Programme and Review Applications.

6. Record Programme

In this test, we will test the record programme function which is used to add the program to university.

The data input in the program record was successfully added to the database and the data contained in the table was in accordance with the database.

7. Applicant

In this test, we will test the Applicant home menu, which has the function to direct the page to Apply for Program.

fdsafdsa

Integration Testing

n this integration testing, the testing will use the Selenium-WebDriver application. The system will run based on the functions that exist in each use case. The results of testing will be exported into JavaScript files.

1. Register Applicant

2. Login

3. Maintain Qualification

25

4. Record Programme

NANTI TESTNYA PAKAI SELENIUM

System Testing

Functional Requirements

At this stage, the system will carry out testing in accordance with the functional requirements that have been determined previously. The following shows the results of testing that has been done.

a. Register Applicant

b. Login Applicant

c. Maintain Qualification

27 d. Record Programme

2. Non-functional

At this stage, the system will carry out testing according to the non-functional requirements that have been determined previously. The following shows the results of testing that has been done.

a. Usability

The system already have a simple interface design and easy to use also easy to understand. 28

b. Accuracy and Precision

The data entered by the user has been appropriate, accurate, and precise and there are no errors. And also, the data is in accordance with the data contained in the database.

Dsadsa

Test Analysis Report

From a series of tests we conducted regarding unit testing, integration testing, and system testing, it can be analyzed that the system developed is functioning properly because it has met the desired requirements. In this testing phase, the use cases tested are Login, Register, Maintain Qualifications, and Record Program.

In unit testing for use cases Maintain Qualification is tested on the Add Qualification and Display List Qualification functions. In the unit testing for use case the Record Program is tested on the Add Program and Display List Program functions. For the Integration Testing stage, all functions in each use case are tested using a predetermined application, Selenium-WebDriver. And, the System Testing is tested by functional and non-functional matching. Some functional tests that have been tested are:

 Register Applicant :

This page is only intended for Applicants, for creating new accounts.

 Login :

This page functions to enter the system, but you must register beforehand if you do not have an account.

 Maintain Qualification :

This page functions to add and view a list of qualifications that have been determined.

 Record Programme :

This page functions is to add and view a list of programs that have been determined.

In Non-functional testing, for Usability the system already has a simple interface design and easy to use also easy to understand and for the Accuracy and Precision the data entered by the user has been appropriate, accurate, and precise and there are no errors. And also, the data is in accordance with the data contained in the database.

Updated Gantt Chart, Trello Board, GitHub

