ĐẠI HỌC CÔNG NGHỆ TPHCM

Khoa Công nghệ thông tin – Viện Đào tạo Quốc tế

🙠🕮🙢



Project: Smart Fire System

Trần Nguyễn Quốc Khang(ID: 1711062481)

Hồ Lê Nhật Linh(ID: 1711060465 )

Lê Văn Truyền (ID: 1711060705)

Nguyễn Hoàng Kỳ (ID: 1711061097)

Class : 17DTHQA1

Table of Contents

[I. Introduction 4](#_Toc29201587)

[A. Overview 4](#_Toc29201588)

[B. Purpose 4](#_Toc29201589)

[C. Product Scope 4](#_Toc29201590)

[II. Overall Description 4](#_Toc29201591)

[A. Product perspective 4](#_Toc29201592)

[B. Product functions 4](#_Toc29201593)

[C. User classes and characteristics 5](#_Toc29201594)

[D. Operating environment 5](#_Toc29201595)

[III. Question to the fire system 5](#_Toc29201596)

[IV. Feasibility 7](#_Toc29201597)

[V. Database for smart fire system 10](#_Toc29201598)

[VI. Smart Fire system architecture 11](#_Toc29201599)

[VII. Requirements 11](#_Toc29201600)

[E. Product requirements 11](#_Toc29201601)

[F. Organizational requirements 12](#_Toc29201602)

[G. External requirements 12](#_Toc29201603)

[VIII. Diagram 13](#_Toc29201604)

[A. Class diagram 13](#_Toc29201605)

[B. Use-case diagram 14](#_Toc29201606)

[C. Sequence diagram 15](#_Toc29201607)

[IX. Features of the program 16](#_Toc29201608)

[A. Login/Sign up screen 16](#_Toc29201609)

[B. Manager screen 18](#_Toc29201610)

[C. User screen 20](#_Toc29201611)

[D. Residents list 21](#_Toc29201612)

[E. Resident state: 24](#_Toc29201613)

[F. Accounts list 25](#_Toc29201614)

[G. Update status 27](#_Toc29201615)

[H. Camera 30](#_Toc29201616)

[I. User profile 32](#_Toc29201617)

[J. Alarm 34](#_Toc29201618)

[K. Exit route 36](#_Toc29201619)

[X. Other requirements and features: 37](#_Toc29201620)

[A. Offline mode 37](#_Toc29201621)

[B. Friend function 37](#_Toc29201622)

[C. Power out 37](#_Toc29201623)

[D. Non functional-requirements 37](#_Toc29201624)

[XI. Demo program: 38](#_Toc29201625)

[A. Installing database 38](#_Toc29201626)

[B. Fire system program 45](#_Toc29201627)

# Introduction

## Overview

Smart Fire System is a system which is used to detect fire early in high buildings, it can notify the users when there is a fire happen, it can also keep track of residents’ location inside the fire as well as exit route. It keeps the information of residents and update residents’ state constantly so the manager can decide a proper action.

## Purpose

* Detect and warn the fire early.
* Provide the status of the residents in the fire.
* Lessen the casualty caused by the fire.

## Product Scope

It can be used in high buildings, hotels, apartments, it can even be used in a wide residential area, where fire needs to be detected as soon as possible to prevent loss.

# Overall Description

## Product perspective

Smart Escape is a system for early detection and warning of fire in high buildings, and provides escape solutions, managing the residents’ condition to respond in time in case of fire.

## Product functions

On the management side:

* Receive alerts on fire locations on installed cameras.
* Send notifications to all users within the area of a fire.
* Monitoring and statistics the status and number of people in the area where the fire occurs.

On the user side:

* Receive fire alerts from the manager.
* Update user’s own safety status as required when a fire occurs.
* Check for safe escape route.
* See current status of relatives when there is a fire warning.
* Provide photos, update fire situation for the manager.

## User classes and characteristics

User class:

* System management and monitoring department.
* Normal users.

## Operating environment

* Surveillance cameras.
* Android, iOS, Linux and Windows devices.

# Question to the fire system

1. Question 1

In SRS version 1.0, page 9, the requirement to use the fire alarm feature is the use must log in into the system, I personally recommend adding a log in screen with a sign-up button.

1. Question 2

In SRS version 1.0, page 10, the update user safe state only available in the customer app, I suggest adding this feature to the manager app as well.

1. Question 3

In SRS version 1.0, page 7, only the manager can raise the alarm , I personally recommend an alarm button in the user app in case the camera cannot detect fire or malfunction , when the user press this button, the manager will receive the notification and check that place, then the manager can confirm whether the alarm is fake or real.

1. Question 4

In SRS version 1.0, regarding the updating the user password, I recommend checking if the password has more than 6 characters and must contains a capital letter and a number.

1. Question 5

In SRS version 1.0, regarding the state of the user, if the user doesn’t log in, then in a fire alarm, how will the system display user state? I suggest the following 2 options:

* Option 1: State will be “Unknown”.
* Option 2: State will be “SOS”.

I hope you can help with this problem.

1. Question 6

In SRS version 1.0, regarding the sign-up screen, in case the user enters a weak password, how to display the message, I recommend following 2 options:

* Option 1: Display a red message above the sign-up username which read : “Password is too weak”.
* Option 2: Display a message box that says: “Password is too weak”.

Please confirm which option you will choose.

1. Question 7

In SRS version 1.0, page 10, when the user :

1. Choose the floor.
2. Choose the block.
3. Choose the “SOS” button.

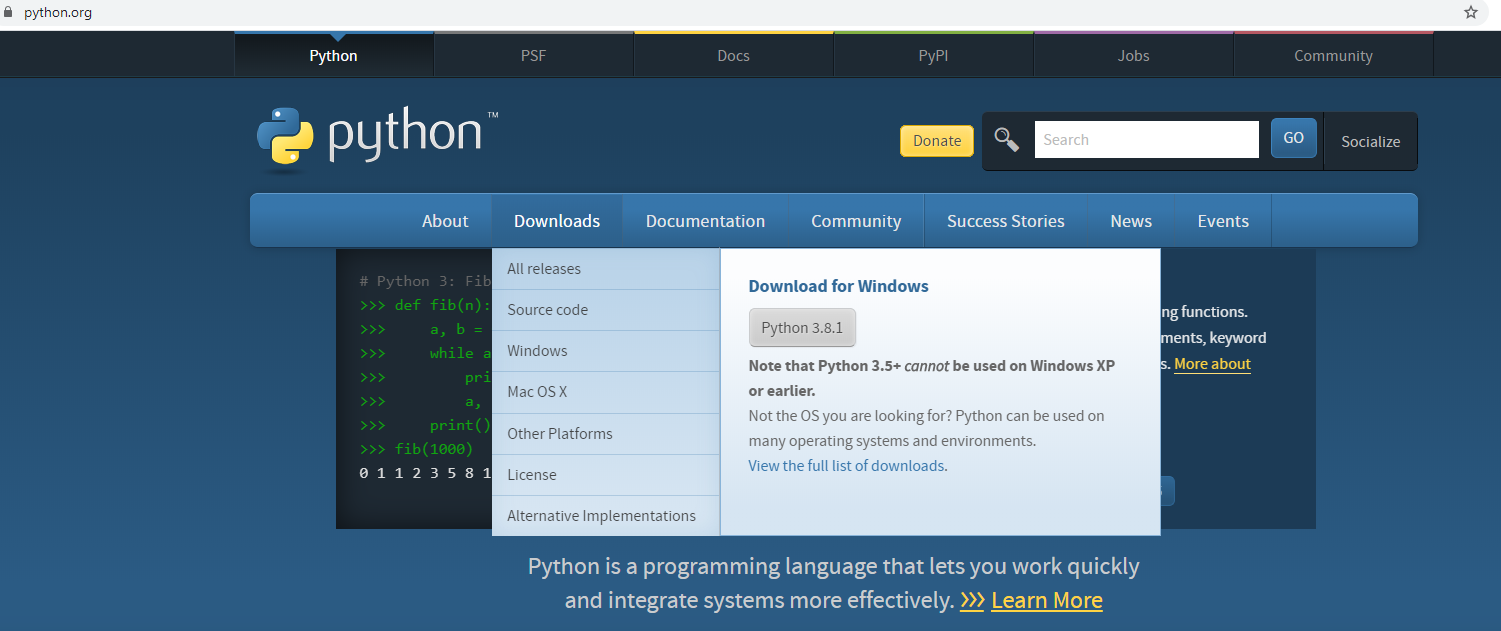
Then in the manager screen, the list of residents will refresh, the state “SOS” will be updated, along with the floor and block numbers, am I understand correctly?

# Feasibility

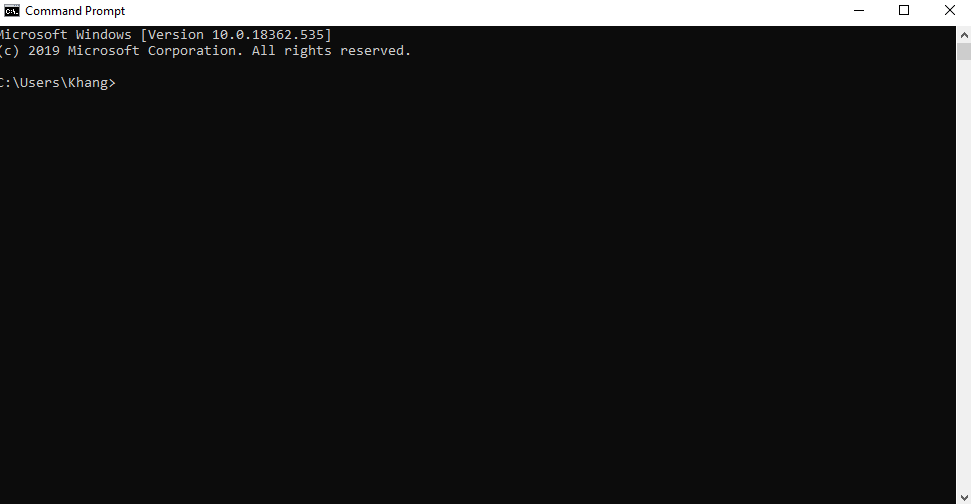
To start building a fire system detection, we need to check if we are capable of building a system like that, the most important thing is we must check if the camera can detect fire or not.

An AI system that able to detect fire has been built on GitHub using python, the link is <https://github.com/tobybreckon/fire-detection-cnn>

### Install python:



### Open cmd and type the following commands to install components:



* pip install tflearn
* pip install tensorflow==1.15rc2
* pip install opencv-python

### Download the pre trained network <http://dx.doi.org/10.15128/r19880vq98m>

### Extract the model to the models(create it) in the folder fire-detection-cnn-master

### Go in to cmd and type python firenet.py models/test.mp4



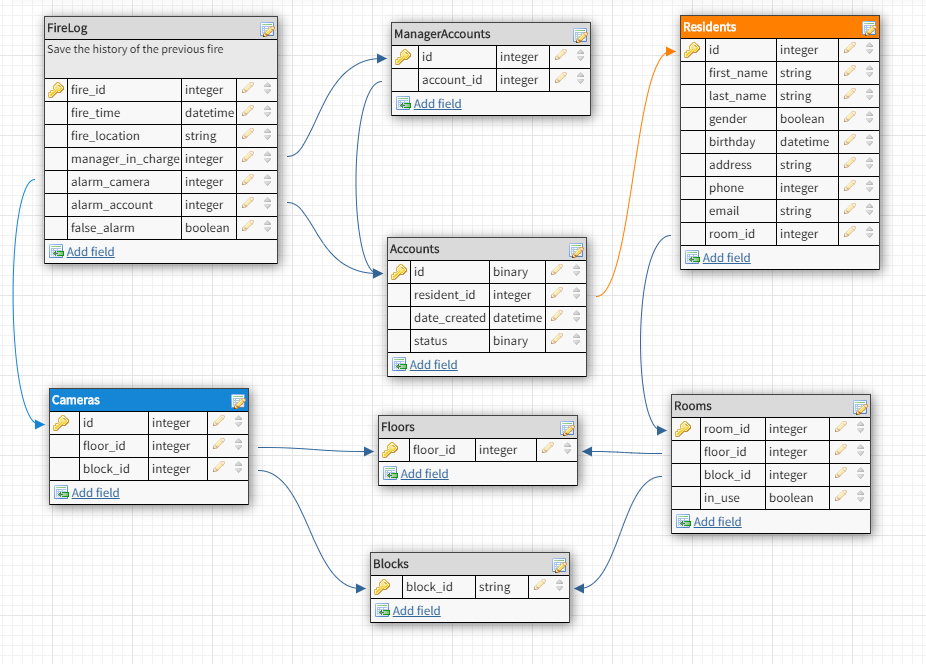
### The AI will announce “FIRE” when there is fire in the video and “CLEAR” when it is safe:



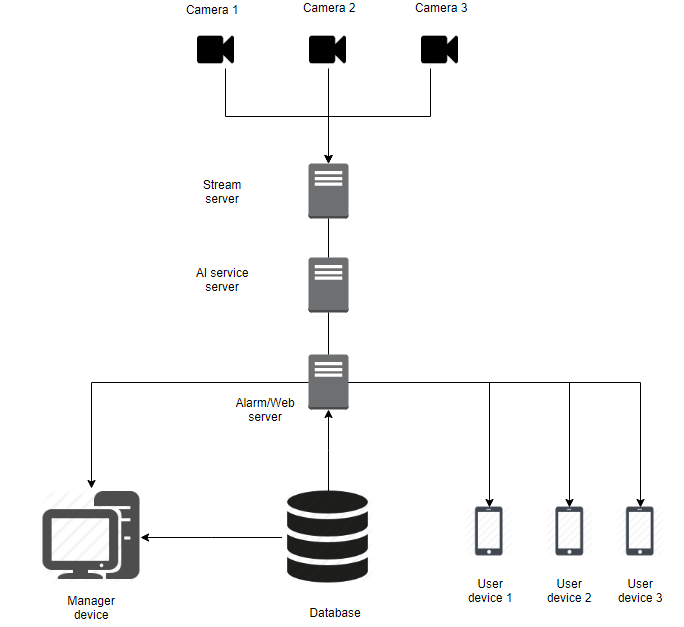


Conclusion : it is totally possible to make a fire system where the camera will detect fire

# Database for smart fire system



# Smart Fire system architecture



# Requirements

## Product requirements

### Usability requirements:

* The product can be used by anyone:

### Efficiency requirements:

#### Performance requirements:

* The product has the respond time is 5 second.
* The product will have the connection timed out time is 10 second.

#### Space requirements:

* The product takes the maximum 200MB of hard drive space.
* The product takes the maximum 100MB of RAM.

### Dependability requirements:

* The product is expected to have the false-positive(false alarm) rate of 0.01%

### Security requirements:

* The system must protect the private information of customer and avoid information leaking

## Organizational requirements

### Environmental requirements:

* The product can work on IOS, Android and simulator on Windows and Linux.

### Operational requirements:

* The system is used to detect fire alarm and inform residents state.
* The system can handle more than 1000 users simultaneously.

### Development requirements:

* The system will be developed using c#, python and java.

## External requirements

### Ethical requirements:

* The system must not disclose any of the resident information.
* The system must choose an escape route to save as many people as possible.

### Regulatory requirements:

* The system must be tested many times in many situations before used.
* The system must meet the privacy, safety and security requirements before released.

### Legislative requirements:

#### Accounting requirements:

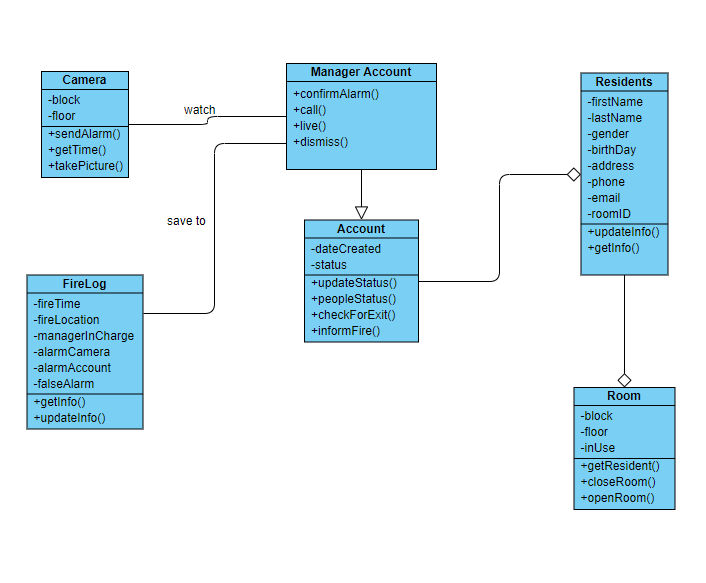
* The system must ensure to work with little to no error, and to raise the alarm as quickly as possible when there is fire.
* The manager of the system will be the first person responsible for any fire incident.

#### Safety/Security requirements:

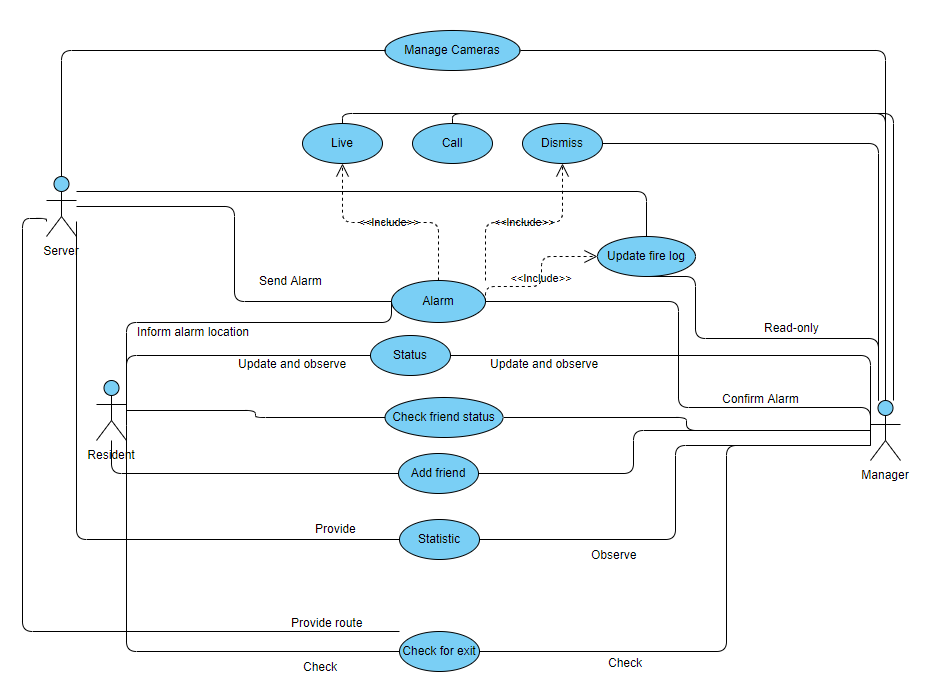
* The system must keep the user information entirely secret.
* The manager of the system only has limited rights and cannot access the user information.

# Diagram

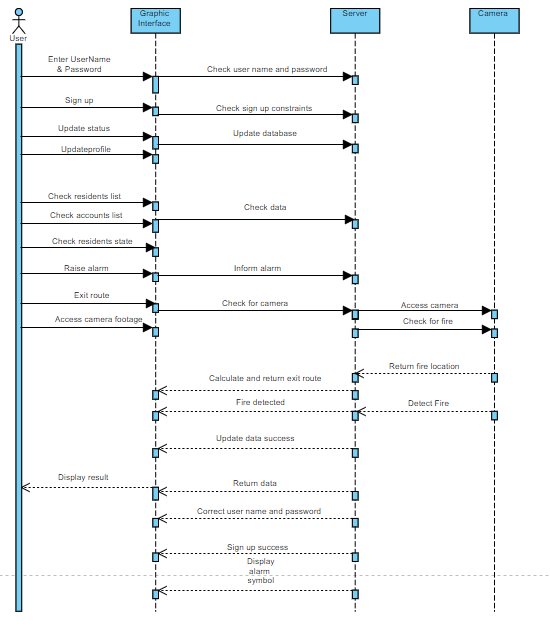
## Class diagram



## Use-case diagram



## Sequence diagram



# Features of the program

## Login/Sign up screen

### Description:

#### For manager:

When the manager enters user name and password, the manager screen will appear, which will let the manager manage resident’s information and state.

#### For user:

When the user enters the correct username and password, the user screen will appear, which will let the resident update the profile and update his/her safety status. If the user doesn’t have an account, user can sign up for a new account.

### Feature details:

#### For both user and manager:

When user and manager open the app, a login screen will appear, user and manager must enter the correct username and password.

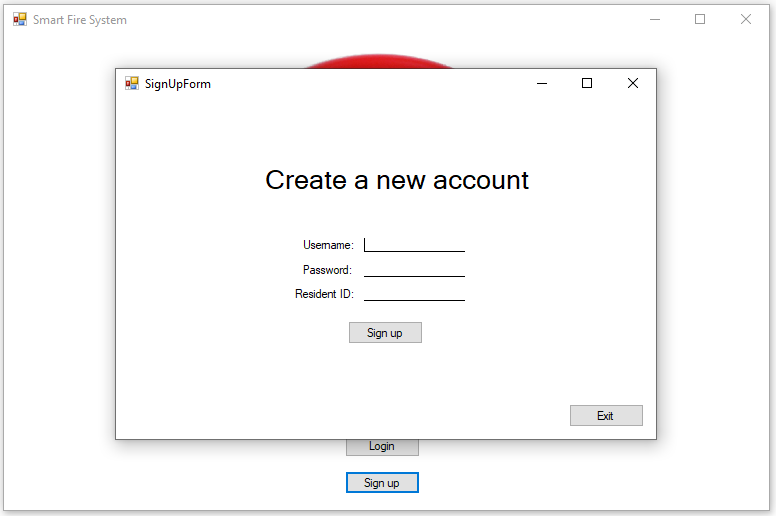


If enter the wrong username or password, the program will play a red message that says: “Wrong user name or password”.



#### For user only:

* User can press the sign-up button to sign up.
* One resident ID can only have one account.
* Username cannot be the same.
* If the account is being used by someone not inside the apartment, the manager can ban or delete the account.



### Condition to use:

* Requirement 1: The device which the user is using must be in online mode.

## Manager screen

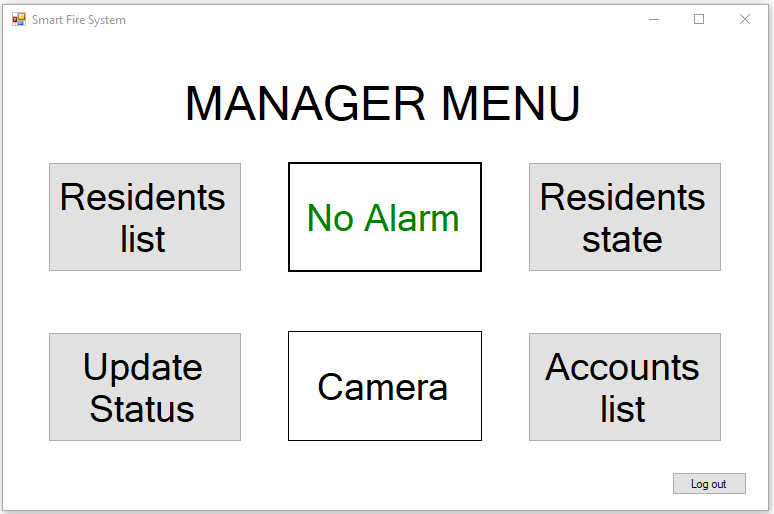
### Description:

This feature is only available for manager, when the manager logs in , the manager will be directed to a manager menu interface which will inform him/her about the building, resident state and raise the alarm if necessary.

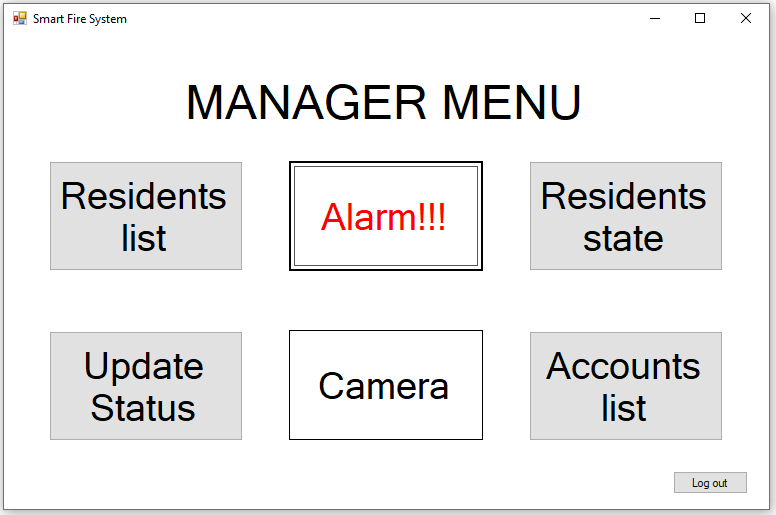
### Feature detail:

1. When the manager logs in, they will be presented with a manager menu screen, there are seven functions that the manager can use:

* **Residents list**: Observe the residents list.
* **Residents state**: Check the resident state.
* **Account list**: Look at the accounts list.
* **Update status**: Update the manager safety status.
* **Camera**: Check the cameras to see if there is any fire, or anyone looks suspicious.
* **Alarm** : Raise the alarm if there is a fire in the area, which will inform all the resident who use the app.
* **Log out**: return to the login screen.



1. When the manager presses the “No Alarm” button, the system will raise the alarm, and the “Alarm!!!” notification will pop out, replace the “No Alarm” message.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

## User screen

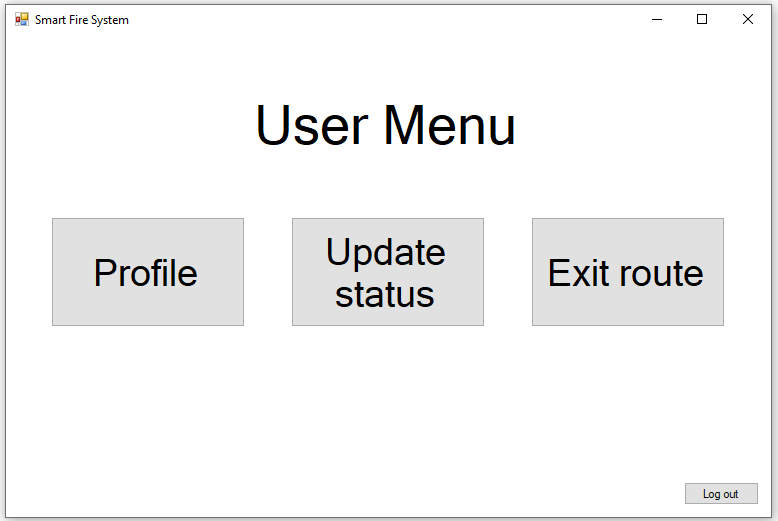
### Description:

This feature is available for user only, the user can see and change the user’s own profile, update safety status and check for the exit route.

### Feature details:

1. Log in as user.
2. The User menu screen will appear, which has three operations the user can choose to do:

* **Profile** : Check and update profile.
* **Update** **status**: Update current state in a fire alarm.
* **Exit route** : Check for safe exit.
* **Log out**: return to login screen.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

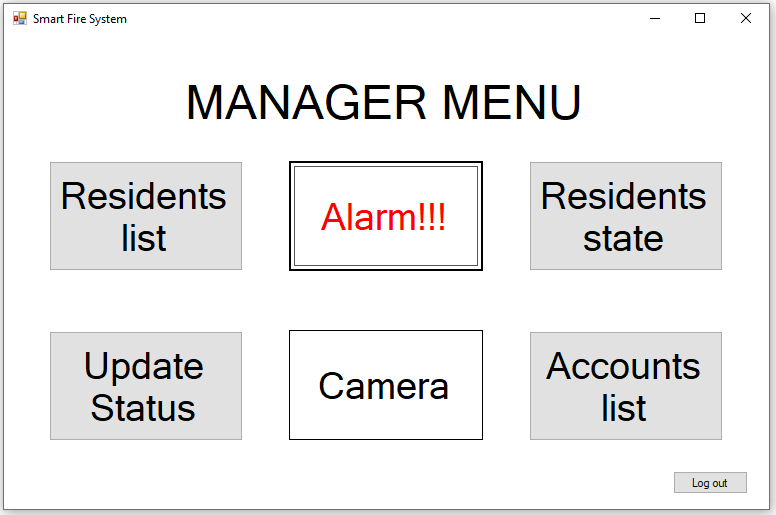
## Residents list

### Description:

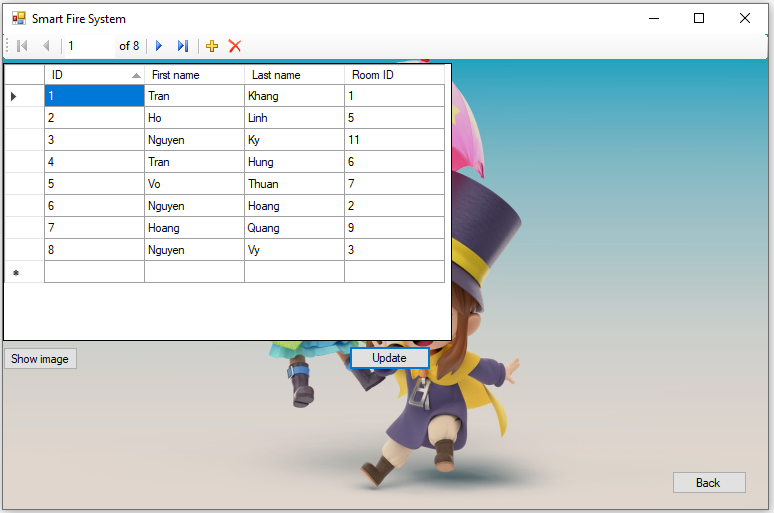
This feature is only available for manager, the manager can see the residents’ info, add more residents into the building database, delete resident or check their identity card( if available).

### Feature details:

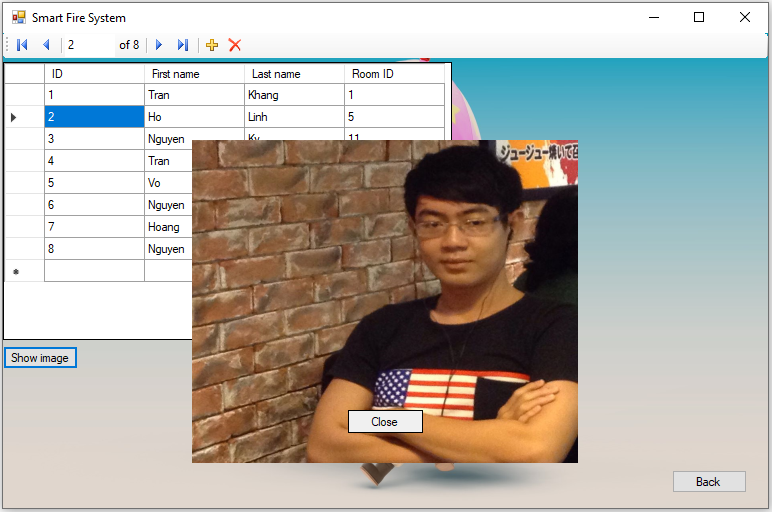
1. Choose the residents list button:



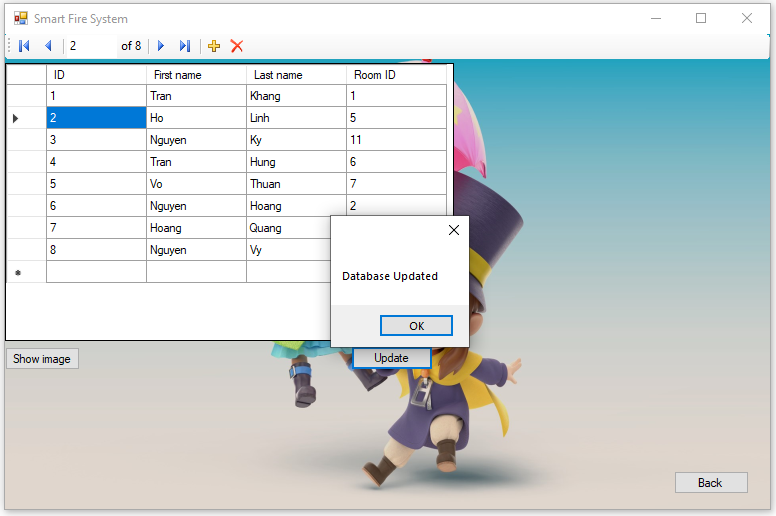
1. The system will show a grid view where the manager can see necessary the information about the residents ( some information mustn’t be shown for privacy purpose).



1. The manager can press “Show image” to show the profile picture of the resident.



1. The manager can update the information of residents, then press the “Update” button, the message box which says: “Database Updated” will appear if update successfully.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

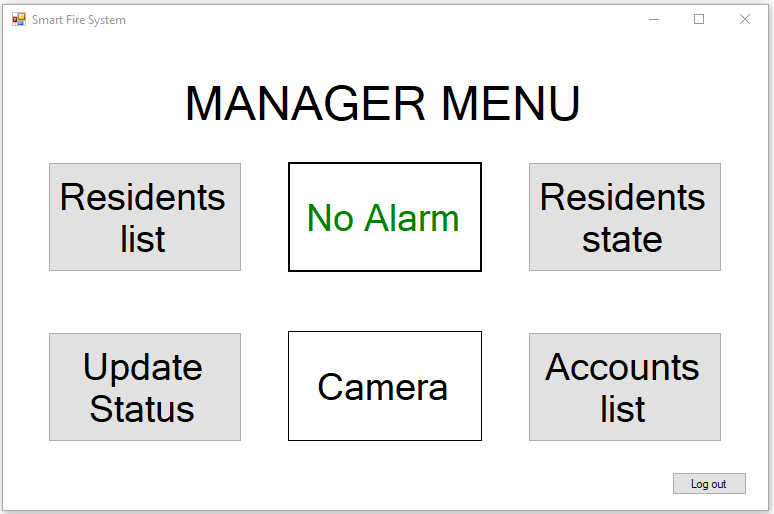
## Resident state:

### Description:

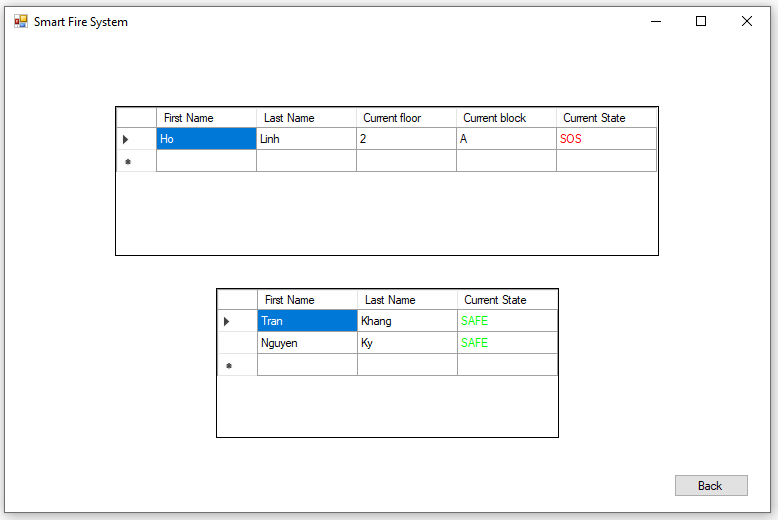
This feature is for manager only, it let the manager have a clear view of the current state of the residents, whether they are safe or not, and if not, then display their location.

### Feature details:

The manager needs to press the “Residents State” button in the manager menu:



Two grid boards will then appear, showing who is safe ( message “SAFE” with green color) and who is in danger (“SOS” message with red color) along with the location of the endangered residents.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

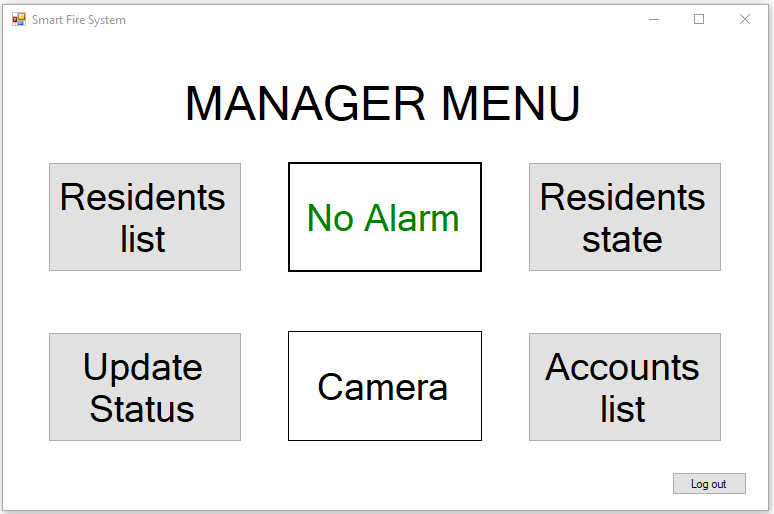
## Accounts list

### Description:

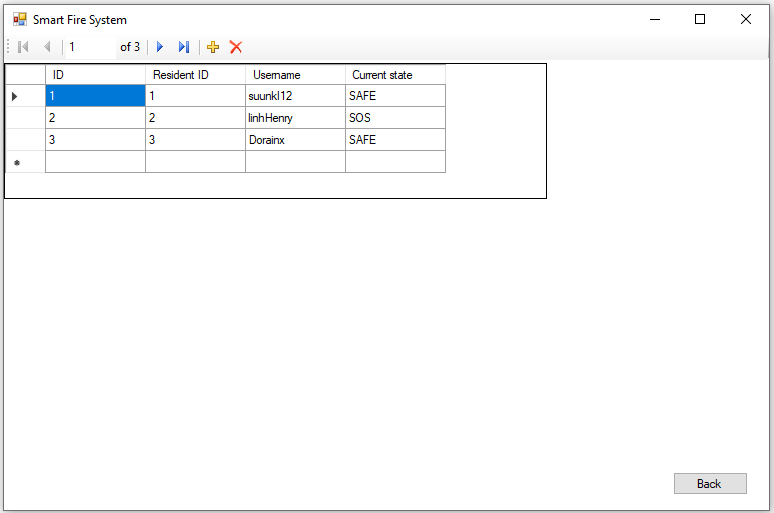
This feature is for manager only, which will let the manager view all the accounts in the database, a resident may or may not has an account, hence the number of accounts will be equal or less than the number of residents.

### Feature details:

Press the “Accounts list” button:



The “Account” screen will appear, the manager can see the username, but not the password, however, they can delete the account.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

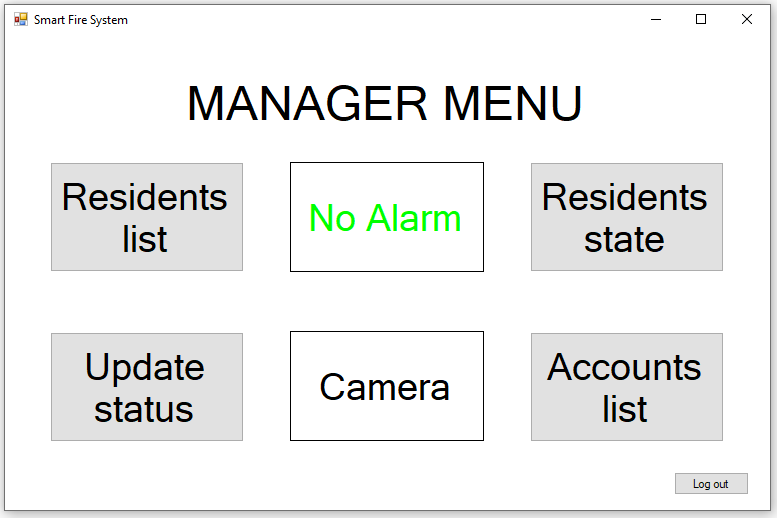
## Update status

### Description

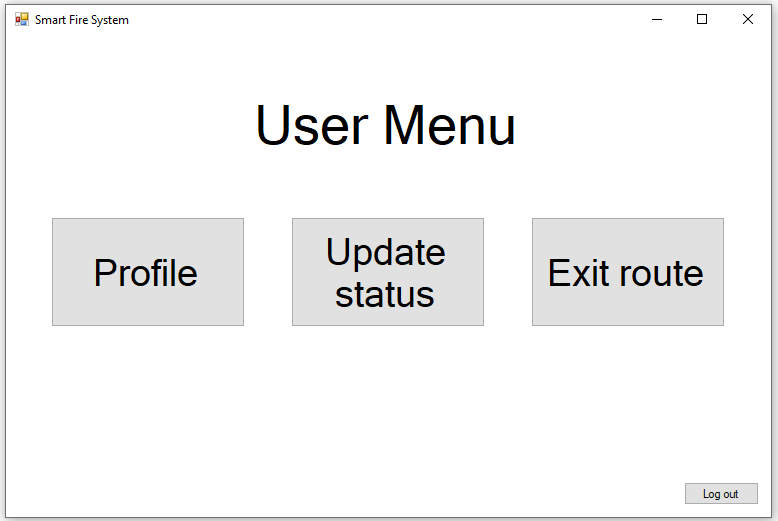
This feature is available for both user and manager, they can inform other people whether they are safe or in danger, as well as where they are.

### Feature details:

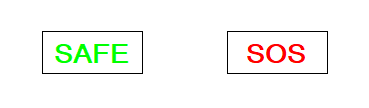
1. Choose “Update status button”
   1. For manager, the button is on the bottom right.



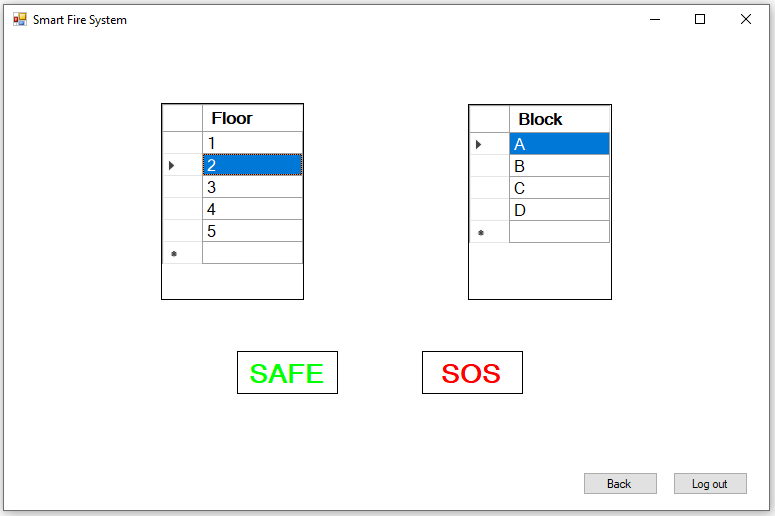
* 1. For user, choose the “Update status” button on the top left.



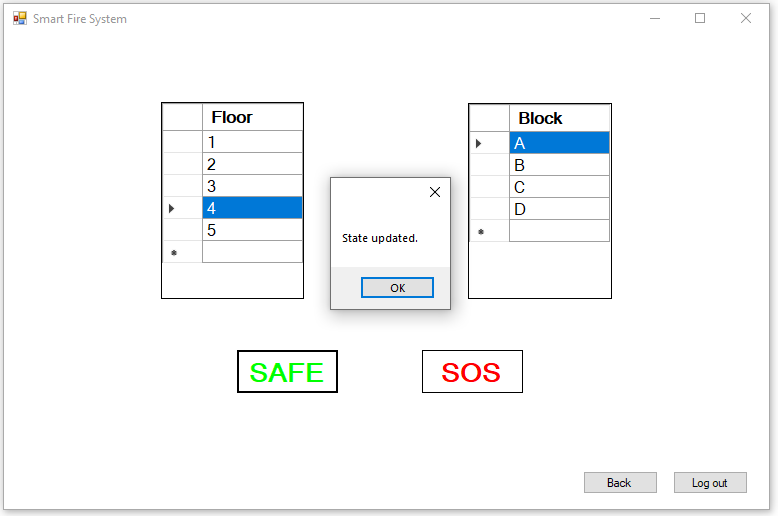
1. If not in danger, choose the “SAFE” button:



1. If in danger, choose the “Floor” and “Block” the user is currently in, they choose “SOS”:



1. The system notification will appear if safety status is updated successfully.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

## Camera

### Description:

*For manager:*

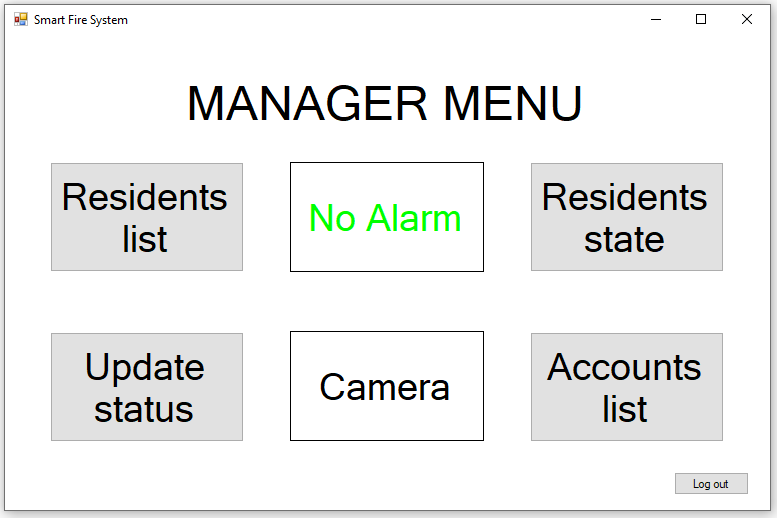
Let the manager keep watch for any fire or criminal in the building, the manager can see all the cameras, or choose any camera for close view.

*For user:*

Whenever a fire happens, a notification will pop up for the user along with the footage of the fire scene.

### Feature details:

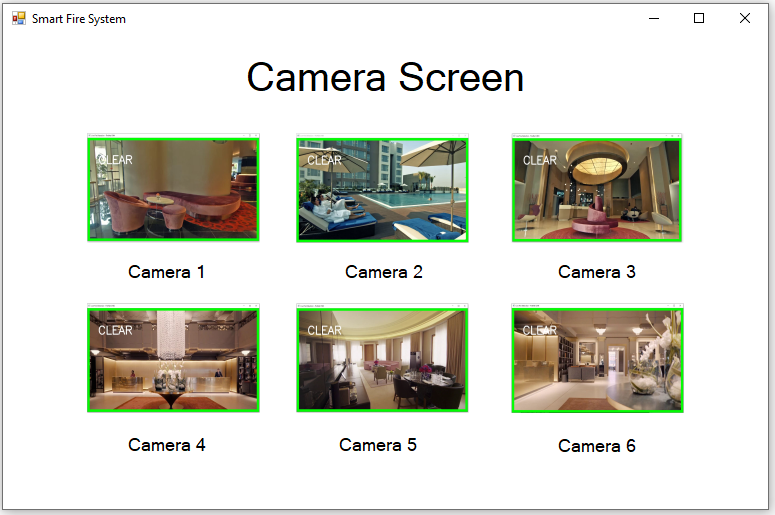
1. Accessing:
   1. Manager choose the camera button:



* 1. When there is a fire notification, user will see the camera:

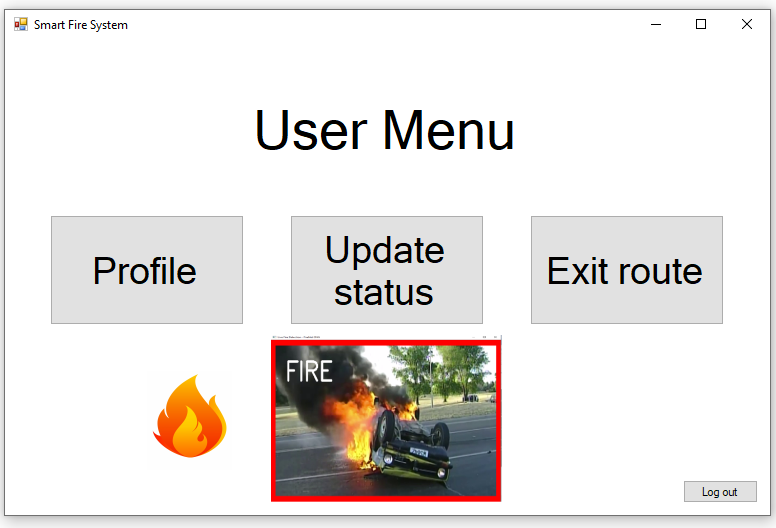
1. Camera screen:
   1. Manager:

Manager will see all the camera in the apartment:



* 1. User:

When there is a fire, a fire notification will pop up to inform the user, along with the camera at the fire location.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

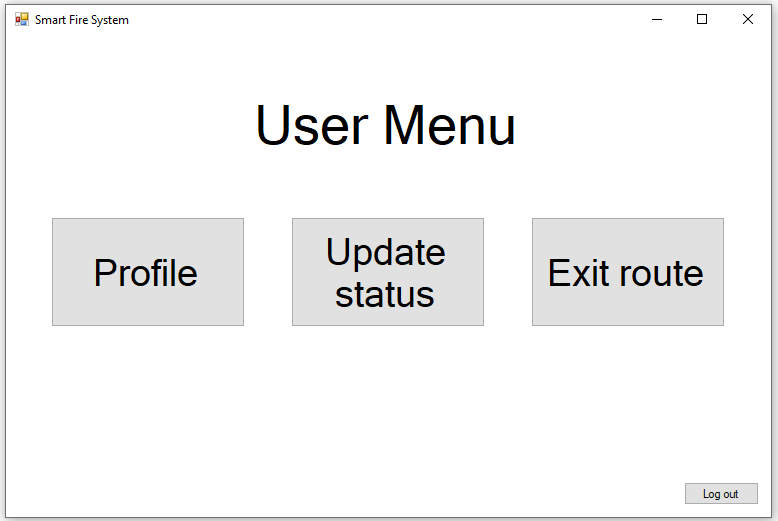
## User profile

### Description:

Only the user can use this feature in order to get the user’s information, user can change some information, id cannot be change.

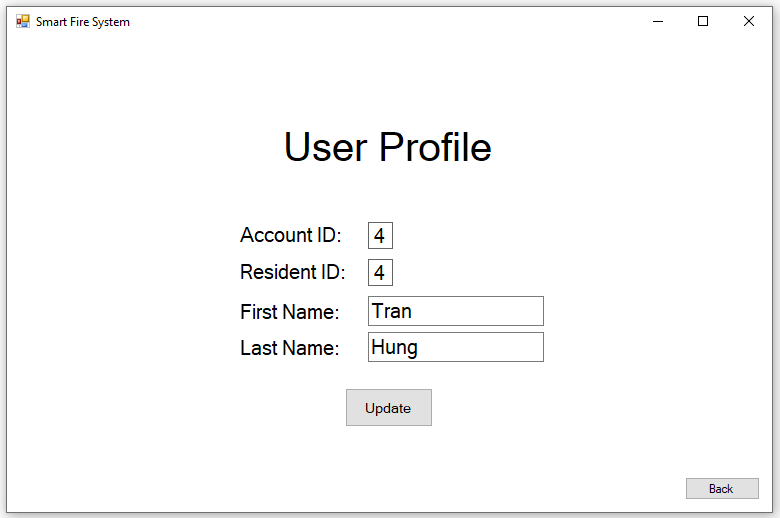
### Feature details:

1. User choose the “Profile” button:



1. The User profile window will pop up, where user can see his/her information, and user can update some of those information.

User cannot adjust Resident id or Account id.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

## Alarm

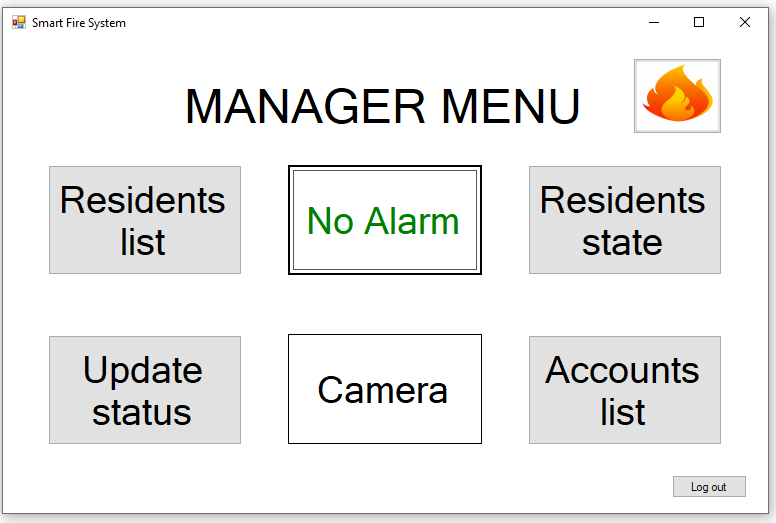
### Description:

This function is used to notify the manager when the camera detects a fire.

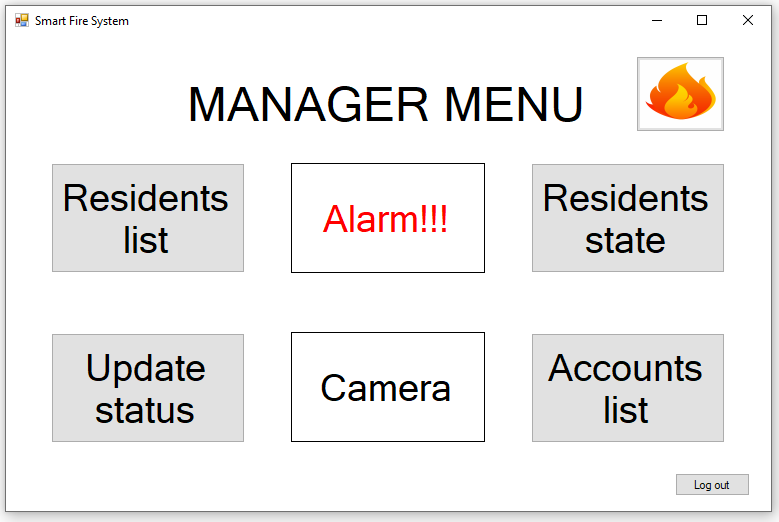
The manager can click on the alarm button to inform the residents.

### Feature details:

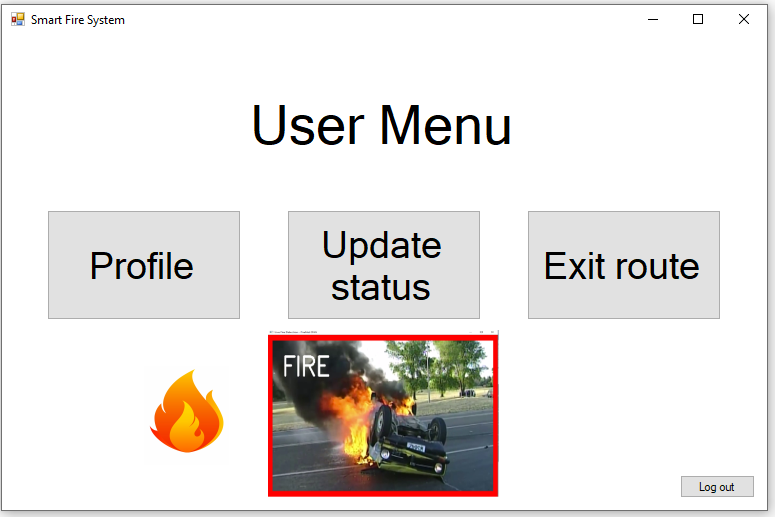
1. When there is a fire, a small fire icon will appear on the top right of the program, the manager can click on the icon to go to camera screen.



1. The manager clicks the alarm button:



1. The resident will see the alarm notification along with the camera at the fire scene.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

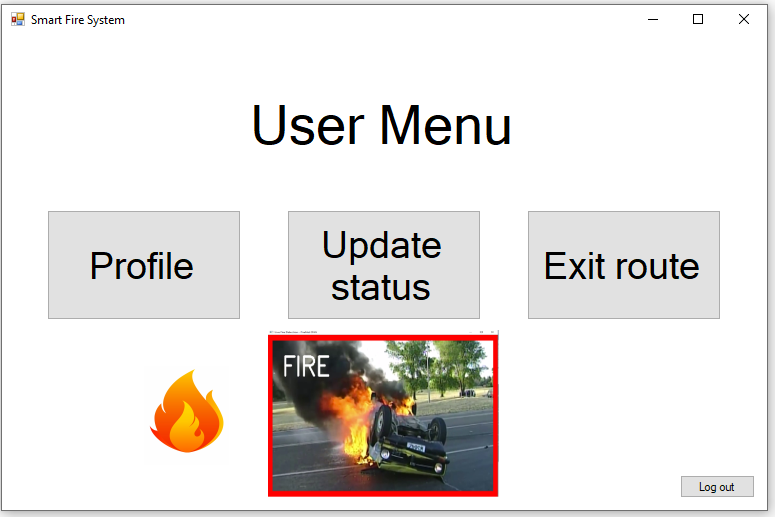
## Exit route

### Description:

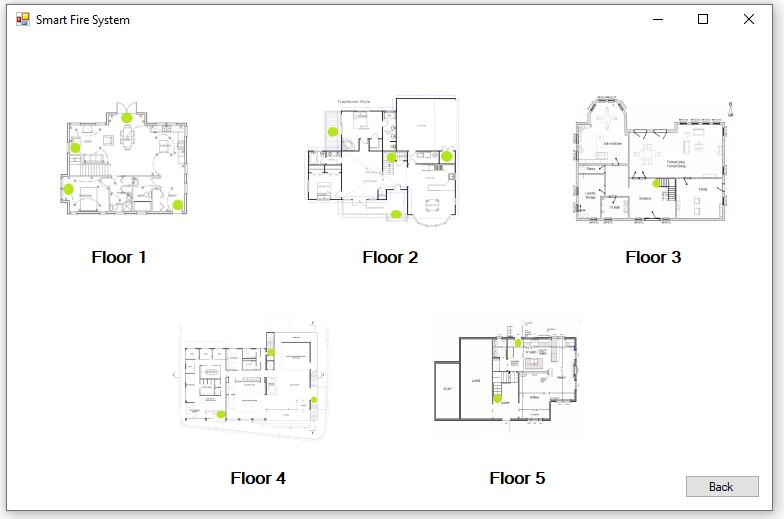
For user only, used to find the safest and fastest exit in a fire, the system will automatically mark the exits near the fire as dangerous.

### Feature details:

1. User choose the exit route button:



1. The exit screen will show up, which tells the user which exit is safe, and which is not.



### Condition to use:

* Requirement 1: User must log in to the system.
* Requirement 2: The device which the user is using must be in online mode.

# Other requirements and features

## Offline mode

If the device is in offline mode, the app will show tips and tricks to help to escape fire.

## Friend function

We will add a feature for user to add friend, so they can keep watch of theirs friend status.

## Power out

The system should be placed far from the buildings so it can still operate when the power is out, which often happens when a fire occurs.

## Non functional-requirements

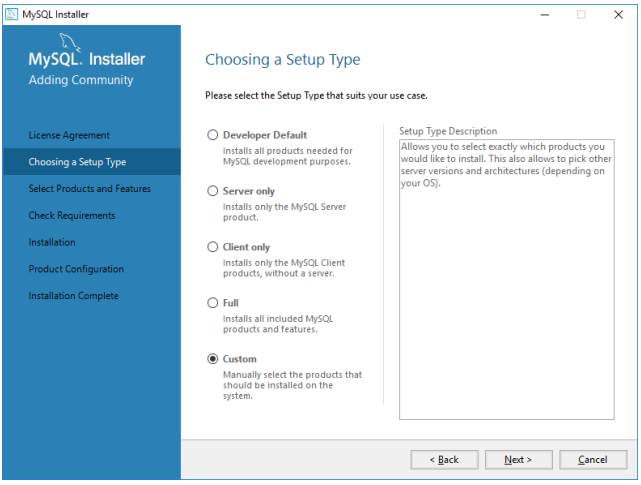
The app should be heavily optimized so it can run even on weak machines.

The app should take only 80MB of storage so it can be downloaded and opened as fast as possible.

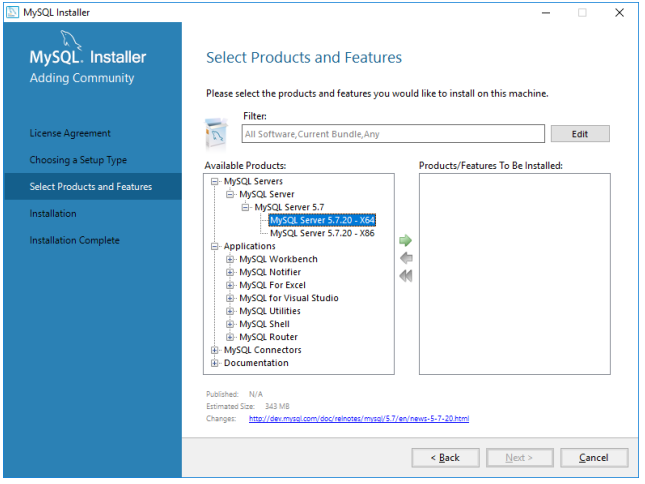
# Demo program

## Installing database

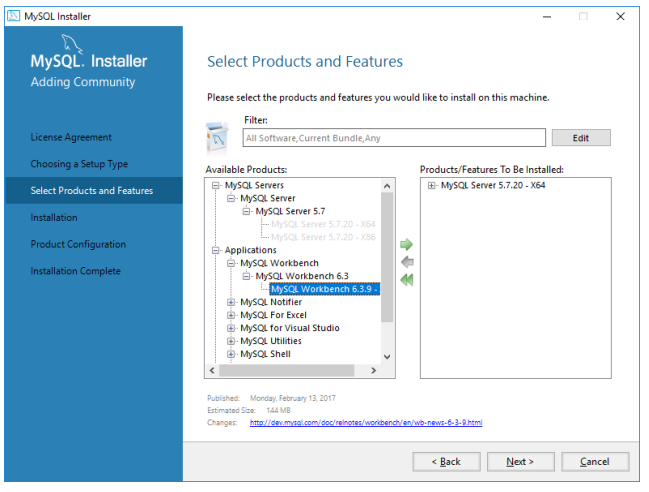
* Download MySQL installer here  
  <https://dev.mysql.com/downloads/installer>
* Run the installer and follow the steps described below:
* Select **Custom**:



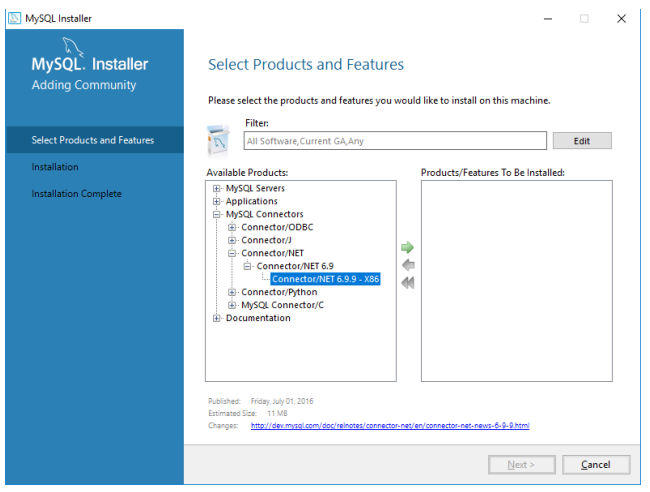
* Select **MySQL Server**:



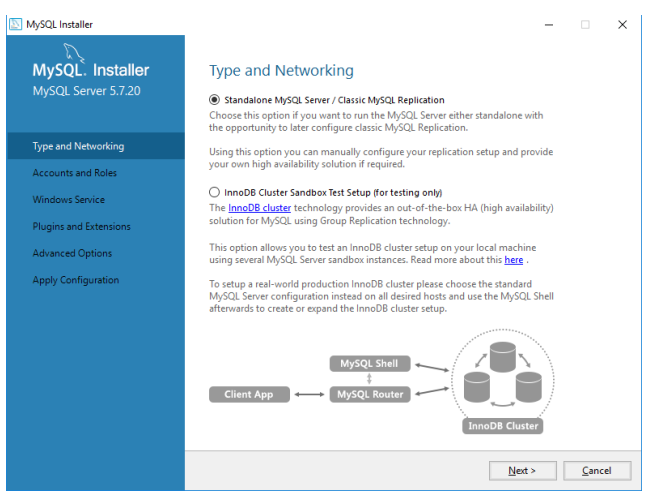
* Select **MySQL Workbench**:



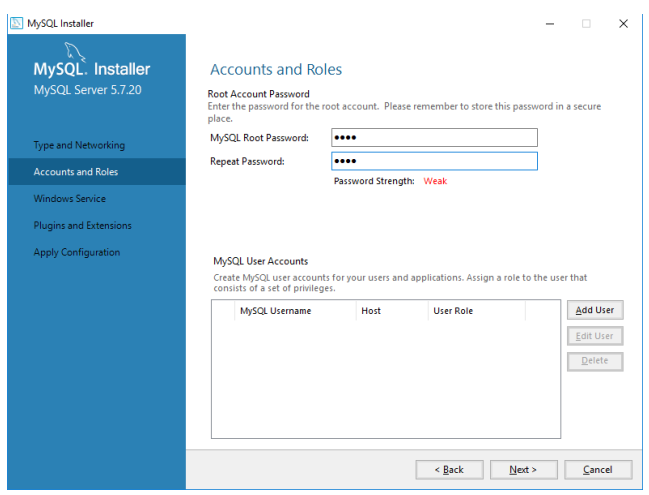
* Select **Connector/NET**:



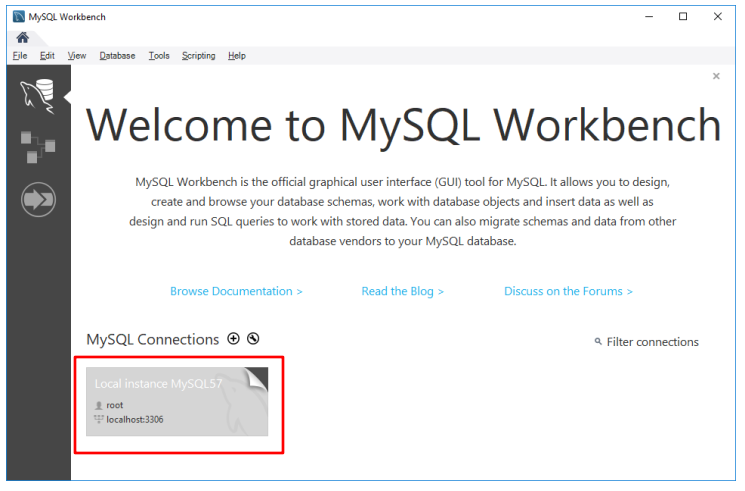
* Select **Standalone MySQL Server:**



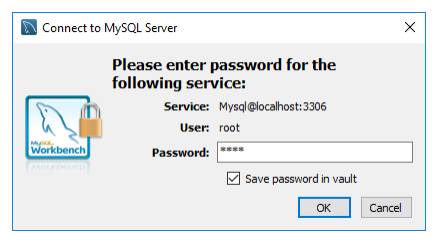
* Set **password = root**:



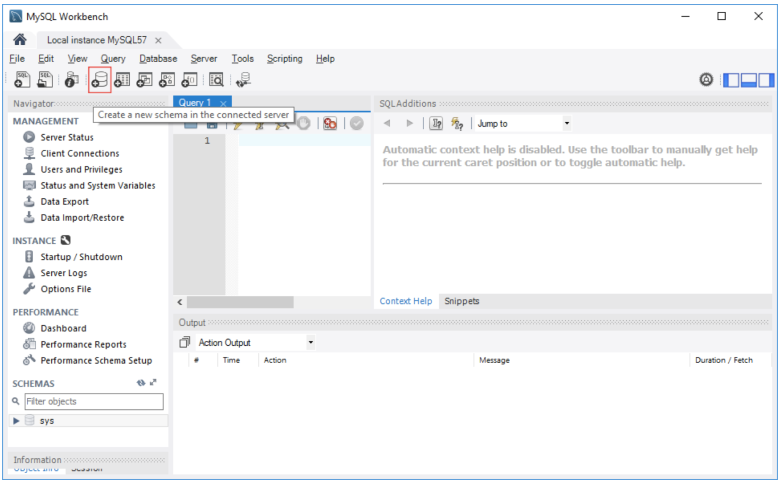
* Run **MySQL Workbench** / Click on the **Local instance:**

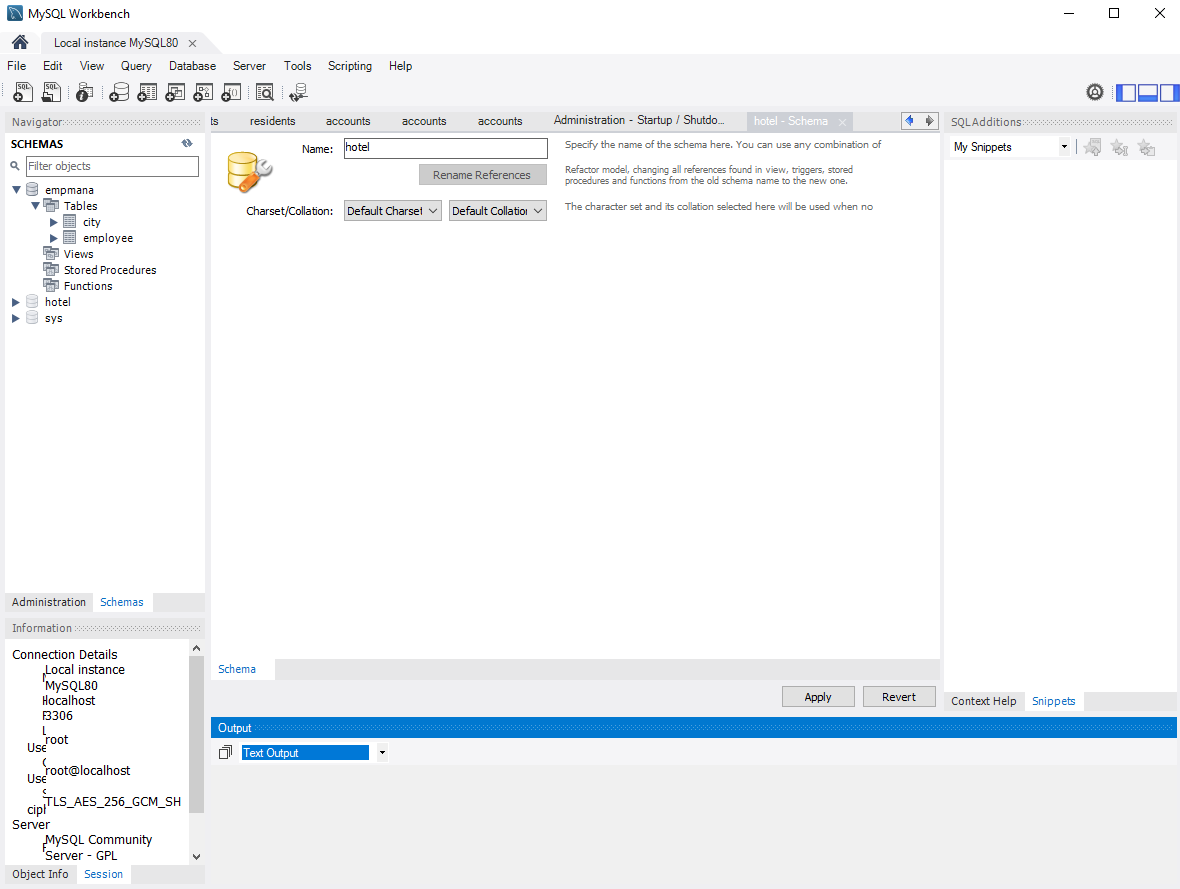


* Enter password (root):



* Click **Create a new schema**

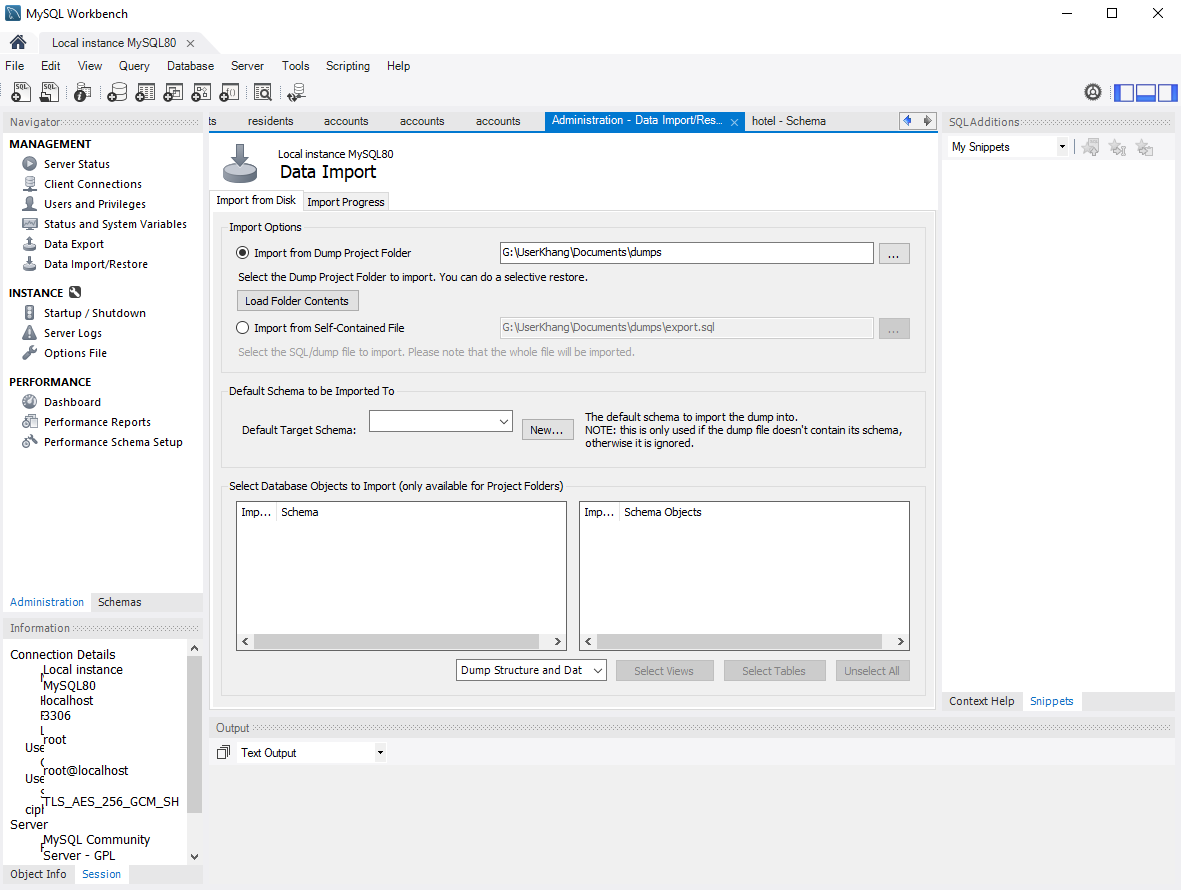


* Name it **hotel**
* 

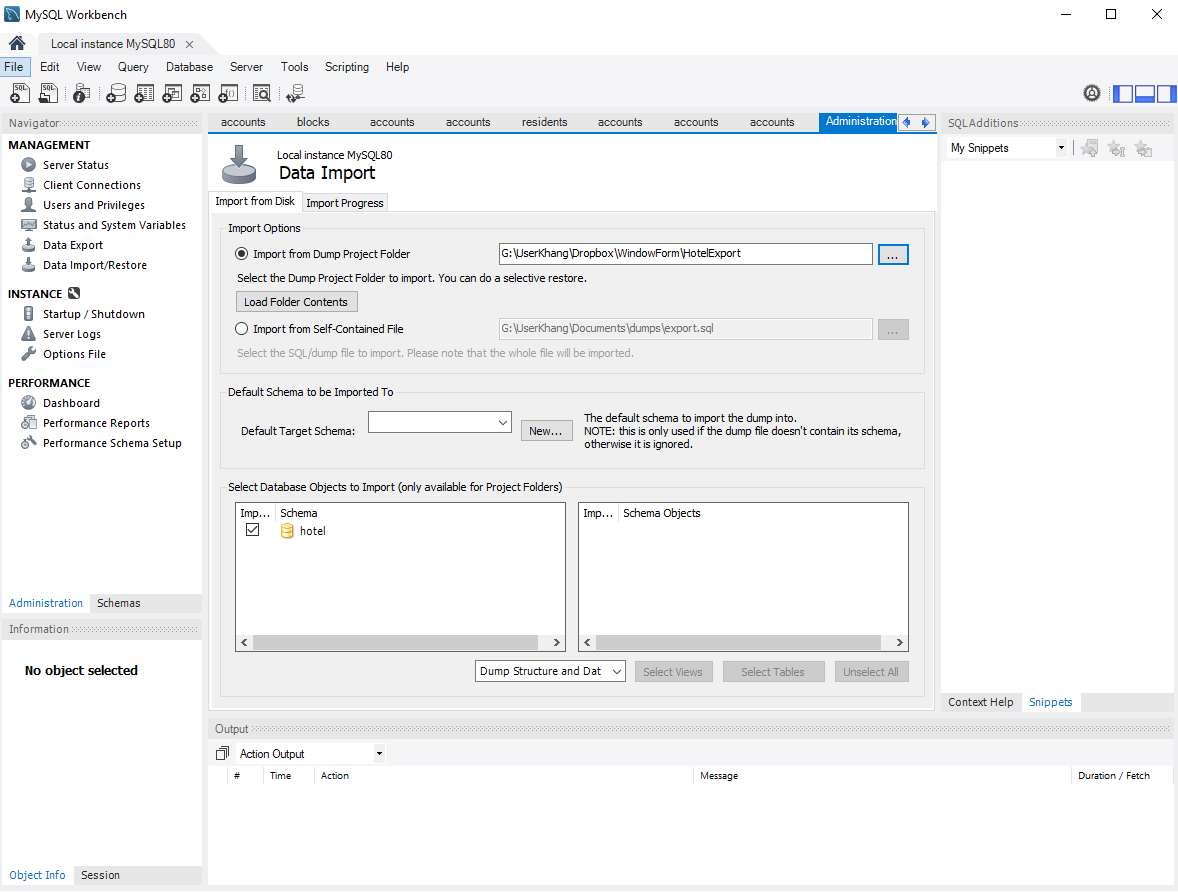
Click apply



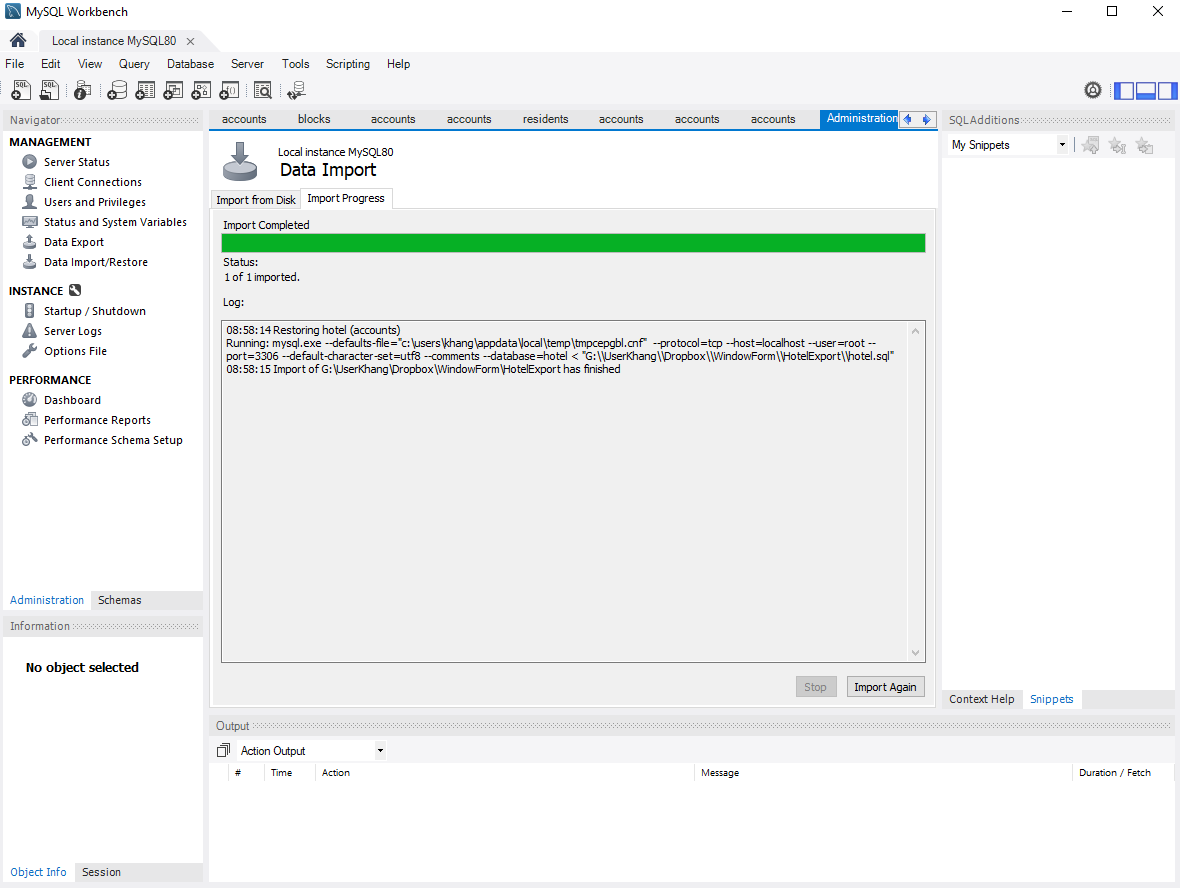
* Go to administration →Data import/restore:



* Choose Hotel export folder from GitHub link:

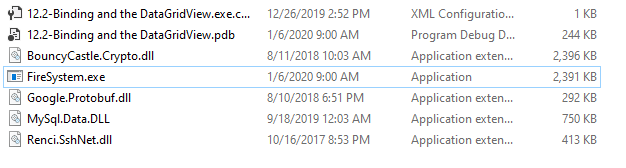


* Click import:



## Fire system program

* Download **ReleaseFireSystem** folder from github
* Run FireSystem.exe



* Manager username : khang
* User username : linhHenry
* All password : 1234
* *Note* : if encounter “Unable to connect to SQL host” error, that means the sql server hasn’t been started yet.