CSCE 5430 SOFTWARE ENGINEERING

10/14/2020

Group Name: Mystic Gang

Group Members: Abhinav mamidipelly

Harshavardhan reddy goli

prakyath reddy kandimalla

udhaya kumar gutta

Image Based Food size and Calorie Estimation

Using CNN Classifier

**Contents**

[Requirements 3](#_Toc54968913)

[Functional requirement 3](#_Toc54968914)

[Non-Functional requirements 5](#_Toc54968915)

[UML design 6](#_Toc54968916)

[Class diagram 6](#_Toc54968917)

[Sequence 7](#_Toc54968918)

[Sequence diagram for User Login 7](#_Toc54968919)

[Sequence Diagram for Admin Login 8](#_Toc54968920)

[Sequence diagram for User Registration 9](#_Toc54968921)

[Sequence diagram of System 10](#_Toc54968922)

[**Use case diagram** 11](#_Toc54968923)

[Test cases 13](#_Toc54968924)

[Who has coded which components? 17](#_Toc54968925)

[User manual and installation 17](#_Toc54968926)

[Test cases scenarios 21](#_Toc54968927)

[Feedback received during the peer review session and actions taken on the feedback. 25](#_Toc54968928)

**Members contribution Table**

|  |  |  |
| --- | --- | --- |
| **Member name** | **Contribution description** | **Overall Contribution (%)** |
| Abhinav Mamidipelly | UML design, Overall Flow of Document | 25 |
| Harsha Vardhan Reddy Goli | Test Cases, Overall Flow of Document | 25 |
| Prakyath Reddy Kandimalla | User Manual and Installation, Document Review | 25 |
| Udhaya Kumar Gutta | Requirements and Feedback | 25 |

# **Requirements**

## Functional requirements:

**ADMIN**: Main user of application

**USER:** An end-user of the application will get to know the calories of food.

1. **Admin Login**:

Login page of Admin, to this page admin, can navigate by **Homepage > Admin** option. For Admin, there is no need for the register; the user id and password are fixed in the database at a time manually. After entering the user id and password admin need to click on the ‘Login’button. The next action method will connect to the database and verifies user id and password; if given login data is valid, then Admin will get the admin home page window.

1. **Upload Calorie Dataset**:

Our application calculates food calories based on food identification from images; the application must maintain food calories from food items. For this, the admin will upload an excel file which has food name and calories per gram. After logging into the application, the admin will upload the dataset, and this data gets stored in the database.

1. **View Dataset**:

After uploading a dataset of food calorie, the admin can view the dataset for verification. Admin after login, admin can click on the view dataset option, then the application will retrieve data from the database and present in tabular format.

1. **User Registration**:

To use users' functionalities, one should create an account and log into the user portal. To create an account, a user should register by providing personal details like Name, email, contact, and password. These details will store in the database, and after registration, the user will get an alert box of 'Successfully Register.

1. **User Login**:

Users can navigate to the user portal after authenticating by login option. Users can enter email id and password, and then the system will verify whether the account is valid or not and allow the user into the portal if it’s right.

1. **Upload Image**:

For food, calorie detection, users should upload a food image and enter the number of grams. The application will return the calories of the food.

1. **Calculating calories and training dataset**:

In this feature, after uploading the image, simultaneously the user needs to give the additional information about the food. Here, it is number of grams of the food. So that our desktop application can calculate the calories of the food. We maintain the food calorie per gram in database. We will be training few food items dataset with CNN algorithm. This dataset consists of 101 food item images each food item categories consists of more than 1000 images; we’ll train the dataset with neural network and get the result.

1. **User output**:

After uploading the image, if the food image is trained in our dataset, we will get the user output in a textbox.

## 

## Non-Functional requirements

1. **Usability**

It is very simple and user-friendly to use, it has very straight forward options for both admin and users. For the users it is easy to sign up with the basic info and can easily fill up and get an account created. Even we can upload the image with a single click on the upload image button and can enter the grams of data in a textbox which helps in getting the accurate result. For the admins it easy to update, upload, and view the data whenever required.

1. **Performance Requirements:**

Our application has high performance as we train data with lots of images. Calorie calculation is fast and depends upon the execution time. As we use more filters in CNN classifier, we will get the accurate results and the image resolution is high. The application will not get breakdown as long everything is running in the background.

1. **Accuracy:**

As we know that CNN is good at filtering the images will be training a food item with lot of images into it. If the image of the user matches with that of the image in the database, we can have an accuracy of nearly 97%.

1. **Extensibility:**

This application we can extend for including some more requirements if it is necessary.

1. **Availability:**

As it is a desktop application, we can run this anytime we want with the required credentials whenever the user wants.

1. **Data Integrity:**

As we are using CNN classifier for training the data, our results are mostly accurate and consistent throughout the lifecycle.

# **UML design**

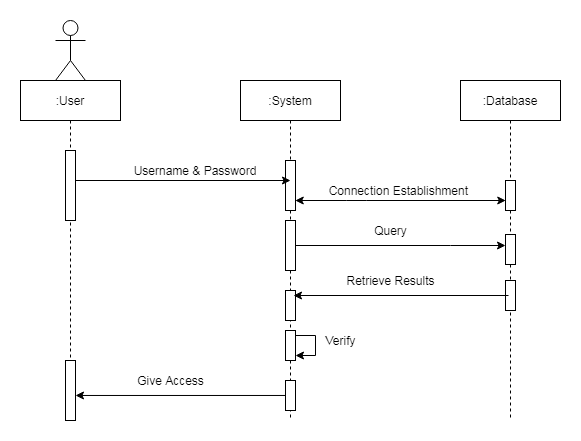
## **Class diagram**

Graphical user interface, application, Teams

Description automatically generated

## **Sequence Diagram**

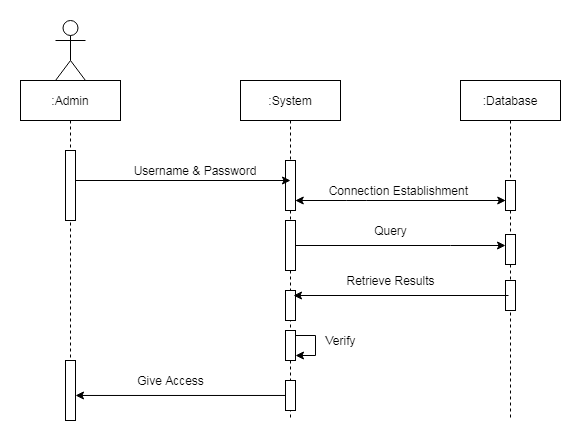
### Sequence diagram for User Login



### 

### 

### Sequence Diagram for Admin Login

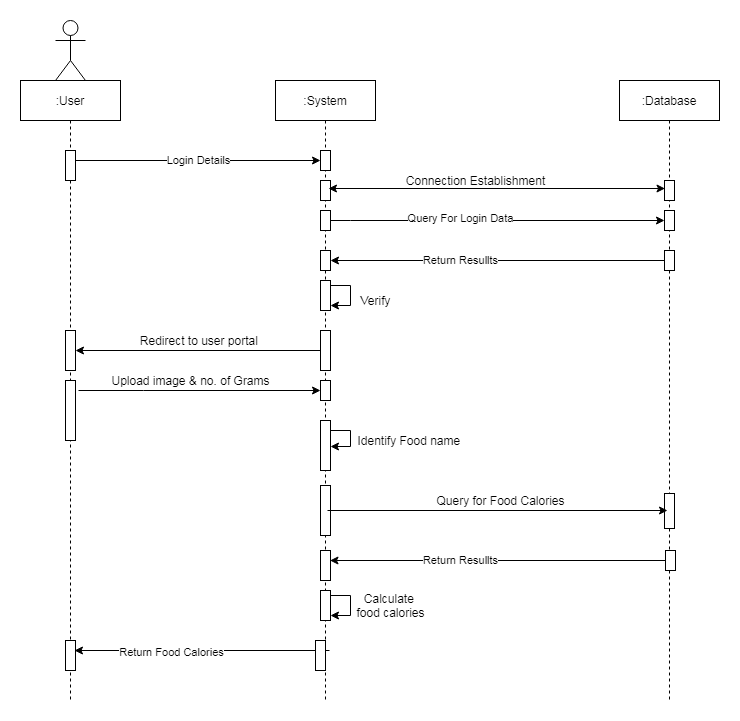


### Sequence diagram for User Registration

### 

### 

### Sequence diagram of System



### 

### Use case diagram

Normal Case:



**Error Case 1:** User Login Error

Graphical user interface, application, chat or text message

Description automatically generated

**Error Case 2:** User Registration Error

Graphical user interface, application, chat or text message

Description automatically generated

# **Test cases**

# 

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Signup** | | | | **Test Priority** | | | **High** |
| **Test case description** | **Positive** | | | | **Post requisite** | | | **NA** |
| **Prerequisite** | **NA** | | | | | | | |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Result** | **Comments** | |
| Launch Application | | Execute Home.py file | Application Home page | Application Home page | | Pass | Successful | | |
| Enter correct name, email id, password, contact no, address and hit register button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\*\*\*  Name: Harsha  Contact no: 9403046367  Address: Frisco | Registration success | Registration success | | Pass | Successful | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Registration** | | | | **Test Priority** | | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | | **NA** |
| **Prerequisite** | **NA** | | | |  | | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** | |
| Launch Application | | Execute Home.py | Application Home Page | Application Home Page | | Pass | Successful | |
| Enter incorrect Email ID, and enter Name, contact no, address and password then hit register button | | Name: Harsha  Email id: [harshagmail.com](mailto:harsha@gmail.com)  Contact no: 9403046367  Address: Frisco  Password: \*\*\*\*\*\*\*\*\*\* | Error message - Invalid Email ID | Error message – Invalid Email ID | | Pass | Successful | |
| Enter incorrect Contact no. and correct Name, password, Email ID, address and hit register button | | Name: Harsha  Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Contact no: 9403046367  Address: Frisco  Password: \*\*\*\*\*\*\*\*\*\* | Error message – Invalid Mobile Number | Error message – Invalid Mobile Number | | Pass | Successful | |
| Don’t enter name and correct email id, contact no, address and hit register button | | Name:  Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Contact no: 9403046367  Address: Frisco  Password: \*\*\*\*\*\*\*\*\* | Error message – Fill out all fields | Error message – Fill out all fields | | Pass | Successful | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Login** | | | | **Test Priority** | **High** | |
| **Test case description** | **Positive** | | | | **Post requisite** | **NA** | |
| **Prerequisite** | **Valid user account** | | | |  |  | |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | **Test result** | | **Test comments** |
| Launch Application | | Execute home.py | Application Home Page | Application Home Page | Pass | | Successful |
| Enter correct email id, password, and hit login button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\*\*\* | Login Success | Login Success | Pass | | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Login** | | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid user account** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Application | | Execute Home.py | Application Home Page | Application Home Page | | Pass | Successful |
| Enter correct email id, wrong password, hit login button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\* | Alert:  Login Fail | Alert:  Login Fail | | Fail | Failure |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **Admin Login** | | | | **Test Priority** | | **High** |
| **Test case description** | **Positive** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid Admin account** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Application | | Execute Home.py | Application Home Page | Application Home Page | | Pass | Successful |
| Enter correct Admin credentials | | Email id: admin  Password: \*\*\*\* | Alert:  Login successful | Alert:  Login Successful | | Pass | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **Admin Login** | | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid admin credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Application | | Execute Home.py | Application Home Page | Application Home Page | | Pass | Successful |
| Enter incorrect Admin credentials | | Email id: admin  Password: \*\*\*\* | Alert:  Login Fail | Alert:  Login Fail | | Fail | Failure |

# **Who has coded which components?**

The project has the following modules and for coding they are distributed among the team members as follows.

* User Home, User registration modules done by Harsha Vardhan Reddy Goli
* Admin Home, Admin Login modules done by Abhinav Mamidipelly
* Database connection done Prakyath Reddy Kandimalla
* User Login module done by Udhaya Kumar Gutta

# **User manual and installation**

The user can run code if s/he has the python 3.8 or higher version already installed or otherwise s/he can download and install, for installation s/he follow the following steps.

1. Download any version of Python 3 from the following link or the python can be downloaded from the Microsoft store.

<https://www.python.org/downloads/>

Graphical user interface, text, application

Description automatically generated

1. Need to install MySQL database software.
2. Need to install PyQt5 module API.

**pip install pyqt5-tools**

Text

Description automatically generated

1. Install MySQL Connector API of python.

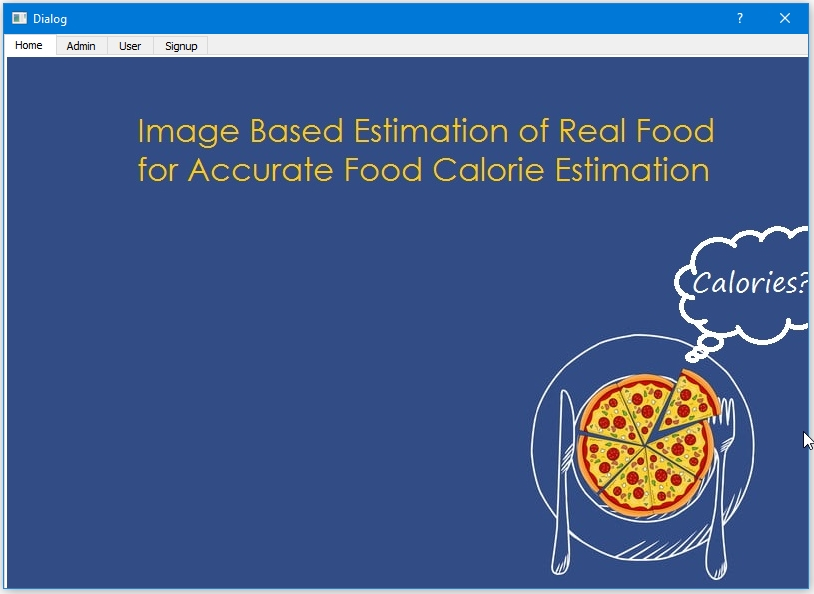
**pip install mysql-connector**

1. Run python homepage file of ‘home.py’

Text

Description automatically generated

1. User can see the homepage of application

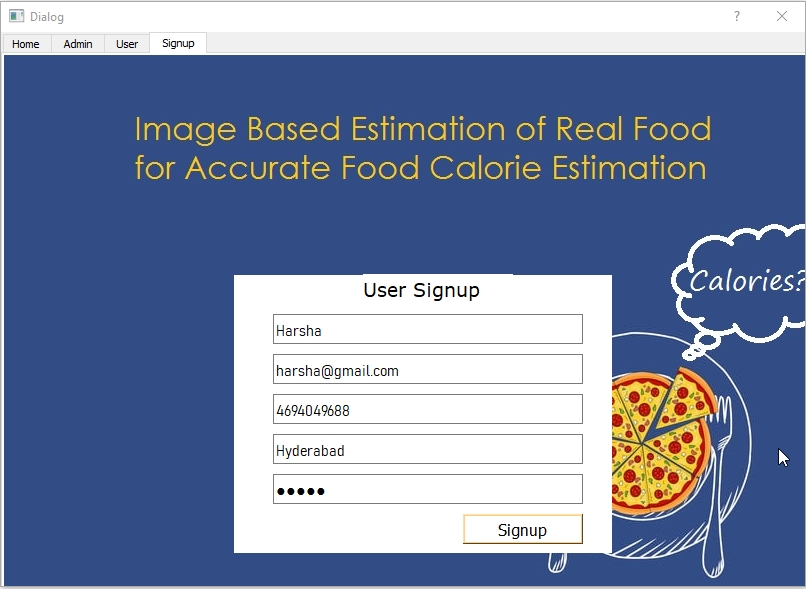


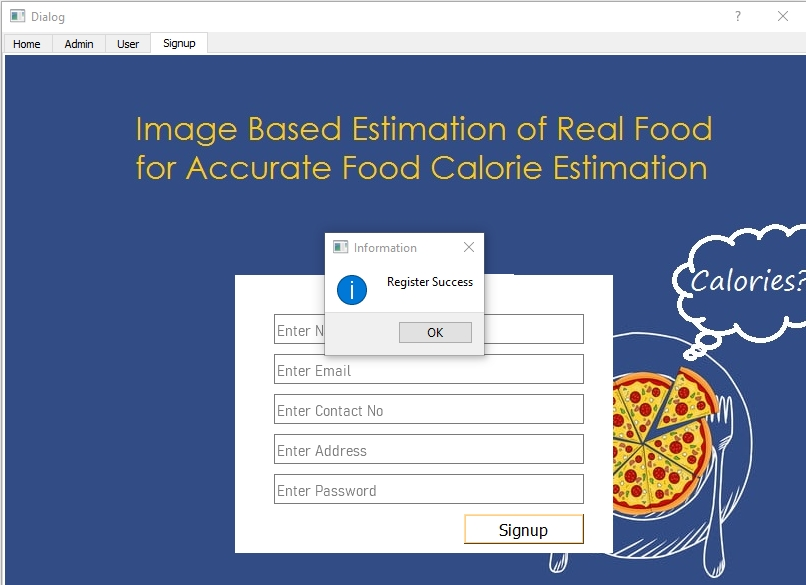
# 

# **Test cases scenario**

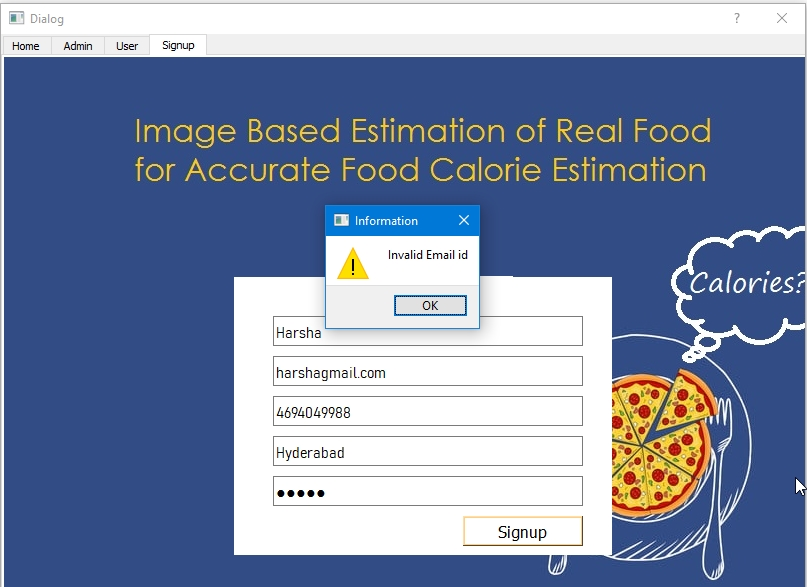
* Test cases screenshots

**Registration successful scenario**

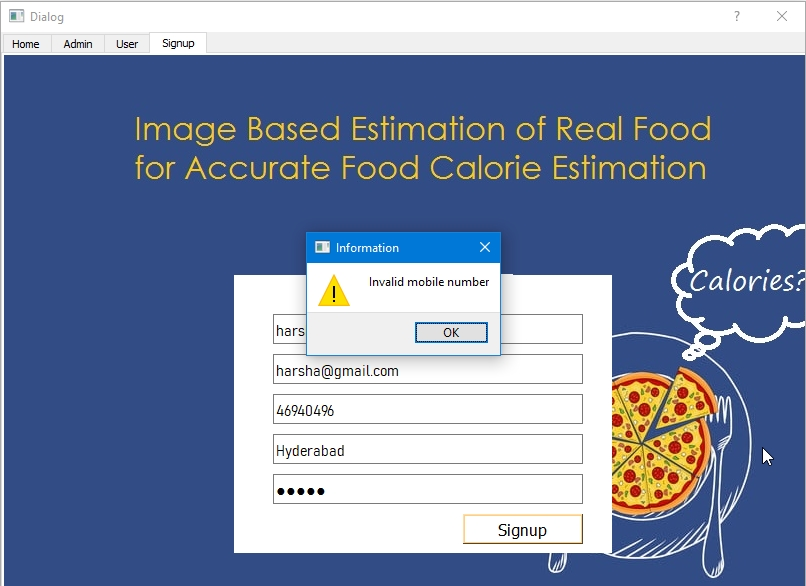




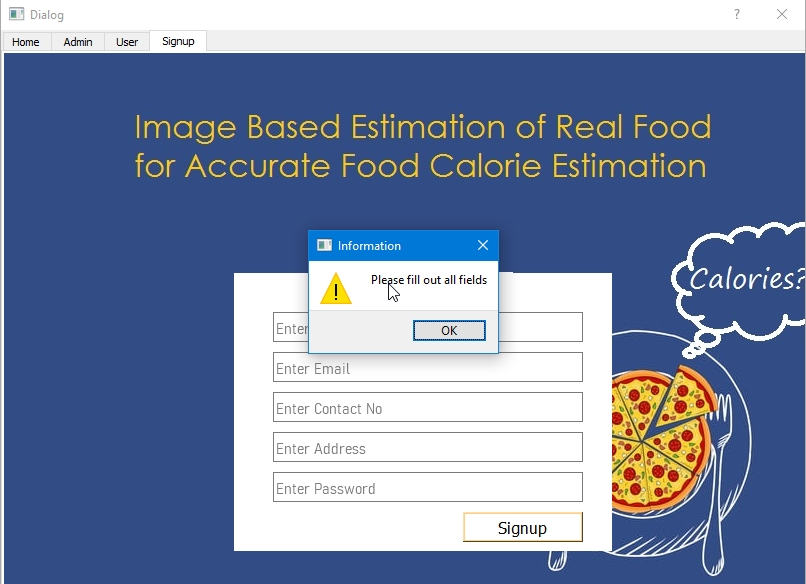
**Invalid email ID during registration**



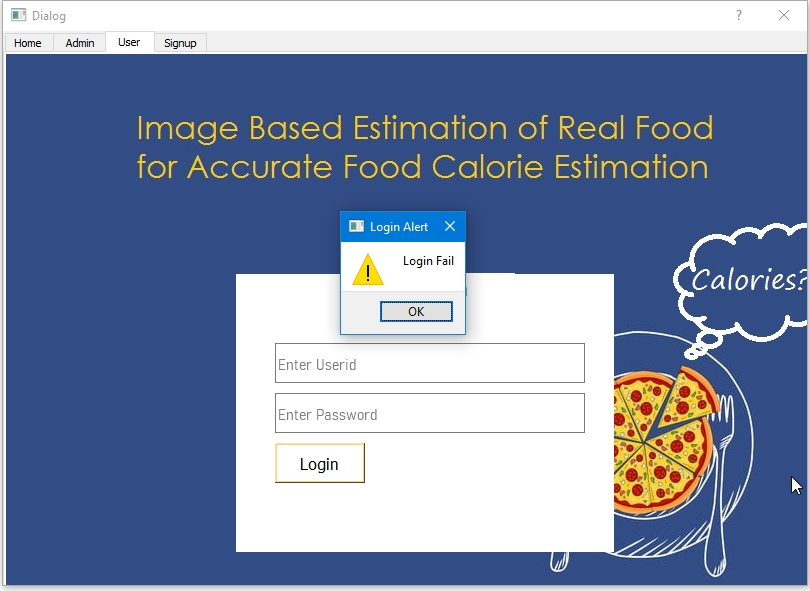
**Invalid Mobile number during registration**

****

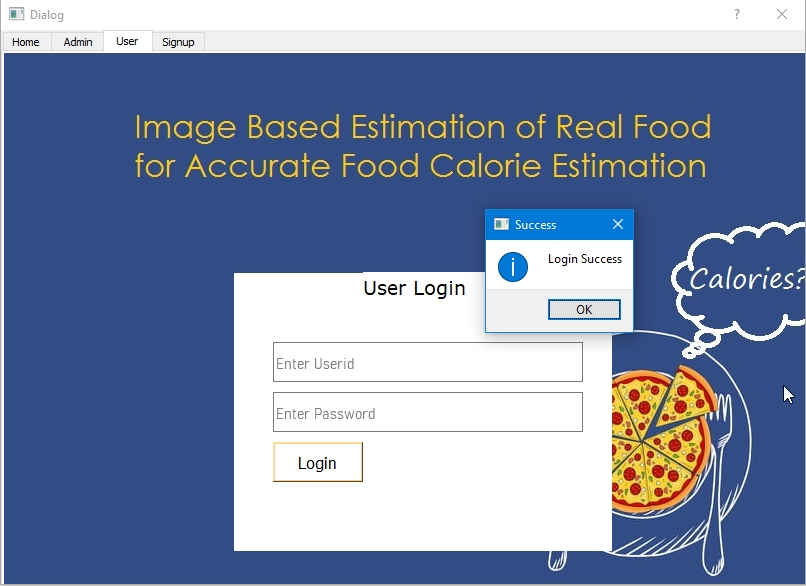
**Blank entries during registration**

****

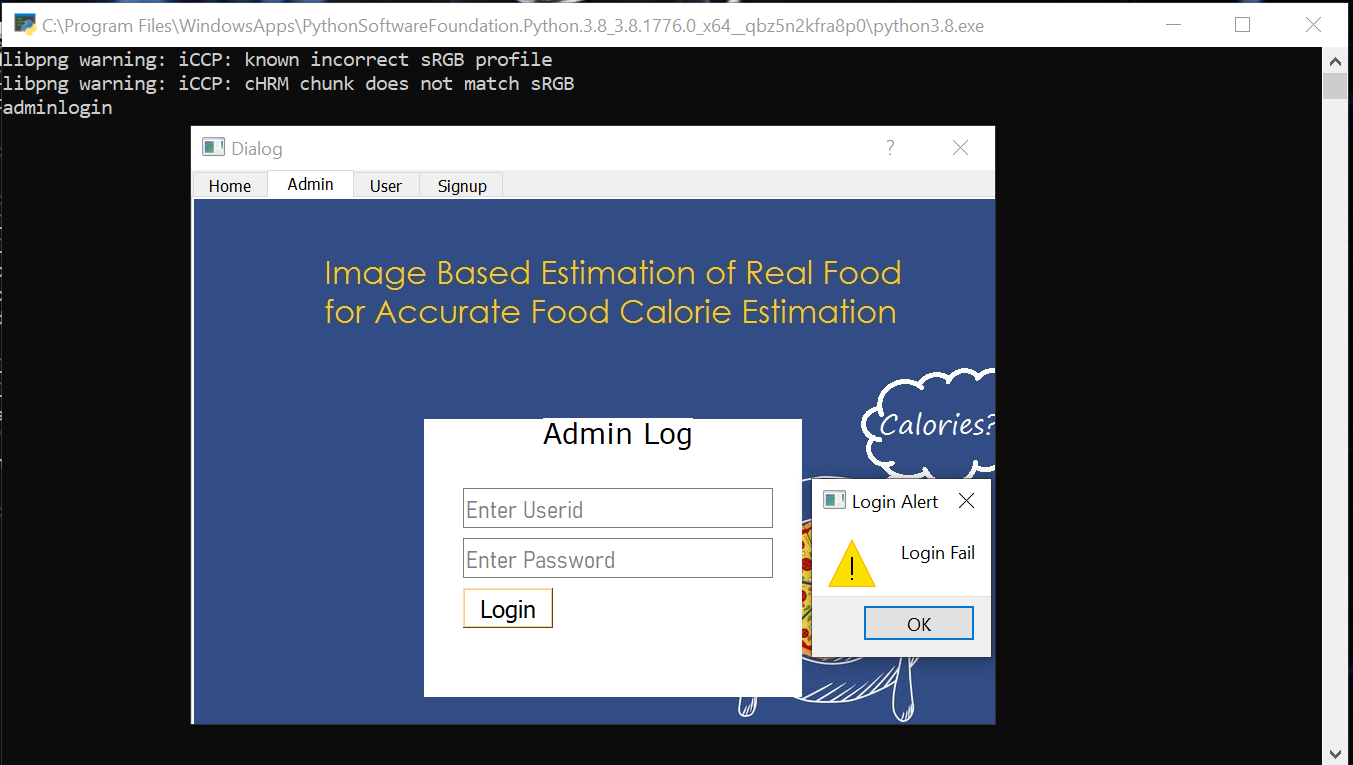
**Invalid login Credentials**



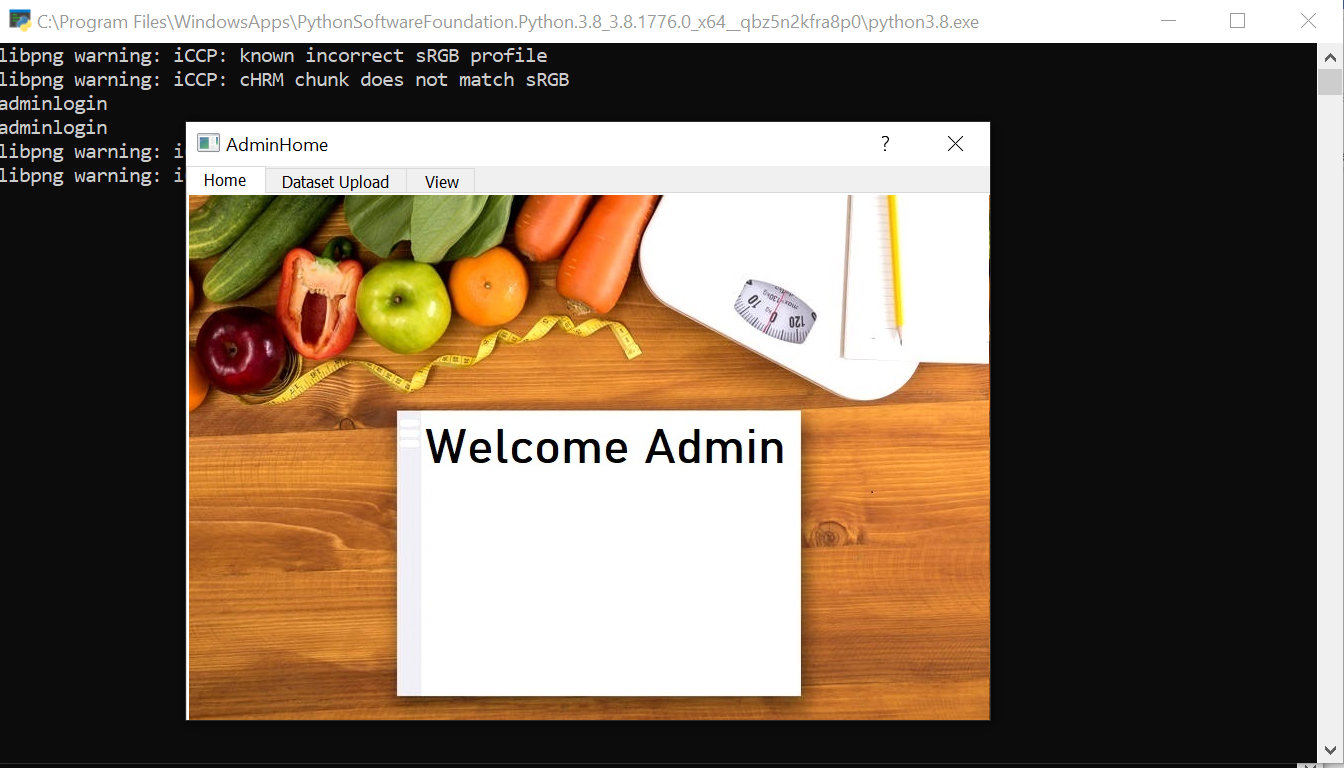
**Successful Login**

****

**Admin Login Failed Scenario**



**Successful Admin Login**



# **Feedback received during the peer review session and actions taken on the feedback.**

We have received feedback from Professor and tech coders. Based on the feedback, we have changed a few things. We have added the document's main features with a clear explanation of what calculating calories information is and displaying the outputs to the users. We have also updated the information about open databases in our document. We have added the UI screenshots for better understanding. We have improved the non-functional requirements sections and removed some unwanted functionalities.

Our peers gave great suggestions to us, and we implemented a few of those. They were stressing about improving our document. We took the suggestion and improved the documentation. They mentioned that some areas in our document were not well addressed (Assumptions, Abbreviations). We took the suggestion and changed that.

They asked us to mention safety requirements and software quality attributes. We did not add these to our project as we thought they weren't necessary. They asked us to specify requirements or suggestions to the developer. We took that suggestion and added a few images for a better understanding in that regard.

They asked us to train more food items. We were not able to include this because of memory management issues. They asked us to give a better explanation of CNN. We took the advice and bettered it.