A Major Project Report on

Snap&Eat

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering in Software Engineering at Pokhara University

By

ISHA BARAL

SWIKRITI TRIPATHI

YAMUNA ADHIKARI

Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(JULY, 2021)

A Major Project Report on

Snap&Eat

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering in Software Engineering at Pokhara University

By

ISHA BARAL

SWIKRITI TRIPATHI

YAMUNA ADHIKARI

Supervisor

Er. SHUBHAM AGRAWAL

Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(JULY, 2021)

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the support and guidelines to complete this project. First and foremost, we would like to express our sincere gratitude towards **Er. Shubham Agrawal,** our project supervisor, for his constant guidance and precious encouragement. Without his invaluable supervision, it would have been a difficult journey for us. His useful suggestions and cooperative behavior for this whole work are sincerely acknowledged. We would like to thank the principal **Mr. Birendra Khadka** and the vice principal **Mr. Ashok Raj Parajuli** for supporting and inspiring us during the project. We are highly indebted to **Er. Sujan Tamrakar,** our project head, for his continuous encouragement and support throughout the project. We would like to thank and express our gratitude to all our respective subject teachers for sharing their precious knowledge, constant support, and guidance.

Lastly, we wish to express our heartfelt gratitude to our families, parents and friends, who have always served as the strongest source of inspiration in all our ventures.

Isha Baral (BE2016SE15)
Swikriti Tripathi (BE2016SE45)
Yamuna Adhikari (BE2016SE46)

Gandaki College of Engineering and Science
Lamachaur, Kaski, Nepal

ABSTRACT

Snap&Eat is an android application that classifies the category of a pizza and estimates its total calories based on the picture uploaded by the user. We trained a CNN model using a dataset of about 1000 pizza images belonging to 10 different categories. Using the model, we predicted the given pizza as either Margherita pizza, Mushroom pizza, BBQ chicken pizza, Mexican pizza, Vegetable pizza, Pepperoni pizza, Hawaiian pizza, Broccoli pizza, Cheese pizza, or White pizza. We estimated the total calories present in the particular pizza. Finally, the picture uploaded by the user is displayed in the newsfeed along with the estimated pizza category and respective calories.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

1.1 BACKGROUND

Food is an integral part of our life. The quality of our life and health condition directly depends on the food we consume daily.

Similarly, we love sharing the picture of our meal on several social media. But are we aware of the total calories we intake per day? Is there a way to track everything we eat?

Keeping all these things in mind, we came up with a solution. We developed an application where we can easily see the total calories we ate with just a picture. Along with it, the users can share their meals with everyone. It is applicable for food enthusiasts and bloggers in general, to promote healthy eating.

1.2 PROBLEM STATEMENT

Owing to a busy schedule and lack of proper knowledge on nutrition and diet, our consumption is going unmonitored these days. Just because some food seems delicious or tastes delicious, we consume them without knowing its calorie count.

Food blogging culture is equally responsible for our eating habits these days. They spend hours making their blogs attractive, which are targeted mainly to attract as many people as possible. They are at fault for not disclosing full information on food like total calories, nutritional level, etc.

Our consumption is much more than what is necessary for our bodies. It harms our health in a long run.

1.3 OBJECTIVES

The main objectives of this application are:

- To recognize the pizza category uploaded by the user.
- To estimate total calories based on the uploaded picture.

1.4 IMPLICATIONS

Our proposed application can be applied in the field of the food industry and restaurants. It can also be used by people in general since people are becoming conscious about their diet these days. It can come in handy when calculating the calories present in our daily food.

Chapter 2

LITERATURE REVIEW

Our application requires a large-sized database. For that, we collected around 1000 images of pizza from several sites like (Instagram, n.d.), (Pinterest, n.d.). These 1000 images are divided into training and testing data i.e. 700 training images and 300 testing images. Likewise, we went through some research papers based on convolutional neural networks and image processing (Manal Chokr, 2017). We reviewed the following applications as they were similar to ours.

- **Calorie Mama**:

  Calorie mama (Inc., 2017) is a calorie tracking app based on deep learning and image classification technology. This application uses its food recognition machine learning AI to identify various foods. This app uses a wide range of comprehensive databases and can recognize varieties of foods. It can also recommend customized diets to its user as well.

CALORIE MAMA
With each photo Calorie Mama becomes better and better at identifying your food!

Figure 1: Calorie Mama

- **Fatsecret- Calorie Counter App**:

Fatsecret (FatSecret, 2020) is an app to find nutritional information for the food you eat and keeps track of your meals, exercise, and weight. It is like a journal to keep track of calories one consumes and their current weight. One extra feature in this application is an exercise diary to record the calories you burn.

Figure 2: Fatsecret

Chapter 3

TOOLS AND METHODOLOGY

3.1 REQUIRED TOOLS

We required various tools for the development of our application. Here are some lists of tools that were used in the project:

- Java, XML: For android development
- Python: For Machine Learning
- MySQL database: For remote storage
- MS-Word: For preparing a report
- DRAW.IO: For UML diagrams
- Mockflow: For wireframe
- MS- Powerpoint: For slides

3.2 APPROACH USED

Every software development methodology acts as a basis for applying specific frameworks to develop and maintain software. Several software development approaches have been used since the origin of information technology. Since the system we developed is small, the agile model was used in our project.

3.3 ALGORITHMS

We incorporated the CNN algorithm in our system.

3.3.1 CNN FOR IMAGE CLASSIFICATION

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take the image as an input, assign importance (learnable weights and biases) to various aspects/objects in the image, and can differentiate one from the other. CNN has an input layer, output layer, and hidden layer. The hidden layers usually consist of convolutional layers, ReLU layers, pooling layers, and fully connected layers.

A classic CNN architecture would look like:

Input ->Convolution ->ReLU ->Convolution ->ReLU ->Pooling ->ReLU ->Convolution ->ReLU ->Pooling ->Fully Connected

The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/characteristics. The role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

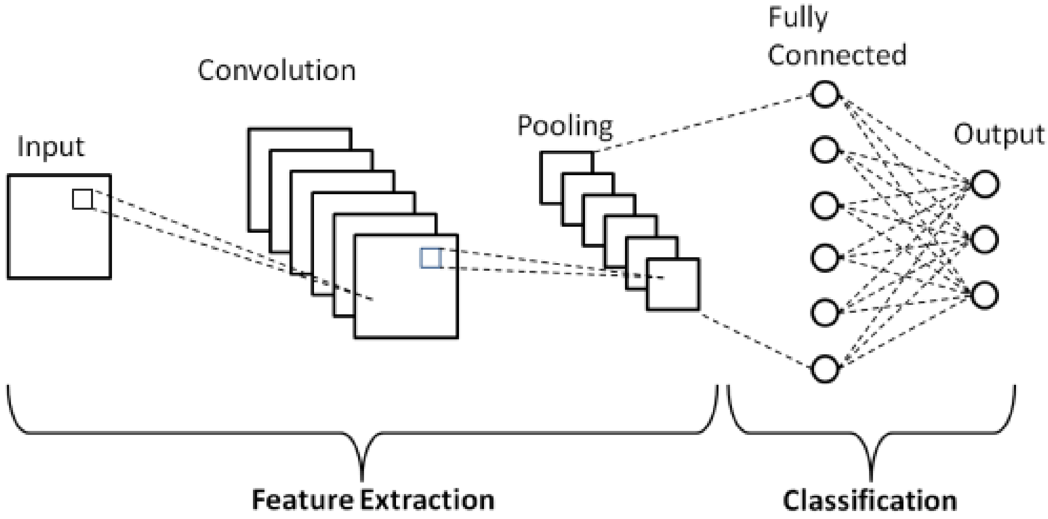


Figure 3: Steps in CNN

Part 1: Image Recognition

The first part of CNN is to filter the input image to extract features from it, and then pool the extracted feature to reduce the data size, and finally add an activation function so that the network is a non-linear function. This can be summed to convolution + pooling + activation layers.

- Convolution: A filter is a matrix that extracts features from an image (this is where the learning takes place). The dot product between the filter values and the image pixel values forms the convolution layer. The values in the filter matrix are updated each time the network performs back propagation. In a CNN, many different filters extract various features in an image. As we progress through the network, the features extracted from an image get more and more specific.

- Re-Lu: Re-Lu (Rectified Linear Unit) is a common activation function used in neural networks. An activation function is used to add non-linearity to the neural network. The Re-Lu activation function only allows positive values. The negative values are not passed. It will speed up the process and bring down the possibility of the occurrence of a dead neuron.

  $$\mathbf{f(x) = (0,max)}$$

  In simple statement relu works as follows

  ```
  if input > 0:
          return input
  else:
          return 0
  ```

- Pooling: A pooling layer reduces the size of the filter layer which allows us to train the model faster. It makes it possible to detect objects in an image no matter where they’re located. Also, it prevents over fitting by dropping unwanted values in the filter matrix.

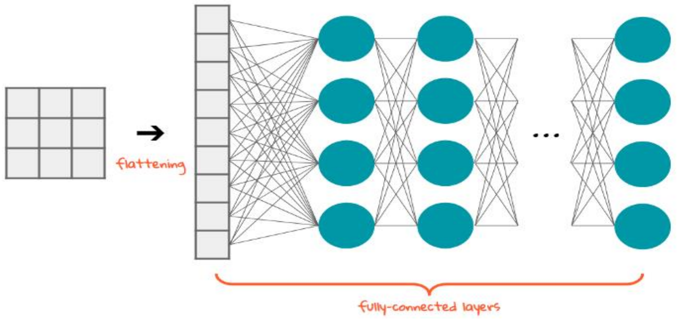


Figure 4: Flattening in CNN

Part 2: Image Classification

Now that all the features of the image have been extracted by the convolution layers, the image is now ready for classification. The whole classification part is a fully connected layer, which starts with a flattening step, then a fully connected layer, and ends with a softmax function.

- Flattening: Before entering our convoluted input into a dense (fully-connected) layer, we must flatten the tensor (our input). Flattening is turning our multi-dimensional input tensor into a 1-D input tensor. Flattening allows us to analyze every single pixel since every single pixel in the image has its own neuron.
- Fully connected layer: The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a softmax activation function in the output layer. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer. The output from the convolution and pooling layers represents high-level features of the input image. The fully connected

layer aims to use these features for classifying the input image into various classes based on the training dataset.

- Softmax function: The output from the Fully Connected Layer is then passed through the softmax function. The softmax function takes a vector of arbitrary real-valued scores and squashes it to a vector of values between zero and one that sums to one.

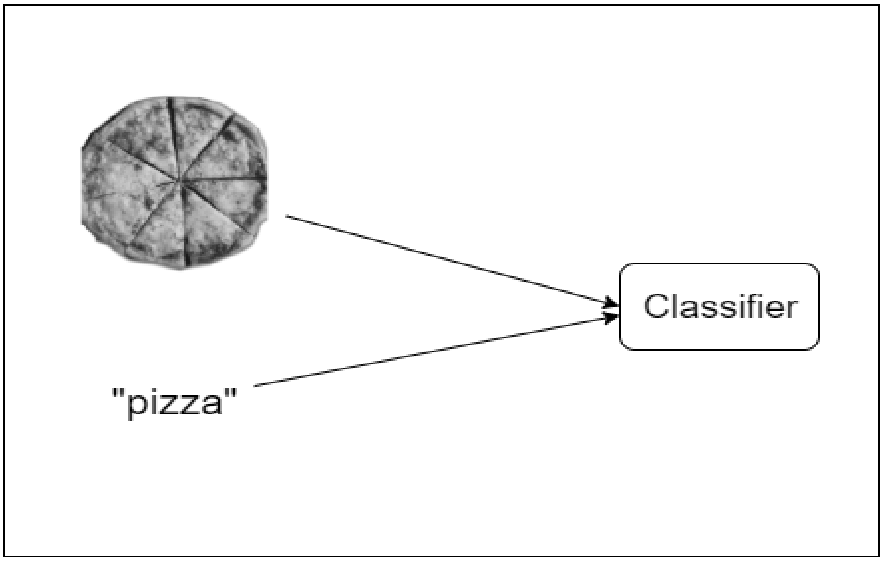


Figure 5: Training the model

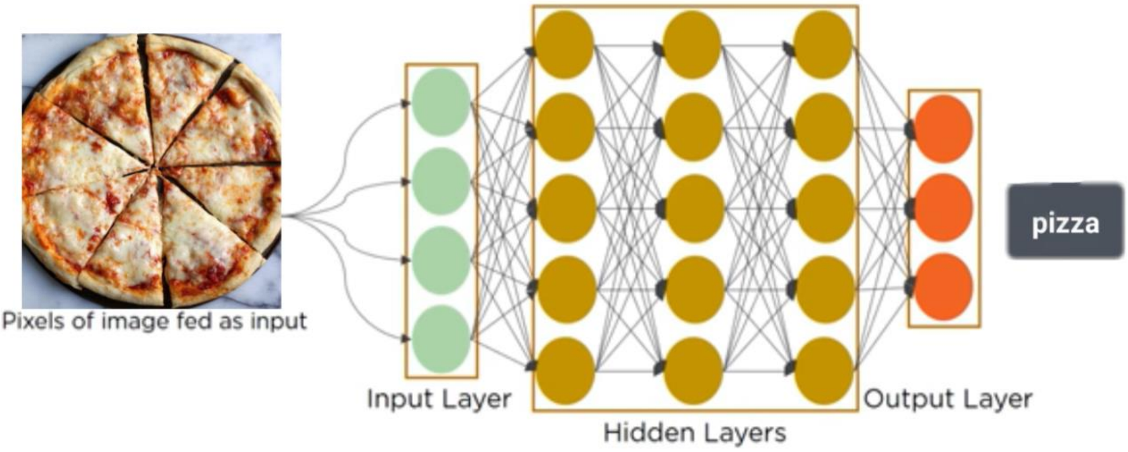


Figure 6: Layers in CNN Model

3.4 WORKING OF OUR APP

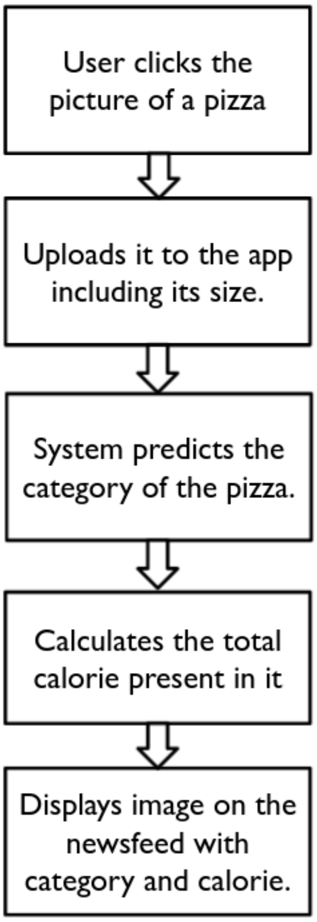

User clicks the picture of a pizza
Uploads it to the app including its size.
System predicts the category of the pizza.
Calculates the total calorie present in it
Displays image on the newsfeed with category and calorie.

Figure 7: Working of our app

Our application allows users to take a picture of the pizza and upload it including its size, for instance, small, medium, large, and so on. Then our system predicts the category of the pizza as either Margherita pizza, Mushroom pizza, BBQ chicken pizza, Mexican pizza, Vegetable pizza, Pepperoni pizza, Hawaiian pizza, Broccoli pizza, Cheese pizza, or White pizza. Then, our system estimates the total calories present in the particular pizza. Lastly, after successful prediction, the picture is displayed on the news feed along with its category and estimated calories.

3.5 DESIGN

3.5.1 USE CASE DIAGRAM

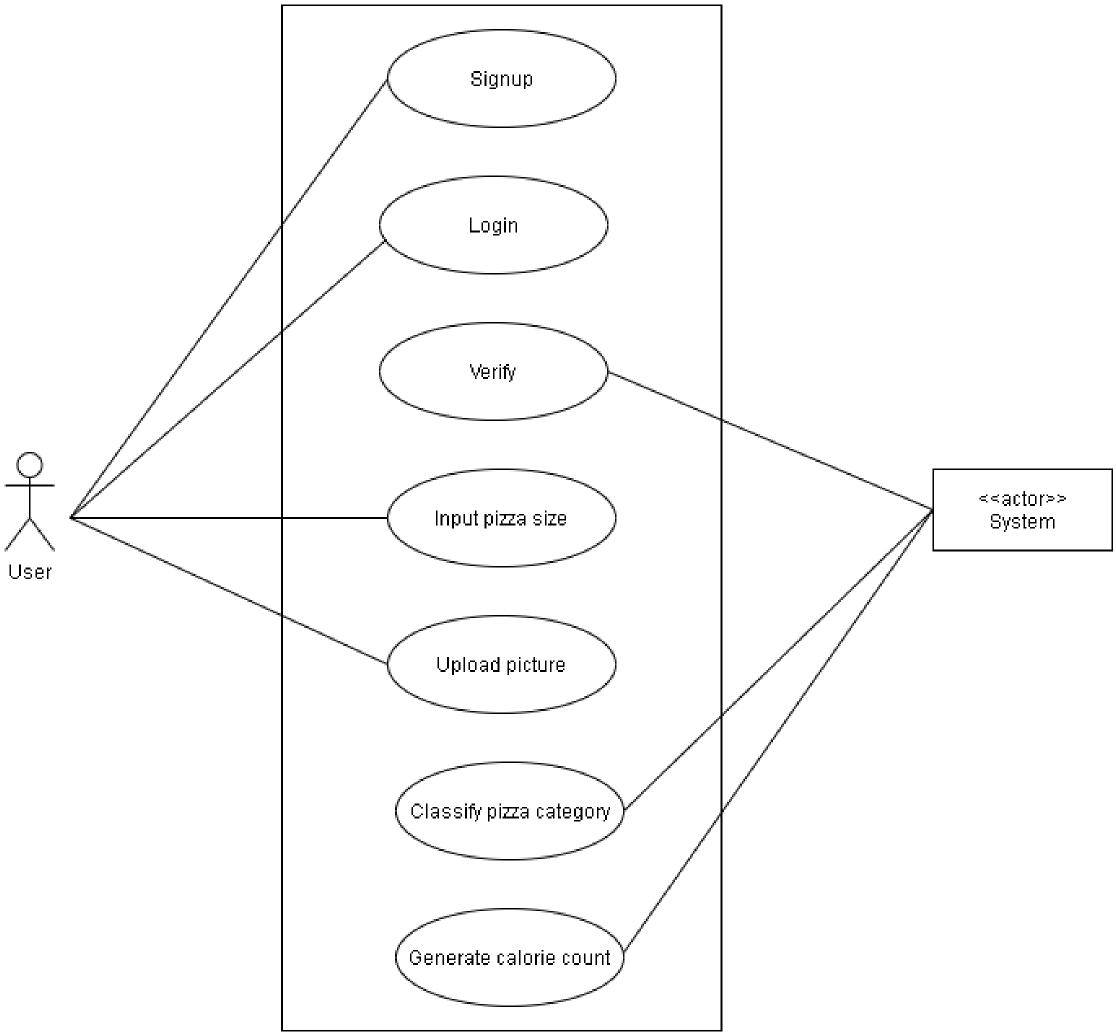


Figure 8: Use case diagram of Snap&Eat

A use case is a software and system engineering term that describes how a user uses a system to accomplish a task.

Use Case UC1: Upload photos

Primary Actor: User

Secondary Actor: System

Stakeholders:

- User: Wants to upload photos

Precondition: User must log in

Postcondition: Pizza calorie is generated

Basic flow:

- User login to the system using password

Alternate flow:

- When the user enters a wrong password, the system notifies.

Use Case UC2: Generate Calorie Count

Primary Actor: User

Secondary Actor: System

Stakeholders:

- System: Calculate the total calories of the pizza

Precondition: The user must upload the picture.

Postcondition: Total calorie of pizza is calculated.

Basic flow:

- User login to the system using a password.
- User inputs the size of pizza.
- The user uploads a picture.
- The system detects and classifies the type of pizza. For example, whether the given pizza is a mushroom pizza or a chicken pizza.
- The system calculates the total calories.

Alternate flow:

- At any time, the user enters a wrong password then the system notifies.
- If a user uploads a photo of other things rather than food, the system will discard the post.

3.5.2 SYSTEM SEQUENCE DIAGRAM (SSD)

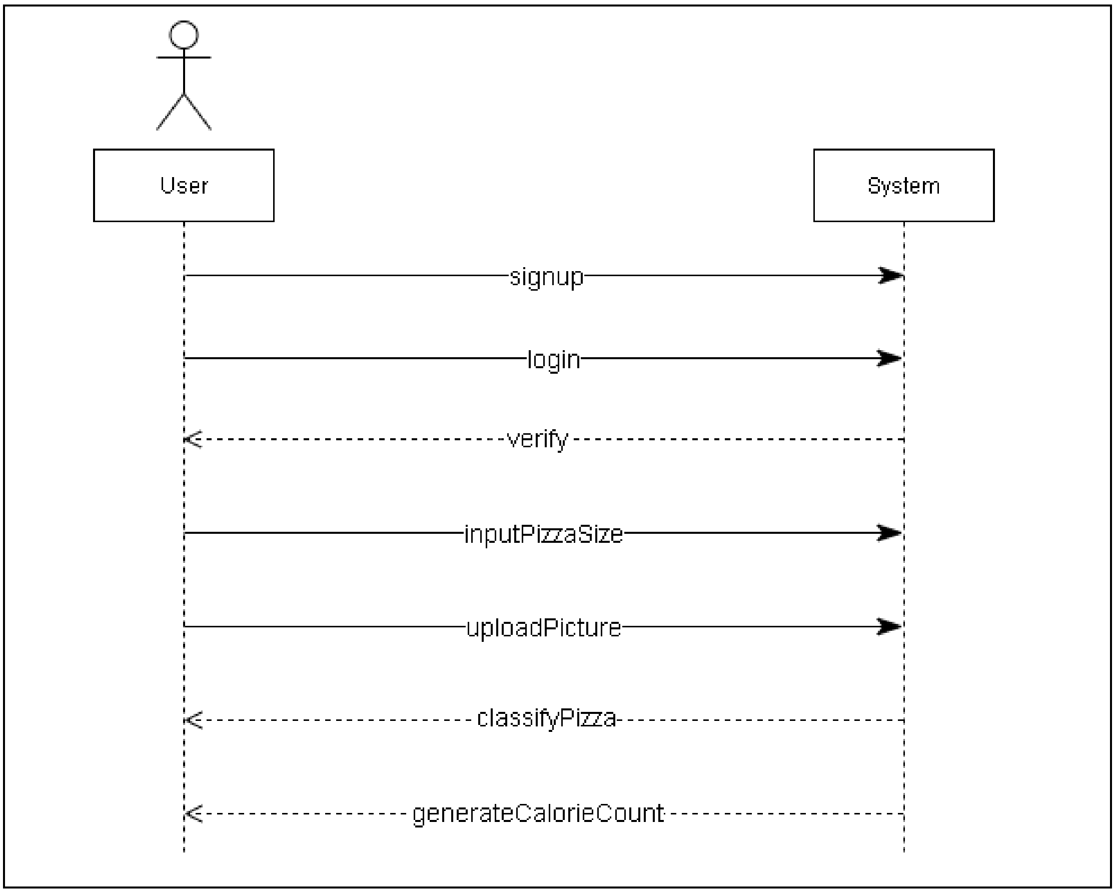


Figure 9: System Sequence Diagram of Snap&Eat

3.5.3 ENTITY RELATIONSHIP DIAGRAM (ERD)

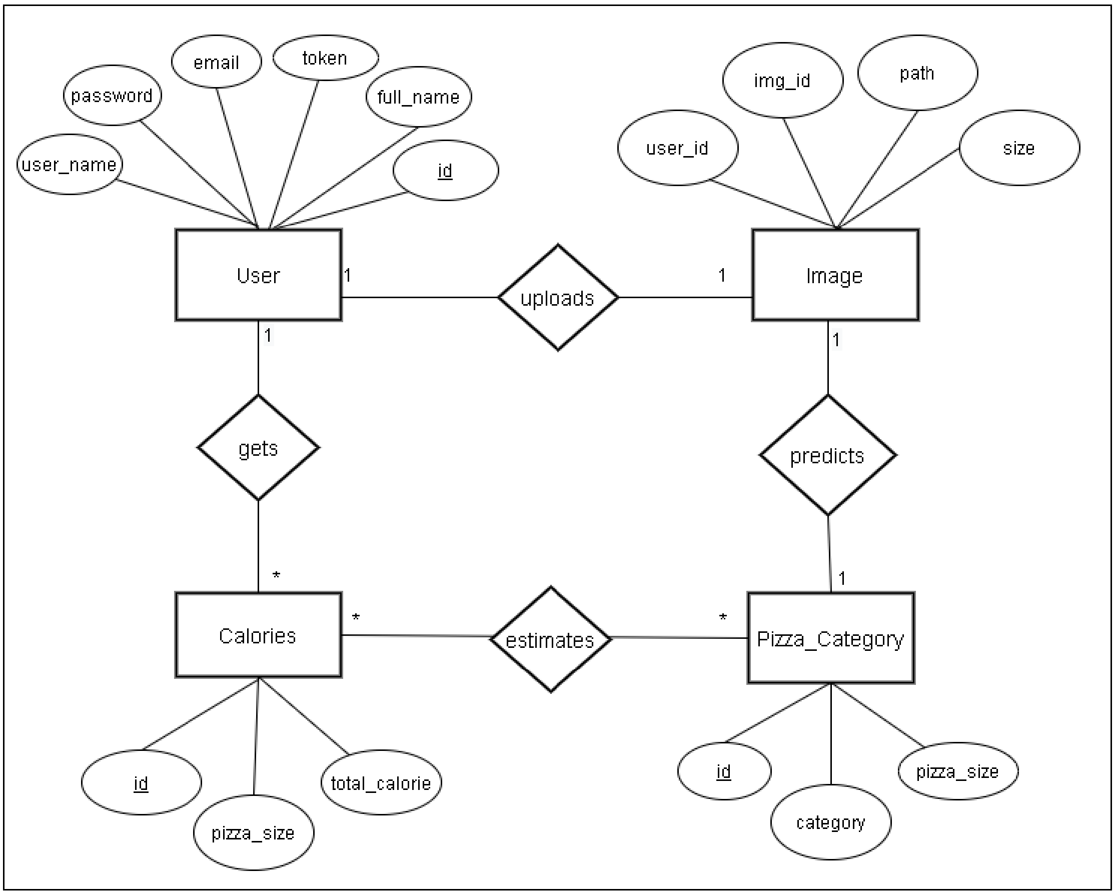


Figure 10: Entity Relationship Diagram of Snap&Eat

3.5.4 DOMAIN CLASS DIAGRAM (DCD)

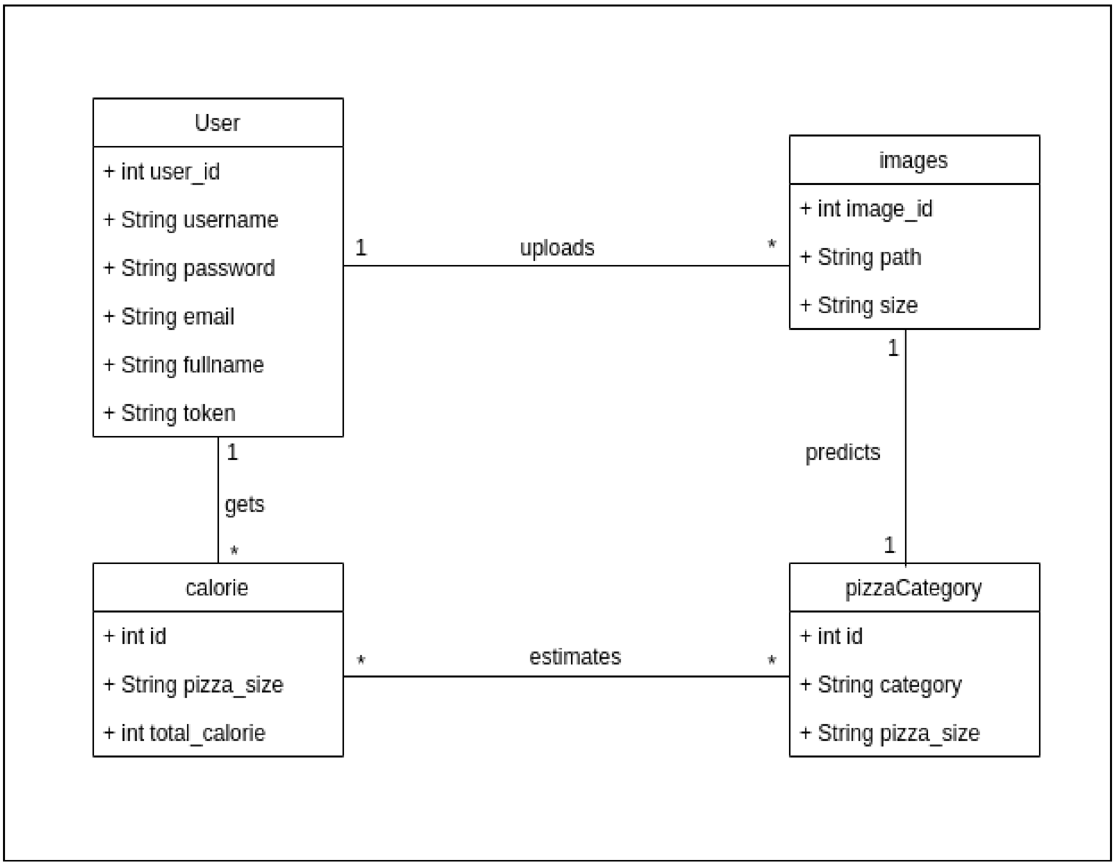


Figure 11: Domain Class Diagram of Snap&Eat

3.5.5 INTERACTION DIAGRAM

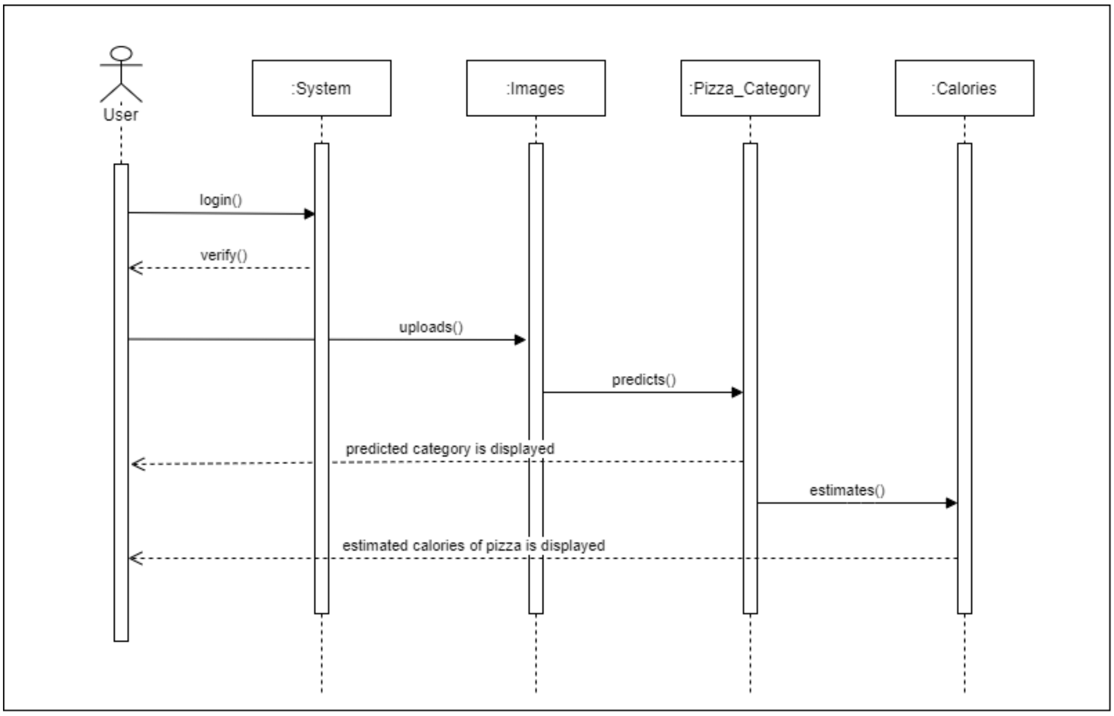


Figure 12: Interaction Diagram of Snap&Eat

Chapter 4
TESTING

Testing is required for evaluating the behavior of the system under different scenarios. By running tests against our app consistently, we can verify our app's correctness, functional behavior, and usability beforehand.

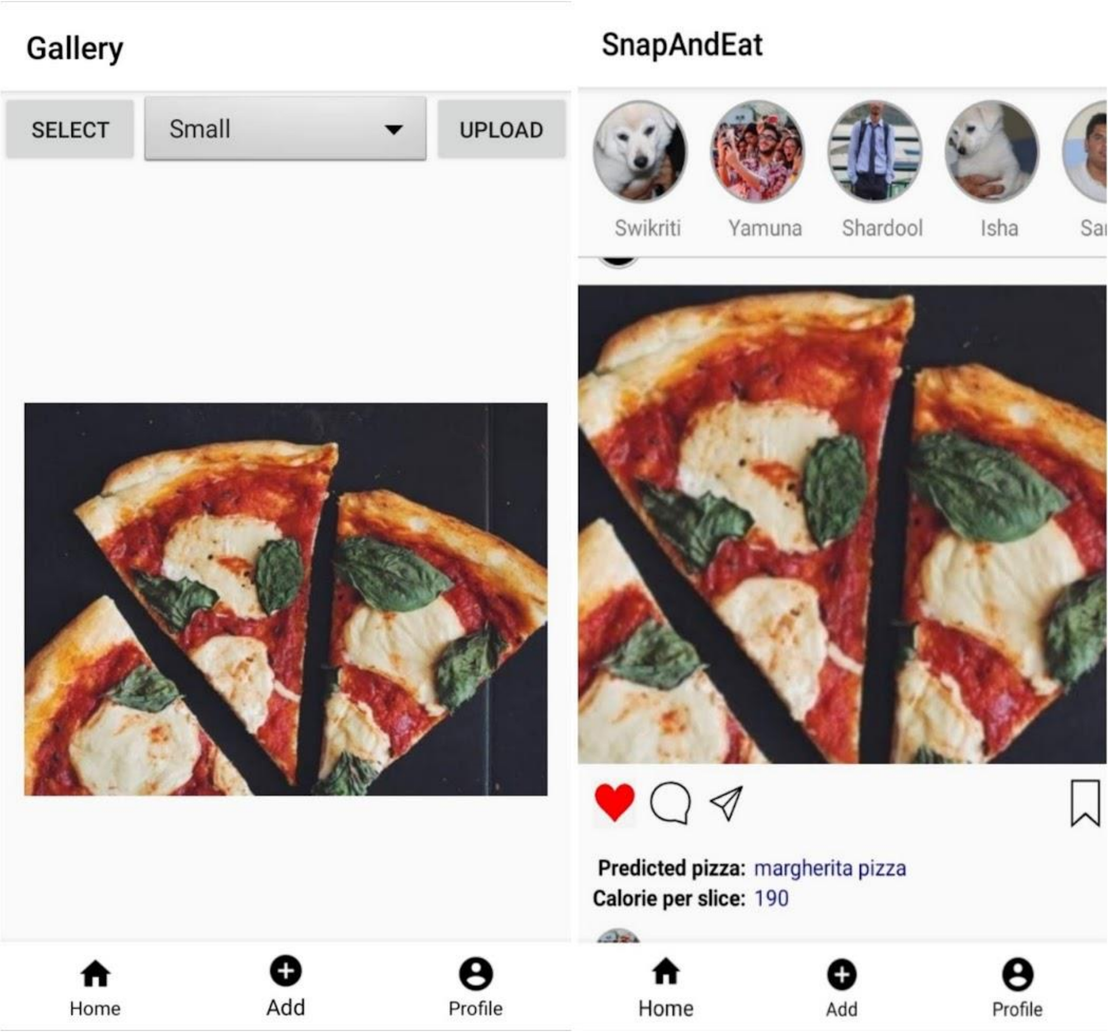
4.1 TEST OBJECTIVES

- To examine the accuracy of the system.
- To make sure that the result meets the specific requirements, needs, and expectations.
- To study the system’s behavior.

4.2 FUNCTIONAL TESTING

Functional testing is the type of testing done against the business requirements of the system. It is a type of black-box testing. It involves the complete integrated system to evaluate the system’s compliance with its specified requirements. This type of testing is to be carried out based on the functional specification document. In actual testing, testers need to verify a specific action or function of the code. The functional testing is carried out before the non-functional testing.

Mostly, this type of testing is performed on the user interface and call flows of the application. As we can see from the picture below, when the user clicks the picture of pizza and provides its size, our application determines the type of pizza and estimates the total calories present in it, which is exactly how is supposed to work. Thus it fulfills its requirements.

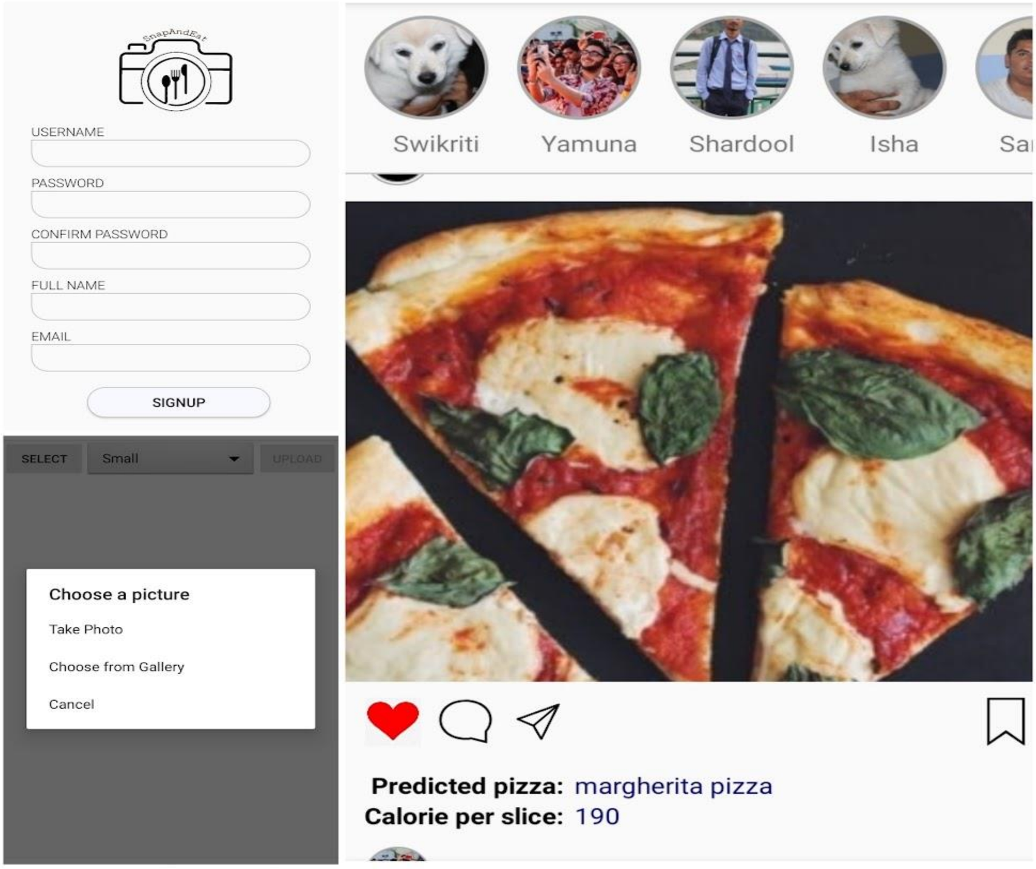


Predicted pizza: margherita pizza
Calorie per slice: 190

Figure 13: Functionality Testing

4.3 ANDROID UI TESTING

This is user-centric testing of the application. In this test phase, items such as visibility of text in various screens of the app, interactive messages, alignment of data, the look and feel of the app for different screens, size of fields, etc. are tested under this. As we can see from the picture below, all the pictures, fields, texts, buttons are perfectly aligned with the screen.

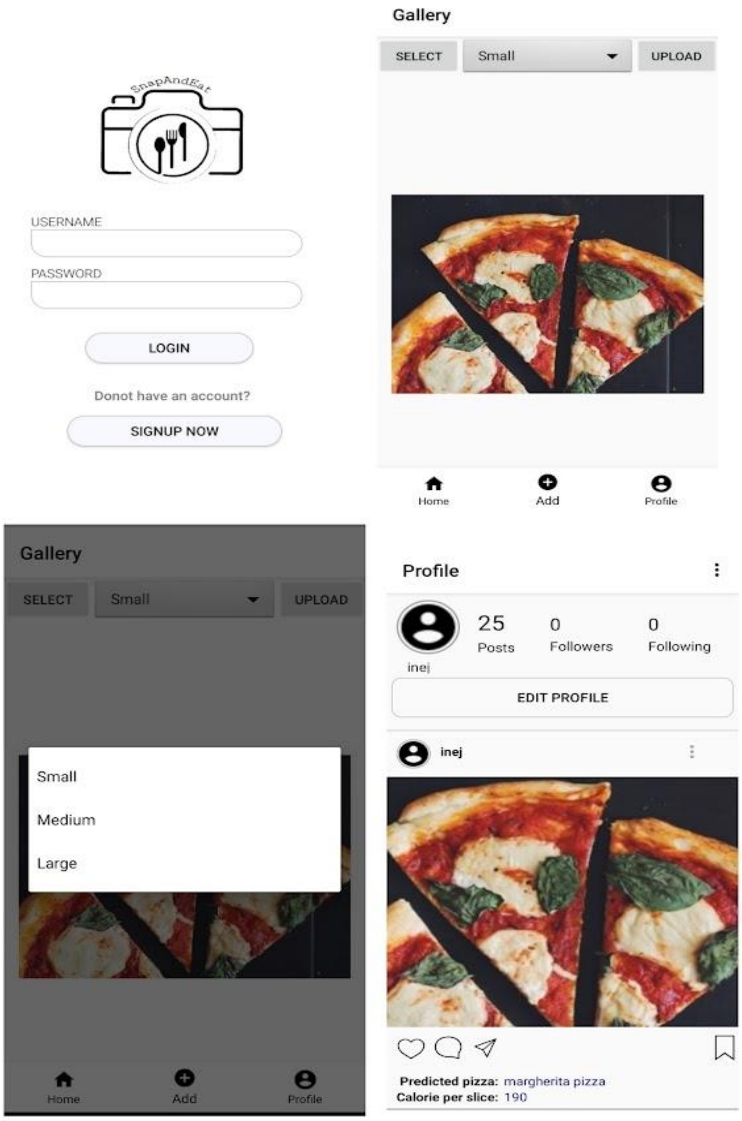


Predicted pizza: margherita pizza
Calorie per slice: 190

Figure 14: Data Alignment with the screen

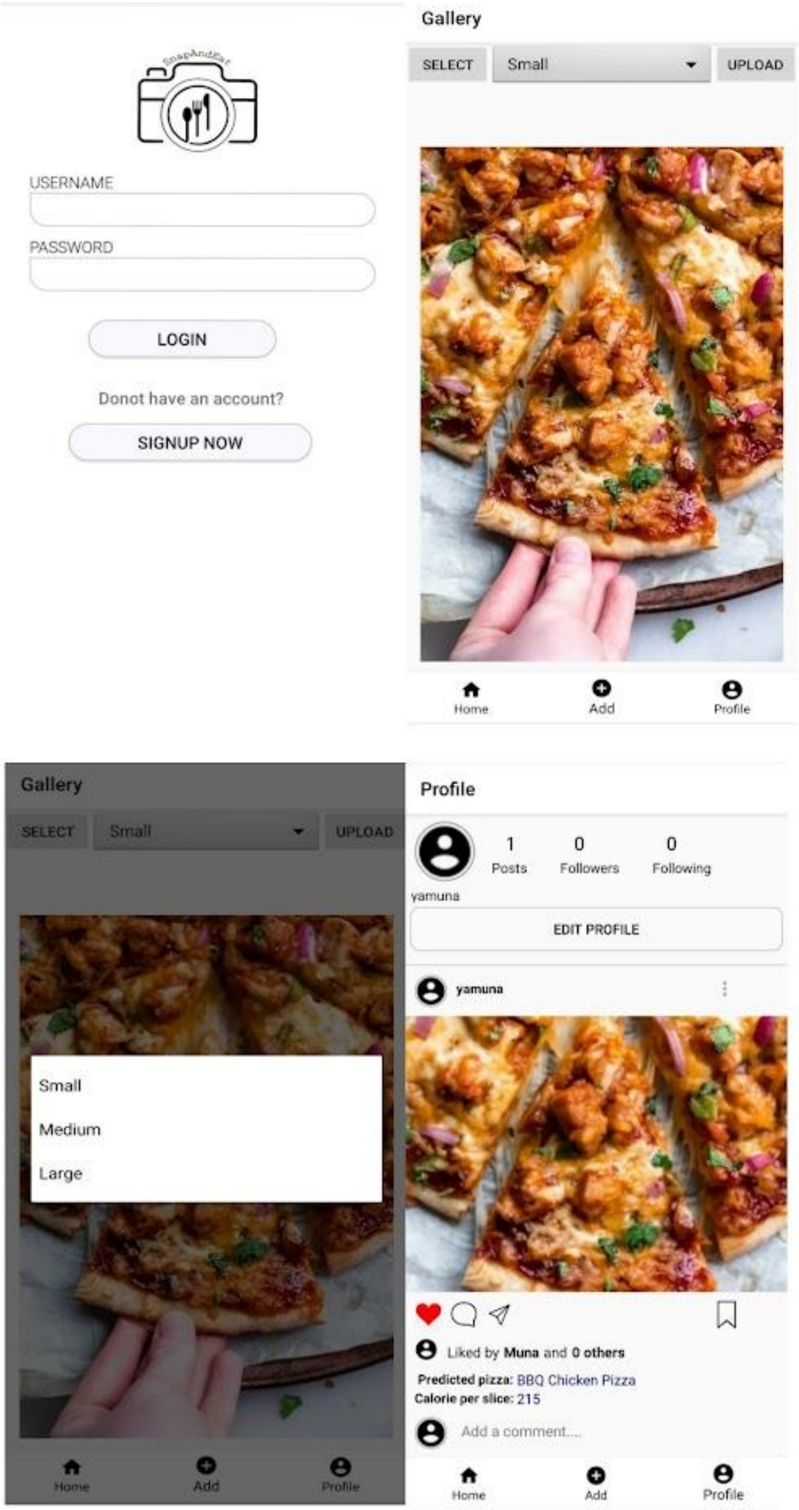
4.4 COMPATIBILITY TESTING

We tested our application across various versions of android devices with different API levels. And, we found that our application works fine across all the devices.



Predicted pizza: margherita pizza
Calorie per slice: 190

Figure 15: Testing on Android 9 (API 28)



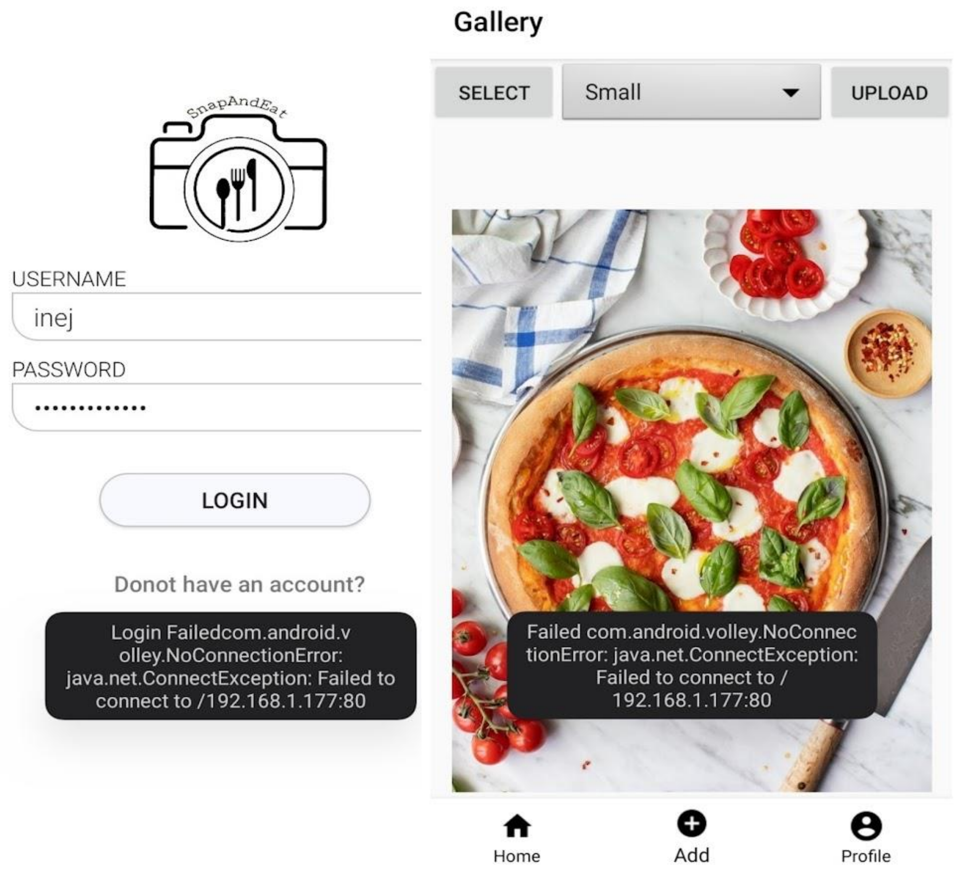
Liked by Muna and 0 others

Predicted pizza: BBQ Chicken Pizza
Calorie per slice: 215

Figure 16: Testing on Android 10(API 29)

4.5 NETWORK TESTING

Our application is an online application that requires an internet connection for its usage. So, the application will not work in case of no internet. Likewise, a message is displayed showing that the user is not connected to the internet.



USERNAME: inej

PASSWORD: ••••••••••••••

Login Failedcom.android.volley.NoConnectionError: java.net.ConnectException: Failed to connect to /192.168.1.177:80

Small

Failed com.android.volley.NoConnectionError: java.net.ConnectException: Failed to connect to / 192.168.1.177:80

Figure 17: Network Testing

4.6 AUTHENTICATION TESTING

The user needs to sign up or log in first to get access to the application. Similarly, certain criteria should be fulfilled in other to gain access, which is shown in the table below.

Table 1: Test Cases

TEST CASE ID	PURPOSE	TEST CASES	EXPECTED	OUTPUT	PASS/FAIL
TC1	Authentication	User name with special characters	Special characters are not allowed in username...	Special characters are not allowed in username...	Pass
TC2	Authentication	User name, first name, last name, left blank	Fill up all the fields properly	Fill up all the fields properly	Pass
TC3	Authentication	Password field left blank	Fill up all the fields properly	Fill up all the fields properly	Pass
TC4	Registration	Password and con-	Passwords do not match	Passwords do not match	Pass

		firm passwords are different			
TC5	Authentication	Password with length less than 8 characters	Password too short	Password too short	Pass
TC6	Authentication	Minimum 8 characters with a valid password	Password accepted	Password accepted	Pass
TC7	Network	Not connected to a network	Please check your internet connection	Please check your internet connection	Pass

Chapter 5
CONCLUSION

We were successful in developing a well-functioning application that meets our aforementioned objectives. Our project demanded a lot of working around with data and databases. Therefore, our first step was to collect datasets of different categories of pizza and train the model. Furthermore, it was our conscious effort in developing an attractive and interactive UI. It was a lot of fun working around the UI design and layout of the application. Likewise, we researched the different algorithms for image classification. We did invest some quality time in learning the CNN algorithm and implementing it. Finally, our application recognized the type of pizza using the CNN algorithm and calculated the total calories present in it. It was both challenging and exciting at the same time working with a newer concept like ours. Besides meeting its requirements, we put our application through a lot of tests and it passed all of them. It can be run efficiently on different versions of android. Thus, with hard work and determination, we were able to materialize our vision and ideas through our application.

BIBLIOGRAPHY

FatSecret. (2020). *Fatsecret*. Retrieved January 2020, from https://www.fatsecret.com/

FitWatch. (2021). Retrieved from Fit Watch: https://www.fitwatch.com/?fbclid=IwAR2uTnui8t77Us3ivf0YJSCjqxPi3z1aoAITVsFfMKb4N70361lyYypjDCE

Inc., A. (2017). *Calorie Mama*. Retrieved January 2020, from https://www.caloriemama.ai/

Instagram. (n.d.). Retrieved January 2020, from https://www.instagram.com/

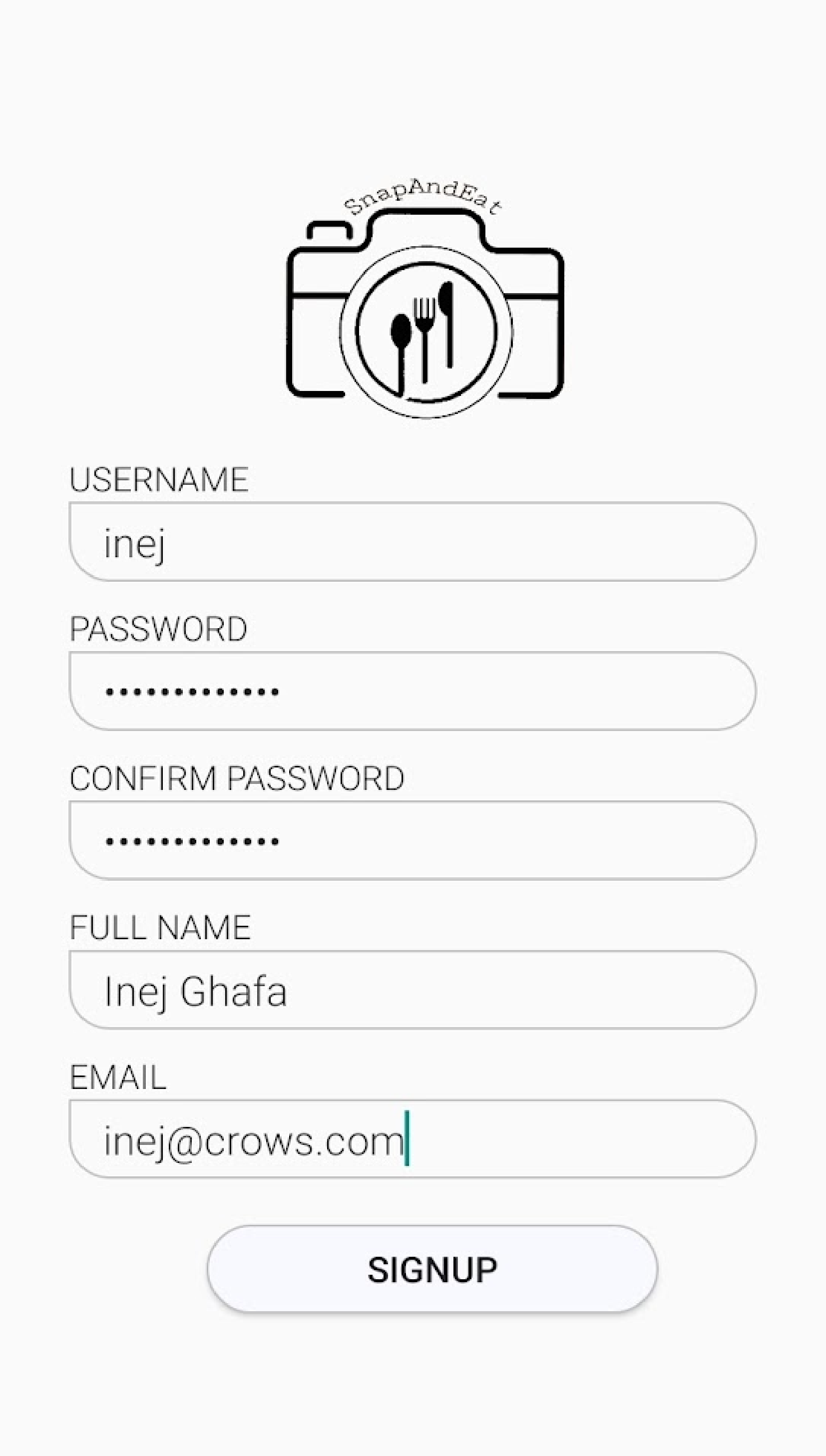
Little Caesers. (2020). Retrieved from Little Caesers Nutritional Guide: https://littlecaesars.ca/en-ca/our-menu/nutrition/?fbclid=IwAR1vpk-jo2WAmLKgIKcK4qTV5lVook9bESgCJMINtJx_Fp1AMsLuyppk93w

Manal Chokr, S. E. (2017). *Calories Prediction from Food Images*. Beirut, Lebanon: www.aaai.org.

Nutritionix. (2021, July). Retrieved from Nutritionix-a Syndigo Company:https://www.nutritionix.com/?fbclid=IwAR2XOCdRBOkB2CL-rZqFXBm9K_XWhUzVPNb2PtghNDacUvulZfWUO_1R_2U

Pinterest. (n.d.). Retrieved January 2020, from https://www.pinterest.com/redstaryeast/pizza/

APPENDICES



USERNAME

inej

PASSWORD

•••••••••••••

CONFIRM PASSWORD

•••••••••••••

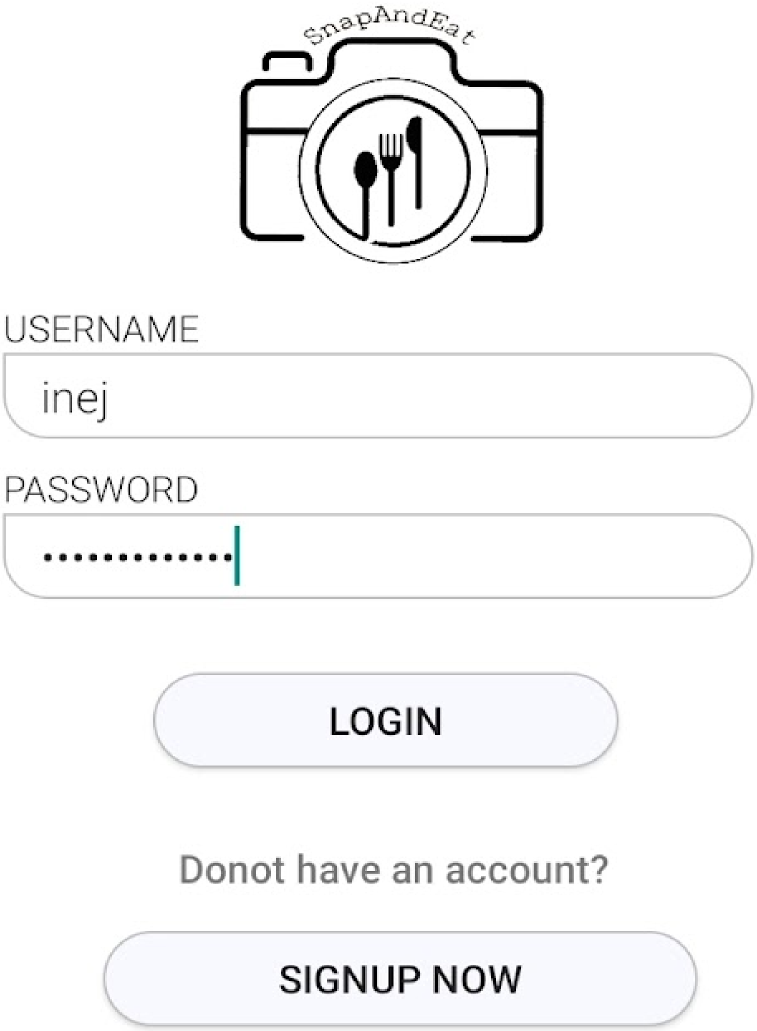
FULL NAME

Inej Ghafa

EMAIL

inej@crows.com

Figure 18: Signup page of Snap&Eat



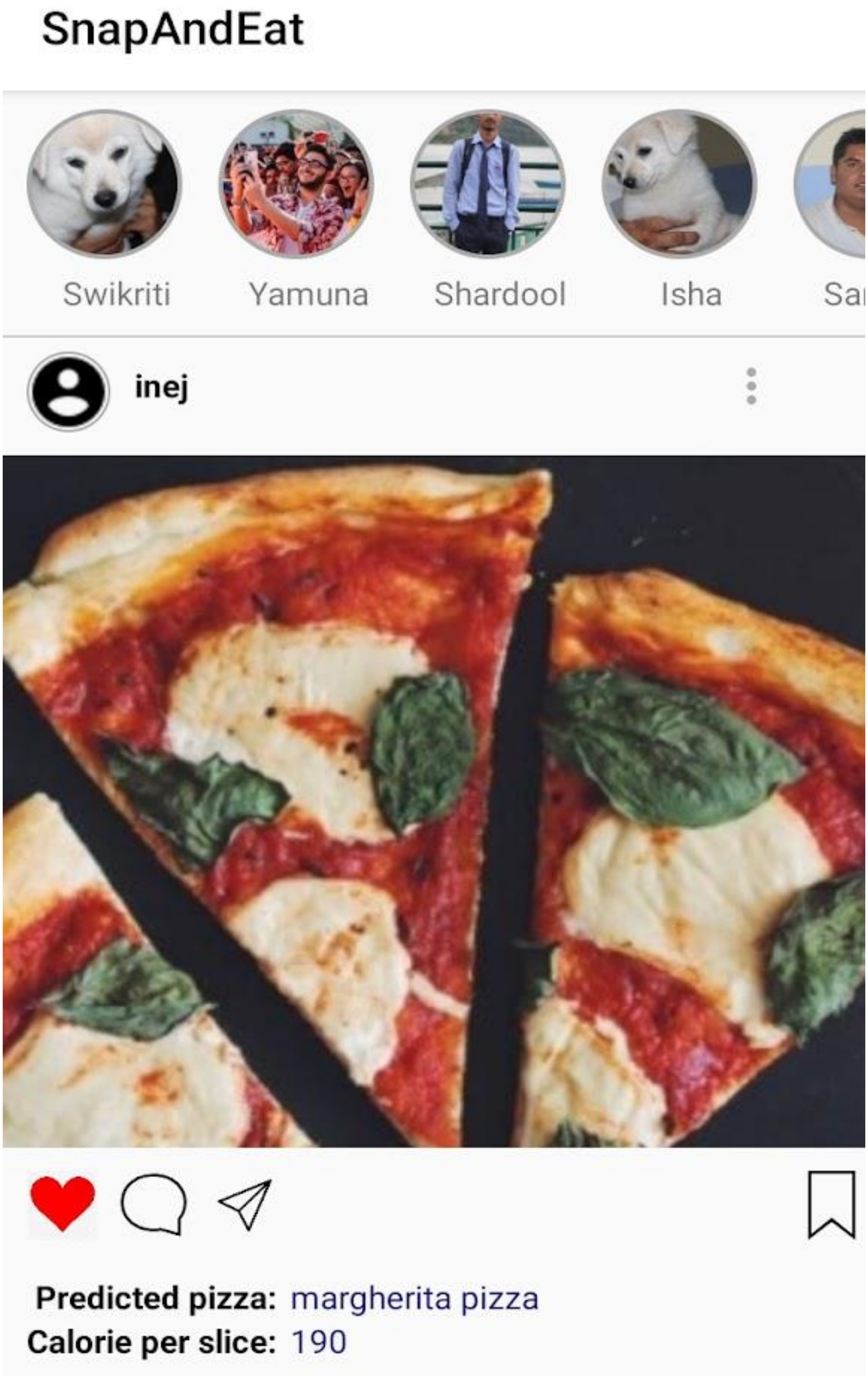
USERNAME

inej

PASSWORD

•••••••••••••

Figure 19: Login page of Snap&Eat

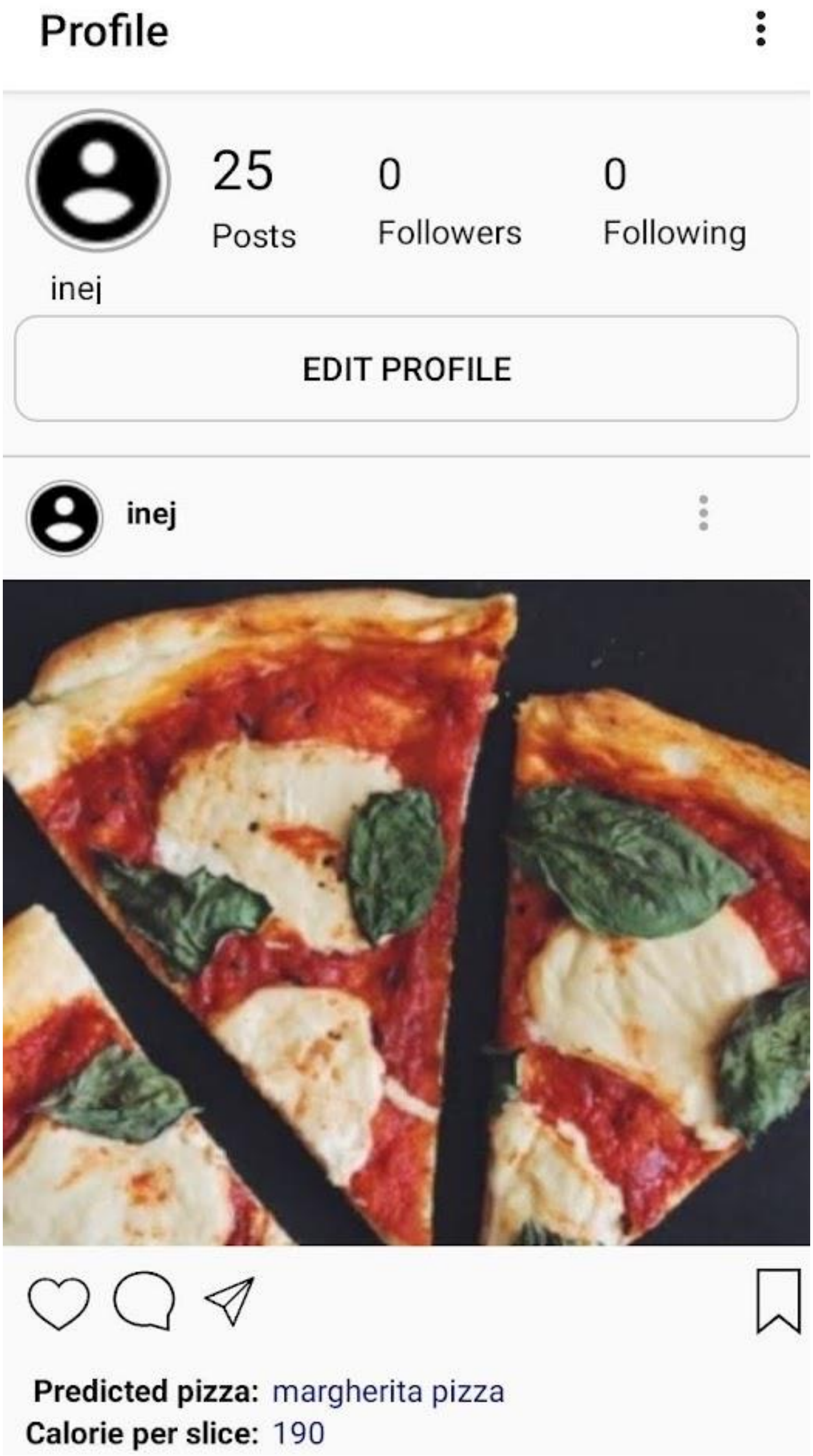


SnapAndEat

Predicted pizza: margherita pizza

Calorie per slice: 190

Figure 20: Feed page of Snap&Eat



Predicted pizza: margherita pizza
Calorie per slice: 190

Figure 21: Profile page of Snap&Eat

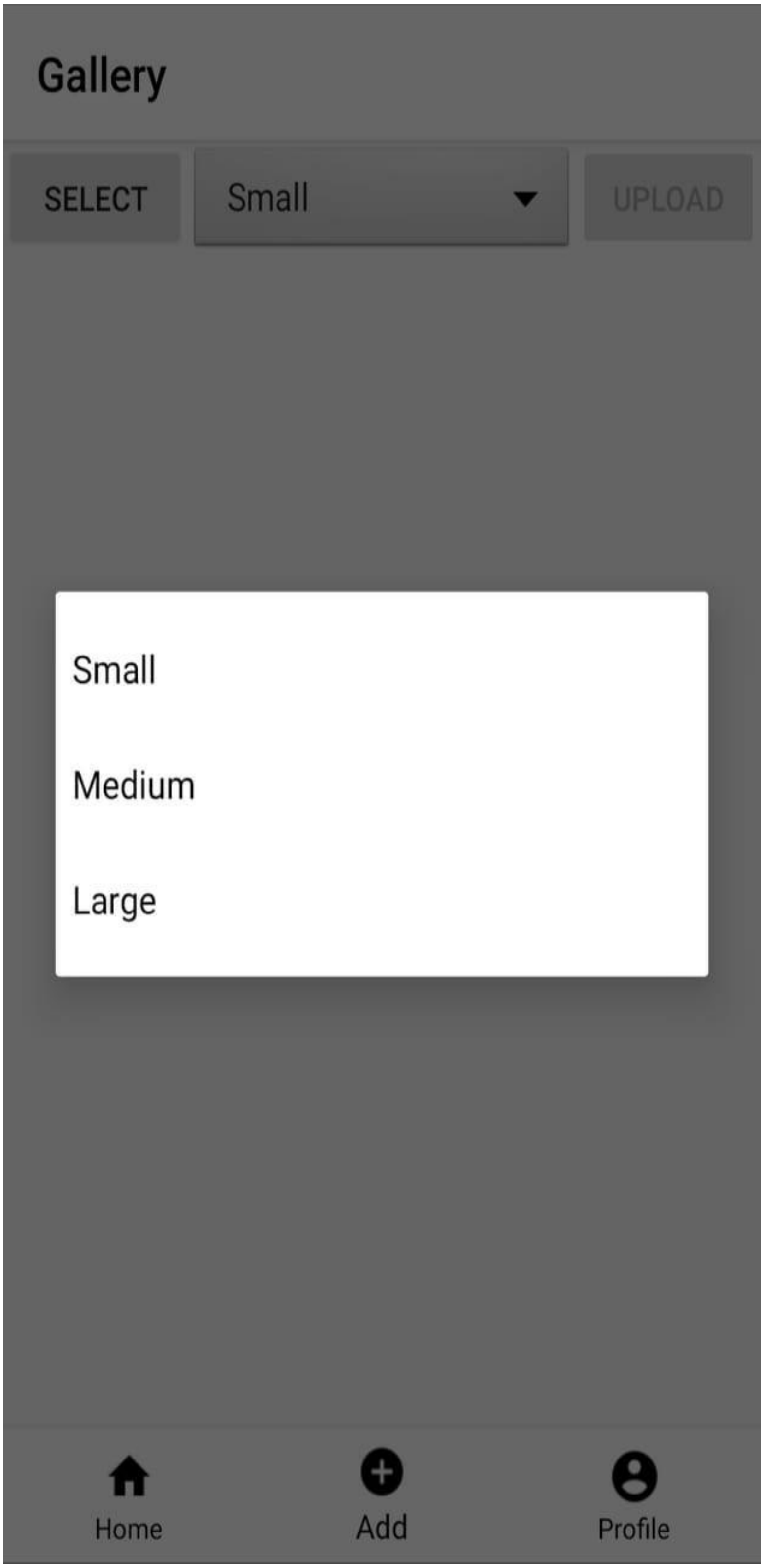
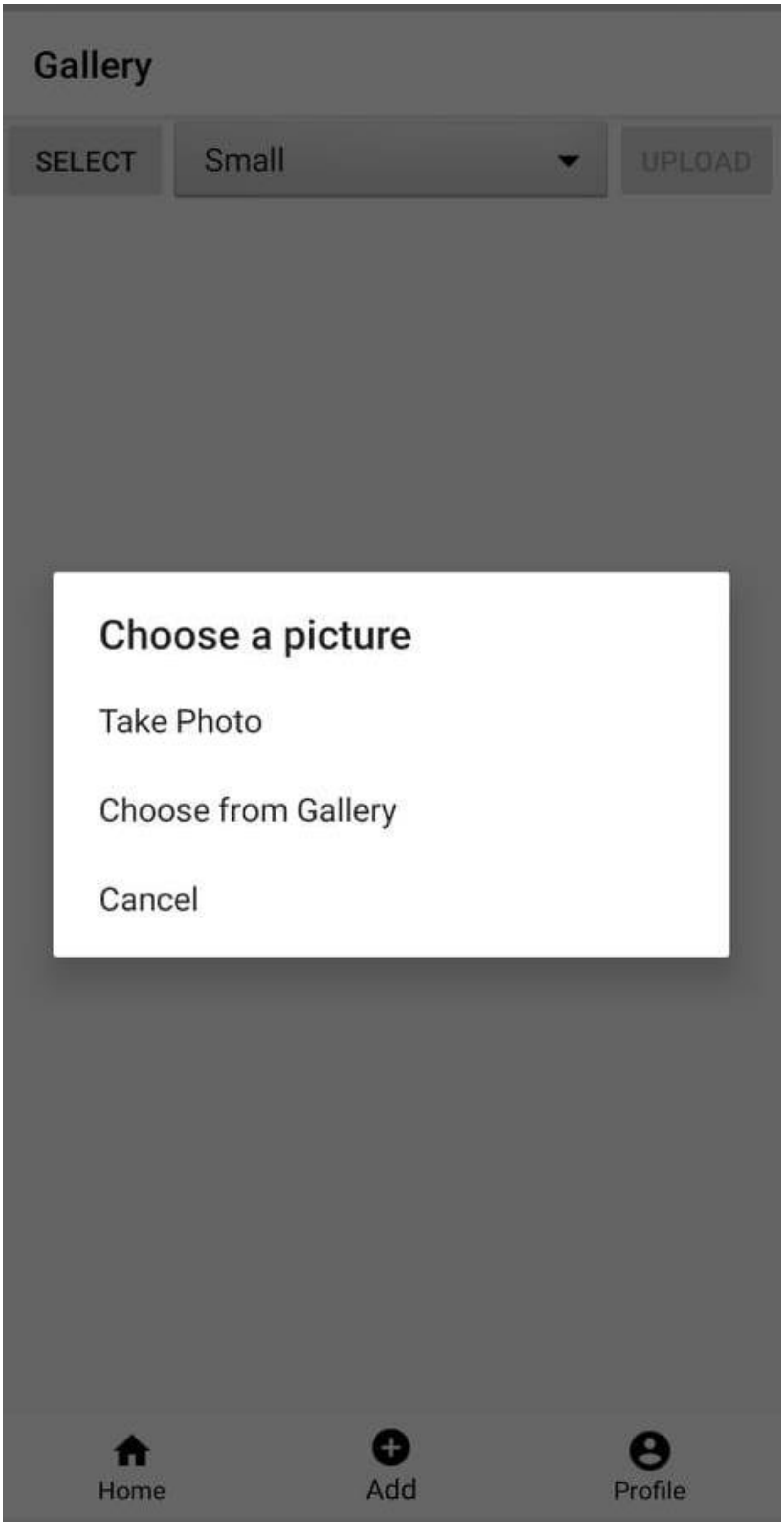


Figure 22: Select pizza size page of Snap&Eat



Choose a picture

Take Photo

Choose from Gallery

Cancel

Figure 23: Select picture page of Snap&Eat

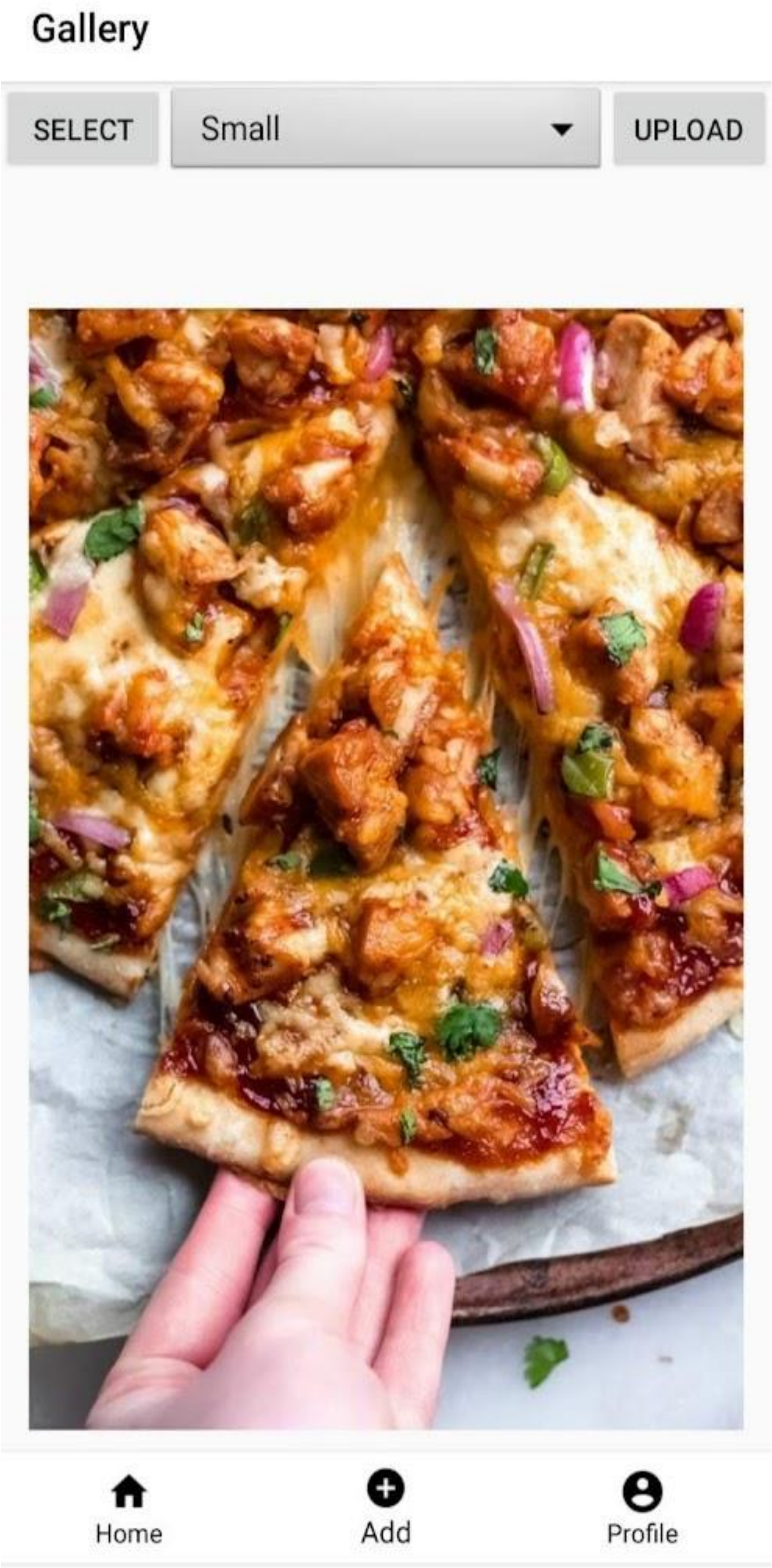


Figure 24: Upload pic page of Snap&Eat