A project report on

OBJECT DETECTION AND RECOGNITION (HELL-CHEF APP)

Submitted in partial fulfillment for the award of the degree of

M-Tech Integrated Software Engineering

By

V. SAI NIKHIL (16MIS0257)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

VELLORE INSTITUTE OF TECHNOLOGY

November 2020

OBJECT DETECTION AND RECOGNITION (HELL-CHEF APP)

Submitted in partial fulfillment for the award of the degree of

M-Tech Integrated Software Engineering

by

V. SAI NIKHIL (16MIS0257)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore Institute of Technology

November 2020

DECLARATION

I here by declare that the thesis entitled “OBJECT DETECTION AND RECOGNITION (HELL-CHEF APP)” submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bonafide work carried out by me under the supervision of Guide Name Prof. P. Prabhavathy.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 24 November 2020 Signature of the Candidate

V. SAI NIKHIL (16MIS0257)

CERTIFICATE

This is to certify that the thesis entitled “OBJECT DETECTION AND RECOGNITION (HELL-CHEF APP)” submitted by V. SAI NIKHIL (16MIS0257) Vellore Institute of Tecnology VIT, for the award of the degree of Name of the degree is a record of bonafide work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

P. Prabhavathy

Signature of the Guide **Signature of the Hod**

Prof. Prabhavathy p **Dr. Shantharajah S P**

Internal Examiner **External**
Examiner

Prof. Jagadeesh G

ABSTRACT

Object detection and recognition has been one of the challenging tasks for many years. They are many techniques involved in this process and they are used in many scenarios in the real time. Things like classroom attendance, offices, traffic, and photo tagging so on. With the help of the advanced AI technologies they are been used in Drones, self-driving cars, automation and in robots for understanding the environment. The visual aspect of classification the objects and recognition of what that object is the key aspect ratio in all these applications. Usually a traditional Convolution neural network has been used for the recognition of the objects.

In our project, we are applying CNN techniques to detect and recognize the fruits and vegetables that are available in the house and recommend the dish that can be prepared using the main key ingredients. We have many applications (app) for side chef, super cook where the recipes and other steps are present. However, we need to manually search for the dishes that matches with the ingredients we have in our house. Many of people doesn't know the ingredients names and it has been a challenge to many people even our parents. In our application, the user can take the pic of the items (vegetables and fruits) that are available in his refrigerator or in the house, using the Hell-chef app there is a smart camera feature for it. The captured image goes to the CNN and identify the items or ingredients and lists the recipes where the user can choose the dish they want to make. This algorithm used in the application provides accurate results and high accuracy.

We train the model using the custom dataset of fruits and vegetables. The ready to use trained weights are available in the GitHub repository of coco weights. However, they cannot be used for our model since we are using the custom dataset and train the network model. Since the dataset is huge, we use google drive to store the data.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to **Prof. Prabhavathy P**, Associate Professor, School of Information Technology and Engineering, Vellore Institute of Technology, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavour. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Deep learning.

I would like to express my gratitude **Dr. G. Viswanathan (Chancellor)**, **Shri Sankar Viswanathan (Vice President)**, Vice Chancellor **Dr. Rambabu Kodali**, Pro-Vice Chancellor **Dr. S. Narayanan**, and Dean **Prof. Balakrushna Tripathy**, to School of Information Technology and Engineering, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to **DR. SHANTHARAJAH S P**, **HOD** of Software and Systems Engineering, SITE, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: 24 November 2020 Name of the student

V. SAI NIKHIL (16MIS0257)

CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE SURVEY

CHAPTER 3

SYSTEM DESIGN

CHAPTER 4

UX AND UI DESIGN

CHAPTER 5

MODULE DESCRIPTION

CHAPTER 6

SYSTEM IMPLEMENTATION

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 OVERVIEW

Object detection and recognition has been one of the challenging tasks for many years. They are many techniques involved in this process and they are used in many scenarios in the real time. Things like classroom attendance, offices, traffic, and photo tagging so on. With the help of the advanced AI technologies they are been used in Drones, self-driving cars, automation and in robots for understanding the environment. The visual aspect of classification the objects and recognition of what that object is the key aspect ratio in all these applications. Usually a traditional Convolution neural network has been used for the recognition of the objects.

In our project, we are applying CNN techniques to detect and recognize the fruits and vegetables that are available in the house and recommend the dish that can be prepared using the main key ingredients. We have many applications (app) for side chef, super cook where the recipes and other steps are present. However, the meals are present where the user select the meal then look for the ingredients. In our application the user can take the pic of the items that are available in his refrigerator or in the house, where it goes to the CNN and identify the items or ingredients and lists the recipes where the user can choose the dish they want to make. This algorithm used in the application provides accurate results and high accuracy.

1.2 OVERVIEW OF HELL-CHEF

Hell-chef is an android application which are similar to other cooking app with all the functionality of step by step cooking with ingredients and also have the functionality of inbuilt shopping list functionality so that the cooking process happen in a single app.

This application also have a unique feature of smart camera functionality where the user can take the picture of the ingredients that are available in their house. The machine-learning algorithm analyze the image taken and provide the recipes that can be cooked using this application. This machine learning is a custom-trained YOLO network, which identifies vegetables and label them. Later using the labels the recipe are generated and the recipes are displayed where user can select the recipe that they want to cook and enjoy the meal.

The database used in the application is firebase database. We can use the default system database, NoSQL database. However, we want the data to be secure and connected to internet. The firebase database is secure and make the unique database for unique customers.

1.3 CHALLENGES PRESENT IN HELL-CHEF

Thera are many challenges present in the hell-chef application. We took the baseline for the challenges for the hell-chef application to solve them and make that application. The challenges were

- Huge competition, similar applications available
- Identifying the appropriate machine learning model for the application
- Identifying the user interface of the application for the customer to understand and use them clearly.
- Training the custom dataset
- Integrating with the application
- Deployment the application to play store

Here the deployment of the application was not achieved because of the app compatibility and the application size. To overcome the competition, we introduced the new feature like ML integration and smart camera feature for smart recipe generation. The app is user friendly and this is achieved by doing a user experience on the user to get the feedback for the application design. The custom training of YOLO is achieved and is available in my GITHUB repository for the other people to use. Since the training of these datas happen for days.

1.4 PROJECT STATEMENT

One of the main challenges in cooking application is that many of the people does not use them since they do not understand the ingredients name in English. I personally felt in the lockdown with my mom cooking food and when we use the application we does not know the posh big ingredients names. We need to google the names to convert them into native form for understanding. Another drawback with applications is that many recipes have many items needed to be bought. Like many of the ingredients are not available right in the house. To solve these issues I created a mobile application that functions as a usual cooking app with the list of recipes available and added the additional feature of shopping cart and smart camera feature where we can cook the recipes from the ingredients we have in the house.

We need not be a pro to know the items, since the app identifies the vegetables and recommend the recipes to us with step by step. Later if we are missing the items that are required for the cooking, the ingredients are marked so that they can be easily added to the cart for shopping them.

1.5 OBJECTIVE

Our objective is to make a user-friendly application that has the machine learning functionality that makes the user life easier. To have all the functionality of the cooking application with few additional f\functionality like secure databases, smart camera, smart recipe feature and shopping list feature.

Chapter 2

Literature Survey

2.1 INTRODUCTION

2.1.1 CNNS FOR FACE DETECTION AND RECOGNITION

Two Stream CNN: In the first part, we detect a single human face from the rest of the image. The output will have the human face coordinate and size of the bounding box as well as the class of it. This module has 6 primary modules, each has one convolution layer, one max-pooling and one leaky ReLU layer. The last layer is connected to fully connected layer one for predicting the location and size.

Cascade CNN: In the above model, we cannot detect multiple faces. We perform sliding window across whole image. Each window is fed into convolution layer for binary classification. second stage, non-maximum suppression(NMS) is used to eliminate highly overlapped detection region. This help to tackle multiple face detection and classification.

2.1.2 ORIGINAL APPROACH FOR THE LOCALIZATION OF OBJECTS IN IMAGES

Sliding window: It detect the present and absent of human face in the image-sliding window. This was CNN based object detection algorithm. However, the drawback of this sliding window is that we need to repeat the process for period of time, which resulting in more computing power.

2.1.3 MEASURING THE OBJECT NESS OF IMAGE WINDOWS

Regional proposal method: They take the potential regions for high possible object detection. This number of regions is reduced than the sliding window. This method is also called as R-CNN. However, the time taken to train the model is slow and also consume much memory.

Regions of Interest (RoI) using the Region Proposal method on the input image, warps each RoI into standard in put size for the neural network and forward them into the CNNs dedicated for image classification and localization and output the class category as well as the bounding box coordinates and sizes.

2.1.4 YOU ONLY LOOK ONCE: UNIFIED, REAL-TIME OBJECT DETECTION

YOLO and SSD: This method does not use the regions to scan for objects. The input image directly goes through the convolution neural network, and they are divided into multiple grids and perform classification score on each grid. They reduce the training and testing time, but the accuracy is not achieved completely.

2.1.5 REAL-TIME CUSTOM OBJECT DETECTION AND RECOGNITION PROJECT

The motivation behind this project were namely, Recognition and detection of objects from a video has been one of the blooming topics. Real time recognition and detection is bounded by many constraints like background, multiple objects, higher FPS etc.

Input will be a video of common objects (COCO) that the YOLO is trained with and output will be the video with bounding boxes and labels around the objects.

Input will be the one from webcam, or a video of an object captured using webcam by drawing bounding boxes, since the network YOLO has been trained with that in the process therefore, output will be the video with confidence labels.

2.1.6 SHORT TERM TRAFFIC PREDICTION

Long Short-Term Memory arrange (LSTM) toward catch of worldly conditions consist of circulation moment. In any case, these time-arrangement expectation strategies are just performed on a solitary street section. At the point when they are applied to the entire street arrange, the computational proficiency will be diminished from numerous forecast undertakings. A few works discovered that learning different errands mutually can improve the forecast execution contrasted and learning them exclusively. Mama et al. utilized Convolutional Neural System (CNN) to separate the longitudinal conditions having rush hour gridlock stream yet the street systems concentrated in these exploration remained communicated to Euclidean cosmos (e.g., 2D pictures) The nearby spatial relationships adapted in this way as it were reflect unstructured spatial closeness, as opposed to organized contiguousness relations of street fragments in et al. proposed a chart convolutional neural system per information single-minded chart channel (GCNN-DDGF) that taken over to learn covered up heterogeneous pairwise spatial relationships. In any case, these charts were set undirected and consequently can't depict the coordinated spread of traffic stream along street systems. Toward the end, Li proposed the dispersion convolution activity happening guided charts near catch the three-dimensional connections of commuter traffic stream, however it has dissemination procedure stood just characterized by a period progress framework on the degree of separable hubs and boundaries. Consequently, the 3-D data got was dissemination difficulty just echoes low-request availability examples having street portions.

2.2 INFERENCE

Two Stream CNN we detect a single human face from the rest of the image. The output will have the human face coordinate and size of the bounding box as well as the class of it. This module has 6 primary modules, each has one convolution layer, one max-pooling and one leaky ReLU layer. The last layer is connected to fully connected layer one for predicting the location and size.

We cannot detect multiple faces. We perform sliding window across whole image. Each window is fed into convolution layer for binary classification. Second stage, non-maximum suppression (NMS) is used to eliminate highly overlapped detection region. This help to tackle multiple face detection and classification it happens in cascade CNN.

The dataset images or videos will be given as input to YOLO network that has been trained. The videos for input will be taken from the /videos folder in the project directory. The output video with the bounding boxes with labels will be stored in the folder /output in the project directory.

In out method of detection and recognition of human face, we use a new method which takes less time and indeed achieves better accuracy.

S. NO	TITLE	AUTHOR	TECHNIQUES USED	REMARKS
1	CNNs for Face Detection and Recognition	Yicheng An, Jiafu Wu, Chang Yue	Two Stream CNN: In the first part, we detect a single human face from the rest of the image. Cascade CNN: In the above model, we cannot detect multiple faces. We perform sliding window across whole image	Two Stream CNN we detect a single human face from the rest of the image. The output will have the human face coordinate and size of the bounding box as well as the class of it. This module has 6 primary modules, each has one convolution layer, one max-pooling and one leaky ReLU layer. The last layer is connected to fully connected layer one for predicting the location and size.
2	Original approach for the localization of objects in images	R. Vaillant, C. Monrocq, and Y. Le Cun	Sliding window: It detect the present and absent of human face in the image-sliding window. This was CNN based object detection algorithm.	Sliding window is that we need to repeat the process for period of time, which resulting in more computing power.

3	Measuring the object ness of image windows	Joseph Redmon, Santosh Divvala	Regional proposal method: They take the potential regions for high possible object detection. This number of regions is reduced than the sliding window. This method is also called as R-CNN. However, the time taken to train the model is slow and also consume much memory.	RoI into standard in put size for the neural network and forward them into the CNNs dedicated for image classification and localization and output the class category as well as the bounding box coordinates and sizes.
4	You Only Look Once: Unified, Real-Time Object Detection	Y. Wong, S. Chen, S. Mau, C. Sanderson	YOLO and SSD: This method does not use the regions to scan for objects. The input image directly goes through the convolution neural network, and they are divided into multiple grids	They reduce the training and testing time, but the accuracy is not achieved completely.

			and perform classification score on each grid.	
5	Real-time Custom Object Detection and Recognition Project	Alexe et al	The motivation behind this project were namely, Recognition and detection of objects from a video has been one of the blooming topics. Real time recognition and detection is bounded by many constraints like background, multiple objects, higher FPS etc. Input will be a video of common objects (COCO) that the YOLO is trained with and output will be the video with bounding boxes and labels	Input will be the one from webcam, or a video of an object captured using webcam by drawing bounding boxes, since the network YOLO has been trained with that in the process therefore, output will be the video with confidence labels.

			around the objects.	
6	Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning Marco Lippi,	Matteo Bertini, and Paolo Frasconi	ARIMA model coupled with a kalman filter is the most accurate model	Computational complexity issues largest training time

Table 1.1 Summary of Literature Survey

Chapter 3

System design

3.1 INTRODUCTION

Object detection and recognition has been one of the challenging tasks for many years. They are many techniques involved in this process and they are used in many scenarios in the real time. Things like classroom attendance, offices, traffic, and photo tagging so on. With the help of the advanced AI technologies they are been used in Drones, self-driving cars, automation and in robots for understanding the environment. The visual aspect of classification the objects and recognition of what that object is the key aspect ratio in all these applications. Usually a traditional Convolution neural network has been used for the recognition of the objects.

In our project, we are applying CNN techniques to detect and recognize the fruits and vegetables that are available in the house and recommend the dish that can be prepared using the main key ingredients. We have many applications (app) for side chef, super cook where the recipes and other steps are present. However, we need to manually search for the dishes that matches with the ingredients we have in our house. Many of people doesn't know the ingredients names and it has been a challenge to many people even our parents. In our application, the user can take the pic of the items (vegetables and fruits) that are available in his refrigerator or in the house, using the Hell-chef app there is a smart camera feature for it. The captured image goes to the CNN and identify the items or ingredients and lists the recipes where the user can choose the dish they want to make. This algorithm used in the application provides accurate results and high accuracy.

We train the model using the custom dataset of fruits and vegetables. The ready to use trained weights are available in the GitHub repository of coco weights. However, they cannot be used for our model since we are using the custom dataset and train the network model. Since the dataset is huge, we use google drive to store the data.

3.2 YOLO NETWORK ARCHITECTURE

You only look once (YOLO) is one of the faster object detection algorithms out there. Though it is no longer the most accurate object detection algorithm, it is a very good choice when you need real-time detection, without loss of too much accuracy.

We are using YOLO v3 The first detection is made by the 82nd layer. For the first 81 layers, the image is down sampled by the network, such that the 81st layer has a stride of 32. If we have an image of 416 x 416, the resultant feature map would be of size 13 x 13. One detection is made here using the 1 x 1 detection kernel, giving us a detection feature map of 13 x 13 x 255.

Then, the feature map from layer 79 is subjected to a few convolutional layers before being up sampled by 2x to dimensions of 26 x 26. This feature map is then depth concatenated with the feature map from layer 61. Then the combined feature maps is again subjected a few 1 x 1 convolutional layers to fuse the features from the earlier layer (61). Then, the second detection is made by the 94th layer, yielding a detection feature map of 26 x 26 x 255.

A similar procedure is followed again, where the feature map from layer 91 is subjected to few convolutional layers before being depth concatenated with a feature map from layer 36. Like before, a few 1 x 1 convolutional layers follow to fuse the information from the previous layer (36). We make the final of the 3 at 106th layer, yielding feature map of size 52 x 52 x 255.

Softmaxing classes rests on the assumption that classes are mutually exclusive, or in simple words, if an object belongs to one class, then it cannot belong to the other. This works fine in COCO dataset.

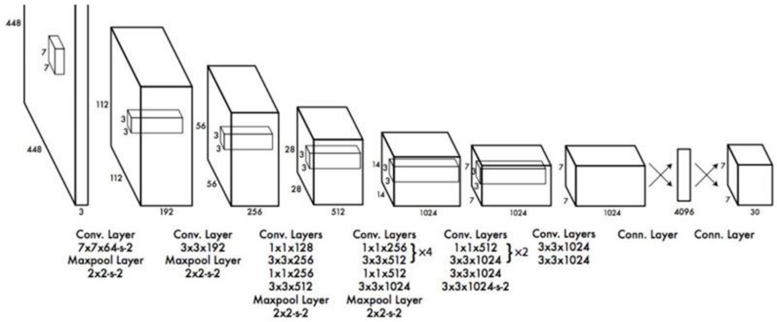


Figure 1.1 YOLO Architecture

3.3 IMPOROVED RESNET

The feedforward network with a single layer is sufficient to represent any function. However, the layer might be massive and the network is prone to overfitting the data. Therefore, there is a common trend in the research community that our network architecture needs to go deeper.

Since AlexNet, the state-of-the-art CNN architecture is going deeper and deeper. While AlexNet had only 5 convolutional layers, the VGG network and GoogleNet (also codenamed Inception_v1) had 19 and 22 layers respectively.

However, increasing network depth does not work by simply stacking layers together. Deep networks are hard to train because of the notorious vanishing gradient problem as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitively small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly.

This may look familiar to you as it is very similar to the Inception module of , they both follow the split-transform-merge paradigm, except in this variant, the outputs of different paths are merged by adding them together, while in [4] they are depth-concatenated. Another difference is that in , each path is different (1x1, 3x3 and 5x5 convolution) from each other, while in this architecture, all paths share the same topology.

The authors introduced a hyper-parameter called cardinality — the number of independent paths, to provide a new way of adjusting the model capacity. Experiments show that accuracy can be gained more efficiently by increasing the cardinality than by going deeper or wider. The authors state that compared to Inception, this novel architecture is easier to adapt to new datasets/tasks, as it has a simple paradigm and only one hyper-parameter to be adjusted, while Inception has many hyper-parameters (like the kernel size of the convolutional layer of each path) to tune.

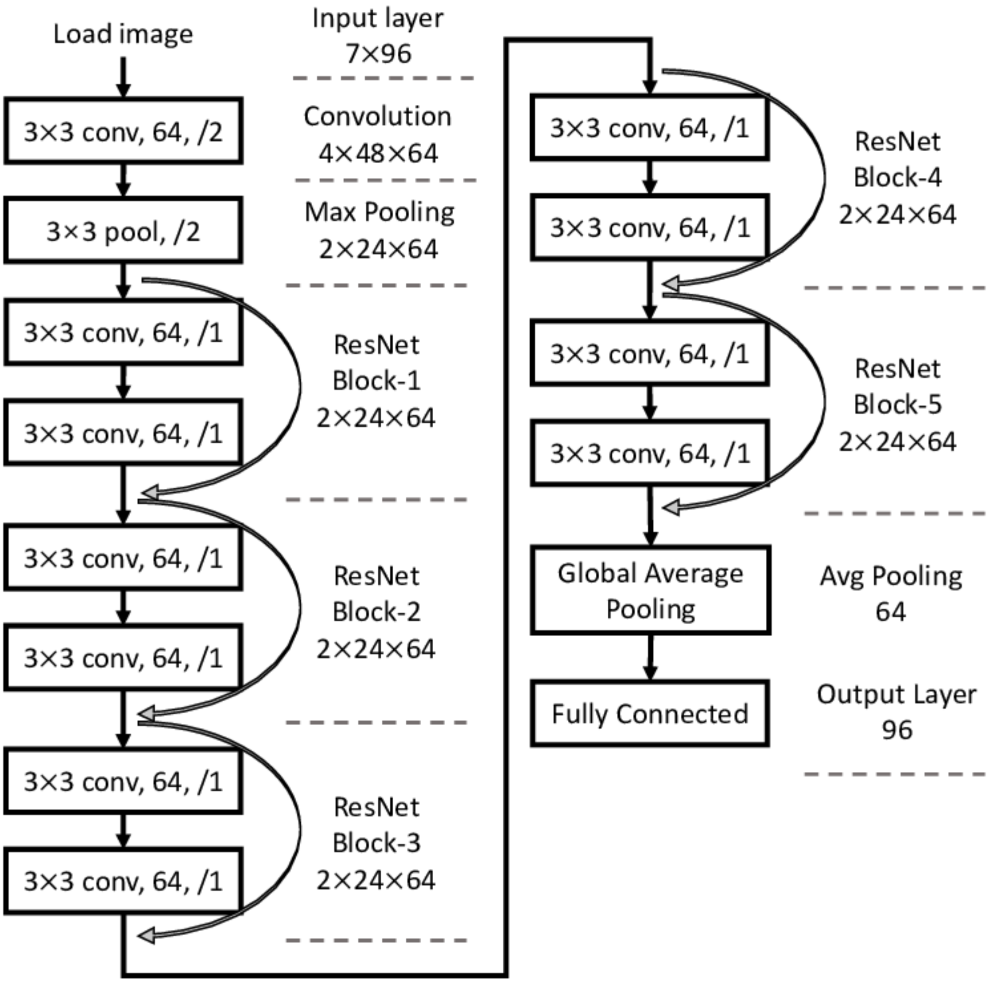


Figure 1.2 RESNET General Architecture

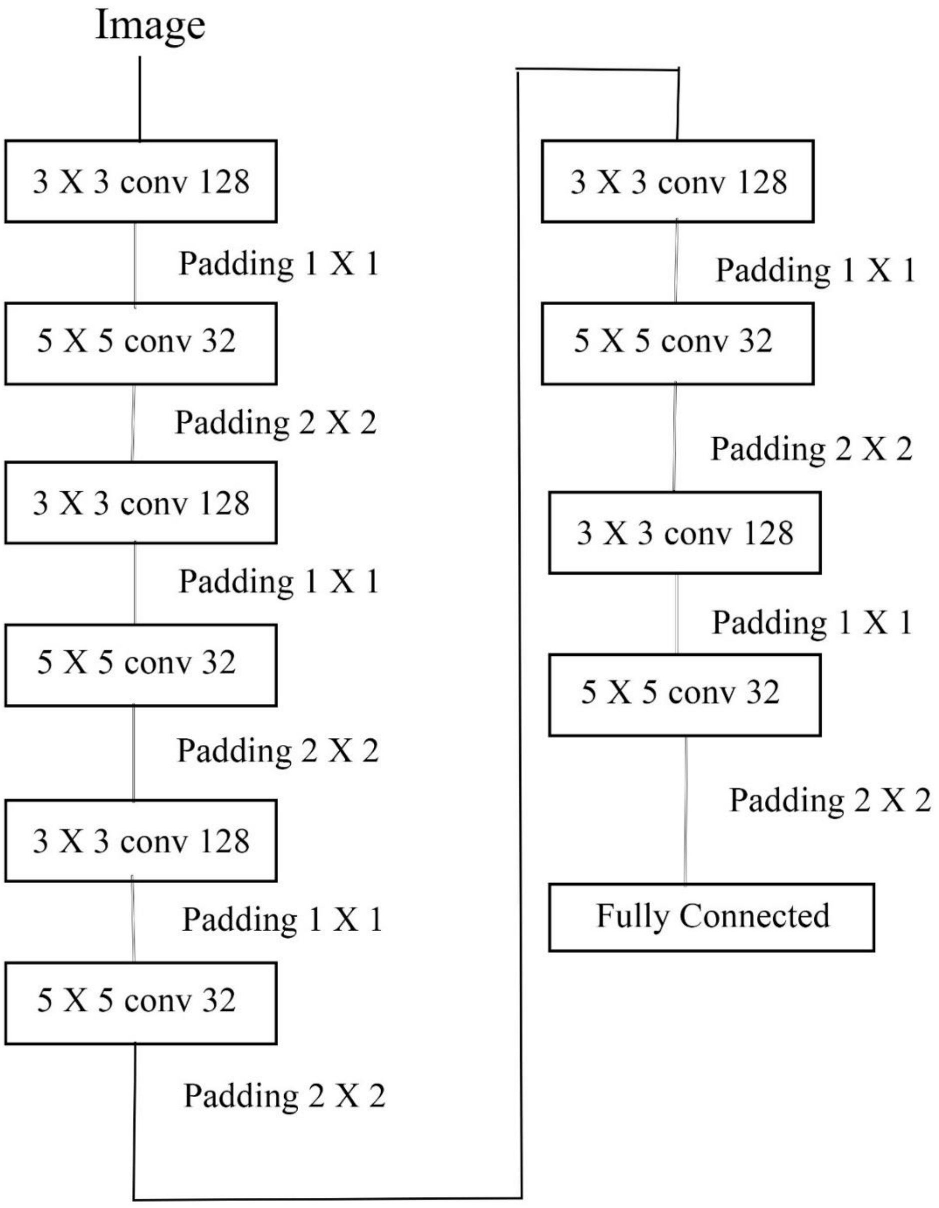


Figure 1.3 Improvised RESNET

3.4 APPLICATION ARCHITECTURE

The Hell-chef application has a series of architecture combined to form a complete model. At first, we have a general application that perform the cooking app task. Then we have the smart camera feature where the smart activity takes place. The image got from the smart camera goes to the YOLO architecture where the detection and recognition of the vegetables takes place from the retrained weights that we trained for days. Then the output contains the image with the labels and the ingredients list. This list is again goes to neural network and for generating the recipes for the user.

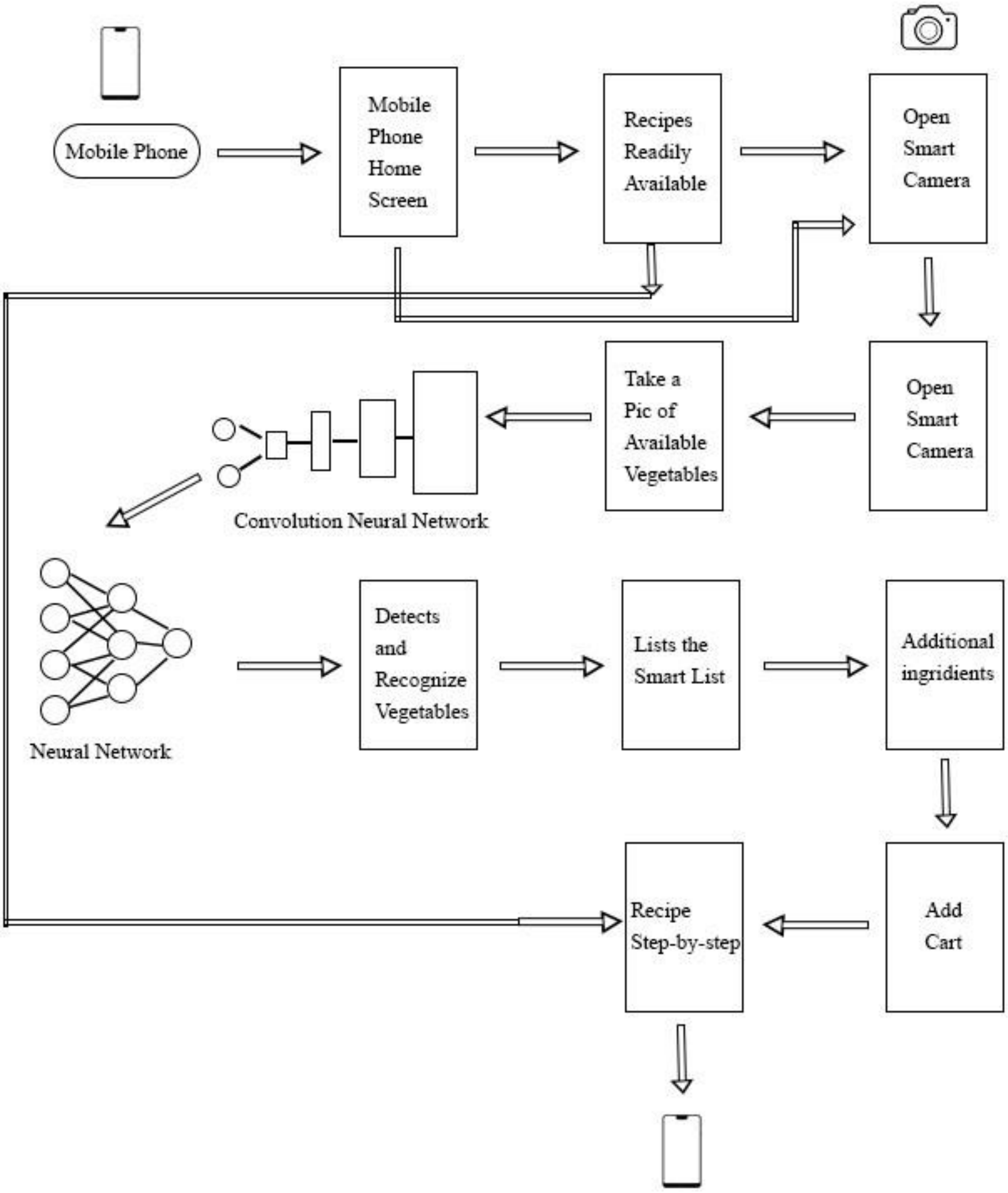


Figure 1.4 Application architecture

Chapter 4

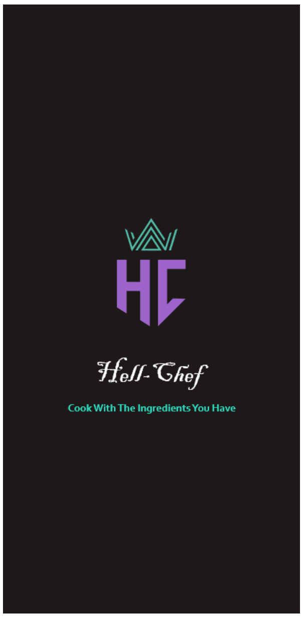
UX and UI Design

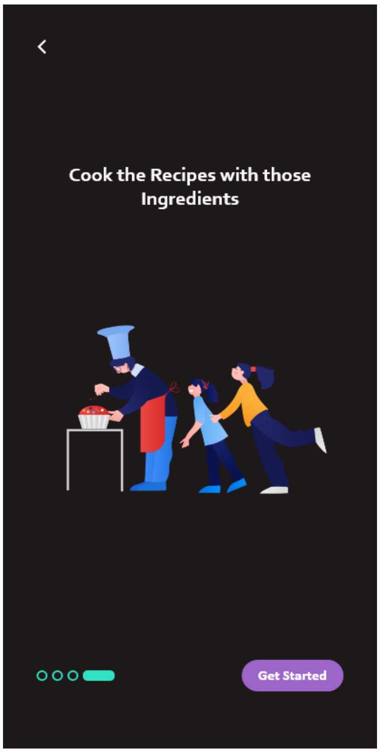
4.1 INTRODUCTION

To make a app successful, we need the functionality and the good design for the user to use our application. Design is a important process in the any part of the software engineering. A good design represent the good product.

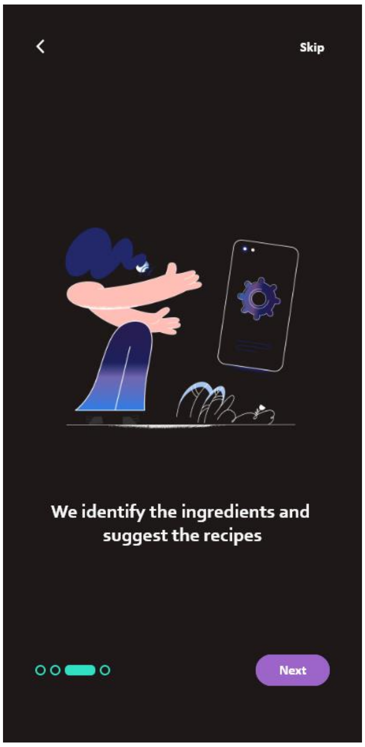
UX and UI design are two different elements of a single consumer experience. UX refers to the user experience, which focuses on how something works and how people interact with it. UI, or user interface, focuses on the look and layout. A UX designer ensures a product makes sense to the user by creating a path that logically flows from one step to the next. A UI designer ensures each page visually communicates that path. Designers also research targeted users to develop a clear understanding of their needs, define interaction models, design wireframes, build prototypes and work on brand color. And they conduct user testing and review metrics and focus-group reactions so they’re able to make the necessary tweaks to enhance the product.

4.2 UI DESIGN FOR HELL-CHEF

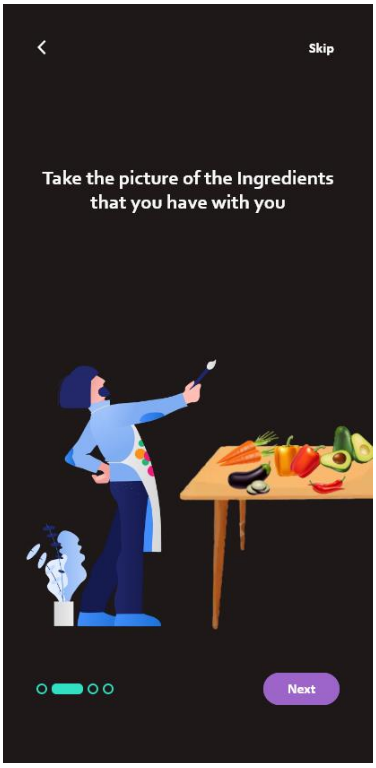




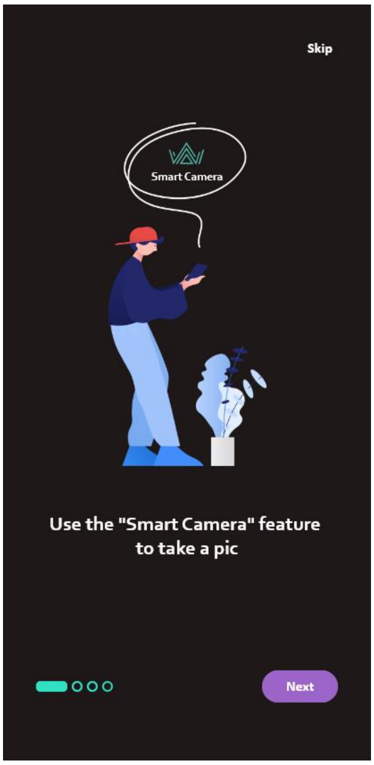
Cook the Recipes with those Ingredients



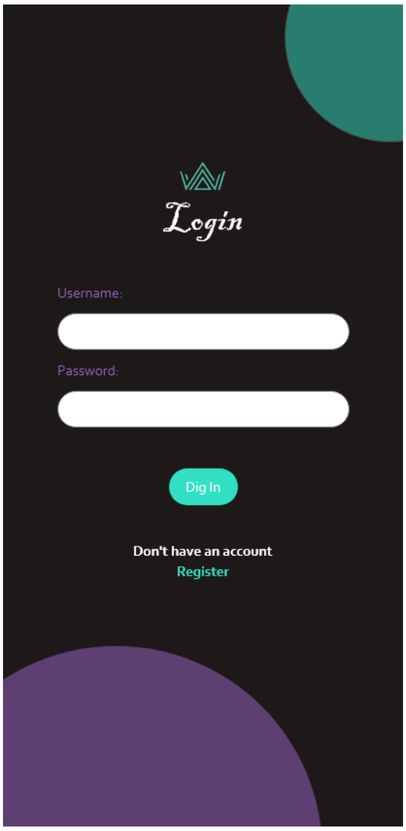
We identify the ingredients and suggest the recipes



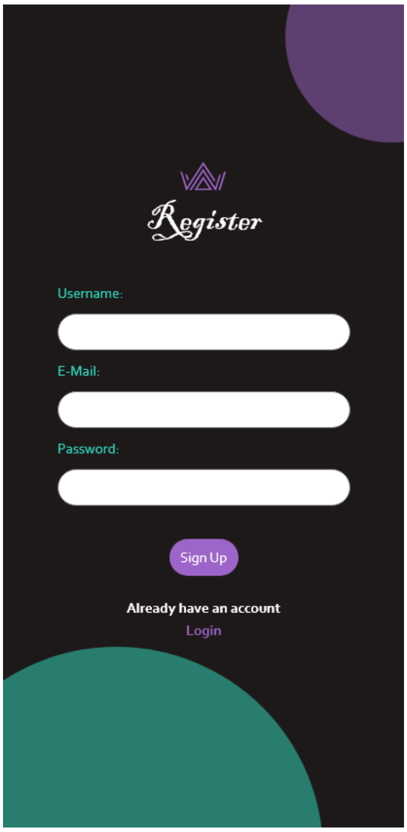
Take the picture of the Ingredients that you have with you



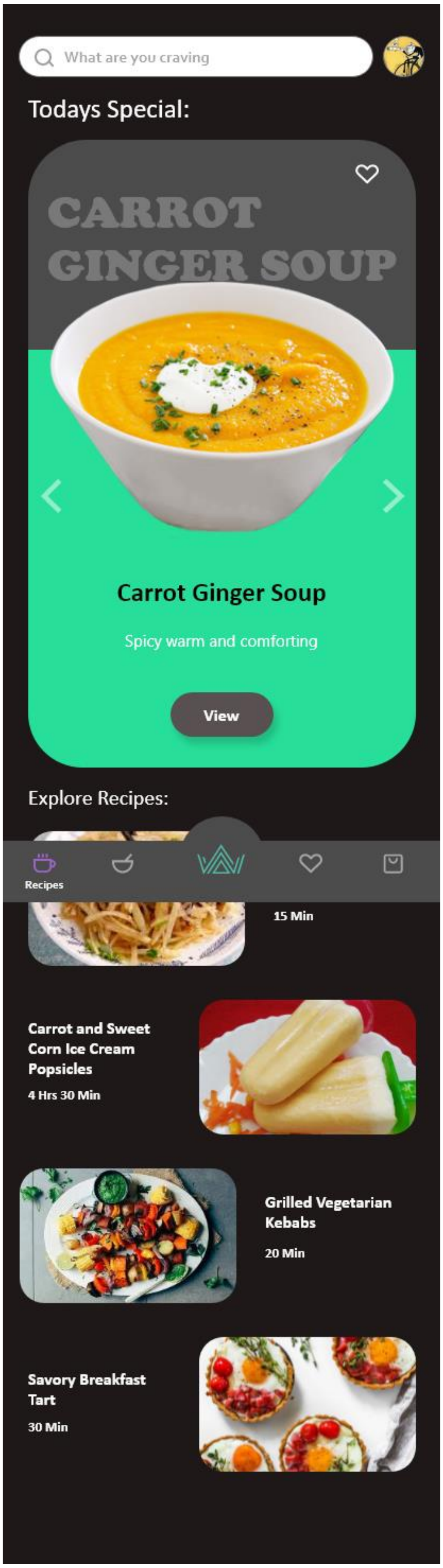
Use the "Smart Camera" feature to take a pic



Don't have an account



Already have an account



What are you craving

Todays Special:

CARROT GINGER SOUP

Carrot Ginger Soup

Spicy warm and comforting

View

Explore Recipes:

Recipes

15 Min

Carrot and Sweet Corn Ice Cream Popsicles

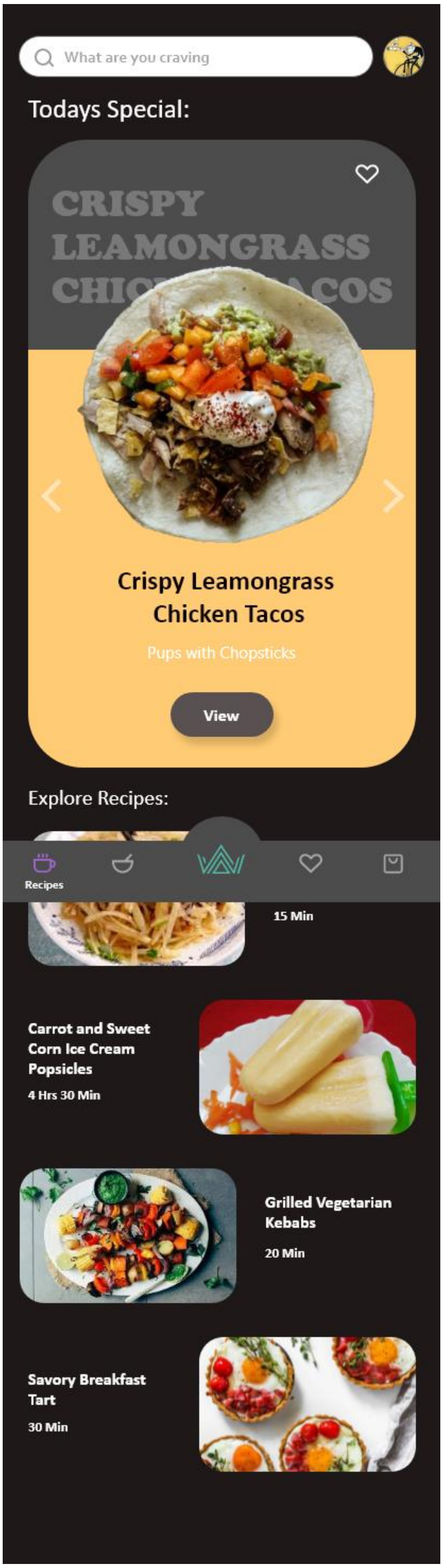
4 Hrs 30 Min

Grilled Vegetarian Kebabs

20 Min

Savory Breakfast Tart

30 Min



What are you craving

Todays Special:

CRISPY LEAMONGRASS CHIC ACOS

Crispy Leamongrass Chicken Tacos

Pups with Chopsticks

View

Explore Recipes:

Recipes

15 Min

Carrot and Sweet Corn Ice Cream Popsicles

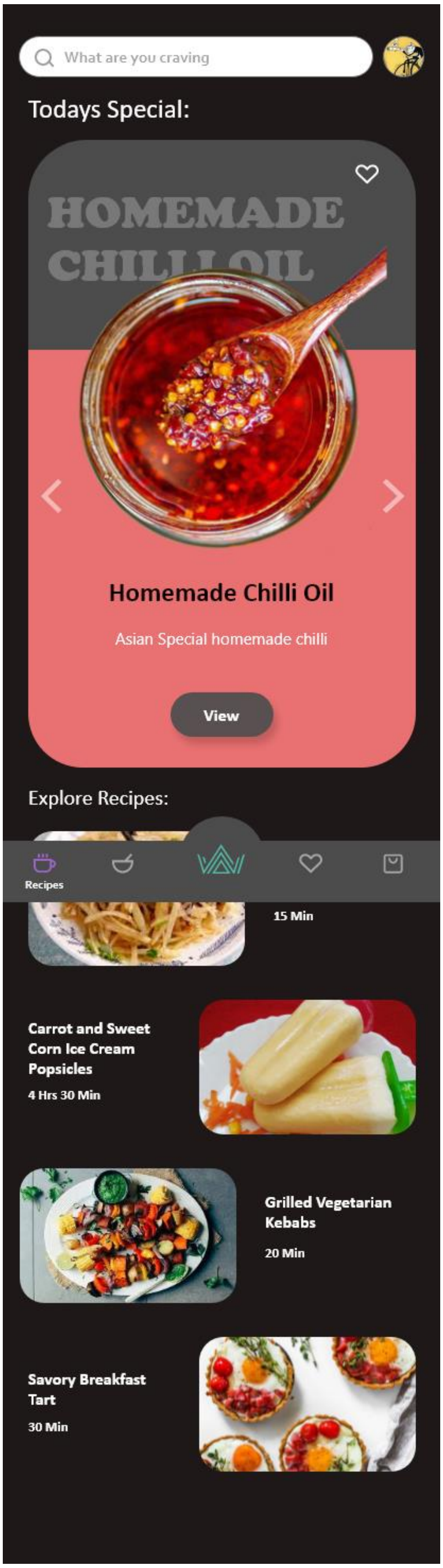
4 Hrs 30 Min

Grilled Vegetarian Kebabs

20 Min

Savory Breakfast Tart

30 Min



Homemade Chilli Oil

Asian Special homemade chilli

15 Min

Carrot and Sweet Corn Ice Cream Popsicles

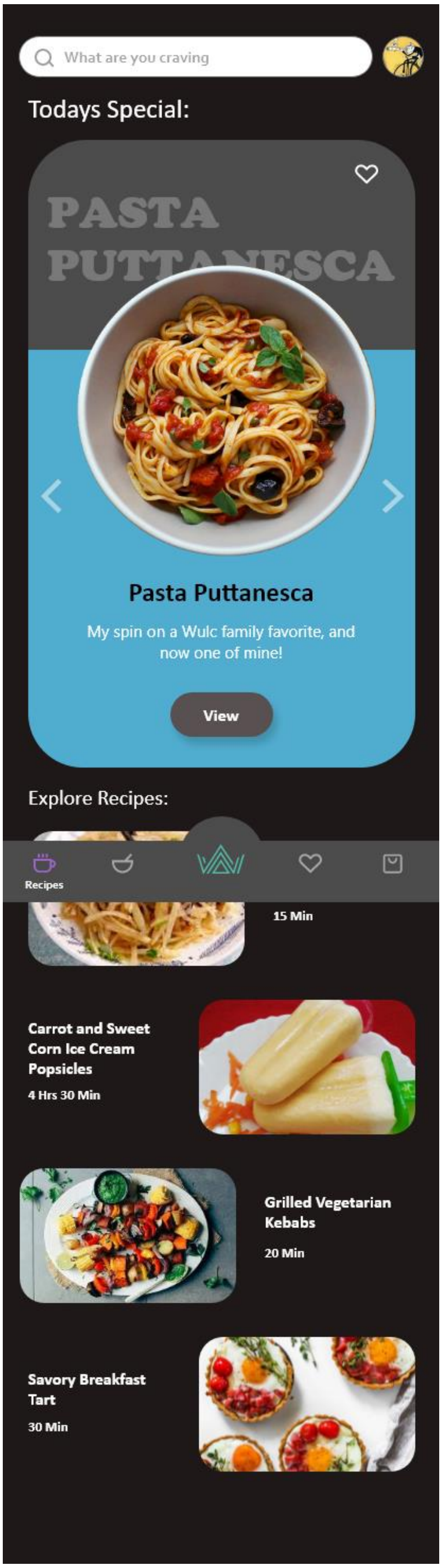
4 Hrs 30 Min

Grilled Vegetarian Kebabs

20 Min

Savory Breakfast Tart

30 Min



Pasta Puttanesca

My spin on a Wulc family favorite, and now one of mine!

15 Min

Carrot and Sweet Corn Ice Cream Popsicles

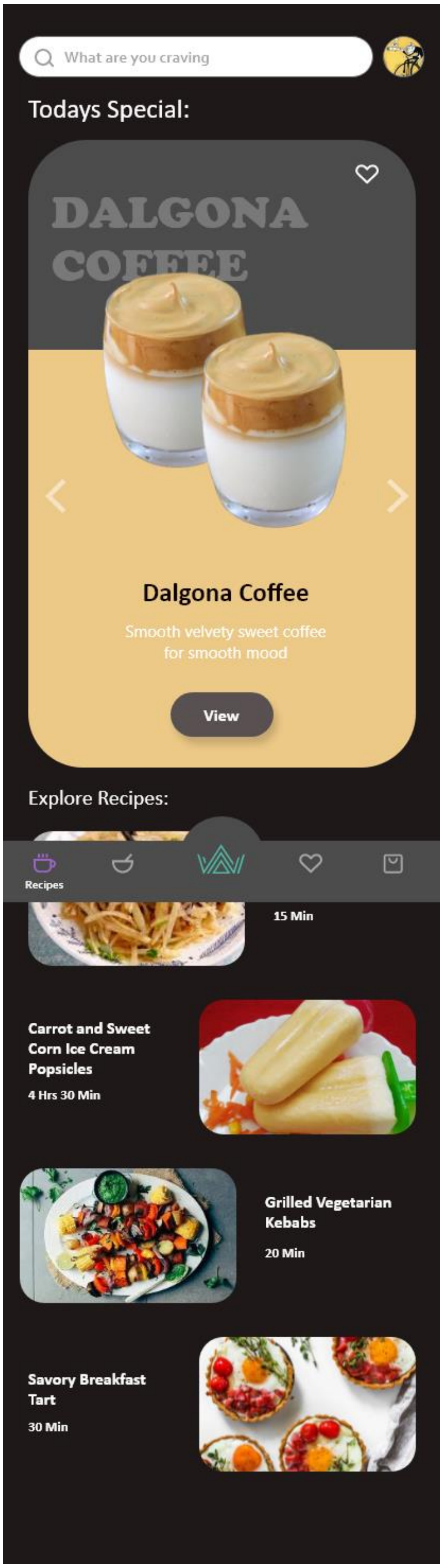
4 Hrs 30 Min

Grilled Vegetarian Kebabs

20 Min

Savory Breakfast Tart

30 Min



Todays Special:

Dalgona Coffee

Smooth velvety sweet coffee for smooth mood

Explore Recipes:

15 Min

Carrot and Sweet Corn Ice Cream Popsicles

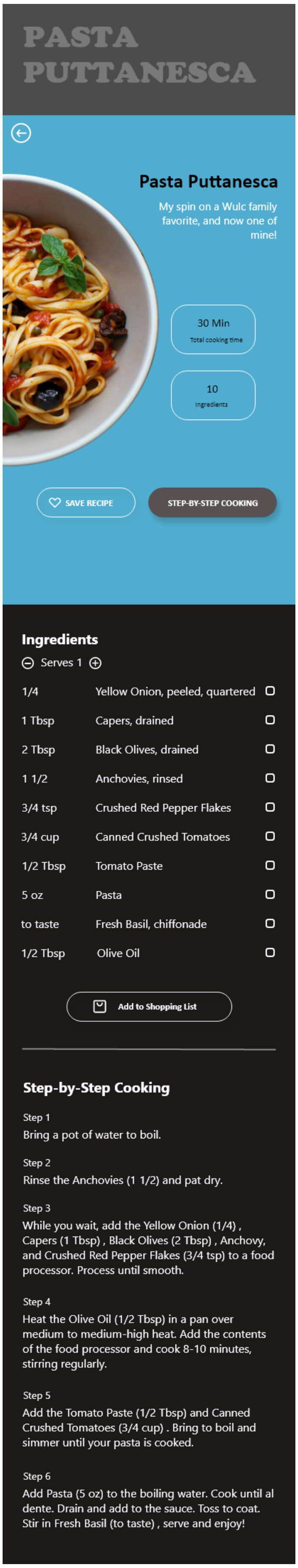
4 Hrs 30 Min

Grilled Vegetarian Kebabs

20 Min

Savory Breakfast Tart

30 Min



PASTA PUTTANESCA

Pasta Puttanesca

My spin on a Wulc family favorite, and now one of mine!

Ingredients

⊖ Serves 1 ⊕

1/4	Yellow Onion, peeled, quartered	□
1 Tbsp	Capers, drained	□
2 Tbsp	Black Olives, drained	□
1 1/2	Anchovies, rinsed	□
3/4 tsp	Crushed Red Pepper Flakes	□
3/4 cup	Canned Crushed Tomatoes	□
1/2 Tbsp	Tomato Paste	□
5 oz	Pasta	□
to taste	Fresh Basil, chiffonade	□
1/2 Tbsp	Olive Oil	□

Step-by-Step Cooking

Step 1
Bring a pot of water to boil.

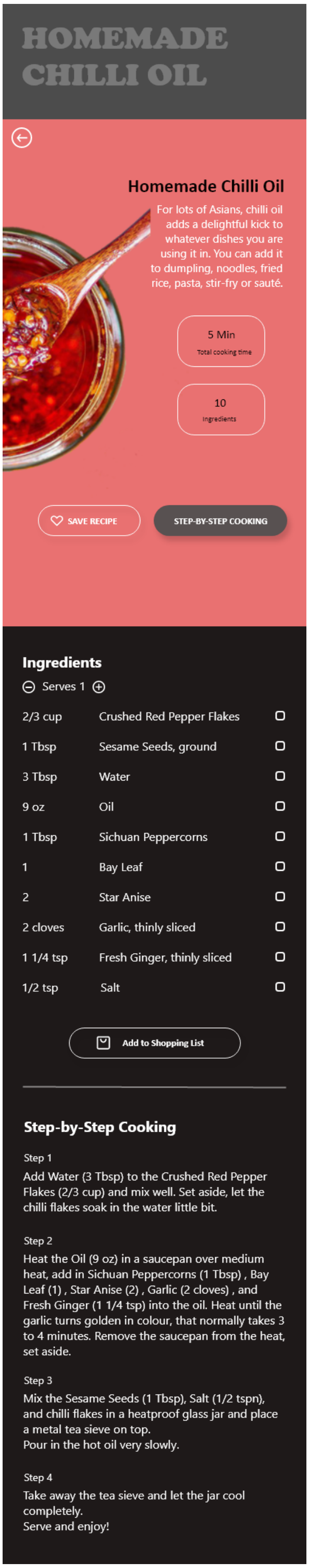
Step 2
Rinse the Anchovies (1 1/2) and pat dry.

Step 3
While you wait, add the Yellow Onion (1/4) , Capers (1 Tbsp) , Black Olives (2 Tbsp) , Anchovy, and Crushed Red Pepper Flakes (3/4 tsp) to a food processor. Process until smooth.

Step 4
Heat the Olive Oil (1/2 Tbsp) in a pan over medium to medium-high heat. Add the contents of the food processor and cook 8-10 minutes, stirring regularly.

Step 5
Add the Tomato Paste (1/2 Tbsp) and Canned Crushed Tomatoes (3/4 cup) . Bring to boil and simmer until your pasta is cooked.

Step 6
Add Pasta (5 oz) to the boiling water. Cook until al dente. Drain and add to the sauce. Toss to coat. Stir in Fresh Basil (to taste) , serve and enjoy!



HOMEMADE CHILLI OIL

Homemade Chilli Oil

For lots of Asians, chilli oil adds a delightful kick to whatever dishes you are using it in. You can add it to dumpling, noodles, fried rice, pasta, stir-fry or sauté.

Ingredients

⊖ Serves 1 ⊕

2/3 cup	Crushed Red Pepper Flakes	□
1 Tbsp	Sesame Seeds, ground	□
3 Tbsp	Water	□
9 oz	Oil	□
1 Tbsp	Sichuan Peppercorns	□
1	Bay Leaf	□
2	Star Anise	□
2 cloves	Garlic, thinly sliced	□
1 1/4 tsp	Fresh Ginger, thinly sliced	□
1/2 tsp	Salt	□

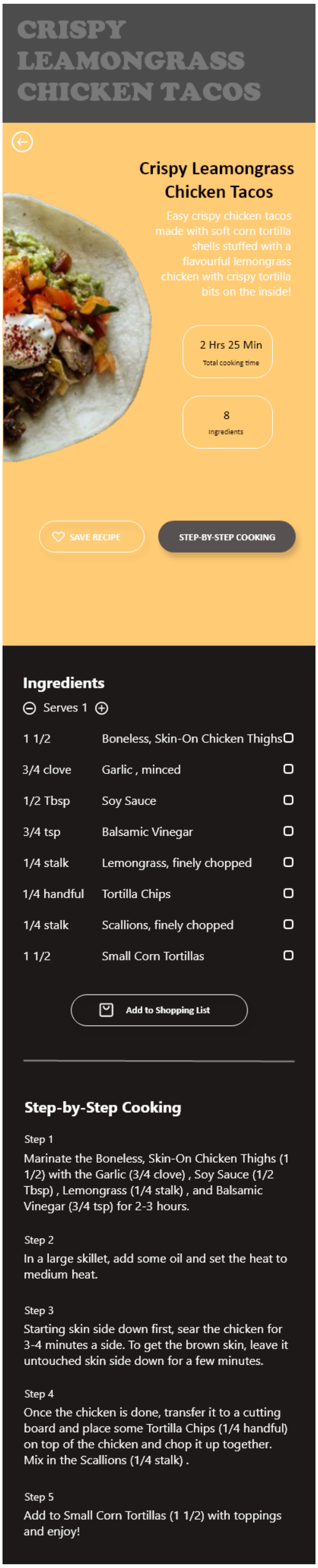
Step-by-Step Cooking

Step 1
Add Water (3 Tbsp) to the Crushed Red Pepper Flakes (2/3 cup) and mix well. Set aside, let the chilli flakes soak in the water little bit.

Step 2
Heat the Oil (9 oz) in a saucepan over medium heat, add in Sichuan Peppercorns (1 Tbsp) , Bay Leaf (1) , Star Anise (2) , Garlic (2 cloves) , and Fresh Ginger (1 1/4 tsp) into the oil. Heat until the garlic turns golden in colour, that normally takes 3 to 4 minutes. Remove the saucepan from the heat, set aside.

Step 3
Mix the Sesame Seeds (1 Tbsp), Salt (1/2 tspn), and chilli flakes in a heatproof glass jar and place a metal tea sieve on top.
Pour in the hot oil very slowly.

Step 4
Take away the tea sieve and let the jar cool completely.
Serve and enjoy!



CRISPY LEAMONGRASS CHICKEN TACOS

Crispy Leamongrass Chicken Tacos

Easy crispy chicken tacos made with soft corn tortilla shells stuffed with a flavourful lemongrass chicken with crispy tortilla bits on the inside!

2 Hrs 25 Min
Total cooking time

8
Ingredients

Ingredients

Serves 1

1 1/2	Boneless, Skin-On Chicken Thighs	□
3/4 clove	Garlic , minced	□
1/2 Tbsp	Soy Sauce	□
3/4 tsp	Balsamic Vinegar	□
1/4 stalk	Lemongrass, finely chopped	□
1/4 handful	Tortilla Chips	□
1/4 stalk	Scallions, finely chopped	□
1 1/2	Small Corn Tortillas	□

Step-by-Step Cooking

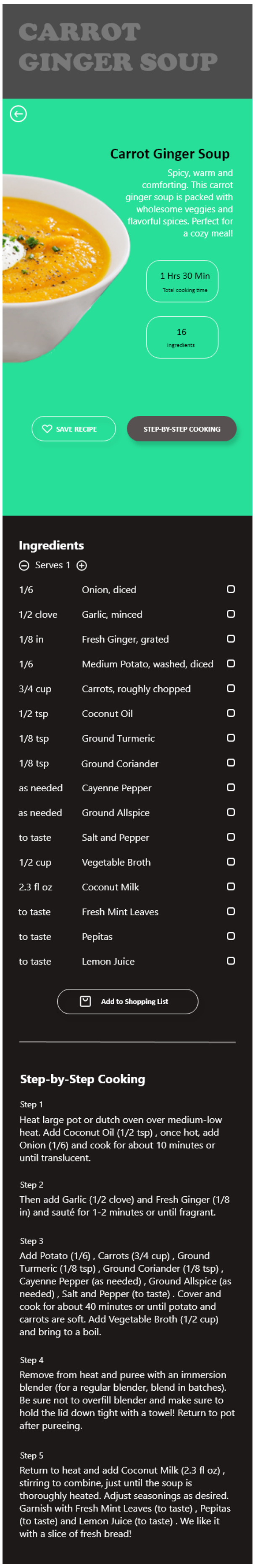
Step 1
Marinate the Boneless, Skin-On Chicken Thighs (1 1/2) with the Garlic (3/4 clove) , Soy Sauce (1/2 Tbsp) , Lemongrass (1/4 stalk) , and Balsamic Vinegar (3/4 tsp) for 2-3 hours.

Step 2
In a large skillet, add some oil and set the heat to medium heat.

Step 3
Starting skin side down first, sear the chicken for 3-4 minutes a side. To get the brown skin, leave it untouched skin side down for a few minutes.

Step 4
Once the chicken is done, transfer it to a cutting board and place some Tortilla Chips (1/4 handful) on top of the chicken and chop it up together. Mix in the Scallions (1/4 stalk) .

Step 5
Add to Small Corn Tortillas (1 1/2) with toppings and enjoy!



CARROT GINGER SOUP

Carrot Ginger Soup

Spicy, warm and comforting. This carrot ginger soup is packed with wholesome veggies and flavorful spices. Perfect for a cozy meal!

1 Hrs 30 Min
Total cooking time

16
Ingredients

Ingredients

Serves 1

1/6	Onion, diced	□
1/2 clove	Garlic, minced	□
1/8 in	Fresh Ginger, grated	□
1/6	Medium Potato, washed, diced	□
3/4 cup	Carrots, roughly chopped	□
1/2 tsp	Coconut Oil	□
1/8 tsp	Ground Turmeric	□
1/8 tsp	Ground Coriander	□
as needed	Cayenne Pepper	□
as needed	Ground Allspice	□
to taste	Salt and Pepper	□
1/2 cup	Vegetable Broth	□
2.3 fl oz	Coconut Milk	□
to taste	Fresh Mint Leaves	□
to taste	Pepitas	□
to taste	Lemon Juice	□

Step-by-Step Cooking

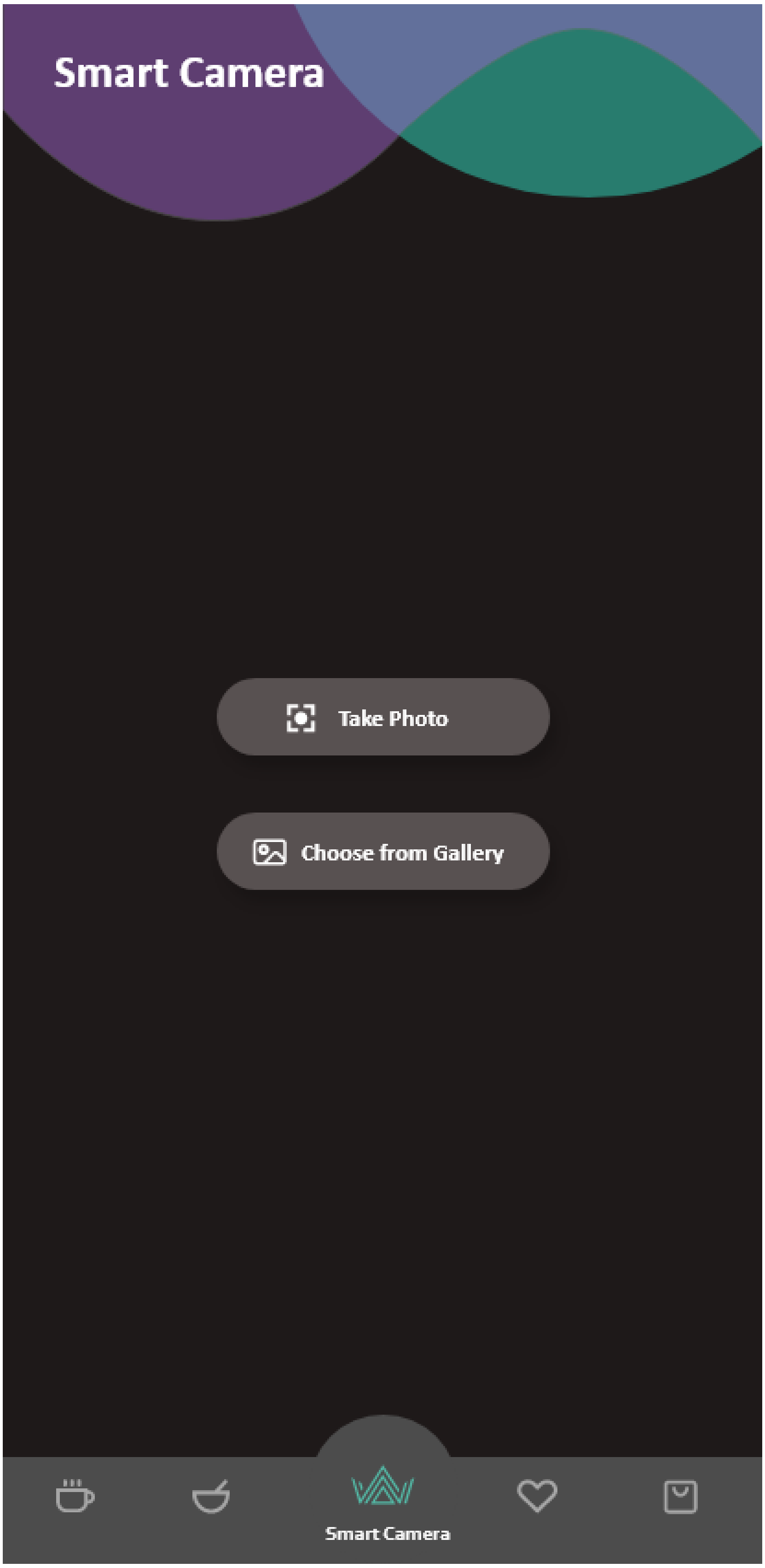
Step 1
Heat large pot or dutch oven over medium-low heat. Add Coconut Oil (1/2 tsp) , once hot, add Onion (1/6) and cook for about 10 minutes or until translucent.

Step 2
Then add Garlic (1/2 clove) and Fresh Ginger (1/8 in) and sauté for 1-2 minutes or until fragrant.

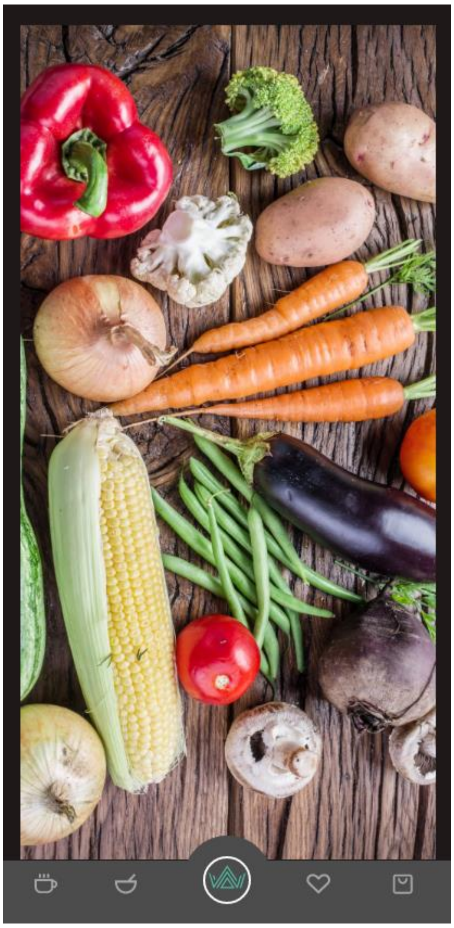
Step 3
Add Potato (1/6) , Carrots (3/4 cup) , Ground Turmeric (1/8 tsp) , Ground Coriander (1/8 tsp) , Cayenne Pepper (as needed) , Ground Allspice (as needed) , Salt and Pepper (to taste) . Cover and cook for about 40 minutes or until potato and carrots are soft. Add Vegetable Broth (1/2 cup) and bring to a boil.

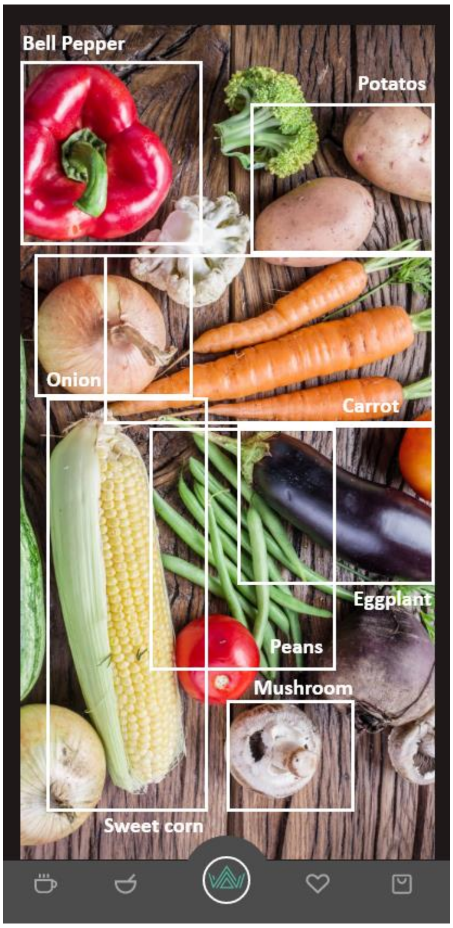
Step 4
Remove from heat and puree with an immersion blender (for a regular blender, blend in batches). Be sure not to overfill blender and make sure to hold the lid down tight with a towel! Return to pot after pureeing.

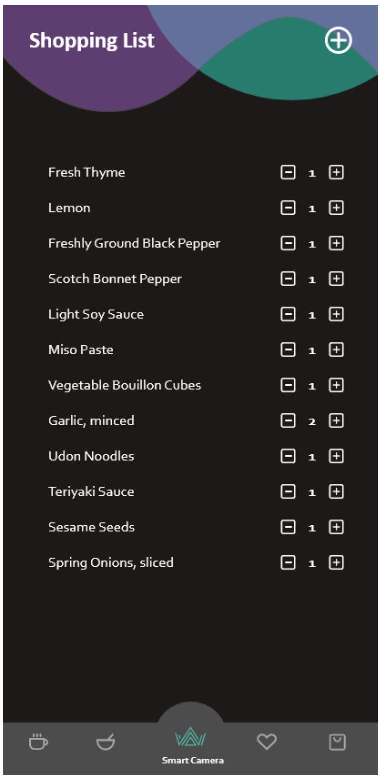
Step 5
Return to heat and add Coconut Milk (2.3 fl oz) , stirring to combine, just until the soup is thoroughly heated. Adjust seasonings as desired. Garnish with Fresh Mint Leaves (to taste) , Pepitas (to taste) and Lemon Juice (to taste) . We like it with a slice of fresh bread!



Smart Camera

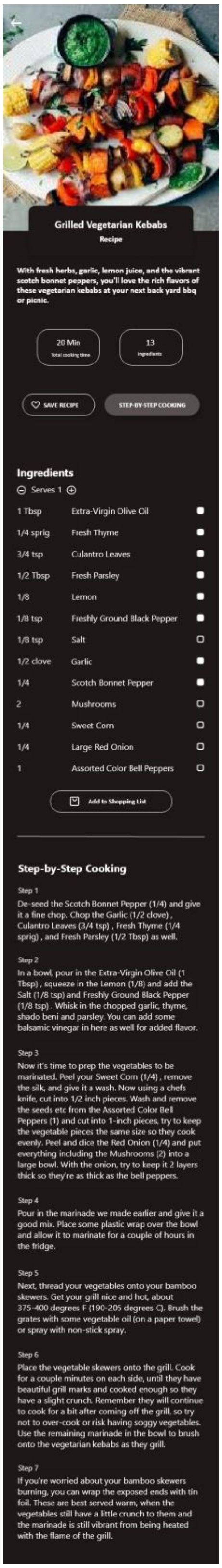






Shopping List ⊕

Fresh Thyme	⊟ 1 ⊞
Lemon	⊟ 1 ⊞
Freshly Ground Black Pepper	⊟ 1 ⊞
Scotch Bonnet Pepper	⊟ 1 ⊞
Light Soy Sauce	⊟ 1 ⊞
Miso Paste	⊟ 1 ⊞
Vegetable Bouillon Cubes	⊟ 1 ⊞
Garlic, minced	⊟ 2 ⊞
Udon Noodles	⊟ 1 ⊞
Teriyaki Sauce	⊟ 1 ⊞
Sesame Seeds	⊟ 1 ⊞
Spring Onions, sliced	⊟ 1 ⊞



Grilled Vegetarian Kebabs
Recipe

With fresh herbs, garlic, lemon juice, and the vibrant scotch bonnet peppers, you'll love the rich flavors of these vegetarian kebabs at your next back yard bbq or picnic.

Ingredients

⊖ Serves 1 ⊕

1 Tbsp	Extra-Virgin Olive Oil	■
1/4 sprig	Fresh Thyme	■
3/4 tsp	Culantro Leaves	■
1/2 Tbsp	Fresh Parsley	■
1/8	Lemon	■
1/8 tsp	Freshly Ground Black Pepper	■
1/8 tsp	Salt	□
1/2 clove	Garlic	■
1/4	Scotch Bonnet Pepper	■
2	Mushrooms	□
1/4	Sweet Corn	□
1/4	Large Red Onion	□
1	Assorted Color Bell Peppers	□

Step-by-Step Cooking

Step 1
De-seed the Scotch Bonnet Pepper (1/4) and give it a fine chop. Chop the Garlic (1/2 clove) , Culantro Leaves (3/4 tsp) , Fresh Thyme (1/4 sprig) , and Fresh Parsley (1/2 Tbsp) as well.

Step 2
In a bowl, pour in the Extra-Virgin Olive Oil (1 Tbsp) , squeeze in the Lemon (1/8) and add the Salt (1/8 tsp) and Freshly Ground Black Pepper (1/8 tsp) . Whisk in the chopped garlic, thyme, shado beni and parsley. You can add some balsamic vinegar in here as well for added flavor.

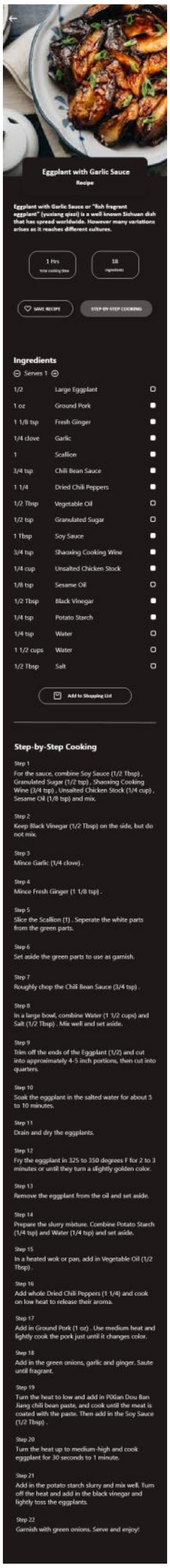
Step 3
Now it's time to prep the vegetables to be marinated. Peel your Sweet Corn (1/4) , remove the silk, and give it a wash. Now using a chefs knife, cut into 1/2 inch pieces. Wash and remove the seeds etc from the Assorted Color Bell Peppers (1) and cut into 1-inch pieces, try to keep the vegetable pieces the same size so they cook evenly. Peel and dice the Red Onion (1/4) and put everything including the Mushrooms (2) into a large bowl. With the onion, try to keep it 2 layers thick so they're as thick as the bell peppers.

Step 4
Pour in the marinade we made earlier and give it a good mix. Place some plastic wrap over the bowl and allow it to marinate for a couple of hours in the fridge.

Step 5
Next, thread your vegetables onto your bamboo skewers. Get your grill nice and hot, about 375-400 degrees F (190-205 degrees C). Brush the grates with some vegetable oil (on a paper towel) or spray with non-stick spray.

Step 6
Place the vegetable skewers onto the grill. Cook for a couple minutes on each side, until they have beautiful grill marks and cooked enough so they have a slight crunch. Remember they will continue to cook for a bit after coming off the grill, so try not to over-cook or risk having soggy vegetables. Use the remaining marinade in the bowl to brush onto the vegetarian kebabs as they grill.

Step 7
If you're worried about your bamboo skewers burning, you can wrap the exposed ends with tin foil. These are best served warm, when the vegetables still have a little crunch to them and the marinade is still vibrant from being heated with the flame of the grill.



Eggplant with Garlic Sauce
Recipe

Eggplant with Garlic Sauce or "fish fragrant eggplant" (yuxiang qiezi) is a well known Sichuan dish that has spread worldwide. However many variations arises as it reaches different cultures.

Ingredients

⊖ Serves 1 ⊕

1/2	Large Eggplant	□
1 oz	Ground Pork	■
1 1/8 tsp	Fresh Ginger	■
1/4 clove	Garlic	■
1	Scallion	■
3/4 tsp	Chili Bean Sauce	■
1 1/4	Dried Chili Peppers	■
1/2 Tbsp	Vegetable Oil	□
1/2 tsp	Granulated Sugar	□
1 Tbsp	Soy Sauce	■
3/4 tsp	Shaoxing Cooking Wine	■
1/4 cup	Unsalted Chicken Stock	■
1/8 tsp	Sesame Oil	□
1/2 Tbsp	Black Vinegar	■
1/4 tsp	Potato Starch	■
1/4 tsp	Water	□
1 1/2 cups	Water	□
1/2 Tbsp	Salt	□

Step-by-Step Cooking

Step 1
For the sauce, combine Soy Sauce (1/2 Tbsp) , Granulated Sugar (1/2 tsp) , Shaoxing Cooking Wine (3/4 tsp) , Unsalted Chicken Stock (1/4 cup) , Sesame Oil (1/8 tsp) and mix.

Step 2
Keep Black Vinegar (1/2 Tbsp) on the side, but do not mix.

Step 3
Mince Garlic (1/4 clove) .

Step 4
Mince Fresh Ginger (1 1/8 tsp) .

Step 5
Slice the Scallion (1) . Seperate the white parts from the green parts.

Step 6
Set aside the green parts to use as garnish.

Step 7
Roughly chop the Chili Bean Sauce (3/4 tsp) .

Step 8
In a large bowl, combine Water (1 1/2 cups) and Salt (1/2 Tbsp) . Mix well and set aside.

Step 9
Trim off the ends of the Eggplant (1/2) and cut into approximately 4-5 inch portions, then cut into quarters.

Step 10
Soak the eggplant in the salted water for about 5 to 10 minutes.

Step 11
Drain and dry the eggplants.

Step 12
Fry the eggplant in 325 to 350 degrees F for 2 to 3 minutes or until they turn a slightly golden color.

Step 13
Remove the eggplant from the oil and set aside.

Step 14
Prepare the slurry mixture. Combine Potato Starch (1/4 tsp) and Water (1/4 tsp) and set aside.

Step 15
In a heated wok or pan, add in Vegetable Oil (1/2 Tbsp) .

Step 16
Add whole Dried Chili Peppers (1 1/4) and cook on low heat to release their aroma.

Step 17
Add in Ground Pork (1 oz) . Use medium heat and lightly cook the pork just until it changes color.

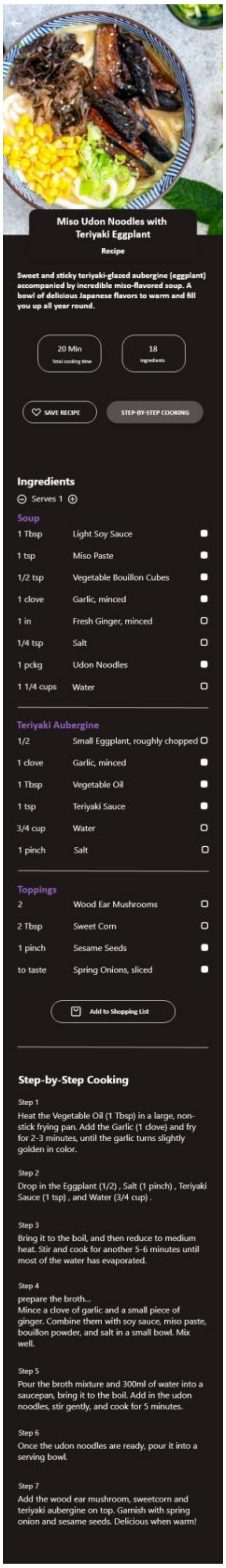
Step 18
Add in the green onions, garlic and ginger. Saute until fragrant.

Step 19
Turn the heat to low and add in PiXian Dou Ban Jiang chili bean paste, and cook until the meat is coated with the paste. Then add in the Soy Sauce (1/2 Tbsp) .

Step 20
Turn the heat up to medium-high and cook eggplant for 30 seconds to 1 minute.

Step 21
Add in the potato starch slurry and mix well. Turn off the heat and add in the black vinegar and lightly toss the eggplants.

Step 22
Garnish with green onions. Serve and enjoy!



Miso Udon Noodles with Teriyaki Eggplant
Recipe

Sweet and sticky teriyaki-glazed aubergine (eggplant) accompanied by incredible miso-flavored soup. A bowl of delicious Japanese flavors to warm and fill you up all year round.

Ingredients

⊖ Serves 1 ⊕

Soup

1 Tbsp	Light Soy Sauce	■
1 tsp	Miso Paste	■
1/2 tsp	Vegetable Bouillon Cubes	■
1 clove	Garlic, minced	■
1 in	Fresh Ginger, minced	□
1/4 tsp	Salt	□
1 pckg	Udon Noodles	■
1 1/4 cups	Water	□

Teriyaki Aubergine

1/2	Small Eggplant, roughly chopped	□
1 clove	Garlic, minced	■
1 Tbsp	Vegetable Oil	■
1 tsp	Teriyaki Sauce	■
3/4 cup	Water	□
1 pinch	Salt	□

Toppings

2	Wood Ear Mushrooms	□
2 Tbsp	Sweet Corn	□
1 pinch	Sesame Seeds	■
to taste	Spring Onions, sliced	■

Step-by-Step Cooking

Step 1
Heat the Vegetable Oil (1 Tbsp) in a large, non-stick frying pan. Add the Garlic (1 clove) and fry for 2-3 minutes, until the garlic turns slightly golden in color.

Step 2
Drop in the Eggplant (1/2) , Salt (1 pinch) , Teriyaki Sauce (1 tsp) , and Water (3/4 cup) .

Step 3
Bring it to the boil, and then reduce to medium heat. Stir and cook for another 5-6 minutes until most of the water has evaporated.

Step 4
prepare the broth...
Mince a clove of garlic and a small piece of ginger. Combine them with soy sauce, miso paste, bouillon powder, and salt in a small bowl. Mix well.

Step 5
Pour the broth mixture and 300ml of water into a saucepan, bring it to the boil. Add in the udon noodles, stir gently, and cook for 5 minutes.

Step 6
Once the udon noodles are ready, pour it into a serving bowl.

Step 7
Add the wood ear mushroom, sweetcorn and teriyaki aubergine on top. Garnish with spring onion and sesame seeds. Delicious when warm!

Figure 1.5 UI Design

Chapter 5

Module description

5.1 LOGIN MODULE

As this a mobile application, we kept a login system for user to login into the application using his mail address and password. In further extension of the application, the user can create his own recipes and they are stored in the cloud. The login helps to access all those details. We use firebase database to authenticate the user to login the application.

5.2 REGISTER MODULE

The users can register for the application. The details are taken as a form and stored in the database. The details like name, username, password, mail address are taken in order to register the user for this application. Where the mail and password is saved in the authentication

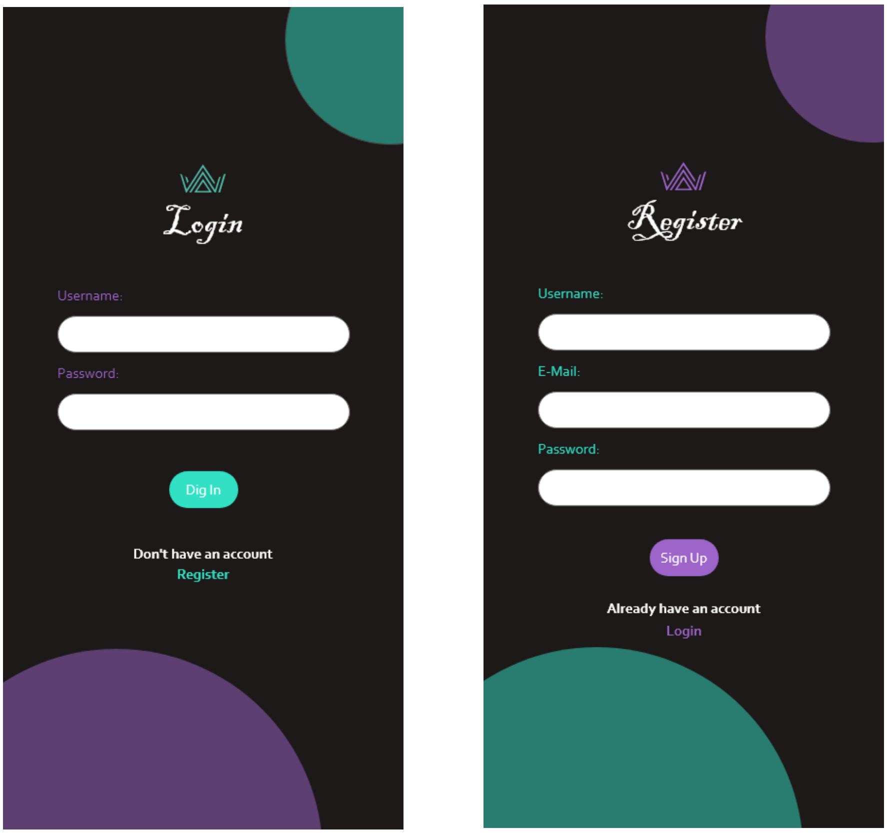


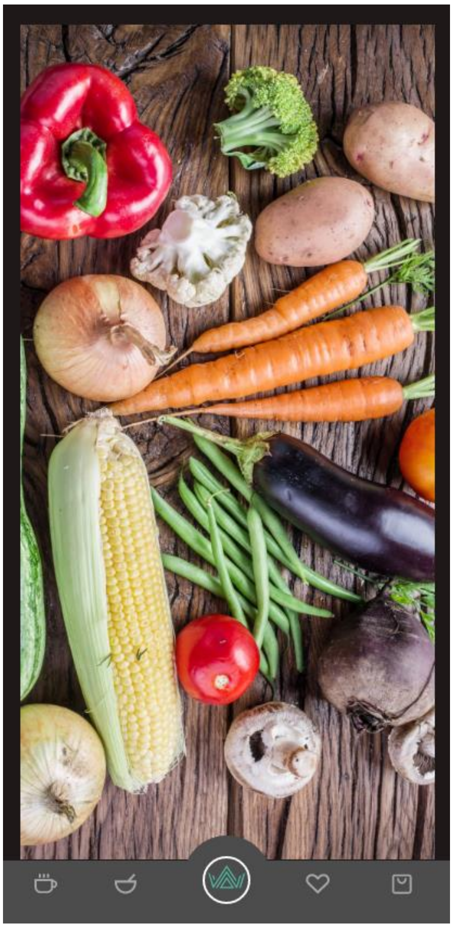
Figure 1.6 Login and Register Module

5.3 RECIPES LIST

After entering into the app, the user gets the basic recipes for breakfast, lunch or dinner. These recipes will have the image, ingredients, step-by-step preparation guide and time that will be taken for the dish to be ready. In further development, we can also include the video of the making, reviews on the dish etc.

5.4 SMART CAMERA

This is the key point in the application. The smart camera take the photo of items present in the house like fruits and vegetables. The image is then sent to the CNN for detection and recognition of the items. These are done by using the algorithms like YOLO and improvised RESNET. The YOLO algorithm helps to detect multiple objects in our case the vegetables and fruits. Then a boundary box is created upon the objects detected. Then the improvised RESNET helps to identify the object and stores the ingredients as a list. The use of improvised RESNET is that the training and testing time and provide accuracy.



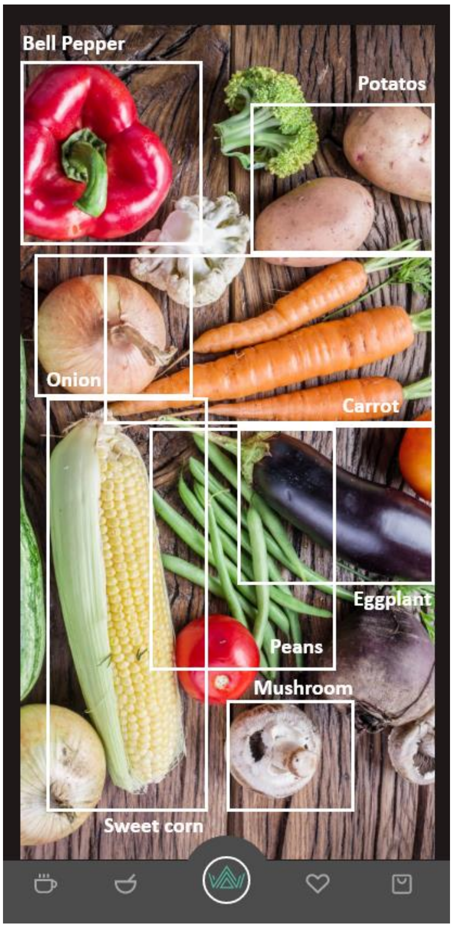


Figure 1.7 Smart camera

5.5 SHOPPING LIST

The items majorly used are now available with us. But, the additional items that are required are listed separately. Those items can be checked manually or can be added to cart for purchasing them. We can add the shopping list directly from the list available.

Chapter 6

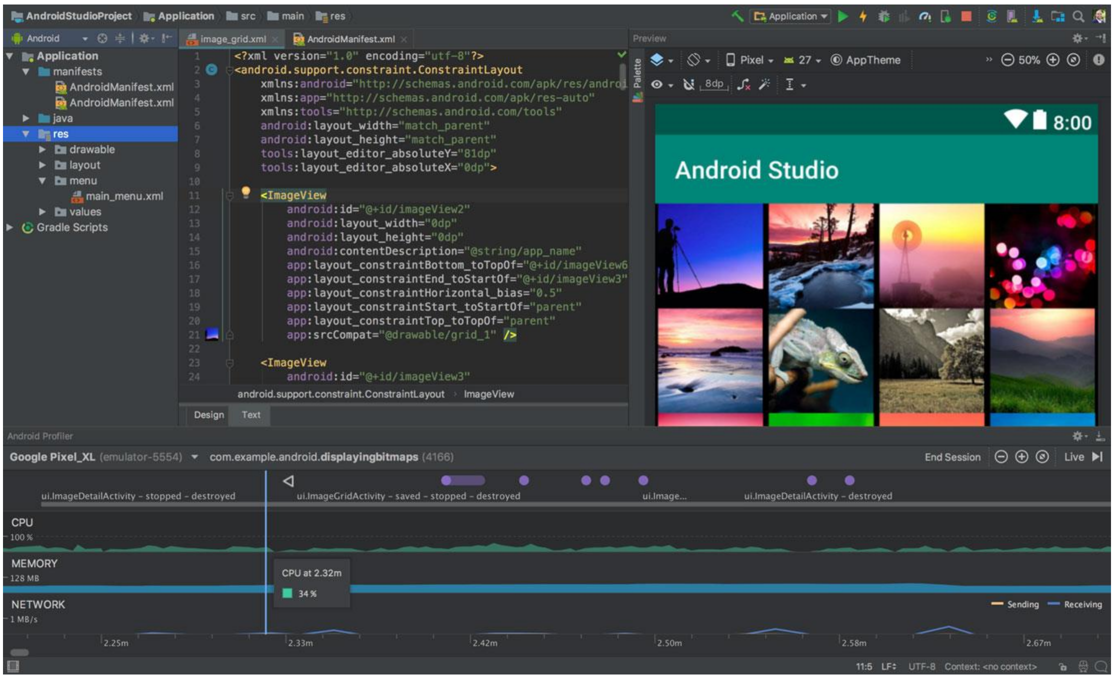
System Implementation

6.1 OVERVIEW OF ENVIRONMENT

We are using many environment for our project since we have different in implementations. We develop the android application in android studio, the machine learning algorithm is developed in the google colab and the database we use firebase database. We integrate the firebase with android studio with API keys. The machine learning we connect using the openCV. For data storage we usually use firebase. But in the case of the regognition image we use google drive for it.

6.2 ANDROID STUDIO

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.



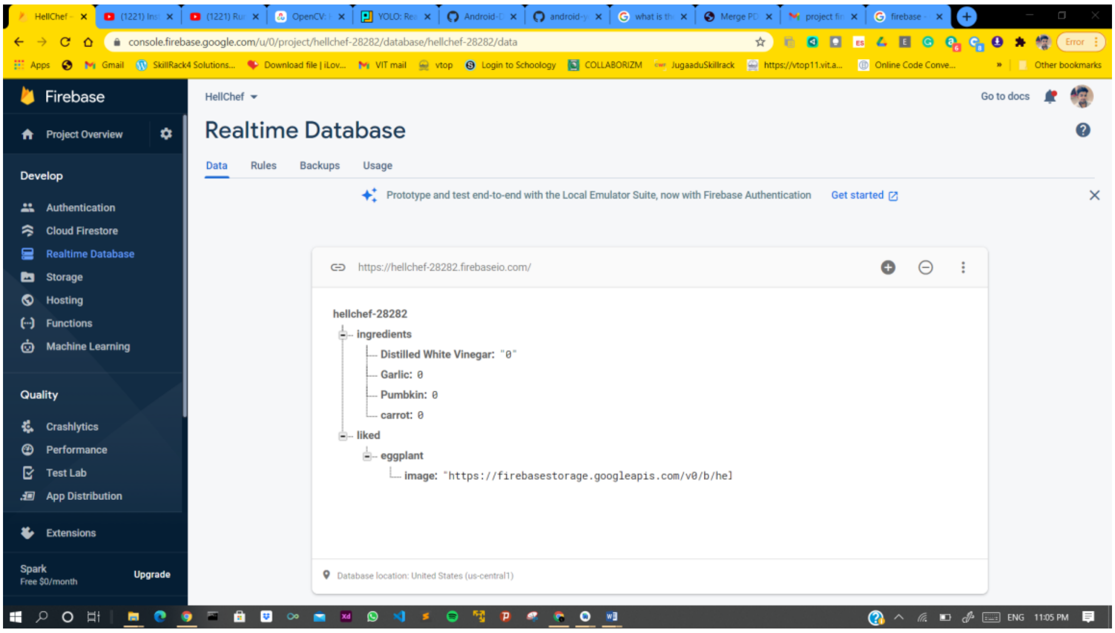
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="81dp"
    tools:layout_editor_absoluteX="0dp">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:contentDescription="@string/app_name"
        app:layout_constraintBottom_toTopOf="@+id/imageView6
        app:layout_constraintEnd_toStartOf="@+id/imageView3"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/grid_1" />

    <ImageView
        android:id="@+id/imageView3"
```

Figure 1.8 Android studio

6.3 FIREBASE

Firebase is a mobile-backend-as-a-service that provides powerful features for building mobile apps. Firebase has three core services: a realtime database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code

Firebase is Google's mobile application development platform that helps you build, improve, and grow your app. Here it is again in bigger letters, for impact: Firebase is Google's mobile application development platform that helps you build, improve, and grow your app.



console.firebase.google.com/u/0/project/hellchef-28282/database/hellchef-28282/data

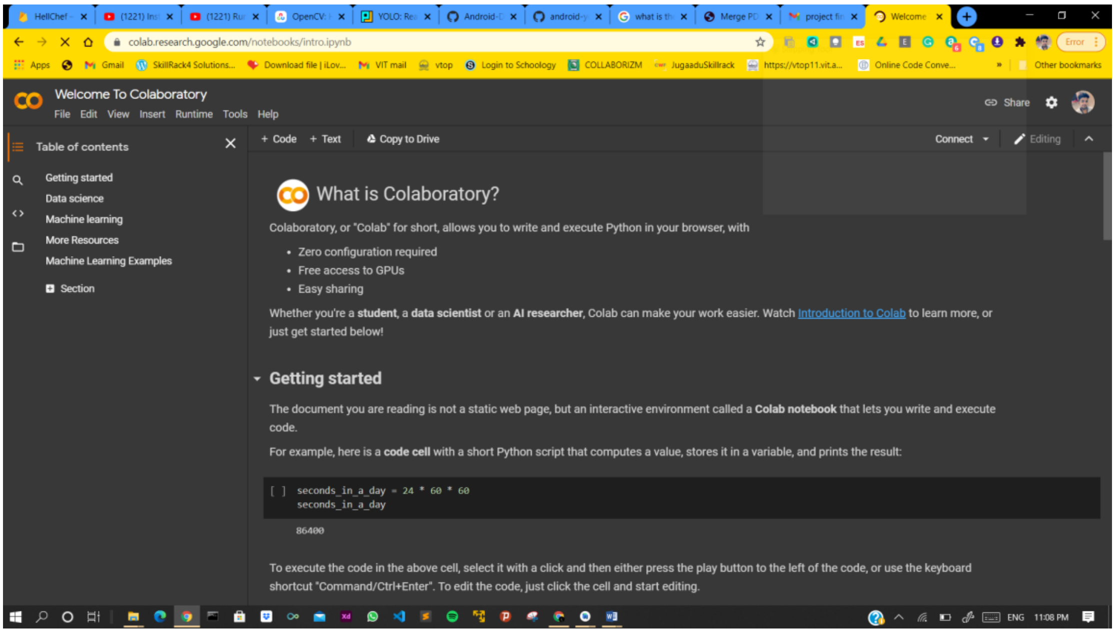
https://hellchef-28282.firebaseio.com/

```
hellchef-28282
 ⊟-- ingredients
 |    |-- Distilled White Vinegar: "0"
 |    |-- Garlic: 0
 |    |-- Pumbkin: 0
 |    └-- carrot: 0
 ⊟-- liked
      ⊟-- eggplant
           └-- image: "https://firebasestorage.googleapis.com/v0/b/hel
```

Figure 1.9 Firebase Database

6.4 GOOGLE COLAB

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. To start working with Colab you first need to log in to your google account, then go to this link https://colab.research.google.com. EXAMPLES: Contain a number of Jupyter notebooks of various examples. RECENT: Jupyter notebook you have recently worked with. GOOGLE DRIVE: Jupyter notebook in your google drive.



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Figure 1.10 Google Colab

6.5 IMPLEMENTATION

```
package com.example.hellchef;



import androidx.appcompat.app.AppCompatActivity;



import android.content.Intent;

import android.os.Bundle;

import android.view.Window;

import android.view.WindowManager;

import android.widget.Toast;



import java.util.Timer;

import java.util.TimerTask;



public class MainActivity extends AppCompatActivity {
    Timer timer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        /*requestWindowFeature(Window.FEATURE_NO_TITLE);

this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        getSupportActionBar().hide();*/
        timer = new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                Intent intent = new Intent(MainActivity.this,Home.class);
                startActivity(intent);
                finish();
            }
        },5000);

        setContentView(R.layout.activity_main);

    }
}
```

```
package com.example.hellchef;
```

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.google.firebase.auth.FirebaseAuth;

public class Recipes extends AppCompatActivity {
    FirebaseAuth mFirebaseAuth;
    FirebaseAuth.AuthStateListener mAuthStateListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recipes);
    }

    public void pasta(View view) {
        Intent in = new Intent(this,Pasta.class);
        startActivity(in);
    }

    public void chilli(View view) {
        Intent in = new Intent(this,Chilli.class);

        startActivity(in);

    }


    public void tacos(View view) {

        Intent in = new Intent(this,Tacos.class);

        startActivity(in);

    }


    public void coffee(View view) {

        Intent in = new Intent(this,Coffee.class);

        startActivity(in);

    }


    public void logout(View view) {

        FirebaseAuth.getInstance().signOut();

        Intent in = new Intent(Recipes.this,MainActivity.class);

        startActivity(in);

    }

}
```

```
package com.example.hellchef.navigation;


import android.content.Intent;

import android.os.Bundle;
import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;


import com.example.hellchef.Additional;

import com.example.hellchef.Chilli;

import com.example.hellchef.Coffee;

import com.example.hellchef.Home;

import com.example.hellchef.MainActivity;

import com.example.hellchef.Pasta;

import com.example.hellchef.R;

import com.example.hellchef.Recipes;

import com.example.hellchef.Tacos;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;


public class RecipesFragement extends Fragment {
    FirebaseAuth mFirebaseAuth;
    FirebaseAuth.AuthStateListener mAuthStateListener;
    final FirebaseDatabase mDatabase = FirebaseDatabase.getInstance();
    DatabaseReference mref = mDatabase.getReference();
    private DatabaseReference myRef;


    ImageView pasta,chilli,tacos,coffee,logout;
    ImageView pasta_liked,chilli_liked,tacos_liked,carrot_liked,coffee_liked;
    public int pasta_count,chilli_count,tacos_count,carrot_count,coffee_count;


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,Bundle
savedInstanceState){
        View view =  inflater.inflate(R.layout.activity_recipes,container,false);


        pasta = view.findViewById(R.id.view_pasta);
        chilli = view.findViewById(R.id.view_chilli);
        tacos = view.findViewById(R.id.view_tacos);
        coffee = view.findViewById(R.id.view_coffee);
        logout = view.findViewById(R.id.imageView21);


        pasta_liked = view.findViewById(R.id.heart1);
```

```
chilli_liked = view.findViewById(R.id.heart2);
tacos_liked = view.findViewById(R.id.heart3);
carrot_liked = view.findViewById(R.id.heart4);
coffee_liked = view.findViewById(R.id.heart5);


myRef = FirebaseDatabase.getInstance().getReference();
//myRef.child("liked");
myRef.child("liked").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if(snapshot.hasChild("pasta"))
        {
            pasta_liked.setImageResource(R.drawable.heart_fill);
            pasta_count = 1;
        }
        else
        {
            pasta_liked.setImageResource(R.drawable.heart_btn);
            pasta_count = 0;
        }
    }


    @Override
    public void onCancelled(@NonNull DatabaseError error) {
```

```
                Toast.makeText(getActivity(),"Something went 
wrong",Toast.LENGTH_SHORT).show();
            }
        });
        myRef.child("liked").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.hasChild("chilli"))
                {
                    chilli_liked.setImageResource(R.drawable.heart_fill);
                    chilli_count = 1;
                }
                else
                {
                    chilli_liked.setImageResource(R.drawable.heart_btn);
                    chilli_count = 0;
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(getActivity(),"Something went 
wrong",Toast.LENGTH_SHORT).show();
            }
        });
```

```
    myRef.child("liked").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(snapshot.hasChild("tacos"))
            {
                tacos_liked.setImageResource(R.drawable.heart_fill);
                tacos_count = 1;
            }
            else
            {
                tacos_liked.setImageResource(R.drawable.heart_btn);
                tacos_count = 0;
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(getActivity(),"Something went
wrong",Toast.LENGTH_SHORT).show();
        }
    });
    myRef.child("liked").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(snapshot.hasChild("carrot"))
```

```
                {
                    carrot_liked.setImageResource(R.drawable.heart_fill);
                    carrot_count = 1;
                }
                else
                {
                    carrot_liked.setImageResource(R.drawable.heart_btn);
                    carrot_count = 0;
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(getActivity(),"Something went
wrong",Toast.LENGTH_SHORT).show();
            }
        });
        myRef.child("liked").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.hasChild("coffee"))
                {
                    coffee_liked.setImageResource(R.drawable.heart_fill);
                    coffee_count = 1;
                }
        else
        {
          coffee_liked.setImageResource(R.drawable.heart_btn);
          coffee_count = 0;
        }
      }

      @Override
      public void onCancelled(@NonNull DatabaseError error) {
        Toast.makeText(getActivity(),"Something went
wrong",Toast.LENGTH_SHORT).show();
      }
    });




    pasta.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent in = new Intent(getActivity(), Pasta.class);
        startActivity(in);
      }
    });

    chilli.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View v) {
        Intent in = new Intent(getActivity(), Chilli.class);
        startActivity(in);
    }
});

tacos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in = new Intent(getActivity(), Tacos.class);
        startActivity(in);
    }
});

coffee.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in = new Intent(getActivity(), Coffee.class);
        startActivity(in);
    }
});
```

```
        pasta_liked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //pastaFunc();
                if(pasta_count==1)
                {
                    myRef.child("liked").child("pasta").removeValue();
                    Toast.makeText(getActivity(),"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
                else
                {
                    myRef.child("liked").child("pasta").child("image").
                            setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/pasta_short.png?alt=media&token=2506d650-b628-4a85-9881-
1aff538f473f");
                    Toast.makeText(getActivity(),"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
            }
        });

        chilli_liked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(chilli_count==1)
                {
                    myRef.child("liked").child("chilli").removeValue();
                    Toast.makeText(getActivity(),"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
                else
                {
                   myRef.child("liked").child("chilli").child("image").
                           setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/chilli_short.png?alt=media&token=b558e7e3-7aae-4c91-a8ea-
e85536808356");
                    Toast.makeText(getActivity(),"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
            }
        });

        tacos_liked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(tacos_count==1)
                {
                    myRef.child("liked").child("tacos").removeValue();
                    Toast.makeText(getActivity(),"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
                else
                {
                    myRef.child("liked").child("tacos").child("image").
                            setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/tacos_short.png?alt=media&token=0008a015-67b4-42f5-8c47-
09acf7c3babe");
                    Toast.makeText(getActivity(),"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
            }
        });

        carrot_liked.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(carrot_count==1)
                {
                    myRef.child("liked").child("carrot").removeValue();
                    Toast.makeText(getActivity(),"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
                else
                {
```

```
          myRef.child("liked").child("carrot").child("image").
              setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/carrot_short.png?alt=media&token=2bdb5dd6-e114-46af-ab7b-
7ed072d17d38");
          Toast.makeText(getActivity(),"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
        }
      }
    });

    coffee_liked.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(coffee_count==1)
        {
          myRef.child("liked").child("coffee").removeValue();
          Toast.makeText(getActivity(),"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
        }
        else
        {
          myRef.child("liked").child("coffee").child("image").
              setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/coffee_short.png?alt=media&token=528aaad3-8d6f-42a3-8d12-
014412cfaf8f");
```

```
}

                    Toast.makeText(getActivity(),"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
                }
            }
        });

        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                Intent in = new Intent(getActivity(),MainActivity.class);
                startActivity(in);
            }
        });

        return view;
    }

    @Override
    public void onStart() {
        super.onStart();
    }
}
```

```
package com.example.hellchef.navigation;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import com.example.hellchef.R;
import com.example.hellchef.Shopping1;
import com.ismaeldivita.chipnavigation.ChipNavigationBar;

public class ShoppinglistFragment extends Fragment {
    ImageView additem;
    ChipNavigationBar chipNavigationBar;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState){
        View view =  inflater.inflate(R.layout.activity_shoppinglist,container,false);
        additem = view.findViewById(R.id.imageView34);
        additem.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Intent in = new Intent(getActivity(), Shopping1.class);
                //startActivity(in);
                Fragment fragment = new Shopping1Fragment();
                FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.fragment_container, fragment);
                fragmentTransaction.addToBackStack(null);
                fragmentTransaction.commit();
            }
        });
        return view;
    }

    @Override
    public void onStart() {
        super.onStart();
    }
    /*
    public void additem(View view) {
        Intent in = new Intent(getActivity(), Shopping1.class);
        startActivity(in);
    }*/
}
```

```
package com.example.hellchef.navigation;


import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.Toast;


import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.ListFragment;


import com.example.hellchef.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;

public class Shopping1Fragment extends Fragment {
    ListView lv;
    EditText edit;
    ArrayList<String> items;
    ArrayAdapter<String> adapter;
    ImageView add;
    ArrayList<String> list_ing = new ArrayList<>();

    private DatabaseReference mDatabase;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState){
        View view = inflater.inflate(R.layout.activity_shopping1,container,false);
    lv = view.findViewById(R.id.listview);

    add = view.findViewById(R.id.add);

    edit = view.findViewById(R.id.editTextadd);

    items = new ArrayList<>();

    return view;

}


@Override

public void onStart() {

    super.onStart();


    add.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v)

        {

            //add code for dialog box for adding item

            String val = edit.getText().toString();

            String res;

            HashMap<String, Object> hashMap = new HashMap<String,Object>();


            if(!val.equals("")){

                //items.add(val);

                hashMap.put(val,0);
mDatabase.child("ingredients").updateChildren(hashMap).addOnSuccessListener(ne
w OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
              Toast.makeText(getActivity(), "Added to Shopping List",
Toast.LENGTH_SHORT).show();

              adapter.notifyDataSetChanged();
              //rest
            }
          }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
              Toast.makeText(getActivity(), "Something went wrong!",
Toast.LENGTH_SHORT).show();
            }
          });

          edit.setText("");
        }
        else{
          Toast.makeText(getActivity(),"Item field is
empty",Toast.LENGTH_LONG).show();

        }
```

```
            }
        });


        adapter = new ArrayAdapter<>(getActivity(),
R.layout.black_text,R.id.list_content,items);
        lv.setAdapter(adapter);
        setUpListViewListener();


        mDatabase = FirebaseDatabase.getInstance().getReference();
        mDatabase.child("ingredients").addValueEventListener(new
ValueEventListener() {


            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                items.clear();
                for(DataSnapshot ds: snapshot.getChildren())
                {
                    List<String> x = Collections.singletonList(ds.getKey());
                    for(int i=0;i<x.size();i++)
                    {
                        items.add((String) x.get(i));
                    }
                }


                adapter.notifyDataSetChanged();
            }


            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(getActivity(), "Failed to retrieve list from database",
Toast.LENGTH_SHORT).show();
            }
        });
    }



    private void setUpListViewListener() {
        lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener()
{
            @Override
            public boolean onItemLongClick(AdapterView<?> parent, View view, int i,
long l) {
                Context context = getActivity();
                String val = items.get(i);
                //Toast.makeText(context,val,Toast.LENGTH_LONG).show();
                mDatabase.child("ingredients").child(val).removeValue();
                Toast.makeText(context,"Item Removed",Toast.LENGTH_LONG).show();
                items.remove(i);
                adapter.notifyDataSetChanged();
                return true;
            }
        });
    }
}
```

```
package com.example.hellchef.navigation;

import android.Manifest;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;


import com.example.hellchef.DisplayImage;

import com.example.hellchef.Gallery;

import com.example.hellchef.R;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.OnProgressListener;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;


import java.util.UUID;


public class SmartcameraFragment extends Fragment {

    ImageView scan,gallery;

    private static final int CAMERA_REQUEST = 1888;

    private static final int MY_CAMERA_PERMISSION_CODE = 100;


    private Uri filePath;

    FirebaseStorage storage;

    StorageReference storageReference;


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState){
        View view =  inflater.inflate(R.layout.activity_smartcamera,container,false);


        scan = view.findViewById(R.id.imageView38);
        gallery = view.findViewById(R.id.imageView39);


        storage = FirebaseStorage.getInstance();
        storageReference = storage.getReference();


        return view;
    }


    @Override
    public void onStart() {
        super.onStart();


        scan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, CAMERA_REQUEST);


            }
        });
        gallery.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                /*Fragment fragment = new Shopping1Fragment();
                FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.fragment_container, fragment);
                fragmentTransaction.addToBackStack(null);
                fragmentTransaction.commit();*/
                Intent in = new Intent(getActivity(),Gallery.class);
                startActivity(in);
            }
        });
    }


    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults)
    {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == MY_CAMERA_PERMISSION_CODE)
        {
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED)
            {
                Toast.makeText(getActivity(), "camera permission granted",
Toast.LENGTH_LONG).show();
                Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, CAMERA_REQUEST);
                //uploadImage();
            }
            else
            {
                Toast.makeText(getActivity(), "camera permission denied",
Toast.LENGTH_LONG).show();
            }
        }
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == CAMERA_REQUEST && resultCode ==
Activity.RESULT_OK) {
            filePath = data.getData();
            Bitmap photo = (Bitmap) data.getExtras().get("data");
            //click_image.setImageBitmap(photo);
            Intent in = new Intent(getActivity(), DisplayImage.class);
            in.putExtra("photo",photo);
            startActivity(in);
        }
    }

    private void uploadImage() {

        if(filePath != null)
        {
            final ProgressDialog progressDialog = new ProgressDialog(getActivity());
            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            StorageReference ref = storageReference.child("Camera/"+
UUID.randomUUID().toString());
            ref.putFile(filePath)
                    .addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            progressDialog.dismiss();
                            Toast.makeText(getActivity(), "Uploaded",
Toast.LENGTH_SHORT).show();
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            progressDialog.dismiss();
                            Toast.makeText(getActivity(), "Failed "+e.getMessage(),
Toast.LENGTH_SHORT).show();
                        }
                    })
                    .addOnProgressListener(new
OnProgressListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                            double progress =
(100.0*taskSnapshot.getBytesTransferred()/taskSnapshot
                                    .getTotalByteCount());
                            progressDialog.setMessage("Uploaded "+(int)progress+"%");
                        }
                    });
        }
    }
}
```

```
package com.example.hellchef;


import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;


import android.content.Context;

import android.os.Bundle;

import android.view.MotionEvent;

import android.view.View;

import android.widget.AdapterView;

import android.widget.Toast;


import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.Query;

import com.google.firebase.database.ValueEventListener;


import java.util.ArrayList;

import java.util.Queue;
```

```
public class Liked1 extends AppCompatActivity {
    RecyclerView recyclerView;
    private DatabaseReference myRef;

    private ArrayList<Messages> messagesList;
    private RecyclerAdapter recyclerAdapter;
    private Context mContext;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_liked1);

        recyclerView = findViewById(R.id.recyclerview);
        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
        recyclerView.setLayoutManager(layoutManager);
        recyclerView.setHasFixedSize(true);

        myRef = FirebaseDatabase.getInstance().getReference();
        messagesList = new ArrayList<>();

        clearAll();

        getDataFromFirebase();
        //setUpRecycleViewListener();
    }


    private void getDataFromFirebase() {
        Query query = myRef.child("liked");
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                clearAll();
                for(DataSnapshot ds:snapshot.getChildren())
                {
                    Messages messages = new Messages();
                    messages.setImageUrl(ds.child("image").getValue().toString());


                    messagesList.add(messages);
                }
                recyclerAdapter = new
RecyclerAdapter(getApplicationContext(),messagesList);
                recyclerView.setAdapter(recyclerAdapter);
                recyclerAdapter.notifyDataSetChanged();
            }


            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                //
            }
        });
    }

    private void clearAll()
    {
        if(messagesList!=null)
        {
            messagesList.clear();
            if(recyclerAdapter!=null)
            {
                recyclerAdapter.notifyDataSetChanged();
            }
        }
        messagesList = new ArrayList<>();
    }
}
```

```
package com.example.hellchef;


import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;


import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

import android.util.Log;

import android.view.View;

import android.widget.CheckBox;

import android.widget.ImageView;

import android.widget.ScrollView;

import android.widget.TextView;

import android.widget.Toast;


import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;


import java.util.ArrayList;

import java.util.HashMap;


//import class Shopping1;


public class Eggplant extends AppCompatActivity {

    private ImageView im,step_by,btn;
    private TextView text;

    int counter = 0;

    int img_count = 0,egg_count;

    TextView x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16, x17,
x18, s;

    CheckBox c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17,c18;

    TextView v;

    String val;

    ScrollView scrollView;

    ArrayList<String> value = new ArrayList<>();


    //FirebaseDatabase database = FirebaseDatabase.getInstance();

    //DatabaseReference myRef = database.getReference("ingredients");

    private DatabaseReference mDatabase;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_eggplant);


        step_by = findViewById(R.id.imageView56);

        text = findViewById(R.id.textView63);

        scrollView = findViewById(R.id.scroll);

        btn = findViewById(R.id.imageView73);

        btn.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View v) {
        step_by.requestFocus();
        focusOnView();
    }
});

mDatabase = FirebaseDatabase.getInstance().getReference();

ImageView im = findViewById(R.id.imageView72);
mDatabase.child("liked").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if(snapshot.hasChild("eggplant"))
        {
            im.setImageResource(R.drawable.saved);
            egg_count = 1;
        }
        else
        {
            im.setImageResource(R.drawable.saved_recipe);
            egg_count = 0;
        }
    }
```

```
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(Eggplant.this,"Something went
wrong",Toast.LENGTH_SHORT).show();
            }
        });
    }

    private final void focusOnView() {
        new Handler().post(new Runnable() {
        //scrollView.post(new Runnable() {
            @Override
            public void run() {
                scrollView.smoothScrollTo(0,text.getBottom());
            }
        });
    }

    public void saved(View view) {
        if(egg_count==1)
        {
            mDatabase.child("liked").child("eggplant").removeValue();
            Toast.makeText(Eggplant.this,"Removed from Liked
Recipes",Toast.LENGTH_SHORT).show();
```

```
        }
        else
        {
            mDatabase.child("liked").child("eggplant").child("image").
                    setValue("https://firebasestorage.googleapis.com/v0/b/hellchef-
28282.appspot.com/o/sr1.png?alt=media&token=4bef9cb3-afba-43f6-ace7-
1f1011a6dc64");
            Toast.makeText(Eggplant.this,"Added to Liked
Recipes",Toast.LENGTH_SHORT).show();
        }
    }

    public void back(View view) {
        Intent in = new Intent(this, Smartrecipe1.class);
        startActivity(in);
    }

    public void shopping(View view) {
        ImageView img = (ImageView) findViewById(R.id.imageView55);
        //img.setImageResource(R.drawable.shop_clicked);
        //Shopping1 sp = new Shopping1();



        c1 = findViewById(R.id.checkBox102);
        c2 = findViewById(R.id.checkBox105);
```

```
c3 = findViewById(R.id.checkBox108);

c4 = findViewById(R.id.checkBox111);

c5 = findViewById(R.id.checkBox114);

c6 = findViewById(R.id.checkBox117);

c7 = findViewById(R.id.checkBox120);

c8 = findViewById(R.id.checkBox123);

c9 = findViewById(R.id.checkBox126);

c10 = findViewById(R.id.checkBox129);

c11 = findViewById(R.id.checkBox132);

c12 = findViewById(R.id.checkBox135);

c13 = findViewById(R.id.checkBox138);

c14 = findViewById(R.id.checkBox141);

c15 = findViewById(R.id.checkBox144);

c16 = findViewById(R.id.checkBox147);

c17 = findViewById(R.id.checkBox150);

c18 = findViewById(R.id.checkBox153);


if(c1.isChecked())

{

    v = findViewById(R.id.textView101);

    val = v.getText().toString();

    value.add(val);

}

if(c2.isChecked())
```

```
{
   v = findViewById(R.id.textView104);
   val = v.getText().toString();
   value.add(val);
}
if(c3.isChecked())
{
   v = findViewById(R.id.textView107);
   val = v.getText().toString();
   value.add(val);
}
if(c4.isChecked())
{
   v = findViewById(R.id.textView110);
   val = v.getText().toString();
   value.add(val);
}
if(c5.isChecked())
{
   v = findViewById(R.id.textView113);
   val = v.getText().toString();
   value.add(val);
}
if(c6.isChecked())
```

```
{
    v = findViewById(R.id.textView116);
    val = v.getText().toString();
    value.add(val);
}
if(c7.isChecked())
{
    v = findViewById(R.id.textView119);
    val = v.getText().toString();
    value.add(val);
}
if(c8.isChecked())
{
    v = findViewById(R.id.textView122);
    val = v.getText().toString();
    value.add(val);
}
if(c9.isChecked())
{
    v = findViewById(R.id.textView125);
    val = v.getText().toString();
    value.add(val);
}
if(c10.isChecked())
```

```
{
   v = findViewById(R.id.textView128);
   val = v.getText().toString();
   value.add(val);
}
if(c11.isChecked())
{
   v = findViewById(R.id.textView131);
   val = v.getText().toString();
   value.add(val);
}
if(c12.isChecked())
{
   v = findViewById(R.id.textView134);
   val = v.getText().toString();
   value.add(val);
}
if(c13.isChecked())
{
   v = findViewById(R.id.textView137);
   val = v.getText().toString();
   value.add(val);
}
if(c14.isChecked())
```

```
{
   v = findViewById(R.id.textView140);
   val = v.getText().toString();
   value.add(val);
}
if(c15.isChecked())
{
   v = findViewById(R.id.textView143);
   val = v.getText().toString();
   value.add(val);
}
if(c16.isChecked())
{
   v = findViewById(R.id.textView146);
   val = v.getText().toString();
   value.add(val);
}
if(c17.isChecked())
{
   v = findViewById(R.id.textView149);
   val = v.getText().toString();
   value.add(val);
}
if(c18.isChecked())
```

```
        {
            v = findViewById(R.id.textView152);
            val = v.getText().toString();
            value.add(val);
        }

        HashMap hashMap = new HashMap();
        for(int i=0;i<value.size();i++) {
            //String index = String.valueOf(i);
            String index = String.valueOf(0);
            String val = value.get(i);
            hashMap.put(val,index);
            //Ingredients in = new Ingredients(val,index);
            //mDatabase.child("ingredients").setValue(in);
            //myRef.setValue(a);
            //Log.d("list",a);
        }

        mDatabase.child("ingredients").setValue(hashMap).addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                img.setImageResource(R.drawable.shop_clicked);
                Toast.makeText(getApplicationContext(), "Added to Shopping List",
Toast.LENGTH_SHORT).show();
```

```
                //rest
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(getApplicationContext(), "Something went wrong!",
Toast.LENGTH_SHORT).show();
            }
        });

        //Intent in = new Intent(Eggplant.this,Shopping1.class);
        //startActivity(in);



    }

    public void increment(View view) {
        if(counter<=3) {
            counter++;
            display(counter);
        }

    }

    public void decrement(View view) {
```

```
    if(counter>0) {
        counter--;
        display(counter);
    }
}

public void display(int counter) {
    switch (counter)
    {
        case 0:
            s = (TextView) findViewById(R.id.textView62);
            s.setText("Serves 1");
            x1 = (TextView) findViewById(R.id.textView36);
            x1.setText("1/2");
            x2 = (TextView) findViewById(R.id.textView42);
            x2.setText("1 oz");
            x3 = (TextView) findViewById(R.id.textView44);
            x3.setText("1 1/8 tsp");
            x4 = (TextView) findViewById(R.id.textView46);
            x4.setText("1/4 clove");
            x5 = (TextView) findViewById(R.id.textView50);
            x5.setText("1");
            x6 = (TextView) findViewById(R.id.textView52);
            x6.setText("3/4 tsp");
```

```
x7 = (TextView) findViewById(R.id.textView54);
x7.setText("1 1/4");
x8 = (TextView) findViewById(R.id.textView56);
x8.setText("1/2 Tbsp");
x9 = (TextView) findViewById(R.id.textView58);
x9.setText("1/2 tsp");
x10 = (TextView) findViewById(R.id.textView60);
x10.setText("1 Tbsp");
x11 = (TextView) findViewById(R.id.textView36);
x11.setText("3/4 tsp");
x12 = (TextView) findViewById(R.id.textView42);
x12.setText("1/4 cup");
x13 = (TextView) findViewById(R.id.textView44);
x13.setText("1/8 tsp");
x14 = (TextView) findViewById(R.id.textView46);
x14.setText("1/2 Tbsp");
x15 = (TextView) findViewById(R.id.textView50);
x15.setText("1/4 tsp");
x16 = (TextView) findViewById(R.id.textView52);
x16.setText("1/4 tsp");
x17 = (TextView) findViewById(R.id.textView54);
x17.setText("1 1/2 cups");
x18 = (TextView) findViewById(R.id.textView56);
x18.setText("1/2 Tbsp");
```

```
    break;
case 2:
    s = (TextView) findViewById(R.id.textView62);
    s.setText("Serves 2");
    x1 = (TextView) findViewById(R.id.textView36);
    x1.setText("1");
    x2 = (TextView) findViewById(R.id.textView42);
    x2.setText("2 oz");
    x3 = (TextView) findViewById(R.id.textView44);
    x3.setText("1/2 Tbsp");
    x4 = (TextView) findViewById(R.id.textView46);
    x4.setText("1/2 clove");
    x5 = (TextView) findViewById(R.id.textView50);
    x5.setText("2");
    x6 = (TextView) findViewById(R.id.textView52);
    x6.setText("1/2 Tbsp");
    x7 = (TextView) findViewById(R.id.textView54);
    x7.setText("2 1/2");
    x8 = (TextView) findViewById(R.id.textView56);
    x8.setText("1 Tbsp");
    x9 = (TextView) findViewById(R.id.textView58);
    x9.setText("1 tsp");
    x10 = (TextView) findViewById(R.id.textView60);
    x10.setText("2 Tbsp");
    x11 = (TextView) findViewById(R.id.textView36);
    x11.setText("1/2 Tbsp");
    x12 = (TextView) findViewById(R.id.textView42);
    x12.setText("1/2 cup");
    x13 = (TextView) findViewById(R.id.textView44);
    x13.setText("1/4 tsp");
    x14 = (TextView) findViewById(R.id.textView46);
    x14.setText("1 Tbsp");
    x15 = (TextView) findViewById(R.id.textView50);
    x15.setText("1/2 tsp");
    x16 = (TextView) findViewById(R.id.textView52);
    x16.setText("1/2 tsp");
    x17 = (TextView) findViewById(R.id.textView54);
    x17.setText("3 cups");
    x18 = (TextView) findViewById(R.id.textView56);
    x18.setText("1 Tbsp");
    break;
case 3:
    s = (TextView) findViewById(R.id.textView62);
    s.setText("Serves 4");
    x1 = (TextView) findViewById(R.id.textView36);
    x1.setText("2");
    x2 = (TextView) findViewById(R.id.textView42);
    x2.setText("3.5 oz");
```

```
x3 = (TextView) findViewById(R.id.textView44);

x3.setText("1 1/2 Tbsp");

x4 = (TextView) findViewById(R.id.textView46);

x4.setText("1 clove");

x5 = (TextView) findViewById(R.id.textView50);

x5.setText("4");

x6 = (TextView) findViewById(R.id.textView52);

x6.setText("1 Tbsp");

x7 = (TextView) findViewById(R.id.textView54);

x7.setText("5");

x8 = (TextView) findViewById(R.id.textView56);

x8.setText("2 Tbsp");

x9 = (TextView) findViewById(R.id.textView58);

x9.setText("1/2 Tbsp");

x10 = (TextView) findViewById(R.id.textView60);

x10.setText("1/4 cup");

x11 = (TextView) findViewById(R.id.textView36);

x11.setText("1 Tbsp");

x12 = (TextView) findViewById(R.id.textView42);

x12.setText("1 cup");

x13 = (TextView) findViewById(R.id.textView44);

x13.setText("1/2 tsp");

x14 = (TextView) findViewById(R.id.textView46);

x14.setText("2 Tbsp");
```

```
    x15 = (TextView) findViewById(R.id.textView50);
    x15.setText("1 tsp");
    x16 = (TextView) findViewById(R.id.textView52);
    x16.setText("1 tsp");
    x17 = (TextView) findViewById(R.id.textView54);
    x17.setText("6 cups");
    x18 = (TextView) findViewById(R.id.textView56);
    x18.setText("2 Tbsp");
    break;
case 4:
    s = (TextView) findViewById(R.id.textView62);
    s.setText("Serves 6");
    x1 = (TextView) findViewById(R.id.textView36);
    x1.setText("3");
    x2 = (TextView) findViewById(R.id.textView42);
    x2.setText("5.5 oz");
    x3 = (TextView) findViewById(R.id.textView44);
    x3.setText("2 Tbsp");
    x4 = (TextView) findViewById(R.id.textView46);
    x4.setText("1 1/2 cloves");
    x5 = (TextView) findViewById(R.id.textView50);
    x5.setText("6");
    x6 = (TextView) findViewById(R.id.textView52);
    x6.setText("1 1/2 Tbsp");
```

```
x7 = (TextView) findViewById(R.id.textView54);

x7.setText("7 1/2");

x8 = (TextView) findViewById(R.id.textView56);

x8.setText("3 Tbsp");

x9 = (TextView) findViewById(R.id.textView58);

x9.setText("1 Tbsp");

x10 = (TextView) findViewById(R.id.textView60);

x10.setText("1/3 cup");

x11 = (TextView) findViewById(R.id.textView36);

x11.setText("1 1/2 Tbsp");

x12 = (TextView) findViewById(R.id.textView42);

x12.setText("1 1/2 cups");

x13 = (TextView) findViewById(R.id.textView44);

x13.setText("3/4 tsp");

x14 = (TextView) findViewById(R.id.textView46);

x14.setText("3 Tbsp");

x15 = (TextView) findViewById(R.id.textView50);

x15.setText("1/2 Tbsp");

x16 = (TextView) findViewById(R.id.textView52);

x16.setText("1/2 Tbsp");

x17 = (TextView) findViewById(R.id.textView54);

x17.setText("9 cups");

x18 = (TextView) findViewById(R.id.textView56);

x18.setText("3 Tbsp");
```

```
                break;
            }


        }
    }
```

Machine Learning code

```
# define helper functions
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline

  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image,(3*width, 3*height), interpolation =
cv2.INTER_CUBIC)

  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()

# use this to upload files
def upload():
  from google.colab import files
  uploaded = files.upload()
  for name, data in uploaded.items():
    with open(name, 'wb') as f:
      f.write(data)
      print ('saved file', name)

# use this to download a file
def download(path):
  from google.colab import files
  files.download(path)
```

6.6 OUTPUT

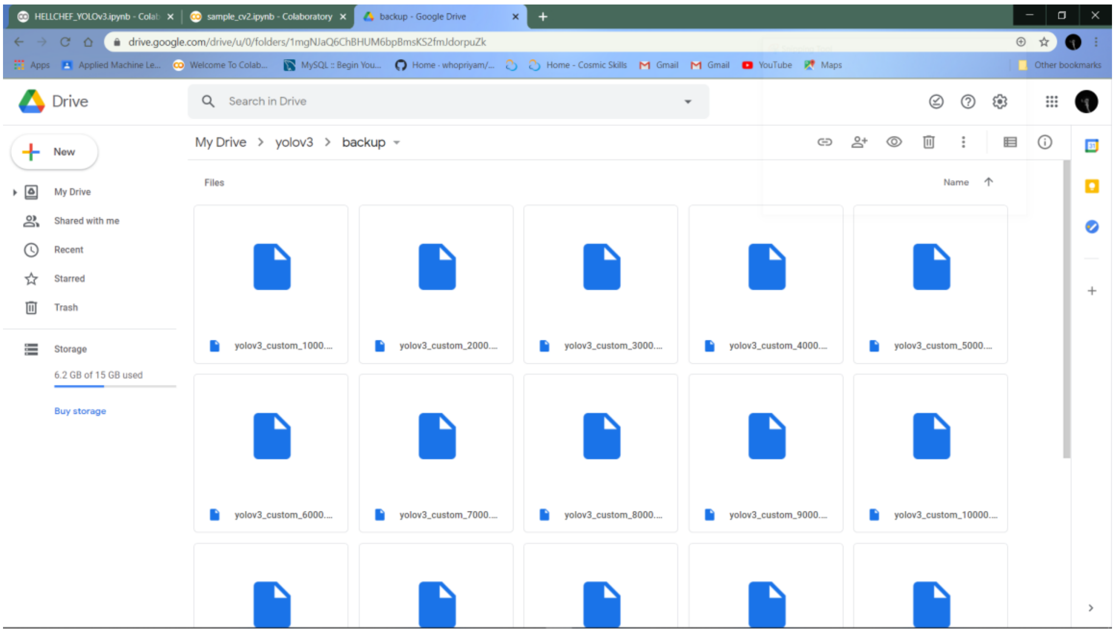
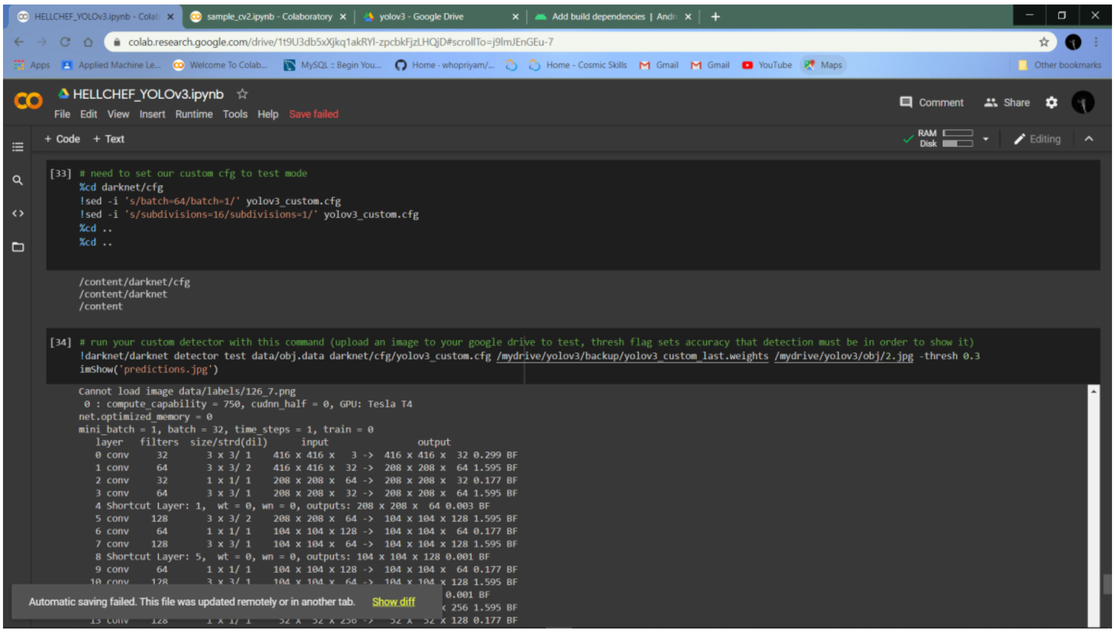


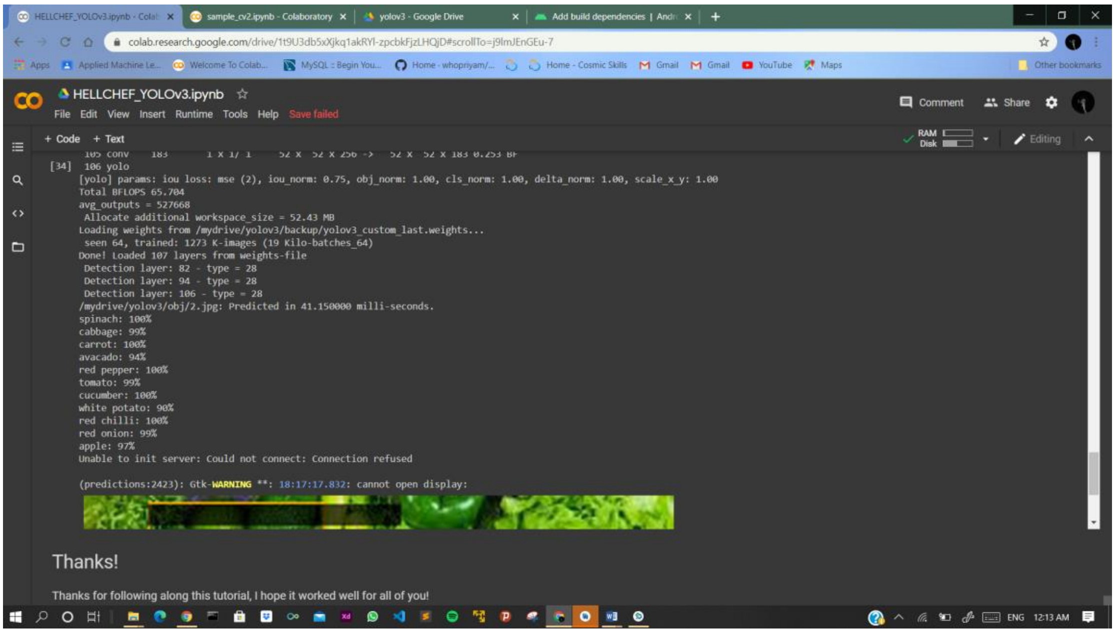
Figure 1.11 YOLO weights



```
[33] # need to set our custom cfg to test mode
     %cd darknet/cfg
     !sed -i 's/batch=64/batch=1/' yolov3_custom.cfg
     !sed -i 's/subdivisions=16/subdivisions=1/' yolov3_custom.cfg
     %cd ..
     %cd ..

     /content/darknet/cfg
     /content/darknet
     /content
```

```
[34] # run your custom detector with this command (upload an image to your google drive to test, thresh flag sets accuracy that detection must be in order to show it)
     !darknet/darknet detector test data/obj.data darknet/cfg/yolov3_custom.cfg /mydrive/yolov3/backup/yolov3_custom_last.weights /mydrive/yolov3/obj/2.jpg -thresh 0.3
     imShow('predictions.jpg')

     Cannot load image data/labels/126_7.png
      0 : compute_capability = 750, cudnn_half = 0, GPU: Tesla T4
     net.optimized_memory = 0
     mini_batch = 1, batch = 32, time_steps = 1, train = 0
        layer   filters  size/strd(dil)      input                output
        0 conv     32       3 x 3/ 1    416 x 416 x   3 ->  416 x 416 x  32 0.299 BF
        1 conv     64       3 x 3/ 2    416 x 416 x  32 ->  208 x 208 x  64 1.595 BF
        2 conv     32       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  32 0.177 BF
        3 conv     64       3 x 3/ 1    208 x 208 x  32 ->  208 x 208 x  64 1.595 BF
        4 Shortcut Layer: 1,  wt = 0, wn = 0, outputs: 208 x 208 x  64 0.003 BF
        5 conv    128       3 x 3/ 2    208 x 208 x  64 ->  104 x 104 x 128 1.595 BF
        6 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
        7 conv    128       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x 128 1.595 BF
        8 Shortcut Layer: 5,  wt = 0, wn = 0, outputs: 104 x 104 x 128 0.001 BF
        9 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
       10 conv    128       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x 128 1.595 BF
                                                                        0.001 BF
                                                                      x 256 1.595 BF
       13 conv    128       1 x 1/ 1     52 x  52 x 256 ->   52 x  52 x 128 0.177 BF
```

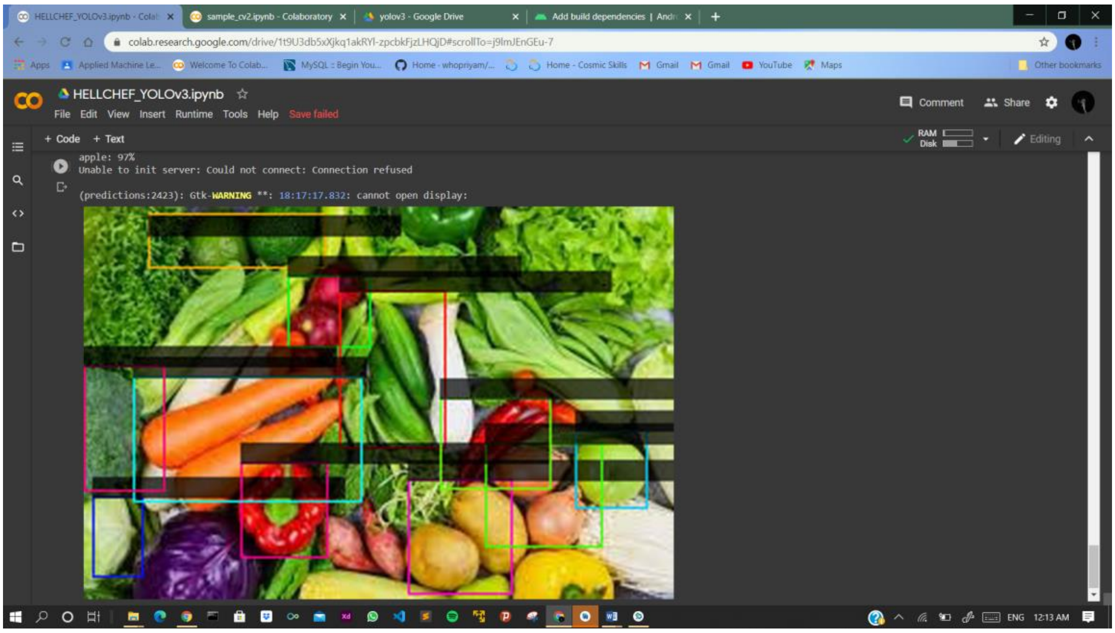
Automatic saving failed. This file was updated remotely or in another tab. Show diff



```
      105 conv    183       1 x 1/ 1     52 x  52 x 256 ->   52 x  52 x 183 0.253 BF
[34]  106 yolo
     [yolo] params: iou loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
     Total BFLOPS 65.704
     avg_outputs = 527668
      Allocate additional workspace_size = 52.43 MB
     Loading weights from /mydrive/yolov3/backup/yolov3_custom_last.weights...
      seen 64, trained: 1273 K-images (19 Kilo-batches_64)
     Done! Loaded 107 layers from weights-file
      Detection layer: 82 - type = 28
      Detection layer: 94 - type = 28
      Detection layer: 106 - type = 28
     /mydrive/yolov3/obj/2.jpg: Predicted in 41.150000 milli-seconds.
     spinach: 100%
     cabbage: 99%
     carrot: 100%
     avacado: 94%
     red pepper: 100%
     tomato: 99%
     cucumber: 100%
     white potato: 90%
     red chilli: 100%
     red onion: 99%
     apple: 97%
     Unable to init server: Could not connect: Connection refused

     (predictions:2423): Gtk-WARNING **: 18:17:17.832: cannot open display:
```

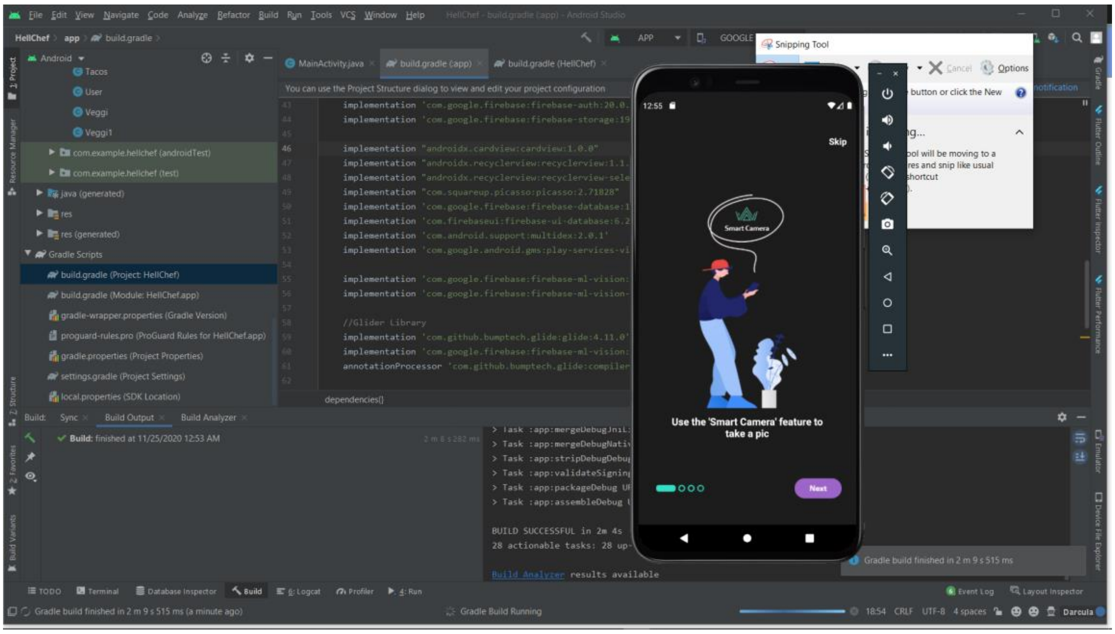
Thanks!

Thanks for following along this tutorial, I hope it worked well for all of you!



```
apple: 97%
Unable to init server: Could not connect: Connection refused

(predictions:2423): Gtk-WARNING **: 18:17:17.832: cannot open display:
```

Figure 1.12 Project Output



You can use the Project Structure dialog to view and edit your project configuration

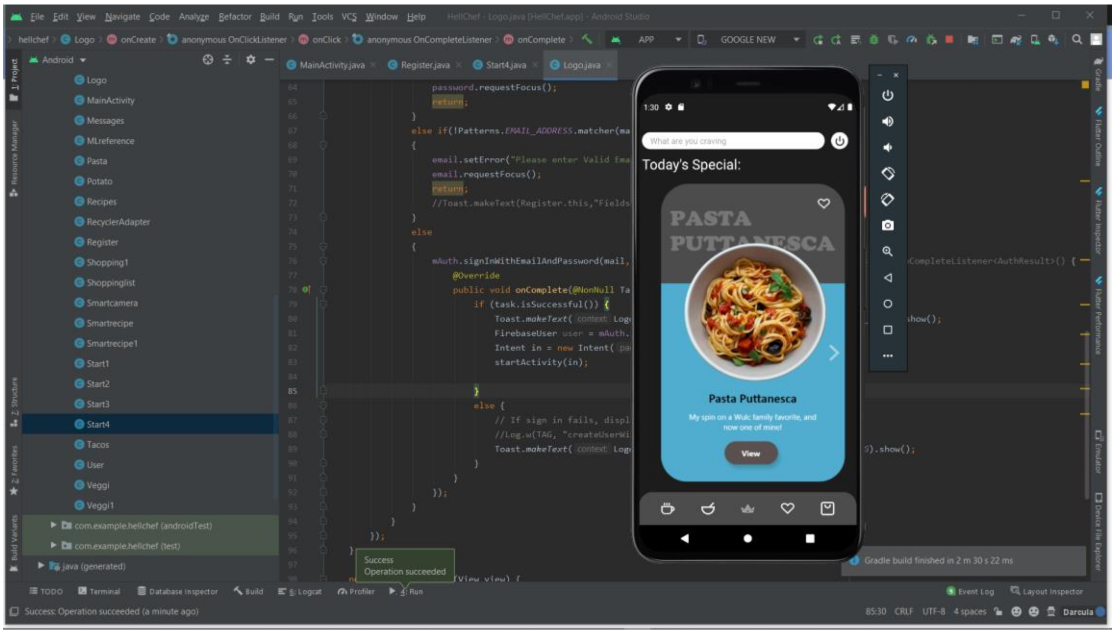
```
    implementation 'com.google.firebase:firebase-auth:20.0.
    implementation 'com.google.firebase:firebase-storage:19

    implementation "androidx.cardview:cardview:1.0.0"
    implementation "androidx.recyclerview:recyclerview:1.1.
    implementation "androidx.recyclerview:recyclerview-sele
    implementation "com.squareup.picasso:picasso:2.71828"
    implementation 'com.google.firebase:firebase-database:1
    implementation 'com.firebaseui:firebase-ui-database:6.2
    implementation 'com.android.support:multidex:2.0.1'
    implementation 'com.google.android.gms:play-services-vi

    implementation 'com.google.firebase:firebase-ml-vision:
    implementation 'com.google.firebase:firebase-ml-vision-

    //Glider Library
    implementation 'com.github.bumptech.glide:glide:4.11.0'
    implementation 'com.google.firebase:firebase-ml-vision:
    annotationProcessor 'com.github.bumptech.glide:compiler
```

Build: finished at 11/25/2020 12:53 AM

```
> Task :app:mergeDebugJniL
> Task :app:mergeDebugNati
> Task :app:stripDebugDebu
> Task :app:validateSignin
> Task :app:packageDebug U
> Task :app:assembleDebug

BUILD SUCCESSFUL in 2m 4s
28 actionable tasks: 28 up-

Build Analyzer results available
```

Gradle build finished in 2 m 9 s 515 ms (a minute ago)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HellChef - build.gradle (app) - Android Studio
MainActivity.java
build.gradle (app)
build.gradle (HellChef)
Tacos
User
Veggi
Veggi1
com.example.hellchef (androidTest)
com.example.hellchef (test)
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: HellChef)
build.gradle (Module: HellChef.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for HellChef.app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Snipping Tool
12:55
Skip
Smart Camera
Use the 'Smart Camera' feature to take a pic
Next
Build Output
Build Analyzer
Gradle Build Running

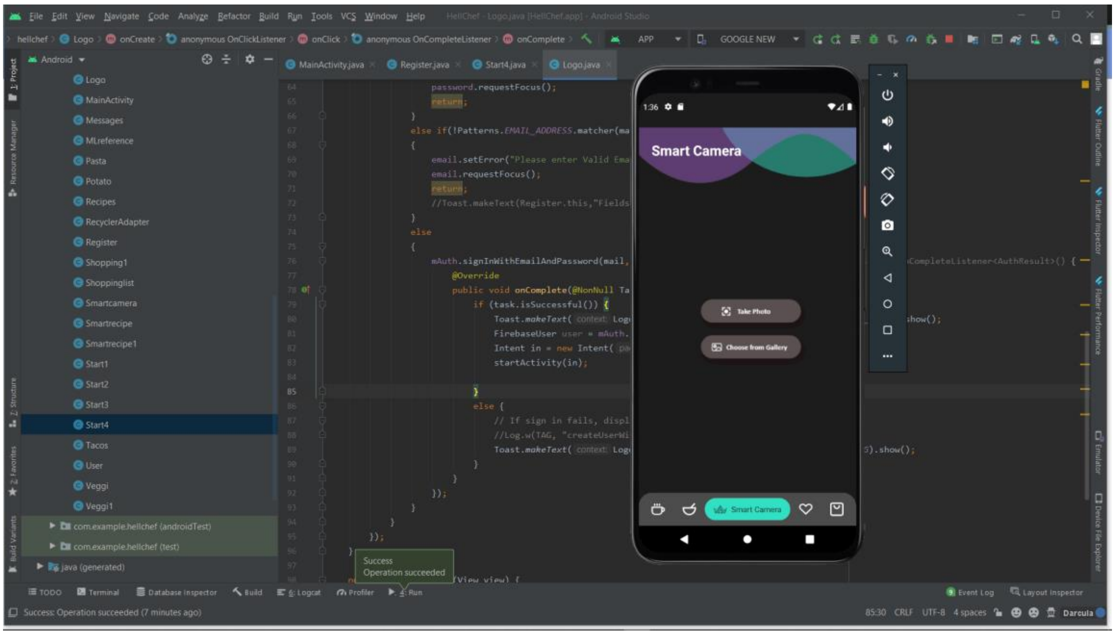


```
                password.requestFocus();
                return;
            }
            else if(!Patterns.EMAIL_ADDRESS.matcher(ma
            {
                email.setError("Please enter Valid Ema
                email.requestFocus();
                return;
                //Toast.makeText(Register.this,"Fields
            }
            else
            {
                mAuth.signInWithEmailAndPassword(mail,
                    @Override
                    public void onComplete(@NonNull Ta
                        if (task.isSuccessful()) {
                            Toast.makeText( context: Log
                            FirebaseUser user = mAuth.
                            Intent in = new Intent( pa
                            startActivity(in);

                        }
                        else {
                            // If sign in fails, displ
                            //Log.w(TAG, "createUserWi
                            Toast.makeText( context: Log
                        }
                    }
                });
            }
        }
    });
}
```

Success
Operation succeeded

Gradle build finished in 2 m 30 s 22 ms

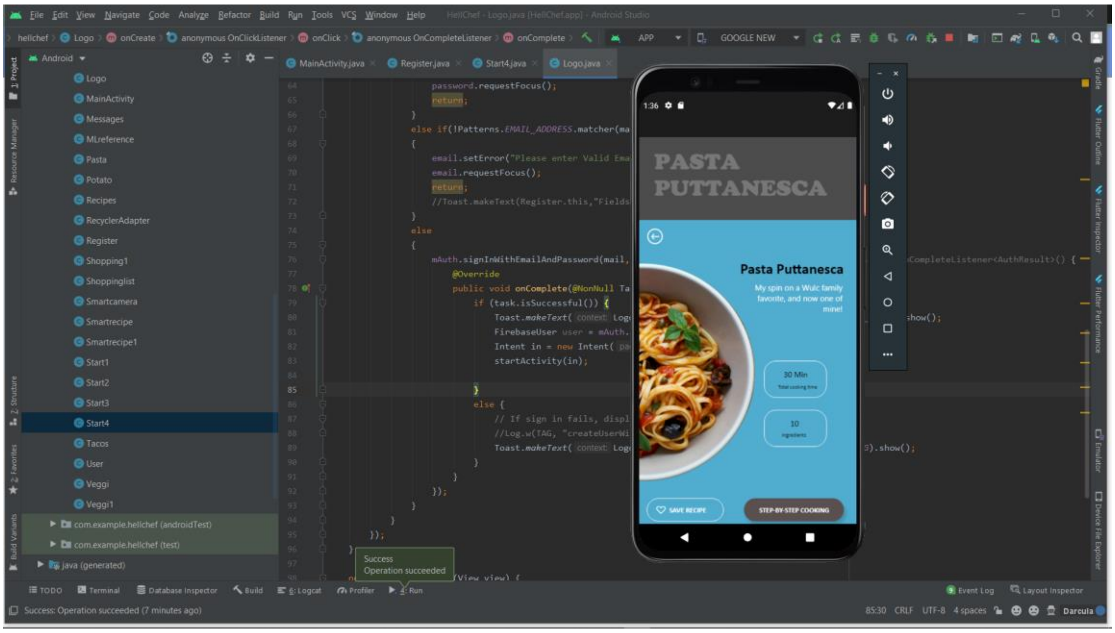
Success: Operation succeeded (a minute ago)


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HellChef - Logo.java [HellChef.app] - Android Studio
hellchef › Logo › onCreate › anonymous OnClickListener › onClick › anonymous OnCompleteListener › onComplete
APP
GOOGLE NEW
Android
MainActivity.java
Register.java
Start4.java
Logo.java
1:36
Smart Camera
Take Photo
Choose from Gallery
Smart Camera
85:30 CRLF UTF-8 4 spaces
Darcula

```
                password.requestFocus();
                return;
            }
            else if(!Patterns.EMAIL_ADDRESS.matcher(ma
            {
                email.setError("Please enter Valid Ema
                email.requestFocus();
                return;
                //Toast.makeText(Register.this,"Fields
            }
            else
            {
                mAuth.signInWithEmailAndPassword(mail,
                    @Override
                    public void onComplete(@NonNull Ta
                        if (task.isSuccessful()) {
                            Toast.makeText( context: Log
                            FirebaseUser user = mAuth.
                            Intent in = new Intent( pa
                            startActivity(in);

                        }
                        else {
                            // If sign in fails, displ
                            //Log.w(TAG, "createUserWi
                            Toast.makeText( context: Log
                        }
                    }
                });
            }
        }
    });
}
```

Success
Operation succeeded

Success: Operation succeeded (7 minutes ago)



```
                password.requestFocus();
                return;
            }
            else if(!Patterns.EMAIL_ADDRESS.matcher(ma
            {
                email.setError("Please enter Valid Ema
                email.requestFocus();
                return;
                //Toast.makeText(Register.this,"Fields
            }
            else
            {
                mAuth.signInWithEmailAndPassword(mail,                    CompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Ta
                        if (task.isSuccessful()) {
                            Toast.makeText( context: Log                      show();
                            FirebaseUser user = mAuth.
                            Intent in = new Intent( pa
                            startActivity(in);

                        }
                        else {
                            // If sign in fails, displ
                            //Log.w(TAG, "createUserWi
                            Toast.makeText( context: Log                 S).show();
                        }
                    }
                });
            }
        }
    });
}
                                     (View view) {
```

Success
Operation succeeded

Pasta Puttanesca

My spin on a Wulc family favorite, and now one of mine!

Success: Operation succeeded (7 minutes ago)

Figure 1.13 Android Output

Chapter 7

Conclusion and future work

Here we made an app that takes the image of the vegetables and provide the recipes from the ingredients we have. This is done by training the model with the custom images. The application may look simple but the integration of the application to firebase and training the model and getting much accuracy is not easy.

There is many opportunities using this application. We can train the custom models and use them to identify the images. We can use these images to identify the faces and other stuff.

. A LSTM NN model has been proposed to forecast the commuter flow in a transit hub. LSTM neural networks are the most efficient neural networks, learns time sequence data in long time reliance .This network is applied to give an adaptable structure for many time series dependent problems. To confirm the viability of the proposed LSTM NN based model. Test results of proposed LSTM NN model shows that MAPE and RMSE have the lowest values. This demonstrates that the model can accomplish better versatility and higher precision.

Chapter 9

References

1. Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi(2016). XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. 2016 ECCV
2. Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. 2018 CVPR
3. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg.SSD: Single Shot Multi Box Detector. 2016 ECCV
4. Joseph Redmon, Ali Farhadi(2017). YOLO9000: Better, Faster, Stronger. 2017 CVPR
5. Mester, Jessica L et al. Analysis of Prevalence and Degree of Macrocephaly in Patients with GermlinePTENMutations and of Brain Weight inPtenKnock-in Murine Model.European Journal of Human Genetics19.7 (2011): 763768.PMC. Web. 30 May 2017.
6. Huang et al, Speed/accuracy trade-offs for modern convolutional object detectors, CVPR 2017.