Mylio

Technical Guide

By Robin O’Shea
Student number: 16369296
Supervisor: Dr.Michael Scriney

Table of Contents

1. Introduction
1.1 Motivation
1.2 Overview
1.3 Business Context
1.4 Glossary
2. Research
3. General Description
3.1 System Functions
3.2 User characteristics
4. System Architecture
4.1 General System Architecture
4.2 Architecture Overview Diagram
5. High level Design
5.1 Description
5.2 Sequence Diagram
6. Implementation

1. Introduction

1.1 Motivation

The aim is to identify and provide information on receipts to users by submitting an image of a receipt into the mobile application.The overall goal of the project attempts to remove the manual process of searching for recipes that he/she can make a while also keeping track of a user's inventory. With such ambiguity around what to eat or what meals to prepare for the week , This projects attempts to capitalize in these areas

Smartphone devices continue to excel in popularity and processing power and the continued push for faster and more reliable information, this application looks to combine these modern advances by using all features of a smartphone. The device should be accessible to people of all ages and allow users to freely capture data from their receipts and acquire information such as actual product names and recipes they can make

1.2 Overview

The project is an image recognition mobile application that will enable the user to submit an image of a receipts via taking a picture from their device.The application will then identify the name of the store to which the receipt belongs and the corresponding products that were purchased by the user .The application will then make a decision of what recipes the user can make based off the products that they own takin from scanning the receipt.The systems design is constructed by obtaining data on grocery food produce within a plethora of grocery stores to choose between to compare against tokens.These tokens being created from image recognition. Mylio is an ambitious idea and for this reason, upon delivery Mylio is operational for four major supermarkets. Tesco, Aldi, Supervalu and Iceland with the opportunity for future expansion in other known supermarkets. The application will be simple in design and cater to people of all ages. It will provide indecisive users of what meals to prepare accordingly

1.3 Business Context

The product can be an independent service provider, allowing companies to influence their own produce while prescribing to the user of what they can prepare .Mylio has the potential for covering a larger variety of receipts from different stores

1.4 Glossary

- **Selenium** - portable framework that provides the tools necessary for web scraping
- **BeatifulSoup** - Python package for parsing HTML and XML documents
- **Sqlite3** - relational database management system that is non client server database engine, it is portable and can be stored locally
- **Flask** - Web framework allowing communication services to android application
- **SQLAlchemy** - toolkit and object relational mapper, used with sqlite to perform less operations
- **Ontology** - core conceptual view of an object
- **Ngrok** - tunnel network allowing of private network connections to be sent across a public network
- **Retrofit** - api that allows requests to be made to given url
- **Firebase** - Mobile and web application development platform
- **FireStore** - scalable database for mobile development
- **Google visions api** - apo that offers powerful and pretrained machine learning models through REST and RPC APIs
- **OCR** - optical character recognition tool, used for conversion of image containing identifiable strings or characters into actual text

2.Research

The main functionalities of the mobile application is based around text recognition and a classification feature from these produced characters. For this reason I researched into optical character recognition tools and how they may be implemented. Making a conclusion based off the accuracy of results taken from an image captured against estimated total cost of implementation
My final decision was that android studio facilitates its own google visions api for character recognition my needs that worked well for what i was trying to achieve

To develop a classification of the produced text against recipes I also needed a solidified database from the corresponding stores. Each Store table is used as a reference to the identity of a product against text. Product names of receipts are very much abbreviations and not the same as actual product names so
research was necessary into string similarity algorithms to match the text to their actual product name counterpart

it was found that Jaro winklers edit distance similarity algorithm was most fitting to my circumstances and is currently implemented as a placeholder until my own string similarity algorithm can be produced and implemented to further improve the accuracy of results. Each product than had to be related to a recipe ingredient and so a mapping needed to be instanced against each product.Further work was done into researching ontologies and how they may work
I also decided to use Flask web framework as the main communication service between the database and the application and so research was necessary into how to setup a flask server and to put it online.

For allowing for communication between flask and my application i researched into Tunnel networking services and used ngrok to rent a public url pointing to my localhost web framework

3. General Description

3.1 System Functions

The mobile application was developed for indecisive users who don't know what to prepare for lunch, dinner or breakfast as well as users who find it hard to make records on products they own. One functionality of the application is to identify the store to which a receipt belongs and return actual product names instead of abbreviations, storing the products into the user's repository

The user interface of the app aims to be simplistic and be user friendly for all audiences.The application includes options to take a picture directly or access the camera roll and select an image from there. Once an image is selected and cropped accordingly. It will be passed through the optical character recognition tool and the title of the receipt will be extracted. From this information, the application will then decide on the parser to be used against the text. Once products are successfully parsed and returned. The application will contact the flask server api to use the string similarity algorithm of the text against the product data in the database

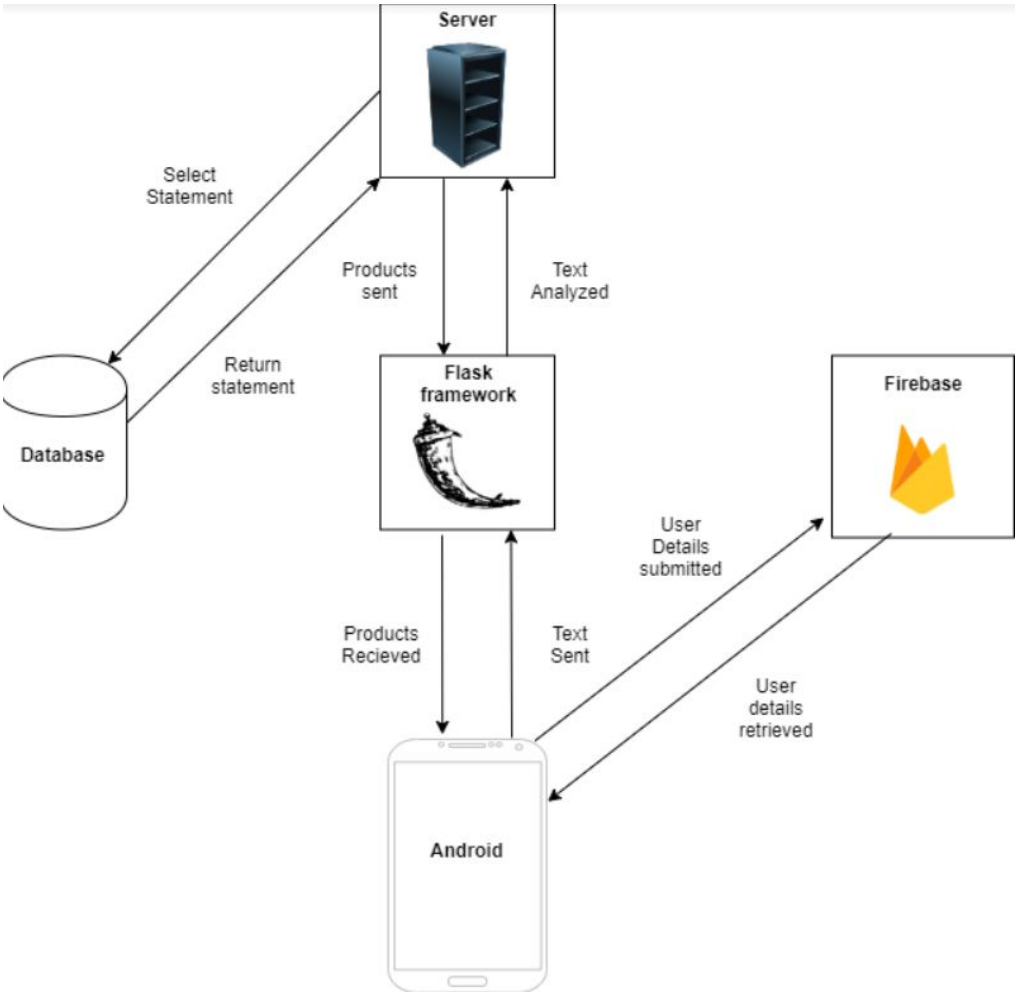
Another feature of the application will be to dynamically prescribing ten recipes based on what the users has in his/her repository.Once again the application communicates back to the host server to discern the ontologies of each product and extract a select statement of recipes containing those ontologies , these recipes are returned in json format storing the recipe name , recipe ingredients and recipe instructions

3.2 User characteristics

A large amount of the user base on the mobile application will be those that wish to get information about new recipes that may be of interest to them while keeping a record of what food they have at home.While also covering users that may have difficulty in deciding on what meals to make or don't know what ingredients they currently own and need help resolving this situation. With these scenarios, the application is clear to users of all ages and adaptable to expanding to more available grocery stores. Giving the user the ability to screenshot an image and use it or select an image previous history within their camera roll

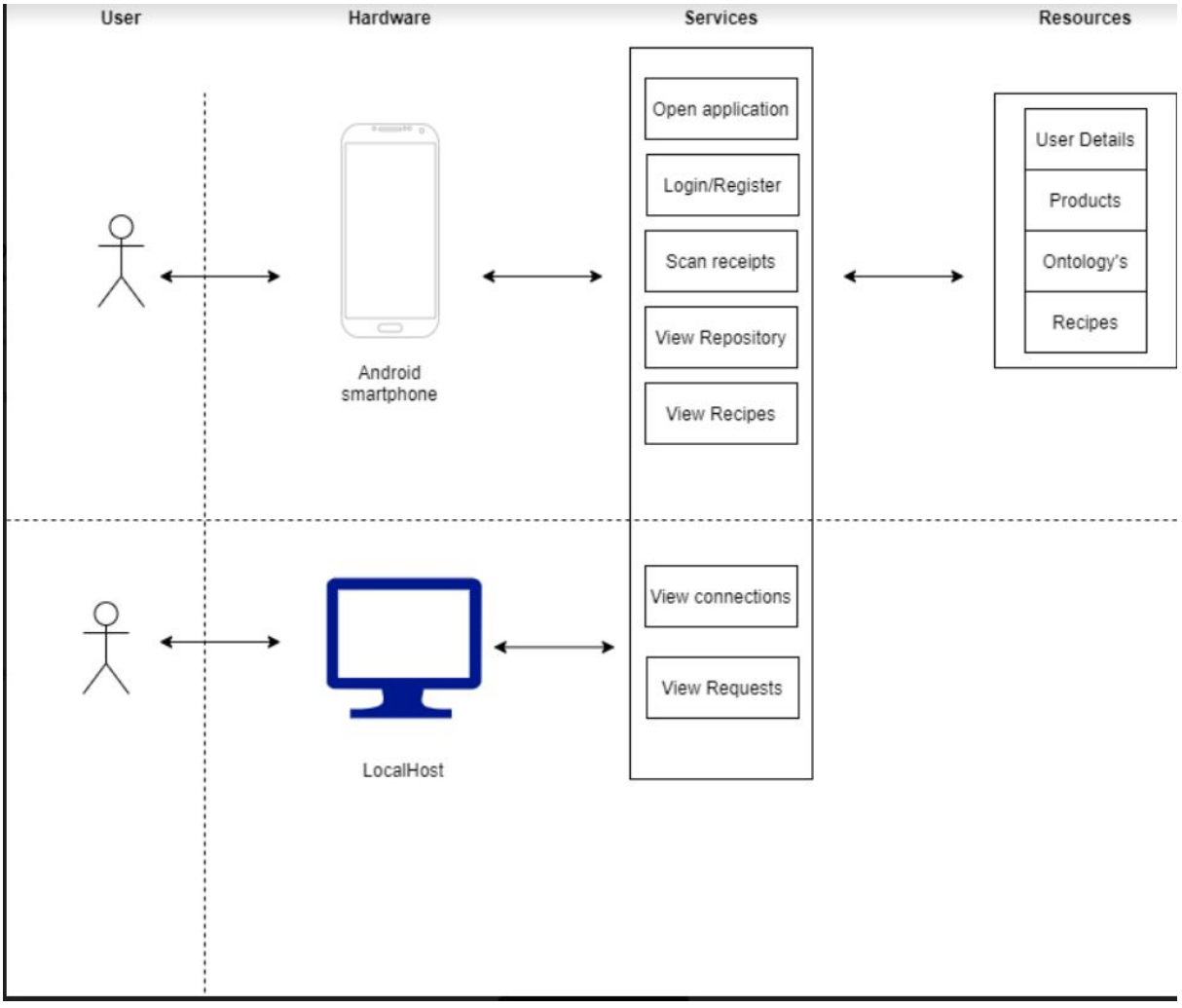
System Architecture

4.1 General System Architecture



The system is built using a communication layer between the framework and the phone. Users will require an android device in order to use this mobile application. Upon capturing an image the perceived text is sent directly to the flask framework to identify the store in which the receipt belongs, Once the store id is received(i.e “Supervalu”) the application will have its own parsers in handling respective stores, The text will than be parsed, filtering out irrelevant strings such as prices, until only a string of products remain. These products are then sent back to the framework to discern actual product id’s. Once retrieved, the users repositories are updated within firebase's firestore document collections, each document being respective of the users id.
The application also includes a login and registration feature. Both of which are authenticated and loaded using firebase.

4.2 Architecture Overview Diagram



The above diagram is an architecture overview diagram for the mobile application, that illustrate at a conceptual level, an understanding of the system based on the users using the application, the delivery channel, services that are provided and external resources that interact with the services inside the architecture

User
The user is anybody who wishes to use the app for their needs of getting information regarding recipes they can make and products that they have purchased. They have one delivery channel to gain access to the app and that is through their android device as shown in the above

Admin
Users that have the authorization to make any changes to the system. This user interacts with the system

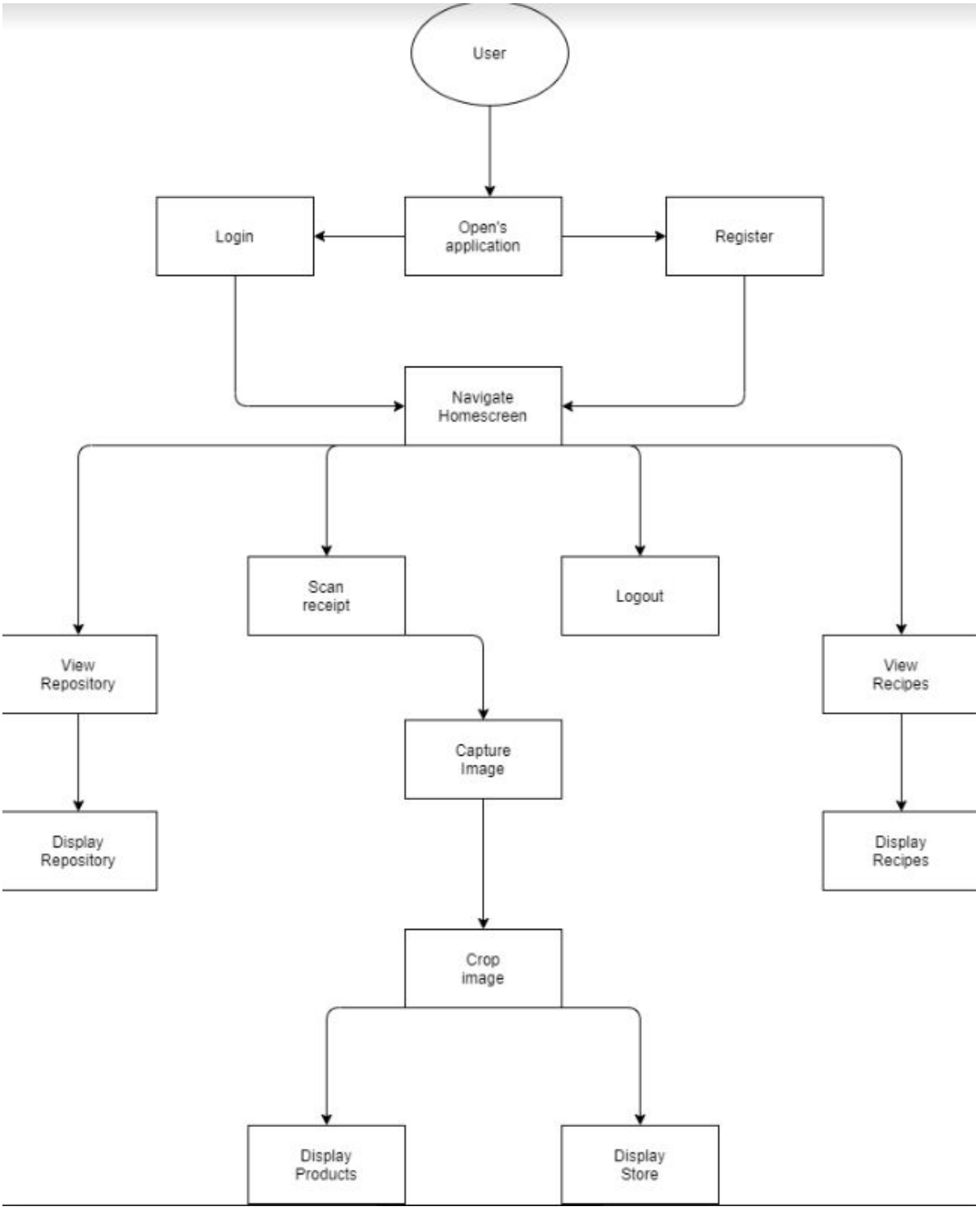
User Hardware
The hardware involved in using the application is android
or for admin the is through the admin console

Services

1. Login the app requires the user to login with their Account they have created in order to access further services. If no account is registered. Account registration procedures follow
2. User can take pictures of receipts that they own and retrieve back actual text of the products on the receipt
3. User can view his/her own list of products that have been scanned from the receipt so far
4. User can view recipes that are worked out to be relevant to the user based on current repository
5. Admin can view all connections made
6. Admin can view all requests to the flask api

Resources
Information provided externally about products recipes or individual ontology of products. Users will continuously look for information about all of the above to identify products and conjugate recipes

5. High level Design



5.1 Description

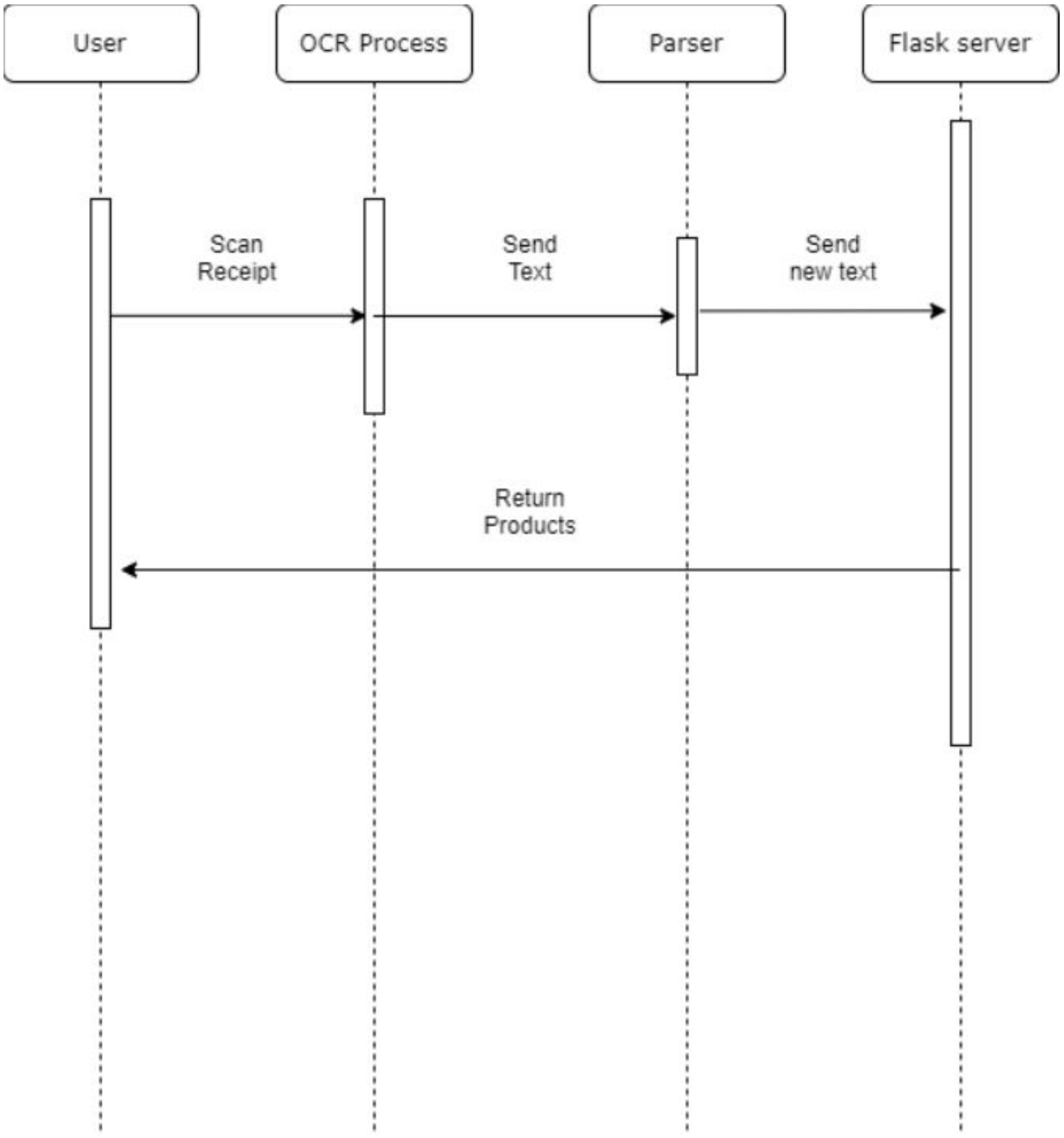
The diagram above illustrates a high level functionality component. When a user enters the application .He/She is greeted with a login or register option.After selecting either option they are directed to a homescreen with the option of logging out, scanning a receipt , viewing current repository or view prescribed recipes. If the user Selects the Scan receipt option, the user will be prompted to take a screenshot of an image.

Two text fields will be displayed, one for the result after the text has been extracted and the second to display the image captured. After the image is cropped both text fields will be filled, with the result field being intermediate to the final result. By clicking the next button, the text will then be parsed accordingly and store and products identified

By clicking the view repository option, the system will check the users current collection of products stored within firestore database and will display products in a listview back to the user

By clicking the view recipes option. Users current repository of products will be analyzed against their respective ontologies and queried against the host database for recipes with similar ingredients. The results will be 10 random recipes will be prescribed and displayed to the user in a listview

5.2 Sequence Diagram



The following above diagram is an interaction diagrams that details how operations are carried out for the given scenario.Here we have a sequence diagram of the process of a user opening the application and using the “Scan” receipt feature

- First the user must capture and image
- Than user must crop the image to frame the receipt in such a way to filter background influence on image and have the title of the store and total price charged in viewing range
- Once the image is captured the OCR(optical character recognition) process will begin.
- After text extracted it is sent to the parser to which the receipt belongs to. The parser will manifest a new string of products filtering unwanted characters or strings
- The newly manifested string is sent to the flask server to find products relatable to that string extracted using string similarity
- Products are than sent back to the user

6. Implementation

For the following, the project is broken down into a front end and back end system

Back end / Python

Orchestrates all web scraping activities for the available supermarket products and recipes facilitated by the application. I did this using Selenium and beautiful soup libraries. All data is stored in csv files and incorporated into the Sqlite3 database. Once the data was readily available.A separate dataset was created for products corresponding ontologies for all stores. This new dataset that is roughly 20,000 rows long was constructed manually. With the database loaded successfully, the back end could also hosts the Flask framework. This constructed api allows for incoming requests about individual datasets , such as handling similarity of strings against dataset values or simply querying a dataset and returning the data in a json format

Front end / Java

The front end is the android application and is responsible for all user interface responsibilities.While also managing image text recognition and individual parsers for each store. The output of which is sent as requests to the back end flask api using Retrofit2 libraries. This side of the system also manages login and registration of users as well as storing personal fields to each user(i.e products that they own) to do this. I utilized cloud service firebase and firestore. Firebase manages user login and registration activities and Firestore database maintains personal details of each user which can be in turn used in the repository activity to display products recorded as owned by the user from each scanning process