HIGH LEVEL DESIGN DOCUMENT

Automated Shopping Cart System

UE19CS390A – Capstone Project Phase – 1

Submitted by:

Kuntal Gorai	**PES2UG19CS198**
SVSC Santosh	**PES2UG19CS346**
Skanda S	**PES2UG19CS391**
Vijay Murugan A S	**PES2UG19CS454**

Under the guidance of

Prof. Prajwala T R
Assistant Professor
PES University

January - May 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

TABLE OF CONTENTS

1. Introduction

Shopping malls have become an integral part of city life and a lot of people commute to these marts to get their daily essentials and groceries. However, we are all forced to go through the tedious task of getting our items billed. We aim to make this whole shopping experience easier. The proposed device will classify items as and when they are put into the cart and append their net price to the bill accessible via the mobile application for easy payment.

2. Current System

Existing systems use RFID tags and barcodes to bill items, which is not feasible, especially for grocery items such as fruits and vegetables. The problem with this system is that it is infeasible to tag individual pieces of fruits and vegetables being a waste of time and effort.

3. Design Considerations

3.1. Design Goals

- The goal of this project is to create an app that can be used by a layman without much hassle.
- We aim to build a smooth and friendly interface for our application, making it intuitive and easy to navigate through the app.
- The device is not clunky and does not hinder the shopping experience of the user.
- The developed Android application is compatible with any regular Android smartphone even with little processing capacity.
- The sensitive user data is encrypted, protecting the privacy of the users.

3.2. Architecture Choices

- The application will only be available for Android devices as maintaining a different codebase for iOS devices alongside the android codebase is tedious and synchronicity between the codebases will be hard to maintain.
- A third-party service like RazorPay will be used to deal with the payments section of the application as the security issues that arise with monetary transactions are beyond our scope.
- We will be using the MobileNet model as it shows the most promise with respect to the accuracy it provides for the time/space it takes to train/deploy the model as compared to models such as YOLOv3, VGG-166, and ResNet.

3.3. Constraints, Assumptions, and Dependencies

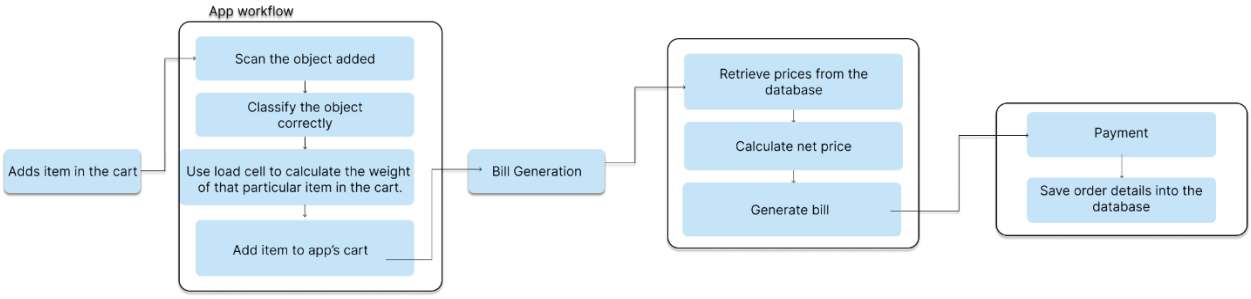
- Hardware constraints:
  i) The capability of a given device to run the model.
  ii) Inaccuracy of load cell.
  iii) Training requires powerful GPUs.

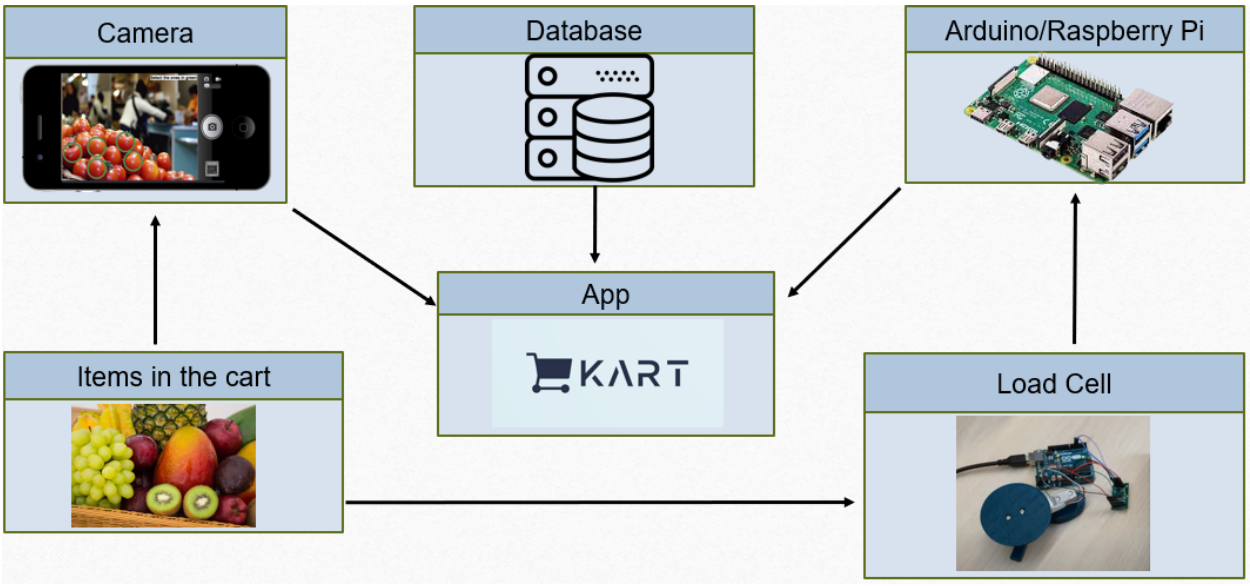
- Software constraints:
  i) Time optimization of algorithms used to classify images
  ii) Time to update the bill in the app.
  iii) A customer shows a cheaper product and places the expensive product into the cart, i.e., fooling the system.

- Assumptions:
  i) Objects are placed in a transparent bag.
  ii) Final Payment is handled by some 3rd party interface.
  iii) The customer has an android phone with minimum computational resources to use the app.

- Dependencies:
  i) Keras.
  ii) TensorFlow.
  iii) OpenCV.
  iv) DarkFlow.

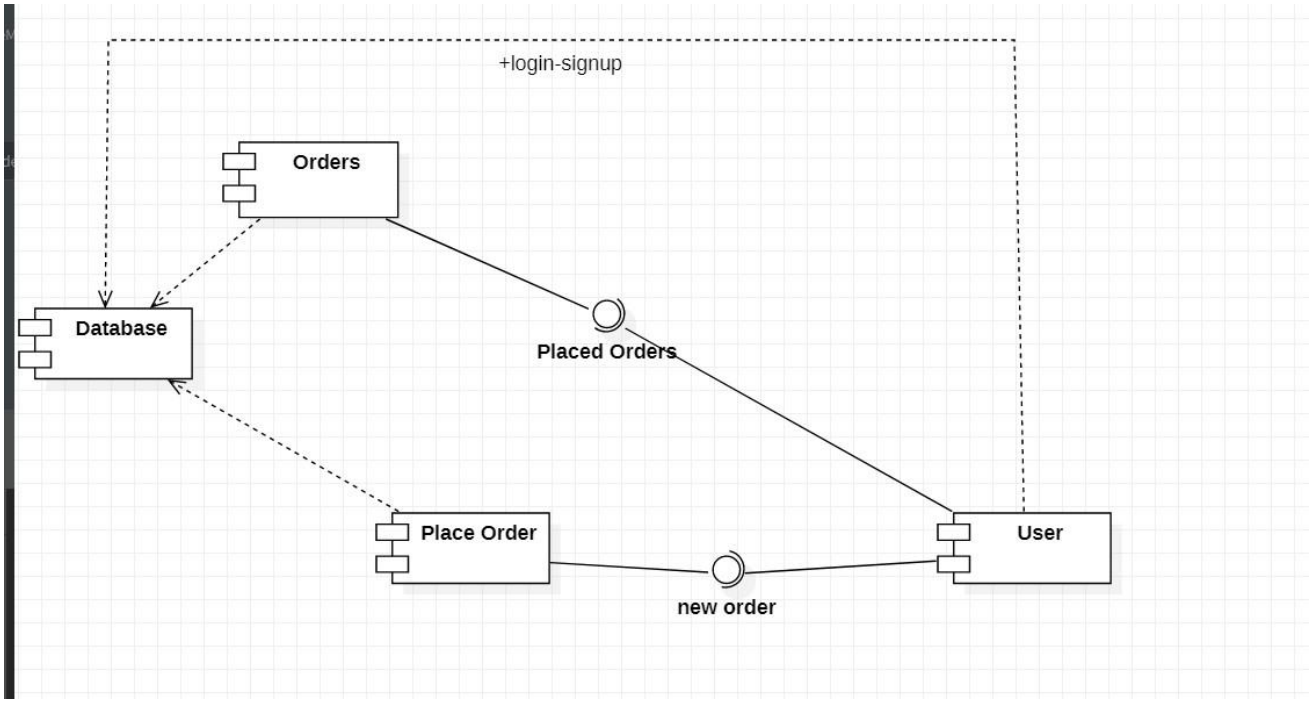
4. High-Level System Design:



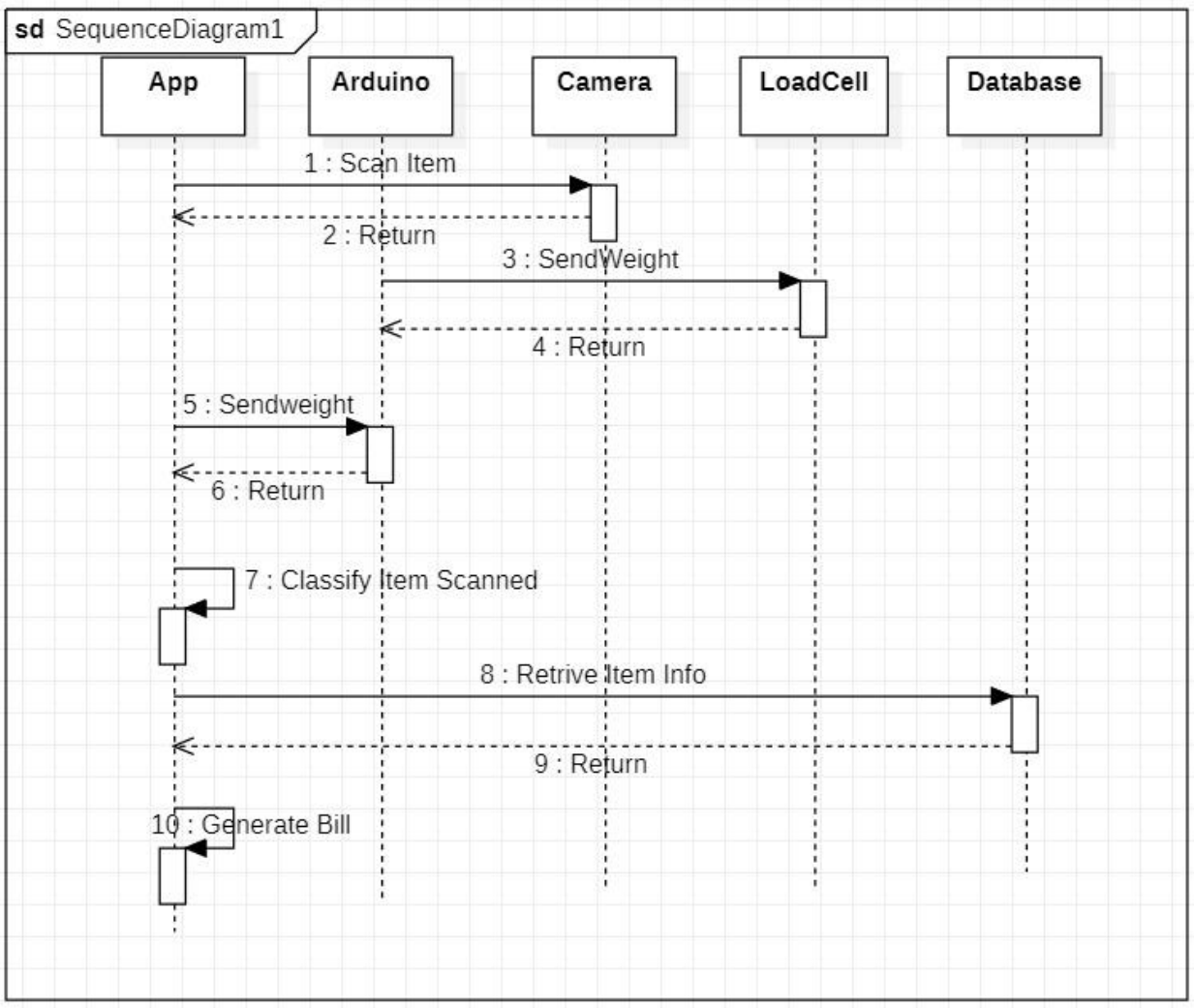


Once a user decides to add a particular item to the physical shopping cart, they have to scan it in front of the smartphone so that it correctly classifies that product using our Machine Learning technique to classify images. The needed amount of that item is placed in the shopping cart, which has a load cell to detect the weight. The load cell is used to get the weight of that item and sends signals to our Arduino which sends the weight of the added product to our Android app which retrieves the corresponding price from the database and cumulatively adds the prices of other items to the final bill.

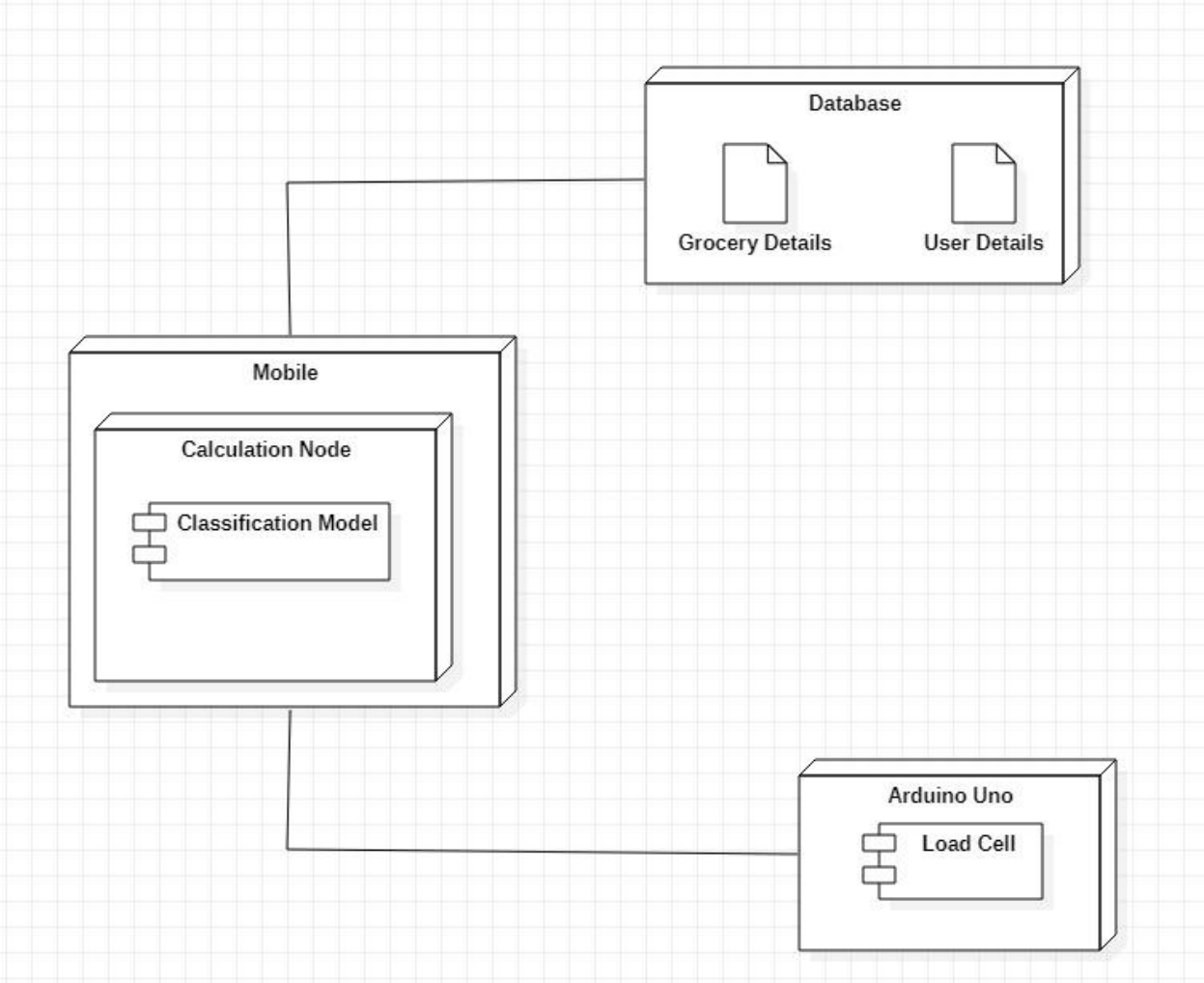
1) Component Diagram:



2) Interaction Diagram:



3) Deployment diagram:

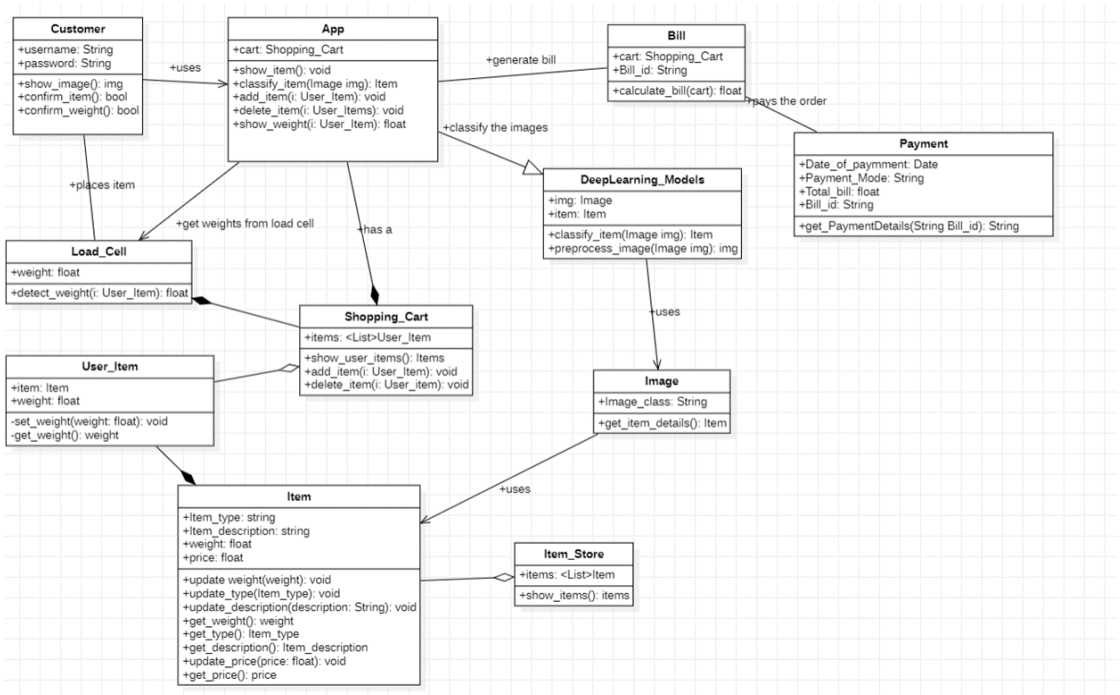


4) Security:

- As the application's cloud storage should be accessible from a variety of devices, the user's account or profile must be secured to prevent unauthorized personnel from accessing his or her account and its contents.
- For classifying items, since access to the camera is required, we need to ensure that this access is approved by the user and that the necessary permissions are obtained.
- Ensuring all security features required for payments are enabled by the third-party payment service.

5. Design Description

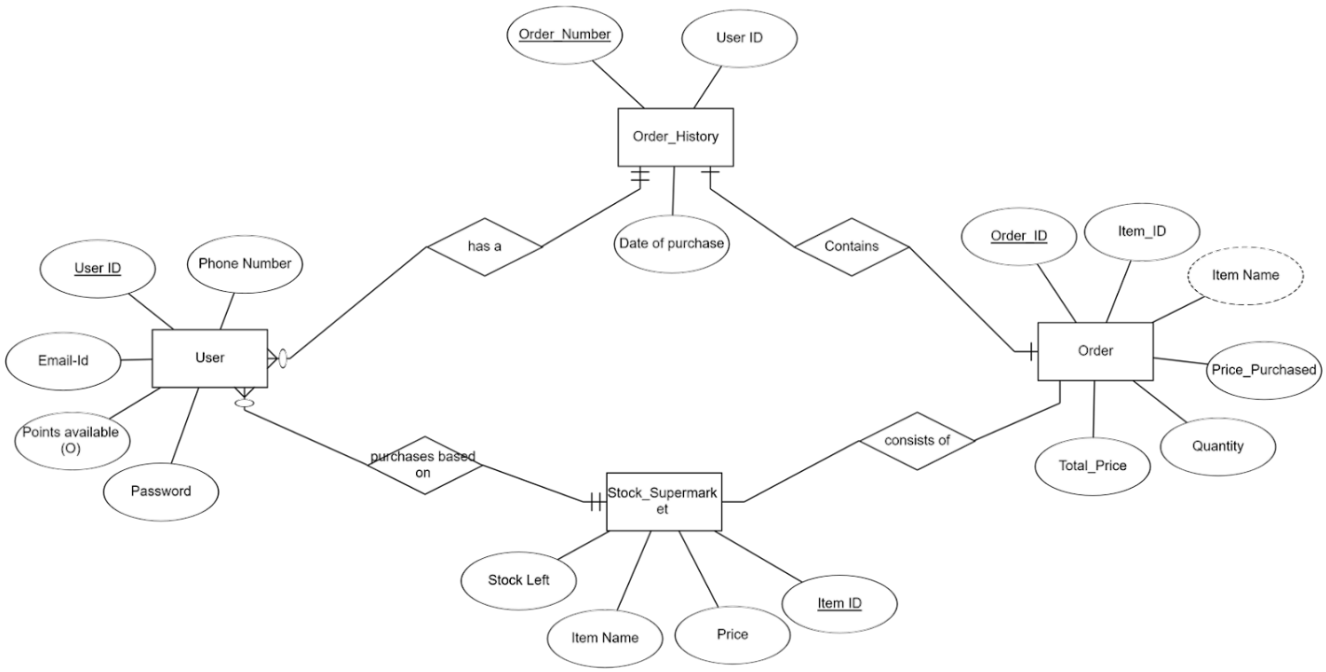
5.1. Master Class Diagram



5.2. Reusability Considerations

- For the android studio component, the same code will be used for each part of the front end, with a few minor changes; i.e. the backbone of the front end code remains the same for each functionality, with a few minor changes.

6. ER Diagram

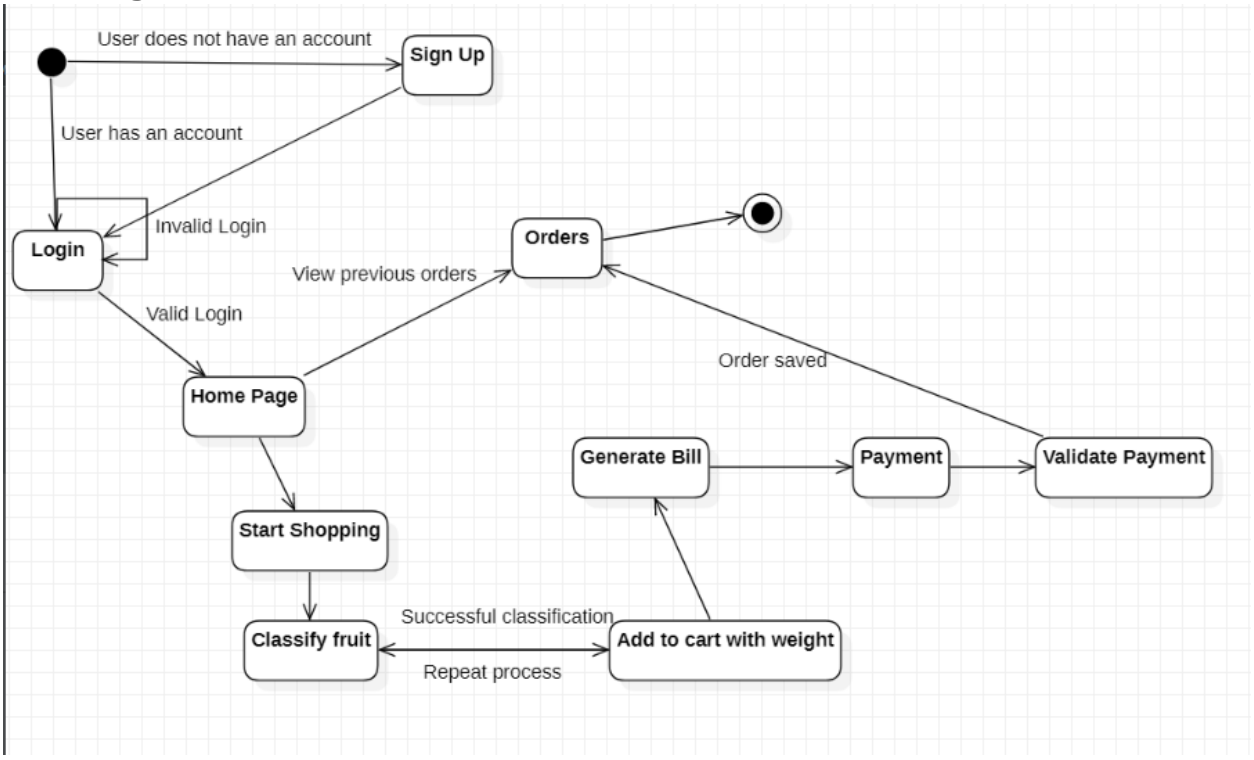


#	Entity	Name	Definition	Type
ENTITIES				
1.	Users	User	Stores all the details of each user.	Tangible
2.	Order History	Order_History	Holds all previous orders.	Tangible
3.	Orders	Order	Holds the details of every new order.	Tangible

4.	Stocks	Stock_Supermarket	A database to hold the stocks in the supermarket.	Tangible
#	**Attribute**	**Name**	**Definition**	**Type (size)**
DATA ELEMENTS				
1.	User	UserID	Stores the unique ID of each user.	VARCHAR(30)
		Phone Number	Stores the phone number of the user.	VARCHAR(15)
		Email-id	Stores the email address of the user.	VARCHAR(100)
		Password	Stores the encrypted password of the user.	VARCHAR(300)
		Points available	Stores the points for exclusive discounts.	VARCHAR(50)
2.	OrderHistory	Order-Number	Stores each order with a unique number	VARCHAR(100)
		UserID	Stores the ID of each user with the corresponding order	VARCHAR(30)
		Date of Purchase	Stores the date of the order placed	DATE
3.	Orders	Order_Id	Stores the unique ID of each order.	VARCHAR(30)
		Item_Id	Stores the unique ID of each item.	VARCHAR(30)
		Item_Name	Stores the item name	VARCHAR(40)
		Price_Purchased	Stores price of purchased item	FLOAT

		Quantity	Stores quantity of an item purchased	FLOAT
		Total_Price	Stores the sum total price to be paid by user	FLOAT
4.	Stock_Supermarket	Item_Id	Stores the unique ID of each item.	VARCHAR(30)
		Price	Stores price of item	FLOAT
		Item_Name	Stores the item name	VARCHAR(40)
		Stock_Left	Stores stock of remaining items	FLOAT

7. State Diagram



8. User Interface Diagrams

There are 6 main pages:

- Login Page
- Signup Page
- Home Page
- Order History Page
- Edit Profile
- Start Shopping Page

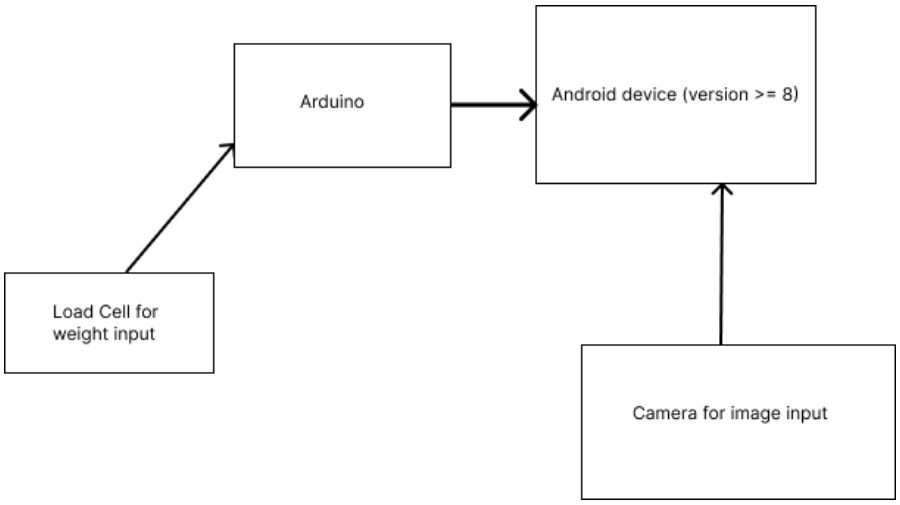
9. Report Layouts

1. Introduction
    1.1Purpose
    1.2Target Audience
2. Design Implementation
    2.1Functional Dependencies
    2.2 Function Description Table

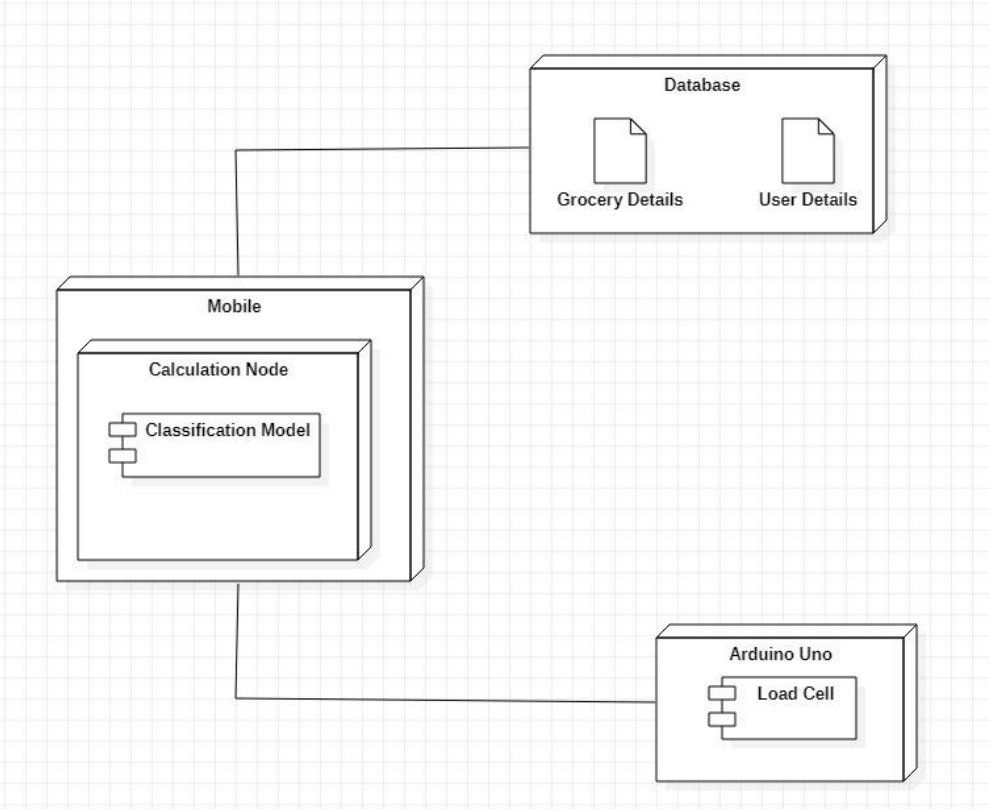
Name	Version	Actors	Pre conditions	During	Post condition	Exception	Additional notes

3. System Documentation
    3.1 Overview
    3.2 Technical Architecture (Training, Testing)
    3.3 Business Architecture
4. Support Documentation
5. User Documentation
6.References

10. External Interfaces



11. Packaging and Deployment Diagram



12. Help

- Demo Video: A demo video can be played for every user the first time he/she uses this in order to give them an idea of how our system works. This would help them get an idea of how to use the system.
- Feedback can be taken from users at the end of the usage to get to know ideas or places on which we can improvise thereby leading to a better system for the end-users.
- We can also provide some questions and answers to the most frequently raised doubts in the mind of the users in the FAQ section.

13. Design Details

13.1. **Novelty:** Previous projects implemented made use of RFID tags which are attached to every item. Also, items like fruits and vegetables don't always have an RFID tag that is considered in our project.

13.2. **Innovativeness:** The concept of using transfer learning enables us to reuse the model developed for any other classification task which could easily enable our app to solve any classification problem easily.

13.3. **Performance:** We aim to train the model to be able to classify the provided item in 3-4 seconds with an accuracy of at least 90 percent.

13.4. **Security:** Making use of Encryption and Decryption techniques to ensure the privacy of the user is protected. Only the admin of the Super Market would be able to modify the weights, price, and other necessary details of that supermarket. In the case of payments, the use of already existing third-party services would deal with the total security of payments.

13.5. **Reliability:** To ensure optimum reliability, we attempt to construct the model with high precision and low latency

13.6. **Portability:** The application is portable as it is designed to support all Android smartphones with versions of Android 8.0 (Oreo) and above. The Arduino, however, is not portable.

13.7. **Reusability:** The concept of using transfer learning enables us to reuse the model developed for any other classification task which could easily enable our app to solve any classification problem easily.

13.8. **Application compatibility:** The application will be designed to support all Android smartphones with versions Android 8.0 (Oreo) and above.

Appendix A: Definitions, Acronyms and Abbreviations

RFID: Radio Frequency identification
YOLO: You Only Look Once
VGG: Visual Geometry Group
ResNet: Residual Neural Network

Appendix B: References

1) https://ieeexplore.ieee.org/abstract/document/9089651
2) https://www.irjet.net/archives/V8/i4/IRJET-V8I4678.pdf
3) https://arxiv.org/pdf/1901.00711.pdf
4) https://link.springer.com/chapter/10.1007/978-3-030-49076-8_1
5) https://www.javatpoint.com/uml-class-diagram
6) https://www.uml-diagrams.org/deployment-diagrams-overview.html
7) https://medium.com/@mobindustry/how-to-integrate-machine-learning-into-an-android-app-best-image-recognition-tools-1ba837c27903
8) https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/Raspberry_Pi_Guide.md
9) https://www.semanticscholar.org/paper/Implementation-of-Smart-Shopping-Cart-using-Object-Oh-Chun/f5ff8985a06e02715dbbbbf4f2b4470289e003f0
10) https://analyticsindiamag.com/deploying-machine-learning-models-in-android-apps-using-python/
11) https://eloquentarduino.github.io/2020/01/image-recognition-with-esp32-and-arduino/

Appendix C: Record of Change History

[This section describes the details of changes that have resulted in the current High-Level Design document.]

#	Date	Document Version No.	Change Description	Reason for Change
1.	**22-04-2022**	**1**	Initial version	**Null**
2.				
3.				

Appendix D: Traceability Matrix

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.
3.1 Product Features	5. Design Description/ Master Class Diagram
3.2 User classes and characteristics	4. High-level system design
5.1 User Interfaces	7. User Interface Diagrams