CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Buyers directly visit the shops or market to buy products like fruits and vegetables. For that buyers need to invest more money and time in day to day life. In the same way seller also face lots of struggle to reach their product to shops. Farmers produce the food products. The food products from the formers are handover to the middle man the food products are taken care by the middle person and hike the price for each products and sell that to the sellers like shopkeepers. Shopkeepers can sell the products in their shops with some profit, here also the seller can hike the product price and fix the amount for each products. The buyers can buy the products in that fixed rate. If that product is delivered to the buyer's house here also the price hike is happen. Buyers buy the product in that hiked price in same case former getting less profit. The middle man and shopkeepers simply gets the profit. That products are not assure the quality. And there is no opportunity to get different products from the single place.

1.2 CHALLENGES

The main challenges that are found in the IWONTMISS system are as follows:

Integrity

Difficult to ensure the integrity of the seller. Whether the seller sells the correct and quality product.

Security

Provides the security for the details of the Seller and Buyer.

Platform

Select the language which will run in all platforms like computer and mobile devices.

Maintenance

Maintain the details of the sellers with available product as well as buyers with their ordered product.

Technology

Learn the technology and complete the project with in the duration.

Deliver the product

Deliver the product to ordered address within the time.so select the seller within deliverable distance from the buyer.

Usage

Motivate the buyer and seller to use this system frequently.

Law issues

Follow the rules and of ecommerce and tax related problems.

1.3 PROJECT SCOPE AND OBJECTIVES

To solve the issues explained in the problem statement the system named IWONTMISS is introduce. The major objective of the system is updating the process done through online. For buyers and sellers this system provide easy way to buy and sell food products. Here the seller denotes the person who produce the product. This system directly connects the buyer with seller which means that process didn’t have any middle man or shopkeeper.so that the buyer is able to get good and healthy product in less amount in that same way the farmer is able to get high profit. Buyers also get different products in one place.

1.4 CHAPTER PLAN

Hereby the chapters of the IWONTMISS system is planned in following manner [4]:

- Chapter two named as system analysis which deals with existing system, proposed system, requirement gathering and explanation about the company.
- Chapter three named as system design which has uml diagram, dfd diagram, module design, and database design.
- Chapter four named as implementation which has explanation about implementation of proposed system.
- Chapter five named as system testing which has testing strategies and test cases.
- Conclusion and future enhancement explained in sixth chapter.
- And it have two appendix chapter first one is screen shot and second one is sample coding.
- Final chapter end with references.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In existing system sellers and buyers are used to go to the market directly to sell and buy food products. There is a middle person between the farmer and buyer. The food products from the farmers are handover to the middle person. The food products are taken care by the middle person and fix the price for each product and the buyers can buy the products in the fixed rate.

Disadvantages of Existing System

- It’s time consuming process.
- It involves middle person.
- Buy more than one product in a single place has less opportunity.

2.2 PROPOSED SYSTEM

To solve the issues in the existing system the system named IWONTMISS is introduced. The major objective of the system is updating the process done through online. For buyers and sellers this system provide easy way to buy and sell food products. Here the seller denotes the person who produce the product. This system directly connects the buyer with seller.so that the buyer is able to get good and healthy product in less amount in that same way the farmer is able to get high profit.

Advantages of Proposed System

- There is no middle man in the process.
- Buyers also get different products in one place.
- It’s consumes less time.

2.3 REQUIREMENT GATHERING

Stake holders of this system

- Aravind Venkatesan M (who proposed and created this system)
- Oribyne technologies (which provides the idea and support through the entire process)
- Buyers in the system
- Sellers in the system
- Firebase
- Gmail
- PSNA College students and staffs of MCA department

Requirements

- Study the buying habits of people.
- Collect the seller details in a region.
- Study the food cycle from the former to the buyer.
- Collect the details of product that are able to sale through online.
- Study the calculations of fixing the rate for each product.
- Visit and Study the websites like bigbasket, flipkart, Ola.

2.4 COMPANY PROFILE

Oribyne is into helping customer’s business systems and information through robust products and customized solutions that hinge on reliable, secure, and integrated technologies. A humble beginning from a single room office to a full-fledged complex, an array of prestigious satisfied customers, a dream team of the most resourceful people in the trade. Yes, Oribyne is green and still growing!! In this ever competitive world of IT consulting, Oribyne has always stood apart.

Oribyne have the prowess, the urge to perfection, the heart to win, the mind to learn, the eyes to observe, and the confidence to conquer. Established in 2009, Oribyne made a very quick reputation by addressing the small and medium business industry's requirement with a core focus on quality.

oribyne research and development team is constantly working to improve company’s services for customers - to ensure that customer requirements are met in the possible way.

Oribyne provides innovative solutions, tailor-made to suit almost any industry. These solutions combine the strengths of our technology with oribyne domain expertise to help customers quickly reach their objectives. If there is anything customers are not obvious with regards to oribyne services, please get in touch with oribyne friendly customer care consultant. We will endeavor to be of assistance to you.

Oribyne vision is to be the world's best IT solutions and IT enabled services provider. Being the best means, providing outstanding quality, service, security, and value, so that we make every customer’s customer smile. We will achieve this through continuous improvement driven by the integrity, teamwork, and innovation of Oribyne people.

This chapter “System analysis” briefly explains about existing system, proposed system and requirement gathering of the project in detail. Next chapter 3, system design which deals with UML, DFD diagram, module and database design of the project.

CHAPTER 3

SYSTEM DESIGN

3.1 UML AND DFD DIAGRAM

Data Flow Diagram

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design) [4].

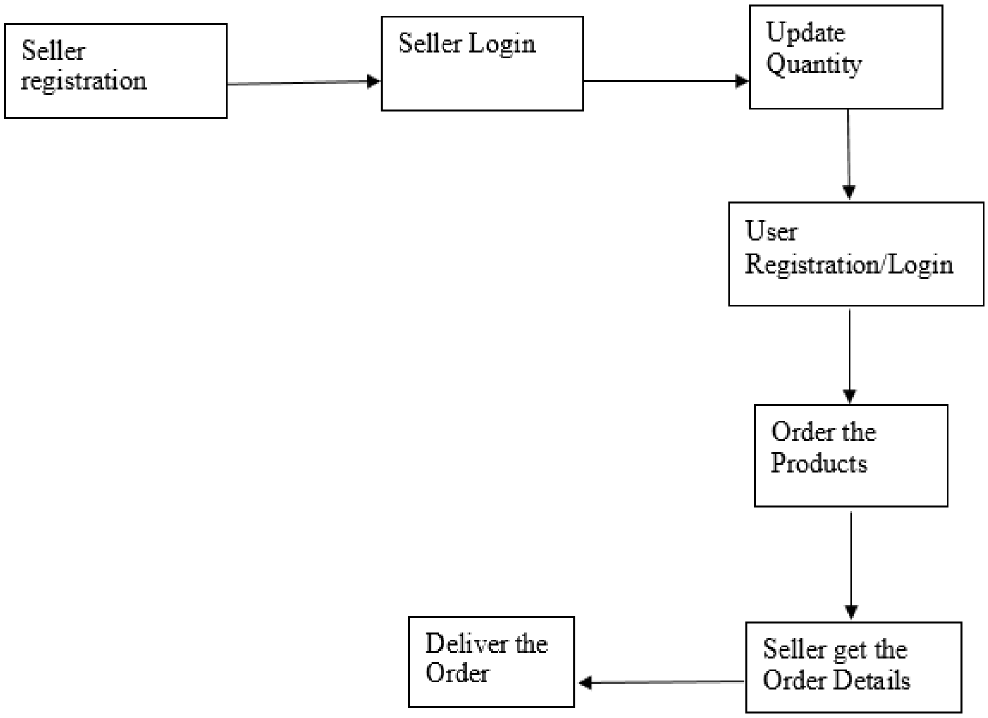


Fig 3.1 Data Flow Diagram

Figure 3.1 illustrate the flow of data in the IWONTMISS system. The flow starts from seller registration. This process is done by the admin. Then seller login into the page then the seller regularly update the product quantity. Likewise user also able to register that registration is done bu the buyer itself. After the login user can view the available product from that buyer can order their needed product. If the order is placed successfully then the seller gets the buyer details along with the order details.

Use Case Diagram

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal [4].

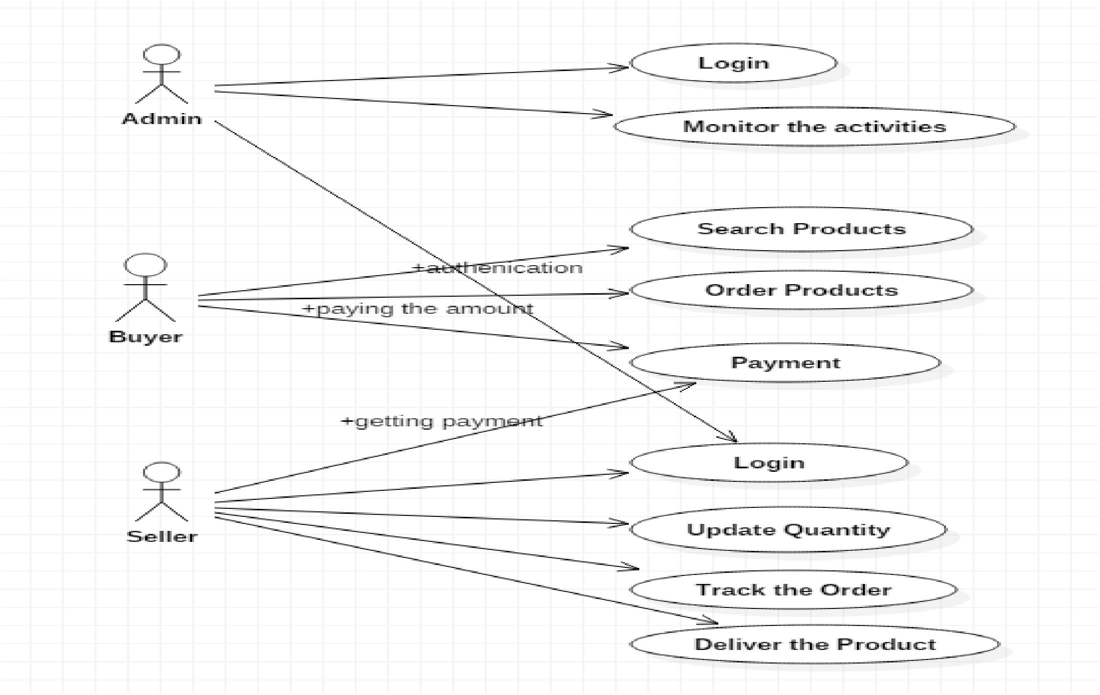


Fig 3.2. Data Flow Diagram

The above figure 3.2 explains the use cases with scenario. The IWONTMISS system contains of three users they are Admin, Buyer and Seller. Each user has their own use cases and some use cases are connected to more than one user. Admin have login and monitor the activities. Buyer able to search and order the products. Seller update the quantity, track and deliver the order. The process of seller login also connected with admin because to verify and create the seller. Payment process also interconnect with buyer and seller.

Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP) [4].

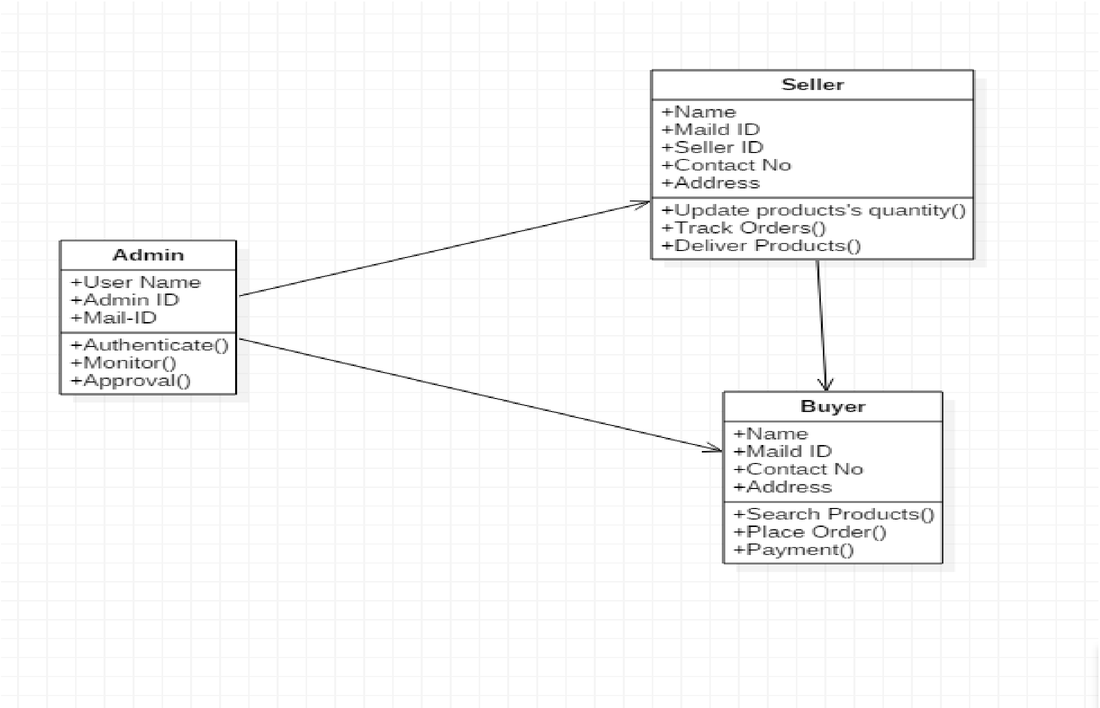


Fig 3.3 Class Diagram

Figure3.3 explains the classes of the IWONTMISS system with its attribute and methods. This system divided into three classes that are Admin, Seller and Buyer. Admin have user name, mail id attribute and authentication monitor and approval methods. Like that seller and buyer have their own attributes name, mail id, contact number and address attributes and seller have update quantity track and deliver the order methods. Buyer have search product place order and payment methods. this class diagram clearly explains what are the attributes or data entities needed for Admin, buyer and seller in the system.

Sequence Diagram

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome. A sequence diagram is an essential component used in processes related to analysis, design and documentation. A sequence diagram is also known as a timing diagram, event diagram and event scenario [4].

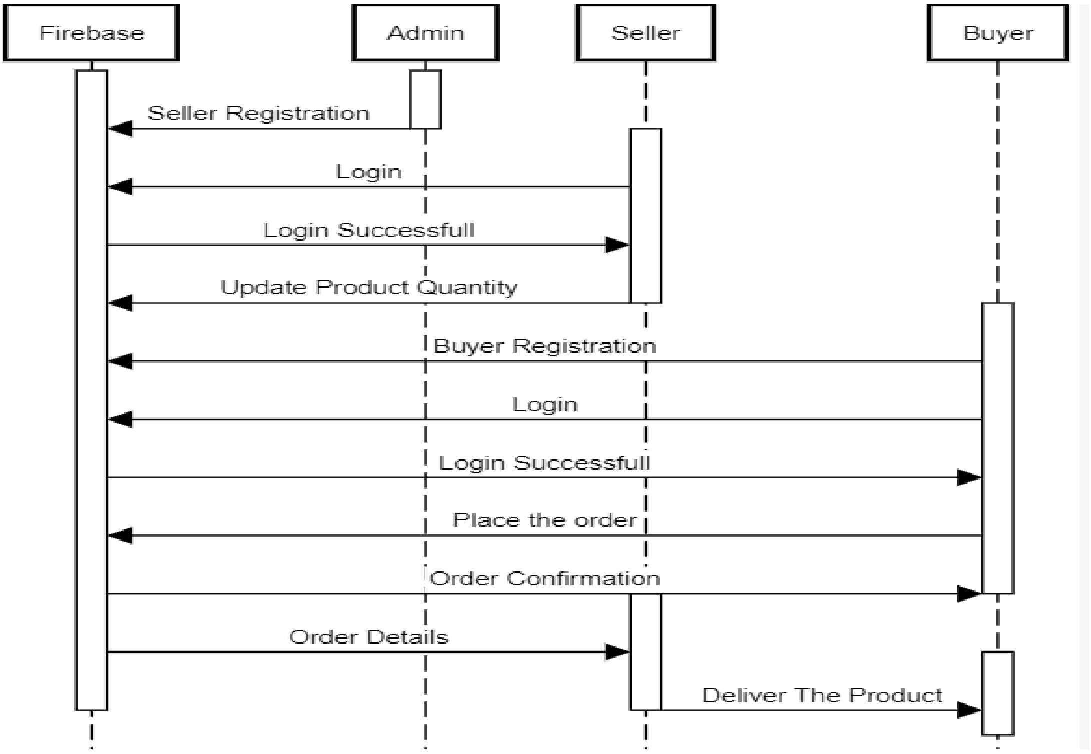


Fig 3.4 Sequence Diagram

Figure3.4 illustrate the flow of events occurs in the IWONTMISS system. This flow is derived from the use case. In this figure rectangular box in the time line of the firebase shows firebase is always will be active. Because the entire system is depends on the firebase. The flow of events in the system are as follows:

Admin register the seller details and the data are store in the Firebase.

After the successful registration the seller login into system. If the login process is successful the seller able to update the quantity of the product and waiting for the order from the Buyer.

Buyer is register using their credentials and the data is verified and stored in the firebase. Then the buyer is login into the system and place the order for their needed product if the order is placed successfully then the buyer is getting the confirmation from the firebase.

After the order placed the seller grits the buyer details along with the ordered product. The seller is deliver the product to the buyer.

ER Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure [4].

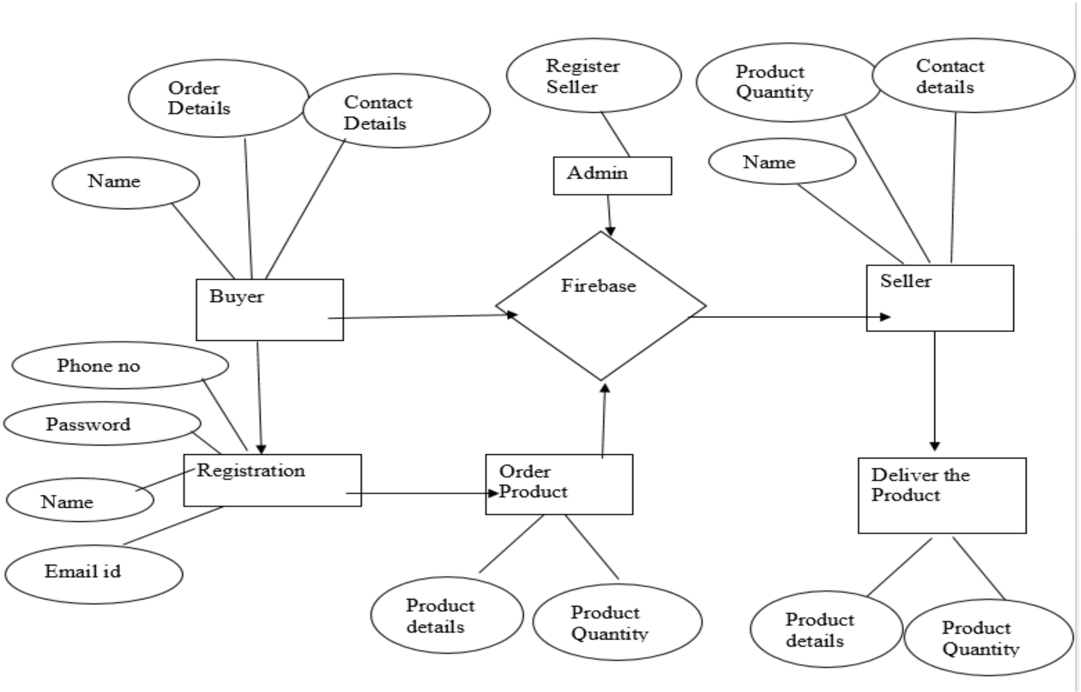


Fig 3.5 ER Diagram

The above figure3.5 explains the relationships between the entities in the IWONTMISS system. Here the base data is firebase, and the Seller and Buyer are the two major database and that two are having their own entities. And the admin register the seller in the database and buyers are able to register them self that data are stored in the firebase. And the seller update the quantity of the product that details are stored in the firebase then it will display to the user based in the quantity. The buyer placed the order and the details are stored in the firebase and sent to the seller. The whole system is depend on the firebase data. So it is controlled by the admin itself. That ER diagram helps to find the databases and data entities of the system.

3.2 MODULE DESIGN

This system is proposed based on 3 necessary modules.

Module 1-Admin Module

Module 2-Seller Module

Module 3-Buyer Module

Admin Module

Admin also have to login the website for add the seller by getting seller name, address, mobile number and admin will create the mail id and password for the seller [A1].

Admin also able to monitor the entire process. Admin is the person who fix the product price. Admin able to visit the seller and buyer page.

All the details of data base in firebase linked to the admin mail id. The entire IWONTMISS system is depend on the admin process. Admin also have the rights to remove any seller or buyer from the system.

Seller Module

Seller module is for sellers who sale the product mostly the formers. Seller need to contact the admin then admin will create the login and password for the seller[A2] along with admin enter the details of the product which the seller wants to sale[A3].

After getting the login credentials the seller is able to login using that details. Then the seller take in to the quantity update page the seller able to mention the quantity of the product each day and also if sale the product outside the system.

And the seller have the orders page in that page the seller view the order made by the buyer with quantity of the product and buyer details such us name , address, mobile number[A5]. The settings page allow the sellers to change the phone number, and password of the seller.

Finally the seller deliver the product to the buyer by their order details.

Buyer Module

Buyer Module is the place where user can order the food items. For that buyers need to login using their username which is mail id of the user and its password. If the buyer is new to the site then need to sign up [A5].

To sign up the buyer go to the signup page and enter the credentials like username, mail id of the user, phone number and password of mail id [A5]. The details of the mail id and password is verified in the firebase and if that entered mail id is exists then the buyer is created and user id for the buyer is generated in firebase itself.

After login process the buyer take into the menu page where all the menu list will show for the user with the price of the product per kg[A6]. Buyer adds their needed products to the card by clicking the cart icon.

After adding products to their cart buyer go to the cart page in that page selected items will show for the buyer with total cost. Buyers allow to increase the quantity or delete the product from cart[A7].

From that cart it goes to checkout process in that page buyers need to verify the amount, address, mobile number and select the payment mode after that place the order[A8] if the order is placed then order successfully placed message will show to the buyer[A9] .

The details of the ordered products and its seller details like seller name phone number and address will show in the orders page.in the settings window the buyers allow to change the user name, address, mail id, password, mobile number and mail id[A10].

3.3 DATABASE DESIGN

Database is an integrated collection of data and provides a centralized access to the data and makes possible to treat data as a separate resource. Usually centralized data managing software is called a Relational Database Management System (RDBMS). The most significant different between RDBMS and other type of Data Management is the separation of data as seen by the program and data as store of on the direct access storage device. This is the difference between logical and physical data [7].

User Table

Description

The following table3.1 is used to store the Buyers details like Name, Id, Mobile number and email id.

Table 3.1 Buyer Table

SERIAL NO	FIELD NAME	DATATYPE	DESCRIPTION
1	Bname	Varchar(50)	Buyer Name
2	Bid	Int(50)	Buyer Id
3	Baddress	varchar(50)	Buyer Adress
4	Bnumber	int(10)	Buyer Mobile Number
5	Bemail	varchar(50)	Buyer Email address

Seller Table

Description

The following table 3.2 is used to store the Seller details like Name, Id, Mobile number, email id and the Product that the seller wants to sale

Table 3.2 Seller Table

SERIAL NO	FIELD NAME	DATATYPE	DESCRIPTION
1	Sname	Varchar(50)	Seller Name
2	Bid	Int(50)	Buyer Id
3	Saddress	varchar(50)	Seller Adress
4	Snumber	int(10)	Seller Mobile Number
5	Semail	varchar(50)	Seller Email address
6	Sproduct	Varchar(30)	Product details which is the seller wants to sell

Product Table

Description

The following table 3.3 is used to store the Product details like Name, Id, Price and Quantity of the product.

Table 3.3 Product Table

SERIAL NO	FIELD NAME	DATATYPE	DESCRIPTION
1	Pid	Int(50)	Product Id
2	Pname	Varchar(50)	Product Name
3	Pprice	int(20)	Price of the Product
4	Pqunt	int(10)	Product Quantity

This chapter “System design” briefly explains about UML, DFD diagram, module and database design of the project. Next chapter 4, implementation which deals background study for the project and implementation process of the software.

CHAPTER 4

IMPLEMENTATION

4.1 BACKGROUND STUDY

Frontend and Backend Software

Ionic

Ionic is a complete open-source SDK for hybrid mobile app development.it is built on top of AngularJs and Apache Cordova. The more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova[8].

Firebase

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

Firebase provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, swift and Node.js applications.

The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side-enforced security rules[7].

Software Description

Ionic

Ionic provides all the functionality which can be found in native mobile development SDKs. Users can build their apps, customize them for Android or iOS, and deploy through Cordova. Ionic includes mobile components, typography, interactive paradigms, and an extensible base theme.

Using Angular, Ionic provides custom components and methods for interacting with them. One such component, collection repeat, allows users to scroll through a list of thousands of items without any performance hits.

Besides the SDK, Ionic also provides services that developers can use to enable features, such as push notifications, A/B testing, analytics, code deploys, and automated builds.

Ionic also provides a powerful command-line interface (CLI), so developers can get started with creating a project with a simple command. The CLI also allows developers to add Cordova plugins and additional front-end packages, enable push notifications, generate app Icons and Splash screens, and build native binaries[7].

Supported Platforms

Ionic is focused on building for modern Web standards and for modern mobile devices. For Android, Ionic supports Android 4.1 and up. For iOS, Ionic supports iOS 7 and up. Ionic supports the Universal Windows Platform for building Windows 10 apps.Ionic Framework, powered by Angular.js also supports BlackBerry 10 apps[7].

Performance

Compared to hybrid applications, mixing Ionic code with native mobile app code in Apache Cordova allows for higher performance of the resulting product.

Utilizing AngularJS (rather than jQuery) allows Ionic to rely on native hardware acceleration (rather than extensive DOM manipulation). Ionic leverages CSS transitions and transforms for animation as a way to leverage the GPU and maximize available processor time[1].

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document[5].

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device[5].

Sass

Sass(Syntactically awesome style sheets) is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). SassScript is a simple scripting language used in Sass files.

JavaScript

JavaScript is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vaSst majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded[1][3].

AngularJS

AngularJS is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, a framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS framework works by first reading the HTML page, which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources[1].

JSON

Java Script Object Notation is an open-standard file format that uses readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems[3].

Firebase

Firebase Authentication

Firebase Authentication is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google (and Google Play Games). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase[7].

Firebase Storage

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage[7].

4.2 IMPLEMENTATION

How to Install Proposed Software

Step 1:

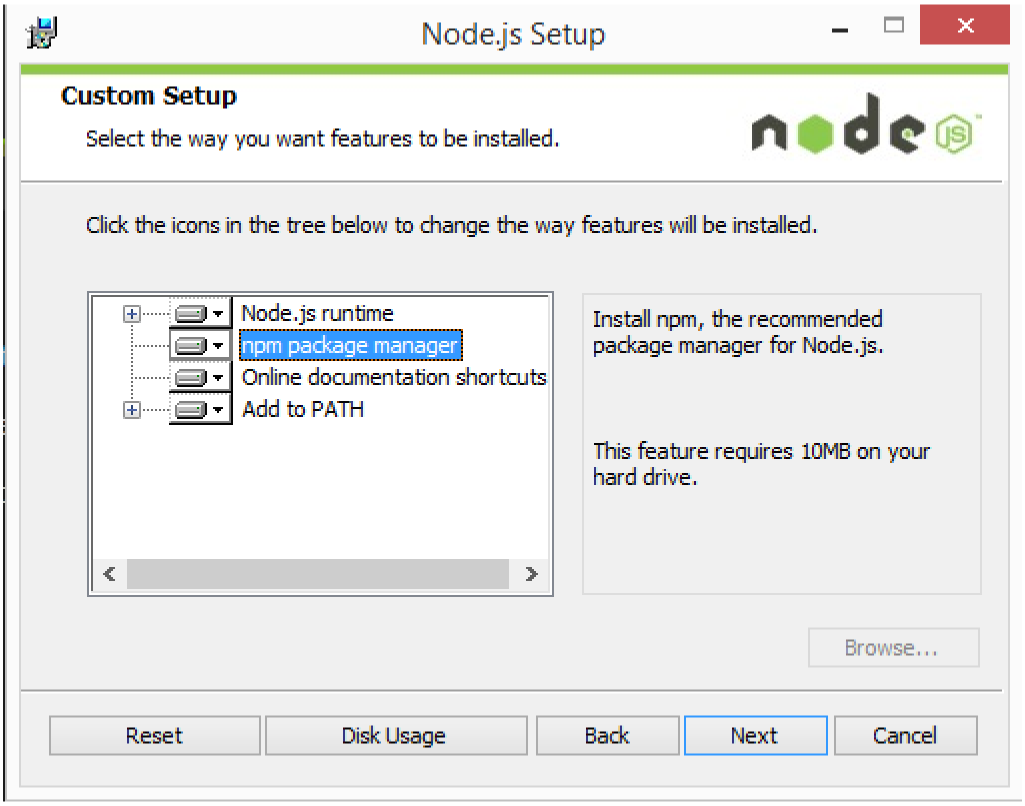
Download Node.js from the official Node.js web site: https://nodejs.org

Step 2:

Run the installer (the .msi file you downloaded in the previous step.)

Step 3:

Follow the prompts in the installer (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings)[2][9].



Node.js Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Node.js runtime
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install npm, the recommended package manager for Node.js.

This feature requires 10MB on your hard drive.

Fig 4.1 Nodej.s Setup

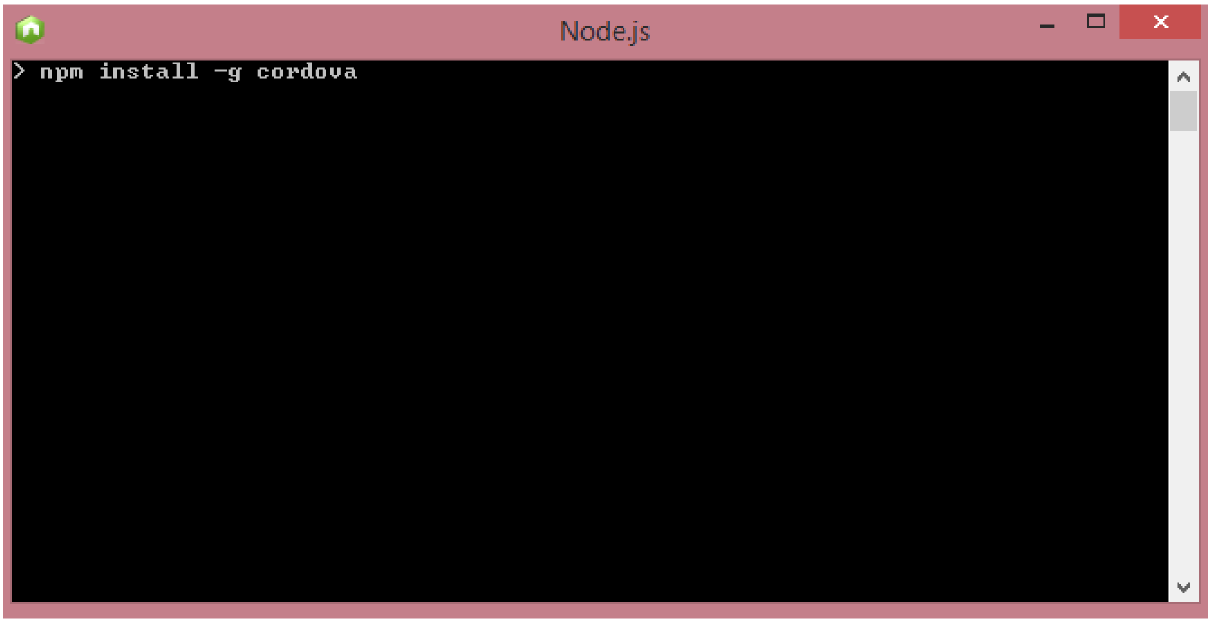
Step 4:

Restart your computer. You won’t be able to run Node.js® until you restart your computer.

Step 5:

Open a command window (Windows), and install Cordova and Ionic:

npm install -g cordova ionic



```
> npm install -g cordova
```

Fig 4.1 Installing Cordova

Step 6:

Navigate (cd) to the **ionic** directory and install the server dependencies:

npm install

Step 7:

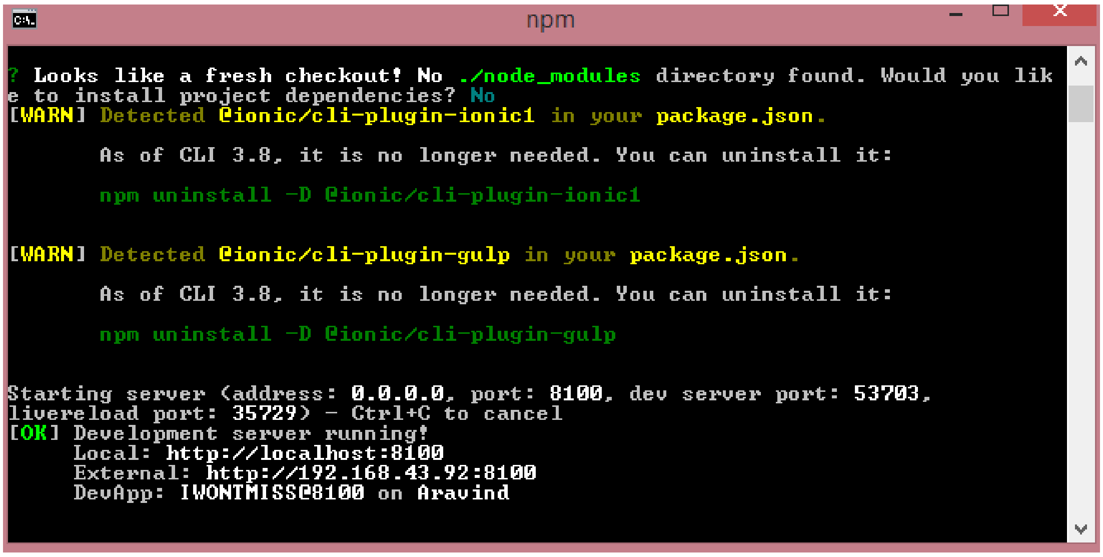
After install the server dependencies successfully then Start the server:

node server

Step 8:

Start the application in a browser using **ionic serve**.

ionic serve



```
? Looks like a fresh checkout! No ./node_modules directory found. Would you lik
e to install project dependencies? No
[WARN] Detected @ionic/cli-plugin-ionic1 in your package.json.

       As of CLI 3.8, it is no longer needed. You can uninstall it:

       npm uninstall -D @ionic/cli-plugin-ionic1

[WARN] Detected @ionic/cli-plugin-gulp in your package.json.

       As of CLI 3.8, it is no longer needed. You can uninstall it:

       npm uninstall -D @ionic/cli-plugin-gulp

Starting server (address: 0.0.0.0, port: 8100, dev server port: 53703,
livereload port: 35729) - Ctrl+C to cancel
[OK] Development server running!
     Local: http://localhost:8100
     External: http://192.168.43.92:8100
     DevApp: IWONTMISS@8100 on Aravind
```

Fig 4.1 Running the application

System Requirement

Hardware Requirement

Processor	Quad Core and above
Ram	2GB
Memory Space	50GB

Software Requirement

Operating System	Windows, Android and IOS
Framework	Ionic
Editor	Visual Studio Code, devtools
Server	Firebase

This chapter “Implementation” briefly explains about background study for the project and implementation process of the software. Next chapter 5, system testing deals with testing, test cases and outcome of the test for this project.

CHAPTER 5

SYSTEM TESTING

5.1 TESTING

Testing is the stage before system implementation where the system is made error free and all the needed modifications are made. The system was tested with test data and necessary corrections to the system were carried out. All the reports were checked by the user and approved. The system was very user friendly with online help to assist the user wherever necessary[4].

Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether the units are fit for use[4].

This IWONTMISS system is divided into several units like seller registration, seller login, update quantity, buyer registration, buyer login, order the product, sending seller details for buyer and buyer details for seller.

Each unit is tested successfully and the test results are noted.

Integration Testing

The system modules were tested as a group. Codes of each module used in the web page were put to test. That test is done to check whether the module is fit for use[4].

For integration testing units of Buyer’s registration, login, order the product and getting the seller details are integrated as a group. Likewise units of seller’s registration, login, update the quantity and getting the buyer details are integrated as a group are tested successfully and the results are noted.

System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements[4].

Admin, Buyer, Seller with firebase are integrated and run in the system.,which has specified requirements of hardware and software.

The entire system is tested successfully and the results are noted.

5.2 TEST CASE

A test case is a set of conditions or variables and inputs that are developed for a particular goal or objective to be achieved on a certain application to judge its capabilities or features. It might take more than one test case to determine the true functionality of the application being tested. Every requirement or objective to be achieved needs at least one test case[4].

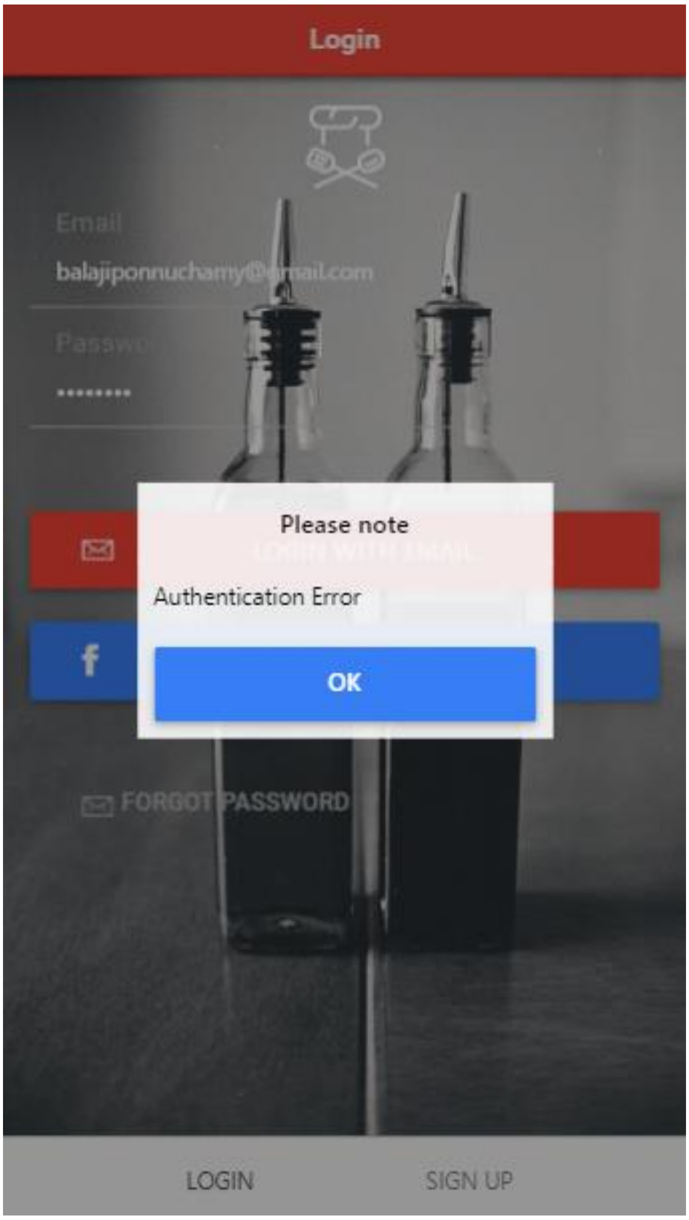
Test Case Results

Table 5.1 Test Cases

S.NO	TESTCASE	EXPECTED RESULT	ACTUAL RESULT
1	Signup With Invalid Username or Password of Gmail in buyer Signup Page	Error message should be display.	Error message is display.
2	Login with valid username and password in both Seller and Buyer Login Page	Error message should be display.	Error message is display.
3	Login with valid username and password in both Seller and Buyer Login Page	Menu Page should be open.	Menu page is open.
4	Register with valid details in Buyer Page	Registration done	User dash board open.
5	Address Field in order page of buyer is empty	Error message should be displayed	Error message is displayed
6	After placing order	A success message should be displayed	Order Successfully placed message is displayed

5.3 OUTCOME OF THE TEST

The above table 5.1 has some of the test cases with its expected and actual result of the test. The following pictures shows outcome of the login error and address field not selected error. That will help this system to ensure the accuracy.



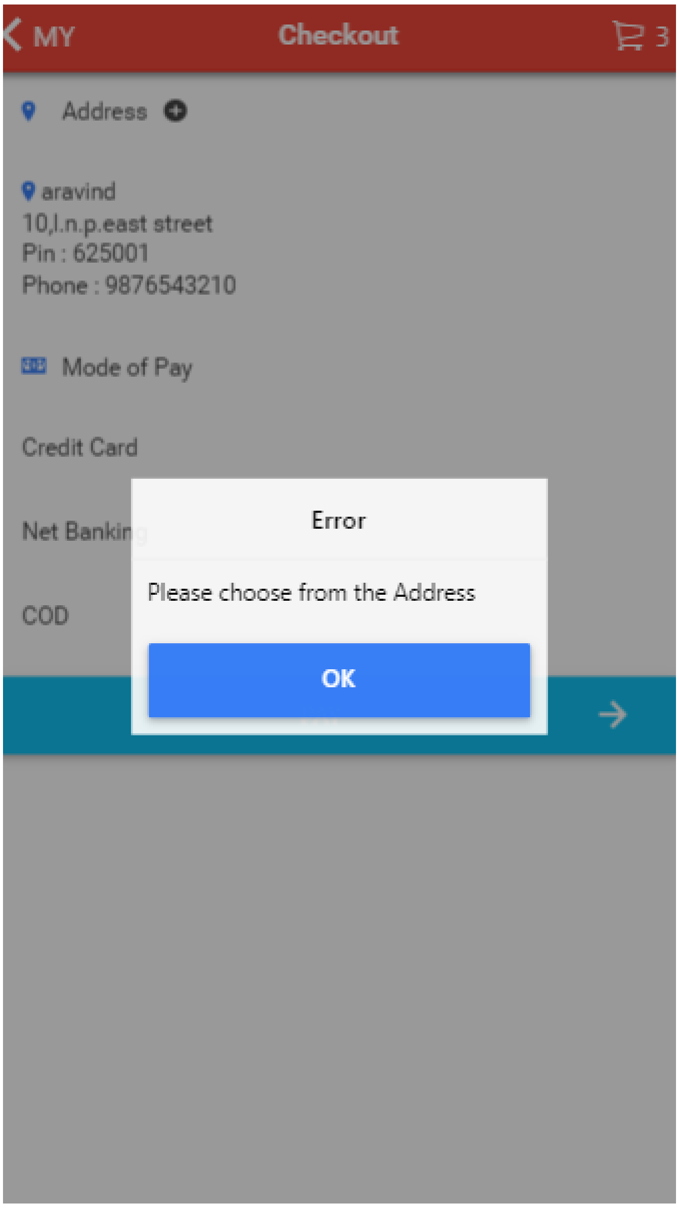
balajiponnuchamy@gmail.com

••••••••

Please note

Authentication Error

Fig 5.1 Authentication error



aravind
10,l.n.p.east street
Pin : 625001
Phone : 9876543210

Error

Please choose from the Address

Fig 5.2 Address field not selected error

This chapter “System testing” briefly explains about testing, test cases and outcome of the test for this project. Next chapter 6 conclusion deals with conclusion and future enhancement of the project.

CHAPTER 6

CONCLUSION

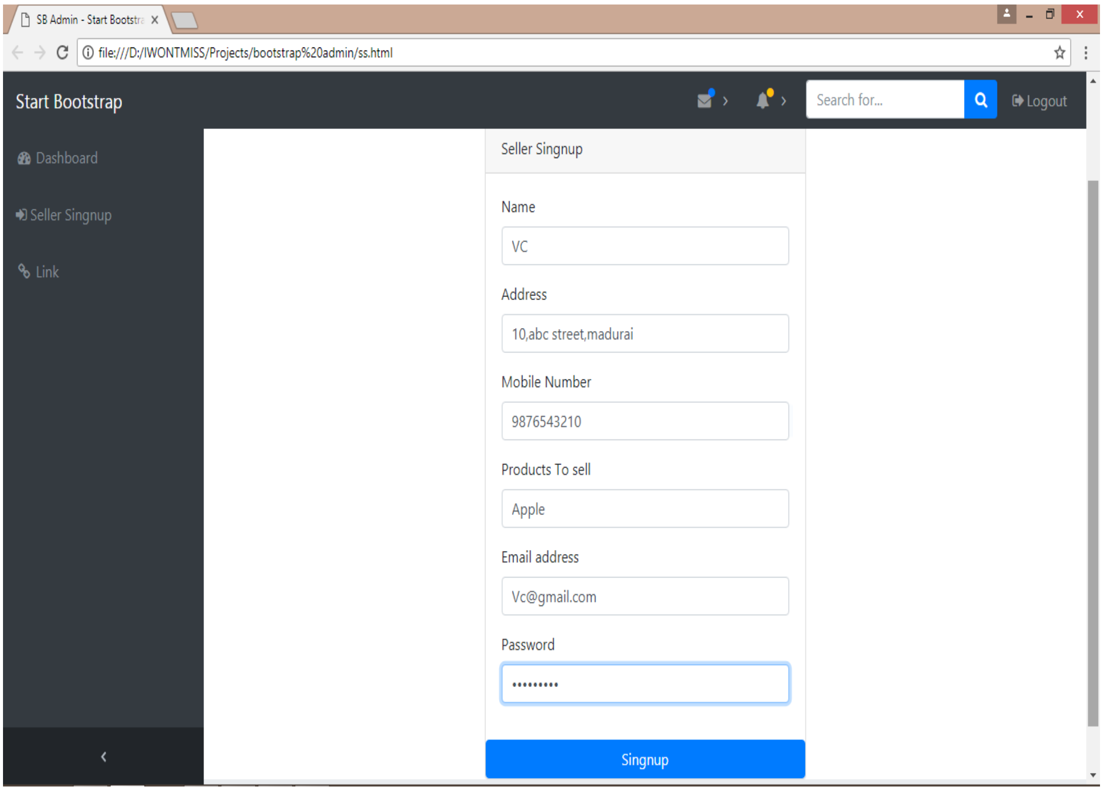
6.1 CONCLUSION

The IWONTMISS system is developed using Ionic Framework as Front end and firebase as Server and this project fully meets the objective. All the data of seller and buyer are maintained in the firebase which will help this project to make it easier as possible. The buyer and seller modules are successfully executed and meet the objective which it has been developed. Majority of the process in the system are done through online, this will help the seller and buyer to save their precious time. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the buyers and sellers associated with this system understands its advantage.

6.2 FUTURE ENHANCEMENT

As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment. Sub admin module can be added to control the traffic. Based on the customer needs the products and its seller will be expanded in this system. This IWONTMISS system can be hosted in webserver to make all the process are done through the online. Online payment can be enabled for buyer, which will help for the buyers as well as sellers. In future, reviews can be added for each seller, based on the reviews the customers can select the seller’s .In future the buyers and sellers need to pay the amount as service charge to access this website, which is also flexible.

APPENDIX-A: SCREEN SHOTS



Seller Singnup

Name

VC

Address

10,abc street,madurai

Mobile Number

9876543210

Products To sell

Apple

Email address

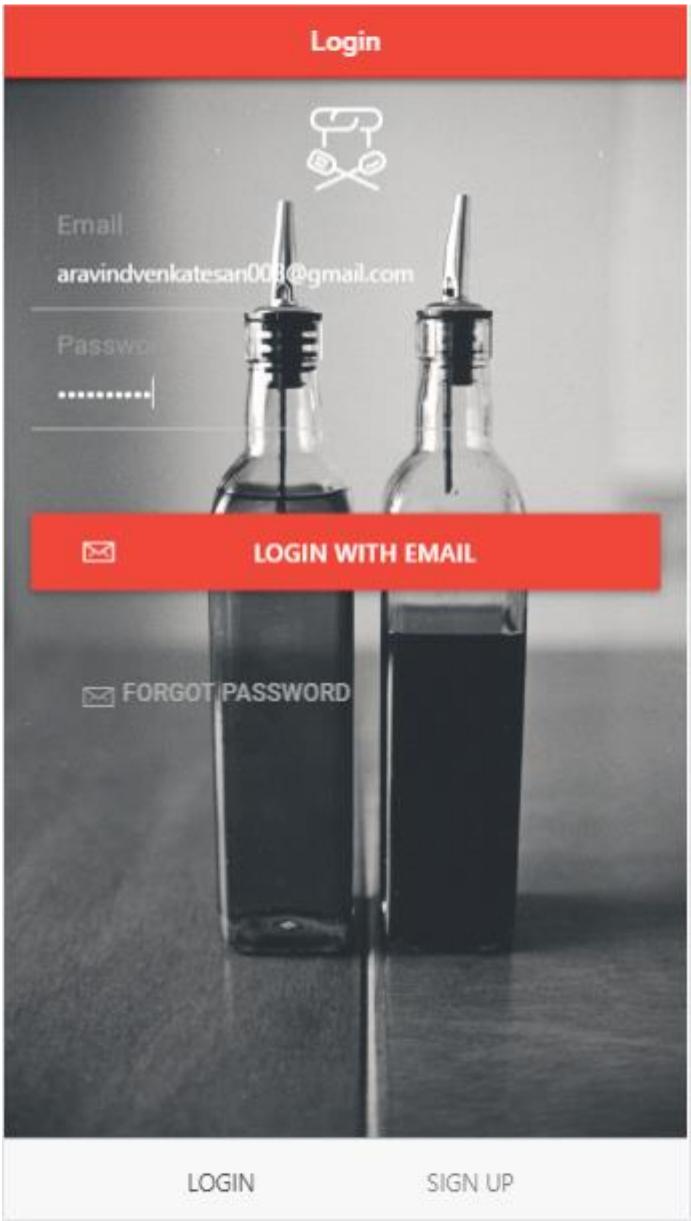
Vc@gmail.com

Password

•••••••••

Singnup

Fig A.1 Seller Signup



Login
Email
aravindvenkatesan008@gmail.com
Passwo
••••••••••
LOGIN WITH EMAIL
FORGOT PASSWORD
LOGIN
SIGN UP

Fig A.2 Login

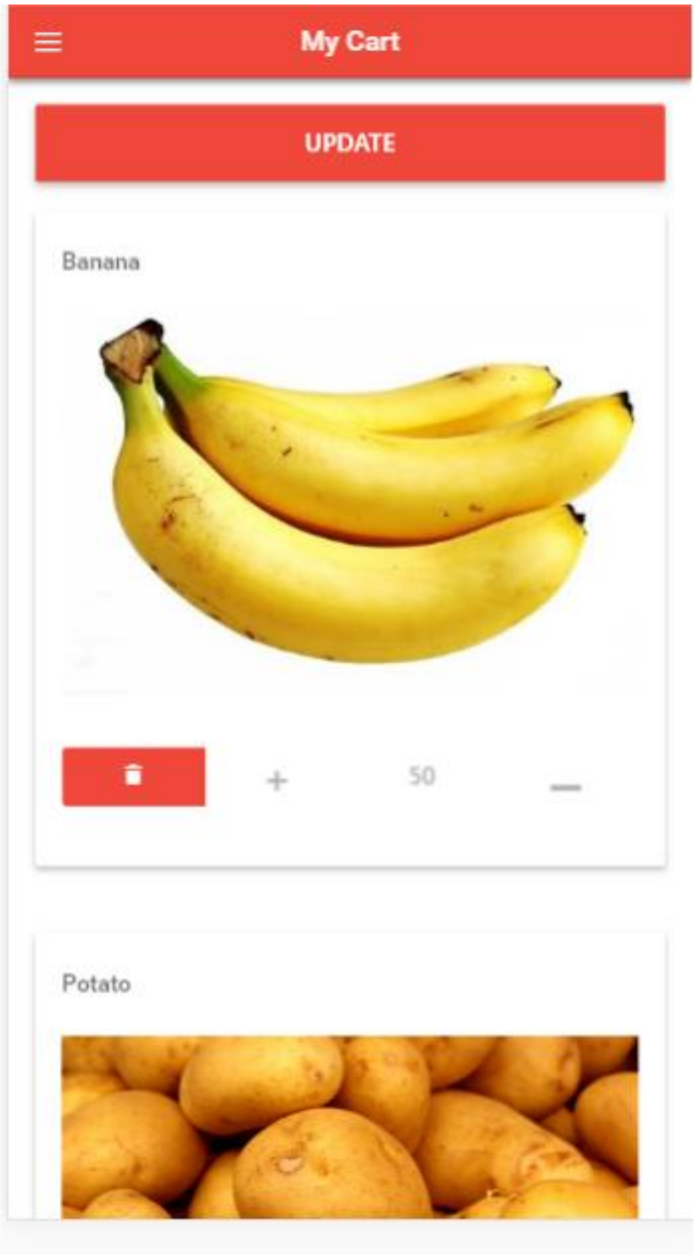
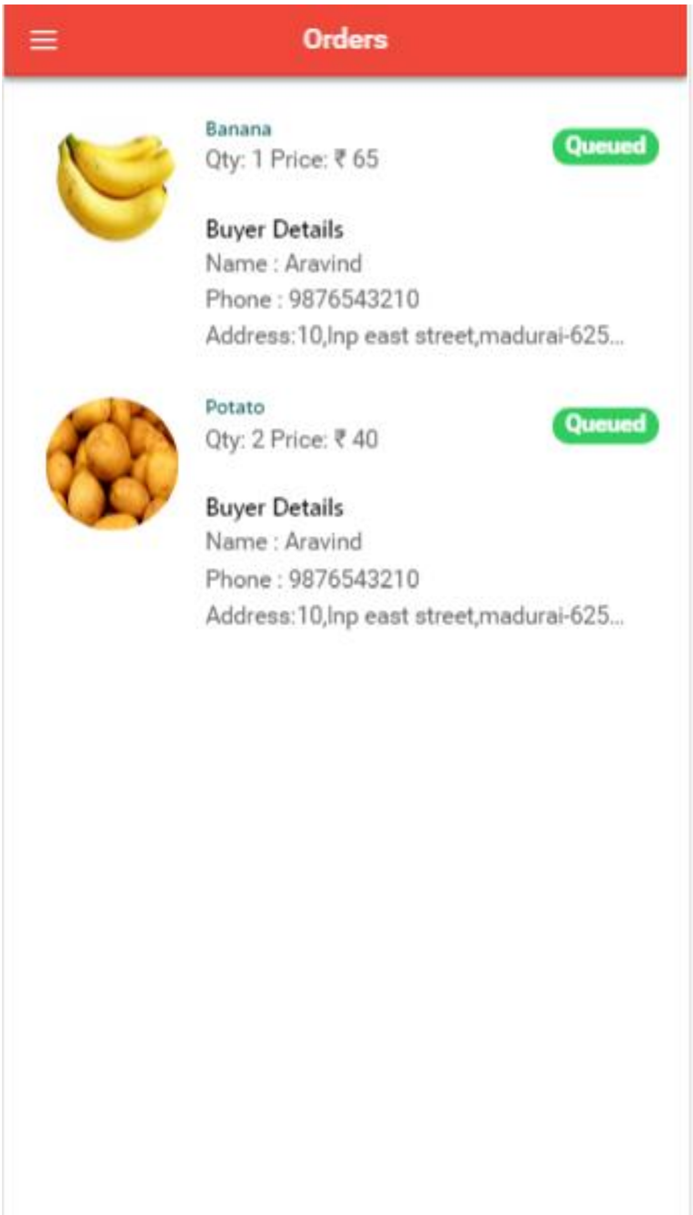


Fig A.3 Seller Update Page



Orders

Banana

Qty: 1 Price: ₹ 65

Buyer Details

Name : Aravind

Phone : 9876543210

Address:10,lnp east street,madurai-625...

Potato

Qty: 2 Price: ₹ 40

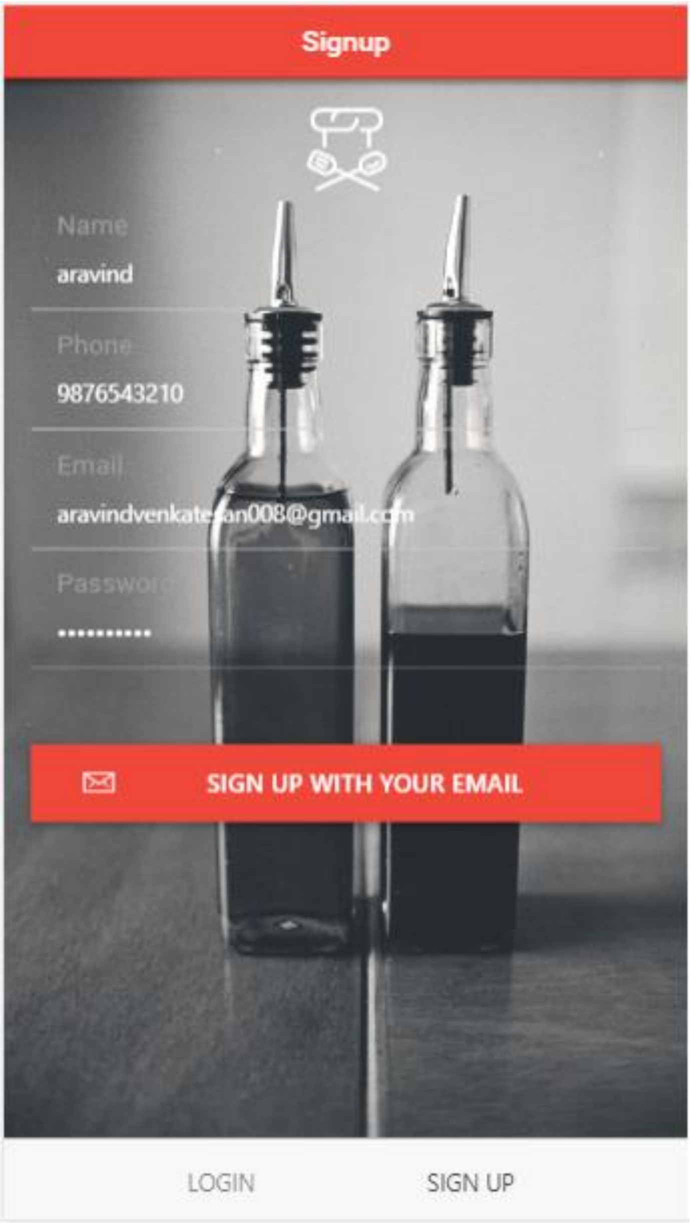
Buyer Details

Name : Aravind

Phone : 9876543210

Address:10,lnp east street,madurai-625...

Fig A.4 Order Details



Signup

Name

aravind

Phone

9876543210

Email

aravindvenkatesan008@gmail.com

Password

••••••••••

Fig A.5 Buyer Signup

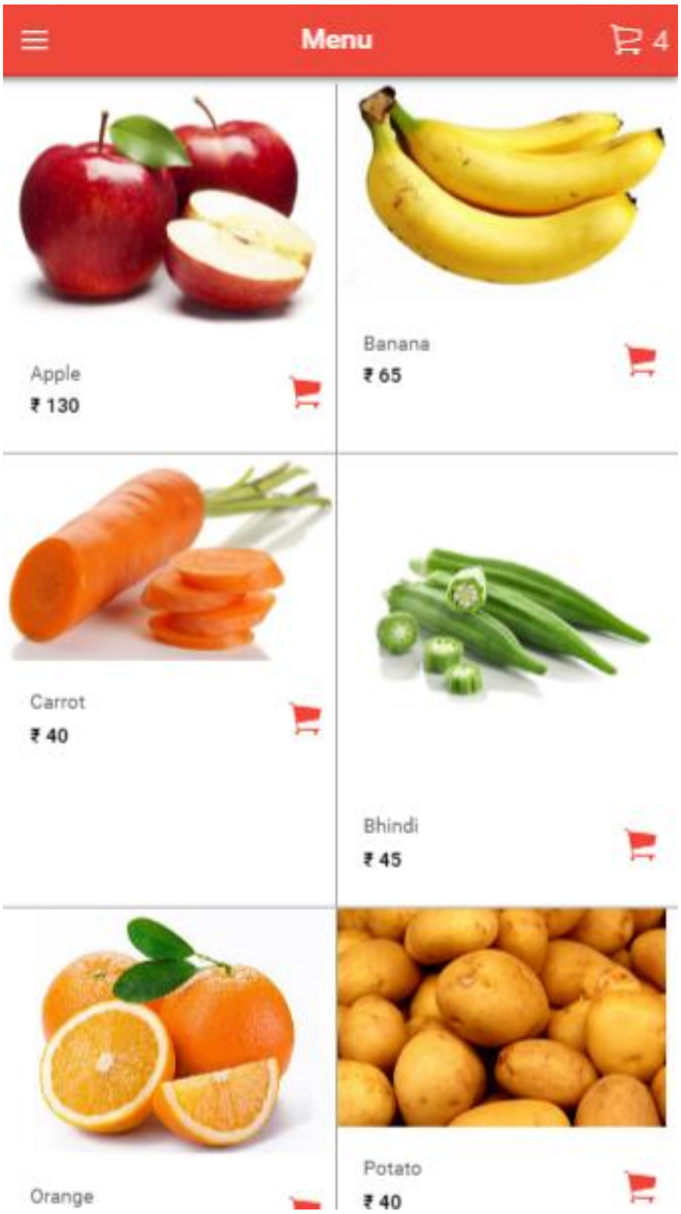
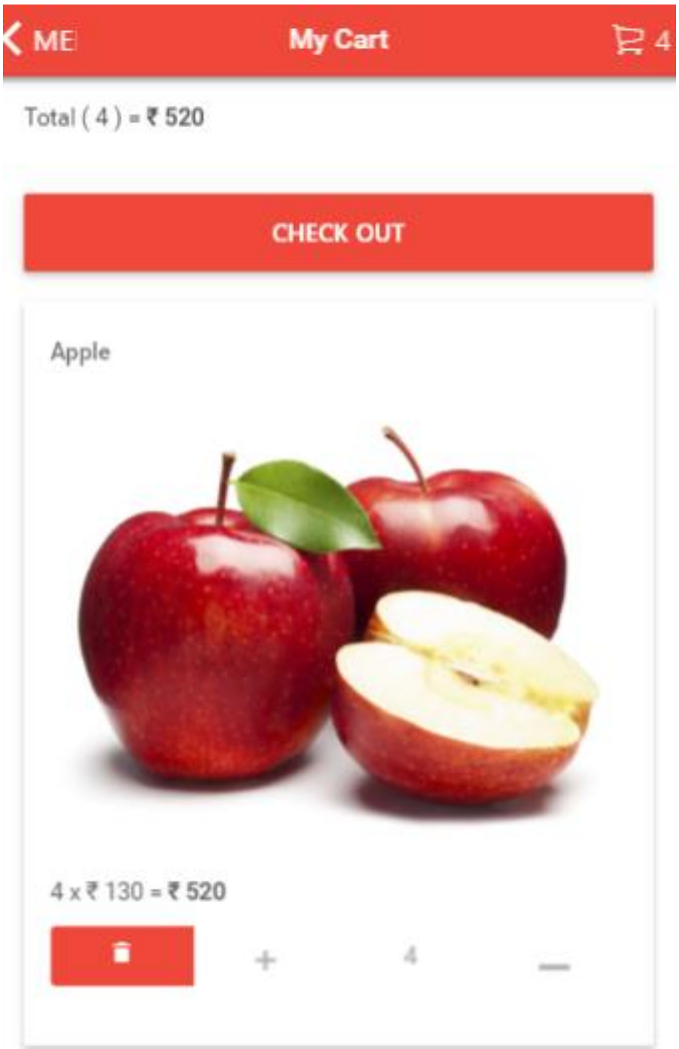


Fig A.6 Menu Page

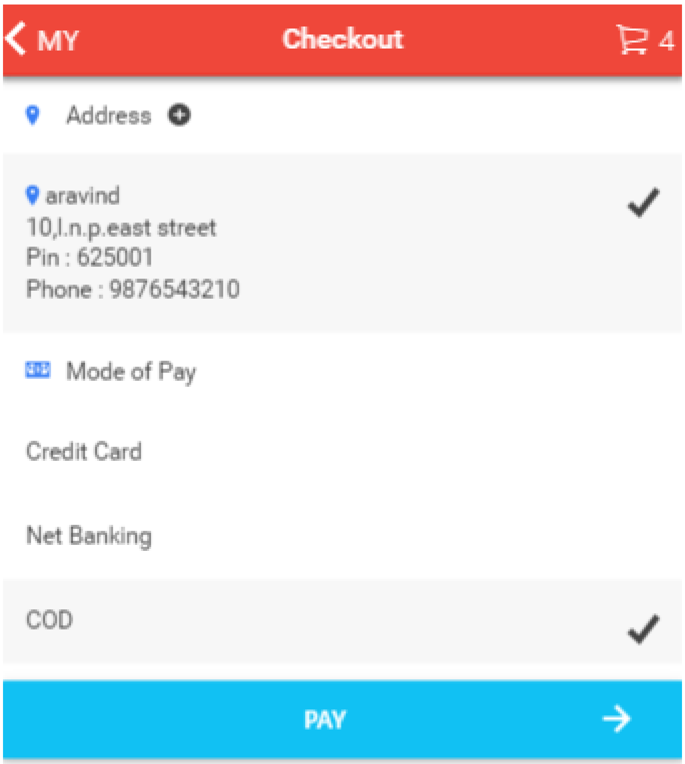


Total (4) = ₹ 520

Apple

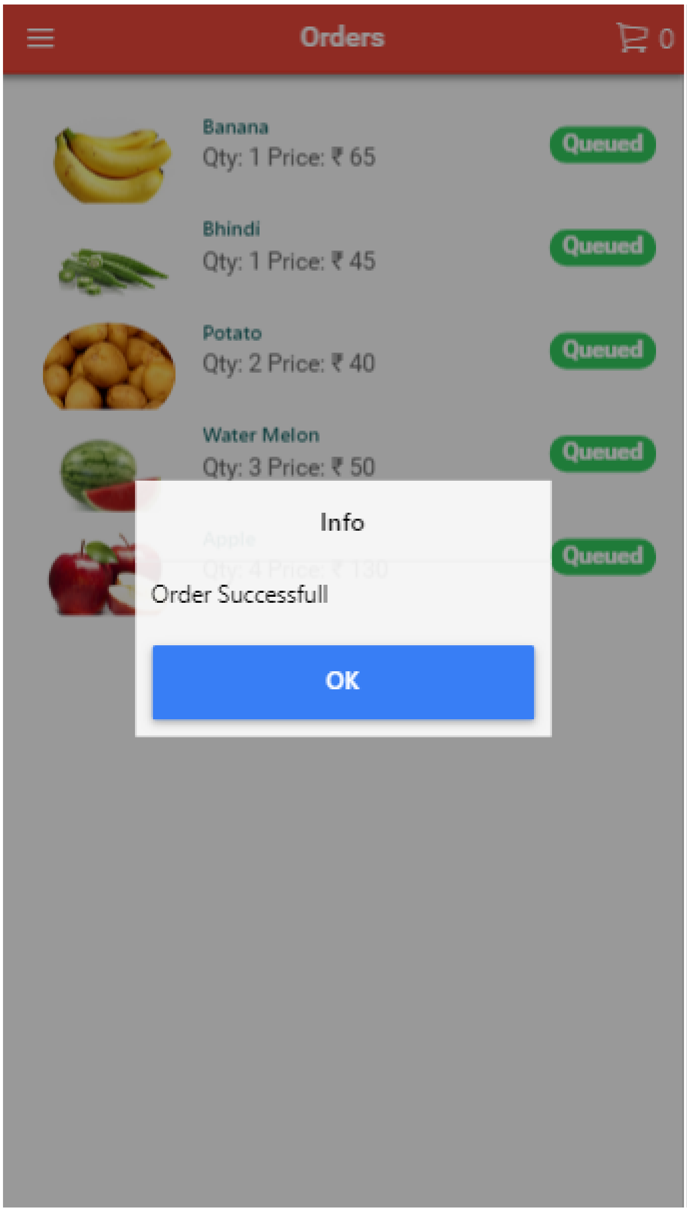
4 x ₹ 130 = ₹ 520

Fig A.7 Buyer Cart



aravind
10,l.n.p.east street
Pin : 625001
Phone : 9876543210 ✓

Fig A.8 Checkout



Banana
Qty: 1 Price: ₹ 65

Bhindi
Qty: 1 Price: ₹ 45

Potato
Qty: 2 Price: ₹ 40

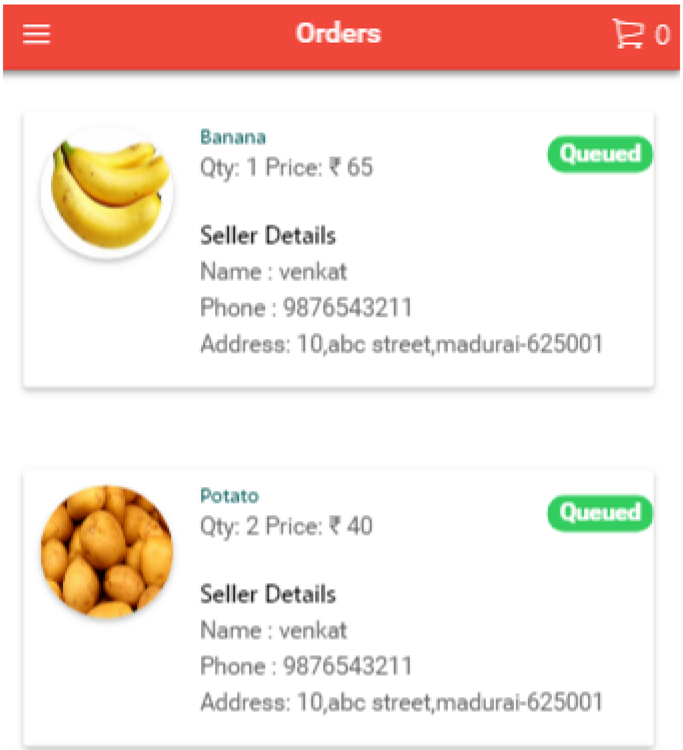
Water Melon
Qty: 3 Price: ₹ 50

Apple
Qty: 4 Price: ₹ 130

Info

Order Successfull

Fig A.9 Order Successfull



Banana

Qty: 1 Price: ₹ 65

Queued

Seller Details

Name : venkat

Phone : 9876543211

Address: 10,abc street,madurai-625001

Potato

Qty: 2 Price: ₹ 40

Queued

Seller Details

Name : venkat

Phone : 9876543211

Address: 10,abc street,madurai-625001

Fig A.10 Orders

APPENDIX-B: CODING

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-
      scalable=no, width=device-width">
    <title></title>



      <!-- IF using Sass (run gulp sass first), then uncomment below and remove the
      CSS includes above
    <link href="css/ionic.app.css" rel="stylesheet">
    -->



    <script src="lib/ionic/js/ionic.bundle.js"></script>

    <!-- cordova script (this will be a 404 during development) -->
    <script src="cordova.js"></script>

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">



<!--=========================================#1. Custom Header
      Part=====================================-->

  <!-- Ionic Meterial Header-->
      <script src="lib/ionic-material/dist/ionic.material.min.js"></script>
  <script src="lib/ionicuirouter/ionicUIRouter.js"></script>
      <link href="lib/ionic-material/dist/ionic.material.min.css" rel="stylesheet">
    <link
    href="https://fonts.googleapis.com/css?family=RobotoDraft%3A400%2C500%2
    C700%2C400italic" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">



  <!--Firebase Header-->
  <script src="https://cdn.firebase.com/js/client/2.2.4/firebase.js"></script>
  <script
    src="https://cdn.firebase.com/libs/angularfire/1.2.0/angularfire.min.js"></script>

  <script src="https://www.gstatic.com/firebasejs/live/3.0/firebase.js"></script>
  <script>
        // You have to replace this section with your custome config
         var config = {
   apiKey: "AIzaSyB0GR6cOM-fTAKS5UjBKZufBkJ8f7fzEO8",
   authDomain: "iwontmiss2018.firebaseapp.com",
   databaseURL: "https://iwontmiss2018.firebaseio.com",
   projectId: "iwontmiss2018",
   storageBucket: "iwontmiss2018.appspot.com",
   messagingSenderId: "619792632986"
  };
        firebase.initializeApp(config);
    </script>

 <!--
    ============================================================
    ==========================================-->
  <style type="text/css">
    .platform-ios .manual-ios-statusbar-padding{
      padding-top:20px;
    }
    .manual-remove-top-padding{
      padding-top:0px;
    }
    .manual-remove-top-padding .scroll{
      padding-top:0px !important;
    }
    ion-list.manual-list-fullwidth div.list, .list.card.manual-card-fullwidth {
      margin-left:-10px;
      margin-right:-10px;
    }
    ion-list.manual-list-fullwidth div.list > .item, .list.card.manual-card-fullwidth >
    .item {
      border-radius:0px;
      border-left:0px;
      border-right: 0px;
    }
  </style>

  <script src="js/app.js"></script>
  <script src="js/controllers.js"></script>
  <script src="js/routes.js"></script>
  <script src="js/services.js"></script>
  <script src="js/directives.js"></script>




</head>
```

```
<body ng-app="app" animation="slide-left-right-ios7">
<div style="">

  <!--#5. indexCtrl -->
  <ion-side-menus  ng-controller="indexCtrl" data-componentid="side-menu21"
    style="" enable-menu-with-back-views="false">
      <ion-side-menu-content  >
          <ion-nav-bar class="bar-assertive" >

              <ion-nav-back-button></ion-nav-back-button>



              <ion-nav-buttons side="left">
                  <!-- #3.  $root.extras -->
                  <button ng-if="$root.extras" class="button button-icon button-clear ion-
    navicon" menu-toggle="left">
                  </button>
              </ion-nav-buttons>

              <ion-nav-buttons side="right">
                <!-- #3.  $root.extras -->
                <!-- #4.  get_total() -->
                <button ui-sref="myCart" ng-if="$root.extras" class="button button-icon
    button-clear ion-ios-cart-outline">
                  {{get_total()}}
                </button>
              </ion-nav-buttons>

          </ion-nav-bar>
          <ion-nav-view></ion-nav-view>
      </ion-side-menu-content>
  <!-- #3.  $root.extras -->
  <ion-side-menu side="left" id="side-menu21" ng-if="$root.extras">

      <ion-header-bar class="expanded">
        <img ng-src="{{&apos;img/fk/&apos;+ user_info.photoURL +&apos;.jpg&apos;}}" class="avatar motion spin fade">
        <div class="menu-bottom">
          {{user_info.displayName}}
        </div>
      </ion-header-bar>

      <ion-content class="stable-bg has-expanded-header">
          <ion-list data-componentid="list1" class=" " id="menu-list1">

              <ion-item data-componentid="list-item1" style="" menu-close="" ui-sref="menu2" id="menu-list-item1" class="item-icon-left  ">
                  <i class="icon ion-ios-home-outline"></i>Menu</ion-item>
              <ion-item data-componentid="list-item2" style="" menu-close="" ui-sref="offers" id="menu-list-item2" class="item-icon-left  ">
                  <i class="icon ion-ios-pricetag-outline"></i>Offers</ion-item>
              <ion-item data-componentid="list-item8" style="" menu-close="" ui-sref="myCart" id="menu-list-item8" class="item-icon-left  ">
                  <i class="icon ion-ios-cart-outline"></i>My Cart</ion-item>
              <ion-item data-componentid="list-item3" style="" menu-close="" ui-sref="lastOrders" id="menu-list-item3" class="item-icon-left  ">
                  <i class="icon ion-ios-clock-outline"></i>Orders</ion-item>
              <ion-item data-componentid="list-item4" style="" menu-close="" ui-sref="favourite" id="menu-list-item4" class="item-icon-left  ">
                  <i class="icon ion-ios-star-outline"></i>Favourite</ion-item>
              <ion-item data-componentid="list-item5" style="" menu-close="" ui-sref="settings" id="menu-list-item5" class="item-icon-left  ">
                  <i class="icon ion-ios-gear-outline"></i>Settings</ion-item>
          <ion-item data-componentid="list-item6" style="" menu-close="" ui-
   sref="support" id="menu-list-item6" class="item-icon-left  ">
            <i class="icon ion-ios-help-outline"></i>Support</ion-item>
          <ion-item ng-click="logout()" data-componentid="list-item7" style=""
   id="menu-list-item7" class="item-icon-left  ">
            <i class="icon ion-log-out"></i>Logout</ion-item>
        </ion-list>
      </ion-content>

    </ion-side-menu>
  </ion-side-menus>
 </div>
 </body>
</html>

.factory('fireBaseData', function($firebase) {
   var ref = new Firebase("https://iwontmiss2018.firebaseio.com/"),
  refCart = new Firebase("https://iwontmiss2018.firebaseio.com/cart"),
  refUser = new Firebase("https://iwontmiss2018.firebaseio.com/users"),
  refCategory = new Firebase("https://iwontmiss2018.firebaseio.com/category"),
  refOrder = new Firebase("https://iwontmiss2018.firebaseio.com/orders"),
  refFeatured = new Firebase("https://iwontmiss2018.firebaseio.com/featured"),
  refMenu = new Firebase("https://iwontmiss2018.firebaseio.com/menu");
 return {
  ref: function() {
   return ref;
  },
  refCart: function() {
   return refCart;
  },
  refUser: function() {
   return refUser;
  },
  refCategory: function() {
   return refCategory;
  },
  refOrder: function() {
   return refOrder;
  },
  refFeatured: function() {
   return refFeatured;
  },
  refMenu: function() {
   return refMenu;
  }
 }
})


.factory('sharedUtils',['$ionicLoading','$ionicPopup',
   function($ionicLoading,$ionicPopup){



  var functionObj={};


  functionObj.showLoading=function(){
   $ionicLoading.show({
    content: '<i class=" ion-loading-c"></i> ', // The text to display in the loading
   indicator
    animation: 'fade-in', // The animation to use
    showBackdrop: true, // Will a dark overlay or backdrop cover the entire view
    maxWidth: 200, // The maximum width of the loading indicator. Text will be
   wrapped if longer than maxWidth
    showDelay: 0 // The delay in showing the indicator
   });
  };
  functionObj.hideLoading=function(){
   $ionicLoading.hide();
  };


  functionObj.showAlert = function(title,message) {
   var alertPopup = $ionicPopup.alert({
    title: title,
    template: message
   });
  };

  return functionObj;

}])
 .factory('sharedCartService',
   ['$ionicPopup','fireBaseData','$firebaseArray',function($ionicPopup,
   fireBaseData, $firebaseArray){

  var uid ;// uid is temporary user_id
  var cart={}; // the main Object



  //Check if user already logged in
  firebase.auth().onAuthStateChanged(function(user) {
   if (user) {
    uid=user.uid;
    cart.cart_items = $firebaseArray(fireBaseData.refCart().child(uid));
   }
  });
```

REFERENCES

[1] Brad Dayley and Brendan Dayley,”Angularjs,JavaScript,andJQuery”, Sams Publishing, 2015.

[2] Cory Gackenheimer,”Node.js Recipes: A Problem-Solution Approach”, Apress, 2013.

[3] Fritz Schneider and Thomas Powell,”Java Script: The Complete Reference”, McGraw-Hill, 2001.

[4] Roger S. Pressman,” Software Engineering: A Practitioner's Approach”, McGraw-Hill Education, 2014.

[5] Thomas Powell,”HTML & CSS: The Complete Reference Fifth Edition”, McGraw Hill Professional, 2010.

[6] **https://angularjs.org/**

[7] **https://firebase.google.com/docs/**

[8] **https://ionicframework.com/**

[9] **https://nodejs.org/en/docs/**