FINAL PROJECT REPORT

For

Electro Basket App

Prepared by

<Aman Khandelwal(U101116FCS007)

Arshita Batra(U101116FCS284)

Chinju Mary George(U101116FCS025)

Siddhant Shah(U101116FCS129)

Shashwat Shah(U101116FCS112)>

<Software Engineering CS-301>

TABLE OF CONTENTS

SOFTWARE DESIGN SPECIFICATION

CHAPTER 1

SOFTWARE REQUIREMENT SPECIFICATION

1.Introduction

1.1 Purpose

This Product is an android app that will be useful for the various Businesses running in our Country.

This app will interact the Salesmen of the company to the Sub-Dealers and the Dealers and request the order of the particular product required by the Shop in his locality or area.

1.2 Document Conventions

There are no such conventions followed in this document.

1.3 Intended Audience and Reading Suggestions

This Document is intended for the Businessmen who will be our primary customers,Developers and Testers. This Product has been implemented under the guidance of college professors.

1.4 Product Scope

This goal of the application is to make the functioning of the businesses smoother and much easier.The Deals within the company will be more transparent and the Dealer can identify which products are in demand.

1.5 References

IEEE SRS template
UML Use Case Diagram

1.6 Terminology

Term	Description
User	Dealer, Sub-dealer and Salesman of the company.
Database	Collection of information of different topics required for the implementation of project.
Graph	The graph will denote the sales of goods geographically.

2. Overall Description

2.1 Product Perspective

The Application is being developed for Electrical Businessmen of Pune, Maharashtra. This is a self sustained Product which will helps the Dealer of the Particular state to see the orders placed by the Sub-dealer and the Salesmen of company and helps to meet their requirement. Their will be 3 kinds of users-
The Dealer: He can see each and every order placed by the Sub-Dealer and the Salesman and then fulfil their requirements.
The Sub-dealer: He can see the orders placed by the Salesmen and place the order to the dealer if he will unable to fulfil the product requirement of the salesmen.
The Salesman: He can place the order made by the shopkeepers in his desired area.

2.2 Product Functions

The major Function of the product is that the Salesmen and the Sub-dealer must be able to place the orders in the app and which will be visible to the dealer and he can keep track of all the functioning of the orders and stocks.

2.3 User Classes and Characteristics

The App will support three privileges,Dealer,Sub-dealer and the Salesmen.
The Dealer can -

- See the Orders placed by the Dealer, Sub-dealer and Salesmen.
- View all the activities of the Sub-dealer and Salesmen.
- Fulfil the demands of Sub-dealer and Salesmen.

The Sub-dealer can -

- Place an Order to the dealer
- See the Orders placed by the Sub-dealers
- Cancel the Order Placed to the Dealer

The Salesman can -

- Place an order to the Sub-dealer.
- Cancel the Placed Order.

2.4 Operating Environment

Our Application will be primarily be Operated on the Android Platform and works on all the upgraded or latest versions of android.

2.5 Design and Implementation Constraints

This app won't work for IOS devices and android versions lower than android Ice Cream Sandwich.

2.6 User Documentation

(Video will be added after completion of the project).

2.7 Assumptions and Dependencies

The applications data can be affected if the **database** is refreshed or not properly maintained and can lead to confusion among the customers.

3. External Interface Requirements

3.1 User Interfaces

Front End Software:
An Android GUI which will have buttons, navigation bar and 4-5 tabs which will differ from Dealer to Salesmen and a my Orders page will display the orders placed by the Customers and will also have a login and Sign up page.

Back end:
The project will be developed on the Android Studio Platform and the **database** used to store the needs and demands will be either fire base or SQL.

3.2 Hardware Interfaces

Android Phones.
Android Tablets.

3.3 Software Interfaces

- We use FireBase for **database** purpose.
- It's the Best for Mobile Developing Application

3.4 Communications Interfaces

1. The Project is Android Based and hence does not require any server protocols and HTTP as such.
2. The App Must Verify the mail id during the sign-up.
3. Internet connection will be used for communicating with the ***database*** and server.

4. System Features

This template explains all the features and description of the application.

4.1 Login/SignUp

4.1.1 Description and Priority

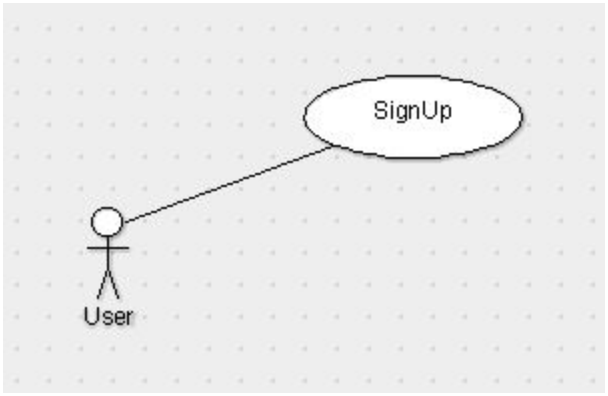
The system's users are given a workspace from which all of their communication will take place where the user can login to the app and will be able to access the App.The priority here is that the user will be authenticated to the app.

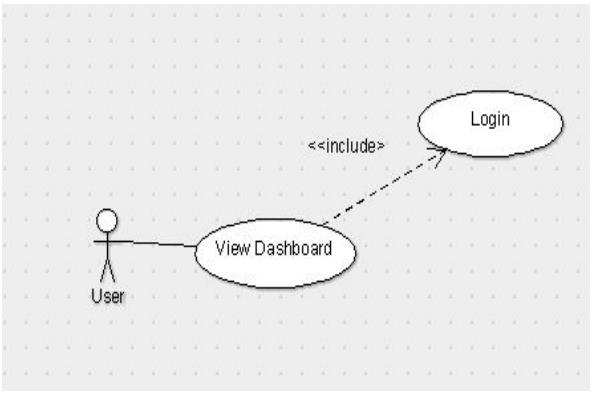
4.1.2 Stimulus/Response Sequences

If the user wants to add any personal details such as name or address it will be updated and the app will create an artifact with specified metadata.

4.1.3 Functional Requirements

The Login/Sign-up Page will help to authenticate the user and display an error if the password or the mobile number is invalid.





4.2 Place Orders

4.2.1 Place Orders(Salesman)

4.2.1.1 Description and Priority

The priority here is that the Salesman can place orders based on the requirements.

4.2.1.2 Stimulus/Response Sequences

If the Salesman want to add or place any order there will be a "place- order" button which when pressed will order all the goods and their quantity .

4.2.1.3 Functional Requirements

In this the Salesman will be provided a list of goods from which order can be placed

4.2.2 Place Orders(Sub-dealer)

4.2.1.1 Description and Priority

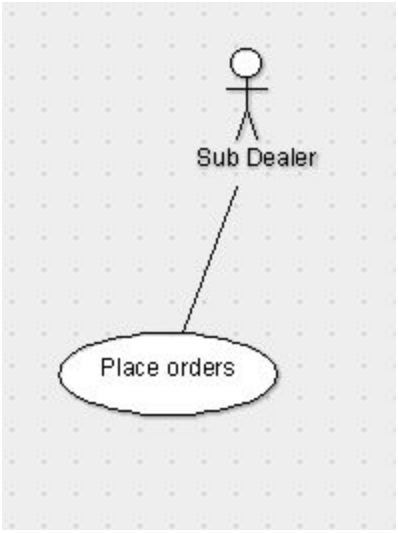
The priority here is that the Sub-dealer can place order based on the requirements.

4.2.1.2 Stimulus/Response Sequences

If the Sub-dealer want to add or place any order there will be a “place- order” button which when pressed will order all the goods and their quantity .

4.2.1.3 Functional Requirements

In this the Sub-dealer will be provided a list of goods from which order can be placed



4.3 View Order

4.3.1 View Order(Dealer)

4.3.1.1 Description and priority

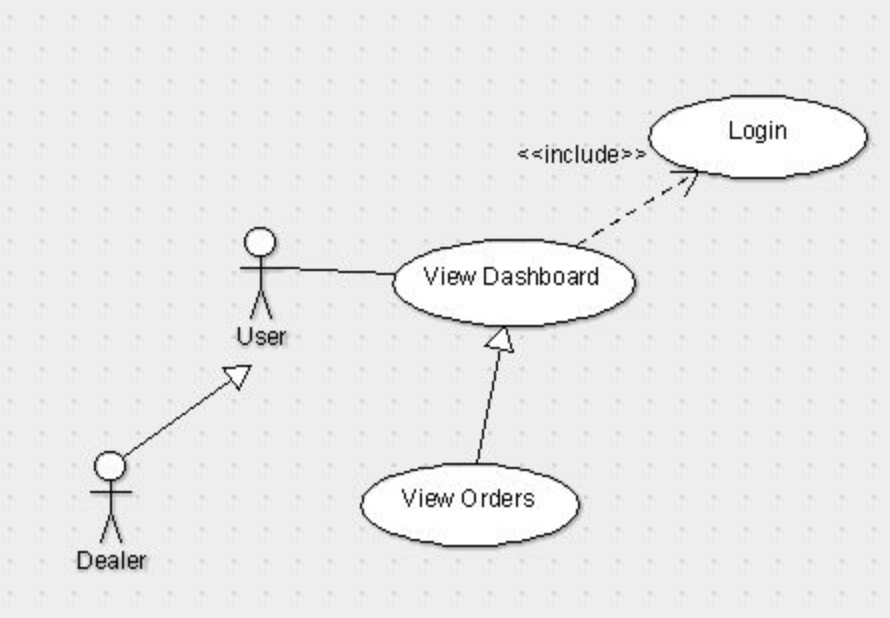
This function will enable the Main Dealer to view all the Orders placed by the Sub-dealer and keep the record of all the Orders.

4.3.1.2 Stimulus/Response Sequences

There will be a ‘View order’ button which when pressed will show list of all the orders and the quantity placed by the Sub-dealer and the time at which the Order is Placed.

4.3.1.3 Functional Requirements

The View Order function will enable the Main Dealer to view and keep track of all the Orders and the time at which order is placed by Sub-dealer.



4.3.2 View Order(Sub-Dealer)

4.3.2.1 Description and priority

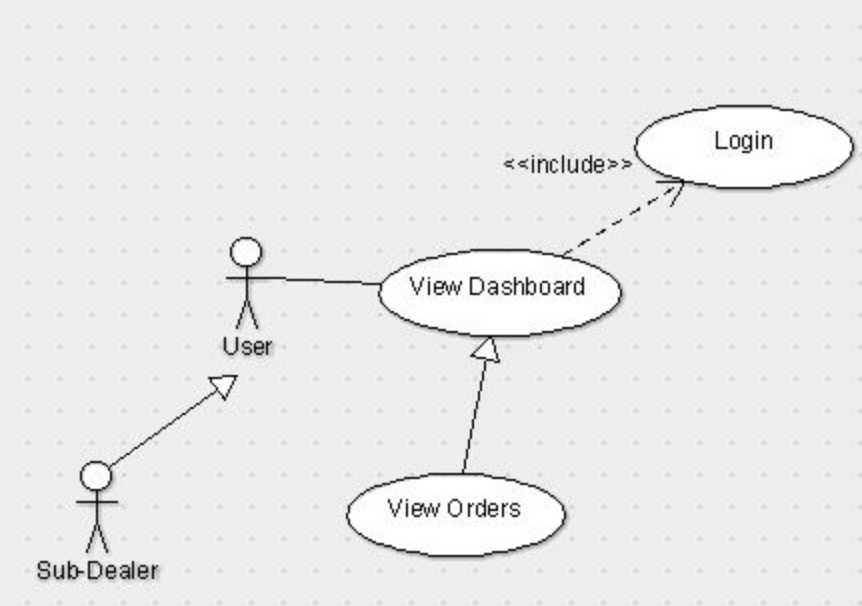
This function will enable the Sub Dealer to view all the Orders placed by the Salesman and keep the record of all the Orders.

4.3.2.2 Stimulus/Response Sequences

There will be a ‘View order’ button which when pressed will show list of all the orders and the quantity placed by the Salesman and the time at which the Order is Placed.

4.3.2.3 Functional Requirements

The View Order function will enable the Sub Dealer to view and keep track of all the Orders and the time at which order is placed by Salesman.



4.3.3 View Order(Salesman)

4.3.3.1 Description and priority

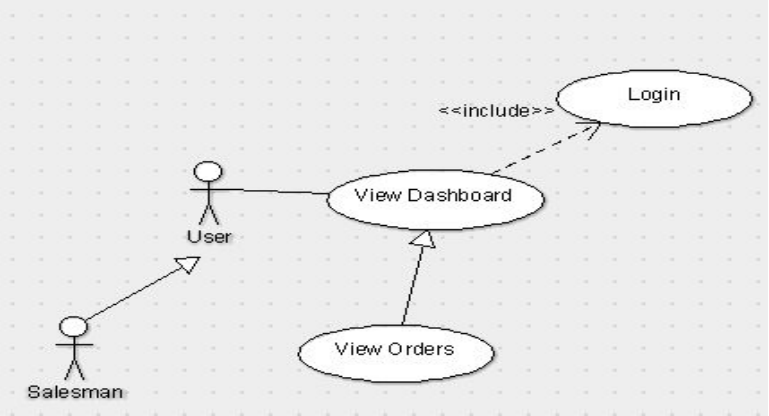
This function will enable the Salesman to view all the Orders he placed and keep the record of all the Orders.

4.3.3.2 Stimulus/Response Sequences

There will be a ‘View order’ button which when pressed will show list of all the orders and the quantity placed by the Salesman and the time at which the Order is Placed.

4.3.3.3 Functional Requirements

The View Order function will enable the Salesman to view and keep track of all the Orders and the time at which order is placed by himself.



4.4 Details

4.4.1 Description and priority

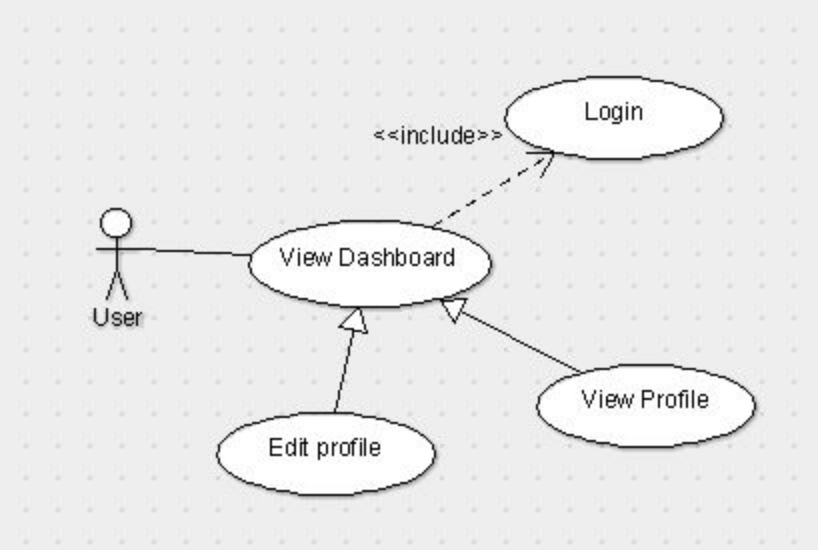
This function will allow the user to view all the details which will include name ,address , phone number,etc.

4.4.2 Stimulus/Response Sequences

In this there will be a button “details” in which user can view its personal information including name ,address ,phone ,number etc.There will be an update button in which user can update its personal details.

4.4.3 Functional Requirements

If the user is logged in, then he can edit his detail but if he is not a user than he needs to sign up and only then he can edit his details.



4.5 Analytics

4.5.1 Description and priority

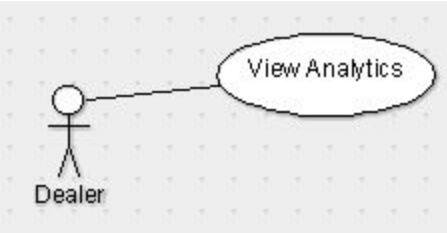
This function will describe the order of goods **graphically** and will explain which goods are order the most and vica versa to the Dealer..

4.5.2 Stimulus/Response Sequences

In this user will be provided a button “analytics” in this there will be a **graphical** representation of order and quantity with there proportion.

4.5.3 Functional Requirements

This function will help to see a **graphical** representation of all the stocks of the particular area.



4.6 .Executive Mapping

4.6.1 Description and priority

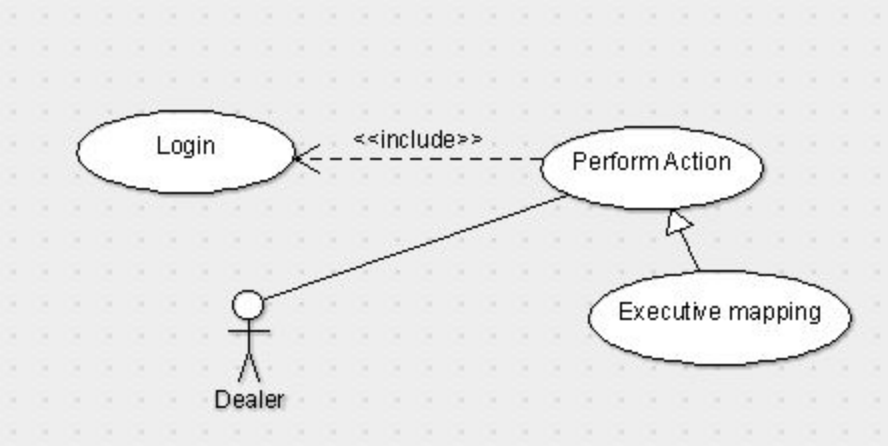
This Function will help to see the location of the Salesman and the time at which he went to a particular store.

4.6.2 Stimulus/Response Sequences

In this user will be provided a button “TRACK” in this there will be a complete record of salesman locations where he went.

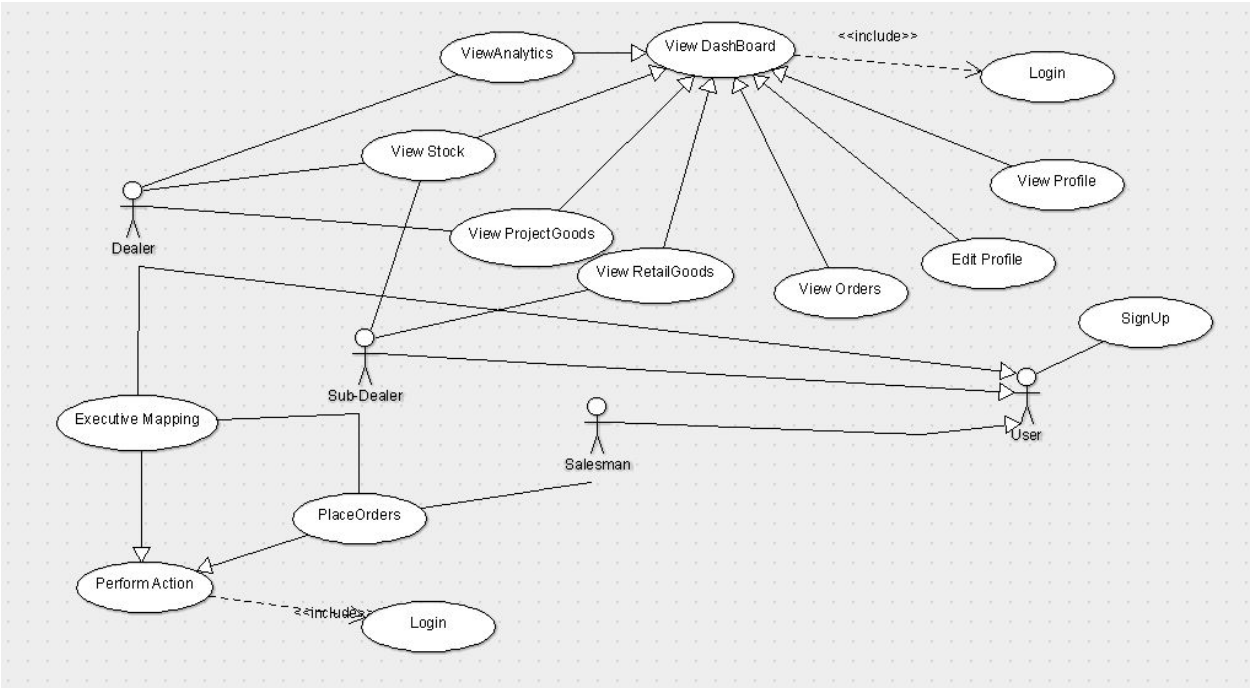
4.6.3 Functional Requirements

This function will help to see a **location** of all the salesman of the particular area.



Make a copy of 1 login use case. And put all arrows to it as include.

The following is the use case diagram for the application:



Use Case Description Table :

Use Case Title	***Description***
1.Dealer Login	The user Dealer needs to login to perform any actions
2.Sub-Dealer Login	The user Sub-Dealer needs to login to perform any actions
3. Salesman Login	The user Salesman needs to login to perform any actions
4. SignUp	New user needs to Sign up and select its respective category to further perform actions.
5.View Orders(Dealer)	Dealer can view order that have been placed.
6. View Orders(Sub-Dealer)	Sub-Dealer can view order that have been placed.
7. View Orders(Salesman)	Salesman can view order that have been placed.
8.Place Orders(Salesman)	Salesman can place or add order of respective goods.
9. Place Orders(Sub-Dealer)	Sub-Dealer can place or add order of respective goods.
10.View Stock	This Function will Help to View the total Stock Available

	with the Dealer.
11.Executive Mapping	This Function Will help to see the location of the Salesman and the time at which he went to a particular store.
12.View Profile	The user can view its personal details.
13.Edit Profile	The user can Edit his/her own Details.
14.Goods	There will be two type of goods i.Retail Goods(For Small Sub-Dealers) ii.Project Goods(For Large Projects)
15.Analytics	The user can view and analyse the sales of goods geographically.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Every action-response must work smoothly with minimum delays possible. The maximum number of user that is supported by the system are 15 users approximately, users exceeding the maximum number can hamper the app or can cause delay.

5.2 Safety Requirements

If there is any damage to the **Database** or if the Application crashes then a mail will generated by the firebase and the Firebase Analytics Report is generated by the firebase and mailed to the admin. Also the **Database** which we use is Google's Firebase which never crashes and has excellent safety measures.

5.3 Security Requirements

This Application Uses a Real time **Database** storage just like many other applications. Also keeping in mind the safety measures only the Authorized Logged in users can view and place orders and no external person can access the information.

5.4 Software Quality Attributes

- **AVAILABILITY:** The Order placed should be available on the specified date and specified time as many Dealers will check the previous records made by the desired people.
- **CORRECTNESS:** The Order should reach the correct Sub-Dealer and Dealer with no loss.
- **MAINTAINABILITY:** The **Database** must be Correctly maintained and updated regularly
- **USABILITY:** The App must satisfy all the needs of the salesmen and the Sub-Dealers

5.5 Business Rules

There Are No Business Rules As Such

6. Other Requirements

No Such Requirements 8

CHAPTER 2

SOFTWARE DESIGN SPECIFICATION

1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, state diagram,class diagram and other supporting requirement information.

1.1 Purpose of this document

This document will define the design of the Business Application. It contains specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document.

1.2 Scope of the development project

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

a. Application for the Business which have proper Hierarchy

b. Dealer can retrieve the information about the Sub Dealer.

c.Ease in Maintaining a Record.

Out of Scope:

a. Cost Related Issues

b. Sub Dealer Can only view its own Stock.

1.3 Definitions, acronyms, and abbreviations IEEE: Institute of Electrical and Electronics Engineers

IEEE: Institute of Electrical and Electronics Engineers
SDS: Software Design Specification

1.4 Overview of document

This SDS is divided into 5 sections with various sub-sections. The sections of the Software Design Document are:

1. **Introduction:** describes about the document, purpose, scope of development project definitions and abbreviations used in the document.

2. **Logical Architecture**: describes Logical Architecture Description and Components.

3. **Execution Architecture:** defines the runtime environment, processes, deployment view.

4. **Design Decisions and Trade-offs:** describes the decisions taken along with the reason as to why they were chosen over other alternatives.

5. **Appendices:** describes subsidiary matter if any.

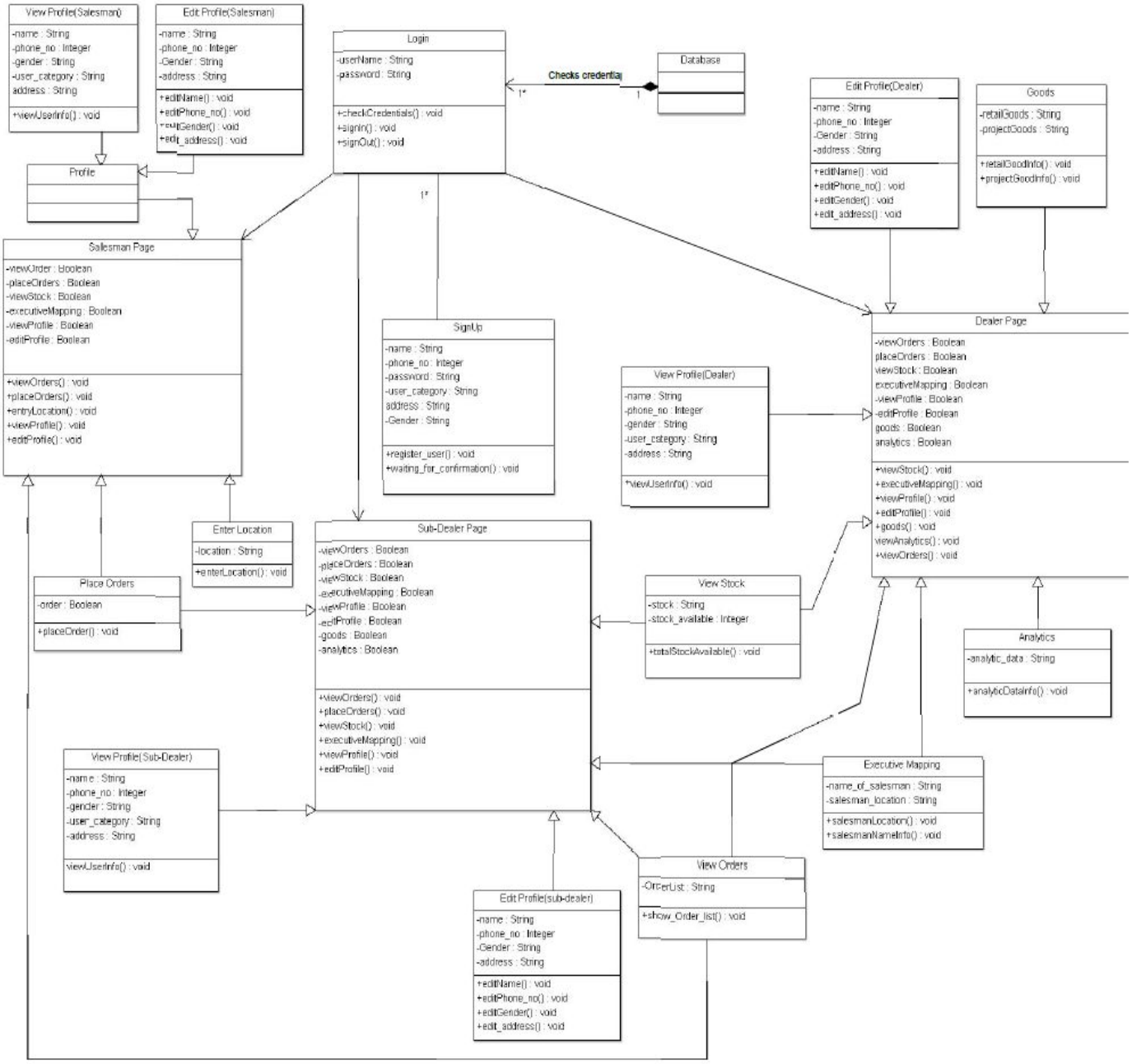
2.1 Logical Architecture Description

2.1.1 Class Diagram explanation:

Most of the classes extends AppCompatActivity class which is being shown by association linkage. It is being shown by black-coloured diamond. The classes which extends AppCompatActivity are: Login,SignUp,Dealer Page, Sub-Dealer page, Salesman, ViewProfile_Dealer, Edit_Profile_Dealer,ViewProfile_Sub-Dealer, Edit_Profile sub-Dealer,
ViewProfile_Salesman, Edit_Profile Salesman,View Stock,goods,place orders,Executive mapping,place order,analytics and view Orders .

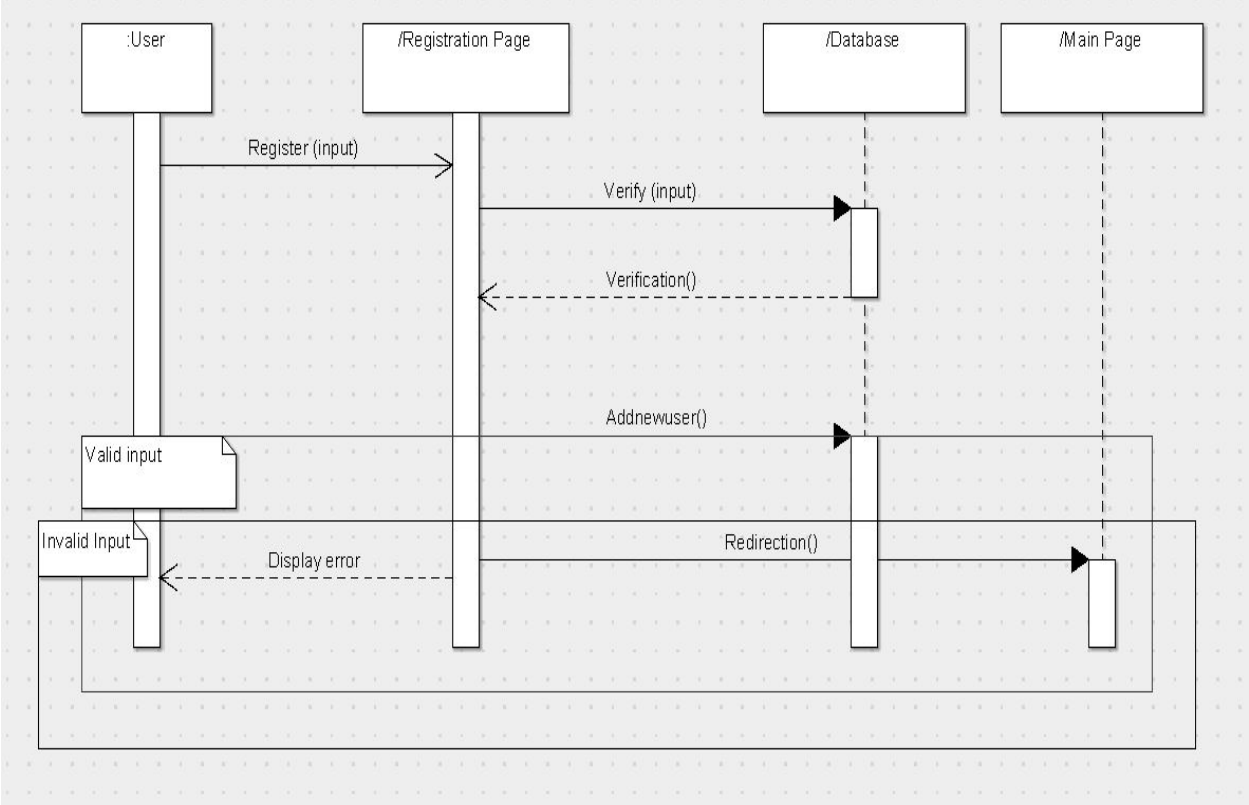
3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

Class Diagram:

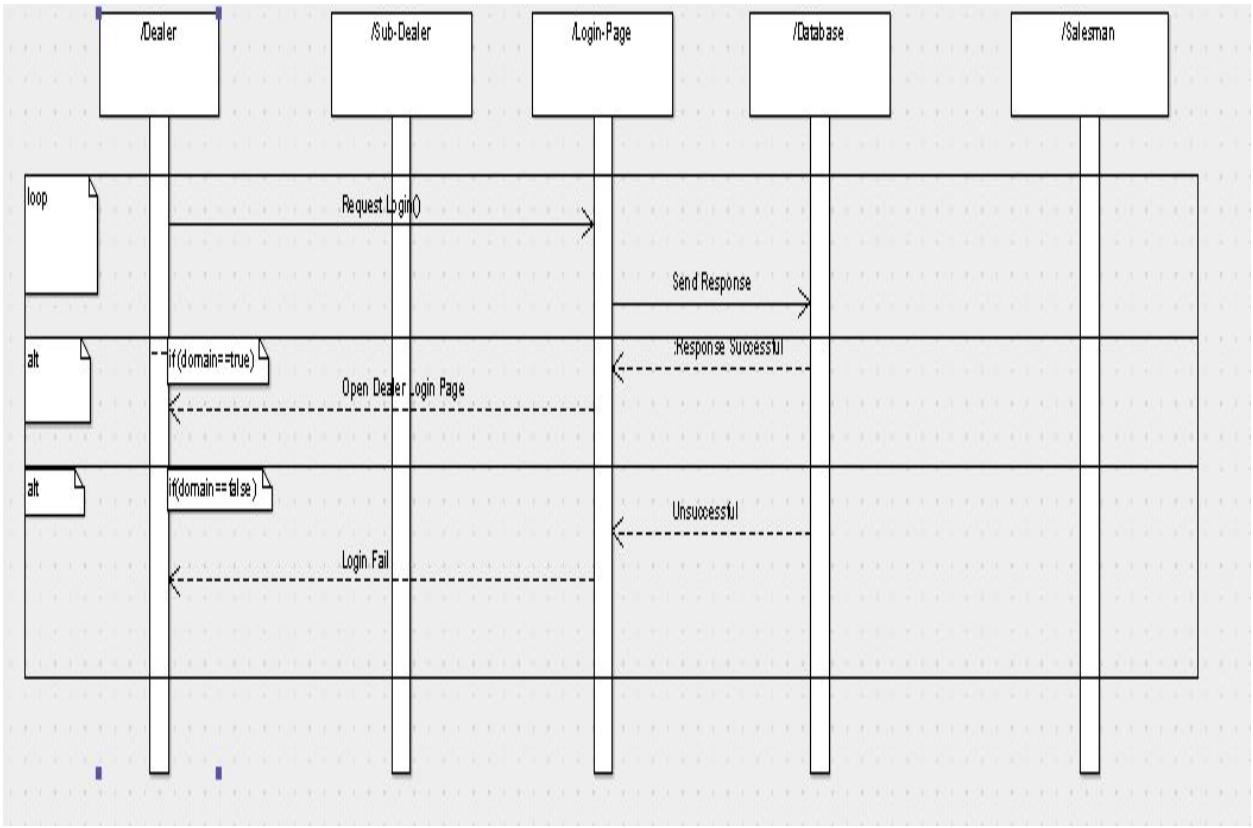


Sequence Diagram:

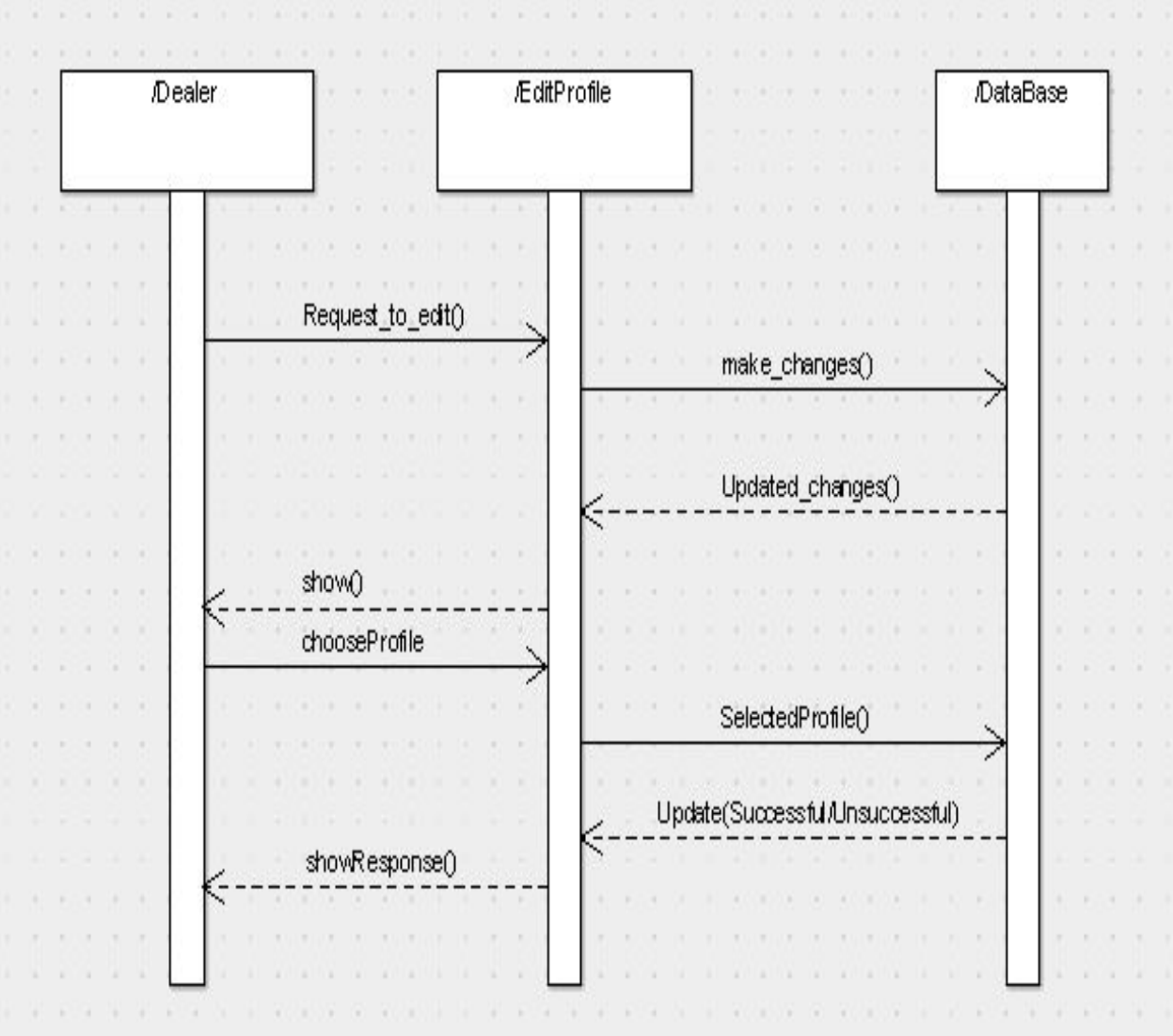
SignUp



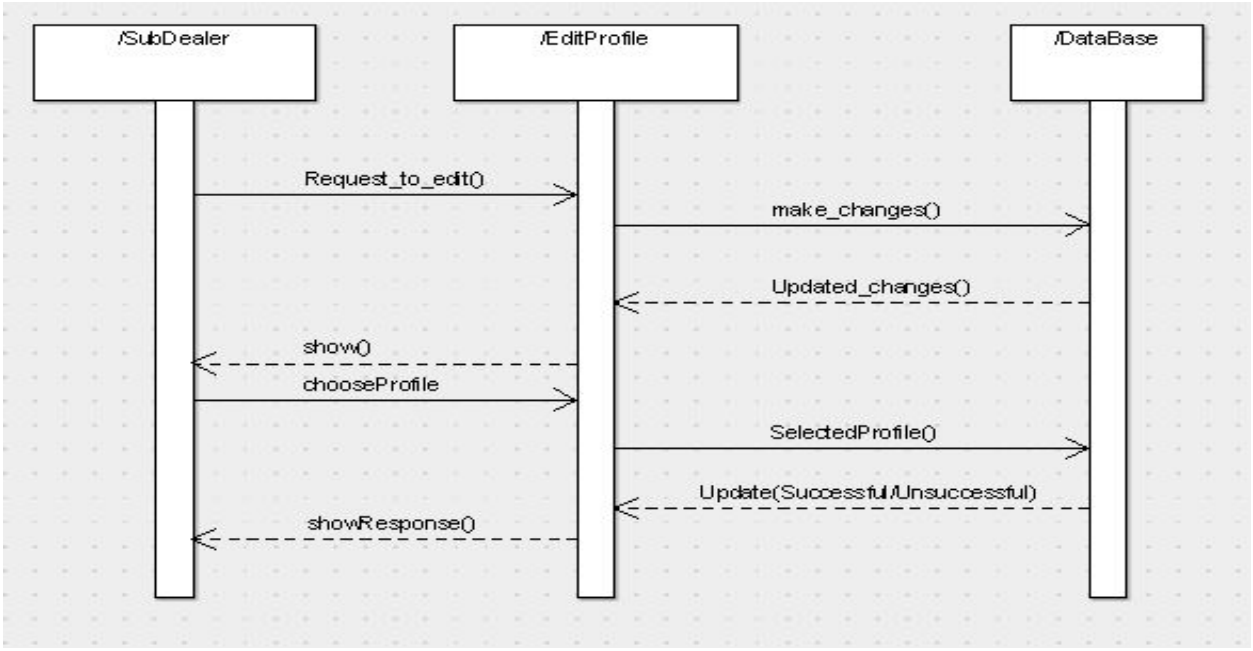
Dealer Login



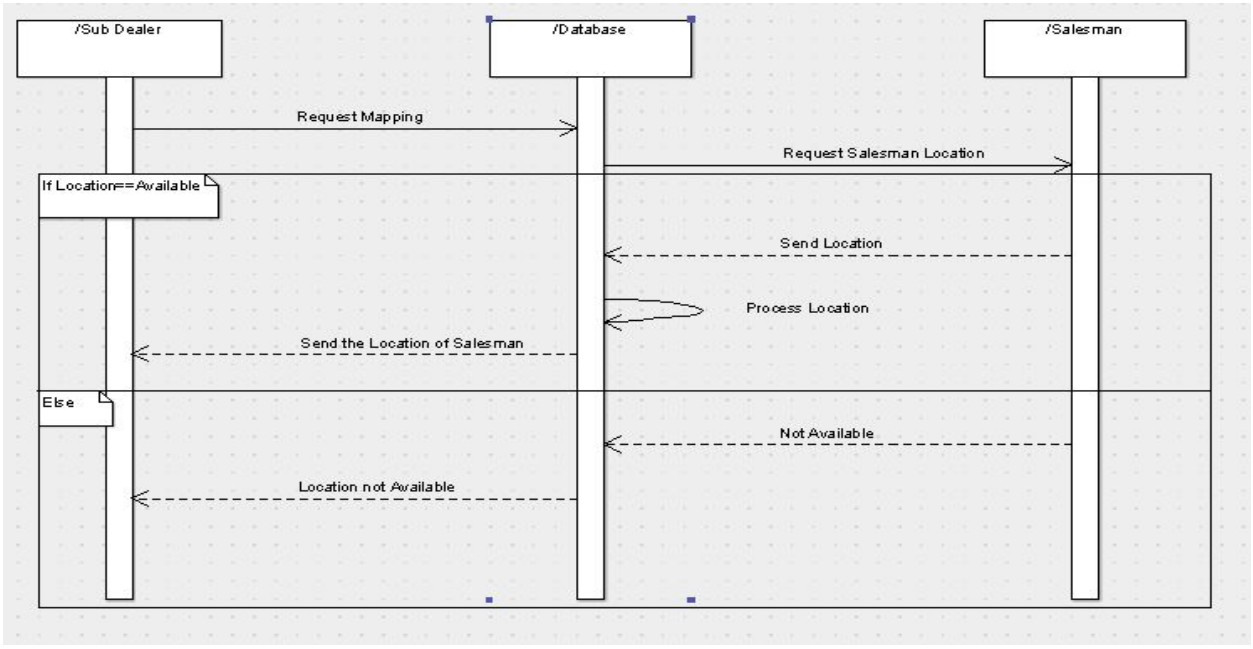
Dealer Edit Profile


/Dealer
/EditProfile
/DataBase
Request_to_edit()
make_changes()
Updated_changes()
show()
chooseProfile
SelectedProfile()
Update(Successful/Unsuccessful)
showResponse()

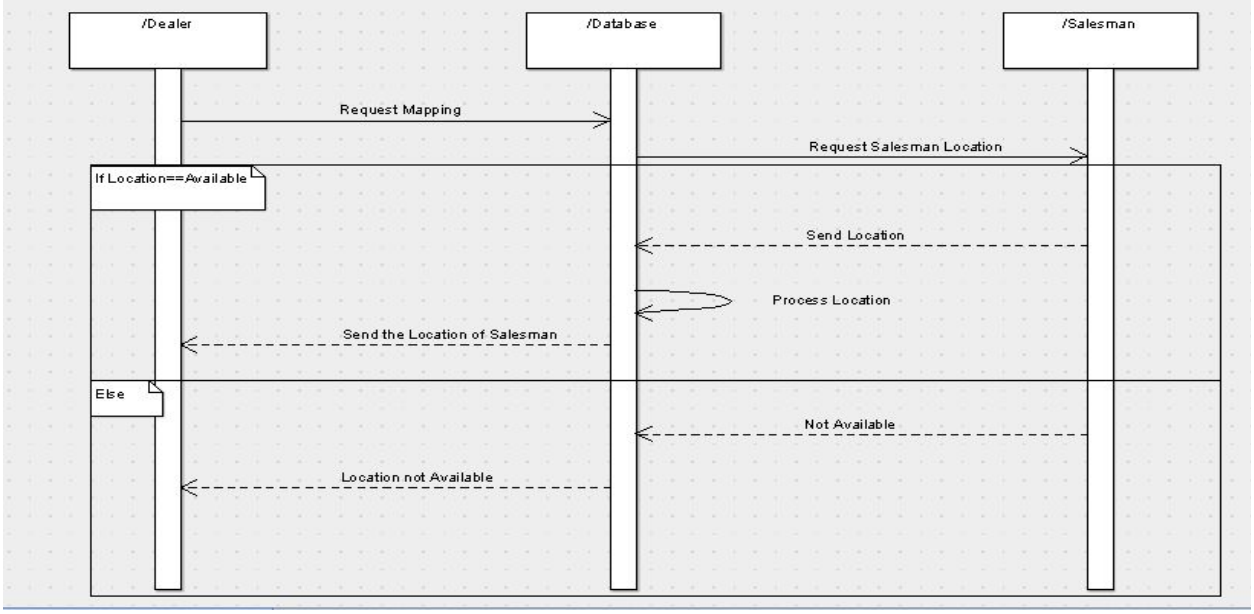
SubDealer Edit Profile



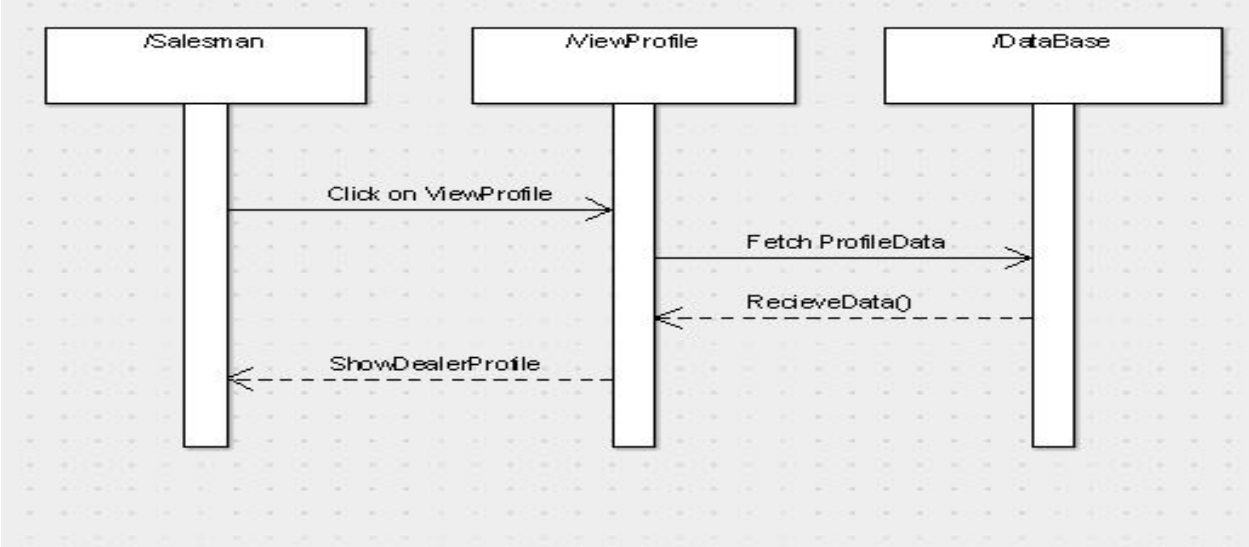
Sub Dealer Executive Mapping



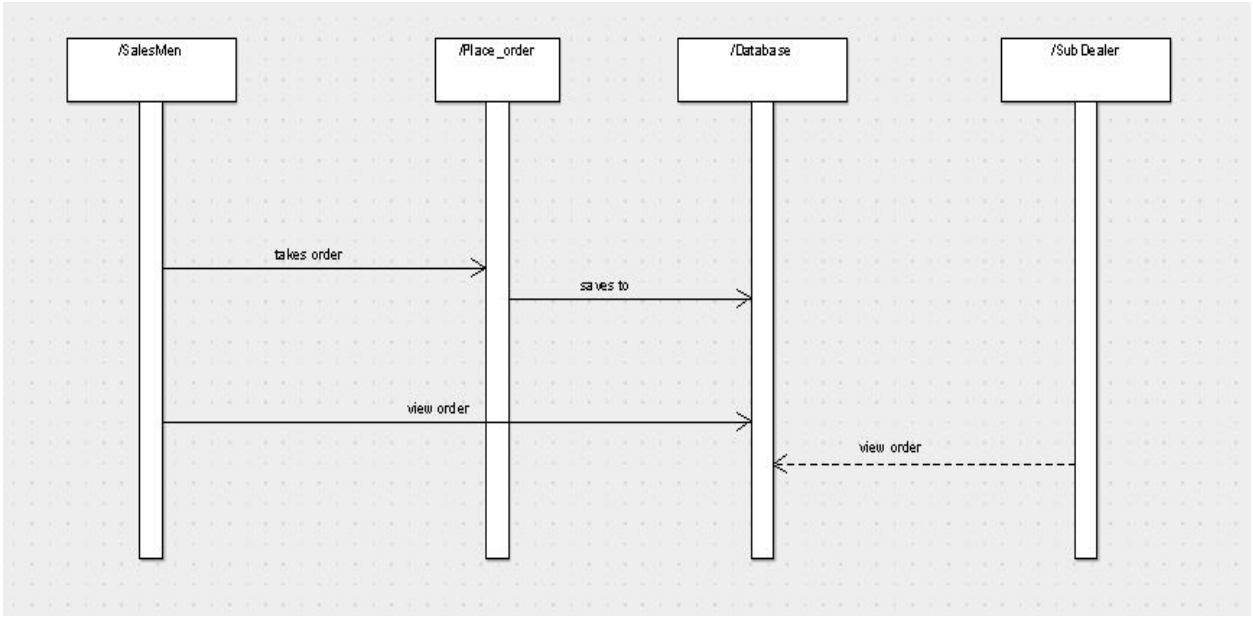
Dealer Executive Mapping



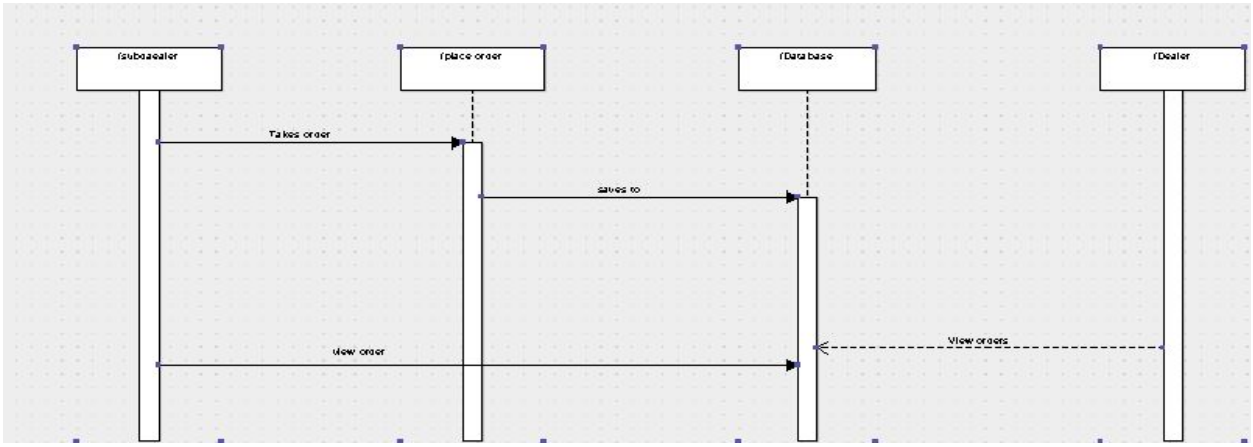
Salesman View Profile



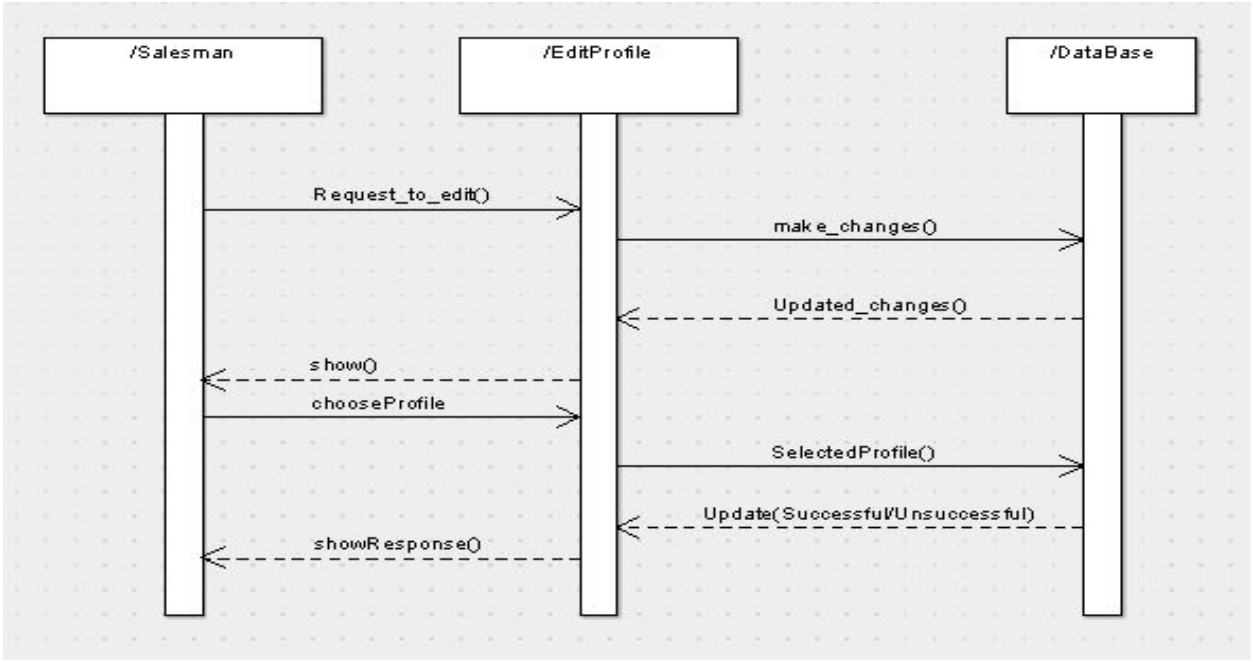
Salesman Place Order



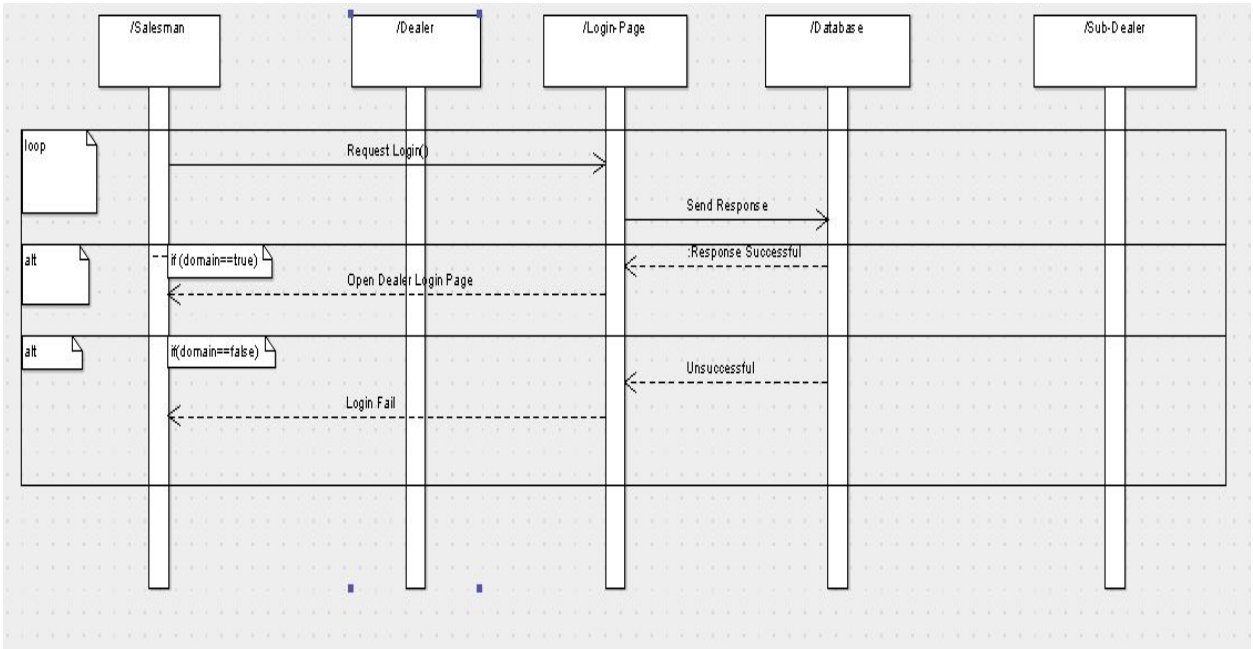
Sub Dealer Place Order



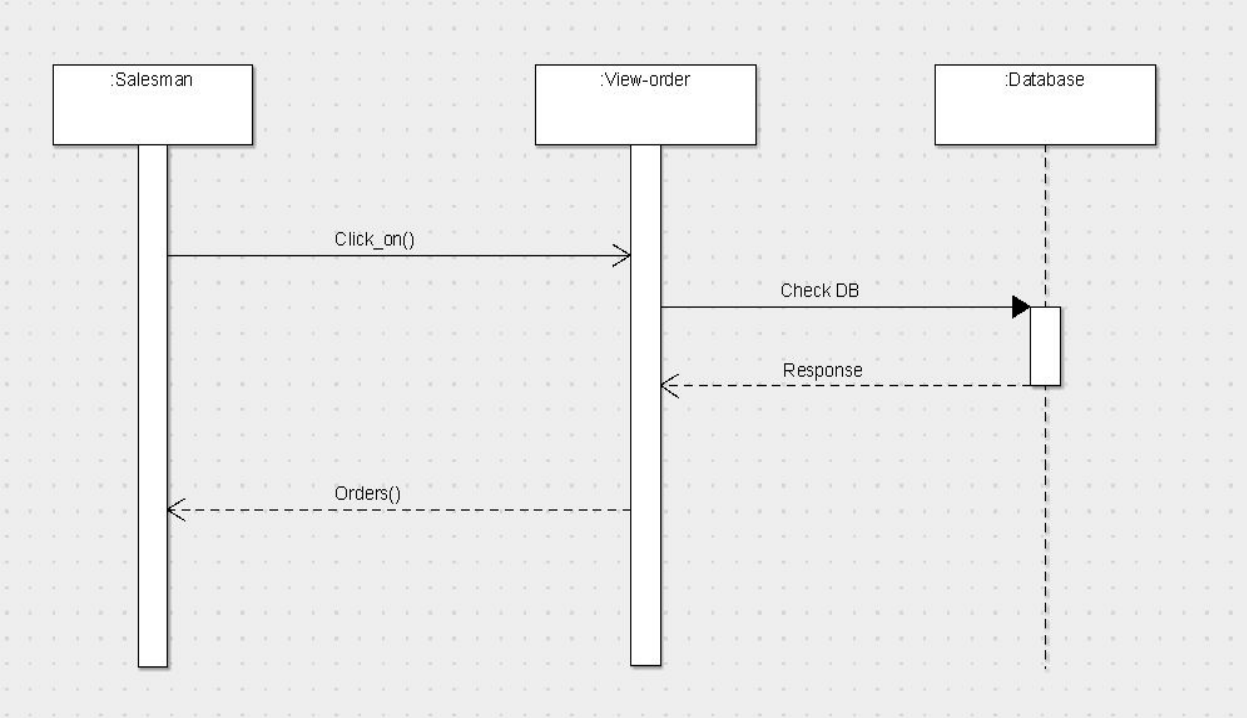
Salesman Edit Profile:



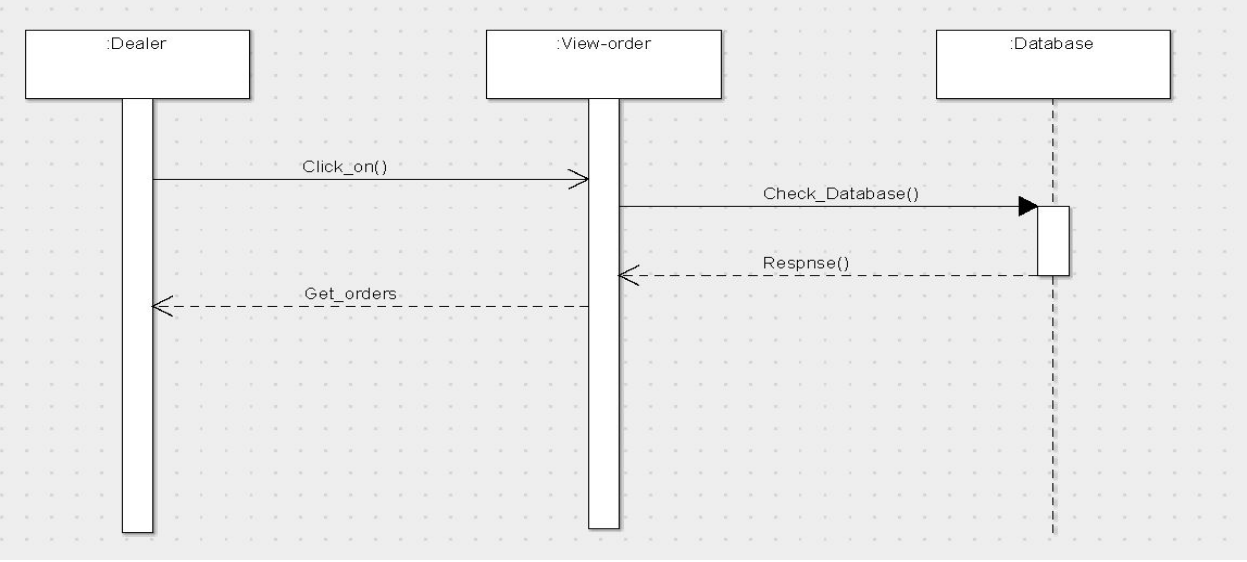
Salesman Login:



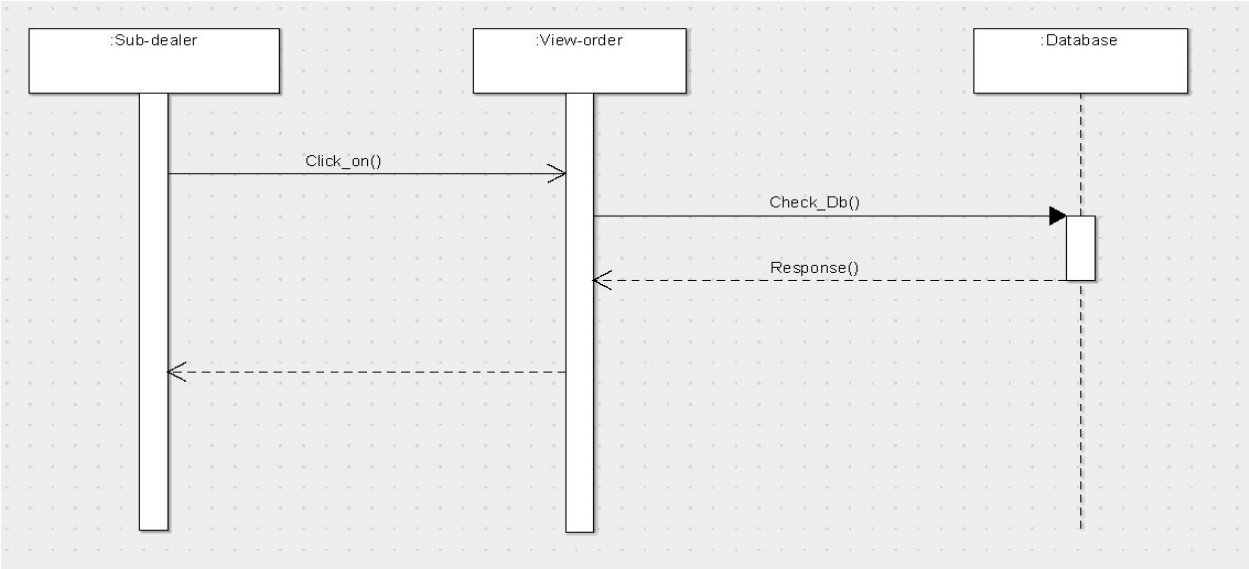
Salesman View Order



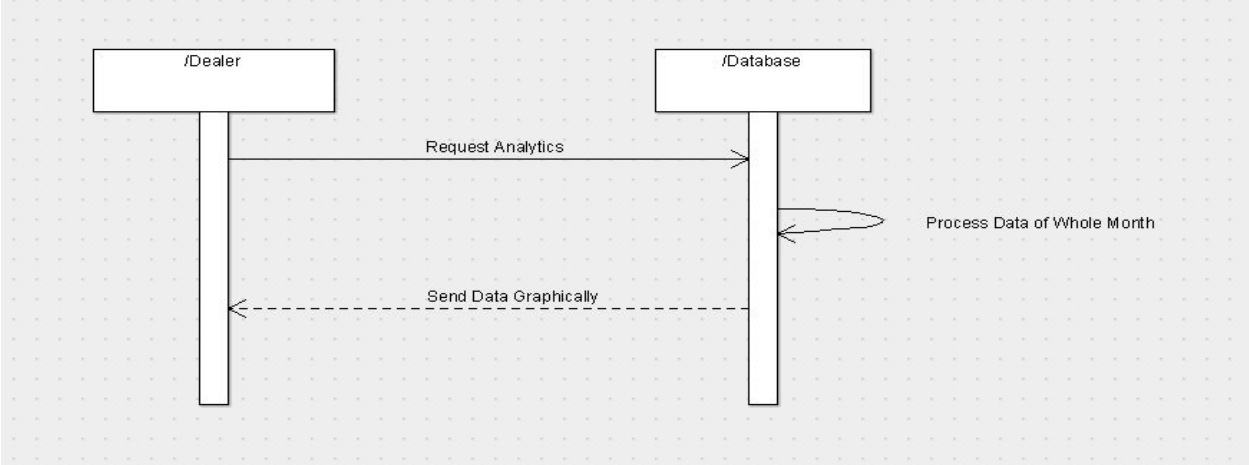
Dealer ViewOrder



Sub-Dealer ViewOrder

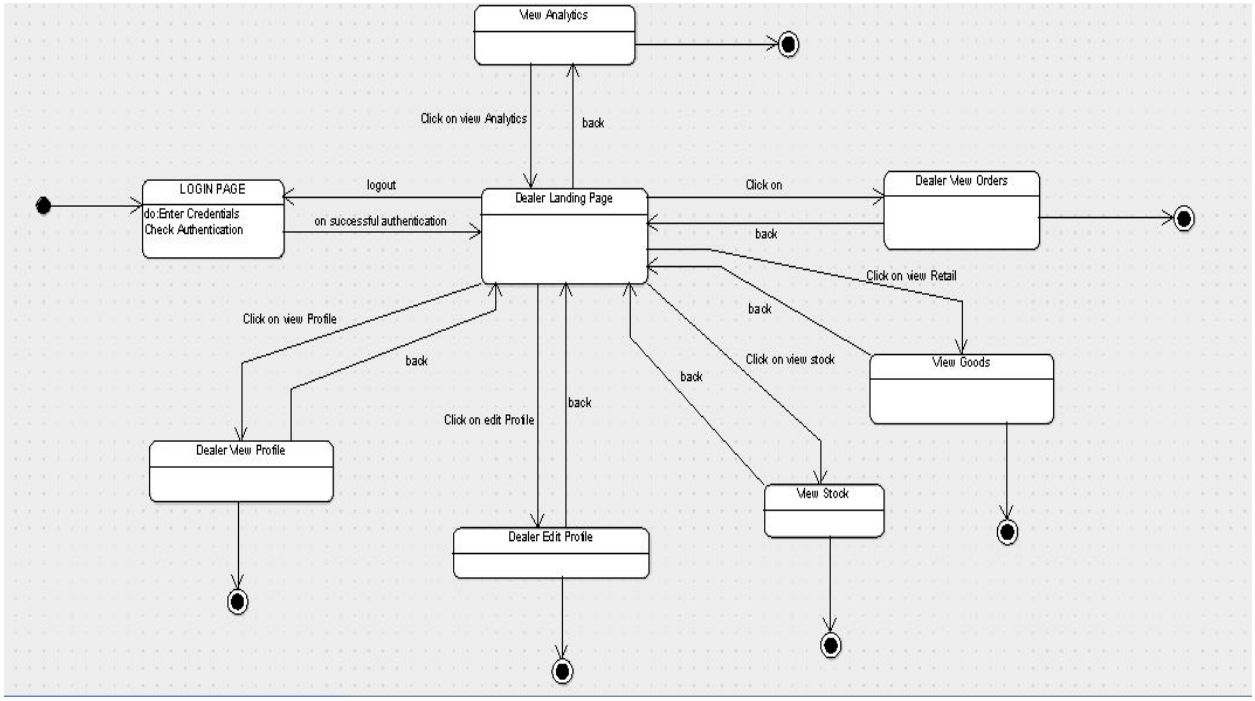


View Analytics

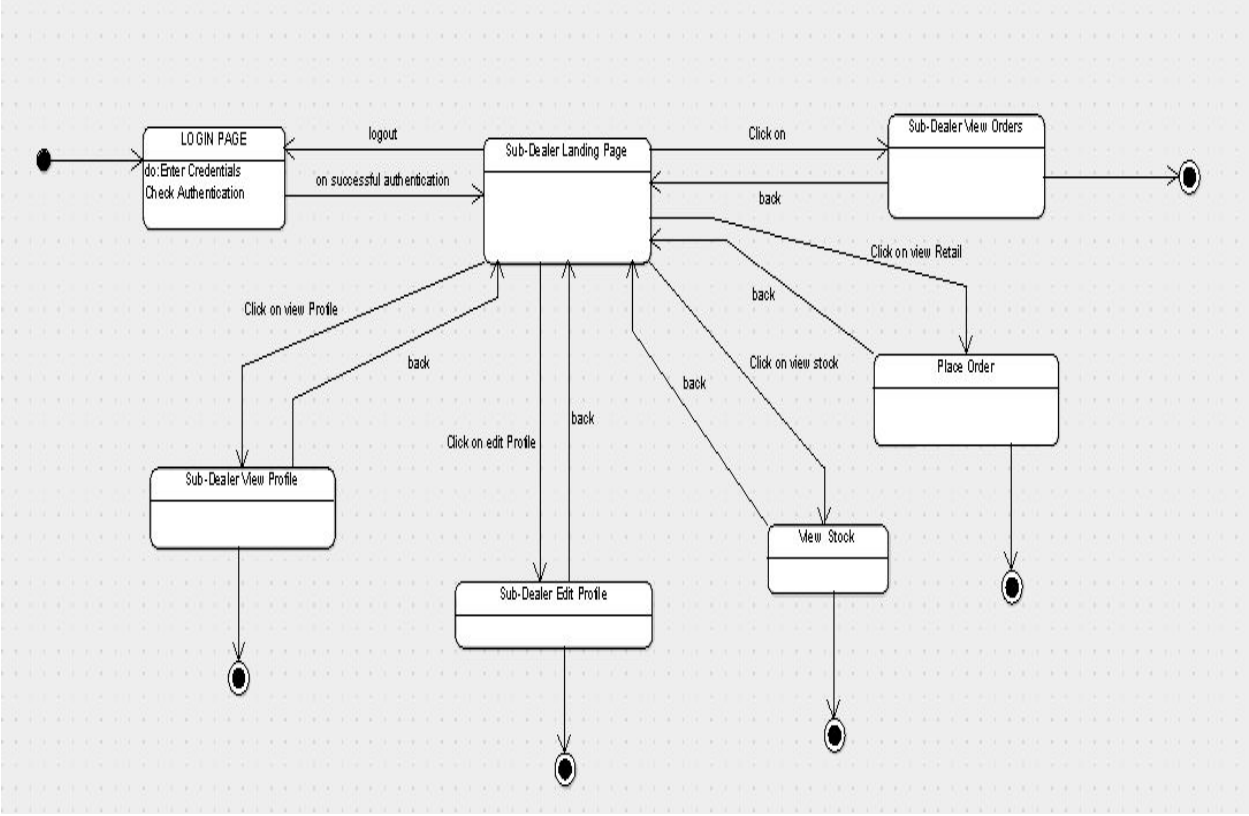


State Diagram:

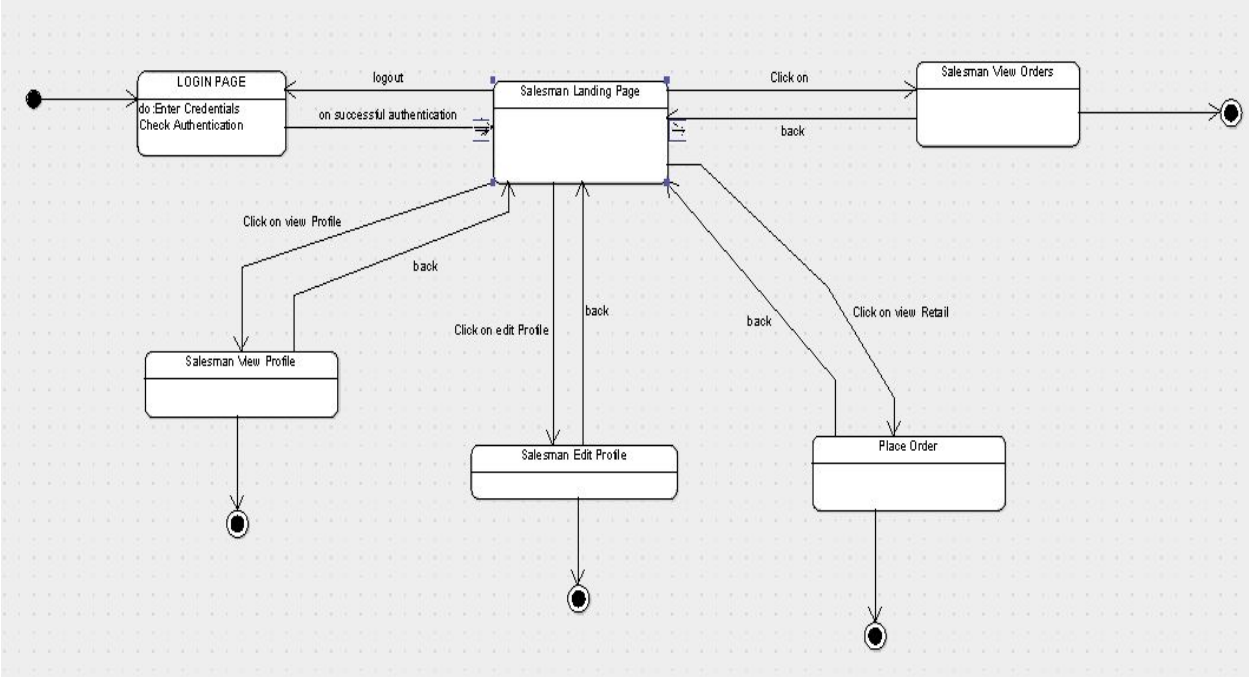
Dealer Login:



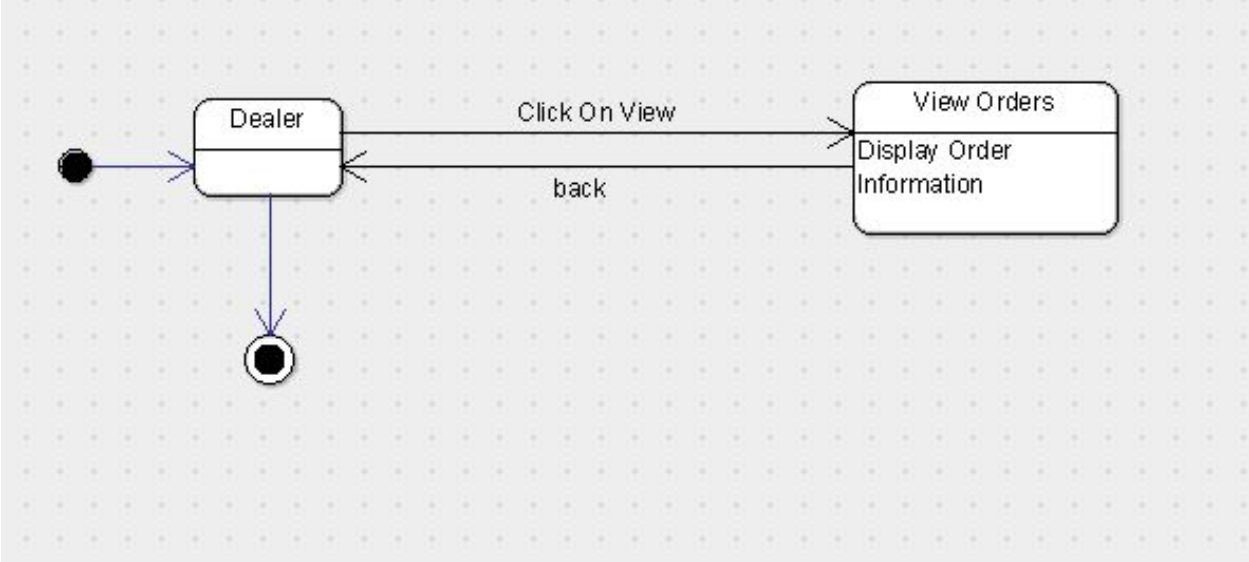
Sub-Dealer Login:



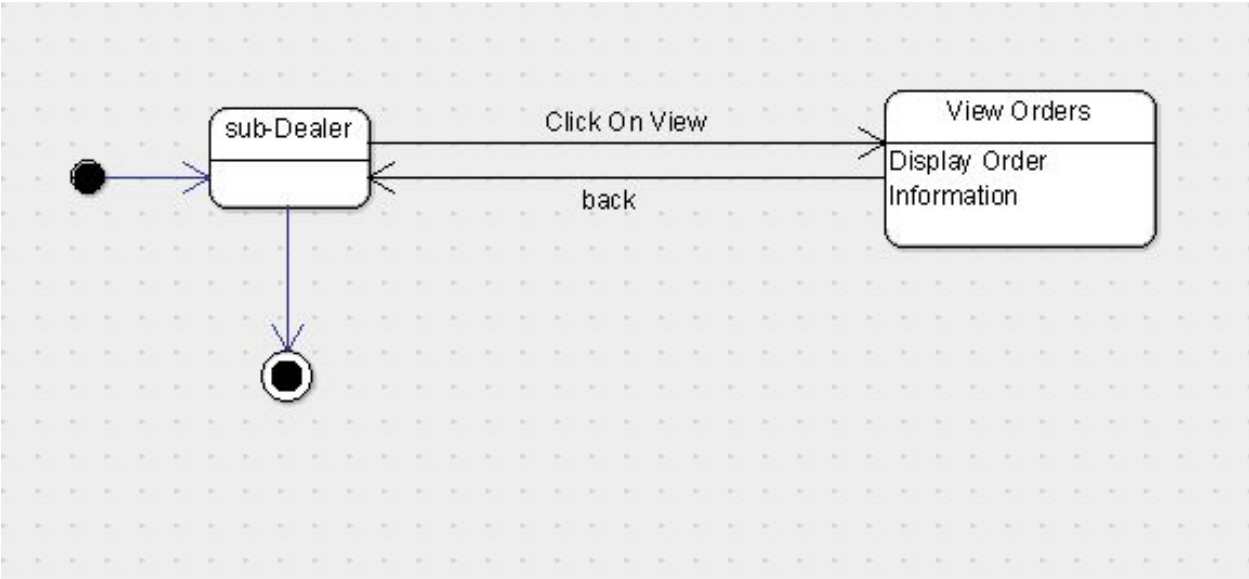
Salesman Login:



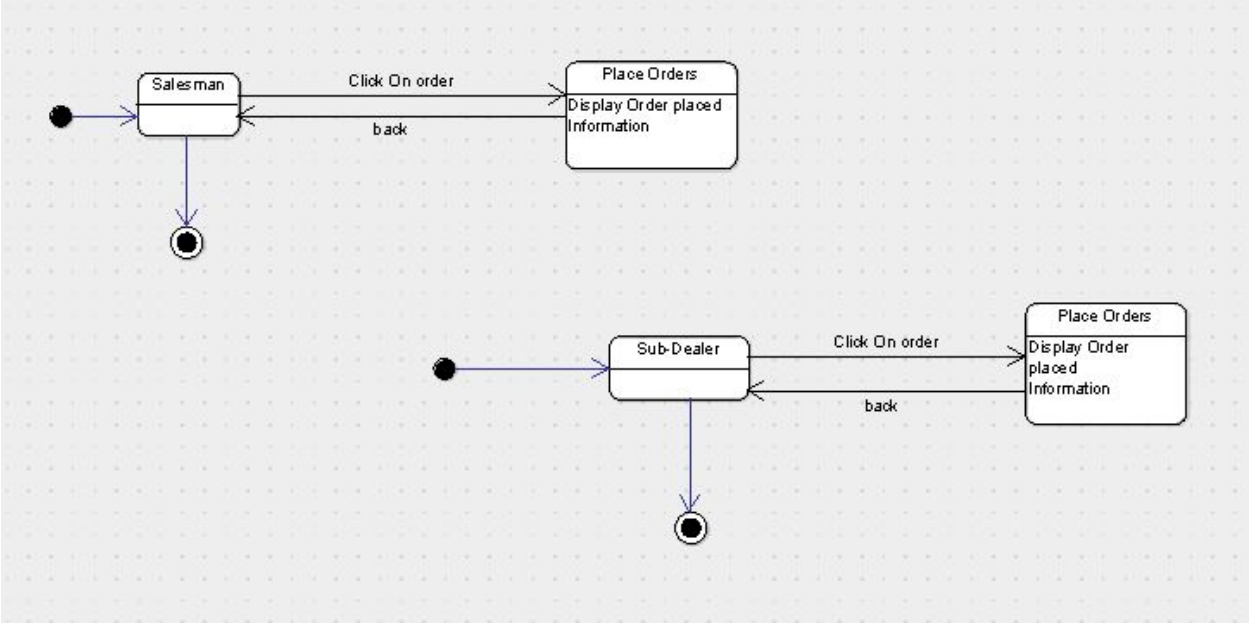
View Order(Dealer):



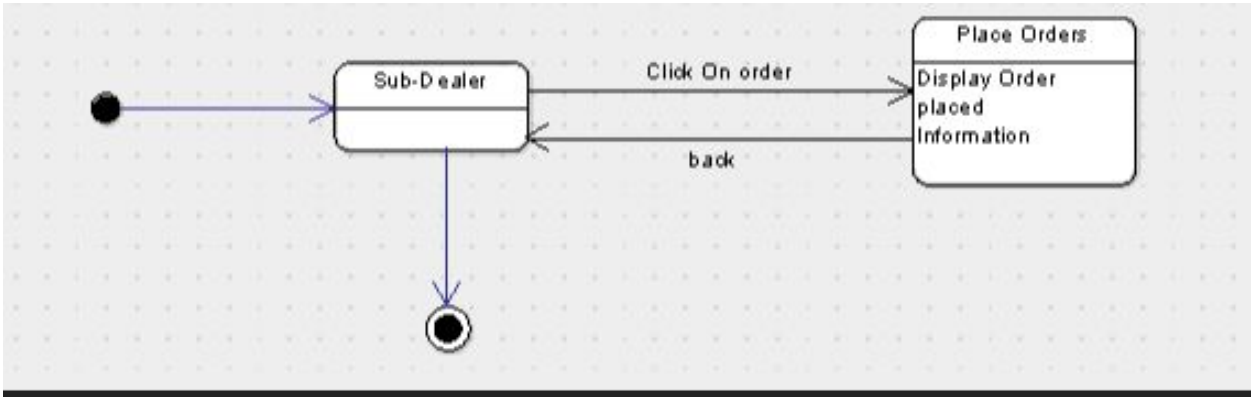
View Order(Sub-Dealer) :



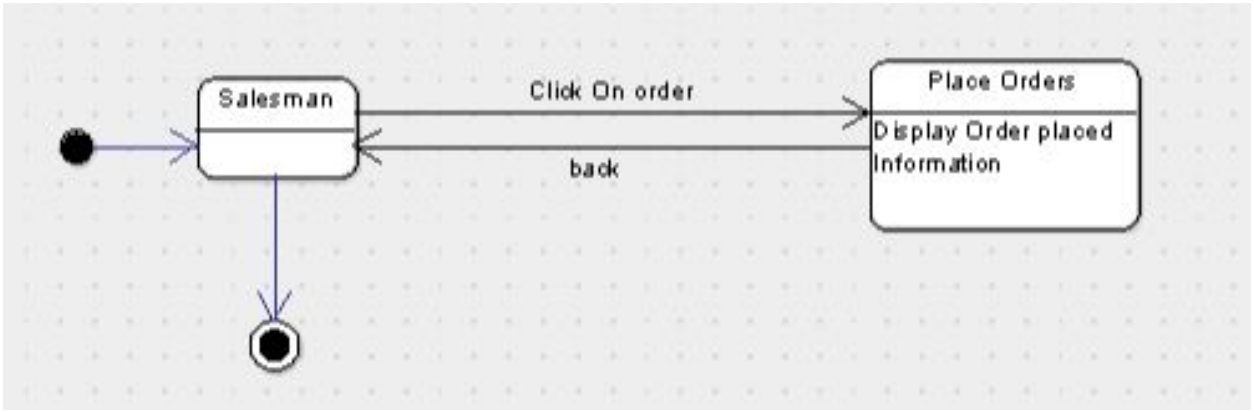
View Order(Salesman):



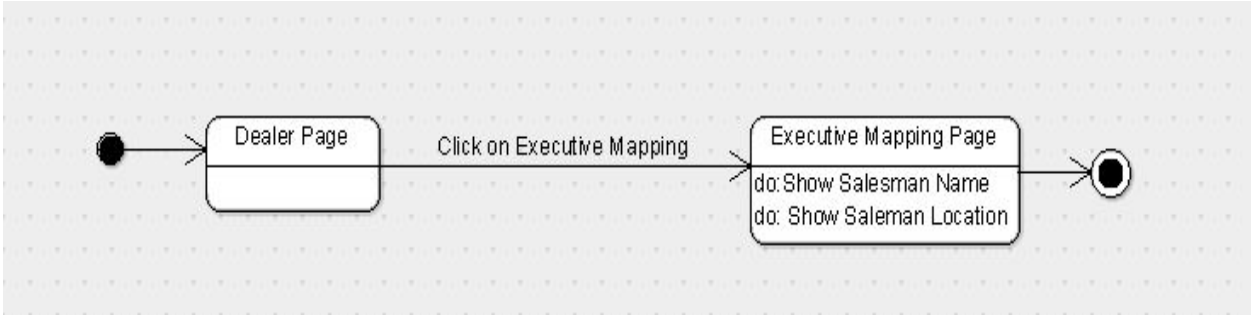
Order Place(Sub-dealer):



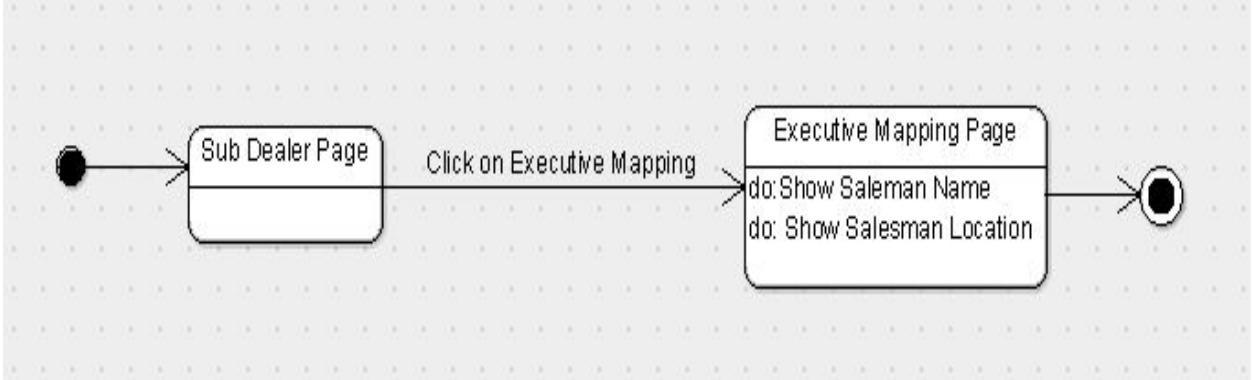
Order Place(Salesman):



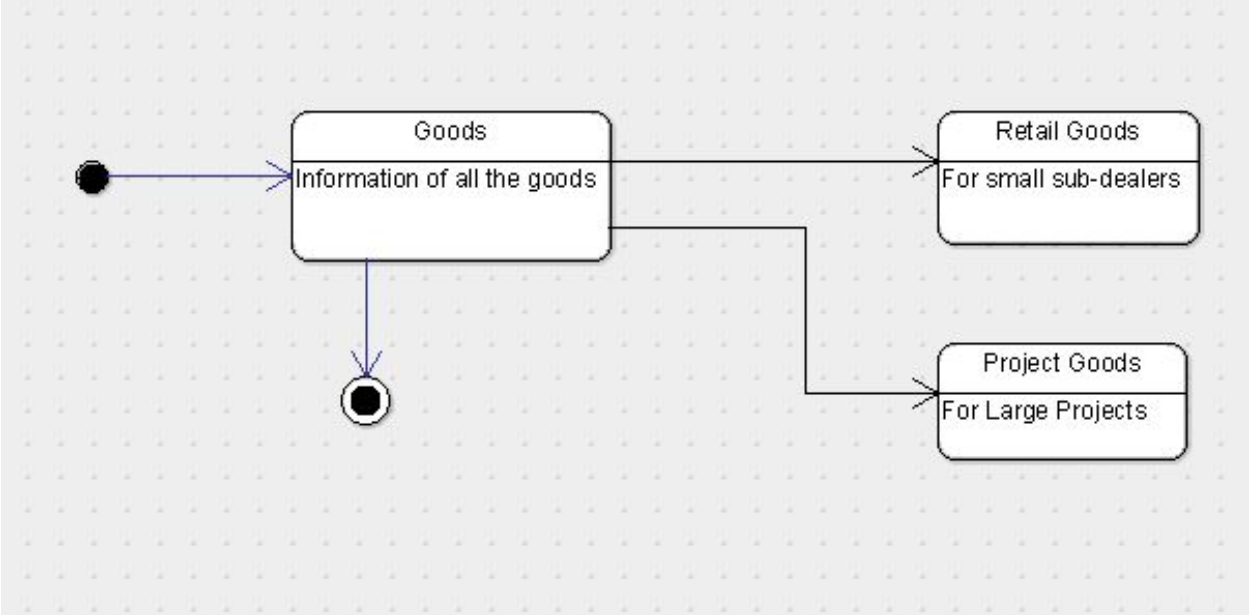
Executive Mapping(Dealer):



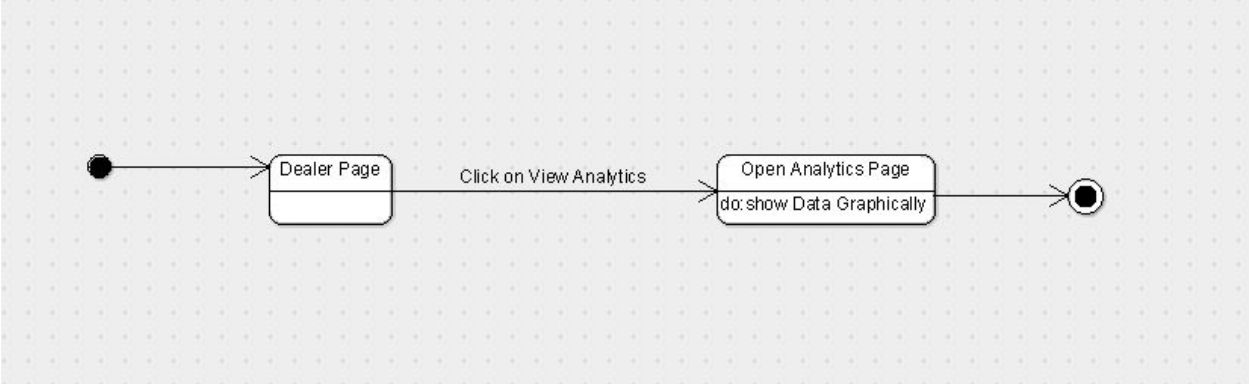
Executive Mapping(Sub-Dealer):



Goods:



Analytics:



3.1.2 Sequence Diagram:

Arrow line signifies there is a send message taken place. Response is being shown by dotted arrows.

3.1.2.1 LogIn:

It allows user to login with mail ,domain and Admin to login with the username and password that are being registered in the database already. It's loops being on same page until the correct information is not given.

3.1.2.2 Sign-up:

the user is not logged in , the user can log in as dealer, sub-dealer or salesman.

3.1.2.3 View Profile :

It allows user to view his profile which includes his details.

3.1.2.4 Edit Profile :

It allows user to edit his profile information.

3.1.2.5: Place Orders :

It allows user to place orders.

3.1.2.6: View Orders:

It allows user to view orders placed.

3.1.2.7 View Stock :

It allows user to view stocks .

3.1.2.8 Executive Mapping :

It allows sub-dealer to help to see the location of the Salesman and the time at which he went to a particular store.

3.1.2.9 : Analytics :

It allows dealer to analyze the sale of goods graphically.

3.1.2.10 Goods :

It allows dealer to see the orders of retail goods(For Small Sub-Dealers) and project goods(For Large Projects).

3.1.3 State Diagram

Initial state is being shown by starting with a black dot. Final State is being shown by the black dot surrounded by an empty circle.

3.1.3.1 Dealer LoginIn Page:

On the Login page Dealer will enter his or her details that will include phone number and password , after entering the login details when the user will click on login button and then user will be transferred on his dashboard page.

3.1.3.2 Sub-Dealer LoginIn Page:

On the Login page Sub-Dealer will enter his or her details that will include phone number and password , after entering the login details when the user will click on login button and then user will be transferred on his dashboard page.

3.1.3.3 Salesman LoginIn Page:

On the Login page Salesman will enter his or her details that will include phone number and password , after entering the login details when the user will click on login button and then user will be transferred on his dashboard page.

3.1.3.4 SignUp :

If the user is not logged in , he or she will click on the signUp button which will transfer the user to a to page where the user is supposed to fill all the necessary details i.e name, address , phone number and password, after filling the required information to sign up , the user will click on the signup button which will register the user as dealer, sub-dealer or salesman into the database.

3.1.3.5 Dealer ViewProfile:

On clicking the ViewProfile button the dealer can view his or her details which will include all the necessary details i.e name, address and phone number.

3.1.3.6 Sub-Dealer ViewProfile:
On clicking the ViewProfile button the sub-dealer can view his or her details which will include all the necessary details i.e name, address and phone number.

3.1.3.7 Salesman ViewProfile:
On clicking the ViewProfile button the salesman can view his or her details which will include all the necessary details i.e name, address and phone number.

3.1.3.8 Dealer EditProfile:
On clicking the EditProfile button the dealer can edit his or her details.

3.1.3.9 Sub-Dealer EditProfile:
On clicking the EditProfile button the sub-dealer can edit his or her details.

3.1.3.10 Salesman EditProfile:
On clicking the EditProfile button the salesman can edit his or her details.

3.1.3.11 PlaceOrders(Salesman):
On clicking the PlaceOrders button the salesman can view all the orders placed.

3.1.3.12 PlaceOrders(Dealer) :
On clicking the PlaceOrders button the dealer can view all the orders placed.

3.1.3.13 PlaceOrders(Sub- Dealer) :
On clicking the PlaceOrders button the sub-dealer can view all the orders placed.

3.1.3.14 ViewStock(Dealer):
On clicking the ViewStock button the dealer can view stock.

3.1.3.15 ViewStock(sub-Dealer):
On clicking the ViewStock button the sub-dealer can view stock.

3.1.3.16 ExecutiveMapping:
On clicking the ExecutiveMapping button the dealer can see the location of the Salesman .

3.1.3.17 RetailGoods:

On clicking the RetailGood button the dealer can see retail goods.

3.1.3.18 ProjectGoods:

On clicking the ProjectGood button the dealer can see project goods.

3.1.3.19 Analytics :

On clicking the analytics button the user can view and analyse the sales of goods geographically.

3.2 Class Name: Login

Description :
The Class allows the user to enter the System by authenticating the entered credentials.

3.2.1 Method 1:checkCrediantials()

Method Description:
When the User Enters the Credentials this method will be called. Inside this method it will check weather the username and password entered is there in the database and correct.Since it is a login activity so several state may occur like

- When user is signed out
- After Login is successful

3.2.2 Method 2:Sign Up()

Method Description:
When the User clicks the 'not a user' link then this method is invoked and takes the user to the Sign Up Page.

3.3 Class Name:Sign Up

Description : The Class allows the user to enter the Credentials in the System and Sign Up for the Application and store the user information.

3.3.1 Method 1:register_User()

Method Description:
This method stores the user information in the firebase and is invoked when Sign Up button is clicked.

3.4 Class Name: Dealer Page

Description : This Class Opens the Main Dealer Page And Enables the User the to use various options like place Orders, View Orders, View Analytics, etc

3.4.1 Method 1:ViewOrders()
Input: Click View Orders Button
Output: Open View Orders Page

Method Description:
This Method Lets the Main Dealer to View the Orders placed by the Sub Dealer, See the Sub Dealers name and the Details of the orders placed by the Sub Dealer.

3.4.2 Method 2:View Stock()
Input: Click View Stock Button
Output: Open Stock Page

Method Description:
This Method Lets the Main Dealer to View the total Stock which is left that is the total stock which is available with the dealer to distribute to the Dealer.

3.4.3 Method 3:Goods()
Input: Click Goods Button
Output: Open Goods Page Which will have 2 Types of Goods.

Method Description:
This Method Lets the Main Dealer to View the Goods which are mainly of two types, which is

- Project Goods
- Retail Goods

The Project Goods Are the Goods which are the orders Placed For the use of Big Scale(Project Purpose) like Government Projects, etc

The Retail Goods are the goods which are goods which have to transport for the Retailers or the Shops.

3.4.4 Method 4:View Analytics()

Input: Click View Analytics Button
Output: Open Analytics Layout

Method Description:
This Method is invoked when the Analytics Button is Placed. This Method will Process the Data and Show the Total Sales of the Goods Graphically.

3.4.5 Method 5:View Profile()

Method Description:
This Option is Available on the View Profile Page and lets the Dealer see his Credentials entered during the time of Sign Up.

3.4.6 Method 5:Edit Profile()

Method Description:
This Option is Available on the View Profile Page and lets the Dealer to Edit His/Her Profile accordingly and let the User Save the Edited Credentials.

3.4.7 Method 6:Executive Mapping()

Input: Click Executive Mapping Button
Output: Open Map Layout

Method Description:
This option is invoked when the mapping button is placed and lets the Dealer monitor the Salesman and get his Location of which shops the Salesman visits.

3.5 Class Name: Salesman Page

Description : This Class Opens the Salesman Page And Enables the User the to use various options like place Orders, View Orders, View Analytics, etc

3.5.1 Method 1:ViewOrders()

Input: Click View Stock Button
Output: Open View Orders Page

Method Description:
This Method lets the salesman to View the Orders placed by him and the Details of the orders.

3.5.2 Method 2:View Profile()

Method Description:
This Option is Available on the View Profile Page and lets the salesman see his Credentials entered during the time of Sign Up.

3.5.3 Method 3:Edit Profile()

Method Description:
This Option is Available on the View Profile Page and lets the salesman to Edit His/Her Profile accordingly and let the User Save the Edited Credentials.

3.5.4 Method 4:EnterLocation()

Input: Click Enter Location Button
Output: Save Salesman Information

Method Description:
This Method Will Store the Location of the Salesman in the Database and Be Used for the Future to View The Salesman's Location.

3.5.5 Method 5: placeOrders() :

Input : Click place order button
Output : Open place order page

Method Description :

This method when invoked will allow the salesman to place an order.

3.6 Class Name:Dealer Edit Profile

Description : This class opens the Edit profile page and enables the dealer to edit his profile information which includes his name,address,password and etc.

3.6.1 Method 1: editName()
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the dealer to edits his name.

3.6.2 Method 2: **editPhoneNo()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the dealer to edits his phone number.

3.6.3 Method 2: **editGender()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the dealer to edits his/her gender.

3.6.4 Method 2: **editAddress()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the dealer to updates his address.

3.6.5 Method 2: **editPassword()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the dealer to edit his password.

3.7 Class Name:Sub-Dealer Edit Profile

Description : This class opens the Edit profile page and enables the sub-dealer to edit his profile information which includes his name,address,password and etc.

3.7.1 Method 1: editName()
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the sub-dealer to edits his name.

3.7.2 Method 2: **editPhoneNo()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the sub-dealer to edits his phone number.

3.7.3 Method 2: **editGender()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the sub-dealer to edits his/her gender.

3.7.4 Method 2: **editAddress()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the sub-dealer to updates his address.

3.7.5 Method 2: **editPassword()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the sub-dealer to edit his password.

3.8 Class Name:Salesman Edit Profile

Description : This class opens the Edit profile page and enables the salesman to edit his profile information which includes his name,address,password and etc.

3.8.1 Method 1: editName()
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the salesman to edits his name.

3.8.2 Method 2: **editPhoneNo()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the salesman to edits his phone number.

3.8.3 Method 2: **editGender()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the salesman to edits his/her gender.

3.8.4 Method 2: **editAddress()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the salesman to updates his address.

3.8.5 Method 2: **editPassword()**
Input: Click Edit Profile Button.
Output : Open Edit Profile Page.

Method Description:
This method lets the salesman to edit his password.

3.9 Class Name: Sub Dealer Page

Description : This Class Opens the Sub Dealer Page And Enables the User the to use various options like place Orders, View Orders of the Salesman, View Profile,Edit Profile , etc

3.9.1 Method 1:ViewOrders()
Input: Click View Orders Button
Output: Open View Orders Page

Method Description:
This Method Lets the Sub Dealer to View the Orders placed by the Salesman, See the Salesman's name and the Details of the orders placed by the Salesman.

3.9.2 Method 2:View Stock()
Input: Click View Stock Button
Output: Open Stock Page
Method Description:

This Method Lets the Sub Dealer to View the total Stock which is left that is the total stock which is available with the SubDealer and when to place an order to the Dealer.

3.9.3 Method 5:Edit Profile()

Method Description:
This Option is Available on the View Profile Page and lets the Sub Dealer to Edit His/Her Profile accordingly and let the User Save the Edited Credentials.

3.9.4 Method 5:View Profile()

Method Description:
This Option is Available on the View Profile Page and lets the Sub Dealer see his Credentials entered during the time of Sign Up.

3.10 Class Name:Dealer View Profile

Description :
This class opens the View profile page and enables the dealer to view his profile information which includes his name,address,password and etc.

3.10.1 Method 1: viewName()
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the dealer to view his name.

3.10.2 Method 2: viewPhoneNo()
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the dealer to view his phone number.

3.10.3 Method 2: **editGender()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the dealer to view his/her gender.

3.10.4 Method 2: **ViewAddress()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the dealer to view his address.

3.10.5 Method 2:**viewPassword()**
Input: Click view Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the dealer to view his password.

3.11 Class Name:Sub-Dealer View Profile

Description : This class opens the View profile page and enables the Subdealer to view his profile information which includes his name,address,password and etc.

3.10.1 Method 1: viewName()
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Subdealer to view his name.

3.10.2 Method 2: **viewPhoneNo()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Subdealer to view his phone number.

3.10.3 Method 2: **editGender()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Subdealer to view his/her gender.

3.10.4 Method 2: **ViewAddress()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Subdealer to view his address.

3.10.5 Method 2:**viewPassword()**
Input: Click view Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Subdealer to view his password.

3.12 Class Name:Salesman View Profile

Description :
This class opens the View profile page and enables the Salesman to view his profile information which includes his name,address,password and etc.

3.10.1 Method 1: viewName()
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Salesman to view his name.

3.10.2 Method 2: **viewPhoneNo()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Salesman to view his phone number.

3.10.3 Method 2: **editGender()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Salesman to view his/her gender.

3.10.4 Method 2: **ViewAddress()**
Input: Click View Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Salesman to view his address.

3.10.5 Method 2:**viewPassword()**
Input: Click view Profile Button.
Output : Open View Profile Page.

Method Description:
This method lets the Salesman to view his password.

3.13 Class Name:View Orders

Description:

This Class will allow the Dealer, Sub Dealer and The Salesman to view his/her Orders and the Quantity of Goods Places .

3.13.1 Method 1:show_order_list()

Input: User Clicks View Order Button
Output: Open A List of Orders

Method Description
This Method when invoked will allow the Dealer, Sub Dealer and The Salesman to view his/her Orders and the Quantity in the Form of List and will show the details of the item /Order Placed

3.14 Class Name: Place Orders

Description:
This class opens the Place order page and enables the sub-dealer to place an order to dealer and salesman to place an order to sub-dealer.

3.14.1 Method 1: placeOrders() :

Input : Click place order button
Output : Open place order page
Method Description :
This method when invoked will allow the sub-dealer and dealer to place an order.

3.15 Class Name: Executive Mapping()

Description:
This Class Enables the Respective User to See the Location of the Sub Dealer and Monitor Him/her .

3.15.1 Method 1:SalesmanNameinfo()

Description:
This method will retrieve the Salesman Information that is the name and return the name of the Salesman and display it to the Requested User.

3.15.2 Method 2:Salesman_Loction()

Description:

This method will retrieve the Salesman Location and and Display it to the Dealer or the Sub Dealer who has requested for it.

3.17 Class Name : Goods()

Description :

This Class will Enable the User to View Goods To the User Which are of Two Types.

3.17.1 Method 1: Retail Info()

This Method will collect the data of Retail Goods and Display it to the Dealer.

3.17.1 Method 2: Project Info()

This Method will collect the information of the Project goods such as goods used for big government projects etc and display it to the Dealer.

3.18 Class Name : viewStocks()

Description:

This Class will Help to View the total Stock Available with the Dealer and subDealer.

3.18.1 Method Name:totalStockAvailable()

Input: click totalStockAvailable page

Output: Open total stock available page

Method Description:

This method will view the total stock available with the Dealer and SubDealer

3.19 Class Name : viewAnalytics()

Description :

This class enables the dealer to show the sales of goods graphically.

3.19.1 Method 1: analyticDataInfo()

Input : Click analyticDataInfo page.

Output : Open graph Analytic page

Method Description:

This method will analyze the sale of goods and according to the data it will shows the sale of goods in different regions graphically.

5.0 Design decisions and tradeoffs

The design decision to use three screens separately for Dealer,Sub Dealer and Salesman for better user interface. It Will help in providing privacy to the Dealer and the Sub Dealer.

A possible tradeoff when considering links is to use buttons instead of items in the menu. This design decision - to use buttons for navigating between screens - is to enhance visibility. Therefore, it is easier for the user to Navigate Through the Map Pages.

6.0 Pseudocode for components

6.0.1 Class Name: Login(Salesman/Dealer/Sub-Dealer)

```
1. Begin
2. valid_input_password := false
3. valid_input_mobileno := false
4. logged_in := false

5. while (logged_in = false) do
6. begin
7. print("Please enter Mobile Number: ") //input Username
8. readln(inputted_mobileno)
9. writeln("Please enter Password: ") //input Password
10. readln(inputted_password)
11. //begin matching process
12. if inputted_mobileno is in Login Database then
13. //match inputted login details with each existing valid set of Login Details from the Login
    Database
14. valid_input_mobileno:= true
15. if inputted_password is in Login_Database then
16. valid_input_mobileno := true;
17. if inputted_password is in Login_Database then
```

18. valid input_password := true //Mobile no. and Password are both matched
19. logged_in := true
20. writeln("Logins details have been validated)
21. else
22. valid input_password := false
23. mobileno was matched but password wasn't
24. logged_in := false
25. writeln("Error your username or password was invalid")
26. end if
27. else
28. valid_input_mobileno := false
29. mobileno wasn't matched
30. logged_in := false
31. writeln("Error your mobileno or password was invalid")
32. end if
33. end while
34. if logged_in = true then
35. writeln("'ou have been logged in")
36. else
37. writeln("'you could not be logged on due to false Login details")
38. end if
39. end

6.0.2 Class Name: Sign in

SignIn()

Input: SignIn through database
Output: user landing page of login successful
1. Intent signInIntent = get signIn Intent from database for the given client
2. if signInIntent==database client
3. then
4. start another activity leading to user landing page
5. else
6. print message login unsuccessful
7. end if

6.0.3 Class Name: DealerViewProfile

Input : ViewProfile
Output : Display Dealer Profile

1.Intent profileIntent = get profile intent from database for the dealer.
2.if profileIntent == database client
3. then
4. shows the dealer details
5. else
6. print message error occured or data altered
7. end if

6.0.4 Class Name:SubDealerViewProfile

Input : ViewProfile
Output : Display Dealer Profile
1.Intent profileIntent = get profile intent from database for the respective sub-dealer.
2.if profileIntent == database client
3. then
4. shows the sub-dealer details
5. else
6. print message error occured or data altered
7. end if

6.0.5 Class Name:SalesmanViewProfile
Input : ViewProfile
Output : Display Salesman Profile
1.Intent profileIntent = get profile intent from database for the respective salesman.
2.if profileIntent == database client
3. then
4. shows the salesman details
5. else
6. print message error occured or data altered
7. end if

6.0.6 Class Name: DealerEditProfile
Input : EditProfile
Output : update his/her Profile
1.Intent editProfileIntent = dealer update or changes his basic information in the database
2.if editProfileIntent == update information
3.then
4.updates the dealer information in the database
5.else
6.no changes made
7.end if

6.0.7 Class Name:SubDealerEditProfile

Input : EditProfile
Output : update his/her Profile
1.Intent editProfileIntent = subdealer update or changes his basic information in the database
2.if editProfileIntent == update information
3.then
4.updates the subdealer information in the database
5.else
6.no changes made
7.end if

6.0.8 Class Name:SalesmanEditProfile

Input : EditProfile
Output : update his/her Profile
1.Intent editProfileIntent = salesman update or changes his basic information in the database
2.if editProfileIntent == update information
3.then
4.updates the salesman information in the database
5.else
6.no changes made
7.end if

6.0.9 Class Name:ViewOrderSub-dealer

Input : view order list
Output : Display order list
1.Intent profileIntent = get order intent from database for the respective sub-dealer.
2.if profileIntent == database client
3. then
4. shows the subdealer the order details
5. else
6. print message error occured or data altered
7. end if

6.0.10 Class Name:ViewOrderDealer

Input : view order list
Output : Display order list

1.Intent profileIntent = get order intent from database for the respective dealer.
2.if profileIntent == database client
3. then
4. shows the dealer the order details
5. else
6. print message error occured or data altered.
7. end if

6.0.11 Class Name:Executive-mapping

Input:view the location
Output:Location displayed

1. Intent profileIntent=get location for Dealer and Subdealer
2. If profileIntent===database client
3. Then
4. Shows dealer the location
5. Else
6. Print message can’t fetch details
7. End if

6.0.12 Class Name: Dealer(view stock)

Input : ViewStock
Output : Display stock to Dealer
1.Intent stockIntent = get stock intent from database for the dealer.
2.if stockIntent == database client
3. then
4. shows the stock details
5. else
6. print message error occured or data altered
7. end if

6.0.13 Class Name:Sub-Dealer(view stock)

Input : ViewStock
Output : Display stock to Sub-Dealer
1.Intent stockIntent = get stock intent from database for the sub-dealer.
2.if stockIntent == database client
3. then
4. shows the stock details
5. else
6. print message error occured or data altered

7. end if

6.0.14 Class Name:placeOrder(salesman)

Input : place order
Output : order placed successfully
1.Intent orderplaceIntent = salesman places order
2.if stockIntent == order placed data successfully saved in database
3.then
4.updates the order list information in the database
5.else
6.no changes made
7.end if

6.0.15 Class Name:placeOrder(sub-dealer)

Input : place order
Output : order placed successfully
1.Intent orderplaceIntent = sub-dealer places order
2.if stockIntent == order placed data successfully saved in database
3.then
4.updates the order list information in the database
5.else
6.no changes made
7.end if

6.0.16 Class Name:Analytics

Input : View analytical graphs
Output : graph displayed
1.Intent analyticalIntent = get graph intent from database for the user.
2.if analyticalIntent == database client
3. then
4. shows the graphical details
5. else
6. print message error occured or data altered
7. end if

CHAPTER 3
SOFTWARE CODING(METRICS)

Lines of Code

Lines of code metrics	Sat	24 Nov 2018 00:04:27 IST									
Package	LOC	LOC(rec)	LOCp	LOCp(rec)	LOCt	LOCt(rec)					
	n/a	1,615	n/a	1,580	n/a	35					
com	n/a	1,615	n/a	1,580	n/a	35					
com.example	n/a	1,615	n/a	1,580	n/a	35					
com.example.siddhant	n/a	1,615	n/a	1,580	n/a	35					
com.example.siddhant.electrobasket	1,615	1,615	1,580	1,580	35	35					
Module	L(Groovy)	L(HTML)	L(J)	L(KT)	L(XML)	LOC	LOCp	LOCt	NCLOC	NCLOCp	NCLOCt
ElectroBasket	23	0	0	0	0	274	0	0	29	0	0
app	33	0	1,615	0	2,849	4,571	4,384	35	4,500	4,325	24
FileType	LOC	NCLOC									
Groovy	56	55									
JSON	55	55									
Java	1,615	1,545									

ProGuard Rules Language File	18	18									
Properties	29	7									
Text	223	n/a									
XML	2,849	2,849									
Project	L(Groovy)	L(HTML)	L(J)	L(KT)	L(XML)	LOC	LOCp	LOCt	NCLOC	NCLOCp	NCLOCt
project	56	0	1,615	0	2,849	4,845	4,384	35	4,529	4,325	24

Complexity Matrix

Complexity metrics	Sat	24 Nov 2018 00:09:38 IST	
Method	ev(G)	iv(G)	v(G)
Dealer.Mapping(View)	1	1	1
Dealer.Project(View)	1	1	1
Dealer.ViewAnalytics(View)	1	1	1
Dealer.ViewOrders(View)	1	1	1
Dealer.ViewProfile(View)	1	1	1
Dealer.ViewStock(View)	1	1	1
Dealer.onBackPressed()	2	2	2
Dealer.onCreate(Bundle)	1	3	3
DealerAnalytics.onCreate(Bundle)	1	1	1
DealerExecutive.onCreate(Bundle)	1	4	4
DealerProjectGoods.onCreate(Bundle)	1	4	4
DealerViewOrders.onCreate(Bundle)	1	3	3
DealerViewProfile.disable()	1	1	1
DealerViewProfile.enable()	1	1	1
DealerViewProfile.onCreate(Bundle)	1	3	3
DealerViewStock.onCreate(Bundle)	1	1	1

ExampleInstrumentedTest.useAppContext()	1	1	1
ExampleUnitTest.addition_isCorrect()	1	1	1
Login.login(String,String)	1	7	7
Login.onCreate(Bundle)	1	3	3
Salesman.Location(View)	1	1	1
Salesman.SalesmanPlaceOrder(View)	1	1	1
Salesman.SalesmanViewProfile(View)	1	1	1
Salesman.ViewPlacedOrders(View)	1	1	1
Salesman.onBackPressed()	2	2	2
Salesman.onCreate(Bundle)	1	7	7
Salesman.requestPermission()	1	1	1
SalesmanPlaceOrders.onCreate(Bundle)	1	3	3
SalesmanPlaceOrders.total()	1	1	1
SalesmanViewPlacedOrders.onCreate(Bundle)	1	3	3
SalesmanViewProfile.disable()	1	1	1
SalesmanViewProfile.enable()	1	1	1
SalesmanViewProfile.onCreate(Bundle)	1	3	3
SignUp.check(EditText[])	3	3	3
SignUp.onCreate(Bundle)	1	5	5
SubDealer.PlaceOrder(View)	1	1	1
SubDealer.Retail(View)	1	1	1
SubDealer.SubMapping(View)	1	1	1
SubDealer.SubViewOrder(View)	1	1	1
SubDealer.SubViewStock(View)	1	1	1
SubDealer.ViewPro(View)	1	1	1
SubDealer.onBackPressed()	2	2	2
SubDealer.onCreate(Bundle)	1	3	3
SubDealerMapping.onCreate(Bundle)	1	4	4

SubDealerPlaceOrder.onCreate(Bundle)	1	3	3
SubDealerPlaceOrder.total()	1	1	1
SubDealerRetailGoods.onCreate(Bundle)	1	4	4
SubDealerViewOrder.onCreate(Bundle)	1	3	3
SubDealerViewProfile.disable()	1	1	1
SubDealerViewProfile.enable()	1	1	1
SubDealerViewProfile.onCreate(Bundle)	1	3	3
SubDealerViewStock.onCreate(Bundle)	1	1	1
Class	OCavg	WMC	
Dealer	1.33	12	
DealerAnalytics	1	1	
DealerExecutive	2	6	
DealerProjectGoods	1.75	7	
DealerViewOrders	1.67	5	
DealerViewProfile	1.29	9	
DealerViewStock	1	1	
ExampleInstrumentedTest	1	1	
ExampleUnitTest	1	1	
Login	2.33	14	
Salesman	1.64	18	
SalesmanPlaceOrders	1.12	19	
SalesmanViewPlacedOrders	1.67	5	
SalesmanViewProfile	1.29	9	
SignUp	3	9	
SubDealer	1.33	12	
SubDealerMapping	2	6	
SubDealerPlaceOrder	1.12	19	
SubDealerRetailGoods	1.75	7	
SubDealerViewOrder	1.67	5	

SubDealerViewProfile	1.29	9	
SubDealerViewStock	1	1	
Package	v(G)avg	v(G)tot	
com.example.siddhant.electrobasket	2.06	107	
Module	v(G)avg	v(G)tot	
app	2.06	107	
Project	v(G)avg	v(G)tot	
project	2.06	107	

CHAPTER 4

TEST DOCUMENT

Requirement ID	Class/Component Name	Method Name	Input parameters or scenarios	Expected Output	Real Output	Status(Pass/Fail)
4.1/4.2/4.3	**Login**	**checkCrediantials()**	username and password	log in successfully	logged in successfully	Pass
			incorrect password	Display an Error	Displays an Error	Pass
		Sign Up()	if not a user	direct to sign up page	directs to sign up page	Pass

4.4	**Sign Up**	**register_User()**	all detals of user	detais get stored in firebase	sign up done and details stored	Pass
4.5	**ViewOrder Dealer**	**ViewOrders()**	Click View Orders Button	Open View Orders Page	Opens View Orders Page	Pass
4.6	**ViewOrder Sub-dealer**	**ViewOrders()**	Click View Orders Button	Open View Orders Page	Opens View Orders Page	Pass
4.7	**ViewOrderSalesman**	**ViewOrders()**	Click View Orders Button	Open View Orders Page	Opens View Orders Page	Pass
4.8	**placeOrder (salesman)**	**placeOrders()**	click on button and place orders	place order	places order	Pass
4.9	**placeOrder (sub-dealer)**	**placeOrders()**	click on button and place orders	place order	places order	Pass
4.1	**View Stock**	**totalStockAvailable()**	click totalStockAvailable page	Open total stock available page	Opens total stock available page	Pass
4.11	**Executive Mapping**	**SalesmanNameinfo()**	Salesman Places its location	Show Loaction of theSaleman	Shows info about Saleman	Pass
		SubDealer_seeloaction()	Click Viwe Mapping Button	Show location of Salesman	Shows info about Saleman	Pass
4.12	**DealerView Profile**	**viewName()**	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		viewPhoneNo()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		viewGender()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass

		ViewAddress()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
4.12	**SubDealer ViewProfile**	**viewName()**	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		viewPhoneNo()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		editGender()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		ViewAddress()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
4.12	**SalesmanViewProfile**	**viewName()**	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		viewPhoneNo()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		editGender()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
		ViewAddress()	Click View Profile Button	Open View Profile Page	Opens View Profile Page	Pass
4.13	**Edit Profile**	**editName()**	Click Edit Profile Button.	Open Edit Profile Page.	Opens Edit Profile Page.	Pass
		editPhoneNo()	Click Edit Profile Button.	Open Edit Profile Page.	Opens Edit Profile Page.	Pass
		editGender()	Click Edit Profile	Open Edit Profile Page.	Opens Edit Profile Page.	Pass

			Button.			
		editAddress()	Click Edit Profile Button.	Open Edit Profile Page.	Opens Edit Profile Page.	Pass
		editPassword()	Click Edit Profile Button.	Open Edit Profile Page.	Opens Edit Profile Page.	Pass
4.14	**Retail Goods**	**Retail Goods()**	Click Retail Goods Button	Open Retail Goods Page.	Open Retail Goods Page.	Pass
		Save()	Click Save Button	Saves the info of Retail Goods	Saves the info of Retail Goods	Pass
4.15	Project Goods	**Project Goods()**	Click Retail Goods Button	Open Retail Goods Page.	Open Retail Goods Page.	Pass
		Save()	Click Save Button	Saves the info of Project Goods	Saves the info of Project Goods	Pass

CHAPTER 5

Minutes Of Meeting

The Client was contacted through phone and the whole structural idea was discussed through the phone. The client sent use the requirement document through mail and a video followed.
This gave us the clear understanding of what the client wanted and the features which he required in the application.

Also the progress of the application was shared regularly with the client through WhatsApp and phone.

Any Changes or additional requirements were informed to us through phone.

The Screenshots and the Video have been attached with this document and presentation.
Also the requirement file has been added.

Requirement File Given By the Client -

In order to make the App useful for our business we need the following requirements in the application –
The Application should have a Login and Signup Page for Dealer, Sub Dealer and Salesman respectively.
The Dealer Must have the Following Access –
1. He Must be able to View Orders Placed by the Sub Dealer
2. He Must be able to View the Stock Left with Him
3. He Must be able to View and Edit His Profile
4. He Must be able to View Project Goods
5. He Must be able to View the Location of the Salesman (Optional)
6. He Must be able to Compare and View Data Graphically (Optional)

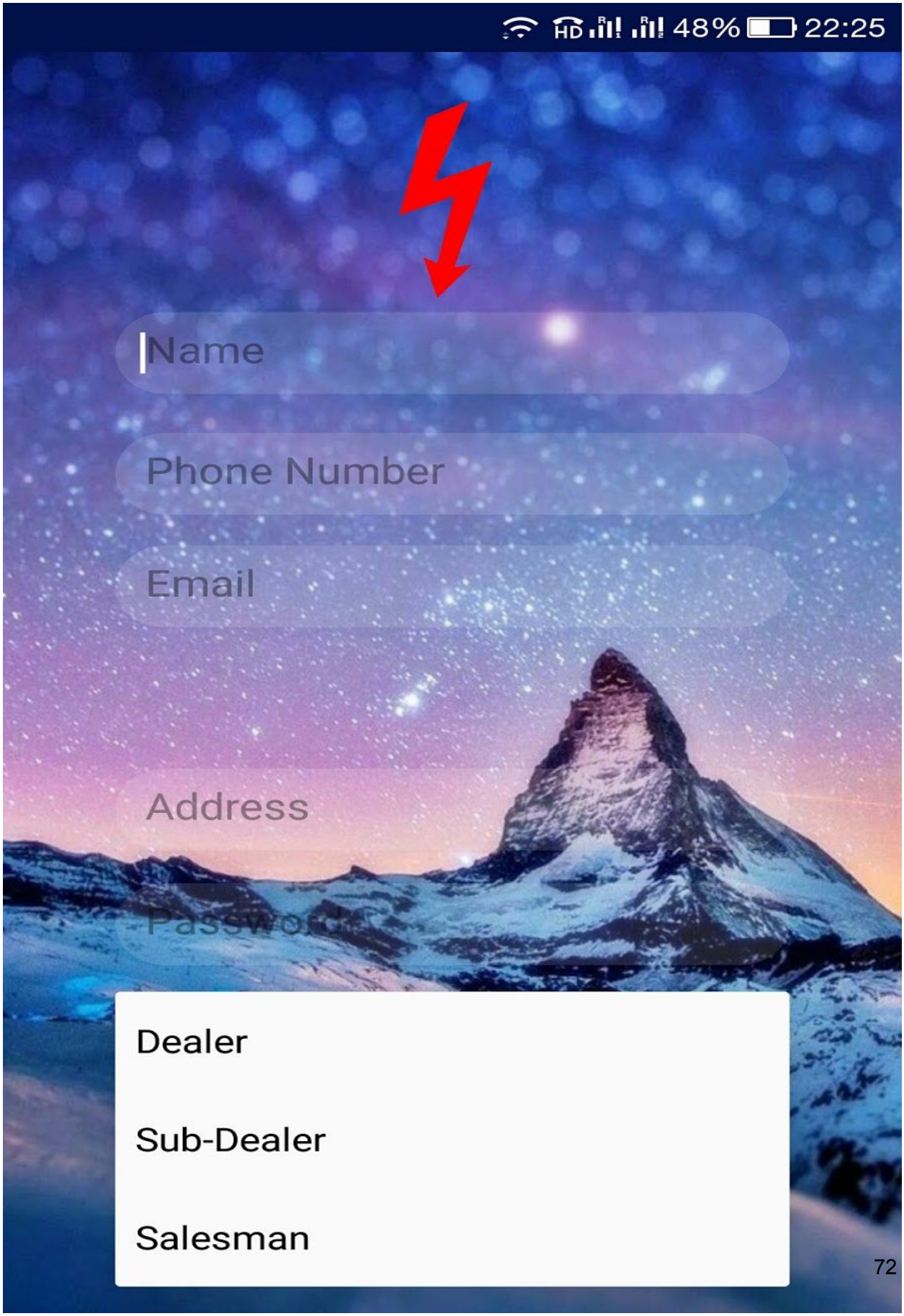
The Sub Dealer Must have the Following Access –
1. He Must be able to View Orders Placed by the Salesman
2. He Must be able to View the Stock Left with Him
3. He Must be able to View and Edit His Profile
4. He Must be able to Place Orders to the Dealer

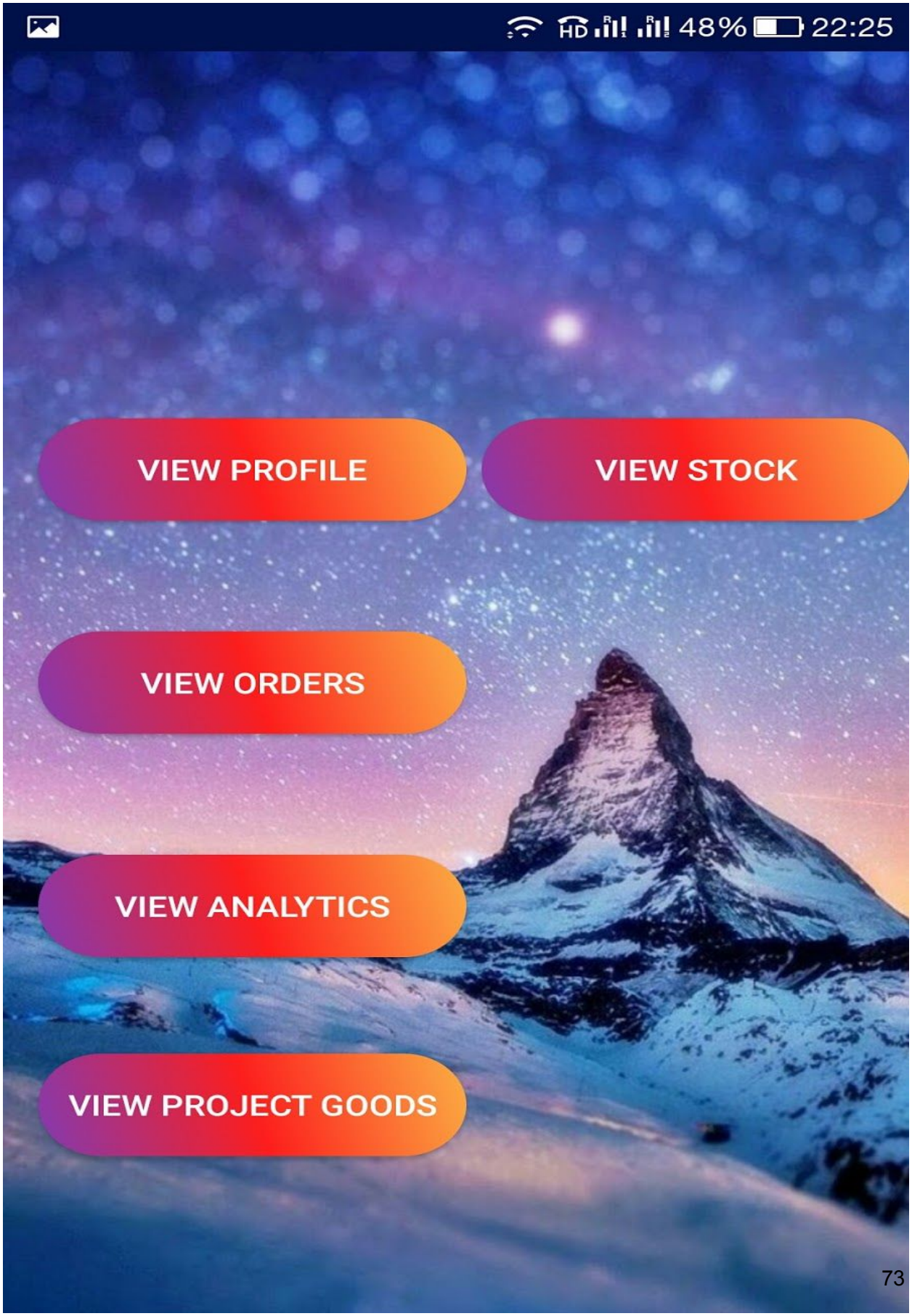
5.He Must be able to View the Location of the Salesman (Optional)

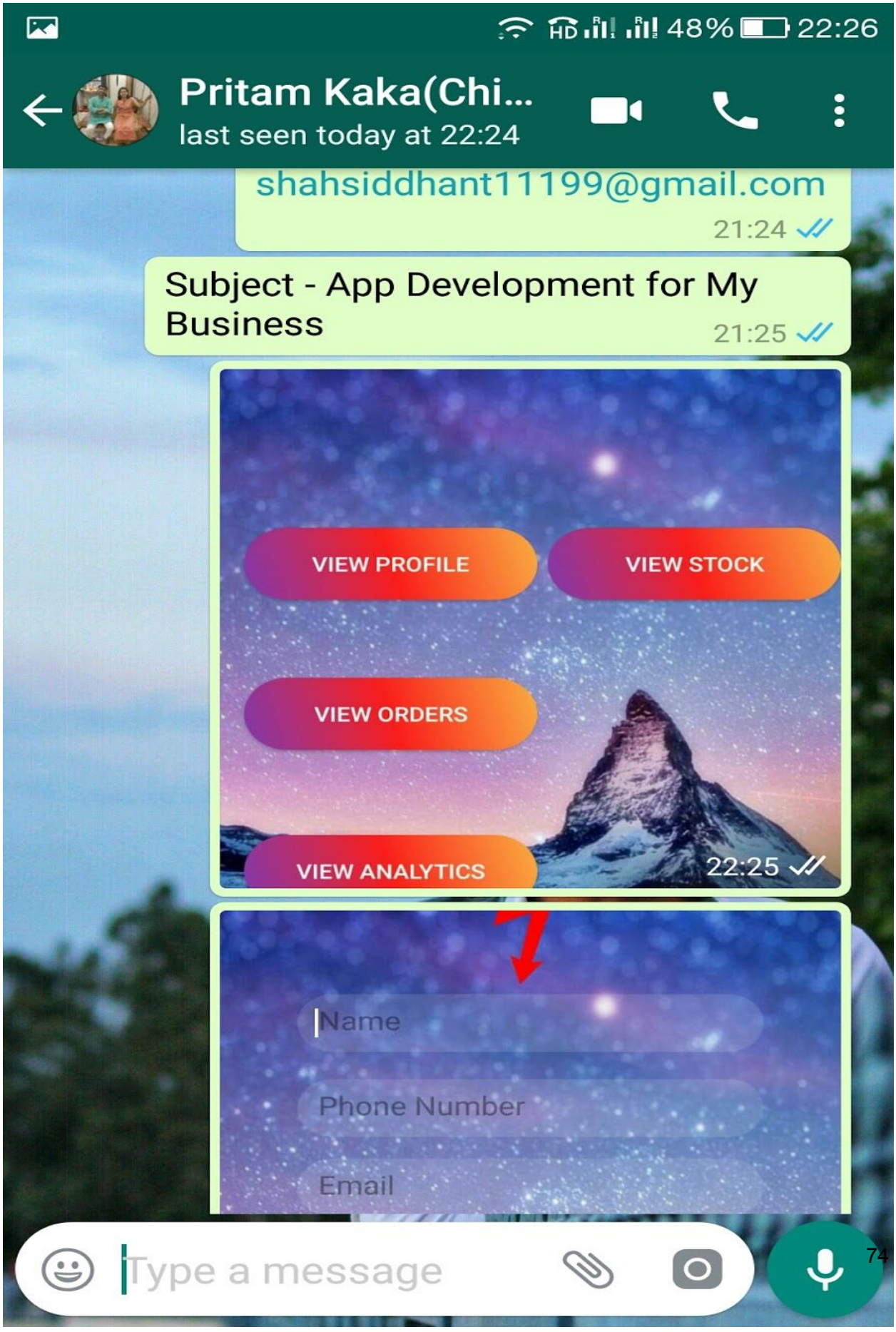
The Salesman Must have the Following Access –

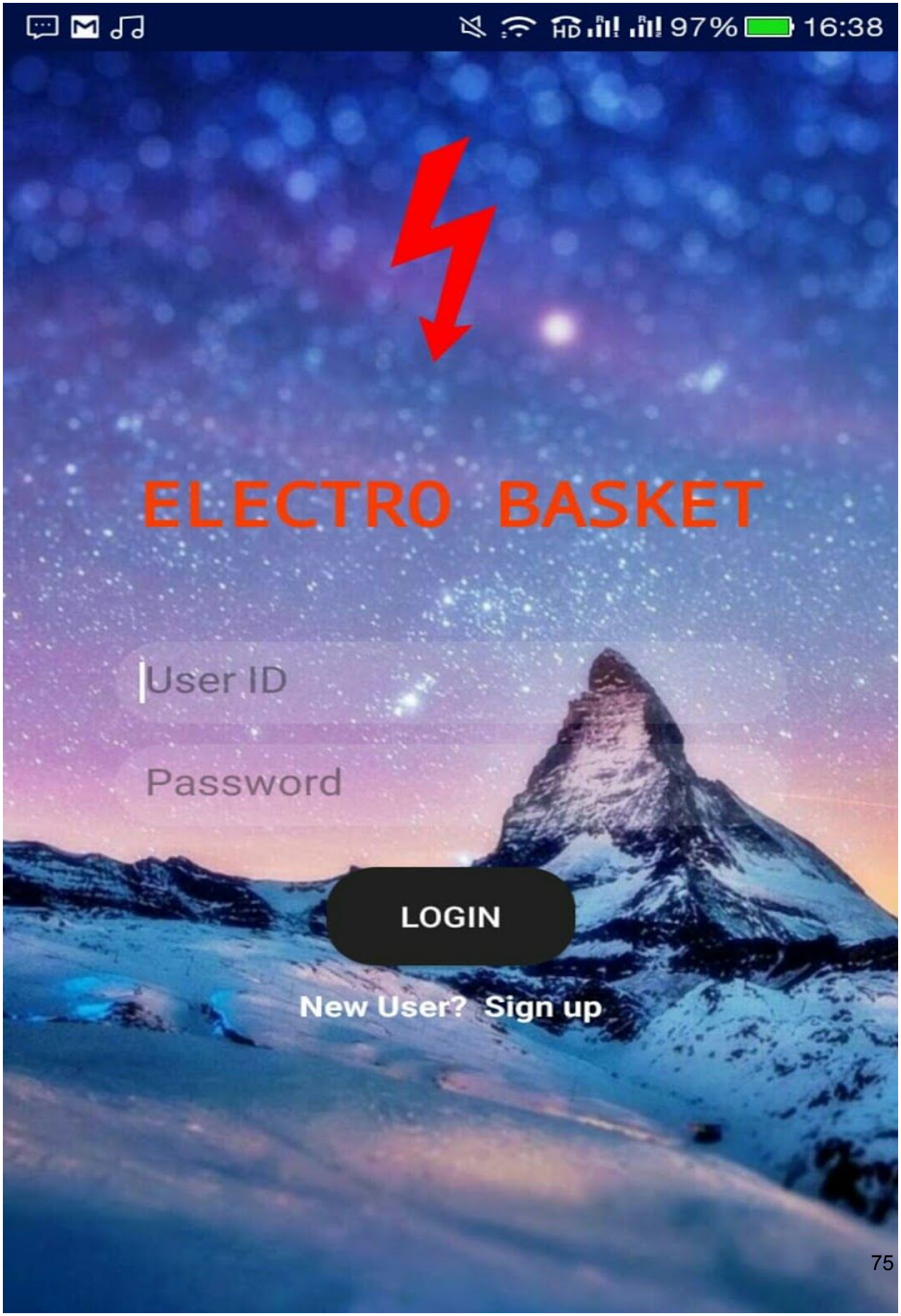
1. He Must be able to Place Orders to the Salesman
2. He Must be able to View and Edit His Profile

The Application must be Efficient and not Crash At times. It Also must have a proper User Interface and easy to Use.


48%
22:25
Name
Phone Number
Email
Address
Password
Dealer
Sub-Dealer
Salesman


48%
22:25
VIEW PROFILE
VIEW STOCK
VIEW ORDERS
VIEW ANALYTICS
VIEW PROJECT GOODS


48% 22:26
Pritam Kaka(Chi...
last seen today at 22:24
shahsiddhant11199@gmail.com
21:24
Subject - App Development for My Business
21:25
VIEW PROFILE
VIEW STOCK
VIEW ORDERS
VIEW ANALYTICS
22:25
Name
Phone Number
Email
Type a message


97%
16:38
ELECTRO BASKET
User ID
Password
LOGIN
New User? Sign up

CHAPTER 6
Nature of the Customer

Our Client Mr. Pritam Shah is the Main Dealer and runs their own business. He is the CSA(Chief Sales Agent) of the Great White Company. He wanted transparency in his business and thus to run his business with ease wanted and application which could track the Salesman and stay in contact with the Sub- Dealer.

So this application was developed keeping in view of how our client wanted it and what features were required.

Pritam Shah
9860800044

CHAPTER 7
Testimonials from Customer

After the completion of the application the apk file was sent to the client and after using the application a feedback video was sent by him as a testimony.
The video has been attached with the presentation.

CHAPTER 8

Tools and Technologies used during the project Development

The following Technologies have been used for the development of the Electro Basket application -

- Android Studio - Its the official platform to develop android application and test our application.
- Firebase - Its an Google online Database which stores the data according to the requirement of the application.
- ArgoUML - It is used to develop state diagrams, sequence Diagram, Class diagram and Use Case diagram.
- Google Docs - It is an easy way to share the documents with the group members and anyone can edit them
- Metrics Reloaded - It is an android plugin used to calculate the coding metrics of our application .

CHAPTER 9

NOVELTY OF THE PROJECT IDEA

The Electro Basket app is made specifically according to the the requirements of our client. Our client is the head of a company named Great White which manufactures all the electronic components required. The client needed an app which helps in better communication and transparency between the Dealer, Sub-dealer and Salesman . Therefore, this app is uniquely made for the client's use which shows it's newness in the existing Market.The electro basket has all the unique options and features that can only be used by Dealer , Sub-dealer and Salesman.

The Application is one of its type and there is no other application which provides the Features that this Application provides.
The Application comes With the "**Executive Mapping**" feature which enables the Dealer and the Sub-Dealer to See the Location of the Salesman Time to Time.
This Application Also Enables the Sub-Dealer and the Salesman to Place an Order of the requirements and then also Calculates the Total Price of the Order.
Hence of all these features make the application Unique and Easy to Use And provide Transparency to Our Client .

CHAPTER 10

SOPHISTICATION VALUE OF THE PROJECT

The making of the Electro Basket application along with the entire documentation took almost 3 months to get ready and complete. The major task was the entire documentation which took more amount of time as the Documentation included Software design ,Architecture Model, Pseudo codes and the entire SRS. Building The Application also took plenty of time as we as a group had to Learn Android Studio and Firebase . The documentation made before the Electro Basket app was a great help in making the application as all the architecture and the design prepared gave a clear view of how and what is to be made . It was easy to make the Layout of the application but the connection of the application was a bit difficult.

The Documentation helped in creating the Structure of the Application and provided a clear view of the features to be made to the Dealer , Sub-Dealer and the Salesman . The Location feature implementation also had to be learned first and then implemented.

Hence, because of the documentation the implicating became much easier.

CHAPTER 11

APPLICABILITY OF THE PROJECT

The application Electro Basket is made for the soul purpose to increase the transparency between the Dealer,Sub-dealer and Salesman which will help in better communication that will lead to efficient and smooth working between dealer, sub-dealer and salesman and will help to save time of the Dealer. The app is only made for the specific use of dealer, sub-dealer and salesman all the options and features are specifically made according to users requirement so the user can make the most out of the App and which will help the user to grow his Business.

This Application can be used to track the Salesman location, can be used by the Salesman to Place order of its requirements to the Sub- Dealer and the Sub Dealer can use it to place its own Requirements to the Dealer.

This Application can be used for the Similar Business which have this type of the Hierarchical System.