***Software Design***

***Specification***

for

*ElectroBasket App*

*Prepared by :*

*<Aman Khandelwal(U101116FCS007)*

*Arshita Batra(U101116FCS284)*

*Chinju Mary George(U101116FCS025)*

*Siddhant Shah(U101116FCS129)*

*Shashwat Shah(U101116FCS112)>*

*<Software Engineering CS-301>*

**Software Design Specification**

Table of Contents

**1. Introduction**

1.1 Purpose of this document

1.2 Scope of the development project

1.3 Definitions, acronyms, and abbreviations

1.4 Overview of document

**2. Conceptual Architecture/Architecture Diagram**

2.2 Structure and relationships

2.3 User interface issues

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

3.1 Logical Architecture Description

3.2 Class name: Login

3.4 Class Name: Dealer page

3.5 Class Name: Salesman page

3.6 Class Name: dealer edit profile

3.7 Class Name: Sub-dealer edit profile

3.8 Class Name: salesman edit profile

3.9 Class Name: sub-dealer page

3.10 Class Name: dealer view profile

3.11 Class Name: sub-dealer view profile

3.12 Class Name: salesman view profile

3.13 Class Name: View orders

3.14 Class Name: place order

3.15 Class Name:executive mapping

3.16 Class Name:goods

3.17 Class Name: view stock

3.18 Class name: view analysis

**4.0 Execution Architecture**

4.1 Reuse and relationships to other products

**5.0 Design decisions and tradeoffs**

**6.0 Pseudocode for components**

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

6.0.2 Class Name: Sign in

6.0.3 Class Name: DealerViewProfile

6.0.4 Class Name:SubDealerViewProfile

6.0.5 Class Name:SalesmanViewProfile

6.0.6 Class Name: DealerEditProfile

6.0.7 Class Name:SubDealerEditProfile

6.0.8 Class Name:SalesmanEditProfile

6.0.9 Class Name:ViewOrderSub-dealer

6.0.10 Class Name:ViewOrderDealer

6.0.11 Class Name:Executive-mapping

6.0.12 Class Name: Dealer(view stock)

6.0.13 Class Name:Sub-Dealer(view stock)

6.0.14 Class Name:placeOrder(salesman)

6.0.15 Class Name:placeOrder(sub-dealer)

6.0.16 Class Name:Analytics

**SOFTWARE DESIGN SPECIFICATION**

**The Software Design Specification Outline**

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**

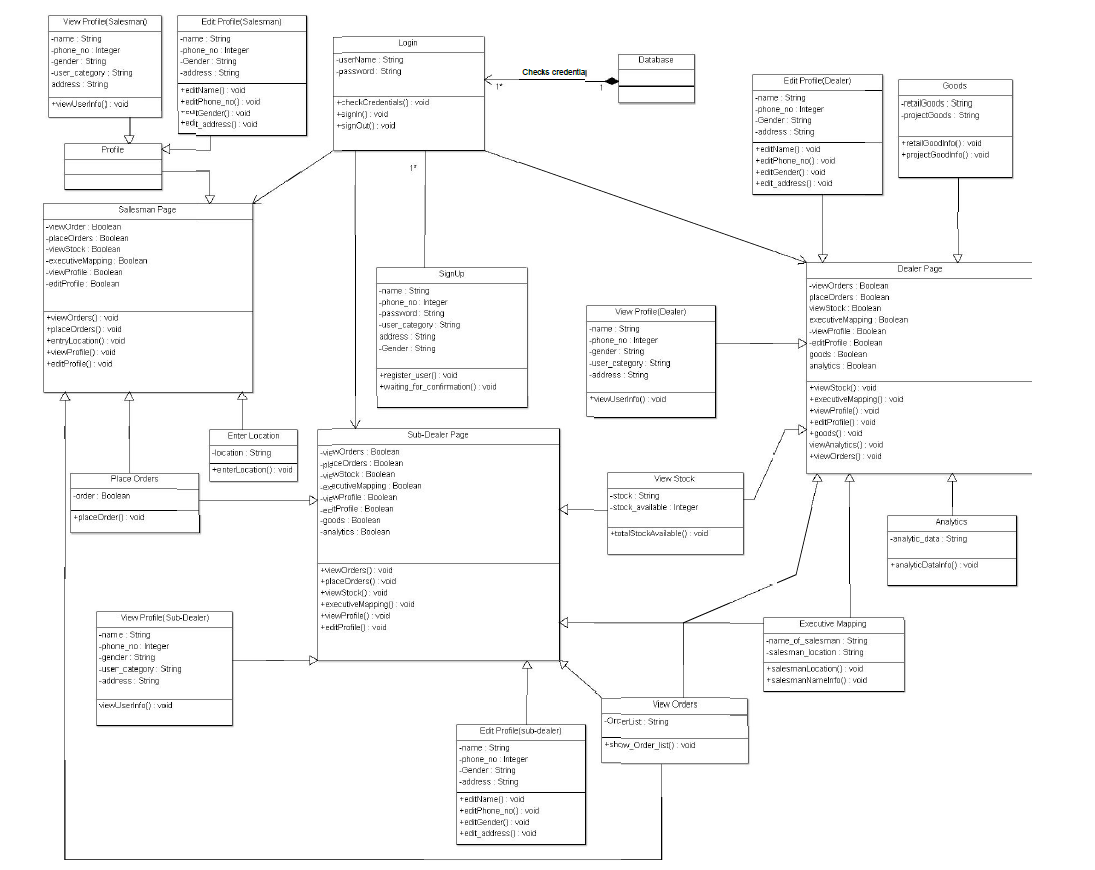
**2.1.1 Class Diagram explanation:**

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 .

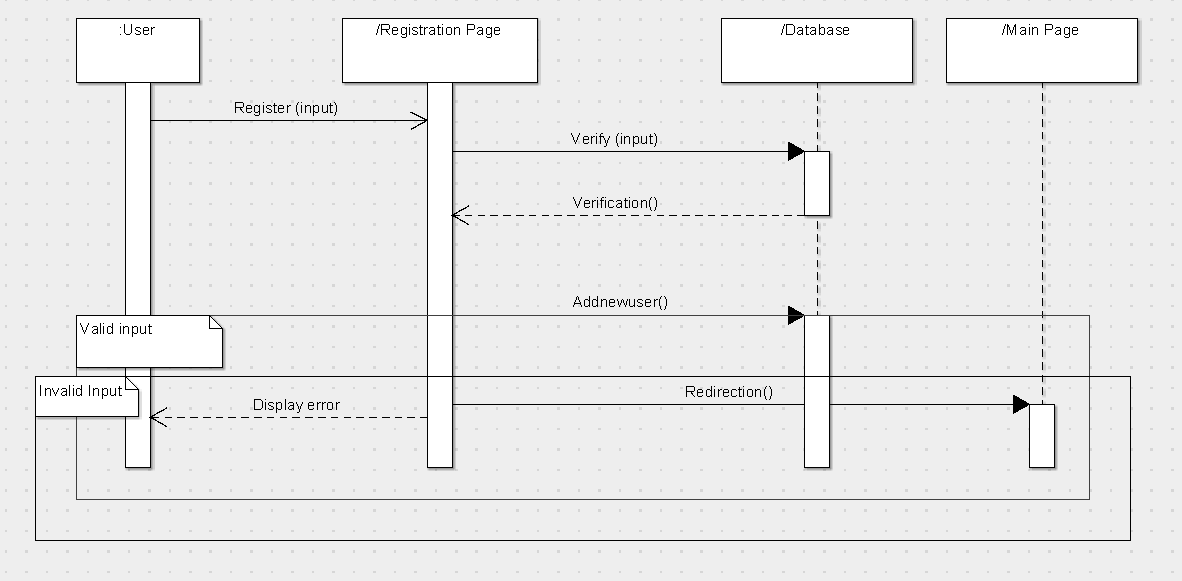
**3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)**

**Class Diagram:**

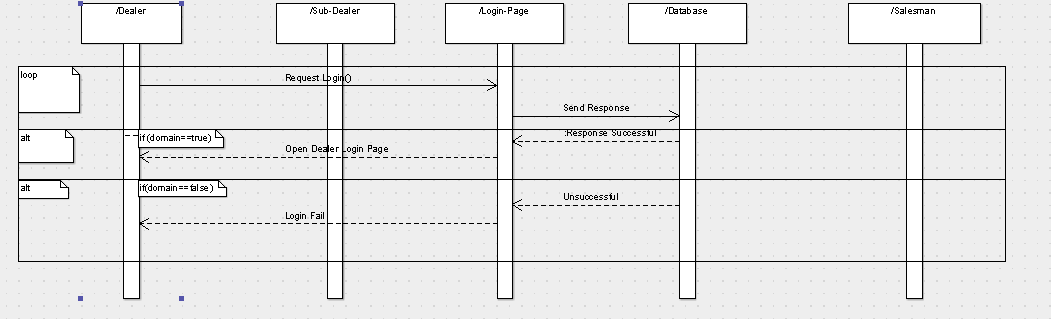
****

**Sequence Diagram:**

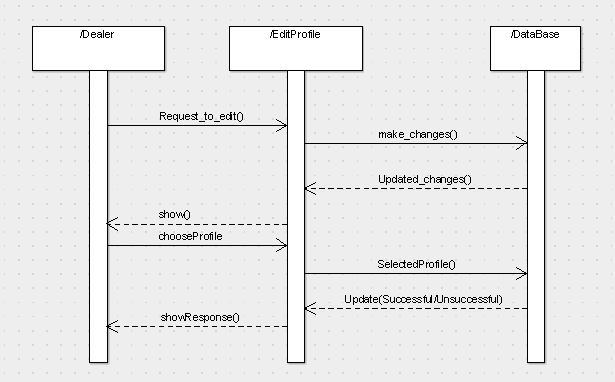
**SignUp**

****

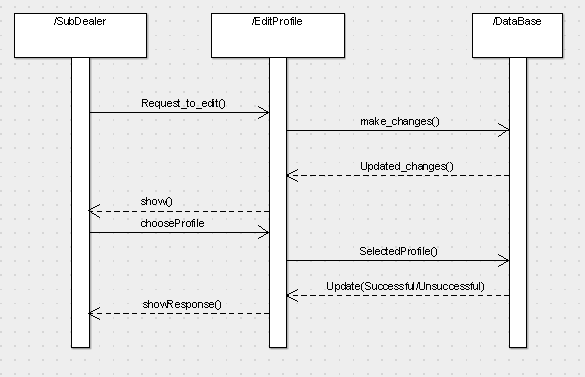
**Dealer Login**

****

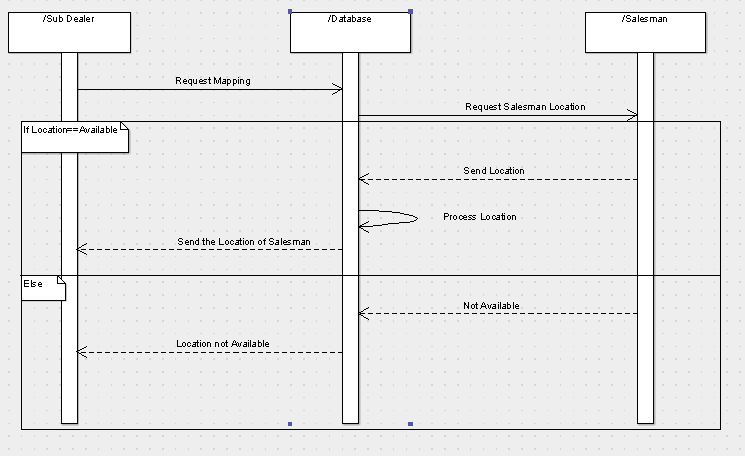
**Dealer Edit Profile**



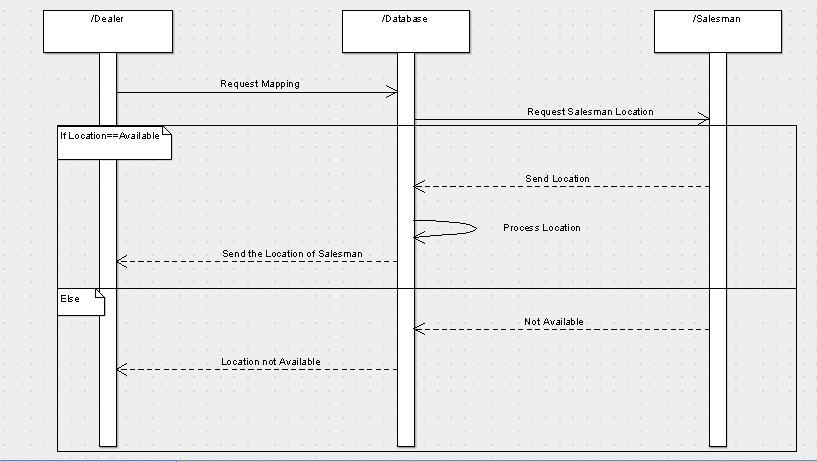
**SubDealer Edit Profile**



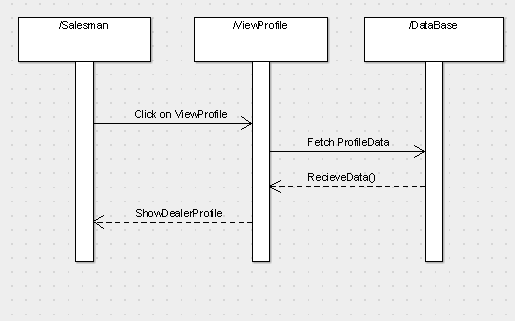
**Sub Dealer Executive Mapping**



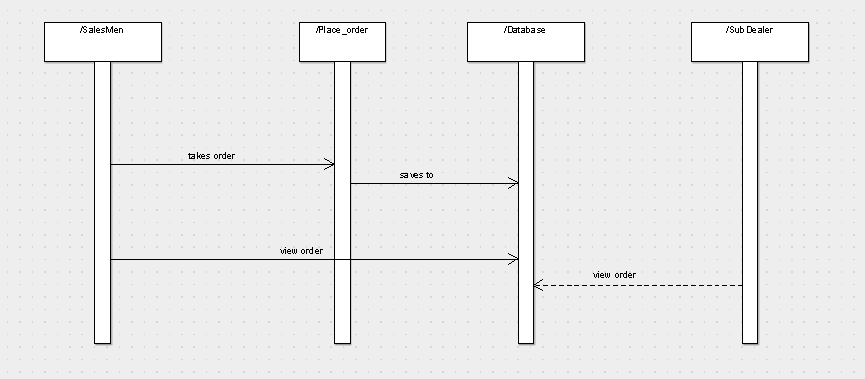
**Dealer Executive Mapping**

****

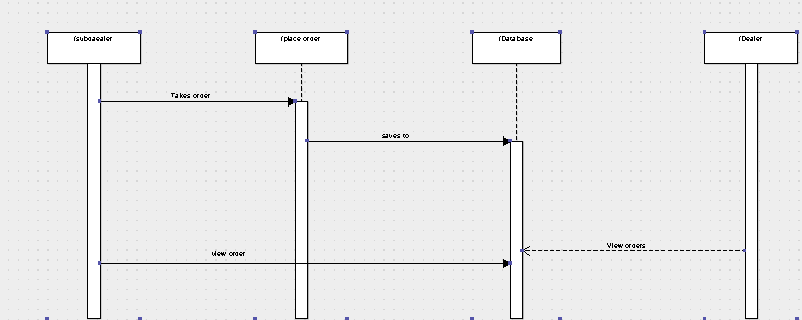
**Salesman View Profile**

****

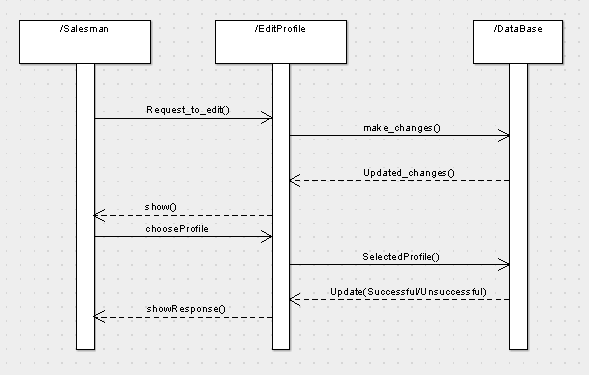
**Salesman Place Order**

****

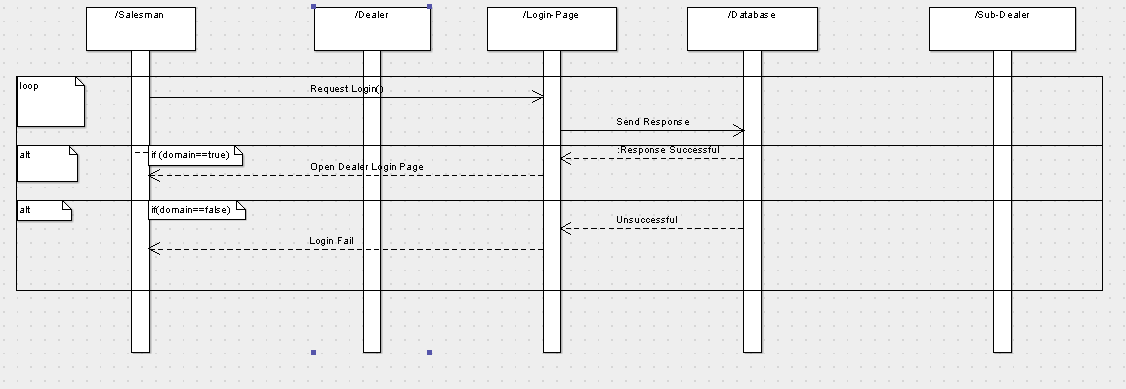
**Sub Dealer Place Order**

****

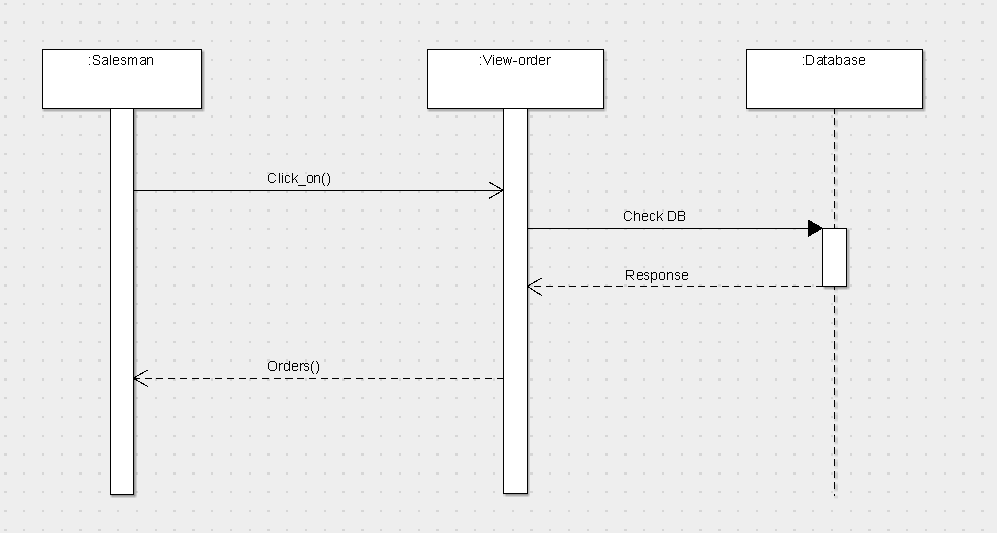
**Salesman Edit Profile:**



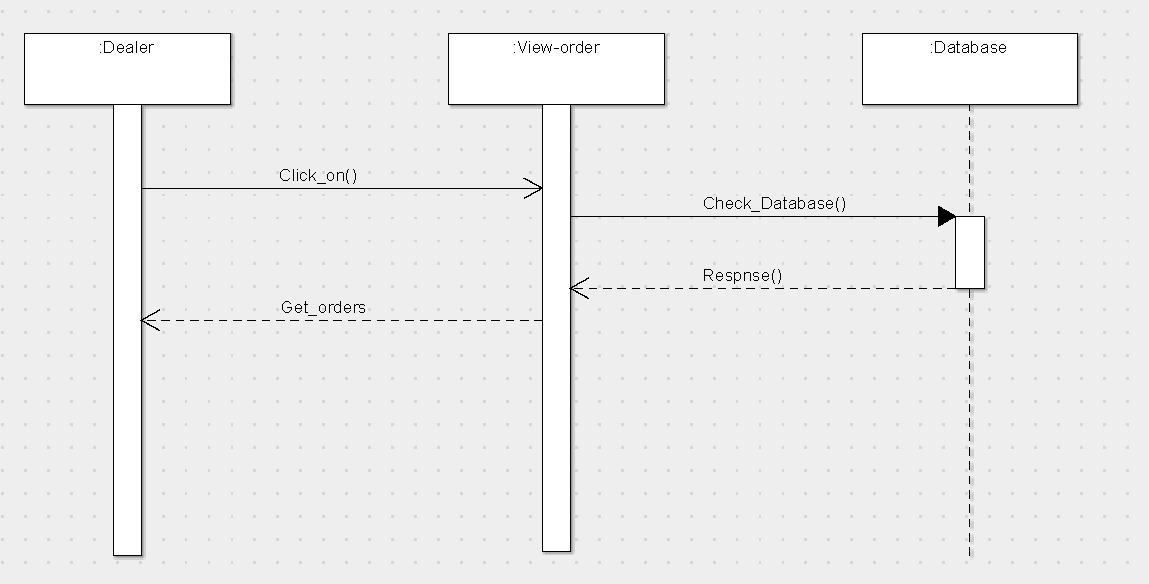
**Salesman Login:**

****

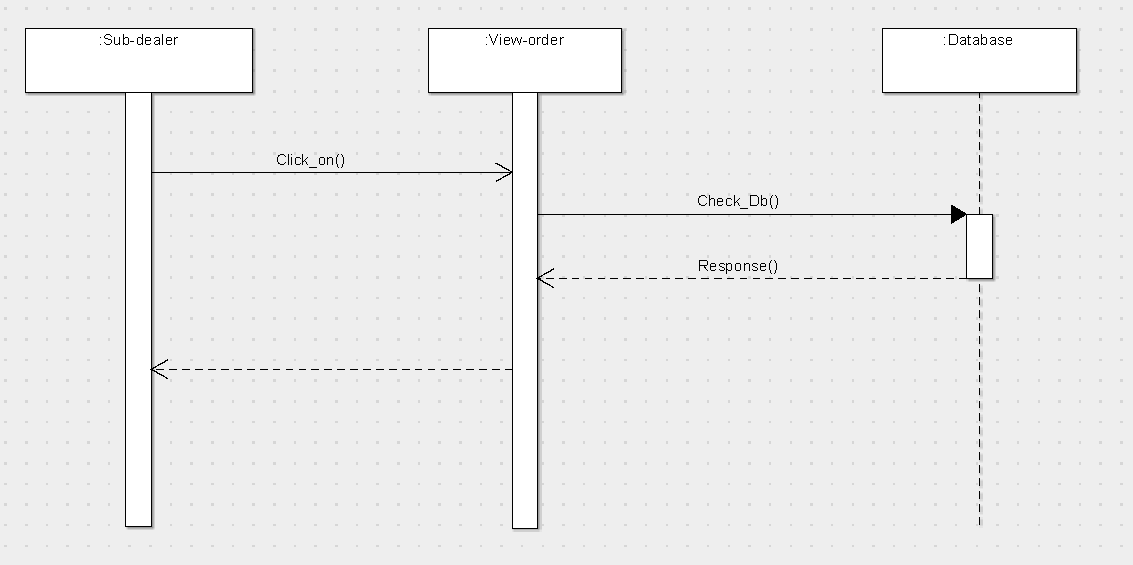
**Salesman View Order**

****

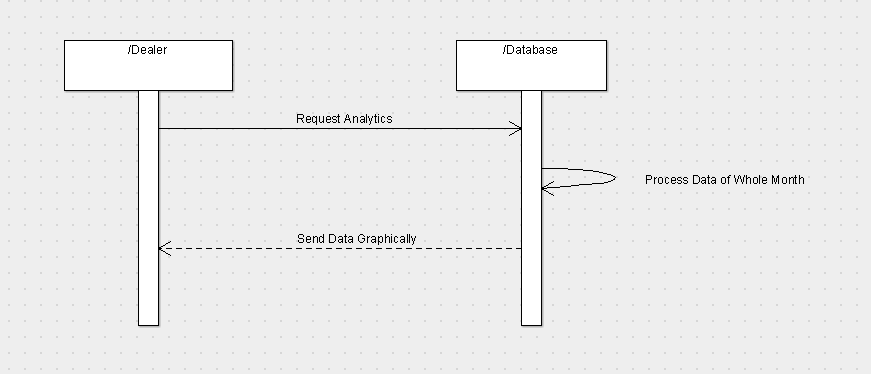
**Dealer ViewOrder**

****

**Sub-Dealer ViewOrder**

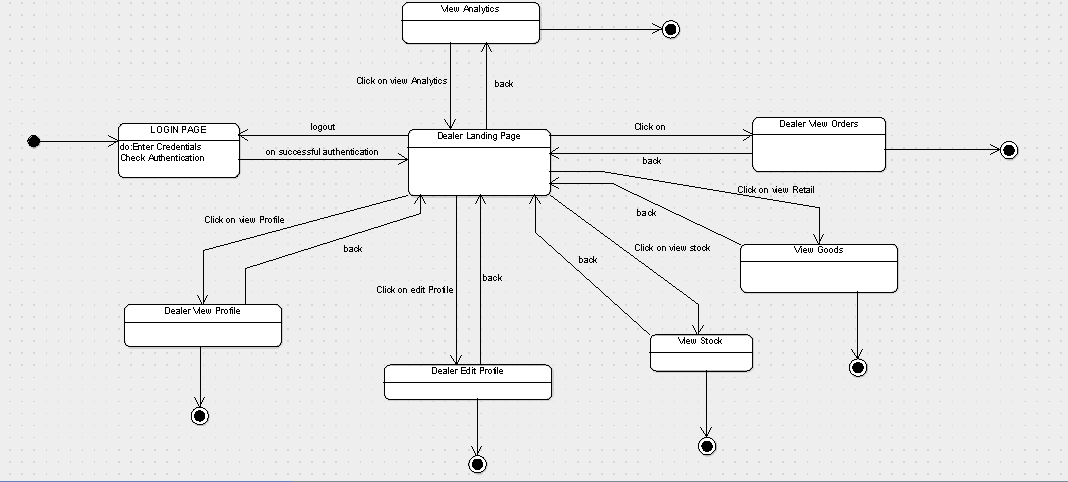
****

**View Analytics**

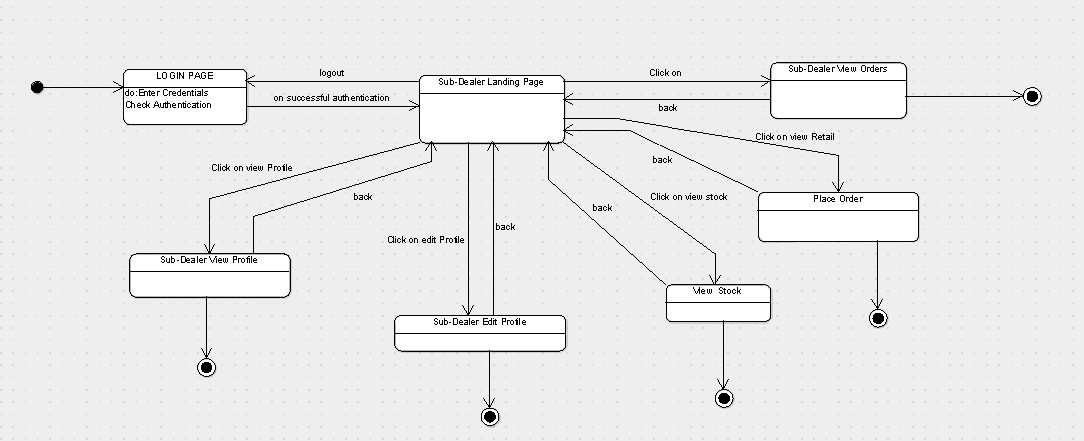


**State Diagram:**

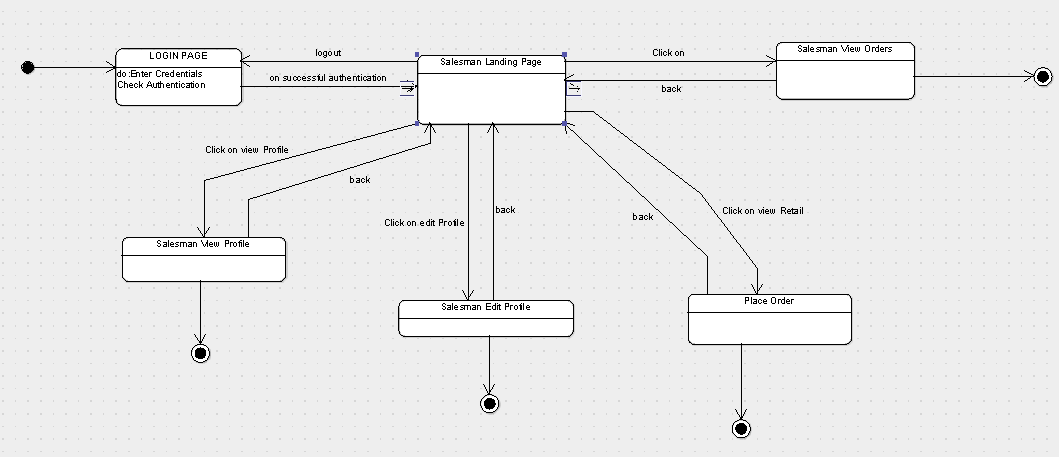
**Dealer Login:**



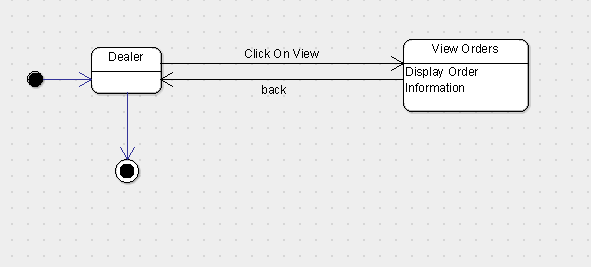
**Sub-Dealer Login:**



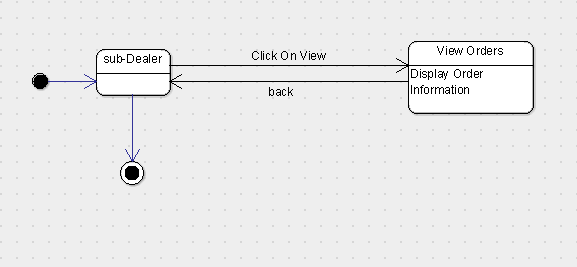
**Salesman Login:**



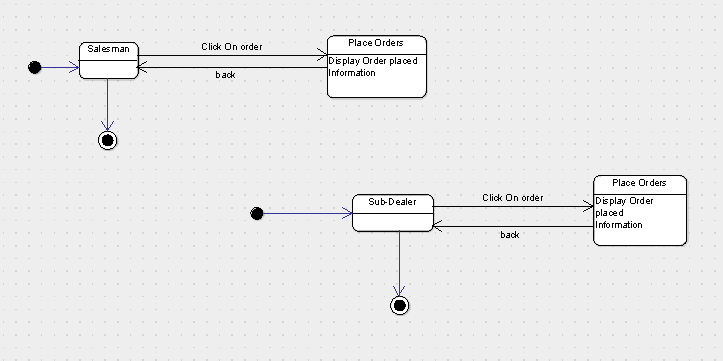
**View Order(Dealer):**

****

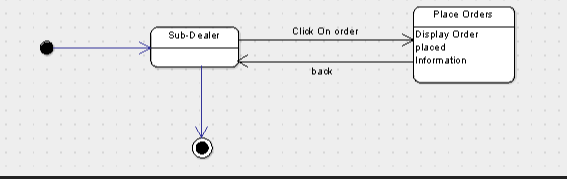
**View Order(Sub-Dealer) :**

****

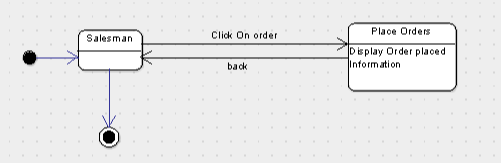
**View Order(Salesman):**



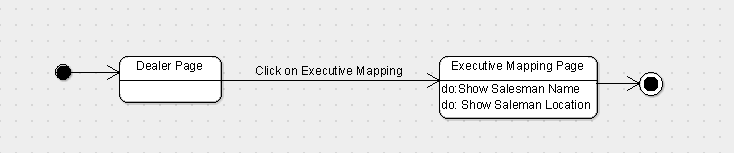
**Order Place(Sub-dealer):**

****

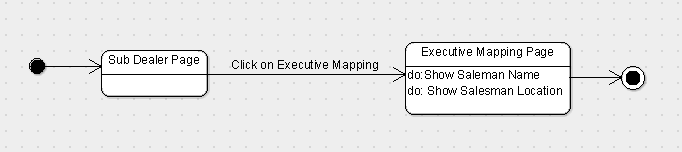
**Order Place(Salesman):**

****

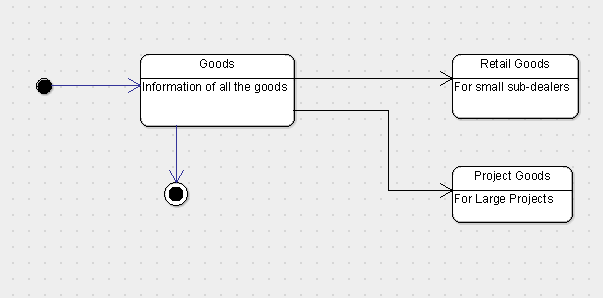
**Executive Mapping(Dealer):**

****

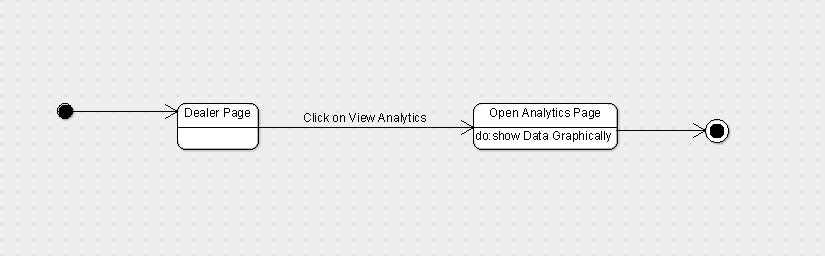
**Executive Mapping(Sub-Dealer):**

****

**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

1. while (logged\_in = false) do
2. begin
3. print("Please enter Mobile Number: ") //input Username
4. readln(inputted\_mobileno)
5. writeln("Please enter Password: ") //input Password
6. readln(inputted\_password)
7. //begin matching process
8. if inputted\_mobileno is in Login Database then
9. //match inputted login details with each existing valid set of Login Details from the Login Database
10. valid\_input\_mobileno:= true
11. if inputted\_password is in Login\_Database then
12. valid\_input\_mobileno := true;
13. if inputted\_password is in Login\_Database then
14. valid input\_password := true //Mobile no. and Password are both matched
15. logged\_in := true
16. writeln("Logins details have been validated)
17. else
18. valid input\_password := false
19. mobileno was matched but password wasn’t
20. logged\_in := false
21. writeln("Error your username or password was invalid")
22. end if
23. else
24. valid\_input\_mobileno := false
25. mobileno wasn’t matched
26. logged\_in := false
27. writeln("Error your mobileno or password was invalid")
28. end if
29. end while
30. if logged\_in = true then
31. writeln("'ou have been logged in")
32. else
33. writeln("'you could not be logged on due to false Login details")
34. end if
35. 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