*Florida International University*

*School of Computing and Information Sciences*

CEN 5011 Advanced Software Engineering - Fall 2016

Software Engineering Focus

Final Deliverable

Phone Shopping Network Version 1.0

**Team Members:**

Khaja Mohammed

Dewan Mohammad Moksedul Alam

**Product Owner(s)**:

Charles Green

Social Mobile USA

**Mentor(s)**:

Charles Green

Mohsen Taheri

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*The Social Mobile USA along with others making an effort to bring the phones that are part of the Lifeline "Obamaphone" program. They seek to provide online shopping facility to the users of these phones. These phones are all in the hands of low income families that could benefit from great deals on necessities as well as help in establishing credit. This is how they have enlisted ADUPS and Florida International University to help in this endeavor.*

*This document provides a solution in the* ***Phone Shopping Network Ver 1.0*** *system****,*** *a user driven approach that allows users of these phones which are a part of Obamacare program to register and purchase items, which are displayed on their mobile screens using ADUPS FOTA(Firmware Over-The-Air) technology. This document contains an extensive list of implemented user stories. It also provides a detailed plan of execution that includes hardware and software specifications. System design is explained through specific diagrams. System validation is also discussed. A glossary with essential domain terminology can also be found. Finally, the last chapter contains a series of appendices with UML samples, screenshots, and progress report.*

**Table of Contents**

[[Introduction 6](#_26fjg5sc0bjg)](#_Toc468996899)

[[Current System 6](#_26fjg5sc0bjg)](#_Toc468996900)

[[Purpose of New System 6](#_26fjg5sc0bjg)](#_Toc468996901)

[[User Stories 7](#_26fjg5sc0bjg)](#_Toc468996902)

[[Implemented User Stories 7](#_26fjg5sc0bjg)](#_Toc468996903)

[[User Story # 124 - Show Product Details 7](#_26fjg5sc0bjg)](#_Toc468996904)

[[User Story # 125 - Show Terms and Conditions 7](#_26fjg5sc0bjg)](#_Toc468996905)

[[User Story # 126 - User Registration 7](#_26fjg5sc0bjg)](#_Toc468996906)

[[User Story # 133 - Backend Support for User Registration 7](#_26fjg5sc0bjg)](#_Toc468996907)

[[User Story # 134- Backend Support for Product 7](#_26fjg5sc0bjg)](#_Toc468996908)

[[User Story # 135- Backend Support for Order Processing 7](#_26fjg5sc0bjg)](#_Toc468996909)

[[User Story # 157 - Management Console 7](#_26fjg5sc0bjg)](#_Toc468996910)

[[Pending User Stories 7](#_26fjg5sc0bjg)](#_Toc468996911)

[[Project Plan 8](#_26fjg5sc0bjg)](#_Toc468996912)

[[Hardware and Software Resources 8](#_26fjg5sc0bjg)](#_Toc468996913)

[[Development Environment 8](#_26fjg5sc0bjg)](#_Toc468996914)

[[Deployment Environment 9](#_26fjg5sc0bjg)](#_Toc468996915)

[[Sprints Plan 10](#_26fjg5sc0bjg)](#_Toc468996916)

[[Sprint 1 10](#_26fjg5sc0bjg)](#_Toc468996917)

[[Sprint 2 10](#_26fjg5sc0bjg)](#_Toc468996918)

[[Sprint 3 10](#_26fjg5sc0bjg)](#_Toc468996919)

[[Sprint 4 11](#_26fjg5sc0bjg)](#_Toc468996920)

[[Sprint 5 11](#_26fjg5sc0bjg)](#_Toc468996921)

[[Sprint 6 12](#_26fjg5sc0bjg)](#_Toc468996922)

[[Sprint 7 13](#_26fjg5sc0bjg)](#_Toc468996923)

[[System Design 14](#_26fjg5sc0bjg)](#_Toc468996924)

[[Architectural Patterns 14](#_26fjg5sc0bjg)](#_Toc468996925)

[[System and Subsystem Decomposition 15](#_26fjg5sc0bjg)](#_Toc468996926)

[[Deployment Diagram 17](#_26fjg5sc0bjg)](#_Toc468996927)

[[Design Patterns 17](#_26fjg5sc0bjg)](#_Toc468996928)

[[System Validation 18](#_26fjg5sc0bjg)](#_Toc468996929)

[[User Story # 126 - User Registration 18](#_26fjg5sc0bjg)](#_Toc468996930)

[[User Story # 133 - Backend Support for User Registration 19](#_26fjg5sc0bjg)](#_Toc468996931)

[[User Story # 135- Backend Support for Order Processing 20](#_26fjg5sc0bjg)](#_Toc468996932)

[[Glossary 22](#_26fjg5sc0bjg)](#_Toc468996933)

[[Appendix 23](#_26fjg5sc0bjg)](#_Toc468996934)

[[Appendix A - UML Diagrams 23](#_26fjg5sc0bjg)](#_Toc468996935)

[[Use case Diagrams 23](#_26fjg5sc0bjg)](#_Toc468996936)

[[Static UML Diagrams 28](#_26fjg5sc0bjg)](#_Toc468996937)

[[Dynamic UML Diagrams 30](#_26fjg5sc0bjg)](#_Toc468996938)

[[Appendix B - User Interface Design 34](#_26fjg5sc0bjg)](#_Toc468996939)

[[Appendix C - Sprint Review Reports 37](#_26fjg5sc0bjg)](#_Toc468996940)

[[References 39](#_26fjg5sc0bjg)](#_Toc468996941)

# Introduction

Phone Shopping Network Ver 1.0 is a private android application that offers users of the phones that are part of the Lifeline "Obamaphone" program, online shopping facility which allows user to register an account and purchase items, which are displayed on their mobile screens using ADUPS FOTA(Firmware Over-The-Air) technology. These phones are all in the hands of low income families that could benefit from great deals on necessities as well as help in establishing credit.

This introductory chapter is divided into four segments. The first section, Current System, describes how the Phone Shopping Network handles the User the user registration and information storing. The second section, Purpose of New System, explains the main objective of Phone Shopping Network Ver 1.0 and how it will impact both the users of the phones in a positive way.

## Current System

We are looking to do a test trial of 500 to 1000 phones that are part of the Lifeline "Obamaphone" program and hopefully expand from there. These phones are all in the hands of low income families that could benefit from great deals on necessities as well as help in establishing credit. This is why the new android application is being proposed.

## Purpose of New System

The process of Phone Shopping Network that the Social Mobile wants to uses the Obamaphone program phones. In the event that a user of the phone register, a user opens the application from his mobile and is prompted with a Terms and Condition page where the user accepts or rejects the terms and conditions of the application, and then the system checks if the user is already registered in the system or it is a new user, if a new user then the registration page is shown where the user register into the application and then finally the product page is shown to the user with the list of products for them.

The backend system allows an administrator to manage the user registration system and push the products on the application.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the Phone Shopping Network Ver 1.0 project. These user stories served as the basis for the implementation of the project’s features. This Comprehensive list of user stories that were implemented for the Ver 1.0. No planned user stories are contained in this chapter given that all of the ones that were requested by the product owners were delivered on time. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

### User Story # 124 - Show Product Details

As a user I would like to see more information of the product presented to me so that I can be able to determine whether it is relevant for me or not.

### User Story # 125 - Show Terms and Conditions

As a service provider, I would like to show the terms and conditions so that I can get user’s approval into the system.

### User Story # 126 - User Registration

As a user, I would like to see the user registration page so that i can register into the system.

### User Story # 133 - Backend Support for User Registration

As a user I would like to have my user information persisted so that I can access my profile from different phone.

### User Story # 134- Backend Support for Product

As an admin I would like to have proper support in the backend of the system for product management so that I can create new products, update and retrieve existing products.

### User Story # 135- Backend Support for Order Processing

As a user I would like to have proper support in the backend of the system to process product order so that I can place my order, update recent orders and retrieve my orders later.

### User Story # 157 - Management Console

As an admin I would like to have an application to manage the entities of the system so that I can do it conveniently and confidently

## Pending User Stories

#### User Story # 137 - Order placement from phone

As a user, I would like to purchase the product I was being shown so that I can get the proper benefit from the application.

# Project Plan

The overall goal coming into Phone Shopping Network version 1.0 was to implement most of the features that were necessary to bring the android application along with the web application to the functionality that existed within the first two iterations of the application. This was to be accomplished by building a brand new application from scratch as this is the first version of the application, an android and spring framework environment has to be used. The backend server application will be hosted on the Amazon web services.

The first segment of this chapter delves into the hardware and software that is necessary to integrate the system in terms of development and deployment. The second segment goes into detail on the individual sprints and which user stories were produced.

## Hardware and Software Resources

In order to develop and deploy the application some hardware and software is required. The following sections will describe the environments needed.

### Development Environment

In order to get involved in the development of the application, the following hardware specs and technical skills are recommended:

#### Computing environment

Any desktop or laptop having any operating system, preferably Windows/Mac/Linux. The following system configuration is preferable for smooth development experience.

* Min of 8Gb of RAM
* At least 2.0 Ghz processor
* Android mobile phone (better if provided by Social Mobile) for testing

#### Technical background

The major technology of development is Java 8. Developers will need to have good understanding of how to develop software using Java along with some best practices of doing this and latest features of Java. Apart from these working knowledge of the following are required:

* **Android application development**
  + Android
  + Experience with Android Studio
* **Server component**
  + J2EE - specially JSP, Servlet, Java web application
  + Hibernate
  + Spring framework
  + Jersey API for RESTful Webservices in Java
  + MariaDB (MySQL) and SQL - some knowledge
* **Web Console**
  + HTML5, CSS3, Javascript
  + Jquery, Bootstrap, KnockoutJS
* **Others**
  + JSON
  + Git and GitHub for source code management system
  + Gradle and Maven for build management
  + Any Java IDE
  + Apache tomcat as application server

### Deployment Environment

* **Server:** The artifacts of the server component and the web console are web archive (.war) files that can be deployed to any java based application server like Apache tomcat, Oracle’s weblogic, glassfish, IBM’s websphere, JBoss, etc.
* **Database:** MariaDB 5.6+ is thought to be used as the relational database service provider. But MySQL server 5.6+ can also be used with the same sql scripts provided with the source code.
* **Android App:** The android application is meant to be installed on the phones during being manufactured. As a result it may not be published to public android app store.

## Sprints Plan

These are the user stories that were released per sprint.

### Sprint 1

(08/29/2016 - 09/09/2016)

**User Story #125 - Show Terms and Conditions**

**Tasks**

* Add a menu item to show Terms and Conditions
* Design a view and its controller for presenting Terms and Conditions

**Acceptance Criteria**

* For the first time when a user interacts with an Ad, terms and conditions must be shown to him/her
* A user can view the terms and conditions on his/her will.

**Modeling**

Refer to UML diagrams in Appendix A: **Figure-UCD#125:** Use case diagram of Terms and Condition

### Sprint 2

(09/12/2016 - 09/23/2016)

**User Story #126 - User Registration**

**Tasks**

* Design a view and its controller for registration module for the user

**Acceptance Criteria**

* If the user is not registered already

**Modeling**

Refer to UML diagrams in Appendix A: **Figure-UCD#126**, Use case diagram for user registration, & **Figure-SD#126:** Sequence diagram of User Registration

### Sprint 3

(09/26/2016 - 10/07/2016)

**User Story #124 - Show Product Details**

**Tasks**

* Design and integrate a view for showing product details
* Establish architecture or convention for the entry point

**Acceptance Criteria**

* User is already registered
* User must accept the Terms and Conditions

**Modeling**

Refer to UML diagrams in Appendix A: **Figure-UCD#124** Show Product Details, & **Figure-SD#124:** Sequence diagram of Show product details

### Sprint 4

(10/10/2016 - 10/21/2016)

**User Story #133 - Backend Support for User Registration**

**Tasks**

* Establish the initial architecture of the backend component
* Design database
* Design and implement ORM model
* Design and implement an API to access user information
* Design and implement a controller for user management
* Integrate user registration from client side

**Acceptance Criteria**

* The user information is stored in a remote persistent storage
* The user can have a way to access old profiles on a new device
* The user information can be updated
* User information can be removed from the system
* A list of user profiles can retrieved

**Modeling**

Refer to UML diagrams in Appendix A: Static UML Diagrams, **Figure-CD#133** Use case diagram for user registration, & **Figure-SD#133:** Sequence diagram of Backend support for user registration**.**

### Sprint 5

(10/24/2016 - 11/04/2016)

**User Story #134 - Backend support for product management**

**Tasks**

* Add necessary relational objects
* Add necessary API for product management
* Add a corresponding DAO in Data Access layer
* Add a corresponding Web service Resource object in service layer

**Acceptance Criteria**

* A new product can be created through the service
* An existing product information can be retrieved
* An existing product information can be updated
* An existing product can be deleted from the system
* A list of products can be retrieved

**Modeling**

Refer to UML diagrams in Appendix A **Figure-UCD#134** Use case diagram for product management

### Sprint 6

(11/07/2016 - 11/18/2016)

**User Story #135 - Backend Support for Order Processing**

**Tasks**

* Add service layer for order management
* Implement data access layer for order management
* Add API for order management
* Create entities and mapping objects

**Acceptance Criteria**

* The backend system has capability to create, modify, retrieve and cancel orders
* After creating an order the systems provides necessary information for next steps
* User must accept the Terms and Conditions

**Modeling**

Refer to UML diagrams in Appendix A: **Figure-UCD#135:** Use case diagram of the OrderService subsystem, **Figure-CD#135:** Class diagram of OrderService subsystem, & **Figure-SD#135:** Sequence diagram of OrderService subsystem

**User Story #157 - Management Console**

**Tasks**

* Add backend support to list the objects
* Add a view to list the users
* Add a view to create and modify product
* Add a view to list the products
* Add a view to list orders

**Acceptance Criteria**

* A visual console to manage the entities of the system is implemented
* Access is restricted to that console

### Sprint 7

(11/21/2016 - 12/02/2016)

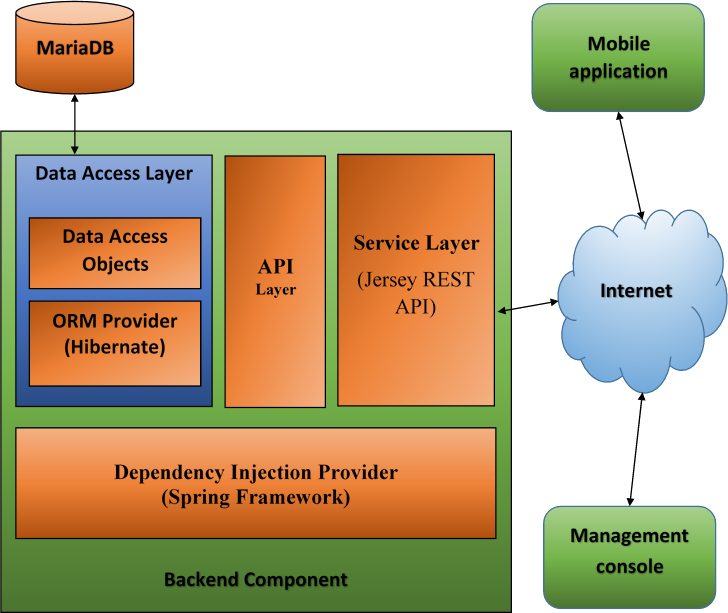
**Tasks**

* Wrapping up the project
* Create videos
* Poster presentation

# System Design

From a top view the whole system follows from Server-Client architecture. Where there is a centralized server side component and two different client side components. The server side component is an application following multi-tier architecture that provides the functionalities to process user requests for the management and persisting the entities of the system. One of the clients is an Android application that is the main user facing piece of the system. Another client application is a web based management console for the administrators of this system to review and modify the entities.

The whole system design is depicted in the following diagram.



**Figure:** System design

## Architectural Patterns

The more detailed overview about the adopted architectures for different parts of the system are as follows.

**Server Component**

The server side component is further divided into several layers.

* **Service Layer:** It is the boundary or interface to other systems. It is composed of several RESTful web service endpoints. This layer takes advantage of another layer which is the core Business Logic Layer.
* **Business Logic Layer (BLL):** This layer consists of some objects those manage the functionalities provided around the main concepts of the whole system. This Business Logic Layer is basically implements an API layer which defines the provided functionalities of the system along with naive implementations. The Business Logic Layer also utilizes the Data Access Layer (DAL).
* **Data Access layer (DAL):** The DAL maintains the communication to and from the database and takes care of the transaction management. The DAL consists of some Data Access Objects (DAOs) and Relational Entities. These objects are designed with the Object Relational Mapping (ORM) service provided by Hibernate following the standard Java Persistence API (JPA).
* **Transaction Manager:** All of these layers and components are wrapped around by a transaction management facility provided by Spring framework. This facility will ensure graceful handling of runtime errors and ensures that no data is incorrectly persisted or presented in such cases.
* **Dependency Injection (DI):** Moreover, the Spring framework binds all these layers and components together using DI.
* **Persistent storage:** MariaDB is used as the relational database for the system.

It is worth notifying in this regard is that, any of these layer can be completely replaced with another implementation to provide a different (possibly improved) implementation just by changing some configuration in the DI configurations.

**Web-based Management Console**

This follows Model-View-ViewModel (MVVM) architecture. We used KnockoutJS library to support this.

**Android Application**

This application mostly follows Model-View-Controller (MVC) architecture.

**Communication protocol**

All the communication among different components are designed to be made over http. The client applications communicate to the server through some RESTful webservice using http. The server communicates to the database using JDBC over http.

## System and Subsystem Decomposition

Phoneshopping Network 1.0 is divided into four major subsystems; Users, Products, Orders and Logistics. All of these subsystems except logistics are composed of UIs (separate in the management console and the android application), service resources, API and implementations, a repository, data access logics to and from the repository and data conversion mechanism from internal model to external model. The proper functionality of the system relies on the interaction among all these subsystems.

**UserProfile Subsystem**

This manages the CRUD operations along with the presentation both on the phone and the management console of “users”. The implemented functionalities related to this subsystem are shown in the use case diagram Figure-UCD#133 in Appendix A.

**ProductService Subsystem**

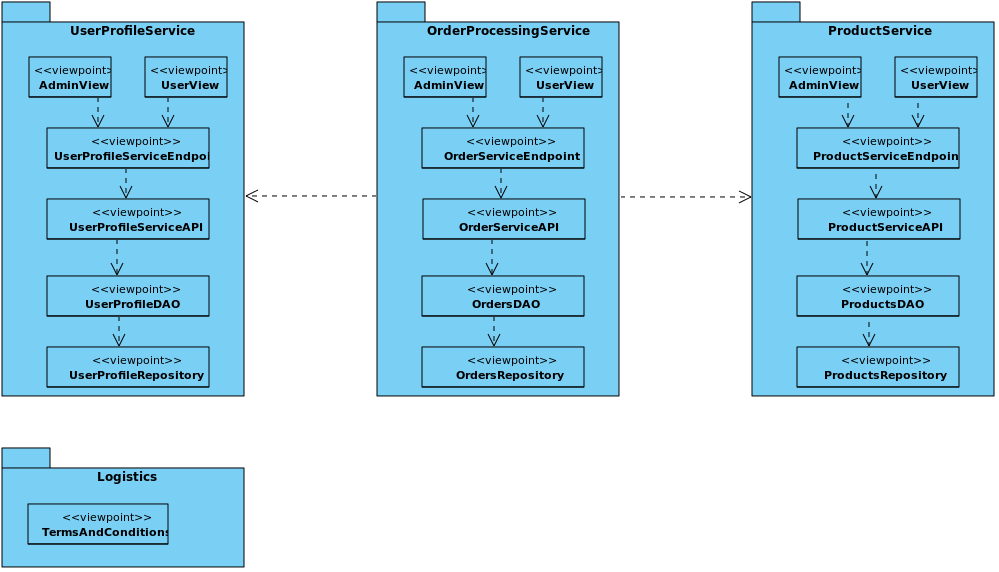
This manages the CRUD operations along with the presentation both on the phone and the management console of “products”. The implemented functionalities related to this subsystem are shown in the use case diagram - Figure-UCD#134

**OrderService Subsystem**

This manages the CRUD operations along with the presentation both on the phone and the management console of “orders”. The implemented functionalities related to this subsystem are shown in the use case diagram Figure-UCD#135 in Appendix A

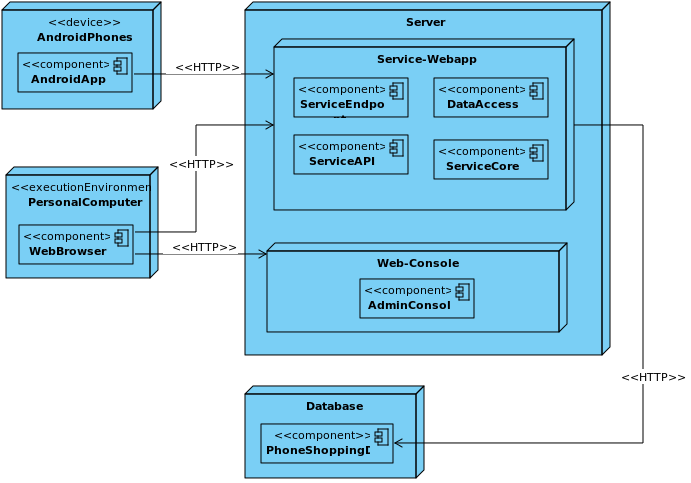
**Logistics Subsystem**

This is not the essential part of the system. This subsystem is thought to be used to organize the static resources, e.g. terms and conditions, about us, contact us, services, etc., of the system to keep the essential elements clean.



**Figure-Package:** Subsystems of the system

## Deployment Diagram



## Design Patterns

The following design patterns were used in the system

* Builder
* Factory method
* Bridge
* Observer
* Strategy

# System Validation

Given that the system is multi-layered in terms of architecture, unit testing is easier to implement. This section contains all of the test cases that were written as features were being implemented.

**User Story # 124 - Show Product Details**

**Unit Tests**

|  |
| --- |
| **Test ID**: U-124-01 |
| **Purpose:** To validate that the terms and conditions must be accepted before using the app. |
| **Preconditions:** The application is run for the first time. |
| **Input:** None |
| **Expected Output**: The view for presenting the terms and conditions page is chosen to show |

|  |
| --- |
| **Test ID**: U-124-02 |
| **Purpose:** To validate that the system displays the user registration page if not registered yet before presenting the product details. |
| **Preconditions:** The user has not been registered yet. |
| **Input:** None |
| **Expected Output**: The view for user registration is chosen to present |

## User Story # 126 - User Registration

**Unit Test**

|  |
| --- |
| **Test ID**: U-126-01 |
| **Purpose:** To test that the application accepts valid input as username |
| **Preconditions:** None |
| **Input:** try to create a user with following information  {"username":"","firstname":"First","lastname":"Last","email":"first\_last@example.com","phone":"+18888888"} |
| **Expected Output**:   * The application should show a failure message showing that the username is missing * No request should be made to server to create a user. |

|  |
| --- |
| **Test ID**: U-126-2 |
| **Purpose:** Assure that a user cannot be created if the user exists in database. |
| **Preconditions:** System must be functional and a user with username “example1” exists and the server can detect the existence and reports accordingly. |
| **Input:** New user creation request is made with the following information:  {"username":"example1","firstname":"First","lastname":"Last","email":"first\_last@example.com","phone":"+18888888"} |
| **Expected Output**:   * A request is made to server to create the userprofile * The application should detect and understand the reason of failure * Shows an error message that the username “example1” is already used. |

## User Story # 133 - Backend Support for User Registration

**Unit Test**

|  |
| --- |
| **Test ID**: U-133-1 |
| **Purpose:** To test that a user is created only if the username is valid |
| **Preconditions:** None |
| **Input:** A request is made to create a user with the following data  {"username":"","firstname":"First","lastname":"Last","email":"first\_last@example.com","phone":"+18888888"} |
| **Expected Output**: The system raises an invalid username exception and send error status. |

**Integration Test**

|  |
| --- |
| **Test ID**: I-133-1 |
| **Purpose:** To test that a user profile will be created only with a non-existing “username”. |
| **Preconditions:** A user with username “example1” exists |
| **Input:** A request is made to create a new user with the following data:  {"username":"example1","firstname":"First","lastname":"Last","email":"first\_last@example.com","phone":"+18888888"} |
| **Expected Output**:   * The system tries to save/insert user information * The database raises a duplicate entry exception * The system should detect the exception and send failure response accordingly with proper error message. |

## User Story # 135- Backend Support for Order Processing

**Unit Test**

|  |
| --- |
| **Test ID**: U-135-1 |
| **Purpose:** To test if the system fails to create an order without the username of the user who actually is placing the order |
| **Preconditions:** All other information are present and valid |
| **Input:** Try to create an order not mentioning any username. |
| **Expected Output**: The system fails to create an order entity by raising an exception mentioning that no valid username is provided as order placer |

**Integration Test**

|  |
| --- |
| **Test ID**: I-135-1 |
| **Purpose:** To test if the system fails to place an order when non existent username is mentioned as order placer |
| **Preconditions:** The user with username “example\_9” is not present and all other information are present and valid. |
| **Input:** Make a request to place an order with the following data  {"username":"example\_9", "billingAddress":{"addressId":11, "addressLineOne":"10274 SW 102 Ave suit 201", "city":"Miami", "state":"FL", "zipCode":"33178"}, "shippingAddress":{"addressId":12, "addressLineOne":"10274 SW 103 Ave suit 202", "city":"Miami", "state":"FL", "zipCode":"33178"}, "products":[{"productId":34,"unitPrice":9.94,"count":5}]} |
| **Expected Output**: The system fails with the exception that no registered user found having username “example\_9”. |

# Glossary

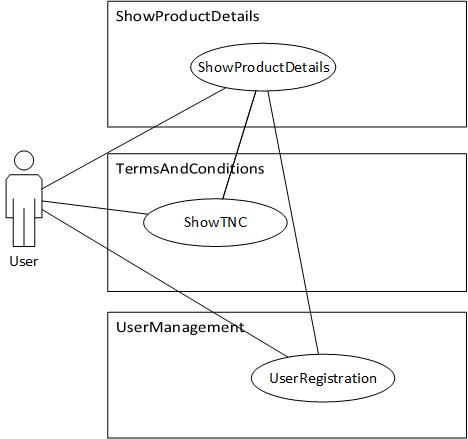
**User:** A user who has a Social mobile phone. Register and purchase products from the Phone Shopping Network application.

**Administrator:** Handles all the factors at the backend such as user registration management, Order Processing, Post Product information, Update user information and other administrative tasks.

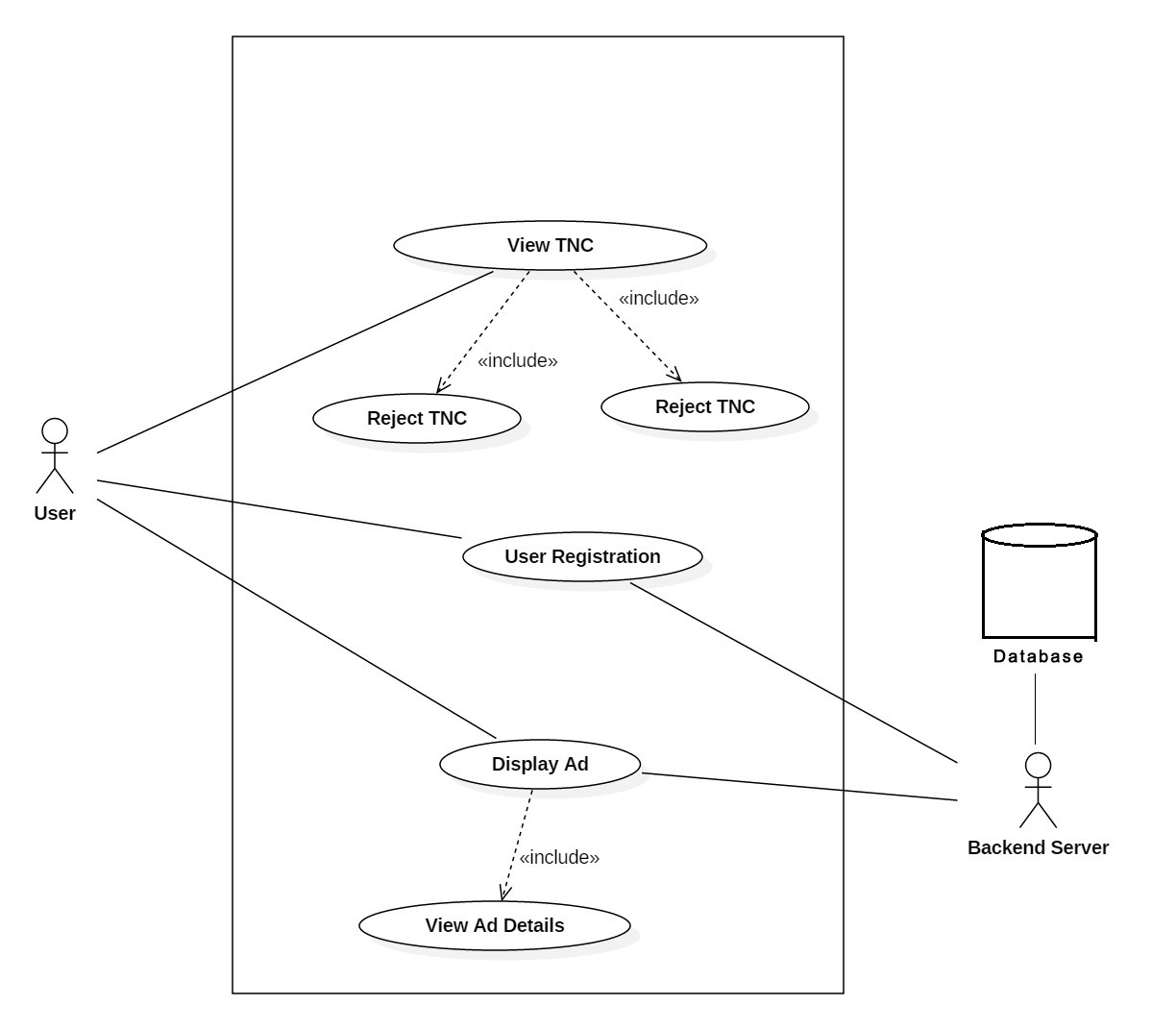
# Appendix

## Appendix A - UML Diagrams

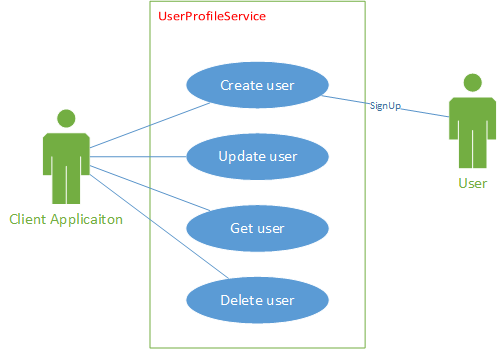
## Use case Diagrams



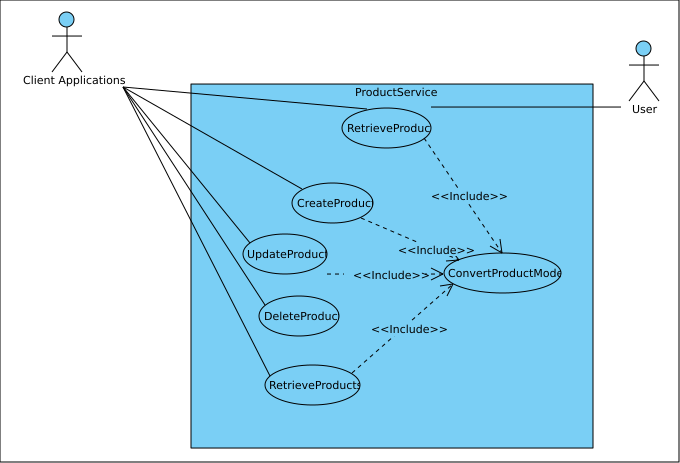
**Figure-UCD#125:** Use case diagram of Terms and Condition & **Figure-UCD#124:** Use case diagram of Show Product Details



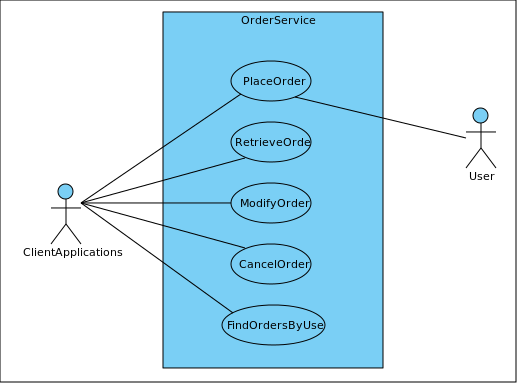
**Figure-UCD#126:** Use case diagram of user registration



**Figure-UCD#133:** Use case diagram of user profile subsystem

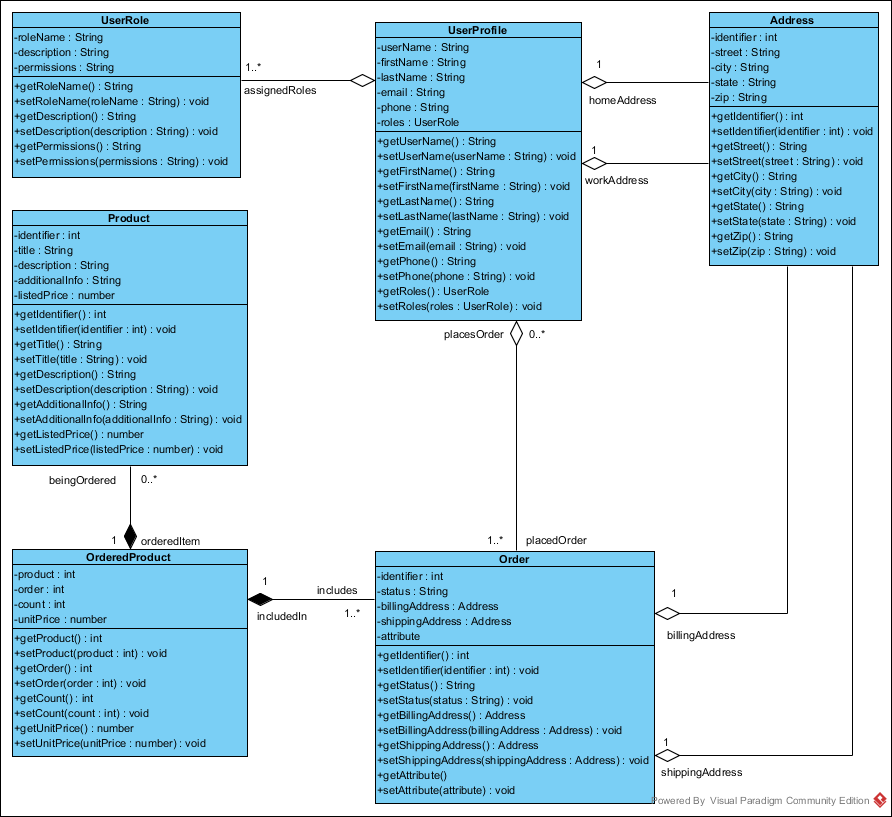


**Figure-UCD#134** Use case diagram for product service management

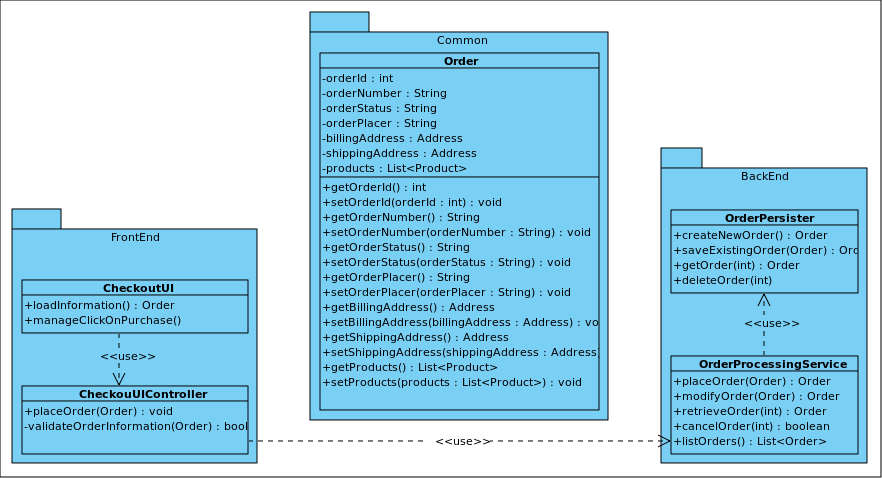


**Figure-UCD#135:** Use case diagram of the OrderService subsystem

## Static UML Diagrams

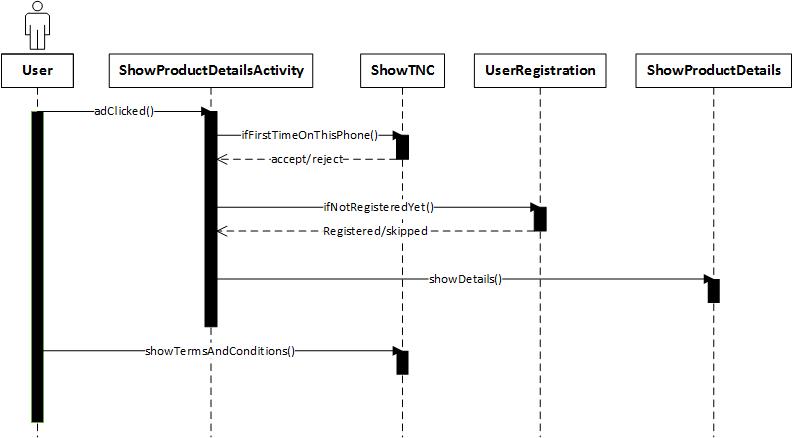


**Figure-CD#133:** Class diagram of Backend support for user registration

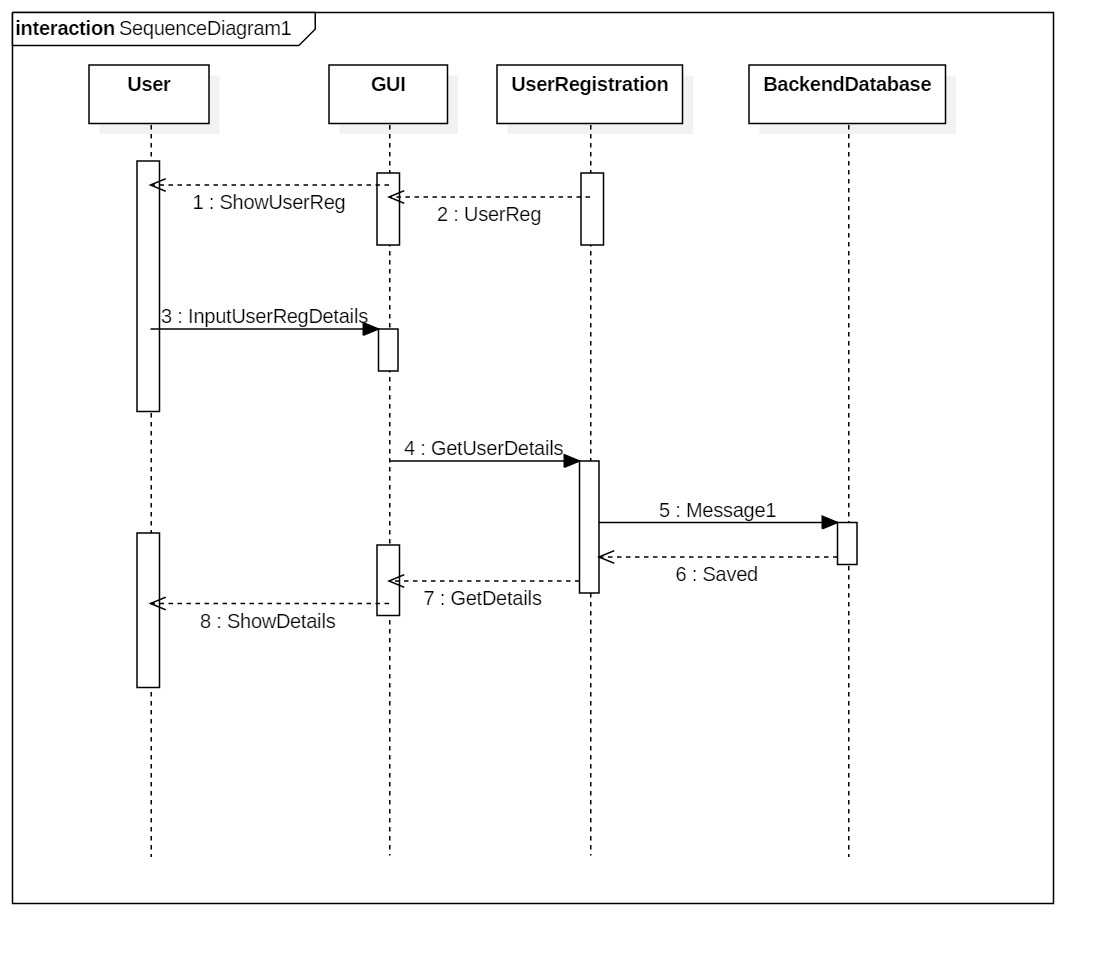


**Figure-CD#135:** Class diagram of OrderService subsystem

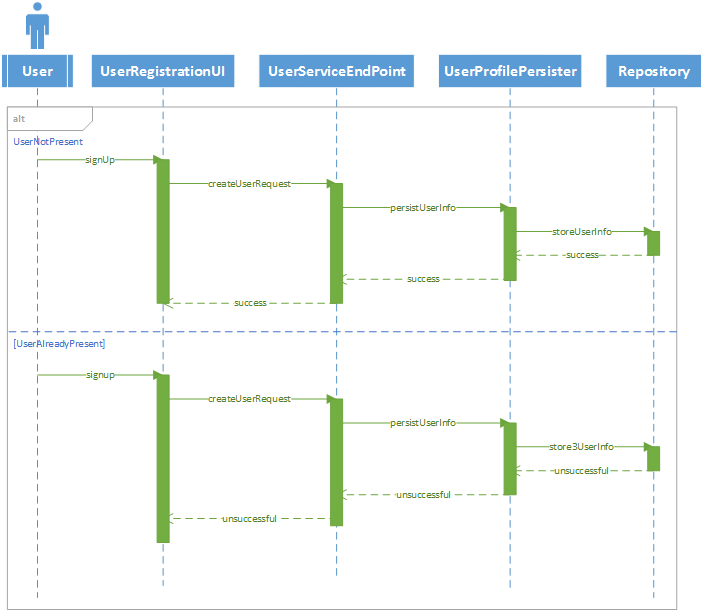
## Dynamic UML Diagrams



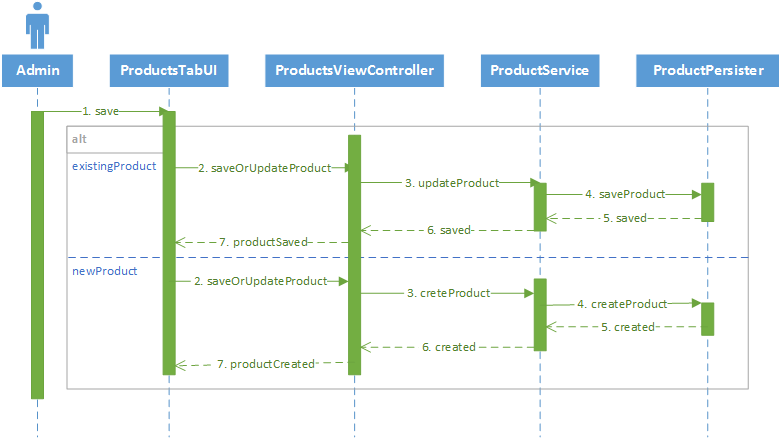
**Figure-SD#124:** Sequence diagram of Show product details



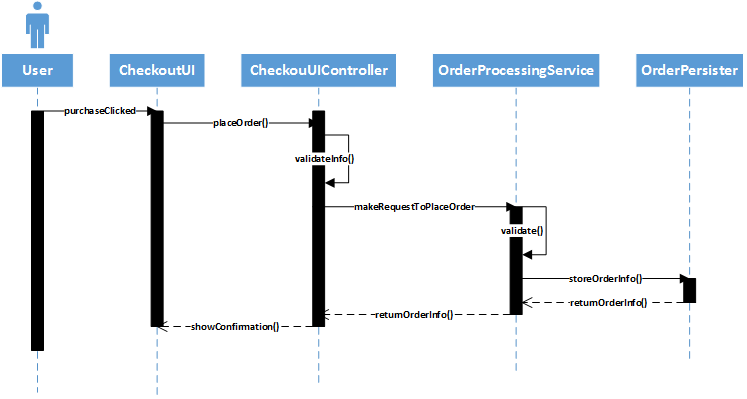
**Figure-SD#126:** Sequence diagram of User Registration



**Figure-SD#133:** Sequence diagram of CreateUser use case

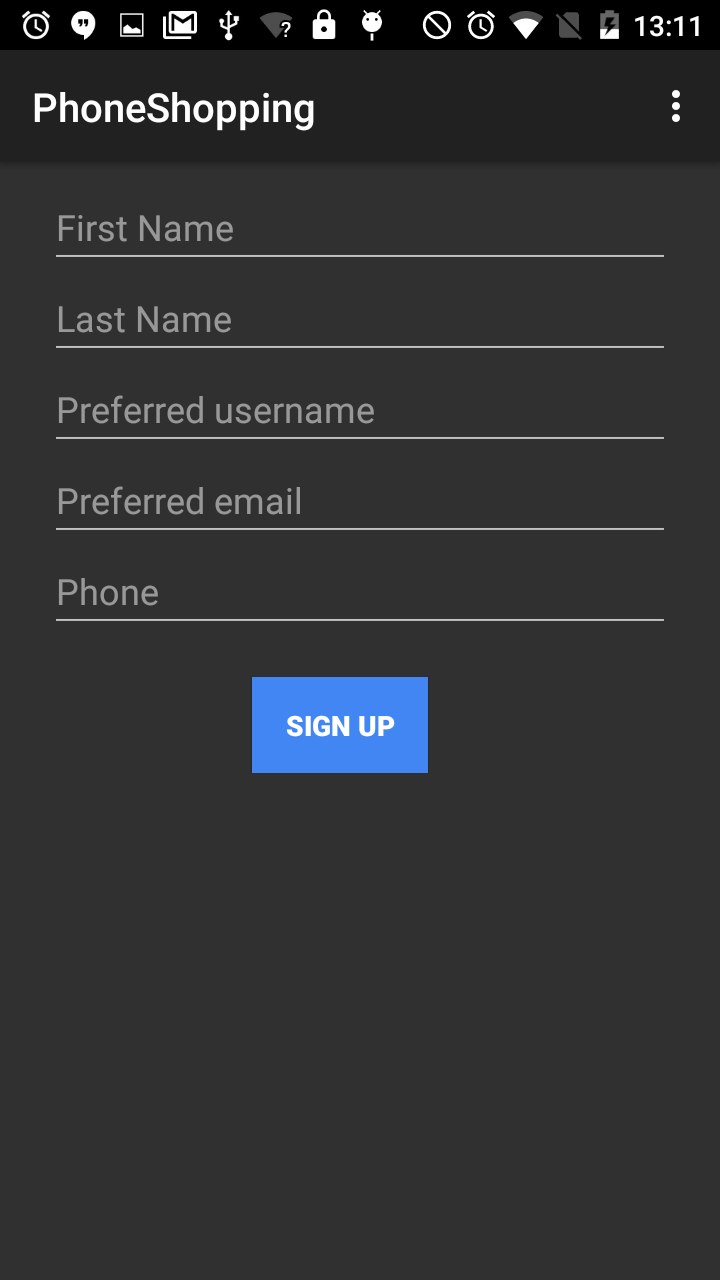


**Figure-SD#134:** Sequence diagram for Create And Update Product use cases

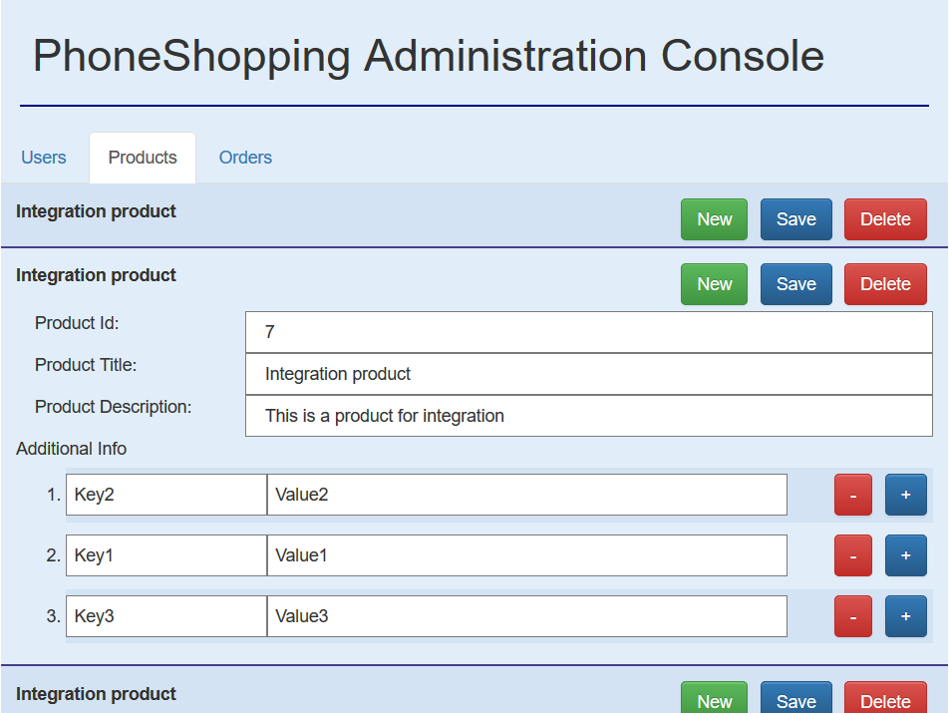


**Figure-SD#135:** Sequence diagram of PlaceOrder use case

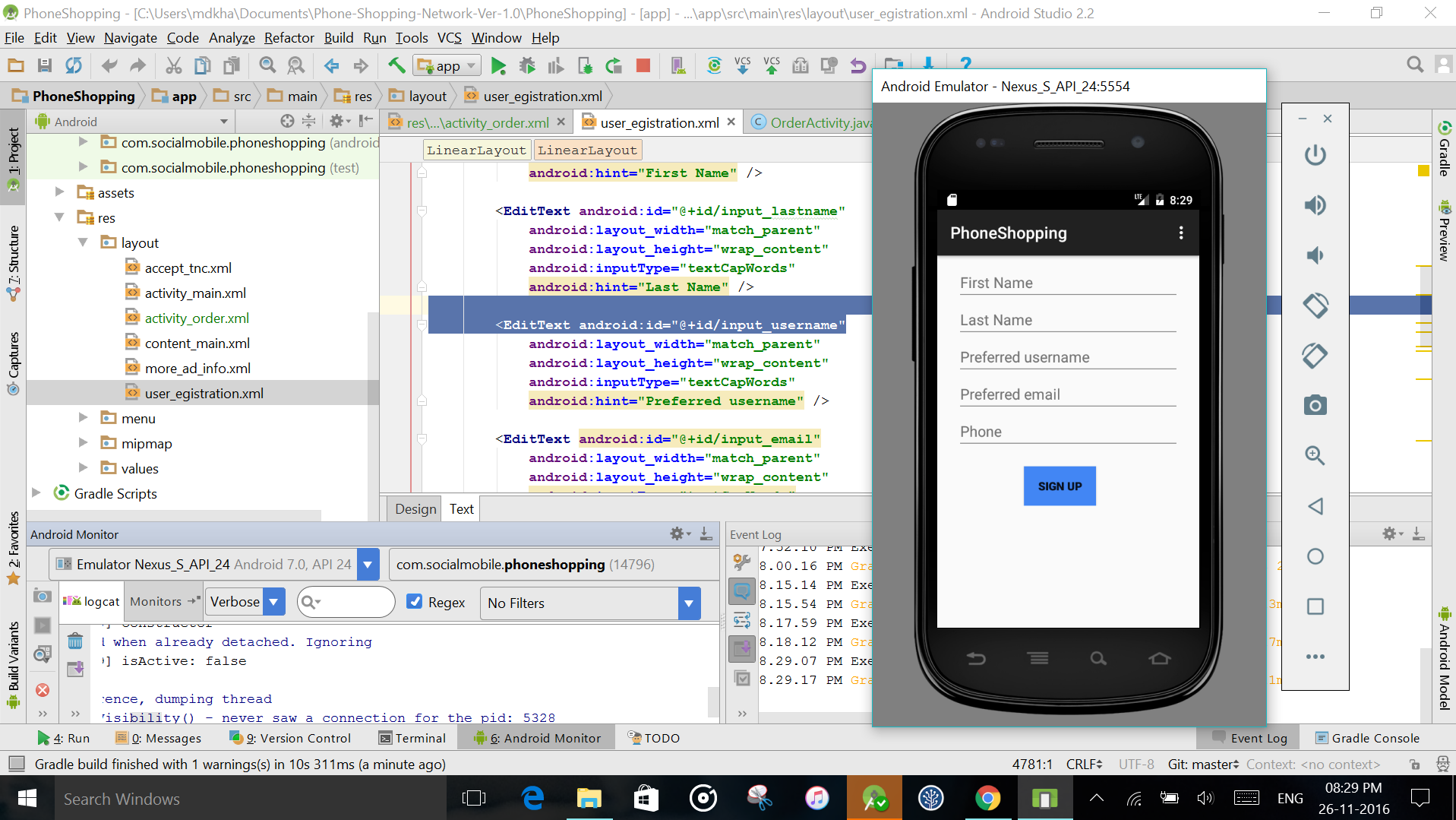
## Appendix B - User Interface Design



**Figure-Screen1:** User registration UI



**Figure-Screen2:** Product view of the management console



**Figure-Dev1:** android application development

## Appendix C - Sprint Review Reports

**Sprint 1 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 09/12/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of **User Story#125** was accepted by the product owner.

**Sprint 2 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 09/26/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of **User Story#126** was accepted by the product owner.

**Sprint 3 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 10/10/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of **User Story#124** was accepted by the product owner.

**Sprint 4 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 10/24/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of **User Story#133** was accepted by the product owner.

**Sprint 5 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 11/07/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of **User Story#134** was accepted by the product owner.

**Sprint 6 Report**

Attendees: Charles Green, Dewan Mohammad Moksedul Alam, Khaja Mohammed

Date: 11/22/2016

Start time: 5:50 pm

End time: 6:30 pm

After the explanation and demonstration, the implementation of the following features were accepted by the product owner.

* User Story#135 and
* User Story#157

# References

Spring Framework (Version 4.3.3.RELEASE). <https://projects.spring.io/spring-framework/>

Jersey-RESTful web services in Java Version 2.23.2. <https://jersey.java.net/>

Hibernate Version 5.2.3.final. <http://hibernate.org/orm/>

Commons-DBCP version 1.4. <https://commons.apache.org/proper/commons-dbcp/>

MySQL JDBC Connector Version 5.1.31. <https://www.mysql.com/products/connector/>

Jackson JSON API for java Version 2.5.4. <http://wiki.fasterxml.com/JacksonHome>

Bootstrap Version 3.3.7. <http://getbootstrap.com/>

JQuery Version 3.1.1. <https://jquery.com/>

KnockoutJS Version 3.3.0. <http://knockoutjs.com/>