Software Requirements Specification

for

NITKart

Version 1.0

Prepared by Akshay Upadhya B

15IT204

National Institute of Technology Karnataka, Surathkal

January 7, 2018

Contents

1. Introduction 3
1.1 Purpose 3
1.2 Definitions, Acronyms, and Abbreviations 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 4
1.5 References 4
2. Overall Description 5
2.1 Product Perspective 5
2.2 Product Functions 5
2.3 User Classes and Characteristics 5
2.4 Operating Environment 6
2.5 Design and Implementation Constraints 6
2.6 User Documentation 6
2.7 Assumptions and Dependencies 7
3. External Interface Requirements 8
3.1 User Interfaces 8
3.2 Hardware Interfaces 8
3.3 Software Interfaces 8
3.4 Communications Interfaces 8
4. System Features 9
4.1 User class 1 – The customer 9
4.1.1 User registration 9
4.1.2 User login 9
4.1.3 User homepage 10
4.1.4 Product search 10
4.1.5 Product Information 10
4.1.6 Purchase 10
4.1.7 Shopping Cart 11
4.2 User class 2 – The Seller 11
4.3 User class 3 – The Administrator 11

5. Other Requirements 11
5.1 Performance Requirements 12
5.2 Safety Requirements 12
5.3 Security Requirements 12
5.4 Software Quality Attributes 12
5.5 Business Rules 12
Appendix A: Glossary 13
Appendix B: Analysis Models 14
Use Case Diagrams 14
Class Diagrams 15
Activity Diagram 16
Sequence Diagram 16
Navigation Diagram 18

Revision History

Name	Date	Reason For Changes	Version
Akshay	23/01/2017	Added User Interfaces under External Interface Requirements Changed Shopping Cart Requirements	1.0
Akshay	27/02/2017	Added Appendix A Added UML Design Diagrams in Appendix B	1.0

1. Introduction

1.1 Purpose

The product is an application named NITKart, version 1.0. It is an Android based shopping application. The aim of this product is to provide online access to the goods available in the shops in NITK so that the students do not have to walk all the way to the shops and bring them back to their rooms. Depending on the seller, delivery facilities can also be arranged. Thus, saving time and increasing the reach of all goods and services provided by the sellers in NITK.

1.2 Definitions, Acronyms, and Abbreviations

Term	**Definition**
User	Someone who interacts with the application
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
System	The e-commerce system
Customer	Someone who is interested in buying a good
Store employees	People who are employed by the organization
Sellers	Stores or people who are wishing to sell their goods through our application

1.3 Intended Audience and Reading Suggestions

The intended audiences of stakeholders for this specification of the CDS include:

- Sellers
- Users, who interact with the system
- Administrators

The users here are the selected beta testing group, who test the system thoroughly, informs the development team of existing bugs, and suggests features.

Sellers can add good which are presented to the buyers through the application on approval by the administrator.

The administrator looks after the entire application. They are responsible for the contents in the application.

1.4 Product Scope

The scope of this project includes user interface software for the NITKart device and maintenance of a database containing global language information.

1.5 References

IEEE. IEEE *Std 830-1998* IEEE *Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

The system will consist of two parts: The Android application and the database. The Android application has two types of user interface: The customer UI and the seller UI.

The customer UI has a provision for login and register. After logging in, the customer can browse for goods. The customer can add up to a specific quantity of products to this. Apart from this, the customer can leave their reviews for a particular good that is visible to all the other users on the system.

The admin UI has access to all the data on the system. Admin and store managers, who have elevated access levels, can add, modify or delete data that are present in the system. The admin user can access data related to products and users. Apart from this, the admin user has access to orders from the customer and also update the order status.

Since this is a data-centric product, it will need somewhere to store the data. For that, a database will be used. The application will communicate with the database to receive data, for the customer side and the application will be able to modify the data if an admin user is accessing it. All of the database connection will go over the internet.

2.2 Product Functions

- The users will be able to search for products. The result will be based on the criteria the user inputs. There are several search criteria and there are filters to make the search results relevant for the user. The administrator can modify these filters.
- On clicking the image of a particular product, the details of the product appear. These details will be provided by the administrator, and is stored in the database.
- If the user wants to buy the product, they can click on a button, which will add the product to a virtual shopping cart unique to the user. This is private to the user.
- The user can review and rate the product, which is publicly visible and to other users.
- The user can order the product and can track the order. The admin can view the orders and update the order status.

2.3 User Classes and Characteristics

There are three types of users that interact with the system: customers, sellers and administrator. Each of these three types of users has different use of the system so each of them has their own requirements.

The customers can use the application to search, compare and purchase the goods. They do not modify the data that is displayed to rest of the users. For searching a product, a user is given various options like the price filter, number of seats etc. so that the user can search efficiently.

The sellers use the same application, but they have a different login. The updated orders are visible to the users. The seller can add or remove products with a quantity limit if necessary.

The administrator uses the same application, but they have a different login. They have all the access of sellers in addition to access to modify and add data in the application. They can adjust the filter options for the customers.

2.4 Operating Environment

The application can be run on Android devices making sure that a maximum number of devices can use the software without any problems. A server for hosting images will be used. The backend will be the Firebase backend provided by Google.

2.5 Design and Implementation Constraints

The internet connection is a constraint for the application. Since the application fetches data from the database over the internet, it is crucial that there is an Internet connection for the application to run.

The application will be constrained by the capacity of the database. Since many users use the database at a particular time, it may be forced to queue incoming requests and therefore increase in the time that it takes to fetch the data.

Since the administrator decides the content inside the application, the data must be backed up in a different location. The content must also be appropriate and the administrator must monitor all of this.

The type of the database used is upto Google as it is the backend provider. Also, the type of server will be decided after comparing the different implementable servers.

2.6 User Documentation

The user documentation will be prepared after the completion of the project. The customers may not refer to this user documentation as the UX for the customer will be simple and easy to understand. The user manual will be in a Wiki format.
TBD.

2.7 Assumptions and Dependencies

Our assumption is that the application will be used on a mobile device with a modern Android OS. Since Android is constantly evolving, use of an old, outdated OS could pose a compatibility, and more importantly, a security issue.

Also, being an internet based application, the server needs to be up all the time. Downtime in the server could mean loss of customers for the organization.

3. External Interface Requirements

3.1 User Interfaces

The first page that the user sees when the app is opened is the login page. If the user is already registered, they may proceed to login to the application. If not, there will be an option to register for new users. Once this authentication is done, the list of products available will be displayed to the user.
The next screen is the list of products. The user should be able to search and filter products. Clicking on any product takes the user to the detailed description of the product, where they can see bigger images of the product along with the description and an option to add a certain quantity to the virtual shopping cart.
Access to the virtual shopping cart is present in all available screens once logged in.
There is a checkout page under development.

3.2 Hardware Interfaces

The hardware connection to the database servers is managed by the underlying operating system (Android) that runs the application.

3.3 Software Interfaces

The only constraint as of now is that the device must be running an Android operating system.
The other constraints are TBD.

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating system on which our application runs.

4. System Features

This section includes the requirements that specify all the fundamental actions of the software system.

4.1 User class 1 – The customer

4.1.1 User registration

4.1.1.1 Description and Priority

To use the application, the user should be able to register through the application. The user must provide username, password and email address. The user can choose to provide a regularly used phone number. This has a high priority, because without this snippet, the entire application would be useless.

4.1.1.2 Stimulus/Response Sequences

On the main screen, login button is pressed. Below the login form, there will be a link to the registration page.

4.1.1.3 Functional Requirements

REQ-1: The registration page must accept name, username, password and phone number as the input.
REQ-2: The password must be verified with a confirm password field.
REQ-3: The username must be unique for each user.
REQ-4: No field must be blank.

4.1.2 User login

4.1.2.1 Description and Priority

Given that a user has registered, then the user should be able to log in to the application. The user with the help of browser password manager can save the login information. This has a high priority, because without this snippet, the entire application would be useless.

4.1.2.2 Stimulus/Response Sequences

On the main screen, login button is pressed. The login form is then shown.

4.1.2.3 Functional Requirements

REQ-1: The login form must have radio buttons to select between user, employee and administrator.

REQ-2: The login form must query for username and password.
REQ-3: In case of an error, the error is shown explicitly.
REQ-4: [TBD] In case the user forgot the password, there must be a “forgot password” link.

4.1.3 User homepage

4.1.3.1 Description and Priority

After login, the user is shown a page, which contains all the products for sale in the showroom. This has a high priority, as any sales cannot be made without this part.

4.1.3.2 Stimulus/Response Sequences

After the user logs into the application, this page is shown.

4.1.3.3 Functional Requirements

REQ-1: All products for sale must be shown on the page.
REQ-2: The goods must be arranged in an order chosen by defaults of filter.
REQ-3: A search functionality is placed on top of the page.

4.1.4 Product search

TBD

4.1.5 Product Information

4.1.5.1 Description and Priority

If the user clicks on the product, the information must appear about the product and all the specifications of the product must be present.

4.1.5.2 Stimulus/Response Sequences

In the homepage, click on the image of a product to bring this page out.

4.1.5.3 Functional Requirements

REQ-1: The information must be clearly presented.
REQ-2: If some data is missing, it should be indicated as such.
REQ-3: There must be a button to buy the products in the cart.

4.1.6 Purchase

4.1.6.1 Description and Priority

If the user wants to buy the product, they can do so by clicking a button which will add it to the virtual shopping cart, and can be checked out.

4.1.6.2 Stimulus/Response Sequences

On the information page of a product, this button will add to the shopping cart.

4.1.6.3 Functional Requirements

TBD

4.1.7 Shopping Cart

4.1.7.1 Description and Priority

The Virtual Shopping cart is analogous to the physical cart one carries around in a shop. One can add or remove items to/from it, and can checkout all items present in it.

4.1.7.2 Stimulus/Response Sequences

There will be an option to open the shopping cart from all available screens once logged in.

4.1.7.3 Functional Requirements

REQ-1: The cart should maintain consistency and integrity, i.e, it should show the same products and quantities everytime the user opens it from any screen.
REQ-2: Once checked out, the products must be removed from the cart.
REQ-3: There should be only one cart available for a user.

4.2 User class 2 – The Seller

TBD

4.3 User class 3 – The Administrator

TBD

5. Other Requirements

TBD

5.1 Performance Requirements

TBD

5.2 Safety Requirements

TBD

5.3 Security Requirements

TBD

5.4 Software Quality Attributes

AVAILABILITY: The system shall be available all the time.

CORRECTNESS: A bug free software which fulfill the correct need/requirements of the client.

MAINTAINABILITY: The ability to maintain ,modify information and update fix problems of the system

USABILITY: software can be used again and again without distortion.

ACCESSIBILITY: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.

ACCURACY: The reliability on the information/output. Can depend/be sure of the outcome.

STABILITY: The system outcome/output won't change time to time. Same output will be given always for a given input.

5.5 Business Rules

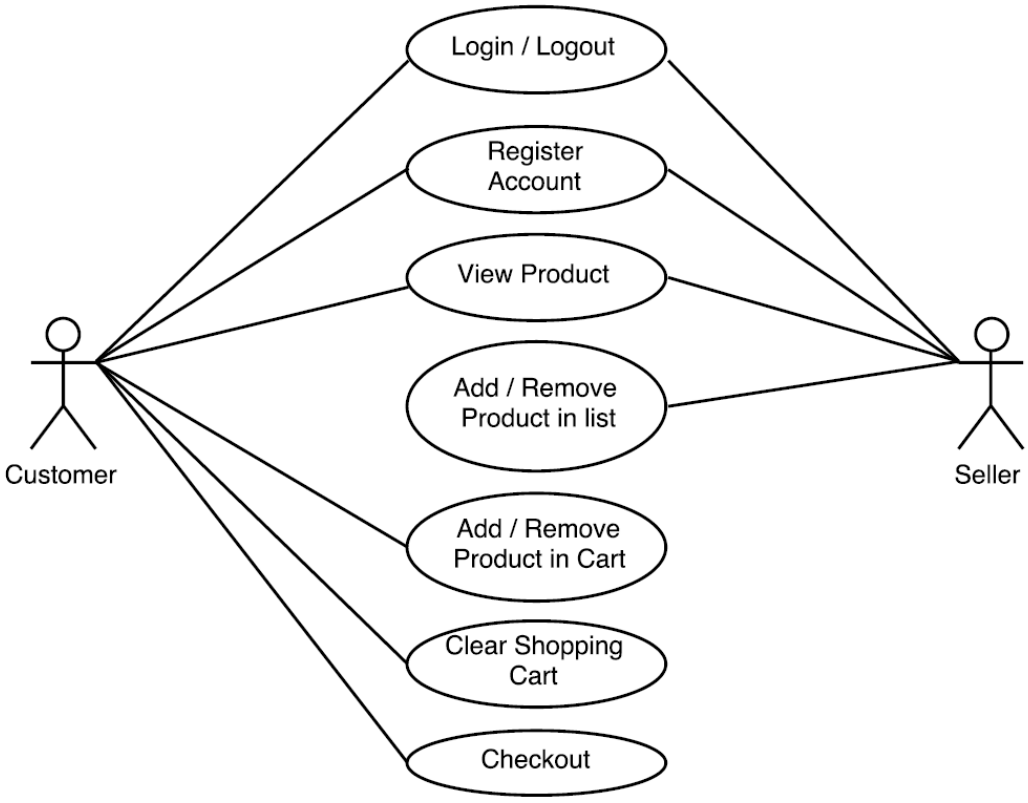
TBD

Appendix A: Glossary

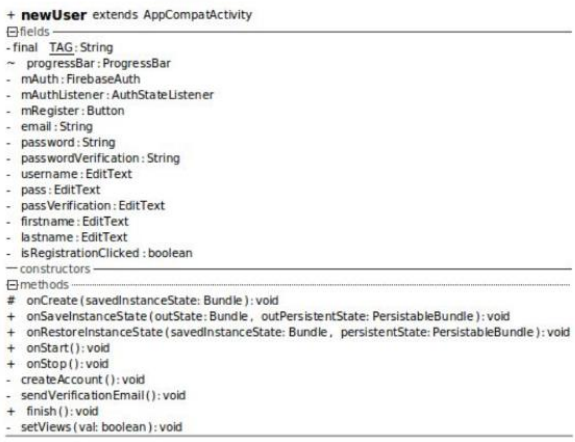
Term	Definition
User	Someone who interacts with the application
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
System	The e-commerce system
Customer	Someone who is interested in buying a good
Store employees	People who are employed by the organization
Sellers	Stores or people who are wishing to sell their goods through our application

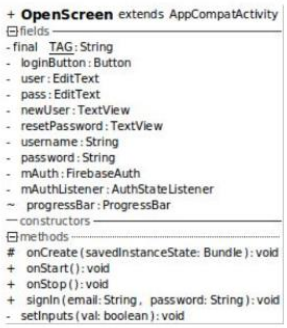
Appendix B: Analysis Models

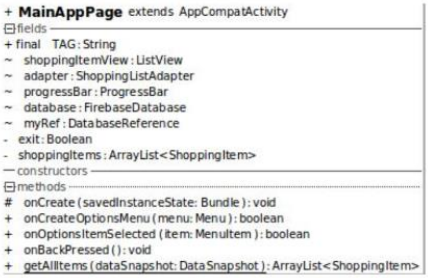
Use Case Diagrams

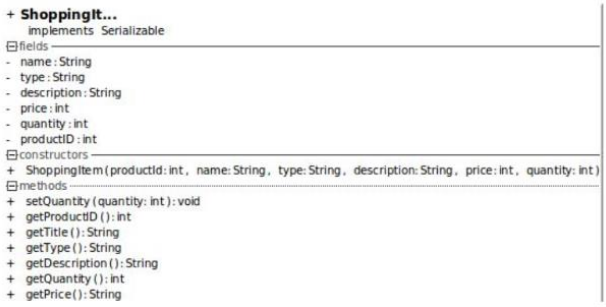


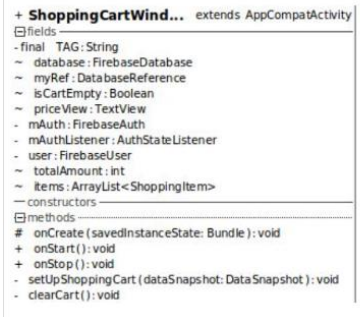
Class Diagrams



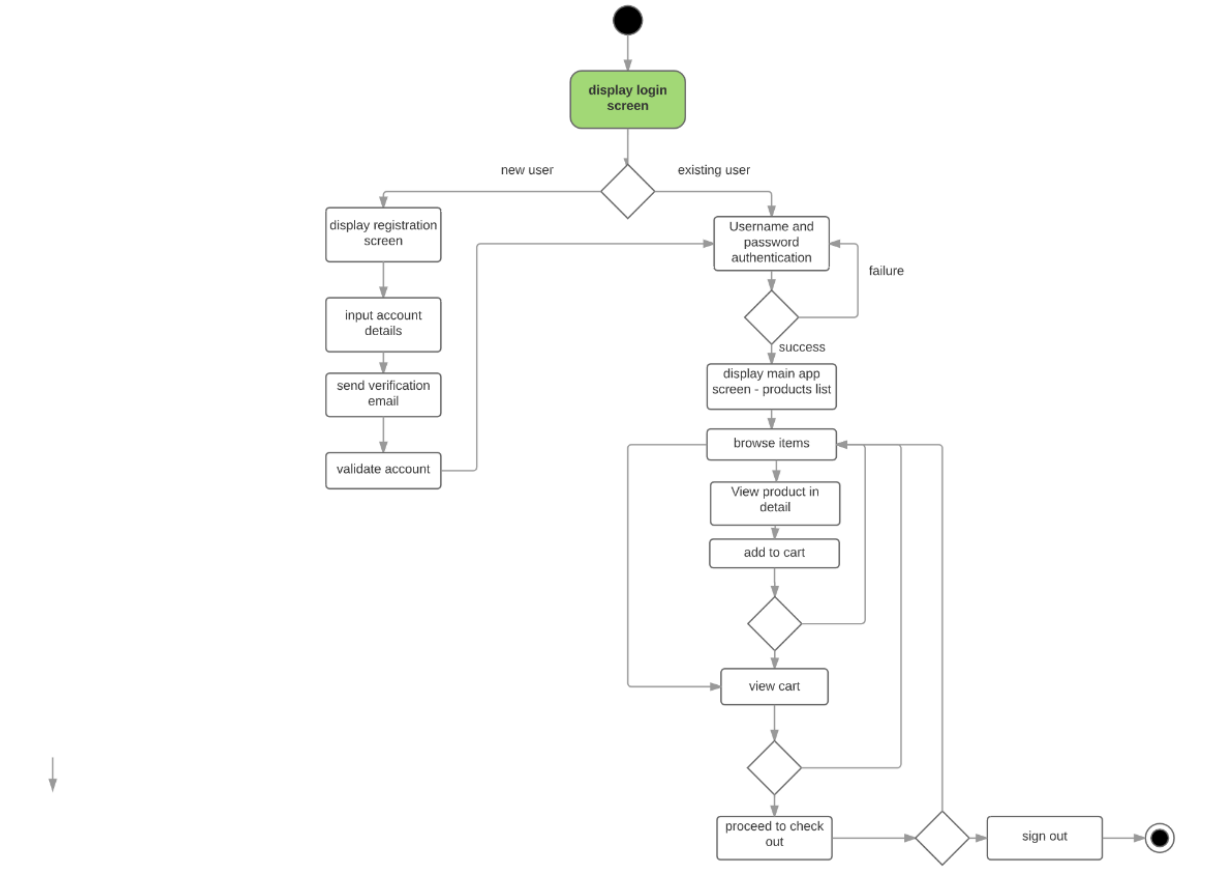






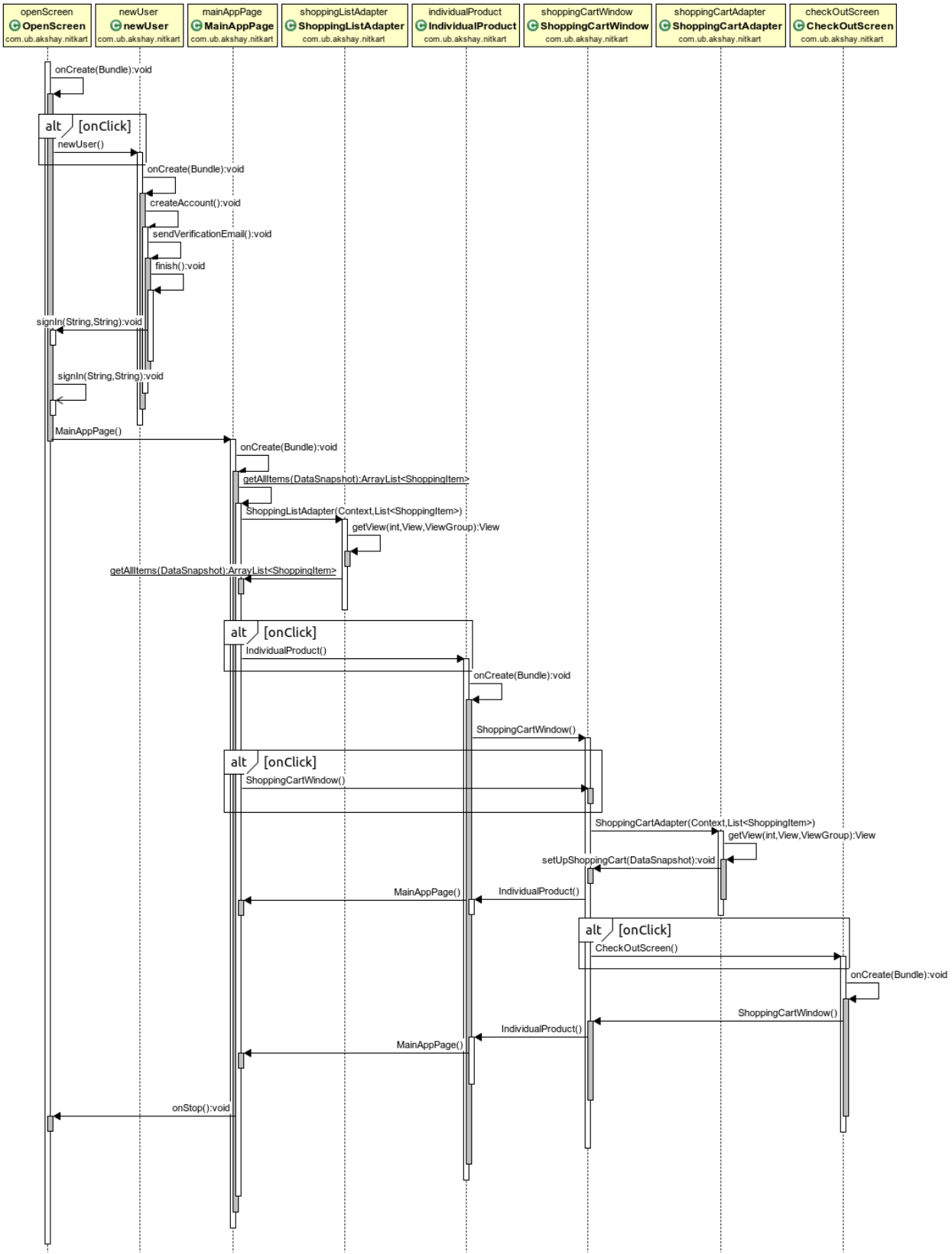


Activity Diagram



Sequence Diagram

(In the page below)


openScreen
OpenScreen
com.ub.akshay.nitkart
newUser
newUser
com.ub.akshay.nitkart
mainAppPage
MainAppPage
com.ub.akshay.nitkart
shoppingListAdapter
ShoppingListAdapter
com.ub.akshay.nitkart
individualProduct
IndividualProduct
com.ub.akshay.nitkart
shoppingCartWindow
ShoppingCartWindow
com.ub.akshay.nitkart
shoppingCartAdapter
ShoppingCartAdapter
com.ub.akshay.nitkart
checkOutScreen
CheckOutScreen
com.ub.akshay.nitkart
onCreate(Bundle):void
alt
[onClick]
newUser()
onCreate(Bundle):void
createAccount():void
sendVerificationEmail():void
finish():void
signIn(String,String):void
signIn(String,String):void
MainAppPage()
onCreate(Bundle):void
getAllItems(DataSnapshot):ArrayList<ShoppingItem>
ShoppingListAdapter(Context,List<ShoppingItem>)
getView(int,View,ViewGroup):View
getAllItems(DataSnapshot):ArrayList<ShoppingItem>
alt
[onClick]
individualProduct()
onCreate(Bundle):void
ShoppingCartWindow()
alt
[onClick]
ShoppingCartWindow()
ShoppingCartAdapter(Context,List<ShoppingItem>)
getView(int,View,ViewGroup):View
setUpShoppingCart(DataSnapshot):void
MainAppPage()
IndividualProduct()
alt
[onClick]
CheckOutScreen()
onCreate(Bundle):void
ShoppingCartWindow()
IndividualProduct()
MainAppPage()
onStop():void

Navigation Diagram