Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Purpose

Every brand, product or anything has some unique symbol that shines up them. In addition, promoting the product is the crucial role in the working of that product. There should be some kind of application, which can provide you a great collection of sample Logos and Videos and makes you abstain from all the hectic work of designing Logos and Videos.

Go Creativity is an Android application, which brings you all different types of Logos and Videos like Professional Logos, Explainer Videos, Whiteboard Videos and many more categories. The application allows customers to make a purchase from different categories. Every module in our e-commerce application has user-friendly and interactive UI.

1.2 Scope of Project

Go Creativity would provide all the best and unique Logos, Videos, etc. all at one place, giving you the best experience of online shopping. We will build, design, develop, and deliver products to users for making out their work more easy and simple. This application will give the user the best experience of online shopping in a very simple and user-friendly way.

The goal of this system is to develop an E-commerce Android application. The system will allow customers to choose and buy products from the massive selection of products in Logos, Videos and other subcategories.

The application will allow customers to shop, track or cancel you are their orders from anywhere. The app also allows customers to search for products. Customers will also have the functionality of safe and secure payment using their PayPal Bank account knowing that all transactions are securely processed. Shopping on the Go Creativity app will offer customers up to the minute updates on the status of their orders and can choose to track it at their convenience through the order tracking option. The system will also be capable of providing a good customer support by chatting within the application

The application will work as an intermediary between the customer and the seller, which will help in maintaining the quality of the product. While similar applications in the market have a direct contact between the customer and seller, which does not maintain the quality as per the customer wishes.

1.3 Glossary

Term	Definition
Customer	The user who purchases the items from the application.
Customer Support	The person who can view and responsible for managing the requirements of the customer.
Web Services	Services which provides collection and maintenance of all the items listed in the application
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Product/Item	Commodities that is being sold/given to a customer.

Table 1-1: Definitions

1.4 Acronyms and Abbreviations

Acronyms and abbreviations used throughout this report:

- Web-API: Web Application Programming Interface
- GUI: Graphical User Interface
- JSON: JavaScript Object Notation
- IDE: Integrated Development Environment

1.5 Overall Features

- ✓ Login
- ✓ Sign Up
- ✓ Browse Items
- ✓ Search Items
- ✓ User Info
- ✓ Shopping Cart
- ✓ Requirements uploading
- ✓ Online Payment
- ✓ Chat functionality
- ✓ Order Tracking
- ✓ Notifications

1.6 Technologies /Tools/Platforms

- Tool: Android Studio, JDK, JRE
- Language: Java
- Diagram Tools: Star UML

Chapter 2

ABOUT THE SYSTEM

2.1. Functional Requirements

R1: System should provide facility to create/open account.

R1.1: System provides facility to Sign Up.

I/P: User name, Password and Email Id.

O/P: Sign Up confirmation.

Error: User name exists, Invalid email id, Password does not match required conditions.

R1.2: System should provide facility to Sign In.

I/P: User name and Password.

O/P: Sign In confirmation.

Error: User name or password invalid.

R2: System should provide facility to manage profile details.

I/P: Old and new password.

O/P: Password change confirmation.

R3: System should provide facility to browse through all items.

R3.1: System should provide facility to browse through LOGOS category.

I/P: LOGO name

O/P: List of LOGOs

Error: LOGO not found

R3.2: System should provide facility to browse Videos and play selected Video item.

R3.2.1: System should provide facility to browse through Video category

I/P: Video name

O/P: List of Videos

Error: Video not found

R3.2.2: System should provide facility to play Video

I/P: Video name

O/P: Video clip with details

Error: Video not found

R4: System should provide facility to filter the items.

I/P: Filter type

O/P: Filtered list of items

Error: Item not found

R5: System should provide facility to maintain items in the cart.

R5.1: System should provide facility to add items in cart

I/P: Item name

O/P: Item addition confirmation

Error: Item not added message

R5.2: System should provide facility to remove items from the cart

I/P: Item name

O/P: Item removal confirmation

Error: Item not removed message

R6: System should provide facility to display and maintain notifications.

R6.1: System should provide facility to view all the notifications

I/P: Click Notification option.

O/P: List of notifications.

Error: Connection error

R6.2: System should provide facility to maintain notifications

I/P: Select Notification

O/P: Notification seen

Error: Notification not found

R7: System should provide facility to buy the item.

I/P: Description

O/P: Description addition confirmation

Error: No description added

R8: System should provide facility to check status of the order.

R8.1: System should provide facility to view the development stage of the order

I/P: Order number and name

O/P: Development details of the order

Error: Order number and name not found

R8.2: System should provide facility to cancel the order

I/P: Order name and number

O/P: Order cancellation confirmation

Error: Order number and name not found

R9: System should provide facility to view the purchase details.

I/P: Order History option

O/P: List of all the orders

Error: Order option not found

R10: System should provide facility to view the trending items.

I/P: Category name

O/P: List of trending items

Error: Category not found

R11: System should provide facility to maintain a chat session between user and admin.

I/P: User name

O/P: Chat session established

Error: user not found, connection error

Error: Item not found

R12: System should provide facility to sort the items

I/P: Sort type

O/P: Sorted list of items

Error: Sort type not found

R13: System should provide facility to request admin to make changes after the item get delivered

I/P: Request for change, Description of the changes

O/P: Approval confirmation, Extra charges

Error: Request not sent, connection error

R14: System should provide facility to logout from the account.

I/P: Click Logout.

O/P: Logout performed.

Error: Connection error

2.2 Non- Functional Requirements

Reliability – All the actions performed by the users are accordingly logged into the database. Users will be also prompted appropriate messages when any wrong action is made hence ensuring reliability.

Usability – The application will only be as usable as deemed useful by the end user. The interface will be made in such a manner that the amount of input from the user is minimized. This will automatically reduce human interaction, thus reducing the possibility of error. User-friendly user interface will ensure the ease of use. Its usability also lies in the fact that the interface made is extremely simple. This will make the useable by any Android users.

Maintainability – The code developed will be highly modular. This will help in mainly two things. Firstly, error handling will be made simple. This is the fact that, whenever an error occurs, the debugger will be able to pinpoint and reach the error cause with minimum effort.

Secondly, a modular code will also ensure extensibility. Any future requirements will be handled easily. Also if the module is required elsewhere it can be easily extended and made use of whatever deemed necessary.

Availability – The application should be available at all times, meaning the user can access it using an android device, only restricted by the downtime of the server on which the system runs. In the case of any failure, a proper message will be displayed.

Portability – The application is only runnable on android devices, so the end-user can have any android device.

2.3 User Interface and Human Factors

The application will have a friendly user interface. The application will only have only one role i.e. the customer. All users will have access to all screens in the application.

2.4 Hardware Consideration

The application can be installed in any android devices having version above Android Jellybean For better graphical interface it is recommended to use version Android Lollipop 5.0 or above it. The android device must have the following characteristics as an average. Hardware requirements:

Android Phone, RAM: 512 MB or greater

2.5 Software Requirements

The user must have an Android smart phone, which can run the Android application.

2.6 Performance Characteristics

The application will be well programmed so that it does not occupy a lot of memory, which will make the application to run smoothly. The end user internet connection will also play a major role in the performance of the application.

2.7 Error Handling

Input Error Handling

The application will be able to give error messages when the user enters wrong input into the fields or leave an important field empty. For example, the user can never enter a negative number in login field. If this happens then the application will give error messages regarding the valid id.

2.8 Quality Issues

This application will be able to run on any Android smart phone having the required version. The application will have error handling mechanism.

2.9 System Modifications:

The application will be able to do all the tasks that are mentioned in functional and non-functional requirements sections. Other functionality such as adding more categories, providing various payment methods can be implemented in future, for bringing more improvement to the system.

2.10 Security Issues:

The application will have user accounts for its users. The passwords stored for every user will be in an encrypted format. The application use SSL (secured socket layer) in all transactions during payment that includes any confidential customer information.

2.11 Project Planning

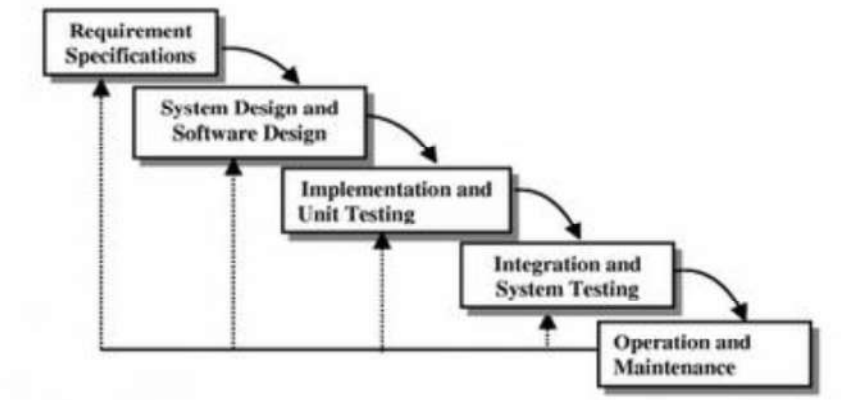
2.11.1 Project Development Model

Iterative Waterfall model will be used in this system.

Mpiric Software proposes to follow Iterative Waterfall Model for the Software Development Life Cycle of the Go Creativity android application project. Iterative development is a way of breaking down the software development of a large application into smaller chunks.

In iterative development, feature code is designed, developed and tested in repeated cycles.

With each iteration, additional features can be designed, developed and tested until there is a fully functional software application ready to be deployed.



Advantages of Iterative Waterfall model:

- In the iterative model, we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on, we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In the iterative model, we are building and improving the product step by step. Hence, we can track the defects at early stages. This avoids the downward flow of the defects.
- In the iterative model, we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In the iterative model, less time is spent on documenting and more time is given for designing.

Disadvantages of Iterative Waterfall model:

- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

2.11.2 Project Plan

- Gather the module definition
- Checking the time schedule feasibility
- Requirement gathering for module
- Analysis on gathered requirement
- Designing
- Coding
- Testing

2.11.3 Milestone and Deliverables

Feasibility analysis phase: 1 week

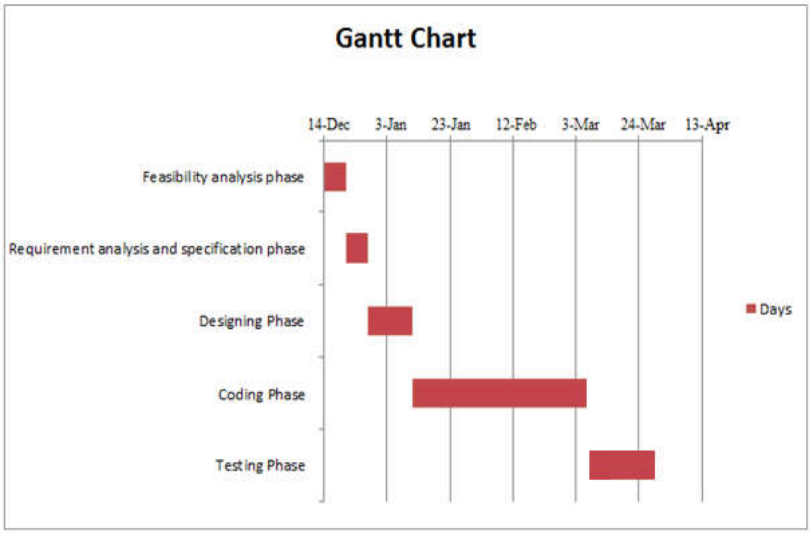
Requirement analysis and specification phase: 1 week

Designing Phase: Approximately 1 week

Coding Phase: Approximately 8 weeks

Testing Phase: Approximately 2 weeks

Gantt chart



Chapter 3

ANALYSIS

3.1 ER Diagram

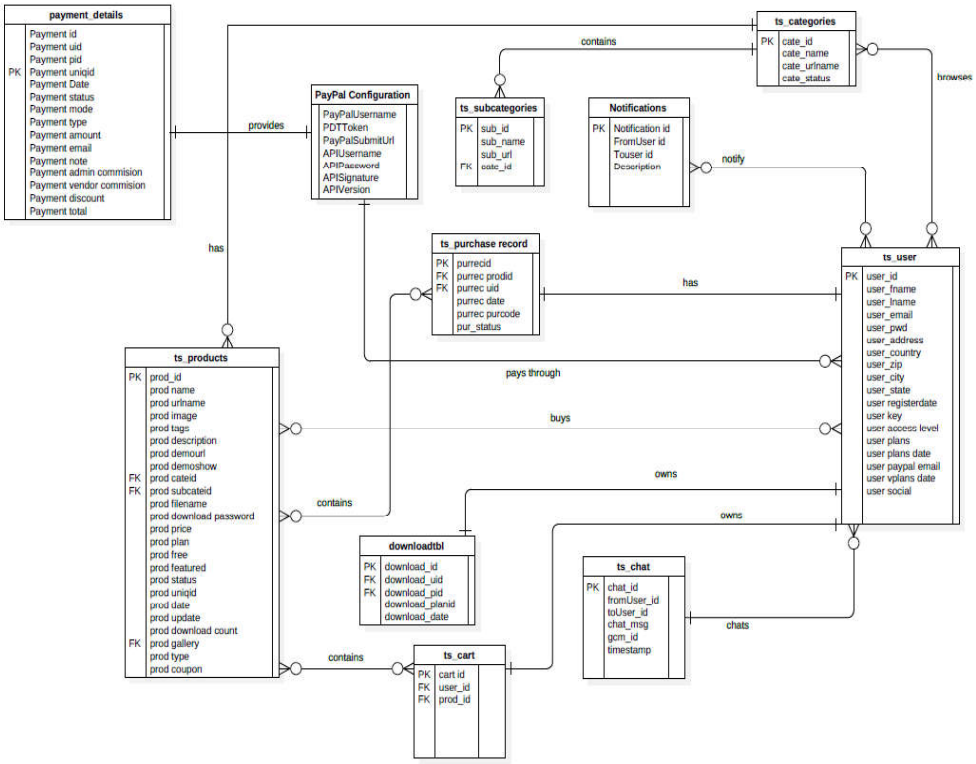


Figure 3.1: ER Diagram

3.2 UML Diagram

The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system. UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

The main purpose of drawing diagram is that user can relate himself/herself with a system for better understanding. We prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system.

For Go Creativity Android application, we have developed below UML diagrams:

- Use Case diagram
- Class diagram
- Sequence diagram
- Activity Diagram
- State Diagram

3.2.1 Use-case Diagram

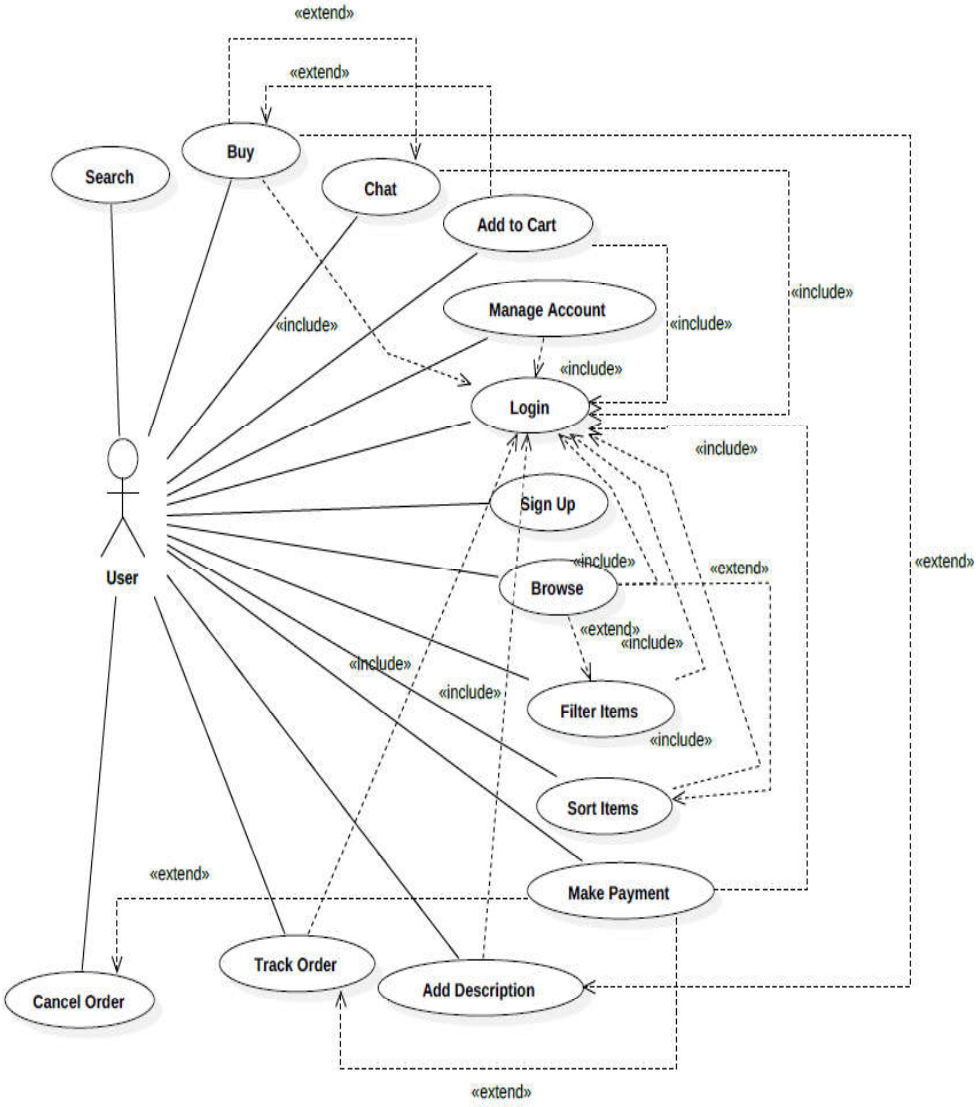


Figure 3.2: Use-case Diagram

3.2.2 Class Diagram

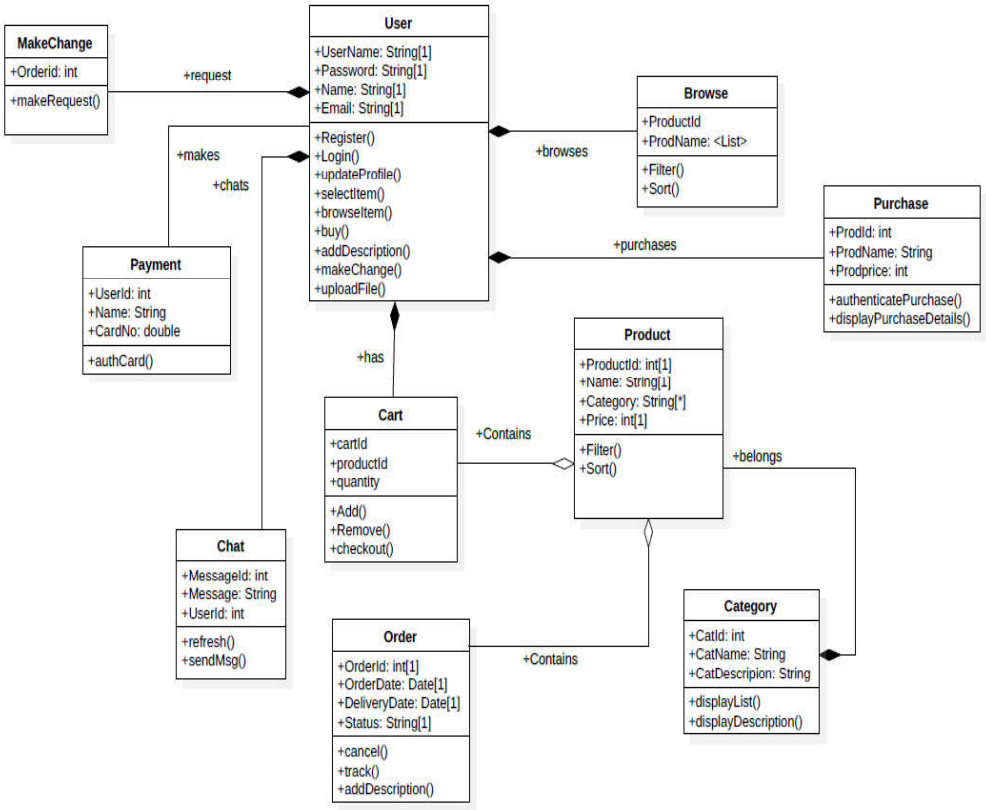


Figure 3.3: Class Diagram

3.2.3 Sequence Diagram

3.2.3.1 Sequence Diagram for Buy function

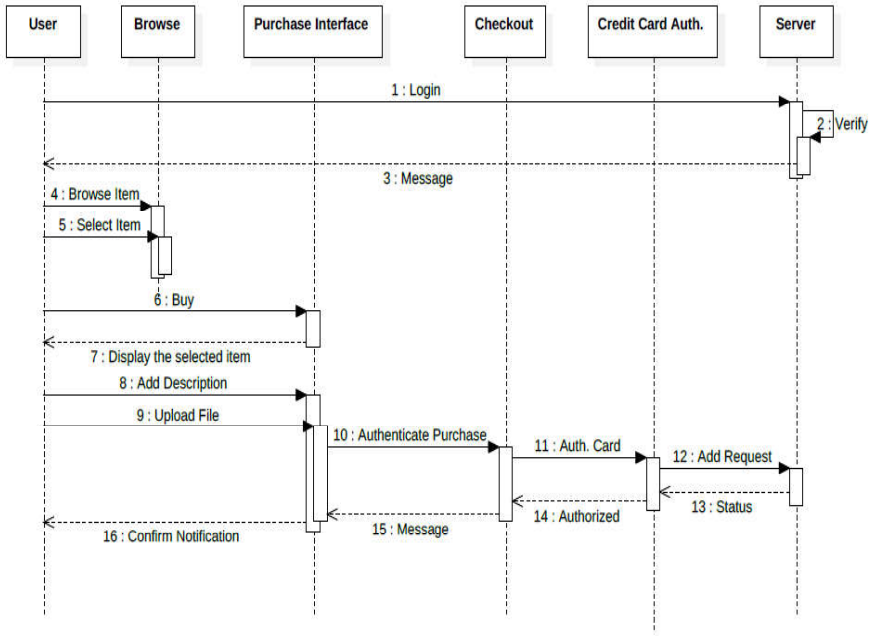


Figure 3.4: Sequence Diagram for Buy activity

3.2.3.2 Sequence Diagram for Request Change function

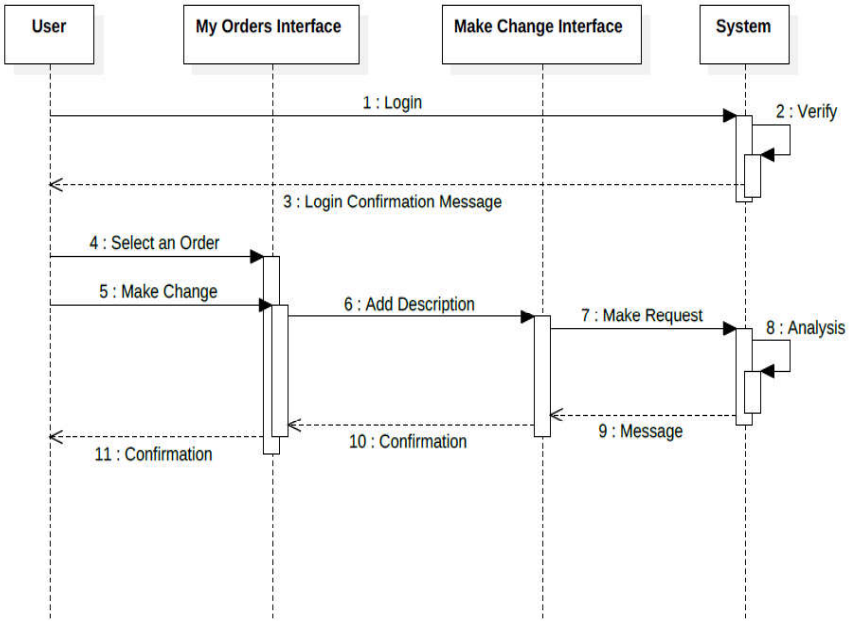


Figure 3.5: Sequence Diagram for Request Change

3.2.4 Activity Diagrams

3.2.4.1 Activity Diagram for Go Creativity

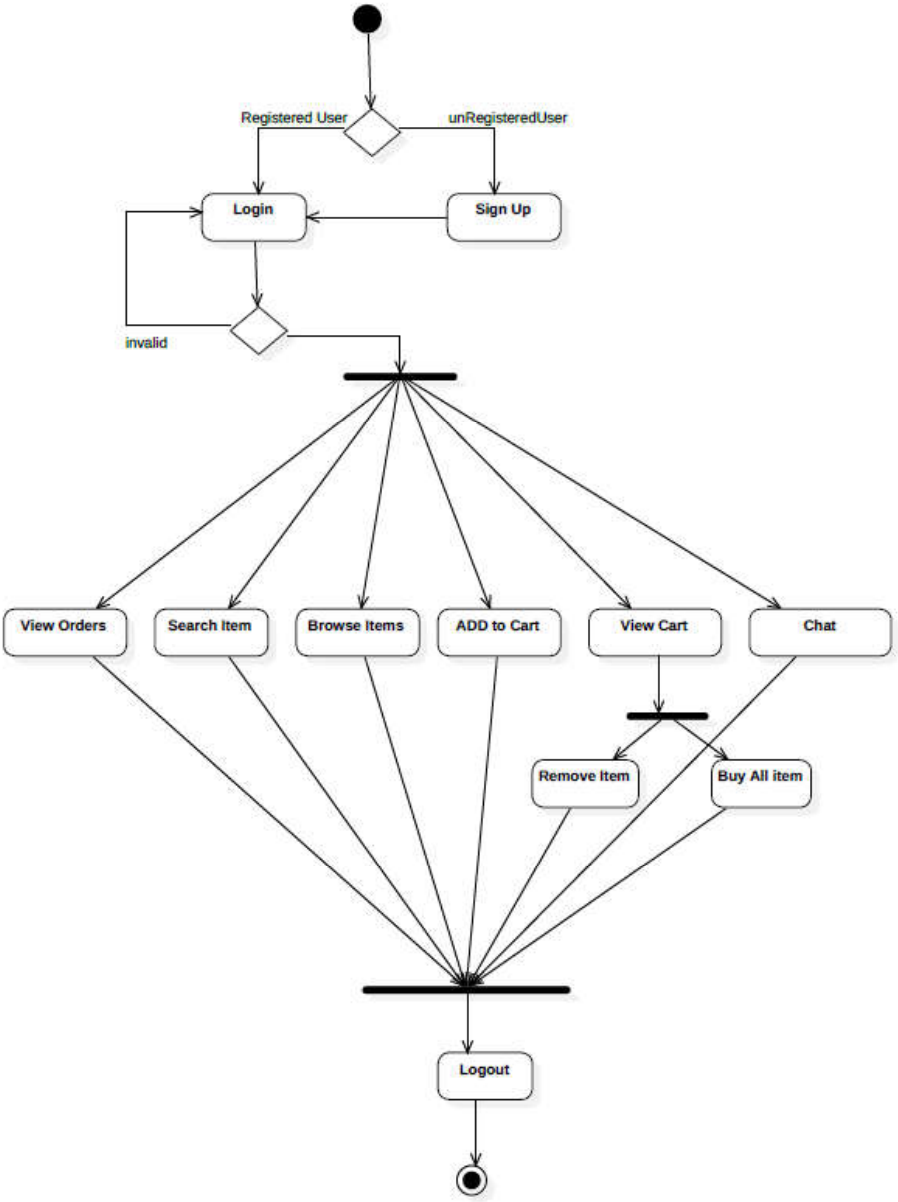


Figure 3.6: Activity Diagram for Go Creativity

3.2.4.2 Activity Diagram for Buy function

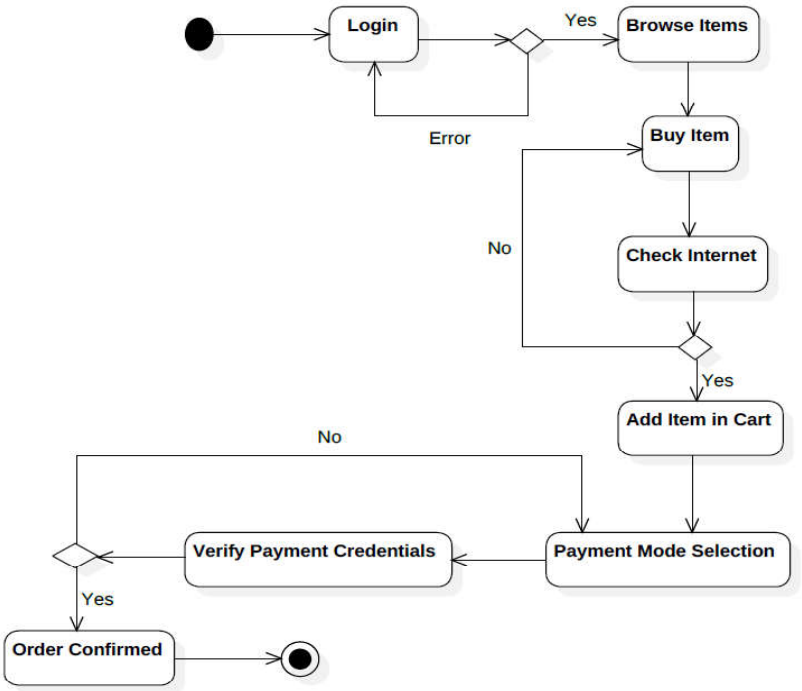


Figure 3.7: Activity Diagram for Buy activity

3.2.4.3 Activity Diagram for Request Change function

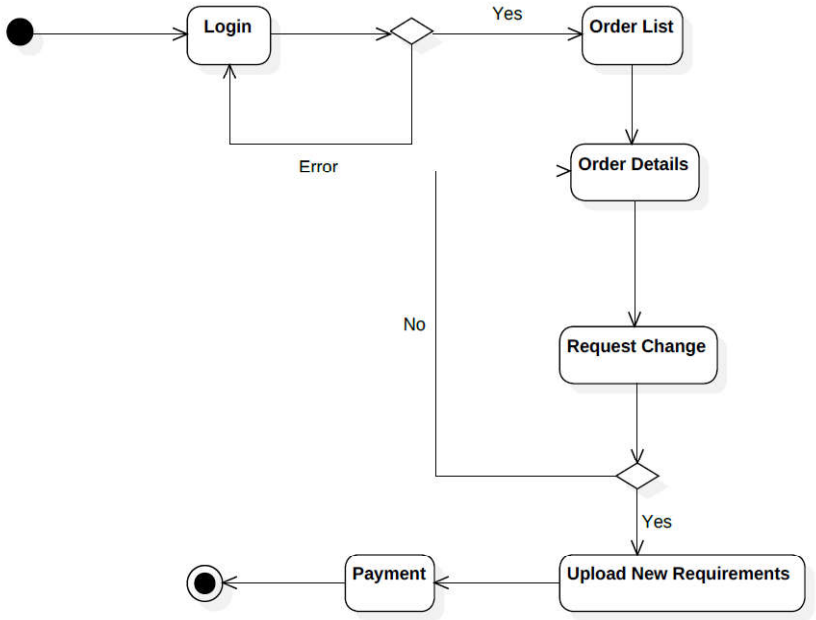


Figure 3.8: Activity Diagram for Request Change

3.2.5 State Diagram

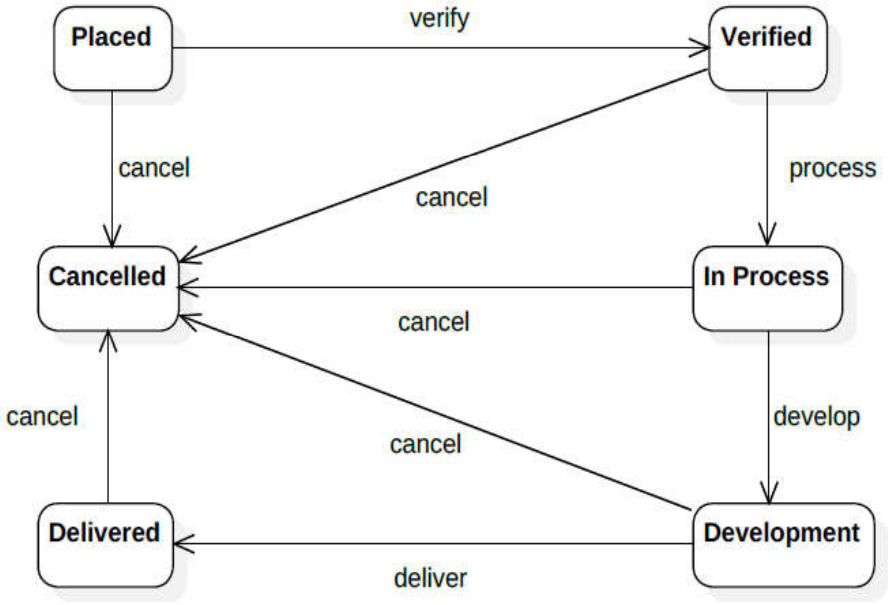


Figure 3.9: State Diagram

Chapter 4

DESIGN

4.1 Database

Categories_table:

Columns	Datatype	Field Size	Constraint
cate_id	int	11	Primary Key
cate_name	varchar	500	Null
cate_urlname	varchar	500	Null
cate_status	tinyint	4	Null

Table 4-1: Categories Table

Paypal configuration:

Columns	Datatype	Field Size	Constraint
PayPalUsername	varchar	150	Primary Key
PDTToken	varchar	250	Null
PayPalSubmitUrl	varchar	250	Null
APIUsername	varchar	250	Null
APIPassword	varchar	150	Null
APISignature	varchar	250	Null
APIVersion	varchar	50	Null

Table 4-2: Paypal Configuration Table

Downloads:

Columns	Datatype	Field Size	Constraint
download_id	int	11	Primary Key
download_uid	int	11	Foreign Key
download_pid	int	11	Foreign Key
download_planid	int	11	Null
download_date	timestamp		Null

Table 4-3: Downloads Table

Notifications:

Columns	Datatype	Field Size	Constraint
NotificationID	int	11	Primary Key
FromUserID	int	11	Null
ToUserID	int	11	Null
Description	varchar	250	Null

Table 4-4: Notifications Table

Chat_table:

Columns	Datatype	Field Size	Constraint
id	int	11	Primary Key
fromUserId	int	11	Null
toUserId	int	11	Null
chat_msg	varchar	250	Null
gcm_id	int	11	Null
timestamp	timestamp	20	Null

Table 4-5: Chat Table

User:

Columns	Datatype	Field Size	Constraint
user_id	int	11	Primary Key
user_uname	varchar	250	Null
user_fname	varchar	250	Null
user_lname	varchar	250	Null
user_email	varchar	250	Null
user_pwd	text		Null
user_mobile	varchar	250	Null
user_address	text		Null
user_country	int	11	Null
user_state	varchar	250	Null
user_city	varchar	250	Null
user_zip	varchar	250	Null
user_registerdate	timestamp		Null
user_status	int	11	Null
user_key	varchar	250	Null
user_accesslevel	int	11	Null
user_plans	int	11	Null
user_plansdate	varchar	250	Null
user_paypalemail	varchar	500	Null
user_vplans	int	11	Null
user_vplansdate	varchar	250	Null
user_social	varchar	250	Null

Table 4-6: User Table

Payment Details:

Columns	Datatype	Field Size	Constraint
payment_id	int	11	Primary Key
payment_uid	int	11	Null
payment_pid	varchar	250	Null
payment_uniqid	varchar	100	Null
payment_date	varchar	250	Null
payment_status	varchar	50	Null
payment_mode	varchar	50	Null
payment_type	varchar	50	Null
payment_amount	varchar	50	Null
payment_email	varchar	250	Null
payment_note	text		Null
payment_admin_commission	varchar	50	Null
payment_vendor_commission	varchar	50	Null
payment_discount	varchar	250	Null
payment_total	varchar	250	Null

Table 4-7: Payment Details Table

Subcategories:

Columns	Datatype	Field Size	Constraint
sub_id	int	11	Primary Key
sub_name	varchar	250	Null
sub_urlname	varchar	250	Null
cate_id	int	11	Foreign Key

Table 4-8: Sub Categories Table

Products:

Columns	Datatype	Field Size	Constraint
prod_id	int	11	Primary Key
prod_name	text		Null
prod_urlname	text		Null
prod_image	varchar	100	Null
prod_tags	text		Null
prod_description	longtext		Null
prod_demourl	text		Null
prod_demoshow	tinyint	4	Null
prod_cateid	int	11	Foreign Key
prod_subcateid	int	11	Foreign Key
prod_filename	varchar	250	Null
prod_downloadpassword	varchar	500	Null
prod_price	varchar	100	Null
prod_plan	varchar	50	Null
prod_free	tinyint	4	Null
prod_featured	tinyint	4	Null
prod_status	tinyint	4	Null
prod_uniqid	varchar	100	Null
prod_date	timestamp		Null
prod_update	timestamp		Null
prod_download_count	bigint	20	Null
prod_gallery	tinyint	4	Foreign Key
prod_uid	int	11	Null
prod_type	varchar	50	Null
prod_coupon	varchar	250	Null

Table 4-9: Products Table

Purchase record:

Columns	Datatype	Field Size	Constraint
purrec_id	int	11	Primary Key
purrec_prodid	int	11	Null
purrec_uid	int	11	Null
purrec_date	varchar	100	Null
prod_file_url	varchar	200	Null
pur_status	varchar	100	Null

Table 4-10: Purchase Record Table

4.2 Front End Interface

Splash Screen

Login Screen

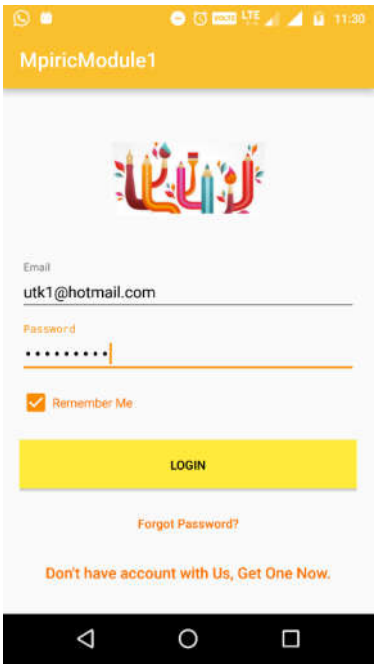
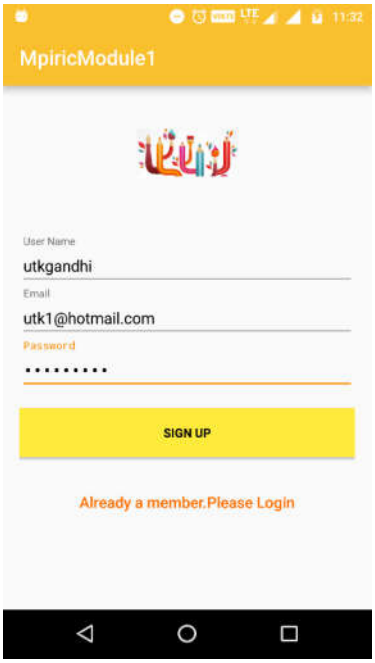


Figure 4.1: Splash & Login Screen

Sign Up Screen



Forgot Password Screen

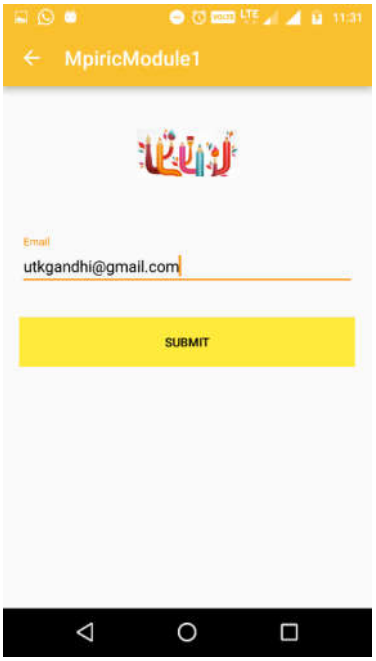
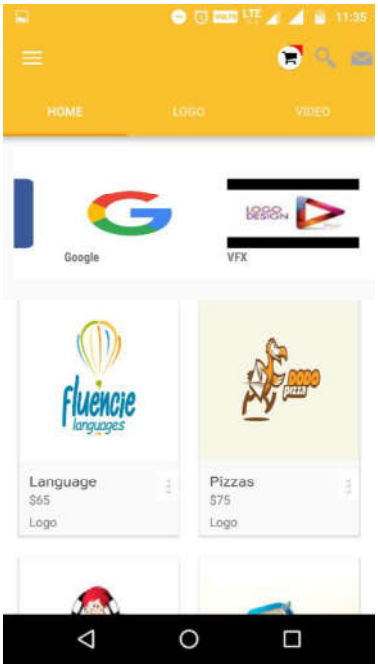


Figure 4.2: Sign Up & Forgot Password Screen

Home Screen



Logo Activity Screen

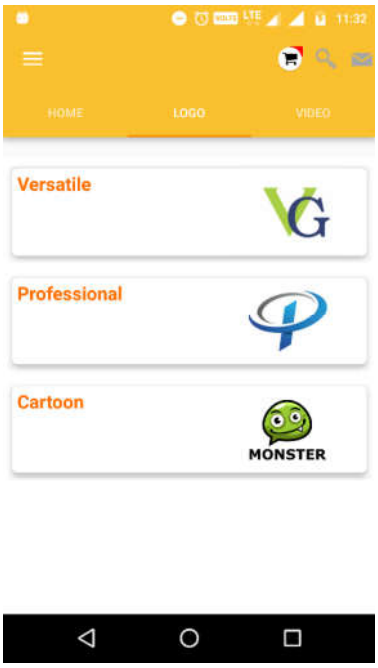


Figure 4.3: Home & Logo Activity Screen

Video Screen

Navigation drawer

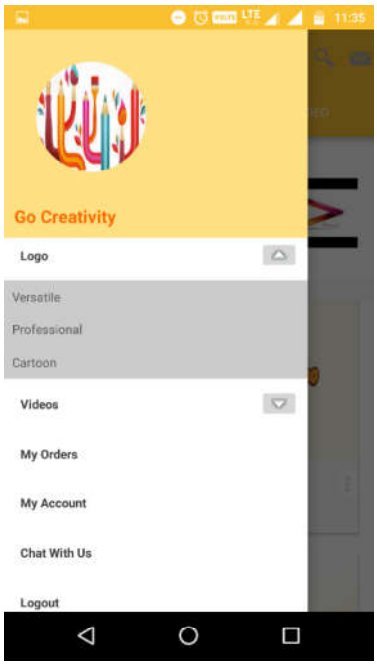
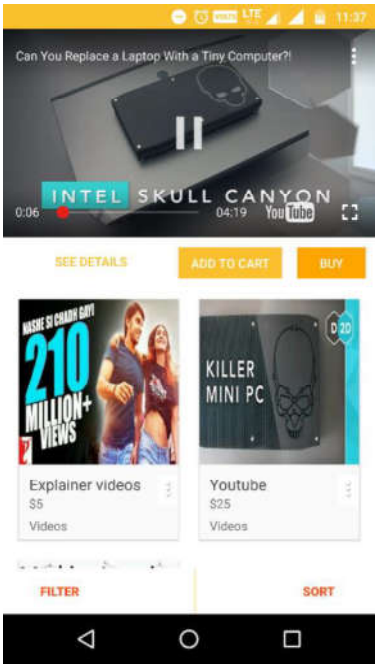


Figure 4.4: Video Screen & Navigation Drawer

Video Items Screen



Buy Item Screen

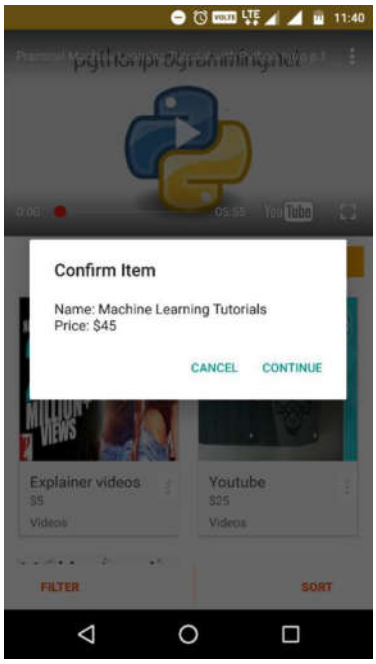
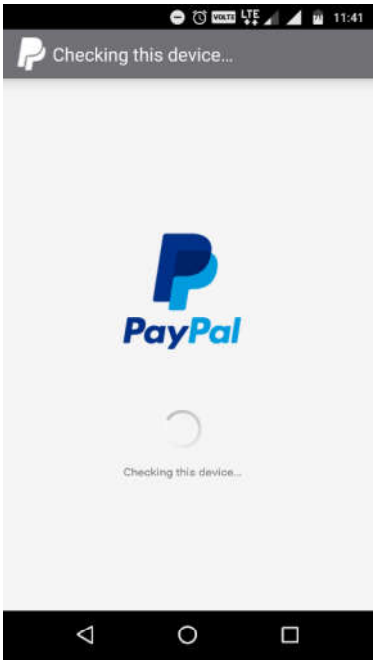


Figure 4.5: Video Items & Buy Item Screen

Payment Screen



PayPal Interface

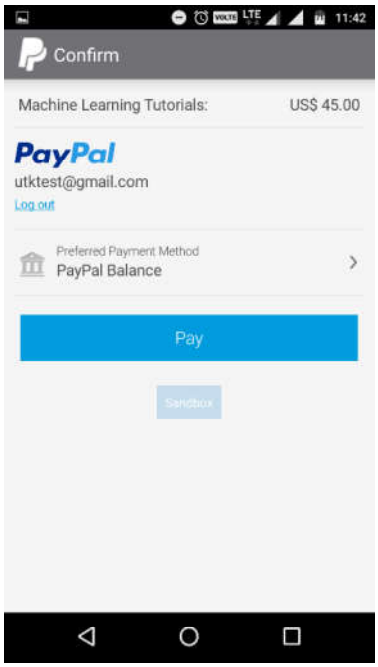
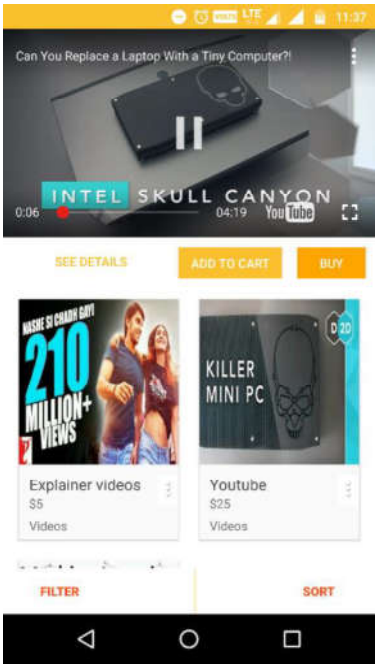


Figure 4.6: Payment Screen & PayPal Interface

Video Items Screen



Buy Item Screen

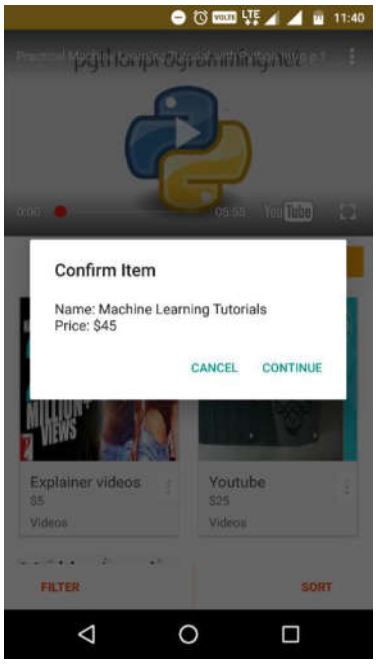
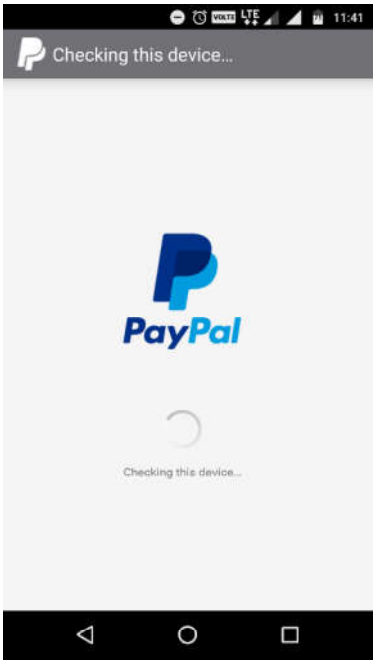


Figure 4.7: Video Items & Buy Item Screen

Payment Screen



PayPal Interface

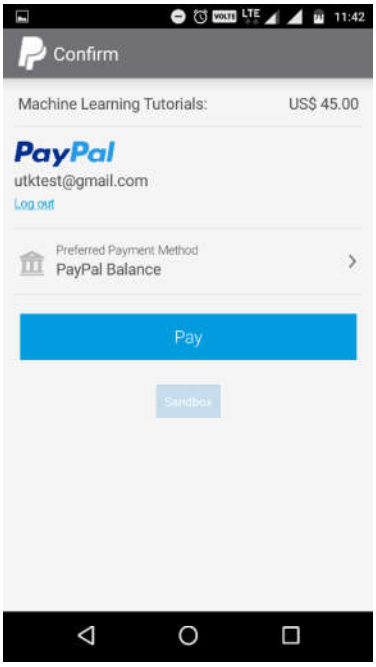
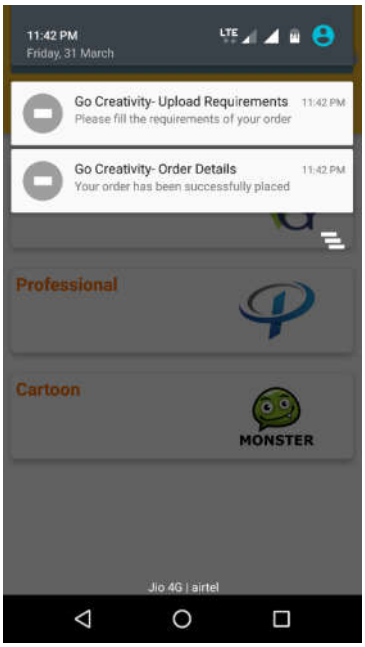


Figure 4.8: Payment Screen & PayPal Interface

Notifications



Sort Items

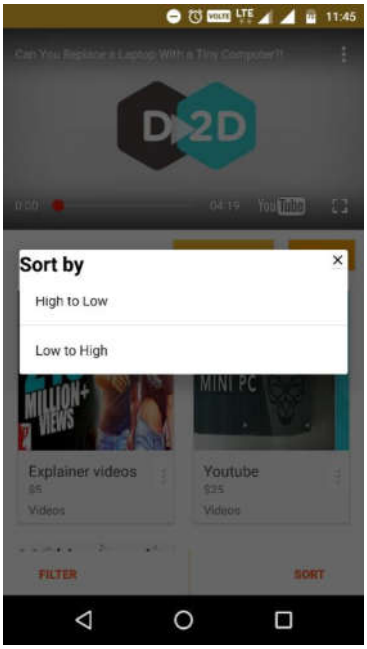
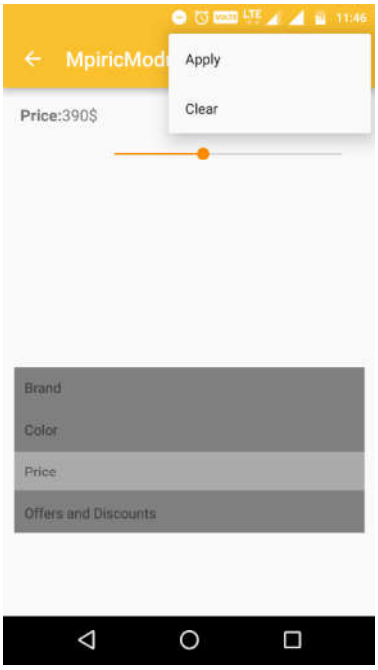


Figure 4.9: Notifications & Sort Items

Filter Items



Cart Screen

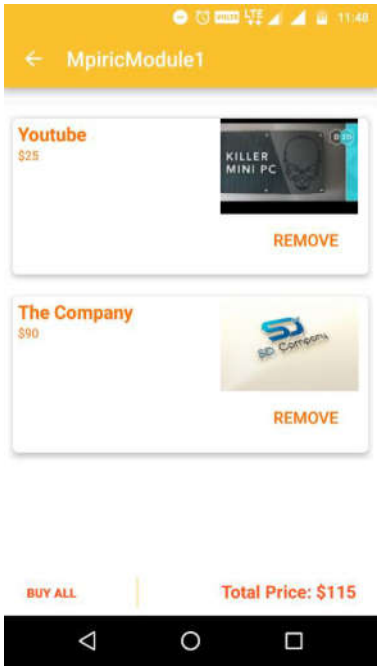
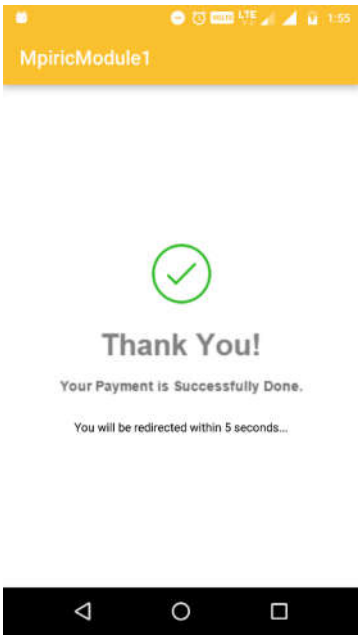


Figure 4.10: Filter Items & Cart Screen

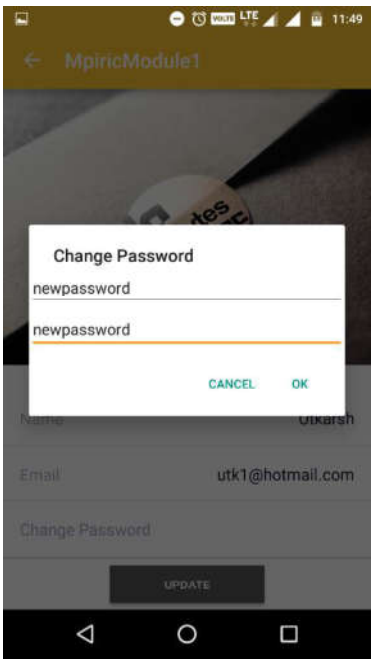
Confirmation of Payment



My Orders Screen

Figure 4.11: Payment Confirmation & My Orders Screen

Change Password Screen



Chat Screen

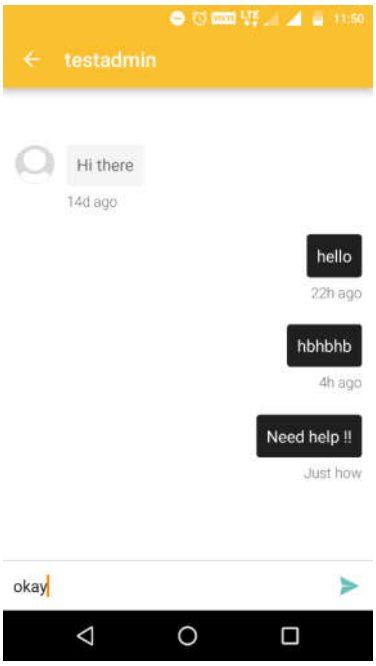


Figure 4.12: Change Password & Chat Screen

4.3 Application Navigation

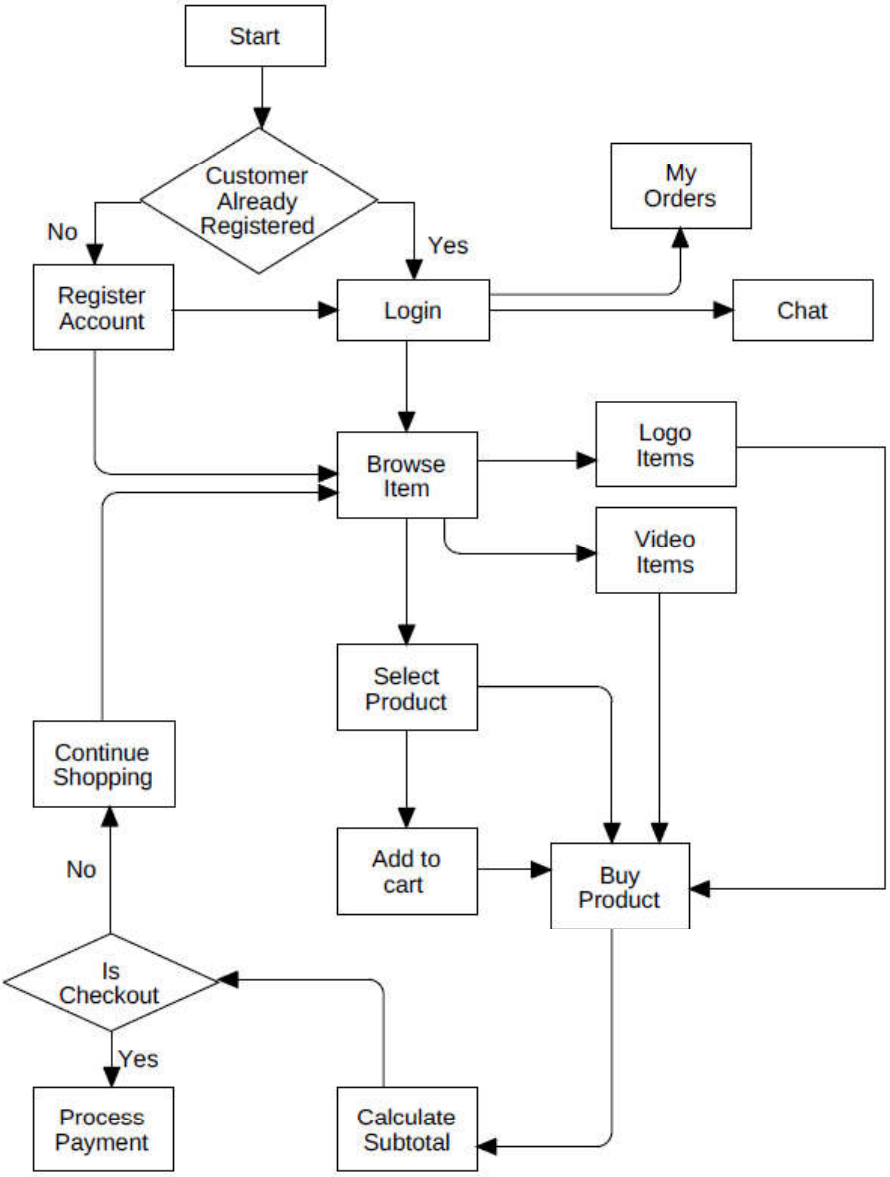


Figure 4.13: Application Flow

Chapter 5

IMPLEMENTATION

5.1 Implementation Details

Software Interfaces:

End User	Android Operating System
Data	Web Services in PHP and MySQL
Programming Language:	Java, PHP, XML
Development Software:	Android Studio

Table 5-1: Table of Software Interfaces

5.2 Performance Requirements

Performance requirements define acceptable response times for system functionality.

The load time for user interface screens shall take no longer than three seconds.

The Login information shall be verified within five seconds.

Web-services shall return results within five seconds.

5.3 Implementation of Modules

- **Add/Remove Item in Cart Module:**

  Abstract: A shopping Cart for every user is maintained in the application, which will allow user to add or remove the item in cart. User can add or remove as many items he/she wants.

  Input Parameter: Item id, Item name, Item price, Item thumbnail

  Output Parameter: Item listed in the cart or removed from the cart depending upon the operation performed.

  Processing Logic: Whenever the user adds/removes an item in the cart, the item id is passed along with URL to the API at the server side and a record is inserted/deleted into the Cart table of database.

  So now whenever the user opens his/her cart, an API is called which returns the data from the database, which contains the item id, name and price in the JSON format. Now the data is parsed and displayed in the Cart of that user.

- **Buy Module:**

  Abstract: The buy functionality is the core feature of an e-commerce application, which allows user to purchase any item displayed in the application.

  Input Parameter: Item id, Item name, Item price, Item thumbnail

  Output Parameter: Item purchased confirmation after the payment is done.

  Processing Logic: Whenever the user buys an item, firstly that item is added to the cart of the user. Now the user is redirected to the Payment activity which contains the PayPal payment interface, which will also displays the price of the item that is being purchased.

  Here the whole payment procedure is done through the PayPal and the API is maintained within the application itself. Once the payment is successfully completed, an API is called which maintains the record of the payment in the database. The user also gets notified through notifications when the payment is done successfully.

- **Chat Module:**

  Abstract: The chat feature allows user to chat directly with the contact support and can solve any kind of doubts.

  Input Parameter: User id, User name

  Output Parameter: List of messages delivered and sent

  Processing Logic: Whenever the user opens the chat interface for the first time, he/she is welcomed with a by default message which is received from the server side. Whenever the user registers for the first time in the application, the user also gets registered automatically for the chat with the Contact support person.

  The chat feature is implemented through GCM. A GCM id is maintained for every user. Therefore, whenever user sends any message the API is called which will put a record of the message in the database.

- **My Orders Module:**

  Abstract: The My Orders activity will display all the orders till now user has placed and the orders which has been delivered. Each order in the list will display the status of the order and the expected delivery date. The user can also cancel and download the order.

  Input Parameter: Item id, User id, Item price, Item name

  Output Parameter: List of all the orders till now.

  Processing Logic: Whenever the user opens the My Orders activity from the side navigation drawer, a JSON format data is received from the API. The data contains the details of all the orders made by the user at that point of time. The data is then parsed into the string format and displayed in the activity.

  The status of the order is displayed using a background service which fetches the data from database is also shown for every order. User can also download the order once the order is completed.

- **Search Module:**

  Abstract: User can search any kind of items by providing a proper input

  Input Parameter: Item name or Item type

  Output Parameter: List of the products related to the search

  Processing Logic: Whenever the user clicks on the search icon, a search widgets gets activate and then user can type the item he/she wants or just the type or tag related to the item.

  When the search text is entered, an API is called which returns the result based on the search text. The results of the search is displayed in the new activity.

5.4 Deployment of Application

In Deployment of this application during the developing time it need to build and run this application, Application runs on emulator of android studio or either I can run on android phone devices.

Application generates .apk file to run this application in devices. This .apk file available in app\build\outputs\apk\app-debug.apk. When we connect the android phone to the computer, the phone availability is visible in run windows with specific device name. If any android device is available then it shows. And from there select device and in appropriate device application is run.

Chapter 6

TESTING

Testing

Testing is the process carried out on software to detect the differences between its behaviour and desired behaviour as stipulated by the requirements specification. Testing is advantageous in several ways. Firstly, the defects found help in the process of making the software reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the software is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

6.1 Testing Plan

The testing sub process includes the following activities in a phase dependent Manner:

a) Create Test Plans.

b) Create Test Specifications.

c) Review Test Plans and Test Specifications.

d) Conduct tests according to the Test Specifications, and log the defects.

e) Fix defects, if any.

f) When defects are fixed continue from activity

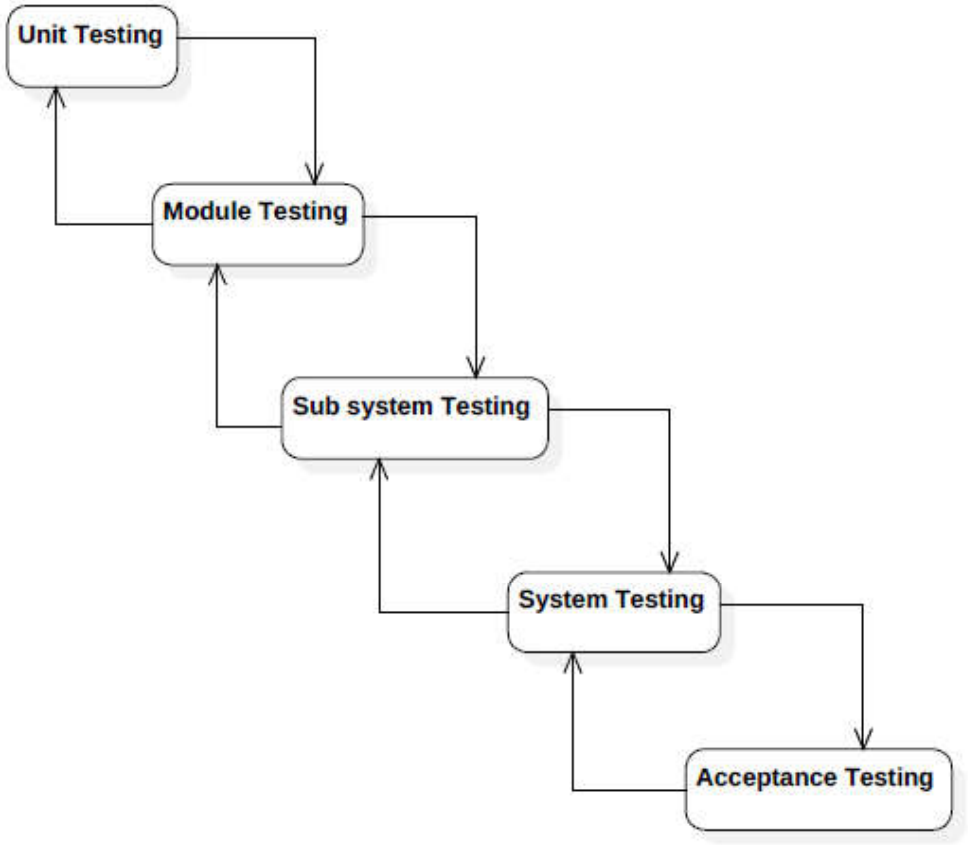


Figure 6.1: Testing Plan

6.2 Testing Strategy

The development process repeats this testing sub process a number of times for the following phases.

a) Unit Testing.

b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors. A brief description of both is given below.

Unit Testing

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

Testing Process

- Checking for availability of Code Walkthrough reports, which have documented the existence of and conformance to coding standards.
- Reviews of Unit Test Specifications verify the Unit Test Specifications conform to the program specifications. Verify that all boundary and null data conditions are included.

6.3 Testing Methods

- **Blackbox and Whitebox Testing**

  In blackbox testing a software item is viewed as a black box, without knowledge of its internal structure or behaviour. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

  In whitebox testing, knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence, more defects than blackbox testing are likely to be found.

  The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing. For this purpose, tools or compilers that can insert test probes into the programs can be used.

- **Code Coverage**

  The way to make sure that you have got all the control flow covered is to cover all the paths in the program during the testing (via whitebox testing). This implies that both branches are exercised for an 'if's statement, all branches are exercised for a case statement, the loop is taken once or multiple times as well as ignored for a while statement and all components of complicated logical expressions are exercised. This is called Path Testing. Branch Testing reports whether entire Boolean expression tested in control structures evaluated to both true and false.

  Additionally it includes coverage of switch statement cases, exception handlers and interrupts handlers. Path testing includes branch testing as it considers all possible combination of individual branch conditions. A simpler version is Statement Testing, which determines if each statement in the program has been executed at least once.

  The coverage via Path Testing includes the coverage via Statement Testing. Since Path Testing is extremely comprehensive it is costly, hence a viable minimum should be measuring Statement Testing coverage.

6.4 Test Cases

6.4.1 Test – Case for Login Activity:

Test Case ID	Test Case	Excepted Result	Test Result
1	Enter valid name and password & click on login button	Application should display home Menu	As expected
2	Enter invalid username and valid password &click on login button	Application should not display home window and Error Message Please enter valid username and password	As expected
3	Enter valid username and invalid password &click on login button	Application should not display home window and Error Message Please enter valid username and password	As expected
4	Enter invalid username and invalid password &click on login button	Application should not display home window and Error Message Please enter valid username and password	As expected
5	Without entering login information click on login button	Application should not display home window and Error Message Please enter username and password	As expected

Table 6-1: Test – Case for Login Activity

6.4.2 Test – Case for Registration Activity:

Test Case ID	Test Case	Excepted Result	Test Result
1	Enter Valid Username, Email and Valid Password conforming to a regular expression	Application should display successful Registration Confirmation.	As expected
2.	Enter Valid Username, invalid Email and invalid password	Application should display error notification for incorrect password	As expected
3.	Enter no Valid Username, email and password leaving all fields blank and press registration button	Application should display error notification for incorrect password	As expected

Table 6-2: Test – Case for Registration Activity

6.4.3 Test – Case for Filter:

Test Case ID	Test Case	Excepted Result	Test Result
1	Enter valid parameter for threshold price and press apply button	Application should display products which are less than or equal to the threshold price	As expected
2	Enter no parameter for categories or subcategories and press apply button	Application should display all items as no parameters for filter are provided	As expected

Table 6-3: Test – Case for Filter

6.4.4 Test – Case for Forgot Password:

Test Case ID	Test Case	Excepted Result	Test Result
1	Enter valid email-id to send password reset notification and press submit button	System should display an email notification and provide a link for password reset	As expected
2.	Enter invalid email-id to send password reset notification and press submit button	System would show no email notification	As expected

Table 6-4: Test – Case for Forgot Password

6.4.5 Test – Case for Buy Item:

Test Case ID	Test Case	Excepted Result	Test Result
1	User clicks on an Item and clicks on buy button.	Item should get Added to Cart and display notification of item details and pricing.	As expected
2.	User clicks on buy button and has network problem.	Item gets Added to Cart and a message of network error is displayed. Payment gateway does not open.	As expected

Table 6-5: Test – Case for Buy Item

6.4.6 Test – Case for Payment Gateway:

Test Case ID	Test Case	Excepted Result	Test Result
1	User buys and item and clicks on pay button.	User gets a message of total amount to be payed and a confirmation message is sent for the payment.	As expected
2.	User has insufficient funds and buys an item.	User gets directed to Payment Gateway but gets an error message of having insufficient funds	As expected
3.	User has network issues and buys an item and clicks on pay button.	Error message of unsuccessful transaction is displayed and user is redirected to Cart page.	As expected

Table 6-6: Test – Case for Payment Gateway

6.4.7 Test – Case for Search:

Test Case ID	Test Case	Excepted Result	Test Result
1	User Clicks on search icon and enters appropriate text.	A list of items related to the search text is displayed.	As expected
2.	User clicks on search icon and enters irrelevant text.	No items displayed in the activity.	As expected

Table 6-7: Test – Case for Search

6.4.8 Test – Case for Sort:

Test Case ID	Test Case	Excepted Result	Test Result
1	User clicks on sort button and select sort by Low to High by price.	All the items will be displayed in ascending manner.	As expected
2.	User clicks on sort button and select sort by High to Low by price.	All the items will be displayed in descending manner.	As expected

Table 6-8: Test – Case for Sort

6.4.9 Test – Case for Logout:

Test Case ID	Test Case	Excepted Result	Test Result
1	User Clicks on Logout Button	User gets logged out and redirected to Login Page	As expected

Table 6-9: Test – Case for Logout

Chapter 7

CONCLUSION & FUTURE EXTENSIONS

7.1 Conclusion

We conclude that this application works successfully with very user-friendly interface. This android application is accessible from anywhere and anytime. The applications contains all the basic features of an e-commerce shopping system like adding and removing item from the cart, buying any item, list of all the orders, chatting with the support system, etc.

The core concept of this application is to provide user different samples of Logos and Videos. So that the user can buy the item by uploading his/her requirements. The item will be developed by keeping in mind all the requirements of the user and will be tested by the person of the management system before it is delivered to the user. Hence, the quality of the item desired by the user will be achieved.

7.2 Future Extensions

Below are the future extensions that can be added into the application for making it more diverse and user-friendly to everyone.

- More categories and subcategories can be added in the application
- User can pay for item through Debit and Credit card
- User can also share the item through different suitable applications.
- User can send also send images or videos to the support system in the chat window
- User can also apply different coupon codes and avail discounts while purchasing the item
- The application can also be efficiently while low internet connection speed

Bibliography

Website referenced

- github.com
- developer.android.com
- stackoverflow.com
- www.tutorialspoints.com
- www.javatpoint.com
- www.androidhive.info