ABSTRACT

We aim to provide a framework of interactive user application in Android with its real time implementation in demo version of an virtual BigBazaar . The project does so by using Google's C2DM service (Cloud to Device Messaging it is available in Android 2.2 & above), Application server (Authenticated by Google server with unique Auth ID) maintained by the shopping mall manager and application installed on device. The application registers itself to get updates from application server at C2DM server. Using our application, the user can browse floor plans, run various searches, get live updates from retailers, rate the product, get offers and other important data. The beauty to the application is added by using data mining to show user the events or products that user would like to checkout. Currently, few applications have been made in this context like ShopDroid, Shop Savvy, etc. When the application is installed Service starts in the device which receives the pushed notifications from application server. The application uses pushing concept for sending the notification hence battery life and the performance of the device is not affected (or degraded).On receiving notification the device can send request for the new data from application server. In our implemented design, the entire database comes with application (SQLite) and runtime information, the registration ID (that is generated by C2DM server) of application are stored on application server. The design addresses the problems in mall management of communicating updates and to the target audience, provide very interactive and easy to use application.

Keywords: Android, SQLite, C2DM, ShopDroid.

INDEX

Chapter Title **Page No.**

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 BazaarDroid

The basic idea is to build a system which aids and improves the shopping experience of users. Many a times while shopping alone, the shopper ends up being confused about his selection and often wishes to seek advice from other shoppers. Also, the shopper spends much time searching the entire retail outlet for some particular color or design or brand.

Project provide framework that provides information, enables customers to interact with each other and shopping mall owners to connect with customers and a whole lot more. The framework finds its use in various places like Events, Conferences, Shopping malls, Hospitals, Museums and many more. The idea is implemented as demo for Big Bazaar, BazaarDroid is an Android application that makes shopping in an shopping mall a lot more convenient and fun. It enables customers to rate any product, provide instant access to product details, floor plans, live updates about the offers in mall.

1.2 Need of the project

Humans need advice on almost every issue, so as to enhance their decision making ability. Have you ever wondered it would be great if the updates regarding offers are sent to you directly whenever it is available? What if we could have the opinion of others on the products we are planning to purchase? Our system uses such a communication system which allows its user to connect to anyone, anywhere by any existing means.

In current scenario where time is the critical resource, everyone seeks to achieve the best in easiest and shortest way. In order to save both time and energy, shoppers would greatly appreciate a system that will show them the entire collection of a retail outlet at one go. Keeping this in mind we come up with the system called BazaarDroid for the shopping world. BazaarDroid would assist the shopper to make a best buy in less than usual time he/she spend on shopping. Also it would assist the retailers to target a large untapped market and hence boast up their sales. Hence a win-win situation for both shoppers and retailers is created.

1.3 Literature survey

BazaarDroid is a system working on two tier architecture..

- Application server- Eclipse Galileo is installed with latest version of apache tomcat server plugin as an Application server for our application. Application server is responsible to send updates and send user the information required to access the products.
- Client:

  Android Phone: At device side we need an android device with Android 2.2 or above installed in it. The android devices need to have Semi-Active connection with the application Server, and continuous connection with Google C2DM Cloud.
- Google Server.(C2DM & Client Login)

  The application server is an white listed google login account with an Auth Code provided to it. The client needs to have an registration ID from the google server for getting updates from the respective application server.

2 PROBLEM DEFINITION AND SCOPE

2.1 Problem Definition

To develop a platform that will enable shoppers to communicate in real time with people at distant locations while shopping and seek their advice regarding their purchases. The platform should also aid the user in making shopping a quick and smart activity.

2.2 Scope

We intend to develop an application for

a. Getting updates hot offers, sharing comments lists etc.
b. Showing the entire product range of the shopping mall

This would enhance the decision making process of the user and enrich the whole shopping experience.

Three basic two emerge out of the project, Device application, the store side application server:

Store-side application Server

- Store database in XML format of different sections and products.
- Contact list – Maintains a list of contacts.

Device-side application (Android Phone)

- Logistics: Access important information about the event such as Product information, Floor Plan, Contact information and more.
- Floor Plans: Convenient access to floor plans.
- Search: Provide various search options like search by Category.
- Live Updates: Stay on top of what's going on. Get live updates from Mall.

- Rating: Get relevant feedback from Client. Conduct live Rating of products.
- Add the product to favorite list.
- Use data mining concept to show project once the product is added to favorite list.

3 SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

3.1.1 Purpose

BazaarDroid is a system which focuses on enhancing the customer experience of shoppers at a shopping mall which hence also profit the retailers. It would achieve this by

a. Enabling shoppers at a shopping mall or at their workplace search various products. This system would thus enable real time reviews or comments on the product.
b. Showing the entire range of products available there while standing at one point.

3.1.2 Intended Audience

This document is intended for developers, end-users, marketing staff and documentation writers.

3.1.3 Scope

We intend to develop a system for shopping world which allows shoppers to:

a. Share reviews with their friends or all other shoppers shopping in the mall.
b. Showing the entire product range of the mall.
c. Send updates to the Android Phone.

This would enhance the decision making process of the user and enrich the whole shopping experience.

Store-side application Server

- Store database in XML format of different sections and products.

Contact list – Maintains a list of contacts.

Device-side application (Android Phone)

- Logistics: Access important information about the event such as Product information, Floor Plan, Contact information and more.
- Floor Plans: Convenient access to floor plans.
- Search: Provide various search options like search by Category.
- Live Updates: Stay on top of what's going on. Get live updates from Mall.
- Comments: Get relevant feedback from Client. Conduct live Rating of products.
- Add the product to favorite list.
- Use data mining concept to show project once the product is added to favorite list.

3.1.4 References

[1] Hello, Android: Introducing Google's Mobile Development Platform by Ed Burnette.
[2] Android in Action, Second Edition W. Frank Ableson, Robi Sen, and Chris King.
[3] Android Wireless Application Development **By:** Shane Conder; Lauren Darcey.
[4] http://developers.android.com
[5] http://code.google.com/android/c2dm/index.html
[6] http://www.vogella.de/articles/Android/article.html
[7] http://developer.android.com/guide/topics/ui/index.html

3.2 Overall description

3.2.1 System Perspective

This system arises out of the need of shoppers to get feedback and suggestions about what to buy and what not from their friends and relatives. This indirectly enhances the decision making ability of the users and enriches the whole shopping experience. This also leads to more productivity and higher revenues for the malls. This system allows users to share images and get them rated. Also the need to shop for the correct products without having to search the entire store has to be addressed.

3.2.2 User Classes and Characteristics

This system would be accessible to a large section of users who visit the mall. The most important clientele for the malls would be their loyalty based customers. This system is likely to be used by users with the following demographic profile:

- Computer literates
- 18 – 45 yrs of age

They would also be the ones who are likely to harness most of the features of this application. The system in order to be used by people of other age groups who are less acquainted with computer should be very easy to handle and use.

3.3 Usage scenario

This section provides a usage scenario for the software. It is the organized information collected during requirements elicitation into use-cases.

1. User: The user is the Android phone user.

2. App: The application in itself is an actor which interacts Application server.

3.3.2 Use-cases:

1. *Start App*
2. *Load Basic Data*
3. *Enter Map*
4. *Browse Floor Plan*
5. *Select any section on Map*
6. *Show list of products*
7. *Select an item*
8. *Browse user list*
9. *Search Item*
10. *Add an item to list*
11. *Push notification to application*
12. *View Notification Details*
13. *Clear Notification*

Use Case Diagrams:

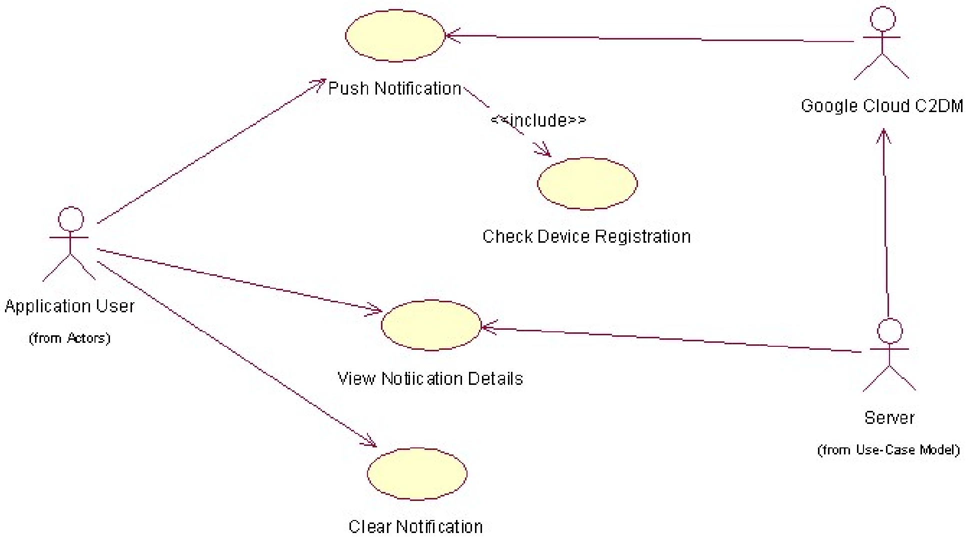


Fig 3.1 : Use case Diagram of sending update

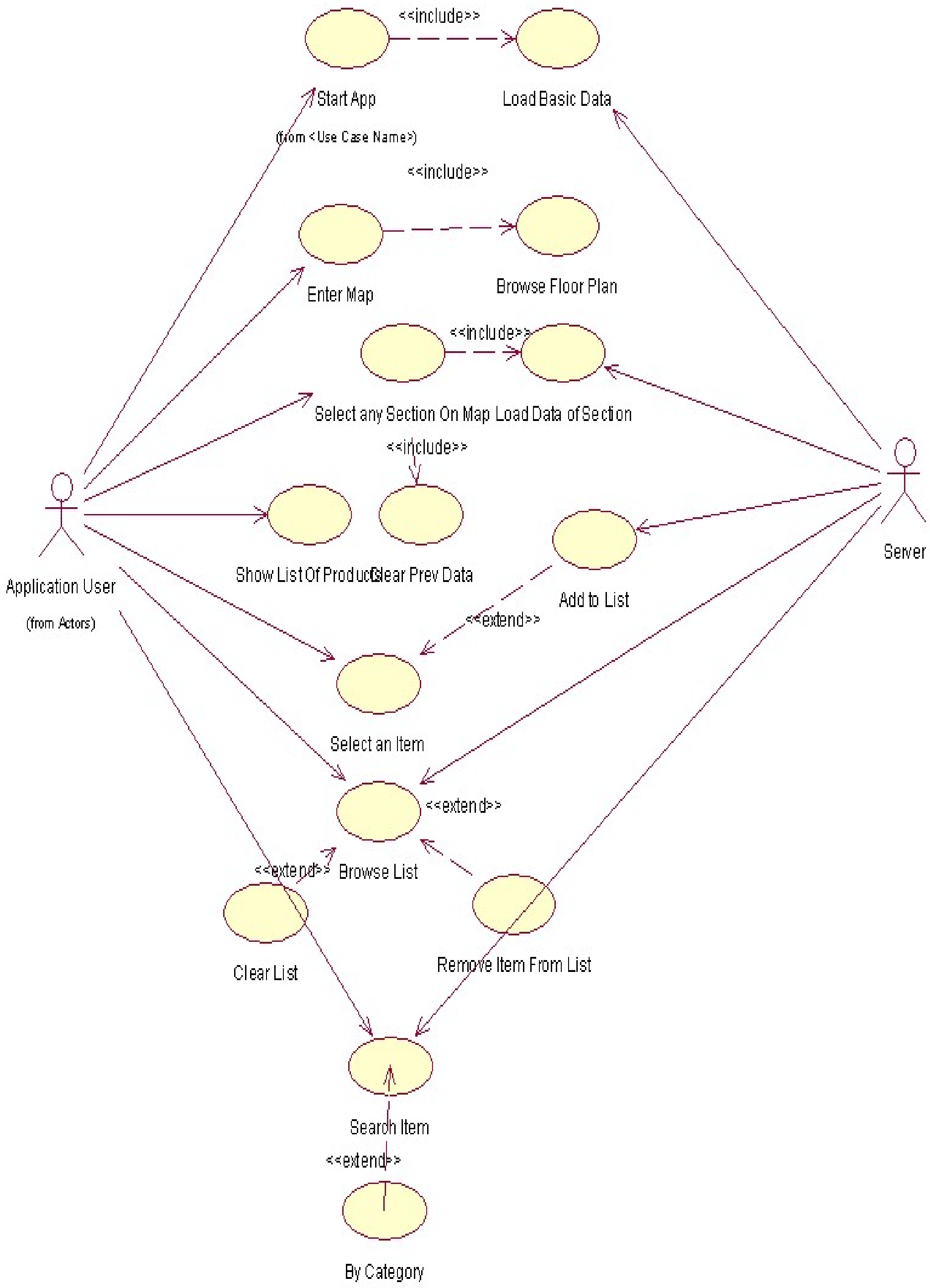


Fig 3.2: Use case diagram for project

Activity Diagram:

Activity diagram shows the behavioral model of the system.

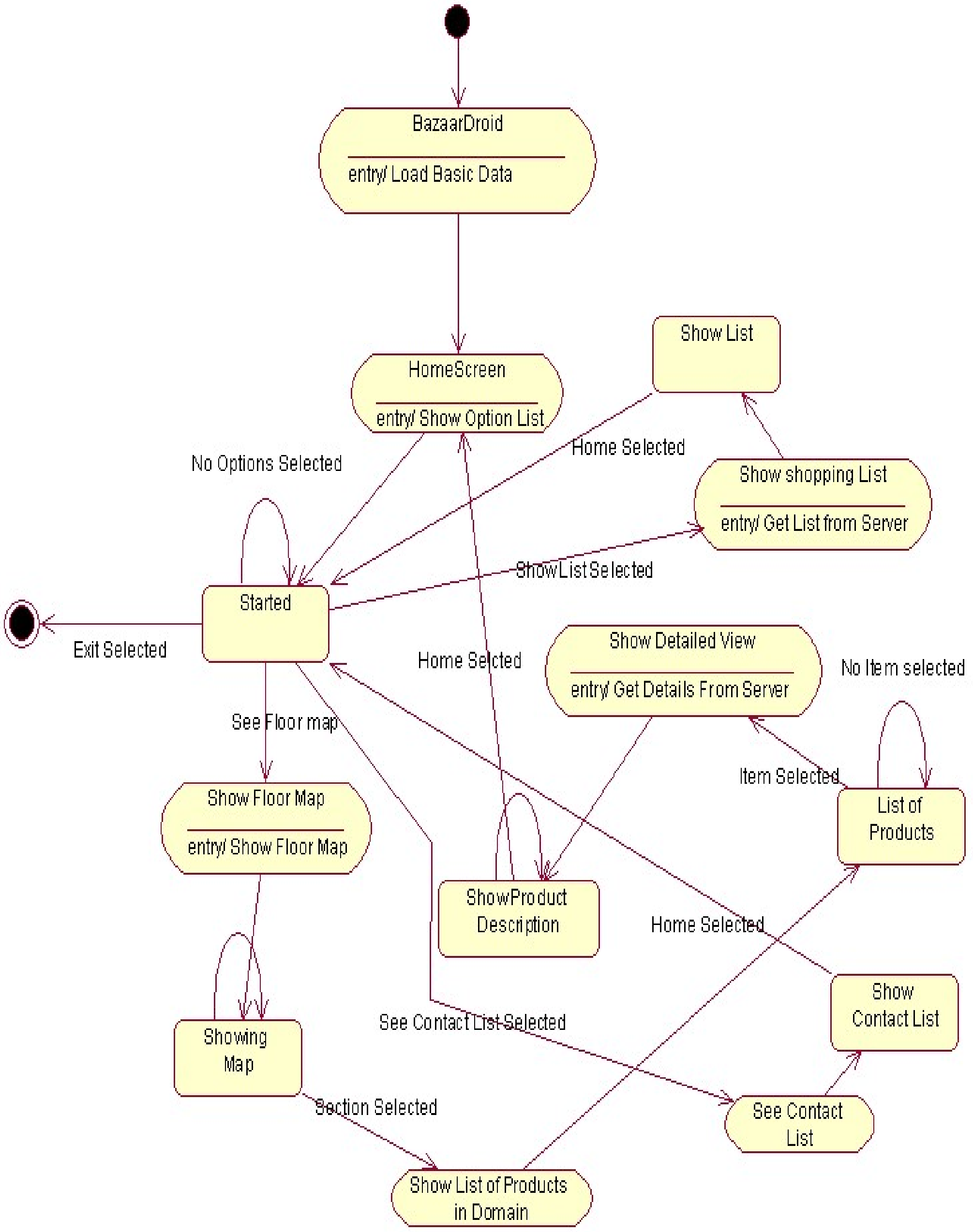


Fig 3.2: Activity diagram for project

3.4 Class Model and Description

3.4.1 Class Diagram

Packages:

- com.project.bazaardroid
- com.google.android.c2dm

Classes:

- com.project.bazaardroid
    - BazaarDroid
    - EnterMap
    - GetNotification
    - ProdDescActitivity
    - FloorGalleryActivity
    - CommentsOnProd
    - SearchActivity
    - SplashScreen
    - C2DMReceiver
    - C2dm
    - AboutProd
    - Contacts
    - FavListActivity
    - DataHelper
    - MyFavListAdapter
    - PublicInfo
    - EnterMap
    - ImageAdapter
    - MyHomeAdapter

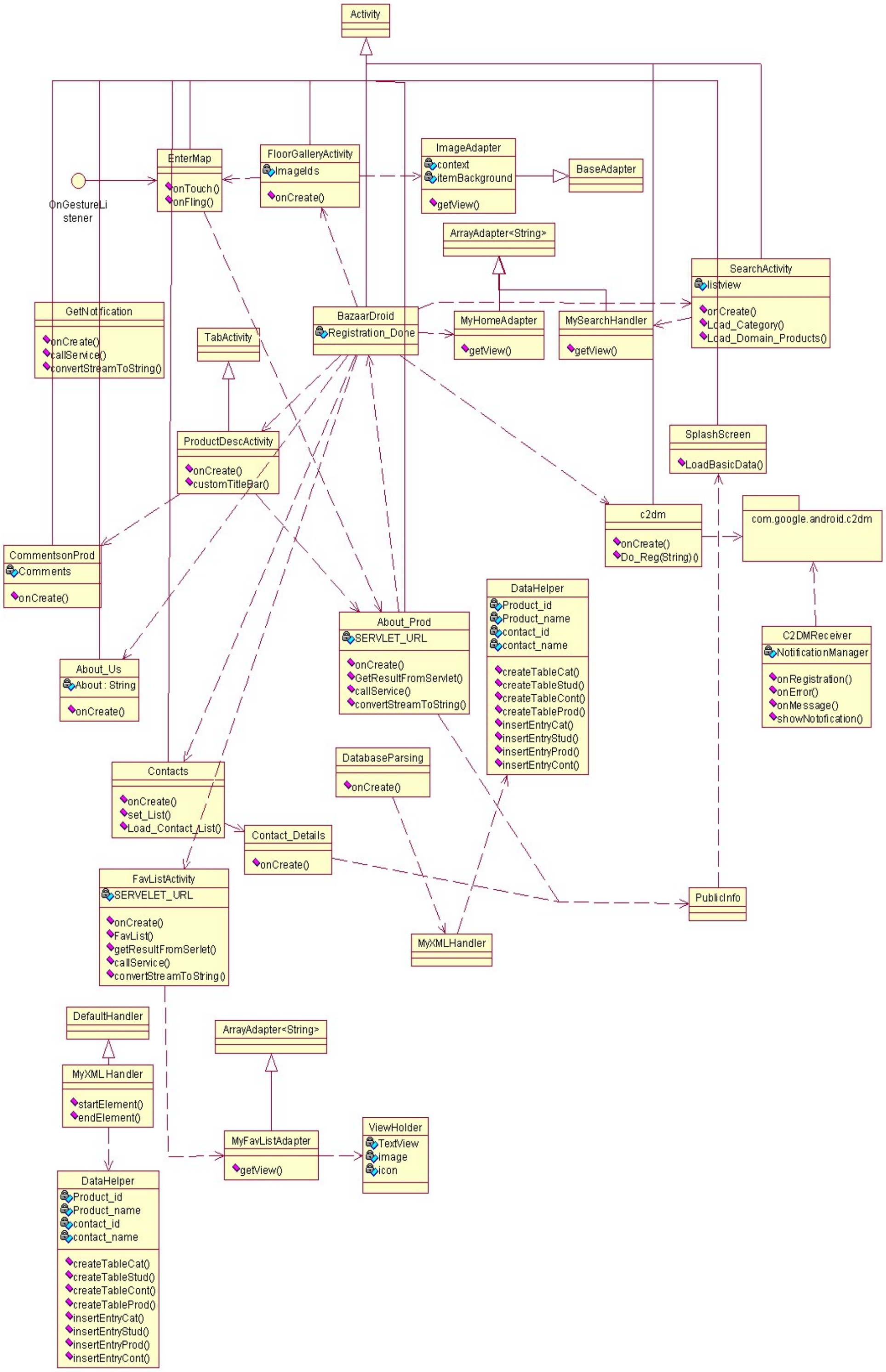


Fig 3.4: Class diagram for project

3.5 External requirements

3.5.1 Interface Requirements

- This system is likely to be used by a wide spectrum of users. So usability will be a critical issue with this project. User Interfaces should be such that users can accomplish the functions with least no of steps and the use has to be hassle free.
- There will be a set of user interfaces so as to facilitate easy communication with the user. It can be accomplished with the help of forms for various purposes including log-in, the main form, etc. Help buttons will be provided at appropriate places for better understanding of the system. Proper error messages will be displayed whenever and wherever required.
- The user interface would be such that the user can access every feature from every screen including going back to the home screen.

3.5.2 Performance Requirements

- The system response time to user actions should be reasonable.
- The mobile device have limited memory and processing speed. The application server needs to process request from client.

3.9 Restrictions, Limitations, and Constraints

- Limited processing power and memory of the Android phone.
- Limited screen size

4 PROJECT PLAN

4.1 Introduction

Here we present the software project plan in detail.

- Problem Definition

  To build an application that will enable ANDROID phone to run as an client and access application Server information and use C2DM service in the mobile device.

- Management and technical constraints

  The main constraint is the limited processing power of the ANDROID phone. The secondary issue is the memory of Android phone as we are using SQLite in device which has limited memory. Third important issue is the parsing of XML file that is present at the server side.

- Purpose of the Document

  This document seeks to provide the Project Plan for the project, BazaarDroid, An application that provides user interface for the shopping mall database and the offers that are present at the Server side, to be developed as a part of Final Year Engineering Project at Pune Institute of Computer Technology with purpose of obtaining a degree in Computer Engineering at the Pune University. This document will be used by faculty of Computer Engineering department and the external guide.

4.2 Project Estimates

- Historical data used for estimates

  The historical data used is many other android open source applications.

- Estimation techniques applied and results

  - Estimation technique

    BASIC COCOMO technique of estimation:

    Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of lines of code (KLOC).

Formula:

Effort Applied = a (KLOC) ^{b}

Development Time = c (Effort Applied) ^{d}

People required = Effort Applied / Development Time

the coefficient values are (for Organic Projects):

- Estimate for technique

Estimate generated for BASIC COCOMO:

Input:

Delivered Source Instructions (thousands) (KDSI)	3
Development Mode	Organic
Average Cost Rate ($/PM)	0

Results:

Effort	8	person-months (PM)
Schedule	6	Months

Development Cost	0	$
Productivity	375	instructions per person-month
Average Staffing	1.3	full-time-equivalent software personnel

Phase Distribution:

	Effort (PM)	**Schedule (mo.)**	**Staff (avg.)**	**Cost**
Plans and requirements *	0.5	0.6	0.8	0
Product Design	1.3	1.1	1.2	0
Programming	5.4	3.8	1.4	0
Detailed Design	2.1			
Code and unit test	3.4			
Integration and test	1.3	1.1	1.2	0

Activity Distribution (Staff) by Phase:

Activity	**Phase**				
	Plans and Requirements	**Product Design**	**Programming**	**Integration and Test**	**Maintenance**
Requirements Analysis	0.4	0.2	0.1	0	0
Product Design	0.2	0.5	0.1	0.1	0

Programming	0	0.2	0.8	0.4	0
Test Planning	0	0.1	0.1	0	0
Verification and Validation	0.1	0.1	0.1	0.4	0
Project Office	0.1	0.1	0.1	0.1	0
CM/QA	0	0	0.1	0.1	0
Manuals	0	0.1	0.1	0.1	0
TOTAL	0.5	1.3	1.5	1.2	0

- Reconciled Estimate
    - ➢ Estimate Cost:

      As we are working on open source software, the only cost involved is the cost of hardware like Android phone.

    - ➢ Estimated Time:

      Studying the android programming will take 2-3 months to start developing on the android platform.

4.3 Risk Management

We have classified the risks into different categories as:

- Development Risks
- Business Risks
- Project Risks
- Requirements Risks

- Risk Table

Catastrophic: Failure to meet the requirement would result in mission failure

Critical: Failure would degrade performance; mission success is Questionable.

Marginal: Failure would result in degradation of secondary mission.

Negligible: Failure would have slight impact

No.	Risks	Category	Probability	Impact
1	Lack of knowledge of Android programming	Development	10%	Critical
2	ANDROID OS will not meet expected functionality of C2DM	Development	5%	Critical
3	Phone may go out of memory	Development	10%	Catastrophic
4	Modules require more testing and further implementation work	Project	20%	Marginal

5	Increased work load	Project	20%	Marginal
6	Deadline may be extended	Business	15%	Marginal

- Overview of Risk Mitigation, Monitoring, Management

No.	Risk	Mitigation	Monitoring	Management
1	Lack of knowledge of Android Programming	Android programming practices are studied.	Discuss about the topic with all the team members and the guide	Find out where all the members lack in knowledge and make an extra effort to read material on the same.
2	ANDROID OS will not meet C2DM services	Upgrade device to Android 2.2 or above	See if the application is running smoothly.	Find the appropriate and latest modifications in ANDROID OS.
3	Out of memory	Only draw the changed instructions	See if the RAM is being utilized efficiently	Retrieve the changed part of the desktop
4	Modules require more testing and further implementation work	Allotting a sufficient amount of time for testing	Get each module checked by the guide.	Increase the man hours invested by each member so that all work can be completed easily within specified time
5	Increased work load	Divide the task uniformly	Perform regular reviews to check if productivity is high.	If a developer is unable to perform to desired standards, reassign the task to other

				developer. Reschedule the planner to balance the workload.
6	Deadline may be extended	Build the first prototype with minimal features	A regular review of the development work by the guide	Decide the no. of prototypes with the added features of each

4.4 Project Task Set

We divided tasks under two layers:

1. GUI layer
2. Event Handler
3. Connection

The different task sets that are critical in the project under GUI layer are: –

<table>
<tr><th>S. No</th><th>Task ID</th><th>Task</th></tr>
<tr><td>1</td><td>T-GUI-1</td><td>Drawing floor maps.
<table>
<tr><th>Subtask ID</th><th>Name of Task</th></tr>
<tr><td>T-GUI-1-1</td><td>Studying ANDROID Layouts and XML.</td></tr>
<tr><td>T-GUI-1-2</td><td>Drawing full in Layouts</td></tr>
</table></td></tr>
<tr><td>2</td><td>T-GUI-2</td><td>Managing mouse pointer at GUI level using trackball
<table>
<tr><th>Subtask ID</th><th>Name of Task</th></tr>
<tr><td>T-GUI-2-1</td><td>Studying trackball programming on ANDROID</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>3</td><td>T-GUI-3</td><td>Canvas toolbox
<table>
<tr><th>Subtask ID</th><th>Name of Task</th></tr>
<tr><td>T-GUI-3-1</td><td>Zooming the canvas</td></tr>
<tr><td>T-GUI-3-2</td><td>Product Information</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>T-GUI-4</td><td>Getting pixel color's and checking appropriate section selected.</td></tr>
<tr><td>5</td><td>T-GUI-5</td><td>Making use of Efficient ways of coding for android GUI.</td></tr>
</table>

The different task sets that are critical in the project under Connection layer are: –

<table>
<tr><th>S. No</th><th>Task ID</th><th>Task</th></tr>
<tr><td>1</td><td>T-CM-1</td><td>Studying and Refining the proper Java code.</td></tr>
<tr><td>2</td><td>T-CM-2</td><td>Upgrading ANDROID OS to 2.2 or above</td></tr>
<tr><td>3</td><td>T-CM-3</td><td>Work on Server side.
<table>
<tr><th>Subtask ID</th><th>Name of Task</th></tr>
<tr><td>T-CM-3-1</td><td>Studying Tomcat extension of server.</td></tr>
<tr><td>T-CM-3-2</td><td>Creating servlets to process request from the client.</td></tr>
<tr><td>T-CM-3-4</td><td>Making XML file which stores database at server side.</td></tr>
<tr><td>T-CM-3-5</td><td>Test the code</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>T-CM-4</td><td>Packet creation</td></tr>
<tr><td>5</td><td>T-CM-5</td><td>Using ANDROID Internet access permissions to allow the device to access Server.</td></tr>
</table>

S.No	Task ID	Task
1	T-1	Combining GUI layer and Connection Management Layer
2.	T-2	Testing

- Task network

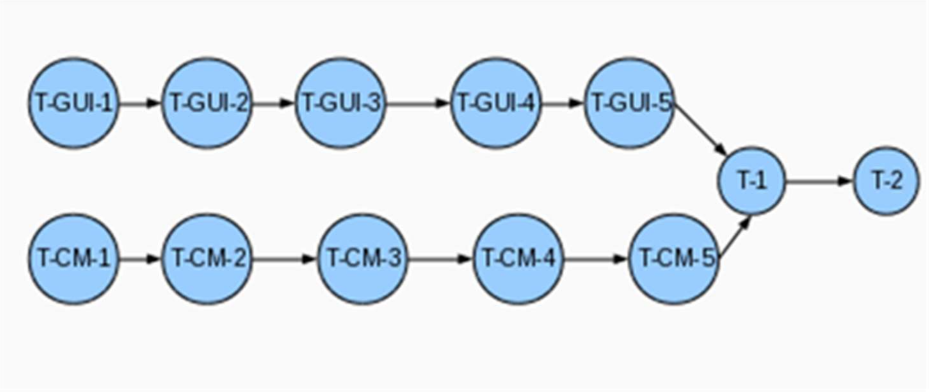


Fig 3.5: Task network

5 DESIGN GUIDELINES

5.1 Introduction

5.1.1 Goal and objective

The purpose of the document is to convey information about the design of architecture i.e. the different layers of the application, component and user interface that will probably be used during the development phase of the Project.

5.1.2 Software Context

It reveals architecture, component design and probable constraints of this application.

5.1.3 Version of Document

Version 1.0

5.2 Architectural Design

The basic concept requires the Application server maintained by the Event managers to be authorized for the c2dm services from google server.

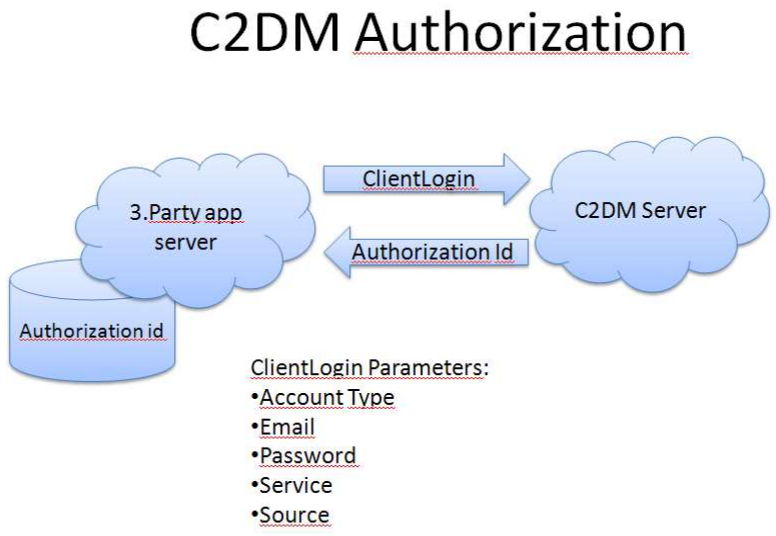


Fig 5.1: C2DM Authorization

Now the Android phone needs to register itself to receive messages from the c2dm server with the e-mail id of the authenticated Application server.

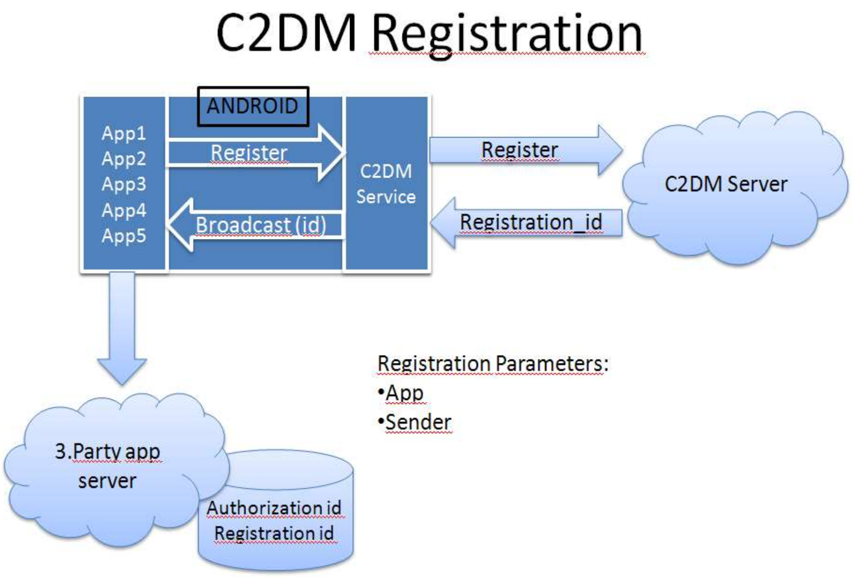


Fig 5.2: C2DM Registration

- Life Of The Message :

The connection is present between Conn Server which is to be used for sending notification to the application about the data or update from application server.

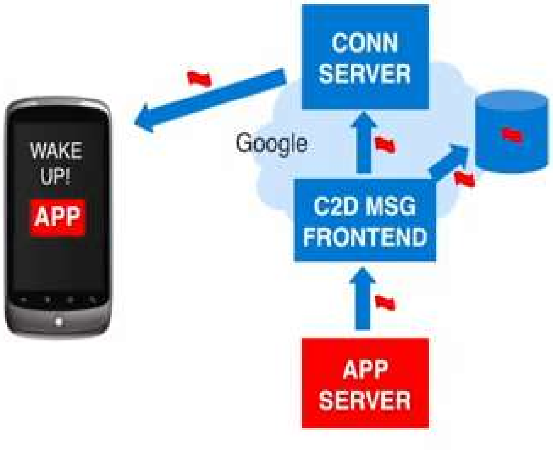
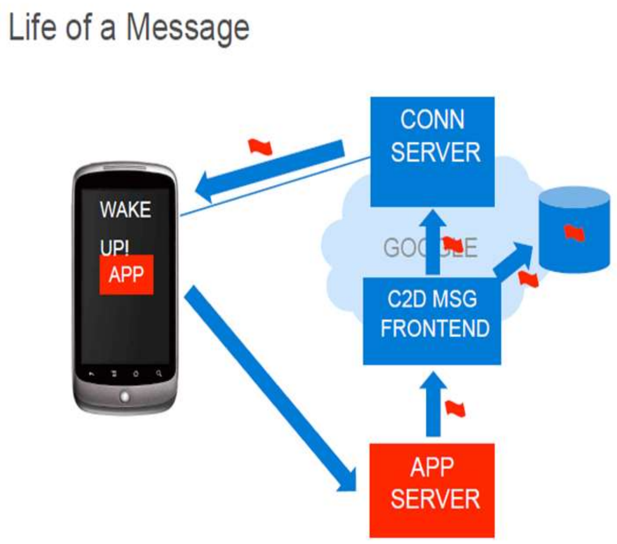


Fig 5.3: Life of Message

Once the message is received by the application it can connect to application server to fetch new data/dynamic update.



5.3 Data Flow Diagrams

DFD Level-0:

A Data Flow Diagram (DFD) is a graphical representation of data through an information system. DFD can be used to provide a clear representation of any business function.

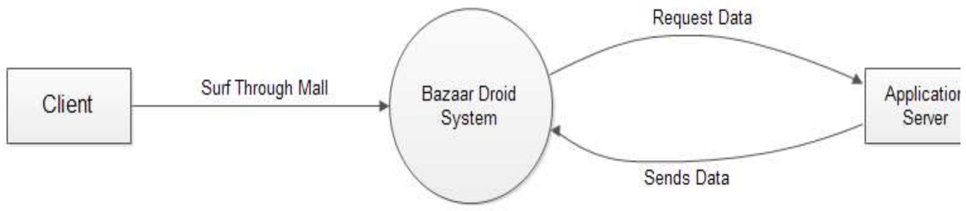


Fig 5.4: DFD Level 0

DFD Level-1:

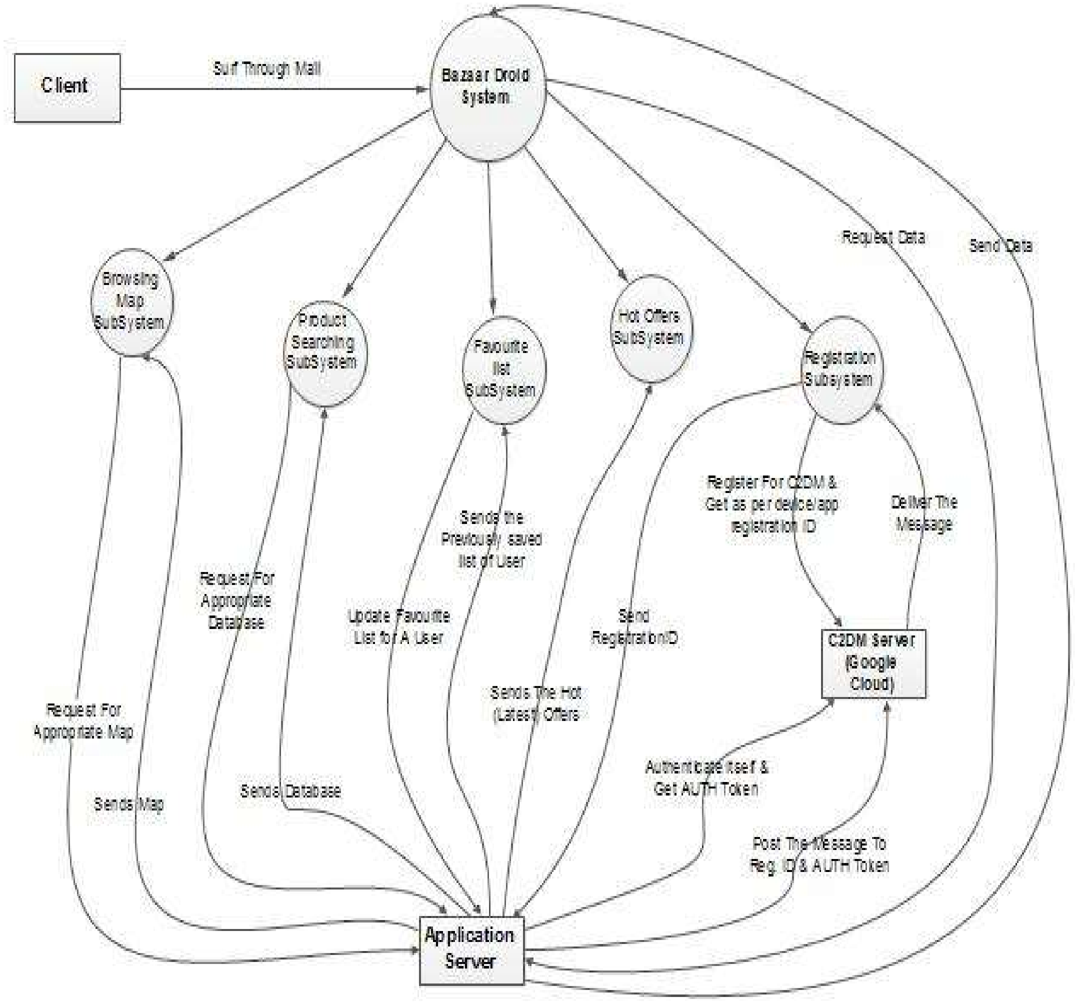

Client
Surf Through Mall
Bazaar Droid System
Browsing Map SubSystem
Product Searching SubSystem
Favourite list SubSystem
Hot Offers SubSystem
Registration Subsystem
Request Data
Send Data
Register For C2DM & Get as per device/app registration ID
Deliver The Message
Sends the Previously saved list of User
Request For Appropriate Database
Update Favourite List for A User
Send RegistrationID
C2DM Server (Google Cloud)
Sends The Hot (Latest) Offers
Request For Appropriate Map
Sends Map
Sends Database
Authenticate itself & Get AUTH Token
Post The Message To Reg. ID & AUTH Token
Application Server

Fig 5.5: DFD Level 1

5.4 Use Case Diagrams

A Use Case Diagram is a type of behavioral diagram defined by UML created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals – represented as use cases – and any dependencies between those use cases.

1.

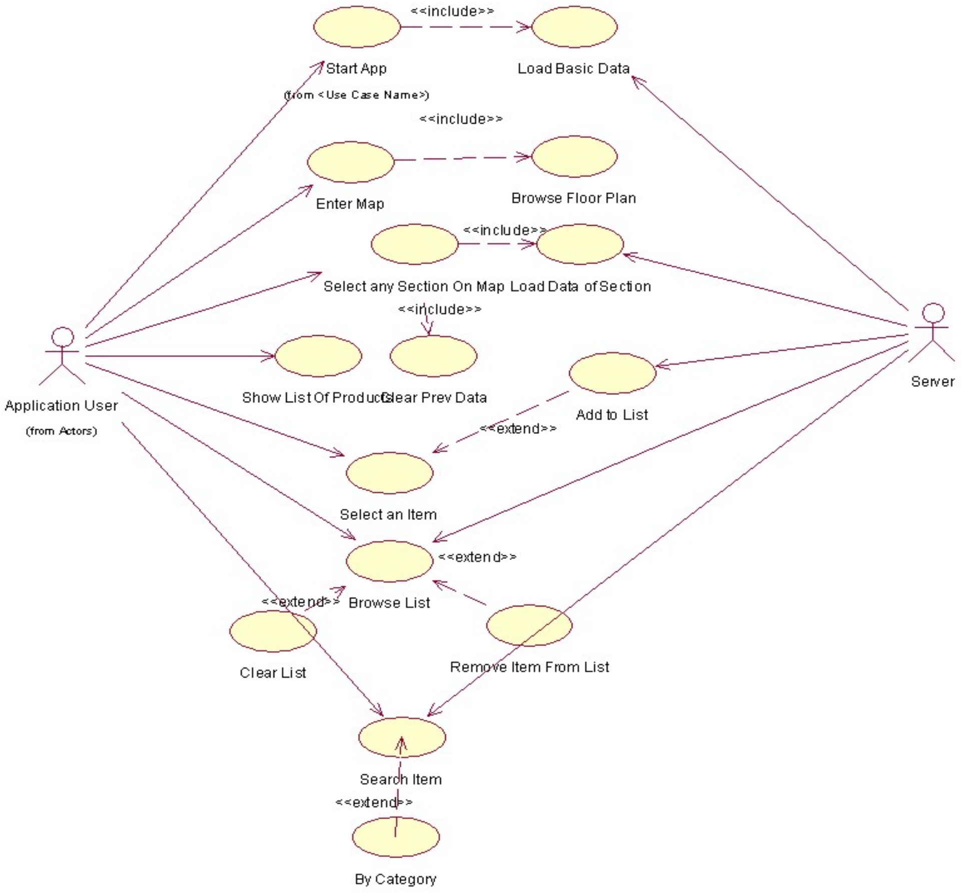


Fig 5.6: Use case diagram-1 for project

2.

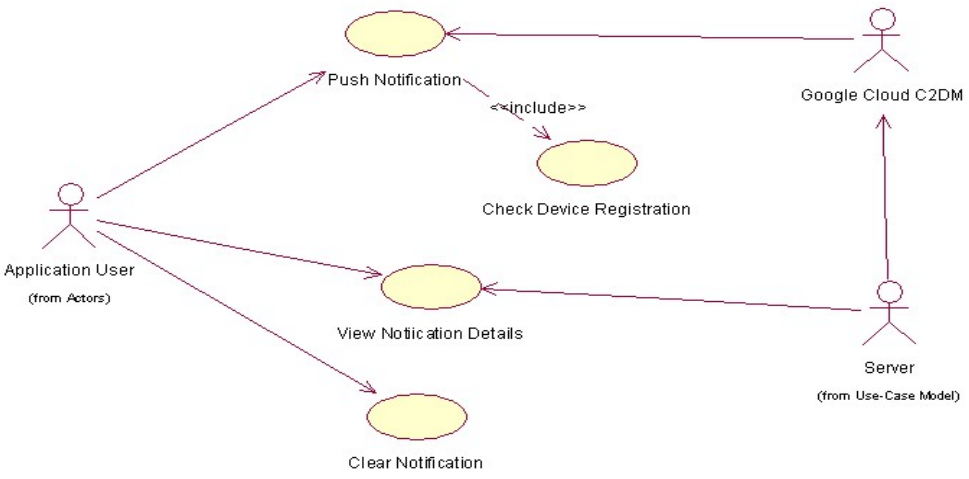


Fig 5.7: Use case diagram-2 for project

5.5 Class Diagram

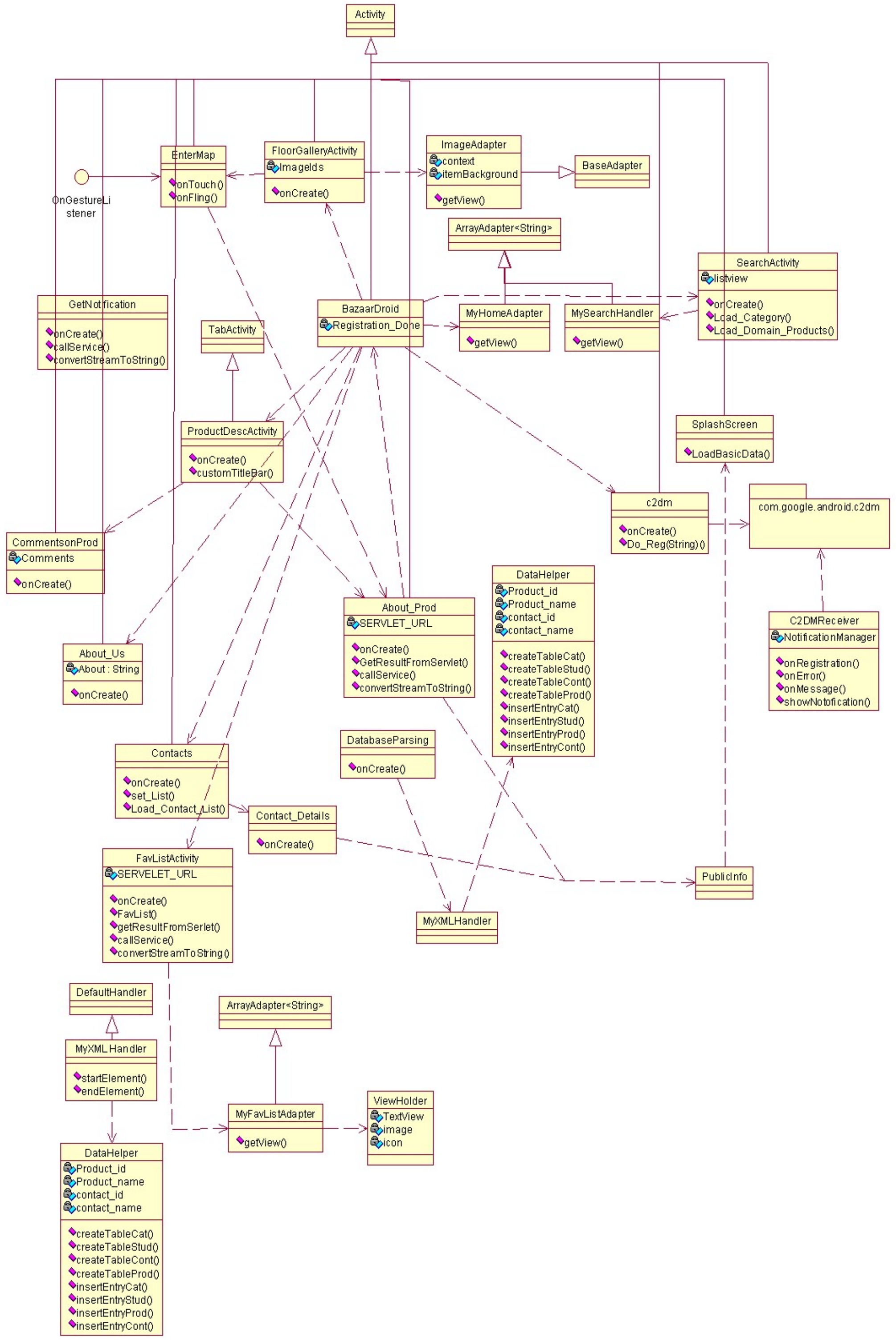


Fig 5.8: Use case diagram for project

5.6 Activity Diagram

An Activity Diagram represents the business and operational step-by step workflows of components in a system. An Activity Diagram shows the overall flow of control.

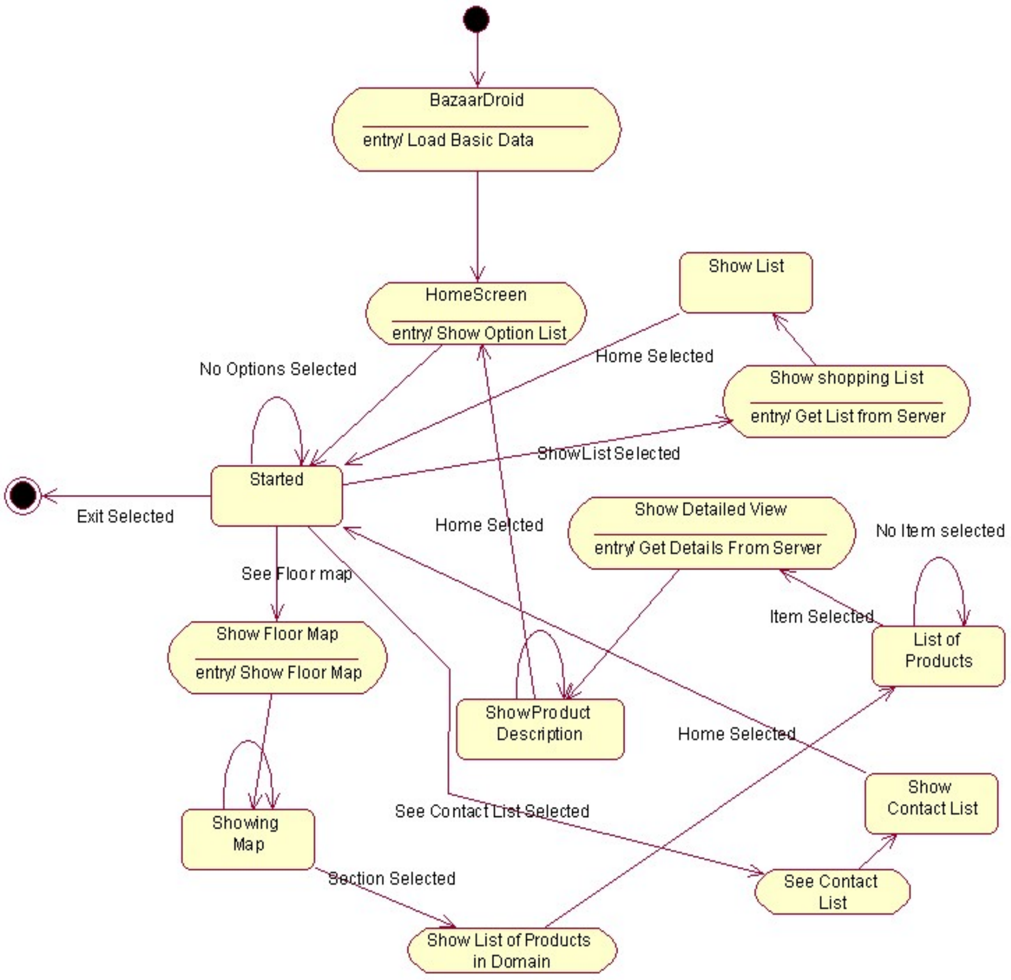
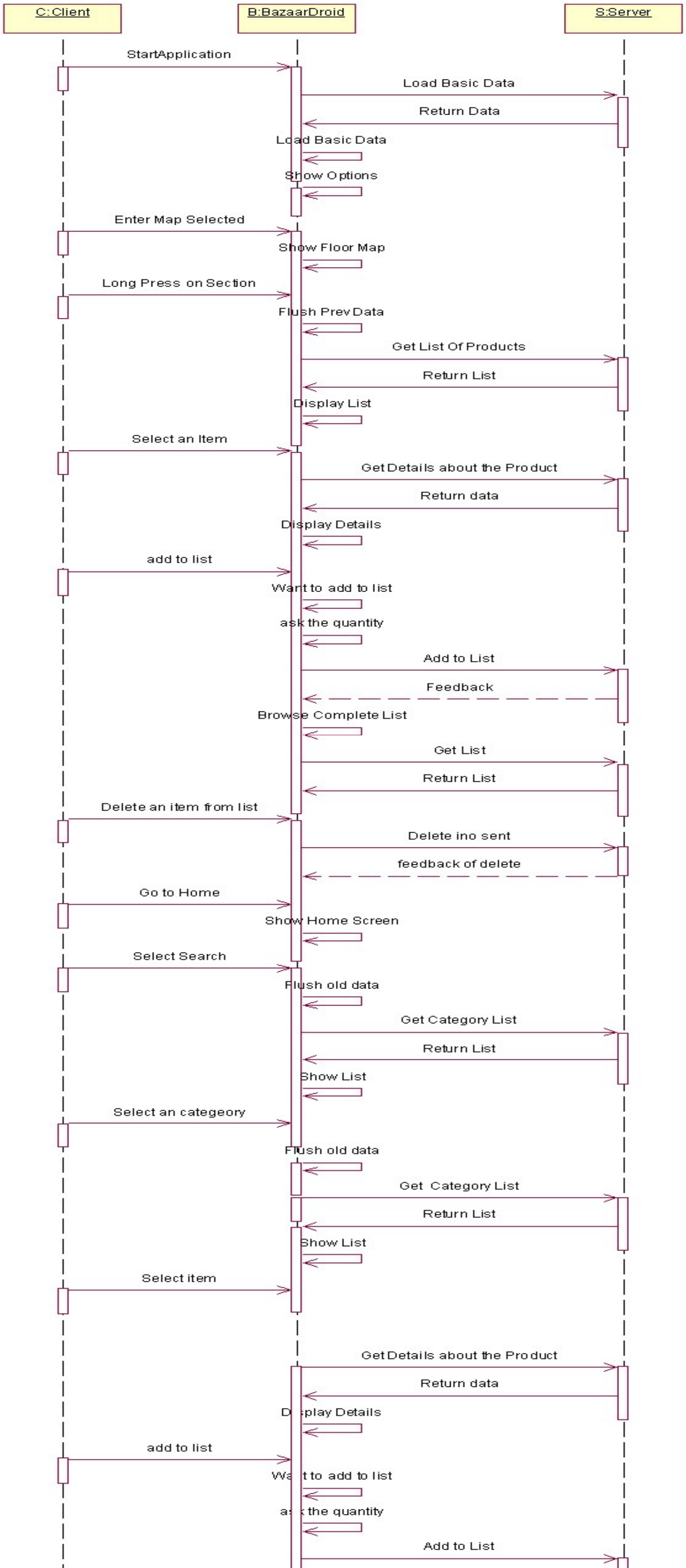
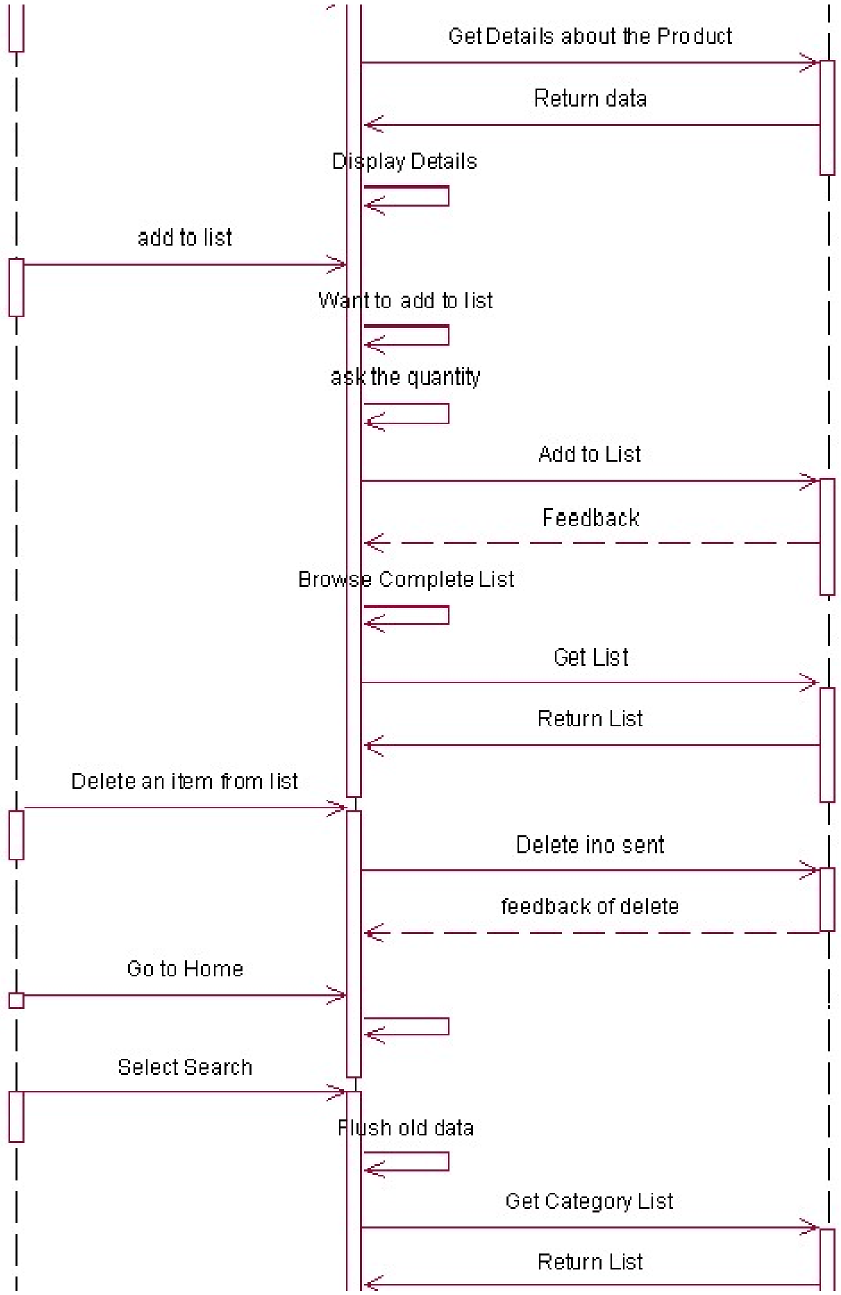


Fig 5.9: Activity Diagram for Mall application

5.7 Sequence Diagram


C:Client
B:BazaarDroid
S:Server
StartApplication
Load Basic Data
Return Data
Load Basic Data
Show Options
Enter Map Selected
Show Floor Map
Long Press on Section
Flush Prev Data
Get List Of Products
Return List
Display List
Select an Item
Get Details about the Product
Return data
Display Details
add to list
Want to add to list
ask the quantity
Add to List
Feedback
Browse Complete List
Get List
Return List
Delete an item from list
Delete ino sent
feedback of delete
Go to Home
Show Home Screen
Select Search
Flush old data
Get Category List
Return List
Show List
Select an categeory
Flush old data
Get Category List
Return List
Show List
Select item
Get Details about the Product
Return data
Display Details
add to list
Want to add to list
ask the quantity
Add to List


Get Details about the Product
Return data
Display Details
add to list
Want to add to list
ask the quantity
Add to List
Feedback
Browse Complete List
Get List
Return List
Delete an item from list
Delete ino sent
feedback of delete
Go to Home
Select Search
Flush old data
Get Category List
Return List

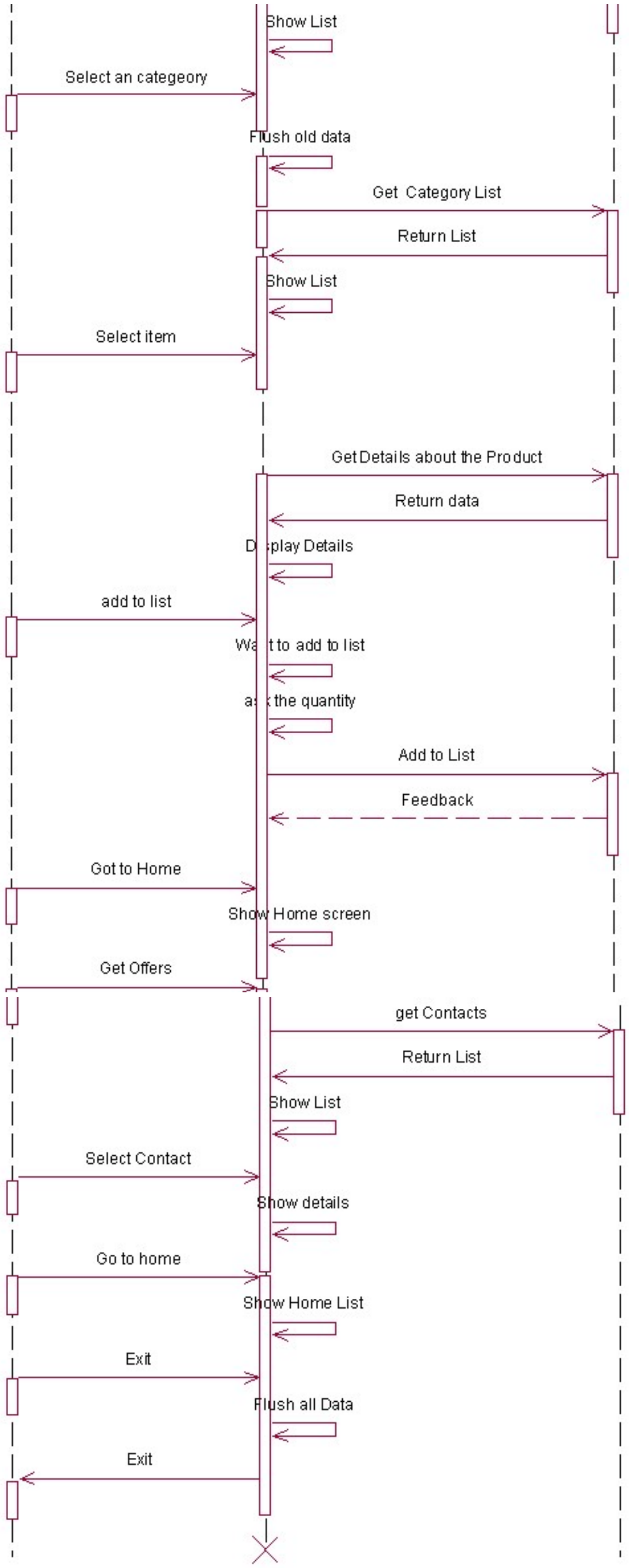


Fig 5.10: Sequence diagram for project

5.8 Component Diagram

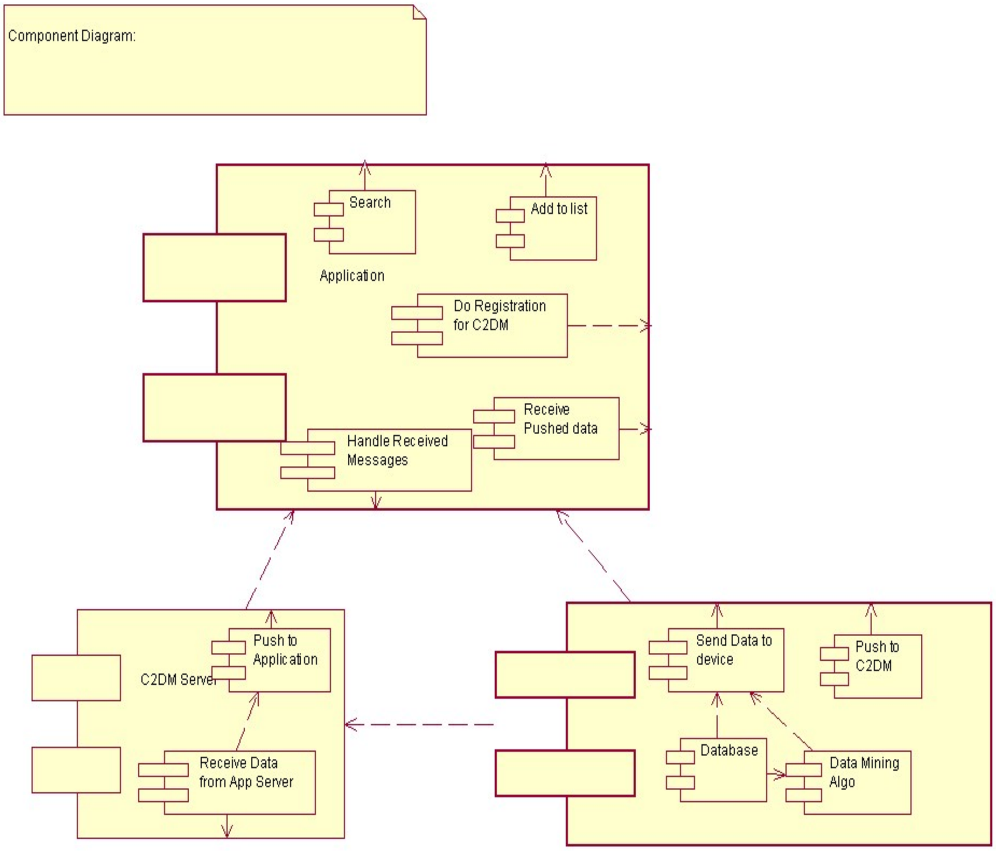


Fig 5.11: Component diagram for project

5.9 Deployment Diagram

Deployment diagrams explain the deployment scenario and hardware and software resources required to deploy the system. Resource demand of this system is such that server should be centralized where business logic resides and all the computing should be on that server only. Web server will work as a centralized computing server, here all business logic resides. This web server communicates with database server through communication system as internet.

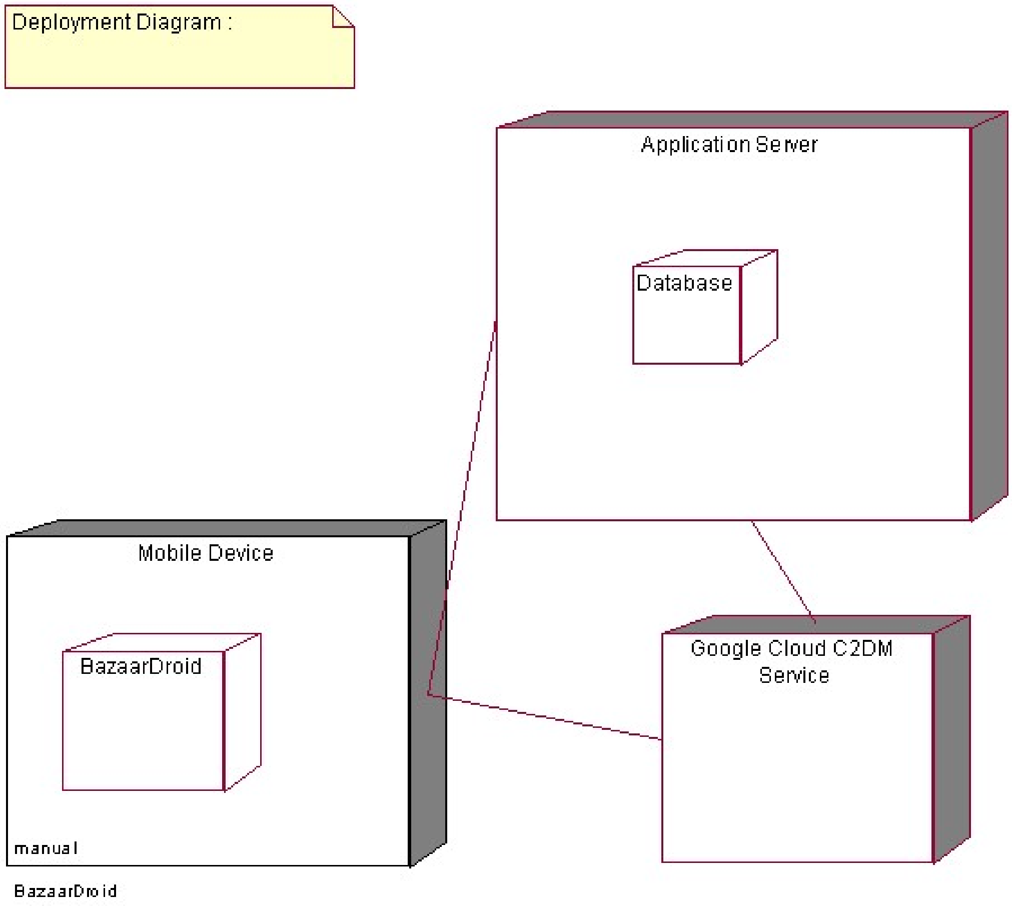


Fig 5.11: Deployment diagram

6 TEST PLAN AND REPORTS

6.1 Goals and Objectives

Software testing is one of the elements in software project that is often referred as verification and validation. Verification refers to set of activities that ensure that the software correctly implements a specific function. Validation refers to different set of activities that ensure that the software that has been built satisfies customer requirements. The results of testing will not only help in knowing which functions perform below average but also helps to make the system more accurate. Testing is considered to be an unavoidable part of any project.

6.2 Statement Of Scope

Software testing is a critical element of software quality assurance and represents ultimate review of specification, design and code generation. The testing process involves executing a program with the intent of finding error. Test plan for BazaarDroid includes testing of the individual components for module interfaces, local data structures, boundary conditions to identify that all modules work properly. Integrating the individual components into the system, simultaneously performing regression testing so that there are no side effects due to integration, then follows it. Finally the system as a whole is tested.

We test the following functionalities of our system

a. Server Side Application

1. To get the device registered for updates.
2. To get Auth ID from google server.

3. To host XML files for the client.
4. To receive reviews from the customer users.
5. To maintain the Favorites List of the customer users..

b. Client(Android Phone) Side Application

1. To get Registration ID from the C2DM servers.
2. To register for update from Application Server.
3. Browse through floor maps.
4. Retrieval of the list of domains.
5. To see the list of products and their details by parsing the XML file.
6. To make and update Favorite List.
7. To show Hot Offers.
8. To post comments.

6.3 Test Plan

The test plan is to be developed for affirming and validating the proper functioning of "BazaarDroid". All features of the product are to be tested, keeping in mind the primary goals of accuracy, efficiency and user convenience.

6.4 Test Strategy

6.4.1 Unit testing

In unit testing, initially the focus is on each component individually, ensuring that it functions properly. Unit testing makes a heavy use of white-box testing techniques and the step can be conducted in parallel for multiple components. Various unit-testing strategies include testing of module interfaces, local data structures, boundary conditions and all independent paths.

Units or components in our project

A. Web Services:

This includes the testing of each of the individual web services. Some web services were tested by invoking them from the browser and certain web services required creation of client stubs to be tested effectively.

B. Store Side Application:

The store side application was first tested for various client side validations and conditions. This application was then integrated with the web services back-end to be able to test the functionality of the store side application.

C. Desktop Side Application:

The desktop side application was first tested for various client side validations and conditions. This application was then integrated with the web services back-end to be able to test the functionality of the client side application.

6.4.2 Integration Testing

The Integration Testing is on design and the construction of the software architecture. The individually tested components are integrated as a complete software package. Black Box test case design techniques are the most prevalent during integration.

Top-down integration testing is an incremental approach to construction of program structure, thereby verifying major control or decision points early in the test process. Bottom up integration testing begins construction and testing with atomic modules i.e. components at the lowest levels in the program structure. Regression testing is re-execution of some subset of tests that have been already been conducted to ensure that changes have not propagated unintended side effects

In case of BazaarDroid the client side application, store side application Server were integrated on a local network and the product as a whole was tested. This included testing all functional specifications of the product.

6.4.3 Black Box And White Box Testing

- Black box testing

  Black box testing is carried out by giving out of the ordinary inputs in order to test the system as a whole and verifying whether the system reacts in a consistent and concise manner. Here no internal knowledge of the system is available. In other words, we can test only in terms of inputs and outputs and have no presumed knowledge regarding module interfaces, data structures, files and so on. The different black box tests that can be carried out are –

  i. To be able to successfully get Registration ID from the google C2DM server, send the Reg ID to application server.
  ii. Adding new products to the store inventory by edting XML file.
  iii. Ability to post comments on the product.
  iv. Browsing store floor maps from client side application.
  v. To add products in Favorite List maintained for each user.
  vi. Viewing received reviews while in a store (mall).

vii. Sending instant offers updates all registered users.

- White box testing

White box testing is carried out by giving specific inputs in order to expose the internal shortcomings of the system. By analyzing the different module interfaces, the way we handle the data structures and general logical flow of control, we can arrive at test cases that validate the robustness of the system. The different white box tests that can be performed are –

i. Going through various paths possible to check whether each reachable line of code is being tested.
ii. Performing the above test with removal of more generic elements.

6.5 Expected Software response

The software responses for the different tests conducted above are as follows –

Successful registration of the users.
Registered Users receiving the product offer updates.
User browses the floor map of the mall.
User can search the products by category.
User can add products in Favorite List maintained by the server.
Posting comments on products and viewing received reviews while in a store (mall).
Users can go through Hot Offers(Latest).
Users can go through the contact list.

After adding a product to the inventory of a store, the product should be visible in the store catalogue when the user browses through the application.

6.5 Testing Resources and Staffing

Tested the Application on Android 2.2 sdk.

6.6 Testing tools and Environment

Main resources used for testing include following tools **:**

Android SDK , Logcat.

6.7 High order testing

6.7.1 Alpha Testing:

The alpha testing is conducted at the developers end by the customer.Alpha tests are conducted in a controlled environment. Each module of the integrated system was tested

- By our colleagues
- By our mentor

6.7.1.1 Expected results

The main responses expected are:

- Usability
- Understandability
- System Stability

Also the correct results are expected for various inputs to the system. The main aim is to detect the errors if any and the user interface problems if the user feels so.

- Pass/fail criterion
  - Pass Criterion
    - No errors are discovered in the system.
    - The user is comfortable with the use of the system.
    - The user is able to sign up, create contact list, send images across the network, post comments, browse store catalogue, text/audio/video chat with other desktop users.
    - System does not crash.
  - Fail Criterion
    - The fail criterion for alpha testing is discovery of errors and complicated user-interface if the user feels so.
    - Also the system should be able to perform the intended functions of the user else it fails.
    - System crashes frequently.

6.8 TEST CASES

Table 6.1: Test Cases

Test Case	Expected Outcome	Actual Outcome
Browse for the floor map of the mall	Showing floor map of the mall	Showing floor map of the mall
Selection of different section on a floor map	Section selected	Section selected
Loading products of a section	Products of a section loaded	Products of a section loaded
Showing details of the product selected	Details of the products shown	Details of the products shown
Add the product to the favorite list of the user	Product added to the favorite list of the user	Product added to the favorite list of the user
Saving Favorite List of user on the server	Favorite List of user saved on the server	Favorite List of user saved on the server
User posting comments on products	Comments posted for the products saved on the server	Comments posted for the products saved on the server
Check the favorite list of the user	User views its favorite list	User views its favorite list
Search the Products by Category	Products shown by category	Products shown by category
User checks the Hot Offers	Hot Offers shown to the user	Hot Offers shown to the user
User registers for updates at C2DM server	C2DM sends back Registration ID	C2DM sends back Registration ID
Server request for Authenticating ID at Google Cloud	Gets AUTH Token	Gets AUTH Token
App. Server pushes	Gets Notification on	Gets Notification on

updates to registered device	Device	Device
Flush SQLite Data before exiting	Data Flushed	Data Flushed
Adding a product to the inventory of a store	The product should be visible in the store catalogue when the user browses through the application.	The product should be visible in the store catalogue when the user browses through the application.
Go through the Contact List	Contact List shown	Contact list shown

7 FUTURE WORK

- The system can be instilled with functionality of online shopping (which involves online ordering and payment and subsequent shipping to the user specified address).
- We can have Application server at store side more intelligent to send automatic updates to device.
- By integrating various image processing libraries, we could achieve a working model of ‘Virtual mirror’. This will help shoppers try out apparels without having to wear them physically. Also, these image processing libraries could be used to simulate different environments for particular shopping occasions like marriages.
- The system can be made smarter by implementing an AI mechanism which will send out intelligent suggestions to shoppers. For instance, it would be able to suggest what pair of shoes available will complement the apparel the shopper is trying.

8 SUMMARY AND CONCLUSION

8.1 Summary

Helping out shoppers make a quicker and better purchase decision was the main motivating factor behind ‘BazaarDroid’. The system has been designed keeping in mind the role which customer relationship management play in the retail industry. ‘BazaarDroid’ is beneficial for both the shoppers and the shop owners. It helps the shoppers make better purchase decisions and improves the footfalls to actual purchase ratio. Thus it brings new business to the shop owners.

Three basic two emerge out of the project, Device application, the store side application server:

Store-side application Server

- Store database in XML format of different sections and products.
- Contact list – Maintains a list of contacts. Also reflects their online/offline status

Device-side application (Android Phone)

- Logistics: Access important information about the event such as Product information, Floor Plan, Contact information and more.
- Floor Plans: Convenient access to floor plans.
- Search: Provide various search options like search by Category.
- Live Updates: Stay on top of what's going on. Get live updates from Mall.
- Rating: Get relevant feedback from Client. Conduct live Rating of products.
- Add the product to favorite list.
- Use data mining concept to show project once the product is added to favorite list.

The Communication structure between the store-side application and desktop application will be as follows:

- Store-side user (Store-side application) – to home user (Device application) duplex communication
- Home (Mobile application) – to home (Desktop application) duplex communication

Currently, the system is based on centralized database architecture. But as and when the number of users grows, the system will be required to migrate to distributed database architecture. Also, the web server will be required to be optimized so that it doesn't suffer any degradation in performance. It will also be required to keep the database architecture hidden from the three modules.

8.2 Conclusion

'BazaarDroid' successfully meets the needs which were the motivational factors in its development. It helps the users communicate in the medium they want to. Its real time working environment helps users get instantaneous responses for their queries. Moreover, 'BazaarDroid' is a feasible solution in the real world scenario.

9 REFERENCES

Bibliography

[1] Hello, Android: Introducing Google's Mobile Development Platform by Ed Burnette.

[2] Android in Action, Second Edition W. Frank Ableson, Robi Sen, and Chris King.

[3] Android Wireless Application Development **By:** Shane Conder; Lauren Darcey

Websites

[4] http://developers.android.com

[5] http://code.google.com/android/c2dm/index.html

[6] http://www.vogella.de/articles/Android/article.html

[7] http://developer.android.com/guide/topics/ui/index.html

[8] http://www.droidnova.com/

[9] http://www.androidpolice.com/