

# **FYP Title**

**Final Year Project**

**Session 2015-2019**

A project submitted in partial fulfillment of the degree of

BS in Software Engineering



Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior College, Lahore

Spring 2019

Type (Nature of project)	[ <input checked="" type="checkbox"/> ] <b>Development</b> [ <input type="checkbox"/> ] <b>Research</b> [ <input type="checkbox"/> ] <b>R&amp;D</b>			
Area of specialization	Software Engineering			
FYP ID	FYP-BSSE-F18-033			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BSEM-F15-113	MOAZZAM JAHANZEB (Project Leader)	mozamattari786@gmail.com	
(ii)	BSEM-F15-090	M.USMAN KHAN	mynameisusman0@gmail.com	
(iii)	BSEM-F15-074	RABEES RIAZ	rabeesbro52@gmail.com	

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

### Plagiarism Free Certificate

This is to certify that, I MOAZZAM JAHANZEB S/D of Ghulam Muhyudin Watto, group leader of FYP under registration no BSEM-F15-113 at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 05-09-2019    Name of Group Leader: MOAZZAM JAHANZEB    Signature: \_\_\_\_\_

Name of Supervisor: Hafsa Tabassum

Co-Supervisor:

Designation: Junior Lecturer

Designation:

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HoD: Dr. Arfan Jaffar

Signature: \_\_\_\_\_

# Project Report

## Bird Lovers

### Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

---

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

---

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Dedication**

*This work is dedicated to my father, friends, team members and teachers who were always there for me when I needed a support to survive my education life.*

*They made me enough passionate from their words and lessons that I gave necessary time and hard work to my education needed to successful. To my Mother who gave me the ability to achieve my dreams. To my family member as a hope for their future.*

## Acknowledgements

I am really thankful to my supervisor **Mam Hafsah Tabassum** who helped me during whole work template of fyp and special thanks to **Mam Amna Khan** and **Sir Hafiz Shahzad** for providing us all the information which we need. **Mam Hafsah Tabassum** also provides us guidelines and also check our proper work flow in their office or through mobile messaging. I respect these teachers and thankful to them for support us.

## Executive Summary

Now-a-days we can see that everyone is moving towards the IT system. IT is providing everything online to us so it is time saving. So we find a problem in our society that people like to birds very much. So they like to keep birds in their houses. There are many places where people are selling birds but the problem is that they have to visit many places to buy suitable birds. So in this way their time is wasted and their money as well. That's the problem which we found in our society. We want address this problem of the society. We want to develop a **mobile application** based system which will provide a single platform to all bird lovers who want to sell or purchase birds. So if they will get all info regarding birds at a single platform, this will be very helpful for them. It will be time saving and money saving as well. There are already many companies who are providing online sell or buy items services but there is not any existing online system where people can purchase or sell small level birds. In this system, seller will be able to put ad of the bird which he want to sell and then other users will be able to see the ads and then they can select a list of birds which they want to buy. Just like the Pet Shop, Parenda Bazaar or olx.com. Additional feature of this system will be a chat box, which will be available for the users to make bargaining (between seller and purchaser). This is briefly overview of our system.

## Table of Contents

Chapter 1 .....	1
Introduction .....	1
1.1. Background.....	2
1.2. Motivations and Challenges .....	2
1.3. Goals and Objectives .....	3
1.4. Literature Review/Existing Solutions .....	3
1.5. Gap Analysis.....	4
1.6. Proposed Solution .....	4
1.7. Project Plan.....	5
1.7.1. Work Breakdown Structure .....	5
1.6.1. Roles & Responsibility Matrix.....	6
1.6.2. Gantt Chart.....	7
2.1. Report Outline .....	9
Chapter 2 .....	11
Software Requirement Specifications .....	11
2.1 Introduction .....	12
2.1.1 Purpose .....	12
2.1.2 Document Conventions .....	12
2.1.3 Intended Audience and Reading Suggestions .....	13
2.1.4 Product Scope.....	13
2.1.5 References.....	13
2.2 Overall Description .....	14
2.2.1 Product Perspective .....	14
2.2.2 Product Functions.....	14
2.2.3 User Classes and Characteristics .....	15
2.2.4 Operating Environment .....	16
2.2.5 Design and Implementation Constraints.....	16
OS: Mobile must have android system to run the application. ....	16
2.2.6 User Documentation.....	16
2.2.7 Assumptions and Dependencies.....	16
2.3 External Interface Requirements .....	16
2.3.1 User Interfaces .....	16
2.3.2 Hardware Interfaces.....	18
2.3.3 Software Interfaces.....	18
2.3.4 Communications Interfaces .....	18
2.4 System Features .....	18
2.4.1 User Registration and User login .....	18
2.4.1.1 Description and Priority .....	18
2.4.1.2 Stimulus/Response Sequences.....	18

2.4.1.3	Functional Requirements.....	19
2.4.2	Add/Remove Product .....	19
2.4.2.1	Description and Priority .....	19
2.4.2.2	Stimulus/Response Sequences.....	19
2.4.2.3	Functional Requirements.....	19
2.4.3	Get Appointment from Doctor .....	21
2.4.3.1	Description and Priority .....	21
2.4.3.2	Stimulus/Response Sequences.....	21
2.4.3.3	Functional Requirements.....	21
2.4.4	Bird's feed related information .....	21
2.4.4.1	Description and Priority .....	21
2.4.4.2	Stimulus/Response Sequences.....	21
2.4.4.3	Functional Requirements.....	22
2.4.5	Chatting room for bargaining.....	22
2.4.5.1	Description and Priority .....	22
2.4.5.2	Stimulus/Response Sequences.....	22
2.4.5.3	Functional Requirements.....	22
2.5	Other Nonfunctional Requirements .....	22
2.5.1	Performance Requirements .....	22
2.5.2	Safety Requirements.....	22
2.5.3	Security Requirements.....	23
2.5.4	Software Quality Attributes .....	23
2.5.5	Business Rules .....	23
2.6	Other Requirements .....	24
Chapter 3	.....	25
Use Case Analysis	.....	25
3.1.	Use Case Model.....	26
3.2.	Fully Dressed Use Cases .....	29
Chapter 4	.....	42
System Design	.....	42
4.1.	Architecture Diagram.....	43
4.2.	Domain Model.....	44
4.3.	Entity Relationship Diagram with data dictionary .....	45
4.4.	Class Diagram.....	49
4.5.	Sequence / Collaboration Diagram .....	50
4.6.	Operation contracts .....	61
4.7.	Activity Diagram .....	65
Full activity:	.....	65
4.7.	State Transition Diagram.....	73
4.8.	Component Diagram .....	74
4.9.	Deployment Diagram.....	75
Chapter 5	.....	76
Implementation	.....	76
5.1.	Important Flow Control/Pseudo codes.....	77
5.2.	Components, Libraries, Web Services and stubs.....	82
5.3.	Deployment Environment .....	82

5.4. Tools and Techniques .....	83
5.5. Best Practices / Coding Standards .....	83
Chapter 6 .....	85
Testing and Evaluation.....	85
6.1. Use Case Testing .....	86
6.2. Equivalence partitioning .....	94
6.3. Boundary value analysis.....	95
6.4. Unit testing .....	96
6.5. Integration testing .....	105
6.6. Performance testing.....	105
6.7. Stress Testing.....	105
Chapter 7 .....	106
Summary, Conclusion and Future Enhancements .....	106
7.1. Project Summary.....	107
7.2. Achievements and Improvements.....	107
7.3. Critical Review .....	107
7.4. Lessons Learnt .....	108
7.5. Future Enhancements/Recommendations .....	108
Appendices .....	109
Appendix A: User Manual.....	110
Appendix C: Information / Promotional Material .....	110
Reference and Bibliography .....	113

## List of Figures

fig 1. 1 Work Breakdown Structure .....	5
fig 1. 2 Gantt Chart .....	9
fig 2. 1 Application Screen Shots .....	17
fig 3.1. 1 User Usecase Model.....	26
fig 3.1. 2 Seller Usecase Model .....	27
fig 3.1. 3 Admin Usecase Model .....	28
fig 4. 1 Architecture design.....	43
fig 4. 2 Domain Model .....	44
fig 4. 3 ERD Diagram .....	45
fig 4. 4 Class Diagram .....	49
fig 4. 5 Sequence Diagram .....	50
fig 4. 6 Activity Diagram .....	65
fig 4. 7 State Diagram .....	73
fig 4. 8 Component Diagram.....	74
fig 4. 9 Deployment Diagram .....	75



# Chapter 1

## **Introduction**

# Chapter 1: Introduction

E commerce has shifted all the business from traditional mode to virtual stores. Users can find everything available on E-stores while sitting in their homes. Pets are really popular in people all over the world. In Pakistan birds are quite popular as pet. Few application available over internet to sell birds but information like bird's feed, bird's disease related recommendations, isn't available at a single platform. Our proposed solution will provide everything for user on a single platform.

## 1.1. Background

Many people who love birds and want to sale or purchase their birds didn't have any proper online application for selling or purchasing the birds. They have to go to market and some time they don't want their desired birds due to high price. According to my survey there are many groups on Facebook for selling or purchasing birds but these groups do not provide satisfaction to customers. As such no proper website or application exists for people, which can entertain them with fully satisfaction of pricing, selling or purchasing the birds, according to my whole survey. Many birds also die due to weather diseases and people don't find the doctor recommendation for their birds online.

## 1.2. Motivations and Challenges

### 1.2.1. Motivations

Now a day everyone is lazy. No-one wants to go to the market and buy items. So we find it as opportunity and decided to choose this project in which we are going to develop an app through which people will be able to sell or purchase products, and it seems that we have bright future by this project.

### 1.2.2. Challenges

In 2017 there was an app launch with the name "Parenda Bazar" which is an selling or purchasing app for birds but this app was not gave complete satisfaction to their customer because they did not have unique feature of purchasing or selling app which become the

way of destruction of customer and the app fail due to don't have unique feature which attract customer towards them.

### **1.3. Goals and Objectives**

#### **1.3.1. Goals**

Our goal is that we want make an application with name **"Bird Lovers"** for the people who love birds. Basically, this is an online birds purchasing and selling app. This app helps the people to sale their birds online with their desire rates instead of go to shop and bargaining with shop boy. We also provide services of doctor for the species affected by any disease. People will also search or explore about related specie feed according to market rates. We will also advertise the birds shop on this app for business purposes. This app will provide online help to **"Bird Lovers"** to check the market rate strategies. We will help those people who get irritation when they have to bargaining with bird shop boy and they don't sale or purchase their birds on their desire rates. We will also add chat box feature for our customer. People want this application because such proper app does not exist in the market for selling or purchasing birds. This is an amazing opportunity for bird lovers to sell or purchase birds with complete satisfaction. Users will be able to find doctor's recommendation on this app. The information of feed of every bird will also be available on this app.

#### **1.3.2. Objectives**

Users can sale or purchase birds easily. The main objective of this platform is that, we are creating easiness for beginners who want to purchase or sale birds by giving them complete knowledge of birds. The other one objective is to entertain users with the desired features which they want in an application.

### **1.4. Literature Review/Existing Solutions**

There are already many websites which are providing the service of selling and purchasing products like **"OLX.com"** and many mobile applications are also available for this purpose like

“PRINDA BAZAR” but those websites are being used for other products selling and purchasing as well and mobile apps are not satisfying the customer needs completely, so there is really need of an updated platform which can satisfy the customers by providing all required features and some extra features as well.

## 1.5. Gap Analysis

Different websites and mobile applications are available for selling and purchasing product but there is not even a single web or mobile app which is properly for birds selling and purchasing and also providing the feed information for the birds and recommendations about the bird’s diseases. So this is the gap which exists in the current available systems (web and mobile apps).

## 1.6. Proposed Solution

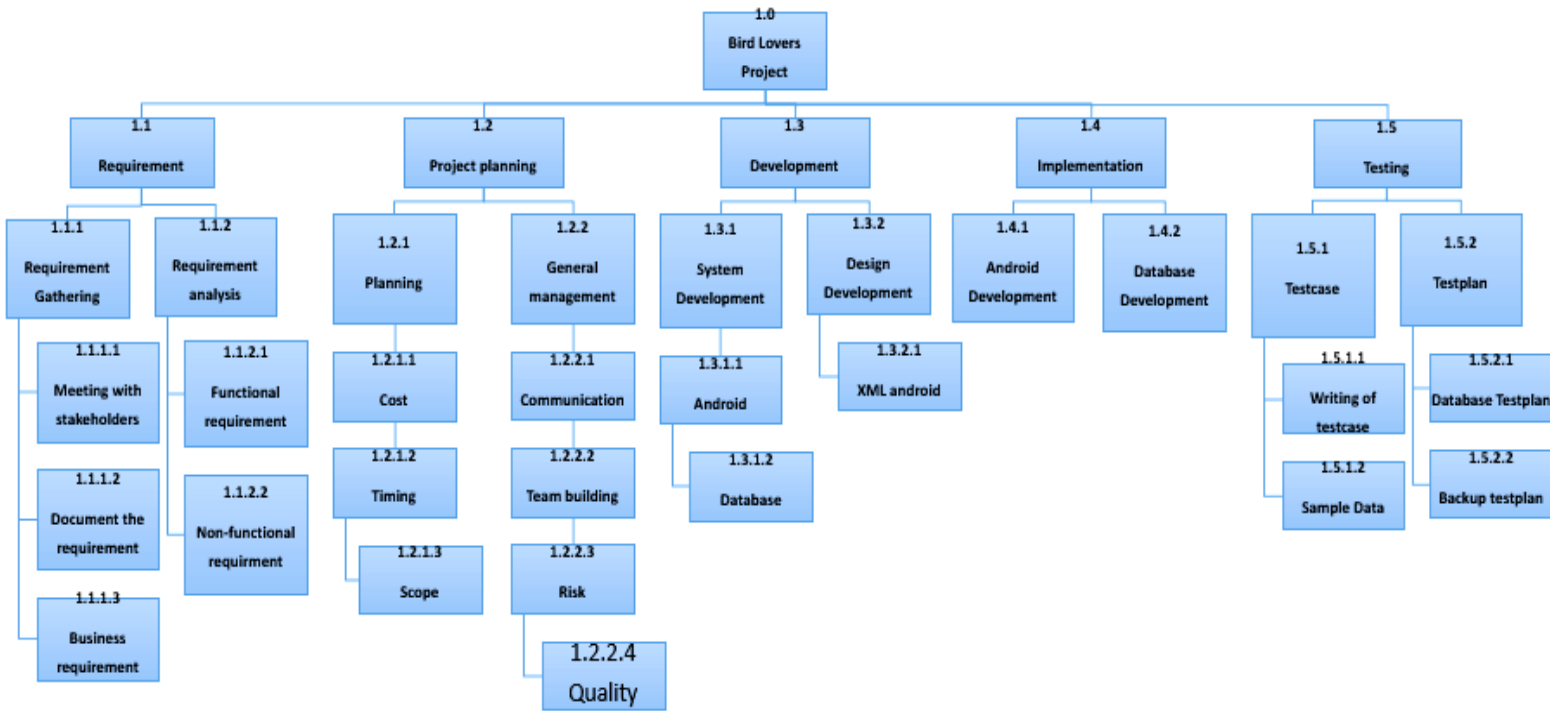
- We make an App “BIRD LOVERS” because in Pakistan many people love birds. You will sell or purchase any kind of bird according to the bird category. You will also compare market prices and also bargaining with seller in chat box.
- Doctor recommendation feature is also added in our application and customer will also get appointment with doctor through this app.
- Feed information about any bird is also added in this app.
- You will be able to buy feed of desired specie through this app.
- We are providing chatroom for the users to make bargaining with the owner of the bird.

## 1.7. Project Plan

The Project plan is as follow:

### 1.7.1. Work Breakdown Structure

fig 1. 1Work Breakdown Structure



### 1.6.1. Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Project Plan	1	<ul style="list-style-type: none"> <li>• Problem Statement</li> <li>• requirements</li> <li>• cost</li> <li>• scope</li> <li>• timing schedule</li> </ul>	07	<ul style="list-style-type: none"> <li>• Moazam</li> <li>• Usman</li> <li>• Rabees</li> </ul>
2	Requirements and Specifications	2	<ul style="list-style-type: none"> <li>• Requirement Gathering</li> <li>• Requirement Analysis</li> </ul>	21	<ul style="list-style-type: none"> <li>• Moazam</li> <li>• Usman</li> <li>• Rabees</li> </ul>
3	Architecture Diagram	3	<ul style="list-style-type: none"> <li>• Architecture Design</li> </ul>	2	<ul style="list-style-type: none"> <li>• Moazam</li> </ul>
4	Use Case Diagram	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	2	<ul style="list-style-type: none"> <li>• Usman</li> <li>• Rabees</li> </ul>
5	Fully Dress Use Case	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	5	<ul style="list-style-type: none"> <li>• Usman</li> <li>• Moazam</li> <li>• Rabees</li> </ul>
6	Domain Model	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	2	<ul style="list-style-type: none"> <li>• Usman</li> <li>• Rabees</li> </ul>
7	Class Diagram	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	3	<ul style="list-style-type: none"> <li>• Moazam</li> <li>• Usman</li> </ul>
8	Operation Contracts	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	4	<ul style="list-style-type: none"> <li>• Usman</li> <li>• Rabees</li> <li>• Moazam</li> </ul>
9	Sequence Diagram	3	<ul style="list-style-type: none"> <li>• UML Design</li> </ul>	4	<ul style="list-style-type: none"> <li>• Moazam</li> <li>• Rabees</li> </ul>

10	Activity Diagram	3	<ul style="list-style-type: none"> <li>UML Design</li> </ul>	3	<ul style="list-style-type: none"> <li>Usman</li> <li>Moazam</li> </ul>
11	State Machine Diagram	3	<ul style="list-style-type: none"> <li>UML Design</li> </ul>	3	<ul style="list-style-type: none"> <li>Moazam</li> <li>Usman</li> </ul>
12	Entity Relationship Diagram	3	<ul style="list-style-type: none"> <li>Database Design</li> </ul>	3	<ul style="list-style-type: none"> <li>Moazam</li> </ul>
13	Software Design Document	3	<ul style="list-style-type: none"> <li>Architecture Design</li> <li>UML Design</li> </ul>	25	<ul style="list-style-type: none"> <li>Moazam</li> </ul>
13	Photo shop designs	3	<ul style="list-style-type: none"> <li>User Interface Design</li> </ul>	8	<ul style="list-style-type: none"> <li>Moazam</li> <li>Usman</li> </ul>
14	XML Layout	3	<ul style="list-style-type: none"> <li>User Interface Design</li> </ul>	4	<ul style="list-style-type: none"> <li>Moazam</li> </ul>
16	Bird Lovers App	4 5 6	<ul style="list-style-type: none"> <li>Implementation</li> <li>Testing</li> <li>Deployment</li> </ul>	150	<ul style="list-style-type: none"> <li>Moazam</li> <li>Usman</li> <li>Rabees</li> </ul>

### 1.6.2. Gantt Chart

2.	Task	Start	Finish	Duration
	Project Plan	27-Sep	4-Oct	7

Requirements and Specifications	4-Oct	25-Oct	21
Architecture Diagram	25-Oct	27-Oct	2
Use Case Diagram	27-Oct	29-Oct	2
Fully Dress Use Case	29-Oct	3-Nov	5
Domain Model	3-Nov	5-Nov	2
Class Diagram	5-Nov	8-Nov	3
Operation Contracts	8-Nov	12-Nov	4
Sequence Diagram	12-Nov	16-Nov	4
Activity Diagram	16-Nov	19-Nov	3
State Machine Diagram	19-Nov	22-Nov	3
Entity Relationship Diagram	22-Nov	25-Nov	3
Software Design Document	25-Nov	20-Dec	25
Photo shop designs	20-Dec	28-Dec	8
XML Layout	28-Dec	1-Jan	4
Implementation	1-Jan	20-Feb	50
Testing	20-Feb	11-Apr	50
Deployment	11-Apr	1-May	20

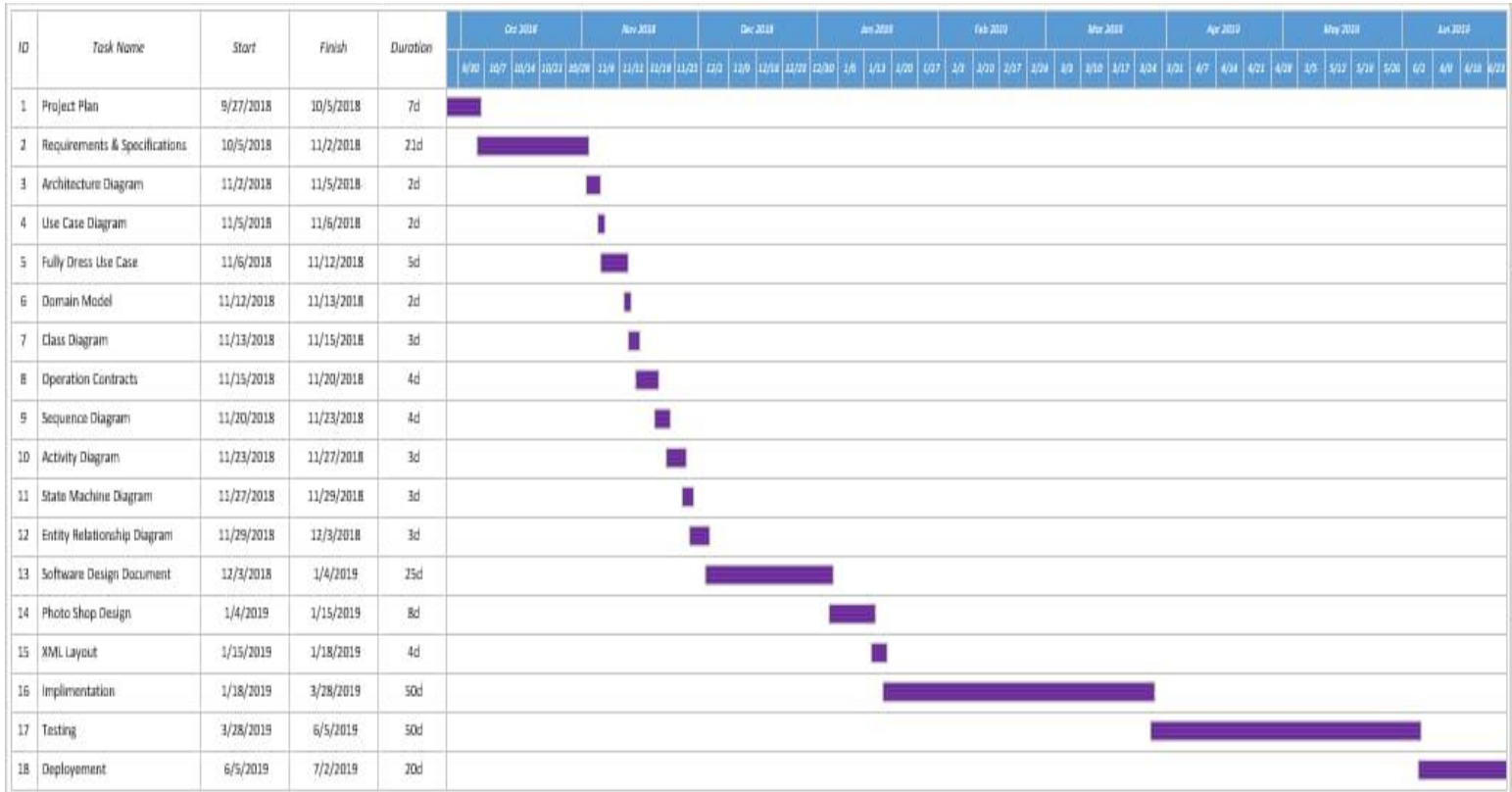


fig 1. 2 Gantt Chart

## 2.1. Report Outline

3. Introduction
4. Background
5. Motivations and challenges
6. Goals and Objectives
7. Literature Review/Existing Solutions
8. Gap Analysis
9. Proposed Solution
10. Project Plan
  - 10.1. Work Breakdown Structure
  - 10.2. Roles & Responsibility Matrix
  - 10.3. Gantt chart



# Chapter 2

## **Software Requirement Specifications**

## Chapter 2: Software Requirement Specifications

### 2.1 Introduction

#### 2.1.1 Purpose

Many people who love birds and want to sale or purchase their birds didn't have any proper online application for selling or purchasing the birds. They have to go to market and some time they don't want their desired birds due to high price. According to my survey there are many groups on Facebook for selling or purchasing birds but these groups do not provide satisfaction to customers. As such no proper website or application exists for people, which can entertain them with fully satisfaction of pricing, selling or purchasing the birds, according to my whole survey. Many birds also die due to weather diseases and people don't find the doctor recommendation for their birds online. So, our purpose is to provide these all features to the users at a single platform.

#### 2.1.2 Document Conventions

##### *Main Section Titles*

- Font: Calibri
- Face: Bold
- Size: 14

##### *Sub Section Titles*

- Font: Calibri
- Face: Bold
- Size: 12

##### *Other Text Explanations*

- Font: Calibri
- Face: Normal
- Size: 12

### 2.1.3 Intended Audience and Reading Suggestions

This SRS document is intended for:

- **Project Developers:**

By viewing this SRS, developer will be able to understand all the areas of the application and can easily check at which area some improvement is needed.

- **Project Testers:**

By viewing this SRS, project testers will be able to easily find the bugs in the system because requirements document is really helpful to find the bugs.

- **Application Users:**

There are such users who always interested to read about the functions of the system, so by viewing this SRS, users will be able to check which functions are being delivered through this application.

### 2.1.4 Product Scope

Bird lovers is the e-commerce system with some social media features for people to buy or sale birds of different categories. This system is available on internet in form of android application. This app provides the features to purchase or sale birds. User can sale or purchase any bird, category wise. User can also chat with other person for bargaining. You can also get doctor recommendation or appointment in any case of disease if your bird has. You can also search for feed information for your desired birds. Advertisement feature of local shops for selling or purchasing birds and feeds are also available. The objective of this system is that users can buy and sell birds of different categories in comparatively market price with satisfied features. Customer will first create his/her account, then access the provided features.

### 2.1.5 References

3 Website UI design Guide:

4 [1]<https://uxplanet.org/golden-rules-of-user-interface-design-19282aeb06b>

- 5 [2][https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-](https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb)
- 6 [rules-of-thumb](https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb)
- 7 Android App UI design Guide:
- 8 [3] [https://developer.android.com/guide/practices/ui\\_guidelines/](https://developer.android.com/guide/practices/ui_guidelines/)
- 9 [4] <https://guides.codepath.com/android/Android-Design-Guidelines>
- 10 Standards:
- 11 [5] <https://www.inflectra.com/ideas/topic/requirements-definition.aspx>
- 12 Use case Document
- 13 [6][http://www.michaelrichardson.com/processes/rup\\_classic/core.base\\_rup/guidances/guidelines/software\\_requirements\\_specification\\_91456CD0.html](http://www.michaelrichardson.com/processes/rup_classic/core.base_rup/guidances/guidelines/software_requirements_specification_91456CD0.html)

## 2.2 Overall Description

### 2.2.1 Product Perspective

Bird Lovers system exists in the form of android application. Bird Lovers provide the feature to users that he will be able to purchase or sale their birds with complete satisfaction. Bird Lovers provides the ecommerce functionality that user can add, remove, update his products, buy and sell products. All the data will store on **Firestore Database**. Customer will be satisfied with the desired features on this app like you can get appointment from doctor through emailing process and you can get information about bird's feed.

### 2.2.2 Product Functions

Bird Lovers provides the functionality to users that they can add, update, delete products.

Provided features are:

#### For Customers

- Register Account
- Add product
- Update product
- Delete product
- View product
- Chatting
- Doctor recommendation

- Doctor appointment

#### **Admin**

- Manage User Accounts
- Manage products
- Manage feed information
- Manage Doctor Recommendations
- Manage Ads

#### **All Users**

- Login
- Logout

### **2.2.3 User Classes and Characteristics**

Bird lovers exists in different types of classes which are as follow:

- **General User:**

In this class, user will create his/her account to perform other functions like view birds based on categories, add, delete, update product.

- **Birds Seller:**

In this class, user will be able to post his/her bird's ad to sell. Later on he/she can update bird's picture or price.

- **Birds Purchaser:**

In this class, user will view the bird's picture and other details and then he/she can contact to the owner to purchase particular bird.

- **Doctor:**

In this class, doctor will provide feed information about the birds and will provide recommendations about the bird's diseases to the admin.

- **Administrator:**

Administrator login the account. Admin can manage the user accounts. He will also manage the doctor recommendation. He will also manage the products and ads.

## 2.2.4 Operating Environment

User can use this app on android operating system. This app will run on android kit Kat versions 4.4 to onwards later versions.

## 2.2.5 Design and Implementation Constraints

**OS:** Mobile must have android system to run the application.

**DB:** Centralized database will be involved, online database will be used.

**Internet:** User's mobile must have to access to internet to use this application.

**Account:** User must have account to perform other activities (like view items, add product).

## 2.2.6 User Documentation

Bird lovers will provide the videos that how you can use this application. These videos will help the users "How we can use the Bird lovers?" If something will not perform perfectly then user send the report to admin. This is very helpful to users for solving the problems. We also provide screen shots for complete understanding.

## 2.2.7 Assumptions and Dependencies

Assumptions and dependencies are as following:

Our system is very simple and online based, so there is no need of any kind of assumptions and dependencies. Just we are assuming that user knows how to use mobile phones and mobile applications. No dependency is needed for this application.

## 2.3 External Interface Requirements

### 2.3.1 User Interfaces

There will be following interfaces:

- Login
- Signup

- Home page containing different products
- If user clicks on any item, a new page will be open providing all information about the product.
- A page containing feed information about the birds.
- A page for user to get appointment from the doctor
- Chat room for buyer and seller to make bargaining
- A page for admin to add/remove product.

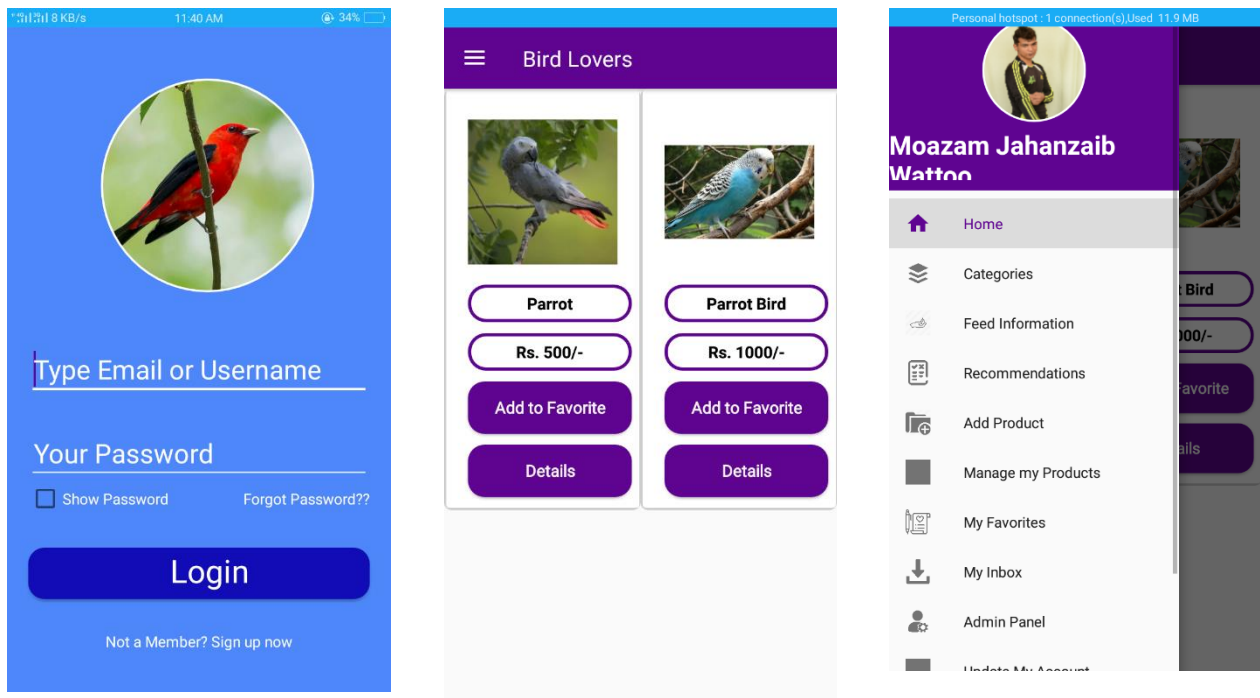


fig 2. 1 Application Screen Shots

### **2.3.2 Hardware Interfaces**

As mobile application doesn't have any designed hardware, so this application doesn't have any hardware interface. A user needs only an android phone to use this application.

### **2.3.3 Software Interfaces**

This application is not using any third-party software. There will be only Firebase database for storing the data.

### **2.3.4 Communications Interfaces**

As it is an online application, so http protocol will be used for the communication over internet.

## **2.4 System Features**

These are some basic features of our mobile application

- User registration and User Login
- Add/Remove product
- Get appointment from doctor
- Birds feed related information
- Chatting room for bargaining

### **2.4.1 User Registration and User login**

#### **2.4.1.1 Description and Priority**

In this feature, the user will be able to be get registered to make view or sell products. Registration for the users is very important so priority of this feature is very high.

#### **2.4.1.2 Stimulus/Response Sequences**

Stimulus: User want make an account

Response: System will save all the information of the user and display home page to the user

Stimulus: User want to login his/her account

Response: System will match the login details and if data matches then home page will be display.

### 2.4.1.3 Functional Requirements

REQ-SF1-1: For registration, user will have to click on “**Don’t have account**”, then registration page will open.

REQ-SF1-2: Then user has to fill the form and click on signup button.

REQ-SF1-3: For login, user must have account.

REQ-SF1-4: User will have to fill the both fields on login page according to his/her account. Then he/she will have to click on login button...

## 2.4.2 Add/Remove Product

### 2.4.2.1 Description and Priority

In this feature, the user (seller) will be able to add products or remove products from the displaying items. This feature also has high priority.

### 2.4.2.2 Stimulus/Response Sequences

Stimulus: user (seller) want add product

Response: System will display message confirming the addition of the product.

Stimulus: user (seller) want remove product

Response: System will display message confirming the removal of the product.

Stimulus: user (seller) want update the product

Response: System will display message confirming that the product has been updated.

### 2.4.2.3 Functional Requirements

REQ-SF2-1: user (seller) will have to click on “**Add Item**” button which will be available in menu bar.

REQ-SF2-2: Then user (seller) will have to fill all the fields and then he/she will have to click on “**Add**” button to add item.

REQ-SF2-3: user (seller) will have to click on “**Remove Item**” button which will be available in menu bar.

REQ-SF2-4: Then user (seller) will have to define the id of the product and then he/she will have to click on “**Remove**” button to add item.

REQ-SF2-4: user (seller) will have to click on “**Update an Item**” button will be available in menu bar.

REQ-SF2-5: Then user (seller) will have to fill all the fields and then he/she will have to click on “**Update**” button to add item.

## 2.4.3 Get Appointment from Doctor

### 2.4.3.1 Description and Priority

In this feature, the user will be able to get an appointment from the available doctors. A mail will be generated from the user side, and if the doctor is available, then they both will be able to communicate with each other.

### 2.4.3.2 Stimulus/Response Sequences

Stimulus: User wants to make an appointment with doctor.

Response: System will send email to the doctor and doctor's reply will be received to the user in his email inbox.

### 2.4.3.3 Functional Requirements

REQ-SF4-1: User have to click "**Feed Info**" Button to view the feed information, there will be info available of the doctor who recommended that feed.

REQ-SF4-2: Then user will have to click on "**Get Appointment**" Button to get the appointment from the doctor. A mail will be sent to the doctor by that particular user's email address.

## 2.4.4 Bird's feed related information

### 2.4.4.1 Description and Priority

In this feature, the admin will be able to add information about the feed which should be given to the birds, and user will be able to view this all information.

### 2.4.4.2 Stimulus/Response Sequences

Stimulus: Admin wants to add information related to bird's feed.

Response: System will update the database and display message confirming the addition of the information.

Stimulus: User wants to view feed related information.

Response: System will fetch the feed information of specific bird and display to the user.

#### **2.4.4.3. Functional Requirements**

REQ-SF5-1: User will have to click on “Feed Info” Button to view the feed information related page.

REQ-SF5-1: User will have to click on **particular bird** about which he/she is interested to get the information related to feed.

### **2.4.5 Chatting room for bargaining**

#### **2.4.5.1 Description and Priority**

In this feature, the user will be able to chat with the owner of the bird to make bargaining, so that the price of the bird can be compromised.

#### **2.4.5.2 Stimulus/Response Sequences**

Stimulus: User wants chat with the owner of bird.

Response: System will create chat box for both (user and owner).

#### **2.4.5.3 Functional Requirements**

REQ-SF6-1: the user who wants buy product, he/she will have to click on the “Chat” button which will be available with the product which is available to be sold

REQ-SF6-2: A chat room will be created against the product id, buyer’s id and seller’s id.

## **2.5 Other Nonfunctional Requirements**

### **2.5.1 Performance Requirements**

The system must be interactive and in any case delay time must be minor .So in every functional response, there should be no instant delays. During the opening of any activity should not be any type of crash or mishap. During storing information in databases, database shouldn’t be crash.

### **2.5.2 Safety Requirements**

Information transmission should be securely transmitted to server without any changes in information

### **2.5.3 Security Requirements**

The main security concern is for users account hence proper login mechanism should be used to avoid hacking.

As we are not collecting any confidential data from the users, so I don't think so that we need high security for this application. But still the data will be kept in form of encryption to avoid any kind of threat.

### **2.5.4 Software Quality Attributes**

The software quality attributes for our website are

1. Correctness

Before the deployment, all the functions of the application will be tested properly and app will be deployed only in case if all the functions of the app will be working correctly.

2. Usability

We have designed very simple interface of our application, so every user can use easily our app without any kind of interruption.

3. Testability

We have divided our application in different modules, so it's very easy for the testers to test the complete application one by one all the modules.

4. Maintainability

As we mentioned that our app is divided in multiple modules, so maintenance of the app is also easy. Any part of the app can be modified in case of any need.

5. Reliability

Our application is well designed, and all it's functions are working properly. So every user can rely on this app. In case of any problem or failure, the application will maintain the record without any kind of loss.

### **2.5.5 Business Rules**

There are some business rules here like

- First business rule will be that our team will deliver the products but delivery charges will be charged from the customer.
- If our team delivered the product, then the product will be non-returnable and the customer will have to pay the charges on delivery in cash form.
- Price of the product will not be fixed. Seller and Purchaser can set their desired price.

## **2.6 Other Requirements**

No other requirements are needed.

# Chapter 3

## Use Case Analysis

## Chapter 3: System Analysis

In this chapter we will see the system working through use case and fully dressed use case.

### 3.1. Use Case Model

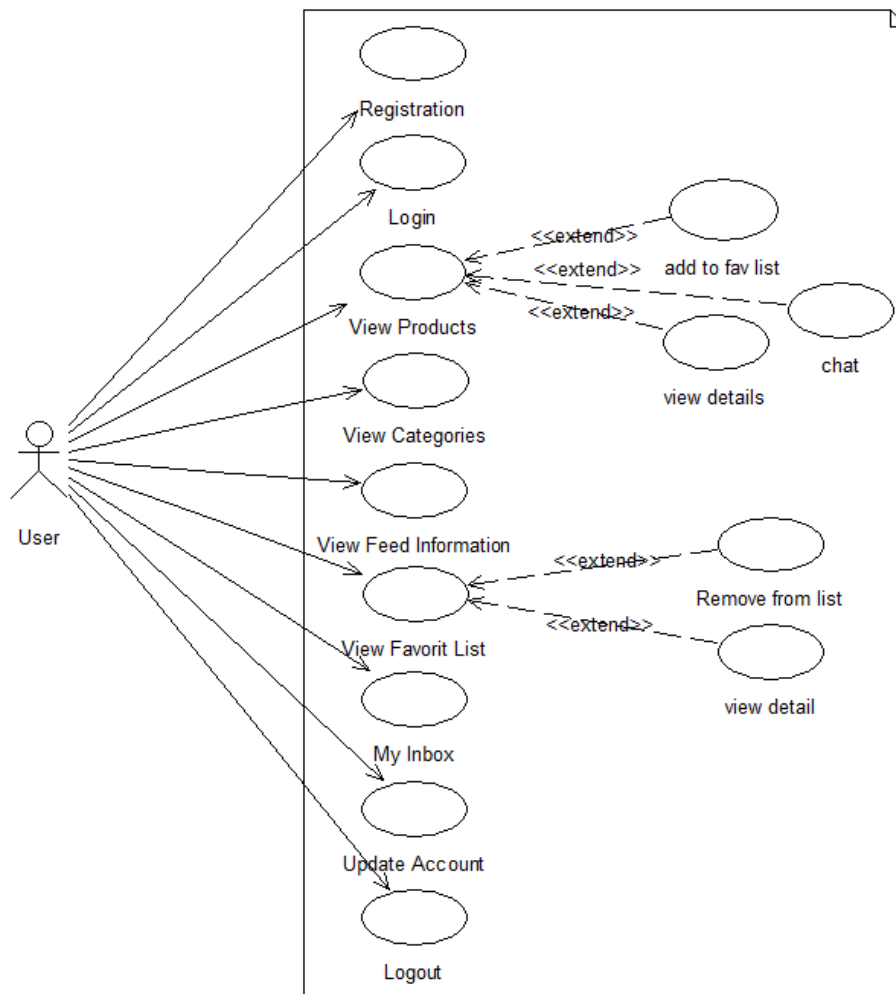


fig 3.1. 1 User Usecase Model

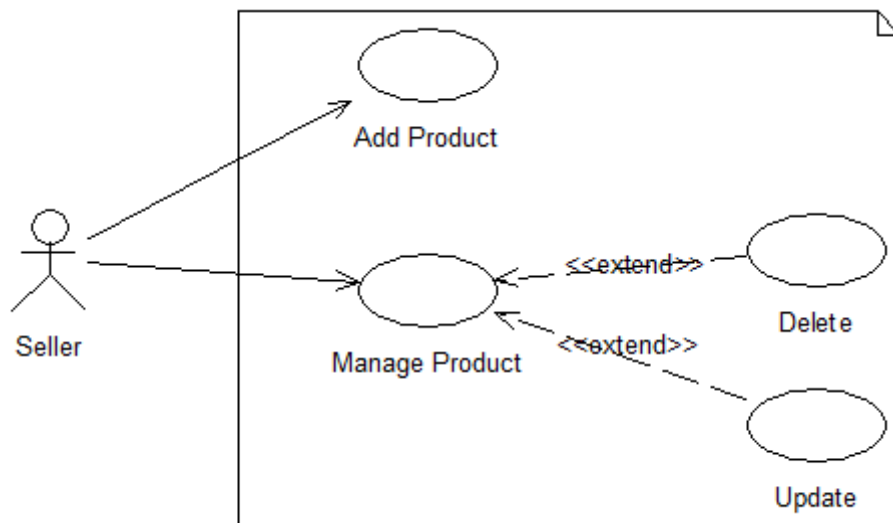


fig 3.1. 2 Seller Usecase Model

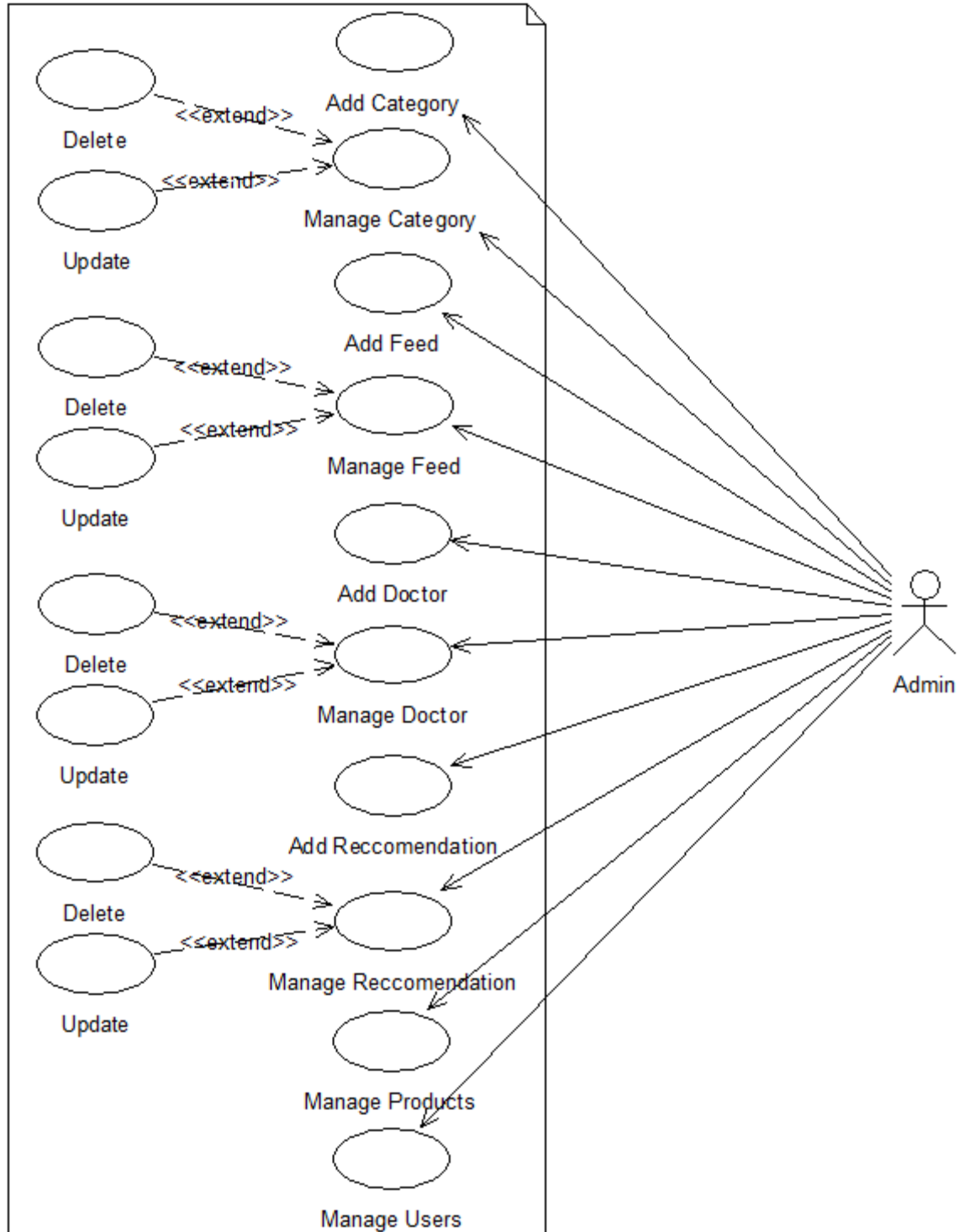


fig 3.1.3 Admin Usecase Model

### 3.2. Fully Dressed Use Cases

Use Case 1: Sign up	
<b>Use Case Name</b>	Sign Up
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. Login page will be visible to the user. User will click on signup button to move on Sign Up Page. Now User will fill all the fields and then submit the data. In this way the user will be registered.
<b>Pre-Condition(s)</b>	1. User must have internet connection.
<b>Post-Condition(s)</b>	1. On successful registration, user will be diverted towards login page.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User open the app.</li> <li>2. System will display login page to the user.</li> <li>3. User will tap on <b>Sign Up</b>.</li> <li>4. System will display the Sign-Up Page.</li> <li>5. User will fill all the fields and submit data.</li> <li>6. System will display the message of successful registration.</li> <li>7. System will divert the user towards login page.</li> </ol>
<b>Alternative Workflow</b>	<p><b>6a. If user not fill all the fields,</b></p> <ol style="list-style-type: none"> <li>1. The system will display the message to fill all the fields.</li> </ol> <p><b>6b. If username is already taken and now not available for this user,</b></p> <ol style="list-style-type: none"> <li>1. The system will display the message to change the username.</li> </ol> <p><b>6c. If username or password is less than 3 characters,</b></p> <ol style="list-style-type: none"> <li>1. The system will display the message to increase no. of characters in username or password.</li> </ol>

Use Case 2: Login	
<b>Use Case Name</b>	Login
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. Login page will be visible to the user. User will enter his username and password. Then user will by diverted towards the homepage.
<b>Pre-Condition(s)</b>	1. User must have internet connection.
<b>Post-Condition(s)</b>	1. On successful login, user will be diverted towards home page.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User open the app.</li> <li>2. System will display login page to the user.</li> <li>3. User will enter his username and password.</li> <li>4. System will divert the user towards home page.</li> </ol>
<b>Alternative Workflow</b>	<p><b>4a. If user not fill both fields,</b></p> <ol style="list-style-type: none"> <li>1. The system will display the message to fill the both fields.</li> </ol> <p><b>4b. If username or password doesn't match,</b></p>

	1. The system will display the message that username or password is not matching, and will ask to check again the username and password
--	---

Use Case 3: Sig Out	
<b>Use Case Name</b>	Sign Out
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>Sign Out</b> option from the navigation drawer. System will divert the user towards the login page.
<b>Pre-Condition(s)</b>	1. User must login to his account.
<b>Post-Condition(s)</b>	
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Sign Out</b>.</li> <li>4. System will divert the user towards the Login page.</li> </ol>
<b>Alternative Workflow</b>	-

Use Case 4: View Products	
<b>Use Case Name</b>	View Products
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page user can view some of the products. And user can also view all the products falling inside a particular category.
<b>Pre-Condition(s)</b>	1. User must login to his account.
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If user tap on <b>add to favorite</b> button, that particular product will be added to user's favorite list.</li> <li>2. If user tap on <b>detail</b> button, the user will be diverted towards the page containing complete detail regarding that particular product.</li> <li>3. If user tap on chat button, the user will be diverted towards the chat room for bargaining with the product's owner.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. Homepage will be containing products. User will be able to view all the products.</li> <li>4. User will be able to view all the products falling inside a specific category through category section.</li> <li>5. User tap on <b>add to favorite</b> button.</li> <li>6. That particular product will be added to user's favorite list.</li> <li>7. User tap on <b>detail</b> button.</li> </ol>

	<ol style="list-style-type: none"> <li>8. The user will be diverted towards the page containing complete detail regarding that particular product.</li> <li>9. User tap on chat button.</li> <li>10. The user will be diverted towards the chat room for bargaining with the product's owner.</li> </ol>
<b>Alternative Workflow</b>	<p><b>6a. If user try to add his own product to his favorite list,</b></p> <ol style="list-style-type: none"> <li>1. System will display message that user can't add his own product to his favorite list</li> </ol> <p><b>10a. If user try to create chat room for his own product,</b></p> <ol style="list-style-type: none"> <li>2. The system will display the message that username or password is not matching, and will ask to check again the username and password</li> </ol>

### Use Case 5: Categories

<b>Use Case Name</b>	Categories
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>categories</b> option from the navigation drawer. All the available categories will display to the user. Then user can select specific category to view products regarding that specific category.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. User must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If user select specific category, all products regarding that specific category will be display to the user.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Categories</b>.</li> <li>4. System will display all the available categories</li> <li>5. User will select the specific category</li> <li>6. All the products falling under that category will be visible to the user.</li> </ol>
<b>Alternative Workflow</b>	

### Use Case 6: Feed Information

<b>Use Case Name</b>	Feed Information
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>feed information</b> option from the navigation drawer. All the available feed information for the birds will display to the user.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. User must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	

<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>feed information</b>.</li> <li>4. System will display all the available feed information for the birds to the user.</li> </ol>
<b>Alternative Workflow</b>	

### Use Case 7: My Favorites

<b>Use Case Name</b>	My Favorites
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>My Favorites</b> option from the navigation drawer. All the available categories will display to the user. Then user can select specific category to view products regarding that specific category.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. User must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If user tap on <b>Remove from favorite</b> button, that particular product will be removed from user's favorite list.</li> <li>2. If user tap on <b>detail</b> button, the user will be diverted towards the page containing complete detail regarding that particular product.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>My Favorite</b>.</li> <li>4. System will display all the user's marked favorite products.</li> <li>5. User will tap on <b>Remove from favorite</b> button.</li> <li>6. System will remove the product from user's favorite list.</li> <li>7. User tap on <b>detail</b> button.</li> <li>8. The user will be diverted towards the page containing complete detail regarding that particular product.</li> </ol>
<b>Alternative Workflow</b>	

### Use Case 8: Recommendations

<b>Use Case Name</b>	Recommendations
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>Recommendations</b> option from the navigation drawer. All the available Recommendations will display to the user.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. User must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	

<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Recommendations</b>.</li> <li>4. System will display all the available Recommendations</li> </ol>
<b>Alternative Workflow</b>	

<b>Use Case 9: Manage My Products</b>	
<b>Use Case Name</b>	Manage My Products
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>Manage My Products</b> option from the navigation drawer. All the user's product will display. Now User can update or delete any of his products.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. User must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If user tap on <b>update</b> button, system will divert the user towards the page where user can update his product.</li> <li>2. If user tap on <b>delete</b> button, system will delete the user's product.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Manage My Products</b>.</li> <li>4. System will display all the user's products.</li> <li>3. User tap on <b>update</b> button.</li> <li>4. System will divert the user towards the page where user can update his product.</li> <li>5. User tap on <b>delete</b> button.</li> <li>6. System will delete the user's product.</li> </ol>
<b>Alternative Workflow</b>	

<b>Use Case 10: Add Product</b>	
<b>Use Case Name</b>	Add Product
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>Add Product</b> option from the navigation drawer. System will open a new page, where user can add new product to sell.

<b>Pre-Condition(s)</b>	1. User must login to his account.
<b>Post-Condition(s)</b>	1. On product added, System will display the message about successful addition of the product.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Add Product</b>.</li> <li>4. System will open a new page, where user can add new product to sell.</li> <li>5. User fill all the fields about the bird's detail.</li> <li>6. After filling the fields, user will submit the data.</li> <li>7. System will display the message about successful addition of the product</li> </ol>
<b>Alternative Workflow</b>	<b>6a. If user not fill all the fields,</b> <ol style="list-style-type: none"> <li>1. The system will display the message to fill all the fields.</li> </ol>

#### Use Case 11: My Inbox

<b>Use Case Name</b>	My Inbox
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None
<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>My Inbox</b> option from the navigation drawer. System will display the all contacts list to the user to whom the user made chat.
<b>Pre-Condition(s)</b>	1. User must login to his account.
<b>Post-Condition(s)</b>	1. If user tap on anyone of the contact list, a chat room will open where the user can chat with the contact person.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>My Inbox</b>.</li> <li>4. System will display the all contacts list to the user to whom the user made chat.</li> <li>5. User tap on anyone of the contact list</li> <li>6. A chat room will open where the user can chat with the contact person.</li> </ol>
<b>Alternative Workflow</b>	-

#### Use Case 12: Update My Account

<b>Use Case Name</b>	Update My Account
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	None

<b>Description</b>	User will open the application. After Login to his account, at home page, user will tap on <b>Update My Account</b> option from the navigation drawer. System will divert the user towards the update page, where he will be able to update his account. Where user will change his data and then submit the updated data.
<b>Pre-Condition(s)</b>	1. User must login to his account.
<b>Post-Condition(s)</b>	1. On submit, the system will display the message that the data updated successfully.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. User will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, user will select the option <b>Update My Account</b>.</li> <li>4. System will divert the user towards the update page.</li> <li>5. System will display the user all his previous account information.</li> <li>6. User will change the data.</li> <li>7. Now user will submit the changed data.</li> <li>8. System will display the message about the successfully updated data.</li> </ol>
<b>Alternative Workflow</b>	-

### Use Case 13: Add Category

<b>Use Case Name</b>	Add Category
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Category</b> option from the navigation drawer. System will open a new page, where admin can add new category.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. On Category added, System will display the message about successful addition of the Category.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. Admin will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, admin will select the option <b>Add Category</b>.</li> <li>4. System will open a new page, where admin can add new Category.</li> <li>5. Admin fill all the fields about the Category detail.</li> <li>6. After filling the fields, admin will submit the data.</li> <li>7. System will display the message about successful addition of the Category</li> </ol>
<b>Alternative Workflow</b>	<b>7a. If admin not fill all the fields,</b> <ol style="list-style-type: none"> <li>1. The system will display the message to fill all the fields.</li> </ol>

Use Case 14: Manage Categories	
<b>Use Case Name</b>	Manage Categories
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Categories</b> option from the navigation drawer. All the available categories will display. Now admin can update or delete any of available categories.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. If admin tap on <b>update</b> button, system will divert the admin towards the page where admin can update particular category. 2. If admin tap on <b>delete</b> button, system will delete the particular category.
<b>Normal Workflow</b>	1. Admin will login to his account. 2. System will display the homepage. 3. From navigation drawer, admin will select the option <b>Manage Categories</b> . 4. System will display all the available categories. 5. Admin tap on <b>update</b> button. 6. System will divert the admin towards the page where admin can update his product. 7. Admin tap on <b>delete</b> button. 8. System will delete the particular category.
<b>Alternative Workflow</b>	

Use Case 15: Add Feed	
<b>Use Case Name</b>	Add Feed
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Feed</b> option from the navigation drawer. System will open a new page, where admin can add new Feed Information.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. On Feed added, System will display the message about successful addition of the Feed Information.
<b>Normal Workflow</b>	1. Admin will login to his account. 2. System will display the homepage. 3. From navigation drawer, admin will select the option <b>Add Feed</b> . 4. System will open a new page, where admin can add new Feed Information. 5. Admin fill all the fields about the Feed detail.

	<ol style="list-style-type: none"> <li>6. After filling the fields, admin will submit the data.</li> <li>7. System will display the message about successful addition of the Feed.</li> </ol>
<b>Alternative Workflow</b>	<b>7a. If admin not fill all the fields,</b> <ol style="list-style-type: none"> <li>1. The system will display the message to fill all the fields.</li> </ol>

<b>Use Case 16: Manage Feed</b>	
<b>Use Case Name</b>	Manage Feed
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Feed</b> option from the navigation drawer. All the available Feeds will display. Now admin can update or delete any of available feeds.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. Admin must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If admin tap on <b>update</b> button, system will divert the admin towards the page where admin can update particular Feed.</li> <li>2. If admin tap on <b>delete</b> button, system will delete the particular feed.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. Admin will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, admin will select the option <b>Manage Feed</b>.</li> <li>4. System will display all the available Feeds.</li> <li>5. Admin tap on <b>update</b> button.</li> <li>6. System will divert the admin towards the page where admin can update Feed Information.</li> <li>7. Admin tap on <b>delete</b> button.</li> <li>8. System will delete the particular Feed.</li> </ol>
<b>Alternative Workflow</b>	

<b>Use Case 17: Add Doctor</b>	
<b>Use Case Name</b>	Add Doctor
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Doctor</b> option from the navigation drawer. System will open a new page, where admin can add new Feed Information.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. Admin must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. On Doctor added, System will display the message about successful addition of the Doctor Information.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. Admin will login to his account.</li> </ol>

	<ol style="list-style-type: none"> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, admin will select the option <b>Add Doctor</b>.</li> <li>4. System will open a new page, where admin can add new Doctor Information.</li> <li>5. Admin fill all the fields about the Doctor detail.</li> <li>6. After filling the fields, admin will submit the data.</li> <li>7. System will display the message about successful addition of the Doctor.</li> </ol>
<b>Alternative Workflow</b>	<b>7a. If admin not fill all the fields,</b> <ol style="list-style-type: none"> <li>1. The system will display the message to fill all the fields.</li> </ol>

<b>Use Case 18: Manage Doctor</b>	
<b>Use Case Name</b>	Manage Doctor
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Doctor</b> option from the navigation drawer. All the available Doctors will display. Now admin can update or delete any of available doctors.
<b>Pre-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. Admin must login to his account.</li> </ol>
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>1. If admin tap on <b>update</b> button, system will divert the admin towards the page where admin can update particular Doctor.</li> <li>2. If admin tap on <b>delete</b> button, system will delete the particular doctor.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>1. Admin will login to his account.</li> <li>2. System will display the homepage.</li> <li>3. From navigation drawer, admin will select the option <b>Manage Doctor</b>.</li> <li>4. System will display all the available Doctors.</li> <li>5. Admin tap on <b>update</b> button.</li> <li>6. System will divert the admin towards the page where admin can update Doctor Information.</li> <li>7. Admin tap on <b>delete</b> button.</li> <li>8. System will delete the particular Doctor.</li> </ol>
<b>Alternative Workflow</b>	

<b>Use Case 19: Add Recommendation</b>	
<b>Use Case Name</b>	Add Recommendation
<b>Primary Actor</b>	Admin

<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Recommendation</b> option from the navigation drawer. System will open a new page, where admin can add new Recommendation Information.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. On Recommendation added, System will display the message about successful addition of the Recommendation Information.
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>Admin will login to his account.</li> <li>System will display the homepage.</li> <li>From navigation drawer, admin will select the option <b>Add Recommendation</b>.</li> <li>System will open a new page, where admin can add new Recommendation Information.</li> <li>Admin fill all the fields about the Recommendation detail.</li> <li>After filling the fields, admin will submit the data.</li> <li>System will display the message about successful addition of the Recommendation.</li> </ol>
<b>Alternative Workflow</b>	<b>7a. If admin not fill all the fields,</b> <ol style="list-style-type: none"> <li>The system will display the message to fill all the fields.</li> </ol>

### Use Case 20: Manage Recommendation

<b>Use Case Name</b>	Manage Recommendation
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Recommendation</b> option from the navigation drawer. All the available Recommendations will display. Now admin can update or delete any of available Recommendations.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	<ol style="list-style-type: none"> <li>If admin tap on <b>update</b> button, system will divert the admin towards the page where admin can update particular Recommendation.</li> <li>If admin tap on <b>delete</b> button, system will delete the particular Recommendation.</li> </ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"> <li>Admin will login to his account.</li> <li>System will display the homepage.</li> <li>From navigation drawer, admin will select the option <b>Manage Recommendation</b>.</li> <li>System will display all the available Recommendations.</li> <li>Admin tap on <b>update</b> button.</li> <li>System will divert the admin towards the page where admin can update Recommendation Information.</li> <li>Admin tap on <b>delete</b> button.</li> <li>System will delete the particular Recommendation.</li> </ol>

<b>Alternative Workflow</b>	
-----------------------------	--

<b>Use Case 21: Manage All Users</b>	
<b>Use Case Name</b>	Manage All Users
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, Admin will tap on <b>Manage All Users</b> option from the navigation drawer. All users list will display. Now Admin can update or delete any of the available users.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. If Admin tap on <b>Make Admin/Make Simple User</b> button, system will divert the Admin towards the page where Admin can update user's user type. 2. If Admin tap on <b>Remove User</b> button, system will remove that particular user.
<b>Normal Workflow</b>	1. Admin will login to his account. 2. System will display the homepage. 3. From navigation drawer, Admin will select the option <b>Manage All Users</b> . 4. System will display all the user's products. 5. Admin tap on <b>Make Admin/Make Simple User</b> button. 6. System will divert the Admin towards the page where Admin can update user's user type. 7. Admin tap on <b>Remove</b> button. 8. System will remove that particular user.
<b>Alternative Workflow</b>	

<b>Use Case 22: Manage All Products</b>	
<b>Use Case Name</b>	Manage All Products
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	None
<b>Description</b>	Admin will open the application. After Login to his account, at home page, Admin will tap on <b>Manage All Products</b> option from the navigation drawer. All the products will display. Now Admin can update or delete any of the products.
<b>Pre-Condition(s)</b>	1. Admin must login to his account.
<b>Post-Condition(s)</b>	1. If Admin tap on <b>update</b> button, system will divert the Admin towards the page where Admin can update product.

	<ol style="list-style-type: none"><li>2. If Admin tap on <b>delete</b> button, system will delete the particular product.</li></ol>
<b>Normal Workflow</b>	<ol style="list-style-type: none"><li>1. Admin will login to his account.</li><li>2. System will display the homepage.</li><li>3. From navigation drawer, Admin will select the option <b>Manage All Products</b>.</li><li>4. System will display all the products.</li><li>5. Admin tap on <b>update</b> button.</li><li>6. System will divert the Admin towards the page where Admin can update particular product.</li><li>7. Admin tap on <b>delete</b> button.</li><li>8. System will delete the particular product.</li></ol>
<b>Alternative Workflow</b>	

# Chapter 4

## System Design

## Chapter 4: System Design

All the system design is depending on these UML diagrams which are as follows:

### 4.1. Architecture Diagram

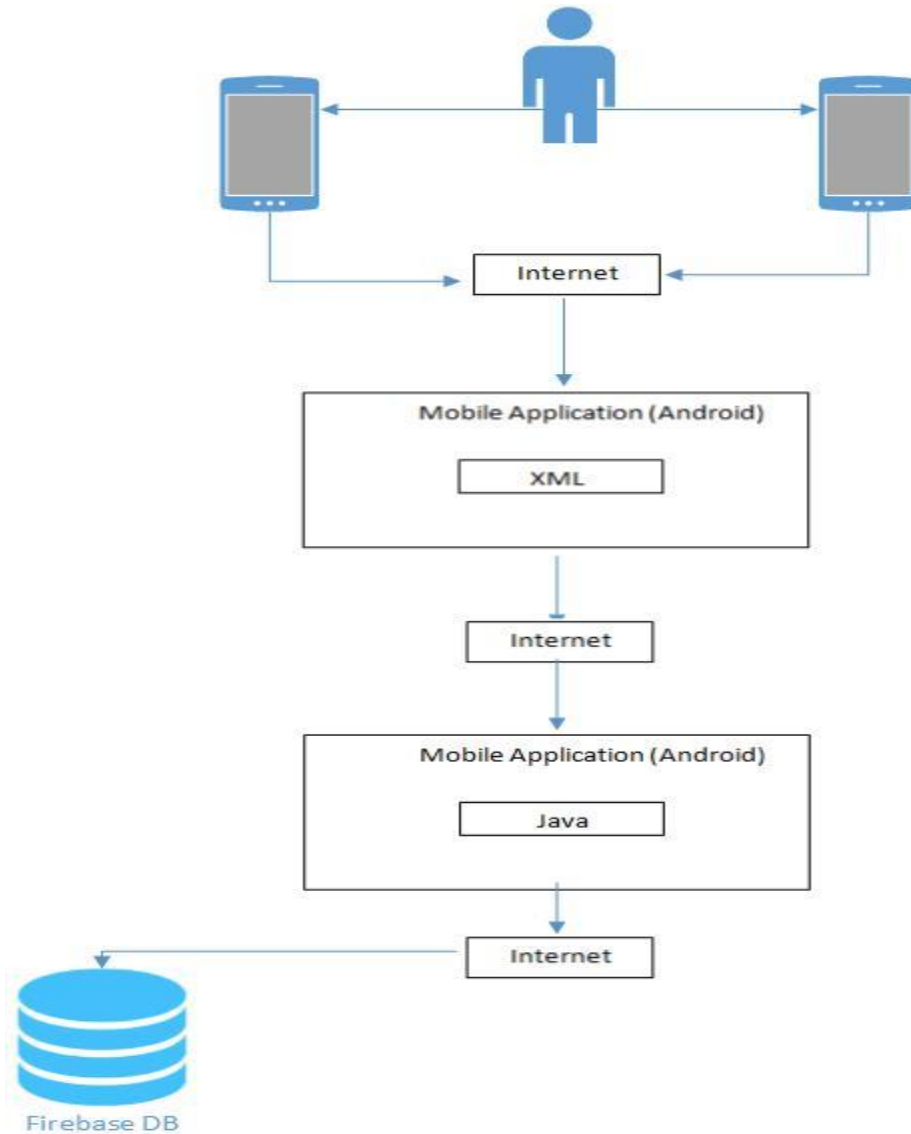


fig 4. 1 Architecture design

## 4.2. Domain Model

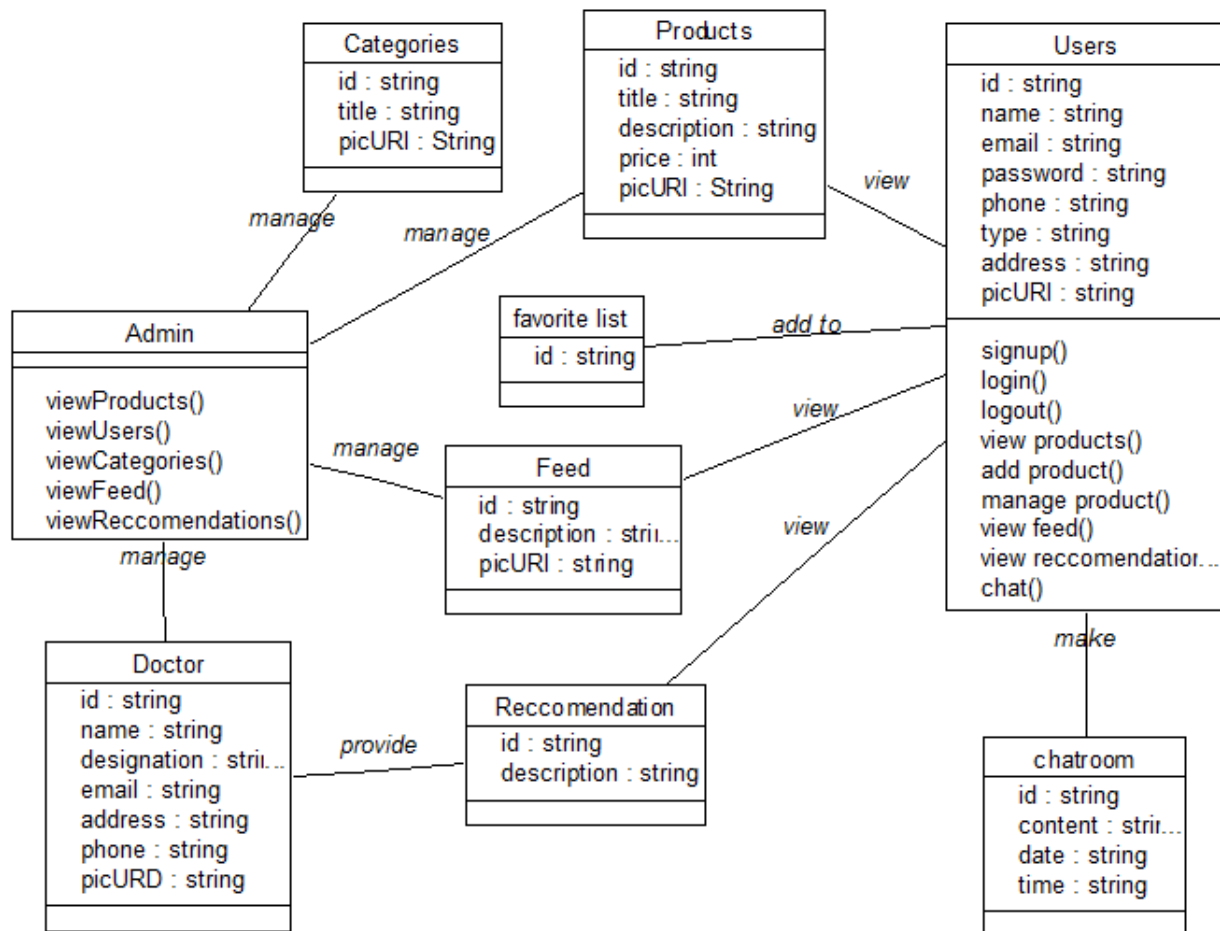


fig 4. 2 Domain Model

### 4.3. Entity Relationship Diagram with data dictionary

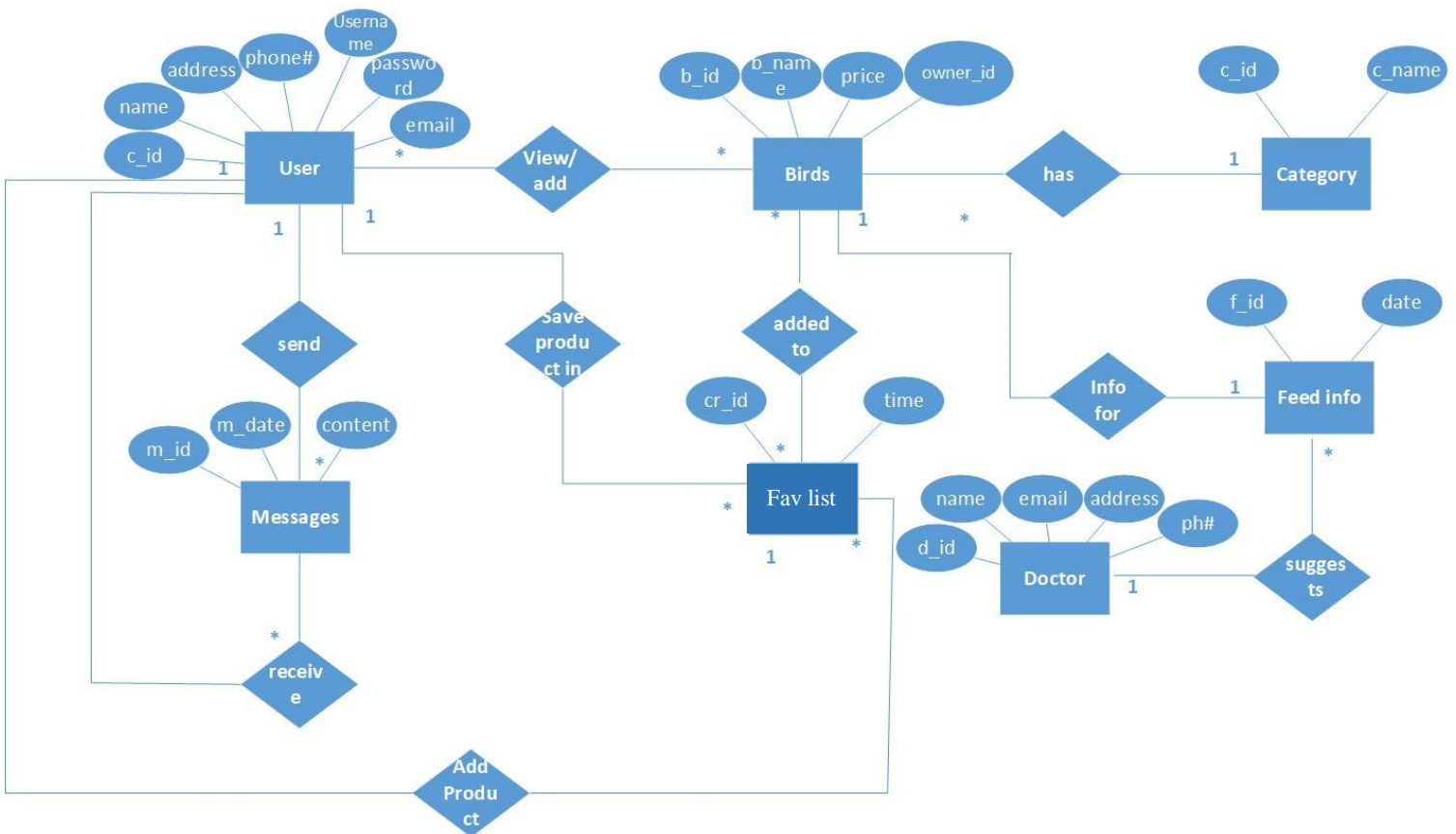


fig 4. 3 ERD Diagram

Entity Name	User	
Entity Description	This table will store the data of all registered users	
Column Name	Column Description	Data type
User id	This column will store the ID of user and it will be auto increment	String
User type	It will store the type of user either ordinary user or admin user	String
name	It will store the name of user	String
username	It will store the username of the user by which he will login	String
password	It will store the password of user's account	String
email	It will store the email address of the user	String

User phone	It will store the phone number of the user	String
User phone	It will store the phone number of the user	String
User address	It will store the address of the user in which area he is living	String

<b>Entity Name</b>	Product	
<b>Entity Description</b>	This table will store the data of products which users will add	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Product id	It will store the ID of the product and it will be auto increment	String
Product title	It will store the name of the product	String
Price	It will store the price of the product	String

<b>Entity Name</b>	Category	
<b>Entity Description</b>	This table will store the data of all categories which admin will add	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Category id	It will store the ID of the category and it will be auto increment	String
Category title	It will store the name of the category	String

<b>Entity Name</b>	Favorite	
<b>Entity Description</b>	This table will store the data of all products marked as favorite by the users	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Favorite id	It will store the ID of the favorite and it will be auto increment	String
Product id	It will store the ID of the product which is being added to the favorite	String
User id	It will store the ID of the user who is going to add product in favorite	String

<b>Entity Name</b>	<b>Doctor</b>	
<b>Entity Description</b>	This table will store the data of all doctors added by the admin	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Doctor id	It will store the ID of the Doctor and it will be auto increment	String
Doctor name	It will store the name of the doctor	String
Doctor's Specification	It will store the specification of the doctor	String
Doctor email	It will store the email address of the doctor	String
address	It will store the address of the doctor where he will be available	String
Phone number	It will store the phone number of the user	String

<b>Entity Name</b>	<b>Feed</b>	
<b>Entity Description</b>	This table will store the data of bird's feed added by admin	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Feed id	It will store the ID of the feed and it will be auto increment	String
Feed titles	It will store the name of the feed	String

<b>Entity Name</b>	<b>Chat</b>	
<b>Entity Description</b>	This table will store the chat between seller and buyer	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Message id	It will store the ID of the message and it will be auto increment	String
Content	It will store the message	String
Date	It will store the date on which message will be sent	Date
Time	It will store the time on which the message will be sent	Date

<b>Entity Name</b>	<b>Recommendations</b>	
<b>Entity Description</b>	This table will store the recommendations given by doctors for the diseases of birds and added by the admin	
<b>Column Name</b>	<b>Column Description</b>	<b>Data type</b>
Recommendation ID	It will store the ID of the recommendation and it will be auto increment	String
Doctor ID	It will store the ID of the doctor who will give the specific recommendation	String
Product ID	It will store the ID of the product for which the recommendation will be given	String
Description	It will store the recommendation given by the doctor	String

### 4.4. Class Diagram

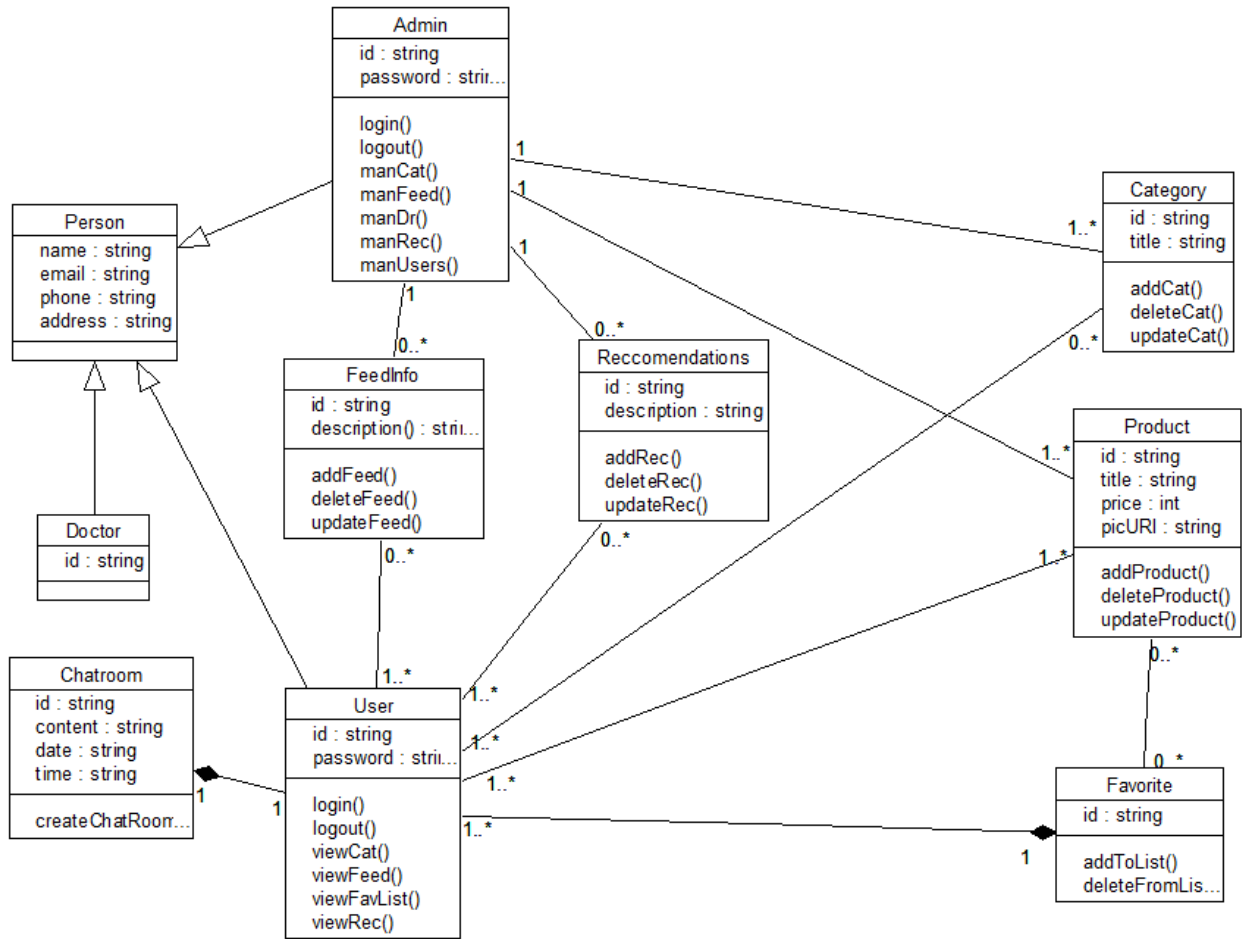


fig 4. 4 Class Diagram

## 4.5. Sequence / Collaboration Diagram

Login:

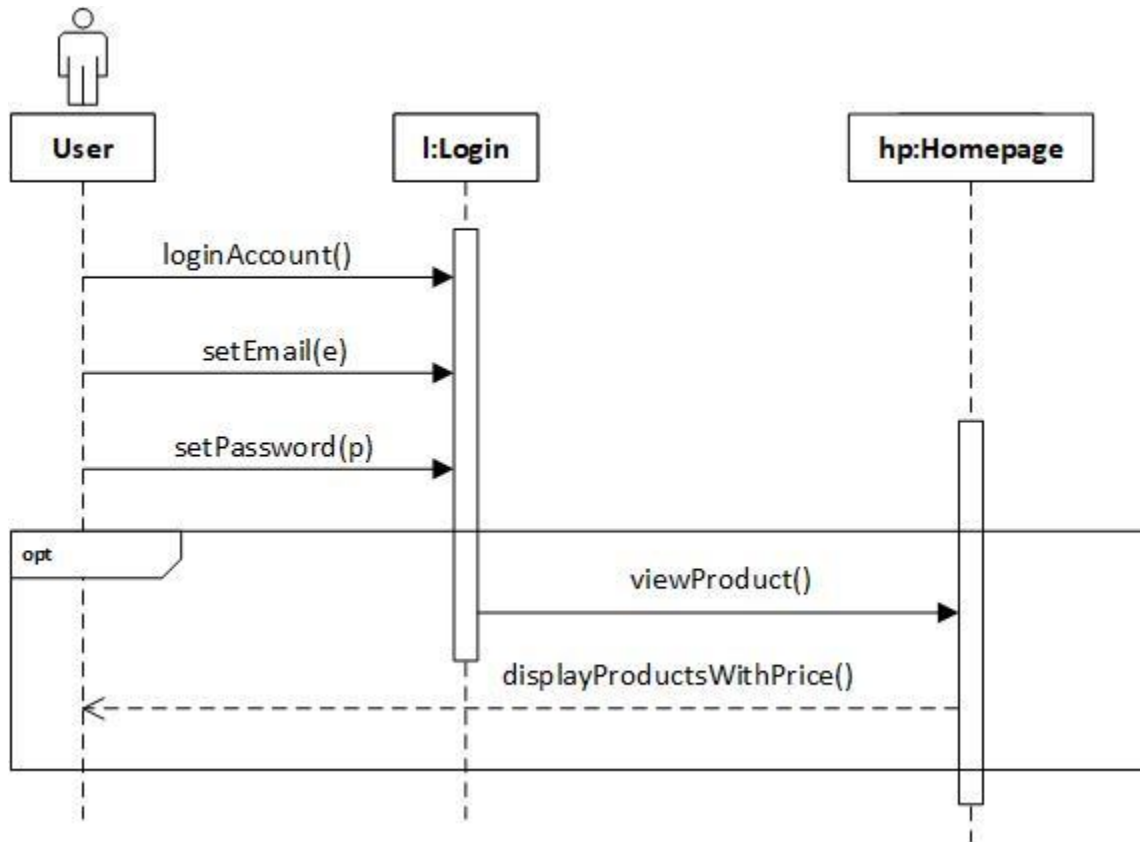
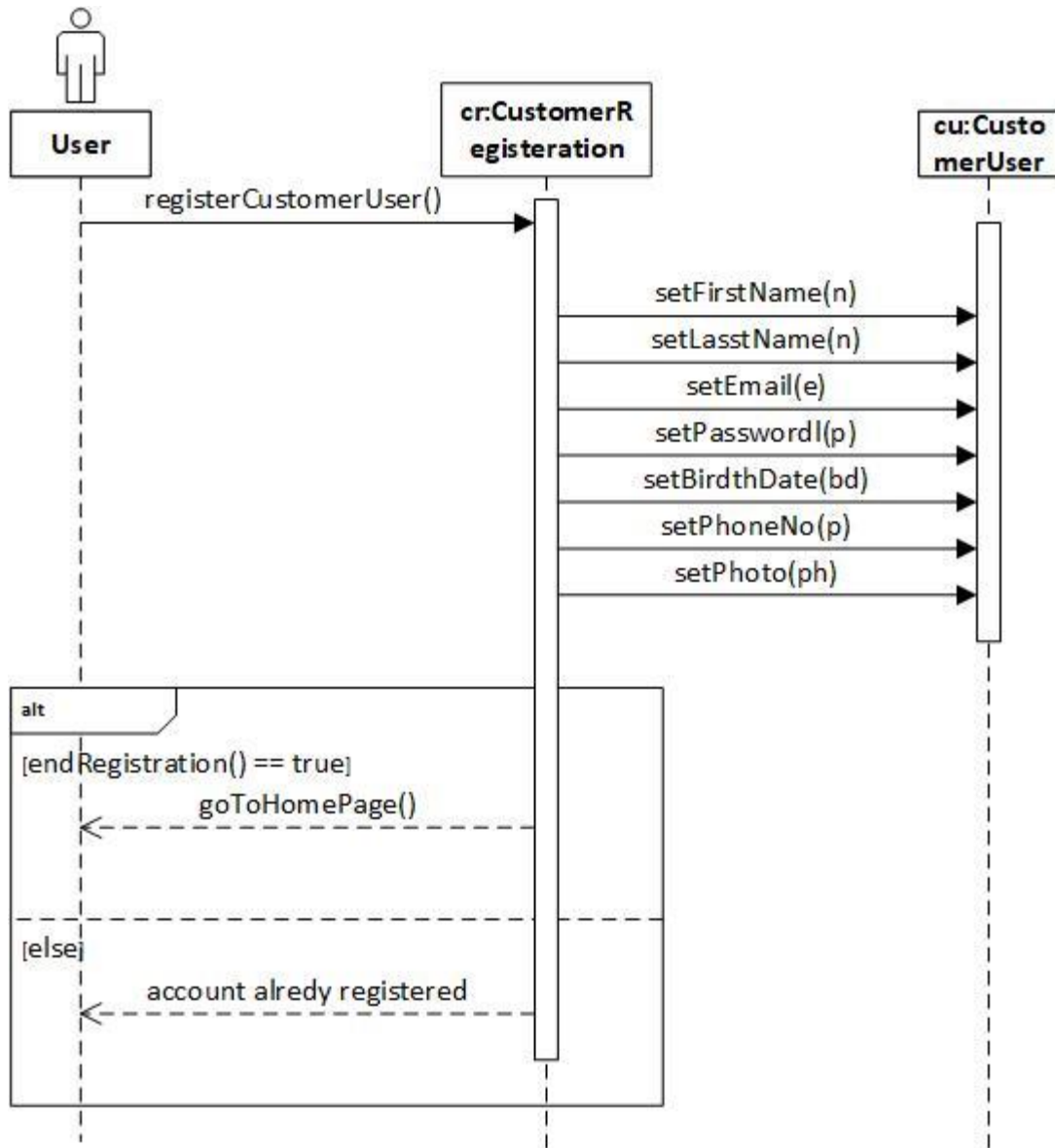
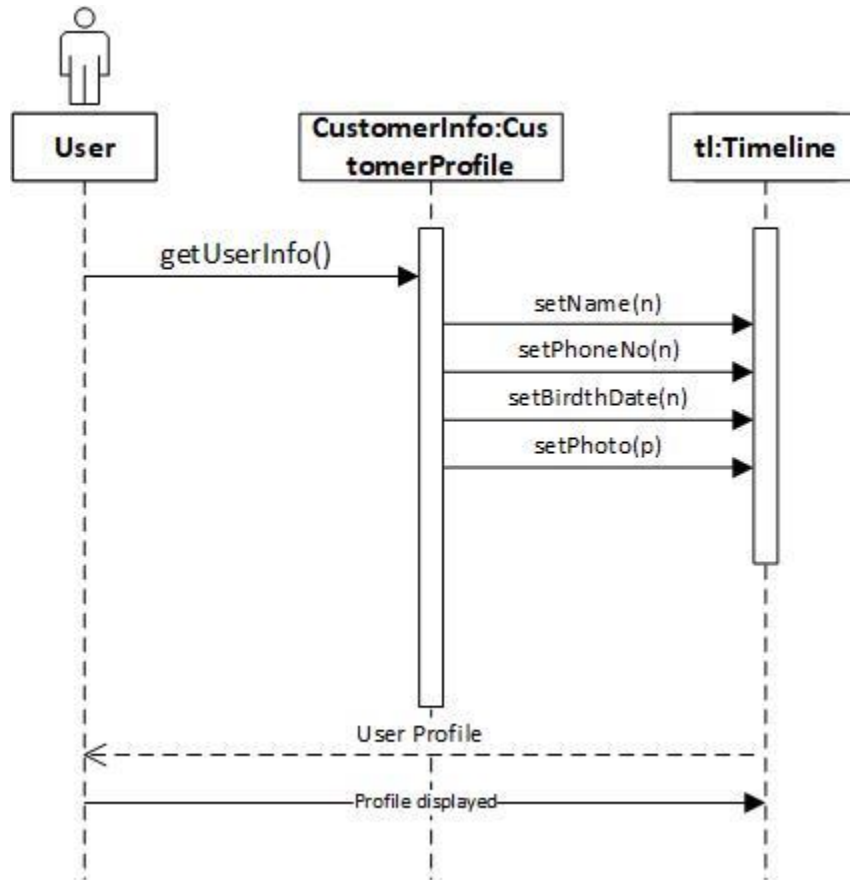
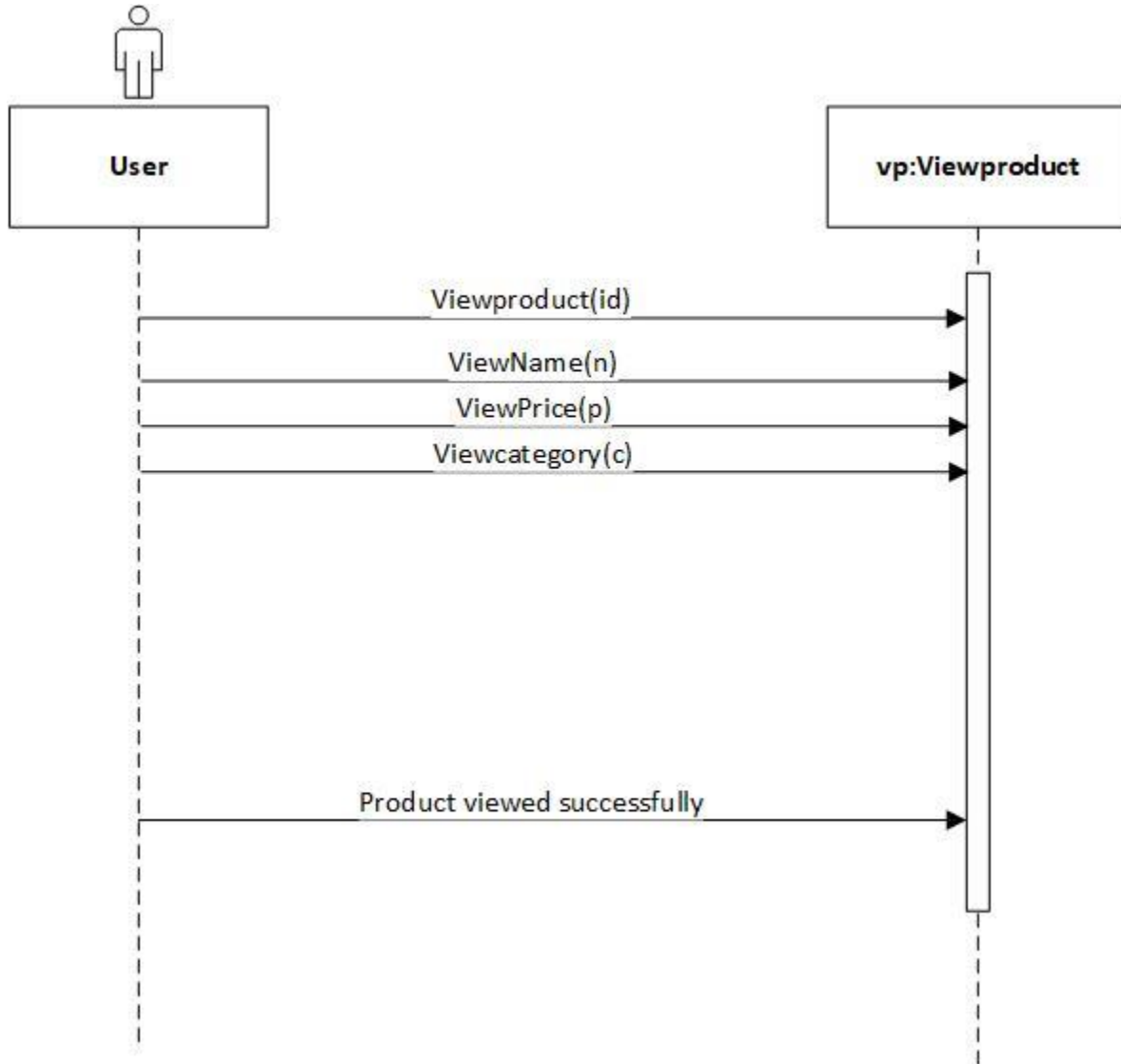


fig 4. 5 Sequence Diagram

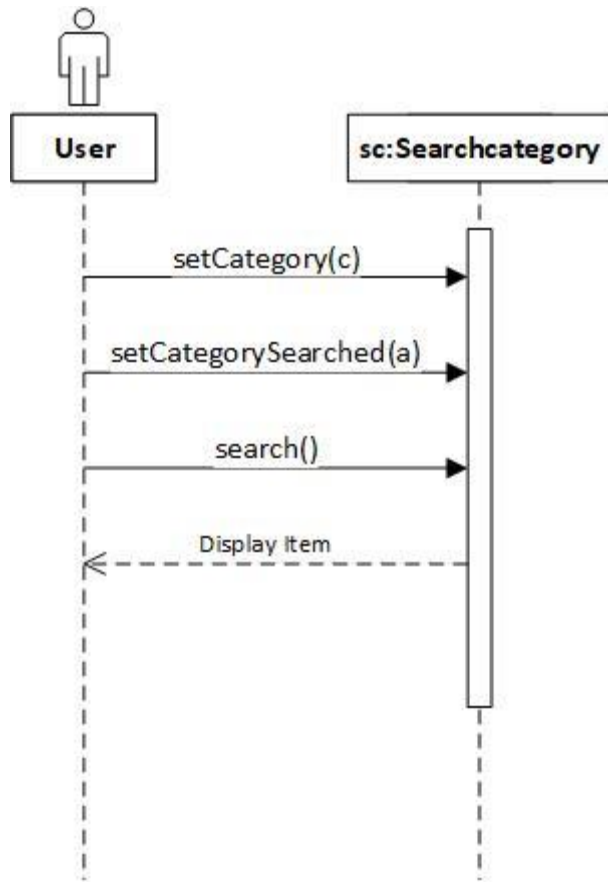
**Customer User Registration:**

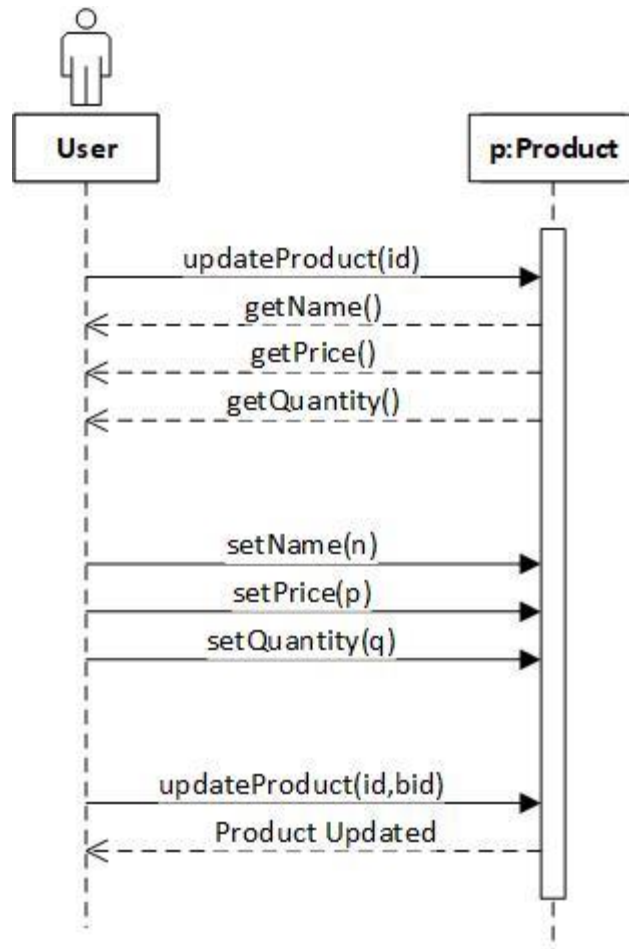
**View Profile:**

**View Product:**



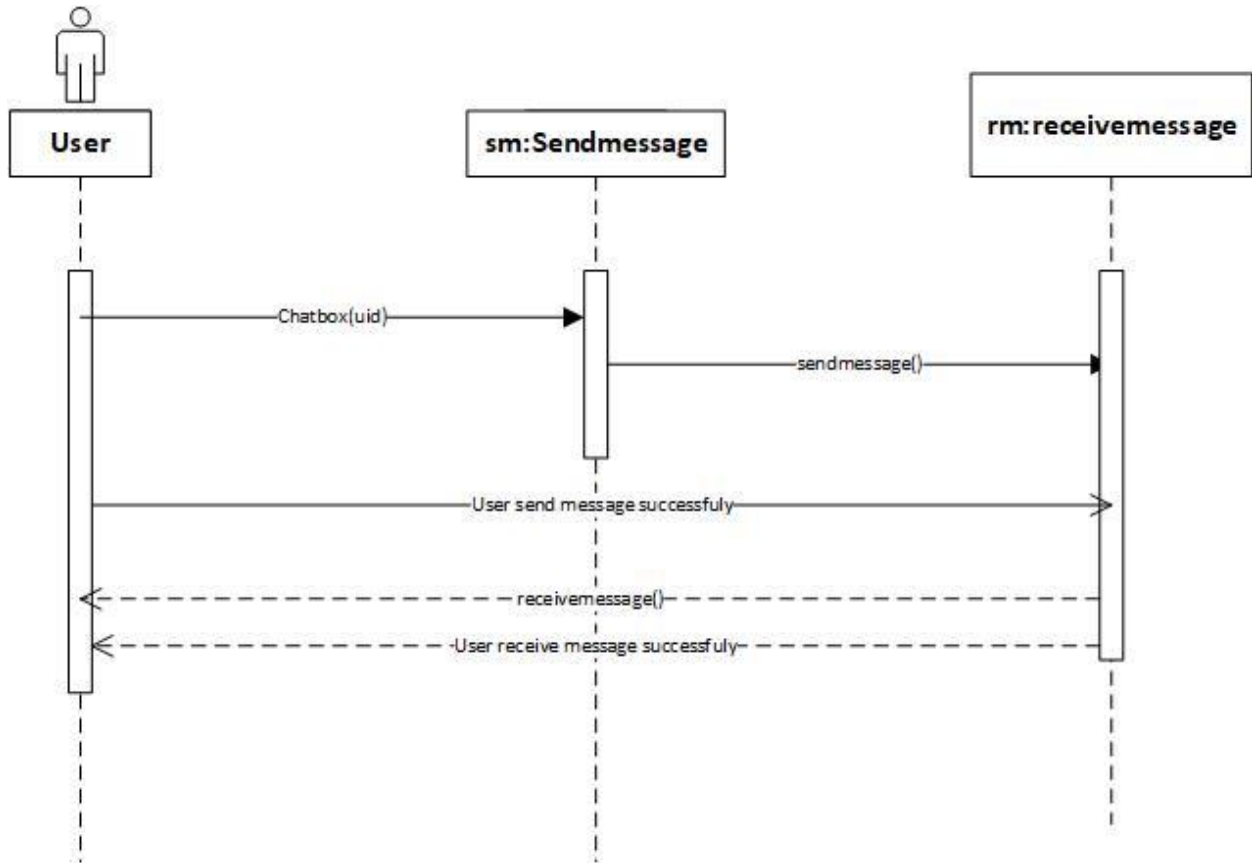
**Search Item:**



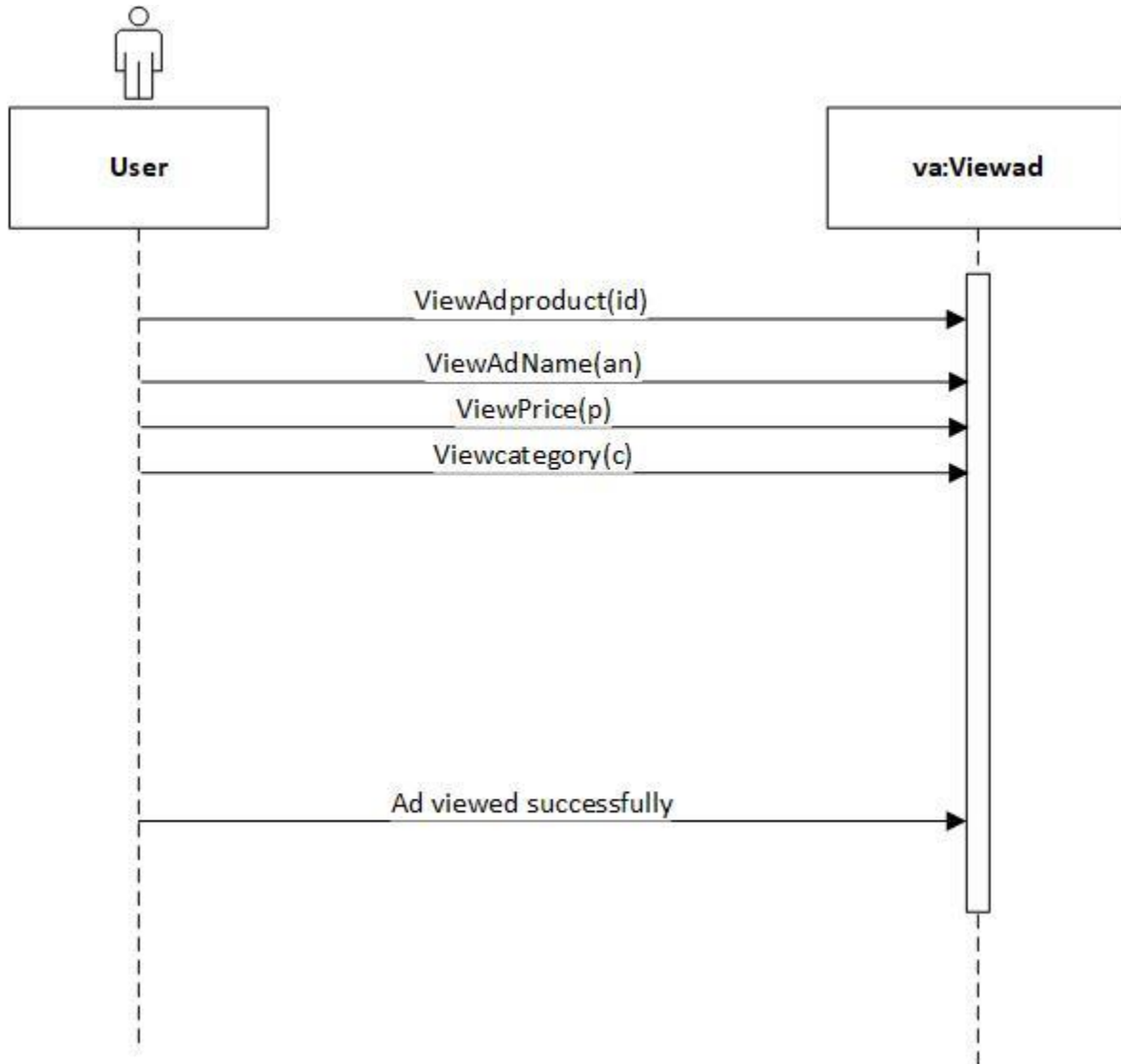
**Update Product:**

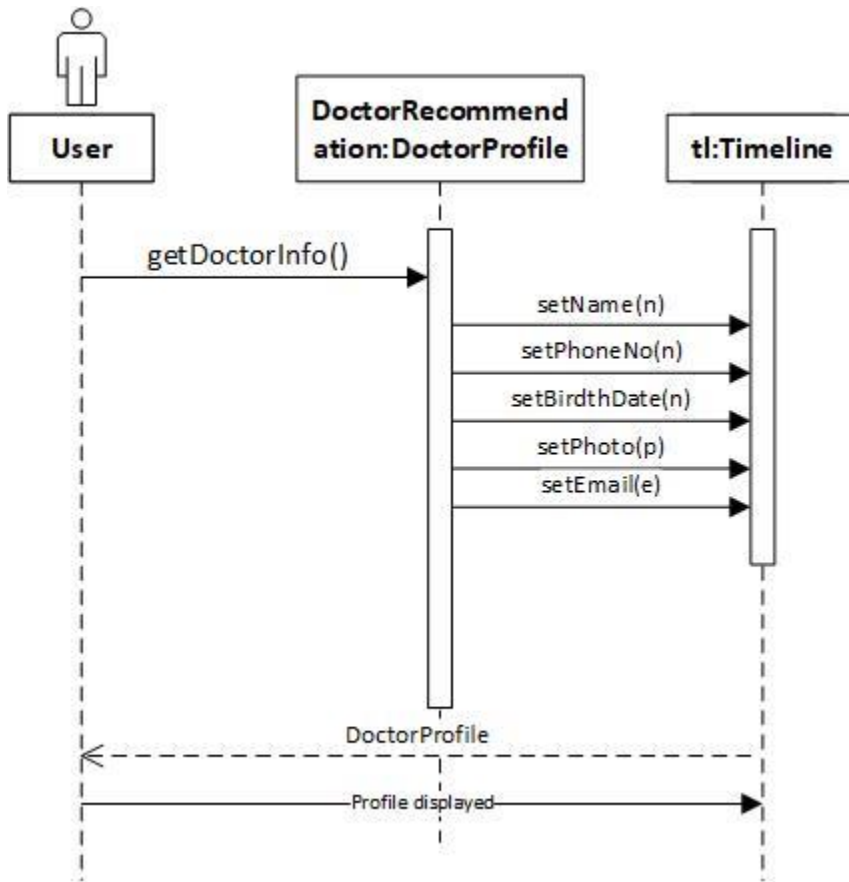
**Del Product:**

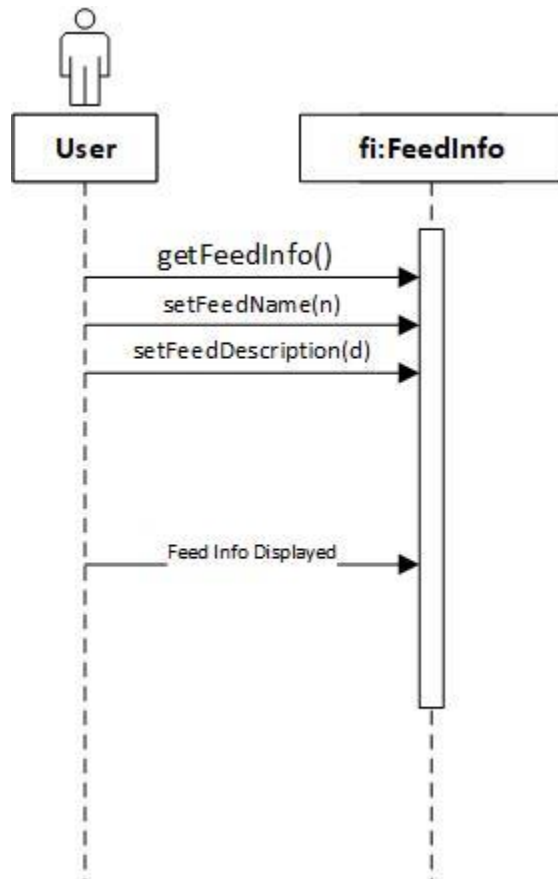


**Chat:**

Ad:



**Doctor Recommendation:**

**Feed:**

## 4.6. Operation contracts

### 1. Operational Contract: Sign Up

**Operation:** Create\_acc (name, username, Password, email, phone, location)

**Cross Reference:** Use case: Signup

**Pre – Conditions:**

- User will fill sign up form requirements and click on sign up button.

**Post – Conditions:**

- A Create Account instance a1 was created (instance creation)
- After successfully account creation, the instance a1 was destroyed (instance destruction).

### 2. Operational Contract: Login

**Operation:** Login (username, Password)

**Cross Reference:** Use case: login

**Pre – Conditions:**

- Account must be created.

**Post – Conditions:**

- A Check Account instance cA1 was created (Instance Creation).
- cA1 was associated with user Account A1 (instance association).
- After successfully logged in, the instance cA1 and A1 were destroyed (instance Destruction).

### 3. Operational Contract: Add Birds Categories

**Operation:** Birds Categories (c\_id, c\_name)

**Cross Reference:** Use case: Birds Categories

**Pre – Conditions:**

- Admin must be login to add bird category.

**Post – Conditions:**

- Add Category ac instant was created (instance creation).
- After addition, the instance ac was destroyed (instance Destruction).

### 4. Operational Contract: Add Doctor Recommendation

**Operation:** Add doctor Recommendation (rec\_id, d\_id, b\_id, description)

**Cross Reference:** Use case: Doctor Recommendation

**Pre – Conditions:**

- Admin must be login and must click on add recommendations. And Admin must have recommendations approved from the doctor.

**Post – Conditions:**

- AAddRecomendationsar instance was created (instance creation).
- Ar instance was associated with b\_id and doc\_id instances (instance association).
- After addition, the instances ar, b\_id and doc\_id were destroyed (instance Destruction).

## 5. Operational Contract: add Feed Info

**Operation:** Add feed Info (fid, bid, description)

**Cross Reference:** Use case: Feed Info

**Pre – Conditions:**

- Admin must be login and must click on add feed info

**Post – Conditions:**

- AAddFeedInfoafi instance was created (instance creation).
- Afi was associated with Bird bid instance (instance association).
- After addition, the instance adi and bid was destroyed (instance Destruction).

## 6. Operational Contract: Chat Box

**Operation:** ChatBox (sender\_id, receiver\_id, content)

**Cross Reference:** Use case: Chat Box

**Pre – Conditions:**

- Users will click on other user profile chat box option for chatting.

**Post – Conditions:**

- A chatRoomcr instance was created (instance creation).
- Cr was associated with sender sid and receiver rid (instance association).
- After sending the message, cr instant was destroyed (instance destruction).

## 7. Operational Contract: Add Item

**Operation:**AddItem (item\_id, cat\_id, cus\_id, price, details)

**Cross Reference:**Use case: add item

**Pre – Conditions:**

- Customer/user must be login and must click on add item button..

**Post – Conditions:**

- AddItem instance was created (instance creation).
- Ait was associated with Category cat\_id instance and Customer cus\_id instance (instance association).
- After addition, the instance ait, cat\_id and cus\_id were destroyed (instance Destruction).

## 8. Operational Contract: Remove Item

**Operation:** Add feed Info (cus\_id, item\_id)

**Cross Reference:**Use case: remove item

**Pre – Conditions:**

- User must be login and must have already added an item.

**Post – Conditions:**

- A RemoveItem instance was created (instance creation).
- ri was associated with item item\_id instance (instance association).
- After remove, the instance ri and item\_id was destroyed (instance Destruction).

## 4.7. Activity Diagram

Full activity:

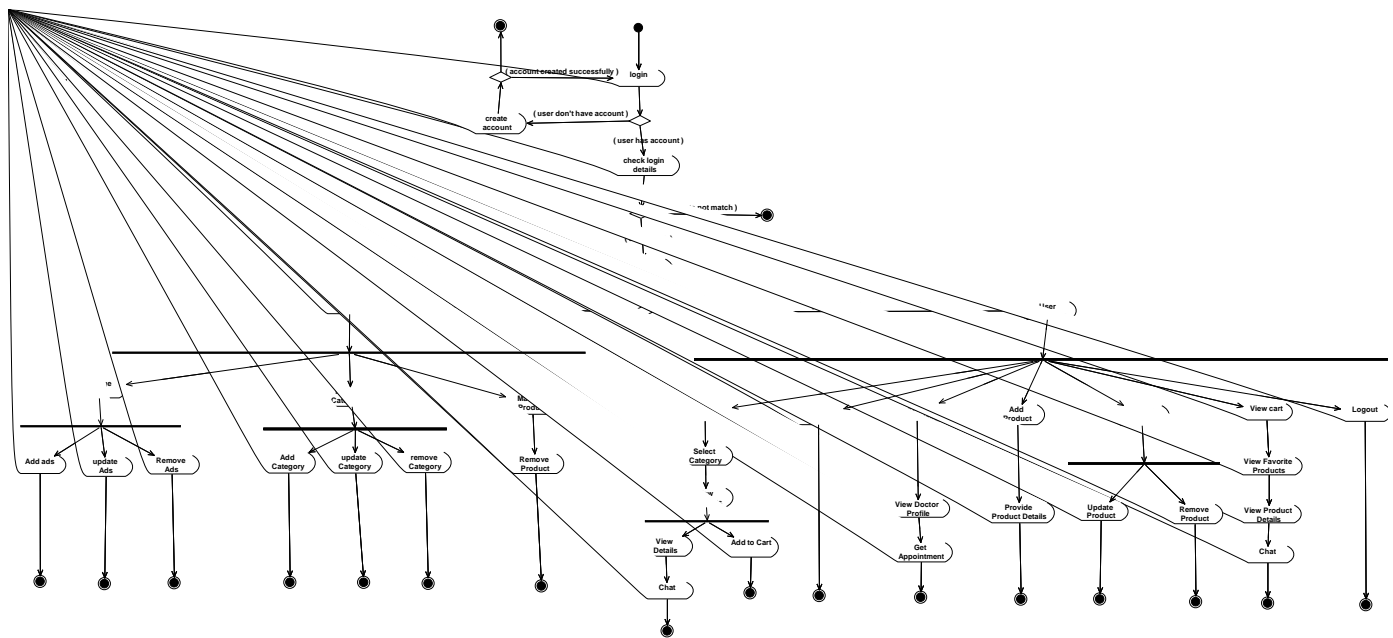
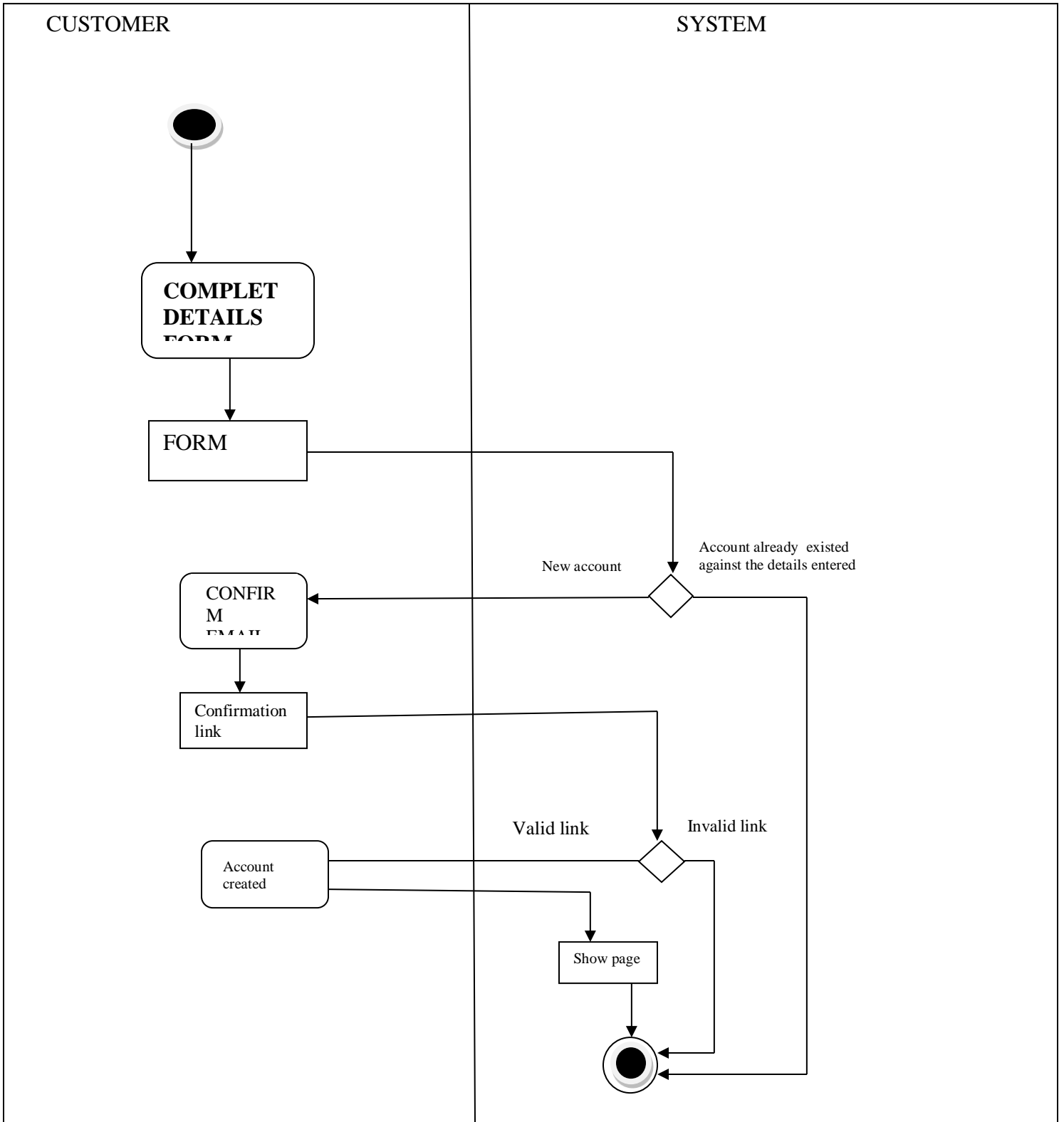
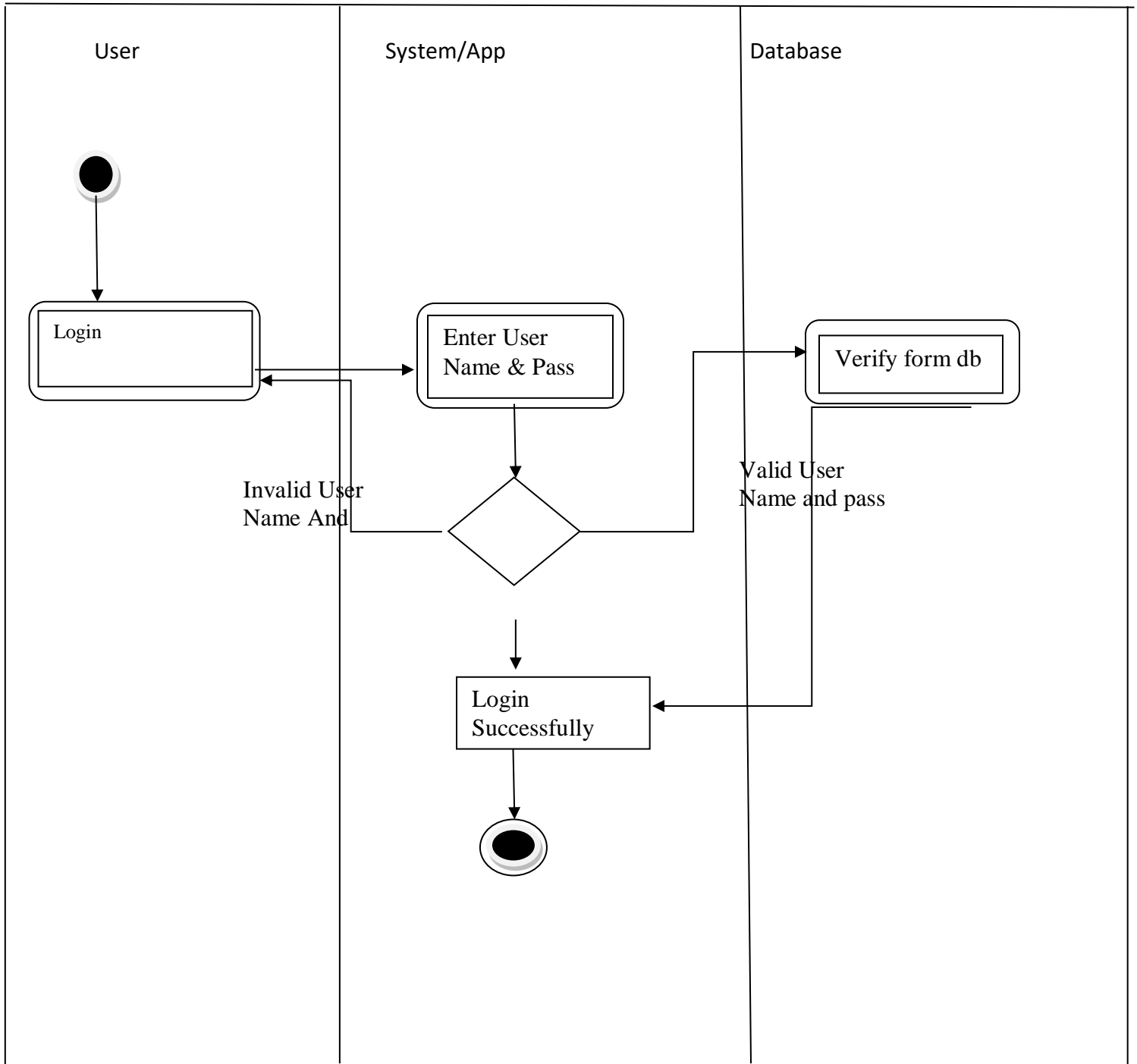


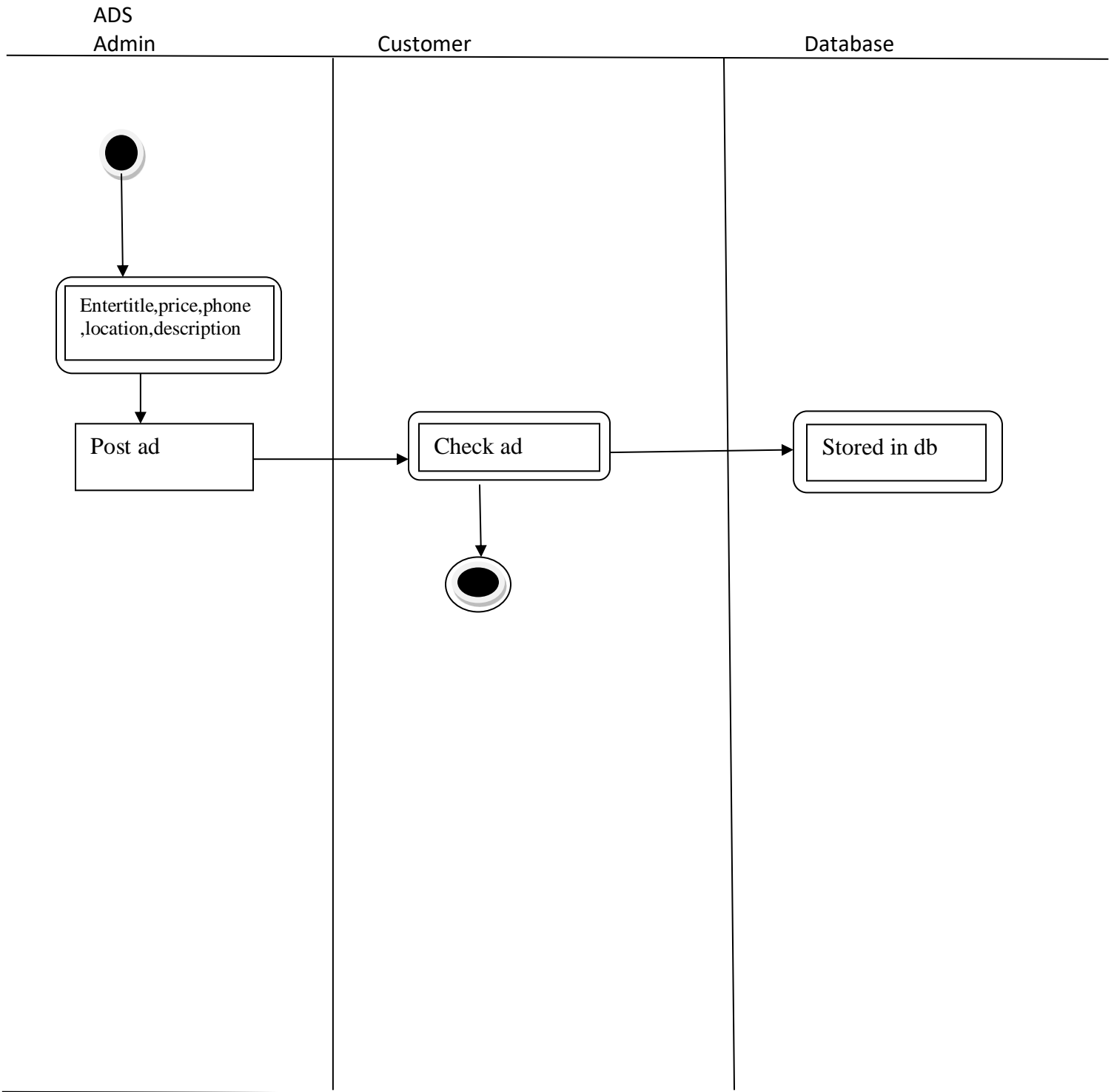
fig 4. 6 Activity Diagram

### Create account

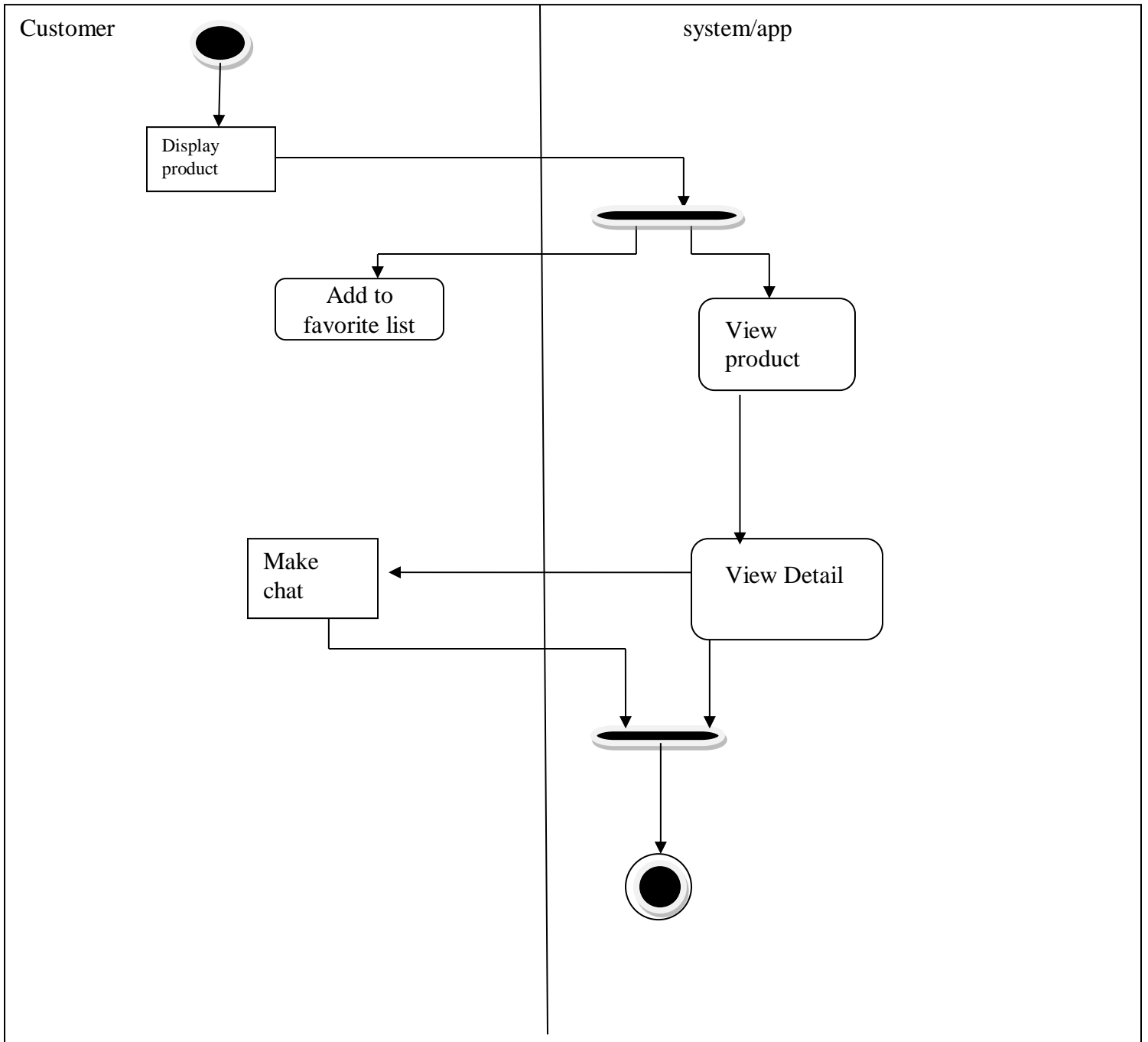


### Login

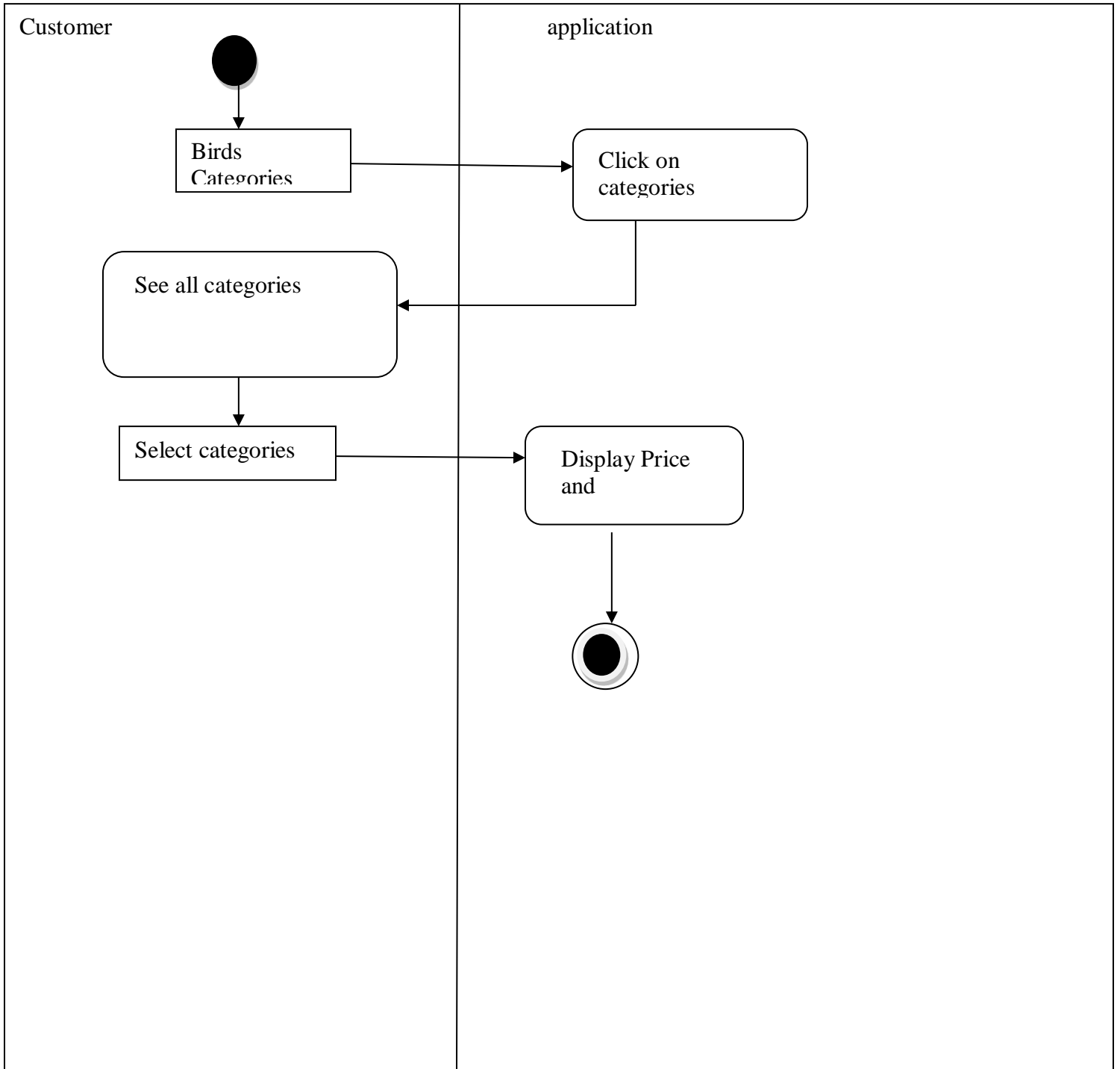




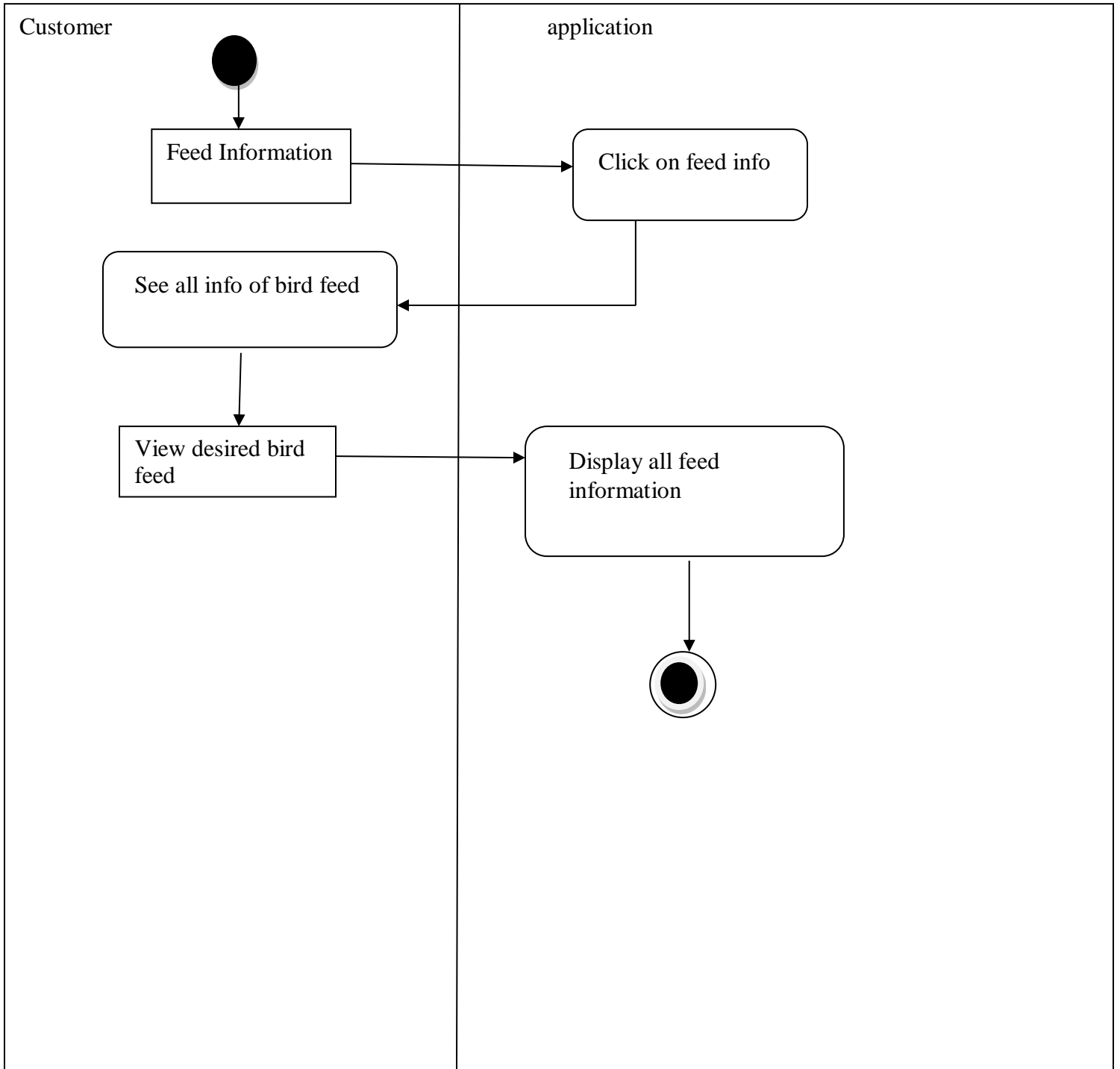
view product



### Categories



### Feed Information





### 4.7. State Transition Diagram

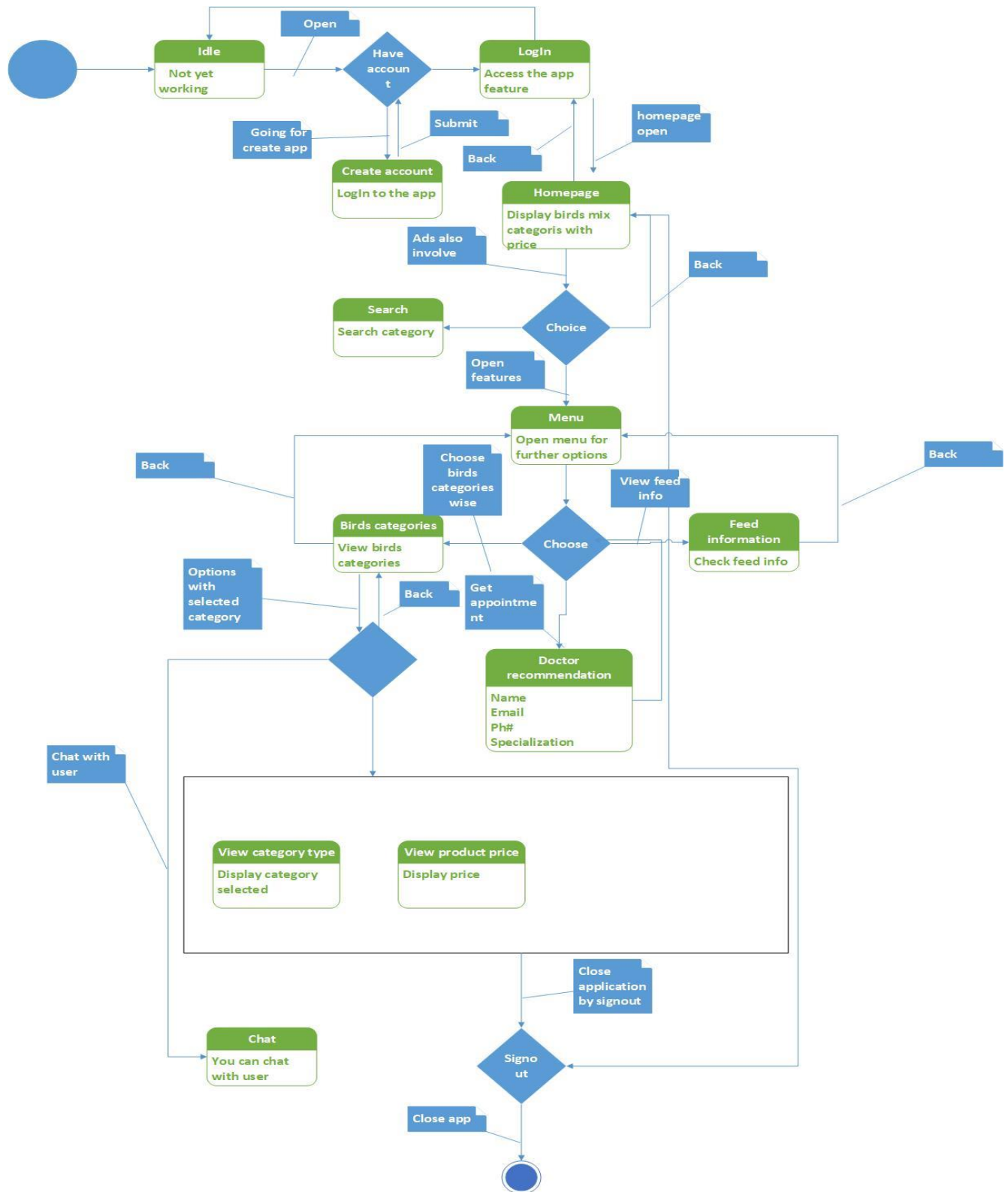


fig 4. 7 State Diagram

### 4.8. Component Diagram

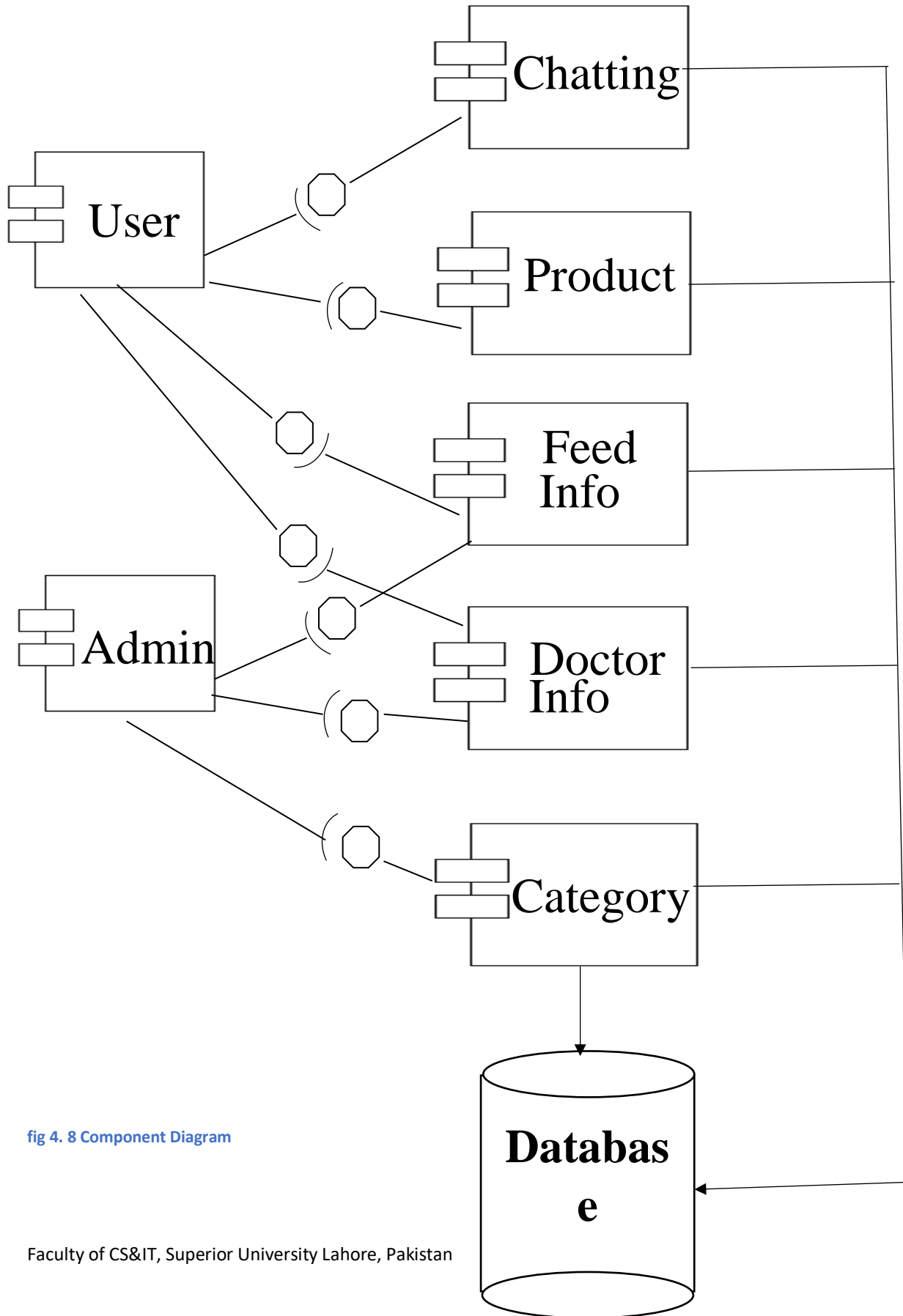


fig 4. 8 Component Diagram

## 4.9. Deployment Diagram

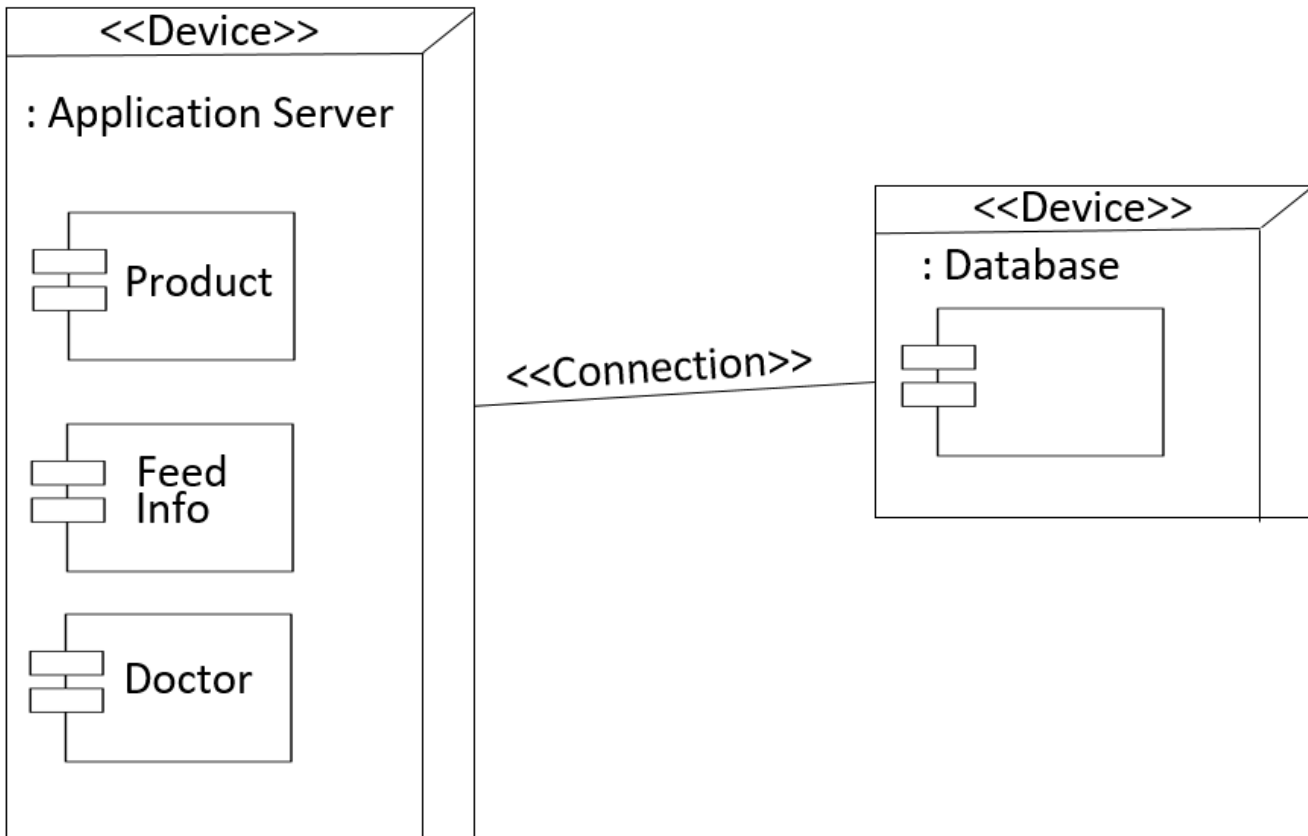


fig 4. 9 Deployment Diagram

# Chapter 5

## Implementation

## Chapter 5: Implementation

In this chapter we will cover the implementation of our project features and implementation flow of the mobile app. The libraries and components will also be discussed in this chapter which will be used to develop the mobile application. We will explain the deployment environment, tools & techniques and standards for the coding which will be used during the development of the app.

### 5.1. Important Flow Control/Pseudo codes

#### 1. Login Activity

- a) After app startup, **Login page** will open.
- b) If user already has account then he will fill the login fields and click on login, if data match with database, then user will be directed towards the **Home Page** activity.
- c) If user doesn't have account he will click on "**Don't have account**" button. And user will be directed towards the **Registration Page** activity.

#### 2. Registration activity

- a) User will fill the form and click on **Registration** button
- b) After registration, user will be directed towards the **Login Page** activity.

#### 3. Homepage activity

- a) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- b) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- c) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- d) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- e) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- f) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.
- g) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.
- h) If user clicks on **sign out** button then user will be directed towards the **Login Page**.
- i) If user clicks on **admin** button then user will be directed towards the **Admin Panel Page**.

#### 4. Categories activity

- a) On this activity, all available categories will appear.
- b) If user clicks on **specific category** button then user will be directed towards the **Category Products Page.**
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page.**
- d) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- e) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page.**
- f) If user clicks on **favorite** button then user will be directed towards the **Favorite Page.**
- g) If user clicks on **add product** button then user will be directed towards the **Add Product Page.**
- h) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page.**

## 5. Category Product activity

- a) All the available products of that particular category will appear on this activity
- b) If user clicks on **add to favorite** button then that particular product will be added to the favorite.
- c) If user clicks on **chat** button then a chat room will be created.
- d) From Menu if user clicks on **home** button then user will be directed towards the **Home page.**
- e) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- f) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- g) If user clicks on **doctor's recommendations** button then user will be directed towards the **8. Recommendations Page.**
- h) If user clicks on **favorite** button then user will be directed towards the **Favorite Page.**
- i) If user clicks on **add product** button then user will be directed towards the **Add Product Page.**
- j) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page.**

## 6. Feed Info activity

- a) On this activity, different feeds information will appear regarding all birds.
- b) From Menu if user clicks on **home** button then user will be directed towards the **Home page.**
- c) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.

- d) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- e) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- f) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.
- g) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

## 7. Recommendations activity

- a) On this activity, doctor's recommendations for the diseases of the birds will be appear.
- b) If user clicks on **View Doctor Profile** button then user will be directed towards the **Doctor Profile Page** activity
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- d) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- e) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- f) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- g) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.
- h) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

## 8. Doctors Profile activity

- a) On this activity, Profile of particular doctor will appear.
- b) If user clicks on **Get Appointment** button then a mail will be sent from user's email address to doctor's email address.
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- d) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- e) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- f) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- g) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- h) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.

- i) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

## 9. Favorite activity

- a) On this activity, all the products marked as favorite by the user will appear.
- b) If user clicks on **Remove from favorite** button then that particular product will be removed from the favorite.
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- d) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- e) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- f) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- g) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.
- h) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.
- i) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.
- j) If user clicks on **admin** button then user will be directed towards the **Admin Panel Page**.

## 10. Add Product Activity:

- a) On this activity, a form will appear, user will fill the form with the details of the product which he wants to sell.
- b) After filling the form, user will click on Add button, and then the product will be added to the database.
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- d) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- e) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- f) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- g) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- h) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.

- i) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

### 11. My Added Product Activity:

- a) On this activity, all the products added by this user will appear.
- b) If user clicks on **Update Product** button then user will be directed towards the **Update Product Page**
- c) If user clicks on **Remove Product** button then that particular product will be removed from the list.
- d) If user clicks on **Sold Product** button then that particular product will be black shaded.
- e) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- f) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- g) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- h) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- i) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- j) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.
- k) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

### 12. Update Product Activity:

- a) On this activity, user will update the details of the product.
- b) After changes, when user will click on the **Update** Button, the details of the product will be updated.
- c) From Menu if user clicks on **home** button then user will be directed towards the **Home page**.
- d) If user clicks on **categories** button from Menu then user will be directed towards the **Categories Page** activity.
- e) If user clicks on **Feed info** button then user will be directed towards the **Feed Info Page** activity.
- f) If user clicks on **doctor's recommendations** button then user will be directed towards the **Recommendations Page**.
- g) If user clicks on **favorite** button then user will be directed towards the **Favorite Page**.
- h) If user clicks on **add product** button then user will be directed towards the **Add Product Page**.

- i) If user clicks on **view added product** button then user will be directed towards the **My Added Products Page**.

### 13. Admin Panel Activity:

- a) On this activity, admin related menu will appear.
- b) From Menu if user clicks on **home** button then user will be directed towards the **Home page**
- c) If admin clicks on **add Category** button then admin will be directed towards the **Category Addition Page**
- d) If admin clicks on **view Categories** button then admin will be directed towards the **Added Categories Page**

### 14. Category Addition Activity:

- a) On this activity a form will appear, after filling the form, if admin click on **Add** then the category will be added to the database.
- b) From Menu if user clicks on **home** button then user will be directed towards the **Home page**
- c) If admin clicks on **view Categories** button then admin will be directed towards the **Added Categories Page**

## 5.2. Components, Libraries, Web Services and stubs

We have used basic components of Android Studio Like

- 1) Creating activities
- 2) Preview to view the demo
- 3) Gradle configuration to merge all the files
- 4) XML editor to develop the front end.
- 5) Java editor to maintain the backend.
- 6) Run command to generate apk file.

No separate libraries are used in this app, just pre-defined libraries which are already available in android studio are being used for the development.

Similarly no Web Services are used in the development.

## 5.3. Deployment Environment

**Operating System:**

Android

<b>Android Version:</b>	At least android kit Kat versions 4.4
<b>Internet:</b>	Mobile must have access to internet

## 5.4. Tools and Techniques

### Android Studio

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smart phones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touch screen devices, Google has further developed Android TV for televisions, Android Auto for cars and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. As of 2015, Android has the largest installed base of all operating systems.

### Google Firebase:

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. The code for firebase is in the public domain and is free for use for any purpose, commercial or private. Firebase is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

## 5.5. Best Practices / Coding Standards

### Naming Conventions

We are using naming conventions in android studio like this:

#### 1. Variables:

All the variables will be in lowercase and name will be according to the data, so easy to understand the purpose of the existing variable.

**2. Classes:**

The names of all the classes will be start with capital letter.

**3. Functions:**

Functions name will be start will lower letter and camel notation will be used to combine more than 1 alphabet for the name of the function.

**4. Parenthesis:**

There will be parenthesis sequence to make it easy to understand the starting and ending body of the loops and conditions.

**5. Comments:**

I will try to comments all the necessary parts of code, so that every reader can easily understand the code and the purpose of the code.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

Testing is the way a product, system or capability under the development is evaluated for correctness and proved to meet the stated requirement. Testing is done at each stage of development and has characteristic unique to the level of test being performed.

### 6.1. Use Case Testing

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Signup1 – Normal Workflow</b>	-
1)	Fill all the fields and submit data	System will display message of successful signup and divert the user towards the login page.
<b>Test Case:</b>	<b>Signup2 – All fields not filled</b>	-
1)	User skip some fields and submit data	System will display message to fill all the fields.
<b>Test Case:</b>	<b>Signup3 – Username not available</b>	-
1)	User choose the username which already has been taken and not available for the current user.	System will display the message to change the username.
<b>Test Case:</b>	<b>Signup4 – Short length username or password</b>	-
1)	User put the username or password containing characters less than 3.	System will display message to increase the length of username or password respectively.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Login1 – Normal Workflow</b>	-
1)	User fill both the fields	System will divert the user towards the home page.
<b>Test Case:</b>	<b>Login2 – Both fields are not filled</b>	-
1)	User doesn't fill both fields and try to login	System will display message to fill the both fields.

<b>Test Case:</b>	<b>Login3 – Username or password not matching</b>	-
1)	User enter the incorrect username or password to login	System will display the message to notify the user about the wrong username or password

TEST CASE(S)	Steps	Expected Results
Test Case:	Update My Account 1– Normal Workflow	-

TEST CASE(S)	Steps	Expected Results
Test Case:	ViewProdcts1 – Normal Workflow	-
-	-	-
Test Case:	ViewProdcts2 – Try to add user’s own product to his favorite list	-
1)	User tap on add to favorite button for his own product.	System will display message that user can’t add his own product to his favorite list.
Test Case:	ViewProdcts3 – Try to create chatroom for user’s own product	-
1)	User tap on chat button for his own product.	System will display message that user can’t create chatroom for his own product.

TEST CASE(S)	Steps	Expected Results
Test Case:	View Categories – Normal Workflow	-
1)	Tap on Specific Category	System will display all the products falling under that particular category

TEST CASE(S)	Steps	Expected Results
Test Case:	Feed Info – Normal Workflow	-
-	-	-

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>My Favorite1– Normal Workflow</b>	-
1)	User Tap on <b>Remove from favorite</b> button	System will remove the product from user's favorite list.
2)	User Tap on <b>Detail</b> button	System will divert the user towards the page containing complete detail regarding that particular product

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Recommendations – Normal Workflow</b>	-

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Manage My Products – Normal Workflow</b>	-
1)	User tap on <b>update</b> button.	System will divert the user towards the page where user can update his product
2)	User tap on <b>delete</b> button	System will delete the user's product

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Add Product1– Normal Workflow</b>	-
1)	User will fill all the fields and submit the data.	System will display the message about successful addition of the product
<b>Test Case:</b>	<b>Add Product2– Not filled all fields</b>	-
1)	User skip some fields and submit the data.	The system will display the message to fill all the fields

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>My Inbox1– Normal Workflow</b>	-
1)	User taps on <b>Specific</b> contact.	System will open a chatroom where the user can chat with the contact person.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Update My Account 1– Normal Workflow</b>	-
1)	System will retrieve the user's account data from database.	System will display that data to the user.
2)	User will make changes to his data, and submit the changed data.	System will display the message about successfully updated data.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Add Category1– Normal Workflow</b>	-
1)	Admin will fill all the fields and submit the data.	System will display the message about successful addition of the Category
<b>Test Case:</b>	<b>Add Category2– Not filled all fields</b>	-
1)	Admin skip some fields and submit the data.	The system will display the message to fill all the fields

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Manage Categories – Normal Workflow</b>	-
1)	Admin tap on <b>update</b> button.	System will divert the admin towards the page where admin can update his product
2)	Admin tap on <b>delete</b> button	System will delete the particular category.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Add Feed1– Normal Workflow</b>	-
1)	Admin will fill all the fields and submit the data.	System will display the message about successful addition of the Feed
<b>Test Case:</b>	<b>Add Feed2– Not filled all fields</b>	-
1)	Admin skip some fields and submit the data.	The system will display the message to fill all the fields

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Manage Feed – Normal Workflow</b>	-
1)	Admin tap on <b>update</b> button.	System will divert the admin towards the page where admin can update feed information.
2)	Admin tap on <b>delete</b> button	System will delete the particular feed.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Add Doctor1– Normal Workflow</b>	-
1)	Admin will fill all the fields and submit the data.	System will display the message about successful addition of the Doctor
<b>Test Case:</b>	<b>Add Doctor2– Not filled all fields</b>	-
1)	Admin skip some fields and submit the data.	The system will display the message to fill all the fields

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Manage Doctor – Normal Workflow</b>	-
1)	Admin tap on <b>update</b> button.	System will divert the admin towards the page where admin can update doctor information.
2)	Admin tap on <b>delete</b> button	System will delete the particular doctor.

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Add Recommendation1– Normal Workflow</b>	-
1)	Admin will fill all the fields and submit the data.	System will display the message about successful addition of the Recommendation
<b>Test Case:</b>	<b>Add Recommendation2– Not filled all fields</b>	-
1)	Admin skip some fields and submit the data.	The system will display the message to fill all the fields

TEST CASE(S)	Steps	Expected Results
<b>Test Case:</b>	<b>Manage Recommendation – Normal Workflow</b>	-
1)	Admin tap on <b>update</b> button.	System will divert the admin towards the page where admin can update Recommendation information.
2)	Admin tap on <b>delete</b> button	System will delete the particular Recommendation.

TEST CASE(S)	Steps	Expected Results
Test Case:	<b>Manage All Users – Normal Workflow</b>	-
1)	Admin tap on <b>Make Admin/Make Simple User</b> button.	System will divert the Admin towards the where Admin can update user's user type.
2)	Admin tap on <b>Remove</b> button	System will remove that particular user.

TEST CASE(S)	Steps	Expected Results
Test Case:	<b>Manage All Products – Normal Workflow</b>	-
1)	Admin tap on <b>update</b> button.	System will divert the Admin towards the page where Admin can update his product
2)	Admin tap on <b>delete</b> button	System will delete the user's product

## 6.2. Equivalence partitioning

### 1) Sign up page:

#### a) Username input:

Here is the test condition for username input box:

**Valid Input:** If entered username is not existing already.

**Valid Input:** If entered username length is greater than 2.

**Invalid Input:** If entered username is existing already, and taken by another user.

**Invalid Input:** If entered username length is less than 3.

#### b) Password input:

Here is the test condition for Password input box:

**Valid Input:** If entered Password length is greater than 2.

**Invalid Input:** If entered Password length is less than 3.

#### c) Phone Number input:

Here is the test condition for Phone Number input box:

**Valid Input:** If entered Phone Number length is exactly equal to 11.

**Invalid Input:** If entered Phone Number length is greater than 11.

**Invalid Input:** If entered Phone Number length is less than 11.

### 2) Update Account page:

#### a) Password input:

Here is the test condition for Password input box:

**Valid Input:** If entered Password length is greater than 2.

**Invalid Input:** If entered Password length is less than 3.

#### b) Phone Number input:

Here is the test condition for Phone Number input box:

**Valid Input:** If entered Phone Number length is exactly equal to 11.

**Invalid Input:** If entered Phone Number length is greater than 11.

**Invalid Input:** If entered Phone Number length is less than 11.

### 6.3. Boundary value analysis

#### Sign up page (Username input field)

Test Scenario Description	Expected Outcome
Username length = 1	System should not accept
Username length = 2	System should not accept
Username length = 3	System should accept
Username length = 10	System should accept

#### Sign up page (Password input field)

Test Scenario Description	Expected Outcome
Password length = 1	System should not accept
Password length = 2	System should not accept
Password length = 3	System should accept
Password length = 10	System should accept

#### Sign up page (Phone Number input field)

Test Scenario Description	Expected Outcome
Phone Number length = 5	System should not accept
Phone Number length = 10	System should not accept
Phone Number length = 11	<b>System should accept</b>
Phone Number length = 12	System should not accept
Phone Number length = 20	System should not accept

## 6.4. Unit testing

This is a developer test that is successively more complex. It being by integrating component

Test type	Unit testing
Test case number	1
Test case name	Registration
Test case description	The user should be enter valid email ,name ,password, phone, address, security question and answer (to get password if forget).
Item to be tested	All the fields must be filled, otherwise system should not let the user get registered.
<b>Specification</b> <b>1. If user fill all fields:</b> System should display message to notify about successfully registration <b>2. If user not fill all fields</b> System will display message to fill all the fields.	

Test type	Unit testing
Test case number	2
Test case name	Login
Test case description	User will open the application. Login page will be visible to the user. User will enter his username and password. Then user will by diverted towards the homepage.
Item to be tested	All the fields must be filled and data should match with the database
<b>Specification</b> <b>1. User will enter his username and password.</b> On successful login, user will be diverted towards home page. <b>2. If user not fill both fields,</b> The system will display the message to fill the both fields. <b>3. If username or password doesn't match,</b> The system will display the message that username or password is not matching, and will ask to check again the username and password	

Test type	Unit testing
Test case number	3
Test case name	Logout
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>Sign Out</b> option from the navigation drawer. System will divert the user towards the login page.
Item to be tested	After login, user must click on logout button and after that user must be divert towards login page

Test type	Unit testing
Test case number	4
Test case name	View Products
Test case description	User will open the application. After Login to his account, at home page user can view some of the products. And user can also view all the products falling inside a particular category.
Item to be tested	All the products falling inside a particular category must display on screen
<p><b>Specification</b></p> <p><b>1. If user try to add his own product to his favorite list,</b> System will display message that user can't add his own product to his favorite list</p> <p><b>If user try to create chatroom for his own product,</b> The system will display the message that username or password is not matching, and will ask to check again the username and password</p>	

Test type	Unit testing
Test case number	5
Test case name	View Categories
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>categories</b> option from the navigation drawer. All the available categories will display to the user. Then user can

	select specific category to view products regarding that specific category.
Item to be tested	All the available categories must display to the user
<b>Specification</b> <b>1. If user select specific category</b> All the products falling under that category will be visible to the user	

<b>Test type</b>	<b>Unit testing</b>
Test case number	6
Test case name	View Feed Information
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>feed information</b> option from the navigation drawer. All the available feed information for the birds will display to the user.
Item to be tested	System must display all the available feed information for the birds to the user.

<b>Test type</b>	<b>Unit testing</b>
Test case number	7
Test case name	My Favorite
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>My Favorites</b> option from the navigation drawer. All the available categories will display to the user. Then user can select specific category to view products regarding that specific category.

Item to be tested	The user must be diverted towards the page containing complete detail regarding that particular product.
-------------------	--

Test type	Unit testing
Test case number	8
Test case name	Recommendations
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>Recommendations</b> option from the navigation drawer. All the available Recommendations will display to the user.
Item to be tested	System must display all the available Recommendations

Test type	Unit testing
Test case number	9
Test case name	Manage My Products
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>Manage My Products</b> option from the navigation drawer. All the user's product will display. Now User can update or delete any of his products.
Item to be tested	System must delete or update the user's product

Test type	Unit testing
Test case number	10
Test case name	Add Product

Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>Add Product</b> option from the navigation drawer. System will open a new page, where user can add new product to sell.
Item to be tested	System must display the message about successful addition of the product
<b>Specification</b> <b>1. If user fill all fields:</b> System must display the message about successful addition of the product <b>2. If user not fill all fields</b> The system must display the message to fill all the fields	

Test type	Unit testing
Test case number	11
Test case name	My Inbox
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>My Inbox</b> option from the navigation drawer. System will display the all contacts list to the user to whom the user made chat.
Item to be tested	A chat room must open where the user can chat with the contact person

Test type	Unit testing
Test case number	12
Test case name	Update My Account
Test case description	User will open the application. After Login to his account, at home page, user will tap on <b>Update My Account</b> option from the navigation drawer. System will divert the user towards the update page, where he will be able to update his account. Where user will change his data and then submit the updated data.

Item to be tested	System must display the message about the successfully updated data
-------------------	---

Test type	Unit testing
Test case number	13
Test case name	Add Category
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Category</b> option from the navigation drawer. System will open a new page, where admin can add new category.
Item to be tested	System must display the message about successful addition of the Category
<p><b>Specification</b></p> <p>1. <b>If admin fill all fields:</b> System will display the message about successful addition of the Category</p> <p>2. <b>If admin not fill all fields</b> The system will display the message to fill all the fields.</p>	

Test type	Unit testing
Test case number	14
Test case name	Manage Categories
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Categories</b> option from the navigation drawer. All the available categories will display. Now admin can update or delete any of available categories.
Item to be tested	System must delete or update the particular category

Test type	Unit testing
Test case number	15
Test case name	Add Feed

Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Feed</b> option from the navigation drawer. System will open a new page, where admin can add new Feed Information.
Item to be tested	System will display the message about successful addition of the Feed
<b>Specification</b> <b>1. If admin fill all fields:</b> System will display the message about successful addition of the Feed <b>2. If admin not fill all fields</b> The system will display the message to fill all the fields.	

Test type	Unit testing
Test case number	16
Test case name	Manage Feed
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Feed</b> option from the navigation drawer. All the available Feeds will display. Now admin can update or delete any of available feeds.
Item to be tested	System must delete or update the particular Feed

Test type	Unit testing
Test case number	17
Test case name	Add Doctor
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Doctor</b> option from the navigation drawer. System will open a new page, where admin can add new Feed Information.
Item to be tested	System will display the message about successful addition of the Doctor
<b>Specification</b> <b>1. If admin fill all fields:</b> System will display the message about successful addition of the Doctor <b>2. If admin not fill all fields</b> The system will display the message to fill all the fields.	

Test type	Unit testing
-----------	--------------

Test case number	18
Test case name	Manage Doctor
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Doctor</b> option from the navigation drawer. All the available Doctors will display. Now admin can update or delete any of available doctors.
Item to be tested	System must delete or update the particular Doctor

Test type	Unit testing
Test case number	19
Test case name	Add Recommendation
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Add Recommendation</b> option from the navigation drawer. System will open a new page, where admin can add new Recommendation Information.
Item to be tested	System will display the message about successful addition of the Recommendation.
<b>Specification</b> <b>1. If admin fill all fields:</b> System will display the message about successful addition of the Recommendation. <b>2. If admin not fill all fields</b> The system will display the message to fill all the fields..	

Test type	Unit testing
Test case number	20
Test case name	Manage Recommendation
Test case description	Admin will open the application. After Login to his account, at home page, admin will tap on <b>Manage Recommendation</b> option from the navigation drawer. All the available Recommendations will display. Now admin can update or delete any of available Recommendations.
Item to be tested	System will delete or update the particular Recommendation.

Test type	Unit testing
Test case number	21
Test case name	Manage All Users
Test case description	System will divert the Admin towards the page where Admin can update user's user type or System will remove that particular user according to given command.
Item to be tested	System must update user's user type or remove that particular user according to given command.

Test type	Unit testing
Test case number	22
Test case name	Manage All Products
Test case description	System will divert the Admin towards the page where Admin can update particular product or System will delete the particular product according to given command
Item to be tested	System must update or delete product.

## **6.5. Integration testing**

Integration testing is done to test the functionality and interfacing between the module. The system is built up for various module which work together to automate the activities. This module is work together in a same less way to achieve the desired result. Integration testing will test for this property of module. These modules display a cause and effect relationships, if data in one module changed then it effects the data to change in some other module also.

## **6.6. Performance testing**

Performance testing is performed to evaluate the performance of component of a particular system under a particular workload during this testing system component are monitored to verify both stability of system under the test. It strives to build performance standard into the implementation and design. These will include speed, stability, reliability and scalability.

## **6.7. Stress Testing**

We normally used stress testing to understand the system limit and at which point these are executed. Stress testing is especially helpful in determining whenever the system will perform in a satisfactory manner if the current load goes well above the expected manner.

# Chapter 7

## Summary, Conclusion and Future Enhancements

## Chapter 7: Summary, Conclusion & Future Enhancements

### 7.1. Project Summary

Now-a-days we can see that everyone is moving towards the IT system. IT is providing everything online to us so it is time saving. So, we find a problem in our society that people like to birds very much. So, they like to keep birds in their houses. There are many places where people are selling birds but the problem is that they have to visit many places to buy suitable birds. So in this way their time is wasted and their money as well. That's the problem which we found in our society. We want address this problem of the society. We want to develop a **mobile application**-based system which will provide a single platform to all bird lovers who want to sell or purchase birds. So if they will get all info regarding birds at a single platform, this will be very helpful for them. It will be time saving and money saving as well. There are already many companies who are providing online sell or buy items services but there is not any existing online system where people can purchase or sell small level birds. In this system, seller will be able to put ad of the bird which he wants to sell and then other users will be able to see the ads and then they can select a list of birds which they want to buy. Just like the Pet Shop, Parenda Bazaar or olx.com. Additional feature of this system will be a chat box, which will be available for the users to make bargaining (between seller and purchaser). This is briefly overview of our system.

### 7.2. Achievements and Improvements

Successfully, the system has been designed in response to the system analysis. All possible error in the program has been eliminated. Necessary validation techniques have been used and normal, abnormal and extremely data was used to test the system. However, doing this project has been a good boost to our confidence as the future IT member of our global village. Later on, some more valuable functionalities will be added to the system.

### 7.3. Critical Review

As the goal of the application is ease of use and to provide an interactive interface, extensive research has been done to gain an insight into the needs and behaviors of various users. The working of the application is made convenient and easy to use for the end user. In online mobile app based shopping systems there are three party who interact with the system, one of

them is customer who wants to buy product, second one who wants to sell product and third one is the system admin. A good mobile app based online shopping system should present the following factors to the customers for better usability.

#### **7.4. Lessons Learnt**

The objective of the Shopping systems is to increase the point of user choice, reduce time used in shopping (finding birds for sale or finding people who want to buy birds) and efficiency in buying birds.

1. To develop an easy way to use app-based interface where users can find products and can view the details of the product and can chat for bargaining.
2. User can add product to sell, delete or update the product.
3. User can view all products available to be sold.
4. User can view recommendations about birds' diseases and feed information about birds.
5. User can get appointment from doctor through email.
6. Admin can add/delete/update products, categories, recommendations, feed information and doctors' list.

#### **7.5. Future Enhancements/Recommendations**

In future we will try to develop our project so that we can add more valuable functionality to our system.

# Appendices

## Appendix A: User Manual

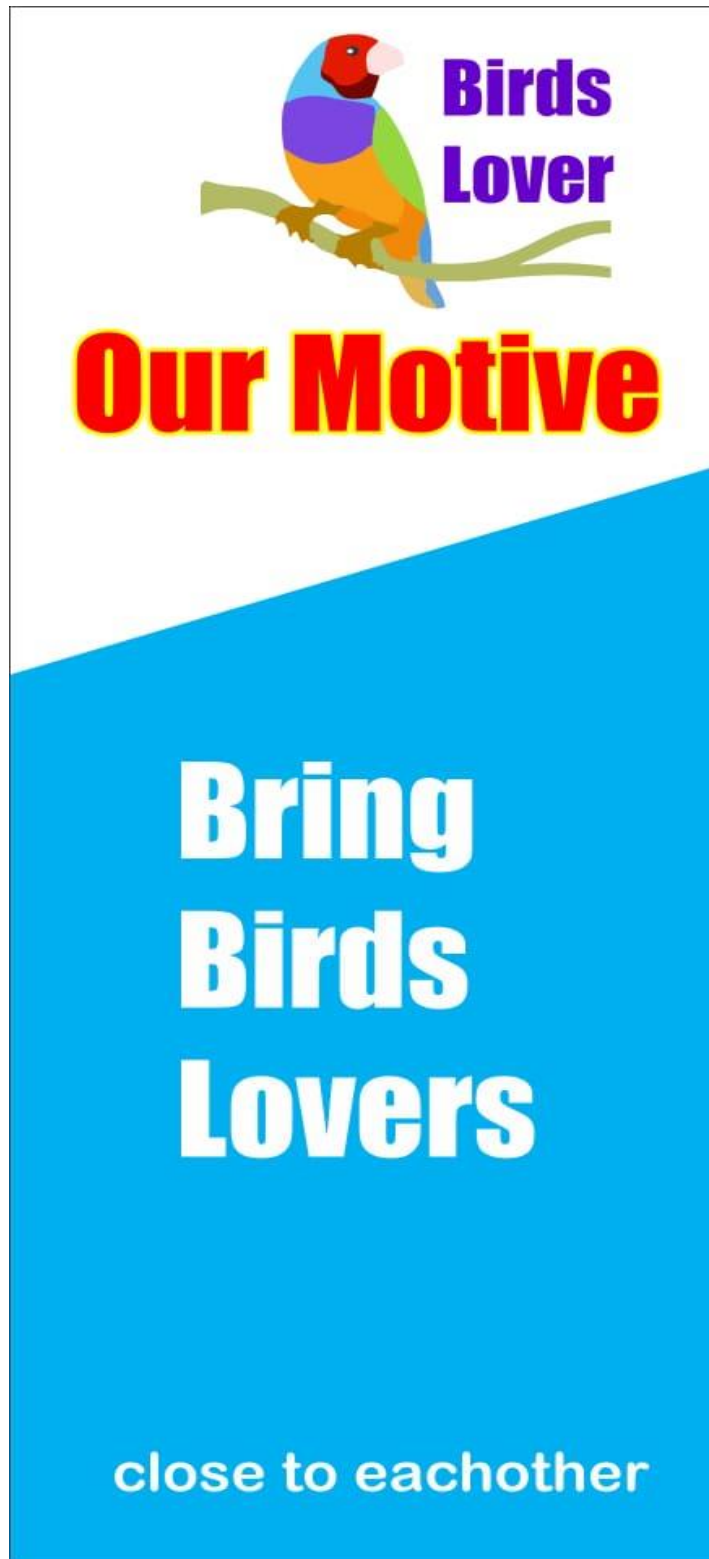
This project is an online application for bird's sale and purchase. There are two types of user we considered in this project. One is customer who will come, sale or purchase any bird and the other one is admin who control all the main activities of this application.

## Appendix C: Information / Promotional Material

### C.1. Broacher

 <p><b>Birds Lover</b></p> <p>You can post Ad to sell your bird</p> <p>You can get information about birds' feed</p> <p>You can get appointment from Doctors by emailing</p>	 <p><b>Birds Lover</b></p> <p><b>Our Uniqueness</b></p>	 <p><b>Birds Lover</b></p> <p><b>Our Motive</b></p>
<p><b>Our Services</b></p>	<p>We save your time</p> <p>We save your money</p> <p>We bring you people close</p> <p>We provide you a single platform but multiple services</p>	<p><b>Bring Birds Lovers</b></p> <p>close to eachother</p>

## C.2. Standee



### C.3. Banner



# Reference and Bibliography

## Reference and Bibliography

- [1] <http://www.parindabazar.com/>
- [2] <https://www.olx.com.pk/>
- [3] <http://www.google.com/>
- [4] <https://petsone.pk/>