

Be Friends: An Android Application

(Friendship based on your Personality Traits)



By:

Muhammad Luqman

Supervised By

Ifrah Farrukh Khan

**Department of Computer Science
Quaid-i-Azam University, Islamabad.**

BS Computer Science

(2015-2019)

ACKNOWLEDGEMENT

First of all, I would like to express my gratefulness to **ALMIGHTY ALLAH** Whose blessings and guidance has been a real source of my success and learning and Who gave me the ability and knowledge to complete this project. Countless salutation upon the Holy **Prophet Hazrat Muhammad (S.A.W.)** source of knowledge for enlightening with the essence of faith in ALLAH and guiding the mankind, the true path of life.

I also thank my parents for the unceasing encouragement, support and attention. I am also grateful to my siblings who supported me through this venture.

I am especially thankful to my supervisor **Ifrah Farrukh Khan** for guiding me throughout the project. I am extremely thankful and indebted to her for sharing expertise, and sincere and valuable guidance and encouragement extended to me.

I take this opportunity to express gratitude to all the faculty members of the Department for their help and support throughout this journey.

I am also thankful to all my friends for encouraging and helping me, without which the completion of this project would not have been possible.

Muhammad Luqman (2015-2019)

ABSTRACT

Be Friends is an android based application that provides social platform to users for making friends according to their personality traits. Application starts by asking the necessary information from the user (Required for signup) like name, email, password etc. In addition, it also asks the user to fill the questionnaire about his personality traits after that, system calculates score of filled questionnaire and store information in user profile.

Each and every user follows the same process to sign up. When the user is registered in the system, the friendship suggestions are given to each user on the basis of friendship selection model for personality traits. The user can send the request to the suggested user. Once the friendship is made then users can communicate with each other.

Table of Contents

List of Figures.....	viii
List of Tables	ix
1 Chapter-1: Software Project Management Plan	1
1.1 Introduction	2
1.2 Project Overview	2
1.3 Project Deliverables	2
1.4 Project Organization.....	2
1.4.1 Software Process Model	2
1.4.2 Roles and Responsibilities	3
1.4.3 Responsibilities	3
1.5 Tools and Techniques.....	3
1.6 Project Management Plan.....	3
1.6.1 Tasks	4
1.6.2 Resources Needed.....	5
1.6.3 Timetable	6
1.6.4 Gantt Chart:.....	7
2 Chapter-2: Project Background and Understanding	8
2.1 Introduction	9
2.2 Big Five Personality Traits.....	9
2.3 Big Five Personality Traits and Facets.....	10
2.4 Table 2. Scale	11
2.4.1 The Big Five Inventory (BFI).....	11
2.4.2 Scoring	12
2.5 Big Five Personality Traits and Friendship Satisfaction.....	13
2.6 Friendship Selection and BFPT Model	13
2.6.1 Selecting Friends.....	13
2.6.2 Being Selected as a Friend.....	13
2.6.3 Selecting Similar Friends.....	13
2.7 Decisive Outcome and Assumptions.....	14
3 Chapter-3: Software Requirements Specification.....	15

3.1	Introduction	16
3.1.1	Purpose.....	16
3.2	Product Scope.....	16
3.2.1	Major Functions:.....	16
3.2.2	Major Inputs:.....	16
3.2.3	Major Outputs:	16
3.2.4	Constraint:.....	16
3.3	Objective	17
3.4	Definitions, Acronyms and Abbreviations.....	17
3.5	Specific Requirements.....	17
3.5.1	External Interface Requirements.....	17
3.6	Functional Requirements.....	18
3.6.1	Use-Case Model.....	18
3.6.2	Use Case Diagram.....	19
3.6.3	Use-Cases Description:.....	20
3.6.4	System Sequence Diagrams.....	35
3.7	Software Quality Attributes	37
3.8	Other non-functional Requirements.....	37
3.9	Domain Model.....	38
3.10	Database Requirements	38
4	Chapter-4: Software Design Document.....	39
4.1	Introduction	40
4.2	Design Overview	40
4.2.1	Requirements Traceability Matrix	40
4.3	System Architecture Design.....	41
4.3.1	Architecture Diagram.....	41
4.4	User Interfaces.....	42
4.4.1	Sign Up	42
4.4.2	Login	43
4.4.3	Friends List	44
4.4.4	Suggested Friends List.....	44
4.5	Sequence Diagram.....	44
4.5.1	SD for Use-Case Respond to Request	45

4.5.2	SD for Use-Case Sign Up	45
4.5.3	SD for Use-Case Start Chat	46
4.5.4	SD for Use-Case View Notification	46
4.6	Class Diagram	47
5	Chapter-5: Software Implementation Document.....	48
5.1	Introduction	49
5.2	Selected Platform	49
5.3	Selected Operating System.....	49
5.4	Selected Language.....	49
5.5	Interfaces of Implemented System.....	49
5.5.1	Login Interface.....	49
5.6	Signup Interface	50
5.7	View Profile Interface	51
5.8	View Friend Suggestions	51
5.9	Chat Interface	52
6	Chapter-6: Software Test Document.....	53
6.1	Introduction	54
6.1.1	Test Approach.....	54
6.2	Test Plan.....	54
6.2.1	Features To Be Tested	54
6.2.2	Features Not To Be Tested	54
6.2.3	Testing Tools And Environments	54
6.3	Test Cases.....	55
6.3.1	TC-1 Login.....	55
6.3.2	TC-2 View Notification	55
6.3.3	TC-3 View Suggested Friend.....	56
6.3.4	TC-4 View Friend List.....	56
6.3.5	TC-5 Remove Friends from list	57
7	Chapter-7: Conclusion and Future Enhancements.....	58
7.1	Conclusion.....	59
7.2	Future Enhancement.....	59
7.2.1	iOS Version.....	59
7.2.2	Connecting with other Social Applications	59

7.2.3 Video and Audio Communication	59
REFERENCES.....	60

List of Figures

Figure 1.1 Timeline View of SPMP	6
Figure 1.2 Gantt Chart	7
Figure 3.1 Use Case Diagram	19
Figure 3.2 SSD for Use-case from Friend List	35
Figure 3.3 SSD for Use-case Respond to a Request.....	35
Figure 3.4 SSD for Use-case Signup	36
Figure 3.5 SSD for Use-case Start Chat.....	36
Figure 3.6 SSD for Use-case View Notification.....	36
Figure 3.7 Domain Model.....	38
Figure 4.1 Architecture Diagram	42
Figure 4.2 Signup.....	42
Figure 4.3 Signup Questionnaire	43
Figure 4.4 Login.....	43
Figure 4.5 Friends List.....	44
Figure 4.6 Suggested Friends List	44
Figure 4.7 SD for Respond to Request	45
Figure 4.8 SD for Sign Up	45
Figure 4.9 SD for Start Chat	46
Figure 4.10 SD for View Notification	46
Figure 4.11 Class Diagram	47
Figure 5.1 Login Interface	49
Figure 5.2 Signup Interface (a).....	50
Figure 5.4 Signup Interface (Questionnaire)	50
Figure 5.6 View Profile Interface	51
Figure 5.5 View Profile Interface	51
Figure 5.7 Suggested Friend List Interface.....	51
Figure 5.8 Chat Interface	52

List of Tables

Table 1.1 Tools and Techniques	3
Table 2.1 Personality Traits Facets	10
Table 3.1 Terms and Abbreviations.....	17
Table 3.2 UC-1: Signup	20
Table 3.3 UC-2: Login.....	21
Table 3.4 UC-3: Logout.....	22
Table 3.5 UC-4: Send Request.....	23
Table 3.6 UC-5: View Notification	24
Table 3.7 UC-6: Respond to a Request.....	25
Table 3.8 UC-7: Decline Request	26
Table 3.9 UC-8: Accept Request	27
Table 3.10 UC-9: View Suggested Friends	28
Table 3.11 UC-10: Remove from Friend List.....	29
Table 3.12 UC-11: View Profile.....	30
Table 3.13 UC-12: Start Chat	31
Table 3.14 UC-13: View Profile.....	32
Table 3.15 UC-14: Generate Suggested Friends List	33
Table 3.16 UC-15: Generate Profile	34
Table 4.1 Requirements Traceability Matrix	40
Table 6.1 TC-1 Login	55
Table 6.2 TC-2 View Notification	55
Table 6.3 TC-3 View Suggested Friends.....	56
Table 6.4 TC-4 View Friend List.....	56
Table 6.5 TC-5 Remove Friend from List	57

1 Chapter-1: Software Project Management Plan

1.1 Introduction

This chapter describes the introduction to Be Friends project and Software Project Management Plan (SPMP) including goals of the project and includes a description of the deliverables and deadlines and Project Organization and Project Management Plan.

1.2 Project Overview

This Project will provide people a social platform by which they can make new friends by using this application. Those people who want to make new friends on the basis of their personality traits, this is the best platform for them. As it provides the functionality of friend's suggestion on the basis of personality traits by filling BFI personality traits questionnaire.

1.3 Project Deliverables

Below are the 5 the project deliverables:

- Software Project Management Plan
- Software Requirement Specification
- Software Design Description
- Implementation
- Software Test Documentation

1.4 Project Organization

Project Organization describes the software process model used for this project, roles and responsibilities, tools and techniques to be used in this project.

1.4.1 Software Process Model

The process model which I used to complete my project is the Incremental Model as.

- The **cost of accommodating changing** customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less than that's required with waterfall model.
- It's easier to get **customer feedback** on the work done during development than when the system is fully developed, tested, and delivered.

1.4.2 Roles and Responsibilities

Following are the roles that define the foundations of the project:

- **Project Management Plan**
Tasks to be performed in every phase of development, progress, and milestone according to the plan.
- **Requirements Specification and Analysis**
Specification and analysis of every phase of the project.
- **Design**
Design of complete project requirements.
- **Implementation (source code)**
The Development phase of source code for the application.
- **Test Plan**
Testing the application, using various testing tools and techniques.
- **Final Product Deliverable**
After successful completion of all the above phases, a final deliverable product will be provided.

1.4.3 Responsibilities

As I am a single developer for this project so there is no division of roles and responsibilities. All the responsibilities regarding the complete project are on me.

1.5 Tools and Techniques

The following table shows tools and techniques to be used in the project:

Table 1.1 Tools and Techniques

Sr.	Tools and Techniques
1	MS Word, MS Visio 2016, Argo Uml, Pages
2	Firebase
3	Android Studio
4	ProjectLibre

1.6 Project Management Plan

Following is the description of the project management plan for this project (Android based Application). This document describes the tasks and deliverables and

milestones throughout the project management plan for this application and defines resources to be needed.

1.6.1 Tasks

There are two main phases in the project plan. Requirement and analysis and design phase of the project. In the requirements and analysis phase, the major tasks are to identify requirements, define use cases, develop analysis model, develop SRS and review SRS.

In the design phase, the major tasks are to develop to design using object-oriented approach. Software Test document, validate input, User manual, develop models and evaluate design.

Following is the description of major tasks of both analysis and design phases.

1.6.1.1 Requirement and Analysis

- **Identify requirements**

The main goal is to define requirements by meeting stakeholders.

- **Define use cases**

I have to define use-cases and make a use case diagram.

- **Develop SRS**

In SRS we define functional and nonfunctional requirements and develop software requirement specification document. It contains project scope, purpose, and objective.

- **Review SRS**

Review the SRS document.

- **Design Phase**

In this phase, the detailed design and interface design will be included.

- **Develop Design**

Develop architectural design and interface design using object-oriented approach.

Develop a system sequence diagram and a class diagram.

- **Evaluate design**

Evaluate and verify the design.

- **Software Test Documentation**

Test plans and test cases are included in this document.

- **Implementation Phase**

Describe how the system will be implemented.

1.6.2 Resources Needed

These are the following resources to needed to complete this project.

1.6.2.1 Software:

- Android Studio
- Argo UML
- Firebase
- ProjectLibre
- MS Word, Pages

1.6.2.2 Human Resources:

- Muhammad Luqman
- Supervisor

1.6.2.3 Hardware Resources:

- PC or Laptop
- Android Phone

1.6.3 Timetable

This figure shows the project timetable:

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		libe Friends	238 days?	2/18/19 8:00 AM	1/15/20 5:00 PM		
2		Problem understanding	1 day	2/18/19 8:00 AM	2/18/19 5:00 PM		
3		Software Project Management Plan	2 days	2/18/19 8:00 AM	2/19/19 5:00 PM		M Luqman;Laptop PC/...
4		Write Introduction	1 day	2/18/19 8:00 AM	2/18/19 5:00 PM		
5		Define Project Organization	1 day	2/18/19 8:00 AM	2/18/19 5:00 PM		
6		Define Project Management Plan	1 day	2/19/19 8:00 AM	2/19/19 5:00 PM	5	Project Libre
7		Analysis and Requirements	98 days?	2/20/19 8:00 AM	7/5/19 5:00 PM	6	M Luqman;Laptop PC/...
8		Software Requirement Specification	52 days?	2/20/19 8:00 AM	5/2/19 5:00 PM		
9		Give Introduction and Overview	0 days	2/20/19 8:00 AM	2/20/19 8:00 AM		
10		Define Purpose ,Scope and objective	1 day	2/20/19 8:00 AM	2/20/19 5:00 PM		
11		Review and refine scope	1 day	2/21/19 8:00 AM	2/21/19 5:00 PM	10	Madam Ifrah
12		Identify Specific Requirements	4 days	2/22/19 8:00 AM	2/27/19 5:00 PM	11	
13		Identify Use Cases	2 days	2/20/19 8:00 AM	2/21/19 5:00 PM		
14		Make UseCase Diagram	7 days	2/22/19 8:00 AM	3/4/19 5:00 PM	13	MS Visio/Argo UML
15		Review and Refine UC Diagram	1 day	3/11/19 8:00 AM	3/11/19 5:00 PM	14	Madam Ifrah
16		Define UseCase descriptions	13 days	3/5/19 8:00 AM	3/21/19 5:00 PM	14	
17		Define System Attributes	5 days	3/25/19 8:00 AM	3/29/19 5:00 PM	16	
18		Make Domain Model	5 days?	4/1/19 8:00 AM	4/5/19 5:00 PM		
19		Identify DataBase Requirements	2 days	4/8/19 8:00 AM	4/9/19 5:00 PM	17	
20		Identify Entities	2 days?	4/23/19 8:00 AM	4/24/19 5:00 PM		
21		Make Entity Relationship Diagram	2 days?	4/26/19 8:00 AM	4/29/19 5:00 PM		
22		Review ERD	1 day?	4/30/19 8:00 AM	4/30/19 5:00 PM		
23		Review and Refine SRS	1 day	5/2/19 8:00 AM	5/2/19 5:00 PM		Madam Ifrah
24		Software Design Description	17 days?	5/3/19 8:00 AM	5/27/19 5:00 PM	23	M Luqman;Laptop PC/...
25		Give Introduction and Overview	1 day	5/3/19 8:00 AM	5/3/19 5:00 PM		
26		Explain Interfaces	1 day?	5/3/19 8:00 AM	5/3/19 5:00 PM		
27		Make System Architectural Design	2 days	5/3/19 8:00 AM	5/6/19 5:00 PM		MS Visio/Argo UML
28							
29							
30		Review and Refine SD	0 days	5/14/19 5:00 PM	5/14/19 5:00 PM	29	Madam Ifrah
31		Identify Classes	1 day	5/15/19 8:00 AM	5/15/19 5:00 PM	30	
32		Make Class Diagram	5 days	5/16/19 8:00 AM	5/22/19 5:00 PM	31	MS Visio/Argo UML
33		Review and Refine Class Diagram	1 day	5/23/19 8:00 AM	5/23/19 5:00 PM	32	Madam Ifrah
34		Review and Refine Software Design D...	2 days	5/24/19 8:00 AM	5/27/19 5:00 PM	33	Madam Ifrah
35		Software Test Document	7 days	6/20/19 8:00 AM	6/28/19 5:00 PM		M Luqman;Laptop PC/...
36		Introduction and Test Plan	2 days	6/20/19 8:00 AM	6/21/19 5:00 PM		
37		Make Test Cases	2 days	6/24/19 8:00 AM	6/25/19 5:00 PM	36	
38		Review and Refine Test Document	3 days	6/26/19 8:00 AM	6/28/19 5:00 PM	37	Madam Ifrah
39		Review Analysis and Design Document	0 days	7/1/19 5:00 PM	7/1/19 5:00 PM	38	Madam Ifrah
40		Provide 1st Deliverable	4 days	7/2/19 8:00 AM	7/5/19 5:00 PM	39	
41		Project Implementation	136 days	7/10/19 8:00 AM	1/15/20 5:00 PM	40	Android Studio;M Luqm...

Figure 1.1 Timeline View of SPMP

1.6.4 Gantt Chart:

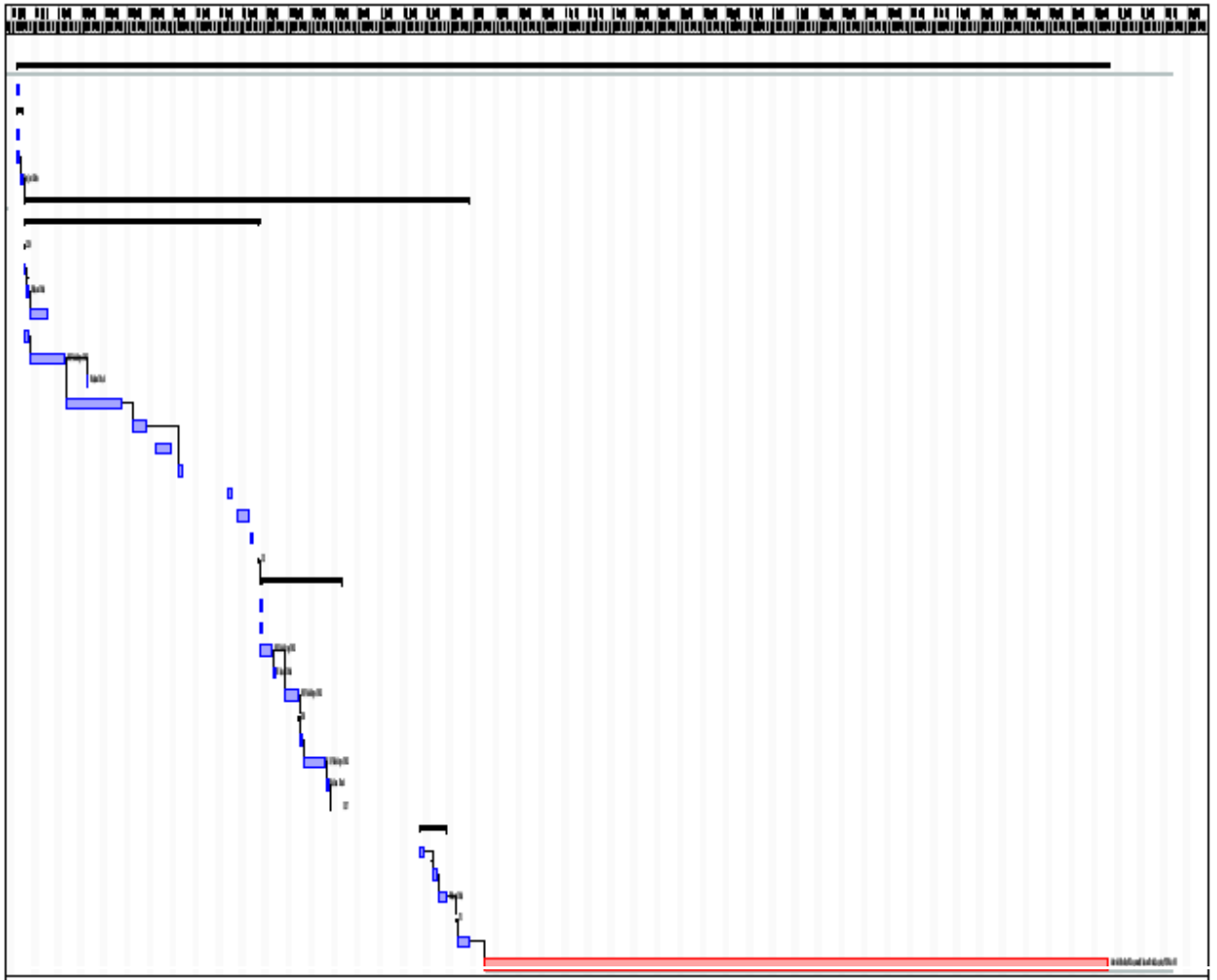


Figure 1.2 Gantt Chart

2 Chapter-2: Project Background and Understanding

2.1 Introduction

This chapter discusses the Big Five Personality Traits Model which highlights the five factors i.e. Openness to Experience, Conscientiousness, Extraversion, Agreeableness and Neuroticism. It also describes big five Inventory as well as Friendship Selection Model for Personality Traits.

2.2 Big Five Personality Traits

Among the best developed models concerning personality traits one is the Big Five personality model. The five factors have shown high rank-order and mean level stability over time and exist in diverse cultures.

Following are the big five personality traits that are disclosed in a person:

- **Openness to Experience** - Usually people who like learning new things and enjoying new experiences score high in openness. Openness includes features such as being insightful and imaginative and having a wide range of interests.
- **Conscientiousness** - People with a high level of awareness are reliable and prompt. Traits include organization, methodology, and thoroughness.
- **Extraversion** - Extraverts derive their energy from interacting with others, while introverts receive their energy from within. Extraversion requires the characteristics of power, voice, and assertiveness.
- **Agreeableness** - These people are friendly, cooperative, and sympathetic. People with low consent may be farther away. Characteristics include being kind, affectionate and compassionate.
- **Neuroticism** - Neuroticism is sometimes referred to as emotional stability. This dimension has to do with one's emotional stability and degree of negative emotions. People with high neuroticism scores often experience emotional instability and negative emotions. Traits include being tense and moody.

2.3 Big Five Personality Traits and Facets

Each of the personality traits further have been divided into the following personality facets:

Table 2.1 Personality Traits Facets

Big Five Dimensions	Facet (and correlated trait adjective)
Extraversion vs. Introversion	Gregariousness (sociable) Assertiveness (forceful) Activity (energetic) Excitement-seeking (adventurous) Positive emotions (enthusiastic) Warmth (outgoing)
Agreeableness vs. Antagonism	Trust (forgiving) Straightforwardness (not demanding) Altruism (warm) Compliance (not stubborn) Modesty (not show-off) Tender-mindedness (sympathetic)
Conscientiousness vs. Lack of Direction	Competence (efficient) Order (organized) Dutifulness (not careless) Achievement striving (thorough) Self-discipline (not lazy) Deliberation (not impulsive)
Neuroticism vs. Emotional Stability	Anxiety (tense) Angry hostility (irritable) Depression (not contented) Self-consciousness (shy) Impulsiveness (moody) Vulnerability (not self-confident)
Openness vs. Closedness to Experience	Ideas (curious) Fantasy (imaginative) Aesthetics (artistic) Actions (wide interests) Feelings (excitable) Values (unconventional)

2.4 Scale

2.4.1 The Big Five Inventory (BFI)

Here are a number of characteristics that may or may not apply to you. For example, do you agree that you are someone who likes to spend time with others? Please write a number next to each statement to indicate the extent to which you agree or disagree with that statement.

- Disagree strongly - 1
- Disagree a little - 2
- Neither agree nor disagree - 3
- Agree a little - 4
- Agree Strongly - 5

Following are the questions one normally sees himself/herself as someone:

1. Is talkative
2. Tends to find fault with others
3. Does a thorough job
4. Is depressed, blue
5. Is original, comes up with new ideas
6. Is reserved
7. Is helpful and unselfish with others
8. Can be somewhat careless
9. Is relaxed, handles stress well
10. Is curious about many different things
11. Is full of energy
12. Starts quarrels with others
13. Is a reliable worker
14. Can be tense
15. Is ingenious, a deep thinker
16. Generates a lot of enthusiasm
17. Has a forgiving nature
18. Tends to be disorganized
19. Worries a lot
20. Has an active imagination

21. Tends to be quiet
22. Is generally trusting
23. Tends to be lazy
24. Is emotionally stable, not easily upset
25. Is inventive
26. Has an assertive personality
27. Can be cold and aloof
28. Perseveres until the task is finished
29. Can be moody
30. Values artistic, aesthetic experiences
31. . Is sometimes shy, inhibited
32. Is considerate and kind to almost everyone
33. Does things efficiently
34. Remains calm in tense situations
35. Prefers work that is routine
36. Is outgoing, sociable
37. Is sometimes rude to others
38. Makes plans and follows through with them
39. Gets nervous easily
40. Likes to reflect, play with ideas
41. Has few artistic interests
42. Likes to cooperate with others
43. Is easily distracted
44. Is sophisticated in art, music, or literature

2.4.2 Scoring

BFI scale scoring (“R” denotes reverse-scored items):

Extraversion: 1, 6R, 11, 16, 21R, 26, 31R, 36.

Agreeableness: 2R, 7, 12R, 17, 22, 27R, 32, 37R, 42.

Conscientiousness: 3, 8R, 13, 18R, 23R, 28, 33, 38, 43R.

Neuroticism: 4, 9R, 14, 19, 24R, 29, 34R, 39.

Openness: 5, 10, 15, 20, 25, 30, 35R, 40, 41R, 44 [3].

2.5 Big Five Personality Traits and Friendship Satisfaction

Among the BFPT, extraversion, agreeableness, conscientiousness and (low) neuroticism were associated with higher levels of friendship satisfaction. How? Neuroticism and friendship satisfaction are less likely found to get along with each other. Theoretically, neuroticism seems highly relevant for friendship satisfaction, but a study found that people high in neuroticism experienced more conflict in their friendships than those low in neuroticism. Also, Openness has comparatively less influence on the prediction of friendship satisfaction in the Big Five predictor [5].

2.6 Friendship Selection and BFPT Model

In three different ways Big Five personality traits affects friendship selection. First, on the basis of personality differences in the number of friends the individuals select. Second, personality affects at what extent an individual can be selected as a friend. Lastly, having similar personality affects friendship selection process.

2.6.1 Selecting Friends

Extraversion is primarily related to one's level of social interaction and is assumed to represent the basic desire to receive benefits from social situations, making extraverted individuals more likely to experience positive impact in social situations. Extraverted individuals may be more likely to choose friends because of this increased positive effect during social interactions.

2.6.2 Being Selected as a Friend

Agreeable individuals tend to exhibit more prosocial and selfless behaviors, such as greater empathy, greater willingness to cooperate, and more conflict strategies based on the integration of views and needs of both partners. These kinds of selfless and sociable behaviors make Agreeable individuals more attractive as good friends. Thus, Agreeableness enhance ability of being selected as a friend.

2.6.3 Selecting Similar Friends

The theory of similarity-attraction emphasizes on the role of similarity in the choice of friendship in personality traits, irrespective of the major effects of personality traits. In two specific traits, Extraversion and Agreeableness, there is similarity that may enhance the selection of friendships. Next, Extraversion tends to be directly linked to the types of communication used to get to know with peers. Extraverted individuals appear to be more talkative and outgoing, whereas during initial interaction, introverts

appear to act more shyly and inhibitedly. When two extraverted individuals meet, they can more easily predict the responses of each other and enjoy this interaction than with a more reserved, shy person. On the other hand, introverts may enjoy more relaxed communication with each other than with more outgoing, talkative extraverts and may also be more able to predict the responses of each other. However, recent studies suggest that agreeability is conveyed mainly by altruistic behavior. If one of the two individuals act egoistically, the person acting altruistically loses more than the one acting egoistically. If both individuals act egoistically, they gain less than when both acts altruistically but at least more than when one of them acts egoistically. Similarities in altruistic behaviors can therefore be expected to result in more beneficial results than dissimilarities in altruistic behaviors. In a similar vein, individuals selecting others with similar levels of Agreeableness may benefit more from this relationship than in Agreeableness selecting those that differ from them. Similarity in Agreeableness can therefore improve the selection of friendships [4].

2.7 Decisive Outcome and Assumptions

Results suggest that in initial friendship selection processes, Big Five factors play an important but differentiated role. Extraversion seems to increase the number of friends selected, while Agreeableness appears to increase the number of friends selected. Moreover, real similarity in these personality traits as well as similarity in Openness appears to play an even greater role in enhancing the selection of friendships over time.

3 Chapter-3: Software Requirements Specification

3.1 Introduction

Software Requirements Specification (SRS) is a document that describes the complete description of how the system is expected to perform its functionality. Functional and non-functional requirements such as performance goals and descriptions of quality attributes are detailed completely in the SRS document.

3.1.1 Purpose

The purpose of this chapter is to describe the functional and non-functional requirements that our project needs. This chapter clears that what end users of the project want and what would be the output of the system. Another purpose is to understand the requirements and analyze them that how those requirements would help in implementing the project with actual product features that end users want.

3.2 Product Scope

The scope of the system represents the boundary of the system. Be Friends application is an Android mobile application that provides a social platform to users. By this application users can make new friends on the basis of his personality traits.

3.2.1 Major Functions:

- Creating or generating a user profile on the basis of Questionnaire of personality traits filled by user on the time of signup.
- Show friend suggestion to user according to user's personality traits
- It will allow the user to send a friend request to suggested friends.
- The System will provide functionality to the user to chat with his friends

3.2.2 Major Inputs:

- The user will enter his information for sign-up (Name, username, password, email etc.)
- The user will fill a questionnaire

3.2.3 Major Outputs:

- The system will generate the user profile.
- The system shows friend suggestion to user by comparing users' profile on the basis of personality traits of other users.

3.2.4 Constraint:

- There is a time constraint as the project to be completed in two semesters.

- This Project is Android based Application.

3.3 Objective

The objective of the system is to create an Android application that will provide a good social platform to people that are interested making friends on the basis of their personality.

3.4 Definitions, Acronyms and Abbreviations

Table 3.1 Terms and Abbreviations

Terms and Abbreviations	Meaning
SPMP	Software Project Management Plan
SRS	Software Requirements Specification
DM	Domain Model
UC	Use-Case
System	Be Friends Application
BFI	Big Five Inventory

3.5 Specific Requirements

This section of Software Requirement Specification document will provide detail about interfaces through which system communicate with the user.

3.5.1 External Interface Requirements

3.5.1.1 User Interfaces:

As it is an Android application so all the android phones are compatible with this application. It has a friendly user interface; screen display of the application will be responsive as it will not matter what is the size of the user mobile screen.

3.5.1.2 Hardware Interfaces

As this project is for an Android operating system, so it will need an Android phone to execute or run this application. The user should have a good internet connection.

3.5.1.3 Software Interface

Android operating system having minimum 4.4 or above will be able to use this application.

3.6 Functional Requirements

This section describes the abstract level detail of use cases and the actor who perform that particular use case.

3.6.1 Use-Case Model

This section provides a detailed description of use cases in the Be Friends Application.

The list of all Use-cases given below:

- Signup
- Login
- Logout
- Send Request
- View Notification
- Respond to a Friend Request
- Decline Request
- Accept Request
- View Suggested Friends
- Remove from Friend List
- View Friends List
- Start Chat
- View Profile
- Generate Suggested Friend List
- Generate Profile

3.6.2 Use Case Diagram

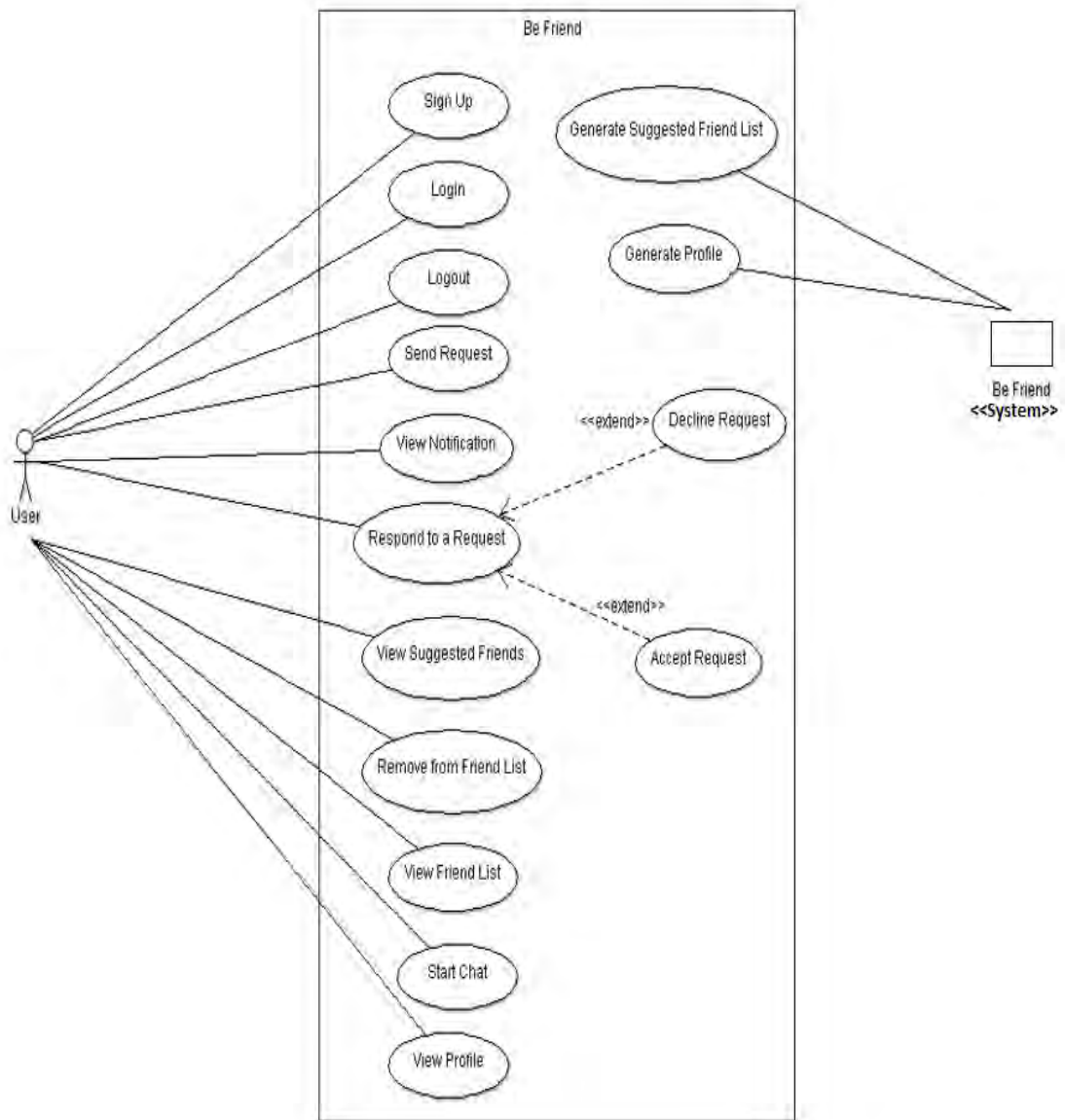


Figure 3.1 Use Case Diagram

3.6.3 Use-Cases Description:

- UC-1: Signup

Table 3.2 UC-1: Signup

Use-case ID	UC-1
Use-case Name	Signup
Primary Actor	User
Stakeholders & Interest	User: Wants to Sign up the Be Friends System.
Pre-Conditions	User installed and executed the application on his device.
Post-Conditions	User signup successfully.
Main Success Scenario	<ol style="list-style-type: none">1. User open the Be Friends application.2. User selects the signup option.3. User enters signup details (name, username, password and questionnaire etc.).4. User selects submit option.
Alternative flows or Extensions	<p>a*. System Fails. System will roll back all the changes.</p> <p>4a. User do not fill all the required fields. System will ask him to fill all the required fields.</p>

- **UC-2: Login**

Table 3.3 UC-2: Login

Use-case ID	UC-2
Use-case Name	Login
Primary Actor	User
Stakeholders & Interest	User: Wants to Login the Be Friends System.
Pre-Conditions	User should be registered in system and have valid username and password.
Post-Conditions	User successfully logged in.
Main Success Scenario	<ol style="list-style-type: none"> 1. User open the Be Friends application. 2. User selects the login option. 3. User enters his username and password. 4. User selects submit option. 5. System Validates username and password and allows the user to login the system.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes</p> <p>5a. Invalid User name or Password.</p> <p>System shows error message to user and asks to enter valid Id and Password.</p>

- **UC-3: Logout**

Table 3.4 UC-3: Logout

Use-case ID	UC-3
Use-case Name	Logout
Primary Actor	User
Stakeholders & Interest	User: Wants to Logout from the Be Friends System.
Pre-Conditions	User must be logged in.
Post-Conditions	User successfully logged out.
Main Success Scenario	1. User selects the logout option.
Alternative flows or Extensions	a*. System Fails. System will roll back all the changes

- **UC-4: Send Request**

Table 3.5 UC-4: Send Request

Use-case ID	UC-4
Use-case Name	Send Request
Primary Actor	User
Stakeholders & Interest	User: Wants to send friend request to suggested friends.
Pre-Conditions	User should be logged in the system.
Post-Conditions	User successfully send friend request.
Main Success Scenario	<ol style="list-style-type: none"> 1. User open the Be Friends application. 2. User selects a suggested friend whom he wants to send request. 3. User enter option to send the request to selected friend. 4. System deliver his request to that user.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-5: View Notification**

Table 3.6 UC-5: View Notification

Use-case ID	UC-5
Use-case Name	View Notification
Primary Actor	User
Stakeholders & Interest	User: Wants to view the notifications.
Pre-Conditions	User should be logged in the system.
Post-Conditions	User successfully viewed the notification.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects the notification option. 2. System will show him all the notification.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-6: Respond to a Request**

Table 3.7 UC-6: Respond to a Request

Use-case ID	UC-6
Use-case Name	Respond to a Request
Primary Actor	User
Stakeholders & Interest	User: Wants to make new friends according to his own choice.
Pre-Conditions	User have at least a request pending.
Post-Conditions	User successfully responds to a request.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects Notification option. 2. System shows all the notifications to user 3. User selects a specific request. 4. User selects decline request option then system goes to UC-7 <p style="text-align: center;">OR</p> <p>User selects accept request option then system goes to UC-8</p>
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p> <p>1a. User does not have any friend request. System shows message.</p>

- **UC-7: Decline Request**

Table 3.8 UC-7: Decline Request

Use-case ID	UC-7
Use-case Name	Decline Request
Primary Actor	User
Stakeholders & Interest	User: Wants to reject a friend request.
Pre-Conditions	User Selected the decline request option.
Post-Conditions	User successfully rejected the friend request.
Main Success Scenario	1. System will put the declined suggestion at bottom of suggested friend list.
Alternative flows or Extensions	a*. System Fails. System will roll back all the changes.

- **UC-8: Accept Request**

Table 3.9 UC-8: Accept Request

Use-case ID	UC-8
Use-case Name	Accept Request
Primary Actor	User
Stakeholders & Interest	User: Wants to accept a friend request.
Pre-Conditions	User selected the accept request option.
Post-Conditions	Request successfully accepted.
Main Success Scenario	1. System adds requested person to the user's friends list.
Alternative flows or Extensions	a*. System Fails. System will roll back all the changes.

- **UC-9: View Suggested friends**

Table 3.10 UC-9: View Suggested Friends

Use-case ID	UC-9
Use-case Name	View Suggested friends
Primary Actor	User
Stakeholders & Interest	User: Wants to view suggested friend by the system to make new friends.
Pre-Conditions	User should be logged in the system.
Post-Conditions	User successfully view the suggested friend list.
Main Success Scenario	<ol style="list-style-type: none"> 1. User open the Be Friends application. 2. User selects an option to view the suggested friends. 3. System display him suggested friends list.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-10: Remove from Friend List**

Table 3.11 UC-10: Remove from Friend List

Use-case ID	UC-10
Use-case Name	Remove from Friend List
Primary Actor	User
Stakeholders & Interest	User: Wants to remove a friend from his friend list.
Pre-Conditions	User should have at least one friend.
Post-Conditions	User successfully removed a friend from his friend list.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects friend list option. 2. User selects a specific friend from friend list. 3. User selects unfriend option. 4. System will show a message.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-11: View Friend List**

Table 3.12 UC-11: View Profile

Use-case ID	UC-11
Use-case Name	View Friend List
Primary Actor	User
Stakeholders & Interest	User: Wants to view his friend list.
Pre-Conditions	User should have at least one friend.
Post-Conditions	User successfully viewed his friend list.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects an option for view friend list. 2. System will show him friend list.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-12: Start Chat**

Table 3.13 UC-12: Start Chat

Use-case ID	UC-12
Use-case Name	Start Chat
Primary Actor	User
Stakeholders & Interest	User: Wants to send message to his friend.
Pre-Conditions	User should be logged in the system.
Post-Conditions	User successfully send message to his friend.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects the chat option. 2. User selects a friend from list. 3. User types message. 4. User selects send option. 5. System delivers the message.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p> <p>7a. Message is not sent</p> <p>System shows the user resend option.</p>

- **UC-13: View Profile**

Table 3.14 UC-13: View Profile

Use-case ID	UC-13
Use-case Name	View Profile
Primary Actor	User
Stakeholders & Interest	User: Wants to view his or his friend's profile.
Pre-Conditions	User must have selected a person from his friend list or himself.
Post-Conditions	User successfully viewed a profile.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects username. 2. System shows him selected profile.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-14: Generate Suggested Friend List**

Table 3.15 UC-14: Generate Suggested Friends List

Use-case ID	UC-14
Use-case Name	Generate Suggested Friend List
Primary Actor	Be Friend
Stakeholders & Interest	User: wants to have friend suggestion matching to his personality.
Pre-Conditions	There should be a new signup.
Post-Conditions	Suggested Friend list successfully generated.
Main Success Scenario	<ol style="list-style-type: none"> 1. System will compare all the user's profile with newly signed user. 2. System will add users to suggested friend list.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

- **UC-15: Generate Profile**

Table 3.16 UC-15: Generate Profile

Use-case ID	UC-15
Use-case Name	Generate Profile
Primary Actor	Be Friends
Stakeholders & Interest	User: Wants to register himself.
Pre-Conditions	User should be selected submit option.
Post-Conditions	User successfully submitted signup form.
Main Success Scenario	<ol style="list-style-type: none"> 1. System will analyze the profile's information. 2. System will calculate the personality traits score against the profile. 3. System will store the profile in the database.
Alternative flows or Extensions	<p>a*. System Fails.</p> <p>System will roll back all the changes.</p>

3.6.4 System Sequence Diagrams

Below are system sequence diagrams of some important use cases showing the success scenario of respective use case.

3.6.4.1 SSD for Use-case Remove from Friend List

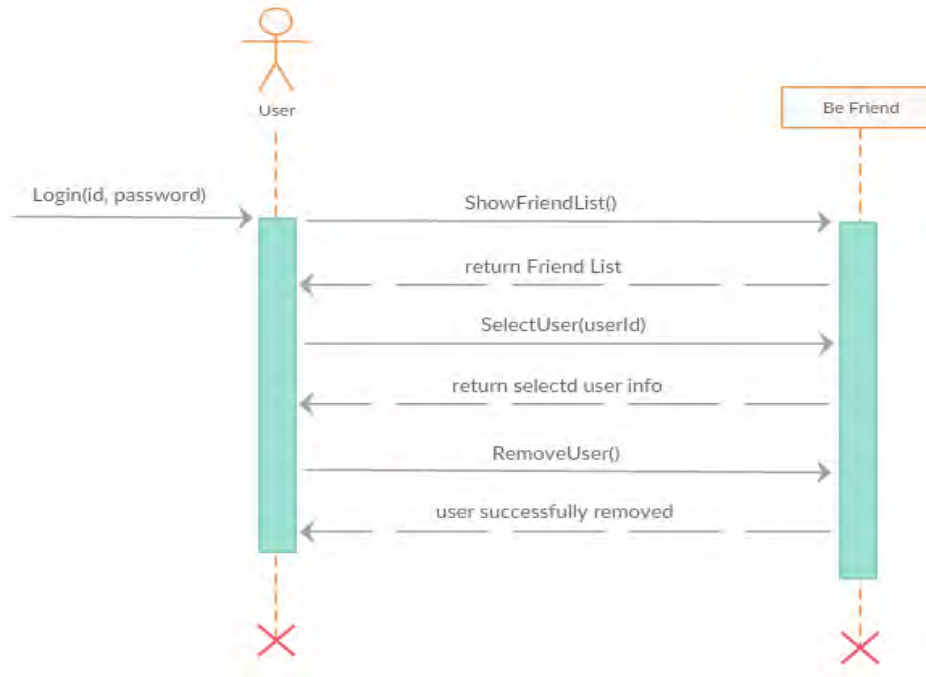


Figure 3.2 SSD for Use-case from Friend List

3.6.4.2 SSD for Use-case Respond to a Request

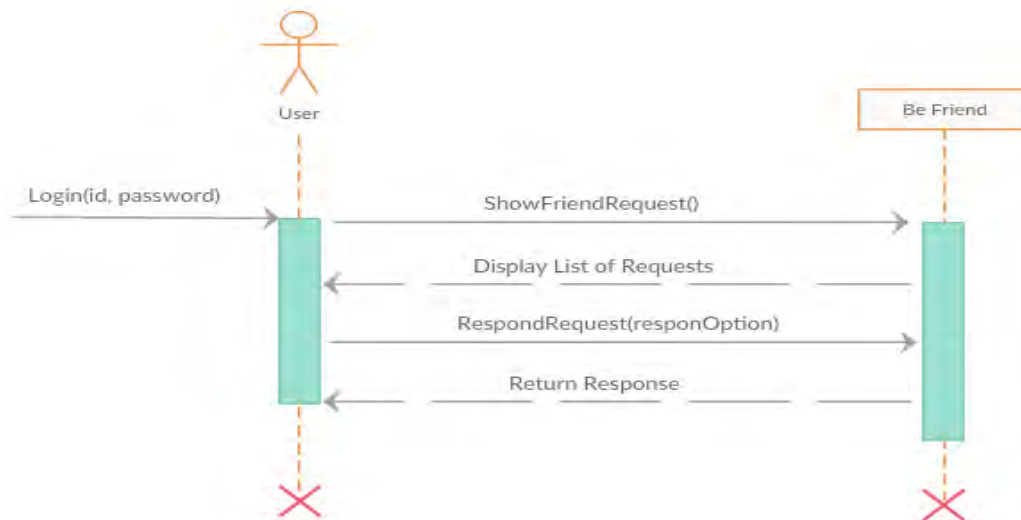


Figure 3.3 SSD for Use-case Respond to a Request

3.6.4.3 SSD for Use-case Signup

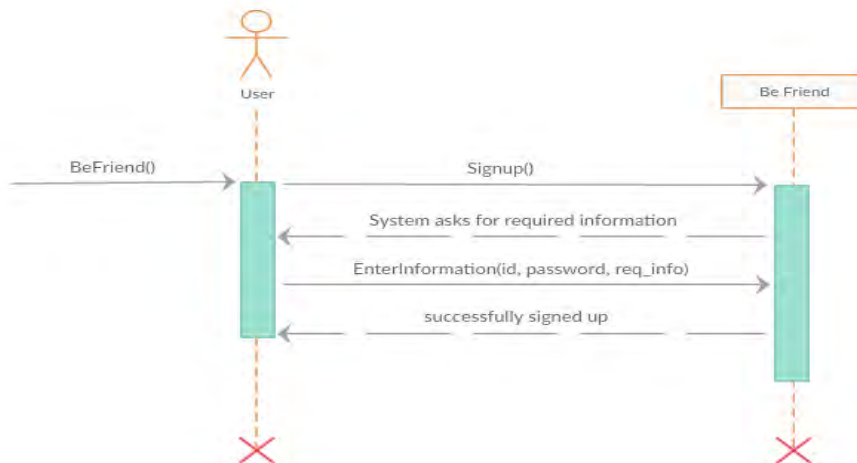


Figure 3.4 SSD for Use-case Signup

3.6.4.4 SSD for Use-case Start Chat

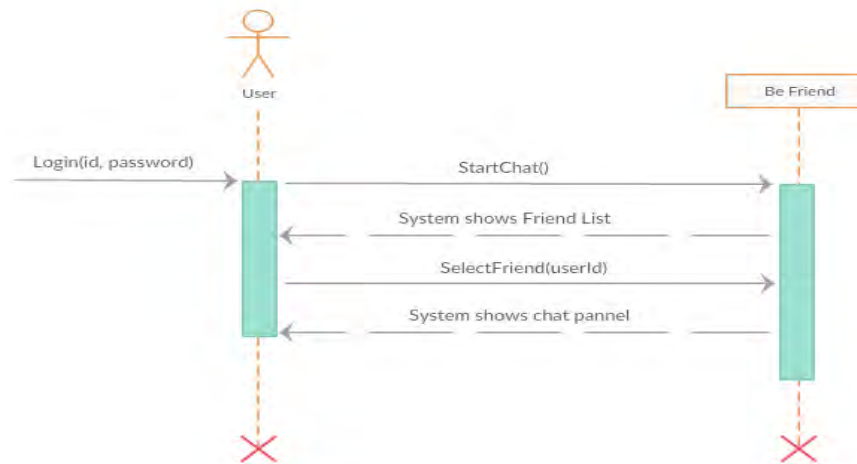


Figure 3.5 SSD for Use-case Start Chat

3.6.4.5 SSD for Use-case View Notification

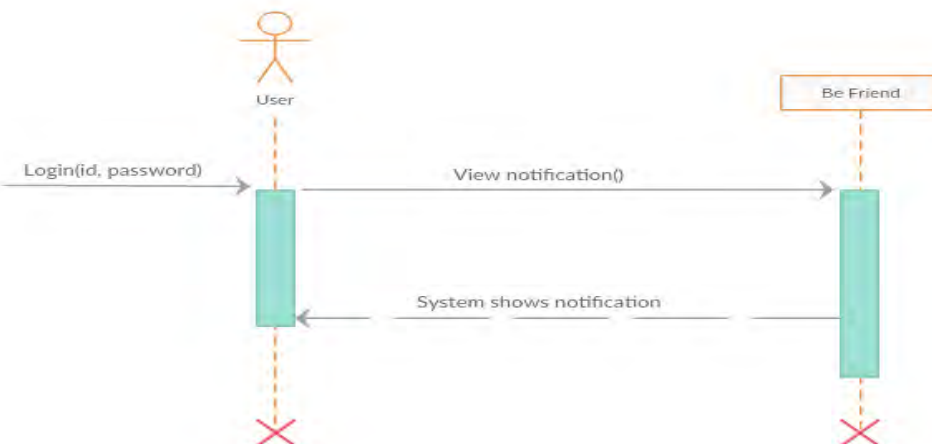


Figure 3.6 SSD for Use-case View Notification

3.7 Software Quality Attributes

Software Quality attributes define all aspects that affect run-time behavior, application design, and user experience.

- **Reliability**

The system should be reliable. The system should not crash or hang, other than an operating system error. System resists against any kind of damaged. The system should able to run for a long time and not get to be hanged or crashed.

- **Availability**

The system should provide his services on every time.

- **Maintainability**

We use the modular approach as it permits future modification and the application should be easy to extend. Write the code that favor implementation of new functions. The session of the user will be maintained as the user is active on the application. After delivery of the product, the system should be maintainable for any kind of flaws.

- **Portability**

The application can be run on all Android Devices having version 4.4 or above. It can't be run on any other platform. This system is Android-based and can be accessed through internet.

3.8 Other non-functional Requirements

- **Performance Requirements**

The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays.

- **Safety and Security Requirements**

- Information transmission should be securely transmitted to the Web Server without any changes in the information.
- The system should make sure that the users' data should remain secure and system prevent users from any other illegal activity.

3.9 Domain Model

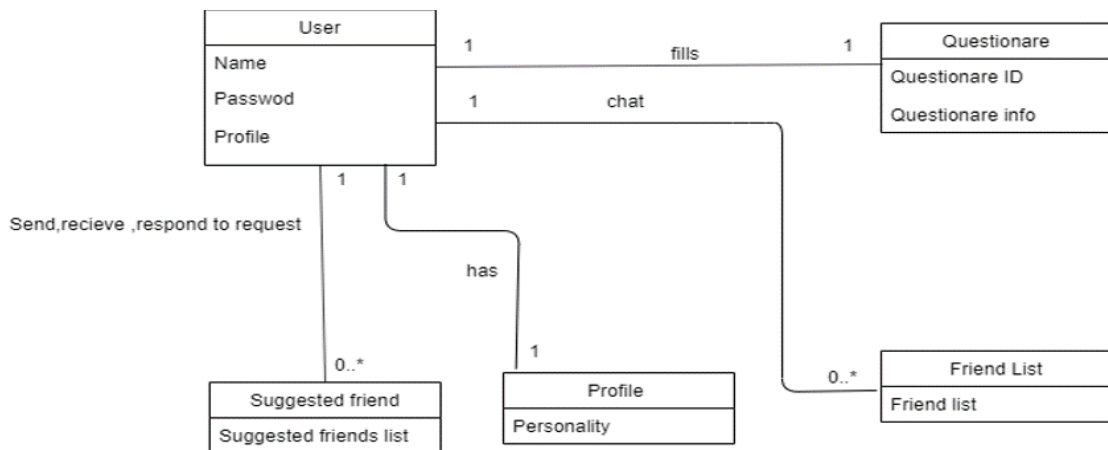


Figure 3.7 Domain Model

3.10 Database Requirements

The Database will be used Firebase to store user data and profile of the user after registration. In the process of registration questionnaire personality traits result, friends list, chat data and friend’s suggestion list will be stored in user’s profile. User profile data will be used to find comparison among the user’s profiles.

4 Chapter-4: Software Design Document

4.1 Introduction

The Software Design Document is a document that provides details about software design. It shows how the software system will be structured to satisfy the requirements. It consists of the description of how the software will meet the requirements. In this document, detailed data structures and algorithms are defined on the basis of architecture.

4.2 Design Overview

In the design phase of system, we describe the system's architecture and complete structure of the system. We discuss how user interacts with the system. In this phase we design interaction diagrams of system which includes class diagram that and sequence diagram which shows interaction between objects.

4.2.1 Requirements Traceability Matrix

Requirement Traceability Matrix (RTM) is a matrix that captures all the requirement and their traceability in a single matrix table. In other words, it is a document that maps and traces user requirement with test cases. The main purpose of Requirement Traceability Matrix is to check all test cases are covered or not. It describes how requirements are mapped with the system sequence diagram, sequence diagram, class diagram, interface and test cases. Whenever, any requirement needs to be changed then you don't need to find that requirement in the whole document. But you just have to go to the requirement traceability matrix.

Table 4.1 Requirements Traceability Matrix

Requirement Id	Requirement Name	System Sequence Diagram	Test Case	Sequence Diagram	Interface
UC-1	Sign Up	✓		✓	✓
UC-2	Login		✓		✓
UC-3	Logout				
UC-4	Send Request				

UC-5	View Notifications	✓	✓	✓	
UC-6	Respond to Friend Request	✓		✓	
UC-7	Decline Request	✓		✓	
UC-8	Accept Request	✓		✓	
UC-9	View Suggested Friends		✓		✓
UC-10	Remove Friend from List	✓	✓		
UC-11	View Friend List		✓		✓
UC-12	Start Chat	✓		✓	

4.3 System Architecture Design

Architectural design describes interaction between major structural elements of the system. Architecture design defines the way in which these components interact and the structure of data that are used by the components.

4.3.1 Architecture Diagram

System Architecture Diagram is used to represent the components of system and interaction between them. Be friend's system is based on 3-Tier architecture. 3-tier architecture is a software architecture in which the user interface (presentation layer), functional process logic (business rules), computer data storage and data access are developed and maintained as independent modules and interacting between components of system is shown in diagram. As each tier is independent, it is easy to make changes in the presentation layer (interface layer) without affecting the other two (business or data access layer).

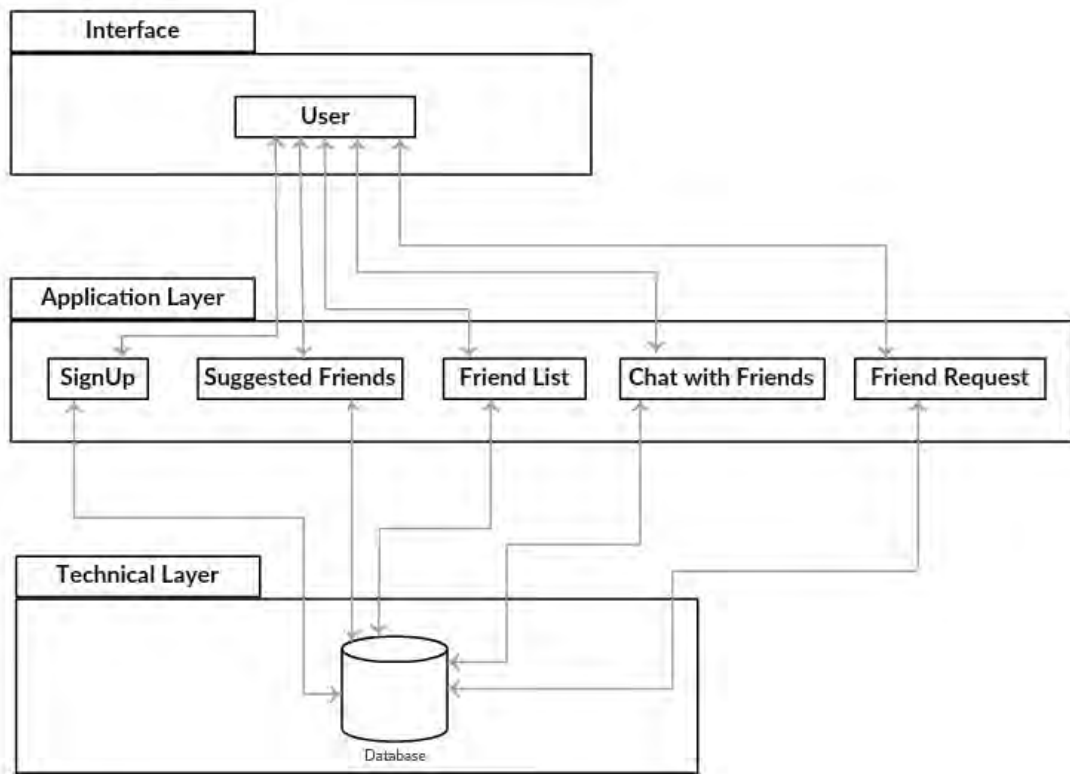


Figure 4.1 Architecture Diagram

4.4 User Interfaces

This section describes the prototypes for the interfaces which will be available for users. This section depicts the prototype for how the user interfaces will look like for this application.

4.4.1 Sign Up

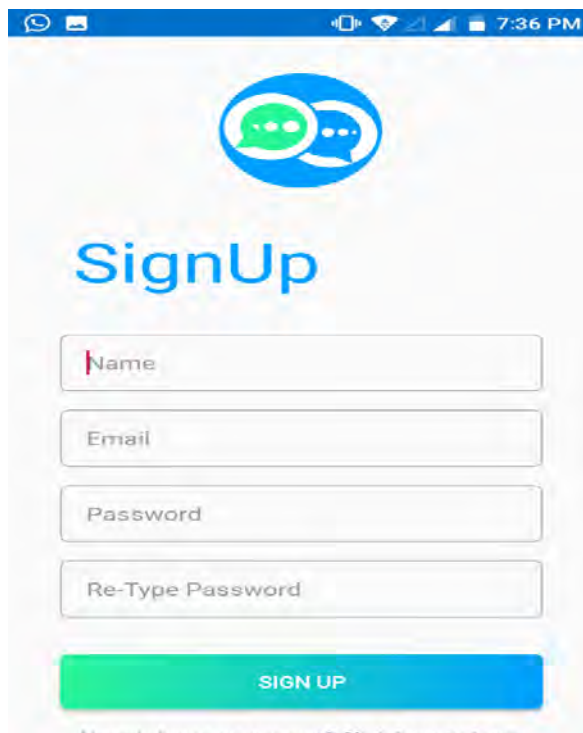


Figure 4.2 Signup

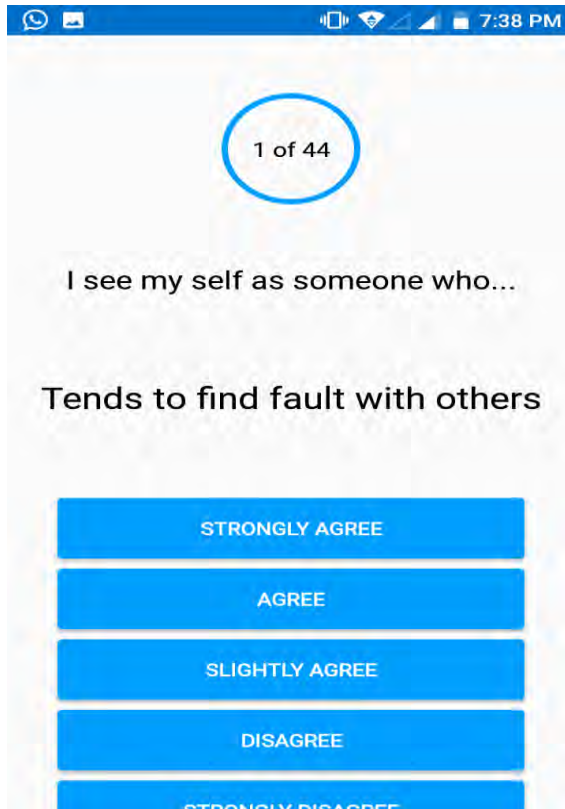


Figure 4.3 Signup Questionnaire

4.4.2 Login

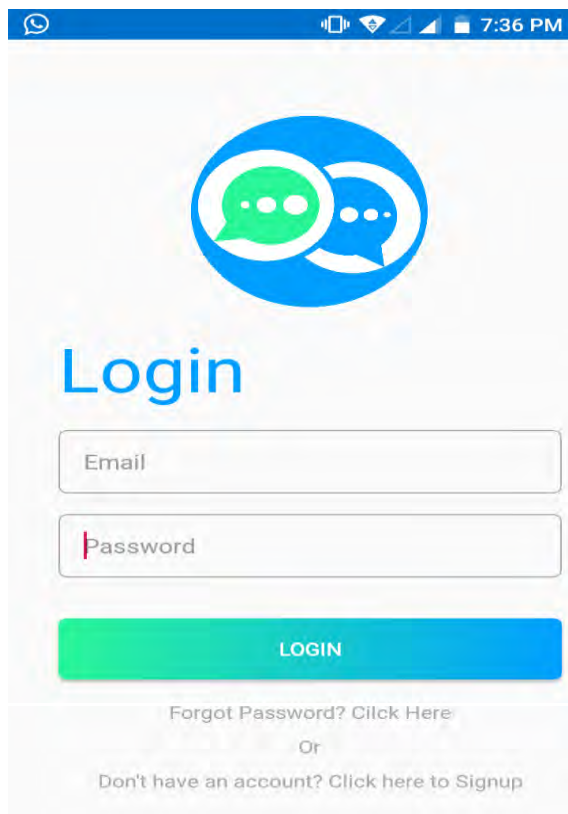


Figure 4.4 Login

4.4.3 Friends List

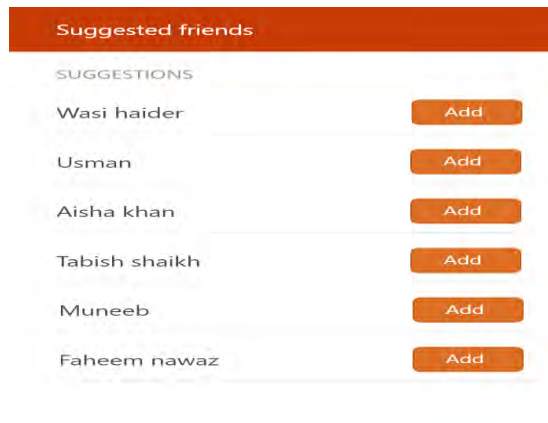


Figure 4.5 Friends List

4.4.4 Suggested Friends List

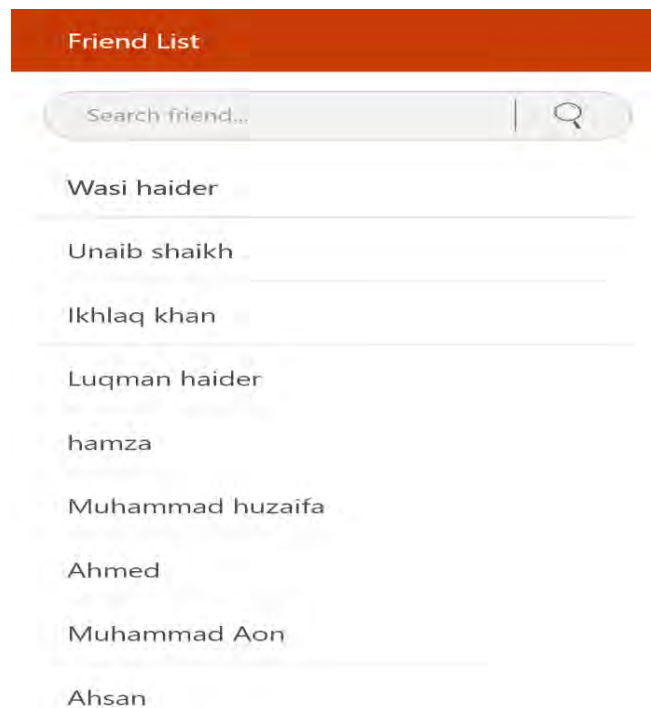


Figure 4.6 Suggested Friends List

4.5 Sequence Diagram

Sequence diagrams are used to present the interaction between actors and objects in a system and interaction between the objects themselves. A sequence diagram shows the interactions that take place during a particular use case or use case scenario.

4.5.1 SD for Use-Case Respond to Request

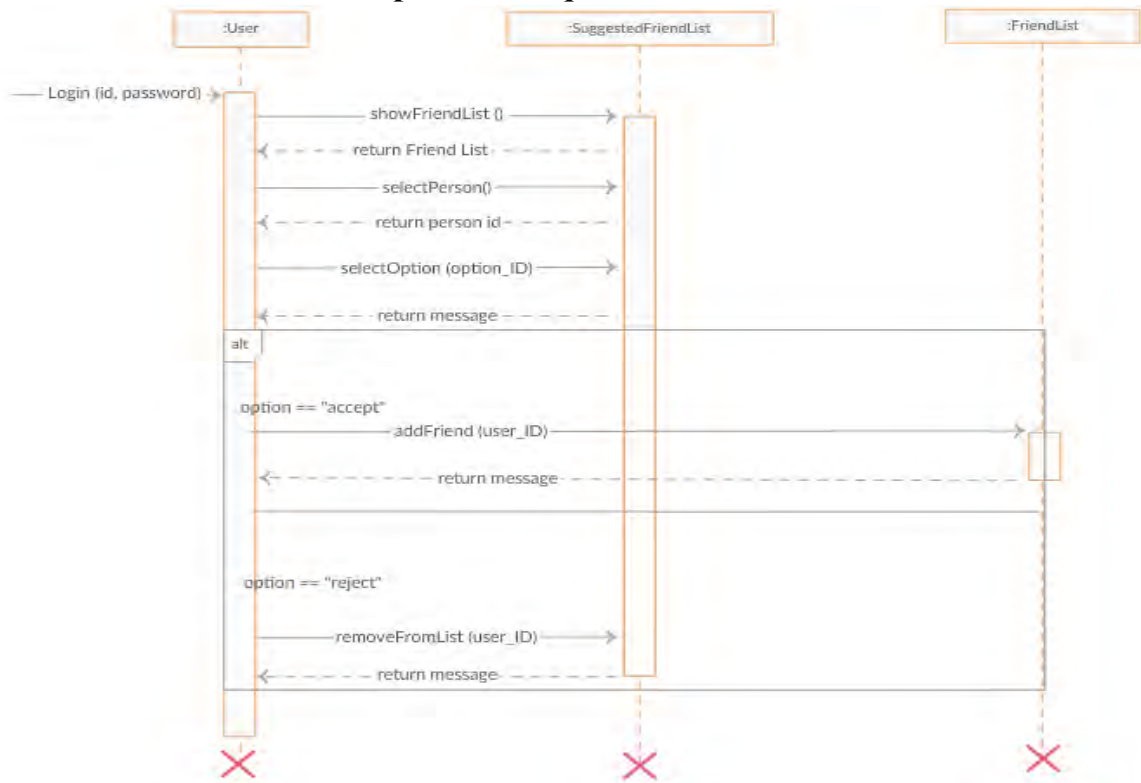


Figure 4.7 SD for Respond to Request

4.5.2 SD for Use-Case Sign Up

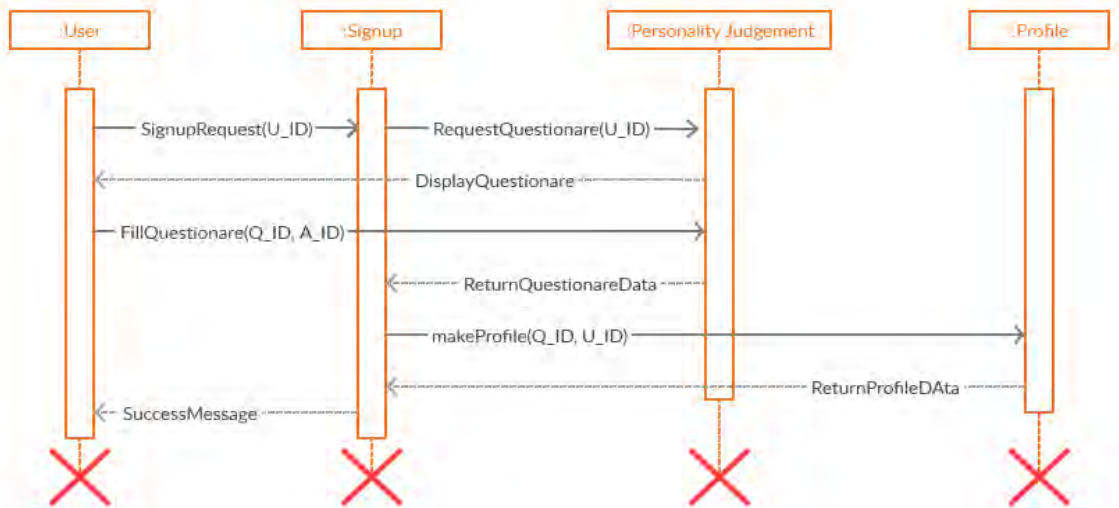


Figure 4.8 SD for Sign Up

4.5.3 SD for Use-Case Start Chat

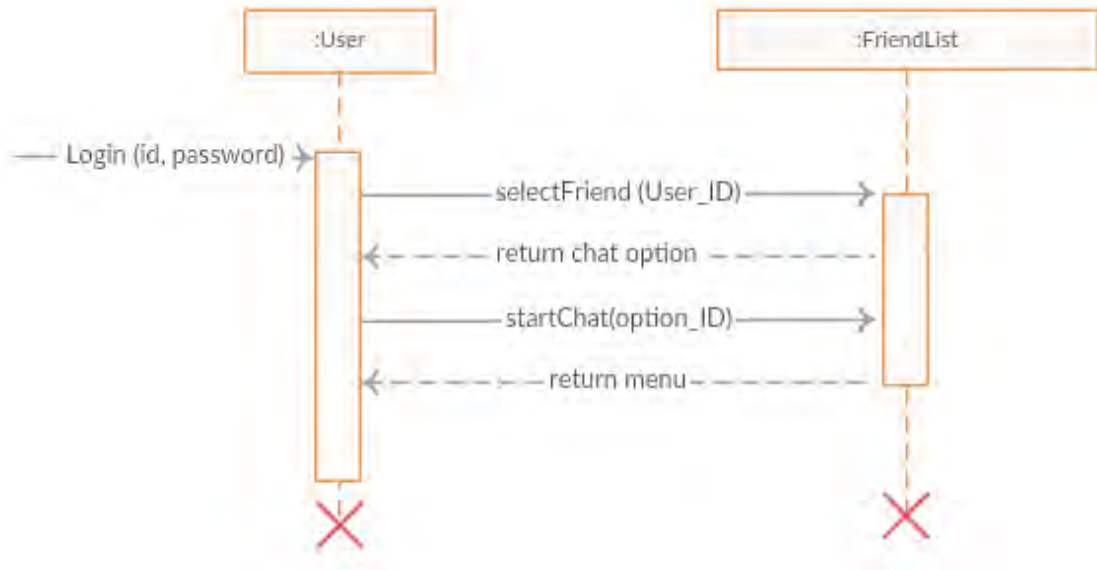


Figure 4.9 SD for Start Chat

4.5.4 SD for Use-Case View Notification

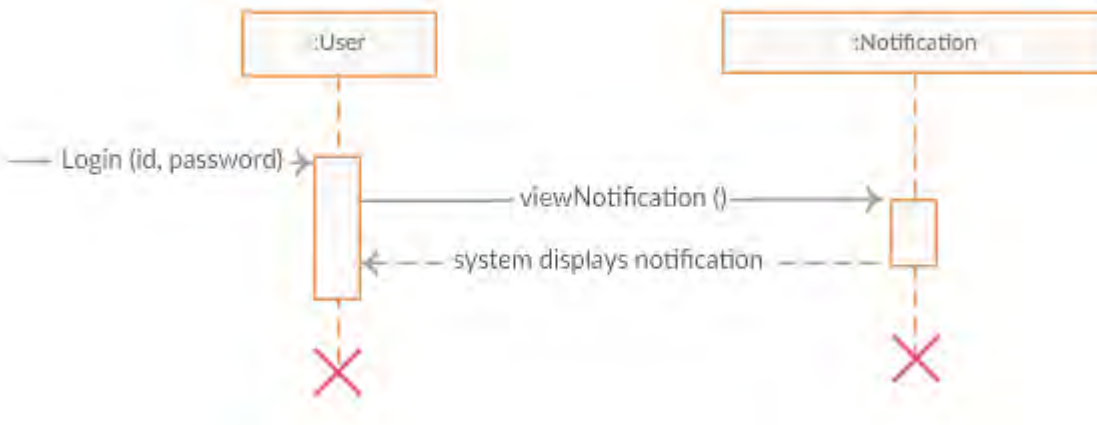


Figure 4.10 SD for View Notification

4.6 Class Diagram

Class Diagram depict the software classes and their relationship. This diagram defines the individual classes along with their attributes, types of the attributes and operations, associations and between classes and navigability that defines attribute visibility and also define non-attribute visibility.

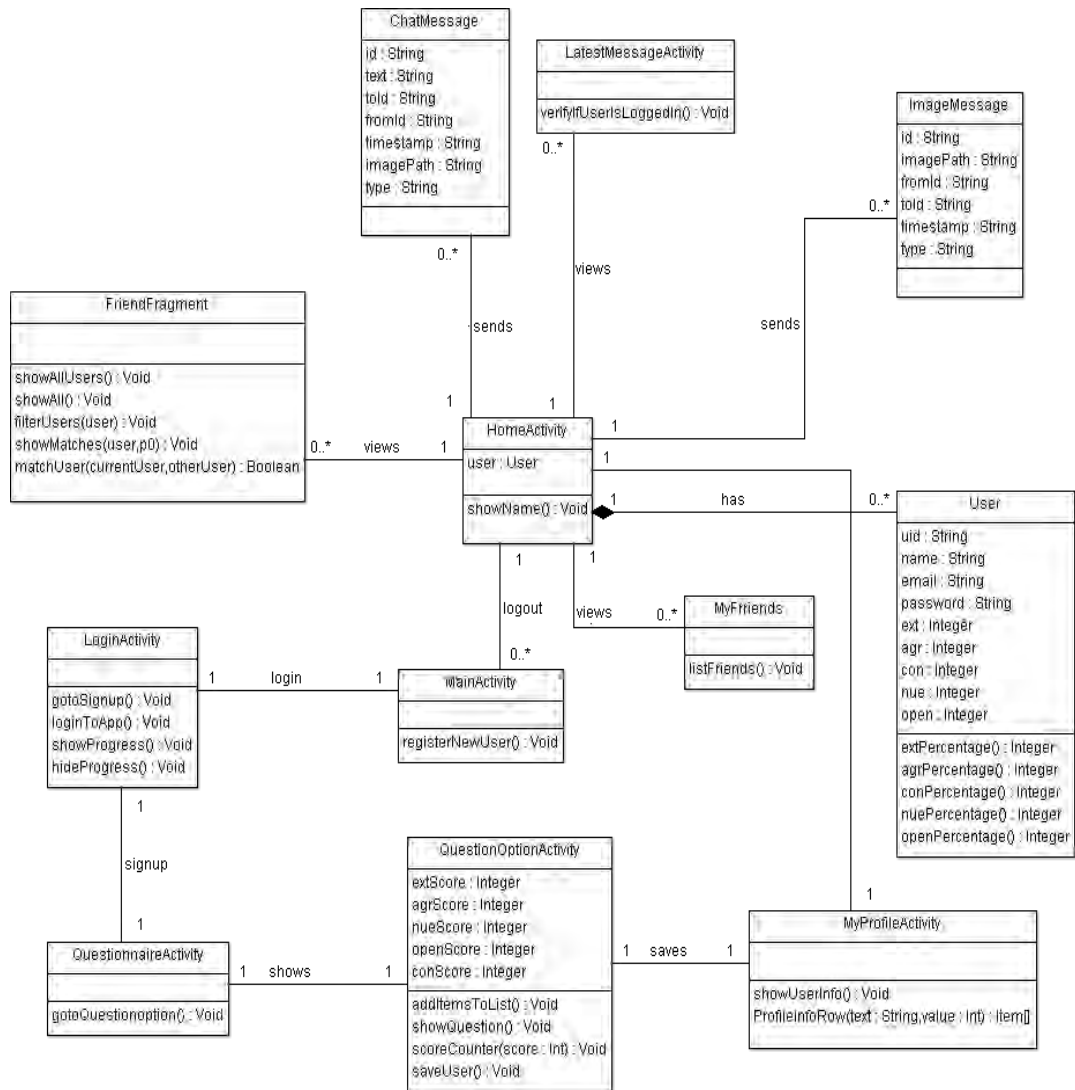


Figure 4.11 Class Diagram

5 Chapter-5: Software Implementation Document

5.1 Introduction

This chapter describes the implementation of the system. In this Chapter, Selected platforms, operating system and language has been described. It includes interfaces of implemented system.

5.2 Selected Platform

As this is android based application so its development requires android studio.

5.3 Selected Operating System

As this is an android based social application so it will run on only android operating system. It can also enhance to iOS operating system in future.

5.4 Selected Language

KOTLIN is the language used to implement the application because it is mature language officially adopted by Google and it makes development much easier.

5.5 Interfaces of Implemented System

Final interfaces of implemented system are given below. These will provide idea about the system.

5.5.1 Login Interface

Login interface for user. User enters login credentials to access system's functionality.

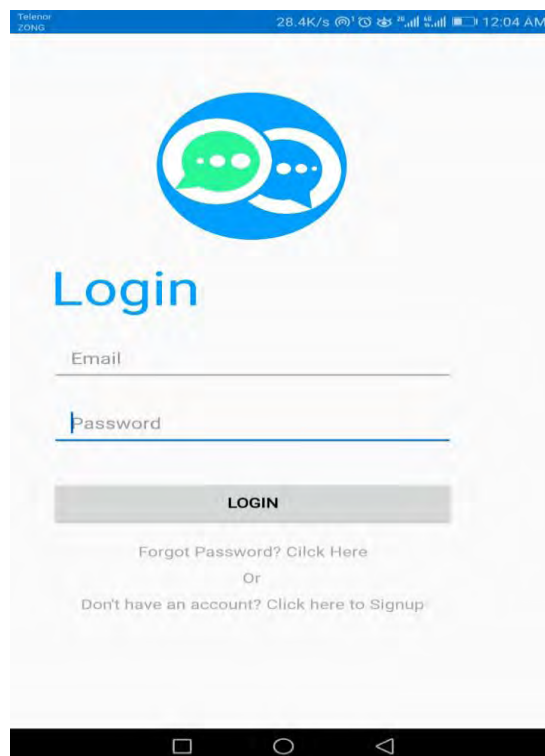
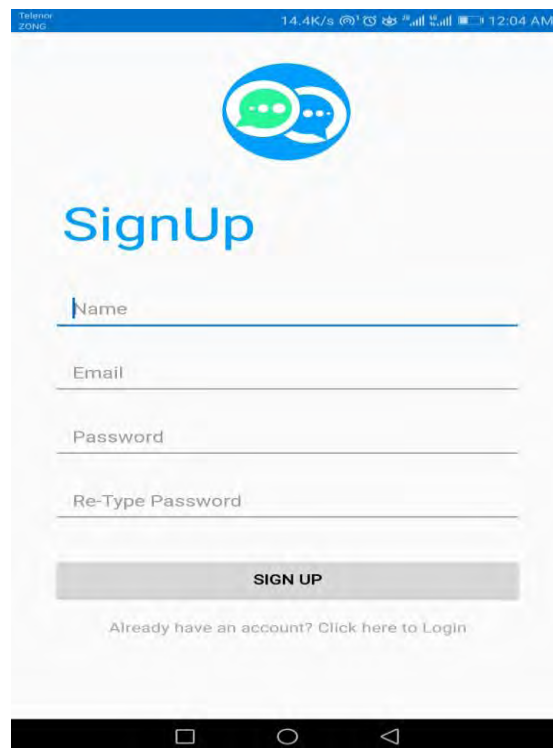


Figure 5.1 Login Interface

5.6 Signup Interface

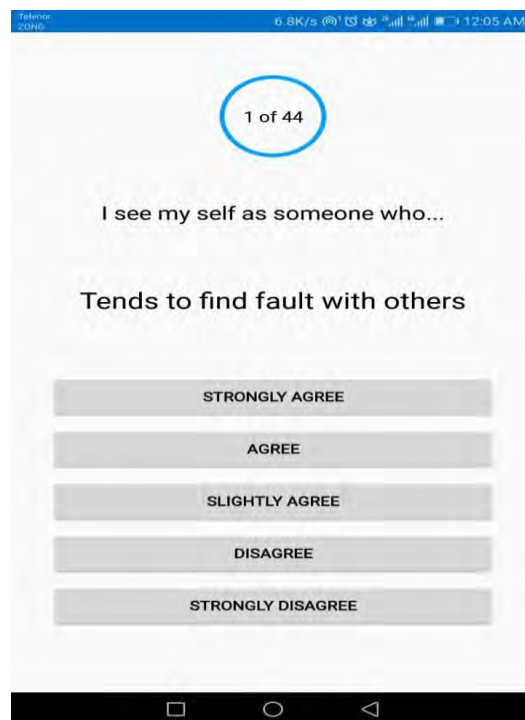
Signup interface is given below. User will first enter his personal information then he is asked to fill the questionnaire.



The screenshot shows a mobile application interface for signing up. At the top, there is a status bar with the text 'Telcel 2016' and '14.4K/s'. Below the status bar is a blue header with a white speech bubble icon containing two green dots. The main heading is 'SignUp' in a large, blue, sans-serif font. Below the heading are four input fields: 'Name', 'Email', 'Password', and 'Re-Type Password'. Each field has a thin blue underline. Below the input fields is a grey button with the text 'SIGN UP' in black. At the bottom of the form, there is a link that says 'Already have an account? Click here to Login'. The bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

Figure 5.2 Signup Interface

(a)



The screenshot shows a mobile application interface for a questionnaire. At the top, there is a status bar with the text 'Telcel 2016' and '6.8K/s'. Below the status bar is a blue header with a white speech bubble icon containing two green dots. The main heading is '1 of 44' in a white font, centered within a blue circle. Below the heading is the text 'I see my self as someone who...' followed by the statement 'Tends to find fault with others'. Below the statement are five horizontal grey buttons with the following text: 'STRONGLY AGREE', 'AGREE', 'SLIGHTLY AGREE', 'DISAGREE', and 'STRONGLY DISAGREE'. The bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

Figure 5.3 Signup Interface (Questionnaire)

5.7 View Profile Interface

User can view his personality traits score in his profile interface given below.

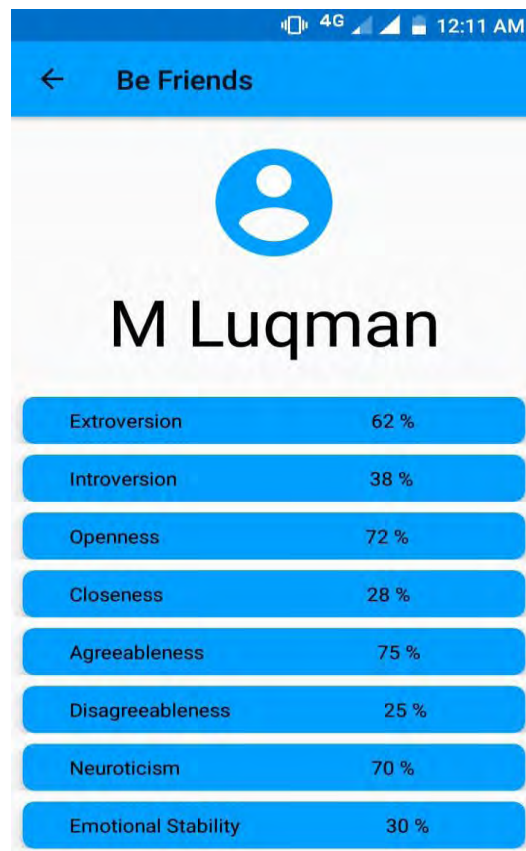


Figure 5.5 View Profile Interface

5.8 View Friend Suggestions

User can see suggested friend list by system according to their personality traits.

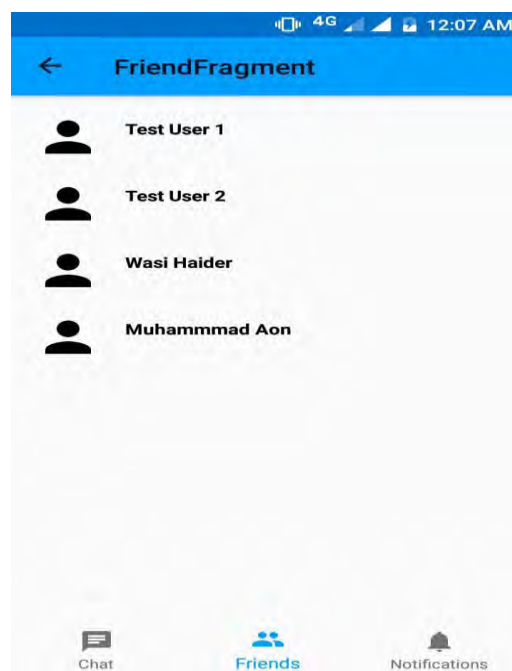


Figure 5.6 Suggested Friend List Interface

5.9 Chat Interface

User starts a text chat with his friend added to his friend list.

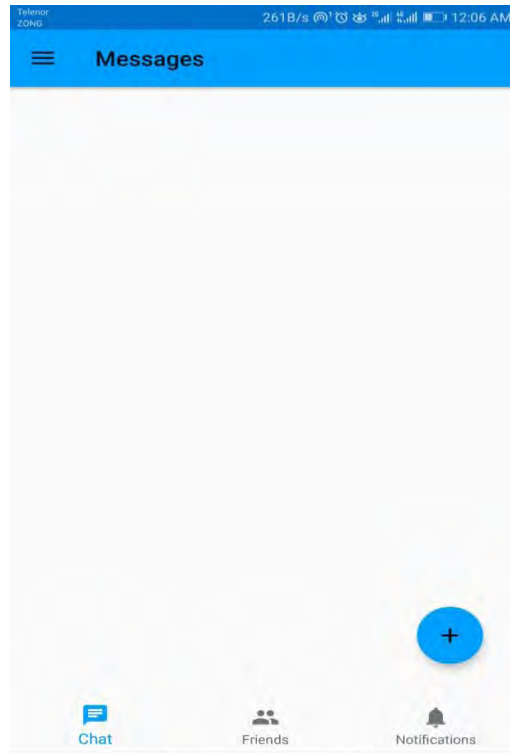


Figure 5.7 Chat Interface

6 Chapter-6: Software Test Document

6.1 Introduction

This chapter illustrates the test approach, which I used in this project, testing tools and environments and the test cases. Testing is the process of evaluating the system or its components with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual requirements according to the ANSI/IEEE 1059 Standard, Testing can be defined as – A process of analyzing the software item to detect the differences between the existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

6.1.1 Test Approach

This section describes the strategy used for the testing of the project. For this project black box testing technique will be used. Acceptance test will be performed for this regard.

6.2 Test Plan

This section describes test strategy, objectives, resources required for testing, test schedule and test deliverables. The test plan serves as a blueprint to conduct software testing activities. It also describes which features to be tested and which are not to be tested.

6.2.1 Features To Be Tested

- Login
- View Notifications
- View Suggested Friends
- Remove Friends from List
- View Friend

6.2.2 Features Not To Be Tested

Processing memory and Speed is not tested here.

6.2.3 Testing Tools And Environments

- Mobile Phone
- Android Operating System

6.3 Test Cases

6.3.1 TC-1 Login

Table 6.1 TC-1 Login

ID	TC-1
Description	Registered user can log in to the application
Tester	User
Setup	Register user with ID “luqman123” and Password “12345678”.
Input	<ol style="list-style-type: none">1. Go to login page of application2. Enter username “luqman123”3. Enter password “12345678”4. Press Login button
Expected Output	User should be successfully logged into the application

6.3.2 TC-2 View Notification

Table 6.2 TC-2 View Notification

ID	TC-2
Description	User can see notifications of friend requests and messages.
Tester	User
Setup	Register user with username user1 and another user with user2 and choose same options for both so they come in suggested list for each other.
Input	<ol style="list-style-type: none">1. Login with user1 credential2. Open Suggested List3. Send a friend Request to user24. Logout for user15. Login with user2 credential6. Open notification by pressing notification icon
Expected Output	User should see a friend request of user1

6.3.3 TC-3 View Suggested Friend

Table 6.3 TC-3 View Suggested Friends

ID	TC-3
Description	User can see suggested friend based on his filled questionnaire.
Tester	User
Setup	Register a user with ID “luqman123” and fill questionnaire and other options for him.
Input	<ol style="list-style-type: none">1. Go to signup page2. Enter username “ali234”3. Enter password “234ui”4. Fill questionnaire with the same answers as for luqman123
Expected Output	User ali234 should see luqman123 in his suggested friend list

6.3.4 TC-4 View Friend List

Table 6.4 TC-4 View Friend List

ID	TC-4
Description	User can remove any friend from his friend list
Tester	User
Setup	Register a user with ID “luqman123” and fill questionnaire and other options for him. Register another user with ID “ali234” and password “234ui” and same options as for luqman123.
Input	<ol style="list-style-type: none">1. Login with luqman123 credential2. Go to suggested friend list3. Send a request to ali2344. Logout from user luqman1235. Login with ali234 credential6. Go to notifications and accept request from luqman1237. Go to Friend list
Expected Output	User ali234 should see luqman123 in the friend list

6.3.5 TC-5 Remove Friends from list

Table 6.5 TC-5 Remove Friend from List

ID	TC-5
Description	User can unfriend any user from his friend list
Tester	User
Setup	Register a user with ID “luqman123” and fill questionnaire and other options for him and another user with ID “ali234” and password “234ui” with same options as for luqman123. Make both user friends.
Input	<ol style="list-style-type: none">1. Login with luqman123’s credentials2. Selects to Friend list3. Select remove friend option for ali234
Expected Output	Both users should not see each other in the friend list

7 Chapter-7: Conclusion and Future Enhancements

7.1 Conclusion

The whole project's development took a lot of efforts and challenges.

Because KOTLIN was a new language for me and there was no mature developers' community of KOTLIN to help out. Finally, the project has been completed within the given period of time. The end product is very user friendly and provides a very interactive feature to the users. Secondly, I had to research and study to understand the human's personality.

Application used Friend Selection Model to suggest friends to the user. The user can send friend requests to suggested friends. Now, user can chat with his friends.

7.2 Future Enhancement

Some improvements can be made in this application in the future enhancement. The application can be enhanced to a much more user friendly.

7.2.1 iOS Version

As this application is only for android operating system. In future it will be available for iOS users also.

7.2.2 Connecting with other Social Applications

In Future, Users will be able to share their memories in the form of pictures, videos and in any form of multimedia. Users will be able to link their accounts with other social media accounts for sharing purpose.

7.2.3 Video and Audio Communication

Currently users can communicate with their friends via text chat only but in future they will be able to make audio and video calls to their friends. They will be also able to exchange any kind of multimedia attachment (audio, image, documents etc.) with their friends.

REFERENCES

- [1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, June 25, 1998.
- [2] Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative development, Third Edition by Craig Larman.
- [3] John, O. P., & Srivastava, S. (1999). The Big-Five trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research* (Vol. 2, pp. 102–138). New York: Guilford Press.
- [4] Maarten Selfhout, William Burk, Susan Branje, Jaap Denissen, Marcel van Aken, and Wim Meeus, (2010), Emerging Late Adolescent Friendship Networks and Big Five Personality Traits: A Social Network Approach, Journal compilation r 2010, Wiley Periodicals, Inc.
DOI: 10.1111/j.1467-6494.2010.00625.x
- [5] Wilson, (2015). *Personality and Friendship Satisfaction in Daily Life*, European Journal of Personality, Eur. J. Pers. (2015)