

# Software Requirements Specification

**Prepared by BlueQuoters  
for the project QuoteShot**

*METU - Department of Computer Engineering  
CENG 491 Senior Design Project I  
Fall 2015-2016*



	<b>Name</b>	<b>ID</b>
Prepared by	Zeynep Havva Dinç	1819283
Prepared by	Burak Aydemir	1745777
Prepared by	Mercan Boz	1941848
Prepared by	Zeliha Yılmaz	1884626

## TABLE OF CONTENTS

<b>1. Introduction</b>	3
1.1 Problem Definition	3
1.2 System Overview	3
1.3 Definitions, acronyms, and abbreviations	4
1.4 Assumptions and dependencies	4
<b>2. Overall description</b>	5
2.1 Product functions	5
2.1.1 Use-case model survey	5
2.1.2 Actor survey	19
2.2 Interfaces	20
2.2.1 User Interfaces	20
2.2.2 Hardware Interfaces	23
2.2.3 Software Interfaces	23
2.2.4 Communications Interfaces	23
2.3 Constraints	23
<b>3. Specific requirements</b>	23
3.1 Functional Requirements	24
3.2 Non-functional Requirements	24
3.2.1 Usability	24
3.2.2 Reliability	24
3.2.3 Performance	25
3.2.4 Supportability	25
<b>4 Data Model and Description</b>	25
4.1 Data Description	25
4.1.1 Data objects	27
4.1.2 Data dictionary	28
<b>5 References</b>	33

## FIGURES

Figure 1: Block Diagram	3
Figure 2: Use Case Diagram	6
Figure 3: Login & Register Design 1	21
Figure 4: Login & Register Design 2	22
Figure 5: Profile Design	23
Figure 6: ER Diagram	27
Figure 7: Class Diagram	28
Figure 8: User Class	29
Figure 9: User Interface	30
Figure 10: Quote Class	31
Figure 11: Book Class	32

## TABLES

Table 1: Acronyms and Abbreviations	4
Table 2: Login to QuoteShot	7
Table 3 : Register to QuoteShot	8
Table 4 : Follow Another User	9
Table 5 : Unfollow Another User	10
Table 6 : Edit Profile	10
Table 7 : Commenting Quote	11
Table 8 : Liking Quote	12
Table 9 : Unliking Quote	12
Table 10 : Sharing Quote	14
Table 11 : Blocking a User	15
Table 12 : Sending Message	15
Table 13 : Report a User	16
Table 14 : Book Exchange Request	17
Table 15 : Book Exchange Offer	18
Table 16 : Settings	18
Table 17 : Logout	19
Table 18 : User Class	29
Table 19 : User Interface	30
Table 20 : Quote Class	32
Table 21 : Book Class	33

# 1. Introduction

The purpose of this document is to specify the system requirements and the overall description of the concept of this project. It contains the constraints of the system, the system functionality using diagrams. It also contains the user and system interfaces. In addition, Logical database design is specified by ER and Class diagrams. This document is an initial reference for developing the first version of the system for development team.

## 1.1 Problem Definition

The purpose of the project “QuoteShot” is to provide a social platform project where the users can share their favourite snapshots of the quotes from the books they read, in their feeds or other social media platforms such as Facebook. They shall also be able to share the quotes in text form generated by the application's text recognition. Trending lists of books of different categories will be generated periodically, based on the sharings of users. On top of all the features, users who are locally close enough, shall be able to exchange books.

## 1.2 System Overview

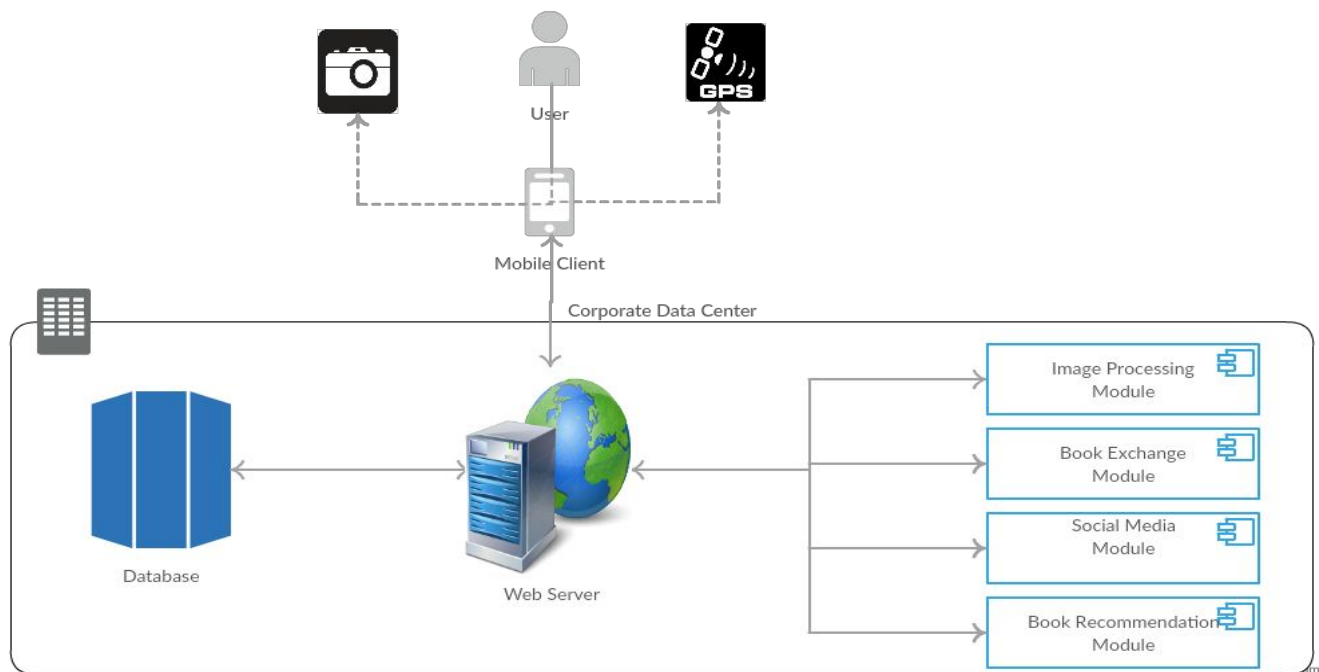


Figure 1: Block Diagram

In QuoteShot, there 4 modules related to web server. The first one is the image processing module. When the user takes photo from his smartphone's camera, the user can crop the image by using image processing module. Then the module makes blur effect on the photo and converts to text form. The second module is the book exchange module. The user can exchange book according to his location by using his smartphone's camera. The third module is social media module. By using this module user can share, like and comment quotes, send/get messages from other users and edit his profile. The last module is the book

recommendation module. This module recommends books and quotes according to quotes which the user shared and liked by using NLP algorithms. Web Server will behave as a bridge between the client software and the database system for the information transactions and do the relevant manipulations such as image processing. The DBMS will store user data, generic book information and quotes, and relations.

### 1.3 Definitions, acronyms, and abbreviations

Abbreviation/Acronym	Definition
IEEE	Institute of Electrical and Electronics Engineers
DBMS	A software package/system to facilitate the creation and maintenance of a computerized database.
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
QS	QuoteShot
UC	Use-case
OS	Operating system
GPS	Global Positioning System. A global system of U.S. navigational satellites developed to provide precise positional and velocity data and global time synchronization for air, sea, and land travel.

*Table 1: Acronyms and Abbreviations*

### 1.4 Assumptions and dependencies

For a fully functioning QuoteShot application, the following criterias are needed:

- The user should have smartphone.
- Smartphone's OS should be Android.
- Smartphone's camera should work properly.
- Smartphone's GPS should work properly.
- Smartphone should have internet connection.

## 2. Overall description

### 2.1 Product functions

#### 2.1.1 Use-case model survey

In this section the main functions of the QuoteShot application are explained with more details by using the use case descriptions.

#### Use-case Names

1. **Login:** User logs in to the system.
2. **Registration:** A non-member user registers to the system.
3. **Follow:** A member user follows another member to get posts about them.
4. **Unfollow:** A member user unfollows another member user followed before.
5. **Edit Profile:** A member user edits their profile information.
6. **Comment:** A member user comments on a post in the system.
7. **Like:** A member user likes a post in the system.
8. **Unlike:** A member user unlikes a post in the system.
9. **Share a Quote:** A member user shares any quote they have visibility to.
10. **Block a User:** A user blocks another user to stop communication with them.
11. **Send Message:** A member user sends a message(text or media) to another member.
12. **Report:** A user reports another user in case of an abuse or any breach of rule.
13. **Settings:** A user changes settings of their account or general settings of application.
14. **Book Exchange:** A user either offers their book for an exchange or searches a book.
15. **Logout:** A user logs out from the system.

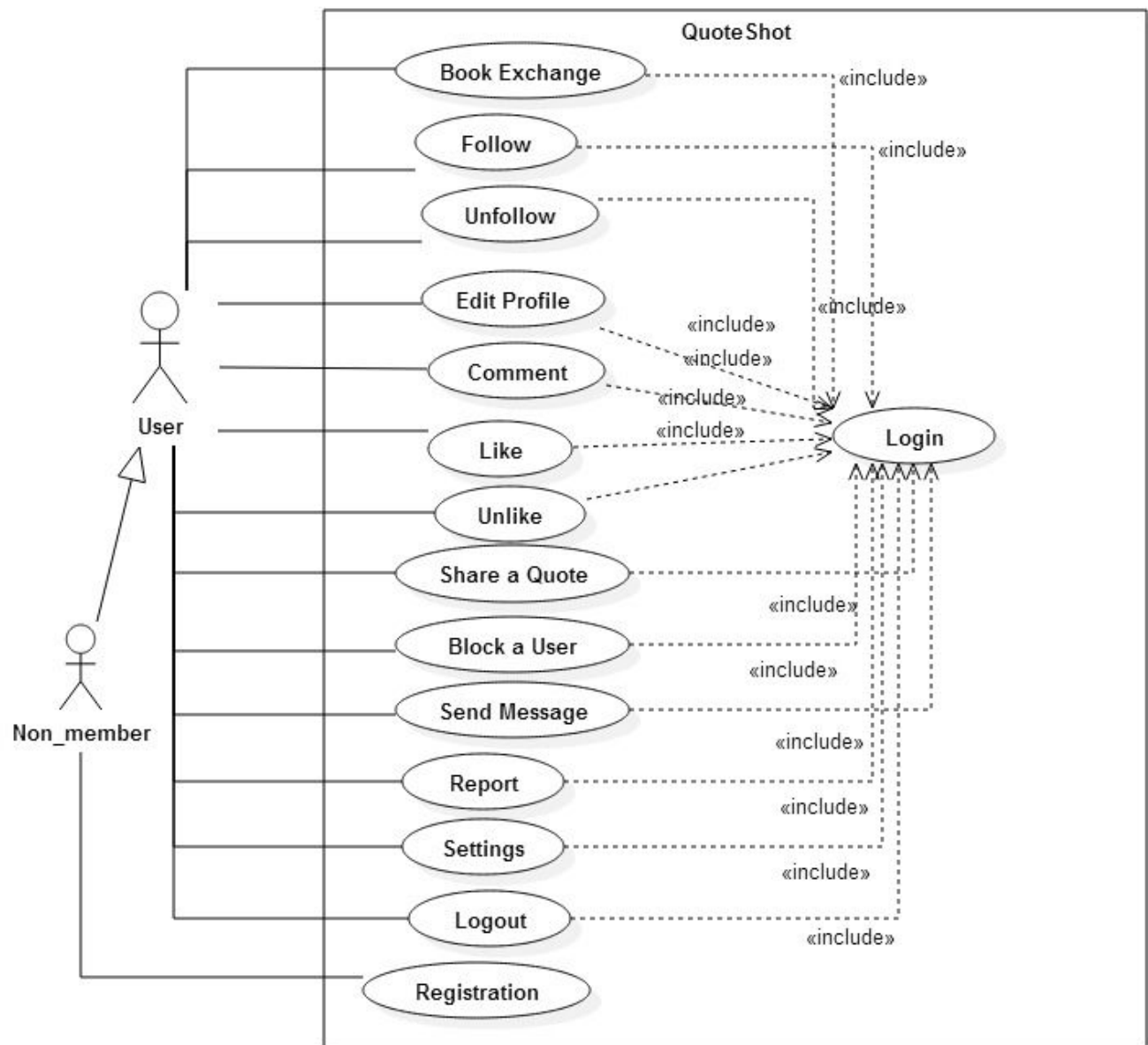


Figure 2: Use Case Diagram

### 2.1.1.1 Login to QuoteShot

Use Case ID	The QS UC.1
Use Case Name	Login
Description	User who completes the registration process, can login to QuoteShot application.
Actors	User
Preconditions	User must have already registered.
Trigger	This use case is activated when user enters username and password and clicks Login button.
Basic Flow	<ol style="list-style-type: none"><li>1. User clicks on “Login” section.</li><li>2. User enters his credentials in the area provided for that.</li><li>3. System checks if the entered login parameters are valid.</li><li>4. System creates a new session for the user.</li></ol>
Alternate Flow	<ul style="list-style-type: none"><li>-User can login via Facebook account.</li><li>-User can login via Gmail account.</li></ul>
Exception Flow	<ul style="list-style-type: none"><li>- If the user provides invalid login information, the system notifies the user and redirects him to login section.</li></ul>
Post Condition	User logs in to the system successfully.

*Table 2: Login to QuoteShot*

#### 2.1.1.1.1 : Req 001

The system shall control password and username properly.

#### 2.1.1.1.2 : Req 002

The system shall control password and username in a secure way.

#### 2.1.1.1.3 : Req 003

The system shall allow user to login by using his facebook & gmail account.

#### 2.1.1.2.4 : Req 004

Whether the user logs in to the system successfully or not the system shall inform the user by showing message.

### 2.1.1.2 Register to QuoteShot

Use Case ID	The QS UC.2
Use Case Name	Registration
Description	User must register to the system in order to be user of QuoteShot.
Actors	Non_member
Preconditions	User is not a member of QuoteShot
Trigger	User clicks the “Register” button.
Basic Flow	<ol style="list-style-type: none"><li>1. User press “Register” button on QuoteShot screen.</li><li>2. User enters his credentials and submits his information for registration.</li><li>3. System checks if the provided information is valid.</li><li>4. A new account is created for the user.</li><li>5. User creates a password for his account.</li><li>6. The user information is kept in database.</li></ol>
Alternate Flow	A user can also register to QuoteShot using Gmail account and the credentials of user is taken from his Gmail account.
Exception Flow	2a. If the user enters invalid or incomplete information, the system notifies the user and returns to the registration screen.
Post Condition	The user becomes a member of QuoteShot application.

*Table 3 : Register to QuoteShot*

#### 2.1.1.2.1 : Req 005

The system shall check the compulsory area whether the user enter or not.

#### 2.1.1.2.2 : Req 006

The system shall check whether provided information is valid or not.

#### 2.1.1.2.3 : Req 007

If the user wants to register by using his gmail account the system shall take the information such as name, email from his account.

#### 2.1.1.2.4 : Req 008

Whether the user register successfully or not the system shall inform the user by showing message.

#### 2.1.1.2.5 : Req 009

The system shall create allocate space in the database for user's information.

### 2.1.1.3 Follow a User

Use Case ID	The QS UC.3
Use Case Name	Follow
Description	User can follow any of his friends to see their posts.
Actors	User
Preconditions	Users can follow the people who accept the follow request.
Trigger	User clicks the Follow button.
Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on “Follow” button and send follow request.</li> <li>2. According to request response user can follow another user.</li> <li>3. User can see the posts of the user.</li> </ol>
Alternate Flow	-
Exception Flow	<ul style="list-style-type: none"> <li>- If the person who was sent follow-up request not accept the request then a user can not follow this user.</li> <li>- If the person who was sent follow-up request had blocked the user, the system does not allow to follow.</li> </ul>
Post Condition	Any post which is shared by friends who is followed by the user can be seen in user’s main page.

*Table 4 : Follow Another User*

#### **2.1.1.3.1 : Req 010**

If the user follows another user the system shall show the posts from followed user.

#### **2.1.1.3.2 : Req 011**

If the user wants to send follow request another user the system shall check whether the user was blocked or not.

#### **2.1.1.3.3 : Req 012**

If the user was blocked the system shall give error message.

#### **2.1.1.3.4 : Req 013**

If the user was sent follow-up request does not accept the request, the system shall not allow the user who wants to follow to see his posts .

#### 2.1.1.4 Unfollow a User

Use Case ID	The QS UC.4
Use Case Name	Unfollow
Description	User can give up follow his friends.
Actors	User
Preconditions	User must have already follow another user.
Trigger	Users click the Unfollow button.
Basic Flow	1. User clicks on “Unfollow” button. 2. User can not see the posts of unfollowed user anymore on main page.
Alternate Flow	1. User clicks on the “report” button on the profile of the user to be unfollowed and automatically unfollows them.
Exception Flow	-
Post Condition	After unfollowing any friends, user will not get any post from them.

*Table 5 : Unfollow Another User*

##### 2.1.1.4.1 : Req 014

The system shall not show posts shared by the user unfollowed.

#### 2.1.1.5 Edit Profile

Use Case ID	The QS UC.5
Use Case Name	Edit Profile
Description	User can edit his profile by changing profile picture, email address, username or location information.
Actors	User
Preconditions	User must have already logged in.
Trigger	User clicks the Edit Profile button.

Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on “Edit Profile” button.</li> <li>2. User redirected Edit Profile page which includes all information about the user.</li> <li>3. User chooses the any part that he wants to change.</li> <li>4. After user finishes his edit process, he clicks “Save” button.</li> </ol>
Alternate Flow	-If a user gives up to edit profile, he can cancel this process by clicking “Cancel” button.
Exception Flow	- If a user changes their credentials with invalid information, the application gives warning and redirects them to the edit profile page.
Post Condition	User profile is updated.

*Table 6 : Edit Profile*

#### **2.1.1.5.1 : Req 015**

The user can change his profile photo.

#### **2.1.1.5.2 : Req 016**

The user can change his name, surname and email address.

#### **2.1.1.5.3 : Req 017**

The user can change the lists of book which he has.

#### **2.1.1.5.4 : Req 018**

The user can change the lists of book which he wants to exchange.

#### **2.1.1.5.5 : Req 019**

When the user clicks save, the system shall update the database.

#### **2.1.1.6 Commenting Quote**

Use Case ID	The QS UC.6
Use Case Name	Comment
Description	User can comment another user’s posts.
Actors	User
Preconditions	User must have already logged in.
Trigger	Users click the Comment icon.
Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on “Comment” icon.</li> <li>2. User writes his comment.</li> <li>3. User clicks Send button.</li> </ol>

Alternate Flow	-
Exception Flow	-
Post Condition	User leaves his comment successfully.

*Table 7 : Commenting Quote*

#### **2.1.1.6.1 : Req 020**

If the user comments any quote the system shall update database.

#### **2.1.1.7 Liking Quote**

Use Case ID	The QS UC.7
Use Case Name	Like
Description	User can like another user's posts.
Actors	User
Preconditions	User must have already logged in.
Trigger	Users click the Like icon.
Basic Flow	1. User clicks on "Like" icon.
Alternate Flow	-
Exception Flow	-
Post Condition	User likes a post successfully.

*Table 8 : Liking Quote*

#### **2.1.1.7.1 : Req 021**

If the user likes any quote the system shall update database.

#### **2.1.1.8 UnLiking Quote**

Use Case ID	The QS UC.8
Use Case Name	Unlike
Description	User can unlike the posts

Actors	User
Preconditions	User must have already login and liked the post.
Trigger	Users click the Unlike icon.
Basic Flow	1. User clicks on “Unlike” icon.
Alternate Flow	-
Exception Flow	-
Post Condition	User unlikes a post.

*Table 9 : Unliking Quote*

#### **2.1.1.8.1 : Req 021**

If the user unlikes any quote the system shall update database.

#### **2.1.1.9 Sharing Quote**

Use Case ID	The QS UC.9
Use Case Name	Share a quote
Description	User can share a quote.
Actors	User
Preconditions	User must have already logged in.
Trigger	Users click the Camera icon.
Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on Camera icon.</li> <li>2.The application opens the camera.</li> <li>3.User takes photo from his book.</li> <li>4.User specify the lines that he wants to share.</li> <li>5.User choose a book name and author name.</li> <li>6.The application directs the user to the sharing page.</li> <li>7.User choose the one of the platform that he wants to share his quote.</li> <li>8.If the user choose main page option, his quote is appeared on his main page and profile.</li> </ol>
Alternate Flow	-If user wants to share his quote in text form on Facebook or Whatsapp the application converts his image form of the quote

	into text form using OCR. Then user can share his quote. -If the user wants to share his quote in image form on Facebook, Instagram or Whatsapp, the application shares as image.
Exception Flow	-
Post Condition	User shares his quote.

*Table 10 : Sharing Quote*

#### **2.1.1.9.1 : Req 022**

When the user wants to share quote the system shall open the camera.

#### **2.1.1.9.2 : Req 023**

When the user takes photo the system shall allow the user to crop the image.

#### **2.1.1.9.3 : Req 024**

If the user wants text form the system shall convert text form properly.

#### **2.1.1.10 Blocking a User**

Use Case ID	The QS UC.10
Use Case Name	Block a user
Description	User can block any other users.
Actors	User
Preconditions	User must have already login.
Trigger	Users click the another user profile.
Basic Flow	1.User clicks another user profile. 2. User clicks block button. 3.The application shows warning dialogue which asks "Are you sure to block this user". 4.User click YES. 5.The application shows successful message.
Alternate Flow	-
Exception Flow	If the user blocked another user before, s/he can not click the block again.
Post Condition	User can not follow blocked user, can not send and take messages blocked user and

	can not see the shared posts of blocked user.
--	---

*Table 11 : Blocking a User*

**2.1.1.10.1 : Req 025**

If the user blocks a user the system shall show message.

**2.1.1.10.2 : Req 026**

If the user blocks another user the system shall not allow connection of two users.

**2.1.1.11 Sending Message**

Use Case ID	The QS UC.11
Use Case Name	Message
Description	User can send message any member of the QuoteShot.
Actors	User
Preconditions	User must have already logged in.
Trigger	User enters another user profile that he wants to send message.
Basic Flow	1. User enters another user profile that he wants to send message. 2. User clicks "Send message" button. 3. User writes his message. 4. User click send button.
Alternate Flow	-
Exception Flow	- If a user who is intended to send message, closed his messaging process from privacy settings, a user who wants to send message, the user can not send message anymore.
Post Condition	User sends message.

*Table 12 : Sending Message*

**2.1.1.11.1 : Req 027**

The system shall not allow the user to send a message to user who blocked him.

**2.1.1.11.2 : Req 028**

The system shall show error message when the user wants to send a message to user who blocked him.

**2.1.1.11.3 : Req 029**

When the user sends message the system shall update database.

#### 2.1.1.12 Report a User

Use Case ID	The QS UC.12
Use Case Name	Report
Description	User can report any member of the QuoteShot.
Actors	User
Preconditions	User must have already logged in.
Trigger	User enters another user profile that he wants to report.
Basic Flow	1. User enters another user profile that he wants to report. 2.User clicks “Report” button. 3.The application opens a report page. 4.User writes his reasons why he wants to report that user. 5.User clicks send button.
Alternate Flow	-
Exception Flow	- If the reasons of report are not accepted from the admins, the user can not report.
Post Condition	User reports another user successfully.

*Table 13 : Report a User*

##### 2.1.1.12.1 : Req 030

When the user reports another user, the system shall show a successful message.

#### 2.1.1.13 Book Exchange Request

Use Case ID	The QS UC.13
Use Case Name	Book Exchange Request
Description	User can send a book exchange request to another user in a limited range of local area.
Actors	User
Preconditions	User must have already logged in.

Trigger	User clicks Search icon to find books that he wants to exchange with.
Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on the “Book Exchange” tab in the application.</li> <li>2. User searches a book to exchange, of a specific name or category.</li> <li>3. User clicks on the search button.</li> <li>4. The application brings a list of the books available for exchange in a specified local area, ordered by the distance to the user.</li> <li>5. User clicks on any offer and may send a message to the user to communicate for an exchange appointment.</li> <li>5. User clicks on “send request” button.</li> </ol>
Alternate Flow	-
Exception Flow	-
Post Condition	User sends a book exchange request successfully.

*Table 14 : Book Exchange Request*

#### **2.1.1.13.1 : Req 031**

The system shall show the books the properly.

#### **2.1.1.13.2 : Req 032**

If the user clicks “send request” button, the system shall send notification.

#### **2.1.1.14 Book Exchange Offer**

Use Case ID	The QS UC.14
Use Case Name	Book Exchange Offer
Description	User can put a book on their exchange list to make it available for exchange offers from other users.
Actors	User
Preconditions	User must have already logged in.
Trigger	User clicks on Offer a Book button in Book Exchange tab to put their book(s) on their exchange list.
Basic Flow	<ol style="list-style-type: none"> <li>1. User clicks on the “Book Exchange” tab in the application.</li> </ol>

	2. User clicks on Offer a Book button. 3. The application brings the list of the owned books the user previously has added their books to. 5. User clicks on any book to be offered and chooses a location for the book. 5. User clicks on “send offer” button.
Alternate Flow	-
Exception Flow	-
Post Condition	User sends a book exchange offer successfully.

*Table 15 : Book Exchange Offer*

#### **2.1.1.14.1 : Req 033**

When the user offers a book the system shall show message.

#### **2.1.1.15 Settings**

Use Case ID	The QS UC.15
Use Case Name	Settings
Description	User can change his settings such as password, privacy and security.
Actors	User
Preconditions	User must have already logged in.
Trigger	User clicks “Settings” button.
Basic Flow	1. User clicks “Settings” button. 2. User choose the item that he wants to change its settings. 3. After user completed his settings he clicks save button.
Alternate Flow	-
Exception Flow	-
Post Condition	User changes his settings successfully.

*Table 16 : Settings*

#### **2.1.1.15.1 : Req 034**

The system shall allow the user to change privacy features .

#### 2.1.1.15.2 : Req 035

When the user clicks save button the system shall show a successful message.

#### 2.1.1.16 Logout

Use Case ID	The QS UC.16
Use Case Name	Logout
Description	User can logout from the application.
Actors	User
Preconditions	User must have already logged in.
Trigger	Users click “Logout” button..
Basic Flow	1.Users click “Logout” button. 2.The application shows warning message which show “Are you sure to logout” 3.User clicks “YES.” 4.The application shows successful message.
Alternate Flow	-
Exception Flow	-
Post Condition	User will be redirected to the login page.

Table 17 : Logout

#### 2.1.1.16.1 : Req 036

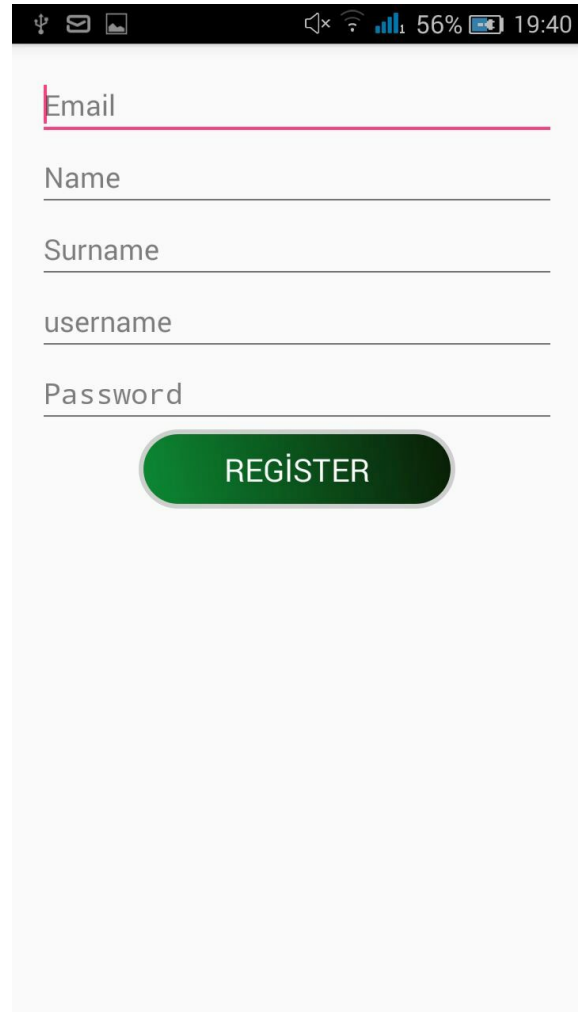
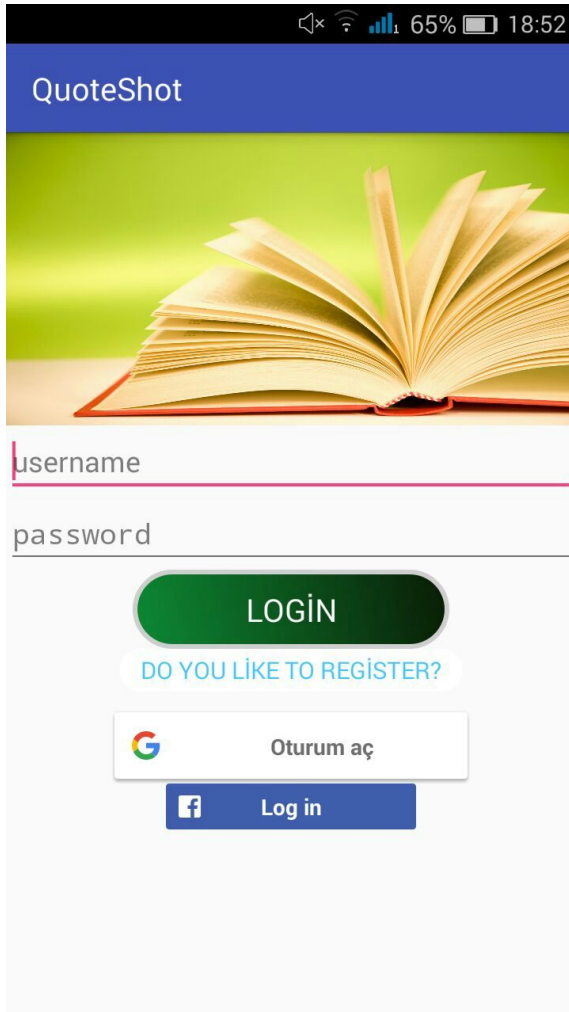
Whether the user logouts to the system successfully or not the system shall inform the user by showing message.

#### 2.1.2 Actor survey

In the QuoteShot application there are two actors which are user and non\_member. To become user a person should have a smartphone and be literate. Also, smartphone’s operating system should be android. As shown in the use case diagram a QuoteShot user can follow or unfollow another user, edit profile, comment, like and unlike a post, share a quote, block a user, send message, report, send book exchange request and logout after login process. A non member also has this options after completes registration process.

## 2.2 Interfaces

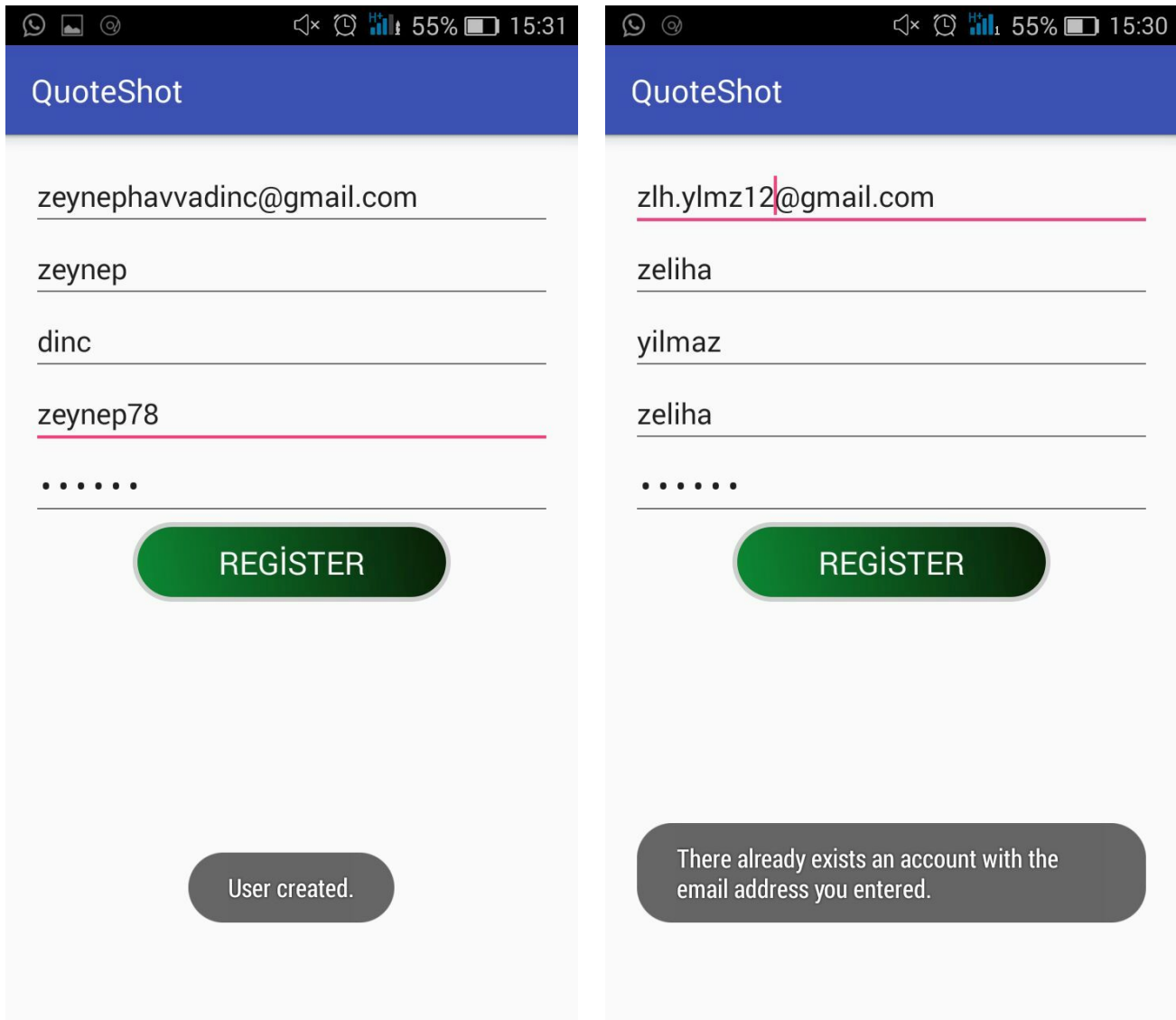
### 2.2.1 User Interfaces



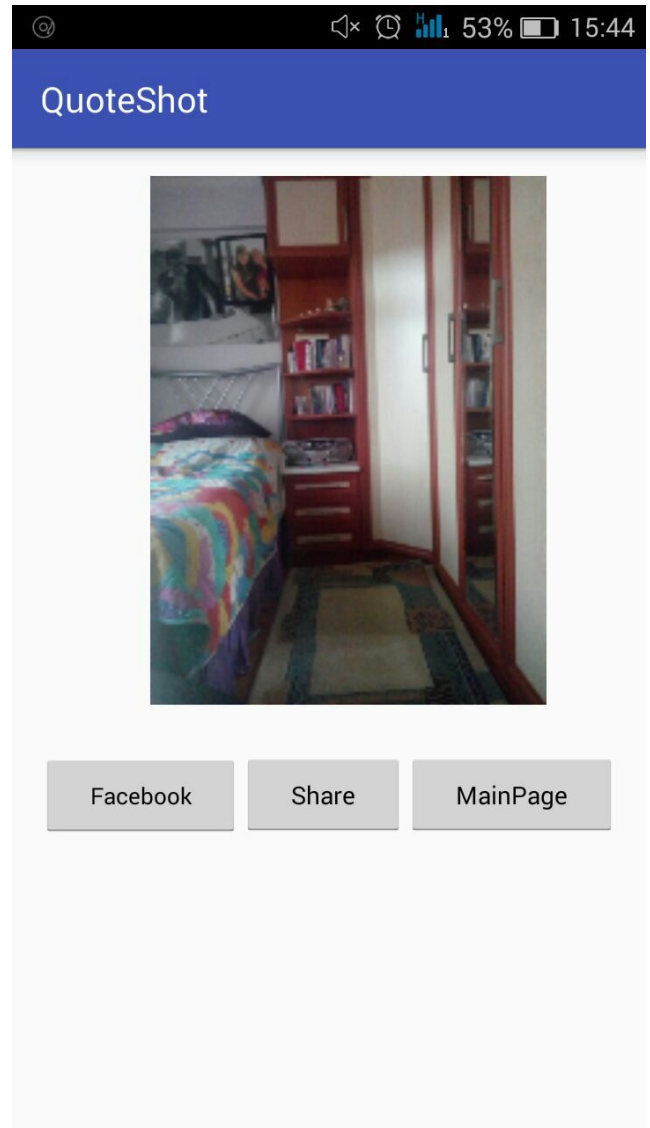
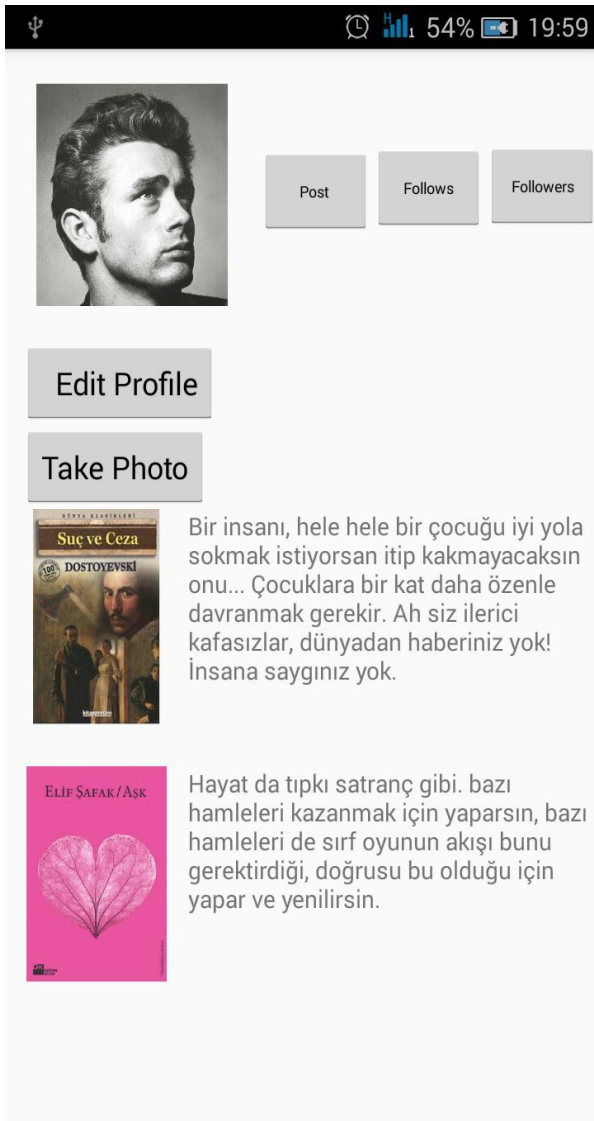
*Figure 3: Login & Register Design1*

Above the two pictures show the login & registration interfaces of QuoteShot. For the login process a user have to enter his username and password. Then user should click login button. If a user have facebook account or google plus account, by clicking bottom of two buttons a user can login to the QuoteShot. If the user is not register of QuoteShot, s/he can click the "Do you like to register" button to register. When the user clicks the button the user see the second picture. In the second picture, there are 5 rows. If a user wants to register of QuoteShot, the user should fill the all of the rows. In the first row, should be written email address. If the email address has used before or user write something invalid the error message will be shown. After entering name and surname, the user should write username. If

the username also has been used the error message will be shown so that everyone has unique username. Finally after entering password and clicking Register button the registration will be finish and the application shows a successful message on the screen. Below the two pictures show the messages of QuoteShot.



*Figure 4: Login & Register Design2*



*Figure 5: Profile Design*

The first picture shows the profile page of user. When the user clicks edit profile the application opens the gallery of the user's smartphone. After selecting one of the picture from the gallery the user can change the profile photo. When the user clicks take photo button the application opens the camera of smartphone. After taking photo the application shows the second picture. User can share his photo on Facebook ,Whatsapp, Instagram or his MainPage by choosing appropriate button.

### **2.2.2 Hardware Interfaces**

This application will work on android phones and tablets. Android devices must have an internet connection in order to run this application. Also since the application based on sharing quotes from books, user can take a photo using camera of the mobile phone and for book exchange a GPS unit must be available in the mobile phone.

### **2.2.3 Software Interfaces**

Since this application is a mobile application, it needs an Android version 4.0 or higher in order to perform. System has another software product which is the network software between database and QuoteShot which is not visible by users. This interface is done via secure network protocols and TCP transport layer application.

### **2.2.4 Communications Interfaces**

The application will use HTTP protocol for communication over internet. The server will be connected to the internet through the Wifi or 3G.

## **2.3 Constraints**

For forend side Android Studio, for backend side Eclipse integrated development environment have been used. The codes has been written in Java. Also, the application has OpenCV library and Tesseract engine. For the server side of the application, Google App Engine which is a platform for developing and hosting web applications in Google-managed data centers has been used. For database, MySQL server has been set up using the Google Cloud SQL service.

## **3. Specific requirements**

This section contains all of the functional and quality requirements of the system in detail for the following stages of development process. Since the audience of this SRS document is not only technical staff but also users, understandability of requirements is tried to be kept in maximum.

The important thing in this section is that the requirements do not contain any design specifics. All the materials that are used for explaining requirements, including diagrams, are characterized by an analysis perspective rather than design perspective. So, understandability is the most important thing in this section.

### **3.1 Functional Requirements**

Detailed information about the functional requirements are given in the section 2.1 with the use case diagram and its descriptions.

### **3.2 Non-functional Requirements**

#### **3.2.1 Usability**

In this section the QuoteShot application will be examined in terms of understandability, learnability, operability and attractiveness.

##### **3.2.1.1 Req 037**

The error messages shall include enough explanation.

##### **3.2.1.2 Req 038**

There shall be a green tick mark end of the successful messages.

##### **3.2.1.3 Req 039**

The dialogue shall include enough explanation.

##### **3.2.1.4 Req 040**

There shall be a red exclamation mark end of the error messages.

##### **3.2.1.5 Req 041**

Almost every decisions of the user, the application checks the what user exactly wants.

##### **3.2.1.6 Req 042**

There shall be relevance icons of buttons and function of buttons to facilitate to understand of users.

#### **3.2.2 Reliability**

Reliability is one of the metrics that are used to measure quality. For reliable software, the system shall be tested during development process. It shall be delivered on time and shall meet the requirements that are specified in this document. In the Quoteshot application when any fault occurs on application or database, it shall be recover in a very short time in terms of reliability.

##### **3.2.2.1 Req 043**

The system should be available always.

##### **3.2.2.2 Req 044**

System should display informative messages when it's components doesn't work properly.

##### **3.2.2.3 Req 045**

Mean time between failures must be at most 2 hours.

### **3.2.3 Performance**

#### **3.2.3.1 Req 046**

QuoteShot will be able to support at least 5.000 users. The capacity can be extended in future if needed.

#### **3.2.3.2 Req 047**

All of the functions that is for retrieving messages, friends lists, posts on the wall of the places etc. should be perform less than 3 seconds.

#### **3.2.3.3 Req 048**

There will be large amount of information to be handled in database such as messages, profile informations, quotes etc. and the server will be enough space to handle this occupation.

### **3.2.4 Supportability**

Since changing is inevitable in today's world, these developments shall designed to any possibility of updating.

#### **3.2.4.1 Req 047**

Design elements should be documented well.

#### **3.2.4.2 Req 048**

Since programming language is object-oriented, program tasks are independent of each other and therefore easier to maintain.

## **4 Data Model and Description**

This part of the SRS is about classes which contains data and their relationships.

### **4.1 Data Description**

This section will give information about the data objects related to this project, the relationship among them, the attributes of the data objects and the complete data model with data objects' functions included.

The software includes 3 data objects: User, book and quote. Users share quotes from book the image and text form. Users can keep the list of books which they own and can specify the the books which they want to exchange. User can share quote by taking photo. After converting image to text form. Image of quote and text of quote are saved in the database.

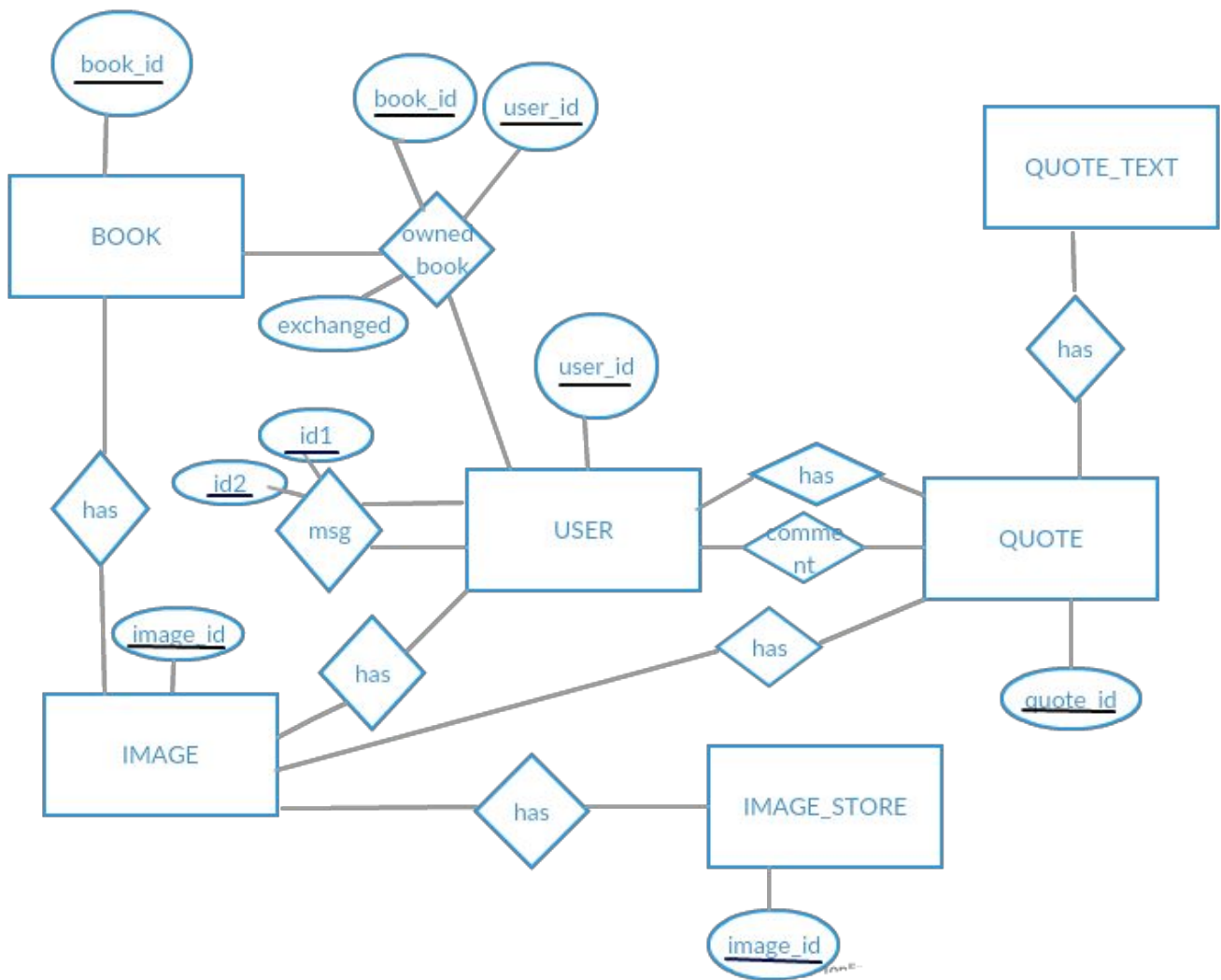


Figure 6: ER Diagram

### 4.1.1 Data objects

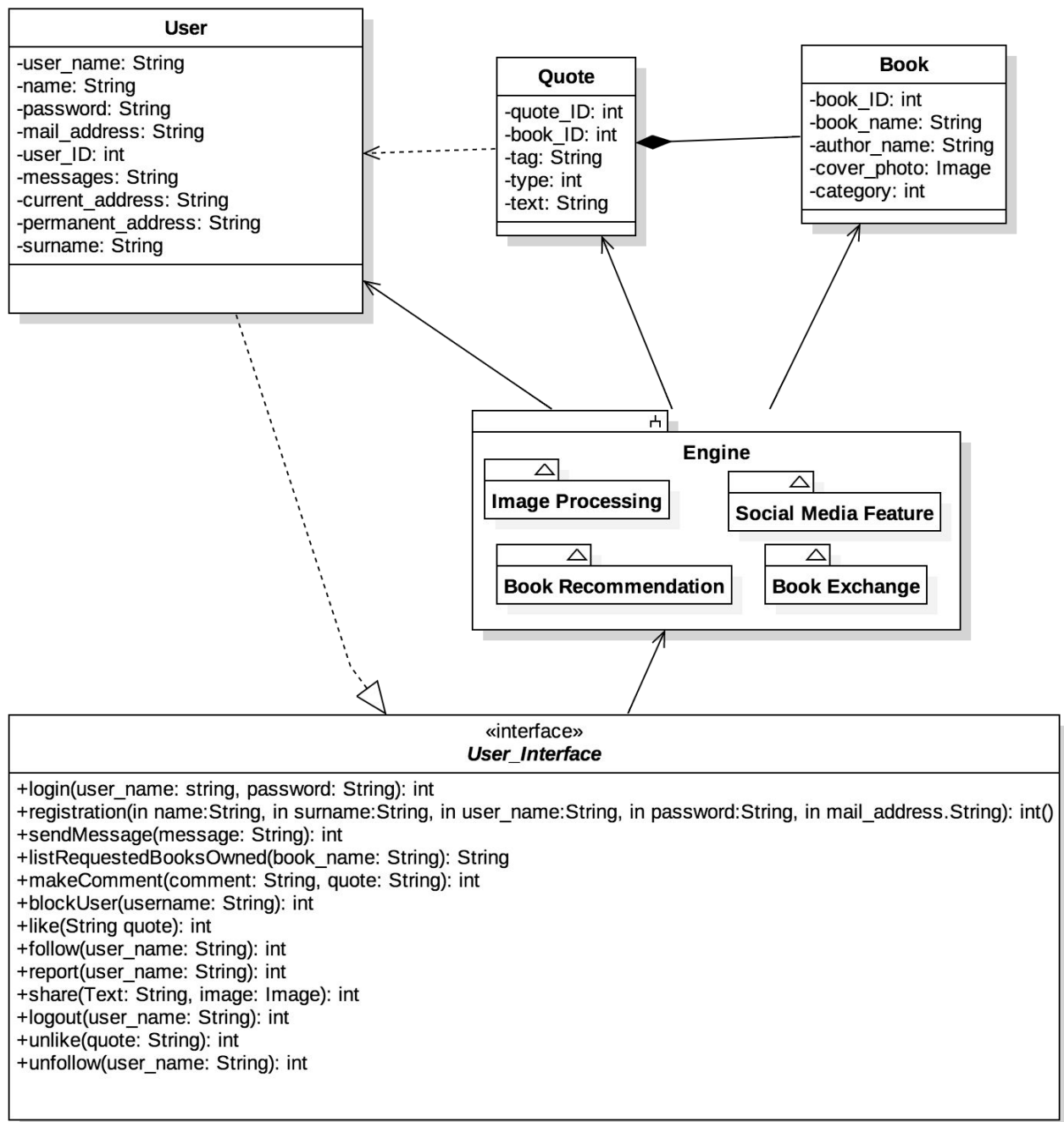


Figure 7: Class Diagram

In the class diagram there are 4 classes. One of them is user interface. User interface is implementation of user. User face also uses the engine which includes book exchange, book recommendation, image processing and social media feature. The engine uses book, quote and user classes. The quote is dependent a user. When the user shares post the quote exists. All of quote must be belong to a book.

## 4.1.2 Data dictionary

### 4.1.2.1 User Class

User
-user_name: String -name: String -password: String -mail_address: String -user_ID: int -messages: String -current_address: String -permanent_address: String -surname: String

*Figure 8: User Class*

User class is designed for users. It stores the user\_name, name, surname, password, email address, userID, messages, current address and permanent address of user. All attributes are private because of security. All of the attribute has getter & setter method to use it. To make simple getter & setter methods were not written. Description of the class attributes are given in the below table.

Access	Return Type / Type	Name	Description
private	String	user_name	the unique alias name of user
private	String	name	name of user
private	String	surname	surname of user
private	String	email_address	email account of user
private	String	password	password of user
private	int	user_id	the unique id of user
private	String	messages	the messages box of user
private	String	current_address	the current address of user
private	String	permanent_address	the permanent address of the user

*Table 18 : User Clas*

#### 4.1.2.2 User Interface

User_Interface
+login(user_name: string, password: String): int +registration(name: String, surname: String, user_name: String, password: String, mail_address.String): int +sendMessage(message: String): int +listRequestedBooksOwned(book_name: String): String +makeComment(comment: String, quote: String): int +blockUser(username: String): int +like(String quote): int +follow(user_name: String): int +report(user_name: String): int +share(Text: String, image: Image): int +logout(user_name: String): int +unlike(quote: String): int +unfollow(user_name: String): int

*Figure 9: User Interface*

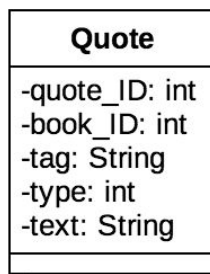
User interface is designed for users. It has 13 methods. Description of the class methods are given in the below table.

Access	Return Type/Type	Name	Description
public	int	login	It takes username and password as arguments then checks the condition of login and returns the status of login.
public	int	registration	It takes name, surname, username, password, email as arguments then checks the condition of registration and returns the status of registration.
public	int	sendMessage	It takes messages as argument and returns status sending message.
public	String	listRequestedBooksOwned	It takes name of a book as argument and returns the lists of users who own the book.
public	int	makeComment	It takes comment and quotes as arguments and returns the status of making comment.

public	int	blockUser	It takes username and returns status of blocking the user.
public	int	like	It takes quote as argument and return the status of liking the quote.
public	int	follow	It takes username as argument and returns the status of following the user.
public	int	report	It takes username as argument and returns the status of reporting(complain) the user.
public	int	share	It takes image and text which wanted to share and returns status of sharing post.
public	int	logout	It takes username and password as arguments then checks the condition of login and returns the status of logout.
public	int	unlike	It takes quote as argument and return the status of unliking the quote.
public	int	unfollow	It takes the username as argument and returns the status of unfollowing of the user.

*Table 19 : User Interface*

#### 4.1.2.3 Quote Class



*Figure 10: Quote Class*

Quote class is designed for quotes. It stores the quoteID, bookID, type, tag and text of quote. All attributes are private because of security. All of the attribute has getter & setter method to use it. To make simple getter & setter methods were not written. Description of the class attributes are given in the below table.

Access	Return Type/Type	Name	Description
private	int	quote_ID	the unique id of quote
private	int	book_ID	the unique id of book
private	String	tag	tags of the quote
private	int	type	the type of quote
private	String	text	itself of quote

Table 20 : Quote Class

#### 4.1.2.4 Book Class

Book
-book_ID: int -book_name: String -author_name: String -cover_photo: Image -category: int

Figure 11: Book Class

Book class is designed for books. It stores bookID, name, author name, cover photo and text category of book. All attributes are private because of security. All of the attribute has getter & setter method to use it. To make simple getter & setter methods were not written. Description of the class attributes are given in the below table.

<b>Access</b>	<b>Return Type/Type</b>	<b>Name</b>	<b>Description</b>
private	int	book_ID	the unique id of book
private	String	book_name	the name of book
private	String	author_name	the name of author of the book
private	Image	cover_photo	the cover photo of the book
private	int	category	the category of the book

*Table 21 : Book Class*

## 5 References

[1] IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984 , vol., no., pp.1-26, Feb. 10 1984, doi: 10.1109/IEEESTD.1984.119205,

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=278253&isnumber=6883>

[2] Appendix C of Don Widrig, Dean Leffingwell, "Managing Software Requirements: A Unified Approach," Addison-Wesley Professional, Release Date: October 1999, ISBN: 0201615932.