|  |
| --- |
| BSc (Hons) in Computing – Year 4 – Software Development |
| Requirements Specification (RS) |
| Social Modifications |

|  |
| --- |
| Keith Feeney  15015556  keith.feeney@student.ncirl.ie  November 2018 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc531533357)

[1.1 Purpose 4](#_Toc531533358)

[1.2 Project Scope 4](#_Toc531533359)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc531533360)

[2 User Requirements Definition 5](#_Toc531533361)

[3 Current Process 10](#_Toc531533362)

[4 Requirements Specification 10](#_Toc531533363)

[4.1 Functional requirements 10](#_Toc531533364)

[4.1.1 Use Case Diagram 10](#_Toc531533365)

[4.1.2 Requirement 1: Create/Edit Profile 10](#_Toc531533366)

[4.1.3 Requirement 2: Upload/Delete media 15](#_Toc531533367)

[4.1.4 Requirement 3: Send message 18](#_Toc531533368)

[4.1.5 Requirement 4: Group chat 21](#_Toc531533369)

[4.1.6 Requirement 5: Chat with Chatbot 23](#_Toc531533370)

[4.1.7 Requirement 6: Rate tattoo shops. 26](#_Toc531533371)

[4.2 Non-Functional Requirements 29](#_Toc531533372)

[4.2.1 Performance/Response time requirement 29](#_Toc531533373)

[4.2.2 Availability requirement 29](#_Toc531533374)

[4.2.3 Recover requirement 29](#_Toc531533375)

[4.2.4 Security requirement 29](#_Toc531533376)

[5 Interface requirements / Application Programming Interfaces (API) 29](#_Toc531533377)

[5.1 GUI 30](#_Toc531533378)

[6 System Architecture 42](#_Toc531533379)

[7 System Evolution 42](#_Toc531533380)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of Social Modifications, a social networking web application and Android app designed for people with tattoos, piercings and other body modifications. As this is a social networking site, Social Modifications will be available to people over the age of 18 only. A chat bot will also be included in the web application and Android app as a help point for various aspects of the social network or just general conversation.

## Project Scope

Social Modifications (SM) is a web application and Android app. It is designed to be a social networking platform for people with, or have an interest in tattoos, piercings and modifications.

SM is a safe space for Users to communicate with other Users and share their body modification experience. The social network also has a chat bot. This chat bot is SM’s alternative to a help section.

With the likes of Facebook, Twitter, Instagram and WhatsApp, these have lengthy Help Sections that can be difficult to navigate through, and even then, the answer may not be clear. With the chat bot, Users can communicate directly to it and get assistance from within the social network. The chat bot also can communicate events, such as the likes of Tattoo Conventions. It can also look up information on tattoo and piercings shops at the User’s request.

The web application and Android app will be copies of each other and they will both need Internet connection. But there will be a cached version available, so Users can still have limited interaction with the social network, even if Internet problems exist.

Users will need to create an account with email and password, or via Facebook’s sign-in API, as only registered Users can use the social network. Google Firebase is used for storage and encryption. System information is stored in Google Firebase Database, while the likes of images will be stored in Google’s Firebase Storage. The Android app will also need to search for accounts, so any linked Facebook account can be used to create an account on SM.

## Definitions, Acronyms, and Abbreviations

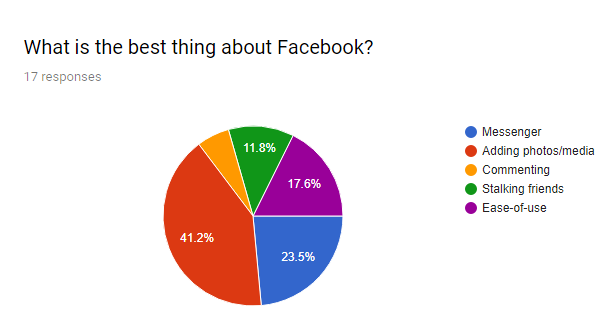
|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| SM | Social Modifications – The working name of the application |
| SN | Social Network |
| User | An end-user that interacts with SM. |
| Firebase Authentication | Used to verify Users as well as handle the register / login |
| Firebase Database | Used to store data in a database (e.g. SQL info) |
| Firebase Storage | Used to store data but not in a database. (e.g. media) |
| AES/AES256 | An encryption method to encrypt passwords. |
| GUI | Graphical User Interface |
| API | Application Programming Interface |
| Modified / Mods | To have tattoos, piercings or other body modifications |

# User Requirements Definition

Having communicated with modified friends about the idea of social networking application designed for them, one major thing kept appearing; simplicity. Many of my friends have said that current social networks have too much content, and most of it is just filler. There’s nothing valuable on them anymore.

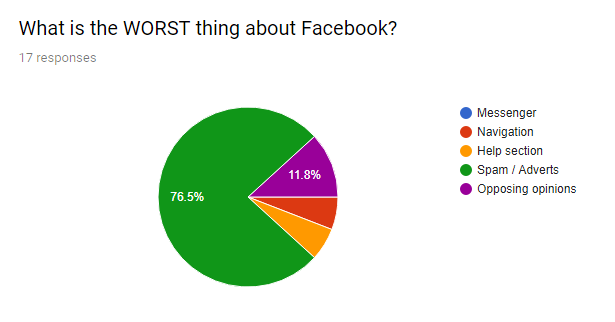
I created a form on Google Forms to see what people thought about social networking in general. As the form was being posted in Facebook Groups and Pages, I thought it would be best to stick with Facebook, as I could guarantee the participants know Facebook to some extent.

The first question was asking what is the best thing about Facebook.



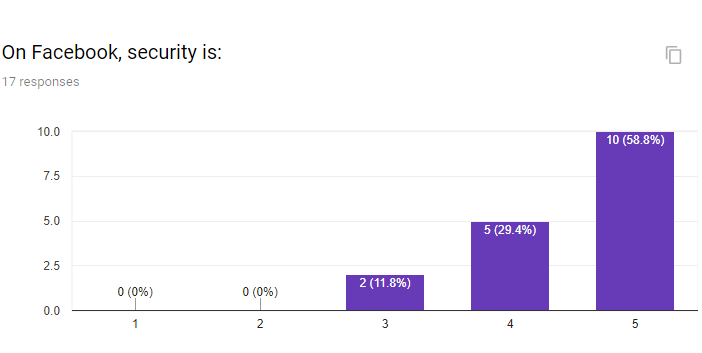
With this, most people decided that adding photos and media was the best thing about Facebook, with Messenger coming in second. Very few people think commenting is important. With Social Modifications, Users need to be able to upload media and well as communicate with each other, as well as look up other Users. Commenting doesn’t seem as important. This may be a future feature.

The second question asked was is the worst thing about Facebook.



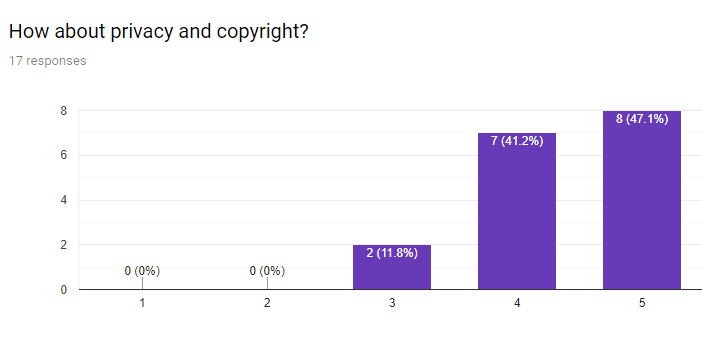
No surprise that ‘Spam/Adverts’ is deemed the worst thing about Facebook. But surprisingly, ‘Opposing opinions’ took the second majority. In SM, how adverts are handled needs to be seriously considered and not take away from the experience of the SN.

The third question looked at Facebook security and how important it is.



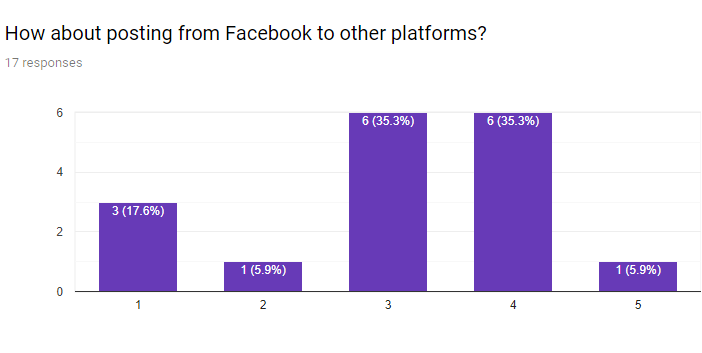
Not a shocker, but security is a major factor for the majority of people. It’s clear that in Social Modifications that data is transmitted securely and that it’s encrypted.

The fourth question looked at Facebook privacy and copyright.



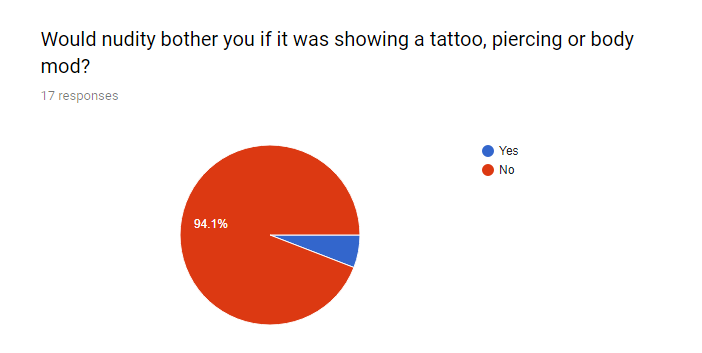
Privacy and copyright are important factors, but they seem to be slightly less important than security. In SM, private albums will exist so that Users can share certain photos with other Users, if they wish to. Another feature is that User’s will be able to put a watermark on their images if they want to.

The fifth question looked at sharing from Facebook to other platforms.



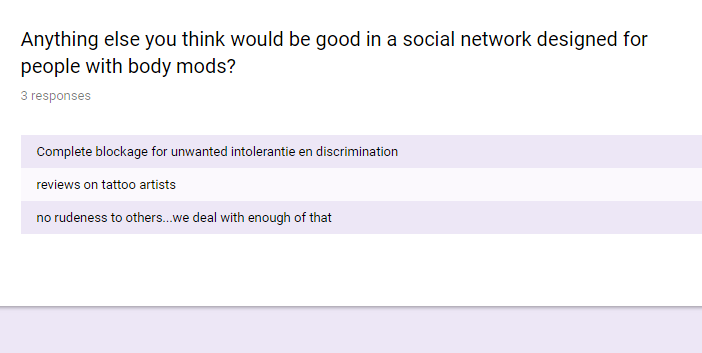
More people have decided that comparing to security and copyright, sharing to other platforms is not required as much. In SM, this could be added in a future update.

The sixth question added whether nudity would bother people I felt this was an important question as from talking with modified friends, modifications and nudity is not a sexual thing, but it’s more like art.



The majority of people agree with what I thought. This also confirms for me that Social Modifications should be for people aged 18 and older.

The final question was an open-ended question, so that people could add what they would like to see in SM.



In SM, approval for new profiles, media, etc. seems to be important and will be implemented. The reviews on tattoo artists is a good idea, but I feel that would need to come from the likes of Google Maps, or Yelp or a similar source. I don’t think it would be beneficial to create a mini-rating site within SM on its initial release. Maybe on a future release.

# Current Process

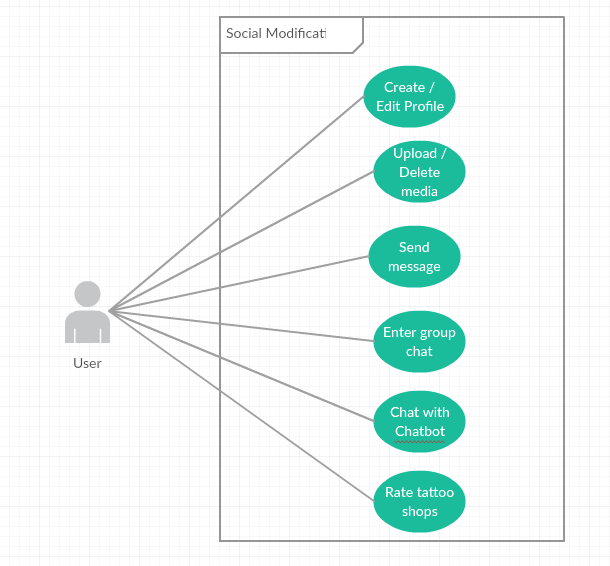
Explain what’s currently out there

# Requirements Specification

## Functional requirements

This section lists the functional requirements in ranked order. Functional requirements describe the estimated outcomes of the System.

### Use Case Diagram



.

### Requirement 1: Create/Edit Profile

#### Description & Priority

This is a very important requirement. Without this, Users will not be able to access SM at all. Users will be able to register an account and if they have one already, they can login. User’s details will be authenticated by Firebase Authentication and it will be held in a Firebase Database.

#### Use Case

**Unique ID:** editProfile

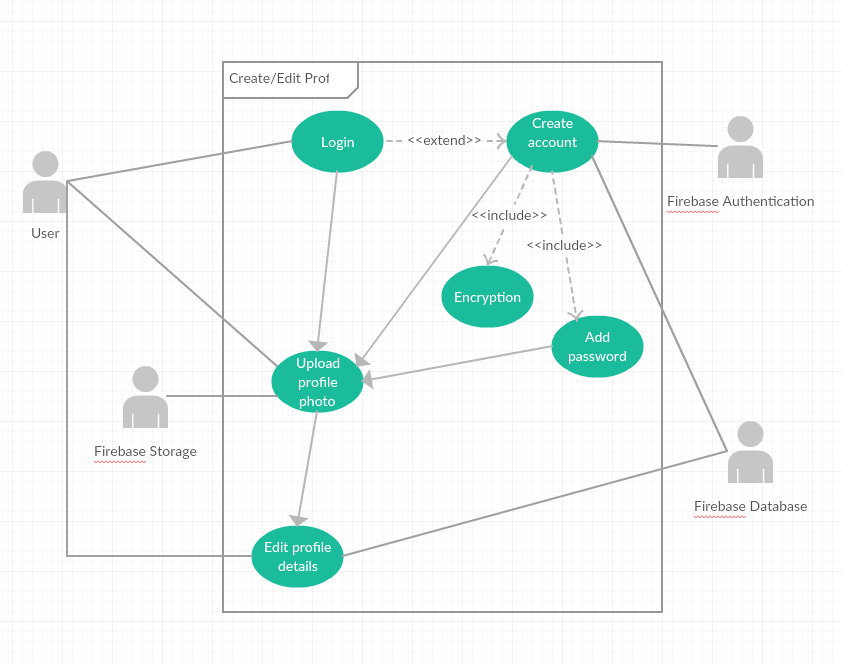
**Scope**

The scope of this use case is to show how the User interacts with the System when creating an account or logging into the System, and also creating their profile.

**Description**

This use case describes how the User creating an account. If the User has previously created an account, they will skip registration and go to the login instead. After this, Users can create and edit their profile.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and is in a wait state for a User.
* It is assumed the Google Firebase is active and working correctly.
* The Login screen is shown to the User.
* A registration link is displayed on the screen.
* A Facebook login button is also available, so Users can login using their Facebook account.
* A token and unique User ID are created on Facebook and on SM to verify the User.
* No passwords are stored on the database, instead, an encrypted version of it is.
* An Error log file is stored within in the app.

**Activation**

The use case starts when a User begins typing into the email and password text fields.

**Main flow**

1. The System displays login screen.
2. The User inputs their email address <See A1, A2, A3, A4, A5 & A7>.
3. The User inputs their password.
4. The System encrypts the User’s inputs with AES.
5. The System creates a connection to Firebase Authentication and Firebase Database. <See E1>
6. The System verifies that the email and encrypted passwords match. <See E2>
7. The System creates a session key and puts it into the cookies.
8. The System shows the Profile page for the User. <See A6>
9. The User taps on Edit Profile.
10. The User uploads a photo
11. The System stores the photo in the Firebase Storage
12. The User edits their details as they require
13. The User selects “Continue”.
14. The System encrypts this information and put it into a Firebase Database.
15. The System shows the User their news feed.

**Alternate flow**

A1 : User uses Facebook to login - Successful

1. The User clicks on “Facebook Login”
2. The User is taken to a Facebook Login Page <See A2>
3. The User enters their email and password for Facebook
4. The User is prompted by Facebook if they want to login to Social Modifications.
5. The User selects “Login”.

<Returns to number 6 in Main Flow>

A2 : User uses Facebook to login - Failed

1. The User clicks on “Facebook Login”
2. The User is taken to a Facebook Login Page <See A2>
3. The User enters their email and password for Facebook
4. The User has not entered the correct details.
5. The User clicks on Back.

<Returns to number 1 in Main Flow>

A3 : User uses Facebook to login – User is already logged in on Facebook

1. The User clicks on “Facebook Login”
2. The User is prompted by Facebook if they want to login to Social Modifications.
3. The User selects “Login”.

<Returns to number 6 in Main Flow>

A4 : User registers - Database

1. The User clicks on “Register”
2. The User inputs their email address <See A7>
3. The User inputs their password
4. The User inputs their password again <See E3>
5. The User enters in their name (or screen name)
6. User enters their date of birth <See E4>
7. The System creates a connection to Firebase Authentication and Firebase Database.
8. The System encrypts the password using AES.
9. The System inputs the details into the Users table.

<Returns to number 6 in Main Flow>

A5 : User registers - Facebook

1. The User clicks on “Register”
2. The User clicks on “Facebook login”

<Returns to number 2 in A1>

A6 : User decides not to edit profile.

<Returns to number 14 in Main flow>

A7 : User created account with Facebook. – Sets password.

<User completes A1>

1. The System takes the User to the Add Password page.
2. The User inputs new password.
3. The User confirms new password
4. The User selects continue.

<Returns to number 7 in Main flow>

**Exceptional flow**

E1 : System cannot connect to Firebase

1. The System cannot connect to Firebase Authentication and Firebase Database.
2. The System displays that technical difficulties are occurring.
3. The System stores what happened into an error log.
4. The System asks the User if they would like to report this error
5. The User selects “Yes”
6. The System sends an email to the developer with the error report.

<Number 6 is skipped if User says “No”>

<Returns to number 1 in Main Flow>

E2: Password not valid - Login

1. The System displays a prompt indicating the User’s email or password is not valid

<Returns to number 1 in Main Flow>

E3: Passwords do not match - Register

1. The System displays a prompt indicating the User’s passwords do not match.

<Returns to number 3 in A4>

E4: User not over 18 - Register

1. The System displays a prompt indicating the User is not over 18.
2. The User clicks “OK” on the prompt.
3. The System locks them out for 10 minutes.
4. The System displays message saying the User has been locked out.

<Returns to number 1 in Main Flow>

**Termination**

When the User is shown their News Feed, this use case ends.

**Post condition**

The System goes into a wait state

### Requirement 2: Upload/Delete media

#### Description & Priority

This Use Case describes how a User uploads or deletes media. This is an important feature as the project would not be a social network without this.

#### Use Case

**Unique ID:** uploadMedia

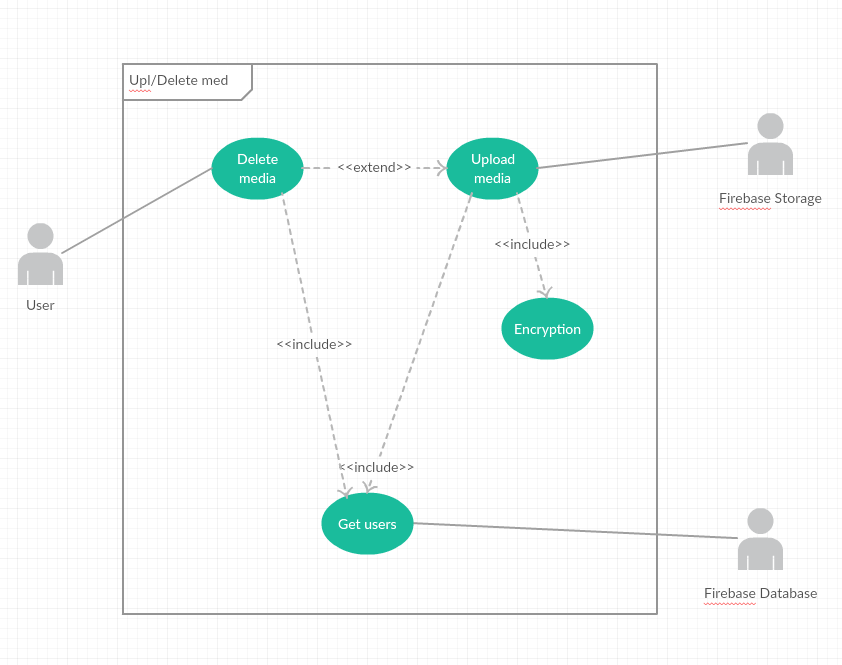
**Scope**

The scope of this use case is to allow a User to upload or delete media (videos and photos) to/from an album.

**Description**

This use case describes how the User can upload or delete media to/from an album. The User will have the option to create an album. The User decides if they wish to share this album with another User.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and working correctly
* The User has created an account through Facebook or via email and password previously (See “Requirement 1: Create/Edit Profile” for how a User creates an account or logs in. This is skipped in this section to avoid repetition)
* Google Firebase is connecting correctly.
* The Main flow contains how to upload media while Alternative flow 1 (A1) explains how to delete media.
* An Error log file is stored within in the app.

**Activation**

This use case starts when the User clicks on Upload Media. On their news feed after they have logged in.

**Main flow**

1. The User selects Upload Media (See A1, E1)
2. The System brings the User to the upload media page.
3. The User selects the relevant file.
4. The System identifies that the file is a photo or video file (i.e. allowed file types)
5. The System brings the User to their albums page
6. The User selects “New album” (See A2)
7. The System allows the User the edit the name of the album
8. The User enters in the new album name
9. The User selects OK
10. The System creates the album with the User’s id and album name (e.g. u12345\_facepics) in Firebase Storage (See A3).
11. The System opens the album and places the file within the album.
12. The System asks the User if they wish to upload more files
13. The User responds “no” (See A4)
14. The System returns the User to their news feed

**Alternate flow**

A1 : <Delete media>

1. The User selects Albums from their news feed.
2. The System opens up the User’s albums.
3. The User selects the relevant album.
4. The System opens said album.
5. The User selects a file.
6. The System plays/views the file and displays options at the bottom of the file, “Prev”, “Next” and “Delete”
7. The User selects Delete
8. The System asks the User if they are sure they wish to delete this file.
9. The User confirms
10. The System deletes the file from Firebase Storage
11. The System checks if there are more files in the selected album. There is. (See A5)

<Returns to number 14 on Main Flow>

A2 : <Albums already created by User>

1. The User selects the relevant album they wish to upload a file into or selects a New album.
2. The System allows the User the edit the name of the album
3. The User enters in the new album name or the already created album if they wish to.

<Returns to number 9 on Main Flow>

A3 : <Albums already exists in User’s account>

1. <Returns to number 11 in Main Flow>

A4: <User wishes to upload more files>

1. The User selects “Yes”

<Returns to number 2 in Main Flow>

A5: <Last file in album is deleted>

1. The System checks if there are more files in the selected album. There is NOT.
2. The System deletes the album from the Firebase Storage

<Returns to number 14 on Main Flow>

**Exceptional flow**

E1 : <Firebase Storage cannot be connected>

1. The System tries to connect to the Firebase storage and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 6 is skipped if User says “no”>

<returns to number 14 on Main Flow>

**Termination**

This Use Case is terminated when the User has uploaded the relevant files and is returned to their news feed.

**Post condition**

The System goes into a wait state

### Requirement 3: Send message

#### Description & Priority

This Use Case describes how a User sends a message to another User. This is an important feature of a social networking site.

#### Use Case

**Unique ID:** sendMesage

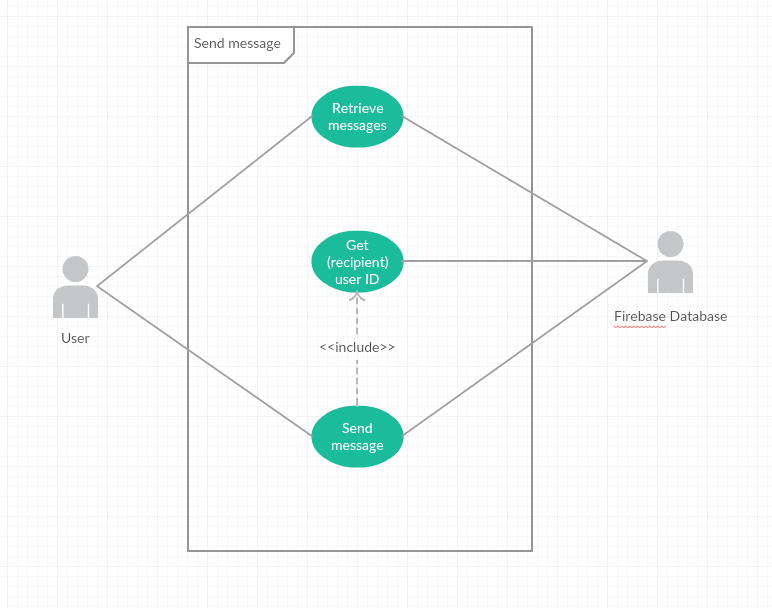
**Scope**

The scope of this use case is to allow a User to send messages to another User.

**Description**

This use case describes how the User can send messages to another User. The messages are stored in the Chat table in the Firebase Database. The messages are not encrypted.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and working correctly
* The User has created an account through Facebook or via email and password previously (See “Requirement 1: Create/Edit Profile” for how a User creates an account or logs in. This is skipped in this section to avoid repetition)
* Google Firebase is connecting correctly.
* Messages (or chats) are stored in the Firebase Database.
* An Error log file is stored within in the app.
* It is assumed this is the User’s first message.
* Only Users who are in the conversation can see the messages (by linking IDs in the Firebase Database)

**Activation**

This use case starts when the selects Messaging from their news feed.

**Main flow**

1. The User selects Messaging from their News Feed.
2. The System presents a list of 10 of the User’s friends (or up to 10 friends) at random to the User.
3. The User selects Send Message.
4. The System opens up a text area where the User can enter another User’s name.
5. The User starts typing the recipient’s name. <See A1>
6. The User selects the relevant recipient.
7. The System retrieved the recipient’s User ID from the Account table in the Firebase Database.
8. The System creates a new Chat table in the Firebase Database.
9. The User enters a message
10. The User selects Send.
11. The System inserts the recipient’s User ID into the “User1” field.
12. The System inserts the sender’s User ID into the “User2” field.
13. The System inserts the message into the “message” field.
14. The System sends an alert to the recipient with the new message.
15. The System stores the chat on the sender’s device.
16. The System stores the chat on the recipient’s device.
17. The User returns to their news feed.
18. After 24 hours, the System deletes the message in the Firebase Database.

**Alternate flow**

A1 : <User has previously sent a message to the recipient>

1. The System displays a list of recipient Users.

<returns to number 6 in Main Flow>

**Exceptional flow**

E1 : <Firebase Database cannot be connected>

1. The System tries to connect to the Firebase Database and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 14 is skipped if User says “no”>

<returns to number 3 on Main Flow>

**Termination**

This Use Case is terminated when the User has successfully sent a recipient User a message.

**Post condition**

The System goes into a wait state

### Requirement 4: Group chat

#### Description & Priority

This Use Case describes how a User connects to a Group Chat Service. The group chat is important as it gives the Users a safe place to communicate.

#### Use Case

**Unique ID:** groupChat

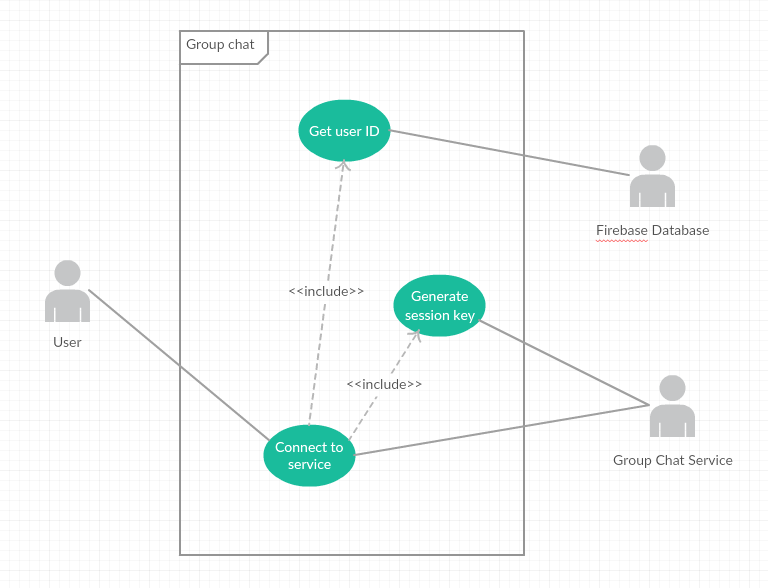
**Scope**

The scope of this use case is to allow a User to login to a Group Chat Service.

**Description**

This use case describes how the User can login to the Group Chat   
Service. The Group Chat Service is a connection to an external source.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and working correctly
* The User has created an account through Facebook or via email and password previously (See “Requirement 1: Create/Edit Profile” for how a User creates an account or logs in. This is skipped in this section to avoid repetition)
* The Group Chat Service is handled by an external source.
* A session key is generated by the System and given to the Group Chat Service.
* An Error log file is stored within in the app.

**Activation**

This use case starts when the selects Group Chat from their news feed.

**Main flow**

1. The User selects Group Chat from their News Feed.
2. The System requests the User to enter their name.
3. The User enters their name
4. The User selects OK (See A1)
5. The System retrieves the User’s ID from the Firebase Database (See E1>
6. The System creates a session key
7. The System logs the User into the Group Chat Service with the User’s ID and the session key. (See E2)
8. The System send the Group Chat Service the name the User typed.
9. The System redirects the User to the Group Chat Service.

**Alternate flow**

A1 : <User has not entered their name>

1. The System retrieves the User’s ID from the Firebase Database
2. The System sets the Username to U + their ID. (e.g. u12345)

<returns to number 6 in Main Flow>

**Exceptional flow**

E1 : <Firebase Database cannot be connected>

1. The System tries to connect to the Firebase Database and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 9 is skipped if User says “no”>

1. The System sends the User to their News Feed

E2 : <Group Chat Service cannot be connected>

1. The System tries to connect to the Group Chat Service and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 11 is skipped if User says “no”>

1. The System sends the User to their News Feed

**Termination**

This Use Case is terminated when the User has successfully logged into the Group Chat Service.

**Post condition**

The System goes into a wait state

### Requirement 5: Chat with Chatbot

#### Description & Priority

This Use Case describes how a User interacts with a Chatbot. Chatbot is a requirement as help should be available to the User.

#### Use Case

**Unique ID:** chatbot

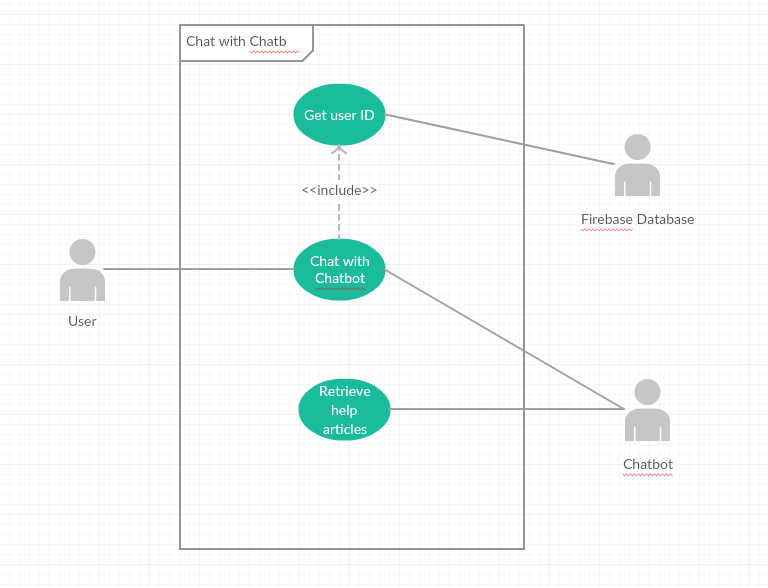
**Scope**

The scope of this use case is to show how a User interacts with the Chatbot.

**Description**

This use case describes how the User interacts with Chatbot. Chatbot is used instead of having a help section. Chatbot is the help section.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and working correctly
* The User has created an account through Facebook or via email and password previously (See “Requirement 1: Create/Edit Profile” for how a User creates an account or logs in. This is skipped in this section to avoid repetition)
* Help articles are stored in Firebase Storage as web (HTML) files.
* The Chatbot is active and working correctly.
* All conversations with Chatbot are recorded.
* An Error log file is stored within in the app.

**Activation**

This use case starts when the selects Help from their news feed.

**Main flow**

1. The User selects Help from their News Feed.
2. The System retrieves the user’s ID in Firebase Database. <See E1>
3. The System opens Chatbot. <See E2>
4. Chatbot sends message to welcome user.
5. The User types in their question and presses the enter key. <See A1>
6. Chatbot ignores words less than 4 characters long.
7. Chatbot searches articles in the Firebase Storage <See A2, E1>
8. Chatbot matches the search terms with the most relevant article.
9. Chatbot displays the article to the User.
10. The User selects OK. <See A3>
11. The System send the User to their news feed.

**Alternate flow**

A1 : <User does not enter any text>

1. The User presses the enter key.
2. Chatbot displays a message saying no text was entered

<returns to number 5 in Main Flow>

A2 : <Chatbot cannot find any relevant article>

1. Chatbot displays a message that no relevant articles were found.
2. Chatbot displays a message saying no text was entered

<returns to number 5 in Main Flow>

**Exceptional flow**

E1 : <Firebase Database/Storage cannot be connected>

1. The System tries to connect to the Firebase Database/Storage and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 6 is skipped if User says “no”>

1. The System sends the User to their News Feed

E2 : <Chatbot cannot be connected>

1. The System tries to connect to the Chatbot and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 7 is skipped if User says “no”>

1. The System sends the User to their News Feed

**Termination**

This Use Case is terminated when the User has viewed an article from Chatbot and has returned to their news feed.

**Post condition**

The System goes into a wait state

### Requirement 6: Rate tattoo shops.

#### Description & Priority

This Use Case describes how a User is able to rate tattoo shops. This is important as this is a project about a site for people with modifications. The users should be able to go to a tattooist / piercer they trust and where other trust as well.

#### Use Case

**Unique ID:** rateShop

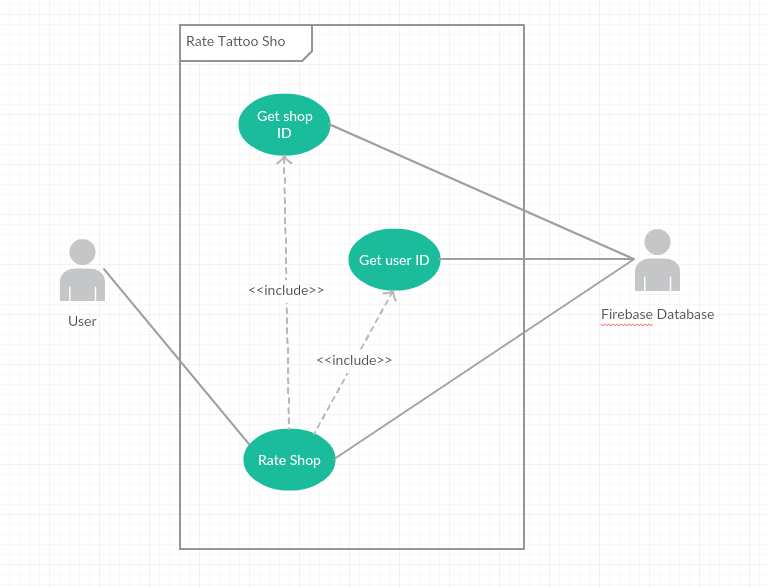
**Scope**

The scope of this use case is to show how a User can rate a tattoo or piercing shop.

**Description**

This use case describes how the User can rate a tattoo or piercing shop. They will also be able to leave anonymous comments about the shops.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The System is active and working correctly
* The User has created an account through Facebook or via email and password previously (See “Requirement 1: Create/Edit Profile” for how a User creates an account or logs in. This is skipped in this section to avoid repetition)
* Shops are stored in the Firebase Database.
* Shop records are created by admins (outside of system)
* Users can email admin/developer to add a shop to the Shops records.
* An Error log file is stored within in the app.

**Activation**

This use case starts when the selects Rate Shops from their news feed.

**Main flow**

1. The User selects Rate Shops from their News Feed.
2. The System opens the main Rate Shop page from the Firebase Database. <See E1>
3. The User selects the relevant shop from the list. <See A1>
4. The System opens the Shop record.
5. The User gives the shop a rating from 1 to 10.
6. The User leaves a comment.
7. The User selects Submit.
8. The System modified the Shop record with the User’s rating and comment.
9. The User selects News Feed
10. The System sends the User to their news feed.

**Alternate flow**

A1 : <User does a search for a shop>

1. The User selects Search.
2. The User enters the shop name or the location.
3. The System retrieves the relevant results from the Firebase Database.

<returns to number 3 in Main Flow>

**Exceptional flow**

E1 : <Firebase Database cannot be connected>

1. The System tries to connect to the Firebase Database and fails.
2. The System stores what happened into an error log.
3. The System asks the User if they would like to report this error
4. The User selects “Yes”
5. The System sends an email to the developer with the error report.

<Number 6 is skipped if User says “no”>

1. The System sends the User to their News Feed

**Termination**

This Use Case is terminated when the User has rated and left a comment on the Shop record.

**Post condition**

The System goes into a wait state

## Non-Functional Requirements

This section specifies other particular non-functional attributes required by the System. Examples are provided below

### Performance/Response time requirement

The System must be able to provide a response to the User in less than 2 seconds. When the System cannot connection to Firebase or the Group Chat Service, a response to the User must be presented in 6 seconds of less.

### Availability requirement

Social Modifications must be available 24/7. To ensure this is the case, 2 back-up connections to Firebase will be developed.

### Recover requirement

At least 2 backups in separate physical locations must be available. If Firebase or Group Chat Service is able to connect to the System, the system will identify the admin/developer at once.

### Security requirement

No plain text passwords are stored anywhere. Encrypted passwords using AES will be stored in the Firebase Database.

Session keys and cookies will be used.

# Interface requirements / Application Programming Interfaces (API)

Input from User

When the user is inputting text, a text field will be used. This is recognised by the System using the variable name.test

Firebase Connection / API

An example of how tables in the firebase database would be used is the Users tables. Tables related to the Users in the Firebase Database are Profile, Messages, Albums and Friends.

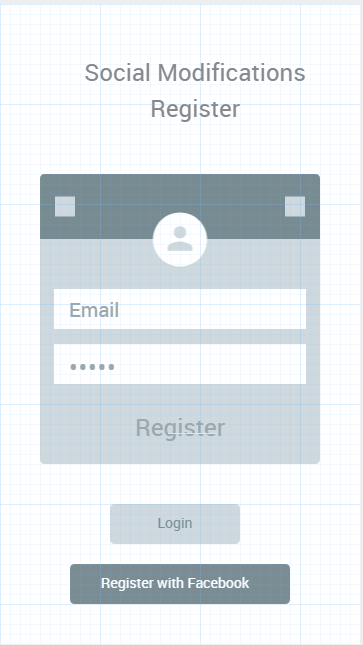
For the user to see their profile, the System will read this information using a connection to the server using API and JSON (a machine-readable format). This ensures (e.g.) all profile layouts are the same for each User.

Similarly, connection to the Chatbot and the Group Chat Service will be using a similar API.

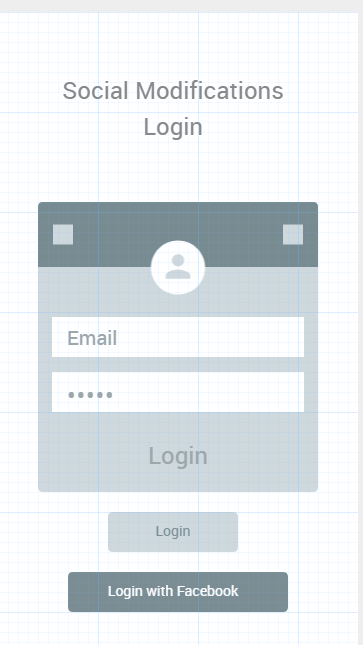
The API will be created by the developer.

## GUI

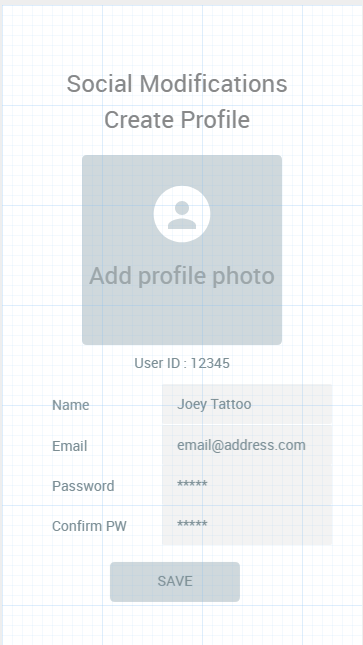
These mock-ups show what Social Modifications would look like on an Android device. In the Web Application, it would look very similar.



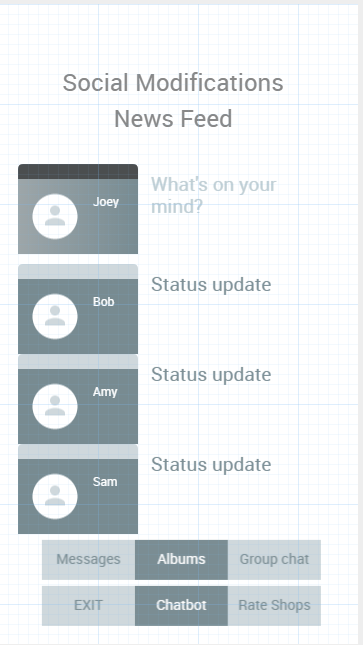
1. This is the main Registration page that is first shown to users when they first view Social Modifications.



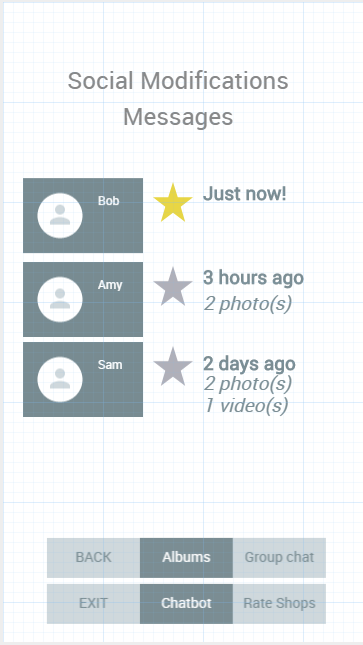
1. This is the login page. The user can select Login from the previous page to get here.



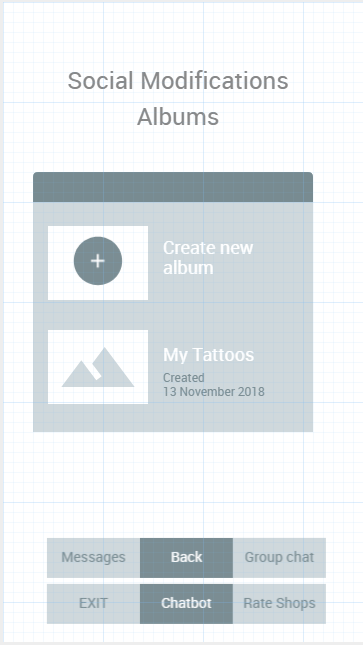
1. This is the Create Profile page. The Edit Profile is similar. Some of the entries are missing due to space. But they would be in Social Modifications when released.



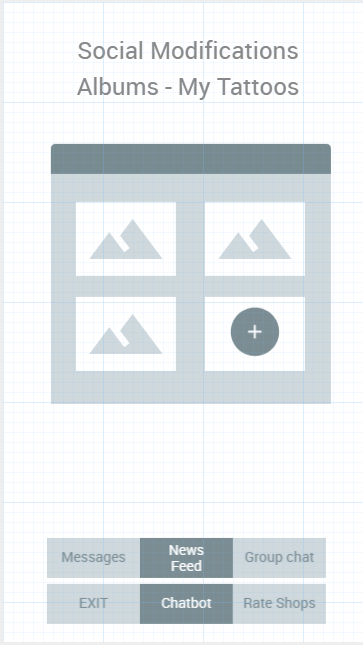
1. This is the News Feed page and would show after the Create/Edit Profile page.



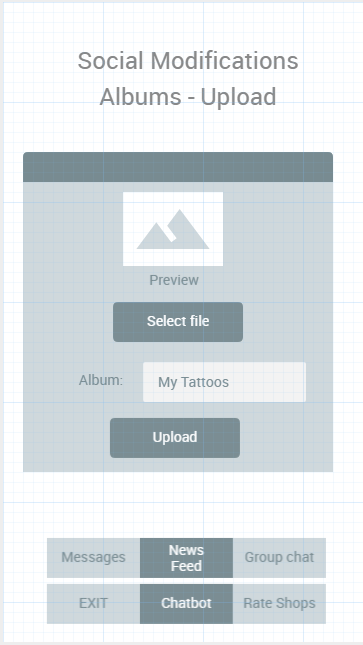
1. This is the Messaging page. Showing the messages would be similar to the messaging the Chatbot, which is further in this document.



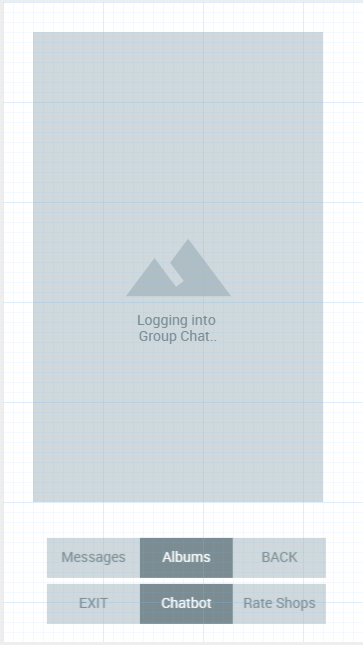
1. The Albums main page, where a User can see their media albums and create a new album.



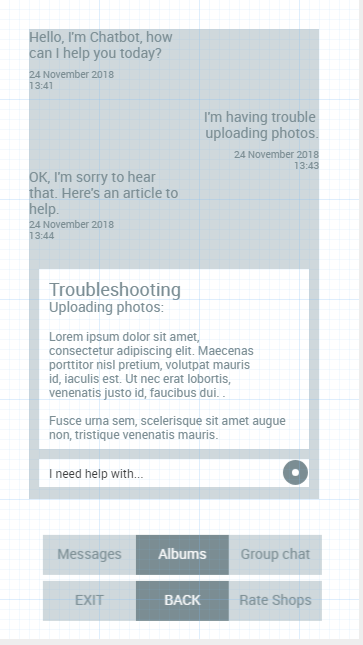
1. This is how the user would see what is in the albums. For example, this user selected the album “My Tattoos”.



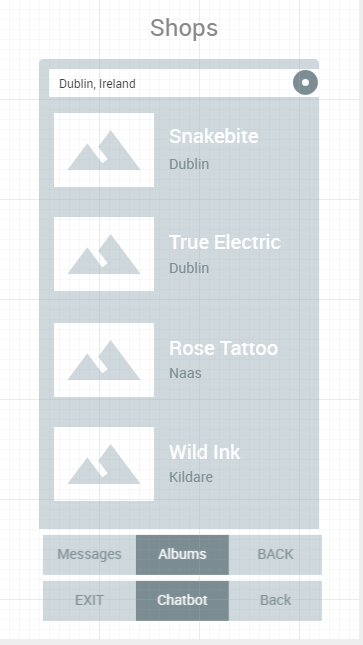
1. After selecting the Plus (+) button to add new media, this is what the user would see. After they select “Select file” and selected the file, a preview of it would show. After uploading, the System would bring them to the main Albums page, with the new media added.



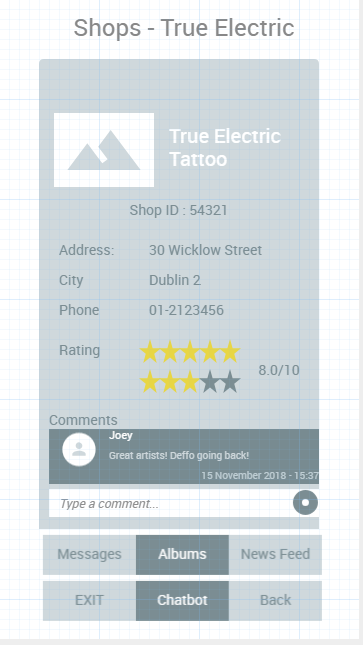
1. When the User selects “Group Chat”, the User would be shown this before the System logs the User into the Group Chat Service.



1. After selecting “Chatbot”, the User would be shown this screen to communicate with the Chatbot.

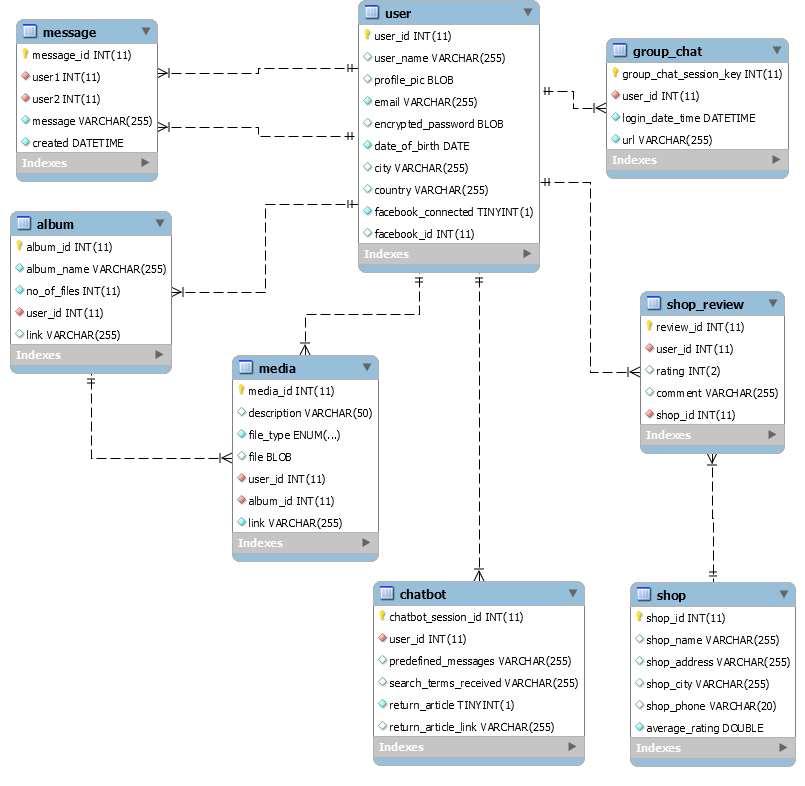


1. This shows how a user would search for a tattoo or piercing shop. This user has searched for “Dublin, Ireland” and got these results.



1. After selecting a shop, this is where the User can leave comments or a rating.

# System Architecture



(Note: This was created in MySQL Workbench. “TINYINT” is actually a Boolean type)

# System Evolution

Over time, Social Modifications could have the likes of video and phone calling. It would also be able to support a video instead of an image on the profile picture.