|  |
| --- |
| BSc (Hons) in Computing – Year 4 – Software Development |
| Requirements Specification (RS) |
| Body Branding Bookings (3B) |

|  |
| --- |
| Joey Tatú  15015556  joey.tatu@student.ncirl.ie  November 2019 |

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](#_Toc24623458)

[1.1 Purpose 4](#_Toc24623459)

[1.2 Project Scope 4](#_Toc24623460)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc24623461)

[2 User Requirements Definition 5](#_Toc24623462)

[2.1 Pages, profiles and bookings 5](#_Toc24623463)

[2.2 Messaging service 6](#_Toc24623464)

[2.3 Payments 6](#_Toc24623465)

[2.4 Automated appointments and stock control 6](#_Toc24623466)

[3 Current Process 6](#_Toc24623467)

[3.1 Personalisation 6](#_Toc24623468)

[3.2 Optimisation 7](#_Toc24623469)

[4 Requirements Specification 8](#_Toc24623470)

[4.1 Functional requirements 8](#_Toc24623471)

[4.1.1 Use Case Diagram 8](#_Toc24623472)

[4.1.2 Requirement 1: Create/Edit Profile 8](#_Toc24623473)

[4.1.3 Requirement 2: Upload/Delete media 12](#_Toc24623474)

[4.1.4 Requirement 3: Send message 15](#_Toc24623475)

[4.1.5 Requirement 4: Group chat 18](#_Toc24623476)

[4.1.6 Requirement 5: Chat with Chatbot 20](#_Toc24623477)

[4.1.7 Requirement 6: Rate tattoo shops. 23](#_Toc24623478)

[4.2 Non-Functional Requirements 25](#_Toc24623479)

[4.2.1 Performance/Response time requirement 25](#_Toc24623480)

[4.2.2 Availability requirement 25](#_Toc24623481)

[4.2.3 Recover requirement 25](#_Toc24623482)

[4.2.4 Security requirement 26](#_Toc24623483)

[5 Interface requirements / Application Programming Interfaces (API) 26](#_Toc24623484)

[5.1 GUI 26](#_Toc24623485)

[6 System Architecture 39](#_Toc24623486)

[7 Google Firebase Database Architecture 40](#_Toc24623487)

[8 System Evolution 40](#_Toc24623488)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of Body Branding Bookings (3B). 3B is an appointment and booking managing web service system, specifically designed for tattooists, piercers and body modifiers (i.e. the artists), and their clients. Features include personalisation for clients, and optimisation techniques and protocols for the system itself.

## Project Scope

3B is a web service system. The fundamental feature of 3B is to assist artists generate times they are available to work and have their clients book an appointment. Another feature is that artists have a page on the site for their work and shop location that clients can rate and leave comments about their experience.

The developer has many tattoos and piercings and knows first hand how unnecessarily complicated booking a tattoo appointment can be and booking an artist that can be trusted. From dealing with deposits, changing appointment dates due to conflicts, it seems that this is far more complex then it needs to be. The project will address this.

3B will have an artist area and a client area. The artist will be able to set when they are available to work and set the likes of prices, sizes, locations (of tattoo/piercing), how long it will take to complete, and so on. Currently, there is no appointment service that fully caters to artists. With body modification becoming more and more popular, it is becoming a necessity.

Another feature includes personalisation. This will be aimed at the client. An example of this would be when the client books a tattoo, they would see artist’s pages for piercings and other body modifications, as well as shops nearby to their booked appointment.

The web application will need an Internet connection to operate correctly. However, a cached version of the appointment schedule for the artists and what appointments have been booked will be generated, so users will not have an issue viewing these, even if internet or capacity issues occur.

Users will need to create an account with email and password, or via social login (e.g. Facebook or Google) as only registered users will be able to create their work schedule and view their appointments, and use the service.

(AWS) Cloud 9 will be used to develop the prototype to create one Use Case and have it working. It will be further developed in Cloud 9 after this, but it is expected that it be transferred to its own domain and hosting before completion. Cloud 9 uses website coding such as HTML, JavaScript, jQuery. MySQL and SQLite will be used for the databases. Encryption will also be used in communication to the databases.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| 3B | Body Branding Bookings, the working name of the project. |
| Artist | A tattooist, body piercer or a body modifier |
| Client | A customer of the Artist |
| User | Either an artist or client. |
| (AWS) Cloud 9 | A cloud-based integrated development system (IDE) for writing, running and debugging code. |
| MySQL / SQLite | A database management system used in Cloud 9. |
| AES | A type of encryption. |

# User Requirements Definition

## Pages, profiles and bookings

Artist pages and client profiles are essentially the same thing, but with different fields and privacy settings. Pages are for finding, rating and leaving a review for the artist, whereas the client profiles will be private and not have these features. The client decides what information is public to a particular artist when booking an appointment.

Artists will be inserting the dates and times they are available to work and the client will be selecting a time slot that the artist has defined. These will be implanted using standard website coding, such as HTML, CSS and JavaScript and will be put into a secure database using SQLite.

## Messaging service

The messaging service will be created using (the coding language) Earling. However, if this becomes unfeasible, a messaging service like SendBird will be used instead. It seemingly has the most simplified API to use for this Project.

## Payments

All payments will be handled using an online service. The ideal solution would be to use PayPal and their API, but Visa and Mastercard’s API will also be considered.

## Automated appointments and stock control

This is where AI plays a role for this project. The idea is that if a client books 4 or more appointments in a 4-month period, the AI will contact the client and ask them if they wish to book an appointment that would fall on a date and time similar to their previous appointments.

With stock control, the artist will input their stock and how much is used for each appointment, such as piercing needles, tattoo ink, gloves, alcoholic wipes, and so on. When a client books an appointment, that stock is set to be removed from the inventory when the appointment is complete. The AI will monitor the stock amounts, and will add it to an order list when it reaches below a certain number, which is set by the artist.

These will be implemented using Java and Java Standard Pages (JSP).

# Current Process

## Personalisation

Brick-and-mortar stores are beginning to put more digital knowhows into their stores. For the clients, this is a great benefit for them by making their experience better. Some of the brick-and-mortar stores are not developing their business’ technological skills, in regard to customer interaction. (Betzing, et al., 2018) However, there are methods of doing this.

When developing from a brick-and-mortar store to a more digitised one, the likes of sensors such as cameras and facial recognition devices can figure out the basics of a person; such as their height, gender, and approximate age. A person’s facial expressions can also be considered to explore whether the client is in a positive or negative mood and whether they are enjoying the service. This data can be obtained by using smart devices on a local network. (Betzing, et al., 2018) (Webner, 2019)

This concept is similar to the physical store Amazon Go by Amazon, where Artificial Intelligence is used to keep track of the clients and store items. (McFarland, 2018)

In this project, it is hoped that a personal experience can be added for the artists and clients. It is expected that personalisation with be sought from a user’s profile, such as their age, gender.

In an example of how this could work; a male client who has just turned 18 and would like to get a tattoo. The system would ask the client to heavily consider whether they would want the tattoo, and to seriously consider avoiding visible areas such as the face, head or hands.

However, if the client is a male in their mid-40s, this message would not be shown to them as the thought process for the client would be different. It could be assumed the client already has a lot of tattoos. Instead, an upload link to share their previously obtained tattoos would be shown. This is developed further in the Requirement Specifications.

## Optimisation

Another consideration of this project is optimisation. If one takes the clothing and textile industry as an example, one can optimise what colours, fabrics and designs are in a high demand. This is completed by eliciting and gathering “Product Usage Information (PUI)”. This is obtained from clients, experts and investors. From this data, the latest styles and what the client wants are known. (Hribernik, et al., 2019)

In this project, optimisation could be used for the general size of the tattoo, the colours of the ink used. If, for example, a lot of tattoos are small and use only use 3 or 4 colours, the system would recognise this and automatically order new colour inks to the artist. On that note, the artist could input how much ink would be used for a particular size and machine learning could be used to order and maintain stock with the artist.

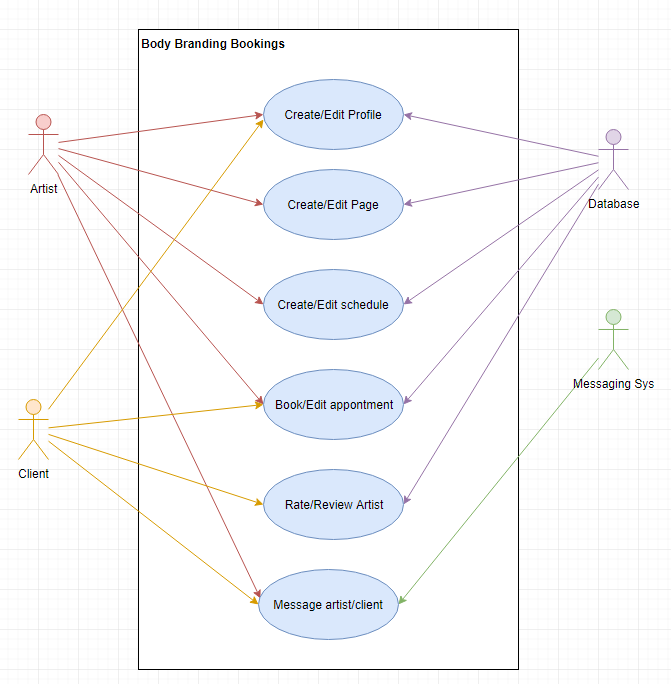
The same would also go for piercings, if a certain ring or stud is used on a regular basis, these could be automatically ordered. The likes of stock control for piercers could also be controlled here. This will be developed with the Requirement Specifications.

# 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, the user will not be able to access 3B. The user’s email and password will be encrypted and authenticated using AES.

#### Use Case

**Unique ID:** createProfile

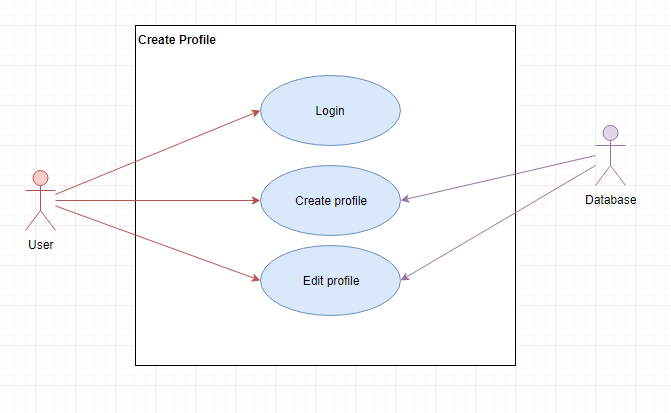
**Scope**

The scope of this use case is to show how the User interacts with the System when creating and editing their profile.

**Description**

This use case describes how the user creates a profile. Alternative flows consider different types of accounts as well as editing.

**Use Case Diagram**



**Flow Description**

**Precondition**

* The system is active and is in a wait state for a user.
* It is assumed the user has previously created an account
* The user is currently not logged into the system.

**Activation**

The use case starts when a user logging into the system.

**Main flow**

1. The System displays login screen.
2. The User inputs their email address and password, and selects Login <See A1, E1, E2 >
3. The System encrypts the User’s inputs with AES.
4. The System creates a connection to MySQL database.
5. The System verifies that the email and encrypted passwords match.
6. The System creates a session key and puts it into the cookies.
7. The System displays the main home page.
8. The User taps on Create Profile. <See A1>
9. The User uploads a photo.
10. The System stores the photo in the Database
11. The User edits their details as required
12. The User selects “Continue”.
13. The System encrypts this information and put it into the MySQL database.
14. The System shows the User the home page.

**Alternate flow**

A1: Editing profile

1. The User clicks on Edit Profile
2. The System loads the User’s profile data into the editable fields.
3. The User edits their photo and profile as they desire.

<Returns to number 13 in Main Flow>

A2: User uses Facebook to login - Successful

1. The User clicks on “Facebook Login”
2. The User is taken to a Facebook Login Page
3. The System waits (for a response from Facebook API)
4. The System receives a response from Facebook
5. The System logs the user into the System <See A2>

<Returns to number 6 in Main Flow>

A3: User uses Facebook to login - Failed

1. The System cannot log the User into the System

<Returns to number 1 in Main Flow>

**Exceptional flow**

E1 : System cannot connect to MySQL

1. The System cannot connect MySQL Database
2. The System displays a message to the user sating that technical difficulties are occurring.
3. The System stores what happened into an error log.

<Returns to number 1 in Main Flow>

E2: Password not valid - Login

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

<Returns to number 1 in Main Flow>

**Termination**

When the User has successfully created or edited their profile, this use case terminates.

**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

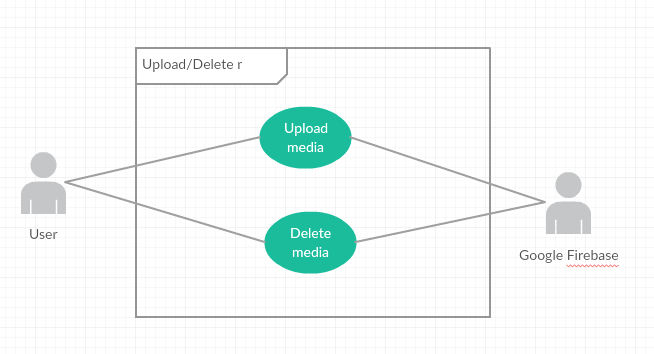
**Scope**

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

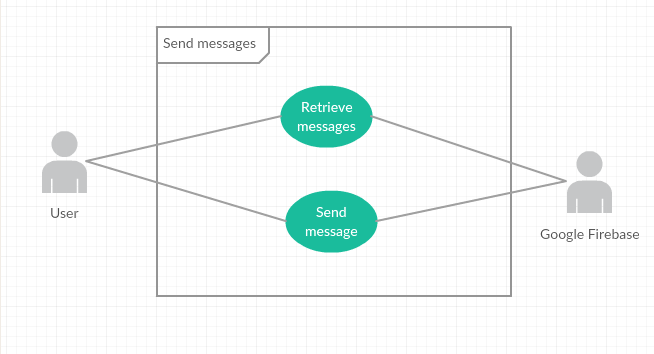
**Scope**

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

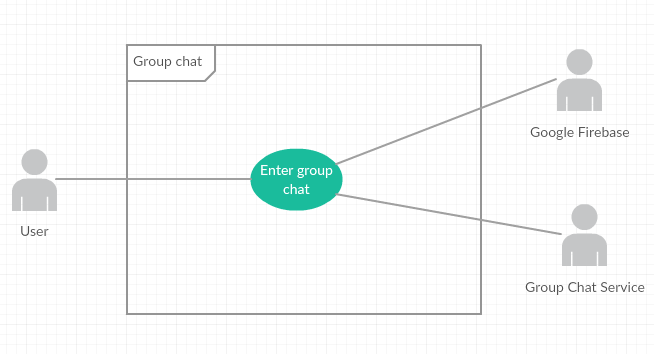
**Scope**

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

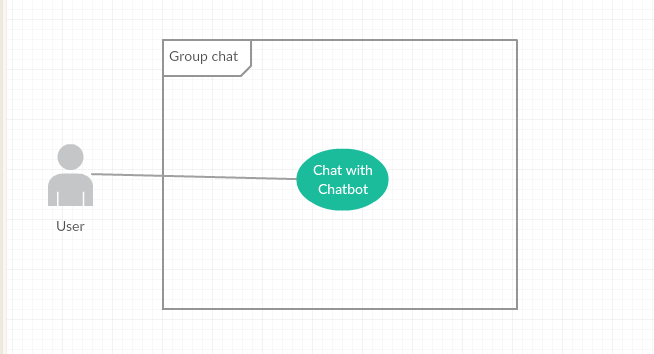
**Scope**

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

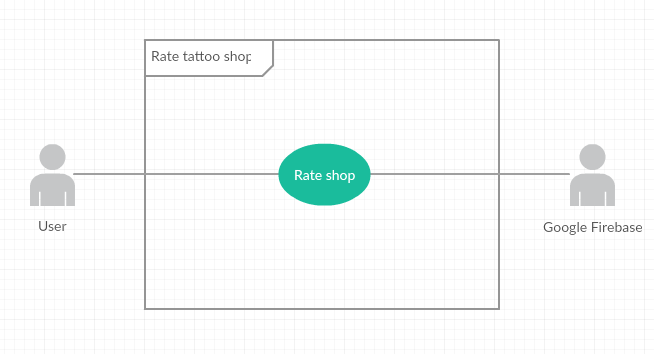
**Scope**

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 unable to connect to the System, the system will identify the admin/developer at once in the forms of email.

### 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.

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.

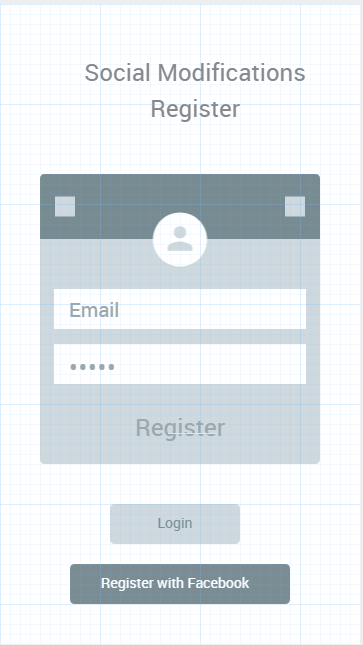
For the user to see their profile, the System will read this information using a connection to the server using API and JSON (a machine-readable format). This ensures (e.g.) all profile layouts are the same for each User.

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

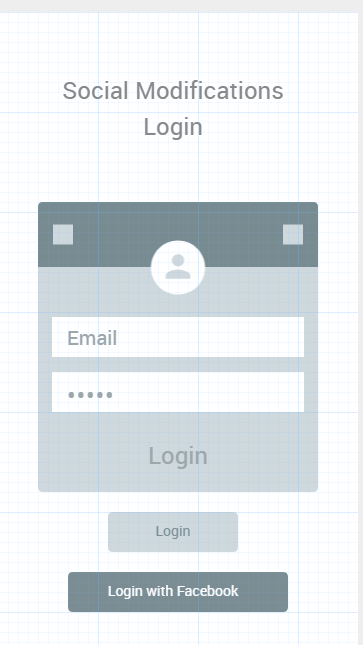
The API will be created by the developer.

## GUI

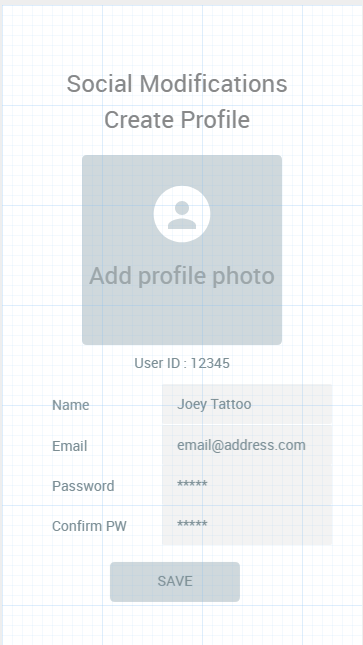
These mock-ups show what Social Modifications would look like on an Android device. In the Web Application, it would look very similar.



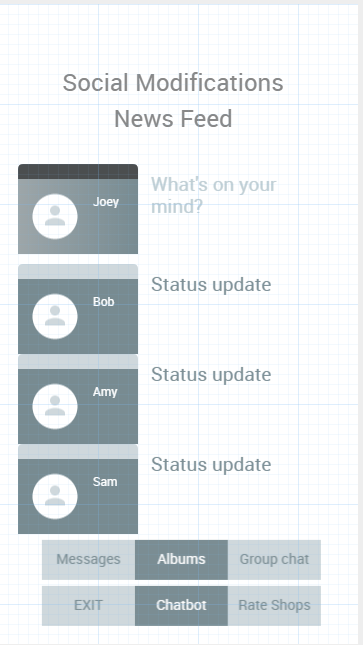
1. This is the main Registration page that is first shown to users when they first view Social Modifications.



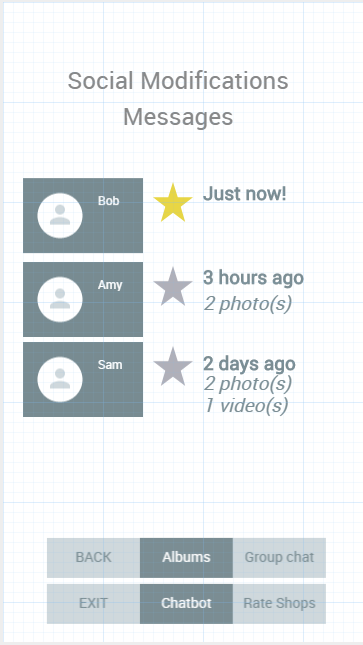
1. This is the login page. The user can select Login from the previous page to get here.



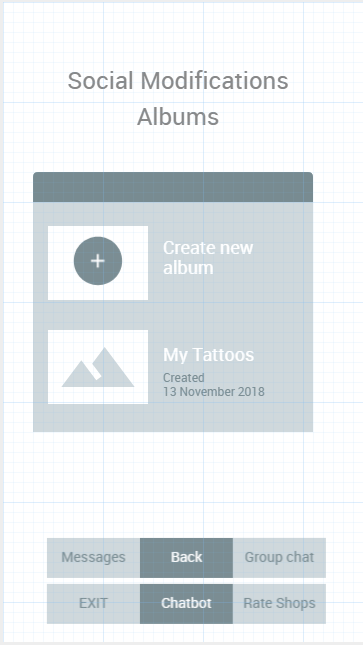
1. This is the Create Profile page. The Edit Profile is similar. Some of the entries are missing due to space. But they would be in Social Modifications when released.



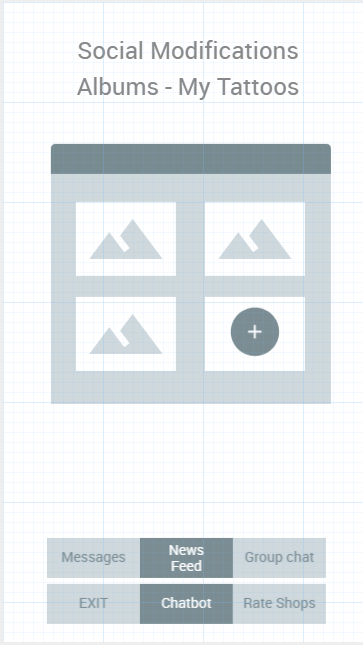
1. This is the News Feed page and would show after the Create/Edit Profile page.



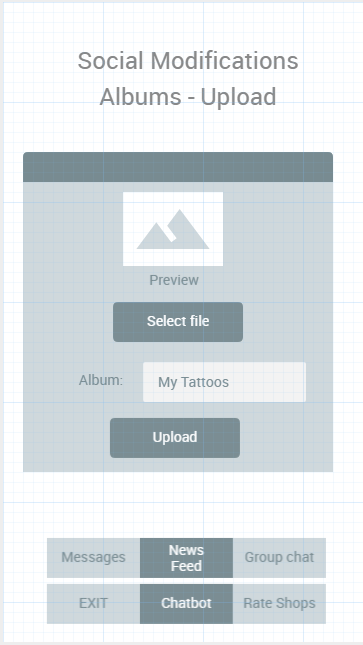
1. This is the Messaging page. Showing the messages would be similar to the messaging the Chatbot, which is further in this document.



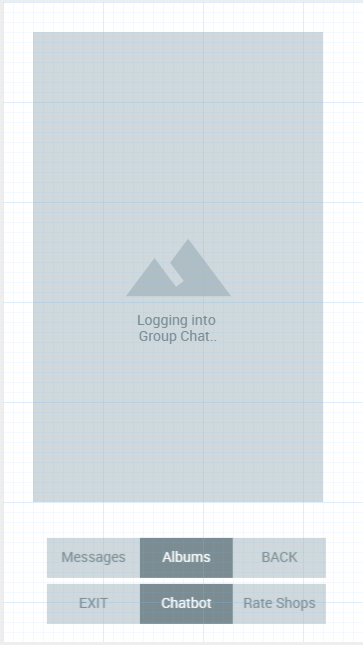
1. The Albums main page, where a User can see their media albums and create a new album.



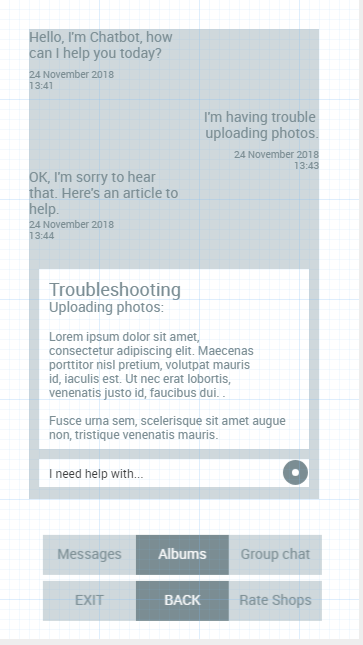
1. This is how the user would see what is in the albums. For example, this user selected the album “My Tattoos”.



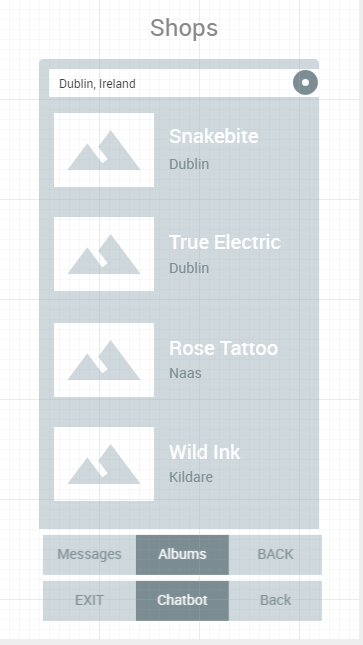
1. After selecting the Plus (+) button to add new media, this is what the user would see. After they select “Select file” and selected the file, a preview of it would show. After uploading, the System would bring them to the main Albums page, with the new media added.



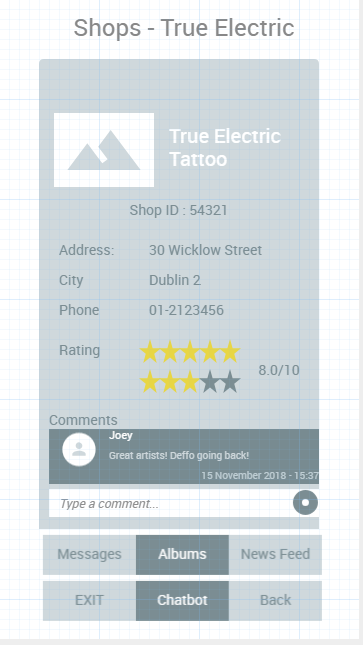
1. When the User selects “Group Chat”, the User would be shown this before the System logs the User into the external Group Chat Service.



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

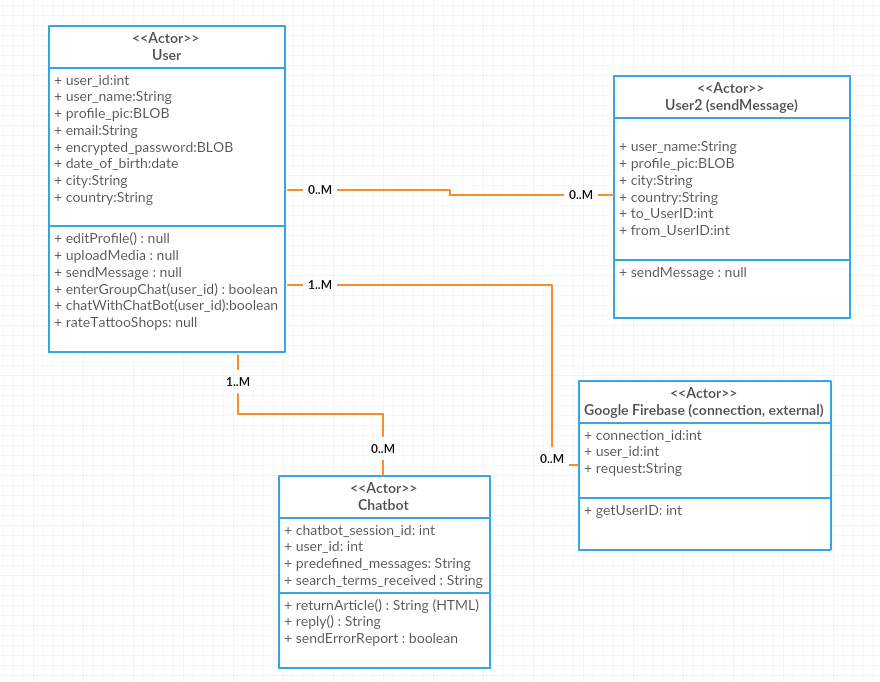


1. This shows how a user would search for a tattoo or piercing shop. This user has searched for “Dublin, Ireland” and got these results.

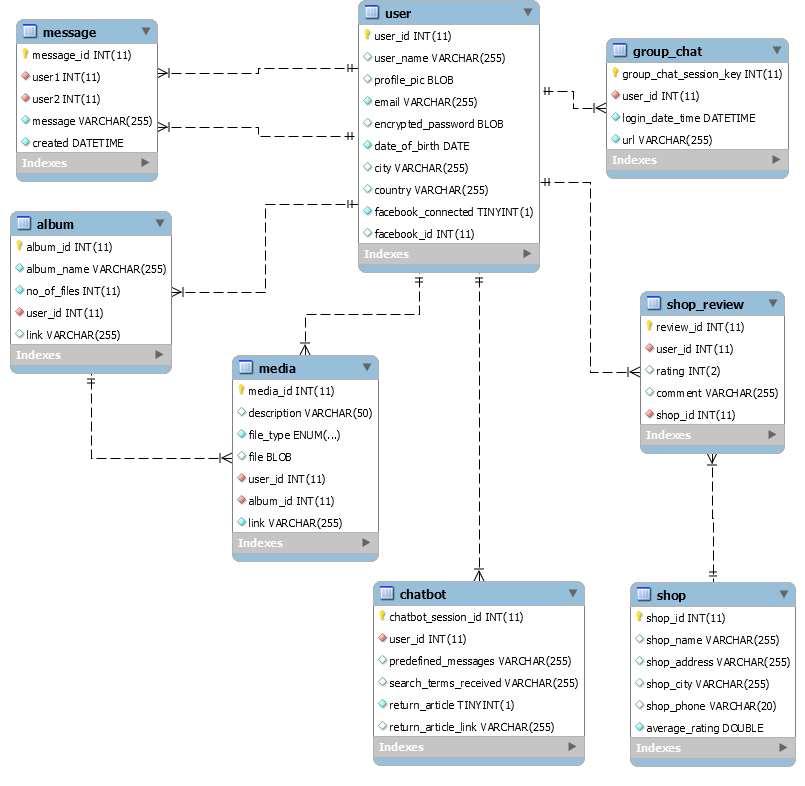


1. After selecting a shop, this is where the User can leave comments or a rating.

# System Architecture



# Google Firebase Database 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.