|  |
| --- |
| 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](#_Toc25064846)

[1.1 Purpose 4](#_Toc25064847)

[1.2 Project Scope 4](#_Toc25064848)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc25064849)

[2 User Requirements Definition 5](#_Toc25064850)

[2.1 Pages, profiles and bookings 5](#_Toc25064851)

[2.2 Messaging service 6](#_Toc25064852)

[2.3 Payments 6](#_Toc25064853)

[2.4 Automated appointments and stock control 6](#_Toc25064854)

[3 Current Process 6](#_Toc25064855)

[3.1 Personalisation 6](#_Toc25064856)

[3.2 Optimisation 7](#_Toc25064857)

[4 Requirements Specification 8](#_Toc25064858)

[4.1 Functional requirements 8](#_Toc25064859)

[4.1.1 Use Case Diagram 8](#_Toc25064860)

[4.1.2 Requirement 1: Create/Edit Profile 8](#_Toc25064861)

[4.1.3 Requirement 2: Create Artist page 11](#_Toc25064862)

[4.1.4 Requirement 3: Create Work Schedule 13](#_Toc25064863)

[4.1.5 Requirement 4: Book appointment 15](#_Toc25064864)

[4.1.6 Requirement 5: Rate Artist 18](#_Toc25064865)

[4.1.7 Requirement 6: Message user 21](#_Toc25064866)

[4.2 Non-Functional Requirements 24](#_Toc25064867)

[4.2.1 Performance/Response time requirement 24](#_Toc25064868)

[4.2.2 Availability requirement 24](#_Toc25064869)

[4.2.3 Recover requirement 24](#_Toc25064870)

[4.2.4 Security requirement 24](#_Toc25064871)

[5 Interface requirements / Application Programming Interfaces (API) 25](#_Toc25064872)

[5.1 GUI 25](#_Toc25064873)

[6 System Architecture 38](#_Toc25064874)

[7 Google MySQL database Architecture 39](#_Toc25064875)

[8 System Evolution 39](#_Toc25064876)

# 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 used in databases. |

# 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.

A bi-monthly stock assessment would be completed by the artist to ensure the actual stock amounts match with the system’s stock amounts.

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, provided they have any. 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 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.

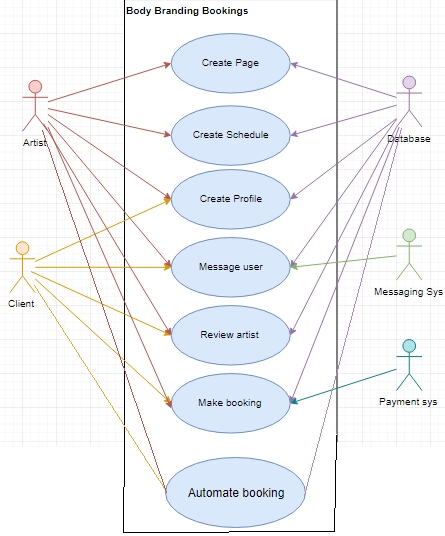
The same would also go for piercings, if a certain ring or stud is used on a regular basis, these could be automatically ordered. The likes of stock control for piercers could also be controlled here. This will be developed with the Requirement Specifications.

# 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

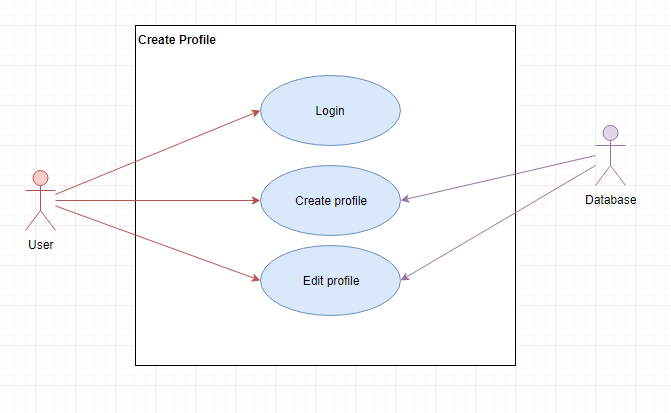
**Scope**

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

(Such details include but not limited to their username, date of birth, current city, a small bio, a photo, and so on)

1. The User selects “Continue”.
2. The System encrypts this information and put it into the MySQL database.
3. 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: Create Artist page

#### Description & Priority

This Use Case describes how an Artist create and edits their Artist Page.

#### Use Case

**Unique ID:** create Page

**Scope**

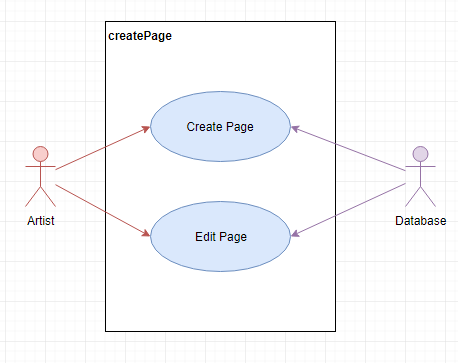
The scope of this use case is to allow a User to upload or delete media (videos and photos) to/from an album.

The scope of this use case is to enable an artist to create an artist page where they can promote their artist skills and promote their business.

**Description**

This use case describes how the artist creates and edits their artist’s page.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly
* The Artist is signed in
* The Artist has not previously created a Page.
* MySQL is connecting correctly.
* An Error log file is stored within the website’s files.

**Activation**

This use case begins when the artist selects “Create Artist Page” from their home menu.

**Main flow**

1. The Artist selects “Create Artist Page” <See A1>
2. The System brings the Artist to the Edit Artist Page settings.
3. The Artist enters the relevant details.

[Such details include but not limited to the shop name, address, type (tattoos, piercings, body mods, all), gallery, and so on]

1. The User selects “Continue”.
2. The System encrypts this information and put it into the MySQL database.
3. The System shows the User the home page.

**Alternate flow**

A1: <Edit Page>

1. The User selects “Edit Artist Page”

<Returns to number 2 in Main Flow>

1. The System loads the Artist’s Page information from the database.

<Returns to number 3 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>

**Termination**

This Use Case is terminated when the Artist has successfully created or edited their Artist Page.

**Post condition**

The System goes into a wait state

### Requirement 3: Create Work Schedule

#### Description & Priority

This Use Case describes how an Artist creates their work schedule, to indicate when they are available to work.

#### Use Case

**Unique ID:** Setched

**Scope**

The scope of this use case is for the Artist to create and edit their work schedule.

**Description**

This use case describes how the Artist can set up and edit their work schedule so Client’s will be able to book time slots when the Artist is available.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly
* The Artist is signed in
* The MySQL database and connection are actively working correctly.
* Appointments (Artist) and bookings (Client) are displayed on a calendar on the website, but stored in the MySQL database.
* An Error log file is stored within the website’s files.

**Activation**

This use case starts when the selects “Appointments” on their home menu.

**Main flow**

1. The Artist selects “Appointments” from their home menu.
2. The System checks if there is a calendar created for this user. <See A1>
3. The Artist selects the days and times they are available to work for the working month.

(Optional: The Artist selects “Repeat”, so the newly created calendar is repeated every month, with no appointments on the calendar)

1. The Artist sets the size, location, estimated time it will take and price (e.g. of a tattoo). This will be used as a guide for the Client to book their appointments.
2. The Artist selects “Save”
3. The System connects to the MySQL database
4. The System inserts the data in the database.
5. The System shows the Artist their home menu.

**Alternate flow**

A1: <Artist has previously created a calendar>

1. The System connects to the MySQL database
2. The System loads the Client’s bookings on the Calendar (if any)
3. The System retrieves the Artist’s calendar data and shows it on screen.

<returns to number 3 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>

**Termination**

This Use Case is terminated when the Artist has successfully created or edited their calendar.

**Post condition**

The System goes into a wait state

### Requirement 4: Book appointment

This Use Case describes how a Client books an appointment

#### Use Case

**Unique ID:** createBooking

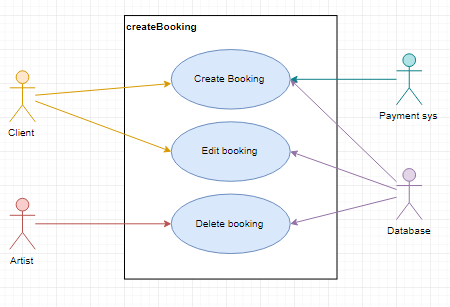
**Scope**

The scope of this use case is for a Client to book and edit an appointment, and for a user (Artist or Client) to delete an appointment.

**Description**

This use case describes how the Client can book and edit their appointments and for either an artist or client to delete an appointment. The Client can only delete their own appointments while the Artist can delete any appointments on their calendar.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly,
* The Client is signed in
* The Artist is signed in
* The MySQL database and connection are actively working correctly.
* Appointments (Artist) and bookings (Client) are displayed on a calendar on the website, but stored in the MySQL database.
* The Payments System is connecting and working correctly.
* An Error log file is stored within the website’s files.

**Activation**

This use case starts when the Client selects “Bookings” on their home menu.

**Main flow**

1. The Client selects “Bookings” from their home menu.
2. The System loads the Bookings page.
3. The Client selects “Add new booking”. <See A1>
4. The System loads the Edit Booking page.
5. The Client inserts the relevant information (size and location of (e.g.) tattoo)
6. The Client selects the date and start time of the booking.
7. The Client uploads reference photos
8. The System shows the price.
9. The Client selects “Continue”
10. The System send the Client to the Payment System.

(External: Payment System put transaction on hold.)

1. The System checks if there are any overlapping appointments. <See A2, E1>
2. The System sets the booking to “unconfirmed”.
3. The System generates a booking reference number and adds it to the appointment.
4. The System inserts the data into the MySQL database.
5. The System loads the Booking Saved page, notifying the Client the booking is saved.
6. The Client returns to the home menu.
7. The System notifies the Artist of a new appointment.
8. The Artist selects “Confirm”. <See A3, E2>
9. The System notifies the Payment System to process payment.
10. The System sets the booking to “confirmed”.
11. The System notifies the Client that their booking is confirmed.

**Alternate flow**

A1: <Edit appointment>

1. The Client selects a booking
2. The Client selects “Edit Booking”.
3. The System retrieves the booking data from the MySQL database.

<returns to number 5 in Main Flow>

A2: <Overlapping appointments>

1. The System checks for the next available timeslot with the required amount of time.
2. The System highlights the date and time
3. The System displays a message saying the Artist is booked, with the suggested new date and time.

<returns to number 6 in Main Flow>

A3: <Artist declines>

1. The Artist selects “Deny”
2. The appointment is deleted from the database.
3. The System notifies the Payment System not to process payment.
4. The System notifies the Client to create another booking.

<Goes to number 10 in A1>

**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 8 in Main Flow>

E2: <Artist doesn’t select either “Confirm” or “Deny”>

1. The System waits 48 hours, or 24 hours before the appointment, whichever is nearer.
2. The System notifies the Client to contact the Artist directly.

<Goes to number 17 in A2>

**Termination**

This Use Case is terminated when the Client has successfully placed a booking or has edited a booking, or a user has deleted the booking.

**Post condition**

The System goes into a wait state

### Requirement 5: Rate Artist

#### Description & Priority

This Use Case describes how a Client can rate an artist and leave a comment on the Artist’s Page.

#### Use Case

**Unique ID:** rate Artist

**Scope**

The scope of this use case is to show how a Client can rate an Artist.

**Description**

This use case describes how the Client can view and rate the Artist via the Artist’s Page.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly,
* The Client is signed in
* The Artist is signed in
* The Artist has previously created an Artist’s Page.
* The MySQL database and connection are actively working correctly.
* An Error log file is stored within the website’s files.
* Users cannot edit or delete ratings.
* Artists need to approve comments before they’re publicly visible.
* If an Artist does not approve a comment, it is deleted.

**Activation**

This use case starts when the Client selects “Rate Artist”.

**Main flow**

1. The Client selects “Rate Artist” from their home menu.
2. The System retrieves and lists all the Artists that the Client booked with in the past. <See E1>
3. The Client selects one of the Artists.
4. The System loads the Artist’s Page.
5. The System calculates the average star ratings and displays it.
6. The System loads the comments that are marked “checked”.
7. The Client selects a star-rating out of 5 stars. <See A1>
8. The Client enters a new review (comment).
9. The Client selects “Post”.
10. The System puts the rating into the “Ratings” array.
11. The System puts the comment into the “Comments” array and marks it unchecked.
12. The System notifies the Client that their comment is pending.
13. The System reloads the page and repeats numbers 4 – 6.
14. The Client returns to the home menu.
15. The system notifies the artist that there is a new comment on their Artist’s Page.
16. The Artist selects “Approve”. <See A2, E2>
17. The System changes “unchecked” to “checked” on the comment.

**Alternate flow**

A1: <Client is editing comment>

1. The System shows an Edit button beside the Client’s comment.
2. The Client selects “Edit”.
3. The Client edits their comment.

<returns to number 9 in main flow>

A3: <Artist declines>

1. The Artist selects “Deny”
2. The comment is deleted from the database.

**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.

E2: <Artist doesn’t select either “Confirm” or “Deny”>

1. The System waits 24 hours
2. The System notifies the Artist of a new comment on their Artist’s Page.
3. The System waits another 24 hours.
4. The System deletes the comment. (No reply from Artist)

**Termination**

This Use Case is terminated when the Client has successfully rated the Artist.

**Post condition**

The System goes into a wait state

### Requirement 6: Message user

#### Description & Priority

This Use Case describes how an Artist can message a Client, and how a Client can message an Artist.

#### Use Case

**Unique ID:** messageUser

**Scope**

The scope of this use case is to show how a User message another User.

**Description**

The scope of this use case is to show how a User message another User. Artists can message Clients and Clients can message Artists.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly,
* The Client is signed in
* The Artist is signed in.
* The MySQL database is active and working correctly.
* The Messaging System is working correctly.
* An Error log file is stored within the website’s files.
* Users cannot edit or delete ratings.
* Artists need to approve comments before they’re publicly visible.
* If an Artist does not approve a comment, it is deleted.

**Activation**

This use case starts when the selects “Messages” from their home menu.

**Main flow**

<Artist messages Client>

1. The Artist selects “Messaging” from their home menu. <See A1>
2. The System displays client names that have had an appointment in the last month or have an appointment in the next month.
3. The Artist selects one.
4. The System records the user ID, the type of user (whether artist or client.
5. The System generates a messaging ID.
6. The System send the user ID, type and messaging ID to the MySQL database. <See E1>
7. The System sends the user ID, type and messaging ID to the Messaging System. <See E2>
8. The System communicates with the Messaging System and displays messages as they are sent and received.
9. The Artist sends messages as required (external)
10. The Artist closes the messaging dialog box.
11. The System requests to stop the messaging session to the Messaging System.
12. The System loads the Messages page.

**Alternate flow**

A1: <Client messages Artist>

1. The Client selects “Messaging” from their home menu.
2. The System displays Artist names that they have or had booking with
3. The Client selects one.
4. The System records the user ID, the type of user (whether Client or client.
5. The System generates a messaging ID.
6. The System send the user ID, type and messaging ID to the MySQL database. <See E1>
7. The System sends the user ID, type and messaging ID to the Messaging System. <See E2>
8. The System communicates with the Messaging System and displays messages as they are sent and received.
9. The Client sends messages as required (external)
10. The Client closes the messaging dialog box.
11. The System requests to stop the messaging session to the Messaging System.
12. The System loads the Messages page.

**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.

E2: System cannot connect to the Messaging System

1. The System cannot connect to the Messaging System
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.

**Termination**

This Use Case is terminated when the User has successfully sent and received messages.

**Post condition**

The System goes into a wait state

### Requirement 6: Create automated appointment

#### Description & Priority

This use case describes how the System creates automatic appointments for the User.

#### Use Case

**Unique ID:** autoBook

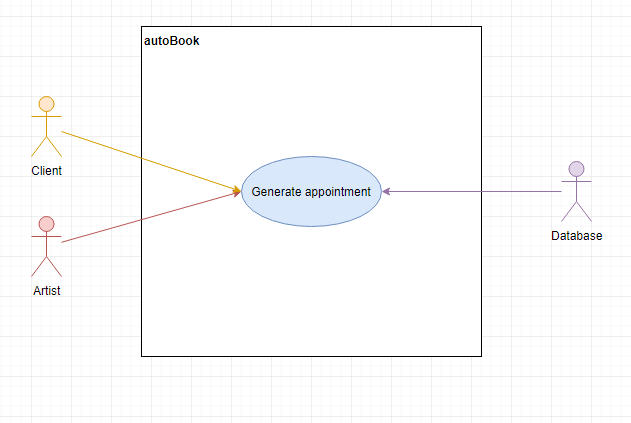
**Scope**

The scope of this use case is to automate a new booking for the Client.

**Description**

The scope of this use case is to automate bookings for the Client. Depending on the number of previous appointments (i.e. more than 4), the system will generate an appointment to be confirmed by the Client and the Artist.

**Use Case Diagram**

****

**Flow Description**

**Precondition**

* The System is active and working correctly,
* The Client is signed in
* The Artist is signed in.
* The MySQL database is active and working correctly.
* An Error log file is stored within the website’s files.
* The Artist and Client need to approve appointment before it’s confirmed
* The Artist and Client are able to turn on this feature

**Activation**

This use case starts every morning at 1am.

**Main flow**

1. The System checks the completed appointment schedule for the day and locates the first appointment.<See E1>
2. The System checks the Client has not opted out for automated appointments. <See A1>
3. The System checks the Artist has not opted out. <See A1>
4. The System adds an appointment for the Client for the Monday 4 weeks away at the same time.
5. The System sets the appointment to unconfirmed.
6. The System checks the Artist’s schedule. <See A2>
7. The System notifies the Client
8. The Client confirms <See A3>
9. The System notifies the Artist.
10. The Artist confirms <See A3>
11. The System sets the appointment to confirmed.
12. The System goes to the next completed appointment that day.
13. The System repeats numbers 2 – 12 until there are no more completed appointments for that day

**Alternate flow**

A1: The Artist and/or Client have opted out of automated appointments

<returns to number 12 in Main Flow>

A2: Artist’s Schedule already has appointment booked for that date and time>

1. The System finds the next available date and time that is not booked

<returns to number 7 in Main Flow>

A3: The Artist and/or Client select deny or do not respond after 24 hours.

<returns to number 12 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.

**Termination**

This Use Case is terminated when the System successfully books new appointments for the Clients and Artists

**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 MySQL or the Messaging Service, a response to the User must be presented in 6 seconds of less.

### Availability requirement

3B must be available 24/7. To ensure this is the case, 2 back-up connections to MySQL will be developed.

### Recover requirement

At least 2 backups in separate physical locations must be available. If MySQL or the Messaging Service is unable to connect to the System, the system will notify the developer at once in the form of an email.

### Security requirement

No plain text passwords are stored anywhere. Encrypted passwords using AES will be stored in the MySQL database.

Session keys and cookies will be used.

# Interface requirements / Application Programming Interfaces (API)

MySQL Connection / API

An example of how tables in the MySQL database would be used is the Users tables. Tables related to the Users in the MySQL database are Bookings, Appointments and Comments.

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.

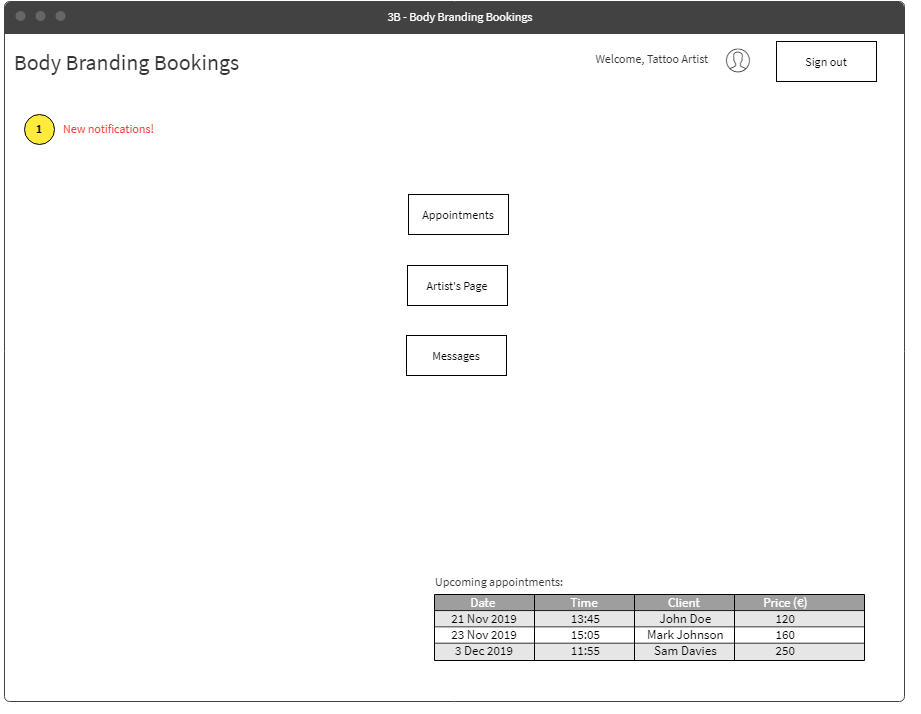
Similarly, connection to the Messaging System will be using a similar API.

The API will be created by the developer.

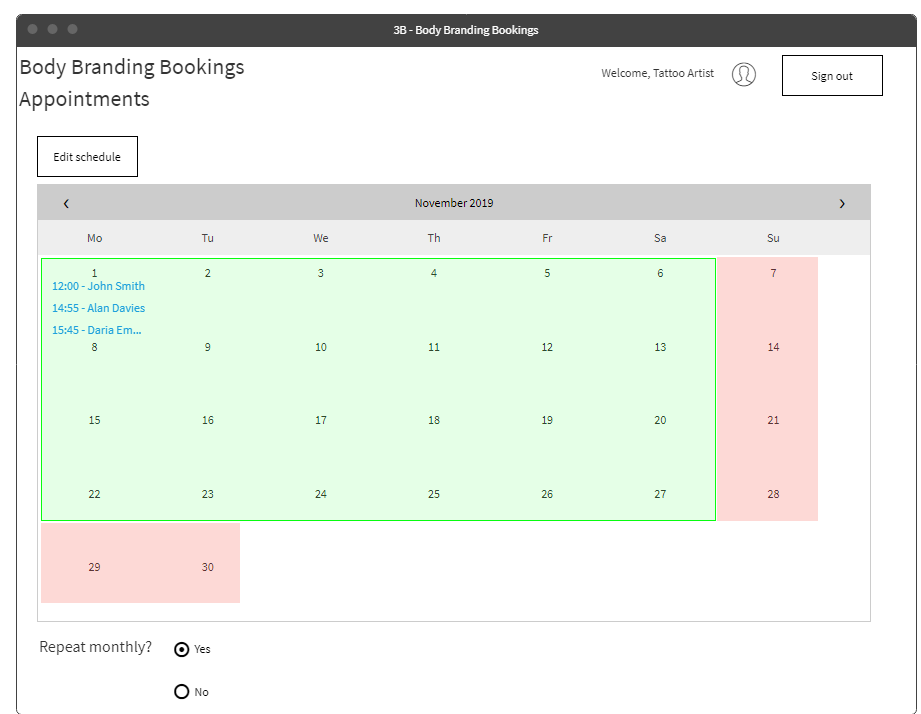
## GUI

These mock-ups show what 3B will look like

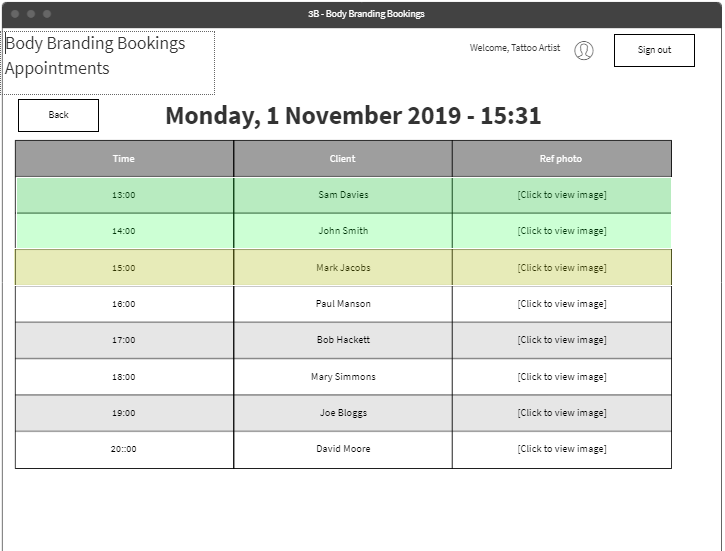
1. This is the main Home Menu for the Artist. It is similar for the Client.



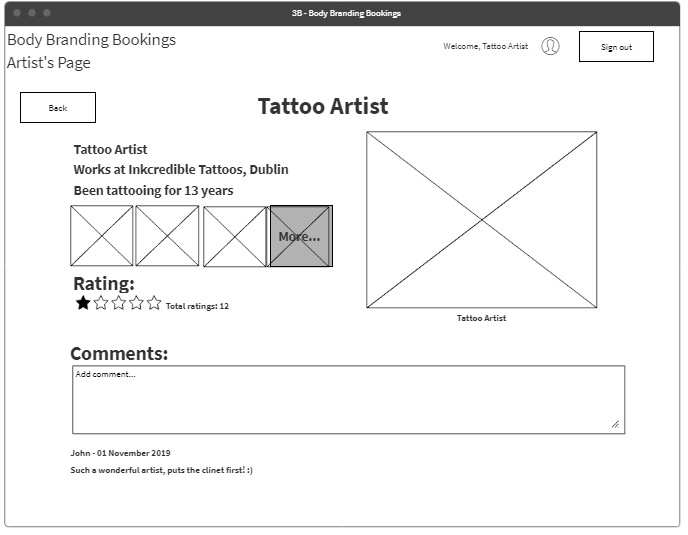
1. The calendar for the Artist to select their work schedule. This example shows how appointments would look on a month-view. The green indicates when the Artist is working and the red is when the Artist is not working.



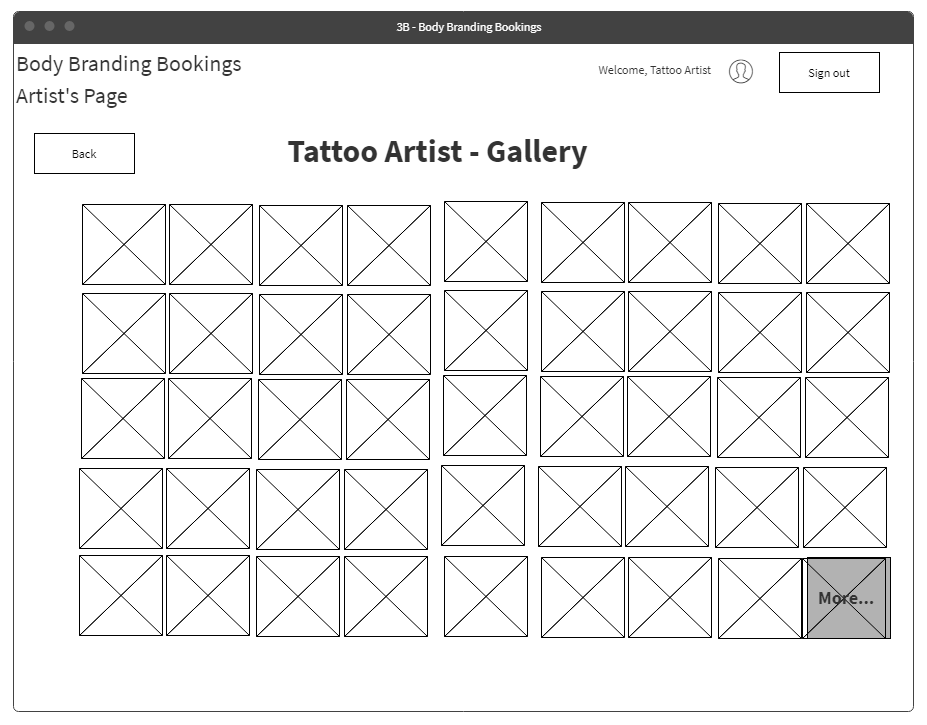
1. An example of the Artist’s calendar showing the appointments for the day. The green shows the appointments that are completed and the yellow shows appointments in progress.



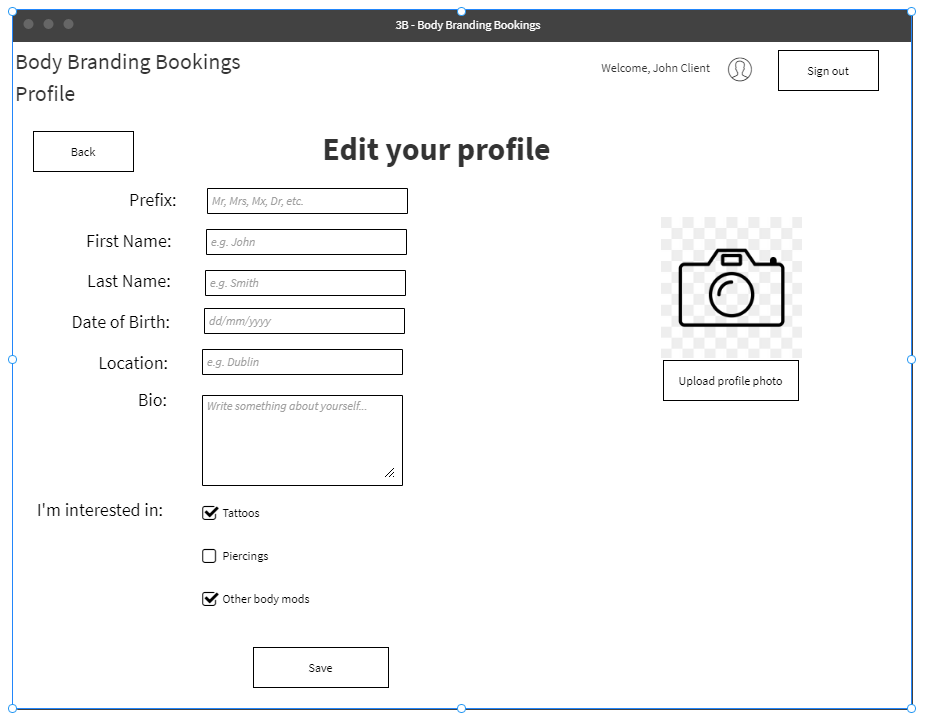
1. This is the Artist page, the “More…” link goes to the Artist’s Gallery.



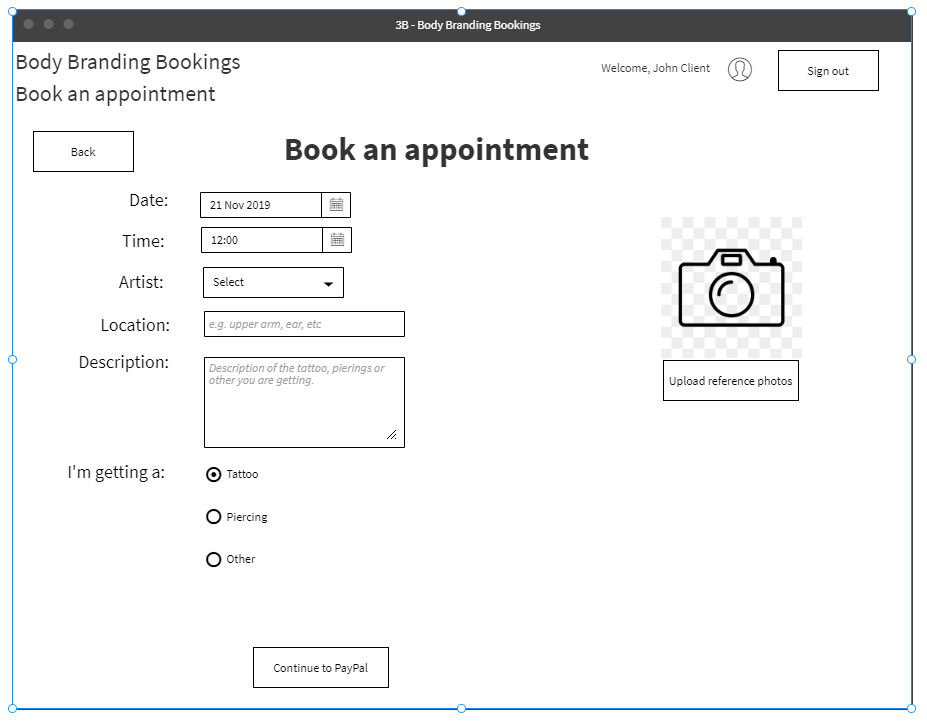
1. This is the Artist’s Gallery, where a Client can view the Artist’s work.



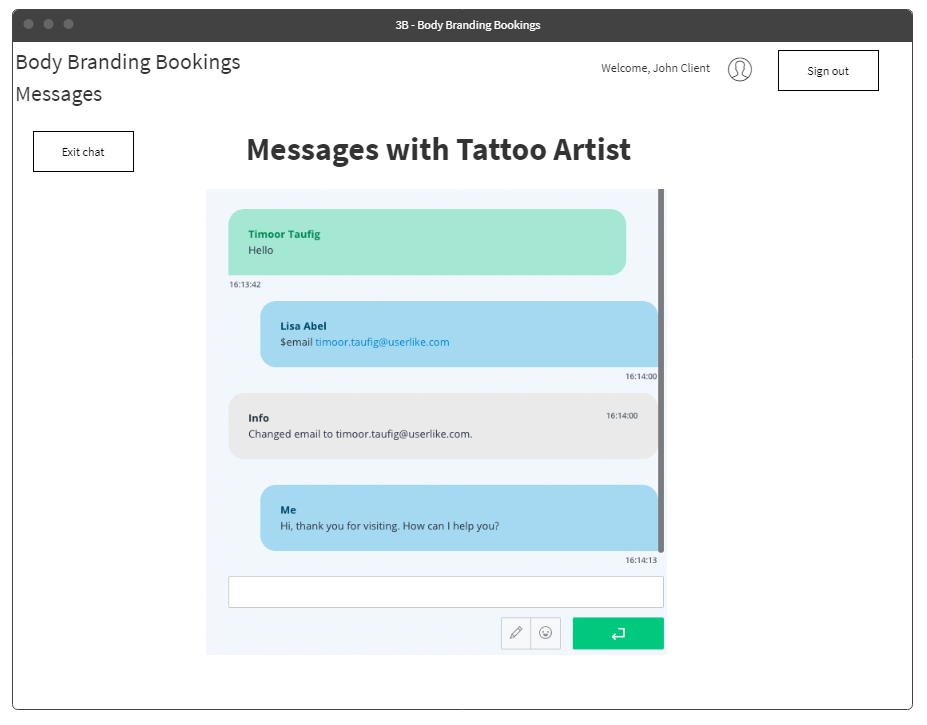
1. This is the Edit Profile page. This is the Client’s view. The Artist’s view has some additional features like where they work and the length of time they’ve been doing their profession.



1. This is how a Client can book an appointment. The Artist field is a drop-down menu of all the Artists listed on the site. The Client can also start typing to select the Artist quicker. Payment for deposit is managed by an external payment system.



1. This is an example of a conversation dialog in the Messages section



# System Architecture

TO BE COMPLETED

# System Evolution