**Setup Document**

ProP Group 3 – Fyre Music Festival (FMF)

Contents

[Clients Request 2](#_Toc23704762)

[Main requirements: 2](#_Toc23704763)

[Client’s requests: 2](#_Toc23704764)

[Processes 3](#_Toc23704765)

[Use Case 3](#_Toc23704766)

[Use Case Diagram 9](#_Toc23704767)

[Website 9](#_Toc23704768)

[Festival 9](#_Toc23704769)

[Functional Requirements 10](#_Toc23704770)

[Table 10](#_Toc23704771)

[GUI 12](#_Toc23704772)

[Status Overview 12](#_Toc23704773)

[Event Check-in Check-out 13](#_Toc23704774)

[Camp Check-in Check-out 13](#_Toc23704775)

[Employee Check-in Check-out 14](#_Toc23704777)

[Shops 14](#_Toc23704778)

[Loan 15](#_Toc23704779)

[ATM App 15](#_Toc23704780)

[Log Transactions 16](#_Toc23704781)

[End-Of-Event 17](#_Toc23704782)

[Wireframe 18](#_Toc23704783)

[Main page 18](#_Toc23704784)

[Tickets page 19](#_Toc23704785)

[Artist page 20](#_Toc23704786)

[Info page 21](#_Toc23704787)

[Profile page 22](#_Toc23704788)

[Sign-up page 23](#_Toc23704789)

[ERD 24](#_Toc23704790)

# Clients Request

## Main requirements:

* A Website
* Application to check-in and check-out
* Application that supports shops and loan stands
* Application to be able to view status overview
* A single database systems
* Application that changes the balance of the visitors based on logfiles

## Client’s requests:

* Website have a sign-up page to purchase tickets
* Website a profile/user page for personal information
* More than one type of tickets
* Indefinite camping area/size
* Tickets and camping reservations are refundable
* ATM transactions must be tracked using logfiles
* There should be an application(s) for borrowing item(s)
* Separate app for camp and employee check-in and check-out

# Processes

## Use Case

**User Case 1**: Sign-up

**Actor(s):**

* Visitor

**Main Success Scenario:**

1. Click the sign-up button
2. Enter the required details
3. Click sign-up
4. Bring the user to the profile page to notify that an account is made

**Extension(s):**

3a. The sign-up button will only be unlocked when all required data are filled

**User Case 2:** Buying tickets

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. Visitor must already have an account
2. Visitor must login foremost

**Main Success Scenario:**

1. Go to the tickets page by clicking the tickets button
2. In tickets page, select what type of ticket to buy by selecting the corresponding choices
3. Select a camping spot by clicking on the drop-down box
4. Ticket number will be sent to user’s email and will be in his/her profile page as well

**Extension(s):**

3a. Buying camping spots is not mandatory, so it is possible to only purchase tickets

**User Case 3:** Viewing artists information

**Actor(s):**

* Visitor

**Main Success Scenario:**

1. Go to the artists page by clicking the artists button
2. In artists page, select which artists you want to view
3. After clicking the selected artists, a new window will pop to show the artist’s information
4. Clicking outside the window will take the user back to the artists page
5. Once returned to the artists page, user may select another artists to view

**User Case 4:** Viewing festival information

**Actor(s):**

* Visitor

**Main Success Scenario:**

1. Go to the info page by clicking the info button
2. Information of the event are in the page

**User Case 5:** Viewing profile page

**Actor(s):**

* Visitor

**Pre-Condition(s):**

* Visitor must login foremost

**Main Success Scenario:**

1. Go to profile page by clicking the profile button
2. Information about the users are in the profile page

**User Case 6:** Changing user’s email

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. Visitor must login foremost

**Main Success Scenario:**

1. Go to the profile page by clicking the profile button
2. Click the change button below the email address
3. User will be taken to a new window
4. User will be asked to enter new email address detail
5. Click confirm to confirm the changes
6. User have successfully change his/her email address

**Extension(s):**

5a. Confirm button will only unlock if the required entry is filled

**User Case 7:** Cancelling tickets purchase

**Actor(s):**

* Visitor

**Pre-Condition(s):**

* Visitor must have bought a ticket
* Visitor must login foremost

**Main Success Scenario:**

1. Go to the profile page by clicking the profile button
2. Click the cancel button beside the ticket
3. A pop-up will show to confirm cancellation
4. If user wants to cancel, click the confirm button in the pop-up, if not, click the cancel button
5. User will be taken back to the profile page with empty ticket number

**Extension(s):**

4a. If ticket is cancelled, refund will take place after the event is finished

**User Case 8**: Login into user’s account

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. User must already have an account

**Main Success Scenario:**

1. On the main page, click on the login button
2. Enter username and password, then click login

**User Case 9:** Cancelling camp reservation

**Actor(s):**

* Visitor

**Pre-Condition(s):**

* Visitor must have reserved a camp spot
* Visitor must login foremost

**Main Success Scenario:**

1. Go to the profile page by clicking the profile button
2. Click the cancel button besides camping
3. A pop-up will show to confirm cancellation
4. If user wants to cancel, click the confirm button in the pop-up, if not, click the cancel button
5. User will be taken back to the profile page with empty camping spot
6. User will be allowed to reserve a camping spot again

**Extension(s):**

4a. If camping spot is cancelled, refund will take place after the event is finished

**User Case 10**: Logout from the website

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. User must already have an account
2. Visitor must login foremost

**Main Success Scenario:**

1. On The main page, click on the sign-out button
2. After clicking, user will be taken back to a refreshed main page

**User Case 11:** Check-in to the festival

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. Visitor must have a ticket

**Main Success Scenario:**

1. Each visitor is given an RFID-tag
2. Visitor scan the RFID-tag at the entrance
3. Visitor enters the festival

**Post-Condition(s):**

1. Visitor must keep his/her RFID-tag at all times

**User Case 11:** Check-out of the festival

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. Visitor must have a ticket

**Main Success Scenario:**

1. Each visitor has an RFID-tag
2. Visitor scan the RFID-tag at the entrance
3. Visitor exits the festival

**Post-Condition(s):**

1. Visitor must not have any borrowed/loan items in his/her possession

**User Case 12:** Buying products

**Actor(s):**

* Visitor
* Employee

**Pre-Condition(s):**

* Visitor must be inside the festival

**Main Success Scenario:**

1. Visitor goes to a stand
2. Visitor tells the employee what he/she wants to buy
3. Employee uses the app to place the order
4. Visitor pays for the product
5. Visitor gets the product

**User Case 13:** Borrowing items

**Actor(s):**

* Visitor
* Employee

**Pre-Condition(s):**

* Visitor must be inside the festival
* Visitor must have his/her RFID-tag

**Pre-Condition(s):**

* Visitor must be inside the festival
* Visitor must have his/her RFID-tag

**Main Success Scenario:**

1. Visitor visits a loan stand in the festival
2. Visitor ask the employee which item he wants to borrow
3. Employee uses the app to scan the requested item
4. Visitor scan his/her RFID-tag
5. Employee lends the requested item

**User Case 14:** Returning borrowed items

**Actor(s):**

* Visitor
* Employee

**Pre-Condition(s):**

1. Visitor must be inside the festival
2. Visitor must have an borrowed Item

**Main Success Scenario:**

1. Visitor goes to a loan stand
2. Visitor gives the borrowed item to the employee
3. Employee scans the borrowed item to ensure that it is returned to the inventory

**User Case 15:** Tracking transactions

**Actor(s):**

* Employee

**Main Success Scenario:**

1. An app that shows the ID number of the visitors
2. Employee press the ID number to see its transactions logfiles
3. Logfiles shows the type of transactions of the selected visitor

**User Case 16:** ATM app

**Actor(s):**

* Visitor

**Pre-Condition(s):**

1. Visitor must have an RFID-tag

**Main Success Scenario:**

1. Visitor goes to the ATM booth within the festival
2. Visitor scan his/her RFID-tag to open his/her account
3. Visitor specify the amount of money that he/she wants to add/withdraw
4. Visitor closes the connection

## Use Case Diagram

### Website

### Festival

# Functional Requirements

**What Should the system do?**

The main reason that this software is being made is the fact that the client wants to be able to manage this current event as easily as possible. The website should allow visitors/user to sign up, buy tickets as well as reserve camping spots. Furthermore it should allow the visitor to check the artists that going to perform in the event. After signing up, the user should be able to check his/her profile in the profile page, and in that very page, users should able to change his/her email.

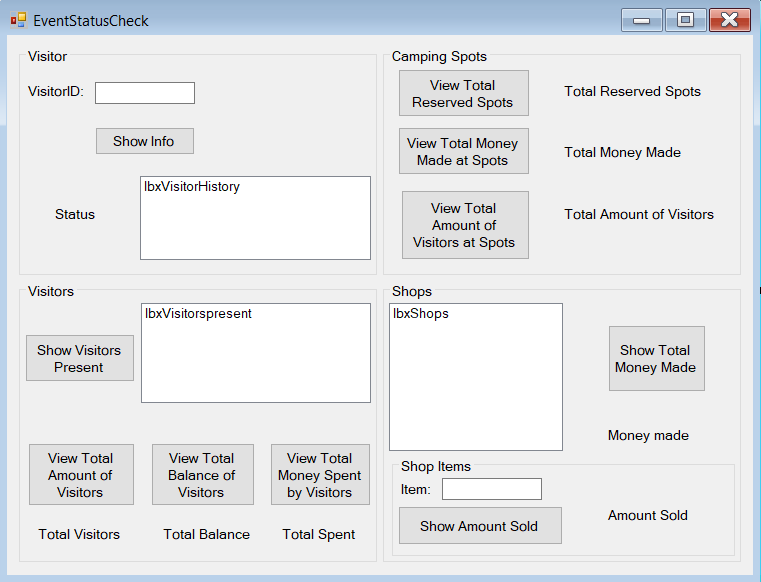
As for the software, we will create separate application for every process, hence each software will be independent from one another however all data is managed by the database for the application.

## Table

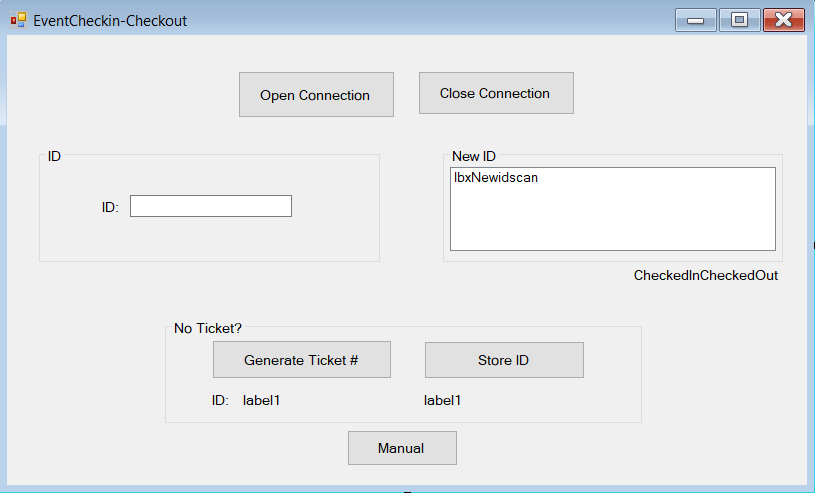
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliverable | Must Have | Should Have | Could Have | Won’t Have |
| *Festival’s Website:* |  | | | |
| Main page consists of festival info and artists page | ✓ |  |  |  |
| Reserving tickets and camp are in tickets page | ✓ |  |  |  |
| Complete artists information in the artists page | ✓ |  |  |  |
| Info page will consist of schedule and map of the festival and contacts information | ✓ |  |  |  |
| User can sign-in and sign-out | ✓ |  |  |  |
| Profile page consists of user’s information and data | ✓ |  |  |  |
| User can cancel their ticket and/or camp reservation in profile page | ✓ |  |  |  |
| User can change their email address in profile page | ✓ |  |  |  |
| *Check-in and Check-out app:* |  | | | |
| User can check-in and check-out of the festival | ✓ |  |  |  |
| Track the number of visitors currently in the festival |  | ✓ |  |  |
| Separate app for camp check-in and check-out |  | ✓ |  |  |
| Separate app for Employee check-in and check-out |  | ✓ |  |  |
| Check-out is not possible if a user is borrowing an item |  | ✓ |  |  |
| *App to buy items:* |  |  |  |  |
| User can buy items in the festival | ✓ |  |  |  |
| Each type of items have their specific ID | ✓ |  |  |  |
| Track how many items are bought in each shop | ✓ |  |  |  |
| *App to loan items:* |  |  |  |  |
| User will be able to borrow item during the festival by scanning the Item | ✓ |  |  |  |
| Each item has their own unique ID | ✓ |  |  |  |
| Track how many items are being loaned |  | ✓ |  |  |
| *App to track transactions:* |  |  |  |  |
| Track all transactions occurred during the festival | ✓ |  |  |  |
| All transactions are in a form of logfiles showed in the app | ✓ |  |  |  |
| Logfiles are stored in the database | ✓ |  |  |  |
| Able to select the transactions of each users | ✓ |  |  |  |
| *Status overview app:* |  |  |  |  |
| Track how many people are in the festival | ✓ |  |  |  |
| Track how many people in the camp site | ✓ |  |  |  |
| Track visitors status and history | ✓ |  |  |  |
| Track total visitors | ✓ |  |  |  |
| Track how many employees are in the festival | ✓ |  |  |  |
| Track borrowed items | ✓ |  |  |  |
| Track transactions | ✓ |  |  |  |
| Track total money sold per shop | ✓ |  |  |  |
| Track total amount of a certain product are sold | ✓ |  |  |  |
| Track total balance of all visitors | ✓ |  |  |  |
| Track total money spent | ✓ |  |  |  |
| Track free and booked camping spots | ✓ |  |  |  |
| *Database supporting all apps* | ✓ |  |  |  |
| *ATM app:* |  |  |  |  |
| ATM is available at both website and on site |  | ✓ |  |  |
| Able to top up money by scanning users QR code | ✓ |  |  |  |
| *Tickets for the festival:* |  |  |  |  |
| More than one type of tickets | ✓ |  |  |  |
| Each tickets produces its own unique ID-numbers (static numbers) | ✓ |  |  |  |
| One type of tickets |  |  |  | ✓ |
| *App to borrow extra equipment* | ✓ |  |  |  |
| *Mobile friendly website* |  |  | ✓ |  |
| *Airplane tickets* |  |  | ✓ |  |
| *Camp area size are definite* |  |  | ✓ |  |
| *Camp location in first page* |  |  |  | ✓ |

# GUI

## Status Overview



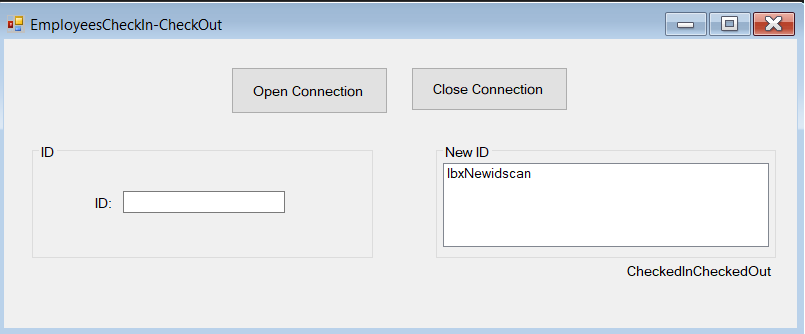
## Event Check-in Check-out



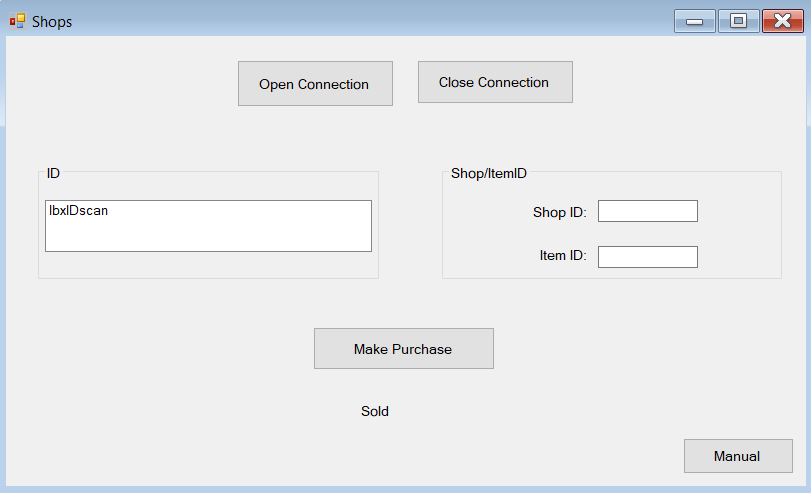
## Camp Check-in Check-out

## 

## Employee Check-in Check-out

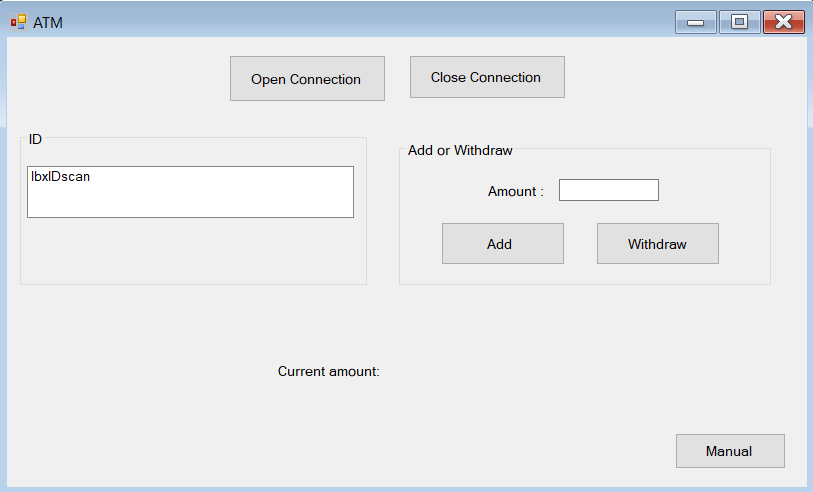


## Shops

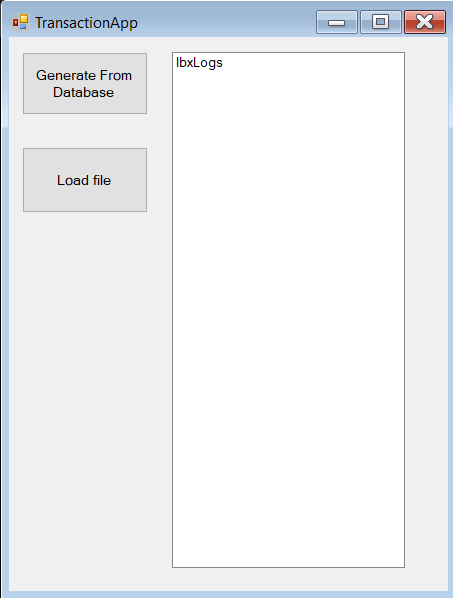


## Loan

## ATM App



## Log Transactions

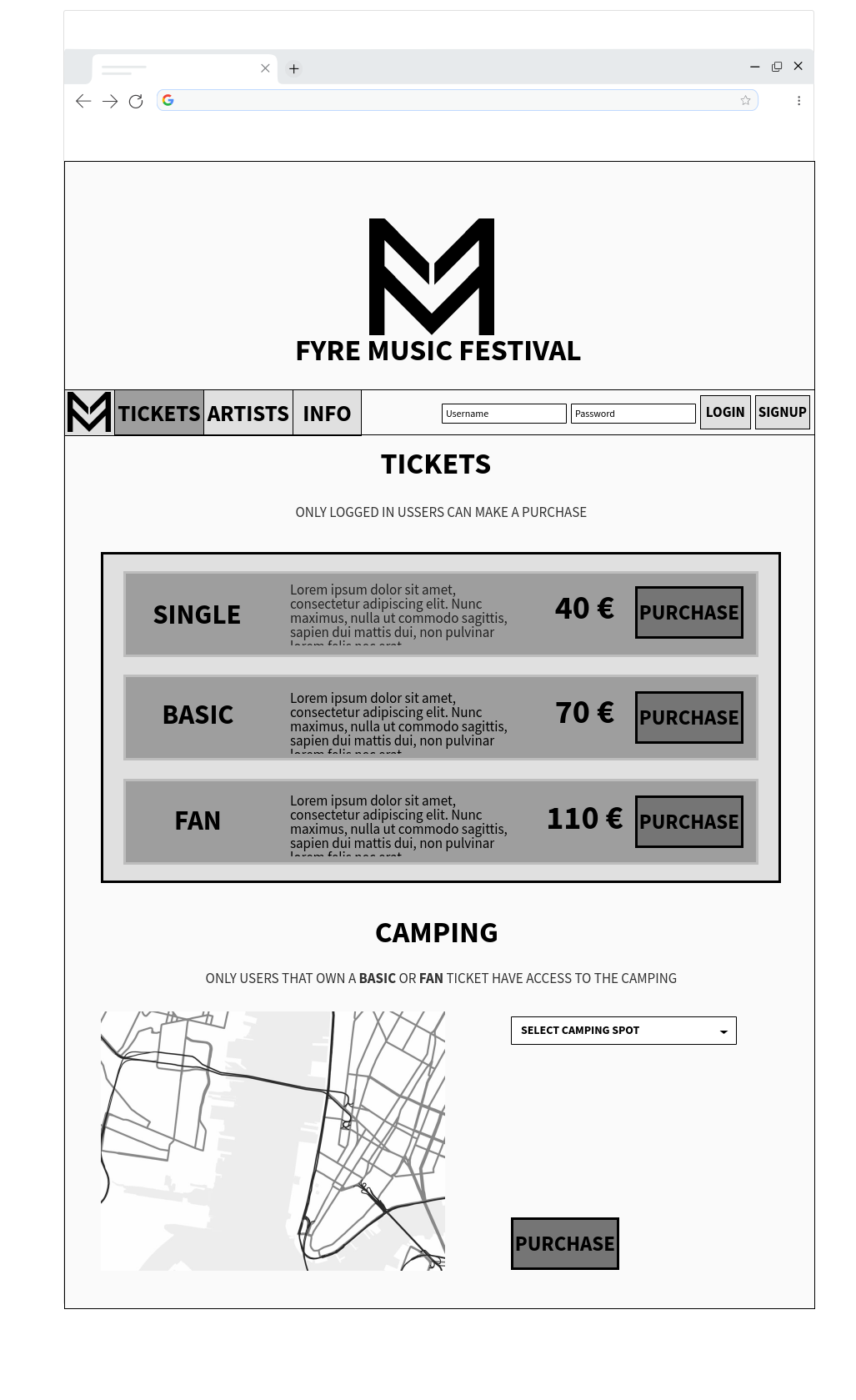


## End-Of-Event

# Wireframe

## Main page

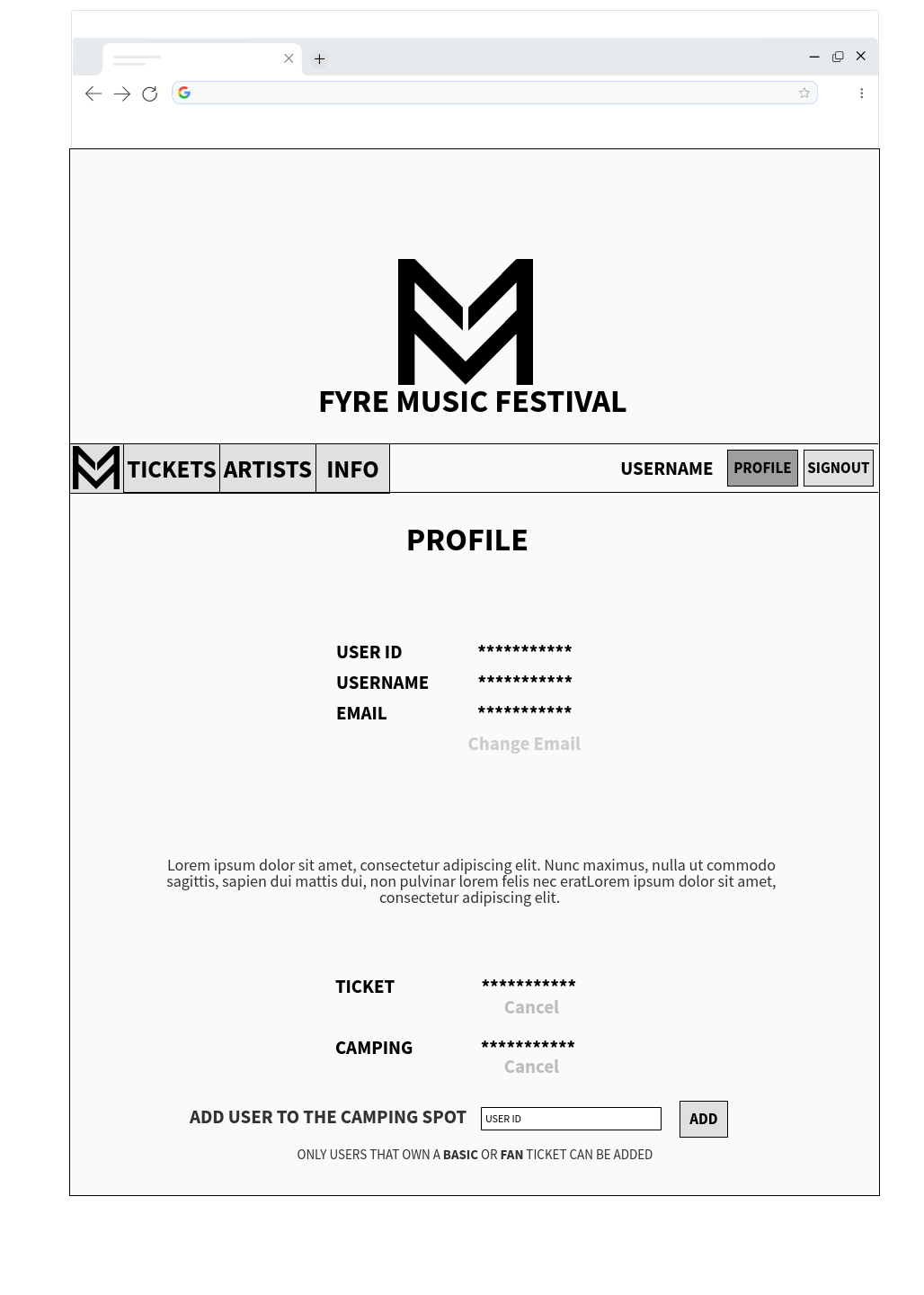
## Tickets page



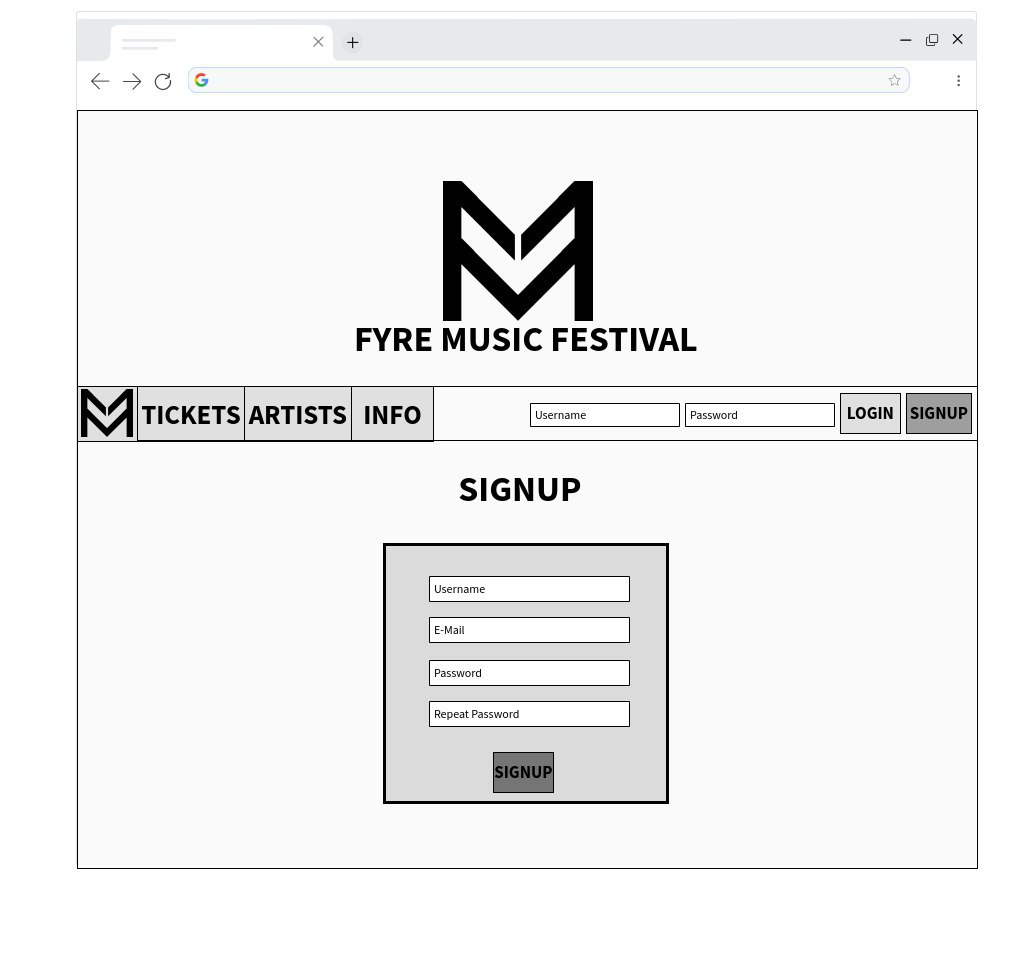
## Artist page

## Info page

## Profile page



## Sign-up page



# ERD

Below is the ERD used for this project. As of now, it has thirteen entities, with Employee/Visitor being subtypes of User and Shop/Loan subtypes of Transactions.

