# Setup Document

v. 0.4

Date: 10.10.2018

Presented by:

Galin Karafiziev

Viktoria Peeva

Artem Baygot

Hristo Slavchev

**Table of content:**

**1. Processes3**

**2. Functional requirements11**

**3. GUI and Wireframe12**

**4. ERD diagram15**

# Processes

Description of the event:

The client mr. Frank de Lepper want to hold the gaming festival about League of Legends. This project plan will provide an information about risks the client faces , the software solutions our team offers and the period of time to complete the project.

The gaming festival itself will consist of several small tournaments about the game “League of Legends” and one big tournament which will be played by the famous teams 9cloud and Fnatic. The festival will be held three days and the winners and visitors will have the chance to rest after each day at our camping spots. There will be stands with snacks and drinks around also there are going to be many stands in which consumers can borrow chargers or stands where the person can withdraw money (atm machines). There will be also merchandise but that will be available only through our applications and website.

Use Case ID: 1, 2, 3

Goal: To book a ticket via the website and get an account for the website/event application

Actors: User, System

Pre-condition: The user must have the website opened on the page “book a ticket”.

MSS:

1. The user clicks a button to buy a ticket.
2. The user fills in a form for registration, which includes first name, last name and email address.
3. The system accepts the form and checks whether the provided data is valid and it is.
4. The system checks if the user is already registered and it is not.
5. System then processes the form and displays a new tab with methods of payments.
6. User picks one payment method.
7. System shows a form for the payment.
8. User fills in the form.
9. System checks if there are any mistakes in the format of the input information and it is.
10. System checks if the provided information is valid and it is.
11. System sends request about a shipping to the database.
12. System sends an email about confirmation about the purchase of the ticket (invoice).
13. The system reloads the page and displays a message informing that there’s been sent an email with the ticket.
14. The system displays an username and password for the website the user should use to login.
15. The user then clicks a login button on the website.
16. System redirects the user to the login page.
17. The user enters the credentials and clicks on login.
18. The system checks if they match with any credentials from the database and it does.
19. The user is redirected to the after-login page.

Extensions:

2.1 The provided data is not valid – message for that and goes back to step 1.

3.1 The user is already registered – message for that goes to step 2.

9.1 There are some mistakes in the input information – system displays message and goes back to step 7.

10.1 The banking information is not valid – system displays message and goes back to step 5.

11.1 The transaction isn’t done – system displays message and goes back to step 7.

18.1 The credentials does not match with any records in the database – message for that and goes back to step 7 or step 1.

Post-condition: The use case is completed and the user has an account with which he can login in the website/applications if he hasn’t registered he won’t receive one or if he hasn’t payed he also won’t get account.

Use Case ID: 4

Goal: To add money to the event-account

Actors: Users, System

Pre-condition: The user must have an event account and be logged in the website.

MSS:

1. The user opens the website on the page with its event profile.
2. The user clicks on button “add money to event account”.
3. The system directs the user to a page with QR code.
4. The user scans the QR code.
5. The system recognizes the QR code and sends request for payment to the user’s banking mobile or desktop application.
6. The user inputs the desired money and clicks on “send”.
7. The system accepts the request and adds the money to the event-account.
8. The system then displays a message that the money was transferred successfully.

Extensions:

3.1 The system doesn’t recognize the QR code – displays message and goes back to step 1.

5.1 The system gives an error and doesn’t accept the transaction – displays message and goes back to step 1 .

Post-condition: The user successfully added money to his event account or if something failed the transaction is not proceeded.

Use case ID: 5, 24, 25, 32

Goal: To buy merchandise

Actors: user, system

Pre-Condition: The user must have an event account and must be logged in the website

MSS:

1. User opens the page with the merchandise and picks one or more items.
2. The system shows whether if it is in stock and it is and shows number of stocks.
3. When user is over with selecting the items he clicks on the “shopping basket” icon.
4. User on the page “shopping basket” clicks on button “order”.
5. System shows a form about shipping.
6. Users fills in the form.
7. System checks if there are any mistakes in the format of the input information and there isn’t.
8. System then processes the form and displays a new tab with methods of payments.
9. User picks one payment method.
10. System shows a form for the payment.
11. User fills in the form.
12. System checks if there are any mistakes in the format of the input information and it is.
13. System checks if the provided information is valid and it is.
14. System sends request about a shipping to the database.
15. System sends an email about confirmation about the shipping.
16. The transaction is done.
17. System makes record of the shipping to the database.

Extensions:

2.1 The system shows that the item is not in stock – system displays message and goes back to step 1.

4.1 There are some mistakes in the input information – system displays message and goes back to step 3.

9.1 There are some mistakes in the input information – system displays message and goes back to step 7.

10.1 The banking information is not valid – system displays message and goes back to step 8.

14.1 The transaction isn’t done – system displays message and goes back to step 7.

Post-condition: The user successfully purchased merchandise from the website or if something went wrong the transaction is not proceeded.

Use Case ID: 9, 27, 28

Goal: To book a camping spot

Actors: User, System

Pre-Condition: The user to have registered in the event and to have event account and to be logged in the website.

MSS:

1. The user opens the application with the camping spot.
2. The system displays a picture with the camping spots.
3. The user selects one.
4. The system sends a message that the camping spot is available to be booked.
5. The system displays a new tab with register form for the camping spot.
6. The user inputs his name and the unique ID of the participants he wants to reserve it with and confirms.
7. The system then checks whether every mentioned person is registered in the list of visitors and if the provided format is valid and it is.
8. The system then displays a text-box with which the user can pay.
9. The user enters his event bank account.
10. The system checks the database for the record and it exists.
11. The system displays a new text-box for a 3 number confirmation number, which the user gets by email.
12. The user inputs it.
13. The system checks if it matches any record of the database and it does.
14. The system makes the transaction.
15. The system then displays a message that the transaction is successful.

Extensions:

4.1 The camping spot is not available – the system displays message and goes back to step 2.

7.1 The provided information is not valid – goes back to step 5.

10.1 The provided information is not valid – goes back to step 8.

13.1 The provided information is not valid – goes back to step 11.

14.1 The system doesn’t accept the transaction. – displays message and goes back to step 8.

Post-Condition: The user successfully booked a camping place.

Use Case ID: 20,21, 34

Goal: To generate and read a log file.

Actors: User, System

Pre-Condition: The user must have an event-account and be logged in the website/application.

1. The user presses a button to purchase an item.
2. The system takes the information on the purchase.
3. The system generates a log file with that information.
4. The system then reads the log file and by order ID.
5. The system stores the log file in the database.
6. The system prints out the log file.

Post-Condition: The system successfully generated and stored a log file.

ID: 16

Goal: To add money via the ATM

Actors: User, System

Pre-Condition: The user must have an event-account and to be in the event.

MSS:

1. The users opens the ATM system.
2. System prints a QR code.
3. The user scans the QR code with its mobile phone.
4. The system recognize it and sends a request to the online bank app.
5. The user then via the banking app transfers money to the event-account.
6. The system accepts the request and adds the money to the event-account.
7. The system displays a message.
8. The system adds the transaction to the database.
9. The system then prints a log with the transaction number.

Extension:

3.1 The user cannot scan the QR code and it goes back to step 1.

4.1 The system cannot recognize the request and goes back to step 2.

6.1 The system cannot accept the request and goes back to step 2.

Post-condition: The user has successfully added money to his event-account

Use Case ID: 13, 23, 31

Goal: To be able to buy drink/snack

Pre-Condition: The user must have an event-account, to be at the event.

MSS:

1. The system opens the application about the drink/snack
2. The user chooses what they desire.
3. The system checks if the desired choice is available to be dispensed and it is.
4. The system prints a QR code.
5. The user scans the QR code with its mobile phone.
6. The system recognize it and checks if there’s enough balance in the user’s event account and there is.
7. The system checks the number of items in stock and there are enough of number of items for the user.
8. The system accepts the transaction.
9. The system displays the a message.
10. The system dispenses the drink/snack.
11. The system modifies the number of items in stock.

Extensions:

4.1 The user cannot scan the QR code and it goes back to step 1.

6.1 The system cannot recognize the request and goes back to step 1.

6.2 The user doesn’t have enough balance in the event account – system displays message and goes back to step 1.

7.1 The system doesn’t accept the transaction and goes back to step 1.

Post-condition: The user successfully purchased snack/drink.

Use Case ID: 11

Goal: To be able to loan materials

Actors: System, user

Pre-Condition: The user must have an event-account, to be at the event.

MSS:

1. The system opens the application about the loaned materials.
2. The user chooses what they desire.
3. System prints screen to scan the barcode of the email.
4. User puts the email for the barcode for scan.
5. System recognizes the barcode of the user.
6. System prints an QR code.
7. The user scans the QR code with its mobile phone.
8. The system recognize it and checks if there’s enough balance in the user’s event account and there is.
9. The system accepts the transaction.
10. The system displays the a message.
11. System allows the user to loan the material.

Extension:

5.1 The system doesn’t recognize the barcode. Goes back to 1.

6.1 The user cannot scan the QR code and it goes back to step 1.

8.1 The system cannot recognize the request and goes back to step 1.

8.1 The user doesn’t have enough balance in the event account – system displays message and goes back to step 1.

9.1 The system doesn’t accept the transaction and goes back to step 1.

Post-Continuation: The user successfully loaned materials if he had enough balance in his event account.

Use Case ID: 15, 26, 29

Goal: To check-out

Actors: System, user, employee

Pre-Condition: The user must have an event-account, to be at the event.

MSS:

1. User opens event-application to transfer the money from the event account to the bank account.
2. Employee scans the QR code and makes a payment request to the system.
3. System accepts the request and withdraws the money from the event-account.
4. System accepts the request and transfers money back to the bank account of the user.
5. Employee then makes request to the system to delete the event-account.
6. System deletes the account.

Extensions:

3.1 System doesn’t accept the request. Goes back to step 2.

6.1 System doesn’t accept request and goes back to step 5.

Post-condition: The user has successfully deregistered from the event system.

Use case ID: 18

Goal: To track the number of visitors

Actors: user, system

Pre-condition: The user must have an event-account.

MSS:

1. System displays a tab to scan the barcode.
2. User puts his barcode there.
3. System checks if the barcode matches any record from the database and it does.
4. System increments by one the trackbar for the scan.

Extensions:

* 1. System can’t find any matching - goes back to step 1.

Post-condition: To successfully track the number of users.

Use Case ID: 22

Goal: To cancel booking of camping place

Actors: user, system

Pre-condition: To have booked a camping place successfully (to have event account too) and to be logged in the website.

MMS:

1. The user clicks on a button “cancel booking”.
2. The system displays a page with the orders.
3. The user clicks on the place he wants to cancel his booking.
4. The system sends a confirmations via message.
5. The user clicks “yes”.
6. The system checks if the booking matches the one from the database and it does.
7. The system removes the booking from the database.
8. The system returns the money of the booking into the event account of the user.
9. The system displays message with confirmation.

Extension:

3.1 The user doesn’t have any bookings – displays a message and goes back to step 2.

Use Case ID: 35, 30

Goal: To check in

Actors: User, system

Pre-condition: To have an event account.

MMS:

1. The user scans the barcode.
2. System checks if it matches any records from the database and it does.
3. The user is allowed to enter.

Extensions:

* 1. The user doesn’t have a barcode – the system displays a message about that and displays a QR code with which the user can pay directly the price of the ticket.

Post-condition: To successfully check in the event.

Use Case ID: 36

Goal: To check in (camping)

Actors: User, system

Pre-condition: To have an event account, be at the event.

MMS:

1. The user enters camping number.
2. The system checks if the camping number matches any records from the database and it does.
3. The system then requests for the unique ID of the user.
4. The user inputs it.
5. The system checks if the unique ID matches the one of the reservation and it does.
6. The system displays a message that the user can enter the camping spot.

Extension:

* 1. The user didn’t book camping spot – the system displays a message then displays a QR code with which the user can pay directly the price of the booking.

Post-condition: To successfully check or book a camping spot.

Use Case ID: 22

Goal: To cancel a booking (camping)

Actors: User, system

Pre-condition: To have an event account and have booked a camping and be in the website.

1. The user clicks on page “my camping spot”
2. The system shows details of the booked camping spot.
3. The user clicks on button “cancel booking”.
4. The system displays a message about confirmation about the cancel of the booking.
5. The user selects he wants to cancel it.
6. The system then deletes this record from the database and displays a message about that.
7. The system makes a transaction to return the money of the reservation in the event-account of the logged in user.

Extensions:

5.1 The user doesn’t want to cancel it – goes back to step 1.

Post-condition: The user has successfully cancel reservation if he had one.

Use Case ID: 17

Goal: To login in successfully in the website/application

Pre-condition: The user must have an account and the website open

Actors: User, System

MSS:

1. The user clicks on “login” button on the website.
2. The system redirects him to a form – login page.
3. The user fills in his credentials
4. The system checks if the provided input is in the correct format and it is.
5. The system checks if the credentials match any records from the database and it does.
6. The system redirects the user to the “home” page.

Extensions:

* 1. The input is in the wrong format – displays message for that and goes back to step 3.

5.1 The system cannot find any matching records from the database – displays message for that and goes back to step 3.

Post-condition: The user has successfully logged in the website.

Use Case ID: 19

Goal: To log out successfully

Actors: User, system

Pre-condition: To be logged inside the website/application

MSS:

1. The user clicks in the navigation bar on button “log out”
2. The system recognizes it and ends the session
3. The user is logged off

Post-condition: The user has successfully logged out of the website.

Use Case ID: 37

Goal: To check out successfully

Actors: User, system

Pre-condition: To have booked a camping spot

MSS:

1. The user clicks on button “check out”.
2. The system finds the matching records in the database and deletes the reservation.

Post-Condition:

To successfully check out of a camping spot.

Use Case ID: 10

Goal: To check the balance of the event-account

Actors: User, system

Pre-condition: To have an account and have the website opened

MSS:

1. The user clicks a button “check balance”
2. The system takes the request.
3. The system finds the record of the balance.
4. The system displays it in a message box.

Post-condition: The user can see his balance successfully.

Use Case ID: 12

Goal: To return the loaned materials

Actors: User, system

Pre-condition: To have an account and be at the event.

MSS:

1. The user clicks on a button “return loaned materials”.
2. The system displays a list of items.
3. The user selects what kind of item it is.
4. The system then takes the loaned material.

Post-condition: To return the loaned material successfully.

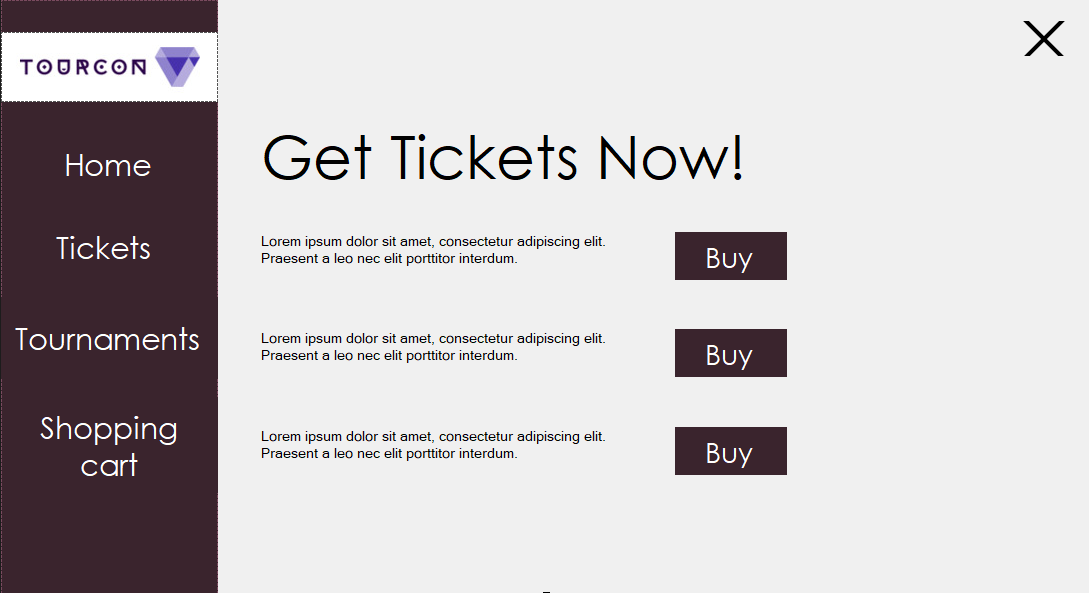
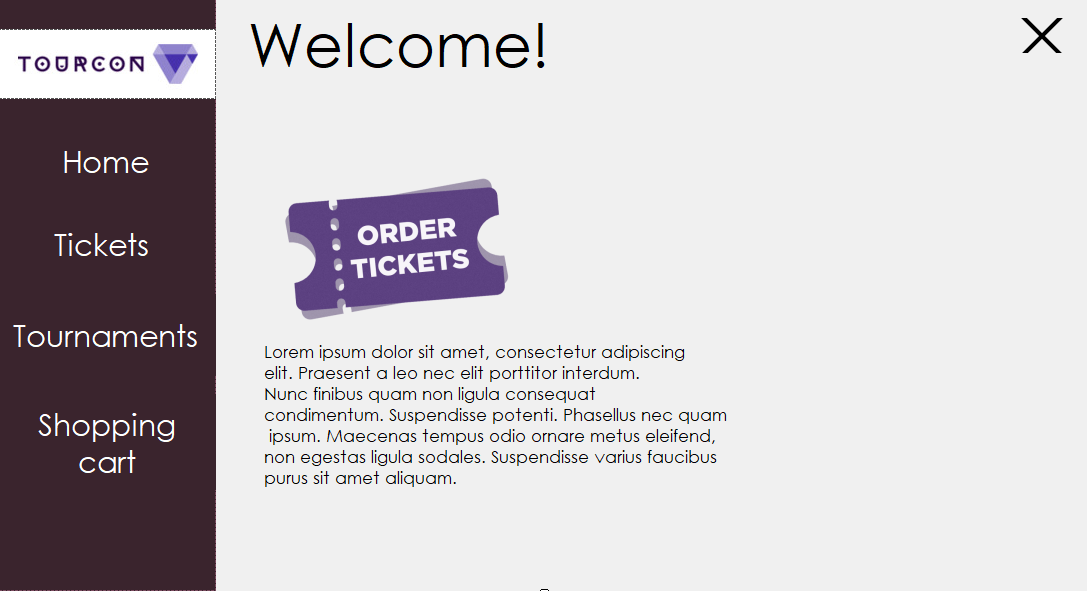
# Functional requirements

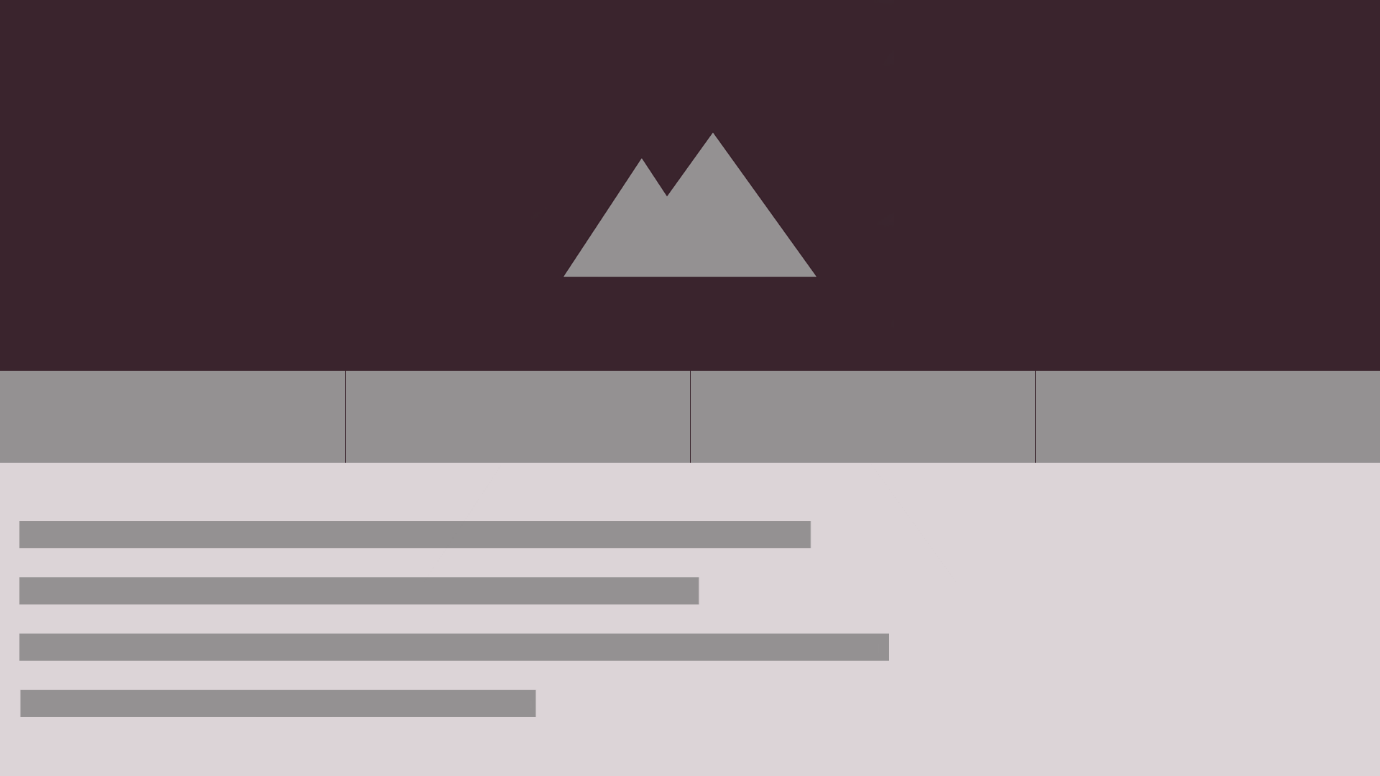
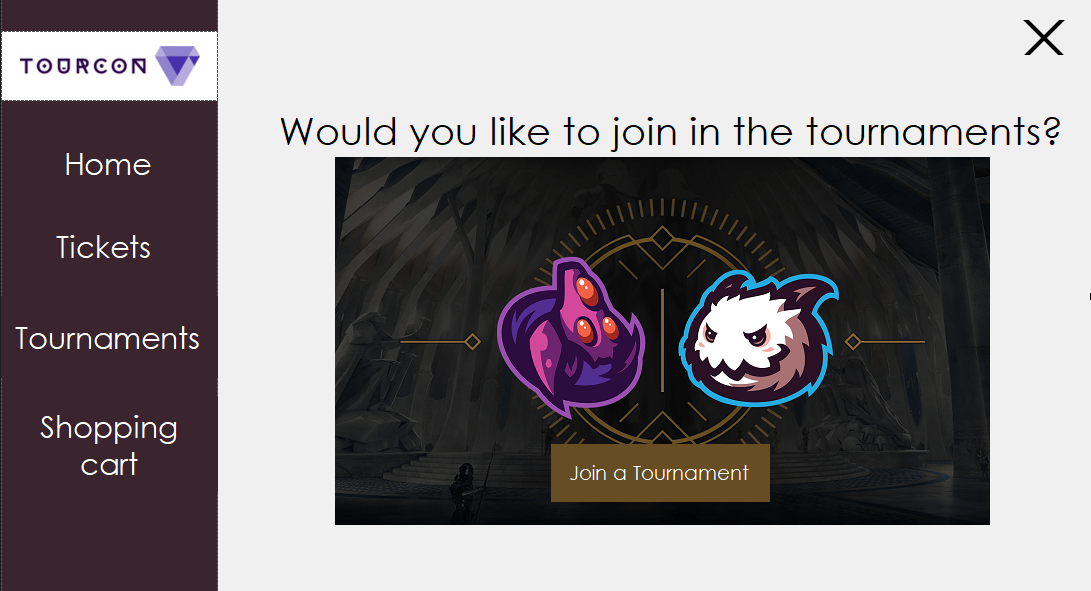
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Functionality | Must Do | Should Do | Could do | Won’t do |
| 1 | To be able to book ticketsvia website | X |  |  |  |
| 2 | To be able to register visitors via website | X |  |  |  |
| 3 | To send the identification barcode via website through email | X |  |  |  |
| 4 | To add money to the event-account via website | X |  |  |  |
| 5 | To be able to buy merchandise via website |  |  | X |  |
| 6 | To be able to show the running live tournaments via website |  |  | X |  |
| 7 | To be able to book a camping spot via application |  |  | X |  |
| 8 | To be able to cancel a booking of a camping spot via application |  |  | X |  |
| 9 | To be able to book a camping spot via the website | X |  |  |  |
| 10 | To be able to check the event balance via application | X |  |  |  |
| 11 | To be able to loan materials via application | X |  |  |  |
| 12 | To be able to return the loaned materials via application | X |  |  |  |
| 13 | To be able to buy snacks/drinks via application | X |  |  |  |
| 14 | To be able to return the loaned materials via application | X |  |  |  |
| 15 | To be able to check out of the event via application | X |  |  |  |
| 16 | To be able to add money to the event account via ATM via application | X |  |  |  |
| 17 | To be able to log in via the website | X |  |  |  |
| 18 | To have a statistics on the number of visitors via application | X |  |  |  |
| 19 | **To be able to log out via the website.** | X |  |  |  |
| 20 | **To be able to read a log file via application.** | X |  |  |  |
| 21 | **To be able to store a log vile via application.** | X |  |  |  |
| 22 | **To be able to cancel a booking via website.** | X |  |  |  |
| 23 | **To be able to select multiple products via application.** | X |  |  |  |
| 24 | To be able to select multiple products via website. | X |  |  |  |
| 25 | To show the number of stock of products via application. | X |  |  |  |
| 26 | **To be able to refund the left event account money on check out via application.** | X |  |  |  |
| 27 | **To be able to select a camping spot via website** | X |  |  |  |
| 28 | **To be able to check if the camping spot is reserved via website** | X |  |  |  |
| 29 | **To be able to change ordered products via application.** | X |  |  |  |
| 30 | **To be able to scan the barcode.** | X |  |  |  |
| 31 | **To be able to change or remove items in order via application** | X |  |  |  |
| 32 | **To be able to change or remove items in order via website** | X |  |  |  |
| 33 | **To be able to cancel order of merchandise via website** |  | X |  |  |
| 35 | **To be able to check in via the application** | X |  |  |  |
| 36 | **To be able to check in camping via the application** | X |  |  |  |
| 37 | **To be able to check out in camping via the application** | X |  |  |  |

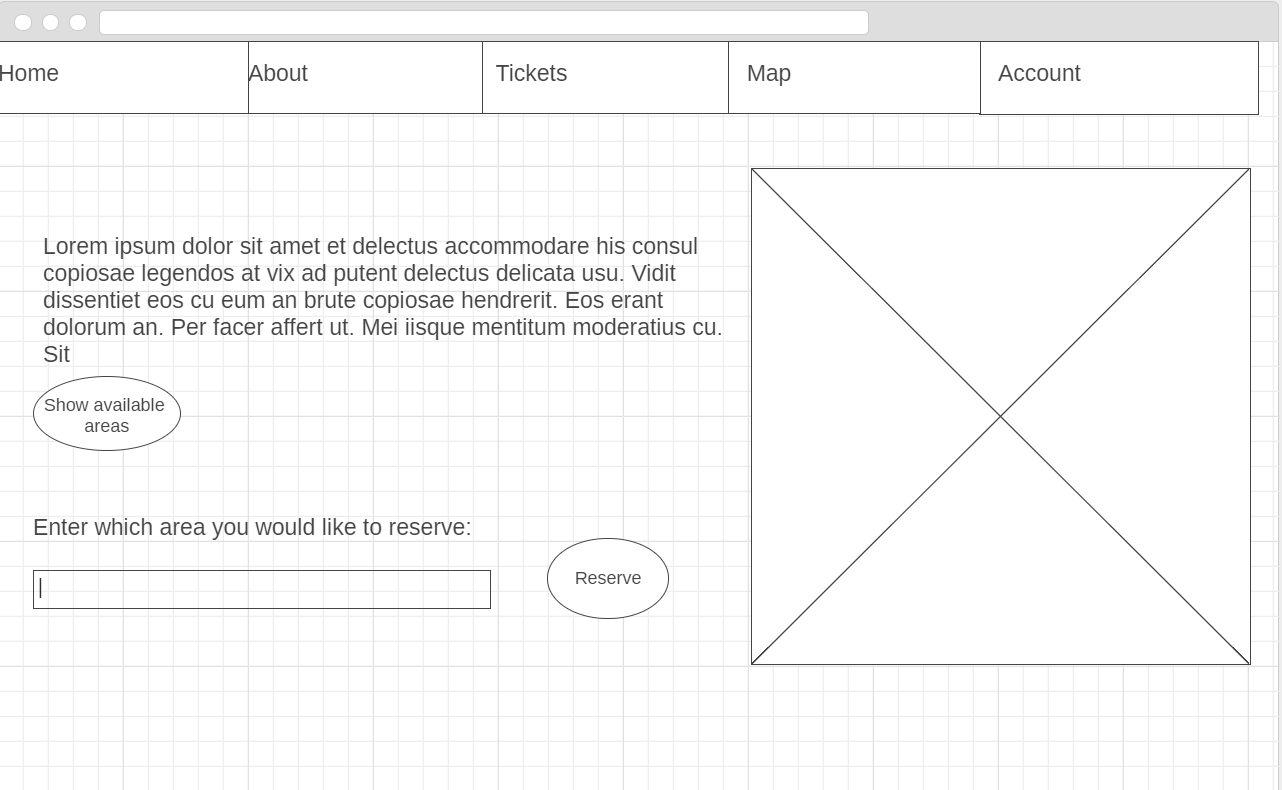
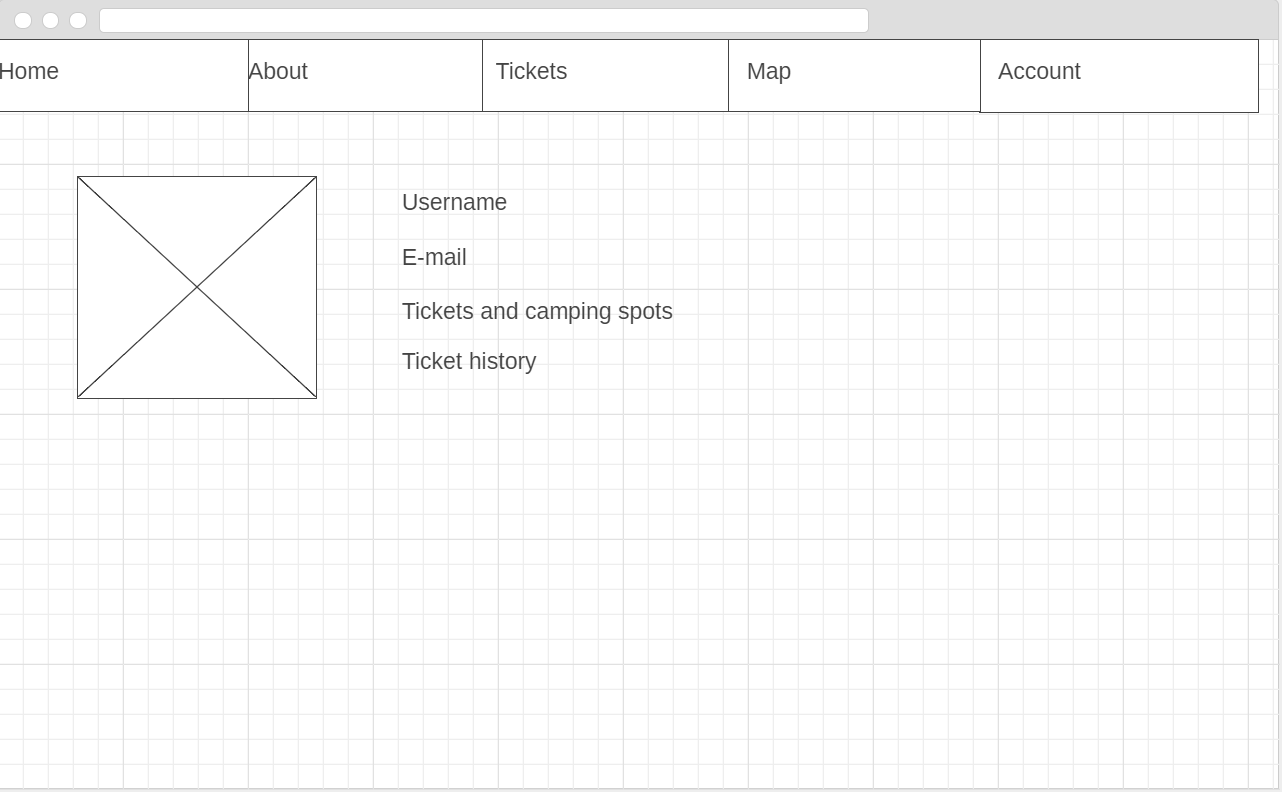
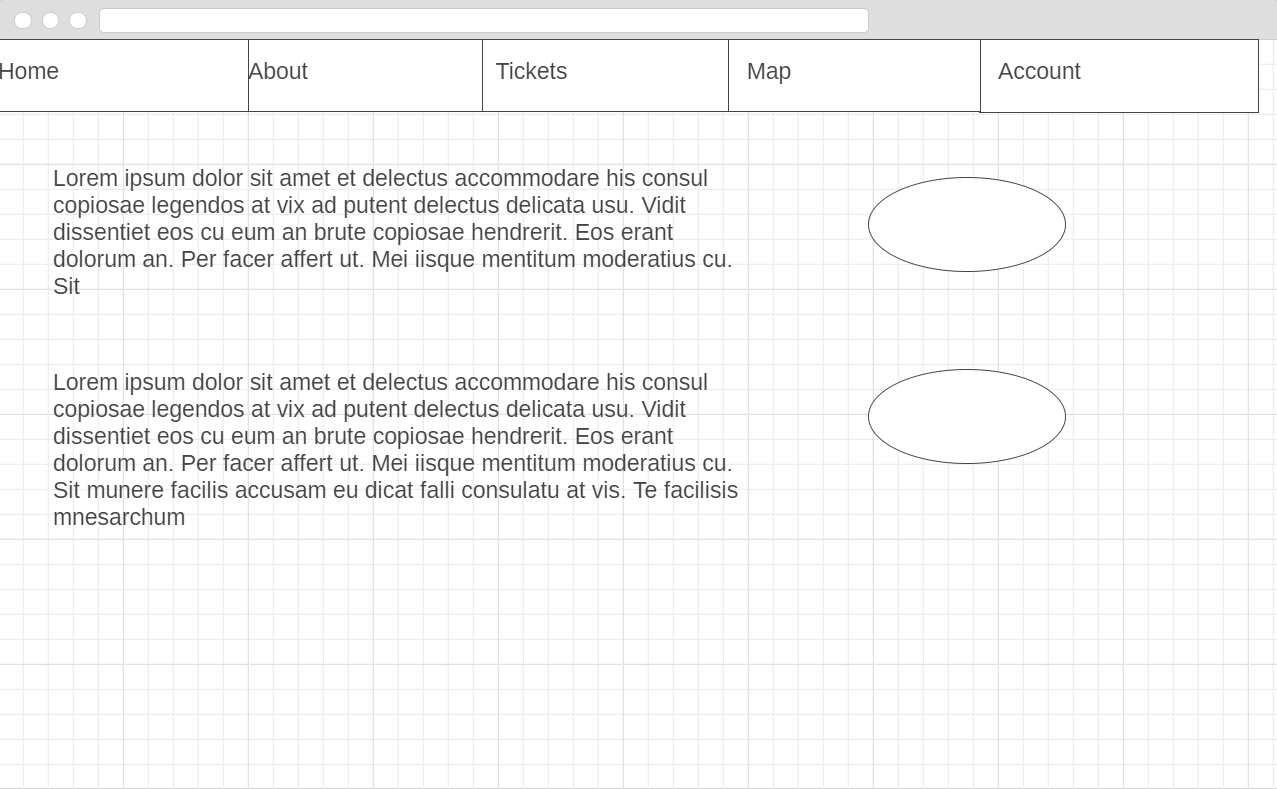
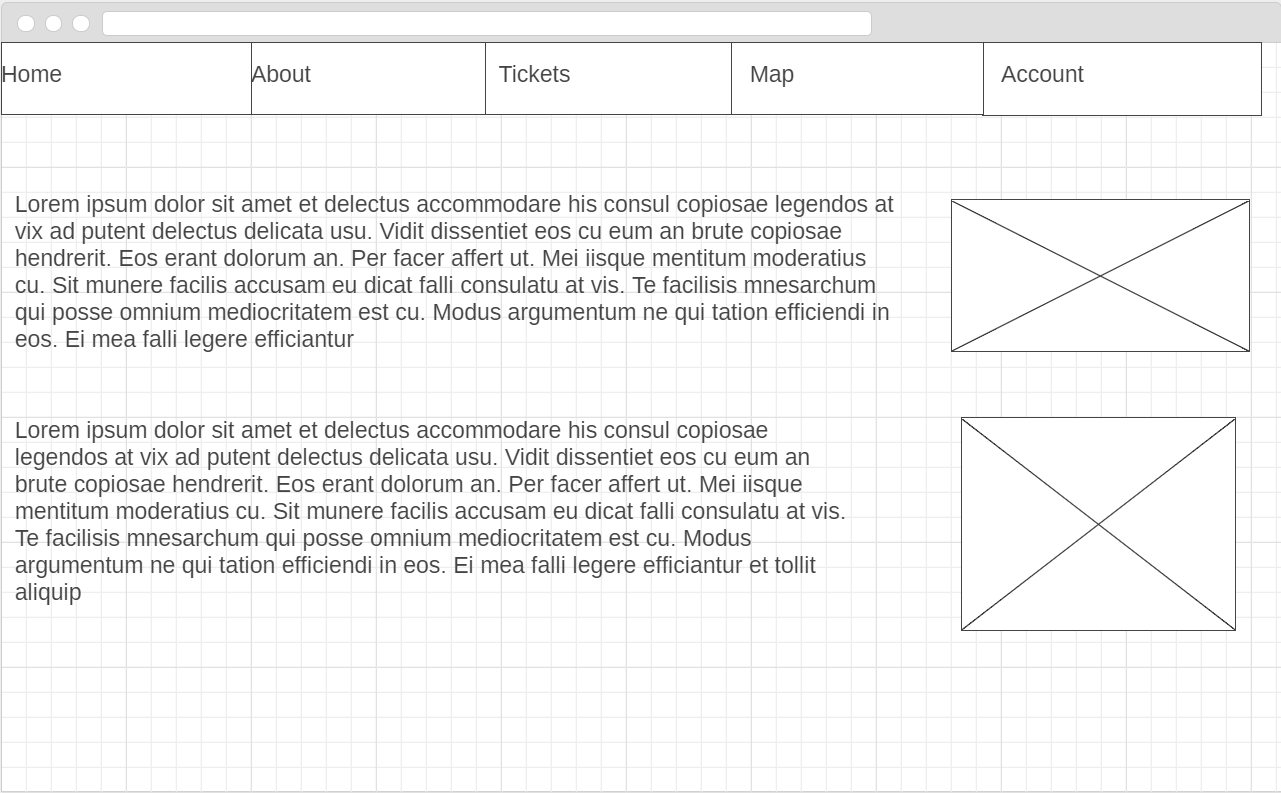
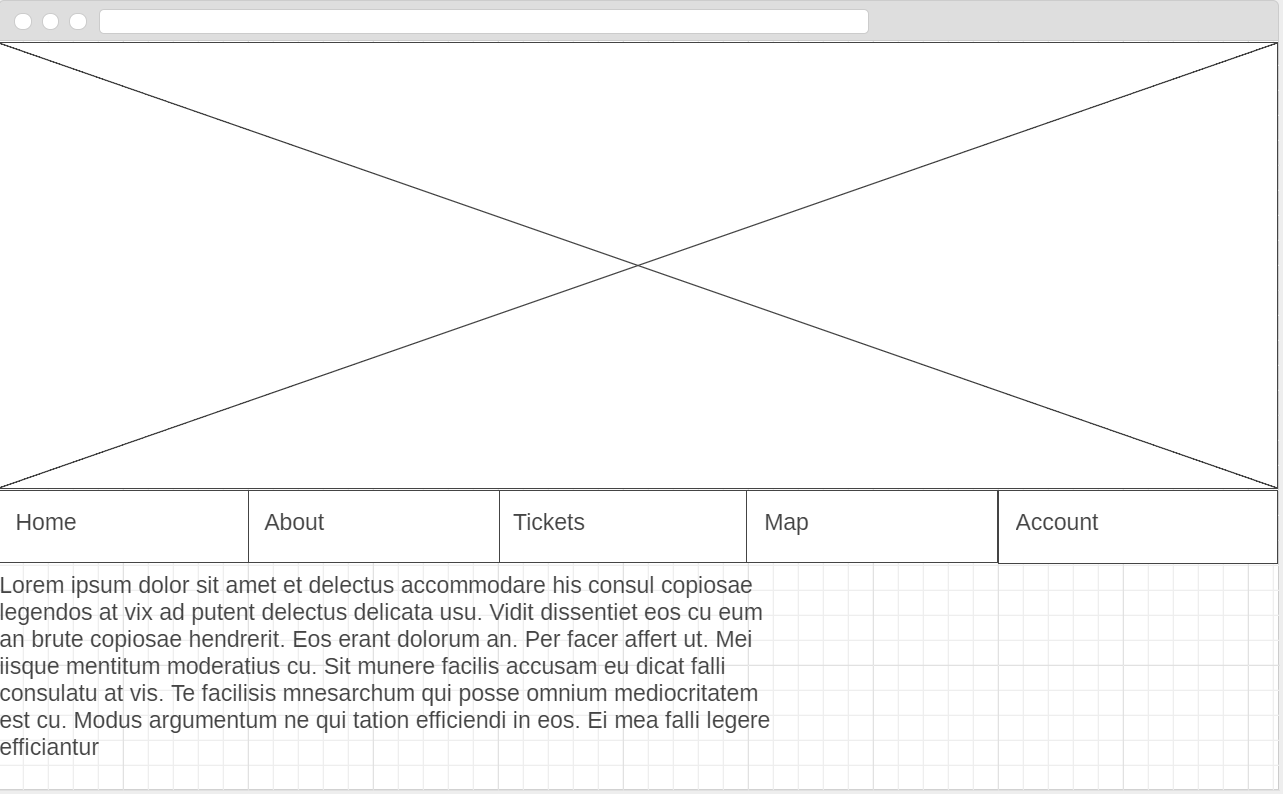
# Nonfunctional Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Type | Requirements | Must | Should | Could | Would |
| 1 | Usability | The program will display the exact amount of each element that is used in the reaction. | X |  |  |  |
| 2 | Security | The data will be private (encrypted) |  | X |  |  |
| 3 | Manageability | All of the errors that can occur will be caught with exceptions. The user will be notified. | X |  |  |  |
| 4 | Manageability | The application will be implemented in such a way that can be reused in the future | X |  |  |  |
| 5 | Manageability | All of the fields and forms will be validated. | X |  |  |  |
| 6 | Usability | When the website is closed the system will check if the last reaction is unsaved, so it can notify the user. |  |  | X |  |
| 7 | Usability | Responsive design and GUI. | X |  |  |  |

# GUI and wireframe

We all agreed that the design for the website and the gui will look the same.





# ERD Diagram

