

# YuckyJoki

Web Development Project

Detailed design document/user case analysis

# Contents

1. Introduction.....	2
2. Terminology:.....	2
3. Database structure: .....	3
4. User case analysis.....	4
4.1. User case: Viewing the front page.....	4
4.2. User case: Registering account.....	4
4.3. User case: Logging into account .....	4
4.4. User case: Filling a form to request a boat ride (not a registered user) .....	5
4.5. User case: Filling a form to request a boat ride (registered user).....	5
4.6. User case: Filling a form to offer a boat ride .....	5
4.7. User case: Browsing open offers and accepting offer .....	6
4.8. User case: Browsing open requests and accepting request .....	6
4.9. User case: Viewing confirmed bookings, editing contact details.....	7
5. User interface mock-up .....	8

## 1. Introduction

This is a detailed design for web development project YuckyJoki. YuckyJoki provides a platform for sharing a boat ride in Oulu River and surrounding sea areas.

User case analysis consists of different scenarios which the user may encounter when using YuckyJoki website. Aim is to map out all features which the project development team must create for achieving high quality product with successful service and user experience. For each case there are different scenarios for different user needs.

YuckyJoki has only one type of users when registering account. This means that all users have same type of account and can have the same view; all users can see offers and requests for boat ride, all user settings are the same, all users can offer a ride and ask for a ride.

User offering a ride can define number of seats and price for each offer, which makes it flexible. Same user can have multiple offers, for example one on Saturday from 16-17, another one from 17-18, etc. Same goes with those looking for a ride, they can define number of seats, date, time, start point and end point for every request that they make, or they can make multiple requests.

Users can browse through a list of open offers/requests and click to book one. Also search function can be available.

## 2. Terminology:

- table: OFFER is what boat owner creates
  - Offers/Available captains can be browsed from front page

- “front page” is a working title for home.html
- table: REQUEST is what passenger creates
  - Requests from passengers can be browsed from **Map** page
  - “map page” is a working title for index.html, “Book and offer” on user interface
- Open requests /open offers that user has made are showed on **Map** page.
- table: BOOKING is created when offer/request has been confirmed by the counterpart.
- Confirmed bookings are showed on **Account** page.

### 3. Database structure:

- table: **user**
  - User ID
  - First name
  - Last name
  - Email address
  - Password
  - Phone number
- table: **stop**
  - stop ID
  - stop Name
  - latitude
  - longitude
- table: **offer**
  - offer id
  - user id
  - stop ID
  - stop ID
  - stop ID (as many stops as the boat owner wants to be available)
  - date
  - start Time
  - end Time
  - seatsAvailable
  - pricePerSeatPerHour (for example)
- table: **request**
  - request id
  - user id
  - starting stop id
  - ending stop id
  - date
  - start time
  - end time
  - number of passengers
- table: **booking**
  - booking id
  - offer id
  - boat owner user id
  - request id
  - passenger user id

## 4. User case analysis

### 4.1. User case: Viewing the front page

**Role: visitor /anyone when not logged in**

Main success scenario:

1. Visitor can see
  - a. YuckyJoki logo and name of the web site
  - b. Page options on the top menu bar
  - c. Log in options
  - d. Link for registration
  - e. 1-2 exciting and beautiful pictures of Oulu River and surrounding sea areas
  - f. *Captains Available*: max five lines of current offers that are available, each line showing only summary: captain first name and last name, date, start time, start point, stop point (data comes from **table: offers**).
  - g. Social media links, contact details
2. Visitor clicks on *Captains available*
  - a. Pop up opens for registration (which means visitor cannot view the details, but must register)

### 4.2. User case: Registering account

**Role: passenger/boat owner (same for both)**

Main success scenario:

1. User is on **Front page** and clicks on “Register here”
2. A pop-up window opens for user to fill in:
  - a. First name
  - b. Last name
  - c. Email address
  - d. Password
  - e. Phone number
  - f. User is required to fill all details, otherwise gets a notification to do so
3. When all details are filled, user presses “Register”
  - a. Software stores new user into database (**table: user**)
  - b. User gets confirmation that account has been registered with the given email address as a username.
  - c. User “*First name Last name*” is now visible in the top right corner (log in button is not visible any more).

### 4.3. User case: Logging into account

**Role: passenger/boat owner (same for both)**

Main success scenario:

1. User is on front page, fills in email and password, and clicks “Log in”
  - a. If username and password do not match, user is informed about this
  - b. If username and password match, user details are fetched from database (**table: user**)
  - c. User “*First name Last name*” is now visible in the top right corner (log in -button is no longer visible)

- d. User gets confirmation that she/he is logged in

#### 4.4. User case: Filling a form to request a boat ride (not a registered user)

**Role: visitor (anyone who is not logged in)**

Main success scenario:

1. Visitor is on **Map** page.
2. Visitor can see map that covers Oulu river, river delta and surrounding sea areas. Map shows stops of public piers. Stops are numbered on the map for identification. Stops are listed next to the map with stop id and name of the stop/pier).
3. Visitor can see links to two forms: 1. *Looking for a Boat Ride*, 2. *Offering a Boat Ride*
4. Visitor clicks on form (either)
  - a. Pop up opens for registration (which means visitor cannot fill the form, but must register)

#### 4.5. User case: Filling a form to request a boat ride (registered user)

**Role: passenger (user looking for a ride)**

Main success scenario:

1. User is logged in and is on **Map** page
2. User can see
  - a. map that covers Oulu river, river delta and surrounding sea areas.
  - b. Map shows stops of public piers. Stops are numbered on the map for identification.
  - c. Stops are also listed next to the map with stop id and name of the stop/pier.
3. User can see links to two forms: 1. *Looking for a Boat Ride*, 2. *Offering a Boat Ride*
4. User chooses 1. *Looking for a Boat Ride*
5. Form opens to fill a request
  - a. Date
  - b. Timeframe (starting time and ending time).
  - c. Number of passengers
  - d. Starting point (stop id and name)
  - e. Ending point (stop id and name)
  - f. User clicks "Submit your request"
6. Form closes.
7. Request is saved into database (table: request).
8. User can see summary of their open request on top of **Map** page (date, starting point, ending point)
9. User can now see a yellow flag on the top right corner of the page next to username indicating a open request.

#### 4.6. User case: Filling a form to offer a boat ride

**Role: boat owner (user offering a ride)**

Main success scenario:

1. User is logged in and is on **Map** page
2. User can see
  - a. map that covers Oulu river, river delta and surrounding sea areas.
  - b. Map shows stops of public piers. Stops are numbered on the map for identification.
  - c. Stops are listed next to the map with stop id and name of the stop/pier.

3. User can see two forms: 1. *Looking for a Boat Ride*, 2. *Offering a Boat Ride*
4. User chooses 2. *Offering a Boat Ride*
5. Form opens to fill an offer
  - a. Date
  - b. Timeframe (starting time and ending time)
  - c. Number of seats available
  - d. Boat type
  - e. Available stop (stop id and name)
  - f. Available stop (stop id and name)
  - g. Available stop (stop id and name)
  - h. Available stop (stop id and name)
  - i. Available stop (stop id and name) (user can choose all stops where boat is able to take passengers on and off board).
  - j. Price per person.
  - k. User clicks "Submit your offer"
6. Form closes.
7. Offer is saved into database (table: offer)
8. User can see summary of their **open offer** on top of **Map** page (date, start time, boat type, number of seats)
9. User can now see a **yellow flag** on the top right corner of the page next to username indicating **an open offer**.

#### 4.7. User case: Browsing **open** offers and accepting offer

**Role: passenger (user looking for a ride)**

Main success scenario:

1. User is on **Front** page
2. User can see summary of *Captains available*
3. User clicks on *Captains available*
  - a. Instead of summary, now user can see list of all captains available and all details. List shows 10 rows sorted by date, earliest date of departure on top. (Data is retrieved from table: offer)
  - b. User clicks on *Show next ten offers*
  - c. Another 10 rows become visible
4. At the end of each row there is an option to click *Book this boat*
  - a. User clicks *Book this boat*
  - b. User is asked to confirm their booking and to agree on terms and conditions (cancelling policy, web site is not responsible for anything etc.)
  - c. User gets notification that their booking is now confirmed and that it can be viewed on **Account** page.
  - d. User can now see a green flag on the top right corner of the page next to username indicating **a confirmed booking**.
  - e. A new column is added on table: **booking**, (column is made of details of the offer and passenger).

#### 4.8. User case: Browsing **open** requests and accepting request

**Role: boat owner (user offering a ride)**

Main success scenario:

1. User is logged in.
2. User is on **Maps** page.
3. User can see
  - a. his/her own active offers on top of the page (or if there isn't any, there is a notification: "You have no active offers")
  - b. map and list of stops
  - c. link to fill form for offer/request
  - d. Summary of *Active requests* from passengers (similar list as *Captains available* on **Front** page)
4. User clicks on *Active requests*
  - a. Instead of summary, now user can see list of all active requests available and all details. List shows 10 rows sorted by date, earliest date of departure on top. (Data is retrieved from **table: offer**)
  - b. User clicks on *Show next ten offers*
  - c. Another 10 rows become visible
5. At the end of each row there is an option to click *Take this request*
  - a. User clicks *Take this request*
  - b. User is asked to confirm this booking and to agree on terms and conditions (cancelling policy, web site is not responsible for anything etc.)
  - c. User gets notification that their booking is now confirmed and that it can be viewed on **Account** page.
  - d. User can now see a green flag on the top right corner of the page next to username indicating a **confirmed booking**.
  - e. A new column is added on **table: booking** and it is made of details of the request and boat owner.

#### 4.9. User case: Viewing **confirmed** bookings, editing contact details

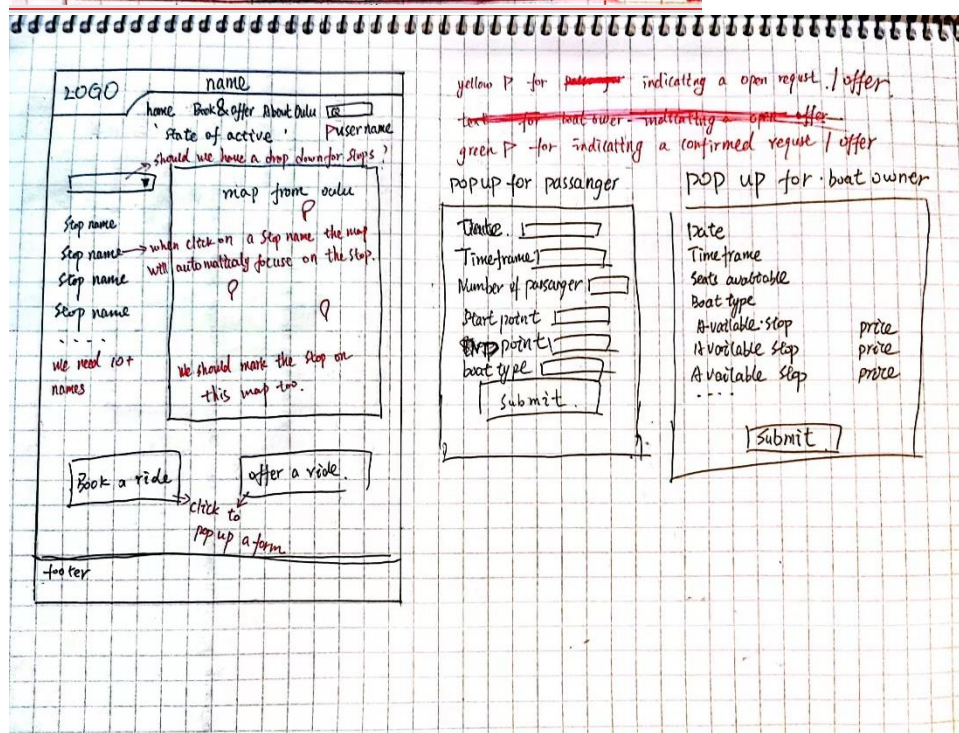
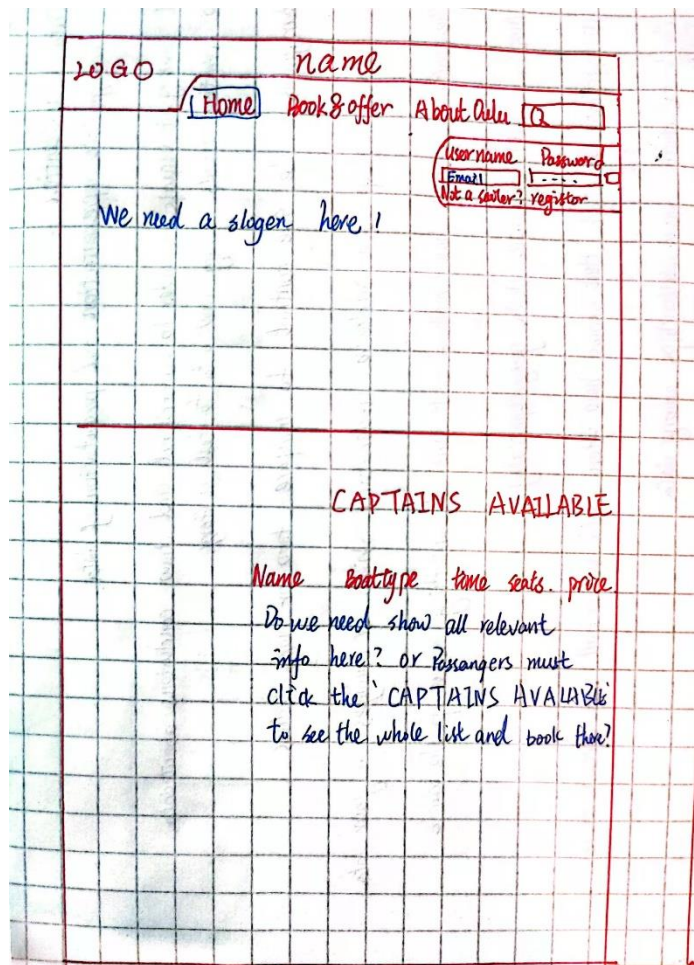
**Role: passenger/boat owner (same for both)**

Main success scenario:

1. User is logged in and on **Account** page
2. User can see
  - a. active bookings with all details (or notification that there are no active bookings)
  - b. registered contact details
  - c. option to edit contact details
3. User clicks on *Edit contact details*
  - i. Form opens with current contact details (phone number, email)
  - ii. User clicks to save changes



Detailed Design for Project YuckyJoki, March 2022  
A. Kirstinä, Y. Li





Click 'Captains available'

Home Book offer About Us

name → captain available.

	name	date	time	seats	start	end	price
1	~	~	~	~	~	~	~
2	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~
5	~	~	~	~	~	~	~
6	~	~	~	~	~	~	~
7	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~
9	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~

1 2 3 ... next

POP UP when click Book

Claim

~

~

~

Info of the order ---

Confirm the Book

Cancel

Click user name to come into Account page.

LOGO

Name

home B ... About

Name

file (Edit)

First name Last name

Email

Phone number

Book info.

Book id offer id request id user id

start end time price

Contact