

# YuckyJoki

Web Development Project

Testing Plan

## Contents

Testing Plan for Project YuckyJoki (Web Development Project) .....	1
1. Introduction.....	2
2. Objects.....	2
3. For what purpose are we testing .....	2
4. What is left outside the scope of testing.....	3
5. Framework.....	3
6. Detailed test plan for database interaction, user interface and user experience .....	3
6.1. Features and database interaction.....	3
6.2. Registration status.....	5
6.3. Style responsiveness.....	5
6.4. Browsers .....	5
6.5. Contents and user experience .....	5

### 1. Introduction

This is a testing plan for project YuckyJoki. YuckyJoki provides a platform for sharing a boat ride in Oulu River and surrounding sea areas.

### 2. Objects

- to evaluate quality of software: goal is to reach high quality (within the resources available).
- to find possible defects in the software and ways to fix them.
- also: to ensure that the set requirements for the project are met (design document is used here as a reference)
- also: test to see if there are any improvement ideas for future, should the project be developed further.

### 3. For what purpose are we testing

- **Functionality and cohesion:** To ensure separate parts of project are merged into one successfully.
- **User interface and user experience:** Smooth and logical navigation in the software.
- **Configuration:** Responsiveness and adjustability in different environments (mobile, desktop and different browsers).
- **Performance:** database and server performance with usual number of users and usual expected transactions.

#### 4. What is left outside the scope of testing

This plan does not include testing for conformance to meet standards and policies, as the project has not defined standards. Secondly, this project has no real-life customer and thus there will be no testing for acceptance carried out by customer. Thirdly, testing plan leaves out testing for performance under unusual conditions and defining a breaking point.

This plan is designed to be carried out at the end of the project, when nearly 100% of all code has been written and joined together into one software. This plan does not include testing which is carried out on developer level during the execution phase, but naturally that has been done while each developer has been testing their own code before merging it into the main branch.

#### 5. Framework

- Testing is carried out by programmers working on the project. There is no user level testing nor separate technical tester.
- Testing targets mainly component and system level testing.
- Test cases are created based on intuition: same type of data that would be expected to be entered in and requested out of this type of software.
- Time and resources are setting limitations on how broadly to test and thus most critical features will take priority (database interaction).
- Aim is both to find errors and to find a cause in order to fix possible defects.

#### 6. Detailed test plan for database interaction, user interface and user experience

Each task is evaluated with status of good, poor, or missing. Good is for functionality that works as expected, poor stands for functionality that exists but could be improved. If feature isn't available, it is evaluated as missing.

Functionality is marked with an asterisk (\*) if it is not included in project's design documentation.

Qualitative features (user experience) will be evaluated with same status (good, poor, missing) and can also include more detailed feedback.

##### 6.1. Features and database interaction

- Registration
  - notification to user that registration was successful
  - database reflects new user registration
- Logging in
  - logging in with username and password
  - database responses to logging request that is made with *correct* credentials
  - database responses to logging request that is made with *incorrect* credentials
  - error if wrong username/password
  - help to reset username/password if it is forgotten\*
  - indication of logged in status (database reflects username in UI)
  - logging out

- Creating routes
  - map with stops (piers)
  - real time user location reflecting on the map
  - creating an offer
  - database reflects new offer
  - creating a request (user must be logged in)
  - database reflects new request
  - creating multiple offers/requests made of different variables
  - indication that user has open offer/request (yellow flag)
- Managing *own* open requests/offers
  - viewing own open requests/offers (retrieved from database)
  - cancelling own open requests/offers (deleted from database)
- Viewing, searching and booking *requests made by other users*
  - Viewing open requests (list)
  - Searching for open requests (search option)
  - Making a booking (booking reflects on database)
  - Agreement to terms and conditions
  - indication that user has a booked a trip based on request (green flag)
- Viewing, searching and booking *offers made by other users*
  - Viewing open offers (list)
  - Searching for open offers (search option)
  - Making a booking (booking reflects on database)
  - agreement to terms and conditions
  - indication that user has a booked a trip based on offer (green flag)
- Getting notification that another user has made a booking based on my offer/request \*
- Handling bookings
  - indication of a booking (green flag in UI)
  - viewing made bookings
  - cancelling made bookings (reflects on database)\*
- Account settings
  - viewing existing account details
  - editing email (reflects on database)
  - editing phone number (reflects on database)
  - editing password\* (reflects on database)
  - viewing changed (new) details
- Feedback
  - sending feedback to developers (user)
  - receiving feedback from users (developer team)

- Internal communication between users
  - sending communicating to other users
  - receiving communication from other users
- Links
  - internal links, navigation inside the software
  - external links\*

## 6.2. Registration status

Test all the same as 6.1 with different log-in status, where applicable.

- not logged in
- logged in

## 6.3. Style responsiveness

Test all pages and pop ups with different screen size

- pc
- mobile/tablet\*

## 6.4. Browsers

Test all the same as 6.1 with different browsers\*

- mobile: chrome, edge, Firefox, opera
- pc: chrome, edge, Firefox, opera

## 6.5. Contents and user experience

Test all the same as 6.1 with user experience perspective.

- Instructions and logic in user experience