

Todo List

Documentation

## Table of contents

|   |    |
|---|----|
| 1 Modification log.....                             | 3  |
| 2 Preface.....                                      | 3  |
| 3 Requirements specification.....                   | 3  |
| 3.1 Functional requirements.....                    | 3  |
| 3.1.1 Registration.....                             | 4  |
| 3.1.2 Login.....                                    | 4  |
| 3.1.3 Adding a project.....                         | 4  |
| 3.1.4 Editing a project.....                        | 5  |
| 3.1.5 Deleting a project.....                       | 5  |
| 3.1.6 Adding a task.....                            | 5  |
| 3.1.7 Editing a task.....                           | 6  |
| 3.1.8 Deleting a task.....                          | 6  |
| 3.1.9 Changing the language.....                    | 6  |
| 3.1.10 Changing the theme.....                      | 7  |
| 3.2 Non-functional requirements.....                | 7  |
| 3.2.1 Layout of the user interface.....             | 7  |
| 3.2.2 Handling errors.....                          | 10 |
| 3.3 Features to be executed.....                    | 10 |
| 3.3.1 Remember login.....                           | 10 |
| 3.3.2 Editing profile.....                          | 10 |
| 3.3.3 Order new password for forgotten one.....     | 10 |
| 3.3.4 Arranging projects.....                       | 10 |
| 3.3.5 Remember selected project between logins..... | 10 |
| 3.3.6 Highlight tasks that are late.....            | 11 |
| 3.3.7 Responsive interface.....                     | 11 |
| 4 User documentation.....                           | 11 |
| 5 Tests.....  | 11 |

## 1 Modification log

| Date       | Version | Comment                                     |
|------------|---------|---|
| 2015-07-15 | 0.0.1   | Document created                            |
| 2015-07-16 | 0.1.0   | Preface was written                         |
| 2015-07-17 | 0.2.0   | Started writing requirements specification  |
| 2015-07-22 | 0.2.1   | Finished writing requirements specification |
| 2015-10-03 | 0.2.2   | Updated requirements specification          |

## 2 Preface

I created this project in order to update my skills as a programmer. This project also serves as a demonstration of my abilities in said area. Original idea for this project comes from school project at a project management class. At the end of the class we had created a simple todo-list program.

The original project was a group project at Laurea university of applied sciences. The group had 20-30 persons in it. The project was made in class where we practised working in projects and in groups. During this class we designed and programmed the program. The original project was executed in Vaadin 6.

I chose this project because todo-list is relatively simple exercise. It is also an excellent project to use as a demonstration of ones abilities. I have done this exercise earlier with JSP and PHP. The second reason for choosing this project is that, since the original project, Vaadin 6 has been updated to Vaadin 7. Because of this it was necessary for me to update my skills, especially considering that changes between the versions are quite big.

The skills I wish to demonstrate in this project are programming, my ability to write in both English and Finnish, as well as test automation with Selenium IDE, to some extent. Through this project I wish to demonstrate my ability to program in general as well as in Java and Vaadin. In this document and the Finnish document I wish to demonstrate my ability to write in both English and Finnish. In these documents I also wish to explain some solutions I've made in the code, as well as the Selenium IDE tests I've wrote.

## 3 Requirements specification

Functions of the program, requirements for those functions and future functions will be explained in this chapter.

### 3.1 Functional requirements

In this chapter the functions and requirements for those functions will be explained.

### 3.1.1 Registration

The user will register to the system via registration form. Following information will be asked in the form:

- Username
- Email
- Password
- Confirm password

In order to successfully register user to the system, following requirements must be fulfilled:

- All fields are required
- Username mustn't be in use in the system already
- The username can be an email, but it can't be different than the one in username field
- The email must be valid
- Email mustn't be in use in the system already
- Password has to be at least eight (8) characters long
- The values in fields "Password" and "Confirm password" must be same

After the registration is complete, the system will perform following tasks:

- A project and two tasks will be created automatically
- User will be logged in automatically

### 3.1.2 Login

User will login to the system via login form. Following information will be asked in the form:

- Username
- Password

In order to successfully login user to the system, following requirements must be fulfilled:

- Both username and password must be correct

After the login is complete, the system will perform following tasks:

- The users first project will be selected and its information will be shown
  - If user doesn't have any projects, no projects -message will be shown

### 3.1.3 Adding a project

By pressing "Add project" button, user can begin adding a new project. After pressing the button, a window with add project form will open. The following fields are in the form:

- Project name
- Project description

In order to successfully add a new project, the following requirements must be fulfilled:

- Field “Project name” must be filled

After project is added, the system will perform following tasks:

- The added project will be selected and its information will be shown

#### 3.1.4 Editing a project

By pressing edit project button, next to the projects name, user can begin editing the project. After pressing the button, a window with edit project form will open. The following fields are in the form:

- Project name
- Project description

In order to successfully add a new project, the following requirements must be fulfilled:

- Field “Project name” must be filled

#### 3.1.5 Deleting a project

By pressing delete project button, next to the edit project button, user can begin deleting the project. After pressing the button, a window with confirmation message will open. In this message the user is asked if he is sure that the project should be deleted. If the user answers yes, the system will perform following tasks:

- All tasks for the project will be deleted
- The project will be deleted

#### 3.1.6 Adding a task

By pressing “Add task” button, user can begin adding a new task. After pressing the button, a window with add task form will open. The following fields are in the form:

- Task name
- Due date
- Priority
- Status
- Task description

In order to successfully add a new task, the following requirements must be fulfilled:

- Field “Task name” must be filled
- Due date must be in format yyyy-MM-dd

### 3.1.7 Editing a task

By pressing edit task button, at the end of the task row in the task list, user can begin editing the task. After pressing the button, a window with edit task form will open. The following fields are in the form:

- Task name
- Due date
- Priority
- Status
- Task description

In order to successfully edit the task, the following requirements must be fulfilled:

- Field “Task name” must be filled
- Due date must be in format yyyy-MM-dd

### 3.1.8 Deleting a task

By pressing delete task button, next to the edit task button, user can begin deleting the task. After pressing the button, a window with confirmation message will open. In this message the user is asked if he is sure that the task should be deleted. If the user answers yes, the system will delete the task.

### 3.1.9 Changing the language

The user must be able to change the language of the user interface. The language will be changed via drop-down menu.

Available languages:

- Finnish
- English

Other requirements:

- On the first visit to the site the system will decide the language based on the location of the user
  - If local language doesn't match any supported language, the language will be set to English
- The system will remember the language user selected even without login

- This will be executed with cookies
- The name of the language must be in the own language despite the language of the user interface
  - Meaning that languages at the menu are always Suomi and English
- Adding a new language must be easy in case of further development
  - This will be executed with language files

#### 3.1.10 Changing the theme

The user must be able to change the theme of the user interface. The theme will be changed via drop-down menu. The following requirements must be fulfilled:

- The names of the themes must be always in English
- The system will remember the theme user selected even without login
  - This will be executed with cookies

### 3.2 Non-functional requirements

In this chapter the non-functional requirement will be explained. This means the layout of the user interface etc.

#### 3.2.1 Layout of the user interface

The user interface is divided into three parts: header, content and footer (Image 1). Header and footer will always stay the same, whether user is logged in or not.

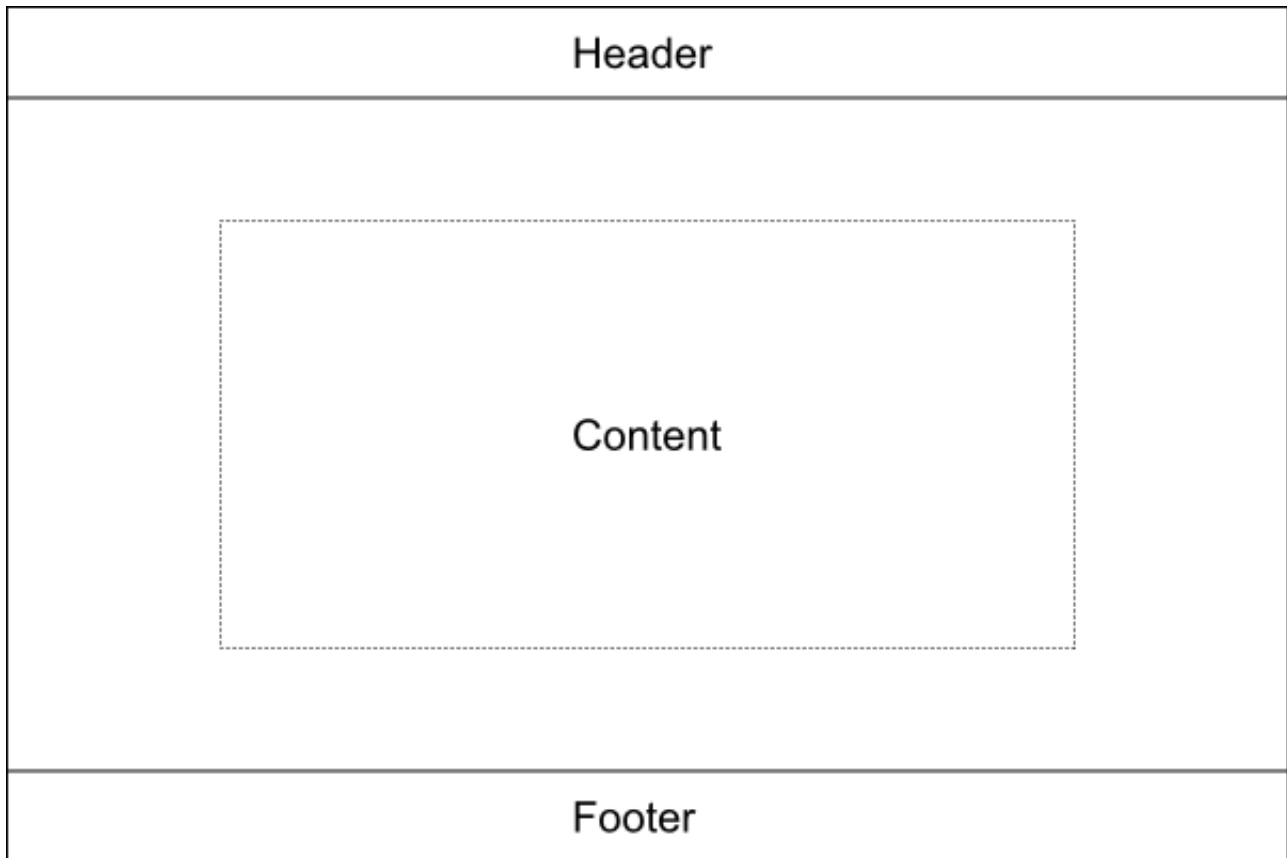


Image 1: Basic layout

The user greeting message, language selector and team selector are in the header (Image 2). All objects are at the right side of the header, as shown in the picture.

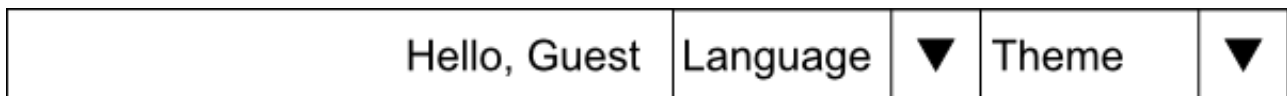


Image 2: Header

The latest editing date and link to the source code and documentation are in the footer (Image 3). Latest editing date will be positioned at the centre of the footer at all situations. The link is located at the right side of the footer.

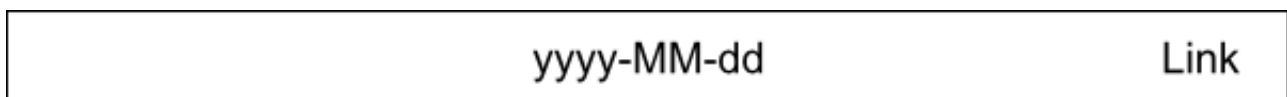


Image 3: Footer

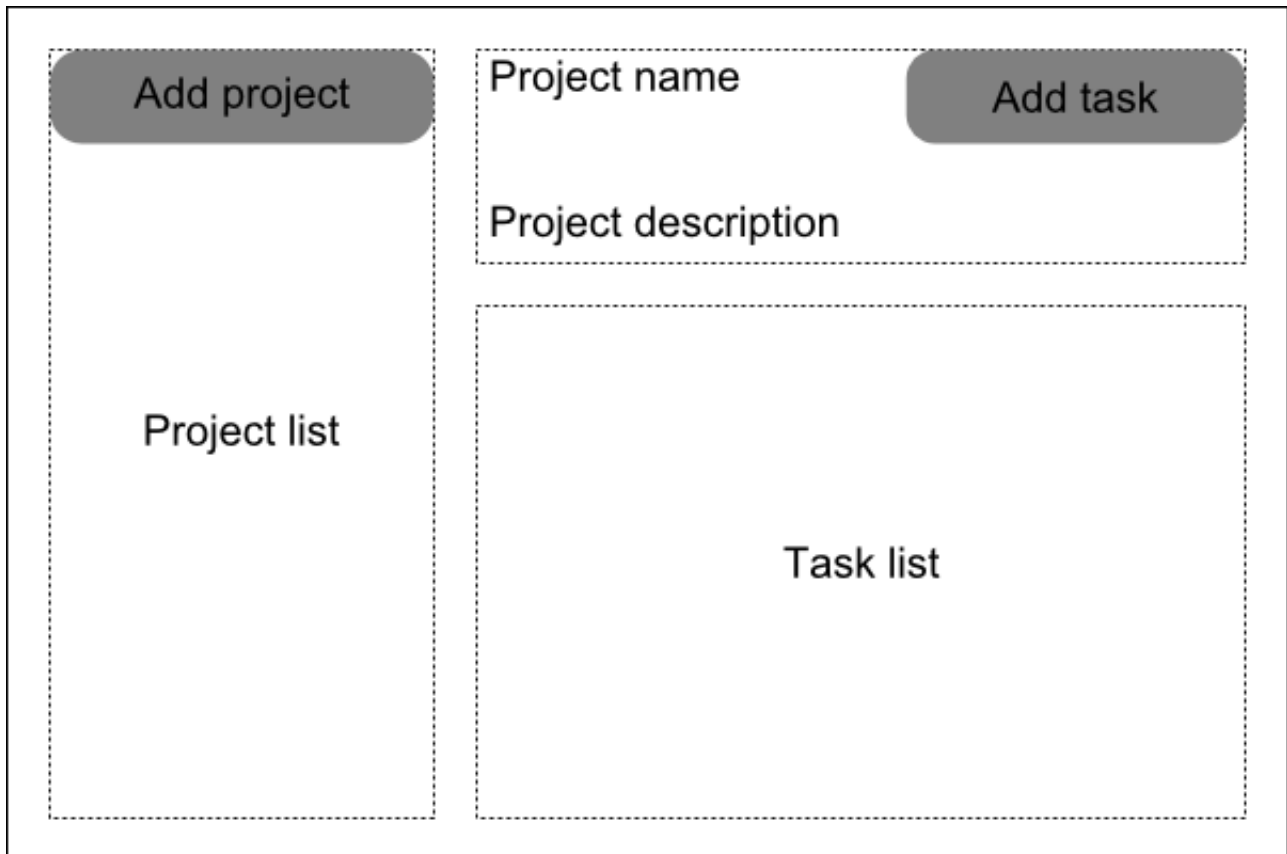
Either main view of form view is located at content. Main view is shown only if the user is logged in. Otherwise a form view is shown.

The main view consist of project list, information of selected project and task list (Image 4). Project list is located at left side of the main view. All the projects user have are listed on the project list. Next to projects name are two (2) buttons; edit project button and delete project button. If the user doesn't have any projects, a message informing this will be shown in the project list. The "Add project" -button will always be on top of the project list.



On the right side of the project list there is project information. This contains project name, project description as well as “Add task” -button. If the user doesn't have any projects, the project information is not shown to the user.

The task list is located below project information. All task to the project are listed in it. Tasks description is not shown in the task list. Each task has its own edit task -button and delete task -button at the end of the task row. If the user doesn't have any projects, task list won't be shown. If the project doesn't have any tasks, a message informing this will be shown in the task list.



The diagram illustrates the main view layout, which is divided into two main vertical sections. The left section contains a large dashed box labeled "Project list" and a grey button labeled "Add project" at the top. The right section contains a dashed box for project information at the top, which includes a "Project name" label, a "Project description" label, and a grey button labeled "Add task". Below the project information box is a large dashed box labeled "Task list".

Image 4: Main view

Form view contains either login view or register view. In this view the fields of the correct form will be shown (More detailed explanations can be found in chapters 3.1.1 and 3.1.2). Below the fields there are a button and a link. The button will send the form. The link, next to the button, will direct the user to the other form view. (Image 5)

The diagram illustrates a form layout within a rectangular container. On the left side, there are two labels: 'Information 1:' and 'Information 2:'. To the right of 'Information 1:' is a rectangular input field labeled 'Field 1'. Similarly, to the right of 'Information 2:' is a rectangular input field labeled 'Field 2'. Below these fields, there are two elements: a dark gray rounded rectangle labeled 'Button' and a plain text label 'Link'.

Image 5: Form view

### 3.2.2 Handling errors

If an error happens, an error message will be shown to the user. The error message will be shown in a notification window that overlays user interface. The message will tell the user what went wrong.

## 3.3 Features to be executed

The features that are not in this version but will be added later are explained in this chapter. Only the main idea of the function is explained and further details will be added when requirements are specified.

### 3.3.1 Remember login

System will remember user login for 30 days. The timer will be set to 30 days every time user visits the site. This feature will be executed with cookies.

### 3.3.2 Editing profile

The user can edit his profile. The profile includes: username, email address and password.

### 3.3.3 Order new password for forgotten one

If the user has forgotten his password, the user can ask a new password to be sent to his email address.

### 3.3.4 Arranging projects

The user can arrange his project to preferred order.

### 3.3.5 Remember selected project between logins

The system will remember which project was selected last time. The user has to toggle this feature at his profile, since this feature is not active by default.

### 3.3.6 Highlight tasks that are late

If a task is late, it will be highlighted at the task list.

### 3.3.7 Responsive interface

The layout of the site is currently designed for a window that is at least 1000 pixels wide. However, this is not realistic, so a responsive layout will be made on a later date. This means that elements size and location will depend on the size of the window.

## 4 User documentation

Functions and features of the executed program will be explained in this chapter.

TODO

## 5 Tests

Written Selenium IDE tests and their usage will be explained in this chapter.

TODO