

Requirement Specification

Web Application

Daniel Blixt
Mårten Isaksson
Daniel Nilsson

Introduction

The purpose of this document is to give a detailed description of the requirements for the web application "Web Application".

The "Web Application" is a web application where dog owners can meet and interact. It is however not a complete social media application for dog owners. Instead it is focuses on people with dogs creating events where others can join and then meetings take place in real life. An example of such an event could be a planned walk with the dog, or maybe play in the park, other users of the web application will then be able to join the event.

Only relevant events will be displayed in users feed and this is filtered on the users location, each new user will thus have to specify where they live in their profile.

Users

There will be two types of users on this system, administrators and regular users.

Regular Users

This is the standard type of user which all new registered accounts will belong to. If a user do not have an account he or she will be asked to create an account on the web application.

Regular users will be able to view nearby event, they will also be able to create and join event, as well as comment on events they have joined or created. Regular users will also be able to edit and delete their own events on comments.

Administrators

Administrators are basically regular users with extended privileges, this includes editing and deleting other users events and comments as well as disabling accounts of misbehaving users.

Language

The user interface of the website will be in English, but there is no requirements on language of user submitted content such as events and comments.

System Requirement

System Description

This system will consist of three parts, a website for user interaction, a web server and a database. The website will be the central point where the users can interact with each other while the web server will handle all the information for the system. User contributed data should be saved and stored in a database.

The website will be used for hosting a personal profile of the user where he or she can enter information about themselves and their dogs. This profile will then be used for posting or joining “dog” events such as walks or playtimes. On the website there will also be a location based feed for each user where other events from other users will be posted.

For the location filtered feed the location of the user must be known, it will therefore be possible for the user to select their location on a map, the location should also be automatically suggested if geolocation is supported by the browser.

The web application should run on IBM Bluemix. Java should be used as the server side language along with Java Server Faces. The client side should be developed using HTML, CSS and Javascript.

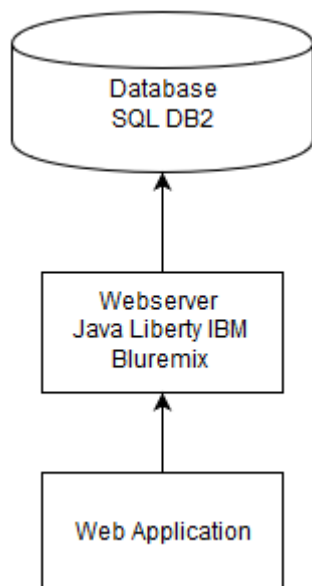


Figure 1 System overview.

User interface

A user of the application should see a login page when he or she opens the website. If the user does not already have an account he or she will be asked to create one to be able to use the application. To create an account three things should be required, the name of the user, his or her email address along with a password for use on the website.

Web Application

Not a user? Please register!

Register now!

User name

Password

Login

Figure 2 Login page of website.

If the user already have an account he or she can use it to login to the website. Each user should have their own profile page. The profile should include personal information such as name, description, location and dogs. If location is not set there will be no constraints on the feed and the latest events from everywhere should be shown.

When this is done they will get access to the main page which is a feed and a menu of different actions such as creating/deleting events, join other events or list events which they have either created or joined.

When a specific event is chosen the user will be taken to that events information page. Here the user can see the information that the creator of the event has posted. They should also be able to send a request to join the event, which the creator will have to approve. If they are approved by the creator they will get access to the event and should then be able to read and post comments on the event page.

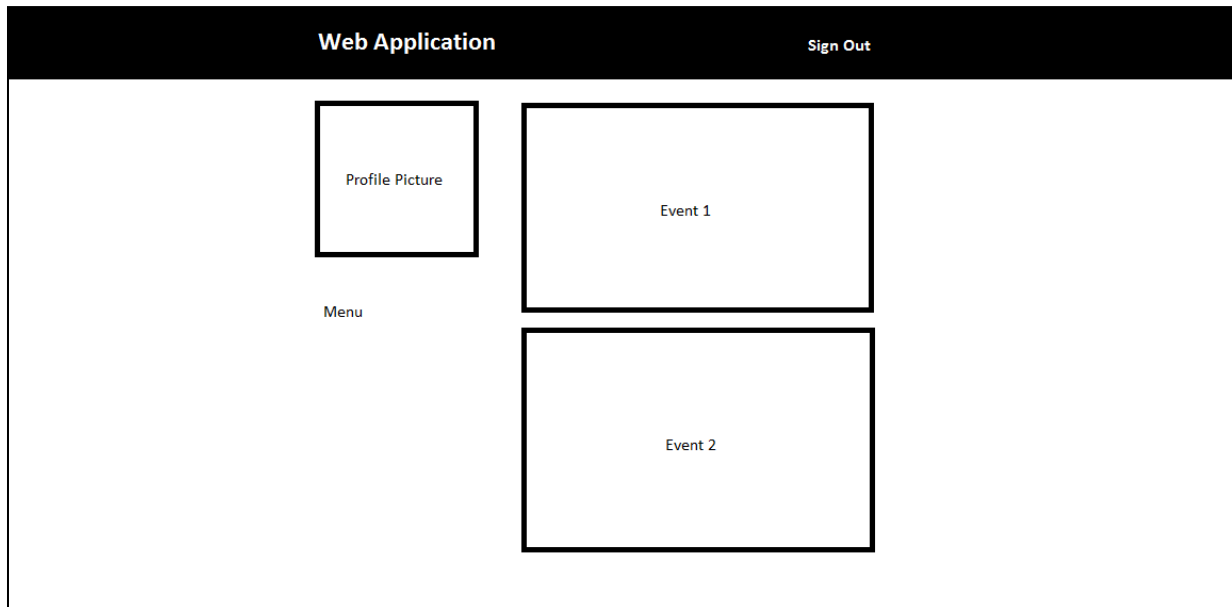


Figure 3 The main page which a signed in user will see.

Functional Requirements

User Class - Regular Users

FR1.1

Title: Register account

Description: A user should be able to visit the website and register an account if they do not already have one. The user must provide their name, email address and password. Only one account for a given email address is allowed.

FR1.2

Title: Sign in

Description: Given that a user has registered an account, then the user should be able to login into the web application and access their feed.

Dependency: 1.1

FR1.3

Title: Forgot password

Description: Given that a user has registered an account, the the user should be able to request a new password if it is forgotten.

Dependency: 1.1

FR1.4

Title: Profile page

Description: A registered user should have their own profile page which they should be able to edit. This included their name, a profile picture, location, gender, description and dogs.

Dependency: 1.1, 1.2

FR1.5

Title: Dogs

Description: A registered and signed in user should be able to add their dogs on their profile page. When registering a dog, information must be entered. This included name, breed, age, gender and a picture.

A user should also be able to edit information about an already added dog as well as removing it completely.

Dependency: 1.1, 1.2, 1.4

FR1.6

Title: Create event

Description: A registered and signed in user should be able create a new event, when creating an event information must be supplied, this includes title, description, date, time and location.

Dependency: 1.1, 1.2

FR1.7

Title: View feed

Description: A registered and signed in user should see nearby event in his/her feed. The events in the feed should be filtered on the event location and the location set by the user in his/her profile.

Dependency: 1.1, 1.2, 1.4, 1.6

FR1.8

Title: Edit event

Description: A registered and signed in user should be able to edit his/her own events. Title, description, time and location should all be possible to edit.

Dependency: 1.1, 1.2, 1.6

FR1.9

Title: Cancel event

Description: A registered and signed in user should be able to cancel his/her own events.

Dependency: 1.1, 1.2, 1.6

FR1.10

Title: Request join event

Description: A registered and signed in user should be able to request to join other users events. The user should not get access to the event until the request has been approved.

Dependency: 1.1, 1.2, 1.6

FR1.11

Title: Approve join request

Description: A registered and signed in user should be able to approve or decline request from other users to join his/her event.

Dependency: 1.1, 1.2, 1.6, 1.10

FR1.12

Title: Comment on event

Description: A registered and signed in user that has joined an event should be able to post comments on that event.

Dependency: 1.1, 1.2, 1.6, 1.10, 1.11

FR1.13

Title: Leave event

Description: A registered and signed in that has joined another users event should be able to leave the event if desired.

Dependency: 1.1, 1.2, 1.6, 1.10

FR1.14

Title: Notifications

Description: A registered and signed in users should see notifications if things have happened. The following event should trigger a notification.

- A user has requested to join your event.
- Join event request has been approved or declined.
- Joined event has been updated.
- Joined event has been cancelled.
- Comment posted on joined or own event.

Notifications should be marked as read once they have been viewed.

Dependency: 1.1, 1.2

User Class - Administrators

Administrators should be able to do everything a regular user can do along with a few other things.

FR2.1

Title: View all events

Description: A signed in administrator should have access to all events and should not have to join an event to be able to view it.

FR2.2

Title: Delete event

Description: A signed in administrator should be able to delete events if desired.

Dependency: 2.1

FR2.3

Title: Delete comment

Description: A signed in administrator should be able to delete comments on events if desired.

Dependency: 2.1

FR2.4

Title: Suspend user

Description: A signed in administrator should be able to suspend a user if desired so that he/she will not be able to use their account anymore.

Appendix

Domain Model

