Advent Calendar

Software Requirements Specification

Version 1

04 December 2014

Florian Geier

Lachezar Asparuhov

Jonas Mattsson

Prepared for

2DV012 Web 2.0 Web programming

Instructor: Tobias Gidlund

Fall 2014

**Table of Contents**

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

2. General Description 3

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Characteristics 3

2.4 General Constraints 4

2.5 Assumptions and Dependencies 4

3. Specific Requirements 4

3.1 External Interface Requirements 4

3.1.1 User Interfaces 4

3.2 Functional Requirements 6

3.2.1 Log in 6

3.2.2 Register 6

3.2.3 View calendars 7

3.2.4 Create calendars 7

3.3 Use Cases 8

3.3.1 User 8

3.3.2 Admin 8

3.4 Classes / Objects 9

3.4.1 Users 9

3.4.2 Calendars 9

3.4.3 Days 10

3.5 Non-Functional Requirements 10

3.5.1 Performance 10

3.5.2 Reliability 10

3.5.3 Availability 10

3.5.4 Security 10

3.5.5 Usability 10

3.7 Design Constraints 11

3.8 Logical Database Requirements 11

4. Analysis Models 11

4.1 Sequence Diagrams 11

# 1. Introduction

## 1.1 Purpose

*The purpose of this document is to give a detailed description of the requirements for the “Advend Calendar” website. It will illustrate the purpose and complete declaration for the*

*development of the system.*

## 1.2 Scope

*The “Advent calendar” website is a website where people can visit a online advent calendar, have access to personalized and public calendars and the admin can publish and assign customized calendars to users.*

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| User | Someone who interacts with the website |
| Admin | A authorized user who has user rights to edit/ create calendars and assign them |
| Calendar | A calendar is a collection of days and a background picture |
| Day/window | Each day contains text and or pictures that the admin can create and edit |

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

## 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

## 2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

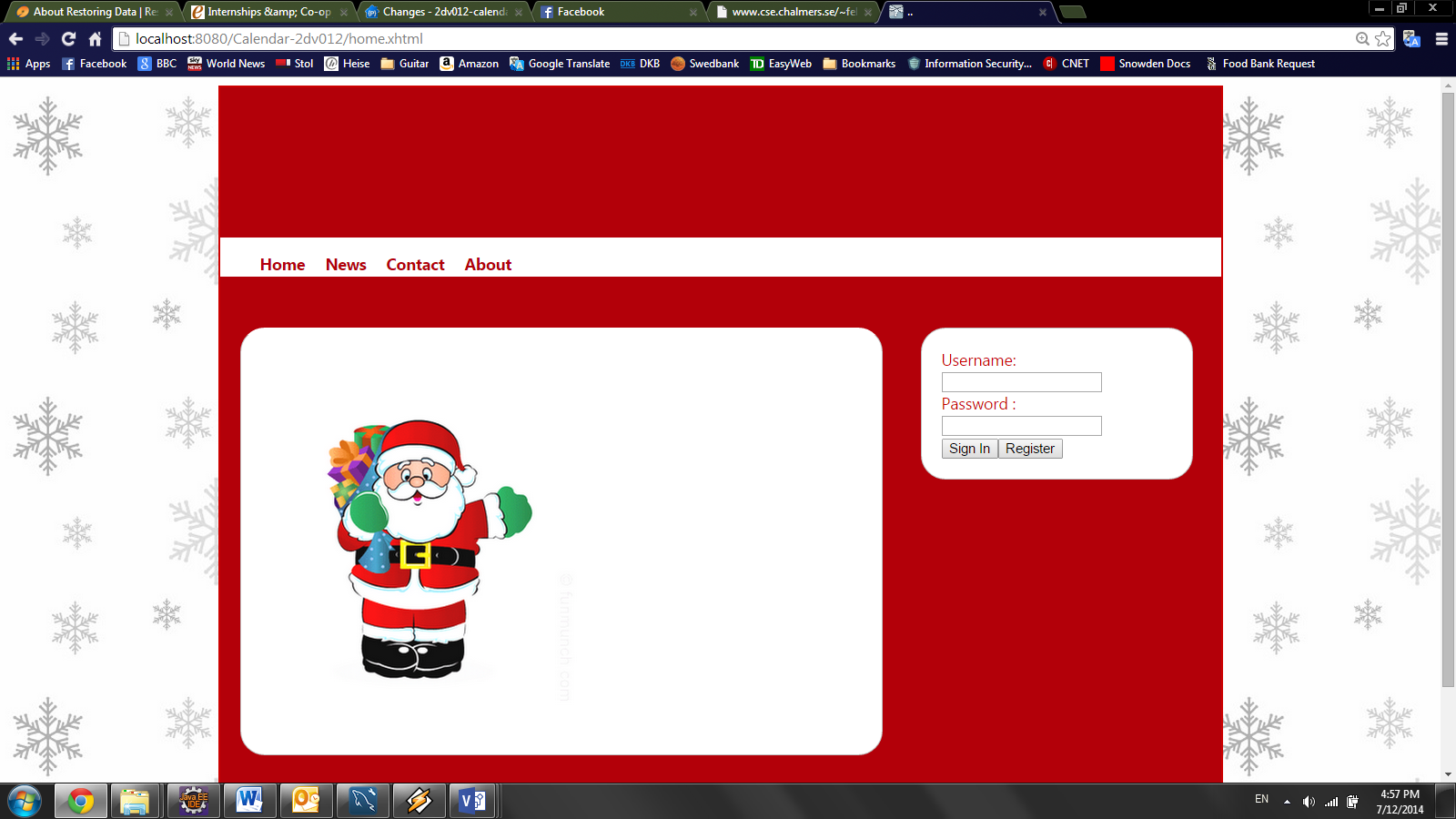
# 

# 3. Specific Requirements

## 3.1 External Interface Requirements

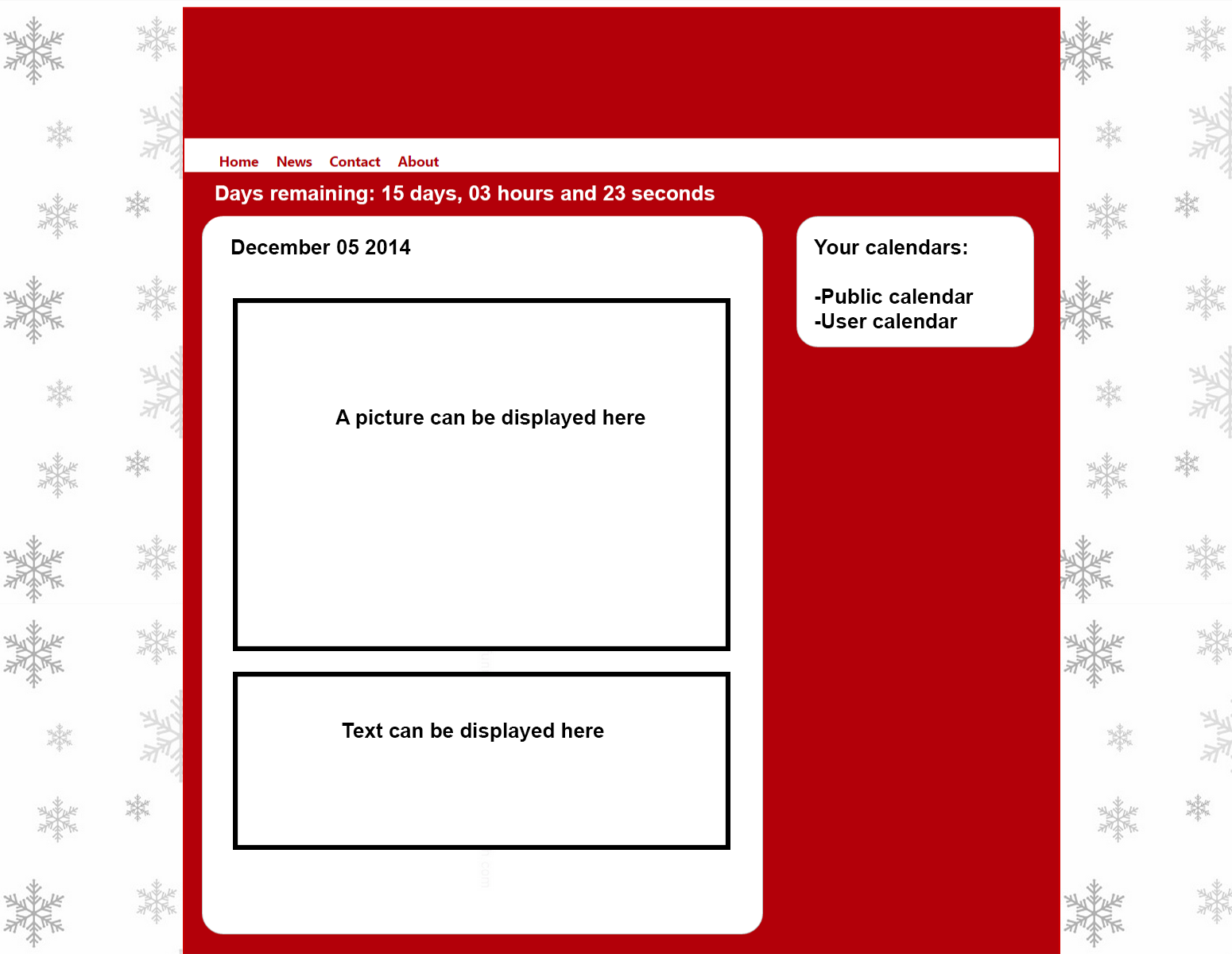
### 3.1.1 User Interfaces

This is how the home page will look like.



After the login the right sidebar will disappear and a list of calendars displayed instead. By selecting one it will be displayed in the center. By default the first calendar will be displayed.





## 3.2 Functional Requirements

### 

### 3.2.1 Log in

3.2.1.1 Introduction

The log in system will authenticate users in order to display user specific calendars. It will also distinguish users from admins.

3.2.1.2 Inputs

Username

Password

3.2.1.3 Processing

The given username will be looked up in the database, if available the password will be compared. If both username and password match the user will be redirected to the following page.

3.2.1.4 Outputs

Redirection to either a success or an “unsuccessful authentication” page.

3.2.1.5 Error Handling

Blank username or passwords will be handled with an on page warning.

### 3.2.2 Register

3.2.2.1 Introduction

Users can register themselves to the website.

3.2.2.2 Inputs

Username

Password

3.2.2.3 Processing

The given username will be looked up in the database, if not available yet the password will be Saved along with the username.

3.2.2.4 Outputs

Redirection to either a success page or an error message for duplicate username.

3.2.2.5 Error Handling

Blank username or passwords will be handled with an on page warning.

### 3.2.3 View calendars

3.2.3.1 Introduction

Users can view either the public calendar or a user specific calendar if existing.

3.2.3.2 Inputs

User clicks on a calendar from a list shown in the sidebar.

3.2.3.3 Processing

The chosen calendar background picture will be shown in the center section. It will contain numbers as links where users can reach days.

Each day is only accessible if the day already has passed or is today.

3.2.3.4 Outputs

Display the calendar.

### 3.2.4 Create calendars

3.2.4.1 Introduction

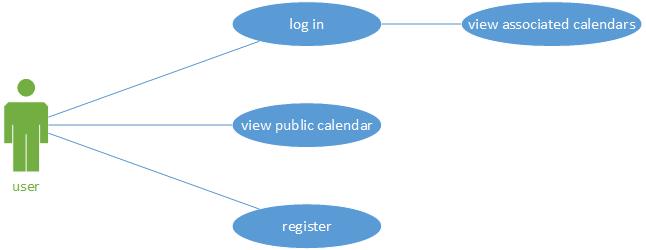
Admins can create calendars and associate them to users.

3.2.4.2 Inputs

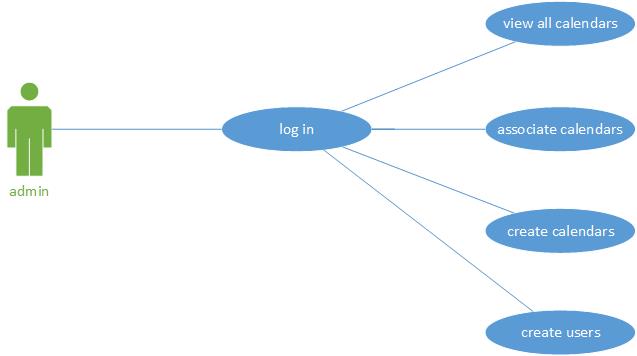
Admin clicks on create calendar and chooses a background picture. He also creates dates in descending order starting from the 24th December.

## 3.3 Use Cases

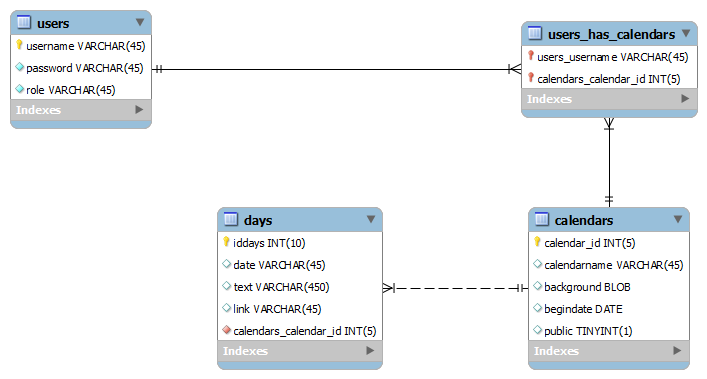
### 3.3.1 User



### 3.3.2 Admin



## 3.4 Classes / Objects



### 3.4.1 Users

3.4.1.1 Attributes

* Username: a unique identifier of the user
* Password the users password, MD5 hashed
* Role distinguishes users from admins.

3.4.1.2 Functions

3.2.1

3.2.2

3.2.3

### 3.4.2 Calendars

3.4.2.1 Attributes

* Calendar\_id: a unique identifier of the calendar
* Calendarname: a user friendly name of the calendar
* background the background picture
* begindate the date of the first window
* public Boolean thee indicates if the calendar is visible public

3.4.2.2 Functions

3.2.3

3.2.4

### 3.4.3 Days

3.4.3.1 Attributes

* iddays: a unique identifier of the day
* date: the date of this day
* text text to be displayed when opening the day
* link link to a picture displayed when opening the day
* calendars\_calendar\_id the id of the calendar associated

3.4.3.2 Functions

3.2.3

3.2.4

## 3.5 Non-Functional Requirements

### 

### 3.5.1 Performance

The performance of the website is depending on the host’s and the user connection.

### 3.5.2 Reliability

The reliability of the website is depending of the host. We did not yet decide on a host.

### 3.5.3 Availability

The availability of the website is depending of the host. We did not yet decide on a host.

### 3.5.4 Security

There is no need for high security systems because neither money transactions take place nor sensitive data will be stored on our website. Anyways all passwords are not saved clear text but are saved in a hashed form.

### 

### 3.5.5 Usability

Our website will be developed for browsers such as Mozilla Firefox, Internet explorer and Chrome. It is not optimized for tablets or smartphones. We will use HTML5 and CSS3 for our Website.

## 3.7 Design Constraints

There are not any recognized design constrains that effect this project. An obstacle that can be taken under consideration is the performance of the web server (Wildfly 8.x) used for this project, but this does not affect the taken design decisions.

## 3.8 Logical Database Requirements

A MySQL database is used. The schema is called **mydb** the username for test purposes is **root** with password **test**. This will be changed in later versions.