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 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 Users 4

3.4.2 Calendars 4

3.4.3 Days 5

3.5 Non-Functional Requirements 5

3.5.1 Performance 5

3.5.2 Reliability 5

3.5.3 Availability 5

3.5.4 Security 5

3.5.5 Maintainability 5

3.5.6 Portability 5

3.6 Inverse Requirements 5

3.7 Design Constraints 5

3.8 Logical Database Requirements 5

3.9 Other Requirements 5

4. Analysis Models 6

4.1 Sequence Diagrams 6

4.3 Data Flow Diagrams (DFD) 6

4.2 State-Transition Diagrams (STD) 6

5. Change Management Process 6

A. Appendices 6

A.1 Appendix 1 6

A.2 Appendix 2 6

# 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

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

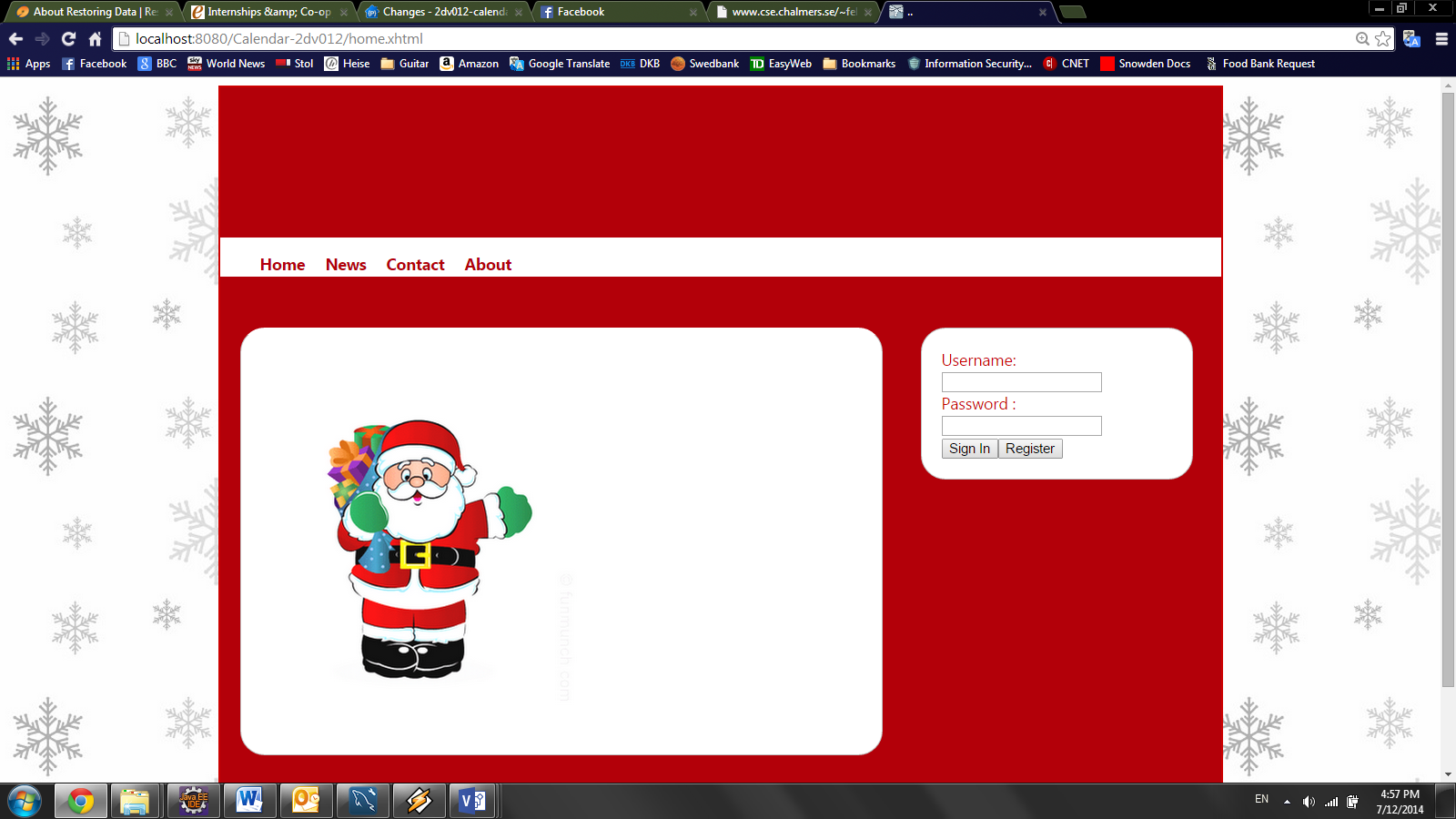
Each requirement in this section should be:

* Correct
* Traceable (both forward and backward to prior/future artifacts)
* Unambiguous
* Verifiable (i.e., testable)
* Prioritized (with respect to importance and/or stability)
* Complete
* Consistent
* Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## 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.

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

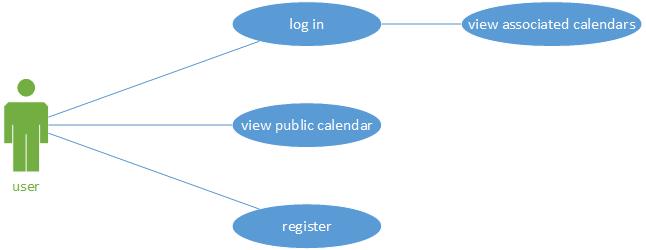
3.2.1.5 Error Handling

### 3.2.2 <Functional Requirement or Feature #2>

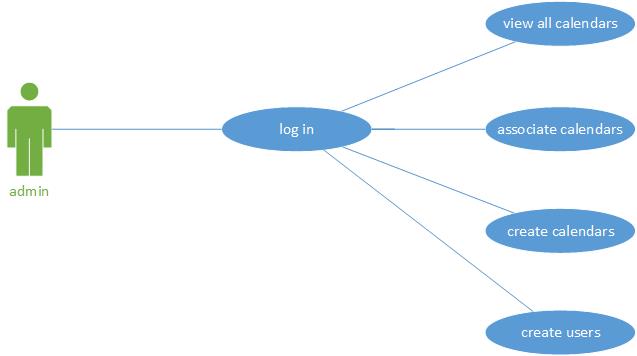
…

## 3.3 Use Cases

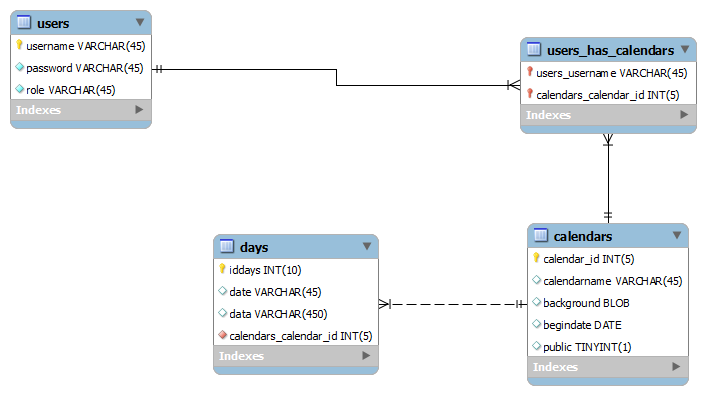
### 3.3.1 User



### 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

<Reference to functional requirements and/or use cases>

### 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

<Reference to functional requirements and/or use cases>

### 3.4.3 Days

3.4.3.1 Attributes

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

3.4.3.2 Functions

<Reference to functional requirements and/or use cases>

## 3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 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

All passwords are not saved clear text but are saved in a hashed form.

## 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.

## 3.9 Other Requirements

Catchall section for any additional requirements.

# 4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## 4.1 Sequence Diagrams

## 4.3 Data Flow Diagrams (DFD)

## 4.2 State-Transition Diagrams (STD)

# 5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2