IT University of Copenhagen

BDSA 2014

Calendar System
System Design Document

Assignment 39

Anders Wind Steffensen - awis@itu.dk
Christopher Blundell - ppma@itu.dk
Pierre Mandas - cnbl@itu.dk

October 2, 2014

Contents

CHAPTER 1

Introduction

1.1 Purpose of the system

The purpose of the Calendar system is to have a standard client-server calendar system like MS Exchange or iCal. The system must provide the user with the ability to save, edit events and be able to retrieve their calendar on any computer with the system installed. The system should furthermore provide the user with multiple kinds of overviews of the events he has planned, and prompt the user with a notification if an event has a notification time.

1.2 Design goals

The goal with the systems usability, is to end up with a product which is user-friendly for all types of users. By applying gestalt laws, the goal is to end up with a clean and simple user interface which is logical for all users, even the unexperienced.
Furthermore it is desired to create a reliable program, preventing users from losing all progress related to unexpected crashes. This will be handled by frequently auto-saving data and synchronizing with the data storage. Furthermore every time the user commits anything, for example a new calendar entry, it will also be saved immediately. Force restarting should not be exceptable and most exceptions must be caught and handled runtime. By doing the abovementioned, data loss will be kept to a minimum by limiting it to users current activity, should a failure occur.
The goal performance-wise is to be able to handle a large number of events daily, without setting a noticable strain on the program. Heavy operations must run in the background, and therefore not disturbing the user while operating the program. The trade-off however will be the loading time of the program. This allows us to load all the initially required data, and prevent long waiting times while operating the system.
When rolling out future updates for the system, a full re-installation of the program will be necessary. This is due to resources allocated to other more desired design goals.
Finally the system will be tested before release, but in a limited way. Key components will be tested, but due to the time frame set for this systems development thorough testing is unattainable. The Calendar is a lightweight system, and stores data on the cloud, so there spacewise a maximum of around 1gb should be achievable. Additionally, in it's current form the CalendarSystem is only runnable on Windows OS

1.3 Definitions, acronyms and abbreviations

None at the time.

1.4 References

None at the time.

1.5 Overview

CHAPTER 2

Current Software Architecture

CHAPTER 3

Proposed Software Architecture

3.1 Overview

3.2 Subsystem decomposition

3.3 Hardware/software mapping

3.4 Persistent data managemen

3.5 Access control and security

3.6 Global software control

3.7 Boundary conditions

CHAPTER 4

Subsystem Services

CHAPTER 5

Glossary