Indhold

1 Introduction

1.1 Purpose of the system

The purpose of the calendar system is to make a calendar system for a workplace, capable of sharing entries and organizing meetings etc. The system is intended to be used in a workplace environment, where people know each other, or at least are acquainted in some way. Thus the program has no need for contact lists and blocking of people, since it is used by people working together, to organize meetings etc.

1.2 Design goals

Usability: The system should be easy to learn.

Fault tolerance: The system should be fault tolerant to loss of connectivity with the calendar server.

The system should use a low amount of bandwidth.

The client part of the system should be compatible with older windows versions.

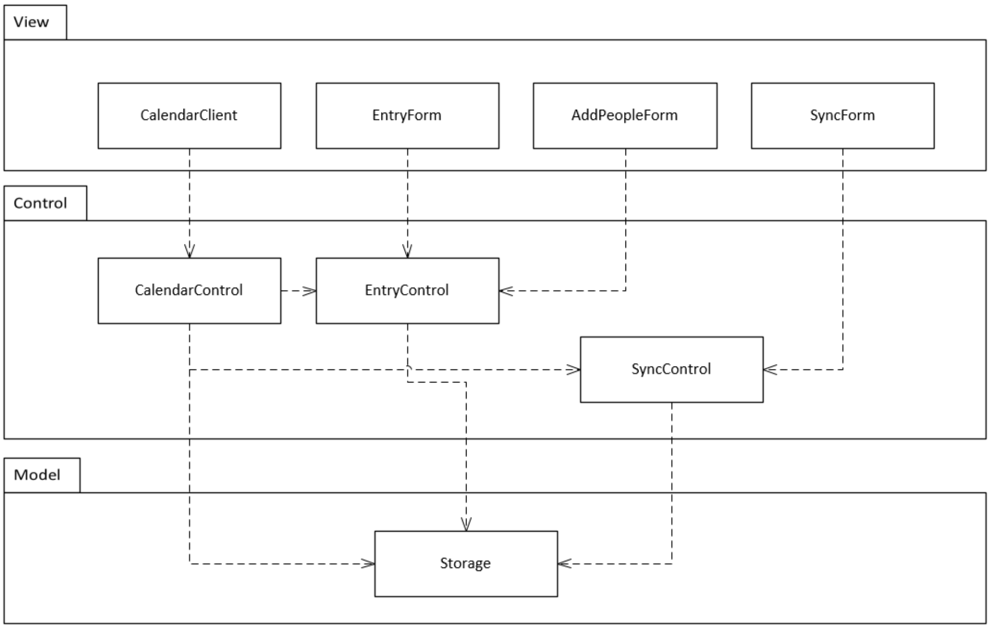
2 Proposed software architecture

2.1 Overview

The following chapter will go through: the chosen architecture patterns and design of the calendar system, the design patterns used to implement it and how the control flow of the program.

2.2 Subsystem decomposition

The figure below shows the subsystem decomposition of the User part of the program.



Figur 1: Subsystems and their decomposition

2.3 Hardware/software mapping

2.4 Persistent data management

The two kinds of objects the Calendar System is meant to store are Users and Entries. The Entries will be available for to create and modify for Users, while the Users can only be created and modified by an administrator. Thus we will need to kinds of access controls[1]. The storage system will provide an interface that enables the usage of three different kinds of storage: Online storage (will be implemented as a relational database), offline storage (will be implemented using serialization) and a test storage with no persistence for testing purposes only.

2.5 Access control and security

There are only two kinds of actors that will be using the Calendar System: User and Administrator. Everyone using the system will have a User account and the administrators of the system will also have an Administrator account in addition to their User account.

2.6 Global software control

2.7 Boundary conditions

[1] see: Access control and security