



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Bc. Jakub Sýkora

Thesis title

Name of the department

Supervisor of the master thesis: Supervisor's Name

Study programme: study programme

Study branch: study branch

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date
Author's signature

Dedication.

Title: Thesis title

Author: Bc. Jakub Sýkora

Department: Name of the department

Supervisor: Supervisor's Name, department

Abstract: Abstract.

Keywords: key words

Contents

Introduction	2
1 Introduction	3
2 Analysis	4
2.1 Calendar Features	4
2.2 SOLID features	6
Conclusion	7
Bibliography	8
List of Figures	9
List of Tables	10
List of Abbreviations	11
A Attachments	12
A.1 First Attachment	12

Introduction

1. Introduction

An example citation required for successful build: Example [20222]

2. Analysis

2.1 Calendar Features

Calendar Events

The essential part of the calendar is its events. Therefore, the user needs to be able to add, remove and modify calendar events. Next to those operations, there is a need to display event data in a structured way to emphasize important information and hide the details. On the other hand, users should not feel restricted by filling many unnecessary fields. Users should always be able to define crucial calendar information - such as start date, event duration, and short text used as a title. In addition, many events occur at some physical or virtual location. Therefore, fields containing location information should be available. As we are creating a calendar application based on SOLID specification with an option to store and query any information as linked data, we could make some event information, such as the physical location of an event, structured and describable by linked data vocabularies. To not decrease usability and allow users to work with the calendar without any linked data knowledge, we need to design an interactive form that would enable users to define information such as an address in a structured way. In the background, such information would be stored as linked data, making them available for further processing by other applications. In the case of address information, an application could access structured address data and make queries such as finding all events occurring in the same city in a given year.

R1 In calendar events, users can define the following structured data that can be consumed and queried by other applications

- Start time and duration of a calendar event
- Optional physical location where an event occurs
- Optional location of an virtual meeting in the form of URL

Next to the data that can be structured, a user should be able to write any other information in a block of unstructured text. File attachments are an additional feature that would increase calendar usability. The calendar application should allow storing arbitrary attachments such as pictures, documents, or recordings.

R2 In calendar events, users can define the following unstructured data

- Optional description text
- Optional file attachments

R3 User can define *recurring events*.

- Multiple future occurrences of single recurring event can be modified in single event update.

Calendar Groups

- R4** Users can group events into calendar groups.
- R5** Users can create and remove calendar groups.
- R6** Calendar events can be assigned from one calendar group to another.

Calendar Views

The application should support displaying calendar events in different calendar views, such as a view of calendar events in a single day or week. Views in different timespans increase the application's usability by hiding non-important events or providing a valuable overview of future events. In the following subsection, we will introduce functional and non-functional requirements that each calendar view should support and then define specific calendar views that should be available in the application.

Shared functionality of calendar views

In this subsection, we define a set of features that each calendar view should offer in the application. Defining such features increases the application's usability by defining expectable behavior shared in each calendar view. As we want to make the calendar application interactive, users should be able to modify calendar events displayed in a calendar view by simply moving the visualization of the event from one timeslot to another. By modifying an event in a calendar view, users can set a different start time for an event or change the duration of an event, as shown in figure [X], where the user *drag and drops* an event to a different location, changing start of the event.

- R7** Users should be able to interactively modify start and duration of events displayed in calendar views.
- R8** Users can create multiple instances of calendars view sets to accomodate specific preferences.
 - User can select which calendar groups to display.

Day view

Week view

Month view

Sharing

- R9** User can create publicly shared calendar group that is readonly
- R10** User can share calendar group only to group of users
 - Calendar group is privately shared and readonly
 - Calendar group is privately shared and modifyable by each user with access

R11 Users can create copy of other calendar groups that are shared with them

R12 Owner of calendar event can add participants to event

- Participants with access to the event can modify participation status

Import or Export of iCalendar Format

R13 Calendar can be exported to iCalendar format

- Publicly shared attachments defined in `ref[]` can be exported

R14 Calendar can be imported from iCalendar format

2.2 SOLID features

Conclusion

Bibliography

Example. Example. *Example*, 20222.

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment