Internet Based Localized Calendar System

Project Proposal

Version 1.0

The Department Calendar Group

~

Duffey, Luke

Joos, Dylan

Rocco, Samuel

~

November 10, 2022

**Table of Contents**

[**Introduction**](#_heading=h.vmz4k7p1msh5) **1**

[1.1 PURPOSE](#_heading=h.1cfsi9lybv4k) 1

[1.2 SCOPE](#_heading=h.o0unqndy76zs) 2

[**Design Model**](#_heading=h.kktvqwhwhr7u) **3**

[2.1. PRODUCT BACKLOG](#_heading=h.61xp4molvzck) 3

[2.3 SPRINT DESCRIPTION](#_heading=h.8fztzdacoxwr) 7

[2.4 SPRINT RETROSPECTIVE](#_heading=h.gqar7nfpo4wh) 7

[2.5 RESULTS](#_heading=h.yt65eervslw) 8

[2.6 DISTRIBUTION OF WORK](#_heading=h.uuw7wez5e29k) 9

[**Source Code**](#_heading=h.2021lbysnjvq) **10**

# Introduction

## **1.1 PURPOSE**

Our goal is to make a web-based scheduling system designed for a company or other organization, such as a college campus. Other goals we plan to achieve include having times that can be scheduled and locked in place for others in a group to see. A user will have the ability to make groups for certain users that can be notified about scheduling times. A system designed for users to have administrative properties on groups. The ability for administrators to assign users different modes of access, such as editor mode or viewing mode. Finally, the ability to link calendars from separate organizations to connect all calendars to one singular application.

## **1.2 SCOPE**

The system is designed for keeping and regulating schedules for a group or organization. Three major inputs will be accounted for. An input to create, add, or import a new schedule will be utilized by the user to start the calendar. Along with an input to share calendars or specific schedules to other users with the ability to assign an administrator. And an input to choose and view your calendar. Time management algorithms will be used to manage multiple schedules for the user to have a clear understanding of scheduled times with different calendar formats to best suit the needs of the user. Formats will include a monthly, a weekly, and a daily calendar depending on the specific need of the user. The application ability to link third-party calendar applications for a unified time managing experience, where users can keep up with a generalized overview of everything happening at once, as the application will understand the formatting and add each scheduled time from a third-party source to a single-glance display.

# Design Model

## **2.1. PRODUCT BACKLOG**

User stories:

UC5 - view calendar: - Priority 1, users can't do anything with a calendar if they can’t see it. Estimate - Small, this should be the easy part there are more than enough resources online to assist.

*“A user wants to check what meetings they have that day. They can open and view their main calendar to show all events for that day.”*

UC4 - add and remove personal events: - Priority 2, It is important that users are able to interact with the calendar, lest it be little more than a JPeg. Estimate - Medium, Requires the front end and the backend implication, which may take more time.

“*A user may have a party coming up. They want to inform others in their group that they will be unavailable at that time. So, they add their event to the group calendar so they may not be scheduled for one on one meetings at those times. Similarly, if that party were to be canceled, then they can remove it from their calendar to notify the group that they are available again*”

UC7 - Create group - Priority 3, if groups are not able to be made, many other use cases are not possible so this should be done first. Estimate - Medium, Requires the front end and the backend implication, which may take more time.

“*A user wants to make a group to make scheduling events for their friends to hang out easier.”*

UC1 - add and remove group events: - Priority 4, A group must exist before adding groups, but a group does not need more than one person to have events, as events like sing up drives are meant to get people into the group. Estimate - Small, may be as simple as a toggle, but will be based on the add and remove personal events use case, so implantation should be simple.

“*An Administrator needs to schedule a new meeting for a club that they are a part of. The Administrator can then use the calendar program to check the availability of other users in their group and add a meeting to the group calendar. Similarly, if a meeting needs to be canceled they may use the program to remove that event from that group’s calendar.* “

UC3 - add and remove users: Priority 5, people need to be in a group for it to really exist. Estimate - Medium, Requires the front end and the backend implication, which may take more time.

“*An Administrator may need to add new people to the group’s calendar. This Administrator can then add as many people as is necessary to the group calendar.*”

UC2 - grant abilities to users: Priority 6, It is more important for the group functionality to work before more minor things. Estimate - Small, Can be built on the back of add and remove users, so should not take that long.

“*An Administrator feels the need to have additional administrators to manage meetings in a workplace. She may add new Administrators from the existing users in the group. Giving them all the privileges an Administrator may have.*”

UC6 - add 3rd party calendar: Priority 7, it is little more than a quality of life feature, It may also have the most complexity depending how other calendar’s export their data. Estimate - Medium, Basic implementation should be easy, thoroughly testing it though will not be.

*“A user may have a calendar using another software. To ease transition to new software they may add all of the events on a 3rd party software. They will be imported and assigned to the same dates and times as the previous program.*”

**2.2 SPRINT BACKLOG**

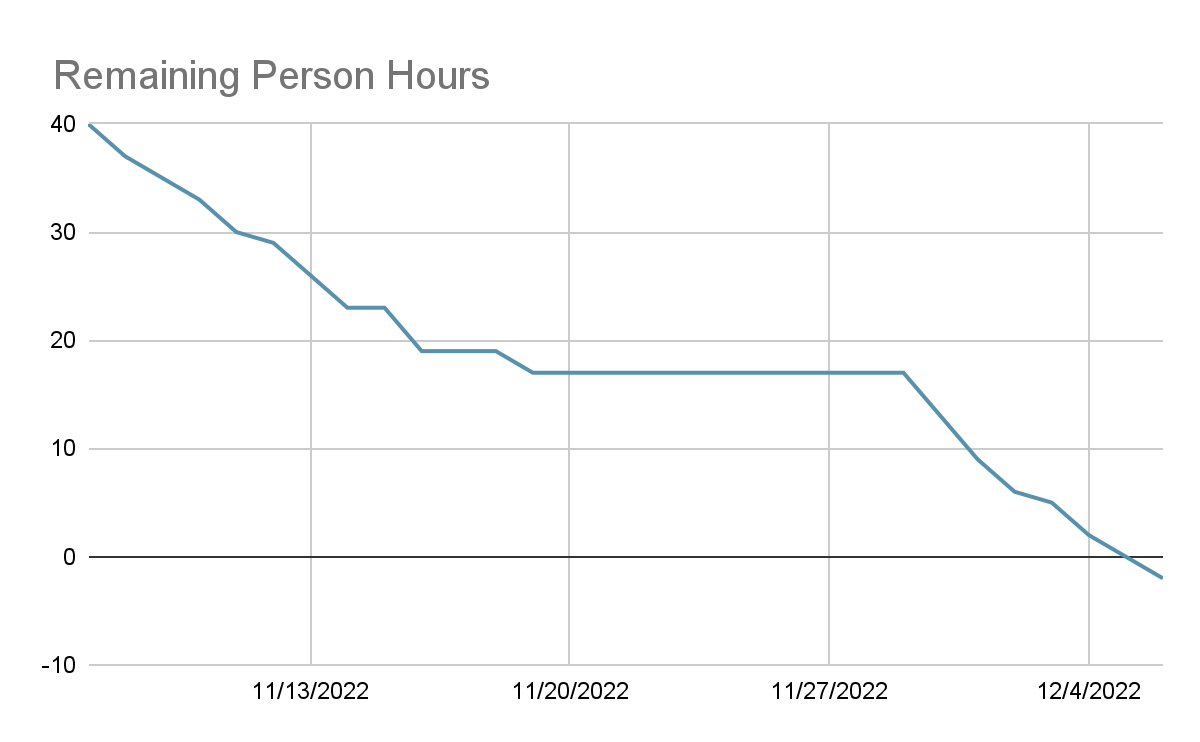
Use Cases Backlog for this sprint

| UC 5 | view calendar |
| --- | --- |
| UC 4 | add and remove personal events |
| UC 7 | Create group |
| UC 1 | add and remove group events |
| UC 3 | add and remove users |
| UC 2 | grant abilities to users |

These Use cases were chosen because they represent the core functions that are needed for any calendar that claims to be able to handle groups. UC6 import 3rd party calendars is nice but, it is the least important of the functions, it also has the greatest chance of going on for a while deepening on how 3rd party calendars structure their data.

## 

## **2.3 SPRINT DESCRIPTION**

This sprint ran pretty smoothly. Sam handled The UI for almost everything, as well as getting the main part of the Calendar working. Luke Handled the front end part of groups and group events. Dylan Handled The BO for groups and events 

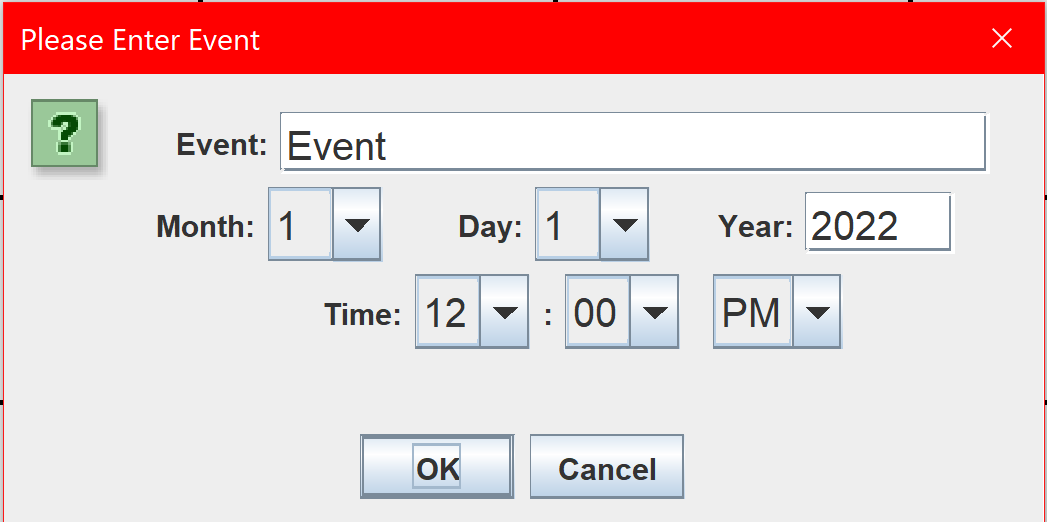
## **2.4 SPRINT RETROSPECTIVE**

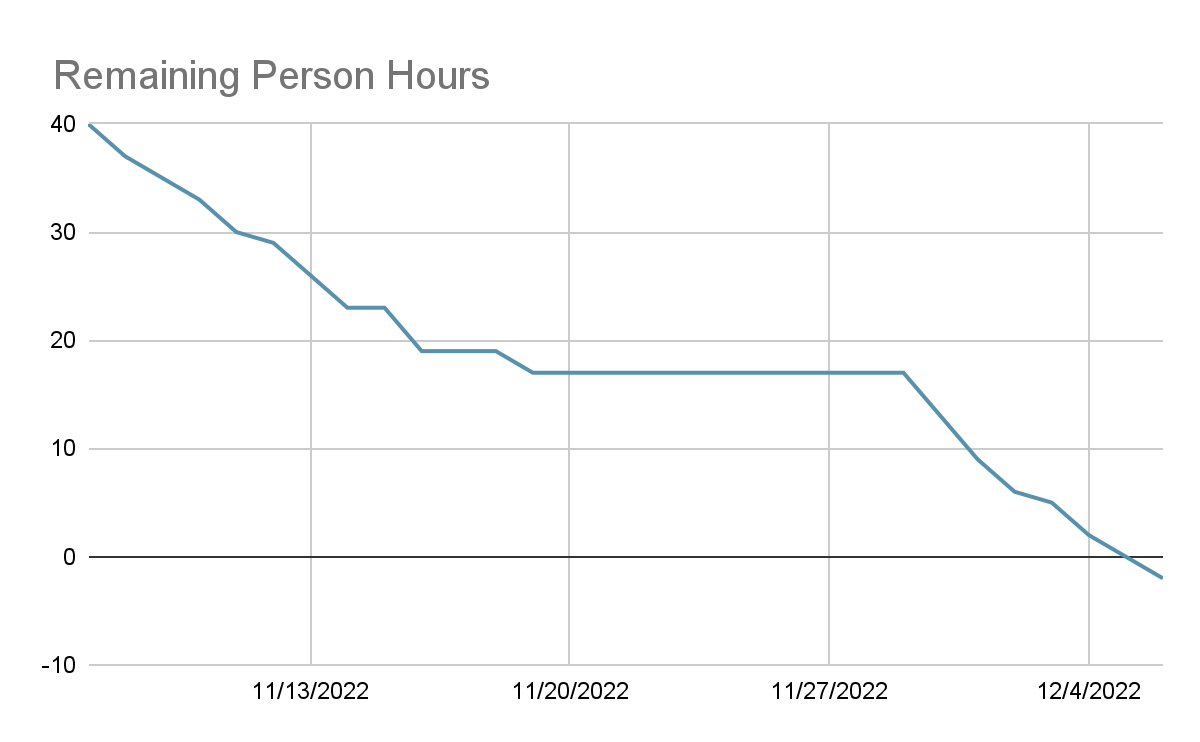
Overall the Sprint went well, with a few problems. The combining and collecting of User Stories went quite well, and if anything our ability to handle each of them was underestimated. However we did not do a good of a job at keeping track of the User Stories. While this didn’t create an actual problem, It was an annoyance as it was hard to tell what other people were doing. The use of GitHub however was a very smart move, allowing us to tackle issues both independently and together when needed. The selection of User Stories might have been too small for the time that we had, and the speed at which we moved, as we got a few User Stories that were originally part of the Sprint Backlog done.

## **2.5 RESULTS**

The entirety of the sprint backlog was implanted, plus some more that was not planned for. Every feature works minus the ability of cross computer communication as time that was bondeon the scope of this sprint.







Scrum meetings happen frequently, about twice a week, but were mostly informal, with little structure as to how they were run.

## **2.6 DISTRIBUTION OF WORK**

## 

| Scrum Master | | Sam |
| --- | --- | --- |
| Task | Person | Success |
| UI | Sam, Luke | Very |
| Events | Sam, Dylan | Very |
| Groups | Luke, Dylan | Very |
| Calendar Import | Sam | Very |

# Source Code

**Original Author of Source Code for CalendarMaker.java and Main.java:**

[**https://gist.github.com/ahmednasserpro/6a0eaadc8e23f5dac0fc94bf29c8d31b**](https://gist.github.com/ahmednasserpro/6a0eaadc8e23f5dac0fc94bf29c8d31b)

**Modified by: Dylan Joos, Samuel Rocco, Luke Duffey**