Calendar App Documentation

Team Something Normal

Abstract:

Calendar is an application designed for students to keep track of their daily events. With so many classes a day, it might be easy to forget a homework or an assignment, so there are built in notification and chat to ask and remind. The requirements were built around the possible use cases the system would have, while the interactive UI was made with java swing. Calendar is an application dedicated to taking events and reminders off the user's mind.

Background story:

With so many things in a day, students and workers sometimes forget daily tasks and suffer the consequences. We wanted an application that would allow these students and workers to set weekly events and reminders for assignments that they receive as well as communicate with others that have the same event to see if they forgot any details. Among many calendar applications out there, like google calendar, we wanted a sophisticated calendar application that just served as an assignment reminder, so we created the calendar app.

Requirements:

1. The system should recognize a host who will have the ability to add people to the calendar through e-mail, or some other form and adjust or restrict the administrative access of other members of the Calendar.
2. The User Host should be able to set and restrict access for different members
a. The System should let the host start with the highest level of access and be able to promote or demote the authority level of other members,
3. They system should recognize new members of the Calendar as having the lowest access (adjustable by the host)
4. The User Members of the lowest priority are only able to view the events on the Calendar and comment on event discussions.
5. The User Member should only be able to view events and allocated resources
6. They System should be able to recognize the resources that are available for each day, and whether those resources are in use
7. The System should support Host facilitated discussion boards per event that host can adjust to be open or closed to responses.
8. The system should support voting, between members of appropriate access level, on which resources should be used for an event given timeslots that can be voted on, and automatically disqualify resources that have passed.
9. The user Host should be able to add events and initiate votes on available time slots.

Analysis:

Used Derby Database

Swing graphical user interface used

The entire project is based in java

Multi-threading and cross internet connection

Automated invitation system that would invite new members and send reminders from company email.

Design:

All our UI uses swing, the login and calendar are intuitive to use and well labeled. We wanted to go for a minimalist design for our UI so the user would be able to navigate quickly through the calendar during a busy day. The chat box and UI was purposefully enlarged for easier clicking and reading. We used a data base to store information easier and it is more secure than an csv file implementation.

Implementation:

We wrote our UI in swing while also building the program functions and the database. Our application heavily focuses on its usage of the multi-threaded Server to allow the opening of multiple clients that can all access the Calendar and its data at the same time.

Evaluation/Testing:

We used Junit to test our program and added, updated and deleted data from the database to test the database. The UI was tested manually, and the code was peer reviewed to catch any bugs that didn’t appear when we ran our application.

Conclusion:

Setting a clear goal is the first thing a group should do, we suffered from an unclear goal which caused multiple setbacks and eventually and fundamentally flawed implementation in our client server design where the properties of the database was not fully understood. Simply starting early and having a lot of meetings does not equate to creating a working program, it is the quality of the project vision and full understanding of the requirements given by the teacher that gets a program working. The project was slow and torture, goals that seemed close only got farther as we went along. So having good googling skills and prior knowledge about the subject is key, because being able to google is very different from understanding what google is telling you. Never less we did enjoy good teamwork moments and a sense of accomplishment up until putting the program together, this taught us a lot about setting goals too high and the importance of setting an clear coal.