CS 130: Software Engineering

Professor: Miryung Kim

TA: Mihir Mathur

UCal

Team Members:
Terry Ye, Alex Xu, Xiwei Ma,
Jingyue Shen, Dinkar Khattar, Jinjing Zhou

October 25, 2019

Project URL: https://github.com/terryyn/CS130-FinalProject

1 Motivation

Even with the rise of information technology, traditional face-to-face meetings are still believed to have more benefits for effective and engaging communication. For college students, offline meetings are especially important for various circumstances, such as working on group projects, studying with classmates, and asking professors questions etc. The problem is that usually a large amount of time and effort are used to decide a meeting time that can fit everybody in. As UCLA students, we actually have lots of common routines. The majority part of our schedule is usually the class planner. Besides, we share the same resources of club activities, campus events, and office hours. Existing tools such as scheduling polls do not take advantage of these UCLA specific information and require lots of manual efforts every time we try to schedule a new meeting.

Our proposed project, UCal, is a web app specifically designed for UCLA students. The aim is to maintain a work/study calendar for each user and help organizing all kinds of internal face-to-face activities. There are multiple simple options provided to create and edit the calendar, such as uploading the .ics files provided by myUCLA, pasting Facebook public event link, or typing customized information manually for any deadlines and activities. Then, our product can use that information in related people's profiles to make organizing meetings and events easily. When a user starts to arrange a meeting, the appropriate time slots can be found by simply aligning selected people's schedules. Moreover, our product would fetch study room availability data around the campus at the selected time. So the location of the meeting can be selected from the suggested study rooms or any custom places. When a meeting is settled, all invitees can receive notifications and their calendars would be updated accordingly. Faculty and TA's can also use this feature to select times for office hours and review sessions, and add events to students' calendars who have the class on their calendar. Thus, our proposed product will let each student browse its filterable calendar at any time and never miss an important activity or deadline, and make it possible to arrange new meetings with other students in a couple of clicks.

2 Feature Description and Requirements

2.1 Populate And Browse Calendar

Upon logging in/signing up, a student user will first be presented with an empty calendar. The user can populate his/her calendar via three methods. The student user can upload his/her calendar data in the format of ics file onto the website. The student user can also add a single event by either manually inputting the title, time, location and category of the event, or providing a link to a Facebook event page and UCal will automatically parse information from the event page and add that to the calendar. In addition, when the student user schedules a meeting (using the feature elaborated in the next section), or gets invited to a meeting created by another user, the meeting will automatically appear on the user's calendar. From the perspective of an instructor user, he/she can add midterm dates or homework/project deadlines for his/her classes, and all students users enrolled in the respective classes will have the events displayed on their calendars.

All events on the calendar will be color coded based on their categories. The user can create different categories for different types of events. When creating an event, the user will be prompted to choose the category of an event, and the user can also change the category of an event later. To display only one or a few categories of events on the calendar, the user can apply filters by selecting the categories that he/she wants to see. The user can also search for a specific event using keywords included in the title of the event.

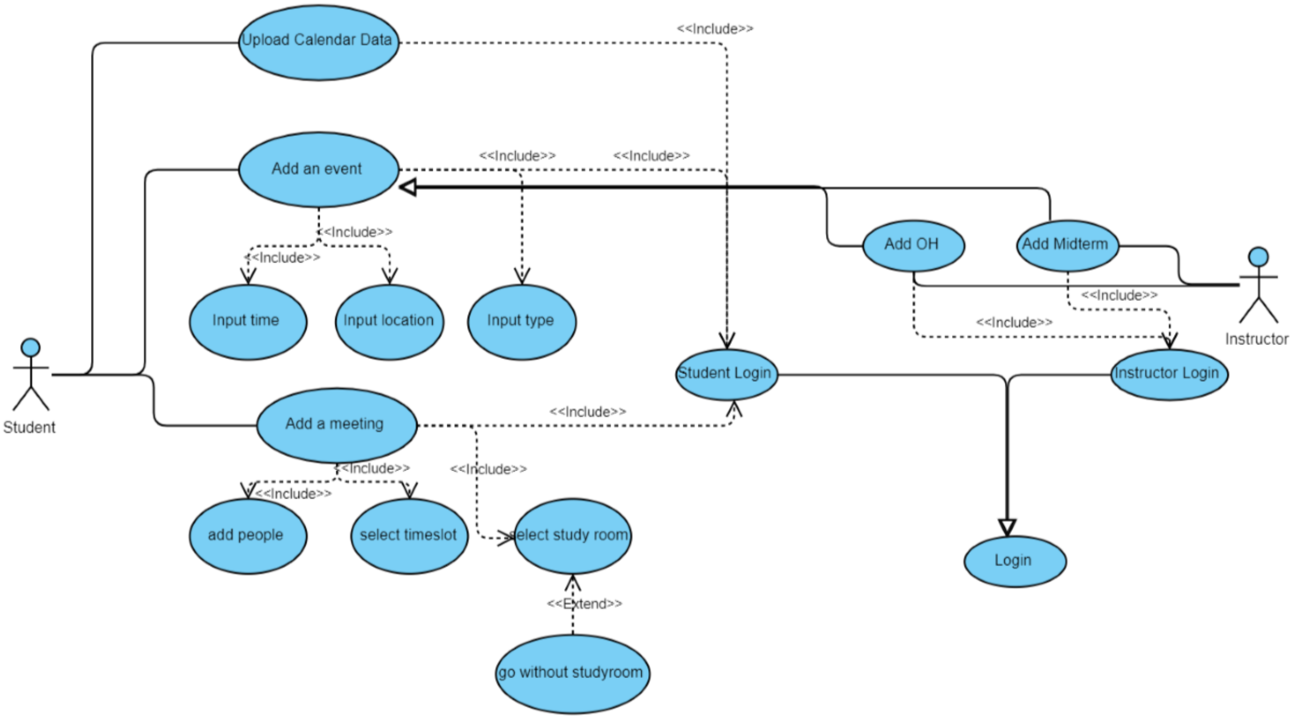
2.2 Schedule Meetings

When a user wants to schedule a meeting, after providing the name of the meeting, up to five likely dates and a list of invitees (identified by email address), UCal will display the timelines of the specified dates. Time blocks during which all invitees are available will be shown in a lighter color, while other time blocks will be shown in a darker color to indicate unavailability. The user will therefore be able to select a time where all invitees will be available to attend the meeting. After the user has selected a time block from the timeline and confirmed the meeting, notifications will be sent out to existing users of UCal and the meeting will automatically be added to their calendars. For invitees who have not signed up on UCal, emails will be sent to the email addresses provided to notify them of the meeting and prompt them to register on UCal.

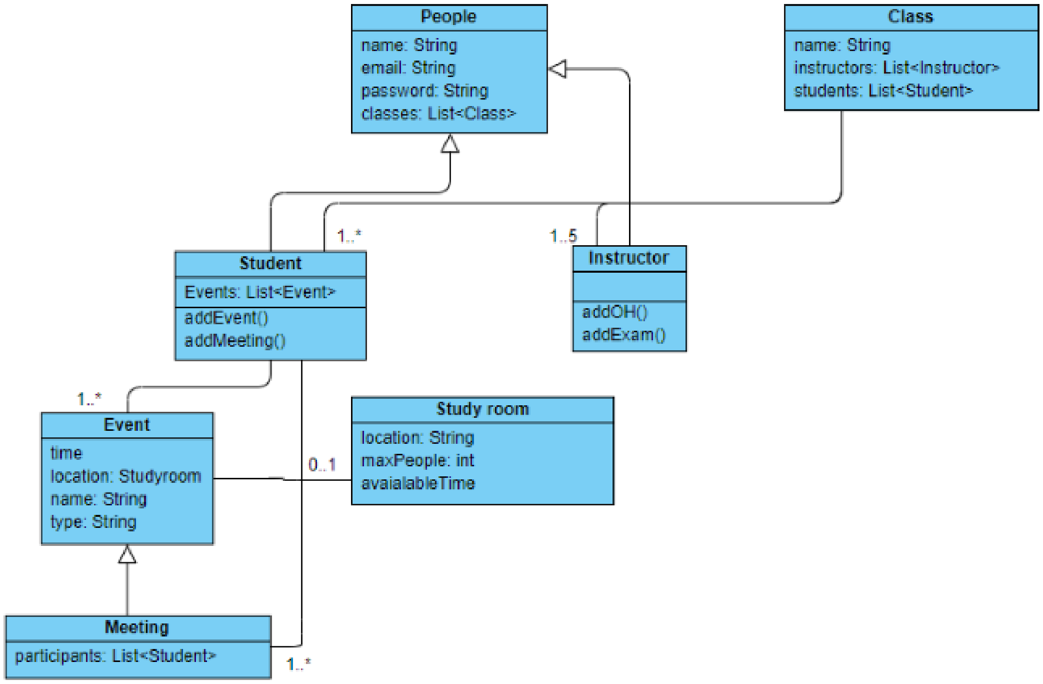
2.3 Find Meeting Venues

When scheduling meetings, the user can either specify a meeting venue him/herself, or try to find an available study room on campus using the web app. After the user has decided on the meeting time, date and number of attendees, the web app will scrape data from all UCLA study room reservation websites and provide the user with a list of available study rooms with the right capacity. In addition, the user can further narrow down the list by applying filters on conditions such as location. The user can hover over each option in the list for more detailed information such as direction, capacity and pictures. Upon deciding on one of the study rooms, the user can click on the link to the reservation website next to the option, and make his/her reservation.

2.4 Diagram

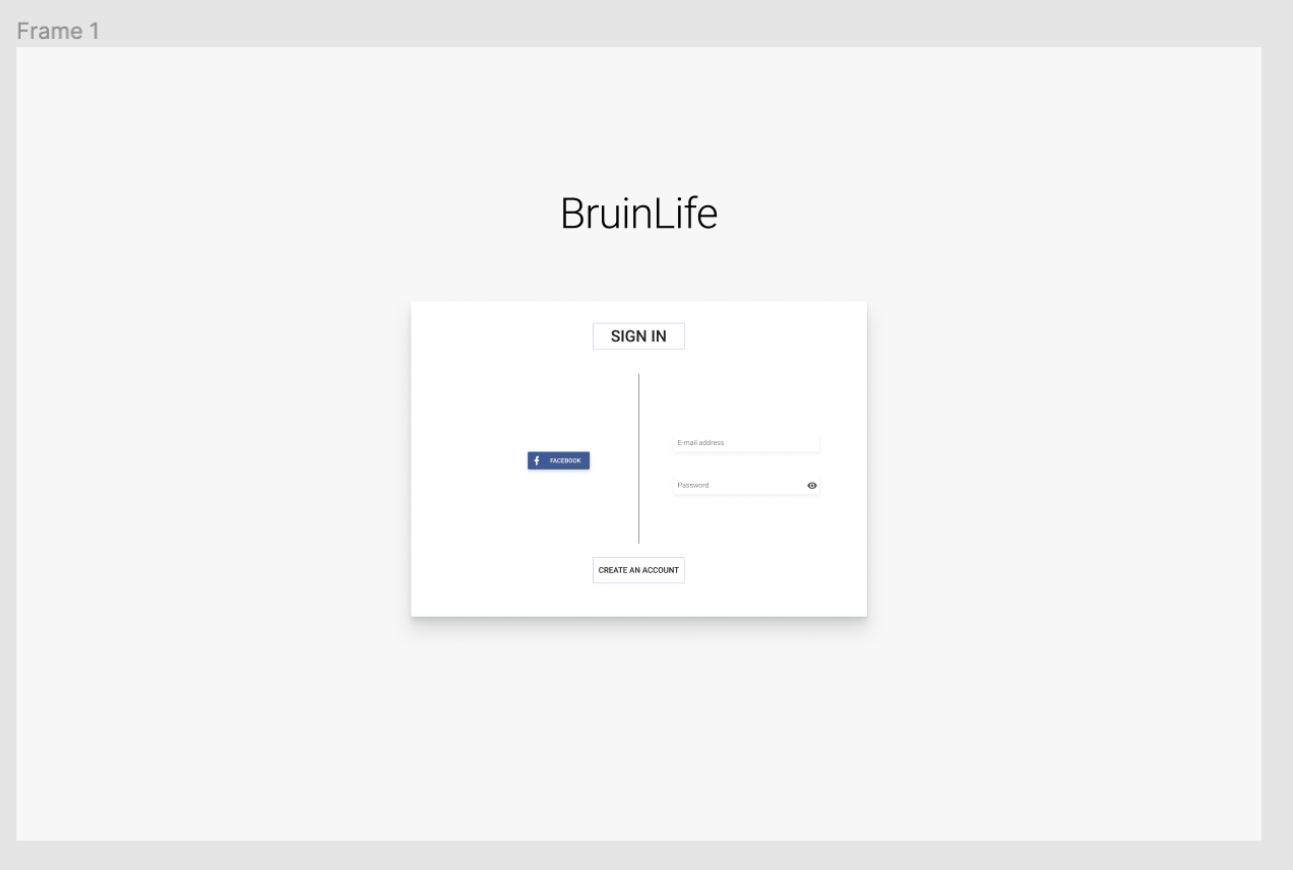


There are two types of users: student and instructor. Student can add event, meeting and upload data while instructor can add midterm and office hour for the class.

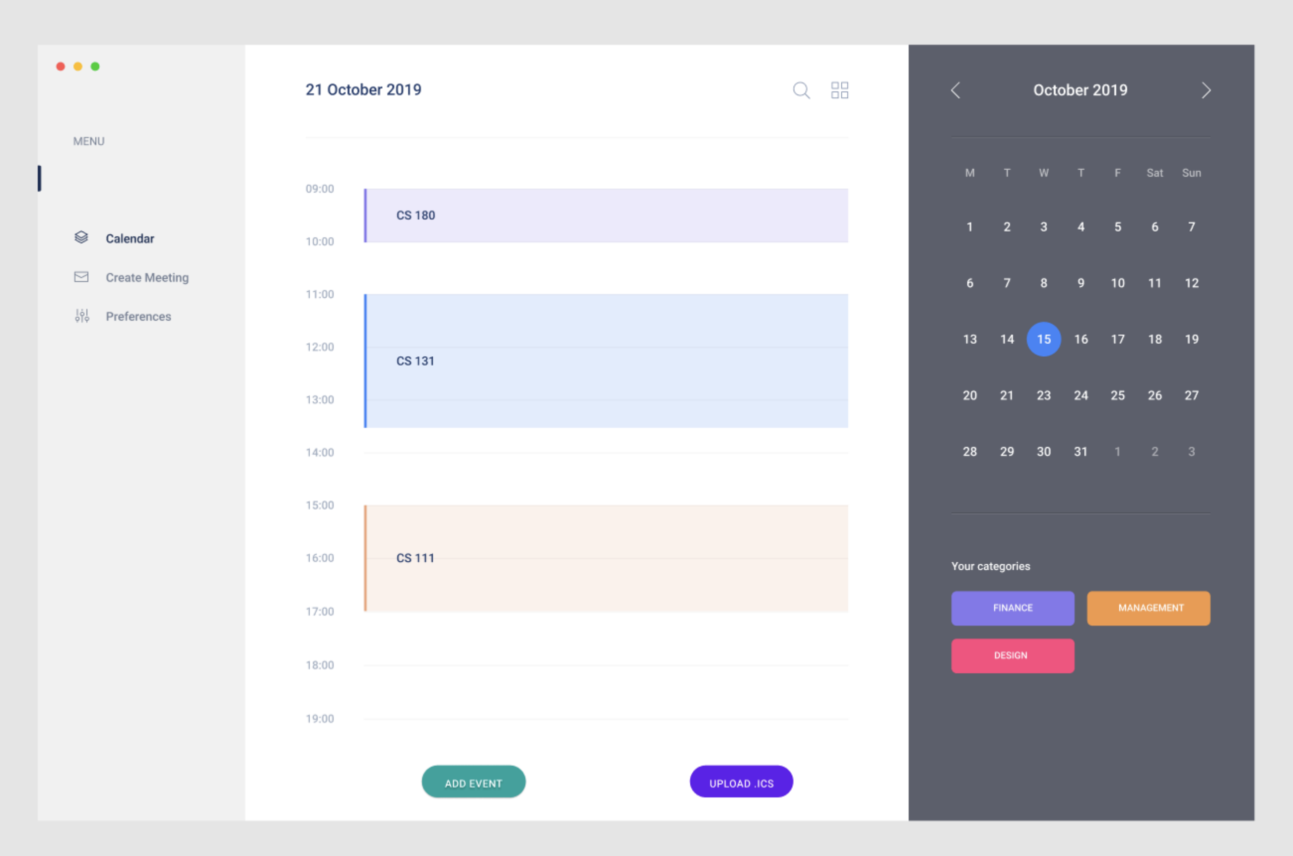


There are two main classes: people and event.

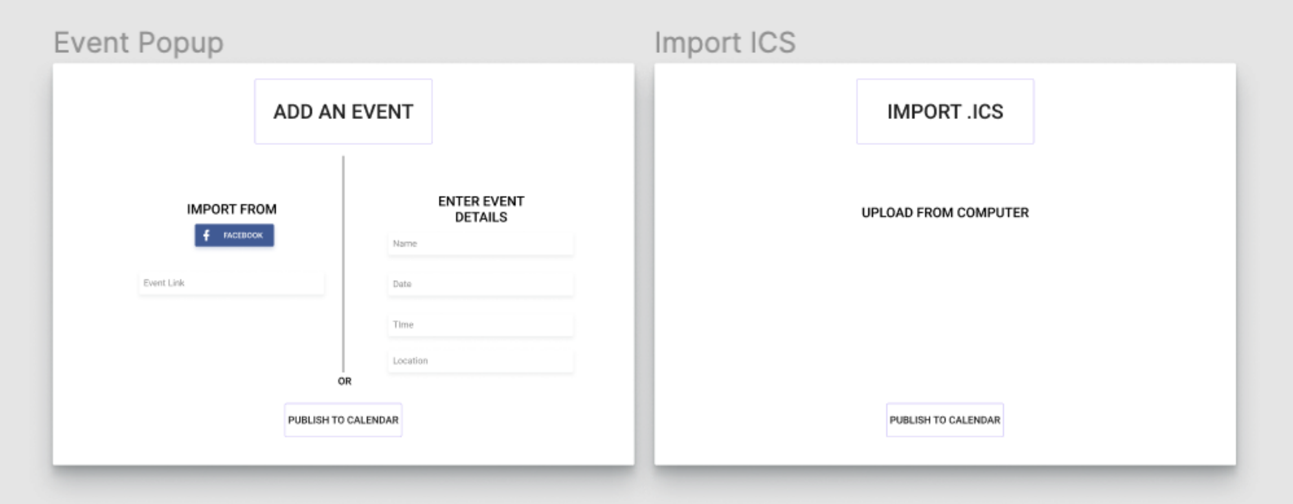
2.5 UI Mockups



Login/sign-up screen. Allows for 3rd-party OAuth.



Main calendar view contains the user's personal schedule. Sidebar allows user to navigate between personal schedule and meeting creation feature. Buttons at bottom of schedule will bring up the modals below.



Modals to add personal events. and import calendar .ics files. Events can be imported using a Facebook link or manually inputted.

3 Feasibility

Technically speaking, this project is feasible to implement. It is a standard web application, with a responsive interactable frontend that is going to be implemented in React and a backend for saving and processing user data. Google Calendar API is going to be used to implement some of the calendar features in our product. We have separated the team into frontend and backend based on the experience of each team member, to make sure the technical skills required are manageable.

From the resource aspect, this project is feasible in that all of the proposed technologies and API are mature and available to use freely. Besides, no other equipment other than laptops is needed. The core features are divided into parts, including operations on individual calendars, arranging meetings/events for different user types, scraping study room availability and notification system. These would be distributed to different team members to ensure the overall efficiency. Considering the time limitation for product development, other features are considered as future extensions. If time permitted, we would implement the proposed future extension, allowing users rate and save study rooms. The information in our database are all voluntarily provided by users and would not be used for any other purposes besides finding time availability. The study rooms data would be scraped from public websites. Thus, all the resources we designed to utilize in our product are feasible.

4 Capability

Our team is capable of the project thanks to the experience at various companies in both front end and back end fields.

Terry learned about backend database when he worked at UCLA to develop course website CCLE. He was also familiar with front-end knowledge like Angular and React while working as an intern at Google. He will help develop the front-end side of the app with JavaScript and React. For Part A he will be making UML diagrams and wrote about features.

Xiwei has worked as a full stack mobile developer while interning with Facebook iOS team this summer. Besides, she has done backend work and testing designs during her internship last year. She also has experience working on open source project and mobile apps for fun. She will mainly contribute to the backend part for this project. For part A, Xiwei wrote the parts about motivation and feasibility in the project's report.

Jingyue has worked as a backend developer during his internship at Google last summer. He also has some frontend knowledge through his course projects and internship. He will mostly work on the backend of this project, but is also willing to learn frontend if needed. For part A, Jingyue wrote about the usage scenario in the report.

Jinjing worked as a backend developer during her internship at Instagram last summer and worked with Django framework and mySQL database. She also has experience working on frontend with React in previous class projects. She will mainly contribute to backend of this project, and will be able to help with frontend if needed. For part A, she wrote about feature descriptions.

Alexander has experience with full-stack development from on-campus involvement in organizations like LA Hacks and Creative Labs, and internships at various startups. He is familiar with a React, Python/JS frameworks, Postgres and some cloud services. He will mostly work on the frontend of this project. For part A, he and Dinkar worked on the designs.

5 Usage Scenario

An example user of our app would be college students who want to discuss some details about a group project in class. His group has many members and he wants to discuss the project in the following 5 days. He is not sure when others are available in the next 5 days and it would be messy to let everyone say their own availability during these days and try to find a proper time. He also tried to use Google Calendar to schedule meetings, but he cannot see some members' calendar since they are not shared publicly. Besides, he also wants to find study rooms available on campus at that time.

Another example user of our app would be a professor. Some students want to schedule a meeting with him/her. But he/she does not have time to answer every student's inquiry about when they are available. Also, he/she wanted to use a calendar to put all his classes and office hours and other possible events together, but he/she does not want everyone sees his personal schedules. So he/she can put all his events in this app, and when students want to schedule a meeting with him/her, students will directly get the possible time without seeing professors real calendar.

Another example user of our app would be a very busy student. He has various types of events. And sometimes he only wants to see one specific type of event on his calendar to get a better understanding of his schedule.

Besides, a student who wants to study alone in a quiet study room can use our apps to find all study rooms available on campus during a specific time period, instead of going to each library's website to find availability of study rooms.