CF/Scheduler

Inspired by Calendly, When2Meet, and the desire to provide free services for the University of the Philippines (UP) community, CScheduler is a free scheduling web-application serving an accessible, user-friendly, and convenient process to schedule meetings, interviews, and appointments for both personal and professional use for:

UP Career Assistance Program for Engineering Students (UP CAPES)
Jasmine Belle Ang (Information Technology Vice President)

DEVELOPED BY:
Serge Alec Rivera
Felix IV Bueno
Philip Angelo Dulva
James Adrian Perez

PROGRESS IN:
CScheduler-WebProfile

IN PARTIAL FULFILLMENT OF:
CS 191 - Software Engineering I
Ligaya Leah Figueroa

Customer Statement of Requirements

The main motivation for this project is the prevalence of online meetings and consultations brought about by the pandemic. While there are already existing software for this demand, most of these applications are catered towards businesses and thus have paid packages which are not ideal or practical for members of the academe to avail of. The project team is aiming to develop a free web application accessed by different bodies of the UP community ranging from student organizations to faculty members. However, as the main client behind the scheduling service, the development team will first satisfy the requirements of UP CAPES.

UP CAPES is a student organization based in UP Diliman (UPD) which aims to assist the UP engineering student body with their career goals through organized events, information dissemination initiatives and financial assistance programs. To tailor these activities for the engineering students, the organization conducts a market research that pinpoints the career needs of the engineering students. One of the phases of said market research is qualitative research. The qualitative research is done through an interview with selected students across all engineering courses in UPD and has the objective of gathering insights to be used for the quantitative research.

One of the goals of this project is to streamline the interview process implemented in the qualitative phase of UP CAPES's market research. Previously, UP CAPES had utilized Calendly, a free online appointment scheduling software which has a user-friendly interface and simple process to schedule one-on-one meetings. The key feature, which made Calendly the organization's option for a scheduling software, is the feature to manage a team of interviewers with a round-robin selection of interviewers upon an interviewee booking. After checking out the different plans of scheduling web applications such as Zoho, GReminders, Doodle and Calendly, it seems that the sought mentioned paid features for free while maintaining the convenient and simple processes in scheduling a meeting. These paid features were tested by the developers of this project by availing of Calendly's free trial of the paid package.

Figure 1. *Booking a slot through Calendly*

Figure 2. *Confirming a slot through Calendly*

Figure 3. *Automated Google Calendar invite sent by Calendly*

The client shared that UP CAPES, with a free trial of Calendly, made use of Calendly in order to have a single person or the scheduling manager collate and set interview schedules allowing a team of interviewers to conveniently manage their common schedules under one calendar. This calendar will select among the team of interviewers through a round-robin fashion whenever an interviewee books an appointment. Figure 1 above shows how interviewees would book their appointments with an interviewer. Note that the calendar shown is already the single calendar generated from the availability of multiple interviewers and the calendar is made available through a link that potential interviewees can click. Figure 2 presents the details requested from the interviewee to complete the booking of appointment. The email address provided in Figure 2 is used to send calendar invites. Figure 3 illustrates the automated Google Calendar invite sent to an interviewee's email address. Observe that the meeting details, Zoom meeting in this case, is attached to the Google Calendar invite. Moreover, cancellation and rescheduling links are also part of the Google Calendar invite.
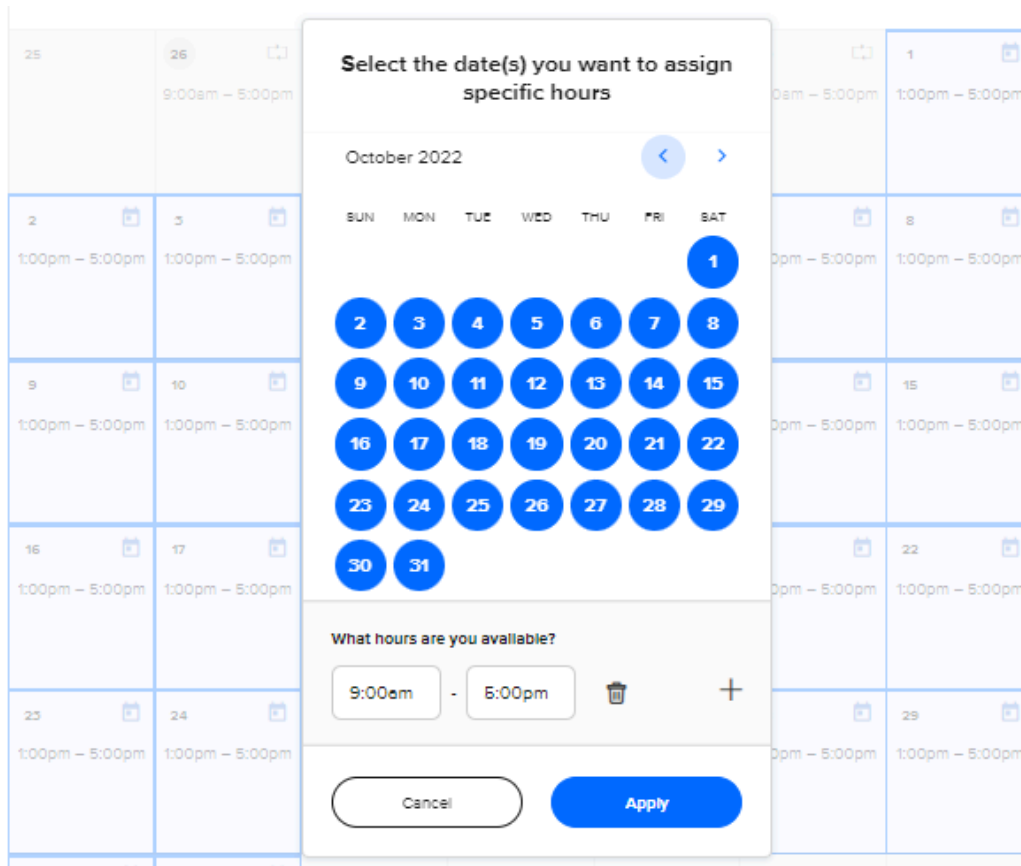
Figure 4. *Calendly configuring interviewer availability*

It must be taken into consideration that the assigned scheduling manager admitted that they used another web application called When2meet to collect the available times of the interviewers in order to have the scheduling manager configure the round-robin interviews for the interviewers in Calendly. The scheduling manager pointed out that Calendly has a slightly poor user experience when it comes to setting the availability of the user. Figure 4 above shows how one can set availability time ranges for a range of days. If an interviewer has different availability time ranges spanning multiple days, they will have to manually set each individual time range for each day.

Figure 5. *Configuring date ranges in When2meet*

Figure 6. *Signing in to When2meet as a unique user*

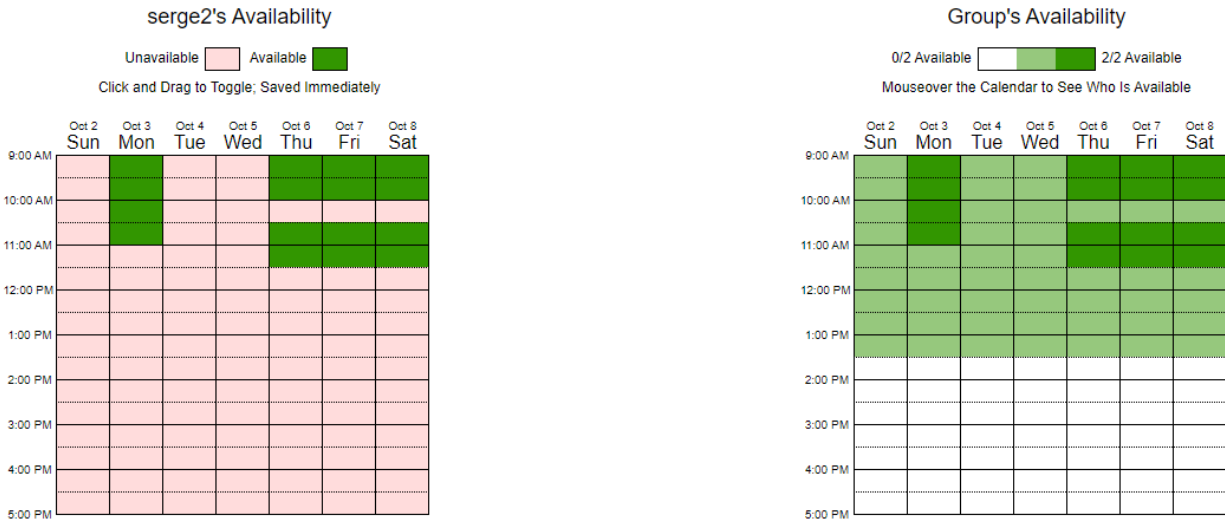**Figure 7.** *When2meet setting availability time ranges for first user*

Figure 8. *When2meet setting availability time ranges for second user*

In contrast to Calendly's setting of availability time ranges, a When2meet user will simply have to drag their cursor on the left calendar in order to configure their availability time ranges across a range of days as seen in Figures 7 and 8. The initialization of a calendar and user are seen in Figures 5 and 6, respectively. The available days in the calendar are also set by dragging the cursor across the desired dates. Note that a When2meet user is simply an alias to an anonymous user so that said user can edit their schedules should the need arise. Registration for an account with an email address in When2meet is not necessary but the name field is a required input field before a user can set their availability.

Aside from Calendly's user experience, according to the client, the round-robin feature is not yet optimized or might not have worked as intended provided that there were complaints regarding some interviewers having significantly more interviewees than other interviewers. As a consequence, interviewers had to communicate among themselves and directly with their respective interviewees in order to manually reallocate and reschedule interviews. There was also a concern that having to reschedule or cancel an interview will automatically send emails to the people involved. This behavior is not ideal if multiple reschedules or cancels will be performed by the interviewers or the interviewees because as a result these automated emails will spam the inboxes of both the interviewers and interviewees.

Having enumerated all the major concerns of the client, the project's requirements is drafted as follows:
   1. Collate the email addresses of the interviewers as a required input identical to Figure 6

2. Collect the availability date and time ranges of interviewers with a user interface and experience as simple and convenient as When2meet (drag cursor around) as illustrated in Figures 7 and 8
3. Generate booking links where interviewees can book appointments and see the availability of the interviewers associated with the generated link in a calendar view similar to Figure 1
4. Collect the interviewee's email address as the last step in booking a time slot for sending of calendar invites
5. Create Google Calendar invites for the interviewer and interviewee which could be sent out by either the web application's account, similar to how Calendly has its own sender, or the interviewer's Google account after an interviewee has booked
   a. If the user opts to have the web application's account generate and send out the invites, no additional step is necessary. The web application's account will send out calendar invites to the email of the interviewer and their respective interviewee.
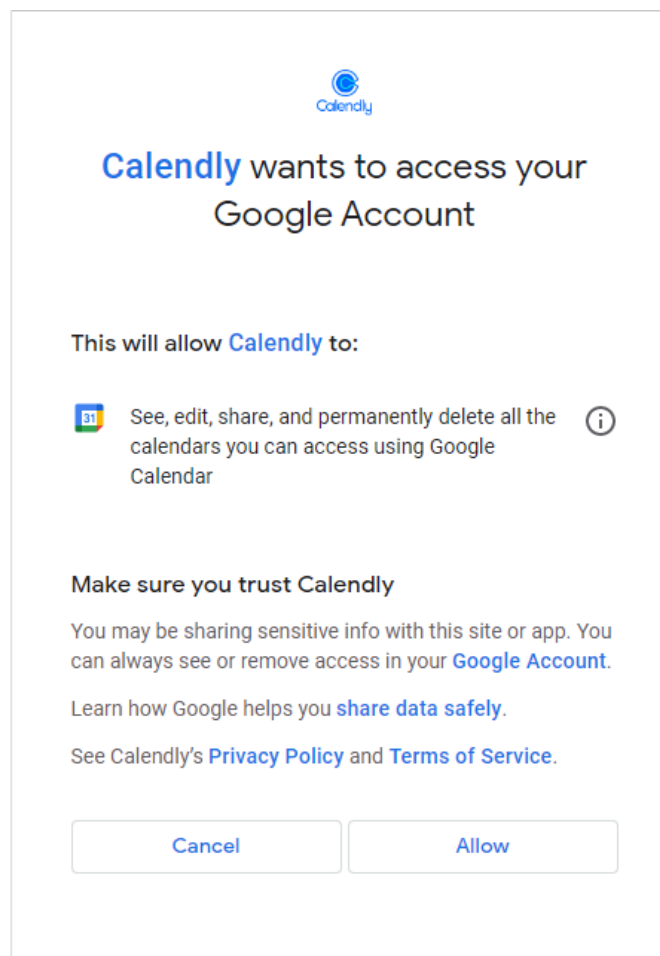

Figure 9. *When2meet setting availability time ranges for second user*

b. If the interviewer wants to send invites through their Google accounts, the interviewer must provide Google authentication in order to access the interviewer's Google Calendar as seen in Figure 9 above. This implies that an option to sign in with their Google accounts should be implemented in the web application. After providing authentication, the Google Calendar invite would originate from the interviewer's account.

6. Before the booking link is generated, allow interviewers to attach messages with the calendar invites for any meeting details

7. Cancel and reschedule meetings through links available in the Google Calendar invite
   a. The rescheduling process will redisplay the calendar view from the associated interviewer's calendar. This is based on Calendly's rescheduling process wherein the calendar view is reused to show the available time slots of the interviewer. The scheduled Google Calendar event generated by the web application or interviewer's account should be deleted and emails should be sent upon a successful rescheduling.

Figure 10. *When2meet setting availability time ranges for second user*

   b. The cancellation process will involve providing a reason and pressing a button similar to Figure 10. A successful cancellation should send out emails containing the reason for cancellation and the scheduled Google Calendar event should be deleted.

8. Form interviewer groups upon a press of a button and allow interviewers to assign themselves or other interviewers to groups

9. Generate a booking link for an interviewer group which has the same features as Requirements 1 to 7 but selection of calendar view to display will depend on the selection algorithm
    a. A round-robin algorithm can be used to queue which calendar view will be accessed by the interviewee clicking on the link.
    b. Alternatively, maximum number of slots can be assigned for each interviewer. Once an interviewer has reached their maximum slots, their calendar views will be deactivated unless an interviewee would cancel their appointments.
10. Concurrent processes (booking, calendar viewing and cancellation) should be taken into consideration and not affect the algorithms present in the web application

Preliminary UI Design

There are several points of access for the CFSchedule web application mainly: default, event link, cancellation link, and rescheduling link. The navigational paths for each point of access is shown in diagrams 1 to 5.
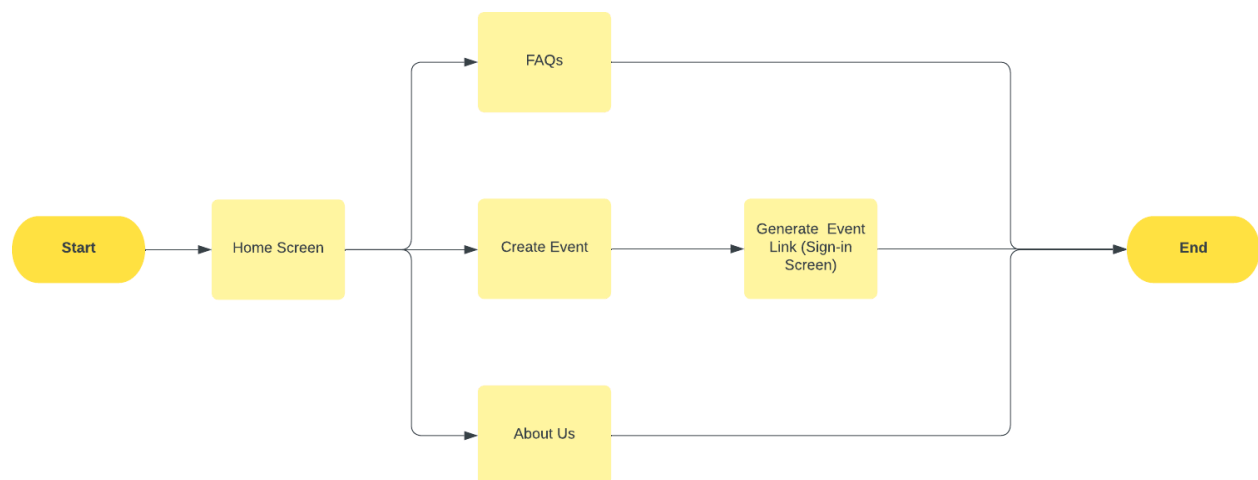
Figure 11. *Website Point of Access: Default*
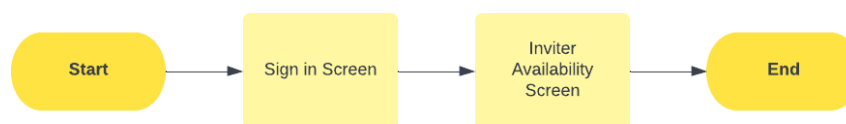
Figure 12. *Website Point of Access: Event Link (Inviter)*
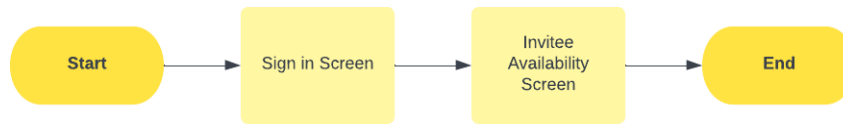
Figure 13. *Website Point of Access: Event Link (Invitee)*
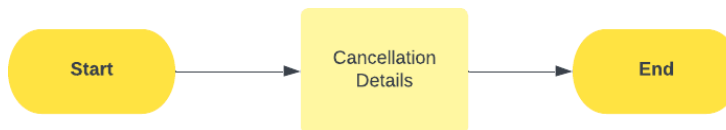
Figure 14. *Website Point of Access: Cancellation Link*

Figure 15. *Website Point of Access: Rescheduling Link*

Screen mockups as well as the prototype can be accessed through the following link: CF/Scheduler_MockUp. Below are the images of the prototypes.

Figure 16. *Home page*

Figure 16. *Create event page*

Figure 17. *Sign in page*
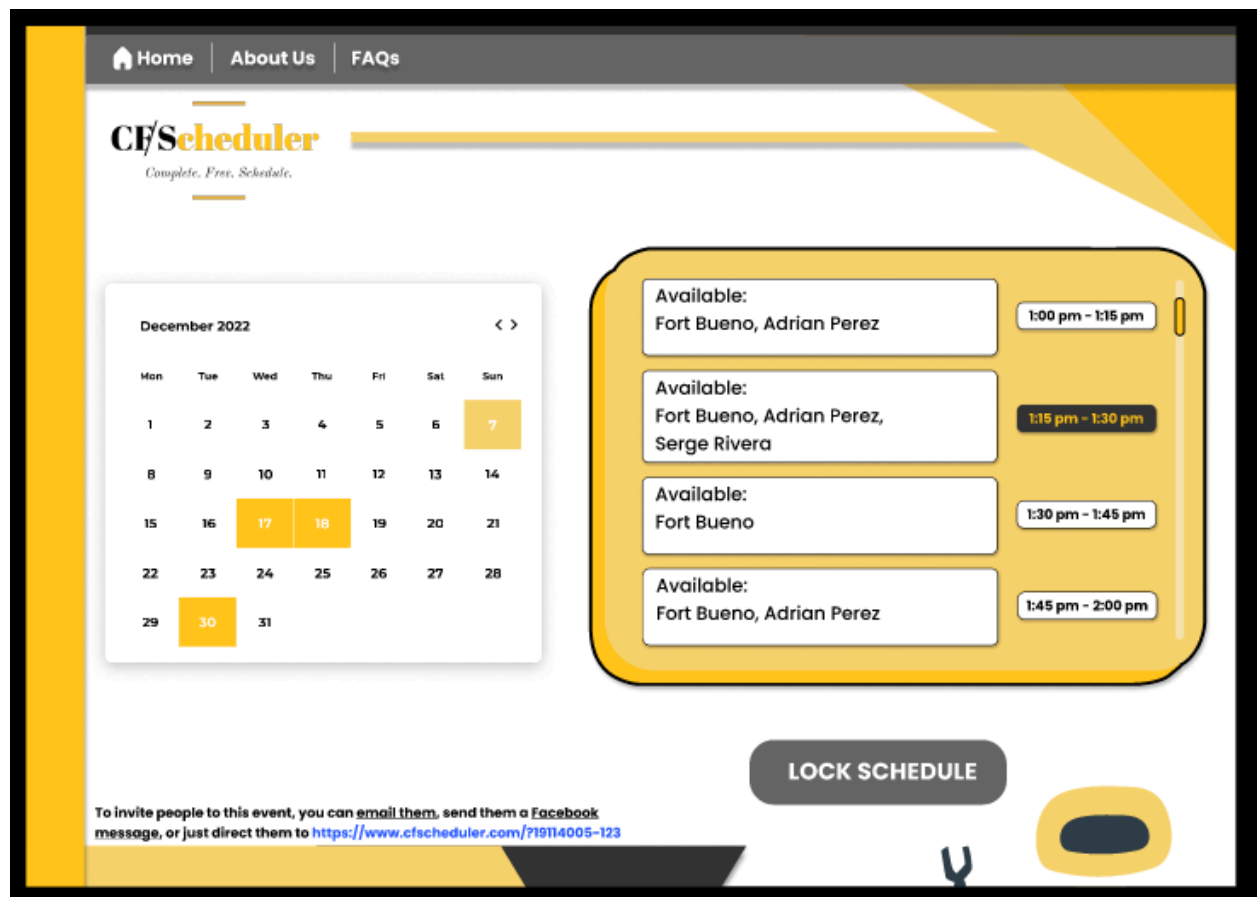
Figure 18. *Availability inviter page*

Figure 19. *Availability invitee page*

Figure 20. *Invitee cancellation page*

Plan of Work

Estimate Costs and Resources

People, time, etc. Use at least 2 estimation techniques (e.g., process vs. problem-based estimation - see chapter 26). What system resources does your system depend upon? Examples are, screen display, disk storage, communication network, or you may be accessing some special sensor/instrument. Describe exact requirements for these resources in order for your system to run. For example, you need color display, with minimum resolution of 640 × 480 pixels; minimum of 2 Gb hard disk space; minimum network bandwidth 56 Kbp
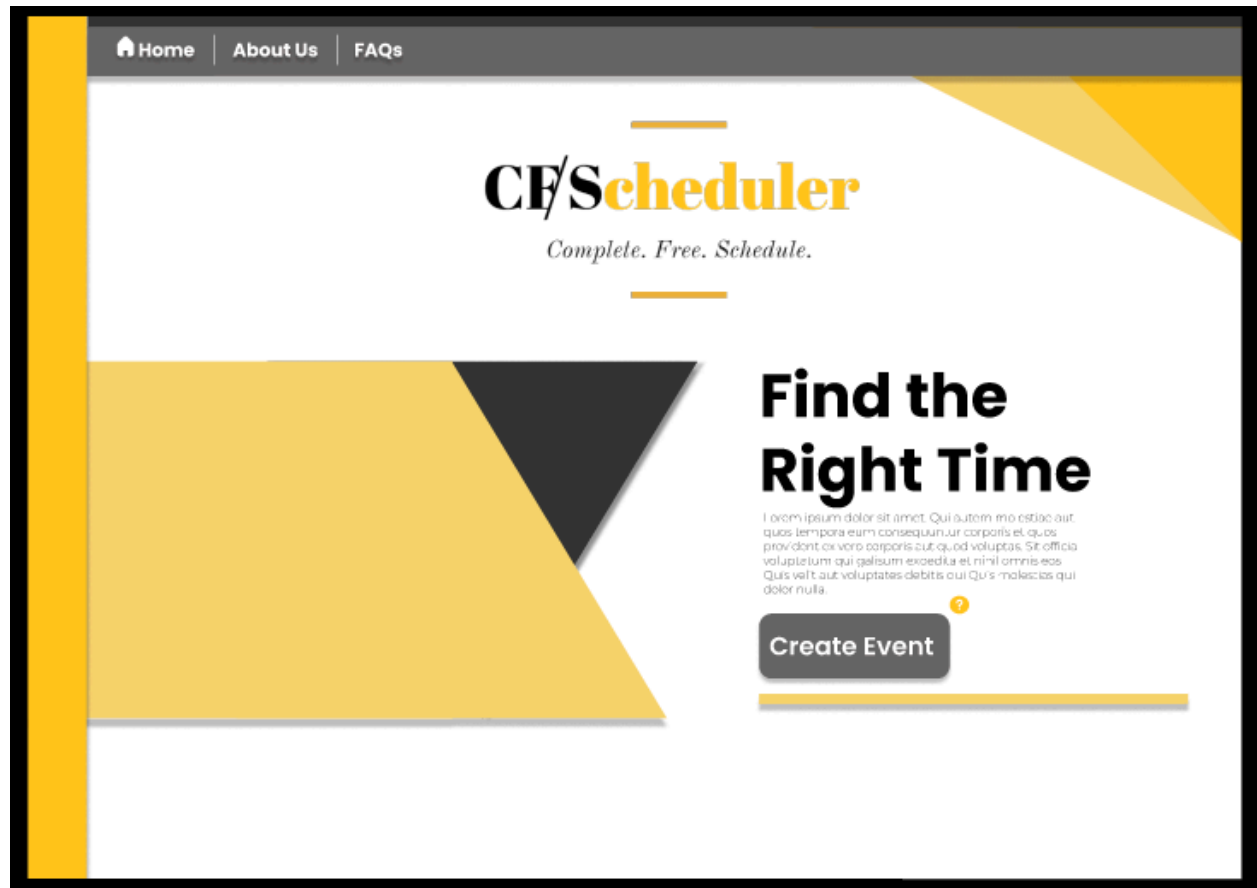
Identify Schedules and milestones

Listed below are the tasks and targeted milestones for this project:
- Search for Clients
- Negotiate Project Requirements
    - Objectives of proposed software
    - Features of proposed software
- Researching and Deciding the Technology to be used in Developing Proposed Software
    - Front-End
    - Back-End
    - Database
- Preliminary UI Design
- Web App Programming (Front-End)
- Web App Programming (Back-End)
- Database and Server Setup
- Software Testing

Figure X. *Process Flow*

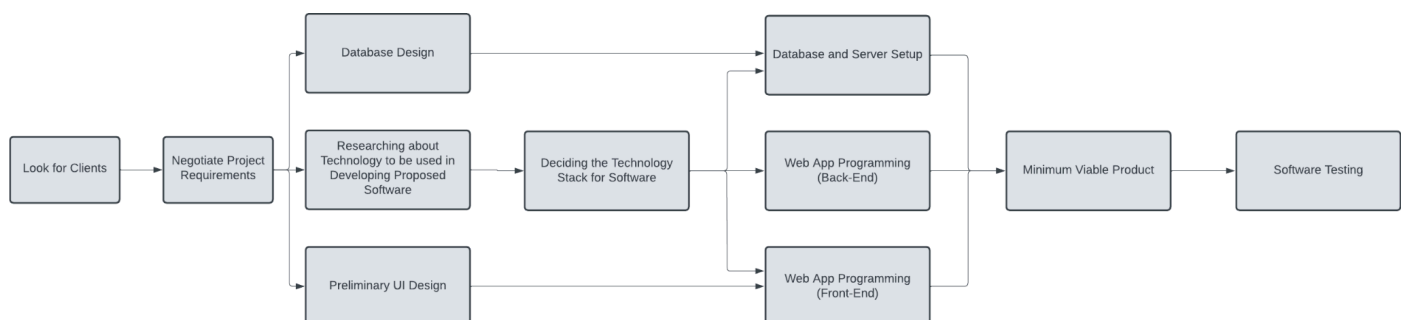| Task	September				October				November				December				January		
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3
Search for Clients	X	X																	
Negotiate Project Requirements		X	X																
Researching about the Technology to be used in Developing the Software				X	X	X													
Deciding the Technology Stack for Software						X	X												
Database Design							X	X	X	X	X	X							
Preliminary UI Design							X	X	X	X	X	X							
Preliminary UI Design: Home Page, About Us, FAQs							X	X	X										
Preliminary UI Design: Create Event, Sign in, Cancellation							X			X	X		X						
Preliminary UI Design: Availability Screen - Inviter, Availability Screen - Invitee											X								
Web App Programming (Front-End)													X	X	X	X	X		
Web App Programming (Back-End)													X	X	X	X	X		
Database and Server Setup													X	X	X	X			
Software Testing																	X	X	X

Figure X+1. *GANTT Chart for CScheduler*

Project metrics

To keep track of the project's progress and its assurance on quality, these metrics will be used:

- Distribution on the effort of finishing software engineering tasks
- Errors uncovered during each and every review period
- Comparison between the time allotted vs actual time finished for each task
- Bugs seen after the development of the product
- Number of scheduled meetings taken

Meetings will also be held twice a month for adjustments of tasks given as well as check-ups on the assigned software engineering tasks to each other. This will allow the team to reduce unnecessary delays as well as save time and spare effort for each member.

WebApp Project Metrics will also be used for a sound assessment of the product for its implementation stage. These metrics are as follow:

- Number of static Web pages (the end-user has no control over the content displayed on the page)
- Number of dynamic Web pages (end-user actions result in customized content displayed on the page)
- Number of internal page links (internal page links are pointers that provide a hyperlink to some other Web page within the WebApp)
- Number of persistent data objects
- Number of external systems interfaced
- Number of static content objects
- Number of dynamic content objects
- Number of executable functions
- Number of scheduled meetings finished
- Number of canceled meetings

To compute for the efficiency in defect removal, we use the following equation:

$$\textbf{DRE = E/(E + D)}$$

Where,
DRE = Defect Removal Efficiency
E = Number of errors found before delivery of the software to the end-user
D = Number of defects found after the delivery of the software to the end-user

To maximize DRE, we need to effectively and efficiently deal with the errors before its implementation stage. To do this, we plan to check for errors and deal with them every meeting.

Effort breakdown

Task	Handling Member
Coordination with Client	Serge, Adrian, Fort, Philip
Web App User Interface Design	Adrian, Fort
Web App Programming (Front End)	Serge, Adrian, Fort
Web App Programming (Back End)	Serge, Philip
Setting Up of Database and Server	Serge, Philip
User Testing	Serge, Adrian, Fort, Philip
Deployment of Web App	Serge, Adrian, Fort, Philip
Software Quality Assurance	Serge, Adrian, Fort, Philip
Operations and Maintenance	Serge, Adrian, Fort, Philip

Project Risks

Risk	Probability	Impact	RMMM
Development Environment - Since the developers are students with no particular experience, quality tools may be hard to find.	60%	2	Codes from similar apps may be reused. Keep researching for tools that may be able to help with developing the software.
Delivery Schedule - implementation of the product may be delayed due to bugs and defects.	50%	2	Constant meetings can reduce delivery delays. Efficiently removing defects as soon as they get found and relentless testing can help mitigate this.
Security Risk - Security may be breached and meeting schedules as well as links may be exposed to the public.	30%	2	Use open-source codes so that they may be tested by other users. Also recycle codes from similar websites.
Quality Assurance - The developers are mainly students who have no prior experience with developing softwares so the quality may be compromised.	30%	4	Referring to already existing similar softwares to ensure that the quality is improved. Constant research on how to improve the software.
End-Product Maintenance - Codes might not be easily maintainable or sustainable which makes it hard to keep up with the latest specs and software.	20%	3	Using sustainable codes during the development process. Ensure readability and comments to make sure that defects and bug fixes can be done easier.