Social Calendar Final Report

Wilson Song[*]
wsong000@citymail.cuny.edu
CUNY City College of New York
Brooklyn, New York

ABSTRACT

Life now can be very busy and it can be very tough to balance work and other responsibilities with social activities. Especially in a large urban city like New York everything seems to move at a faster pace, so it can be tough to manage your days while keeping a healthy social life. So I am proposing a web application called Social Calendar that will allow users' friends and family to share an interactive calendar so that everyone can see each other's daily schedules This web application will be built similar to the Google Calendar App while adding many new features that will make this calendar application unique. This application will appeal to users of all ages from friends that want to plan social gatherings to families and parents who want to keep track of their kids schedules and activities to plan appointments or events.

ACM Reference Format:
Wilson Song. 2018. Social Calendar Final Report. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Social Calendar is a web application that allows users to collaborate on a single calendar. The main goal of this project is to allow groups of friends or families to be in sync and to be able to know where everyone is or is doing at any specific time. This problem arises when my mom would ask my about my school schedule everyday because she would need to know when to cook lunch or dinner and when I have free time to help her so I proposed this project to help families with similar problems. Having a shared calendar allows everyone to know each other's schedules to be able to plan events ahead of time. Using such a calendar app can be very complicated for some parents so one of the objectives of this project is to make it as user-friendly as possible and one of the ways of doing that is to allow a lot of customization to that the user can choose how they want to use the app and what they want to use it for. There are plenty of calendar applications on the app store but they do not have the specific features that I am adding which I believe will add a lot of utility for everyone.

So to build this application I used Django as the backend. Django is a strong framework that will give us all the tools necessary to create the web application. The backend is mostly taken care of using Django so the frontend is where most of my work will be.

2 BACKGROUND AND RELATION TO PRIOR WORK

I am an undergraduate majoring in computer science. I've completed countless projects but this is the biggest project I will have attempted. Previously I worked in a team and designed a food delivery application using python with QT as the user interface design. At first I wanted to build this project using React Native because of its flexibility and the ability to work as a IOS/Android/Web application which I thought was crucial for this project. But the problem was that I had little experience using Javascript which is required for React. After seeing other project proposals and realizing that a lot of people are deciding on using Django, I was inspired and decided to go that route too because my strongest and most comfortable language to use. So instead of starting and learning from scratch I already had a strong basis. I did a lot of research on how to use Django and I learned that I still must use Javascript and CSS to build the web pages so I learn those languages as well. Overall, I've done a lot of work to learn how to use Django and learn Javascript from tutorials all over the internet. So coming into this project I was at first overwhelmed and scared that this project would be too hard.

I've done a lot of research and practice with python Programming over the summer during my internship to prepare myself to tackle this project. I was still relatively inexperienced when it came to developing front-end but during my internship I was exposed to so much new technology and I was forced to learn things I never used before in order to complete my tasks which made me confident that I'll be able to learn JavaScript, HTML, and CSS. Like I said earlier, I was using a lot of python in my work so I felt very confident in the back end development since Django uses Python.

My objective coming into this project is to creating something functional and to add as much of the proposed features as I can. This is one of the first times that I am working on such a big project by myself so I can only rely on myself to do the research which could be a challenge but also a benefit in that I'll become a lot more self-sufficient. I had big hopes of adding a lot of features and creating a beautiful simple application.

3 DESCRIPTION

This project will be created with a focus on the front-end while incorporating a small portion of back-end to store the data. The following will show all the implemented features for the application as well as the standard of success and schedule used for each feature.

[*] Both authors contributed equally to this research.

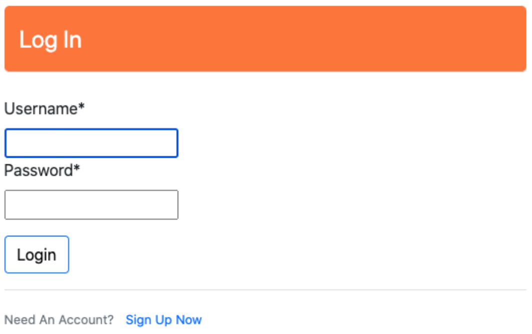
Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

3.1 Login/Sign-Up

The first and most important feature is the login/sign-up and authentication. This allows users to login to their own accounts where they can access their own calendars and groups while adding their friends as well. Django has a default authentication system already and it was even easier to implement considering I was using their default database SQLite. So the most important task here for me was to style the login and sign-up page. This was my first attempt at using JavaScript, HTML, and CSS. I found that there are lots of great styling tools out there to make things much easier such as Bootstrap which I used to style the sign up page. The sign up page has verification to check if the user already exists in the database and also checks for password strength. The password will not be allowed if it contains the user's username or name in it. Overall, I was very happy with the login and sign-up page as it looks very simple and functions perfectly.

I worked on this feature during Week 1: August 26 - September 5 and Week 2: September 6 - September 12. The implementation itself was relatively simple but most of the time was spent learning and reading documentation on JavaScript, HTML, and CSS. These two weeks were very important because it created a strong foundation for my journey to learning front end development.



Log In

Username*

Password*

Login

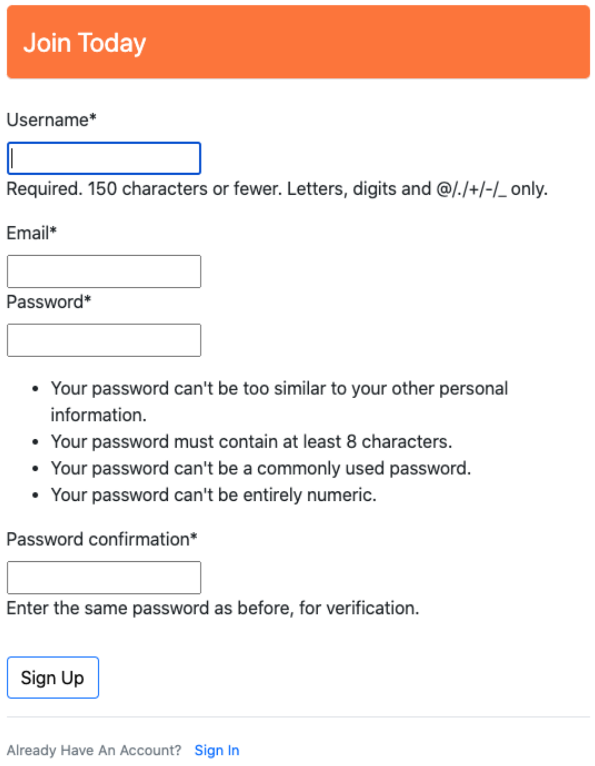
Need An Account? Sign Up Now

Figure 1: The first page the users will see when accessing this application is this login page.

3.2 Calendar UI

Coming into developing this Calendar UI, I knew I wanted something that was simple and easy to understand. I feel that a lot of applications nowadays designed to look clean and simple for the user. In the beginning I thought I needed to creater the UI with the help of the Google Calendar API but it turns out that I can create the Calendar UI purely with JavaScript, HTML, and CSS and then connect it to the back end. I found this way to be much simpler since there was already a lot of HTML Calendar Templates online that I can take a inspiration from. This also means that I don't have to deal with the complexity of using an API just yet as I wanted to move through this project as fast as possible so I can have ample time in the end do refine anything I need to.

On Figure 3. we can see the fully implemented Calendar UI. This is the page that the user will first see when they log in. On the top left, it shows the user's name with a drop-down menu, a profile picture with a drop-down menu, and a button labeled "Personal Calendar". The drop-down menu with the user's name allows us to access the dashboard which we will talk about later on. The drop-down menu with the profile picture will allow the user to change their profile picture. The "Personal Calendar" button will allow the user to access their own personal that looks just like the one in the Figure. There are also three buttons near the top that are used for the groups feature that we will talk about later on as well. So the Calendar UI itself was made to look as simple as possible with no extra distractions. As shown in Figure 4. the user can interact with the calendar by clicking on each day to see the events on the Calendar; clicking on a day will bring the user to another page showing the events and furthermore the specific event details including descriptions, start and end time, and priority. The user can also switch the months by clicking the two arrow buttons at the top of the calendar.



Join Today

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

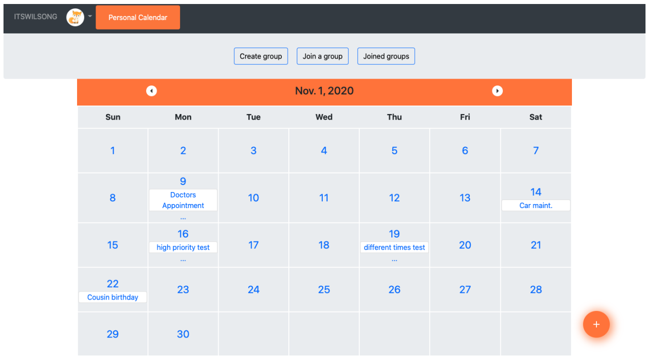
Sign Up

Already Have An Account? Sign In

Figure 2: This is the sign up page styled using Bootstrap

Overall, I'm quite satisfied with the look of the actual calendar but I was not sure about how the events look on the calendar. Though this was as much as I can do due to my lack of experience on front end development. But it still made me think of ideas to make this cleaner which was to just have a circle dot at the bottom of each number date on each block of the calendar. This circle dot will show that there are events on that specific day where as no circle dot will mean that there are no events on that specific day. This circle dot will also change color depending on how "busy" or how many events are on that day; maybe red for busy and green for less busy.

I worked on the calendar UI during Week 3: September 13 - September 19 and Week 4: September 20 - September 26. It also took a little more time here and there depending on if the UI wasn't working correctly with some new feature that I added. I think the UI went well. I learned that there are a lot of HTML templates online that a front end developer can draw inspiration from or a newbie like me can learn from. This made it a lot easier to create the calendar UI which I thought was going to be one of the toughest tasks to complete.



Nov. 1, 2020

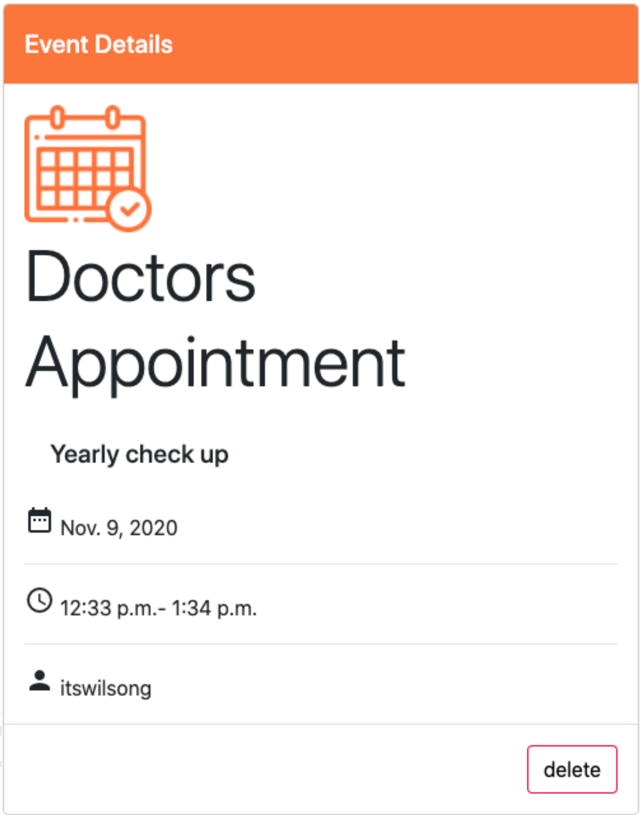
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9 Doctors Appointment ...	10	11	12	13	14 Car maint.
15	16 high priority test ...	17	18	19 different times test ...	20	21
22 Cousin birthday	23	24	25	26	27	28
29	30					

Figure 3: The full Calendar UI that is interact-able.

3.3 Add/Remove Events

The next feature that I implemented was the ability to add/remove events on the calendar. The user can either add an event to the calendar by clicking on a date block directly on the calendar or by clicking the orange circle with a plus sign in it on the bottom right. Both will bring up the same "Create Event" form as shown on Figure 5. At first the form had 5 fields: event name, event description, date of the event, start time, and end time (priority scale was added as a part of the priority system.) The user can type whatever event name and description they want. For the date and times, the user can either type it or just use the drop-down menu to create it. Clicking enter or create button will add the event into the database and onto the calendar visually. In order to remove an event from the calendar, the user would have to go into event details as shown on Figure 4. to find the delete button. This simply removes it from the database as well as the calendar.

Overall, I felt that this form was perfect in that it asked for everything that was needed to create the event. I can't really think of anything else that I could add to make it functionally better. Maybe some cosmetic changes would be fine or the ability to just select a time duration instead of selection start and end time but that would mean I would need to account for a whole new type of event that doesn't have any time limitations but only duration limitations.



Doctors Appointment

Yearly check up

Nov. 9, 2020

12:33 p.m.- 1:34 p.m.

itswilsong

Figure 4: Event details page that will open when the user clicks on an event on the calendar. This page shows all the specific details that a user will not be able to see otherwise.

I worked on this feature during Week 5: September 27 - October 3. It was relatively quick as I only needed to learn how to create a form with front end development as Django allows for easy creation of the form in the back end. All I had to do was to style it and then connect it. I felt really good after this as I finished the task fast and felt that my front end capabilities were already advancing.

3.4 Create/Join Groups

The next feature I implemented was the groups feature. It is required that user's have the ability to share and collaborate on a single calendar so the firstly, I would need to make it so that users can create and join groups. I also wanted it so that users can join as many groups as they wish ie. family calendar, work calendar, school calendar. At first I was thinking of some way to have users be able to create a group then invite friends to it but then that would require some sort of friends system which I haven't thought of implementing just yet. My next idea was to have some invite link similar to Zoom or discord where users can send out links to friends which they can click on and it would automatically make them join the group. I also saw this as too complicated so I thought of using randomly generated codes. So the idea is that when a user creates a new group, a code will be randomly generated and stored to the database. This code is the link to the group. So if a user wants to invite another user to their group, all they have to do is to give them the group code for which that user can just enter on a form field shown on Figure 7. when they click the button "Join a Group" shown on Figure 3. This way was very simple to implement because all I

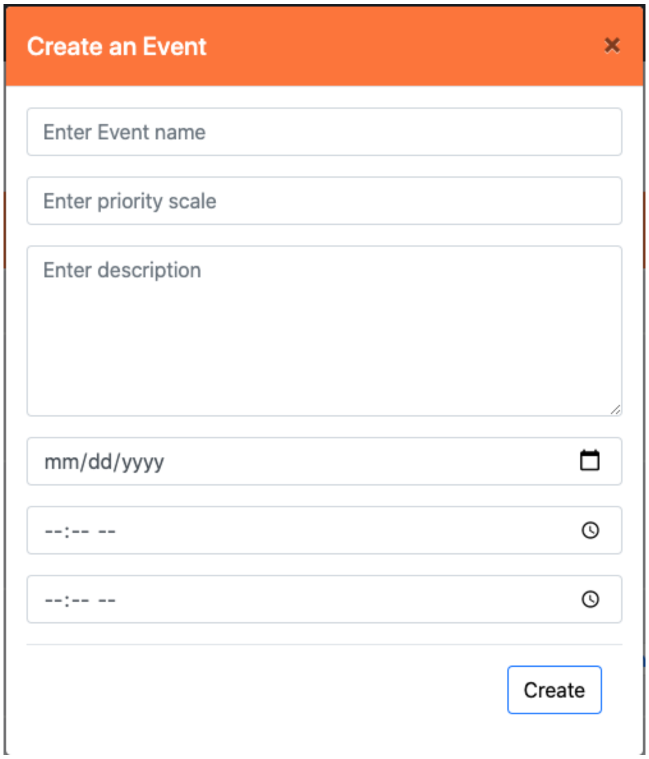


Figure 5: Event details page that will open when the user clicks on an event on the calendar. This page shows all the specific details that a user will not be able to see otherwise.

had to do was to do a look up and search the database for the code then add them to that group. Users can also see their current joined groups by clicking the "Joined Groups" button which will bring them to a list shown on Figure 8. Shown on Figure 9. is the groups credentials page which can be accessed by clicking the dashboard button in the drop-down menu under the user's name. It just shows the group codes for each group that the user has created.

Overall, I felt like this feature was very successful. The way that it functions was much simpler to anything I thought of before implement the feature. Simpler is very important as it allows it to be more user-friendly. Perhaps to make this feature better, I can add other options so the user can choose how they want to inboard users to their group. Especially, if I were to incorporate some sort of friend list system then I would also need to add a invite friend to group option via the friends list

I worked on this feature during Week 6: October 4 - October 10. This feature was easier for me to implement so it only took one week which felt really good as well. I just used a variety short form fields for the front end and just connected it to insert and search query's in the back end to create/search the groups.

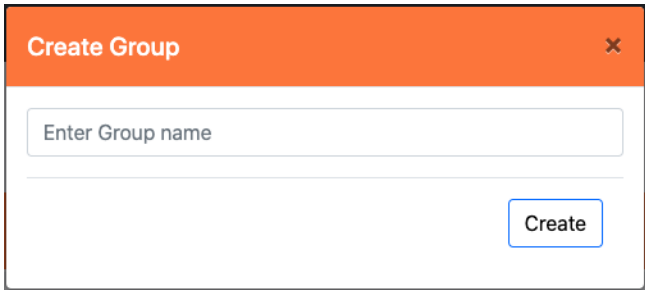


Figure 6: Form field that allows the user to create a group.

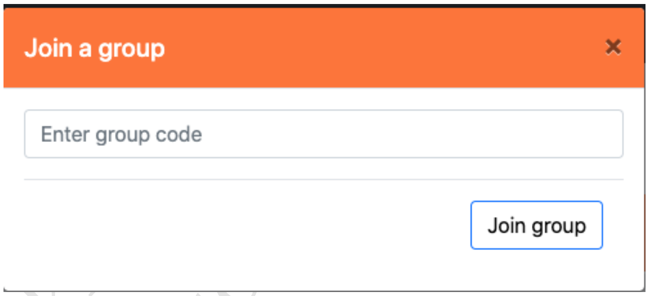
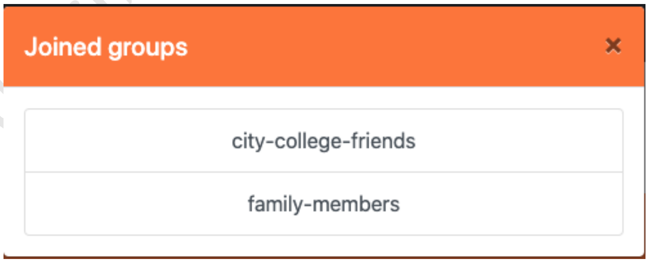


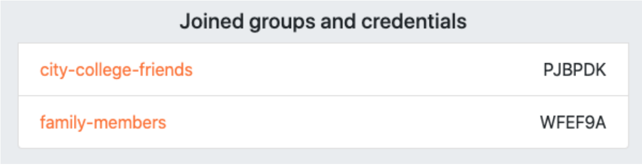
Figure 7: Form field that allows the user to enter a group code and then join the group.



city-college-friends

family-members

Figure 8: This is what a list of the user's current joined groups would look like.



Joined groups and credentials

city-college-friends	PJBPDK
family-members	WFEF9A

Figure 9: A groups credentials page which can be accessed by clicking the dashboard button in the drop-down menu under the user's name. This page shows the users the code for each group they created so that they can send it to other users for them to join.

3.5 Shared Calendar

Implementing this feature was really easy since I already implemented the groups feature which was a major part of this feature. All I had to do was assign a calendar instance to each group just like how each user would have a different personal calendar. So it was just a little tweaking in the back end database and to the groups code when it gets created and it was all done.

I worked on this feature during Week 7: October 11 - October 17. I didn't require the whole week so I was able to use some time to do research and brainstorm for my next feature which was the priority system.

3.6 Priority System

The next feature I implemented was the priority system. This would be a huge feature that would make my calendar application different than others. The priority system allows some events to hold higher precedence over other events. This feature is intended to allow parents to schedule doctor appointments for their kids even if it takes place during another event because a doctor appointment will always have higher priority over anything else. It will allow parents to clearly see when is the best time to schedule appointments because schedules can get hectic and confusing at times. Others can use this feature in other ways as well such as making their work days the highest priority so others can see that they are inflexible at these times and if they want to plan an event, another day or time has to be chosen.

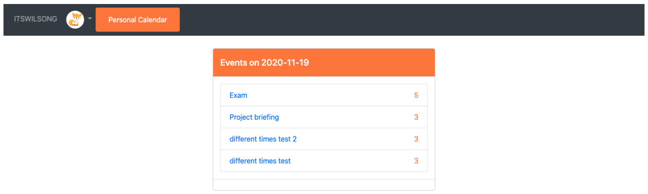
As shown on Figure 5. when a user creates an event, they have to enter 1-5 for the priority scale. This number denotes how important that event is compared to others with 5 being the highest. I felt that 5 priorities was a good number as there was not too much nor too little.

So the way it works is that when an event is created with low priority and then another event event is created with a higher priority on the same date with overlapping times then the higher priority will be considered more important and will be the one shown on the calendar. As you can see on Figure 11. some events have a "..." under the event name. This means that there are other events on that date and it may have lower priority than the event displayed on the calendar. When we click on the "..." we will be directed to the priority order page as shown on Figure 10. which shows all the events on that date with their priorities. The user then can click on each event to see their details.

Another scenario happens when two users create two different events on the same date with overlapping times. This will cause a warning letting the user know that they have overlapping times as shown on Figure 12. The event will then not be added.

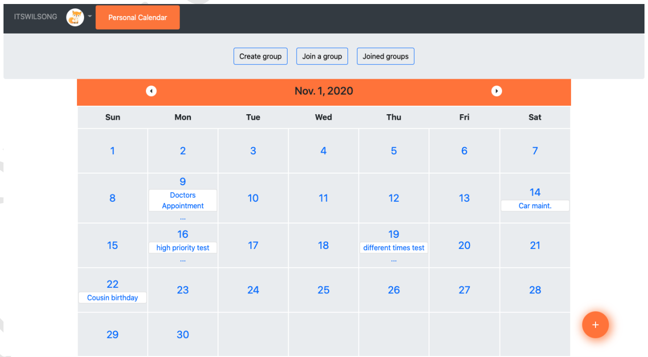
Overall, I don't feel too satisfied about the priority system so far. It was a lot tougher to implement than I thought because there are a lot of edge cases and scenarios I have to account for. Also, the way I wanted to implement it was to have it be more interact-able with each user to determine the priority scale but I was unable to do that due to my experience and time constraints. I had some different versions of the priority system but this was one that was actually functional to demo. But this is a feature that I will need to work more on and refine. This may include re-thinking the whole idea as well to make it better.

I worked on this feature during Week 8: October 18 - October 24 and Week 9: October 25 - October 31 and Week 10:November 1 - November 7 and Week 11:/November 8 - November 14. This was a feature that took the most time because I had so many iterations of it where it didn't work well so I scrapped it. I needed to at least get something to work for the demo so I settled for this version for now but I kept working on it but I was unable to get something better to work.



Events on 2020-11-19	
Exam	5
Project briefing	3
different times test 2	3
different times test	3

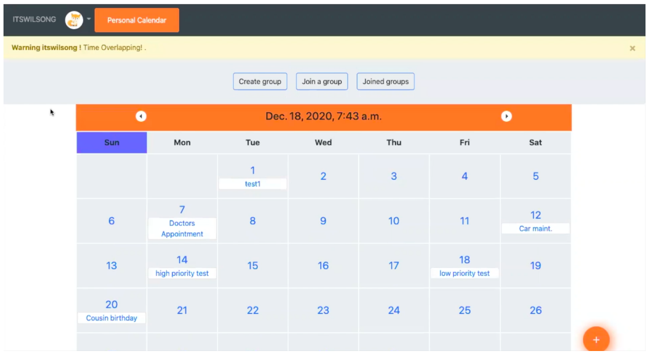
Figure 10



Nov. 1, 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9 Doctors Appointment ...	10	11	12	13	14 Car maint.
15	16 high priority test ...	17	18	19 different times test ...	20	21
22 Cousin birthday	23	24	25	26	27	28
29	30					

Figure 11



Warning itswilsong ! Time Overlapping! .

Dec. 18, 2020, 7:43 a.m.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 test1	2	3	4	5
6	7 Doctors Appointment	8	9	10	11	12 Car maint.
13	14 high priority test	15	16	17	18 low priority test	19
20 Cousin birthday	21	22	23	24	25	26

Figure 12

3.7 Auto-Decide System (Not in Demo)

A feature that I also tried to implement was the auto-decide feature where users can have the option for the calendar to automatically fit some event into their schedule. The parameters would be the time duration it takes for the event then the optional time constraints for when the user starts and ends their day. This would allow users to slip in small things they need to do in their busy schedule such as picking up something from the store.

I worked on this during Week 12:November 15 - November 21 but I did not have enough time to finish. I believe this feature would be easier to implement than the one before but due to time constraints I wasn't able to get it functional to demo.

4 EVALUATION

I felt that I completed a lot for a project that I think was quite ambitious for one person. I had to learn and research a lot just to complete even the simplest tasks but I am still disappointed in my progress because I don't have all the features that I set out to finish flushed out especially the priority system which was suppose to be a very unique feature.

4.1 Next Steps

I still want to be working on this project and I have a lot of other features that I want to be implementing for the application to be completed.

These features include:

- Friends List - It will allow users to add each other to a friend list. This will allow for easier group feature as well as a chatting system which will make this application more "Social".
- Better Groups UI - I think the groups UI including the groups credentials page could be more visually appealing with more features such as different ways to invite people or a way to remove people from your group. A whole page dedicated to the groups feature would most probably be a lot more useful.
- Profile Customization - The ability to change your username, email, phone number, password etc. as well as visual customization's on the calendar itself. I want it to feel more personalized to you or your group.
- Invite People - This would mean the ability to invite people to your event. This means that this would also put that specific event on their schedule with a certain priority making them having to plan around it because it might be a very important event that the user has to attend as well.
- Notifications - Using Twilio Sendgrid we can easily send push notifications to the phone or through email when an event is about to occur for the user.
- Calculation of route - With the use of the Google Maps API, it can calculate the time it takes to go from one place to another so the feature would help further plan out the user's day by showing them how much time they need to travel from one event to another. This feature will allow users to enter a location for each event that they add to the calendar and then would check if they can make it to that next event that they want to schedule. This sort of feature has a lot of complications so I will make it just an option for users so enable because I still want to keep the user-friendliness while allowing for extra capabilities for others that want it.

4.2 Different Approach

Another whole approach I thought of doing was to switch away from using Django and create the application using Dart and Flutter. I was very inspired by this one YouTuber that I came across who was able to create beautiful mobile applications within a few days. The applications were also cross-platform enabled making it even better because I want anyone to be able to use this app to make their live easier. I also seen that Dart and Flutter are now becoming very popular even more so than React as a Mobile development language and I was very intrigue to learn it because it may be the next most important and wanted language in the industry. If I were to create my application using this language then it also puts me far ahead of others in the field. Also the style of the applications that can be created using Flutter looks very similar to the simple style that I want for my application. So I may want to switch the project to by developed using Flutter once I start learning it.

5 CONCLUSION

I think that the project turned out less than I had expected in the beginning but due to some circumstances and personal issues, I felt like I did complete this project well enough for now. In the previous prospectus, I had set the goal to complete all the main features, including the Calendar UI, adding events, creating groups feature for shared calendar, priority system, and auto-decide feature but the priority system was much more difficult for me than I thought so it took a lot longer so my implementation for the auto-decide feature was not ready for the final demo. Though I missed out on this one requirement feature, I still felt that the application was somewhat complete. The best part really was just learning the skills needed to do this project. I invested a lot of time into researching and learning Django, JS, HTML, and CSS. All of these technologies I have not really used much in the past or received exposure to, especially, the frontend technology. I'd say that I'm very strong when it comes to the backend so learning Django was more about reading the documentation and with my experience in Python, it was very smooth. But when it came to the front-end and learning JS, HTML, CSS, it was tougher because I had absolutely no prior exposure to these types of languages and working with the front-end. But it became very fun when I started to get a hang on it since everything started to fall into place once you learn the basics. I was able to even transition these new skills to my internship as well. Besides tech skills, I've also developed the knowledge on how to tackle a new language and found my way of learning it in a short amount of time which is very important in the ever-changing computer science field. There's always new and better technology that I will have to adapt to quickly or fall behind with the old. Overall, I think that my biggest obstacle was learning new front end technology which was absolutely necessary for my project to succeed. But in the end I think my arsenal of skills expanded dramatically and that these skills and this project will prepare very well for the work in the industry. I will be continuing this project in my free time as I think that it has a lot of potential.

Start the appendix with the "`appendix`" command:

```
\appendix
```

and note that in the appendix, sections are lettered, not numbered. This document has two appendices, demonstrating the section and subsection identification method.

6 REFERENCES

[1]“Google Calendar API” Google Developers, https://developers.google.com/calendar

[2]“Python Google Calendar” PyPi, 12 Feb. 2018 https://pypi.org/project/python-google-calendar-api/

[3]“Django” Django https://www.djangoproject.com/

[4]“Bootstrap Introduction” Bootstrap https://getbootstrap.com/docs/4.5/getting-started/introduction/

[5]“HTML Tutorial” W3Schools https://www.w3schools.com/html/

[6]“CSS Overview” W3Schools https://www.w3.org/Style/CSS/Overview.en.html

[7]“CSS Introduction” W3Schools https://www.w3schools.com/css/

[8]“JAVASCRIPT CSS CALENDAR” CSS Script https://www.cssscript.com/tag/calendar/

[9]“Django 3.1.4 Documentation” readthedocs https://django.readthedocs.io/en/stable/contents.html

[10]“HTML Forms” w3schools.com https://www.w3schools.com/html/html$_forms.asp$

[11]“Your first form” MDN Web Docs https://developer.mozilla.org/en-US/docs/Learn/Forms/Your$_first_form$