Bernard Snowden
Chris Madrigal
Prateek Kukreja

Final Reflection

What went well:

Communication is key in any group project and we feel communication was one of our strong points. There were not any sort of surprises we had to deal with, if there was a problem our members reached out to the team. When code was pushed, the other team members usually take the time to look at the results and give feedback. There were no unanticipated absences during team meetings either. All in all, the the ensured communication allowed us to achieve a smooth workflow the last few weeks.

Project planning was also something we did a good job on. Whenever design was a topic, we all sat down in our meetings and either drew them up on the chalkboard or used the wireframing tools taught to us in recitation. Ensuring that we all had the same idea in terms of look and design allowed for faster work on the looks of the UI when we eventually got to it towards the end.

Work allocation was done pretty well too. This was also a byproduct of project planning as our design allowed for easily modular components, which then could be divided amongst our three members. While some tasks were definitely harder than others, we ensured that we can still dynamically re-allocate tasks when needed.

What could have been improved?:

There are several aspects of the project that we could have improved on. UI safety and error messages could have been provided in the implementation so that users would no when wrong inputs were given by them. This would have also made our system more learnable. A second focus on the project that we could have improved on was providing a better security mitigation against csrf attacks. In addition to improved security, the UI on our main activities page could have been more intuitive in the sense that we showed all of their activities for each day ordered by time. This would allow users to immediately visualize their day and also immediately see if anything is wrong about any possible inputs that that they gave. If a user were to see the discrepancy then they could remove that event and add in a correct entry. One last item for improvement would have been code quality and make sure we adhere to code quality norms from the beginning.

What would we do differently?:

If we were given the chance to re-do the project, we would have changed our approach to working on our core algorithm. We would have tested running through some examples without writing any code, meaning we would have created potential schedules by hand. In doing so, we would have learned the complexities of the algorithm before writing any code and this would have prevented us from the coding rework we had to do during implementation. We would have also liked to test with potential users to learn how we could improve our design.

Bernard Snowden
Chris Madrigal
Prateek Kukreja

Key lessons learned:

Team:

- Learned a lot about Diabetes and the medications involved in treating Diabetes
- The pain points of those with chronic illnesses
- Earlier testing of algorithms
- Consistent early user studies on the app

Individual:

Member	Lesson
Prateek Kukreja	Modularize work as much as possible because it'll make debugging significantly easier and a more manageable codebase.
Chris Madrigal	Early code review is always good.
Bernard Snowden	Task management is essential for group projects; UI can always be improved;