INST377 FINAL PROJECT

Collab Schedule Time App

Team members:
Anthony J. Aruldoss: anthonyaruldossjr@gmail.com
James Geleta: jimmygeleta13@gmail.com
Robert Barclay: rgbarc@gmail.com
Evan Joyce: ejoyce74@gmail.com
Hamdi Kamus: hamdikamus@gmail.com

Link to Github Repository for development: https://github.com/ejoyce1/INST377

ec2-52-202-163-26.compute-1.amazonaws.com

Information Problem:

All of our team members are in the iSchool and each semester have to complete group projects in multiple different classes. Part of being in a group project involves meeting outside of class. We have all first hand seen how hard it is to find the optimal time to meet with groups. The biggest problem is that every group member has a different schedule. Some may commute, have more classes than others, or work which prevent people from being able to meet. Currently, it is hard to schedule a time to meet, as our team as well as many others resort to simply stating different times that they are available to meet in a group chat and then selecting a time that works for everyone. It may take long for everyone to respond with their times as well, and by the time everyone has put in their input, a lot of time may have passed by. This may be a problem when it comes to deadlines, and therefore having technology take care of this problem could make it smoother and more efficient.

Strategies/solutions to solve the problem:

Our group decided to make an application to resolve the problem of finding a common time to meet when doing group work. This would be something that can be accessible online so that every student with internet access can be able to contribute to the form. This app allows a group leader to "sign up" and create a group account. They will create a group username, group password and add their email. With this group account, any of the group members can sign in. Upon signing in, group members are greeted with a quick and easy form that they can complete. The form then finds the optimal meeting time and displays it back to the user. In order to effectively gauge everyone's availability, we deployed a strategy that involves getting as much

information as possible from every user. This includes information about commuting, whether the person works full-time, part-time, or not at all, and having every time and date available for selection. Once all of the information is compiled, an optimal meeting time is given.

Rationales and justifications on system design and technology:

The rationale behind using Bootstrap for the HTML and CSS is that it allowed us to make the application look formal without becoming too complex. We decided to make it a simplistic design because users do not want to click through a multitude of buttons to get to what they want. Having the design be simple allows for one main page where all of the information is stored and it is easy to read. Think about when you surf the web for shopping or news. Many sites are complex and have multiple page levels and excess information. Our application provides a clear path to the desired information without lots of advertisement.

How the final system solves the problem:

Our final system solves our information problem because it finds the optimal meeting time for a group based on the information given. The system allows for a group leader to create a group, and afterwards every member of that group inputs their availability, as well as if they commute and whether they work full time, part time, or don't work. It also allows for comments which the group leader can view. These additional questions aside from their availability indicates how flexible an individual's time is. Our system also takes in input for the worst time to meet as well, which all helps to find an optimal time. At the end, our system matches available times with all group members inputs to select the meeting time, a time where everyone in the group is available, which is how our system solves the problem of finding meeting times for groups.

Challenges faced and the impact on the final design:

The first challenge that we face is implementing a session id. After discussing with the professor, we came to the conclusion that for now it would be easier to create a login with username and password. Another challenge we faced was sharing code. After much deliberation we decided to use Github as our code sharing platform because it was easy for everyone to make an account and access all of the files. We also face challenges when trying to learn some of the frameworks we initially wanted to use. One of these was laravel which proved to be too hard to learn and time consuming so we decided to proceed with other frameworks.

In terms of the group itself, we faced the challenge of finding optimal meeting times to work on the project. This is part of the reason we wanted to do this project in the first place. It caused us to push back our work until we could all meet together and caused us to not be on the same page when communicating remotely.

Missing Functionality:

Our application has a fully functional login/logout system, and takes a posts user data to the database. The main function that would complete the app would be to aggregate data per group and find the most optimal times to meet based on every group members data. However, after extensive testing and experimenting this functionality was not fully completed. In light of that,

Future work:

For the future, it would be ideal to implement the google calendar API to allow those users to link this meeting to any other events in their calendar. It is also necessary to implement a chat communication system between users. This will allow members of groups to communicate on the application to talk about necessary arrangements or compromises for meeting times. Right now we have users accessing the system through a username and password but we think that it may be beneficial to be able to access it through a session id. In addition, it may be beneficial to implement a feature where users can go back and change their preferences. This will allow meeting times to be updated continuously. An alert system may be beneficial because it will notify all users when they get matched with a meeting time and if it changes and updates, it will send a push notification with the new meeting time.