Criteria A: Planning

Problem definition:

My client, the manager of a frisbee community, faces challenges in efficiently managing community information and fostering member engagement due to the lack of a bulletin board. The lack of management in the frisbee community can lead to the loss of members because of unpleasant experiences, such as the schedules of events not being clearly communicated to members. This, in turn, reduces the membership fees used for covering costs related to equipment and fields. It may ultimately result in financial loss and the dissolution of the community. The client has tried online spreadsheets for managing sign-ups and allowing users to initiate events; however, they were not effective as they became disorganized over time: the spreadsheet became cluttered, leading to confusion on the participant lists and event schedules. Moreover, changes made on spreadsheets aren't assigned to users, creating overlap in sign-ups. Currently, users don't have a space to share ideas or propose events. Furthermore, due to the lack of a dedicated communication channel, users can only communicate in a fragmented manner on social media. Additionally, the community lacks a clear schedule of events planned, causing members to miss important events. As spreadsheets have limited space, it's difficult to sign up for events or see other participants. (See **Fig. 10** and **Fig. 11** in Appendix A)

Proposed solution:

There are web and mobile applications; however, mobile apps can only be accessed on mobile devices. Web applications, on the other hand, are accessible on any device with a browser. Therefore, I propose a web application called "FrisbeeCrew" for my client to bring all the necessary features in one place and provide a space for all frisbee community members to engage in. This may allow better communication within members and prevent disorganization of information.

Many tools exist for creating web applications, including JavaScript, Python, and C++. I proposed Python as it has an extensive library selection, making it possible to meet my client's large-scale web app needs, whereas JavaScript lacks the same breadth of libraries, and C++'s libraries focus more on performance rather than rapid web development [1]. I opted for Flask among Python frameworks like Django because of its flexibility, which is crucial for specialized web apps. Unlike Django, which is a strict and inflexible framework [2], Flask works with numerous tools, for example, Flask-Login and relational databases, allowing me to tailor the website to the community's requirements.

Since information of frisbee events need to be stored for users to view and sign up, I proposed to include a database in the solution. I propose MySQL, a relational database over non-SQL database because the data structures of the web application are complex and have relationships. This database fits my client's needs as it is designed to handle large amounts of data efficiently, which allows potential expansion of the community in the future. It ensures data integrity and provides robust security features, including user authentication and data encryption [3]. MySQL also offers excellent support for concurrent access, making it ideal for dynamic web applications such as social networks [4]. This combination is compatible with various devices and browsers, allowing a wide range of users to access the application.

Success Criteria:

1. The website has a login and registration system. *[issue tackled: "changes made on spreadsheets aren't assigned to users."]*
2. Users are able to create/read/update/delete discussions, including title and content. *[issue tackled: "users don't have a space to share ideas or propose events."]*
3. Users are able to create/read/update/delete events, including time, date, and content. *[issue tackled: "users don't have a space to share ideas or propose events."]*
4. The website has an email-like offline chat function that allows members to message each other directly. *[issue tackled: "the lack of a dedicated communication channel"]*
5. The website has a calendar for all events planned. *[issue tackled: "lacks a clear schedule of events planned"]*
6. Users are able to sign up to participate in events and see the number of mutual sign-ups with other users. *[issue tackled: "it's sometimes difficult to sign up for events or see other participants"]*

WORD COUNT: 523

Works Cited

[1] Bradley, Stephen. “Differences Between Modern Coding Languages: Python vs Java vs C++ vs Javascript.” *Be A Python Dev*, 26 July 2019, https://beapython.dev/2019/07/26/a-coders-toolbelt-python-vs-java-vs-c-vs-javascript/. Accessed 19 September 2024.

[2] Simplilearn. “Django Vs. Flask: Understanding The Major Differences.” *Simplilearn.com*, 13 August 2024, https://www.simplilearn.com/flask-vs-django-article. Accessed 22 August 2024.

[3] Data, Stitch. “PostgreSQL vs. MySQL: 9 key criteria to drive your database decision.” *Stitch Data*, 2022, https://www.stitchdata.com/resources/postgresql-vs-mysql/. Accessed 10 September 2024.

[4] Erickson, Jeffrey. “MySQL: Understanding What It Is and How It's Used.” *Oracle*, 29 August 2024, https://www.oracle.com/mysql/what-is-mysql/. Accessed 17 September 2024.