Raymond

Problems & Solutions

1. Working on integrating features that were designed proposed and outlined by someone else

   Solution: Consult with the person that made the feature and ask them exactly how they envisioned the feature when it was initially drafted.

What went well?

1. Being more accountable due to the nature of the work being individually driven, this allows me to allocate my time.

Brent

Problems & Solutions

1. **Lack of knowledge for the React framework.**

   During this sprint, I wrote a lot of front end Javascript code for React. I lacked basic knowledge of React hooks, components, and a basic understanding of the framework overall.

   Solution: Sacrifice 1-2 hours of free time per week ahead of time to learn the framework and grasp a basic understanding.

What went well?

1. Team collaboration this sprint went well, especially when coordinating times to meet in person to complete work items.
2. Although not pretty, I was able to create a functional front end for my feature that works as intended.

Gideon

Problems & Solutions

1. **Ineficient use of database calls in DAO Layer**

   Too many databases calls leads to a hit in server performance

Solution: Spend at least two hour to research how to efficiently combine multiple database calls into a stored procedure. This will lead to a greater increase in performance.

What went well?

1. **Collaboration**
    - There was good, clear, and concise communication with all members of the teams as to when we were going to work, and more importantly what we were going to tackle that day.

Vivian

Problems & Solutions

1. Since I was not the original creator of the feature, I was slightly unsure exactly what the original creator of the feature had in mind. For example, what would everyone who is collaborating on the itinerary have the itinerary items on their feature?
    a. Spend 1 hour talking to the original creator of the feature before deciding on major features.

What went well?

1. It was easier to find time to work since we did not have to collaborate with others.

Joshua

Problems & Solutions

2. **Inefficient use of database calls in DAO Layer**
   Too many databases calls leads to a hit in server performance
   Solution: Spend at least an hour to research how to efficiently combine multiple database calls into a stored procedure. This will lead to a greater increase in performance.

What went well?

1. Implementing input validation in the backend
2. Adding authorization for hyperlink sharing

Long

Problems & Solutions

1. **Lack of experience in web application deployment and server set up.**

   In the process of setting up our production environment, we encounters numerous errors such as permission and time out due to our lack of experience in deploying a web application which lead to a lengthy process of debugging.

   Solution: In order to improve on this, we decided that team members should do their own research on this topic in order to be more prepared when it comes to server configuration and set up to reduce time resources.

What went well?

1. **Setting up AWS Microsoft SQL Server.**

   We were able to back up the synchronous database server on the cloud to be in sync with our local database including, tables, stored procedures, functions and data.

Conclusion

There were a few things to take away from this sprint. While the nature of the work that we need to complete is mostly individually driven, we still need to consult with one another for a variety of reasons, some people in the team have to integrate features which they did not initially draft, so this leads to a loss in translation when looking at the BRD, to minimize confusion there will be a continuous feedback loop of the team members asking one another if their interpretation of how a feature on the BRD is describe is captured by the LLD and implementation. Additionally, the transition of our database from a local file system to a cloud production environment went quite smoothly. All of our tables, stored procedures, functions, and other database aspects were successfully transferred to AWS RDS. All members of the team were able to connect the remote database through SQL Server Management Studio and manipulate the data in our tables.