ADR-Second Sprint

Context

During this Sprint numerous features were planned to improve the overall user experience of the website. This meant new APIs had to be used in order to give rise to a better UI, security and persistent data storage. The stories planned for this Sprint where as follows:

- As a developer, I need to implement password hashing so that the user details are protected. Estimated duration: 4 days, Sprint Value:5
- As a developer, I need to implement a feature that sends email to users after they have signed up. Estimated duration: 4 days, Sprint Value:5
- As a developer, I need to implement a functionality that splits the expenses equally so that users can see their debts. Estimated duration: 2 days, Sprint Value:4
- As a developer, I need to implement a feature that notifies other users that a particular user has posted an expense. Estimated duration: 3 days, Sprint Value:4
- As a developer, I need to implement functionality that allows users to post expenses so that they can be captured on an expenses table. Estimated duration: 3 days, Sprint Value:5
- As a developer, I need to implement an option to allow users to reset their password so that they can access the website even if they have forgotten their password. Estimated duration: 3 days, Sprint Value:4
- As a developer, I need to implement a functionality that enables users to see what other users have paid for a particular expense. Estimated duration: 2 days, Sprint Value:4
- As a developer, I need to implement functionality to allow users to view debts so that they can make payments. Estimated duration: 3 days, Sprint Value:4
- As a developer, I need to create a payment option so that users can settle their expenses. Estimated duration: 3 days, Sprint Value:3
- As a developer, I need to implement a functionality that allows users to change their credentials so that their details can stay up to date. Estimated duration: 4 days, Sprint Value:3
- As a developer, I need to implement functionality that sends an email with a reset password verification code so that users can reset their password.. Estimated duration: 4 days, Sprint Value:5
- As a developer, I need to implement a functionality that allows users to see what others are owing for a particular expense. Estimated duration: 2 days, Sprint Value:4

Decision

The main architectural decisions that were taken during this Sprint involve the use of a persistent database (SQL) for data storage, Nodemailer for notifications, Bcrypt for securing user passwords, Express-session and Express-mysql-session to track users.

Status

Accepted.

Consequences

The use of the numerous APIs already listed above proved difficult and time-consuming. The team struggled with learning how to use the SQL database to make queries as many stories planned for this Sprint relied heavily on this. The use of Bcrypt to implement hashing was also tricky initially as the team member assigned to the story had no past experience with the use of Bcrypt. Nodemailer did not prove to be too difficult. In general, learning how to use APIs (esp. Express-session and Express-mysql-session) proved to be time consuming and thus meant that some stories had to be moved back to the last Sprint. These are mentioned in the Sprint Retrospective. The problems encountered in trying to use new API's resulted in the Sprint lasting 9 days. Besides the heavy cost in terms of time, the APIs used for this Sprint enabled implementation of a better user experience. This is noticeable from how most developer stories are essentially queries which capture important information that users might want to access easily. Using Bcrypt means that the user passwords are hashed and hence secured. The use of Nodemailer also allows for a better UI by ensuring that users get notifications of all activity in their group. The Sprint velocity achieved for this Sprint is 27/49.