

Note we need to push this onto github and then submit the github link (PDF)

Project: Spell Saver			
George Bauer	Leon Steinbach	Nhan Do	Trinity
Project Description	<p>Spell Saver is a banking software that allows people to have better spending habits. It aims to help people track their spending and know how well they are spending. There is a built in pie chart that lets the user develop their ideal spending habits, and then the user can compare them to their actual spending habits. The pie chart is directly connected to their transactions that are due so it will automatically show a pie chart based on the user's recent transactions. The transactions page lets the user put in all their recent or future payments. There are special categories that they can also assign each transaction. Then the calendar will show up when the transactions are due helping the user determine when they need to pay for something. If the user would like to they can also opt in to get emails for transactions that are due. There they will get email notifications about a summary of transactions that are due soon and transactions that have been paid. The account system allows for users to be able to login and logout of their account without their transactions being refreshed so they will always be accessible on multiple devices.</p>		

Project Tracker: [Link to Project Tracker](#)

The screenshot shows a project management interface with four main columns:

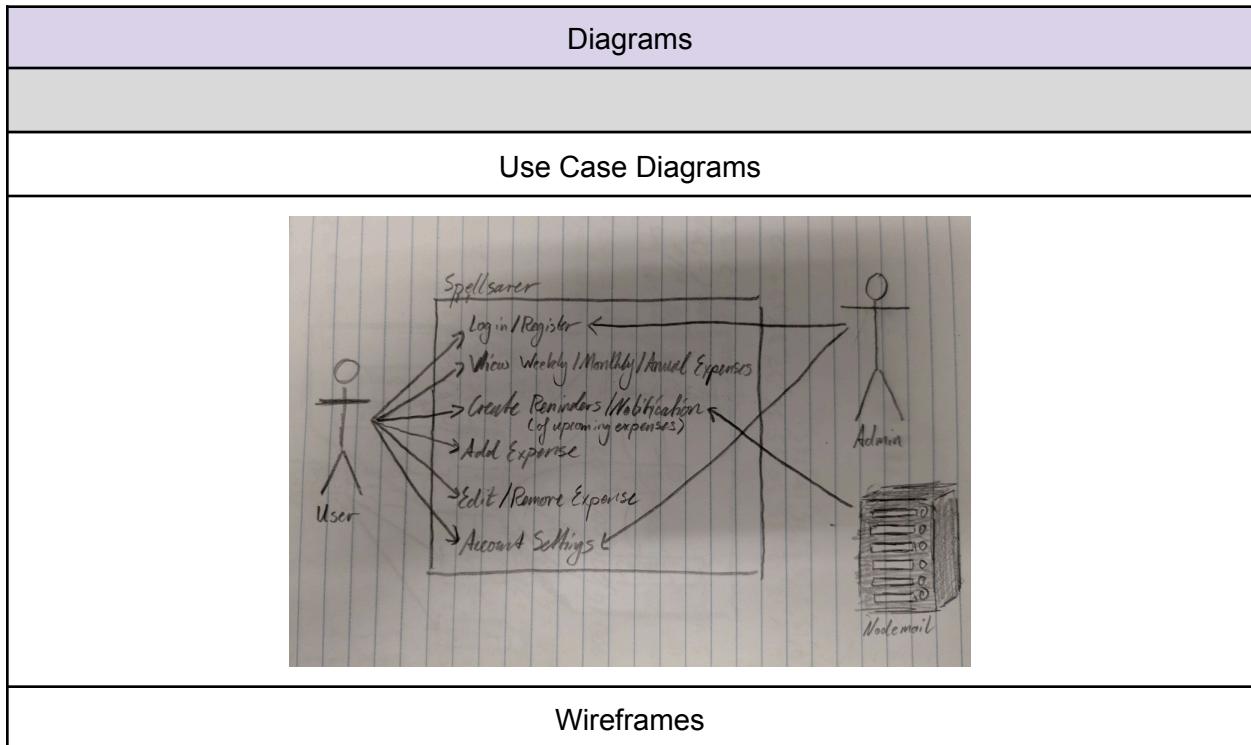
- Icebox:** Contains items labeled "Ideas for the future", "Changing the theme", and "User Profile".
- Todo:** Contains items labeled "This item hasn't been started".
- In Progress:** Contains items labeled "This is actively being worked on".
- Done:** Contains items labeled "This has been completed" and includes a list of tasks such as "Switching between Login/Register", "Show a history of recent payments", "Create a settings page", "Creating an Account", and "Having an FAQ".

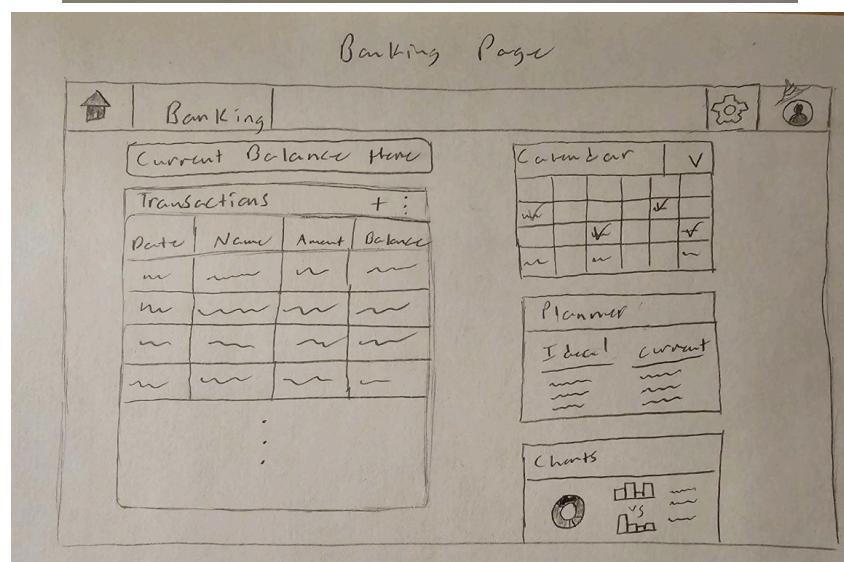
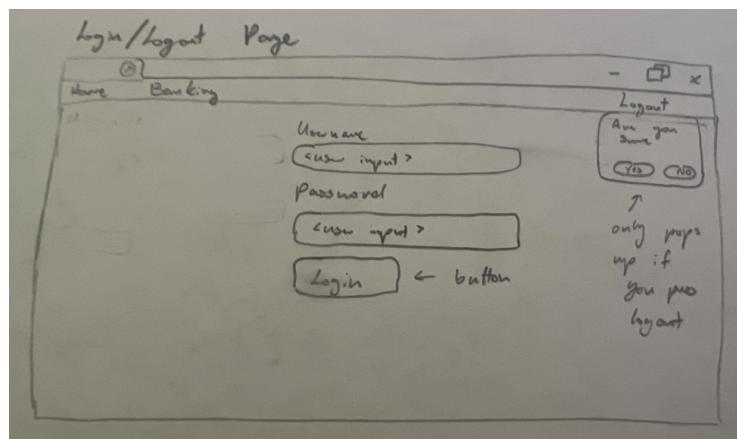
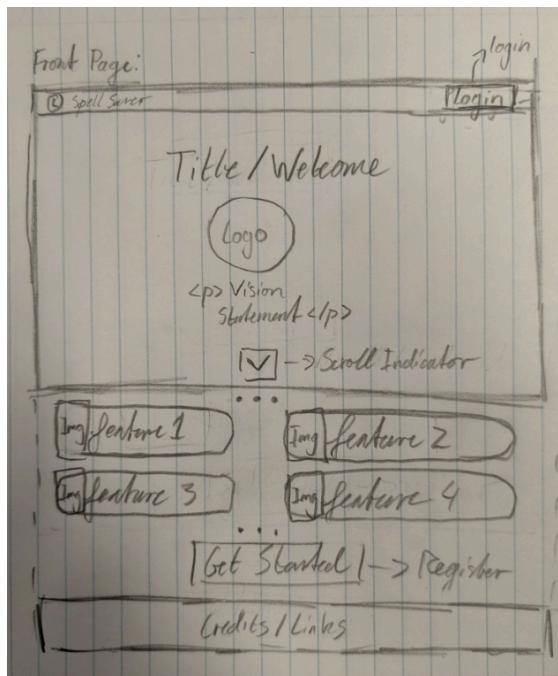
Each card in the columns includes a "User Profile" section and a "test9540/CSCI3308-GroupProject-011-3" reference. Buttons for "Add item" are visible at the bottom of each column.

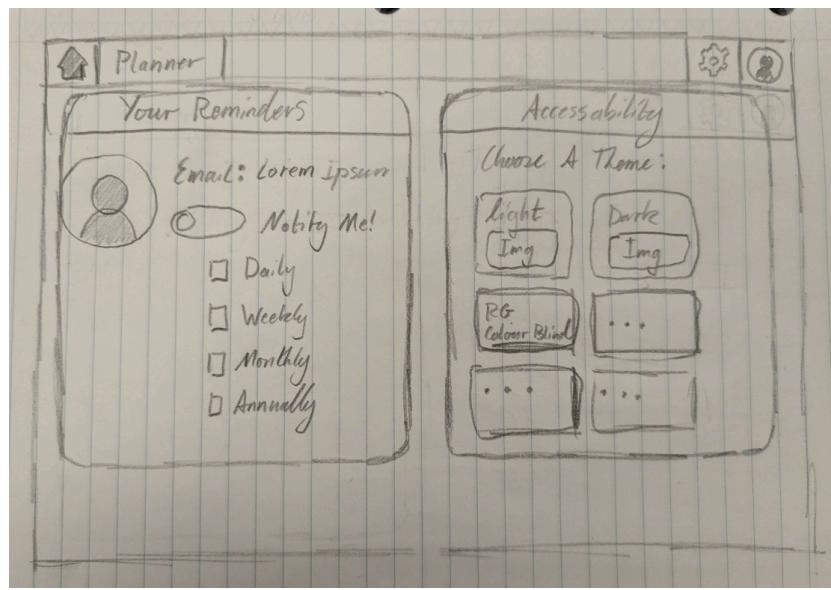
Demo Video	Demo Recording.mov
VSC	https://github.com/lest9540/CSCI3308-GroupProject-011-3

Contributions (no more than 100 Words)	
Name	Description (no more than 100 Words)
George	I focused mainly on the transaction history feature on the banking page. Users were able to input a set budget amount and then add transactions to their account. The transactions would then be displayed in a table on the banking page, automatically calculating deductions from the budget. The transactions history would populate the pie charts and calendar on the banking page as well. I also spent time maintaining the GitHub repository, dealing with merge conflicts and issues with rebasing.
Trinity	I worked on having redirects on the login page and the logout page. If you were on the login page it would have a popup. After that I worked on the calendar for the banking page. Which would link up to the transactions. So when clicking on the day on a calendar it would show the corresponding transaction that was due. It would also let you switch the months on the calendar to show future transactions. I also added a today button on the calendar that would take you to today's date and transactions.
Nhan	My focus was on setting up the pie charts, frontpage and bug fixes when its

	<p>needed. For the pie charts I had set up one where the user can create their own by inputting percentages for each category (there are 4) and another pie chart that would extract from the transaction history to calculate the percentages from actual data. As for the frontpage, I had set up section scrolling, anchor point links to automatically scroll down to them and generally some decorating for it as well. Other than that, I helped with making test cases and testing/fixing bugs.</p>
Leon	<p>My primary focus feature throughout the project was creating an opt-in mailing service that sends spending summaries. This was split into two sub features; the option to opt into summaries which necessitated a new boolean entry on our user table and send an affirmation of opting-in to the user's email, and a background process that interacts with the database to send out summary emails on a regular schedule to all users opted into summaries. I also spent time trying to maintain the javascript pages and doing simple bug fixes on broken test cases.</p>







Test Results	<p>User Register Page Test Cases: 2 test cases that utilizes a positive and negative test case to check whether or not there is already a user (username) that has already been registered. For the positive test case, if there isn't anyone registered with the username it will return a 200 code for success. Else for the negative test case, if the username is already taken, it will still return a code 200 but alongside with a message that indicates that the username was already taken.</p> <p>User Settings Page Test Cases: This test case tests out if the user has enabled EmailOptIn (NodeMailer) if they were successfully able to connect to the NodeMailer it will return a code 200</p> <p>Redirect/Render Test Cases: 2 test cases for testing the redirects and render of pages using the login page. Its purpose here is just to test if the basic renders and redirects would work as normal to help with bug testing. As setting up test cases for this ensures that the bug simply isn't a problem with the render/redirects. The results should return a 200 code for a successful render/redirect</p>
--------------	--

Deployment	https://spellsaver.onrender.com/front
------------	---