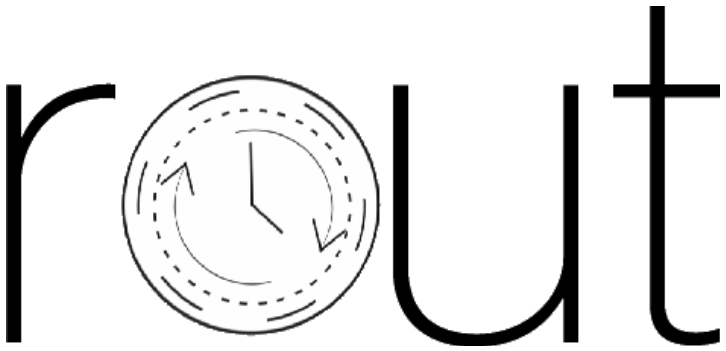

CS 30700 Sprint 1 Retrospective
03.15.2021

Team 17 :
Steven Bass
Luke Irons
Logan Sweeney
William White
Austin Wilson

What went well?

In general, nearly everything went well. We got the core of the website implemented with the routing feature and finished the login/registration feature with the DB. The website has few errors and is navigable easily. All in all, we did a pretty good job!

User Story #1

As a user, I would like to register a new Rout account.

#	Description	Estimated Time	Owner
1	Create UI element for account creation	2 Hours	Logan
2	Implement client-server-database communication protocol for storing user information	2 Hours (each)	Steven/Logan
3	Securely salt and hash stored passwords in the database	2 Hours	Steven
4	Use Let's Encrypt free SSL to encrypt communications between client and server	2 Hours	Steven

Completed:

Login sessions are stored in the DB and kept for up to 2 hours or when the user chooses to logout. Passwords are hashed and stored in the DB and correctly log the user in when combined with the username in the login screen.

User Story #2

As a user, I would like to login to my Rout account.

#	Description	Estimated Time	Owner
1a	Create login page	2 Hours	Logan
1b	Create invalid login UI	2 Hours	Logan
2a	Handle login requests server side	4 Hours	Steven
3	Keep user logged in by utilizing cookies	3 Hours	Logan

Mostly Completed:

Again, login session is stored in the DB and cookies are also assigned to user login sessions to keep them logged in regardless of if the tab is closed or if the user navigates away from the login screen. Fuzz tests were not implemented, but will be implemented for future sprints.

User Story #3

As a user, I would like to logout of my Rout account.

#	Description	Estimated Time	Owner
1	Create logout UI element	2 Hours	Logan
2	Handle logout requests server side	2 Hours	Steven
3	Create cookie system for keeping user logged out between sessions	2 Hours	Logan

Completed:

User session and cookies are deleted from the DB upon logout or when user session reaches 2 hours in length.

User Story #5

As a developer, I would like a MySQL database for storing account information

#	Description	Estimated Time	Owner
1	Install MySQL on AWS E2 machine	30 min	Steven
2	Configure security and user settings	30 min	Steven
3	Install and configure PHPMyAdmin	1 hr	Steven

Completed:

MySQL database has been created and properly stores user information.

User Story #6

As a user, I would like to be able to customize my nickname on my profile.

#	Description	Estimated Time	Owner
1	Create UI element for allowing users to change nickname	2 Hour	Logan
2	Modularize UI elements to display username/nickname	2 Hour	Logan
3	Manage database storage of user nicknames	2 Hours	Steven
4	Handle server side validation of nicknames	2 Hours	Steven

Completed:

Although changing the nickname was not possible due to UI integration issues when combining Austin and Logan's work, nicknames are properly stored in the DB upon user registration.

User Story #7

As a user, I would like to be able to customize my profile with a profile picture.

#	Description	Estimated Time	Owner
1	Handle server storage system for profile pictures	3 Hours	Steven
2	Create UI element for displaying profile pictures on Rout pages	2 Hour	Logan
3	Create UI element for allowing users to change profile pictures	2 Hours	Logan

Completed:

Although users cannot change profile pictures, they are able to store them initially and display the profile picture on login. Profile pictures are stored in the DB correctly and are easily fetchable.

User Story #8

As a user, I would like the UI of the website to be easily understood and used.

#	Description	Estimated Time	Owner
1	Use AWS to host the website	2 Hours	Steven
2	Create a UI panel to display Google Maps	2 Hours	Austin
3	Implement UI design for the main webpage	10 Hours	Austin
4	Implement text boxes for user data	3 Hours	Austin

Completed:

The UI of the website has been designed for the main page and format has been designed for all further pages needed. The page was designed with a navigation bar that would always be at the top of the screen so that all other possible pages would be easily accessible and returning to the main page would be easy, by clicking on the logo. The text boxes are also correctly logging the information imputed by the users.

User Story #9

As a user, I would like for outside temperature to be displayed in the route planning screen.

#	Description	Estimated Time	Owner
1	Find location for and create UI panel for temperature	1 Hour	Austin
2	Implement the outside temperature into the panel	2 Hours	Austin

Mostly Completed:

The outside temperature is currently being displayed in a status manor, set to West Lafayette, as the places or current location information of a user has not yet been implemented. I am calling on an API to obtain information given a city and state displaying the temperature. This is located in a weather box on the navigation bar that is show/hideable.

User Story #10

As a user, I would like for outside weather conditions (rain, snow, etc) to be displayed in the route planning screen.

#	Description	Estimated Time	Owner
1	Find location for and create UI panel for weather	1 Hour	Austin
2	Implement the outside weather into the panel	2 Hours	Austin

Mostly Completed:

The outside weather is currently being displayed in a status manor, set to West Lafayette, as the places or current location information of a user has not yet been implemented. I am calling on an API to obtain information given a city and state displaying the weather with an icon representing the information (i.e. sunny = sun, cloudy = cloud). This is located in a weather box on the navigation bar that is show/hideable.

User Story #11

As a user, I would like for date/time information to be displayed in the route planning screen.

#	Description	Estimated Time	Owner
1	Find location for and create UI panel for date/time information	1 Hour	Austin
2	Implement the date/time information into the panel	2 Hours	Austin

Completed:

The date/time is currently being displayed by taking the internal time of the machine. It is not linked up to a timer so it updates on open or showing of the weather box. This is also located in a weather box on the navigation bar that is show/hideable.

User Story #12

As a user, I would like the website to have a visually appealing color palette.

#	Description	Estimated Time	Owner
1	Research best color design for UI	1 Hour	Austin
2	Implement new colors in the UI	2 Hours	Austin

Completed:

The color design of the webpage was made to have a dark blue as the navigation bar and footer, with the dark blue color being used in many other locations like hovering of buttons, while the main pages have a white background. All UI elements were implemented with this color design in mind.

User Story #13

As a user, I would like to adjust the color of the UI to dark mode.

#	Description	Estimated Time	Owner
1	Design a dark mode version of the UI	1 Hour	Austin
2	Implement dark mode into the UI	3 Hours	Austin

Completed:

For the dark mode, I decided that changing the color of the navigation bar was unnecessary, so currently the dark mode will just change the background colors of pages to a blue that is only slightly dark, being easy to look at but still lighter than the navigation bar. Dark mode is saved on exit of the website and opening of the website and even things like opening on new browsers.

User Story #14

As a developer, I would like to implement a pathfinding algorithm which uses the Google Maps Directions API.

#	Description	Estimated Time	Owner
1	Create an algorithm that generates routes utilizing the Google Maps Directions API.	12 Hours (total)	William, Luke
2	Debug and test algorithm using unit tests	5 Hours	William

Completed:

Through use of the Google Maps Javascript API and the Directions API, we were able to implement an algorithm that finds a path from a starting point to an area located a certain distance away and then back to the same starting point.

User Story #15

As a developer, I would like to use the Google Maps API to create possible routes.

#	Description	Estimated Time	Owner
1	Develop a method to utilize the Google Maps API to pick a starting point and create a route.	6 Hours (total)	William, Luke
2	Implement method	10 Hours (total)	William, Luke
3	Test by creating routes and determining whether they are suitable running routes, debug etc.	3 Hours	William

Completed:

Using the Google Maps Javascript API examples, we created a method to allow the user to pick a starting point and passed that point to a method which created and displayed a route. We utilized several of the API's features which made sure the route only passed through places that can be walked on and avoided highways.

User Story #16

As a user, I would like to plot routes with specific waypoints to pass through.

#	Description	Estimated Time	Owner
1	Develop a method to manually utilize the Google Maps waypoint feature in the route generation algorithm.	10 Hours	Luke
2	Debug and test algorithm using unit tests	4 Hours	Luke
3	Connect the waypoint algorithm to the map UI	2 Hours	Luke

Completed:

Using the Google Maps Javascript API (and the DirectionsService object, with its associated waypoint usage), we were able to implement a waypoint feature that allows the user to place down specific waypoints on the map. When generating the route, the algorithm will use these waypoints as parts of the route pathing.

User Story #17

I would like to drop a pin to use as a start point.

#	Description	Estimated Time	Owner
1	Use a custom marker as a starting point when running the route-generation algorithm.	2 Hours	William
2	Test and debug the algorithm under the new criteria	2 Hours	William
3	Connect the marker choice to the map UI	2 Hours	William

Completed:

Based on an example from the Google Maps Javascript API, we created a method that added a marker to use as a start point wherever the user clicked on the map. The marker and its coordinates are added as the start to be sent to the routing method.

What did not go well?

User Story 2:

As a user, I would like to login to my Rout account.

#	Description	Estimated Time	Owner
2b	Write fuzz tests for user form fields	4 Hours	Steven

Not Completed:

Due to time constraints with deploying the web service, robustness testing for the login form had to be postponed. This is not a major issue, as manual testing covered the needs for the sprint 1 prototype, but robustness is an important consideration moving into sprints 2 and 3, and this task will be included in the backlog for sprint 2.

#	Description	Estimated Time	Owner
1	Create UI element for allowing users to change nickname	2 Hour	Logan
3	Create UI element for allowing users to change profile pictures	2 Hours	Logan

Not Completed:

Unfortunately, users cannot currently change their nickname or profile picture once registered. This is now an easy fix, but since we ran into issues merging the two UI elements together we were not able to develop a user information screen. Once that is implemented, we will easily be able to make SQL queries to change the DB entry.

User Story #4

As a user, I would like to be able to reset my password through email.

#	Description	Estimated Time	Owner
2	Create system for sending reset “code” to email	2 Hours	Logan

Not completed:

This one was a bit tricky. We opted to use nodemailer, a JS react library module, for our email service and successfully set up an SMTP gmail account to use for sending emails. Unfortunately, nodemailer kept erroring out whenever the smtp transporter was created and we were not able to fix the issue before the deadline. This user story will be fixed for future sprints.

User Story #9

As a user, I would like for outside temperature to be displayed in the route planning screen.

#	Description	Estimated Time	Owner
2	Implement the outside temperature into the panel	2 Hours	Austin

Not completed:

Locally the weather API works, but on our hosted server there is an error for asking for http requests which are not secure. Will have to change the requests to the API to be https requests.

User Story #10

As a user, I would like for outside weather conditions (rain, snow, etc) to be displayed in the route planning screen.

#	Description	Estimated Time	Owner
2	Implement the outside weather into the panel	2 Hours	Austin

Not completed:

Locally the weather API works, but on our hosted server there is an error for asking for http requests which are not secure. Will have to change the requests to the API to be https requests.

User Story 14:

As a developer, I would like to implement a pathfinding algorithm which uses the Google Maps Directions API.

#	Description	Estimated Time	Owner
3	Connect the algorithm to the map UI on the main web page.	2 Hours	William

Not completed:

Although we anticipated being able to integrate the routing algorithm within the main webpage during this sprint, we underestimated how difficult it would be to actually do this. This problem mainly stems from the use of Typescript in writing the algorithm code, which we cannot easily integrate with the Node usage of the main web page UI. This will be a problem addressed in Sprint 2.

How should you improve?

For the first sprint we did a good job in regular meetings that we held most Mondays, Wednesday, and Fridays. We were mostly on topic during these meetings working hard towards deadlines and communicating current problems and goals. We think that us meeting often and talking about what we were concerned about or having troubles with was a very good pattern we started with in sprint 1. We think it is good we started doing this early and we hope to continue working closely through sprints 2 and 3.

The greatest technical issue we faced was the integration of the individual work which had been done over weeks 1 and 2 of the sprint. The time this required is the main cause of our incomplete user stories. To remedy this going forward, we will be branching off of the sprint 1 prototype repo and ensuring that each member is responsible for the integration of their individual components in the larger whole as those modules are completed, rather than saving all integration tasks for the final week.

Going into the first sprint we were unsure of exactly what had to be done or what user stories we were going to pursue. This led to some flexibility and gray area work distribution of the sprint 1 planning document. Some members did tasks and user stories that were assigned to different members. Although we don't believe that this was completely a bad thing, we think that after getting a start on our project now, we can be more specific and follow the sprint 2 planning document more closely for sprint 2. This will hopefully give all members an even workload consisting of what they are planning to do for this upcoming sprint.

Overall, we believe that we had a very productive first sprint with almost no problems, except for integration. Going forward we hope the changes we proposed above will make our future sprints even more productive and will reduce the little errors and difficulties we had.