**‘Surf Safari’ Senior Project Plan**

**Quarter 2**

Kevin Moreland

**Overview of Quarter 1**

All except for the last deadline were met for quarter 1 of the project. The website now displays an interactive map that can switch between a satellite view and a topology/land data view (I will also need to add a view that visually shows swell data in the ocean). Users can register accounts, sign in, and sign out. Map markers update appropriately so that users can save surf spot markers and see them on their next login. Features that I did not yet implement fully in the backend are implemented in the frontend (the weather/swell information and NASA satellite images. Measuring distance between two points has not been implemented in the frontend yet).

**Updated Goals for Features**

Just as in the project plan for quarter 1, I have divided my list of features into two groups, ‘basic features’ and ‘stretch goal features’. The ‘basic features’ are the bare minimum features I need to be able to consider this a complete project. The ‘stretch goal features’ could add a lot of great functionality to the project and will be implemented if I am able to complete my other work on time.

Features that have been completed are in green, features that I have decided not to implement are in red, and new features or features moved into a different section are in bold

*Basic Features*

1. Live swell information (height, period, and direction) for any coastal point using the NOAA API (I am removing this because it is included in point #10. If I can get forecast data, I will also have data for the current day.)
2. Interactive map with satellite imagery
3. Topology data overlayed on map
4. Display of wind, air temperature, and water temperature (if available) at locations picked on map (UI is implemented, but test data is filled in, not real data yet)
5. Tide information
6. Ability to calculate distance between points
7. Ability to save marked spots on map with names and descriptions
8. Ability for users to create accounts
9. Search bar to jump to locations on the map
   1. Ability to use search bar to search for surf spots that were found and marked by the user (users assign names to surf spots they find)
10. **Forecast of surf at specific coastal point (Moved from stretch goal to basic feature because I realize this is an essential feature of the website, and it should have always been here)**
11. **Ability to see satellite images of a specific location at a specified date using the NASA Earth API (Moved from stretch goal to basic feature because this seems fairly easy to do and very helpful for spotting surf in remote locations)**
12. **Mark known popular surf spots on map (Moved from stretch goal features to basic features. This will make the website much more usable since when discovering remote surf spots, a user must be sure it isn’t already mapped. Additionally, this will allow the website to be used like a normal surf forecasting site for known surf spots, which could be very useful even if it isn’t the main purpose of the project)**

*Stretch Goal Features*

1. An overlay of the map showing wind currents
2. Displaying webcams located near the ocean using the Windy API for webcams
3. An overlay of the map showing all swells with their power and direction in the ocean
4. **Ability to download a report of all saved surf spots, or individual spots (Moved from basic feature to stretch goal because I realize this may not be very useful, and I would like to prioritize other features)**
5. **Ability to share information on social media about a surf spot discovered (will include a satellite image of spot, the user’s given name and description for the spot, and current swells hitting the spot) (Moved from basic feature to stretch goal because I realize this may not be very useful, and I would like to prioritize other features)**
6. Seafloor topology overlayed on map (Removed because I believe it would be difficult to get much value from this when determining the surf at a location and because after more research, I have found that this data would be difficult or impossible to find)
7. Information about dangerous ocean life or hazards specific to locations (sharks, jellyfish, unclean water, etc) (After more research, I wasn’t able to find a good database with this information)
8. Flight and/or drive time and cost estimates to travel to a discovered location from current location
9. (Not a feature) Host a virtual competition to discover the best surf spot with Surf Safari. The best, most remote spots will win (possibilities could be a cash prize generated by a contest entrance fee or recognition on the Surf Safari website).

**Timeline for Quarter 2**

I structured my timeline so that every 7 days, I have a new deliverable due. Before, I made the time period 10 days, but this didn’t make as much sense since my deadlines landed on different days of the week each time. Now, I know that I have a deadline almost every Friday (some are spread out more for more difficult tasks). I have frontloaded my work and made my deadline a couple weeks before the end of the quarter. This will give me a buffer for any unexpected circumstances. If I have extra time, I will work on my stretch goals listed above.

|  |  |
| --- | --- |
| **Goal** | **Deadline** |
| An API is setup to retrieve swell data from NOAA and calculate forecasts of swells for a location given its coordinates. | 4/16 |
| Weather data for any location can be retrieved and displayed. | 4/23 |
| Ability to see NASA satellite images at a specific given location and time. | 4/30 |
| Users can calculate distance between two points on the map | 5/7 |
| Known popular surf spots are marked on map. User can use the search bar to search for surf spots they have labeled. | 5/14 |
| Global swell map added to visually show swells and their height/period in the ocean | 5/28 |

**Goals for Quarter 2**

Once all the frontend work is finished and some of the backend is complete in Quarter 1, I hope in Quarter 2 to:

1. Finish the backend code for retrieving weather conditions
2. Setup the backend code for registering/logging in users
3. Setup a database in AWS to store user account information and data associated with accounts
4. Create a test suite for all my backend code to test edge cases and automatically detect errors before I release updates
5. Implement as many of the stretch-goal features as I can before the end of the quarter