SENG2021

Software Engineering Workshop

Trimester 1, 2019

Team ‘idk’

Final Report

Angela Chan z5210294

Madeleine Dobbie z5112961

Fangyi Guo z5175113

Hui Lin z5167876

Jessica Phung z5205460

Theme

Our app is an all in one beach safety app which allows Australian beach goers to access information regarding weather and beach safety. With 85% of Australians living within 50km of the coast, the beach plays an important role in Australian life. Our application takes away the stress of a beach visit by using an intelligent algorithm to calculate the relative safety of beaches in relation to other beaches in the area. We also provide more information on the beach including current and forecasted weather, directional services to get our users to the beach and also access to favoriting ability and community review system which allows user to view and contribute to reviews left on beach profiles. The intended audience of this site is all Australian beach goers however, we believe families who are looking for safe beaches, surfers who care a lot about surf conditions and tourists who would like to get some local knowledge about different beaches in the area will benefit the most from our site. Our app satisfies a market need as there are currently no equivalent beach information sites out there, particularly ones that deal with the safety of a beach.

Architecture

Programming languages: Python, JavaScript, HTML5, CSS3
Frameworks: Django, Bootstrap
APIs: Google Maps, Google Places, Google Directions, Google Geocoding, Dark Sky

Frontend: For our frontend, Bootstrap is used for designing websites through HTML templates. Our group has prior knowledge with HTML, so it allowed us to quickly implement a web page as it was easy to use and understand. These allowed for a simplistic styling for the user.
Currently we are using vanilla JavaScript because our team doesn't have a background in JavaScript, so this cuts down additional time needed to learn the frameworks after learning the basics of JavaScript. In order to interact with Google Maps and Dark Sky API JavaScript is crucial, and the features we currently need to implement can be applied without additional JavaScript frameworks such as React or Angular.
For the format of our website, we decided to use CSS3 in order to present a better appearance to our web pages because it provides vast range of options and also enhances the creation of our design. Cooperated with CSS3, HTML5 can perform and present more naturally and display with a satisfiable format which can help users to easily understand and utilize our websites.

Middleware: No middleware was needed for our website.

Backend: Our group decided to use Django for the backend of our website. The built-in admin allows us to easily manage the data and quickly build an app to present the data within a short amount of time. Since our group does not have a background with using database management languages, Django's object-relational mapper allows us to easily work with features that require

storage, such as a user system, rating system, and favorite system. Django allows JavaScript, CSS and HTML files to be easily linked to the backend. Also, our group has more familiarity with Python, which is why we chose Django over Node.js, as we can start coding with reduced time watching tutorials. Furthermore, Django allows us to create an application with high cohesion and low coupling because in Django the main application is split into sub applications (such as a user application and a map application). This allows individual team members to work on these sub-applications, while the applications are still linked together.

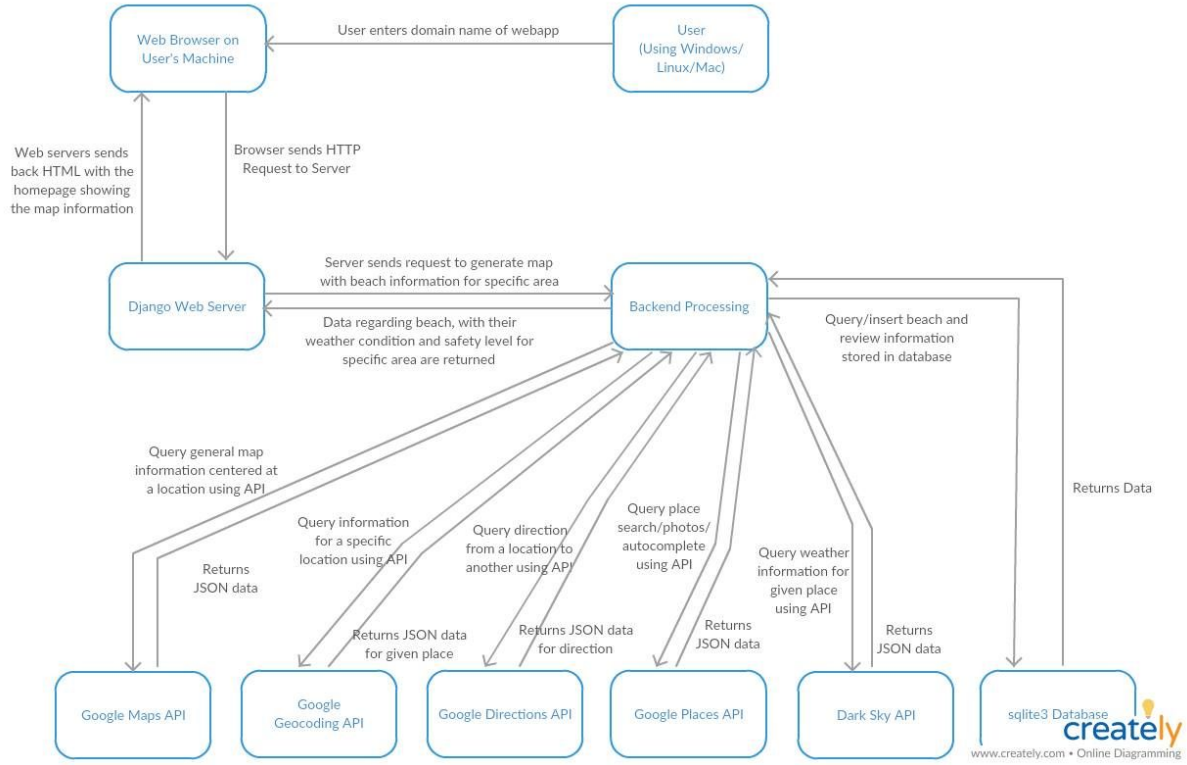
External Data Sources:
We are using the Google Maps and Dark Sky APIs as external data sources for locations of beaches as well as weather suitability of beaches.
Google Maps, Google Places, Google Directions, Google Geocoding: We are using these Google provided APIs to generate maps of beaches, and allow the user to search for beaches. While the Google Maps API displays maps of beaches, the Google Places API is used to retrieve details about the beach such as its address. Google Directions is used in our directions page for the user to find out how to travel to a selected beach. The Geocoding API was used to convert beach names back into coordinates. While there are other alternatives such as Mapbox we felt that these APIs had more additional features which could easily be integrated. For reexample, the feature of adding an autocomplete search bar onto the map to find beaches via Google Places is integrated with the ability to retrieve transport directions to the specific beach searched via Google Directions. If we weren't using Google APIs we might need to integrate multiple APIs that provide separate features from different companies, and this integration would be more difficult. Overall, the information retrieved from these APIs provides such as the complete address, name, and location of beaches allows us to return results on the homepage, as well as generate profile and directions pages for beaches.

Dark Sky: We are using Dark Sky Weather API to retrieve weather details on certain beaches. These details are then used to calculate the "safety" of a beach, and the details are also displayed to the viewer. While some API's may show little to no variance between nearby location, this weather API is well documented and more accurate than other weather API's we've tried. Dark Sky allows us to find information such as a beach's ozone and UV level, precipitation rate etc., which would be used as factors in calculating a beach's "safety". These details would also be displayed on a beach's profile page.

Browser used: Since Internet Explorer doesn't support any versions of ECMAScript above ECMAScript5, Internet Explorer would not be preferred. Our website would be able to run on Chrome, Safari, Firefox, and ideally as many browsers as possible which support more updated versions of ECMAScript.

Machine used: Ideally users can use any desktop machine regardless of operating system to access our web application provided they are using a supported browser. Our web application would not support mobile access as adapting the site for mobile would cost too much time.

Architecture Diagram



Key benefits + Summary: Overall our key architectural choices were made while considering time constraints, pre-existing knowledge of members, and most importantly implementation of features. Our choice of using the Django framework alongside vanilla JavaScript reduces the complexity that comes with learning many frameworks, for example in comparison to learning the MEAN stack. The benefits of choosing the external data sources of Google Maps and Dark Sky API is that these APIs provide more detailed and accurate information than most of their competitors.

User Stories

-

Requirement 1: Sites providing information on beaches don't display whether weather on a current day is ideal for visiting

Feature 1.1: View a colour-coded map based on ideal weather conditions, red being not ideal, green being ideal, yellow being somewhere in between
As a user **I want to** view a map that displays beaches within a certain radius of my location **so that** I can identify which beaches are suitable to visit based on today's weather
GIVEN that I am on the homepage of the site

AND I have pressed a button to enable location access
OR I have searched for a location
THEN I the homepage loads a colour-coded map with nearby beaches based on my current location or my searched location
Priority: 1

Requirement 2: Most sites don't allow users to provide and see other's thoughts and reviews for a particular beach

Feature 2.1: Users can provide reviews of visited beaches
As a user **I want to** be able to leave reviews of visited beaches **so that** other users can make informed decisions about the available beaches.
GIVEN that I am on the beach profile
AND I can click a button that redirects me to a rating page
THEN I can leave a review including a rating and comment
Priority: 7

Feature 2.2: Users can delete reviews they made
As a user **I want to** be able to delete reviews I posted **so that** I can take them down if my opinion of a beach changes.
GIVEN that I am on my user profile
AND i click delete review on the review I want to delete
THEN the review will be removed from the site
Priority: 13

Requirement 3: Many sites don't provide directions to nearby beaches.

Feature 3.1: By default, a list of beaches within a certain radius of my location will be displayed
As a user **I want to** view the closest beaches to me **so that** I can access the beaches with the least distance from me
GIVEN that I am on the homepage
AND I have pressed a button to allow location access
OR I have
THEN a list of beaches within my radius will be shown on the homepage
AND I can see how close they are in kilometers from me
Priority: 2

Feature 3.2 User can find route from inputted location to a certain beach
As a user **I want to** get directions to a specific beach **so that** I know how to access it
GIVEN that I am on a beach's profile page

AND I have inputted my starting location
THEN the map on the beach's profile page will display a route from the starting location to the beach
Priority 12

Requirement 4: Current sites don't allow users to filter beaches based on search filters

Feature 4.1: Option to check 'only view popular beaches' when viewing maps
As a user **I want** to view only well reviewed beaches **so that** it is easier to find beaches that other users have enjoyed
GIVEN that I am on the home page
AND I check 'only view popular beaches'
THEN the map shows only 4 and 5 star reviewed beaches
Priority: 6

Feature 4.2: Search bar to search beach by location
As a user I want to be able to search for a beach by location so that I can view information on a specific beach.
GIVEN I am on the homepage
AND I have enabled location access
AND I search for a location in Australia
THEN The homepage map displays all of the beaches closest to this location
Priority: 3

Feature 4.3: Filter beaches based on their safety level.
As a user **I want to** search beaches with my preferred safety level **so that** I can view beaches with a specific safety level.
GIVEN that I am on the homepage
AND I have enabled location access
AND I have checked my preferred safety level
THEN I will only see "green" beaches on the homepage map
Priority: 5

Requirement 5: Most sites don't allow users to save beaches that they prefer for future visits

Feature 5.1: Saving beaches when logged in
As a user **I want to** save my favorite beaches **so that** it's easier to access them for future visits
GIVEN that I am logged in
WHEN I am on the profile page of a beach
THEN I can press a button to save that beach

Priority: 10

Feature 5.2: Viewing saved beaches on a page
As a user **I want to** view my saved beaches in a separate page **so that** I can access them for future visits
GIVEN that I am logged in
AND I click on my profile page
THEN I can view a page listing all of my saved beaches
Priority: 11

Feature 5.3: Users can login to the site
As a user **I want to** login to the site **so that** I can save my favorite beaches to visit again
GIVEN I am on the login page
AND I have provided my login details
WHEN I click the login button
THEN I am able to login in to the site, and am redirected to my profile
Priority : 8

Feature 5.4: Users can sign up to the site
As a user **I want to** create an account **so that** I can access additional features such as a review
GIVEN that I am on the sign-up page
AND I provide my details to create an account
WHEN I click “sign up”
THEN I can see a confirmation that my account is created
Priority: 9

Feature 5.6: Users can remove saved beaches
As a user **I want to** remove the beaches I’ve saved **so that** when I’m no longer interested in these beaches I don’t have quick access them
GIVEN that I am logged in
WHEN I am on the profile page of a beach of a beach that I’ve saved before
THEN I can press a button to remove that beach from my favourites
Priority: 14

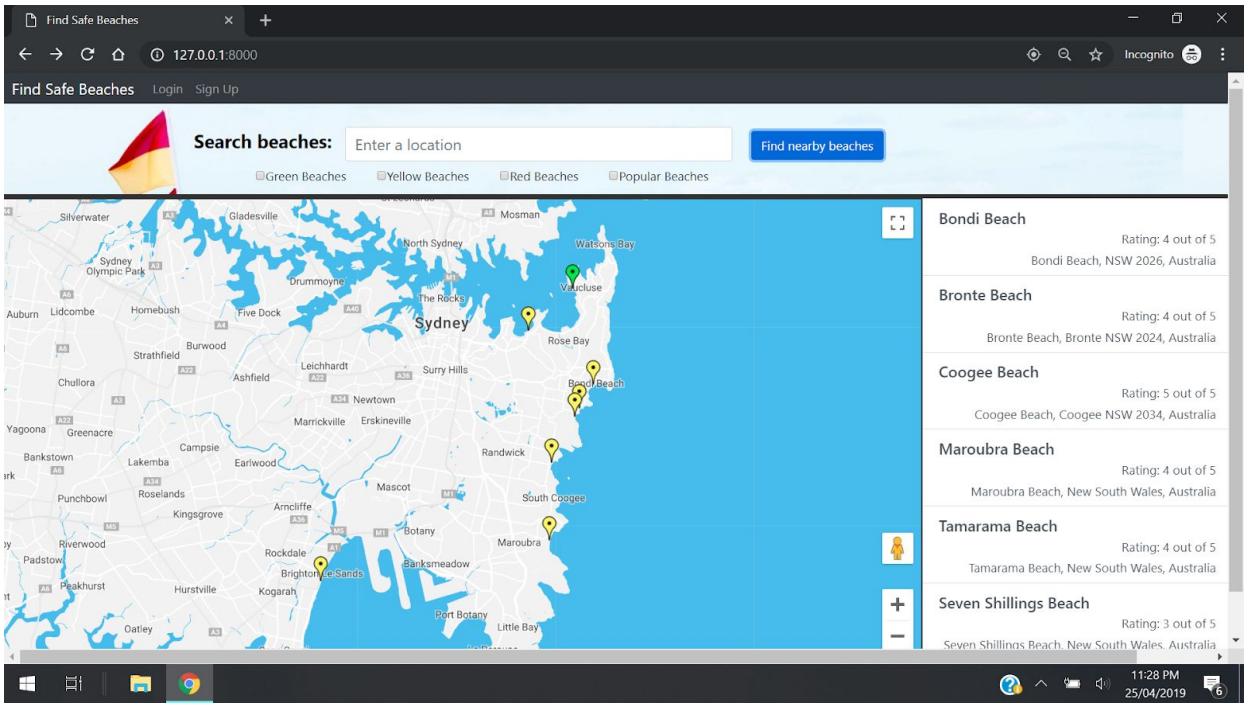
Requirement 6: Most sites don’t allow users to view beach profiles with details on their weather

Feature 6.1: View beach profiles
As a user **I want to** view beach profiles **so that** I can see an in-depth weather condition for today
GIVEN that I am on the homepage
AND I search for a location

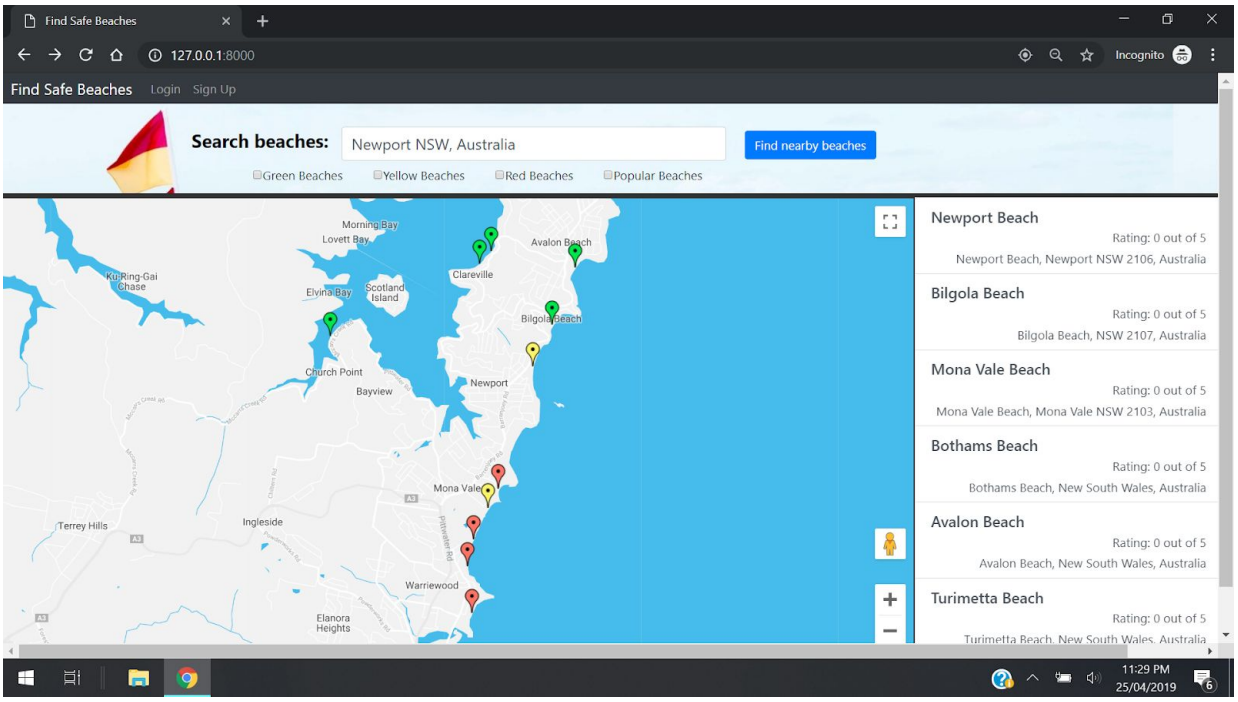
OR I have pressed a button to enable location access
WHEN I click on an item in the list of search results
AND I am redirected to a page with details specific to that beach
THEN I can view weather details about that beach
Priority: 4

Interface

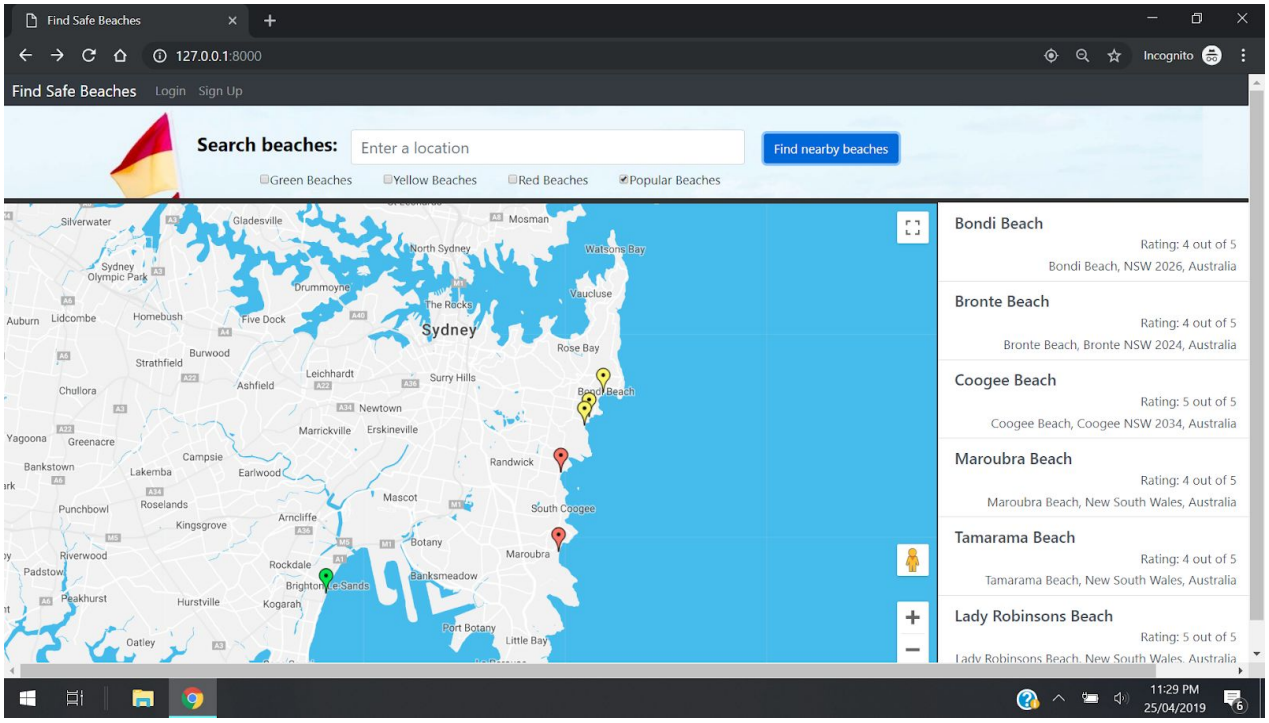
Home Page - Finding Nearby Beaches US1.1



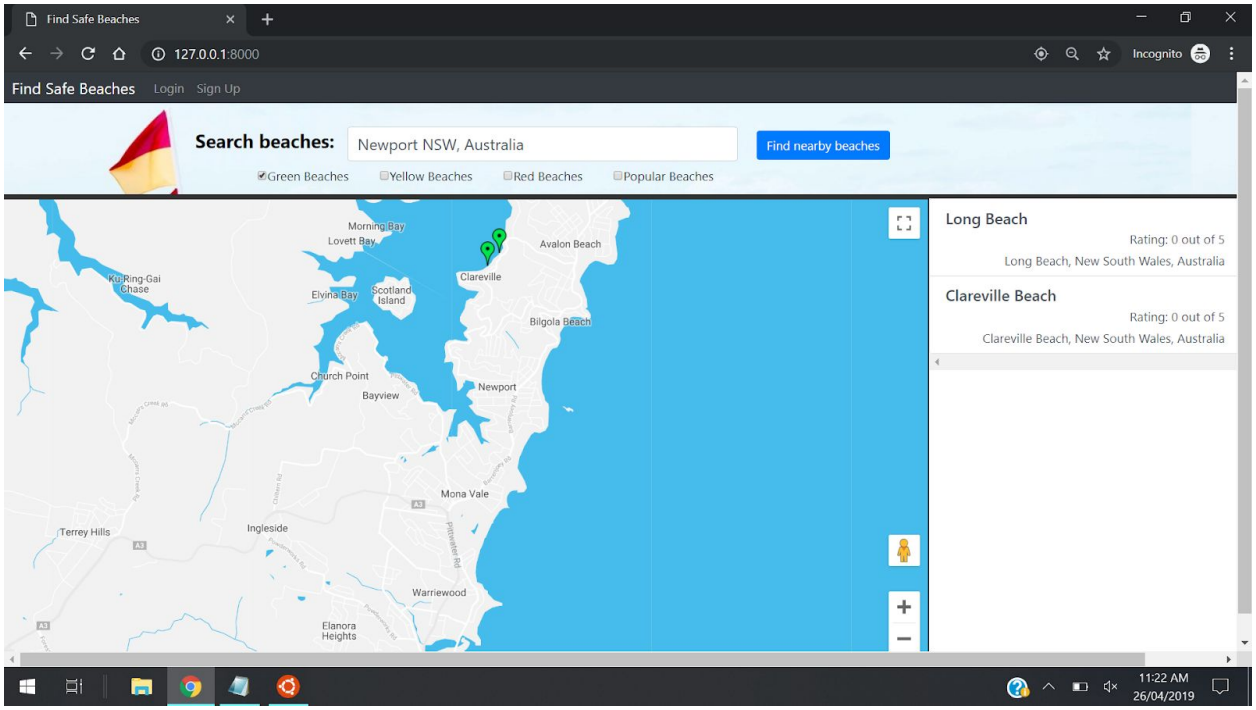
Home Page - Searching a specific location US4.2



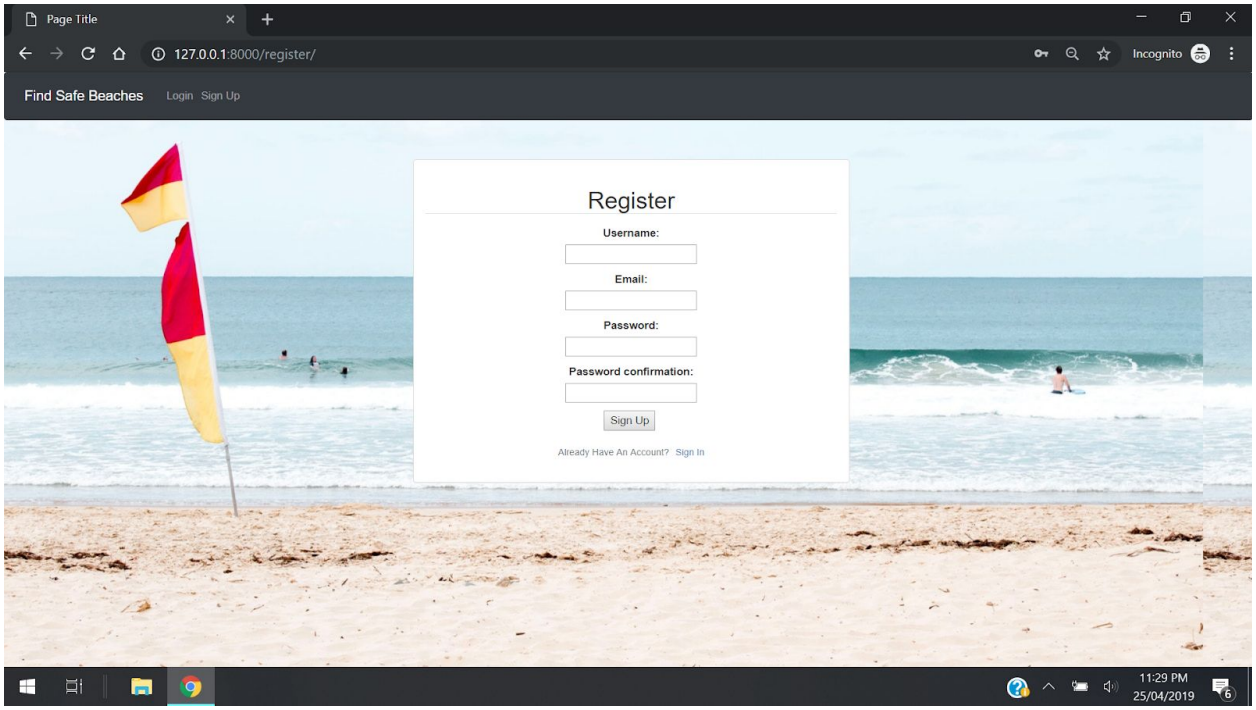
Home Page - Filtering by popularity US4.1



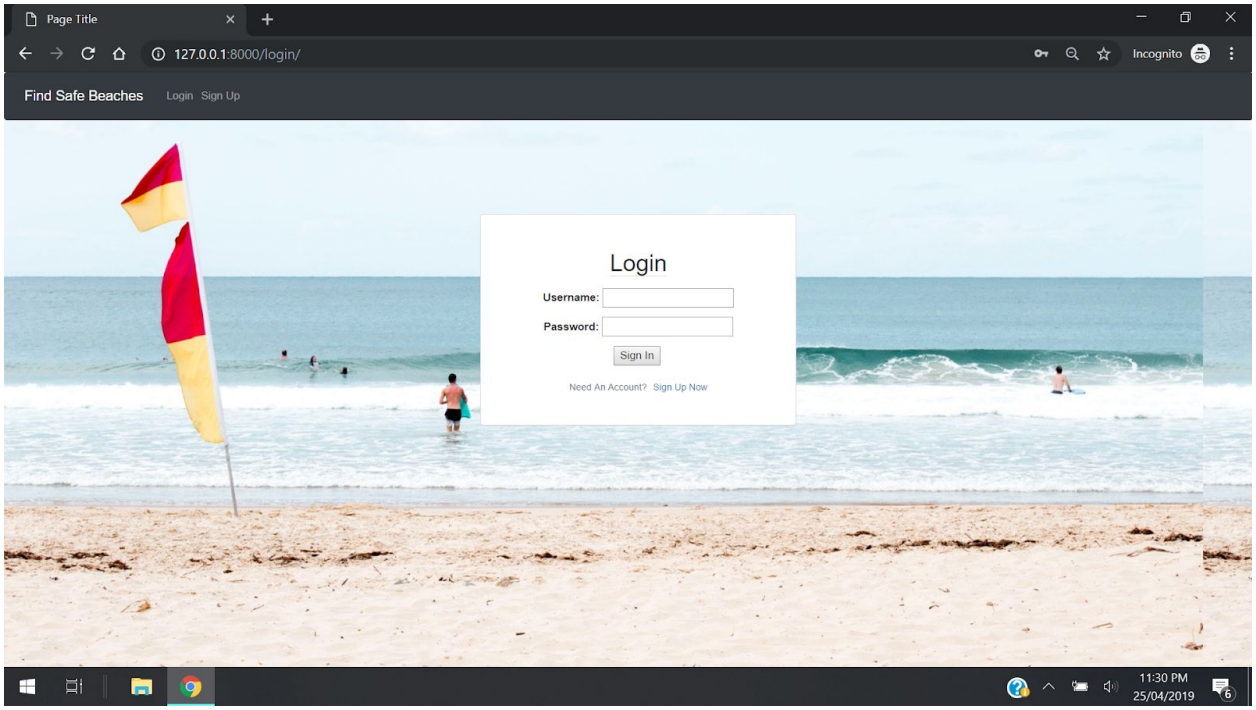
Home Page - Filtering by safety US4.3



Sign In Page US5.4

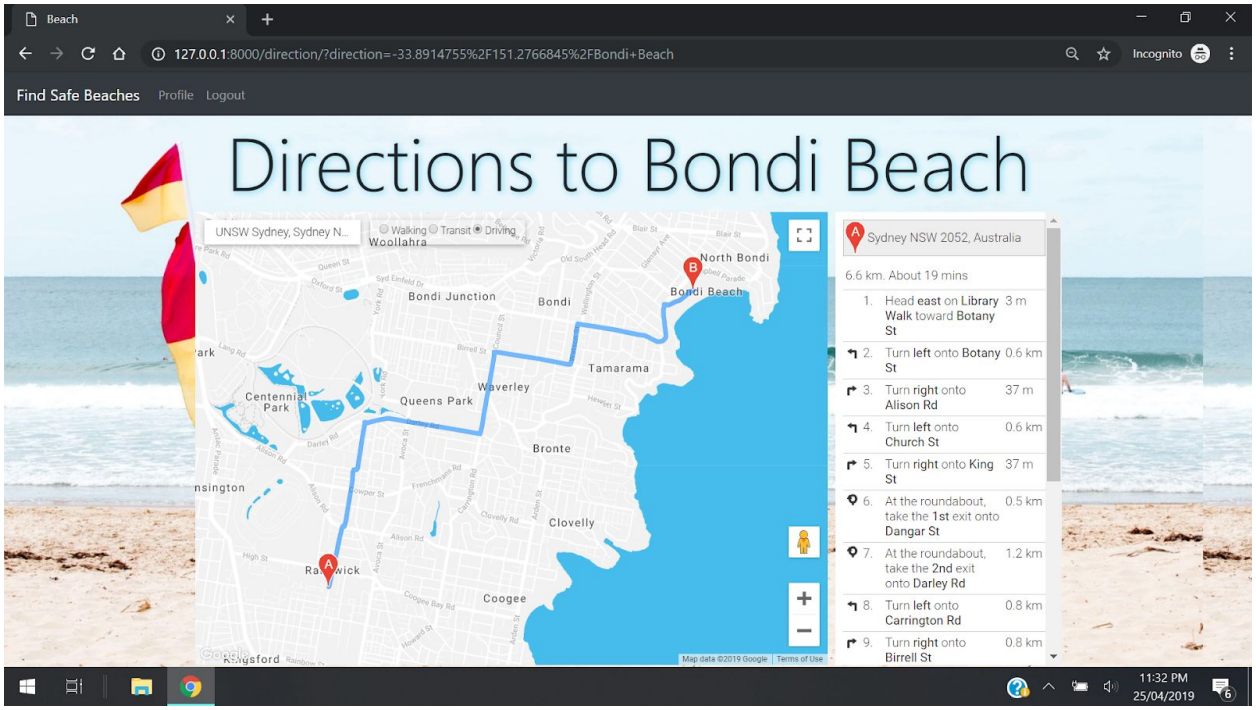


Login Page US 5.3

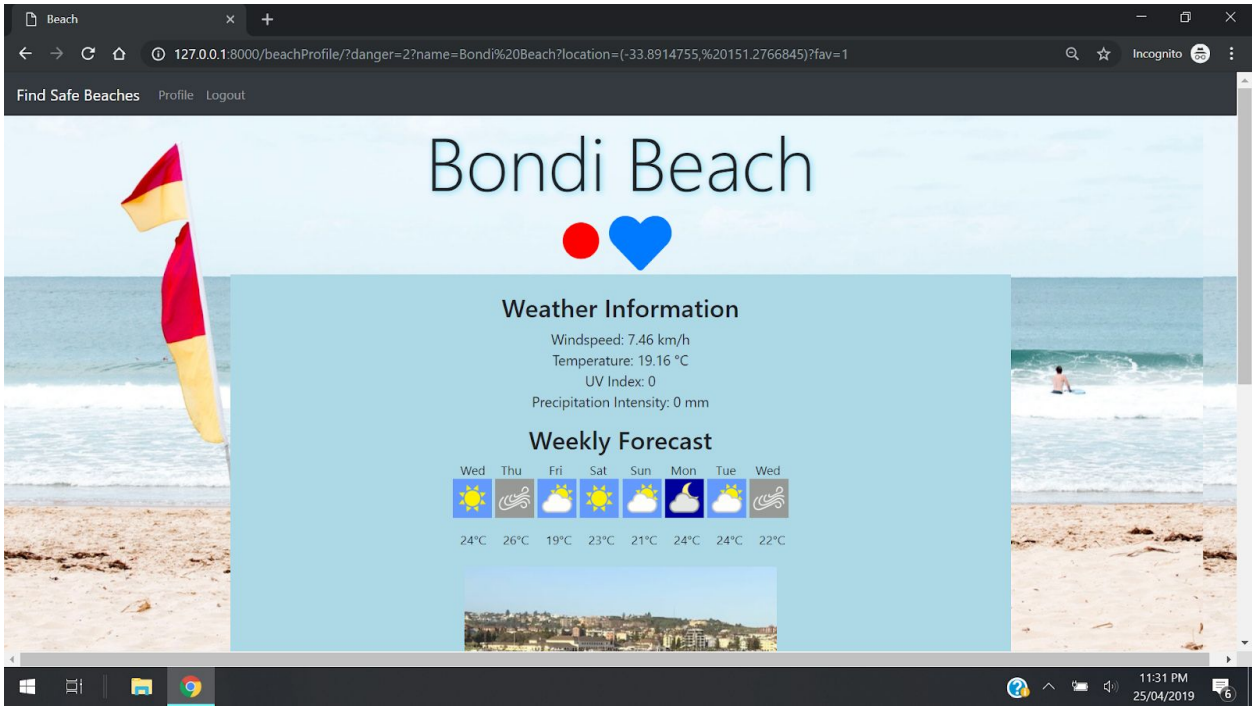


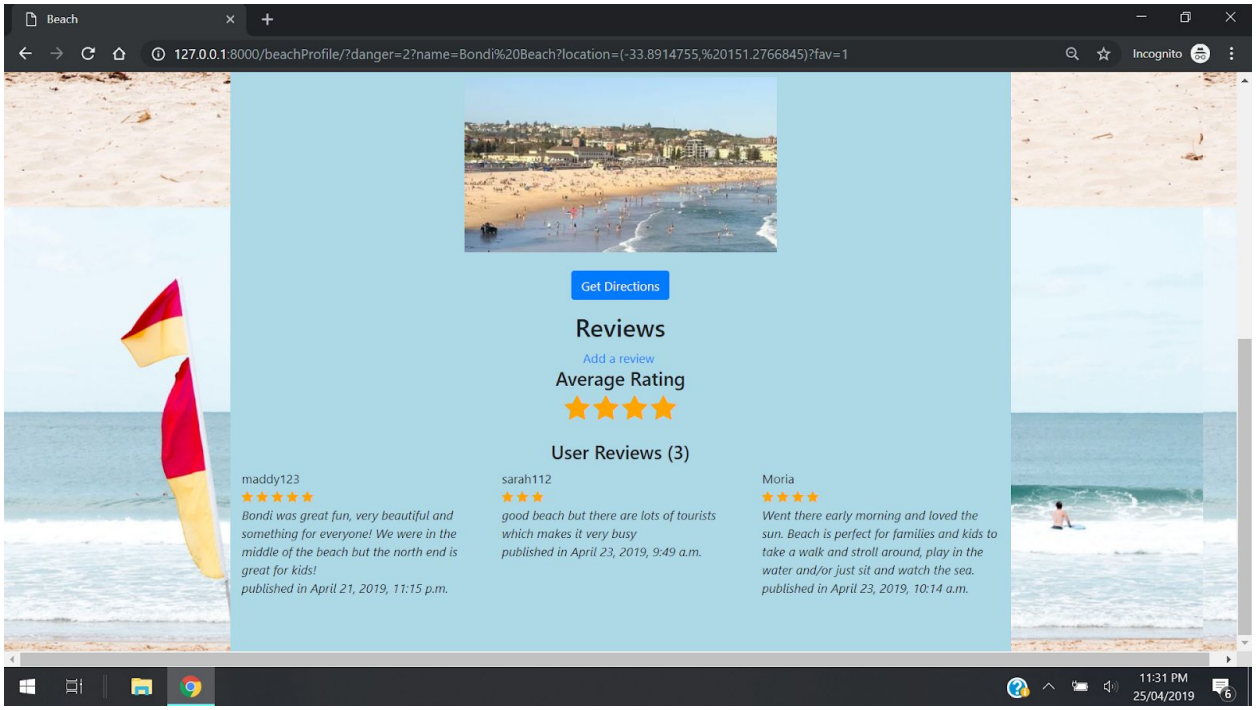
User Profile Page US2.2,5.2

Directions page US3.1,3.2

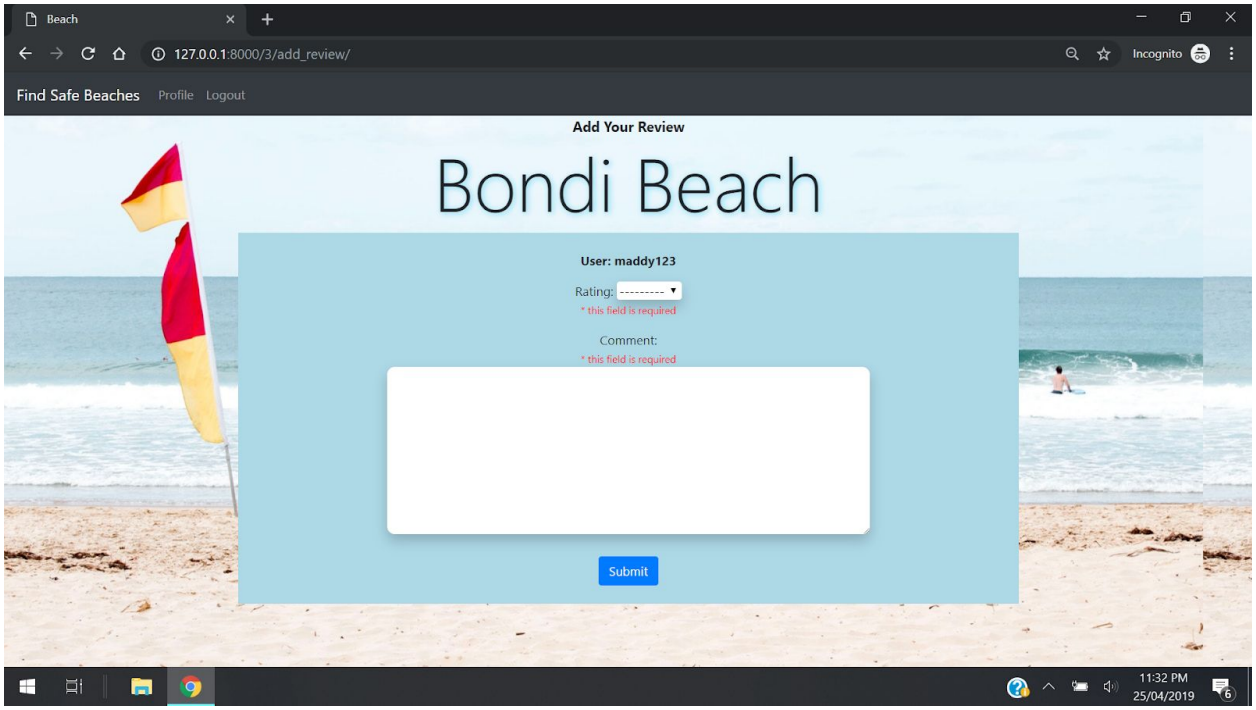


Beach Profile Page US5.1,5.6,6.1

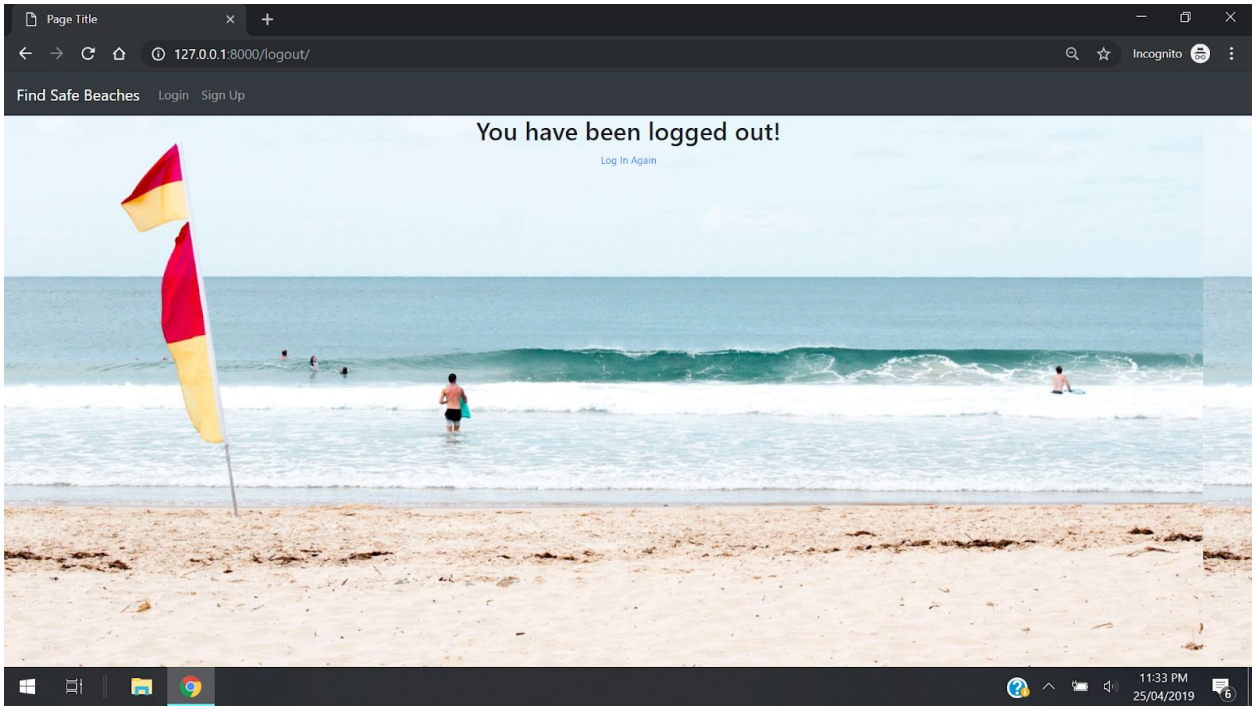




Add Review Page US2.1

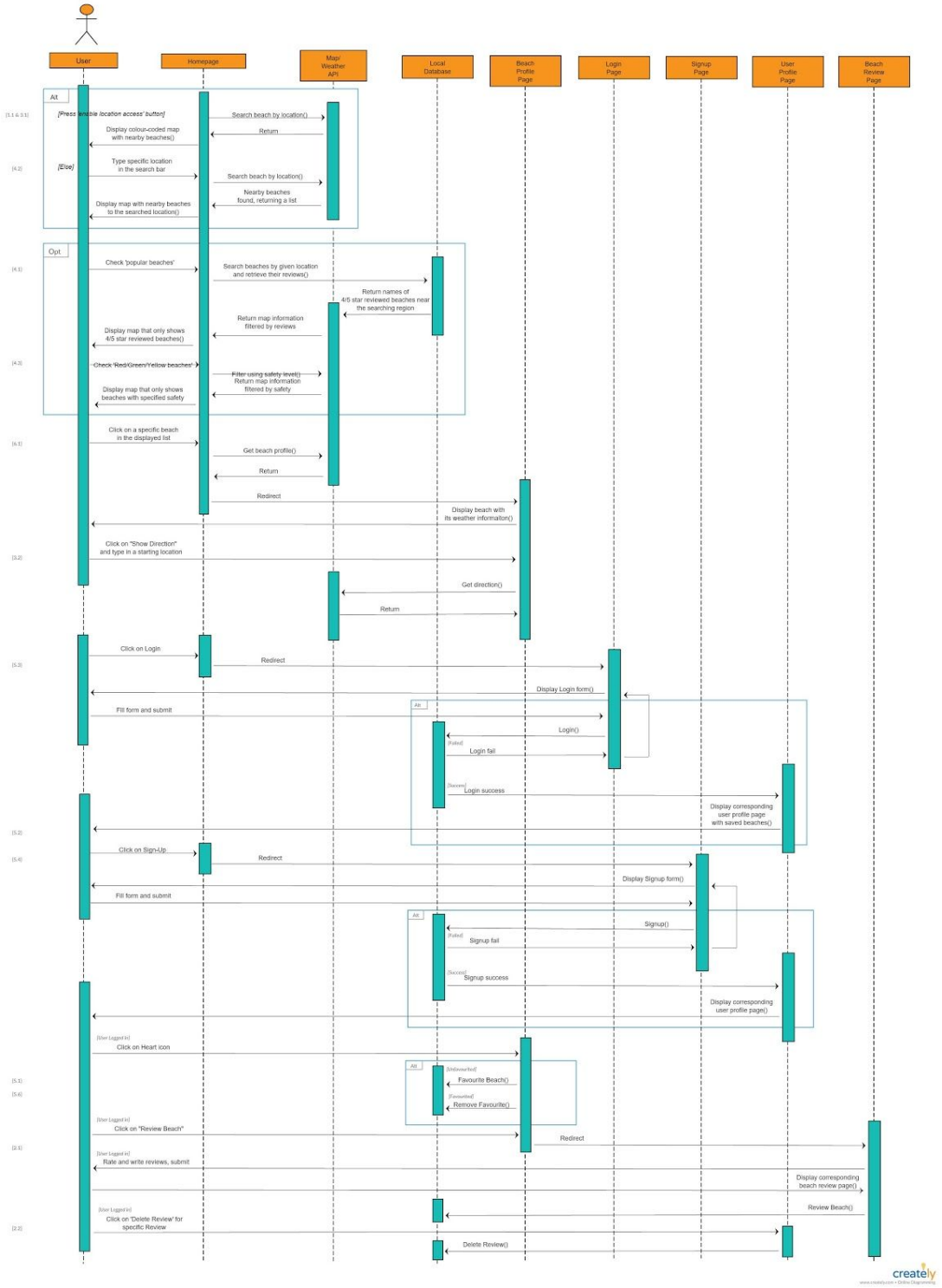


Logout Page



Sequence diagram

(A zoomed in version can be viewed in the Deliverable 5 folder on our github)



Reflection

Angela

Responsibilities/organization of the team?

User stories were divided amongst team members so that each team member could complete the feature they responsible for. Similarly, written aspects of the deliverable were also divided into tasks. We met once or twice a week to update each other of our progress, as well as complete the deliverables. We also communicated using a group chat on days we weren't meeting. Overall I felt we were well organised and communicated well with each other, and so we didn't fall behind in terms of completing our deliverables on time.

How did the project go in your opinion?

The project was difficult at the beginning because we didn't have prior experience in Javascript, and didn't know what APIs could achieve what we needed. For example, we weren't fixed on which weather API to use until week 5, since some of the weather APIs we tried before weren't accurate enough. Once we had a clear idea of how we were going to code the project we were able to complete features much faster. I think that after deliverable 2 our velocity increased. In terms of working as a team I think we worked smoothly together from the beginning.

Any issues/problems encountered?

I think the main issue was our lack of experience beforehand in Javascript, as well lack of experience coding with loose requirements. For example in the subject COMP1531 we were given all the requirements, and we didn't have practice generating ideas, so deciding what our idea was challenging at first. Our lack of experience in Bootstrap as well as using CSS led to our UI not being styled until the last minute, since functionality was the priority. The limited time frame we needed to complete our project also added a challenge since we were testing features as we added coded more features, instead of having a separate testing phase. This meant that some bugs weren't caught and fixed until we were nearing the deadlines.

Would you do it any differently now?

Ideally we would retrieve further details about beach safety such as availability of coast guards and weather hazards, but this would require knowledge of data science such as how to use pandas (python library for data analysis), since there are no APIs which provide this information. Instead, this data would need to be scanned in from the Australian Bureau of Meteorology's website, which would have been too difficult for us with little experience and limited time. Also, our UI could definitely be improved so that certain features are more clear to the user, for example it was mentioned in Deliverable 4 that the "favourite" icon was confusing. Implementing features such as text when hovering over icons would make user experience less confusing.

Madeleine

Responsibilities/organization of the team

I feel we successfully split the workload during the project. Meetings were held once or twice a week to discuss progress made on assigned tasks and to receive help from other group members if deadlines were not going to be met. For this reason, we were able to complete all the user stories we set out to and stayed on track, rarely falling behind deadlines we set for each other. Our framework, Django, also allowed great teamwork as the main application can be split into sub-applications, allowing all the group members to be working on different pages simultaneously.

How did the project go in your opinion?

In my opinion, the project was a success. We were able to complete all our user stories and we created a web app that is functional and has an aesthetically pleasing and reasonably intuitive user interface. We also worked well as a team and fell into a natural hierarchy which made the group cohesive and there were virtually no conflicts. We also managed to meet all of the deliverables with little trouble despite the time constraints feeling large and so far our marks have reflected this. Everyone in our group contributed to the project and I am pleased with the overall experience of this course. While our group was cohesive, I feel like individually we all struggled with the learning curve as we were all new to dealing with apis, javascript and their interactions. For this reason, coding was slow at the beginning, but by about week 7, we fell into a rhythm and we were significantly more productive.

Any issues/problems encountered?

I feel that most of the problems we encountered during the project came down to the learning curve from learning how apis functioned and implementing all the languages as one cohesive unit. As we were learning as we went, we often didn't use the best coding practices the first time around and so our project required a lot of patch coding and revisiting certain already completed user stories in order to make them functional for edge cases and in harmony with other user stories. I also think that our user interface could be both more stylish and more useable. No one in the group has had much experience with creating a good UI and our final project, while overall was good, definitely showed this.

Another issue we faced was with our weather api. It took us a long time to decide on a weather api as we had problems with the first two we picked. The first one we chose wasn't entirely free which was an issue we ran into a few times during this course which severely limited our options. The next api we chose wasn't accurate and gave the same values for all the beaches in our area. This caused a significant time delay as we had to learn how to interact with both of these apis only to have to start from scratch twice.

Would you do it any differently now?

I feel if we completed this course again with all the knowledge of apis we have now, we would be able to implement one of two more to bring more value to our site. I would add an additional

weather api to get more weather data to add to the integrity of our safety algorithm. I would also take more time to search for another api that gives information regarding other kinds of information for beach goers ie. if the beach is patrolled, last shark sighting. This would add another feature to our site to help it stand out from the crowd and also up user engagement.

Jessica

Responsibilities/organization of the team?

Meetings were held at least twice a week to discuss the tasks that were completed, tasks that needed to be completed, as well as any factors that might hinder our ability to complete tasks assigned by the given deadline. Several Google documents were created to work collaboratively on the deliverables, such as the report and speeches. By working in subgroups on different tasks resulted in a very effective use of time. This allowed for the team to have the time for internal testing and modification of the design, so that a tested and refined product could be presented at the final demonstration.

How did the project go in your opinion?

From the outset of the project the group had a clear idea of the end goal of the project, which meant that people stayed on task and were unlikely to go off tangent. Our team was able to share ideas and opinions, as well as help one another with completing different tasks. Our group was a cohesive team that worked effectively for the entirety of the project.

The group maintained a meticulously developed schedule to ensure tasks were completed on time, while also leaving empty space in the plan to mitigate any unforeseen problems that might occur. Accounting for unforeseen problems was very beneficial for keeping the group on track with the tasks as there were multiple times where tasks took longer than initially intended. Our team felt that although time constraints were a little heavy for the project, we ultimately were satisfied with the prototype developed.

Overall, this project had led to the increase in development of team-building and communication skills, both of which have a high importance within the working industry. The team feels that this experience bestowed many valuable lessons, not only gaining experience on working collaboratively within a group environment but also gaining exposure to the many languages needed to develop the prototype for this project.

Any issues/problems encountered?

The issues encountered were not necessarily related to team cohesion or arguments among team members but more so related to time constraints on accomplishing certain tasks by a given deadline. The constant workload of different course assignments and the SENG2021 project, coupled with the fact that the team needed to learn several languages presented some issues to the team. The major issue within the project was the limited time frame to implement features of the project to the best of our abilities. Most of the team didn't have an prior knowledge to some of the languages/ frameworks used within our project. This resulted in a lot of time spent learning the basics.

Would you do it any differently now?
If we were to redo this project, there were some things that we could have changed in order to make the development of the project alot more smoother and allow for more features to be implemented in the time possible.
Our group could have tried more API integration throughout the project. We could have used more image APIs throughout the project, such as within the beach profile (slideshow of a certain beach). We could have integrated the Google login API. This would have allowed users to login with their own existing Google accounts, instead of having to create a new login for our new website. Given another chance, we would have definitely reassessed the time frame given and made sure that we could implement features that we wanted to the best of our abilities and to the highest quality.

Fangyi

Responsibilities/organization of the team?
During the discussion of every week, we will try to resolve some problems together for the section we have been assigned before. Also at the end of our discussion, we will distribute the work for the next deliverable and try to finish them separately. In order to achieve high efficiency for the team, we will learn our section independently by ourselves by watching various tutorials online. But as a group, when we are during the meeting, other members will help to solve some specific difficulties together as well. For my own responsibility during the whole project, I was responsible for the review features and favourite features. But I got some issue with favourite feature so other group members helped to finish it. And for the review part, I was also working with Jessica for the form design of the review page. Angela helped with dealing some errors in order to let it work properly. At the stage for designing, I worked with members to finish the user stories as well.

How did the project go in your opinion?
Honestly, I think this project went pretty well and was beyond my expectation when we finished all the functionalities. At the start of the design, I thought it must be difficult to execute the section related to map. But all team members worked together and resolved all the problems that appeared. We were learning from the tutorials and searching for required documents online which helped us to start with coding. We've learned how to deal with errors by ourselves and also large amount of knowledge about the project. That is a really important skill we learned during our work. I think self-learning and teamwork can be the reason that lead us to finish our project with high quality.

Any issues/problems encountered?
Solving problems can be a good way for learning. When I started with the favorite section, I got problems with how to link users and beaches together. How can I distinguish favourite or not favorite in the beach profile? So I searched for any idea in google but still stuck here. Then, Angela helped me to solve these issues. Thus I learned how to deal with favourite which gave

me some ideas with the review after. Although these two have some different features but there are also some similar required knowledge. Some other problems I encountered are some small errors. I referred some issues posted on overflow stack and resolved them.

Would you do it any differently now?
If we are able to redo the project, I thought some features can be improved. For the user reviewed page, we can modify it with better format. The safety consideration for beaches can use more apis to get the result. Also we can add some introduction in beach profile in order to help users to know about the beach. Some more detailed information should be put onto the beach profile page. On the map, the info window can be developed with pictures and we may be able to click the picture for the beach to direct to the profile page. If we got more time, maybe we can also allow users to click the beach names in the tables on the right side of the map to go the profile page. In my own opinion I thought we can allow users to see other users' all reviews in a page. For example if we click the user name on a review, it can direct us to the page with all reviews the specific user written down before. That can help users to know one user's ideas with different beaches.

Hui

Responsibilities/organization of the team?
This project for the team is well-organized, two regular meetings were held during the week, one with the mentor and one without. We were able to discuss with the mentor about issues that we encountered during the project, as well as reporting the progress in order to know that we are on the track. For the other meeting, every team member shares their own opinion for the project and build the part of the system which they are responsible for. Outside of meeting time, we held group chat in order to discuss anything that is relevant to the project, therefore we can keep in touch with each other and check everyone's progress.
My responsibility for this project was the displaying of beach profile, as well as their directions. Both frontend and backend were implemented; in the other hand, I also made several diagrams (sequence diagram and software architecture diagram) for the system design.

How did the project go in your opinion?
In my opinion, I think the project went well. Our team was able to share ideas and cooperate these ideas in order to build a better system. Everyone was engaged during the implementation, even when we encountered conflicts about our thoughts on the project, we were able to discuss and find the optimal solution for the issue. The workload for each team member was divided evenly, everyone was able to participate equally; when someone finishes their parts earlier, they were able to help out others, too.
Furthermore, in order to complete the project in the given amount of time, regular checks for the project progress were made. Even though the time constraint was a little bit stressful for this project, our team was still able to complete each task on time and the quality was satisfying.

Any issues/problems encountered?
The main issue encountering was the time management and lack of programming experience for a web application. Since there is only nine weeks for project design and implementation, as well as the team is newly-formed (we did not know each other prior to the project), the time constraint may seem to be heavy. In addition, since not everyone is familiar to javascript, API requests, django and other technical components that we utilized for this project, we have to learn how to use them during the implementation, which costs a large amount of time. In addition, since the project time is limited, some ideas that raised during the project design was erased due to the limitation of programming ability.
Other minor issues for the project can be lack of team-working experience, at the start of the project, conflicts for ideas were made, however we were able to solve those conflicts through discussion. When it leads to the end of the project implementation, they are much easier to be resolved since we are more familiar to teamwork compare to the beginning.

Would you do it any differently now?
In order to implement a more user-friendly web application, several features can be added to the project. One thing that can be improved is to make the website more 'social friendly'. For example, users can be friends with each other and send invitations when they want to go to the beach with their friends. In addition, since the target user for our website is beach goers, the website can implement features that displays surrounding entertainments to help users planning their vacations. For example, we can include the best restaurants, available hotels around the searched beaches, thus users can utilize this website without accessing any additional websites for their beach visiting holiday.

Project Code

To access our code, please view the README instructions on our github repository:
https://github.com/unsw-se2021/idk