Part 1- Software Architecture

Programming languages: Python, JavaScript, HTML5, CSS3
Frameworks: Django, Bootstrap
APIs: Google Maps, Dark Sky
Deployment: Heroku

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.

Deployment: We are using Heroku to deploy our application so that we don't have to deal with the complexities of building our own infrastructure services. Further, Heroku allows us to easily deploy updated versions of code, and supports the programming languages we've chosen (JavaScript and Python).

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: We are using the Google Maps API to generate maps of beaches, and allow the user to search for beaches. While there are other alternatives such as Mapbox we felt that Google Maps API had more additional features such as allowing the addition of a search bar onto our map, as well as its ability to return a list of places objects that are beaches in order to gain additional information on the beaches. This additional information Google Maps provides such as the complete address, name, and location of beaches allows us to generate profile 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.

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.

Part 2a – Initial Software Design – 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

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

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

Part 2b – Initial Software Design – Sequence diagram

Diagram is on next page – please zoom to 200% to view, corresponding user stories are labelled on the left side.