CSC 431

Sandsational: Your Go-To Beach Guide

Software Requirements Specification (SRS)

Team 2

Nicey Raiyani	Project Manager
Simay Pala	System Architect
Elizabeth Varela	Requirements Engineer

Version History

Version	Date	Author(s)	Change Comments
1.0	2/24/23	Nicey Raiyani Simay Pala Elizabeth Varela	First draft
2.0	5/2/23	Nicey Raiyani Simay Pala Elizabeth Varela	Second draft
3.0	5/5/23	Nicey Raiyani Simay Pala Elizabeth Varela	Third draft

Table of Contents

Table of Tables

Table of Figures

1. System Requirements

1.1 Functional Requirements

1.1.1 Account Sign Up

Title	Account Sign Up
Description	The system allows users to create an account using either their phone number or email address. This functionality allows users to access additional features and personalize their experience besides just searching for beaches.
Priority	2
Precondition(s)	The user has a valid phone number or email address that they wish to use for their account.
Basic Flow	1. User selects “Sign Up” from the initial options. 2. A form appears with fields “name”, “email/phone number” and “password”. 3. User enters the needed information. 4. The account is created and stored.
Postconditions(s)	User is directed to the login page.
Use Case Diagram	3.1.1

1.1.2 Continue as Guest

Title	Continue as Guest
Description	Allows the user to use the application without signing up, but with limited features. The user can only use the search engine, but cannot save favorites, view their activity, or leave reviews.
Priority	1
Precondition(s)	User must open the application/website.
Basic Flow	1. User selects “continue as guest” from the initial options.

Postconditions(s)	User is directed to the search bar.
Use Case Diagram	3.1.1

1.1.3 Account Log In

Title	Account Log In
Description	Feature that allows the user to sign in to their preexisting account, which allows them to gain access to all the features available. These features include the search engine, saving beaches as favorites, and writing/viewing reviews.
Priority	2
Precondition(s)	User must have already created an account
Basic Flow	1. User selects “Log In” from the initial options. 2. User enters email/phone number and password.
Postconditions(s)	User is directed to the home page.
Use Case Diagram	3.1.1

1.1.4 Create User Profile

Title	Create User Profile
Description	Allows the user to add personal information to their profile such as their default location, age, and favorite beaches. This information will be used for providing more personalized results.
Priority	3
Precondition(s)	User must have already created an account.
Basic Flow	1. User will be asked the following information: a. Location b. Age c. Favorite beaches (optional)

Postconditions(s)	User is able to get beach recommendations and search for beaches.
Use Case Diagram	3.1.1

1.1.5 Display Home Page

Title	Display Home Page
Description	Limited access feature which allows the user to access the search button, their favorite beaches, and beach reviews they've written all from one place once they're signed in. Only signed in users can get to this page.
Priority	0
Precondition(s)	Users must open the app/website to search and must be logged in.
Basic Flow	1. User opens the app/website. 2. User logs in if they want to get access to the full Sandsational features. 3. User begins to search for beaches, view previously saved beaches, or review account activity.
Postconditions(s)	User is able to search beaches, view their favorite beaches, or view their account activity.
Use Case Diagram	3.1.1

1.1.6 Search

Title	Search
Description	Users can search beaches on this page based on their preferences. Users can select when, where, and what they want the beaches that return from the search to feature. The users can also select filters and how to sort the results to help narrow their search.
Priority	0
Precondition(s)	User must open the app and be signed in, or user must open the website and be either signed in or have continued as a guest.

Basic Flow	1. User opens the app/website. 2. Users click the search bar. 3. Users type the location of the beach. 4. Users type what activity/thing they want the beach to offer (ie. surfing, restaurants, etc.). 5. Users have the option to select other filters to narrow the search with and/or how to sort the results. 6. They can pick the beach they desire from the results.
Postconditions(s)	User is able to view the details of a beach they found from the search results.
Use Case Diagram	3.1.1

1.1.7 Filter the Search

Title	Filter the Search
Description	Users can filter their search according to their needs, they will be in the form of checkboxes so users can select one or multiple attributes through which they want to filter the beaches.
Priority	0
Precondition(s)	User must have access to the home page
Basic Flow	1. User opens the home page and starts searching 2. User can filter the beaches according to: a. Weather b. Activities available: surfing, sunbathing, kiteboarding c. Available parking d. Pet friendly e. Etc.
Postconditions(s)	User will receive and view search results accurate to the filters chosen.
Use Case Diagram	3.1.1

1.1.8 Sort the Search

Title	Sort the Search

Description	Users can sort their search according to their priorities. This will make it easier to make a choice as the sort feature can organize the results based on distance, ratings, or relevance.
Priority	0
Precondition(s)	User must have access to the home page
Basic Flow	1. User opens the home page and starts searching. 2. The beaches can be sorted according to: a. Location (nearest beaches) b. Highest Rated c. Relevance
Postconditions(s)	User views search results organized by the sorting method of their choice.
Use Case Diagram	3.1.1

1.1.9 View Selected Beach

Title	View Selected Beach
Description	Users can select the beach of their choice to see detailed information regarding this beach.
Priority	0
Precondition(s)	Users must have searched for beaches of their preference and have gotten this beach as a search result or have saved it as a favorite beach.
Basic Flow	1. User opens the home page and starts searching. 2. User can sort and filter the beaches as they choose. 3. User receives search results back as a list. 4. User selects the beach of their choice to view details and more information regarding the beach.
Postconditions(s)	User is able to view the details of a beach they search for.
Use Case Diagram	3.1.1

1.1.10 View Favorite Beaches

Title	View Favorite Beaches

Description	Users have easy access to the beaches that they previously added to their favorites. This feature is only available to users that are registered and signed in.
Priority	2
Precondition(s)	User must have added some beaches to their favorites list.
Basic Flow	1. User accesses the home page. 2. User selects the “favorites” option.
Postconditions(s)	User can browse through their favorite beaches list.
Use Case Diagram	3.1.1

1.1.11 Add Reviews

Title	Add Reviews
Description	Users can add ratings and reviews about the beaches they visited. This feature is only available to users that are registered and signed in.
Priority	2
Precondition(s)	User must be logged in.
Basic Flow	1. User selects the beach of their choice. 2. User selects “write review” to leave a rating and a testimonial for the selected beach.
Postconditions(s)	The review is uploaded to the specific beach.
Use Case Diagram	3.1.1

1.1.12 View Reviews

Title	View Reviews
Description	Users can view the reviews and ratings of a beach they select through their favorites or through the search results.
Priority	2

Precondition(s)	User must have selected a beach to view its details.
Basic Flow	1. User selects a beach from the search results or from the list of their favorite beaches. 2. User can view the reviews and ratings of the beach they have selected.
Postconditions(s)	Registered user can write reviews for this selected beach.
Use Case Diagram	3.1.1

1.1.13 Log Out

Title	Log Out
Description	The user can log out from their account if they're logged in. This will also save all the information of the user before they are logged out to make sure that the next time they log in their favorite beaches and activity is updated.
Priority	2
Precondition(s)	User must have an account and be logged in.
Basic Flow	1. User is registered and logged in. 2. User picks the "logout" option from settings
Postconditions(s)	User is logged out from their account.
Use Case Diagram	3.1.1

1.2 Non-Functional Requirements

1.2.1 User Credential Security

Title	User Credential Security
Description	All users will need to use a password for their account to make sure only they have access to it. The password will have basic requirements to make sure it is secure. User passwords will be encrypted. This is to protect from potential security attacks and protect users' personal information.
Priority	1
Applicable FR(s)	1.1.1, 1.1.3, 1.1.4, 1.1.8

1.2.2 Filtering Accuracy Based on Preferences

Title	Filtering Accuracy Based on Preferences
Description	Each beach has several features hence when the user chooses to apply filters to their search, the algorithm must maintain the accuracy of the filters by only displaying beaches that match all the attributes selected.
Priority	0
Applicable FR(s)	1.1.7

1.2.3 Ranking Based on User Data Collected

Title	Ranking based on user data collected
Description	If no sort attribute is selected, the default ranking system of the order of beaches displayed would be the "highest rated" ones because we want to make sure that we recommended the best beaches to our users.
Priority	3
Applicable FR(s)	N/A

1.2.4 Search Results Performance Speed

Title	Search Results Performance Speed
Description	Optimize search results performance speed to ensure that it meets user expectations and remains competitive with other search engines
Priority	0
Applicable FR(s)	1.1.6, 1.1.7

1.2.5 App Availability and Downtime

Title	App Availability
Description	The app should be available to users at all times, with only 20 minutes of downtime allowed per month.
Priority	3
Applicable FR(s)	N/A

2. System Constraints

2.1 Tool Constraints

2.1.1 Data Availability

Title	OpenWeather API and Leaflet Library
Description	Access to reliable and up-to-date data about weather conditions, water temperatures, beach facilities and more. Obtaining the data and using it may require purchasing licenses from the data providers.
Priority	2

2.1.2 Web Application Framework : Backend

Title	Ruby
Description	The back-end framework would be done in Ruby.
Priority	0

2.1.3 Mobile Application Framework : Backend

Title	Flutter
Description	The back-end framework would be done in Flutter.
Priority	0

2.1.4 Azure

Title	Azure Cloud Storage

Description	To help maintain the scalability and security of the data. Will help reduce the expense of physical data storage. Utilize the LAMP stack instance of Azure.
Priority	3

2.2 Language Constraints

2.2.1 Angular : Frontend

Title	Angular 14
Description	The open-source framework can be used to build a robust scalable application.
Priority	0

2.2.2 Database Interactions

Title	SQL
Description	To handle the data provided initially and the constant data gathered from users.
Priority	0

2.3 Platform Constraints

2.3.1 Mobile Application Constraints

Title	IOS or Android
Description	Flutter will help build an app for both these platforms.
Priority	5

2.4 Hardware Constraints

2.4.1 Mobile Device Compatibility

Title	iPhone, Samsung, Pixel
Description	The map functions might not be supported by some phone hardware, so the regulations must be met.
Priority	5

2.5 Network Constraints

2.5.1 Internet access

Title	Internet Access

Description	Users would need internet access as the app requires the data loaded from the search engine to be passed through with an internet connection.
Priority	0

2.5.2 Access Database

Title	Access User database
Description	Clients will need to access the user database for information.
Priority	2

2.6 Deployment Constraints

2.6.1 Azure App Service

Title	Azure App Service
Description	The web application will be deployed with Azure App service which will be in coherence to the Azure cloud storage. This will also be used for making the mobile application.
Priority	0

2.7 Transition & Support Constraints

2.7.1 End of Life

Title	End of life for the project
Description	Once the class ends, the developers will cease to continue working on it, unless the transition is done to another team that is not taking this class.
Priority	5

2.8 Budget & Schedule Constraints

2.8.1 Budget Constraints

Title	Budget
Description	There is no initial funding for this project, as we are just college students. Therefore, everything utilized must be free.
Priority	5

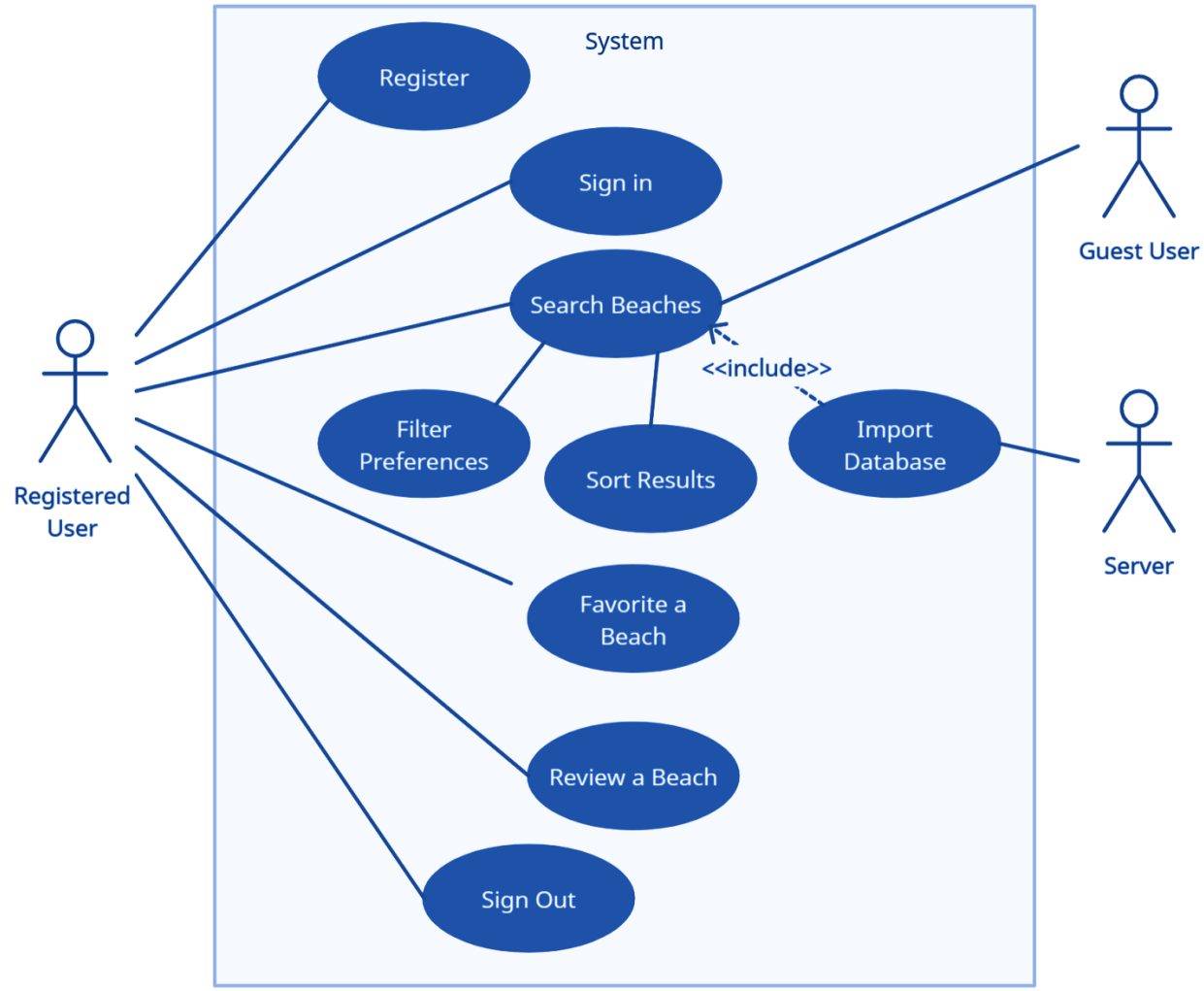
2.9 Miscellaneous Constraints

2.9.1 Scribe and Elastic Search

Title	Search Engine
Description	Scribe and Elastic Search will be the basis of our search engine these will help optimize the search and enable the integration of the several features mentioned.
Priority	0

3. Requirements Modeling

3.1 User v. System



4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 View Analytics

Title	User analytics
Description	To enhance the effectiveness and usability of our application, we plan to gather data on app usage. By collecting this data over time, we aim to identify patterns in wait times and leverage these insights to provide more optimal recommendations to our users.
Priority	3
Precondition(s)	Users must use our app for an extended period, while we establish databases and systems for analyzing the data, to obtain actionable insights that improve our app's functionality and user experience. And continue to give reviews about their experience.
Postconditions(s)	The data will be used to make more effective recommendations on the beaches

4.2 Non-Functional Requirements

4.2.1 Maintain accurate data

Title	Location Data

Description	The location or temperature data needs to be accurate for the users
Priority	0
Applicable FR(s)	View Analytics