Atlas Itinerary Project Design

By Gabriel Taylor, Dustin Zook, Sorelle Djuissi, and Gabriyel Sorensen

# Project Parts

## Design Description

### The Front End

When the user accesses the page, they will be prompted to sign up using their Google account. Beneath the option to sign up using their Google account, returning users will be shown how they can sign in through a message with an embedded link.

Once users have finished signing up or signing in, they are greeted with the website’s Home Page. The Home Page’s main focus is the “About section”. This section will explain to users the ins and outs of the website. The background of the group behind it will be included in the About Section, as well as details on how to navigate the site. Details like the options in the navigation bar, what each of those options does, and how to navigate those sections. Each option in the navigation bar will be displayed as a label at the top of the page; each label will have an embedded link that will take users to the respective page for the option. The Navigation Bar has two main purposes: to allow users to either start creating an itinerary or to allow users to view old itineraries.

If the user chooses to select Create Itineraries, they will be presented with a page where they can create itineraries. There will be a text box for their Travel Destination. The text box for their destination will be blank, allowing users to input their destination’s name. The text box will also serve as a search bar, showing predicted destinations based on what is already typed. As an example, this feature is similar to the Google search bar. If the user does not enter a destination, the application will be able to detect the user’s current location and generate suggestions for their Travel Destination.

Once the destination searched for is found, the user will be presented with a sidebar containing several categories. These categories will be presented as three buttons. When a button is pressed, a series of cards relevant to the selection will be displayed. The default category button pressed will be lodging. Lodging will allow users to select locations where they can stay for the day, nap in the evening, or sleep for the night. Most users will probably select one location for all three of these things to happen. The second category will be dining. Dining will allow users to select a location to get food at. For breakfast, should the lodging provide it, the user will be able to select the lodging as the dining for breakfast. The final category provided will be the attractions category, where users can search through the places that the area has that are frequented during visits.

The lodging category will contain subcategories that the users can choose from. The subcategories will be listed in a filter option that appears next to the Destination search bar. Once a filter option is selected, the options will appear as a series of cards that the user can select. Each card will contain a list of relevant details about the lodging. The most commonly picked lodging will no doubt be hotels. Hotels offer the basics a traveler might need, things like a shower, a bed, maybe a fridge, a closet for clothes, and maybe even a TV. The next option will be a Bed and Breakfast. These are typically small, privately-owned lodgings in a personalized setting. They are very home-like. Finally, users may choose from specialty lodgings like ecolodges, hostels, or even mountain huts if the area contains such.

Attractions: Similar to the lodging category, attractions will also be represented by an Attractions button. When selecting attractions, the filter will appear next to the search bar, unless it is already there from when on of the other categories was selected. This time it will promote new subcategories related to attractions that users may want to visit. Examples of the subcategories are attractions like: boat tours and water sports, shopping, as well as nearby nature and parks.

Dining: Dining will be the third button users can click. This option performs almost identically to the first two. The filter option allows users to pick between a café, a sit-down restaurant, or fast food. Fast food restaurants are the simplest to understand. They are quick and concise, allowing users to go in and come out in a timely manner. Sit-down restaurants require more patience and should be planned around and not just dropped in with the hope of fitting into the schedule. Cafes are a little nuanced, giving users an atmosphere to take things at their own pace with mainly coffee and other beverages meant to boost productivity during stationary work.

More Navigation: Each Lodging, Attraction, or Dining location will be displayed as a list of cards. Each of these cards will display an image of the location and the name of the location. A paging system will allow for the maximum number of cards to be displayed. The cards will be organized according to their rating. When one of the cards is clicked, more images of the location will be displayed as well as a description of the location, the rating of the location, and an add to itinerary button. After selection, add to the itinerary, the user will be asked for the date and time for the location. Four text boxes will appear, one for the start and end date as well as the start and end time.

After the location is successfully added to the itinerary, a sidebar will appear with the current itinerary being built. The itinerary will be displayed in a calendar view, with saved locations displayed as bullet points on each day. Each day may be expanded where the locations and events for the day are listed in chronological order. Users can choose to select events and move them around or even delete them.

Lastly, there will be a Save As button. The Save As, when clicked, will prompt users to name their itinerary. Naming the itinerary will make it easier for the user to identify later in the View Itineraries option. Once the name is input into the system, the itinerary is saved.

Viewing Itineraries: After the user successfully creates an itinerary, they will be able to go to the View Itineraries page. The first thing the users will notice is that the itineraries they built are not presented to them in a card form. The card will consist of the name of the itinerary, the start date, and the end date of the itinerary. If the user clicks on the itinerary card, they will be able to see more of the details they included in the itinerary. This will be possible because, after clicking on the itinerary, the card will expand to a fuller view.

The view of the card will be a calendar-like view, with bullet points for each event that will occur on each of the days that the itinerary spans. Clicking on a day has a similar behavior to clicking on an itinerary. The day expands to a fuller view with the events for the day listed in chronological order. However, unlike in Create Itinerary, the user will first see the day in Viewing Mode. This will mean that the user will not be able to make changes. Near the top right of the expanded day card will be an Edit Mode button that will switch the user into Edit Mode. After they are in Edit Mode, they will be able to make changes to the day as if they were in Create Itineraries.

In Edit Mode, users will be allowed to select events. Once the event is selected, the users will be able to access the delete button for the event, where they can remove it from their day. If the user should press the delete button for the event, they will be shown a warning message. The message will display a message to the effect of, “Are you sure you want to delete this event and two options for proceeding will be offered. The first option will be to the effect of “no, I am not sure. KEEP THE EVENT.” While the second option will be, “yes. DELETE THE EVENT.” The important sections will be in all caps to help the user understand the decision more clearly.

Edit Mode will also allow users to move events. Once an event is selected, users can move those events around. After moving the event to a different time, the official displayed time of the event will be updated. If this event is not available at the set time, a warning alert will be provided in red next to the event, letting the user know, “This event is not available at this time.”

Similarly to events, users will also be able to select days. Selecting a day will allow the user to delete the day. Attempting to delete the day will result in a similar message of, “Are you sure you want to delete this day and all of its events?” Again, a set of options, “no, I am not sure. KEEP THE DAY,” and “yes. DELETE THE DAY,” will be provided. Selecting the day will also allow users to move the day’s events to another day. Similarly to with events, if a day’s events are not all available on the new day they are moved to, the user will be notified with a red message next to every event that is not available on the new day.

The back end

React: Behind the beautiful and fun cards and bullet points, systems like React are working to make sure everything stays in one piece. React operates as a kind of merging of HTML and JavaScript. In React, each piece on a page is a component. Starting on the Sign Up page, the Google sign-up option will be a component that links to the Google api. This will reduce the amount of complicated measures the developers will have to take to ensure user security, as Google is a larger multi-million dollar company that has been at this for a long time. In JavaScript, developers are required to have a first letter, capital letter naming convention. JSX code requires the same kind of naming convention, with a capital letter for the first letter of components. In the case of the Sign Up page, the name for the overarching component might be “function SignUpPage{}”.

How components work: When React is running, the functions will have other, more specific functions embedded in the main function component. For instance, the function for the component on the sign-up page responsible for the message “Already have an account? Sign in here” will have a function in it that accesses a URL link to the sign-in page while also being a function inside of the function SignUpPage{}. Because of this nested method of making each component, some actions can be repeated. The Sign Up and Sign In pages will both have a function component that accesses the api for Google accounts. So instead of writing this twice, making a component call will be more effective.

SupaBase: Atlas Itinerary is not just going to be flashy cards and functioning links. Users will build itineraries for themselves, which means the development team will need a way to organize user itineraries into separate spaces, specifically for each user. This will require the use of a database. The database manager of choice for this project is SupaBase.

React+SupaBase: Supabase boasts the ability to be easily connected to by Software Development Kits like React. Unlike the stereotypical database, SupaBase does not use SQL but instead uses JavaDoc api code. The project will be tedious as it is. With the requirement of learning new technology and the time required to do so. Any advantages offered like easy connections, make the project run smoother. In the case of SupaBase and React, being able to use similar functions and coding languages between the software systems will help make the process more streamlined and reduce the potential for careless mistakes.

Meshing with Google: The database also boasts integrability with user authentication, like the Google api this project uses. One of the team’s main concerns was with the security risks that come with making a stack or organizing account names from scratch, but SupaBase integrates with Google’s api so well that this concern should be trivial in the long run.

As a result of SupaBase

## Revised requirements

## Block diagram

## User Interface Storyboard

## Message documentation

## Storage documentation

## Misc documentation

## appendix