

# TripExpert

## Summary

TripExpert is an interactive trip-planning web application for SPTrans, São Paulo's public transport system. Users can plan a trip by selecting their location of departure, their destination, and the time at which they would like to depart for their trip. Our application will then provide users with multiple possible trips they can take to get to their destination. The proposed routes will be optimized for either price or time, depending on the user's preferences. Each trip shown will have details provided about its duration, length, stops, price, departure time, and arrival time. The app will also have an account system, and authenticated users can save their planned trips to a personal list that can be accessed and updated at any time. In addition, the app will visualize selected routes on a map of São Paulo.

## Description

TripExpert is a trip-planning web application for São Paulo public transport. First, users will be able to log in with a username and password to access their own personal version of the application. Then, they will be able to select an origin, destination, and departure time and submit them. The app will return some paths between the origin and destination, optimized for either fare price or travel time depending on user selection.

Users will then be able to save these trips on their account for later use. Each user will have a list of trips that they can add to, delete from, and edit based on their needs. As the creative component, these trips will be visualized on a map of São Paulo.

The application will use the Public Transportation Service dataset on Kaggle, which contains information on the São Paulo bus system, known as SPTrans. The data contains routes, schedules, fares, and everything else that is required to make a scheduling application. The data is provided in a format called GTFS (General Transit Feed Specification), which was designed by Google to implement transit into Google Maps.

## Usefulness

TripExpert is intended to enhance the public transportation experience in São Paulo for commuters, visitors or even city planners.

Regular users can plan their daily routes efficiently and be able to access previously used routes. They could use the application to plan their daily trip to work via public transport, and they could use the pathfinding features to find a route to a new place in town. Tourists would find the application useful to understand and navigate São Paulo's extensive public transportation network, if they choose to travel by bus to nearby tourist spots in the city. City planners could

also use the app as an easy way to visualize the SPTrans GTFS dataset, which would be useful for analyzing and improving the city's transit system.

Our application is similar to Google Maps, which is a navigation application developed by Google that supports navigation, real-time GPS tracking, and more. Both TripExpert and Google Maps help users plan routes from a given origin to a given destination. Our application differs in a few key ways. Firstly, our application will be optimized specifically for navigation using the SPTrans public transport system in the region of São Paulo. Secondly, our application will have a more streamlined and accessible interface. Google Maps offers pictures, 3D “VR”-like imagery, worldwide routing, GPS navigation, location reviews, and more. Our application is designed to focus specifically on finding routes for users, and as a result, our application interface is more streamlined and easy to navigate. Additionally, our application allows users to select an optimization parameter of their choice - either fare price or travel time - that our graph routing algorithms will take into account when computing routes between users' selected origin and destination.

By allowing users to quickly find routes optimized for their individual goals (i.e. minimizing either fare price or travel time), and honing in on personalized route-planning functionality, wherein users can save their planned routes, our application will solve the problem of finding efficient, personalized routes for anyone looking to travel around São Paulo using the SPTrans transit system.

## Realness

The data will come from the [Public Transportation Service dataset](#) currently hosted on Kaggle [1]. The dataset contains information on fares, routes, frequencies, stops, and stop times for SPTrans, São Paulo's public transport system. There are about 40,000 records in the various files in the dataset, including geographic coordinates that allow for map visualization alongside the transit planning.

The data will require some advanced queries to access full trip data, but since the data uses the GTFS format, which was created by Google with developers in mind, there is a lot of great documentation on how exactly to process the data [2]. The use of the standardized GTFS data means that the application could later be expanded to more cities, provided that they publish their data publicly.

[1] <https://www.kaggle.com/datasets/mateuscco/sao-paulo-transportation-service>

[2] <https://developers.google.com/transit/gtfs/reference>

## Functionality

This application will offer several key features:

- Route Planner: Allows users to input their start and destination addresses to receive a detailed transit route. (e.g. Time estimates, transfers, price etc.)

- This functionality fulfills the “database *search*” requirement. In selecting an origin and destination, users will be effectively *searching* for locations contained within our database which they can set as their origin and destination.
- My Trips: Users can save routes they frequently use or plan to use again in the future. This makes it quick and easy to select a route without having to input the start and end points each time; common locations can be auto-filled when planning a trip
  - This functionality fulfills the *insert new records*, *update records*, and *delete rows from the database* requirements. By “saving routes”, users will effectively be adding new records to our database, which will contain the information about each trip that the user has saved. Each time a user saves a route, a row(s) will be added to our database. Each time a user deletes a route, a row(s) will be deleted from our database.
- Map Visualization: An interactive map displaying estimated routes, stops.
  - This will be the *creative component* of our application, providing a visualization of the data stored in our database that helps the user see their routes through a geographic lens.
- Pathfinding: In the backend of our application, we will use graph algorithms to find routes between a given origin and destination, optimized for fares or time, using the GTFS data.
  - Our pathfinding algorithms will undoubtedly use multiple (2+ advanced queries, utilizing techniques such as *aggregation via GROUP BY* and *subqueries*, thereby fulfilling the requirements for using advanced queries.

# UI Mockup

## Log In Page

*This is the first page a user will see when they enter the website.*

**TripExpert**

### Log In

Email	<input type="text" value="aturing@gmail.com"/>
Password	<input type="password" value="secretPassword123"/>
<input type="button" value="Log in"/>	

*Don't have an account yet?*

[Sign Up](#)

## Sign Up Page

**TripExpert**

### Sign Up

*Enter Your Information Here:*

First Name	<input type="text" value="Alan"/>
Last Name	<input type="text" value="Turing"/>
Email	<input type="text" value="aturing@gmail.com"/>
Password	<input type="password" value="secretPassword123"/>
<input type="button" value="Sign Up"/>	

*Have an account?*

[Log In](#)

## Plan My Trip Page

TripExpert

[Sign up](#) [Log Out](#)

Plan A Trip

My Trips

Enter Your Trip Details Here:

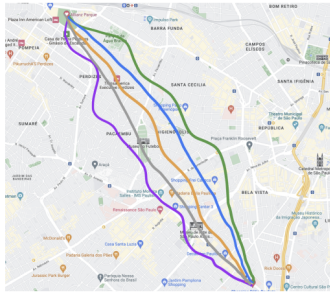
Starting Point

Destination

Departure Time (BRL)

[Directions](#)

Possible Trips on a Map



Trips:

	<u>Departs From</u>	<u>Arrives At</u>	<u>Departure Time</u>	<u>Arrival Time</u>	<u>Price</u>	<u>Add to My Trips</u>
<input type="radio"/>	{departure_point_1}	{arrival_point_1}	{departure_time_1}	{arrival_time_1}	{price_1}	<input type="checkbox"/>
<input checked="" type="radio"/>	{departure_point_2}	{arrival_point_2}	{departure_time_2}	{arrival_time_2}	{price_2}	<input checked="" type="checkbox"/>
<input type="radio"/>	{departure_point_3}	{arrival_point_3}	{departure_time_3}	{arrival_time_3}	{price_3}	<input type="checkbox"/>
<input checked="" type="radio"/>	{departure_point_4}	{arrival_point_4}	{departure_time_4}	{arrival_time_4}	{price_4}	<input checked="" type="checkbox"/>
<input type="radio"/>	{departure_point_5}	{arrival_point_5}	{departure_time_5}	{arrival_time_5}	{price_5}	<input type="checkbox"/>

## My Trips Page

TripExpert

[Sign up](#) [Log Out](#)

Plan A Trip

My Trips

My Trips:

	<u>Departs From</u>	<u>Arrives At</u>	<u>Departure Time</u>	<u>Arrival Time</u>	<u>Price</u>	<u>Remove from My Trips</u>
<input checked="" type="radio"/>	{departure_point_2}	{arrival_point_2}	{departure_time_2}	{arrival_time_2}	{price_2}	<input type="checkbox"/>
<input checked="" type="radio"/>	{departure_point_4}	{arrival_point_4}	{departure_time_4}	{arrival_time_4}	{price_4}	<input type="checkbox"/>

# Work Distribution

## Colin Kingsley

- Frontend & UI development & implementation
  - Developing an interactive frontend for the application that allows users to engage with our application as described in *Functionality* and *Usefulness*
  - Matching the design and inter-page transitions as specified in *UI Mockup*
  - Transforming route data passed from the backend databases to the frontend into color-coded routes visualized on a map of the São Paulo area
  - Using Javascript to implement a dynamic, accessible, and functional user interface that matches the specifications laid out in *UI Mockup*
    - Using a frontend web development framework, such as React or a similar framework that works well with Javascript, in support of this goal
- Some backend work, communicating with Yangfei to ensure that our JS/React frontend and his Node.js backend communicate seamlessly

## Kaushik Varadharajan

- Development of a Node.js backend
  - Implementing the functionality of the My Trips page with insert and delete queries
- Backend Pathfinding Algorithm
  - Effectively querying backend databases for data needed to compute optimal routes using advanced query techniques like aggregation, subqueries, etc.
  - Interpreting the graph data given by the GTFS standard to compute fares and travel times between given stops
  - Implementing the [A\\* Algorithm](#) or a similar algorithm to determine a set of optimal routes for a user between a specified origin and destination

## Yangfei Dai

- Frontend-Backend connection
  - Implementing efficient communication (e.g. API calls, sending and receiving user and route data, sending and receiving user authentication credentials to and from the backend & databases) between the application frontend and backend
  - Setting up the server environment where our backend system will run
  - Setting up a HTTP client to make the API requests. Implement the logic for making the API calls and handling responses from Frontend
- Development of a Node.js backend
  - Using Node.js to communicate effectively with our JS-based frontend, sending necessary data

- Communicating with Kaushik to effectively structure the Node.js backend such that it implements the A\* pathfinding algorithm to compute personalized routes for users

### Denis Chen

- Backend & database implementation
  - Creating databases on GCP using MySQL with proper schema that effectively store and organize:
    - User data, such as
      - A user's first and last name, email, and password
      - A user's saved routes
    - "Stop data" - data detailing the location of various *stops* in the SPTrans transit system located throughout São Paulo
    - Fare price data - data detailing the price of various inter-stop routes
    - Travel time data - data detailing the amount of time various inter-stop routes will take
    - Route data - detailing the set of routes between stops available in through the SPTrans transit system located throughout São Paulo
  - Integrating communication between MySQL-on-GCP database and Node.js backend
    - Communicating with Yangfei and Kaushik to effectively pass necessary data to Node.js backend such that routes can be computed for users using the A\* pathfinding algorithm
    - Developing advanced queries in collaboration with Yangfei and Kaushik that allow the Node.js backend to accurately and efficiently obtain data required for pathfinding, user authentication, and more