Yantra’s Road Warrior

Overview

RoadWarrior is an application for anyone who wants to organize their travel better and have their trip sequence and steps all in one place. It makes traveling much more enjoyable and helps travelers focus on the fun part of the trip more than the coordination and hassles part.

The application enables travelers to view their trip and its transition between transportation, stay and local travel in one single dashboard. It organizes itself to present the best view to the travelers based on the information it collects and is also flexible to be changed according to the user's preference.

Use Cases

The following are the use cases for the application between the three main actors - Traveler/User, System, Travel Agency/Agent.

User/Traveler Actor Use Cases

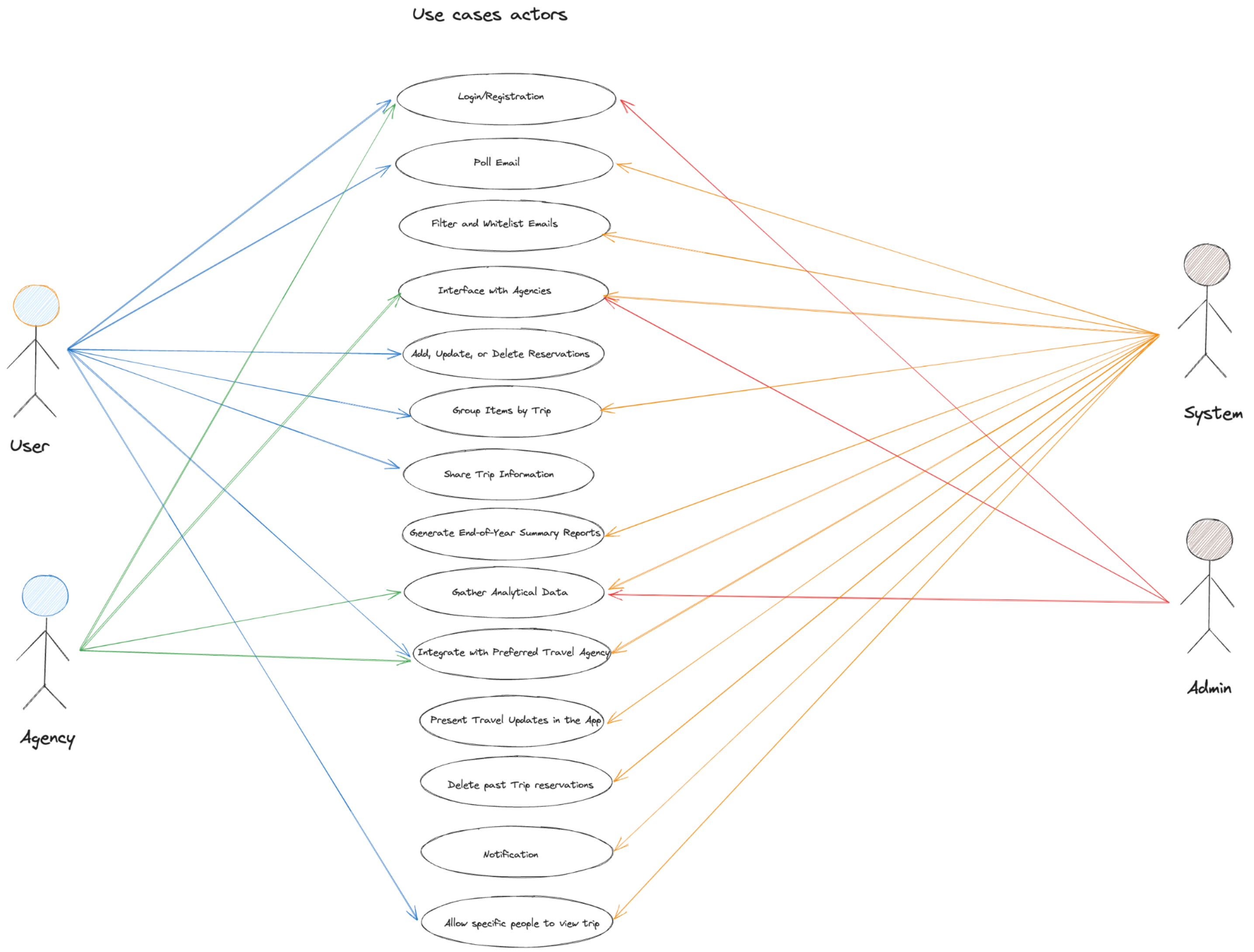
1. Login - Secure Authentication and access.
2. Registration - User Registration with User Information along with consent to access email with access information stored.
3. Add, Update or Delete Reservations/Bookings - User should be able to manually add and update trip information or new trips in the dashboard with the details.
4. Group Items by Trip - While the system by default will group items, users can move the items between trips and edit the dashboard.
5. Share Trip Information - Users will be able to share their trip information in social media with a high level of textual content.
6. Allow Specific Users to View the Trip - Users can share a link to the trip dashboard which is accessible as a view only mode to selected users decided by the user who shares.
7. Integrate with Prefered Travel Agency - User will be able to chat or communicate over mail with a travel agent she chooses.

System Actor Use Cases

8. Poll Email - The system polls the email provider to scan for travel related information.
9. Filter and Whitelist Emails - Based on User Configuration the system filters and whitelist email addresses.
10. Interface with Agencies - The system connects to standard APIs of third party travel agencies to get travel and booking related information.
11. Group items by Trip - The system based on the dates and destination information creates a trip grouping with all bookings of a trip.
12. Generate Reports - The system generates reports from user trip bookings to provide insights for third party consumers.
13. Present Travel Trip Updates - The system notifies the user on changes and updates to their trip.
14. Delete Past Trips - The system automatically removes older trips from the dashboard.

Agent/Agency

15. Login/Registration - Agencies have a separate login into the system.
16. Integrate with Prefered Travel Agent - Agent who is chosen as a preferred agent can respond to user queries through an authentication interface either through chat or email.
17. Allow API Access (Interface with Agencies) - Agent can approve API access from the Road Warrior system.
18. Gather Analytical Data - Agency can view insights and reports of trips.


Use cases actors
Login/Registration
Poll Email
Filter and Whitelist Emails
Interface with Agencies
Add, Update, or Delete Reservations
Group Items by Trip
Share Trip Information
Generate End-of-Year Summary Reports
Gather Analytical Data
Integrate with Preferred Travel Agency
Present Travel Updates in the App
Delete past Trip reservations
Notification
Allow specific people to view trip
User
Agency
System
Admin

Architecture Characteristics

Availability - There is a clear expectation for the system to be available at all times with 5 minutes unplanned downtime per month at max. 99.99% SLA.

Reliability - The users get the data in their dashboard with a max of 5 minute lag and the web response time is 800 ms while mobile first paint time is 1.4 seconds.

Scalability - Total users are 15 million and weekly active users are 2 million. There is a potential for the active users to spike during vacation times and holidays.

Security - Trip data is user sensitive data and privacy requirements are critical. Users will not be willing to share their trips and booking information to the public. Travel agencies would want secure infrastructure through which their APIs are accessed. Additionally the third parties who access travel patterns and insights would be paying for the data consumption which needs to be kept secure.

Usability - Dashboards must be easily consumable across devices. They should be quick to provide the relevant information and notify changes to the users as well.

Performance - Dashboard load time and interaction with the interface should be sub second and without any lag.

Interoperability - The data from third party agencies shall be retrieved from standard APIs. In certain cases the third parties provide callback data through webhooks and events. The system needs to consider extension of new agencies and provide end points for notification of events as well in the long run.

Multilingual - The UI and the data should be developed and stored in unicode compliant format with support for multiple languages.

Learnability - One of the monetizing modes is to provide travel patterns of users to third party agencies. In the long run there can be use of this data to predict and alert travel agents on the bookings and trips which can potentially add value. Learnability is one important aspect to be considered.

Architecture Decision Records

Microservice Architecture

Context

The application needs 99.99% availability. It also needs to scale between 2 to 10 million users with a total of 15 million users. The application has multiple dedicated data sources from where the data needs to be collected and processed.

Decision

The microservices based architecture helps to scale services independently and ensures easier implementation of failovers by isolating deployment units with clear boundaries. The architecture will use a combination of events and explicit calls to components to satisfy the requirements.

Scheduled Third Party Calls

Context

The application requires trip data to be collected from third party email providers by parsing trip related emails and also by accessing third party APIs. Both these operations are expensive operations considering multiple APIs and email parsing being an expensive operation.
Cost being API access cost and 15 million users to be updated on their trips.

15 Million Email Accounts to be parsed at minimum and 15 * 3 = 45 Million API calls at the minimum to the third party agency APIs to update the users trip information once.

The company is a startup. While the competitors are giving 5 minute updates, we are making an assumption that 5 minute updates are applicable for users actively looking at the dashboard and during the last week before their travel date.

Decision

During non dashboard access time and for the users whose trip start date is more than one week away we will be calling the email API and third party APIs for trip data only twice in a day.

For active and close to travel date users(within a week) we will be providing notification and UI updates within every 4 minutes.

The active sessions of the users are stored in a distributed cache for the service to check for which users would need 5 minute updates.

Event Notification Capability

Context

The Application shall notify changes to the trip information. This is because the traveler has to know the changes to the booking and timings of the hotels and the transport arrangements and connections.

Decision

Any change to the trip information across all the bookings of the trip will be notified as a push notification into the App and an email shall be sent to the user’s email address.

Analytics Database

Context

The data from the different travelers are consolidated and insights from the data are provided to the travel agencies and third party consumers. This is part of the requirement. This has a potential to generate a revenue stream for the business

Decision

A dedicated analytics database for the trips will be invested upfront. This will help us to run analytical queries and data processing over the trip data of 15 million users. The insights generated from the analytics query will be provided as reports and within the user interface of the app for the agency users.

Prefered Agency Configuration

Context

The user can chat with a preferred agency to get assistance through the app.

Decision

We are providing the user to choose a preferred travel agent during the registration process among a list of travel agents who provide us with the technical capabilities to reach them through APIs or online chatbot. Based on this configuration the chat plugin within the app will contact the specific travel agent. If the online presence is not available at the time of a help request from the user, the transcript is sent through email and the response is captured and put into the chat content.

Guest or Read-Only Access

Context

The user can share their trip data with select friends. We want to provide read only access to trip information apart from a plain text or a simple social media post. This will help us to increase foot fall to the application and potentially convert other users to register for their own trips as well.

Decision

A read only view with specific authentication for individual trips of the user shall be provided and a login flow for such guest users will be made available.

Overall Architecture

Description

The application would follow the microservice architecture style. The two key services are MailParserService and TravelAPIAggregator.

MailParserService is responsible to read the emails and check for trip related emails and updates from them. It parses the mails based on whitelisting of the user and /or based on keywords of travel, booking and trips.

TravelAPIAggregator is an event listener as well as API invoker. Based on the external agencies capabilities it calls the APIs or provides listeners or hooks for them to notify trip related information.

Both these services use an inbuilt logic to construct trip data and store them into the TripDatabase which is a NoSQL database.

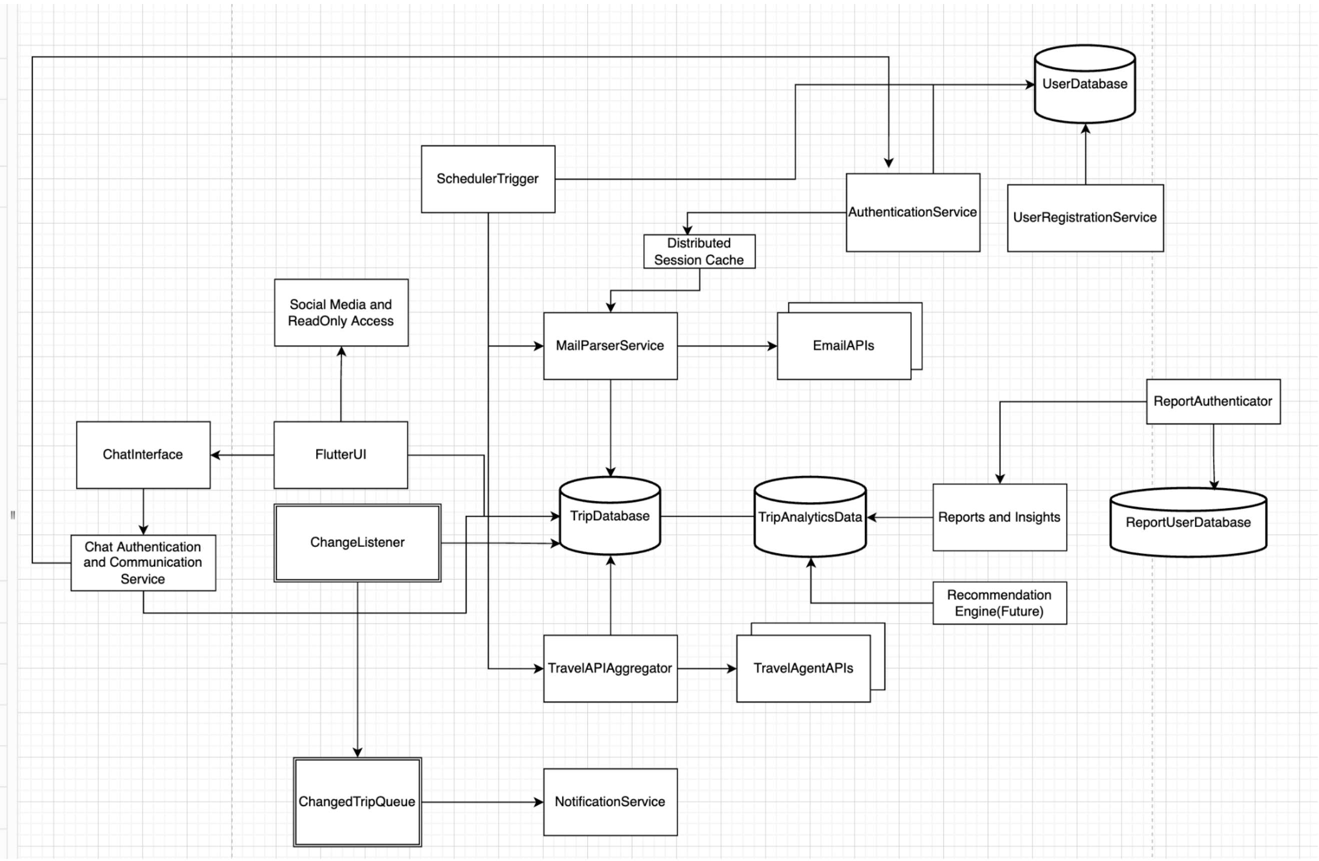
The other key services are AuthenticationServices for users and data consumers(UserDatabase), a reporting analytics database for generating long term data insights(TripAnalyticsData), and a chat/help interface with the preferred travel agent of the user.

We recommend Flutter as the UI technology which provides rich user interface experience and is a framework where we can write once and run on multiple devices and browsers.

There is a notification triggered from the system if the TripData is changed and there is a listener(ChangeListener) and queue(ChangedTripQueue) capabilities required for the same.

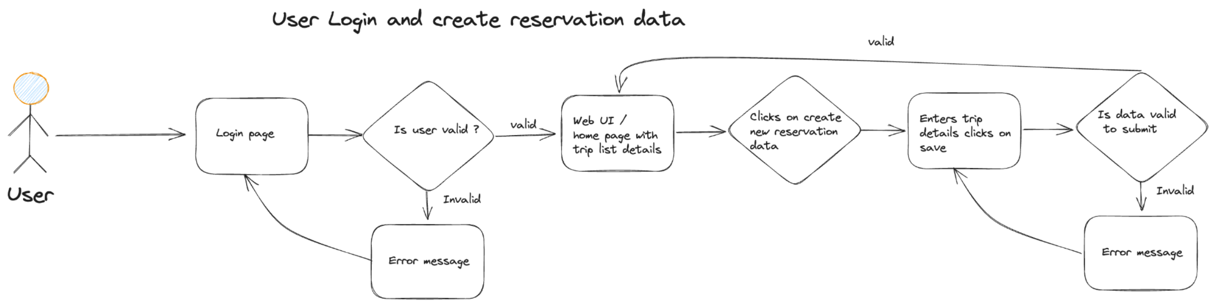
(components flow represented in the diagram below)

Diagram

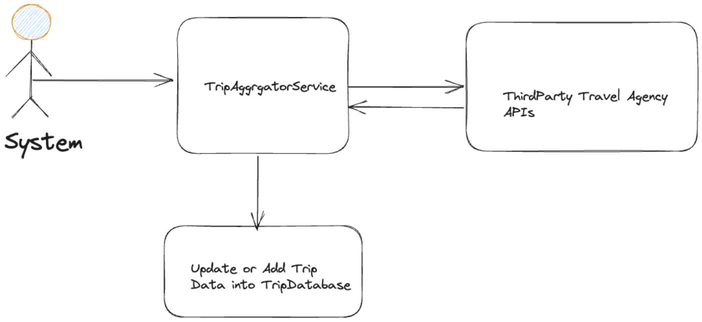

UserDatabase
SchedulerTrigger
AuthenticationService
UserRegistrationService
Distributed Session Cache
Social Media and ReadOnly Access
MailParserService
EmailAPIs
ReportAuthenticator
ChatInterface
FlutterUI
TripDatabase
TripAnalyticsData
Reports and Insights
ReportUserDatabase
Chat Authentication and Communication Service
ChangeListener
Recommendation Engine(Future)
TravelAPIAggregator
TravelAgentAPIs
ChangedTripQueue
NotificationService

Detailed Flows of Key Use Cases based on the Architecture

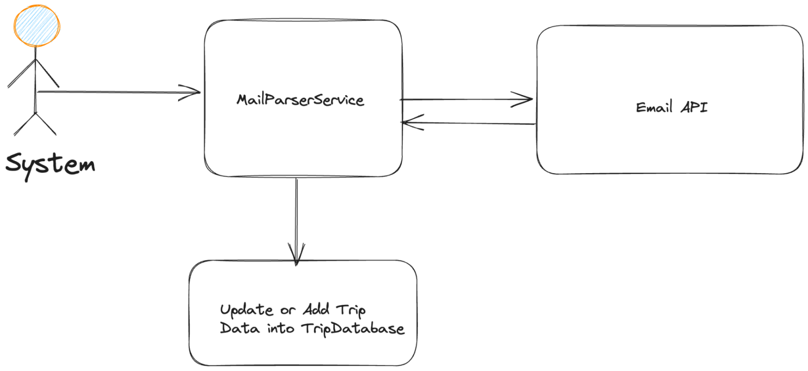
User Login and Trip Addition


User Login and create reservation data
User
Login page
Is user valid ?
valid
Invalid
Error message
Web UI / home page with trip list details
Clicks on create new reservation data
Enters trip details clicks on save
Is data valid to submit
valid
Invalid
Error message

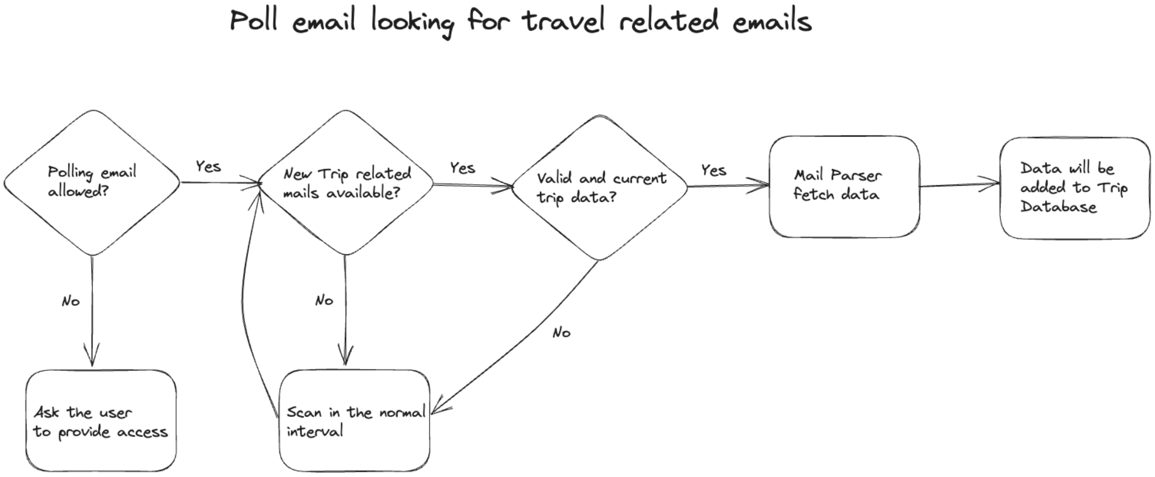
Listening to and Accessing Third Party Travel Agency APIs


System
TripAggrgatorService
ThirdParty Travel Agency APIs
Update or Add Trip Data into TripDatabase

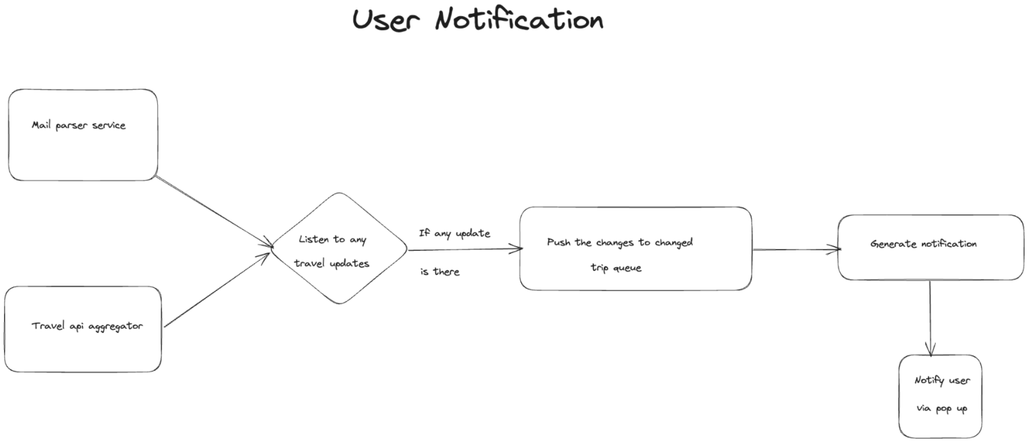
Accessing Emails and Finding Trip Information



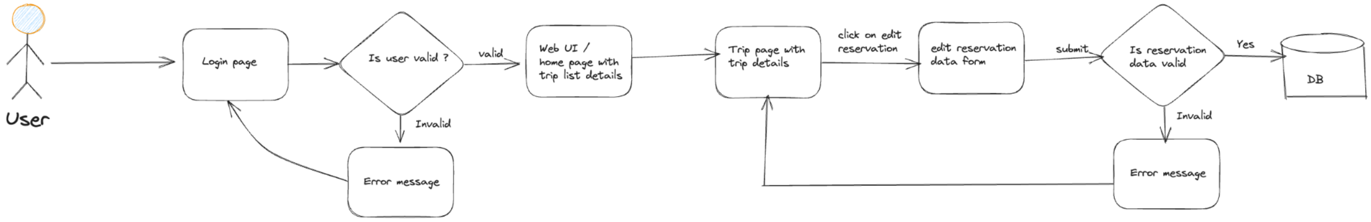
Email Parsing Flow Detail


Poll email looking for travel related emails
Polling email allowed?
Yes
No
Ask the user to provide access
New Trip related mails available?
Yes
No
Scan in the normal interval
Valid and current trip data?
Yes
No
Mail Parser fetch data
Data will be added to Trip Database

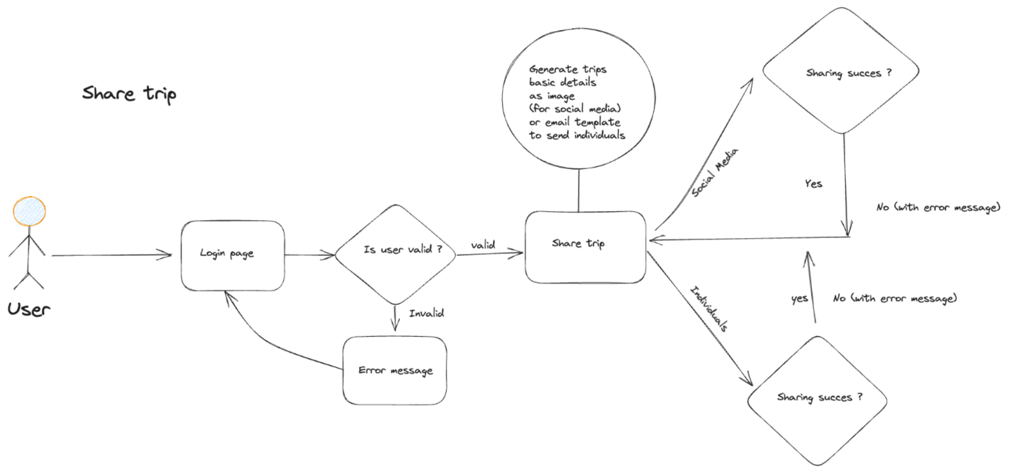
User Notification on Trip Data Change



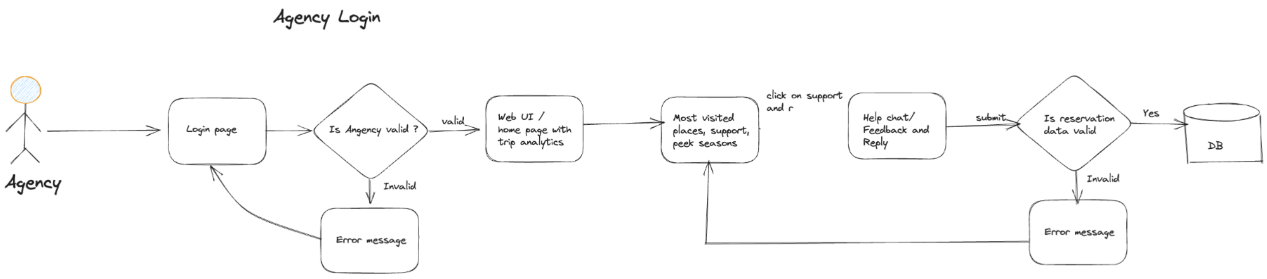
User Editing Trip Information in the Dashboard


User
Login page
Is user valid ?
valid
Invalid
Error message
Web UI / home page with trip list details
Trip page with trip details
click on edit reservation
edit reservation data form
submit
Is reservation data valid
Yes
Invalid
Error message
DB

User Sharing their trip with friends with view only access

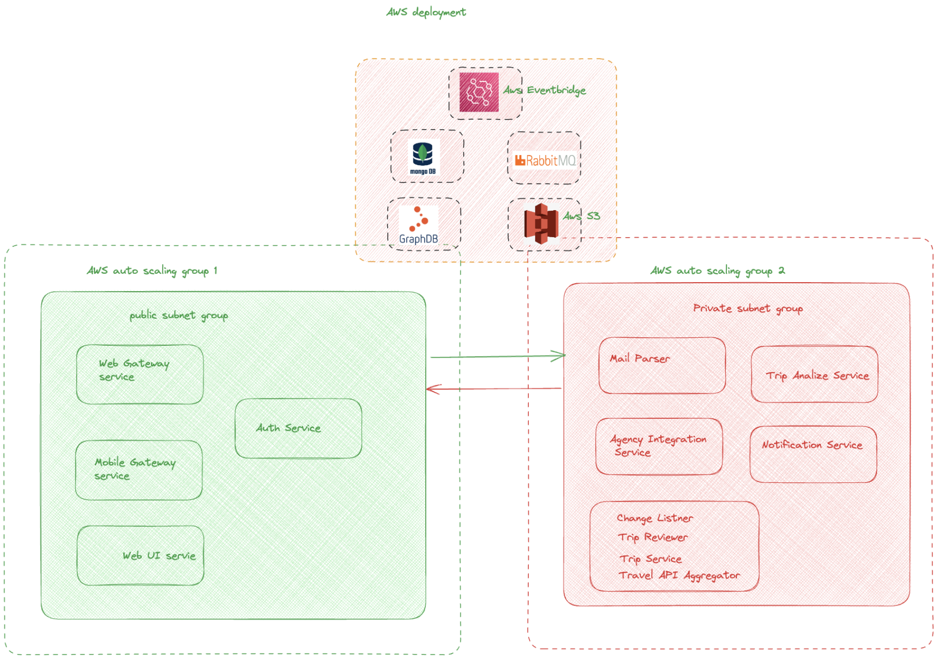


Third Party or Agency user who has specific access to data


Agency Login
Agency
Login page
Is Angency valid ?
valid
Invalid
Error message
Web UI / home page with trip analytics
Most visited places, support, peek seasons
click on support and r
Help chat/ Feedback and Reply
submit
Is reservation data valid
Yes
DB
Invalid
Error message

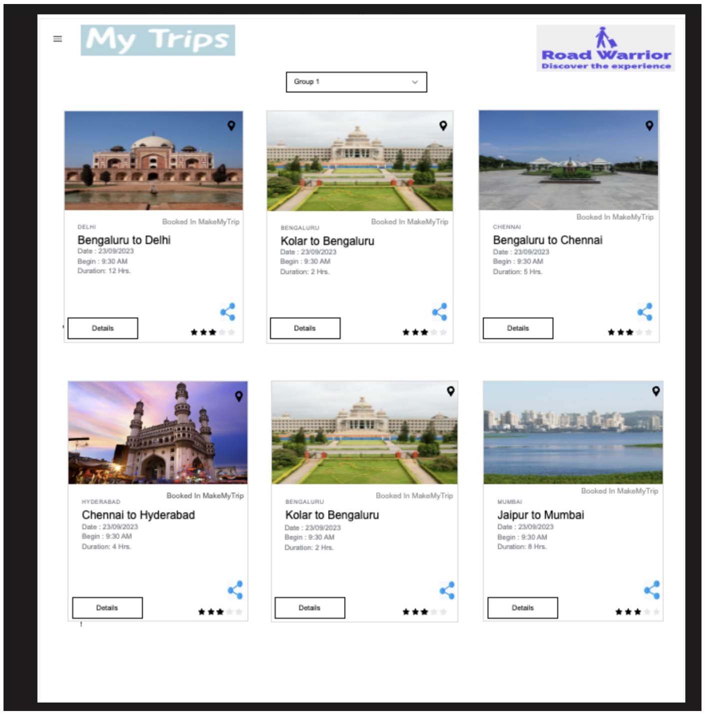
Technical View Exploration

AWS Components which can suite some of the Architecture Requirements



Initial Mockups

Dashboard View



Booked In MakeMyTrip
DELHI
Bengaluru to Delhi
Date : 23/09/2023
Begin : 9:30 AM
Duration: 12 Hrs.

Booked In MakeMyTrip
BENGALURU
Kolar to Bengaluru
Date : 23/09/2023
Begin : 9:30 AM
Duration: 2 Hrs.

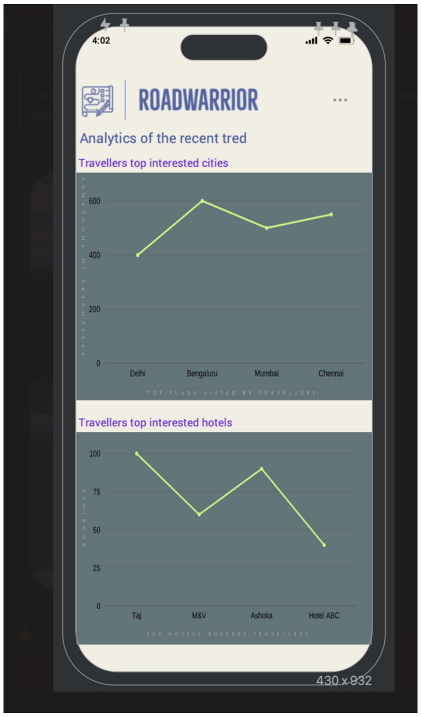
Booked In MakeMyTrip
CHENNAI
Bengaluru to Chennai
Date : 23/09/2023
Begin : 9:30 AM
Duration: 5 Hrs.

Booked In MakeMyTrip
HYDERABAD
Chennai to Hyderabad
Date : 23/09/2023
Begin : 9:30 AM
Duration: 4 Hrs.

Booked In MakeMyTrip
BENGALURU
Kolar to Bengaluru
Date : 23/09/2023
Begin : 9:30 AM
Duration: 2 Hrs.

Booked In MakeMyTrip
MUMBAI
Jaipur to Mumbai
Date : 23/09/2023
Begin : 9:30 AM
Duration: 8 Hrs.

Analytics View



Analytics of the recent tred

Travellers top interested cities

Travellers top interested hotels