Software Requirement Specification (SRS)

User Stories

Local Farmer

Local farmers drive on rural roads, and need to go to towns on a regular basis.
As a local farmer, I want to drive to the nearest flea market to sell my goods. I want to know the weather forecast for my trip since I leave at consistent times each week. I'm also driving on gravel roads so the weather might be more affecting than it should.

Acceptance Criteria

- I want to be assured that the app takes into account road conditions.
- To improve my future experience, I want to be able to give feedback regarding my route such as if it's a gravel road and wet after last week's rain. This improvement of risk calculation should be specific to my route and not to some oak's road in a different part of the country.

Daily Commuter

Daily commuters are the average Joe that drives to work/school/university everyday.
As a working daily commuter, I want to ensure that I arrive at work on time. Weather happens, and for that reason I want to be able to take that in account. There's also traffic on my road to work, so that's important. However, to check Google Maps everyday is tedious and time consuming.

Acceptance Criteria

- I want to set up some routine alerts so that I don't have to check my phone every morning.
- Since I'm commuting daily, I want to be able to set routine alerts such that I can be notified when it's a bad day to be on the road.

Delivery Personnel

Delivery personnel are responsible for delivering packages to customers on a daily basis.
I need to make 25 trips a day to keep my family fed. It is fast pace and highly dangerous with many of my fellow employees ending up in accidents. I could use an application that reduces the risk of my trips and increases my chances of safely returning home everyday.

Acceptance Criteria

- I want to know what routes create the greatest risks.
- I need to also be warned of possible risks in areas to know which jobs to take and which jobs to stay clear from.
- Since my truck is limited by its speed I want to let that be taken into account as well.

Long-haul Vacation Goers

A long-haul vacation goer is someone who takes on long roadtrips, mostly with their families. As I'm planning my vacation, I'm always worried about my family's safety when travelling long distances. For that I am checking Google Maps well in advance to plan the route, but it's not possible to see the risk associated to the route.

Acceptance Criteria

- I want to plan my trip well in advance (in terms of date and time) and get ongoing alerts and recommendations with regards to my trip.
- I want to avoid highways that are under construction.

Functional Requirements

1. User Management Subsystem
   1.1. The system shall securely authenticate the user.
   1.2. The system shall manage authentication token lifecycles.
   1.3. The system shall make sure certain pages are protected when not authenticated.
   1.4. The system shall have user options such as car type to adjust the risk.
2. Risk Map Subsystem
   2.1. The system shall provide an immersive risk map that gives the user a risk overview based on their location.
   2.2. The system shall improve the risk overlay based on user feedback.
3. Trip Planning Subsystem
   3.1. The system shall allow the user to plan their trips by giving start and end locations.
   3.2. The system shall allow for multi-stop itineraries.
4. Routine Alerting Subsystem
   4.1. The system shall alert users who planned their trips if there are added risk to the route.
   4.2. The system shall allow the user to setup routines which will notify them on a daily basis.

Non-functional Requirements

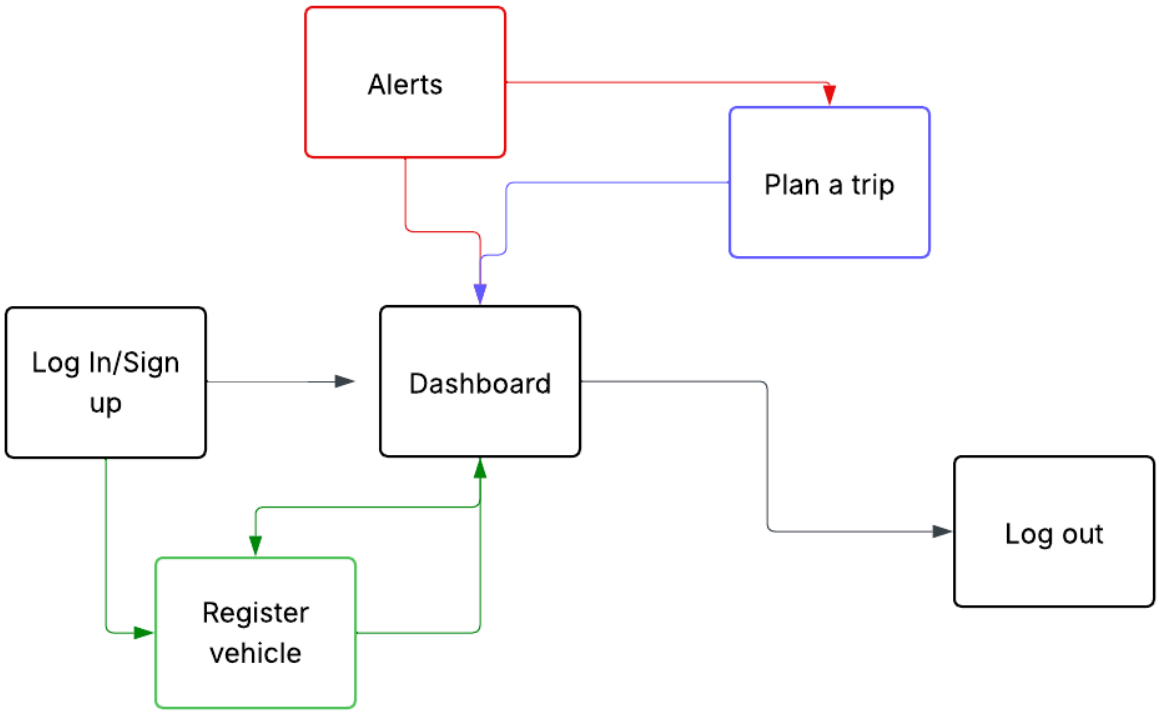
1. Immersive user experience

1.1. Welcoming dashboard.
1.2. Friendly dialog to log in and gather data.

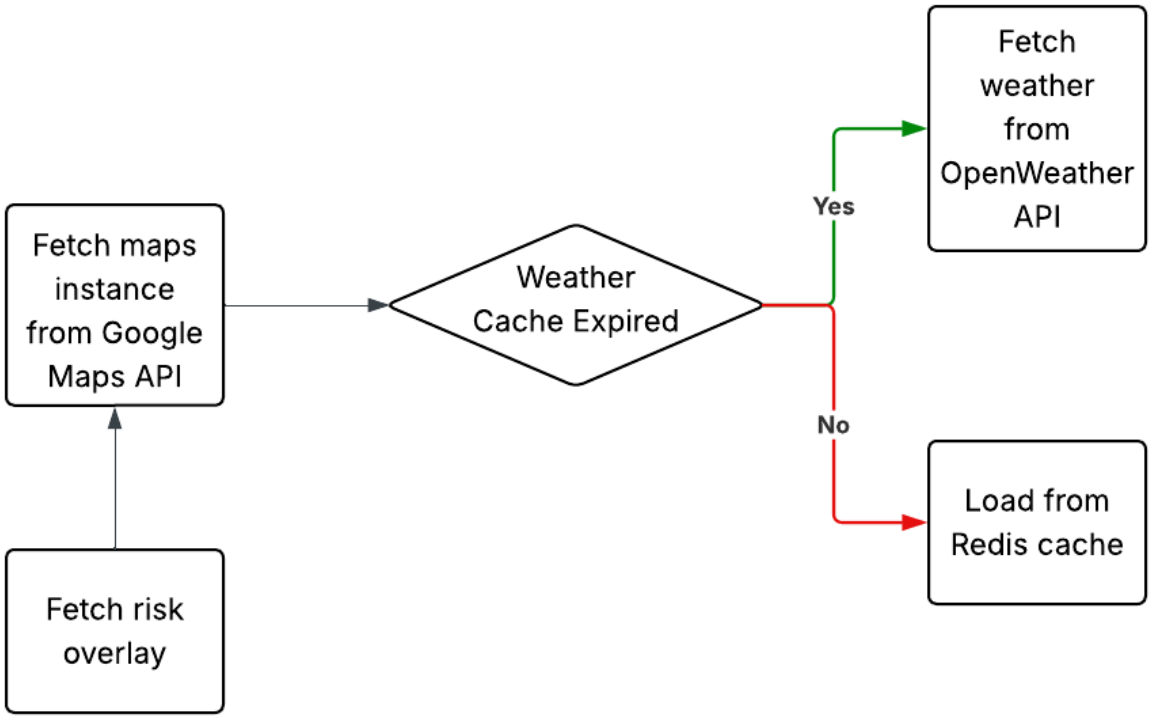
2. Simple navigation that minimizes clicks
   2.1. Have a clear flow of data.
   2.2. Guide the user, never leave the user guessing what to do next.
3. A feeling of a companion
   3.1. The user should not see the app as another chore that his life insurance requires off him, but rather see it as a buddy that wants to keep him safe.
   3.2. The app should be apart of his daily life on the road.

Data Flows

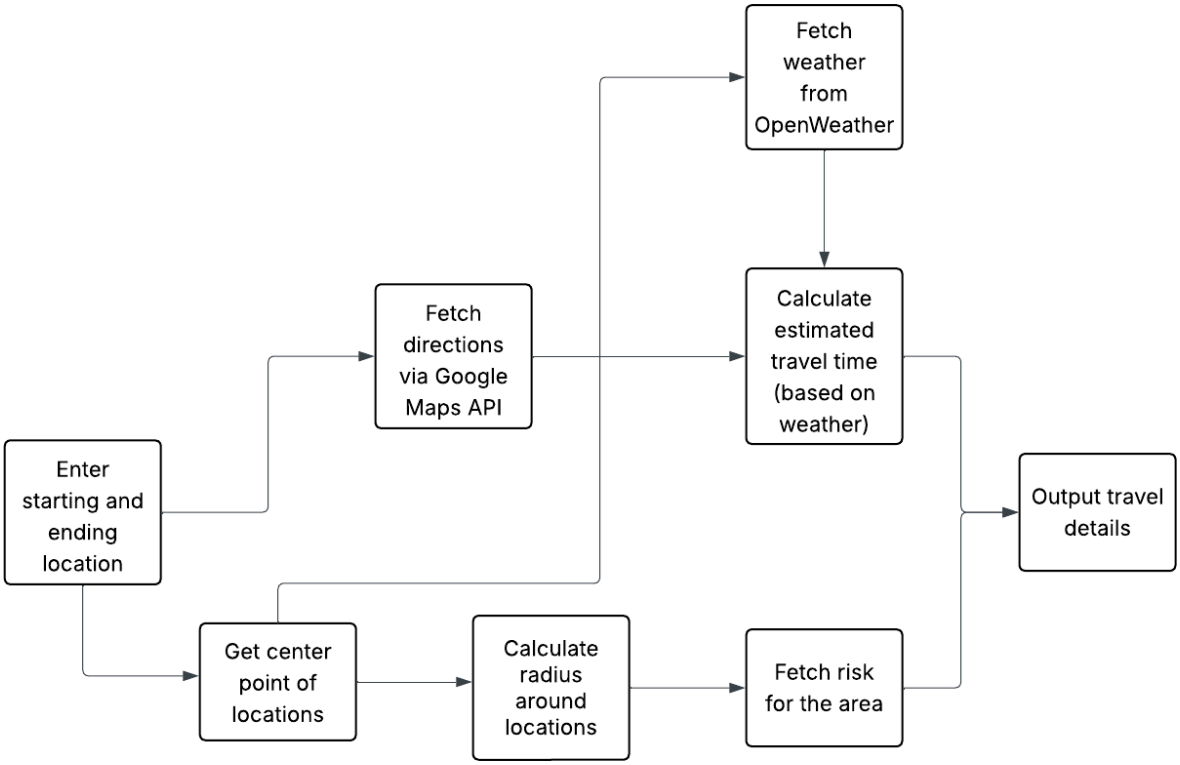
High-level System Data Flow



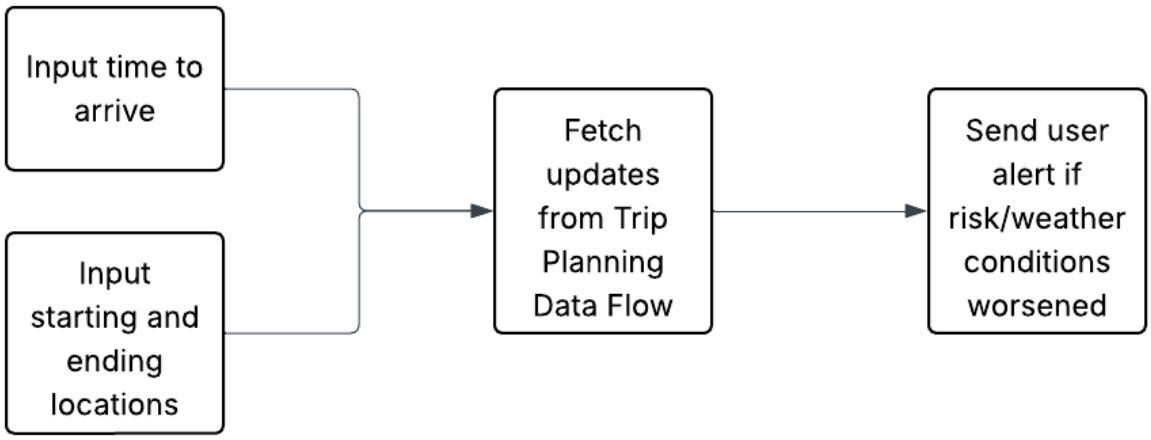
Risk Map Data Flow



Trip Planning Data Flow



Routine Alerting Data Flow



Architectural Patterns

Our architecture follows a simple client-server architecture. The client-server pattern allows for quick iteration and a simplified understanding of our codebase. Furthermore, this pattern also allows to offload our expensive business logic to the server instead of relying on the client's hardware.

Tech Stack

For our tech stack we chose NextJS as our full-stack framework. This allows for quick iteration and performs well at scale. We make use of a relational PostgreSQL database which will allow us to expand into future plans (such as implementing an XGBoost regressor on top of our data). To keep our types synced up, we use Prisma as our ORM. If time allows, we'll develop a high performance FastAPI server to cater for our Machine Learning needs.