RIDE WITH ME

CS5224 – Cloud Computing

Final Report

16 April 2023

Aalok Nandlal Chhabria	(A0042604U)
Parul Bansal	(A0268257R)
Qiao Guanheng	(A0248358R)
Sreelakshmi Edakulathil Chellappan	(A0268357N)

Ride with Me – A Cloud-based Car-pooling Service

1. Executive Summary

Ride-hailing and taxi services in Singapore are in strong demand. These services include all forms of online and offline bookings that connect drivers and passengers. In Singapore, the revenue generated by this sector is an estimated US$1.35 billion and market penetration stands at 32.9% in 2023 [1]. One important segment of the ride-hailing market is car-pooling. Given steep prices imposed by private ride-hailing operators and the environmental impact of carbon emissions contributed by the transportation industry, we believe that drivers and passengers have compelling reasons to move to car-pooling.

The objective of our web application is to offer car-pooling services that match drivers and passengers (or riders) who are travelling in the same or similar directions. Our web application is named "*Ride with Me*" and is implemented using the Software-as-a-Service service model deployed on AWS cloud. Unlike typical ride-haling applications which match drivers and passengers based on their current location and do not consider the destination, our underlying matching algorithm takes into account the users' starting points, destinations, trip timing and conducts a route analysis to ensure minimal amount of detour for the driver. This intelligent algorithm recommends drivers to riders, and vice-versa, while leveraging Google's Routes API.

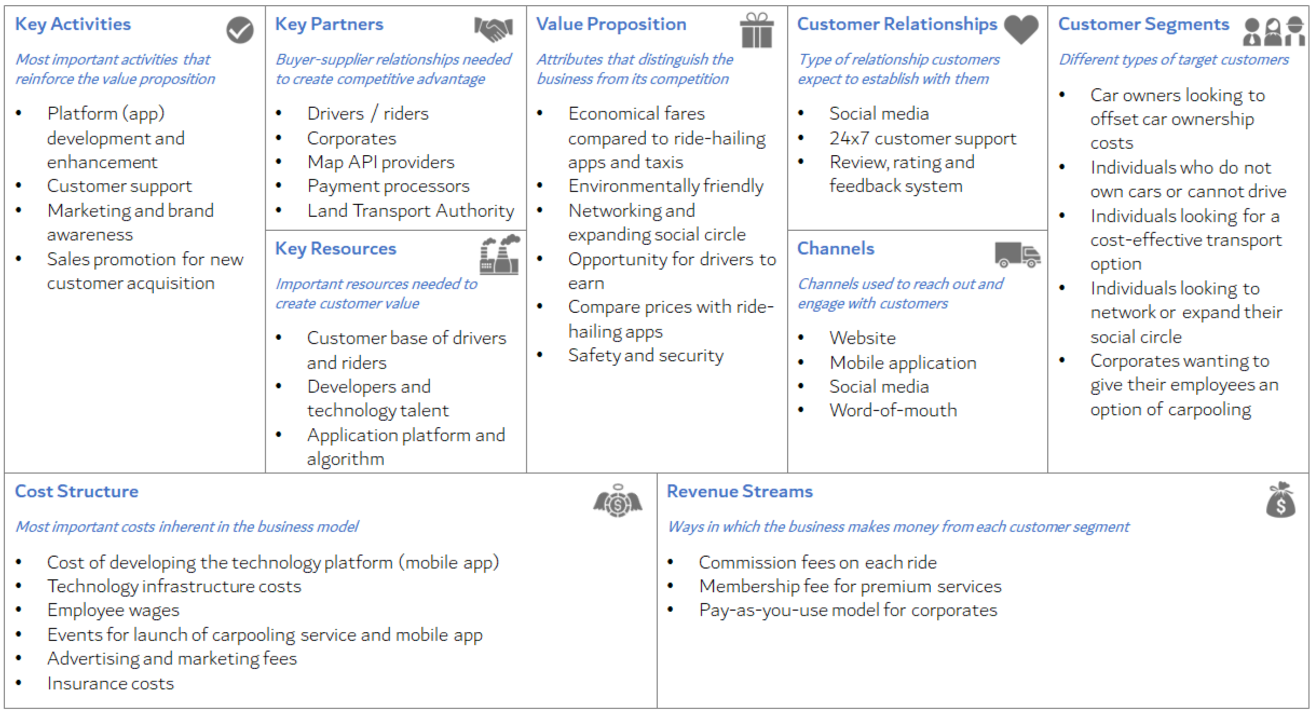
2. Business Case Identification

The ride-hailing market in Singapore is dominated by 3 key players - Grab, ComfortDelgro and Gojek. In research published by Blackbox [2], these applications are the most used by consumers with Grab dominating usage at 74% compared to ComfotDelgro's 12% and Gojek's 11%. With its dominance in the market and high take-up rate among riders, Grab introduced a steep increase in rates through a surge pricing model based on demand and supply. This has made the use of ride-hailing apps in Singapore very expensive for consumers.

Having said that, an alternative and more economical mode of ride-hailing is the concept of car-pooling. Our project focuses on creating a cloud-based application "*Ride with Me*" which offers consumers the option to share a ride with someone who is travelling in the same or similar direction.

3. Business Model

The diagram below depicts the business model canvas [3][4] of *Ride with Me*. Some key features are explained in further detail below.



Key Activities

Most important activities that reinforce the value proposition

- Platform (app) development and enhancement
- Customer support
- Marketing and brand awareness
- Sales promotion for new customer acquisition

Key Partners

Buyer-supplier relationships needed to create competitive advantage

- Drivers / riders
- Corporates
- Map API providers
- Payment processors
- Land Transport Authority

Key Resources

Important resources needed to create customer value

- Customer base of drivers and riders
- Developers and technology talent
- Application platform and algorithm

Value Proposition

Attributes that distinguish the business from its competition

- Economical fares compared to ride-hailing apps and taxis
- Environmentally friendly
- Networking and expanding social circle
- Opportunity for drivers to earn
- Compare prices with ride-hailing apps
- Safety and security

Customer Relationships

Type of relationship customers expect to establish with them

- Social media
- 24x7 customer support
- Review, rating and feedback system

Channels

Channels used to reach out and engage with customers

- Website
- Mobile application
- Social media
- Word-of-mouth

Customer Segments

Different types of target customers

- Car owners looking to offset car ownership costs
- Individuals who do not own cars or cannot drive
- Individuals looking for a cost-effective transport option
- Individuals looking to network or expand their social circle
- Corporates wanting to give their employees an option of carpooling

Cost Structure

Most important costs inherent in the business model

- Cost of developing the technology platform (mobile app)
- Technology infrastructure costs
- Employee wages
- Events for launch of carpooling service and mobile app
- Advertising and marketing fees
- Insurance costs

Revenue Streams

Ways in which the business makes money from each customer segment

- Commission fees on each ride
- Membership fee for premium services
- Pay-as-you-use model for corporates

Figure 1 - Business Model Canvas

The 3 main value propositions for our customers are the following:

- Less expensive compared to existing ride-hailing options because the driver is not part of a commercial entity but an individual offering this service for a nominal fee.
- Environmentally friendly since the driver and rider are sharing a single vehicle to their destination. This also results in less traffic on the road.
- Drivers and riders have an opportunity to network and expand their social circle by meeting like-minded individuals during their journeys.

3.1. Revenue Model

A major component of our business model is the revenue generation process. The primary mode of revenue will be through a commission on each trip facilitated by our platform. This would be in the form of a 10% fee applied on the payment received by the driver.

Secondly, we also propose a freemium model where the basic service of requesting for a car-pooling driver or rider is free but a fee applies to premium services such as package delivery.

Lastly, we propose to charge companies on a pay-as-you-use basis depending on the number of their employees using *Ride with Me*.

3.2. Target Users

The best opportunity for car-pooling is usually on weekday mornings and evenings when there is a large movement of people and traffic across various locations in Singapore. This movement is a consequence of professionals heading to or returning from their workplace. Hence, our main target users are professionals and young adults.

Professionals who own cars and drive to work will form the driver customer base. On the passenger front, professionals and young adults who are early on in their careers and prefer an economical ride will form the passenger customer base.

We also propose to target corporates through our web application. Individuals who sign up with their companies can receive discounted rates. Note that this feature is not part of the prototype.

3.3. Comparison with Available Services

The main comparable car-pool services in Singapore are Ryde and Grab Hitch [5]. We set up accounts on both these platforms and found that they make recommendations to drivers based on the nearest available passenger. The limitation with type of matching is that it is a one-dimensional, location-based approach. It does not consider the starting and final destination of the <u>driver</u> which would mean that any riders who might be "*along the way*" of the driver's route will not be considered. We aim to change and improve this approach in *Ride with Me.* Our matching algorithm will not only consider the passengers' starting point and destination but also the drivers'. This will give us a more robust and intelligent matching service.

4. SaaS Architecture

The diagram below illustrates how drivers and riders interact with the *Ride with Me* application.

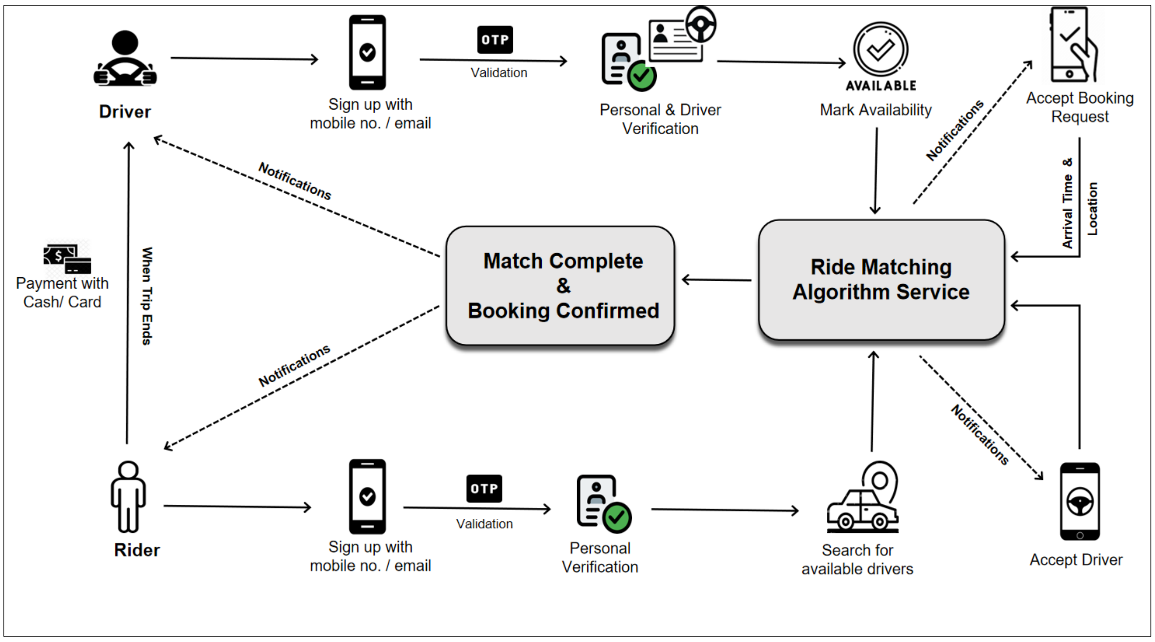


Figure 2 - User Interaction Workflow

The diagram below provides a high-level overview of our cloud-based service.

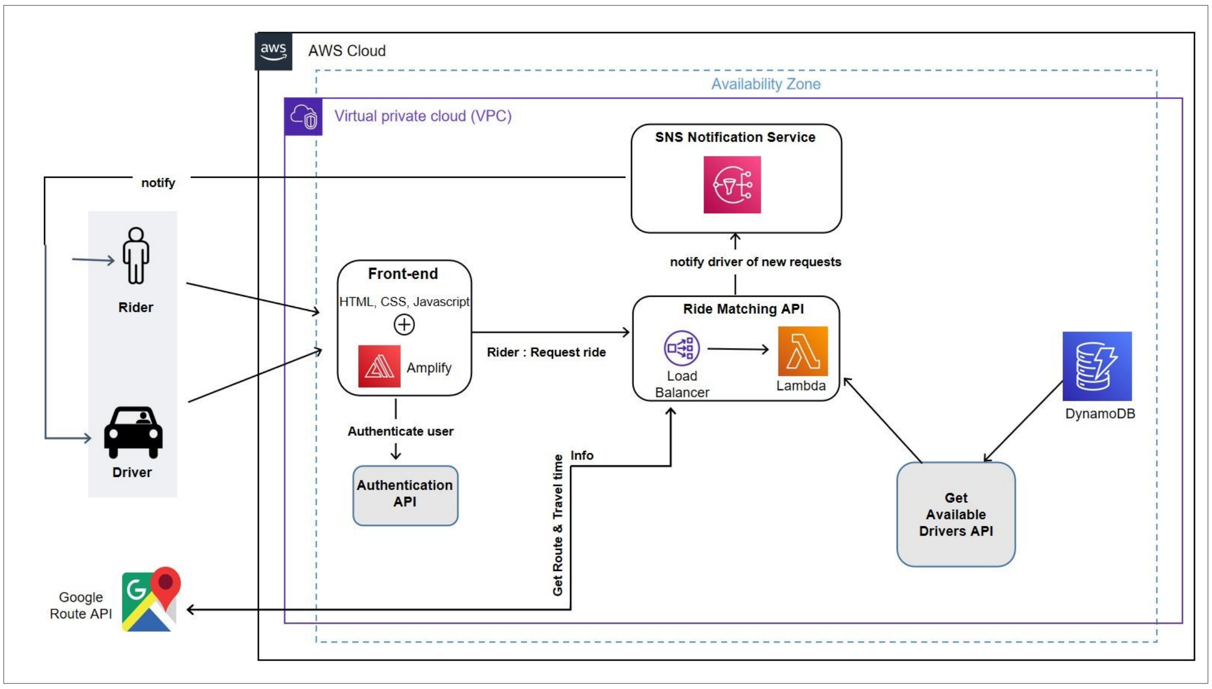


Figure 3 - System Overview

The *Ride with Me* application and service is made up of the following main components:

1. Front-end user interface (UI)

- Module for user authentication – allows drivers and riders to register for a new account or login to an existing one. This includes a form to enter user and account details.
- Module to request for a driver or rider – allows users to enter their trip information (start and end locations) and the desired pick-up time. Based on the routing algorithm, users are given an option choose their driver / rider and confirm their choice.

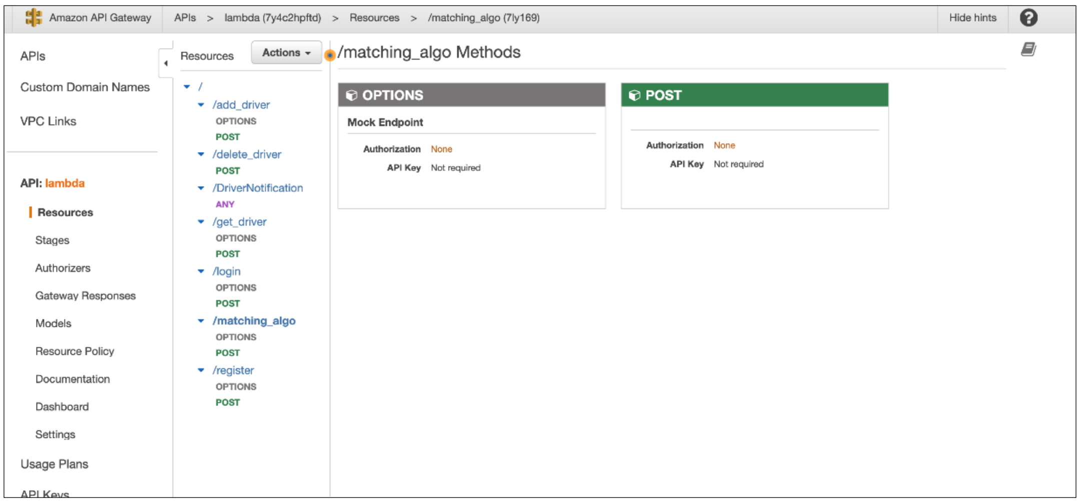
2. Back-end application logic
   - Module for user authentication – invokes a user registration API which stores user and account details into a database and a login API which reads from the database to verify login credentials.
   - Module for route matching – takes the users' start and end locations as inputs and invokes Google's Routes API which provides distance and travel time information. The application compares the routing results and matches best possible drivers and riders.
   - Module for user notifications – based on the route matching results, this module sends information to a server. The server then produces a message which is added to a message queue. Thereafter, notifications are sent to drivers and riders through the application. Users then decide whether to accept or reject the ride and the notification module provides a final confirmation of whether the trip is confirmed.
3. Data storage
   - Module for database service – stores user and transactional data used by *Ride with Me*. We have a database with the following 2 main tables:
     - User details and account credentials
     - Trip details including driver and rider, start and end locations and times

5. SaaS Implementation

The following cloud-based components have been used to implement *Ride with Me*. The web application is accessible here: https://main.dsonn1b7hjj5h.amplifyapp.com/.

5.1. AWS API Gateway

We used API Gateway as a bi-directional communication channel to connect the front-end interface to the back-end logic. POST and OPTIONS method are defined in API Gateway and linked to corresponding Lambda functions in the back-end. Our front-end code initializes requests and sends user input to a specific method in the API Gateway. Then, the corresponding Lambda function is triggered and handles the request with implemented business logic. This approach is used for the main matching algorithm when drivers become available and riders request for a ride. Lastly, the API Gateway will pass the result generated by Lambda back to the front-end and the driver-passenger matches will be rendered on the web page.



OPTIONS — Mock Endpoint
Authorization None
API Key Not required

POST
Authorization None
API Key Not required

Figure 4: List of implemented APIs

A summary of all the API endpoints is as follows:

- Register API – registers a new user on the *Ride with Me* platform
- Login API – performs the authentication process when a user logs in
- Add driver API – adds a new driver on the *Ride with Me* platform
- Get driver information API – retrieves information about a driver is available and accepting rides
- Delete driver API – removes a driver from the list of available drivers after he has been assigned a trip
- Notify driver API – informs a driver that a rider has accepted his trip. We need to manually add driver end points in the SNS.
- Google Routes API – sends driver and rider information (origin, destination, date and time of trip) to the Google Routes API
- Matching algorithm API – analyse the information returned by the Routes API and based on the criteria, matches drivers with riders, and vice versa. If a driver's original route and rider's route do not cause a re-routing of more than 6KM, the driver and rider are matched.

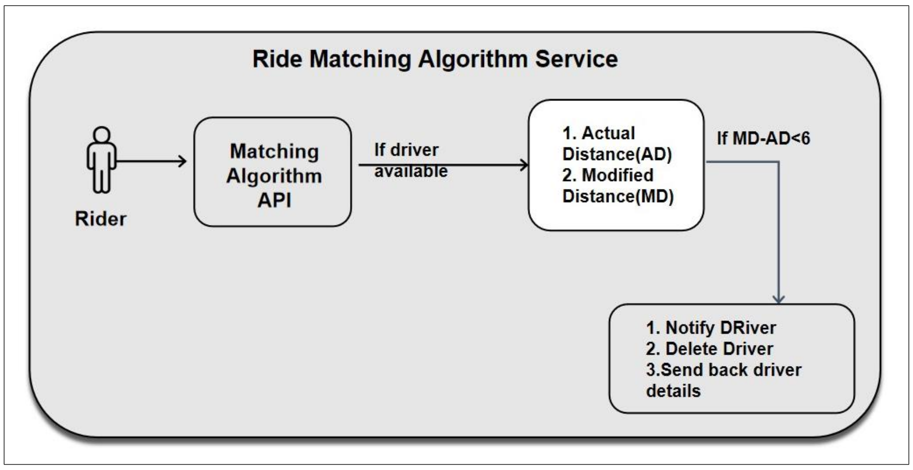
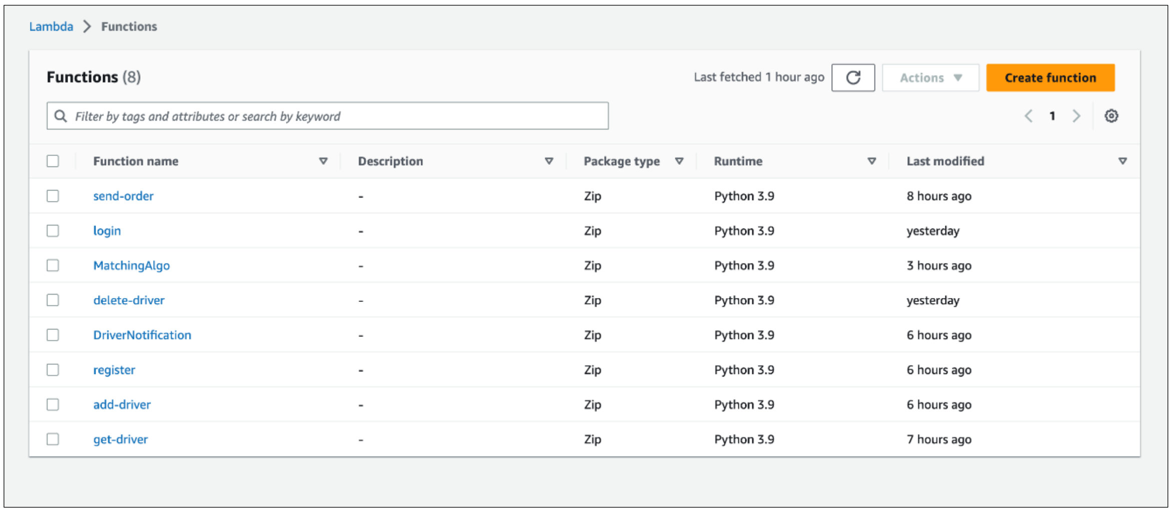


Figure 5: Ride Matching Algorithm Service

5.2.AWS Lambda

AWS Lambda is a serverless compute service that allows developers to create Python functions. Its pricing depends only on the number of calls executed by the Lambda functions. It also manages auto scaling automatically. AWS Lambda provides basic packages like NumPy which we used for implementing our matching algorithm. We use Lambda to handle the API requests sent from Amplify via API Gateway, including getting the starting and final destination of users from DynamoDB and matching drivers with riders. We added python layers to support external python libraries such as google maps and requests and datetime. We used AWS Lamda to create our API's and deployed them using API gateway.



Functions (8)

Function name	Description	Package type	Runtime	Last modified
send-order	-	Zip	Python 3.9	8 hours ago
login	-	Zip	Python 3.9	yesterday
MatchingAlgo	-	Zip	Python 3.9	3 hours ago
delete-driver	-	Zip	Python 3.9	yesterday
DriverNotification	-	Zip	Python 3.9	6 hours ago
register	-	Zip	Python 3.9	6 hours ago
add-driver	-	Zip	Python 3.9	6 hours ago
get-driver	-	Zip	Python 3.9	7 hours ago

Figure 6: List of Implemented Lambda Functions

5.3. AWS Amplify

AWS Amplify is a cloud-based solution which allows web developers to build and host full-stack applications. It manages elastic load balancing (ELB) automatically. We linked our GitHub repository with AWS Amplify and deployed our front-end code to it. Amplify automatically detects the *index.html* file as the homepage of our web application and provides a URL to it. Every time we update our code and synchronize it with Amplify, it will automatically provision, build, deploy and verify to ensure that the latest version of our web pages is deployed. We Used AWS amplify to deploy our application.

5.4. AWS DynamoDB

AWS DynamoDB is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications. With *Ride with Me*, we used DynamoDB to store our users' and cars' information. This includes the user's name, mobile number and email, among others. For the cars, this includes the license plate number and model, among others.

5.5. Google Routes API

The Google Routes API helps find the ideal route from point A to B, calculates ETAs and distances for matrices of origin and destination locations.

5.6. AWS Simple Notification Service

Amazon Simple Notification Service (SNS) sends notifications two ways, A2A and A2P. A2A provides high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. We used SNS to send text messages to driver about their riders in notification API

5.7. AWS CloudWatch

Amazon CloudWatch collects and visualizes real-time logs, metrics, and event data in automated dashboards to streamline application infrastructure and maintenance.

6. Economic Factors

The table below lists the cloud services used and their estimated costs. We made the following assumptions to calculate the cost: There are 1000 users per day, including drivers and riders, and each user makes 5 API calls. The data storage size of DynamoDB is 10 GB. There are 20000 SQS sent per month.

Service	Monthly cost
API Gateway	$0.15
Lambda	$0.11
DynamoDB	$6.25
Amplify	$12.37
SNS	$0

7. Auto Scaling & ELB

As we are utilising a SaaS cloud service model, the auto scaling and elastic load balancing will be managed by AWS Lambda and AWS Amplify, respectively. When there is an increase in the number of users accessing the webpages and requesting for rides, AWS Lambda and AWS Amplify will assign more resources to handle the higher traffic.

8. Conclusion

The *Ride with Me* application implements a proprietary algorithm which takes into consideration the locations, routes and travel times of several users before offering matching suggestions. This calculation engine which will determine the similarity of different routes is a unique feature. Secondly, another great feature would be the ability for riders to compare the prices they see on *Ride with Me* to those on Grab, Comfortdelgro and Gojek. This would allow them to monitor how much they can save by car-pooling on *Ride with Me*. This feature is not part of the prototype.

It is also important to acknowledge the limitations of *Ride with Me*. As we propose the use of Google's Routes API, we appreciate that there is a reliance on third-party software which increases the risk for our business. If the Routes API malfunctions, our routing algorithm will be negatively affected.

References

[1] *Ride-hailing & taxi - Singapore: Statista market forecast*. Statista. (n.d.). Retrieved April 14, 2023, from https://www.statista.com/outlook/mmo/shared-mobility/shared-rides/ride-hailing-taxi/singapore

[2] *Ride-hailing apps: Who's winning the Race Across Southeast Asia?* Blackbox Corp. (2022, May 12). Retrieved February 26, 2023, from https://blackbox.com.sg/everyone/ride-hailing-apps-whos-winning-the-race-across-southeast-asia

[3] *A quick look at Karos Car-pooling's business model.* Jugnoo. (2022, June 6). Retrieved March 9, 2023, from https://jugnoo.io/a-quick-look-at-karos-car-poolings-business-model/

[4] *Uber business model canvas*. Dribbble. (n.d.). Retrieved March 12, 2023, from https://dribbble.com/shots/8412551-Uber-Business-Model-Canvas

[5] *Top 3 carpool apps for Singaporeans - cash mart singapore*. Cash Mart. (2022, May 31). Retrieved April 14, 2023, from https://cashmart.sg/top-carpool-apps-for-singaporeans/

[6] Stewart, M. *Ryde's CEO on overcoming adversity and the value of hard work*. CNA Luxury. (2021, September 1). Retrieved March 9, 2023, from https://cnaluxury.channelnewsasia.com/people/ryde-ceo-terence-zou-189671

[7] Google. (n.d.). Google. Retrieved March 12, 2023, from https://developers.google.com/maps/documentation/routes/overview

[8] Mercier, A. (2015). *Amplify*. Amazon. Retrieved March 12, 2023, from https://aws.amazon.com/amplify/

[9] Hendrix, R. W. (1983). *Lambda*. Amazon. Retrieved March 12, 2023, from https://aws.amazon.com/lambda/

[10] Rangel, D. (2015). *Amazon DynamoDB*. Amazon. Retrieved April 14, 2023, from https://aws.amazon.com/dynamodb/

[11] Treichler, R., & Hardmeier, C. (2005). *Amazon SNS*. Amazon. Retrieved April 14, 2023, from https://aws.amazon.com/sns/

[12] *Amazon Cloudwatch - Amazon Web Services (AWS)*. Amazon. (n.d.). Retrieved April 14, 2023, from https://aws.amazon.com/cloudwatch/