

**Success Story**

**Unthinkable enables a consortium of taxi service providers in Qatar to develop a taxi booking application for their 10,000+ taxis**



**Customer**

Consortium of taxi companies in Qatar

**Country**

Qatar

**Industry**

Travel & Transportation

**About The Client**

The client is one of the major taxi services providers in Qatar. It was formed exclusively for online cab operations in Qatar by a consortium of more than 180 Limousine operators in Qatar. Currently, 10,000+ taxis are associated with the consortium. Their unique operating model makes them affordable to common people yet at the same time beneficial for drivers as they charge a nominal commission only to maintain the system.

**Technology Stack**

		
		

## Business Situation

A majority of Qatar's Limousine businesses had a partnership with several multi-national ride-hailing app companies to provide the services. These businesses were facing challenges in this partnership due to fewer margins and high expenses in maintaining luxury vehicles like Limousine. In addition, the ride-hailing companies increased their commissions and lowered the fares, creating financial issues for Limousine companies and drivers. It started to impact the entire Limousine market adversely, forcing many to bring their business down.

As a solution to this problem, the Limousine businesses in Qatar formed a consortium. The consortium decided to develop their own ride-hailing mobile application that enables booking of taxis from the businesses registered with the consortium.

The consortium approached Unthinkable Solutions to transform their vision into a market fit product. They needed a technology partner who could proactively contribute to design and ideation of their application, rather than just provide development services. The consortium chose Unthinkable for our best cost/value balance, extensive expertise in application development as well as our innovation oriented approach. The requirement was to:

- ✔ Develop a taxi booking application with two variants, one for the taxi drivers and other for the passengers. The passengers could search for a nearby driver, based on their location and destination of their travel, send a ride request to the nearest available driver, book a ride, travel, and pay using the app. On the other hand, the drivers' app could let the drivers see ride demands and navigate, accept ride requests, and complete the requested rides.
- ✔ To develop an admin panel for the car owners where they could have a bird's eye view of their fleet as well as can see the stats up to a granular level such as total rides, total revenue, daily/weekly and monthly trips and transactions, number of canceled rides, total passengers and drivers registered, etc. The admin could also control various factors such as verifying drivers, managing demand surge, discounts & coupon codes, managing vehicle categories, etc.
- ✔ Develop a scalable and robust architecture for the application which could help them scale and add more travel businesses to the consortium.

## The Solution

The engagement started with the Unthinkable team setting up a strategic requirement gathering process in place. This was done through our unique Discover & Frame workshop, wherein we document app ideas, target market, persona, business model, competition, revenue streams, etc. This helps us get a clear understanding of what is the client's expectations and effectively develop a project roadmap. Once, all the aforementioned aspects were documented, our expert business analyst team analyzed the application requirements, created user stories, feature lists, process flow diagrams and clickable prototypes of the application. On the techno-commercial front, they suggested the most viable technology stack, identified third party tools and Integration and proposed a block level architecture of the application.

The user experience was one of the project's cornerstones. The UI/UX of the application was a result of an extensive research by our business analyst team to study the behavior and motivation factors of the defined user persona. We focused on creating intuitive usability for both the applications. The passenger app was designed to easily allow passengers to book a taxi with minimum clicks, make payments securely as well as easily locate the taxi in real time. Similarly, the driver's app was designed considering the ease of use for drivers. Emphasis was laid to enable drivers to quickly register their car, accept rides and easily locate the passenger.

- ✔ Envisaging the utter need of scalability, and robustness of the application, a cloud-based architecture was designed so that the applications are exquisitely responsive to increase in number of drivers or passengers. To maintain the cost-performance ratio, team Unthinkable selected Amazon Web Services (AWS) for building an elastic infrastructure, for its brilliant payas-you-go models which ensured that the client only pays for the resources utilized and don't have to invest hundreds of dollars upfront. The entire infrastructure was built on AWS, where the primary infrastructure of the app was distributed across approximately 10 microservices. Kubernetes was used for managing multiple microservices and with Elastic Kubernetes Service (EKS), the app was made to handle over a hundred booking requests per minute. Some of the key technology challenges that Unthinkable resolved were:
- ✔ For each microservice deployment, up to 10 pods were used. EKS and configured pods were used for each service, such as dashboard, account, trip services, etc.
- ✔ To store the container images, Amazon Elastic Container Registry (ECR) was used.
- ✔ For caching of non-changing data, such as details of driver and passenger, Redis Node of Memcache service was used.
- ✔ For SMS notifications the Amazon SNS service was utilized
- ✔ For storing static files such as profile images, documents (Driver and Passenger) and invoice .pdf file, etc., the Amazon S3 service was utilized
- ✔ Entire DNS mapping was being managed by Route53 services. Logging was handled by CloudWatch and configured alarm notification to keep track of resource utilization at the time of peak traffic surge
- ✔ For database, PostgreSQL was used and for CI/CD of microservices, Jenkins pipelines was utilized

Team Unthinkable developed the application that consisted of three different variants: a native Android & iOS application for drivers, a native Android and iOS application for the passengers and a web based application for business owners. The apps enable riders and drivers to connect with each other. The admin panel is a dashboard that enables the admin to manage major activities that are performed at the rider and driver end.

#### Highlights of the passenger app:

- ✔ The rider app enables the users to schedule a ride, book a ride for self/others, add multiple drops between the pickup & drop location, check the driver's details as he accepts the ride (driver name, number, vehicle number, rating, profile picture, etc).
- ✔ When a customer books a ride, the driver will be selected based on their availability in the minimum defined radius. If more than one driver is found in that radius, then the selection is done on the basis of rating and number of rides served. If no rider is found within a defined radius, then a notification is passed to the riders to search for a ride in a bigger radius.

- ✓ Riders can check their past rides with details such as date, time, location, fare, etc. They can also check individual trip details with info such as driver profile, driver name, car type, etc., along with a text box to share the experience.

#### Few highlights of the driver app:

- ✓ The driver can sign up and can register himself with details such as name, photo, license, RC, etc., which is further approved by the admin. The driver can view the available rides and accept them.

The drivers can go offline or online to show their availability to pick a ride. When

- ✓ the driver is online, he can receive ride requests. If another driver with the same vehicle is online, the driver gets an alert that he can't go online.

- ✓ A driver has only one minute to accept a ride. If he doesn't respond, the request is passed to another driver. On accepting a ride, the driver can see the rider's name, the distance between driver and rider, and a button to contact the rider, and directions to navigate to the rider. As the trip begins, the driver gets updated about the ETA.

- ✓ Drivers can check the trip details and history. The trip history shows the list of all past rides and info such as date & time of the trip, pick & drop location, total fare, and driver's commission. The drivers can check details for a specific trip such as date & time, pick & drop location, trip number, ratings, and comments given by a rider.

- ✓ The drivers can check areas with high demand on a map with gradient colors. He is informed about new ride requests, canceled requests, and reached location through push notifications.

#### Few highlights of the admin panel

- ✓ The admin dashboard has information such as total number of vehicles, number of live vehicles, the current-day status of the number of vehicles, trips, earning, commission

The admin can define parameters for calculating an estimated cost of a trip on

- ✓ the basis of factors such as base fare, market demand, total distance, tariff, ride time, vehicle type, etc.

- ✓ For different types of vehicles, the admin manages the tariff on the basis of parameters such as minimum fare with KM limit, fare per additional KM, waiting time, the number of stops, ride cancellation time, set price on the basis surge based on-demand factor, etc.





## The Impact

The client was able to launch and market the product in a timely manner as Unthinkable ensured that the deliverables were provided on time, while maintaining utmost quality. Within a few months of its launch, the application has received an amazing response from businesses associated with the consortium as well as the passengers. By utilizing Amazon Web Services, team Unthinkable was able to achieve a number of benefits such reduced operational costs by 40% using Amazon EKS, easy prediction of daily and weekly spikes in usage, reduced service downtime with self-healing container ecosystem etc. More than 77% of developers' effort was reduced in launching/updating the application through microservices.

**Is there a digital platform you want to build or take to the next level?**

Setup a personalized consultation with our technology expert.

**Let's Talk**