

Innovative Features in Uber Clone Applications: The Extra Storage Solution

¹Gulshan Verma, ²Aryan Verma, ³Arpan Mukherjee, ⁴Nisha Rathore

^{1,2,3}Student (BCA 6th), ⁴Assistant Professor

^{1,2,3,4}Département of Computer Science and Engineering, Amity University Chhattisgarh

¹vgulshan419@gmail.com, ²aryanver207@gmail.com,

³arpanmmukherjee7125@gmail.com, ⁴nrathore@rpr.amity.edu

Abstract

Ride-hailing apps have greatly improved urban mobility by offering convenient and effective transportation options. This research paper investigates a groundbreaking improvement to an Uber-like application that incorporates additional language assistance and enhanced storage capabilities, addressing varied user requirements in urban regions. The additional language capability guarantees smooth interaction between drivers and passengers from different language backgrounds, fostering inclusivity and accessibility. In the meantime, the additional storage feature provides a specific area beneath unused seats, enabling passengers to easily carry small items like groceries, office documents, or packages. This feature functions on a rental pricing model, allowing drivers to bill users according to the length of storage. The article addresses the technical framework, enhancements in user experience, and possible economic advantages of these improvements. By incorporating multilingual features and extra storage, the Uber clone app enhances service efficiency while broadening market access, making ride-sharing more suitable for urban logistics and varied user demographics.

Keywords: - Ride-hailing innovation, Uber clone application, Multilingual support, Extra storage feature, AI-driven vehicle matching, IoT-based luggage management, Scalable transportation technology

I. INTRODUCTION

Finding a ride through the mobile application has made our busy lives considerably more convenient and efficient. The clone script for young entrepreneurs is designed with a plethora of features that enable a better reach out than its original prototype. These innovative applications are carefully aimed with most of the crucial features necessary for optimal performance. So, why not take this opportunity to start brainstorming an idea that could slice into the existing features that aren't fully explored or propounded with the prototype?

For instance, how does extra storage work in this advanced clone application? Here in this paper we are providing how we can manage extra storage and provide comfortability to person.

The application in consideration is about an idea of adding extra storage in the clone app that would be very much beneficial for both the driver and the rider especially in the metropolitans popping up with people.

This new feature provides extra storage, which is an empty space attached beneath the seats, this storage will help in storing vegetables, edibles, small packages, office file, etc., in the place where it is hard to find an empty spot in the commercial city. To keep safety in mind the storage will not be attached to the driver seat neither it will be placed in the seats occupied by the passenger. Customers in need to avail the extra storage can request the feature. The rider gets notified about the extra storage. By renting the storage, driver charges according to 1 hour, 2 hours or 4 hours. An availability feature is an available spot that is free or free up after the set time. Thus, the new innovative feature is the “Extra Storage” solution enhancing the clone application. Now a days uber clone application have become increasingly popular due to their potential to replicate proven business model and adapt them to new market [1]

II. Literature Review

Uber Technologies Inc. stands out as one of the most groundbreaking firms in the transportation sector, changing how individuals arrange and enjoy their rides. Launched in March 2009 by Garrett Camp and Travis Kalanick, Uber began with a straightforward concept—to offer a convenient, affordable substitute for traditional taxis through smartphone technology. Since its debut in San Francisco in 2010, Uber has grown internationally, providing services in more than 70 countries and 10,000 cities around the globe

Origin and Early Growth

The concept for Uber originated when Garrett Camp, one of the co-founders of StumbleUpon, had difficulty hailing a taxi in San Francisco. He imagined an application that would link passengers to nearby drivers through GPS technology. The firm's original name was Uber Cab, and its inaugural service featured black luxury vehicles aimed at high-end clientele. Nonetheless, following regulatory pressures from taxi firms, Uber removed "Cab" from its name in 2011 and rebranded itself as Uber.

Uber's rise in popularity surged quickly because of its convenience, cost-effectiveness, and user-friendliness. The pricing model of the company, featuring dynamic (surge) pricing according to demand, enabled greater flexibility in ride availability

Expansion and Innovations

In 2012, Uber launched UberX, an affordable ride service that permitted ordinary drivers to utilize their personal cars, enhancing Uber's availability to the wider audience. This action placed Uber in direct rivalry with taxis and ride-sharing companies such as Lyft. Throughout the years, Uber broadened its offerings beyond conventional ride-hailing:

Uber-POOL (2014): Launched ride-sharing to lower expenses and alleviate traffic.

Uber-Eats (2014): A meal delivery service enabling users to order food from affiliated restaurants.

Uber-Freight (2017): A logistics service connecting trucking firms with companies requiring cargo delivery. Now a days the rise of cab services like Uber and its clone has significantly changed urban transportation dynamic in India [2]

III. Proposed Model

The addition of the Extra Baggage Feature in Uber clone models fills a notable void in ride-hailing services by allowing passengers to choose vehicles according to their luggage requirements. Conventional ride-sharing services frequently neglect baggage limitations, causing difficulties for travelers, especially those commuting to airports and business executives. By introducing an AI-based vehicle selection system focused on luggage, this model enhances user experience, increases ride efficiency, and maximizes driver usage.

The approach utilizes React.js for the frontend,

Node.js combined with Express.js for the backend, and MongoDB for data handling, guaranteeing a scalable and effective architecture. This advancement enhances the customer contentment and loyalty while also providing a competitive advantage in the ride-hailing industry. This model offers a sustainable revenue chance by improving ride-hailing convenience and has the potential for scalability in airports and travel hubs.

Overview of Uber Clone Models with Extra Baggage Feature

The Uber replica model has revolutionized contemporary transport by providing tailored ride-hailing options for various user requirements. Although conventional ride-sharing platforms effectively link riders to drivers, they frequently neglect an important factor—space for passenger luggage. Numerous travelers, particularly airport commuters, business people, and families, have difficulty locating vehicles that can accommodate their luggage adequately. The updated "Extra Baggage Feature" in the Uber clone model seeks to address this concern by pairing passengers with vehicles suited to their luggage needs, providing a smooth and effective ride experience

Existing Uber Clone Models

Numerous Uber clone apps mimic the essential features of Uber while incorporating distinct adjustments. The most prevalent models consist of:

Fundamental Ride-Hailing System: Links passengers to local drivers for regular urban trips.

Carpooling System: Enables several riders to share a trip, lowering travel expenses and traffic congestion.

Luxury Drive Model: Provides top-tier vehicles for elite clientele.

Bike Taxi Model: Offers fast, affordable rides on two wheels.

Rental & Outstation Model: Allows for long-range reservations with rental choices

Introducing the Extra Baggage Feature

The revamped "Extra Baggage Feature" is a sophisticated AI-based ride distribution system that improves user satisfaction by:

Luggage-Driven Vehicle Choice: Passengers indicate the quantity and dimensions of their luggage when arranging a ride. The application recommends appropriate vehicles considering the space for luggage.

Booking by Category

Compact Luggage Rides: These rides are most appropriate for travelers with light and minimal luggage like small backpacks, laptop cases, or handbags. It is perfect for those traveling short distances, daily commuters, or students who don't need much luggage capacity.

Medium Luggage Transport: This choice caters to travelers with a fair amount of luggage, generally consisting of 2 to 3 medium-sized bags or suitcases. It is perfect for business travelers or small families taking short trips, providing ample space without requiring a larger vehicle.

Large Luggage Transport: Large baggage transport options are designed for travelers with multiple or oversized luggage. SUVs or vans are designated to hold luggage such as strollers, travel trunks, or several suitcases, providing convenience for airport transfers, lengthy journeys, or family holidays.

Luggage Pricing Structure: The ticket price includes an additional charge depending on the number and size of bags transported. This system guarantees fairness, as travelers only pay for the storage area they truly utilize. It enables improved cost allocation among users with varying storage requirements.

Baggage Capacity Confirmation: Drivers must consistently refresh the amount of available luggage space in their cars using the app. This

real-time update assists the system in preventing booking discrepancies and guarantees that the passenger receives a vehicle that accurately meets their luggage needs.

Luggage Support Service: To enhance convenience, passengers may ask the driver for help with their luggage. This service is particularly beneficial for senior travelers, individuals with bulky or cumbersome luggage, and those requiring assistance with loading or unloading their belongings at pick-up and drop-off locations.

Benefits of the Extra Baggage Feature

Traveler Convenience: This functionality to removes the late-hour hassle and trouble of attempting to fit luggage into a car that doesn't have enough room. Choosing the right ride type beforehand allows travelers to enjoy a more seamless, stress-free experience, especially during time-critical trips such as airport transfers or business travel.

Enhanced Ride Distribution: The system smartly pairs passengers with vehicles by considering seating and luggage capacity, guaranteeing the best vehicle assignment. This results in increased passenger contentment, fewer cancellations, and a more effective utilization of fleet assets.

Improved Driver Utilization: Operators of a larger vehicles, once potentially underused, are now paired more efficiently with riders needing extra room. This boosts their booking rates and income while decreasing downtime, enhancing the platform's attractiveness for various drivers.

Competitive Advantage: Uber clone applications can set themselves apart from competitors by addressing a distinct pain point—sufficient luggage capacity. Providing this specific solution increases the service's value, draws in a niche audience (such as airport commuters, families, business travelers), and improves brand loyalty and customer retention. [3]

IV. Methodology of Uber Clone Models with Extra Baggage Feature

The Uber replica model featuring the Extra Baggage Function is created to effectively pair riders and drivers according to luggage needs, improving user ease and ride personalization. The architecture of the system adopts a systematic and expandable method, utilizing contemporary web technologies to guarantee excellent performance and responsiveness. The frontend utilizes React.js, providing a dynamic and intuitive interface for effortless interaction. The backend utilizes Node.js with Express.js, allowing quick server-side processes and management of ride requests, baggage sorting, and driver selection. Information is stored and handled using MongoDB, a versatile NoSQL database that enables scalability and effective querying of user and ride data. The complete application is developed and enhanced using Vite, a rapid build tool that speeds up development and boosts loading durations. This technology framework guarantees a dependable, instantaneous ride-hailing service with additional assistance for handling extra luggage.

Steps for Implementing Features

a) User Ride Request Including Luggage Information:

Travelers input the quantity and dimensions of their luggage during the reservation process. The system classifies the request as Small, Medium, or Large Luggage Rides.

b) Algorithm for Matching Vehicles Using AI:

The backend fetches available vehicles that meet the luggage capacity criteria from MongoDB. Node.js manages ride assignments by utilizing the current availability of drivers. Drivers continuously update the availability of luggage space.

c) Pricing and Verification:

The pricing for extra baggage is determined and displayed before the ride is confirmed. Payments are incorporated through external gateways.

d) Tracking and Notifications in Real Time:

React.js provides a seamless user interface with real-time ride monitoring. WebSocket (through Socket.io) allows for immediate notifications.

e) Completion of Ride and Feedback:

Travelers assess the quality of the service and baggage handling experience. MongoDB saves reviews for upcoming enhancements

Studies indicate that a well-designed driver- side interface is crucial for the efficiency of the services [4]



Figure 1: Mobile App Development Process for Uber Clone with Enhanced Baggage Acc

V. Implementation of extra luggage feature

```
import React from 'react';
import { FaMotorcycle, FaCar, FaTruck } from 'react-
```

```
icons/fa';
import { GiElectric } from 'react-icons/gi';
const VehiclePanel = ({ selectVehicle, fare,
setConfirmRidePanel, setVehiclePanel,
setLuggagePanel }) => {
const handleVehicleSelect = (vehicleType) =>
{
selectVehicle(vehicleType);
setLuggagePanel(true); // Show the
luggage panel when a vehicle is selected
};
```

```
return(
<div className='p-4'>
<h4 className='text-xl font-semibold mb-4'>Select
Vehicle Type</h4>
```

```
<div className='flex flex-col space-y-6'>
```

```
{/* Two-Wheelers */}
```

```
<div>
```

```
<h5 className='text-lg font- semibold mb-
2'>Two-Wheelers</h5>
```

```
<div className='flex space-x-4'>
```

```
<button
```

```
className='p-2 border border- gray-300 flex items-
center space-x-2 rounded- lg hover:border-black'
```

```
onClick={()=>
handleVehicleSelect('electric-scooter')}>
```

```
<span>Electric Scooter</span>
```

```
</button>
```

```
<button
```

```
className='p-2 border border- gray-300 flex items-
center space-x-2 rounded- lg hover:border-black'
```

```
onClick={()=>
```

```
handleVehicleSelect('petrol-scooter')} >
```

```
<FaMotorcycle />
```

```
<span>Petrol Scooter</span>
```

```
</button>
```

```
<button
```

```
className='p-2 border border- gray-300 flex items-
center space-x-2 rounded- lg hover:border-black'
```

```
onClick={()=> handleVehicleSelect('bike')} >
```

```
<FaMotorcycle />
```

```
<span>Bike</span>
```

```

</button>
</div>
</div>
{ /* Cars */ }
<div>
<h5 className='text-lg font- semibold mb-2'>Cars</h5>
<div className='flex space-x-4'>
<button
className='p-2 border border- gray-300 flex items-center space-x-2 rounded- lg hover:border-black'
onClick={() => handleVehicleSelect('hatchback')} >
<FaCar />
<span>Hatchback</span>
</button>
<button
className='p-2 border border- gray-300 flex items-center space-x-2 rounded- lg hover:border-black'
onClick={() => handleVehicleSelect('sedan')} >
<FaCar />
<span>Sedan</span>
</button>
<button
className='p-2 border border- gray-300 flex items-center space-x-2 rounded- lg hover:border-black'
onClick={() => handleVehicleSelect('suv')} >
<FaCar />
<span>SUV</span>
</button>
</div>
</div>
{ /* Autos */ }
<div>
<h5 className='text-lg font- semibold mb-2'>Autos</h5>
<div className='flex space-x-4'>

```

```

<button
className='p-2 border border- gray-300 flex items-center space-x-2 rounded- lg hover:border-black'
onClick={() => handleVehicleSelect('auto')} >
<FaTruck />
<span>Auto</span>
</button>
<button
className='p-2 border border- gray-300 flex items-center space-x-2 rounded- lg hover:border-black'
onClick={() => handleVehicleSelect('electric-auto')} >
<GiElectric />
<span>Electric Auto</span>
</button>
</div>
</div>
<button
onClick={() => setConfirmRidePanel(true)}
className='bg-black text-white px-4 py-2 rounded-lg mt-6 w-full' >
Confirm Ride
</button>
</div> ); };
export default VehiclePanel;

```

Explanation of code

This React component, Vehicle Panel, creates a user interface for choosing a vehicle type for a ride-hailing or booking service. It classifies vehicles into three categories: Two-Wheelers, Cars, and Autos, with each featuring buttons that include icons depicting various vehicle choices such as electric scooters, bicycles, hatchbacks, sedans, SUVs, and autos. When a user presses a button, it activates the handle Vehicle Select function, which defines the chosen vehicle type and shows a luggage selection panel using set Luggage Panel(true).

A "Confirm Ride" button at the bottom enables users to continue by displaying the confirmation panel through set Confirm Ride Panel(true). The element utilizes React icons for graphics and Tailwind CSS for design.

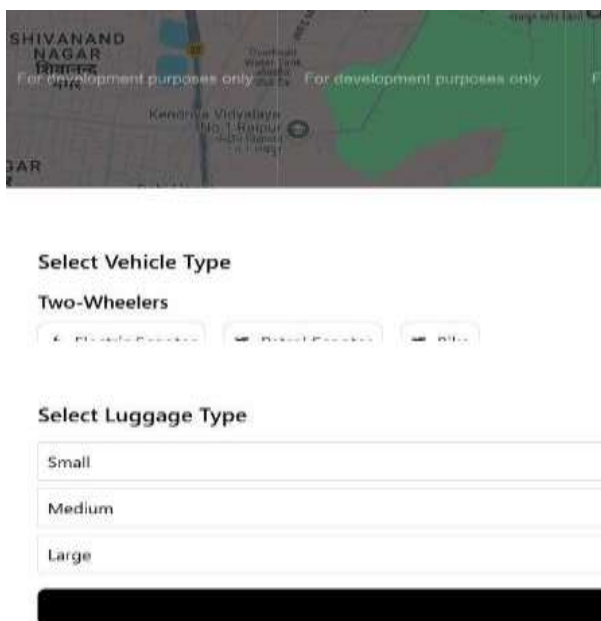


Figure 2: Luggage Selection Interface

VI. Analysis

Market Analysis of Ride-Sharing Services with Extra Baggage Feature

The worldwide ride-sharing industry is expected to expand notably, fueled by urban development, rising smartphone usage, and the need for affordable transportation. The market, led by companies such as Uber, Lyft, and local rivals, is changing with fresh innovations to tackle user challenges. A significant problem encountered by passengers, particularly travelers, airport commuters, and families, is the absence of luggage space in typical ride- hailing services. Now a days research highlights how user preferences and technological trends affect the growth of ride-sharing app. [5]

Market Demand for Extra Baggage Feature

Increasing Urban Mobility Requirements: Major

cities face significant demand for ride- hailing options, as passengers frequently need adequate room for their luggage.

Airport and Long-Distance Passengers: Travelers heading to and from airports face challenges in locating suitable taxis that can accommodate their luggage effectively.

Business and Shopping Travelers: Professionals with presentation items or shoppers making significant purchases gain from customized vehicle choices

Competitive Advantage of the Feature

Conventional ride-hailing services pair passengers according to seating availability, overlooking baggage limitations. If we classifying rides as Small, Medium, and Large Luggage Rides, this advancement improves user contentment and service effectiveness.

Market Opportunity: Revenue Expansion: High pricing for additional luggage allowance can boost driver income and platform profits.

Customer Loyalty: Providing personalized ride choices boosts customer loyalty and sets the business apart from competitors.

Scalability: The capability can be extended to airports, hotels, and travel centers, creating new business possibilities. Now a days the inclusion of real-time GPS Tracking has enhanced user confidence and transparency in ride-sharing platforms

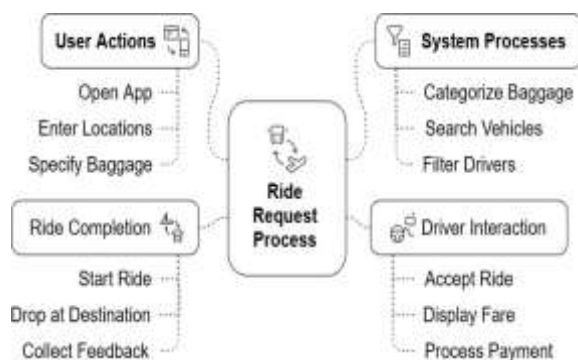


Figure 3: Journey of a Ride Request in an Uber Clone Model with Extra Baggage Feature

Explanation

The flowchart illustrates the "Ride Request Process" in a ride-hailing application, segmented into four main elements: User Actions, System Processes, Driver Interaction, and Ride Completion. It begins with User Actions, as the user launches the app, inputs pickup and drop-off points, and provides baggage information. These inputs trigger System Processes that classify luggage, look for available vehicles, and sort drivers according to the ride specifications. The subsequent phase entails Driver Interaction, during which drivers get the ride request, assess the fare, and choose whether to accept the ride. Once accepted, the system shows the fare and gets ready for payment processing. The last phase is Ride Completion, during which the ride starts, the user arrives at the destination, and feedback is gathered. This process guarantees a seamless and organized riding experience by combining user feedback, automated backend functions, and real-time driver interaction, leading to effective and user-friendly ride oversight.

VII. Conclusion

The Extra Baggage Feature solves a major gap in ride-hailing by matching riders with vehicles based on luggage needs—ideal for airport, business, and shopping trips. Using AI and built with React, Node.js, and MongoDB, it offers real-time baggage alerts, tiered options, and dynamic pricing. Drivers benefit from better ride distribution and extra earnings. Scalable to delivery services, the model promises a smarter, more inclusive, and future-ready ride-hailing experience.

I. Future Scope

The Extra Baggage Feature can reshape ride-hailing by meeting growing luggage transport needs. Key expansion areas include:

Travel Partnerships: Pre-book luggage-friendly rides via airlines or hotels.

Self-Driving Cars: AI taxis with adjustable luggage space.

IoT Monitoring: Real-time trunk space tracking with smart sensors.

Parcel Delivery: Extend to package and luggage transport services.

Enhanced Security: Add smart locks, baggage screening, and dynamic pricing.

References

- [1] Ghosh, S., & Biswas, S. (2020). Design and Development of Online Cab Booking System. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(4), 1526-1529.
- [2] Kumar, V., & Pundir, D. (2020). Impact of Online Cab Services on Urban Transportation- A Study on Uber and Ola in India. *International Journal of Management (IJM)*, 11(4), 140-151.

[3] Nayak, N., & Sahoo, S. (2019). Study of Online Cab Booking Applications. International Journal of Recent Technology and Engineering (IJRTE), 8(2), 3472-3476.

[4] Jalil, M. M. A., Islam, M. J., Kabir, M. N., & Hasan, M. (2021). Ride Sharing System with Real Time Location Tracking. International Journal of Engineering Research & Technology (IJERT), 10(4), 409-412.

[5] Muthukumar, S., & Sathya, R. (2020). A Study on Mobile Application for Cab Services.