Long Distance Carpooling Application for Hokies

By: Michael Richter, Charlie Nguyen, Paul Blackburn, and Angela Mattamana

Big Idea

Cheaper than Uber (Uber not for long distance)

Cheaper than Airfare

More credible than Facebook Pages

More accessible than traditional rideshare boards

Freshman don’t have to be picked up by their parents

Roles

Michael Richter: Front end development
Charlie Nguyen: Mobile application
Paul Blackburn: Full stack, web application
Angela Mattamana: Backend development

Driver Functional Requirements (1 - 4)

Analyze driver speed and reduce their payout if they speed.

Driver score (feedback provided by riders)

Submit the trip details (start time, start location, and final destination) and the number of car seats open

Must be able to accept and decline riders based on rider profile

Rider Functional Requirements (5 - 8)

Rider score (feedback provided by driver) and profile (picture)

Must be able to purchase trunk allocation (or additional seat) for luggage.

Search for rides based on proximity to desired final destination

Submit requests for detours (stop along the way, etc.)

Sign Up Functional Requirements (9 - 12)

Riders sign up for app with Hokie Passport

Drivers must sign up with valid license and insurance.

Prevent drivers with points on their license from signing up

After graduation, must not be able to use app

Payment Functional Requirements (13 - 16)

Supports Hokie Passport, Venmo, and Zelle as valid peer to peer payment methods.

Multiple payment methods for one account (must be encrypted)

The driver receives payment via a payment method from each of the riders upon completion of the ride.

Validate payment methods to ensure their is sufficient funds.

Non Functional Requirements

Screen Adaptation for multiple different devices

Mobile Application and Web Application to be accessible to all users.

Availability on a common plane to access our application (Apple’s App Store, Google's Play Store)

Reliability of requesting rides and sending information (Transaction Summary, email confirmation)

Ensure Drivers and Riders don’t flake at the last minute

Algorithm Design

Pricing algorithm which determines a fair price based on historical data

Surge algorithms

Trip score to determine if a potential carpool is a good deal

Questions

Push all liability on users

Don’t profit, simply facilitate carpools (drivers are not independent contractors)

Emphasize peer to peer payments methods