Team Number: 4
Team Name: Alpine, Ltd.
Team Members: Allen Zhang, Caleb Noel, Daniel Lisle, Erica Shivers, Justin Lee
Application name: Alpine
Application Description:

Ride-sharing website that connects users with identical destinations in the Colorado area. The website allows users to browse information on various hiking landmarks and ski areas. Information for hiking spots may include the length of the hike, terrain, level of difficulty, and altitude. Information for ski areas may include snow depth, snow type (packed powder, powder, etc.), and ticket costs.

Users can put in their chosen destination with a planned date of travel and the website will match with other travelers. Drivers and passengers can create groups. Users can chat with their group to plan the trip. To facilitate cost-sharing, the website calculates gas cost and share-per-passenger given vehicle information.

Vision Statement: "*Have more fun in the mountain by meeting new friends in a car ride"*

Version Control: github.com/CSCI-3308-CU-Boulder/3308Summer21_300_4

Development Method:
We will use the Scrum methodology. We will adopt the user story, backlog, and sprint paradigms, but have no need for a Scrum Master or Product Owner. As sprint team members, we will share maintenance and prioritization of the backlogs.
Jira Board: https://csci-3308-summer1-4.atlassian.net/jira/software/projects/AL4/boards/1

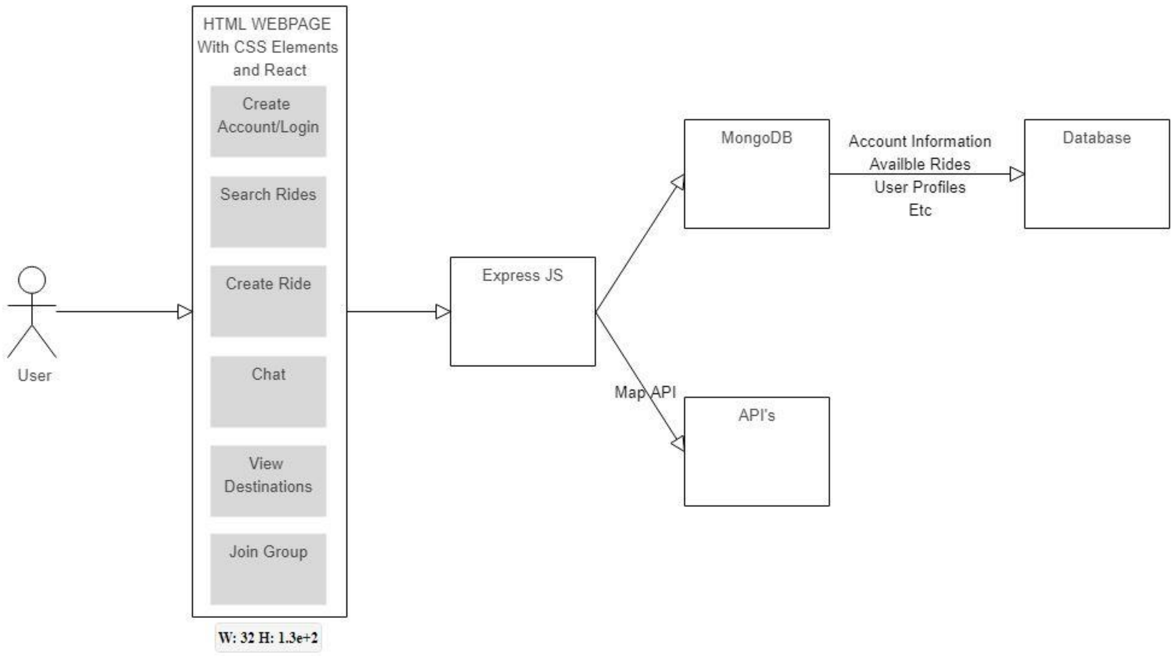
Communication Plan: We will primarily use Discord for communication. Using separate channels in our server, we will organize correspondence for different stages and branches of development. We will communicate mostly in text channels, with voice calls for major Agile events.

Meeting Plan:
- Voice calls for sprint planning sessions, reviews, and retrospectives
- Text communication during sprints
- Team meetings Wednesday 9am-11am on Discord
- TA meetings Monday 9:00am-9:20am https://cuboulder.zoom.us/j/92297973059

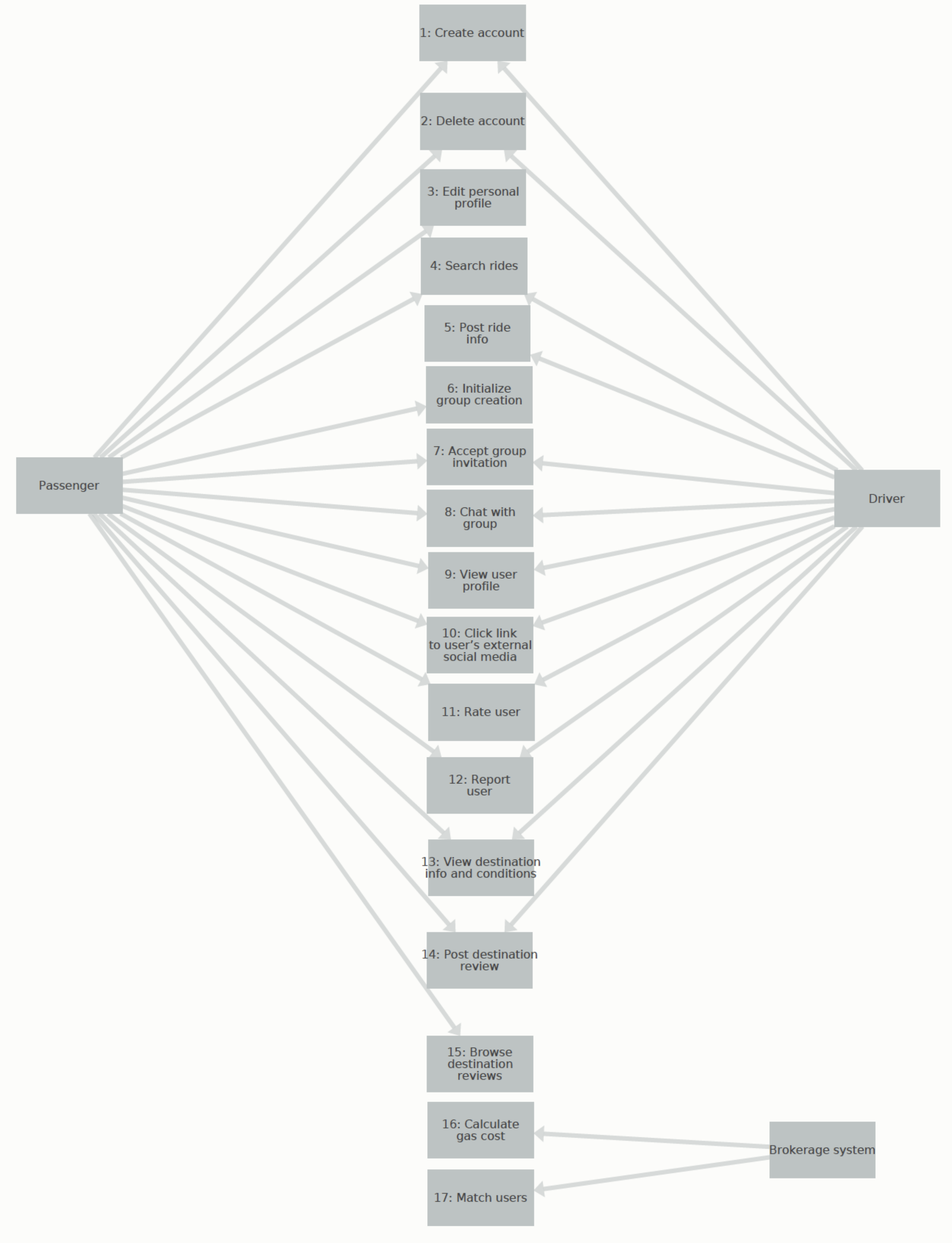
Proposed Architecture Plan:

- The technologies to be used on the back-end include Express.js, and MongoDB which gets rid of using overly complicated SQL queries.
- The technologies to be used on the front-end include react, materialize-CSS, and HTML.
- The front-end and back-end will communicate with each other using the handlebars API of Express.js.
- React will be responsible for the behavioral and state storage part of the website, materialize-CSS for styling, MongoDB as the database, and Express.js as an API for communication between back-end and front-end.

Architecture Diagram



Use case diagram


1: Create account
2: Delete account
3: Edit personal profile
4: Search rides
5: Post ride info
6: Initialize group creation
7: Accept group invitation
8: Chat with group
9: View user profile
10: Click link to user’s external social media
11: Rate user
12: Report user
13: View destination info and conditions
14: Post destination review
15: Browse destination reviews
16: Calculate gas cost
17: Match users
Passenger
Driver
Brokerage system