

Zoom Link: <https://cuboulder.zoom.us/j/91432440899>

**Team Number** - 01

**Team Name** - Pocket Mechanics

**Team Members:**

Name	GitHub Username	Email
Nanda Min-Fink	nandamin-fink	nami7214@colorado.edu
Eva Pavlik	eva-pavlik	evpa9944@colorado.edu
Rey Stone	rast4675	rast4675@colorado.edu
Sabir Saklayen	ssaklayen	sasa4287@colorado.edu
Sami-ul Ahmed	sami-ul	saah8460@colorado.edu

## **Application Name:** Pocket Mechanics

**Application Description:** We are creating a website with the overall purpose of providing an easy-to-use car-parts finder and shipper. First, the user can login to the website, and have 0, 1, or more cars (makes, models, year, etc.) saved to their profile. That way, they can have their cars' information saved and choose which car they are shopping for, and the website will ensure compatible matches with the user's search.

The website will compare compatible parts found on RockAuto.com and the part prices as listed on rock auto, amazon, and ebay, to help the user find the best deal for their car. Then, using Google Maps' API, the site will provide nearby stores or garages the given part can be shipped to if the user is interested in having the part installed for them. Otherwise, the user will be able to ship it to their house. The site will also provide links to tutorial videos on how to install the part themselves if the user would rather install the part themselves.

**Audience:** Anyone with a car! Anyone who needs to replace a part in their car and wants to find the best deal for the given part while ensuring compatibility. The website will be friendly for those who don't know a lot about cars/car parts.

**Vision Statement:** For car owners **who** need a unified, easy-to-use platform to find compatible auto parts at the best prices, **Pocket Mechanics** provides an all-in-one platform that compares prices, saves your vehicle profiles, and even connects you with mechanics to install your parts. **Unlike** the traditional way of going to a mechanic shop and having them find parts, or finding your own parts and later finding out that they are incompatible with your vehicle, **our product** cuts out the middleman and empowers you to make informed, cost-effective decisions.

**Version Control:** <https://github.com/Sami-ul/auto-part-finder>

We will use feature based branching (see Github)

**Development Methodology:** We will use a simplified version of the Agile methodology. The 5 steps for an Agile workflow include: defining project goals, breaking down tasks and milestones, assigning tasks and responsibilities, creating the timeline and schedule, and monitoring progress/adjusting as necessary.

- Defining project goals
  - In our next meeting, we will discuss what major goals we need to accomplish to complete our project. Since we have 4-weeks to produce our application, this equates to about 4 sprints. Our main goal is to produce an application that matches our team's vision which is listed under our vision statement in this document. Subgoals will be listed in the next bullet point as we break down our main goal.
- Break down tasks and milestones
  - We will confer together as a team to decide what are the major tasks and milestones necessary to achieve our main goal. The following is a quick list of tasks that we will need to accomplish (This list is not comprehensive nor in order yet):
    - Wireframing/Use-Case diagram: We will use these processes to imagine what our application will look like, frame-by-frame, interaction-by-interaction.
    - Handlebars Templating: From the wireframe designs and the use-case diagram we will need to build a template for our application including our main layout, partials for nav bar, header and footer as well as whatever partials will be required as per our wireframe design
    - Theme: We need to choose some consistencies in theme, like colors, layout, icons, etc. These can be applied via CSS or other styling methods. These first three tasks are very related.
    - Auto-Parts Lookup API: We need some type of API to be able to locate a vehicle part given a set of parameters for a vehicle. In our last meeting, it was suggested that we look into building our own custom API via web scraping.
    - Part Locator/Installation on Map API: Our intent is to add the functionality to find physical locations close to the end-user that retails the requested auto-part. Additionally, if the end-user opts for professional installation, this same API can be used to find an installation shop/auto mechanics garage close to the end-user. There are a couple free map locator APIs, we need to check and see how Google Map's API works and if it will allow us to accomplish this task(it should I think, but may have a paywall. If there is a paywall, we could perhaps do something similar to the Auto-Parts Lookup API by using Google Maps, scraping the results and presenting those results tailored to our application's needs.
    - Reviews API (May not need this): This API will be used to collect reviews about parts/installation shops. Similarly we will need to check for a pre-existing API that will aid us, or we can web scrape Yelp/Amazon/Rock Auto Parts etc..
    - User Management System: This task can be broken down further as it will contain a few components -

- Hashing algorithm/function for password security (Could consider OAUTH but might be overkill)
  - Database where one entry contains a username, hashed password and is tied to a related table that contains that user's profile (Name, shipping address, phone number, email, contact method) and then subtables related to each individual profile that contains each vehicle profile for each of the user's vehicles. Vehicle profiles will likely also be broken into related subtables. This is one idea design. When we meet next, we will be able to discuss the best approach to accomplish this task
  - Registration system to add a new user and their vehicles. This actually will probably fall more under Backend Routing rather than right here, but we will need to ensure the individual(s) working on those tasks collaborate together to stay in sync.
- Backend Routing: This task will involve writing several different routes with many different actions taken on each route depending on what the service of that route may be. This will tie together the User Management System, API use and collection of results and will render the appropriate Handlebars template accordingly. This task will involve synchronization with the Frontend pairs and Pairs who develop the APIs and User Management system.
    - Initial subtask for this will include creating .env files for security. docker-compose.yaml (we will use this again in web deployment), SQL init files, and index.js skeleton including all necessary dependencies
  - Extra Features/Experiments: Time-dependent, if there are additional features we want to add or experiment with and we are ahead of schedule, we can do that here.
    - Feature to take photo of car to autofill vehicle profile
  - Local Testing/Debugging: We will likely be debugging and testing individual tasks/components on our own as we develop each piece. This task is after completing our prototype and before web deployment.
  - Web Deployment: This task will involve finding a host service that will allow us to host our application. The service will need to allow us to use containers, postgresql and nodejs as per our tech stack. This task will involve writing whatever configuration files that will be necessary to link our frontend with our backend, uploading our application with correct directory structure and anything else that will be required (It's possible we could do all of this thru GitHub pages which would simplify this task incredibly)
  - Presentation: This task can also be broken down into pieces:
    - Images/video of application usage & expected results (maybe development too? Like wireframe screenshots, notes, code screenshots, meeting summaries etc.
    - Audio/Video walkthrough of our application
    - Audio/Video explanation of issues we had along the way

- Audio/Video explanation of things we think we could improve/change given more time/resources
  - Milestones will be chosen at the next team meeting. We have 4 weeks to complete this project, suggested path for milestones would include Wireframing/Theme first week, Frontend Templating, API development User Management system no later than beginning of third week. Backend routing between third and fourth week. Debugging and deployment beginning of fourth week, Presentation the rest of the 4th week.
- Assign Tasks and Responsibilities:
  - We will meet as a team to arrange these. It will involve using our task list, pairing related tasks and then assigning by team member strengths/preferences/available bandwidth
- Create Timeline and Schedule:
  - This falls in line with the milestones and task delegation. We have determined our hard deadline for completion is April 16th, 2025. As a team, when we meet next, we will develop more streamlined and specific deadlines per task.
- Monitor Progress and adjust as necessary
  - As this step implies, we will work together as a team to monitor each other unless the team wants to assign someone to oversee that progress is continuing forth as scheduled. We have a weekly check-in with our TA on Wednesdays. Additionally, our team will commit to checking in bi-weekly via our communication channel to ensure each member is up-to-date with progress/ongoing tasks/changes and anything else deemed important for that week.

**Communication Plan:** We will use Discord as our primary channel of communication.

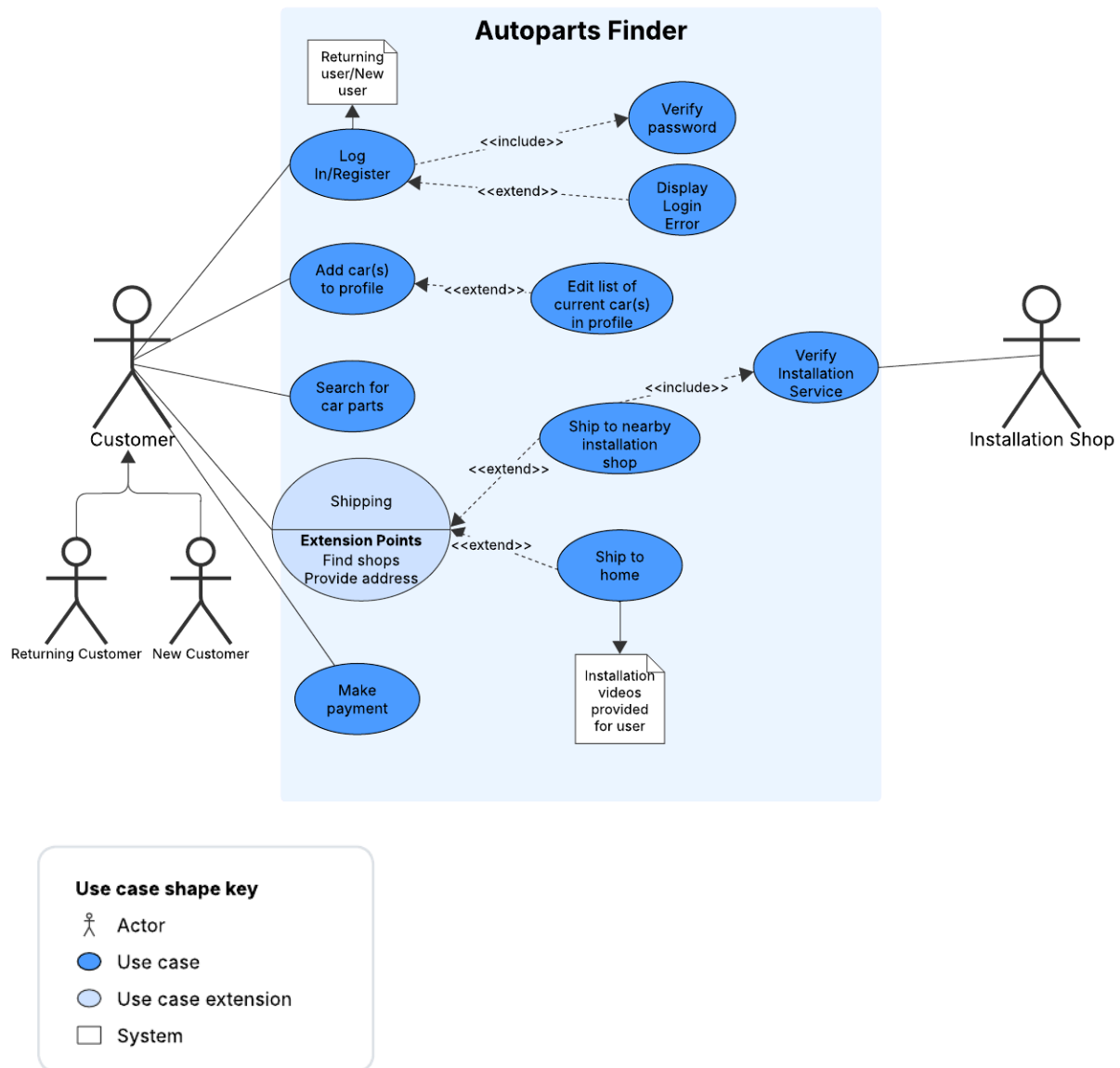
### Meeting Plan:

Team Meeting: Sunday at 8PM Weekly on Zoom

Weekly Meeting with TA: Wednesday @ 12:55 PM

### Use-Case Diagram:

[https://lucid.app/lucidchart/5b5fa990-a280-42e7-ab33-c7782a0e58c4/edit?viewport\\_loc=-137%2C10%2C1909%2C1039%2C.Q4MUjXso07N&invitationId=inv\\_82b06bc0-47eb-4f87-9f78-c1f272106b01](https://lucid.app/lucidchart/5b5fa990-a280-42e7-ab33-c7782a0e58c4/edit?viewport_loc=-137%2C10%2C1909%2C1039%2C.Q4MUjXso07N&invitationId=inv_82b06bc0-47eb-4f87-9f78-c1f272106b01)



## Wireframe:

<https://www.figma.com/design/s35s7Jo9h3vMiXDKKsWakU/Auto-Parts-Finder?node-id=0-1&p=f&t=ZWNScoW7FjJzb6eL-0>

**Goal:** All in one platform to find parts based on prices for any vehicle based on location. Make finding parts compatible with your car more simple.

Wireframe screenshots:

(See the poorly formatted pages below)

