TEARv - Terrain Exploration And Rescue Vehicle

Members: Rahul Balla, Dominic Miller, Xu He, Rishabh Ramsisaria, Shaurya Sinha

Product Backlog

Problem Statement

First responders and rescue operatives put their life at risk when inspecting areas struck by natural disasters. The manual process of looking for survivors and checking for damage puts human lives at risk. Our product can decrease the potential hazards in rescue operations by mechanizing the search and scout process. Compared to other products in the current market our product, TEARv, is a cost-efficient, maneuverable, customizable, remote controlled vehicle with a live-feed. Our vehicle will store and transmit live video to a mobile app. Moreover, the core of the car, being a Raspberry Pi, will be able to connect to wifi and receive signals and instructions from the app itself.

Background Information

Audience

Our primary audience for this project consists of organizations which focus on post-disaster excavation and rescue operations to assess damages and further risks. Secondary or other potential users for our product include wildlife enthusiasts and photographers who can use it to silently observe animals in their natural habitats. Security agencies can also conduct surveillance, and children can use it for learning about robotics and programming as well.

Similar Products and Limitations

From our research, a specific commercial product targeted towards post-disaster management doesn't exist in the market today. The closest products available are basic RC cars with limited functionalities which have little to no room for customization or improvement. Some of the basic RC cars with a camera available in the market include the 1/18 All-Terrain FPV Real-Time Video Truck w/ Built-In Camera ($40 Amazon) and I_spy Mini Wireless Spy Tank RC Car with 0.3mp HD Camera ($39.95 Amazon). However, having a Raspberry Pi as the main processing unit allows us to install different sensors such as proximity sensors, temperature sensors and RH sensors which allow us to specialize and customize the car to achieve better functionality at a similar price point.

Another type of products with similar functionality are drones. Today, drones have a lot of applications and a huge market for surveillance and photography. Some of the major

manufacturers of drones are DJI, Parrot, and 3D Robotics. However, drones also have some situations where they are not as useful as a road vehicle. They are usually loud and cannot be flown in small or restricted spaces such as tunnels or indoors, whereas our potential product will be more quiet and maneuverable. Also, drones are not as sturdy, tend to get damaged or break easily and are also much more expensive with most models pricing upwards from $500.

Functional Requirements

Mobile Application

Registration

1. As a user, I would like to be able to register for a *TEARv* account using username, password, and email so that my information would remain personal to me
2. As a user, I would like to be able to recover both, my username and password, if I don't remember it so that I do not get locked out of my account using my registered email
3. As a user, I would like to have the option to change my username and password
4. As a user, I would like to be able to register multiple products under the same account to use for different purposes such as one for surveillance and another for testing

Video

5. As a user, I would like to be able to receive a live feed from the vehicle so that I can see where I am going and navigate appropriately
6. As a user, I would like to be able to record videos in real-time so that I can capture important information
7. As a user, I would like to be able to store videos locally so that I can view them later also
8. As a user, I would like to view the history of previously recorded videos so that I can visit them later again
9. As a disaster rescue operative, I would like the resolution to be good enough to analyze the video feed to determine structural integrity of the buildings and the rubble

Controllability

10. As a user, I would like to be able to rotate the camera mounted on the vehicle so that I can get a 360° view of my surroundings
11. As a user, I would like to move the car up, down, right, and left through a mobile app so that I can maneuver the car through difficult terrain
12. As a user, I would like to have the ability the control the speed of the car through the mobile app so that I move the car in varying speeds (if time allows)
13. As a user, I would like the car to navigate autonomously from one place to another so that the user can focus on other tasks as well (if time allows)

Connectivity

14. As a user, I would like to be able to connect to the car via wifi to be able to operate the car from a large distance
15. As a user, I would like to be able to connect to the car via bluetooth in case there isn't wifi (if time allows)
16. As a user, I would like the car to have its personal internet dongle to provide connectivity to the module (if time allows)
17. As a user, I would like the car to be able to return to the location of the phone it was connected to in case of a connection loss (if time allows)

UI & UX

18. As a user, I would like the app to be neat and easy to navigate so that it does not take time to learn
19. As a disaster rescue operative, I would like the car to have the ability to report the average temperature readings of the surrounding area so that I know how hot the environment is
20. As a user, I would like the temperature readings to be displayed live on the screen as part of the live feed
21. As a user, I would like the temperature reading to change colours according to the level of danger (red for high or unsafe temperatures, green otherwise)
22. As a user, I would like the app to report if any of the sensors are malfunctioning so that they can be replaced (if time allows)
23. As a user, I would like the vehicle to report back battery life as accurately as possible so that I can get the vehicle back to safety before the battery runs out (if time allows)
24. As a user, I would like to be able to have a GPS location reading so that I can know where the car is exactly at all times throughout its operation (if time allows)
25. As a user, I would like frequent software updates so that the vehicle can continue to run efficiently (if time allows)
26. As a user, I would like to be able to store data from sensors such as the temperature and relative humidity (RH) sensors and display them on the app in graphical form so that they can be analyzed easily (if time allows)

Hardware

Controllability of car

27. As a user, I would like to be able to rotate the camera mounted on the vehicle so that I can get a 360° view of my surroundings

28. As a user, I would like the hardware to respond to movement commands quickly so that the actions and movements of the car happen in real time and can be seen through the camera
29. As a project owner, I would like to reverse the polarity of the motors so that the car can move in all directions smoothly
30. As a disaster rescue operative, I would like to have a grid laser displayed on the terrain in front of the car so that it can simulate depth perception and highlight cracks or bumps in the terrain (if time allows)
31. As a disaster rescue operative, I would like the car to have the ability to sense surrounding rubble and not crash into them so that it does not get damaged (if time allows)

Modules

32. As a user, I would like to have the car be easy to modify using additional sensors or add-ons
33. As a disaster rescue operative, I would like to have the car equipped with flashlights so that I will be able to evaluate surroundings in low-light situations (if time allows)

Non-Functional Requirements

Architecture and Performance

The software to control the remote controlled vehicle will be built with a completely separate frontend and backend. This will allow for effective creation of sub-teams and efficient division of labor.

The backend will be written in Python and model a RESTful API (Representational State Transfer) using Flask. Flask is an easy-to-use micro web framework that lets developers build backends very quickly and allows for very fine-grained control of the application. Functionality can be added as required using powerful extensions, which include ORM (Object-Relational Mapping), http authentication among others. This ensures that the backend can respond to requests as quickly as possible (<500 ms) and is not weighed down by extra, unnecessary features that come out of the box in other frameworks.

The frontend will be an android application developed in Android Studio using Java and XML. The only way the frontend will be allowed to communicate with the backend is by making requests to the API using clearly defined endpoints. This kind of separation of the frontend and backend will make it easy to develop other frontends, should we choose to do so, in the future (like iOS or web).

Control Ability

A Raspberry Pi 3B will be an access point for changing the motions of the RC car, which is powered by H-Bridge motors, as well as managing data and modules. The commands will be sent through an android application and authenticated by Pi using IP addresses and unique serial number.

Connectivity and Scalability

Wireless communication between the application and Raspberry Pi is enabled by TCP client-server socket. The range of the communication depends on the frequency, transmission power, antenna type, the location, and environment. A typical wifi range is around 50 meters indoor and 90 meters outdoor. Portable wifi and extender will be a choice of customers. Communication between motors and Pi will be achieved using GPIO of Pi. A backup system such as offline reverse route will be implemented to address unexpected connection failure situation. If time allows, we also want to include bluetooth connection.

Usability

The user interface should be straightforward and user-friendly for controlling the car as well as navigating to different data display pages. There are various mobile applications for controlling an RC car, however, none of them will be applicable to our product because all the other functionality that our product will have such as temperature sensing. Color scheme will be carefully considered for our application to provide a better user experience since it will present relevant graphs for selected data. We also want to ensure that our application is accessible and well-presented on all phone screen sizes and resolutions. The car will be able to carry certain weight of customized add-ons such as the CO2 sensors.

Security

Like any application which stores user data, security is of utmost importance to *TEARv*. Flask's extensions for database access and user authentication come out of the box with measures that protect against common exploits like SQL injection. All requests to the API will be authenticated to prevent abuse and ensure only verified users can control the vehicle. The personal databases will only be available to the user and access will not be distributed publically to any unrelated users or other affiliated organizations.

Hosting and Deployment

Since the frontend and backend will be developed as completely disjoint entities, they can be deployed and maintained/updated separately. The backend will be deployed to a remote server running at all times on the Raspberry Pi for as long as the vehicle is operating.