

Group 2 - RoboPaw Design Project Final Summary

Ayush Patel, Bogdan Catana, Anirudh Yallapragada, Shane Abraham

For the design project, we are creating a robotic dog arm, RoboPaw. RoboPaw will be a tool for dog owners to help complete certain tasks. Currently, there isn't a pre-existing system to go off as RoboPaw is a new product for this type of system.

For design goals, the robotic dog arm has to be comfortable for the dog to use for every size and there needs to be stability while completing tasks. The app associated with RoboPaw should have different sections that users can easily access. These sections include the user's personal profile page, accessing the robotic arm through a valid connection, assigning a task and receiving the data that the robotic arm generates, rating the performance of each task, and a forum section for users to interact with each other. These different features in the app will make it easier for the user to navigate through and the administrators to manage.

For data management of the AI implementation, there will be use of different software applications such as Amazon Web Services (AWS), Google Cloud Platform (GCP), Tableau, and Microsoft Power BI. The data we receive will be monitored continuously to ensure that the AI implementation of the robotic arm is performing well. To ensure access control and safety, there will be constant monitoring of the hardware parts that will be used to keep the robotic arm in control. There will be software protections to ensure no viruses are installed in the app. Since the code of the software will be open-sourced, there will need to be a security measure where the base code isn't changed. For the app, there will be different sections for the user to access.

For the final system design, there will be a use of various architectural patterns such as the model-view-controller architecture that will be needed for clear UI-data separation for the mobile application. Client-server will be used for keeping the controls centralized to ensure safety-critical applications.

For the object design, there will be different packages used, including `com.robopaw.arm`, `com.robopaw.vision`, `com.robopaw.remote`, and `com.robopaw.ai`. There are a few subsystems that will be factored into RoboPaw's design. For the Robotic Arm Control subsystem, there will be an `ArmController`, `GripSensor`, and `BatteryManager`. The second subsystem is about the robotic arm's vision, which contains a `CVProcessor` and `ObjectTracker` systems. The third subsystem is the robotic arm's remote controlling capabilities, which include a `CommandHandler` and `FirmwareUpdater`. The fourth subsystem is about the AI task processing of the robotic arm, which includes a `TaskInterpreter`, `TaskScheduler`, and `TaskExecutor`.

One of the main issues with RoboPaw is ensuring that the robotic arm can handle various tasks regardless of difficulty. There needs to be a process that goes over the scalability of RoboPaw. Some off-the-shelf products can be used in the design process of RoboPaw. Certain software systems that are used for AI and ML training can be used for training the robotic arm. Certain hardware systems have been used for robotic arms such as the VEX robotic toolkit which will have some hardware components that will be used. The latest hardware components will be used to ensure that RoboPaw is updated with the latest features. These different software and

hardware components are ready-made so that any functionality issues can be resolved. The software libraries that come with the AI and ML training models and the robotic arm toolkits will be used throughout the design process. As there aren't any specific robotic dog arms out in the market, there won't be an issue with copying products.

Certain new problems might come with RoboPaw. As the purpose of RoboPaw is to help complete tasks assigned by dog owners, that will result in needing fewer people to help those dog owners. For example, service dog owners would suffice with their dogs and RoboPaw would help take care of them and complete tasks, which would mean fewer people would be needed to care for the service dog owner. There needs to be boundaries placed for the original systems to remain in place. The user might be frustrated when assigning a task, understanding the response of the robotic arm, and knowing the limitations of RoboPaw. Therefore, it is possible that the user response can be negative in the beginning. The environment that RoboPaw is in might present certain limitations, such as the weather, WiFi connection, collision hit, and battery life. Depending on the environment, RoboPaw might not have been trained in it, which would lead to it not being able to complete the assigned task.

There are certain risks that can come into play, such as technical issues (glitches, AI and ML implementation, connection), hardware issues (incorrect components, sensors, motor, stabilizing components), and less coverage of potential tasks being assigned. These factors depend on how the user is using RoboPaw as these factors are situational and shouldn't be a consistent risk.

There will be a team dedicated to hardware and software, which will start with around 20 people on each team. As RoboPaw's demand increases, there will be bigger teams. The salaries starting will be around \$80,000, and grow depending on the demand of RoboPaw. It will take roughly 1-1.5 years to get everything released for RoboPaw (app and robotic arm). There will be various tests and inspections that will take place before the release of RoboPaw. The robotic arm itself should cost around \$150,000 to make, and around \$300,000 to make a top-tier robotic arm. The software used should be in the range of \$50,000 to \$100,000, which includes research. There will be savings made through the development process. A lost opportunity cost would be using different hardware and software components that might be worse than other ones available.

A couple of ideas that will be reconsidered at a later date include adding different types of gadgets to the robotic arm and having a scheduling feature that would allow the user to set a routine of having a task completed at a certain time and day and have that task completed automatically. While there won't be a specific software language that is used, Python will be ideally used for the various AI and ML implementations. There will be software frameworks such as Tensorflow and Movelt that will help with the robotic arm's computer vision and control. The use of these software frameworks will be tested to see which one fits better for RoboPaw according to the testing guidelines that are set.

Overall, the design process of RoboPaw was an amazing experience for our group. We communicated with each other daily on Discord and in the weekly meeting to ensure that everyone was on the same page and knew everyone's progress in their tasks and helped each other along the way.