

Requirements Analysis Document

Project Information

Project Name	Team ID	Mentor(s)
Pi Vision	31	Niek Aukes & Cristi Angheluș

Security by Design Checklist

Security Mechanism	Brief Justification	Risks	Mitigation Controls	Applied?
Authentication	The user does a gesture and when recognized they can use the system	Someone could guess / program could false positive on the pass gesture	Authentication through password	✓
Stand-by option	The program should close after a certain period of inactivity to enhance the security of the system.	The program closes too fast or does not close after change of users	Automatic timeout, re-authorization after a specific amount of "idle-time"	✓
Camera Access Control	Only the system should be able to access the webcam for recognition.	Malware or other processes could hijack the webcam feed.	Limit camera device access (permissions, OS-level control), encrypt video feed if it is sent remotely	✓
System Integrity (Raspberry Pi)	Device should run trusted software only	Unauthorized software/tampering could compromise system	prevent/prohibit uploading other code to the raspberry Pi after the product is finished/ working/ optimized. (done is done)	✓

Introduction:

Purpose of the Application:

The aim for our group is to develop a hands-free system; the main purpose will be to control and use your computer without the aid of physical elements such as mouse, keyboard or controllers. To realize such idea, we will use a webcam to capture and then process hand symbols to later translate them into inputs using computer vision.

Limitations of Existing Solution:

There are a handful of available computer vision programs to control your computer using a webcam; all of the ones that we were able to find mainly focus on simply controlling the mouse. We also plan to make it possible to play certain games too. Essentially, our project will not be only limited to mouse control, but it will also include keyboard controls.

Scope

Resources Needed:

Hardware:

Below, it is possible to find a list of the different hardware materials that will be used for the realization of our project:

1. RASPBERRY PI 5 - 4GB
2. Logitech c270 HD WEBCAM

Software:

Having mentioned the hardware part of our project, below is presented a list of the software elements that will be used for the realization of the project:

Python

- Due to the fact that most, if not all, computer vision frameworks are built using Python, the Raspberry Pi also supports Python. Therefore, we will be writing our code in Python.

Libraries and frameworks

- *HaGRID* = hand-gesture recognition
- *MediaPipe framework* = a machine learning framework powered by Google

SMART Goals:

Goal 1:

Specific: Complete gesture recognition on the Raspberry Pi

Measurable: The Raspberry Pi will be able to detect and recognise a fixed number of gestures.

Achievable: Attempt to use Google's MediaPipe library to streamline the process, if it is not possible, use OpenCV to make our own detection system. Then use a pretrained model for gesture recognition.

Relevant: This goal is the base of the system, without gesture recognition the system will not be able to function.

Time-bound: Completed by sprint 4.

Goal 2:

Specific: Integrate a user login functionality to grant all users secure access to their accounts

Measurable: Implement a username-password type of login (and possibly SSO) and test it

Achievable: This would be achievable by registering a personal gesture for every user.

Relevant: Having a way to protect users account is a major concern to ensure users privacy and guarantee security

Time-bound: The login page should be completed anytime before **Week 5** to be able to test it properly and guarantee it correctly works.

Goal 3:

Specific: Improve the learnability of the system by designing a synthetic, always-available help menu.

Measurable: Implement a command which, any time during the session, will automatically open a list with all the commands, their name and what they correspond to.

Achievable: Use the right set of libraries to finally code the shortcut which will translate the hand gesture to open the help menu.

Relevant: Granting an always-available help menu will allow for a better user experience and will allow the user to easily learn how to use the product.

Time-bound: Complete the help menu by **Week 9** as it should be completed after the program has been finished and tested.

Product Features:

Functional Requirements:

The system should allow users to control a computer without using a keyboard or mouse, but solely using hand gestures.

The system should allow to completely substitute a mouse, with each designed gesture being able to click and scroll

The system should be able to substitute some of the Core keyboard inputs such as ESC, the keys W, A, S, D, backspace, space and enter.

The system should allow users to change the mapping of gestures to different keystrokes.

The system should distinguish between the left and right hand, and be able to handle different commands for each one.

Non-Functional Requirements:

The system should not take more than 0.5 seconds to compute a hand gesture.

The system should not have more than a 5-second delay for device control.

The system should be able to recognize hand gestures correctly as long as there is sufficient light.

The system should have a fixed range of gestures for the user to use.

Security Requirements:

Implement user login before allowing the user to change different key mappings.

Implement a gesture-based authentication mechanisms which will allow users to input the access password to their accounts

Implement a stand-by mechanisms which will disconnect the user after a certain period of inactivity (namely a period of where non-valid inputs have been detected)

Conclusion

In summary, our group's project aims to shift away from traditional input systems, focusing on using the hands as the sole means of input. Although this may be a project that has already been conceived and implemented in the past, our research shows that various companies have always limited themselves to replacing only the mouse. Our group, on the other hand, aims to replace the entire traditional input set: mouse and keyboard. We will maintain the security of the system by implementing: an authentication system that will require the use of a password, and a standby mode that will disconnect the user's activity if there are no valid inputs for a certain period of time. In the near future, we will mostly focus on implementing the UI, together with the user authentication; moreover we will start to familiarize with the libraries and start to implement the system.

References

Before completely committing to this project, we wanted first to do some research and understand if such an idea would be plausible or not in the given 10 Weeks we have to develop it.

HaGRID (<https://www.kaggle.com/datasets/innominate817/hagrid-sample-30k-384p/code>)

First of all, we conducted research to understand what gestures could be used and how they could be captured and interpreted. We then searched the link above to study the set of gestures that could be used for our project. This gives us the ability to implement more commands and associate them with more inputs, helping us to better realize our idea.

MediaPipe

framework(https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer)

Not only do we have to recognize the hand gestures, but we also need to assign them to specific values, customizing them according to our preferences and our interests. This could be done using the above-mentioned MediaPipe framework, a cross-platform machine learning framework by Google which will allow us to have a trained ML model rather than creating one ourselves, which will be difficult and require more time than the one available to us.

Usage of AI Tools

We used ChatGPT in the beginning to find out if it was technically feasible to run a computer vision application to detect hand symbols in real time