Ruben Arutyunov

Lance Willet

Nate Diemer

Security Application Using Facial Recognition with Raspberry Pi Camera

Functions

1. Facial Recognition: The application will identify a person’s face using the Raspberry Pi Camera.
2. Store Image: The application will store an image of a person’s face in a database to be referenced later.
3. Facial identification: The application will correctly identify a user by comparing a stored image to live video taken by the Raspberry Pi Camera.
4. Store Pertinent Information: The application will store information such as the time and date when a stored image is added to a user account using facial identification.

Challenges

Note: As our project has changed direction from the initial presentation some challenges have been dropped and added to accommodate the new design.

1. Novelty Vs. Practicality: One of the big issues with our project is to create a solution that could be implemented for practical security usage. We have yet to identify any limitations that our usage of the Raspberry Pi would have vs. typical security equipment, but the novelty of the design is the price point and portability of the Raspberry Pi, as well as functionality that can be built from the ground up.

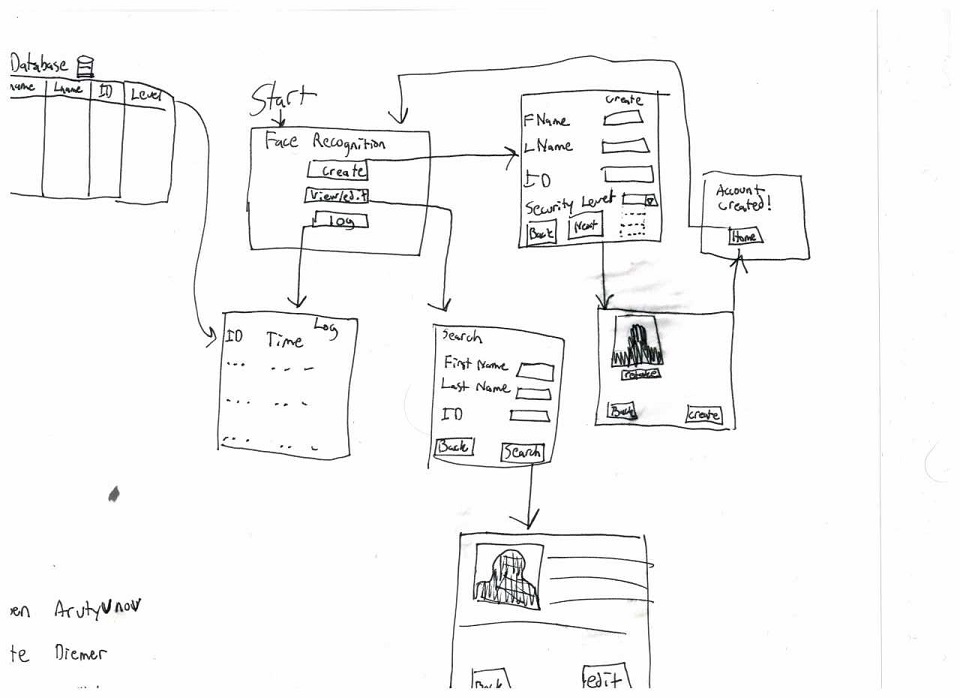
Our Solution: We will meet weekly to discuss progress, and also to voice feedback on project progress. During these meetings, we will discuss how we can better make use of the unique features the Raspberry Pi bring to the table in a security application, and as new features are implemented, and the interface created, work to optimize the advantages of the Raspberry Pi.

1. Connecting Front-End to Back-End: A design challenge we will have to overcome is linking our interface to our python code. Currently, we intend to create our interface in Visual Studio.

Our Solution: Right now, implementation of the interface is on the backburner as we are working towards getting our code ready and working on creating a database. As such, this solution is a little more abstract, but currently we intend to seek out a Visual Studio plug-in that will allow us to activate python scripts on-click.

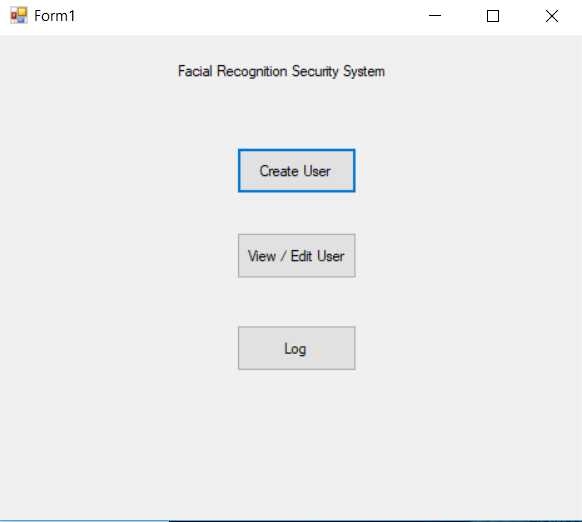
Prototypes

Lowest Possible Fidelity

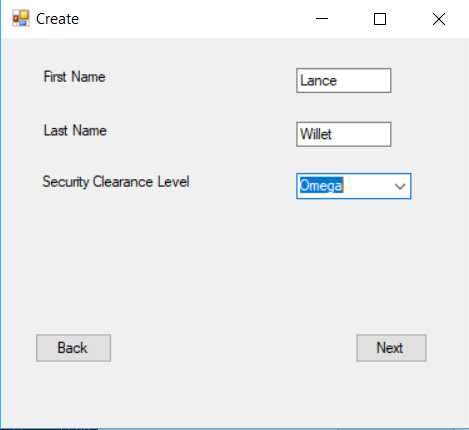
****

Working Prototype

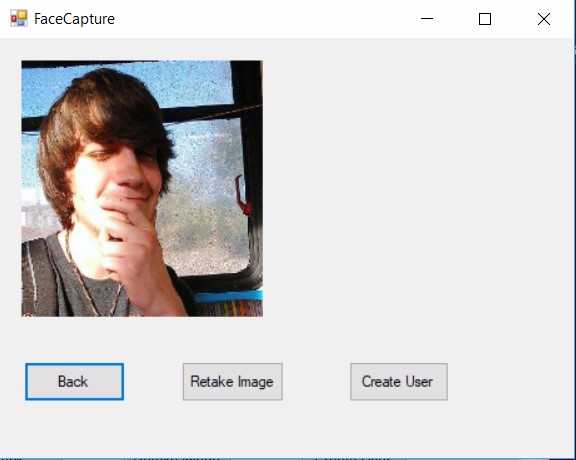
1. Home Screen



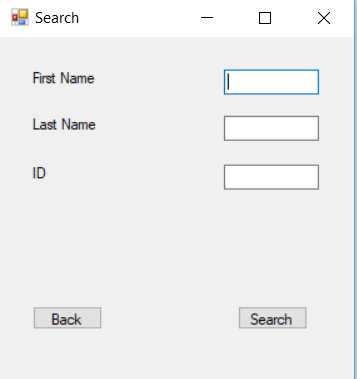
1. Create Account



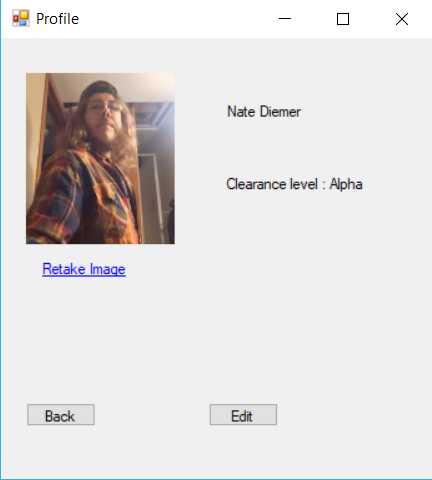
1. Add Image



1. Search for Account



1. Account Information



Design Discussions

The novelty of our design comes from its ability to take advantage of both the portability and price point of the Raspberry Pi to create a functional, robust, and adaptable security system. Raspberry Pi is also growing in recognition, becoming more of a house-hold name, making this the perfect time to showcase just how powerful of a tool it is. However, many of the current common uses of the Raspberry Pi are primarily focused on entertainment, such as the “RetroPie,” which allows the user to turn their Raspberry Pi into a powerful video game emulation machine. This project, however, will help to showcase the adaptability and power that the Raspberry Pi offers any consumer outside of its more common applications. As initially stated, our team wanted to couple the usage of new hardware, such as the Raspberry Pi, with rapidly growing and expanding software concepts, such as facial recognition, and this is the perfect project to couple them while making good use of the HCI concepts we are learning.

Responsibilities

Ruben - Ruben’s primary responsibilities involve making sure the camera and the raspberry pi interface properly and that the faces can be stored in, and retrieved from, a database. Ruben is also in charge of handling the transfer of user input to the backend, and taking care of the transfer of user requested information to the interface.

Nate – Nate’s responsibility is to design the pages where new users will be created and stored in the database. Nate is also in charge of documentation.

Lance- Lance’s responsibility is to design the pages where users can be queried and edited. Lance is also in charge of designing the page that displays the log information.

Tentative Timeline

Week of 10/23: Design Document, 2nd presentation Preparation, alpha UI, implement OpenCV

Week of 10/30: 2nd Presentation on 11/1, continue UI work, move OpenCV to Raspberry PI

Week of 11/06: Complete Initial UI, connect frontend to backend

Week of 11/13: Create Database & connect it to backend, create interaction logging feature

Week of 11/20: Tweak Algorithms (# of images captured, optimal facial recognition to camera distance/ etc)

Week of 11/27: Study for final, loose ends, prepare for final presentation

Week of 12/4: Final presentation