Requirement Analysis and Specification Report

Functional Requirements

1. User is able to create an account.
    - 1.1. User will get the option to create an account.
    - 1.2. User will register with their first name, last name, an email and a password.
        - 1.2.1. User must provide a valid email.
        - 1.2.2. The password must be English letters or numbers
        - 1.2.3. The password must be at least eight characters long with at least one uppercase letter.
        - 1.2.4. User must reenter the password
        - 1.2.5. All fields and requirements must be fulfilled in order to register.
    - 1.3. Once completed, show a link to the landing page.

    Description: This functional requirement will take the user through the process of creating an account.To register for an account, the user will be prompted to input their name, email and password. Passwords must follow all the guidelines mentioned above. The user must complete all input fields in order to complete registration.

2. User is able to login with credentials.
    - 2.1. On the landing page, user will input their email and password and click the "Login" button.
    - 2.2. User will then be redirected to the dashboard if the credentials are valid.
        - 2.2.1. If the user's credentials are not valid, incorrect username or password will be displayed.
            - 2.2.1.1. There will be an option for the user to reset their password if they fail to log in.

    Description: This functional requirement will walk the user into the process of logging into their account. If the user is registered they will input their email and password. If credentials are valid, they will be redirected to the homepage. If their credentials are not valid, incorrect username or password will be displayed by the system.

3. User is able to change their password.
    - 3.1. When logged in, user has access to update their current password by clicking the "Change Password" button.
        - 3.1.1. User is prompted to provide their current password, and what they want their new password to be.
        - 3.1.2. The password must be English letters or numbers

    - 3.1.3. The new password must be at least eight characters long with at least one uppercase letter.
  - 3.2. If user forgets their password, they can click the “Forgot Password” button shown under the log in application.
    - 3.2.1. User will be asked for the email linked to their account.
    - 3.2.2. An email will be sent to the provided email with a link to reset their password.
    - 3.2.3. The user’s new password must follow the same guidelines as the initial password they created.

Description: This functional requirement describes how a user can update their current password. The user will input their current password in order to make a new password. If the user forgets their current password they can click a “Forgot Password” button. The user will then be prompted to input their email, and they will receive a link to create their new password.

4. User is able to view the live feed at all times.
   - 4.1. The Raspberry Pi is connected to a power source and the internet, enabling the user with 24/7 surveillance.
   - 4.2. User is able to access the live feed through hitting the “Live Feed” button on their homepage.
     - 4.2.1. User is taken to a new page that shows the surveillance area of the camera.
       - 4.2.1.1. User is able to see if people walking past the camera are friends or strangers based on the box that appears around their head.
       - 4.2.1.2. If the person seen in the camera is a friend, their name will also appear.
   - 4.3. The cameras are recording video in resolution 720P (1280*720) with 15FPS (Frame Per Second) in 5 Mbps (Megabits Per Second) bitrate and stored as MP4.
     - 4.3.1. This allows the application to have enough data to identify the person, but not so much as to occupy excessive space in the database.

Description: This functional requirement describes how the user can access the live feed of the security camera at all times. By clicking on the “Live Feed” button in their user homepage, they are taken to a new page that shows the camera’s current feed. The application still allows for recognition of people during this live feed and the user is able to see the box around a person’s head indicating if they are a friend or stranger. If they are a friend, the friend’s name will also appear.

5. User is able to view past feeds.
   - 5.1. User can find old feeds by clicking the “Past Feeds” button on the homepage.

5.1.1. Past feeds are composed of 2 minute segments.

5.1.2. Past feeds are sorted by date and time.

5.1.3. If the user clicks on a specific date and time, a new page with all recordings that apply will be presented.

5.2. The storage page will show how much time remain in minutes it could record

5.3. There will be feeds stored until the storage is not long able to store one more recording.

5.3.1. While the storage is full, the earliest recordings will be deleted to free up space in the database.

5.3.1.1. When deletion occurs, the earliest files will be deleted until a sizeable buffer of storage capacity is present to accomodate for new recordings.

Description: This functional requirement describes how the user is able to view past feeds taken from the surveillance camera. By clicking the "Past Feeds" button in their homepage, they are taken to a new page displaying different dates and times that were recorded. The user can click on any of these recordings and see the feed from that specific date and time.

6. The Raspberry Pi collects and stores information.

6.1. The Raspberry Pi will have an SD card that collects data from the camera.

6.1.1. Images and videos taken from the camera will be saved in the Raspberry Pi's SD card.

6.2. The Raspberry Pi will be connected to a database.

6.2.1. When the SD card reaches a certain capacity, the files will be copied to the database.

6.3. The files are deleted from the Raspberry Pi's SD card after being copied to the database.

6.3.1. The process restarts everytime the Raspberry Pi's SD card reaches its capacity.

6.3.1.1. When deletion occurs, the earliest files will be deleted until a sizeable buffer of storage capacity is present to accomodate for new recordings.

Description: This functional requirement describes how the Raspberry Pi will be handling data organization. Image and video files will be saved in the Raspberry Pi's SD card. When it reaches its capacity, the files will be transferred to the database. The SD card in the Raspberry Pi will then be cleared and the process will restart.

7. The system is able to differentiate people who appear in view of the camera.

7.1. The code to differentiate people will utilize an open source API for image analysis and recognition.

7.1.1. The open source API is OpenCV.

7.1.2. OpenCV will be used to take in the characteristics of the person's face and analyze them.

7.2. To indicate whether the person shown in the camera's feed is a friend or stranger, a box will appear around their head.

7.2.1. If the person is a friend, the box around their head will be green and include their name in the border.

7.2.2. If the person is a stranger, the box around their head will be red and read "Stranger" in the border.

7.3. The indicators if the person in the camera feed is a friend or stranger will be present in both current feed and past feed.

Description: This functional requirement describes how the system can differentiate between people. The code for this functionality will utilize OpenCV, which is an open source image analysis and recognition API. OpenCV will provide the functionality to identify different people. Implementing our own code along with it results in the user being able to see if the person in their camera's feed is a friend or stranger. This identification is present in the current and past feeds.

8. User can enable notifications.

8.1. User can choose to be notified if any motion is detected.

8.1.1. If the camera detects any motion in its surveillance area, the user can be sent an email indicating this, and a link to open the camera's current feed.

8.1.2. The email will also contain the date and time this was detected so the user can look back on it.

8.2. User can choose to be notified if a friend is detected.

8.2.1. If the camera detects a friend in its surveillance area, the user can be sent an email indicating this, along with the friend's name, and a link to open the camera's current feed.

8.2.2. The email will also contain the date and time this was detected so the user can look back on it.

8.3. User can choose to be notified if a stranger is detected.

8.3.1. If the camera detects a stranger in its surveillance area, the user can be sent an email indicating this, and a link to open the camera's current feed.

8.3.2. The email will also contain the date and time this was detected so the user can look back on it.

Description: This functional requirement describes the different notifications the user is able to receive via email. The user can be notified for motion, friends, or strangers detected within the camera's surveillance area. The notification will be sent to their email, will include the type of notification it is (motion, friend, or stranger) and a link to watch the current feed, and will include the date and time so the user has the ability to go to the past feeds and watch the feed at the time the detection took place.

9. User is able to register a person as a "friend".
    - 9.1. User is redirected to a page for registering a friend after clicking the "Register a Friend" button.
        - 9.1.1. This page requires the user to provide 5 different pictures of the person they want to register, along with the person's name.
        - 9.1.2. After providing the 5 pictures and a name, the user clicks the "Register" button.
        - 9.1.3. User is shown a confirmation page on completion of the above steps.
        - 9.1.4. User is sent a confirmation email when the person has been successfully added as a "friend".

    Description: This functional requirement describes how a user is able to register someone to be recognized by the system as a "friend". The user can click on the "Register a Friend" button in the home page and be taken to a new page to perform this. They are required to provide 5 pictures and the person's name.

Non-functional Requirements

1. Website will be accessible at all times.
    - 1.1. The system will be accessible to the user 24/7.

    Description: This non-functional requirement describes the accessibility of the system.

2. System shall allow the user to permanently delete the account.
    - 2.1. If the user has no use of the system, they will have the option to permanently delete their account.
    - 2.2. In the settings tab in the homepage there should be a "Delete Account" button
        - 2.2.1. User will be prompted to input their email and then will be deleted from the system.

    Description: This non-functional requirement describes how the user will be able to delete their account if they have no use of the system anymore.

3. System will allow user to login within 2 seconds.
   3.1. Website will display login window
      3.1.1. Website will have a display box for user-name and password.
      3.1.2. Also along with a submit button.
      3.1.3. Website will have a forgotten password box.

   Description: This non-functional requirement describes how fast the login page will be displayed and what will be on it

4. System will allow user to change their password within 2 seconds of attempt.
   4.1. When the button is pressed, an email will be sent with details about how to update password in a secure manner.
   4.2. Site will verify that the correct email has been sent to the correct user.

   Description: This non-functional requirement describes what will happen when the user wants to change their password.

5. System shall allow a user to login within 6 seconds of an attempt.
   5.1. The website will be fully displayed with correct information.
      5.1.1. Website will be organized and laid out nicely to avoid confusion.
   5.2. Site will display updated information.
      5.2.1. Live feed available to watch.

   Description: This non-functional requirement describes how the site will load, with what information and how fast.

6. Website will display the live feed within 7 seconds of being clicked on.
   6.1. When a user clicks the link to view the live feed it will be pulled up in a separate browser.
      6.1.1. In the separate browser, the live feed will take up the entire screen.
   6.2. There will be a button to click to go back to the home page.
      6.2.1. Live feed will then close.

   Description: This non-functional requirement describes how the live feed will be displayed and how fast it will be loaded.

7. There will be a button that lets the user add a friend within 10 seconds.
   7.1. Once the button is clicked, a new page will open with details about how to add.
   7.2. After information is added, the site will gather the information and add the new person in under 10 seconds.
      7.2.1. After this, the user will have added a registered friend.

Description: This non-functional requirement describes how fast the website will add a new person and how it will be displayed.

8. Website will be organized to ensure efficiency for the user.

    8.1. Headings and links will be added.
    8.2. A preview of the live feed will be available.
    8.3. An event log will be pulled up.
        8.3.1. Who has entered the house and if they are a friend or an unregistered guest.

Description: This non-functional requirement describes how the site will look.

9. There will be a button that allows the user to contact help.
    9.1. If an unregistered user is detected, the user will be able to push a button that sends authorities to the house.
        9.1.1. The event will be recorded and stored in a negative event log that can be watched whenever.

Description: This non-functional requirement describes how the user can contact help if needed.