

User Stories Report

Thermal Scanning App

Colter Roche, Jose Bastardo

Senior Design 1

COP4934C.01

November 23, 2020

Contents

1	Epics and User stories	2
2	Derived tasks and Pseudocode	3
2.1	Detect faces in video	3
2.2	Calculate temperature discrepancy and adjust camera	3
2.3	Measure temperature within the segment of the thermal image that corresponds to the user's face	4
2.4	Check if user is registered	4
2.5	Send notifications to user and alerts to screen	4
2.6	Compare user temperature to past temps and check for too high temp	5
2.7	Register facial data through app	5
2.8	Register users through app	5
2.9	Allow admin to control data requested and storage duration .	6
2.10	Produce reports for organization and users	6
3	Team Member Participation	7

1 Epics and User stories

Epic	User Story
Provide facial recognition for user identification and measure user temperature	As a user, I want the system to recognize me in under 2 seconds so I can save time
	As a kiosk attendant, I want to see a confirmation that the user is recognized, for security purposes
	As a kiosk attendant, I want temp measurements to be accurate withing a degree, so I do not have to check extra people/miss people
	As a kiosk attendant, I want the thermal camera to self-calibrate, to avoid the need for time consuming troubleshooting
Provide notifications and reports to users and attendants	As a kiosk attendant, I should be alerted if a registered user is detected with a high temperature to perform a manual temperature check
	As a kiosk attendant, I should be alerted if a person is not recognized as a registered user
	As a user, I should be alerted if my temperature is too high
Provide a user onboarding system	As a user, I want to register through a mobile app, for easier remote registration
	As a user, I want to register facial data through the app, so I can skip in person registration
	As a system admin, I want to have control over what information is collected from users and how long the data is stored, to increase security

Table 1: Epics and User Stories

2 Derived tasks and Pseudocode

The following tasks and Pseudocode were derived from the user stories listed in Table 1.

2.1 Detect faces in video

Face detection is used to determine whether a user has entered the view of the camera in order to begin facial recognition and measuring their temperature. This is done using the opencv library in Python.

```
1      Read camera stream
2      Use OpenCV to detect faces
3      Store face pixel array in var
4      Apply smoothing and post processing
5      Return section array
```

2.2 Calculate temperature discrepancy and adjust camera

The thermal camera is known to lose accuracy depending on the surrounding temperature and weather conditions. A hot plate will be placed in the vision of the thermal camera and will be used to calibrate it. This will provide more consistent and accurate measurements.

Algorithm 2.2: Thermal cam calibration.

```
1      timer: every 3 minutes
2      on timer firing
3          select section of video containing calibration device
4          measure temperature
5          if measured temperature does not equal set value for temperature
6              subtract set from measured temp
7              adjust calibration by result
```

2.3 Measure temperature within the segment of the thermal image that corresponds to the user's face

Once the user's face has been segmented from the video stream, the rgb and thermal images need to be compared and the correct pixels measured for temperature. Because the resolution is different between the images, additional calculations are required. The data from the thermal camera is then returned.

Algorithm 2.3: Temperatur reading.

1	Receive array of pixel data containing face
2	Calculate corresponing pixel locations <i>in</i> thermal image
3	return thermal readings from those locations

2.4 Check if user is registered

Once a face is detected within the cameras vision, the application will use facial detection to store the users facial data and compare it to the facial data already stored in the database to check whether the has been registered through the system.

Algorithm 2.4: Facial info search.

1	Connect to database
2	Receive tuple with facial data files
3	Loop for length <i>in</i> tuple
4	If new data is equivalent to registered data
5	User is registered
6	Break loop

2.5 Send notifications to user and alerts to screen

If a high temperature is detected by the system, the application will send an alert through the screen on the tablet and to a kiosk attendee and the user. The system will also be able to send an alert to the attendee if the user is not registered.

Algorithm 2.5: Temperature notification

1	If high temperature detected
2	Display high temperature alert on screen
3	Connect to database
4	Search for email using user id
5	Send message to user email
6	Send alert to attendee email

2.6 Compare user temperature to past temps and check for too high temp

The application will consider the average temperature of the user by searching through the user's past scans and taking the average of their temperature history. A limit will be set based on their average temperature. If the user's temperature passes that limit it will be considered a high temperature.

Algorithm 2.6: Check user temperature

```
1      Search database for user temperature history using user id
2      Receive array with past temperatures
3      Initialize counter at 0
4      Initialize sum at 0
5      Loop for size of array
6          Sum is equal to sum plus tuple[counter]
7          Average is equal to sum divided by size of array
8          Limit is equal to average plus 3 degrees Fahrenheit
9          If limit is greater than current temp
10             User temp passes
11         Else
12             User temp fail
```

2.7 Register facial data through app

To reduce the amount of time the user needs to spend registering in person, and to improve registration times for large userbases, users will be able to register for the facial recognition from a companion app. Several images will be captured, the algorithm will be trained, and then the images will be deleted and the model stored in the database.

Algorithm 2.7: Register facial data

```
1      App receives access to phone camera
2      Use facial detection to detect face
3      If face detected
4          Take pictures of user
5          Convert pictures into facial recognition data files
6          Send files to database
```

2.8 Register users through app

Users will be required to provide some identifying piece of information to create an account and eventually register facial recognition data

Algorithm 2.8: Register user data

```
1      User enters identifier
2      User enters password
3      Form entries are verified for format with regex
4      If info matches format
5          Create new entry in database
6          Store user info
7          Send user confirmation email
8      if user confirms
9          finalize user registration
```

2.9 Allow admin to control data requested and storage duration

User data collected for registration will vary based on use case. Most use cases will minimize data collected to an Employee ID number, but the admin may want to collect more data. To account for these variations, the application will allow the admin to set what information will be collected and how long the data will be stored.

Algorithm 2.9: Manage User data settings

```
1      Display list of options with on/off toggles
2      Display data storage duration with default value of 1 month
3      if at least one option is selected
4          enable save button
5      on save button clicked
6          set user data requirements
7          set data expiration time
```

2.10 Produce reports for organization and users

Companies and users will be able to view reports of temperature measured and time/location the data was collected. Company reports will list all users for a period of time, including times, location and temperature. Users will have access to similar reports, but only applying to them.

```
1      Request time period for reports (Default is 1 month)
2      if Company
3          query db for users recorded during that period
4          display sortable table listing users, locations and temperature
5      if User
6          query db for users recorded during that period
7          display sortable table listing locations and temperature for that user
```

3 Team Member Participation

Participation was split as follows:

- Colter Roche: 50%
- Jose Bastardo: 50%