Economic Home Security System

DESIGN DOCUMENT

Group 42
Client - Goce Trajcevski
Advisor - Goce Trajcevski
Kamini Saldanha - Full Stack Engineer
Andrew Tran - Full Stack Engineer
Lucas Jedlicka - Engineering Lead
Sohum Sawant - Security and Testing Lead
Uma Abu - Full Stack Engineer
Merin Mundt - Project Manager
Email: sdmay20-42@iastate.edu
Team Website:
http://sdmay20-42.sd.ece.iastate.edu/
Revised: 10/06

Executive Summary

Development Standards & Practices Used

All of our software development will be done using Agile Development.

We will work on requirements in two week sprints.

We will conduct frequent maintenance procedures on our procedures.

Summary of Requirements

- Notification upon detection of people or cars.
- 10 frames per second per camera stream at worst.
- Support any mobile device as a client when that is Android 4.4 and above or IOS 11 and above.
- Support Real Time Streaming Protocol(RTSP). To support concurrency with a user's existing security system.
- Differentiate between threatening and non-threatening detections.

Applicable Courses from Iowa State University Curriculum

Computer Science 309, Software Engineering 319, Computer Science 311, Computer Science 228, Computer Science 227

New Skills/Knowledge acquired that was not taught in courses

Designing web applications using a Node JS based tech stack.

Machine Learning libraries to detect motion.

Authentication Tokens for security purposes.

We all have to learn Python to develop the backend.

Table of Contents

List of Figures

List of Tables

List of Symbols

List of Definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

Thank you to Goce Trajcevksi for agreeing to serve as a client and advisor for this project. We acknowledge the time you are taking to work with us and resources you are providing us with to ensure the success of this project.

1.2 Problem and Project Statement

Given how common theft is in this day and age, we have set out the goal to build a smart home security system from recycled smartphones. Our security solution would give users the opportunity to disconnect from the cloud and avoid the monthly fees that most security systems charge. With the use of RTSP (Real-time Streaming Protocol) which will allow for the extension and adoption of inplace security systems. We plan to target users who already have IP cameras or a potential excess of old smartphones.

1.3 Operational Environment

The intended operational environment for the product is a home with areas that have reliable power and is safe from the elements (i.e. water or extreme temperatures). The supported phones are intended to be used 24/7 with the web application running. Existing IP cameras will have the ability to be used in conjunction to all cameras the user sets up and viewable from the web application. A network connection is required.

1.4 Requirements

FR1. Get streams from smartphones and IP cameras from web application.

FR2. View streams from smartphones and IP cameras from web application.

FR3. Detect people and objects from video streams.

FR4. Filter people and objects from video streams.

FR5. Save recordings when an event occurs.

FR6. Send notifications when an event occurs.

FR7. Proactive security features (i.e. camera panning, camera coordination).

1.5 Intended Users and Uses

Actor 1. Homeowner in need of a security system with access to unused smartphones or IP cameras.

Actor 2. Homeowner with an existing security system with access to unused smartphones or IP cameras.

1.6 Assumptions and Limitations

- Potential for low performance in old and cheap smartphones:

    - Given that users will be using recycled smartphones, there is no quality assurance for how well the smartphone will work with our security system.
    - This means we will have to perform extensive testing with different smartphone models and softwares.
- Camera limitations in low light and quality
    - Smartphone cameras are not designed to work as well at night. This will mean low visibility in low light.
    - The quality of the camera will lower the older the phone is.
- Streaming at a bit rate which can be processed in a timely manner.
    - If the bitrate on streams is too high, the processes will take too much time. However, if the bitrate is too low we cannot ensure optimal motion detection. In order to achieve optimal performance we need to find the correct medium.
- Lack of time and learning curve
    - Some members are not proficient in all the technologies selected. For example, the entire team is not proficient in Python.
    - We only have 1 year to finish this project.
- Server
    - May not be able to handle the amount of load coming in from constant streaming.
- Storage issues
    - We will have to be selective with storage in order to not take up too much space.

1.7 Expected End Product and Deliverables

The expected delivery dates, end product, and deliverables are subject to change.

Deliverables:

V1 Design Document (October 6, 2019)

Server (November 15, 2019)

Basic Web Application (November 15, 2019)

V2 Design Document (November 3, 2019)

Expected end product (expected delivery date: April 15, 2020):

Web Application

REST APIs

User manual

Demo system

2. Specifications and Analysis

2.1 Proposed Design

Functional Requirements:

- Get and view streams from many places
- Detect and filter people and objects
- Proactive security features (i.e. camera panning, camera coordination)
- Save recordings and send notifications when an event occurs

Non-functional Requirements:

- Bitrate should be sufficient to detect people
- Security protocols (i.e. encryption of the streams, user authorization using industry norms)

Current project design:

On the server side, a machine learning library to detect motion. We are also working on solutions to handle and process user streams. We are also currently considering different hardware to be our base server upon which to test. Up till now, our team has started to design the requirements for an API to to transmit information between our client and server. Additionally, we have decided to use a SQL database to store user information.

2.2 Design Analysis

So far we have begun work on our API to send information between our server and client side. One of our struggles has been working with sending so many streams to process all at once. We are currently working on optimizing this process as it is currently inefficient.

We have also been learning a lot of Python as it is needed to work with the machine learning library we have chosen. This will be a long process for the team as none of us have much experience with Python.

We have also been working on implementing user access tokens for security purposes. Our solution of using OAuth tokens seems to be optimal for what we need out of it.

2.3 Development Process

We will be following Agile software development. The following are reasons for choosing an Agile environment.

- Client engagement and collaboration.

- Quick deliverables.
- Flexibility.
- Focus on the user.

2.4 Design Plan

Currently, our design plan includes a camera network consisting of IP cameras and compatible smartphones (Android 4.4+ and iOS 11+). Our smartphone selection is restricted due to browser implementations of the getUserMedia API used to access the cameras. The backend, which has a tech stack consisting of Python with Django framework, will be used to move the streams. FFMPEG is used for video transcoding and capturing frames. YoLo object detection is used for placing bounding boxes on humans or cars. The web application will serve as a view to support two primary functions, linking a smartphone or IP camera to the security network and viewing the live streams or saved clips. ReactJS is the library of choice for the web application.

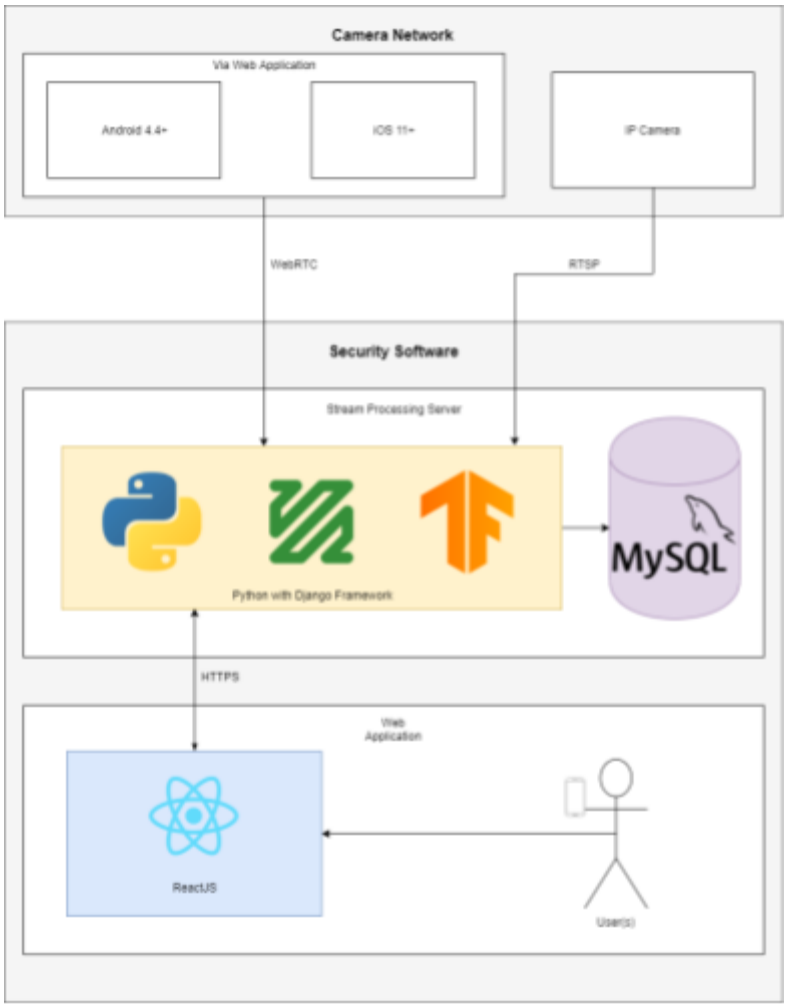

Figure 1. Conceptual Sketch

10/15	11/01	11/15	12/01	12/19	01/20	02/15	03/01	03/15	04/15
Define project goals and requirements	Begin Prototype	Get/view streams from smartphones	Finalize Design	Save video streams	Object detection	Proactive Features	Refactoring as Needed	Project Clean Up	Finished System
Design Draft	Gather project equipment	Basic web UI	Communication between systems	Authentication	Get/view streams from IP cameras	Security	Tested System		
Research		Motion detection		User role added					
		Set up server							

Figure 2. Project Timeline

3. Statement of Work

3.1 Previous Work And Literature

Dube, Ryan. "How to Build a Security Camera Network Out Of Old Smartphones." MakeUseOf, 12 July 2017, https://www.makeuseof.com/tag/how-to-build-a-security-camera-network-out-of-old-smartphones/.

This article describes how to build a security camera network out of old smartphones. This is the same problem statement as our senior design project. This author talks about information such as placing the smartphone, setting up the camera IP monitoring station, features of his security camera network and how to create a surveillance system.

Chen, et al. "An Implementation of Faster RCNN with Study for Region Sampling." ArXiv.org, 8 Feb. 2017, https://arxiv.org/abs/1702.02138.

This is a paper written by two Computer Science professors under the topic of Computer Vision and Pattern Recognition. In this paper, the authors talk about how they adapted the join-training scheme of Faster RCNN framework from Caffe to TensorFlow as a baseline implementation of object detection. We are also planning to also use TensorFlow as a programming language in object detection.

Similar Smartphone Security Application "AlfredCamera." AlfredCamera, https://alfred.camera/.

AlfredCamera is a home security smartphone security application that is already on the market that has the goal of making security free and easy for everyone. To use AlfredCamera, you need to download the application on two smartphones, sign into your gmail account on both smartphones, set one camera as the viewer and the other as the camera to enjoy the application.

3.2 Technology Considerations

The major trade off we made was choosing a local solution rather than a cloud based solution for our services. The reason we made this choice is because our product is intended to be cheap and easily implementable for potential users. If we had chosen a cloud based solution the overall cost would be a lot more. We would have to pay monthly fees for cloud based infrastructure which would essentially make our product irrelevant as the entire point of this project is to be an economic solution to a problem which has been addressed many times. We want to create a solution that can be used by just owning a simple computer and a few old cameras or smartphones.

3.3 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

3.4 Possible Risks And Risk Management

Currently we have hardware limitations as we do not have a machine which can serve as a server for our project. We need to obtain a server to we can proceed in our project.

3.5 Project Proposed Milestones and Evaluation Criteria

10/15	11/01	11/15	12/01	12/19	01/20	02/15	03/01	03/15	04/15
Define project goals and requirements Design Draft Research	Begin Prototype Gather project equipment	Get/view streams from smartphones Basic web UI Motion detection Set up server	Finalize Design Communication between systems	Save video streams Authentication User role added	Object detection Get/view streams from IP cameras	Proactive Features Security	Refactoring as Needed Tested System	Project Clean Up	Finished System

3.6 Project Tracking Procedures

We will update which requirements have been met within using gitlab issues. This allows team members to see who is working on what and see what has been accomplished.

3.7 Expected Results and Validation

The desired outcome for this project is to create a home security solution which is easy for our users to set up and doesn't cost much for them to implement.

It is critical that we meet our non-functional requirements based on time performance as this is a home security system and we cannot risk have major latency delays.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

4.2 Feasibility Assessment

Realistic projection of what the project will be. State foreseen challenges of the project.

4.3 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just "X" hours per week for the number of weeks that the task is active

4.4 Other Resource Requirements

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

4.5 Financial Requirements

We have a total budget of $200 to spend on any project costs for the duration of 1 year given to us by our client.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

We will support any mobile device as a client that features the following softwares and above - Android 4.4 and iOS 11. To ensure a successful product, we have to test on the following softwares and any versions released after.

5.2 Hardware and software

- Indicate any hardware and/or software used in the testing phase
- Provide brief, simple introductions for each to explain the usefulness of each

We will use a mockito testing framework to test our API. This will be useful as mockito allows you to make service calls and create instances of different services. This we can ensure that data is transmitting properly.

5.3 Functional Testing

Test to see that Users can be added to the database.

Test to see that Motion Detection notifications work.

Test to see User authentication works.

Unit testing on all modular functions we develop on frontend and backend.

Test our server to see if it can process multiple streams.

Test to see our bitrate on streams matches requirements.

5.4 Non-Functional Testing

Test to see if notifications to user are fast.

Test to see if Users can access streams in a quick time(5 seconds)

Test front end facing applications responsiveness.

5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

6. Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 Appendices

Not yet applicable