DashCam Defender

Design Document

Team number 11

Professor Joseph Zambreno

Evan Timmons - Team Leader

Cobi Mom - Chief Mobile Engineer

Danny Yip - Test Engineer

Scott Vlasic - Head Hardware Concept Engineer

Durga Darba - Data Architect

Ismael Duran - Full Stack Engineer

sddec20-11@iastate.edu

https://sddec20-11.sd.ece.iastate.edu

Table of Contents

Executive Summary

Development Standards & Practices Used

Software practices

- Make code correct first and fast second
- Always test your code
- Agile Methodology
- Integration with hardware
- Simple design
- Pair programming
- Continuous integration

Hardware practices

- Integration with software
- Always test and make sure it is working
- Agile Methodology

Summary of Requirements

- Knowledge in mobile development
- Hardware development
- Database Design

Applicable Courses from Iowa State University Curriculum

- ComSci 309
- CPRE 288
- COMS 228
- SE/CPRE 185
- COMS 363
- CPRE 310

New Skills/Knowledge acquired that was not taught in courses

- Hardware Assembly
- Dart Programming Language
- Flutter Framework
- Hardware / Software integration
- API Calls

1. Introduction

1.1. Acknowledgement

We acknowledge Dr. Joseph Zambreno as our senior design advisor.

1.2. Problem and Project Statement

The roads as we know it are full of reckless drivers. What if there was a way to automatically detect when you are near a reckless driver? The solution that we are proposing is the ability for everyday consumers to come together and crowdsource information on reckless drivers. This is done by utilizing DashCam Defender, a product designed by us to automatically scan license plates and report reckless drivers when they are encountered instantly.

1.3. Operational Environment

The end product will be used in a car where it will not be exposed to extreme conditions. Therefore, we will have to construct a container to protect the device from consumer damages.
Another condition we must consider is the camera's ability to capture data given poor weather. To ensure this, we will have to use a camera that returns images in 1080p or higher.

1.4. Requirements

For the automated license plate reader we need a machine learning program to accurately read license plates. The camera that will be needed is a dash camera that can record in 1080p. A mobile application will be developed for the consumer to interface with the machine learning program/dashcam.

1.5. Intended Users And Uses

We have three major intended users for our product:

- Police Departments
  - Police can lookup license plates and their last known locations to solve crimes faster
- Everyday Drivers
  - The everyday driver can report poor drivers and get alerts to notify them of when they are in the presence of a poor driver.
- Insurance Companies
  - Insurance companies can use driver ratings and dashcam footage to adjust their rates and claims

1.6. Assumption And Limitations

Assumption:
The end product will be able to work under a certain degree of poor weather. It will not be able to work if the weather is too bad, that the camera is not able to capture any clear picture. The end product will be used in the United States.

Limitations:
The end product will be a dashcam and a mini pc in the client's car, completing the Dashcam Defender apparatus. The cost to produce the end product shall not exceed one hundred and fifty dollars. The system will not require more than 12 Volts. (The most common voltage found in an Automobile Auxiliary Power Outlet)

1.7. Expected End Product And Deliverables

The expected end result is that we have a functional product that is able to attach to a car. It will be able to read license plates accurately and also send data from the users phone.

2. Specifications and Analysis

2.1. Proposed Approach

Our proposed approach is to find existing resources that will help us fill in for things that we need. A mobile application will be developed utilizing the Flutter Framework. The camera and mini PC have to work together and the other half of the team will work on that with existing resources found. So far we have found an open-source automated license plate reader that we are going to use. We will purchase a camera and a NVIDIA Jetson computing board for the ML computation.

2.2. Design Analysis

Currently we have a mock database to test the connection between the hardware and the software. From here, we need to move into figuring out how to organize and send the information to fill the tables in the database. Our final version of the product will be doing this automatically, so we need to ensure the information can be processed efficiently. On the hardware side, our goal is to be able to read license plates at a rate of 5 per minute. On the software side, the program has to be able to multithread efficiently enough to process each license plate and store the information separately. The main strength of the project is in the database and the pushing and pulling of the data from it. The weakness, currently, is in the way we are pushing the data when the device doesn't have easy access to the internet.

2.3. Development Process

We are going to be following the Agile Methodology for our development processes. Specifically, we're going to split up into two separate teams of three; one team focused on software and the other focused on hardware. We think Agile is the best choice for the development process because it will allow us to create small goals that are attainable as well as foster communication and collaboration within each group.

2.4. Conceptual Sketch

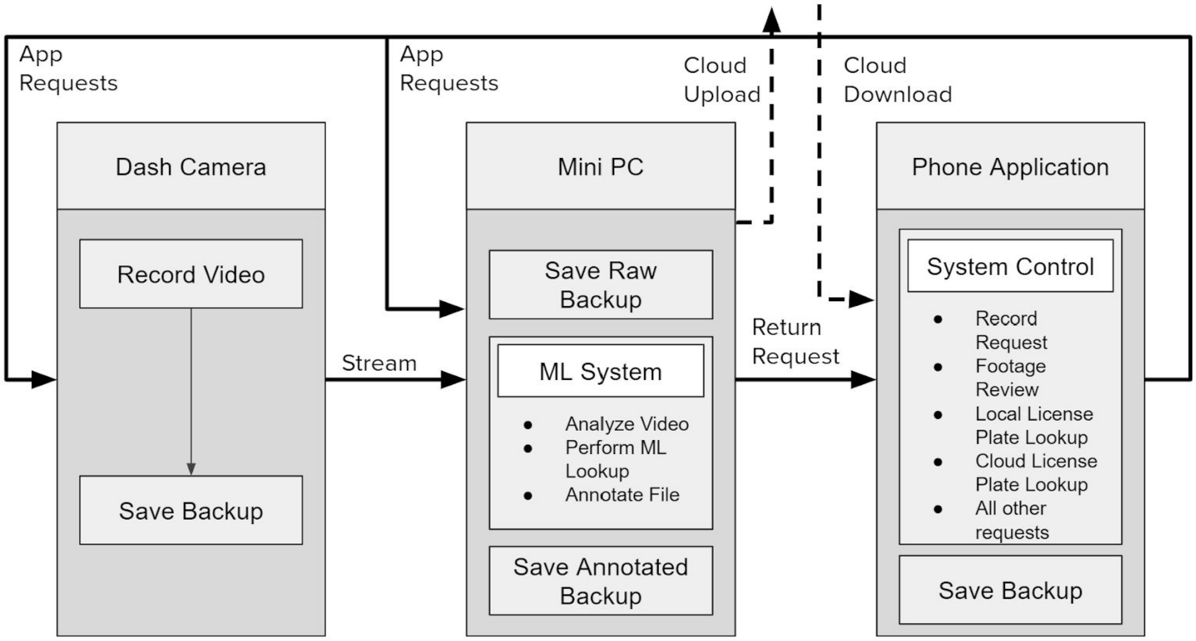

Figure 1.1: System Level Conceptual Sketch made by Evan Timmons

3. Statement of Work

3.1. Previous Work And Literature

Existing open source technology: OpenALPR | Existing company: Rekor Systems

3.2. Technology Considerations

Limited computing power available
Very expensive existing technology
Lack of reporting features
Not User friendly software

3.3. Task Decomposition

Hardware

Camera and computing system

Software

Database and mobile application

3.4. Possible Risks And Risk Management

Causing driving accidents by distracted drivers

3.5. Project Proposed Milestones and Evaluation Criteria

Functional hardware
Functional software
Integrated Hardware and software
Fully functioning Prototype
Aesthetic Implementation

3.6. Project Tracking Procedures

Trello for tracking work and Google Drive for tracking documentation.

3.7. Expected Results and Validation

A fully operational system allowing at least one user to report cars with videos based upon their license plates

4. Project Timeline, Estimated Resources, and Challenges

4.1. Project Timeline

Insert Gantt chart here with timeline
Will include dates for:

Functional hardware
Functional software
Integrated Hardware and software
Fully functioning Prototype
Aesthetic Implementation

4.2. Feasibility Assessment

Realistically the final prototype may not fully integrate with the software by the end of the project, however both sides of the project will likely be working individually

4.3. Personnel Effort Requirements

Insert table here with tasks and timeline requirements

4.4. Other Resource Requirements

Other resources include, a car, a driver, and a 3D printer.

4.5. Financial Requirements

~$500 as detailed in an ongoing parts spreadsheet

5. Testing and Implementation

5.1. Interface Specifications

We are going to use black box testing on our system if some part is not complete yet

5.2. Hardware and software

We will decide to choose the best testing hardware and software once we have some part done.

5.3. Functional Testing

We are going to test all the functionalities for our mobile application and website

5.4. Non-Functional Testing

We are going to test and make sure it supports a certain number of users at the same time

5.5. Process

We are going to record our tests as much as possible.

5.6. Results

Not applicable yet

6. Closing Material

6.1. Conclusion

In order to help drivers on the road, we need a better way to hold reckless drivers accountable. With our app and dash cam, we can do this. Furthermore, we can help insurance companies and the police get a better idea of people on the road. By meeting our specifications through the milestones we've set, we will be able to finish this project in a timely fashion.

6.2. References

OpenALPR software - Will implement in IEEE style in coming weeks

6.3. Appendices

Not Applicable Yet

List of Figures

Only figures used so far, made by Evan Timmons. Used in section 2.4 "Conceptual Sketch"
Additional figures will be cited here when used