Project Plan
for
Beep.Brake

Table of Contents

- Overview
- Goal and Scope
  - Android App
    - In Scope:
    - Out of scope:
  - Web App
    - In Scope:
    - Out of scope:
- Deliverables
- Risk Management
- Scheduling and Estimates
  - Semester 1
  - Semester 2
- Measurements and Metrics
- Technical Process
  - Tools
  - Artifacts

Overview

Beep.Brake is an android app focused on reducing car accidents through forward collision warning. The phone is mounted behind the windshield with the camera facing the road. Video analysis shall be performed to determine objects in the vehicle's path and determine which lane the user is driving in. If a warning is triggered video data is recorded with time, date, location, speed, brake pedal status, x-y-z acceleration and direction of vehicle.

Beep.Brake is for beginner and experienced drivers looking to stay safe as they travel. The application is specifically designed to run a low end androids to increase the accessibility of the app. The application will be designed to use a no cell data therefore not interfering with the user's cell phone plan.

Schedule
Development responsibilities

Goal and Scope

There are two primary parts to the end-goal of this project: an Android app and a web app. The Android app is what will perform forward collision warning, through the use of GPS, camera, and other sensors if available. For each warning issued, a 12 second datalog of all sensor data will be cataloged and sent to a web server. The web app will access this cataloged information and display events and general statistics.

Android App

In Scope:

- Software shall run on Android 4.4 (Kitkat)
- System shall identify objects in the vehicle's forward path
    - Cars and Pedestrians
- Estimate distance to object
- Estimate time to collision
- Issue audio/visual warning if the object is too close or too quick
- Warning sensitivity is adjustable
- System shall have adjustable thresholds for warning time and related detection parameters
    - Size of the object
    - Lateral clearance
    - Warning time

    - Number of lanes
- System shall capture a 12 second video
    - 6 seconds before warning
    - 6 seconds after warning
- Datalog shall record 12 seconds with 100ms resolution
    - Time, date (recorded once), GPS position, speed, brake pedal status, x-y-z acceleration, heading of the host vehicle each time the system issues a warning.
- Data shall be pushed to a web server (via cellular data or wifi)
    - Warnings made
    - Datalog
    - 12 second video of the warning
- Interface with OBD-II via bluetooth for additional detection parameters

Out of scope:

- Barometric pressure (weather conditions etc.)
- Lane Departure

Web App

In Scope:

- Full authentication for Analyst, Application
- Save data pushed by the android app
- View video recording with warning information
- Provide analysis metrics such as number of warnings in a location, number of warnings responded to by braking
- Accept data only from authorized application users

Out of scope:

- App users viewing their own log data

Deliverables

- Project Website Sep. 14, 2015
- Project Plan (this document) Sep. 19, 2015
- Domain Model Sep. 19, 2015
- Midterm Peer Eval Oct. 11, 2015
- Interim Project Presentation Nov. 30, 2015
- Project Plan update Jan. 29, 2016
- Final Presentation May 2, 2016
- Final Technical Report May 16, 2016

- Final release May 16, 2016

Risk Management

See Risks stand-alone document.

Scheduling and Estimates

Semester 1

Name: Scope Clarification
Estimated Completion Date: 9/14
Details: The team will have the projected scope for the entirety of the project defined

Name: Demonstration of OpenCV capability on Android
Estimated Completion Date: 9/29
Details: To understand if the projected project scope is feasible, the team needs to investigate the hardware and ensure that the expected behavior is possible on the provided hardware.

Name: Application Capturing Frames
Estimated Completion Date: 10/5
Details: The application will be able to capture frames that will be available for the hardware to examine and run potential image analysis algorithms on.

Name: Vehicle Recognition
Estimated Completion Date: 11/16
Details: The application will be able to discern vehicles from frames that it has recorded. Given an image of a <generic car> it will be able to identify the boundaries of the recognized vehicle. The application will beep when it has recognized a vehicle.

Name: Web Server and Database creation
Estimated Completion Date: 1/25
Details: The web server and database will be created and exist. No model or information is stored in the system at this point.

Semester 2

Name: Data Model creation
Estimated Completion Date: Week 2

Details: The data model for both the web server and internal android storage will be created and in place. This will allow for us to be ready to push data at any time directly into either form of storage.

Name: Lane Detection
Estimated Completion Date: Week 3
Details: The phone application will be able to recognize which lane of traffic the user is driving in.

Name: Web API creation
Estimated Completion Date: Week 4
Details: Endpoints for the web service will be available for the android application to hit. Functionality behind the endpoints is not required.

Name: Vehicle Range Finding
Estimated Completion Date: Week 5
Details: The application can determine how far a recognized object is from its current point given two or more images.

Name: Data Storage Functionality
Estimated Completion Date: Week 6
Details: The server can process data pushed to it from the API and properly store it in the database.

Name: Time to impact calculations
Estimated Completion Date: Week 7
Details: The application can determine the probable time to impact from factors implemented into the processing.

Name: Collision Warning Algorithm
Estimated Completion Date: Week 12
Details: The application will alert the user if a collision is imminent in a timely manner (according to our research)

Name: Database Data Retrieval
Estimated Completion Date: Week 12
Details: A user on the front end of the web service can retrieve individual events from the collision warning database.

Name: Configurable Application Warnings
Estimated Completion Date: Week 13
Details: Settings on the application can be personalized per user to augment when the application will warn them of an impending collision.

The team will work as a whole in order to reach the first milestone for the android app, and break into two teams when the first milestone is completed. The app team will continue forward on the Beep.Brake application while the web team will begin work on the server and database.

Measurements and Metrics

Measurements

- Fourup chart
- Hours logged
- SLOC
- # Of bugs
- Cyclomatic Complexity
- Camera Frames analyzed per second

Metrics

- Test Coverage
- Function Points
- # of bugs per SLOC
- Avg time to detect lane
- Avg time to detect object
- Garbage collection monitoring

Technical Process

The project sponsor wants to ensure that each completed feature is thoroughly tested and functional before the team moves forward to the next feature on the system. For this reason, the team has chosen an incremental development process that allows us to build each feature based off of a fully functional base system.

Tools

- Relational Database (TBD) - app local database and web aggregate database
- Android - Standard OS usage
- OpenCV - Image capture and Image Processing
- IntelliJ/Android Studio - Integrated Development Environment for Android development
- Bluestacks - Fully functional Android emulator
- Web Server (TBD) - Serve up pages
- Trello - Task coordination and tracking
- GitHub Issues - Issue tracking

Artifacts

- Four Up Chart
- Timesheets
- Requirements Document
- Domain Model
- Software Development Plan
- Project Website
- Project Presentation
- Project Poster
- Technical Report