

Pothole Detection

Yagnahaun Jonnadula

Anurag Peddi

TOC

Introducing: Pothole Detector

- Problem Statement
- Development Process
- Likely Impact
- Team Size

System Analysis

- Stakeholders
- Competitive Analysis
- Requirements

Detailed View

- Software Subsystems
- Hardware Subsystems
- The Image Processing Scheme
- Brief Overview Of The Model
- System UML Diagrams

Test Plan

- Automation
- Testing

Management Frameworks

- Change Or Control Management
- Packaging
- Build

Risk Management

Conclusion



Introducing: Pothole Detector

Problem Statement

The goal of our project is to design a Pothole detection System which assists the driver in avoiding potholes on the roads, by giving the driver prior warnings. For example, it can be like a buzzer or series of LED, if the driver is approaching a pothole driver may be warned regarding the pothole on the road. In this, we propose a robust and straightforward design of a portable and affordable device that can alert the driver about the detected pothole. The hardware system installed in a moving vehicle can automatically detect and report potholes via image-processing of Raspberry-Pi microcontroller. The detailed images of the pothole and its location are stored and viewed through the GOOGLE API.

Index Terms—Pothole Detection, Hardware system, Raspberry-Pi microcontroller, GOOGLE API.



Likely Impact

01

In this fast-moving world, people want to reach their destinations as soon as possible. Some apps suggest the routes which can make us reach our goal early, and other apps show all possible ways with traffic congestion for whatever reason it may be, such as google maps and many more.

02

But there are fewer apps that tell the condition of the road, whether it is good to travel or not, whether it suggests the route to the driver based on the state of the way. So this model mainly focuses on the travel safety of the passengers and updates the passengers with the best route to travel.

03

Due to weather conditions, improper construction and overloading of vehicles the roads are getting damaged. The scope of the project lies, where the irregularity of the public roads affects the people



Development Process



The strategy utilized to develop the module is Agile model.

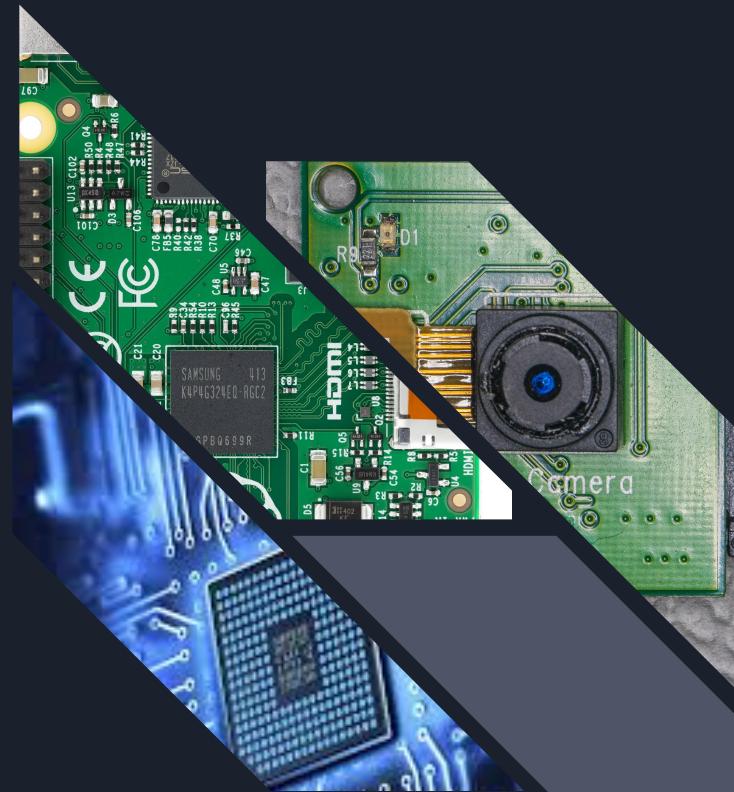
Agile Model was selected for the following reasons:

- It allows for an improved quality of the product
- It focuses on the user reviews about the product
- It involves stakeholder engagement with the team
- It provides a unique opportunity for clients or customers to be involved throughout the project
- Another key benefit of an Agile methodology is that, unlike a Waterfall model, it allows for change in the functionality of the model

Team Size

The team can be broadly classified into 6 categories:

- Software Designer and Developer
- Project Team
- Database Administrator
- Change Control Board
- Testers
- Analysts





System Analysis

Stakeholders

The below are the stakeholders taking part in the total scenario:

- User (Driver)
- Data Analyst who takes the data provided by the devices and perform some data analyst.
- Software Developer who debugs any issues after the installation.
- Project Manager who manages the project.
- Tester who tests the model.
- Change control board.
- Project team.



Competitive Analysis

People in this time need instant solutions for problems; one of the issues is that they need to reach their destinations early, and many applications can do this work. Still, what our app does differently makes us stand out from others, it not only suggests which road reaches your destination faster but also indicates which route is most safe to travel.

The work which can be done by our app is the most critical work, which makes it to find its way into the minds of the users.



Requirements

The embedded device processes the incoming stream of video and detects any pothole in the flow, if any, it immediately intimates the user with a signal. It also saves the information of pothole identified and coordinates in a file for cross-validation.

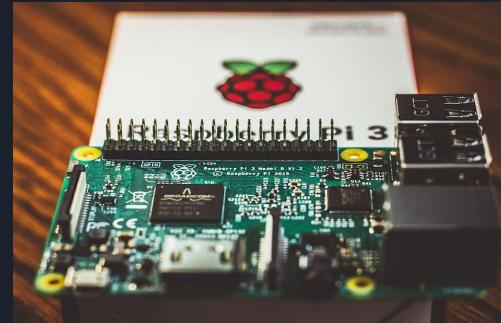
- 01 The system should be restarted every 6 hrs, for the cache to flush and the pi camera should be replaced every 5 years for the model to work properly.
- 02 The user need not to provide any of their confidential details for the module to work.
- 03 For the module to work, the car should be running and should also limit their speed to under 40kmph.
- 04 As we are using a pre-trained model for detecting the potholes, the time constraints will be on alert the driver about the pothole and suggesting to him the best and safe possible route.



Detailed View

Hardware Sub-System

- Raspberry Pi 3
- Raspberry Pi 5MP Camera Module
- Mini Night Vision Camera 1080P
- Jumper Wires



Raspberry Pi 3

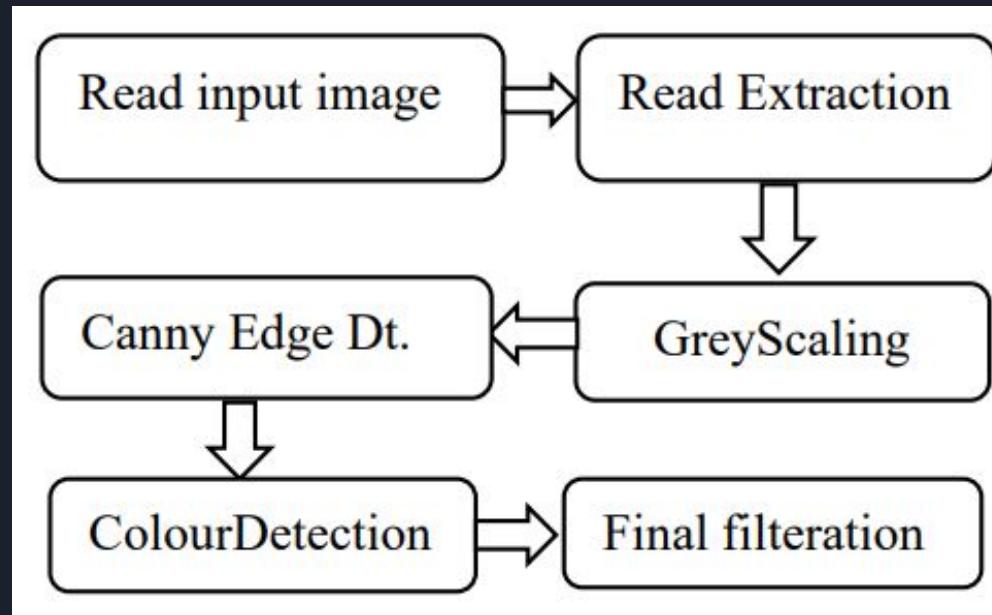
Software Sub-System

- AWS CloudBuild
- AWS Config

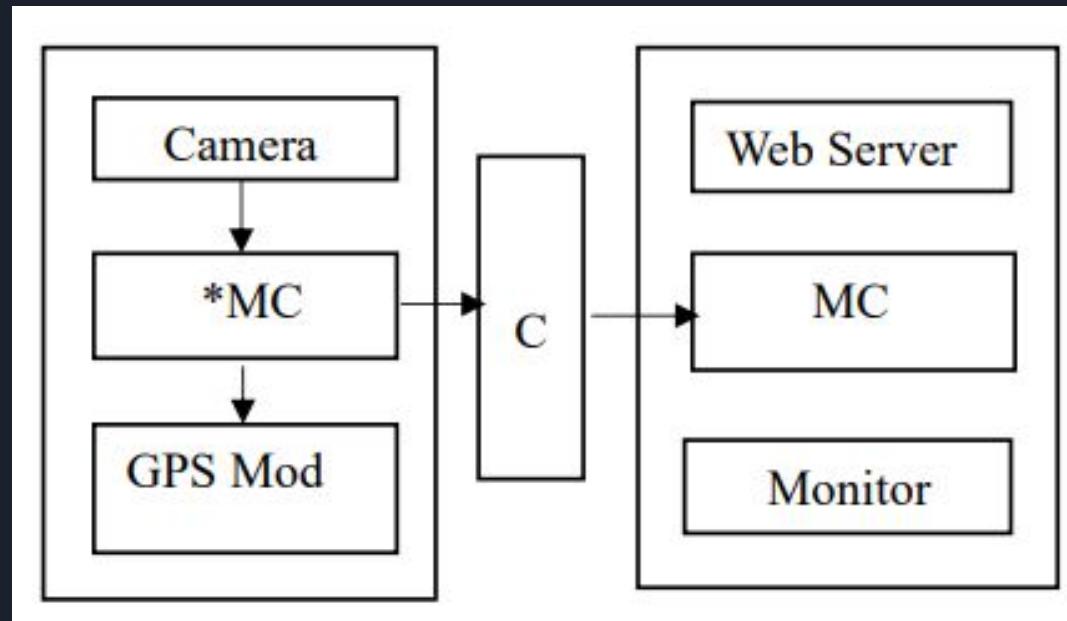


AWS

The Image Processing Scheme

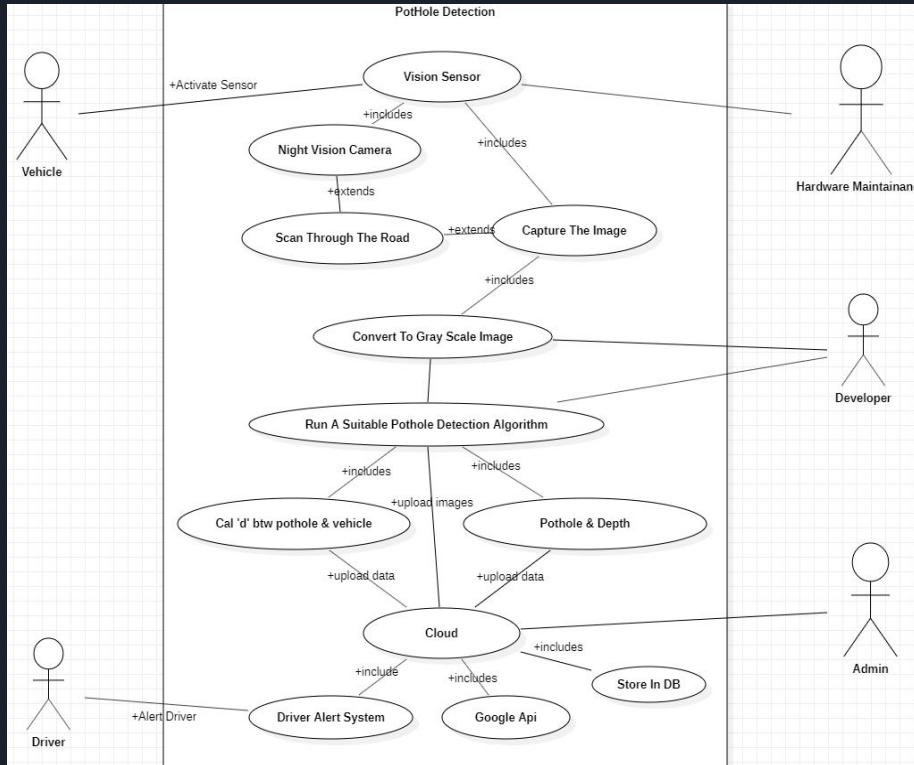


Brief Overview Of The Model

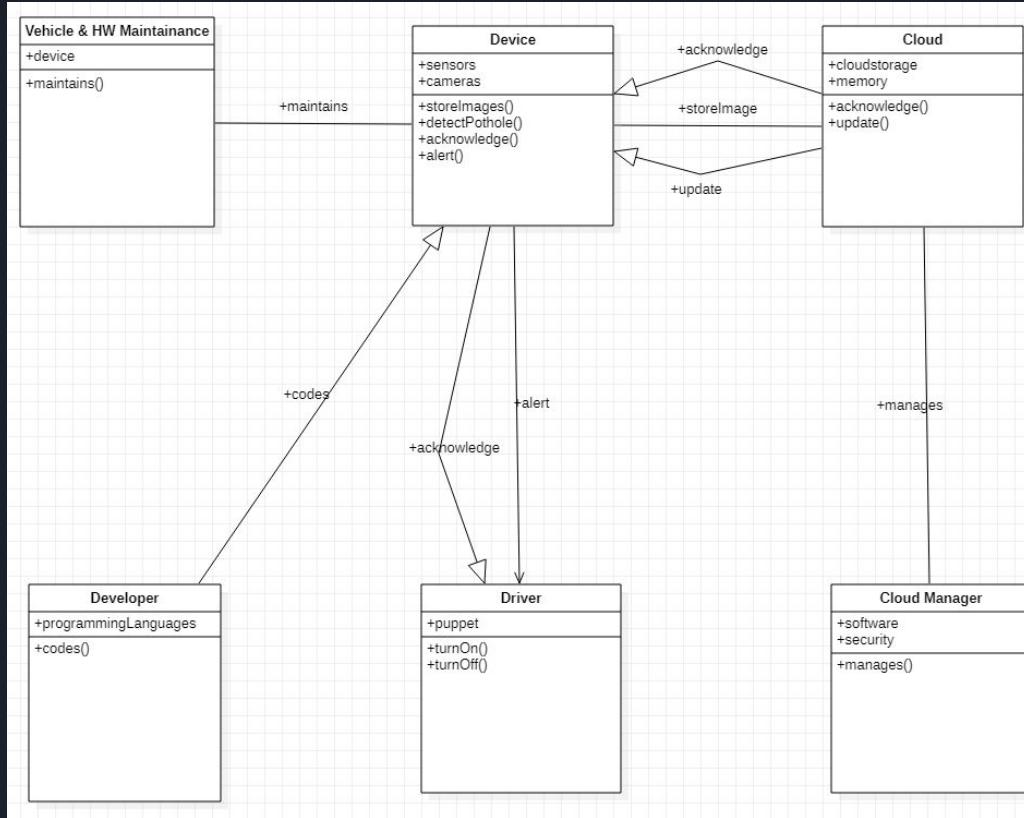


System UML Diagrams

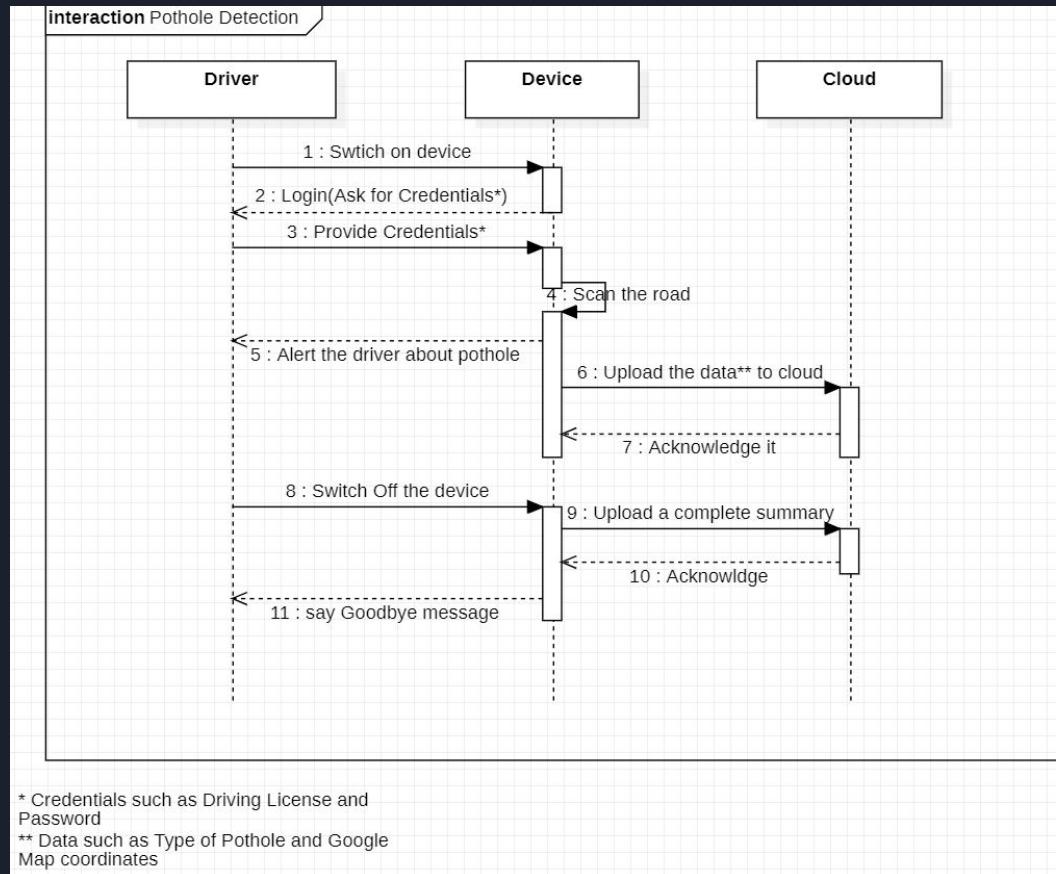
- Use Case Diagram



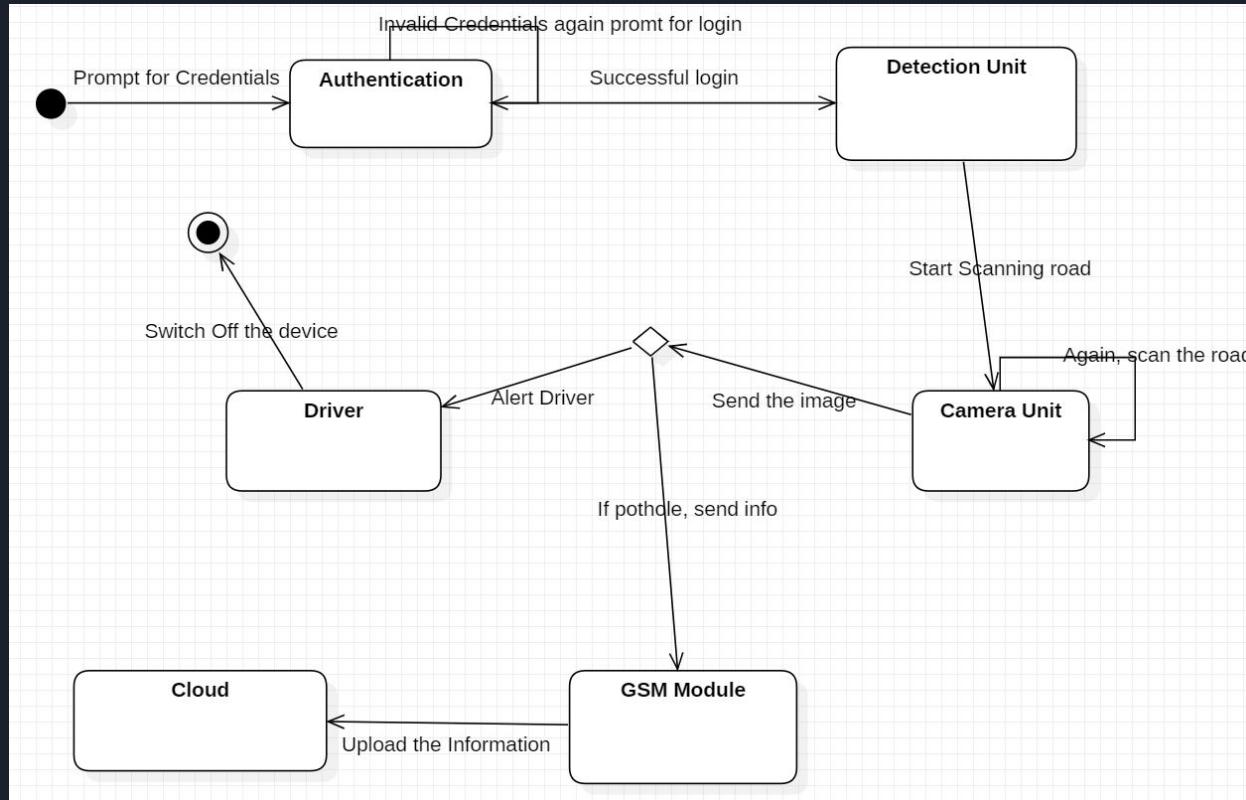
● Class Diagram



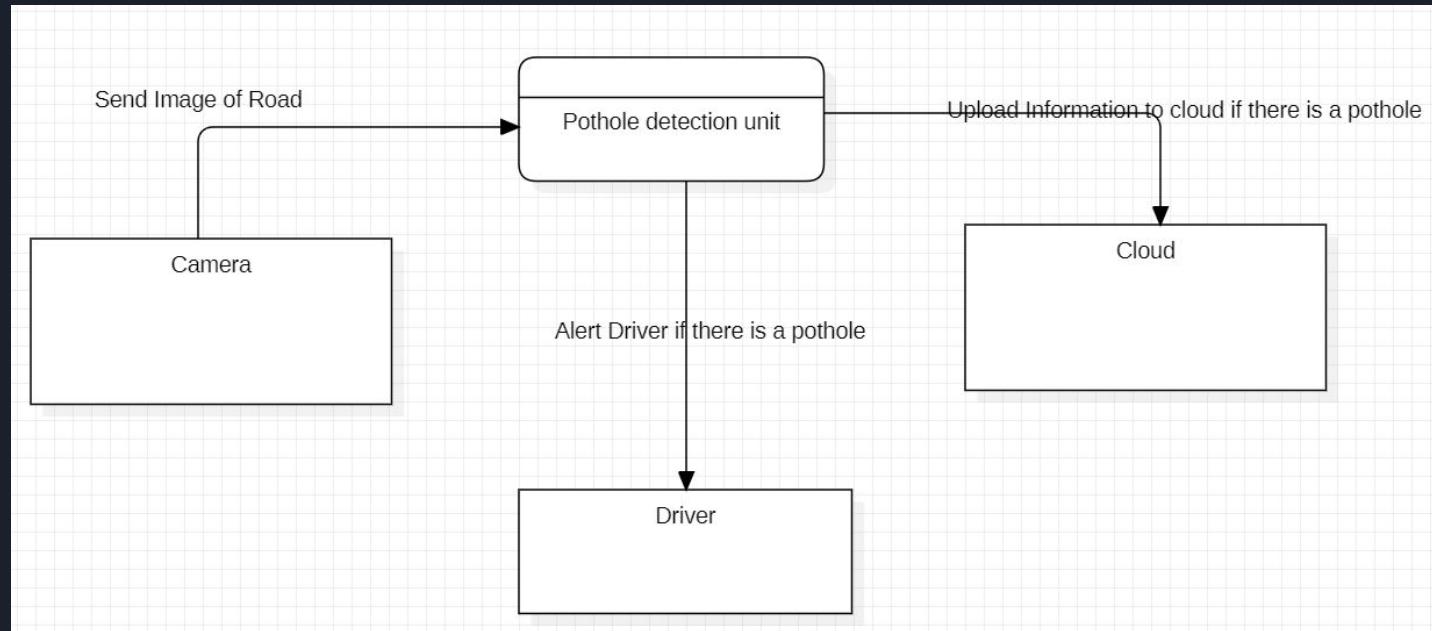
● Sequence Diagram



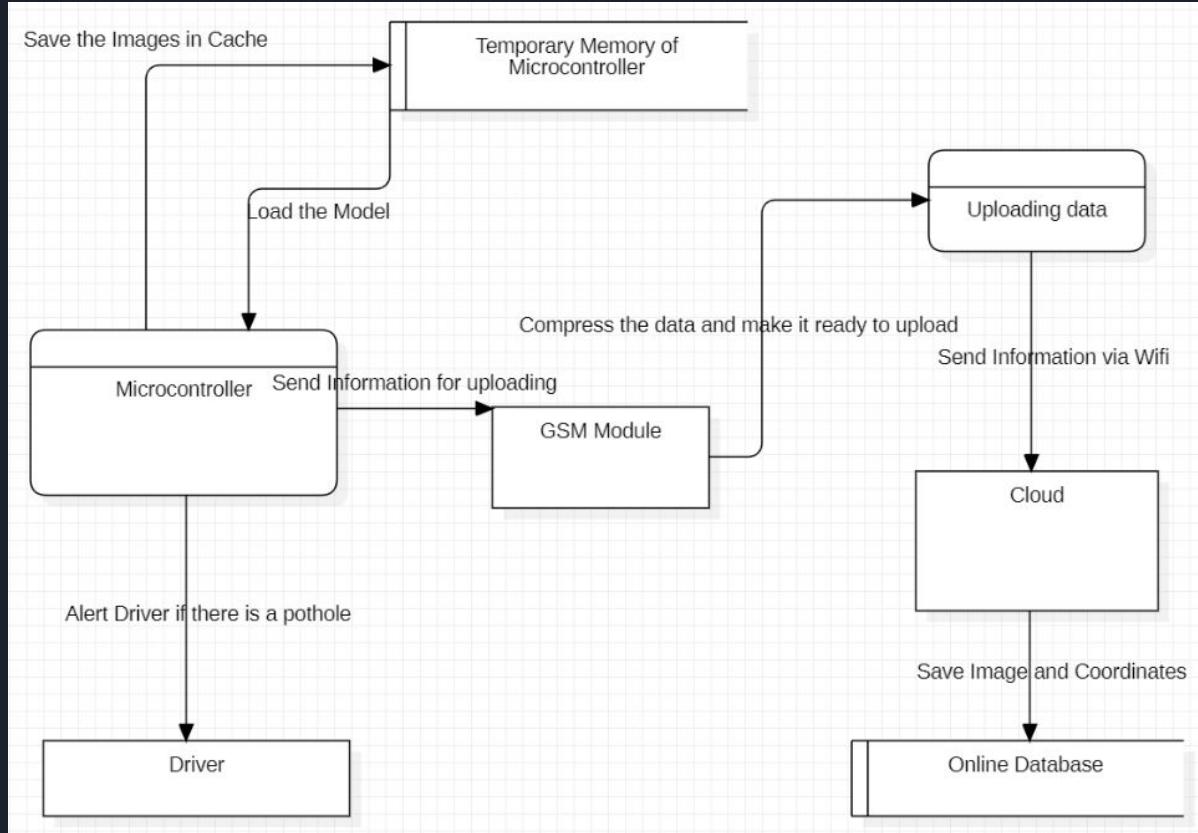
• State Diagram



- **Level 0 Data Flow Diagram**



- **Level 1 Data Flow Diagram**



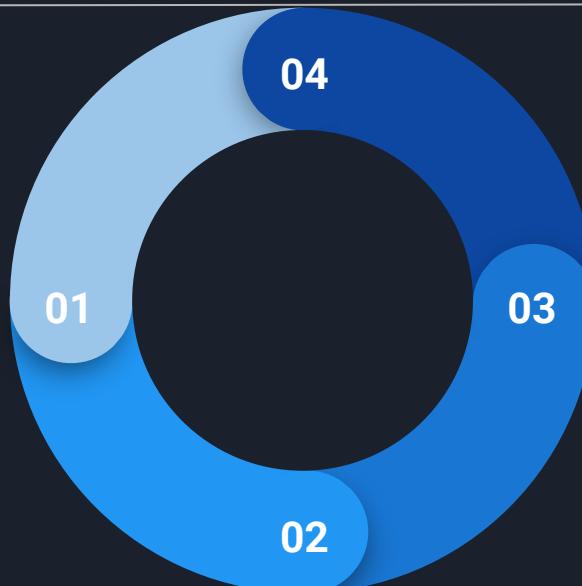


Test Plan

Automation

When the device is switched on, the device scans through the road.

If the device finds any pothole it runs through the algorithm programmed in it.



This device assists the driver in avoiding potholes on the roads, by giving the driver prior warnings.

It uploads the location and image of the pothole to the cloud



Testing

- **Unit Testing**

Software test automation tools enable you to simplify testing and reduce time to release by automating functional tests for your applications. Unit testing tools which we are going to use are:

- Pytest
- Travis CI



Travis CI



Pytest



- **Validation Testing**

For validation testing, we use the images which we have collected from sites to check whether all the scripts are bug-free. This validation testing is also done out during the demonstration to our stakeholders(excluding developers).

- **Defect Testing**

For defect testing, we record some real-world videos of road and present it, which not only includes images of the road but also pictures of some patterns which are similar to potholes, to measure how accurate it is detecting potholes when both potholes and potholes like patterns are present.



Management Frameworks

Change or Control Management

Replacement of raspberry pi should be made once every five years and additional updates every month. AWS Config is a service that let's assess, audit, and evaluate the configurations of AWS resources which we are using.

Build

The version control system which we are using is git. AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy.

Packaging

To make the python scripts standalone, we are using a package called pyinstaller. AWS packaging automates the process of packaging and publishing software to managed Windows and Linux instances across the cloud landscape, as well as to on-premises servers, through a single simplified interface.



Project timeline





Risk Management

Risks

The hardware risks which the device faces is external accidents/shocks, which are unpredictables example:

- Someone hitting it
- Snow covering the lens
- The camera being covered by some object(paper).
- Rainwater going into the circuits and frying them.

And the some of the software risks are privacy and security of data. Time to time updates about the potholes to the cloud and intimate the driver about it correctly . The data should be transmitted only in a secure environment.



Risk Mitigation

Hardware

The above hardware risks can be avoided if the device have a dedicated slot where it is safe enough and should detect the potholes as well.

Software

The privacy and the security of the device should be maintained by a good security analyst so that it would not be lost. And the software developer should schedule time to time checks about the cloud management and the software.



Conclusion

Challenges

The challenges in front are, the budget as we can't take that to a high budget model so that it can be used by many and should get a high pixel camera with a low budget and the model should not make false interpretations of the pothole and make the user feel sure about this model.

Possible Extensions

There could be even more possibilities of extending the model to a state at which it not only detects the potholes but also detecting the fixed potholes and updating the earlier data and detecting speed breakers where the drivers need to slow down. Another extension that we can make is to estimate the depth of the pothole and to categorize them to a different level of danger and alert the driver about that danger, whether he can travel on that road or not. And can be notified to higher officials for the fixing of these potholes.

- To make it budget-friendly.
- Reduce the latency between the uploads.
- Detect potholes even when the car is at a very high speed.

Thank you!

Our work can be found at

https://github.com/yagnahaunj1410/Pothole_Detection_Using_Image_Processing

And

<https://github.com/AnuragAnalog/Pothole-Detection>

