|  |
| --- |
| Name: **Rathnayaka Rathnayaka** |
| Student Reference Number:**10747919** |



|  |  |  |
| --- | --- | --- |
| Module Code: **PUSL2021** | Module Name: Computing Group Project | |
| Coursework Title: **Driver Rescue System Technical Specification Document** | | |
| Deadline Date:**2022/01/02** | | Member of staff responsible for coursework: **Mr.** **Pramudya Hashan Tilkaratne** |
| Programme: **BSc (Hons) Computer Security** | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.   * **Rathnayaka Rathnayaka - 10747919 -BSc (Hons) Computer Security** * **Bandara Yapa - 10749057 -BSc (Hons) Computer Networks** * **Rathnayake Rathnayake - 10747887 -BSc (Hons) Computer Science** * **Sandaruwan Rajapaksha - 10747909 -BSc (Hons) Computer Networks** * **Subasinghe Rajarama - 10747883 -BSc (Hons) Computer Science**   ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***    Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed : | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

\*Please delete as appropriateSci/ps/d:/students

**ASAP**

**Caller**

Driver Rescue System

Technical Specification Document

Contents

[Table of Figures & Tables 0](#_Toc98163139)

[Chapter 1 – introduction 1](#_Toc98163140)

[Chapter 2 - System Analysis 2](#_Toc98163141)

[Chapter 6 - Development tools and technology 3](#_Toc98163142)

[Development methodology 3](#_Toc98163143)

# Table of Figures & Tables

[Figure6 1 (Development Methodology for IOT) 3](file:///C:\Cloud\OneDrive%20-%20University%20of%20Plymouth\2nd%20Year\Computing%20Group%20Project\Project_ASAPCaller\Documents\3.%20Interim%20Report\PUSL2021_Interim_Report.docx#_Toc98181856)

# Chapter 1 – introduction

# Chapter 2 - System Analysis

# Chapter 6 - Development tools and technology

## Development methodology

During the project discussion we have decided to prototype the hardware side of the project by Arduino and Raspberry-Pie. We decided that because we wanted to check what is the most efficient and what is the most compatible with other requirement as well.

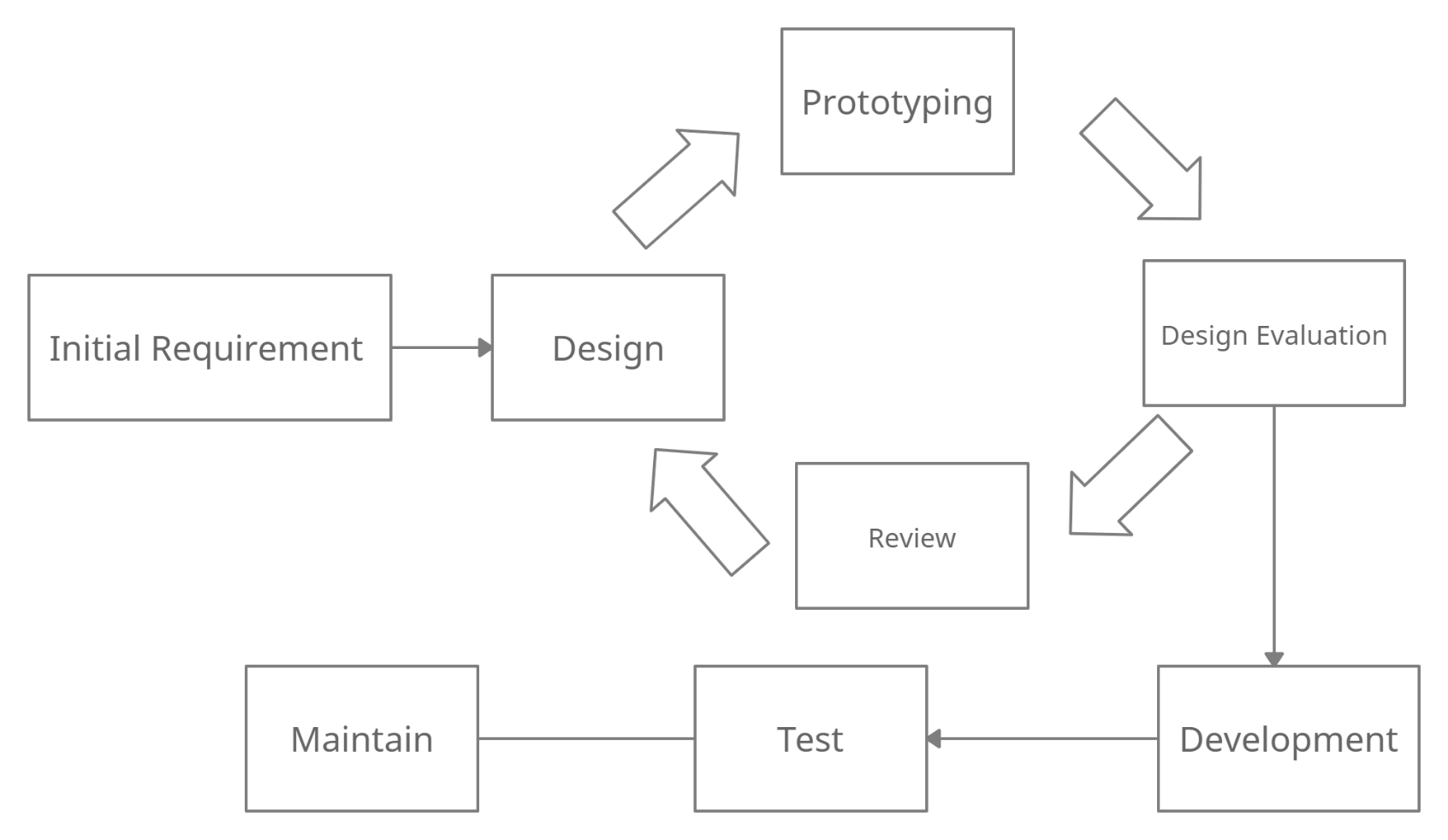
 During the prototyping stage we decided to implement Arduino based protype to the finale build because of it can be configured and directly integrated with Windows OS systems which our GUI of the project will run on and it is easy to configure with Windows Presentation Foundation Applications.

Figure6 1 (Development Methodology for IOT Device)

When it comes to the software development of this project, we use a different development approach. We decided to use a different approach because we wanted to develop the applications, without being affected by the hardware like Arduino.

Diagram

Description automatically generatedThis project comes with two applications. One is a Desktop application and other one is for Mobile. To work this project, vehicle must have a prebuilt computer runs Windows OS. Most of the modern vehicles comes with a prebuilt minicomputers that are mounted on the dashboard of a vehicle. This desktop application runs on that minicomputer.

Figure6 SDLC for application development

Both Desktop and Mobile applications are straight forward. They include UI and some triggers to detect any kind of unusual momentum of the vehicle like accidents and crashes. So, the approach is waterfall methodology.

Requirements for these two apps are simple. To send a message to the emergency services with the GPS location if there is no response from the driver incase of an accident. The optional features is to send a message to the driver when the drivers is outside, and something unusual momentum detected on the vehicle. There is no need to trigger the desktop application if the driver is outside.

To implement these applications, we do not have to wait until the hardware side which includes the sensors and control unit are build, to completed. The desktop application is build using Windows Presentation Foundation(WPF) and it includes the main triggers for the project. It will include the thresh values to trigger the event. When the event is triggered, UI will prompt to get the user response. If there is no any response, the WPF application will activates the GSM Model to send a massage to the emergency services with location where it is triggers. When it comes to the mobile application, it will only triggers when the driver is outside the car. The scenarios are vehicle is hit by some other vehicle when it is parked or any attempt to a thievery is happening. The sensors install in the device will detect this kind of motion and alert the driver by sending message to the mobile application. These are the only requirements to be implemented in the project. There will be not any modifications to the project. So, the best approach to develop the software are by waterfall model.

## Programming languages and tools

The software development tool that is being used mostly is Visual Studio. Most of the main functionality is included in the WPF application, so it is built on Visual Studio. The language is being used in WPF code behind is Visual C#. To develop the GUI for the WPF, XAML is used. XAML is an Extendible Application Markup Langued that is used in many Windows environment. By using XAML developers get access to fully customize any element that are used in application. The framework used by WPF is .NET.

When it comes to the functionalities, C# is the programming language is used in WPF. All the business logics and functionalities are implemented by using C#. C# is an Object-Oriented Programming Language, so we can easily handle all the models used in the application.

All of these are used in Visual Studio Environment. Visual Studio is a user-friendly developer tool that comes with many features to use during the development.

When it comes to the Mobile Application Development, we have decided to use Kotlin programming language. Kotlin is also an Object-Oriented Programming Language. So, design of the mobile application is reliable as WPF. Most importantly, Kotlin is native for android development, so we can integrate and use to all the functionalities of an Android mobile phone. Kotlin is now Google’s favorite Android development language. To develop the mobile application using Kotlin is done by Android Studio.

We are using Arduino Controller Unity to read the sensor values. So, C++ language is used to develop Arduino program to read sensor values. This program is developed in Arduino IDE.

## Third Party Components and Libraries.

One of the sensors we are using in the project is a \_\_\_sensor. The sensor is MPU6050. To read the values from that sensor, we have used a library call, MPU6050.h. It is developed by \_\_\_\_. This library will do most of the calculation to get somewhat meaning full values to use in the project. This library will return raw values of current momentum of the sensor.

## Algorithms

The only algorithm that will be used in this project is to get a meaning full reading from the raw data that are returning from MPU6050 library. We do a small calculation to get the current momentum of the device and send the readings to the WPF application.

## Hardware tools

In this project, we have used Arduino as the controller unit, a \_\_\_\_ sensor, a shock detection sensor and a GSP module. As discussed in the pernicious topics, these are the only hardware tools that are used in the project.

### Arduino Board

The Arduino board we are using is Arduino Uno. This board is enough to integrate all the sensors and modules to work with.

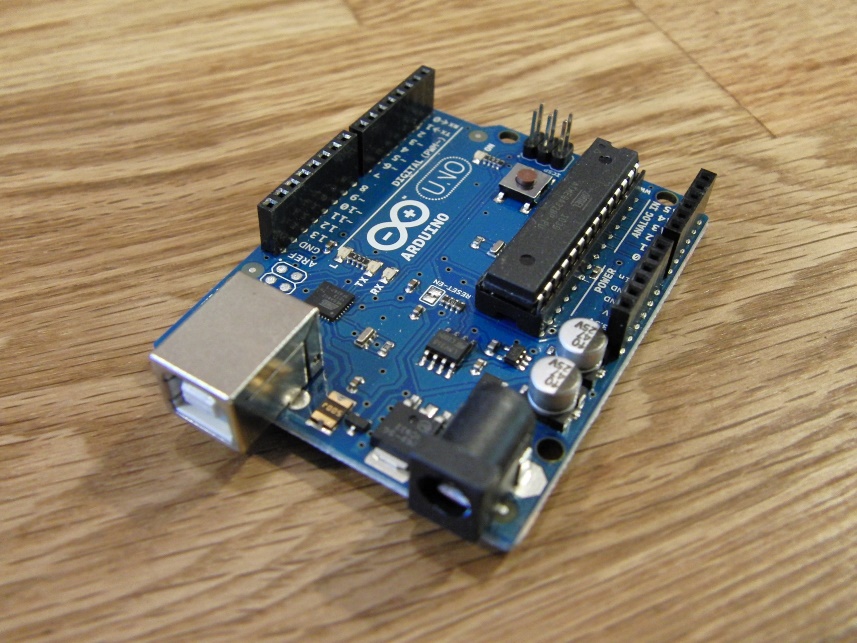
For any optimization purpose, we might be using ESP8266 NodeMCU

Figure6 4 ESP8266

Figure6 Arduino Uno

### MPU6050 Sensor

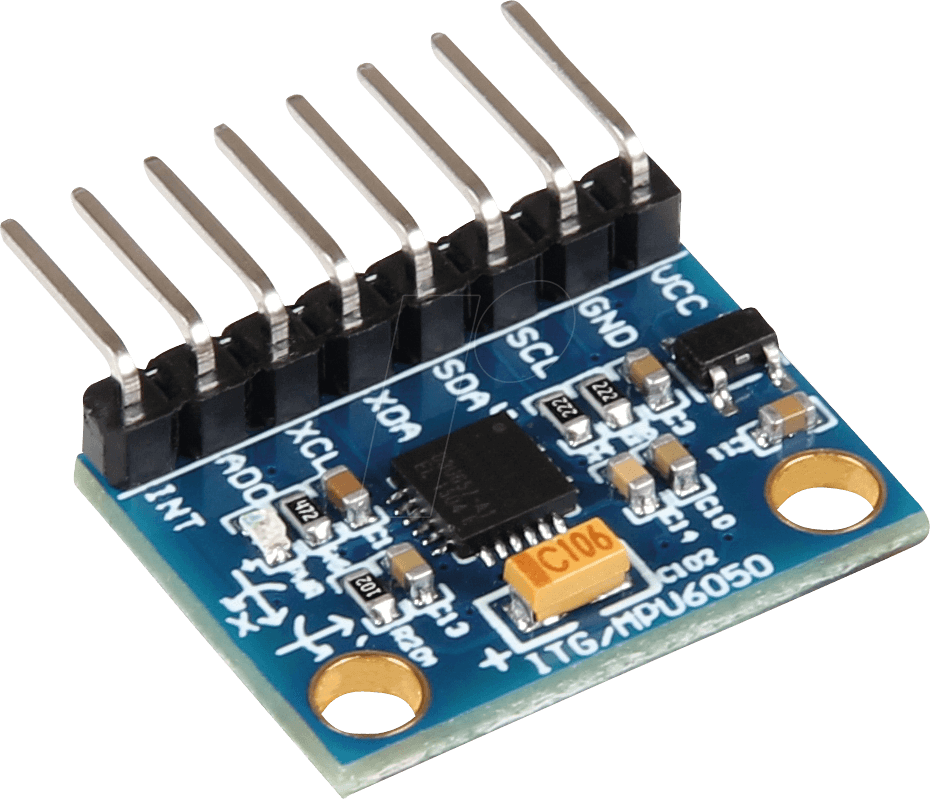
 This sensor detects any rotation of the sensor. This is used to detect if a vehicle is flip over by an accident. Sometimes a shock might not happen to detect by the shock sensor. So, this sensor is more like a backup sensor. This sensor will read the values in X, Y, Z axis.

Figure6 5 MPU6050

### Vibration sensor SW-420

 This is the main sensor of the project. Most of the accident cause a shock. This sensor will detect any shocks that will caused by a crash

Figure6 6 SW-420

***GSM GPRS MODULE***

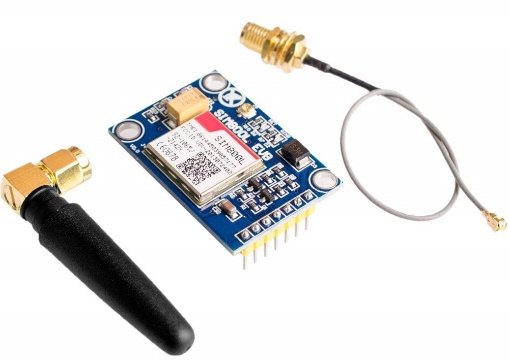
This module is used to send a message to any near by emergency service including the current location.

Figure6 7 SIM800L V2.0 5V

SIM800L V2.0 5V

## Summery

In this chapter we have discus about the technologies and hardware/ software tools we are using to build this project. To integrate with sensors, we are using an Arduino based controller Unit (Arduino Uno/ ESP8266). Controller unit will read the sensors values, calculate them into meaning full values and send them to the WPF Application through serial port and continue with any logics included in the application. From there, it will connect the mobile application that is build in Kotlin to interact with the user. The final build of this project is a working prototype that we will demonstrate all the functionalities and features.