Smart Kart (Project Initiation Document)

George Jacob Anthony Kokinis

November 2021

Student Number 201910280
Word Count: XXXX

Contents

1 Project Background and Purpose **3**
- 1.1 Objectives . . . 3
  - 1.1.1 Primary Objectives . . . 3
  - 1.1.2 Secondary Objectives . . . 3
  - 1.1.3 Tertiary Objectives . . . 4
- 1.2 Scope . . . 4
- 1.3 Deliverables . . . 4
- 1.4 Constraints . . . 5
- 1.5 Assumptions . . . 5

2 Project Rationale and Operation **6**
- 2.1 Project Benefits . . . 6
- 2.2 Project Operation . . . 6
- 2.3 Options . . . 7
  - 2.3.1 Which Platform? . . . 7
  - 2.3.2 Which Language? . . . 7
- 2.4 Risk Analysis . . . 8
- 2.5 Resources Required . . . 8

3 Project Methodology and Outcomes **9**
- 3.1 Initial Project Plan . . . 9
  - 3.1.1 Tasks and Milestones . . . 9
  - 3.1.2 Schedule Gantt Chart . . . 10
- 3.2 Project Control . . . 12
- 3.3 Project Evaluation . . . 12

A Initial Kanban Board **13**

B Risk Analysis Table **14**

Chapter 1

Project Background and Purpose

1.1 Objectives

This project seeks to create a software application for a smartphone, that detects when a vehicle the phone is in enters an "Average Speed Check zone", starts tracking the vehicle's speed, and gives the driver an audible alert if they are at risk of breaking the speed limit.

The usefulness of this project is the safety benefits it may provide. If the application produced can reduce the frequency that a driver needs to check their speedometer, then they will be more aware of road conditions; which can lead to safer driving.

This project is motivated by the increasing deployment of average speed check zones throughout the UK (doubling between 2013 and 2016 (source)), and the potential they may have for leading to distracted driving.

1.1.1 Primary Objectives

The Primary Objectives of this project are:

- To develop an app (for a smartphone) that determines when the user is in an "average speed check zone", and begins tracking average speed.
- Said app then warns, with an audible warning, if the user's average speed is above the limit.
- Voice commands may be used to launch the app at the user's request (for example, if there is temporary speed check area).

1.1.2 Secondary Objectives

If time allows, the project may achieve the following objectives:

- The app allows for the user to manually set what the speed limit is (for example if a road has a temporarily reduced speed limit).
- The app allows for setting the audible warning to a custom sound.

1.1.3 Tertiary Objectives

In many territories globally, the use of devices to detect speed cameras is illegal, and apps are either explicitly or potentially illegal It would be ideal if the app could detect it was in such a territory, and prevent its own usage.

As well, the app may allow the users to contribute to an open dataset of average speed check zones.

1.2 Scope

There is scope within the project to provide an app that is cross platform, using technologies such as Flutter, UNO, and .NET MAUI (née Xamarin.Forms)[1]. This would allow users of the major mobile ecosystems to use the app.

However, doing so requires a developer to be familiar with the relevant tooling and languages (Dart for Flutter, $C_{\#}$ for UNO and MAUI). If a developer is not familiar, learning the tooling may take up a significant amount of time that should be spent on the actual programming. Hence, we are presented with a choice to be made during the analysis stage of the project.

There is also scope to build an independent database of average speed check areas; this would bring "server-side" programming into the project. Alternatively, there may already be public datasets available for use. Whether to use existing datasets (if possible) or build an independent one is another decision for the analysis stage.

Should the project opt to use public dataset(s), allowing for contribution to such datasets from within the app would serve the public good and benefit the app's users; so this may well be within the scope of the project.

1.3 Deliverables

The Project will deliver a smartphone application that meets the Primary Objectives (defined in section 1.1.1).

The project will have met its objectives when the app matches provides features defined by the Primary Objectives. If the project **also** implements Secondary or Tertiary Objectives, it may be considered to have "exceeded" its objectives.

[1] Flutter (citation), UNO (Citation), .NET MAUI (Citation).

1.4 Constraints

Testing the application in the most *direct* way - using the app while driving - may be constrained by:

- Law: It's illegal to use a mobile phone whilst driving, under the (source, (Road Vehicles (Construction and Use) Regulations, 1986(amended 2003):110))
- Insurance: most motor insurers aren't favourable to claimants using their mobile phones. Furthermore, most insurance doesn't cover "testing" on public nor private roads.
- Ethics: The University of Hull's Research Ethics Policy (source) defines "non-malfeasance" as the "avoidance of harm". Testing the application in the above way flies in the face of this, as driving distracted is an easy way to cause harm.

The obstruction this causes to testing can be mitigated through the use of an emulated device and "falsified" location data.

Otherwise, there are no external constraints known at this time.

1.5 Assumptions

There are currently no unknowns for this project; hence, there are no assumptions to be made.

Chapter 2

Project Rationale and Operation

2.1 Project Benefits

Successful delivery of this project will most benefit car drivers (primarily in the UK, but possibly in other territories), with phones, who often drive in areas with average speed check zones.

By reducing the time spent monitoring their speedometer, they can be more conscious of the road around them, and be more alert to potential incidents.

As well, by providing *advance* warning of exceeding the average speed, they may be able to drive in a smoother manner and avoid the "kangarooing" effect, as seen when a driver brakes harshly to avoid a ticket; this can reduce congestion and prevent accidents.

2.2 Project Operation

Agile Software Development is a paradigm, commonly defined by the Manifesto (source); within this paradigm are many varied methodologies. For projects where there is only one team member, methodologies such as Kanban and Scrum (as modified by Scrum for One (source)) are optimal. (source??).

The Kanban board is a useful tool to understand the current state of the project; what is in progress, what needs to be started, etc. Combined with the user-input and cyclical nature of the Agile paradigm, this is ideal for a project like this one, with one developer and fast development times.

The overall planning and schedule for this project is laid out in the Gantt Chart in section 3.1.2. The intention is to stick with this schedule; however, the agile paradigm requires flexibility, so this may change with time.

2.3 Options

As discussed in the section on Scope (1.2), for developing this project there is certainly one choice to be made: is the app developed using one of various cross-platform frameworks, or using platform-native frameworks.

In that discussion, it was already decided that using a cross-platform framework is not in the scope; however, even within the realm of platform-native, there are more choices to be made.

2.3.1 Which Platform?

The Project's nature requires a mobile platform. However, there is a choice within this; despite Android and iOS' apparent dominance of the market (with (source) claiming combined a 99.19% share, as of September 2021), there are various other mobile platforms, such as Tizen[1], KaiOS, and Sailfish OS; as well as mobile implementations of desktop platforms, such as pureOS.

Despite this variety, Android stands tall as the easiest to develop for and most accessible platform. The Android Open Source Project means that obtaining an image to test on, or examining the OS' source to aid in debugging, is relatively easy. Hence, Android is an ideal platform to develop this project on. Furthermore, using Android Jetpack[2] whenever possible will provide the project with a consistent base to build upon.

In comparison, developing platform-native applications for iOS devices requires a MacOSX device to use Xcode, and a payment of $99 if you wish to distribute your application. The other mobile platforms have significantly small market share.

2.3.2 Which Language?

Android applications have historically been written in Java, and the foundation of the OS is a JVM (originally Dalvik but later ART). The ecosystem around Java on Android is mature and well-documented.

However, in 2011 JetBrains™ announced Kotlin, a JVM language "having the features so desperately wanted by the developers" (source). Being a JVM language, it was inherently "usable" on Android; but the Android Team announced "first-class support" in 2017 (source) and "Android development will become increasingly Kotlin-first," in 2019 (source).

Hence, this project could develop the app in Kotlin or in Java. Kotlin is now the contemporary language, and Google suggests that "If you're starting a new project, you should write it in Kotlin" (source); however, Java already has a massive ecosystem surrounding it.

[1] Backed by the Linux Foundation, but primarily developed by Samsung.

[2] Android Jetpack (citation)

2.4 Risk Analysis

The most major risks to the Project at this time are those associated with Covid-19: Potential future lockdowns, mutations of the virus, supply issues, and potential infection of the Author. These risks can be mitigated by the actions of the Author, but only to an extent; it depends upon the actions of others as well.

There are various other risks, such as personal illness, strikes, and equipment breakdowns; the analysis table is in Appendix B.

2.5 Resources Required

There are no abnormal resources required for this project. All that is needed is a computer to develop on, and an android phone (or the AVD emulator) to test with.

If any of these were unavailable (for example, if the computer being used became non-functional), it would delay the project until an alternative could be sourced, or until the University's Labs are available.

Chapter 3

Project Methodology and Outcomes

3.1 Initial Project Plan

3.1.1 Tasks and Milestones

The tasks for developing the application can be broken down into Milestones. Within milestones, tasks may be performed asynchronously and simultaneously; but generally, milestones are dependant on the previous milestone.

Milestone 1: Design and Architecture.

- Evaluate potential application designs and architectures.
- Plan the overall application architecture.

Milestone 2: UI and UX.

- Plan the User Interface (UI) design.
- Plan the User Experience (UX).
- Implement the UI.
- Implement the UX.
- Testing of the UI and UX.

Milestone 3: Average Speed Algorithm.

- Evaluation of Algorithms for determining average speed.
- Implementation of Algorithm.
- Algorithm Testing.

Milestone 4: Determining if the Device is in an Average Speed Check Zone.

- Evaluation of potential Datasets.

- Design of Data Model.
- Implementation of Data Model.

Milestone 5: Glue Code, Bug Fixes, Further Objectives, and Delivery.

- Creation of any "glue code" necessary; for example, changes to build tools, code necessary to publish on the Play Store/F-Droid, etc.
- Bug fixes: Any final bug fixes for issues not found during testing in prior milestones.
- If time permits, Secondary and Tertiary Objectives will be delivered under this milestone.
- Delivery of the final Artefact.

Alongside these tasks, there is the singular long-running task of writing the final report.

3.1.2 Schedule Gantt Chart

The Gantt chart here is not as detailed as the task list above. This is due to the project using Agile and Kanban, meaning that tasks can be running out-of-order, simultaneously, and asynchronously; which isn't easily represented on a Gantt chart. Hence, the chart is more of a "timeline", and this PID also includes a snapshot of the initial state of the Kanban board, found in appendix A. This initial state may change upon commencement of the report and the creation of User Stories, Epics, etc.

GanttChart here!

3.2 Project Control

Management of the project will be through the use of the Kanban board. The start of any development session begins by inspecting the board to understand the state of play. Then, the developer shall choose a task that is already in development (or if there are none, a task that hasn't been started), and work on it for the session.

Performance will be monitored based on the rate of progress through tasks. Should tasks take too long to complete, this is an indicator of a low performance. However, the inverse, tasks being completed quickly, is not inherently an indicator of high performance; it may well mean that tasks are not being completed to sufficient depth.

Whether the project is successful will be judged based on the resulting artefact, and whether this artefact has met the objectives in 1.1.1.

3.3 Project Evaluation

Evaluation of the Project's artefact(s) will be based on how close it adheres to the objectives in 2.2.

In terms of User Evaluation, testing the entire application is limited by the constraints layed out in 1.4. However, the UI/UX could possibly be tested by users, independent of the underlying application.

Appendix A

Initial Kanban Board

Appendix B

Risk Analysis Table

Risk	Example	Likelihood	Severity	Impact	Mitigation Taken
Total Equipment Failure.	The Author's equipment (computer, phone) experiences total failure, preventing progress on the project until repaired.	1	4	4	Regular Maintenance is performed on the equipment, and backups are made. As well, the date for this Project is stored as a repository on Github, so data loss should be minimal.
Partial Equipment Failure.	A part of the Author's equipment fails, degrading the ability to progress with the project.	1	3	3	Regular maintenance, backups, Github; also, alternate devices are available.
Covid Lockdown	The UK goes into another lockdown due to rising covid cases, preventing access to campus and	2	3	6	This is a public health issue; mitigation requires collaboration across society.