IntelliV – Fleet Management

Writer: Callum Gray

Institution: The University of the West of England

Course: Software Engineering for business

Supervisor: Dr Raj Ramachandran.

**Cover Page**

**Software Project**

-Aims & Objectives of the project

-The end goal of the project

-Current systems in place for the proposed customer (F service) TRANMAN

-google trends queries?? (Shows need for application)?

# **Chapter 1 – Introduction**

# Overview

The following report will show extensive reflection and summary of my final year software development project. ‘IntelliV – A fleet management service application to help track and trace the location of various equipment within the fire service.’ Throughout my application development, I will be using the SDLC (Software Development Life Cycle) to review my progress with reflection on both successes and improvements that could be made at all stages of development.

# 1.2. Problem statement

Currently, the fire service is looking into how they can employ a ‘cohesive mind’ approach across all counties within the UK in respect of their fleet management approach. Some of the counties use pen and paper approaches for daily vehicle checks and logging, some use one software and others another. The purpose behind my project is to understand what the fire service needs and how I can best implement a software system that could be used widely across the service to improve their daily tasks, whilst upholding a secure environment that is adopted by all. Some suggested problems with the fire service’s current way of works include but are not limited to.

* Pen and paper methods (Open to potential loss of forms or tampering.)
* Human Error
* Lying those checks have been completed

# **Chapter 2 – Software Development Methodology & Techniques**

# 2.1. Overview

Below I will address the SDLC stages and include a brief description of how these stages will also be implemented within my project. Throughout my development, I will be using a waterfall development approach which will also be addressed in this chapter alongside the use and evidence of my version control by Gitlab. Further in the report, I will address each stage of the SDLC in further detail in relation to my project work and link to other aspects of software development such as requirements analysis.

# 2.2. Software Development Life Cycle

The SDLC is a process that is used within the software development industry to design, develop and test the software before deployment. FIND REFERENCE AND OVERALL MEANING OF SDLC FOR HERE

# 2.2.1. Stage 1 – Planning

This stage of the SDLC includes thinking of potential solutions to the problem ahead and building scope for the project. Here the team or individual will consider what resources and costs the development of this project may incur. For me personally, this stage is where I developed initial plans and ideas and began asking questions for how I could address the problem ahead. This area of development was key to my project as it led to my understanding of the time available and resources, I could pull upon to get this project done by its deadline, therefore, assisting me in finetuning the project scope to not be over-ambitious with my application development.

# 2.2.2. Stage 2 – Systems Analysis & Requirements

Here begins the real work, stage 2 includes gathering any information available from potential stakeholders and enables you to obtain a detailed definition for the systems requirements (Existek, 2017) Here development teams will consider any functional and non-functional requirements for the solution and analyse any current systems that are in place to make sure that the new system will reach the requirements of the project where other systems have previously failed. In relation to my individual project, this stage of the SDLC involved me interviewing Peter Warner (CHECK TITLES) and completing literature reviews on currently available systems out there. I also managed to find a website that details the ideal requirements for an application that would be used within the fire service. (WEBSITE LINK) During this development stage, I also began to research potential languages/software that could be used to begin building my application that would best suit the needs of the end-user.

# 2.2.3. Stage 3 – Systems Design

At this stage of the SDLC, it is down to the developers to begin high-level designs of the proposed system to enable them to see whether they will be able to achieve the requirements built in the last development stage. For my software project, I intend to include wireframes, Business process modelling and use case models to display the system and potential uses/requirements that need to be addressed for successful completion of the project.

# 2.2.4. Stage 4 – Development & Programming

This stage of development progresses once all requirements have been decided and a rough plan of the proposed system is developed. Within my project, this is the main bulk of my artefact creation – the fleet management application. This is where I will begin the development referring to requirements set by myself within the requirements analysis throughout the project.

# 2.2.5. Stage 5 Testing

Before deployment of my application to the client I need to conduct the testing phase of the SDLC, this helps with quality assurance testing my proposed system against requirements and seeing if any errors can be found and debugged before deployment of the project. Here I will be using a testing table and conducting tests on a user basis to see if there are any issues within my program that I can debug. I hope to have testing conducted by a firefighter however due to the ongoing coronavirus crisis I am not sure how feasible this will be therefore I may be employing an unbiased alternative to conduct testing for me.

# 2.2.6. Stage 6 – Deployment & Maintenance

Once the testing stage of the SDLC has been completed we can move on to deployment and maintenance of the application, normally with software development a maintenance team will be assigned to the product to help clients with any issues they may find post-release. For me this is going to be the degree show where I will be showcasing my work to lecturers and fellow students for feedback, maintenance and official deployment could be a potential option after course completion should I wish to follow that route.

# 2.3. Waterfall Model

During this project, I will be employing the waterfall model to conduct work on this project. This model follows a sequential flow of development where each phase needs to be fully completed before moving on to the next phase can begin. This is my chosen management model for the project due to its ease of understanding and ability to deploy against my project, alongside waterfall I will also be making use of Kanban – a Trello board will be used to enable me to always stay on task and not lose myself within the workload.

Discuss each stage of my waterfall pattern

Include link/images to Trello board to demonstrate the use of Trello

THEORY BEHIND KANBAN/WATERFALL – for small development.

# 2.4. GitLab VERSION CONTROL

Any work completed on the application side of the development will be made using visual studio code and Gitlab where versions will be committed upon working changes to my defined space of ‘IntelliV – Fleet Management Application’. Gitlab provides me with tools that can enable me to monitor progress made and iron out any areas of development that issues may occur; this development technique also helps reduce risks of lost progress as the work will be loaded onto a cloud platform therefore also opening me up to the possibility of working on various devices should one fail.

IMAGE OF GITLAB VERSIONS WHEN DONE

# **Chapter 3 – Requirements Analysis**

-- Scope of system

-- Quality assurance

-- Functionality – FR and NFR?

-- reliability

--usability

--efficiency

--maintainability

--portability

-- MOSCOW model? (Must haves, should haves, could haves, wont haves)

|  |  |
| --- | --- |
| Must Haves: | Should Haves: |
| Could Haves: | Wont Haves: |

**-Method**

-Research into implementation technologies

-Rendering (using flutter so all devices should be fine but maybe talk about this? And its pros/cons?)

--REACT VS Flutter/dart?

-application scalability

-Database usage? Camunda – removes the need for the database (see Dilshan)

-software development models & techniques

# **Chapter 4 – Data Collection**

# 4.1Overview

# 4.2. Interview transcription

-findings from interviews developed

-refer to website link found that addresses potential application requirements

# 4.3 potential solutions? i.e., QR VS NFC

# 4.4 Software Research

Systems review matrix – how did I decide to use flutter vs others?

**Systems Review Matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Complexity: | Device Compatibility: | Firebase Compatibility: | Extensions: | Cost: | Community Support: | Total: |
| Software: |  |  |  |  |  |  |  |
| React |  |  |  |  |  |  |  |
| Xamarin |  |  |  |  |  |  |  |
| Flutter |  |  |  |  |  |  |  |

Hackernoon – 11 popular cross-platform tools for app development

Above is the systems review matrix completed by myself using secondary research to help me decide the best language/software to use within my project development. Each rating is scored out of 5 and then totalled to give me my final solution of software choice.

# **Chapter 5 - Design**

# 5.1 Overview

# 5.2 Use case diagrams

# 5.3 BPMN diagrams

# 5.4 Sequence Diagram

# 5.5 Swot analysis

|  |  |
| --- | --- |
| Strengths: | Weaknesses: |
| Opportunities: | Threats: |

# 5.6.1 UI mockup design

# 5.6.2 wireframes

# **Chapter 6 - Implementation**

# -Overview

# -Application

# -hardware used

# -Software used

# -User guide?

# **Chapter 7 - Testing**

# 7.1 Test Table

# **Chapter 8 - Conclusion**

# 8.1 critical reflection

# References

# Interview transcript

# Ethics proposal checklist