C O V E N T R Y
U N I V E R S I T Y

Faculty of Engineering, Environment and Computing

School of School of Computing, Electronics and Mathematics

MSC SOFTWARE DEVELOPMENT

M08CDE INDIVIDUAL PROJECT

A Solar powered surveillance solution to monitor traffic congestion on a mobile application

Author: [redacted]

SID: [redacted]

Supervisor: Leonid Smalov

Submitted in partial fulfilment of the requirements for the Degree of Master of Software Development

Academic Year: 2016/17

Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc) has been acknowledged within the main report to an entry in the References list.

I agree that an electronic copy of this report may be stored and used for the purposes of plagiarism prevention and detection.

I understand that cheating and plagiarism constitute a breach of University Regulations and will be dealt with accordingly.

Copyright

Signed: [redacted] Date: 07/11/2017

Office Stamp

Abstract

This project is a Solar powered surveillance solution to monitor traffic congestion on a mobile application. The surveillance system in context has been put together with an aim to reduce the negative impact of traffic jam on road users using various busy roads around the United Kingdom. To accomplish this, a combination of hardware and software components has been put together to achieve this.

The hardware system end is a uniquely designed apparatus consisting of cameras installed on busy junctions to monitor the traffic behaviour. These can rotates up to 360 degrees horizontally, and up to 180 degrees vertically thus covering all angles on any road junction, or highway. The hardware side of this system has been programmed to capture images, in all strategic angles of the road, in extremely short time intervals, through out the day, enough to interpret the traffic situation in that area.
For each interval set captured, the time stamped images are processed into a suitable format and uploaded to the server for further processing.
The sustainability and reliability of this hardware component relies highly on a resilient power management sub-system that has been designed to ensure operation on a 24/7 basis.

The Software end of this integrated system is comprised of a set of modules that have been stitched to process images and make them available on a mobile platform accessible to basic day to day road users while ensuring strict privacy and harmony in the public transportation system.
The software end is designed on the client-server model. A server is used to receive, process and store images from the locally positioned camera units around various areas whereas the client is a mobile platform based on Android operating system that is used to provide a reliable and efficient usable experience while monitoring traffic and planning a road user's trip.
To ensure and maintain privacy, all images availed on this platform are filtered using image processing algorithms to eliminate pedestrian faces and car registration plates before made available for application end users.

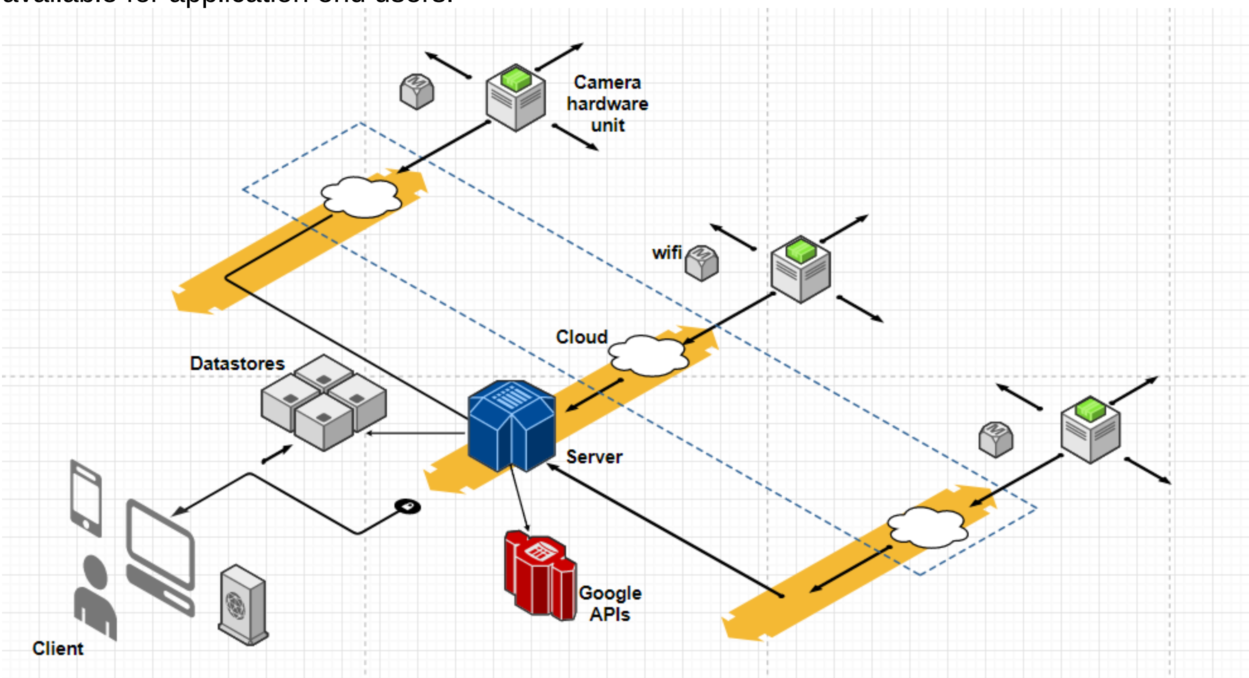


Table of Contents

[This table is constructed automatically by Word, by collecting items set in Heading 1, 2 or 3 style. To update the table, right-click on it and select “Update Field” > “Update entire table”.]

Additional Materials on the Accompanying CD

Acknowledgements

1 Introduction

Described in this document is a set of information put together to show how image processing can be used to assist in the negative impacts of traffic jam on car owners. A device has been put together to work with a phone based mobile application developed on Android platform to accomplish the goal in context. The purpose of this is to put power in the day to day road user's hands that gives them access to 3-minute traffic updates using *processed* images on a user navigation led map layout.

1.1 Background to the Project

In a busy economy, a driver spends at least an average of 10 hours in a week. Within that time, the party affected experience delays, accidents from congestion, unexpected gas problems among others.

The idea was induced from frustration experienced during a journey, without knowledge, drivers get stuck in traffic jam they could have avoided. The idea of this project was to assist drivers to make knowledgeable trips with information that meets the eye (images) on their smartphone device *safely*.

"Public transport has not been able to satisfy the transportation needs of the population leading to rise in vehicle ownership". Aparajita, et al (2014).
The road traffic surveillance system is a computerized solution that will aid in reduction of traffic jam on major roads around Kampala city. The product will mainly use camera technology. Images of different roads around Kampala city will be taken by cameras elevated on high ground and these images will be uploaded onto a server which can be accessed on different mobile devices using mobile phone applications on either android or windows platforms by end-users..

The system will guide drivers, traffic police and wardens on less congested routes around the city.

1.2 Project Objectives

- To collect potential user requirements. These include basic day to day road users.
- To discover the best possible solutions to empower the user over unpredicted traffic jam.
- To assemble the required hardware and software components to support the solution and it's purpose.
- To develop the necessary artifacts (software) needed to produce public safe content to a multitude of users at anytime.
- To establish swift and reliable communication between the remote hardware and the required processes to provide user friendly content.
- To test and ascertain the solution for mainly usability as well as durability, reliability, stability and maintainability.

1.3 Overview of This Report

This report contains information that has been detailed to explain the traffic surveillance system developed. The document describes the architecture, system design and development of a road traffic surveillance system.

The report helps translate software requirements into a representation of software components, interfaces and data necessary for the actual implementation phase.

The report shows how the road traffic surveillance system will be structured to satisfy the problem in context and it is also the primary reference for code development.

The report will be used by system programmers, system architects/developers, System analysts, User Interface designers, database designers and Graphic designers.

1.4 Definitions and Acronyms

Table 1: A table showing a list of abbreviations used in the document.

Acronym	Definition
SSH	Secure Shell
CCTV	Closed Circuit Television
PTZ	Pan tilt zoom
GCMID	Google Cloud Messaging

2 Literature Review

"Due to the poorly planned road networks, a common outcome in many developing regions is the presence of small critical areas which are common hot-spots for congestion." Vipin Jain, Ashlesh Sharma, et al (2012). This congestion indirectly sums up many negative events, fuel wastage, delays to important appointments. The solutions currently devised by governments and infrastructure development authorities are somewhat long-term. Expansion of roads would be the ideal solution to tackle this problem, others include construction of bridges and fly overs. In Capital cities, structures have been re-designed and road networks revised.

Research undertaken by respected organisation INRIX shows that the cost derived from traffic congestion is predicted to sky rocket to an enormous 63% by 2030, estimated at £21 billion per year. "Due to traffic jam a substantial portion of working hours have to be left on streets which indirectly put adverse impact on economy." Khaled Mahmud, et al (2012) Estimation between 2013 and 2030 sums up to £307 billion in the United Kingdom. Such a case in a developed economy reflects worse situations in developing countries. Congestion of this multitude is attributed to the speedy growth of the population in urban areas.

Uganda, nicknamed the pearl of Africa is an ideal example for a developing economy suffering from the congestion disease. A Statistics report compiled in 2012 estimated over 90,000 new motor vehicles registered in that year. The low per capita income realised for over a decade equates an unchanging road network. Truly any economy in this situation is silently failing.

In South Africa, one of the most developed economies in Africa, a survey (TomTom Traffic survey) concluded that 78% of the 3.8 million vehicle operators in capital city Johannesburg alone suffer daily from impact of prolonged traffic congestion. 342,000 are forced to cancel or postpone important meetings and more than 40% reach late for work. Without this congestion, there would be more positive lifestyles for the citizens of this economy.

In New South Wales-Sydney region, using live updates of traffic on busy roads, providing traveller information and initiating customised alerts all designed on a geographical map layout have been improvised by the transport sector to mitigate the effects of the congestion. This website has been equipped with minute by minute updates of image content on various busy junctions, floods, road works, alpine conditions, live updates of road hazards, fires, major events, and other information all laid out on a map view freely accessible to the public.

In developed countries, large technology giants such as Google have devised means of tracking the congestion by utilising crowd source methods to provide graphically interpretable information to auto mobile operators regarding traffic. These road visuals are derived from GPS locations of all mobile devices connected to Google servers. Red for extreme congestion, orange for mild, and green for clear traffic are the key colours used to signify the current nature of traffic.

Other means of transport have however been devised. Business Insider lists some being train rails, buses, smart cars among others.

The traffic congestion monitoring system in context doesn't eliminate all the setbacks from traffic congestion. However, it introduces a dynamic and flexible platform that avails the vehicle owners control over road trips. Up to date analysis of traffic jam at various peak times, ability to report disasters and accidents, are among the functionalities of the solution. This indirectly gives the end user ability to cut costs on fuel used, minimise repercussions of accidents, better planning of the day's tasks among others. Road expansion is obviously not expected in the immediate future; therefore, this can be a temporary alternative to relieve from stress and frustration incurred from traffic congestions. Meetings can be better planned with consideration to the journey plan. The solution attempts to present mobile phones as a reliable platform to achieve all this.

This solution contrasts using closed circuit television (CCTV) surveillance mechanisms which entail much complexity and are costly to implement for this task. Fixed or PTZ (pan tilt zoom) surveillance options are available, however, video streaming is expensive to undertake unless it's for security purposes. "Real-time video transmission is a challenging task, especially when typical error handling mechanisms, such as retransmission, cannot be used" Francois Caron (2015).

A hardware component customized and programmed on a microcontroller to capture images in short time intervals using a camera module, specifically designed for monitoring traffic congestion would be the ideal alternative to meet a low-cost budget for a developing economy, with consideration that many of these cameras are to be deployed on busy junctions around the city suburbs. The captured data can be stitched for fast clips to analyse a traffic congestion in a time frame. 360-degree Panorama images can be created to simplify navigation of the area.

In United Kingdom, the Highways England Traffic Management Technology Framework (TMT2) supports the use of wireless CCTV services. However, these are only available to organisations in charge of managing the public road network but rather not for private use by the public for traffic congestion monitoring or any other context. The information collected is used for various applications such as road surveillance (for security purposes), congestion monitoring and queue detection by traffic authorities, asset protection, road surveys, health and safety, temporary roadwork schemes, among many others. Unlike this, the goal of the project described strives to tailor a platform available straight to the public while embedding all the above-mentioned applications. Elimination of sensitive information such as vehicle plates, faces makes the image content eligible for public access. The project aims at respecting privacy laws and maintaining harmony in the public transport sector whilst improving on the life on the road.

Rationale over existing solutions.

- Panorama view (360 degrees horizontally and 180 degrees vertically) of a road junction using an android mobile application.
- Provision of vast alternatives during route selection, travel planning, scheduling and so much more.
- Utilization of cheap hardware component and use of green energy to provide flexible and dynamic surveillance.
- Image processing algorithms ensure privacy remains a top priority.

- In comparison to video streaming, image compression schemes utilized ensure less data, memory and processing resources used on the end user platform.
- The solution is multipurpose. Besides helping motor operators, traffic officers can use the system to monitor and control traffic based on information analyzed from the platform.

A key component on the hardware end of this architecture revolves around a microcontroller. This mini-computer is a chip used in various applications such as fuel meters at gas stations, washing machines, coffee makers, microwave ovens, in cars for air bags. The domain scope far exceeds imagination. Formulated and designed by Gary Boone in early 1971, initially called TMS180NC, then boasting of 3000 bits of program memory, and RAM of 128 bits. This chip comes at extremely low cost for unlimited functionality especially if paired with other peripherals. Today, over 4 billion microcontrollers are sold off in a year. This accounts for the over 20 embedded chips in a basic household's electronics. "Embedded devices are more prolific than ever before, with the IoT and its applications being a key driver, including Smart Cities, Smart Homes, Agricultural Technology, Industry 4.0 and associated communities." Steven J Johnston, et al (2017).

The Raspberry Pi Microcontroller used for this project is an open source hardware innovation with early concepts launched in 2006 to promote computer science education in developing countries and schools. Support of over 10 operating systems such as Android, Debian, Linux among others all from a memory stick makes the Raspberry Pi one of a kind. A Camera Module, SG90 Tower Pro Servo Motor, Stepper Motor, Sim 900 GPRS/GSM Board and a Micro SD card module comprise the peripherals paired to the Raspberry Pi board for the hardware unit used for this project. The board is powered by a 12 Volt rechargeable battery and 4000 maH power bank connected to a solar panel with a regulator to moderate flow of current.

In 2011, Cisco systems, an American technology conglomerate, predicated that the number of IoT devices would by the year 2020 exceed 50 billion. The number of already connected devices was estimated to be more than 8 billion in 2017.

Mobile applications have increasingly become a popular platform to address vast information technology opportunities and challenges. As a growing trend, statistics estimated the collective number of smartphone owners worldwide to have exceeded a 2.1 billion mark by 2016. With these numbers, mobile applications usage has been accustomed in new domains, leisure, entertainment, marketing, sales, medicine among others.
"The combination of mobility, context-awareness, and proactiveness enables a new class of user-centric and mobile lifestyle applications that become increasingly important" M.J. van Sinderen et al.

Usage of a custom mobile devices is justified for this project for reasons of reliability, mobility, usability, accessibility among others. Web access mediums can as well be introduced later for access of content over the any browser.

3 Methodology

"The field of embedded product development has wide diversity of products with different needs and domain-specific problems so that no single method is applicable, but rather many methods and practices are needed for different situations." Matti Kaisti, et al (2013)

Development of a system that works with embedded hardware components required careful design and creation. For the traffic surveillance system, I decided that both the software and hardware undergo development hand in hand. For example, to test a pulse on a Tower Pro Servo motor requires me make a python script. Both hardware and software may involve a lot of trial and error. In addition to that, the architectures seemed like they would require adjustments from time to time, to best suit the system functional and non- functional requirements.

With knowledge that software development projects from time to time suffer from long development cycles, poor interpretation of customer expectations, unpredictable schedules, late delivery, quality control issues and at times developer burnout. Unpredictable product delivery at times leads to schedule pressure, and shoddy planning. Now, schedule pressure is followed by long hours and attempt for shortcuts to the main goal. Long hours lead to mental burnout and shortcuts lead to software defects. These defects eventually result into more long hours debugging. Debugging itself is a long process and may result in occurrence of all the above. "Increased demand for better technology and perpetual global expansion continue to provide developers with many project opportunities for success, as well as failure". Kaitlynn M.Whitney, et al (2013)

The agile software development principle is most favourable for this kind of project. To produce a working set of components for the traffic surveillance system involves continuous evolved development while analysing end user requirements. "This methodology provides a pragmatic way of adapting to changing requirements at any point during the lifecycle of a software project". Prof. Hironori Washizaki, et al

As a plan driven methodology, it has enabled easy tracking of the project milestones especially during design and implementation. With it, am guaranteed room for adjusting or changing the architecture to perform the main goal to the best ability through revising old stages.

3.1 Summary of agile development stage activities.

3.1.1 Conception

To have a surveillance system that has an all-round view of the road junction on a 24/7 basis to assist drivers to skip congestion areas is the goal at hand. To achieve this, I had to use my imagination to plan how to make this happen.
The idea is to construct a system that works to the best of its requirements while providing room for refinement and optimization in the earliest stages of design.
At conception, below were the targets;

- I need a camera to capture a full view of the road junction.
- I need the view to be updated on a mobile device for every 3 minutes.

- I need the view to not have private information.
- I need the camera to stay on.

3.1.2 Initiation

To capture a full view of the road junction.
In most public places, to get a versatile view requires installation of multiple cameras each pointing in its direction. The unit cost for each apparatus installed at a junction far exceeds the budget for the traffic surveillance network system. I imagined assembling a set of components to work together to capture and upload the view.

To update the view every 3 minutes
This requires me to write a set of instructions to be followed by the camera. The camera must receive these instructions from a processing chip. The processing chip must have a reliable timer which triggers the camera to capture an image.

To not have private information in the view.
"Historically, privacy was almost implicit, because it was hard to find and gather information. But in the digital world, whether it's digital cameras or satellites or just what you click on, we need to have more explicit rules - not just for governments but for private companies." – Bill Gates. Harmony is a necessity when sharing roads among drivers. To maintain this, all data captured must in some way be stripped of sensitive information. Vehicle plates and pedestrian faces appeared to be the most important. Algorithms to detect faces and characters exist. However, how efficient they are will depend on the quality of the input.

To have the camera stay on.
24/7 traffic surveillance would be a myth. The kind of power supply to maintain surveillance operations had to be not only efficient but durable and reliable. Solar energy ideally fits the bill as Kampala is always shining. A tropical environment can support the surveillance hardware but only if the power supply is properly regulated.

3.1.3 Analysis

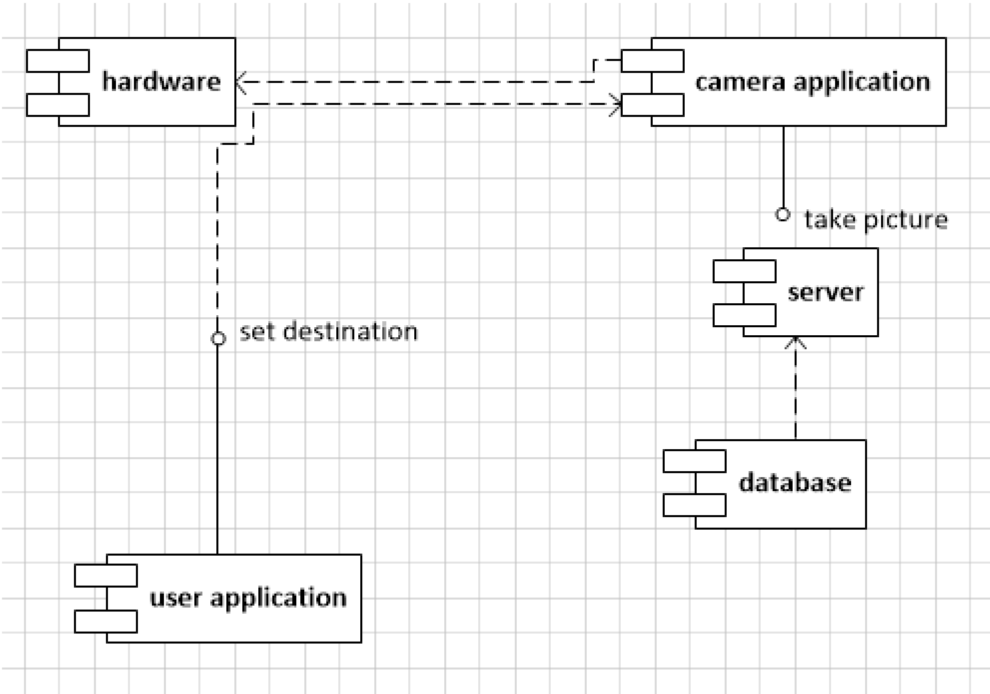
The reason for this system stems from driver frustration due to traffic jam. The traffic surveillance system requires prioritisation in module development. Clear understanding of how the architecture should pan out is demanded. How the components will be assembled was the purpose of this analysis. The budget had to be critically analysed and revised.

3.1.5 Design

I had to carry out accurate calculations on the hardware end all components perform smoothly as required. The design of the entire system was based on a client-server model. The end user sends data requests through the mobile device and the server responds after authentication. The data requested is stored in a secure data store with an expiry period of 24 hours. The data captured is processed for faces and car registration plates before being availed for access. The hardware end relies on regulated solar energy to power the components. Power regulation techniques include utilisation of an external board to switch between power states accordingly, tilting of the solar panel as the sun moves among others.

3.1.6 Construction

From the design, components are interconnected and scripts are written to control each component. Instructions and schedules are loaded through the Pi microcontroller. These are written in Python programming language and stored in a central location.
The Server APIs are written in C# programming language. These interfaces are exposed on the network using Internet Information Services and accessed using routes. The Get and Post methods sent to the server are all authorized using a unique authentication code. Each request sent to the server requires this code to be attached on the header. The API responsible for handling post requests from the hardware end requires parameters to be attached to the request. These parameters are encapsulated in a data transfer object model comprising of the camera id, the location, the image name, and the image string (expected in base 64 format). The image data is processed from base 64 to normal PNG format, passed to the Google face and text detection cloud API algorithms before being stored on the hard disk.



3.1.7 Testing.

Each subcomponent is initially tested using white box techniques. Working sample code has been written to test each component for reliability, performance, and efficiency. The microcontroller is tested for scalability. Since it controls all the peripherals, it is ideal that it doesn't experience any faults now or soon. The server is tested for fluency, questions like can the server handle 10 requests simultaneously, and at what rate are crucial to be addressed.

3.1.8 Deployment.

The hardware components are mounted on a long pole, long enough to provide a bird's view of a junction. The installation point is studied critically to ensure the camera captures a view of all roads leading from the junction. The server is transferred from local network to a

centralised cloud accessible network. It should be easily accessed from any location around the world.
The mobile application is uploaded onto the Google play application store for download by the end users.

3.2 Rationale for selection of the agile methodology.

This kind of methodology has enabled me source for feedback and adjust the traffic system. End user collaboration and interaction during design and development is a necessity to the success of the surveillance system. An example would be, if the end users requested for more intervals on a certain road? The system can be adjusted, refined and rebuilt to suit that requirement. Using it supports user involvement during the development process ensuring for no drastic diversions from product requirements.

This technique gives me regular feedback on progress by breaking the project construction into iterations that are usually one or two weeks in duration. The output of each iteration is a working software piece of the architecture. Each iteration is like a stand-alone mini project scheduled to expire after a specific amount of time, and delivering some executable version of the product. In early iterations, the end-product only run in a test environment as a prototype.

The hardware development process can be divided into small demonstrable units. For each unit completed through an iteration, the end user can get feedback on the specifications during demonstrations. Before the final version of the hardware end is made available, each iteration involves simulation and tests using real working code.

This kind of development technique enables to add new features or make enhancements to the surveillance system. The system is expected to be used year and after year and evolve along with the market and changing hardware and technology. The mobile application can be refuelled with new features for the end-user such as in-app accident reports to the authorities, Graphical Traffic congestion analysis among others.

This technique enables the development cost to be manageable since the final product is produced eventually after the prototype has been approved. Any changes will always be made on the prototype and tested for feasibility.

4 Requirements

Requirements collection

Using the elicitation process, the requirements of the system were gathered from the environment and end users.
The real problem identified was the frustration incurred on road users by traffic congestion. In addition to that, the road users waste a lot of precious time on the road, instead of doing other valuable and productive activities. Fuel costs affect lifestyle, and the congestion indirectly affects economic growth.

This presents a challenge to developing economies to orchestrate ways to curb the congestion and it's effects on road users and pedestrians. Costly solutions include expansion of roads and deployment of vast numbers of traffic officers to help with the issue at hand.

This challenge has forced the hand of technology related companies to devise means to solve traffic jam using advanced technology equipment and algorithms.
The traffic surveillance system appears to be another alternative while maintaining harmony on the transportation network, it aims to assist drivers to plan their routes using image content. It provides road users with frequently updated images while enabling routing from one location to another. To achieve this the system relies on mobile hardware units deployed all over Uganda in strategic busy road junctions.

The solution strives to be of low cost compared to current measures aimed at helping with the similar problem.
If this solution is not taken aboard, so many Ugandans will continue to drive unpreparedly and without knowledge of what to expect from their daily driving experiences. Road accidents will not be handled much efficiently without the help of the public. This solution enables road authorities to work in conjunction with responsible road users to identify and report un usual behaviour on the roads.

4.1 The technical requirements of the system

- Ability to capture images on a 24/7 basis in 3 minute short time intervals. The hardware components need a reliable power management system to ensure they run at all times. Operation is mostly during the day since traffic congestion is rare in the wee hours of the night.
- At the camera end, the unit should be able to automatically shutdown after each interval, and as well automatically power up after 3 minutes to capture and upload another set of data.
- The hardware should have an active internet connection to ensure all data that is captured is uploaded on to the main server for processing. There should be a back up connection incase the main connection fails for unforeseen reasons.
- In an aim to minimize cost, each hardware unit should have one camera capable of capturing scenes in all angles of the busy road.

- The camera should be mounted on a rotation object capable of achieving up to 360 degrees rotation horizontally and 180 degrees vertically. This in the end supports for a panorama of the area.
- The system have the ability to wash images of faces and any characters mostly being vehicle registration plates. Each image captured shouldn't be released to end users unless processed for this.
- Original copies are to be disposed off in an ethical manner and not stored or shared with any third parties. The processed copies should be disposed as well since they aren't used for any security purposes and thus their content isn't important after 24 hours.
- The hardware unit must have water resistance capabilities. It should be cased in protective gear that ensures safety from effects of harsh weather and storms.
- The server should be able to authenticate and handle all incoming requests from mobile end users as well as handle all posts from the camera hardware.

4.2 Non functional requirements

- Recoverability.
  The hardware components should have recoverability traits. Incase of a fault or a failure in the system, the hardware sub system should be mechanisms to resume daily operation.

- Maintainability.
  The solution should be maintainable and upgradable over time. With change in design and requirements, the components should be easily replaceable or adjustable to suit the new requirements.

- Safety
  The traffic surveillance system should ensure safety of the environment with it's existence. Where poles have been installed, warning signs should be attached to inform pedestrians of it's existence and purpose. Data captured should as well be disposed off in a safe manner.

- Portability.
  Portability of the hardware components is a very important requirement. The set of components put together should be in a manageable size while still ensuring all successful operation.

- Usability
  The mobile application should be appealing to the end user. The activity story should be easy to interpret and navigate. Usability is a very important requirement.

- Performance
  The mobile platform shall have swift response to user requests. The hardware end of the system has to be capable of handling the operation of capturing, processing and transmitting data to the server at a considerable speed. The Server should be able to handle over 200 simultaneous requests without faults. The hardware rotation system shall ensure that over

1000 cycles can be made without failure. The power management system shall supply and monitor power sequences for periods of time without shutting down.

- Scalability
  The traffic system shall be able to allow for component change without distorting the intended purpose but rather improving with additional resources. The software application shall be improved both on user interface and core features. The Server used for handling posts, image processing and end user requests should be scalable. With time, the number of cameras deployed may increase. If this situation arises, the server response time and throughput shouldn't be affected in any way.

- Capacity.
  The communication between the hardware shall run on a 2.4Ghz band to ensure for reliable throughput of up to 10 images sent every 5 minutes to the server instantly. The system database shall be able to store each image for a duration of 24 hours before they expire.

- Availability.
  The hardware components shall be designed to ensure they perform all required processes that is rotation, capturing, image processing and upload without failure on a 24/7 basis. This shall ensure that end users have up to date information at any point or time of request during the day available through their mobile devices. The server shall be tailored for failure handling and ease of troubleshooting.

- Reliability.
  The power management of the system shall provide current required for running the circuitry boards and embedded components during operation times without failure. The mobile application on the user end shall be easily accessible and easy to use for the end user without lags or unrecognized behavior.

- Recoverability.
  The hardware system shall be integrated smartly with the power supply to handle power black outs behavior. The hardware should manage to start itself, that is start and shutdown certain components when not in use, to track sun light, to reset rotations in the case of abnormal behavior. The mobile platform shall have swift fault response in the best possible friendly manner.

- Security.
  The software system shall ensure privacy of personal content that may be captured in the images. The system shall rely on existing technology to eliminate this information. The system shall also ensure that user personal information
- Regulatory. The system once installed shall not in any way violate existing privacy laws
- Environmental. The system shall utilize green energy for operation.

4.3 System stakeholders.

Specific types of individuals shall help to identify the requirements and test system for feasibility.

- Motor operators/drivers including taxi operators and clients, and basic day to day road users.
- Traffic Authorities.
- Emergency vehicle operators.
- Pedestrians.

4.4 Technical environment.

The surveillance system shall run on various architectures.

- The hardware end shall use a microcontroller to control operations on the cameras and the motors using instructions compiled in python language.
- The server shall run on C-Sharp based languages to offer interfaces for handling posts and requests at the end user platform.
- The mobile platform shall run on Android or IOS mobile operating systems to provide end users with the required information.

4.5 Domain constraints

The traffic surveillance system environment poses traits specific to it that may hinder its operation.

- The system should be operational 24/7, in areas where the sun is in low supply, new mechanisms should be introduced such as sunlight tracking to ensure batteries are charged at optimum levels.

4.6 Elicitation methods

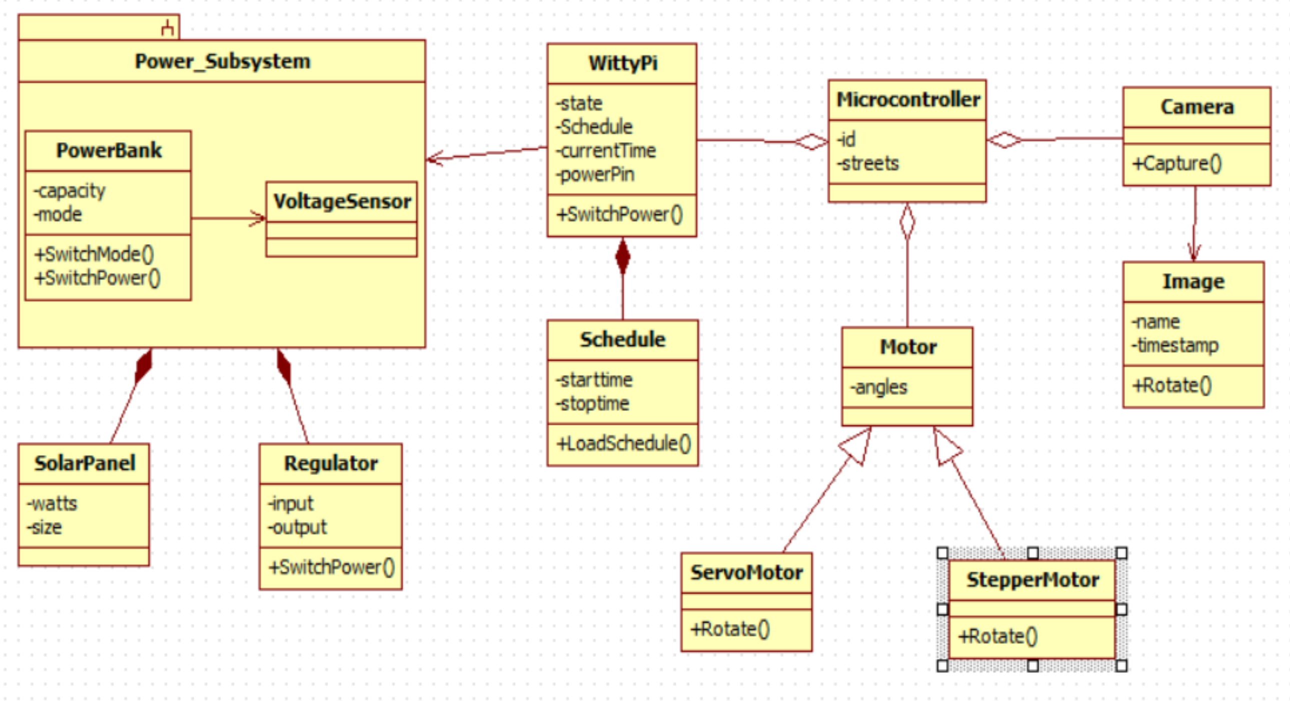
To define the requirements of this system, certain methods had to be implemented so as to compile, assess and review end product requirements.

- Focus groups. To determine the reaction of end users on the end product, small groups are formed to revise the traffic congestion issue and how best this solution can assist. Notes taken assist in the design and engineering process.
- Interviews. Road users are asked a set of questions concerning traffic congestion and how it affects their day to day lives. They are then introduced to the idea of having a mobile application that can assist in eliminating some of the effects of traffic jam. Their response is vital.
- Brainstorming. Certain situations arise that require one to try out random new things as thoughts suggest.

5 Analysis

5.1 Analysis of various system functionalities

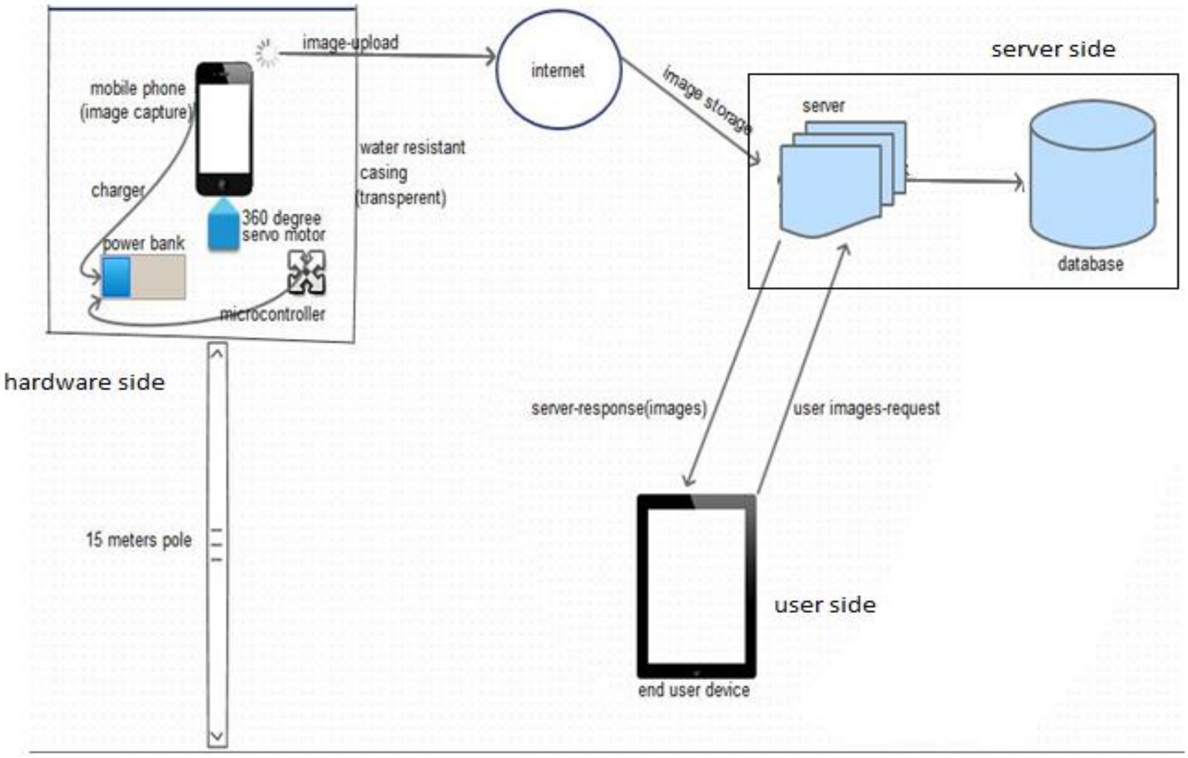
Analysis of the image capturing process



Junctions through research have proven to be the root cause of traffic congestion. A minor accident at times as well could cause a jam on a busy road. Other causes include road maintenance works, road upgrades, police checks among others.

These places are the best place to install a camera, However, because of the uniqueness of different junctions and busy roads, the capturing process must however be dynamic and flexible.
To achieve above, the following are some of the ways it can best be done;

- Use individuals to stand on busy road junctions with mobile devices to capture and upload status of traffic through a mobile application to an online server for different bust areas around Uganda. This option would be the easier and most reliable since chances of failure are mitigated by these individuals. It would provide employment to people in the long run. It however appears to be expensive to accommodate labor of over 100 people to do this job daily in this current economy.
- Using multiple surveillance cameras pointing in different directions of the road junction on raised ground. Each camera records event of the road it is surveilling. These surveillance cameras however are costly as they require an high quality HDD storage and cable connections.
- Mount an android phone on a motor and rotate it, enclose it and pin it on a pole. It is preloaded with most of the features required for the operation such as a Camera, storage, processor, 3g, Wifi, Bluetooth, battery, GSM, among others. An android application can be written specifically to capture and upload images using the phone’s hardware components.



- Constructing a custom-built device (set of components) interconnected and pre-programmed to capture images, upload them onto a server. using a motor to rotate one camera both being controlled by a microcontroller.

Conclusion:
Between all the above options, the smart choice would be to design an architecture with a set of components designed for this specific job. The system should be smart, recoverable and reliable.

Analysis of availability of the System to the end user.

The system must be readily available to the end user. Once a request is sent to the server for image content, that data should be updated and available to the end user as required. There should be no failures or faults to hinder this exchange. Camera hardware unit should be capturing, rotating, and uploading images as required.

To achieve this, the power supply has to be one of the following;

- Solar energy: Green energy utilization offers the opportunity to contribute to the environments development. Besides that, it is considered a cheap option because of its ability to draw energy from the sun to rechargeable batteries. A solar panel can be used for this project along with a solar power regulator to moderate current flowing to the batteries. This ensures that the system will always be on as long as there is power in the batteries. System operations can be made in short time intervals with auto system power ups and downs.

The battery inputs and outputs have to be smartly monitored to ensure continual performance of the system without utilizing a lot of energy.
The solar panel would need to be about 10 watts of power and exposed to the sun for at least 6 hours per day to ensure reliability. The solar panel requires a regulator to regulate current flowing to the batteries. Show ref
To make this option more efficient and reliable, motors can be mounted on the edges of the panel. This helps to achieve small rotations overtime as the sun sets from the east to the west.

- Another alternative would be to use direct current. For each camera installed, power lines have to be available to support operations. This option seems the best as it is more stable and reliable. This however requires extra road work and makes the project cost expensive. Direct current poses more threat to failure because of short circuits on the main grid. Direct current requires purchase of long cables to draw current. It results to monthly bills to the public electricity authority. In Uganda, power cuts have a 20% percent chance of happening in a day. Ref. Without back up batteries, this inconveniences surveillance operations.

Conclusion
As much as direct current seems to be the obvious and safest option, the evolvement of solar technology is promising. Green energy utilisation is a great alternative for this kind of project.

Analysis on eliminating faces and vehicle plates from images

To ensure harmony, privacy and confidentiality during operation, the traffic surveillance system must ensure the public is not affected in any way possible. No harm should be derived from the content of the data collected for this task.
The following are some of the alternatives.

- To use Open CV or Google python libraries on the microcontroller to eliminate faces from each image immediately after capturing it. This option proves to be safe since all data transmitted to the Server is already clean and meets privacy standards.
- Another alternative would be to upload captured images to the server. Once images are received, they are stored and put in a threaded queue on the server for processing. The server has more processing power and resources to carry out this function. The Server relies on either Google cloud APIs to process the images or Open CV library.
- The other option would be to perform the elimination process on the end user's device. Before an image is displayed, it is processed through the mobile application to ensure no private content isn't displayed to the end user.

Conclusion
From analysis, using the server seems the most convenient option.

Uploading images to the server over a network

All data captured by the camera is deemed important in the traffic monitoring process. Each image captured should be uploaded instantly to the main server. Failure of this or rise of any errors or faults deems the solution unreliable.

The following are some of the options available to process and transfer data to the main server;

- Using inbuilt WIFI on a Raspberry Pi microcontroller to transfer data over a nearby active internet connection. This option would require the image data to be added as base 64 parameters on a URL post request.
- In a developing economy like Uganda, WIFI is still an evolving technology and thus unreliable and limited. An alternative would be to partner with a telecom operator to provide a 3g connection to each camera installed around. This option requires attachment of a 3g module on each Raspberry Pi microcontroller. Reliable as this would be, it is expensive to undertake.
- The Raspberry Pi offers a LAN interface, data packets can be transmitted through LAN cables to nearby server for processing. This option appears to offer the fastest data transfer rate and throughput.
- Another option would be to transfer data captured to nearby remote servers over WIFI on a private network, there on handling the processing and uploading of images to the main server.
- Incase a cheap smartphone mobile device is being used to capture data, it makes the entire process much easier and smoother. It comes preloaded with WIFI and 3g shield, thus providing more flexibility.

Conclusion
Using WIFI seems to be the easier option. The hotspots however must have strong signal and bandwidth enough to support transfer of data packets at any time of the day. In Uganda, internet strength from service providers tends to deep in speed and performance during the day compared to night time.

Analysis of data integrity and management

All data collected through operations of the surveillance system is sensitive. Before the images are stripped of faces and number plates, it is important that they are kept in secure locations inaccessible to the general public or malicious hackers. Harmony is a very big requirement for the successful operation of this system.
The data has an expiry period of 24 hours, after expiry, an image should be disposed safely to not only recycle storage capacity, but to ensure safety. This data is also disposed because it is not used for security surveillance but rather for the benefit of the public.

Storage options for this data may include;

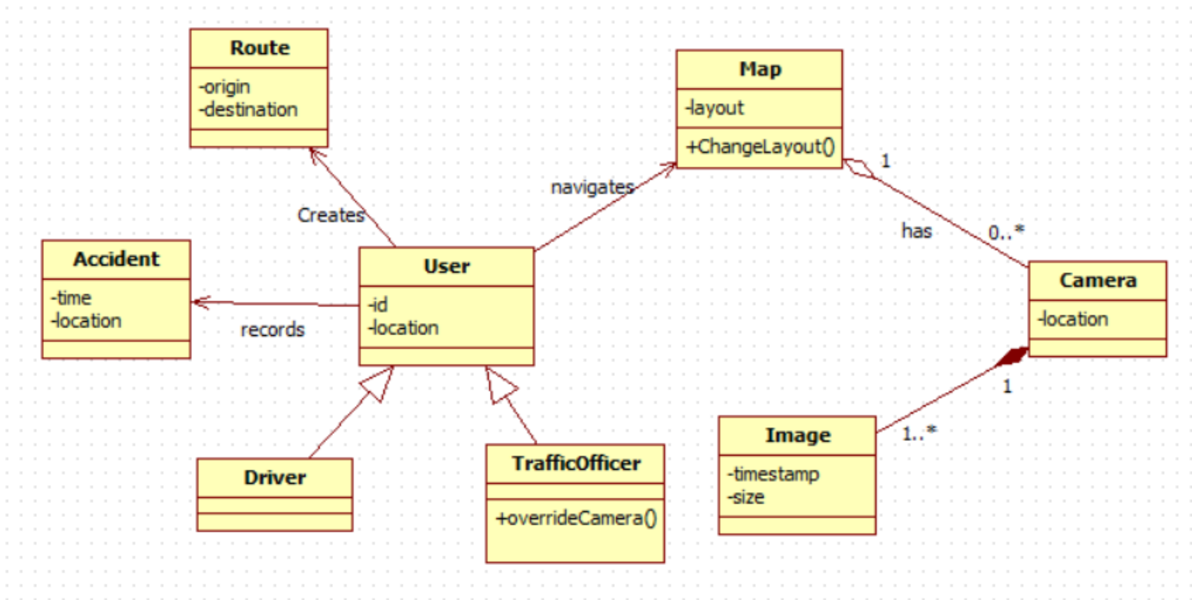
- Dropbox is a publicly available cloud storage medium. It already ensures for authentication of requests and provides developer interfaces. This therefore makes development of the mobile application easier since images can be accessed over these fluent APIs. The medium provides content management options such as uploading, deleting, editing, and archiving of data.

- Developing a custom-built server. This server would be responsible for handling posts from the different cameras, editing data, storing, and deleting the data. The Server however must be accessed from any location online since cameras are installed in many remote locations.

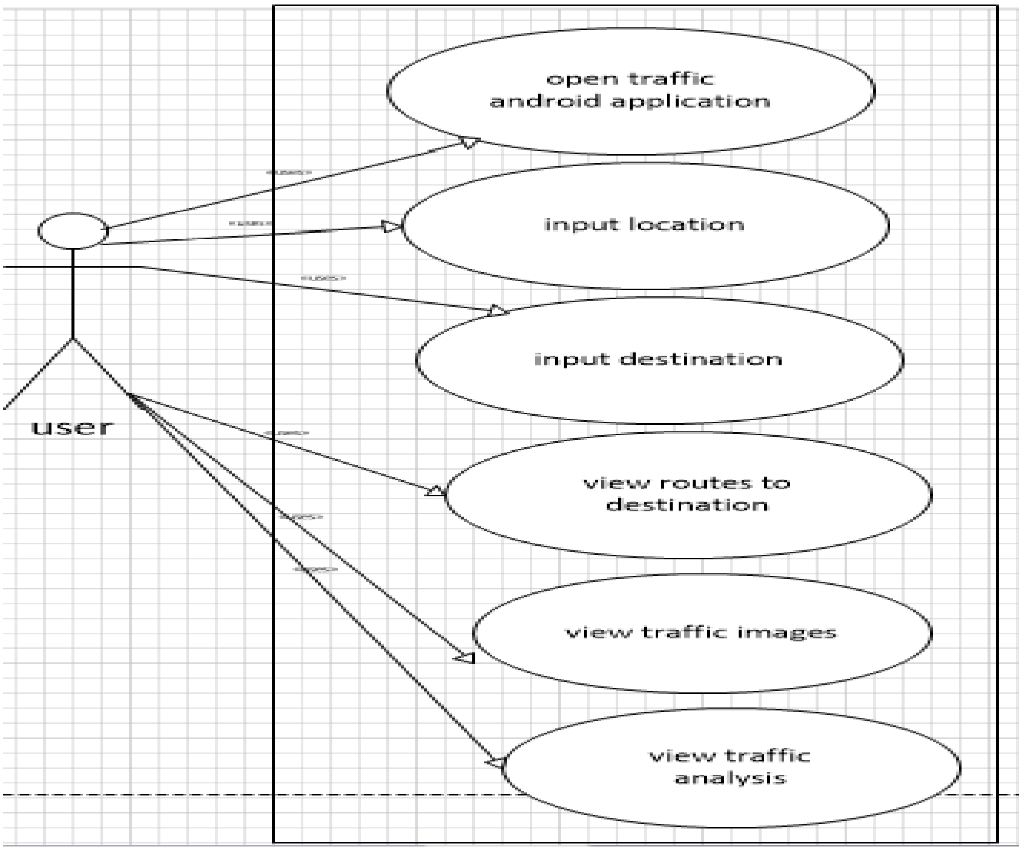
Conclusion

A custom online server is a long shot especially with security requirements. All requests must be on a secure network to achieve success. Dropbox or any other online cloud storage mediums require payments. Technical and legal complications may as well arise with clouds during upgrades and scaling of the surveillance system.

Analysis of the end user experience.



The end user must see frequently updated images evidenced by time stamps. Each camera on the map shall represent a junction on the map showing processed images of that area.



5.2 Use case Narratives

Table 3: Use case narrative for the Open application case

Use case name	Open application
Actors	Driver, Police traffic officer and traffic warden
Preconditions	Download the road traffic surveillance system on a device.
Main flow	For the driver, police officer or traffic warden to open the application, they should have downloaded it on their devices. Device can be a mobile phone or tablet.
Alternative flow	If the user doesn't download the application, he or she will not be able to open it.
Post condition	1. View different routes 2. See camera positions 3. Set destination 4. View route images

Table 4: Use case narrative for the See camera position case

Use case name	See camera position
Actors	Driver, Police traffic officer and traffic warden
Preconditions	1. Download road traffic surveillance application 2. Open the application 3. Set destination
Main flow	For a user to see the camera positions, he or she should have

	downloaded the application, opened it and set his or her destination
Alternative flow	The user can view the route images and then see the camera positions on that route
Post condition	1. Select a camera to use 2. View the route on which the camera is placed

Table 5: Use case narrative for Choose a destination case

Use case name	Choose a destination
Actors	Driver
Preconditions	1. Download the road traffic surveillance application 2. Open the application
Main flow	A driver chooses a destination after he or she has downloaded the application and opened it.
Alternative flow	
Post condition	1. See camera positions 2. View route images

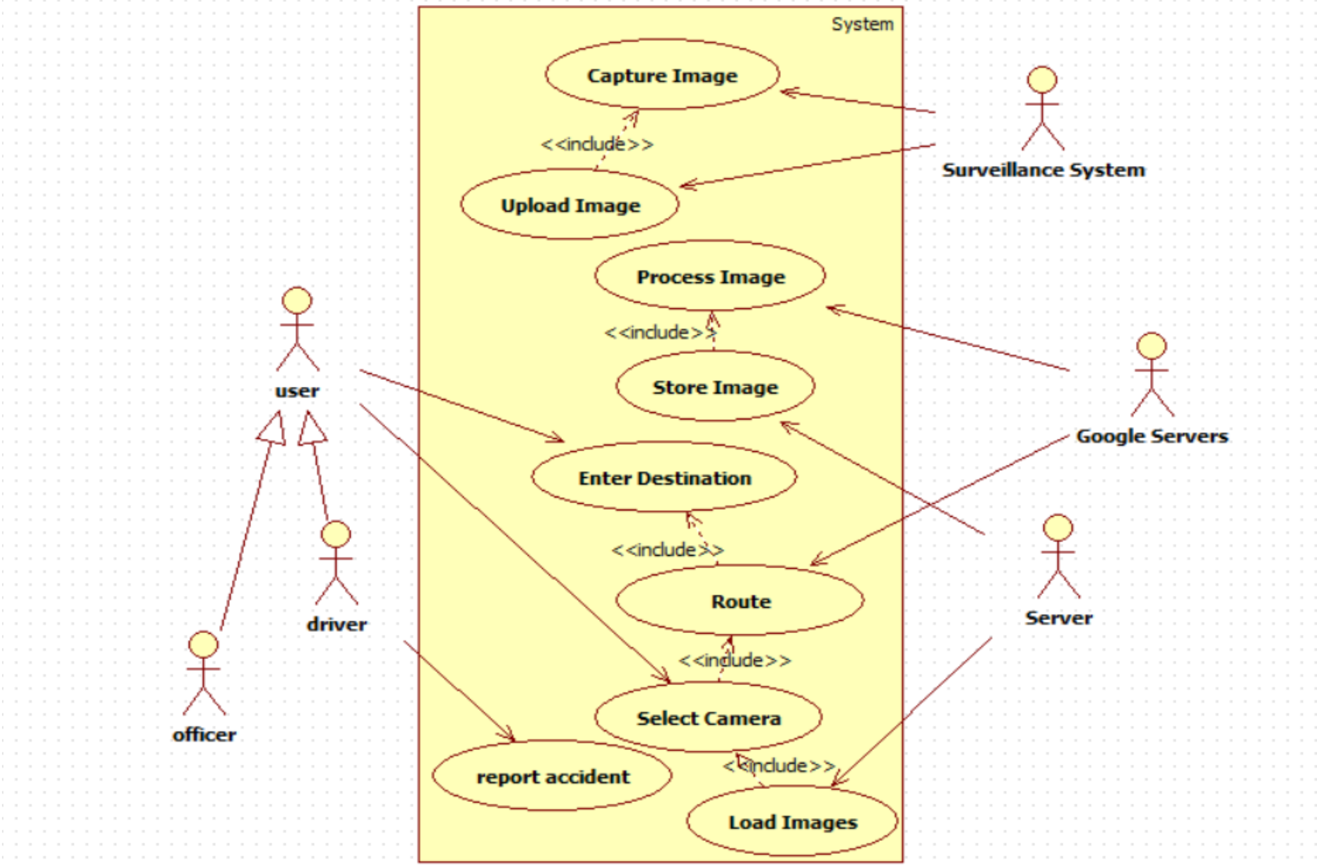
Table 6: Use case narrative for See routes to destination case

Use case name	See routes to destination
Actors	Driver, Police traffic officer and traffic warden
Preconditions	1. Download the road traffic surveillance application 2. Open the application 3. Set a destination
Main flow	For a user to see routes to destination, he or she should download the road traffic surveillance application, open the application and the set his or her destination.
Alternative flow	
Post condition	1. View route images

Table 7: Use case narrative for View route image case

Use case name	View Route images
Actors	Driver, Police traffic officer and traffic warden
Preconditions	1. Download the road traffic surveillance application 2. Open the application 3. Set destination 4. See routes to destination 5. View camera positions
Main flow	A user should 1st download the road traffic surveillance application, open it, set his or her destination, see the routes on that destination, view camera positions and the view route images on that camera.
Alternative flow	
Post condition	1. Close the application.

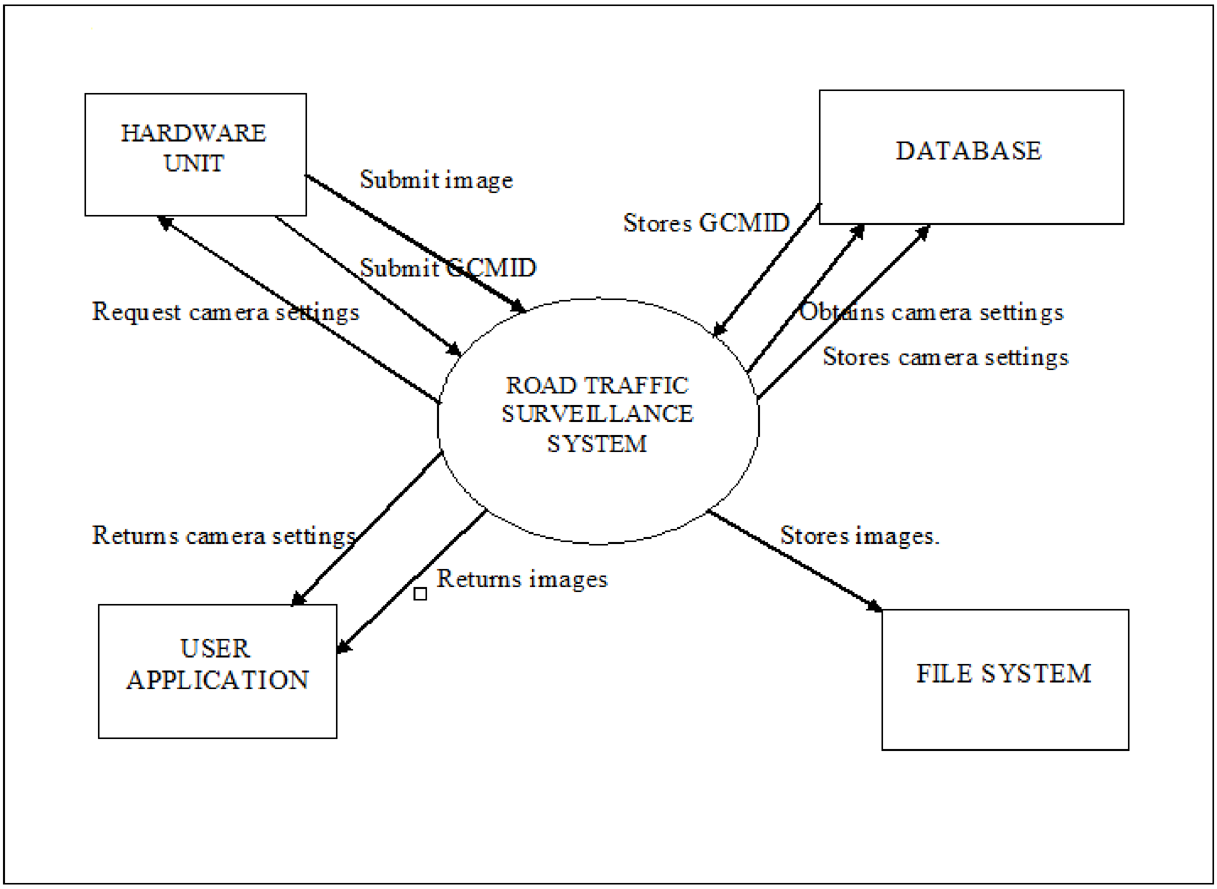
Overall system use case



6 Design

“Given that software designs are relatively easy to turn out, and essentially free to build, an unsurprising revelation is that software designs tend to be incredibly large and complex.” Jack W. Reeves, et al (2005).

To achieve a low unit cost, the design of the traffic surveillance system must be critically revised.



6.1 Hardware design.

To achieve a 24/7 life time span for the hardware units, solar energy comes in handy. Considering Uganda is a tropical country with temperatures of the sun shines averaging at 22 degrees Celsius for an average of 6 hours in a day, raises the chances of the availability of the software system.

A 10 watt solar panel is used to charge a rechargeable power bank and 24 volt batteries. All these sub components are interconnected with a solar power regulator which ensures current flowing from the solar panel doesn’t affect the batteries.

With consideration that most batteries get drained when they aren’t charged in a long time, and can’t power on unless manually operated, a special type of power supply must be devised. The voltaic 4000maH power bank comes with an always on mode that is switched on by pressing the power button for six seconds. This feature ensures that even without charge for a long time, the power bank can still automatically resume operation without any manual human interaction. The power bank can be paired with a 24-volt rechargeable battery to provide extra power back up in desperate scenarios.

A microcontroller is used on the hardware end to controls most modules such as the stepper motor, the servo motor and the camera module. The microcontroller used for this system is a Raspberry Pi 3 Model B. It comes within built WIFI 802bg/11n and Bluetooth 4.0, 4 USB ports, one LAN port, a GPIO interface and a camera interface.
The microcontroller is very key on the hardware design of the system. It is loaded with a set of instructions for each of the components. It has instructions for the following functions.

- To capture an image using the Camera module.
- To label and store an image with a time stamp.
- To upload each image captured over the network to the main server using URLs.
- To provide response from the server on the status of the post.
- To rotate the servo motor using pulse width modulation.
- To control stepper motor movement and rest from one point to another before an image is captured.

The microcontroller relies on a board called a Witty Pi 2 module for automatic start up and shutdown. The Witty Pi is mounted onto the GPIO pins of the microcontroller and power supply is redirected through it. The board comes with a battery that is used to keep the current time. With that in memory, the Witty Pi can easily power up and shutdown the Raspberry Pi at scheduled times or in intervals. A set of instructions is written and compiled for the Witty Pi to follow. For a 5-minute interval, the instruction would be in this format;
5 MIN ON: 3 MIN OFF

With the Witty Pi connected, power consumption on the Raspberry Pi is easily reduced. It also ensures safety of the microcontroller and its modules in cases of unexpected power faults or failures.

The microcontroller uses an active internet connection to upload captured images. The internet connection can be either over WIFI or a 3g connection.

The system has been designed such that for each image captured, an upload takes place automatically. Therefore, assuming 4 images are to be captured in different angles, for each image captured, a URL post is made immediately. This proves to be much more efficient that capturing all at once and uploading all at once.

Modules
Camera Module: The camera used to capture the images has 8 megapixels and is portable. It is easy to connect to the raspberry pi and can operate without a preview screen. This camera is mounted on mounts of a servo motor which are pinned on a stepper motor. The camera is very sensitive to noise and therefore images may be distorted accordingly.

Stepper Motor: The stepper motor uses 4 GPIO pins to achieve rotation. It also uses one 5v pin on the Raspberry Pi and one ground pin. Without the stepper motor, 360 horizontal rotation of the camera is impossible.

Servo motor: The servo motor is used to control vertical tilt on the camera. This is mostly required on valley like roads, or slanting roads. With this ability, the camera can be tilted easily for different angles of different roads.

6.2 Server design

The Server is a key component in the design of the traffic surveillance system. It is the central processing unit of the whole system. It is used for the following:

- To provide an interface to the camera units to upload images and meta data. The images are attached on the URL request made to the server in base 64 format. The server shall authenticate and validate each incoming request before processing the attached parameters and data. For security reasons, this is mandatory to ensure data integrity.
- The Server shall be responsible for removing sensitive content (faces and vehicle number plates) in each image sent through from the cameras before being stored on to the hard disk storage. This can be done using pre-existing Google algorithms for faces and character recognition. The Server should also log any errors during this operation for later analysis.
- The Server shall also analyze images using Google algorithms to derive statistics about road junctions that may seem important to the end user when planning their trip.
- The Server shall also need to authenticate any requests for this processed data. Each user shall have a unique id to use to access this information. The ID shall be created immediately on application download and first time usage.
- The Server shall recycle all images that have reached the 24 hours expiry duration. These images are to be deleted off the storage to firstly create space for new images. Secondly, because the images are not needed for reference matters such as security.
- The Server will use asynchronous interfaces to manage numerous simultaneous requests from multitudes of end users. Requests for images will be queried using URLs. Each image will have it's unique URL address for both its thumbnail and main image.

6.3 Mobile platform design

The mobile application platform is built on a map layout. The map layout shall enable the end user to geographically interpret various locations of installed cameras.

The application shall be downloaded from either the Google play Store for Android versions and Apple store for IOS versions.
On first opening the application, a unique ID number shall be assigned to the user for verification purposes.

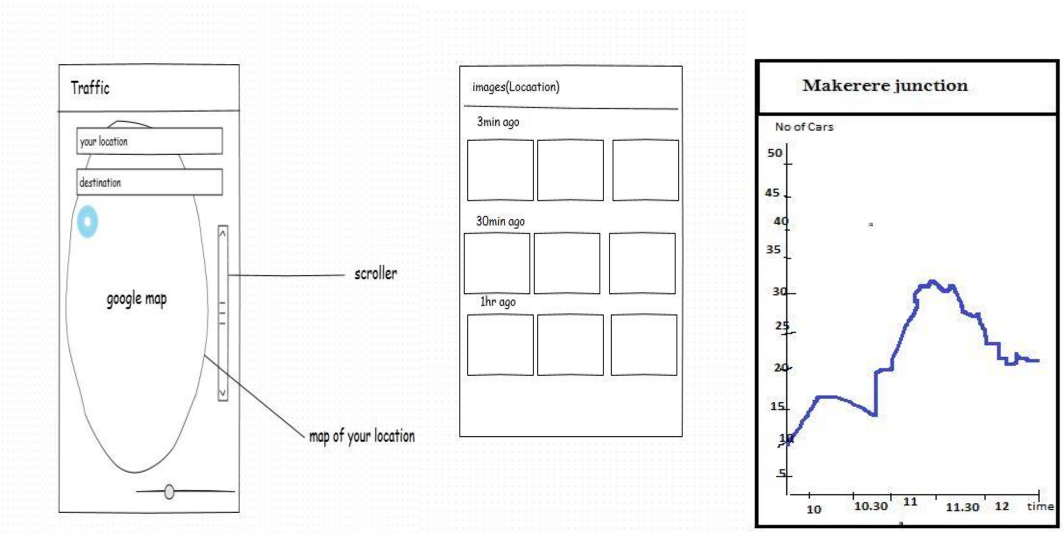
The mobile application shall be provide drag and zoom capability on to the map layout as they end user navigates different areas on the map.
The mobile application needs to be lightweight taking up less storage space while consuming limited processing power and cellular data on the end user's device. To do this, techniques such as threading can be utilised for example when downloading images among others.

The mobile application shall avail the user with fields to enter their current location as well as destination before their can perform a route. The application should be able to validate the data and strip it of unusual characters like slashes among others. It should also perform validation on required fields before routing the locations. Other validation may be on logging of an accident.

The mobile application shall utilise exceptions to handle any abrupt failures during execution. These exceptions should be logged and sent to administrators for review. Incase of any faults, the mobile application should have a recoverable trait so as to resume user experience.

The application shall only display available cameras once a route has been done successfully. Cameras displayed on the map shall only be those with locations close to the route. This enables the end user to clearly monitor traffic congestion on their route and make alternative decisions. The user shall be able to select a camera which will appear as an icon on the map. On click of the icon should pop a new window where the panorama of that junction or road is downloaded for the user to monitor the area.

Screen sketches for the end user application.



Screen Objects and actions

User location: Shows the user's geographical location.

User destination: Allows user to choose their destination by typing in where they are going and allows user to see what their current destination is.

Routing button: Displays paths the user can use to travel from their location to their destination.

Camera icons: Shows the various positions of the cameras which are available to the users. On Clicking the camera buttons users can see the images taken by that camera

Google maps v2: Displays a map showing the users current location, pin showing their current destination, routes, and cameras on those routes

Long clicking the map will set a new destination.

6.4 Design Rationale

The road traffic surveillance system is an original development, the choosing of the various components came after a long time of gambling with architectures that could provide the final desired system. Because of limited resources, critical decisions had to be made about all the hardware components to be used. The following design decisions have been taken during development of the road surveillance system;

- Existing camera technologies use ‘video’ data which requires a stable server and many resources to transmit the video data to the end user’s mobile devices hence the option of using ‘image’ data which can easily be transmitted and accessed by system users. These surveillance cameras also require wireless internet connection, for a system that will use many cameras, this seems to be an extreme in resources.
- Most surveillance camera transmit live video, this fact left us only with the option of developing a custom hardware device for capturing images.

This image shows 3 different cameras being used to capture images from different angles. These require wires to transmit video data. Instead we decided to use one camera to capture images from as many angles as possible.

Figure 7: Diagram showing a camera mounted on higher ground.

- Most existing image capturing cameras don't have internet capabilities and also cannot be programmed to upload images or also capture images at desired time intervals.
- For power supply, direct current was considered, but also this proved to be a very expensive option hence the option of the solar with a power bank. The solar power bank can offer fulltime power supply and it's cheaper. Electricity on the other hand can be cut off suddenly at any point.

7 Implementation

The implementation of the traffic surveillance system was an iterative process. One by one, modules had to be designed and improved. Currently, the prototype developed tries to meet the end-product requirements and therefore isn't fit for public use.

Operating systems (Environments) used for building the System Modules

- Raspbian for the hardware (Linux)
- Windows for the Server and Android mobile application

7.1 Hardware Implementation.

7.1.1 Languages used

- Python 2.7

7.1.2 Tools used for assembling the hardware

- Raspberry Pi 3 Model B. A board with wireless capabilities and a strong processor needed to control peripherals. It comes with 1 GB memory, a storage card slot, HDMI port and others.
- Tower pro Servo Motor. A servo motor has an output shaft that can be position to a particular angle or position through pulse width modulation (PWM), a coded signal sent through the control wire. Servos have a total rotation capacity of up to 180 degrees
- Stepper Motor. This motor comprises of a magnetic rotating shaft that contains electromagnets that provide flexible rotation. They provide very precise positioning and speed control using multiple coils organized in groups.
- Witty Pi. This is an extension board that helps to add real time clock capabilities and safe power management on the Raspberry Pi. With it you can properly shutdown and start up the microcontroller.
- Voltaic Power Bank. A versatile USB power back up designed to be used on solar projects. It comes with an always on feature that is very helpful for this project.
- 10-watt Solar Panel. Used to harness free and clean green energy needed to power the hardware components.
- Solar energy regulator. Needed to regulate current flowing from the solar panel to the power battery backup.

7.1.3 Activities

To connect a 4 wire bipolar stepper motor to a Raspberry Pi

Tools used

- a ULN2003A Bridge Board
- Raspberry Pi
- External Power source
- L298N H-bridge Board
- 12V DC Bipolar Stepper
- 4 x F to F Jumper cables

- 6 x M to F Jumper cables

Procedure

1. First, I Connected the stepper motor's bridge to the Raspberry Pi. I connected port IN1, IN2, IN3, IN4 of the bridge to PINS 16, 11, 13, and 15 respectively. For power, I connected the GND and the Positive on the bridge to PINS 6 and 4 respectively.
2. Next, I connected the bridge to the stepper motor.
3. I booted up the Raspberry Pi
4. I then wrote a script to control the stepper motor movement.

Code snippets for rotating the motor.

The code snippet shows a method to rotate the motor in any angle.

```
def move_to(self, angle):
    """Take the shortest route to a particular angle (degrees)."""
    # Make sure there is a 1:1 mapping between angle and stepper angle
    target_step_angle = 8 * (int(angle / self.deg_per_step) / 8)
    steps = target_step_angle - self.step_angle
    steps = (steps % self.steps_per_rev)
    if steps > self.steps_per_rev / 2:
        steps -= self.steps_per_rev
        print "moving " + `steps` + " steps"
        if self.mode == 2:
            self._move_acw_2(-steps / 8)
        else:
            self._move_acw_3(-steps / 8)
    else:
        print "moving " + `steps` + " steps"
        if self.mode == 2:
            self._move_cw_2(steps / 8)
        else:
            self._move_cw_3(steps / 8)
    self.step_angle = target_step_angle
```

Issues

I struggled initially to connect the cables from the bridge to the raspberry pi. I misinterpreted the PI board architecture GPIO to set the correct mode that fits the connection.

To connect and control Pi camera using the Raspberry Pi camera port.

Tools used

- Camera
- Raspberry Pi

Procedure

1. To install the Raspberry Pi Camera module, I inserted the camera cable into the Raspberry Pi. I slotted the camera cable into the connector located between the ethernet and HDMI ports of the Raspberry Pi board, with the silver connectors on the cable end facing the HDMI port.
2. Connecting the board to external power source to boot the raspberry pi.

3. From the pi terminal, i ran a command “sudo raspi-config”. Initially the camera drivers weren’t installed, so I ran more commands “sudo apt-get update” and “sudo apt-get upgrade”. Running the first command again showed the camera option.
4. I navigated to the camera option and enabled it, and thereafter rebooted the Raspberry Pi.
5. To capture an image, on the terminal, I ran a command “raspistill -o traffic.jpg”. The image can be called any other name.

Code snippets for capturing images

To capture 4 images therefore in 4 distinct angles, you can use the sleep function on the Time class object and a for loop to capture 4 different images with each name stored in a position of an array. The snippet below shows a method to capture and upload an image

```
import time
import picamera
import datetime
import base64
import requests
import urllib
import json

def get_file_name():
        return datetime.datetime.now().strftime("%Y-%m-%d_%H.%M.%S.jpg")

cam = picamera.PiCamera()
cam.resolution = (2592, 1944)
Filename = []
Images_64 = []
Image_path = []
def TakePhoto(fileName):
    time.sleep(1)
    imgpath = "/home/pi/photo/" + fileName
    print("file name = %s" % (imgpath))
    cam.capture(imgpath)
    image_64 = str(base64.b64encode(open(imgpath, "rb").read()).decode("ascii"))
    payload = {"Image64":image_64, "ImageName":fileName,"CameraId":"Humber360"}
    headers = {'Content-Type': 'application/json', 'Accept':'text/plain'}
    r = requests.post("http://192.168.0.8:8025/api/account/Img", json = payload, headers = headers)
```

Precautions

- The camera can be destroyed by static electricity if not carefully powered.
- I made sure I discharged myself by touching an earthed object

Issues

- The first camera module I purchased was defected and none of the above commands worked. For a long time I struggled to test it until I made conclusion the module was dysfunctional.
- The camera module cable was too short. This affected the rotation process.
- The camera cable is designed such that images captured can be affected if the camera is deployed in a noisy environment.

To connect a Servo motor to a Raspberry Pi

Tools used

- Tower Pro Servo Motor
- 3 jumper cables
- Raspberry Pi

- External power source

Procedure

- I connected the jumper cables to the servo motor. The power, ground and control pins of the motor were inserted on the Raspberry Pi GPIO board PINS 2, 9 and 8 respectively.
- Next, I compiled a script to control the servo's rotation.

Code Snippets

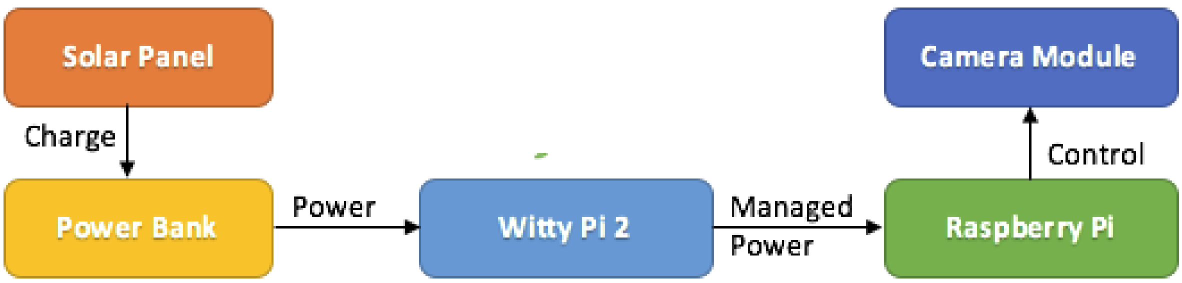
The code below uses pulse width modulation to send signals to the servo motor. The duty cycle ranges from 0 to 12 signifying angles 0 to 180 degrees.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
servoPin = 8
GPIO.setup(servoPin, GPIO.OUT)
pwm = GPIO.PWM(servoPin, 50)
pwm.start(4.5)

try:
    while True:
        pwm.ChangeDutyCycle(7.5)
        time.sleep(3)
        pwm.ChangeDutyCycle(9.5)
        time.sleep(3)

except KeyboardInterrupt:
    pwm.stop()
    GPIO.cleanup()
```

Issues

The Servo motor shakes especially with angles close to 0 or 180 degrees. At times, when the Raspberry Pi gets powered, the servo motor exhibits an unusual behaviour.

How to connect Witty Pi 2 to Raspberry Pi GPIO pinout board



Tools used

- Witty Pi 2
- Voltaic Power Bank
- Raspberry Pi 3
- Solar Panel

Procedure

1. I began by mounting the Witty Pi onto the Raspberry Pi GPIO pinout. I ensured that the camera module has a gap to go out.
2. Using a USB cable, I connected the Witty Pi to the power bank output port.
3. Then I connected the power bank to the Solar panel using charger cables running from the solar power regulator.
4. I booted the Witty Pi by pressing it’s switch
5. I created a schedule to control the Pi power states.

Code snippets

Below is a snippet showing a schedule created to shut down the Raspberry Pi after 5 minutes and start it up after 3 minutes.

Issues

Initially, I had installed an operating system that has a splash screen before the actual operating system loads The Witty Pi requires that the operating system loads first time. Therefore, I had to load the memory card with a new Raspbian operating system using disk image.

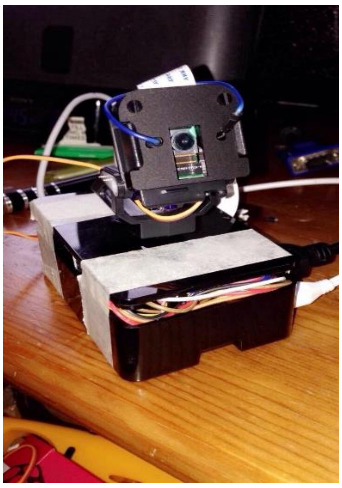
I also had to transfer already established connections from the Raspberry Pi GPIO to the Witty Pi GPIO.

Every time I tried to power up the system, it would shut down immediately. The Witty Pi uses GPIO pin 7 to control power states, which I was initially using to control the stepper motor. I struggled for a while since I initially had no knowledge of this. I therefore had to transfer from PIN 7 to PIN 16.

Mounting the camera modules together

Procedure

1. I had to screw the servo motor on to the pan-tilt frame.
2. I inserted the Pi Camera into the same frame for firm grip.
3. I mounted the frame on to the Stepper motor.
4. I then put the Raspberry Pi with the Witty Pi, the stepper motor and it’s bridge inside a small box to carry out real tests.

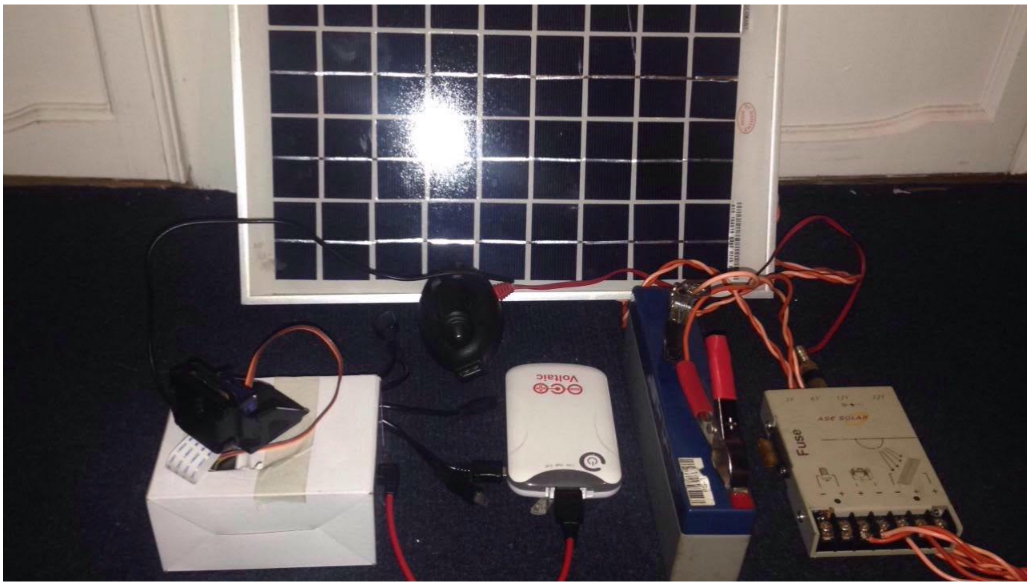


Issues

I failed to find a stable mount to hold the pan-tilt frame onto the stepper motor. I tried to use gum and wires but it still wasn't stable. The frame kept falling off particularly when the motor would rotate to far angles towards 360.

The stepper motor also tends to take the nearest route. For example, if it was in an angle 280, and It's next destination is angle 10 degrees, it would move anti-clockwise thus messing with the camera cable.

Final Assembly



7.2 Software end implementation

Developing the Server

Procedure

- Modelling the data to be used
- Creating controller Get and Post APIs
- Creating authentication methods for accessing the server.
- To create methods for sending requests to third parties.
- Assigning routes to the server controller methods
- Preparing storage locations for the different data types

Code Snippets

The snippet below contains a method to get an image using a byte array. The method requires the request to have the file name parameter.

```
public string ImgFolder()
{
    return string.Format("/content/uploads/userdata/{0}/imgs", GetSpot().SpotId);
}
[HttpGet]
[Route("Image/{fileName}")]
0 references | 0 requests | 0 exceptions
public HttpResponseMessage GetFile(string fileName)
{
    // var trialImage = "Far_Gosford_St.jpg";
    var filePath = HttpContext.Current.Server.MapPath(ImgFolder() + "/" + fileName + ".jpg");

    // Bitmap response;
    Bitmap responseThumb;
    try
    {
        responseThumb = new Bitmap(filePath);
        //responseThumb = (Bitmap)responseThumb.GetThumbnailImage(50, 50, null, IntPtr.Zero);
        using (MemoryStream ms = new MemoryStream())
        {
            responseThumb.Save(ms, System.Drawing.Imaging.ImageFormat.Png);
            HttpResponseMessage result = new HttpResponseMessage(HttpStatusCode.OK);
            result.Content = new ByteArrayContent(ms.ToArray());

            result.Content.Headers.ContentType = new MediaTypeHeaderValue("image/png");
            return result;
        }
    }
    catch (Exception ex)
    {
        return Request.CreateErrorResponse(HttpStatusCode.NotImplemented, "Failed to load image" + ex.Message + " " + filePath);
    }


}
```

The snippet below shows a data model class for an image. The class has a method to convert an image to base 64.

```
public string Bio { get; set; }

1 reference | 0 exceptions
public string Img { get; set; }
1 reference | 0 exceptions
public string Url { get; set; }

1 reference | 0 exceptions
public List<string> AllImages { get; set; }
0 references | 0 exceptions
public string ImageToBase64(string path)
{
    using (System.Drawing.Image image = System.Drawing.Image.FromFile(path))
    {
        using (MemoryStream m = new MemoryStream())
        {
            image.Save(m, image.RawFormat);
            byte[] imageBytes = m.ToArray();
            string base64String = Convert.ToBase64String(imageBytes);
            return base64String;
        }
    }
}
0 references | 0 exceptions
public string ImgFolder()
{
    return string.Format("/content/uploads/spotdata/{0}/imgs", Id);
}
```

The snippet below converts and image from base 64 as it's posted from the camera end.

```
public void WriteImage()
{
    // Create the folder if it doesn't exist
    FileService.CreateFolder(User.ImgFolder());

    byte[] imageBytes = Convert.FromBase64String(Image64);
    using (var im = Image.FromStream(new MemoryStream(imageBytes)))
    {
        ImageFormat frmt;
        if (ImageFormat.Png.Equals(im.RawFormat))
        {
            //ImageName += ".png";
            frmt = ImageFormat.Png;
        }
        else
        {
            //ImageName += ".jpg";
            frmt = ImageFormat.Jpeg;
        }
        string path = HttpContext.Current.Server.MapPath(User.ImgFolder()) + "/" + ImageName;
        if (File.Exists(path)) {
            File.Delete(path);
        }
        im.Save(path, frmt);

    }

}
```

The snippet below contains code instructions to detect and cover text in an image using Google character recognition API methods.

```
public static void DetectDraw(string filePath) {
    // [START get_vision_service]
    var client = ImageAnnotatorClient.Create();
    // [END get_vision_service]
    // [START detect_face]
    var response = client.DetectText(ImageFromFile(filePath));
    //var response = client.DetectFaces(ImageFromFile(filePath));

    using (var image = System.Drawing.Image.FromFile(filePath))
    using (System.Drawing.Graphics g = System.Drawing.Graphics.FromImage(image))
    {
        var cyanPen = new System.Drawing.Pen(System.Drawing.Color.
            FromKnownColor(System.Drawing.KnownColor.Cyan), 3);
        Brush myBrush = new SolidBrush(System.Drawing.Color.Red);
        int count = 0;
        foreach (var annotation in response)
        {
                count++;
            if (count != 1) {
                g.FillPolygon(myBrush, annotation.BoundingPoly.Vertices.Select(
                (vertex) => new System.Drawing.Point(vertex.X, vertex.Y)).ToArray());

            }

        }

        int lastDot = filePath.LastIndexOf('.');
        string outFilePath = lastDot < 0 ?
            filePath + ".edit" :
            filePath.Substring(0, lastDot) + ".edit" + filePath.Substring(lastDot);
        image.Save(outFilePath);
    }
}
```

Issues

The algorithms used to detect faces can only be efficient if the image is of a high quality.

Developing the mobile application

- Designing application functionalities.
- Sketching application screens.
- Developing application layout
- Creating a connection class to the server
- Writing application logic.
- Debugging the application on an actual device.

Code Snippets.

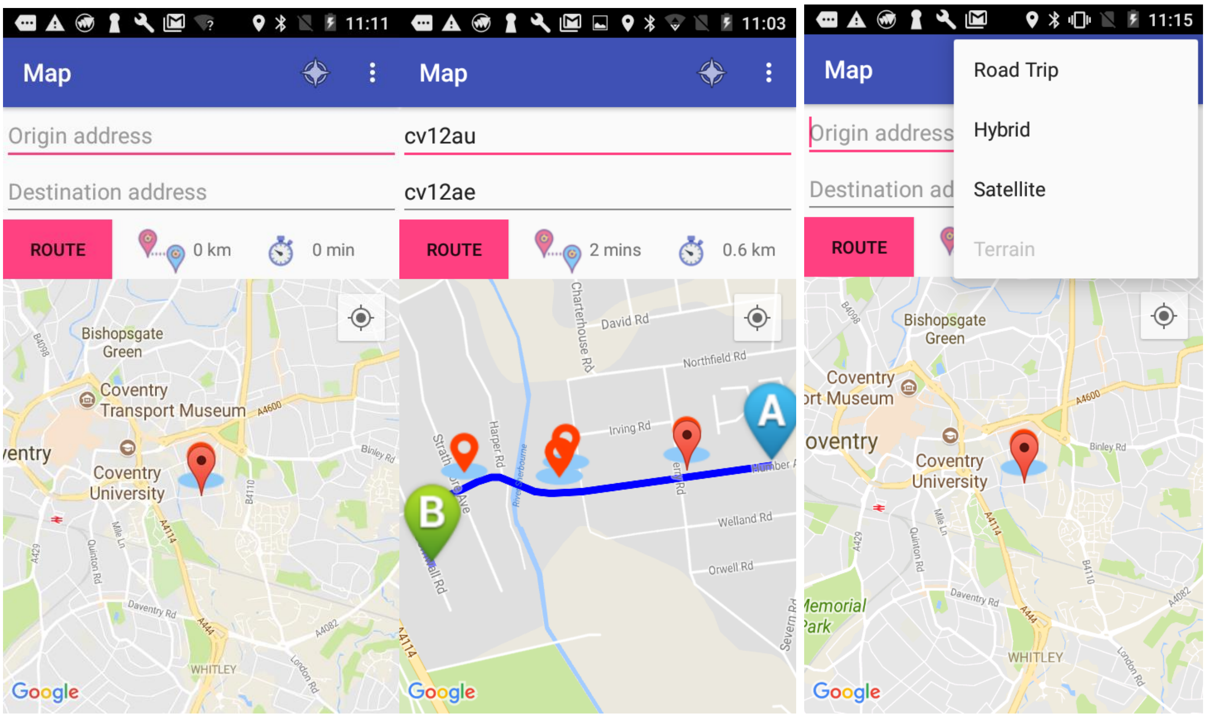
The snippet below shows a class designed to load images on a queue system.

```
public ThumbnailDownloader(Handler responseHandler){
    super(TAG);
    mResponseHandler = responseHandler;
}
public interface ThumbnailDownloadListener<T>{
    void onThumbnailDownloaded(T target, Bitmap thumbnail);
}

public void setThumbnailDownloaderListener(ThumbnailDownloadListener<T> listener){
    mThumbnailDownloaderListener = listener;
}
public void queueThumbnail(T target, String url){
    Log.i(TAG, "Got a URL: " + url);
    if(url == null){
        mRequestMap.remove(target);
    }else{
        mRequestMap.put(target, url);
        mRequesthandler.obtainMessage(MESSAGE_DOWNLOAD, target).sendToTarget();
    }
}

@Override
protected void onLooperPrepared(){
    mRequesthandler = (Handler) handleMessage(msg) → {
            if(msg.what == MESSAGE_DOWNLOAD){
                T target = (T) msg.obj;
                Log.i(TAG, "Got a request for URL: "+mRequestMap.get(target));
                handleRequest(target);
            }
    };
}
public void clearQueue() { mRequesthandler.removeMessages(MESSAGE_DOWNLOAD); }
private void handleRequest(final T target){
    try{
        final String url = mRequestMap.get(target);
```

The snippet below contains a method to load image bytes from a URL string

```
public byte[] getUrlBytes(String urlSpec) throws IOException{
    URL url = new URL(urlSpec);
    HttpURLConnection connection = (HttpURLConnection)url.openConnection();
    try{
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        InputStream in = connection.getInputStream();

        if(connection.getResponseCode() != HttpURLConnection.HTTP_OK){
            throw  new IOException(connection.getResponseMessage()+":with "+ urlSpec);
        }
        int bytesRead = 0;
        byte[] buffer = new byte[1024];
        while((bytesRead = in.read(buffer))> 0){
            out.write(buffer, 0, bytesRead);
        }
        out.close();
        return out.toByteArray();
    }finally {
        connection.disconnect();
    }

}
```

Application screenshots



7.3 System deployment strategy.

Before I can deploy the system, I must ensure the end product meets all the specified requirements.

7.3.1 Deploying the server

To deploy the server, I must pay for a domain on the worldwide internet and hosting services so that the mobile platform can access data remotely from anywhere around the world. The API calls structure can then be changed from local IP address to a normal web address.

7.3.2 Deploying the mobile application

The mobile application has been developed using an Android, an open source platform for mobile applications. For the application to be used by the public, I must upload the application to the Google Play Store where I have to specify it's purpose, application domain and provide sample screenshots. The store also offers me the ability to choose the countries in which the application operates.

7.3.4 Deploying the hardware

I must design a weather proof casing for the assembled hardware components. Thereon mount them on to a 10-meter pole. This pole will be made of steel and painted to prevent rust. For every pole, a hole has to be dug on the junction where surveillance will take place.
There has to be a nearby wireless active internet connection. For areas without wireless, I must add a 3g module to provide internet.
Once the hardware is installed, I shall use a program called "Putty" to access and control the Raspberry Pi remotely over SSH.

7.4 Version Control.

I utilised GitHub to track my project's progress and address existing issues. For every module is developed, I ensured to monitor process

7.5 Code

https://www.dropbox.com/s/uiveqwro8p47n6r/Trafficap-Code.rar?dl=0

8 Testing

The traffic surveillance system is made up of numerous sub modules integrated together for a common goal of capturing, uploading and processing image data that is used to monitor traffic behaviour on a road junction. Each of these units have to be tested independently to ensure that they are perfect for the designated task and can work seamlessly after system integration.

The testing stage sometimes takes a longer duration than other stages as bugs are uncovered, fix and the system components re tested over and over until the desired outcome is achieved. Hardware Components to be unit tested include the following;

- Raspberry Pi Microcontroller and it's hubs. It has to successfully control all connected peripherals while ensuring their safety during faults.
- The camera module which is used to capture image data using instructions from the microcontroller
- The stepper motor used for 360 degrees horizontal rotation.
- The servo motor used for 180 degrees vertical rotation.
- The power bank used to provide current on the hardware modules
- The solar panel and it's regulator, are used to charge the power bank, and the back up batteries
- The Witty Pi is used for scheduling on/off states

White box testing

This type of testing supports unit, interconnection and system levels of the system testing process. Each unit can be tested on it's own, there after tested on how it operates with other modules, and then overall system can as well be tested.

For each component mentioned above, testing can be done to uncover faults, to gauge performance, efficiency, and to ensure they meet enlisted requirements

8.1 Areas tested

Testing the rotation system on the Stepper motor.
The stepper motor is tested for seamless rotation, interpretation of instructions, strength and torque. On different power connections, the stepper motor seems to provide different performance. When using a 5v adapter to provide direct current, the stepper motor provides much better and seamless performance compared to the unstable solar power source. Integration tests were done to determine that the motor can firmly grip and rotate the servo motor and camera housing.

Testing the servo motor.
The Servo motor is supposed to rotate gently and swiftly in any direction. On testing for smooth rotation to an angle, it becomes unstable on certain pulses leading to extreme angles of 0 degrees and 180 degrees. Tests proved that a **bridge** is required to regulate current flowing on the servo motor.

Testing the Raspberry pi camera
Purpose of tests were to check for quality of images captured.

Unit tests done initially helped discover that the first camera module purchased had some faults as it didn't respond to the capture command on the terminal. Integration tests with the raspberry pi on the replacement module however responded swiftly to the rasptill command.
On system level, the test was done to ensure the camera cable remain intact even with frequent horizontal and vertical rotations.

Testing the Witty Pi reliability.
Testing was carried out to ensure the Witty Pi can store current times, shut down and power up the microcontroller safely.
No unit tests were done on this one as success can be gauged once its integrated with the raspberry pi.
First integration test produced abnormalities as the Witty Pi requires the operating system start up not to load the splash screen.
Once the new operating system was installed, the Witty Pi proved to meet all requirements expected of it. Manual and automatic shutdowns were safe and successful. Loading of new power state schedules was successful at first trial.

Testing the Solar power management system.
The solar panel is tested for durability and efficiency.
Integration tests involved putting the solar panel outside for 14 days while charging the backup batteries and the power bank through the solar power regulator.
Test results proved that the solar panel was efficient enough to carry out the designated task. It was however unreliable when there was little or no sunshine.
It was also discovered, the power bank unless switched to an always on mode can't resume charging when current flows again.

Testing the efficiency and speed of WIFI to transfer images over a network.
The raspberry pi has an inbuilt WIFI module, this enables transfer of images over the network.
Throughput and response time are tested when an image is sent from the Raspberry Pi to the main server using HTTP protocol.
Using python 2.7, the image is converted to base 64 format and attached to the URL as a key value pair parameter. Other meta data is added as well and a URL post request is made to the server's address on the specified API.
This test was a success as the image was received, processed and stored on the Server.

Testing the Server for scalability, throughput and response time.
The server needs to be able to handle multiple requests simultaneously.
To test for this, I had to use three separate mobile devices to make the same request at the same time and the local server seemed to perform reliably.

Testing facial recognition and character recognition algorithms.
Each image captured shouldn't contain any faces or numbers.
The first unit test on the server comprised of a random image. In C#, the google libraries for detecting faces are referenced. There on the location of the sample image is provided as a parameter along with the new file name.
Test results proved that the algorithms work swiftly for high quality images compared to low quality images.

9 Project Management

9.1 Project Schedule

9.1.1 Tasks

Hardware end tasks.
Analyse requirements to develop the hardware component.
Acquire the necessary modules to be used.
Test each component individually.
Assemble components.
Case components for safety.
Test entire unit performance.

Server end tasks.
Analyse the server required capabilities.
Develop data models.
Build server APIs.
Secure server access.
Test server scalability and performance on local network.
Upload the server to online network.

Mobile platform breakdown
Produce application scenarios
Sketch view layouts
Create data models
Develop application layouts
Implement application functionalities
Test application for usability and efficiency
Upload application to publicly accessible store.

9.1.2 Budgets

At project conception, the goal was to not exceed £100 per hardware unit assembled. I poses knowledge in various programming languages and therefore no budgets were made for design and development of software related modules.

Final costs

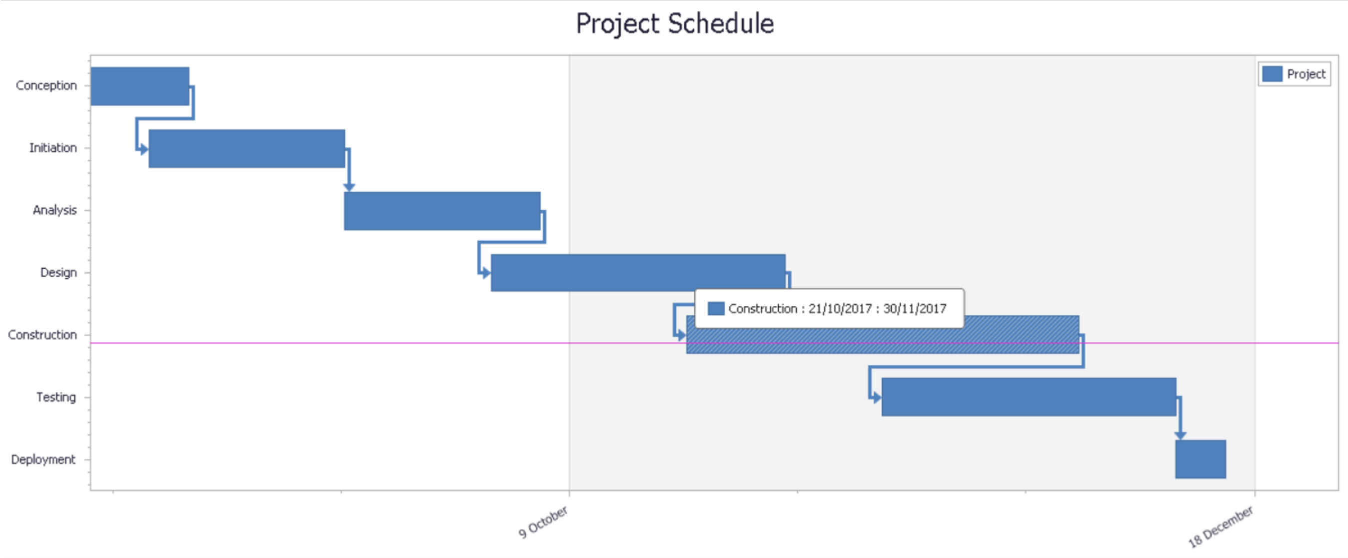
#	Item	Purpose and description	Vendor	Cost in pounds (£)
1	Raspberry Pi Starter Kit	Kit includes a Raspberry Pi board, a power adapter, a 16 gb memory card, a unit case, HDMI output cable, and a LAN cable. Needed to control peripherals such as camera, motors and others. Purchased from	The Pi Hut	£55.00
2	Witty Pi 2	Necessary to regulate power states on the hardware unit	UUGear on Amazon	29.99

3	Pan-Tilt HAT	For mounting servo motor and the camera for firm grip and rotation. Comes with a Pan-Tilt board loaded with necessary libraries.	The Pi Hut	£15.00
4	Raspberry Pi Camera Module V2	8mp camera	The Pi Hut	£24.00
5	150M miniature WIFI Module	Acts as backup incase of signal issues	The Pi Hut	£9.00
	Total			**£132.00**

9.1.3 Work breakdown structure.

Level 1	Level 2	Level 3
1 Traffic surveillance system	1.1 Conception	1.1.1 System Evaluation and Endorsements.
		1.1.2 Produce Project Charter and Ethics.
		1.1.3 Upload ethics and deliver charter.
		1.1.4 Review submissions.
		1.1.5 Approve ethics and charter.
	1.2 Initiation	1.2.1 Create preliminary scope statements.
		1.2.2 Determine project team.
		1.2.3 Kick off meeting
		1.2.4 Develop the project brief
		1.2.5 Submit project brief
	1.3 Analysis	1.3.1 Project brief analysis
		1.3.2 Requirements modelling
		1.3.3 Adjustments
	1.4 Design	1.4.1 Kick off Meeting
		1.4.2 Verify and approve end product requirements.
		1.4.3 Design system architecture
	1.5 Construction	1.5.1 Kick off Meeting
		1.5.2 Procure hardware and Software
		1.5.3 Assemble system
		1.5.4 Install system
		1.5.5 Risk Management
	1.6 Testing	1.6.1 Prepare system testing plan
		1.6.2 Debugging
		1.6.3 Fixing bugs
		1.6.4 Restructure system components
	1.7 Deployment	1.7.1 Kick off Meeting
		1.7.2 Upload to live server
		1.7.3 Installing hardware

Below are some of the milestones for this project and the scheduled start and finish dates.



No.	Milestone	Description	Start	End
1	Conception	The project conception stage is very critical to its success. Conceptualising ways on how to construct the system hardware, server and mobile platform required research, brainstorming, and revising current artifacts.	21st August 2017	31st August 2017
2	Initiation	Initiation starts before the conception ends. Ideas already formulated are initiated even before the whole project is thought through.	27th August 2017	16th Sept 2017
3	Analysis	Analysis of end product requirements and possible options took multitude of the project time.	16th Sept 2017	6th October 2017
4	Design	Designing the system architecture, data models and relationships. Changes in architectures affected the project schedule.	1st Oct 2017	31st Oct 2017
5	Construction	Implementing the designs took me some time. Some components were purchased faulty. This prolonged the construction process. Hardware	21st Oct 2017	30th Nov 2017

		assembly was the most outstanding in this phase.		
6	Testing	Testing the hardware, server and mobile applications took me the longest time as faults discovered needed to be fixed.	10th Nov 2017	10th Dec 2017
7	Deployment	Deployment process involved installation of cameras and the server, Uploading the mobile application on to the play store for public access was a swift process.	10th Dec 2017	15th Dec 2017

9.2 Risk Management

Risk Management was vital during the evolution of the traffic surveillance system. To breakdown, the task involved the following activities;
1. Risk Assessments
2. Risk Mitigation

9.2.1 Risk Assessments

Assessments were done mainly on the hardware architecture.
The following are the risks identified after making an assessment;

- The development tools utilised
- Project budgets and schedules
- System Productivity risks
- Increasing requirements list
- Compromising on designs
- Requirements collection techniques
- Standard of externally used systems

The following are some of the direct risks involved traffic surveillance system;

#	Risk	Probability (/1)	Severity (/5)	Priority
	Productivity risks	0.6	4	3
	Requirements elicitation methods	0.7	5	1
	Budgets and schedules		3	6
	Ever changing and growing requirements list	0.2	2	2
	Development tools used	0.4	5	4
	Compromising on designs	0.1	5	5

Probability Scale (of occurrence) 0.00-0.15: Almost sure not, 0.15-0.25: Not likely, 0.25-0.35: Not very likely, 0.35-0.45: Somewhat less than an even chance, 0.45-0.55: An even chance, 0.55-0.65: Somewhat greater than an even chance, 0.65-0.75: Likely, 0.75-0.85: Very Likely, 0.85-0.95: Almost sure, 0.95-0.99: Extremely sure, 1.00: certain
Severity 1: minimal, 2: minor, 3:Moderate, 4:Significant, 5: Severe

Risk Analysis

Productivity Risks
The traffic surveillance system poses a productivity risk in the context of helping road drivers dodge traffic jam while utilising the mobile driven platform.
The level of satisfaction has to be measured over time. If a road driver can manage to load and interpret image content on junctions and use that manoeuvre around traffic congested roads at any time of the day would conclude a success.
However, chances could be that the driver doesn't understand the image content and therefore makes wrong decisions leading him in a traffic jam.

Requirements elicitation methods
Using questionnaires to collect information on the problem at hand and the suggested solution may or may not be enough to produce perfect system requirements. Researching artifacts, conducting interviews, brainstorming are among the few other alternatives to elicit end product requirements.

Budgets and schedules.
Since the traffic surveillance system undergoes conception all through to deployment, it is very vital that the project schedule produced, revised and followed carefully. At times, projects such as these that involve invention and innovation tend to deviate for planned budgets and often exceed the schedule. New requirements arise which usually constrain the system design and development budget, other factors may include calls for change in system architecture which concludes some already purchased components useless.

Ever changing and growing requirements list.
In 80 percent of systems developed, customer and end-product requirements are likely to change or even protrusion of new ones. Not to be considered a calamity, but this disease tends to distract initial plans, affect budgets and schedules and at times demoralise human resource.

Development tools used
The tools utilised on the hardware end haven't been tested for traffic surveillance purposes. Using a Raspberry pi microcontroller for example may not be the most ideal way to control a motor and a camera. The tools used to design and build components pose a risk to the project credibility. Using UML notation to layout the system's logical data and module relationships may or may not be the ideal method. Special attention is required on data modelling and information flow especially with embedded systems involved.

Compromising on designs
The budget could be one reason certain components may be left out during system implementation. Because of such compromises, the final product faces a risk of fault or failure. Rather than cut costs, reliability must be a priority. The end results have to be aggressively analysed and calculated in case any compromises arise.

9.2.2 Mitigation plan

For all the risk identified that may affect the success of the traffic surveillance project, ways have been devised to handle the impact or even prevent the occurrence of the risk effects.

Risk transfer
Depending on the capacity of a parties involved in the traffic surveillance system, the risk can be shifted to the best candidate best possible to suppress the impact of the risk. During project

development for example, risk of productivity concerning privacy was shifted to the Google cloud APIs.

Risk avoidance
Some risks cannot be afforded and therefore the best mitigation technique has been ideally to eliminate their chance of occurrence. Changing key parameters connected to that risk is the solution to best address these risks. In the design of the surveillance system, after realisation that backup batteries often get drained, the alternative was to use an always on power bank. Using solar energy over direct current ideally reduces the development and installation cost for a camera unit. A quantitative analysis to assess the productivity of solar energy over direct current helped to mitigate this risk.

Risk control
Most risks cannot be fully suppressed. With that in mind, I devised several ways necessary to manage the impact or likelihood of these risks. The surveillance system requires an active internet connection to be able to upload and communicate with the server. To control the adverse effects of the risk of no internet connection involved me purchasing an extra WIFI dongle that connects to a backup connection. In the case that one is down, another will be available to execute operations.

Risk Buffering
The traffic surveillance system is an invention and as development is undergoing, new risks are uncovered. Buffering has helped to absorb some of the effects of these risks. Enhancements to the budget and schedule, addition of extra modules to ensure project success among others are some of the buffers I did. The design process was allocated more time to mitigate any productivity risks that may arise when the system has been deployed to a real environment.

Risk acceptance
I had to bear some of the risks, but before I could, I undertook proper analysis of the severity of the risk impact. Acceptance required having a backup plan and solution in place in case the risk matured. For example, being it that the system uses solar energy, a scenario could arise where the sun doesn't shine for a long period of time. The backup plan is to use a Witty Pi which has a small battery to store time.

9.3 Quality Management

The degree to which the surveillance system serves it's purpose is the measure of it's quality.

9.3.1 Main factors considered to conclude the level of the system's quality.

Accessibility.
The mobile platform user interface should be seamlessly operational. The image content should be readily available and updated frequently as required.

Accuracy.
The ability to eliminate faces and vehicle registration plates from each image captured requires extensive accuracy from the algorithm used. Privacy is very crucial to qualify the system as a high-quality product.

Adaptable.
The system should be able to adapt to changes in weather, changes in internet connection speeds during different times of the day.

Availability.
As a 24/7 surveillance system, availability is very important. It is required that the system provides end users with 3 minute updates of traffic in different areas from morning to evening without any lags or inconsistencies.

Other quality attributes for the surveillance system include durability, complexity, efficiency, design, documentation, consumable, fault tolerant among others.

9.3.2 Techniques

Flow charts.
Graphical representation of system flow steps and their relationships. This technique shows all the activities, decision points and the order of processing. It helped to identify some of the risks that may arise and thus supporting development of test procedures.

Benchmarking.
So many surveillance systems have been developed. Some rotate and tilt, some use WIFI, some use cellular data. All these kinds helped to set benchmarks for which the traffic surveillance system must meet.

Quality meetings.
Frequent meetings were made with project stakeholders to make collective decisions concerning the current and target quality of the surveillance system.

9.4 Social, Legal, Ethical and Professional Considerations

Professional Issues

Education and accreditation: Designers and developers working on the project must poses an IT related degree accredited by a recognized professional body. Individuals who are deemed incompetent or without technical knowledge, raise issues for the project during its development life cycle. Persons with any other unrecognized guarantors can’t be trusted to work on the traffic surveillance system.

Level of expertise: Any participant in this project must have the necessary skill set to operate the several tools and apply any advanced techniques to be used in the various stages of system development. The simulation tools, analysis tools such as UML, the design and modelling tools, the implementation skills among others.

Trust: as the individual working on this project, I must portray a certain level of trust since a lot of confidential information will be shared with me during interviews, during data collection and requirements analysis.

Codes of conduct

Take public consideration: The traffic surveillance system has a direct impact on the public to both help and harm socially, and economically. Therefore, the professional must pay a great deal of care to the target environment when designing the system.

Faithfulness: As the main person working on this project, I must in all ways be faithful to other stakeholders or client as well as any other people affiliated to the project. I must as well upraise and respect the software professional field.

Codes of Practice

Technical competence: Persons working on the traffic surveillance system must meet the level of competence required by the regulatory organ to develop, design or distribute any Software or software related products, while respecting and maintaining data integrity, privacy of the client, and client sensitive information, and ideas during the life time of the project.

Impartiality: I must show equal treatment of all rivals concerning any misunderstandings that arise during the time of development and management of the system. Two different supervisors suggested different methodologies, I had to eliminate bias.

Attention: The British Computer Society (BCS) code of practice strongly urges one to ensure to pay critical attention to the requirements of those working for you. The client's personal desires and requirements shall always assume first priority.

Management: All issues that arise in the system's management process will be handled internally. Processes involving client registration, system feature upgrade all carry the values of privacy, integrity and security.

Standards: Certain standards have been set so that i can compare with to govern my professional activities. Ranging from quality management, assurance, security of data in information systems, safety standards in hardware systems among others.

Ethical decision making

I must ensure that I do what is right with or without the knowledge of the end user or customer. During development, I ensured my motives were always to be positive. In the case of a huge change or privacy requirements, I had to either consult with the Project supervisor or do the right thing.

Computer Law

The computer law covers a large spectrum. So many things are liable to a traffic surveillance system computer professional such as;

- Contract laws with third parties, mode of payment for materials used, licence renewability.
- Intellectual property laws may be placed on the Traffic surveillance system designs, and the codes compiled.
- The ownership of the traffic surveillance system is clearly identified from the conception. Any copyrights should be put in place as well as agreements on specific attributes concerning the project and its resources or outcomes.
- Data protection laws may apply in the event that the system has been deployed.

Ergonomics

I experienced long sitting hours during the development cycle, long LED light exposure. I tried to surround myself with a conducive environment to do the set tasks.

Inclusivity

For blind end users, an option for voice recognition can be added in the surveillance system.

Privacy laws

The surveillance system captures images from public areas. These images at most times contain sensitive information. Pedestrian identities are exposed, as well as car registration plates, and other privacy attributes. Issues arose when the algorithms used to eliminate the parts in the image was proved imprecise especially because of the quality of the images.

10 Critical Appraisal

"Success and failure are difficult to define and measure, since they mean different things to different people" Thomas & Fernández, et al (2008)

CAN REPETITIVELY UPDATED IMAGES CAPTURED FROM AN AERIAL VIEW IN DIFFERENT ANGLES OF BUSY ROADS HELP REDUCE INCONVIENIENCE OF TRAFFIC CONGESTION?

The research question above has helped to identify the problem and the planned solution. The traffic surveillance system's purpose is best left to opinion of the end users. If by any chance it assists and improves on the road life style of Ugandans, then it can be deemed relevant. Personally, I experienced so many huddles from traffic jam and thus felt like developing this solution would be ideal with consideration to current or planned solutions.

The solution aims to help the following categories of people ;

- Regular automobile operators to carefully plan trips.
- Emergency vehicle operators to find the fastest and safest route.
- Police officers and other traffic authorities to monitor traffic behavior so as to know when to intervene.
- Construction companies carrying out renovations.
- Attentive road users to report accidents.

The positive outcomes of this work may include;

- Provision of new dynamic in surveillance sector. Using images reduces on the resources used compared to existing video related surveillance modes out there.
- Support for green energy utilization. Solar is preferred in this project to direct electricity for reasons aimed to support the betterment of our environment.
- Inclined personal experience in embedded systems. As a software developer, I sought out to learn embedded systems to widen my knowledge and skills.
- The project aims to address the criticality of traffic congestion and how it affects both developing and developed economies.
- This work can be used as an off the shelf resource to improve on. Future researchers can pick ideas to further supplement their own research.
- This research work contains new ideas and inventions which makes it of great value.
- User analysis of traffic behavior patterns at different hours of the day.

The negative outcomes;

- The solution is not helpful to individuals who have defects such as blindness.
- This work has been compiled in only English Language.
- The project cannot be effective in areas with poor internet connection.

11 Conclusions

11.1 Achievements

I have managed to learn a variety of programming languages I had no exposure to initially.
I have learnt how to use the terminal in Linux to execute commands and request for updates.
I have managed to learn how to power and assemble embedded hardware components.
During meetings with my supervisor, I learnt how to work with other people to make progress on projects.
I learnt how to do extensive research and cite references.

11.2 Future Work

The current product can be enhanced to provide traffic analytics, such as different traffic levels on a junction during different hours of the day.
The mobile application can be enhanced to use panorama view to present graphical imagery to the end user.

12 Student Reflections

Personal performance
I realised during project life cycle that I was scared to tackle most of the challenges I encountered during product development phase. Procrastination was a big hindrance to my performance and affected my project's deliverables.

Problems encountered
I experienced challenges when it came to assembling the modules into one unit. As an invention, I struggled to keep it as small and flexible as possible. Achieving a durable and stable end-product became a frustrating challenge for me.

Lack of similar projects made referencing and research a taboo for me. I had to brainstorm most of the ways of how to design the system.

I also struggled very much to keep the project cost at per. With each build, I kept realising more and more modules were required to achieve the product's desired functionalities. Consultations with domain experts was necessary to overcome this.

Bibliography and References

Steven J Johnston (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. Available at: http://www.mdpi.com/2079-9292/6/3/51/htm [Accessed 12th October 2017]

Steven F. Barrett. (2013). Arduino Microcontroller Processing for Everyone! 3rd Edition. Wyoming.

Prof. Hironori Washizaki. Agile and Extreme Software Development. Available at: https://link.springer.com/article/10.1186/1687-3963-2013-15 [Accessed 3rd July 2017]

Matti Kaisti (2013). EURASIP Journal on Embedded Systems.

Vipin Jain, Ashlesh Sharma et al. (2012). Road traffic congestion in the developing world. In Proceedings of the 2nd ACM Symposium on Computing for Development, Atlanta, GA.

Live Traffic NSW. Available at: https://www.livetraffic.com [Accessed 9th April 2017].

A. E. Aburawi, S. Salic, M. Krampell. (2016). New model of emergency communication for disaster relief in remote areas, Innovations in Information Technology (IIT) 12th International Conference on, pp. 1-5, Abuja.

Francois Caron, Stéphane Coulombe (2015). Video Error Correction Using Soft-Output and Hard-Output Maximum Likelihood Decoding Applied to an H.264 Baseline Profile, Circuits and Systems for Video Technology IEEE Transactions on, vol. 25, pp. 1161-1174. Montreal.

Wireless CCTV. Site Security. Available at: http://www.wcctv.co.uk [Accessed 11th April 2017].

INRIX. Press Release. Available at: http://inrix.com/press-releases/traffic-congestion-to-cost-the-uk-economy-more-than-300-billion-over-the-next-16-years/ [Accessed 11th April 2017]

Jonckie. (2011). Driving in congested traffic. Available at: http://www.insurancechat.co.za/2011-05/traffic-survey-reveals-frustrations-of-traffic-congestion-in-south-africa/ [Accessed on 11th April 2017]

Appendix A – Project Specification

Phillip Lubanga
Project Brief.docx

Appendix B – Interim Progress Report and Meeting Records

[Include here the interim progress report and supporting documentation submitted]
Progress report by 05th

PROJECT NAME	ROAD TRAFFIC SURVEILLANCE SYSTEM			
PROJECT MANAGER	LUBANGA PHILLIP			
PROJECT DESCRIPTION	A computerized solution for drivers to aid in reduction of road traffic on busy roads and major city junctions using camera technology to capture clear images of the various roads to be accessed by drivers.			
ID	**TECHNICAL ASSUMPTIONS**	**FUNCTIONAL REQUIREMENTS**	**STATUS**	
001	Users should have android powered devices	The system should be able to upload images onto a server.	Complete	
002	Users should have the application on their devices	The system should be able to capture images of roads at different angles up to 360 degrees after every 3 minutes.	Complete	
003	Users should have the application on their devices	The system should be able to save old images for viewing at a later stage.	Complete	
004	Full time internet connection.	The system should be able to update itself after a period of x minutes when a new image is taken.	Complete	
005	People aren’t so tech-savvy or educated	The system should enable end-users access images from the server on their mobile devices.	Complete	
006	Privacy is a must	The system must eliminate faces and vehicle registration plates from each image	Incomplete	

Meeting records
1st meeting with Main Supervisor (Leon smalov) on 18th September 2017.

Agenda: To clarify end-product design, operation and effectiveness.

Minutes

- Purchase casing material online. Material should be strong and resistant to mold, weather conditions of various kinds.
- Choose a case study
- Use a stepper motor to achieve 360-degree rotation instead of a servo motor.
- Start working on the project introduction.
- Use Github for version control.

2nd meeting with Main Supervisor (Leon Smalov) on 29th September 2017
Agenda: To assess current progress

Minutes

- Start on mobile application and develop sketches.
- Workout the end user story and experience.
- Figure out how to ensure for privacy during system operations.
- Research on existing traffic surveillance systems.

3rd meeting with Main supervisor (Leon Smalov) on 16th October 2017.
Agenda: To assess how to cut project costs.

Minutes

- Minimize on project complexity
- Interact with potential end users on current progress.
- Re-assess project requirements.
- Track project milestones and stick to schedule.

4th Meeting with Main supervisor (Leon Smalov) on 6th November 2017.
Agenda: To get criticism to derive motivation on project improvement.

Minutes

- Choose internet connection source.
- Produce new end-product uses.
- Start unit testing.
- Ensure to design a reliable power management sub system.

5th Meeting with Main supervisor (Leon Smalov) on 4th December 2017.
Agenda: Assess current system evolution and progress.

Minutes

- Develop easy and precise presentation.
- Define project methodology used.
- Refine product for presentation.
- Correct errors and faults found.

6th Meeting with Main supervisor (Leon Smalov) and Assistant supervisor (Bentaleb) on 8th December 2017.
Agenda: To present current project progress.

Minutes

- Assess project legal, and social issues identified.
- Identify in report factors that affected design.
- Produce a failure handling plan.
- Assess country regulations for which system is to be deployed.
- Prepare for deployment.

Appendix C – Requirements Specification Document

Capturing images.
Uploading images.
Processing images to remove faces and sensitive information.
Storing images.
Routing from one location to another
Show available cameras on selected route
Responding to user requests
Reliability
Efficiency
Maintainability.

Appendix D – User Manual

Hardware Manual

Press the switch on the Witty Pi to power on and off the hardware components.
Press the switch on the regulator to start and stop current flow.
Press the switch on the Power Bank to power the system.
Long press the switch on the Power Bank to put on the Always on Mode.

The camera starts rotation after 10 seconds of system start up and shutdowns after 5 minutes.

Mobile application manual.

Click the Trafficap icon on the applications list to start the application.
To route, enter your origin and destination address and click route
The available cameras on that route will show up
Select a camera to view the traffic around that area.

Appendix E – Project Presentation

Trafficap.pptx

Appendix F – Certificate of Ethics Approval

ethics-certificate.pdf

Appendix X – As required

Code

https://www.dropbox.com/s/uiveqwro8p47n6r/Trafficap-Code.rar?dl=0