

The Software development of “Smart Gate” for UTAR

By

Chan Siew Meng

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirements

For the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2015

REPORT STATUS DECLARATION FORM

Title: The Software Development of “smart gate” for UTAR

Academic Session: JANUARY 2015

I _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

278, Kampung Timah

31800 Tanjung Tualang

Perak

Supervisor's name

Date: _____

Date: _____

The Software development of “Smart Gate” for UTAR

By

Chan Siew Meng

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirements

For the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled “ The Software development of “Smart Gate” for UTAR ” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisor, Dr. Goh Hock Guan who gave me advice and guidance throughout the project whenever I met an obstacle.

Next I would like to thank my project group mate Navindra Raj with whom we discussed the design and the implementation of the whole system. He always gave some idea and suggestion whenever a problem was faced.

Last but not least, I would like to thank my friend and family, without their support, I wouldn't be able to complete this FYP.

Once again, thank you all very much.

ABSTRACT

This project stimulates how the RFID technology can be apply to solve the manual vehicle inspection system by using MICAz. This project focuses on software development that tailored for UTAR. The software will connect the MICAz to perform the identity checking and act as vehicle sensor to detect entrance of vehicle. It also link with webcam to produce live feed display on the software. The software will also display the identification result to the screen. The software to be used for development for the software of this project is Java Eclipse, NetBean, MySQL and TinyOS.

TABLE OF CONTENTS

TITLE	i
ABSTRACT	ii
TABLE OF CONTENT	iii
LIST OF FIGURES	v
LIST OF ABBREVIATIONS	vi

CHAPTER 1.0 INTRODUCTION	1
1.1 Problem Statement	1
1.1 Problem Statement and Motivation	2
1.2 Project Scope	3
1.3 Project Objectives	4
1.4 Impact, Significant and Contribution	5
1.5 Project Background	6

CHAPTER 2.0 LITERATURE REVIEW	7
2.1 Review of Technology	7
2.2 Review on Existing Solution	8
2.2.1 Comparison of Existing Solution	9
2.3 Concluding remark	10

CHAPTER 3 SYSTEM METHODOLOGY	11
3.1 System Development Model	11
3.2 System Requirement	15
3.2.1 Hardware	15
3.2.2 Software	16
3.3 Functional Requirement	17

3.4 Project Milestones	18
3.5 Estimated Cost	19
3.6 Concluding remark	20
CHAPTER 4.0 SYSTEM DESIGN	21
4.1 System Architecture	21
4.2 Functional Model in the System	23
4.3 System Flow	27
4.4 Database design	29
4.5 GUI design	30
4.6 Concluding Remark	30
CHAPTER 5.0 SYSTEM IMPLEMENTATION	31
5.1 Hardware Setup	31
5.2 Software Setup	32
5.3 Setting and Configuration	33
5.4 System Operation	33
CHAPTER 6.0 SYSTEM EVALUATION AND DISCUSSION	35
6.1 System Testing and Performance Metric	35
6.2 System Testing and Result	36
6.3 Project Challenge	36
6.4 SWOT	37
6.5 Objective Evaluation	38
6.6 Concluding remark	38

CHAPTER 7.0 CONCLUSION	39
7.1 Conclusion	39
7.2 Recommendation	40
REFERENCE	41
Appendix A	42

LIST OF FIGURES

Figure Number	Title	Pages
Figure 3.1.1	Waterfall model	11
Figure 3.1.2	Agile model	12
Figure 3.1.3	Spiral model	13
Figure 3.1.4	Prototype model	14
Figure 3.2.1.1	MICAz connect with 51-pin butterfly board	15
Figure 3.4.1	Gantt Chart of project milestones	18
Figure 4.1.1	Flow chart of overall system	21
Figure 4.1.2	Flow chart of overall system 2	22
Figure 4.2.1	flowchart for camera live feed	23
Figure 4.2.2	flowchart for video recording	24
Figure 4.2.3	flowchart for license plate recognition	25
Figure 4.2.4	flowchart for MICAz	26
Figure 4.5.1	UI design of the system	30
Figure 5.1.1	The Arduino connect with a servo	32
Figure 5.4.1	Screen capture of the system UI	34

LIST OF TABLES

Table Number	Title	Pages
Table 2.1.1	Comparison if existing system	9
Table 3.5.1	Estimated Cost for project development	19
Table 3.5.2	Estimated Cost for project commercialize	19
Table 4.4.1	Table 1 of the database	30
Table 4.4.2	Table 2 of the database	30
Table 5.1.1	Table for the used MICAz pin	31

LIST OF ABBREVIATIONS

RFID	Radio Frequency Identification
IDE	Integrated Development Environment
SDLC	Software Development Life Cycle
GUI	Graphical User Interface
USB	Universal Serial Bus
UTAR	University Tunku Abdul Rahman
OS	Operating System
OpenCV	Open Computer Vision
WAMP	Web application Apache MySQL PHP
ANPR	Automatic Number Plate Recognition

CHAPTER 1.0 INTRODUCTION

1.1 Problem Statement

Every morning around 8:00am, there will be vehicle congestion happen in front of UTAR gate. This situation is endangering the students as the car is congested, student is not easy to cross the road and exit the student hostel also very difficult. The road is not only use by car, it use by also motorist and cyclist. As the vehicle is congested they will be more difficult to move and need to be caution during their ride as many car or bus will pass through them. Although there is cyclist path available but it only fits 2 bicycle at a time and during morning there is plenty of student using the path so some of the cyclist will using the road beside the cyclist path. This will increase the congestion of the road. Eventually due to road congestion, student and staff of UTAR will be late for class and accident will happen more frequently.

Recently there is another issue arise where robber able to enter campus area and rob some of the student inside campus area. The security issue of the UTAR area need to be concern. Although the guard at the gate try their best to inspect every vehicle that passing through but there is too many car passing through each day. Sometime they may miss one or two car during a day. Furthermore they unable perform inspection during bad weather. Hence, these provide chances for those who with bad intention able to enter campus premises such as robber try to rob student or rapist hide inside toilet waiting for victim.

Often there will be at least three guard perform the vehicle and student id inspection at UTAR gate. One will check the cyclist, another will check for car and the one will check motorist.

1.2 Motivation and Problem Statement

Motivation of project comes from electronic toll collection. The electronic toll collection had already been implementing in various place and country. Apartment entry and lift floor also using passive RFID. It can be use for deploy in solving the vehicle congestion problem happening in UTAR especially in the morning. Not only the vehicle congestion problem, the vehicle that is invalid to enter the campus able to enter during raining weather. This shows there exists security problem.

This project is able to solve the problem mentioned above. These problems need to be solve because of the vehicle congestion will affect students and staff. It will cause the punctuality problem as both student and staff will late for class and thus affect the teaching or study progression, as student might miss out something and for lecturer will need to extend time to cover the syllabus. Another reason is for security purpose, it will affect the student if there is a rapist able to enter the campus and make sexual harassment towards the student.

1.3 Project scope

This project will focus on software development. It total consists two parts, the configuration to simulate the behaviour of RFID. Due to active RFID is expensive therefore shift the project to stimulate the situation using MICAz. The other parts is the user interface, it will be develop in java language for its object oriented advantages.

For the MICAz, few is using for transmit the id, two will use for weight sensor. It will display the result at the computer that connected with base station for receiving data. For the camera, the camera is use for record the traffic that passes through the gate for whole day. After that, it will query the time in the database for those invalid entrance and convert it into images.

This project will program the MICAz to perform the function as stated above. The MICAz for id will only perform the id transmission only. For the two weight sensor, one will be use for enter and exit of the detection area so that we can determine whether the vehicle have the tag or don't.

For the user interface, we will provide the live feed from webcam and display it out. This can ease the guard that he doesn't need to stand outside but still get a view on the road condition. Besides it also contain the feature of recording, the other camera will record the vehicle that doesn't have tag and save it into a folder. This can be use for the university staff to take further action and is also serve as a prove for illegal entrance. Furthermore, this software also consists of vehicle plate recognition feature, this feature is use to prevent the tag to be use on other car. The software will compare the plate number store in the database and compare it with the current vehicle. If the plate shown is not same then, it will trigger the video recording and display it out on the user interface.

1.4 Project objectives

Aims to improve the time require passing through gate. Current practice of vehicle inspection is very time consuming. It require manually check and compare with the list provided on the vehicle one by one. This is very important as the project want to solve the vehicle congestion problem.

Aims to reduce staff require to perform inspection. By implementing the RFID it will take less time and require less man power for the system because the system will do the inspection job. The guard will only need to handle some case like parents enter campus to inquiry academic structure.

Aims to eliminate illegal entrance. The RFID is not easy to duplicate, without the tag the vehicle will mark as invalid entrance. Also during nasty weather, the staff unable to perform vehicle inspection, therefore those unauthorized people will gain the chance to enter. With the data query from the database, it can be easily determine whether the tag is belong to the vehicle or not. Current practice that using the sticker can be easily duplicate and the car license plate number can be easily erase due to it is written not printed on the sticker.

Generally this project try to develop a piece of software to have a better car inspection system. A report will generate each day to summarize the traffic of that particular day. It is very hard for staff to record down how many vehicle pass today and how many of it are invalid entry.

This project focus on develop a piece of software that will work with RFID and camera. A java base GUI will develop display validity of vehicle entrance in real time. The recorded video by camera will capture image from video according to the time store in database and display out in the GUI.

This project focuses on software development for the system. The hardware part is handling by my group mate.

1.5 Impact, significance and contribution

This project will benefit UTAR student and staff or any private area which need a smart gate software. The reason for solving this problem is that time is very valuable to everyone. By solving the vehicle congestion problem, it will take less time for people to reach the destination earlier. The second reason is the security problem, sometimes the guard is not there due bad weather or they are changing shift, then those who are potentially dangerous may gain entrance for example rapist. Furthermore the system operates at lower cost, due to fewer staff needed to maintain the system. This system will also operate under bad weather for example, staff unable to inspect vehicle during raining but this system will be able to do it.

The solution proposed is able to solve these existing manual inspection problems. It requires less staff to be on the location and provide more accuracy and efficiency vehicle inspection.

1.6 Background Information

Radio Frequency identification RFID basically is using radio frequency to transmit id. It function just as the barcode that appear in shops. The RFID consists of two part, the reader and the tag. The tag consists of id and the reader “read” the id in the tag. The different from it compare to barcode is it has ability to “read” at longer distance, didn’t require line of sight, more secure and read multiple tag in short time. It had been use for highway toll collection in certain country.

In this project, a software is develop for identify the passing vehicle is valid to enter or not. The result will display on monitor and store into database. A simple report will generate for summary of total vehicle pass through, how many are valid entry and how many don’t. This software will store the time of each vehicle entry or exit in database. Those invalid entry will be display on report.

For this project, MICAz is use instead of RFID. It is because the RFID development kits is expensive, and MICAz provide similar function that able to stimulate the project. Furthermore MICAz is available at campus and have experience on handle it before, therefore MICAz is chosen for this project.

CHAPTER 2.0 LITERATURE REVIEW

2.1 Review of technology

OpenCV – open source library for computer vision. Originally written in C++, now supporting multiple platforms such as Python, Java and MATLAB. This library mainly focus on real-time image processing. This library is use for grab image from webcam and display out in this project.

Some of the feature provided by OpenCV are imagedata manipulation, image and video I/O, basic image processing, structural analysis, camera analysis, basic GUI and object recognition.

OpenIMAJ – open source library for multimedia content analysis. Develop pure in Java language, support multiple OS. There are total of five module in this software, image analysis and processing, audio analysis and processing, video analysis and processing, web analysis and processing, machine learning k-nearest neighbor and clustering and hardware connection like Kinect and GPS receiver. The call of the function is easy and responsive. The documentation on how to use the software is available download on their website.

LTI-CIVIL – A Java library for grab image from video source such as USB camera and webcam. Support multiple OS. Doesn't have much documentation online.

Sarxos webcam capture API – develop in Java, design for common use in webcam. Support multiple capture framework and multiple OS. Depend on maven which are available online, provide basic sample code such as live view from web camera.

2.1.1 Summary of the Technologies Review

In this project, OpenCV is choose to be use to integrate with camera. This is because the community for this software is large, many other had been use this library so it is choose to be use in the project. Another reason is, we got try out the Sarxos webcam sample code, the actual start of the live view and video recording takes too long, for our project we need the video record to be trigger as fast as possible. By using and tested with OpenCV it give us a

satisfy time consuming. Aside from that, the code for the video recording and live view is also simple and understandable. Thus OpenCV is use in our project.

Comparing the ease of use, the OpenCV and OpenIMAJ these two are more or less similar. Both function calling is simple, basically 2 to 4 lines of coding is already capable to display out the feed from camera. Adding a few more lines then the configuration will be more specific, such as frame rates and resolution.

2.2 Review on existing solution

Marina's car park and installations access control – Design to have a reliable access control of private and public car park management. This whole system consists of three things sliding door, camera and barrier.(RFID access control n.d.)

The entrance is design with this flow, first validate the proximity member card then the sliding door open. After the sliding door open the camera will read the registration number, the barrier will only open after checking the registration number is correct in the system. For the public entry, the car registration number will be print on ticket when they going through the gate.

Advantages of this system is the security aspect of the system is high. It only allow the vehicle enter after it check through the car registration number. At the same time it reveal it disadvantages which is time consuming, the vehicle need to wait for the camera to recognize the number and check it through the system. Furthermore, the checking is start after checking in the proximity cards. Since it is not done simultaneously, the time consume will be much longer.

Tracking and access control for vehicles by active RFID – Design to have the access control on a fuel distribution for trucks. The access for the fuel distribution center is design to operate 24 hour and 7 days a week so there will be no restriction on when the trucks need to fuel up the truck, they can access anytime. As there will be no worker at night, this system will provide those who authorize to enter the center and fuel up the truck.(Tracking and access control n.d.)

This system will get the data from RFID and then it will send the data to a server. The server will store the data and at the same time the system start to work. The system is include of detect presence of vehicles, control the enter and leaving using barriers, store the history and the detail of the vehicle passes through and the server is remote control by a software that manage all the parameter by internet.

Access control device for car parks Ktransit – This system main benefit is allow to permit different access at specific hours and days. In a car park, it consists of many kinds of user such as owned space, shared place and company place. It is difficult to configure each user individually. The other advantages for this system will be login and permission assignment are easy and it is centralize using internet. It also provide a software to manage the configuration processing and login.(Access control device n.d.)

2.2.1 Comparison of existing solution

Existing System	Advantages	Disadvantages	Critical Comment
Marina's car park	Using RFID for access control Read the registration number before letting in.	Time consuming as need to wait for the registration number to be check on the system	It have the function of read the registration plate but the time consuming for the vehicle to enter takes too long
Tracking and access control for trucks	Store the history of vehicle passing through. Supervise the floor using internet.	Doesn't validate the vehicle license plate.	This system unable to detect for the trucks that borrow tag from other trucks
Ktransit car park	Allow certain tag to be pass through only in certain time	Doesn't validate the vehicle license plate.	This system not prevent the tag been borrow out to other.

Table 2.1.1 Comparison if existing system

2.3 Concluding remark

There are few technology review to be done as we doesn't learn it before. Among the few reviewed technology, OpenCV had been selected to be used in our project. OpenCV is selected is because it have a large amount of resource found on the internet. The ease of use also one of the reason we select it. OpenCV is a powerful image processing library, although it first release on C++ but later on they support other programming language as well. Their release on support Java makes our programming easy because our development is more prefer to Java language.

Compare the existing solution, we can conclude that our project is more suitable for the situation in UTAR. Our project provide, quick entrance for vehicle which the three reviewed project also cannot achieve because they control the access with a barrier. Another function that we provide is the live feed of the camera so the user able to monitor the situation without step out of the guard house. Another camera function will be video recording of those who illegal entrance, it will be serve as prove when confront with the illegal passer.

Therefore, this project is cost-saving by reduce the man power needed for the sticker inspection. This project also solve the vehicle congestion problem that occur in certain time. The illegal entrance will also be solve as student won't able to duplicate or transfer the tag easily.

CHAPTER 3.0 SYSTEM METHODOLOGY

3.1 System Development Models

This project will use the prototyping methodology to do software development. It is a System Development Life Cycle (SDLC) that the system is continually revised until a presentable system is implemented. It selected for our project because the nature of our project will require continuous configuration, development and implementation to get the best outcome.

There exists other SDLC such as Waterfall Model, Agile Model and Spiral Model. **Waterfall model** require task to be completely done then proceed to next task, there is no return to previous task from current task. It is a sequential design process, it consists total of 5 phases, which are, requirement analysis, design, implementation, verification and maintenance.

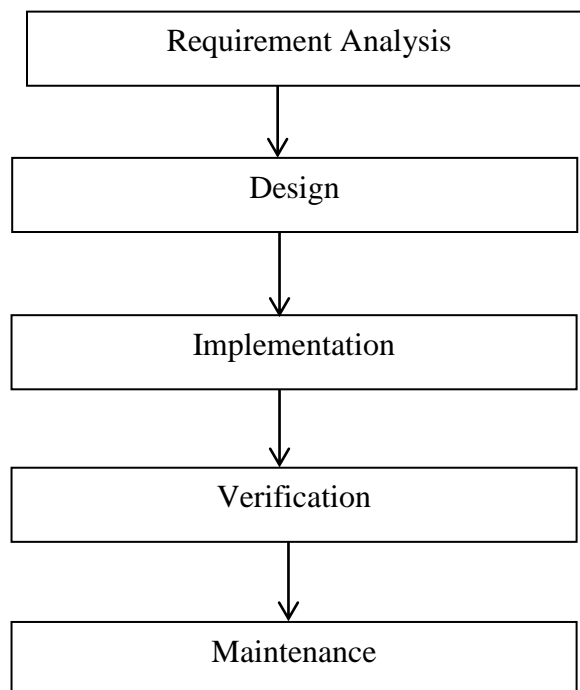


Figure 3.1.1 Waterfall model

For our project, we need to repeat deploy, testing and configure according to the situation. Therefore Waterfall Model cannot be our project SDLC since we can't return to implementation once go to verification.

Next is **Agile Model**, this model more focus to user requirement. The whole process will continue improving according to user. Working software is continuous show to customer and get feedback from them to get the best possible product that satisfy customer. Since this model is based on user, if the user is not sure on what product they want, the development is difficult. This model will have 5 phases basically and will continue iterate until user satisfy the outcome.

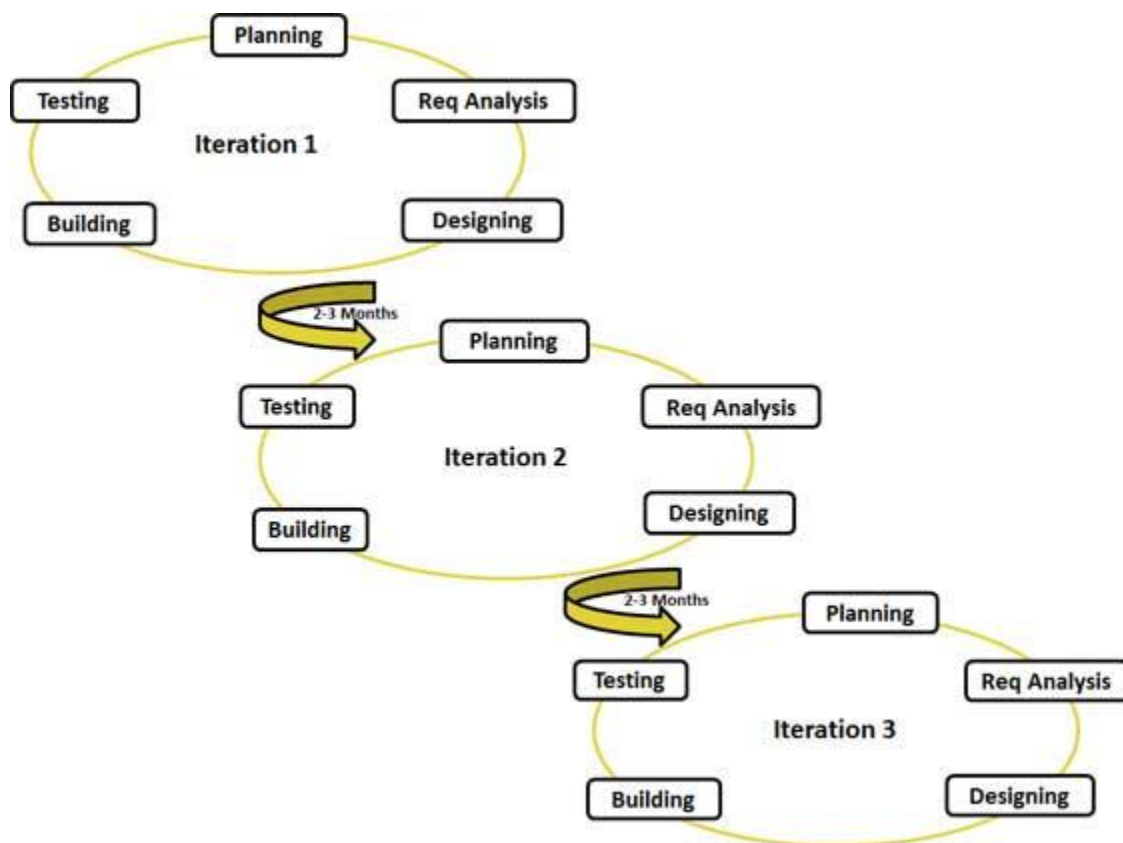


Figure 3.1.2 Agile model

(Photo taken from http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm)

Since our project doesn't tailor the software to the user but more design for the environment and our project have less interaction with user, hence, this SDLC also not suitable for our project.

Third is the **Spiral Model**, this model focusing on the risk that will face by the project. It has 4 phases, planning, risk analysis, engineering and evaluation. This model is similar to Agile Model as it also will iterate repeatedly until user get what they want. There is a phase that other model doesn't have which is risk analysis. This phase is to identify possible risk or problem then solve the risk. Any risk found will be needed to have solution and implement it on the project.

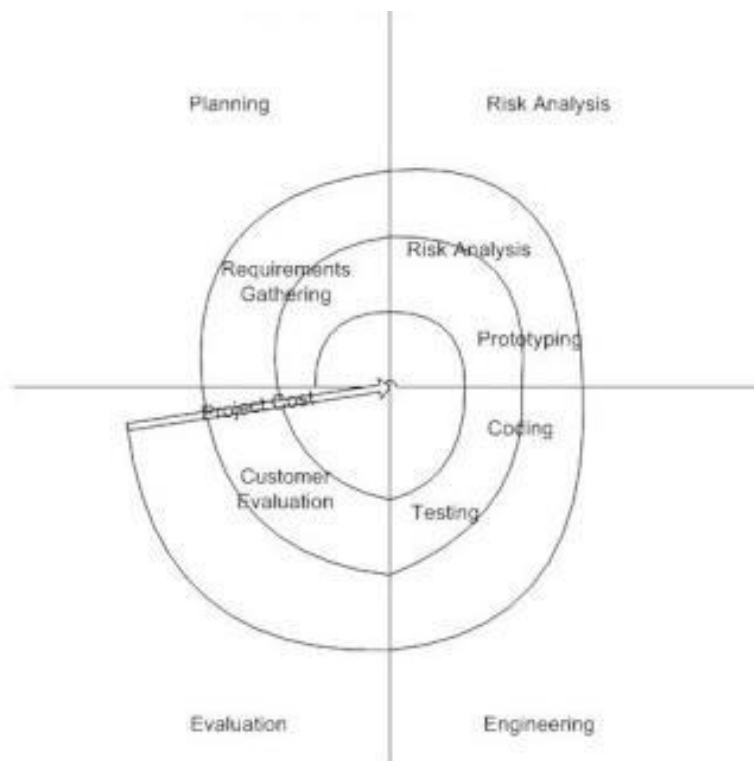


Figure 3.1.3 Spiral model

(Photo taken from <http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/>)

This model is design for high risk project, it have large amount of risk analysis. But this model require experienced risk analysis and doesn't suitable for small project. For our

project, the risk is minimum, cost and risk evaluation is not important and we don't have risk analysis expertise. Thus this model will not fit our project.

Last is the **Prototype Model**, this model emphasize on get a view and test on the product base on known requirement. It will bring the user to have a better understand on the system. Advantages for this model will be error will be found sooner and missing functionality can be identified. This model are easily become more complex compare to original idea.

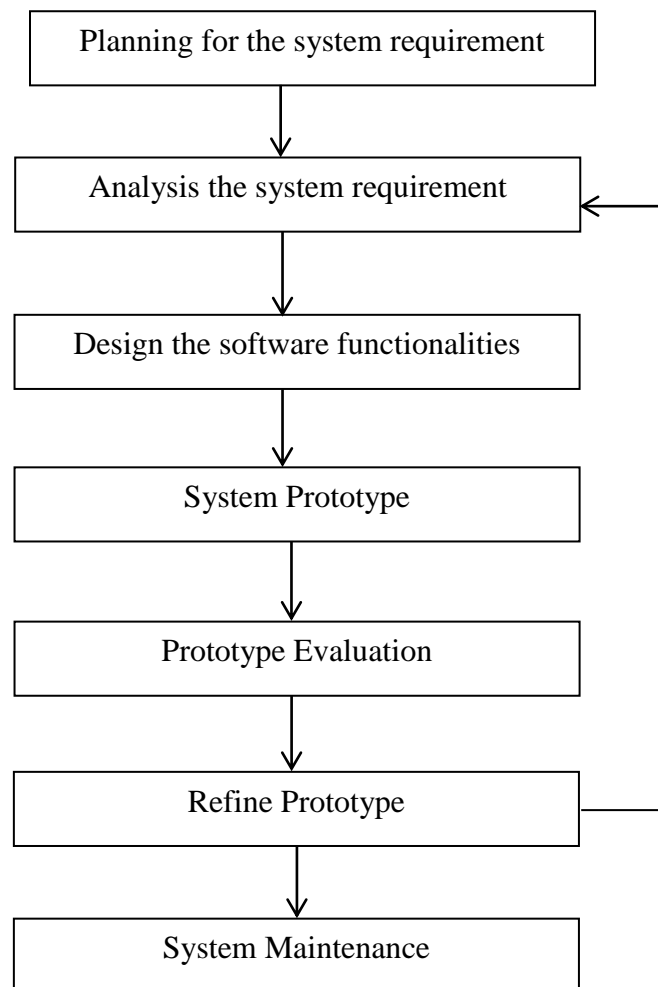


Figure 3.1.4 Prototype model

Since our project had multiple functionality and each of them is linked together, it is crucial to make sure each and every functionality is working before merge the whole system together. A prototype of each functionality need to be made and a final prototype when all the component are linked together to ensure the whole system perform what it should be.

3.2 System requirement

This project will be using OpenCV to capture image from webcam and produce live view. For the image processing part, it will be using JavaANPR. This JavaANPR consists of image processing such as extracting, resizing, template matching, brightness comparison and more. The JavaANPR also have the Optical character recognition algorithm which are neural network and k-Nearest Neighbors algorithm.

3.2.1 Hardware

MICAz – a microcontroller which using ZigBee protocol for communication. Mainly use in wireless sensor network. Power supply by battery make it able to work independently. Connected with a 51-pin breakout board for simulation of weight sensor.

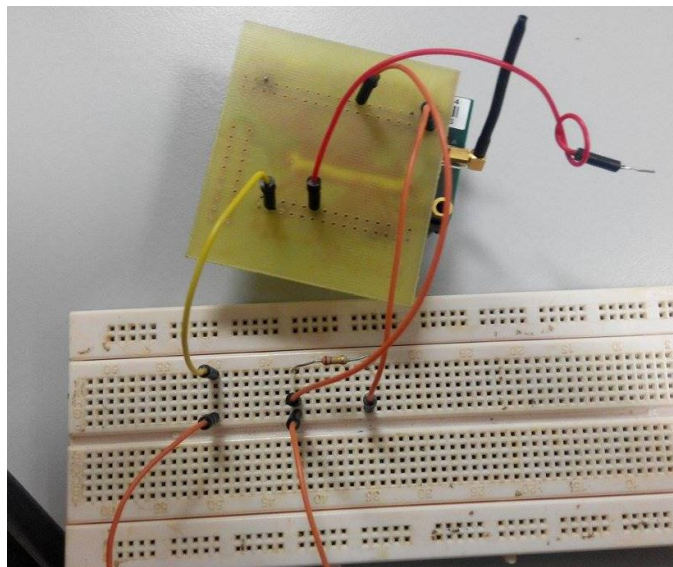


Figure 3.2.1.1 MICAz connect with 51-pin butterfly board

Programming board – a USB based programming interface for MICAz. It upload the program form computer to MICAz through USB and programming board. Another function for this programming board is act as a base station to receive packet from MICAz with a MICAz attach to it. This programming board is power by USB.

Personal Computer – A computer that run on Window OS and with a virtual Machine that run with TinyOS. All the camera and the programming board will connect to the computer

through USB. TinyOS use to for the MICAz data receiving and windows use to display the UI and camera feed, recording and database.

3.2.2 Software

JavaJDK – a java software development kit, which enable the computer to run on Java Application develop in our system. This package is for developer, it can perform developing, debugging and monitoring the Java application.

TinyOS – a Operating System that require for the programming of MICAz. It also used to display the packet receive by the base station. The packet analysis will also to be do in this OS using Java program.

OpenCV – This is a opens source image processing library. To be use in the webcam live feed and video recording. Resource used by OpenCV is lesser.

WAMP server – It is a server contain MySQL database to store the records of the vehicle pass through the gate. The table can be create and edit through PHP. It is easy to use and deploy. This software is also a freeware.

JavaANPR – a automatic number plate recognition software develop in Java language. Originally develop to recognize Russia vehicle plate. Modified and implement to suit Malaysia car plate.

3.3 Functional requirement

Grab image from webcam

In order to produce live feed and video recording, grabbing the streaming of images produce by webcam is essential. The images grabbed will need to be further manipulation to let it display on the screen. For the video recording, it will undergo some function to make the images to become a video file and to be store it in the computer. Without the images that grab from webcam, we are unable to produce the live feed and record the video.

Wireless Sensor Network (MICAz)

This is the configuration for the MICAz and nodes to communicate wirelessly using ZigBee. The nodes will gather data from the weight sensor, then proceed to send the packet back to base station where it connected to the PC through USB and display the data on screen. Without the network, the generated data is unable to send to the computer for further analyzing.

Image processing (vehicle license plate recognition)

This step is to analyze the image to get where is the location of the license plate from a image of a vehicle. This processing consists of compare the brightness, hue, saturation to roughly estimate the location of plate. After that it will perform the template matching using the Euclidean distance matching algorithm to determine the plate. Without this model the Optical Character Recognition is unable to run with the whole image of vehicle.

Artificial Intelligence (character recognition)

This is a continuous from the image processing, it is require to perform the Character Recognition to get the character of the license plate. It will first learn the character and number, and then it will compare with the image and come to conclusion that what is the number from the image. In this project there are two learning algorithm, one is Neural Network another is k-Nearest Neighbor Algorithm.

User interface

This is to create a user interface where the relevant info will be show on screen. In this UI it will consists of live feed from webcam, detail of the tag pass through, history of passed tag, a frame will pop out when the recording is running and vehicle plate number. A login page will be develop to prevent other people gain the info of the tag detail.

Database

Database need to be created to store the valid tag id along with the student and vehicle info which own the tag. Later on when the tag passes through the gate, the student and vehicle info will display the UI. It is also function as a checking of tag which random tag id will be flag as invalid enter as the program will check for the existence of tag in the database. Without the database, the valid tag along with the detail of student and vehicle will require to be hardcode in the system and the history of tag passed the gate will not be possible to display.

3.4 Project Milestone

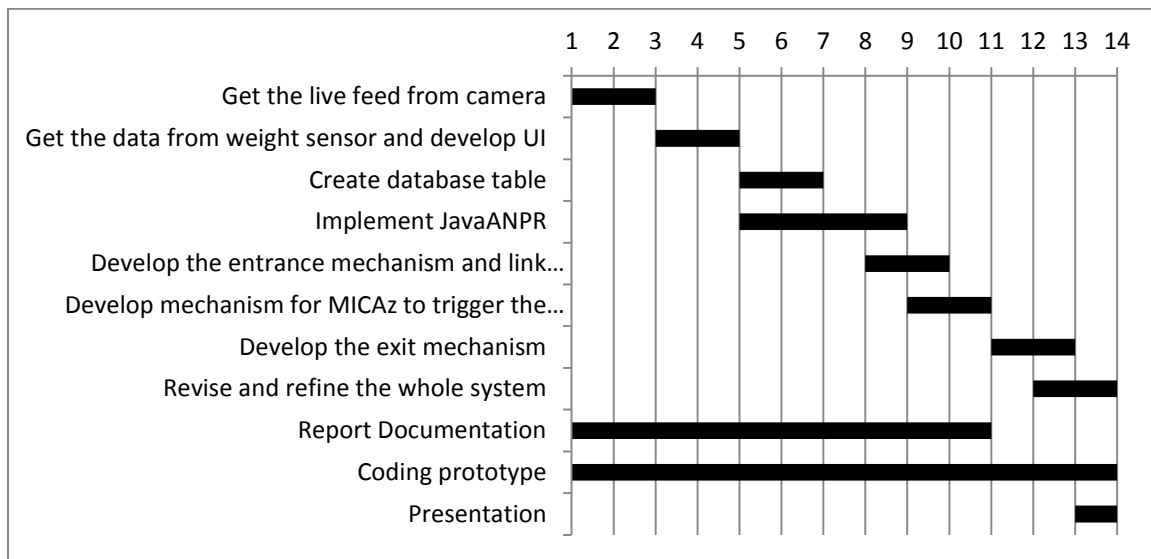


Figure 3.4.1 Gantt Chart of project milestones

3.5 Estimated Cost

In general, the cost for my part of project will require no cost to develop the project. The equipment for this project is done by my group mate.

Software	Price
Java IDE(eclipse/netbeans)	Freeware
TinyOS	Freeware
OpenCV	Opensource library
WAMP server	Freeware
Total Cost	RM 0.00

Table 3.5.1 Estimated Cost for project development

Next is the cost estimation for commercialization done by my group mate on the hardware of this project. The hardware for our project usage is supplied by UTAR and some of it is self-owned so there will be total RM0.00 on the hardware part also

Item	Quantity	Price per Unit (RM)	Total Price (RM)
Active RFID Tag	3000	50.00	150,000.00
Active RFID Reader	2	5000.00	10,000.00
Batteries	3000	5.00	15,000.00
Webcam	3	200.00	600.00
Above-ground weight sensor	4	2500.00	10,000.00
Boom-gate	1	2800.00	2800.00
Personal Computer (PC)	1	1500.00	1500.00
WampServer, MySQL,	1	Freeware / Open Source /	-

TinyOS, Java JDK 1.7, Oracle VM VirtualBox, OpenCV		BSD-License	
Total:			189,900.00

Table 3.5.2 Estimated Cost for project commercialize

*The figure of 3000 batteries and RFID tag is get from the current number allocation for stickers.

3.6 Concluding remark

The SDLC selected by this project is the Prototype Model, it enable us to have a clearer view on the progress of the project and solve the problem if it been identified. The functional requirement will provide us a to-do list and keep the project on track. The project milestones serve as a schedule for the project to know which component should be done within the time frame. The estimated cost will provide us a rough calculation on how much we can earn of the project been commercialize.

CHAPTER 4.0 SYSTEM DESIGN

4.1 System architecture

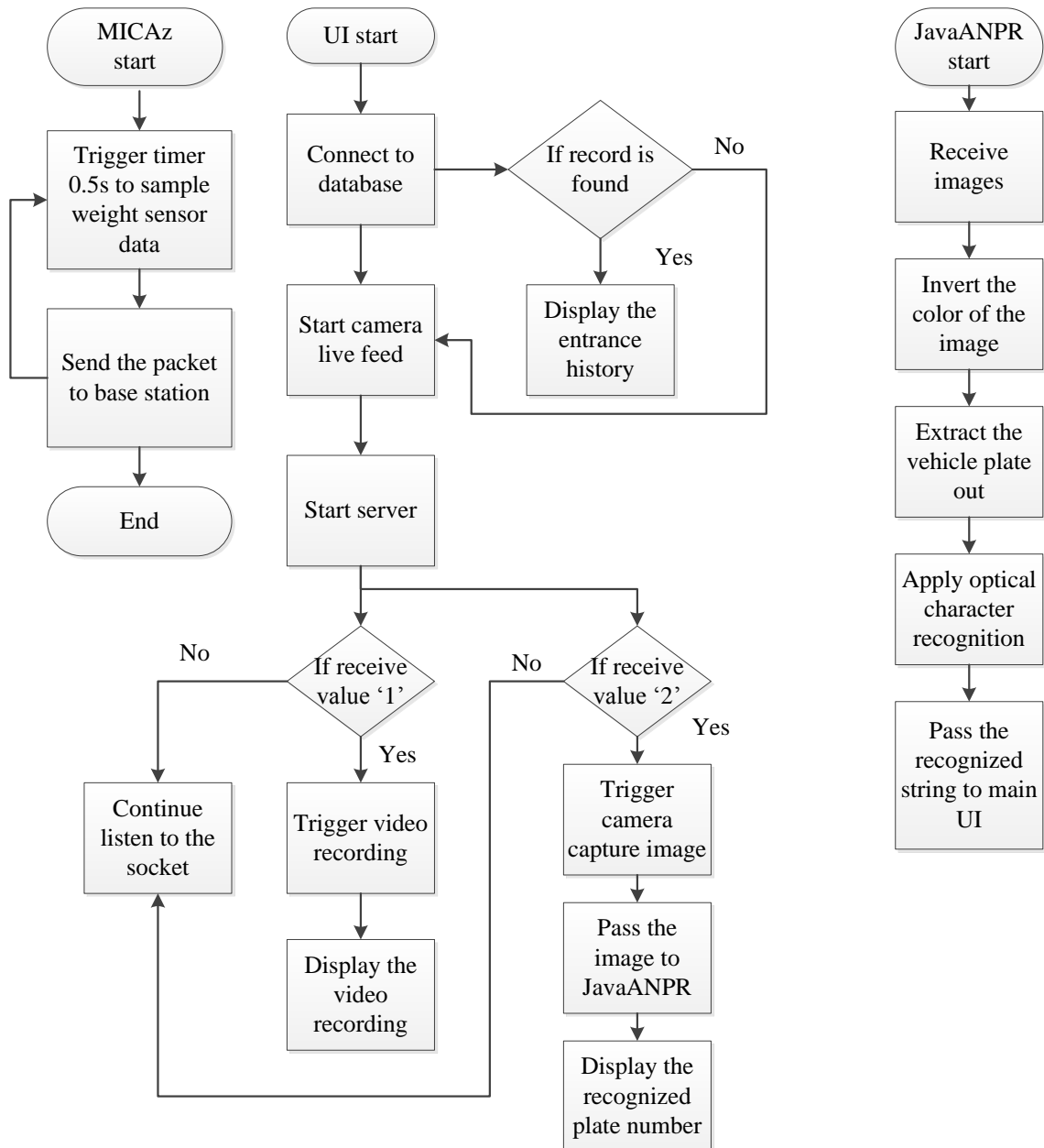


Figure 4.1.1 Flow chart of overall system

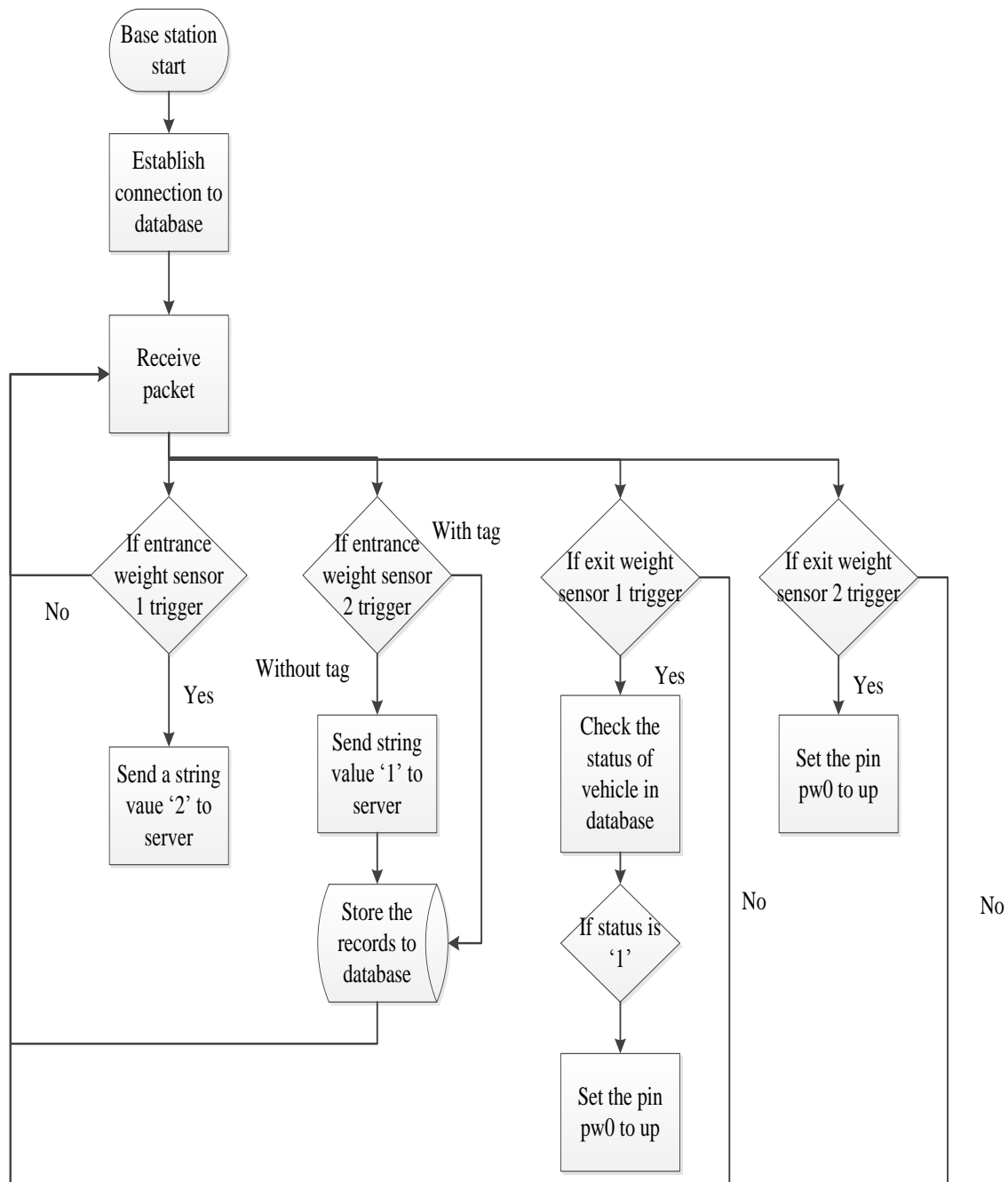


Figure 4.1.2 Flow chart of overall system 2

The two above figures show the flow chart of the whole system. The communication between the components is done using simple TCP connection. A server is establish in the main UI and other component act as a client and send the string value to server. Server will call the respective function according to string received.

4.2 Functional model in the system

Camera live feed

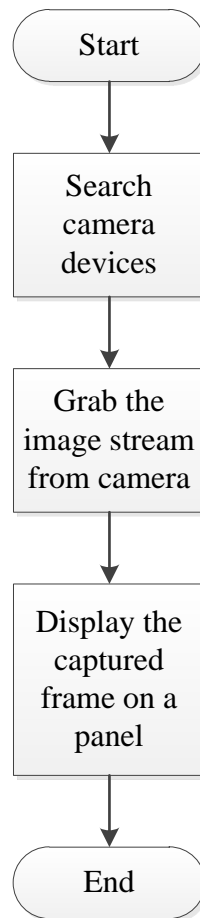


Figure 4.2.1 flowchart for camera live feed

To produce the camera live feed, the program will first search the available camera device. If none of the device found, the program will terminate. After the camera is found, then it will proceed to check on the resolution of the camera, allocate buffer accordingly and start to grab the images from camera. Then it would create a frame which later on will put together in the UI and display the grabbed image. Hence the live feed for camera is complete. This program will be call after display out the main UI.

Video recording

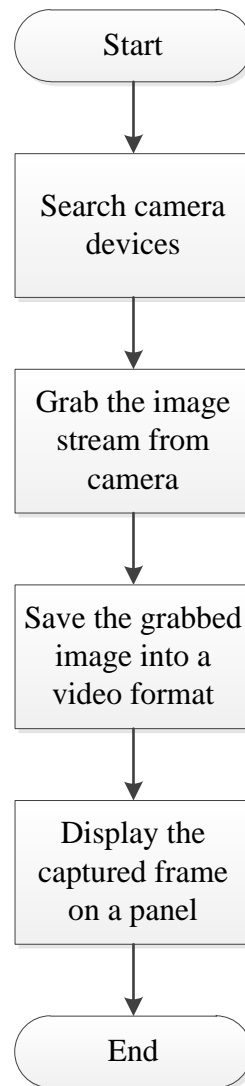


Figure 4.2.2 flowchart for video recording

Basically it is same with the camera live feed, it will search for the available camera, due to the live feed already take one of the camera, therefore we need another camera to perform the recording. The difference from the live feed is, after grab the image from camera, it save the images into a video file format. After it finish convert the format, it will proceed to save it at the location specify by the program. During the recording, it will also show the frame on the UI so that user is aware of the video recording is running.

Vehicle license plate recognition (JavaANPR)

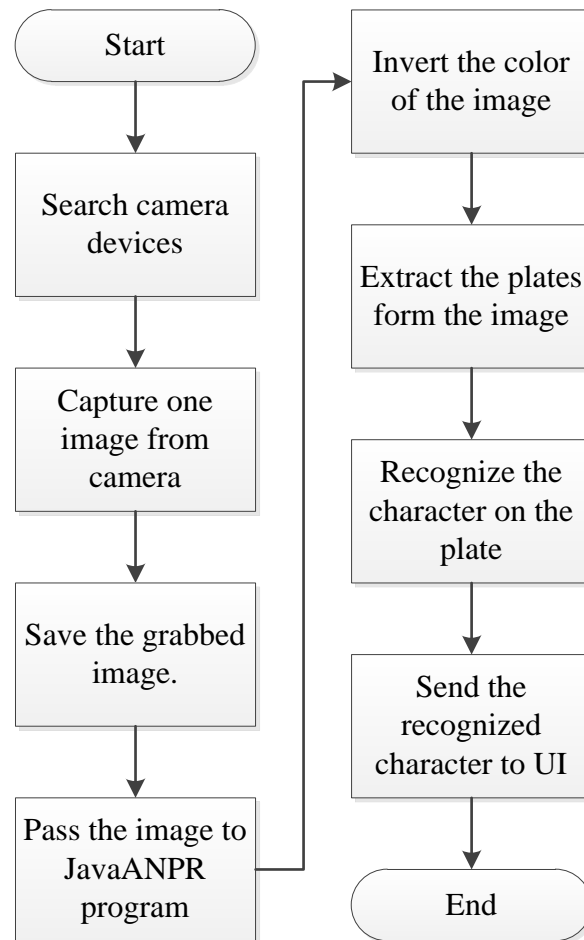


Figure 4.2.3 flowchart for license plate recognition

The program starts with search for camera, then it capture a picture using another camera. Afterwards, it save the image and pass the image to the core program of JavaANPR. The program will first invert the color of the image because originally it design to recognize the plate with white background and black fonts, so we need to invert the color of the image as color of Malaysia plate is inverse with it. Next the program will start to extract out the plate using Euclidean Distance Matching algorithm. After it successfully get the plate it will undergo k-Nearest Neighbor which is a algorithm for artificial intelligence to do the character recognition. Finally it will store the recognize character into string and send to main UI for display purpose.

MICAz

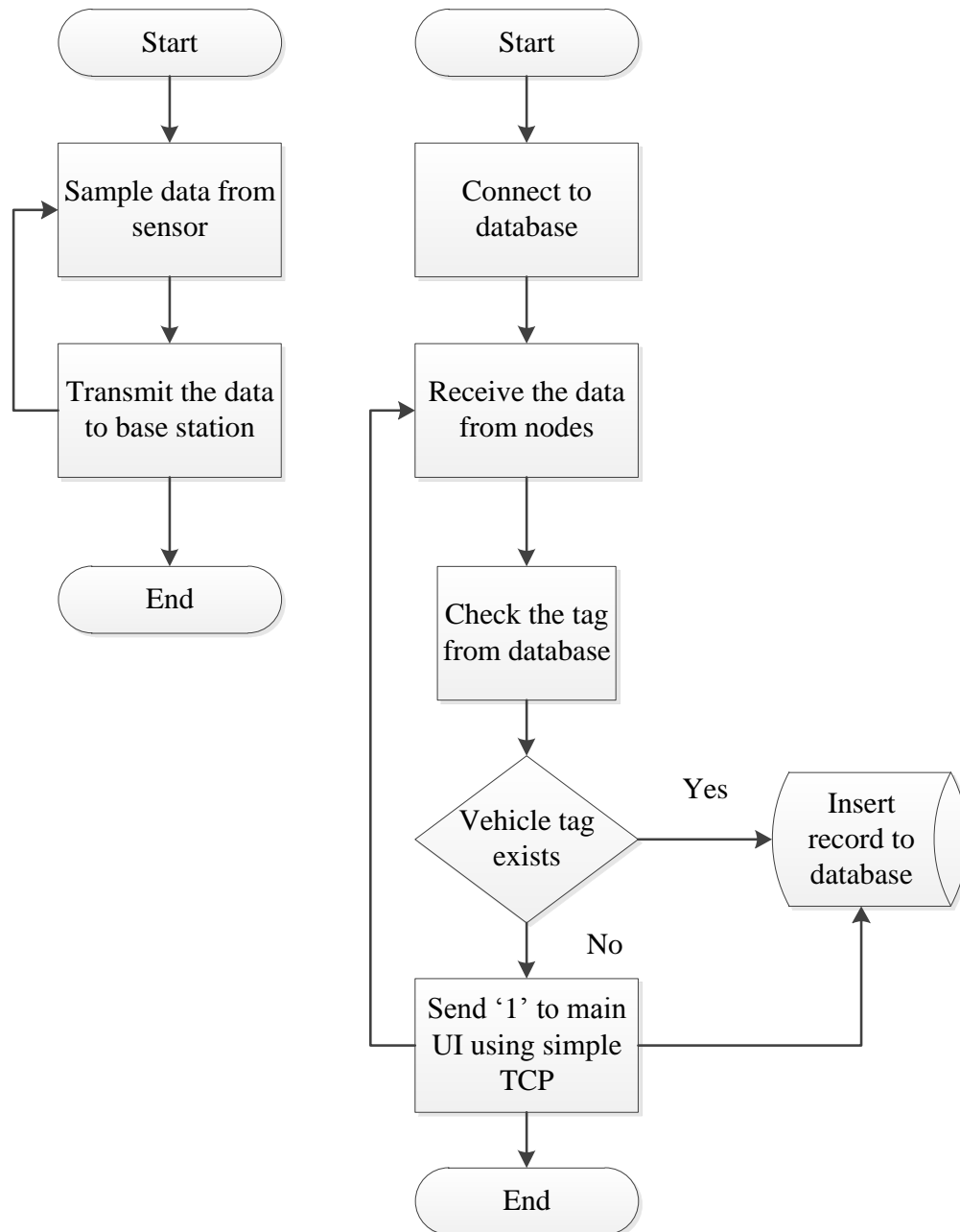


Figure 4.2.4 flowchart for MICAz

The diagram above show two MICAz left one is use for weight sensor data sampling and transmission. Right one is the base station, which receive the packet generate by nodes. At the same time, this base station also act as a reader for the RFID tag. After it receive the packet, it will check the id of the nodes, it is use to differentiate the nodes use for weight

sensor and the nodes use for tag. If it receive the tag from weight sensor node, it will trigger event accordingly, for the vehicle which have tag, it will put the record into database. For the vehicle that don't have tag, it will send '1' to main UI to trigger the video recording and insert the records to database.

4.3 System flow

In this project, we are using 4 sensors, 3 cameras, 1 base station (tag reader), multiple nodes (tag) and a pc running on windows with Java1.6 or later version with virtual machine running on TinyOS.

First, we acquire the data from sensors, the environment to in here is using TinyOS. For this project we do this by using a simple "weight sensor". This weight sensor is actually a simple circuit which give off one when it two wires touch together. To get the sensor data, this project is using MICAz to sample the data from the wires. The MICAz will sample the data every 0.5 seconds. In each 0.5 seconds, it will send the data back to the computer using ZigBee protocol. At the receiver side, our program will further using the data to make it meaningful. For every two data with the value of '1' will be flag as car enter the tag reading area. This is because car has front wheel and back wheel, thus it will trigger the sensor twice. Once the exiting sensor 1 trigger, it will check on the status of vehicle in database, if the status is '1' then it will pull the pin PW0 to up to open the boom-gate.

At the receiver side, we use Java PingPong to receive the data generated by MICAz. This is the program to determine the status of vehicle. There are four possible situations, first is the vehicle enter with a tag, second is the vehicle enter without tag, third is vehicle enter using other people tag and last one is the vehicle enter with a tag that doesn't exists with database. To determine does it have the tag, it will query the database, if the result return is 1, then the tag id is in database. For those vehicles that don't have tag, we determine it by after the vehicle pass through the second sensor. The tag which doesn't exist in the database will return a '0'.

For each vehicle entrance, whether it is valid or invalid, the data will be recorded and store into database. For the vehicles which have tag, the tag number along with the detail such as date time, car model color and etc. The id for the vehicle don't have tag will be '999'. These

Bachelor of Information Technology (HONS) Communication And Networking
Faculty of Information and Communication Technology

records will be show on the UI as information for the user to know how which vehicle had been passing through. Since the nature of the MICAz is generate the packet every 0.5 seconds it need some mechanism to control the data that get into database, else it will be have tons of redundant data. In this program we uses Boolean to control it. There will be also one action to be trigger for the vehicle without tag and with invalid tag. This Java PingPong will send a string with number '1' to the windows, using simple TCP connection from VirtualMachine. This will follow to trigger the camera recording program to record the vehicle entrance.

The main Java program that run in window will consists of, UI, camera live view, database records and vehicle plate recognition. To produce camera live view, we using the opencv which is a programming library design dealing with machine vision. In this program, first it will find the camera device, after a camera device is found it will proceed to grab the frame produce by the camera and display it out on part of UI. The same concept is applied on the video recording program, the only difference is we grab the frame and save the video file on a location in harddisk. To save the video we using the function from ffmpeg call ffmpeg frame recorder. This function enables us to decide on the output format, frame rates and recording duration.

For vehicle license plate recognition, this project is using JavaANPR an open source number plate recognition software. Since the software originally designs to recognize the Russian car plate, we do some modification in it. The plate in Russia is background white color and the fonts are in black color. This is reverse of color in plate in Malaysia, so we invert the image color before proceeding to recognize it. To start recognize it, it will start with finding 3 bands from the image. The bands is the where the image having the brightness level which are lower will select as bands. These bands are crops of the original image horizontally and it is the place where the plate is most likely at.

After getting the band, it will further try to find the location of the plate from the bands get from earlier. Same as above it will try to find 3 plates in each band. In here it is using the Edge detection to extract the possible plate out. After it gets the possible plate, the image will go through Hough Transformation which is a function that performs the shape

matching. For vehicle license plate, it will normally have rectangle shape. So the Hough Transformation will perform the rectangle shape template matching. Afterwards the plate will be normalize. Next it will identify the width height ratio of the plate, if the width height ratio of the identify plate is exceed maximum or below minimum of pre-define number, it will identify as not a plate then proceed with identify another plate.

For the plates that match the defined width height ratio, it will go through brightness detection for further segment out the character from the plate. Same as above, the character getting from the plate must match the minimum and maximum character acquire else it unable to proceed. For those possible character get from the plate, it will undergo a few more comparison such as brightness, contrast, hue and saturation to identify eliminate out the space on the plate.

Finally, the image will undergo character recognition algorithm. For this software, there are two available options to choose. One is Euclidean distance pattern matching and the other one is neural network. By default this program is running on Euclidean distance matching algorithm. Some final touch up to the software, it has the syntax analysis for recognized plate. This is done by input the arrangement of the plate structure in a text file. For example in Malaysia, normally the expected plate will be 3 characters follow by 4 numbers. So the program will try to correct the plate using this pattern.

4.4 Database Design

There are total 2 database table created for this project. One is used to store the valid tags along with student info, it consists of Tag ID, Owner Name, Vehicle model, Vehicle number plate, Owner contact info and status. The status is use for indicate the vehicle had been pass through the gate, 1 is pass through the entrance and 0 is exited the gate. Another table is used to store the date time of which tag pass through. In this table it only consist 2 columns, Tag ID and timestamp. These two table is later to be joint query and display on the UI.

Field	Data type
Tag ID	Integer
Owner name	Varchar
Vehicle model	Varchar
Vehicle number plate	Varchar
Contact info	Integer
Status	Integer

Table 4.4.1 Table 1 of the database

Field	Data type
Tag ID	Integer
Timestamp	Datetime

Table 4.4.2 Table 2 of the database

4.5 GUI design

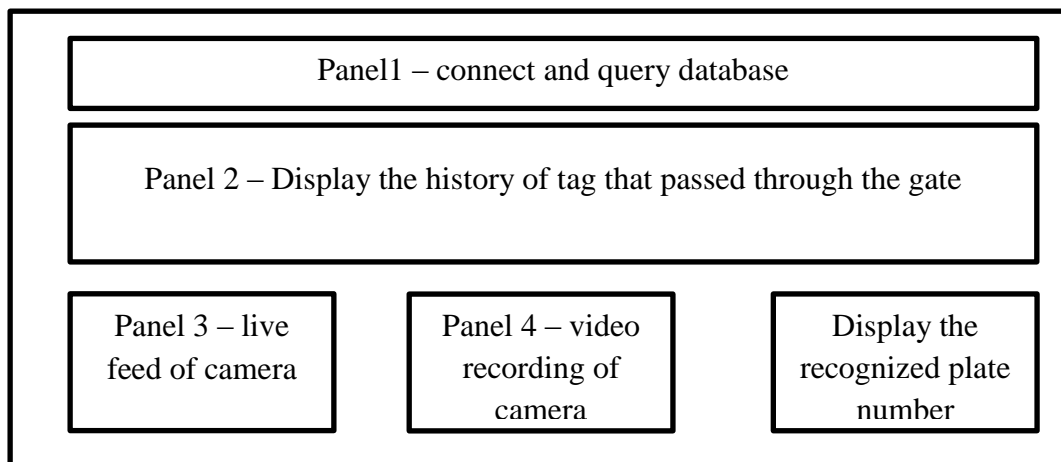


Figure 4.5.1 UI design of the system

4.6 Concluding remark

By looking at the whole system flowchart, it will give an overview on the whole system flow and how the components are link together. To have a further detail on how each components work will explain in functional model and system flow. In summary all the components are linked back to the main UI.

CHAPTER 5.0 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

MICAz

The MICAz is connected to a 51-pin Butterfly Board with total of 4 wires solder to act as weight sensors. From the 51-pin only few pin is used for this project. These pin are as follow.

Pin name	Function
GND	Connect to ground
VCC	Provide power for weight sensor
PW0	Send the signal to Arduino
ADC7	First weight sensor for exit
ADC8	Second weight sensor for exit
ADC5	First weight sensor for entrance
ADC6	Second weight sensor for entrance

Table 5.1.1 Table for the used MICAz pin

Arduino

A servo is use to simulate the boom-gate. It have 3 pin, one is use for ground, another is use to power up the servo and last one is to send the signal to the servo. This servo is connected to Arduino a micro-controller to send the signal to the servo and control the behavior of the servo. Below show the image of Arduino connect with the servo. The upper part with a stick is the servo and down there right hand side is the Arduino. The left hand side is a button to use for manual-override the boom-gate to open or close.

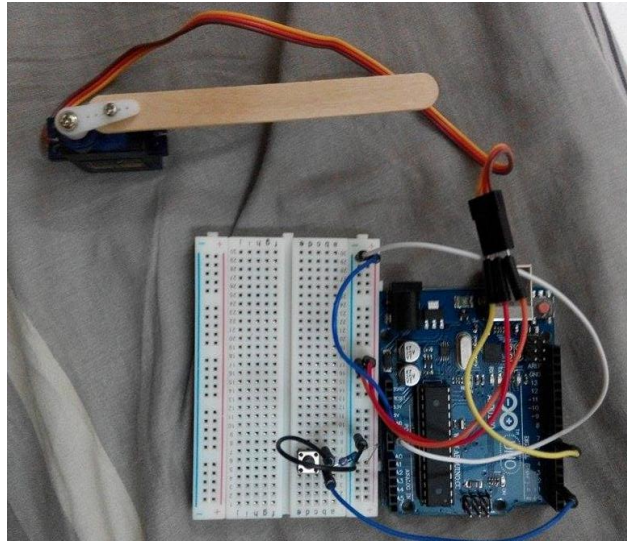


Figure 5.1.1 The Arduino connect with a servo

5.2 Software setup

TinyOS installation

To install the TinyOS, we need to install a virtual machine first, because we need the TinyOS to be run in virtual machine. The TinyOS is needed to program the MICAz and use as a interface for the base station to receive the packet. Further analysis is done on the TinyOS.

To install the TinyOS we need to copy the folder from UTAR lab because the OS in the lab is already install in the virtual machine format and the OS is also installed the essential program such Java and Cygwin in the OS. Therefore there is no need for user to further install, just copy the folder and open it with Virtual Machine, then it will work.

OpenCV installation

Download the OpenCV version 2.4.9 or later version from their official website. Install the application by double click on it. Select the place for the OpenCV to be install.

Eclipse installation

Download the latest version Eclipse IDE from their official website. Make sure the HDD fulfill the minimum requirement for installation. Proceed to install.

Java JDK installation

Go to oracle website, download Java SE development kit 7 or later version. Install the application, this application also included with the Java runtime environment which enable the computer to run on java application.

WAMP server installation

To install Web application, Apache, MySQL, PHP server, go to wamp server official website, download the latest version and install it.

5.3 Setting and configuration

To enable OpenCV to be usable in Eclipse we need to set up the path for the Eclipse to recognize OpenCV. Steps are as follow:

1. Go to Windows, then find Prefrence.
2. Expand Java -> build path -> User libraries
3. Click New
4. Type the new library name, for this case type OpenCV, click OK.
5. Click Add External JARs.
6. Browse for the location where you install OpenCV.
7. Expands the jar file, select Native library location, clicl Edit.
8. Click External Folder, browse and select x84 if computer is 32bit, x64 for 64bit.
9. Click OK.

5.4 System Operation

Below show the screenshots of the main UI that will be show to user. This UI consists of few part, most upper part is to key in the database connection, follow by it show the history to tag that pass through the gate along with the info. At the bottom part got two video, left hand side is the live feed from camera 1 and the left hand side is the video recording screen, it will only pop out during video recording. This image is capture by my groupmate as he have the multiple webcam.

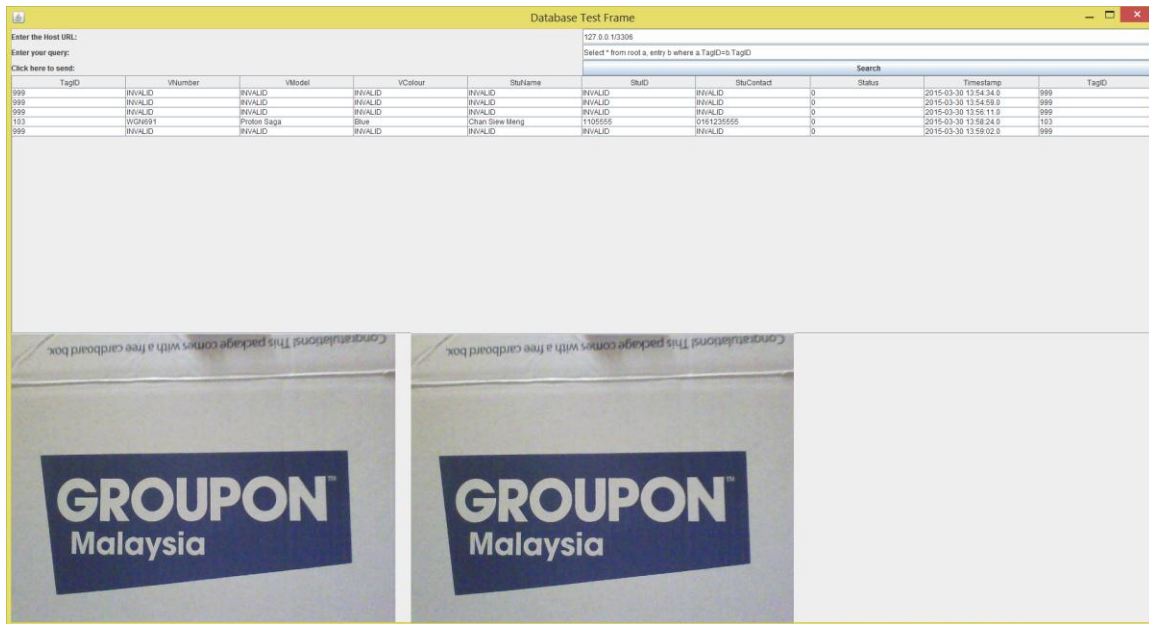


Figure 5.4.1 Screen capture of the system UI

5.5 Concluding Remark

For hardware we will be setup a few things, MICAz, 51-pin expansion Butterfly board, Arduino and servo. All this hardware need to be program properly to perform nicely. For the software setup, we require few thing OpenCV, WAMP server, Eclipse, TinyOS and Java JDK. All these software can be found and download form internet. Installation step also very straight forward. The tricky part will be configure eclipse to use OpenCV library. It require to go through some configuration before using the OpenCV. Lastly is the screen capture of our system main UI.

CHAPTER 6.0 SYSTEM EVALUATION AND DISCUSSION

6.1 System testing and performance metrics

The system testing is done separately for each component before merge together. First we will test on the camera live feed. We will try to run it to ensure the program is working. Once the camera is produce live feed, we start to run the video recording program. This is to test whether the program is able it to run on different camera at the same time. To determine the video recording is successful or not, we will check on the video produce by the program. To check the video, we need to determine is the video file in correct format or not, next we check whether the video playable or not. If all of above can be performed, the camera component is considered success.

Next component will be MICAz, we test it by looking at the message received at base station. To test on the weight sensor, we will touch the two pins that soldered with wire together. As a result we will obtain a reading of 1024, if we cannot get that reading, this might be cause by pin malfunction or weak soldering.

After that, we will be test with three possible situations, vehicle enters with tag, vehicle enters without tag, and vehicle enters with invalid tag. At this stage, database hasn't implement into the code yet, the testing is using hardcode. For first situation, the sending value '1' shouldn't be trigger, second and third situation should send '1' to server.

For the server, we will run it and display out to see does it receive the value '1' when the MICAz detect invalid entrance. After confirmation of receive value '1' the video recording will be implement into the system. We will test it again with the invalid entrance and check on the video recording.

The database will be used in two places, UI and MICAz. For the UI it will be in a joint query of two table and display at one of the panel. In MICAz we need to query the tag ID from database to check is the tag ID is valid. Next we need to insert the history of passed through vehicle into database. To test on does the record been insert into database, we use the command of database to show the data and test it with different number of tag, some tag ID is in database, some is not exists and see how it perform.

In order to test the whole system, we need to create a few situation to ensure the system perform as design. First situation is vehicle enter with a valid tag, second is vehicle enter with a invalid tag and third is vehicle enter without tag. For the vehicle enter with a valid tag, database must be able to query out the data, video recording should not be triggering and the date time along with tag ID need to be insert to the database. The second situation, the system need to be determine that the tag ID is not valid and trigger the video recording, the tag ID and data time will be insert to the database as well. For the last situation, the video recording should be trigger and date time for the entrance should be insert, since there is no tag ID it will be number 999 for the vehicle who don't have tag. All three situation mention above the history of the passed vehicle also need to be show on the UI. Live feed need to be show un-interrupted during the testing and video recording need to produce a playable video file.

6.2 System testing and result

We test using different tag ID to check on the MICAz program, some tag ID exists in database, some don't. The result getting is the MICAz program able to sort the tag ID correctly. If it is testing with two tag open at the same time, the program won't run correctly.

Another test will be on the vehicle recognition, different images with different angle supply to the program and check on the final output. Result is the vehicle with less skew will be more accurate.

6.3 Project challenges

The development of mechanism that control events to be trigger in MICAz. This is a crucial step to link the MICAz with the camera. The events need to be trigger correctly according to the situation. Filter need to be done to ensure no duplicate data insert into database.

The implementation of vehicle license plate recognition. This components consists of a lot program to make it work. It is hard to raise the accuracy of the recognize character as there are many part that are affecting the final outcome. Aside from the software part, the hardware itself will affect the recognition, different angle of captured image will produce different result same goes to the resolution of the image.

Combining all the hardware and software together to get the final result. This is challenging because the components are developed in a different platform, we need to choose 1 main platform and link all of the components together through a certain method. In this project, we use a simple TCP connection to send a message.

The reliability of the tag is also one of the challenges, as the packet transmission direction is unpredictable, the reaching time of the packet is also unable to determine. There might be a situation that the tag packet located at behind come before the current tag.

6.4 SWOT

Strength

- Fast and accurate vehicle access control.
- Providing live feed and video recording
- Require minimum human intervention
- Provide high security

Weakness

- If two tag distance is short will create problem
- License plate recognition is not accurate
- System is complex to deploy
- Require configuration to new environment

Opportunity

- A system that make use of RFID technology
- Security is high
- Rising the need of gated community
- Trend changing from barcode to RFID

Threat

- Complex to deploy
- Transmission of packet may loss due to external factor
- People tends to stick on old system which they comfortable at

6.5 Objectives evaluation

The staffs require to do the vehicle inspection had been reduce to minimum 1 person and they don't need to stand at outside of the guard house to perform the inspection.

The illegal entrance had been reduce since the system able to detect it automatically by checking does the vehicle got tag or not. Furthermore we have a mechanism for one tag only enter once, the tag that is enter will store into database, therefore unless the tag pass through the exit gate, else it is unable to enter again. It is more difficult to duplicate the tag compare with the use of sticker for vehicle inspection. It is impossible to change the vehicle plate number as it is store in the database, previously it just wrote on the sticker and can be erase off easily then write another plate number to it.

The time consuming for the vehicle to pass through the gate had been significantly reduce as the system perform inspection is lot faster than the manual inspection.

6.6 Concluding remark

This section contains the info on how we test on the components to know that the program is working. We observe how the system reacts to the situation and make modification if the system is not working as expected. This section also includes the evaluation of objective to see if it meets our defined objective. Beside that it contains also the SWOT analysis of the project.

CHAPTER 7.0 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

The problem created by the manual vehicle inspection system is solved by deploy this project. There are few problem created by manual inspection, the inspection is time consuming, sticker can be tampered easily and it require many staff to perform the inspection. This project objectives is aim to solve this problem, the implementation of RFID will solve the problem stated above. The final system of this system will be fast and accurate of vehicle inspection, increase the automation of the system so less human resource require and strengthen the security of the system by detecting the illegal entrance.

This project evaluate the technologies that use to develop the system. Comparing them and use the technology that suits this project. Beside it also review the existing solution that are in the market and compare the functionality between them and our project. For the SDLC Prototype was choose among three other reviewed model. The functional requirement is determined and the estimation cost is done.

The overview on how the system is expect to be run and how each component to be run is already list in this report. The flow chart of the whole system is drawn. The design of GUI is also draw in the report.

The setup for both hardware and software setup is explained in the chapter 5. System testing had been carried out and document in the chapter 6. SWOT analysis towards the project is done and evaluation of met objective also been documented.

7.2 Recommendation

This project now can be enhance by develop to show the number of parking slot available on each area and centralize it in a database. A mobile application can be develop to show the number of parking slot available in area. This will ease up the driver as they aware of which area have parking slots so they no need to drive around to find the slots which waste energy, petrol and time.

The next enhancement will be develop a component that enable user to insert, delete and edit the database records in the system. It will be design to be ease of use, because the current WAMP server had too many functions which can cause confusing sometimes.

Develop a mechanism that able to determine which are the tag own by the current vehicle. Now our system is unable to determine it as there will be a chance of the tag belong to vehicle at behind will be receive first.

BIBLIOGRAPHY

Gady Agam, 2006 *Introduction to programming with OpenCV*. Available from :<
[http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-
intro.html#SECTION00021000000000000000](http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html#SECTION00021000000000000000)> [8th April 2015]

Kimaldi, *RFID access control to a marina's car park*. Available from :
http://www.kimaldi.com/kimaldi_eng/solutions/access_control/rfid_access_control_to_a_marina_s_car_park> [8thApril 2015]

Kimaldi, *Tracking and access control for vehicles by Active RFID* Available from :
http://www.kimaldi.com/kimaldi_eng/solutions/active_rfid_technology/tracking_and_access_control_for_vehicles_by_active_rfid [8th April 2015]

Tutorials Point, *SDLC – Agile Model*. Available from :<
http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm> [8thApril 2015]

ISTQB Exam Certification, *What is Prototype model- advantages, disadvantages and when to use it?*

. Available from :< <http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/#wrap>> [8th April 2015]

APPENDICES

Appendix A

3.5 Two-Axis Magnetometer (MTS310 Only)

The magnetometer circuit is a silicon sensor that has a unique bridge resistor coated in a highly sensitive NiFe coating. This NiFe coating causes the bridge resistance of the circuit to change. The bridge is highly sensitive and can measure the Earth's field and other small magnetic fields. A useful application is vehicle detection. Successful test have detected disturbances from automobiles at a radius of 15 feet. The sensor is the Honeywell HMC1002 sensor. A detailed specification sheet is found at <http://www.ssec.honeywell.com>. The output of each axis (X, Y) is amplified by an instrumentation amplifier U6, U7. The amplified output is available at ADC5 and ADC6. Power is controlled to the magnetometers by setting signal PW5. Each instrumentation amplifier (U6, U7) can be tuned using the digital potentiometer PT1 that is controlled via the I2C bus.

⚠ **WARNING:** The NiFe core of the magnetic sensor is extremely sensitive. However, it is also subject to saturation. Saturation occurs when the sensor is exposed to a large magnetic field. Unfortunately the MTS310 circuit does not have an automatic saturation recovery circuit (set/reset). This limitation prevents the magnetometer from being useful in applications

requiring DC response (for example compassing). There are four pads label S/R (Set/Reset) available on the PCB for adding an external set/reset circuit.

3.6 Turning Sensors On and Off

All of the sensors have a power control circuit. The default condition for the sensor is off. This design helps minimize power draw by the sensor board.

In order to turn sensors on, control signals are issued to the power switches. Table 3-3 below lists the control settings.

Table 3-3. Control Settings for the Sounder and Sensors

Sensor/Actuator	Control Signal
Sounder	PW2
Microphone	PW3
Accelerometer	PW4
Magnetometer	PW5
Temperature (RT2)	INT2/PW0 [†]
Photocell (R2)	INT1
Temperature(RT2)(MTS300CB/MTS310CB)	PW0

⚠ **NOTE:** Only one of the INT1 and INT2/PW0 signals should be activated at a time. See Section 3.3.

Figure A.1 : Datasheet for the magnet sensor board.

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S3, Y3	Study week no.:	2
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock Guan		
Project Title:	The Software Development of “Smart Gate” for UTAR		

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Discuss more detail with group mate about overall system implementation, such as adding and eliminate some minor element from project.
Further discuss the design and draft out on implementation.
Able to produce live feed from webcam using “maven”.

2. WORK TO BE DONE

Need to produce live feed from webcam.

Start to have further implementation on micaz

3. PROBLEMS ENCOUNTERED

Unable to deploy the live feed from webcam at group mate computer.

Unable to borrow out the fyp equipment, hence having slower progress as need to travel to campus for equipment.

4. SELF EVALUATION OF THE PROGRESS

Progress is slower as equipment unable to take out from campus.

Need to schedule time to stay at campus for fyp equipment.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S1, Y3	Study week no.:	4
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock Guan		
Project Title:	The Software Development of “Smart Gate” for UTAR		

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Decide to change the sensor type from magnetic sensor to simple weight sensor.
Done some minor change to previous code.
Start to look at some Vehicle license plate reorganization information.

2. WORK TO BE DONE

Change the MICAz code to perform the weight sensor.

3. PROBLEMS ENCOUNTERED

Value getting from pin has some noises, solve it by changing to ADC pin which it segment the voltage into different level.

4. SELF EVALUATION OF THE PROGRESS

Although at first changing the sensor type cannot get the desire result, but after try it on other pin, we overcome the problem.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S1, Y3	Study week no.:	6
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock	Guan	
Project Title:	The Software for UTAR	Development of “Smart Gate”	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Study the javaanpr software, try to localize it to detect Malaysia car plate.

2. WORK TO BE DONE

Make the javaanpr software to be recognize Malaysia car plate

3. PROBLEMS ENCOUNTERED

Too less information found online.

Unable to understand the source code.

Documentation of the javaanpr doesn't explain much on coding.

4. SELF EVALUATION OF THE PROGRESS

Progress is slow due to Chinese New Year holiday.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S1, Y3	Study week no.:	8
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock Guan		
Project Title:	The Software Development of “Smart Gate” for UTAR		

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Continue study on the javaanpr software as not much open source vehicle license plate recognition software available at the same time searching for other similar source code.

Success to trigger the recording with MICAz using simple client-server connection.

2. WORK TO BE DONE

Javaanpr can detect Malaysia car plate.

3. PROBLEMS ENCOUNTERED

Found one alternative to javaanpr but the installation process is too troublesome will discuss this issue teammates.

4. SELF EVALUATION OF THE PROGRESS

Progress is late on schedule

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S1, Y3	Study week no.:	10
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock Guan		
Project Title:	The software Development of “Smart Gate” for UTAR		

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Continue study the javaanpr, able to change invert the color of the image.
Connect the MICAz with the database.
Develop the mechanism for MICAz to insert only relevant records to database.

2. WORK TO BE DONE

Javaanpr should be integrate in the system

3. PROBLEMS ENCOUNTERED

Report formatting is wrong. Need to re-type the most of the report.

4. SELF EVALUATION OF THE PROGRESS

Progress is late need to speed up the progress more.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year:	S1, Y3	Study week no.:	12
Student Name & ID:	Chan Siew Meng - 1002259		
Supervisor:	Mr. Goh Hock Guan		
Project Title:	The Software Development of “Smart Gate” for UTAR		

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Further study on the Javaanpr to ensure the work flow of the code to be soon include it with the system.

Demo on incomplete system to supervisor.

Re-do some of the part in report.

2. WORK TO BE DONE

Final touch up for the system, prototype need to be generated, system need to be include with all the component.

Report need to be finalize

Need to develop the exiting algorithm

3. PROBLEMS ENCOUNTERED

The progress is extremely late on schedule. Many part need to be done.

4. SELF EVALUATION OF THE PROGRESS

Report need to be done asap as the submission date is near.

System touch up will also need to be done after report submission.

Supervisor's signature

Student's signature