

# Daily activity application linked with public transportation system

Zeyad Fathi Sfaxi, Mahmood Sarzil, Karishma Patel A/P  
Balvantry

SESSION 2016

FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY  
MAY 2016

# Daily activity application linked with public transportation system

BY

Zeiyad Fathi Sfaxsi, Mahmood Sarzil, Karishma Patel A/P  
Balvantry

SESSION 2016

THIS PROJECT REPORT IS PREPARED FOR  
FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY  
IN PARTIAL FULFILLMENT  
FOR  
BACHELOR OF COMPUTER SCIENCE (HONS)  
WITH SPECIALIZATION IN  
INFORMATION SYSTEMS  
FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY

MAY 2016

The copyright of this thesis belongs to the author under the terms of the Copyright Act 1987 as qualified by Regulation 4(1) of the Multimedia University Intellectual Property Regulations. Due acknowledgment shall always be made of the use of any material contained in, or derived from, this thesis.

© *The Killers*, 2016

All rights reserved

## Declaration

I hereby declare that the work in this thesis have been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

---

*The Killers*

Faculty of Computing & Informatics

Multimedia University

Date: 17<sup>th</sup> May 2016

## Acknowledgements

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our lecturer, Mr. Wan Ruslan Yusoff, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in learning and practicing software engineering principles and activities.

Furthermore we would also like to acknowledge with much appreciation the senior students, who helped us by providing us with their experience.

A special thanks goes to our friends, who helped us by assembling the parts and giving suggestions about the task.

Many thanks go to our parents who are the reason of us being able to study in MMU. Moreover, they have invested their full effort in guiding us in achieving our goals.

Last but not least, we have to appreciate the guidance given by other lecturers as well as the workers especially in the faculty of computing and informatics.

# Contents

<b>Declaration</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>Contents</b>	<b>ix</b>
<b>Abbreviations and Acronyms</b>	<b>x</b>
<b>Management Summary</b>	<b>xi</b>
<b>1 Software Project Planning</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.1.1 Project Brief Description . . . . .	1
1.1.2 Project Objectives . . . . .	2
1.2 Project Scope . . . . .	3
1.3 Installation . . . . .	3
1.4 Execution . . . . .	4
1.5 Feedbacks and updates . . . . .	5
<b>2 Estimated Budget</b>	<b>6</b>
<b>3 Estimated Timeline</b>	<b>7</b>
3.1 Estimated Timeline . . . . .	7
3.1.1 Gantt Chart . . . . .	8

---

<b>4</b>	<b>Project Organization and Structure</b>	<b>9</b>
4.1	Manpower . . . . .	9
4.2	Project Organization and Structure . . . . .	10
<b>5</b>	<b>Software Process Model</b>	<b>13</b>
5.1	Prototyping model . . . . .	13
5.1.1	Description . . . . .	13
5.1.2	Reasons for choosing prototyping model . . . . .	15
<b>6</b>	<b>Software Components or Modules needed</b>	<b>16</b>
6.1	Software Components and Modules needed . . . . .	16
6.2	Software Architecture . . . . .	17
<b>7</b>	<b>Conclusions</b>	<b>18</b>
7.1	Conclusions . . . . .	18
7.2	Recommendations for Future Work . . . . .	18
<b>A</b>	<b>Settings in the Preamble Section</b>	<b>23</b>
A.1	Section Title . . . . .	23
A.1.1	subsection title . . . . .	23
A.1.2	subsection title . . . . .	23
<b>B</b>	<b>Inserting Inline Tables</b>	<b>24</b>
B.1	Section Title . . . . .	24
<b>C</b>	<b>Inserting Inline Graphics</b>	<b>25</b>
C.1	Section Title . . . . .	25
<b>D</b>	<b>Inserting Inline Codes</b>	<b>26</b>
D.1	Formatting Source Codes . . . . .	26
<b>E</b>	<b>Inserting Inline Equations</b>	<b>27</b>
E.1	Section Title . . . . .	27
E.1.1	subsection title . . . . .	27

# List of Figures

1.1	Gantt Chart: Estimated Timeline for a prototype . . . . .	4
3.1	Gantt Chart: Estimated Timeline for a prototype . . . . .	8
4.1	Organization Chart . . . . .	10
5.1	Prototype Model Diagram . . . . .	14
6.1	Software Architecture . . . . .	17



# List of Tables

4.1	Table of organization roles . . . . .	12
B.1	Tutorial Sections Schedule . . . . .	24

# Listings

D.1	Create mylatex1.tex file using nano small font . . . . .	26
D.2	Create mylatex1.tex file using nano tiny font . . . . .	26

# Abbreviations and Acronyms

x

---

<b>FCI</b>	Faculty of Computing and Informatics
<b>MMU</b>	Multimedia University
<b>MTS</b>	My Transportation Schedule (Software name)
<b>Q.A.</b>	Quality Assurance

# Management Summary

xi

---

Procedural design is considered the low level algorithmic design of software codes or computer programs. It covers the design of procedures, that is, the inner details of the steps and instructions the program are to execute to achieve their intended objectives.

Procedural design activities can also be interpreted as the design activities at the level of software components. The software constructor will use these procedural designs and write software codes to implement and execute those designs.

# Chapter 1

## Software Project Planning

### 1.1 Introduction

#### 1.1.1 Project Brief Description

MTS (My Transportation Schedule) is a daily activity application that could be installed in any Android (An operating system works in mobiles and tablets) device. It focuses on the people who are using the public transportation system in Cyberjaya.

The buses among cyberjaya will be provided by a GPS tracker that would track the bus and update the database. The user will be able to organize his timetable through MTS.

The user is only required to insert the time, the date and location in Cyberjaya of the event, MTS guarantees that the user will not be late on any event, by choosing the best route considering the density as well as the traffic. MTS is

designed to be able to alarm the user if the target destination cannot be reached on time.

### **1.1.2 Project Objectives**

- Minimizing the chance of being late.
- Avoid wasting time waiting in the bus station for the bus to come.
- Build a software with a good user experience (UX).

## 1.2 Project Scope

MTS covers only buses in Cyberjaya. For further prototypes, MTS might be extended to have a wider coverage and more public transportation methods such as trains and Taxis.

For starting prototypes, MTS has only four engineers working on it. Which leads to make the starting prototypes used by only few hundreds users to ease the process of finding and fixing bugs.

For the first prototypes, the application would be based on Android operation system.

## 1.3 Installation

MTS needs two servers that are the main server and the backup server to be installed. The backup server will be a copy of the main server. The backup server will be in service automatically if the main server stopped working.

Figure 1.1 shows the relationship between servers and their databases.

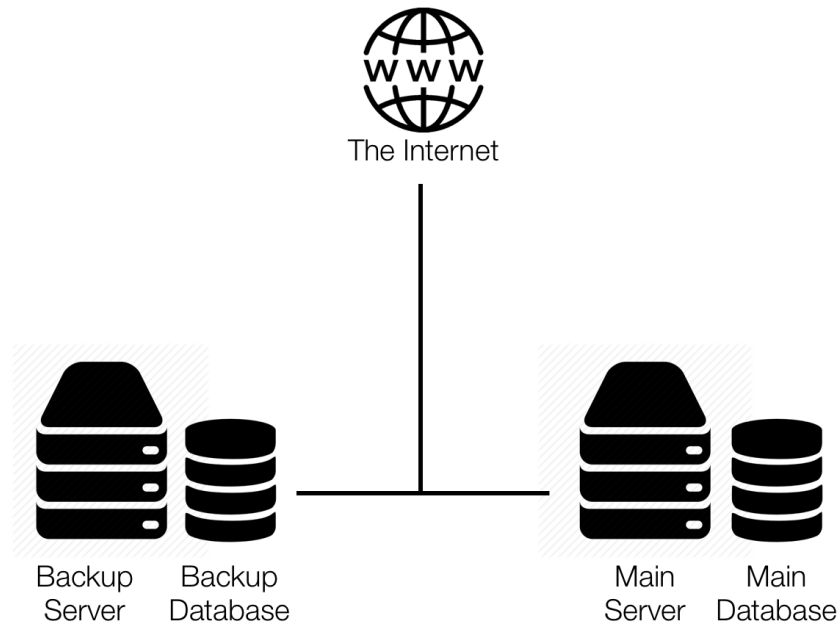


FIGURE 1.1: Gantt Chart: Estimated Timeline for a prototype

A chip equipped with a GPS tracker and a SIM card will be installed to every public transportation bus in cyberjaya. The GPS tracker will manage the location tracking function which gets every bus's location from the GPS Satellite and update it in the database, while the sim card will be providing the chip with the Internet connection.

## 1.4 Execution

The first two prototypes will be available only for the developers; For the sake of developing and improving. MTS will be available in Google Play after the release of the third prototype. For the ease of testing the prototype, 200 users



will be capable of using MTS by then. However, the software will be available for all cyberjaya users after the delivery of the fifth prototype.

During execution, feedbacks will be noted from stakeholders in order to list the following prototype's requirements.

## **1.5 Feedbacks and updates**

At the first two prototypes the developers will list their feedbacks. By the third prototype the users will be able to use and give their feedback about MTS. A feedback is considered as soon as it's received and the update will be held to the next prototype to be out.

## Chapter 2

### Estimated Budget

# Chapter 3

## Estimated Timeline

### 3.1 Estimated Timeline

Because MTS will be developed by using prototyping model, It is estimated that the thired prototype would be stable enough to be published in the Google Play - The market of android's applications -.

The second and first prototype are to find bugs and security issues in earlier time which reduce the risk and minimize the cost of fixing.

Each prototype needs at least twenty-one weeks to be fully deployed.

Furthermore, the work of a following prototype would be established on the nineteenth week of previous prototype.

### 3.1.1 Gantt Chart

Below the Gantt Chart describes estimated timeline for each prototype.

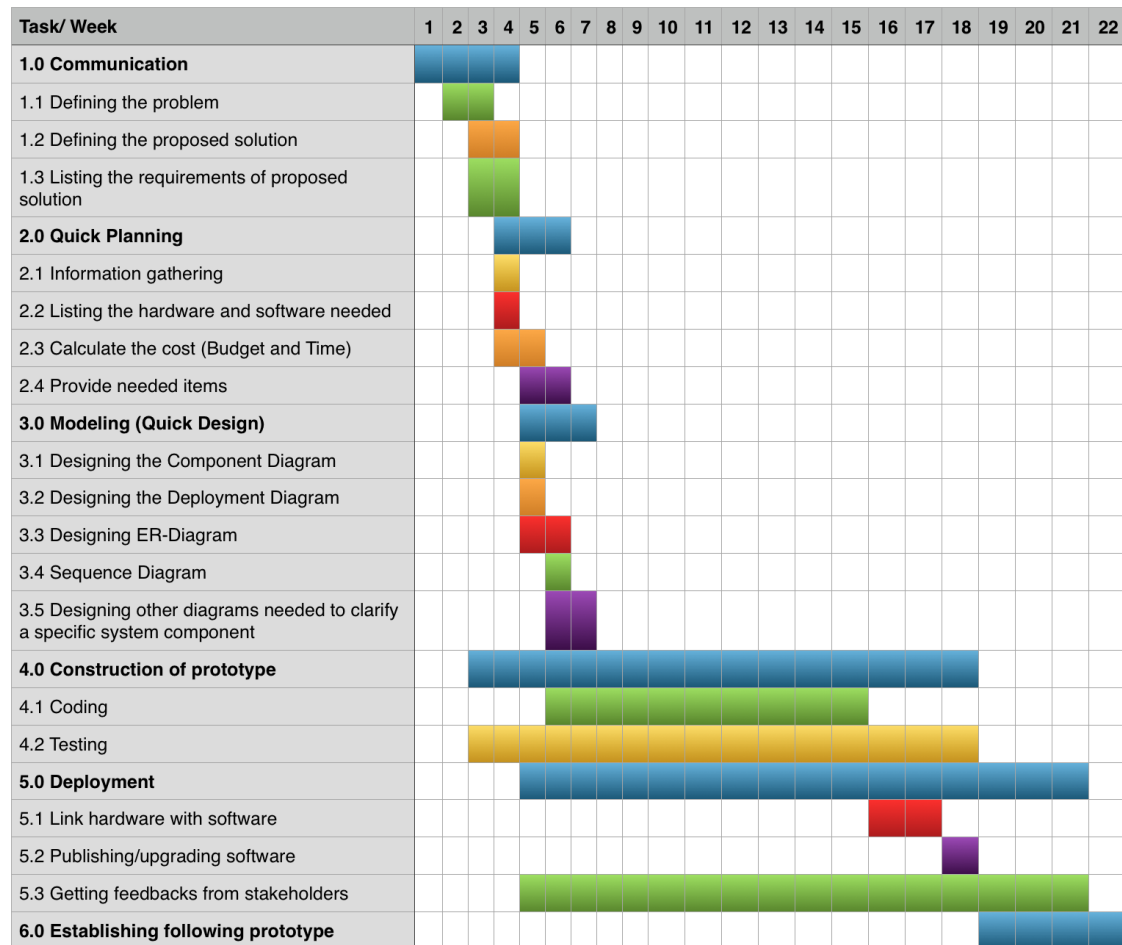


FIGURE 3.1: Gantt Chart: Estimated Timeline for a prototype

## Chapter 4

# Project Organization and Structure

### 4.1 Manpower

This project is handled by four Software engineering students. That leads everyone to handle more than one position which produce a higher quality product after each prototype.

## 4.2 Project Organization and Structure

Team members are having a role rotation plan that designed to enhance the product by team members abilities. Below is the organization chart shows basic roles:

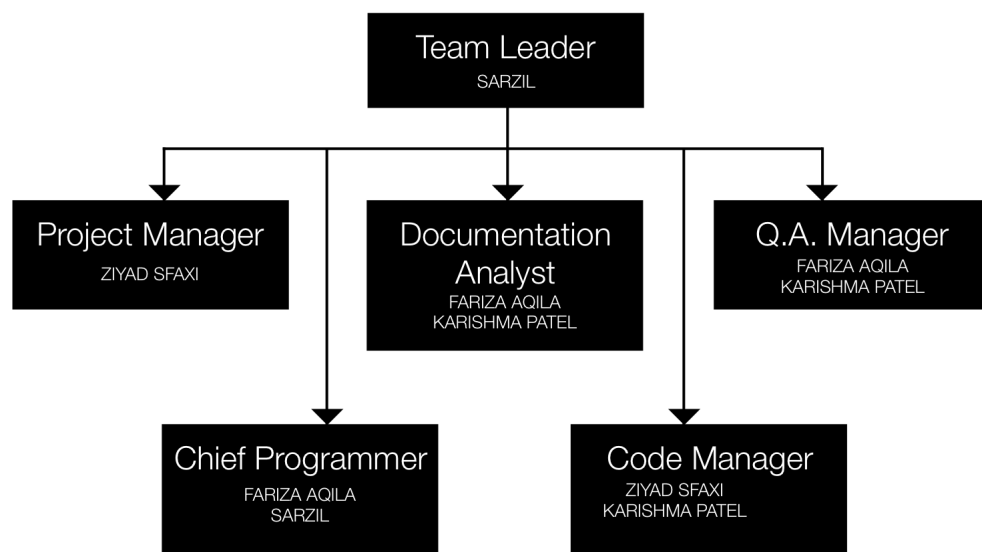


FIGURE 4.1: Organization Chart

Below is a table that describe every role:

ORGANIZATION	DESCRIPTION
<b>Team Leader</b>	<ul style="list-style-type: none"><li>• Serve as main contact with the managers.</li><li>• Ensuring the team is consistently delivering working software to the standards the department expects.</li><li>• Ensuring the team is self-organising so that we take collective responsibility for the work we do.</li></ul>
<b>Project Manager</b>	<ul style="list-style-type: none"><li>• Rotate roles to maximize the result of team abilities.</li><li>• Coordinate all those resources and the team's tasks – to make sure that work is done in the proper sequence with a minimum of time.</li><li>• Ensuring the scope project is delivered and the body of work is accomplished.</li></ul>

<b>Q.A. Manager</b>	<ul style="list-style-type: none"><li>• Assures the viability, functionality and effectiveness of essential tools.</li><li>• Anticipates program release problems and takes coorrective action, escalation as needed, to resolve and achieve commitments.</li></ul>
<b>Documentation Analyst</b>	<ul style="list-style-type: none"><li>• Ensure that all documents have no errors in file-names or submissions.</li><li>• Perform evaluations and document audits.</li></ul>
<b>Chief Programmer</b>	Convert the design and to a programming language.
<b>Code Manager</b>	Maintain the record of code files and the make file(s)

TABLE 4.1: Table of organization roles



# Chapter 5

## Software Process Model

### 5.1 Prototyping model

#### 5.1.1 Description

Prototyping model has five main phases which are listed below:

**Communication:** Gathering and discussing the requirements from and with users.

**Quick Planning:** Establishes a plan for software engineering work, addresses technical tasks, resources, work products, and work schedule.

**Modeling (Quick Design):** A quick plan that brings together customer requirements, business needs and technical considerations.

**Construction of prototype:** Combines code generation and testing uncover errors.

**Deployment, Delivery, and Feedback:** Involves delivery of software to the customers (Users) evaluation and feedback.

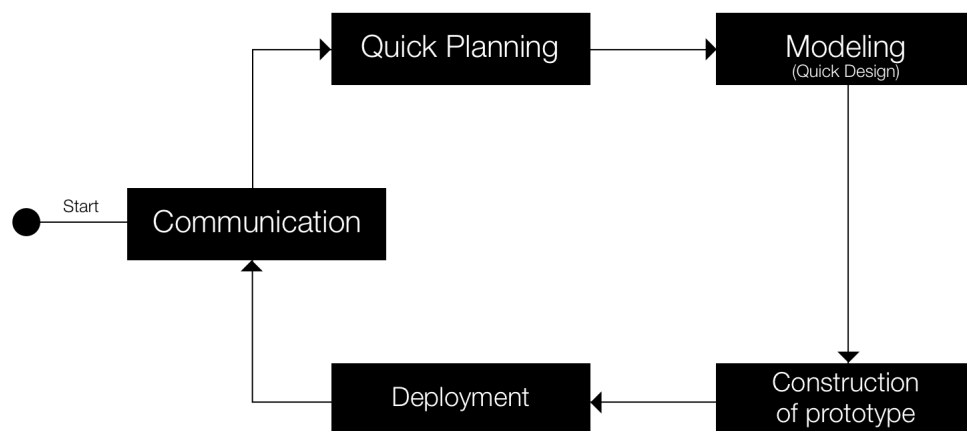


FIGURE 5.1: Prototype Model Diagram

### **5.1.2 Reasons for choosing prototyping model**

By end of 2016 the population in Cyberjaya will be about 100,00, and most of people in Cyberjaya uses buses as their cheap and easy transportation method [20]. That makes MTS highly required and might be used by most of people who live in Cyberjaya. For that reason, bugs would be found easily and bugs would need a quick fix to reduce the damages and failures.

An advantage of Prototyping model is that the quicker user feedback is available which leads to better solutions and fast error fixing. In other words, users are actively involved in the development.

MTS is an system that needs to have interactions with the end users, therefore, Prototype model should be used. Moreover, MTS is an online system which has direct interfaces with many users at the same time. MTS might has a high amount of interaction with the end-users so prototype model is best compared to other models.

# Chapter 6

## Software Components or Modules needed

### 6.1 Software Components and Modules needed

Below the list that shows and, describe components and modules needed to build MTS.

**Web server** description

**Backup server** description

**Database** description

## 6.2 Software Architecture

Below is an architecture diagram shows the software architecture.

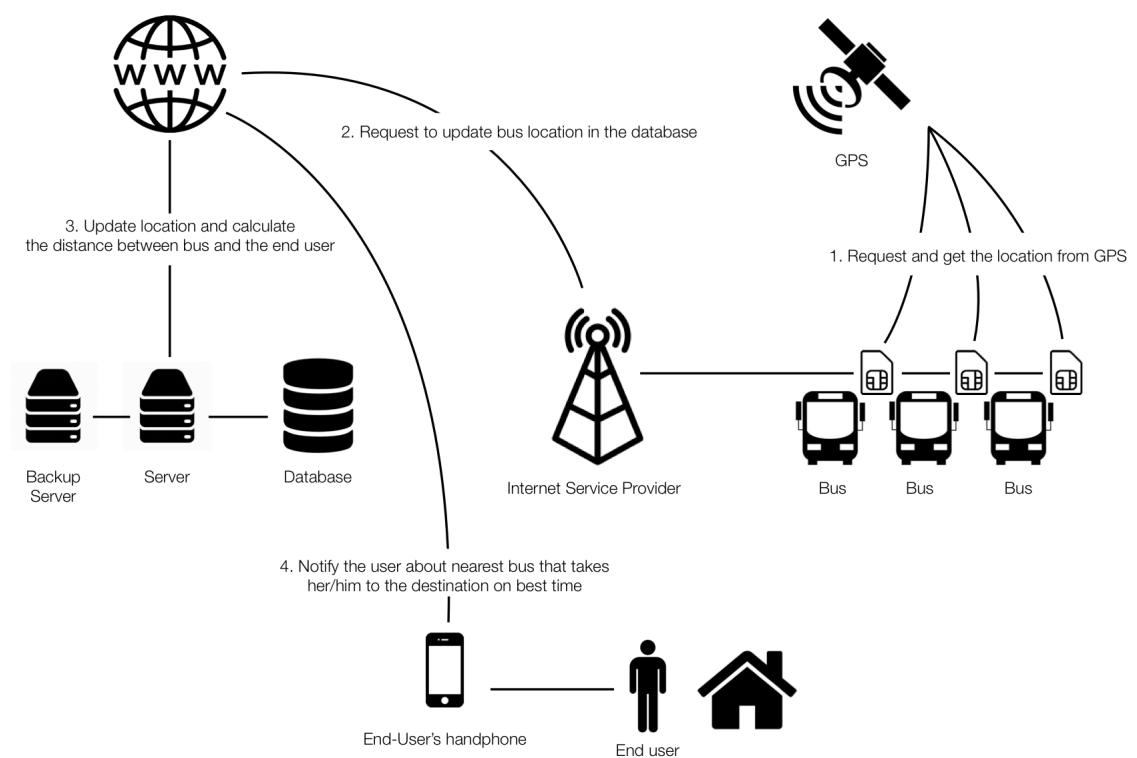


FIGURE 6.1: Software Architecture

# Chapter 7

## Conclusions

### 7.1 Conclusions

### 7.2 Recommendations for Future Work

Conclusions Recommendations for Future Work

# Bibliography

- [1] AVR Tutorials. *Microcontroller vs Microprocessor*, 2012. URL <http://www.avr-tutorials.com/general/comparison-between-microcontroller-and-microprocessor>. Retrived on 04/07/2015.
- [2] Himanshu Choudhary. *Difference between Microprocessor and Microcontroller*, 2012. URL <http://www.engineersgarage.com/tutorials/difference-between-microprocessor-and-microcontroller>. Retrived on 04/07/2015.
- [3] Genuino. *Arduino*, 2015. URL <https://www.arduino.cc/en/Guide/Introduction>. Retrived on 07/07/2015.
- [4] Genuino. *Arduino Mega 2560*, 2015. URL <https://www.arduino.cc/en/Main/arduinoBoardMega2560>. Retrived on 08/07/2015.

- 
- [5] Softpedia. *Arduino*, June 18th 2015. URL <http://www.softpedia.com/get/Programming/Other-Programming-Files/Arduino.shtml>. Retrived on 08/07/2015.
- [6] 2015. URL <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. Retrived on 13/07/2015.
- [7] James Bruce. *Arduino vs Raspberry Pi*, May 24th 2013. URL <http://www.makeuseof.com/tag/arduino-vs-raspberry-pi-which-is-the-mini-computer-for-you/>. Retrived on 13/07/2015.
- [8] Lauren Orsini. *Arduino vs Raspberry Pi*, May 7th 2015. URL <http://readwrite.com/2014/05/07/arduino-vs-raspberry-pi-projects-diy-platform>. Retrived on 13/07/2015.
- [9] Diffen. *Analog vs. Digital*. URL [http://www.diffen.com/difference/Analog\\_vs\\_Digital](http://www.diffen.com/difference/Analog_vs_Digital). Retrived on 14/07/2015.
- [10] Chigg. *Definition of Analog Signal*, 2015. URL <http://www.chegg.com/homework-help/definitions/analog-signal-4>. Retrived on 14/07/2015.
- [11] Chigg. *Definition of Digital Signal*, 2015. URL <http://www.chegg.com/homework-help/definitions/digital-signal-4>. Retrived on 14/07/2015.



- 
- [12] Wikipedia. *General-purpose input/output*, October 21st 2014. URL [https://en.wikipedia.org/wiki/General-purpose\\_input/output](https://en.wikipedia.org/wiki/General-purpose_input/output). Retrived on 14/07/2015.
- [13] Wikipedia. *Linux*, July 15th 2015. URL <https://en.wikipedia.org/wiki/Linux>. Retrived on 14/07/2015.
- [14] Wikipedia. *Ubuntu*, July 15th 2015. URL [https://en.wikipedia.org/wiki/Ubuntu\\_%28operating\\_system%29](https://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29). Retrived on 14/07/2015.
- [15] Wikipedia. *QT (Software)*, July 12th 2015. URL [https://en.wikipedia.org/wiki/Qt\\_%28software%29](https://en.wikipedia.org/wiki/Qt_%28software%29). Retrived on 14/07/2015.
- [16] Wikipedia. *programming language*, July 6th 2015. URL [https://en.wikipedia.org/wiki/Programming\\_language](https://en.wikipedia.org/wiki/Programming_language). Retrived on 15/07/2015.
- [17] Wikipedia. *C (programming language)*, July 15th 2015. URL [https://en.wikipedia.org/wiki/C\\_%28programming\\_language%29](https://en.wikipedia.org/wiki/C_%28programming_language%29). Retrived on 15/07/2015.
- [18] Wikipedia. *C++ (programming language)*, July 14th 2015. URL <https://en.wikipedia.org/wiki/C%2B%2B>. Retrived on 15/07/2015.
- [19] Wikipedia. *Postgre SQL*, July 10th 2015. URL <https://en.wikipedia.org/wiki/PostgreSQL>. Retrived on 15/07/2015.

- 
- [20] Wikipedia. *Cyberjaya*, May 16th 2016. URL  
<https://en.wikipedia.org/wiki/Cyberjaya>. Retrived on 16/05/2016.

# Appendix A

## Settings in the Preamble Section

### A.1 Section Title

To understand the basic ideas of software design Add and remove tabs Working with tabs in gedit allows you to keep an eye on several in a single window.

#### A.1.1 subsection title

The tab that is larger than the other tabs the file that is currently open. The smaller tabs indicate other files that are available to work on.

#### A.1.2 subsection title

To understand the place of software in software engineering What is design? What is Software Design?

# Appendix B

## Inserting Inline Tables

### B.1 Section Title

To understand the basic ideas of software design Add and remove tabs Working with tabs in gedit allows you to keep an eye on several in a single window.

TUTORIAL SECTIONS SCHEDULE				
Section	Lecturer	Weekday	Time	Venue
TC201	WRuslan	MON	09:00AM-11:00AM	AR1003
TC202	WRuslan	TUE	09:00AM-11:00AM	AR1003
TC203	WRuslan	WED	04:00PM-06:00PM	AR1003
TC204	WRuslan	FRI	02:30PM-04:30PM	AR1003

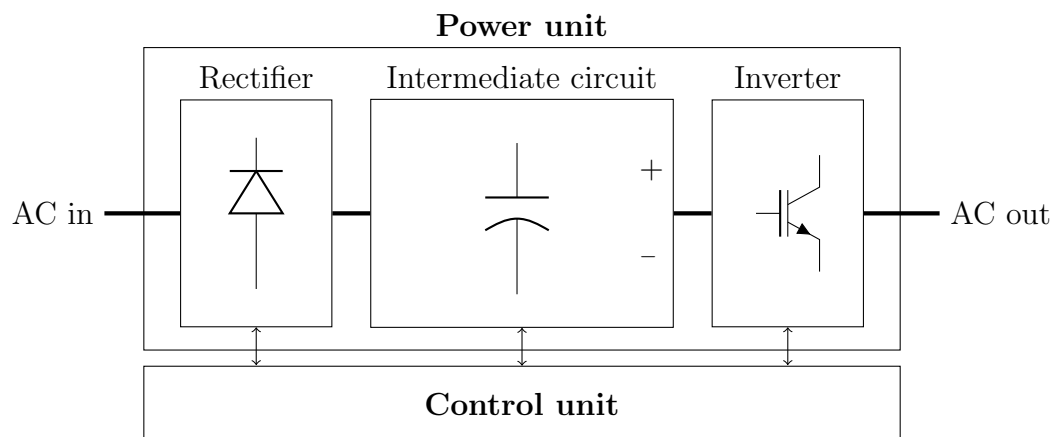
TABLE B.1: Tutorial Sections Schedule

# Appendix C

## Inserting Inline Graphics

### C.1 Section Title

To understand the basic ideas of software design Add and remove tabs Working with tabs in gedit allows you to keep an eye on several in a single window.



# Appendix D

## Inserting Inline Codes

### D.1 Formatting Source Codes

Using small font

```
1
2 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ nano mylatex1.
   tex
3 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ ls -l
4 total 4
5 -rw-rw-r-- 1 wruslan wruslan 15 Oct 10 07:48 mylatex1.tex
6 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ cat mylatex1.
   tex
7 Bismillah WRY.
8 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$
```

LISTING D.1: Create mylatex1.tex file using nano small font

Using tiny font

```
1 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ nano mylatex1.tex
2 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ ls -l
3 total 4
4 -rw-rw-r-- 1 wruslan wruslan 15 Oct 10 07:48 mylatex1.tex
5 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$ cat mylatex1.tex
6 Bismillah WRY.
7 wruslan@wruslan-ub1404-fujitsuNB:~/latex01$
```

LISTING D.2: Create mylatex1.tex file using nano tiny font

# Appendix E

## Inserting Inline Equations

### E.1 Section Title

To understand the of software design Add and remove tabs Working with tabs in gedit allows you to keep an eye on several files in a single window.

#### E.1.1 subsection title

The tab bla bla EQUATION

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \tag{E.1}$$