|  |  |
| --- | --- |
|  | MINISTRY OF EDUCATION AND TRAINING |

FPT UNIVERSITY

Capstone Project Document

**DESIGN AND CONSTRUCTION SUN DRYING WET CLOTHES SYSTEM**

|  |  |
| --- | --- |
| **Group 2** | |
| **Group members** | **Hoàng Phi Long – SE62021**  **Nguyễn Đình Phong – SE**  **Trịnh Bình – SE** |
| **Supervisor** | **Nguyễn Đức Lợi** |
| **Ext. Supervisor** | **N/A** |
| **Capstone Project code** | **DCDCS** |

-Ho Chi Minh City, **June 26th 2018**

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 2](#_Toc522039690)

[List of Tables 5](#_Toc522039691)

[List of Figures 8](#_Toc522039692)

[Definitions, Acronyms and Abbreviations 10](#_Toc522039693)

[A. Introduction 11](#_Toc522039694)

[1. Project Information 11](#_Toc522039695)

[2. Introduction 11](#_Toc522039696)

[3. Current Situation 11](#_Toc522039697)

[4. Problem Definition 12](#_Toc522039698)

[5. Proposed Solution 12](#_Toc522039699)

[5.1 Feature Functions 12](#_Toc522039700)

[5.2 Advantages and Disadvantages 12](#_Toc522039701)

[6. Functional Requirements 13](#_Toc522039702)

[7. Role and Responsibility 14](#_Toc522039703)

[8. Conclusion 14](#_Toc522039704)

[B. Software Project Management Plan 15](#_Toc522039705)

[1. Problem Definition 15](#_Toc522039706)

[1.1 Name of this Capstone project 15](#_Toc522039707)

[1.2 Problem Abstract 15](#_Toc522039708)

[1.3 Project Overview 15](#_Toc522039709)

[2. Project Organization 18](#_Toc522039710)

[2.1 Software Process Model 18](#_Toc522039711)

[2.2 Roles and Responsibility 19](#_Toc522039712)

[2.3 Tools and Techniques 20](#_Toc522039713)

[3. Project Management Plan 21](#_Toc522039714)

[3.1 System Development Life-cycle 21](#_Toc522039715)

[3.2 Plan Detail 23](#_Toc522039716)

[4. Coding Convention 26](#_Toc522039717)

[C. Software & Hardware Requirement Specification 28](#_Toc522039718)

[1. User Requirement Specification 28](#_Toc522039719)

[2. System Requirement Specification 28](#_Toc522039720)

[2.1 External Interface Requirement 28](#_Toc522039721)

[2.2 System Overview Usecase 30](#_Toc522039722)

[2.3 Usecase Details 32](#_Toc522039723)

[3. Hardware Requirement Specification 33](#_Toc522039724)

[3.1 Hardware Interface 33](#_Toc522039725)

[3.2 Communication Protocol 46](#_Toc522039726)

[4. System Attributes 49](#_Toc522039727)

[4.1 Usability 49](#_Toc522039728)

[4.2 Reliability 49](#_Toc522039729)

[4.3 Availability 49](#_Toc522039730)

[4.4 Maintainability 49](#_Toc522039731)

[4.5 Portability 49](#_Toc522039732)

[4.6 Performance 50](#_Toc522039733)

[4.7 Security 50](#_Toc522039734)

[5. Conceptual Diagram 51](#_Toc522039735)

[D. Software & Hardware Design Description 52](#_Toc522039736)

[1. Design Overview 52](#_Toc522039737)

[2. System Architectural Design 53](#_Toc522039738)

[2.1 API Web Server Architectural Design 54](#_Toc522039739)

[2.2 Android Application Architectural Design 56](#_Toc522039740)

[2.3 Hardware System Architecture 60](#_Toc522039741)

[3. Component Diagram 62](#_Toc522039742)

[4. Detailed Description 64](#_Toc522039743)

[4.1 Class Diagram 64](#_Toc522039744)

[4.2 Class Diagram Explanation 71](#_Toc522039745)

[4.3 Interaction Diagram 88](#_Toc522039746)

[5. Interface 96](#_Toc522039747)

[5.1 Guest Interface 96](#_Toc522039748)

[5.2 User Interface 98](#_Toc522039749)

[6. Database Design 106](#_Toc522039750)

[6.1 Entity Relational Database (ERD) 106](#_Toc522039751)

[6.2 Data Dictionary 106](#_Toc522039752)

[7. Algorithms 107](#_Toc522039753)

[7.1 System Control 107](#_Toc522039754)

[E. System Implementation & Testing 109](#_Toc522039755)

[F. Software & Hardware User’s Manual 110](#_Toc522039756)

[G. Acknowledgement 111](#_Toc522039757)

[H. Appendix 112](#_Toc522039758)

# List of Tables

[Table 1: General Roles and Responsibilities of Member 14](#_Toc522041085)

[Table 2: Hardware development environment requirement for DCDCS System 17](#_Toc522041086)

[Table 3: Software development environment requirement for DCDCS System 18](#_Toc522041087)

[Table 4: Roles and responsibilities 20](#_Toc522041088)

[Table 5: Tools and techniques 20](#_Toc522041089)

[Table 6: Project task planning 22](#_Toc522041090)

[Table 7: Plain Detail - Requirement Analysis 23](#_Toc522041091)

[Table 8: Plain Detail - Design 24](#_Toc522041092)

[Table 9: Plain Detail – Implementation 25](#_Toc522041093)

[Table 10: Plain Detail –Testing 25](#_Toc522041094)

[Table 11: Plain Detail –Maintenance 25](#_Toc522041095)

[Table 12: Specifications for rain sensor 34](#_Toc522041096)

[Table 13: Specifications for Arduino mega 2560 36](#_Toc522041097)

[Table 14: Specifications for DHT11 37](#_Toc522041098)

[Table 15: Specifications for BH1750 37](#_Toc522041099)

[Table 16: Specifications for LCD5110 38](#_Toc522041100)

[Table 17: Specifications for 4x4 matrix keypad 39](#_Toc522041101)

[Table 18: Specifications for limit switch 40](#_Toc522041102)

[Table 19: Specifications for DC Motor GA37 41](#_Toc522041103)

[Table 20: Specifications for solar panel 42](#_Toc522041104)

[Table 21: Specifications for charge controller 43](#_Toc522041105)

[Table 22: Specifications for GTZ5S-E battery 44](#_Toc522041106)

[Table 23: Specifications for NodeMCU 45](#_Toc522041107)

[Table 24: Data dictionary for conceptual diagram 51](#_Toc522041108)

[Table 25: Component diagram dictionary 63](#_Toc522041109)

[Table 26: API Web server class diagram dictionary 67](#_Toc522041110)

[Table 27: Hardware controller class diagram dictionary 70](#_Toc522041111)

[Table 28: Attribute dictionary for UserModel 71](#_Toc522041112)

[Table 29: Method dictionary for UserModel 71](#_Toc522041113)

[Table 30: Attribute dictionary for ModelModel 72](#_Toc522041114)

[Table 31: Method dictionary for ModelModel 72](#_Toc522041115)

[Table 32: Attribute dictionary for MessageModel 73](#_Toc522041116)

[Table 33: Method dictionary for MessageModel 73](#_Toc522041117)

[Table 34: Attribute dictionary for CustomerModel 74](#_Toc522041118)

[Table 35: Method dictionary for CustomerModel 74](#_Toc522041119)

[Table 36: Attribute dictionary for ProductModel 75](#_Toc522041120)

[Table 37: Method dictionary for ProductModel 75](#_Toc522041121)

[Table 38: Method dictionary for Mongoose 76](#_Toc522041122)

[Table 39: Method dictionary for UserController 76](#_Toc522041123)

[Table 40: Method dictionary for ModelController 77](#_Toc522041124)

[Table 41: Method dictionary for MessageController 77](#_Toc522041125)

[Table 42: Method dictionary for CustomerController 78](#_Toc522041126)

[Table 43: Method dictionary for ProductController 78](#_Toc522041127)

[Table 44: Method dictionary for Router 79](#_Toc522041128)

[Table 45: Method dictionary for Auth 79](#_Toc522041129)

[Table 46: Attribute dictionary for SystemController 80](#_Toc522041130)

[Table 47: Method dictionary for SystemController 80](#_Toc522041131)

[Table 48: Attribute dictionary for DryerController 81](#_Toc522041132)

[Table 49: Method dictionary for DryerController 81](#_Toc522041133)

[Table 50: Method dictionary for WifiHandler 81](#_Toc522041134)

[Table 51: Attribute dictionary for DCMotor 82](#_Toc522041135)

[Table 52: Method dictionary for DCMotor 82](#_Toc522041136)

[Table 53: Method dictionary for LightSensorHandler 82](#_Toc522041137)

[Table 54: Attribute dictionary for LCDHandler 83](#_Toc522041138)

[Table 55: Method dictionary for LCDHandler 83](#_Toc522041139)

[Table 56: Attribute dictionary for SwitchHandler 83](#_Toc522041140)

[Table 57: Method dictionary for SwitchHandler 83](#_Toc522041141)

[Table 58: Attribute dictionary for RFHandler 84](#_Toc522041142)

[Table 59: Method dictionary for RFHandler 84](#_Toc522041143)

[Table 60: Attribute dictionary for KeypadHandler 84](#_Toc522041144)

[Table 61: Method dictionary for KeypadHandler 84](#_Toc522041145)

[Table 62: Attribute dictionary for RainSensorHandler 85](#_Toc522041146)

[Table 63: Method dictionary for RainSensorHandler 85](#_Toc522041147)

[Table 64: Method dictionary for Central Controller 85](#_Toc522041148)

[Table 65: Attribute dictionary for BHT1750 86](#_Toc522041149)

[Table 66: Method dictionary for BHT1750 86](#_Toc522041150)

[Table 67: Attribute dictionary for Wire 86](#_Toc522041151)

[Table 68: Method dictionary for Wire 86](#_Toc522041152)

[Table 69: Attribute dictionary for DHT 87](#_Toc522041153)

[Table 70: Method dictionary for DHT 87](#_Toc522041154)

[Table 71: <Guess> Login screen fields table 97](#_Toc522041155)

[Table 72: <Guess> Login screen buttons/hyperlinks table 97](#_Toc522041156)

[Table 73: <User> Home screen fields table 99](#_Toc522041157)

[Table 74: <User> Home screen buttons/hyperlinks 99](#_Toc522041158)

[Table 75: <User> DC Motor control screen fields 100](#_Toc522041159)

[Table 76: <User> DC Motor control screen buttons/hyperlinks 101](#_Toc522041160)

[Table 77: <User> Setup and control dryer screen fields 102](#_Toc522041161)

[Table 78: <User> Setup and control dryer screen buttons/hyperlinks 102](#_Toc522041162)

[Table 79: <User> Select controlling product screen fields 103](#_Toc522041163)

[Table 80: <User> Select controlling product screen buttons/hyperlinks 104](#_Toc522041164)

[Table 81: <User> User's profile screen fields 105](#_Toc522041165)

[Table 82: <User> User's profile buttons/hyperlinks 105](#_Toc522041166)

[Table 83: Entity diagram data dictionary 106](#_Toc522041167)

# List of Figures

[Figure 1: Waterfall methodology 18](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039841)

[Figure 2: Hardware system overview usecase diagram 30](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039842)

[Figure 3: Android application overview usecase diagram 31](#_Toc522039843)

[Figure 4: System block diagram 33](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039844)

[Figure 5: Rain sensor 34](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039845)

[Figure 6: Arduino Mega 2650 R3 35](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039846)

[Figure 7: Module DHT11 36](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039847)

[Figure 8: Module Light sensor BH1750 37](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039848)

[Figure 9: Module LCD5110 38](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039849)

[Figure 10: 4x4 Matrix keypad 39](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039850)

[Figure 11: Limit switch 40](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039851)

[Figure 12: DC Motor GA37 41](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039852)

[Figure 13: 10W Solar Panel 42](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039853)

[Figure 14: Solar Charge Controller 43](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039854)

[Figure 15: GTZ5S-E Battery 44](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039855)

[Figure 16: NodeMCU v1.0 45](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039856)

[Figure 17: I2C Protocol 46](#_Toc522039857)

[Figure 18: SPI Protocol 47](#_Toc522039858)

[Figure 19: UART Protocol 47](#_Toc522039859)

[Figure 20: Arduino communicates with server through HTTP Protocol 48](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039860)

[Figure 21: Conceptual diagram 51](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039861)

[Figure 22: System overview architecture 53](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039862)

[Figure 23: API Web server architecture 54](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039863)

[Figure 24: Android application overview architecture 56](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039864)

[Figure 25: Android application internal architecture 58](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039865)

[Figure 26: Hardware system architecture 60](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039866)

[Figure 27: Component diagram 62](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039867)

[Figure 28: API Web Server Class Diagram Part 1 64](#_Toc522039868)

[Figure 29: API Web Server Class Diagram Part 1 64](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039869)

[Figure 30: API Web Server Class Diagram Part 2 65](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039870)

[Figure 31: API Web Server Class Diagram Part 3 66](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039871)

[Figure 32: Hardware system controller class diagram part 1 68](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039872)

[Figure 33: Hardware system controller class diagram part 2 69](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039873)

[Figure 34: Control system with android app sequence diagram 88](#_Toc522039874)

[Figure 35: Update system information sequence diagram 89](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039875)

[Figure 36: Control DC activity diagarm 90](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039876)

[Figure 37: Control Dryer activity diagram 91](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039877)

[Figure 38: Auto control activity diagram 92](#_Toc522039878)

[Figure 39: Determine action based on sensors’ data activity diagram 93](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039879)

[Figure 40: Determine dc motor action activity diagram 94](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039880)

[Figure 41: Determine dryer action activity diagram 95](#_Toc522039881)

[Figure 42: <Guess> Login screen 96](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039882)

[Figure 43: <User> Home screen 98](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039883)

[Figure 44: <User> Control DC Motor screen 100](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039884)

[Figure 45: <User> Setup and control dryer screen 101](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039885)

[Figure 46: <User> Select controlling product screen 103](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039886)

[Figure 47: <User> User's profile screen 104](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039887)

[Figure 48: Entity Relational Database 106](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039888)

[Figure 49: System Control overview flowchart 108](file:///D:\Capstone%20Document\Capstone%20Project%20Document.docx#_Toc522039889)

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| DCDCS | **D**esign and **C**onstruction sun **D**rying wet **C**lothes **S**ystem |
| RF | Radio frequency |
| HTTP | Hypertext transfer protocol |
| I2C | Inter-Integrated Circuit |
| UART | Universal asynchronous receiver-transmitter |
| DIY | Do it yourself |
| REST | Representational state transfer |
| API | Application programming interface |
| GPIO | General-purpose input/output |
| I/O | Input/Output |

# Introduction

## Project Information

* **Project name:** DESIGN AND CONSTRUCTION SUN DRYING WET CLOTHES SYSTEM
* **Project Code:** DCDCS
* **Product Type:** Embedded Device, Android Application, API Web Server
* **Start Date:** 14/06/2018
* **End Date:** 31/08/2018

## Introduction

In this document, we introduce a solution for automatic clothes drying system. We build a system, which use rain sensor to detect rain, ESP8266 for communication between Android application and embedded device.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system will help resolve some aspects of the problem that the current face recognition systems are facing today.

## Current Situation

Vietnam is a rainy country, with 6 months of sunshine and 6 months of rainy. Vietnamese people prefer drying their clothes under sunshine, wind over using clothes dryer or another dryer machines. When the rainy season comes, Vietnamese people tend to worry about their clothes at home being wet by rains. There are a few solutions to solve this problem as known as “Smart Clothesline Rigs”. This device really expensive and not really that smart. With 13.000.000 VND, you can have controllable system with UV light, build-in dryer and remote control within 30 meters. However, this system is not really solve the core problem: Automatically collecting clothes. Therefore, we come to this solution, helping Vietnamese people not to worry about their clothes during rainy season.

## Problem Definition

With systems currently available on market

Advantage of their system:

* UV disinfection
* Built-in dryer
* Strong structure can lift up to 25kg of clothes
* Below are disadvantages of current situation:
* Current systems have high production costs
* Hard to extend
* Control manually when the electricity is down
* Cannot automatically collecting clothes when rain

## Proposed Solution

Our proposed solution is designing and construction automatic clothes drying system called DCDCS to solve missing feature of current “Smart Clothesline Rigs”. Our system will help users automatically collect clothes when it is a rain. It is much cheaper, easy to install and mobile and extendable.

DCDCS system includes a mobile app and an embedded device with following functions:

### Feature Functions

* **Mobile App:**
  + Control the system through wireless
  + Check weather information
  + Check system status
* **Embedded Device:**
  + Check system status
  + Control system through hard buttons

### Advantages and Disadvantages

* **Advantages:**
  + Low costs
  + Can detect rain very fast
  + Can control with mobile app
  + Use solar energy and have battery to storage unused energy
* **Disadvantages:**
  + Cannot detect whether the clothes is dry or not
  + Cannot detect whether rain is over or not

## Functional Requirements

Functional requirements of the system are listed as below:

* Embedded system component:
  + RESTful API communication through wireless
* Use Arduino Mega 2560 as a central circuit unit
* Show information about the system
* Time
* Temperature
* Humidity
* Control dryer
* Control clothesline
* Power supply component:
* Power supply operates for the entire system
* Distributed voltage 5V and 12V
* Auto charging
* Storing energy
* User component:
* Control the system from Android application through wireless
* Turning on/off build-in dryer
* Set timer for dryer
* Control the clothesline
* Check system status and weather
* Edit user information
* Name
* Address
* Mobile phone
* Etc
* Mobile Application component:
* Communicate with system through wireless and by REST API
* Show information about the system
* Time
* Temperature
* Humidity
* Weather (Rain or not)
* System status

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full name | Role | Position | Contact |
| 1 | Nguyễn Đức Lợi | Project Manager | Supervisor | loinnd@fpt.edu.vn |
| 2 | Hoàng Phi Long | Developer | Leader | longhpse62021@fpt.edu.vn |
| 3 | Nguyễn Đình Phong | Developer | Member | phongndse@fpt.edu.vn |
| 4 | Trịnh Bình | Developer | Member | binhtse@fpt.edu.vn |

Table 1: General Roles and Responsibilities of Member

## Conclusion

* Research to determine and implement the appropriate MCU for the Central Control Unit and other nodes
* Design and implement integrate PCB board.
* Research and implement NoSQL Database, RESTful API, Mobile Application.
* C, C++ embedded into Arduino.
* Use software in design PCB, Schematic such as OrCAD, Proteus
* Communication technique: TCP, HTTP

# Software Project Management Plan

## Problem Definition

### Name of this Capstone project

* Official name: Design and construction sun drying wet clothes system
* Vietnamese name: Thiết kế và xây dựng hệ thống phơi đồ tự động
* Abbreviation: DCDCS

### Problem Abstract

Vietnamese people work all day long. They spend time at evening and night to do their housework. One of the housework that is washing clothes then drying them. However, Vietnam is a rainy country. During rain season, everybody very worry about their drying clothes at home getting wet.

### Project Overview

#### Current Situation

Below are the problems encountered in the project:

* **Hard to improve the system:** Our system is a very simple system. However, to improve the system is a hard mission. Our system currently cannot detect when the clothes are dry, when the rain is stopped for auto collecting clothes or continue drying wet clothes. To do so, it requires mathematics model called Hidden Markov Models. However, due to the lack of knowledge in statistics and linear algebra; we are currently unable to implement this model.
* **Lack of knowledge in telecommunication:** While using ESP8266, we found out that there are some interferences during transmission. Without telecommunication, we do have hard time to detect the problem.

#### The Purposed System

According to the technology researches, we found that the simple rain sensor and ESP8266 Wi-Fi module is capable in solving the problem. We can use rain sensor detect raining and ESP8266 for wireless communication.

We assign task responsibility vertically to make sure if any member in this project fail in our team, harm would be minimized for the project.

We also build a mobile application for real-time demonstration.

#### The Boundaries of the system

Our system provides these functions:

* Automatically control clothesline when there is a rain or at night.
* Dryer system so that user can dry their clothes on rainy days
* Control system via RF Remote control
* Control system via Button on the system
* Check system status and control system via Mobile application

#### Future plans

* Implement Hidden Markov Models (HMM) for rain forecasting
* Implement the system can determine when the rain has stopped using HMM
* Build a website for user to check their account information and control the system along with mobile application
* Build a system that can detect whenever the clothes is dry or wet

#### Development Environments

##### Hardware Development Environment Requirement

For CCU clothes drying system

|  |  |
| --- | --- |
| Component | Hardware |
| Mainboard | Arduino Mega 2560 |
| Communication | Wire and cable |
| Devices | - Module real-time clock DS1307  - Rain sensor  - Humidity and Temperature sensor DHT11  - Light sensor BH1750  - DC Motor  - Nokia 5110 LCD  - 4x4 Matrix keypad  - Limit switches  - Solar Panel  - Battery  - … |
| Power source | 5V – 12V |
| Android Device | Any android mobile phone has 3G/4G or Wi-Fi connection |

Table 2: Hardware development environment requirement for DCDCS System

##### Software Development Environment Requirement

|  |  |
| --- | --- |
| Software | Name / Version |
| Operating System | Windows 7 or above |
| Environment/Run-time | Adruino Mega 2560  NodeJS |
| Modeling tool | Draw.io for UML  Proteus 8 for PCB Board |
| IDE | Visual Studio Code  Arduino IDE |
| DBMS | MongoDB |
| Source control | Git-scm and Github |
| Communication tools | Facebook Messenger  Gmail |

Table 3: Software development environment requirement for DCDCS System

## Project Organization

### Software Process Model

This project is developed using modified waterfall model. We apply modified waterfall model because it suitable with current situation in our team. We choose this model because of the following reasons:

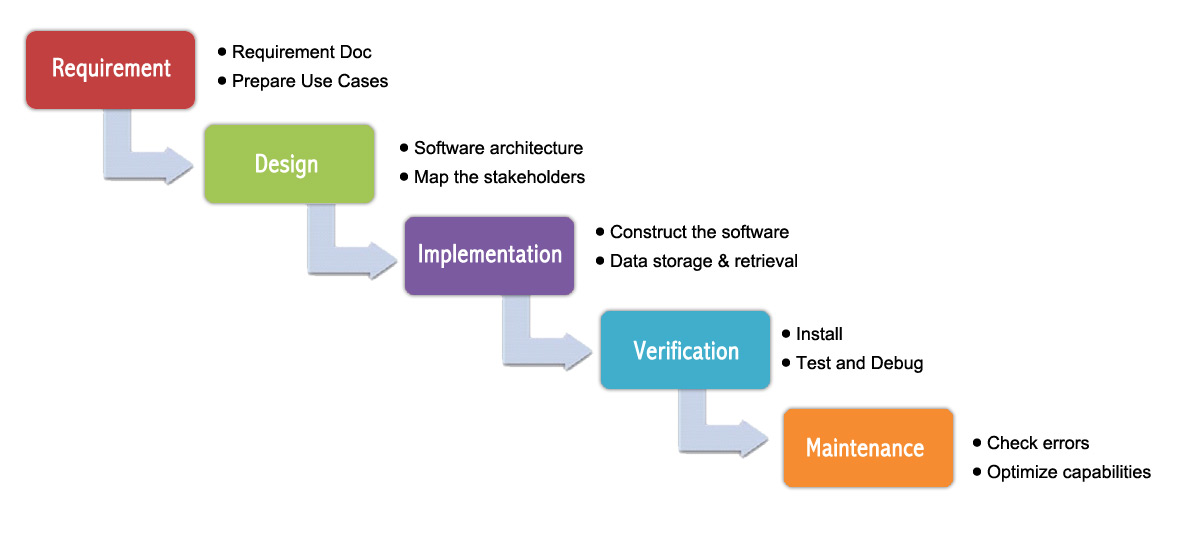
* This project is 4-months long due to the FPT University Capstone Project timeline, which can be consider a short project.
* Based on researches and current clarified face recognition system, the requirements of this project are stable, clear, fixed and well-understood by all team members.
* The Modified Waterfall Model involves verification and validation between the phases, so any deviations can be corrected immediately, providing the customer satisfaction, so this is preferred.

Figure 1: Waterfall methodology

### Roles and Responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| No | Fullname | Role | Responsibilities |
| 1 | Nguyễn Đức Lợi | Supervisor, Project Manager | * Specify user requirement * Advisor for ideas and solutions * Give out techniques and business analysis support |
| 2 | Hoàng Phi Long | Team leader, developer, tester | * Managing process * Dividing tasks for team member * Create test plan * Clarifying requirements * Coding * Testing * Verify document * Managing budget * Database design |
| 3 | Nguyễn Đình Phong | Team member, developer, tester | * Create test plan * Database design * Clarifying requirements * Prepare document * Coding * Testing * GUI Design |
| 4 | Trịnh Bình | Team member, developer, tester | * Create test plan * GUI Design * Database design * Clarifying requirements * Prepare document * Coding * Testing |

Table 4: Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tools | |
| Developing tools | Visual Studio Code  Arduino IDE |
| Database system management | MongoDB |
| Source Control | Git-scm and Github |
| Models and Diagrams tool | Draw.io |
| Techniques | |
| Embedded System | C/C++ , Arduino SDK |
| API Web Server System | ExpressJS & NodeJS |
| Mobile Application | React Native, Javascript |

Table 5: Tools and techniques

## Project Management Plan

### System Development Life-cycle

Below are all the major tasks that need to be performed sequentially during the development of the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constraints | Risks |
| Requirement Analysis | - Identify and clarify main functions.  - Prepare task plan.  - Research mechanics of collecting clothes system  - Research solar energy circuit | - Report No. 1 Introduction.  - Project Management Plan  - Task sheet  - Prototypes | 14 man-days | N/A | - Missing requirement.  - Unclear project’s scope.  - Lack of member share of understand. |
| Design | - Identify hardware and software requirements.  - Decide software architecture.  - GUI design using top-down break down.  - Design database. | - Report No. 2 Software Project Management Plan.  - Report No. 3 Software Requirement Specification.  - Report No. 4 Software Design Description. | 20 man-days | Depend on “Requirement Analysis”. | - Misunderstood or unclear system’s requirement.  - Lack of practical experience leading to unreasonable design. |
| Implementation | - Collect temperature, humidity datasets.  - Build hardware system  - Implement embedded software system  - Implement Android GUI.  - Build REST API | - Demonstration application.  - Report No.5 System Implementation & Test. | 50 man-days | Depend on “Design”. | - Lack of practical experience and knowledge.  - Human mistake.  - Broken hardwares due to wrong implementation  - Interference signal while ESP8266 communicate with Http Protocal |
| Testing | - Prepare test plan and test case.  - Test all functions and results. | Report No.5 System Implementation & Test. | 20 man-days | Depend on “Implementation”. | - Lack of experience.  - Not enough time for performing test.  - Missing bugs.  - Human resource. |
| Maintenance | - Deploy the system.  - Create the user’s manuals. | Report No.6 Software User’s Manual. | 10 man-days | Depend on “Testing”. | - Lack of experience and knowledge.  - Human mistake.  - User’s manual may be difficult for user to understand and confuse. |

Table 6: Project task planning

### Plan Detail

#### Phrase 1: Requirement Analysis

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Research mechanics of collecting clothes system | - Research on current systems, their strengths and weakness. | Hoàng Phi Long  Nguyễn Đình Phong |
| 1. Research solar energy | - Research on current systems, their strengths and weakness.  - Research how to convert solar to electricity and charge into batter | Nguyễn Đình Phong  Trịnh Bình |
| 3. Identify and clarify main functions | Define main and needed functions the system must include. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 4. Create system introduction | Complete Introduction Report. | Hoàng Phi Long |
| 5. Software Project Management Plan | Prepare Project Management Plan. | Hoàng Phi Long |
| 6. Prototype | Build a prototype of system and mobile application. | Nguyễn Đình Phong  Trịnh Bình |
| 7. SRS | Create SRS document. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |

Table 7: Plain Detail - Requirement Analysis

#### Phrase 2: Design

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify hardware and software detail design | Find out the suitable hardware and software for the system, as well as its minimum and recommended requirements. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 2. Decide software architecture | - Define the major software components and interfaces.  - Draw core flow diagram, use case diagram, prototype…  - Group meeting to review and modify. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 3. Decide Android App GUI | - UX/UI Design for Android Application | Nguyễn Đình Phong  Trịnh Bình |
| 4. Design database | - Design database for the system. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |

Table 8: Plain Detail - Design

#### Phrase 3: Implementation

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Collect temperature, humidity datasets | Program a small embedded program to collect data from sensors | Nguyễn Đình Phong  Trịnh Bình |
| 2. Construct hardware system | Build system from hardware components  Draw and print PCB board | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 3. Implement embedded software system | Develop embedded program to control the system. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 4. Implement Android GUI | Using React Native and Expo to implement Android Application GUI with fake datas | Hoàng Phi Long  Nguyễn Đình Phong |
| 5. Build REST API | Using NodeJS & ExpressJS building REST API for Mobile app and the system | Hoàng Phi Long  Trịnh Bình |

Table 9: Plain Detail – Implementation

#### Phrase 4: Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Integration testing | Write test case and testing system. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |
| 2. Alpha testing | Do alpha test with customer. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |

Table 10: Plain Detail –Testing

#### Phrase 5: Maintenance

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Installation guide | Write installation guide. | Hoàng Phi Long |
| 2. User Manual | Write user manual. | Hoàng Phi Long  Nguyễn Đình Phong  Trịnh Bình |

Table 11: Plain Detail –Maintenance

# Software & Hardware Requirement Specification

## User Requirement Specification

User is a person who use our device and mobile application. These are functions that user can use:

* Login to mobile application
* Control system to collecting or drying clothes by RF Remote control
* Control system to collecting or drying clothes by button on hardware
* Control system to collecting or drying clothes by android application
* Check information of the system
* Setup and control dryer to dry their clothes when there is a rain
* Manage/edit contact or account information (Name, Address, Mobile phone, Username, Password, …)

## System Requirement Specification

### External Interface Requirement

#### User Interface

The user interface uses English language for mobile application, hardware display interface. General requirement for graphics user interface should be simple, clear, intuitive, and reminiscent. The User interface should design with the following rules:

* User interface is created by using model top-down, left-right design.
* The interface design is an iterate process includes: design, sketching, prototyping, user assessment.
* Some design principles will be taken into consideration:
  + How To Design A Great User Interface – WDD Staff

#### Hardware Interface

Server:

* RAM: 512MB
* CPU: Intel Xeon X5550 @ 2.67GHz
* Disk Storage:
  + Operating System: Minimum 512MB (depends on Operating system)
  + Runtime Environment: 55MB
  + Application server: 60MB
  + Total: 615 MB

Android Phone:

* RAM: Minimum 512MB
* Operating System: Android 4.4 or later
* Network connection: Wi-Fi 802.11 a/b/g/n/ac, 3G, 4G/LTE
* Disk Storage: Minimum 16MB

### System Overview Usecase

#### D:\Capstone Document\Diagrams\DCDCS System Use Case.jpgHardware System Usecase

Figure 2: Hardware system overview usecase diagram

#### Android Application Usecase

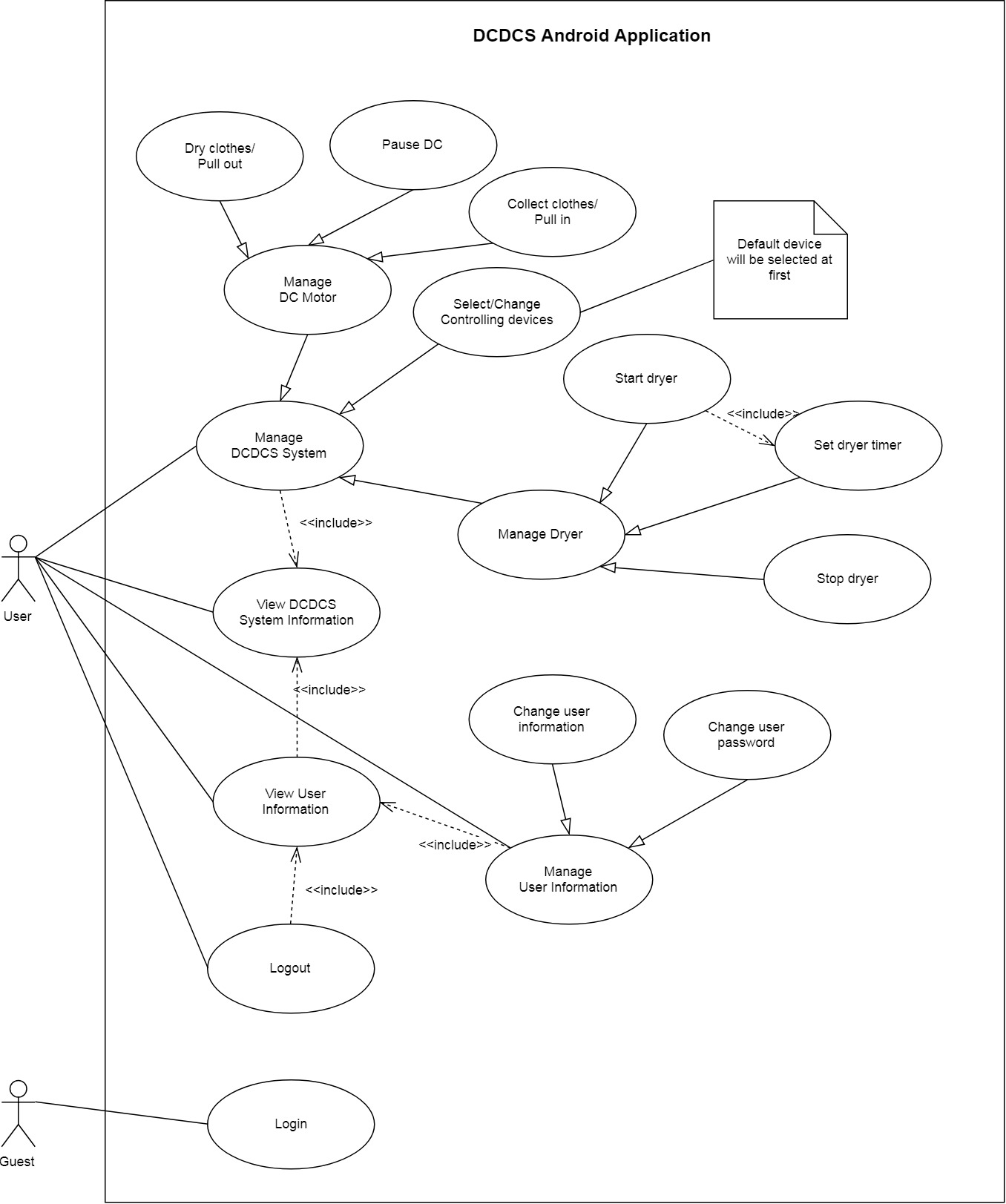


Figure 3: Android application overview usecase diagram

## Hardware Requirement Specification

### Hardware Interface

The hardware interface must have satisfied the following requirements:

* Easy to replace
* Low-cost module
* Easy to implement

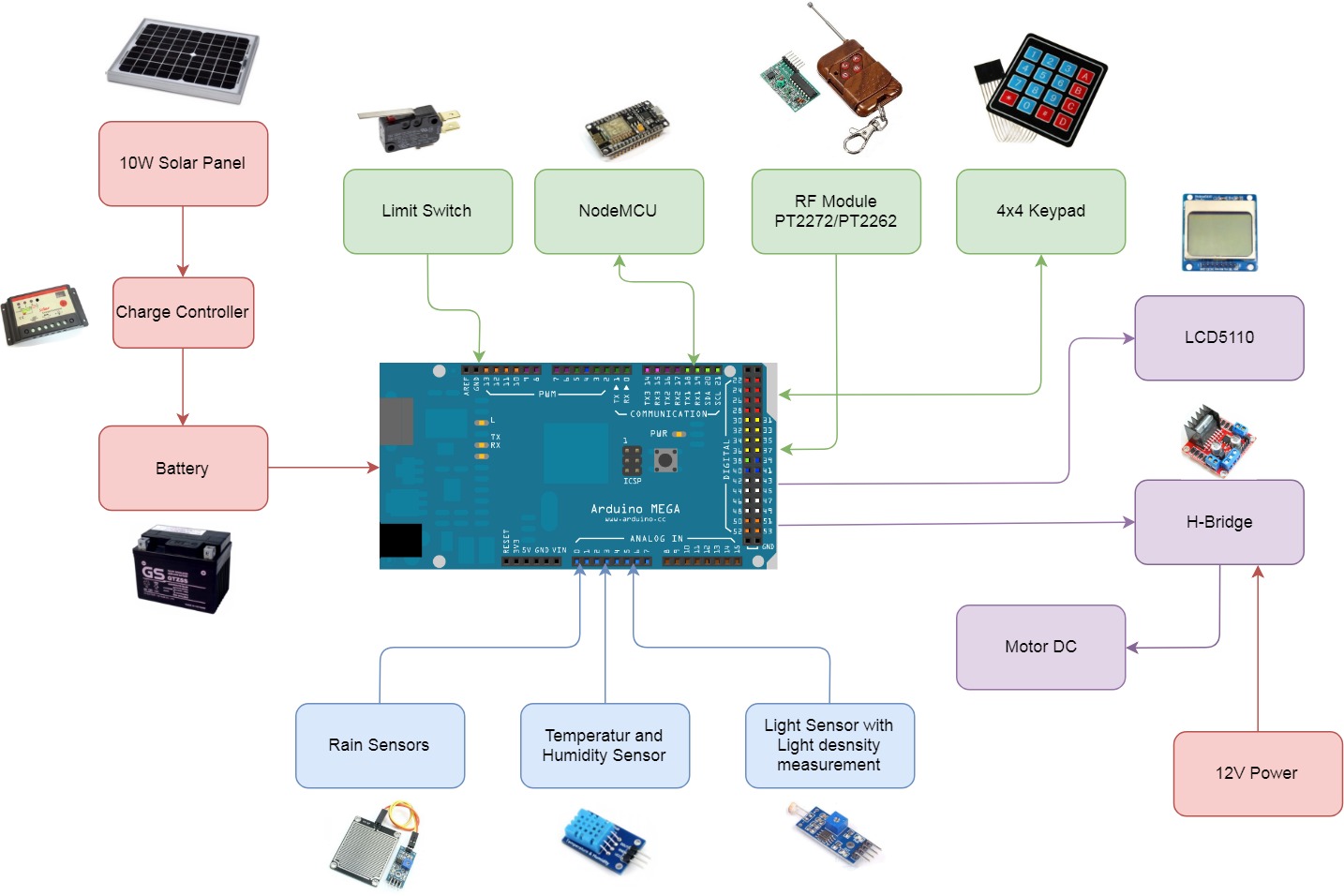
Based on project requirement we have choose following hardware components

Figure 4: System block diagram

## System Attributes

### Usability

* Users can install the system easily.
* Users can learn how to use the system quickly.
* The system is divided into several components to facilitate the installation into maintenance

### Reliability

* Control signal from mobile application controls correctly the system 95% accuracy
* The system must detect the rain correctly with 99% accuracy
* Received data from temperature and humidity sensors with 90% accuracy
* Response the action from RF Remote with 90% accuracy
* Response the action from buttons with 95% accuracy

### Availability

* The mechanical component requires electrical system to work well.
* System must reply within 3 seconds from Wi-Fi command and less than 1 second from RF Remote or buttons onboard.
* System must work in cloudy day and night
* Wi-Fi broadcast range is maximum 50 meters

### Maintainability

* The system is subdivided into modules so it is easy to replace.
* Hardware components are easy to find in the market or DIY.

### Portability

* Easy to construct
* Mobility
* User can use the mobile application on devices running Android 4.0 or later

### Performance

* Fast signal response, less than 3000ms.
* Fast system response, less than 1000ms
* System can handle 1000 requests at one time

### Security

* Each role of user has a specific permission to interact with the system.
* System always checks for authorization and authentication before doing anything.
* Only Administrator can grant permission to other roles.
* System owner can only control their own systems
* Only system owner can only control their systems

# Software & Hardware Design Description

## Design Overview

This document describes the technical and user interface design of DCDCS System. It  
includes the architectural design, the detailed design of common functions and business  
functions and the design of database model.

The architectural design describes the overall architecture of the system and the  
architecture of each main component and subsystem.

The detailed design describes static and dynamic structure for each component and  
functions. It includes class diagrams, class explanations and sequence diagrams for each  
use cases.

The database design describes the relationships between entities and details of each  
entity.  
Document overview:

* Section 1: Introduction
* Section 2: Gives an overall description of the system architecture design
* Section 3: Gives component diagrams that describe the connection and  
  integration of the system
* Section 4: Gives the detail design description which includes class diagram,  
  class explanation, and sequence diagram to details the application functions
* Section 5: Describe screens design
* Section 6: Describe a fully attributed ERD
* Section 7: Describe algorithms

## D:\Capstone Document\Diagrams\System Architecture Diagram.jpgSystem Architectural Design

Figure 22: System overview architecture

## D:\Capstone Document\Diagrams\Component Diagram.jpgComponent Diagram

Figure 27: Component diagram

COMPONENT DIAGRAM DICTIONARY: DESCRIBE COMPONENTS

|  |  |
| --- | --- |
| Component Name | Description |
| RF Component | Component to handle RF Remote |
| Rain Sensor Component | Component to handle Rain sensor |
| Keypad Component | Component to handle Keypad |
| NodeMCU Component | Component to handle Wifi, API Request/Response |
| Processing Component | Component to control the system |
| Light Sensor Component | Component to handle Light sensor |
| Dryer Component | Component to handle dryer |
| Display Component | Component to display system’s information |
| DC Motor Component | Component to handle DC motor |
| API Handler | Component to handle API Request/Response on Android |
| Controllers | A group of components that help control android app |
| (View) Login | Login screen |
| (View) Home | Home screen |
| (View) User Profile | User profile screen |
| System Database | Component to handle with database |
| Mongoose | Component to handle request/response and mapping document to Javascript object |
| Controllers | A group of components that help handling API request |
| Express Web Server | A component help build a API server |

Table 25: Component diagram dictionary

## Detailed Description

### Class Diagram

#### D:\Download\Chrome\API Web Server Class Diagram Part 1.jpgAPI Web Server

Figure 28: API Web Server Class Diagram Part 1

Figure 29: API Web Server Class Diagram Part 1

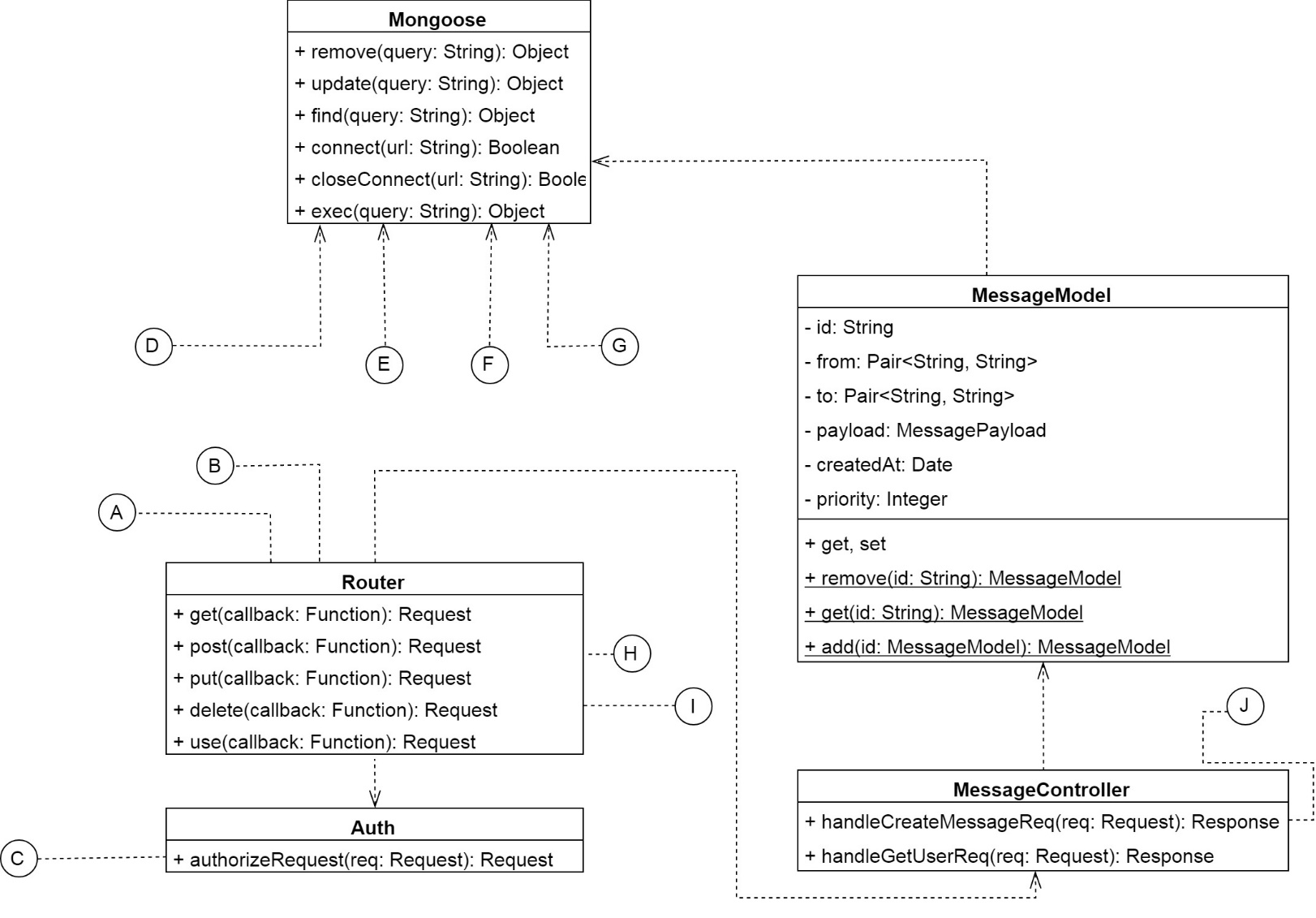


Figure 30: API Web Server Class Diagram Part 2

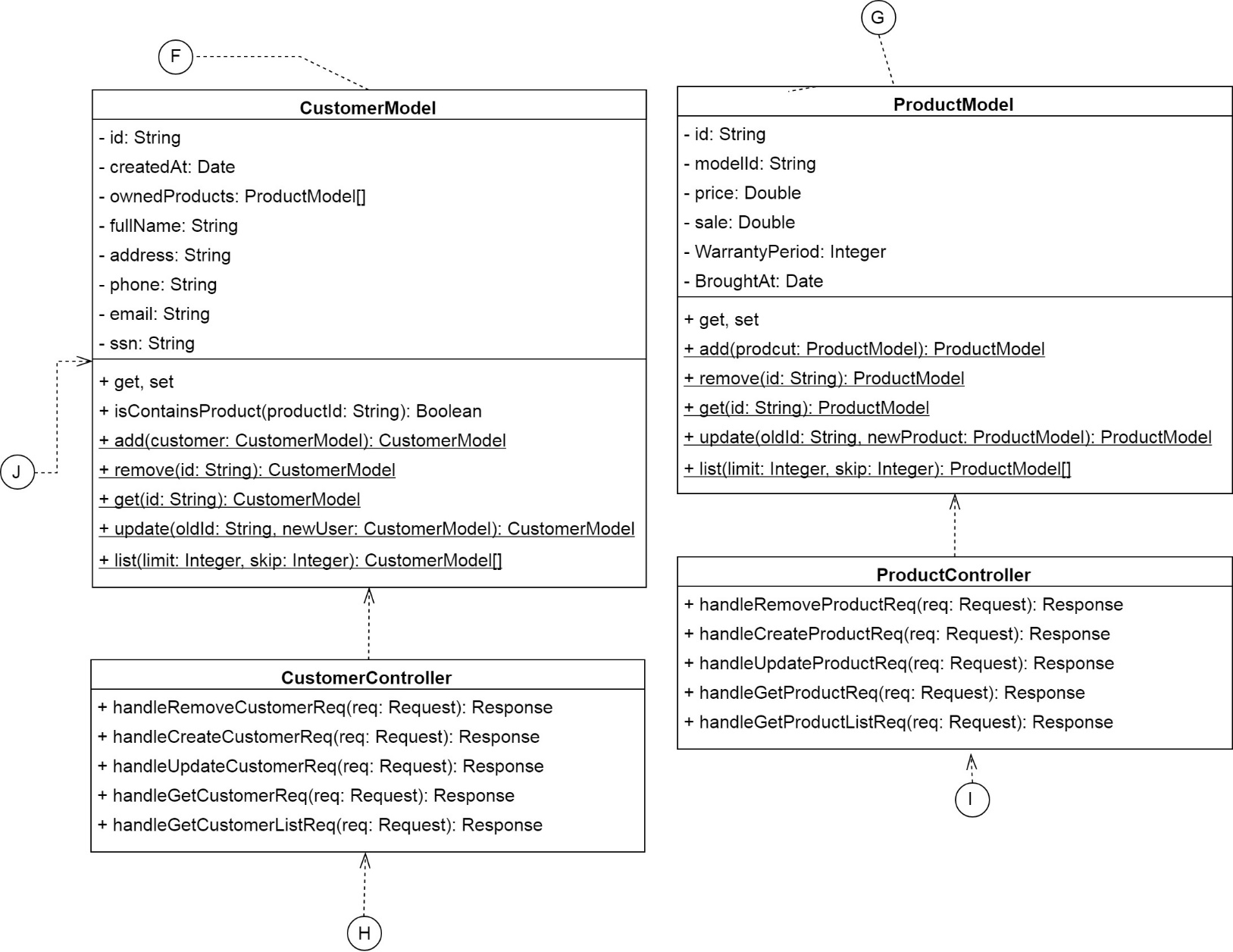


Figure 31: API Web Server Class Diagram Part 3

|  |  |  |
| --- | --- | --- |
| Class Name | Mapped Column on Conceptual Diagram | Description |
| CustomerModel | Customer | Contains customer information |
| ProductModel | Product | Contains product information |
| UserModel | User | Contains user account information |
| ModelModel | Model | Contains model of product information |
| MessageModel | Message | Contains message which used to communicate with hardware system |
| CustomerController | N/A | This class has functions that will handle any request about customer |
| ProductController | N/A | Contains functions that will handle any request about product |
| UserController | N/A | A class with functions that will handle any request about user, login, change password, etc. |
| ModelController | N/A | Contains functions that will handle any request about product model |
| MessageController | N/A | A class has functions that will allow user to publish and get action message |
| Auth | N/A | Authorize request based on access token |
| Router | N/A | A class that listen to request so that the server can call the correct controller |
| Mongoose | N/A | A class help connect and communicate, handle request/response from MongoDB |

Table 26: API Web server class diagram dictionary

#### D:\Capstone Document\Diagrams\System Controller Class Diagram Part 1.jpgHardware System

Figure 32: Hardware system controller class diagram part 1

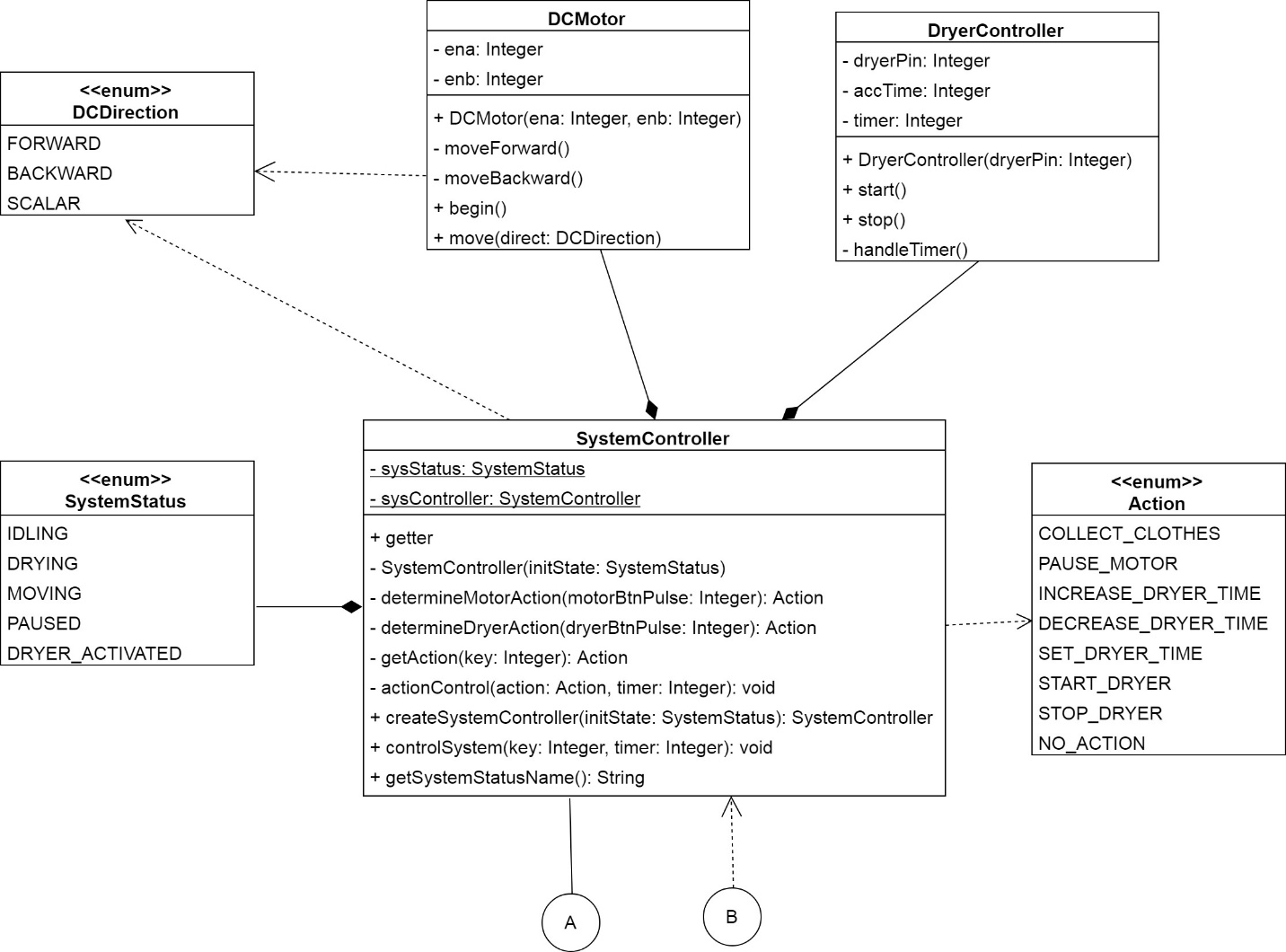


Figure 33: Hardware system controller class diagram part 2

|  |  |
| --- | --- |
| Class Name | Description |
| CentralController | The class that receive data from another class and tell SystemController class to control the system correctly |
| SystemController | This class will determine and control system with given action key |
| SwitchHandler | Handler class for limit switch |
| RFHandler | Handler class for limit switch |
| KeypadHandler | Handler class for 4x4 matrix keypad |
| LightSensorHandler | Handler class for light sensor to read light density and determine it is night or day |
| RainSensorHandler | Handler class for rain sensor. |
| WifiHandler | Handle event from NodeMCU that send through I2C Protocol |
| LCDHandler | A class that help print to LCD more easier |
| Wire | External library that help communicate with another device via I2C Protocol |
| DHT | An external library that help reading data from DHT Module |
| BHT1750 | An external library that help reading data from Light Sensor Module |
| DCMotor | This class help controlling dc motor to collect or dry clothes |
| DryerController | This class help controlling dryer fan |
| Action | This is an enum that descriptions the control action of the system |
| SystemStatus | This is an enum that descriptions the status of the system |
| DCDirection | This is an enum that descriptions the status of the dc motor |

Table 27: Hardware controller class diagram dictionary

## Database Design

### D:\Capstone Document\Diagrams\image\Entity Diagram.jpgEntity Relational Database (ERD)

Figure 48: Entity Relational Database

### Data Dictionary

|  |  |
| --- | --- |
| Entity Name | Description |
| Customer | Contains information of customer who brought our product |
| Product | Contains information about product |
| Model | Contains information about product’s model |
| User | Contains information about account of the system |
| Messages | A message queue, contains a message to communicate with hardware system |

Table 83: Entity diagram data dictionary

## Algorithms

### System Control

#### Definition

System has many ways to control the system; i.e. RF Remote, Android application, hardware button. From these controllers, they can control many another devices like DC Motor to collect or dry clothes.

#### Define Problem

While using multiple controller at the same time. It causes a collision that leading to the system doesn’t work correctly.

#### Solution

We use one thread and blocking I/O to sequentially reading each controller. Therefore, when we’re handling a single controller. Another controller will be ignored.

#### Pros & Cons

* Pros:
  + No more collisions
  + Easy to control because the system now works on priority of the controller
  + Easy to extends when there are new controller
  + Memory reduced due to using only a single thread
* Cons:
  + An action takes longer time than user to complete (due to the priority)

#### Algorithm Complexity

* Time: O(n) with n is the number of controller
* Space: O(1) because we don’t use any additional spaces

#### Overview Flowchart

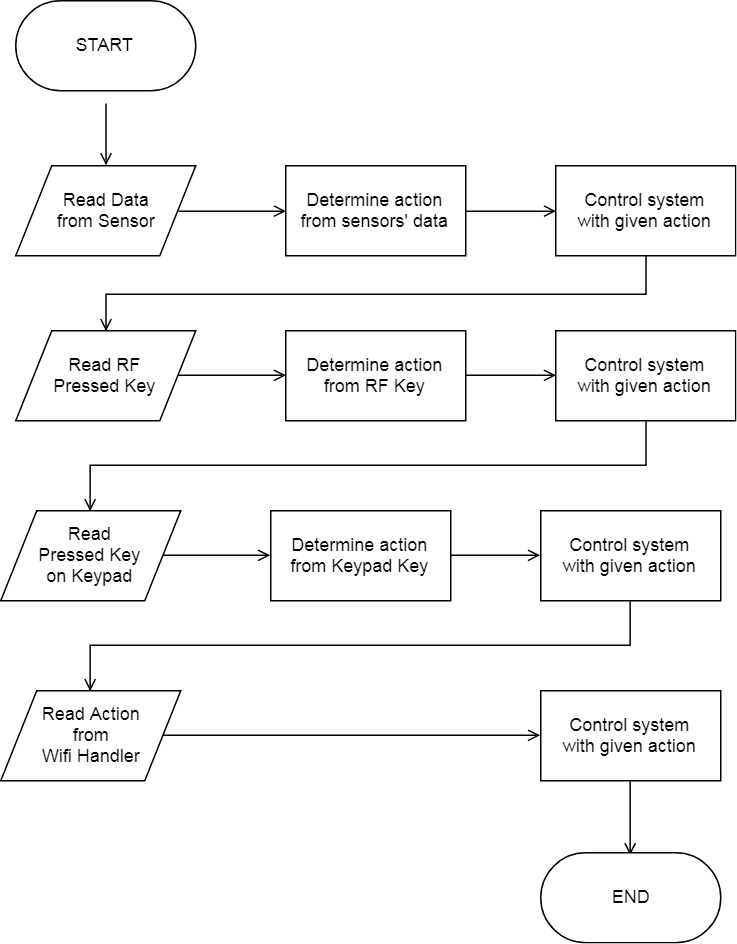


Figure 49: System Control overview flowchart

# Task Sheet

# Appendix

* Flux Architecture: <https://facebook.github.io/flux/>
* Bit Twiddling Hacks: <https://graphics.stanford.edu/~seander/bithacks.html>