|  |
| --- |
|  |
| Capstone Project Document |

**Mini Explorer System**

----------------------------------------------------------------

|  |  |  |
| --- | --- | --- |
| **MEx Team** | | |
| **Group Members** | Luyện Bảo Anh | SE03747 |
| Phạm Minh Hoàng | SE03769 |
| Lê Xuân Hướng | SE03388 |
| Phùng Đức Luật | SE03164 |
| Đỗ Cao Phong | SE03196 |
| Đặng Ngọc Tú | SE03591 |
| **Supervisor** | Hoàng Xuân Sơn | |
| **Project code** | MEx | |

**- Hoa Lac, 01/2017 –**

# SIGNATURE PAGE

AUTHOR:

REVIEWERS:

APPROVAL: Hoang Xuan Son

Supervisor

Record of change

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Item | A,M,D | Change Description | Reason for Change | Rev. Number | Author |
| 22/05/2017 | Create Introduction | A | Create Introduction | Report 2 | 1.0 | TuDN |
| 14/06/2017 | Requirements | A |  | Report3 | 1.0 | LuatPD |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Project management plan

**2.1. Definition Problem**

The report 1 is clearly specified reason why MEx project was chose to develop. It is an overview concept about MEx system and be discussed some main function of existing system.

You now have the knowledge of the system’s scope. This document will present project planning to get the target. All the tasks and time to implement, the resource of the system, and the risk maybe meet during development.

***2.1.1. Name of this Capstone Project***

This Capstone project named Mini Explorer System, abbreviated as MEx.

1. ***Boundaries of the System***
2. *Boundaries of the System*

The system under development of this Capstone Project will include:

* The controller has the task of sending the request via wireless, saving control information, controlling device.
* Wireless is an information bridge between the controller and car.
* A central circuit board in the car has responsible for data exchange with the gateway through Arduino to transmit, receive and process information from user.
* User manual, Test Document
* Design circuit broad, Design Document
* Source code Android App and Arduino

*2. Development Environment*

Below is the list of hardware and software requirements needed for development environment:

***Hardware requirements***

o Develop:

* + Arduino/WeMOS
  + Raspberry Pi
  + Sensor, servo motor, resistors, capacitors, wire…
  + Personal Computers with 4 Gigabytes of RAM or more
  1. Test:
     + Personal Computers

***Software requirements***

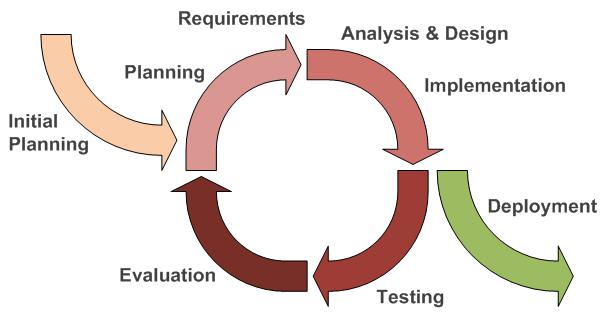
o Operating System: Windows 8.1, 10 Pro – 64bit

o Design software: Proteus 7.8

o IDEs: Android Studio v5.0 and SDK tools, JDK 7, Arduino-1.6.4-windows,Python

o Document: Microsoft Office 2016, Microsoft Project 2016

1. **Project organization**
   1. ***System Process Model***



*Figure 2-1: Iterative and Incremental Software Process Model*

This figure above describes the information and products flow lifecycle process model. MEx project uses the Iterative and Incremental Software Process Model.

Iterative and Incremental Software Process Model is a method of software development that is model around a gradual increase in feature additions and a cyclical release and upgrade pattern.

The Iterative and Incremental Software Process Model is most use when the scope of the project is big, the major requirements were defined clearly, some more detail will be added in time, and for the newbie group in software development.

By using this software process model, we break down the developing system task into series of smaller tasks are completed separately, evaluated, and subsequently re-worked until the system’s performance adequately. In addition, the iterative model is easier than other models when the issues were discover. They are feedback to the team, and solution found while the project is still in development.

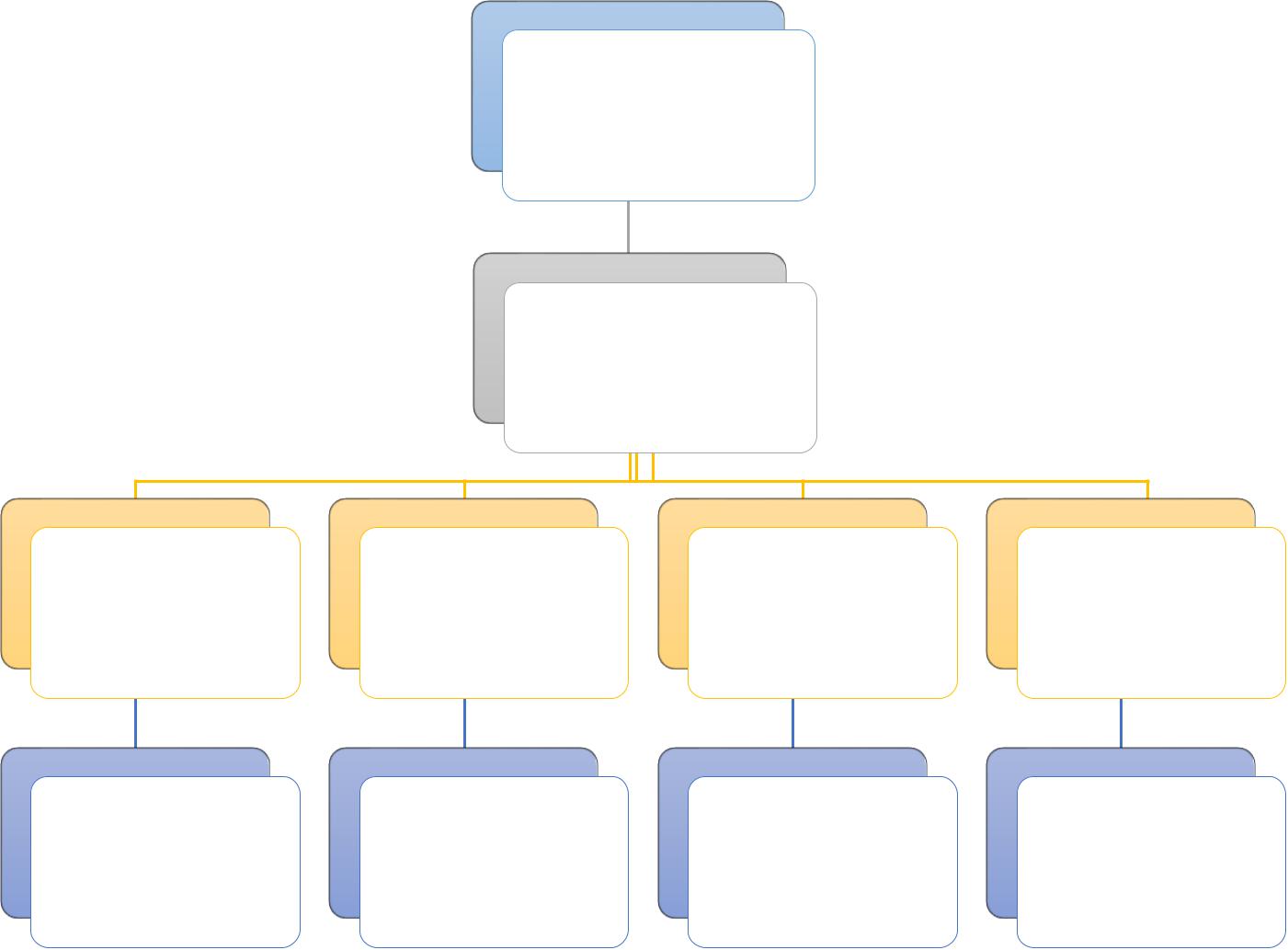
1. ***Roles and Responsibilities***
   1. *Organization and Structure*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Roles** |  |  | **Responsibility** |  |  |
|  |  |  |  |  |  |  |
|  | Project Manager |  |  | Planning developing schedules, allocating resources, keeping | |  |
|  |  |  |  | on schedule, coordinating communication, generally | |  |
|  |  |  |  | responsible for keeping the team’s focus on main goal and tries | |  |
|  |  |  |  | to keep the project team focused on the right goal at a time. | |  |
|  |  |  |  |  | |  |
|  | Technical Leader |  |  | Responsible for the underlying architecture for the hardware |  |  |
|  | (Hardware and Software) |  |  | system and software program, assigning tasks, mentoring |  |  |
|  |  |  | people, reporting. Technical leaders is a reference book for |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | other team members. |  |  |
|  |  |  |  |  |  |  |
|  | Quality Assurance Manager |  |  | Ensuring the products meet the certain standards of quality | |  |
|  |  |  |  | from requirements. | |  |
|  |  |  |  |  | |  |
|  | Test Leader |  |  | Responsible for test execution, including test set-up and test |  |  |
|  |  |  |  | run, evaluation of test run and error recovery, defect logging |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | and test results recording. |  |  |
|  |  |  |  |  |  |  |
|  | Developer |  |  | Involve to code product | |  |
|  |  |  |  |  | |  |
|  | Designer |  |  | Involve to design product |  |  |
|  |  |  |  |  |  |  |
|  | Tester |  |  | Involve to test product | |  |
|  |  |  |  |  |  |  |

*Table 2-1: Project Structure*

|  |  |
| --- | --- |
| *2.2.2.2. Project Team Member* | |
|  |  |
|  |  |
| **Team Member** | **Roles** |
|  |  |
| AnhLB | Project Manager, Technical Leader, Developer, Tester |
|  |  |
| HuongLX | Designer, Technical Leader, Developer, Test |
|  |  |
| PhongDC | Technical Leader, Developer, Tester Leader |
|  |  |
| HoangPM | Designer, Developer, Tester |
|  |  |
| LuatPD | Designer, Developer, Tester |
|  |  |
| TuDN | Designer, Developer, Tester |
|  |  |

*Table 2-2: Project Team Member*

SonHX (Supervisor)

AnhLB

(Project Manager)

Technical Designer Developer Testing and QA

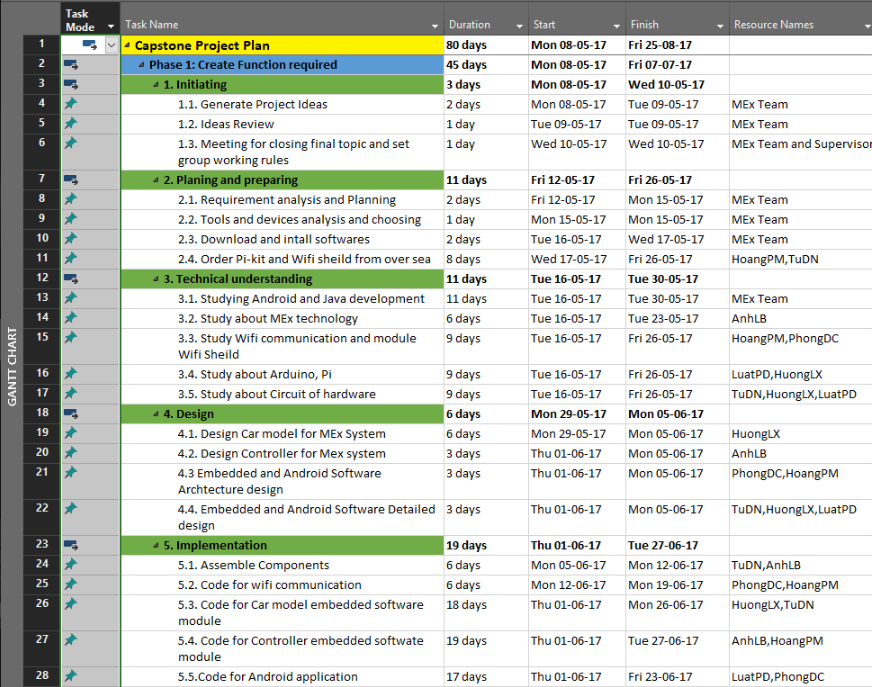
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - AnhLB | - HoangPM | All of the | All of the |  |
| - HuongLX | - LuatPD |  |
| member | member |  |
| - PhongDC | - TuDN |  |
|  |  |  |

*Figure 2-2: Project Team Member*

***2.2.3. Tools and Techniques***

* Programing languages: Java, Android, Python
* Process Model: Iterative and Incremental Software Process Model.
* IDEs: Android Studio, Arduino v1.6.4
* Design tool: Altium Designer 9
* Other:
  + Google driver for desktop
  + Microsoft Word 2016
  + Microsoft Excel 2016
  + Microsoft PowerPoint 2016
  + Microsoft Project 2016

1. **Project management plan**





*Figure 2-3: Project Management Plan*

Refer to [MEx\_ProjectPlan.mpp]

***2.3.2. Human Resource***

Human resource

* + - Team member
    - Supervisor

Non – human resource

* + Equipment: Personal Computers, Arduino
  + Building: FPT University, Thachhoa, Thachthat, Hanoi
  + Building: FPT University’s Library, Thachhoa, Thachthat, Hanoi

***2.3.3. Meeting Minutes***

All meeting minutes will be written follow this template:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Meeting/Project*** | Mex | |  |  |  |
|  |  |  |  |  |  |
| ***Date of Meeting:*** | *15/5/2017* |  | ***Time: (Type)*** | *2hours* |  |
|  |  |  |  |  |  |
| ***Meeting Called By:*** | *AnhLB* | | ***Location:*** | *FPT University* |  |
|  |  |  |  |  |  |
| ***Note Taker:*** | *TuDN* | | ***Time Keeper:*** | *PhongDC* |  |
|  |  |  |  |  |  |
| 1. Meeting Objective: |  |  |  |  |  |
|  |  |  |  |  |  |
| Brainstorming all functions of systems | | |  |  |  |
|  |  |  |  |  |  |
| 2. Attendance |  |  |  |  |  |
|  |  |  |  |  |  |
| Name | Roles |  | E-mail | Phone |  |
|  |  |  |  |  |  |
|  | Project |  | [@fpt.edu.vn](mailto:Duclqse02946@fpt.edu.vn) |  |  |
|  |  |  |  |  |  |
|  | Tester |  | [@fpt.edu.v](mailto:Tungntse02945@fpt.edu.vn)n |  |  |
|  |  |  |  |  |  |
|  | Developer |  | [@fpt.edu.vn](mailto:Anhpvse02918@fpt.edu.vn) |  |  |
|  |  |  |  |  |  |
| 3. Content: |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | *Table 2-3: Meeting Minutes Template* | | |  |  |

***2.3.4. Risk Management Plan***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Probability** | |  | **Prevention** |  |  | **Correction** |  | **Impact** |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 1 | **Miscommunication** | Medium | |  | After a meeting, one |  |  | When it becomes |  | High |  |
|  |  |  | |  | group member |  |  | clear that |  |  |  |
|  |  |  | |  | creates an interview |  |  | miscommunication |  |  |  |
|  |  |  | |  | report. Every |  |  | is causing problem, |  |  |  |
|  |  |  | |  | participant or |  |  | the team members |  |  |  |
|  |  |  | |  | absence person |  |  | are gathered in a |  |  |  |
|  |  |  | |  | should get a copy of |  |  | meeting to clear |  |  |  |
|  |  |  | |  | this report. Team |  |  | thing up. |  |  |  |
|  |  |  | |  | members should not |  |  |  |  |  |  |
|  |  |  | |  | hesitate to ask |  |  |  |  |  |  |
|  |  |  | |  | questions if they are |  |  |  |  |  |  |
|  |  |  | |  | unclear. |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 2 | **Design Error** | High | |  | The design should be |  |  | When error in the |  | Medium |  |
|  |  |  | |  | reviewed very |  |  | design are noticed |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  | critically. Team leader |  |  | PM or team leader |  |  |  |
|  |  |  | |  | should be consulted |  |  | should be consulted |  |  |  |
|  |  |  | |  | frequency on his |  |  | to help correct the |  |  |  |
|  |  |  | |  | opinion about the |  |  | design errors as soon |  |  |  |
|  |  |  | |  | feasibility and the |  |  | as possible. Also all |  |  |  |
|  |  |  | |  | correctness of certain |  |  | the work, that |  |  |  |
|  |  |  | |  | design decisions. |  |  | depends on the |  |  |  |
|  |  |  | |  |  |  |  | faulty design, should |  |  |  |
|  |  |  | |  |  |  |  | be halted until the |  |  |  |
|  |  |  | |  |  |  |  | error is corrected. |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 3 | **Hardware Failure** | Low | |  | Check all of hardware |  |  | Creating a list of |  | High |  |
|  |  |  | |  | before buying. Being |  |  | store that is selling |  |  |  |
|  |  |  | |  | sure and testing about |  |  | this hardware. |  |  |  |
|  |  |  | |  | current and volt of this |  |  | Checking it exist if |  |  |  |
|  |  |  | |  | hardware before |  |  | having plan goes to |  |  |  |
|  |  |  | |  | using. |  |  | buy. |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 4 | **Illness or absence** | Medium |  |  | Team members should |  |  | By ensuring that |  | Medium |  |
|  | **of team member** |  |  |  | warn their team leader |  |  | knowledge is shared |  |  |  |
|  |  |  |  |  | timely before a |  |  | between team |  |  |  |
|  |  |  |  |  | planned period of |  |  | members, work can |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | absence. |  |  | be taken over |  |  |  |
|  |  |  |  |  |  |  |  | quickly by someone |  |  |  |
|  |  |  |  |  |  |  |  | else if a person gets |  |  |  |
|  |  |  |  |  |  |  |  | ill. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | **Requirement** | Medium |  |  | Carefully brainstorm |  |  | Team meetings with |  | High |  |
|  | **change** |  |  |  | system’s features |  |  | supervisor to |  |  |  |
|  |  |  |  |  | among team members. |  |  | determine whether |  |  |  |
|  |  |  |  |  | Regularly hold |  |  | new feature should |  |  |  |
|  |  |  |  |  | meeting to define and |  |  | be implemented or |  |  |  |
|  |  |  |  |  | discuss all the features |  |  | not. Team leaders |  |  |  |
|  |  |  |  |  | of systems. Design |  |  | create |  |  |  |
|  |  |  |  |  | system carefully. |  |  | implementation plan |  |  |  |
|  |  |  |  |  | Analyze all the |  |  | for implemented |  |  |  |
|  |  |  |  |  | possible cases to |  |  | features and sent to |  |  |  |
|  |  |  |  |  | minimize the change |  |  | team members. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | **Time shortage** | High |  |  | Project manager |  |  | Lacking time is the |  | High |  |
|  |  |  |  |  | should create more |  |  | fatal problem, can |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | spare time and |  |  | run project to failure. |  |  |  |
|  |  |  |  |  | calculate plus 20% |  |  | PM should analysis |  |  |  |
|  |  |  |  |  | buffer time. |  |  | and has change on |  |  |  |
|  |  |  |  |  |  |  |  | the next phase. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

*Table 2-4: Risk Management Plan*

***2.3.5. Communication Plan***

* 1. *Communication between team members*
* *Weekly meeting schedule*: By using Iterative and Incremental Process Model, MExProject System will be divided into a series of small tasks, each task will be assigned to team members by Technical Leader and depend on difficulty, and Technical Leader will assigned deadlines for each task. We will have a meeting every Thursday, Friday and Monday to report

the progress of whole team’s tasks. Any member who doesn’t finish his/her task (without reasonable explanation), will be fined. If there is any issue, we will discuss and find solution together. If it is too difficult and can’t be solved by ourselves, we will ask our supervisor for advises.

* Unscheduled meeting: If someone has an important problem want to be solved immediately, we will have a meeting for discussion.
* Communication channel: Our main communication channels are face-to-face meeting, email, Facebook, Skype. However, we sometimes can make a phone call or instant message if someone has problem.
  + 1. *Communication with Supervisor*
  + *Face-to-face* meeting: Weekly on every Thursday afternoons to make sure thatsupervisor can keep tracking of the team’s progress.
  + *E-*mail: Gmail is the fastest way to get device and document checking fromsupervisor.
  + *Mobile phone:* is used to get time and place arranged for the meeting every weekly.

1. **Projection Directory**

|  |  |  |  |
| --- | --- | --- | --- |
| Main folder | Sub-folder | Purpose |  |
|  |  |  |  |
|  | Meeting minutes | Store project meeting minutes |  |
|  |  |  |  |
|  | Report 1 | Store final deliverables of report 1 |  |
|  |  |  |  |
|  | Report 2 | Store final deliverables of report 2 |  |
| Project’s |  |  |  |
| Report 3 | Store final deliverables of report 3 |  |
| Document |  |  |  |
| Report 4 | Store final deliverables of report 4 |  |
|  |  |
|  |  |  |  |
|  | Report 5 | Store final deliverables of report 5 |  |
|  |  |  |  |
|  | Report 6 | Store final deliverables of report 6 |  |
|  |  |  |  |
|  | Final Report | Store final deliverables of final report |  |
|  |  |  |  |
| Plan |  | Store project plan, Task list |  |
|  |  |  |  |
| Resource |  | Store template needed in project |  |
|  |  |  |
| Tool | Store tool needed in project |  |
|  |  |
|  |  |  |  |
| Working space | Each team members has a folder | Team member’s working area |  |
|  |  |  |  |
| Reference |  | Store reference needed in project |  |
|  |  |  |  |
|  | *Table 2-5: Projection Directory* | |  |

# Requirements

**3.1. User Requirements Specification**

## 3.1.1 User requirement

* User use smartphone ( or remote controller) connect to car’s raspberry pi through WIFI
* User can use smartphone or remote controller to control the movement of the car and the camera
* User can use smartphone to switch control between smartphone or remote controller
* In remote controller mode, user can use smartphone to see the camera view

## Car requirement

* Right direction under the control of user
* Operation of the car must be stable and safe
* Easy to receive signal from user

## Android application requirement

* Well design with minimum cost and maximum quality
* Has long durability in period of time
* Be simple and user-friendly

## Remote Controller requirement

* Auto connect to system
* Low-cost hardware module

**3.2. System Requirement Specification**

## 3.2.1 Interface requirement

### 3.2.1.1 User Interface

User interface (UI) of mobile application must be design base on Flat UI Design. In this project, we only develops mobile application on Android. We ensure that the navigation options in the application will be similar; and all error occurring and exception handling will be catch and display for user with friendly messages.

## 3.2.1.2 Hardware interface

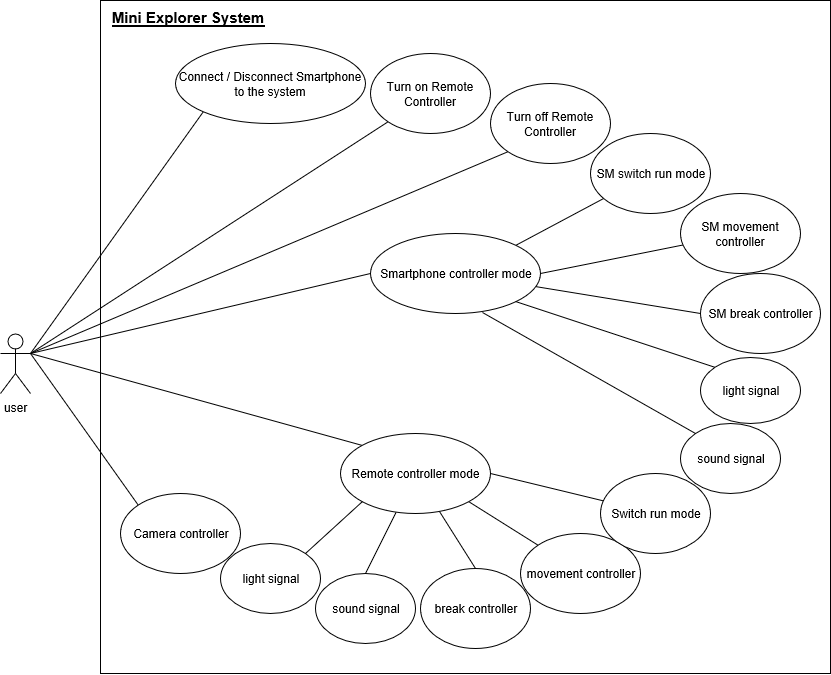
* **Smartphone device** : the application only runs on smart phone which support Android 4.4+ ; the component helps user to control the car and also display the “view” from the car’s camera
* **Arduino WEMOS** : Use to control the car in remote controller, the signal will be transfer to Raspberry PI through WIFI
* **Raspberry PI** : Receive control signals from remote controller(or Smartphone device) and control the operation motors/Servo; also transfer video data from camera to smartphone device

## 3.2.1.3 Software interface

<\*Missing software interface>

**3.3. System Features**

3.3.1 General use Case Diagram



## 3.3.2 Functional requirement

### 3.3.2.1 Connect smartphone

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 1 SPECIFICATION | | | |
| Use-case No. | UC001 | **Use-case version** | 1.0 |
| Use-case Name | Connect smartphone | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to connect to system though WIFI  Goal : Successfully connected to the system  Triggers :   * Connect to the car’s WIFI * Start Android Application * Tab “Connect device” button   Pre-condition : N/A  Post-condition :   * Android application is started successful * WIFI of car have turn on   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Connect to car’s WIFI** | | 2 | **User** | **Tab “Connect device” button** | | 3 | **System** | **Check if any device has connected**   * **Return true if has** * **Return false if has not** | | 4 | **System** | **Display connects message**   * **“Successful” if system check return “false”** * **“Failed to connect” if system check return “true”** | | 5 | **System** | **If connect fail, try again**  **If connect successful, continue …..** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.2 Disconnect Smartphone

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 2 SPECIFICATION | | | |
| Use-case No. | UC002 | **Use-case version** | 1.0 |
| Use-case Name | Disconnect smartphone | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to disconnect to system  Goal : Successfully disconnected to the system  Triggers :   * Tab “disconnect device” button   Pre-condition :   * Android application is connect successful with system   Post-condition :   * After disconnect user can connect again.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab “Disconnect device” button** | | 2 | **System** | **Display disconnects successful message** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.3 Turn on remote controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 3 SPECIFICATION | | | |
| Use-case No. | UC003 | **Use-case version** | 1.0 |
| Use-case Name | Turn on remote controller | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow User to turn on remote controller and connect it to WIFI system  Goal : Successful connect to the system  Triggers :  Pre-condition :   * The remote controller is turned off   Post-condition :   * The remote controller is turned on * The remote controller will auto connect to the system after turn on   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn on the remote controller** | | 2 | **System** | **Send a signal to car’s system** | | 3 | **System** |  |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.4 Turn off remote controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 4 SPECIFICATION | | | |
| Use-case No. | UC004 | **Use-case version** | 1.0 |
| Use-case Name | Turn off remote controller | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow User to turn off remote controller and also disconnect it to the system  Goal : Successful connect to the system  Triggers :  Pre-condition :   * The remote controller is turned on   Post-condition :   * The remote controller is turned off   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn off the remote controller** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.5 Enter smartphone controller mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 5 SPECIFICATION | | | |
| Use-case No. | UC005 | **Use-case version** | 1.0 |
| Use-case Name | smartphone controller mode | | |
|  | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to enter “Smartphone controller” mode  Goal : Successful enter the controller mode to control car’s movement  Triggers :   * Open the android application * Tab on “smartphone controller” button   Pre-condition :   * Android application is connect successful with system   Post-condition :   * Can only control the car by smartphone * User can turn back to enter another mode   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab “smartphone controller”** | | 2 | **System** | **Send a signal to car’s system** | | 3 | **System** | **Disable the “remote controller” if it has connected** | | 4 | **System** | * **If successful : Display the Screen controller** * **If failed : return and try again** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.6 Enter Remote controller mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 6 SPECIFICATION | | | |
| Use-case No. | UC006 | **Use-case version** | 1.0 |
| Use-case Name | remote controller mode | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to enter “remote controller” mode  Goal : Successful enter the controller mode to control car’s movement  Triggers :   * Open the android application * Tab on “remote controller” button   Pre-condition :   * Android application is connect successful with system * Remote controller is connect successful with system   Post-condition :   * Can only control the car by remote controller * User can turn back to enter another mode * Smartphone only use to display the “Camera view”   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab “remote controller”** | | 2 | **System** | **Send a signal to car’s system** | | 3 | **System** | **Disable the “smartphone controller”** | | 4 | **System** | **If successful : Display the “Camera view”**  **If connect failed : return and try again** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.7 RM Movement controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 7 SPECIFICATION | | | |
| Use-case No. | UC007 | **Use-case version** | 1.0 |
| Use-case Name | RM Movement controller | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to control car movement  Goal : Control movement of the car  Triggers :   * Turn the steering wheel to left or right that user want the car to turn in that direction. * Hold the “Pedal” button to increase the movement speed of the car   Pre-condition :   * Android application is connect successful with system * Enter “Smartphone controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn the steering wheel to left or right and also hold the “Pedal” button** | | 2 | **System** | **Send a signal to car’s system to control the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.8 RM Switch run mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 8 SPECIFICATION | | | |
| Use-case No. | UC008 | **Use-case version** | 1.0 |
| Use-case Name | RM Switch run mode | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to control car run mode (Run forward / backward)  Goal : Control run mode of the car  Triggers :   * Tab “Forward/Backward” button to switch between run forward and backward mode   Pre-condition :   * Android application is connect successful with system * Enter “Smartphone controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab “Forward/Backward” button** | | 2 | **System** | **Send a signal to car system to switch run mode**   * **“0” : run forward** * **“1” : run backward** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.9 RM break controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 9 SPECIFICATION | | | |
| Use-case No. | UC009 | **Use-case version** | 1.0 |
| Use-case Name | RM Break controller | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to decrease car movement  Goal : decrease movement of the car  Triggers :   * Hold “Break” button to decrease movement of the car   Pre-condition :   * Android application is connect successful with system * Enter “Smartphone controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Hold “Break” button** | | 2 | **System** | **Send a signal to car’s system to decrease the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.10 Movement controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 10 SPECIFICATION | | | |
| Use-case No. | UC010 | **Use-case version** | 1.0 |
| Use-case Name | Movement controller | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use Remote controller to control car movement  Goal : Control movement of the car  Triggers :   * Turn the steering wheel to left or right that user want the car to turn in that direction. * Hold the “Speed” pedal to increase the movement speed of the car   Pre-condition :   * Remote controller is connect successful with system * Enter “Remote Controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn the steering wheel to left or right and also hold the “Speed” pedal** | | 2 | **System** | **Send a signal to car’s system to control the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.11 Switch run mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 11 SPECIFICATION | | | |
| Use-case No. | UC011 | **Use-case version** | 1.0 |
| Use-case Name | Switch run mode | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use Remote controller to control car run mode (Run forward / backward)  Goal : Control run mode of the car  Triggers :   * Switch “Forward/Backward” lever to switch between run forward and backward mode   Pre-condition :   * Remote controller is connect successful with system * Enter “Remote controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Switch “Forward/Backward” lever** | | 2 | **System** | **Send a signal to car system to switch run mode**   * **“0” : run forward** * **“1” : run backward** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.12 Break controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 12 SPECIFICATION | | | |
| Use-case No. | UC012 | **Use-case version** | 1.0 |
| Use-case Name | Break controller | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use Remote controller to decrease car movement  Goal : decrease movement of the car  Triggers :   * Hold “Break” pedal to decrease movement of the car   Pre-condition :   * Remote controller is connect successful with system * Enter “Remote controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Hold “Break” pedal** | | 2 | **System** | **Send a signal to car’s system to decrease the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.13 Camera Controller

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 13 SPECIFICATION | | | |
| Use-case No. | UC013 | **Use-case version** | 1.0 |
| Use-case Name | Camera controller | | |
| Author | TuDN | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow User to control the camera ( turn left / right)  Goal : Successful control the camera module  Triggers :   * Turn the smart phone to left or right that user want the camera turn in that direction   Pre-condition :   * Smart phone is connect successful with system   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn the smart phone to left / right** | | 2 | **System** | **Send a signal to car system** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

## 3.2.3 Non-Functional Requirement

### 3.2.3.1 Reliability

* Mobile application and Remote Controller must work correctly with car system, have no conflict between devices
* Car’s angle, speed, run mode must be the same as the display on the mobile

### 3.2.3.2 Availability

* The mobile application and Remote controller are easy to connect to system to transform data.
* Power of components like Remote controller, Car System, Camera and smart phone is stable and available

### 3.2.3.3 Maintainability

* Microcontroller and all components can be replaced easily
* Implementation code must be follow coding standard, clearly commented for maintaining and enhancing system in the future

### 3.2.3.4 Performance

* System has to response one command in less than 100ms

**3.3. Infrastructure and Tools**

## 3.3.1 Hardware

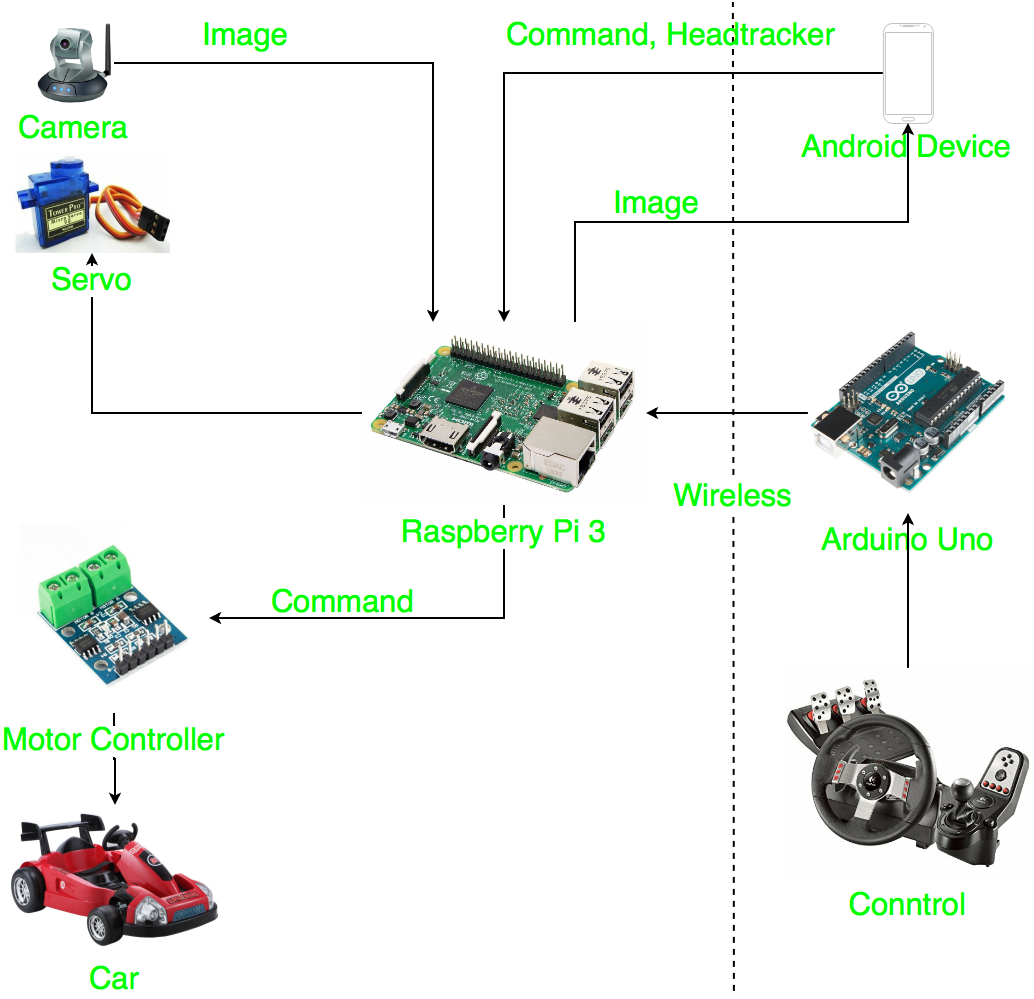
|  |  |  |
| --- | --- | --- |
| No | Item | Why we use it? |
| 1 | Raspberry Pi 3 |  |
| 2 | Arduino WEMOS D1 |  |
| 3 | Arduino UNO |  |
| 4 | DC motor |  |
| 5 | Servo MG996R |  |
| 6 | Rotary resistance |  |
| 7 | PS1/PS2 motor controller |  |
| 8 | Webcam Logitech C170 |  |
| 9 | Pin Cell Laptop |  |
| 10 | Frame mica |  |
| 11 | Module L9110s |  |
| 12 | Servo SG90 |  |

## 3.3.2 Software and tool

* Android Studio
* Python IDLE 3.5
* Arduino IDE
* Proteus 8
* Fritzing

# IMPLEMENTATION

## 4.1. Proposed system architecture

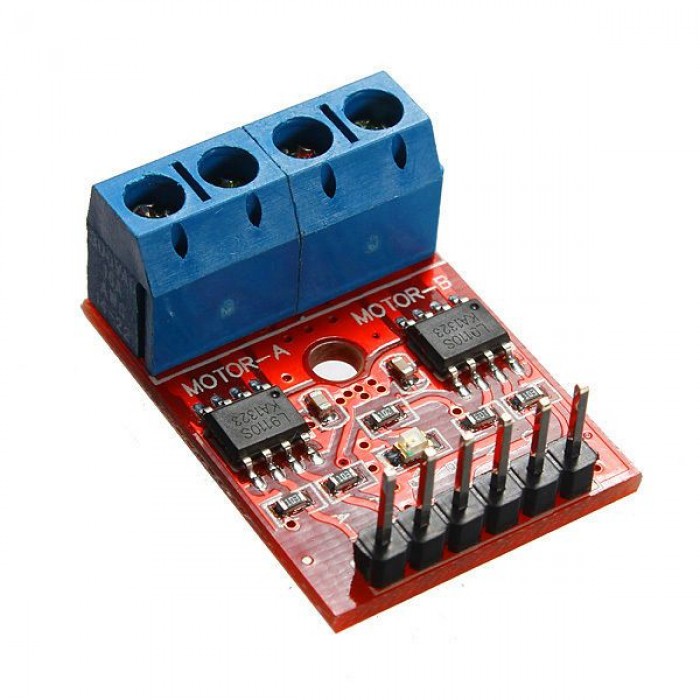


**Figure 4-1: System architecture**

## 4.2. Analysis and selection of tools, devices

### 4.2.1. Motor controller

We used 2 motor combined with L9110s module to control car. The module will control 2 motor with customize speed.



**Figure 4-2: Module L9110s Figure 4-3: Motor**

When the car go forward or backward, two motor will turned the same direction and speed.

**Figure 4-4: Run forward**

### 4.2.2. Servo car

We used servo (MG996R) to control the car. When the car turn left (or right), the servo will turn left (or right).



**Figure 4-5: Servo MG996R**

### 4.2.3. Servo camera

We used servo (SG90) to control the camera (Webcam Logitech C170). When driver turn Android device to left (or right), servo will turn in the same direction.

### 

**Figure 4-6: camera turn**

### 4.2.4. Remote Controller

### 

### 4.2.5. Supply power for Raspberry PI

## 4.3 Interface design

### 4.3.1. Hardware design

### 4.3.2. Software design

## 4.4. Testing