SUPERVISORS’ COMMENT

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# OPPONENTS’ COMMENT

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

ABSTRACT

**TÓM TẮT {font: TimeNew Roman, bolt, size: 14, căn lề: center**}

{Để 01 dòng trống}

Tên đề tài: ……………………………………………………………………………….

Sinh viên thực hiện: ………………………………….. …………………………………

Số thẻ SV: ……………………… Lớp: …………………………………………………

{Nội dung tóm tắt trình bày tối đa trong 1 trang} {Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

|  |  |
| --- | --- |
| THE UNIVERSITY OF DANANG  **UNIVERSITY OF SCIENCE AND TECHNOLOGY**  INFORMATION TECHNOLOGY FALCUTY | **SOCIALIST REPUBLIC OF VIET NAM**  Independence - Freedom - Happiness |

**GRADUATION PROJECT REQUIREMENTS**

Student Name: Thuy Tien Thi DANG Student ID: 102130180

Student Name: Phuong Thao Thi NGUYEN Student ID: 102130177

Class:13T4 Department: Information Technology

Major: Computer Networking And Communication

1. *Name of project:* Motorcycle Tracking System
2. *This project’s results:* ☐ *Are protected by an intellectual property agreement*
3. *Initial data:*

……………………………………..……………………………………………..……......……………………………………………………………………………………………..………………………………….…..………………………..………………………

1. *Project contents:*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Charts and drawings (please specify the drawing type and size):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Supervisor:* The Xuan Ly NGUYEN, PhD – Ho Thuy Tien TRAN, PhD.
2. *Date of assignment:*  *…./…/2017*
3. *Date of completion: …./…/2017*

|  |  |
| --- | --- |
|  | *Danang, 05/ 2017* |
| **Head of Division**…………………….. | **Supervisor** |

ACKNOWLEDGEMENTS

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified} ………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

………………………………………………………………………………..…………

Ghi chú: Sinh viên có thể trình bày “Lời cảm ơn” trong phần “Lời nói đầu”

GUARANTY

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

{Lời cam đoan của sinh viên thực hiện đồ án tốt nghiệp cam đoan về liêm chính học thuật}

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

TABLE OF CONTENTS

*{Để 2 dòng trống tại đây}*

*{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}{In trên 2 mặt giấy từ trang này đến hết phần “PHỤ LỤC”}*

Tóm tắt

Nhiệm vụ đồ án

Lời nói đầu và cảm ơn i

Lời cam đoan liêm chính học thuật ii

Mục lục iii

Danh sách các bảng biểu, hình vẽ và sơ đồ v

Danh sách các cụm từ viết tắt vi

Contents

[CHAPTER 1: THEORIES & TECHNOLOGIES 2](#_Toc513762261)

[1.1. Hardware 2](#_Toc513762262)

[1.1.1. Devices 2](#_Toc513762263)

[1.1.2. Arduino programming language 6](#_Toc513762264)

[1.2. Server 6](#_Toc513762265)

[1.2.1. Nodejs programming language 6](#_Toc513762266)

[1.2.2. RESTfull & API 6](#_Toc513762267)

[1.2.3. NoSQL & mongoDB 6](#_Toc513762268)

[1.2.4. Host Heroku & Database mLab 7](#_Toc513762269)

[1.3. Application 7](#_Toc513762270)

[1.3.1. iPhone Operating System 7](#_Toc513762271)

[1.3.2. Swift programming language 8](#_Toc513762272)

[CHAPTER 2: ANALYSIS & DESIGN 9](#_Toc513762273)

[2.1. Analyis 10](#_Toc513762274)

[2.1.1. Overview 10](#_Toc513762275)

[2.1.2. Feature 10](#_Toc513762276)

[2.2. Design 10](#_Toc513762277)

[2.2.1. Database 10](#_Toc513762278)

[2.2.2. Application mobi 10](#_Toc513762279)

[CHAPTER 3: DEPLOYMENT AND TESTING 10](#_Toc513762280)

[3.1. 10](#_Toc513762281)

**KẾT LUẬN** 68

**TÀI LIỆU THAM KHẢO** 70

**PHỤ LỤC**

LIST OF FIGURES AND TABLES

BẢNG 1.1 {size 13}..........................................................................................................

BẢNG 1.2 …….................................................................................................................

BẢNG 1.3 …….................................................................................................................

…….……..........................................................................................................................

HÌNH 1.1 ...........................................................................................................................

HÌNH 1.2 ...........................................................................................................................

HÌNH 1.3 ...........................................................................................................................

…….……..........................................................................................................................

***Ghi chú:***

* Mỗi bảng, hình vẽ/ sơ đồ phải được đánh số và có tên;
* Đánh số bảng và đánh số hình vẽ/ sơ đồ riêng. Quy luật đánh số như sau:
  + Chữ số thứ nhất chỉ tên chương;
  + Chữ số thứ hai chỉ thứ tự bảng biểu, sơ đồ, hình,…trong mỗi chương.

ABBREVIATION

KÝ HIỆU:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

CHỮ VIẾT TẮT:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

*Ghi chú:*

* Ký hiệu: mỗi mục ký hiệu gồm ký hiệu và phần tên gọi, diễn giải ký hiệu.
* Cụm từ viết viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa, thường được lặp nhiều lần trong đồ án.

OVERVIEW

**MỞ ĐẦU {font: TimeNew Roman, bolt, size: 14, căn lề: center**}

{Để 2 dòng trống}

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

{Trong phần này, cần trình bày về: Mục đích thực hiện đề tài, mục tiêu đề tài, phạm vi và đối tượng nghiên cứu, phương pháp nghiên cứu, cấu trúc của đồ án tốt nghiệp}

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

# THEORIES & TECHNOLOGIES

## Hardware

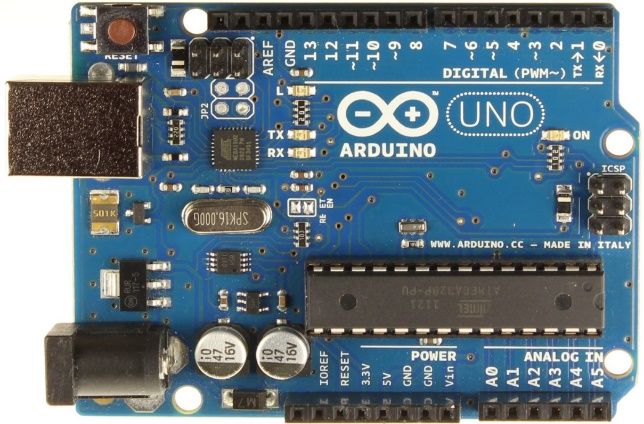
### Devices

* ***Arduino:*** Arduino is a microprocessor-based board that aims to build applications that interact with each other or with the environment. The hardware consists of an open-source board designed on an 8-bit AVR microprocessor platform, or 32-bit ARM Atmel. The current models are equipped with a USB interface, six analog input pins, and 14 digital I / O pins that are compatible with a variety of expansion boards.

*Specifications:*

|  |  |
| --- | --- |
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328) |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

****

* Module sim900A: This is an ultra compact and reliable wireless module. The SIM900A is a complete Dual-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications.Featuring an industry-standard interface, the SIM900A delivers GSM/GPRS 900/1800MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration of 24mmx24mmx3mm, SIM900A can fit in almost all the space requirements in user applications, especially for slim and compact demand of design.

*Specifications*:

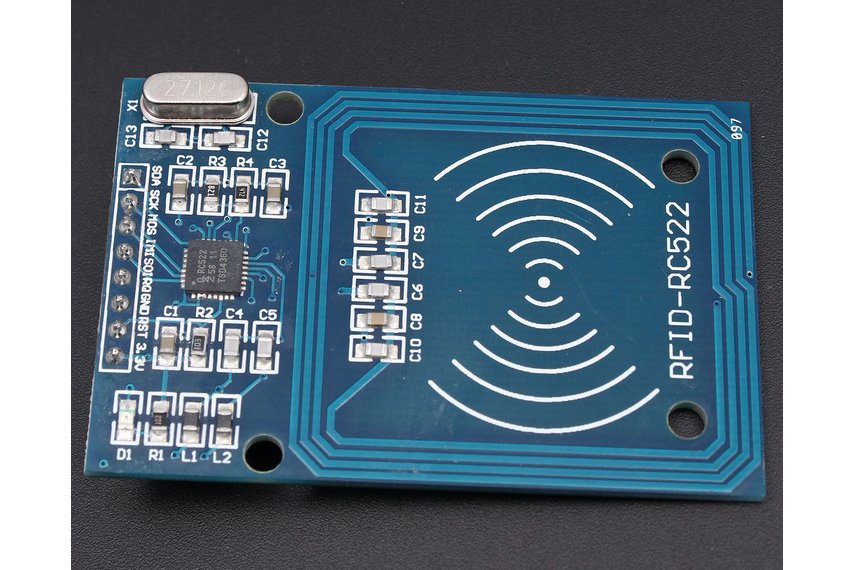
|  |  |
| --- | --- |
| Quad-Band | 850/ 900/ 1800/ 1900 MHz |
| Dual-Band | 900/ 1900 MHz |
| GPRS multi-slot class | 10/8GPRS mobile station class B |
| Compliant to GSM phase | 2/2+Class 4 (2 W @850/ 900 MHz) |
| Class 1 | 1 W @ 1800/1900MHz |
| Control via AT commands | GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands |
| Low power consumption | 1.5mA(sleep mode) |
| Operation temperature | -40°C to +85 °C |



* ***Module RFIP RC522:*** MF RC522 is a highly integrated read and write card chip applied to the 13.56MHz contactless communication. Launched by the NXP Company, it is a low-voltage, low-cost, and small-sized non-contact card chip, a best choice for intelligent instrument and portable handheld devices.   
  The MF RC522 uses advanced modulation and demodulation concept which fully presented in all types of 13.56MHz passive contactless communication methods and protocols. In addition, it supports rapid CRYPTO1 encryption algorithm to verify MIFARE products. MFRC522 also supports MIFARE series of high-speed non-contact communication, with a two-way data transmission rate of up to 424kbit/s. As a new member of the 13.56MHz highly integrated reader card series, MF RC522 is much similar to the existing MF RC500 and MF RC530 when there are also great differences. It communicates with the host machine via the serial manner which needs less wiring. You can choose between SPI, I2C and serial UART mode (similar to RS232), which helps reduce the connection, save PCB board space (smaller size), and reduce cost.

Specifications:

|  |  |
| --- | --- |
| Working current | 13—26mA/ DC 3.3V |
| Standby current | 10-13mA/DC 3.3V |
| sleeping current | <80uA |
| peak current | <30mA |
| Working frequency | 13.56MHz |
| Card reading distance | 0～60mm（mifare1 card） |
| Protocol | SPI |
| data communication speed | Maximum 10Mbit/s |
| Card types supported | mifare1 S50、mifare1 S70、mifare UltraLight、mifare Pro、mifare Desfire |
| Dimension | 40mm×60mm |
| Working temperature | -20—80 degree |
| Storage temperature | -40—85 degree |
| Humidity | relevant humidity 5%—95% |
| Max SPI speed | 10Mbit/s |



* **Module GPS NEO-6M:** The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. 0These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix0 (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches,enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

*Specifications*:

|  |  |
| --- | --- |
| Power Supply Range | 3 V to 5 V |
| Model | GY-GPS6MV2 |
| Ceramic | antenna |
| EEPROM | for saving the configuration data when powered off |
| LED | signal indicator |
| Antenna Size | 25 x 25 mm |
| Module Size | 25 x 35 mm |
| Mounting Hole Diameter | 3 mm |
| Default Baud Rate | 9600 bps |
| Dimension | **22mmX30mm** |
| Height | **13mm** |
| Weight | **12g** |



### Arduino programming :

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

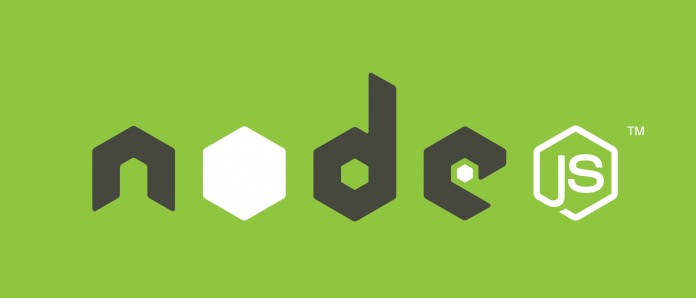
The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.



## Server

### Nodejs programming language

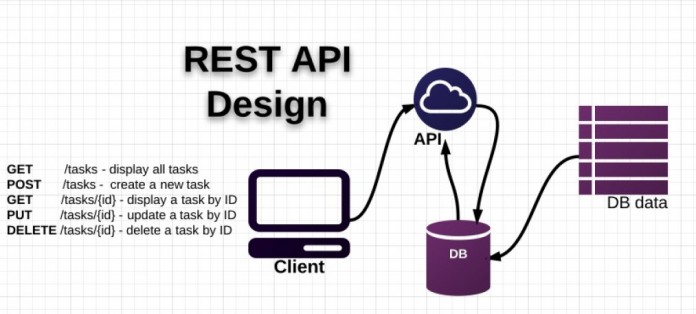
Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[5] unifying web application development around a single programming language, rather than different languages for server side and client side scripts.



### RESTfull & API

Representational State Transfer (REST) is an architectural style that defines a set of constraints and properties based on HTTP. Web Services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet. REST-compliant web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. Other kinds of web services, such as SOAP web services, expose their own arbitrary sets of operations.

In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables or remote calls. POSIX, Windows API and ASPI are examples of different forms of APIs. Documentation for the API is usually provided to facilitate usage and reimplementation.



### NoSQL & mongoDB

* **NoSQL:** A NoSQL (originally referring to "non SQL" or "non relational") database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Such databases have existed since the late 1960s, but did not obtain the "NoSQL" moniker until a surge of popularity in the early twenty-first century, triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com.NoSQL databases are increasingly used in big data and real-time webapplications.NoSQL systems are also sometimes called "Not only SQL" to emphasize that they may support SQL-like query languages.
* **MongoDB:**
* MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
* The document model maps to the objects in your application code, making data easy to work with
* Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data
* MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use
* MongoDB is free and open-source, published under the GNU Affero General Public License



### Host Heroku & Database mLab

* **Heroku:** Heroku is a cloud platform as a service (PaaS) supporting several programming languages that is used as a web application deployment model. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Rubyprogramming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it lets the developer build, run and scale applications in a similar manner across all the languages.



* **Database mLab:** mLab is a fully managed cloud database service that hosts MongoDB databases. mLab runs on cloud providers Amazon, Google, and Microsoft Azure, and has partnered with platform-as-a-service providers.



We use heroku to deploy the server to the internet, and use mLab to store database

## Application

### iPhone Operating System

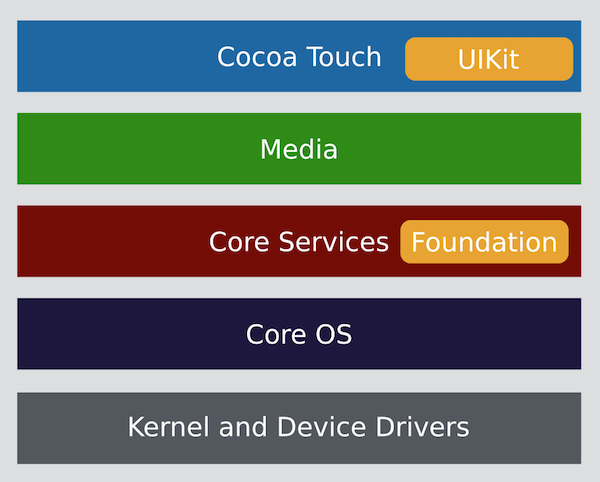
* ***Introduction***

iOS is the operating system that runs on iPad, iPhone, and iPod touch devices. The operating system manages the device hardware and provides the technologies required to implement native apps. The operating system also ships with various system apps, such as Phone, Mail, and Safari, that provide standard system services to the user.

* ***iOS layered architecture***

Architecture of iOs is a layered architecture. At the uppermost level iOS works as an intermediary between the underlying hardware and the apps you make. Apps do not communicate to the underlying hardware directly.

Apps talk with the hardware through a collection of well defined system interfaces. These interfaces make it simple to write apps that work constantly on devices having various hardware abilities.



* ***Cocoa Touch***: is high level set of APIs which developers seem to spend most of their time in. This is where a lot of the high level APIs that help developers get things done are located. There are a few key/famous APIs at this level that are really important, UIKit being an example of one. There are also mechanism exposed at this level to help developer do things like multi-tasking, laying out their application, designing the application to be responsive on multiple screen sizes, handling complex user interacts through gesture recognition, capturing application state and interacting with other applications.
* ***Media***

The Media Layer is were a lot of the more complex graphics/rendering/audio technologies relate APIs reside. If you’ve even had a passing interest in Apple announcements you might of heard of one API within this layer which is called “Metal”. Metal is supposed to be a fairly low level graphics API suitable for use by games engine developers and people who make graphically intensive applications.

Other pieces in the media layer include ones centered on audio & video, to allow you to interact with the camera/video recorder and microphones of an iOS device. As far as the browser is concerned a lot of the APIs that expose the same type of functionality come from the “UI Backend” layer. You might have a high level binding in JavaScript to allow you to access the web cam (WebRTC for instance).

* ***Core Services***

Core Services is as low as most applications get. This is where you find one of the most fundamental iOS APIs, the “Foundation” API. This is a related piece of the architecture called Core Foundation which is a C based API and collection of data types which developer rarely need to go near. Foundation on the other hand is an Objective-C (and Swift) based API.

* ***Core OS***

The Core OS is the deep dark bowels of iOS. Specifically though, this is where you would go to get fine grain control over data from the Bluetooth radio, various security related services, the Network Extension framework (for working with VPNs) and a variety of little System APIs. There is also a very interesting set of APIs for “discrete signal processing” (DSP) and other complex mathematical operations. If you wanted to get really heavy into Instragram filters this would be the place to go.

https://intellipaat.com/tutorial/ios-tutorial/ios-architecture/

### Swift programming language

* **Introduction**

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns.

The goal of the Swift project is to create the best available language for uses ranging from systems programming, to mobile and desktop apps, scaling up to cloud services. Most importantly, Swift is designed to make writing and maintaining correct programs easier for the developer. To achieve this goal, we believe that the most obvious way to write Swift code must also be: safe, fast, and expressive.

On December 3, 2015, the Swift language, supporting libraries, debugger, and package manager were published under the Apache 2.0 license with a Runtime Library Exception. Swift is now free to be ported across a wide range of platforms, devices, and use cases.

* **Architectural overview**

The features of Swift are designed to work together to create a language that is powerful, yet fun to use. Some additional features of Swift include:

* Closures unified with function pointers.
* Tuples and multiple return values.
* Generics.
* Fast and concise iteration over a range or collection.
* Structs that support methods, extensions, and protocols.
* Functional programming patterns, e.g., map and filter.
* Powerful error handling built-in.
* Advanced control flow with do, guard, defer, and repeat keywords.

The Swift language is managed as a collection of projects, each with its own repositories. The current list of projects includes:

* The Swift compiler command line tool.
* The standard library bundled as part of the language.
* Core libraries that provide higher-level functionality.
* The LLDB debugger which includes the Swift REPL.
* The Swift package manager for distributing and building Swift source code.
* Xcode playground support to enable playgrounds in Xcode. [[[1]](#endnote-1)]

### Open Source & CocoaPods

Open-source software (OSS) is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. Open-source software may be developed in a collaborative public manner. According to scientists who studied it, open-source software is a prominent example of open collaboration.

CocoaPods manages library dependencies for Xcode projects. The dependencies for your projects are specified in a single text file called a Podfile. CocoaPods will resolve dependencies between libraries, fetch the resulting source code, then link it together in an Xcode workspace to build your project. Ultimately the goal is to improve discoverability of, and engagement in, third party open-source libraries by creating a more centralized ecosystem.

CocoaPods supports almost every way to get source code: git, svn, bzr, http and hg. It is acceptable to use own private code repository to manage dependencies. It only requires a git repo, no server necessary.

# ANALYSIS & DESIGN

## Analyis

### Introduction

A device will be mounted on the user's vehicle, which will take the values of the coordinates, velocity… of the vehicle and send them to the designated server. Via the mobile application, users can track the location of the vehicle, view the journey history and set parameters for vehicle maintenance.



### Feature

* Detect user’s motor coordinates, speed, altitude

When user move, the system detect current coordinates, current altitude, current speed. The system automatically send data received to the server.

* Send message to user’s phone (sms) / app

When user’s motorcycle is affected by unauthorized use. The system automatically send the message via SMS to user’s phone and show notification on the app.

* Maintain accessories of uses’s motor

On the application, user can detect

## Design

### Database

We use NoSQL to design the database. With two collections: user, device.

Collection User:

*{*

username: {

type: String,

required: true,

unique: true,

},

password: { type: String },

nameDisplay: { type: String, default: '' },

phone: { type: String, default: '' },

email: {

type: String,

required: true,

unique: true,

},

avatarUrl: { type: String, default: '' },

gender: {

type: [{

type: String,

enum: ['male', 'female', 'other']

}]

},

auth: { type: int, default: 0 },

createdDate: {

type: Date,

default: Date.now

},

city: { type: String, default: '' },

bio: { type: String, default: '' },

addressDetail: { type: String, default: '' },

myDevice: {

type: [

{

deviceId: String,

},

],

},

}

Collection Device:

{

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', default: null },

serialNumber: { type: Number, required: true, unique: true, },

stealStatus: { type: Boolean, default: false },

totalDistance: { type: Number, default: 0 },

vehicle: {

type: [

{

nameVehicle: { type: String, default: '' },

typeVehicle: { type: String, default: '' },

capacity: { type: String, default: '' },

}

],

default: [],

},

vehicleStatus: {

type: [

{

lat: { type: Number, default: 0 },

log: { type: Number, default: 0 },

alt: { type: Number, default: 0 },

courseDeg: { type: Number, default: 0 },

speed: { type: Number, default: 0 },

timeStamp: { type: Number, default: 0 },

status: { type: Number, default: 0 },

address: { type: String, default: '' },

}

],

default: [],

},

accessories: {

type: [

{

nameAccessory: {

type: [{

type: String,

enum: ['Oil', 'Air Filter', 'Bugi', 'Chain']

}],

default: '',

},

distanceToChange: { type: Number, default: 0 },

timesChange: { type: Number, default: 0 },

currentDistance: { type: Number, default: 0 },

updated: { type: Date, default: Date.now },

}

]

},

is\_delete: { type: Boolean, default: false },

}

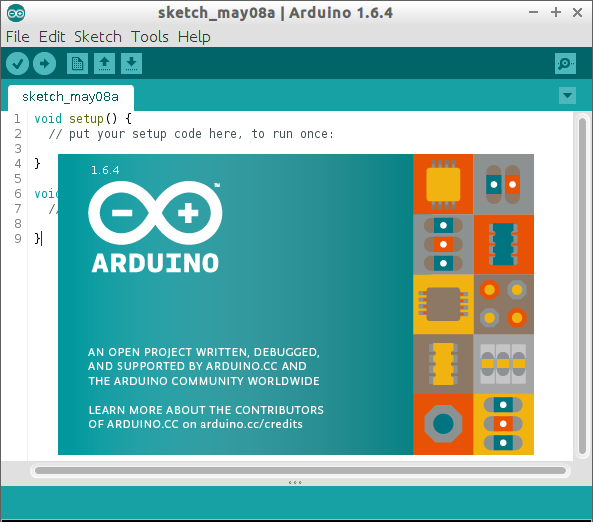
### Application

# DEPLOYMENT

## Tools

### Arduino software, Visual Studio Code & Xcode

Arduino is an open-source electronics platform based on easy-to-use hardware and software.



Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.



Xcode is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [macOS](https://en.wikipedia.org/wiki/MacOS) containing a suite of [software development](https://en.wikipedia.org/wiki/Software_development) tools developed by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) for developing software for macOS, [iOS](https://en.wikipedia.org/wiki/IOS), [watchOS](https://en.wikipedia.org/wiki/WatchOS), and [tvOS](https://en.wikipedia.org/wiki/TvOS). First released in 2003, the latest stable release is version 9.3 and is available via the [Mac App Store](https://en.wikipedia.org/wiki/Mac_App_Store) free of charge for [macOS High Sierra](https://en.wikipedia.org/wiki/MacOS_High_Sierra) and [macOS Sierra](https://en.wikipedia.org/wiki/MacOS_Sierra) users. [Registered developers](https://en.wikipedia.org/wiki/Mac_Developer_Program) can download [preview releases](https://en.wikipedia.org/wiki/Software_release_life_cycle) and prior versions of the suite through the [Apple Developer](https://en.wikipedia.org/wiki/Apple_Developer) website.



With hardware, we used the arduino software to code.

With server, we used the Visual Studio Code software to code.

With client (application), we used the Xcode version 9.3 software to code.

### Git & bitbucket

Git is a [version control system](https://en.wikipedia.org/wiki/Version_control_system) for tracking changes in [computer files](https://en.wikipedia.org/wiki/Computer_file) and coordinating work on those files among multiple people. It is primarily used for [source code management](https://en.wikipedia.org/wiki/Source_code_management) in [software development](https://en.wikipedia.org/wiki/Software_development), but it can be used to keep track of changes in any set of files. As a [distributed revision control](https://en.wikipedia.org/wiki/Distributed_revision_control) system it is aimed at speed,data integrity, and support for distributed, non-linear workflows. . (https://git-scm.com)

Bitbucket is a web-based [version control](https://en.wikipedia.org/wiki/Version_control) [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) [hosting service](https://en.wikipedia.org/wiki/Shared_web_hosting_service) owned by [Atlassian](https://en.wikipedia.org/wiki/Atlassian), for [source code](https://en.wikipedia.org/wiki/Source_code) and development projects that use either [Mercurial](https://en.wikipedia.org/wiki/Mercurial) (since launch) or [Git](https://en.wikipedia.org/wiki/Git_(software)) (since October 2011) [revision control](https://en.wikipedia.org/wiki/Revision_control) systems. (<https://bitbucket.org>)



### Postman

API testing is a type of [software testing](https://en.wikipedia.org/wiki/Software_testing) that involves testing [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) directly and as part of [integration testing](https://en.wikipedia.org/wiki/Integration_testing) to determine if they meet expectations for functionality, reliability, performance, and security. Since APIs lack a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface), API testing is performed at the [message layer](https://en.wikipedia.org/wiki/Communications_protocol#Layering). API testing is now considered critical for automating testing because APIs now serve as the primary interface to [application logic](https://en.wikipedia.org/wiki/Application_logic) and because [GUI tests](https://en.wikipedia.org/wiki/Graphical_user_interface_testing) are difficult to maintain with the short release cycles and frequent changes commonly used with [Agile software development](https://en.wikipedia.org/wiki/Agile_software_development) and [DevOps](https://en.wikipedia.org/wiki/DevOps)

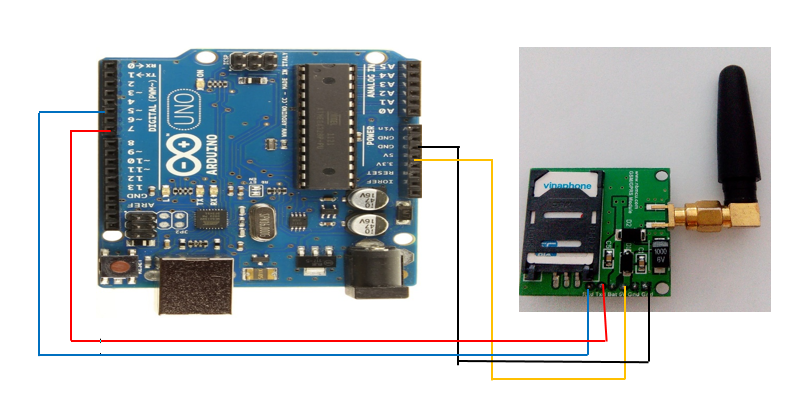
Postman is a software which helped us test API.



## Deployment

### Hardware

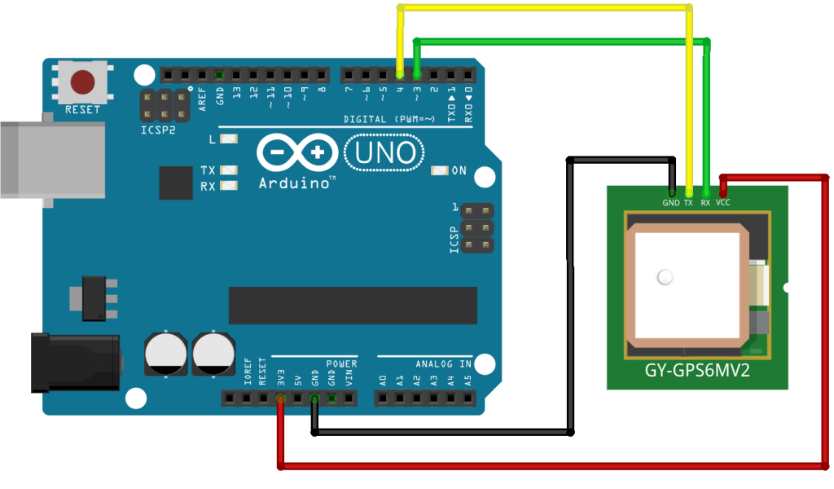
* Connect **Arduino Uno** with **SIM900A**



* + Purpose:
    - Send SMS: Send sms to your phone when your motor may be stolen and when start system
    - Send data to server: Take coordinates from GPS and send it to our server https://smart-moto.herokuapp.com/
  + How to connect

|  |  |
| --- | --- |
| **SIM900A Module** | **Arduino** |
| RX | 5 |
| TX | 7 |
| VCC | 3.3V |
| GND | GND |

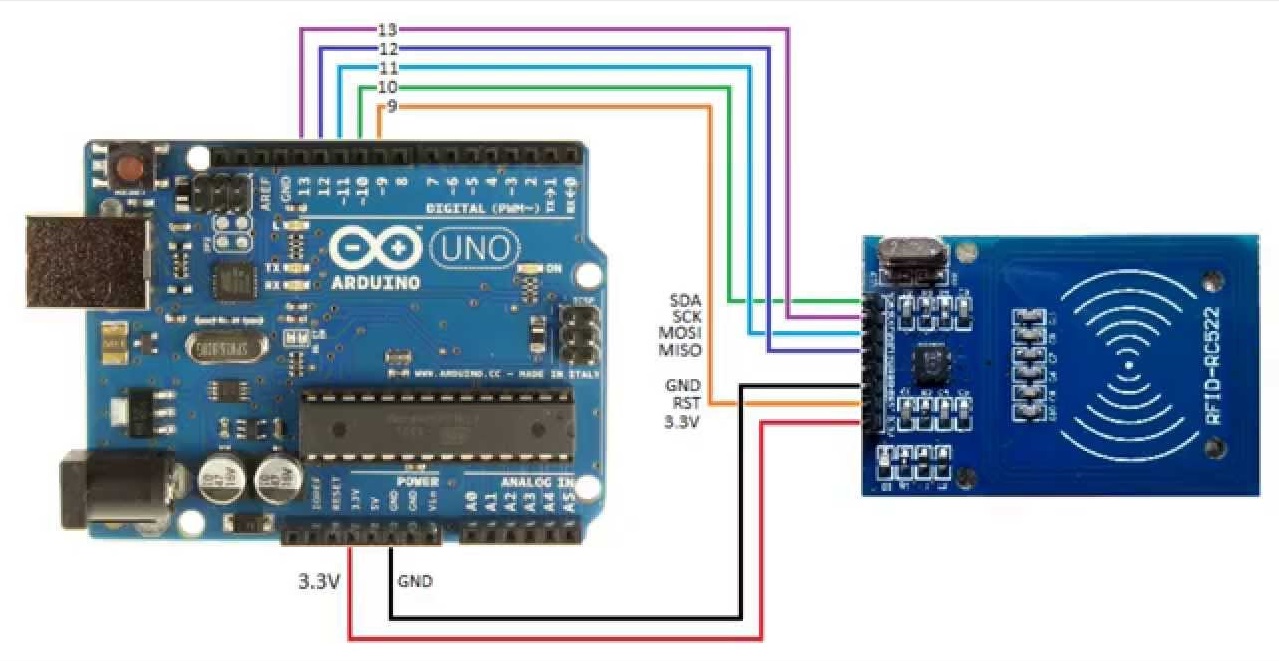
* Connect **Arduino Uno** with **GPS**

****

* Purpose
  + Get coordinates, speed, altitude of your device
* How to connect

|  |  |
| --- | --- |
| **GPS Module** | **Arduino** |
| RX | 4 |
| TX | 3 |
| VCC | 3.3V |
| GND | GND |

* Connect **Arduino Uno** with **RFID**

****

* Purpose
  + Scan your idCard to verify th correct owner
* How to connect

|  |  |
| --- | --- |
| **Module RFID** | **Arduino** |
| RST | 9 |
| SDA | 10 |
| MOSI | 11 |
| MISO | 12 |
| SCK | 13 |
| VCC | 3.3V |
| GND | GND |

* Deploy code to hardward
  + Connect your hardward with arduino software via USB port.
  + Choose correct port to connect with arduino uno.
  + Open hardward code which is attached with this report
  + Click upload button to upload code to your arduino uno

### Server

Precondition:

* Nodejs

Extract source code which is attached with this report to computer

Open Terminal, go to the folder and install dependencies by typing this command:

npm install ⏎

npm start ⏎

### Application

Extract source code which is attached with this report to computer.

Open Terminal, go to the folder and install dependencies by typing this command:

* $ pod install ⏎

After installing dependencies, open .xcworkspace file in Xcode.

Connect your device to the laptop and start building the app and run by hitting shortcut ⌘ + R or select menu Product → Run.

The app will be installed on the device but it will be blocked, unlock by go to Phone Settings → General → Device management → Applications and Developers, then trust the installed app.

Build again and the app runs with no doubt.

## Demonstration of main features

**Ghi chú về trình bày mỗi trang của đồ án**

* Page layout: cỡ giấy A4; lề trái: 3cm, lề phải: 2cm, lề trên: 2,5cm, lề dưới: 2,5cm; header và footer: from edge: 1,6cm;
* Đánh số trang: bắt đầu đánh số trang từ phần “MỞ ĐẦU”;
* Tiêu đề chương, mục, tiểu mục: xem chi tiết như minh họa ở trên;
* Mục Header: Tên đề tài (định dạng: font Time New Roman, Italic, size 10, căn lề: giữa);
* Mục Footer: Sinh viên thực hiện, giảng viên hướng dẫn, đánh số trang (định dạng: font Time New Roman, size 10);
* Chú dẫn bảng: nằm trên bảng, đánh số theo chương và số lũy tiến theo số thứ tự của bảng trong chương;
* Chú dẫn hình: nằm dưới hình, đánh số theo chương và số lũy tiến theo số thứ tự của hình trong chương;
* Đánh số công thức: bên phải công thức, đánh số theo chương và số lũy tiến theo số thứ tự của công thức trong chương;
* Nên sử dụng các chức năng về Bookmark, Caption, Cross-Reference, Format Heading,… của Microsoft Word hoặc các phần mềm soạn thảo tương tự; cần tổ chức theo dạng “Long Document”.

**KẾT LUẬN {size 14**}

{Để 2 dòng trống}

Nội dung kết luận {Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

**Ghi chú về phần Kết luận**

* Phần Kết luận cần phải nêu được những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất và kiến nghị (nếu có);
* Trong phần này, có thể định dạng các điểm/ mục kết luận theo dạng Outline hoặc Numbering hoặc Bullets.

**TÀI LIỆU THAM KHẢO**

{bold, size 14}

{Để 2 dòng trống}

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

**Ghi chú:**

Sinh viên xem “Quy định về liêm chính học thuật” ban hành kèm theo Quyết định số 29/QĐ-ĐHBK ngày 09/01/2017 và “Hướng dẫn trích dẫn và lập Danh mục tài liệu tham khảo” được ban hành theo văn bản số 30/HD-ĐHBK ngày 09/01/2017 để thực hiện trích dẫn và lập Danh mục tài liệu tham khảo.

**PHỤ LỤC 1**

{bold, size 14}

{Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}

**PHỤ LỤC 2**

{bold, size 14}

{Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}

1. [↑](#endnote-ref-1)