A project report on

Mobile Detection With CupCarbon

Submitted in partial fulfillment of the requirements

For the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

By

S. Farhana	**[164G1A0525]**
P. Harshitha	**[164G1A0529]**
M. Mallika	**[164G1A0550]**
C. Jaya Prakash	**[174G5A0502]**

Under the Guidance of

Mr. T. Venkata Naga Jayudu M.Tech., (Ph.D)
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPUR
(Accredited by NAAC with 'A' Grade, Affiliated to JNTUA, Approved by AICTE, New Delhi)

2019-2020

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTHAPURAMU

(Accredited by NAAC with 'A' Grade, Affiliated to JNTUA, Approved by AICTE, New Delhi)

Rotarypuram Village, B.K. Samudram Mandal, Ananthapuramu – 515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(B. Tech Program accredited by NBA)

SRIT
Empowering Knowledge

Certificate

This is to certify that the project report entitled Mobile Detection with Cup Carbon is the bonafide work carried out by **SHAIK FARHANA, P.HARSHITHA, M.MALLIKA, C.JAYAPRAKASH** bearing Roll Number **164G1A0525, 164G1A0529, 164G1A0550, 174G5A0502** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2019-2020.

Signature of the Guide	**Head of the Department**
T .Venkata Naga Jayudu M.Tech.,(Ph.D)	Dr. G K Venkata Narasimha Reddy M.Tech., Ph.D
Assistant Professor	Professor and HOD

Date:

Place: Rotarypuram

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our effort with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our guide **Mr. T. Venkata Naga Jayudu, Assistant Professor, Computer Science and Engineering**, who has guided us a lot and encouraged us in every step of the Project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We are very much thankful to **Dr. G K Venkata Narasimha Reddy, Professor, Computer Science and Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. T Hitendra Sarma, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

we also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey my gratitude to our family who fostered all the requirements and facilities that we need.

FARHANA SHAIK (164G1A0525)
P. HARSHITHA (164G1A0529)
M. MALLIKA (164G1A0529)
C. JAYAPRAKASH (174G5A0502)

ABSTRACT

CupCarbon is a Smart City and Internet of things Wireless Sensor Network (SCI-WSN) simulator. Its objective is to design, visualize, debug and validate distributed algorithms for monitoring, environmental data collection, etc., it can also be used to create environmental scenarios such as fires, gas, mobiles, and generally within educational and scientific projects. It can help to visually explain the basic concepts of sensor networks and how they work; it may also support scientists to test their wireless topologies, protocols, etc. The Mobile detection with CupCarbon specifies the detection of Mobile devices. CupCarbon provides a real environment with a map in that wireless nodes are placed and perform the detection. Senscript is used to generate the code for detection and this code is implemented in CupCarbon console. After Detection of mobiles the message can be sent to the sink. The proper simulation has been conducted and the results shows that efficient mobile detection done with minimum energy consumption.

LIST OF CONTENTS

List of Figures

List of Tables

CHAPTER – 1

INTRODUCTION

In recent years, there has been increasing issues relating to the use of mobile phones in restricted areas. The mobile phone provides many ways for a student to cheat in an examination hall. The mobile phones are strictly prohibited inside the examination rooms, Temples etc.,one of the existing approaches is to ensure the people are free of mobile phones in examination hall or restricted area is by manual inspection in the entrance. Manual inspection cannot fully reveal the people having mobile phones all the time.

Let us take a scenario of Examination hall. The Simulation will ensure the connectivity between a student sitting inside the hall and outsiders have been considerably increased a burden to invigilators to ensure that malpractices are not committed during exams. A student may constantly communicate with other students outside the examination hall via Email and text messages. They can exchange information such as question and answer through WhatsApp, Email attachments etc. By using a mobile camera a student can take the snap shot of the question paper and send to other students for help. Sometimes there are more possibilities for leaking the question papers. Nowadays mobile has internet connectivity so that a student can post questions in online and gets quick response and in addition they can search for answers in searchengines.

There are many scenarios related restricted use of mobile. One of the solutions to this problem is Detecting Mobile by using CupCarbon. This ensures to detect the mobile whenever the mobile presence is detected or sensed by the sensor nodes. Usually the mobile detection can be done by using hardware like mobile detectors which contain some of the drawbacks include efficiency and maintenance possibly serve the process of mobile detection so that we are using sensor nodes to detect themobile.

As a good solution, wireless sensor networks (WSNs) represent a flexible, low cost and highly efficient technology that can be used for mobile detection. A wireless sensor network is usually composed of a few sinks and a large quantity of small sensor nodes, which are able to sense, process and communicate data [4].

ProjectOverview

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. Wireless sensor networks (WSN) are composed of a finite set of sensor devices geographically distributed in a given indoor or outdoor environment (usually predefined). A WSN aims to gather environmental data and the node devices placement may be known or unknown a priori.Some application fields include tracking, monitoring, surveillance, building automation, military applications, and agriculture, amongothers.

Cup Carbon is a smart city and IoT wireless sensor network (WSN) simulator. It is a new platform for 2D/3D design, visualization and the simulation of radio propagation and interferences in IoT networks. Cup Carbon offers two simulation environments. It first enables the design of mobility scenarios and the generation of natural events such as fires and gas as well as the simulation of mobiles such as vehicles and flying objects. The second simulation environment represents a discrete event simulation of wireless sensor networks which takes into account the scenario designed on the basis of the first environment.Applications are Smart city, Environmental Protection and sustainable resources, Smart home, Smartwater.

The main scope of project is to sense the presence of mobile phones to prevent the use of same in the Restricted Areas. This system can detect any cell phone. In the above mentioned conditions a sensor node is placed in the network. Whenever a mobile phone passes through the WSN sensor nodes these sensor nodes detect the mobile and send the signal to the sink.

ProblemStatement

In our day to day life the usage of mobile phones has been increased. In restricted area such as exam venues, Temples etc., this system will be used to detect the mobile phones present in the area by the radio frequency signals which are transmitted by them by using the sensors that are placed in the respected areas in CupCarbon wireless sensor network.

Objectives

The main objectives of our project are

- Detection of mobile through networkmodel.
- Reduces Energyconsumption.

CHAPTER – 2

LITERATURE SURVEY

2.1 Related Work

Numerous literatures pertaining about Wireless Sensor Network have been published already and are available for public usage. A comprehensive understanding of Wireless Sensor Networks and CupCarbon can be helpful for us to solve the problem of Mobile Detection.

Mobile detection in WSN involves monitoring the behavior of Sensors and mobile devices in order to estimate, detect, or avoid undesirable behavior of sensors. The main aspect is to focus more on Real world implementation and Energy consumption. Let us know about the Cupcarbon software. Cupcarbon is a simulator specially designed for WSN networks. Its objective is the design, visualization, and validation of algorithms for environmental monitoring, data collection, etc. It allows creating scenarios with environmental stimulus as fires, gases, and mobile objects within scientific and educational projects. It offers two simulation environments. On the one hand, it enables the design of scenarios with mobility and generation of natural events and on the other, the simulation of discrete events in WSNs. The network can be designed thanks to its intuitive graphical interface making use of the OpenStreetMap (OSM) where the sensors can be placed directly. In addition, each sensor can be individually configured by its command-line making use of a language specially designed for CupCarbon calledSenScript.

The main Parts of Cupcarbon are shown in below figure:

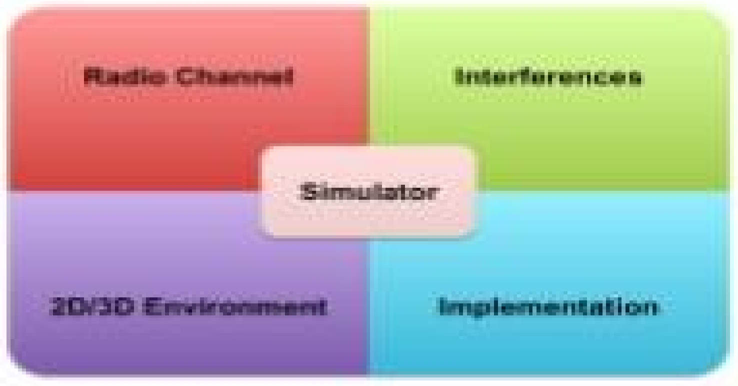


Figure 2.1: The main Parts of cup carbon

CupCarbon does not implement all protocol layers. For this reason, it is said that its true functionality is to complement the existing simulators to observe the consumption of natural resources and the effect of buildings and streets in urban networks. The energy consumption can be calculated and graphically displayed as a function of the simulation time. In addition, it allows observing the visibility of propagation and interference models. CupCarbon is able to simulate the ZigBee, LoRa and Wi-Fi protocols. On the other hand, although CupCarbon does not stand out for having a wide variety of protocols integrated the fact that it has been implemented using Java permits the creation of different algorithms.

Within the selection section of the radio parameters, we can choose between ZigBee, Wi-Fi and Lora. By default, the IEEE 802.15.4 (ZigBee) standard is automatically set for each new added sensor. One of the most interesting competences of this tool is the possibility of deploying the network on different types of maps on which Google Maps stands out. In this way, a more realistic simulation can be achieved. The following devices can be displayed on the different maps [20]:

- ➢ Sensor Nodes: nodes in charge of taking the variables of themedium.
- ➢ Media Sensor Nodes: sensor capable of taking variables from the environment and its sensing unit is directional. It has a form of a cone that can be modified with the SenScript
- ➢ Base Station (Sink): It is like a sensor node with the peculiarity that its battery is infinite. • Mobile: the mobile device can be driven through a route created withmarkers.
- ➢ Markers: these markers can be used for different tasks suchas:
  - Adding sensors can be done randomly delimiting the area on which you want to deploy.
  - Creatingroutes.
  - Adding or drawingbuildings.

Moreover, by extending the Solver section of the menu, we can use different algorithms such as the Jarvis algorithm that, given a cloud of points, surrounds them with lines in the positive direction.

The parameters used in this Cupcarbon are shown below table:

Table 2.1: Parameters used in this Cup carbon

Parameters	Simulation Parameters
	Values
Initial Energy	0.5J
Eelec	50nJ/bit/m2
Eamp	100pJ/bit/m2
Packet size	2000 bits
Signaling packet size	64 bits

Most of the existing simulators (as mentioned in the previous section) are mainly used to develop routing protocols. However, they do not offer real-time interference and radio propagation modeling for smart cities and IoT applications. The main idea behind proposing CupCarbon is to keep simulation time short while taking into account a realistic evaluation of the wireless interference in a 3D environment with an accurately simulated radio channel. The CupCarbon platform is developed with the following primaryobjectives:

- To study the deployment of WSN by considering the mobility and the availability of the radio spectrum.
- To simulate and analyze the performance of a proposed WSN in a 2D/3Denvironment.
- To study the feasibility and reliability of communication in thenetwork.
- To detect high radio interference zones in thenetwork.
- To simulate the radio propagation accurately in a real-time urbanenvironment.
- To better visualize the simulation results to debug and validate a developedalgorithm.

Cupcarbon is a WSN simulator specially designed for Smart Cities and Internet of Things applications. Its objective is to design, visualize, and validate the proposed algorithms for environmental monitoring, data collection, etc. It offers following two simulation environments. The first simulation environment allows the design of mobility scenarios and the generation of natural events such as fires as well as the simulation of mobiles like vehicles and flying objects (e.g. UAVs, insects, etc.).The second environment describes a discrete event simulation of WSN which considers the scenario designed on the basis of the first environment.

Any WSN can be prototyped with its intuitive graphical interface using OpenStreetMap (OSM) framework. The sensor nodes can be directly placed on the map. In addition, each sensor node can be individually configured by its command-line with the script called SenScript. From SenScript, it is also possible to generate codes for hardware platforms such as Arduino/XBee. In CupCarbon, the energy consumption of the nodes can be calculated and graphically displayed as a function of the simulation time.

Moreover, it enables the visibility and impact of wireless interference and signal propagation. CupCarbon can simulate the PHY layers based on ZigBee, LoRa and Wi-Fi protocols (Which is also discussed in this document). CupCarbon offers the simulation of scenarios and algorithms on several levels. For example, on the first level, it will determine the nodes of interest in the given area on the map, followed by a level to analyze the nature of the communication between these nodes to accomplish a given task (e.g., detection of an event). On the final level, describing the nature of the data routing to the base station if an event is detected. CupCarbon also allows configuring the nodes dynamically so that the nodes can be split into separate networks or merge in differentnetworks.

CupCarbon represents the main kernel of the ANR project PERSEPTEUR that aims to develop algorithms for an accurate simulation of the propagation and interference of signals in a 3D urbanenvironment.

SenScript is the script used to program sensor nodes of the CupCarbon simulator. It is a script where variables are not declared and without types, but they can be initialized (set command). For string variables, it is not necessary to use the quotes. A variable is used by its name and its value is determined by $. It is possible to use the instruction function to add complex and additional functions programmed in Java (in a source code modeonly).

Example:

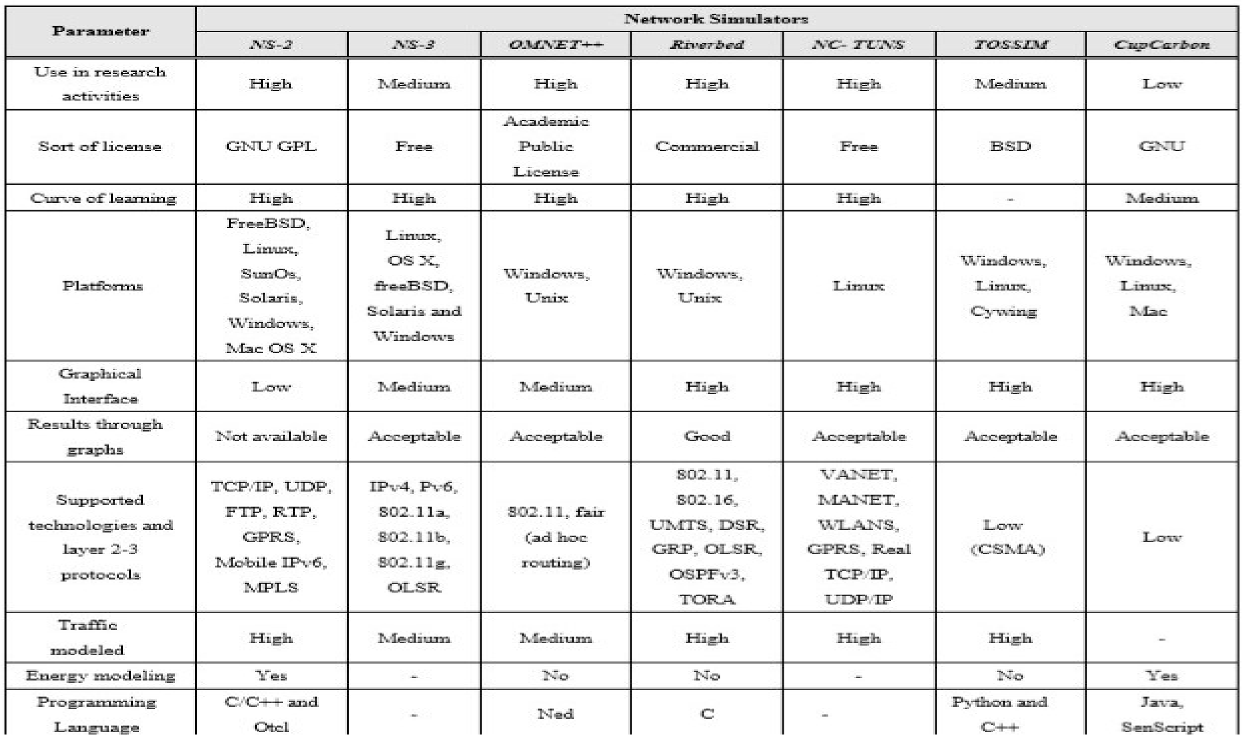
SenScript: setxabcd	≈ Java: String x = "abcd";
SenScript: set yx	≈ Java: String y = "x";
SenScript: set y$x	≈ Java: String y =x;

2.2 Existing System

A little number of literatures pertaining about Cupcarbon usage have been published already and are available for public usage.

A case study on all the existing wireless sensor network by Cristina López-Pavón, Sandra Sendra1, Juan F. Valenzuela-Valdés which describes the difference between all the WSN like NS-2,omnet++ etc., and also a case study regarding usage of Cupcarbon simulation tool.[1]

Table 2.2: Comparison of network simulators



Parameter	Network Simulators						
	NS-2	*NS-3*	*OMNET++*	*Riverbed*	*NC- TUNS*	*TOSSIM*	*CupCarbon*
Use in research activities	High	Medium	High	High	High	Medium	Low
Sort of license	GNU GPL	Free	Academic Public License	Commercial	Free	BSD	GNU
Curve of learning	High	High	High	High	High	-	Medium
Platforms	FreeBSD, Linux, SunOs, Solaris, Windows, Mac OS X	Linux, OS X, freeBSD, Solaris and Windows	Windows, Unix	Windows, Unix	Linux	Windows, Linux, Cywing	Windows, Linux, Mac
Graphical Interface	Low	Medium	Medium	High	High	High	High
Results through graphs	Not available	Acceptable	Acceptable	Good	Acceptable	Acceptable	Acceptable
Supported technologies and layer 2-3 protocols	TCP/IP, UDP, FTP, RTP, GPRS, Mobile IPv6, MPLS	IPv4, Pv6, 802.11a, 802.11b, 802.11g, OLSR	802.11, fair (ad hoc routing)	802.11, 802.16, UMTS, DSR, GRP, OLSR, OSPFv3, TORA	VANET, MANET, WLANS, GPRS, Real TCP/IP, UDP/IP	Low (CSMA)	Low
Traffic modeled	High	Medium	Medium	High	High	High	-
Energy modeling	Yes	-	No	No	-	No	Yes
Programming Language	C/C++ and Otcl	-	Ned	C	-	Python and C++	Java, SenScript

Application of mobile detection by using IOT presented by K.Parvateesam, G.A. ArunKumar

The main scope of application is to sense the presence of an activated mobile phone and camera from a distance of one and-a-half meters to prevent the use of same in the examination halls[2].

Introduction:

- The main scope of project is to sense the presence of an activated mobile phone and camera from a distance of one and-a-half meters to prevent the use of same in the examinationhalls.

- This system can detect any active cell phone i.e. when someone is trying to make a call or receive a call, sending a message or receiving amessage

Existing Model:

- The radio frequency signals are transmitted from wireless camera and mobile phone during the video transmission, incoming call and outgoing call, text messages from one gadget toanother.
- The detector will detect the transmitted signal and then it is gives as input to Arduinomicrocontroller.
- As soon as the Arduino microcontroller receives the signal, it will turn ON the beep alarm and the information will be displayed on the LCD display.
- This system will be used to detect the mobile phones and the wireless hidden camera present in a room by the radio frequency signals which are transmitted bythem.

Proposed Model:

- This Model detects RF signal which are emitted from the mobile phones and the hidden wireless camera during incoming and outgoing calls, messages and direct video transmission from one device to another with the alertsystem.
- This will continuously monitors the RF level with in a room. If the RF signal level increases a warning alert through buzzer is implemented along with message is displayed through the LCD display as RF signal is detected or DeviceDetected.

Limitations:

- Detects the device but don't show the location of thatdevice
- Don't use any graphicalinterface.
- More EnergyConsumption.

Application of emergency detection by using cupcarbon by Massinissa Saoudi, Ahc`ene Bounceur, Reinhardt Euler, Tahar Kechadi, Alfredo Cuzzocrea .A new approach for early forest fire detection, which is based on the integration of Data Mining techniques into sensor nodes. The idea is to partition the node set into clusters so that each node can individually detect fires using classification techniques. The approach is validated using the CupCarbon simulator. [3]

Introduction:

- Environmental emergencies are presented as natural events and also human-induced accidents, that may cause severe environmental damage augment air pollution, engender climate changes and, most important, loss of humanlives.

- Where environmental emergencies cannot be stopped, their early detection is necessarilyimportant.

Existing Model:

- Wireless sensor networks (WSNs) represent a flexible, low cost and highly efficient technology that can be used for forest fire detection.
- A wireless sensor network is usually composed of a few sinks and a large quantity of small sensor nodes, which are able to sense, process and communicatedata
- In this Model, the fire detection is done on a cluster-head level. However, we suppose that every node in the WSN contains all the requiredfunctions.
- In this way, a communication overhead between neighboring nodes is avoided and each sensor node can detect fire locally by itself. This allows to reduce the energy consumption and to improve the performance of theWSN.

ProposedModel:

- Our work is based on measuring and combining real data from different sensors (temperature, humidity, light and smoke) and using the Artificial Neural Network classifier applied to data for fire detection.
- A node detects fire locally by itself, then it discards normal values and transmits only abnormal values to the sink for fire localization and to inform thefirefighters.

Limitations:

- Use of Artificial Neural Network creates an unexplained behavior of thenetwork.
- Less efficient data mining techniques.

The user guide to use the Cupcarbon simulation tool by Umber Noreen. The main objective of this platform is to design and simulate Wireless Sensor Networks dedicated to Smart-city and IoT applications [4].It allows to validate distributed algorithms in a 2D/3Denvironment, taking account of the city buildings in which to deploy the network, the mobiles, and using accurate models of radio propagation and interferences in that environment.

2.3 ProposedSystem

The aim of the proposed system is Mobile detection system overcome most of the drawbacks that are defined above.Here it detect the mobiles in the network, it can extend range, it can detect the moving mobiles, it uses graphical interface for real world purpose.CupCarbon is provided with map in that map create nodes and perform the detection. After Detection of mobiles the message has to be sent to the sink. And the overall Simulation is done in Graphical User Interface.The deployment of WSN for mobile detection should consume energy very efficiently because the replacement of batteries may be too costly, impractical or even not possible. The energy consumption should also be balanced among nodes in order to maximize the life-time of WSN.

CHAPTER-3

FEASIBILITY STUDY

Feasibility analysis is an assessment of the practicality of a proposed plan or method. And it reduces the development risks. The major areas are considered in feasibility analysis are as follows.

The feasibility study concerns with the considerations made to verify whether the system is to fit to be developed in all terms. Once an idea to develop software is put forward the question that arises first will pertain to the feasibility aspects. It involves developing and understanding of the selectedprogram.

This documentation presents the results of an independent study of the feasibility of completing the Mobile detection. It enhances the probability of success by addressing and mitigating factors early on that could affect the project. There are different aspects in the feasibility study.

- TechnicalFeasibility
- OperationalFeasibility
- SocialFeasibility

3.1 Technical Feasibility

The technical capability of the personnel as well as the capability of the available technology should be considered. The technical aspects we have included are suitable for the modern environment and the technological tools which we have used to train and run our model are quite preferable.

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance,costs etc.A number of issues have to beconsidered while doing a technicalanalysis.

Understand the different technologies involved in the proposed system before commencing the project, we have to be clear about the technologies that are to be required for the development of thesystem.

- This project involves Senscript as a programming language and CupCarbon as a platform environment for analyzing and manipulating the algorithms and for easy running of the senscript code.
- For implementing the user interface, we already installed the java platform as the Cupcarbon is based on the JavaPlatform

3.2 Operational Feasibility

Proposed projects are beneficial only if they can be turned into information systems that will meet the organization operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed.

It determines if the human resources are available to operate once it has been executed. It focuses on the complexity of the problem and check if the given solution will solve the problem. Our solution is to install the sensors wherever we want once the sensors are activated then it sends the data to the Destination without any delay for this no human is specially required to check the mobilepresence.

- Since the proposed system was to help reduce the maintenance of the system once the project is loaded then it detects the mobile.
- Here another advantage is that the rate replacement of sensor nodes is also decreased as the energy consumption is decreased and life timeincreases.

3.3 SocialFeasibility

Social feasibility is one of the feasibility studies where the level of acceptance of the people is considered regarding theproject.

This involves the feasibility of the proposed solution to generate economic and social benefits. As we are dealing with the mobile detection in restricted areas this solution will help more benefits since we are trying to detect the mobiles in restricted areas by providing the models.

- It describes the effect on users from the usage of mobiles are not allowed in restricted areas and it will be very useful to control the mobileusage.
- It describes how you propose to ensure user co-operation with the authorities to follow their rules.

CHAPTER – 4

REQUIREMENT SPECIFICATIONS

Software Requirement Specification is the starting point of the software developing activity. As system yow more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software is initiated by the client needs. The SRS is the means of translating the ideas of the minds of the clients (the input) into a formal document (the output of the requirement phase). The SRS phase consists of two basic activities.

The process is order and more nebulous of two, deals with understand the problem, the goal and constraints. Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The requirements phase terminates with the production of validate SRS document. Producing the SRS document is the basic goal of this phase. The software Requirements Specification begins the translation process that converts the software requirements into the language the developers will use. The SRS draws on the use-cases from the User Requirement Document and analyses the situations, and omissions before development progresses significantly under mistaken assumptions. To run our project on a specified computer, that must configure the following hardware and software requirements.

4.1 HardwareRequirements

- Processor: Intel i3 andabove
- Ram :4GB
- Hard Disk :1TB

4.2 SoftwareRequirements

- Operating Systems – Windows, Linux,Mac
- Platform – Cupcarbonv-4.2
- Programming in –Senscript

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

CHAPTER - 5

TECHNOLOGY

Technologies Used:

- Java
- CupCarbonSimulator

Java InstallationSteps:

Java Development Kit (JDK) allows you to code and run Java. Java Development Kit (JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But it's recommended that you install only latest version.

The Java Development Kit (JDK) can be downloaded

fromhttp://oracle.com/technetwork/java/javase/downloads/index.html

The Eclipse IDE can be downloaded fromhttp://www.eclipse.org/downloads/.

The Android platform SDK starter package can be downloaded from

http://developer.android.com/sdk/index.html.

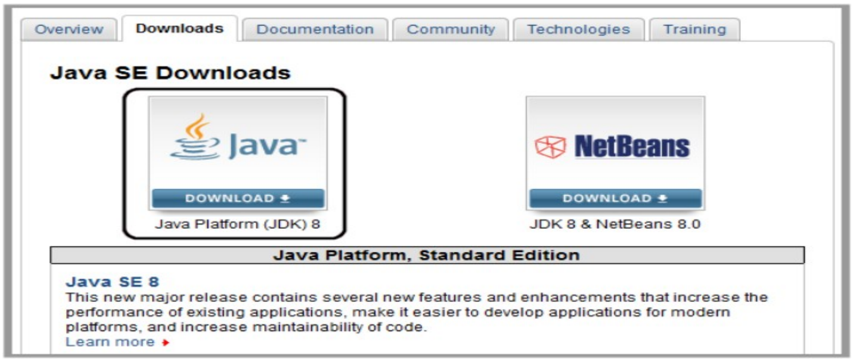
The Android Development Tools (ADT) Plug-in can be downloaded from http://developer.android.com/sdk/eclipse-adt.html. The plug-in contains project templates and Eclipse tools that help in creating and managing Android projects.

The Android SDK is not a full development environment and includes only the core SDK tools, which are used to download the rest of the SDK components. This means that after installing the Android SDK Tools, you need to install Android platform tools and the other components required for developing Android applications.

After downloading Android SDK Tools package, double click it to initiate the installation process. The Android SDK Manager window opens. The dialog boxes you see from now are form windows installer, and the screens may vary from other operating system installers.

Downloading the JDK:

To download the Java Development Kit (JDK), launch your web browser (e.g. Internet Explorer) and go to this



Java SE Downloads

Java Platform (JDK) 8

JDK 8 & NetBeans 8.0

Java Platform, Standard Edition

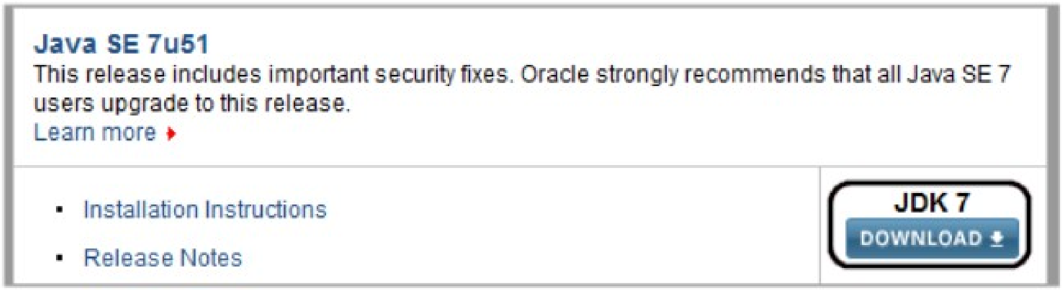
Java SE 8
This new major release contains several new features and enhancements that increase the performance of existing applications, make it easier to develop applications for modern platforms, and increase maintainability of code.
Learn more

Figure 5.1: Java Development Kit

Web address:

http://www.oracle.com/technetwork/java/javase/downloads/index.html. This page shows many download options. The top of the page shows the most current JDK download options (e.g. JDK 8 or JDK 9).

You want to download the "Java Platform (JDK)", so click on the download button circled above. Note that more recent versions of Java may be available by the time you begin your course. It is OK to install the latest Java version for your course work. You may, however, choose to install an earlier version (e.g. Java 7 or Java 8) by scrolling down on this same page.



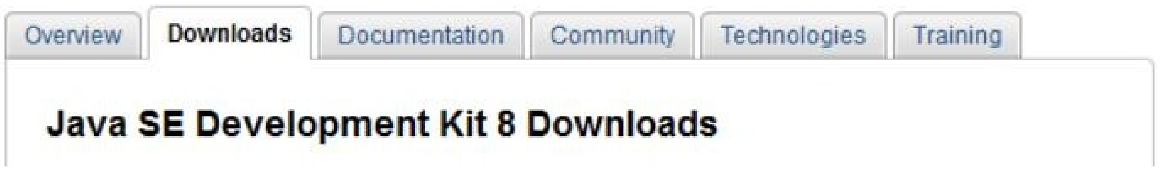
Java SE 7u51
This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
Learn more

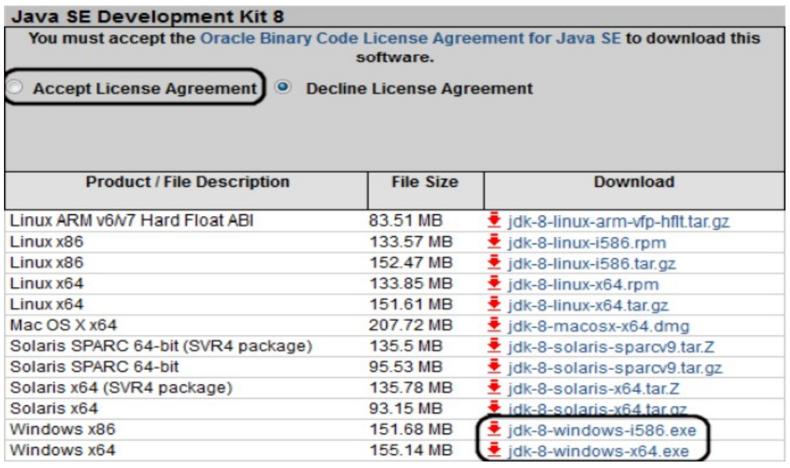
- Installation Instructions
- Release Notes

Figure 5.2: Java Platform (JDK)

After clicking on the Download button for the Java Platform, you will arrive at the download page.

Scroll down to the first download block which contains individual links for each operating system. You need to click the "Accept License Agreement" radio button and then click on the link for your particular operating system. Choose the "Windows x86" line for 32-bit Windows or the "Windows x64" line for 64- bit Windows. Your exact filenames may vary depending on the actual JDK version.

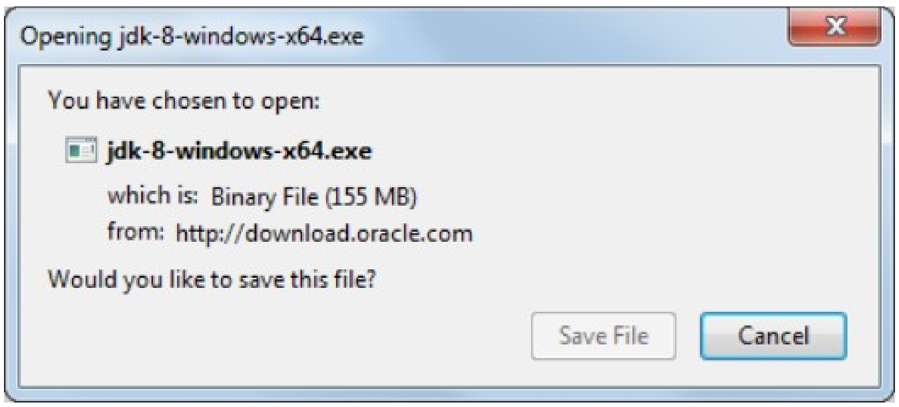



Java SE Development Kit 8

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	83.51 MB	jdk-8-linux-arm-vfp-hflt.tar.gz
Linux x86	133.57 MB	jdk-8-linux-i586.rpm
Linux x86	152.47 MB	jdk-8-linux-i586.tar.gz
Linux x64	133.85 MB	jdk-8-linux-x64.rpm
Linux x64	151.61 MB	jdk-8-linux-x64.tar.gz
Mac OS X x64	207.72 MB	jdk-8-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	135.5 MB	jdk-8-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	95.53 MB	jdk-8-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	135.78 MB	jdk-8-solaris-x64.tar.Z
Solaris x64	93.15 MB	jdk-8-solaris-x64.tar.gz
Windows x86	151.68 MB	jdk-8-windows-i586.exe
Windows x64	155.14 MB	jdk-8-windows-x64.exe


Opening jdk-8-windows-x64.exe

You have chosen to open:

jdk-8-windows-x64.exe

which is: Binary File (155 MB)

from: http://download.oracle.com

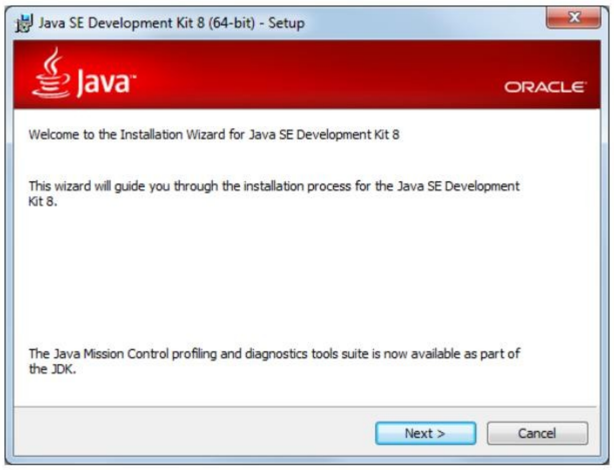
Would you like to save this file?

Figure 5.3: jdk-8-windows-x64.exe

As soon as you click to download the file, a pop-up window will appear with options to “Save”, “Save File”, or similar phrasing. The look of this window will depend on the version of Windows and the Internet browser you are using to download the file. The following screenshot is from Mozilla Firefox.

Select “Save File” or “Save” to save the file to a location on your local hard drive. You can save it to your Desktop or some other file folder. Remember this location so you can find it later! Some browsers may have a very different download experience, so follow the windows for your browser. Oracle updates the exact version of the JDK frequently. Our examples show JDK version 8.0, but keep in mind that the version available to you at the time of download will be different as Oracle releases new patches. Once the file is saved, use your Windows Explorer to find and run the program by double-clicking in it. Depending on your version of Windows and security settings you may get a security popup; click on “Run” or “OK” tocontinue.

When setup is launched you should see the following screen: This is the first screen in the install process. Click “Next” to continue.



Welcome to the Installation Wizard for Java SE Development Kit 8

This wizard will guide you through the installation process for the Java SE Development Kit 8.

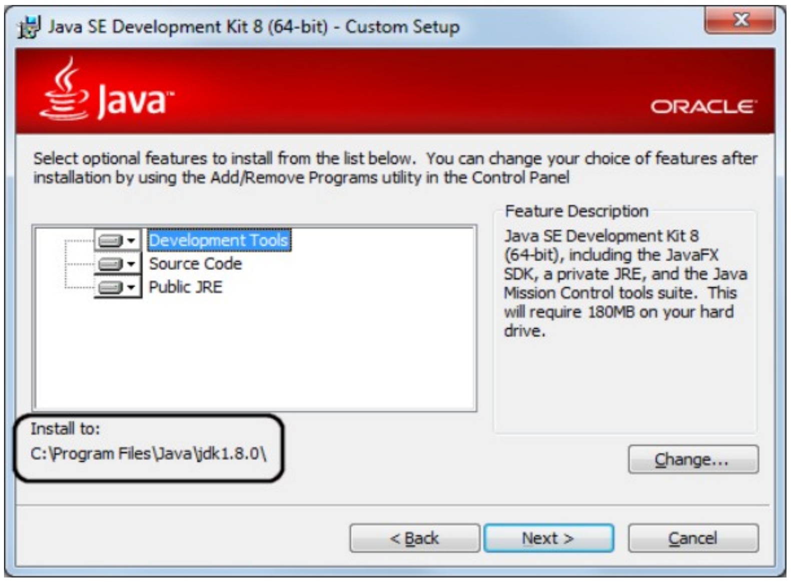
The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.

Figure 5.4: Java SE Development Kit 8(64-bit)-setup

This next screen lists all of the possible JDK options that can be installed. Since we will be covering the basics of Java in this course, you can just accept the defaults and simply click on the “Next” button to continue. There is no need to make any changes on this screen.

Write down the “Install to:” location as you will need this path for a later activity!

The next screen will display a simple progress bar while the JDK files are being installed. This process could take anywhere from seconds to minutes, depending on the speed of your computer.



Java SE Development Kit 8 (64-bit) - Custom Setup

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

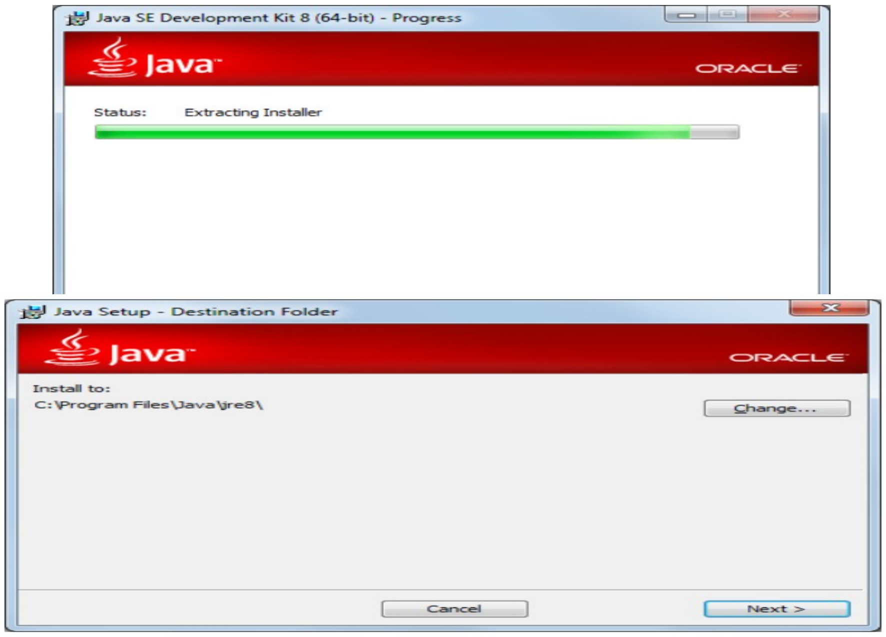
Development Tools
Source Code
Public JRE

Feature Description
Java SE Development Kit 8 (64-bit), including the JavaFX SDK, a private JRE, and the Java Mission Control tools suite. This will require 180MB on your hard drive.

Install to:
C:\Program Files\Java\jdk1.8.0\

Figure 5.5: Java SE Development Kit 8(64-bit)-custom Setup

When the JDK is finished installing, the installation program will install the JRE (Java Runtime Environment) files. The screen above will allow you to choose the directory where the JRE will be located. We recommend



Java SE Development Kit 8 (64-bit) - Progress

Status: Extracting Installer

Java Setup - Destination Folder

Install to:
C:\Program Files\Java\jre8\

Figure 5.6: Java Setup-Destination Folder

That you allow the files to install in the default directory, as shown below.

Once you choose the “Next” button, the installation will display another progress bar. This will show the progress of the installation of the JRE files.

The next screen will simply show the progress of your JRE installation. In this first step, the installation program will automatically download additional files from the Oracle website. This is a large program andwill take some time!

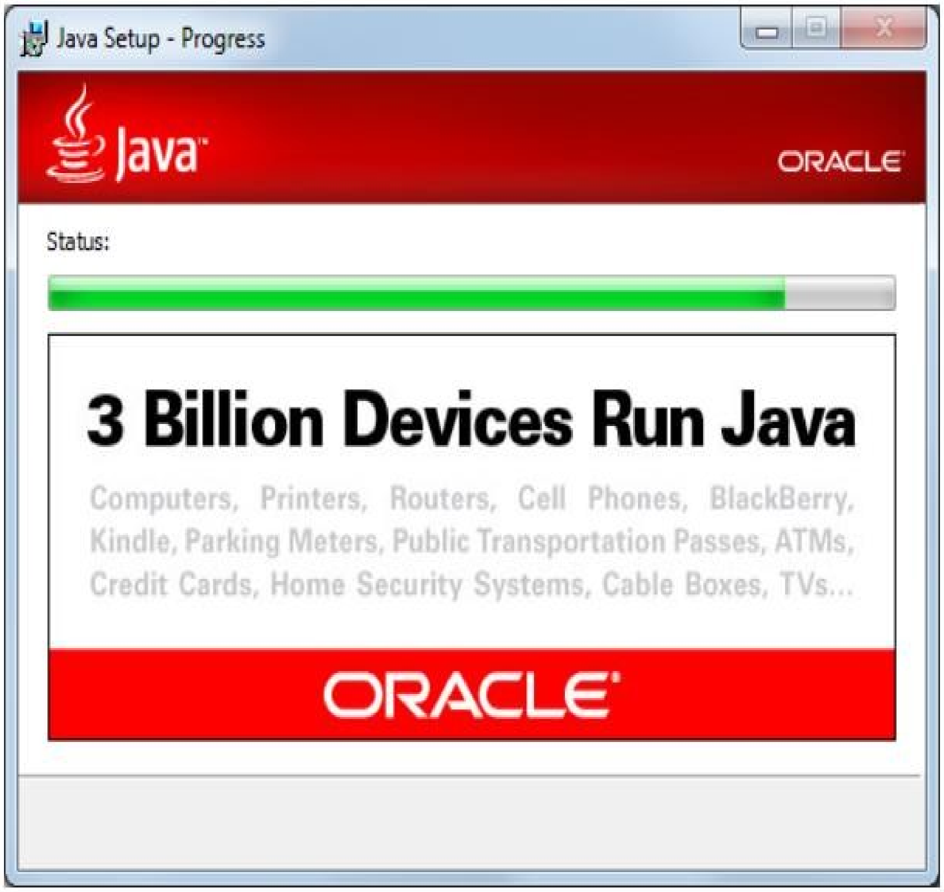
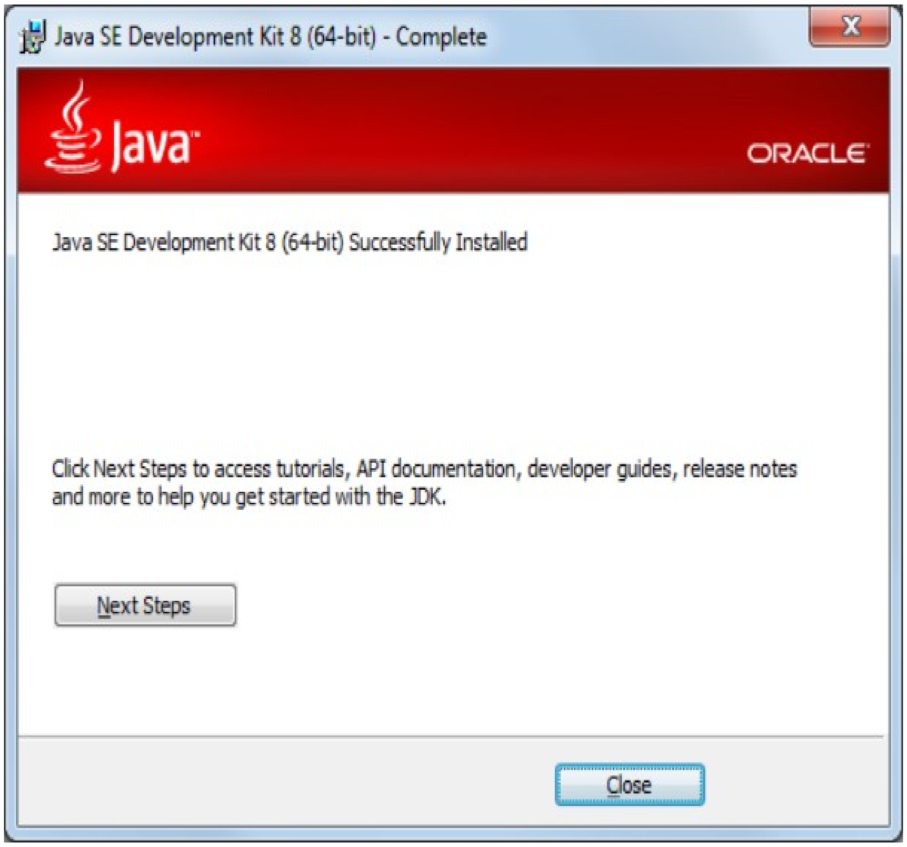

Java Setup - Progress
Java
ORACLE
Status:
3 Billion Devices Run Java
Computers, Printers, Routers, Cell Phones, BlackBerry, Kindle, Parking Meters, Public Transportation Passes, ATMs, Credit Cards, Home Security Systems, Cable Boxes, TVs...
ORACLE

Figure 5.7: Java Setup-progress

At this point, the installation of the JDK files is complete. Click on the “Close” button on this screen, to finish the setup.



Java SE Development Kit 8 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Figure 5.8:Java SE Development Kit8 (64-bit)-Complete

Registration of the JDK with Oracle is not required. If you are prompted for registration of any kind, you can cancel out of or close those windows. Congratulations! You have finished the installation of the JDK and JRE in your Windows computer.

CupCarbonInstallation:

Step 1: Open Google chrome

Step 2: Enter the link www.cupcarbon.com

Step 3: click on "CupCarbon - A Smart City & IoT WSN Simulator – Google Chrome Beta (pane)"

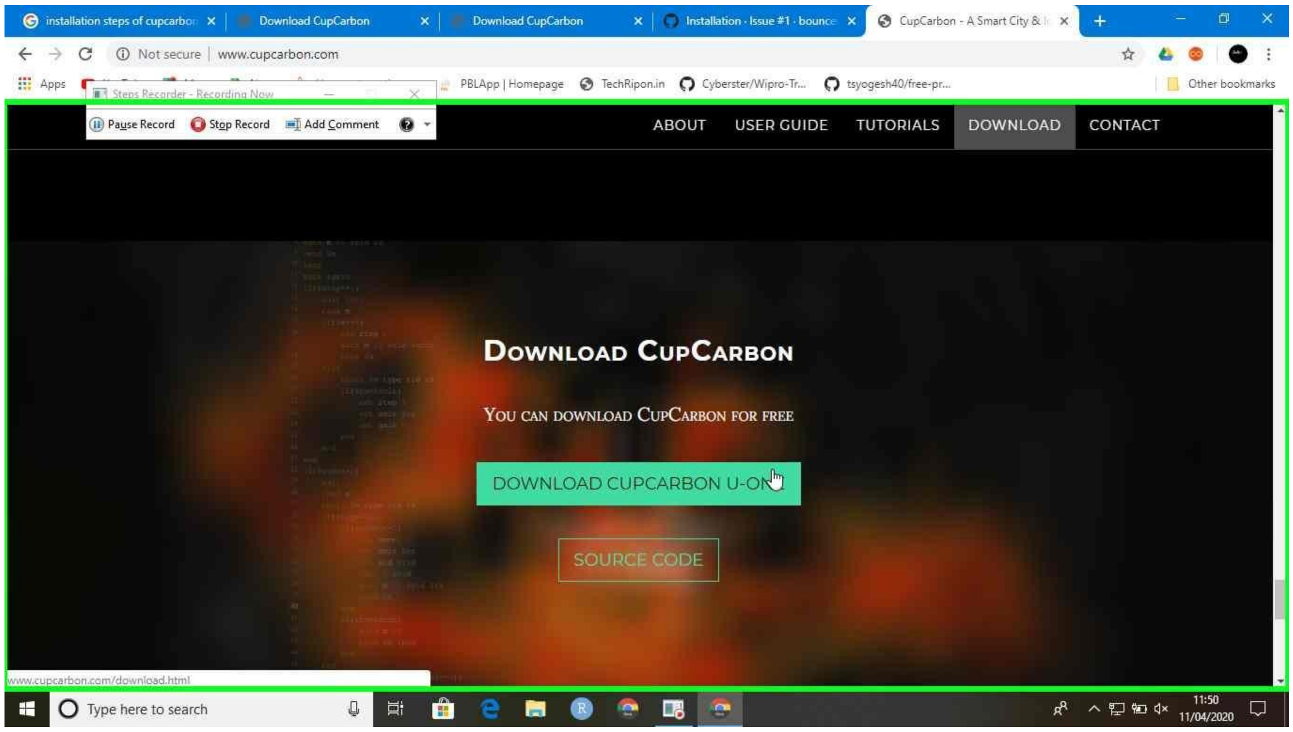


Figure 5.9:Download cup carbon

Step 5: Click on the download we will be directed into another tab as shown below.

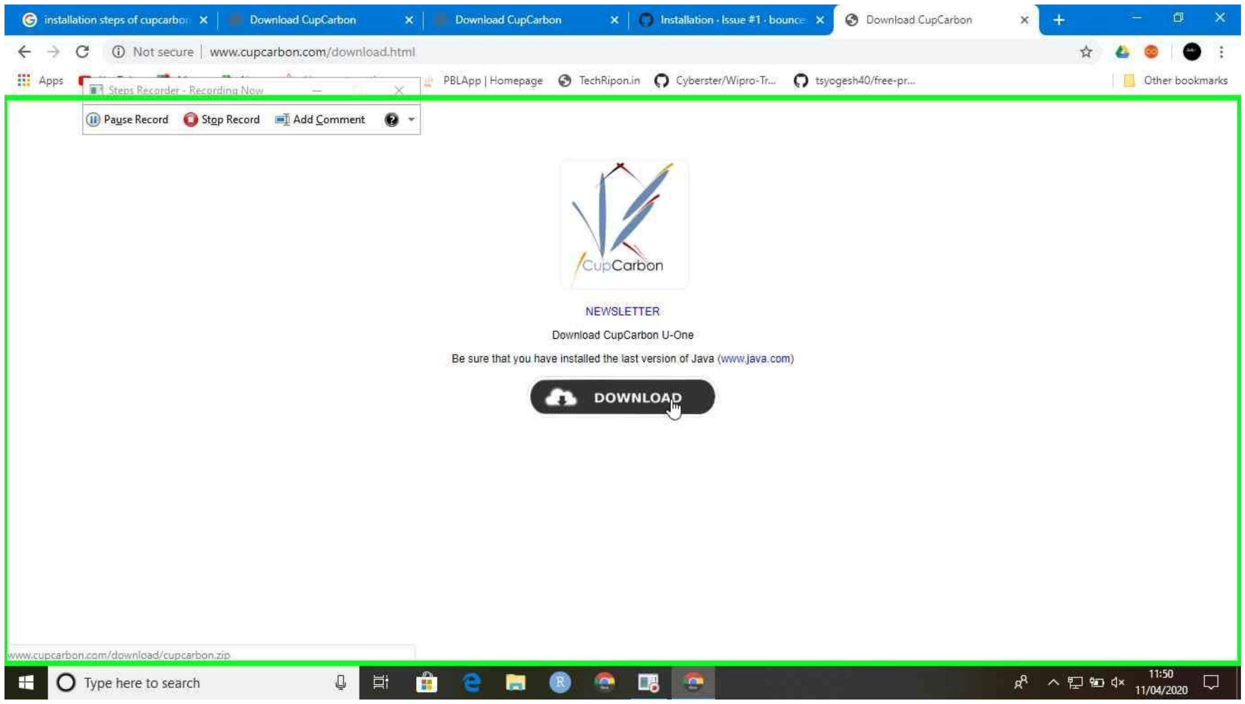
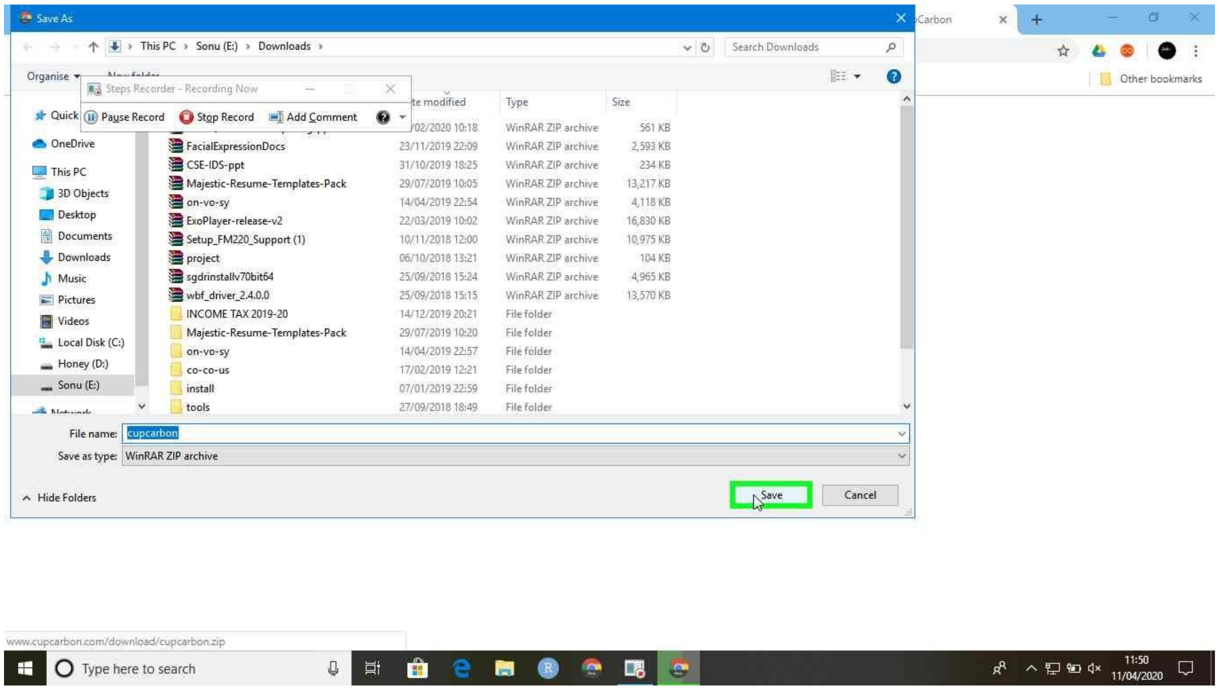

NEWSLETTER
Download CupCarbon U-One
Be sure that you have installed the last version of Java (www.java.com)
DOWNLOAD

Figure 5.10**:** Download the Cupcarbon Zip File

Step 6: Download the Cupcarbon Zip File and save it.


Save As

This PC › Sonu (E:) › Downloads ›

Name	Date modified	Type	Size
	/02/2020 10:18	WinRAR ZIP archive	561 KB
FacialExpressionDocs	23/11/2019 22:09	WinRAR ZIP archive	2,593 KB
CSE-IDS-ppt	31/10/2019 18:25	WinRAR ZIP archive	234 KB
Majestic-Resume-Templates-Pack	29/07/2019 10:05	WinRAR ZIP archive	13,217 KB
on-vo-sy	14/04/2019 22:54	WinRAR ZIP archive	4,118 KB
ExoPlayer-release-v2	22/03/2019 10:02	WinRAR ZIP archive	16,830 KB
Setup_FM220_Support (1)	10/11/2018 12:00	WinRAR ZIP archive	10,975 KB
project	06/10/2018 13:21	WinRAR ZIP archive	104 KB
sgdrinstallv70bit64	25/09/2018 15:24	WinRAR ZIP archive	4,965 KB
wbf_driver_2.4.0.0	25/09/2018 15:15	WinRAR ZIP archive	13,570 KB
INCOME TAX 2019-20	14/12/2019 20:21	File folder	
Majestic-Resume-Templates-Pack	29/07/2019 10:20	File folder	
on-vo-sy	14/04/2019 22:57	File folder	
co-co-us	17/02/2019 12:21	File folder	
install	07/01/2019 22:59	File folder	
tools	27/09/2018 18:49	File folder	

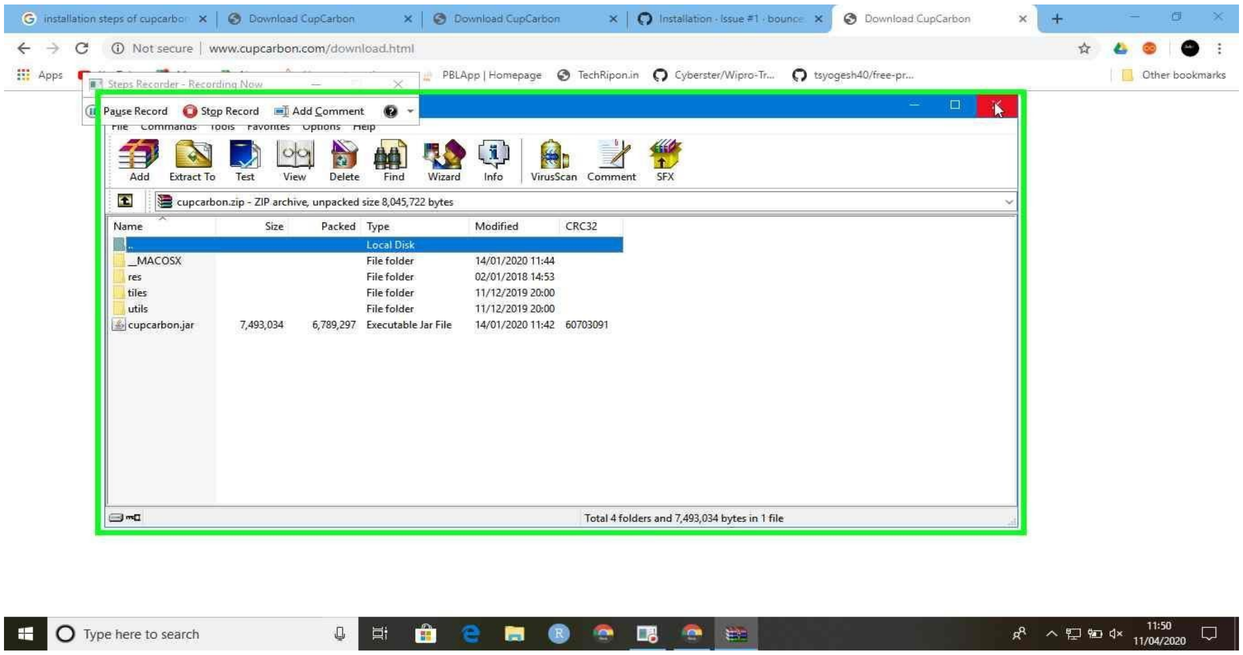
File name: cupcarbon

Save as type: WinRAR ZIP archive

www.cupcarbon.com/download/cupcarbon.zip

Figure 5.11**:** Cupcarbon Zip File and save it

Step 7: Extract the Zip File


www.cupcarbon.com/download.html

cupcarbon.zip - ZIP archive, unpacked size 8,045,722 bytes

Name	Size	Packed	Type	Modified	CRC32
..			Local Disk		
__MACOSX			File folder	14/01/2020 11:44	
res			File folder	02/01/2018 14:53	
tiles			File folder	11/12/2019 20:00	
utils			File folder	11/12/2019 20:00	
cupcarbon.jar	7,493,034	6,789,297	Executable Jar File	14/01/2020 11:42	60703091

Total 4 folders and 7,493,034 bytes in 1 file

Figure 5.12: Extract the Zip File

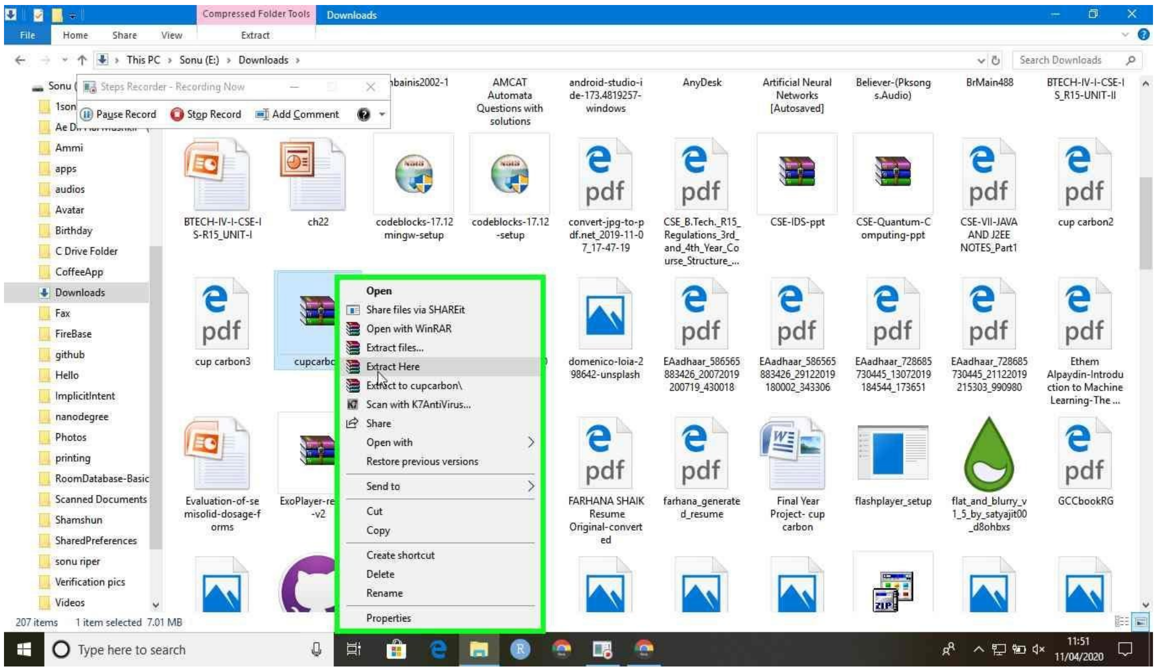


Figure 5.13: Extract the Zip File into the folder

Step 8: After extracting the Zip file you can have a java executable file with name Cupcarbon. Open that executable file then you will be directed into the Cupcarbon user interface.

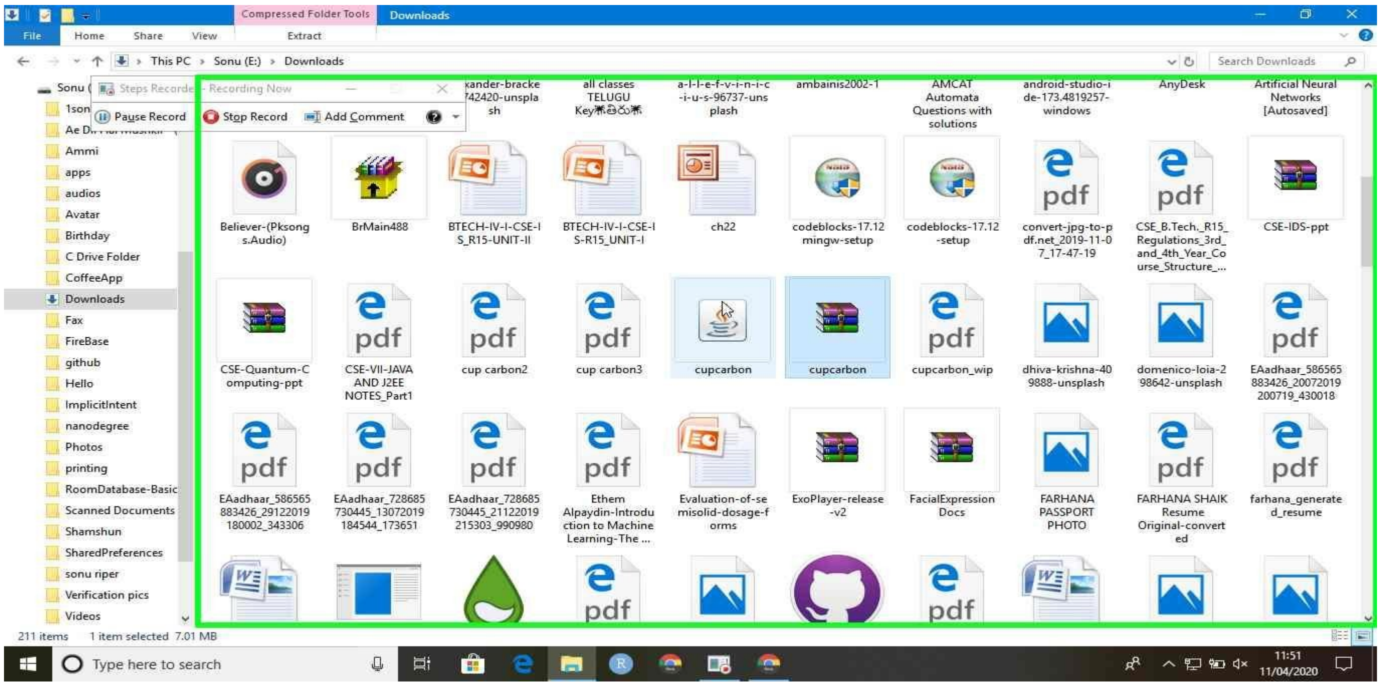


Figure 5.14: java executable file with name Cupcarbon

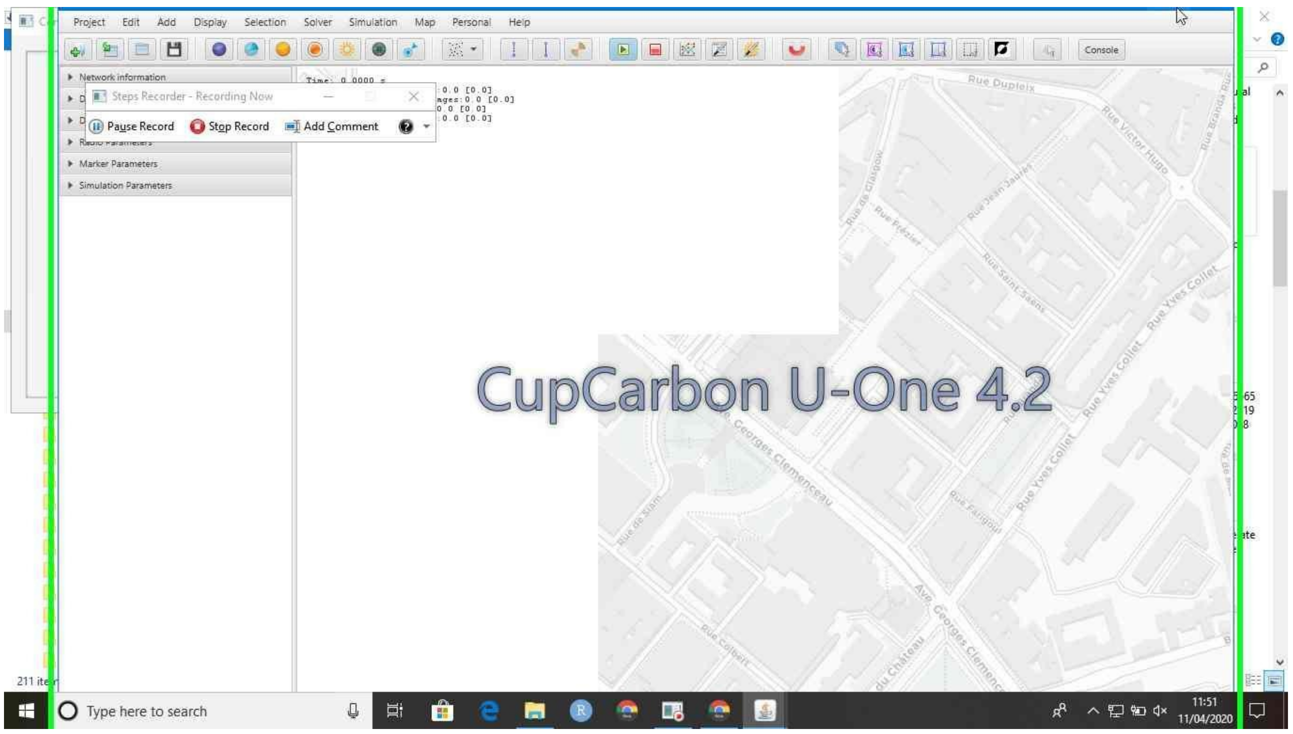


Figure 5.15: Cup Carbon User Interface

CHAPTER – 6
DESIGN

6.1 UMLIntroduction

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, this focuses on the behavioral modeling, implementation modeling and environmental modelviews.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningfulmodels.
2. Provide extendibility and specialization mechanisms to extend the coreconcepts.
3. Be independent of particular programming languages and developmentprocess.
4. Provide a formal basis for understanding the modelinglanguage.
5. Encourage the growth of OO toolsmarket.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate bestpractices.

Usage of UML inProject:

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks.

Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

Use CaseDiagrams:

In software engineering, a use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from the external point of view. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system.

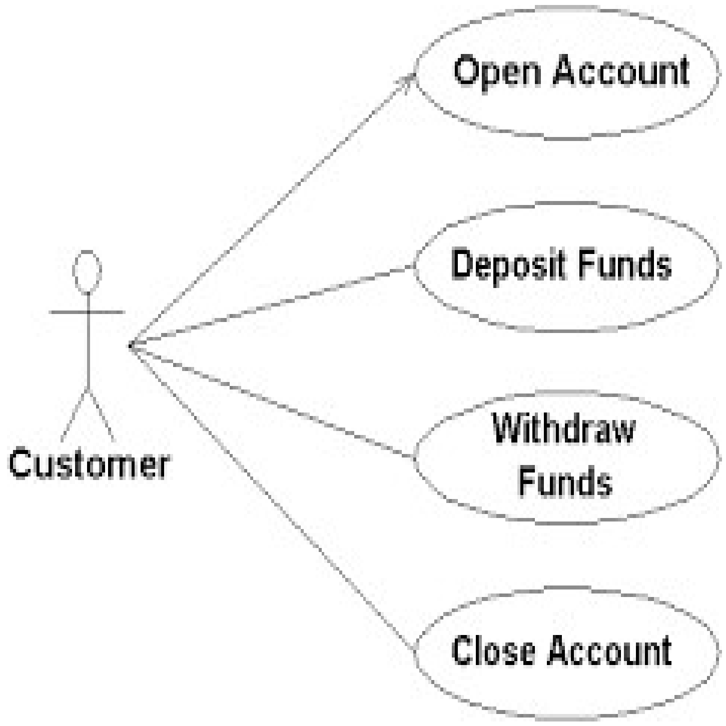


Figure 6.1: Usecase Diagram

SequenceDiagrams:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams

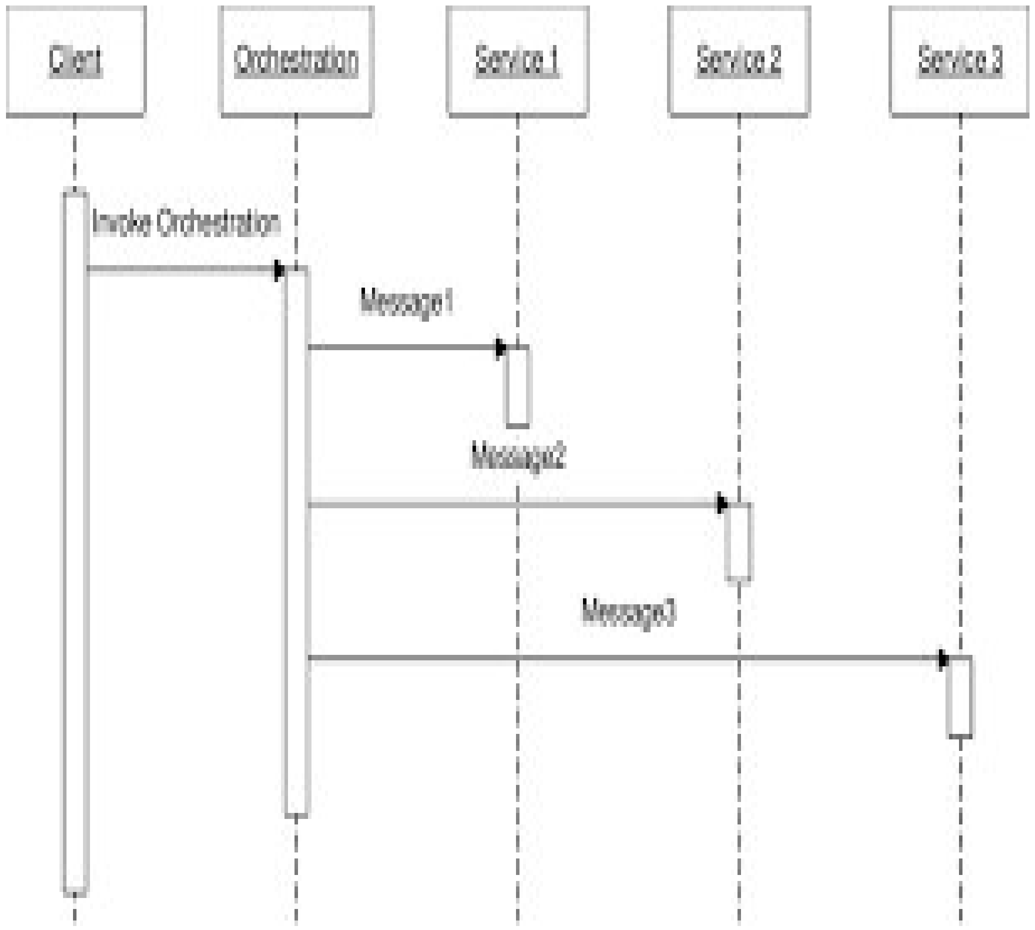


Figure 6.2: Sequence Diagram

ActivityDiagrams:

Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the over rill flow ofcontrol.

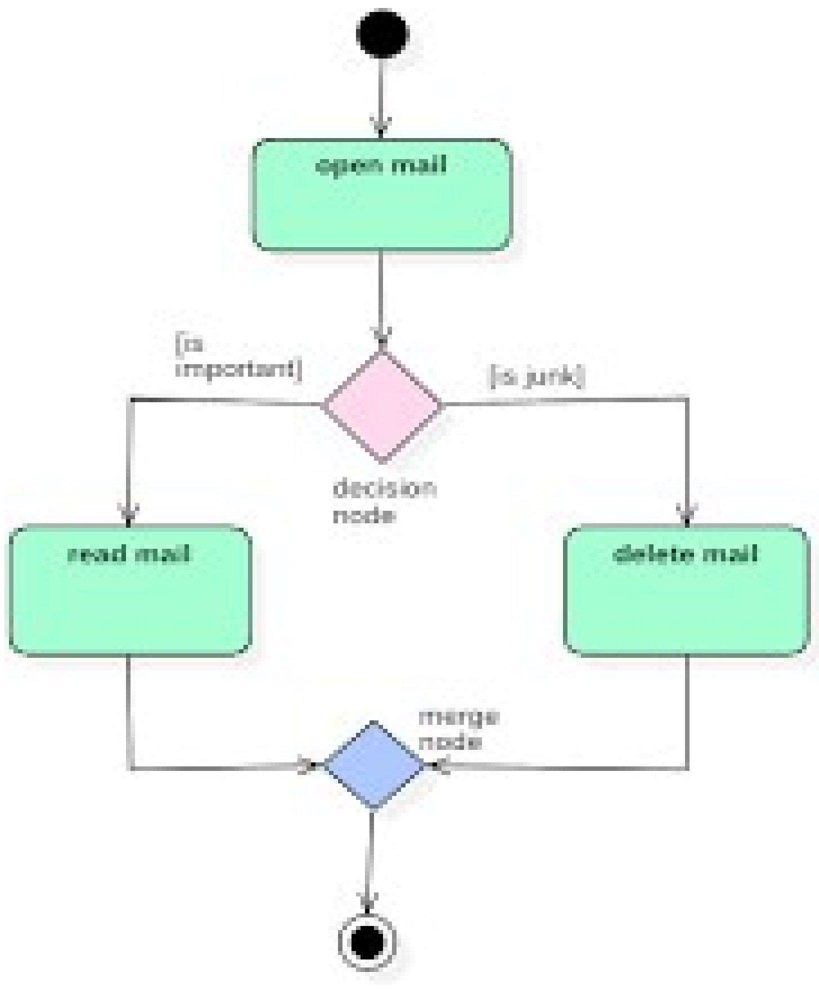


Figure 6.3: Activity Diagram

6.2 UMLDiagrams

UsecaseDiagram:

Usecase Diagram contain the 2 actors namely sink and sensor along with 3 activities as shown in the below figure

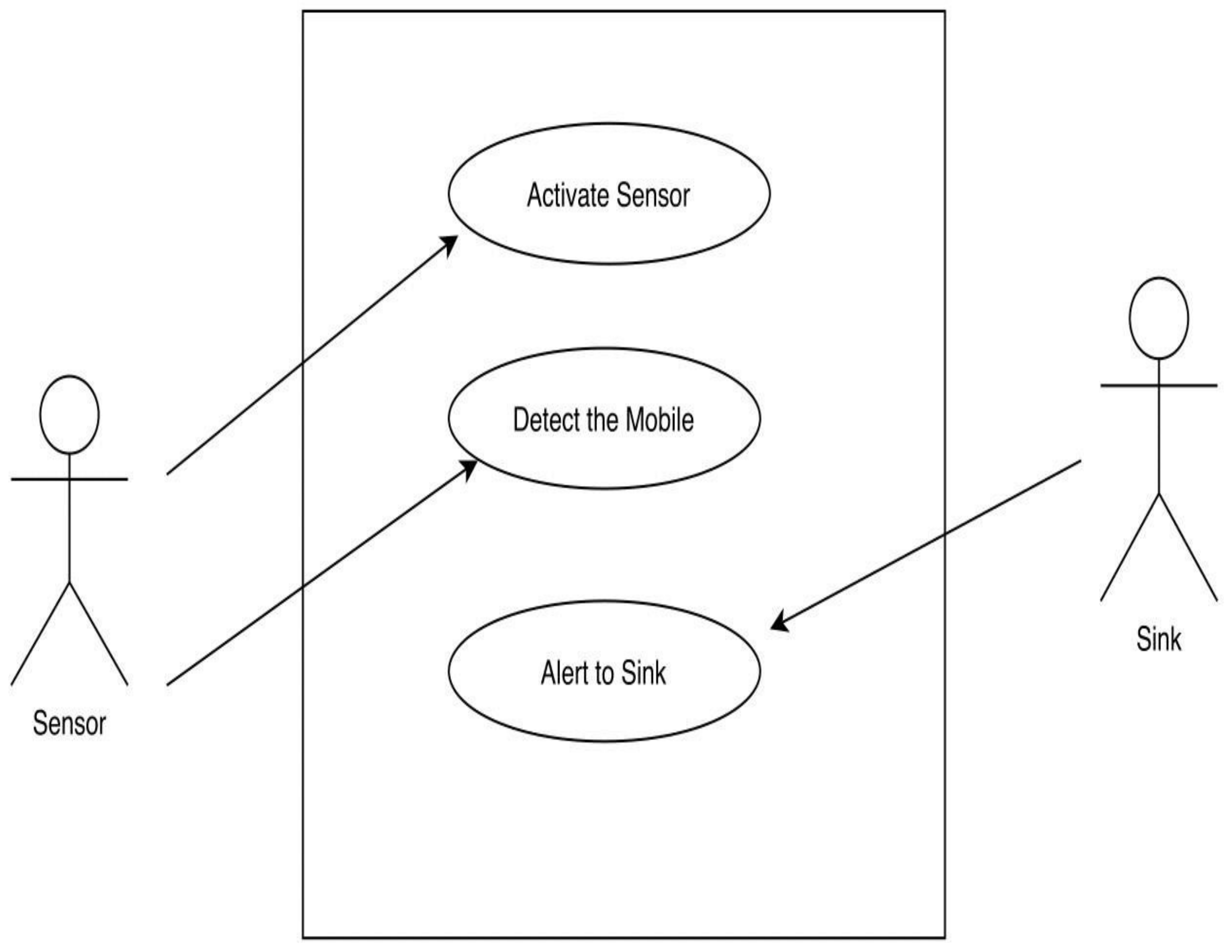


Figure 6.4: Usecase Diagram

Sequence Diagram:

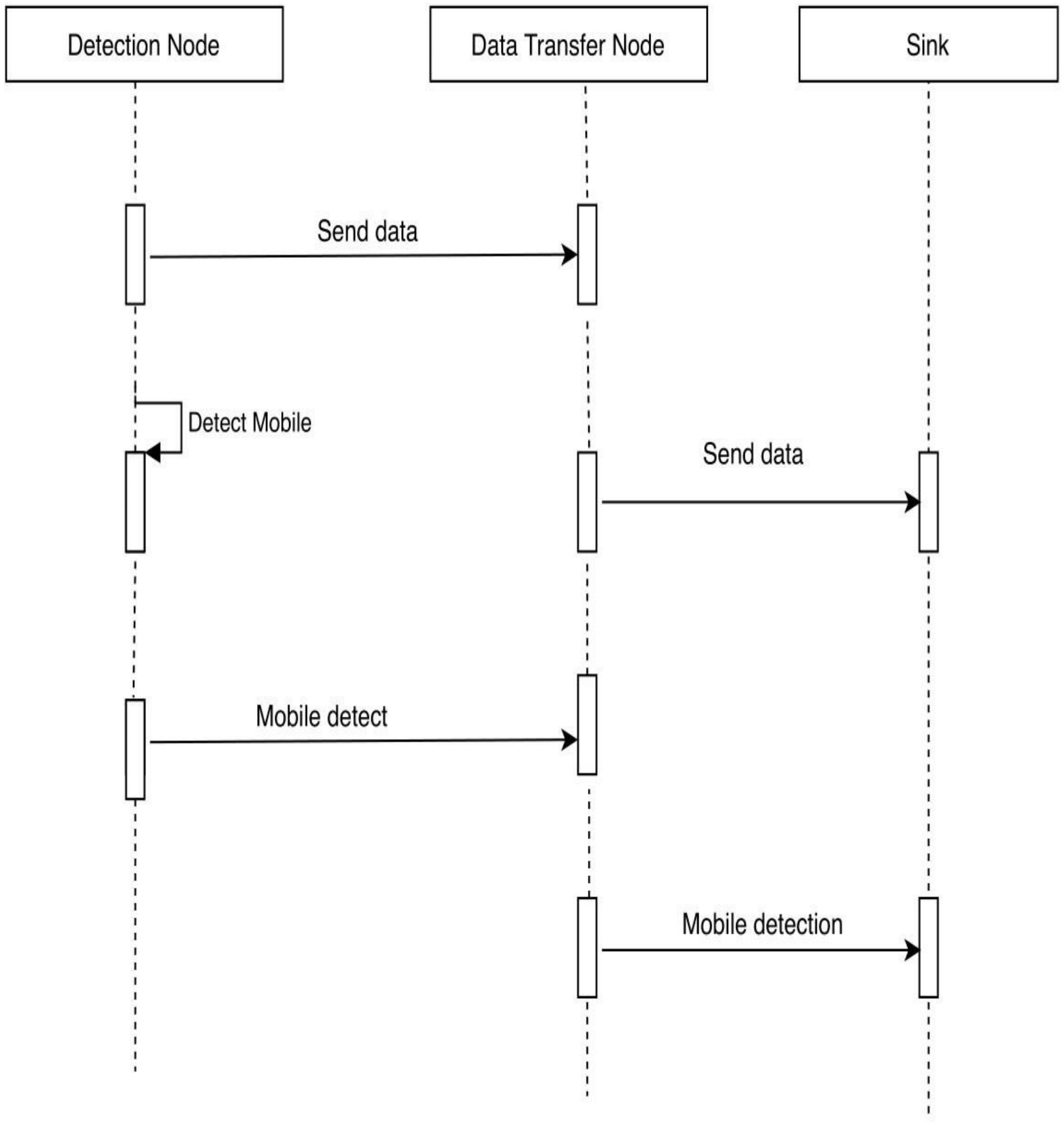


Figure 6.5: Sequence Diagram

Activity Diagram:

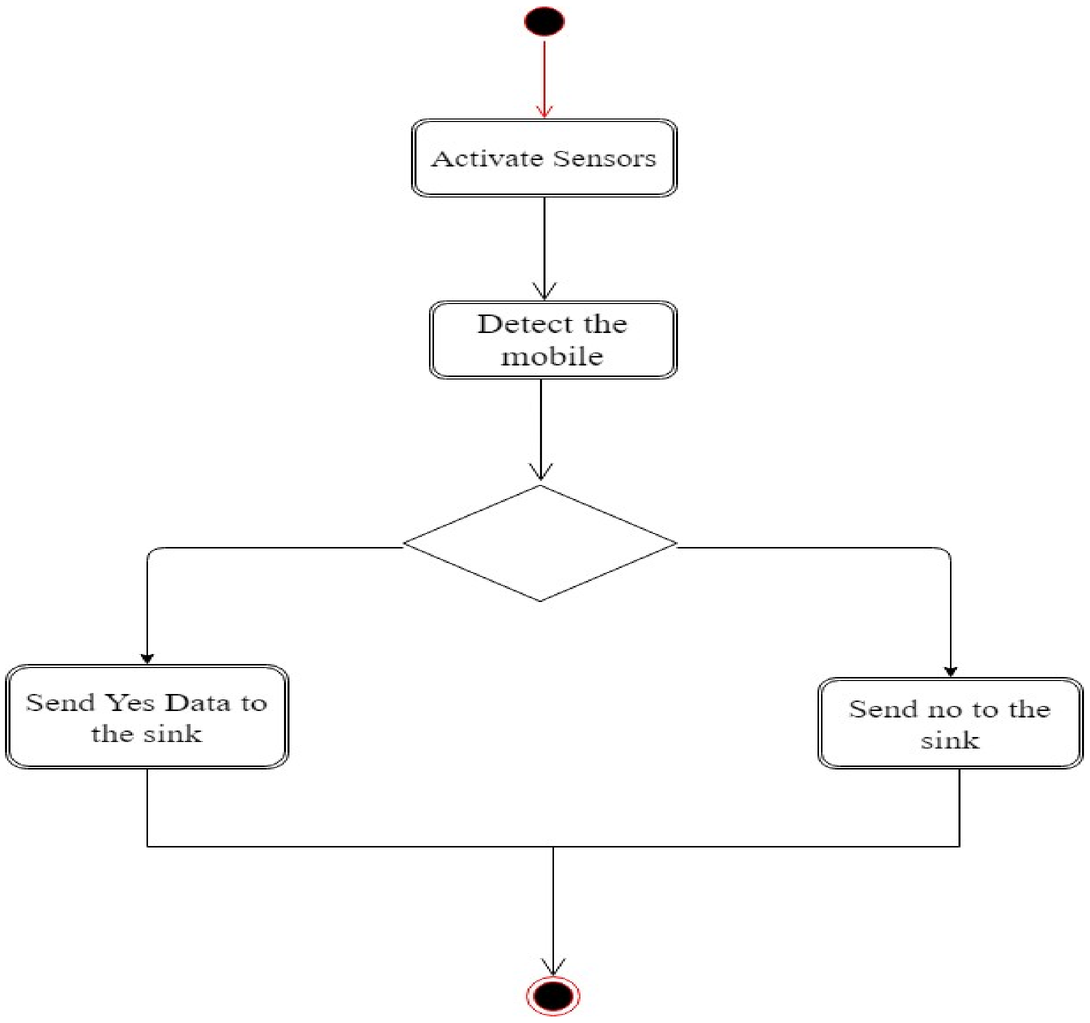


Figure 6.6: Activity Diagram

CHAPTER-7

IMPLEMENTATION

The Implementation of Mobile detection with the Cupcarbon by using Senscript as follows. Steps involved in the Implementation of Mobile Detection withCupCarbon:

1. Create a new project: this can be done either by clicking on the “New project” icon of the toolbar or on the menu Project à new project. Choose the name (File name: MobileDetection) and the place where you want to save your project. The project will create a new folder with the given project name. Inside this folder, 1 file (mobileDetection.cup) and 8 other directories will be created. The content of each directory is given in thefollowing:
   a. config: it contains the simulation parameters file, the building list file, the marker list file and two other directories (sensor and sensor_radios) that contains the list of sensor nodes (one file by sensor node) and the list of the radio modules of eachsensor.
   b. gps: it contains the list ofroutes
   c. logs: it contains the logfile
   d. results: it contains the simulation results (a csvfile)
   e. scripts: it contains the SenScript files of theproject
   f. netevents: it contains the natural eventfiles
   g. tmp,network: are used by thesimulator
2. Add a new sensor node on the map: click either on the Add Sensor icon of the toolbar or from the menu bar (Add à Add Sensor Node). Then, click on the map where you want to add the sensor node. Another click will lead to another new sensor node and so on. To stop adding sensor nodes, just click on the right button of the mouse. You can also click on the icon of the toolbar, or by typing on the escape [esc] button of thekeyboard.
3. Open the SenScript Window: the SenScript window can be opened by clicking on the icon of the toolbar or from the menu Simulation à SenScriptWindow.
4. Write the script: add the followingscript
   a. in the text area part of the SenScript window, Add the name of this script in the File namefield
   b. then click on the Savebutton

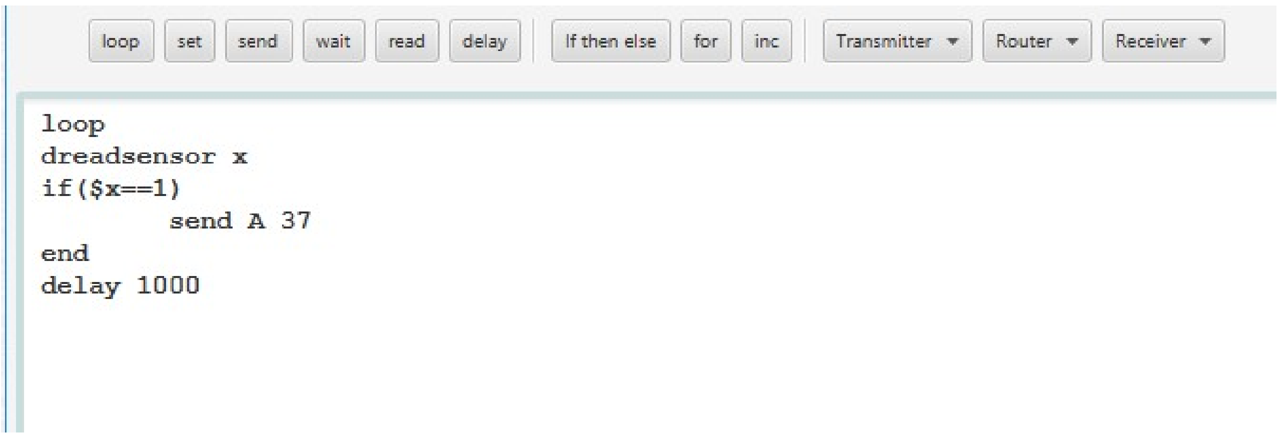
c. Just in the left part of this field. This will create a file '.csc' in the directoryscripts.

d. Finally, close the SenScriptWindow.

5. Assign the SenScript file to the sensornode:
6. Select the sensor node on themap
7. Go to Device Parameters in the left part of the mainwindow
8. Then, select the '.csc' file in the field Scriptfile
9. And then, click on the apply button just in theright
10. Note that once the script is assigned to a sensor, the center will be colored in orange. This can help to detect graphically sensors withoutscripts.
11. Run the simulation: For this example, there is no need to parameterize the simulation, just click on the run simulation button in the toolbar or in the simulation parameters menu in theleft.
12. Simulation results: in this example the simulation results shows a message displayed by thesensor.
13. Save the project: Click on the icon to save theproject.

Code Implemented in senscript language. There are three types of Nodes namely detection node, Transmission node, Sink

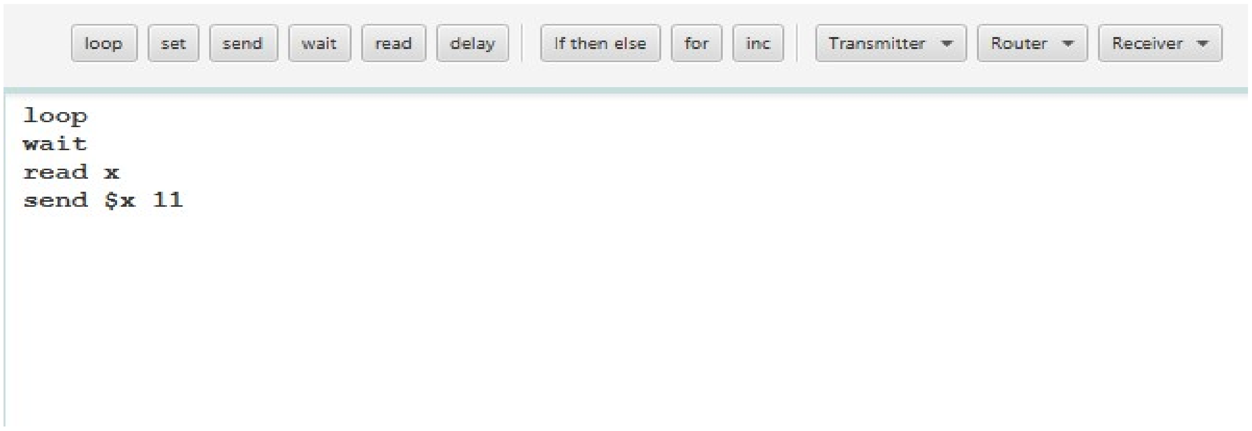
The code for Detection node:



```
loop
dreadsensor x
if($x==1)
        send A 37
end
delay 1000
```

Figure 7.1:Code for detection node

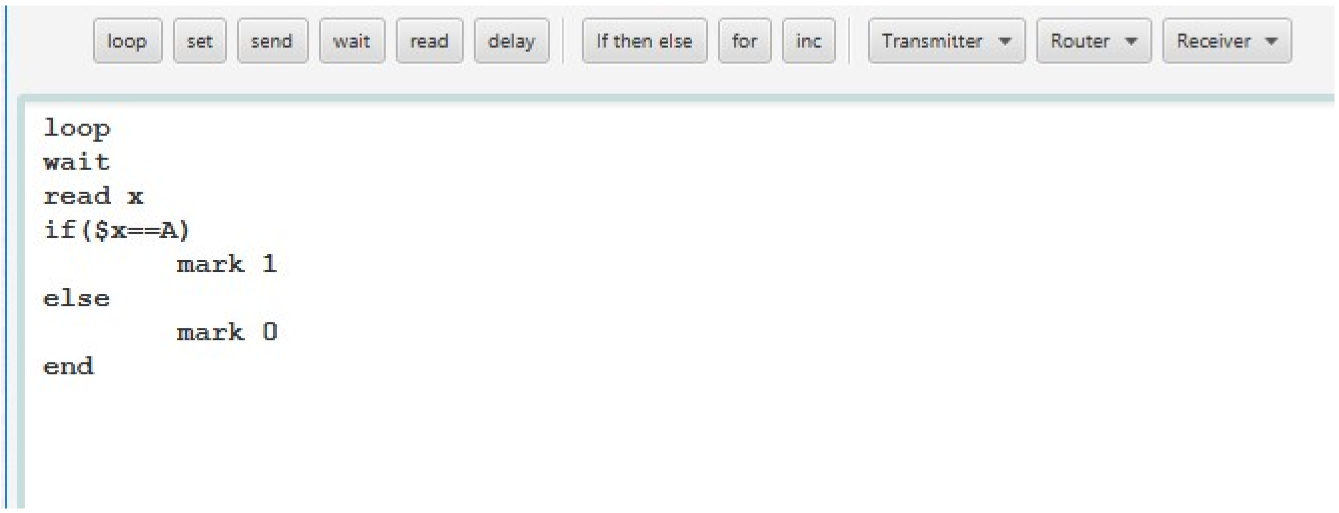
The code for Transmission node:



```
loop
wait
read x
send $x 11
```

Figure 7.2: Code for transmission of a node

The code for Sink:



```
loop
wait
read x
if($x==A)
        mark 1
else
        mark 0
end
```

Figure 7.3: Code for a Sink

CHAPTER – 8

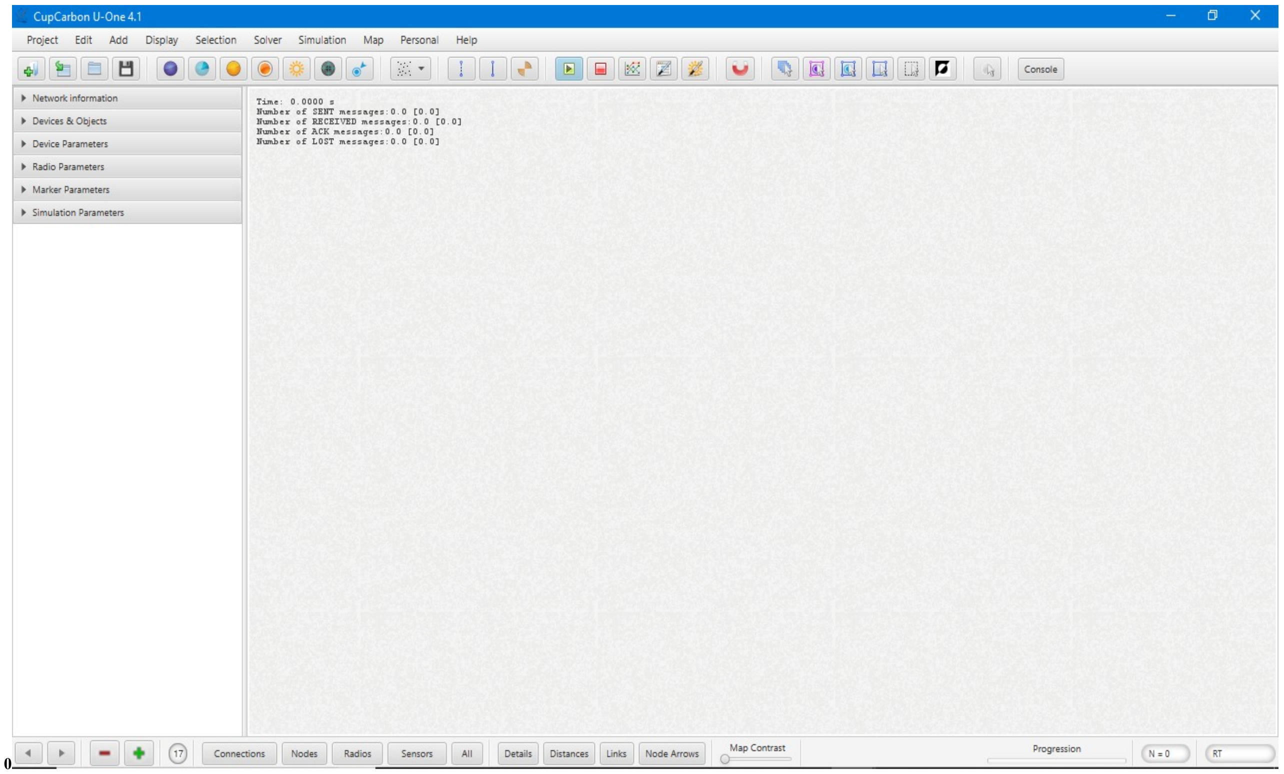
RESULT ANALYSIS

This project has been successfully executed its source code. Initially there were some errors and bugs occurred while running the code. By resolving them, the code is fully free from errors and bugs.

The following are screenshots of some of the outcomes while executing the source code and some are the screenshots of environment which we have used in ourproject.

Screen-1: Home Page of User Interface

The user Interface of the cup carbon before the file is loaded



```
Time:  0.0000 s
Number of SENT messages:0.0 [0.0]
Number of RECEIVED messages:0.0 [0.0]
Number of ACK messages:0.0 [0.0]
Number of LOST messages:0.0 [0.0]
```

Figure 8.1: Home Page of User Interface

Screen-2: Loading a file in User Interface

For loading the file first we have to go the “File” option there you go to load the file if already created or you can create a new file.

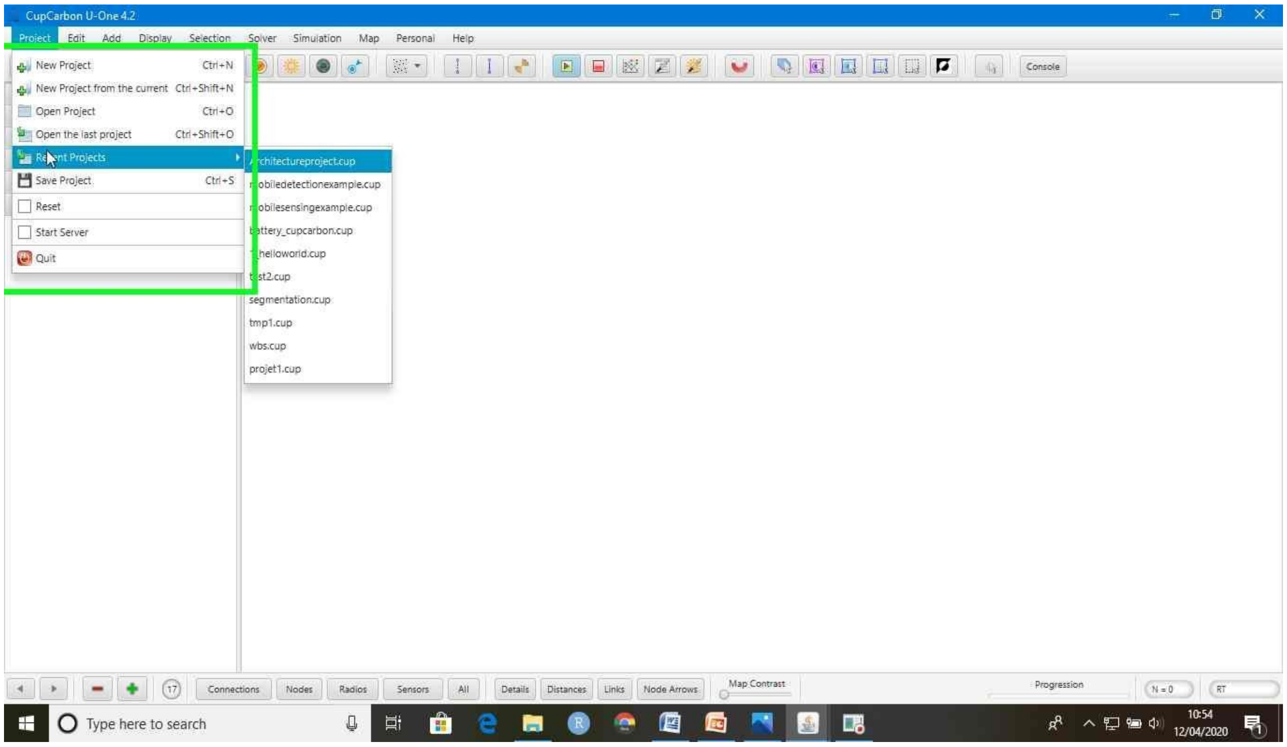


Figure 8.2: Browsing a file

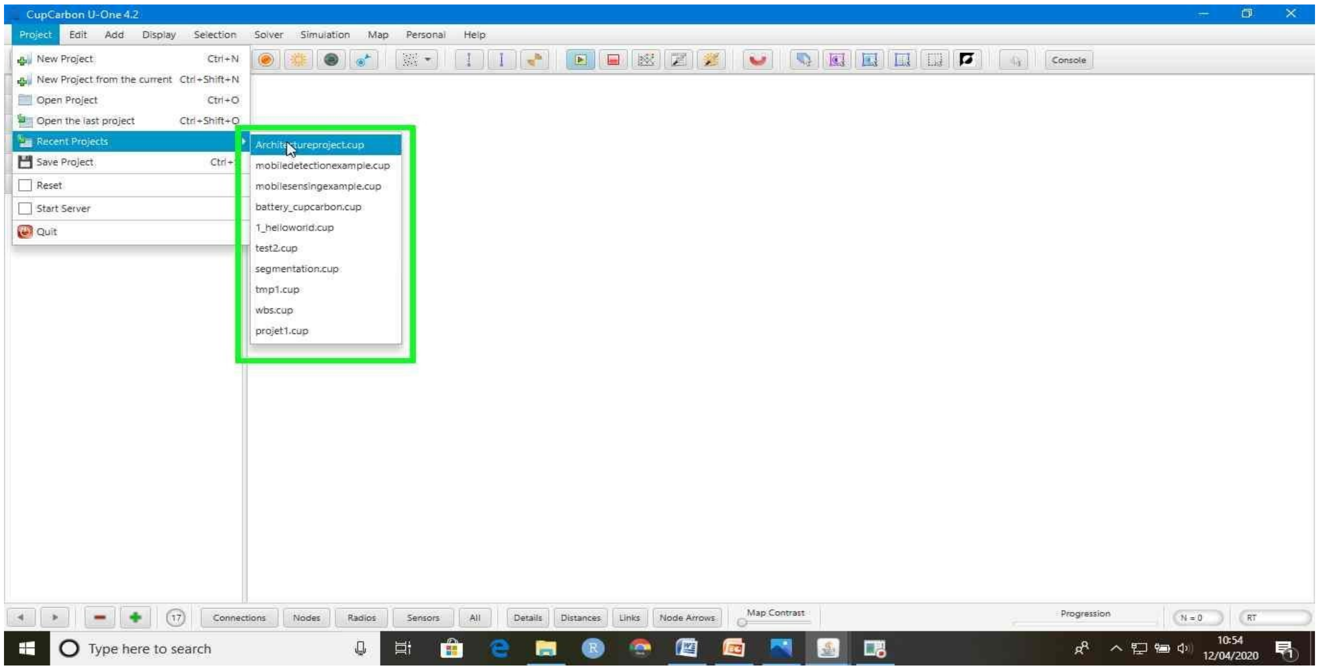
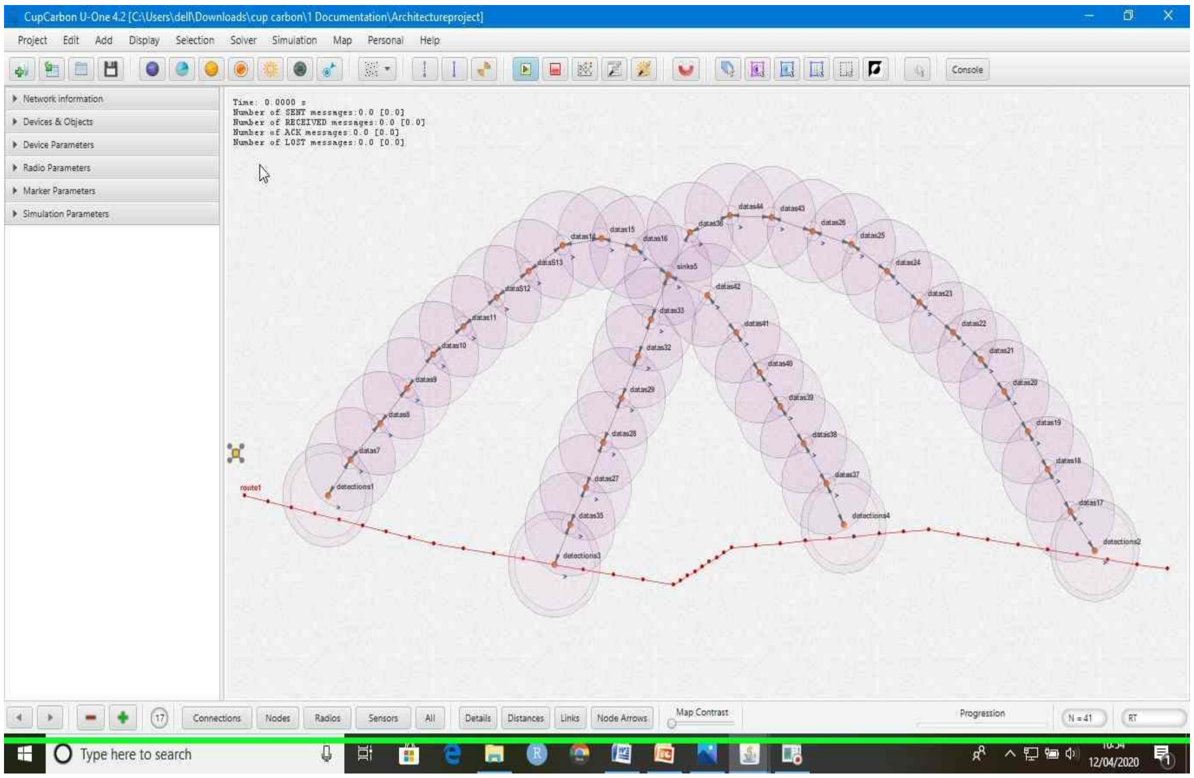


Figure 8.3: Loading a file in User Interface

Screen-3: After loading the project in User Interface

After loading the project in User Interface looks like as shown in below figure here we already created a project with nearly 40-50 sensor nodes and connected all these sensor nodes for transferring the message.

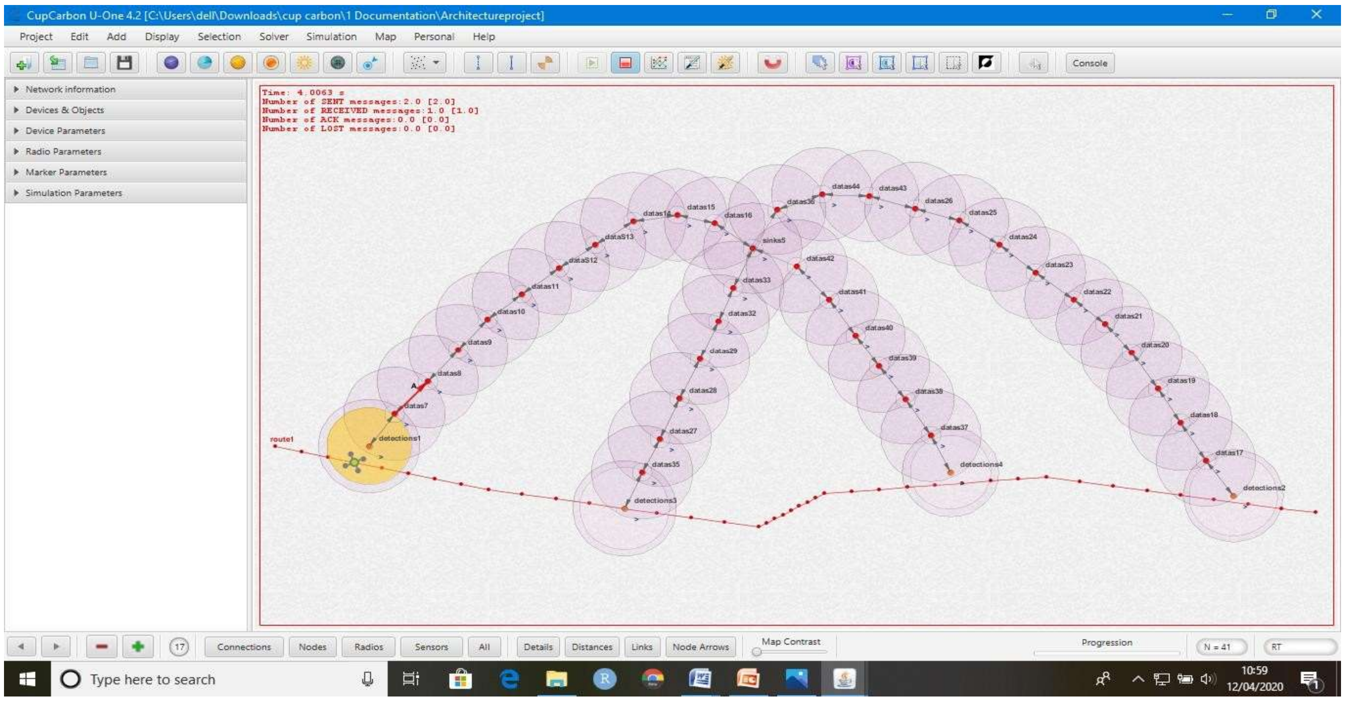

CupCarbon U-One 4.2 [C:\Users\dell\Downloads\cup carbon\1 Documentation\Architectureproject]
Project Edit Add Display Selection Solver Simulation Map Personal Help
Console
Network information
Devices & Objects
Device Parameters
Radio Parameters
Marker Parameters
Simulation Parameters
route1
sinks5
detections1
detections2
detections3
detections4
Connections Nodes Radios Sensors All Details Distances Links Node Arrows
Map Contrast
Progression
N = 41
RT
Type here to search
12/04/2020

```
Time: 0.0000 s
Number of SENT messages:0.0 [0.0]
Number of RECEIVED messages:0.0 [0.0]
Number of ACK messages:0.0 [0.0]
Number of LOST messages:0.0 [0.0]
```

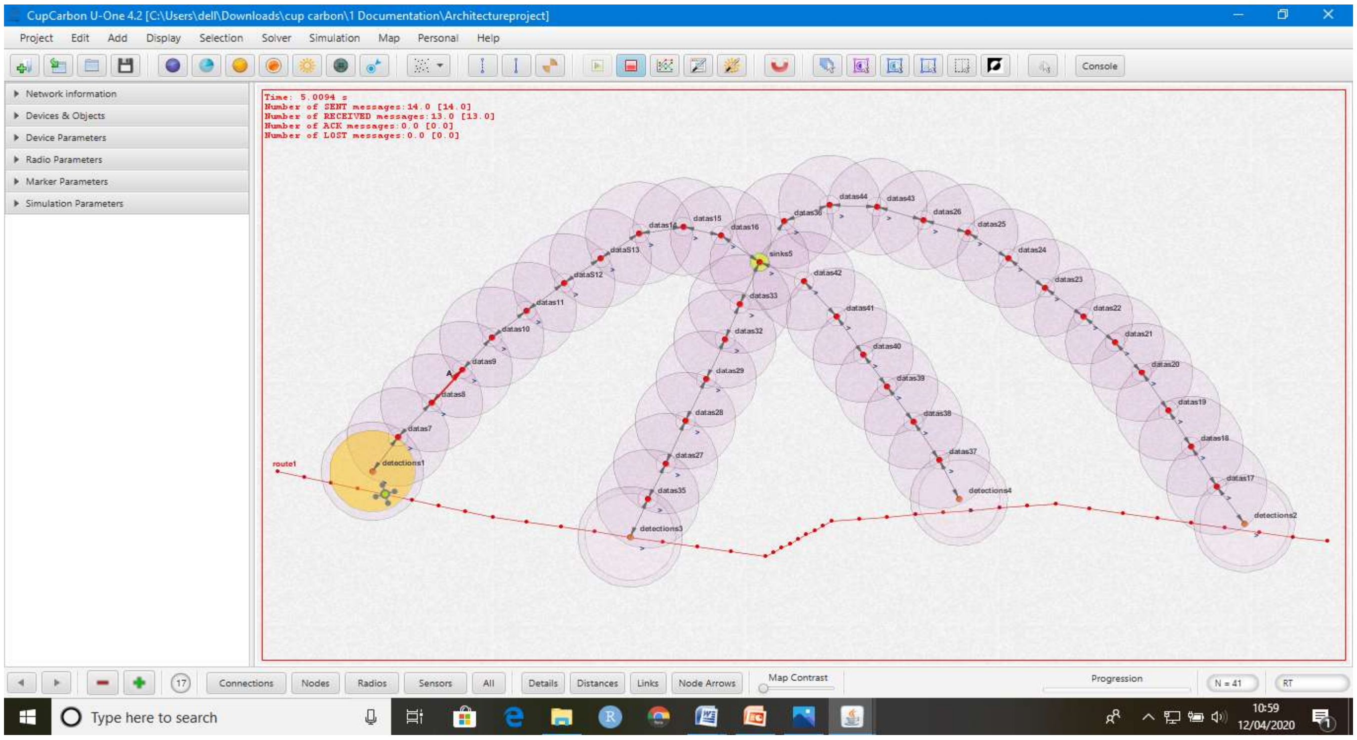
Figure 8.4: After loading the project in User Interface

Screen-4 Execution of the Project

The execution of the project involves setting the simulation parameters hence set the simulation time to 200 so that it becomes easy to us to watch the simulation parameters. Then apply the changes you have made and run the simulation process.



```
Time: 4.0063 s
Number of SENT messages:2.0 [2.0]
Number of RECEIVED messages:1.0 [1.0]
Number of ACK messages:0.0 [0.0]
Number of LOST messages:0.0 [0.0]
```


CupCarbon U-One 4.2 [C:\Users\dell\Downloads\cup carbon\1 Documentation\Architectureproject]
Project Edit Add Display Selection Solver Simulation Map Personal Help
Console
route1
detections1
sinks5
detections2
detections3
detections4
N = 41
10:59
12/04/2020

```
Time: 5.0094 s
Number of SENT messages:14.0 [14.0]
Number of RECEIVED messages:13.0 [13.0]
Number of ACK messages:0.0 [0.0]
Number of LOST messages:0.0 [0.0]
```

Figure 8.5: Execution of the Project

Screen- 5: Graph for Energy Consumption

For the evaluation of graphs there should some things we have to do like marking the "results" tab in Simulation Parameters and then selecting the nodes for which the maps are drawn on the graphs. Here all nodes are selected .After selecting all nodes run the simulationprocess.

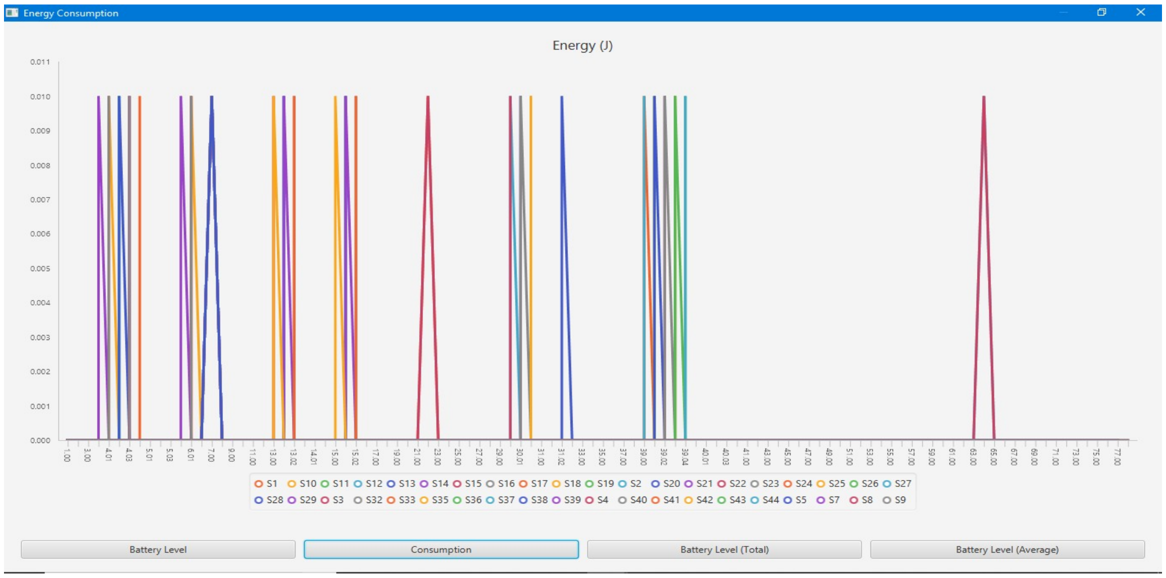


Figure 8.6: Graph for EnergyConsumption

Screen-6: Graph for Battery Usage

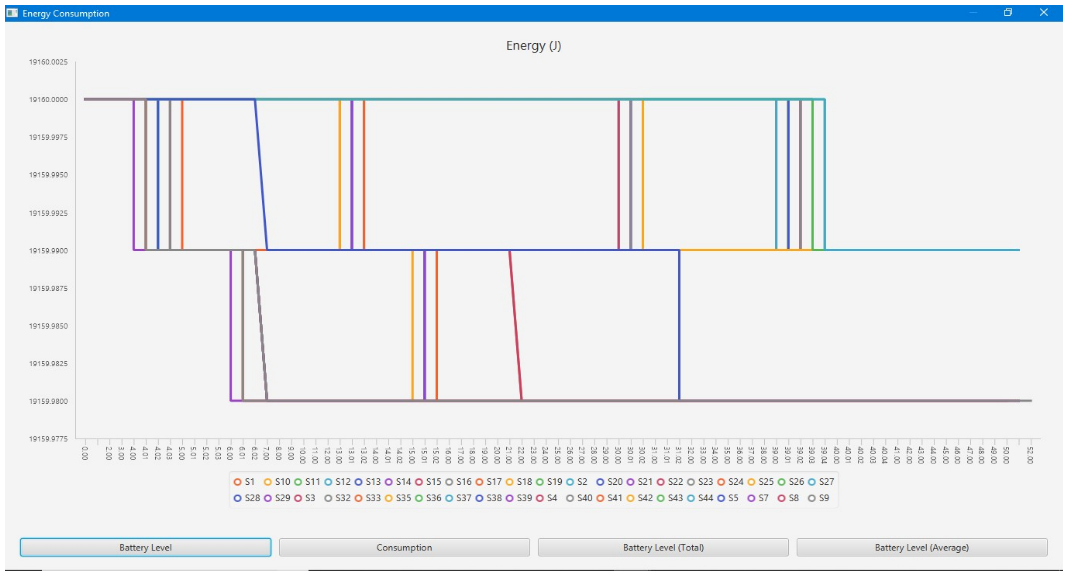


Figure 8.7: Graph for Battery Usage

As we can see, when the algorithm is executed, the time during which the network remains active is longer, with an improvement of the 20%.

However, CupCarbon could be used as a complement to the rest. First of all, as mentioned, it is an application written in Java, so a possible solution to its greatest disadvantage (the absence of integrated protocols) is the modification of the source code. In this way, we can create functions that implement the desired algorithms that are not integrated in the tool. In addition, the possibility of modifying the source code, that is, the open source is another greatadvantage.

On the other hand, CupCarbon has the possibility of observing the consumption of resources that occur due to the nature of urban networks. The sensors can be displayed directly on the map. Additionally, the graphical interface is another of its advantages; it is intuitive for the user and allows performing all the functionalities through it. Another advantage is the lack of important prerequisite requirements.

The only ones we need is to access to the program and programming in Java. For these reasons, intuitive interface and reduced prior knowledge, it is a suitable tool for academic or networking environments. Finally, the presence of energy modeling that can be plotted is another advantage since it is an important parameter to be tested inWSNs.

As the project results of energy consumption and battery usage of all sensor nodes and single sensor nodes can be viewed in Graph viewer. Below are the diagrams of the energy consumption using Cupcarbon graph viewer. As a result it is possible to keep all the nodes of the network alive for 20% more time than using a conventional Dijkstra algorithm. Additionally, it is verified that the total energy dissipated from the network is distributed more equitably. Consequently, a more efficient network is achieved.

CONCLUSION

The results obtained thus conclude that CupCarbon shows the less energy consumption and long lifetime of a sensor node in the network. As a result of our modified Dijkstra, it is possible to keep all the nodes of the network alive for 20% more time than using a conventional Dijkstra algorithm. Additionally, it is verified that the total energy dissipated from the network is distributed more equitably. Consequently, a more efficient network is achieved.

Finally, the following conclusions can be drawn. The evaluation of CupCarbon has not been satisfactory enough, although, given its continuous updates, its perfection and full development of some of its most interesting features are expected, such as the possibility of migration of code to the Arduino platform. Although CupCarbon has allowed us to develop the algorithms that we wanted, the fact of having to work with different programming languages makes it not recommended for certain applications. However, CupCarbon could be used as a teaching complement in subjects related to networks.

Future Scope:

- Future work will be the possibility of developing other protocols focused on saving energy consumption forWSNs.
- Development of more efficient integrated Protocols to increase the batterylife.

REFERENCES

[1] Umber Noreen. “Interference modeling for 3D simulator of sensor networks dedicated to smart cities. Modeling and Simulation”. University de Bretagne occidentale - Brest, 2018.English.

[2] Prof Ahcène Bounceur “CupCarbon User Guide” Version U-One4.1

[3] Network Protocols and Algorithms ISSN 1943-3581 2018, Vol. 10, No. 2 –”Evaluation of CupCarbon Network Simulator for Wireless Sensor Networks“.

[4] Massinissa Saoudi, Ahc`ene Bounceur, Reinhardt Euler, Tahar Kechadi, Alfredo Cuzzocrea –” Intelligent Data Mining Techniques for Emergency Detection in Wireless SensorNetworks”.

[5] FaridLalem, MuathAlshaikh, Ahc`eneBounceur, ReinhardtEuler, LamriLaouamer, LaurentNana, AncaPascu–“ Data Authenticity and Integrity in Wireless Sensor Networks Based on a WatermarkingApproach”.

[6] Cristina López-Pavón, Sandra Sendra1, Juan F. Valenzuela-Valdes - “Evaluation of CupCarbon Network Simulator for Wireless SensorNetworks”.

[7] Faisal Alzyoud, Nesreen AL Sharman, Thamer Al-Roosan, YahyaAlsalah -“ Smart Accident Management in Jordan using Cup CarbonSimulation”.

[8] Umber Noreen, AhcèneBounceur, Laurent Clavier – “Modeling Interference for Wireless Sensor NetworkSimulators”.

[9] AhcèneBounceur, Laurent Clavier, Pierre Combeau, Olivier Marc, RodolpheVauzelle, Arnaud Masserann, Julien Soler, Reinhardt Euler, Taha Ahmed Mohammed Al Wajeeh, Vyas Devendra – “CupCarbon: A New Platform for the Design, Simulation and 2D/3D Visualization of Radio Propagation and Interferences in IoT Networks ”.

[10] K.Parvatessam, G.A. ArunKumar –“PRESENCE OF ACTIVE MOBILE PHONES AND HIDDEN CAMERADETECTION”.

[11] Claudio Fiandrino, Andrea Capone w, Giuseppe Cacciatore,-“ CrowdSenSim: a Simulation Platform for Mobile Crowdsensing in Realistic Urban Environments”.

Websites :

[12] www.cupcarbon.com

[13] https://java.com/en/download/