CrossRef DOI of original article: 10.34257/GJCSTDVOL21IS1PG21

Development of a Portable IP-Based Remote Controlled System for Mobile Robot

Daramola O. A.[1], Obe O.[2] and Oriolowo A.[3]

[1] Federal University Of Technology

Received: 11 June 2021 Accepted: 2 July 2021 Published: 15 July 2021

Abstract

The use of Mobile Robots to interact with objects in remote locations has proved to be useful in areas not easily accessible or too dangerous for humans. Various means have been used to remotely operate or control Mobile Robots. These range from wired connection to Wireless connection like radio frequency signal and more recently internet controlled Mobile Robot using the TCP/IP protocol stack. However, the problem of remote control dependence on the Mobile Robot Platform or configuration has made it difficult to switch controllers between Mobile Robots. In this work, a portable IPbased remote control system has been designed and implemented to remove the constraint imposed by the Mobile Robot's platform in choosing the control interface. The system developed was built on three loosely coupled components working together to ensure a high degree of Control interface portability. The Mobile Robot Gateway component was used to receive and send data from the Mobile Robot

---

Index terms— mobile robot, remote control, command hub, rest architecture, TCP/IP, IP-based

1 Introduction

obile robots were defined by Posadas et al (2008) as physical agents that move and interact continuously while embedded in a dynamic environment. A mobile robot is also described as a situated and embodied agent endowed with mobility (Obe and Dumitrache, 2012). Control of mobile robots is categorized into three types namely, autonomous control, semi-autonomous control, and Tele operation. Autonomous control implements various control algorithms that control mobile robots in their environment without human intervention. Semiautonomous control allows a user to instruct the mobile robot on what to do but the robot decides how the task is carried out. This mode of control is supervisory. In a Tele operated control, the mobile robot is entirely controlled by the user. Oxford dictionary (2015) defines "remote control" as the control of a machine or apparatus from a distance utilizing radio or infrared signals transmitted from a device. Teleoperation is often used in place of Remote control in research or technical environments. Teleoperation means controlling or doing work at a distance (Wichmann et al, 2014). The meaning of the distance, however, can vary. The distance can be physical, such as an operator controlling a robot at a remote location. The distance could also be a change in scale, for example, a surgeon using teleoperation to conduct surgery at the microscopic level. In 2014, Wichmann et al. reported that teleoperation systems are designed using a master-slave system model where the control system depends heavily on the mobile robot platform.

Various communication technologies have been used to achieve teleoperation for a mobile robot. These technologies include Bluetooth, Infrared, WIFI, GSM, Internet of Things (IoT) (Chikurtev, 2019), and Internet (Luimula et al., 2007). While some of these technologies impose a constraint on the range of operation, GSM and the Internet have proven to overcome such barriers (Chin et al, 2003, Ankit, 2014 **??**nd Juang&Juang, 2016). The Internet which is based on TCP/IP enables connectivity of billions of devices worldwide, giving access to communication, data, pictures, videos, and even real images of distant environments (Siegwart and Saucy, 1999). The advancement in the development of internet applications has made it the ideal testing ground for sophisticated new applications, such as video-conferencing and remote control systems (Oboe & Fiorini, 1997). However, only a few examples of real physical interaction with distant places are available at the moment. Goodrich and Schultz (2007) defined two categories of interaction, remote and proximate. Remote interaction refers to the situation where the human and the robot are separated spatially or even temporally (i.e., Opportunity Mars rover), while with proximate interaction the humans and the robots are collocated.

This research work aims at creating a portable IP-based remote-controlled system for mobile robot interaction. According to James (1997), a software unit is portable (exhibits portability) across a class of environments to the degree that the cost to transport and adapt it to a new environment in the class is less than the cost of re-development. Portability in general terms is the usability of an object or component of the system in multiple environments without any modification in the internal structure.

Internet Protocol (IP) is a protocol in the Internet Layer of Transport Control Protocol/Internet Protocol (TCP/IP) model. It hides the underlying physical network by creating a virtual network view (Lydia et al, 2006). It is an unreliable, best-effort, and connectionless packet delivery protocol. Best-effort means that the packets sent by IP might be lost, arrive out of order, or even be duplicated. IP was designed with the assumption that a higher layer of the protocol will address these anomalies. According to Lydia et al., (2006), one of the reasons for using a connectionless network protocol was to minimize the dependency on specific computing centers that used hierarchical connection-oriented networks. Available remote control systems require specialized hardware and software that are heavily dependent on the mobile robot's platform or architecture. In this work, an architecture is proposed to eliminate the dependency of the remote control system on the robot's platform.

There are issues with existing mobile robot remote control systems which include but are not limited to, operation range, portability, and the multi-interface option of the system. Various researchers have addressed some of these problems, however, the problem of switching a remote control system from one mobile robot to another with little or no reconfiguration has not been dealt with. In Chin et al. (2003), a Real-Time Remote Control Architecture Using Mobile Communication was proposed. GPS, GSM, and GIS were applied in the development of real-time communication for remote data transfer which solved the range of operation problems, but the system lacked portability. Jose et al. (2004) illustrated a Java/Matlab-Based Environment for Remote Control System Laboratories. Matlab/Simulink and the Quanser Win Con environment were used to develop a control system using the HTTP server to process client requests. The client program is a java applet written in Java. The range of operation issue was eliminated by the architecture but failed to address portability and multi-interface options. In Mobile Robot Temperature Sensing Application via Bluetooth, Abdullah and Poh (2011) developed a control system using the KC-21 Bluetooth module and PIC16F877A microcontroller for remote temperature sensing. The system has a limited range of operations. Ankit et al. (2014) in Controlling of Remote Robot through mobile phone using DTMF Signal, developed a remote control system using microcontroller, CDMA modem, and DTMF signal. The system lacks portability and multi-interface options. Almali et al. (2015) developed a Wireless Remote control for Mobile robots operating in dangerous or narrow places for human beings. A 433MHz RF transceiver module was used to establish a connection between the mobile robot and the computer controlling it. The system has a limited operating range and also lacked portability.

This research work addresses the issue of portability and multi-interface options for mobile robot remote control systems. This will facilitate the use of one remote control system with a multi-interface for different mobile robots on several platforms, this will give room for using a robust interface for the control system at a particular time.

2 II.

3 Related Literature

Oboe and Fiorini, 1998 in "A design and control environment for internet-based telerobotics describe the environment for the design, simulation, and control of internet-based force-reflecting telerobotics using a segment of the network to connect the master to the slave. Simulation of the complete telerobotic system and emulation using a planar force-reflecting master and a virtual slave uses a MatLab-Simulink program interfaced with a set of dedicated routines for internet modeling. The issues in the variable time-delay system were addressed by using the delay parameters acquired from the network probe to design the controller. However, the work does not address multiple interfaces that could be used as the master or controller. Lung et al., 2002; designed an internet-based human-assisted robotic controller system but the security of the web page was not considered as anyone that stumbles on the web application can control the robot. Chin et al., 2003 adopted the use of G3 (global positioning system (GPS), global system for mobile (GSM), and geographic information system (GIS)) system in developing realtime communication and remote control systems for robot real-time remote control, navigation, and surveillance. The designed system is not portable as it was only implemented on a Windows platform. Visual Basic, which is the choice of programming language, is not supported on other operating systems. Jose et al., 2004; developed a Java/Matlab-Based Environment for Remote Control System Laboratories, illustrated with an Inverted Pendulum -a novel environment that provides 24-hours-a-day access to a Web-based lab for the remote control of different didactic setups. A detailed description of an environment for the teleoperation of real Lab via the Internet, was achieved however, the design cannot be replicated on robots with different platforms due to the heavy dependence on Matlab/Simulink.

 10.34257/GJCSTDVOL21IS1PG21

Abdullah & Poh, 2011; Mobile Robot Temperature Sensing Application via Bluetooth developed a Bluetooth-based control system for remote measurement of temperature from the robot's surrounding environment. The range of operation is limited to ten (10)

4 Method a) System Overview

The proposed portable IP-based remote control system for Mobile Robots is aimed at solving two basic problems usually encountered with common Mobile Robot remote control systems. The first problem is the communication range between the Mobile Robot and its remote control interface. This range limit is entirely dependent on how far the communication link can transmit or receive data sent from both ends (usually 10 Meters for Bluetooth, 100 Meters for WIFI). The second problem is the portability of control interfaces with different platforms to provide different user interfaces that can be used to control different Mobile Robots. To solve the above-mentioned problems, the new system is divided into three functional parts namely: Command hub, Control Interface, and Mobile Robot Gateway.

5 b) System Architecture

Figure 1 shows the entire architecture of the Mobile Robot Remote Controlled System. The system components run separately on different devices, performing different roles in achieving the overall goal of the system. The components are also loosely coupled to ease upgrade and maintenance without affecting the operation of the other ones.

6 i. Command Hub Design

The Command Hub is at the center of the entire Remote control system for mobile robots. It is one of the three components required to develop the full system as shown in figure **??** The control interface component in the system is the one that interacts with the user and based on the operation performed on it, it will create and send an appropriate HTTP request to the command hub respectively. The control interface performs two basic functions. The first function is to take users' actions and transform them into a valid HTTP request required by the command hub. The second function is getting telemetry data from the command hub to create visual feedback for the users. The structure of this component is therefore organized along with its functions. Two modules are provided, one to constantly fetch new or updated data from the command hub to update the telemetry view of the mobile robot's vitals while the other module handles user interactions with the control interface and makes appropriate HTTP requests to the command hub.

Different Control Interfaces are designed based on the two modules explained earlier to expand the choice of control for each Mobile Robot in the system. Available control interface includes:

Cross-Platform Mobile Interface: This type of interface is meant for users that prefer to control their Mobile robots from a smartphone. The interface wasdesigned with Apache Cordova, HTML 5, CSS 3, and JavaScript. This ensures that the resulting interface can be used on Windows Phone, Android Phone, and iPhone as well.

Web Application Interface: This type of interface is available to users that want to control their Mobile Robots through a web browser. The web browser is used to access Web applications deployed alongside the command hub to provide the control interface. This Control interface is designed with ASP.Net MVC 4, HTML 5, CSS 3, and JavaScript. A variety of web browsers are supported.

Desktop Interface: In this type of control interface, the traditional Desktop application is used to provide the interface to the users. C# programming language is used to develop this interface.

7 iii. Mobile Robot Gateway Design

This is the component that interfaces directly with the Mobile Robot. It plays a major role in connecting the robot to the remote control interface through the command hub. There are two different types of Mobile Robot Gateway namely, Internal Mobile Robot Gateway and External Mobile Robot Gateway. The internal Mobile Robot Gateway runs locally on the robot's operating system. This type of Gateway is only supported by Mobile Robots capable of running scripts, executable programs and whose hardware supports networking and direct connectivity to the Internet. The Gateway is implemented as part of the required software running locally on the robot's platform. The Gateway runs directly on the Mobile Robot's control board as shown by Robot 2 in figure 1. External Mobile Robot Gateway does not run on the robot platform, it runs on a remote computer connected to the Mobile Robot through Bluetooth or a WIFI device. This type of Gateway is meant for Mobile Robots that do not have shields or devices to directly connect to the Internet thereby using the remote computer's Internet connectivity for its operation. In the case of External Mobile Robot Gateway, a remote computer connected to the Internet is placed within the communication range of the Mobile Robot, the computer shares its internet connection with the Mobile Robot. The Gateway is also executed on the remote computer to serve as the conduit through which communication is established with the control interface via the command hub. Robot 1 in figure 1 makes use of the External Mobile Robot Gateway.

8 Gateway Operation Circle

The Mobile Robot Gateway operates in a circle of three operations namely Sense/Listen, Think, and Act. Figure 4 shows the interaction of Gateway operations in a single circle.

? Sense/Listen Operation:-in this operation, the Gateway awaits new commands from the command hub through an active Web Socket if available or through an HTTP request to the REST API exposed by the command hub. If the Mobile Robot is in autonomous or semi-autonomous mode, values from the proximity sensors are fetched by the Gateway. Any data acquired from this operation is then passed on to the THINK operation in the circle. ? Think -analysis of data passed from the Sense/Listen operation is carried out during this operation. It is mainly responsible for command interpretation and conversion into a simpler form that could be handled on the Mobile Robot easily.

After the interpretation and conversion task has been carried out, the result is sent to the ACT operation to take necessary action. ? Act: -this part is responsible for generating the control sequence based on the input received from the THINK operation. It determines the destination of the resulting control sequence (i.e. Mobile Robot or Command hub). The control sequence now determines the behavior the Mobile Robot will exhibit. This operation is also responsible for sending telemetry data to the command hub. After this operation is carried out, control is returned to the Sense/Listen operation iv. Command Design One of the roles of the Command Hub is the generation of commands to be sent to the Mobile Robot. These commands are generated based on the interaction of the users with the control Interface. Each action performed by the user on the control interface corresponds to a specific command meant for the Mobile Robot to execute. When an action is performed on the control interface, a request is sent to the REST API (Application Programming Interface) exposed by the Command Hub. The Command Hub now maps the request to the corresponding command to be generated and sent to the Mobile Robot. For example, if the user presses the "Move Forward" Button on the Control Interface, an HTTP POST request is sent to http://Command-Hub-DomainName/api/move/1, then "f" command is generated and sent to the Mobile Robot. The "Command-Hub-DomainName" is the domain name of the server hosting the Command Hub. Table 1 shows the comprehensive list of all Uri, HTTP verbs, and associated commands mapped to them.

9 Global Journal of Computer Science and Technology

Volume XXI Issue I Version I At the hardware level, two robots were designed to validate the workability of the remote control system whose architecture was shown in figure 1. These Mobile Robots have different platforms and configurations each of which corresponds to the two types of Mobile Robots in figure 1. The Mobile Robots are: SleekBot V1: This was designed to use Arduino Nano R3 as its main control board. All other actuators and sensors are connected to the main control board. In this configuration, a Bluetooth module is used to communicate with an external Mobile Robot Gateway.

Figure 6 shows the breadboard schematics of the robot's hardware configuration.

SleekBot V2: This is the second Mobile Robot, it was designed to use Raspberry Pi 2 Model B as its main control board while Arduino Nano R3 was used as a slave control board. In this configuration, a USB camera with a two degree of freedom is connected to the mainboard and Internal Mobile Robot Gateway was used because the main control board is capable of running executable programs and it is also capable of connecting to the internet directly. Figure **??** shows the breadboard schematics of SleekBot V2.

10 Result and Discussion

11 a) System Implementation

The system was implemented in three stages, each stage corresponds to the deployment of each component that made up the system. For web interface users, the system provides a landing page where navigation to other modules is accessed. Figure 8 shows the home page layout.

12 i. Mobile Robot Gateway Stage

Two types of Mobile Robot Gateway were implemented. The specific implementation used was determined by the robot's ability to connect directly to the internet. In a scenario where the Mobile Robot can connect directly to the internet, a python program was used to implement the Gateway. The second implementation was used when the Mobile Robot connects to the internet through another computer. A desktop application written in C# was used to implement the Gateway on the computer with internet connectivity.

13 ii. Command Hub Stage

At this stage of implementation, a web API written in C# was used to implement a REST-Based service that acts as the command hub for the system. The service was deployed to Internet Information Service (IIS) with MS SQL Server 2012 to act as the backend storage for the system.

14 User Registration

This module is used to register all users making use of the control platform. All valid users of the system need to be authenticated before they are granted access to control any mobile robot remotely. Figure ?? shows the user registration interface for web interface users while figure 10 shows the registration interface for mobile users.

15 User Login

This module is used for authenticating all users making use of the system. All users must pass through this module to use the system. Figure 11 and figure 12 show the login user interface (UI) for web and mobile users.

16 Main Menu

The main menu provides navigation to important modules the user can use to perform different tasks in the system. Figure 13 shows the main menu for mobile interface users

17 Mobile Robot Registration

To make use of any robot on the platform, the user needs to register the robot. Figure 14 shows the interface dedicated to the registration. After registration, all registered Mobile Robots by the user is displayed in the Mobile Robot List as shown in figure 15.

18 Settings

The settings module is used to configure the parameters required for the interface to work properly. Figure 16 shows a settings Interface for mobile Users

19 Remote Control Interface

This interface provides the necessary widget to remotely control mobile robots. Actions performed on this interface translate to a command to be executed on the mobile robot. Figure 17 and figure 18 show the remote control interface for web and mobile interface users.

20 Conclusion

The goal of this research work to develop a portable IP-based multi-interface remote-controlled system for mobile robots was achieved. The system offers the use of a single remote control device across different mobile robots and the use of a multi-interface for a single mobile robot.

The solution architecture is based on three loosely coupled components performing various tasks at different levels to collectively achieve a single aim of having a portable control system. At the core of the system is a REST-based web service handling communication between the user interface and the mobile robot. Various programming languages were used at different levels to achieve the overall goal of the system. A cross-platform mobile application, Web application, and the desktop client was developed to serve as the system's user interface. Implementation of the system was carried out on three different mobile robots based on different platforms. The system was evaluated based on the time complexity of the algorithm used in its components. The result shows that the system time of execution is linear (O (n)).

The system developed is not without its weakness, hence the need to improve some parts that are currently inefficient in its mode of operation. These include:

1

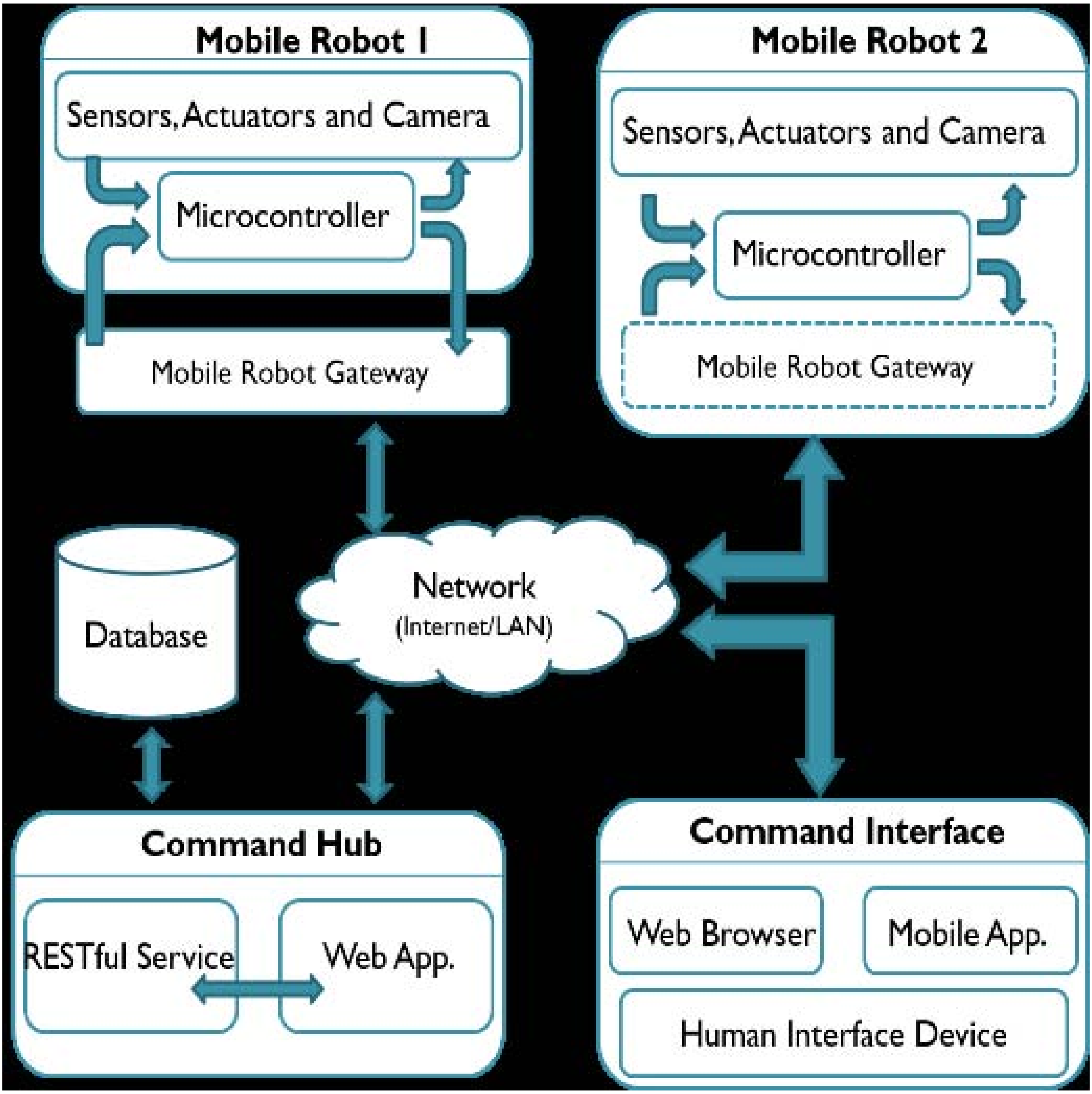


Figure 1:

10.34257/GJCSTDVOL21IS1PG21

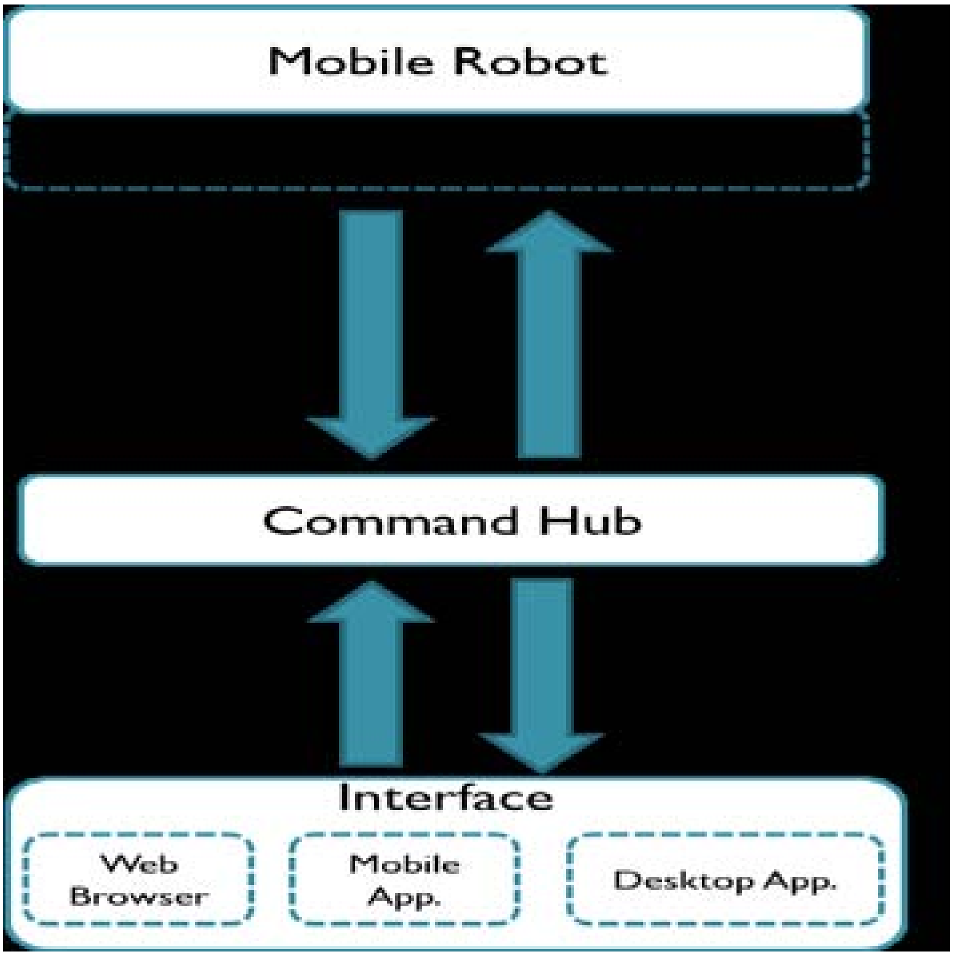

Mobile Robot
Command Hub
Interface
Web Browser
Mobile App.
Desktop App.

Figure 2:

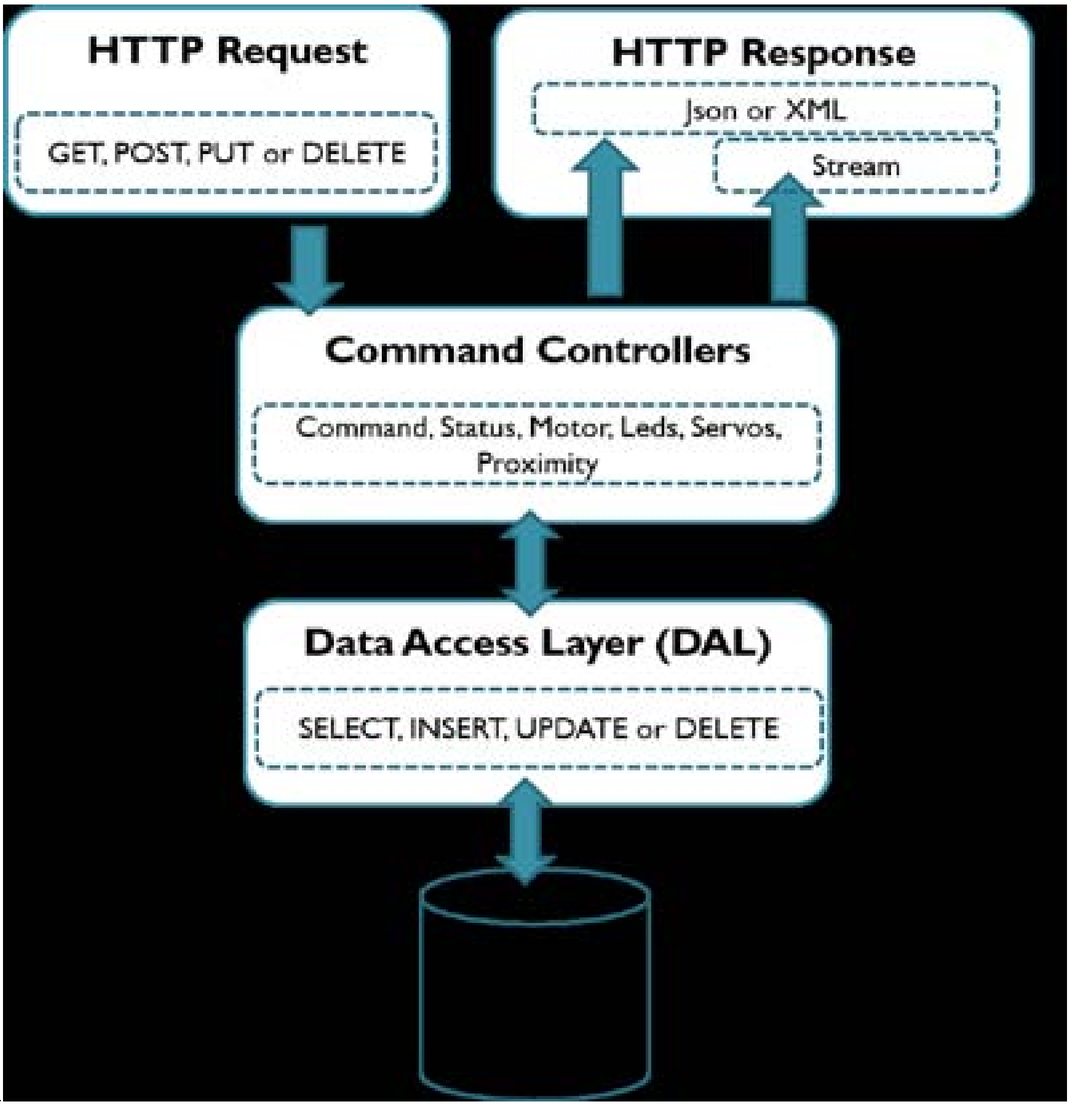
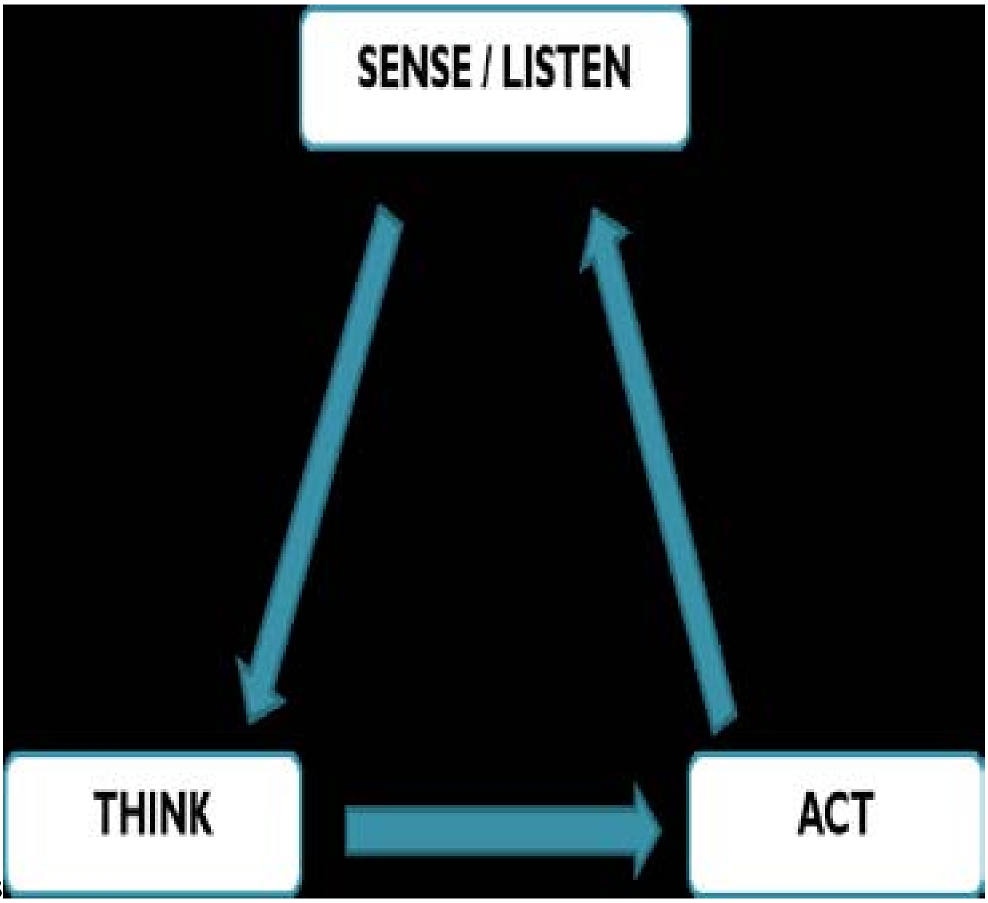
1 

Figure 3: Figure 1 :

 10.34257/GJCSTDVOL21IS1PG21



3

Figure 4: Figure 3 :

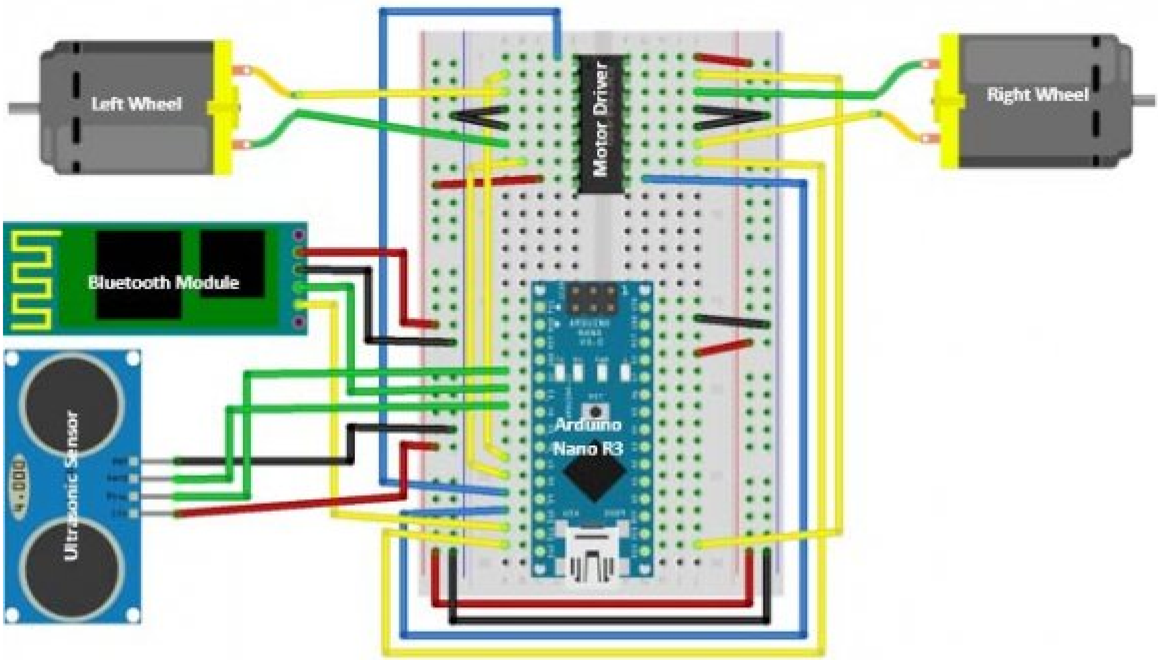
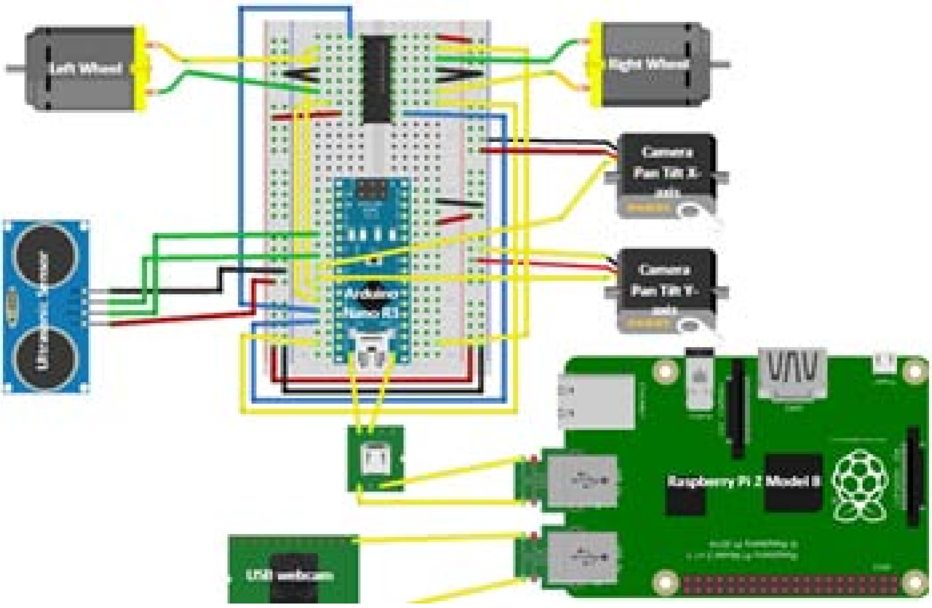


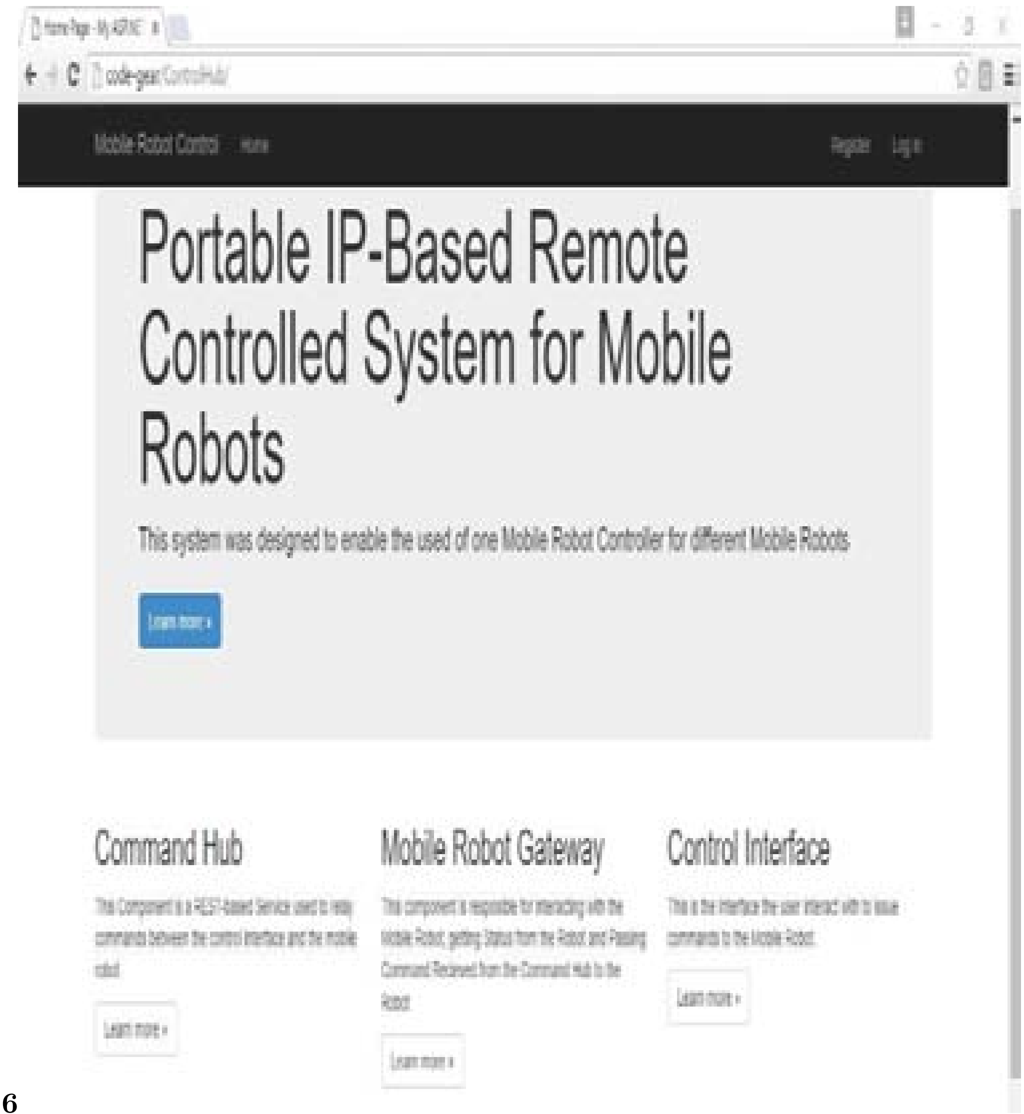
Figure 5:



4

Figure 6: Figure 4 :

 10.34257/GJCSTDVOL21IS1PG21



6

Figure 7: Figure 6 :

8

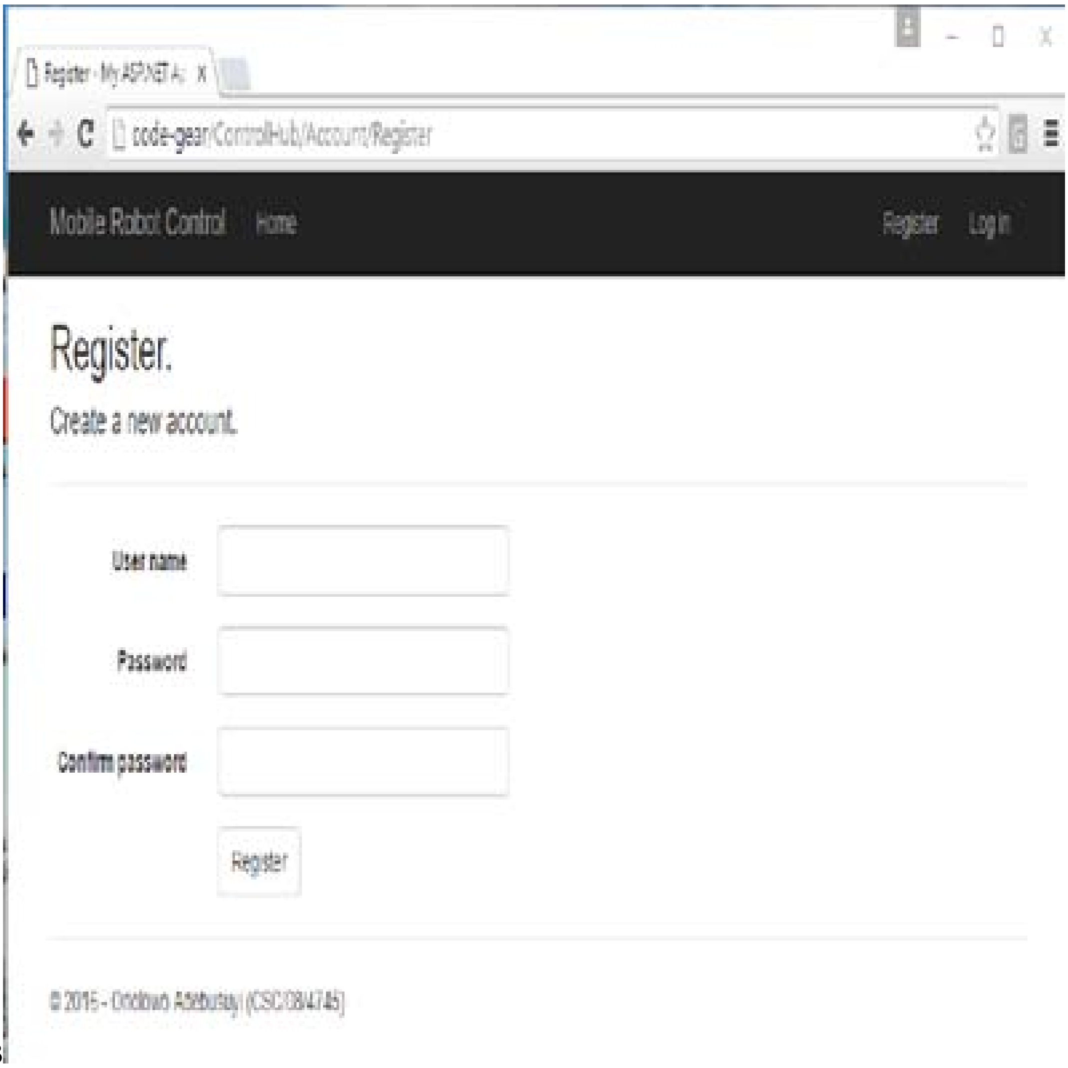


Figure 8: Figure 8 :

10.34257/GJCSTDVOL21IS1PG21

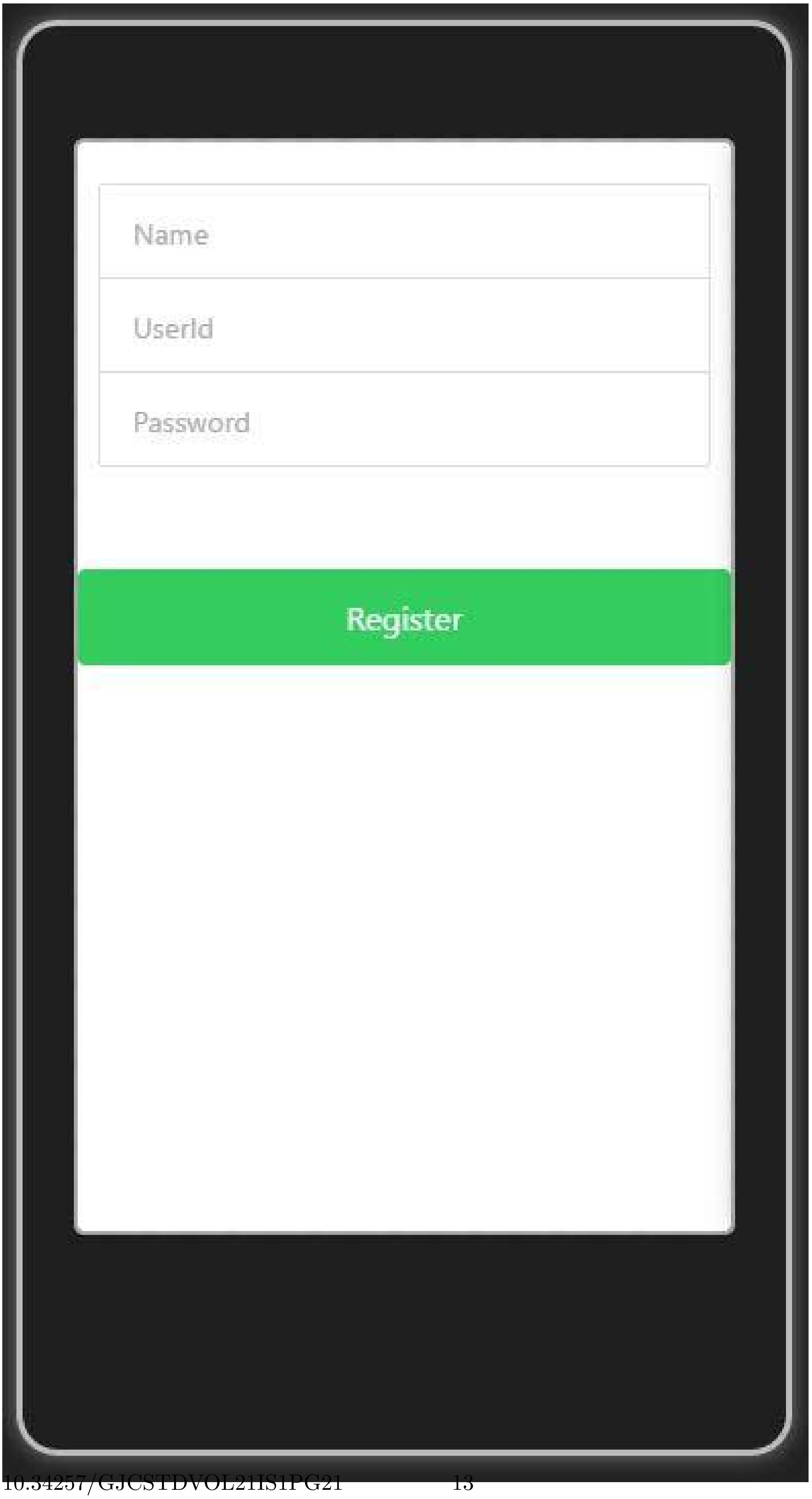
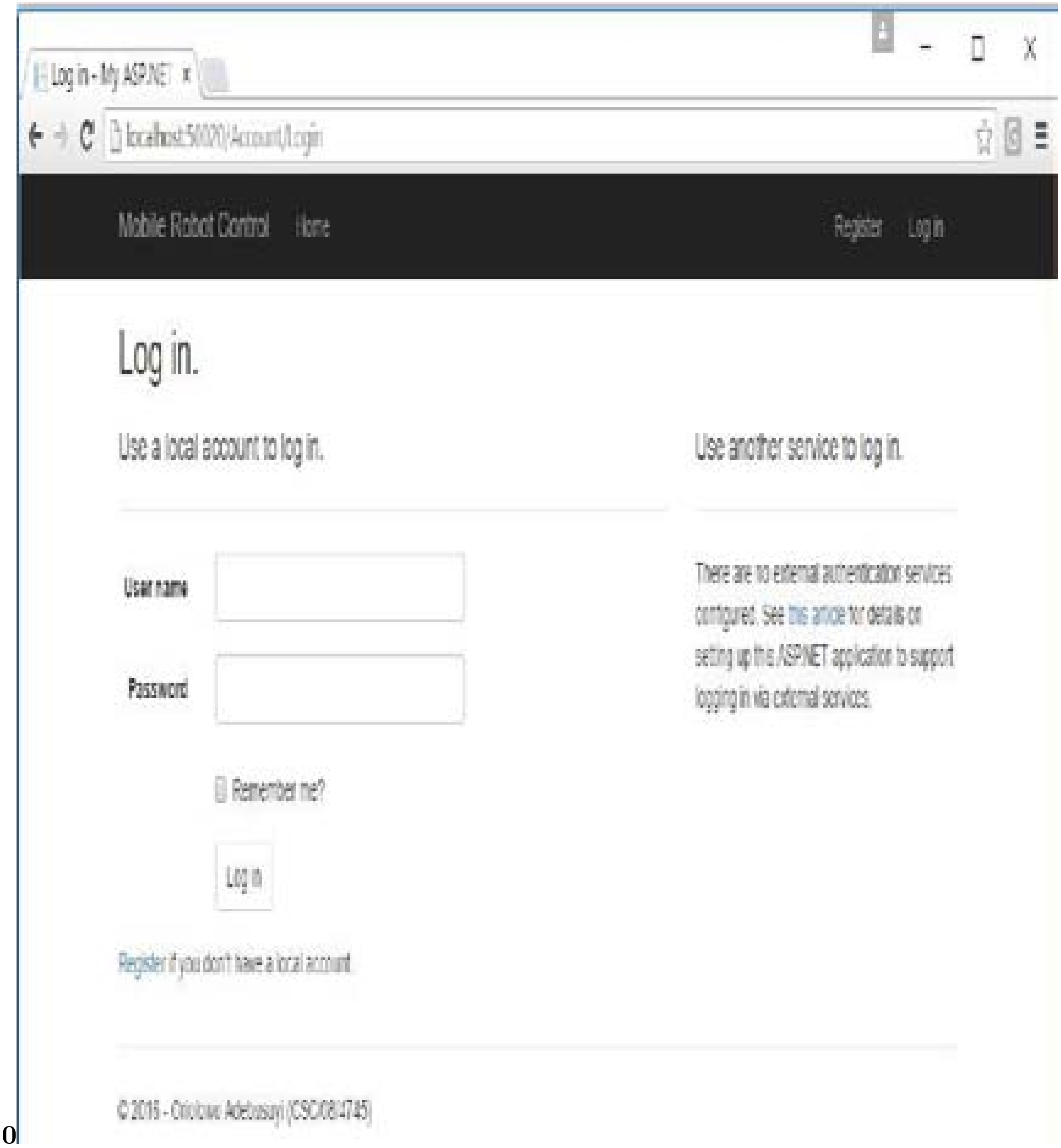
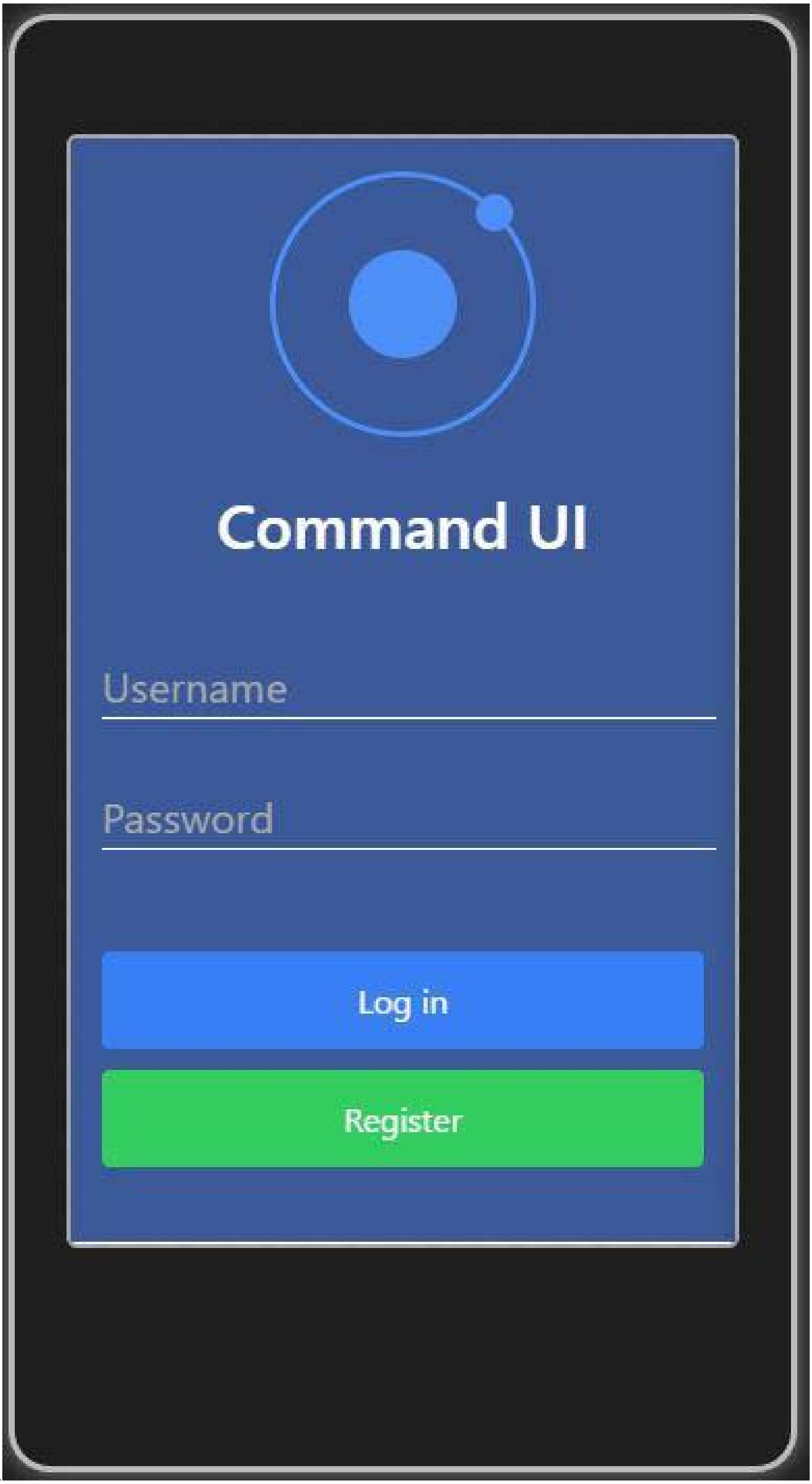


Figure 9:



10

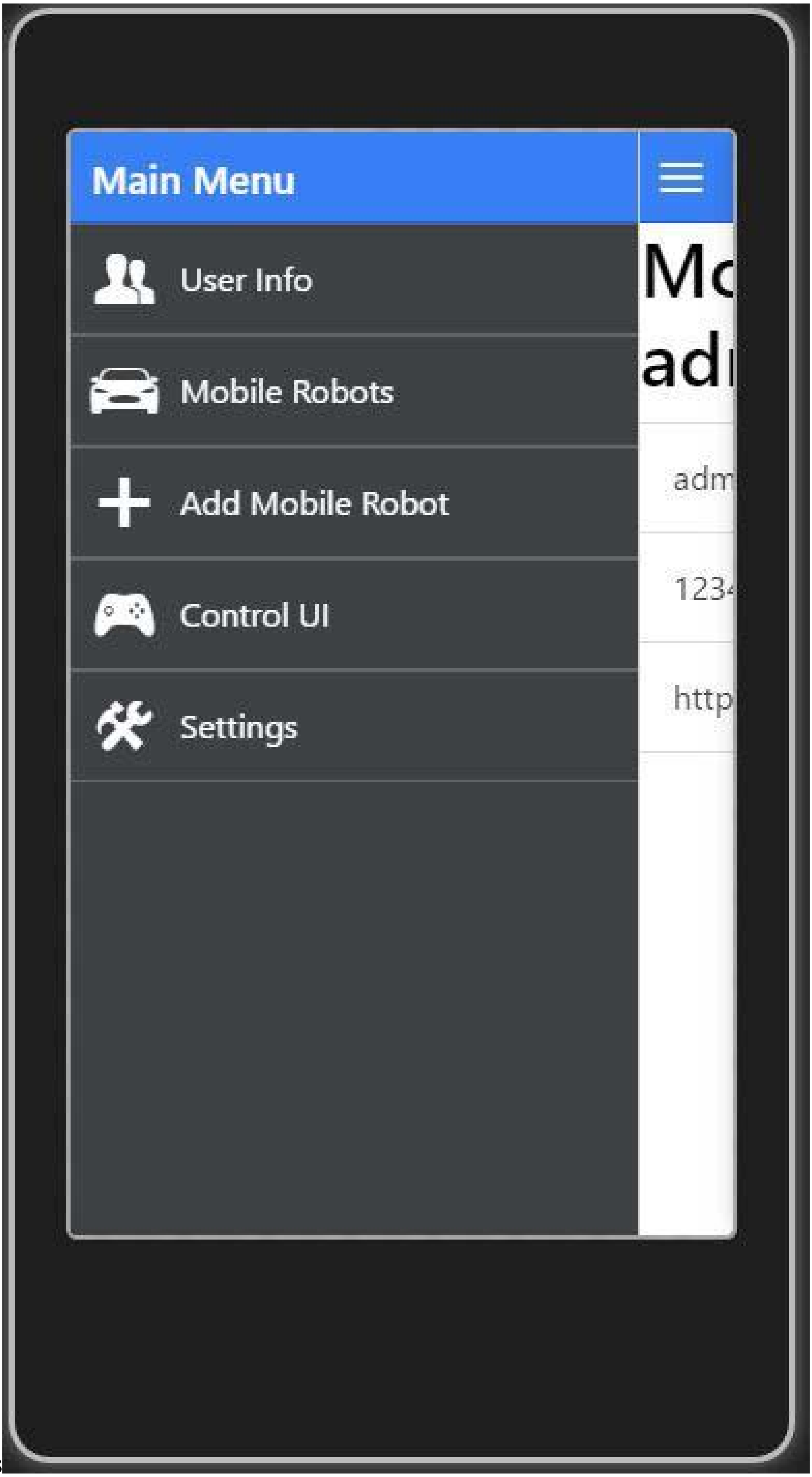
Figure 10: Figure 10 :



11

10.34257/GJCSTDVOL21IS1PG21

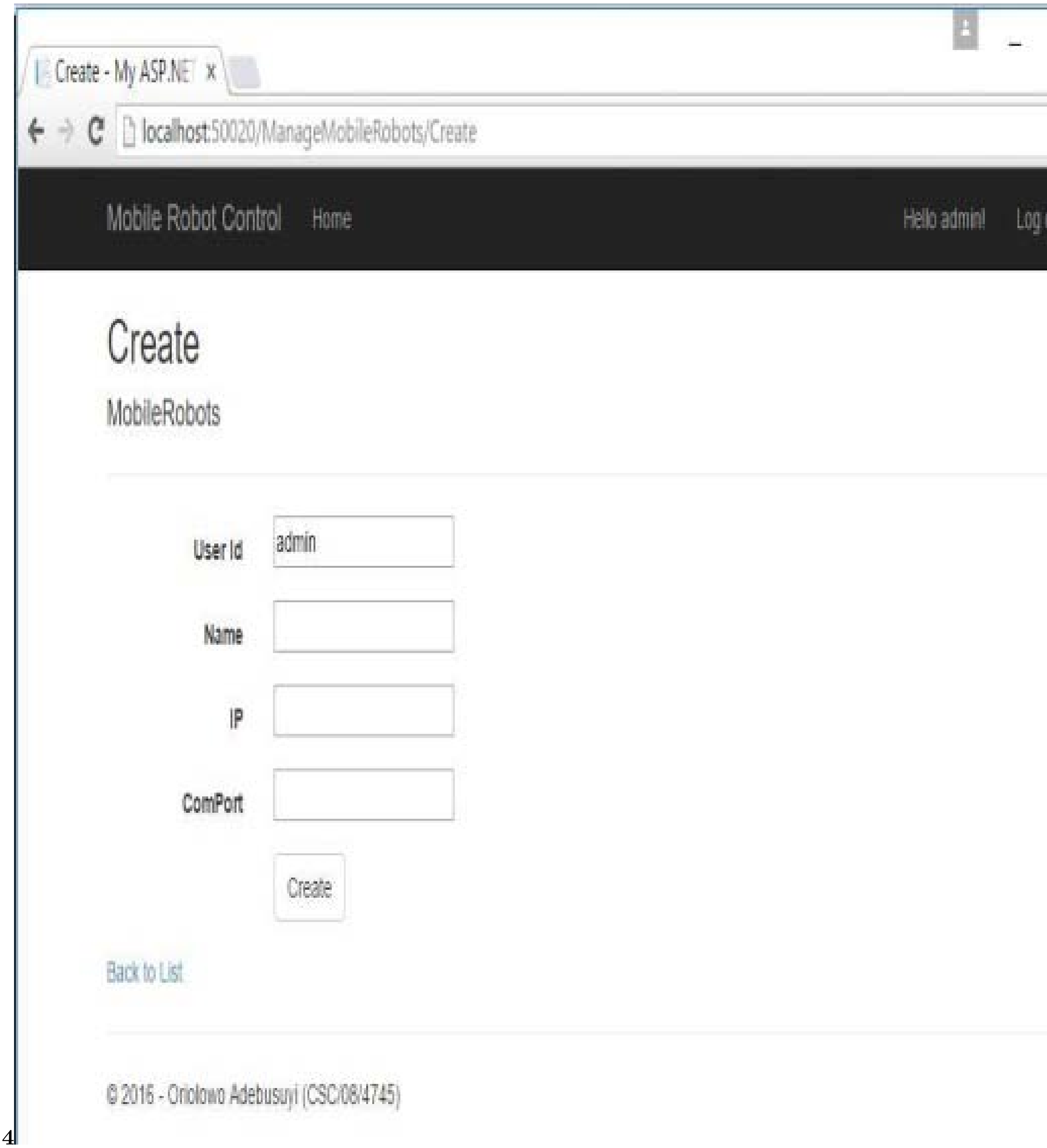
Figure 11: Figure 11 :



13

Figure 12: [16]Figure 13 :

10.34257/GJCSTDVOL21IS1PG21



14

Figure 13: Figure 14 :

15

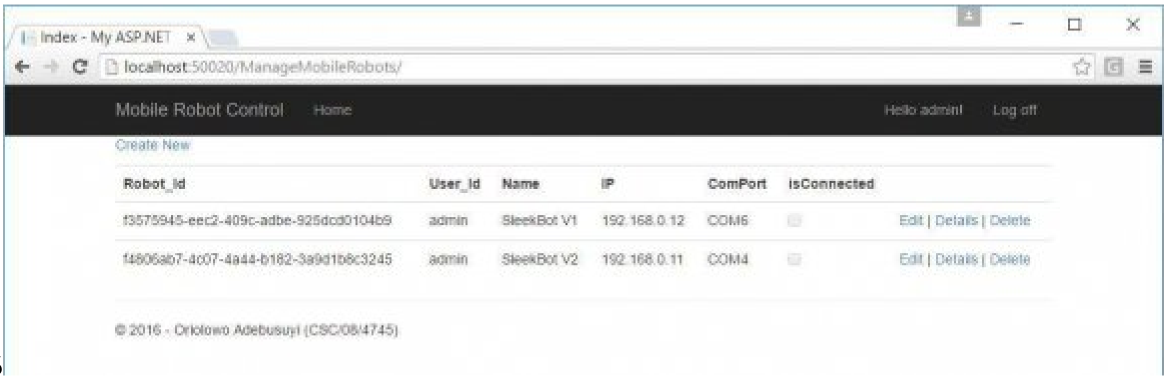
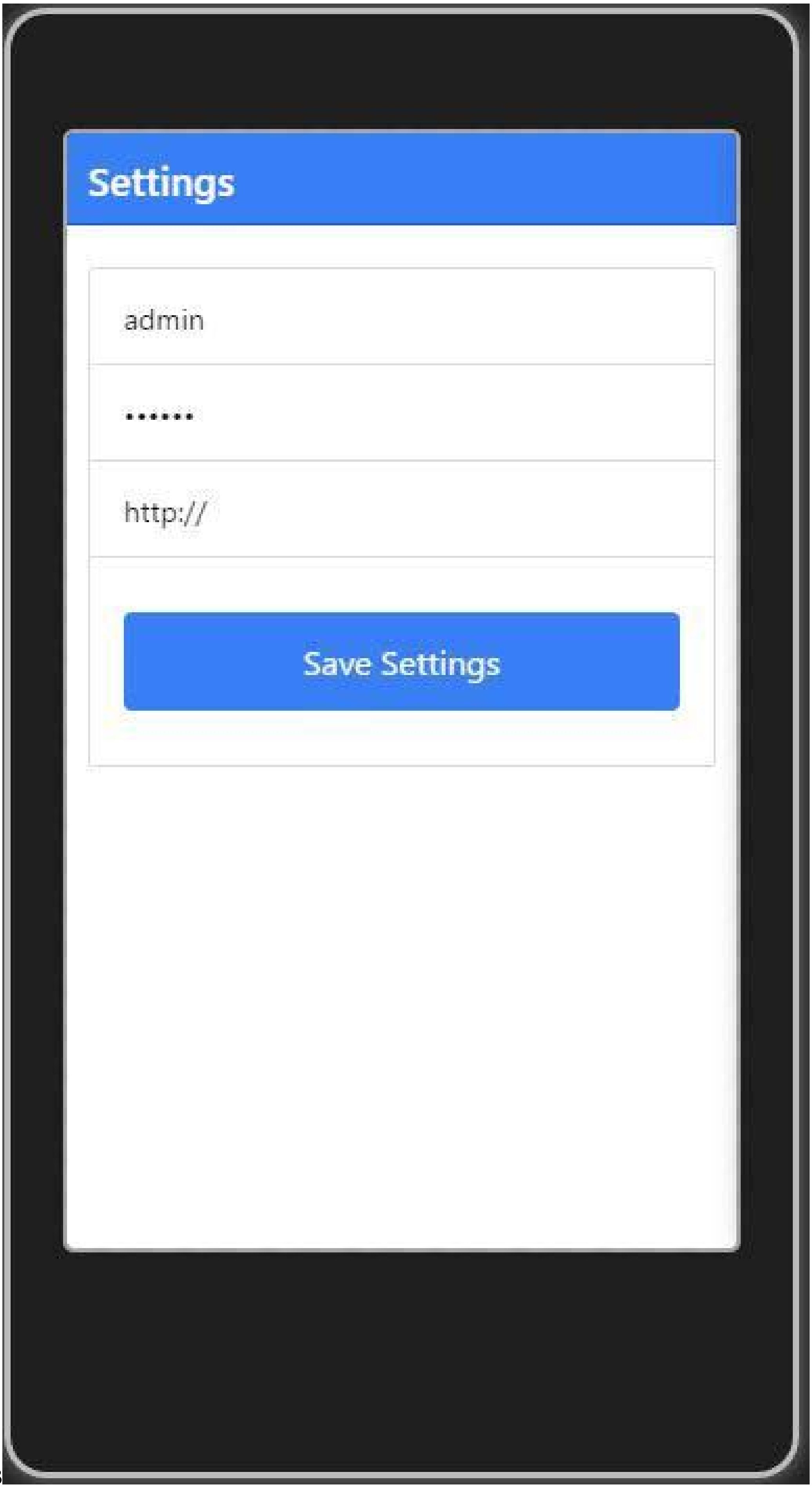


Figure 14: Figure 15 :



10.34257/GJCSTDVOL21IS1PG21

Figure 15: Figure 17 :Figure 18 :

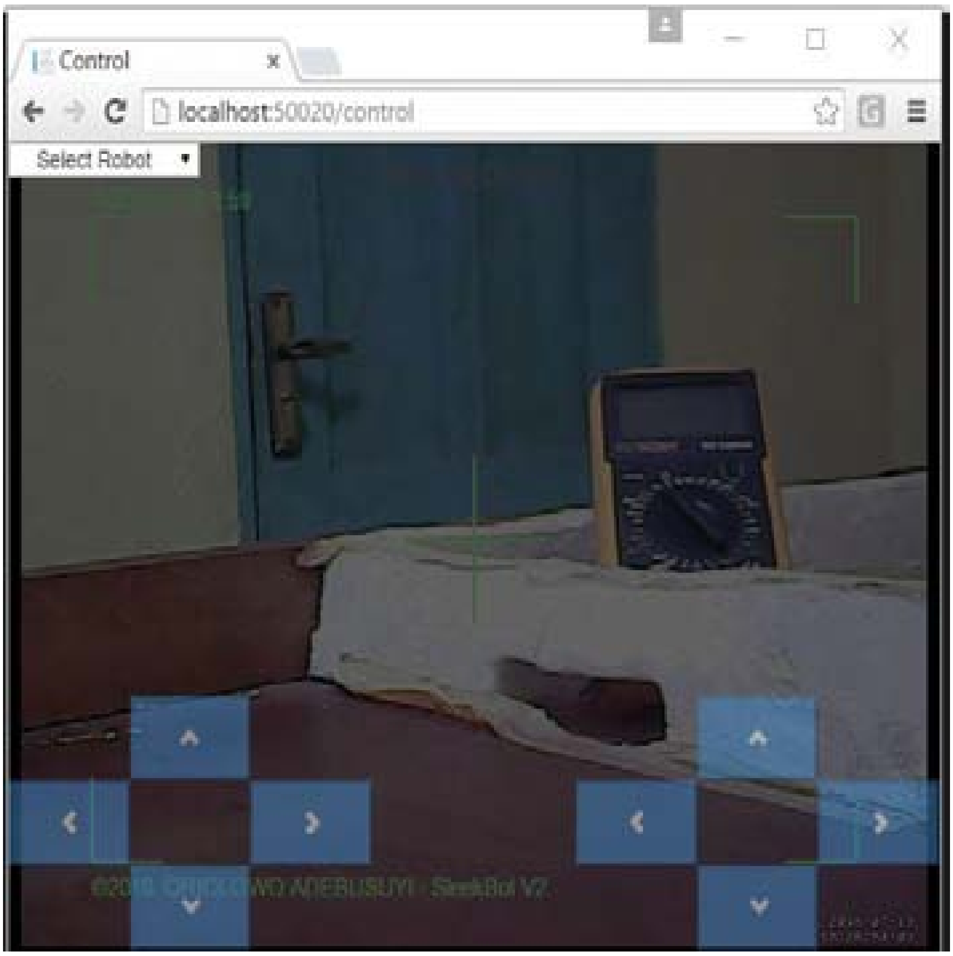


Figure 16:

1

URI	HTTP VERB	COMMAND	DESCRIPTION
/api/move	GET	W	Get Wheels State
/api/move/1	POST	F	Move forward
/api/move/2	POST	b	Move Backward
/api/move/3	POST	r	Move Right
/api/move/4	POST	l	Move Left
/api/move/5	POST	s	Stop
/api/speed	GET	v	Get Robot Speed
/api/speed/value	POST	v,value	Set Robot Speed to value
/api/leds	GET	d	Get all LED Status
/api/leds/id	GET	d,id	Get the Status of LED id
/api/leds/id/status	POST	d,id,status	Set the state of LED id with status
/api/servo	GET	c	Get Status of all Servo
/api/servo/id	GET	c,id	Get state of Servo id
/api/servo/id/pos	POST	c,id,pos	Set the position of Servo id to position
/api/motor	GET	m	Get the state of all Motor
/api/motor/id	GET	m,id	Get state of motor id
/api/motor/id/pwm	POST	m,id,PWM	Set the PWM signal of Motor id
/api/prox	GET	p	Get state of all proximity sensor
/api/prox/id	GET	p, id	Get state of proximity sensor id

[Note: v. Mobile Robot Design and Configuration]

Figure 17: Table 1 :

Raspberry Pi 2 Model B: this is a credit card
size computer capable of running a trim-down version of
the Linux operating system. It has a quad-core
processor with a clock speed of 900 MHz, 1 GB RAM,
and 40 GPIO (General-purpose Input Output) pins. This
was used with Arduino Nano R3 to develop SleekBot V2
Mobile Robot
ePuck Mobile Robot: this a differential drive
educational mobile robot based on a dsPIC30F6014A
microcontroller chip running at 60 MHz clock speed. It
has 8 KB RAM, 144 KB flash Memory, 3D
accelerometer, VGA camera, 8 infra-red sensors,
Bluetooth for wireless communication with a computer,
speaker and LED. It was used as one of the Mobile
Robots used in testing the workability of the developed
remote control system.
Software
Programming Languages: C, C#, Python, JavaScript
Interface Design: HTML 5, CSS 3, Win form
Backend: MS SQL Server 2012
Framework: .Net Framework, Apache Cordova, Ionic
Framework
Web Server: Internet Information Service (IIS) 8
ii. User Interface Documentation
Home Page

[Note: b) Implementation Toolsi.]

Figure 18:

Get Command from Command Hub
If Command != null
Map command to the Mobile Robot Kinematics
Execute Command
End If
End While
While Interrupt NOT available
Get Mobile Robot Status
Send Status to Command Hub
End While
Time Complexity of Mobile Robot Gateway algorithm
Number of Operation
c) System Evaluation
Algorithm for Command Hub
Initialize Command Hub
While interrupt NOT available
Get HTTP request from client
Map request to command controller
Construct response message
Return response message
End While
Time Complexity of Command Hub algorithm
Number of Operation $= 1 + 1 + N(1 + 1 + 1 + 1)$
$= 2 + N(4)$
$= 2 + 4N$
Number of Operation $= 2 + 4N$
In Big O notation, the algorithm executes in linear time i.eO(n)
Algorithm for Command Interface
Initialize Command Interface
While UserAction != null
Construct command object
Send command object to Command Hub using HTTP Request
Get Response for the HTTP Request
If Response != null
Process Response
Update Interface
End If
End While
While Interrupt NOT available
Get Status from Command Hub
Update Interface
End While
Time Complexity of Command Interface algorithm
Number of Operation $= 1 + 1 + N(1 + 1 + 1 + 1 + 1) + N(1 + 1)$

Figure 20:

10.34257/GJCSTDVOL21IS1PG21

[). remote controldefinition of remote control in English from the Oxford dictionary Oxford dictionary (2015)] '). remote controldefinition of remote control in English from the Oxford dictionary'. http://www.oxforddictionaries.com/definition/english/remote-control *Oxford dictionary*, 2015. June 13.

[Oboe and Fiorini ()] 'A design and control environment for internet-based telerobotics'. R Oboe , P Fiorini . *International Journal of robotics research* 1998. 17 (4) p. .

[Jose et al. ()] 'A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum'. S Jose , D Sebastián , P Rafael , M Fernando . *IEEE transactions on education* 2004. 47.

[Royakkers and Van-Est ()] 'A literature review on new robotics: automation from love to war'. L Royakkers , R Van-Est . *International journal of social robotics* 2015. 7 (5) p. .

[Lambèr and Rinie ()] 'A Literature Review on New Robotics: Automation from Love to War'. R Lambèr , V E Rinie . *International Journal of Social Robotics* 2015. 2015. p. 295.

[Chin et al. (2003)] 'A Real-Time Remote Control Architecture Using Mobile Communication'. E L Chin , L Chih-Ching , H An-Sang , W Chih-Chen . *IEEE transactions on instrumentation and measurement* 2003. August 2003. 52 (4) .

[Gates ()] 'A robot in every home. The leader of the PC revolution predicts that the next hot field will be robotics'. B Gates . *Sci Am* 2007. 296 p. .

[Pravin and Shalini ()] 'Accelerometer Based Hand Gesture Controlled Robot'. V Pravin , T Shalini . *International Journal of Science and Research (IJSR)* 2013. p. .

[Obe and Dumitrache ()] 'Adaptive Neuro-Fuzzy Controller with Genetic Training for Mobile Robot Control'. O O Obe , I Dumitrache . *International Journal of Computer Communications and Control* 2012. 7 (1) p. .

[Posadas et al. ()] 'Agent-based distributed architecture for mobile robot control'. J L Posadas , J L Poza , J E Simo , G Benet , F Blanes . *Engineering Applications of Artificial Intelligence* 2008. 2008. 21 p. .

[Alliance for Telecommunications Solutions ATIS ()] 'Alliance for Telecommunications Solutions'. http://www.atis.org/tg2k/_bot.html *ATIS* 2001. (Telecom glossary "bot)

[Turkle ()] *Alone together. Basic Books*, S Turkle . 2011. New York, Why.

[Lung et al. ()] 'An Experiment in Internet-Based'. N Lung , S N Wyatt , L Vincenzo . *Proceedings. ICRA'02. IEEE International Conference*, (ICRA'02. IEEE International Conference) 2002. 2002. 2 p. .

[James ()] *Bringing Portability to the Software Process*, D M James . 1997. West Virginia University, Dept. of Statistics and Computer Science

[Ankit et al. (2014)] 'Controlling of Remote Robot through mobile phone using DTMF Signal'. J Ankit , Y Hemant , K Y Raj , S Rajendra . *International Journal of Science, Engineering and Technology Research* 2014. April 2014. 3 (4) .

[Dutta and Zielinska ()] V Dutta , T Zielinska . arXiv:1505.07593. *Networking technologies for robotic applications*, 2015. (arXiv preprint)

[Hockstein et al. ()] 'History of robots: from science fiction to surgical robotics'. N G Hockstein , C G Gourin , R A Faust , D J Terris . *Journal of robotic surgery* 2007. 1 (2) p. .

[Harris ()] *How Robots Work. How Stuff Works*, T Harris . http://science.howstuffworks.com/robot.htm 2007.

[Goodrich and Schultz ()] 'Humanrobot interaction: a survey'. M A Goodrich , A C Schultz . *Foundations and Trends in Human-Computer Interaction* 2007. 1 (3) p. .

[Luimula et al. ()] 'Improving the Remote Control of a Mobile Robot Using Positioning and Ubiquitous Techniques'. M Luimula , K Saaskilahti , T Partala , S Pieska , J Alaspaa , A Lof . 10.1109/COASE.2007.4341744. *2007 IEEE International Conference on Automation Science and Engineering*, (Scottsdale, AZ) 2007. 2007. p. .

[Robotiq ()] *Industrial Robots: 5 Most Popular Applications*, Robotiq . http://blog.robotiq.com/bid/52886/Industrial-robots-5-most-popular-applications 2014.

[Siegwart and Saucy (1999)] 'Interacting with mobile robots on the web'. R Siegwart , P Saucy . *IEEE International Conference on Robotics and Automation (ICRA)*, 1999. May.

[Bekker ()] *Introduction to Terrain Vehicle Systems*, M G Bekker . 1969. Ann Arbor: University of Michigan Press.

[Sparrow (2007)] 'Killer robots'. R Sparrow . http://www.pantheon.org/articles/t/talos.html *Talos Encyclopedia Mythica* 2007. April 29. 2016. 24 (1) p. . (J ApplPhilos)

[Robin ()] *Massachusetts Institute of Technology*, R M Robin . 2000. (Introduction to AI Robotics)

[Abdullah and Poh (2011)] 'Mobile Robot Temperature Sensing Application via Bluetooth'. M F L Abdullah , L M Poh . *International Journal of Smart Home* 2011. July 2011. 5 (3) .

[Oboe and Fiorini ()] R Oboe , P Fiorini . *Internet-Based Telerobotics Problems and Approaches: Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference*, 1997.

[Bekker ()] *Off-The-Road Locomotion*, M G Bekker . 1960. Ann Arbor: University of Michigan Press.

[Esteller-Curto et al. ()] 'Proposal of a REST-based architecture server to control a robot'. R Esteller-Curto , E Cervera , A P Del Pobil , R Marin . *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012. 2012. p. . (Sixth International Conference)

[Juang and juang ()] 'Remote control of a mobile robot for indoor patrol'. S Y Juang , J G &juang . 10.3390/app6030082. https://doi.org/10.3390/app6030082 *Applied Sciences* 2016. 6 (3) p. 82.

[Polk ()] *RoboNexus 2005 robot exhibition virtual tour. Robonexus Exhibition*, I Polk . http://www.virtuar.com/click/2005/robonexus/index.htm 2005. 2005.

[Dictionary ()] *Robot -definition of a robot in English from the Oxford dictionary*, Oxford Dictionary . http://www.oxforddictionaries.com/definition/english/robot 2016.

[Nicholas ()] *Robots in space*, W Nicholas . http://www.universetoday.com/43750/robots-in-space 2009.

[Lydia et al. ()] *TCP/IP Tutorial and Technical Overview. International Technical Support Organization*, P Lydia , T B David , D Chuck , F Jason , L Wei , M Carolyn , R Nicolas . 2006.

[Chikurtev et al. ()] 'The communication system for remote control of service robots'. D Chikurtev , I Rangelov , K Yovchev , N &chivarov . 10.1016/j.ifacol.2019.12.470. https://doi.org/10.1016/j.ifacol.2019.12.470 *IFAC-PapersOnLine* 2019. 52 (25) p. .

[Wichmann et al. ()] 'The integration of mobile (tele) robotics and wireless sensor networks-A survey'. A Wichmann , B D Okkalioglu , T Korkmaz . *Computer Communications* 2014. 51 p. .

[Adamides et al. ()] 'User interface considerations for telerobotics: The case of an agricultural robot sprayer'. G Adamides , C Katsanos , G Christou , M Xenos , G Papadavid , T Hadzilacos . *Second International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2014) (pp. 92291W-92291W). International Society for Optics and Photonics*, 2014.

[Zunt ()] *Who did actually invent the word "robot" and what does it mean? The Karel Capek website*, D Zunt . http://capek.misto.cz/english/robot.html 2007.

[Almali et al. ()] 'Wireless remote control of a mobile robot'. M N Almali , K Gürçam , A Bayram . *International Journal of Scientific Research in Information Systems and Engineering* 2015. IJSRISE.