

Ministry of Education and Science of Ukraine
Ternopil Ivan Pul'uj National Technical University

(full name of higher education institution)

Faculty of Computer Information Systems and Software Engineering

(faculty name)

Computer Systems and Networks Department

(full name of department)

EXPLANATORY NOTE

for diploma project (thesis)

bachelor

(educational-proficiency level)

topic: A Computer System for Home Automation with Java Technology
and Raspberry Pi

(Комп'ютерна система для автоматизації будинку з використанням
технології Java та Raspberry Pi)

Submitted by: fourth year student group ICI-42

Specialism (field of study) _____

123 Computer Engineering

(code and name of specialism (field of study))

(signature)

Ugwu V.O.

(surname and initials)

Supervisor

(signature)

Lutskiv A.M.

(surname and initials)

Compliance check

(signature)

Stadnyk N.B.

(surname and initials)

Referee

(signature)

(surname and initials)

ABSTRACT

A computer system for monitoring and managing smart homes // Diploma thesis // Ugwu valentine Okechukwu// Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, group ICI-42 // Ternopil, 2020 // p. – , fig. - 27, tabl.– 1, code snip. –7, append.1 – , bibliogr. – 14.

Keywords: JAVA, SPRING-BOOT, IoT, AUTOMATION, RASPBERRY PI, FIREBASE, SENSORS.

This thesis is on smart home automation, a raspberry pi will be used to create a model of a smart home and demonstrate the basics how home automation is made possible.

For this project i have selected Java for the backend as it is one of the leading programming languages for the Internet of Things. the user interface is designed with React framework. The purpose of separating the frontend from the backend is to demonstrate how applications can communicate (API), because this is also integral in understanding how things communicate.

CONTENTS

INTRODUCTION.....	5
1 MAIN PART.....	7
1.1 Hardware and software components justification.....	7
1.1.1 IoT parts justification.....	7
1.1.2 Software justification.....	9
1.1.2.1 Java Programming language.....	10
1.1.2.2 The Spring Framework.....	12
1.1.2.3 PI4J library.....	17
1.1.2.4 SQLite database.....	18
1.1.2.5 React Framework.....	19
1.1.3 Cloud services justification.....	22
1.1.4 Network protocols justification:.....	25
1.1.4.1 Message Queuing Telemetry Transport protocol.....	25
1.1.4.2 REST over HTTP or HTTPs protocol.....	26
1.1.4.3 GraphQL query language communication approach.....	29
1.1.4.4 gRPC protocol.....	30
1.1.4.5 Protobuf protocol.....	31
1.1.5 Cloud native approach.....	31
1.2 Hardware components implementation.....	32
1.3 Software components implementation.....	34
1.3.1 Back-end development.....	34
1.3.2 Testing the REST API through Postman.....	37
1.3.3 Front-end development.....	39
1.3.4 Developed methods.....	40
2 SPECIAL PART.....	42
2.1 Testing layers of application.....	42

CSDP 123.061.00.00 EN				
<i>Ch.</i>	<i>List.</i>	<i>№ docum..</i>	<i>Sign</i>	<i>Date</i>
		<i>Ugwu V.O.</i>		
	<i>Supervisor</i>	<i>Lutskiv A.M.</i>		
	<i>Reviewer</i>			
	<i>N. Contr.</i>			
	<i>H. of Dept</i>	<i>Osukhivska H.</i>		
<i>A Computer System for Home Automation with Java Technology and Raspberry Pi</i>				
		<i>Letter</i>	<i>Page</i>	<i>Pages</i>
		3		
<i>TNTU, dept.CE, ICI-42</i>				

2.2 Demonstrating the project on a local network.....	43
2.2 Software deployment.....	49
2.2.1 Back-end deployment.....	49
2.2.2 Front-end deployment.....	50
3 TECHNICAL - ECONOMIC ASSESSMENT.....	52
4 OCCUPATIONAL SAFETY AND HEALTH.....	60
CONCLUSIONS.....	70
REFERENCES.....	71

INTRODUCTION

A smart home is a house that has the capacity of utilizing web associated gadgets to guarantee/give remote management and monitoring. Any gadget in the house that utilizes power and web can be associated with your home system and at command. which can be provided by remote control, a cell phone or even by voice. The home responds. The majority of these applications identify with home security, indoor regulator guideline, home theater and entertainment.

The rise of this new innovation is because of the enormous and stunning accomplishment of cell phones and tablet PCs. These ultra-versatile PCs are all over the place and their consistent associations with the web implies they can be designed to control a great deal of other online gadgets. It's everything about the Internet of Things.

The Internet of Things is an expression, it alludes to the products and items that are interconnected and recognizable through computerized systems, permitting us to either control or get information from these objects(things). This web-like spread of items is showing signs of improvement consistently. The entirety of the hardware in your house are reasonable game for this tech upheaval, from your heater to your fridge.

IoT helps us work much more intelligent and quicker, live more astute, and oversee our lives, yet it additionally helps in our prosperity in the behind the scenes. Notwithstanding our shrewd home gadgets, IoT is undoubtedly a fundamental technology in business and industry, as it gives organizations an ongoing look into the inward activities of their organization's frameworks. From the processing plant floor to the client's entryway, IoT conveys ongoing experiences into everything from machine conditions and execution to supply chain and logistics operations.

IoT empowers organizations to mechanize procedures and set aside much cash on work. It additionally diminishes squander and improves administration conveyance, making it

less expensive to manufacture and deliver merchandise and give real straightforwardness into client exchanges. IoT permits organizations to diminish

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		<i>5</i>

absolute expenses, guarantee wellbeing, and increment quality from start to finish, which means a win win for all individuals involved. Thus, customer merchandise are more affordable to create, dispatching is progressively unsurprising, and organizations can scale quicker, invigorating our economy while conveying a truly necessary feeling of fulfillment we can count on.

More astute Catastrophic event management, the capacity to anticipate, with staggering exactness, beginning of conditions that promote forest fires before they gain out of power or before the even begin, permit regulation groups to react faster, while empowering first responders to carry out targeted evacuations better and more aligned to the real situation in the field. This idea may likewise apply additionally to the more brilliant flood recognition or improved and quicker response to landslides, seismic tremors, torrential slides, and other catastrophic events.

More astute Social insurance, wearable gadgets that distinguish a lot of medical issues, in all probability before they even happen, and quickly administer life-sparing medications or send, as quickly as time permits, emergency personnels with detailed information put in their grasp or alert relatives depending upon the circumstance.

The ideas above barely cover the surface of the magnitude of possibilities afforded by the dawn IoT age, and they all offer a typical and unquestionably not incidental descriptive word here: "smarter". In the event that "smarter" is characterized by the combination of access to data and the capacity to place that data to use in important and suitable manners, at that point the guarantee of "the Internet of Things "is, essentially, a considerably "more smarter" planet that keeps us more secure.

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		6

1 MAIN PART

1.1 Hardware and software components justification

1.1.1 IoT parts justification

A Raspberry-PI is a little single-board PCs created by the Raspberry Pi Establishment to help advance instructing of some essential software engineering standards, both in organizations of learning and under-developed nations. The first model turned out to be significantly more famous than foreseen, selling outside its objective commercial center for utility in fields like robotics. It now's generally utilized even in research ventures, such as climate observation because of its minimal effort and movability. It doesn't come with peripherals, (for example, consoles and mice) or cases. Be that as it may, a couple of frill are added to a few official and unofficial bundles. The Raspberry-PI equipment has developed through a few forms that include varieties in the sort of the central processing unit, measure of memory limit, networks, and peripheral gadget support.

Table 1.1 - Comparison of different Raspberry PI devices

	Raspberry pi 1	Raspberry pi 2	Raspberry pi 3	Raspberry pi 4
SOC Type	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837B0	Broadcom BCM2711
GPU	VideoCore IV 1080p@30	VideoCore IV	VideoCore IV	VideoCore VI
CPU clock	700 MHz	900 MHz	1.4 GHz	1.5 GHz
RAM	512 MB	1 GB	1 GB DDR2	1 GB , 2 GB, 4 GB LPDDR4
Wi-Fi (onboard)	-	-	2.4GHz and 5GHz 802.11 b/g/n/ac	2.4GHz and 5GHz 802.11 b/g/n/ac
Bluetooth (onboard)	-	-	4.2, BLE	5.0

					CSDP 123.061.00.00 EN		
<i>Ch</i>	<i>List..</i>	<i>№ docum..</i>	<i>Sign</i>	<i>Date</i>	MAIN PART		
<i>Develop.</i>	<i>Ugwu V.O.</i>						
<i>Supervisor</i>	<i>Lutskiv A.M.</i>						
<i>Reviewer.</i>							
<i>N. Contr.</i>							
<i>H. of Dept</i>	<i>Osukhivska H.</i>				Letter	Page	Pages
					H	7	44
					TNTY. dept.CE, ICI-42		

Sensors and peripherals which can be utilized with Raspberry-PI comprises:

1. Temperature and humidity sensors. The DHT11 might be an easy to integrate temperature and humidity sensor. It isn't the quickest sensor around yet its modest value makes it helpful for testing or ventures where you don't require new readings on numerous occasions a second. The gadget just requires three associations with the Pi. +3.3v, ground and one GPIO pin.

The gadget itself has four pins however one of these isn't utilized. You can purchase the 4-nail gadget to its own or as a major aspect of a 3-pin module.

The modules have three pins and are anything but difficult to interface legitimately to the Raspberry-Pi's GPIO header:

- Humidity: 20-80% (5% precision);
- Temperature: 0-50°C ($\pm 2^\circ\text{C}$ exactness) .

The makers don't suggested that you read information from this gadget more than once per 2 seconds. On the off chance that you do you may get erroneous readings.

The 4-pin gadget would require a resistor (4.7K-10K) to be set between Pin 1 (3.3V) and Pin 2 (Information).

The 3-pin modules will as a rule have this resistor included which makes the wiring a touch simpler. Consequently I got hold of the module which I could then join to the Pi with a bit of 3-way Dupont link.

Various providers may wire the module sticks distinctively so check the PCB markings to spot Vcc (+), information and Ground (-).

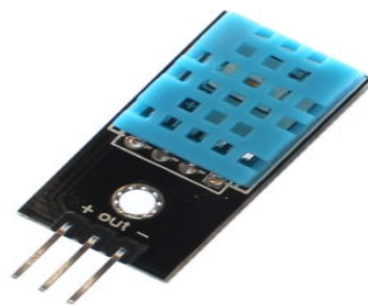


Fig. 1.1- DHT11 temperature sensor

2. Relays. The GPIOs of the Raspberry Pi work with 3.3V, despite the fact that it likewise has a 5V pin. In any case, numerous gadgets require a higher voltage. All together not to combine the circuits, one can utilize transfers, which are essentially switches. This has the favorable position that you can likewise switch circuits with higher voltages with the Raspberry-PI, without gambling something.

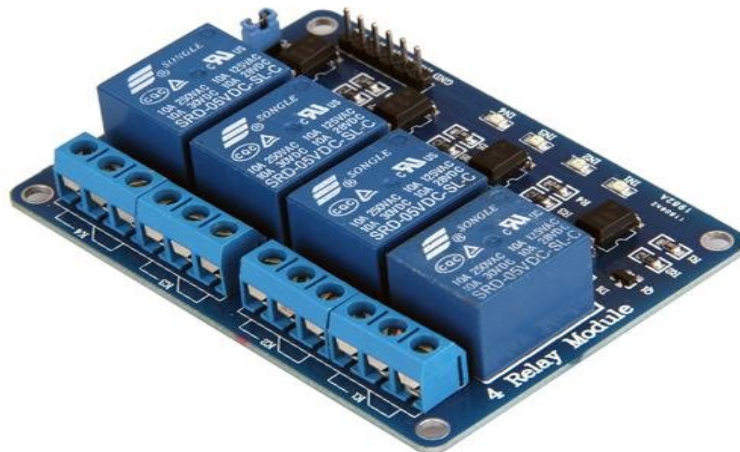


Fig. 1.2 - Relays

3. Led lights. The light-emitting diode (LED) is today, one of the most vitality effective and rapidly developing lighting advances. High quality led lights can last more, they are increasingly tough, and furthermore offer preferable light quality over different sorts of lighting. An energy proficient lighting technology, LEDs can possibly radically change the fate of lighting. Last multiple times longer and use in any event 75% less vitality, contrasted with glowing lighting. Almost no heat. To compare, brilliant bulbs discharge 90% of their energy as heat and CFLs discharge about 80% of their vitality as heat.

1.1.2 Software justification

Software needed to create project comprises libraries, software frameworks and software development kits. Architecture of the developed software is depicted on

the figure 1.3. User interface (UI) provides possibility to manage home automation system in comfortable and natural for user way throughout web-interface. Web-interface can be accessible from any desktop or mobile device web-browser. Business-logic provides ability to work with devices and database, encapsulates all processes needed for data processing. Database allows to persist all of the data and which are gathered from devices.

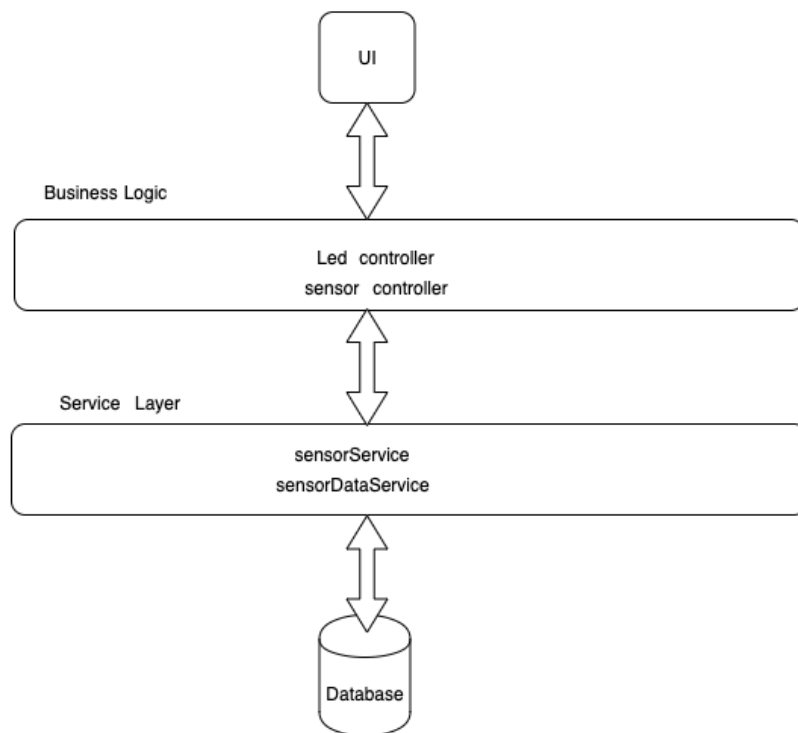


Fig. 1.3- Architecture frontend - backend

1.1.2.1 Java Programming language

Decided to write back-end part on Java programming language, because it is one of the top and leading programming languages for IoT. Java is an object oriented programming language, has built in capabilities that proves useful in programming the Internet of Things, it is also highly portable and does not have hardware limitations. The code can be used where JVMs are common such as smartphones and

servers, also in smallest machines making it ideal for IoT software. Java version 11.2 was used in this project.

The Spring Boot frame work was chosen because of its ease of integration with REST API and the massive HTTP support.

A broadly useful programming language that's class-based, object-situated, simultaneous and explicitly intended to have as barely any execution conditions as could be expected under the circumstances. It is proposed to empower designers "compose once, run anyplace", implying that an arranged Java code can run on all platforms that supports Java without the need for recompilation. Lets say you code and compile a Java program on UNIX, it will be able to run on Microsoft Windows, Mac, or any LINUX machine with none changes to the source code. WORA is accomplished by aggregating a Java program into what is known as a byte code. The configuration of a byte code is platform autonomous. The Java Virtual Machine (JVM), empowers the byte code to run, free of the stage [10].

Java has numerous highlights. A portion of those are particular to Java and a couple of these are regular among different languages.

1. Platform Autonomous: java compiles to byte code unlike some other programming languages like C++ or C. This byte code is appropriated over the web and interpreted by the Virtual Machine (JVM) paying little mind to the platform it is being executed on

2. Security – Java's safe component it empowers developers to create malware free, alter free frameworks/software programs. Confirmation procedures depend on open key encryption.

3. Portability – Being platform unbiased makes Java versatile. Java Compiler is written in ANSI C with a perfect portability limit, which might be a POSIX subset.

4. Strong – In an attempt to eliminate error prone programs, the Java programming language stresses on compile time and runtime checking

5. Multithreaded – Java is multi-threaded, making it possible to write programs that can perform many tasks simultaneously.

6. Java i interpreted – The byte code is interpreted on the fly to local machine instructions and isn't put away anyplace. The procedure of improvement is

progressively fast and increasingly scientific since the linking is a gradual and lightweight procedure.

7. Superior – Java utilizes “just in time compilation”, making it superior.

8. Java does garbage collection — Java makes use of automatic garbage collector to manage memory in the lifecycle of an object. The developer decides when objects are made, and in this way the Java runtime is obligated for recuperating the memory once objects are no longer being used. When no references to an item remain, the inaccessible memory gets qualified to be freed automatically.
[7]

1.1.2.2 The Spring Framework.

Spring makes it effortless to make enterprise java applications. It provides software developers with everything they would need to grasp the Java development in an enterprise environment, with assistance of the build tool “gradle” and Kotlin as a substitute language on the JVM, and with the adaptability to make numerous sorts of designs depending on the needs of the application. As of Spring version 5.1, Spring currently requires JDK 8+ (Java SE 8+) and provides an "out-of-the-box" support for JDK 11 LTS. Java SE version 8 update 60 is typically suggested on the grounds that the base fix discharge for Java 8.

Spring underpins a wide scope of utilization situations. In an enormous venture, applications frequently exist for an all-encompassing time and need to run on a Java Development Kit and application server whose upgrade cycle is outside control of the developer. Others may run as one container with the server installed, conceivably during a cloud domain. However others could likewise be independent applications, (for example, clump or combination outstanding burdens) that don't need a server. The framework is open source. It has a large and dynamic network that gives nonstop input upheld a different scope of genuine use cases. This has helped in the progressive development of spring overtime

The term "Spring" signifies different things in different settings. After a period of time, other Spring ventures are based on the Spring System. often, when

developers say "Spring", they mean the entire group of projects. The documentation centers around the establishment of the Spring System itself. The Spring System is isolated into modules. Applications can pick which modules they need. At its heart of the framework is modules of the core container, including an arrangement model and a reliance infusion component. More than that, the Spring Structure is equipped for giving fundamental help to different application models, including informing, value-based information and perseverance, and web. The Spring MVC web system is included in the package, and in correspondence, the Spring WebFlux responsive web structure.

Spring first appeared in 2003 as a reaction to the unpredictability of the first J2EE determinations. While a few designers may think about Java EE and Spring to be in rivalry, the truth of the matter is that Spring is integral to Java EE. The Spring programming model incorporates the java EE platform specification with all around carefully selected details from the EE umbrella:

- JMS;
- JPA (JSR 338);
- Bean Approval (JSR 303);
- WebSocket Programming interface (JSR 356);
- JSON Restricting Programming interface (JSR 367);
- Simultaneousness Utilities (JSR 236);
- Servlet Programming interface (JSR 340);

also JTA/JCA arrangements for exchange coordination, if need be.

The Spring System has extraordinary help for Reliance Infusion (DI) and standard Explanations (JSR 250) details, which application designers may like to utilize as opposed to the Spring-explicit components gave by the Spring Framework.

As of Spring version 5.0, it now requires the Java EE 7 level (e.g Servlet 3.1+, JPA 2.1+) as a base- while additionally furnishing out-of-the-crate incorporation with more up to date APIs at the Java EE 8 level. This continues Spring completely good with for example Tomcat 8 and 9, WebSphere 9, and J Boss EAP 7.

Overtime the role of Java EE in application advancement has developed. In the young stages of Java EE and Spring, applications were made to be sent to an

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		13

application server. currently, with the guide of Spring Boot, applications are made in a devops and cloud-accommodating way, with the installed Servlet holder and trivial to vary. As of Spring System 5, a WebFlux application doesn't use the Servlet Programming interface directly and may run on servers, (for example, Netty) that aren't Servlet holders.

Spring keeps on developing and progressively advancing Past the Spring System, there are different project, such as Spring Boot, Spring Security, Spring Information, Spring Cloud, Spring Group, among others. It's critical to recollect that each undertaking has its own ASCII content document storehouse, issue tracker, and discharge rhythm. See spring.io/ventures for the whole rundown of Spring ventures.

The core values of the Spring Structure:

— Give decision at each level. Spring permits you to concede plan choices as late as could be expected under the circumstances. For instance, you'll switch persistence suppliers through design without making changes your code. The equivalent is valid for a few other framework concerns and incorporation with outsider APIs.

— Oblige various points of view. Spring grasps adaptability and isn't obstinate about how things ought to be finished. The system underpins a wide scope of use needs with alternate points of view.

— Keep up solid in reverse similarity. The advancement of spring has been overseen cautiously to compel hardly any breaking changes between variants. Spring structure additionally underpins an astutely picked scope of JDK renditions and outsider libraries to encourage upkeep of utilizations and libraries that rely on Spring.

— API design. The Spring group invests huge amounts of thought and energy into making APIs that are natural which span across numerous renditions and many years.

— Set exclusive requirements for code quality. The framework applies a powerful accentuation on significant, current, and precise javadoc. It's one among just a couple of ventures which will guarantee clean code structure with no roundabout conditions between bundles.

At its core, Spring Framework is essentially only a container for dependency injection, with a couple of convenient layers included in it. It helps create Java application quicker and all the more advantageously

Also in the developed project dependency injection was used. Envision we are writing a Java class which lets us get to a persons table in our database. We would call these classes DAOs (Data access object) or repos. So, we are going to write a PersonDAO class:

```
public class PersonDao {
    public Customer findById(Integer id) {
        // execute a sql query to find the user
    }
}
```

The class has only a single method which lets you find a person in the database table by their particular IDs. Our PersonDAO needs a database association to execute the fitting SQLite inquiry. in the ecosystem of Java, we (normally) get that database association from another class, this class is called a DataSource. Our code would now look something like this:

```
import javax.sql.DataSource;
public class PersonDao {
    public Customer findById(Integer id) throws SQLException {
        try (Connection connection = dataSource.getConnection()) {
            PreparedStatement selectStatement =
                connection.prepareStatement("select *
                    from person where id = ?");
            // use the connection etc.
        }
    }
}
```

Now a question that might arise is, “where does our PersonDao get its dataSource dependency from?” Clearly the DAO relies upon a substantial DataSource to fire those SQLite queries. The native solution would be to make

another DataSource through a constructor, each time we need one. Along these lines, to interface with a SQLite database our PersonDAO could resemble this:

```
import com.mysql.cj.jdbc. SQLiteDataSource;
public class PersonDao Dao {
    public Person findById(Integer id) {
        SQLite DataSource dataSource = new SQLite DataSource();
        dataSource.setURL("jdbc:SQLite:/home/pi/db.sqlite");
        try (Connection connection = dataSource.getConnection()) {
            PreparedStatement findStatement =
                connection.prepareStatement("select * from persons where id = ?");
            // execute the statement.. and convert
            the raw jdbc result set to a person
            return person;
        }
    }
}
```

We need to interface with a SQLite database; henceforth we are utilizing a SQLite DataSource We utilize our recently made DataSource.

To manage instances of our program we will use Dependency Injection of Spring Framework. It reduce amount of code which developer should write to handle instances of objects life-cycle.

Spring framework provides two ways to carry out dependency injection:

- By Constructor;
- By Setter method.

Lets discuss advantages of Dependency Injection:

- DI permits a customer the malleability of being configurable. Just client's conduct is fixed.
- Testing can be performed utilizing mock articles.
- Freely couple engineering.
- DI doesn't require any adjustments in code conduct it very well may be applied to inheritance code as refactoring.
- DI permits a customer to dispose of all information on a solid usage that must utilize. It is progressively reusable, increasingly testable, progressively intelligible code.

— DI makes it conceivable to dispose of, or possibly decrease pointless conditions.

— DI permits simultaneous or free turn of events.

— DI diminishes coupling between a classification and its reliance.

1.1.2.3 PI4J library

The promise of the PI4J venture is to gracefully an extension between the local equipment and Java for full access to the Raspberry Pi in with a Java-accommodating item arranged methodology. it is an open source venture created by proficient and experienced programming engineers. Notwithstanding the basic crude equipment get to usefulness, this undertaking additionally endeavors to gracefully a gathering of cutting edge includes that make working with the Raspberry Pi a progressively helpful and mind boggling experience for Java developers..

Project Website: <http://www.pi4j.com>

Source Repository: <https://github.com/Pi4J/pi4j>

Fundamental Highlights:

— Fare and un-export GPIO pins;

— Design GPIO pin heading;

— Design GPIO pin edge location;

— Control/compose/set GPIO pin states;

— Peruse GPIO pin states;

— Send and get information by means of RS232 sequential correspondence.

Propelled Highlights:

— Heartbeat GPIO pin states;

— Tune in for pin state changes;

— Naturally set GPIO states on program end (GPIO shutdown);

— Triggers for computerization dependent on pin state changes;

— Getting to framework and system data from the Raspberry-PI;

					CSDP 123.061.00.00 EN	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		17

— Incorporates some Covering classes which helps in direct access to WiringPi Library from Java.

1.1.2.4 SQLite database

SQLite can be characterized as a product library that gives an electronic database the board framework, "light" in SQLite signifies "light weight", this is with regard to setup, database administration, and required resource. One can store any value in any column, paying little heed to information type because the database utilizes dynamic sorts for tables. SQLite database allows for a solitary database association that can get to numerous database documents simultaneously. This presents numerous pleasant features like joining tables in various databases or copying information between databases in just a single command.

Here are a few highlights of the SQLite RDMS:

1. Serverless— Typically, a RDBMS, for example, MySQL, PostgreSQL, and so forth., requires a different server procedure to work. The applications that need to get to the database server use TCP/IP convention to send and get demands. This is called customer/server engineering. SQLite works in an unexpected way, the database doesn't require a server to run, it is instead incorporated with the application that gets to the database. Applications that interact with the SQLite db read and write directly from the database files stored on disk.

2. SQLite is Independent—which means it requires little help from the operating system as well as any outer library, making SQLite usable in such a large number of situations. Most particularly in embedded devices (e.g cell phones, game consoles, handheld media players)

3. SQLite requires Zero-configuration—Due to the server-less engineering, the developer doesn't have to install the database before utilizing it. There is no server procedure that must be configured. Likewise, SQLite doesn't utilize any setup documents.

4. SQLite is transaction— this means that All exchanges in SQLite are completely ACID-compliant. this implies that all queries and changes are Consistent, Atomic, Isolated, and Durable.[8]

1.1.2.5 React Framework

React.js is an open-source JavaScript library that is utilized for rendering UIs explicitly for SPA(single-page applications). It's primarily utilized for dealing with the view layer for web and portable applications. The magnificence of React is that the made UI parts are reusable.

For instance, in my project I used the following components: `GitButton.js` , `Led.js` , `Sensors.js`

```

1  import React from 'react'
2
3  const Sensor = (props) => {
4    return (
5      <React.Fragment>
6        <button onClick={()=>props.getTempAndHumidity()} className="br-pill tempBtn bg-navy washed-
7          get temp and humidity</button>
8        { props.isLoading ? <p className="fw5 tc i dark-green">getting data..</p> :
9          <div>
10         <div className="tempData">temperature</div>
11         <div className="humidityData">humidity</div> <br/>
12         {
13           props.temperature.length !== "" ?
14           <div>
15             <div className="tempDisplay dataBox">{props.temperature} &#x2103;</div>
16             <div className="humidityDisplay dataBox">{props.humidity} % </div>
17           </div>
18           :null
19         }
20       </div>
21     )
22   }
23   {props.tempErr? <p className="errMsg fw8">{props.tempErr}</p>:null}
24 </React.Fragment>
25 )
26 }
27 export default Sensor
28

```

Fig. 1.4 - Screenshot of React components included in the program

The code for the `GitButton` is shown on figure 1.5.

```

App.js  GitButton.js x  firebase.json  Led.js
src > Components > GitButton.js > ...
2
3  const GitButton = () => {
4    return (
5      <a className="no-underline near-white bg-animate bg-near-black hover-bg-gray inline-flex items-center ma2 tc br2 pa2" href="https://github.c
6        <svg className="dib h2 w2" fill="currentColor" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 16 16" fillRule="evenodd" clipRule="ev
7          <span className="f6 m13 pr2">GitHub</span>
8        </a>
9    )
10 }
11
12 export default GitButton
13 |

```

Fig. 1.5- GitButton code

Because of the power of react i did not have to write this code myself, it's gotten from a third party library which is included in the project – “tachyons”.

React permits designers to make web applications that can render UI dependent on state, without reloading the page. The fundamental motivation behind React is a quicker, progressively adaptable, and basic arrangement. It works just on UIs in the application. This relates to the view in the MVC layout. It is regularly utilized with a blend of other JavaScript libraries or structures.

A few significant highlights of React

1. JSX — In React as opposed to utilizing customary JavaScript for templating, it utilizes JSX. JSX is a basic JavaScript that permits HTML citing and utilizes these HTML label sentence structure to render subcomponents. HTML language structure is processed into JavaScript calls of React System. We can likewise write in unadulterated old JavaScript.

2. Utilizing React Native — React native is a portable applications building system utilizing just Javascript. It utilizes a proportionate structure as react, allowing you to use/incorporate an upscale versatile UI library/decisive segments. It utilizes an identical central UI building obstructs as customary iOS and Android applications. The best a piece of utilizing react native is to permit/receive segments written in Objective-C, Java, or Quick.

3. Single-Way information stream — In React, a gathering of changeless qualities are passed to the segments renderer as properties in its HTML labels. The part can't straightforwardly change any properties yet can pass a get back to work

with the assistance of which we can do adjustments. This total procedure is comprehended as "properties flow down; actions flow up".

4. The Virtual Document Object Model — React makes an in-memory game plan store which figures the progressions made at that point then refreshes the browser. This permits an uncommon element that allows the software engineer to code as though the whole page is rendered on each change while respond library just renders segments that truly change. Virtual DOM is shown on figure 1.6.

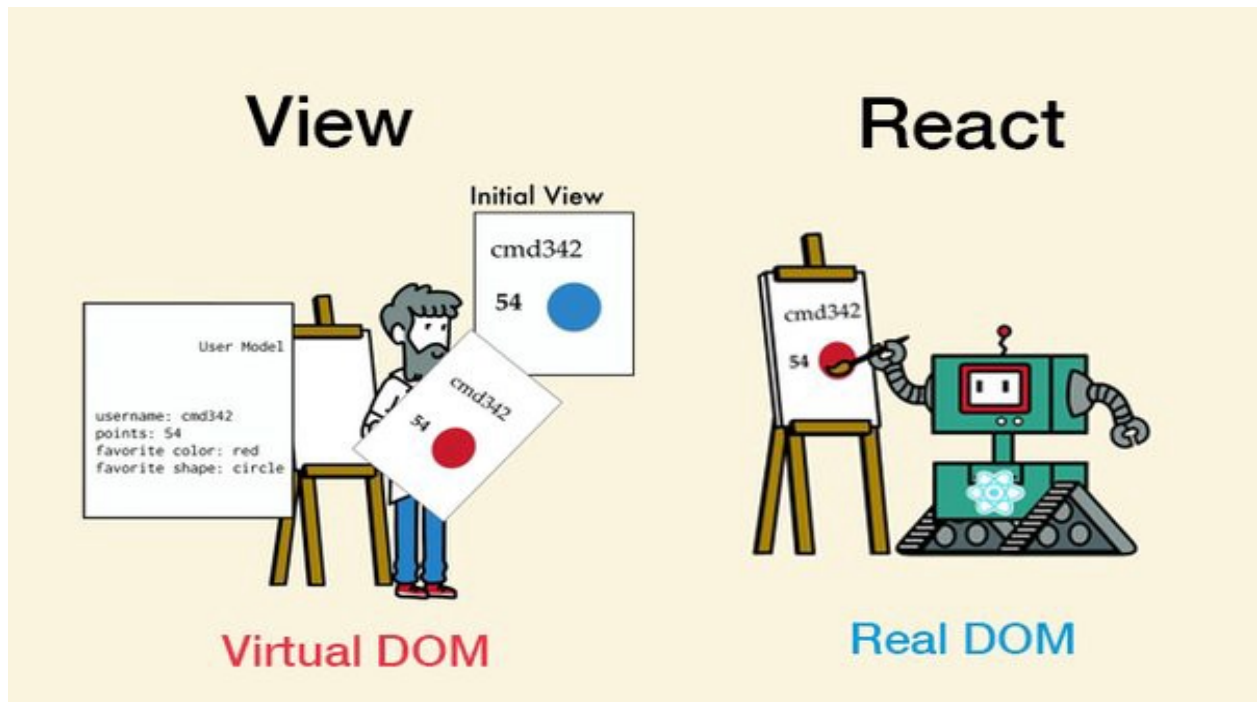


Fig. 1.6 - The react virtual Dom

There are various open-source tools for making the development of front-end web application simpler, such as Angular. Let us investigate the benefits of React over other serious innovations or structures.

1. Straightforwardness — ReactJS is easier to get a handle on immediately. Incredible documentation and backing, the part based methodology, all around characterized lifecycle techniques, and utilization of downright JavaScript make Respond extremely easy to learn, develop an expert web (and mobile) applications, and bolster it.

2. Simple to discover — Anybody with an essential past knowledge in programming can undoubtedly comprehend React while Angular and Ember are referenced as 'Domain-specific Language', suggesting that it is hard to learn them. For React, you just need essential information on CSS and HTML.

3. Local Methodology — Developer who work with React frequently want to make versatile applications (React native). What's more, with React's broad code reusability is bolstered. So simultaneously, we can make IOS, Android and Web applications.

4. Availability of data binding — React utilizes single direction information authoritative and an application engineering called flux pattern, the progression of information to parts through one control point – the dispatcher. It's simpler to investigate independent segments of enormous ReactJS applications.

5. Execution — React doesn't offer any idea of an inherent holder for reliance. You can utilize Browserify, Require JS, EcmaScript 6 modules which we will utilize by means of Babel, ReactJS-di to infuse conditions consequently.

6. Testability—ReactJS applications are overly simple to test. React views are frequently rewarded as elements of the state, so we will control with the state we go to the ReactJS view and take a look at the output and triggered actions, events, functions, etc.

1.1.3 Cloud services justification

Cloud computing alludes to the conveyance of on-request registering administrations from applications to processing power customarily over the web and also on a pay-as-you go basis. Rather than possessing and maintaining up their data centers, organizations can lease access to anything from applications to storage from a cloud specialist co-op. One of the many favorable position of utilizing distributed computing administrations is the capacity of firms to evade the forthright expense and intricacy of possessing and keeping up their own IT foundation, and rather essentially purchase what they use, when they use it.

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		22

Thusly, suppliers of distributed computing administrations can appreciate noteworthy economies of scale by conveying an equal administrations to a decent scope of customers.

Cloud services covers a gigantic scope of alternatives, from capacity, systems administration, and handling power through to tongue preparing and artificial intelligence additionally as standard office applications. Practically any assistance that doesn't expect you to be genuinely near the precarious edge of the pc equipment that you basically are utilizing would now be able to be conveyed by means of the cloud.

cloud computing supports an immense number of administrations. This incorporates buyer administrations like hurray mail or the icloud back-up of the photographs on your iphone, however to the administrations which grant huge endeavors to have every one of their information and run the entirety of their applications inside the cloud. Netflix depends on distributed computing administrations to run its video gushing assistance and its different business frameworks as well, and have assortment of different associations.

Cloud computing is gradually being default for a most applications, more software vendors are offering their applications as administrations over the web rather instead of as standalone products as they battle to switch to a membership model. Be that as it may, there's a potential drawback to cloud computing, in that it likewise can present new expenses and new dangers for organizations utilizing it.

Building the foundation to support cloud computing currently represents a one third of all IT expenditure around the world, consistent with research from IDC. Meanwhile spending on traditional, in-house IT continues to slip as computing workloads still move to the cloud, whether that's public cloud services offered by vendors or private clouds built by enterprises themselves.

Analytics shows that worldwide consumption on cloud administrations will reach up to \$260bn this year up from \$219.6bn. It's pace of development is likewise quicker than the investigators anticipated. Yet, it isn't completely clear what extent of that request is originating from organizations that truly need to move to the cloud and

the manner in which much is being made by software vendors who now just offer cloud variants of their items.

One of the most significant cloud approaches for cloud-native developers is a Platform-As-A-Service (PaaS). In this project, firebase hosting is used to host the frontend part of the program. firebase is google's PaaS solution.

Platform-as-a-Service is the next layer up - just as the hidden storage, organizing, and virtual servers this will likewise incorporate the apparatuses and programming that designers need to assemble applications on: that would incorporate middleware, the executives, working frameworks, and improvement devices.

Let's discuss a few benefits of cloud computing which have great impact on this project development:

— The specific advantages will vary contingent upon the kind of cloud administration getting utilized however, on a very basic level, utilizing cloud administrations implies organizations not looking for or keep up their own figuring foundation.

— Purchasing servers, updating applications or operating systems, or decommissioning and evacuating equipment or software when it's outdated will never again be important, in light of the fact that it is completely dealt with by the provider. For product applications, similar to email, it can signify change to a cloud supplier, rather than put stock in-house abilities. An organization that centers around running and making sure about these administrations is presumably going to have preferable abilities and experienced staff over a little business could stand to lease, so cloud administrations could likewise be prepared to convey a more secure and proficient support of finish clients.

— Exploiting cloud services, organizations can move quicker on their activities and test out ideas without huge forthright expenses and long obtainment, since firms just need to buy the assets they expend. This idea of business spryness is generally referenced by cloud advocates as a key advantage. The capacity to turn up new administrations without the time and vitality identified with conventional IT acquisition should mean that's simpler to encourage going with new applications

quicker. Furthermore, if a substitution application is by all accounts an uncontrollably mainstream ,the versatile idea of the cloud implies it's simpler to scale it up quick.

— For a company with an application that has enormous peaks in use, for example that's applications just utilized at a particular time or year, it's going to bode well to have it facilitated inside the cloud, rather than have committed equipment and programming laying inactive for a significant part of the time. Moving administrations like email or CRM to a cloud facilitated application could evacuate a weight on interior IT staff, and if such applications don't produce a lot of upper hand, there will be minimal other effect. Moving to a cloud model likewise moves spending from capex to opex, which can be helpful for a couple of organizations.

1.1.4 Network protocols justification:

In this subchapter will analyze protocols which can be used for IoT projects and which are suitable for developed home automation system: MQTT, REST over HTTP, XML-RMS, GraphQL and Protobuf.

1.1.4.1 Message Queuing Telemetry Transport protocol

Message Queuing Telemetry Transport (MQTT) is an informing convention that furnishes resource-constrained network clients with an unburdened alternative with an unburdened choice to circulate telemetry data that is light-weight. The convention, which utilizes a distribute/buy in correspondence design, is utilized for machine-to-machine (M2M) correspondence and assumes a critical job inside the web of things (IoT).

The MQTT convention might be a decent choice for remote systems that have differing levels of inactivity because of intermittent data transfer capacity requirements or untrustworthy associations.

The MQTT convention covers two subjects, a customer and a merchant. A MQTT merchant might be a server, while the customers are the associated gadgets. At the point when a device or customer needs to send information to a server or

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		25

agent it's known as a distribute. At the point when the activity is turned around, it's known as a buy in.

In the event that the association from a buying in customer to a dealer is broken, at that point the merchant will cushion messages and push them twisted the supporter when it's back on the web. On the off chance that the association from the distributing customer to the representative is separated all of unexpected, then the dealer can close the association and send endorsers a reserved message with guidelines from the distributor. While the TT in MQTT represents Telemetry Transport, the MQ is as to an item called IBM MQ.

Most significant cloud services suppliers, including Amazon Web Administrations (AWS), Google Cloud, IBM Cloud and Microsoft Purplish blue, support MQTT.

1.1.4.2 REST over HTTP or HTTPs protocol

REST stands fo representational state transfer. REST is a web based design, that utilizes HTTP Convention. It revolves around resource where every component may be a resource and a resource is accessed by a standard interface using HTTP standard methods. This practice was first introduced in the year 2000 by Roy Fielding. With REST architecture, a server comprehensibly gives access to assets and REST customer gets to and changes these assets. Every last one of these asset are distinguished by URIs/worldwide IDs. The REST administration utilizes different delineations to speak to an asset like text, JSON, XML. JSON is the most well known one.

The four HTTP techniques are commonly utilized in REST based architecture.

GET – Provides a read only access to a resource.

POST – Used to create a new resource.

DELETE – Used to remove a resource.

PUT – Used to update an existing resource or create a new resource.

Restful web service. A web service might be an assortment of open conventions and gauges utilized for trading information between applications or

frameworks. Applications coded in various programming languages and running on different stages can exchange information over PC systems utilizing web services, just like the Internet during a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is thanks to the utilization of open standards.

Web services supported with REST Architecture are referred to as RESTful web services. These web services make use of HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation like JSON and set of HTTP Methods. [5]

Lets discuss the advantages of REST. The primary favorable position of utilizing REST, both from a customer and server's side viewpoint, is that REST-based associations happen utilizing develops that are recognizable to any individual who knows about utilizing the internet's HTTP.

For example, REST-based connections all convey their status utilizing standard HTTP status codes. so all engineers taking a shot at the code base comprehends that a 404 methods a mentioned asset wasn't found; a 401 code suggests that the solicitation wasn't approved; a 200 code infers that all is well; and a 500 infers that there was an unrecoverable application blunder on the server.

Similarly, certain details, for instance encryption and information transport trustworthiness are unraveled not by including new systems or advancements, however rather by depending on notable Secure Attachments Layer (SSL) encryption and Transport Layer Security (TLS). Along these lines, the entire REST design is made upon ideas with which most engineers are familiar with.

REST is additionally a language-free structural style. REST-based applications are regularly composed utilizing any language, be it Java, Kotlin, NET, AngularJS or JavaScript. Up to a programing language can make electronic solicitations utilizing HTTP, it's feasible for that language to be wont to conjure a RESTful Programming interface or web service. Essentially, RESTful web services are frequently composed utilizing any language, so engineers entrusted with executing such administrations can pick innovations that business best for their circumstance.

The other favorable part of utilizing REST is its inescapability. On the server side, there are a spread of REST-based frameworks for helping engineers make RESTful web services, including RESTlet and Apache CXF. From the client side, the entirety of the new JavaScript structures, for example, JQuery, Node.js, Angular and EmberJS, all have standard libraries incorporated with their APIs that make conjuring RESTful web benefits and devouring the XML-or JSON-based information they return a relatively clear undertaking.

Lets' discuss the disadvantages of REST.

The upside of REST utilizing HTTP builds additionally makes limitations, be that as it may. Huge numbers of the limitations of HTTP moreover become inadequacies of the rest of engineering. For instance, HTTP doesn't store state-based data between demand reaction cycles, which recommends REST-based applications must be stateless and any state the executives errands must be performed by the customer.

Correspondingly, since HTTP doesn't have any system to send pop-up messages from the server to the customer, it is hard to execute any kind of administrations where the server refreshes the client without the usage of client side surveying of the server or another kind of web hook.

From an execution point of view, a standard issue with REST is that the indisputable reality that engineers burden precisely being REST-based. Some product designers erroneously consider whatever may not be REST based to be RESTful. Driving this normal misinterpretation about REST is the indisputable certainty that it's a style of architecture, so there's no reference usage or conclusive standard which will affirm whether a given structure is RESTful. Therefore, there's talk on whether a given Programming interface fits in with REST-based standards.

XML-RPC-based protocols. And also there are REST alternatives which are very close to REST and RPC principles.

Alternate technologies for making SOA-based frameworks or making APIs for conjuring remote micro services incorporate XML over HTTP (XML-RPC), CORBA, RMI over IIOP and the Straightforward Item Access Convention (Cleanser).

Every innovation has its own arrangement of points of interest and hindrances, yet the convincing element of REST that separates it is that the undeniable certainty that, as opposed to requesting a developer codes with a lot of custom conventions or make an exceptional information group for trading messages between a customer and a server, REST demands the most ideal approach to actualize a system based web administration is to effortlessly utilize the fundamental construct of the network protocol itself, which within the case of the web is HTTP.

This is a significant point, as REST isn't planned to apply just to the web; rather, its standards are proposed to apply to all conventions, including WEBDAV and FTP.[6]

1.1.4.3 GraphQL query language communication approach

GraphQL is one among the preeminent present day methods of building and questioning APIs. GraphQL refers to a language structure that portrays the best approach to welcome information, and is generally wont to stack information from a server to a customer. GraphQL has three primary attributes:

- It permits the customer indicate precisely what information it needs.
- It disentangles accumulation of information from various sources.
- It utilizes a caring framework to clarify information.

With GraphQL, the client is in a situation to shape one call to bring the predetermined data rather than to build a few REST solicitations to get an identical.

GitHub utilizes GraphQL as it offers greater adaptability for the engineers. The choice to precisely produce the information that a client needs might be an extraordinary bit of leeway over sending various REST calls to get an identical. To create the information utilizing REST calls would require a two phase process — One to gather the information on the client and along these lines the other to bring the information about the association the client is related with. GraphQL lightens this two-advance procedure.

```

1 {
2   alllinks {
3     url
4     postedBy {
5       name
6       votes {
7         link {
8           url
9         }
10      }
11    }
12  }
13 }

```

```

{
  "data": {
    "alllinks": [
      {
        "url": "http://graphql.org/",
        "postedBy": null
      },
      {
        "url": "https://jwt.io/",
        "postedBy": {
          "name": "Test User",
          "votes": [
            {
              "link": {
                "url": "https://jwt.io/"
              }
            }
          ]
        }
      }
    ]
  }
}

```

Fig. 1.7 – GraphQL schema and query example

A GraphQL inquiry might be a string that's sent to a server to be deciphered and satisfied, which at that point returns JSON back to the customer.

Characterizes an information shape: the essential thing you'll see is that GraphQL inquiries reflect their reaction. This makes it simple to anticipate the type of the data came back from an inquiry, likewise on compose an inquiry in the event that you perceive the information your application needs. Progressively significant, this makes GraphQL exceptionally simple to discover and utilize. GraphQL is proudly determined by the information prerequisites of items and of the fashioners and designers who assemble them.

1.1.4.4 gRPC protocol

gRPC is a modern cutting edge open source remote procedure call (RPC) structure which will run anyplace. It gives client and server applications the capacity to talk straightforwardly, and makes it simpler to make associated frameworks. gRPC stage is created by Google, it was reported and made open source in late Feb

2015. The letters "gRPC" are a recursive abbreviation which proposes, gRPC Remote System Call.

gRPC has two sections, the gRPC convention, and in this manner the information serialization. As a matter of course gRPC uses Protobuf for serialization, yet it's pluggable with any kind of serialization you might want to use, with certain caveats.

1.1.4.5 Protobuf protocol

Protobuf is the default serialization protocol for the information sent among customers and servers. The encoding takes into account little messages and brisk interpreting and encoding.

Protobuf forgoes zero-duplicate of information, similar to another information trade techniques, (for example, Cap'n Proto or Flatbuffers), rather picking encoding and interpreting bytes. This makes the information littler at the benefit of getting the chance to dedicate CPU to encoding and unraveling messages. Not at all like some other serialization designs like JSON or XML, protobuf attempts to reduce the overhead of encoding by giving specifically fields in an encoded parallel arrangement that it can rapidly navigate in an anticipated way.

Thus, after protocol analysis was decided to use REST as a technology for communication. This decision based on simplicity, wide use, compatibility and easy to securing of endpoints by encrypting with HTTPS and different authentication methods.

1.1.5 Cloud native approach

Nowadays there's a widely used idea of "cloud native". Also there is a cloud native foundation which has suggest principles of building cloud application. The Cloud Native Computing Foundation charter pronounces their objective with micro services are the following:

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		31

— Altogether increment the general dexterity and practicality of utilizations. The establishment has plans to shape the development of the innovation to propel the best in class for application the executives, and to frame the innovation omnipresent and just accessible through dependable interfaces.

— The picked language for inter-service correspondence is one among the key segments of a micro service engineering. At the point when you pick such a language it should meet some center prerequisites, for example, brisk to create, versatile to an evolving situation, and be performant on the wire just as in your application. gRPC appears to fulfill these objectives, which brings about a design that's increasingly coordinated and viable, so in that sense it has all the earmarks of being a vital decision for being Cloud Native.

1.2 Hardware components implementation

To build home automation system there are hardware requirements:

- Raspberry-PI with Raspbian OS installed;
- breadboard;
- DHT11 sensor;
- LEDs;
- cables;
- host computer for remote ssh connection (we will be writing the code in an IDE and pull it to the Raspberry-PI from GitHub), which needed only on development stage.

The LED and the DHT11 are both connected to the breadboard and then to the Raspberry-PI:

- the anode of the LED is connected to GPIO 4 pin on the Raspberry-PI;
- the cathode of the LED is connected to ground.

The DHT11 has 3 pins. From the left:

- the ground pin is connected to the ground(gnd) GPIO pin on the Raspberry-PI;
- the middle pin is connected to the GPIO2 on the Raspberry-PI;

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		32

— The gcc pin is connected to the 3.3 vdc power supply.

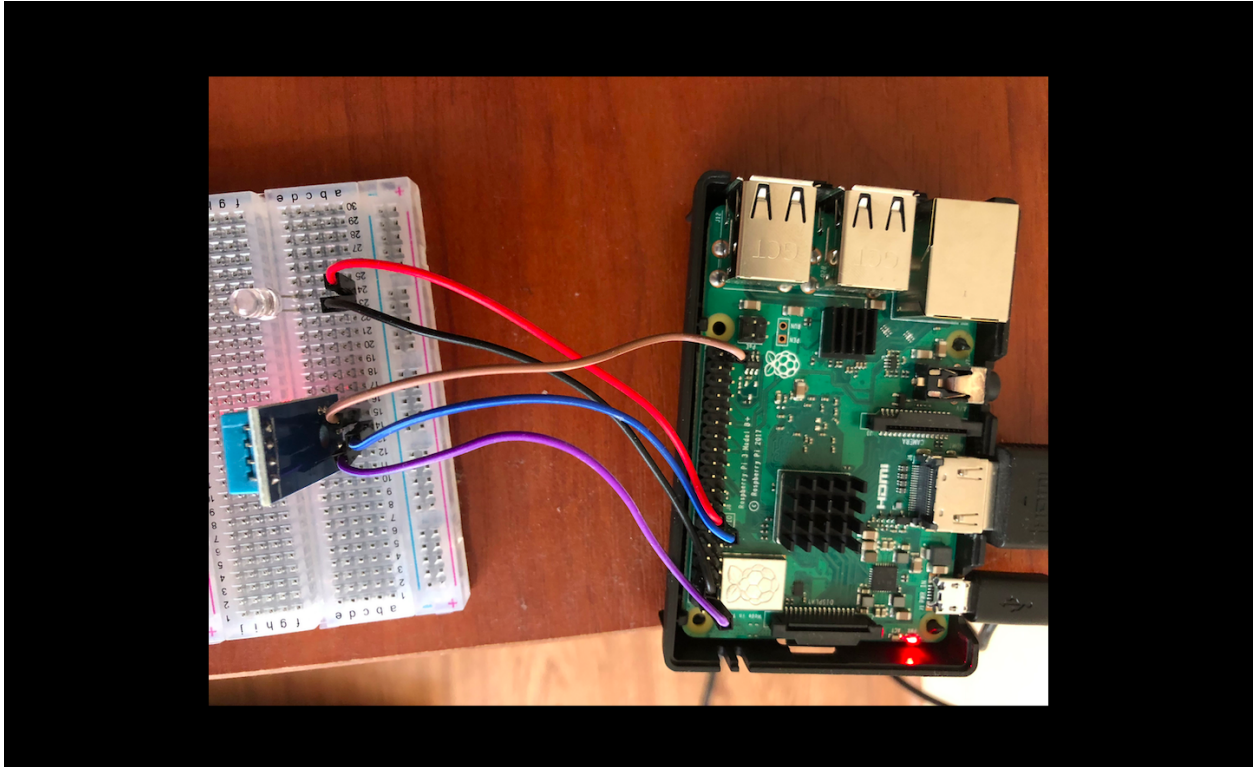


Fig. 1.8 - The home automation system prototype using a Raspberry-PI

Breadboards — these are utilized to help you interface segments to complete your essential circuit. The vertical sections of the breadboard are called terminals, while the level long columns are called power rails because they are for the most part, used to interface the power supply to the breadboard. The positive rails are demonstrated by red lines, while the negative rails are shown by dark ones.

A breadboard association with an electronic segment is made with the assistance of leg-like structures made of metal. These are classified "leads", and they can shift in size. The shorter ones are regularly alluded to as pins. A breadboard association with an electronic part is made with the assistance of leg-like structures made of metal. These are classified "leads", and they can shift in size. The shorter ones are frequently alluded to as pins

The supply of power for this breadboard will be originating from the Raspberry-PI.

1.3 Software components implementation

1.3.1 Back-end development

This chapter starts with the backend implementation of the software.

The backend implementation was done with a java framework and springboot source code is available at <https://github.com/miles-blaq/pi-Project.git>

To initialize the project with spring initializer was used Spring Initializer resource: <https://start.spring.io/>

We should be sure to include the “spring-boot-starter-web” dependency, other dependencies can be added from MVN repository. Our project library dependencies are shown in the code of build.gradle-file below.

```
plugins {
    id 'org.springframework.boot' version '2.3.0.RELEASE'
    id 'io.spring.dependency-management' version '1.0.9.RELEASE'
    id 'java'
}

group = 'com'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compile group: 'com.pi4j', name: 'pi4j-core', version: '1.2'
    compile group: 'com.pi4j', name: 'pi4j-device', version: '1.2'
    compile group: 'com.pi4j', name: 'pi4j-gpio-extension', version: '1.2'
    compile group: 'com.pi4j', name: 'pi4j-native', version: '1.2', ext: 'pom'
    testImplementation('org.springframework.boot:spring-boot-starter-test') {
        exclude group: 'org.junit.vintage', module: 'junit-vintage-engine'
    }
}

test {
    useJUnitPlatform()
}
```

Fig. 1.9 -build.gradle-file

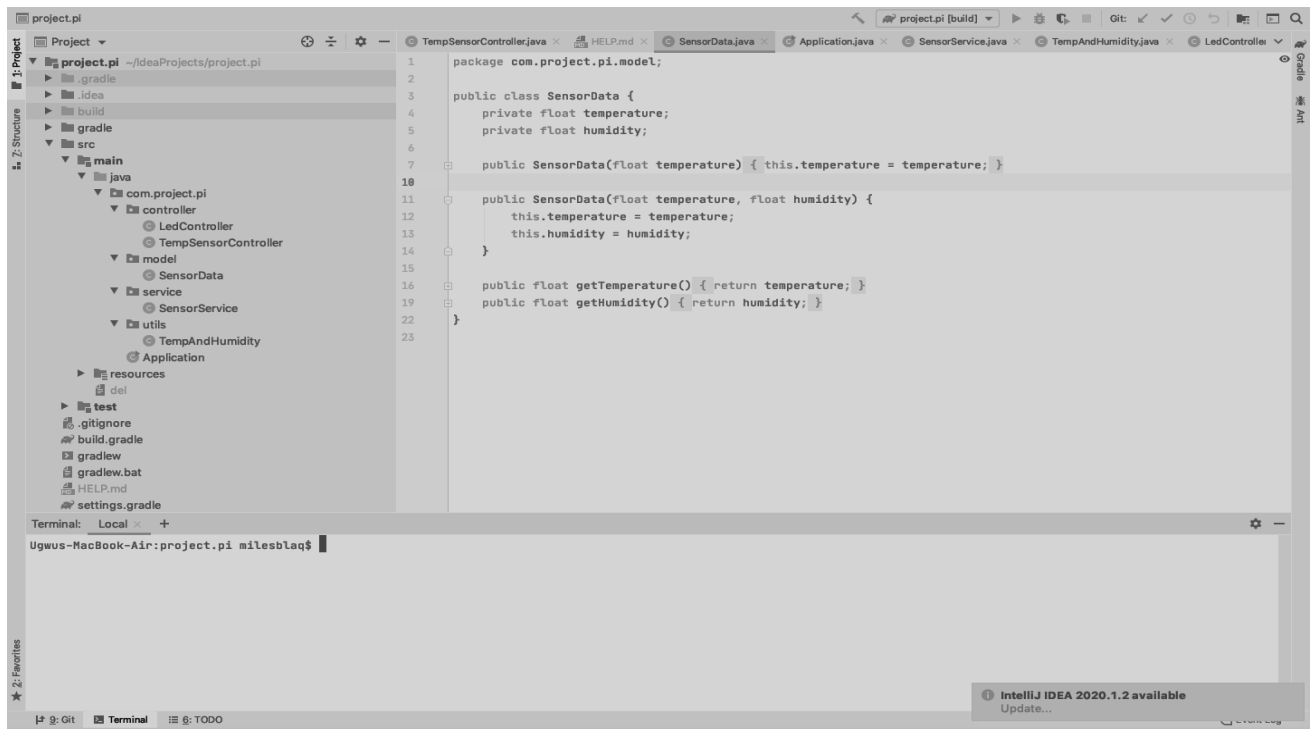


Fig. 1.10 - project.pi Spring Boot application

At first we will analyze the structure of our back-end Java code. This is the class where the Spring Boot application is configured. It must contain at least a public static void main method, which will serve as an entry point for the app, and the annotation `@SpringBootApplication`:

```

package com.project.pi;

import ...

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

}

```

The model package will contain classes that define what our data will look like. Also these are models of entities which also can be persisted into database and requested by REST-controller end-point. It can be used as a mediator before transferring and receiving data.

SensorData.java class defines the model for the data we will be getting from the DHT11 sensor on the Raspberry PI. This will be used to instantiate the sensorService once the methods are called from the controllers

```
import com.project.pi.model.SensorData;
import com.project.pi.utils.TempAndHumidity;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class SensorService {

    @Autowired
    private TempAndHumidity tempAndHumidity;

    public SensorData getSensorData() {
        return new SensorData(tempAndHumidity.getTemperature(), tempAndHumidity.getHumidity());
    }

    public SensorData getTemperatureData() { return new SensorData(tempAndHumidity.getTemperature()); }
    public SensorData getHumidityData(){
        return new SensorData(tempAndHumidity.getHumidity());
    }
}
```

Fig. 1.11 - SensorData model

The annotation @Service indicates that the bean is holding the business logic. @Autowired annotation lets Spring resolve and inject collaborating beans into our bean. The controller accepts input and converts it to commands for the model or view by triggering the methods in the service layer (Fig. 1.12).

```

5 import com.project.pi.service.SensorService;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.CrossOrigin;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.RestController;
11
12 @RestController
13 public class TempSensorController {
14
15     @Autowired
16     private SensorService sensorService;
17
18     @GetMapping("/tempAndHumidity")
19     @CrossOrigin(origins = "http://192.168.0.101:3000")
20     public ResponseEntity<SensorData> reqSensorData() {
21         SensorData sensorData = sensorService.getSensorData();
22         return ResponseEntity.ok(sensorData);
23     }
24
25     @GetMapping("/temperature")
26     @CrossOrigin(origins = "http://192.168.0.101:3000")
27     public ResponseEntity<SensorData> reqTemperatureData() {
28         SensorData temperatureData = sensorService.getTemperatureData();
29         return ResponseEntity.ok(temperatureData );
30     }
31 }

```

Fig 1.12 - The controller

We have end-points which are handled by our controller and process HTTP requests and annotated with the follow annotations:

- `@RestController` annotation is a specialized version of the controller, it simplifies the creation of Restful web services;
- `@GetMapping` annotation sets the url path of the method, and maps it as a GET request;
- `@CrossOrigin-` enables cors(cross origin resource sharing) with authorized urls, this helps secure your data.

1.3.2 Testing the REST API through Postman

First we have to build this program to test the paths, as just compiling and running the spring boot application on the host machine will produce some errors, the pi4j is not detecting GPIO pins.

```

TempSensorController 21
model 22
SensorData 23
service 24
public float getTemperature() {
    getData();
    return temperature;
}

Terminal: Local x +
=====|_=====|_/_/_/_/_/
:: Spring Boot :: (v2.3.0.RELEASE)

2020-06-19 14:43:57.501 INFO 68361 --- [main] com.project.pi.Application : Starting Application on Ugwuw-MacBook-Air.Local with PID 68361 (/Users/milesblaq/IdeaProjects/project.pi/build/libs/project.pi-0.0.1-SNAPSHOT.jar started by milesblaq in /Users/milesblaq/IdeaProjects/project.pi)
2020-06-19 14:43:57.512 INFO 68361 --- [main] com.project.pi.Application : No active profile set, falling back to default profiles: default
2020-06-19 14:44:01.302 INFO 68361 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2020-06-19 14:44:01.384 INFO 68361 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-06-19 14:44:01.386 INFO 68361 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.35]
2020-06-19 14:44:01.797 INFO 68361 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-06-19 14:44:01.797 INFO 68361 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 4088 ms
2020-06-19 14:44:02.402 INFO 68361 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-06-19 14:44:03.079 INFO 68361 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-06-19 14:44:03.120 INFO 68361 --- [main] com.project.pi.Application : Started Application in 8.29 seconds (JVM running for 9.958)

```

Fig. 1.13 – Spring Boot application running

Build the application on your IDE and execute the following command to start up the tomcat server:

```
java -jar build/libs/project.pi-0.0.1-SNAPSHOT.jar
```

Open the Postman and navigate to localhost:8080/, the mapping set in the LedController.java and it should return “Hello world”

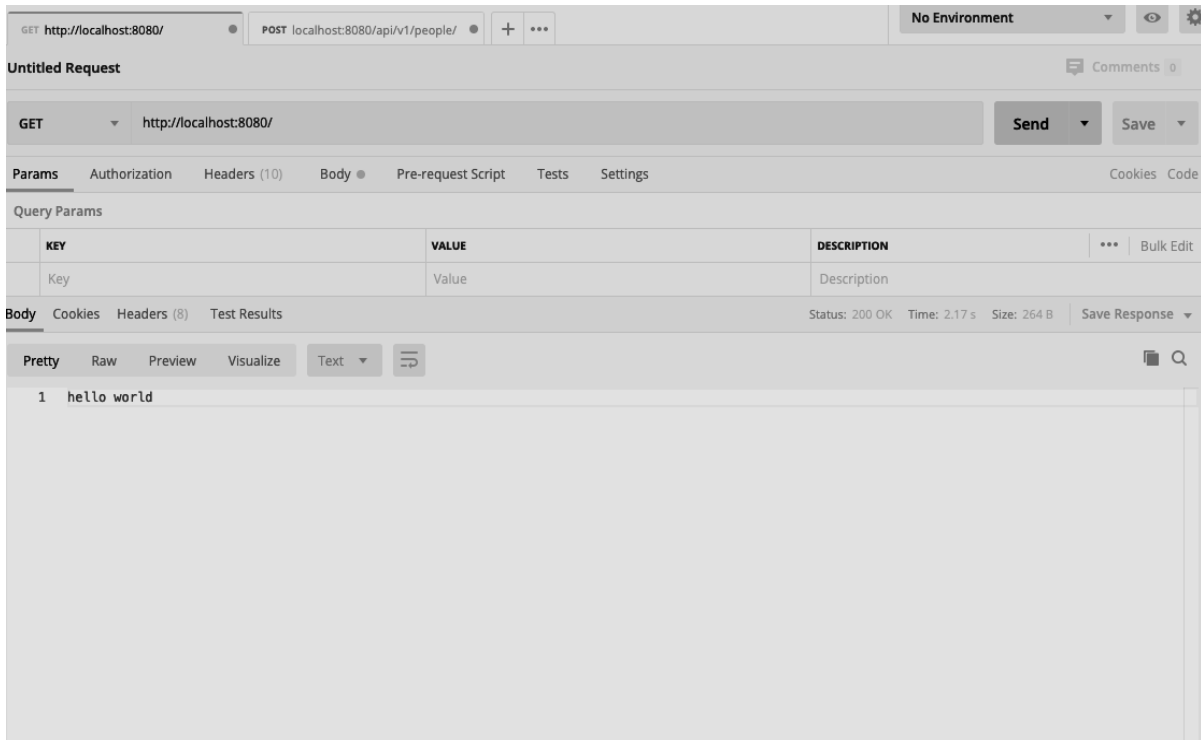


Fig. 1.14 - Testing API endpoints using postman

1.3.3 Front-end development

At first we should initialize project with create-react-app:

```
npx create-create-app r-pi-frontend
```

The next step is dependencies installation:

```
npm i -S tachyons (or any other preferred design library)
```

```
class App extends Component{
  state = {
    ledState:"",
    btnState:"on",
    ledErrMsg:"",
    isLoading:false,
    temperature:"",
    humidity:"",
    tempErr:"",
    blink:"",
    isBlinking:false
  }
  proxy = "http://192.168.0.103:8080"

  changeToggleBtnState = ()=>{
    let [ledState] = this.state
    if(ledState === "Light is off"){
      this.setState({btnState:"on"})
    }else{
      this.setState({btnState:"off"})
    }
  }

  toggle = ()=>{
    axios(`${this.proxy}/toggle`)
      .then(response => this.setState(()=>{
        return{
          ledState:response.data
        }
      }), ()=>this.changeToggleBtnState())
      .catch(error => this.setState({ledErrMsg:"server is temporarily down"}))
  }
}
```

Fig. 1.15 - Initializing the front-end

The frontend is separately from the back end of the project and make api calls to it. This was seamless because the Spring Boot backend is set up as a REST-service and that works really well with JavaScript. The code for front-end part can be found at the my GitHub repo: <https://github.com/miles-blaq/pi-frontend.git>

1.3.4 Developed methods

In our program used constant which points to the back-end URL and contains IP-address and port number:

```
proxy = "http://192.168.0.103:8080"
```

1. This method calls on the toggle method on the back-end and triggers it, then updates the states based on the result of the promise:

```
toggle = ()=>{
  axios(`${this.proxy}/toggle`)
  .then(response => this.setState(()=>{
    return{
      ledState:response.data
    }
  }, ()=>this.changeToggleBtnState()))
  .catch(error => this.setState({ledErrMsg:"server is
temporarily down"}))
}
```

2. The following method calls on the blink method and updates the state as well:

```
blinkLed = ()=>{
  axios(`${this.proxy}/blink`)
  .then(response => this.setState(()=>{
    return{
      blink:response.data,
      isBlinking:true
    }
  }, ()=> this.blinkToFalse()))
```



```

        .catch(error => this.setState({ledErrMsg:"something went
wrong"}))
    }

```

3. The next method retrieves temperature and humidity data from the Raspberry Pi and renders it on the website:

```

getTempAndHumidity = ()=>{
    this.setState({isLoading:true})
    axios(`${this.proxy}/tempAndHumidity`)
    .then(response => this.setState({
        temperature:response.data.temperature,
        humidity:response.data.humidity,
        isLoading:false
    )))
    .catch(error => this.setState({tempErr:"server is
down",isLoading:false}))
}

```

Thus, we are done with the development of both the front-end and the back-end and in the Special Part their testing will be shown.

2 SPECIAL PART

2.1 Testing layers of application

In this part of diploma thesis we will test work of the components (Fig.2.1) on all of the application layers.

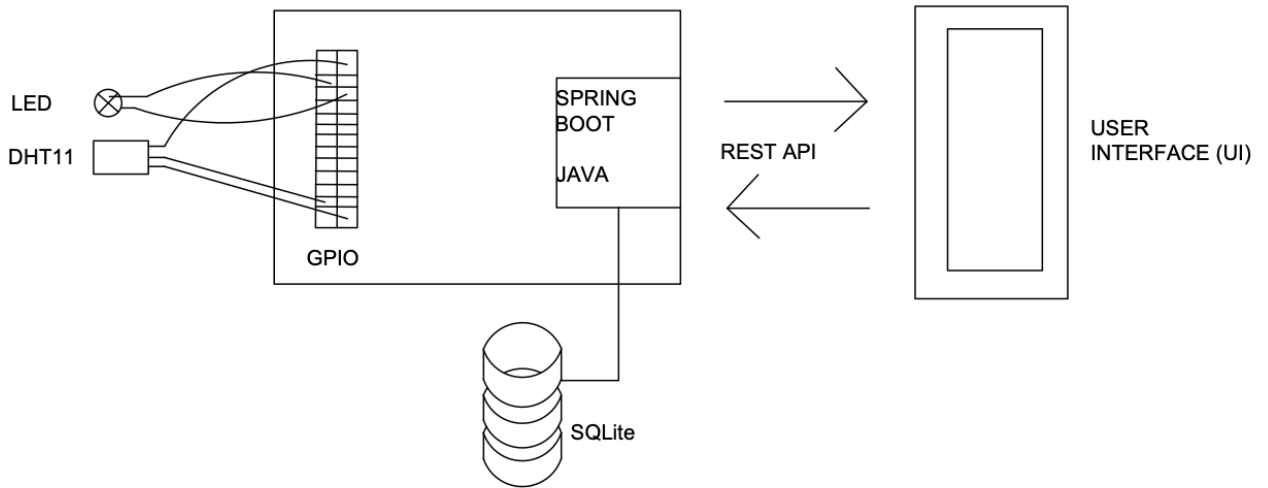


Fig 2.1 - A visual representation of the program logic

Software layers of our home automation system can be represented as sequence diagram (Fig. 2.2). At first person should be authenticated in the user interface (UI). UI is implemented as a front-end part. Controller will serve all of the front-end requests and send them to back-end logic. Business logic is implemented on the service layer which communicates with data access layer. Data access layer comprises database and sensors data. All of these parts should be tested.

CSDP 123.061.00.00 EN				
<i>Ch.</i>	<i>List</i>	<i>№ docum.</i>	<i>Sign</i>	<i>Date</i>
		<i>Ugwu V.O.</i>		
	<i>Supervisor</i>	<i>Lutskiv A.M.</i>		
	<i>Reviewer.</i>			
	<i>N. Contr.</i>			
	<i>H. of Dept</i>	<i>Osukhivska H.</i>		
SPECIAL PART				
		<i>Letter</i>	<i>Page</i>	<i>Pages</i>
		H	42	9
<i>TNTY. dept.CE, ICI-42</i>				

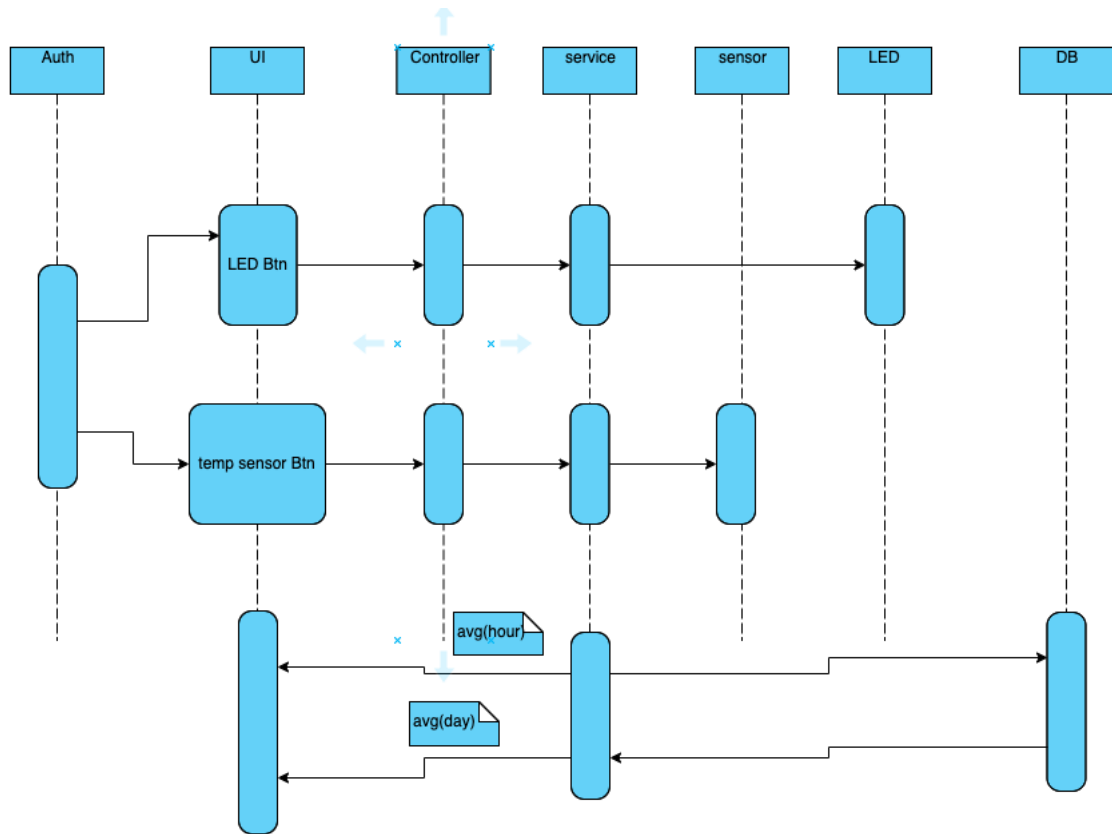


Fig. 2.2 - A sequence diagram showing the flow of data when a user issues a command from the web interface

2.2 Demonstrating the project on a local network

First we will ssh into the Raspberry-PI, to get the IP-address of the Raspberry-PI on the local network, run the command. Finding Raspberry-PI host with the hostname raspberrypi.local shown on the figure 2.3:

```
ping raspberrypi.local
```

```
milesblaq — -bash — 80x24
Ugwus-MacBook-Air:~ milesblaq$ ping raspberrypi.local
PING raspberrypi.local (192.168.0.100): 56 data bytes
64 bytes from 192.168.0.100: icmp_seq=0 ttl=64 time=8.516 ms
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=3.641 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=2.309 ms
^C
--- raspberrypi.local ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.309/4.822/8.516/2.668 ms
Ugwus-MacBook-Air:~ milesblaq$
```

Fig. 2.3 – Pinging of raspberrypi.local

Then we will ssh into it (Fig. 2.4):

```
ssh pi@<ip addr>
```

```
milesblaq — ssh pi@192.168.0.100 — 80x24
64 bytes from 192.168.0.100: icmp_seq=0 ttl=64 time=8.516 ms
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=3.641 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=2.309 ms
^C
--- raspberrypi.local ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.309/4.822/8.516/2.668 ms
Ugwus-MacBook-Air:~ milesblaq$ ssh pi@192.168.0.100
The authenticity of host '192.168.0.100 (192.168.0.100)' can't be established.
ECDSA key fingerprint is SHA256:QPF9qlgsM9Q0gYMD1CaK0M0jasyufGofTLH5az0TwFk.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.0.100' (ECDSA) to the list of known hosts.
pi@192.168.0.100's password:
Linux raspberrypi 4.19.118-v7+ #1311 SMP Mon Apr 27 14:21:24 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 23 10:09:58 2020
pi@raspberrypi:~$ git clone https://github.com/miles-blaq/pi-Project.git
```

Fig. 2.4 – ssh into the Raspberry-PI

We will clone the repository from github and cd into the project dir build the project with gradle , by running:

```
gradle build
```


http://localhost:3000/

```
Ugwus-MacBook-Air:r-pi-frontend milesblaq$ npm start
```

```
Starting the development server...
```

Fig. 2.7 - Starting React development server

It will only take a few seconds and we will navigate to <our machine's ip addr>:3000 (e.g 192.168.0.103:3000), because we will need to be on the local network.

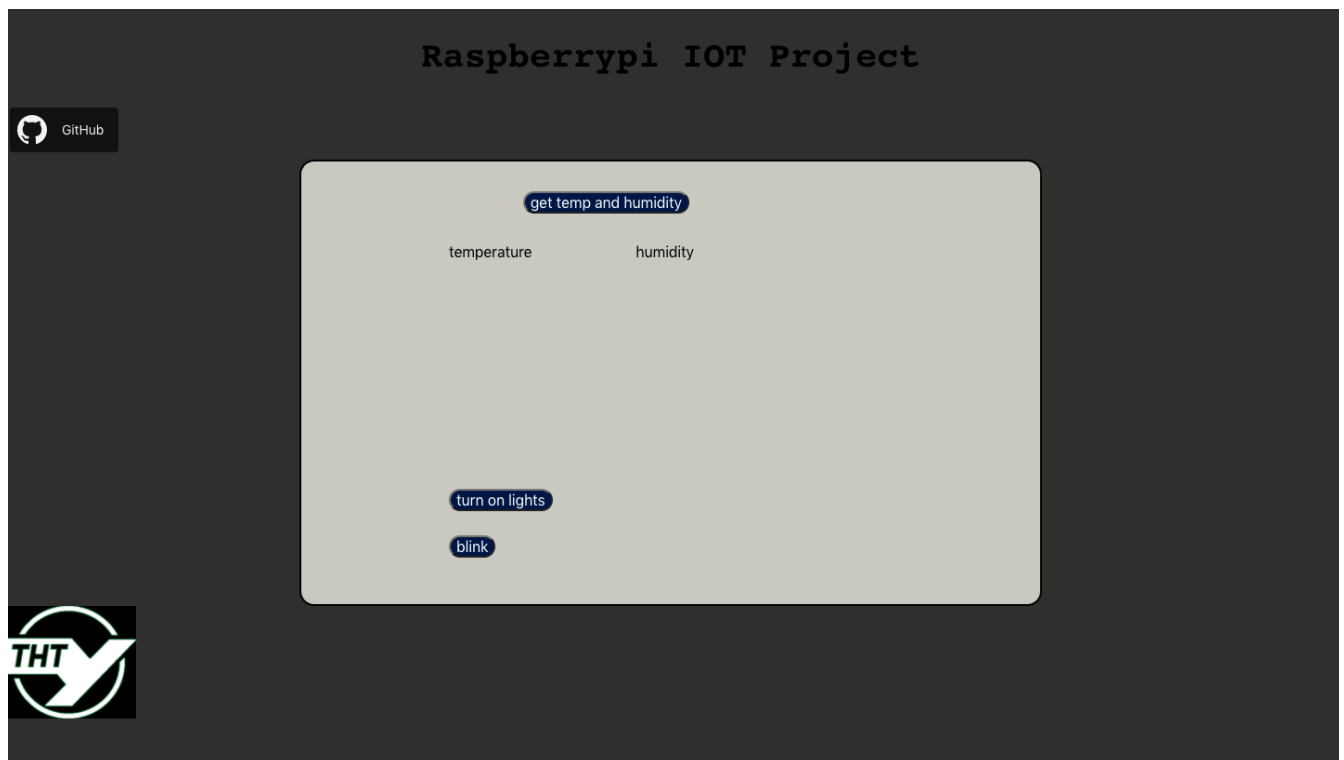


Fig. 2.8 - The User interface

Now let's test it out and click a few buttons. We can control the Led lights to turn on and off.

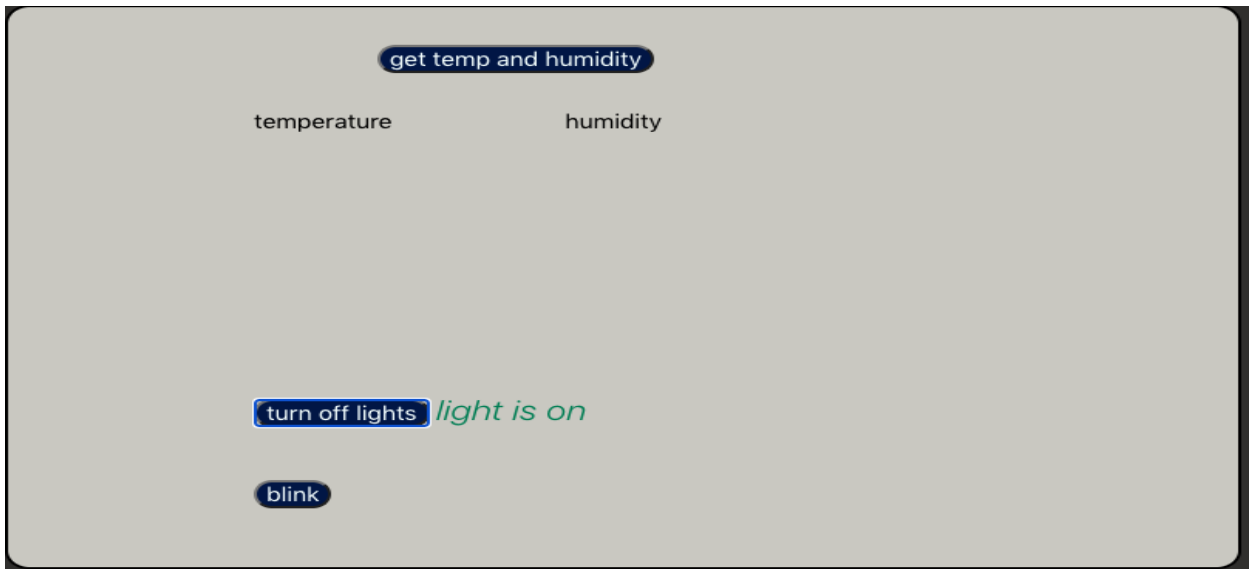


Fig. 2.9 - Testing the Led lights on from the UI (software level)

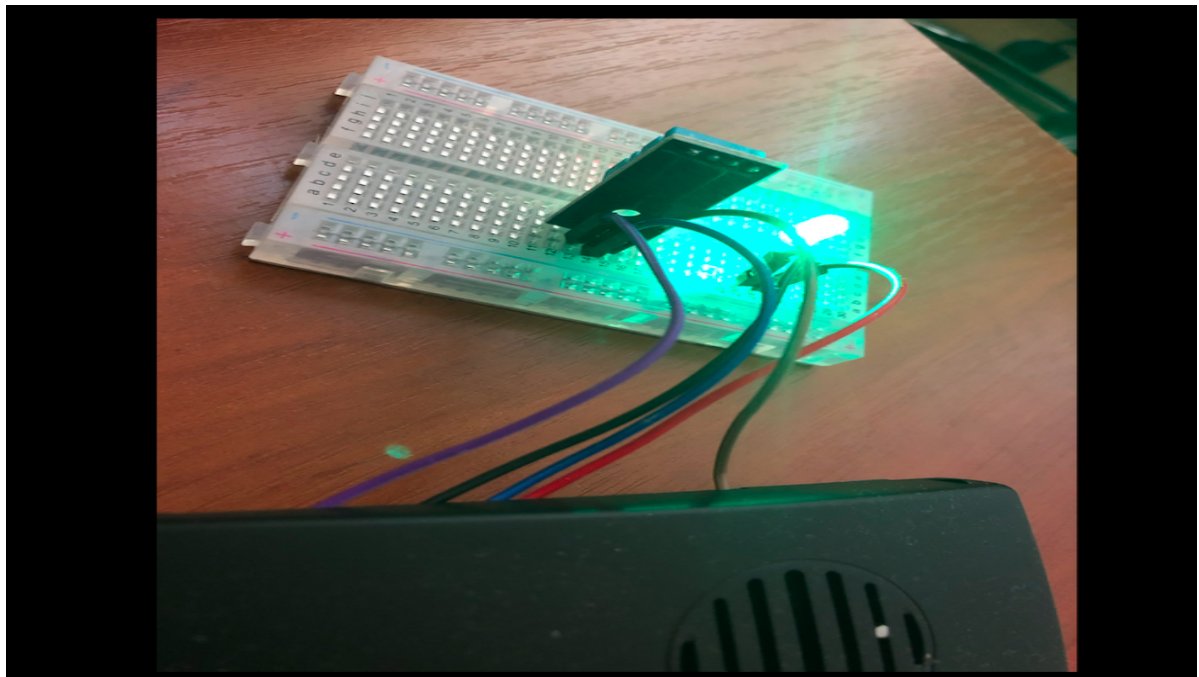


Fig. 2.10 - Testing the Led lights on from the UI (device level)

Now we make blinking of the led (Fig.2.11).

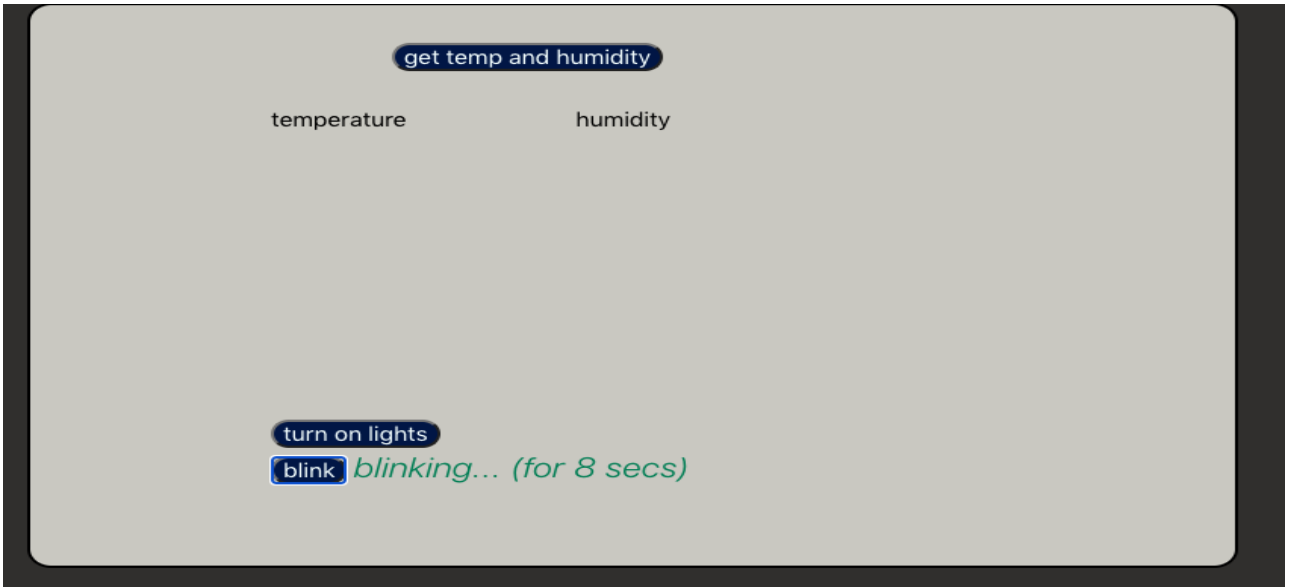


Fig. 2.11 - Testing Led blink

Now we are testing how to get temperature and humidity values from device (Fig.2.12).

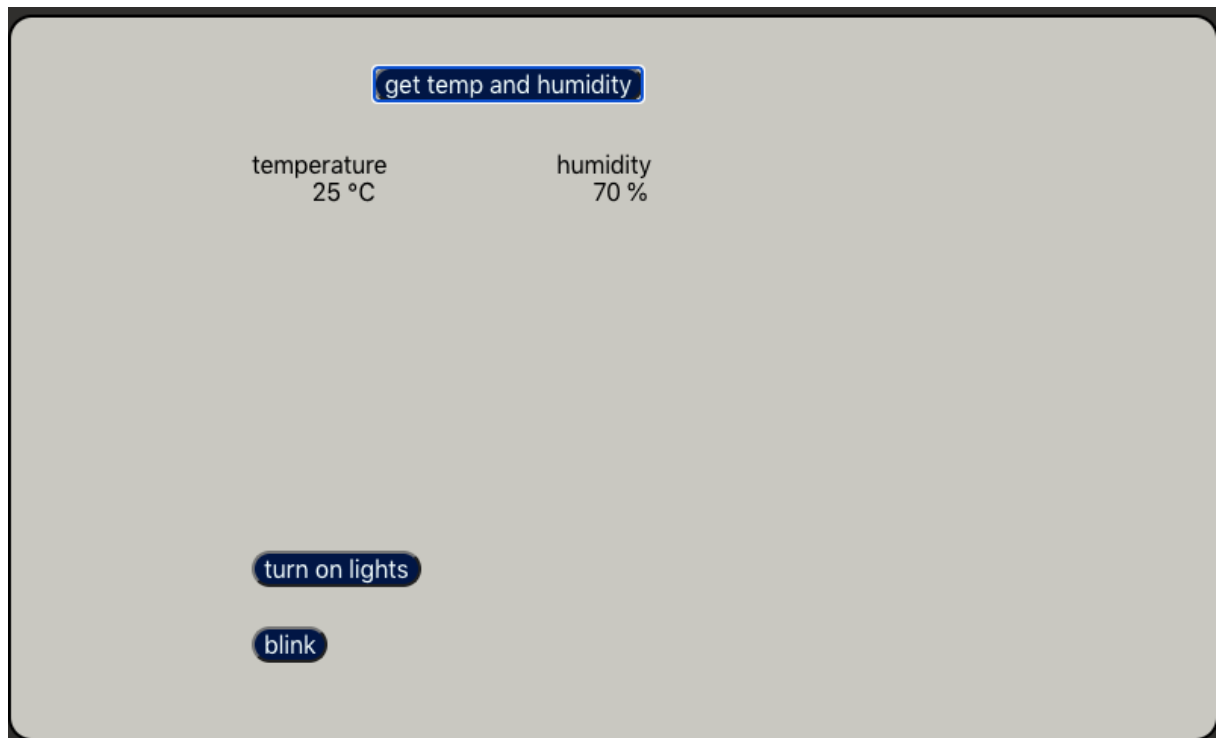


Fig 2.12 - Testing temperature and humidity sensors

That was a successful demo, our model for home automation system works, now obviously aside from not well-designed UI, there are many many more improvements that can be added to this project. Also important to note that we can

persist a database to store temperature and humidity information and display the avg daily or weekly temperature in a smart home.

The blink program can easily be implemented as an alarm system. With more sensors, we could write more programs and make the sensors work together, e.g. adding a motion sensor and setting the Led to blink when it detects movement. In a smart home, this could prove a very useful security feature.

2.2 Software deployment

2.2.1 Back-end deployment

We will analyze how to generate a docker image for our application. First we need to install and to set up Docker on the Raspberry-PI:

```
curl -sSL https://get.docker.com | sh
```

Now we have to do some Docker configurations. We're going to add our non-root user to the "docker" group, so we can have the ability to run containers which can be used to obtain root privileges on the docker host. In our case, our non root user is "pi", so we will run the command:

```
sudo usermod -aG docker pi
```

1. Set docker to auto start:

```
sudo systemctl enable docker
```

3. Restart the docker Daemon:

```
sudo systemctl start docker
```

It is important to take note that we can only run Docker images meant to work using ARM architecture. due to this, we will have to base it on an image like "hypriot". To achieve this goal we will create a "dockerfile" in the root directory of the project with the contents:

```
FROM hypriot/rpi-java
ADD project.pi-0.0.1-SNAPSHOT.jar /opt/project.pi-0.0.1-SNAPSHOT.jar
```

					CSDP 123.061.00.00 EN	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		49

```
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "/opt/project.pi-0.0.1-SNAPSHOT.jar"]
```

Once the Dockerfile is set up, then we can build the Docker image:

```
docker build -t pi_project
```

2.2.2 Front-end deployment

Client side of the project is deployed on Firebase, which was explained in the chapter 1.1.3. These are the main steps to deploying a react application to Firebase.

First step is to open the Project directory and install firebase CLI tool. Before this step we must have created firebase account.

```
npm i -g firebase-tools
```

Run “firebase login” and sign in with our account details. Initialize the project:

```
firebase init
```

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll set up a default project.

? What Firebase project do you want to associate as default? Example app (example-app-fd690)

=== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? database.rules.json
✓ Database Rules for example-app-fd690 have been downloaded to database.rules.json.
Future modifications to database.rules.json will update Database Rules when you run
firebase deploy.

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
✓ Wrote build/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
✓ Firebase initialization complete!
```

Fig 2.13 - Deployment console

To build we run the command:

```
npm run build
```

And to deploy our application we will run the command:

```
firebase deploy
```

The front-end for this project is currently hosted with the url <https://pi-project-iot.web.app/>

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		51

CHAPTER 3

TECHNICAL - ECONOMIC ASSESSMENT

It's difficult to gauge the effect of IoT will have on the economy. Let's produce a gander at the results of IoT on Gross domestic product development, development dissemination, work markets, guideline, and rivalry. Without a doubt IoT will help monetary development by some key measurements (for example Gross domestic product). Propelled economies will to a great extent advantage of these increases, we hope that creating economies will likewise have the option to see considerable additions from key IoT selection.

How might one start to qualify the impacts the Internet of Things (IoT) has and will have on the economy, when IoT itself is so difficult to characterize, it tends to be hard to gauge all the manners by which it can influence monetary patterns, legitimately or by implication

Utilizing the US for instance we will be taking a gander at four classifications of anticipated financial effect. With an end goal to parse out what impact IoT really has on the economy

- Gross domestic product Development
- Development Circulation
- Work Markets
- Guideline, Lease Chasing, and Rivalry

Gross domestic product Development

Presently, how about we consider IoT as an intermediary for machine-to-machine (M2M) interchanges. How about we utilize Total national output (Gross domestic product) development as an intermediary for residential, monetary development.

Wilderness Financial matters has anticipated that in 30 years, a 10 % expansion in M2M communications could expand the US's Gross domestic product by \$2.3 trillion. In any case, they additionally state that this amazing forecast may really undershoot financial development due to IoT

The future effect of M2M likely could be progressively extreme, our evaluations don't consider the roundabout effect of IoT on profitability, which would have an extra thump on the impact on financial development. Indeed, even with these admonitions present, the examination gives further exact proof that interest in IoT innovation expands Gross domestic product altogether." (Source: Wilderness Financial matters).

Since IoT is part to a bigger story of computerized change, its future effects on the economy can't be isolated from those bigger snares. In any case, the selection of IoT in businesses, for example, farming, vitality, producing, and urban advancement vows to return critical financial additions. Burn through administration of either assets or time can support cost reserve funds and modern efficiency. Guarantee to extend chances to adapt the sources of info and yields of IoT organizations.

Development Dissemination

Gross domestic product is useful in taking into account the typical jargon to think about economies—for instance, assisting with looking at the size of the US economy to that of the Ecuadorian economy. Be that as it may, it's not the most telling measurement of financial development. Gross domestic product per capita is another measurement that is regularly used to catch financial development, it mentions to us what development resembles comparable to populace size. It bodes well for the US's Gross domestic product to be higher than Ecuador's, on the grounds that it's a greater nation, yet how can it contrast with different nations of its size. Gross domestic product per capita can assist us with contrasting development in an increasingly corresponding manner. Indeed, even still, in light of the fact that every one of these numbers is solid, they don't generally give us any understanding into distribution of financial development. It truly could all be going toward one

industry, or one district of the world, or one class of individuals. As indicated by a report from a counseling office, it is anticipated that the advantages of embracing IoT will deliver the most incentive in "cutting-edge economies." In any case, seeing this hopefully roughly 40 percent of this worth will follow to "creating economies."

Effective financial development due to IoT relies upon some key mainstays of vital reception, as indicated by Boondocks Financial matters:

Financial development is extremely reliant on political positivity, social strength, exchange laws, advertise status, and numerous elements, however at a naturally visible level, we can foresee that there will be champs and washouts in the advanced change that are as of now continuous. Since vital appropriation is significant for creating economies, and on the grounds that development due to IoT will work for cutting-edge economies, almost certainly, IoT will fortify disparities in worldwide markets.

Work Markets

The subject of robotization and employment misfortune appears to be nearly drained now: we know work misfortune will occur, and it might be calamitous for the time being. In any case, these work stuns are just starting to appear, so a conversation about IoT and the economy would be deficient without recognizing this.

Gracefully stuns because of employment misfortune will influence the work advertise, appropriated all through time dependent on the mechanization course of events of various businesses. Division of work market will swing toward the tech area, with new openings made to oversee across the board development in the reception of innovation. Combined with this, administration division occupations, the employments that can't be robotized may encounter skip back development because of new cost investment funds at organizations that would now be able to stand to recruit more individuals and extend their organizations. High quality administrations (for example hand-fermented espresso as a help understanding, though it's unmistakably progressively proficient to computerize the blending procedure) and administrations that no one but people can quite give (for example

administrations like face to face treatment, direction advocate) may turn out to be increasingly important here and there, because of their relative wastefulness combined with popularity.

Guideline, Lease Chasing, and Rivalry

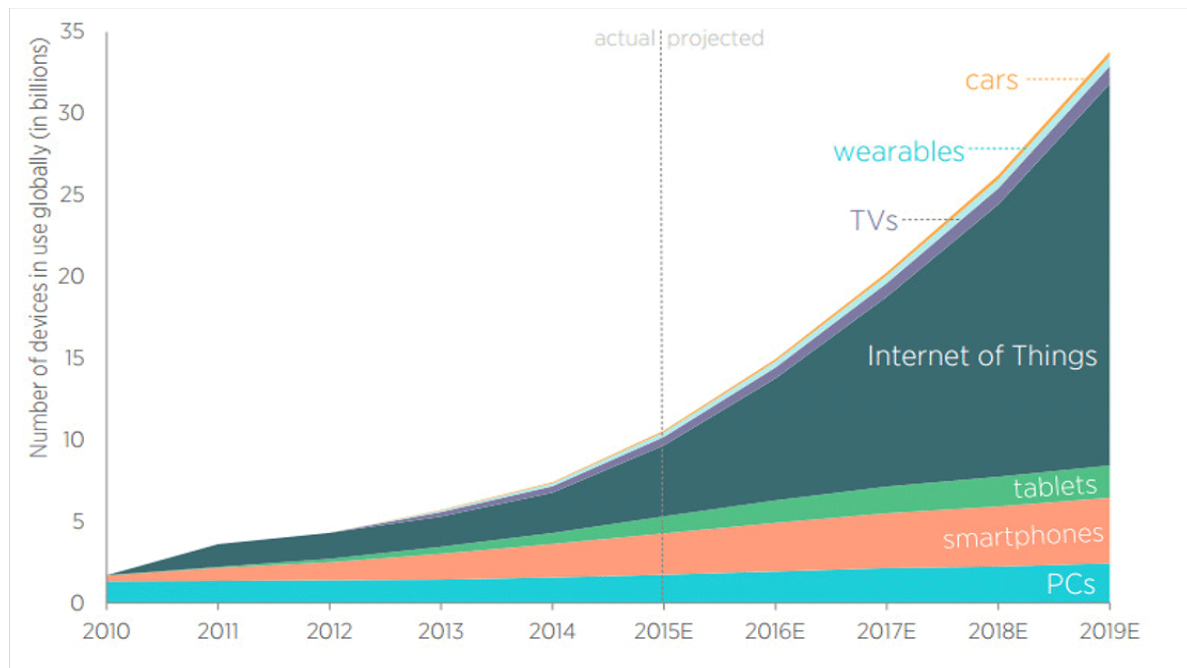
Much the same as some other innovative headways, IoT arrangements will be met with both energy and fundamental alert. Security bargains, large information concerns, and so on, and individuals are standing by to gain by it. Tech organizations are vulnerable to everything. There isn't just the political support for making IoT gauges and antitrust laws, there's likewise financial avocation.

Guideline in the Business can serve numerous jobs relying upon how it is created and executed. Guideline can support rivalry or smother it. In light of how political assets are utilized to control antitrust conduct, the making of IoT principles can serve buyers, somewhat like the FCC "free [ing] up more range which can be utilized for IoT correspondences.

Guidelines of the IoT economy will be basic to keeping up open trust and enduring development in the area. Notwithstanding, this guideline will be effective as long as it supports rivalry, reinforce security gauges, joins a different extent of IoT partners, and is adaptable to adjust to new industry advancements.

IoT is A piece of a Bigger Advanced Change. The Web of Things is one bit of a bigger riddle. Or on the other hand, rather, the development of IoT innovation will be energized by and combined with equal development in related enterprises. Truly, IoT will drive financial development, yet no, the monetary advantages of this development won't exude fairly across national or mechanical or financial lines. Indeed, work misfortune will occur and truly, there will be powers to advertise and in any case that will race to attempt to mitigate this stun. Guidelines will ideally be presented, however its subtleties and predispositions are yet to be determined.[1]

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		55



Source: John Greenough, "The Internet of Everything 2015," *Business Insider Intelligence*. Produced by Adam Thierer and Andrea Castillo, Mercatus Center at George Mason University, 2015.

Fig. 3.16 - The Internet of Everything, devices in use globally

ON World reports that a gauge of around 100 million Web associated remote lights will be ready for action by the year 2020.

Projections by industry experts present a dream of the more drawn out term where billions of once in the past torpid "things" effectively sense, react, and speak with the individuals, and situations as well as different gadgets around them. The quantity of associated buyer gadgets model wearables, televisions, and insightful vehicles will develop bit by bit however spectacularly. Keen machines and atmosphere control gadgets will turn into a progressively typical family unit protests in the coming decades. Arranged assembling, creation, and modern conveyance gadgets will to a great extent drive the extension inside the all out number of IoT gadgets.

Anticipated Monetary Advantages

The gigantic development of the absolute number of IoT gadgets is anticipated to flexibly significant financial and social advantages inside the method

of cost investment funds, esteem creation, upgrades in profitability, and general monetary development. Significantly improved robotization methods and mechanical observing will support producers and merchants to rapidly decide wasteful aspects, decline squander, and smooth out procedures. Purchaser wellbeing estimation advances will help in advancing protection wellbeing rehearses and recognize chance elements while crisis reaction interchanges can give close to moment care in perilous conditions.

The Internet of Things has the planned to upset both business and strategy. For example, Amazon is continuously utilizing associated robots to find items on its distribution center retires and carry them to laborers, setting aside both time and cash. In like manner, the clinical area can be upset by the utilization of associated gadgets watching the constant strength of patients. There are numerous ways the gadgets of the Internet of Things may influence the economy. Cell phones associated are as of now causing some impedance by allowing independent companies and people the capacity to handily take installments without costly registers or MasterCard preparing hardware; rather, they utilize an application on a regular tablet or cell phone, a simple card peruser and an online association. Administrations from organizations, for example, PayPal process these installments, taking a little cut of every exchange. The IoT is likewise expected to disturb different enterprises, for example, protection, on the grounds that the capacity to have sensors on pretty much anything can mean early discovery of a wide range of dangers and threats, and will permit them to compensate clients for embracing these gadgets.

The IoT will additionally mechanize forms and truly increment productivity, which will influence companies' main concerns. Having installed innovation which can precisely transmit conditions promptly, frequently there are significant decreases in misuse of transitory merchandise, materials lost to assembling issues, time lost to sudden machine or framework breakdowns, bringing about cost investment funds. Expanded access to information that can be assembled

progressively may likewise prompt better and all the more opportune business choices. Organizations have been known to adapt information, and now they'll have considerably more data.

The IoT ought to significantly increment different kinds of employments, particularly jobs identified with the devices themselves and the information that they assemble. This may incorporate deals and support of the related equipment, advancement of gadgets and investigation programming, and information examination, albeit a few organizations are chipping away at projects to do a ton of the systematic hard work. More IT staff and client support agents may be required for checking the administrations associated with a ton of these gadgets. The server farms fundamental for distributed computing can likewise need to extend or increment in number. Cisco evaluates that the data made yearly by IOT gadgets will ascend from 113.4 petabytes (ZB) in 2013 to 403 ZB

Machine Exploration reports in April 2015 shows that the estimation of the Web of Things was around \$900 billion of every 2014, and an anticipated ascent to \$4.3 trillion by 2024, a sum more noteworthy than the economies of the numerous nations. Different experts put the imminent incentive at \$6.2 trillion by 2025 [source: Intel]. Jumping aboard with the snare of Things may take enormous IT ventures from organizations and need changes to business forms, new hardware, and significantly more Web transmission capacity, stockpiling ability and staff to deal with the new innovation. Be that as it may, ideally the arrival on speculation will be more noteworthy.

Aside from costs, savvy urban areas, structures, and houses fueled by the snare of Things likewise can go an all-inclusive path toward cutting waste, toxins, and nursery discharges, making our advanced lives progressively maintainable as time goes on. The gadgets of the IoT can likewise give us loads of accommodations that enable us to spare our endeavors for all the more expertly and expressly compensating assignments.

Looking into these economic impacts of IoT has made me aware of the possibilities of the considerable number of things we can do due to IoT innovation, not later on, but at the present time. There are an assortment of home things that the tech savvy or cash loaded are as of now utilizing to mechanize their lives. You can purchase existing biological systems of gadgets that cooperate (apple home kit, Amazons home kit), or purchase sensors and small processing gadgets and make a portion of your own (as showed in this Project).

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		59

CHAPTER 4

OCCUPATIONAL SAFETY AND HEALTH

The role of IoT plays in wellbeing of workers can only be understood once we identify the major causes of work place accidents. As indicated by the National Safety Council, mishaps in the workplace that bring about representative injury happen at regular intervals. This shows mishaps are more typical than the vast majority would think. How about we inspect the significant reasons why work environment mishaps may occur.

Workers Lacking data about the activity dangers

A few representatives don't generally comprehend the dangers related with their work. This absence of data makes them work indiscreetly without considering their own and colleagues' security. An IoT arrangement that can be applied for this situation, is to gather information through IoT and give enough workers data with respect to work dangers.

Disregarding wellbeing techniques and directions

Carelessness is another reason for mishaps at a work place. Inability to follow wellbeing techniques and directions legitimately influences representatives' security. Vital estimates must set up to guarantee the two representatives and the executives follow all wellbeing conventions.

Interruptions at the work place

Interruptions at a work spot will consistently add to the event of mishaps. Here are a couple of instances of work environment interruptions:

- calls
- Commotion
- Email notices

- Extraordinary temperature
- Amateurish worker propensities

Poor working condition

Poor working conditions also adds to working environment mishaps that can be deadly. The development business for instance has a high pace of work related mishaps, the vast majority of which happen because of a horrible condition of the working environment. This may incorporate Uncovered wires, mess, ineffectively put away hardware, old office writing material. It is imperative to Assess the worksite before work can be started, just as the intensive investigation of the gear.

Carelessness of an individual laborer

Another neglected motivation behind why working environment mishaps are normal. At the point when workers feel they are totally in charge, they may disregard the straightforward things that may cause hurt. For example, laborers overlook wearing Individual Defensive Gear. IoT in the working environment is certainly one of the most imaginative approaches to guarantee wellbeing that your organization could embrace.

Here are a few favorable circumstances of the use of IoT programming to work environment wellbeing.

Expanding the comprehension of the workplaces with the guide of associated sensors

At the point when representatives have a superior comprehension of the workplace, mishaps are less inclined to happen. Natural sensors will be set up to distinguish any potential security danger, be that as it may, this highlights may not be sufficient to totally dodge every conceivable disaster. This is the place the utilization of associated sensors comes in, in the event that one sensor triggers, a great deal of potential risks can be recognized and taken care of on schedule. For example, organizations that make IoT gadgets can add associated sensors to

clothings like eyeglasses, head protectors, and footwear helping the articles to make and address crucial wellbeing data.

Surveilling for potential perils

A significant element of IoT programming is what is designated "prescient support", which is liable for observing hardware in the working environment. The thought here is that the product will have the option to distinguish basic disappointments in the associated gadgets, making it conceivable to complete remedial measures before any shocking mishaps.

Following the physical condition of workers to guarantee wellbeing

Ordinary assessment of organization properties combined with following of the state of being of employees(which is conceivable gratitude to significant advances in wellbeing checking innovation) can advance work environment security.

Improving security consistence

Preparing programs should be made for workers to empower them to effectively distinguish potential dangers to their wellbeing. IoT innovation can help in checking and keeping up a protected workplace, through viable record keeping. In this specific case, records of consistence preparing are assessed and the executives can ensure all workers have had satisfactory preparing.

Improving the effectiveness of salvage tasks

Every associated gadget will be prepared to do adequately distinguishing a potential mishap. In the appalling occasion of a mishap, the salvage activities can be rapidly be done, making nature more secure

Ecological sensors

Utilizing ecological sensors can help forestall mishap wounds in the work environment. particularly for mishaps associated with the inability to distinguish a potential danger to one or a few workers. IoT gadgets can assemble however much data from the workplace as could reasonably be expected.

Here are a portion of the uses of IoT sensors in the work environment.

Track, accumulate and investigate worker's information continuously

Prescient examination and simulated intelligence are getting progressively increasingly well known. This sort of innovation has been utilized in health applications that can follow representative information continuously, empowering the executives to dispose of potential hazard to their employee's wellbeing before it occurs. Moreover, if a circumstance that bargains security emerges, a reasonable arrangement dependent on close to home information investigation can be given.

Getting cautioned on injury dangers or mishaps

All IoT gadgets used to follow work environment wellbeing are associated. This implies any hazard can be immediately enrolled as an alarm and a speedy reaction managed relying upon the idea of the issue. Workers with ailments, as cardiovascular issues joined by hypertension, for the most part wear electronic gadgets model, wristbands to caution them of any dangers, an association with such frameworks matched with GPS will assist with pinpointing the specific area of the representative.

Temperature, air quality and other ecological measurements observation

These kinds of information when investigated progressively will make it conceivable to get any abnormalities. All working environments ought to have the accompanying sorts of sensors:

- Surrounding temperature sensors
- Air quality sensor
- Mugginess sensors

- Combustible gas sensor

Offer natural information and biometrics of workers between chiefs by means of WiFi, Bluetooth or Z-Wave

IoT gadgets which associated through WiFi, Bluetooth, Zigbee or Z-wave, can surrender administrators to date information on the wellbeing of their representatives. With the correct frameworks set up, all dangers for a working environment injury can be effectively identified. Once these are recognized, businesses can be immediately informed if and when any ecological sensors gets activated.

Fire anticipation with IoT innovation

In opposition to the convictions of a great many people, work environment fires are very normal. Truth be told, the greater part of these flames are coincidental and could have been really forestalled whenever distinguished before. For this situation, IoT can accomplish the accompanying:

- Fire wellbeing sensors that can identify spikes in temperature to offer signs to when a fire may spread.

- Monitoring the vitals of firemen- the wellbeing of firefighters's can be undermined while they firefight, thusly, having programming that can follow their vitals and help recognize any medical issues that may bring undesirable results is essential. PHASER is perhaps the best innovation right now accessible which can follow all vitals, for example, internal heat level, circulatory strain, and heartbeat. Data would then be able to be sent back to the station to ensure no representatives endure any wounds during the activity.

- Improving the upkeep of fire tracking frameworks. New innovation that empowers following of any potential fire perils can be introduced and it report them if essential.

- Establishment of solid ready frameworks at first; for example, shrewd innovation like Home can help forestall fires by utilizing brilliant advances to

interface your whole frameworks together, Z-wave and ZigBee are probably the best association conventions to utilize.

Utilization of Cutting edge Investigation

Propelled investigation is the utilization of profoundly dependable techniques for information examination to create usable data that discovers the arrangements of occasions that craftsmanship yet to happen. It does this by making "consider the possibility that" figurings (ascertaining the chance of something occurring by utilizing an alternate arrangement of qualities or information. Progressed examination have definitely settled on recognizing issues and dynamic simpler than it used to be.

The best instances of data analytics include:

- Demonstrative investigation
- Illustrative investigation
- Prescriptive examination utilizing man-made intelligence and enormous information

- Prescient examination

Prescient demonstrating

this is a strategy for cutting edge investigation that can deliver an alternate number of potential results from information mining. Be that as it may, so as to do this you will require programming that can without much of a stretch break down the information gave, that is the place IoT representative security programming comes in. Workplace mishaps can likewise be anticipated by this type of cutting edge examination, which considers all the potential frameworks that may fall flat and lead to injury.

Breaking down ecological and representative information continuously

One of the viable approaches to distinguish and respond to mishaps on time is by having a framework exclusively intended to follow both representative and natural information. Worker information, similar to individual data, capabilities,

					<i>CSDP 123.061.00.00 EN</i>	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		65

any physical and emotional wellness conditions can be followed continuously, all gratitude to AI, all past data can be taken care of into a framework which can investigate this information and anticipate a potential disappointment. With a live examination of information, issues can be recognized rapidly by ready frameworks.

Social affair representative information and applying the correct information perception strategies to it likewise helps in distinguishing any medical issues that may occur for a worker, It's additionally essential to accumulate and examine conduct information, more organizations will have the option to stay away from future issues when this sort of information examination is paid attention to. Offering security arrangements dependent on information investigation

When different instruments, for example, prescient rebuilding are set up, work directors will be able to discover answers for issues before they occur. For example, shortcoming in a security arrangement of an organization can be anticipated dependent on information examination, and once this issue has been distinguished, measures can be taken to fix it. The entirety of this is conceivable gratitude to huge information examination procedures and their solid visual portrayal. By gathering accurately composed and broke down information, supervisory crews can have all the data they have to settle on a significant choice. The more data accumulated and deciphered, the better mechanical IoT wellbeing systems can be produced to support workers.

An examination uncovered that about 83% of assembling pioneers see expanded profitability from wellbeing programs. Probably the most ideal ways producing organizations can manage security concerns and dangers is by improving specialist wellbeing with IoT in assembling is truly.

- Embedding sensors into apparatus that will gather information on the working state of machines. On the off chance that any machine is probably going to breakdown or not work suitably, the data is sent specialized controllers who would then be able to take quick measures, sparing a great deal of lives.

- Giving laborers wearables can help guarantee their wellbeing and security in the working environment. These wearables with sensors can follow temperature changes, stickiness levels, unsafe gases, substance introduction, and furthermore clamor levels, they can likewise can check wellbeing parameters of laborers, things like pulse varieties, feelings of anxiety, and other information identified with their prosperity, any early indications of medical issues can be sent to the wellbeing and security directors immediately where fitting prudent steps would then be able to be taken. Guaranteeing the security of representatives working in remote areas.

- IoT can be of incredible assistance in guaranteeing fast salvage activities during a fiasco. Sensors inserted wherever will accumulate ongoing information from the site which can be utilized by the salvage group to plan and lead salvage and recuperation procedure on schedule. This converts into higher odds of sparing the lives of more specialists.

With rise of IoT, fabricating firms will observe improved working environment perceivability. The capacity to follow everything about the workplace will help fabricating organizations limit mishap rates, improve crisis reaction time, and decrease consumption on specialist remuneration.

IoT has genuinely supported organizations over an assortment of enterprises to smooth out their activities and increment efficiency, ongoing information assortment and detailing additionally can possibly definitely improve working conditions for representatives everywhere throughout the world. As per the National Wellbeing Chamber, business related wounds that necessary clinical meetings were about 4.5 million of every 2017, which cost an expected measure of \$161.5 billion in clinical costs, compensation and profitability misfortunes, authoritative costs and the sky is the limit from there. Representatives of development, producing and farming businesses experienced the absolute most elevated quantities of wounds and passings, yet there has been an expansion in

working environment security as new advances become increasingly moderate and simple to actualize.

Despite the fact that IoT gadgets can't totally shield laborers in high hazard businesses from each classification of danger, they do support office and worksite chiefs address an enormous number of preventable dangers.

Five emerging innovations that can procedure and recognize working environment patterns with more noteworthy exactness, including:

- Profound jump investigation
- Man-made reasoning
- Distributed storage
- Large information preparing
- Interconnected gadgets

Together, these advancements are equipped for collecting a monstrous measure of natural information that can prompt noteworthy answers for probably the most widely recognized working environment risks, for example, harsh moistness, unnecessary commotion levels and undesirable air quality, high temperatures et.t.c.

Prescient hardware support

Modern hardware requires routine support and fixes throughout their operational life expectancy, yet the issue is that it's not generally clear when a bit of gear needs some work. Laborers working these incredible machines face various types of dangers if an inner segment were to suddenly come up short, or if a security include were to breakdown during support. Counteraction of enormous scope mishaps has consistently been a significant worry for creation and assembling offices, this has lead more organizations to grasp a time sensitive support plan, and despite the fact that this procedure has assisted with decreasing genuine and surprising breakdowns, it could likewise bring about the substitution

of machine segments that have not yet exhausted, costing a great deal of cash over the long haul.

an article distributed via Mechanization World toward the beginning of Spring shows that a large number of the main robotization merchants have just begun implanting improved wellbeing and checking highlights into their frameworks, instead of turning them out in the after creation stage. By doing this hardware producers are wanting to help prescient support methodologies that takes into consideration less vacation without risking the prosperity of gear administrators.

Why adaptable IoT framework is critical

As we keep on observing the development and improvement of IoT innovation, worksite administrators will have the option to send a large number of wellbeing centered gadgets that will guarantee laborers are educated about the specific states of their creation hardware, alongside any perils that may happen. Be that as it may, before modern players can begin utilizing these gadgets, they should from the outset guarantee that their systems administration frameworks have the transfer speed abilities to deal with the preparing of all the created information. By exploiting distributed computing, huge information examination and IoT-empowered gadgets depends adaptable specialized apparatuses that can be scaled without the requirement for a total framework outstrip[9]

CONCLUSIONS

IOT is a promising and exciting technology, it is rapidly being adopted by both businesses and consumers, a lot is yet to be learned about how much effective we can make “things” communicate, the possibilities are truly endless. There are tons of emerging start-ups providing IoT solutions. And even big companies are adopting it as well. Amazons Alexa is a very good example, which works seamlessly with a lot of other home kit devices. Imagine a world where all your home tasks are automated, and we can all make better use of that time.

Home automation systems have truly struggled to find a mainstream audience, partly because they require their users to be a little tech savvy, but the adoption from big companies and the fast proliferation of smart devices provide an easier way for even tech novices to be a part of this amazing and exciting future.

In these diploma thesis developed prototype of home automation system which allows to:

- measure temperature and humidity in the house;
- manage heating based on the average temperature and humidity in the house;
- predict further natural gas or electricity consuming based on the long-term saved data;
- turn on and turn off the light in the house.

This project was developed by using hardware and software components:

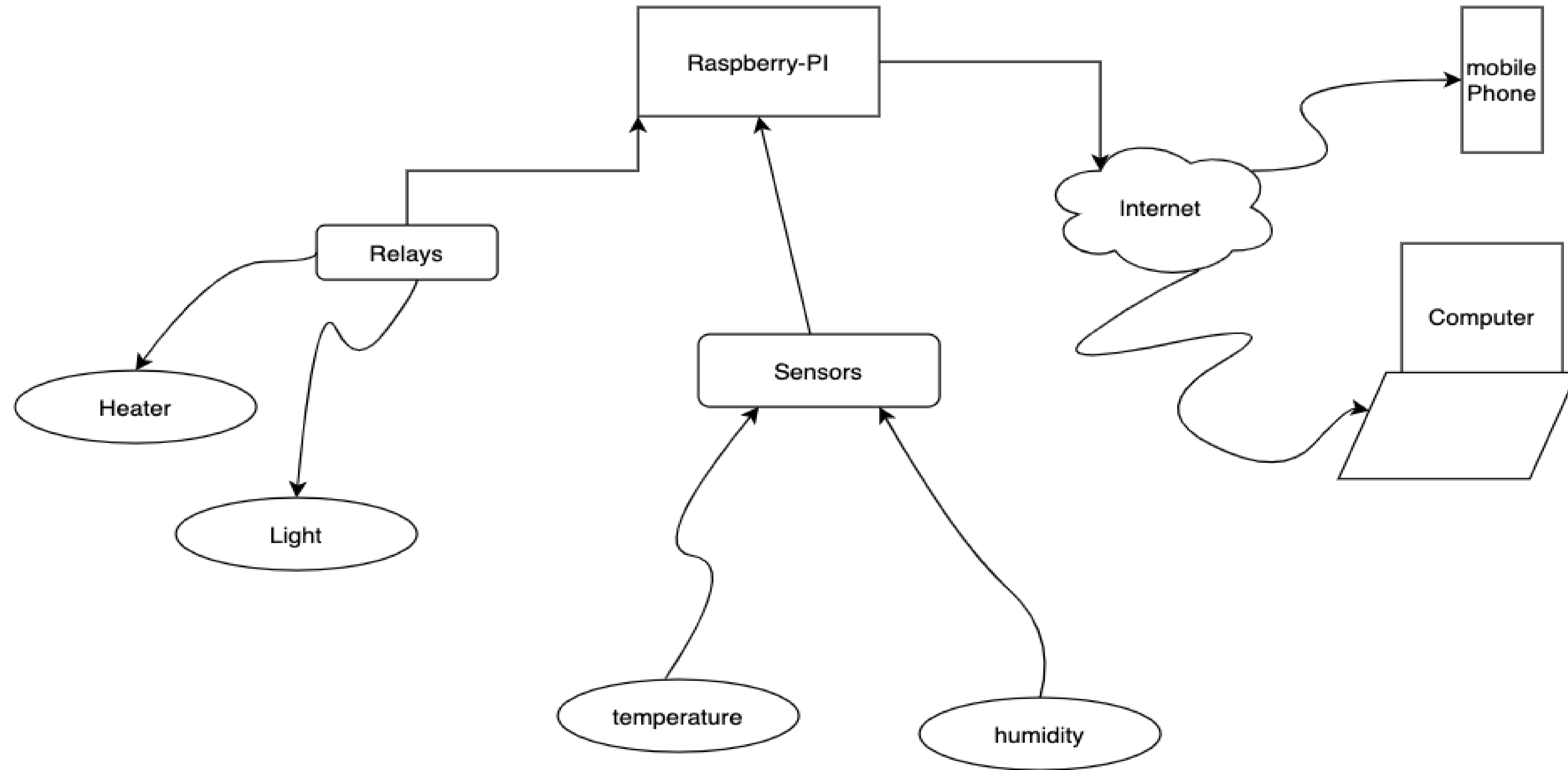
- Raspberry PI 3B+;
- DHT11 temperature and humidity sensors;
- leds;
- Java 11 SDK;
- Pi4J v 1.3 library;
- Spring Boot v 2.3 framework;
- React v 16.13.1.

					CSDP 123.061.00.00 EN	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		70

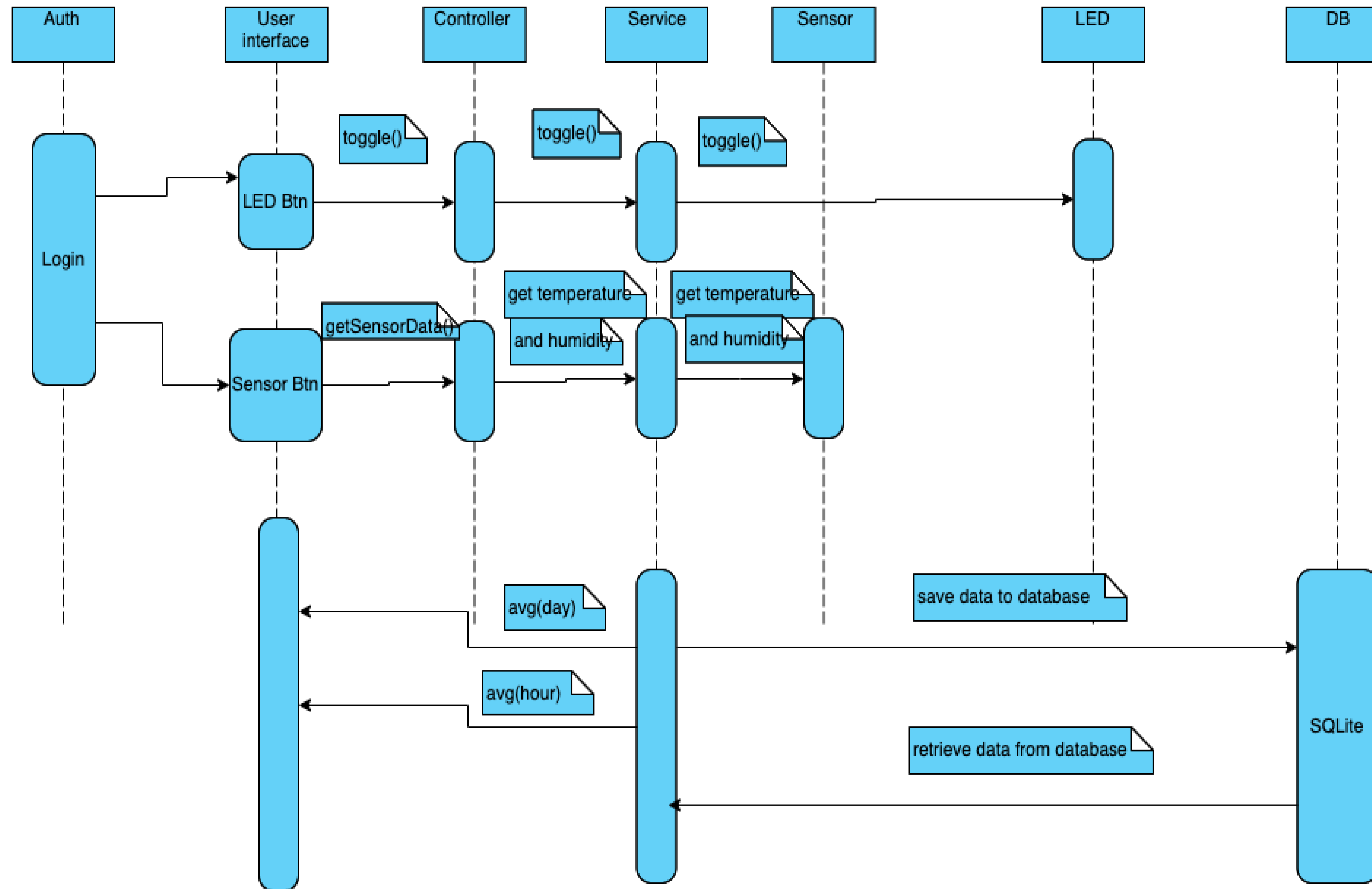
REFERENCES

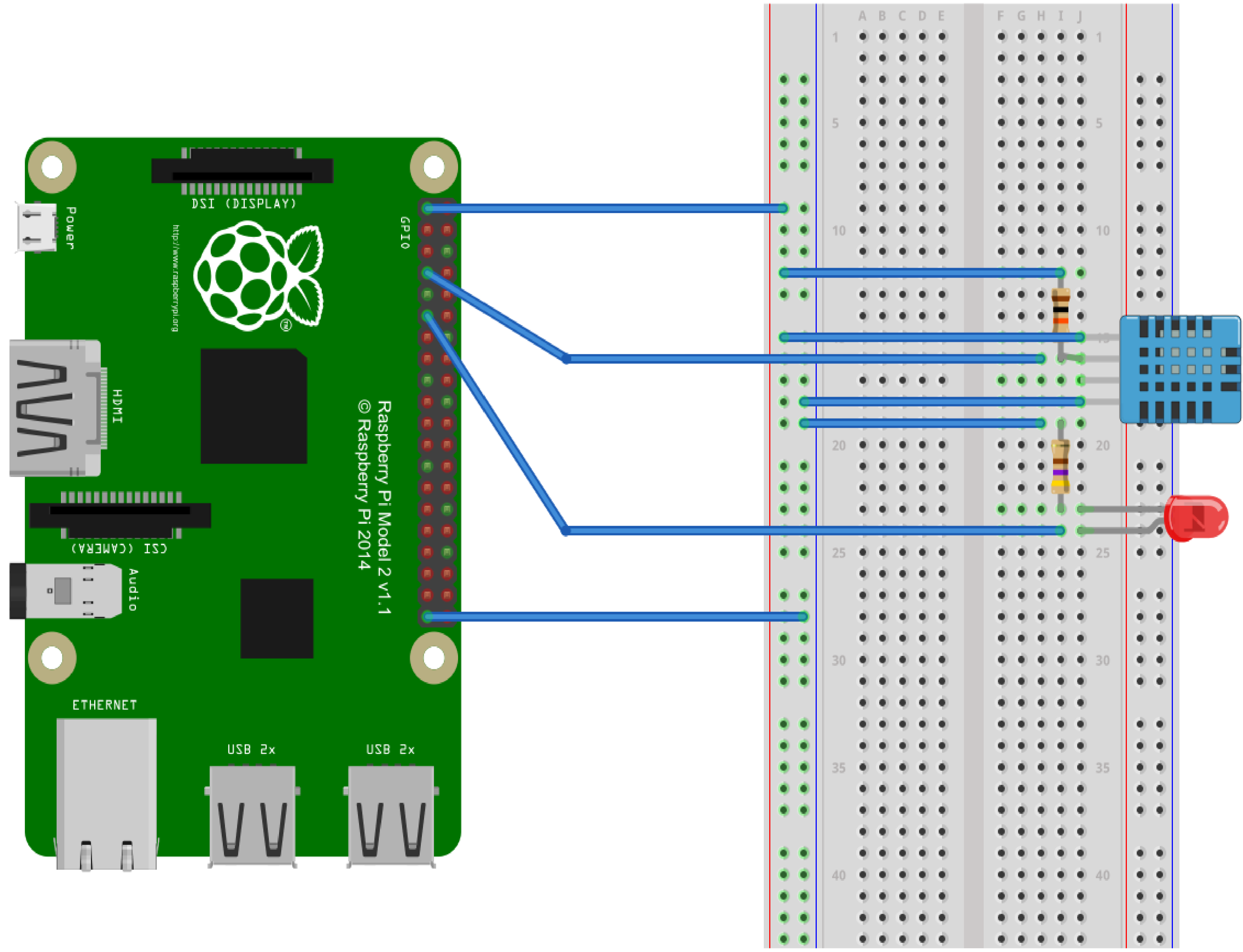
1. Sloan H. How IoT Impacts the Economy / Hannah Sloan// [Electronic Resource] Access mode: URL: <https://www.iotforall.com/how-iot-impacts-economy/>
2. Cosmina I. Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools / Iuliana Cosmina, Rob Harrop, Chris Schaefer, Clarence Ho// Apress, New York, 2017, 849 p.
3. Rouse M. MQTT (MQ Telemetry Transport) /Margaret Rouse// [Electronic Resource]Accessmode:URL:<https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport>
4. Thierer A. Projecting the Growth and Economic Impact of the Internet of Things / Adam Thierer, Andrea O'Sullivan// [Electronic Resource] Access mode: URL: <https://www.mercatus.org/publications/technology-and-innovation/projecting-growth-and-economic-impact-internet-things>
5. RESTful Web Services - Introduction [Electronic Resource] Access mode: URL: https://www.tutorialspoint.com/restful/restful_introduction.htm
6. Rouse M. REST (REpresentational State Transfer) /Margaret Rouse// [Electronic Resource] Access mode:URL:<https://searchapparchitecture.techtarget.com/definition/REST-REpresentational-State-Transfer>
7. Gupta L. Java Tutorial /Lokesh Gupta// [Electronic Resource] Access mode: URL: <https://howtodoinjava.com/java/basics/java-tutorial/>
8. SQLite Home Page [Electronic Resource] Access mode: URL: <https://www.sqlite.org>
9. Burkhalter M. How IoT devices are improving workplace safety/ Max Burkhalter [Electronic Resource] Access mode: URL: <https://www.perle.com/articles/how-iot-devices-are-improving-workplace-safety-40183647.shtml>
10. How to Improve Workplace Safety with IoT/ Digiteum Team// [Electronic Resource] Access mode: URL: <https://www.digiteum.com/iot-workplace-safety>

					CSDP 123.061.00.00 EN	<i>Page</i>
<i>Ch.</i>	<i>Page</i>	<i>Nº docum.</i>	<i>Sign</i>	<i>Date</i>		71

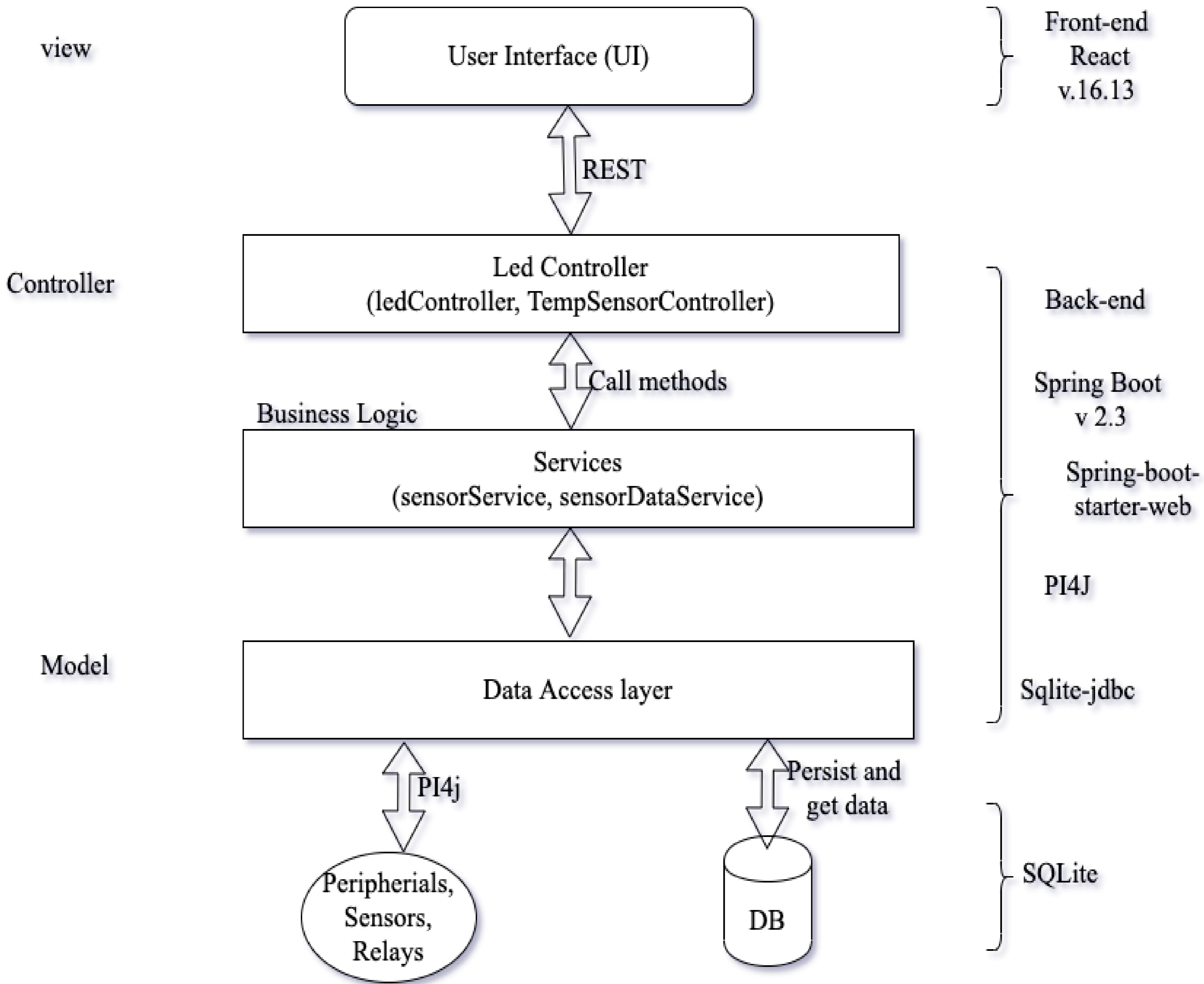


				CSDP 123.061.00.05		
Ch	List.	Nr docum.	Sign	Date	Home Automation System Components	
Develop		Ugryu V.O.			Letter	Mass
Supervisor		Lutskiv A.M.			Scale	
Reviewer					Page	Pages
N.Contr.					TNTV. dept.CE, ICI-42	
H. of Dept.		Osukhivska H.				





				CSDP 123.061.00.04			
Ch	List.	Nr docum.	Sign	Date	Letter	Mass	Scale
					DHT11 and LED Connection to Raspberry Pi		
					Page	Pages	
					TNTY. dept.CE, ICI-42		



					CSDP 123.061.00.01		
Ch	List	Nr docum.	Sign	Date	Home Automation System Application Layers		
Develop		Ugwu V.O.			Letter	Mass	Scale
Supervisor		Lutskiv A.M.					
Reviewer					Page	Pages	
N Contr.					TNTY. dept.CE, ICI-42		
H. of Dept.		Osukhivska H.					

Comparison of different Raspberry-PI devices

	Raspberry pi 1	Raspberry pi 2	Raspberry pi 3	Raspberry pi 4
SOC Type	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837B0	Broadcom BCM2711
GPU	VideoCore IV 1080p@30	VideoCore IV	VideoCore IV	VideoCore VI
CPU clock	700 MHz	700 MHz	1.4 GHz	1.5 GHz
RAM	512 MB	1 GB	1 GB DDR2	1 GB , 2 GB, 4 GB LPDDR4
Wi-Fi(onboard)	-----	-----	2.4GHz and 5GHz 802.11 b/g/n/ac	2.4GHz and 5GHz 802.11 b/g/n/ac
BlueTooth (OnBoard)	-----	-----	4.2, BLE	5.0
Prices	\$20	\$35	\$55	\$80

					CSDP 123.061.00.02			
Ch	List	Nr docum.	Sign	Date	Comparison of Different Raspberry-PI Devices	Letter	Mass	Scale
Develop		Ugwu V.O.						
Supervisor		Lutskiv A.M.						
Reviewer						Page	Pages	
N Contr.								
H. of Dept.		Osukhivska H.						TNTY. dept.CE, ICI-42

Network API Comparison			
Capability/Architecture	REST	GraphQL	gRPC
Compressed Payload	gzip	gzip + requested fields	Binary
Asynch Requests	HTTP	Requires AMQP	HTTP/2
Reduced Requests	Requires a Composite API	Retrieves just what is specified in the query	Write your funtion
Market Standards	Lots of Tools and Standards	Experimental Use	Experimental Use
Reuse	Based on Business Domains	Specific function	Specific function
Compatability with Event Architecture or Reactive	Synchronous Requests	Mutation Syntax and Asynchronous Calls	Asynchronous Calls and Syntax flexibility
Optimized Payload Parsing	JSON Serialization	JSON serialization	Binary Serialization