

Body Temperature Detecting System

MING GAO, Washington University in St. Louis, USA

JUNLIN LONG, Washington University in St. Louis, USA

YUXUAN TAO, Washington University in St. Louis, USA

In these two years, the world is surrounded by the COVID-19, it has brought a great negative influence on our lives and economy, but in the United States most of people is not strict about the prevention and control of COVID-19, in many places people even do not wear masks, so we do not know if someone around us has a fever or risk of COVID-19. In China, everyone is using a health code that scans their temperature when entering every public place to confirm their health, and also records everyone's trace information in this way, so that when an infected person is found, the system can make an accurate determination of the infected person's whereabouts based on this trace data. In turn, the infected person and his contacts will be controlled to prevent further spread of COVID-19. Such prevention and control measures can also be used in the U.S. environment. Taking WashU as an example, we would like to design a system that can identify and measure the temperature of students by their student cards to monitor the temperature of students entering the building and collect the trace data for testing at the same time.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; **Redundancy**; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Body temperature detection, COVID-19, Amazon Web Services, Amazon Relational Database Service, Raspberry pi, JSP web page

ACM Reference Format:

Ming Gao, Junlin Long, and Yuxuan Tao. 2018. Body Temperature Detecting System. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 6 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Our system is used to detect the body temperature and record the trace information. We designed the hardware device as a door access control system, set up at the entrance of each public buildings, and when entering the building, you need to swipe your student ID card and measure your body temperature. If the measured body temperature is normal, the system will give you the access to get into. At the same time, the system should also have an error correction mechanism, if the visitor only swipe student card without detecting the temperature, the system will prompt to retake the temperature, and will not give the access. And if someone takes a hot drink to swipe the card, if the system incorrectly detects the temperature of the hot water cup, it will not be seen as a fever for mistake, because the temperature of hot water is much higher

Authors' addresses: Ming Gao, Washington University in St. Louis, St. Louis, USA, chris.m.gao@gmail.com; Junlin Long, Washington University in St. Louis, St. Louis, USA, junlin.l@wustl.edu; Yuxuan Tao, Washington University in St. Louis, St. Louis, USA, tao.yuxuan@wustl.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/8-ART111 \$15.00

<https://doi.org/10.1145/1122445.1122456>

than the human body temperature, the system will identify the temperature beyond the normal human body temperature range and prompt a new temperature measurement. If there is a person with a fever wants to enter the building, the system will detect the high temperature and prompt a ban on entry. Each temperature measuring device will upload relevant data to the database in real time, including ID, temperature, time, and location. Also, when high temperature data upload, the system will automatically find out the travel data of the fever person for the last three days and identify potential contacts who were in the same building with him on the same day. Finally, the system will send an email warning to these contacts, reminding where they may have contact with the fever person, and please get a test as soon as possible to eliminate the risk of COVID-19. We also have a web page as a visual interface to show the content of the database in real time, and it can also be used to search and edit the information in the database, for example, if a fever person is found, all the recent trace data of this person can be searched directly.

2 GOALS AND REQUIREMENTS

For the system, there are several main functions that need to be implemented, such as:

- Realization of the NFC card information
- The contactless temperature detection
- Display temperature and ID information
- Real-time information sending from the device to the database
- visualization interface to show and operate information in database
- Search program which is activated by high temperature
- Automatically send emails, etc

When high temperature is detected, the following actions need to be triggered.

- Cancel the access of ID card
- Summarize the traces of the ID
- Find out people contacted with the fever person
- Send warning E-mails to contacted people

Based on the above requirements, we design the system for each function

3 DESIGN

3.1 System framework

Raspberry pi send data to the database in AWS, and the search program on cloud 9 get data from RDS and find out target people then send email to them by SNS, as you can see in figure 1.

3.2 Hardware design

In the design of hardware, we used Raspberry pi as MCU, LCD1602 to display information, MLX90614 as temperature sensor because it

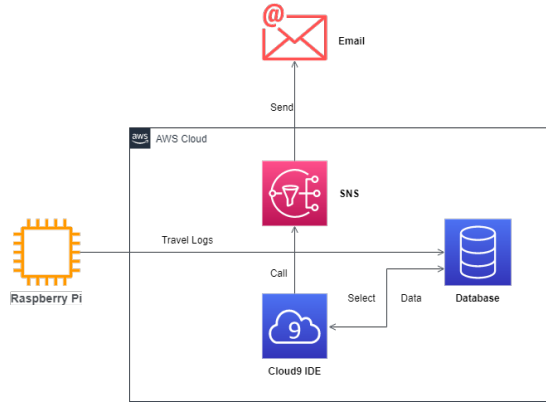


Fig. 1. System framework

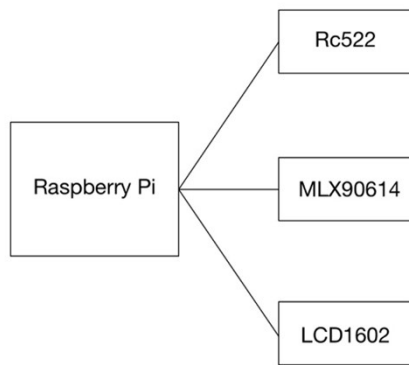


Fig. 2. Hardware Design

can detect temperature without touch, and RC522 as NFC module to identify ID card. And use green and red light to represent access or not, as you can see in figure 2.

3.2.1 MLX90614. The MLX90614 is an Infra-Red thermometer for non-contact temperature measurements, as you can see in figure 3. Both the IR sensitive thermopile detector chip and the signal conditioning ASIC are integrated in the same TO-39 can. Integrated into the MLX90614 are a low noise amplifier, 17-bit ADC and powerful DSP unit thus achieving high accuracy and resolution of the thermometer. The thermometer comes factory calibrated with a digital SMBus output giving full access to the measured temperature in the complete temperature range(s) with a resolution of 0.02°C. The user can configure the digital output to be PWM. As a standard, the 10-bit PWM is configured to continuously transmit the measured temperature in range of -20°C to 120°C, with an output resolution of 0.14°C. Its performance is suitable for use as a temperature sensor for this system.

3.2.2 **RC522.** The RC522 RFID module based on MFRC522 IC from NXP is one of the most inexpensive RFID options. It usually comes with an RFID card tag and key fob tag having 1KB memory. It can be used to identify the ID cards' information, as you can see in figure 4.

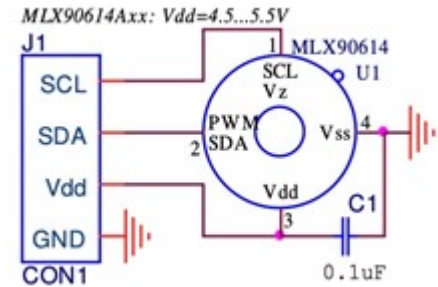


Fig. 3. Simplified block diagram of the MLX90614

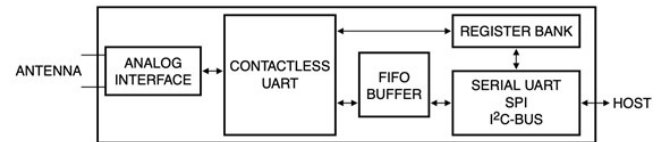


Fig. 4. Simplified block diagram of the MFRC522

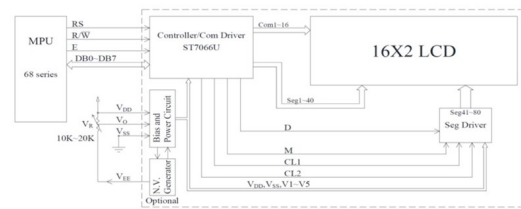


Fig. 5. Simplified block diagram of the lcd1602

3.2.3 LCD1602. LCD1602 character-type liquid crystal display is a kind of dot matrix module to show letters, numbers, and characters and so on. It's composed of 5x7 or 5x11 dot matrix positions; each position can display one character. There's a dot pitch between two characters and a space between lines, thus separating characters and lines. The model 1602 means it displays 2 lines of 16 characters. It can be used to display temperature and id information in this device, as you can see in figure 5.

3.3 Database design

A database is required in our system for restoring information obtained from sensors, and other information such as users' personal information and location information. Information is divided into mainly three categories:

- (1) Travel logs obtained from sensors, including body temperature of user, user (card) ID, the location where he visited, the date and accessibility of him.
- (2) User information, including user ID, his name and contact information (email).
- (3) Location information, including location ID and its name.

Thus, we designed a collection of three tables to restore each category of information in a database. In order to restore and present

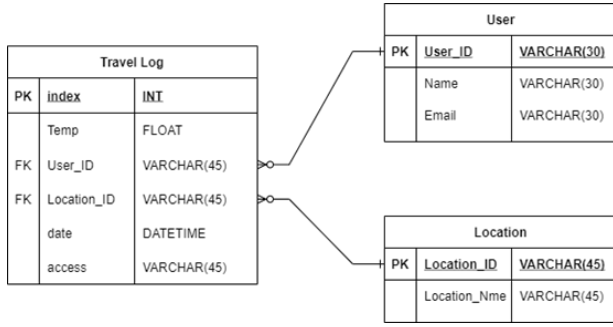


Fig. 6. Entity relationship diagram

the relations between each table, a relational database is need. As you can see from figure 6, the entity relationship diagram of our database. The name of each table and entities are listed below:

- (1) TravelLog: index (as primary key), Temperature, User ID, Location ID, Date, Accessibility
- (2) User: User ID (as primary key), Name, Email
- (3) Location: Location ID, Location Name

3.4 Functions based on AWS

To achieve our goals of selecting fevered people and sending warning emails for those who may have close contact with them, two main functions are required in our system:

- (1) Selecting targeted people: According to the record of fevered person, the system will automatically select people who may have close contact with him in the last three days.
- (2) Sending warning emails: For people selected from the automated-selecting system, send emails to warn them about this situation.

3.5 User interface design

As mentioned in the plan, we intend to make an interactive interface that can be used to operate the database, so that we can change or find the required information without opening the database, such as changing the user's name, ID and email, or finding the required user's whereabouts information, which will greatly increase the convenience of our products, and people who can't operate the database can also operate the required information.

4 IMPLEMENTATION

4.1 hardware buildup

The hardware pinout diagram is shown in table 1.

LCD1602 and MLX90614 are using the same pins.

Then we connect modules with raspberry pi.

The physical hardware setup diagram is shown in figure 7:

4.2 Raspberry Pi program function

On the Raspberry Pi, the code is mainly used for:

- (1) screen display
- (2) NFC module
- (3) temperature detect module

Module	Module Pins	Raspberry Pi Pins	Notes
RC522 Header	3.3V	1	3.3V
	RST	22	GPIO25
	GND	6	Ground
	IRQ	–	Not connected
	MISO	21	GPIO9
	MOSI	19	GPIO10
	SCK	23	GPIO11
	SDA	24	GPIO8
LCD1602	SDA	3	SDA
	SCL	5	SCL
	VCC	17	3.3V
	GND	25	Ground
MLX90614	SDA	3	SDA
	SCL	5	SCL
	VCC	17	3.3V
	GND	25	Ground
LED Lights	Green	11	GPIO0
	Red	13	GPIO2

Table 1. Hardware pinout diagram

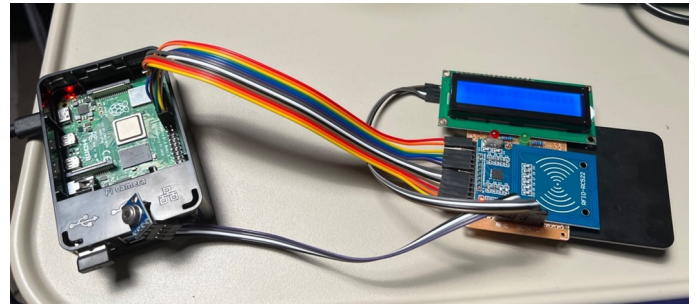


Fig. 7. Physical hardware setup

- (4) database communication.

The main execution logic of the program is as follows:

- (1) Program start-up:
 - (a) The screen enters the initial detection, displays the setting text, and controls the backlight switch
 - (b) Connect to the database
 - (c) NFC enter wait
- (2) NFC card swipe, when the NFC detects the card, the program starts to execute:
 - (a) Read the NFC card number
 - (b) Turn on the temperature detection module
 - (c) determine whether the temperature is normal
 - (d) The screen displays the card number and whether it is passable or not
 - (i) Judgment condition: temperature less than 36 or higher than 42, prompt to re-swipe the card
 - (ii) greater than 37.5 rejected
 - (iii) 36-37.5 is passable
 - (e) Upload the test data result

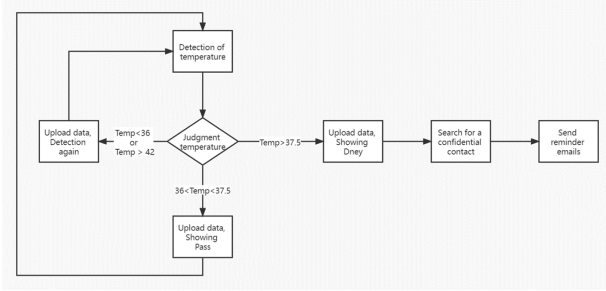


Fig. 8. Program block diagram

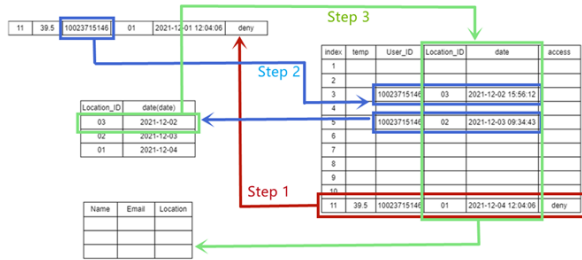


Fig. 9. Steps for selecting targeted people

(3) Program shutdown:

- Close the database connection
- Release the Raspberry Pi to use the channel

The program block diagram is shown in figure 8.

4.3 Database and function implementation

As we mentioned in section 3.3, a relational database is needed in our system. Since AWS is required in the rest of our system, Amazon Relational Database Service (Amazon RDS) would suit our demands well. Amazon RDS provides an environment to easily manage relational databases in the cloud. We implemented our database in Amazon RDS based on MySQL, one of the most widely used database engine.

For the implementation of functions, to select targeted people, three main steps needed to be achieved, as you can see in figure 9. First, from the record of fevered person, we can receive the information of his user ID. Second, according to the user ID, if we go back to the TravelLog table, we will select a list of locations where and when he had visited. At last, on the basis of each pair of visited location ID and date, after searching through the TravelLog table, the information of targeted people: name, email, and visited location are obtained.

To send warning emails, we implemented this function by using AWS Cloud 9, a cloud-based IDE to deploy and configure our programs. And Amazon Simple Notification Service (Amazon SNS), a messaging service provided by AWS to send our alert emails. In cloud 9 we deploy a program, for each 100 seconds, it will automatically detect whether someone has fever in last 100 seconds records. If detected, the program will execute selecting function and call Amazon SNS APIs to send warning emails to targeted people.

index	Temp	User_ID	Name	Location_ID	Location_Name	date	access
1	36.4	1223232033	Andy	01	Olin Library	2021-12-04 08:43:30	welcome
2	37.0	15158244167	Tom	02	Jubel Hall	2021-12-04 08:44:02	welcome
3	36.9	71132185180	Chris	02	Jubel Hall	2021-12-04 08:46:10	welcome
4	36.4	215137179181	Tom	01	Olin Library	2021-12-04 10:12:24	welcome
5	36.0	3577171181	Emma	01	Olin Library	2021-12-04 10:12:24	welcome
6	37.0	151108163181	Olivia	03	Saver Hall	2021-12-04 10:14:10	welcome
7	36.6	155196177181	Alex	05	Brown Hall	2021-12-04 10:23:11	welcome
8	36.5	313623163	Sophia	05	Brown Hall	2021-12-04 10:23:11	welcome
9	36.7	1722219926	Charlotte	05	Brown Hall	2021-12-04 10:27:11	welcome
10	37.1	129925927	Amelia	06	DUC	2021-12-04 16:42:13	welcome
11	36.8	10023715146	Logan	06	DUC	2021-12-04 16:44:12	welcome
12	36.5	968313147	William	06	DUC	2021-12-04 16:47:54	welcome
13	36.4	15158244167	Tom	02	Jubel Hall	2021-12-04 19:07:20	welcome
14	36.4	71132185180	Chris	02	Jubel Hall	2021-12-05 15:09:45	welcome

Fig. 10. Initial web page

index	Temp	User_ID	Name	Location_ID	Location_Name	date	access
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11	36.5	10023715146	Logan	01	Olin Library	2021-12-04 12:04:06	deny

Fig. 11. Operation button at the bottom

4.4 User interface implementation

We know that web pages written in HTML and JavaScript cannot operate on the database, so JSP Web pages are an appropriate choice. Download Tomcat and JDBC drivers to connect to the database, and they are also used to connect to the AWS database. The next step is to design the web page. First, I divide the web page into three parts as needed. They are 1) the home page used to display the information of all people who have swiped the card (Shown in Figure 10), including index number, name, ID, temperature, location, time, etc. At the bottom of this page, there are four buttons, including change ID or name, change location name or ID, search by name and search by location ID (Shown in Figure 11).

2) Delete and input the part of the database. This page contains two pages, a table with all user IDs, names and email information extracted from the database, a delete button, and an input button; The other has a table with all location names and IDs, the same two buttons (Shown in Figure 12). Press the delete button, and the data of the corresponding row will be deleted directly. Press the enter button, and you will jump to a page with an input box. Enter the corresponding information in the input box, and then click the submit button, and this information will be entered into the table of the database (Shown in Figure 13).

3) search part, two pages, one through name search and the other through location ID search, there is an input box on each page. After input, click the search button to pop up a page with corresponding information (Shown in Figure 14, Figure 15).

The web page is written in Java. Put the source code file in the corresponding folder of Tomcat, you can open the web page locally and operate the database.

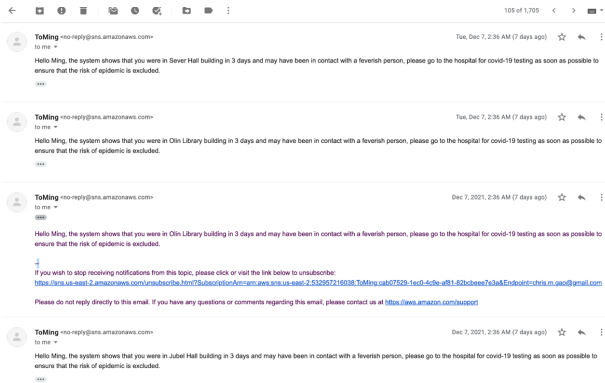


Fig. 20. Received emails

6 PROBLEMS AND SOLUTIONS

For the misjudgment mechanism of temperature measurement: If someone holds a hot drink in his hand, in order to prevent the system from judging it as a fever, we set the temperature to a certain threshold, and when we set it above 42 degrees Celsius, it is considered to be outside the normal human body temperature range and is a false temperature measurement. The system will show try again and will not active the high temperature search function.

One of the problems we encountered during the implementation of functions based on AWS was the intervals of each selection. In the early stages, we decided to select targeted people at the exact time after each record sent into database. However, such design will cause heavy burden on our system and huge latency for execution. We came up with the solution that we deploy an automated executing program, and it has proper sleeping intervals to leave enough execution time for selecting at one time.

Multiple return values for sub-queries. In the implementation of selecting targeted people, step 2 will return a list of values instead of single value, conflicting the rules of sub-query sentences, that the return value of a sub-query should has single value. To avoid such problem, we had to divide the whole 3-step query sentence into two parts: one for selecting target location and date and get a list of it, the other is a loop to process each element in the list.

When writing a web page, we encountered many problems because we were not familiar with it. For example, the title of the search box was garbled, which was solved by changing the coding method; When connecting to the database, the number of connections reaches the maximum and cannot be connected. Therefore, the function of automatic page refresh is cancelled to ensure that the number of connections will not be too large; How to extract information from multiple tables and display it on one table is solved by field simultaneous table. Other program errors caused by variable assignment are numerous, which shall be solved one by one after careful inspection.

7 LESSONS LEARNED

In this project, we further improved our use of Raspberry Pi and learned to use the various sensor modules to work together to

achieve a function. And gradually improve the functionality according to the needs and problems in the project. In the process, we also learned about the use of AWS, in many functions AWS can bring great convenience to our project.

As for the database part, this project is a great way of practicing knowledges we learnt in former class. To obtain a skill, not only should we learn knowledges in lectures, but also required practical experiences. For example, during the implementation of functions in database, we decided to use a long and complicated sub-query sentence without noticing the rules of single return value. Only through this practice did we notice such important feature existed in SQL sentences. And this experience will benefit much in our future works.

For user interface, we learned how to establish JSP Web pages, how to use JSP to connect to the database, how to delete and insert the contents in the database, and search and display the contents of the database. In addition, we also learned simple database establishment and operation methods.

8 CONCLUSIONS AND FUTURE WORK

We use raspberry pie, temperature sensor MLX90614 and NFC module rc522 to receive data and send it to AWS RDS database. We screen users with fever in cloud9, search people who have contact with patients with suspected fever in the previous three days and send emails to them. In addition, an administrator web page is established using JSP. In the web page, you can see the information of all users who have swiped the card, modified the ID and name information of users and buildings, and searched relevant information through user name or building ID. The whole product has perfect functions and can effectively help covid-19 epidemic protection.

In the future, the accuracy of the temperature sensor can be more strictly checked to see the limitations of its temperature detection. In addition, the design of JSP Web page can be optimized, the administrator login function can be added, pictures can be added, the layout can be improved to make it more beautiful and try to solve the problem that the web page can only be opened locally.