COMP4971 – Project Report

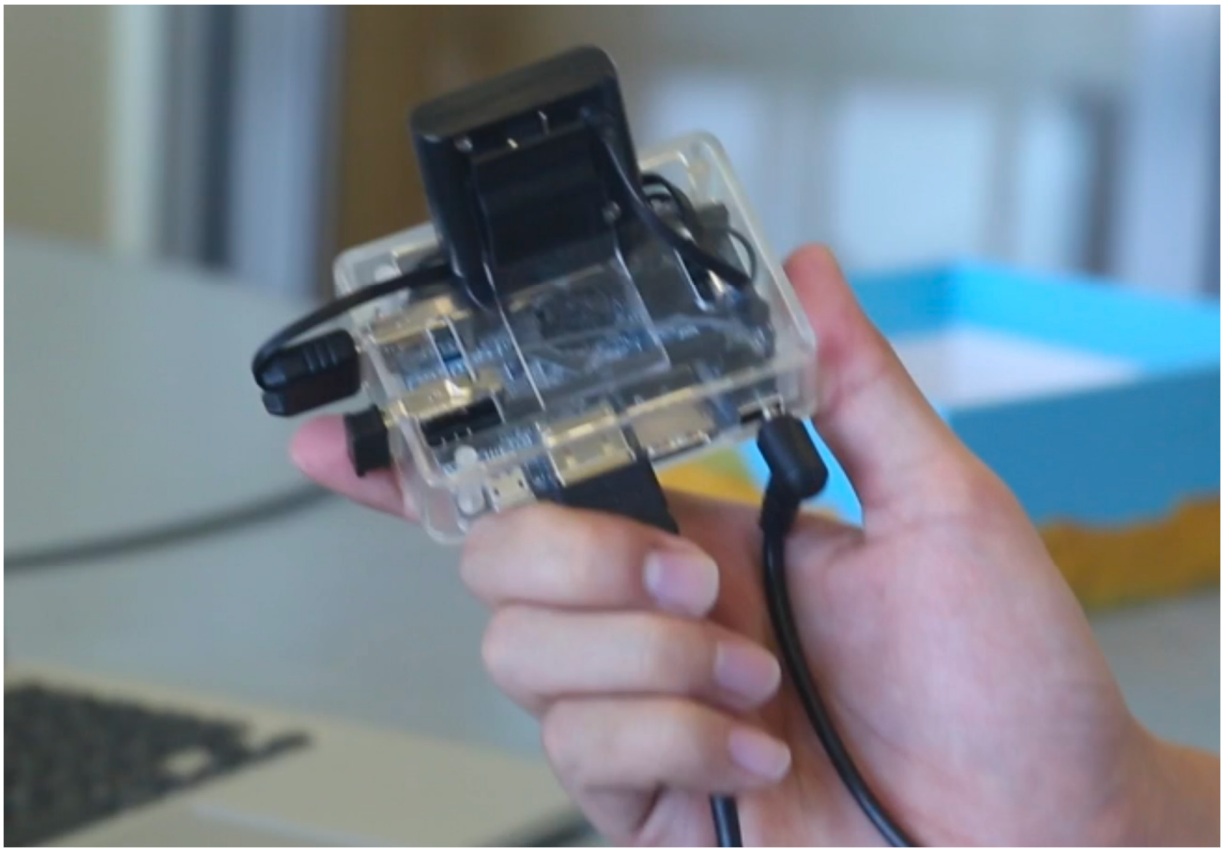
IoT on Campus Project

ZHONG, Zixuan

Collaborated with **LIU, Cheng & TANG, Kai Man**
Supervised by **Prof. David Rossiter**

1. Abstract

UST.one is an IoT Project aiming to simplify students' life in HKUST campus. We did a survey on several most common issues many students encounter on campus and decided to build a laundry room machine real-time monitoring system to assist hall residents. Students can use this system in two ways. First, before doing the laundry, students can go to our website or open our app to check whether there is any available machine. Second, by setting a reminder after starting a machine, one can be notified by the app in time once her/his clothes are ready.



mini-computer with camera

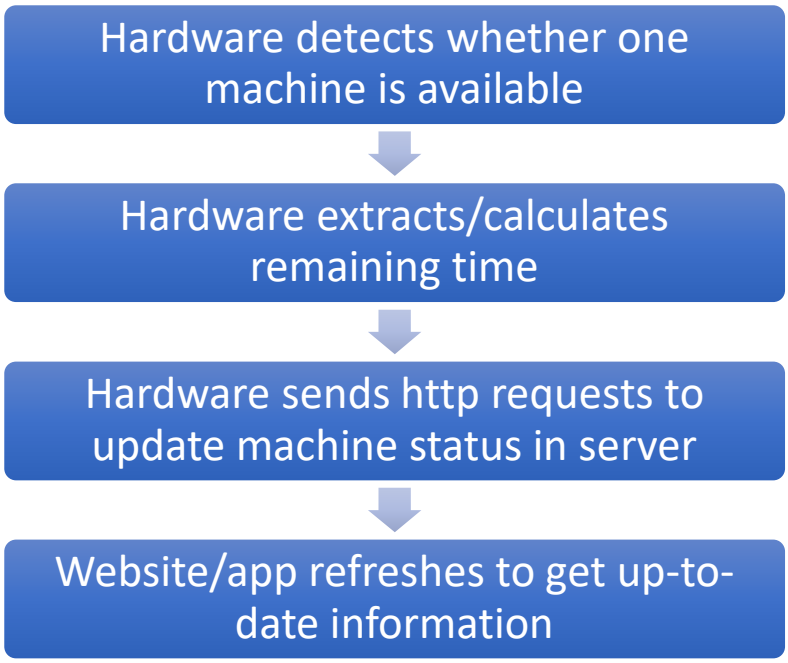
2. Introduction

2.1. Motivation

The idea of building such a system came from students' complaints at peak time when there are no available machines in the laundry room while students cannot get this information unless they go to the laundry room in person. "Internet of things", whose concept is to connect people with smart devices, could absolutely be applied to solve this problem.

2.2. Concept

The whole flow of this system is simple. And the procedure is executed periodically.



2.3. Approaches

2.3.1. Hardware

Generally, the approach determines what hardware to be used for embedded systems. In preliminary investigations, we proposed several measures.

2.3.1.1. Detecting Machine Vibration

Vibration is one of the most obvious characteristics to tell whether one machine is being used. But this approach is not very feasible. First, even when the machine is running, it does not vibrate all the time, detection device only gets instantaneous states. Second, the duration of each

run varies, which makes prediction inaccurate. Third, detection may be incorrect as machines are likely to be affected by adjacent ones.

2.3.1.2. Detecting Electricity Current

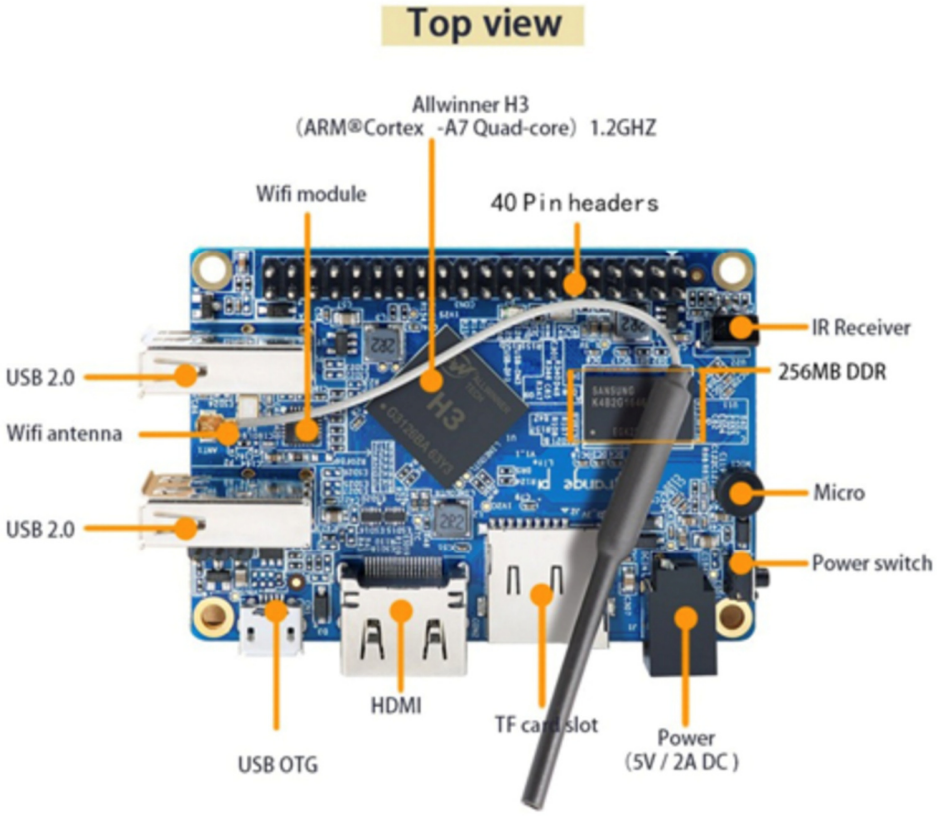
It makes sense that when running, the current of machines raises drastically. However, all machines in HKUST halls using three-phase power supply and their power interfaces are not accessible. On the one hand, there are no suitable commercial products. On the other hand, hand-made devices take high risks.

2.3.1.3. Machine API

We found City University of Hong Kong (CityU) already launched a similar app which consumes APIs of machines directly. However, most machines in HKUST are outdated. Although SHRLO is considering purchasing new washers and dryers with programmable ports, it will take years for all machines to be replaced.

2.3.1.4. Camera With OCR

After other approaches became not feasible, using camera seems to be the only feasible one. The display screen on each machine is a reliable source of machine status information. mini-computers with cameras can be attached on machines. Cameras focus on the countdown part of display screens and take pictures periodically. Mini-computers pre-process pictures, extract data and send them to the server. Orange PI Lite [1] is an open-source single-board computer suitable for this job. It has tiny size and powerful functions. It is very convenient for developers that Wi-Fi module and HDMI port are integrated into the board.



top view of Orange PI Lite
(from: http://www.orangepi.org/orangepilite/)

2.3.2. Website

The functions of the website are not complicated, so PHP is suitable. To shorten time and optimize quality, we decided to choose Laravel [2], the most popular open source PHP framework as the skeleton of our website. Many disadvantages of PHP language are eliminated in Laravel. Furthermore, it uses RESTful API, which makes Laravel can serve as the server codes of our mobile application.

2.3.3. Mobile Application

Same as the website, mobile application does not contain very complicated functions. To take the advantage of mastered knowledge of HTML, CSS, JavaScript, shorten development time, develop iOS and android together, we chose Ionic [3] Framework, a hybrid framework.

3. Development

3.1. Hardware and Programs Embedded

3.1.1. Main Programs

3.1.1.1. Libraries Used in Main Python Program

- openCV (image capturing and processing)
- numpy (matrix handling)
- json (storing configuration parameters)
- Commands (call outer library to recognize digits from processed images)
- regular expression (process output of out library)
- HTTP requests (sending request to the server)
- HTTP socket (get IP address)

3.1.1.2. SSOCR Library

SSOCR (Seven Segment Optical Character Recognition) [4] is a program run in Command-line Interface to recognize digits of a seven-segment display. An image of one row of digits is used for input and the recognized number is written to the standard output. We installed it into each mini-computer by source code so that python programs can use it by command line library.

Algorithm:

sample images of SSOCR

In this image, the left border of a digit is represented by a red column, the right border as a blue column. Horizontal green lines of digit width show connected vertical digit parts. The grey rectangles represent the digit dimensions.

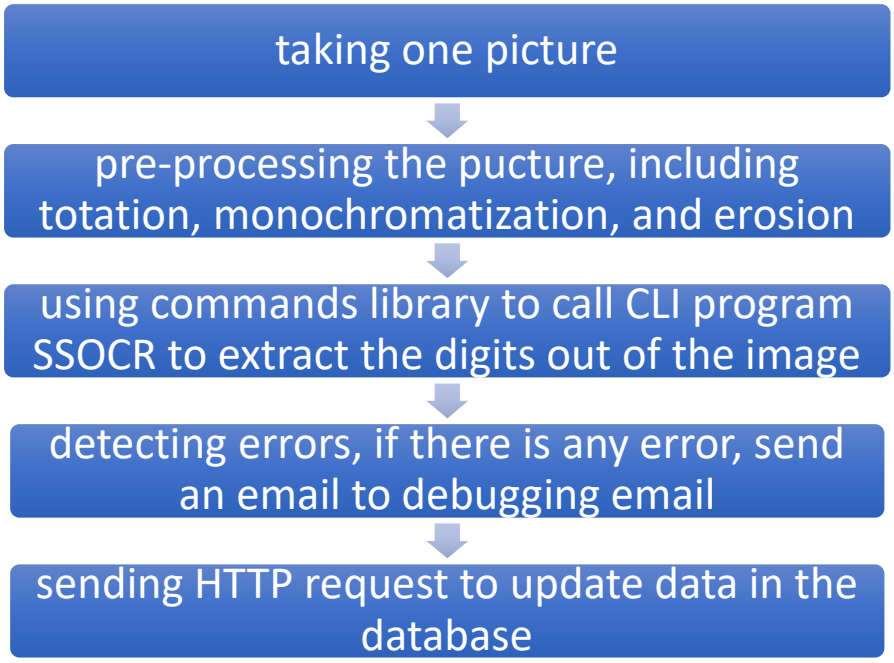
Pixels found by the vertical scan line are shown in red, green and blue for the top, middle and bottom third. Those found by the horizontal scan lines are shown in red and green for the left and right half of the digit. No scanlines are used to recognize a one.

3.1.1.3. Some helper functions in main.py:

- internet_on(): check whether the internet connection is right
- local_log(text: string): record all ssocr output into tolog.txt
- http_log(text: string): record all http request response into http_log.txt
- error_log(text: string): record all errors into error_log.txt
- get_ip(): return the current IP address of this mini-computer
- rotate_bound(image: opencv_image, angle: interger): use opencv to rotate the picture by any angle and fill the margin to make the picture always rectangle.
- Process_image(image: opencv_image): pre-process the picture and store two pictures, cropped.jpg and output.jpg

3.1.1.4. Infinite loop

The main part of the main.py program is an infinite loop whose cycle time is 15 seconds.



3.1.1.5. Send_email.py Library

Besides the main.py, there is a send_email.py file serving as a library for sending debugging emails. It will send an email containing the original images and ssocr output to our debugging email address.

3.1.1.6. conf.json

The configuration file, conf.json, has several important parameters for identifying the device and pre-processing images.

Here is the content of an original JSON (JavaScript Object Notation) [5] file.

{"id": 0, "token": "undefined", "ssocr_rotate": 0, "erode": 3, "number": "undefined", "crop": [0, 0, 0, 0, 0, 480], "hall": "undefined", "illumination": 240, "type": "undefined"}

when installing new devices, we only need to set ‘id’ and ‘token’. Identity information will be set by the server automatically and these parameters can be adjusted on the user-friendly admin web page.

3.1.2. SSH

By utilizing SSH, the procedure of installation and maintenance of devices is simplified markedly. For installing new devices, we only need to install openssh server [6] by one command, the following installation of other libraries can be done remotely. For maintenance, developers only need to ssh it to check log files and restart the main program.

3.1.3. Service

Because for the microcomputer, main.py may be the main job, we created a custom service in Linux system [7] so that the main program runs all the time.

Create a file in path: /lib/systemd/system/ named ustone.service

Add codes in:

```
[Unit]
Description=UST.one Service
After=multi-user.target
[Service]
Type=idle
ExecStart=/usr/bin/python /home/ustone/main.py > /home/ustone/ustone.log 2>&1
[Install]
WantedBy=multi-user.target
```

Make it executable:

```
$ sudo chmod 644 /lib/systemd/system/ustone.service
```

Reload system services:

```
$ sudo systemctl daemon-reload
```

Enable the custom service:

$ sudo systemctl enable ustone.service

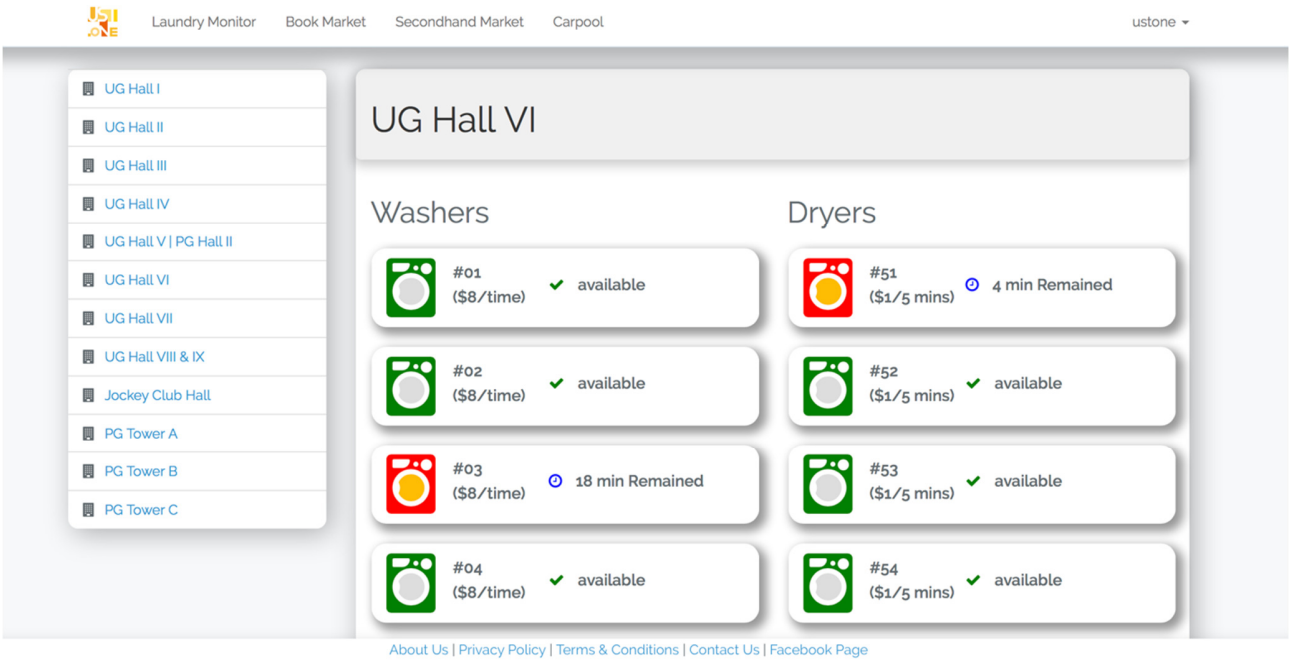
3.1.4. Cron [8]

main.py program may be interrupted accidentally, to make it more stable, we created a cron job to restart the ustone service every 30 minutes. Add this line at the end of the cron file:

0,30 * * * * /bin/systemctl restart ustone.service

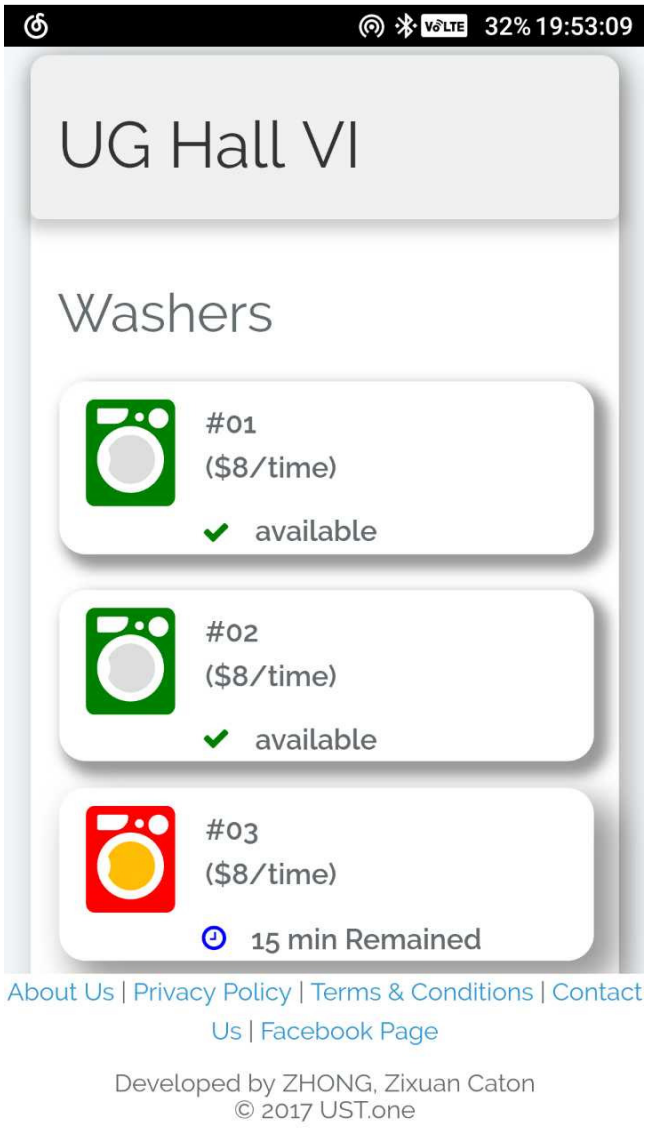
3.2. Website

3.2.1. User Interface



screenshot of website

user can easily check information of all machines in each laundry room.



mobile view of website

even users visit the website with mobile devices, the view is nice.

3.2.2. Basic Logic

The website handles all requests sent by all devices, calculates the ending time and updates corresponding records in the database. The front-end page fetches data from the database, calculates the remaining time, and display them. Front-end pages refresh automatically every 20 seconds.

3.2.3. Admin Page

screenshot of admin page

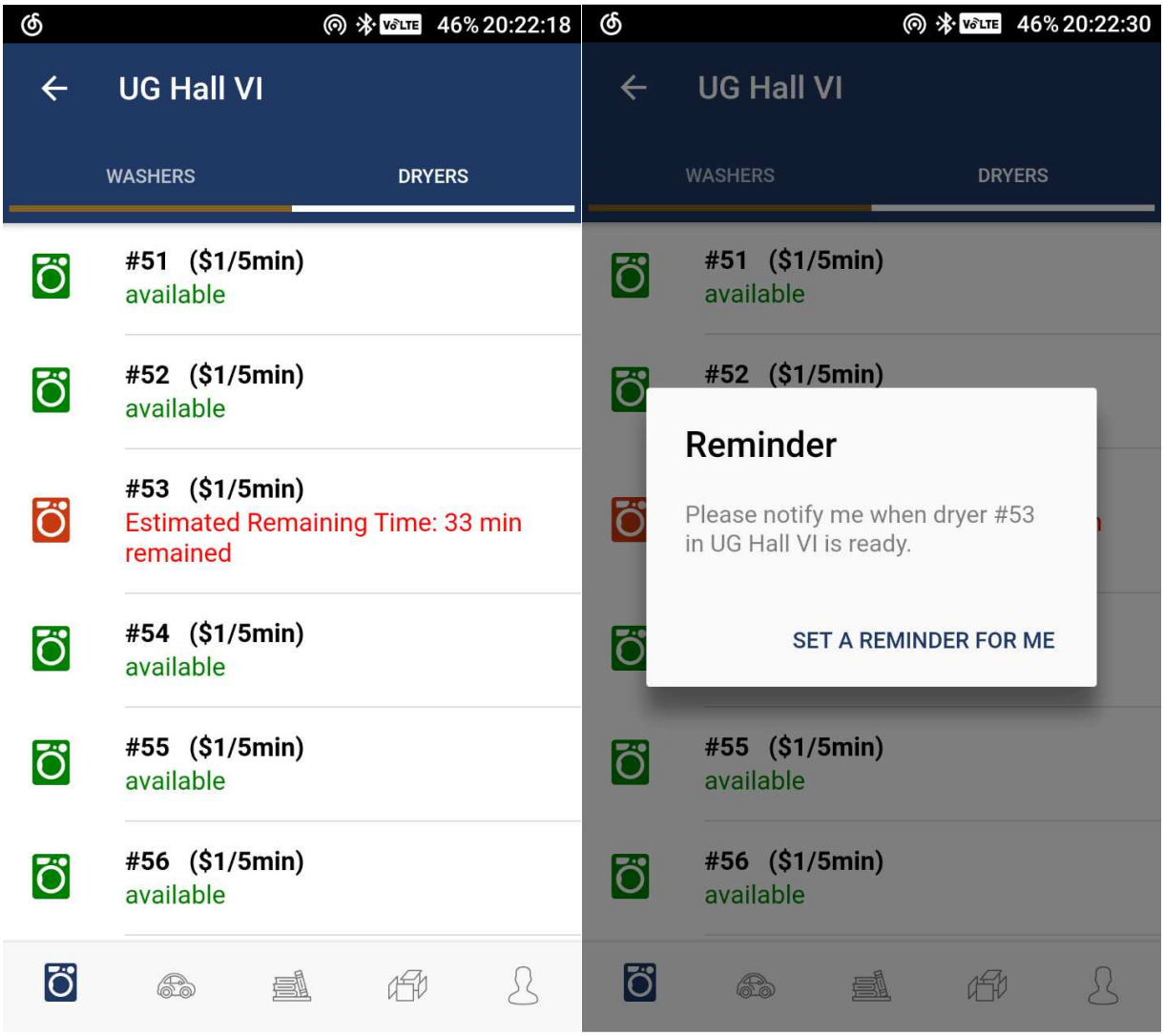
For the convenience of installation, debugging, and maintenance, we built an admin page, like above.

Only admin users can access this page. a, b, c, d, E, F are 6 parameters to determine the red part, mini-computer crops images according to these parameters and transfer the red part to a rectangle, like the picture in the middle. Then, it extracts only bright enough part of the image and converts it into the gray image, rotates it by a specified angle, and erodes the image specified times. SSOCR can easily extract the correct digits from the right-most image.

Besides, admin can change the price of each machine and set whether it is in service easily.

3.3. Mobile Applications

3.3.1. User Interface



mobile application view

Most functions are the same as those of the website, but the app has one more function, reminder.

4. Implementation

At this point, we have implemented 6 devices in UG Hall VI. The testing results are satisfying. We plan to implement all devices hall by hall and teach hall officers how to adjust those parameters so that all devices could always be in the best condition.

5. Promotion

5.1. Sticker on Devices

As the system consists many microcomputers which will be mounted on the laundry machines and the dryers, well-designed stickers will be post on the each of the microcomputers to explain the purpose of its installation. Thus, to promote the system to the target users(residents) who must use laundry machines/dryers.

5.2. Posters Endorsed by SHRLO

With the aid and support from the SHRLO, large posters will be posted in the all the residential halls to advertise this system and encourage residents to use the system.

5.3. Online Video Promotion

A promotional video is filmed by our team which will be posted on our website to better promote our system in a more vibrant way. The video will explain the inconveniences of checking the availability of the laundry machine or the dryer manually and how our system can rectify that and simplify user's life.

6. Conclusion

Throughout the project, we had encountered many difficulties and dilemmas. Such as choosing the mechanism to check the availability of the machine, optimizing cost, stability and feasibility, learning new algorithm and programming language, co-operating with the related department for the experiment, and applying our textbook knowledge to the real-life problem. From this project, we have learnt how to tackle a problem in different approaches and how to decide the best approach that is the most feasible and suitable under given circumstance. After many trade-offs, brain-storms, experiments and decisions, we finally chose to use camera and OCR. With technical problems are being settled, we also should study the demand from our target users and promote it to the UST's community. By conducting this project, we better understand the engineering ethics and spirits. In a real-life situation, engineering is much more than solving the problem. It involves co-operating with different parties and how to optimize the solution under restrictions. We also learn that computer science is not only just writing code, but also keep trying new things and have an open and innovative mind. The best things we take away from this project are that we better understand what is engineering and become more open-minded. We are also much more resilient to difficulties after conducting this project.

7. Contribution Table

	ZHONG, Zixuan	LIU, Cheng	TANG, Kai Man
Leadership	L	C	
Website backend (logic & data)	L		C
Website front-end (design & view)		L	C
Mobile application	L	C	
Hardware operation system (Linux service, cron & ssh)		C	L
Hardware program (main.py & conf.json)	C		L
Debugging & testing	L	C	C
Contact with relative department officers and product vendors		L	C

L: leads, C: contributes

8. References

[1] Z, Yifan. "What's Orange Pi Lite?", orangepi.org, 2017. [Online]. Available: http://www.orangepi.org/orangepilite/. [Accessed: 12- Aug- 2017].

[2] T. Otwell, "Laravel - The PHP Framework For Web Artisans", *Laravel.com*, 2017. [Online]. Available: https://laravel.com. [Accessed: 12- Aug- 2017].

[3] "Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular", *Ionic Framework*, 2017. [Online]. Available: https://ionicframework.com. [Accessed: 12- Aug- 2017].

[4] "Seven Segment Optical Character Recognition", *Unix-ag.uni-kl.de*, 2017. [Online]. Available: https://www.unix-ag.uni-kl.de/~auerswal/ssocr/. [Accessed: 12- Aug- 2017].

[5] "JSON", *Json.org*, 2017. [Online]. Available: http://www.json.org. [Accessed: 12- Aug- 2017].

[6] "openSSH", *Openssh.com*, 2017. [Online]. Available: http://www.openssh.com. [Accessed: 12- Aug- 2017].

[7] "What is Linux?", *Linux.com | The source for Linux information*, 2017. [Online]. Available: https://www.linux.com/what-is-linux. [Accessed: 12- Aug- 2017].

[8] "Newbie: Intro to cron", *Unixgeeks.org*, 2017. [Online]. Available: http://www.unixgeeks.org/security/newbie/unix/cron-1.html. [Accessed: 12- Aug- 2017].