A Better Laundry System for HKU Dorms

Interim Report

Atiab Bin Zakaria (3035718230)

Ho Joshua (3035689831)

Supervised by Dr. Jia Pan

22 January 2023
Abstract

A most prominent issue faced by the dormitory residents at the University of Hong Kong (HKU) is the uncertainty mixed into the laundry process since they do not know about the availability of the machines before physically going to the laundry room. This report presents a project to modernize the process by introducing an application for users to be able to check exactly how soon they can truly begin their laundry. The method proposed is to use a camera and code to detect the time-remaining information displayed on the machine’s screen since the ageing machines at the dorms do not have ways of accessing this information programmatically. At this point in the project, the acquisition and preliminary testing of the hardware-software architecture have been completed, and the immediate next steps include the implementation work of the proposed method.
Acknowledgements

We would like to thank Dr. Jia Pan for his guidance as our supervisor, the Department of Computer Science for facilitating the hardware acquisition, and Miss Mable Choi for her instructions in the written components.
# Table of Contents

Abstract

Acknowledgements

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

1 Introduction

  1.1 Background: Outdated laundry systems at HKU
  1.2 Current solutions
  1.3 Motivation and contribution
  1.4 Objectives and approach
  1.5 Outline

2 Methodology

  2.1 Overview
  2.2 Backend
  2.3 Frontend
  2.4 Tuya IoT Platform
  2.5 System Workflow
    2.5.1 Adding New Devices
    2.5.2 Machine Reservation
    2.5.3 User Identification
  2.6 Summary

3 Project Status

...
3.1 Tuya Cloud Project Setup 10
   3.1.1 Asset Structure 10
   3.1.2 Device Registration 11
   3.1.3 Tuya Cloud API 12
3.2 Pi Camera and SSOCR Performance 12
3.3 Web App Wireframes 14
3.4 Outline and Progress 18
3.5 Learnings and next steps 19

4 Conclusion 21

References 22
List of Figures

Figure 1. Smart Industry App Workflow 5
Figure 2. Device Registration Workflow 6
Figure 3. Machine Reservation Workflow 7
Figure 4. User Identification Workflow 8
Figure 5. Overall System Workflow 9
Figure 6. Asset Structure of the Project 10
Figure 7. Authorized Personnel of a Dorm 11
Figure 8. List of Registered Devices of a Dorm 11
Figure 9. API response of Tuya API a 12
Figure 10. Python Script to Control the Camera 12
Figure 11. Example Image of the Pi Camera 13
Figure 12. SSOCR Responses 14
Figure 13. Login Page 14
Figure 14. Home Page 15
Figure 15. Date Selection Page 15
Figure 16. Time Selection Page 16
Figure 17. Reservation Page with QR code 17
Figure 18. Reservation Page with Time Left on a Machine 17
List of Tables

Table 1. Project timeline 18
## Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>HKU</td>
<td>The University of Hong Kong</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>OCR</td>
<td>Optical Character Recognition</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>QR</td>
<td>Quick Response</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background: Outdated laundry systems at HKU

With the rise of the adoption of internet-connected home appliance devices such as smart laundry machines and dryers in modern times, the features expected by the users of such devices have changed from simple utility to being able to conveniently access comprehensive operational information about the devices such as the status (in use or not) and time remaining for becoming available on personal devices such as smartphones and laptops. Laundry systems in the dormitories provided by The University of Hong Kong (HKU) to house its students have not kept up with the advances in technologies in this aspect of residents' living experience at the facility. In comparison to the features provided by modern machines, the laundry workflow at these residences is rather outdated. Users are always having to guess with low certainty about the availability of machines since they do not have any way of knowing the states in advance of actually going to the laundry room (likely with a heavy load of clothes). Furthermore, they are required to ensure having enough balance in a proprietary dorm-provided debit card to pay for their machine use. The only method of topping up this card is using cash at a kiosk located in a common area separated by numerous floors from the laundry room. Once these initial hurdles are cleared by a user and they have successfully started laundering their laundry items, they must ensure to note the time remaining till completion reported on the machine's display to be able to collect the load promptly. This is important to avoid unwanted handling of the clothes by others during busy times.

1.2 Current solutions

Indeed, a simple way of solving many of the issues mentioned above is to replace the currently deployed machines with newer internet-connected ones. However, it would be a significantly expensive endeavour and at the same time, would raise questions about the fate of the machines to be replaced since discarding working devices is quite irresponsible from an environmental sustainability standpoint. As noted by Shakya et al. (2021), the research in the space of modernizing existing laundry facilities is sparse. The authors of that paper highlight another group of researchers who were able to achieve the detection
of whether a machine is in use or not by monitoring the power drawn by the machine in its different operational states. However, the approach taken by Shakya et al. (2021), which is based on monitoring the vibrations detectable on the machine's body, avoids having to modify the electrical power delivery mechanism to the machines which might be desirable in the context of a power source connected to numerous machines such as in a dorm environment.

1.3 Motivation and contribution

While both of these approaches, i.e., based on power monitoring or vibration monitoring, can detect the state of the machines, neither can report how long the machine will remain in the in-use state. We believe implementing a method for users to be able to know when they can find a machine available will be a welcome quality-of-life improvement in their already busy lives as HKU students.

1.4 Objectives and approach

As described in Section 1.2, it would be convenient to be able to implement modernization projects without altering the existing infrastructure such as the electrical power source. Hence, we intend to pursue an approach that augments the machinery that is already deployed in the student residences. To summarize, our main objectives are to build an app that allows dorm residents to reserve and check the status of their machines and an admin page for administrative tasks e.g., price setting.

1.5 Outline

This report is organized into four chapters: the first of which provides the necessary context for our motivation to propose this project as well as what we hope to achieve from carrying it out. In the second chapter, we define how we plan to accomplish the goals listed at the outset, including the hardware-software architecture and backend-frontend designs. The third chapter lays out the tentative timeline for the project, reports the current progress, and the planned immediate next steps. Finally, in the fourth chapter, the report ends with a reiteration of our motivation and our methods of moving forward with the project.
2 Methodology

The following sections in this chapter describe details of the approach we are taking with this project. Starting with a high-level overview to facilitate a general understanding of the methods, we then proceed to provide more information about the specifics of the backend and frontend architectures.

2.1 Overview

The guiding principle, as discussed in Section 1.4, for deciding on an approach for us was to not disrupt the existing equipment. As such, we chose to try implementing our solution based on augmentation: using external hardware to programmatically extract the time-remaining information displayed on the machine’s screen. Once access to such up-to-date information is established, we can move to focus our work on presenting it to the users in an easy-to-consume medium which, in our case, is a web application. The basic strategy of recognizing the time remaining reported by the machine is to point a camera module connected to a computing unit that can process the captured images by applying optical character recognition (OCR) and extracting the information which it can then relay to the user-facing frontend application.

In terms of hardware, the primary component is a Raspberry Pi single-board computer with a camera module connected to it. The Raspberry Pi is intended to function as the “gateway” to send all information related to the machine to the backend server and assist in controlling the machines via smart plugs.

2.2 Backend

We have identified the FastAPI web framework for implementing the application programming interface (API) functionality on the backend for the frontend application to consume the extracted data. The rationale for choosing FastAPI is that it is a modern, fast (high-performance) web framework for building APIs with the Python language with a focus on standard Python type hints which aids in avoiding a common class of bugs in software development associated with incompatible data types. The FastAPI code is meant to additionally include the logic to control the camera module by which it captures an image of the machine’s screen.
Using an OCR engine, the textual data is extracted and made available for access by other applications (the frontend application in our case) via an HTTP endpoint of the backend API. Initially, we planned on using the Tesseract OCR engine (whose development is led by researchers at Google). However, in our testing, we have found its performance to be unsatisfactory in terms of execution time on the Raspberry Pi hardware as well as significant inaccuracy for the seven-segment font used in the machine displays. Hence, an alternative named SSOCR which is specifically developed for seven-segment fonts has been identified and we are currently exploring its capabilities.

2.3 Frontend

The tooling of choice for designing and architecting the frontend application is the Next.js framework. The main draw for Next.js is that it is based on the powerful React library for web frontend applications while also providing convenient features such as file-based routing and server-side rendering which allows for building web applications with maintainable architectures.

2.4 Tuya IoT Platform

Wi-Fi smart plugs are used to control the usability of machines. They will be switched on according to whether the machines are being reserved or not. Tuya will be used to control the smart plugs via APIs.

Tuya is a popular IoT platform that offers cloud services for developers to build their own IoT projects. As of Sep. 30, 2022, Tuya has accumulated over 647,000 registered developers from over 200 countries and regions. Its high technology standard and reputation make it a suitable and reliable platform to use.

Tuya offers an asset-user system design for developers which allows customized asset structures that meet various business requirements. An asset is essentially a group of IoT devices. For example, an asset can be all IoT devices that are sensors or all IoT devices located in a certain room. This is beneficial as it allows the smart plugs to be differentiated according to which dorm, they are located in. Another benefit is that it allows only certain personnel to control the assets e.g., adding new smart plugs. In short, the asset-user system provided by Tuya allows for greater security and organization.
2.5 System Workflow

This section gives further details on the workflow of various procedures. There are two main groups of people in the system, users and administrators. Users are the residents who reside in dorms and administrators who are responsible for the initial setup.

2.5.1 Adding New Devices

The initial setup of the system is to register the smart plugs into the cloud. Tuya provides a separate app called Smart Industry dedicated for registering IoT devices into the cloud (see Fig. 1).

Only authorized personnel of the project are allowed to login into the app and control or add devices to their respective assets. The following steps are taken (see Fig. 2).
The Pi stores the smart plug ID as it needs to provide it as a parameter in future API requests. A single database table will hold all machine and reservation information. The columns include

- Name of the machine
- ID of the machine
- ID of the resident who reserved the machine
- Status on whether the user has started using the machine
- Time left indicated by the washer
- Date and time at which the machine becomes available again

Initially, only the name and ID of the machine are written. The other columns will be later written along the workflow. After the machine is available in the
app, the Pi will look for updates on whether any users have reserved the machine from the FastAPI server.

2.5.2 Machine Reservation

Once the machine is shown in the app. Users can then reserve the machine. The following diagram briefly explains the workflow of the initial reservation (see Fig. 3).

![Machine Reservation Workflow Diagram]

*Figure 3 Machine Reservation Workflow*

2.5.3 User Identification

Upon reservation, users can use the machine upon identification during their specified time slots. The following diagram briefly explains the workflow of this process (See Fig. 4).
2.6 Summary

Putting it all together, the overall architecture of the proposed system is illustrated in Fig. 5.
Figure 5 Overall System Workflow
3 Project Status

This chapter lays out our estimated sequence of tasks for the project over the year as well as what we plan to focus our efforts on immediately next.

3.1 Tuya Cloud Project Setup

The following section is dedicated to showing the progress made in setting up the asset organization of the cloud project and device registration.

3.1.1 Asset Structure

As the project is dedicated to HKU dorms. There will be an asset corresponding to each dorm (see Fig. 6).

Figure 6 Asset Structure of the Project

The assets can be seen as a tree structure with HKU as the root and three of the dorms as the child nodes. Only three assets are created for demonstration purposes.
Authorized personnel are added to specific assets to control the IoT devices in that asset (See Fig. 7). First, the phone or email of the personnel should be used to add them to the cloud project then he/she can be later added to a specific asset. Authorized personnel of an asset also have permission to child assets as well. In practice, these personnel are the developers or maintainers of the system. This structure will help organise our app and only show relevant machines to our users.

3.1.2 Device Registration

The smart plugs are listed in their respective asset with the device ID shown in the cloud project after registration (See Fig. 8).
3.1.3 Tuya Cloud API

Tuya provides APIs for their cloud services, and this will be used for our web and mobile app (see Fig. 9). For example, APIs to retrieve all devices located in a particular asset or device status can be retrieved.

![Figure 9 API response of Tuya API](image)

3.2 Pi Camera and SSOCR Performance

Tests have been carried out to assess the performance of the Pi Camera and SSOCR software.

![Figure 10 Python Script to Control the Camera](image)
A short python script with the help of the picamera library has been created to capture example digits from the Pi Camera (see Fig. 10). As closed-ranged images are taken rather than long ones, the lens of the camera was adjusted for better resolution. PNG is used as the format of the image as other formats like JPEG compress data thus the quality is lowered. However, the quality seems to still be lacking thus follow-up steps are to be taken (see Fig. 11).

Next, the SSOCR software was tested on better-quality images. Several constraints were found. Firstly, SSOCR requires images to be monochrome i.e., black and white thus image pre-processing must be done and changing the colours is one of them. Several libraries can help solve this issue including Pillow which is a Python Imaging Library and NumPy. Another issue found was that the number of digits had to be explicitly stated. The reason is unknown and will be further investigated. Other than the two problems stated, the software seems to extract the correct number which is 1.00 (see Fig. 12).
3.3 Web App Wireframes

This section shows the main wireframes of our app. The login page will allow residents to log in using their UID and PIN (see Fig. 13). There is another login option for the admin. That is where they can set prices for the machines and do other tasks.

Figure 13 Login Page
Upon login, users will be greeted with the home page showing all machines available for use. Only the machines in their dorms are shown (see Fig. 14). If the user does not live in any dorms, nothing will be shown.

**Figure 14 Home Page**

Upon clicking a machine, users can select a date and several 1-hour sessions (see Fig. 15 and 16). Sessions that are booked will be coloured red for indication.

**Figure 15 Date Selection Page**
Upon reservation, users can go to the reservation section to see their status. Initially, a QR code will be shown which can be used to start the machine (see Fig. 17 and 18). After scanning the QR code, the time left on the washer will be shown instead.
Figure 17 Reservation Page with QR code

Figure 18 Reservation Page with Time Left on a Machine
### 3.4 Outline and Progress

Table 1 below outlines the tentative timeline for the project. The Status column logs the progress we have achieved so far in the corresponding tasks.

*Table 1 Project timeline*

<table>
<thead>
<tr>
<th>Date</th>
<th>Tasks</th>
<th>Status</th>
</tr>
</thead>
</table>
| Early September 2022  | • Search and order appropriate hardware  
• Set up GitHub organization                                           | • Raspberry Pi 3 and camera module have been acquired  
• GitHub organization has been created at https://github.com/fyp-bzho |
| 2 October 2022        | • Deadline for deliverables of Phase 1 (Inception)  
  o Detailed project plan  
  o Project web page       | • Project plan submitted  
• Project web page is live and accessible at https://wp.cs.hku.hk/2022/fyp22023/       |
| Mid October 2022      | • Evaluate the functionalities and suitability of the acquired hardware for the project  
• Set up basic technological architecture, e.g., connect devices, install OS, register IoT developer account for API access | • Successfully powered up the acquired hardware  
• Basic architecture is in place and has been preliminarily tested |
<p>| Mid November 2022     | • Prepare a prototype of the UI of the web app                      | • Complete                                                                                       |</p>
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late November 2022</td>
<td>• Detail the reservation system's functionalities</td>
<td>• In progress</td>
</tr>
<tr>
<td>Early January 2023</td>
<td>• Start frontend development</td>
<td>• In progress</td>
</tr>
<tr>
<td>22 January 2023</td>
<td>• Deadline for deliverables of Phase 2 (Elaboration)</td>
<td>• In progress</td>
</tr>
<tr>
<td></td>
<td>• Preliminary implementation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Detailed interim report</td>
<td></td>
</tr>
<tr>
<td>1 February 2023</td>
<td>• First presentation</td>
<td>Pending</td>
</tr>
<tr>
<td>18 April 2023</td>
<td>• Deadline for deliverables of Phase 3 (Construction)</td>
<td>Pending</td>
</tr>
<tr>
<td></td>
<td>• Finalized tested implementation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Final report</td>
<td></td>
</tr>
<tr>
<td>17-21 April 2023</td>
<td>• Final presentation</td>
<td>Pending</td>
</tr>
<tr>
<td>3 May 2023</td>
<td>• Project exhibition</td>
<td>Pending</td>
</tr>
</tbody>
</table>

### 3.5 Learnings and next steps

From our progress to this point, we have learnt to expect unexpected issues to crop up along the way. For instance, a salient issue we faced initially was acquiring the Raspberry Pi and camera module hardware (apparently due to the ongoing silicon chip shortage worldwide). Thankfully, we were able to source the appropriate hardware with the help of the Department of Computer Science.

As listed in Table 1, we would like to turn our immediate attention to start implementing the frontend user interface (UI) as well as discussing and elaborating on the details of the APIs that might be used for our backend server. Progress on the OCR was made but a bit more has to be done such as changing
the images to monochrome, quality handling etc. After all is done, we plan to test out the app in real time.
4 Conclusion

Anecdotal evidence of one of the most common issues that cause dissatisfaction among residents of HKU dormitories is the friction involved in the process of completing their regular laundry. We have identified an aspect of this problem that might significantly improve the whole process: Providing a degree of certainty about the availability of a machine. This report has described how we are planning to attempt this modernization effort via a combination of hardware and software.

Now that the majority of the preparatory work in our plan has been completed, the near future of the project will involve the bulk of the implementation work. In that venture, we are expecting to expend considerable effort on the software development, testing, and debugging aspects on both the frontend and backend.
References