A Better Laundry System for HKU Dorms
Detailed Project Plan

Atiab Bin Zakaria
Ho Joshua
Supervised by Dr. Jia Pan

1 Project Background

Laundry is an inevitable task which people perform routinely. It is estimated that an average person spends about 4 hours per week on laundry including preparing, washing and drying, and putting away the clothes.

The typical process of doing laundry in a dormitory of The University of Hong Kong (HKU) usually involves the following. Firstly, students need to add value to their physical smart card provided by the administration using cash at a designated kiosk at the residence. They then need to move to the laundry room and try and find a machine that is available to use.

The washers deployed at the dorms are still traditional, i.e., there are no “smart” features available that are common in modern machines. For example, remote control using a smartphone app including start and stop capabilities. These smart washers have built-in hardware that enables such communication with applications over Wi-Fi. Other universities around the world have already developed feature-rich, digital laundry management systems. For instance, The Hang Seng University of Hong Kong has provisioned a dedicated app for dormitory residents to use electronic payments, check availability of the washers, prices and other information through their smartphones. However, HKU dormitories still lag behind in this regard with their largely manual laundry process.

In a dormitory environment which involves sharing laundry machinery, a number of inconveniences can present themselves to the residents. Pertinent of such issues is the uncertainty about finding a machine that is not in use. Often, residents find themselves moving a heavy laundry load all the way to the laundry room only to find that they cannot use any of the machines. Another common scenario is that some may forget to retrieve their clothes in a timely manner, which may force other residents to have to take the clothes out to use the machine. This is not ideal for either party since the clothes may need special handling care, while it is wasteful of the person next in line’s time.

HKU students are often busy with schoolwork in general and as such, they tend to value how they balance their time between academics and social commitments. Any savings gained from mundane yet necessary tasks such as laundry are likely to be a welcome development in the schedules of HKU dorm residents.
Hence, an improved laundry system for them would allow them to use their time better on more important matters in life.

2 Project Objective

The main objective of this project is to create a web app for a streamlined and fair laundry process for HKU dormitory residents. The app aims to allow residents to:

1. Check the availabilities of washers
2. Reserve an available washer
3. Monitor the time left for the washers in use
4. Be notified when a washer becomes available

It is to be noted that the project does not plan to modify any of the existing hardware deployed at the residence.

3 Project Methodology

3.1 Hardware

![Diagram of hardware components]

Figure 1: Interaction between different hardware components

The project plans to use an Internet of Things (IoT) smart plug device and the single-board computer Raspberry Pi integrated with a camera module at the core of its hardware implementation (see Figure 1). Essentially, the Raspberry
Pi will host the code to control camera behavior and send appropriate API calls to the smart plug to switch it on when the washer is reserved and switch it off after the session is over. The smart plug connected between the washer’s plug and the outlet is also meant to access the voltage usage of the machine. The camera will be used to identify users to make sure the user is living in this dorm and has booked the washer using their university card. This design is meant for each washer.

3.2 Washer Usage Tracking

Two potential options have been identified for accessing data related to the usage status of the machines: either using the camera to keep track of the time left indicated on the machine’s screen with the help of open source optical character recognition algorithms such as Google’s Tesseract or interpreting the voltage usage of the machine to understand its status. It is to be noted the latter method is low resolution in the sense that is only able to provide on-or-off information about the machine.

3.3 Web App

The web app is designed to be the frontend for presenting all the information to the residents as well as the interface for user actions such as the reservation system. The proposed technology of choice is the modern JavaScript web app framework Next.js (based on the React library).

3.4 Reservation System

![Diagram of system behavior upon successful identification]

Figure 2: Behavior of the system upon successful identification
The availabilities of the machines are presented in the web app and once a resident reserves a machine, the user interface is updated to mark the machine reserved. The reservation should only last for a short duration, e.g., 10 minutes and the person needs to identify themselves at the camera to avail the reservation.

The proposed way to identify users consists of two options, either using the student’s card as identification with the help of OCR or using QR codes generated within the web app. Either way the camera will be used in the verification process and upon successful identification, i.e., the student lives in the dorm and has booked the washer, an API request will be sent to allow usage of the washer with the help of the smart plug (see Figure 3).

3.5 API

There are multiple API providers that allow retrieval and addition of information from/to their smart devices. For example, the company Smart Things which is a subsidiary of Samsung, develops its own smart devices and cloud platforms for consumers to use (see Figure 2). Users of the platform can then access their connected devices through an API to remotely control it. Other companies that provide similar services include Google, Tuya Smart etc.

4 Project Schedule and Milestones

Aside from the formal deliverables such as the interim and final reports required by the department, two major categories of deliverables have been identified: hardware and software.

In terms of hardware, the necessary and fully functioning hardware will be gathered, namely, a Raspberry Pi model with required specifications e.g., Wi-Fi, SD card readers to install the OS, USB ports, power supply, a smart plug that is accessible through APIs and able to record voltage and a camera that can be programmed via Raspberry Pi. The functionalities of these components
must be checked as well. e.g., checking whether the voltage can be accurately measured and whether there is any significant latency. If any problems are found, alternatives will need to be identified.

For the software, the backend logic for the monitoring and reservation components of the system will have to be implemented (likely using the Python language and its ecosystem of packages). Additionally, the frontend web app that consumes the backend will have to be developed as explained in section 3.3.

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early September 2022</td>
<td>Search and order appropriate hardware; Set up GitHub organization</td>
<td>Raspberry Pi 3 and camera module have been acquired; GitHub organization has been created</td>
</tr>
<tr>
<td>2 October 2022</td>
<td>Deadline for deliverables of Phase 1 (Inception); Detailed project plan; Project web page</td>
<td>In progress</td>
</tr>
<tr>
<td>Mid October 2022</td>
<td>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</td>
<td>Pending</td>
</tr>
<tr>
<td>Mid November 2022</td>
<td>Prepare a prototype of the UI of the web app; Detail the reservation system’s functionalities</td>
<td>Pending</td>
</tr>
<tr>
<td>Late November 2022</td>
<td>Start frontend development</td>
<td>Pending</td>
</tr>
<tr>
<td>Early January 2023</td>
<td>Start backend development</td>
<td>Pending</td>
</tr>
<tr>
<td>9-13 January 2023</td>
<td>First presentation</td>
<td>Pending</td>
</tr>
<tr>
<td>22 January 2023</td>
<td>Deadline for deliverables of Phase 2 (Elaboration); Preliminary implementation; Detailed interim report</td>
<td>Pending</td>
</tr>
<tr>
<td>18 April 2023</td>
<td>Deadline for deliverables of Phase 3 (Construction); Finalized tested implementation; Final report</td>
<td>Pending</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>