Evolution in Electrical and Electronic Engineering Vol. 4 No. 2 (2023) 369-378

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Autonomous Self Check-In KeyBox and Reservation for Homestay Operation

Wan Muhammad Ezzadam [1], Nor'aisah Sudin [1]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.02.046
Received 17 January 2023; Accepted 08 September 2023; Available online 30 October 2023

Abstract: Homestay check-in procedure usually requires physical interaction between owner and guests for registration along with key issuance to the guest. Sometimes, homestay owners are not available and not able to physically meet-up with guests to issue keys. Moreover, with the recent Covid-19 outbreak, people are encouraged to minimize physical interaction to subdue the outbreak of Covid-19. To automate reservation and check-in process of homestay operation while also minimize physical interaction between owner and guest, an Autonomous Self Check-In KeyBox and Reservation for Homestay Operation is developed. There are three main components in the system which are reservation form, admin webpage and KeyBox. This work is focused on providing a web-based system which is developed using HTML, PHP and MariaDB while KeyBox is developed using NodeMCU ESP32. To integrate different platforms, uses of Application Programming Interface (API) and Relational Database Management System (RDMS) are required to allow interaction between platforms. The designed system is able to automize reservation and check-in process while also minimizing physical interaction between homestay owner and guest.

Keywords: Keybox, Homestay, Web-Based System, Contactless System, Internet Of Things (Iot)

1. Introduction

In recent years, the demand for autonomous systems and the implementation of Internet of Things (IoT) technology has rapidly increased. The purpose of an autonomous system usually is to provide user convenience and minimize human efforts. Most autonomous systems implement IoT technology which is built from the connection between intelligent devices and other computing devices [1].

As the implementation of IoT technology has been outspread, it also can be adapted to Homestay Industry. Homestay check-in procedure usually requires physical interaction between owner and guests for registration along with key issuance to the guest. The owner also needs to verify and check the accuracy of registration information during the reservation process [2]. Sometimes, homestay owners

*Corresponding author: noraisah@uthm.edu.my

are not available and not able to physically meet-up with guests to issue keys. This is when an Autonomous Self Check-In KeyBox and Reservation for Homestay Operation comes in handy, the designed system enables guests to proceed with the registration and check-in process without physical meet-up with the owner.

With the recent Covid-19 outbreak, people are encouraged to minimize human interaction to subdue the outbreak of Covid-19. For homestay operations, where registration and check-in process usually are done by the owner meeting up physically with guests. As the procedure involves interaction between homestay's owner and guest, it makes both owner and guest vulnerable to Covid-19 spread.

Moreover, contactless self-check in system allows guests to make registration and check in through a webpage and access keys from a KeyBox hidden near the homestay entrance using passcode acquired from the system. Additionally, the developed system can provide convenience for homestay's owner where the owner does not require to meet up with guests directly. This also indirectly saves time and energy for both owner and guest. This developed system is able to automate reservation and check-in process for homestay operation while minimizing physical interaction between homestay owner and guest.

2. Methodology

The developed system consists of KeyBox, admin webpage and reservation form which are integrated and linked with each other using a cloud server to form a complete system. The KeyBox is powered by NodeMCU ESP32 while admin webpage and reservation form developed using HTML, PHP and MariaDB database. After development, the system is deployed on a web hosting so that the system is able to be accessed through web browser.

2.1 Application Programming Interface (API)

An application programming interface (API) is a set of rules and protocols that defines how software programs can interact with one another. It allows different programs to communicate with each other and share data, functions, and processes.

APIs are commonly used to enable a web application to access a database, or to allow two different systems to communicate with each other. They can be used to expose the functionality of a software library or framework, or to create a custom interface for a web service [3].

There are several different types of APIs, including web-based APIs, operating system APIs, and library-based APIs. Web-based APIs allow different web-based systems to communicate with each other over the internet, while operating system APIs allow different programs to access the functionality of an operating system. Library-based APIs allow programs to access the functionality of a software library [4].

In this developed system, there are many APIs implemented to ensure each platform can interact with each other. Such as HTTPClient a library-based API is used is ESP32 to establish connection with database thru HTTP request to PHP script. jQuery API used in webpage to retrieve and alter database and add some animation to the webpage. PHPMailer and Gmail API is used to send response email to quest when admin approve or reject their reservation request.

2.2 Cloud storage

Cloud storage technology first debuted in 2002, when Amazon launched a variety of related services. According to Leonid Arshinskiy et al [5], Amazon Web Services is the first cloud storage and other storage such as Dropbox, Google, and Azure storage by Microsoft started appearing in 2008.

Cloud storage is a type of cloud computing model that stores data on the Internet via cloud computing. Cloud storage is a more cost-effective and scalable option than storing content on premise hard drives or storage networks.

2.3 Relational Database Management System (RDMS)

RDBMS stand for Database Management System, DBMS is a system software responsible for the creation, retrieval, updating and management of the database. It ensures that our data is consistent, organized and is easily accessible by serving as an interface between the database and its end-users or application software [6]. Compared to Database Management System (DBMS), RDMS stores data in the form of a collection of tables and relations can be defined between the common fields of these tables. Most modern database management systems are based on RDMS.

2.4 Overall system block diagram

Figure 3.2 shows the overall block diagram of the system where KeyBox, admin webpage and reservation form are linked and connected via a database which are accessible via PHP scripts. The reservation form allows a flexible check-in process for homestay guests. Guests can access the reservation form via any web browser on an electronic device with an internet connection. After the check-in process, all the guest's credentials are stored in database and the system will send passcode to the guest using the data from the reservation form after admin approve the request from Admin Webpage. Then guests can enter the passcode on KeyBox to access the keys of the homestay during check-in day. In Admin Webpage, admin also can monitor key-box status and access homestay log history.

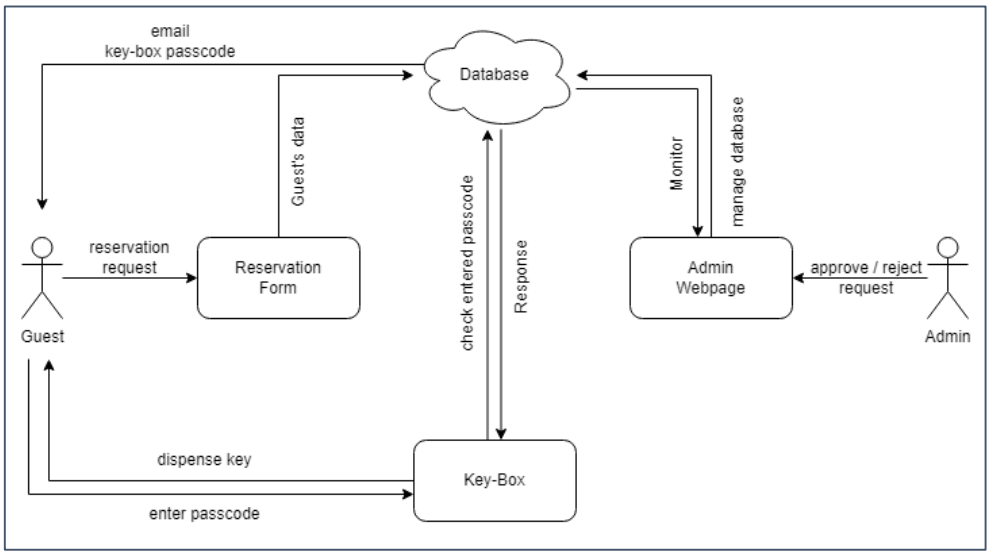

Figure 1: Overall block diagram of developed system

2.5 KeyBox

Figure 2 shows the interconnection between ESP32 and database used to interact with database and Figure 3 shows the flowchart of the Keybox operation. Keybox read keypad on release and store the input in String data type. Guest can press "*" to clear and re-enter the passcode or press "#" to enter. The Keybox will send entered passcode to the webpage (PHP script) via HTTP Request GET Method API. The PHP script will compare received passcode with database and the reply accordingly.

Once the correct passcode is entered, the LCD will display "PASSED" and the Keybox will unlock and dispense keys for the guest. However, it will prompt user to try again when wrong passcode

is entered. The dispense mechanism uses 12V solenoid door lock to unlock and a relay to trigger solenoid using microcontroller.

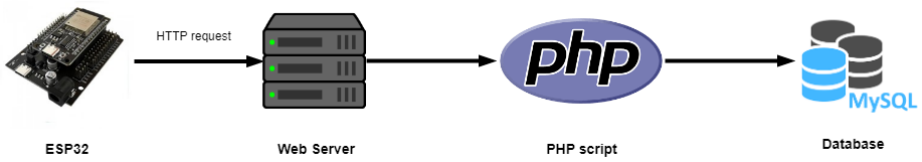

Figure 2: Interconnection between ESP32 and database

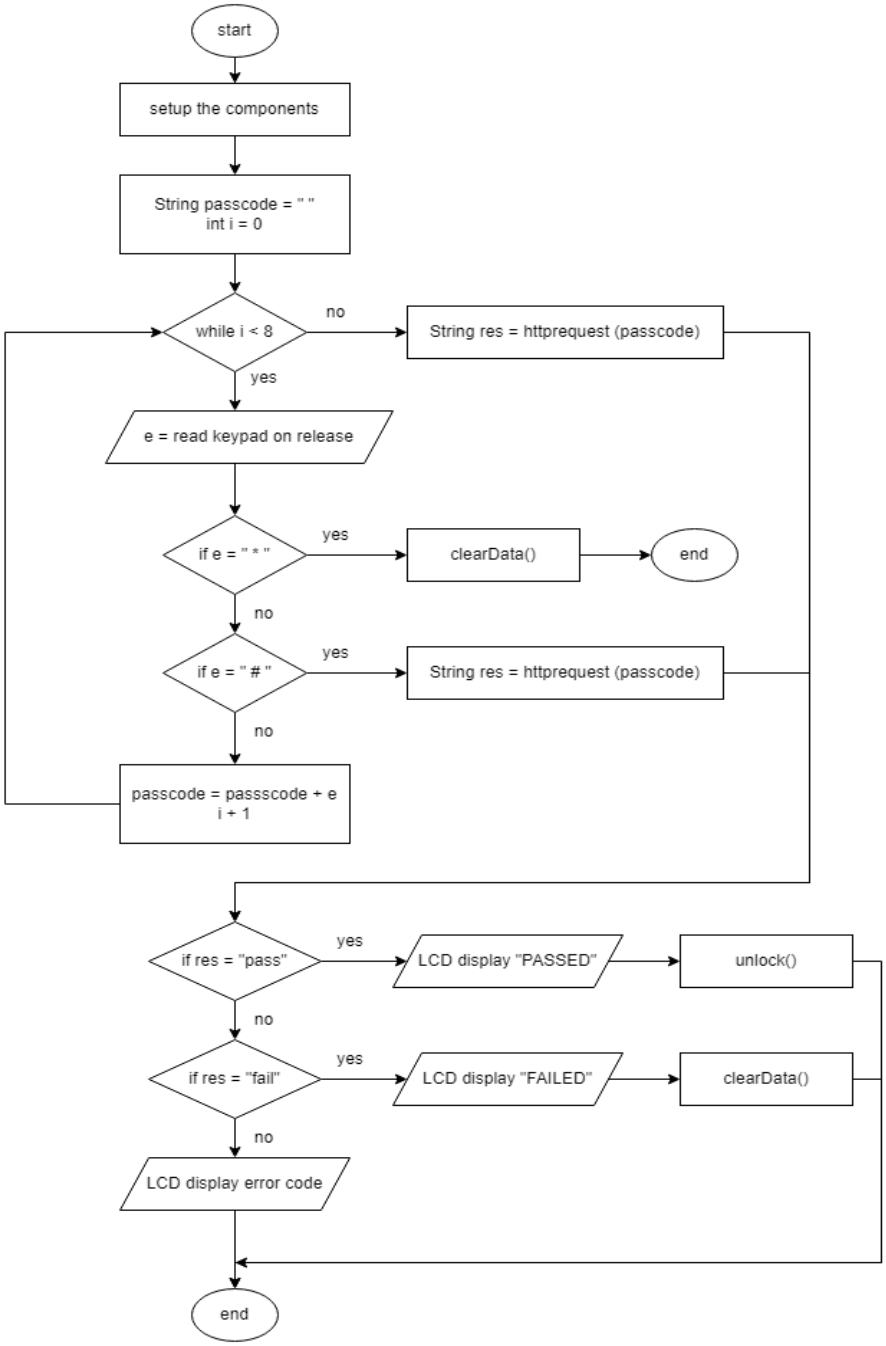

Figure 3: Flowchart of KeyBox operation

2.6 Reservation form

Figure 4 illustrates the flowchart of reservation form operation. Guests can access reservation forms with an address link using any web browser. Guests are required to fill in all information and check reservation dates before being allowed to submit the reservation form. Reservation form uses jQuery API to check dates requested by guest are available before proceeding to form submission. Reservation forms are mainly written in HTML, CSS, JavaScript, and PHP script.

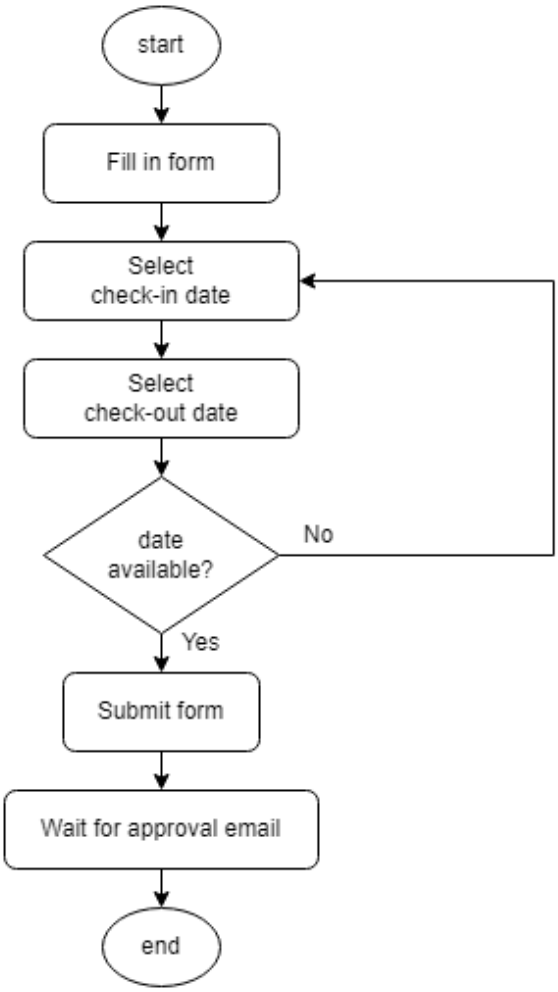

Figure 4: Reservation form flowchart

2.7 Admin webpage

Admin webpage allows admin or homestay owner to monitor and manage homestay system remotely. Admin can accept and reject reservations made by guests and monitor log history of homestay guest. PHP session is implemented to build login feature on admin webpage for little more security feature.

Multiple PHP scripts are used to manage databases accordingly from admin webpage such as insert accepted request, delete rejected request, and retrieve data from database. PHP scripts are also used to send email to guests regarding their reservation status.

2.8 Database Server

Designed system uses relational database management system (RDMS) to manages the storage, retrieval, and manipulation of data in database. Database server allows other programs or computers to access data in database. This enables web-based integration into designed KeyBox.

Database uses MariaDB server type and UTF-8 Unicode charset. Protocol used to connect with database is via TCP/IP. Data is arranged into three database tables which are **"userdata"**, **"admindata"** and **"homestay"**. Figure 5 shows the table structure of **"userdata"** which is used to store all information about homestay reservation made by guest via reservation form webpage. Information such as full name, mobile number, email, check-in date and check-out date input by guest. While column **"id"** with auto increment is used to give unique id for each reservation request, column "passcode" store passcode randomly generated by PHP script for KeyBox and **"status"** with label **"0"** for pending and **"1"** for approved.

Figure 6 depicts table structure for **"admindata"** used to store admin login information such as username, password, and email from admin webpage registration. Figure 7 shows table structure for **"homestay"** used to signify KeyBox status either locked or unlocked. When a guest enters the correct passcode on check-in date to take keys, status will be unlocked. When entered on check-out date to return keys, status will be locked. This data is displayed on the admin webpage for admin monitoring.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	fullname	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	mobile_no	text	utf8mb4_general_ci		No	None			Change Drop More
☐	4	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	nric	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
☐	6	checkindate	date			No	None			Change Drop More
☐	7	checkoutdate	date			No	None			Change Drop More
☐	8	passcode	int(8)			No	None			Change Drop More
☐	9	timestamp	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
☐	10	status	int(11)			No	None			Change Drop More

Figure 5: "userdata" table structure

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	username	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	password	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More

Figure 6:"admindata" table structure

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	status	int(1)			No	None			Change Drop More

Figure 7: "homestay" table structure

3. Results and Discussion

This section discusses the functionality of each sub system and ensures software meets requirements and is free from major bugs. This testing verifies the software's overall functionality and software behaves as expected.

3.1 Reservation form testing

A reservation form should enable guests to make a reservation. The reservation is then registered on database and waiting for admin response regarding the request. This page has form validation to prevent invalid data stored in the database. The form validation is done using HTML5 feature and JavaScript. Users can check date availability for check-in and check-out date by clicking "check availability" hypertext. When user clicks on the hypertext, a HTTP request is sent to database along with user entered check-in date and check-out date to check the date availability.

Figure 8 shows the prompt message when chosen date available. Users are allowed to submit the reservation form. Otherwise, the user will be prompted to check date availability. After verifying date availability, user can submit reservation form by clicking "Submit" button and will be directed to the other page which displays successful reservation status as shown in Figure 9. Then the user can wait for an approval email when admin responds to the request.

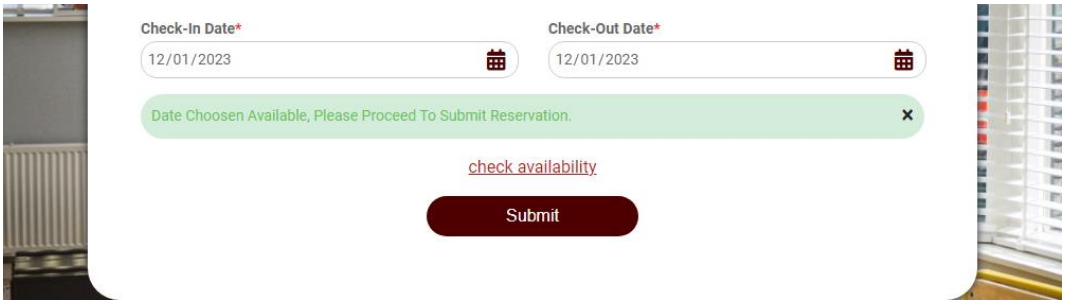

Figure 8: Prompt message when chosen dates are available

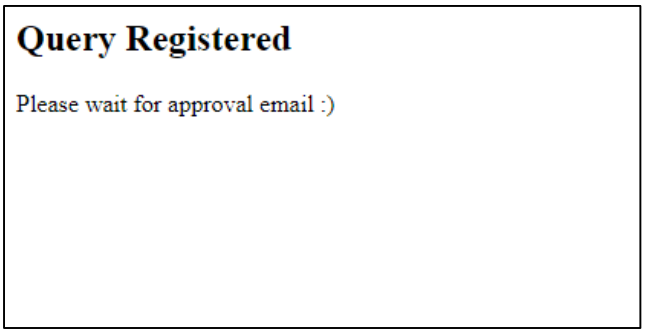

Figure 9: Successful reservation submission

3.2 Admin webpage testing

Before accessing the admin webpage, a login is required to increase the security factor of the overall system. The webpage utilizes “Session” method in PHP to direct to another page with login information data which is stored in browser session storage. If user input incorrect password or username, there will be prompt that showing incorrect password is entered.

Upon successful log-in, the user will then be redirected to the home page shown in Figure 10 where there will be log history of guest list, pending request approval and status of KeyBox in homestay that indicate either is locked or unlocked. On “Notification” panel, admin can review reservation request and either approve it or reject it. Upon admin response, an API is called to send email accordingly to the recipient as shown in Figure 11 and Figure 12.

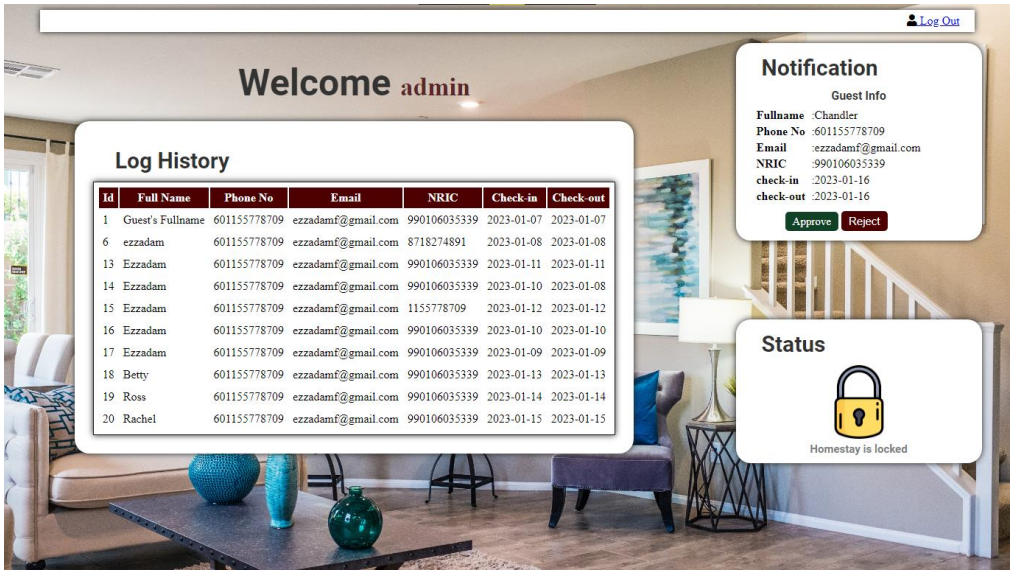

Figure 10: Homepage of admin webpage

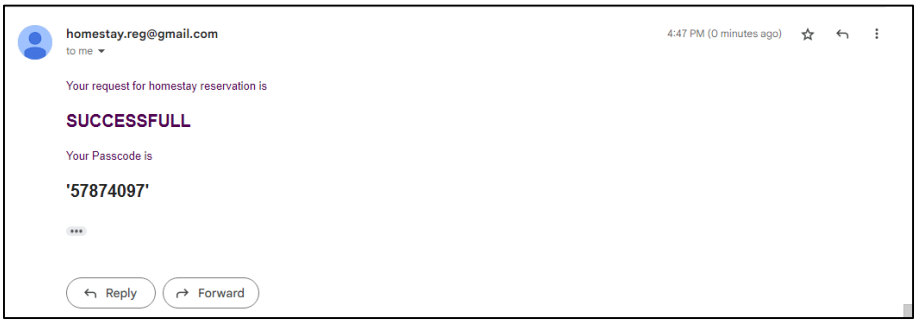

Figure 11: Approval email sent to guest's email

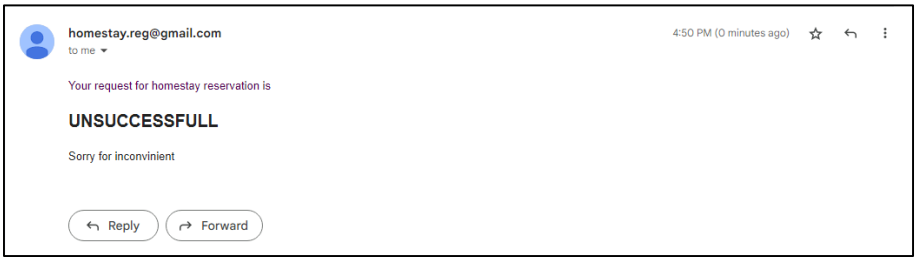

Figure 12: Rejection email sent to guest's email

3.3 KeyBox testing

The main components used in developing KeyBox are LCD with I2C module, keypad, solenoid lock, relay and NodeMCU ESP32 with base board. Other components are brackets for door lock, 12V battery with holder and enclosure box as casing for the prototype.

The KeyBox is designed and developed as shown in Figure 13. Upon booting, the KeyBox will connect to WiFi and display WiFi status on LCD. After few seconds, LCD will display "ENTER PASSCODE" as shown in Figure 14(a) and user can enter passcode received from email. User can press "*" to clear screen and enter passcode again and press "#" to enter. When user press "#", KeyBox will send entered passcode to server for further processing.

When user enters correct passcode on check-in date to take keys, server will update KeyBox status to "unlocked" before unlocking KeyBox and LCD display "UNLOCKING" as shown in Figure 14(b). When the user enters the correct passcode on check-out date to return keys, server will update KeyBox status back to "locked" before unlocking the KeyBox to allows user to place the keys. When the wrong passcode entered, LCD will display "WRONG PASSCODE" for few seconds as shown in Figure 14(c) before user can try again. If the KeyBox is unable to connect to WiFi or server, LCD will display "WIFI ERROR" as shown in Figure 14(d).

Figure 13: KeyBox design

(a)

(b)

(c)

(d)

Figure 14: KeyBox LCD display

4. Conclusion

For conclusion, Autonomous Self Check-in KeyBox and Reservation for Homestay Operation is successfully developed using IoT and MariaDB database. The developed system is fully functional and has successfully met all the objectives set. However, there are some limitations in the system therefore there is room for improvement. For software, further optimization can be made in terms of responsiveness and security features while also resolving minor bugs found in the system. The reservation page/form can be improved by adding rental/service fee information, a better rental calendar for organizing reservation process and a function for payment procedures. As for hardware improvements, redesigning KeyBox to be much more secure and rigid while also improving the unlocking mechanism for dispensing keys. For additional, server also can be upgraded to have better server uptime and responsiveness.

Thus, the system introduces an automated reservation and check-in process for homestay operation while minimizing physical interaction between guest and homestay owner.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] S. Ibrahim, V. K. Shukla, and R. Bathla, "Security Enhancement in Smart Home Management Through Multimodal Biometric and Passcode," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 2020, pp. 420–424. doi: 10.1109/ICIEM48762.2020.9160331.

[2] "6 Stages of the Guest Check-In Procedure," Setupmyhotel, 2022[Online]. Available: https://setupmyhotel.com/train-my-hotel-staff/front-office-training/778-stagesof-check-in-procedure.html. [Accessed: 11-Jan-2023].

[3] J. Ofoeda, R. Boateng, and J. Effah, "Application programming interface (API) research: A review of the past to inform the future," *International Journal of Enterprise Information Systems*, vol. 15, no. 3, pp. 76–95, Jul. 2019, doi: 10.4018/IJEIS.2019070105.

[4] S. M. Sohan, F. Maurer, C. Anslow, and M. P. Robillard, "A study of the effectiveness of usage examples in REST API documentation," in *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2017, pp. 53–61. doi: 10.1109/VLHCC.2017.8103450.

[5] L. Arshinskiy and G. Shurkhovetsky, "Methods of Information Security in Cloud Storages," in *Transportation Research Procedia*, 2022, vol. 61, pp. 455–461. doi: 10.1016/j.trpro.2022.01.074.

[6] M. C. Fernandez-Baizán, E. Menasalvas Ruiz, and J. M. Peña Sánchez, "Integrating RDMS and Data Mining Capabilities Using Rough Sets," in *Knowledge Management in Fuzzy Databases*, O. Pons, M. A. Vila, and J. Kacprzyk, Eds. Heidelberg: Physica-Verlag HD, 2000, pp. 371–384. doi: 10.1007/978-3-7908-1865-9_22.