



RESEARCH ON SMART REFRIGERATOR USING RASPBERRY-PI

Saparov Mekan^{1*}, Ryuichiro Ohyama²

^{1,2}Tokai University, Hiratsuka, Japan.

*Corresponding author

DoI: <https://doi.org/10.5281/zenodo.7782411>

Background and purpose of development. A smart home is a house or apartment equipped with advanced technology to improve quality of life and comfort. The technology includes smart devices, automated procedures, and remote controls, all of which can be managed through a central control unit. The goal is to make these homes accessible to everyone, with a focus on user-friendly technology. I want to develop new technologies and create a low-budget smart refrigerator to solve small daily problems such as meal planning and grocery shopping.

Development goals. I intend to develop a smart refrigerator that can be easily integrated with various devices such as mobile phones, PC, or tablets. With this, people can easily access information about the availability of products in their refrigerator and quickly check what they need to restock without having to physically open the fridge. When the quantity of a product drops below a certain level, a notification will be sent to the owner to remind them to restock.

Additionally, the mobile application will have a custom shopping list feature, where necessary items will be automatically added. The project will also make use of barcodes to make the process more efficient and convenient. Another key feature of the project is the ability to suggest recipes based on the ingredients available in the refrigerator, this will help save time and make daily life more comfortable.

Development stages. The development of this project is divided into three stages: the backend, the Raspberry Pi setup, and the mobile interface. One of the first steps was deciding on a framework for the backend. After looking at options like Laravel, Django, and NodeJS, we decided that Laravel would be the best choice for this project. One of the main reasons for this decision is that Laravel's Eloquent ORM is easy to use and allows for seamless interaction with databases using an object-oriented approach. Additionally, Laravel's Blade templating engine makes it easy to separate logic and presentation, making it a great option for developers with limited experience in creating templates [1]. It's important to note that these reasons are specific to Laravel and may not apply to Django or Node.js. However, we took the specific needs of this project into account when making our decision and determined that Laravel would be the best fit for the backend of this project.

Setting up the Raspberry Pi component of the project was relatively easy. I wrote a Python script that allows us to read the barcodes of products and send that information to the backend for further processing.

The last step of the project was to build a mobile app for users to view the contents of their fridge. Initially, I thought of creating a hybrid app that would work on both iOS and Android. But after further thought, I decided to create a native iOS app as it has a larger user base than Android. To make the development process simpler, I chose to use SwiftUI over UIKit as it gives a more straightforward approach.

Working principles. The system (figure 1) allows users to track the contents of their refrigerator using a mobile app. Users add products to the fridge by inputting info into the app, which is then sent to a database for easy access from a smartphone. When product used,

a Raspberry Pi scans barcodes on products as they're used to update the database and inform the user of changes.



Figure 1. Working principle of the developed smart refrigerator

This information is also sent to a web server for access from any device with internet access. The app also provides recipe suggestions based on the products in the fridge, helping users make use of their food inventory. The system aims to make it easier for users to track their refrigerator contents.

REFERENCES

- [1]. "Laravel installation" <https://laravel.com/docs/9.x/installation>