APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 1510-1528
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

AutoFeed: Smart IoT Fish Feeding

Mohamad Faiz Rosdin[1], Rozita Abdul Jalil[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: rozita@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.081

Article Info

Received: 12 June 2025
Accepted: 18 June 2026
Available online: 30 June 2026

Keywords

IoT Fish Feeder, Automated Fish Feeding, ESP32 Microcontroller, Flutter Mobile App

Abstract

The AutoFeed: Smart IoT Fish Feeding aims to help the fish owner take care of their fish by automating the feeding process, reducing the manual method, and ensuring proper nutrition. Many fish owners face difficulties in maintaining regular feeding schedules, which can lead to overfeeding or underfeeding the fish, causing harm. This project automates the feeding process using IoT integration, reducing the need for pet owner intervention while ensuring the fish receive the correct amount of food at the right time. The system integrates an ESP32 Microcontroller, Ultrasonic Sensor, and LCD I2C 16x2, along with a mobile application built using Flutter and connected to Firebase to store data, control the feeder, set feeding schedules, monitor food level, and receive low food level alerts. The system successfully automated the feeding process and future work could include smart home integration and AI-based adaptive scheduling.

1. Introduction

Fish are animals that people of all ages enjoy keeping and caring for. There are many types of fish, and most require extra care to stay healthy. Proper care helps maintain their scales and fins in good condition, preserving their beauty. To keep fish healthy, the owner should regularly change their water to prevent irritation to their scales. Maintaining a proper diet and feeding them on a consistent schedule also helps reduce stress, which can otherwise make them sick.
The fish owner usually having the same problem which is the diet of the food. This is said because the owner usually busy with their daily life such as work or vacation which will leave the fish hours without any food and can cause harm to them. Since the fish cannot obtain the food by themselves, they rely on the owner to get food.. Certain fish will eat their species to maintain their diet, which is what the fish owner does not want to happen. The fish need consistent hours of food in order for them to be healthy and grow. The development of the automated fish feeder by applying various methods and approaches to satisfy the fish owner [2].

Many issues have arisen in which the owner doesn't have the time to feed the fish regularly on a daily basis, which results in poor growth of the fish and depleted health. This all can happen since the owner cannot monitor their fish's health and nutrition regularly because of the job or any work they had during the day. The owner also cannot put the food too much on the aquarium or pond since it will make the water more toxic and lead to stress and eventually the fish die since the fish growth is based on feed ration, genetics and water temperature [9]. If the pellet is added in quantity too much, it will affect the water quality and deteriorate the growing environment of the fish [3]. Poor water quality can induce oxidative stress in fish, potentially leading to mortality [1].

2. Literature Review

This section discusses existing systems to compare them with the proposed system. Each system offers its own strengths and weaknesses, depending on how well it meets the needs of the user.

2.1 Organizational Case Study

Feeding the fish might be the hardest for the fish owner who doesn't have time to take care of the diet for the fish which will significantly affect the growth and health of the fish. Relying on manual feeding method will consume time of the owner and also need the present of the fish owner in order to feed the fish. For this case study, the issue is for the fish owner who struggle to keep maintaining the fish diet and the focus is for the fish store, Jai Joran, that also having the same problem to feed the fish.

The business has been providing fish and also fishing equipment for the customer. The focus is on the Koi fish which is the beautiful among the fish. It might be the famous fish to be keep in a pond because the colour and the behaviour of the fish that will attract the attention of the eye that look at them. The koi also one of the fish that require food in small amount yet repetitive so it allow them to digest the food.

The problem arises when the business owner or their worker on vacation, leaving their fish unattended and hungry which will affect their health. The worker also cannot give too much food for one time feeding because it will make the water toxic and lead to the stress of the fish. This problem is faced not only by business owners but also by fish owners, who all struggle to maintain their fish's diet and health.

2.2 Internet of Things (IoT)

Nowadays, technology has evolving and the creation of Internet of Things (IoT) has been created. The latest Technology IoT, Internet of Things, which has been use to every device, every sensor, every software that are connected to each other and those devices we were able to access remotely through a smart phone or computer is called Internet of Things [5]. The IoT technology has become more popular each day and the creation more technology using the IoT has been built. IoT has been use to interconnect a number of devices, people, data, and processes, by allowing them to communicate with each other seamlessly [4]. IoT can help to improving the different processes to be more quantifiable and measurable by collecting and processing large amount of data [4].

Since the IoT technology allows for more flexible project or product to be build, it has been the popular technology to be develop. IoT is not only a thing, it actually has a lot of form to it. IoT devices which are the second components of the IoT platforms, could be in the form of sensors, actuators, IoT gateways or any device that joins the cycle of data collection, transmission, and processing [4]. IoT consist a lot of components in it which can be choose to be use. There is no limit in term what the IoT is being used for because it allows to interconnect to number of devices. IoT technology, one of the world's hottest technology areas, is now gradually being used in various fields [10].

2.3 IoT Based Automatic Fish

Feeding the fish is one of the important parts in keeping the fish since the growth of the fish is based on their feed [7]. Because of that, the development of the project is focusing on to save the fish owner time and the labour of it to feed the fish. In their study, it shows that most of the fish owner still using the manual or conventional method to feed their fish. This method is not efficient since it requires human labour and time. Since this is the problem that they want to solve, they have proposed this system which allow the fish owner to feed their fish far away from the pond by using the IoT technology [7].

The system will be using the Blynk application as the controller to control the hardware and monitoring the system. The hardware will be connected to the ESP8266 microcontroller which will control all of the component. It will use the proximity sensor to detect the level of food in the container and servo motor to dispense the food. Thus, the owner might forget to feed the fish and make the feeding schedule becoming unordered. To solve the problem for this, they develop the system which the user can set timer or schedule to feed the fish based on the fish need [7]. This will save both user time and labour. Then, the IoT Based Automatic Fish Feeder also will allow the user to monitor the food level in the container by sending the data whether the container is empty or the container still has the food in it.

2.4 Green Energy Powered Autonomous IoT Catfish Feeder

Rapid development in science and technology to ease and aid human in solving problems that may hinder productivity and reduce efficiency [6]. One of the technologies that will help to increase the productivity and efficiency is autonomous fish feeder. The owner of the fish also might forget to feed them based on the schedule which is a problem since it vital for the fish health and growth [6]. They develop a system using the IoT technology in order to solve the problem.

The system will be using Arduino Mega 2560 as the microcontroller, ESP-8266 module as Wi-Fi module, RTC DS3231 clock module, MG996 servo to open and close the food storage lid, DC motor 755 80 W with BTS 7960 driver motor for food launcher, 4x4 keypad to set feeding schedule. The system doesn't have mobile application but it integrated with Telegram application where it sends data through the Telegram. The fish feeder also cannot be set the timer since it will dispense the food 3 times a day. This is to save the fish owner time and increase the efficiency of the feeding method.

2.5 IoT System on Dynamic Fish Feeder

The main problem to maintaining a fish pond is the food that given to the fish whether it's enough or not and feed the fish by using the manual method [8]. To solve this problem, they develop a system using the IoT technology which will be use the ESP8266 as the microcontroller, ultrasonic sensor is programmed with the controller to detect the level of food, waterproof ultrasonic to detect existing fish, two servo motors were used to move the waterproof sensor to attract the fish and to dispense the food to the fish when existed, and will be using Blynk mobile application to monitor the fish feeder.

To save the fish owner time, the system will be using a sensor to detect the food and automatically drop the food from the fish feeder container. This is to make the feeding method become more efficient since the owner doesn't need to be there at the pond to feed the fish. The Blynk application for the IoT System on Dynamic Fish Feeder will be used to monitor the food level in the container, temperature and humidity in the food container.

2.6 AutoFeed: Smart IoT Fish Feeding

The AutoFeed: Smart IoT Fish Feeding will develop to fulfil the user need in order for it to solve the user problem which is the feeding method. The AutoFeed: Smart IoT Fish Feeding also will be run with all the feature running accordingly. The user for the system will be require to login to the application in order for them to control the fish feeder. The login is required as it will be used to store data and display it to the user. The admin can manage and see all the user that register to the application. Then, the system will have the feed schedule module which will allow the user to set the timer to feed the fish. This is to allow more flexibility to the user and feed the fish based on their need. The user also can set the timer for each day meaning that user just need to set just once and doesn't need set the timer every day.

Furthermore, the develop system also will be having the food level detection module which will allow the user to see the food left in the container without they need to manually check. This module also will show notification to user smart phone when the food is left in the container is less than 20%. The develop system will show the food level using percentage since it is the easier method to show to the user and user can easily understand it. Using percentage for the food level might be the most suitable for this system. Then, the monitoring module allows user to monitor the pond via livestream using the ESP32-Cam that attached to the food container. The user profile is to allow user change about their basic detail and the profile picture. The last module will be the report module which will show how many amounts of food that been used each week, month, and year. This is to allow the user to track their fish intake and to see the growth of fish intake.

2.7 Comparison with the Existing Systems

The IoT based automated fish feeder system address some challenges such as inconsistent feeding time and manual effort by integrating components such as Esp32 Microcontroller and sensor. While existing system offer the solution with basic functionalities but it still lacking the advance feature. For this part, the AutoFeed: Smart IoT Fish Feeding will be compared with the other three existing system to be referred as the reference to develop a functional and efficient automated fish feeder.

Table 1 *System Comparison*

Features/System	IoT Based Automatic Fish Feeder	Green Energy Powered Autonomous IoT Catfish Feeder	IoT System on Dynamic Fish Feeder	AutoFeed: Smart IoT Fish Feeding
User Log In	No	No	No	Yes
Feed Schedule	Yes	Yes	No	Yes
Food Level Detection	Yes	Yes	Yes	Yes
Monitoring	No	No	No	Yes
Report	No	Yes	Yes	Yes
User Profile	No	No	No	Yes

Table 1 compare three existing system regarding to the IoT based automated fish feeder. The IoT Based Automatic Fish Feeder will only have two features which are feeding schedule and food level detection but it only detects whether the food container is empty or not. Then, the Green Energy Powered Autonomous IoT Catfish Feeder consist of three features which are feeding schedule, food level detection and report. The system doesn't have its own application and it uses Telegram to send the data. After that, the IoT System on Dynamic Fish Feeder will be having 2 features which are food level detection and report.

The AutoFeed: Smart IoT Fish Feeding will have all the listed features in order for the system to be efficient functional to the user. This to allow user to fully utilize the system and fully automated the feeding method which can ease the user. The application will user friendly so it can be easy to understand and easy to use.

3. Methodology

The System Development Life Cycle (SDLC) will be used to develop the system project with the involving of the phases such as planning, analysis, design, implementation, and system. The chosen model of the SDLC that will be use during this project will be the Prototype Model.

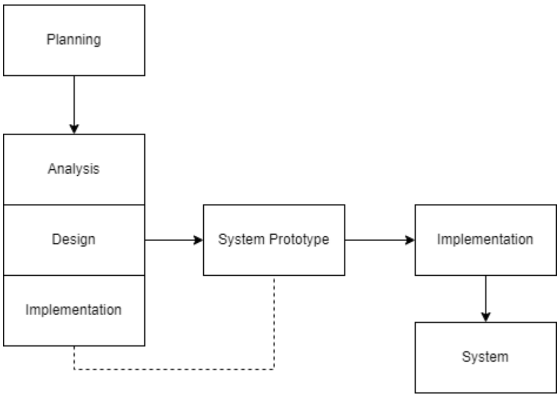


Fig. 1 *Prototyping Model*

For Autofeed: Smart IoT Fish Feeding system will be use the prototype model as the development methodology. The prototype model is a software development method where a working prototype of a system is created early in the project. The protype will show the main feature and the feedback from user is the key for this method because we can create the system based on user need. The prototype will show the user what might the system feature will be look like. The timeline of the project is achieved and followed. The Gantt chart is shown in Appendix A.

Table 2 *Software development phase and activities*

Phase	Task	Output
Planning	• Determine the objective, scope and the requirement for the project.	• Develop Gantt chart to know the project timeline and also to have the clear idea of what the project will achieve.
Analysis	• Gathering the user need and features that are needed in the project through interview.	• To understand the user need and the features for the system.
Design	• Creating the user interface and the database that will be use in the system.	• To create the user interface to know the flow of the system and how it will operate.
Implementation	• Develop the system	• To create the model of the system that are working based on the feature.
System Prototype	• Build the prototype that are fully working to be present to the stakeholder.	• To gather feedback from the stakeholder regarding to the prototype.
Implementation	• Make changes to the prototype based on the user feedback to achieve user need.	• To allow the system to fully achieve the user requirement and need.
Deployment	• Deploy the system.	• To deploy the finish system.

4. Analysis and System Design

The analysis identified user needs and problems in managing fish feeding schedules. System design focused on integrating IoT hardware with a mobile app to ensure smooth and automated operation. Usability and reliability were prioritized to improve user experience.

4.1 System Requirement Analysis

In this section, it will outline of the overall features of the AutoFeed: Smart IoT Fish Feeding, functionalities and to ensure that the system is being developed to meet the user expectation and to make sure that they know how to use the application according to the flow of the system. The system requirement analysis is a process to identify, analyse, and documenting the functional and non-functional requirement. This is to allow the system to fulfil the user need and the project objective. By documenting the functional and non-functional requirements, this analysis will ensure that the design meets the user expectations for accuracy, ease of use, and functionality.

Table 3 *Functional Requirement*

No	Module	Description
1	User Authentication Module	• Allow the new users to register new account before login. • Allow the existing users to login with the email and password. • Redirect the valid users to home page when successful login. • Allow the admin to see the new registered user.
2	Feeding Schedule Module	• Allow the user to set the timer to feed the fish. • Allow the user to set timer based on their fish need. • The food will dispense based on the time set by the user.
3	Food Level Detection Module	• Allow the user to see the food level left in the container. • Allow the notification to display when food level reach below 20%. • Allow the user to refill the food before it runs out.
4	Monitoring Module	• Allow user to view the live feed from the pond. • Allow user to monitor the fish and pond through application.
5	Profile Module	• Allow the user to update their profile. • Allow the administrator to see the updated user profile. • Allow the user to interact with the application.
6	Report Module	• Allow to user to track the total food that has been used each week, month, and year. • Allow the user to track the growth of the fish feeding behaviour.

Table 4 *Non-functional Requirement*

No	Requirement	Description
1	Performance	• The system should be always usable
2	Operational	• The loading time required for an application is no more than 1 minute
3	Security	• The system should be user friendly
4	Reliability	• The system should be able to work properly.

Table 5 *User Requirement*

No	Requirement
1.	All users must be able to enter their email and password for registration and login requirement.
2.	The user will get notification when the food level in the container less than 20%.
3.	The user can set the fish feeding schedule by selecting the time they want.
4.	The user can get the real time food level in the food container.
5.	The user can power on or off the device using the application.
6.	The user should be able to update their information in the profile.
7.	The user should be able to see the total amount of fish food they use each month.
8.	The administrator should be able to see all the user.

4.2 System Analysis

The system analysis results focused on describing the system details with the included diagrams which: Context Diagram, Level 0 Data Flow Diagram (DFD 0), Entity Relationship Diagram (ERD) and flowchart. Appendix B contains the context diagrams to present the overview of the interaction between the system and its user. The diagram will indicate the system's limit, display how the external entities such as the user and AutoFeed device interact with it. Context diagrams also show the input and output to and from its user and system. Appendix C contains the Level 0 Data Flow Diagram (DFD 0), which is a graphical representation of the flow of data or input from an entity through a process, generating output to another entity or stored in data storage. DFD shows each input and output for each entity and process. Appendix D contains the Entity Relationship Diagram (ERD), is a visual representation of the AutoFeed: Smart IoT Fish Feeding system, consisting of four main entities: Admin, User, Schedule, and Device. This is to ensure the efficient fish feeding system. Appendix E contains the flowchart, a diagram showing the steps used to visualize, analyze, and communicate processes.

4.3 System Design

In this phase, the system design will be involving to create the blueprint of the system architecture, components, modules, and data that were required by the system. The system design will ensure that the system will meet the user requirement while also focusing on the technical challenges. By focusing on the efficiency and functionality, it provides a solid framework for the development, maintenance, and potential future enhancements. During this phase, the interface and database were designed to establish the architectural and functional framework for the Autofeed: Smart IoT Fish Feeding.

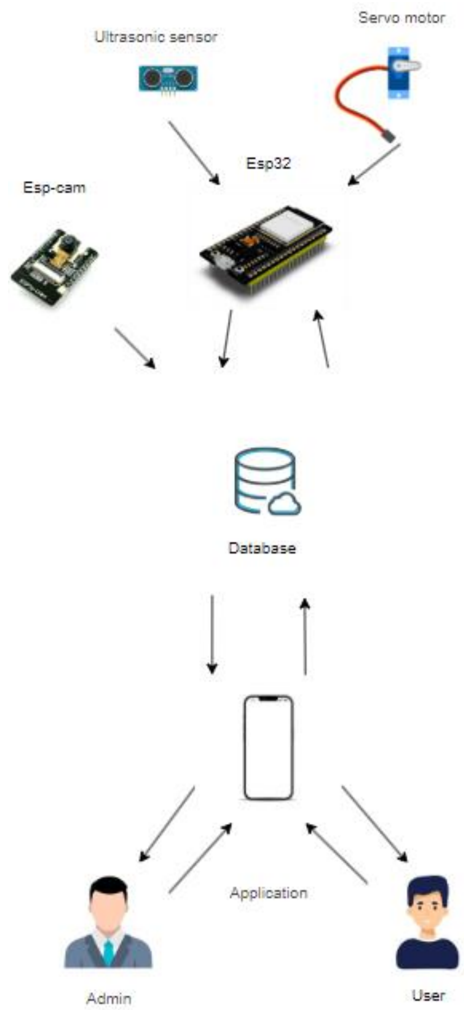


Fig. 2 *System Architecture*

The following are the tables from the database that have been designed and extracted from the entity relationship diagram. The database for this system was created using the Firebase. The schema tables for the AutoFeed: Smart IoT Fish Feeding are:

i. `Admin (AdminID, email, password)`
ii. `User (UserID, name, email, password, phone, food_usage, feedingLog, foodLevel)`
iii. `Device (DeviceID, power, foodLevel, UserID, ScheduleID)`
iv. `Schedule (ScheduleID, day, time, enabled, UserID)`

The objective of interface design is to create attractive and simple layouts which allow smooth communication between humans and systems. The Autofeed: Smart IoT Fish Feeding systems interface design focuses on intuitive layouts and easy navigation. While the user interface makes it easier to set up feeding schedules, check food levels, and report monthly food usage. Figure 3 – 8 shows the interface design.

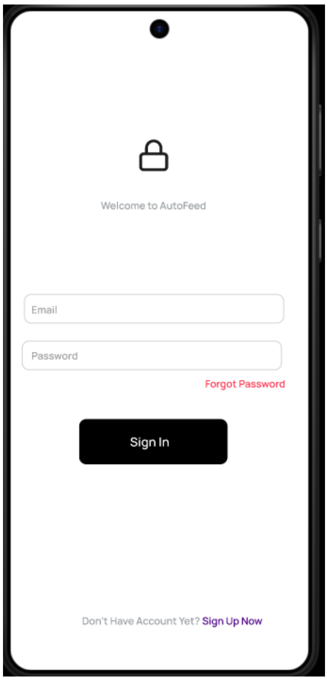


Fig. 3 *Login interface*

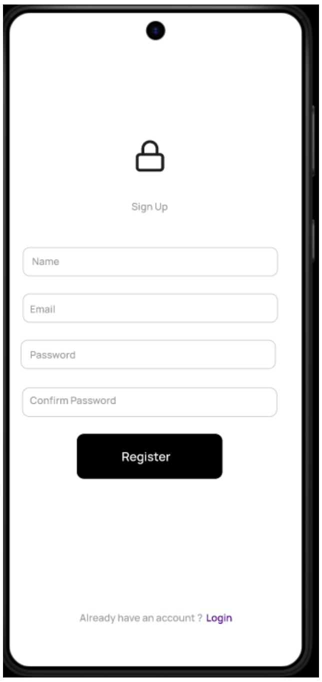


Fig. 4 *Register Interface*

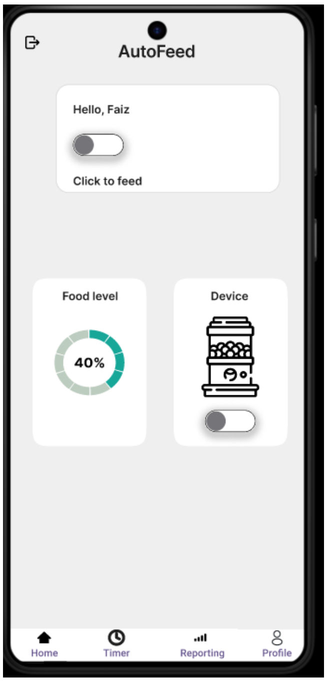


Fig. 5 *Home Interface*

Fig. 6 *Timer Interface*

Fig. 7 *Reporting Interface*

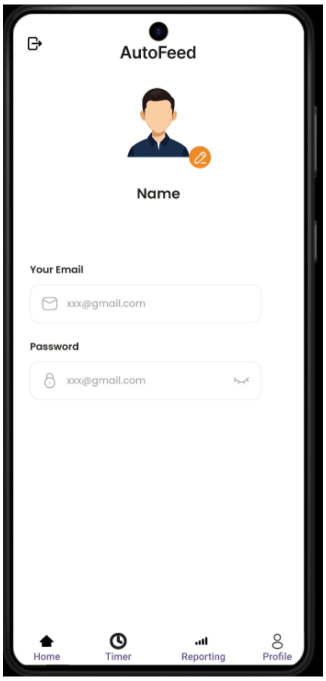


Fig. 8 *Profile Interface*

5. Implementation

The Autofeed: Smart IoT Fish Feeding system integrates an ESP32 microcontroller with a servo motor to automate the feeding process based on schedules that user set. A mobile application, developed using Flutter, provides features for set the feeding times, monitoring food levels in the container, and tracking monthly food usage. The system supports user roles with functionality for admin management and user profile updates, ensuring efficient operation and accessibility.

5.1 System Main Interface

This section will be displaying the design and layout of the system mobile application along with the code for clarification.

5.1.1 User Authentication Interface

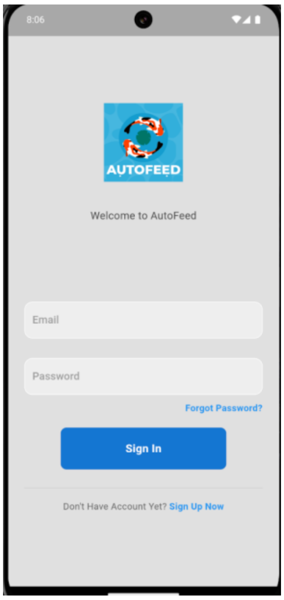


Fig. 8 *Login Interface*

```
class Login extends StatefulWidget {
  final Function()? onTap;
  const Login({super.key, required this.onTap});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  final String adminEmail = "admin@autofeed.com";

  void signInUser() async {
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (context) {
        return const Center(child: CircularProgressIndicator());
      },
    );

    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text.trim(),
      );
```

Fig. 9 *Login Source Code*

5.1.2 Feeding Schedu Interface

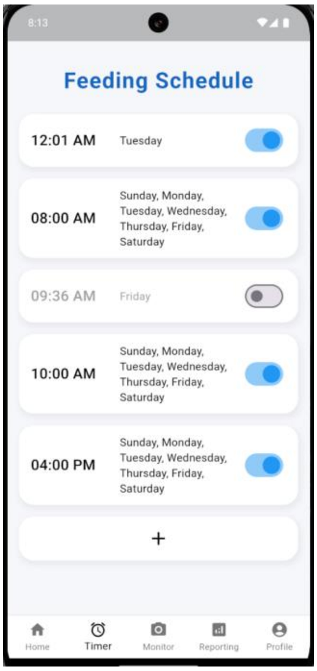


Fig. 10 *Feeding Schedule Interface*

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFF5F7FA),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Title
              const SizedBox(height: 16),
              Center(
                child: Text(
                  'Feeding Schedule',
                  style: TextStyle(
                    fontSize: 30,
                    fontWeight: FontWeight.bold,
                    color: Colors.blue.shade800,
                    letterSpacing: 1.2,
                  ), // TextStyle
                ), // Text
              ), // Center
              const SizedBox(height: 16),
```

Fig. 11 *Feeding Schedule Source Code*

5.1.3 Monitoring Interface

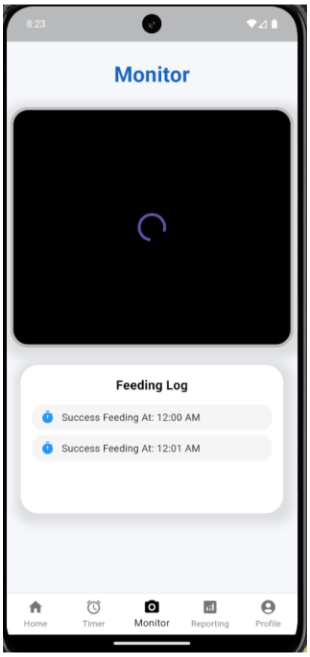


Fig. 12 *Monitoring Interface*

```
// Camera Stream
Container(
  width: 400,
  height: 340,
  decoration: BoxDecoration(
    color: Colors.black,
    borderRadius: BorderRadius.circular(25),
    border: Border.all(color: Colors.grey.shade400, width: 4),
    boxShadow: [
      BoxShadow(
        color: Colors.black.withOpacity(0.1),
        spreadRadius: 10,
        blurRadius: 10,
        offset: const Offset(3, 5),
      ), // BoxShadow
    ],
  ), // BoxDecoration
  child: streamUrl.isEmpty
      ? const Center(child: CircularProgressIndicator())
      : Transform.rotate(
          angle: 3.1416, // 180 degrees
          child: Mjpeg(
            stream: streamUrl,
            isLive: true,
            fit: BoxFit.cover,
          ), // Mjpeg
        ), // Transform.rotate
), // Container
```

Fig. 13 *Monitoring Source Code*

5.1.4 Profile Interface

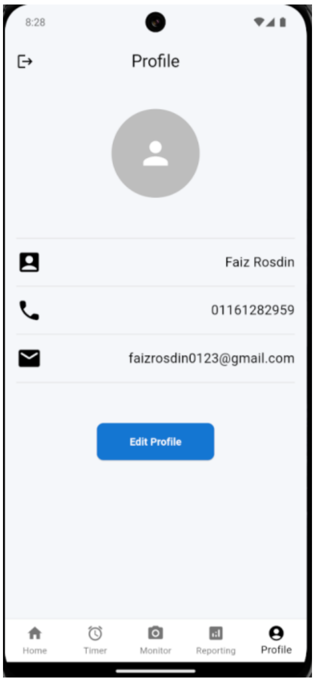


Fig. 14 *Profile Interface*

```
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFF5F7FA),
    appBar: AppBar(
      backgroundColor: const Color(0xFFF5F7FA),
      title: const Text(
        'Profile',
        style: TextStyle(
          color: Colors.black,
          fontSize: 23.0,
          fontWeight: FontWeight.normal,
        ), // TextStyle
      ), // Text
      leading: IconButton(
        onPressed: () {
          showDialog(
            context: context,
            builder: (BuildContext context) {
              return AlertDialog(
                title: const Text('Confirm Logout'),
                content: const Text('Are you sure you want to log out?'),
                actions: [
                  TextButton(
                    onPressed: () {
                      Navigator.of(context).pop();
                    },
                    child: const Text('No'),
                  ), // TextButton
                  TextButton(
                    onPressed: () {
                      signUserOut(context);
                      Navigator.of(context).pop();
                    },
```

Fig. 15 *Profile Source Code*

5.1.5 Report Interface

Fig. 16 *Report Interface*

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFF5F7FA),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Container(
          color: const Color(0xFFF5F7FA),
          child: Column(
            children: [
              Padding(
                padding: const EdgeInsets.all(16.0),
                child: Text(
                  'Report',
                  style: TextStyle(
                    fontSize: 30,
                    fontWeight: FontWeight.bold,
                    color: Colors.blue.shade800,
                    letterSpacing: 1.2,
                  ), // TextStyle
                ), // Text
              ), // Padding
              Padding(
                padding: const EdgeInsets.all(16.0),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
                    const Text(
                      "Select View:",
```

Fig. 17 *Report Source Code*

5.2 System Testing

System testing is the final phase of the software testing where we ensure the completes system is evaluated to meets all the specified requirements. It verifies the functionality, performance, and compatibility of all integrated modules, such as user authentication, feeding schedule, monitoring, profile and reporting. This testing is to identify the problem and issue before the deployment of the system, ensuring the Autofeed: Smart IoT Fish Feeding system works reliably in real-world conditions.

Table 6 *Test Case User Authentication*

User Authentication Module	Description	Expected Results	Actual Results	Pass/Fail
A1-1	Register New User	User data is saved to the database	User successfully registered	Pass
A1-2	Register with Existing Email	System rejects duplicate email registration	Duplicate email detected and registration blocked	Pass
A1-3	Login with Correct Credentials	User is granted access to the system	Login successful	Pass
A1-4	Login with Incorrect Credentials	System displays an error message	Error shown, access denied	Pass
Food Level Detection Module	Description	Expected Results	Actual Results	Pass/Fail
A2-1	Detect Full Container	System displays 100%	Full level shown correctly	Pass
A2-2	Detect Half-Filled Container	System displays 50% level	Food level displayed as 50%	Pass
A2-3	Detect Low Food Level	System alerts user when food is low	Low food notification displayed	Pass
Feeding Schedule Module	Description	Expected Results	Actual Results	Pass/Fail
A3-1	Add Feeding Schedule	Schedule is saved to the database	Schedule saved successfully	Pass
A3-2	Edit Feeding Schedule	Changes are updated in the database	Schedule updated successfully	Pass
A3-3	Delete Feeding Schedule	Schedule is removed from database	Schedule deleted successfully	Pass
A3-4	View Feeding Schedule	List reflects real-time schedule data	Feeding schedule displayed correctly	Pass
Monitoring Module	Description	Expected Results	Actual Results	Pass/Fail
A4-1	Open Camera Feed	Live stream from ESP32-CAM is displayed	Live video feed shown	Pass
A4-2	Resume Camera Feed	Feed reconnects after temporary loss	Video feed restored	Pass
A4-3	Send Stream Video to Database	Video stream link is sent to database	Stream data saved properly	Pass
Profile Module	Description	Expected Results	Actual Results	Pass/Fail
A5-1	View Profile	User info is retrieved from database	Profile information shown	Pass
A5-2	Update Profile	Updated info is saved to the database	Changes saved successfully	Pass
A5-3	Delete Profile/Account	User account is removed from database	Account deleted successfully	Pass

Table 6 (cont)

Report Module	Description	Expected Results	Actual Results	Pass/Fail
A6-1	View Weekly Report	Displays total food usage per week	Weekly usage shown correctly	Pass
A6-2	View Monthly Report	Displays total food usage per month	Monthly usage displayed	Pass
A6-3	View Yearly Report	Shows total usage for the year	Yearly data chart shown	Pass
A6-4	No Data Scenario	Chart is displayed with a flat line or zero values	Flatline chart shown correctly	Pass

5.3 User Acceptance Test

The user acceptance testing for the AutoFeed: Smart Fish Feeding system, as shown in Appendix B, receive highly positive feedback from the 10 respondents. For the setup of feeding schedules, 80% of users reported they were able to do so very easily, while 20% experienced some difficulty. The system's reliability in feeding fish at the scheduled times received a good response which 100% of users stating it always functioned as expected. The food level detection module was stated accurate by 80% of users, while 20% say it slightly inaccurate. Navigating the app interface was also positively received, with 80% rating it very easy and 20% easy. In terms of understanding how to use features such as setting a schedule and checking the food level, 60% of users found everything clear, while 40% reported most features were clear. The user interface was good, with 70% strongly agreeing it was visually appealing, 10% agreeing, and 20% remaining neutral.

Next is the application responsiveness, with 80% rating it very fast, 10% fast, and 10% average. When asked about system stability, 70% of users encountered no issues, while 30% experienced only minor bugs. The food usage reporting feature was found very useful by 80% of users and somewhat useful by 20%. Understanding the report data (weekly, monthly, yearly) was rated very easy by 80% and easy by 20%. Overall satisfaction was high, with 70% of users very satisfied, 20% satisfied, and 10% neutral.

Finally, when asked if they would recommend the system to others, 100% of users responded with a definitely. This feedback indicates a highly functional and user-friendly system that meets the needs of its users effectively.

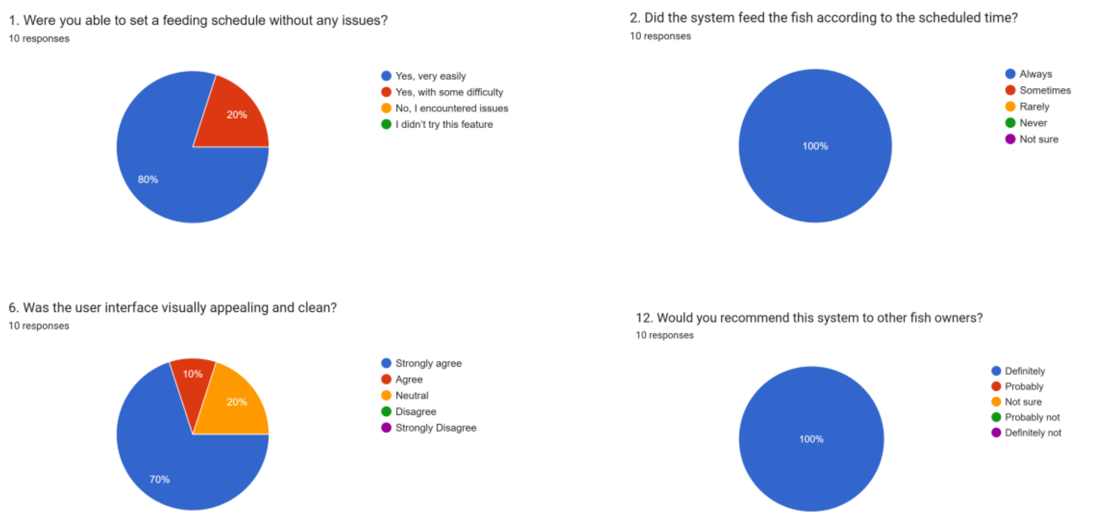


Fig. 18 *Pie Chart Result for System Functionality*

6. Conclusion

The Autofeed: Smart IoT Fish Feeding system's implementation significantly improves user ease and effectiveness in managing fish feeding. The system allows for accurate feeding schedules, real-time food level monitoring, and monthly food usage tracking using IoT technologies and mobile applications. By reducing manual intervention and optimizing resource utilization, these features help to ensure the well-being of fish. The system's modular design and user-friendly interface promote adaptability and scalability, making it suitable for a variety of fish-keeping environments.

Some important findings are highlighted by the Autofeed: Smart IoT Fish Feeding system. The technology efficiently automates feeding schedules, reducing manual feeding and ensuring consistency. While the user-centric design, which includes different admin and user roles, guarantees effective operation targeted to variousof needs, real-time food level monitoring improves resource management. The system is very flexible to different fish-keeping conditions thanks to the integration of IoT technology, which allows for smooth connectivity and control from afar. These results show that the system can effectively handle important issues in fish feeding management and offer reliable methods for maintaining fish health.

Future research could explore advanced IoT integrations to enhance the functionality of the Autofeed system, such as using sensors to detect water quality. Using artificial intelligence or machine learning algorithms for real-time data analysis could provide deeper insights into feeding patterns and optimize feeding schedules based on fish behaviour. Additionally, studying user feedback and behavioural patterns could guide further optimise of the system's interface and usability.

Acknowledgement

The authors would like to thank the Faculty of Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Mohamad Faiz Rosdin, Rozita Abdul Jalil;* ***data collection:*** *Mohamad Faiz Rosdin, Rozita Abdul Jalil;* ***analysis and interpretation of results:*** *Mohamad Faiz Rosdin, Rozita Abdul Jalil;* ***draft manuscript preparation:*** *Mohamad Faiz Rosdin. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Andrian, K. N., Wihadmadyatami, H., Wijayanti, N., Karnati, S., & Haryanto, A. (2024). A comprehensive review of current practices, challenges, and future perspectives in Koi fish (Cyprinus carpio var. koi) cultivation. *Veterinary World*, 1846–1854. https://doi.org/10.14202/vetworld.2024.1846-1854

[2] Mohd, I. I., Azizan, N. H. B., & Elfadil, N. (2020). Design and development of microcontroller-based automatic fish feeder system. College of Engineering, Universiti Malaysia Pahang; College of Computing, Fahad Bin Sultan University.

[3] Chiu, C., Liao, T., Chen, C., & Kao, S. (2022). AIOT Precision Feeding Management System. *Electronics*, *11*(20), 3358. https://doi.org/10.3390/electronics11203358

[4] Motlagh, N. H., Mohammadrezaei, M., Hunt, J., & Zakeri, B. (2020). Internet of things (IoT) and the energy sector. *Energies*, *13*(2), 494. https://doi.org/10.3390/en13020494

[5] Gowda, V. D., Sridhara, S. B., Naveen, K. B., Ramesha, M., & Pai, G. N. (n.d.). INTERNET OF THINGS: INTERNET REVOLUTION, IMPACT, TECHNOLOGY ROAD MAP AND FEATURES. *Advances in Mathematics: Scientific Journal*, *9*. https://doi.org/10.37418/amsj.9.7.11

[6] Saipul, N., Puriza, M. Y., Asmar, N., Rachmani, A. A., Satria, H., Anzari, Y., & Yandi, W. (2023a). Green energy powered autonomous IoT catfish feeder prototype based on Arduino Mega and ESP-8266. *IOP Conference Series Earth and Environmental Science*, *1267*(1), 012037. https://doi.org/10.1088/1755-1315/1267/1/012037

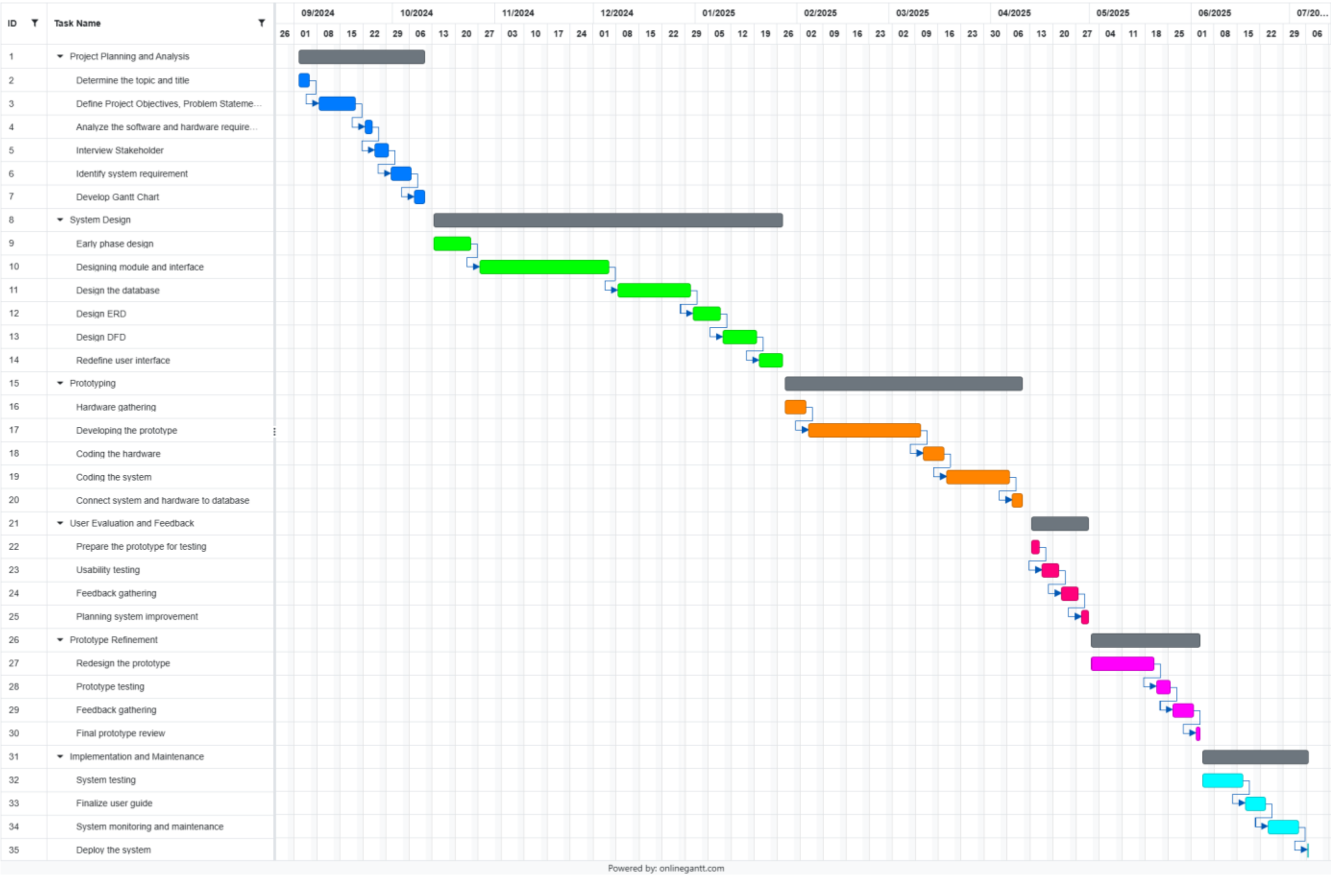
[7] Gunova, V., & Idris, M. (2023b, November 30). *Rancang bangun automatic Fish Feeder berbasis Internet of things (IoT)*. https://abecindonesia.org/proceeding/index.php/abec/article/view/385

[8] Kassim, M., Zulkifli, M. Z., Ya'acob, N., & Shahbudin, S. (2021c). IoT system on dynamic fish feeder based on fish existence for agriculture aquaponic breeders. *Baghdad Science Journal*, *18*(4(Suppl.)), 1448. https://doi.org/10.21123/bsj.2021.18.4(suppl.). 1448

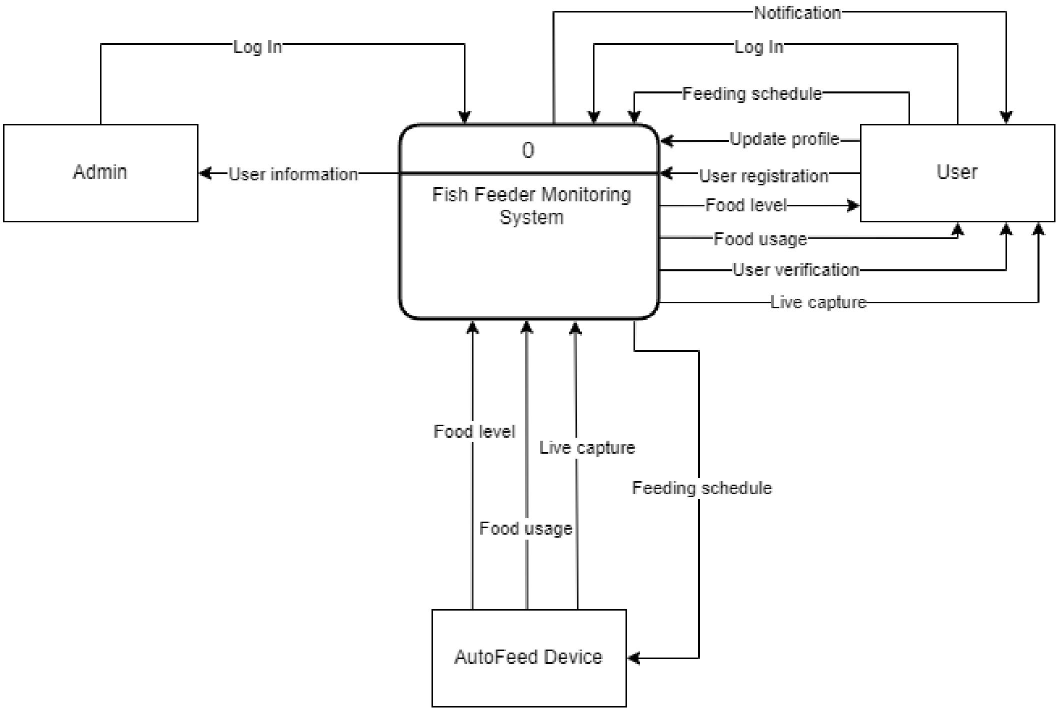
[9] Chahid, A., N'Doye, I., Majoris, J. E., Berumen, M. L., & Laleg-Kirati, T. (2021). Fish growth Trajectory tracking via reinforcement learning in precision aquaculture. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.07251

[10] Mu, X., & Antwi-Afari, M. F. (2023). The applications of Internet of Things (IoT) in industrial management: a science mapping review. *International Journal of Production Research*, *62*(5), 1928–1952. https://doi.org/10.1080/00207543.2023.2290229

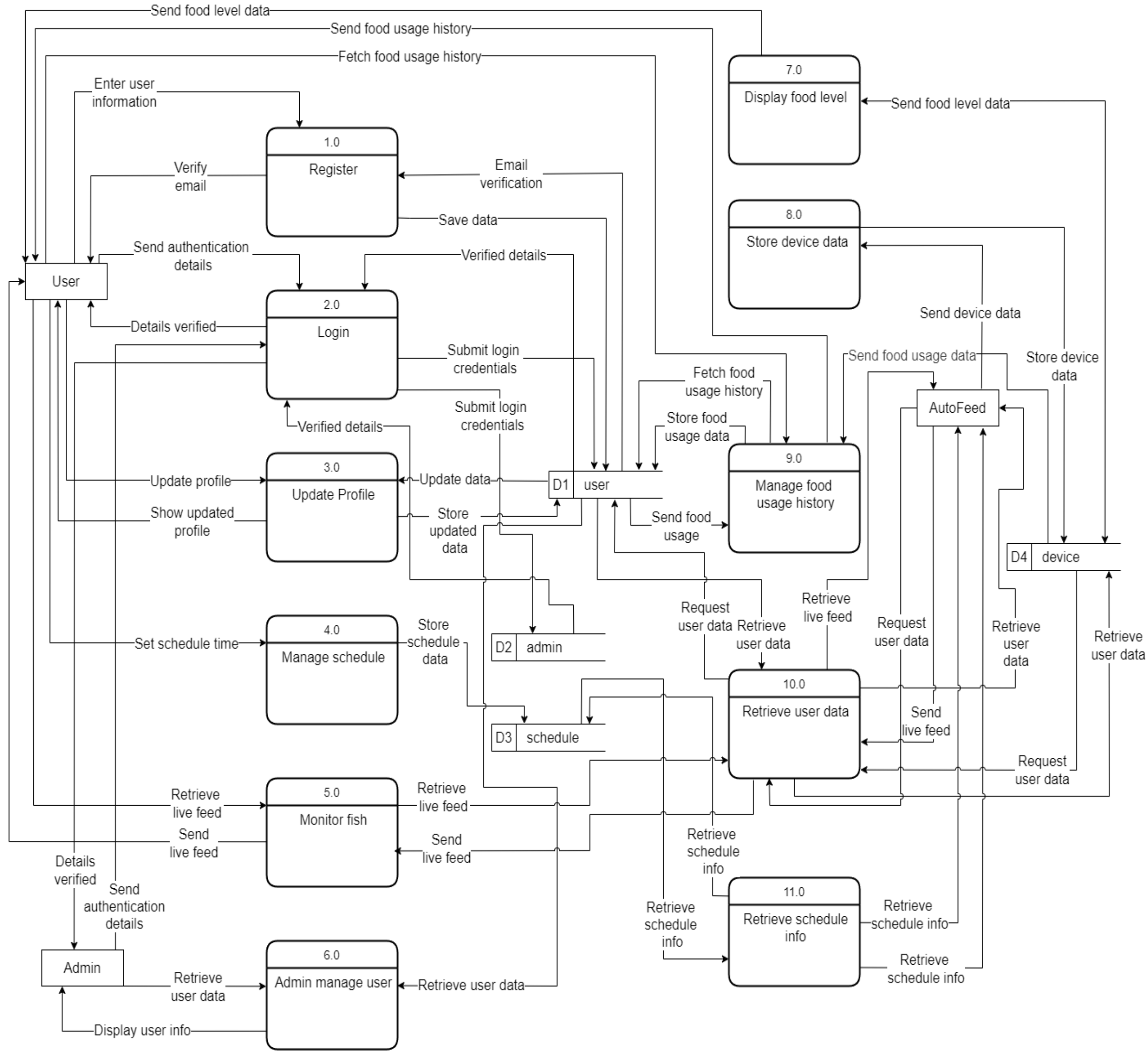
Appendix A: Gantt Chart



Appendix B: Context Diagram

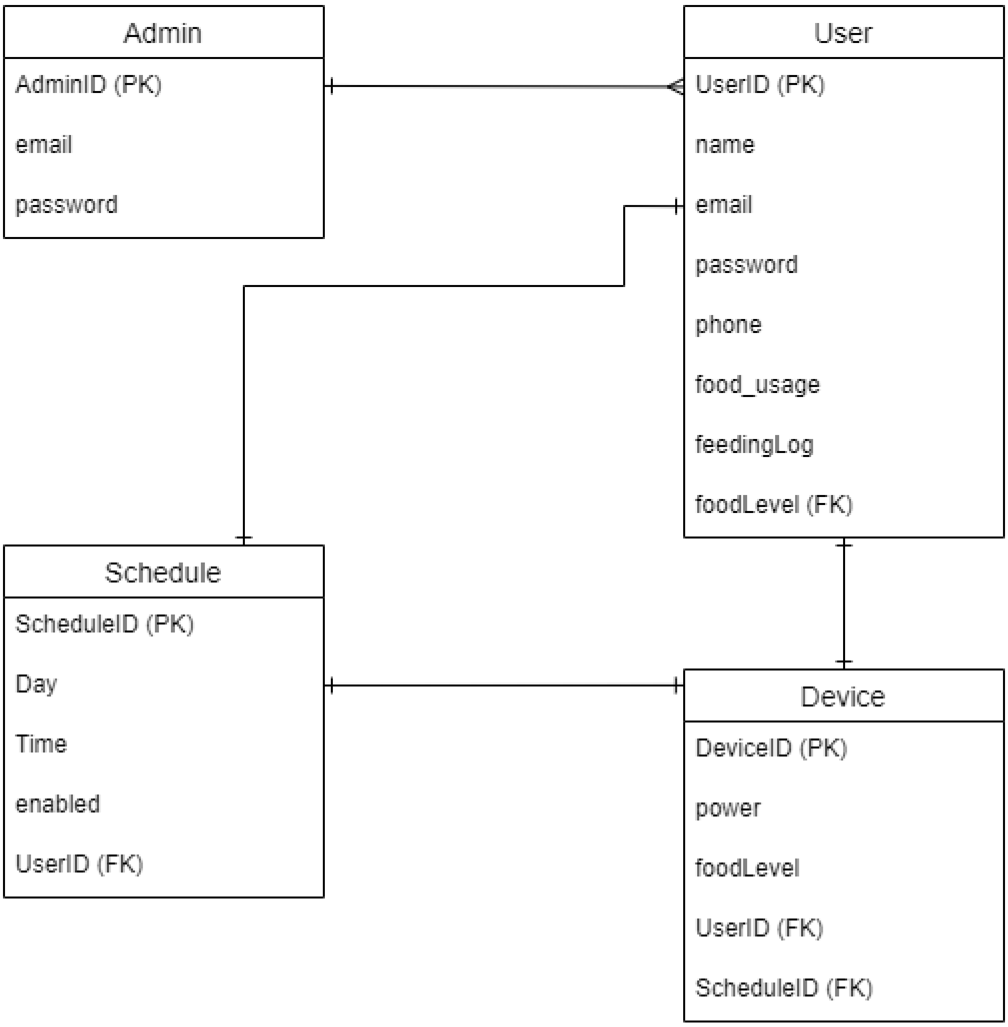


Appendix C: Data Flow Diagram Level 0



Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1510-1528

Appendix D: Entity Relationship Diagram



Appendix E: Flowchart