PUSL3190 Computing Individual Project

Final Report

A Sensor-Based System for Advanced Beehive Monitoring that utilizes Signal Processing with Artificial Intelligence

Supervisor: Ms. Hirushi Dilpriya

Name: Nisal Liyanage

Plymouth Index Number: 10898808

Degree Program: BSC (HONS) Computer Science

Acknowledgment

I would like to express my deepest gratitude to everyone who has contributed to the development of this project, "A Sensor-Based System for Advanced Beehive Monitoring that Utilizes Signal Processing and Artificial Intelligence”.

First and foremost, I am sincerely grateful to my supervisor, Ms. Hirushi Dilpriya, for her invaluable guidance, support, and mentorship throughout the course of this project. Her expertise and insights have played a crucial role in shaping the direction and success of this work. I am also especially thankful to Mr. Chandasiri from the Engineering Faculty, who has been the backbone with the implementation of hardware and electronic components. His technical knowledge and assistance were essential to bringing the system to life.

Additionally, I would like to extend my heartfelt thanks to Mr. Vimukthi, the director at the Bandarawela Bidunuwewa Bee Unit, as well as the fellow researchers and beekeepers at the unit, for their cooperation and support in various aspects of the project.

Finally, I am deeply grateful to Mr. Tharanga, a beekeeper in Horana, for his help in acquiring equipment and resources related to beekeeping, including beehives and other necessary materials.

Abstract

Bees are nature's little warriors who contribute immensely to sustaining life here on earth and to maintaining the balance of our ecosystem. However, in recent years, bee populations around the world, including in Sri Lanka, have faced significant declines in health and numbers. This global decline poses a major threat to ecosystems and agricultural sustainability. Continuous monitoring of beehives is therefore essential to ensure the health and survival of bee colonies. Unfortunately, in Sri Lanka, beekeepers currently monitor their hives manually, which presents several challenges for both the bees and the beekeepers. Traditional beekeeping practices often lack efficient monitoring systems, making hive management and health assessments difficult. Moreover, manual inspections can disturb the bees, causing them stress and potentially affecting productivity. To address these issues, a non-intrusive, sensor-based, computerized remote hive monitoring system has been developed**,** incorporating IoT technology, signal processing, and AI/ML-based techniques. The system integrates various sensors to collect real-time data on both hive internal and external temperatures and humidity levels, as well as hive weight. The main deliverable of the developed system is a user-friendly mobile application designed to empower beekeepers. Key features include offering real time hive sensor data, detecting and notifying beekeepers of the optimal time to harvest the hive, detecting the queen's presence or absence by analyzing bee sounds, providing indications of hive health status, and logging and tracking current and past queen details for each hive. Through the implementation of this solution, the author aims to elevate traditional Sri Lankan beekeeping practices to the next level by enabling remote, real-time hive monitoring, significantly reducing the need for manual inspections. This not only enhances operational efficiency for beekeepers but also minimizes disturbances to the bees, contributing to their well-being and productivity. (Word Count: 287)

Keywords— Beekeeping in Sri Lanka, Advanced beehive monitoring, Smart beehives, Hive health monitoring, Queen detection

Table of Contents

Chapter 01 | Introduction **11**
1.1 Overview ***11***
1.2 Problem Definition ***11***
1.3 Solution ***12***
1.4 Project Aim ***12***
1.5 Scope ***12***
1.6 Project Objectives ***13***
1.7 Project Deliverables ***14***
1.8 Structure of the Report ***14***
Chapter 02 | Background Study **15**
2.1 Literature Review ***15***
2.2 Limitations of the Existing Systems ***18***
Chapter 03 | Methodology **20**
3.1 Overview ***20***
3.2 Project Planning ***21***
3.2.1 Feasibility Analysis 21
3.2.2 Work Breakdown Structure 24
3.2.3 Risk Analysis 24
3.2.4 Stakeholder Analysis 26
3.2.5 Time Plan 26
3.3 Requirements Analysis ***29***
3.3.1 Requirements Gathering Techniques 29
3.3.2 Requirements Specification 34
3.3.3 Use Case Analysis. 35
3.3.4 Personas Development 38
3.3.5 Requirements Prioritization 39
3.4 Design ***41***
3.4.1 Circuit Diagram 41
3.4.2 Use Case Diagram 42
3.4.3 Storyboards 43
3.4.4 Class Diagram 44
3.4.5 Database Diagrams. 45
3.4.6 High Level Architecture Diagram 47
3.4.7 Wireflow Diagram (Paper-Based) 48
3.4.8 Level 0 DFD Diagram 49
3.4.9 UI Designing. 50
3.5 Implementation ***54***

3.5.1 Overview 54
3.5.2 Hardware Setup 55
3.5.3 Firmware Integration 60
3.5.4 Cloud Setup 63
3.5.5 Queen Detection Model 64
3.5.6 Health Analysis Model 78
3.5.7 Backend Server 90
3.5.8 Beekeeper Mobile App 92
3.5.9 Admin Web App 97
***3.6 Testing* 100**
3.6.1 Unit Testing 100
3.6.2 Integration Testing 101
3.6.3 System Testing 102
3.6.4 User Acceptance Testing 103
Chapter 04 | Evaluation & Reflections on Project Outcomes 104
***4.1 Academic Findings & Research outcomes* 104**
***4.2 End-Project Assessment* 105**
***4.3 Post-Mortem Analysis* 109**
Chapter 05 | Conclusion 111
***5.1 Limitations* 111**
***5.2 Future Work* 112**
***5.3 Final Remarks* 114**
References 115
Appendix 120
***Appendix A: Scientific and Biological Context* 120**
***Appendix B: User Guide* 124**
B.1 Minimum Platform Specifications 124
B.2 The normal real-world workflow of the system 124
B.3 Installation & Setup 125
B.4 How to use the beekeeper mobile application 127
B.6 How to use the admin web application 132
***Appendix C: Code snippets* 137**
***Appendix D: Use Cases Continuation* 142**
***Appendix E: Unit Test Cases* 147**
***Appendix F: Integration Test Cases* 152**
***Appendix G: Feedback Form & Results* 157**
***Appendix H: Supporting Project Documents* 166**
H.1 Project Proposal 166

H.2 PID 179
H.3 Interim 197
Appendix I: Records of supervisory meetings ***226***

List of Figures

Figure 1: Graph showing the number of publication articles that utilize each type of sensor in beehive monitoring (Last Update: 2023). Reprinted from (Dorin, 2023). 15
Figure 2: World Map showing the distribution of the previously done research publications on smart beehives. (Last update: 2023). Reprinted from (Dorin, 2023) 16
Figure 3: Graph showing the number of research publications that focus on each bee species. (Last Update: 2023). Reprinted from (Dorin, 2023) 17
Figure 4: Tasks carried out in each stage in the Waterfall methodology 20
Figure 5: Work Breakdown Structure 24
Figure 6: Gannt Chart 27
Figure 7: Photos taken during the field visit to the Bandarawela Bidunuwewa Bee Unit 29
Figure 8: During the visit to an apiary in Horana 30
Figure 9: During the zoom interview with the director at Bandarawela Bidunuwewa Bee Unit 30
Figure 10: Questions that were asked and the answers that were received during the zoom interview 33
Figure 11: Hardware Requirements 35
Figure 12: Software Requirements 35
Figure 13: Persona for Mr. Vimkuthi 38
Figure 14: Persona for Mr. Tharanga 39
Figure 15: Circuit Diagram 41
Figure 16: Use Case Diagram 42
Figure 17: Storyboard for using the app to detect whether the hive is queenless or queenright 43
Figure 18: Storyboard for using the app to view live sensor data from a particular hive 43
Figure 19: Class Diagram 44
Figure 20: Database design diagram for the firestore database 45
Figure 21: Database design diagram for the firebase realtime database 46
Figure 22: High level architecture diagram 47
Figure 23: Paper based wireflow diagram 48
Figure 24: Level 0 DFD 49
Figure 25: Figma UI designs for the mobile app 53
Figure 26: DHT22 sensor 1 55
Figure 27: DHT22 sensor 2 55
Figure 28: Load cell mounted on a wooden frame 56
Figure 29: The ESP2 used in the project placed to the circuit 56
Figure 30: The prototype of the IOT module 58
Figure 31: Implemented IOT prototype placed to a beehive in Rajagiriya 58
Figure 32: The implemented final IOT product 59
Figure 33: The final IOT product placed on a beehive in Athurugiriya 59
Figure 34: The final IOT product placed on a beehive in Katana 59
Figure 35: The programming language, development environment and version control used to implement and test the firmware 60
Figure 36: The data processing pipeline for temperature and humidity data 61
Figure 37: The data processing pipeline for weight data 61
Figure 38: The programming languages, development environments and other tools utilized for the queen model implementation 64
Figure 39: The machine learning pipeline implemented for the queen detection model 65
Figure 40: The Python libraries and frameworks used for various tasks in the queen model implementation 65
Figure 41: Data preprocessing steps for the queen model 68
Figure 42: The CNN architecture 72

Figure 43: The overall confusion matrix across all the folds for CNN_4 experiment..... 76
Figure 44: The average learning curves across all the folds for the CNN_3 scenario..... 77
Figure 45: The programming languages, development environments, and other tools used for implementing the health model..... 78
Figure 46: The machine learning pipeline implemented for the hive health analysis model 78
Figure 47: The Python libraries and frameworks used for various tasks in the hive health model implementation. 79
Figure 48: Data preprocessing steps carried out for the health model..... 80
Figure 49: How inside and outside humidities have changed with time in the Rajagiriya beehive .. 81
Figure 50: How inside and outside temperatures have changed with time in the Rajagiriya beehive81
Figure 51: The average inside and outside temperatures for each hour of the day in the Athurugiriya beehive..... 82
Figure 52: The average inside and outside humidities for each hour of the day in the Athurugiriya beehive..... 82
Figure 53: The variation of inside and outside temperatures across 2 weeks of time in the Athurugiriya beehive 83
Figure 54: The variation of the temperature differential across 2 weeks of time in the Athurugiriya beehive..... 84
Figure 55: The variation of inside and outside humidities across 2 weeks of time in the Athurugiriya beehive..... 85
Figure 56: The variation of the humidity differential across 2 weeks of time in the Athurugiriya beehive..... 86
Figure 57: The anomaly scores for each data point in both the training and testing healthy hive samples 88
Figure 58: The synthetic data points and their anomaly scores..... 89
Figure 59: The framework, development environments, and other tools used to implement the backend server 90
Figure 60: The workflow of the /predict_queen endpoint 91
Figure 61: The workflow of the /analyze_health endpoint..... 91
Figure 62: The programming language, framework, development environments, and other tools used to implement the beekeeper mobile application 92
Figure 63: The programming language, framework, development environments, and other tools used to implement the admin web application..... 97
Figure 64: Future Enhancements..... 113

List of Tables

Table 1: The main limitations found in past research studies and the solutions for those identified limitations given through the developed solution. 18
Table 2: Expenses for the sensors and microcontroller 22
Table 3: Expenses for power setup, wiring and connections of the IOT device 22
Table 4: Expenses for the hive setup and beekeeping training 23
Table 5: Risk Analysis Matrix 24
Table 6: Skateholder Analysis 26
Table 7: Time Plan 26
Table 8: Use Case Analysis for login 35
Table 9: Use Case analysis for viewing live sensor data 36
Table 10: Use Case analysis for detecting the queen's presence 37
Table 11: Use Case analysis for monitoring hive's health 37
Table 12: Requirements Prioritization Table 39
Table 13: The 8 tasks carried out in the implementation process 54
Table 14: Summary of the used sensors 55
Table 15: Summary of the components used in the power supply 57
Table 16: Summary of the Arduino libraries used in the firmware 62
Table 17: Summary of the used firebase components 63
Table 18: The summary of the two projects where the dataset gained its beehive sound recordings from 66
Table 19: The distribution of "queen present" and "no queen" samples for each hive 67
Table 20: Summary of the extracted features 68
Table 21: The experiments carried out with SVM 70
Table 22: The experiments carried out with Extreme Gradient Boosting (XGBoost) 71
Table 23: Tuned hyperparameters for the XGBoost classifier 71
Table 24: The experiments carried out with Convolutional Neural Networks (CNN) 71
Table 25: The tuned hyperparameters in CNN 72
Table 26: Evaluation Metrices 73
Table 27: The precision, recall, and F1-score equations calculated separately for each class 74
Table 28: The evaluation results obtained from various ML model training 75
Table 29: Summary of the newly introduced features 80
Table 30: The values set for the hyperparameters for the One Class SVM model training 87
Table 31: Overview of the decision_function() and predict() methods 88
Table 32: The Dart packages used in the beekeeper mobile application 92
Table 33: Mapping mobile interfaces with requirements 93
Table 34: Mapping web interfaces with requirements 98
Table 35: Summary of unit testing performed 100
Table 36: Summary of integration testing performed 101
Table 37: A critical analysis of the extent to which each initial objective was achieved by the end of the project. 106
Table 38: Evaluation on how well the project performed compared to the initial plan 107
Table 39: Various changes made to the original project plan, the reasons for these changes, and their impact 108

List of Equations

Equation 1: Mathematical expression to determine how many segments each audio file should be divided into to obtain 1-second-long samples. 68
Equation 2: Mathematical expression to determine the time interval for each interval 68
Equation 3: The equation for the precision (weighted average)..... 73
Equation 4: The equation for the recall (weighted average) 73
Equation 5: The equation for the f1-score (weighted average) 73
Equation 6: The equation for accuracy 73
Equation 7: The equation for precision 74
Equation 8: The equation for recall..... 74
Equation 9: The equation for f1-score..... 74

Chapter 01 | Introduction

1.1 Overview

Bees play a crucial role in sustaining life on Earth and maintaining the balance of our ecosystem (Khan, 2023). Albert Einstein emphasized this significance with his famous quote: “If the bee disappeared off the surface of the globe, man would only have four years of life left.” Bees are the largest pollinators on the planet, contributing to around 33% of the global food supply (Shaden A. M. Khalifa, 2021). They help fertilize and cross-pollinate many plant species, which is essential for the reproduction of various plants, including wildflowers, fruits, vegetables, and nuts like almonds (Christina T. Liang, 2016). Without bees, food security for both humans and animals would be at risk (Vidushi Patel, 2021).

Besides playing an important ecological role, bees also produce some important resources humans exploit in apiculture. Perhaps the most popular of all bee products is honey; though being a delicious and healthy food ingredient, it is also successfully applied in many medicinal and therapeutic fields (E.R.H.S.S. Ediriweera, 2012). Apart from honey, various other products such as beeswax, royal jelly, bee pollen, and propolis are cultivated in beekeeping for cosmetic and medical use (Isayas Asefa Kebede, 2024).

The beekeeping industry in Sri Lanka is conducted on a very small scale, often limited to a handful of beekeepers and bee hobbyists. Nevertheless, beekeeping represents a thriving sector within Sri Lanka's agricultural landscape, offering significant potential for both economic growth and environmental stewardship. This industry, with its plenty of benefits, can contribute substantially to the nation's development.

1.2 Problem Definition

However, in recent years, bee populations around the world as well as in Sri Lanka have faced significant declines in health and numbers. This is evident through reduced honey yields, smaller colony sizes, higher mortality rates among worker bees, infertility in queen bees, and the occurrence of colony collapse disorder. Many factors contribute to this decline, including air pollution, widespread malnutrition, harmful pesticides, deforestation, and other human activities. In the beekeeping industry, poor monitoring practices can further harm bee populations (Peter Hristov, 2020).

Therefore, continuous monitoring of beehives is essential to ensure the health of the colony. This includes checking for pests or diseases, confirming the queen bee's presence, and ensuring there is enough honey for extraction. Unfortunately, in Sri Lanka currently beekeepers monitor their hives manually, which causes several challenges for the beekeepers and bees.

Regular manual monitoring of hives is often time-consuming, inefficient, and risky, as beekeepers may get stung while inspecting hives (Yiyao Zheng, 2024). Additionally, many beekeepers overlook regular hive monitoring due to their busy schedules (Braga, 2021). Traditional manual monitoring can also make it difficult to gather reliable information about the health of the bee colony and often ignore important environmental factors like temperature and humidity (Pikri, 2022). As a result, the quality and quantity of honey can suffer due to insufficient monitoring of hive conditions, and these manual methods can be intrusive, causing stress to the bees (Braga, 2021), (Yiyao Zheng, 2024).

1.3 Solution

Therefore, there is a pressing need for non-intrusive methods to monitor beehives remotely and accurately, benefiting both the beekeepers and the bees. In order to provide a non-intrusive solution for monitoring beehives, a computerized hive monitoring system has been developed, incorporating IoT technology, signal processing, and AI/ML-based approaches.

1.4 Project Aim

The aim of this project is to empower Sri Lankan beekeepers with a low-cost and user-friendly mobile application enabling real-time informed decision-making for optimal hive management and the overall wellbeing of the colony.

1.5 Scope

The current scope of the implemented system encompasses the IoT module, cloud database integration, a mobile application for beekeepers, a queen detection model, a hive health monitoring model, and a web interface for administrators to manage user and hive details. The system currently utilizes three sensors to measure hive weight, internal temperature and humidity, and external temperature and humidity.

The IoT module is designed to be energy-efficient and independent, powered by lithium-ion batteries recharged via a solar panel during daytime. The implemented mobile application supports real-time hive monitoring, historical data visualization, optimal harvest time estimation, detection of the presence or absence of the queen bee, management of current and past queen details, and hive health analysis.

The web application allows administrators to manage beekeeper, hive, and apiary records.

Future enhancements not yet implemented include early swarming detection methods and more advanced algorithms for hive health analysis. Currently, the system does not integrate microphones to directly detect queen presence through hive sound. However, the mobile app allows beekeepers to manually record hive sounds using their phones placed near the hive. Future versions may incorporate onboard microphones for automated sound detection to detect the presence of the bee queen.

1.6 Project Objectives

With respect to the developed sensor-based system for advanced beehive monitoring:

- **Main Objective**

    1. Provide beekeepers and the bee researchers with a mobile application to remotely monitor their beehives.

- **Specific Objectives**

    1. Track the normal honeybee activities and hive conditions. (e.g., temperature variations during the day).
    2. Visualize real-time data from the hive.
    3. Offer insights into historical trends of the hive (e.g.: How the weight of the hive has changed overtime).
    4. Deliver insights on hive health by detecting anomalies in normal bee activities and hive conditions.
    5. Provide insights into the optimal time for harvesting.
    6. Detect the presence of the queen bee automatically using acoustic signals.
    7. Maintain a record of the queen bees' history of the hive.

1.7 Project Deliverables

The implemented project consists of three main deliverables:

1. Hardware Setup – includes the IoT module and the power supply.
2. Mobile Application – designed for beekeepers to monitor hive data.
3. Web Application – for the admin to manage users, apiaries, and hives.

1.8 Structure of the Report

This chapter introduces the developed sensor-based system for advanced beehive monitoring. Chapter 02 is dedicated to literature study. Chapter 03 delves into discussion about the method of approach highlighting how the project has been progressed through planning, requirements gathering and analysis, design, implementation and testing phases. Chapter 04 focuses on the evaluation and reflection on project outcomes, while chapter 05 focuses on limitations and future work. The report ends with the references list followed by the appendix.

Chapter 02 | Background Study

2.1 Literature Review

Human beekeeping has roots that go all the way back to ancient Egypt and ancient Greece (Whitfield, 1956). As the consumption of bee products such as honey and wax rises, the need to monitor beehives also increases over time. This is evident in Aristotle's involvement in beehive monitoring in the 4th century BCE (Carlson, 2015).

Until the end of the 20th century, beehive monitoring was primarily conducted using manual methods. However, with advancements in technology, sensor-based systems have become a common area of research in beehive monitoring (Dorin, 2023).

Various sensors have been used in beehive monitoring to track different internal hive parameters, such as temperature (mostly DHT22 sensors were used), humidity (mostly DHT22 sensors were used), CO2 concentration (Gas sensors and MG811 sensor were used) , sound (mostly electret microphones were used) , vibrations (various vibration sensors were used), and the overall weight of the hive (load cells were used) (Cecchi, 2020), (Dorin, 2023).

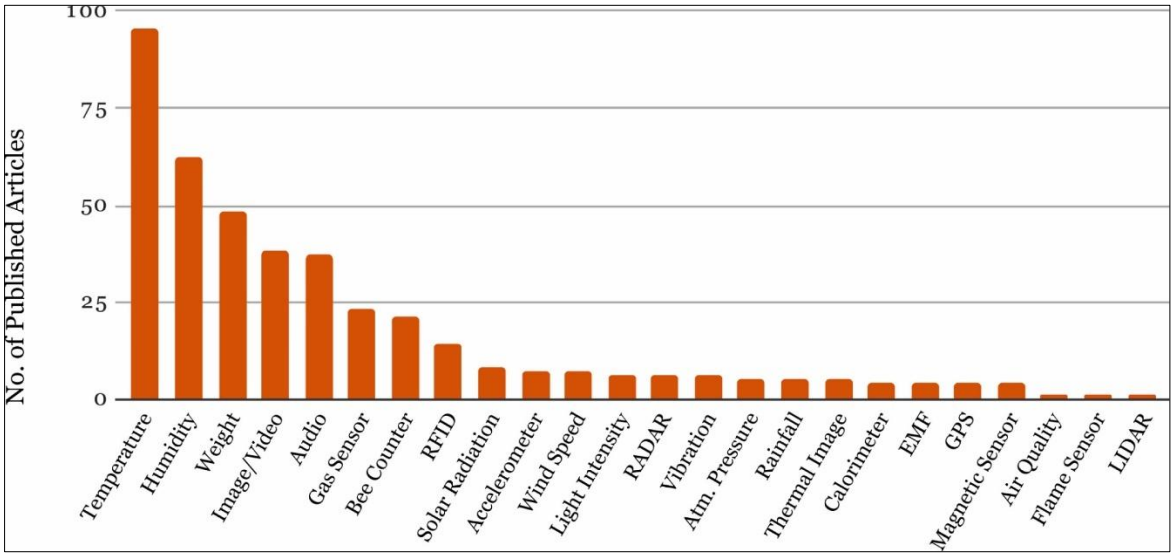


Figure 1: Graph showing the number of publication articles that utilize each type of sensor in beehive monitoring (Last Update: 2023). Reprinted from (Dorin, 2023).

Most of the research on sensor-based beehive monitoring has been conducted in North America, Europe, China, Africa, Brazil, and Australia. There is a noticeable lack of research

conducted in Sri Lanka and many other South Asian countries (Dorin, 2023). Even in India, there are very few papers on sensor-based beehive monitoring, which may be due to the relatively smaller size of the beekeeping industry and limited resource availability in these regions.

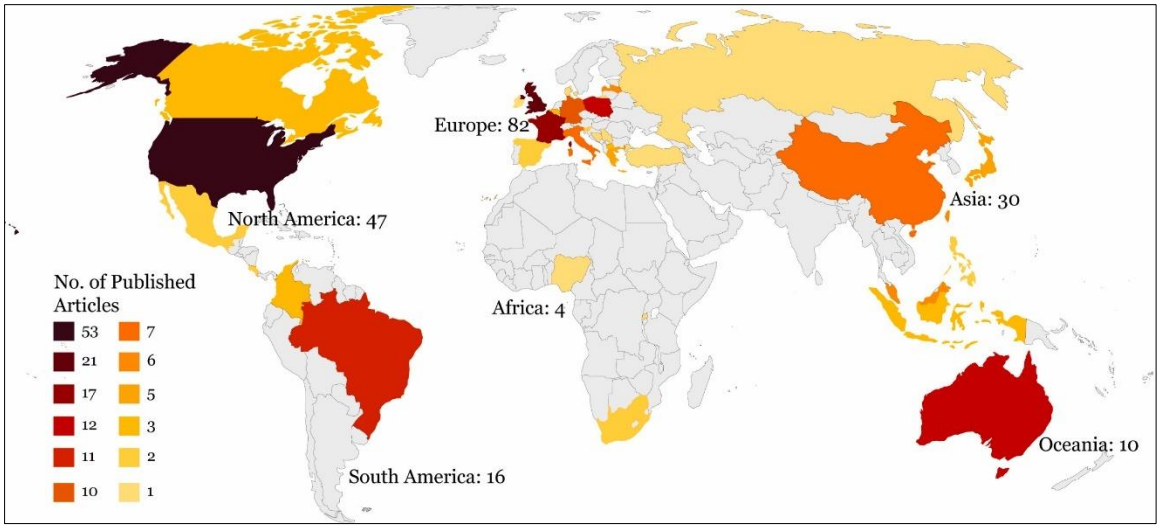


Figure 2: World Map showing the distribution of the previously done research publications on smart beehives. (Last update: 2023). Reprinted from (Dorin, 2023)

According to (Dorin, 2023), most of the previous research done on smart beehives largely focused on European honeybees, followed by stingless bees and bumble bees. There is a significant lack of focus given to the Asian honeybees in the past research. But in this study the focus was given to the Asian honeybees, as this is the most commonly found honeybee species in Sri Lanka.

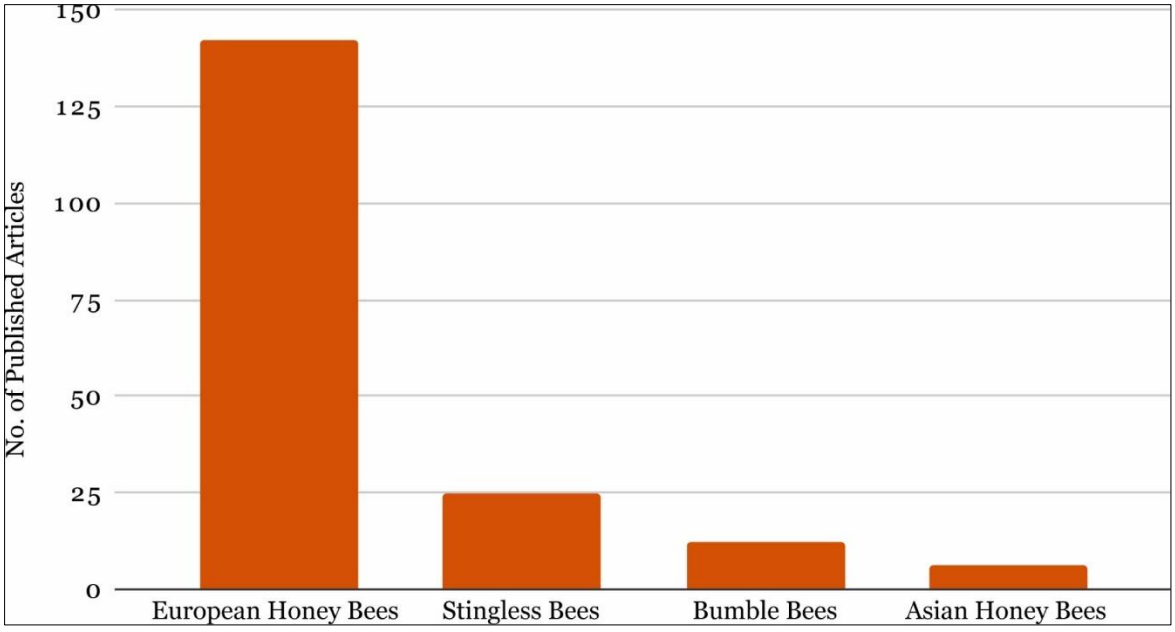


Figure 3: Graph showing the number of research publications that focus on each bee species. (Last Update: 2023). Reprinted from (Dorin, 2023)

Several studies (Zacepins, 2017), (Seritan, 2018), (Bayir, 2016), (Cecchi, 2020), (Rector, 2008) have utilized load cells for hive weight measurement, which can be a direct indicator of honey production and colony size. However, most research has focused primarily on analyzing seasonal patterns in hive weight changes. One study analyzed daily weight variations in beehives and found a significant difference in daily weight changes during the winter and summer months (Cecchi, 2020). This study will focus on utilizing load cell data to determine if the hive is ready to harvest or not, which is by far the most needed requirement of the beekeepers.

With the use of sensor-based technology to monitor beehives, beehive health monitoring has become a major area of interest among various researchers (Cazier}, 2020), (Dorin, 2023). In a healthy hive, bees tend to regulate the internal temperature and humidity for brood production and development (Kridi, 2016). Any anomaly in the internal temperature and humidity patterns may indicate a possible issue within the colony (Hegde, 2023). One study (Weiss, 2017) implemented a system to monitor the beehive colony health using hive inside temperature. Furthermore, one study (Edwards-Murphy, 2016) utilized decision trees for beehive health monitoring using hive-inside relative humidity, temperature, carbon dioxide and oxygen levels. Hive weight measurement is also used to gain insights into hive health (Rector, 2008), (Dorin, 2023). Some research tracks the activity at the hive entrance as a method to monitor hive health (Andrijević, 2022).

The queen bee is essential for the colony's survival. The queen must be healthy and fertile to ensure reproduction and the overall survival of the colony. Sometimes the queen bee may be infertile, unhealthy, dead, or attacked by intruders or pests. Beekeepers manually check hive frames to detect eggs or larvae to confirm the queen's presence. If eggs and larvae exist, it indicates that the queen bee is present (Cejrowski, 2018).

This process is time-consuming and stressful to the bees. To address this, various studies (Cejrowski, 2018), (Orlowska, 2022), (Uthoff, 2023), (Abdollahi, 2022), (Doinea, et al., 2024) have explored detecting the queen's presence or absence using sensor technology and AI. Most research focuses on acoustic signals emitted by bees, which change when the queen bee is absent. (Uthoff, 2023). The major issue when creating a classifier for detecting queen presence/absence is the lack of generalization when tested on other hives (Cejrowski, 2018). As a result, a more generalized model for queen detection needs to be developed. This study will focus on creating a robust and generalized model to detect the presence of the queen bee using online audio datasets and audio samples from a local hive.

Previous research lacks the utilization of mobile applications for beehive monitoring, except for a few research studies with limited features (Branco, 2023), (Pikri, 2022). In this study, a mobile application will be developed to monitor live data in the hive, environmental conditions outside the hive, hive health status, and queen detection. It will also provide features to track the queen bee's history.

2.2 Limitations of the Existing Systems

Table 1: The main limitations found in past research studies and the solutions for those identified limitations given through the developed solution.

Existing System	Main Limitation	Solution
(Cejrowski, 2018),	The bee queen detection model's **limited ability to generalize across different hives**.	The bee queen detection model is trained and tested using a dataset that contains data from multiple hives.
(Orlowska, 2022)	The model is not integrated into a real-world application	The developed queen detection model is

		integrated with a flutter mobile application.
(Branco, 2023)	The developed hardware prototype was only tested in a natural laboratory, and not with an actual real-world beehive.	The developed hardware prototype was placed and tested on two different real-world beehives at different places
(Braga, 2021)	The method used to analyze hive health relies on cameras and image processing techniques, making it a relatively expensive approach.	Using only cheap temperature and humidity (DHT22) sensor to analyze the colony's health.
(Mansor, 2022), (Branco, 2023)	The developed mobile application has only specific and limited functionalities.	The developed mobile application provides beekeepers with a variety of features, including apiary and hive management, viewing live and historical data, determining the optimal harvesting time, gaining insights into hive conditions, and managing queen details

Chapter 03 | Methodology

3.1 Overview

The project follows the waterfall methodology, which ensures a structured and sequential approach to development. The following diagram illustrates the various tasks that were carried out at each stage of this project using the Waterfall methodology.

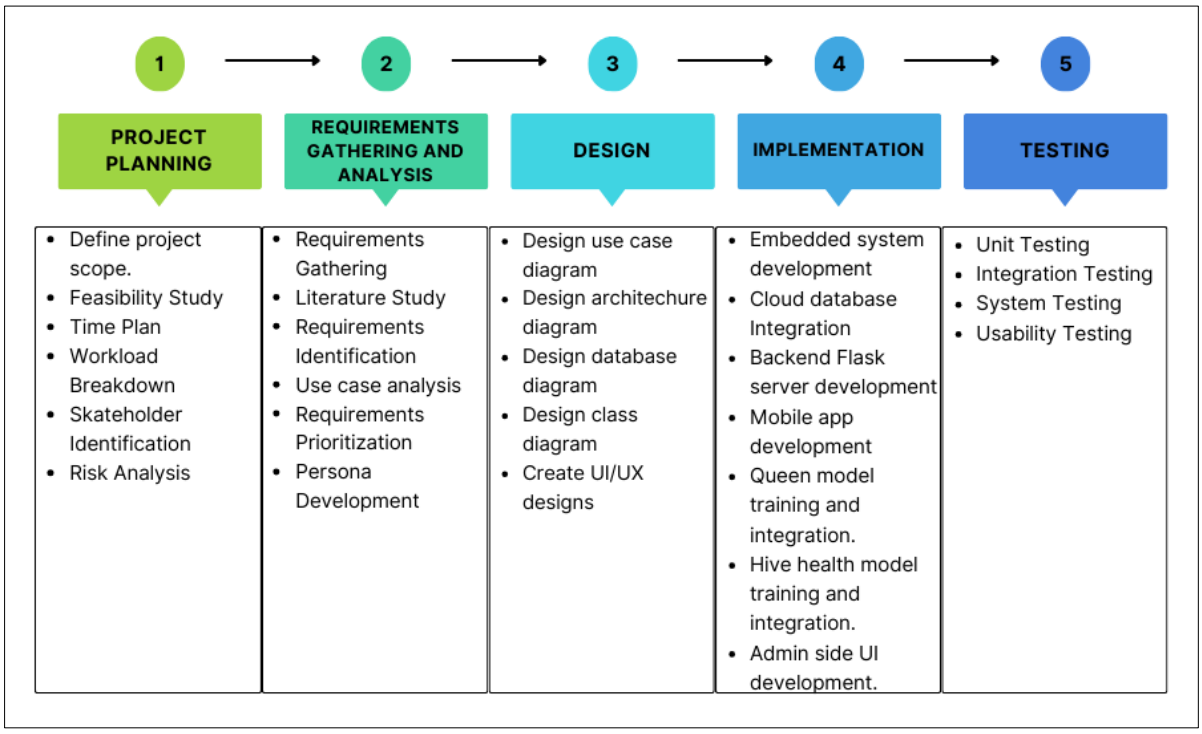


Figure 4: Tasks carried out in each stage in the Waterfall methodology

3.2 Project Planning

3.2.1 Feasibility Analysis

3.2.1.1 Operational Feasibility

Under operational feasibility it is accessed whether the system is able to survive in the real-world and is able to integrate into existing beekeeping practices in Sri Lanka. The final delivery is a simple, user-friendly mobile application, which the beekeepers can easily use without requiring much advanced technical skills. The existing approach to monitor the beehives is inspecting beehives manually through naked eye, which is time consuming. The system is able to successfully automate manual checkups, allowing the beekeepers to monitor the hives remotely, thus saving their time and effort. This ensures that the system can be successfully integrated with existing beekeeping practice. The system is able to handle multiple hive nodes across multiple apiaries. Furthermore, cloud-based data storage ensures that the hive data can be accessed from anywhere. This ensures scalability of the system. Additionally, the maintenance of the system is very minimal, limited to checking sensor coverings and ensuring proper power supply. Therefore, it is evident that the system is operationally feasible.

3.2.1.2 Technical Feasibility

Under technical feasibility, it is assessed whether the necessary technology and resources are available to develop the system. The required hardware and software components have been defined above. The ESP32 microcontroller, along with DHT22, rainfall sensors, and load cells, are available in Sri Lanka. Similarly, the necessary software components are also accessible. Access to Firebase services is straightforward, and devices with the required operating systems are readily available. Additionally, Arduino IDE, PyCharm, and Arduino Studio are free and easily accessible. The system requires a stable Wi-Fi connection for the ESP32 to send data to Firebase, which is also feasible. Therefore, it is evident that the system is technically feasible.

3.2.1.3 Economic Feasibility

Under economic feasibility, it is assessed whether the implementation of the system remains

within the budget constraints. The table below highlights the budget plan that was estimated during the feasibility study at the planning stage. (Note: Actual expenses have varied slightly.)

Table 2: Expenses for the sensors and microcontroller

Category	Type	Purchased/ Borrowed	Quantity	Price (Rs)
Sensors + Microcontroller	DHT22 Temperature & Humidity sensor	Purchase	2	950
	15 kg load cell	Borrow	1	-
	Hx711 Load Cell Amplifier	Purchase	1	290
	Rainfall Sensor	Purchase	1	250
	ESP32-WROOM-32 Microcontroller	Borrow	1	-
			Total	**2440**

Table 3: Expenses for power setup, wiring and connections of the IOT device

Category	Type	Purchased, Borrowed	Quantity	Price (Rs)
Power + Wiring + Connections	20W Sollar Panel	Purchase	1	4,000
	Lithium Ion Battery	Purchase	2	700
	2-Battery BMS	Purchase	1	110
	Printed Circuit Board	Borrow	1	-
	2 pin terminal block connectors	Purchase	1	30
	3 pin terminal block connectors	Purchase	3	40
	4 pin terminal block connectors	Purchase	1	70
	7805 1A regulator	Purchase	1	50

	Waterproof box enclosure for sensors	Purchase	1	1500
			Total	**7280**

Table 4: Expenses for the hive setup and beekeeping training

Category	Type	Purchased, Borrowed	Quantity	Price (Rs)
Hive Setup + Beekeeping Training	Wooden hive + Bee Colony	Purchase	1	10,000
	Load Cell Wooden Frame	Purchase	1	2,000
	Additional Wooden hive (for testing purposes)	Purchase	1	2,000
	Bee Keeping Training	-	-	5,000
	Continuous Management of the beehive	-	-	8,000
			Total	**27,000**

Full cost: ***Rs. 36,720***

The total cost is approximately 36,720, with the majority allocated to hive setup, management, and beekeeping training — components not directly related to the system implementation. Since some expensive sensors and devices were planned to be borrowed, the development of the system remained within budget constraints. Therefore, it is evident that the implementation of the system is economically feasible.

3.2.1.4 Schedule Feasibility

Under schedule feasibility, it is assessed whether the system can be completed within the given time period. The time plan and Gantt chart in section 3.1.3 indicate that the project can be finished before the deadline (April 9th, 2025). Therefore, the project is considered to be timely feasible.

3.2.2 Work Breakdown Structure

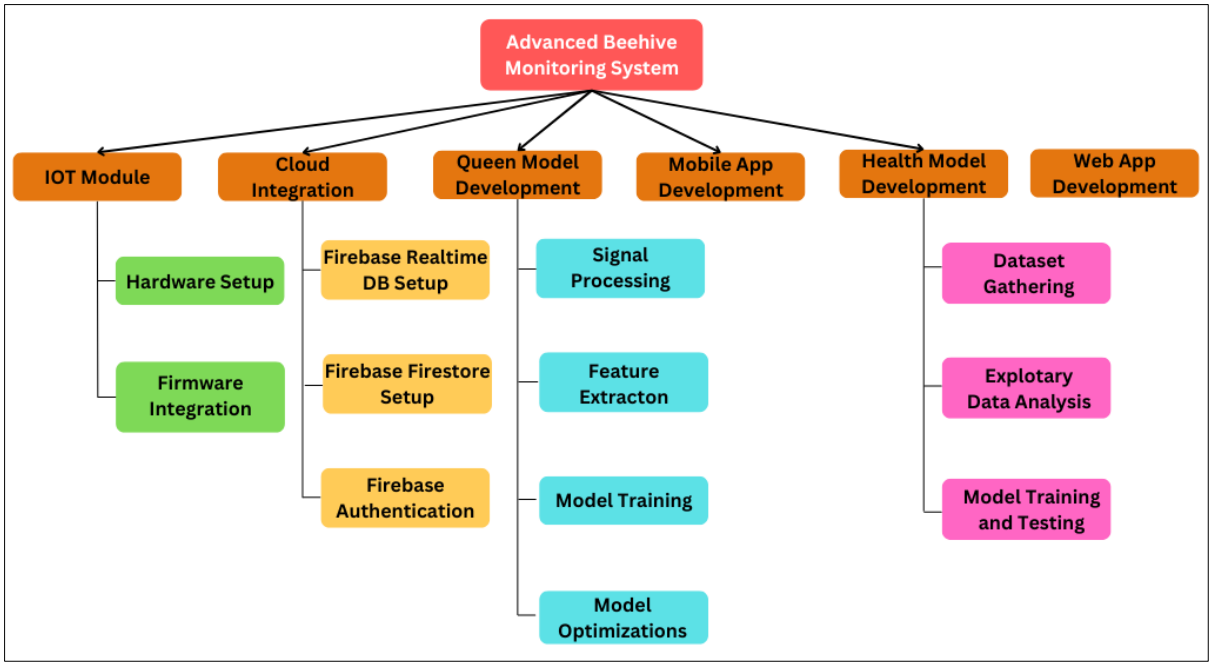


Figure 5: Work Breakdown Structure

3.2.3 Risk Analysis

The risk analysis matrix created during the project planning stage is shown below.

Table 5: Risk Analysis Matrix

Component	Risk	Likelihood	Impact	Mitigation Strategy
IOT Device	Malfunctions in sensors	High	High	Buy additional sensors, Purchase from reliable sources.
	Incorrect readings due to calibration issues with the load cell	Medium	High	Perform regular calibrations, Ensure the stability of the frame, Apply waterproof paint to avoid environmental impacts from temperature and humidity.
	Power Supply Issues	High	High	Use solar power to charge the batteries within the day, Use power banks to address power constraints during the night.

	Connectivity Issues between the IOT device and the firebase real-time database	High	High	Reduce unnecessary data in the JSON payload, Adjust the timeout configurations, Use a faster and reliable network.
	Data Losses	Medium	Low	Use linear interpolation to estimate the missing data.
Queen Detection Model	Uncertainty regarding the dataset's reliability and the experiment procedure	Low	Moderate	Read previous research done using the same dataset and try to reproduce their results.
	Model's lack of generalization	High	High	Use cross-validation as well as validate with new real-world sound samples.
	Lower accuracy when using cross validation	High	High	Use data augmentation techniques and experiment with different types of extracted features and classification models
	Machine power and memory constraints for model training	High	Moderate	Use google collab notebooks for model training
Mobile App	Integration and usability issues with the AI model	High	High	Use model-compression techniques to reduce the model size, Experiment with different flutter libraries to record the voice in .wav format.
	Bugs and crashes during use	Moderate	Low	Perform extensive testing

3.2.4 Stakeholder Analysis

Table 6: Skateholder Analysis

Stakeholder	Role/Interest	Influence	Engagement Strategy
Beekeepers	End users who monitor hive data & receive insights	High	Feedback collection after implementation and contribution to usability testing.
Bee Researchers	Interested in hive health and environmental data	Medium	Regular collection of advice, guidance, and feedback.
Local Farmers	May benefit from pollination.	Low	Awareness programs after the deployment
Admin/Manager	Oversees beekeeper registration and hive details	High	Feedback collection after implementation and contribution to usability testing.
University Supervisor	Guides the project and ensures its success	High	Project updates, Regular collection of advice, guidance, and feedback.

3.2.5 Time Plan

Table 7: Time Plan

Task ID	Task	Dependencies	Duration (weeks)	Start Date	End Date
1	Project Initiation, and Idea Finalization	-	2	Sep 1, 2024	Sep 14, 2024
2	Project Planning and Researching	1	3	Sep 14, 2024	Oct 5, 2024
3	Requirement Gathering and Analysis	2	2	Oct 5, 2024	Oct 20, 2024
4	Diagram Designing	3	1	Oct 20,	Oct 26,

				2024	2024
5	Implementing the Hardware setup	3	4	Oct 20, 2024	Nov 16, 2024
6	Collecting the dataset	5	4	Nov 16, 2024	Dec 14, 2024
7	Creating and training the Bee Queen Presence classification model	5	2	Nov 16, 2024	Nov 30, 2024
8	UI designing	7	2	Nov 30, 2024	Dec 14, 2024
9	Analyzing the collected dataset	8	3	Dec 14, 2024	Jan 4, 2025
10	Developing the mobile application	9	5	Jan 4, 2025	Feb 8, 2025
11	Testing	10	2	Feb 8, 2025	Feb 22, 2025
12	Documentation	11	2	Feb 22, 2025	March 8, 2025

Start Date: September 01, 2024

End Date: March 08, 2025

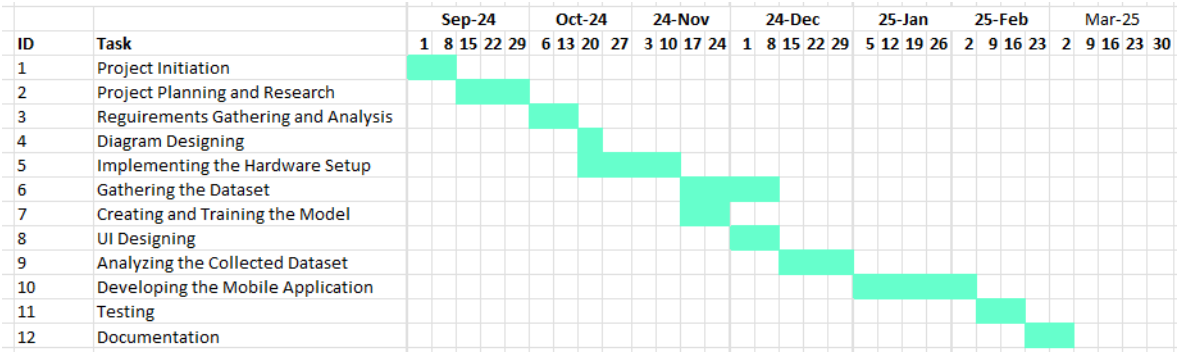


Figure 6: Gannt Chart

3.3 Requirements Analysis

3.3.1 Requirements Gathering Techniques

3.3.1.1 Observations

As the primary requirements gathering technique, a filed visit to the Bandarawela Bidunuwewa Bee Unit was carried out.

Figure 7: Photos taken during the field visit to the Bandarawela Bidunuwewa Bee Unit

The visit to the bee unit helped to gain educational knowledge regarding the behavior of bees and standard beekeeping practices.

Apart from that, a local apiary belonging to a beekeeper in Horana was also visited to collect additional requirements.

Figure 8: During the visit to an apiary in Horana

3.3.1.2 Interview

A Zoom interview was conducted with Mr. Vimukthi, who is the director at Bandarawela Bidunuwewa Bee Unit in order to verify and confirm the gathered requirements.

Figure 9: During the zoom interview with the director at Bandarawela Bidunuwewa Bee Unit

The following were the questions that were asked and the answers that were received during the interview.

1	Sir, what is the most commonly farmed bee species in Sri Lanka?
	Mostly Asian Honey Bees. Some bee keepers use certain breeds of imported European Honey Bees.
2	Sir, what is the current method commonly used by Sri Lankan bee keepers to detect the queen's precense or absence?
	Beekeepers must carefully inspect each frame by hand to identify the queen or detect the presence of eggs.
3	After detecting queen's absence what do the bee keepers usually do?
	Remove any queen cells if any is visible. Add an artificially bred, undisabled, active and healthy queen,that has a longer body. Queen's lenght usually determines the queen's productivity.
4	Will it be beneficial if an automatic method is introduced for beekeepers to monitor the queen's absense or precense in the hive remotely?
	Yes. It will help inexperienced bee keepers, as they lack the essential experience to check for the queen manually. It will also help professional bee keepers to save their time and effort.

5	As for the research I have conducted, Asian Honey Bees tend to maintain hive inside temperature around 34.5-35.5 celcius, given that the colony is healthy. For your knowledge do you agree with this? Or are there any distinctive temperature patterns noticed among asian honey bee colonies in Sri Lanka?
We haven't conducted much research on that. In certain occassions, we used thermometers to measure the hive temperature of the hives, specially during the wing virus outbreak.That gave us higher temperatures.	

6	Same question for hive inside humidity. According to the researh I codnucted, it is learnt that asian honey bees regulate the hive inside humidity around 75%, given that the colony is healthy. What are your opinions about this?
I haven't given much though to the hive humidity. But have heard from the books that it is also an important factor for the bee colony's health.	

7	How do Sri Lankan bee keepers check for the colony's health?
They have to visit the hives regularly and check each of the frame, in order to detect any abnormalities within the hive. Such as detecting any signs of pests or diseases or unsusual behaviour of the bees.	

8	What if an automatic method is developed, so that the bee keepers can monitor and analyze the bee colony's health remotely?
It would be beenficial, and is the most needed for bee keepers.	

9	If we develop a system for the bee keepers to monitor their hives remotely, what would be the best deliverable according to you? Mobile App? Web App? SMS?
Mobile app would be the best. It will be simple and efficient, as most of the bee keepers have a mobile phone with them.	

Figure 10: Questions that were asked and the answers that were received during the zoom interview

3.3.2 Requirements Specification

3.3.2.1 Functional Requirements

- The system must allow the users to log in successfully with correct credentials.
- The system must allow the admin to delete, update or edit user, apiary and hive details.
- The system must display only the apiaries that belong to the particular user after logging in.
- The system must display only the specific hives that belong to the selected apiary after choosing an apiary.
- The system must display real-time hive-specific sensor data.
- The system must provide past sensor data in graphs for trend and pattern analysis
- The system must allow the beekeeper to record bee sound through the app using the mobile microphone.
- The system must be able to process raw audio signals and predict whether the bee queen is present in the hive or not.
- The system must display the classification result: whether the bee queen is present or absent.
- The system must display a hive's history of past queens.
- The system must allow the beekeeper to add details of a new queen bee.
- The system must allow the beekeeper to update if the current queen has been replaced.
- The system must analyze the hive health and provide real-time insights.

3.3.2.2 Non- Functional Requirements

- The system must load and display real-time data within 5 seconds.
- The system must support multiple hives and large datasets without performance degradation.
- The system must use data encryption and user authentication to protect sensitive information.
- The system must provide a user-friendly interface for easy navigation.
- The system must support both Android and iOS devices.
- The codebase should be modular for easy updates and bug fixes.

3.3.2.3 Hardware/Software Requirements.

Figure 11: Hardware Requirements

Figure 12: Software Requirements

3.3.3 Use Case Analysis.

Use cases were designed in order to analyze the user requirements. Few designed use cases are shown below. **Appendix D: Use Cases Continuation** contains all the designed use cases.

Table 8: Use Case Analysis for login

Use Case ID:	**1**

Use Case Name:	Login		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper	Admin	
Description:	Both the beekeepers and admins must be able to login to their respective apps using the correct login credentials.		
Preconditions:	Beekeepers are already registered by the admin. The system inbuilt admin credentials are given to the admin		
Postconditions:	Successful login to the respective apps if the login credentials are correct.		
Normal Course of Events:	1. The user enters their username. 2. The user enters their password. 3. The user clicks on the login button.		
Alternative Courses:	-		
Notes:	1. The beekeeper cannot sign up on their own, as they can only be registered in the system when the business installs the IoT device on one of their hives. Therefore, registering or signing up the beekeeper is the responsibility of the admin. Not that of the beekeeper himself. This prevents unnecessary signups.		

Table 9: Use Case analysis for viewing live sensor data

Use Case ID:	**2**		
Use Case Name:	View live sensor data from each hive		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to view live sensor data from each hive.		
Preconditions:	The beekeeper logs in and selects a specific hive. The selected hive has been previously installed with an IoT module by the business, and the module is powered and functioning properly.		
Postconditions:	Beekeeper can successfully view live sensor data from each hive		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. View the live sensor data of the selected hive.		
Alternative Courses:	-		
Notes:	• The live sensor data received should be from independent hives. The live data can only be viewed correctly if the installed IOT module is powered correctly, and sensors do not have any malfunction. Each hive has their own unique sensor module, so the upcoming live data's origin is distinguished through the given Hive ID.		

Table 10: Use Case analysis for detecting the queen's presence

Use Case ID:	3		
Use Case Name:	Detect the Queen's absence or presence.		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to detect whether the queen bee is present in the hive (*queenright*) or absent (*queenless*) by analyzing audio data collected from the hive.		
Preconditions:	The beekeeper is logged into the system, The mobile device's microphone is functioning properly and is sensitive enough to capture bee sounds.		
Postconditions:	The beekeeper receives a reliable prediction indicating whether the hive is queenright or queenless.		
Normal Course of Events:	1. The beekeeper visits the hive where queen presence needs to be detected. 2. The beekeeper navigates to the UI section for audio recording and queen presence detection. 3. The device is held near the hive or placed inside on top of the brood box. 4. The beekeeper clicks the "Record Audio" button to begin recording the audio. 5. After approximately one minute, the beekeeper clicks "Stop Audio. 6. The beekeeper then clicks the "Predict" button to process the audio data and detect queen presence. 7. The prediction result is displayed to the beekeeper.		
Alternative Courses:	1. The beekeeper may click the "Play" button to listen to the recorded audio before making a prediction.		
Notes:	• The device's microphone should be sensitive to bee sounds • The background noise should be minimal when taking the audio • The recording should take around 1 min to get a generalized and more accurate prediction result. • Repeat the steps a few times to obtain an accurate average result before being 100% certain whether the hive is queenright or queenless.		

Table 11: Use Case analysis for monitoring hive's health

Use Case ID:	4		
Use Case Name:	Monitor hive's health status		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last	4/15/2025

		Updated:	
Actor:	Beekeeper		
Description:	Beekeepers should be able to receive real-time insights into hive's health status. (whether the hive is healthy or not).		
Preconditions:	The beekeeper logs in and selects a specific hive. The selected hive has been installed with an IoT module by business, and the module is powered and functions properly.		
Postconditions :	The beekeeper receives a reliable prediction indicating whether the hive is healthy or not.		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. View the health status of the selected hive.		
Alternative Courses:	-		
Notes:	• In order to determine the current health status of the hive, analyze at least data from the past 24 hours in order to get a more accurate result for the hive health status.		

3.3.4 Personas Development

Figure 13: Persona for Mr. Vimkuthi

Figure 14: Persona for Mr. Tharanga

3.3.5 Requirements Prioritization

Table 12: Requirements Prioritization Table

Requirement	Type	Priority	Reason
Login	Functional	***Moderate***	Needed but not essential at this phase of the development
Manage hives, apiaries and beekeeper details	Functional	***Low***	Not essential for the basic operations of the system. As the purpose of this requirement can be accomplished through direct access to the database
Display Realtime live sensor data	Functional	***High***	A critical user requirement

Display the specific apiaries of the logged in beekeeper	Functional	***Moderate***	This requirement is only needed when the beekeeper manages hives at multiple places, which happens rarely in Sri Lanka
Display the specific hives of the selected apiary	Functional	***Moderate***	Only required when the hives are organized into apiaries.
Vizualize past sensor data	Functional	***High***	Crucial to analyze hive trends and bee behavior
Queen presence/ absence detection	Functional	***High***	A critical user requirement
User friendliness and simplicity of the interfaces	Non-functional	***High***	Crucial for user experience
Compatibility with both IOS and Android	Non-functional	***Moderate***	iPhone usage is minimal among Sri Lankans
Analyze hive health	Functional	***High***	A critical user requirement
Security and privacy assurance	Non-functional	***Moderate***	Required but not essential at this phase of the development
Manage the current queen details	Functional	***Low***	Not a key user requirement.
Manage past queens' details	Functional	***Low***	Not a key user requirement

3.4 Design

3.4.1 Circuit Diagram

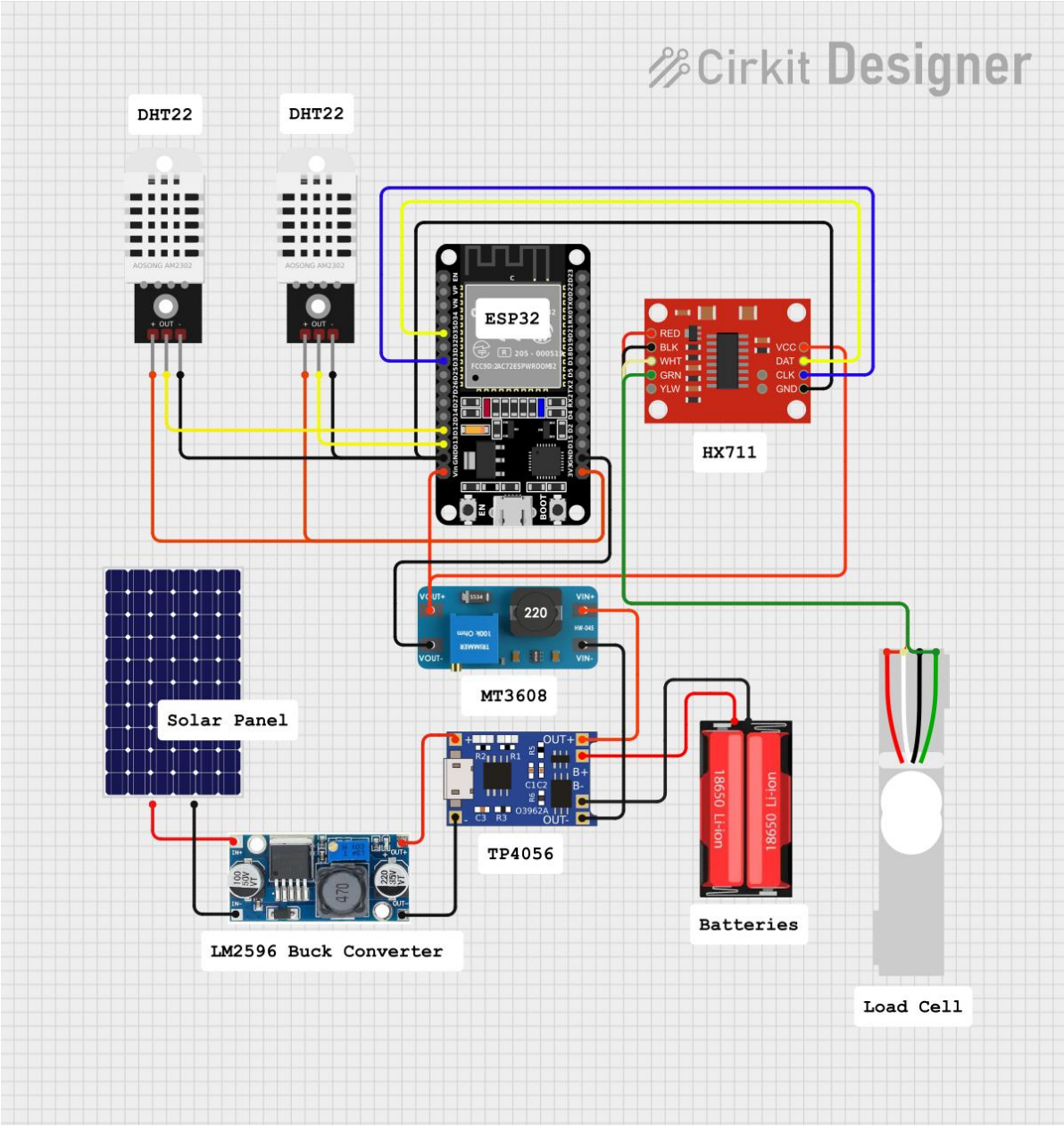


Figure 15: Circuit Diagram

3.4.2 Use Case Diagram

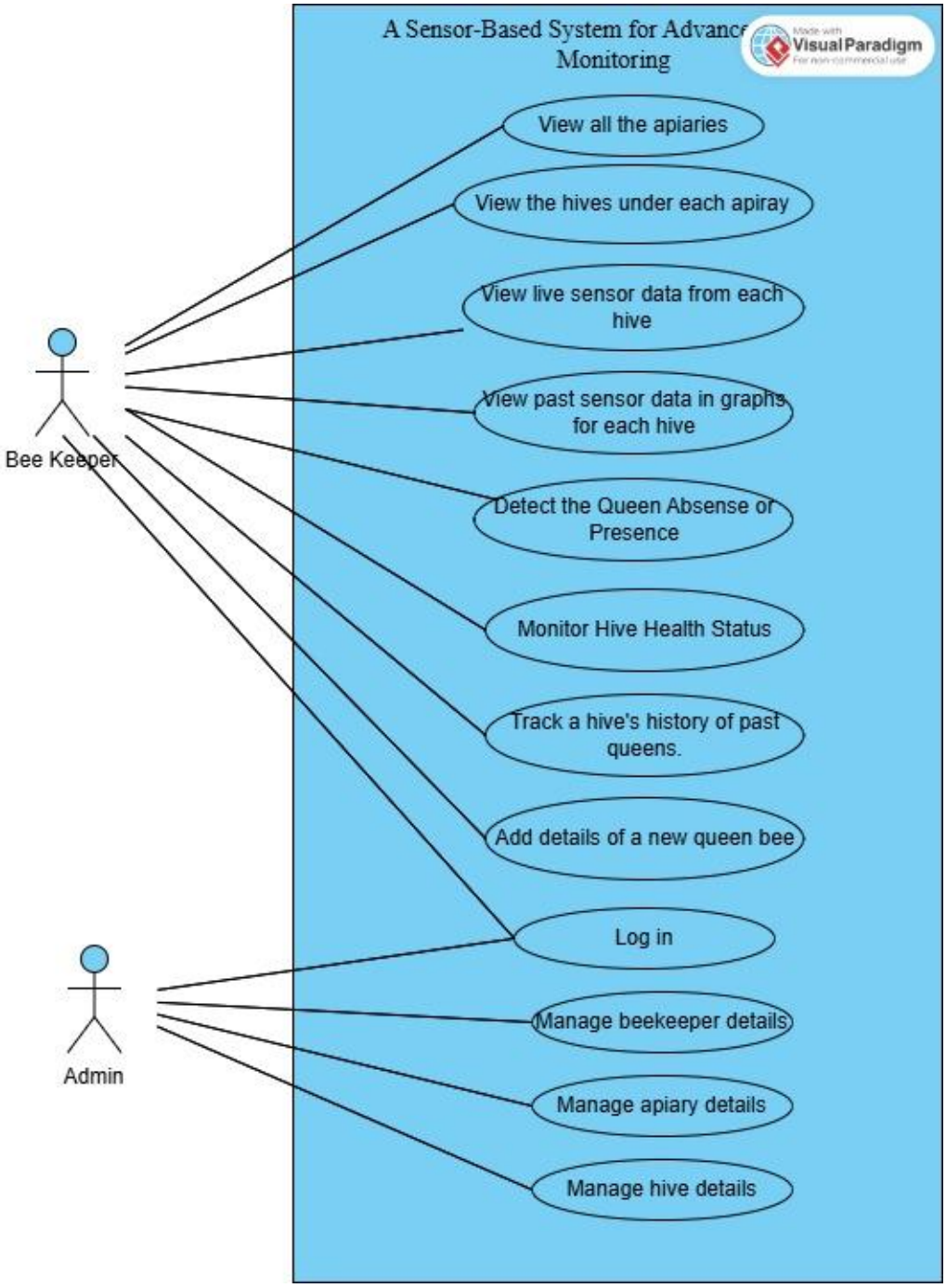


Figure 16: Use Case Diagram

3.4.3 Storyboards

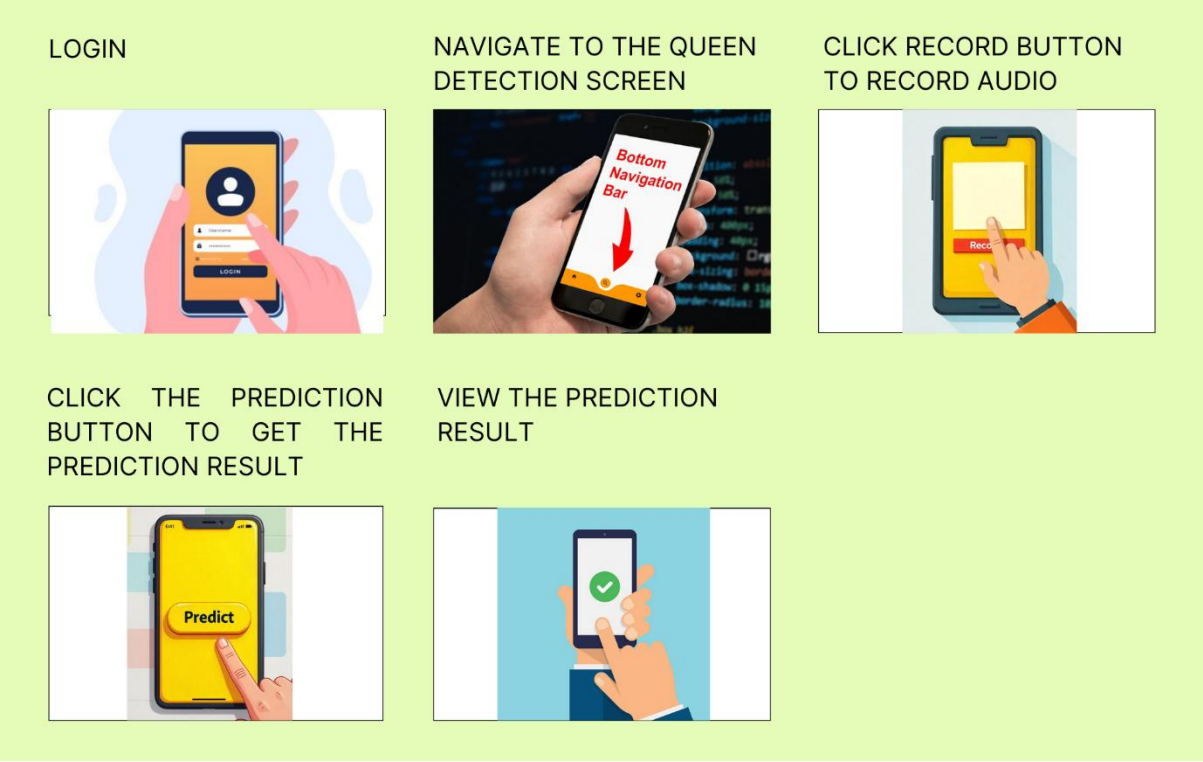


Figure 17: Storyboard for using the app to detect whether the hive is queenless or queenright

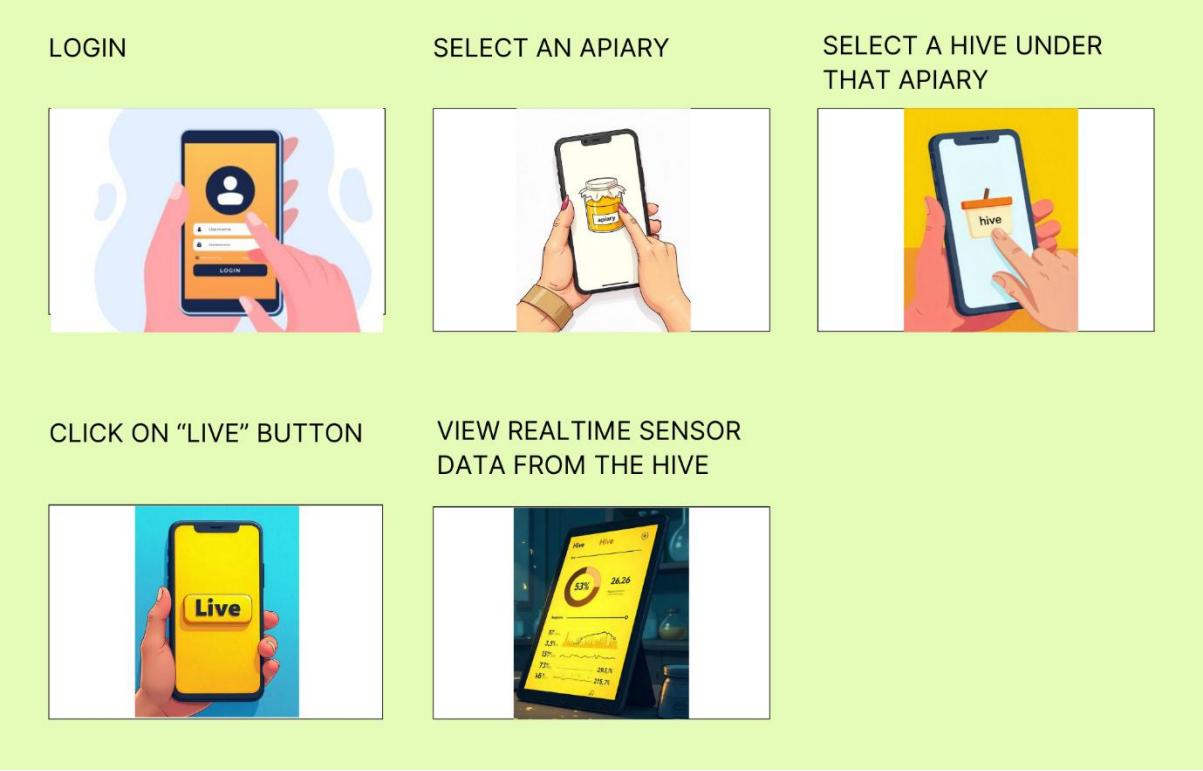


Figure 18: Storyboard for using the app to view live sensor data from a particular hive

3.4.4 Class Diagram

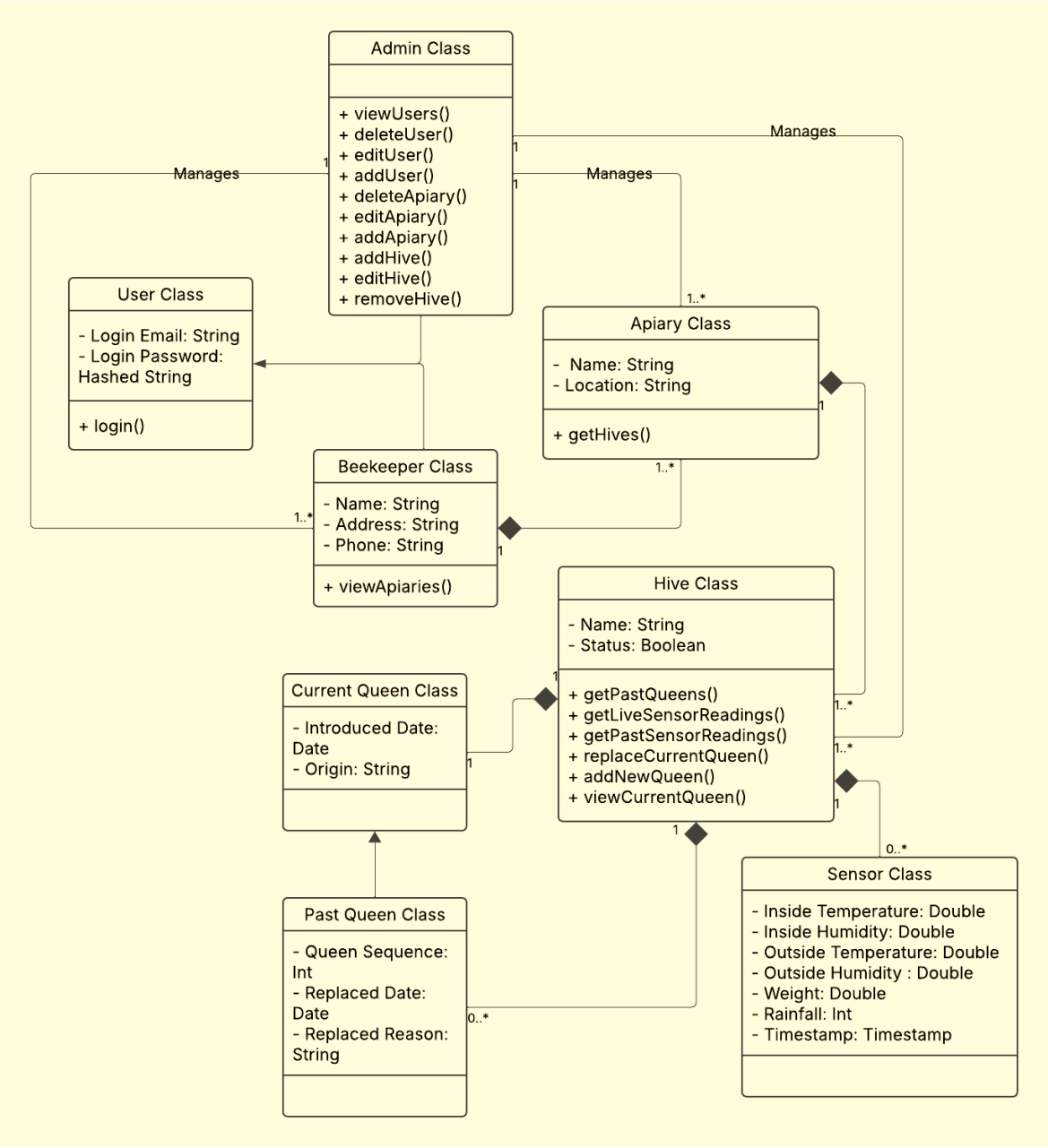


Figure 19: Class Diagram

The class diagram above has 8 main classes, such as sensor class, hive class, apiary class, etc.

3.4.5 Database Diagrams.

In this project, the Firebase was used for both the Realtime Database and the backend database. As Firebase is a NoSQL database, there is no strict need for an ER diagram. Therefore, a database design diagram that supports NoSQL databases has been drawn for both the backend database and the Realtime Database.

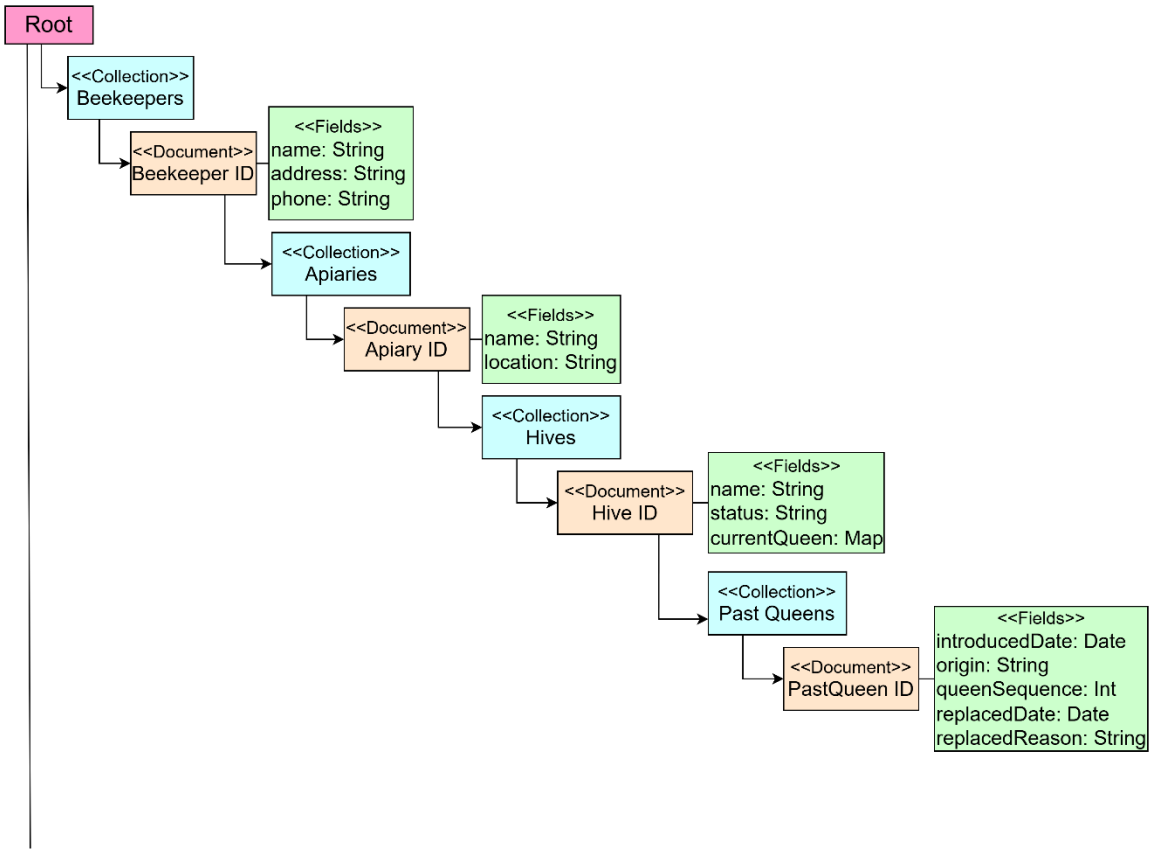


Figure 20: Database design diagram for the firestore database

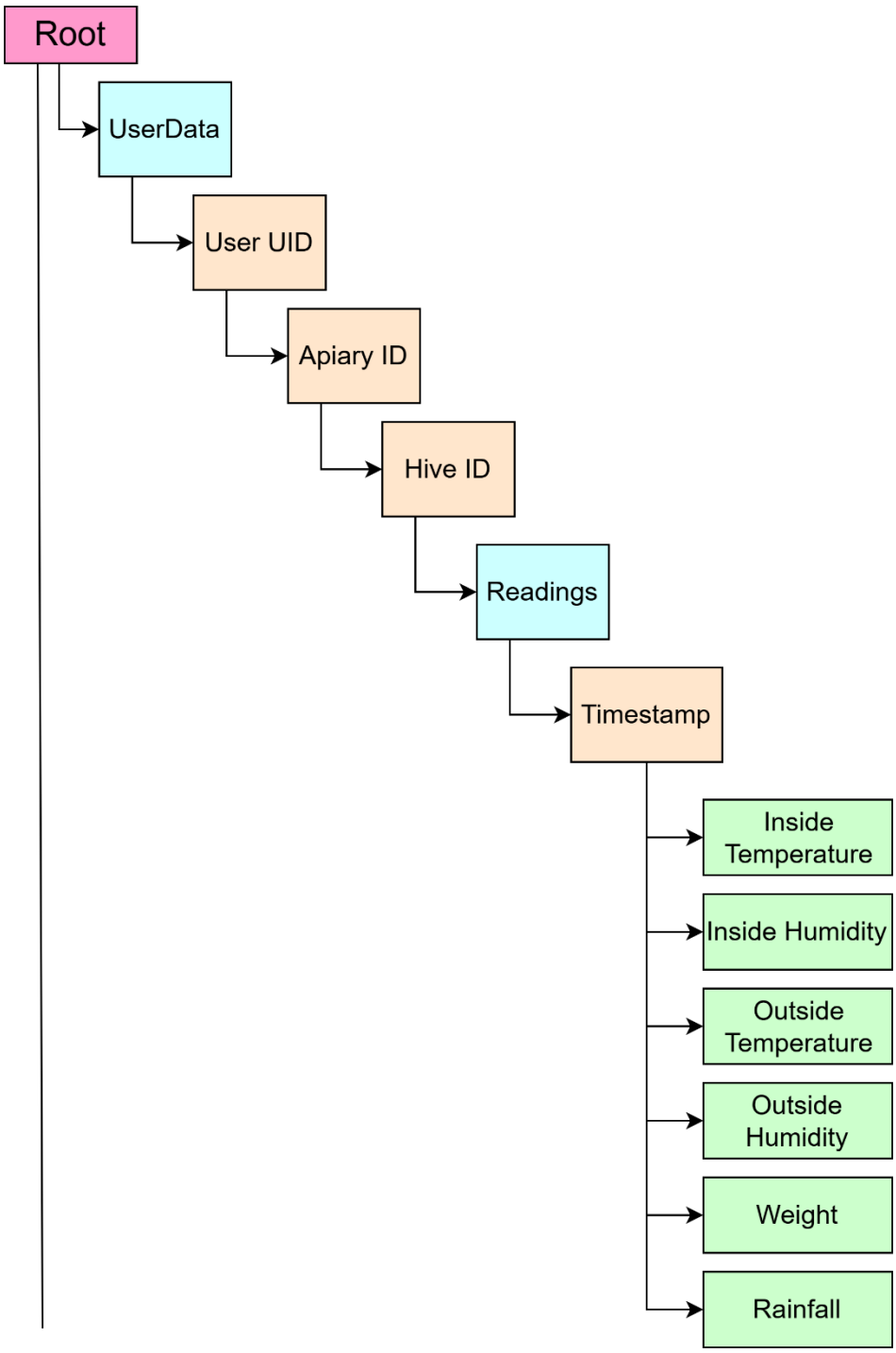


Figure 21: Database design diagram for the firebase realtime database

3.4.6 High Level Architecture Diagram

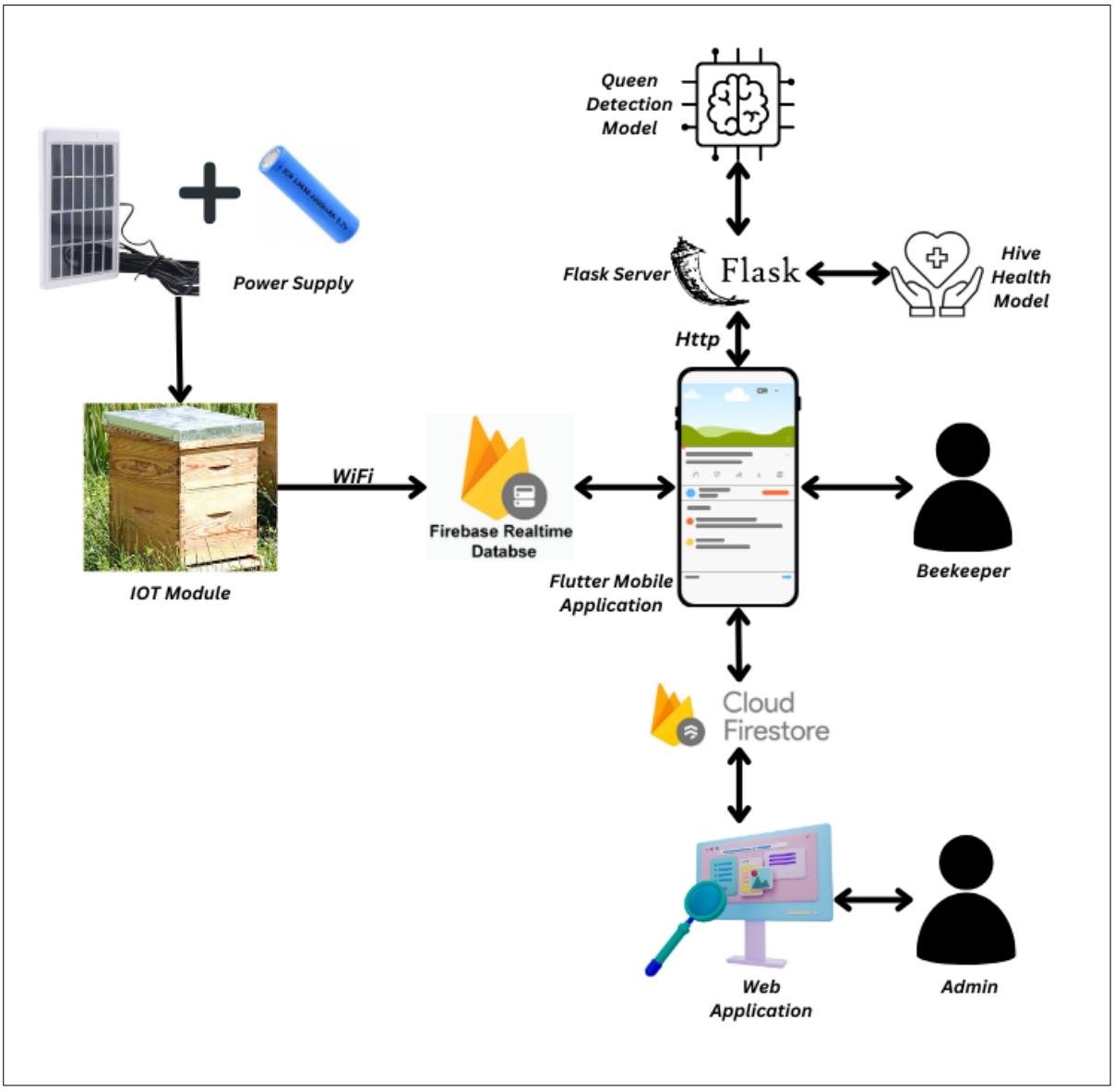


Figure 22: High level architecture diagram

3.4.7 Wireflow Diagram (Paper-Based)

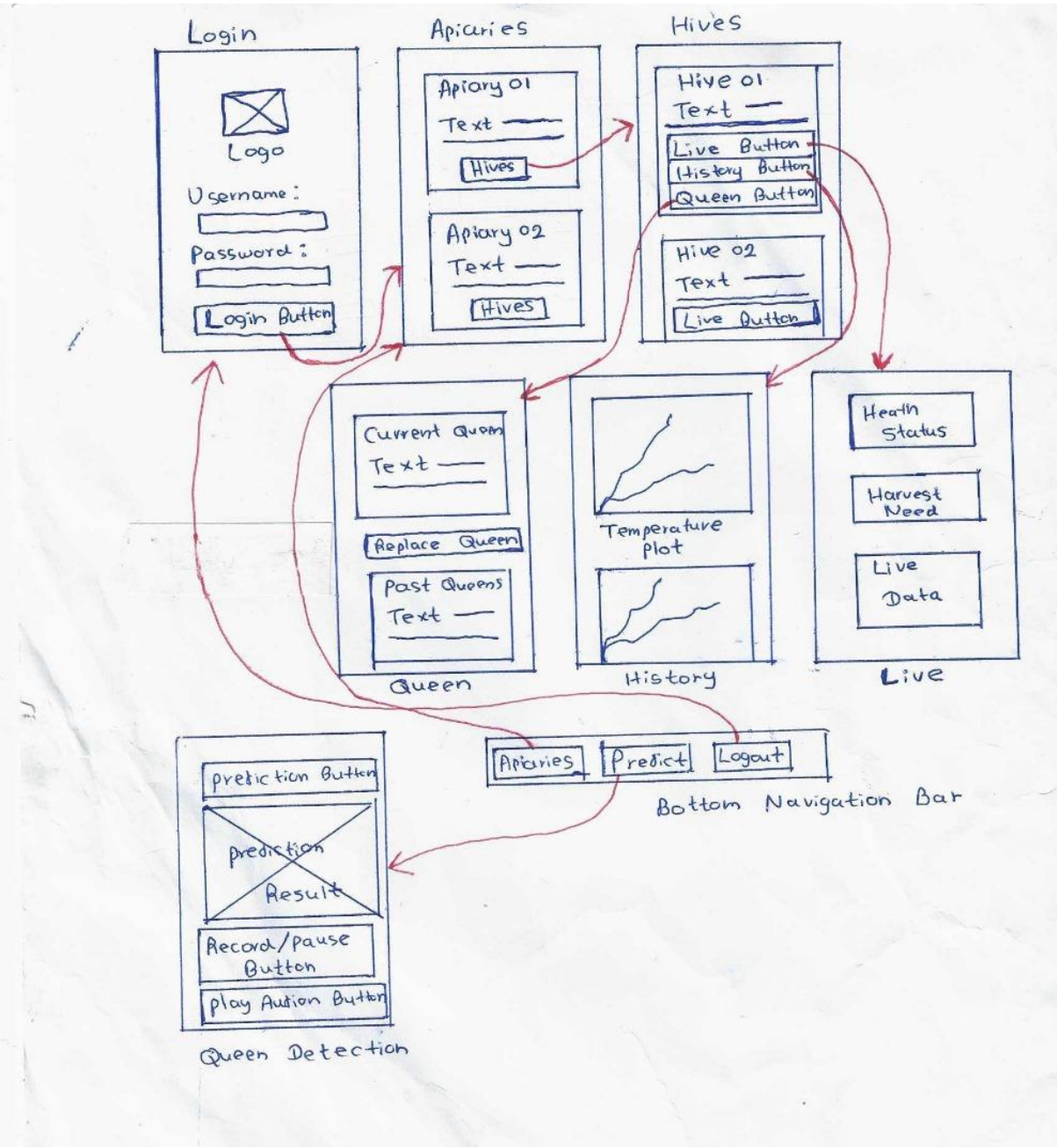


Figure 23: Paper based wireflow diagram

Paper-based prototypes are very common in software development processes as they allow developers and designers to quickly visualize and test ideas before moving to digital implementation.

3.4.8 Level 0 DFD Diagram

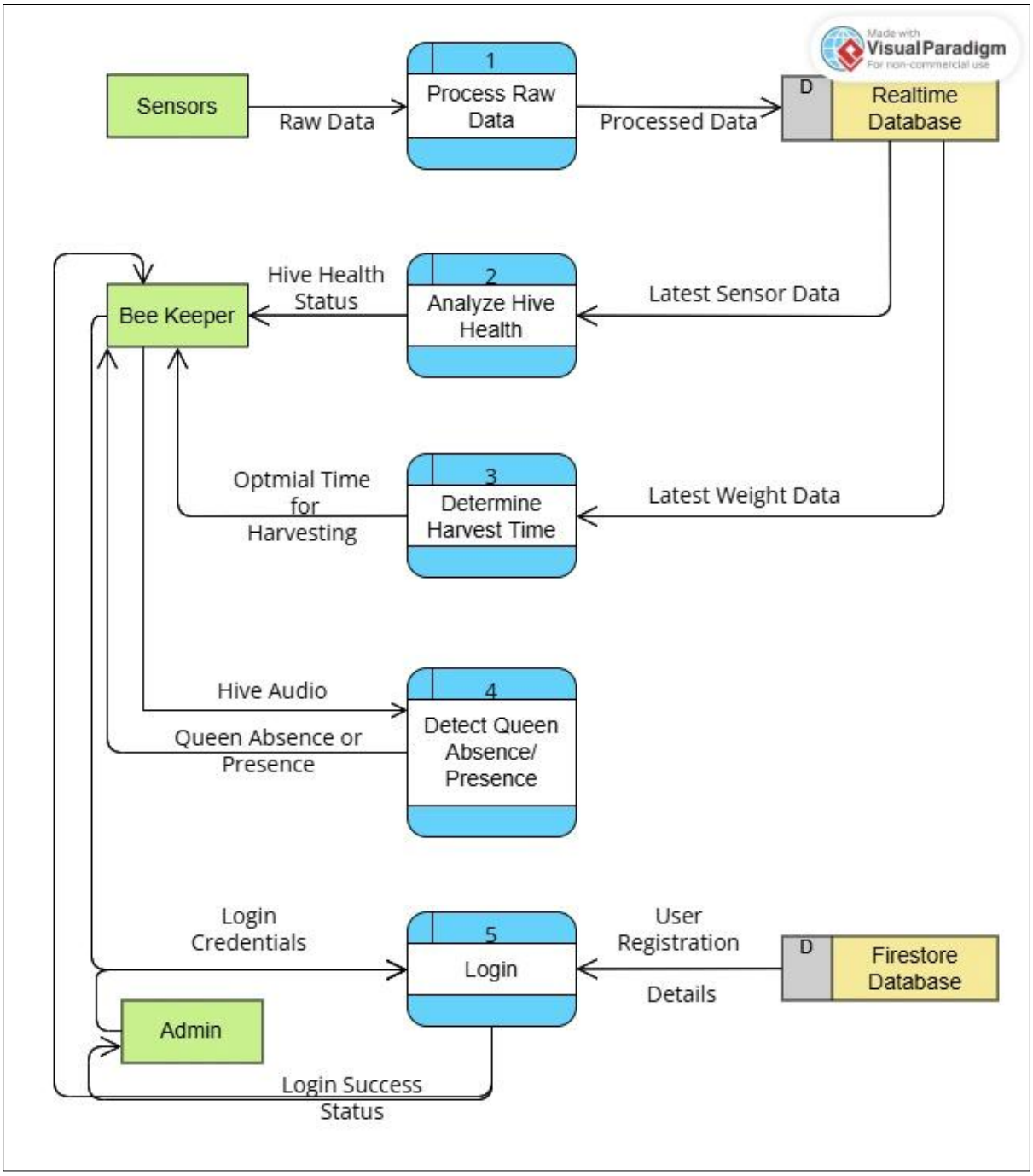
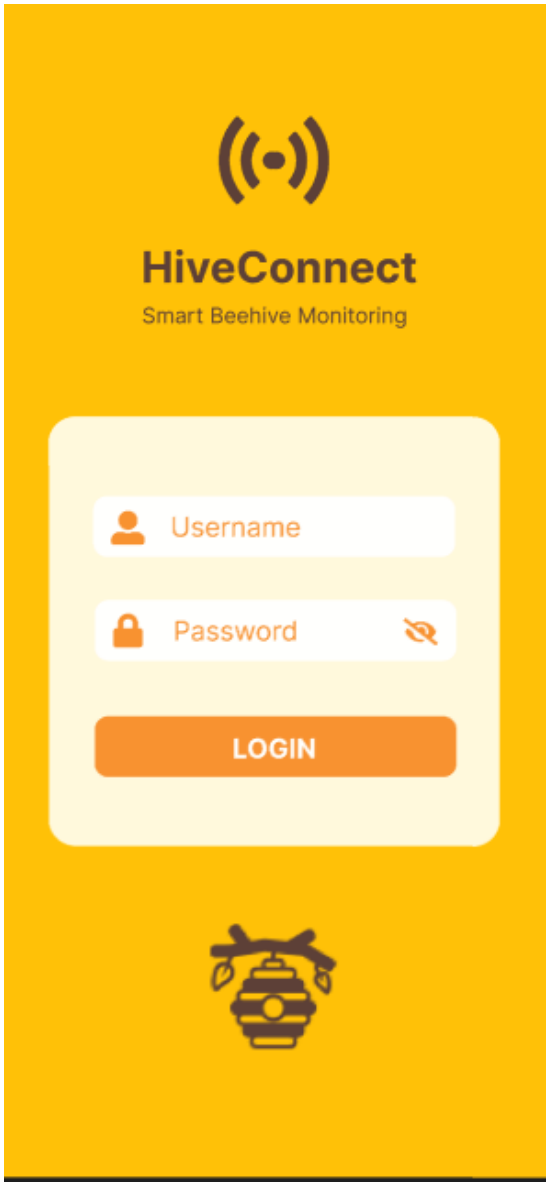


Figure 24: Level 0 DFD

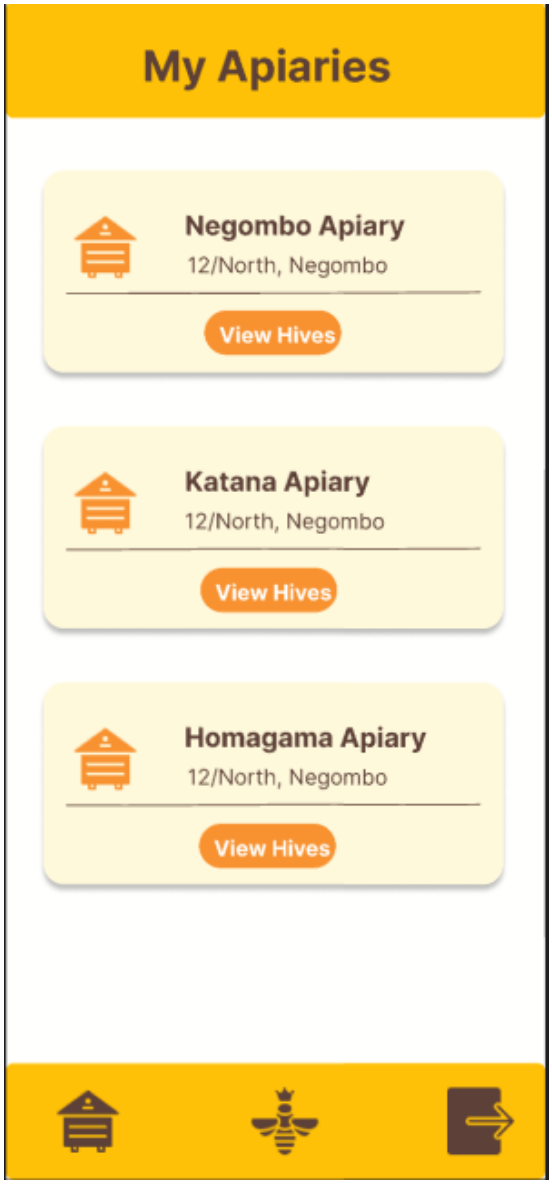
3.4.9 UI Designing.

Figma was used for UI designing. All the designed UI can be accessed from: https://www.figma.com/design/baCojaEdPngw7sL5Fr36dv/FYP-UI?node-id=0-1&t=03OBJJUJLZeZVp0M-1

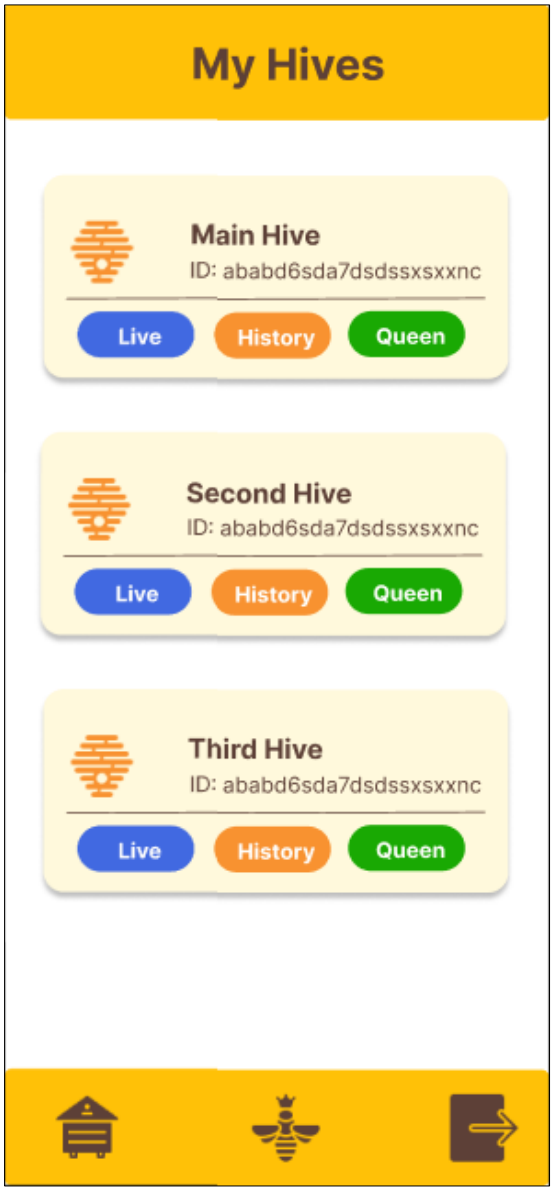
Login Screen



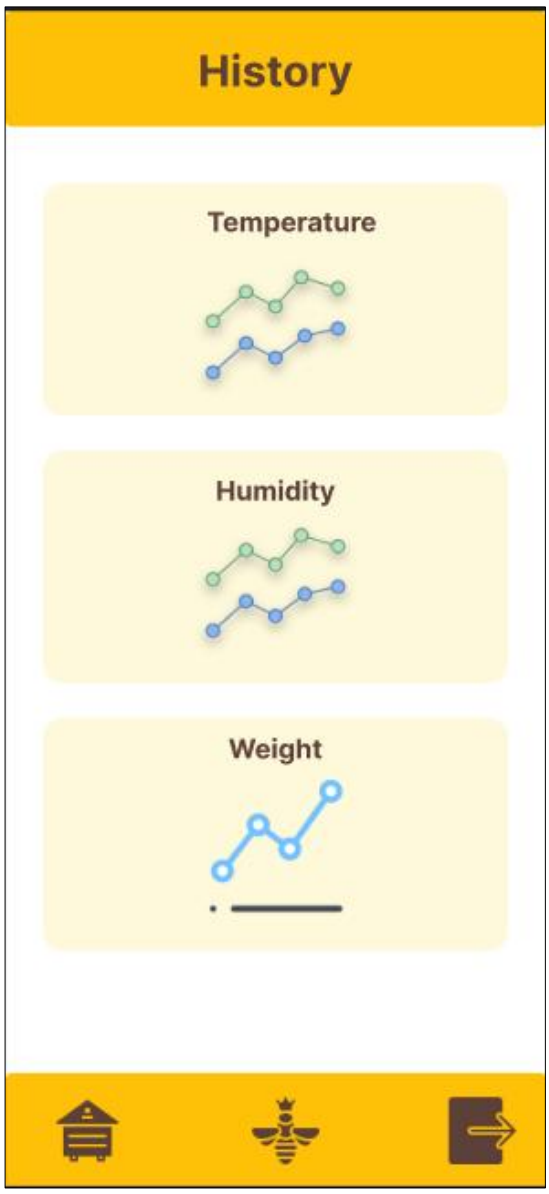
Apiaries Screen



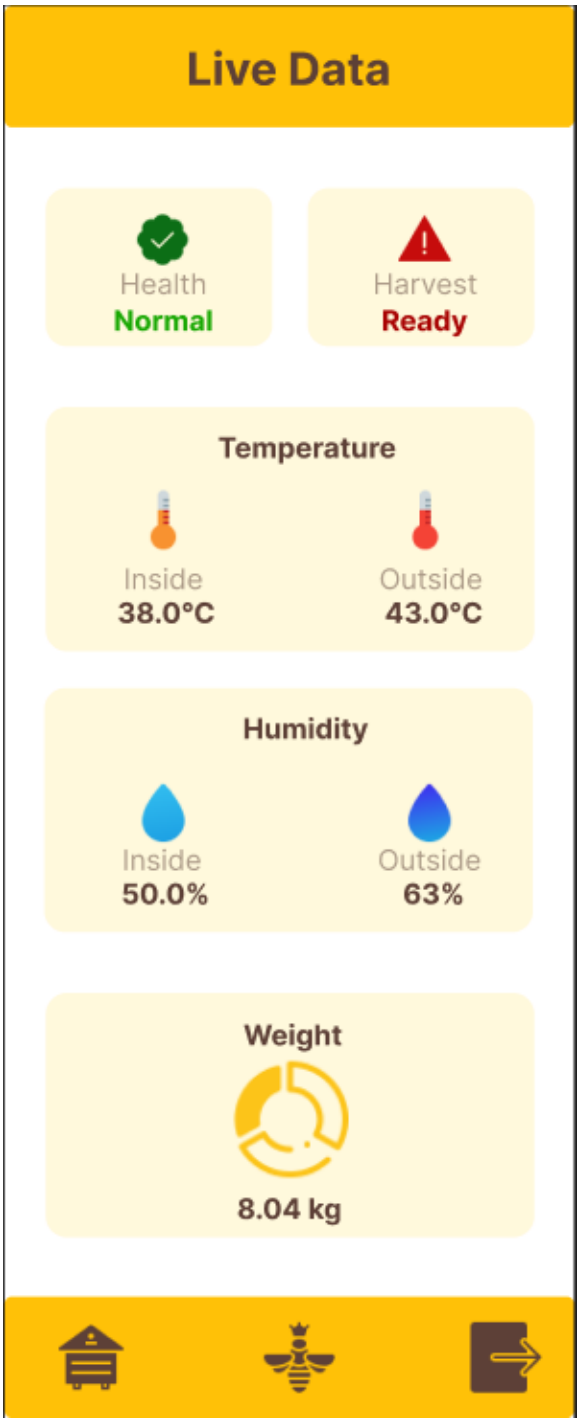
Hives Screen



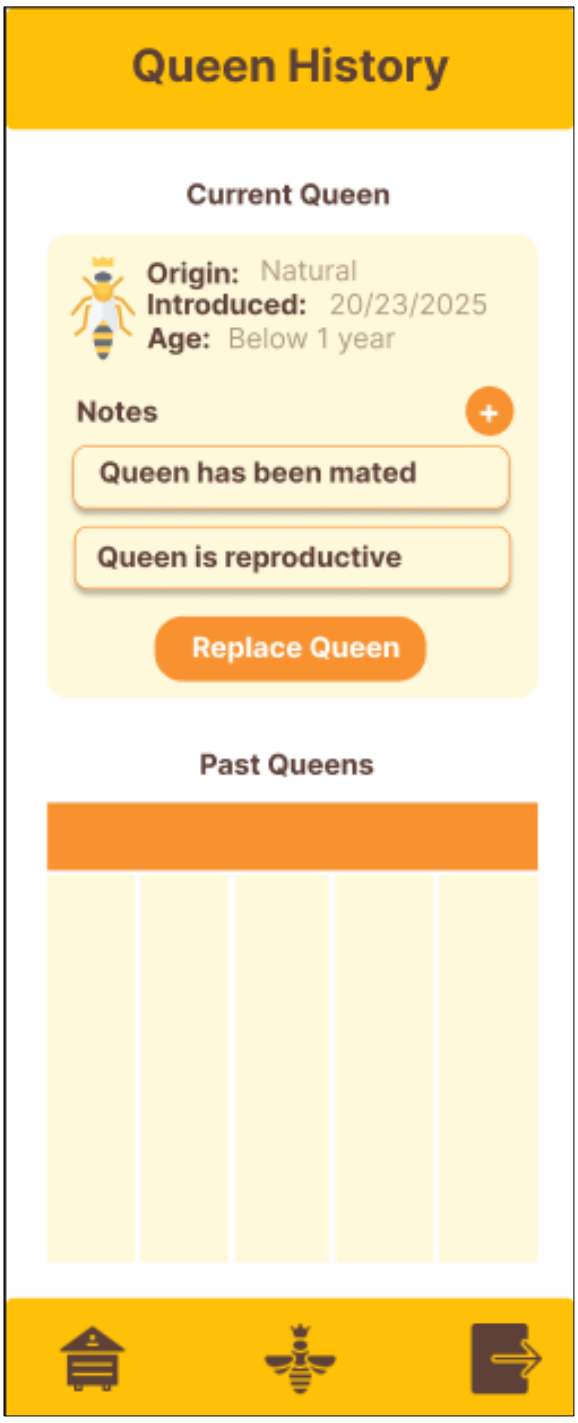
History Screen



Live Screen



Queen History Screen



Queen Detection Screen

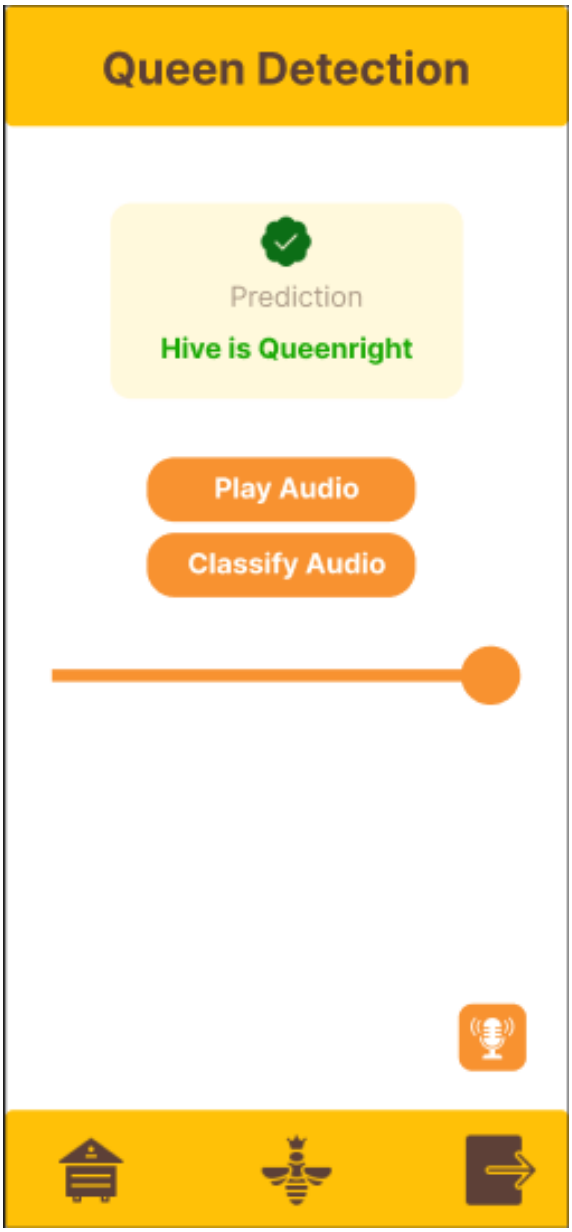


Figure 25: Figma UI designs for the mobile app

3.5 Implementation

3.5.1 Overview

As the waterfall methodology was followed, the implementation phase was carried out after the design phase. The implementation phase consists of a total of 8 tasks. These tasks are depicted on the table below.

Table 13: The 8 tasks carried out in the implementation process

Task/ Component	Description
Hardware Setup	Implementation of the hardware setup, including sensor integration, microcontroller wiring, soldering, and power supply configuration.
Firmware integration	Development and deployment of the code that runs on the microcontroller.
Cloud Setup	Establishment of the cloud infrastructure, covering both the real-time database and the data storage for other structured data, as well as cloud authentication.
Queen Detection Model Implementation	Includes the ML pipeline carried out in order to train a model to detect the queens' presence or absence using bee sound samples
Hive Health Analysis Model Implementation	Includes the ML pipeline carried out to develop a model that assesses hive health based on sensor data.
Backend Server Implementation	Development of the backend server to connect the machine learning models with the mobile application.
Mobile Application Development (for Beekeepers)	Implementation of the mobile app designed for beekeepers to monitor their hives.
Web Application Development (for Administrators)	Creation of the admin portal that enables administrators to register beekeepers and manage apiary and hive information.

Each of these tasks contributed to the successful deployment and functionality of the overall system. In detail discussion on each of these tasks will be covered in the subsequent sections.

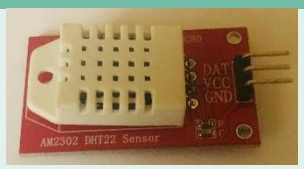
3.5.2 Hardware Setup

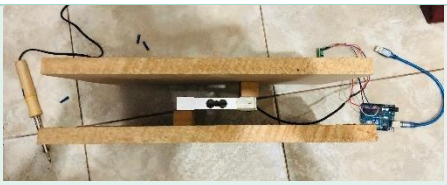
This task involves the IoT module setup, the power supply setup, and the final assembly of the system. The IOT module involves wiring, soldering, and other physical tasks related to the circuit with the sensors and the microcontroller. The power supply involves the power supply setup that was implemented to power the IOT module.

A comprehensive illustration of the sensor, ESP32 wiring, and power supply setup is provided in the **3.4.1 Circuit Diagram** section. This diagram offers a detailed visual representation of how each sensor is connected to the microcontroller and other relevant components. It also illustrates the power supply flow — from the solar panel to the lithium-ion batteries, and finally to the IoT module.

The IOT module contains a total of 3 sensors. A detailed description of each of these sensors is illustrated on the table below.

Table 14: Summary of the used sensors

Sensor	Image	Why used in the project?	Placement
DHT22	 **Figure 26:** DHT22 sensor 1	To measure the outside temperature and outside humidity	Outside the hive
DHT22	**Figure 27:** DHT22 sensor 2	To measure the inside temperature and inside humidity	Inside the hive

15Kg Load Cell + HX711 Amplifier	 **Figure 28:** Load cell mounted on a wooden frame	To measure the weight of the hive.	Outside the hive

The ESP32 NodeMCU was used as the microcontroller, which receives sensor data in a timely manner and then processes the raw data and sends the processed data to the cloud database.

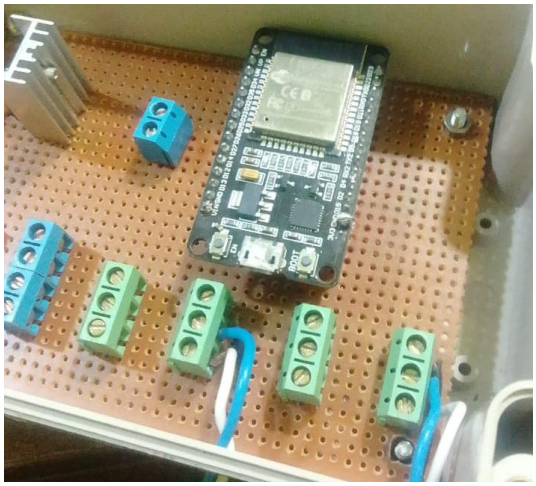


Figure 29: The ESP2 used in the project placed to the circuit

The rationale behind choosing the ESP32 NodeMCU as the microcontroller is as follows:

- The ESP32 has a dual-core processor and ample memory resources that enables efficient processing of sensor data as shown in (Babiuch, 2019).
- The ESP3 comes with built-in Wi-Fi and Bluetooth capabilities.
- The ESP32 provides a variety of digital and analogue input/output pins.
- The ESP32 platform is well-supported by a large and active community of developers, providing access to a wealth of resources, libraries, and tutorials as shown in (Maier, 2017).
- The ESP32 is a cost-effective device.

The IOT module is powered by 2 lithium-ion batteries which were connected parallelly to each other. These batteries are then powered through a 20W solar panel. All the components used

for the power supply setup are shown in the table below.

Table 15: Summary of the components used in the power supply

Component	Image	Why used for the power supply?
Solar Panel		Captures solar energy to charge the batteries during the day.
LM2596 Buck Converter		Steps down 17V from the solar panel to 5V for the TP4056 module.
TP4056 Module with Protection		Charges the lithium-ion batteries from the current sent from the solar panel
2 Parallel Lithium-ion Batteries		Provides the power for the IOT module even at night using the stored energy.

MT3608 Boost Converter		Boosts battery output to 5V for the ESP32 and sensors

The completion of the IoT module and power supply setup is marked by the final assembly of the circuit and all components, which are housed within a box enclosure. This setup was then placed on three different real-world beehives (two were used to gather the datasets, one was used as an experiment) to begin collecting data for analysis in the health model implementation section.

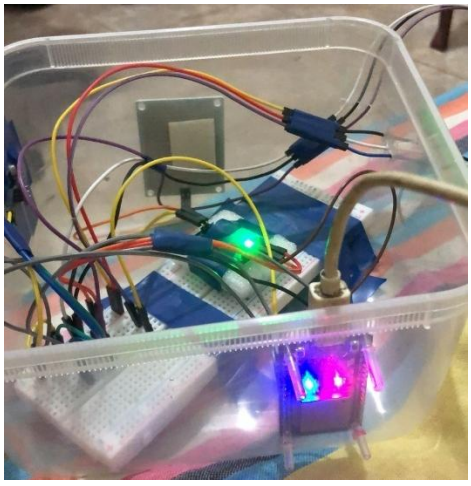


Figure 30: The prototype of the IOT module

Figure 31: Implemented IOT prototype placed to a beehive in Rajagiriya

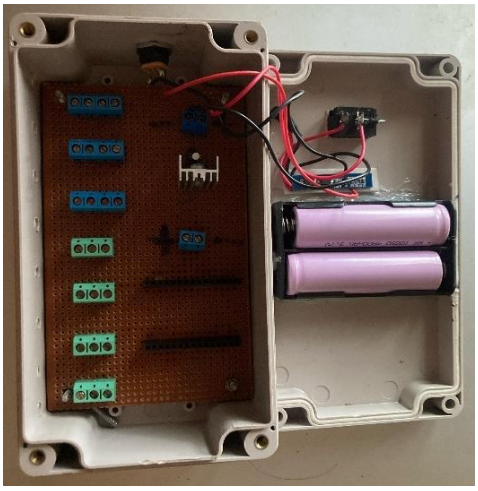


Figure 32: The implemented final IOT product

Figure 33: The final IOT product placed on a beehive in Athurugiriya

Figure 34: The final IOT product placed on a beehive in Katana

3.5.3 Firmware Integration

After completing the hardware setup, the ESP32 microcontroller was programmed with firmware to calibrate the sensors, collect sensor data, process it, and send the data to the cloud database. The following figure illustrates the programming language, development environment used to implement and test the firmware, as well as the version control system employed for the firmware.

Figure 35: The programming language, development environment and version control used to implement and test the firmware

The ESP32 was flashed with two different firmware images, each with its own purpose. They are as follows:

1. Calibration Firmware: The first program specifically designed to determine the calibration values for the sensors.
2. Final Operation Firmware: The second program that uses the calibration values and is designed to retrieve accurate readings from the sensors, process the raw sensor data, and send the processed data to the cloud database via WiFi.

Before uploading the final firmware, proper calibration needed to be performed in order to get accurate readings from the sensors**.** Out of the three sensors used in the project, only the load cell required calibration through the calibration firmware image. The calibration process was performed after the load cell was mounted on a metal platform, which was supported by two wooden frames on either side. After that, the ESP32 was flashed with the calibration firmware that utilizes the HX711.h library to calculate the calibration factor for the load cell, based on the readings obtained after placing a known weight on the top frame.

One of the main processes in the final operational firmware image is to process the retrieved raw sensor readings. The following are the data processing pipelines used in the final operation firmware.

DHT22 temperature and humidity (Inside and Outside) data processing:

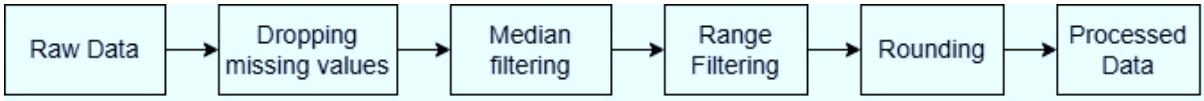


Figure 36: The data processing pipeline for temperature and humidity data

1. Raw data:
    - Raw temperature and humidity values retrieved from the DHT22 sensors.
2. Dropping missing values:
    - Isnan() method was used to filter out missing values.
3. Median Filtering
    - getMedian() method was used over 10 readings to get the middle value in order to reduce noise and reduce the influence of the outliers.
4. Range Filtering
    - Readings are rejected if they are out of the physical range (e.g., temperature < -40°C or > 80°C).
5. Rounding
    - Rounded to 2 decimal places.
6. Processed Data
    - Processed outside and inside humidity and temperature data which are ready to be sent to the cloud database.

HX711 load cell weight data processing:

Figure 37: The data processing pipeline for weight data

1. Raw data:
    - Raw weight values retrieved from the load cell via HX711.

2. Averaging:
    - Takes the average of 5 weight values across a short time span.
3. Dropping missing values:
    - Drops any nan values among the averaged weight values.
4. Median Filtering
    - getMedian() method was used over 10 averaged weight values to get the middle value.
5. Range Filtering
    - Invalid if weight < -1000g or > 5000g.
6. Rounding
    - Rounded to 2 decimal places.
7. Processed Data
    - Processed weight data which is ready to be sent to the cloud database.

After data processing, the final operational firmware formats the processed data as strings, adds them to a FirebaseJson object, and then sends the data to the cloud database via Wi-Fi.

The final operational firmware makes sure that the ESP32 enters a deep sleep every time for 15 minutes after retrieving, processing, and sending the sensor data to Firebase. This was done to minimize energy waste and achieve energy efficiency by turning the ESP32 off when it has no task to perform.

The following table summarizes all the libraries used in both the final and calibration firmware images, along with their respective purposes.

Table 16: Summary of the Arduino libraries used in the firmware

Library	Purpose
Arduino.h	Enables standard Arduino functions for programming
WiFi.h	Enables WiFi connectivity for the ESP32
Firebase_ESP_Client.h	Enables access to the Firebase Realtime database for the ESP32.
Adafruit_Sensor.h	Provides a standard interface for reading

	sensor data.
DHT.h	Allows to retrieve temperature and humidity readings from the DHT22 sensor
time.h	Enables NTP time synchronization for accurate timestamps.
HX711.H	Allows to receieve weight readings from the HX711 load cell amplifier
TokenHelper.h, RTDBHelper.h	Enable token handling and response management for Firebase integration.

3.5.4 Cloud Setup

Firebase was used for all cloud-related operations in the project. The three main Firebase components utilized are summarized in the table below.

Table 17: Summary of the used firebase components

	Firebase Realtime Databse	Firebase Firestore	Firebase Authentication
Purpose	To store and continuously update real-time sensor data from the beehives	To store structured information about registered beekeepers, their apiaries, and the hives under each apiary.	To handle user login and registration for beekeepers and the admin accessing the system
Rationale for choosing	The Firebase Realtime Database's ability to stream and synchronize live data optimally (Li, 2018).	Firestore's ability to handle complex, hierarchical, and user-centric data (Kesavan, 2023).	Firebase Authentication's simplified setup of secure sign-in systems with built-in support for email/password

			(Ranganath Manohar Rane, 2021).
Connected with	Final operation firmware flashed to the ESP32, Beekeeper mobile application	Beekeeper mobile application, Admin web application	Beekeeper mobile application, Admin web application

3.5.5 Queen Detection Model

One of the primary requirements of this project is to automatically detect the presence or absence of the queen bee without disturbing the hive. The **Appendix A: Scientific and Biological Context** section below and the **Chapter 02 | Background Study** section above discuss the critical role of the queen bee in colony health, the adverse effects of queen loss, existing queen detection methods, and previous research conducted on identifying queen presence using bee acoustic signals. The programming languages, development environments and other tools utilized for the queen model implementation is depicted in the figure below.

Figure 38: The programming languages, development environments and other tools utilized for the queen model implementation

The machine learning pipeline implemented for the queen detection model includes the stages depicted in the graph below.

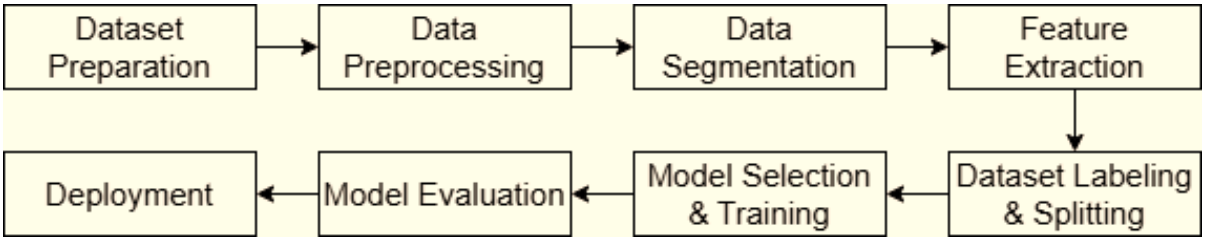


Figure 39: The machine learning pipeline implemented for the queen detection model

This section covers dataset preparation, data preprocessing, data segmentation, feature extraction, dataset labeling and splitting, model selection and training, and model evaluation. The deployment of the model will be discussed separately in the Flask Server section. The following figure shows the Python libraries and frameworks used for various tasks in the queen model implementation.

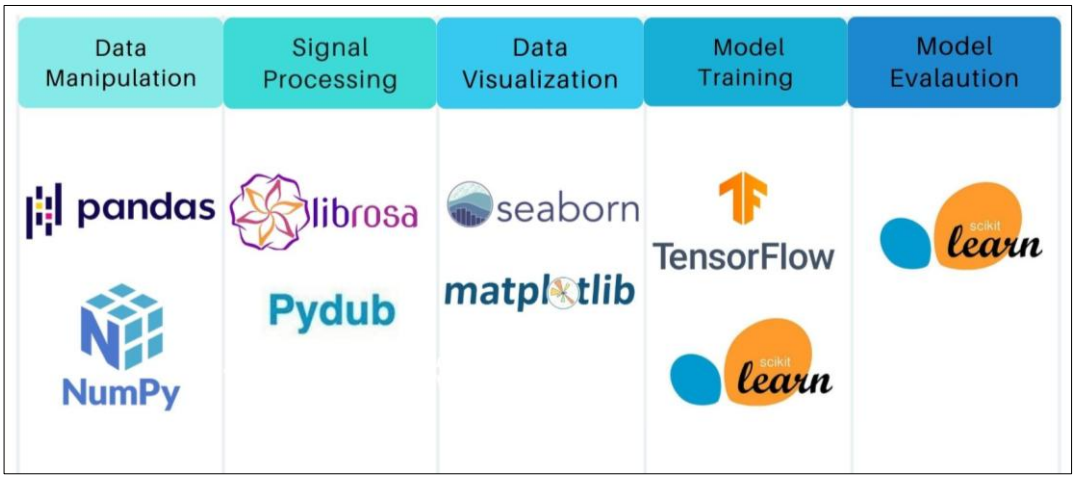


Figure 40: The Python libraries and frameworks used for various tasks in the queen model implementation

Dataset Preparation:

To build an effective classification model, the dataset must include sound recordings from both queenright (with queen) and queenless colonies. Additionally, to ensure the model is generalized and not hive-specific, it is important to create a dataset with samples from multiple hives under varying conditions.

The main source for the dataset is the publicly available dataset created in (Benetos, 2018) , which was based on sound recordings from the Open-Source Beehive (OSBH) project (osbeehives, 2013) and the NU-Hive project (Isidoro, 2018). The following table briefly

summarizes these two projects which the dataset gained its beehive sound recordings from.

Table 18: The summary of the two projects where the dataset gained its beehive sound recordings from

Project	How is the dataset collected?	Unique Characteristics
OSBH Project	The dataset was gathered via a citizen science project that invited the general public to record beehive sounds along with the corresponding hive status (hive is healthy, hive is queenless,etc)	• Different recording setups: People used various microphone types and other devices placed in different positions of the hives, with many background noises in the recordings. • Variety of hives: The data comes from different types of beehives, bee species, climates, and times of the year. • Rich and diverse: This mix of setups and hive conditions makes the dataset special and useful for real-world research.
NU-Hive project	The dataset was collected from two beehives in a controlled environment. Sound recordings were made and labeled based on whether the queen bee was present or not.	• Controlled environment: Unlike the OSBH dataset, NU-Hive recordings come from a consistent setup with a controlled environment. • Labeled hive states: Each recording is marked with the queen bee's presence or absence. • Background sounds: Most external noises are from traffic, car horns, and birds, making the data cleaner and easier to analyze.

The final combined dataset consists of sound recordings from six different hives, with 60% of the samples sourced from the OSBH dataset and the remaining 40% from the NU-Hive dataset. The following table shows the distribution of "queen present" and "no queen" samples for each hive. Some hives contain both types of recordings, while others contain only one.

Table 19: The distribution of "queen present" and "no queen" samples for each hive

Hive Name	Have Queenright sample?	Have Queenless samples?
CF001	✗	✓
CF003	✓	✗
CJ001	✗	✓
GH001	✓	✗
Hive1	✓	✓
Hive3	✓	✓

Additional audio recording samples were then added to the dataset by the author, collected from a single hive in Sri Lanka under the queen present condition.

Data Preprocessing:

The public dataset contained beehive recordings in WAV and MP3 formats. The first step was to extract only the bee sound segments, since the recordings also included various non-bee sounds. Fortunately, the researchers had already labeled the time intervals in which pure bee sounds were heard (marked as “bee”) and those with external noises. These sound intervals were provided in .lab files for each recording. Using Pandas, the .lab files were loaded to extract the start and end times for both bee and non-bee sound segments. Based on these time intervals, only the pure bee sound samples were extracted from the recordings.

The second preprocessing step was to resample all bee sound samples to a common sampling rate. A sampling rate of 22 kHz was chosen to ensure consistency across all recordings. This was done using Librosa’s resample () method.

The third preprocessing step was to convert all audio samples to single-channel (mono) signals in order to ensure uniformity and simplify processing.

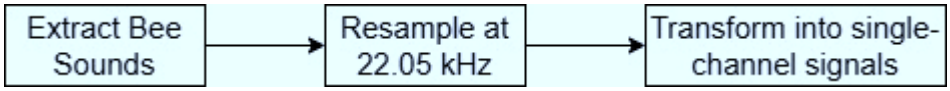


Figure 41: Data preprocessing steps for the queen model

Data Segmentation:

The preprocessed audio samples originally had varying durations. However, for feature extraction and model training, it was necessary to ensure uniformity in sample length. Therefore, each preprocessed audio sample was segmented into 1-second-long clips. The following is the mathematical technique used for audio segmentation.

- If the audio sample contains N samples,
- If the sampling rate of the audio samples are L (L= 22050),

$$\boldsymbol{Number\ of\ Segments}\ (\boldsymbol{N_{segemnts}}) = \lfloor \boldsymbol{N/L} \rfloor$$

Equation 1: Mathematical expression to determine how many segments each audio file should be divided into to obtain 1-second-long samples.

- Each segment is created by,

$$\boldsymbol{Segment_k\ Time\ Interval} = \boldsymbol{x}[\boldsymbol{k} \times \mathbf{22050}: (\boldsymbol{k} + \mathbf{1}) \times \mathbf{22050}]$$

Equation 2: Mathematical expression to determine the time interval for each interval

Where:

- x is the full audio signal,
- k is the segment number: 1, 2, 3, …, $N_{segemnts}$

Feature Extraction:

Different types of features were extracted and tested with various models to determine the best combination. The following table summarizes the feature extraction methods considered in this study.

Table 20: Summary of the extracted features

Feature	Feature	Description	Library	Used	in	the

Name	Type		Used	Final Model?
MFCC	Time-Frequency	MFCCs are coefficients that capture the spectral envelope of sound based on human hearing perception .	Librosa	Yes
Mean STFT	Frequency-domain	Short-Time Fourier Transform averaged over time	Librosa	No
Mean CQT	Time-Frequency	Constant-Q Transform averaged across time	Liborsa	No
Mel Spectrogram	Time-Frequency	Represents the distribution of power across Mel frequency bands	Librosa	No
Zero Crossing Rate	Time-domain	Measures how often signal crosses zero amplitude	Librosa	No

Data Labelling and Splitting:

The dataset consists of various extracted features that needed to be labeled and split for model training. All the records containing features extracted from queen-present segmented audio samples were labeled as 1 (representing the queenright scenario), while the rest were labeled as 0 (representing the queenless scenario). Two approaches were used for data splitting:

1. **Holdout Validation** → In this method, a certain percentage of the data was used for training while the remaining portion was used for testing (GeeksForGeeks, 2025). Various combinations of train-test splits were experimented with, such as 70–30 and 60–40. The train_test_split() method from sklearn.model_selection was used for this purpose.

2. **K-Fold Cross Validation** → In this approach, the entire dataset was split into k equal subsets (folds). In each iteration, one-fold was used for testing while the remaining k−1 folds were used for training. This process continued until every fold had been used once for testing (GeeksForGeeks, 2025). In this project, 5-fold cross-validation was used considering the size of the dataset. The KFold class from sklearn.model_selection was used

to implement this.

Model Selection & Training:

The primary model selected for the experiments was the deep learning-based Convolutional Neural Network (CNN) for queen bee audio classification, following the approach used in previous research on the same dataset by Orłowska (2022). In addition to CNN, other machine learning models such as Extreme Gradient Boosting and Support Vector Machine (SVM) were also evaluated. Extreme Gradient Boosting was chosen for its strong performance on structured audio features (Eduardo Fonseca, 2017), while SVM was considered due to its effectiveness in handling high-dimensional feature spaces typically found in audio data (Jia-Ching Wang, 2006).

Some important experiments that were carried out under model training are depicted in the tables below.

Table 21: The experiments carried out with SVM

Experiment Index	Extracted Features	Validation Type	Methodology for splitting	Kernal Function	Has class imbalance handled?
SVM_1	Mean MFCCs	Holdout Validation	70% for training and 30% for testing	Linear	No
SVM_2	Mean MFCCs	K-Fold Cross Validation	5- Folds	Linear Kernal	No
SVM_3	Mean MFCCs	K-Fold Cross Validation	5- Folds	RBF Kernal	No
SVM_4	Mean STFTs	K-Fold Cross Validation	5-Folds	RBF Kernel	No

Table 22: The experiments carried out with Extreme Gradient Boosting (XGBoost)

Experiment Index	Extracted Features	Validation Type	Methodology for splitting	With tuned hyperparameters?	Has class imbalance handled?
XGB_1	Mean MFCCs	K-Fold Cross Validation	5-Folds	No	No
XGB_2	Mean MFCCs	K-Fold Cross Validation	5-Folds	Yes	No

To tune the parameters for the XGBoost classifier, **GridSearchCV** was used. **GridSearchCV** was employed to search for the best hyperparameters through 5-fold cross-validation. The following hyperparameters were defined and tuned for the XGBoost classifier:

Table 23: Tuned hyperparameters for the XGBoost classifier

Hyperparameter	Definition
learning_rate	Step size shrinking.
n_estimators	Number of boosting rounds (trees).
max_depth	Maximum depth of the decision tree.
subsample	Specifies the fraction of the training data to randomly sample for each boosting round
colsample_bytree	specifies the fraction of features (columns) to randomly sample each tree.

Table 24: The experiments carried out with Convolutional Neural Networks (CNN)

Experiment Index	Extracted Features	Validation Type	Methodology for splitting	Has class imbalance handled?
CNN_1	Mean	Holdout	70% for training	No

	MFCCs	Validation	and 30% for testing	
CNN_2	Mean MFCCs	K-Fold Cross Validation	5-Folds	No
CNN_3	Mean MFCCs	K-Fold Cross Validation	5-Folds	Yes
CNN_4	Mean STFTs	K-Fold Cross Validation	5-Folds	No

The initial CNN model architecture and hyperparameters are given below. These values differ only slightly in the final model. Since the main focus of the study was on experimenting with different feature combinations, the neural network architecture and hyperparameter tuning for the CNN model were not explored in depth.

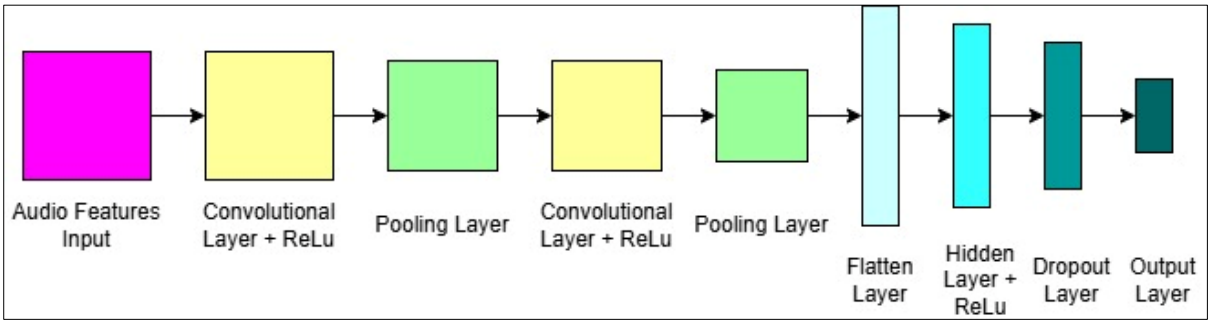


Figure 42: The CNN architecture

Table 25: The tuned hyperparameters in CNN

Name	Value
Optimizer	Adam
No Training Epochs	20
Loss Function	Categorical Cross Entropy

In the CNN_3 experiment, to address class imbalance, the class weights approach was applied. This penalizes misclassifications of the minority class more heavily during training, helping the model become less affected by the class imbalance in the dataset.

Model Evaluation:

Various evaluation metrices were used in order to evaluate the performance of the implemented ML models. The table below summarizes the used evaluation metrices.

Table 26: Evaluation Metrices

Performance Metric	Acronym	Purpose	Equation
Precision (Weighted Avg)	P	Measures the proportion of correct positive predictions (Queen or No_Queen) across both classes, weighted by class support.	$\boldsymbol{R} = \frac{(\mathbf{PQ} \times \mathbf{SQ}) + (\mathbf{PN} \times \mathbf{SN})}{\boldsymbol{T}}$ **Equation 3:** The equation for the precision (weighted average)
Recall (Weighted Avg)	R	Measures the proportion of actual positive samples correctly identified across both classes, weighted by class support.	$\boldsymbol{R} = \frac{(\mathbf{RQ} \times \mathbf{SQ}) + (\mathbf{RN} \times \mathbf{SN})}{\boldsymbol{T}}$ **Equation 4:** The equation for the recall (weighted average)
F1-score (Weighted Avg)	F1	Provides the harmonic mean of precision and recall for both classes, weighted by class support.	$\boldsymbol{F1} = \frac{(\mathbf{F1}\boldsymbol{Q} \times \mathbf{SQ}) + (\mathbf{F1}\boldsymbol{N} \times \mathbf{SN})}{\boldsymbol{T}}$ **Equation 5:** The equation for the f1-score (weighted average)
Accuracy	Acc	Measures the overall proportion of correct predictions (both positive and negative) among all samples.	$\boldsymbol{Acc} = \frac{\mathbf{T}\boldsymbol{N} + \boldsymbol{TP}}{\boldsymbol{T}}$ **Equation 6:** The equation for accuracy

In the above table:

- PQ = Precision for Queen Class

- PN = Precision for No_Queen class
- RQ = Recall for Queen class
- RN = Recall for No_Queen class
- F1Q= F1-score for Queen class
- F1N = F1-score for No_Queen class
- SQ = Number of samples for Queen class in the testing data.
- SN = Number of samples for No_Queen class in the testing data.
- T = Total samples in the testing data = S_Q+S_N
- TP = True Positives (correctly predicted Queen samples)
- TN = True Negatives (correctly predicted No_Queen samples)

The following table shows the precision, recall, and F1-score equations calculated separately for each class (Queen and No_Queen):

Table 27: The precision, recall, and F1-score equations calculated separately for each class

Metric	Equation per Class
Precision	$\boldsymbol{Pi = \frac{\mathrm{TP}i}{(TPi + FPi)}}$ **Equation 7:** The equation for precision
Recall	$\boldsymbol{Ri = \frac{\mathrm{TP}i}{(TPi + FNi)}}$ **Equation 8:** The equation for recall
F1-score	$\boldsymbol{F1i = 2 \times \frac{\mathrm{P}i \times \mathrm{R}i}{(Pi + Ri)}}$ **Equation 9:** The equation for f1-score

In the above table:

- *i* represents the class index (Queen or No_Queen).
- TP*i*, FP*i*, FN*i* are the true positives, false positives, and false negatives for class *i*.

For metrics visualization, confusion matrices and classification reports from sklearn.metrics

were used.

The evaluation results obtained from various ML model training experiments are shown in the table below.

Table 28: The evaluation results obtained from various ML model training

Experiment Index	F1score	Recall	Precision	Accuracy
SVM_1	0.91	0.91	0.91	90.62%
SVM_2	0.89	0.89	0.89	88.69%
SVM_3	0.99	0.98	0.99	98.38%
SVM_4	0.96	0.97	0.97	96.43%
XGB_1	0.99	0.98	0.99	98.40%
XGB_2	1.00	1.00	1.00	99.71%
CNN_1	0.99	0.99	0.99	99.26%
CNN_2	0.99	0.99	0.99	99.16%
CNN_3	0.99	0.99	0.99	99.43%
CNN_4	0.96	0.96	0.96	95.63%

All three models under both validation methods performed well on the dataset for the queen detection scenario. This demonstrates the success and effectiveness of the dataset for queen detection. Experiments using mean STFTs, such as SVM_4 and CNN_4, produced slightly lower results compared to those using MFCCs. The higher results in cross-validation indicate that the model is able to classify the queen detection task effectively and consistently. A comparatively lower performance was observed for the SVM when using the linear kernel as the kernel function. It is also noticeable that CNN and XGBoost outperformed the SVM. While XGBoost and CNN delivered similar accuracy during cross-validation, CNN is preferable because it can handle larger datasets and more complex patterns, making it a more robust choice for future scalability. Applying class imbalance handling to the CNN slightly enhanced the model's performance, achieving up to 99.43% accuracy." The overall confusion matrix across all the folds and the average learning curves across all the folds for this scenario (CNN_3) are shown below.

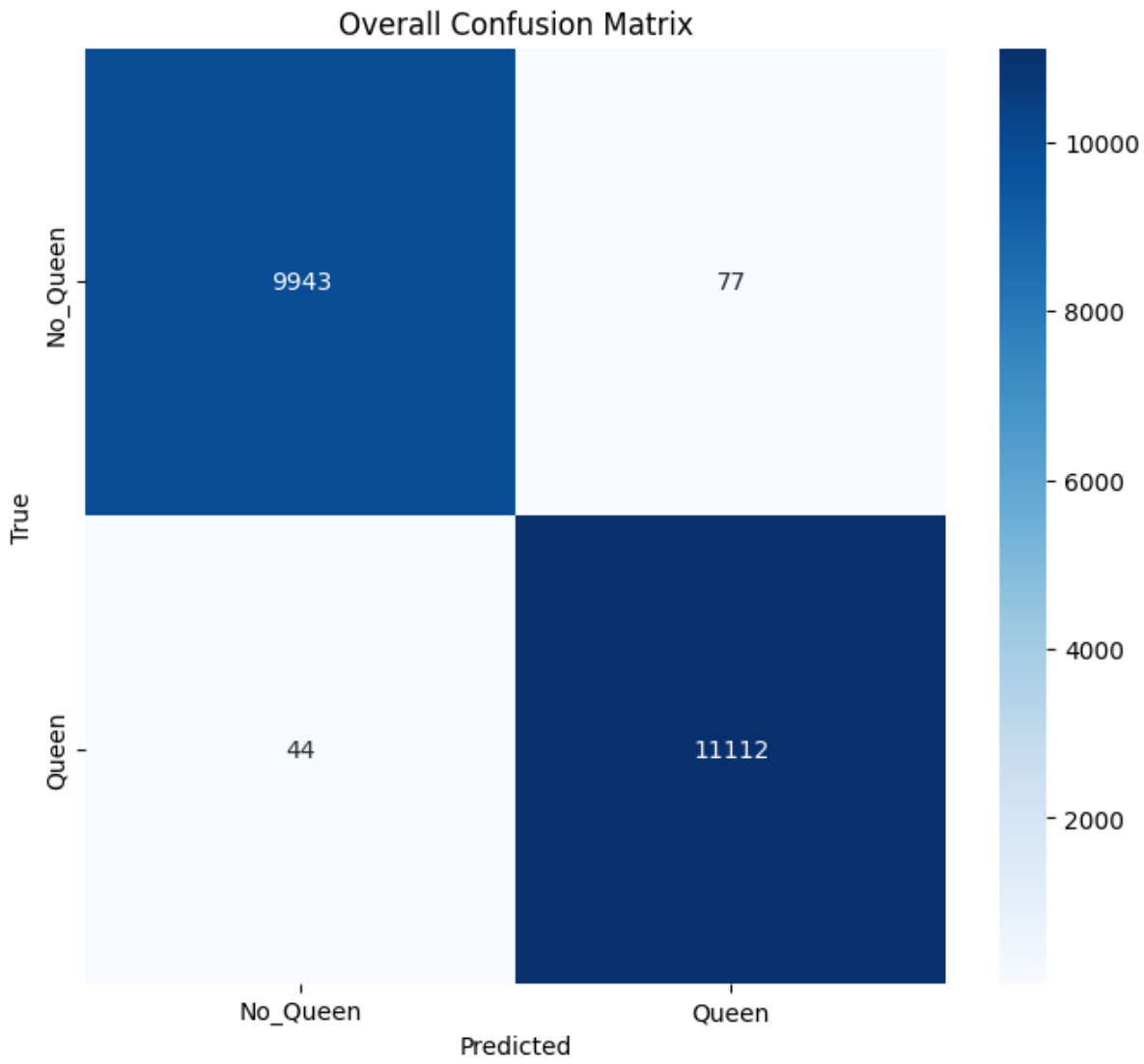


Figure 43: The overall confusion matrix across all the folds for CNN_4 experiment

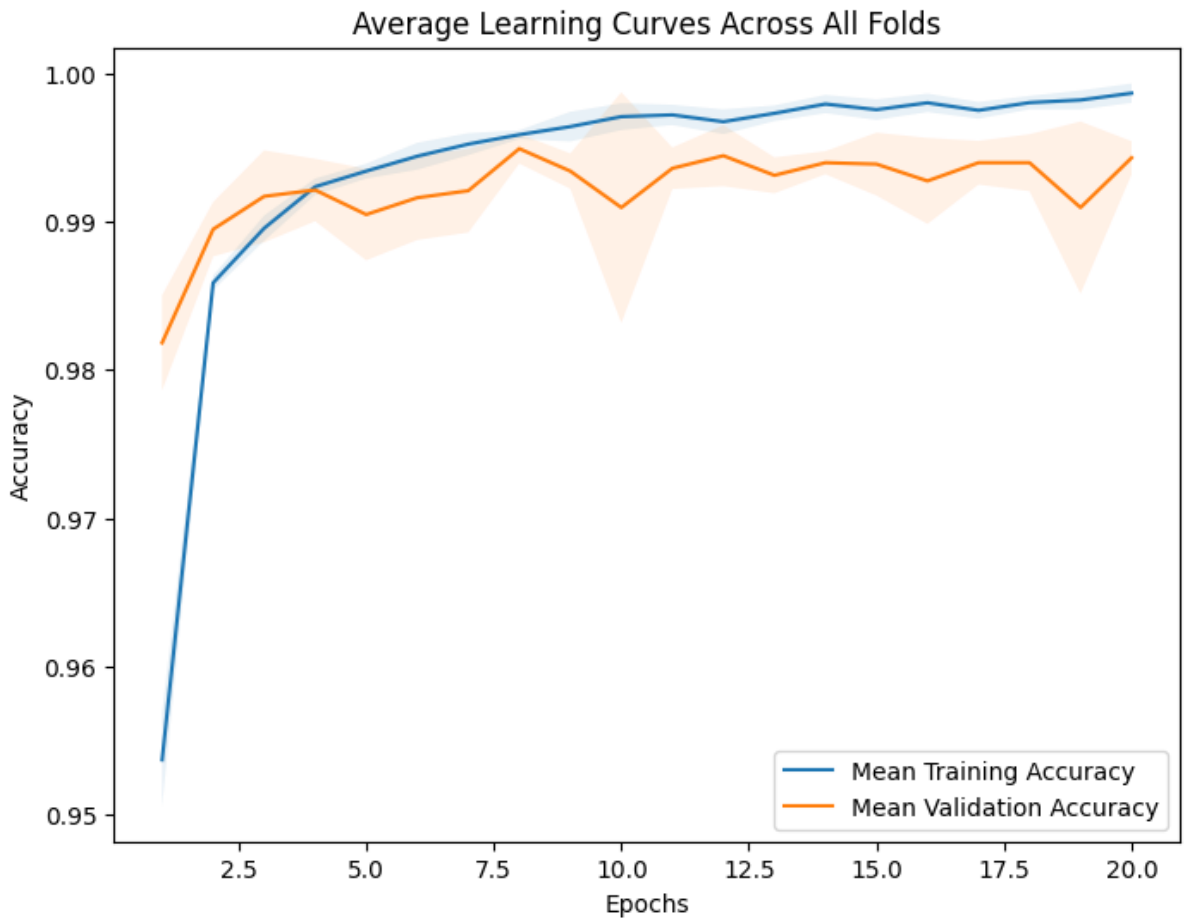


Figure 44: The average learning curves across all the folds for the CNN_3 scenario

Model deployment:

The model trained under CNN_3 experiment was saved in H5 format for later integration with the backend server. This model was trained using extracted MFCCs and applied class weight adjustments to mitigate the impact of class imbalance on the model's performance.

3.5.6 Health Analysis Model

Another crucial requirement of this project is to analyze the hive's health status and detect any abnormal behavior remotely. As discussed in the background section, this study explores hive temperature and humidity patterns to assess hive health by detecting anomalies in normal patterns using unsupervised anomaly detection techniques. The programming languages, development environments, and other tools used for implementing the health model are depicted in the figure below.

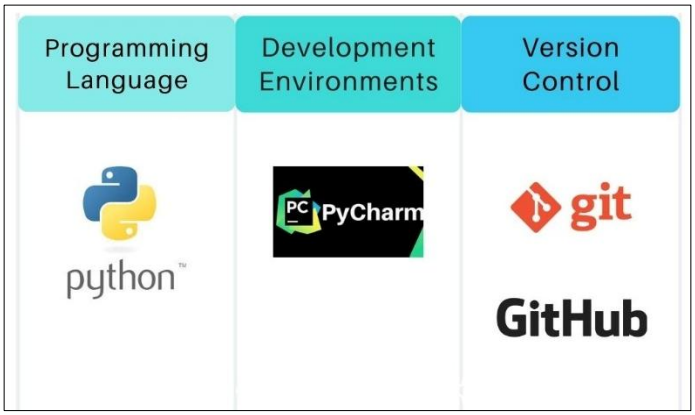


Figure 45: The programming languages, development environments, and other tools used for implementing the health model

The machine learning pipeline implemented for the hive health analysis model includes the stages depicted in the graph below.

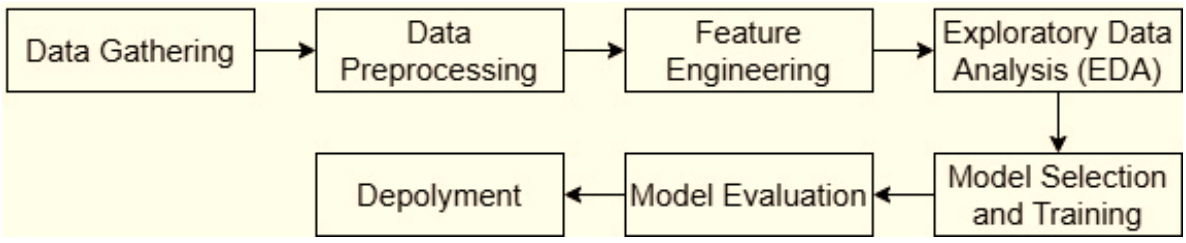


Figure 46: The machine learning pipeline implemented for the hive health analysis model

This section covers data gathering, data preprocessing, feature engineering, model selection and training, and model evaluation. The deployment of the hive health model will be discussed separately in the Flask Server section. The following figure shows the Python libraries and frameworks used for various tasks in the hive health model implementation.

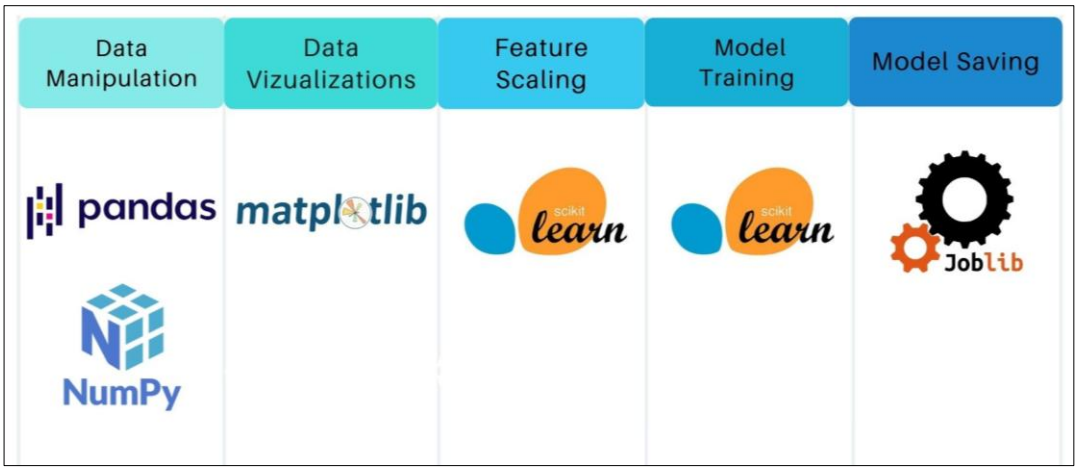


Figure 47: The Python libraries and frameworks used for various tasks in the hive health model implementation.

Data Gathering:

The implemented IoT module, along with the power supply, was placed in two different hives—one located in Rajagiriya and the other in Athurugiriya—as described in the *Hardware Setup* section above. The purpose was to collect temperature and humidity data from both the hive inside and hive outside. Both experiments focused on Asian honeybees, a species that has received limited attention in prior research, as noted in the *Literature Review* section.

In the Rajagiriya beehive, the device was deployed continuously for 17 hours, while in the Athurugiriya beehive, it was placed over a period of two weeks (not continously). All gathered data was stored in the Firebase Realtime Database, as explained in the *Cloud Setup* section. The collected dataset includes readings for the following four parameters:

- Inside temperature
- Inside humidity
- Outside temperature
- Outside humidity

Data Preprocessing:

The dataset from the Rajagiriya beehive contained invalid data for an entire week, as the device was removed from the hive and kept outside due to a technical issue. The first preprocessing step involved removing all readings recorded during this period. Next, both the datasets were checked for missing (NaN) values using the pandas.DataFrame.isna() method, but no such

values were found. After that, the dataset was examined for duplicate rows using the pandas.DataFrame.duplicated() method, and again, no duplicates were detected. Finally, as a preprocessing step, the timestamp strings were converted to datetime format and sorted by time using pandas.DataFrame.to_datetime() and pandas.DataFrame.sort_values() methods respectively.

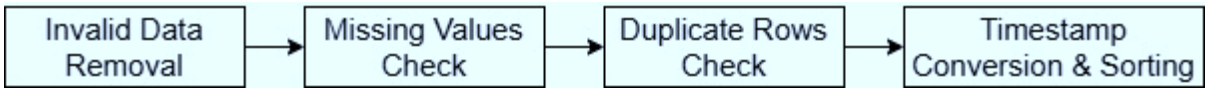


Figure 48: Data preprocessing steps carried out for the health model

Feature Engineering:

Under feature engineering, various new features were introduced from the existing ones. All the newly introduced features are summarized in the table below.

Table 29: Summary of the newly introduced features

New Feature	New Feature Type	Used Existing Features	Logic
Humidity Differential	Double	Inside Humidity, Outside Humidity	Inside Humidity - Outside Humidity
Temperature Differential	Double	Inside Temperature, Outside Temperature	Inside Temperature - Outside Temperature
Hour	Integer	Timestamp	Retrieved hour (1-24) from the timestamp values

Exploratory Data Analysis (EDA):

Exploratory Data Analysis results for the Rajagiriya beehive dataset are shown below. These results have already been discussed in detail by the author in a previous publication (Liyanage, 2024). Therefore, in this study, the results will be explored briefly.

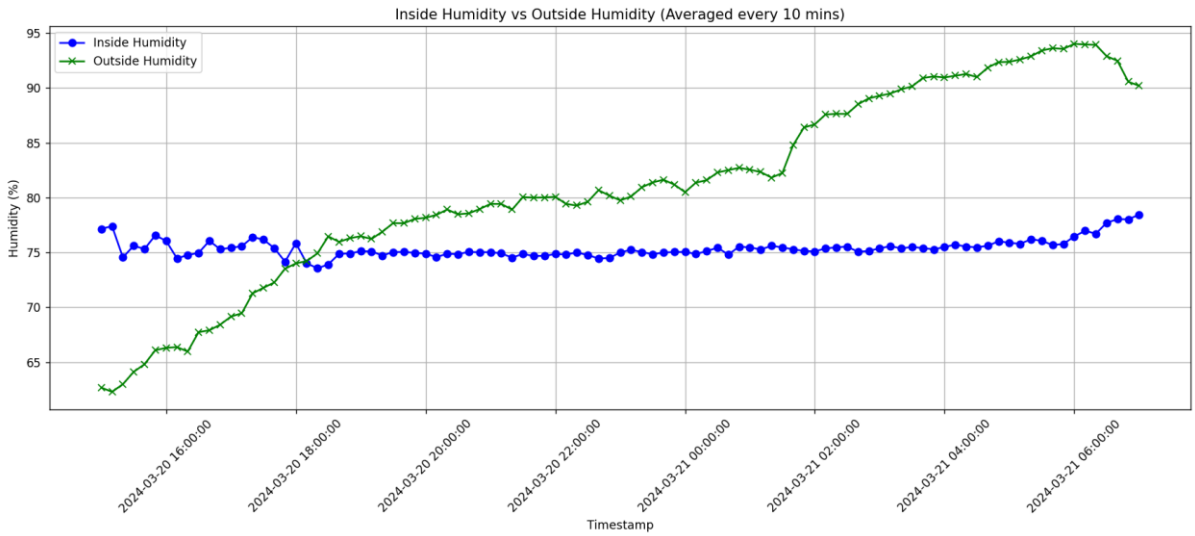


Figure 49: How inside and outside humidities have changed with time in the Rajagiriya beehive

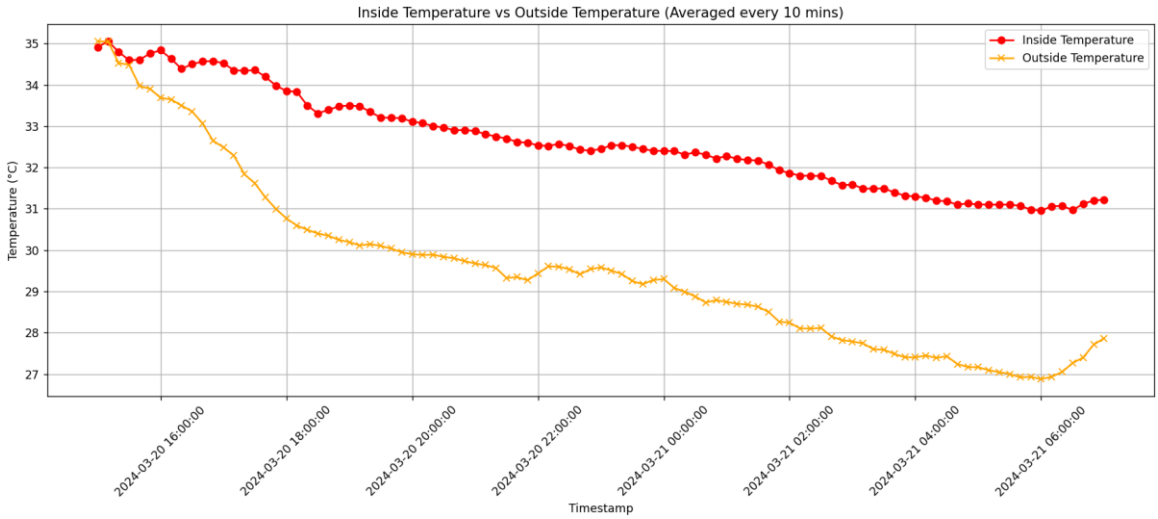


Figure 50: How inside and outside temperatures have changed with time in the Rajagiriya beehive

The graphs clearly show how bees regulate the inside temperature and humidity within an optimal range although the outside temperature and humidity values change significantly over the 17 hours. This supports the findings of previous research, which discussed how a healthy beehive can regulate its internal temperature and humidity regardless of external environmental changes as shown in the above **Chapter 02 | Background Study** chapter.

To analyze the general behavior of the Athurugiriya hive throughout the day, data was collected over a period of two weeks. For each hour of the day (e.g., 0,1,….23), the average temperature and humidity values were calculated across all the days. The results are depicted in the plots

below. It is evident that the same observation made with the previous hive dataset applies here as well: the bees regulate the internal temperature and humidity of the hive effectively, regardless of external environmental conditions. However, the exact values differ slightly between the two hives.

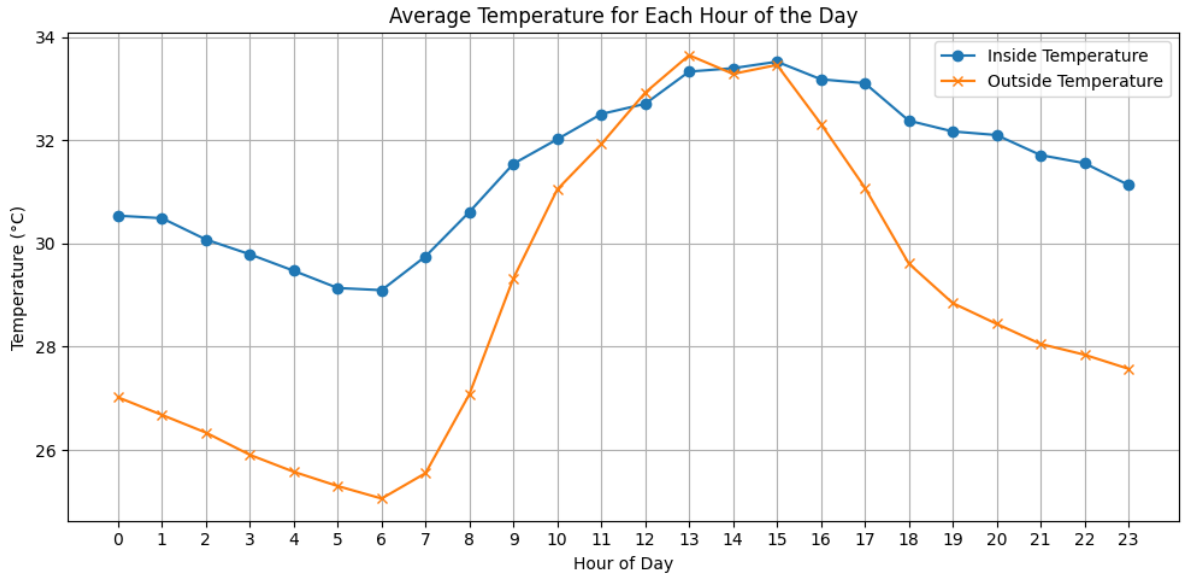


Figure 51: The average inside and outside temperatures for each hour of the day in the Athurugiriya beehive

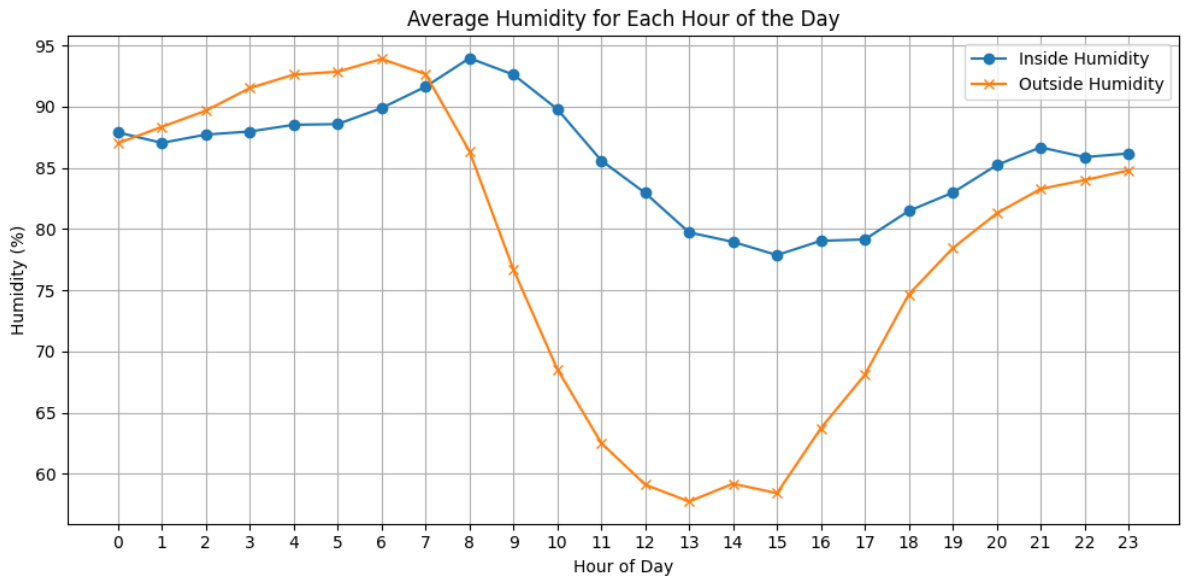


Figure 52: The average inside and outside humidities for each hour of the day in the Athurugiriya beehive

From the plot results, one thing is clear: although the bees regulate internal temperature and humidity, these values can still be significantly influenced by external temperature and

humidity conditions.

The following plots show how the inside and outside temperatures, inside and outside humidities, as well as the temperature and humidity differentials changed over the two-week period.

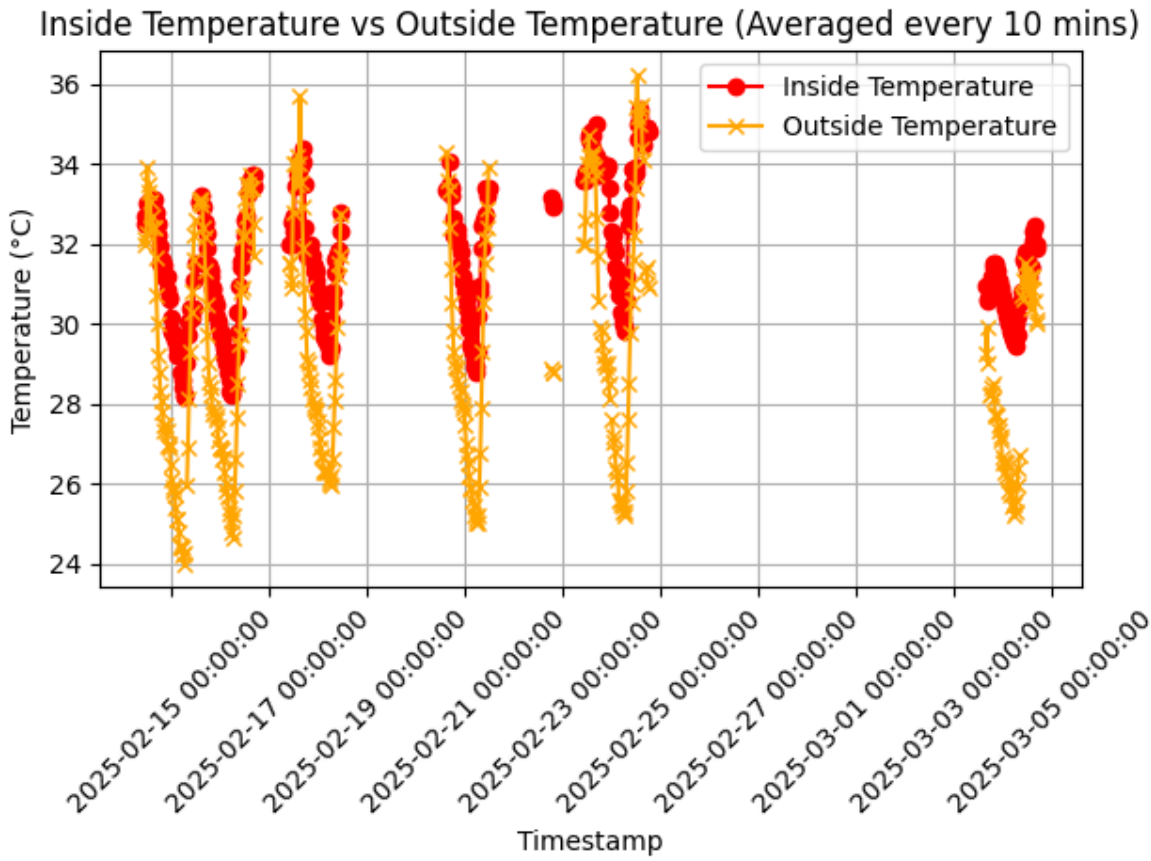


Figure 53: The variation of inside and outside temperatures across 2 weeks of time in the Athurugiriya beehive

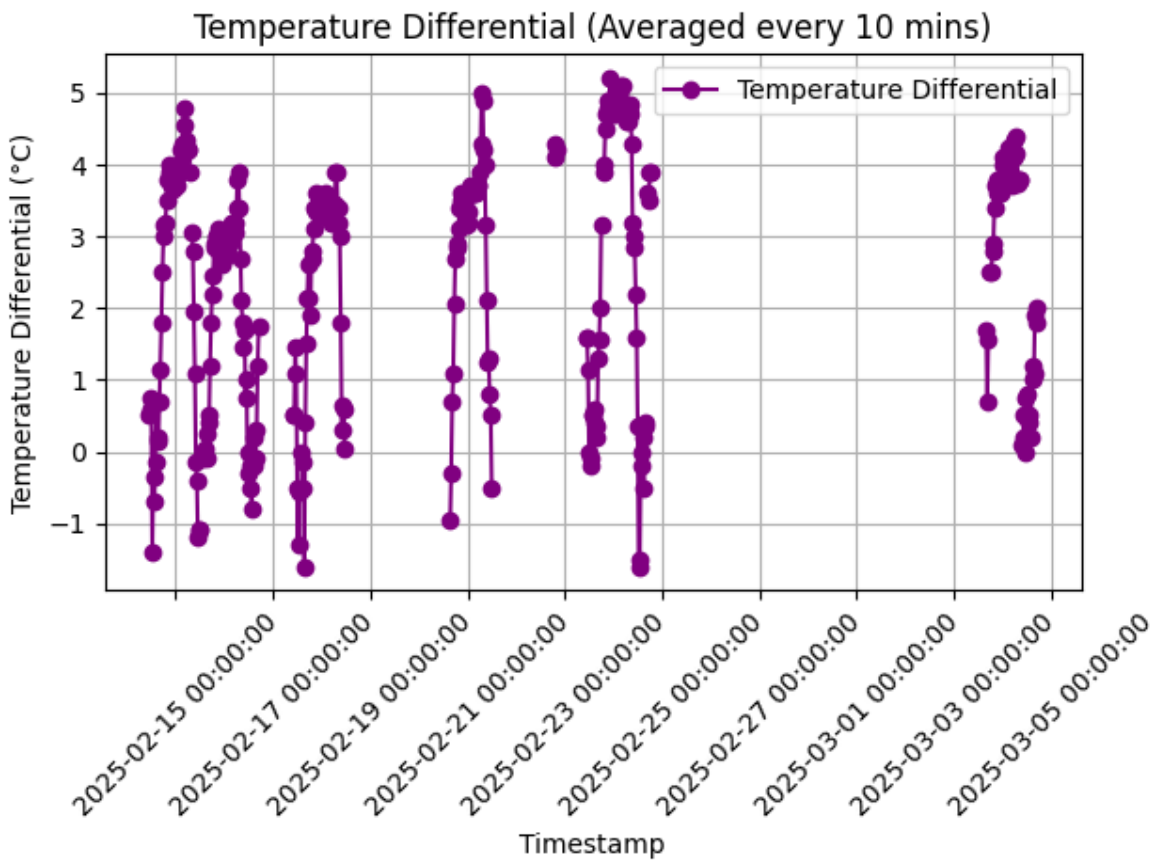


Figure 54: The variation of the temperature differential across 2 weeks of time in the Athurugiriya beehive

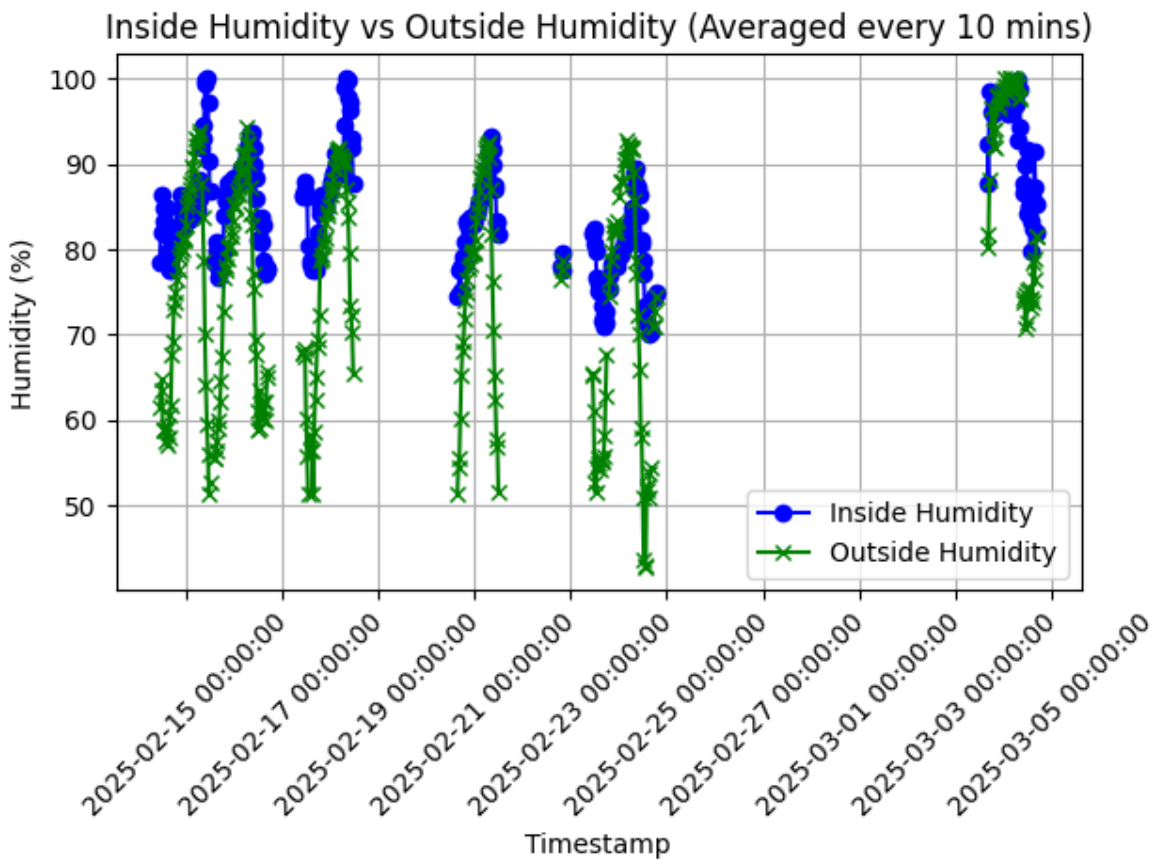


Figure 55: The variation of inside and outside humidities across 2 weeks of time in the Athurugiriya beehive

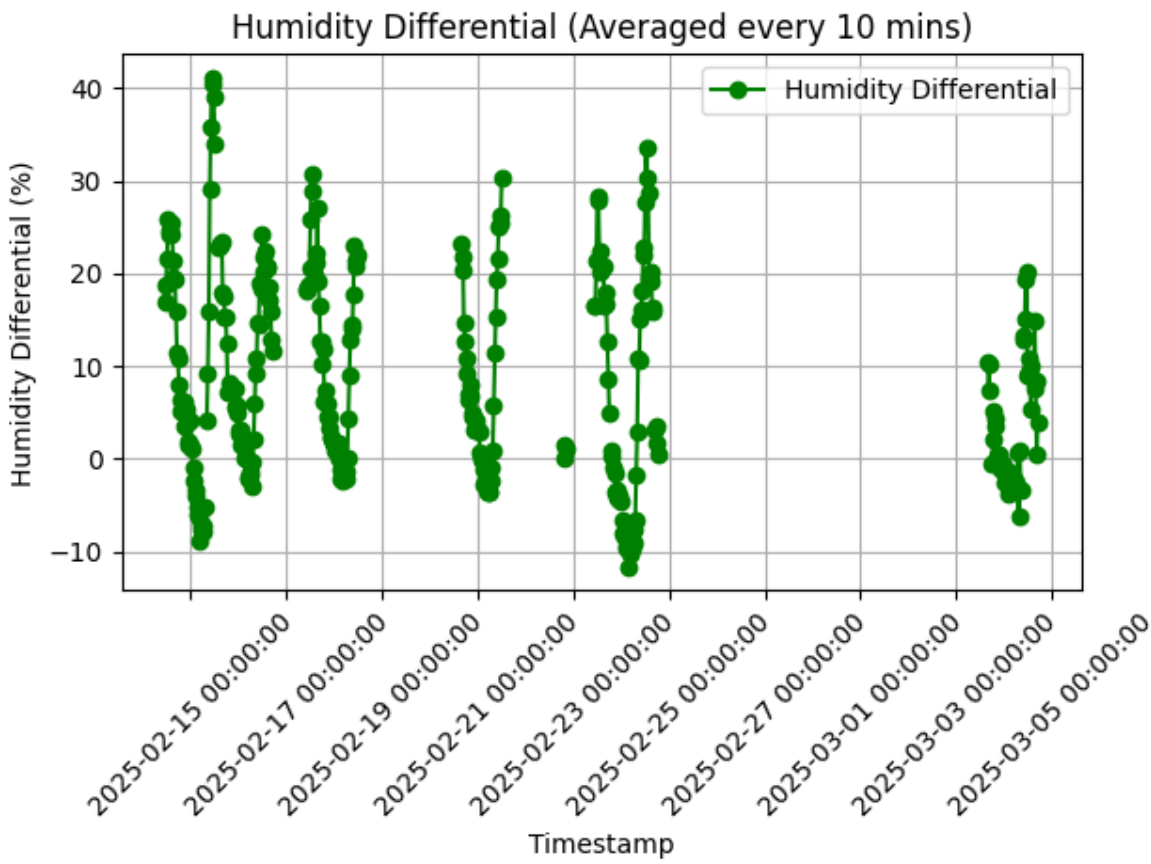


Figure 56: The variation of the humidity differential across 2 weeks of time in the Athurugiriya beehive

The results above show that the daily inside humidity and temperature values have a consistent daily cyclical pattern across all the days. As this hive is a healthy and strong one, it is clear that these are the temperature and humidity patterns of a healthy hive. Any deviations from these normal daily cyclical patterns can be an indication of unhealthy or weak hives, as weak hives are unable to regulate these patterns in the same way a strong colony with healthy bees does. Based on this hypothesis, it is evident that any anomalies noticed in the temperature and humidity values—deviating from those of a healthy hive—could indicate underlying issues within the hive and suggest potential unhealthiness in the colony.

Model Selection and Training:

From the experimental results obtained in the Exploratory Data Analysis section above, it is clear that the model should be an unsupervised novelty detection model, trained on healthy hive metrics, so that it can identify any anomalies deviating from normal behavior as potentially unhealthy. One-Class SVM was selected as the model, as (Shawe-Taylor, 2015)

highlighted its effectiveness in novelty detection problems similar to this scenario. For model training, the Athurugiriya dataset was used, as the other dataset spans only 17 hours and contains limited data.

Four features were selected to train the model. They are:

1. Inside Temperature
2. Inside Humidity
3. Humidity Differential → Beacause outside humidity can influence the inside humidity patterns as discussed above.
4. Temperature Differential → Beacause outside temperatue can influence the inside temperature patterns as discussed above.
5. Hour of the Day → Because time of the day can also influence the inside temperature and humidity patterns.

In this study, the Athurugiriya dataset—which contains only healthy samples—was divided into two parts: 80% was used for training the model, and the remaining 20% for testing. This approach is uncommon for novelty detection problems, where typically the entire healthy dataset is used for training. However, this split was necessary due to the unavailability of other datasets from different hives to test the model.

The values set for the hyperparamters for the One Class SVM model training is given below.

Table 30: The values set for the hyperparameters for the One Class SVM model training

Hyperparameter	Value
Kernel	RBF (Radial Basis Function)
gamma	auto
nu	0.05 (this means 5% of the training data are set to be anomalies)

Model Evaluation:

After training the One-Class SVM model, the decision_function() and predict() methods were applied to both the training and testing healthy hive samples.

Table 31: Overview of the decision_function() and predict() methods

Function	Definition	Meaning of the Values
Decision_function ()	The distance of each data point from the hyperline to the trained model.	Positive value means it's a normal/healthy scenario. Negative value means it's anomalous or unhealthy scenario.
Predict()	Gives classification - whether the data point belongs to healthy or anomalous scenarios.	1 for normal scenario. -1 for anomalous scenario.

To evaluate the healthy test samples, the false positive rate was used. The false positive rate for the test samples was 6.45%, meaning that only 6.45% of all the healthy test samples were misclassified as -1 (unhealthy or anomalous) when the default threshold was used.

Then, a custom threshold was calculated for the model to achieve a 1% false acceptance rate for the healthy test samples. This threshold was determined to be -0.1771.

The plot below visualizes the anomaly scores for each data point in both the training and testing healthy hive samples. The red line represents the default threshold, while the green line represents the custom threshold of -0.1771. The majority of data points stand above both thresholds, which is an indication that the model correctly classified most healthy samples.

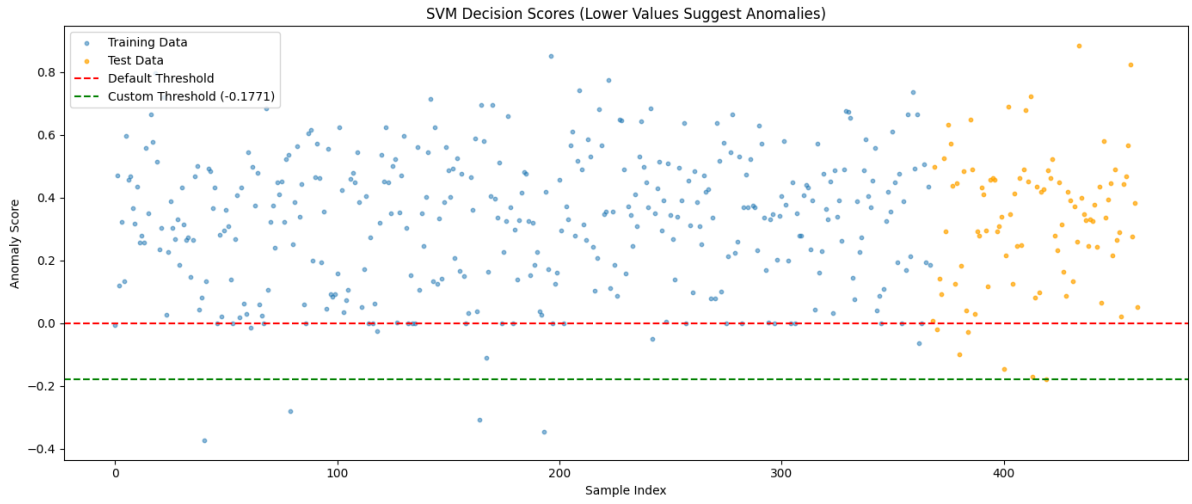


Figure 57: The anomaly scores for each data point in both the training and testing healthy hive

samples

In addition, a synthetic dataset was created based on the healthy dataset, where the feature ranges were selected outside of the healthy range to simulate unhealthy (anomalous) data points. The plot of these synthetic data points and their anomaly scores is shown below. All points are below the custom threshold, indicating that the model successfully classified these samples as anomalous.

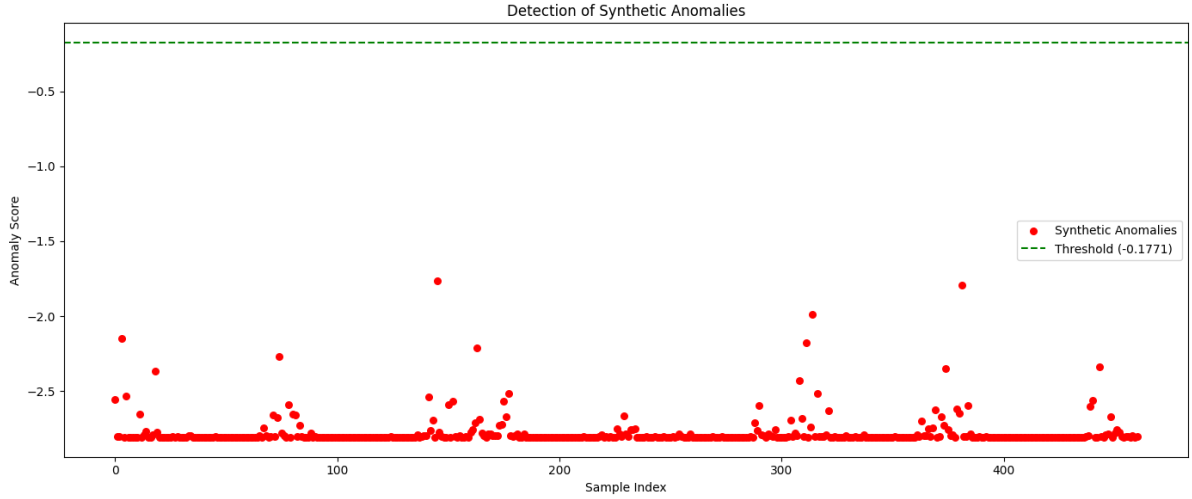


Figure 58: The synthetic data points and their anomaly scores

The model evaluation is now complete. Any data point with a score above the threshold is classified as a healthy scenario. Conversely, if the score is below the threshold, the data point is classified as an anomaly.

3.5.7 Backend Server

Both the health model and the queen model were deployed on the backend server that was implemented. The backend server acts as an intermediary between the trained models and the mobile application. The framework, development environments, and other tools used to implement the backend server are as follows:

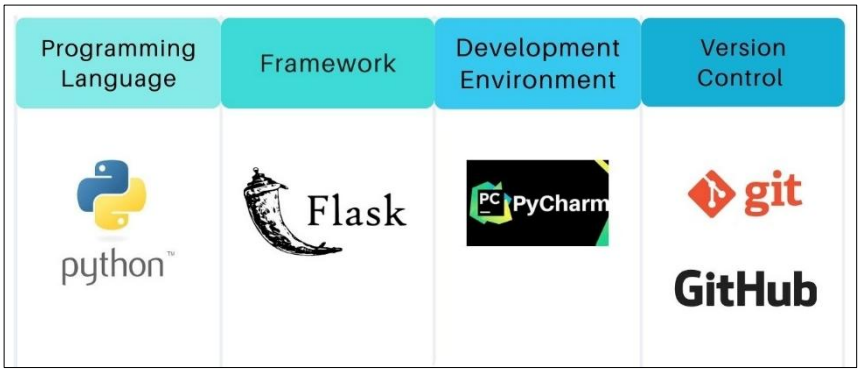


Figure 59: The framework, development environments, and other tools used to implement the backend server

The fask server has two API endpoints. One to process hive health analaysis an the next to detct queen's absensce or presense.

The below graph shows the workflow of the /predict_queen endpont which takes the audio sample sent from the mobile application as a HTTP request, validates the avaiability and the format of the audio sample, preprocesses the audio (resample the audio at 22.05kHz and transform to single channel), splits the audio to 1 second long chunks, extracts MFCCs from each chunk, scales the features, reshapes the features for the model's input , gets the prediction result from the model for each chunk, and finally sends the most predicted class to the mobile application as a HTTP response

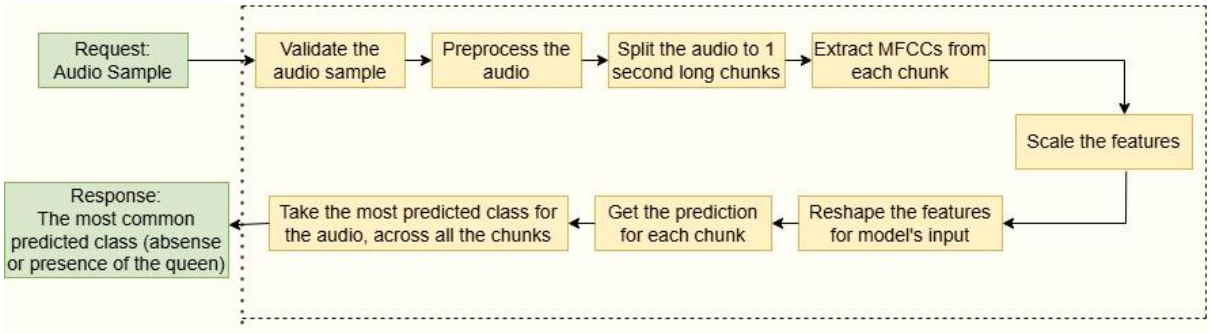


Figure 60: The workflow of the /predict_queen endpoint

The below graph shows the workflow of the /analyze_health endpoint which takes the latest sensor data sent from the mobile application as a HTTP request, validates the data in the request, extracts the required features for the model from the request data (inside temperature, inside humdity, hour, temperature differential, humidity differential), scales the features, gets the decision scores for each record using the trained health analysis model, checks whether each record is an anamoly or not based on the decision scores, counts the number of anamaolies, deteremines whether the majority records are classified as anomalies or not, and finally the sends the majority prediction (whether an anamaly or not) back to the mobile app as the response.

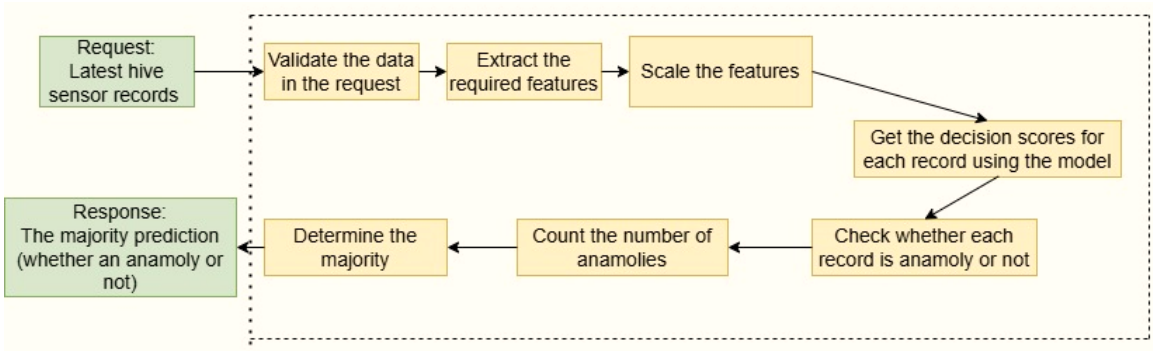


Figure 61: The workflow of the /analyze_health endpoint

3.5.8 Beekeeper Mobile App

The programming language, framework, development environments, and other tools used to implement the beekeeper mobile application are as follows:

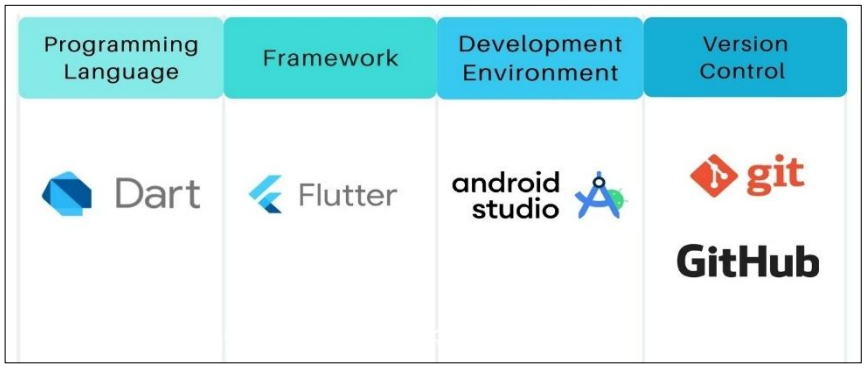


Figure 62: The programming language, framework, development environments, and other tools used to implement the beekeeper mobile application

The following table summarizes all the Dart packages used in the beekeeper mobile application implementation.

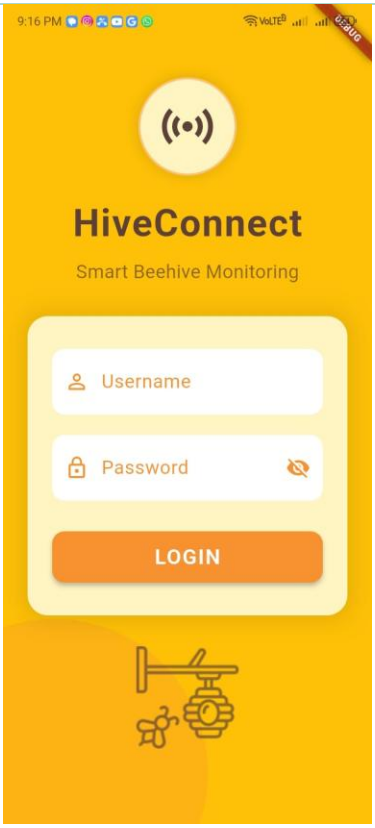
Table 32: The Dart packages used in the beekeeper mobile application

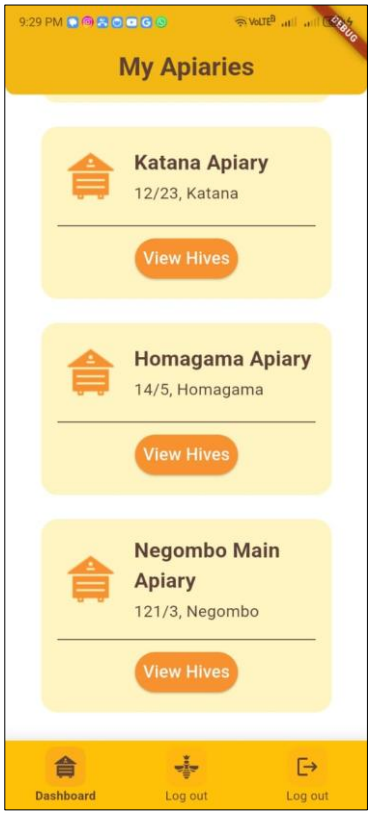
Dart Package	Why used in the project?
firebase_core	To initiliaze and connect the app to firebase services
firebase_auth	To enable user authentication with firebase
percent_indicator	To display live weight value in a circular percentage indicator
fluttertoast	To display short pop up messages
firebase_database	To enable interaction with the firebase realtime database
syncfusion_flutter_charts	To display past hive metrices in line charts
cloud_firestore	To enable interaction with the firebaase firestore
intl	To internationalize timestamps in the realtime database
path_provider	To locate the saved recorded audio in the queen detection interface
record	To record bee sounds to detect the queenbee's presence or absense
audioplayers	To play the recorded bee sounds
http	To comminicate with the backend server for queen detection and

	health analysis tasks

The table below maps each of the implemeneted mobile application's interfaces with the functional requirments defined in the requirement analysis section above.

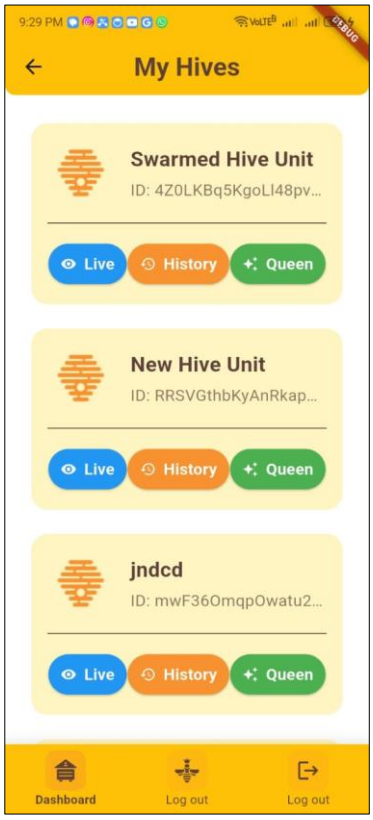
Table 33: Mapping mobile interfaces with requirements

Interface	**Mapped functional requirements**
 **Login Screen**	• The system must allow the beekeepers to log in successfully with the correct credentials.



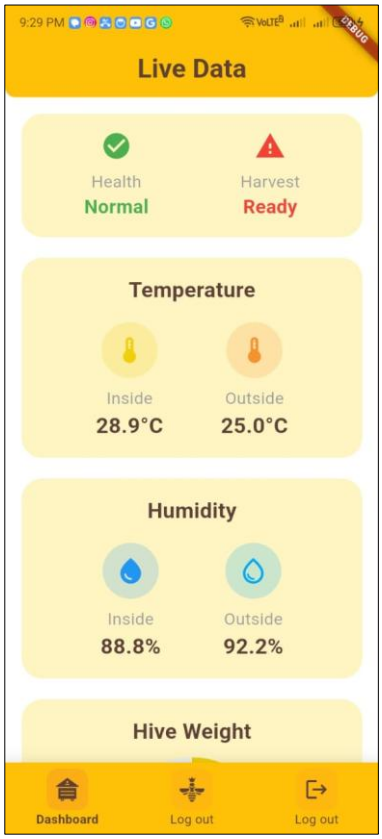
Apiary Screen

- The system must display only the apiaries that belong to the particular beekeeper after logging in.



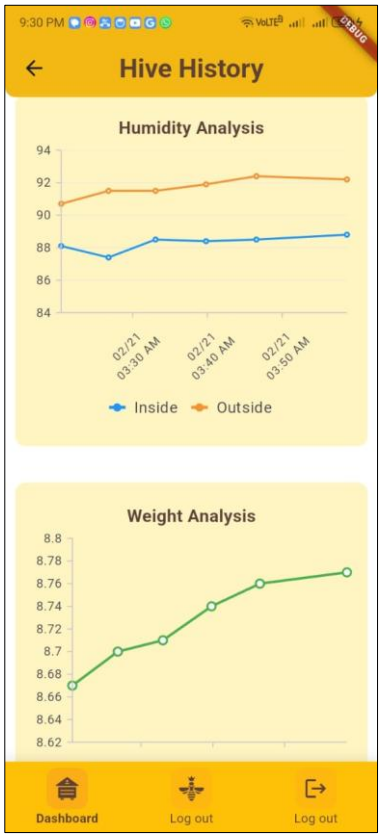
Hives Screen

- The system must display only the specific hives that belong to the selected apiary after choosing an apiary.



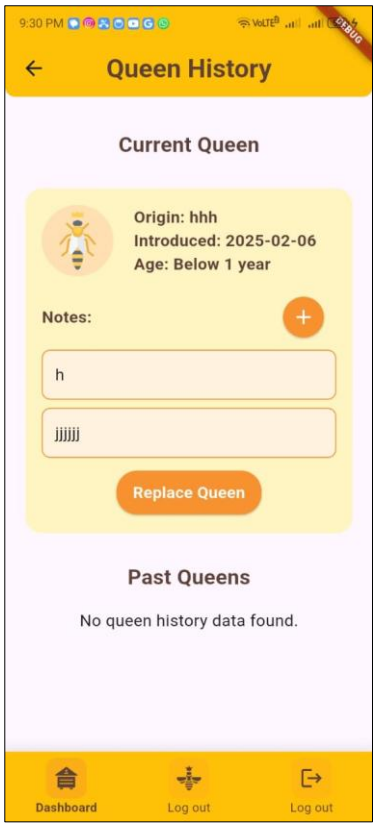
Live Screen

- The system must display real-time hive-specific sensor data.
- The system must analyze the hive health and provide real-time insights.
- The system must determine the optimal time for harvesting and provide real-time insights.



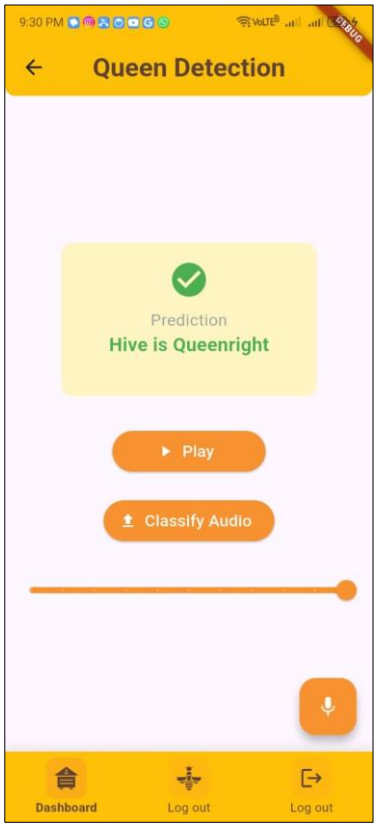
- The system must provide past sensor data in graphs for trend and pattern analysis

History Screen



Queen History Screen

- The system must display a hive's history of past queens.
- The system must allow the beekeeper to add details of a new queen bee.
- The system must allow the beekeeper to update if the current queen has been replaced.



Queen Detection Screen

- The system must allow the beekeeper to record bee sound through the app using the mobile microphone.
- The system must be able to process raw audio signals and predict whether the bee queen is present in the hive or not.
- The system must display the classification result: whether the bee queen is present or absent.

3.5.9 Admin Web App

The programming language, framework, development environments, and other tools used to implement the admin web application are as follows:

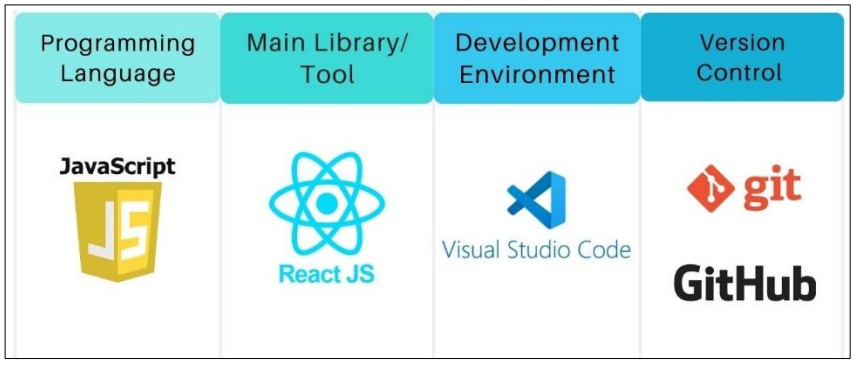
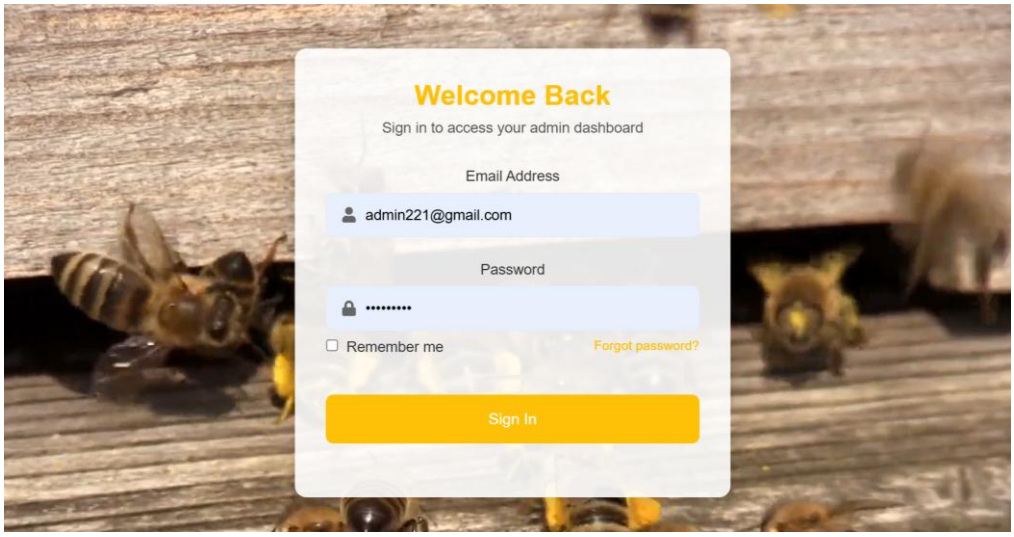


Figure 63: The programming language, framework, development environments, and other tools used to implement the admin web application

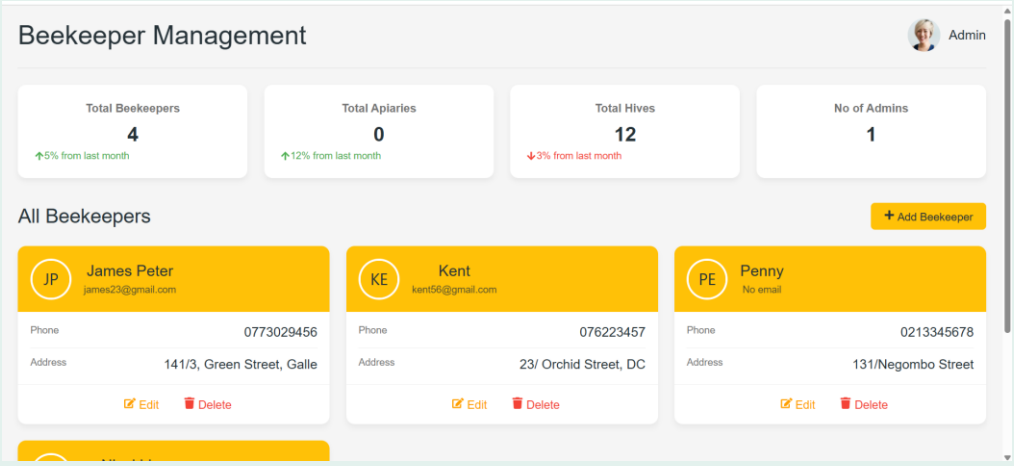
The table below maps each of the implemeneted admin web application's interfaces with the functional requirments defined in the requirement analysis section above.

Table 34: Mapping web interfaces with requirements



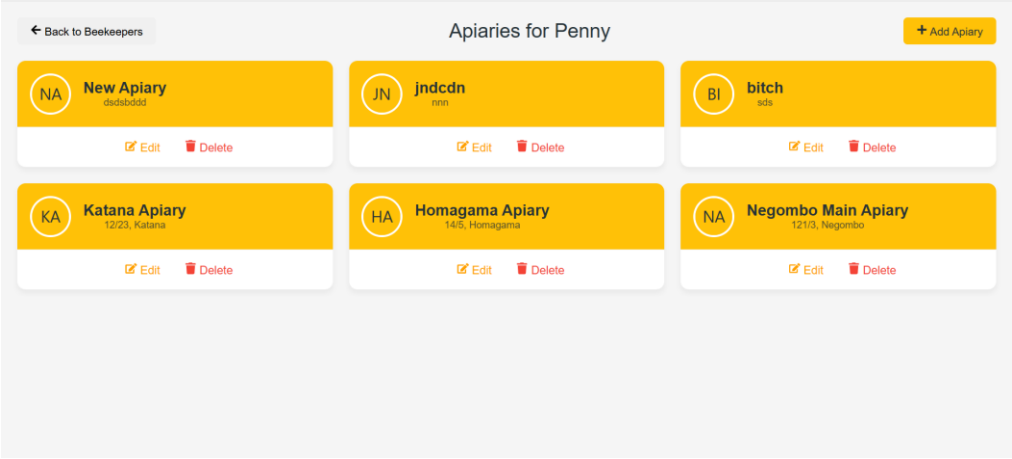
Login Page

- The system must allow the admin to log in successfully with the correct credentials.



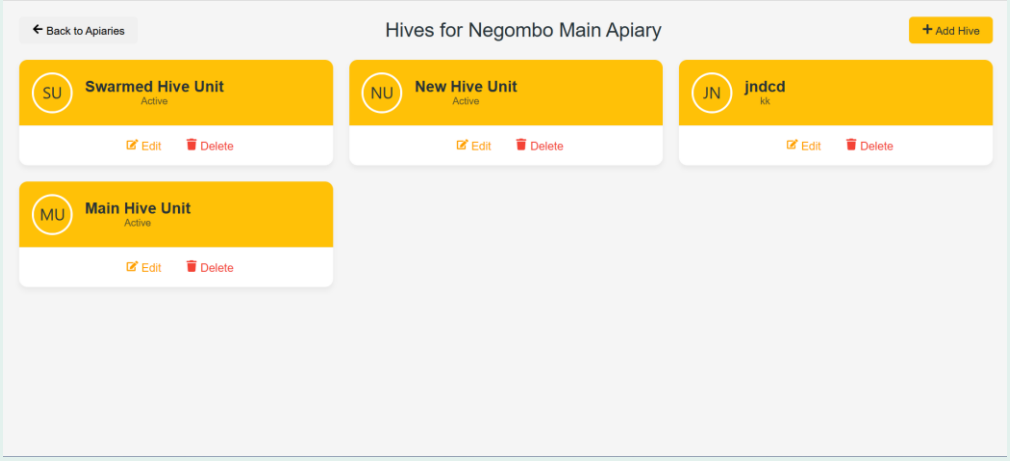
Beekeeper Page

- The system must allow the admin to manage (add, delete, update and view) beekeeper details



Apiary Page

- The system must allow the admin to manage (add, delete, update and view) apiary details under each beekeeper



Hives Page

- The system must allow the admin to manage (add, delete, update and view) hive details under each apiary

3.6 Testing

3.6.1 Unit Testing

Unit testing involves verifying individual components or units of code to ensure they function correctly in isolation. Various test cases for unit testing were designed and executed manually for the components described in the implementation section above. Some important designed and executed unit test cases can be accessed in **Appendix E: Unit Test Cases**. The table below summarizes some of the unit testing performed for each component of the system.

Table 35: Summary of unit testing performed

Component	Tested Unit	Test Objective	Test Case IDs
Hardware Setup	Wiring of Sensors	Verify correct wiring of all sensors (2x DHT22, 1x Load Cell) to ESP32	Unit Test Case 01
	Solar Panel Setup	Verify solar panel outputs proper voltage to LM2596	Unit Test Case 02
	Lithium-ion Batteries	Verify parallel battery output	Unit Test Case 03
Firmware Integration	Calibration Code	Verify load cell calibration correctness	Unit Test Case 04
	DHT22 Data Processing	Verify the data processing pipeline's correctness on the raw DHT22 sensor data.	Unit Test Case 05
	Load Cell Data Processing	Verify the data processing pipeline's correctness on the raw weight data.	Unit Test Case 06
	Energy Efficiency	Verify deep sleep functionality after data upload	Unit Test Case 07
	WiFi Connection	Verify successful WiFi	Unit Test Case 08

		connection	
Queen Model	Feature Extraction	Verify correct extraction of MFCC features	Unit Test Case 09
	Model Loading for Prediction	Verify that the model can be loaded correctly without data loss	Unit Test Case 10
Health Model	Model Loading for Prediction	Verify that the model can be loaded correctly without data loss	Unit Test Case 11
Beekeeper Mobile App	Audio Recording	Verify that the audio can be successfully recorded through the device's mic	Unit Test Case 12

3.6.2 Integration Testing

Integration testing examines the interactions between different components or modules within the system to ensure they work together seamlessly. It verifies that data flows correctly between components and that interfaces are properly integrated. Building upon the functionality validated in unit testing, integration testing assesses how well individual components integrate with each other to perform complex system functions. For instance, integration testing for this project involves confirming that sensor data is accurately transmitted to the Firebase database, retrieved for analysis, and visualized in the mobile application. It also verifies the data transmissions between the Flask server and the mobile application for model predictions, the data transmission between the Cloud Firestore database and both the web and mobile applications and the authentication of users with firebase authentication. Some important designed and executed integration test cases are included in **Appendix F: Integration Test Cases**. The table below summarizes key integration tests that were performed.

Table 36: Summary of integration testing performed

Tested Integrated Components	Test Objective	Test Case IDs
Firmware + Firebase Realtime Database	Verify that the sensor data are accurately transmitted to the Firebase Realtime	Integration Test Case 01

	database	
Firebase Realtime Database + Mobile Application	Verify that the mobile application retrieves relevant sensor data from the Firebase Realtime database	Integration Test Case 02
Queen Detection Model + Flask Server	Verify that the requests made to the /queen_detect API endpoint are processed correctly and that the prediction is received from the queen model and sent back as the response.	Integration Test Case 03
Health Analysis Model + Flask Server	Verify that the requests made to the /analyze_health API endpoint are processed, and that the prediction is received from the health model and sent back as the response	Integration Test Case 04
Flask Server + Mobile Application	Verify that the communication between the Flask server and the mobile application works correctly	Integration Test Case 05, Integration Test Case 06
Firebase Authentication + Mobile Application	Verify that the login functionality in the mobile application is successfully authenticated by Firebase Authentication.	Integration Test Case 07
Firestore Database + Mobile Application	Verify that the communication between the Firestore Database and the mobile application works correctly	Integration Test Case 08
Firestore Database + Web Application	Verify that the communication between the Firestore Database and the web application works correctly	Integration Test Case 09

3.6.3 System Testing

The system testing was conducted to evaluate the full functionality and overall performance of the entire system, including hardware setup, firmware, cloud setup, flask server, beekeeper mobile application, and the admin web application. For the beekeeper app's functionalities, end-to-end tests were conducted to assess the smooth workflow from logging in to viewing

apiaries and hives, retrieving live data, visualizing past data in graphs, receiving real time insights on hive health and harvesting, managing current queen details, viewing the past queen's info and getting the prediction on the queen's presence or absence with raw audio. Admin workflows were also tested, including creating, updating, and deleting beekeepers, hives, and apiaries. This is a summary on how system testing was performed in the developed system to ensure its smooth workflow and overall functionality.

3.6.4 User Acceptance Testing

Under UAT, the developed beekeeper mobile application was evaluated from the users' perspective to ensure it meets their needs and expectations. UAT was conducted with two actual beekeepers. They were provided with the fully implemented application and guided to use the application. After that they were asked to evaluate its functionality, user experience and performance through a small online feedback form they were provided with. The responses received from the feedback form are presented in **Appendix G: Feedback Form & Results**. Overall, the feedback was positive; the users were satisfied with the mobile app's features and its ease of use. They pointed out that Sinhala language support and a separate UI providing guidance on how to use the app can be added to further enhance user experience.

Chapter 04 | Evaluation & Reflections on Project Outcomes

4.1 Academic Findings & Research outcomes

This project contributes academically to the fields of precision agriculture, bee sound analysis, advanced beehive monitoring, Sri Lankan smart agriculture, embedded systems, and AI-assisted environmental monitoring.

Key findings and outcomes include:

- The CNN model's high accuracy (97%), even during cross-validation, demonstrates that the colony's sound patterns change significantly when a queen is present versus when the colony becomes queenless.
- Most previous research lacks a generalized model for queen detection, as they have trained the model with sound samples from one particular hive. In this project a dataset containing bee sounds across 7 different hives was used to train the model. This approach enhanced the model's generalizability, making it effective across multiple hives rather than being hive-specific.
- The findings confirmed that Asian honeybees regulate in-hive temperature and humidity within optimal ranges, despite fluctuations in external environmental conditions.
- Although in-hive temperature and humidity are not solely dependent on outside temperature and humidity, these external parameters still exert a significant influence.
- In-hive temperature and humidity exhibit clear daily patterns, with measurable differences according to the time of day, highlighting bees' dynamic environmental control.
- The developed hive health analysis model successfully identified anomalies: it performed well on both real test data from healthy hives and synthetic data simulating unhealthy conditions, proving its ability to detect deviations from normal hive behavior.
- The overall data flow—from sensors to ESP32, to the real-time database, to the mobile app, and then through the Flask server—was validated to work reliably, supporting real-time monitoring and analysis.

4.2 End-Project Assessment

The implemented Beehive Monitoring System aimed to develop a smart IoT-based solution to help beekeepers remotely monitor hive conditions in real time, analyze hive health, detect queen presence, and optimize harvesting decisions. The project delivered three main components as planned:

1. The hardware setup, which includes the IoT module and power supply,
2. The mobile application for beekeepers, and
3. The web-based admin portal.

The beekeeper mobile application is connected to the Firebase Realtime Database, which stores real-time sensor data sent from the ESP32 microcontroller. This enables the app to display both live and historical hive data, allowing beekeepers to track trends and patterns over time.

The mobile application and backend server work together to provide meaningful insights into hive health and queen presence or absence. This is achieved through trained machine learning models that analyze bee audio samples recorded from the device's audio to predict the queen's presence and live sensor data from the hives to predict whether the hive is healthy or not.

The system also uses Firestore to store structured data about beekeepers, their apiaries, hives, and both current and past queens of each hive. The Firestore database is integrated with both the mobile application and the web admin portal. While the admin can add, edit, and delete information related to beekeepers, apiaries, and hives through the web portal, beekeepers can view their hive and apiary details and log information about current and past queens via the mobile app.

The interaction between all components—hardware, firmware, cloud databases, mobile app, admin portal, backend server and the trained ML models—functioned correctly as intended and was confirmed through thorough testing as described in the above Testing section.

The table below presents a critical analysis of the extent to which each initial objective (objectives that were defined at the very beginning of the project) was achieved by the end of the project. The initial objectives considered here are listed down in the original project proposal document which can be accessed from **Appendix H: Supporting Project Documents**.

Table 37: A critical analysis of the extent to which each initial objective was achieved by the end of the project.

Initial Objective	Status	Comments
Visualize live data from the hive	Achieved	Sensor data are updated real time in the app's interface from the Firebase Real-time database
Offer historical data analysis and trend forecasting to support informed decision-making.	Partially Achieved	The mobile app displays past sensor data in graphs but lacks the feature to forecast future trends and patterns.
Deliver insights into hive health by detecting anomalies in normal bee activities and hive conditions.	Achieved	The app is able to provide real-time insights into hive health status (whether the hive is healthy or not) through the trained novel detection model, which is able to detect novelties from the normal hive conditions as anomalies
Provide insights into the optimal time for harvesting.	Achieved	The app takes the average of the latest weight data records and decides whether the hive is ready to harvest or not if the average hive weight exceeds the given threshold.
Automatically detect the presence of the queen bee using acoustic signals.	Achieved	The app allows the users to record the bee sounds through the device's mic and then process the sound to predict whether the hive is queenless or queenright through the trained queen

		detection model.
Automatically record any swarming even ts that occur	Not Achieved	This feature was not included in the scope of the final implemented product
Maintain a record of the past queen bees' history within the hive.	Achieved	The Firestore database was used to store the details of the past queen bees of each hive

The majority of the initial objectives were met. However, the swarming detection feature was not achieved, and forecasting future hive trends was also not accomplished.

The following table summarizes and evaluates how well the project performed compared to the initial plan in key areas: timeline, budget, and quality.

Table 38: Evaluation on how well the project performed compared to the initial plan

Key Area	How well did the project perform?	Reason
Timeline	The planned timeline was extended by 1 and half months	• Addition of new features on the way • Stuck with various experiments with the queen detection model due to its low performance in the group-fold cross-validation scenario.
Budget	Planned Budget was exceeded by Rs.10,000	• Unexpected hive management expenses, such as issues with wax moth infections • The need for additional power supply components was more than expected.
Quality	The project performed well in terms of quality.	• Achieved the expected quality standards satisfactorily after thorough testing.

The project performed well in terms of quality, as the final outcome met the defined quality objectives. However, the project underperformed in terms of timeline and budget, as both factors were significantly exceeded compared to the initial plan.

The following table summarizes and evaluates the various changes made to the original project plan, the reasons for these changes, and their impact.

Table 39: Various changes made to the original project plan, the reasons for these changes, and their impact

Initial Plan	Change	Reason	Impact
Perform signal processing, feature extraction, model training, and evaluation using the device through PyCharm.	Performed signal processing and feature extraction on the device using PyCharm; model training and evaluation were done in Colab.	The device had RAM constraints for model training, and it was not feasible to upload a large dataset to Google Drive to perform the full workflow from signal processing in Colab.	The pipeline workflow was divided into two stages: local processing (signal processing and feature extraction) and cloud-based model training/evaluation, optimizing resource use and ensuring the process could be completed efficiently.
Integrate the saved queen detection model with the app using TensorFlow Lite.	Implemented a Flask server to connect the models with the app.	Limited number of Dart libraries for audio processing	Allowed for more robust integration with the mobile app while addressing the limitations in the Dart ecosystem.
Managing beekeeper, apiary,	Implemented a web admin portal so that	Offering a more comprehensive and	Streamlined admin management;

and hive details are done by the admin directly through the Firestore database.	the admin can manage beekeeper, apiary, and hive details using the web app.	user-friendly interface	provides a more intuitive and efficient interface for administrators to manage data.

4.3 Post-Mortem Analysis

Overall, the project objectives were well-chosen and aligned closely with real-world beekeeper needs, as confirmed through data gathering and user acceptance testing sections. These objectives included enabling remote monitoring of hive conditions, queen detection, tracking past queen details, hive health analysis, and optimal harvest time determination.

The product was well specified in terms of hardware, app functionality, and backend integration. One gap was underestimating the complexity of audio processing for queen detection, which delayed development and caused significant changes in mid-project. Better initial research into hardware limitations and audio machine learning requirements could have refined the scope earlier.

The stakeholders (beekeepers and bee researchers at the Bandarawela Bee Unit) were supportive and responsive, especially in providing field access and feedback. Regular check-ins were conducted to confirm requirements and project outcomes, ensuring that development stayed on track and aligned with stakeholder expectations.

The Waterfall methodology worked well for most of the project, as many requirements were stable from start to finish (which is rare for long projects). Waterfall provided a structured, step-by-step process. However, when some requirements changed mid-way (such as shifting the bee sound recording from an electret microphone to the mobile device's app), an Agile Scrum methodology would have worked better. In hindsight, Agile might have been the best overall approach for this project due to its flexibility.

ESP32 was an excellent choice due to its ease of integration with the Realtime Database and strong performance. Firebase Realtime Database was also a great choice for syncing data seamlessly across devices and apps. The Flask server provided flexibility to handle backend

processing and machine learning. The Flutter framework worked well for the mobile application and React was the best choice for the web application, offering real-time UI updates. The use of the TensorFlow framework provided flexibility in developing the CNN model for queen detection. Librosa worked well for signal processing, and Keras was effective for machine learning and preprocessing tasks.

The development of this project was an enlightening experience for the author. The author had never developed a project of this scope before, and signal processing and audio classification were brand new areas that required significant learning and practice. Despite the steep learning curve, the author demonstrated resilience and adaptability in tackling the complexities involved with implementing the queen detection model.

Feedback received from UAT showed that beekeepers suggested offering Sinhala language support in the application and suggested having a separate interface offering proper user guide. Apart from that, the overall feedback was positive, confirming the usefulness of the system and the need for continuous development.

Various lessons were learned from the implementation of this project. These include the importance of proper planning with thorough risk analysis, which could have helped mitigate many challenges faced during implementation. Another key lesson was the need for strong communication with stakeholders and the importance of prioritizing documentation. Ultimately, this project proved that with hard work and proper planning, it is possible to achieve a project of significant scope.

Chapter 05 | Conclusion

5.1 Limitations

The developed system has delivered a promising solution; however, it has several limitations that pave the way for future enhancements.

Firstly, although the queen detection model showed high performance under holdout cross-validation and K-Fold cross-validation (where the entire dataset was divided into k folds and evaluated without grouping by hive), the dataset was not evaluated using Group K-Fold cross-validation. This method involves training the model on data from all hives except one and then testing it using sound data (both queenright and queenless) from the excluded hive—ensuring the model is truly generalized and not hive-specific. While K-Fold cross-validation demonstrated generalization across the 7 hives it was trained on, Group K-Fold cross-validation is crucial to confirm that the model can accurately classify sounds from completely unseen hives. This type of validation could not be carried out in this project because most hives in the dataset contained either queenless or queenright data, but not both, making group-based evaluation impractical.

Another limitation is that the model was trained with a dataset that primarily made of sound samples from European honeybees. Only a few samples from local hives were included under the queenright scenario. To improve model accuracy and relevance for Sri Lankan beekeepers—where Asian honeybees (Apis cerana) are predominant—it is essential to collect and train the model on a larger dataset from local Asian honeybee colonies.

Currently, beekeepers must physically visit a hive and place the mobile device near or inside the hive to record bee sounds, which are then analyzed to predict whether the hive is queenless or queenright. Although this method benefits beekeepers by eliminating the need to inspect each frame manually (a task that is difficult even for experienced beekeepers, as queens are often hard to locate), it still introduces manual intervention and requires active participation from the beekeeper for sound recording.

The hive health analysis model was trained solely on normal, healthy hive data. For testing purposes, synthetic unhealthy data was generated. To achieve more reliable results, it is

essential to collect real-world unhealthy hive data and test the model's accuracy using actual samples from diseased or compromised hives.

The system also paid less attention to security considerations, especially regarding the protection of data transmission and storage. Additionally, testing was limited to manual functional testing and did not include comprehensive performance, security, or automated testing methodologies, which are important for ensuring long-term robustness and scalability.

5.2 Future Work

Under future work, improving the limitations identified above and adding new features will be considered. The graph below depicts the planned future enhancements for this project.

Figure 64: Future Enhancements

5.3 Final Remarks

This project has successfully demonstrated the potential of integrating IoT and machine learning technologies to enhance hive monitoring and management for beekeepers. Despite challenges related to hardware limitations and the complexity of audio processing, the system achieved its primary objectives and delivered practical value. Through the implementation of this work, the author aims to support Sri Lankan beekeepers by providing advanced hive management and monitoring tools, including real-time data collection, automated queen detection, and hive health analysis. The application has the potential to revitalize the beekeeping industry in Sri Lanka by making hive inspections more efficient, improving overall colony health management, and ultimately benefiting both the beekeepers and the bees themselves.

References

1. Abdollahi, M. a. G. P. a. F. T. H., 2022. Automated Beehive Acoustics Monitoring: A Comprehensive Review of the Literature and Recommendations for Future Work. *Applied Sciences,* 12(8).
2. Andrijević, N. a. U. V. a. A. B. a. H. D. a. S. B., 2022. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics,* Volume 11, p. 783.
3. Anon., n.d.
4. Babiuch, M. a. F. P. a. S. P., 2019. Using the ESP32 Microcontroller for Data Processing. *2019 20th International Carpathian Control Conference (ICCC),* pp. 1-6.
5. Bayir, R. a. A. A., 2016. The monitoring of nectar flow period of honey bees using wireless sensor networks. *International Journal of Distributed Sensor Networks,* Volume 12.
6. Benetos, N. a. E., 2018. *To bee or not to bee: Investigating machine learning approaches to beehive sound recognition.* Surrey,UK, Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE).
7. Braga, D. a. M. A. a. S. F. a. P. V. a. A. A., 2021. An Intelligent Monitoring System for Assessing Bee Hive Health. *IEEE Access,* Volume 9, pp. 89009-89019.
8. Branco, D. C. a. J. M. a. H. M. a. F., 2023. BHiveSense: An integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices. *Journal of Open Innovation: Technology, Market, and Complexity,* Volume 9, p. 100110.
9. Carlson, R. D., 2015. *The Honey Bee and Apian Imagery in Classical Literature,* Seattle, WA: University of Washington.
10. Cazier}, A. {. B. a. D. {. G. a. R. R. a. E. {. H. a. B. {. F. a. J. {., 2020. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture,* Volume 169, p. 105161.
11. Cecchi, S. a. S. S. a. T. A. a. O. S., 2020. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors,* 20(9), p. 2726.
12. Cejrowski, T. a. S. J. a. M. H. a. G. D., 2018. Detection of the Bee Queen Presence Using Sound Analysis. In: s.l.:Springer, pp. 297-306.
13. Christina T. Liang, B. G. K. H. W. T., 2016. The Role of Honey Bees as Pollinators in

Natural Areas. *Natural Areas Journal,* 36(4), pp. 478-488.

14. Diogo Braga, Ana Mudureura, Fabio Scotti, Vincenzo Piri, Ajith Abraham, 2021. *An Intelligent Monitoring Systemfor Assessing Bee Hive Health.* Portugal, s.n.
15. Doinea, M. et al., 2024. IoT Embedded Smart Monitoring System with Edge Machine Learning for Beehive Management.. *International Journal of Computers, Communications & Control,* 19(4).
16. Dorin, A. Z. a. A., 2023. A framework for better sensor-based beehive health monitoring. *Computers and Electronics in Agriculture,* 210(0168-1699), p. 107906.
17. E. R. H. S. S. Ediriweera, N Y S Premarathna, 2012. *Medicinal and cosmetic uses of Bee's Honey – A review.* Sri Lanka, s.n.
18. E.R.H.S.S. Ediriweera, N. Y. S. P., 2012. Medicinal and cosmetic uses of Bee's Honey – A review. *AYU (An International Quarterly Journal of Research in Ayurveda),* 33(2), pp. 178-82.
19. Eduardo Fonseca, R. G. D. B. O. S. E. G. X. S., 2017. *Acoustic Scene Classification by Ensembling Gradient Boosting Machine and Convolutional Neural Networks.* Munich, Germany, Workshop on Detection and Classification of Acoustic Scenes and Events.
20. Edwards-Murphy, F. a. M. M. a. W. P. a. O. J. a. P. E., 2016. B+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture,* Volume 124, pp. 211-219.
21. GeeksForGeeks, 2025. *Cross Validation in Machine Learning.* [Online] Available at: https://www.geeksforgeeks.org/cross-validation-machine-learning/ [Accessed 25 04 2025].
22. Hegde, A. D. a. A. P. a. S., 2023. *HiveLink, an IoT based Smart Bee Hive Monitoring System.* [Online] Available at: https://arxiv.org/abs/2309.12054 [Accessed 19 10 2024].
23. Isayas Asefa Kebede, H. F. A. D. A. G. D., 2024. Bee products and their processing: a review. *Pharmacy & Pharmacology International Journal,* 12(2), pp. 5-12.
24. Isidoro, S. C. a. A. T. a. S. O. a. P. R. a. S. R. a. N., 2018. A Preliminary Study of Sounds Emitted by Honey Bees in a Beehive. *Journal of The Audio Engineering Society,* p. https://api.semanticscholar.org/CorpusID:116308926.
25. Jia-Ching Wang and Jhing-Fa Wang and Cai-Bei Lin and Kun-Ting Jian and Kuok, W., 2006. Content-Based Audio Classification Using Support Vector Machines and Independent Component Analysis. *18th International Conference on Pattern*

Recognition (ICPR'06), Volume 4, pp. 157-160.

26. Jia-Ching Wang, J.-F. W. C.-B. L. K.-T. J. K. W., 2006. Content-Based Audio Classification Using Support Vector Machines and Independent Component Analysis. *18th International Conference on Pattern Recognition (ICPR'06),* Volume 4, pp. 157-160.
27. Kesavan, R. a. G. D. a. T. D. a. S. J. a. M. C., 2023. Firestore: The NoSQL Serverless Database for the Application Developer. *2023 IEEE 39th International Conference on Data Engineering (ICDE),* pp. 3376-3388.
28. Khan, S. A., 2023. Review of the Role of Bees as Ecosystem Engineers in Nature. *Science Reviews - Biology,* 2(4), pp. 1-6.
29. Kridi, D. S. a. d. C. C. G. N. a. G. D. G., 2016. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Elsevier Science Publishers B. V.,* 127(C), p. 221–235.
30. Li, W.-J. a. Y. C. a. L. Y.-S. a. T. S.-C. a. H. S., 2018. JustIoT Internet of Things based on the Firebase real-time database. *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE),* pp. 43-47.
31. Liyanage, N. a. A. C. a. P. T. a. N. D. a. B. I. S. a. C. L., 2024. IoT-Based Smart Beehive Monitoring System. *2024 6th International Conference on Advancements in Computing (ICAC),* pp. 247-252.
32. Maier, A. a. S. A. a. V. Y., 2017. Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. *2017 Internet Technologies and Applications (ITA),* pp. 143-148.
33. Mansor, M., 2022. *IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping using Lora Technology..* s.l, s.n.
34. Orlowska, A. a. F. D. a. G. J.-P. a. C.-R. D., 2022. Honey Bee Queen Presence Detection from Audio Field Recordings Using Summarized Spectrogram and Convolutional Neural Networks. In: s.l.:s.n., pp. 83-92.
35. osbeehives, 2013. *OSBH – Open Source Beehives.* [Online] Available at: https://fablabbcn.org/projects/osbh-open-source-beehives [Accessed 24 04 2025].
36. Peter Hristov, R. S. N. P. B. N., 2020. Factors Associated with Honey Bee Colony Losses: A Mini-Review.. *Veterinary sciences,* 7(4), p. 166.
37. Pikri, M. a. I. S. a. F. N. a. M. M. a. M. S., 2022. IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping Using LoRa Technology. In: s.l.:s.n.,

pp. 334-342.

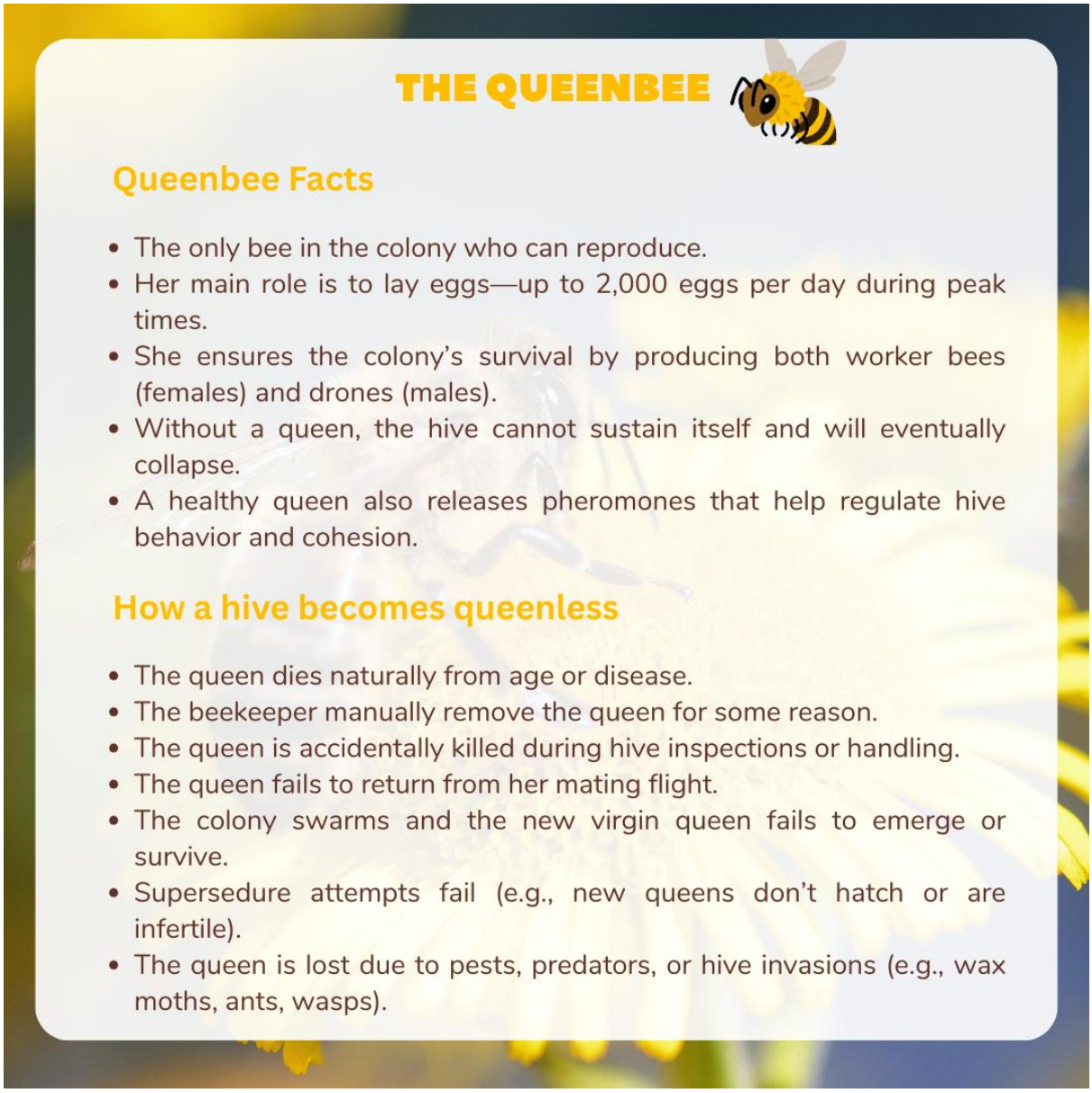
38. Ranganath Manohar Rane, S. S. K., 2021. A Research Paper on Firebase Authentication. *IJSRD - International Journal for Scientific Research & Development,* 9(5).
39. Rector, B. a. M. G. a. H. N., 2008. Within-day variation in continuous hive weight data as a measure of honey bee colony activity. *Apidologie,* Volume 39, pp. 694-707.
40. sayas Asefa Kebede,Haben Fesseha, Gebremeskel, Abrahim Dawed Ahmed, Gelan Dule, 2024. *Bee products and their processing: a review.* Ethiopia, s.n.
41. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. Low cost platform for monitoring honey production and bees health. *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR),* pp. 1-4.
42. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. *Low cost platform for monitoring honey production and bees health.* s.l., 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR).
43. Shaden A. M. Khalifa, E. H. E. A. A. S. A. A. A. E.-W. A. F. A. S. G. M. M. F. A. C. Z. S. H. D. M. M. A.-D. M. F. H. G. K., 2021. Overview of Bee Pollination and Its Economic Value for Crop Production. *Insects 2021,* 12(8), p. 688.
44. Shawe-Taylor, J., 2015. *Novelty Detection with One-Class Support Vector Machines.* Switzerland, Springer International Publishing.
45. Uthoff, C. a. N. H. M. a. V. B. M., 2023. Acoustic and vibration monitoring of honeybee colonies for beekeeping-relevant aspects of presence of queen bee and swarming. *Computers and Electronics in Agriculture,* Volume 205.
46. Vidushi Patel, N. P. E. B. L. B. &. B. B., 2021. Why bees are critical for achieving sustainable development. *Ambio,* 50(1), pp. 49-51.
47. Weiss, M. a. M. P. a. F. W. a. S. L. a. S. T. a. M. B. a. A. K., 2017. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie,* Volume 48.
48. Whitfield, B. G., 1956. Virgil and the Bees: A Study in Ancient Apicultural Lore. *Greece and Rome,* 3(2), p. 99–117.
49. Yiyao Zheng , Xiaoyan Cao, Shaocong Xu, Shihui Guo, Rencai Huang, Yingjiao Li, Yijie Chen, Liulin Yang, Xiaoyu Cao,Zainura Idrus, Hongting Sun, 2024. *Intelligent beehive monitoring system based on internet of things and colony state analysis.* China, s.n.
50. Yiyao Zheng, X. C. S. X. S. G. R. H. Y. L. Y. C. L. Y. X. C. Z. I. H. S., 2024. Intelligent

beehive monitoring system based on internet of things and colony state analysis. *Smart Agricultural Technology}*, 9(1), p. 100584.

51. Zacepins, A. a. P. A. a. O. V. a. K. A. a. E. S., 2017. Solution for automated bee colony weight monitoring. *Agronomy Research*, Volume 15, pp. 585-593.

Appendix

Appendix A: Scientific and Biological Context



THE QUEENBEE

How a hive becomes queenless?

- The queen succumbs to viral, bacterial, or fungal diseases that target her specifically or weaken the colony.
- The queen is intentionally killed by the worker bees themselves (e.g., if she becomes infertile, weak, or produces poor pheromones, workers may “ball” and kill her).
- The queen’s health got affected from extreme weather, lack of food, or pesticide exposure.

What happens when a hive becomes queenless?

- Worker bees begin raising emergency queen cells from young larvae (typically under 3 days old) to start making a new queen.
- The colony becomes restless and disorganized in the absence of queen pheromones.
- If no viable larvae are available, some workers may start laying unfertilized eggs (resulting in drones only).
- The hive’s overall population and productivity decline without a fertile queen.
- Colony may swarm as a survival mechanism, losing part of the population.

THE QUEENBEE

Why Monitoring the Queen Bee's Status Is Crucial for Hive Health?

- Bees don't always succeed in raising a new queen, especially if conditions are poor.
- There's a long broodless period while a new queen develops and mates, which weakens the colony.
- If no queen emerges in time, worker bees may start laying unfertilized eggs, leading to colony collapse.
- Sometimes the new queen is weak or poorly mated, causing long-term hive failure.
- Late-season queen replacement is risky because mating opportunities decrease.
- Beekeepers may prefer to introduce a proven, high-quality queen to ensure strong hive performance.
- Without a queen, the colony often becomes weak and may need manual feeding to stay alive and regain strength.

THE QUEENBEE

How beekeepers intervene in queenless situations?

- Introduce a high-quality mated queen from a trusted source.
- Transfer an extra queen from another healthy hive.
- Add a frame of eggs or young larvae from a strong colony to allow the bees to raise a new queen.
- Sometimes allow the colony to make a new queen naturally but monitor the process closely.
- Remove extra or unhealthy queen cells to ensure only the strongest queen develops.
- Replace a weak or underperforming queen if one emerges but is not healthy.
- Combine the queenless colony with a smaller queenright colony if requeening is not successful.
- Monitor the hive carefully after intervention to ensure the new queen is accepted and begins laying eggs.

Appendix B: User Guide

B.1 Minimum Platform Specifications

Note: These specifications apply after the deployment of the backend server and web application, and after the mobile app is installed to playstore.

- Mobile App:
  - Android 8.0 (Oreo) or above.
  - Internet Connectivity
  - Minimum 2 GB RAM.
  - 100 MB free storage space.
  - Microphone permission enabled.
- Web Application:
  - Modern browser (Chrome, Firefox, Edge) updated to the latest version.
  - Internet Connectivity
- IoT Hardware:
  - ESP32 with connected sensors (load cell, and the 2 DHT22 sensors) powered using the implemented power supply or any other power source
  - Internet Connectivity

B.2 The normal real-world workflow of the system

1. The business who owns the system places the IOT module with the power supply on a hive belonging to the client.
2. The system admin creates a new beekeeper account with a login email and password and other beekeeper details through the admin web application
3. The admin adds a new apiary under that beekeeper and then registers the hive details for the hive where the hardware setup was previously installed.
4. The admin uploads the firmware to ESP32, including the client's Wi-Fi credentials and

the specific hive ID, which was retrieved from the web app for the registered hive.

5. The admin provides the beekeeper with the login credentials (email and password) that were used to register the beekeeper in step 2

6. The beekeeper can access the apiary and then the specific hive that was registered through the mobile app after logging in.

6. The beekeeper can visualize live sensor data, historical data, and real-time analytical insights from that particular hive.

B.3 Installation & Setup

Note: Currently, none of the system components (web app, mobile app, Flask backend server, or IoT firmware) are deployed on live servers or available on app stores. This installation and setup guide applies after the full system is deployed.

Web Application (This setup will be done by the admin):

1. Accessing the Web Application:
    - Open your preferred browser (Chrome, Firefox, or Edge) and navigate to the provided URL for the web application, which will be shared after deployment.
2. User Authentication:
    - Log in to the system using the admin credentials provided upon deployment.
3. Adding a New Hive:
    - After successfully logging in, you'll be prompted to register the beekeeper/user, apiary, and then the hive.
    - Follow the on-screen instructions to input the necessary details for the new hive (e.g., hive name, location). A hive ID will be generated for the new hive.

IoT Hardware (This setup will be done from the business side or by the admin himself):

1. Preparing the IoT Module:
    - Power on the ESP32 and ensure the IoT device is connected to the hive via the required sensors (load cell and DHT22).
    - Make sure the device has an active internet connection through Wi-Fi.
2. Deploying the Sensors:
    - Mount the load cell and DHT22 sensors in the hive at optimal locations (e.g., DHT22 near the brood area, load cell for weight measurement under the hive).
3. Powering the IoT Device:
    - Ensure the IoT module is powered via the provided power supply or other compatible sources (e.g., solar panels, rechargeable batteries).
    - Verify that the module remains connected to the network to transmit data continuously.
4. Flashing the ESP32:
    - Flash the ESP32 with the updated firmware, which includes the new Wi-Fi username and password, as well as the hive ID generated after the hive registration through the web admin application.

Mobile App Setup (This installation and setup will be done by the system user/ beekeepers):

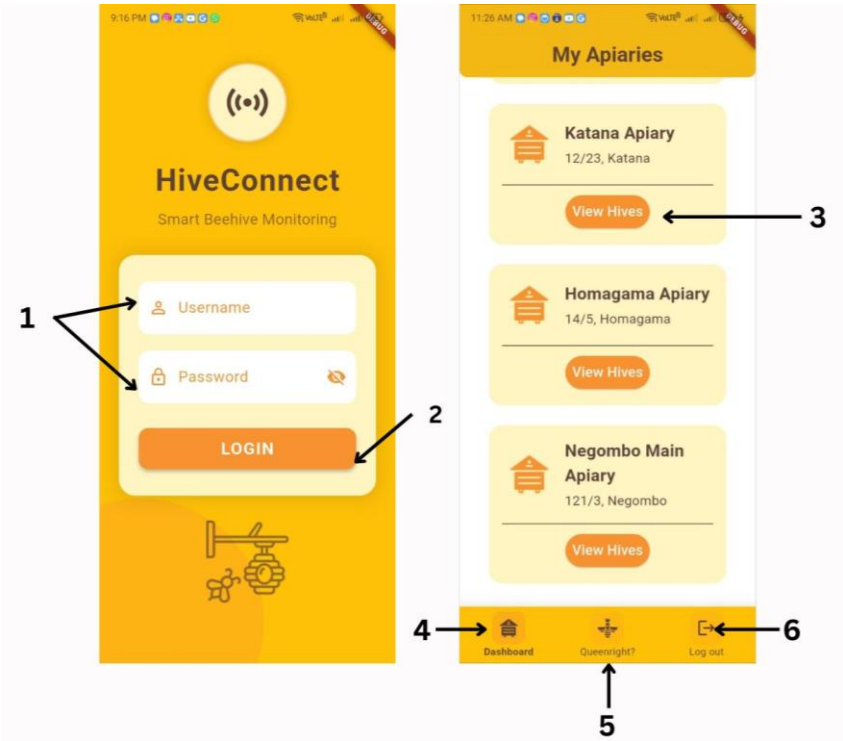
1. Download the Mobile App:
    - The mobile app will be available in the Google Play Store once it has been published.
2. Permissions:
    - Once the app is installed, grant the necessary permissions, such as Microphone access for data collection and Internet connectivity.

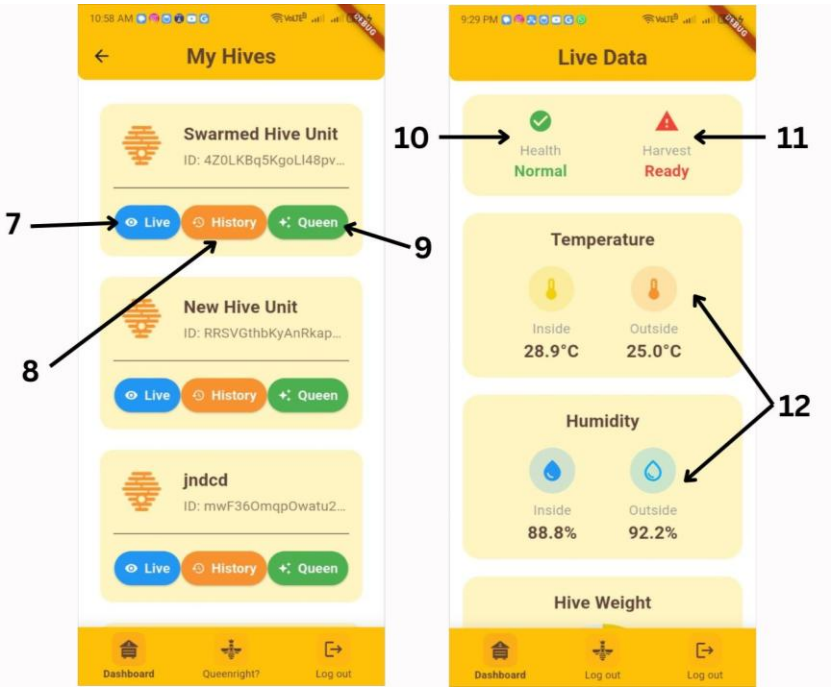
3. Accessing Hive Data

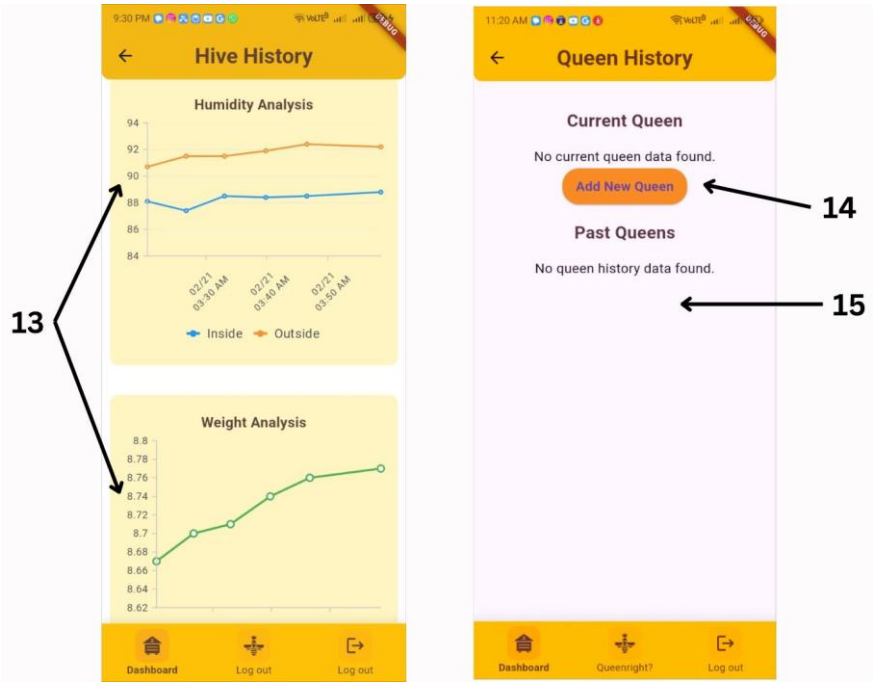
Note: After the new hive has been registered through the web admin application and the ESP32 has been flashed with the updated firmware, you can access the hive data

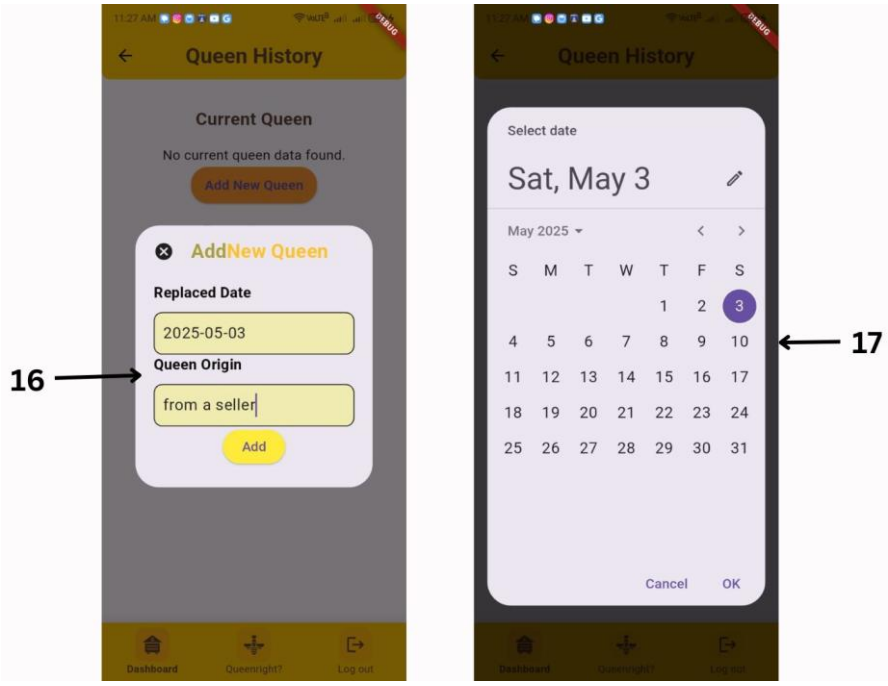
- Login with the credentials provided by the admin after registering the new user.
- Navigate to the specific hive.
- Access the data collected by the sensors in real time.

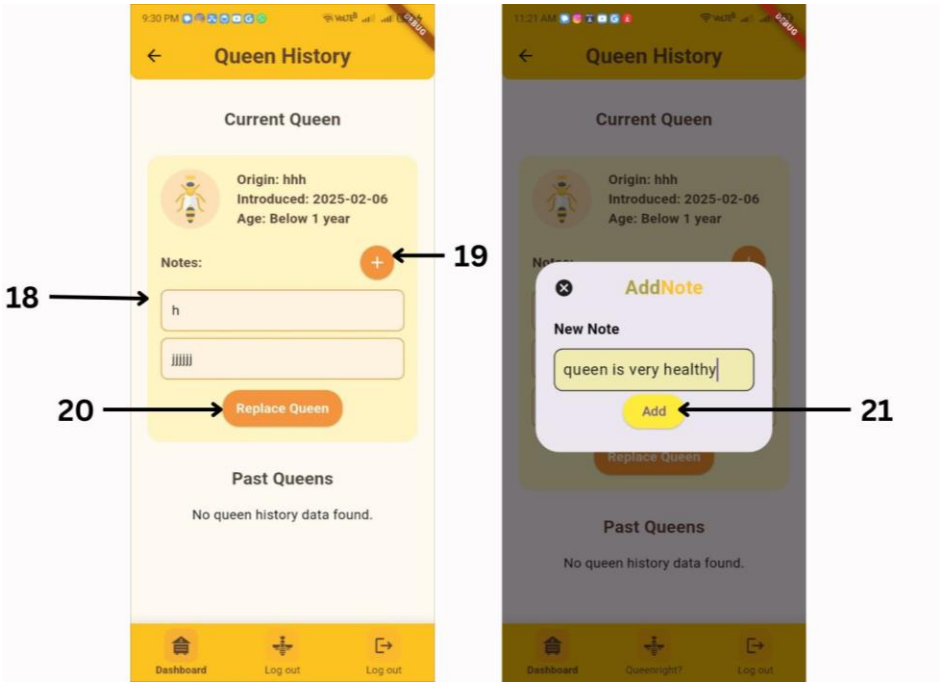
B.4 How to use the beekeeper mobile application

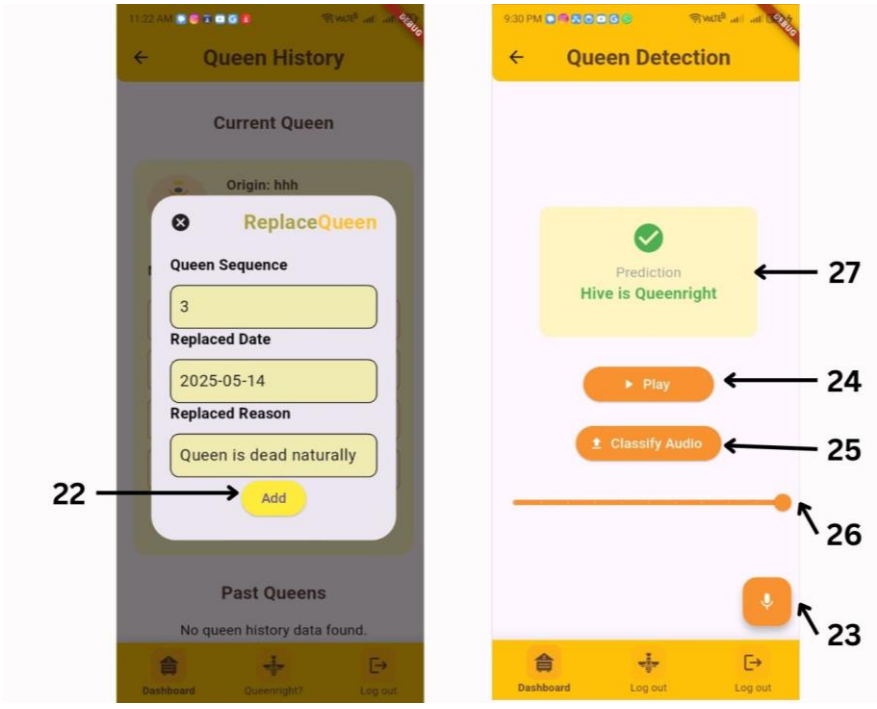



My Hives
Swarmed Hive Unit
ID: 4Z0LKBq5KgoLl48pv...
Live
History
Queen
New Hive Unit
ID: RRSVGthbKyAnRkap...
jndcd
ID: mwF36OmqpOwatu2...
Dashboard
Queenright?
Log out
Live Data
Health
Normal
Harvest
Ready
Temperature
Inside
28.9°C
Outside
25.0°C
Humidity
88.8%
92.2%
Hive Weight
7
8
9
10
11
12


Hive History
Humidity Analysis
Inside
Outside
Weight Analysis
Dashboard
Log out
Queen History
Current Queen
No current queen data found.
Add New Queen
Past Queens
No queen history data found.
Queenright?
13
14
15


Queen History
Current Queen
No current queen data found.
Add New Queen
AddNew Queen
Replaced Date
2025-05-03
Queen Origin
from a seller
Add
16
Dashboard
Queenright?
Log out
Queen History
Select date
Sat, May 3
May 2025
S M T W T F S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
Cancel
OK
17


Queen History
Current Queen
Origin: hhh
Introduced: 2025-02-06
Age: Below 1 year
Notes:
19
18
h
jjjjjjj
20
Replace Queen
Past Queens
No queen history data found.
Dashboard
Log out
Log out
Queen History
Current Queen
Origin: hhh
Introduced: 2025-02-06
Age: Below 1 year
AddNote
New Note
queen is very healthy
Add
21
Replace Queen
Past Queens
No queen history data found.
Dashboard
Queenright?
Log out

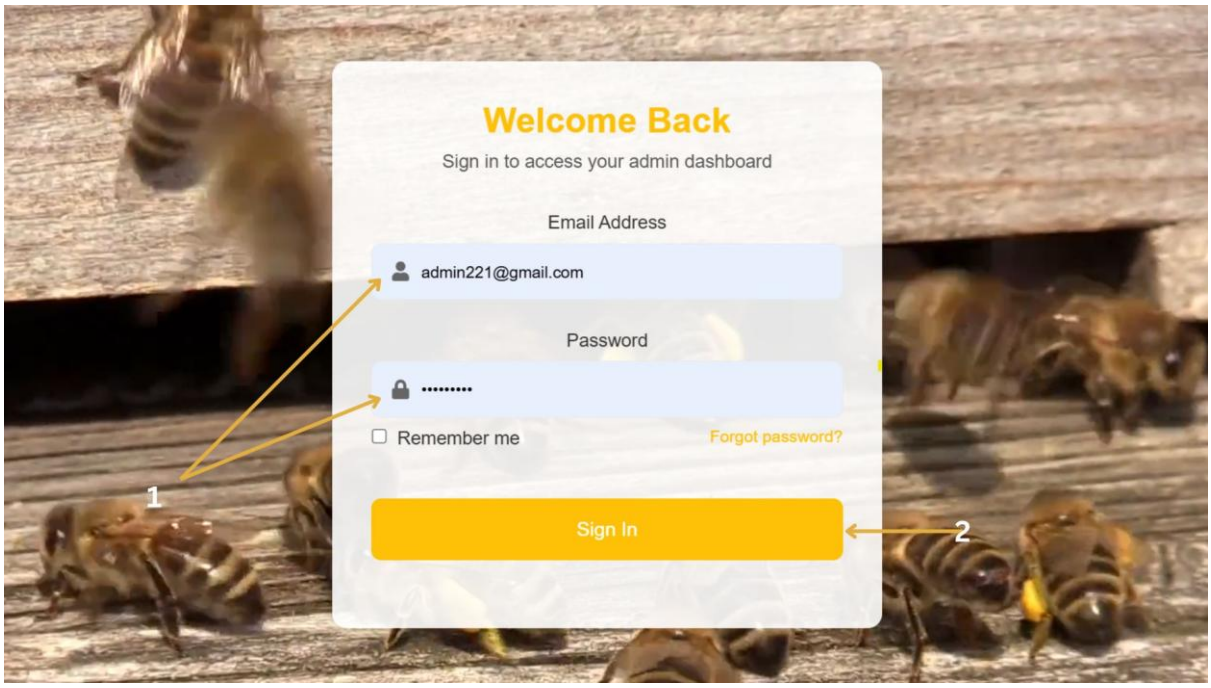


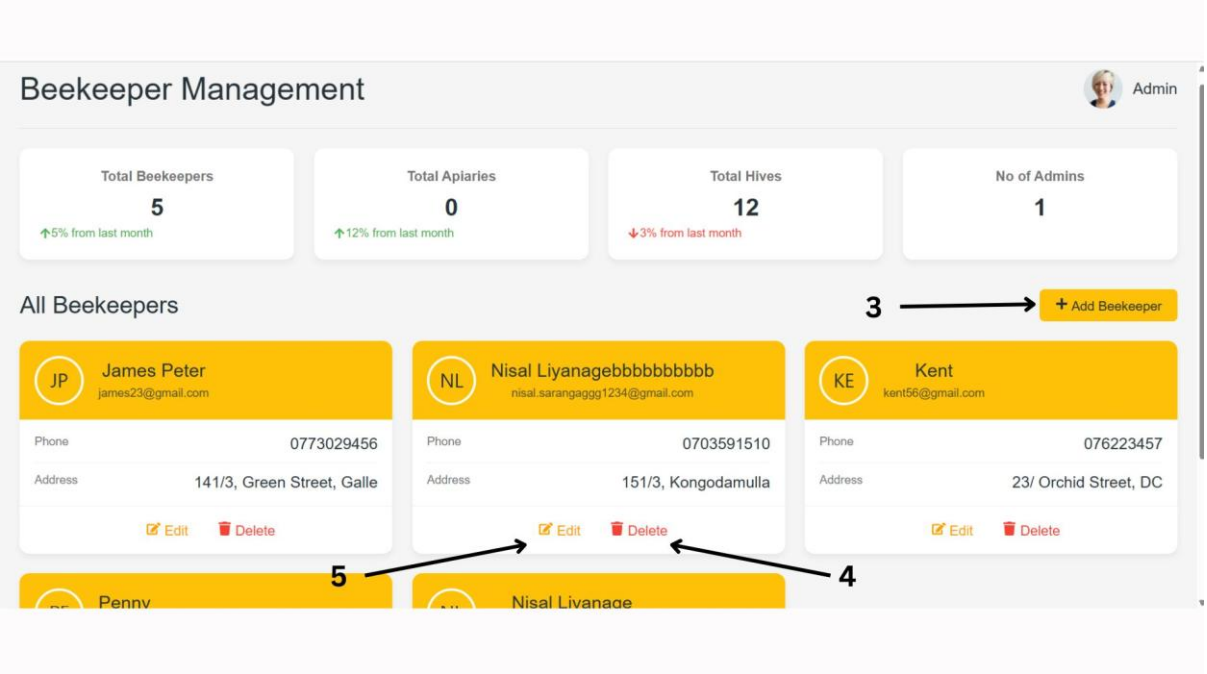
Index	Description
1	Enter the login credentials as provided by the admin
2	Click on the login button to login to the system
3	View each apiary belonging to you. Click on the 'View Hive' button under each apiary to view all the hives belonging to that apiary
4	Navigate to the dashboard/ apiaries screen
5	Navigate to the queen detection screen
6	Navigate to the login screen (logout)
7	Click on the 'Live' button under each hive to visualize live data from that particular hive
8	Click on the 'History' button under each hive to visualize past data from that particular hive

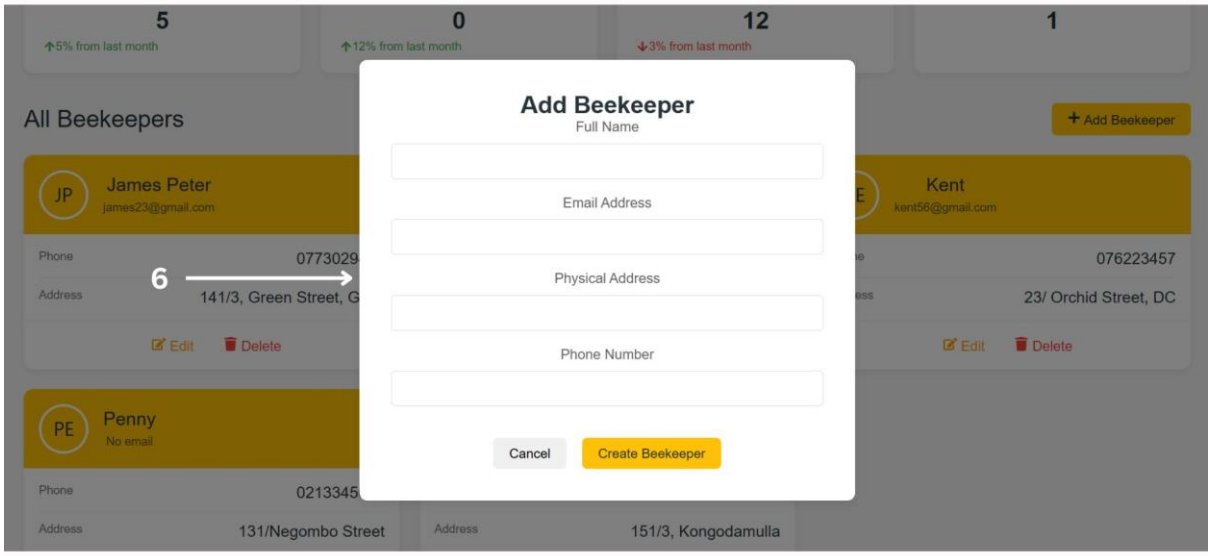
9	Click on the ‘Queen’ button under each hive to view the current and past queen details of that particular hive
10	View real-time insights into the health status of the selected hive
11	View real-time insights into the harvest need (hive is ready to harvest or not) for the selected hive
12	View real-time sensor data from the selected hive
13	Visualize historical data in graphs for the selected hive
14	Click on “Add New Queen” button to add a new queen to the selected hive
15	View the past queen details for the selected hive
16	Add the current queen details of the selected hive and click on the “Add” button in the popped-up dialog box
17	Select the date from the calendar for the “Introduced Date” input field in the “Add New Queen” dialog box and “Replaced Date” input field in the “Replace Current Queen” dialog box
18	View the added current queen details
19	Click on the “+” button to add a new note about the current queen
20	Click on the “Replace Queen” button to replace the current queen
21	Add a new note for the current queen from the popped-up dialog box
22	Add the details about replacing the current queen in the popped-up dialog box
23	Click on the “Play/ Pause” button to start and stop sound recording through the device
24	Click on the “Play” button to play a recorded sound

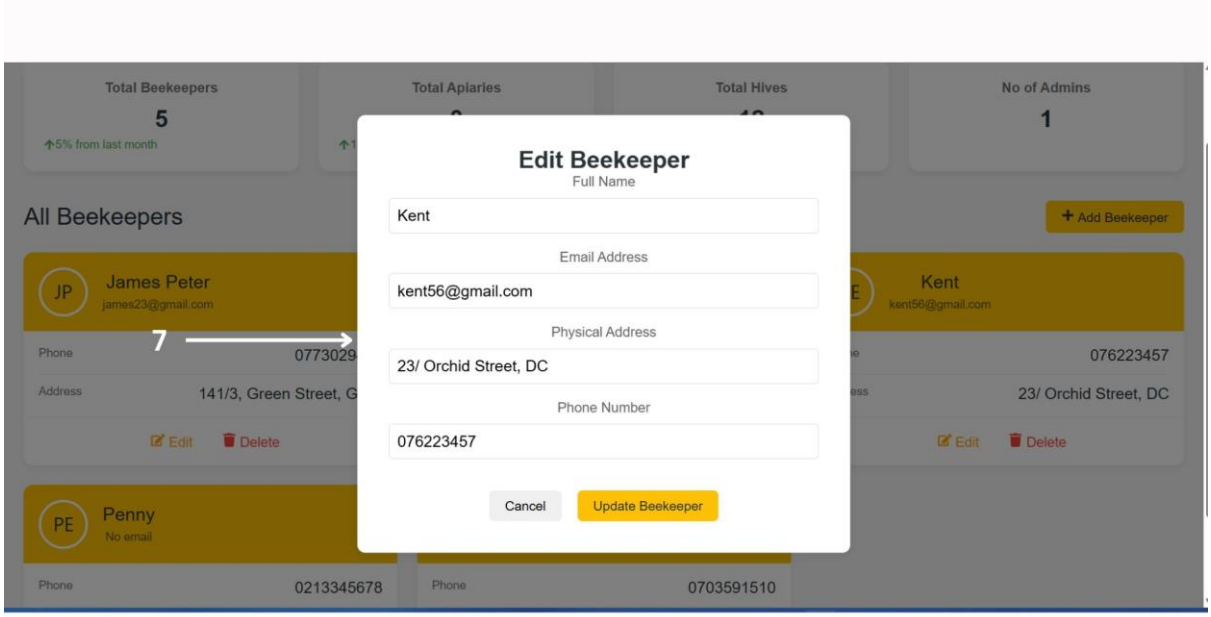
25	Click on the “Classify Audio” button to get the prediction result (queenless or queenright) from the recorded bee sounds
26	Adjust the voice level of the recorded bee sounds
27	View the prediction result for the recorded audio sample

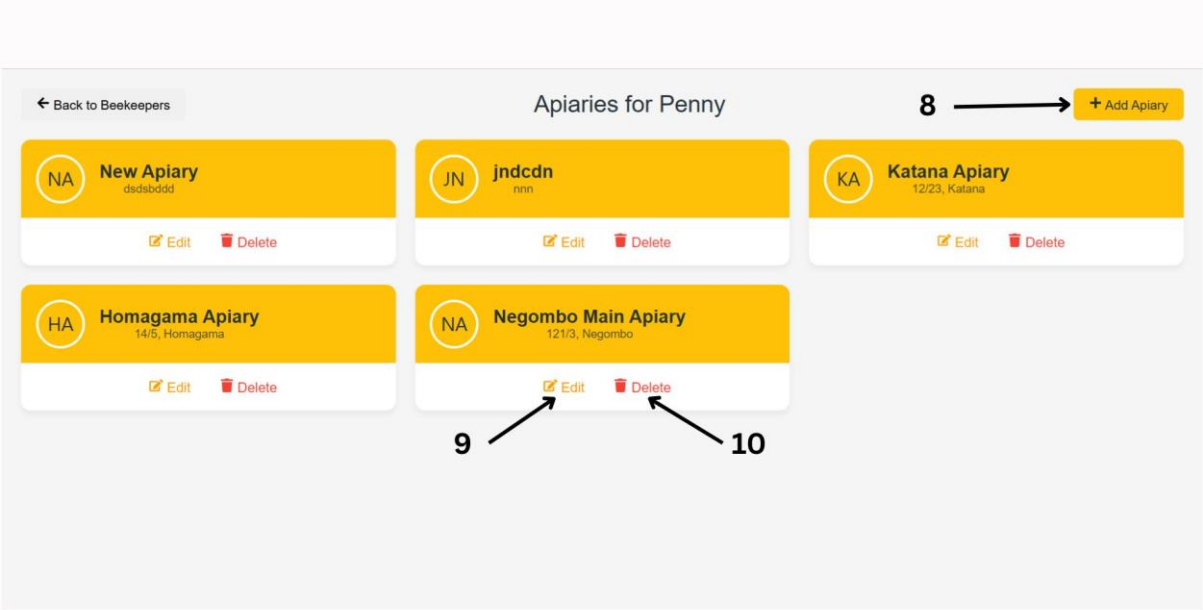
B.6 How to use the admin web application

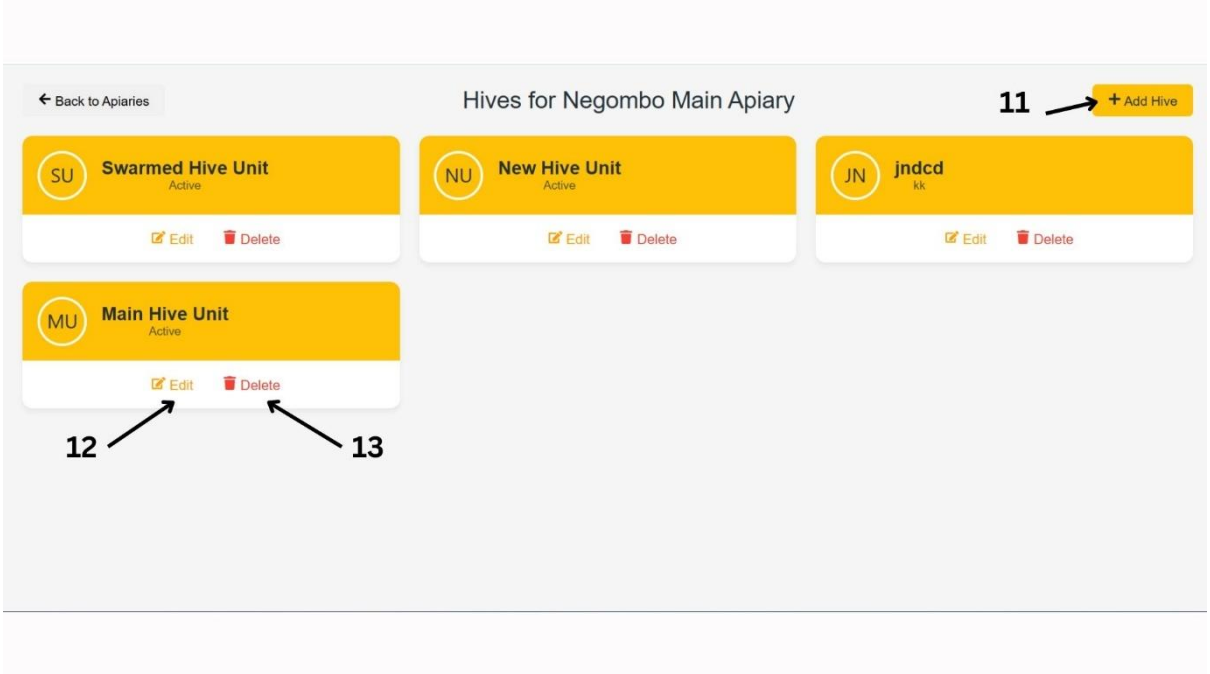



Beekeeper Management
Admin
Total Beekeepers
5
5% from last month
Total Apiaries
0
12% from last month
Total Hives
12
3% from last month
No of Admins
1
All Beekeepers
3
+ Add Beekeeper
JP
James Peter
james23@gmail.com
Phone
0773029456
Address
141/3, Green Street, Galle
Edit
Delete
NL
Nisal Liyanagebbbbbbbbbb
nisal.sarangaggg1234@gmail.com
Phone
0703591510
Address
151/3, Kongodamulla
Edit
Delete
KE
Kent
kent56@gmail.com
Phone
076223457
Address
23/ Orchid Street, DC
Edit
Delete
5
4
Penny
Nisal Liyanage


5
5% from last month
0
12% from last month
12
3% from last month
1
All Beekeepers
+ Add Beekeeper
JP
James Peter
james23@gmail.com
Phone
0773029
Address
141/3, Green Street, G
Edit
Delete
6
Add Beekeeper
Full Name
Email Address
Physical Address
Phone Number
Cancel
Create Beekeeper
Kent
kent56@gmail.com
076223457
23/ Orchid Street, DC
Edit
Delete
PE
Penny
No email
Phone
0213345
Address
131/Negombo Street
Address
151/3, Kongodamulla


Total Beekeepers
5
5% from last month
Total Apiaries
Total Hives
No of Admins
1
All Beekeepers
+ Add Beekeeper
JP
James Peter
james23@gmail.com
Phone
0773029
Address
141/3, Green Street, G
Edit
Delete
7
Edit Beekeeper
Full Name
Kent
Email Address
kent56@gmail.com
Physical Address
23/ Orchid Street, DC
Phone Number
076223457
Cancel
Update Beekeeper
Kent
kent56@gmail.com
076223457
23/ Orchid Street, DC
Edit
Delete
PE
Penny
No email
Phone
0213345678
Phone
0703591510


Back to Beekeepers
Apiaries for Penny
8
+ Add Apiary
NA
New Apiary
dsdsbddd
Edit
Delete
JN
jndcdn
nnn
Edit
Delete
KA
Katana Apiary
12/23, Katana
Edit
Delete
HA
Homagama Apiary
14/5, Homagama
Edit
Delete
NA
Negombo Main Apiary
121/3, Negombo
Edit
Delete
9
10



Index	Description
1	Enter the admin credentials
2	Click on “Sign in” Button to log into the system
3	On the dashboard page click on the “Add Beekeeper” to add a new beekeeper
4	On the dashboard page click on the “Edit” to edit any beekeeper
5	On the dashboard page click on the “Delete” to delete any beekeeper
6	Add new beekeeper details through the “Add Beekeeper” dialog box
7	Add the details of the selected beekeeper that you want to edit using the “Edit Beekeeper” dialog box.
8	On the apiaries page click on the “Add Apiary” to add a new apiary (“Add Apiary” dialog box will be popped up)
9	On the apiaries page click on the “Edit” to edit any apiary (“Edit Apiary” dialog box will be popped up)

10	On the apiaries page click on the “Delete” to delete any apiary
11	On the hives page click on the “Add Hive” to add a new hive (“Add Hive” dialog box will be popped up)
12	On the hives page click on the “Edit” to edit any hive (“Edit Hive” dialog box will be popped up)
13	On the hives page click on the “Delete” to delete any hive

Appendix C: Code snippets

C++ Code to process raw humidity data from the DHT22 sensor:

```
float processDHTHumidity(float rawHumidity) {
    // Process humidity data
    if (isnan(rawHumidity) || rawHumidity < 0 || rawHumidity > 100) {
        return 0.0; // Invalid reading
    }


    // Clip to valid range
    if (rawHumidity > 100) rawHumidity= 100;

    // Round to whole number
    return round(rawHumidity * 100)/100.0;
}

float processWeight(float rawWeight) {
    // Process weight data
    if (rawWeight < -1000 || rawWeight > 5000) { // Adjust these thresholds based on your application
        return 0.0; // Invalid reading
    }


    // Round to 2 decimal place
    return round(rawWeight * 100) / 100.0;
}
```

Python code the segment the audio samples to one-second-long clips:

```python
# Process each audio file in the label folder
for file_name in os.listdir(label_path):
    if file_name.endswith('.wav'):
        file_path = os.path.join(label_path, file_name)

        # Load audio file
        y, sr = librosa.load(file_path, sr=sample_rate)  # Ensure 22.05 kHz

        # Split audio into one-second segments
        num_segments = len(y) // segment_length  # Number of one-second segments

        for i in range(num_segments):
            # Extract a one-second segment
            segment = y[i * segment_length: (i + 1) * segment_length]

            # Save the segment with a unique filename in the corresponding output folder

            segment_filename = f'{file_name}_{i}.wav'
            segment_path = os.path.join(output_folder, hive_name, label_folder, segment_filename)
            os.makedirs(os.path.dirname(segment_path), exist_ok=True)


            # Write segment to file
            sf.write(segment_path, segment, sample_rate)
            print(f'Saved segment: {segment_path} with label {label_folder}')
```

Python code to extract MFCC features:

```python
def extract_mfcc(audio_path):
    audio, sr = librosa.load(audio_path, sr=sample_rate)
    mfcc = librosa.feature.mfcc(y=audio, sr=sr, n_mfcc=n_mfcc)
    if mfcc.shape[1] < max_frames:
        pad_width = max_frames - mfcc.shape[1]
        mfcc = np.pad(mfcc, pad_width=((0, 0), (0, pad_width)), mode='constant')
    else:
        mfcc = mfcc[:, :max_frames]
    return mfcc
```

Python code to handle class imbalance for CNN:

```python
# Calculate class weights to handle imbalance
class_weights = class_weight.compute_class_weight(
    class_weight='balanced',
    classes=np.unique(labels),
    y=labels
)
class_weights_dict = {i: weight for i, weight in enumerate(class_weights)}
print("Class weights:", class_weights_dict)
```

Python code to initialize 5-Fold cross validation:

```
# Initialize 5-fold cross validation
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

Python code to create the CNN model:

```
# Function to create model
def create_model():
    model = Sequential([
        Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(n_mfcc, max_frames, 1)),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(64, kernel_size=(3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        Flatten(),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(2, activation='softmax')
    ])

    model.compile(
        optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )
    return model
```

Python code to train the one class SVM on healthy data samples:

```
# Train the One-Class SVM
svm = OneClassSVM(kernel='rbf', gamma='auto', nu=0.05)
svm.fit(X_train_scaled)
```

Python code to analyze health model's performance:

```
# Analyze false positive rate on test data
test_false_positives = (df.loc[test_indices, 'prediction'] == -1).sum()
test_false_positive_rate = test_false_positives / len(test_indices)
print(f"False Positive Rate on test data: {test_false_positive_rate:.2%}")

print(df["prediction"])

# Determine optimal threshold based on desired false positive rate
# Sort scores in ascending order
sorted_scores = np.sort(df.loc[test_indices, 'score'])
# Set threshold to achieve approximately 1% false positive rate
threshold_index = int(0.01 * len(sorted_scores))
threshold = sorted_scores[threshold_index]
print(f"Recommended threshold for ~1% false positive rate: {threshold:.4f}")
```

Python code in flask to get the prediction for a audio sample using the saved queen detection model:

```python
try:
    if not os.path.exists(audio_file_path):
        print(f"❌ Invalid or missing audio file: {audio_file_path}")
        return None

    # Extract MFCC features (multiple chunks)
    mfcc_chunks = extract_mfcc(audio_file_path)
    if mfcc_chunks is None:
        return None

    predictions = []
    for mfcc_features in mfcc_chunks:
        features_reshaped = mfcc_features.reshape(1, -1)

        # Scale the features
        scaled_features = queen_scaler.transform(features_reshaped)

        # Reshape back for CNN input: (1, height, width, channels)
        features = scaled_features.reshape(1, N_MFCC, MAX_FRAMES, 1)

        # Predict for this chunk
        y_pred = np.argmax(queen_model.predict(features), axis=1)[0]
        predictions.append(int(y_pred))

    # Return the most common predicted class
    final_prediction = Counter(predictions).most_common(1)[0][0]
    return final_prediction
```

Appendix D: Use Cases Continuation

Use Case ID:	5		
Use Case Name:	View all apiaries		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to view all the apiaries belonging to him.		
Preconditions:	Beekeeper is logged in; Admin has added apiary/s under that beekeeper.		
Postconditions:	Beekeeper can successfully view all his registered apiaries.		
Normal Course of Events:	1. Go to dashboard. 2. View all the registered apiaries.		
Alternative Courses:	1. Can navigate to the same content through the navbar.		
Notes:	• After a successful login, the beekeeper can view all their registered apiaries, which were previously added by the admin. These registered apiaries contain hives where the IoT device was installed by the business.		

Use Case ID:	6		
Use Case Name:	View all hives		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to view all the hives under each apiary		
Preconditions:	Beekeeper is logged in, Beekeeper access a specific apiary, Admin has added hive/s under that apiary.		
Postconditions:	Beekeeper can successfully view all his registered hives under the selected apiary.		
Normal Course of Events:	1. Select a specific apiary. 2. View all the registered hives under that apiary.		
Alternative Courses:	2. Can navigate to the same content through the navbar.		
Notes:	• After selecting a specific apiary, the beekeeper can view all their registered hives under that specific apiary, which were previously added by the admin.		

Use Case ID:	7		
Use Case Name:	View past sensor data in graphs		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage

Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to visualize past sensor data from each hive in plots in order to identify patterns, trends and anomalies.		
Preconditions:	The beekeeper logs in and selects a specific hive. The selected hive has been installed with an IoT module by business, and the module is powered and functions properly.		
Postconditions:	Beekeeper can successfully visualize the past sensor readings in plots.		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. View the selected hive's past sensor data in graphs.		
Alternative Courses:	-		
Notes:	• The sensor readings coming from the sensor module placed on a specific hive should be stored in a database.		

Use Case ID:	**8**		
Use Case Name:	Track a hive's past queens' details.		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to track the details of the past queens of the selected hive.		
Preconditions:	The beekeeper logs in and selects a specific hive. The beekeeper has previously added details of the selected hive's past queen/s.		
Postconditions:	The beekeeper should be able to view all the details regarding the selected hive's past queen/s.		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. Navigate to past queens' section. 4. Track the hive's history of past queens.		
Alternative Courses:	-		
Notes:	-		

Use Case ID:	**9**		
Use Case Name:	Replace the current bee queen and add the details of the new queen.		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025

Actor:	Beekeeper		
Description:	Beekeepers should be able to replace the current bee queen and add details of the new bee queen at the same time through the system.		
Preconditions:	The beekeeper logs in and selects a specific hive. The hive's current queen was lost or removed and has been replaced with a new queen.		
Postconditions:	The current queen's details should be successfully updated with those of the new queen. The replaced queen's information should be accurately recorded in the queen history section.		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. Navigate to the current queens' section. 4. Click on the "Replace Queen" button. 5. Add the reasons for the current queen's replacements. 6. Add the details of the newly added queen. 7. Click on the confirm button 8. Check if the current queen's details are updated and the previous queen's details are added in the queen history section.		
Alternative Courses:	-		
Notes:	• This feature allows the beekeeper to update the current queen's details, ensuring that the hive's queen information remains accurate and up to date.		

Use Case ID:	**10**		
Use Case Name:	Add new notes about the current queen		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	Beekeeper		
Description:	Beekeepers should be able to add frequent new notes regarding the current queen's health and productivity.		
Preconditions:	The beekeeper logs in and selects a specific hive.		
Postconditions:	The added notes should be applied to the current queen and must be displayed on the UI.		
Normal Course of Events:	1. Select a specific hive from the list. 2. Click on the corresponding hive card to open its details. 3. Navigate to the current queens' section. 4. Click on the "Add Comment" button to add a new note 5. Write down a comment in the given text field. 6. Click on the "Confirm" button. 7. Check if the current queen's details are updated with the newly added notes.		
Alternative Courses:	-		
Notes:	• This feature allows the beekeeper to add comments about the queen, such as "Egg-laying normal," "Aggressive behavior," "Reduced brood pattern," etc.		

Use Case ID:	11		
Use Case Name:	Manage beekeeper details		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	The admin		
Description:	The admin should be able to register a new beekeeper, edit his details or delete details upon the beekeeper request.		
Preconditions:	The admin is logged in.		
Postconditions:	The admin should be successfully able to add, edit or delete beekeeper details.		
Normal Course of Events:	1. Navigate to beekeeper section. 2. Perform CRUD operations on beekeeper collection.		
Alternative Courses:	-		
Notes:	-		

Use Case ID:	12		
Use Case Name:	Manage apiary details		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025
Actor:	The admin		
Description:	The admin should be able to register a new apiary under a specific beekeeper, edit its details or delete details upon the beekeeper request.		
Preconditions:	The admin is logged in.		
Postconditions:	The admin should be successfully able to add, edit or delete apiary details.		
Normal Course of Events:	1. Navigate to the apiary section. 2. Perform CRUD operations on apiary collection.		
Alternative Courses:	-		
Notes:	-		

Use Case ID:	13		
Use Case Name:	Manage hive details		
Created By:	Nisal Liyanage	Last Updated By:	Nisal Liyanage
Date Created:	4/13/2025	Date Last Updated:	4/15/2025

Actor:	The admin		
Description:	The admin should be able to register a new hive under a specific apiary of a beekeeper, edit its details or delete details upon the beekeeper request.		
Preconditions:	The admin is logged in.		
Postconditions:	The admin should be successfully able to add, edit or delete hive details.		
Normal Course of Events:	1. Navigate to the hive section. 2. Perform CRUD operations on hive collection.		
Alternative Courses:	-		
Notes:	-		

Appendix E: Unit Test Cases

Test Case ID	Unit Test Case 01
Test Case Description	This test case verifies proper wiring of sensors with the ESP32
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	The circuit diagram for the sensors and ESP32 have been designed.
Test Steps	1. Inspect wiring according to circuit diagram. 2. Use a multimeter to measure current flow.
Pos/ Neg	Positive
Input Data	NA
Expected Results	Proper wiring with no loose solder joints
Actual Results	No loose solder joints were noticed, wiring and soldering was done according to the designed circuit diagram.
Status	Pass
Defect ID	NA

Test Case ID	Unit Test Case 02
Test Case Description	This test case verifies whether the solar panel outputs proper voltage to LM2596
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	Solar panel has been connected with the LM2596
Test Steps	1. Expose the panel to sunlight 2. Measure voltage output
Pos/ Neg	Positive
Input Data	NA
Expected Results	Output voltage between 16V-18V
Actual Results	Solar panel outputs a 17.4 V voltage
Status	Pass
Defect ID	NA

Test Case ID	Unit Test Case 03
Test Case Description	This test case verifies the parallel battery output of the 2 lithium-ion batteries
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	The two lithium-ion batteries are connected parallelly
Test Steps	1. Measure output of connected batteries

Pos/ Neg	Positive
Input Data	NA
Expected Results	Output ~3.7V-4.2V depending on charge level
Actual Results	The battery outputs 4.0V
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 04**
Test Case Description	This test case verifies load cell calibration correctness
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. The load cell is connected to HX711. 2. The load cell is properly mounted 3. The load cell is connected to the ESP32
Test Steps	1. Flash ESP32 with calibration firmware 2. Place known weight 3. Read calibration factor 4. Place another weight and get the weight reading
Pos/ Neg	Positive
Input Data	NA
Expected Results	The new weight reading is given accurately
Actual Results	The new weight was given correctly
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 05**
Test Case Description	This test case verifies the data processing pipeline's correctness on the raw DHT22 sensor data.
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. DHT22 is connected to ESP32
Test Steps	1. Flash the ESP32 with the code 2. Make the DHT22 data pin outputs null few times by disconnecting the data wire.
Pos/ Neg	Negative
Input Data	NA
Expected Results	The raw data should be processed correctly, and null values should be eliminated
Actual Results	The null values were eliminated, and the raw data were processed as expected
Status	Pass
Defect ID	NA

Test Case ID	Unit Test Case 06
Test Case Description	This test case verifies the data processing pipeline's correctness on the raw weight data.
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. The load cell is connected to ESP32.
Test Steps	1. Flash ESP32 with the code 2. Place a weight on the load cell 3. Make the HX711 data pin outputs null few times by disconnecting the data wire.
Pos/ Neg	Negative
Input Data	NA
Expected Results	The raw weight values are retrieved and processed correctly.
Actual Results	The null values were eliminated, and the weight raw data were processed as expected
Status	Pass
Defect ID	NA

Test Case ID	Unit Test Case 07
Test Case Description	This test case verifies deep sleep functionality after data upload
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Final firmware is uploaded to the ESP32
Test Steps	1. Monitor ESP32 current usage 2. Check sleep entry
Pos/ Neg	Positive
Input Data	NA
Expected Results	ESP32 enters deep sleep mode for 15 minutes
Actual Results	ESP32 entered deep sleep for 15 minutes as expected
Status	Pass
Defect ID	NA

Test Case ID	Unit Test Case 08
Test Case Description	This test case verifies if the ESP32 is able to connect with the WiFi.
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. WiFi is on

	2. WiFi password and username are given to the firmware 3. Proper debug commands are added in the firmware. 4. ESP32 is flashed with firmware
Test Steps	1. Check for the serial monitor logs
Pos/ Neg	Positive
Input Data	NA
Expected Results	ESP32 should be successfully connected to the Wi-Fi
Actual Results	ESP32 successfully connected to the Wi-Fi
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 09**
Test Case Description	This test case verifies correct extraction of MFCC features
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Dataset is preprocessed
Test Steps	1. Extract the MFCC features on each preprocessed audio data 2. Check for the shape of the extracted features
Pos/ Neg	Positive
Input Data	NA
Expected Results	Each audio segment should have a valid MFCC feature vector.
Actual Results	Valid MFCC feature vectors have been retrieved from all the audio samples as expected
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 10**
Test Case Description	This test case verifies that the saved queen model can be loaded correctly without data loss
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Model was trained 2. Model was saved as ‘h5’ file.
Test Steps	1. Reload the saved model 2. Test the model’s prediction ability with a sample input audio
Pos/ Neg	Positive
Input Data	NA
Expected Results	The model should be able to give the correct prediction class for the given input audio

Actual Results	The model successfully gave the correct prediction class for the given audio as expected.
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 11**
Test Case Description	This test case verifies that the saved health model can be loaded correctly without data loss
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Model was trained 2. Model was saved as 'joblib' file.
Test Steps	1. Reload the saved model 2. Test the model's prediction ability with a sample input array that contains various values for hive related data for the healthy scenario
Pos/ Neg	Positive
Input Data	NA
Expected Results	The model should be able to give the correct prediction as healthy
Actual Results	The model gave the prediction as healthy
Status	Pass
Defect ID	NA

Test Case ID	**Unit Test Case 12**
Test Case Description	This test case verifies that the audio can be successfully recorded through the device's mic
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. The permission to accept the mic for the app has been given
Test Steps	1. Click the record button to record audio 2. Play the audio.
Pos/ Neg	Positive
Input Data	NA
Expected Results	The audio played should be exactly as it was recorded.
Actual Results	The audio played back exactly as it was recorded.
Status	Pass
Defect ID	NA

Appendix F: Integration Test Cases

Test Case ID	**Integration Test Case 01**
Test Case Description	This test case verifies the integration between firmware and the firebase Realtime database
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Final firmware has been flashed to the ESP32 2. Firebase Realtime database has been created. 3. Access was given to the firmware to perform data passing to the Realtime database
Test Steps	1. Give WiFi connection to the ESP32 2. Check if the live sensor data are sending to the firebase Realtime database
Pos/ Neg	Positive
Input Data	NA
Expected Results	Live sensor data sent from the ESP32 are getting saved in the Realtime database.
Actual Results	The live sensor data got stored in the Realtime database as expected
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 02
Test Case Description	This test case verifies the integration between the firebase Realtime database and the beekeeper mobile application
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Realtime database has sensor data 2. The beekeeper mobile app was given permission to access the Realtime database.
Test Steps	1. Logs into the mobile application 2. Check whether it is possible to see both the live data and past data on the mobile app's UIs.
Pos/ Neg	Positive
Input Data	NA
Expected Results	Live sensor data and past data should be displayed on the mobile app.
Actual Results	Both live and past data were displayed on the mobile app as expected.
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 03

Test Case Description	This test case verifies the integration between the queen detection model and the flask server
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	The saved model has been integrated with the flask server
Test Steps	1. Runs the flask server 2. Make a HTTP request with a bee sounds audio file to the flask's queen detection API endpoint from Postman 3. Check the response results
Pos/ Neg	Positive
Input Data	NA
Expected Results	The correct prediction result should be received in the response without any error
Actual Results	The correct prediction result was received as expected
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 04
Test Case Description	This test case verifies the integration between the health analysis model and the flask server
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	The saved model has been integrated with the flask server
Test Steps	1. Runs the flask server 2. Make a HTTP request with an array of sample sensor data to the flask's health analysis API endpoint from Postman 3. Check the response results
Pos/ Neg	Positive
Input Data	NA
Expected Results	The correct prediction result should be received in the response without any error
Actual Results	The correct prediction result was received as expected
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 05
Test Case Description	This test case verifies the integration between the mobile application and the flask server via the /predict_queen API endpoint
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author

Test Executed Date	07/04/25
Preconditions	1. Flask server is running 2. Both the app and the server have been connected to the same Wi-Fi 3. An audio sample has already been recorded by the mobile app
Test Steps	1. Click on the predict audio button in the mobile app 2. Check the prediction result displayed on the app
Pos/ Neg	Positive
Input Data	NA
Expected Results	The correct prediction result should be displayed on the app
Actual Results	The correct prediction result displayed on the app as expected
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 06
Test Case Description	This test case verifies the integration between the mobile application and the flask server via the /analyze_health API endpoint
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. Flask server is running 2. Both the app and the server have been connected to the same Wi-Fi 3. Realtime database has live sensor data
Test Steps	1. Logs into the app 2. Select an apiary and then a hive 3. Click on live data button 4. Check if the current health status of the hive is displayed on the app's UI
Pos/ Neg	Positive
Input Data	NA
Expected Results	The correct health status should be displayed on the app
Actual Results	The correct health status displayed on the app as expected
Status	Pass
Defect ID	NA

Test Case ID	Integration Test Case 07
Test Case Description	This test case verifies the integration between the mobile app and firebase authentication
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. A user has been created

	2. The app has been granted permission to use firebase authentication feature
Test Steps	1. Add the username for the authenticated user 2. Add the password for the authenticated user 3. Click on the login button
Pos/ Neg	Positive
Input Data	Login credentials
Expected Results	Successful login
Actual Results	The login was successful as expected
Status	Pass
Defect ID	NA

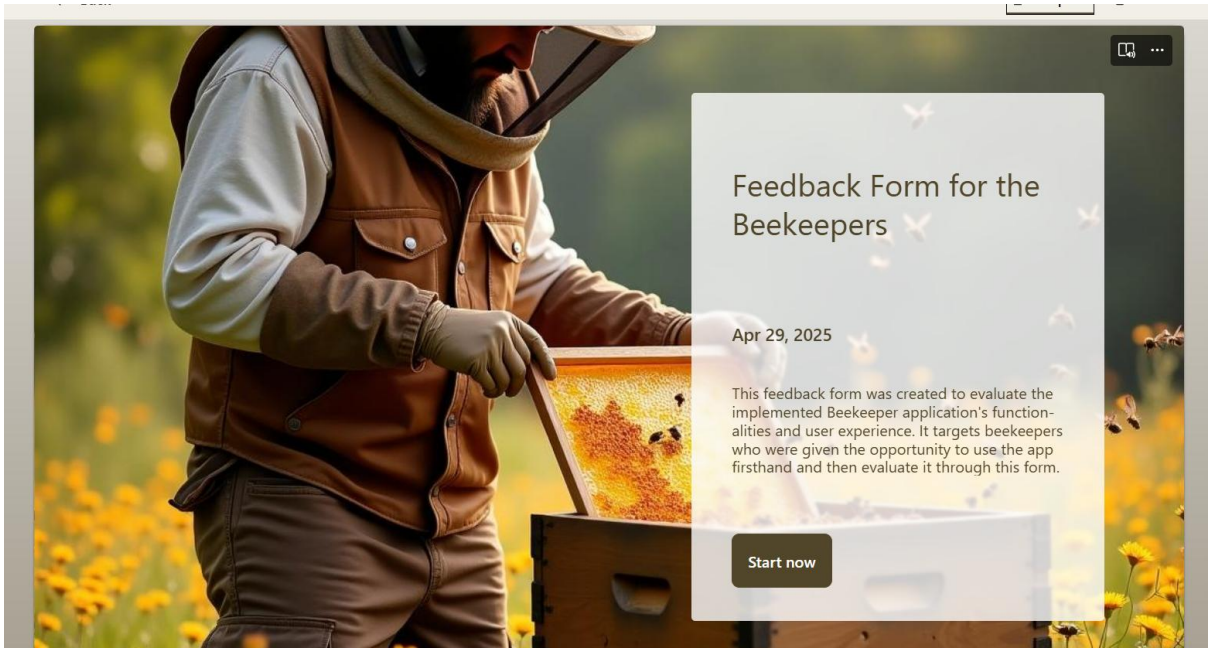
Test Case ID	Integration Test Case 08
Test Case Description	This test case verifies the integration between the mobile app and firestore database
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. A user has been created 2. Data for the created user has been added to the firestore database 3. The app has been granted permission to access the data in the firestore database
Test Steps	1. Log in to the mobile app as an authenticated user. 2. Verify that all apiary and hive details associated with the user are accessible.
Pos/ Neg	Positive
Input Data	NA
Expected Results	Apiary and hive details associated with the logged in user should be displayed on the app
Actual Results	The apiary and hive details of the logged in user are successfully displayed on the app
Status	Pass
Defect ID	NA

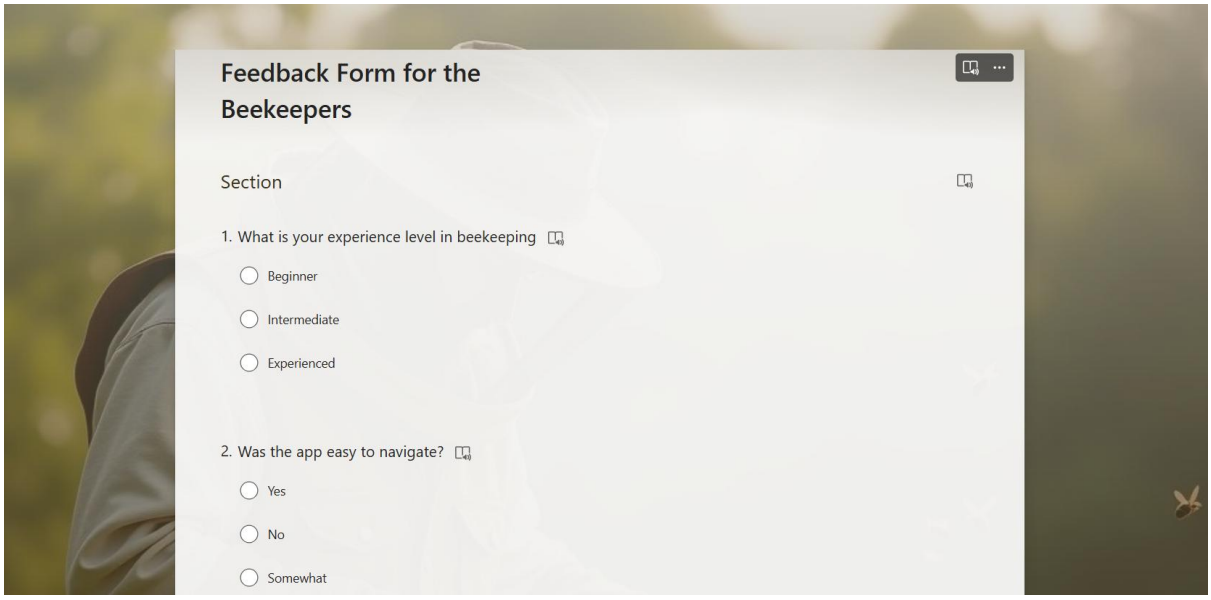
Test Case ID	Integration Test Case 09
Test Case Description	This test case verifies the integration between the web app and firestore database
Test Designed By	Author
Test Designed Date	07/03/25
Test Executed By	Author
Test Executed Date	07/04/25
Preconditions	1. A firebase user for the admin has already been created. 2. Firestore database contains data 3. Web app has access to the firestore database
Test Steps	4. Logs into the web app with admin credentials

	5. Check whether CRUD operations can be performed on the beekeeper, apiary and hive collections on the firestore database
Pos/ Neg	Positive
Input Data	NA
Expected Results	CRUD operations on the beekeeper, apiary, and hive collections should be successfully performed through the web app
Actual Results	The CRUD operations on the collections were able to be performed as expected.
Status	Pass
Defect ID	NA

Appendix G: Feedback Form & Results

Feedback Form:





3. Were all the buttons, icons, and labels clear and understandable?

- Yes
- No
- Somewhat

4. Was the interface visually appealing and comfortable to use?

- Yes
- Needs Improvement

5. Were you able to log in successfully using the given credentials?

- Yes
- No

6. After logging in, did the app show only your own apiary and hive data?

- Yes
- No

7. Were you able to view real-time hive sensor data correctly?

- Yes
- No

8. How satisfied are you with the feature that displays real-time hive health status?

- Very satisfied
- Somewhat satisfied
- Very dissatisfied

9. How satisfied are you with the feature that indicates whether the hive is ready for harvest in real time?

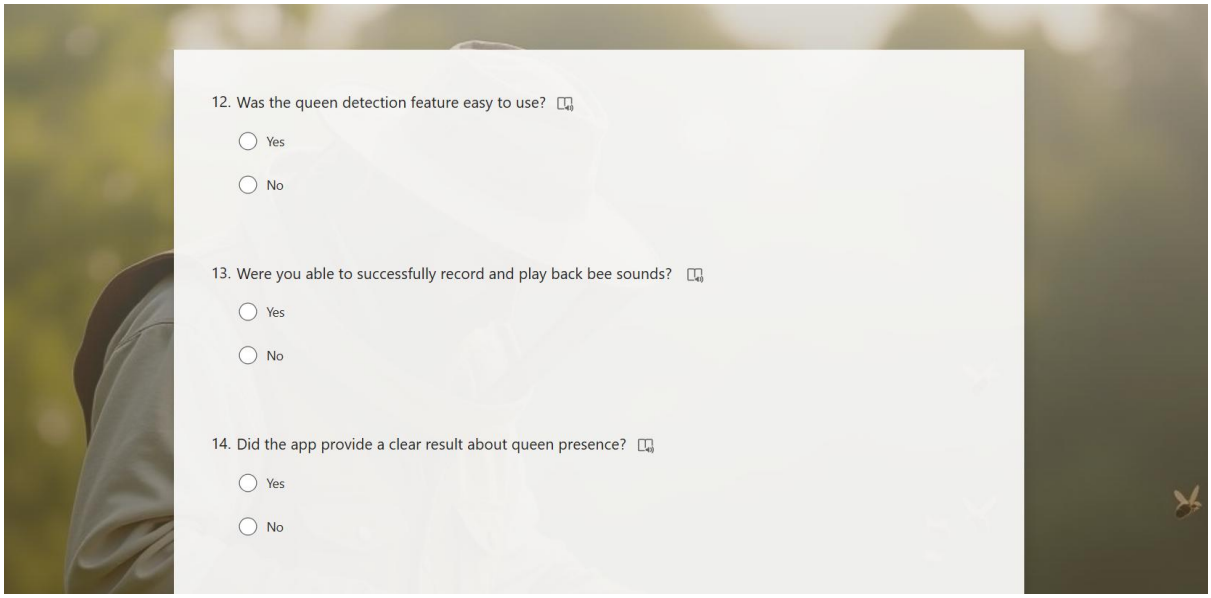
- Very satisfied
- Somewhat satisfied
- Very dissatisfied

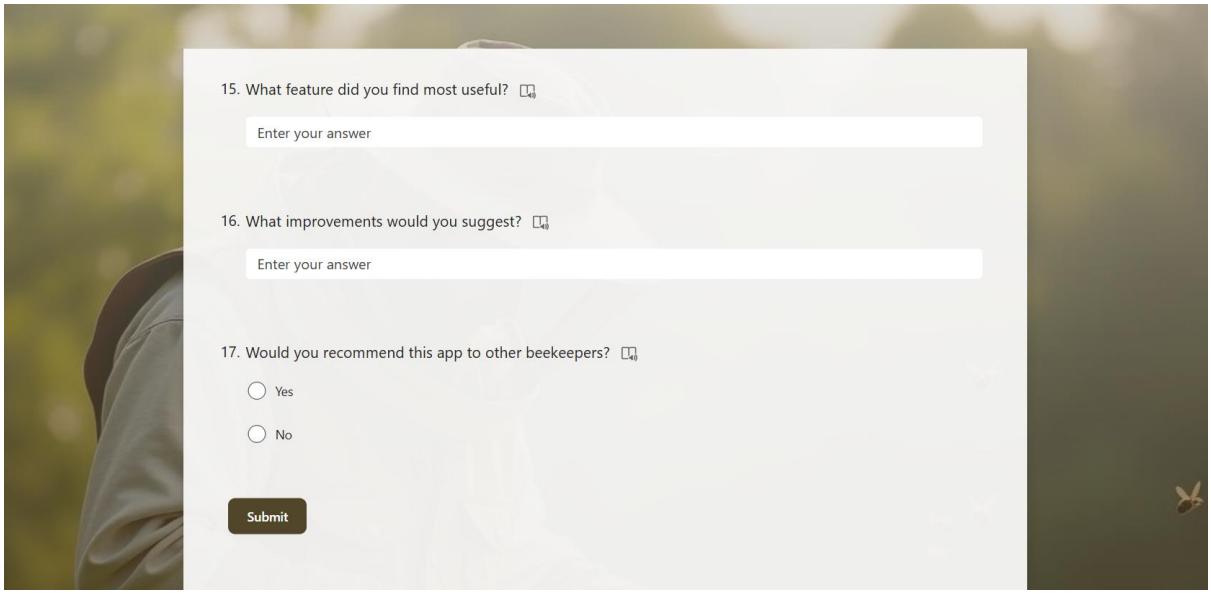
10. Did the charts/graphs of past data help you understand hive trends?

- Yes
- No

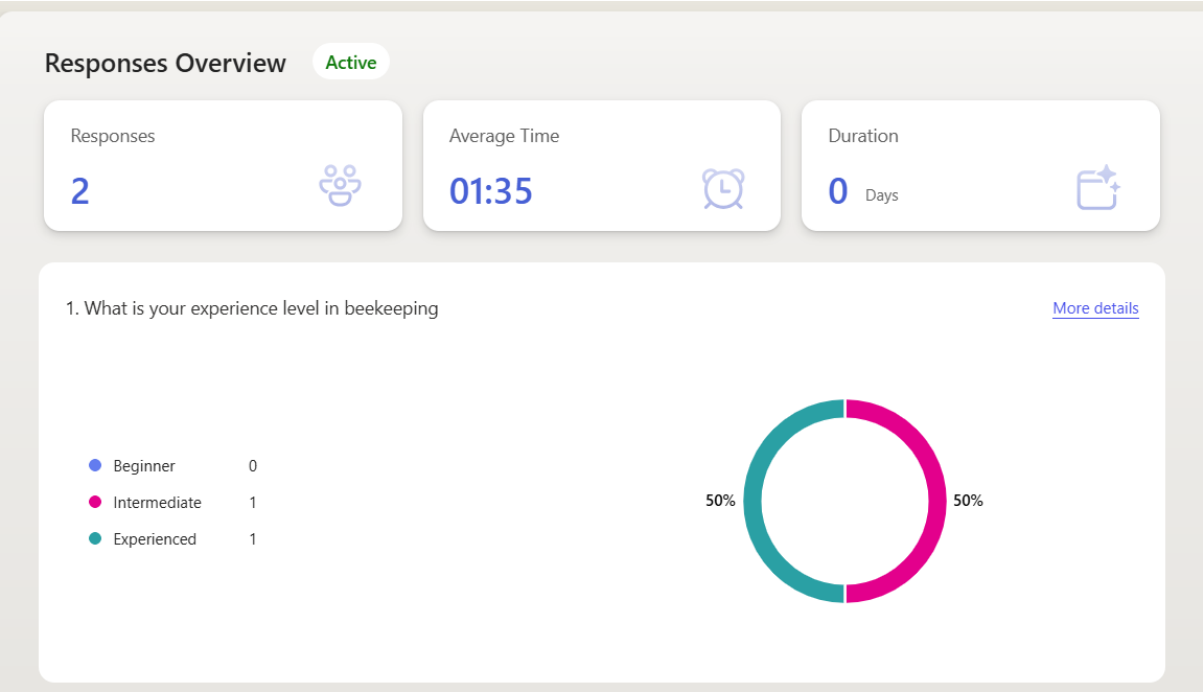
11. Was the feature to log and view current and past queen details helpful?

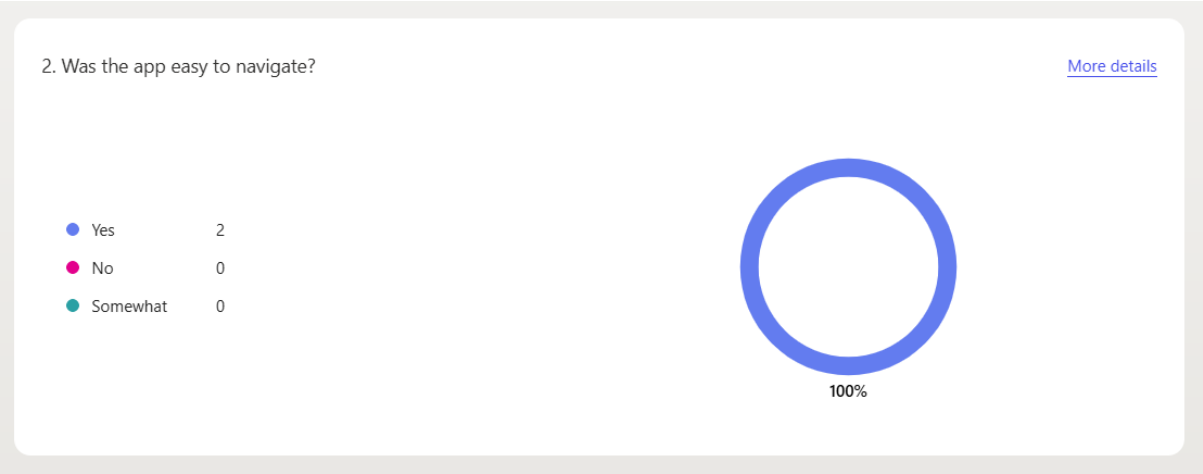
- Yes
- No


12. Was the queen detection feature easy to use?
Yes
No
13. Were you able to successfully record and play back bee sounds?
Yes
No
14. Did the app provide a clear result about queen presence?
Yes
No


15. What feature did you find most useful?
Enter your answer
16. What improvements would you suggest?
Enter your answer
17. Would you recommend this app to other beekeepers?
Yes
No
Submit

Feedback Form Results:

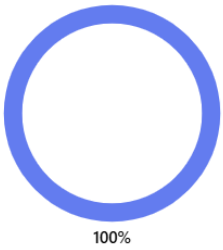




4. Was the interface visually appealing and comfortable to use?

More details

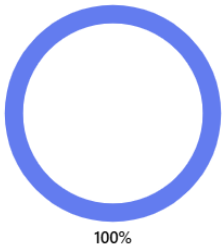
Yes	2
Needs Improvement	0



5. Were you able to log in successfully using the given credentials?

More details

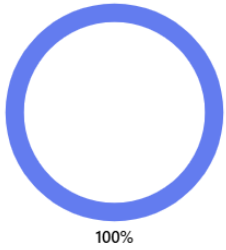
Yes	2
No	0



6. After logging in, did the app show only your own apiary and hive data?

More details

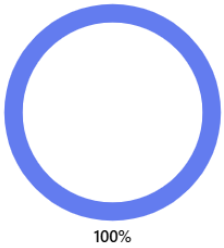
Yes	2
No	0



7. Were you able to view real-time hive sensor data correctly?

More details

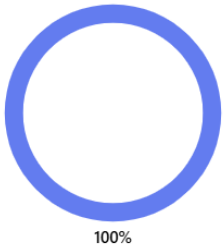
Yes	2
No	0



8. How satisfied are you with the feature that displays real-time hive health status?

More details

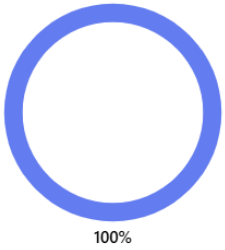
Very satisfied	2
Somewhat satisfied	0
Very dissatisfied	0



9. How satisfied are you with the feature that indicates whether the hive is ready for harvest in real time?

More details

Very satisfied	2
Somewhat satisfied	0
Very dissatisfied	0



10. Did the charts/graphs of past data help you understand hive trends?

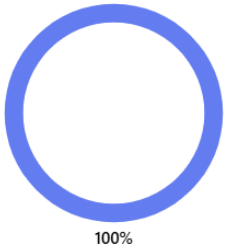
More details

Yes	1
No	1

11. Was the feature to log and view current and past queen details helpful?

More details

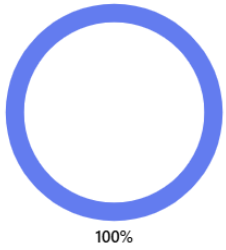
Yes	2
No	0



12. Was the queen detection feature easy to use?

More details

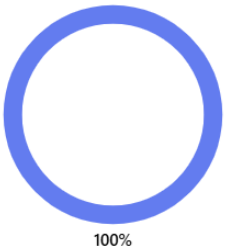
Yes	2
No	0



13. Were you able to successfully record and play back bee sounds?

More details

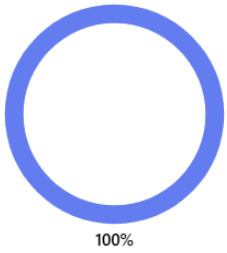
Yes	2
No	0



14. Did the app provide a clear result about queen presence?

More details

Yes	2
No	0



15. What feature did you find most useful?

More details

2
Responses

Latest Responses
"Live hive health update"
"Queen detection using bee sounds"

16. What improvements would you suggest?

More details

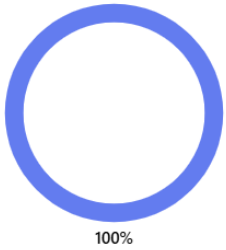
2
Responses

Latest Responses
"A UI that contains proper guide to use the app"
"Sinhala language support"

17. Would you recommend this app to other beekeepers?

More details

Yes	2
No	0



Appendix H: Supporting Project Documents

H.1 Project Proposal

PUSL3190 Computing Individual Project

Project Proposal

A Sensor-Based System for Advanced Beehive Monitoring that utilizes Signal Processing and Time Series Forecasting

Supervisor: Ms. Hirushi Dilpriya

Name: Nisal Liyanage
Plymouth Index Number: 10898808
Degree Program: BSC (HONS) Computer Science

1

Chapter 01 | Problem Definition

Bees play a crucial role in sustaining life on Earth and maintaining the balance of our ecosystem. Albert Einstein emphasized this significance with his famous quote: “If the bee disappeared off the surface of the globe, man would only have four years of life left.”. Bees are the largest pollinators on the planet, contributing to over 33% of the global food supply. They help fertilize and cross-pollinate many plant species, which is essential for the reproduction of various plants, including wildflowers, fruits, vegetables, and nuts like almonds. Without bees, food security for both humans and animals would be at risk.

In addition to their role as pollinators, bees produce valuable products for humans, such as honey, royal jelly, pollen, beeswax, and honeycombs. Beekeeping industry focuses on harvesting these precious products. Honey, in particular, is not only a nutritious food source but also has uses in traditional medicine.

The beekeeping industry represents a thriving sector within Sri Lanka's agricultural landscape, offering significant potential for both economic growth and environmental stewardship. This industry, with its plenty of benefits, can contribute substantially to the nation's development.

However, in recent years, bee populations around the world as well as in Sri Lanka have faced significant declines in health and numbers. This is evident through reduced honey yields, smaller colony sizes, higher mortality rates among worker bees, infertility in queen bees, and the occurrence of colony collapse disorder. Many factors contribute to this decline, including air pollution, widespread malnutrition, harmful pesticides, deforestation, and other human activities. In the beekeeping industry, poor monitoring practices can further harm bee populations.

Therefore, continuous monitoring of beehives is essential to ensure the health of the colony. This includes checking for pests or diseases, confirming the queen bee's presence, and ensuring there is enough honey for extraction.
Unfortunately, traditional hive monitoring methods used in Sri Lanka present several challenges for beekeepers and bees. These methods are often time-consuming, inefficient,

and risky, as beekeepers may get stung while inspecting hives. Additionally, many beekeepers overlook regular hive monitoring due to their busy schedules. Traditional practices can also make it difficult to gather reliable information about the health of the bee colony and often ignore important environmental factors like temperature and humidity. As a result, the quality and quantity of honey can suffer due to insufficient monitoring of hive conditions, and these manual methods can be intrusive, causing stress to the bees.

Therefore, there is a pressing need for non-intrusive methods to monitor beehives remotely and accurately, benefiting both beekeepers and bees.

Chapter 02 | Project Description

2.1 Proposed Solution

In order to provide a non-intrusive solution for monitoring beehives, a computerized system incorporating IoT technology, data analysis, mobile and web development, as well as AI and ML-based approaches, is proposed.

2.2 Project Scope

The proposed system uses various sensors to monitor humidity, air quality, and temperature inside the hive. By incorporating internal hive parameters with external factors such as rainfall, outside humidity, and outside temperature, the system aims to monitor hive health by detecting anomalies in temperature, humidity, and air quality patterns inside the hive. Hive weight is used to determine the optimal time for harvesting and assess the overall growth of the colony. Anomalies in hive weight can also indicate potential issues with the hive's health.

The IoT-based embedded system with sensors will be placed in a real beehive to create a dataset that will be further analyzed in order to deduce insights into hive health and bee activities. Furthermore, previous research has shown that bees change their sounds when they lose their queen. Therefore, the project will exploit an available dataset containing

acoustic signals, utilizing signal processing and an AI-based classification model to predict the absence of the queen.

The developed model and the hive insights derived from the data analysis will be displayed on a mobile application, allowing beekeepers to remotely monitor hive conditions. Additionally, beekeepers can detect the queen's presence or absence by recording sounds from the beehive using their mobile phone's microphone, which will be analyzed through the developed application to determine the absence or presence of the queen bee. Furthermore, the mobile application will also be able to detect swarming if it occurs. However, to predict pre-swarming behavior, data collected from a real swarm before and after the event is needed, so this aspect will not be included in the current scope.

2.3 Objectives

With respect to the proposed sensor-based system for advanced beehive monitoring:

- **Main Objective**
    1. To provide beekeepers with a mobile application to remotely monitor their beehives.
- **Specific Objectives**
    1. To track the normal honeybee activities and hive conditions during the day (e.g., temperature variations during the day).
    2. To visualize live data from the hive.
    3. To offer historical data analysis and trend forecasting to support informed decision-making.
    4. To deliver insights on hive health by detecting anomalies in normal bee activities and hive conditions.
    5. To provide insights on the optimal time for harvesting.
    6. To automatically detect the presence of the queen bee using acoustic signals.
    7. To automatically record any swarming events that occur.

8. To maintain a record of the queen bee's history within the hive.

2.4 Target Users

The proposed solution primarily targets Sri Lankan beekeepers, enabling them to monitor their hives remotely and conveniently while minimizing the need for manual hive monitoring. The system provides insights into hive health, colony growth, nectar flow, queen presence/absence, swarming events, and more.

Furthermore, the proposed system also targets bee scientists and researchers. These professionals can utilize the system to monitor beehives more efficiently, allowing them to gather important data about bee activities and behaviors.

Keywords: IOT Beehive, Smart Beehive, Mobile Application for beehive monitoring, Bee Queen Detection, IOT in Beekeeping, Sri Lankan Beekeeping, Queen Bee, Beehive data analysis, beehive insights, swarming detection, AI and ML based approaches for beehive data analysis.

Chapter 03 | Research Gap

Human beekeeping has roots that go all the way back to ancient Egypt and ancient Greece (Whitfield, 1956). As the consumption of bee products such as honey and wax rises, the need to monitor beehives also increases over time. This is evident in Aristotle's involvement in beehive monitoring in the 4th century BCE (Carlson, 2015).

Until the end of the 20th century, beehive monitoring was primarily conducted using manual methods. However, with advancements in technology, sensor-based systems have become a common area of research in beehive monitoring (Dorin, 2023).

Various sensors have been used in beehive monitoring to track different internal hive parameters, such as temperature, humidity, CO2 concentration, sound, vibrations, and the overall weight of the hive (Cecchi, 2020).

Most of the research on sensor-based beehive monitoring has been conducted in North America, Europe, China, Africa, Brazil, and Australia. There is a noticeable lack of research

in Sri Lanka and many other South Asian countries (Dorin, 2023). Even in India, there are very few papers on sensor-based beehive monitoring, which may be due to the relatively smaller size of the beekeeping industry and limited resource availability in these regions.

Several studies (Zacepins, 2017), (Seritan, 2018), (Bayir, 2016), (Cecchi, 2020), (Rector, 2008) have utilized load cells for hive weight measurement, which can be a direct indicator of honey production and colony size. However, most research has focused primarily on analyzing seasonal patterns in hive weight changes. One study analyzed daily weight variations in beehives and found a significant difference in daily weight changes during the winter and summer months (Cecchi, 2020).

With the use of sensor-based technology to monitor beehives, beehive health monitoring has become a major area of interest among various researchers (Cazier}, 2020), (Dorin, 2023). In a healthy hive, bees tend to regulate the internal temperature and humidity for brood production and development (Kridi, 2016). Any anomaly in the internal temperature and humidity patterns may indicate a possible issue within the colony (Hegde, 2023).One study (Weiss, 2017) implemented a system to monitor the beehive colony health using hive inside temperature. Furthermore, one study (Edwards-Murphy, 2016) utilized decision trees for beehive health monitoring using hive-inside relative humidity, temperature, carbon dioxide and oxygen levels. Hive
weight measurement is also used to gain insights into hive health (Rector, 2008), (Dorin, 2023). Some research tracks the activity at the hive entrance as a method to monitor hive health (Andrijević, 2022).
The queen bee is essential for the colony's survival. The queen must be healthy and fertile to ensure reproduction and the overall survival of the colony. Sometimes the queen bee may be infertile, unhealthy, dead, or attacked by intruders or pests. Beekeepers manually check hive frames to detect eggs or larvae to confirm the queen's presence. If eggs and larvae exist, it indicates that the queen bee is present (Cejrowski, 2018).

This process is time-consuming and stressful to the bees. To address this, various studies (Cejrowski, 2018), (Orlowska, 2022), (Uthoff, 2023), (Abdollahi, 2022), (Doinea, et al., 2024) have explored detecting the queen's presence or absence using sensor technology and AI. Most research focuses on acoustic signals emitted by bees, which change when the queen

bee is absent. (Uthoff, 2023). The major issue when creating a classifier for detecting queen presence/absence is the lack of generalization when tested on other hives (Cejrowski, 2018). As a result, a more generalized model for queen detection needs to be developed. This study will focus on creating a robust and generalized model to detect the presence of the queen bee using online audio datasets and testing it on a beehive.

Several studies (Dorin, 2023), (Uthoff, 2023) also use classification models based on acoustic signals to predict pre-swarming events. This research study will not focus on predicting pre-swarming but instead will develop a method to detect swarming that has already occurred using weight and temperature variations as proposed in (Cecchi, 2020).

Previous research lacks the utilization of mobile applications for beehive monitoring, except for a few research studies with limited features (Branco, 2023), (Pikri, 2022). In this study, a mobile application will be developed to monitor live data in the hive, environmental conditions outside the hive, hive health status, queen detection, and swarming detection. It will also provide features to track the queen bee's history and the hive's swarming history.

Chapter 04 | Requirement Analysis

4.1 IoT Module

For the IoT module, various sensors will be used to monitor the internal hive parameters. This includes the DHT22 sensor for humidity and temperature, and the MQ135 sensor for air quality monitoring. A load cell with the HX711 amplifier will be used for hive weight measurement. For monitoring external weather conditions, sensors such as the rainfall sensor and the BH1750 sensor for light intensity will be utilized. The ESP32 microcontroller will be employed due to its built-in Wi-Fi capabilities and powerful processing.

4.2 Realtime Database

To store real-time data sent from the ESP32, the Firebase Realtime Database will be used because of its efficient data synchronization and seamless integration with IoT applications.

4.3 Queen Detection Model

Datasets for queen detection will be obtained from publicly available datasets. For signal pre-processing and feature extraction, Python libraries such as Librosa will be utilized. For the classification part, machine learning frameworks like TensorFlow and Keras will be used to build and train the models.

4.4 Mobile Application

The mobile application will be developed using the Flutter framework due to its cross-platform capabilities and ease of use. To analyze hive health and track anomalies in temperature and humidity patterns, Flutter libraries such as syncfusion_flutter_charts, mlkit, quiver, statistics, tensorflow_lite_flutter, and rx_dart will be used to enhance data visualization and user experience.

4.5 Database for Mobile Application

Firestore Database will be employed to store data related to the user, hive history, including bee queen history and swarming history, due to its flexible data structure and scalability.

Chapter 05 | Requirement Analysis

Table 1 Expenses for the sensors and microcontroller

Category	Type	Planning to (Purchase, Borrow)	Quantity	Price (Rs)
Sensors + Microcontroller	DHT22 Temperature & Humidity sensor	Purchase	2	860
	15 kg load cell	Borrow	1	-
	Hx711 Load Cell Amplifier	Purchase	1	290
	Rainfall Sensor	Purchase	1	250
	MQ-135 Air Quality Sensor	Purchase	1	550
	BH1750FVI Digital Light Intensity Sensor	Purchase	1	450

	ESP32-WROOM-32 Microcontroller	Borrow	1	-
			Total	**2400**

Table 2 Expenses for power setup, wiring and connections of the IOT device

Category	Type	Planning to (Purchase, Borrow)	Quantity	Price (Rs)
Power + Wiring + Connections	8W Sollar Panel	Purchase	1	4,000
	Lithium Ion Battery	Purchase	2	1,400
	2-Battery BMS	Purchase	1	110
	Printed Circuit Board	Borrow	1	-
	2 pin terminal block connectors	Purchase	1	30
	3 pin terminal block connectors	Purchase	5	200
	4 pin terminal block connectors	Purchase	2	70
	7805 1A regulator	Purchase	1	50
	Waterproof box enclosure for sensors	Purchase	1	1500
			Total	**7360**

Table 3 Expenses for the hive setup and beekeeping training

Hive Setup + Beekeeping Training	Wooden hive + Bee Colony	Purchased	1	10,000
	Load Cell Wooden Frame	Purchased	1	2,000
	Additional Wooden hive (for testing purposes)	Purchased	1	2,000
	Bee Keeping Training	-	-	5,000

	Continuous Management of the beehive	-	-	8,000
			Total	**29,400**

Full cost (for devices): 2400 + 7360 = ***9760/=***

Additional costs: ***29400/=***

Chapter 06 | Time Plan

Table 4 Time Plan

Task ID	Task	Dependencies	Duration (weeks)	Start Date	End Date
1	Project Initiation, and Idea Finalization	-	2	Sep 1, 2024	Sep 14, 2024
2	Project Planning and Researching	1	3	Sep 14, 2024	Oct 5, 2024
3	Requirement Gathering and Analysis	2	2	Oct 5, 2024	Oct 20, 2024
4	Diagram Designing	3	1	Oct 20, 2024	Oct 26, 2024
5	Implementing the Hardware setup	3	4	Oct 20, 2024	Nov 16, 2024
6	Collecting the dataset	5	4	Nov 16, 2024	Dec 14, 2024
7	Creating and training the Bee Queen Presence classification model	5	2	Nov 16, 2024	Nov 30, 2024
8	UI designing	7	2	Nov 30, 2024	Dec 14, 2024
9	Analyzing the collected dataset	8	3	Dec 14, 2024	Jan 4, 2025
10	Developing the mobile application	9	5	Jan 4, 2025	Feb 8, 2025

11	Testing	10	2	Feb 8, 2025	Feb 22, 2025
12	Documentation	11	2	Feb 22, 2025	March 8, 2025

Start Date: ***September 01, 2024***

End Date: ***March 08, 2025***

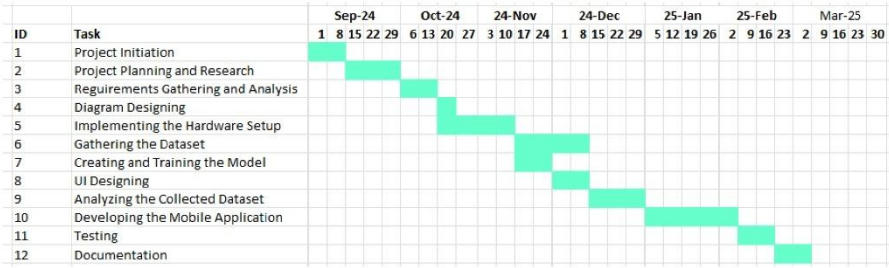


Figure 1 Gantt Chart

References

1. Abdollahi, M. a. G. P. a. F. T. H., 2022. Automated Beehive Acoustics Monitoring: A Comprehensive Review of the Literature and Recommendations for Future Work. *Applied Sciences,* 12(8).
2. Andrijević, N. a. U. V. a. A. B. a. H. D. a. S. B., 2022. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics,* Volume 11, p. 783.
3. Bayir, R. a. A. A., 2016. The monitoring of nectar flow period of honey bees using wireless sensor networks. *International Journal of Distributed Sensor Networks,* Volume 12.
4. Branco, D. C. a. J. M. a. H. M. a. F., 2023. BHiveSense: An integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices. *Journal of Open Innovation: Technology, Market, and Complexity,* Volume 9, p. 100110.
5. Carlson, R. D., 2015. *The Honey Bee and Apian Imagery in Classical Literature,* Seattle, WA: University of Washington.
6. Cazier}, A. {. B. a. D. {. G. a. R. R. a. E. {. H. a. B. {. F. a. J. {., 2020. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture,* Volume 169, p. 105161.
7. Cecchi, S. a. S. S. a. T. A. a. O. S., 2020. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors,* 20(9), p. 2726.
8. Cejrowski, T. a. S. J. a. M. H. a. G. D., 2018. Detection of the Bee Queen Presence Using Sound Analysis. In: s.l.:Springer, pp. 297-306.
9. Doinea, M. et al., 2024. IoT Embedded Smart Monitoring System with Edge Machine Learning for Beehive Management.. *International Journal of Computers, Communications & Control,* 19(4).
10. Dorin, A. Z. a. A., 2023. A framework for better sensor-based beehive health monitoring. *Computers and Electronics in Agriculture,* 210(0168-1699), p. 107906.
11. Edwards-Murphy, F. a. M. M. a. W. P. a. O. J. a. P. E., 2016. B+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture,* Volume 124, pp. 211-219.
12. Hegde, A. D. a. A. P. a. S., 2023. *HiveLink, an IoT based Smart Bee Hive Monitoring System.* [Online]
    Available at: https://arxiv.org/abs/2309.12054
    [Accessed 19 10 2024].
13. Kridi, D. S. a. d. C. C. G. N. a. G. D. G., 2016. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Elsevier Science Publishers B. V.,* 127(C), p. 221–235.
14. Orlowska, A. a. F. D. a. G. J.-P. a. C.-R. D., 2022. Honey Bee Queen Presence Detection from Audio Field Recordings Using Summarized Spectrogram and Convolutional Neural Networks. In: s.l.:s.n., pp. 83-92.
15. Pikri, M. a. I. S. a. F. N. a. M. M. a. M. S., 2022. IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping Using LoRa Technology. In: s.l.:s.n., pp. 334-342.
16. Rector, B. a. M. G. a. H. N., 2008. Within-day variation in continuous hive weight data as a measure of honey bee colony activity. *Apidologie,* Volume 39, pp. 694-707.
17. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. Low cost platform for monitoring honey production and bees health. *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR),* pp. 1-4.
18. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. *Low cost platform for monitoring honey production and bees health.* s.l., 2018 IEEE International Conference on

Automation, Quality and Testing, Robotics (AQTR).
19. Uthoff, C. a. N. H. M. a. V. B. M., 2023. Acoustic and vibration monitoring of honeybee colonies for beekeeping-relevant aspects of presence of queen bee and swarming. *Computers and Electronics in Agriculture,* Volume 205.
20. Weiss, M. a. M. P. a. F. W. a. S. L. a. S. T. a. M. B. a. A. K., 2017. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie,* Volume 48.
21. Whitfield, B. G., 1956. Virgil and the Bees: A Study in Ancient Apicultural Lore. *Greece and Rome,* 3(2), p. 99–117.
22. Zacepins, A. a. P. A. a. O. V. a. K. A. a. E. S., 2017. Solution for automated bee colony weight monitoring. *Agronomy Research,* Volume 15, pp. 585-593.

PUSL3190 Computing Individual Project

PID Document

A Sensor-Based System for Advanced Beehive Monitoring that utilizes Signal Processing and Machine Learning

Supervisor: Ms. Hirushi Dilpriya

Name: Nisal Liyanage
Plymouth Index Number: 10898808
Degree Program: BSC (HONS) Computer Science

Chapter 01 | Introduction

Bees play a crucial role in sustaining life on Earth and maintaining the balance of our ecosystem (Khan, 2023). Albert Einstein emphasized this significance with his famous quote: "If the bee disappeared off the surface of the globe, man would only have four years of life left." Bees are the largest pollinators on the planet, contributing to around 33% of the global food supply (Shaden A. M. Khalifa, 2021). They help fertilize and cross-pollinate many plant species, which is essential for the reproduction of various plants, including wildflowers, fruits, vegetables, and nuts like almonds (Christina T. Liang, 2016). Without bees, food security for both humans and animals would be at risk (Vidushi Patel, 2021).

Besides playing an important ecological role, bees also produce some important resources humans exploit in apiculture. Perhaps the most popular of all bee products is honey; though being a delicious and healthy food ingredient, it is also successfully applied in many medicinal and therapeutic fields (E.R.H.S.S. Ediriweera, 2012). Apart from honey, various other products such as beeswax, royal jelly, bee pollen, and propolis are cultivated in beekeeping for cosmetic and medical use (Isayas Asefa Kebede, 2024).

The beekeeping industry in Sri Lanka is conducted on a very small scale, often limited to a handful of beekeepers and bee hobbyists. Nevertheless, beekeeping represents a thriving sector within Sri Lanka's agricultural landscape, offering significant potential for both economic growth and environmental stewardship. This industry, with its plenty of benefits, can contribute substantially to the nation's development.

However, in recent years, bee populations around the world as well as in Sri Lanka have faced significant declines in health and numbers. This is evident through reduced honey yields, smaller colony sizes, higher mortality rates among worker bees, infertility in queen bees, and the occurrence of colony collapse disorder. Many factors contribute to this decline, including air pollution, widespread malnutrition, harmful pesticides, deforestation, and other human activities. In the beekeeping industry, poor monitoring practices can further harm bee populations (Peter Hristov, 2020).

Therefore, continuous monitoring of beehives is essential to ensure the health of the colony. This includes checking for pests or diseases, confirming the queen bee's presence, and ensuring there is enough honey for extraction. Unfortunately, in Sri Lanka currently beekeepers monitor their hives manually, which causes several challenges for the beekeepers and bees. Regular manual monitoring of hives is often time-consuming, inefficient, and risky, as beekeepers may get stung while inspecting hives (Yiyao Zheng, 2024). Additionally, many beekeepers overlook regular hive monitoring due to their busy schedules (Braga, 2021). Traditional manual monitoring can also make it difficult to gather reliable information about the health of the bee colony and often ignore important environmental factors like temperature and humidity (Pikri, 2022). As a result, the quality and quantity of honey can suffer due to insufficient monitoring of hive conditions, and these manual methods can be intrusive, causing stress to the bees (Braga, 2021), (Yiyao Zheng, 2024).

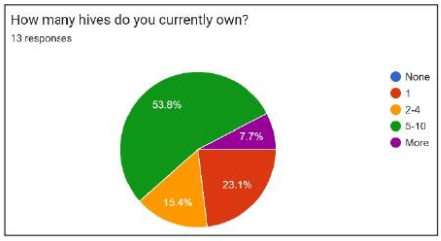
Chapter 02 | Business Case

2.1 Business Need

Therefore, there is a pressing need for non-intrusive methods to monitor beehives remotely and accurately, benefiting both the beekeepers and the bees. In order to provide a non-intrusive solution for monitoring beehives, a computerized hive monitoring system is being developed, incorporating IoT technology, signal processing, and AI/ML-based approaches. The end goal is to empower Sri Lankan beekeepers with a user-friendly mobile application enabling informed decision-making for optimal hive management and the well-being of bee colonies.

Table 1 The challenges/issues faced in traditional beehive monitoring in Sri Lanka and the solutions/ improvements that the project brings.

Challenge/ Issue	Solution/ Improvement
Regular manual inspections cause stress to the bees.	The developing solution causes less stress to the bees as it reduces regular human inspections.

Regular manual monitoring of each and every hive is time-consuming, as most beekeepers manage around 5-10 beehives (as for the results from the questionnaire shown in figure 1)  *Figure 1 Results from the Questionnaire, that is collected from 15 beekeepers*	Through a single mobile application, the beekeeper is able to monitor and keep track of all the hives he manages, without physically visiting them.
Regular manual monitoring can cause risk to the beekeepers of possible bee stings.	The developing solution allows the beekeepers to monitor their hives remotely, thereby reducing the risk of possible bee stings.
Early signs of hive issues or hive requirements are often overlooked in manual inspections.	The system allows beekeepers to obtain real-time, up to date data on hive conditions, allowing early detection of hive issues or hive requirements.
Manual inspections only allow limited data collection, resulting in insufficient insights into hive conditions.	The system collects data on various hive conditions, such as temperature and humidity (which cannot be effectively monitored manually), providing sufficient insights into hive health.
Unable to track the historic trends of the hive.	The system allows data storage, so the historical trends of the hive can be tracked which would result in informed decision making.

2.2 Business Objectives.

The proposed solution primarily targets:

1. Sri Lankan Beekeepers.
2. Bee Researchers and Scientists.

The following are the business objectives the system provides for each user:

1. **Beekeepers:**
   1.1. Minimize manual hive inspections and reduce the number of hive visits by **70%.**
   1.2. Save time for the beekeepers.
   1.3. Detect hive issues and identify hive requirements at an early stage.
   1.4. Reduce the risk of possible bee stings.
   1.5. Enhance honey quality and quantity by maintaining a less stressed and healthy bee colony.
   1.6. Early detection of hive issues can reduce hive operational and management costs by **50%**. *(e.g.: early identification of a disease allows the beekeeper to take timely action before it spreads, significantly reducing costs associated with treatment, resources, and hiring professional beekeeping services.)*

2. **Bee Researchers:**
   2.1. Monitor the behaviors of the Asian honeybees in Sri Lanka.
   2.2. Analyze the optimal hive conditions for a stronger colony with increased honey yield.

Chapter 03 | Project Objectives.

With respect to the proposed sensor-based system for advanced beehive monitoring:

- **Main Objective**

    1. Provide beekeepers and the bee researchers with a mobile application to remotely monitor their beehives.

- **Specific Objectives**

    1. Track the normal honeybee activities and hive conditions. (e.g., temperature variations during the day).
    2. Visualize real-time data from the hive.
    3. Offer insights into historical trends of the hive (e.g.: How the weight of the hive has changed overtime).
    4. Deliver insights on hive health by detecting anomalies in normal bee activities and hive conditions.
    5. Provide insights into the optimal time for harvesting.
    6. Detect the presence of the queen bee automatically using acoustic signals.
    7. Record any swarming events that occur.
    8. Maintain a record of the queen bee's history within the hive.
    9. Maintain a record of the swarming history of the hive.

Chapter 04 | Literature Review

Human beekeeping has roots that go all the way back to ancient Egypt and ancient Greece (Whitfield, 1956). As the consumption of bee products such as honey and wax rises, the need to monitor beehives also increases over time. This is evident in Aristotle's involvement in beehive monitoring in the 4th century BCE (Carlson, 2015).

Until the end of the 20th century, beehive monitoring was primarily conducted using manual methods. However, with advancements in technology, sensor-based systems have become a common area of research in beehive monitoring (Dorin, 2023).

Various sensors have been used in beehive monitoring to track different internal hive parameters, such as temperature (mostly DHT22 sensors were used), humidity (mostly DHT22 sensors were used), CO2 concentration (Gas sensors and MG811 sensor were used) , sound (mostly electret microphones were used) , vibrations (various vibration sensors were used), and the overall weight of the hive (load cells were used) (Cecchi, 2020), (Dorin, 2023).

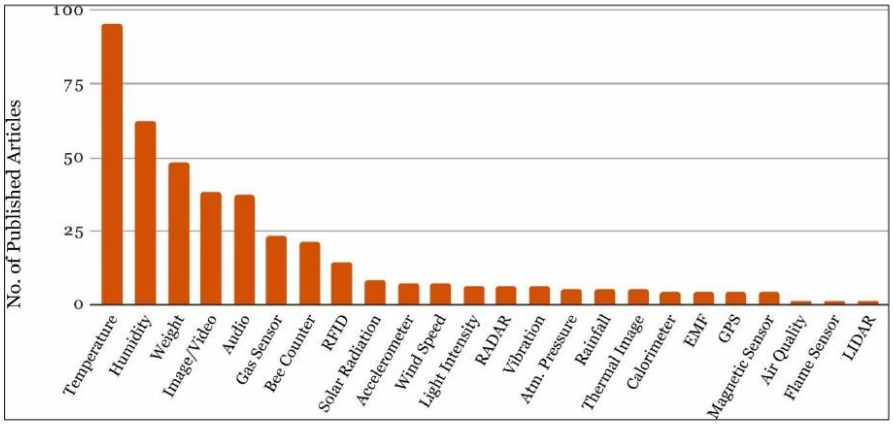


Figure 2 Graph showing the number of publication articles that utilize each type of sensor in beehive monitoring (Last Update: 2023). Reprinted from (Dorin, 2023)

Most of the research on sensor-based beehive monitoring has been conducted in North America, Europe, China, Africa, Brazil, and Australia. There is a noticeable lack of research in Sri Lanka and many other South Asian countries (Dorin, 2023). Even in India, there are

very few papers on sensor-based beehive monitoring, which may be due to the relatively smaller size of the beekeeping industry and limited resource availability in these regions.

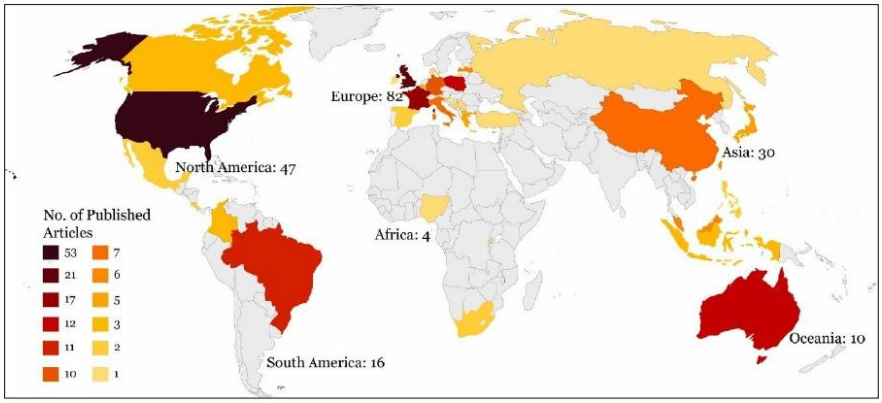


Figure 3 World Map showing the distribution of the previously done research publications on smart beehives. (Last update: 2023). Reprinted from (Dorin, 2023)

According to (Dorin, 2023), most of the previous research done on smart beehives largely focused on European honeybees, followed by stingless bees and bumble bees. There is a significant lack of focus given to the Asian honeybees in the past research. But in this study the focus was given to the Asian honeybees, as this is the most commonly found honeybee species in Sri Lanka.

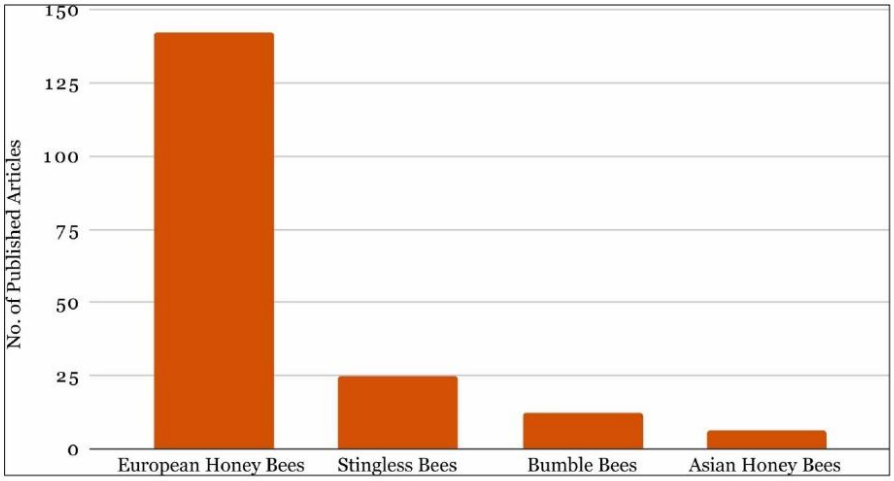


Figure 4: Graph showing the number of research publications that focus on each bee species. (Last Update: 2023). Reprinted from (Dorin, 2023)

Several studies (Zacepins, 2017), (Seritan, 2018), (Bayir, 2016), (Cecchi, 2020), (Rector, 2008) have utilized load cells for hive weight measurement, which can be a direct indicator of honey production and colony size. However, most research has focused primarily on analyzing seasonal patterns in hive weight changes. One study analyzed daily weight variations in beehives and found a significant difference in daily weight changes during the winter and summer months (Cecchi, 2020).

With the use of sensor-based technology to monitor beehives, beehive health monitoring has become a major area of interest among various researchers (Cazier}, 2020), (Dorin, 2023). In a healthy hive, bees tend to regulate the internal temperature and humidity for brood production and development (Kridi, 2016). Any anomaly in the internal temperature and humidity patterns may indicate a possible issue within the colony (Hegde, 2023).One study (Weiss, 2017) implemented a system to monitor the beehive colony health using hive inside temperature. Furthermore, one study (Edwards-Murphy, 2016) utilized decision trees for beehive health monitoring using hive-inside relative humidity, temperature, carbon dioxide and oxygen levels. Hive weight measurement is also used to gain insights into hive health (Rector, 2008), (Dorin, 2023). Some research tracks the activity at the hive entrance as a method to monitor hive health (Andrijević, 2022).

The queen bee is essential for the colony's survival. The queen must be healthy and fertile to ensure reproduction and the overall survival of the colony. Sometimes the queen bee may be infertile, unhealthy, dead, or attacked by intruders or pests. Beekeepers manually check hive frames to detect eggs or larvae to confirm the queen's presence. If eggs and larvae exist, it indicates that the queen bee is present (Cejrowski, 2018).

This process is time-consuming and stressful to the bees. To address this, various studies (Cejrowski, 2018), (Orlowska, 2022), (Uthoff, 2023), (Abdollahi, 2022), (Doinea, et al., 2024) have explored detecting the queen's presence or absence using sensor technology and AI. Most research focuses on acoustic signals emitted by bees, which change when the queen bee is absent. (Uthoff, 2023). The major issue when creating a classifier for detecting queen presence/absence is the lack of generalization when tested on other hives (Cejrowski, 2018). As a result, a more generalized model for queen detection needs to be developed. This study will focus on creating a robust and generalized model to detect the presence of the queen bee using online audio datasets and testing it on a separate local beehive.
Several studies (Dorin, 2023), (Uthoff, 2023) also use classification models based on acoustic signals to predict pre-swarming events. This research study will not focus on predicting pre-swarming but instead will develop a method to detect swarming that has already occurred using weight and temperature variations as proposed in (Cecchi, 2020).

Previous research lacks the utilization of mobile applications for beehive monitoring, except for a few research studies with limited features (Branco, 2023), (Pikri, 2022). In this study, a mobile application will be developed to monitor live data in the hive, environmental conditions outside the hive, hive health status, queen detection, and swarming detection. It will also provide features to track the queen bee's history and the hive's swarming history.

Chapter 05 | Method of Approach

5.1 Overview

The project follows the waterfall methodology, which ensures a structured and sequential approach to development. The system comprises an IoT module incorporating sensors and a microcontroller, a real-time database, an AI model to predict the presence of the queen, and a Flutter mobile application integrated with a backend database. A brief description of each of these components is provided below.

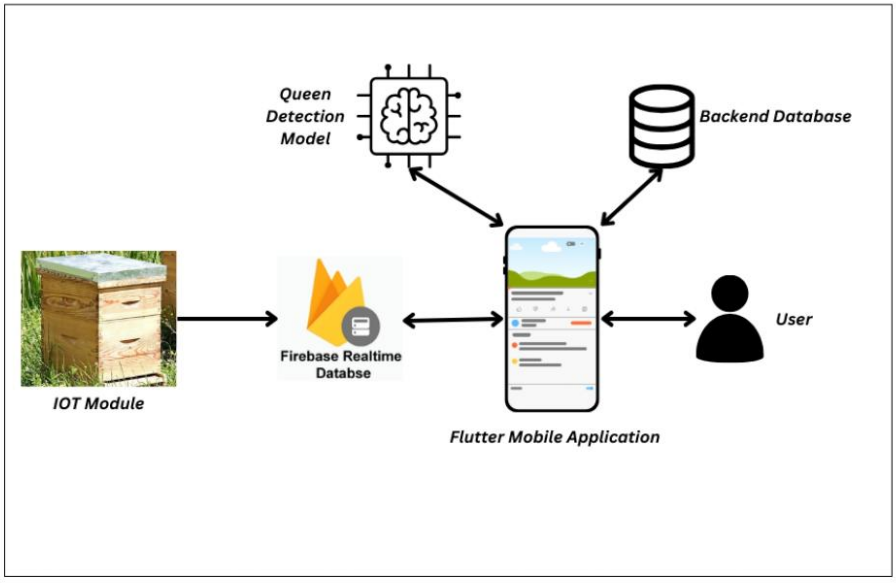


Figure 5 Conceptual Diagram of the Implementing System

5.2 IoT Module

For the IoT module, various sensors are used to monitor the internal hive parameters. This includes the DHT22 sensor for humidity and temperature, and the MQ135 sensor for air quality monitoring. A load cell with the HX711 amplifier will be used for hive weight measurement. For monitoring external weather conditions, sensors such as the rainfall sensor and the BH1750 sensor for light intensity will be utilized. The ESP32 microcontroller will be employed due to its built-in Wi-Fi capabilities and powerful processing.

5.3 Realtime Database

To store real-time data sent from the ESP32, the Firebase Realtime Database will be used because of its efficient data synchronization and seamless integration with IoT applications.

5.4 Queen Detection Model

Datasets for queen detection will be obtained from publicly available datasets. For signal pre-processing and feature extraction, Python libraries such as Librosa will be utilized. For the classification part, machine learning frameworks like TensorFlow and Keras will be used to build and train the models.

5.5 Mobile Application

The mobile application will be developed using the Flutter framework due to its cross-platform capabilities and ease of use. To analyze hive health and track anomalies in temperature and humidity patterns, Flutter libraries such as syncfusion_flutter_charts, mlkit, quiver, statistics, tensorflow_lite_flutter, and rx_dart will be used to enhance data visualization and user experience.

5.6 Database for Mobile Application

Firestore Database will be employed to store data related to the user, hive history, including bee queen history and swarming history, due to its flexible data structure and scalability.

Chapter 06 | Initial Project Plan

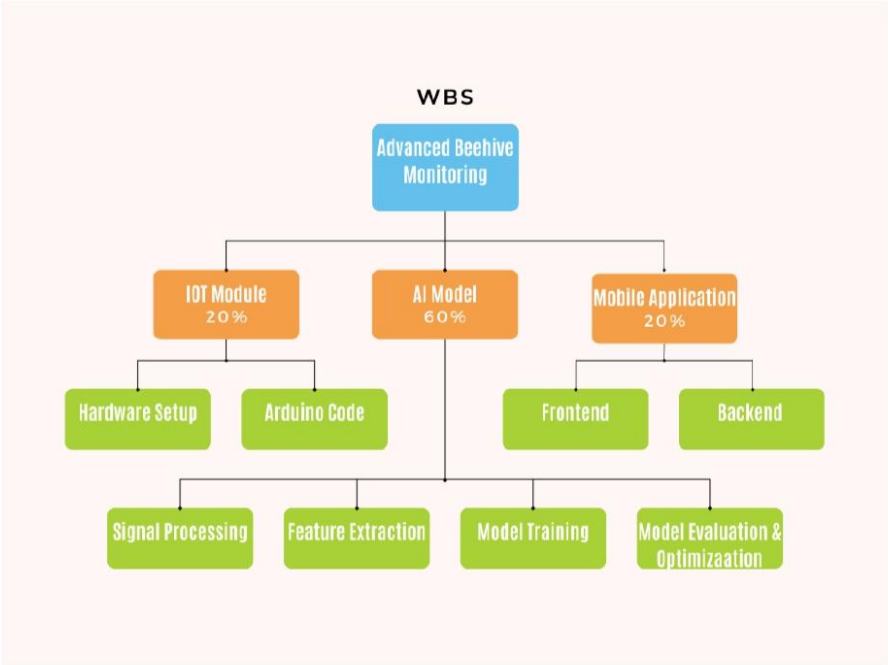


Figure 6 Work Breakdown Structure

Table 2 Time Plan

Task ID	Task	Dependencies	Duration (weeks)	Start Date	End Date
1	Project Initiation, and Idea Finalization	-	2	Sep 1, 2024	Sep 14, 2024
2	Project Planning and Researching	1	3	Sep 14, 2024	Oct 5, 2024
3	Requirement Gathering and Analysis	2	2	Oct 5, 2024	Oct 20, 2024
4	Diagram Designing	3	1	Oct 20, 2024	Oct 26, 2024
5	Implementing the Hardware setup	3	4	Oct 20, 2024	Nov 16, 2024

6	Collecting the dataset	5	4	Nov 16, 2024	Dec 14, 2024
7	Creating and training the Bee Queen Presence classification model	5	2	Nov 16, 2024	Nov 30, 2024
8	UI designing	7	2	Nov 30, 2024	Dec 14, 2024
9	Analyzing the collected dataset	8	3	Dec 14, 2024	Jan 4, 2025
10	Developing the mobile application	9	5	Jan 4, 2025	Feb 8, 2025
11	Testing	10	2	Feb 8, 2025	Feb 22, 2025
12	Documentation	11	2	Feb 22, 2025	March 8, 2025

Start Date: ***September 01, 2024***

End Date: ***March 08, 2025***

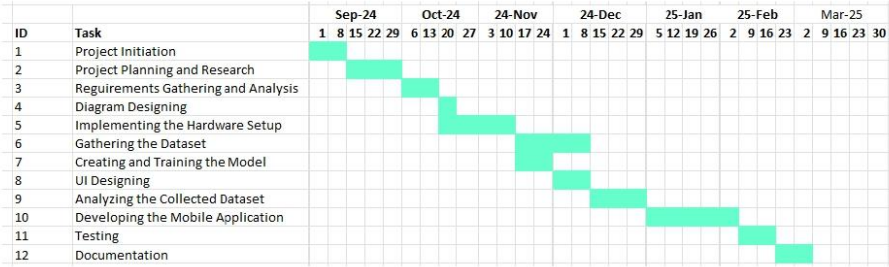


Figure 7 Gantt Chart

Chapter 07 | Risk Analysis

Table 3 Risk Analysis Matrix

Component	*Risk*	*Likelihood*	*Impact*	*Mitigation Strategy*
IOT Device	Malfunctions in sensors	**HIGH**	**HIGH**	Buy additional sensors, Purchase from reliable sources
	Incorrect readings due to calibration issues with the load cell	**MEDIUM**	**HIGH**	Perform regular calibrations, Ensure the stability of the frame, Apply waterproof paint to avoid environmental impacts from temperature and humidity.
	Power Supply Issues	**HIGH**	**HIGH**	Use solar power to charge the batteries within the day, Use power banks to address power constraints during the night.
	Connectivity Issues between the IOT device and the firebase real-time database	**HIGH**	**HIGH**	Reduce unnecessary data in the JSON payload, Adjust the timeout configurations, Use a faster and a reliable network.
	Data Losses	**MEDIUM**	**LOW**	Use linear interpolation to estimate the missing data.
AI Model	Uncertainty regarding the dataset's reliability and the experiment procedure	**LOW**	**MODERATE**	Read previous research done using the same dataset and try to reproduce their results.
	Model's lack of generalization	**HIGH**	**HIGH**	Use cross-validation as well as validate with new real-world sound samples.
	Lower accuracy when using cross validation	**HIGH**	**HIGH**	Use data augmentation techniques and experiment with different types of extracted features and classification models
	Machine power and memory constraints for model training	**HIGH**	**MODERATE**	Use google collab notebooks for model training
Mobile App	Integration and usability issues with the AI model	**HIGH**	**HIGH**	Use model-compression techniques to reduce the model size, Experiment with different flutter

				libraries to record the voice in .wav format.
	Bugs and crashes during use	**MODERATE**	**LOW**	Perform extensive testing

References

1. Abdollahi, M. a. G. P. a. F. T. H., 2022. Automated Beehive Acoustics Monitoring: A Comprehensive Review of the Literature and Recommendations for Future Work. *Applied Sciences,* 12(8).
2. Andrijević, N. a. U. V. a. A. B. a. H. D. a. S. B., 2022. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics,* Volume 11, p. 783.
3. Anon., n.d.
4. Bayir, R. a. A. A., 2016. The monitoring of nectar flow period of honey bees using wireless sensor networks. *International Journal of Distributed Sensor Networks,* Volume 12.
5. Braga, D. a. M. A. a. S. F. a. P. V. a. A. A., 2021. An Intelligent Monitoring System for Assessing Bee Hive Health. *IEEE Access,* Volume 9, pp. 89009-89019.
6. Branco, D. C. a. J. M. a. H. M. a. F., 2023. BHiveSense: An integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices. *Journal of Open Innovation: Technology, Market, and Complexity,* Volume 9, p. 100110.
7. Carlson, R. D., 2015. *The Honey Bee and Apian Imagery in Classical Literature,* Seattle, WA: University of Washington.
8. Cazier}, A. {. B. a. D. {. G. a. R. R. a. E. {. H. a. B. {. F. a. J. {., 2020. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture,* Volume 169, p. 105161.
9. Cecchi, S. a. S. S. a. T. A. a. O. S., 2020. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors,* 20(9), p. 2726.
10. Cejrowski, T. a. S. J. a. M. H. a. G. D., 2018. Detection of the Bee Queen Presence Using Sound Analysis. In: s.l.:Springer, pp. 297-306.
11. Christina T. Liang, B. G. K. H. W. T., 2016. The Role of Honey Bees as Pollinators in Natural Areas. *Natural Areas Journal,* 36(4), pp. 478-488.
12. Diogo Braga, Ana Mudureura, Fabio Scotti, Vincenzo Piri, Ajith Abraham, 2021. *An Intelligent Monitoring Systemfor Assessing Bee Hive Health.* Portugal, s.n.
13. Doinea, M. et al., 2024. IoT Embedded Smart Monitoring System with Edge Machine Learning for Beehive Management.. *International Journal of Computers, Communications & Control,* 19(4).
14. Dorin, A. Z. a. A., 2023. A framework for better sensor-based beehive health monitoring. *Computers and Electronics in Agriculture,* 210(0168-1699), p. 107906.
15. E. R. H. S. S. Ediriweera, N Y S Premarathna, 2012. *Medicinal and cosmetic uses of Bee's Honey – A review.* Sri Lanka, s.n.
16. E.R.H.S.S. Ediriweera, N. Y. S. P., 2012. Medicinal and cosmetic uses of Bee's Honey – A review. *AYU (An International Quarterly Journal of Research in Ayurveda),* 33(2), pp. 178-82.
17. Edwards-Murphy, F. a. M. M. a. W. P. a. O. J. a. P. E., 2016. B+WSN: Smart beehive with

preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture,* Volume 124, pp. 211-219.
18. Hegde, A. D. a. A. P. a. S., 2023. *HiveLink, an IoT based Smart Bee Hive Monitoring System*. [Online]
Available at: https://arxiv.org/abs/2309.12054 [Accessed 19 10 2024].
19. Isayas Asefa Kebede, H. F. A. D. A. G. D., 2024. Bee products and their processing: a review. *Pharmacy & Pharmacology International Journal,* 12(2), pp. 5-12.
20. Khan, S. A., 2023. Review of the Role of Bees as Ecosystem Engineers in Nature. *Science Reviews - Biology,* 2(4), pp. 1-6.
21. Kridi, D. S. a. d. C. C. G. N. a. G. D. G., 2016. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Elsevier Science Publishers B. V.,* 127(C), p. 221–235.
22. Mansor, M., 2022. *IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping using Lora Technology..* s.l, s.n.
23. Orlowska, A. a. F. D. a. G. J.-P. a. C.-R. D., 2022. Honey Bee Queen Presence Detection from Audio Field Recordings Using Summarized Spectrogram and Convolutional Neural Networks. In: s.l.:s.n., pp. 83-92.
24. Peter Hristov, R. S. N. P. B. N., 2020. Factors Associated with Honey Bee Colony Losses: A Mini-Review.. *Veterinary sciences,* 7(4), p. 166.
25. Pikri, M. a. I. S. a. F. N. a. M. M. a. M. S., 2022. IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping Using LoRa Technology. In: s.l.:s.n., pp. 334-342.
26. Rector, B. a. M. G. a. H. N., 2008. Within-day variation in continuous hive weight data as a measure of honey bee colony activity. *Apidologie,* Volume 39, pp. 694-707.
27. sayas Asefa Kebede,Haben Fesseha, Gebremeskel, Abrahim Dawed Ahmed, Gelan Dule, 2024. *Bee products and their processing: a review.* Ethiopia, s.n.
28. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. Low cost platform for monitoring honey production and bees health. *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR),* pp. 1-4.
29. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. *Low cost platform for monitoring honey production and bees health.* s.l., 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR).
30. Shaden A. M. Khalifa, E. H. E. A. A. S. A. A. A. E.-W. A. F. A. S. G. M. M. F. A. C. Z. S. H. D. M. M. A.-D. M. F. H. G. K., 2021. Overview of Bee Pollination and Its Economic Value for Crop Production. *Insects 2021,* 12(8), p. 688.
31. Uthoff, C. a. N. H. M. a. V. B. M., 2023. Acoustic and vibration monitoring of honeybee colonies for beekeeping-relevant aspects of presence of queen bee and swarming. *Computers and Electronics in Agriculture,* Volume 205.
32. Vidushi Patel, N. P. E. B. L. B. &. B. B., 2021. Why bees are critical for achieving sustainable development. *Ambio,* 50(1), pp. 49-51.
33. Weiss, M. a. M. P. a. F. W. a. S. L. a. S. T. a. M. B. a. A. K., 2017. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie,* Volume 48.
34. Whitfield, B. G., 1956. Virgil and the Bees: A Study in Ancient Apicultural Lore. *Greece and Rome,* 3(2), p. 99–117.
35. Yiyao Zheng , Xiaoyan Cao, Shaocong Xu, Shihui Guo, Rencai Huang, Yingjiao Li, Yijie Chen, Liulin Yang, Xiaoyu Cao,Zainura Idrus, Hongting Sun, 2024. *Intelligent beehive monitoring system based on internet of things and colony state analysis.* China, s.n.
36. Yiyao Zheng, X. C. S. X. S. G. R. H. Y. L. Y. C. L. Y. X. C. Z. I. H. S., 2024. Intelligent beehive monitoring system based on internet of things and colony state analysis. *Smart Agricultural*

Technology}, 9(1), p. 100584.
37. Zacepins, A. a. P. A. a. O. V. a. K. A. a. E. S., 2017. Solution for automated bee colony weight monitoring. *Agronomy Research,* Volume 15, pp. 585-593.

PUSL3190 Computing Individual Project

Interim Report

A Sensor-Based System for Advanced Beehive Monitoring that utilizes Signal Processing and Machine Learning

Supervisor: Ms. Hirushi Dilpriya

Name: Nisal Liyanage

Plymouth Index Number: 10898808

Degree Program: BSC (HONS) Computer Science

Chapter 01 | Introduction

1.1 Overview

Bees play a crucial role in sustaining life on Earth and maintaining the balance of our ecosystem (Khan, 2023). Albert Einstein emphasized this significance with his famous quote: "If the bee disappeared off the surface of the globe, man would only have four years of life left." Bees are the largest pollinators on the planet, contributing to around 33% of the global food supply (Shaden A. M. Khalifa, 2021). They help fertilize and cross-pollinate many plant species, which is essential for the reproduction of various plants, including wildflowers, fruits, vegetables, and nuts like almonds (Christina T. Liang, 2016). Without bees, food security for both humans and animals would be at risk (Vidushi Patel, 2021).

Besides playing an important ecological role, bees also produce some important resources humans exploit in apiculture. Perhaps the most popular of all bee products is honey; though being a delicious and healthy food ingredient, it is also successfully applied in many medicinal and therapeutic fields (E.R.H.S.S. Ediriweera, 2012). Apart from honey, various other products such as beeswax, royal jelly, bee pollen, and propolis are cultivated in beekeeping for cosmetic and medical use (Isayas Asefa Kebede, 2024).

The beekeeping industry in Sri Lanka is conducted on a very small scale, often limited to a handful of beekeepers and bee hobbyists. Nevertheless, beekeeping represents a thriving sector within Sri Lanka's agricultural landscape, offering significant potential for both economic growth and environmental stewardship. This industry, with its plenty of benefits, can contribute substantially to the nation's development.

1.2 Problem Definition

However, in recent years, bee populations around the world as well as in Sri Lanka have faced significant declines in health and numbers. This is evident through reduced honey yields, smaller colony sizes, higher mortality rates among worker bees, infertility in queen bees, and the occurrence of colony collapse disorder. Many factors contribute to this decline, including air pollution, widespread malnutrition, harmful pesticides, deforestation, and other human activities. In the beekeeping industry, poor monitoring practices can further harm bee populations (Peter Hristov, 2020).

Therefore, continuous monitoring of beehives is essential to ensure the health of the colony. This includes checking for pests or diseases, confirming the queen bee's presence, and ensuring there is enough honey for extraction. Unfortunately, in Sri Lanka currently beekeepers monitor their hives manually, which causes several challenges for the beekeepers and bees.

Regular manual monitoring of hives is often time-consuming, inefficient, and risky, as beekeepers may get stung while inspecting hives (Yiyao Zheng, 2024). Additionally, many beekeepers overlook regular hive monitoring due to their busy schedules (Braga, 2021). Traditional manual monitoring can also make it difficult to gather reliable information about the health of the bee colony and often ignore important environmental factors like temperature and humidity (Pikri, 2022). As a result, the quality and quantity of honey can suffer due to insufficient monitoring of hive conditions, and these manual methods can be intrusive, causing stress to the bees (Braga, 2021), (Yiyao Zheng, 2024).

1.3 Solution

Therefore, there is a pressing need for non-intrusive methods to monitor beehives remotely and accurately, benefiting both the beekeepers and the bees. In order to provide a non-intrusive solution for monitoring beehives, a computerized hive monitoring system is being developed, incorporating IoT technology, signal processing, and AI/ML-based approaches. The end goal is to empower Sri Lankan beekeepers with a user-friendly mobile application enabling informed decision-making for optimal hive management and the well-being of bee colonies.

1.4 Project Objectives

With respect to the developed sensor-based system for advanced beehive monitoring:

- **Main Objective**
    1. Provide beekeepers and the bee researchers with a mobile application to remotely monitor their beehives.
- **Specific Objectives**
    1. Track the normal honeybee activities and hive conditions. (e.g., temperature variations during the day).
    2. Visualize real-time data from the hive.

3. Offer insights into historical trends of the hive (e.g.: How the weight of the hive has changed overtime).
4. Deliver insights on hive health by detecting anomalies in normal bee activities and hive conditions.
5. Provide insights into the optimal time for harvesting.
6. Detect the presence of the queen bee automatically using acoustic signals.
7. Maintain a record of the queen bee's history within the hive.

Chapter 02 | System Analysis

2.1 Facts Gathering Techniques

2.1.1 Observations

As the primary requirements gathering technique, a filed visit to the Bandarawela Bidunuwewa Bee Unit was carried out.

Figure 1 Photos taken during the field visit to the Bandarawela Bidunuwewa Bee Unit

The visit to the bee unit helped to gain educational knowledge regarding the behavior of bees and standard beekeeping practices.

Apart from that, a local apiary belonging to a beekeeper in Horana was also visited to collect additional requirements.

Figure 2 Visit to an apiary belonging to a beekeeper in Horana, Sri Lanka

2.1.2 Interview

A Zoom interview was conducted with Mr. Vimukthi, who is the director at Bandarawela Bidunuwewa Bee Unit in order to verify and confirm the gathered requirements.

Figure 3 During the zoom online interview with the director at Bandarawela Bidunuwewa Bee Unit

The following were the questions that were asked and the answers that were received during the interview.

1	Sir, what is the most commonly farmed bee species in Sri Lanka?
	Mostly Asian Honey Bees. Some bee keepers use certain breeds of imported European Honey Bees.
2	Sir, what is the current method commonly used by Sri Lankan bee keepers to detect the queen's precense or absence?
	Beekeepers must carefully inspect each frame by hand to identify the queen or detect the presence of eggs.
3	After detecting queen's absence what do the bee keepers usually do?
	Remove any queen cells if any is visible. Add an artificially bred, undisabled, active and healthy queen,that has a longer body. Queen's lenght usually determines the queen's productivity.
4	Will it be beneficial if an automatic method is introduced for beekeepers to monitor the queen's absense or precense in the hive remotely?
	Yes. It will help inexperienced bee keepers, as they lack the essential experience to check for the queen manually. It will also help professional bee keepers to save their time and effort.

Figure 4 Question 1-4 asked during the zoom interview

5	As for the research I have conducted, Asian Honey Bees tend to maintain hive inside temperature around 34.5-35.5 celcius, given that the colony is healthy. For your knowledge do you agree with this? Or are there any distinctive temperature patterns noticed among asian honey bee colonies in Sri Lanka?
We haven't conducted much research on that. In certain occassions, we used thermometers to measure the hive temperature of the hives, specially during the wing virus outbreak.That gave us higher temperatures.	
6	Same question for hive inside humidity. According to the researh I codnucted, it is learnt that asian honey bees regulate the hive inside humidity around 75%, given that the colony is healthy. What are your opinions about this?
I haven't given much though to the hive humidity. But have heard from the books that it is also an important factor for the bee colony's health.	
7	How do Sri Lankan bee keepers check for the colony's health?
They have to visit the hives regularly and check each of the frame, in order to detect any abnormalities within the hive. Such as detecting any signs of pests or diseases or unsusual behaviour of the bees.	

Figure 5 Questions 5-7 asked during the zoom interview

8	What if an automatic method is developed, so that the bee keepers can monitor and analyze the bee colony's health remotely?
	It would be beenficial, and is the most needed for bee keepers.
9	If we develop a system for the bee keepers to monitor their hives remotely, what would be the best deliverable according to you? Mobile App? Web App? SMS?
	Mobile app would be the best. It will be simple and efficient, as most of the bee keepers have a mobile phone with them.

Figure 6 Questions 8 and 9 asked during the zoom interview

2.2 Existing Systems

Human beekeeping has roots that go all the way back to ancient Egypt and ancient Greece (Whitfield, 1956). As the consumption of bee products such as honey and wax rises, the need to monitor beehives also increases over time. This is evident in Aristotle's involvement in beehive monitoring in the 4th century BCE (Carlson, 2015).

Until the end of the 20th century, beehive monitoring was primarily conducted using manual methods. However, with advancements in technology, sensor-based systems have become a common area of research in beehive monitoring (Dorin, 2023).

Various sensors have been used in beehive monitoring to track different internal hive parameters, such as temperature (mostly DHT22 sensors were used), humidity (mostly DHT22 sensors were used), CO2 concentration (Gas sensors and MG811 sensor were used) , sound (mostly electret microphones were used) , vibrations (various vibration sensors were used), and the overall weight of the hive (load cells were used) (Cecchi, 2020), (Dorin, 2023).

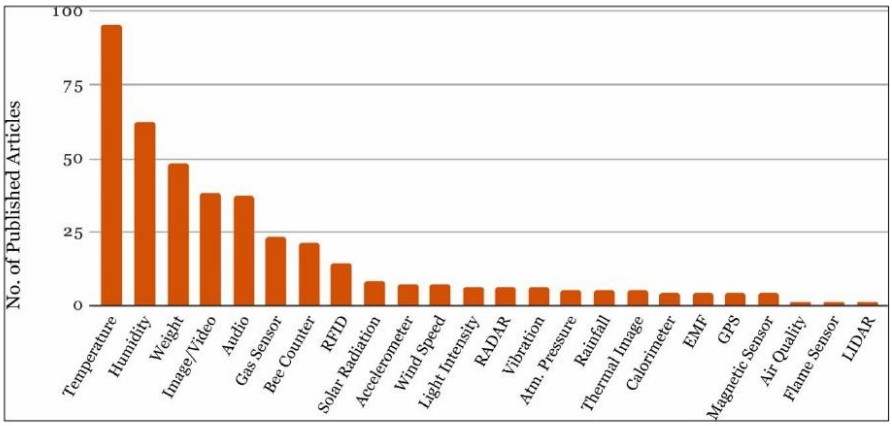


Figure 7 Graph showing the number of publication articles that utilize each type of sensor in beehive monitoring (Last Update: 2023). Reprinted from (Dorin, 2023)

Most of the research on sensor-based beehive monitoring has been conducted in North America, Europe, China, Africa, Brazil, and Australia. There is a noticeable lack of research in Sri Lanka and many other South Asian countries (Dorin, 2023). Even in India, there are very few papers on sensor-based beehive monitoring, which may be due to the relatively smaller size of the beekeeping industry and limited resource availability in these regions.

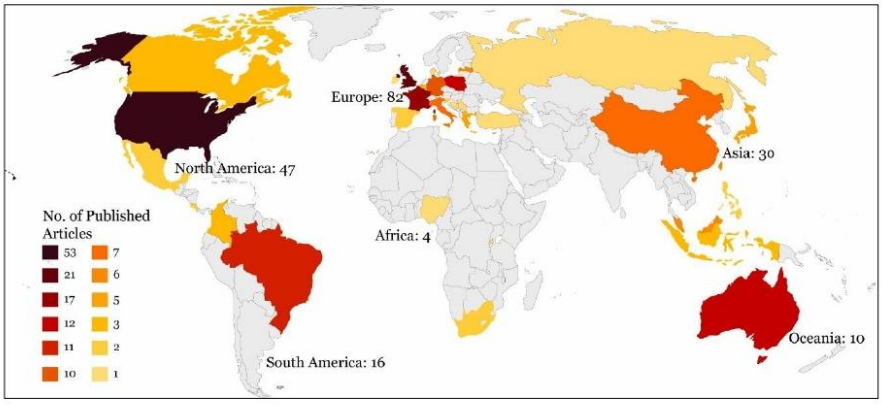


Figure 8 World Map showing the distribution of the previously done research publications on smart beehives. (Last update: 2023). Reprinted from (Dorin, 2023)

According to (Dorin, 2023), most of the previous research done on smart beehives largely focused on European honeybees, followed by stingless bees and bumble bees. There is a

significant lack of focus given to the Asian honeybees in the past research. But in this study the focus was given to the Asian honeybees, as this is the most commonly found honeybee species in Sri Lanka.

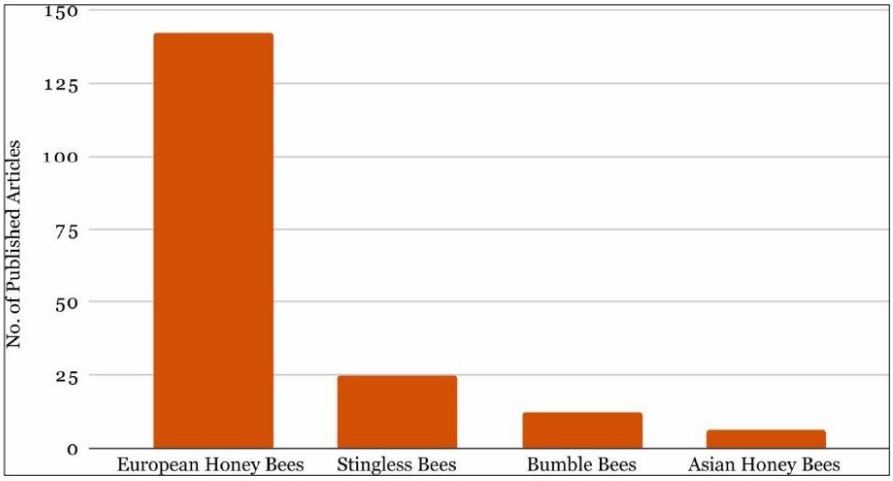


Figure 9: Graph showing the number of research publications that focus on each bee species. (Last Update: 2023). Reprinted from (Dorin, 2023)

Several studies (Zacepins, 2017), (Seritan, 2018), (Bayir, 2016), (Cecchi, 2020), (Rector, 2008) have utilized load cells for hive weight measurement, which can be a direct indicator of honey production and colony size. However, most research has focused primarily on analyzing seasonal patterns in hive weight changes. One study analyzed daily weight variations in beehives and found a significant difference in daily weight changes during the winter and summer months (Cecchi, 2020).

With the use of sensor-based technology to monitor beehives, beehive health monitoring has become a major area of interest among various researchers (Cazier}, 2020), (Dorin, 2023). In a healthy hive, bees tend to regulate the internal temperature and humidity for brood production and development (Kridi, 2016). Any anomaly in the internal temperature and humidity patterns may indicate a possible issue within the colony (Hegde, 2023).One study (Weiss, 2017) implemented a system to monitor the beehive colony health using hive inside temperature. Furthermore, one study (Edwards-Murphy, 2016) utilized decision trees for beehive health monitoring using hive-inside relative humidity, temperature, carbon dioxide and oxygen levels.

Hive weight measurement is also used to gain insights into hive health (Rector, 2008), (Dorin, 2023). Some research tracks the activity at the hive entrance as a method to monitor hive health (Andrijević, 2022).

The queen bee is essential for the colony's survival. The queen must be healthy and fertile to ensure reproduction and the overall survival of the colony. Sometimes the queen bee may be infertile, unhealthy, dead, or attacked by intruders or pests. Beekeepers manually check hive frames to detect eggs or larvae to confirm the queen's presence. If eggs and larvae exist, it indicates that the queen bee is present (Cejrowski, 2018).

This process is time-consuming and stressful to the bees. To address this, various studies (Cejrowski, 2018), (Orlowska, 2022), (Uthoff, 2023), (Abdollahi, 2022), (Doinea, et al., 2024) have explored detecting the queen's presence or absence using sensor technology and AI. Most research focuses on acoustic signals emitted by bees, which change when the queen bee is absent. (Uthoff, 2023). The major issue when creating a classifier for detecting queen presence/absence is the lack of generalization when tested on other hives (Cejrowski, 2018). As a result, a more generalized model for queen detection needs to be developed. This study will focus on creating a robust and generalized model to detect the presence of the queen bee using online audio datasets and testing it on a separate local beehive.

Previous research lacks the utilization of mobile applications for beehive monitoring, except for a few research studies with limited features (Branco, 2023), (Pikri, 2022). In this study, a mobile application will be developed to monitor live data in the hive, environmental conditions outside the hive, hive health status, and queen detection. It will also provide features to track the queen bee's history.

2.3 Use Case Diagram

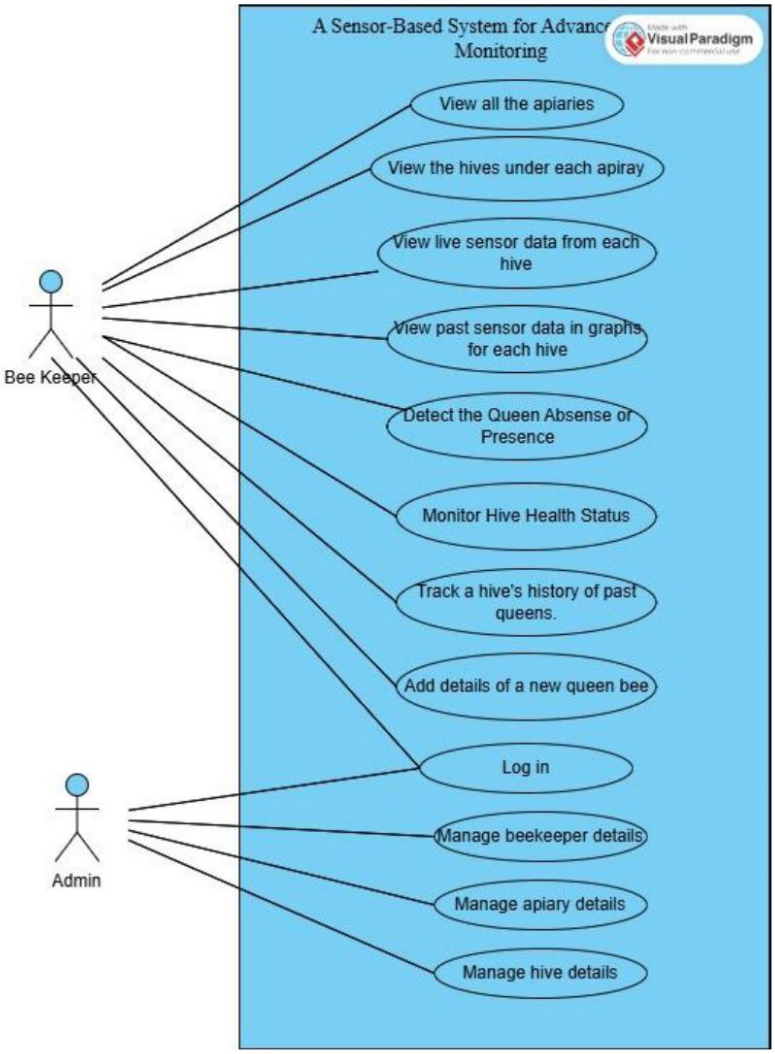


Figure 10 Use case diagram

2.4 Limitations of the Existing Systems

Table 1 The main limitations found in past research studies and the solutions for those identified limitations given through the developed solution.

Existing System	Main Limitation	Solution
(Cejrowski, 2018),	The bee queen detection model's **limited ability to generalize across different hives**.	The bee queen detection model is trained and tested using a dataset that contains data from multiple hives.
(Orlowska, 2022)	The model is not integrated into a real-world application	The developed queen detection model is integrated with a flutter mobile application.
(Branco, 2023)	The developed hardware prototype was only tested in a natural laboratory, and not with an actual real-world beehive.	The developed hardware prototype was placed and tested on two different real-world beehives at different places
(Braga, 2021)	The method used relies on cameras and image processing techniques, making it a relatively expensive approach.	Using only cheap temperature and humidity (DHT22) sensor to analyze the colony's health.
(Mansor, 2022), (Branco, 2023)	The developed mobile application has only specific and limited functionalities.	The developed mobile application provides beekeepers with a variety of features, including apiary and hive management, viewing live and historical data, determining the optimal harvesting time, gaining insights into hive conditions, and managing queen details

Chapter 03 | Requirements Specification

3.1 Functional Requirements

- The system must allow the users to log in successfully with correct credentials.
- The system must allow the admin to delete, update or edit user, apiary and hive details.
- The system must display only the apiaries that belong to the particular user after logging in.
- The system must display only the specific hives that belong to the selected apiary after choosing an apiary.
- The system must display real-time hive-specific sensor data.
- The system must provide past sensor data in graphs for trend and pattern analysis
- The system must allow the beekeeper to record bee sound through the app using the mobile microphone.
- The system must be able to process raw audio signals and predict whether the bee queen is present in the hive or not.
- The system must display the classification result: whether the bee queen is present or absent.
- The system must display a hive's history of past queens.
- The system must allow the beekeeper to add details of a new queen bee.
- The system must allow the beekeeper to update if the current queen has been replaced.

3.2 Non- Functional Requirements

- The system must load and display real-time data within 5 seconds.
- The system must support multiple hives and large datasets without performance degradation.
- The system must use data encryption and user authentication to protect sensitive information.
- The system must provide a user-friendly interface for easy navigation.
- The system must support both Android and iOS devices.
- The codebase should be modular for easy updates and bug fixes.

3.3 Hardware/Software Requirements.

Figure 11 Hardware Requirements

Figure 12 Software Requirements

Chapter 04| Feasibility Study

4.1 Operational Feasibility

Under operational feasibility it is accessed whether the system is able to survive in the real-world and is able to integrate into existing beekeeping practices in Sri Lanka. The final delivery is a simple, user-friendly mobile application, which the beekeepers can easily use without requiring much advanced technical skills. The existing approach to monitor the beehives is inspecting beehives manually through naked eye, which is time consuming. The system is able to successfully automate manual checkups, allowing the beekeepers to monitor the hives remotely, thus saving their time and effort. This ensures that the system can be successfully integrated with existing beekeeping practice. The system is able to handle multiple hive nodes across multiple apiaries. Furthermore, cloud-based data storage ensures that the hive data can be accessed from anywhere. This ensures scalability of the system. Additionally, the maintenance of the system is very minimal, limited to checking sensor coverings and ensuring proper power supply. Therefore, it is evident that the system is operationally feasible.

4.2 Technical Feasibility

Under technical feasibility, it is assessed whether the necessary technology and resources are available to develop the system. The required hardware and software components have been defined above. The ESP32 microcontroller, along with DHT22, rainfall sensors, and load cells, are available in Sri Lanka. Similarly, the necessary software components are also accessible. Access to Firebase services is straightforward, and devices with the required operating systems are readily available. Additionally, Arduino IDE, PyCharm, and Arduino Studio are free and easily accessible. The system requires a stable Wi-Fi connection for the ESP32 to send data to Firebase, which is also feasible. Therefore, it is evident that the system is technically feasible.

4.3 Outline Budget

Table 2 Expenses for the sensors and microcontroller

Category	Type	Purchased, Borrowed	Quantity	Price (Rs)
Sensors + Microcontroller	DHT22 Temperature & Humidity sensor	Purchased	2	950
	15 kg load cell	Borrowed	1	-
	Hx711 Load Cell Amplifier	Purchased	1	290
	Rainfall Sensor	Purchased	1	250
	ESP32-WROOM-32 Microcontroller	Borrowed	1	-
			Total	**2440**

Table 3 Expenses for power setup, wiring and connections of the IOT device

Category	Type	Purchased, Borrowed	Quantity	Price (Rs)
Power + Wiring + Connections	20W Sollar Panel	Purchased	1	4,000
	Lithium Ion Battery	Purchased	2	700
	2-Battery BMS	Purchased	1	110
	Printed Circuit Board	Borrowed	1	-
	2 pin terminal block connectors	Purchased	1	30
	3 pin terminal block connectors	Purchased	3	40
	4 pin terminal block connectors	Purchased	1	70
	7805 1A	Purchased	1	50

	regulator			
	Waterproof box enclosure for sensors	Purchased	1	1500
			Total	**7280**

Table 4 Expenses for the hive setup and beekeeping training

Category	Type	Purchased, Borrowed	Quantity	Price (Rs)
Hive Setup + Beekeeping Training	Wooden hive + Bee Colony	Purchased	1	10,000
	Load Cell Wooden Frame	Purchased	1	2,000
	Additional Wooden hive (for testing purposes)	Purchased	1	2,000
	Bee Keeping Training	-	-	5,000
	Continuous Management of the beehive	-	-	8,000
			Total	**27,000**

Full cost: ***Rs. 36,720***

Chapter 05 | System Architecture

5.1 Class Diagram

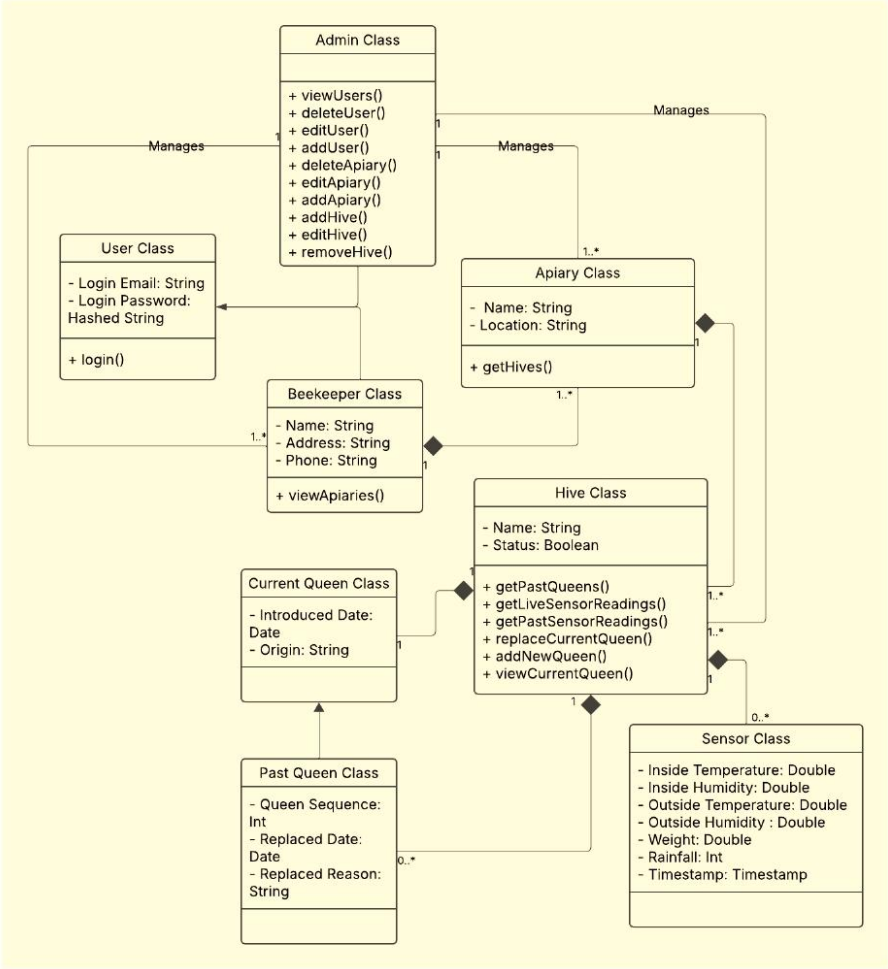


Figure 13 Class diagram

5.2 ER Diagram.

In this project, the Firebase was used for both the Realtime Database and the backend database. As Firebase is a NoSQL database, there is no strict need for an ER diagram. Therefore, a database design diagram that supports NoSQL databases has been drawn for both the backend database and the Realtime Database.

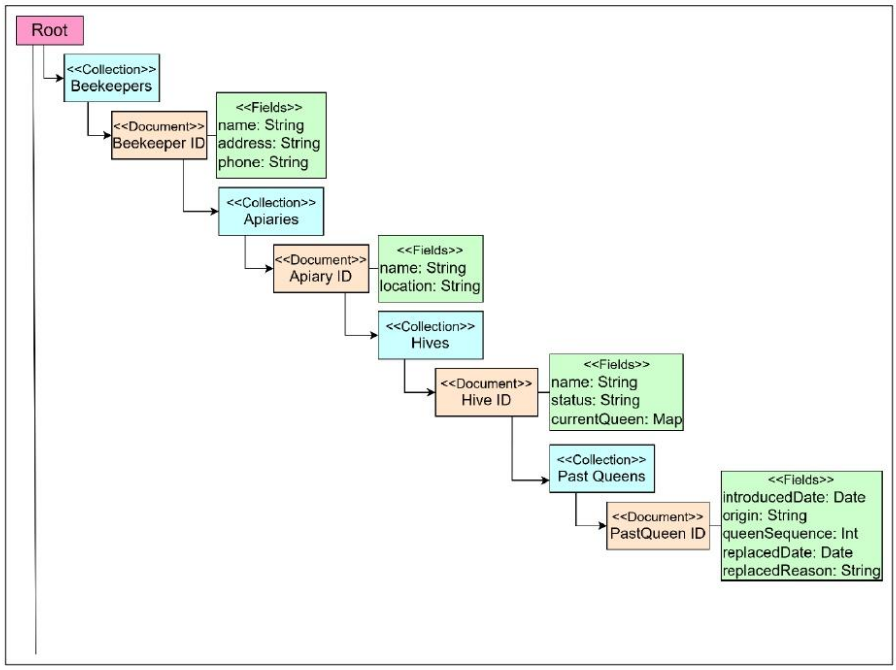


Figure 14 Database design diagram for the firesbase firestore database

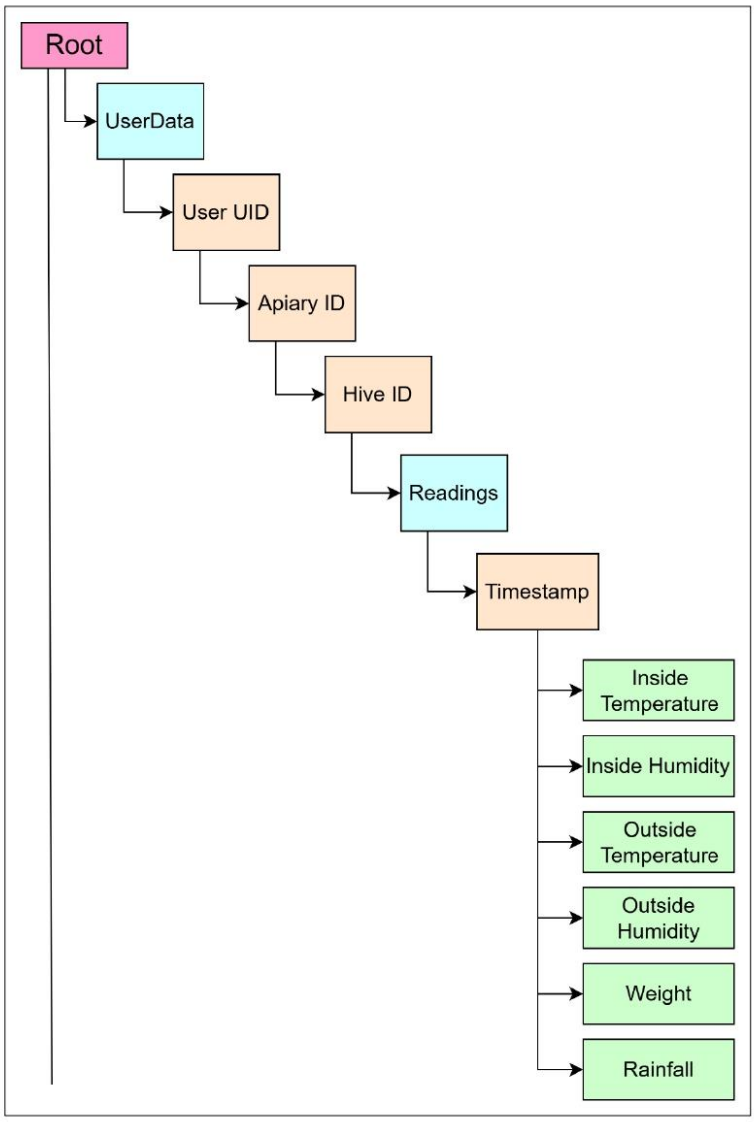


Figure 15 Database design diagram for the firebase realtime database

5.3 High Level Architecture Diagram

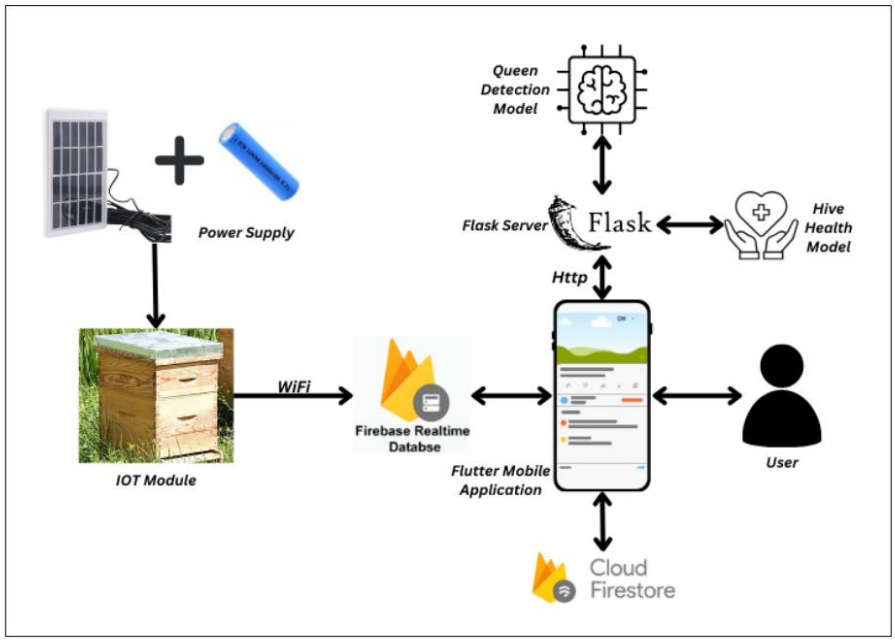


Figure 16 High level architecture diagram

Chapter 06 | Development Tools and Technologies

5.1 Methodology

The project follows the waterfall methodology, which ensures a structured and sequential approach to development. The following diagram illustrates the various tasks that were carried out at each stage of this project using the Waterfall methodology.

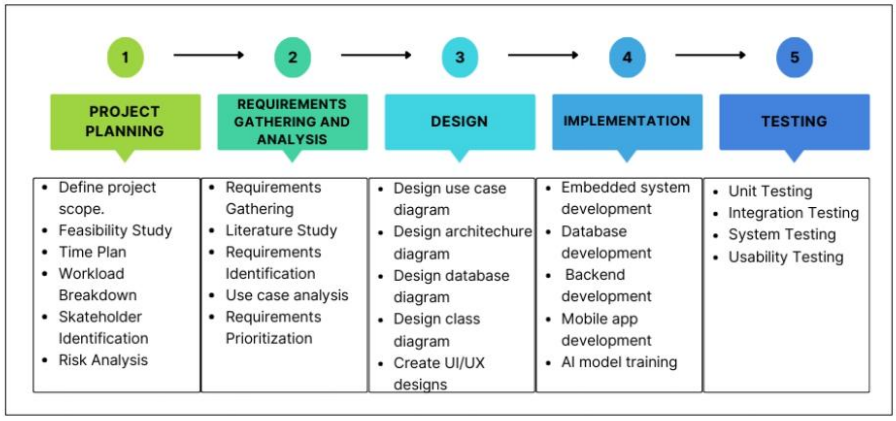


Figure 17 Tasks carried out in each stage in the Waterfall methodology

5.2 Programming languages and tools

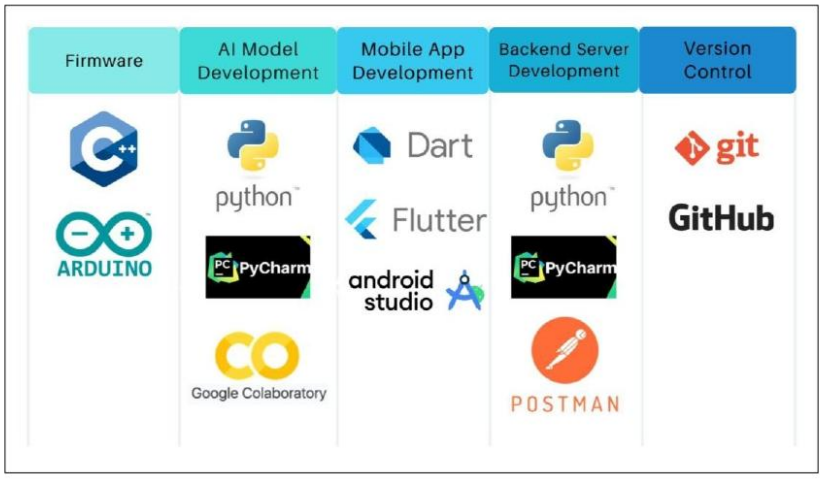


Figure 18 Programming languages and tools

5.3 Third Party Components and Libraries

In the firmware code certain libraries that are compatible with ESP32 have been used such as:

1. WiFi.h → To connect ESP32 with WiFi,
2. Firebase_ESP_Client.h → To connect Google Firebase with the ESP32.
3. Adafruit_Sensor.h → To provide a common interface for multiple sensors.
4. DHT.h → To read data from DHT22 sensors.
5. HX711.h → To read data from the HX711 load cell amplifier.

The following are all the other libraries and third-party components used in signal processing, data visualization, model development and backend development tasks.

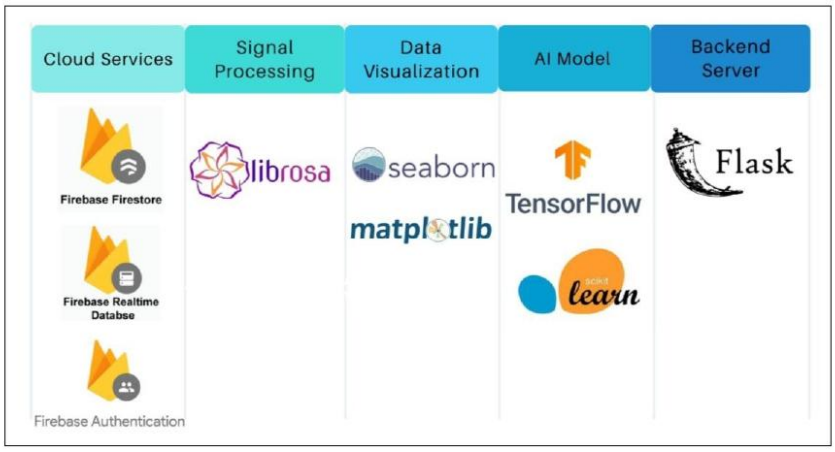


Figure 19 Third party components and libraries

5.4 Algorithms

Table 5 Algorithms used for signal processing and model development

Task	*Algorithms*
Signal Processing	MFCC (Mel-Frequency Cepstral Coefficients) Algorithm
Model Development	CNN (Convolutional Neural Network), K-Fold Cross Validation Algorithm

Chapter 07 | Discussion

Beekeepers in Sri Lanka face challenges with traditional beehive monitoring practices, that affect both the wellbeing of the colony and the management of the hives. This study delves into discussion on the sensor based advanced beehive monitoring system that was implemented to overcome these challenges faced by the beekeepers in Sri Lanka. Chapter 01 of the report highlighted the background and overview of the problem definition and the solution. Chapter 02 highlighted the requirements gathering techniques, while Chapter 03 delved into a deeper discussion of the analysis of the collected requirements. Chapter 04 focused on feasibility study. Chapter 05 portrayed the design diagrams. Finally, Chapter 06 highlighted the development tools and technologies. Various challenges were faced while implementing the system, such as hardware limitations with the PC for model training, limitations with power supply for the IOT module, and noise reduction for the audios that are being recorded from the mobile phone's microphone. The remaining work will largely focus on web application development for the admin side and improving model performance. As for the future work, integrating a mic in the embedded system, using tinyML techniques to compress the model, and enhancing security of the application will be considered.

References

1. Abdollahi, M. a. G. P. a. F. T. H., 2022. Automated Beehive Acoustics Monitoring: A Comprehensive Review of the Literature and Recommendations for Future Work. *Applied Sciences,* 12(8).
2. Andrijević, N. a. U. V. a. A. B. a. H. D. a. S. B., 2022. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics,* Volume 11, p. 783.
3. Anon., n.d.
4. Bayir, R. a. A. A., 2016. The monitoring of nectar flow period of honey bees using wireless sensor networks. *International Journal of Distributed Sensor Networks,* Volume 12.
5. Braga, D. a. M. A. a. S. F. a. P. V. a. A. A., 2021. An Intelligent Monitoring System for Assessing Bee Hive Health. *IEEE Access,* Volume 9, pp. 89009-89019.
6. Branco, D. C. a. J. M. a. H. M. a. F., 2023. BHiveSense: An integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices. *Journal of Open Innovation: Technology, Market, and Complexity,* Volume 9, p. 100110.
7. Carlson, R. D., 2015. *The Honey Bee and Apian Imagery in Classical Literature,* Seattle, WA: University of Washington.
8. Cazier}, A. {. B. a. D. {. G. a. R. R. a. E. {. H. a. B. {. F. a. J. {., 2020. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture,* Volume 169, p. 105161.
9. Cecchi, S. a. S. S. a. T. A. a. O. S., 2020. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors,* 20(9), p. 2726.
10. Cejrowski, T. a. S. J. a. M. H. a. G. D., 2018. Detection of the Bee Queen Presence Using Sound Analysis. In: s.l.:Springer, pp. 297-306.
11. Christina T. Liang, B. G. K. H. W. T., 2016. The Role of Honey Bees as Pollinators in Natural Areas. *Natural Areas Journal,* 36(4), pp. 478-488.
12. Diogo Braga, Ana Mudureura, Fabio Scotti, Vincenzo Piri, Ajith Abraham, 2021. *An Intelligent Monitoring Systemfor Assessing Bee Hive Health.* Portugal, s.n.
13. Doinea, M. et al., 2024. IoT Embedded Smart Monitoring System with Edge Machine Learning for Beehive Management.. *International Journal of Computers, Communications & Control,* 19(4).
14. Dorin, A. Z. a. A., 2023. A framework for better sensor-based beehive health monitoring. *Computers and Electronics in Agriculture,* 210(0168-1699), p. 107906.
15. E. R. H. S. S. Ediriweera, N Y S Premarathna, 2012. *Medicinal and cosmetic uses of Bee's Honey – A review.* Sri Lanka, s.n.

16. E.R.H.S.S. Ediriweera, N. Y. S. P., 2012. Medicinal and cosmetic uses of Bee's Honey – A review. *AYU (An International Quarterly Journal of Research in Ayurveda),* 33(2), pp. 178-82.
17. Edwards-Murphy, F. a. M. M. a. W. P. a. O. J. a. P. E., 2016. B+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture,* Volume 124, pp. 211-219.
18. Hegde, A. D. a. A. P. a. S., 2023. *HiveLink, an IoT based Smart Bee Hive Monitoring System.* [Online]
Available at: https://arxiv.org/abs/2309.12054 [Accessed 19 10 2024].
19. Isayas Asefa Kebede, H. F. A. D. A. G. D., 2024. Bee products and their processing: a review. *Pharmacy & Pharmacology International Journal,* 12(2), pp. 5-12.
20. Khan, S. A., 2023. Review of the Role of Bees as Ecosystem Engineers in Nature. *Science Reviews - Biology,* 2(4), pp. 1-6.
21. Kridi, D. S. a. d. C. C. G. N. a. G. D. G., 2016. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Elsevier Science Publishers B. V.,* 127(C), p. 221–235.
22. Mansor, M., 2022. *IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping using Lora Technology..* s.l, s.n.
23. Orlowska, A. a. F. D. a. G. J.-P. a. C.-R. D., 2022. Honey Bee Queen Presence Detection from Audio Field Recordings Using Summarized Spectrogram and Convolutional Neural Networks. In: s.l.:s.n., pp. 83-92.
24. Peter Hristov, R. S. N. P. B. N., 2020. Factors Associated with Honey Bee Colony Losses: A Mini-Review.. *Veterinary sciences,* 7(4), p. 166.
25. Pikri, M. a. I. S. a. F. N. a. M. M. a. M. S., 2022. IoT-Based Temperature and Humidity Real-Time Monitoring System for Beekeeping Using LoRa Technology. In: s.l.:s.n., pp. 334-342.
26. Rector, B. a. M. G. a. H. N., 2008. Within-day variation in continuous hive weight data as a measure of honey bee colony activity. *Apidologie,* Volume 39, pp. 694-707.
27. sayas Asefa Kebede,Haben Fesseha, Gebremeskel, Abrahim Dawed Ahmed, Gelan Dule, 2024. *Bee products and their processing: a review.* Ethiopia, s.n.
28. Seritan, G. C. a. E. B.-A. a. A. F. C. a. A. F. C. a. T. S., 2018. *Low cost platform for monitoring honey production and bees health.* s.l., 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR).
29. Shaden A. M. Khalifa, E. H. E. A. A. S. A. A. A. E.-W. A. F. A. S. G. M. M. F. A. C. Z. S. H. D. M. M. A.-D. M. F. H. G. K., 2021. Overview of Bee Pollination and Its Economic Value for Crop Production. *Insects 2021,* 12(8), p. 688.
30. Uthoff, C. a. N. H. M. a. V. B. M., 2023. Acoustic and vibration monitoring of honeybee colonies for beekeeping-relevant aspects of presence of queen bee and swarming. *Computers*

and Electronics in Agriculture, Volume 205.

31. Vidushi Patel, N. P. E. B. L. B. &. B. B., 2021. Why bees are critical for achieving sustainable development. *Ambio,* 50(1), pp. 49-51.
32. Weiss, M. a. M. P. a. F. W. a. S. L. a. S. T. a. M. B. a. A. K., 2017. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie,* Volume 48.
33. Whitfield, B. G., 1956. Virgil and the Bees: A Study in Ancient Apicultural Lore. *Greece and Rome,* 3(2), p. 99–117.
34. Yiyao Zheng , Xiaoyan Cao, Shaocong Xu, Shihui Guo, Rencai Huang, Yingjiao Li, Yijie Chen, Liulin Yang, Xiaoyu Cao,Zainura Idrus, Hongting Sun, 2024. *Intelligent beehive monitoring system based on internet of things and colony state analysis.* China, s.n.
35. Yiyao Zheng, X. C. S. X. S. G. R. H. Y. L. Y. C. L. Y. X. C. Z. I. H. S., 2024. Intelligent beehive monitoring system based on internet of things and colony state analysis. *Smart Agricultural Technology},* 9(1), p. 100584.
36. Zacepins, A. a. P. A. a. O. V. a. K. A. a. E. S., 2017. Solution for automated bee colony weight monitoring. *Agronomy Research,* Volume 15, pp. 585-593.

Appendix I: Records of supervisory meetings

UNIVERSITY OF PLYMOUTH

NSBM GREEN UNIVERSITY TOWN

PUSL3190 Computing Individual Project
Student Progression Report
[Student Copy]

01. Student Name ....Nisal Liyanage....
02. Plymouth Index Number ....10898808....
03. Degree Program ....BSc. Computer Science....
04. Supervisor Name ....Ms. Hirushi Dilpriya....
05. Project Title ....Beehive Data Analysis....

Meeting Number	Meeting 01	Meeting 02	Meeting 03	Meeting 04	Meeting 05	Meeting 06	Meeting 07
Date	18/09/2024	27/09/2024	04/11/2024	12/11/2024	11/12/2024	18/12/2024	07/02/2025
Student Signature							
Supervisor Signature							

Meeting Number	Meeting 08	Meeting 09	Meeting 10	Meeting 11	Meeting 12	Meeting 13	Meeting 14
Date	2025/02/14	2025/05/02	2025/04/23				
Student Signature							
Supervisor Signature							

[presentation]

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 1

Date : 18/09/2024

Project Title : Beehive Data Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

* project Overview
* what objectives will be in the project scope.

Items to be completed before the next supervisory meeting:

① ~~Collecting~~ find an online dataset related to beehive metrices.

② perform some sort of Explotary Data Analysis.

Supervisor (Signature & Date)

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No: 2

Date : 27/09/2024

Project Title : Beehive Data Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

1. The findings of the EDA, from the previous meeting.
2. What else can be do?

Items to be completed before the next supervisory meeting:

1. Define the project scope clearly.
2. Explore more similar Datasets.

Supervisor (Signature & Date)

Instructions to the supervisor: Do not sign if the ...

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : 04/11/2024

Project Title : Beehive Data Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

1. Compare & contrast the datasets & the analysis permored on each
2. Discussed the limitations of these datasets.
3. Discussed about the project scope & IOT part

Items to be completed before the next supervisory meeting:

1. ~~The first~~ Permorm full analysis on the queen Detection Dataset.
2. Finish the IOT setup

.....

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank

UNIVERSITY OF PLYMOUTH NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : 12/11/2024

Project Title : Beehive Analysis

Name of the Student : ~~BSc Computer Science~~ Misal Liyanage

Students ID : ~~MSIT~~ 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

1. IOT module's progress.
2. The ~~perm~~ formance of the queen detection model.

Items to be completed before the next supervisory meeting:

1. Place the device & take a dataset
2. Enhance the model's performance.
3. Start the mobile app implementation.

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank

UNIVERSITY OF PLYMOUTH NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : 11/12/2024

Project Title : Beehive Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hrushi Dilpriya.

Items discussed:

1. The results of the queen model
2. What else can we improve?
3. Mobile app's Figma UI Design

Items to be completed before the next supervisory meeting:

1. Start the ~~mo~~ mobile app implementation.
2. Keep working with the IOT module.
3. Improve the UI

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : ~~18/12/2024~~ 07/02/2025

Project Title : Beehive Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Ithushi Dilpriya.

Items discussed:

* Overall summary of what have done so far.
* What are to be done in the future.

Items to be completed before the next supervisory meeting:

1. Research on the hive health analysis model
2. Implement until the live dashboard page in the mobile app

.....

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank

UNIVERSITY OF PLYMOUTH NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : 14/03/2025

Project Title : Beehive Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya.

Items discussed:

* Overall work flow
* Discussed the findingings related to the health anlaysis. model
* Discussed the workflow the app

Items to be completed before the next supervisory meeting:

1. Fully implement the mobile app
2. Train the hive haath model
3. ~~Start~~ Implement the web portal for admin

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign if the above h**

UNIVERSITY OF PLYMOUTH | NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : ~~09/~~ 23/04/2025

Project Title : Beehive Data Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

① The presentation
② Improve UI
③ Make model visualizations more appealing

Items to be completed before the next supervisory meeting:

—

.....

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank.

UNIVERSITY OF PLYMOUTH NSBM GREEN UNIVERSITY TOWN

Final Year Project – Supervisory meeting minutes

Meeting No:

Date : 02/05/2025

Project Title : Beehive Data Analysis

Name of the Student : Nisal Liyanage

Students ID : 10898808

Name of the Supervisor : Ms. Hirushi Dilpriya

Items discussed:

1. Overall workflow
2. Final Report

Items to be completed before the next supervisory meeting:

1. Improve the report

Supervisor (Signature & Date)

Instructions to the supervisor: **Do not sign** if the above boxes are blank