Product Requirements Document (PRD)

Risk	Perspective
Value	User
Usability	User
Feasibility	Engineering
Viability	Business

Problem Description

> "The framing of a problem is often far more essential than its solution." — **Albert Einstein**

The home beekeeper
The practice of beekeeping has been around for a long time, and the common practices of beekeepers has not changed much since it was “modernized” in the middle of the 1800’s. While many agricultural practices have evolved to take advantage of modern technology that automates monitoring tasks, beekeeping largely requires direct inspection of hives by beekeepers. And yet hives are very complex and generate a large amount of data. And there are consequences if the data is not acted upon soon enough. Sometimes a problem may not be discovered until it’s too late.

The Agricultural Business:
Agricultural businesses that rely on pollinators, such as honeybees, face challenges in optimizing their crop yields. These businesses are interested in increasing crop pollination rates by honeybees and improving hive health. To achieve this, they need tools to monitor the activity and health of bee colonies effectively. Additionally, they require a way to ensure the well-being of the bees while they are on their farms. A problem they'd like to address is finding a technology solution that provides real-time hive monitoring and allows them to optimize their land use while supporting bee colonies and overall environmental health.

The Environmental Conservationist:
Environmental conservationists are concerned about the decline in bee populations and its potential impact on ecosystems. They need a way to track and monitor bee colonies in the wild or in natural habitats to better understand the factors contributing to bee population decline. Monitoring bees in the wild can be challenging and invasive. Conservationists seek a non-disruptive technology that can help them collect data on wild bee populations and their behavior, allowing for informed conservation efforts and the preservation of biodiversity.

Scope

The initial scope of this problem is focused on creating a consumer grade product that can be used to manage hive data and provide early detection and prevention of health issues. Wireless sensors would provide real-time data for temperature, humidity, and bee behaviors and more. This monitoring system would organize its data and provide automatic notifications regarding important events. A stretch goal would be to include automatic responses to certain event occurrences, like increasing/decreasing humidity.

Use Cases

Early Disease Detection
Use Case: A beekeeper receives an immediate alert on their smartphone when the monitoring system detects unusual temperature fluctuations in the hive. This notification allows the beekeeper to investigate and address potential health issues in the colony, preventing the spread of diseases.

Optimal Hive Environment
Use Case: An agricultural business utilizes the system to maintain optimal environmental conditions in the hives during all weather conditions. The monitoring system automatically adjusts temperature and humidity levels to ensure bee comfort, which results in increased pollination rates and improved crop yields.

Remote Hive Monitoring and management
Use Case: Commercial beekeepers with hives in multiple locations remotely monitor all their hives from a centralized dashboard. They can quickly identify and address issues, even from a distance, ensuring the well-being of their bee colonies and reducing the need for on-site visits.

Use Case: Beekeepers can remotely access hive data and make decisions about hive management. For instance, they can schedule inspections only when the system detects anomalies, reducing disruptions to the bees and saving time and resources.

Wild Bee Population Monitoring
Use Case: Environmental conservationists install sensors in wild bee habitats. The system collects data on the behavior and health of wild bee populations without disturbing their natural environment. Researchers use this data to assess the well-being of wild bee populations and support conservation efforts.

Purpose and Vision (Background)

The B Ring envisions a future where beekeeping is revolutionized by advanced technology. Bees are complex creatures, and hives are even more intricate. Experienced beekeepers often grapple with the overwhelming amount of data associated with beekeeping. The B Ring serves a higher purpose – to simplify the management of hive data and harness the power of automation for beekeepers of all levels. By separating valuable insights from the noise, it transforms into a comprehensive hive monitoring solution and data management system. The vision is to empower beekeepers to make informed decisions and safeguard bee colonies efficiently, preserving the crucial role of bees in our ecosystems and agricultural economies.

Bees play an indispensable role in maintaining a healthy environment and sustaining an efficient agricultural economy. However, the complexities of hive management have limited the potential benefits that bees can provide. The B Ring represents an innovative approach that has the power to amplify the contributions of bees while simplifying beekeeping, catering to both hobbyist beekeepers and large-scale honey producers.

Stakeholders

Beekeepers (Users):

Update Frequency: Regular, especially during hive inspections and problem detection.
Type of Updates: Real-time data on hive conditions, alerts for issues.
Decision-Makers: Main decision-makers regarding hive management.
Decisions: Adoption of the B Ring system, implementing automation.
Requirements: Detailed, real-time hive data for informed decisions.

Agricultural Research Teams:

Update Frequency: Occasional, especially for research collaborations.
Type of Updates: Research findings related to bee behavior and hive conditions.
Decision-Makers: Research leads and Principal Investigators.
Decisions: Collaborative research projects and funding applications.
Requirements: Scientific data on the B Ring's impact on bee behavior.

Hobbyist Beekeepers' Associations:

Update Frequency: Periodically for outreach and collaboration.

Type of Updates: Workshops, webinars, and educational materials.
Decision-Makers: Association leaders.
Decisions: Collaborative educational events and initiatives.
Requirements: Information on how the B Ring aids hobbyist beekeepers.

Environmental Conservation Organizations:

Update Frequency: Periodically, especially for partnership opportunities.
Type of Updates: Data on bee health and environmental impact.
Decision-Makers: Environmental Program Managers.
Decisions: Collaborative projects to protect bee populations.
Requirements: Information on the environmental benefits of the B Ring.

Beekeeping Product Marketers:

Update Frequency: Regular for marketing materials.
Type of Updates: Product descriptions, promotional content.
Decision-Makers: Marketing Managers.
Decisions: Marketing strategies and campaigns.
Requirements: Data on the B Ring's unique selling points.

Potential Investors:

Update Frequency: During fundraising rounds.
Type of Updates: Business growth, investment opportunities.
Decision-Makers: Investors.
Decisions: Investment and funding decisions.
Requirements: Information on business progress and investment opportunities.

Preliminary Context

Assumptions

Data Sorting Feasibility: We assume that the vast amount of data generated by the bee monitoring sensors can be effectively sorted and analyzed using appropriate sensors and software, considering the data's complexity and volume.

Beekeeper Adaptability: We assume that beekeepers, who traditionally rely on manual methods, can readily adapt to data-driven techniques and incorporate the technology into their beekeeping practices.

Accessibility of Environmental Monitoring: We assume that environmentalists and conservationists monitoring wild bee populations can utilize tools in the form of mobile applications to collect and analyze data effectively.

Cross-Platform Implementation: We assume that the monitoring algorithms developed for the project can be efficiently implemented on various platforms, including iOS, Android, and PC, to cater to a wide range of users and their preferred devices.

Availability of Open Source Sensor Libraries: We assume that open-source libraries and resources for bee monitoring sensors and related technologies are accessible and readily obtainable to support the project's sensor technology needs.

Sensor Data Accuracy: We assume that the sensor data collected will be accurate and reliable for monitoring bee behavior, hive conditions, and environmental factors, considering the critical nature of the data for informed decision-making.

Bee Health and Behavior Consistency: We assume that the behavior and health patterns of bees remain relatively consistent, allowing the monitoring algorithms to detect anomalies and health issues effectively.

Constraints

Power and Efficiency Constraints: The project must adhere to constraints related to power consumption and efficiency, ensuring that the B Ring device does not consume excessive electricity and can operate reliably on limited power sources, such as batteries or solar panels.

Wireless Broadcast Regulation Constraints: The project is subject to regulatory constraints related to wireless data broadcast, including adherence to specific frequency bands, transmission power limits, and licensing requirements for radio frequency usage.

Deadline Constraints: There are tight project deadlines in place to bring the product to market by spring 2024. Keeping to these time constraints is essential to meet market demand and business goals. The project faces time constraints due to the involvement of students who may have limited availability outside of academic commitments.

Adoptability Constraints: Traditional beekeepers may be resistant to adopting new, data-driven beekeeping methods. The project must address the challenge of convincing beekeepers to embrace this technological change and provide resources and support for a smooth transition.

Budgetary Constraints: The project operates within specific budgetary constraints that limit expenditures for research, development, and production. Effective resource management is crucial to stay within budget and achieve the project's financial goals.

Technical Resource Constraints: Technical constraints may arise due to limitations in hardware, software, or sensor technology. The project must work within these constraints and find solutions that optimize functionality while working within technical limitations.

Resource and Expertise Constraints: The project may face constraints related to the availability of skilled staff and expertise in areas such as sensor technology, data analysis, and software development. Maximizing the effective use of available resources and expertise is essential.

Dependencies

Market Adoption: The project's success depends on the adoption of new technology by beekeepers, which may significantly differ from traditional beekeeping practices. The extent to which beekeepers embrace data-driven techniques affects the project's market penetration and acceptance.

Testing and Development by Independent Beekeepers: The project may rely on the active participation of independent beekeepers who serve as testers and contributors to further develop the product. Feedback and insights from beekeepers are essential for refining the technology.

Continuation of Previous Project Goals: The project builds upon the progress made in previous project sprints. It is dependent on the successful achievement of deadlines, goals, and milestones set in earlier stages. Meeting these past commitments is critical for the project's continuity.

Availability of Data: Data dependencies exist, particularly for real-time environmental and hive data. The project relies on the availability of accurate and up-to-date data to provide beekeepers with valuable insights and monitoring capabilities.

External Partners and Vendors: External vendors or partners may be involved in the supply of necessary components or sensor technology. The project depends on the timely delivery of these components and the reliability of external parties.

Sequential Tasks in Product Development: The project's development process involves various sequential tasks, from sensor implementation to software integration. These tasks need to be completed in a specific order, and delays in any one of them can affect the overall project timeline.

Market Assessment and Competition Analysis

Manual Data Collection and Ledger Systems:

Fit: Manual data collection is simple to set up and does not require sophisticated technology. It's suitable for small-scale beekeepers.

Not a Good Fit: It lacks automated tracking features, making it impractical for beekeepers with multiple hives. Additionally, it does not provide real-time data insights.

Apiary Book:

Fit: Apiary Book is a widely used solution globally for beekeeping management. It offers valuable features and insights.
Not a Good Fit: While it is a strong competitor, it may not address specific regional needs, such as those of Pacific Northwest (PNW) beekeepers. It may lack certain features tailored to this specific region.

Hivetracks:

Fit: Hivetracks offers a user-friendly interface for hive management, making it accessible to beekeepers.

Not a Good Fit: This solution may not include the suite of sensors and real-time data collection capabilities that the B Ring project aims to provide. It focuses more on hive management and crowd-sourced data.

Analysis:

The existing alternatives primarily focus on hive management and data organization. They lack comprehensive real-time data collection and sensor integration, which is a key focus of the B Ring project.

Manual data collection is limited by its lack of automation, making it impractical for beekeepers with multiple hives.

Apiary Book is a strong global competitor but may not cater to the specific needs and regional requirements of PNW beekeepers.

Hivetracks offers user-friendly hive management but lacks the extensive sensor capabilities of the B Ring project.

The B Ring project aims to position itself as a specialized and sensor focused solution that addresses the specific needs of PNW beekeepers while providing real-time data collection and insights. While alternatives exist, the project's unique focus on data collection, automation, and regional specialization differentiates it from the competition.

Target Demographics (User Persona)

Lisa is a 45-year-old small-scale honey producer who manages a modest beekeeping operation. She is passionate about honey production and wants to ensure her bees are healthy and productive. Lisa seeks an affordable and user-friendly solution to monitor her hive's health and receive timely notifications for honey collection.

Bart is a 28-year-old CEO of a large-scale honey production company that operates in multiple states. He oversees a vast network of beehives and needs an advanced system to monitor the health and productivity of these hives. Bart is looking for a comprehensive solution to manage a large number of hives efficiently.

Homer is an environmentalist with a keen interest in the vital role that bees play in pollinating crops. He wants to contribute to environmental research by tracking the bee populations required for pollination in specific crop areas. Homer seeks a solution that allows him to collect and analyze data related to bee activities and their impact on pollination.

Requirements

User Stories and Features (Functional Requirements)

User Story	Feature	Priority	GitHub Issue	Dependencies
As a hobby beekeeper, I need automated notifications for events like optimal honey collection, so I can harvest honey at the right time.	The system must send notifications to the user when the conditions for honey collection are optimal.	Must Have	TBD	Real-time data collection and analysis capabilities.
As a hobby beekeeper, I need solutions to be cost-effective and user-friendly, ensuring that the system is accessible and affordable.	The product should provide a cost-effective and user-friendly solution for beekeeping.	Must Have	TBD	Scalable hardware and software design.
As a large-scale beekeeper, I need scalable solutions for large-scale honey production that provide technology for data analysis, enabling efficient management of multiple hives.	The system must be scalable and capable of monitoring and managing a large number of beehives efficiently.	Must have	TBD	Scalable hardware and software design.
As a large-scale beekeeper I need the product to provide a data export feature for large-scale beekeepers to perform extensive data analysis.	The product should offer real-time monitoring of hive conditions, collecting data for analysis.	Should have	TBD	Data export and analysis tools.
As an environmental bee supporter, I need tools for environmental research and analysis to understand the impact of bee populations on crop pollination.	The system should include features for data collection and analysis related to bee populations and crop pollination.	Must have	TBD	Data collection mechanisms and analysis tools.

As an environmental bee supporter, I'd like the product to support remote data access, enabling me to monitor bee populations and pollination from anywhere.	The system should include remote access features, allowing users to monitor bee-related data remotely.	Should have	TBD	Remote access infrastructure and user-friendly interface design.
As a user, I don't require the ability to control all hive equipment remotely.	The product will not include features for remote control of most hive equipment.	Will not have	N/A	Remote control may have safety and regulatory concerns, and it falls outside the primary monitoring and data analysis focus.
As a user, I'm not looking for support for hive-related e-commerce functions	The system will not offer e-commerce features related to hive products	Will Not have	N/A	E-commerce functionality falls outside the scope of hive data collection and analysis

Non-Functional Requirements

Real-time Data Collection:

- The system must collect and process hive data in real-time with a delay of no more than 5 seconds from the moment of data acquisition.
- The real-time data collection should apply to all sensors, including temperature, humidity, bee behavior, and environmental factors.
- The system should support simultaneous real-time data collection from multiple hives, with each hive data stream isolated.

Performance:

- The system should have a response time of less than 500 milliseconds for user interactions, ensuring a smooth and responsive user experience.
- The data storage and retrieval process should handle large datasets efficiently, and queries should return results within 2 seconds.

Scalability:

- The system should be designed to scale up to support an increased number of hives. Ideally it should be able to handle data from at least 1,000 hives simultaneously.
- Scalability should be cost-effective, allowing for the addition of new hives without significant infrastructure changes.

Security:

- All communication between the data collection device and the central server must be encrypted using SSL/TLS to ensure data privacy and integrity.
- User authentication and authorization should be implemented securely to protect user accounts and hive data.

Reliability:

- The system must be available for access 24/7, except during scheduled maintenance windows. Downtime should be minimized during maintenance, with advance notifications to users.
- Frequent backups and redundancy should be in place to ensure data integrity.

Budget and Resource Constraints:

- The project should be executed within the given budget and time frame.
- The system should utilize cost-effective technology solutions without compromising functionality and quality.

Data Privacy and Compliance:

- The product should provide clear data usage policies and comply with relevant data protection regulations.

Accessibility:

- The user interface of the system should be designed to be accessible to users with disabilities.

Cross-Platform Compatibility:

- The product should work seamlessly on multiple devices and screen sizes, including desktop computers, tablets, and mobile devices, without sacrificing usability.

Documentation and Coding Standards:

- All code should be well-documented, following coding standards and best practices to ensure maintainability and collaboration among developers.
- Documentation should include a comprehensive user guide to assist users in utilizing the system effectively.

Data Requirements

Hive Information:
Data Structure: JSON or XML
Details: Hive ID, location, date of setup, hive type, owner information, and current status (active or inactive).

Environmental Data:
Data Structure: JSON or XML
Details: Real-time sensor data including temperature, humidity, air quality, light intensity, and weather conditions. Timestamps and sensor IDs for each data point.

Bee Behavior Data:
Data Structure: CSV
Details: Behavior logs with timestamps, hive ID, and descriptions of bee behaviors such as foraging, swarming, comb building, and brood care.

Hive Health Data:
Data Structure: JSON or XML
Details: Records of hive health indicators, including disease prevalence, pest infestations, colony strength, honey production, and queen activity.

User Data:
Data Structure: Relational Database (such as MySQL)
Details: User profiles with personal information, login credentials, hive ownership, and notification preferences.

Sensor Calibration Data:
Data Structure: JSON or XML
Details: Information about sensor calibration, including calibration coefficients and calibration timestamps. Used to ensure data accuracy.

Notification Data:
Data Structure: JSON or XML
Details: Records of notifications sent to users, including message content, recipient ID, timestamp, and delivery status.

Logs and Metadata:
Data Structure: Log Files
Details: System logs and metadata, including system events, errors, user activities, and API requests. Timestamps and log levels for debugging and auditing.

Backup Data:
Data Structure: Encrypted Database Backup
Details: Encrypted backups of all critical data to ensure data recovery and integrity in case of system failure.

Analytics Data:
Data Structure: TBD
Details: Aggregated and historical data for data analysis and research purposes, including trends, patterns, and insights.

Integration Requirements

Without knowing specific information about the software and hardware systems it’s hard to come up with integration requirements. Some possible requirements can be assumed from what we’ve assumed about the data requirements, and the functional and non-functional requirements.

Data Integration Requirements:

Weather Data Service:
Integration Point: Some external weather API
Data Exchange: Real-time weather data (temperature, humidity, precipitation, wind speed, etc.) for hive locations.

Geospatial Mapping Service:
Integration Point: An external Mapping Service (for example Google Maps)
Data Exchange: Geospatial data for visualizing hive locations and environmental data on maps.

Sensor Calibration Service:
Integration Point: Internal Calibration Tool
Data Exchange: Retrieval and upload of sensor calibration parameters.

Functional Integration Requirements:

User Notification Service:
Integration Point: A Notification Module
Functionality: Sends real-time notifications to users based on hive data and events.
Requirements: Immediate and reliable notification delivery with options for user preferences.

Hive Health Analysis Module:
Integration Point: Internal Data Analysis Module and external Data analysis API
Functionality: Analyzes hive health data and provides health reports to beekeepers.
Requirements: Fast and accurate data processing with the possibility to generate insights and recommendations.

Behavior Pattern Recognition:
Integration Point: Internal Machine Learning Module or API
Functionality: Identifies and logs bee behavior patterns from sensor data.
Requirements: Real-time pattern recognition, with continuous learning and pattern refinement.

Non-Functional Integration Requirements:
Security and Authentication:
Integration Point: User Data Management Module
Requirements: Secure authentication and access control to protect user data.

Performance Monitoring:
Integration Point: System Logging and Monitoring Module or external API
Requirements: Constant system monitoring for performance, errors, and real-time alerts.

User Interaction and Design

The layout could first be based on a website style design that works a bit like the canvas interface. This could then be adapted to work as a phone ap or be expanded into a complete software product. The basic design could be built around a dashboard that abstracts the feature as cards.

Dashboard Mockup:

The dashboard is the central hub where users can quickly access critical hive data immediately. The design should focus on simplicity, clarity, and intuitive data visualization.

Interactions: Users can interact with widgets displaying hive status, environmental/weather conditions, and notifications. They can also set preferences for the display of data.

Mockup Description: A clean and organized dashboard featuring real-time data cards for each hive, weather updates, and notifications. Users can click on a hive card for detailed information.

Hive Detail Page:
When a user clicks on a hive from the dashboard, they should be presented with a detailed view that offers in-depth information about that hive's health and activity.

Interactions: Users can scroll through historical data charts, view sensor data, and access hive-specific settings.

Mockup Description: The hive detail page includes line charts showing temperature and humidity trends, images from the hive's camera, and buttons for settings adjustments.

Notification Center:
Notifications are crucial for real-time alerts. The notification center should provide a straightforward way to view and manage alerts.

Interactions: Like an email system, users can see notifications categorized by importance, mark them as read, or possibly take specific actions in response to alerts.
Mockup Description: The notification center displays notifications with time stamps, and users can mark them as read, dismiss them, or click to see more details.

Settings Panel:
For changing settings or user preferences, the settings panel should be user-friendly.

Interactions: Users can customize notification preferences, sensor calibration, and connectivity settings.
Mockup Description: A clean settings panel with tabs for general preferences, notifications, sensor settings, and connectivity options.

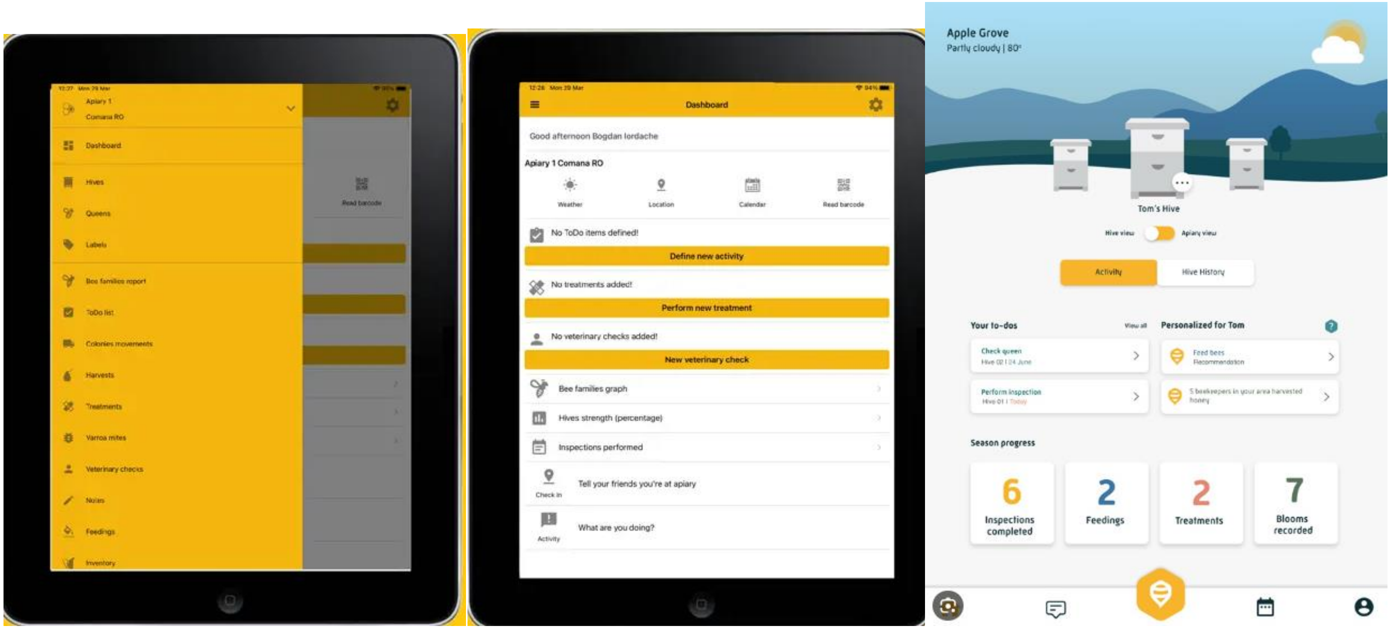
Data Analysis and automation:
System should present insights clearly and actionable recommendations.

Interactions: Users can view summarized insights, compare hive performance, and set up automated responses based on insights.

Mockup Description: A data analysis page displaying insights about hive health, environmental conditions, and suggested actions.

A few competitors like Hive Tracker and Apiary Book, have a similar interface design. Inspiration for our product could initially come from products that already have strong market adoption.

Potential Dashboard interfaces:



Milestones and Timeline

The milestones and timeline, like the rest of this document, is an **iterative** process. Your user stories will get more detailed as you approach implementation, hence you will have better estimates, and will be able to revise your timeline.

Milestone 1: Project Setup (1-2 weeks)

- Define project scope and requirements.
- Gather necessary resources and tools.
- Set up the development environment.
- Create project repository and initial documentation.

Milestone 2a: User Dashboard (4-6 weeks)

- Display real-time hive data, including temperature, humidity, and status.
- Implement data visualization and analytics tools.
- User story: "As a beekeeper, I want to access a dashboard to monitor the status of my hives and analyze collected data."

Milestone 2b: Data Collection (4-6 weeks)

- Create data structures and APIs for storing hive data.
- Implement real-time data collection from hive sensors and Store collected data in a database.
- User story: "As a beekeeper, I want to collect data from hive sensors in real-time for analysis."

(Milestone 2a and 2b should happen concurrently)

Milestone 3: Automated Notifications (3-4 weeks)

- Implement automated notification system for important hive events.

- Set up alerts for temperature or humidity deviations.
- Send notifications to beekeepers via email or mobile app.
- User story: "As a beekeeper, I want to receive notifications when hive conditions require my attention."

Milestone 4: Scalability and Performance Optimization (3-4 weeks)
- Optimize the system for scalability to handle a growing number of hives.
- Enhance system performance for real-time data processing.
- Perform load testing and optimization.
- User story: "As a large-scale beekeeper, I want a system that can scale with my expanding honey production."

Milestone 5: Documentation and Testing (2-3 weeks)
- Create user guides and developer documentation.
- Conduct thorough testing, including unit, integration, and user acceptance testing.
- Address any bugs or issues identified during testing.
- User story: "As a user, I want clear documentation for using the system effectively."

Milestone 6: Deployment and User Training (2-3 weeks)
- Prepare for the product launch.
- Deploy the system to production.
- Provide training and support to users.
- User story: "As a beekeeper, I want training to effectively use the system."

Goals and Success Metrics

Project Goals:

- Develop a comprehensive hive monitoring system that provides real-time data on hive conditions, bee behavior, and environmental factors. Ensure that beekeepers can make informed decisions based on this data.

- Create a user-friendly and intuitive dashboard for beekeepers and environmentalists to access hive data and analytical tools. The interface should be accessible on various devices, including smartphones, tablets, and desktop computers.

- Implement an automated notification system to alert beekeepers about critical hive events, such as temperature or humidity deviations. This system should help prevent hive issues and optimize honey collection.

- (A stretch goal) is to provide specialized tools for environmentalists to study bee populations and their impact on crop pollination, fostering research and insights into the ecological importance of bees.

- Scalability: Ensure that the system can scale to accommodate the needs of small-scale beekeepers, large-scale honey producers, and environmental researchers.

Success Metrics:

Data Collection Frequency: Measure the frequency of data collection from hive sensors.
Baseline: Data collected every 15 minutes.
Target: Increase the frequency to data collected every 5 minutes within six months.

User Adoption Rate: Track the number of registered users and their activity on the platform.
Baseline: 100 users.
Target: Achieve 500 users within the first year.

Notification Response Time: Monitor the time taken by beekeepers to respond to critical notifications.
Baseline: 24 hours.
Target: Reduce response time to less than 4 hours within six months.

System Load Handling: Evaluate the system's performance under heavy loads.
Baseline: 100 hives.
Target: The system should handle data from 1000 hives simultaneously without performance degradation.
User Satisfaction Score: Collect user feedback through surveys or reviews.
Baseline: 3 out of 5 satisfaction score.
Target: Achieve a user satisfaction score of 4 out of 5 within one year.

Pollinator Health Improvement: Measure the positive impact on pollinator health by tracking the bee population growth and colony strength in areas where the system is implemented.
Baseline: No data.
Target: Demonstrate a measurable increase in pollinator health and bee colony strength.

Tracking Method:

- Data collection frequency can be tracked through the system's log files and timestamps of data entries.
- User adoption rate and user satisfaction score can be measured through user registration, activity logs, and feedback surveys.
- Notification response time can be tracked through timestamped notification delivery and user responses.
- System load handling can be evaluated through load testing and performance monitoring.

Open Questions.

My biggest question is regarding what has already been completed. I expect this will inform me on what else needs to be done. I know that this project has gone through several iterations already and will lead to a product being released next year. This implies that much of the backend for the product is already complete. I wonder what else needs to be complete before it can be linked to a front-end system. Also, I wonder if the goal is to link it to a single frontend system, or if it will interface with a family of interface systems. Will web design come into play? Do I need to learn about ap development on a particular platform? How much knowledge should I already have about beekeeping? I am somewhat concerned that I may not have all the skills needed for this project. If that's the case, I want to find out how I can self-learn to get myself up to speed on any topics that I will need to understand so that I can helpfully contribute to the B Ring team.

Here are some specific questions regarding the project:

What is the current status of the project, and what has already been accomplished in terms of development and data collection systems?

With the project having gone through several iterations, what components of the backend are already complete, and what aspects need further development before linking to a front-end system?

Will the product interface with a single front-end system, or is it designed to work with multiple interfaces? How do these interfaces align with the goals and user personas?

Will web design be part of the project scope, and what are the design considerations that need to be taken into account for the user interface?

Do I need to learn app development for a specific platform (e.g., iOS, Android)? If so, what resources and technologies are recommended for gaining these skills?

Will I need to understand beekeeping practices and terminology? What resources are available to self-learn and get up to speed in this domain?

Project Goals: What are the overarching goals for the B Ring project for our specific team.

Out of Scope

Advanced Environmental Tracking: Environmental tracking, particularly extensive and complex environmental analysis, might be considered out of scope for the initial product. The focus could be on basic environmental data collection.

Legacy Platforms: Rolling out the software to older versions of operating systems or less common platforms (beyond iOS, Android, and Windows) might not be necessary initially. The project can prioritize modern and widely-used platforms.

Scalability: The initial versions may not emphasize scalability. However, data collected during usage can help identify potential scalability challenges, which can be addressed in later project phases.

Integration with Numerous Front-End Systems: The project might initially target integration with a specific front-end system or a limited set of interfaces rather than a broad range of front-end solutions.

Advanced User Authentication and Authorization: Complex user management systems, including extensive role-based access control, may not be part of the initial scope.