SHARPFRIDGE : Intelligent Refrigerator System

SHARPFRIDGE is a smart refrigerator system that increases the freshness and durability of food items by providing optimal storage conditions. This system aims to reduce food waste and minimize environmental impact by providing real-time monitoring and alerts.

OBJECTIVES

REAL-TIME ALERTS AND NOTIFICATIONS: IMPLEMENT A SYSTEM TO INFORM USERS OF ANY DEVIATIONS FROM OPTIMAL STORAGE CONDITIONS.

FRESHNESS AND LONGEVITY: MAİNTAİN OPTİMAL TEMPERATURE AND HUMIDITY LEVELS TO PRESERVE THE FRESHNESSAND EXTEND THE SHELF LIFE OF STORED FOOD.

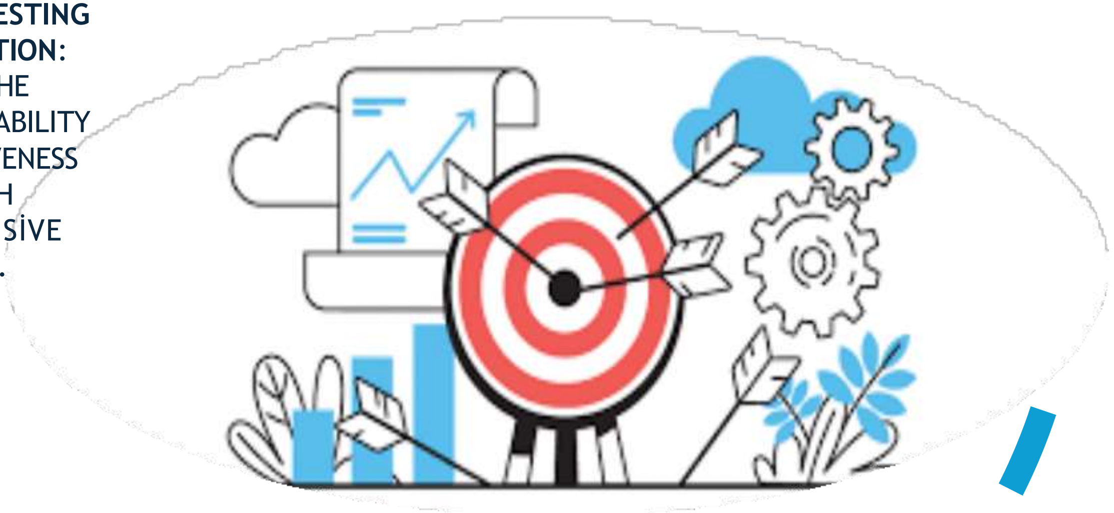
USER-FRIENDLY INTERFACE: PROVIDE AN INTUITIVE INTERFACE FOR EASY MONITORING AND CONTROL OF THE REFRIGERATOR SYSTEM.

THOROUGH TESTING AND VALIDATION: ENSURE THE SYSTEM'S RELIABILITY AND EFFECTIVENESS THROUGH COMPREHENSİVE TESTING.

MULTI-USER ACCESS: ALLOW MULTIPLE USERS TO ACCESS AND MANAGE THE SYSTEM SIMULTANEOUSLY.

ACCURATE STORAGE PERİOD DETERMİNATİON: DETERMINE STORAGE PERIODS BASED ON THE FOOD PRODUCTİON AND CONSUMPTION DATES ENTERED BY USERS.

CUSTOMIZABLE COMPARTMENTS: ENABLE USERS TO SET DIFFERENT COMPARTMENTS WITH ADJUSTABLE TEMPERATURE AND HUMIDITY LEVELS TAILORED TO EACH TYPE OF FOOD



Functional Requirements

- System shall allow users to monitor temperature, humidity , storage levels and status of food with using sensor.

- System shall notify the user about optimal storage conditions via the user-friendly interface.

- Alerts shall give information about the affected compartment and the deviation in the values that should be.

- The system shall adjust cooling and humidity levels based on the type of food that stored and environmental conditions.

- The cooling and humidity adjustments shall be automatically by the system without user intervention.

- The users shall be able to easily adjust settings for each different compartments and food types.

- The system shall undergo rigorous testing to ensure reliability and effectiveness in maintaining optimal storage conditions.

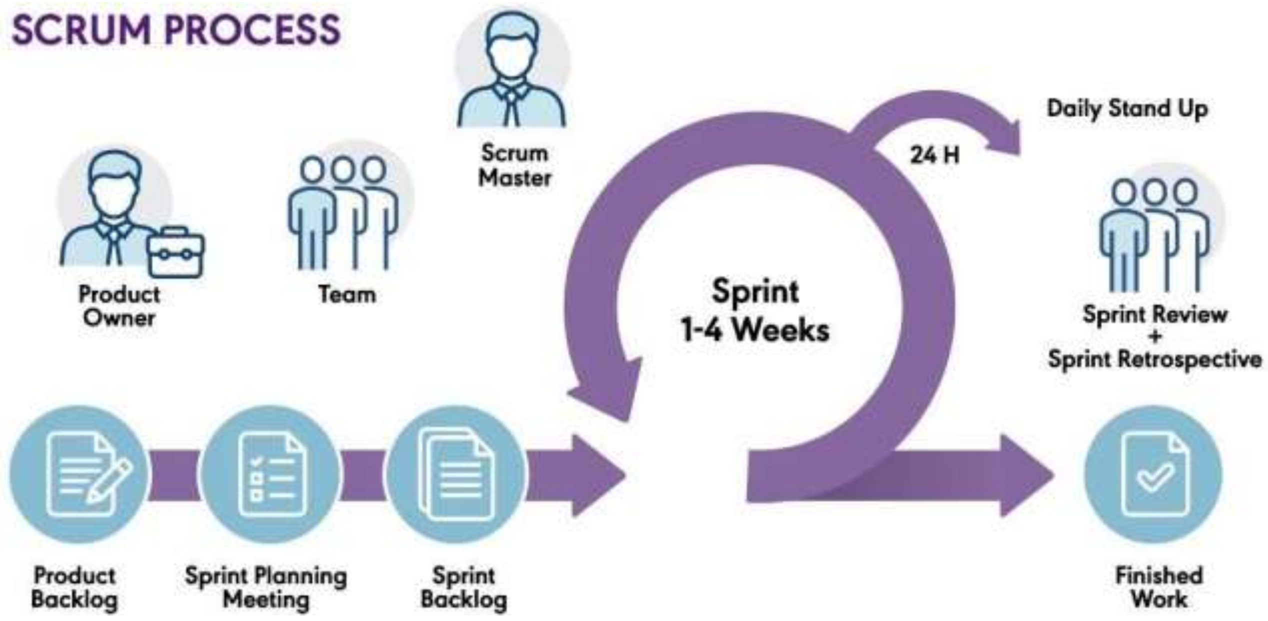
- The system should recalculate humidity and temperatures after a power outage.

Non-Functional Requirements

- The system shall operate reliably under varying environmental conditions.
- The system shall have fail-safe mechanisms to prevent spoilage in case of sensor failure or power outage.
- User data, including login credentials and food storage information, shall be encrypted and securely stored.
- Access to sensitive features such as temperature adjustment shall be restricted to authorized users.
- The system architecture shall support scalability to accommodate additional sensors or features in future upgrades.
- Instructions and help resources shall be available within the interface for user assistance.
- The system shall be compatible with a wide range of food storage containers and packaging materials.
- The user interface shall be intuitive and easy to navigate, requiring minimal training for users.

Software Process Model

- **Reason to Choose This Model**
  - In this application hardware is necessary, creating prototypes is essential. Scrum enables us to develop and test prototypes.
  - SharpFridge has both software and hardware development. Scrum supports -functional teams to work together effectively.
  - Scrum makes customers actively involved throughout the project, providing feedback at every stage.



- Scrum minimizes these costs by promoting , feedback regularly, allowing the project to adapt to technological advances and changes. It includes daily standup meetings and regular sprint reviews.
- They are essential in communication.

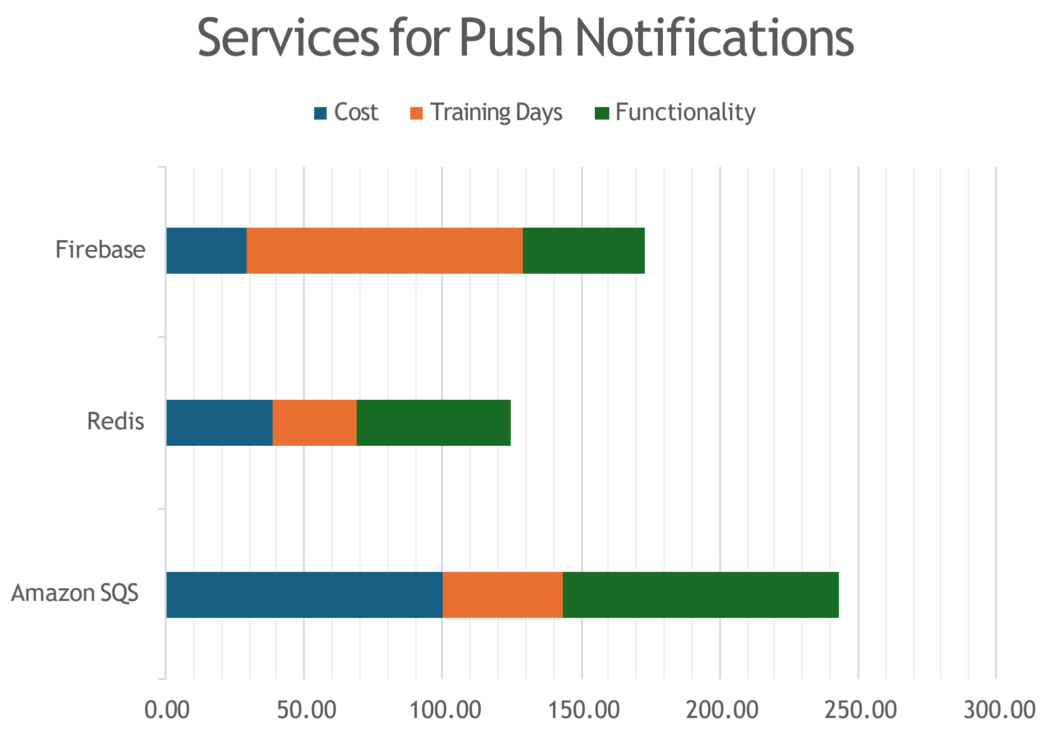
MEASUREMENTS

- Effort: Total person hours spent on development and testing.
- Schedule Compliance: Compared to the Gantt chart and predicted resources.
- Code Reuse: Percentage of automated or finished tasks via pre-written code or libraries. (E.g. using libnet in order to send packages)
- Number of Changes: Rate of changes made and ongoing changes.
- Product Quality: Combination of defects(mismatches),errors and bugs.
- Testing effort: Time, budget and workforce allocated to test the components.
- Product Size: Number of components, required documentation size, training material, embedded software size in bytes.
- Data management: Cost and length of completed sprints/Estimated amount for all sprints.

Additional Software Tools: Notification Services

Tool	Amazon SQS	Redis	Firebase
Cost	$1,552.42/mo	$600/mo	$450/mo
Training Days	10	7	23
Functionality	90	50	40

Tool	Amazon SQS	Redis	Firebase
Cost	**100**	38.7	29.0
Training Days	43.5	30.4	**100**
Functionality	**100**	55.5	44.4



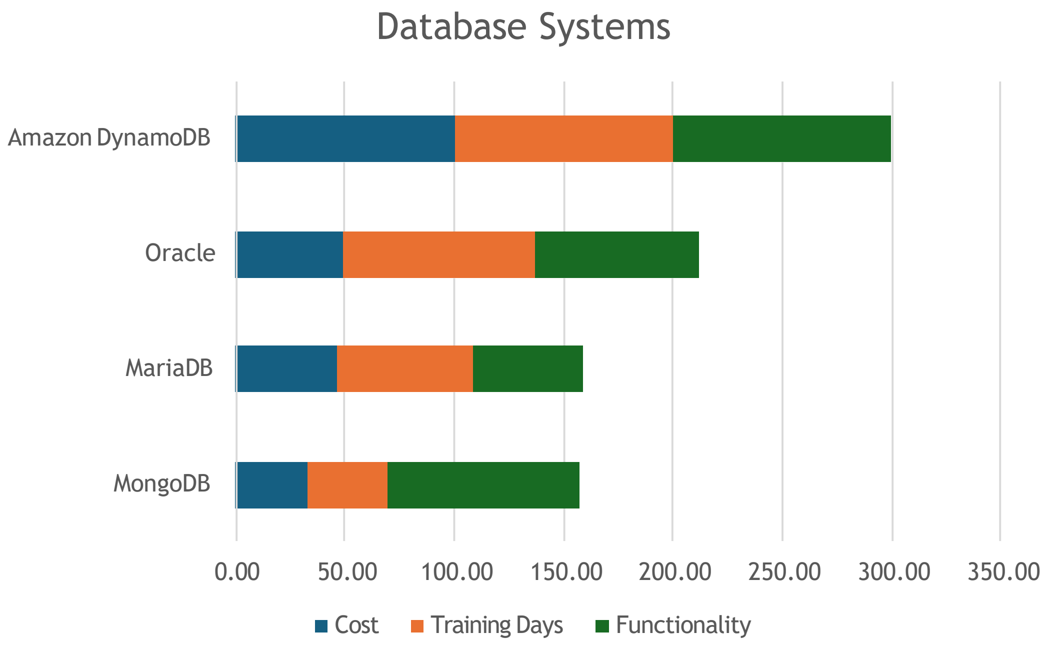
Amazon SQS is selected due to

- Integration with DynamoDB
- Integration with Django Celery
- Flexible and robust

Additional Software Tools: Database

Tool	MongoDB	MariaDB	Oracle	DynamoDB
Cost	$300/Mo	$430/Mo	$460/Mo	$930/Mo
Training Days	3	5	7	8
Functionality	70	40	60	80

Tool	MongoDB	MariaDB	Oracle	DynamoDB
Cost	32.2	46.2	49.4	**100**
Training Days	37.5	62.5	87.5	**100**
Functionality	87.5	50	75	**100**



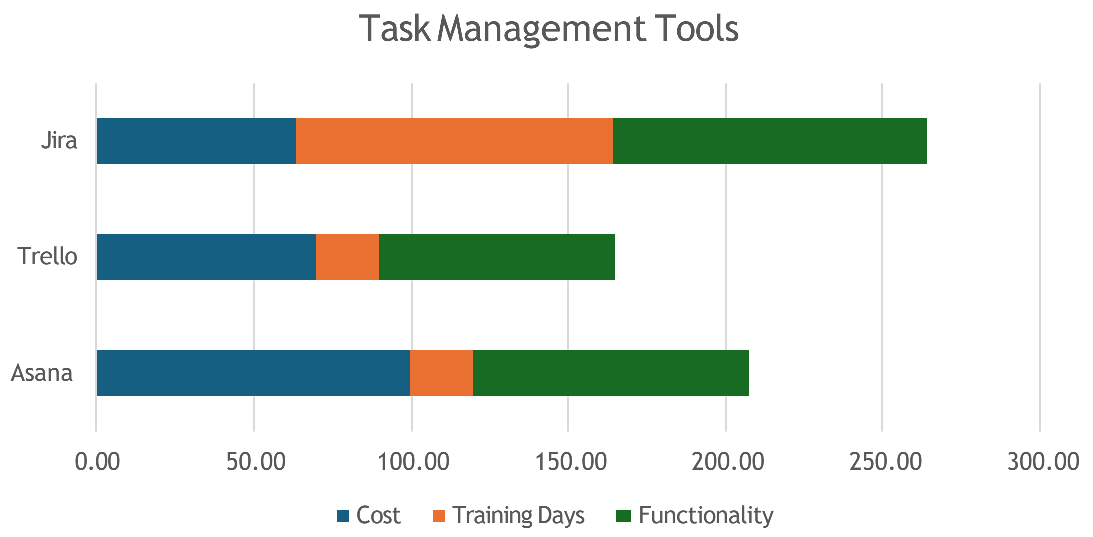
Amazon DynamoDB is selected due to

- Best performance in high traffic
- Globally distributed servers

Additional Software Tools: Project Management

Tool	Asana	Trello	Jira
Cost	$250/Mo	$175/Mo	$160/Mo
Training Days	1	1	5
Functionality	70	60	80

Tool	Asana	Trello	Jira
Cost	**100**	70	64
Training Days	20	20	**100**
Functionality	87.5	75	**100**



Jira is selected due to

- Low cost
- Agile development support

Programming Languages

React Native

- Axios

{ api }

Django

- Celery (For SQS connection)
- Boto3 (For DynamoDB connection)

Embedded C++

- Connected to the internet

- STAKEHOLDERS

#	STAKEHOLDERS	DESCRIPTION
1	**Chain Restaurant Owners/Managers**	They are the primary stakeholders as they are directly impacted by the inefficiencies of traditional refrigeration systems economically.
2	**Restaurant and Kitchen Staff**	They interact directly with the refrigerator system on a daily basis. They are the group that benefits most from the system's interface and functionality to food storage and retrieval processes.
3	**Customers of the Restaurant**	Indirectly impacted by the quality of food served at restaurants. Because of the freshness and quality of the food items stored in the refrigerator system affects their dining experience, they are also stakeholders.
4	**Environmental Conservation Organizations**	They are concerned with reducing food waste and minimizing environmental impact. The effectiveness of the intelligent refrigerator system help to reduce food waste and promote sustainable practices.
5	**Suppliers/Vendors**	Since they are provide food items to chain restaurants, the food items they supply must be stored properly and maintain their quality. Therefore, it minimizes losses for both parties.

#	STAKEHOLDERS	DESCRIPTION
6	**Regulatory Authorities**	Their role involves guaranteeing adherence to food safety and storage guidelines, while smart refrigerator systems aid chains in ensuring food safety.
7	**Technology Providers**	Firms specializing in sensor technology generate revenue through the creation of software development tools and other technical elements. The enhanced value of their technology occurs with the effective integration and widespread acceptance within intelligent refrigerator systems.
8	**Competitors**	Competitors encompass a range of firms working on similar intelligent refrigeration systems or alternative solutions to tackle inefficiencies in food storage, including developers of conventional refrigerators. The responses and strategies of SharpFridge could potentially steer the course and marketing approaches of these other companies.
9	**Developers of SharpFridge**	As they allocate their workforce and budget towards this project, the sales performance and maintenance of SharpFridge can significantly influence their branding and contribute to the expansion of their company.

RISKS

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISKS
1	1	2	Security Breaches
5	2	7	Hardware Supply Chain Disruptions
2	7	9	Inaccurate Sensor Readings
7	3	10	Technical Failures
6	4	10	Regulatory Changes
3	8	11	Compatibility Issues
4	9	13	Insufficient User Training
8	5	13	Competition
10	6	16	Budget Constraints
6	10	16	Consumer Shifts

SOFTWARE NEEDS

- Operating System
- Application Software
- Networking Software
- Firmware
- Analytics and Reporting Software
- Integration Software

HARDWARE NEEDS

- Sensors
- Cooling System
- User Interface
- Communication Equipment
- Supporting Equipment
- Data Storage
- Environmental Monitoring Equipment
- Power Management
- Physical Infrastructure
- Safety Features

SUPPORT NEEDS

- Technical Support
- Procurement Support
- IT Support
- Training and Education
- Maintenance and Service
- Regulatory Compliance Support

Project Schedule

Project Planning and Controlling

- Duration : 22 Days
- Project Scope Definiton
- Stakeholder Analysis
- Budget Management
- Risk Assessment
- Creating Project Schedule

Critical activities: defining project scope, stakeholder analysis, budget management, risk assessment, and creating the project schedule.

Hardware

- Duration : 14 Days
- Identifying Extra Requirements
- Vendor Proposal Analysis
- Evaluation and Vendor Selection
- Procurement

We focus on identifying extra requirements, analyzing vendor proposals, and conducting procurement. These steps are vital to ensure that we have the right components for the project's hardware needs.

Sensor Integration

- Duration : 28 Days
- Sensor Quality Selection
- Physical Installation
- Sensor Prototyping
- Developing Low-Level Software
- Integration with Low-Level Software

Extensive phase ensuring sensor functionality and integration.

Tasks: sensor quality selection, physical installation, prototyping, developing and integrating low-level software.

User App Design

- Duration : 14 Days
- UX Design
- Backend Development
- Visual Design
- Prototyping
- Usability Testing

Focus on creating a user-friendly application.

Includes UX design, backend development, visual design, prototyping, and usability testing.

Testing and Validation

- Duration : 24 Days
- Test Planning
- Test Execution
- Defect Tracking
- Validation
- User Acceptance Testing

Ensuring product quality and reliability.

Involves test planning, execution, defect tracking, validation, and user acceptance testing.

Documentation and Deployment

- Duration :

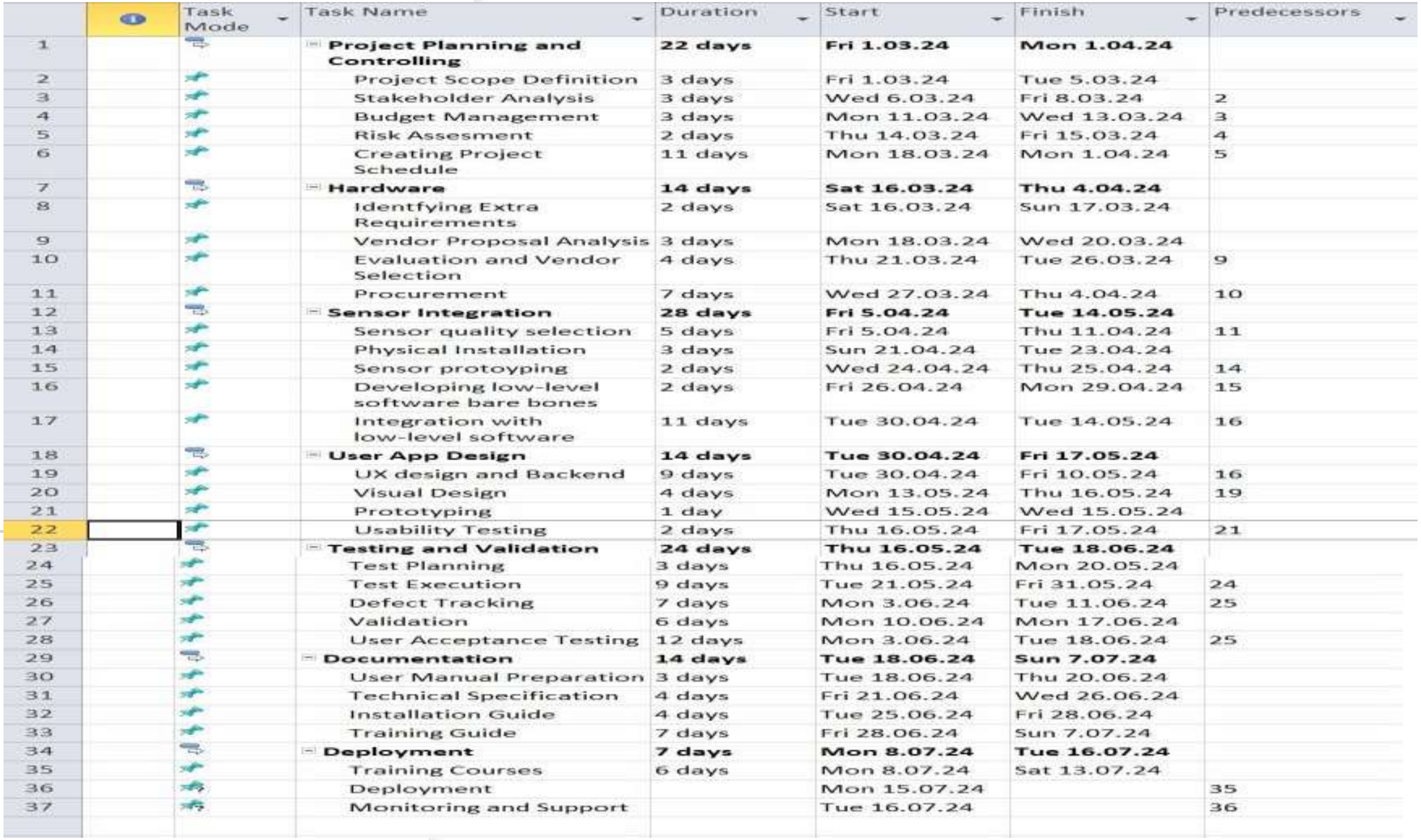
  14 Days(documentation)

  7 Days(deployment)
- Training Guide
- Training Courses
- Deployment
- Monitoring and Support

Documentation: user manuals, technical specifications, installation and training guides.

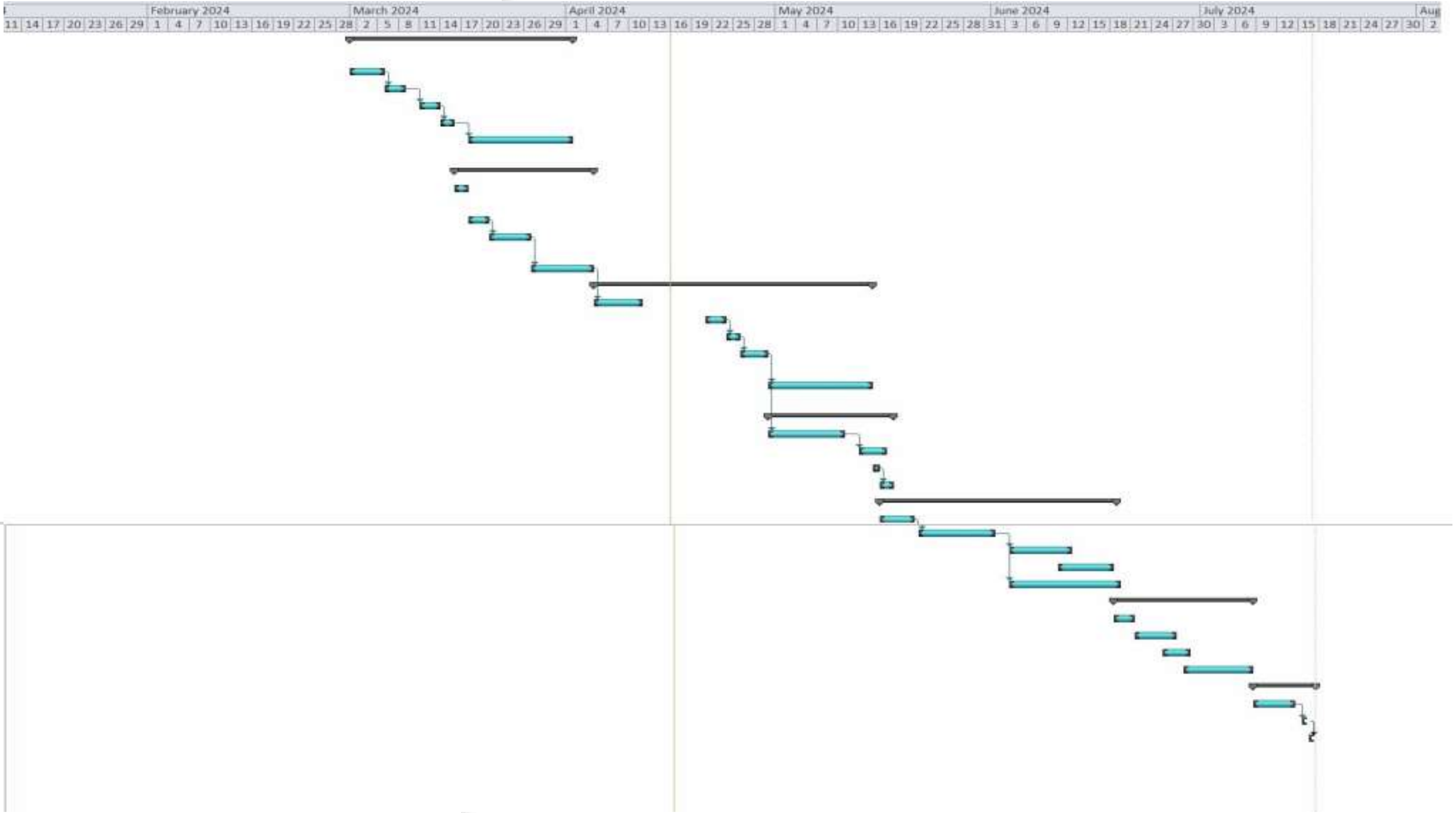
Deployment: training courses, software deployment, monitoring, and support.

Gantt Chart and Schedule



	ⓘ	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1			**Project Planning and Controlling**	**22 days**	**Fri 1.03.24**	**Mon 1.04.24**	
2			Project Scope Definition	3 days	Fri 1.03.24	Tue 5.03.24	
3			Stakeholder Analysis	3 days	Wed 6.03.24	Fri 8.03.24	2
4			Budget Management	3 days	Mon 11.03.24	Wed 13.03.24	3
5			Risk Assesment	2 days	Thu 14.03.24	Fri 15.03.24	4
6			Creating Project Schedule	11 days	Mon 18.03.24	Mon 1.04.24	5
7			**Hardware**	**14 days**	**Sat 16.03.24**	**Thu 4.04.24**	
8			Identfying Extra Requirements	2 days	Sat 16.03.24	Sun 17.03.24	
9			Vendor Proposal Analysis	3 days	Mon 18.03.24	Wed 20.03.24	
10			Evaluation and Vendor Selection	4 days	Thu 21.03.24	Tue 26.03.24	9
11			Procurement	7 days	Wed 27.03.24	Thu 4.04.24	10
12			**Sensor Integration**	**28 days**	**Fri 5.04.24**	**Tue 14.05.24**	
13			Sensor quality selection	5 days	Fri 5.04.24	Thu 11.04.24	11
14			Physical Installation	3 days	Sun 21.04.24	Tue 23.04.24	
15			Sensor protoyping	2 days	Wed 24.04.24	Thu 25.04.24	14
16			Developing low-level software bare bones	2 days	Fri 26.04.24	Mon 29.04.24	15
17			Integration with low-level software	11 days	Tue 30.04.24	Tue 14.05.24	16
18			**User App Design**	**14 days**	**Tue 30.04.24**	**Fri 17.05.24**	
19			UX design and Backend	9 days	Tue 30.04.24	Fri 10.05.24	16
20			Visual Design	4 days	Mon 13.05.24	Thu 16.05.24	19
21			Prototyping	1 day	Wed 15.05.24	Wed 15.05.24	
22			Usability Testing	2 days	Thu 16.05.24	Fri 17.05.24	21
23			**Testing and Validation**	**24 days**	**Thu 16.05.24**	**Tue 18.06.24**	
24			Test Planning	3 days	Thu 16.05.24	Mon 20.05.24	
25			Test Execution	9 days	Tue 21.05.24	Fri 31.05.24	24
26			Defect Tracking	7 days	Mon 3.06.24	Tue 11.06.24	25
27			Validation	6 days	Mon 10.06.24	Mon 17.06.24	
28			User Acceptance Testing	12 days	Mon 3.06.24	Tue 18.06.24	25
29			**Documentation**	**14 days**	**Tue 18.06.24**	**Sun 7.07.24**	
30			User Manual Preparation	3 days	Tue 18.06.24	Thu 20.06.24	
31			Technical Specification	4 days	Fri 21.06.24	Wed 26.06.24	
32			Installation Guide	4 days	Tue 25.06.24	Fri 28.06.24	
33			Training Guide	7 days	Fri 28.06.24	Sun 7.07.24	
34			**Deployment**	**7 days**	**Mon 8.07.24**	**Tue 16.07.24**	
35			Training Courses	6 days	Mon 8.07.24	Sat 13.07.24	
36			Deployment		Mon 15.07.24		35
37			Monitoring and Support		Tue 16.07.24		36

Gantt Chart and Schedule


February 2024
March 2024
April 2024
May 2024
June 2024
July 2024
Aug