PROJECT DOCUMENTATION

PROJECT PROPOSAL

Project: Weather Prediction Web App to Optimize Solar Power Plants

Release: February 2024

Date: 07th February 2024

PRINCE2

Author: P.U.Nisansa (Project Manager)

Owner: Dr. Yasas Jayaweera (Project Executive)

Client: Mr. xzy Karunaratne

Version No: 1.0

1 PROJECT PLAN HISTORY

1.1 Document Location

This document is only valid on the day it was printed.

The source of the document will be found on the project's PC in location.

1.2 Revision History

Date of this revision:

Revision date	Previous revision date	Summary of Changes	Changes marked

1.3 Approvals

This document requires the following approvals.

Signed approval forms are filed in the Management section of the project files.

Name	Signature	Title	Date of Issue	Version
Dr. Yasas Jayaweera		Project Supervisor	07/02/2024	1.0
Mr. xyz Karunaratne		CEO – z Energy (Pvt) Ltd.	07/02/2024	1.0
P.U.Nisansa	panagoda	Project manager	07/02/2024	1.0

1.4 Distribution

This document has been distributed to:

Name	Title	Date of Issue	Version
P.U.Nisansa	Project manager	07/02/2024	1.0
x	Startup manager	07/02/2024	1.0
P.U.Nisansa	Risk manager	07/02/2024	1.0
P.U.Nisansa	Quality manager	07/02/2024	1.0
x	Scheduling Manager	07/02/2024	1.0

Table of Contents

LIST OF TABLES

LIST OF FIGURES

2 Introduction

x Energy (Pvt) Ltd is a pioneering company dedicated to addressing the challenge of maintaining consistent power output from solar energy systems for various weather conditions. To achieve this goal, x Energy (Pvt) Ltd needs a sophisticated system capable of dynamically predicting weather patterns and corresponding power output. Team Unicoders proposes the development of a dynamic and user-friendly website for x Energy (Pvt) Ltd. The system aims to provide predicted weather data and power output estimates for upcoming days, enabling informed decision-making and efficient management of solar energy systems.

2.1 BACKGROUND OF THE CLIENT/ PROJECT

xyz Karunaratne leads x Energy (Pvt) Ltd, a company specializing in solar energy solutions. They face challenges due to unpredictable weather, which impacts their power output. To address this issue, they require a system that predicts weather using an API and simplifies decision-making for their solar machines.

x Energy has partnered with Team Unicoders to develop a user-friendly website. This platform will fetch weather predictions and inform the solar machines accordingly. For example, if tomorrow's forecast is rainy, the system will notify the machines to reduce power output. Conversely, if it's sunny, the machines will prepare for increased output. The website will also display power output estimates for each day, facilitating efficient planning and operation of the solar systems.

2.2 PROBLEM STATEMENT

In the field of renewable energy nowadays, the primary challenge revolves around the inconsistency of power output from solar panels, primarily due to unpredictable weather conditions. This variability in sunlight availability disrupts continuous power generation and poses significant risks to the integrity and functionality of associated batteries and system components. Such irregular power output undermines the reliability of energy supply and compromises the overall effectiveness and sustainability of solar energy solutions.

To address this issue, x Energy (Pvt) Ltd recognizes the critical requirement for innovative technologies capable of mitigating the negative effects of weather fluctuations on solar energy systems. While the company is pioneering the development of advanced super

batteries to enhance energy storage capacities, the success of these solutions hinges upon the ability to accurately anticipate and adapt to changing weather patterns. Therefore, the creation of a comprehensive system that accurately forecasts weather conditions and dynamically adjusts power output accordingly is essential for maximizing the performance and durability of solar energy systems.

2.3 NEEDS STATEMENT

Since the client has requested a way to predict the way with the help of API and a user-friendly UI which is showing the power output and a graph, we have decided to create a web-based system for x Energy (Pvt) Ltd.

2.4 SOLUTION AND OBJECTIVES

To address the challenge of maintaining consistent power output from solar energy systems between varying weather conditions, x Energy (Pvt) Ltd, in collaboration with Team Unicoders, proposes a comprehensive solution using an advanced weather prediction and power output optimization.

1. Integration of Google Graphcast for Weather Predictions:
    - With the use of Google Graphcast's predictive powers, the suggested system will use weather data as input to produce forecasts for the upcoming 10 days.
    - The system's integration of accurate weather forecasts will facilitate proactive decision-making and the enhancement of solar energy systems by taking advantage of predicted weather patterns.
2. Algorithm Development for Power Output Calculation:
    - Team Unicoders will receive unique methods from x Energy (Pvt) Ltd so they can precisely determine electricity output according on the current weather.
    - These algorithms will dynamically modify power production in real-time by considering variables like temperature, cloud cover, and sunshine intensity.

Main objectives of this project as follows,

- Provide Accurate Weather Predictions
- Optimize Power Output
- Enhance System Resilience
- Enable Informed Decision-Making

3 PROPOSED TECHNICAL APPROACH

The proposed technical approach focuses on integrating Google Graphcast for accurate weather predictions and collaborating with x Energy (Pvt) Ltd to develop power output calculation algorithms. Through this collaboration, we aim to dynamically adjust power output based on forecasted weather conditions. Additionally, we prioritize user experience by designing an intuitive interface and implementing robust security measures. Rigorous testing and ongoing maintenance ensure system reliability and effectiveness. To facilitate efficient development and adaptation to evolving requirements, the agile software development methodology was employed in the creation of this application. Overall, our approach aims to optimize energy generation, enhance system resilience, and provide users with actionable insights for efficient management of solar energy resources.

3.1 DEVELOPMENT METHODOLOGY

We have chosen to adopt the Agile scrum software development methodology for the development of the Weather Prediction Web App to Optimize Solar Power Plants system. Agile scrum methodology is well-suited for our project due to its emphasis on rapid development through iterative cycles known as sprints. Each sprint is dedicated to delivering a set of prioritized features and functionality, allowing for quick releases and continuous improvement. This approach enables us to respond quickly to changing requirements and market demands, while also facilitating ongoing feedback and collaboration with stakeholders. By embracing Agile principles, we aim to ensure the successful planning, control, and delivery of our project, ultimately leading to the creation of a robust and efficient solution for optimizing solar power plants.

Why we choose agile scrum development methodology?

- Rapid Iterative Development
- Customer-Centric Approach
- Flexibility and Adaptability
- Continuous Improvement and Quality Assurance
- Efficient Team Collaboration
- Risk Management

Overall, by embracing Agile scrum principles, we aim to maximize efficiency, transparency, and customer satisfaction throughout the development process.

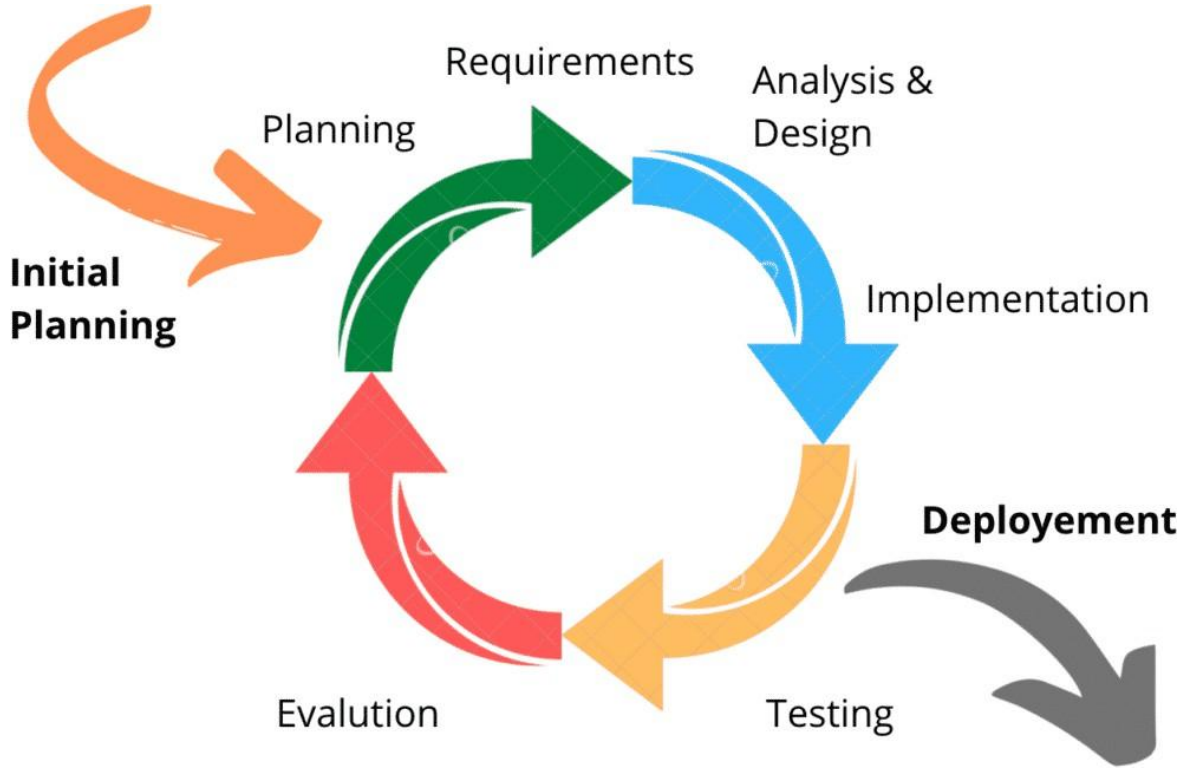

Figure 3:1 Agile methodology.

Planning: The planning phase emphasizes close collaboration with the client. The project manager and team thoroughly assessed the project's potential, considering objectives, scope, goals, and research questions to ensure comprehensive management.

Design: During this stage, the system's design is developed, focusing on creating a user-friendly interface for the website. Elements such as branding, layout, navigation, and presentation are carefully considered to enhance user experience and align with project goals.

Implementation: In this phase, the development team brings the design concepts to life by implementing the website. Frontend and backend components are developed, database integration is undertaken, and necessary APIs are implemented to ensure seamless functionality.

Testing: Testing plays a crucial role in ensuring the quality and reliability of the website. Various levels of testing, including functional testing, are conducted to verify website features and functionality, ensuring a smooth user experience and overall performance.

3.2 REQUIREMENT GATHERING

Requirements were gathered from the client through a physical meeting, facilitating direct communication and comprehensive understanding of the client's needs. During this meeting, the project team engaged in open dialogue with the client, asking relevant questions to gather and clarify requirements. By actively listening to the client's feedback and concerns, the team ensured a thorough understanding of the project objectives, expectations, and specifications. Through this collaborative process, the team obtained valuable insights into the client's preferences, priorities, and desired outcomes, laying the foundation for the successful execution of the project.

3.3 ARCHITECTURE DIAGRAM

The architecture diagram serves as a visual representation of the structural design and components comprising the Weather Prediction Web App to Optimize Solar Power Plants system. This diagram illustrates the organization and interactions of various elements within the system, providing a comprehensive overview of its functionality and infrastructure. By delineating the relationships between frontend and backend components, external APIs, data processing algorithms, and deployment environments, the architecture diagram offers valuable insights into the system's design and operation. It serves as a foundational tool for understanding, communicating, and refining the system architecture throughout the development lifecycle.

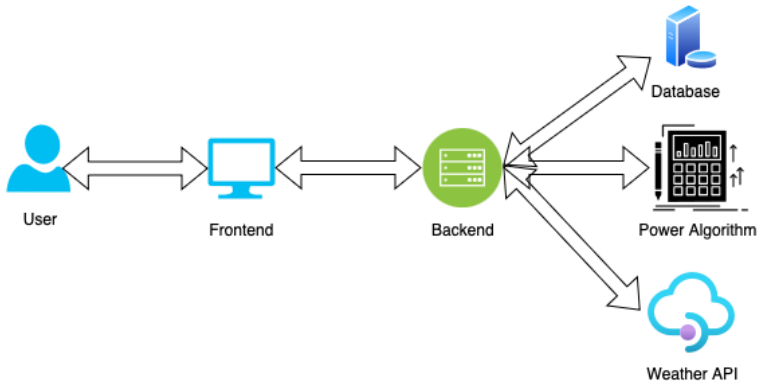

Figure 3:2 Architecture diagram.

3.4 FUNCTIONAL REQUIREMENTS

1. **User Authentication**: The system must support user authentication to ensure secure access to the platform.
2. **Weather Data Integration**: The system should integrate with external weather data sources to retrieve current and forecasted weather information. This data will be used for predicting weather patterns and optimizing power output.
3. **Power Output Prediction**: Based on weather data and other relevant factors, the system should predict the expected power output of solar power plants for upcoming days. This prediction will enable users to make informed decisions regarding energy management.
4. **Dashboard**: The system should provide a user-friendly dashboard where user can view weather forecasts, predicted power output, historical data, and other relevant information. The dashboard should be customizable and display information in a clear and intuitive manner.
5. **Security and Access Control**: The system must enforce robust security measures to protect sensitive data and ensure authorized access only. Role-based access control should be implemented to manage user permissions and privileges effectively.

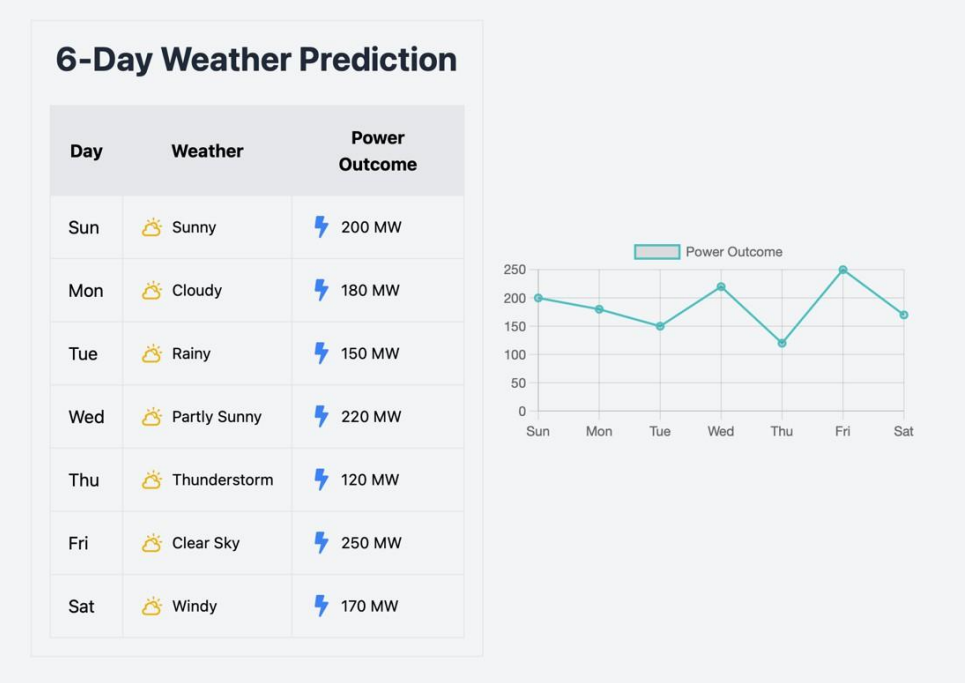

Figure 3:3 User dashboard.

3.5 NON-FUNCTIONAL REQUIREMENTS

- **Performance**: The system should be responsive and able to handle concurrent user requests efficiently, ensuring minimal latency and downtime.
- **Availability**: The system is available 24/7 with limited downtime.
- **Security**: The system must adhere to industry-standard security practices to safeguard sensitive data and protect against unauthorized access, data breaches, and cyber-attacks.
- **Usability**: The system should be intuitive and easy to use, requiring minimal training

3.6 IMPLEMENTATION AND DEVELOPMENT REQUIREMENTS

Software requirements:

- MongoDB Atlas
- Visual Studio Code
- Libraries- TailwindCSS, Vite-Server
- Microsoft word
- Microsoft PowerPoint
- Windows 7/MAC OS X or above

Hardware requirements:

- PC
- WIFI router
- CPU i3 or above
- RAM-4GB or above
- Disk-20GB

3.7 RUNNING ENVIRONMENT REQUIREMENTS

Recommended requirements:

- Windows 10/Mac OS X Desktop
- 2 GHz or above processor
- 32/64 – bit processor
- Disk-2GB
- Web browser

3.8 QUALITY ASSURANCE PLAN

The Quality Assurance (QA) plan for the Weather Prediction Web App to Optimize Solar Power Plants system ensures top-notch quality, reliability, and performance. QA Manager strictly tests every aspect, from unit to system level, using industry-standard tools and practices. Bugs are tracked and promptly addressed, with continuous integration/deployment pipelines automating processes for efficiency. Comprehensive documentation and training sessions ensure clarity and competence among team members. Regular feedback loops drive continuous improvement, guaranteeing compliance with all regulatory and organizational standards. Through this QA plan, we're committed to delivering a high-quality, dependable, and user-friendly solution.

Objective - Ensuring functionality, usability, performance, and security of the Weather Prediction Web App.

Test Strategy: The testing approach for the Weather Prediction Web App to Optimize Solar Power Plants system will encompass various test types to ensure comprehensive coverage and quality assurance. The scope of testing will include functional, usability, performance, and security aspects of the system.

Test types:

- Functional testing – Verify that all the features work as expected
- Usability testing – Evaluate user interface and user experience and chatbot
- feature to ensure that they are user friendly and pleasing
- Performance testing – Test the website and chatbot’s response time
- Security testing – Identify and address potential security vulnerabilities

4 EXPECTED PROJECT RESULTS

The expected project results encompass the development of a fully functional Weather Prediction Web App, providing accurate forecasts and optimizing solar power plant performance. Users will enjoy an enhanced experience through an intuitive interface, ensuring efficient energy management. The system will demonstrate optimal performance, even under high traffic, and robust security measures will safeguard user data. Deliverables will meet industry standards, saving positive user feedback and increasing efficiency. Successful deployment will ensure accessibility and reliability, aligning with defined requirements and fulfilling project objectives. Overall, these results promise real benefits, contributing to the application's success and user satisfaction. The outcomes are,

- Development of a fully functional Weather Prediction Web App
- Accurate forecasting capabilities for optimal solar power plant performance
- Improved user experience through an intuitive interface
- Successful deployment ensuring accessibility and reliability
- Alignment with predefined requirements and project objectives to meet Client needs

4.1 DELIVERABLES

- Project brief
- Project proposal
- Software requirement specification report
- Prototypes
- Progress review
- Final product

4.2 MEASURES OF SUCCESS

- **Accurate Weather Forecasts**: Achieving a high level of accuracy in predicting weather conditions for the specified timeframe, ideally with a low margin of error.
- **Power Output Optimization**: Demonstrating an improvement in solar power plant performance based on forecasted weather conditions, resulting in increased energy production or efficiency.
- **Client Satisfaction**: Receiving positive feedback from Client regarding the usability, functionality, and reliability of the Weather Prediction Web App.

- **System Performance**: Maintaining optimal system performance, including fast response times and minimal downtime, even during peak usage periods.
- **Successful Deployment**: Completing the deployment of the Weather Prediction Web App without major issues or disruptions, ensuring accessibility and reliability for client.

5 BUDGET

Table 5-1 Budget of the project

Expenses	**Description**	**Basis**	**Unit**	**Rate**	**Total (LKR)**
Planning					
Meetings (Google meet)	Internet charges	Hourly	4	200	800
Development					
Designing UIs	Design UIs for the web application	Hourly	12	800	9600
Website implementation	Implement function according to the designed UIs	Hourly	45	800	36000
Database design & implementation	Design and develop the database and store user data secured	Hourly	12	700	8400
Testing					
QA testing	Check the quality of the application and bug fixing	Hourly	6	500	3000
Documentation					
Preparation of documentation	Prepare required documents for the project	All documents	N/A	N/A	5700
Setting Up					
Domain	Domain name for the web application	Per year	1	5000	5000
Hosting	Hosting the web application and the database	Per year	1	16000	16000
Maintenance	N/A	Monthly	12	3000	36000
Total					**120,500**

6 ROLES AND RESPONSIBILITIES

Table 6-1 Roles and responsibilities of the project

ROLE	NAME	RESPONSIBILITY
Client	Mr. xyz Karunaratne	Client of the Project
Project Manager	P.U.Nisansa	Requirement Gathering. Analyzing. Communicating with the Client. Directing the team and updating them. Managing the team, resources, schedules, and deliverables
Start-up manager	x	Ensuring the System's effective launch and initial operations. Strategic planning, acquiring resources, adhering to the law, establishing a team, managing finances, supervising operations, marketing, and performance evaluation.
Risk manager	P.U.Nisansa	Locating, evaluating, and reducing risks. Identify potential risks, assessing their possibility and impact, creating mitigation measures, keeping an eye on risk indicators.
Quality manager	P.U.Nisansa	Quality assurance of the final product. Setting quality goals, performing tests and inspections.
Scheduling manager	x	Developing and overseeing the project timeline. Planning project timetables, keeping track of progress, modifying schedules as necessary, working with team members, and communicating routine schedule updates to stakeholders

7 SCHEDULE

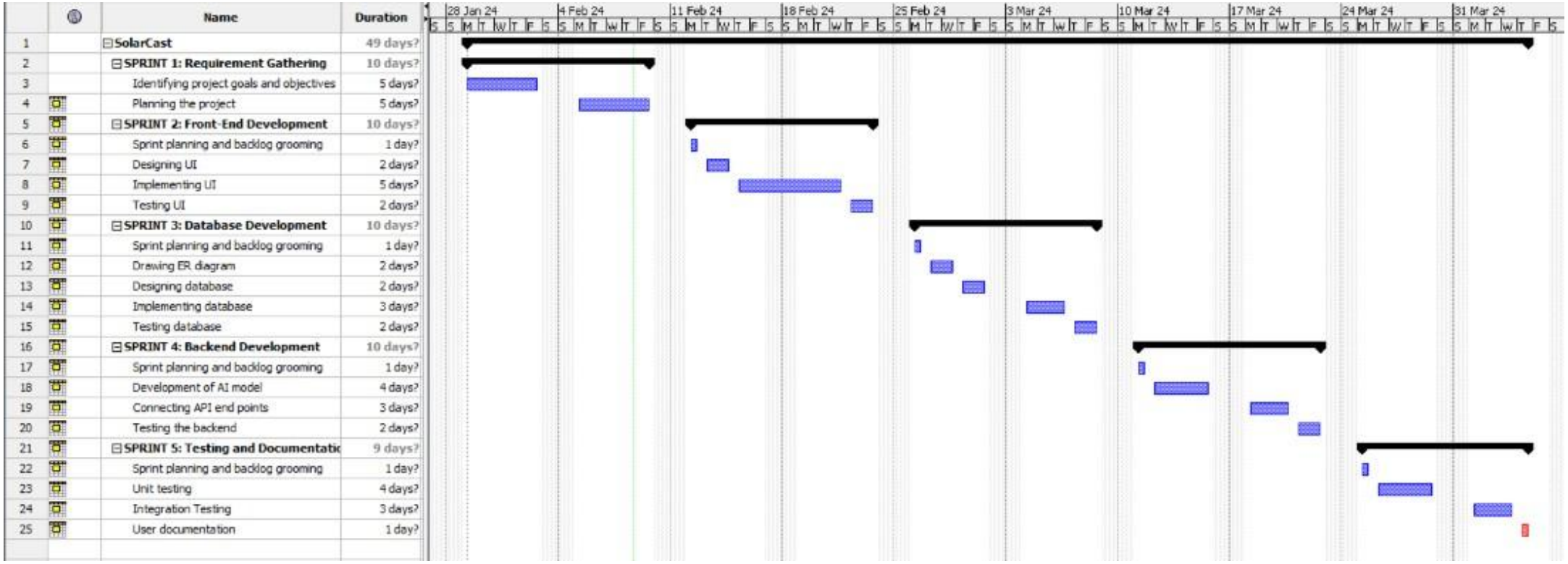

Figure 6:1 Gantt chart

SolarCast – Task List

Name	Start Date	End Date
SPRINT 1: Requirement Gathering	**2G/1/24**	**G/2/24**
• Identifying project goals and objectives	29/1//24	2/2/24
• Planning the project	5/2/24	9/2/24
SPRINT 2: Front-End Development	**12/2/24**	**23/2/24**
• Sprint planning and backlog grooming	12/2/24	12/2//24
• Designing UI	13/2/24	14/2/24
• Implementing UI	15/2//24	21/2/24
• Testing UI	22/2/24	23/2/24
SPRINT 3: Database Development	**26/2/24**	**8/3/24**
• Sprint planning and backlog grooming	26/2/24	26/2/24
• Drawing ER diagram	27/2/24	28/2/24
• Designing database	29/2/24	3/1/24
• Implementing database	4/3//24	6/3/24
• Testing database	7/3/24	8/3/24
SPRINT 4: Backend Development	**11/3/24**	**22/3/24**
• Sprint planning and backlog grooming	11/3/24	11/3/24
• Development of AI model	12/3/24	15/3/24
• Connecting API end points	18/3/24	20/3/24
• Testing the backend	21/3/24	22/3/24
SPRINT 5: Testing and Documentation	**25/3/24**	**4/4/24**
• Sprint planning and backlog grooming	25/3/24	25/3/24
• Unit testing	26/3/24	29/3/24
• Integration Testing	1/4/24	3/4/24
• User documentation	4/4/24	4/4/24

Figure 6:2 Task list

8 REFERENCES

javatpoint. (n.d.). *software-engineering-agile-model*. Retrieved from javatpoint: www.javatpoint.com