**Classification of Fresh and Rotten Fruits using Computer Vision**

**Software Requirements Specification**

Version 1.0



**Group Id: S2402F4B7E (BC200418293 & MC220201538)**

**Supervisor Name: *Zaid Ismail***

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 20/05/2024 | 1.0 | A Software Requirements Specification (SRS) is a comprehensive document that outlines the detailed requirements and specifications of a software project. It is considered as a foundation or base of the project as it consist of Scope, Functional and Non Functional Requirement, Use Case Diagram, Usage Scenarios, Adopted Methodology and Work Plan of the Project. This SRS document provides a clear and unambiguous description of the field of computer vision. In this SRS we will learn that the main objective of the project is to develop an AI-Based system capable of differentiating between fresh and rotten fruits. | **(BC200418293 & MC220201538)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project) 4](#scope)
2. [Functional Requirements Non Functional requirements 4](#FRNFR)
3. [Use Case Diagram 6](#UCD)
4. [Usage Scenarios 7](#UCS)
5. [Adopted Methodology](#Adopted) 10
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#workplan) 14

**SRS Document**



**Scope of Project:**

The project aims to develop an automated system leveraging computer vision and deep learning techniques to classify fresh and rotten fruits effectively. The scope encompasses several integral components, starting with image preprocessing to ensure optimal compatibility and processing efficiency. Through the implementation of Convolutional Neural Networks (CNNs), the system will analyze visual characteristics extracted from a labeled dataset of fruit images, enabling accurate differentiation between fresh and spoiled fruits. A user-friendly interface will facilitate seamless interaction, allowing users to upload test images and receive classification results promptly, along with confidence scores. Performance evaluation will be conducted using metrics such as accuracy, precision, recall, and F1-score, providing comprehensive insights into the system's classification capabilities. Documentation will play a crucial role in outlining the system architecture, implementation details, and user guidelines, ensuring clarity and transparency. Additionally, comprehensive reports summarizing the performance evaluation results will offer valuable insights and facilitate informed decision-making. Ultimately, the project aims to provide a robust and efficient solution that empowers farmers and stakeholders to maintain agricultural productivity while minimizing food waste through timely identification of fresh and rotten fruits.

**Functional and non Functional Requirements:**

**Functional Requirements:**

Functional requirements specify what the software system should do, describing its specific functionalities, features, and behaviors. Functional requirements provide a clear understanding of the system's intended purpose and its expected capabilities. Following are the detailed functional requirements for our project.

* **Image dataset import:**

The system should be able to import the image dataset of fresh and rotten fruit from the specified link.

* **Image Preprocessing:**

The system should preprocess the images in the dataset, which may include tasks such as resizing, normalization, or applying filters to enhance image quality.



* **Training with Convolutional Neural Network (CNN):**

The system should use a CNN model to train the image dataset for classification.

The system should utilize one of the transfer learning techniques (e.g., Inception, Xception, VGG, ResNet) to enhance the training process and leverage pre-trained models.

* **Evaluation on Test Data:**

The system should evaluate the trained model on the separate test dataset to measure its classification performance.

* **Testing of Computer vision:**

Finally the system should get a random input image and then evaluate and give result that weather that random image is rotten or fresh by classifying its result.

* **User Interface:**

Our system should have a user interface that allows our users to interact with the system easily it also provide options for users to input images for classification and receive classification results that is a rotten or fresh fruit image.

* **Documentation and Reporting:**

Create comprehensive documentation detailing the system architecture, implementation details, and user guidelines.Generate reports summarizing the performance evaluation results, providing insights into the system's classification accuracy and performance metrics.

**Non-Functional Requirements:**

Non-functional requirements for the computer vision project define the qualities and constraints that are not directly related to the system's specific functionalities but are crucial for its overall performance, usability, and other aspects. Following are the detailed Non-functional requirements for our project.

* **Performance of Model or System:**

The response time and the accuracy of our system can describe the performance of our system.

* **User Friendly System:**

The user interface should be user-friendly and allow our user to easily input images for classification and understand the classification results.



* **Security and Privacy:**

The system should implement appropriate security measures to protect the confidentiality and integrity of the image dataset, trained models, and user data.

* **Portability:**

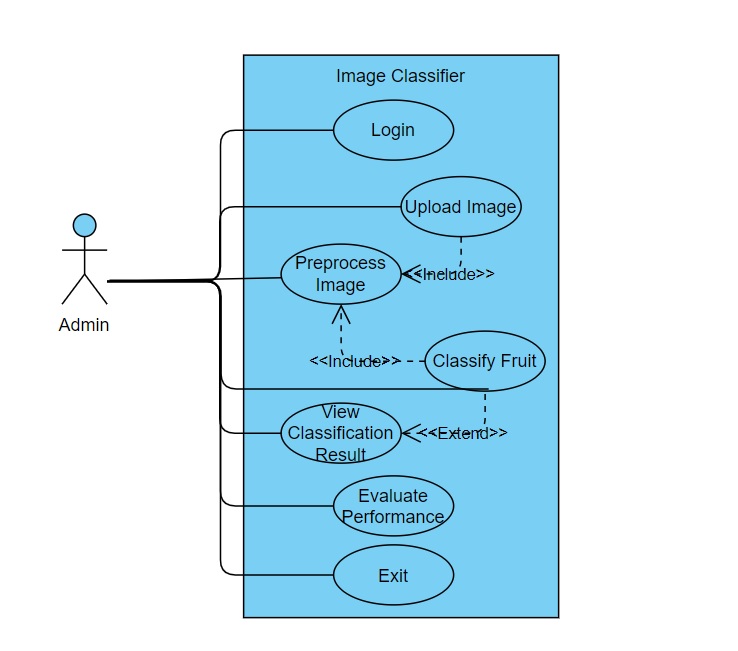
Our system must be portable across different platforms or environments, enabling deployment on various operating systems or other platforms.

* **Compatibility:**

The system should be compatible with commonly used web browsers and ensure compatibility with different image formats i.e bitmap, jpg etc.

Use Case Diagram(s):

Figure 1: Use Case Diagram



Usage Scenarios:



Usage Scenario 1:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | Login | |
| **Use Case Id** | UC001 | |
| **Description:**  The admin logs into the system by providing a username and password. Upon successful verification, access to the system’s functionalities is granted. | | |
| **Pre Conditions:**   The admin must be registered in the system.   The system must be connected to the database. | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. Open user interface | | 1.Un-Recognized image format. |
| 1. Admin enters their username and password. | |
| 1. The system verifies the credentials. | |
| 4. Admin is successfully logged in. | |
| **Post Conditions:**   If the login is successful, the admin is granted access to the system’s functionalities.   If the login fails, an error message is displayed, and the admin remains on the login screen. | | |
| **Alternative Path:**  Invalid Credentials:   Admin enters an incorrect username or password.   The system displays an error message: "Invalid username or password."   Admin is prompted to re-enter the credentials or reset the password.  **Forgot Password:**   Admin clicks on "Forgot Password".   The system prompts the admin to reset the password via email. | | |
| **Author:** (BC200418293 & MC220201538) | | |

Usage Scenario 2:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Upload Image** | |
| **Use Case Id** | **UC002** | |
| **Description:**  The admin uploads an image of a fruit (either fresh or rotten) for classification. | | |
| **Pre Conditions:**   The admin must be logged in.   The image file must be in a supported format (JPG, PNG). | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. Admin selects the "Upload Image" option. | | 1. If the provided image is not of a rotten or fresh fruit.  2. If the file is too large, the system displays a message: "File size exceeds the allowed limit." |
| 1. Admin browses and selects an image from the file system. | |
| 1. The system validates the image format | |
| 4. The system uploads and displays the image for further processing | |
| **Post Conditions:**   The image is successfully uploaded, ready for classification.   If the image is not uploaded, the system prompts the admin to retry. | | |
| **Alternative Path:**  **(Unsupported File Type):**   Admin selects a file that is not a valid image format (e.g., PDF).   The system shows an error message: "Unsupported file format. Please upload a JPG or PNG image."   Admin is prompted to select a valid file format. | | |
| **Author:** (BC200418293 & MC220201538) | | |



Usage Scenario 3:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | Classify Fruit | |
| **Use Case Id** | UC003 | |
| **Description:**  The system classifies the uploaded image as either fresh or rotten using a Convolutional Neural Network (CNN) model. | | |
| **Pre Conditions:**   The admin must have successfully uploaded an image.   The CNN model must be properly loaded and functional. | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. Admin uploads an image. | | If the image preprocessing fails (e.g., unreadable format), the system shows an error message: "Failed to process the image. Please upload a v alid image." |
| 1. The system preprocesses the image (cropping, resizing). | |
| 1. The system uses the CNN model to classify the image as fresh or rotten. | |
| 1. The classification result is displayed to the admin. | |
| **Post Conditions:**   The fruit is classified as either fresh or rotten.   If classification fails, the system shows an error message. | | |
| **Alternative Path:**  **Classification Failure**   The image quality is too poor for classification.   The system displays an error message: "Unable to classify the image. Please upload a clearer image."   Admin is prompted to upload a higher quality image. | | |
| **Author:** (BC200418293 & MC220201538) | | |

Usage Scenario 4:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | View Classification | |
| **Use Case Id** | UC004 | |
| **Description:**  This use case is about the user will view the result of classification which is displayed by the system. | | |
| **Pre Conditions:**  User must have a specific rotten or fresh fruit image in his local drive. | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. Model will display the result. | | 1. If image have both the rotten or fresh fruit in it.  2. If the image does not contains a rotten or fresh fruit the error accour. |
| 1. User will open the user interface to check the result. | |
| 1. User will check the accuracy of the result. | |
| 1. User will verify that the given image is classified as a rotten or fresh fruit well or wrongly. | |
| **Post Conditions:**  User has successfully completed its tasks for classification of his image. | | |
| **Alternative Path:**  **(Result Not Available):**   The system fails to retrieve the classification result.   The system displays an error message: "Classification result unavailable. Please try again."   Admin can retry or exit the process. | | |
| **Author:** (BC200418293 & MC220201538) | | |



Usage Scenario 5:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | Evaluate Performance | |
| **Use Case Id** | UC005 | |
| **Description:**  The admin evaluates the performance of the classification system by reviewing metrics such as accuracy, precision, recall, and F1-score. | | |
| **Pre Conditions:**   The system must have previously classified images for evaluation.   The evaluation metrics must be stored or computed based on previous classifications. | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. Admin selects "Evaluate Performance" from the system menu. | | 1. If the performance metrics are unavailable due to system failure, the system shows an error: "Unable to retrieve performance data. Please try again later." |
| 1. The system retrieves classification performance metrics (accuracy, precision, recall, F1-score). | |
| 1. The system displays these metrics to the admin. | |
| **Post Conditions:**   The system displays performance metrics to the admin.   If the system fails to retrieve the metrics, an error message is displayed. | | |
| **Alternative Path:**  Performance Data Unavailable:   The system encounters an issue when trying to retrieve performance data (e.g., corrupted data, network issues).   The system displays an error message: "Performance data not available. Please try again later." | | |
| **Author:** (BC200418293 & MC220201538) | | |



Usage Scenario 6:

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | Exit system | |
| **Use Case Id** | UC006 | |
| **Description:**  This use case is about if a user has completed his desired tasks and want to logoff or exit the system. | | |
| **Pre Conditions:**  A satisfied user want to exit the system. | | |
| **Task Sequence / Action** | | **Exceptions** |
| 1. User will go to close button. | | . |
| 1. Click the exit button. | |
| **Post Conditions:**  User has successfully exit the model or the system of the computer vision. | | |
| **Alternative Path:**  Unsaved Data:   Admin selects "Exit" but has unsaved data or incomplete actions.   The system shows a message: "You have unsaved changes. Do you want to save before exiting?"   Admin can choose to save the changes or exit without saving. | | |
| **Author:** (BC200418293 & MC220201538) | | |

**Adopted Methodology:**

Software development methodology refers to the process, approach, or framework used to plan, design, develop, and deliver software applications or systems. It provides a structured and systematic approach to managing and executing software development projects. A methodology defines a set of guidelines, practices, and principles that guide the entire software development lifecycle, including requirements gathering, design, coding, testing, deployment, and maintenance.

There are various software development methodologies, each with its own set of principles and practices. Some commonly used methodologies include:

* Agile software Development
* Incremental model
* Crystal Methods
* Extreme programming



* Waterfall model
* Spiral model

Using a combination of the Spiral and Waterfall methodologies can provide certain advantages in software development projects. Hence I use these two methodologies for my project. Furthermore VU process model is also the combination of these two methodologies.

* **Waterfall Model:** The Waterfall model is a traditional software development methodology that follows a sequential and linear approach. It is called the "Waterfall" model because the development process flows steadily downwards, just like a waterfall, through a series of predefined phases. Each phase in the Waterfall model is completed before moving on to the next phase, and there is little to no overlap or iteration between the phases. Figure 2 below shows the waterfall model.

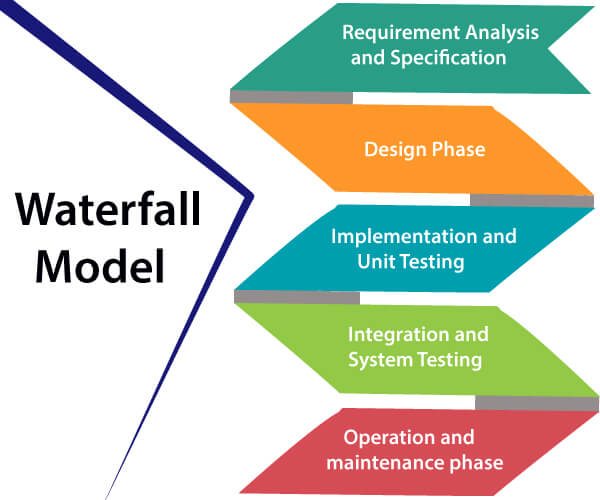


Figure 2: Waterfall Model

* **Spiral Model:** The Spiral model is a software development methodology that combines elements of both waterfall and iterative development models. It was proposed by Barry Boehm in 1986 and is based on a risk-driven approach. The Spiral model incorporates regular cycles of prototyping, risk analysis, and evaluation to guide the software development process. It follows a spiral-shaped progression, with each loop representing a phase of the development cycle. Figure 3 below shows the spiral model.

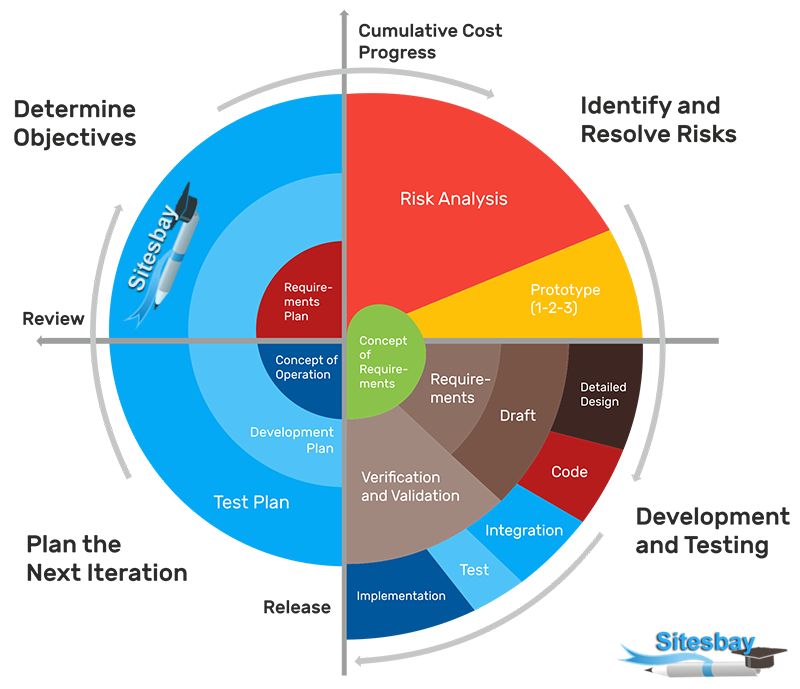


Figure 3: Spiral Model

Virtual University (VU) follows its own software development process model known as the Virtual University Software Process (VUSP) model. The VUSP model is specifically designed to cater to the needs of software development projects within the context of the Virtual University of Pakistan. Here we also adopt the VU process model for our project. Following Figure 4 shows the VU process model which is also our methodology.

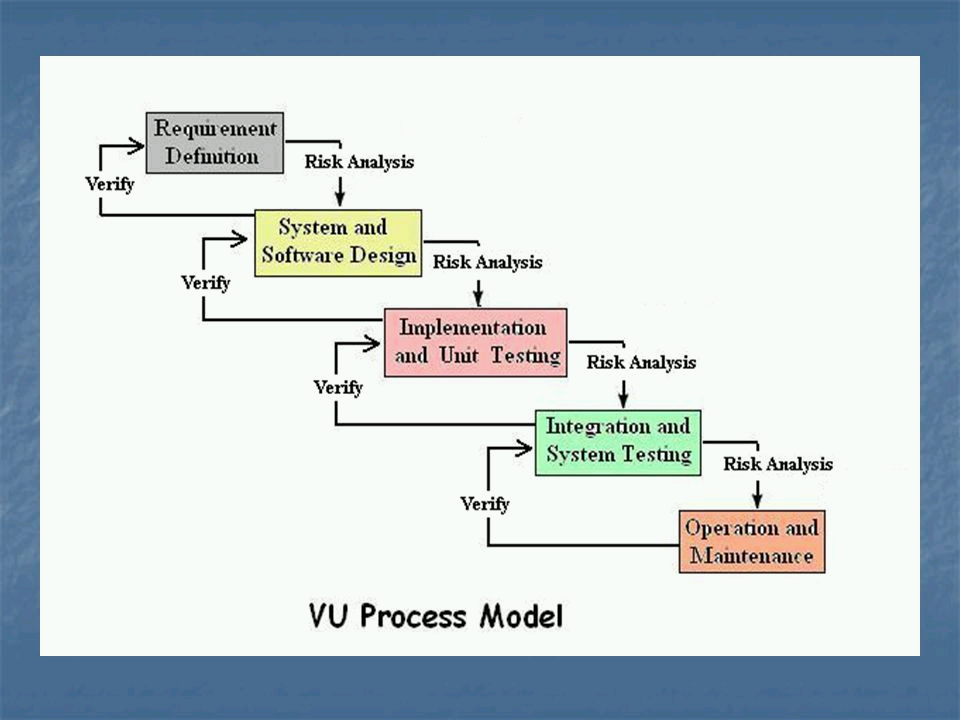


Figure 4: VU Process Model

**Benefit of VU Process Model:**

* This is sequenced model
* Structured approach.
* This model is easy to understand and use
* Each phase has specific deliverable
* In this model phases are completed and processed one at a time.
* Avoidance of risk is enhanced.
* Quality assurance.
* Collaboration and communication.
* Flexibility and adaptability.

**Why we use this methodology:**



* This model is used when cost and risk evaluation is important.
* As this model is efficient in project management.
* This methodology is also used when users are unsure of their requirement.
* It is also used when significant changes are expected.

**Work Plan**

Following figure 5 displays work plan for my project Insha’Allah.

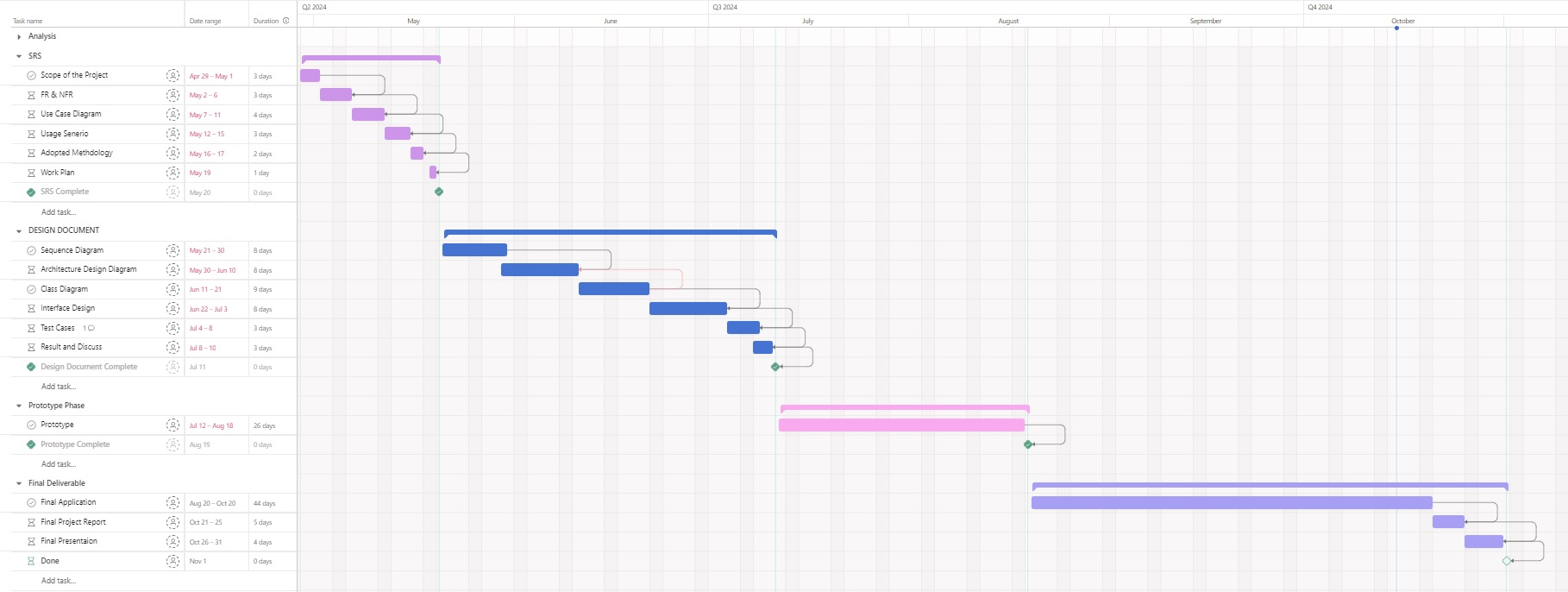
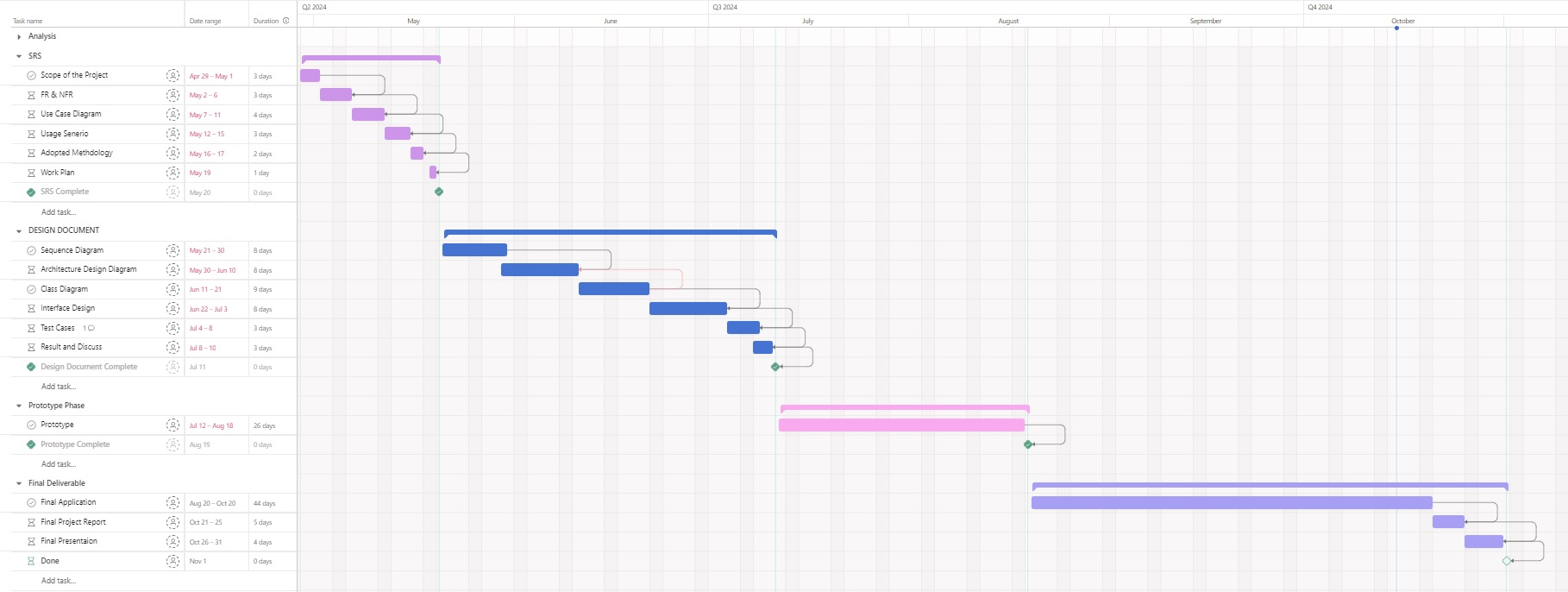


Figure 5: Work Plan

