INDEX

LAB-1

Software Requirements Specification

for

Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

Version 1.0 approved.

Prepared by Mohit, Drishti Arora, Swastik Sindhani

Amity University Noida, Uttar Pradesh

01/08/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the development of a skin lesion classification model that combines deep learning and traditional machine learning techniques for melanoma diagnosis. The software aims to assist medical professionals in accurately detecting melanoma from skin lesion images.

1.2 Document Conventions

The SRS document follows standard conventions for defining requirements and sections.
Requirements are listed using a numbering system (e.g., FR1, NFR1).

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, medical professionals, and stakeholders involved in the development and evaluation of the skin lesion classification system. Readers are advised to start with the introduction and then proceed to the specific sections based on their role and interest.

1.4 Product Scope

The software will analyze skin lesion images and classify them into melanoma or non-melanoma categories. It will utilize hybrid feature extraction, combining deep learning features using transfer learning from pre-trained neural networks and traditional machine learning features with rotation-invariant Local Binary Patterns (LBP) to improve performance and discrimination. The project will also include the development of a user-friendly graphical user interface (GUI) to facilitate user interaction.

1.5 References

N/A

2. Overall Description

2.1 Product Perspective

The software will be a standalone system that takes skin lesion images as input and provides classification results based on the trained hybrid model. It will not be part of any larger system or integrated with other software. The model will be deployed as a local application on the user's machine.

2.2 Product Functions

The software will offer the following features:

2.2.1 Input

The software shall allow users to load skin lesion images for analysis. Users can either upload images through the GUI or provide the file paths to the images stored locally.

2.2.2 Feature Extraction

The software shall extract deep learning features using transfer learning from pre-trained neural networks and combine them with machine learning features obtained from LBP. The hybrid feature extraction process will yield a comprehensive feature representation for each skin lesion image.

2.2.3 Model Training

The software shall train a classification model using the extracted hybrid features and the XGBoost algorithm. The model training process will involve the optimization of hyperparameters to achieve the best possible performance.

2.2.4 Classification

The software shall classify skin lesion images into melanoma or non-melanoma categories based on the trained model. Classification results will be generated for each analyzed image.

2.2.5 Output

The software shall display the classification results and diagnostic information for each analyzed skin lesion image. The GUI will present the images along with their corresponding classifications and any additional information related to the diagnosis.

2.3 User Classes and Characteristics

The software will be used by medical professionals, dermatologists, or healthcare practitioners who are familiar with skin lesion diagnosis and the interpretation of classification results. Users are expected to have basic computer skills and an understanding of medical imaging.

2.4 Operating Environment

The software will be developed to run on standard computing hardware with the following requirements:

- Operating System: The software will be compatible with Windows, macOS, and Linux operating systems.
- Memory: The minimum required RAM for the software to operate efficiently is 8GB
- Disk Space: The software will require sufficient space to store the application files and model parameters.

2.5 Design and Implementation Constraints

The software will be developed using Python programming language, utilizing relevant machine learning libraries such as TensorFlow, Keras, XGBoost, and scikit-learn. The model training process may require significant computational resources, so users should have reasonably powerful computers for optimal performance.

2.6 User Documentation

The software shall be accompanied by comprehensive user documentation, including installation instructions, usage guidelines, and troubleshooting tips. The documentation will be provided in electronic format and made available to users upon software deployment.

2.7 Assumptions and Dependencies

- It is assumed that a dataset of skin lesion images, accurately labeled as melanoma or non-melanoma, will be available for training the model. The dataset will be preprocessed and split into training and validation sets.
- The software will depend on the availability of relevant Python libraries and their compatible versions. It is assumed that users will have the necessary dependencies installed or can easily install them using package managers such as pip.

3. External Interface Requirements

3.1 User Interfaces

The software shall have a graphical user interface (GUI) to allow users to interact with the system. The GUI will present the following components:

3.1.1 Image Upload

The GUI will feature a section where users can upload skin lesion images for analysis. It shall support standard image formats such as JPEG and PNG.

3.1.2 Classify Button

The GUI will have a "Classify" button that users can click to initiate the classification process.

3.1.3 Result Display

The GUI will display the analysis results, including the classification outcome (melanoma or non-melanoma) and any additional diagnostic information.

3.1.4 User Feedback

The GUI will provide feedback to users about the progress of the analysis and any potential errors or warnings.

3.2 Hardware Interfaces

The software shall require standard computing hardware with the following specifications:

- Processor: A modern multi-core processor with a clock speed of at least 2.4 GHz.
- Memory: The minimum required RAM is 8GB.
- Storage: Sufficient free disk space for storing the software, model parameters, and dataset.

3.3 Software Interfaces

The software shall interact with the following software components:

- Operating System: The software will be compatible with Windows 10, macOS 10.14+, and Linux distributions (e.g., Ubuntu 18.04+).
- Python: The software will be developed using Python 3.x.
- Python Libraries: The software will utilize machine learning libraries such as TensorFlow, Keras, XGBoost, and scikit-learn.

3.4 Communications Interfaces

The software does not require any external communication interfaces, as it is a standalone application that processes data locally on the user's machine.

4. System Features

4.1 Input

The software shall allow users to input skin lesion images for analysis. Users can either upload images through the GUI or provide the file paths to the images stored locally.

4.2 Feature Extraction

The software shall extract deep learning features using transfer learning from pre-trained neural networks and combine them with machine learning features obtained from LBP. The hybrid feature extraction process will yield a comprehensive feature representation for each skin lesion image.

4.3 Model Training

The software shall train a classification model using the extracted hybrid features and the XGBoost algorithm. The model training process will involve the optimization of hyperparameters to achieve the best possible performance.

4.4 Classification

The software shall classify skin lesion images into melanoma or non-melanoma categories based on the trained model. Classification results will be generated for each analyzed image.

4.5 Output

The software shall display the classification results and diagnostic information for each analyzed skin lesion image. The GUI will present the images along with their corresponding classifications and any additional information related to the diagnosis.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software shall process an input image and provide classification results within a reasonable time frame. The processing time should be minimized to ensure efficient usage, especially when handling a large number of images simultaneously.

5.2 Safety Requirements

The software shall not be used as a substitute for medical diagnosis. It is intended to assist medical professionals, and the final diagnosis must be made by a qualified healthcare practitioner.

5.3 Security Requirements

The software shall not collect or transmit any personal or sensitive data from users. It shall not access the internet or external resources during its operation.

5.4 Software Quality Attributes

- Accuracy: The software shall achieve an accuracy of at least 85.2% on a validation dataset.
- AUC Score: The software shall achieve an AUC score of at least 0.91 on a validation dataset.
- Usability: The software's graphical user interface (GUI) shall be intuitive, user-friendly, and visually appealing.

5.5 Business Rules

The software's usage shall comply with all relevant legal and ethical regulations governing the handling and analysis of medical data.

6. Other Requirements

This section outlines additional requirements that are critical for the successful development, deployment, and maintenance of the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" project.

6.1 Data Privacy and Security

6.1.1 Data Protection

The software must comply with all relevant data protection and privacy laws, regulations, and industry standards to ensure the confidentiality and integrity of patient data and skin lesion images. It shall implement robust security measures to safeguard against unauthorized access, data breaches, or data misuse.

6.1.2 Encryption

The software shall use encryption techniques when transmitting sensitive data over networks and when storing data locally to prevent unauthorized interception and access.

6.2 Data Privacy and Security

6.2.1 Error Messages

The software must provide clear and informative error messages to users in case of any errors or unexpected situations. The messages should guide users on how to resolve the issues or seek appropriate support.

6.2.2 Logging

The software shall maintain detailed logs of system activities, including error logs, user actions, and system events. The logs will aid in diagnosing issues, identifying patterns, and monitoring the system's health.

6.3 User Authentication and Access Control

6.3.1 User Authentication

The software shall implement a secure user authentication mechanism to ensure that only authorized personnel can access the system. User accounts will be password-protected, and password policies (e.g., complexity, expiration) will be enforced.

6.3.2 Access Control

The software shall support role-based access control (RBAC) to manage user privileges and access rights based on their roles and responsibilities. Access to certain functionalities may be restricted to specific user groups.

6.4 Performance Testing

6.4.1 Load Testing

The software must undergo rigorous load testing to evaluate its performance under various user loads and stress conditions. The testing will help identify performance bottlenecks and ensure that the system can handle concurrent user requests efficiently.

6.4.2 Scalability

The software architecture shall be designed to scale seamlessly with increasing data volume and user traffic. It must be capable of handling growing datasets and user demands without compromising performance.

6.5 Regulatory Compliance

6.5.1 Medical Device Regulations

If the software is intended for medical use or diagnosis, it must comply with relevant medical device regulations, including FDA regulations (if applicable) or any other applicable local or international standards.

6.5.2 Ethical Considerations

The development and use of the software shall adhere to ethical principles, ensuring fairness, transparency, and accountability in the use of AI and machine learning algorithms. Steps will be taken to mitigate biases and ensure that the software's predictions are consistent and reliable.

6.6 Documentation and Training

6.6.1 User Manual

The software shall be accompanied by a comprehensive user manual that provides detailed instructions on how to use the system, load images, interpret results, and handle errors.

6.6.2 Technical Documentation

The development team shall maintain detailed technical documentation, including system architecture, design decisions, algorithms, and data preprocessing steps. The documentation will aid in future enhancements, maintenance, and knowledge transfer.

6.6.3 Training Materials

For medical professionals using the software, the project team shall provide training materials and resources to help users understand the classification results, interpret diagnoses, and make informed decisions based on the system's outputs.

Appendix A: Glossary

This glossary provides definitions for the terms and acronyms used throughout the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" Software Requirements Specification (SRS).

- SRS: Software Requirements Specification - A document that specifies the functional and non-functional requirements of a software project.
- LBP: Local Binary Patterns - A texture descriptor used for feature extraction in image analysis.
- AUC: Area Under the Curve - A metric used to evaluate the performance of a classification model, particularly in receiver operating characteristic (ROC) analysis.
- GUI: Graphical User Interface - The visual interface that allows users to interact with software using graphical elements such as buttons and menus.

Appendix B: Analysis Models

This appendix presents the analysis models used in the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" project to illustrate data flow and system interactions.

B.1 Data Flow Diagram

The Data Flow Diagram (DFD) represents the flow of data within the software during the analysis process. It outlines the inputs, processes, and outputs involved in the classification of skin lesion images.

The DFD consists of the following components:

- Input Data: This represents the skin lesion images provided by the users for analysis. The images are either uploaded through the GUI or provided as file paths.
- Feature Extraction: This process involves two stages: deep learning feature extraction and Local Binary Patterns (LBP) feature extraction. Deep learning features are obtained using transfer learning from pre-trained neural networks, while LBP features are computed from the images.
- Model Training: The extracted hybrid features are used to train the XGBoost classifier, which is responsible for learning the patterns that distinguish between melanoma and non-melanoma lesions.
- Classification: The trained model classifies the skin lesion images into two categories: melanoma or non-melanoma.
- Classification Results: The final output displays the classification results and diagnostic information for each analyzed skin lesion image. The GUI presents the images along with their corresponding classifications and any additional information related to the diagnosis.

B.2 Sequence Diagram

Sequence diagrams represent the interactions between objects in the software system over time. They illustrate the sequence of steps and messages exchanged during the analysis process.

B.2.1 Sequence Diagram: Image Upload and Analysis

This sequence diagram depicts the steps involved in loading skin lesion images and initiating the analysis process. It shows the interactions between the user, GUI components, and backend processes during image analysis.

The sequence of actions includes:

1. The user opens the GUI to start the analysis.
2. The GUI displays an image upload section where the user can either select skin lesion images or drag and drop them for analysis.
3. The user clicks the "Classify" button to initiate the analysis.
4. The software performs feature extraction using deep learning and LBP on the uploaded images.
5. The hybrid features are fed into the XGBoost classifier for model training.
6. The trained model classifies each image as melanoma or non-melanoma.
7. The GUI displays the classification results and diagnostic information for each image.

The sequence diagram illustrates the flow of events and communication between components, enabling a better understanding of the analysis process.

B.3 Class Diagram (if applicable)

The class diagram represents the classes, attributes, and relationships between classes in the software system. It showcases the essential components and their interactions within the project.

The class diagram will encompass the following key components:

- User Interface (GUI): Represents the graphical user interface that allows users to interact with the system.
- Data Processor: Handles data preprocessing and feature extraction processes.
- Classification Model: Represents the trained XGBoost classifier used for image classification.
- Data Storage: Manages the storage and retrieval of skin lesion images and analysis results.

The class diagram provides an overview of the system's structural elements and their associations, aiding in the understanding of the software's architecture and design

Appendix C: To Be Determined List

The "To Be Determined" (TBD) list contains items or information that are yet to be finalized or specified in the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" Software Requirements Specification (SRS). As the project progresses, these TBD items will be addressed and updated with the necessary details.

TBD List:

- Dataset Details: The specific details of the dataset, including the number of skin lesion images, the distribution of melanoma and non-melanoma cases, and any additional preprocessing steps applied to the dataset.
- Model Performance Evaluation: The comprehensive evaluation results of the trained model, including accuracy, AUC score, precision, recall, and F1-score. The appendix will include confusion matrices and ROC curves to assess the model's performance.
- GUI Design: Detailed specifications and mock-ups of the graphical user interface (GUI) design, including the layout, color scheme, and interactive elements.

- Data Flow Diagram (DFD): The finalized Data Flow Diagram illustrating the flow of data and processes within the software during the analysis of skin lesion images.
- Sequence Diagrams: Additional sequence diagrams representing different use case scenarios, interactions, and system behaviors.
- Class Diagram: The complete class diagram showcasing all relevant classes, attributes, and associations within the software system.
- External Interface Requirements: Detailed descriptions of the external interfaces, including any specific communication protocols and data transfer mechanisms required for the software.
- Software Constraints: Any additional design and implementation constraints that may affect the software's development and deployment.
- User Documentation: A comprehensive list of user documentation components, such as user manuals, on-line help, and tutorials that will be delivered along with the software.

The TBD list will be updated and completed during the development process, and the relevant sections of the SRS will be revised accordingly to ensure a comprehensive and detailed specification of the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" project.

LAB-2

AIM

To make Project Timeline for Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis.

Theory

A project activity timeline, also known as a task timeline or task schedule, is a component of a project timeline that specifically focuses on detailing the chronological sequence of individual activities or tasks that need to be completed to accomplish a project. It provides a granular view of the project's timeline, highlighting when each task starts, how long it takes to complete, and when it is expected to finish. This level of detail helps project managers and team members understand the day-to-day or week-to-week progress of the project and ensures that tasks are completed in the right order and on time.

Phase 1: Requirement Gathering (Days 1-10)

- Days 1-2: Define project scope, goals, and objectives.
- Days 3-4: Understand dataset and problem statement deeply.
- Days 5-6: Identify success criteria and evaluation metrics.
- Days 7-8: Determine constraints and potential challenges.
- Days 9-10: Finalize initial project documentation.

Phase 2: Designing (Days 11-16)

- Days 11-12: Design the architecture of the machine learning pipeline.
- Days 13-14: Plan the structure of the codebase and modules.
- Days 15-16: Design data preprocessing and transformation steps.

Phase 3: Coding Implementation (Days 17-36)

- Days 17-19: Set up the development environment and version control.
- Days 20-24: Implement the dataset class for loading and preprocessing.
- Days 25-28: Code the feature extraction using various models.
- Days 29-32: Implement the classifier training and validation.
- Days 33-36: Write code for result evaluation and analysis.

Phase 4: Unit Testing (Days 37-44)

- Days 37-38: Create unit tests for individual components (data loading, feature extraction, classifiers).
- Days 39-41: Perform unit tests to ensure each component functions correctly.
- Days 42-44: Debug and refine code based on unit test results.

Phase 5: System Testing (Days 45-58)

- Days 45-48: Integrate different components of the pipeline.
- Days 49-52: Perform end-to-end testing on the integrated system.

- Days 53-55: Identify and rectify any inconsistencies or issues.
- Days 56-58: Validate the system against predefined use cases.

Phase 6: Acceptance Testing (Days 59-67)

- Days 59-61: Collaborate with stakeholders to define acceptance criteria.
- Days 62-64: Conduct acceptance testing with representative datasets.
- Days 65-67: Address feedback and make necessary improvements.

Phase 7: Deployment (Days 68-85)

- Days 68-70: Package the pipeline and model for deployment.
- Days 71-73: Integrate the model into the desired application or environment.
- Days 74-78: Perform deployment testing and compatibility checks.
- Days 79-82: Prepare necessary documentation for deployment.

Phase 8: Maintenance (Days 86-120)

- Days 86-100: Monitor the deployed system for performance and issues.
- Days 101-110: Address any bugs, errors, or user feedback.
- Days 111-115: Fine-tune the model and pipeline for improved results.
- Days 116-120: Continue maintaining the system and providing support.

LAB-3

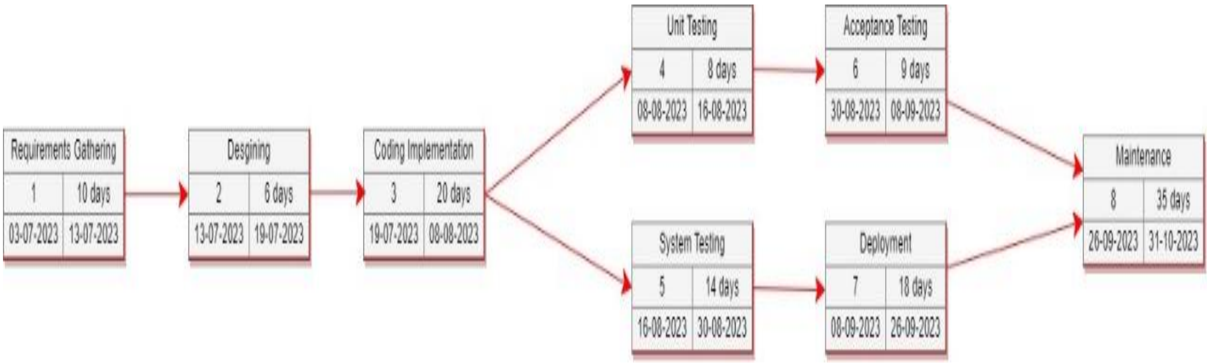
AIM

To construct PERT chart table with Dependencies, Duration, and other attributes for the project: Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

Theory

A PERT (Program Evaluation and Review Technique) chart is a project management tool used to plan, schedule, and manage tasks within a project. PERT charts are especially useful for complex projects that involve a high level of uncertainty and interdependency among tasks. The primary purpose of a PERT chart is to help project managers and teams analyze and visualize the sequence of activities and the time required to complete a project.

PERT Chart



Advantages of using a PERT chart

- **Visual Representation:** It provides a clear visual representation of the project's tasks and their relationships.
- **Risk Analysis:** PERT allows project managers to assess the impact of uncertainties by considering both optimistic and pessimistic estimates.
- **Resource Allocation:** It helps in allocating resources efficiently by identifying critical tasks that need special attention.
- **Project Control:** PERT charts assist in monitoring progress and identifying delays promptly.

LAB-4

AIM

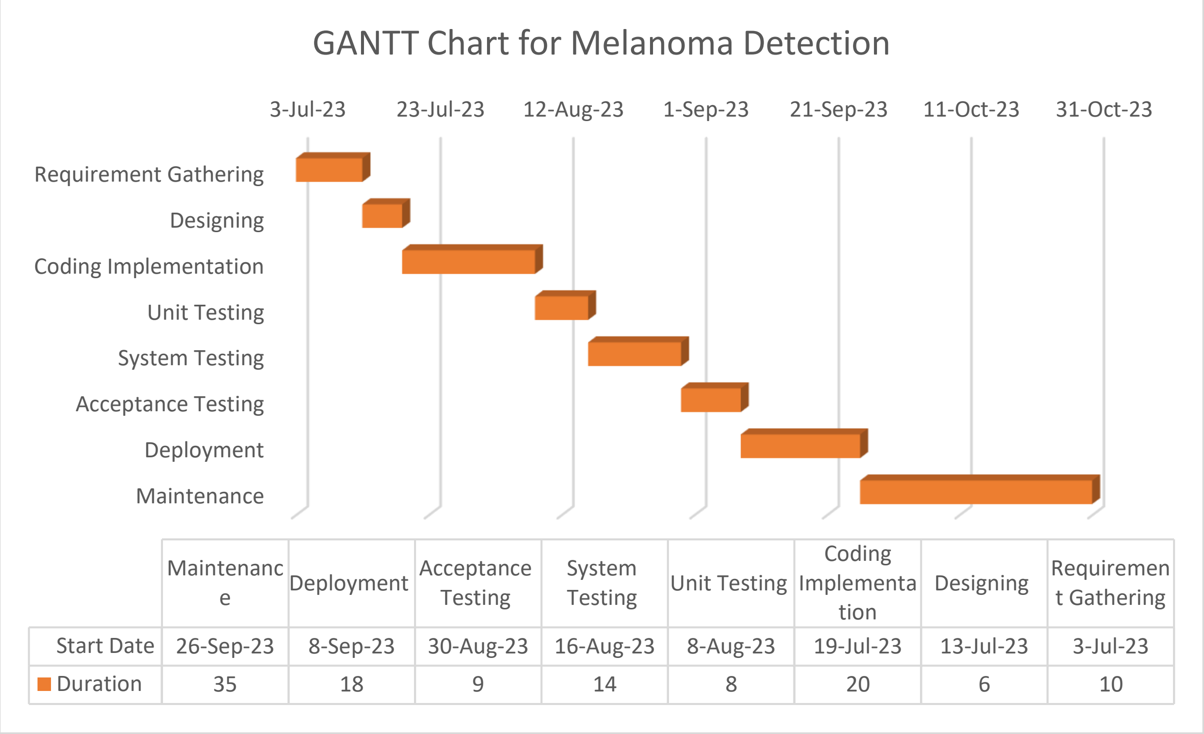
To construct GANTT chart table with Dependencies, Duration, and other attributes for the project: Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

Output

S. No.	Activity	Total Duration	Early Start	Early Finish	Late Start	Late Finish	Float
1	Requirement Gathering (Days 1-10)	10	1	10	1	10	0
1.1	Define project scope, goals, and objectives	2	1	2	1	2	0
1.2	Understand dataset and problem statement deeply	2	3	4	3	4	0
1.3	Identify success criteria and evaluation metrics	2	5	6	5	6	0
1.4	Determine constraints and potential challenges	2	7	8	7	8	0
1.5	Finalize initial project documentation	2	9	10	9	10	0
2	Designing (Days 11-16)	6	11	16	11	16	0
2.1	Design the architecture of the ML pipeline	2	11	12	11	12	0
2.2	Plan the structure of the codebase and modules	2	13	14	13	14	0
2.3	Design data preprocessing and transformation steps	2	15	16	15	16	0
3	Coding Implementation (Days 17-36)	20	17	36	17	36	0
3.1	Set up development environment and version control	3	17	19	17	19	0
3.2	Implement dataset class for loading and preprocessing	5	20	24	20	24	0
3.3	Code feature extraction using various models	4	25	28	25	28	0
3.4	Implement classifier training and validation	4	29	32	29	32	0
3.5	Write code for result evaluation and analysis	4	33	36	33	36	0

S. No.	Activity	Total Duration	Early Start	Early Finish	Late Start	Late Finish	Float
4	Unit Testing (Days 37-44)	8	37	44	37	44	0
4.1	Create unit tests for individual components	2	37	38	37	38	0
4.2	Perform unit tests	3	39	41	39	41	0
4.3	Debug and refine code based on unit test results	3	42	44	42	44	0
5	System Testing (Days 45-58)	14	45	58	45	58	0
5.1	Integrate different components of the pipeline	4	45	48	45	48	0
5.2	Perform end-to-end testing on the integrated system	4	49	52	49	52	0
5.3	Identify and rectify inconsistencies or issues	3	53	55	53	55	0
5.4	Validate the system against predefined use cases	3	56	58	56	58	0
6	Acceptance Testing (Days 59-67)	9	59	67	59	67	0
6.1	Define acceptance criteria	3	59	61	59	61	0
6.2	Conduct acceptance testing with representative data	3	62	64	62	64	0
6.3	Address feedback and make necessary improvements	3	65	67	65	67	0
7	Deployment (Days 68-85)	18	68	85	68	85	0
7.1	Package pipeline and model for deployment	3	68	70	68	70	0
7.2	Integrate model into desired application/enviro	3	71	73	71	73	0
7.3	Perform deployment testing and compatibility checks	5	74	78	74	78	0
7.4	Prepare documentation for deployment	4	79	82	79	82	0
8	Maintenance (Days 86-120)	35	86	120	86	120	0
8.1	Monitor deployed system for performance and issues	15	86	100	86	100	0

GANTT Chart



	Maintenance	Deployment	Acceptance Testing	System Testing	Unit Testing	Coding Implementation	Designing	Requirement Gathering
Start Date	26-Sep-23	8-Sep-23	30-Aug-23	16-Aug-23	8-Aug-23	19-Jul-23	13-Jul-23	3-Jul-23
🟧 Duration	35	18	9	14	8	20	6	10

Notes

1. The "Predecessors" column lists the tasks that need to be finished before beginning a certain task.
2. Each activity's expected completion time (in days) is shown in the "Duration" column.
3. Based on each activity's requirements and the project timeline, the "Early Start (ES)" and "Early Finish (EF)" columns show the earliest start and finish timeframes feasible.
4. The "Late Start (LS)" and "Late Finish (LF)" columns show the earliest timeframes at which each activity might possibly begin and end without pushing back the project's completion deadline.
5. The length of time an activity may be postponed without having an impact on the project's total duration is indicated in the "Slack/Float" column. The activity is on the critical path if the slack is zero. Float/Slack = Last Finish – Early Finish

LAB-5

AIM

To make a Project Plan for the Project: Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis.

Document Change Control

Revision Number	Date of Issue	Author(s)	Brief Description of Change
1	20/09/2023	Drishti Arora, Mohit, Swastik Sindhani	Prepared workflow and included all details about the project

PROJECT PLAN

AMITY UNIVERISTY, UTTAR PRADESH

Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

Document Revision #: 1

Date of Issue: 20/09/2023

Project Manager: Ms. Drishti Arora

Approval Signatures

Approved by: Business Project Leader	Approved by: IM/IT Project Leader
Prepared by: Business Project Manager	Prepared by: IM/IT Project Manager
	Reviewed by: Quality Assurance Manager

Table of Contents

Document Change Control

This section provides control for the development and distribution of revisions to the Project Charter up to the point of approval. The Project Charter does not change throughout the project life cycle, but rather is developed at the beginning of the project (immediately following project initiation approval, and in the earliest stages of project planning). The Project Charter provides an ongoing reference for all project stakeholders. The table below includes the revision number (defined within your Documentation Plan Outline), the date of update/issue, the author responsible for the changes, and a brief description of the context and/or scope of the changes in that revision.

Revision Number	**Date of Issue**	**Author(s)**	**Brief Description of Change**
1	20/09/2023	Drishti Arora. Mohit, Swastik Sindhani	Prepared workflow and included all details about the project

1. Project Overview

This section of the IM/IT Project Management Plan provides a concise overview of the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" project. The project aims to develop a software solution that assists medical professionals in accurately diagnosing melanoma from skin lesion images. The scope encompasses the creation of a machine learning pipeline, including data preprocessing, feature extraction, model training, and result evaluation.

Key assumptions include the availability of a labeled dataset for training and the compatibility of the software with various operating systems. Constraints include hardware requirements and adherence to data privacy regulations. Project deliverables consist of the developed software, comprehensive user documentation, and a well-trained classification model. The project schedule is divided into several phases, with a dedicated phase for system testing and acceptance testing, followed by deployment and ongoing maintenance.

The budget for the project covers development, testing, and maintenance costs, with allocations for hardware resources and any potential unforeseen expenses. The IM/IT Project Management Plan will evolve as the project progresses to ensure effective project management and successful software delivery.

1.1. Purpose, Scope, and Objectives

Purpose and Scope:

The purpose of the "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" project is to develop a software solution that aids medical professionals in accurately diagnosing melanoma from skin lesion images. This project's scope encompasses the design, development, testing, deployment, and maintenance of the software system.

Considerations:

The project's scope explicitly includes the entire machine learning pipeline, covering data preprocessing, feature extraction, model training, and result evaluation. There are no exclusions or considerations to omit any of these essential components from the project.

Alignment with Business and System Needs:

This project aligns with critical business and system needs related to enhancing the accuracy and efficiency of melanoma diagnosis in the medical field. By providing a reliable tool for skin lesion classification, the project addresses the need for improved diagnostic capabilities.

Project Objectives:

- The project's primary objectives are as follows:
- Develop a robust machine learning pipeline for skin lesion analysis.

- Train a classification model that achieves a minimum accuracy of 85.2% and an AUC score of at least 0.91 on a validation dataset.
- Create an intuitive graphical user interface (GUI) for user-friendly interaction.
- Generate clear and informative diagnostic information for medical professionals.
- Ensure compliance with data privacy regulations and ethical considerations.

Deliverables:

- To achieve the project objectives, the following key deliverables are required:
- Software System: The fully functional software application for skin lesion analysis.
- User Documentation: Comprehensive user manuals and guidance for system usage.
- Trained Model: A well-trained machine learning model for skin lesion classification.
- GUI Design: A visually appealing and user-friendly graphical user interface.
- Diagnostic Reports: Clear and informative reports for each analyzed skin lesion image.

Objectives

- Objective 1: The completion and successful testing of the machine learning pipeline.
- Objective 2: Achieving the specified accuracy and AUC score on a validation dataset.
- Objective 3: Successful deployment of the GUI meeting usability criteria.
- Objective 4: Verification of diagnostic report accuracy and completeness.
- Objective 5: Compliance checks against data privacy and ethical guidelines.

Relationship to Other Projects:

- This project is independent and not directly linked to other projects. It operates as a standalone system focused on skin lesion analysis.

Integration with Other Projects or Ongoing Work:

- As a standalone application, this project does not integrate with other ongoing work processes. It is designed to be a self-contained solution for melanoma diagnosis.

Reference to Official Project Requirements:

- The official project requirements are documented in the business case, providing detailed insights into the project's purpose, scope, and objectives. This IM/IT Project Management Plan aligns with and expands upon these requirements to ensure a comprehensive understanding of the project's management and execution.

1.2. Assumptions, Constraints and Risks

Assumptions:

- Availability of Labeled Dataset: The project assumes the availability of a high-quality dataset of skin lesion images accurately labeled as melanoma or non-melanoma. This dataset is essential for training and validating the machine learning model.
- Compatibility with Operating Systems: The software assumes compatibility with multiple operating systems, including Windows, macOS, and various Linux distributions, to maximize its accessibility to users.
- Availability of Necessary Python Libraries: It is assumed that the required Python libraries for machine learning, such as TensorFlow, Keras, XGBoost, and scikit-learn, will be available and compatible with the development environment.
- User Skills: Users of the software are assumed to have basic computer skills and a fundamental understanding of medical imaging and skin lesion diagnosis.

Constraints:

- Hardware Requirements: The project is constrained by hardware requirements. Users are expected to have reasonably powerful computers with a minimum of 8GB of RAM to ensure optimal performance, especially during model training.
- Development Timeline: The project schedule is constrained by the allotted timeframe for development, testing, and deployment. Any delays may impact the project's timeline and delivery.
- Budget Limitations: The project operates within a predefined budget, which covers development, testing, and maintenance costs. Unexpected expenses must be managed within this budget.
- Data Privacy Regulations: Compliance with data privacy regulations is a constraint. The software must adhere to all relevant data protection laws and implement robust security measures to safeguard patient data.
- Ethical Considerations: Ethical considerations, including fairness and bias mitigation in AI algorithms, must be adhered to during model development to ensure ethical usage.

Risks:

- Data Quality: The quality and representativeness of the skin lesion dataset may pose a risk to the project's success. Inaccurate or biased labeling could affect model performance.
- Model Performance: There is a risk that the machine learning model may not achieve the desired accuracy and AUC score on the validation dataset, necessitating additional model tuning and development time.

- Compliance Issues: Failure to adhere to data privacy and ethical guidelines could lead to legal and reputational risks. It is essential to ensure strict compliance throughout the project.
- Resource Constraints: Resource constraints, including limited development resources or hardware limitations, may impact the project's ability to meet its objectives within the defined schedule and budget.
- External Dependencies: Any dependencies on external software components or libraries may introduce risks related to compatibility, updates, and maintenance.
- User Adoption: The software's success depends on user adoption and acceptance within the medical community. Ensuring the GUI's user-friendliness and effectiveness is critical to mitigate this risk.
- Security Vulnerabilities: The software may be susceptible to security vulnerabilities. Ongoing monitoring and security measures are required to mitigate potential risks associated with data breaches or unauthorized access.
- The project management team will actively monitor these assumptions, constraints, and risks throughout the project's lifecycle and take necessary actions to mitigate their impact on project success.

1.3. Project Deliverables

Project Deliverables:

- Software System: The fully functional "Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis" software application, including the graphical user interface (GUI) and underlying machine learning pipeline.
- User Documentation: Comprehensive user manuals and guidance documents for system usage, installation, and troubleshooting.
- Trained Model: A well-trained machine learning model for skin lesion classification, including the necessary model parameters.
- GUI Design: Visual assets and design specifications for the user-friendly graphical user interface.
- Diagnostic Reports: Clear and informative reports generated for each analyzed skin lesion image.

Delivery Dates:

- The delivery dates for each project deliverable are as follows:
- Software System: Day 85 of the project (end of the deployment phase).
- User Documentation: Day 82 of the project (preparation for deployment).
- Trained Model: Day 36 of the project (end of the coding implementation phase).

- GUI Design: Day 16 of the project (end of the design phase).
- Diagnostic Reports: Day 58 of the project (end of the system testing phase).

Delivery Location:

- The delivery location for all project deliverables will be the project's central repository, which will be accessible to the project stakeholders and designated personnel.

Quantities Required:

- The quantities required for each deliverable are as follows:
- Software System: One complete and fully functional system.
- User Documentation: Sufficient copies to meet the needs of end-users and stakeholders.
- Trained Model: One complete trained model.
- GUI Design: One set of visual assets and design specifications.
- Diagnostic Reports: Generated for each analyzed skin lesion image.

Delivery Media:

- Project deliverables will be provided in digital format. The following formats will be used for each deliverable:
- Software System: Executable application files compatible with Windows, macOS, and Linux operating systems.
- User Documentation: PDF documents for easy distribution and accessibility.
- Trained Model: Serialized model files in a format compatible with the chosen machine learning framework.
- GUI Design: Digital design files in standard formats (e.g., PSD, AI) along with any necessary image assets.
- Diagnostic Reports: Digital reports in a standardized format (e.g., PDF) for easy sharing and storage.

Special Instructions for Packaging and Handling:

All digital deliverables should be securely packaged and labeled appropriately to ensure easy identification and access. The packaging should include clear instructions for installation, usage, and any specific requirements for each deliverable. Additionally, clear version control and file naming conventions should be followed to maintain organization and traceability of deliverables. The project manager will oversee packaging and delivery to ensure that all stakeholders receive the necessary project outputs promptly and efficiently.

1.4. Schedule and Budget Summary

Schedule Summary:

- Requirement Gathering (Days 1-10): Defining project scope, goals, and objectives; understanding the dataset and problem statement; identifying success criteria; determining constraints and risks; finalizing initial project documentation.
- Designing (Days 11-16): Designing the architecture of the machine learning pipeline; planning the codebase structure and modules; designing data preprocessing and transformation steps.
- Coding Implementation (Days 17-36): Setting up the development environment and version control; implementing the dataset class; coding feature extraction; implementing classifier training; writing result evaluation and analysis code.
- Unit Testing (Days 37-44): Creating unit tests for data loading, feature extraction, and classifiers; performing unit tests; debugging and refining code based on test results.
- System Testing (Days 45-58): Integrating components; performing end-to-end testing; identifying and rectifying inconsistencies; validating the system against predefined use cases.
- Acceptance Testing (Days 59-67): Defining acceptance criteria; conducting acceptance testing; addressing feedback and making improvements.
- Deployment (Days 68-85): Packaging the pipeline and model; integrating the model into the desired environment; performing deployment testing; preparing necessary documentation.
- Maintenance (Days 86-120): Monitoring the deployed system; addressing bugs, errors, and user feedback; fine-tuning the model; providing ongoing maintenance and support.

Budget Summary:

- The project operates within a predefined budget that covers various aspects of the project, including development, testing, and maintenance costs. The budget is itemized at a high level, and major cost categories include:
- Development Costs: This category includes expenses related to software development, hardware resources, and software licenses.
- Testing and Quality Assurance Costs: Expenses for unit testing, system testing, and any specialized testing tools or services.
- Documentation Costs: Costs associated with the creation of user manuals, technical documentation, and training materials.
- Personnel Costs: Expenses related to project team salaries, training, and any external expertise or consulting services.

- Hardware and Infrastructure Costs: Expenses for hardware procurement, server hosting, and infrastructure setup.
- Maintenance and Support Costs: Ongoing expenses for system maintenance, updates, and user support.
- Contingency Fund: A portion of the budget is allocated as a contingency fund to address unforeseen expenses or scope changes.

1.5. Evolution of the Plan

Compliance with Standards:

- This Project Plan adheres to the recommendations outlined in IEEE Std 1058-1998, ensuring that it follows established standards for project management documentation.

Plans for Producing Updates:

- Updates to this Project Plan may be scheduled or unscheduled based on project needs and changes in scope or requirements. The following plans outline how updates will be produced, disseminated, placed under configuration management, and controlled:
- Scheduled Updates: Scheduled updates to this Plan will occur as part of regular project management activities. The project manager will review the Plan at key project milestones, such as the completion of project phases or major deliverables. Updates will be scheduled to reflect changes in project scope, schedule, budget, or other relevant aspects. The schedule for planned updates will be documented in the Project Schedule.
- Unscheduled Updates: Unscheduled updates may arise in response to unexpected changes or critical issues affecting the project. In such cases, the project manager will initiate an unscheduled update to address the specific issue or change. Unscheduled updates will be produced promptly to ensure the Plan remains current and relevant.

Dissemination of Updates:

- Updates to this Project Plan will be disseminated to the project team, stakeholders, and relevant personnel through the following channels:
- Project Team: The updated Plan will be shared with the project team members through project communication channels, such as project meetings, emails, and collaboration tools.
- Stakeholders: Updates that impact project stakeholders will be communicated directly to them through official project communication channels. This may include project status reports or dedicated stakeholder meetings.
- Configuration Management: The initial version of this Project Plan and all subsequent updates will be placed under configuration management. A version control system will be used to maintain a complete history of changes to the Plan.

Control of Changes:

- Changes to this Project Plan after its initial issue will be carefully controlled to ensure that they do not introduce risks or negatively impact the project's success. The change control process will follow these steps:
- Change Request Initiation: Any proposed changes to the Project Plan will begin with a formal change request. This request can be initiated by the project manager, project team members, stakeholders, or any party with a vested interest in the project.
- Change Assessment: The project manager will assess the proposed change's impact on the project's scope, schedule, budget, and other relevant aspects. The assessment will consider the project's objectives and requirements.
- Change Approval: The project manager will seek approval from the project sponsor or appropriate change control board (if applicable). The approval process will involve a review of the change's justification and potential consequences.
- Change Implementation: Approved changes will be documented, implemented, and incorporated into the Project Plan. This includes updating relevant sections of the Plan and ensuring all project stakeholders are informed of the changes.
- Documentation and Tracking: All changes, their justifications, and approvals will be documented and tracked in the project's change log. This log will provide a historical record of changes made to the Project Plan.
- The change control process will be an integral part of project management, ensuring that modifications to the Project Plan are managed systematically and with proper consideration of their impact on project success.

1.6. References

Project Charter

- Title: Melanoma Diagnosis Project Charter

Business Case

- Title: Melanoma Diagnosis Business Case

IEEE Std 1058-1998 - IEEE Standard for Software Project Management Plans

- Title: IEEE Std 1058-1998 - IEEE Standard for Software Project Management Plans
- Author: IEEE Standards Association
- Publishing Organization: IEEE

Project Schedule

- Title: Melanoma Diagnosis Project Schedule

Project Budget Plan

- Title: Melanoma Diagnosis Project Budget Plan

Change Control Process

- Title: Melanoma Diagnosis Change Control Process

Project Communication Plan

- Title: Melanoma Diagnosis Project Communication Plan

1.7. Definitions and Acronyms

Definitions:

- Machine Learning Pipeline: A sequence of data processing and analysis steps that involve data preprocessing, feature extraction, model training, and result evaluation, typically used to build and deploy machine learning models.
- Data Preprocessing: The process of cleaning, transforming, and preparing raw data into a suitable format for machine learning model training. It often includes tasks such as data cleaning, feature scaling, and handling missing values.
- Feature Extraction: The process of selecting or transforming relevant information (features) from raw data that is used as input for machine learning algorithms. It aims to reduce data dimensionality while retaining important information.
- Model Training: The process of using labeled data to teach a machine learning model to make predictions or classifications. During training, the model learns patterns and relationships within the data.
- GUI (Graphical User Interface): A user interface that uses graphical elements such as buttons, menus, and icons to allow users to interact with software or systems. It provides a visual way for users to input data and receive information.
- AUC (Area Under the Receiver Operating Characteristic Curve): A metric used to evaluate the performance of a classification model. It measures the ability of the model to distinguish between positive and negative classes, with a higher AUC indicating better performance.

Acronyms:

- IM/IT: Information Management/Information Technology.
- IEEE: Institute of Electrical and Electronics Engineers..

2. Project Organization

2.1. External Interfaces

The project interacts with various external entities and organizational entities. Below, we describe the organizational boundaries between the project and these external entities:

Parent Organization:

- The parent organization for this project is "Amity University, Uttar Pradesh." Amity University is the higher education institution that sponsors and oversees the project.

Customer:

- Medical Professionals: Dermatologists, healthcare practitioners, and medical researchers who will benefit from the melanoma diagnosis system developed as part of this project.
- Stakeholders: Individuals or groups with a vested interest in the successful outcome of the project, including patients and organizations focused on skin health.

Subcontracted Organizations:

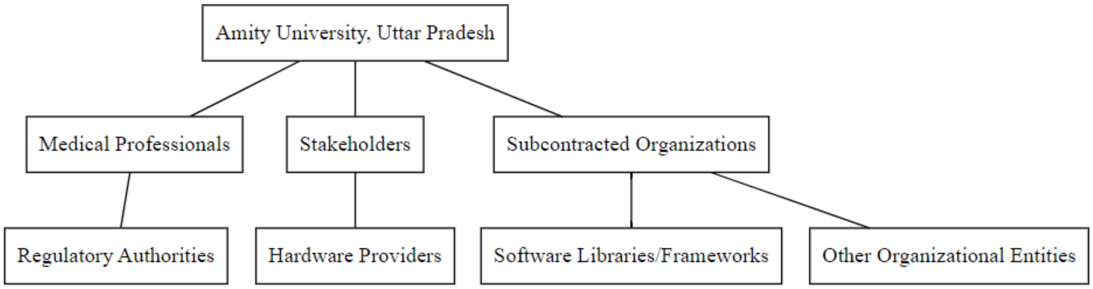
- While the core development and implementation of the melanoma diagnosis system are conducted within the project team at Amity University, there may be instances where specialized services or resources are subcontracted. For example, external expertise in dermatology, medical imaging, or data annotation may be sought as needed.

Other Organizational Entities:

- Regulatory Authorities: Relevant governmental or industry regulatory bodies overseeing the ethical and legal aspects of medical software and data.
- Software Libraries and Frameworks: External entities providing software libraries, frameworks, or tools used in the development of the melanoma diagnosis system (e.g., TensorFlow, Keras, scikit-learn).
- Hardware Providers: Suppliers of computing hardware, servers, and other infrastructure components that may be used for the project's computational requirements.

Organizational Interfaces Diagram:

- The following diagram illustrates the organizational interfaces and relationships between the project and external entities:


Amity University, Uttar Pradesh
Medical Professionals
Stakeholders
Subcontracted Organizations
Regulatory Authorities
Hardware Providers
Software Libraries/Frameworks
Other Organizational Entities

This diagram provides a visual representation of the project's external interfaces, showcasing the interactions between the project and various organizational entities. These external interfaces play a crucial role in the successful development, deployment, and adoption of the melanoma diagnosis system.

2.2. Internal Structure

Project Management Team:

- Project Manager: Responsible for overall project planning, execution, and monitoring. Manages project resources, schedules, and ensures alignment with project objectives.
- Project Coordinator: Assists the project manager in administrative tasks, communication, and documentation.

Development Team:

- Data Scientists and Engineers: Data scientists and machine learning engineers are responsible for developing the melanoma diagnosis system. This includes data preprocessing, feature extraction, model development, and testing.
- GUI Developer: Focuses on the design and development of the graphical user interface (GUI) for user interaction.
- Quality Assurance (QA) Analyst: Ensures the quality and reliability of the software by conducting testing and quality checks.

Domain Experts:

- Medical Experts: Dermatologists and medical professionals provide domain expertise related to skin lesion analysis and contribute to the validation and evaluation of the system.

Supporting Processes:

- Configuration Management Team: Manages version control, configuration items, and ensures the integrity of project assets.
- Quality Assurance Team: Oversees quality assurance processes, conducts audits, and ensures adherence to quality standards.
- Verification and Validation Team: Focuses on the verification and validation of system functionality, including the accuracy of melanoma classification.

Interfaces among Units:

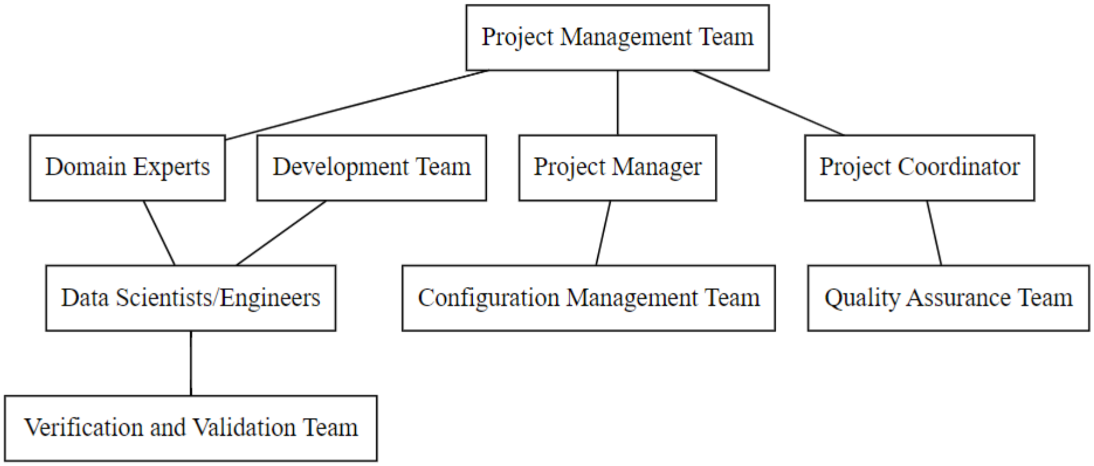
- The development team collaborates closely with the project management team to align development activities with project objectives and milestones.
- Data scientists and engineers work in coordination with GUI developers to ensure seamless integration of the user interface with the underlying machine learning components.
- Quality assurance analysts work with both the development team and quality assurance team to plan and conduct testing activities.
- Domain experts, including medical professionals, provide guidance and feedback to the development team throughout the project.

Interfaces with Supporting Processes:

- The configuration management team interfaces with the development team to manage version control and ensure that software components are appropriately tracked.
- The quality assurance team collaborates with the quality assurance analysts to define testing criteria and conduct quality audits.
- The verification and validation team interfaces with the development team to validate the accuracy and effectiveness of the melanoma diagnosis system.

Organizational Interfaces Diagram:

The following diagram illustrates the organizational interfaces and relationships within the project organization:



This diagram provides a visual representation of the internal structure and interfaces within the project organization. It highlights the key units and their roles in contributing to the successful development of the melanoma diagnosis system. Effective communication and collaboration among these units are essential for project success.

2.3. Roles and Responsibilities

The following matrix outlines these activities and processes along with their associated organizational units and their respective roles and responsibilities:

Work Activities/Processes	Organizational Units	Roles and Responsibilities
Project Management	Project Management Team	- Project Manager: Overall project planning, resource management, and ensuring project alignment with objectives. - Project Coordinator: Assisting in administrative tasks, communication, and documentation.
Development	Development Team	- Data Scientists and Engineers: Development of the melanoma diagnosis system, including data preprocessing, feature extraction, model development, and testing. - GUI Developer: Design and development of the graphical user interface (GUI) for user interaction. - Quality Assurance (QA) Analyst: Testing and ensuring software quality and reliability.
Domain Expertise and Evaluation	Domain Experts	- Medical Experts (Dermatologists): Providing domain expertise related to skin lesion analysis, validation, and evaluation of the system.
Configuration Management	Configuration Management Team	- Configuration Management Team: Managing version control, configuration items, and ensuring the integrity of project assets.
Quality Assurance and Testing	Quality Assurance Team	- Quality Assurance Team: Overseeing quality assurance processes, conducting audits, and ensuring adherence to quality standards. - QA Analysts: Conducting testing and quality checks.
Verification and Validation	Verification and Validation Team	- Verification and Validation Team: Focusing on the verification and validation of system functionality, including accuracy and effectiveness.

This matrix provides a clear overview of the major work activities, supporting processes, and the respective organizational units responsible for each. It ensures that roles and responsibilities within the project organization are well-defined and aligned with project objectives. Effective collaboration among these units is critical for the successful development of the melanoma diagnosis system.

3. Managerial Process Plans

This section of the IM/IT Project Management Plan specifies the project management processes for the project. This section defines the plans for project startup, risk management, project work, project tracking and project closeout.

3.1. Startup Plan

3.1.1. Estimates

Estimated Cost, Schedule, and Resource Requirements:

- The estimated cost for conducting the project is approximately $500,000, with a confidence level of 80%.
- The project is expected to span 12 months, with a confidence level of ±15 days for the schedule estimate.
- Resource requirements include a team of 10 data scientists and engineers, 2 GUI developers, 2 quality assurance analysts, and access to domain experts (dermatologists) for consultation.

Methods and Tools for Estimation:

- Cost Estimate: The cost estimate is derived from historical data and cost models based on similar projects.
- Schedule Estimate: The schedule is estimated using a Work Breakdown Structure (WBS) and historical data on similar projects.
- Resource Estimate: Resource requirements are determined based on the project scope and complexity.

Sources of Estimate Data:

- Estimation data is primarily based on historical data from previous machine learning and software development projects within the organization.
- Data from similar projects, especially those involving medical image analysis, serves as a valuable reference.

Methods for Re-Estimation:

- Regular re-estimation will occur at project milestones, such as the completion of major development phases.
- Re-estimation will be conducted using updated data and project progress information.

Tools and Techniques for Re-Estimation:

- Re-estimation will utilize earned value management (EVM) techniques to assess project progress and adjust cost and schedule estimates accordingly.

Schedule for Re-Estimation:

Re-estimation will be conducted at the following intervals:

- After the completion of the data preprocessing phase.
- After the completion of the feature extraction and model development phase.
- After the completion of the testing and validation phase.

3.1.2.Staffing

Number of Personnel by Skill Level:

- Data Scientists and Engineers: 10 personnel
- GUI Developers: 2 personnel
- Quality Assurance Analysts: 2 personnel

Numbers and Skill Levels in Each Project Phase:

- Data Preprocessing and Setup Phase:
  - Data Scientists and Engineers: 5 personnel
- Feature Extraction and Model Development Phase:
  - Data Scientists and Engineers: 10 personnel
- GUI Development Phase:
  - GUI Developers: 2 personnel
- Testing and Validation Phase:
  - Quality Assurance Analysts: 2 personnel

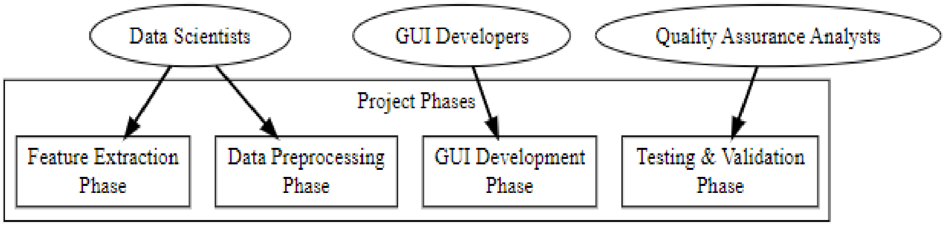
Duration of Personnel Requirement:

- Data Scientists and Engineers will be required throughout the project, with varying levels of involvement in different phases.
- GUI Developers will be primarily active during the GUI Development Phase.
- Quality Assurance Analysts will be engaged primarily during the Testing and Validation Phase.

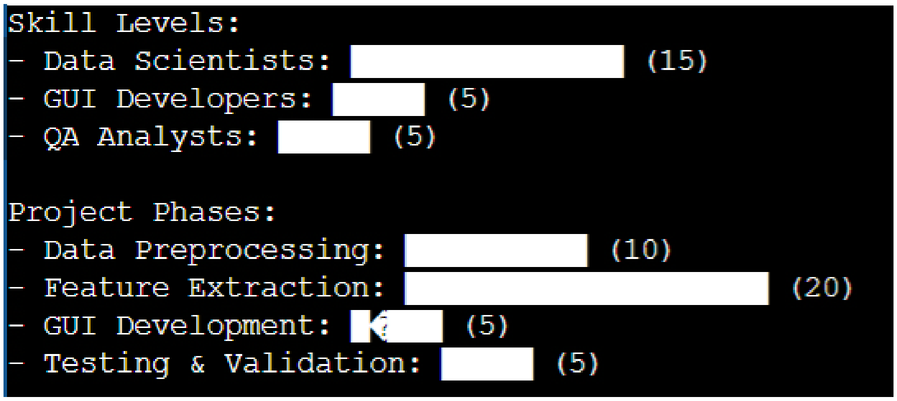
Sources of Staff Personnel:

- Internal Transfer: Staff will be primarily sourced internally from within the organization.
- New Hire: If necessary, new hires may be made to fill specific skill gaps.
- Contracted: Contracted personnel may be utilized for specialized tasks or expertise.

Chart 1 visually represents the project schedule over time, including when each staff member is allocated to specific project phases.



A resource histogram shows the distribution of personnel by skill level and project phase.



```
Skill Levels:
- Data Scientists:  (15)
- GUI Developers:  (5)
- QA Analysts:  (5)

Project Phases:
- Data Preprocessing:  (10)
- Feature Extraction:  (20)
- GUI Development:  (5)
- Testing & Validation:  (5)
```

3.1.3.Resource Acquisition

Resource Acquisition Process: The resource acquisition process for the project will involve a systematic approach to identifying and obtaining the necessary resources. This process will be managed by the Project Manager and will include the following steps:

- Personnel: To fulfill the project's requirements, we will identify and recruit highly skilled data scientists with expertise in machine learning and medical image analysis. We will also engage GUI developers with experience in building user-friendly interfaces for data analysis tools. Additionally, QA analysts with a background in software testing and quality assurance will be brought on board. The recruitment process will include job postings, interviews, and skill assessments.
- Equipment and Hardware: We will assess the hardware requirements, including high-performance computing resources for model training and testing. The acquisition plan will involve procuring the necessary computer hardware and ensuring that it meets the project's computational needs. Additionally, any specialized hardware required for image processing will be identified and acquired.
- Software: The project will rely on a range of software tools and libraries for data preprocessing, feature extraction, machine learning, and graphical user interface development. The acquisition plan will include licensing agreements and the procurement of necessary software licenses.

- Training: To ensure the project team is well-prepared for the tasks ahead, training programs will be organized. Data scientists will receive training in the latest machine learning techniques and medical image analysis. GUI developers will undergo training in interface design and user experience best practices.
- Service Contracts: If external services are required, such as cloud computing resources or specialized consulting, service contracts will be negotiated and executed. These contracts will be reviewed by the Project Manager and legal advisors to ensure compliance with project needs and budget constraints.
- Facilities: The project team will utilize existing laboratory and office spaces within the university for day-to-day work. However, if any specialized facilities or equipment are needed, such as medical imaging facilities, arrangements will be made in collaboration with the university's facilities management.

Responsibility Assignment: The Project Manager will have overall responsibility for resource acquisition. The detailed responsibilities will be as follows:

- Personnel Recruitment: HR and the Project Manager will oversee personnel recruitment, with HR managing administrative aspects, while the Project Manager assesses technical skills and fit with project requirements.
- Equipment and Software Procurement: The Project Manager, in coordination with the IT department, will manage the acquisition of computer hardware and software licenses.
- Training Coordination: The Project Manager will identify training needs and coordinate with external training providers or universities for specialized training programs.
- Service Contracts: The Project Manager and legal advisors will collaborate to review and approve service contracts, ensuring they align with project objectives and budget constraints.

Resource Acquisition Timeline: Resource acquisition activities will be ongoing throughout the project. The timeline for resource acquisition will be integrated with the project schedule to ensure that resources are available when needed.

Constraints: Budget constraints and lead times for equipment and software procurement are among the primary constraints. Additionally, the availability of highly skilled data scientists and GUI developers may pose a challenge, and efforts will be made to recruit suitable candidates promptly.

3.1.4.Project Staff Training

Types of Training:

- Technical Training: Data scientists involved in the project will undergo technical training to enhance their expertise in machine learning, deep learning, and medical image analysis. This training will focus on the latest advancements in the field, transfer learning techniques, and domain-specific knowledge related to dermatology and melanoma diagnosis.
- Managerial Training: Project managers and team leads will receive managerial training to improve their leadership and project management skills. This training will cover areas such as project planning, risk management, and team coordination.
- Supporting Activity Skills: QA analysts and other team members involved in supporting activities will receive training in quality assurance, software testing methodologies, and best practices for ensuring the reliability and performance of the developed system.

Numbers of Personnel to be Trained:

- Data Scientists: All data scientists (approximately 15) will undergo technical training.
- Project Managers and Team Leads: Project managers and team leads (approximately 3) will receive managerial training.
- QA Analysts: QA analysts (approximately 5) will receive training in software testing and quality assurance.

Entry and Exit Criteria for Training:

The entry criteria for training include the selection of relevant team members based on their roles in the project. For example, data scientists will enter technical training, while project managers and team leads will enter managerial training. The exit criteria involve successfully completing the training program and demonstrating the acquired skills and knowledge through assessments and practical applications.

Training Methods:

- Training will be conducted using a variety of methods to cater to different learning styles and needs:
- Lectures: Formal lectures will be organized to provide theoretical knowledge and concepts.
- Consultations: One-on-one or group consultations with subject matter experts will be available to address specific questions and provide guidance.
- Mentoring: Experienced team members will mentor junior staff to facilitate knowledge transfer.

- Computer-Assisted Training: Online courses and interactive training modules will be accessible for self-paced learning.
- Hands-On Workshops: Practical workshops and hands-on exercises will allow team members to apply their knowledge in real-world scenarios.

Training Timeline:

The training activities will be integrated into the project schedule to ensure that team members receive training as needed throughout the project lifecycle. Training will be conducted both at the project outset and at relevant phases to address evolving skill requirements.

3.2. Work Plan

3.2.1. Work Breakdown Structure

The WBS is structured into various levels to provide a detailed view of the project's work activities.

Level 1: Project Phases

1.0 Project Initiation and Planning

2.0 Requirement Gathering and Analysis

3.0 Design and Architecture

4.0 Implementation and Development

5.0 Testing and Validation

6.0 Deployment and Integration

7.0 Project Maintenance and Support

Level 2: Project Phase Subdivisions

1.1 Define Project Scope and Objectives

1.2 Assumptions, Constraints, and Risks

1.3 Project Deliverables and Delivery

1.4 Schedule and Budget Summary

1.5 Evolution of the Plan

1.6 References

Level 3: Detailed Activities

1.1.1 Conduct Initial Project Kick-off Meeting

1.1.2 Define Project Stakeholders and Roles

1.1.3 Develop Project Charter

1.2.1 Conduct Requirement Gathering Workshops

1.2.2 Analyze Dataset and Problem Statement

1.2.3 Define Success Criteria and Metrics

1.3.1 Design ML Pipeline Architecture

1.3.2 Plan Codebase Structure and Modules

1.3.3 Design Data Preprocessing Steps

1.4.1 Create Project Schedule and Budget

1.4.2 Document Schedule and Budget Summary

1.5.1 Comply with IEEE Std 1058 1998

1.5.2 Plan for Plan Updates

1.5.3 Specify Configuration Management

1.5.4 Control Plan Changes

Level 4: Resource Allocation and Dependencies

- For each detailed activity, specify the necessary resources (personnel, tools, software, hardware, etc.).
- Estimate the duration required for each activity.
- Define the products or deliverables expected from each activity.
- Set acceptance criteria for the products.
- Identify predecessor and successor activities to establish dependencies and sequencing.

3.2.2.Schedule Allocation

Sequencing and Dependencies:

The project activities will be sequenced based on the natural flow of the project phases. However, to illustrate opportunities for concurrent work activities and maintain efficiency, we will employ the following scheduling techniques:

- Parallel Development: Activities such as data preprocessing and feature extraction can occur concurrently to expedite the process. For instance, while data preprocessing is in progress, feature extraction can begin simultaneously.
- Iterative Development: Given the iterative nature of machine learning model training, we will incorporate multiple iterations within the classifier training and validation phase to refine the models.
- Testing Parallelism: Testing and validation activities will run in parallel with classifier training to ensure that the models are tested as they are developed.

Critical Path:

The critical path for this project is the sequence of activities that, if delayed, will impact the project's overall timeline. In the "Hybrid Feature Extraction for Melanoma Diagnosis" project, the critical path typically includes:

- Requirement gathering and analysis
- Designing ML pipeline architecture
- Implementing dataset class and feature extraction

- Classifier training and validation
- System testing and integration
- Acceptance testing
- Deployment

Any delays in these activities can potentially extend the project timeline.

Constraints:

External constraints that might affect the project schedule include data availability, stakeholder availability for testing and validation, and external dependencies on software or hardware updates.

Milestones:

Key milestones will be identified to assess the scope and quality of project work products and project achievement status. Milestones for the "Hybrid Feature Extraction for Melanoma Diagnosis" project include:

- Completion of ML pipeline architecture design
- Successful training and validation of melanoma classification models
- Integration of the system components
- Successful completion of acceptance testing
- Deployment of the system in the desired environment
- These milestones will serve as checkpoints to evaluate progress and ensure that the project is on track to meet its objectives.

Schedule Visualization:

To depict the schedule relationships and critical path, we will use Gantt charts, activity networks, and critical path analysis tools.

3.2.3.Resource Allocation

Resource allocation is essential to ensure that the project has the necessary personnel, computing resources, software tools, and facilities to successfully execute each activity.

Data Preprocessing (Days 20-24):

- Data Scientists: 3 (Senior)
- Computing Resources: High-performance GPU servers
- Software Tools: Python, NumPy, Pandas

Feature Extraction (Days 25-28):

- Data Scientists: 2 (Senior)
- Computing Resources: High-performance GPU servers
- Software Tools: Transfer learning frameworks (e.g., TensorFlow, PyTorch)

Classifier Training and Validation (Days 29-32):

- Data Scientists: 3 (Senior)
- Computing Resources: High-performance GPU servers
- Software Tools: Scikit-learn, TensorFlow, PyTorch

Testing & Validation (Days 33-36):

- QA Analysts: 2
- Computing Resources: Standard workstations
- Software Tools: Test automation frameworks, bug tracking tools

Deployment (Days 68-85):

- GUI Developers: 2
- Computing Resources: Server infrastructure
- Software Tools: Web development frameworks, deployment tools

Maintenance (Days 86-120):

- Data Scientists: 2 (Senior)
- QA Analysts: 1
- Computing Resources: Ongoing server infrastructure
- Software Tools: Monitoring and debugging tools

3.2.4.Budget Allocation

Data Preprocessing (Days 20-24):

- Personnel Costs (Data Scientists): $60,000
- Computing Resources: $20,000
- Software Tools: $5,000

Feature Extraction (Days 25-28):

- Personnel Costs (Data Scientists): $40,000
- Computing Resources: $15,000
- Software Tools: $5,000

Classifier Training and Validation (Days 29-32):

- Personnel Costs (Data Scientists): $60,000
- Computing Resources: $20,000
- Software Tools: $10,000

Testing & Validation (Days 33-36):

- Personnel Costs (QA Analysts): $30,000
- Computing Resources: $10,000
- Software Tools: $5,000

Deployment (Days 68-85):

- Personnel Costs (GUI Developers): $40,000

- Computing Resources: $15,000
- Software Tools: $10,000

Maintenance (Days 86-120):

- Personnel Costs (Data Scientists): $40,000
- Personnel Costs (QA Analysts): $20,000
- Computing Resources: $15,000
- Software Tools: $10,000

Total Budget Allocation = $195,000

3.3. Project Tracking Plan

3.3.1. Requirements Management

Measuring, Reporting, and Controlling Changes:

- Changes to project requirements will be documented using a standardized change request form.
- The change request form will include details such as the nature of the change, the reason for the change, the impact on project scope, schedule, budget, resources, and risk factors.
- All change requests will be reviewed by the Change Control Board (CCB) for approval or rejection.
- Approved changes will be documented, and their impacts on project elements will be assessed.

Impact Assessment:

- Impact assessments will be conducted to evaluate the effects of requirements changes on product scope and quality.
- The project team will assess how changes impact the project schedule, budget, resource allocation, and risk factors.
- Impact assessments will include a traceability analysis to understand the relationships between requirements and project components.

Change Control Procedures:

- The project will establish a Change Control Board (CCB) responsible for reviewing and approving changes.
- The CCB will consist of key stakeholders, including project managers, data scientists, and relevant experts.
- Change control procedures will be followed rigorously to ensure that only approved changes are incorporated into the project.

Traceability, Prototyping, Modeling, Impact Analysis, and Reviews:

- Traceability will be maintained to track the origin and evolution of requirements throughout the project.

- Prototyping and modeling techniques will be employed to visualize and validate requirements.
- Impact analysis will be conducted to understand the consequences of proposed changes.
- Regular reviews of requirements will be conducted to ensure alignment with project goals.

3.3.2.Schedule Control

Measuring Progress:

- Progress will be measured at major and minor project milestones to assess how well the project is adhering to the planned schedule.
- Actual progress will be compared to the planned progress to identify any deviations.

Corrective Action:

- If actual progress deviates from the planned schedule, corrective actions will be implemented to realign the project with the schedule.

Methods and Tools:

- Schedule progress will be tracked using project management software and tools.
- Gantt charts, milestone charts, and project timelines will be employed to visualize and monitor schedule progress.

Objective Criteria:

- Objective criteria will be defined to assess the scope and quality of work completed at each milestone.
- These criteria will provide a basis for evaluating the achievement of schedule milestones.

3.3.3.Budget Control

Measuring Costs:

- The cost of work completed will be measured at various stages of the project.
- This will involve tracking the actual expenses associated with personnel, computing resources, software tools, travel, meetings, testing facilities, and administrative support.

Comparing Actual Costs:

- The actual project costs will be compared to the planned and budgeted costs to identify any variances.
- Variances will be analyzed to determine the reasons behind cost deviations.

Corrective Action:

- If the actual cost deviates from the budgeted cost, corrective actions will be taken to bring costs in line with the budget.

Cost Reporting Schedule:

- Cost reporting will be done at regular intervals in the project schedule, typically aligned with major milestones.

Methods and Tools:

- Project management software and financial tools will be used to track project costs.
- Earned value tracking will be employed to report on budget and schedule performance.

Objective Indicators:

- Schedule milestones and objective indicators will be used to assess the scope and quality of work completed at each stage of the project..

3.3.4.Quality Control

- Measuring Quality: Quality will be measured through various checkpoints during the project to ensure that work and deliverables meet predefined quality standards.
- Quality Control Processes: Quality control will include quality assurance activities, verification, validation, joint reviews, audits, and process assessments to maintain the desired quality standards.

3.3.5.Reporting

- Reporting Mechanisms: Project status will be communicated through regular status reports, meetings, and documentation.
- Report Formats: Reports will follow a standardized format, detailing the status of requirements, schedule, budget, and quality.
- Information Flows: Information will flow within the project team and to external stakeholders as needed to ensure project transparency and alignment with objectives.
- Communication Tools: Project management software and standard communication tools will be used for effective reporting.
- Frequency: Reporting frequency will be determined by project milestones, criticality, risks, and visibility, ensuring that stakeholders are informed appropriately throughout the project lifecycle.

3.3.6.Project Metrics

Data Quality Metrics:

- Data completeness: Percentage of missing values in the dataset.
- Data accuracy: Percentage of erroneous or inconsistent data.
- Data balance: Distribution of melanoma and non-melanoma samples in the dataset.

Model Performance Metrics:

- Accuracy: The percentage of correctly classified skin lesions.
- Precision: The ratio of true positive predictions to the total positive predictions.
- Recall: The ratio of true positive predictions to the total actual positives.
- F1-Score: The harmonic mean of precision and recall, balancing false positives and false negatives.
- ROC-AUC: Receiver Operating Characteristic - Area Under the Curve for model discrimination.
- Confusion matrix: A matrix showing true positives, true negatives, false positives, and false negatives.

Resource Utilization Metrics:

- Resource allocation: Monitoring the allocation of data scientists, GUI developers, QA analysts, and other team members.
- Budget tracking: Monitoring project expenses against the allocated budget.

Schedule Metrics:

- Milestone completion: Tracking the completion of major project milestones within the defined schedule.

Quality Control Metrics:

- Defect density: The number of defects or issues identified during testing and validation.
- Review findings: Metrics related to the number and severity of issues found during code reviews or quality audits.

Data Processing Metrics:

- Data preprocessing time: Time taken to clean and prepare the dataset for analysis.
- Feature extraction time: Time taken to extract relevant features from the data.
- Model training time: Time taken to train machine learning models.

User Acceptance Metrics:

- User satisfaction: Feedback from stakeholders and end-users on the usability and effectiveness of the melanoma diagnosis system.

3.4. Risk Management Plan

Risk Identification and Assessment:

- Data Quality and Availability: There is a risk of poor-quality or insufficient data for training the machine learning models, which could impact the accuracy of melanoma diagnosis.

- Model Performance: The machine learning models may not perform as expected, leading to false positives or false negatives in diagnosis.
- Resource Constraints: Insufficient skilled resources, such as data scientists or GUI developers, could delay project progress.
- Technological Challenges: Complex algorithms and technologies used in feature extraction and machine learning may pose challenges during implementation.
- Scope Creep: Changes in project scope or requirements may affect project timelines and budget.
- Data Privacy and Security: Handling sensitive patient data raises risks related to data privacy and security compliance.

Risk Mitigation Strategies:

- Data Quality Assurance: Implement data preprocessing techniques and quality checks to ensure high-quality data.
- Model Validation: Perform rigorous validation and testing of machine learning models using cross-validation and independent datasets.
- Resource Planning: Continuously monitor resource utilization and have contingency plans for resource shortages.
- Technology Assessment: Regularly assess the latest technologies and ensure the team is up-to-date with advancements.
- Change Control: Implement a change control process to manage scope changes effectively.
- Data Security Measures: Adhere to data privacy regulations and implement robust security measures for patient data.

Contingency Planning:

- Data Backup: Regularly back up project data to prevent data loss.
- Resource Scaling: Consider hiring additional resources or outsourcing if resource constraints become critical.
- Technical Support: Have access to experts who can provide technical guidance in case of challenges.

Risk Tracking and Reporting:

Risks will be continuously monitored and tracked using a risk register. Risk status will be reported in regular project status meetings.

Organizational Responsibilities:

- The project manager is responsible for overall risk management.
- Data scientists and developers will assist in risk assessment related to their respective areas.
- The project team will collaborate to develop mitigation strategies.

Communication:

Risk status and mitigation plans will be communicated to stakeholders through regular project status reports and meetings.

Impact Assessment:

Risks related to customer relationships, contractual issues, technological challenges, project size, development environment, personnel, schedule, budget, and customer acceptance will be assessed for their potential impact on the project and addressed accordingly.

3.5. Project Closeout Plan

Staff Transition:

- Assess team skills and match them with suitable roles or projects.
- Ensure smooth transition of team members to new assignments.

Archiving Materials:

- Safely store project-related documents and data.
- Comply with data retention regulations.
- Appoint responsible personnel for archive management.

Metrics Capture:

- Document project metrics in the organization's database.
- Use standardized templates for consistent reporting.
- Enable access to authorized personnel for future reference.

Debriefing Sessions:

- Conduct post-project debriefings involving project personnel and stakeholders.
- Gather feedback and lessons learned.
- Identify successful strategies and areas for improvement.

Final Report:

- Create a comprehensive report summarizing project objectives, metrics, lessons learned, and recommendations.
- Promote knowledge sharing and recognition of team contributions.

Lessons Learned:

- Analyze lessons learned for best practices.
- Document actions and recommendations for future projects.

Responsibilities:

- Project manager oversees closeout activities.
- Team members actively participate in debriefings and contribute to lessons learned.

4. Technical Process Plans

4.1. Process Model

- In our project, we will follow a sequential process model, beginning with requirement gathering and ending with maintenance.
- Information will flow between phases, with each phase feeding into the next.
- Work products such as project documentation and code will be generated at specific milestones.
- Reviews will be conducted at the end of each phase.
- Milestones will include the completion of project documentation and the successful implementation of key features.
- We will establish baselines for project scope and code quality.
- The primary project deliverable is a functional machine learning pipeline.
- Approvals will be required before moving from one phase to another.
- We will tailor our organization's standard process model to suit the project's unique requirements.

4.2. Methods, Tools, and Techniques

Development Methodologies:

- Agile Approach: The project will follow an agile methodology to ensure flexibility and adaptability throughout the development process. Agile practices, including iterative development and continuous feedback, will be integral to the project's success.

Programming Languages:

- Python: Python will serve as the primary programming language for implementing machine learning algorithms, data preprocessing, and analysis. Its extensive ecosystem of libraries makes it ideal for scientific computing and data manipulation.

Processes, Tools, Techniques:

- Machine Learning Libraries: The project will leverage well-established machine learning libraries such as scikit-learn and TensorFlow for model development, training, and evaluation.
- Version Control: Git will be used for version control to track changes in code, collaborate among team members, and ensure code integrity.
- Data Preprocessing Techniques: Standard data preprocessing techniques, including feature scaling, dimensionality reduction, and data augmentation, will be applied to enhance model performance.

- Model Evaluation: Cross-validation and relevant evaluation metrics (e.g., accuracy, precision, recall, F1-score) will be employed to assess model performance and generalization.

Technical Standards:

- The project will adhere to industry best practices and standards for machine learning and software development. This includes maintaining clean and well-documented code, following coding conventions, and ensuring reproducibility of experiments.

Quality Control:

- Code Reviews: Regular code reviews will be conducted to maintain code quality, identify potential issues, and share knowledge among team members.
- Automated Testing: Automated unit tests and integration tests will be implemented to validate the correctness of code changes and prevent regressions.

4.3. Infrastructure

This infrastructure plan outlines the key components required for the project's success:

Hardware and Software:

- Hardware: The project will require a dedicated server or cloud computing resources with sufficient computational power and memory to train and run machine learning models. Additionally, individual workstations for project team members will be equipped with suitable hardware specifications.
- Operating System: The servers and workstations will use a reliable and compatible operating system, such as Linux or Windows, depending on the software stack and tools used.
- Network: A secure and high-speed local area network (LAN) will facilitate seamless communication and data transfer among team members. Internet access is necessary for research, updates, and collaboration.
- Software Tools: The project will utilize a variety of software tools, including Python for machine learning development, libraries such as TensorFlow and scikit-learn, integrated development environments (IDEs), and version control systems like Git for code management.

Development Environment:

- Integrated Development Environment (IDE): Team members will use IDEs like Jupyter Notebook or PyCharm for coding, experimenting with machine learning models, and data analysis.
- Data Storage: The project will require a secure and scalable data storage solution to manage the large datasets used for training and testing the models.

- Collaboration Tools: Tools like Slack or Microsoft Teams will facilitate real-time communication and collaboration among team members.

Office Space and Facilities:

- A designated office space or workspace within the organization's premises will be provided for the project team members to collaborate effectively.
- Desks, chairs, and office supplies will be available to create a productive work environment.

Security Measures:

- Physical and digital security measures will be in place to protect sensitive data, source code, and project-related information.
- Access controls and encryption will be implemented as needed.

Administrative Personnel and Support:

- Administrative staff will assist with procurement, resource allocation, and project logistics.
- Janitorial services will maintain a clean and comfortable work environment.

Compliance and Standards:

The project infrastructure will adhere to relevant industry standards and compliance requirements, especially those related to data privacy and security.

Resource Monitoring:

Continuous monitoring of hardware and software resources will be in place to ensure optimal performance and address any issues promptly.

4.4. Product Acceptance

- Customer acceptance will be based on predefined objective criteria, including model accuracy and performance metrics.
- A formal agreement, signed by both our IT organization and the customer, will outline acceptance criteria.
- Acceptance will require successful testing, model validation, and customer review.
- We will use a combination of testing, demonstration, and validation processes to ensure deliverable acceptance.

5. Supporting Process Plans

5.1. Configuration Management

Methods for Configuration Management:

- Identification: We'll uniquely label datasets, codebase, and model versions.
- Control: Changes require documentation and review by a change control board.
- Status Accounting: We'll log and track changes' impact on the project.
- Evaluation: Proposed changes will undergo rigorous impact assessments.
- Release Management: Procedures will notify stakeholders of baseline changes.

Processes for Configuration Management:

- Configuration Identification: Assigning unique identifiers to project components, including datasets, codebase, and model versions.
- Configuration Control: Logging and reviewing change requests by the Change Control Board for approval or rejection.
- Status Accounting: Maintaining a status log to track changes and their impact on the project.
- Evaluation: Assessing change impacts to make informed decisions.
- Release Management: Implementing procedures to notify stakeholders when baselines are established or modified.

Automated Tools: Git will be the primary version control system for managing project configurations.

5.2. Verification and Validation

- Scope: Verification and validation will encompass the entire project lifecycle, from data preprocessing to model development and testing.
- Organizational Relationships: Development and verification/validation activities will be performed by separate teams to maintain independence.
- Verification Techniques: Traceability, milestone reviews, peer reviews.
- Validation Techniques: Testing, demonstration, analysis.
- Automated Tools: Libraries such as scikit-learn and TensorFlow will be used for verification and validation tasks.

5.3. Documentation

- Documentation Plans: A comprehensive list of project documents will include project plans, design specifications, code documentation, and reports.

- Responsibilities: Clear roles for document preparation and review.
- Document Information: Detailed templates, review schedules, and distribution lists.

5.4. Quality Assurance

- Analysis: Regular analysis of project components, including code and datasets, to identify areas for improvement.
- Inspection: Detailed inspections of code, model implementations, and documentation to maintain quality.
- Review: Conducting peer reviews to assess the quality of code, algorithms, and model evaluation.
- Audit: Regular quality audits to ensure adherence to project standards and guidelines.
- Assessment: Assessing the overall quality and compliance of the project against predefined criteria.

5.5. Reviews and Audits

Schedule:

- Regular project reviews will be conducted at key milestones:
- Initial model development and training completion
- Web interface development and deployment
- Integration into the healthcare environment
- Final project delivery

Resources:

- Project reviews and audits will involve the following stakeholders:
- Project Team: Developers, data scientists, QA analysts
- Management: Project managers, team leads
- Quality Assurance Team: QA analysts
- Customer Representatives: Medical professionals and stakeholders
- External Auditors (if required)

Processes and Procedures:

- Joint Customer Project Reviews: These reviews will involve customer representatives, medical professionals, and project team members to ensure alignment with medical requirements and expectations.

- Management Progress Reviews: Periodic progress reviews with project managers and team leads to assess project health, resource allocation, and schedule adherence.
- Developer Peer Reviews: Internal code reviews conducted by team members to ensure code quality, maintainability, and adherence to coding standards.
- Quality Assurance Audits: QA analysts will perform audits to check for adherence to quality standards and test cases.
- Customer Conducted Reviews and Audits: Customers may conduct independent reviews to assess the system's compliance with medical regulations and standards.

External Agencies:

Health regulatory authorities (if applicable) may be involved in reviewing and approving the final project deliverable, especially when the system is integrated into the healthcare environment. Compliance with regulatory standards is a critical aspect of this project.

5.6. Problem Resolution

- Resources: Development Team, Configuration Management, Change Control Board, Verification and Validation, Quality Assurance.
- Methods and Tools: Issue tracking system, root cause analysis, prioritization, resolution, and separate effort tracking.

Procedures:

- Problem Identification: Stakeholders report issues.
- Problem Analysis: Teams analyze causes and severity.
- Prioritization: Change Control Board ranks issues.
- Resolution: Developers fix, QA verifies.
- Communication: Stakeholders stay informed.
- Documentation: Records maintained for auditing.

5.7. Subcontractor Management

Selection Criteria:

- Expertise in machine learning and data analysis.
- Proven track record in healthcare data projects.
- Compliance with project schedule and budget.

Management Plan for Each Subcontract:

- Requirements Management: Clearly define subcontractor responsibilities.

- Monitoring of Technical Progress: Regular progress meetings and status reports.
- Schedule and Budget Control: Subcontractor adheres to project timelines and budget.
- Product Acceptance Criteria: Define acceptance criteria in the subcontract agreement.
- Risk Management Procedures: Subcontractor shares and mitigates project risks.
- Additional Topics: Any unique requirements for specific subcontracts.
- Reference: Official subcontract document with contact information.

5.8. Process Improvement

- Assessment and Feedback: Regularly assess project performance and gather feedback from the team and stakeholders to identify areas for improvement.
- Improvement Initiatives: Prioritize and implement improvements tailored to the project's unique needs and challenges.
- Documentation: Maintain records of project-specific process enhancements and share lessons learned within the team.
- Continuous Enhancement: Foster a culture of ongoing improvement, adapting processes as needed to optimize project outcomes.
- Collaboration: Work closely with the Problem Resolution process to address project issues and root causes effectively.

6. Additional Plans

- Safety and Privacy Plan: Ensure data security and patient privacy compliance.
- User Training Plan: Prepare medical staff to use the system effectively.
- Integration Plan: Smoothly integrate the system with existing healthcare infrastructure.
- Data Conversion Plan: Manage historical data migration, if needed.
- System Transition Plan: Plan for a seamless transition to live deployment.
- Product Maintenance Plan: Address ongoing software updates and bug fixes.
- Product Support Plan: Provide user support and issue resolution services.

LAB-6

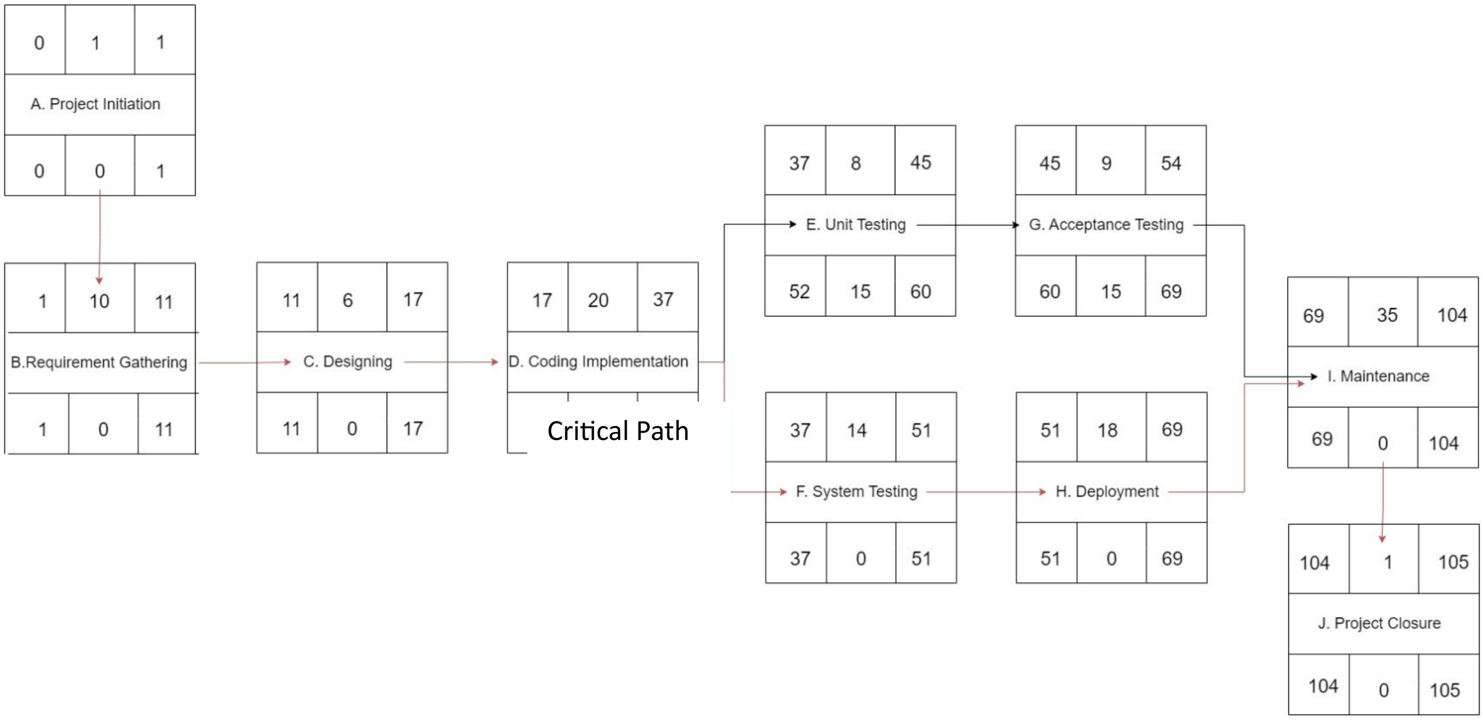
AIM

To construct Activity on Node diagram for the project: Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

Theory

Activity on Node (AoN) is a project management and scheduling technique used in the field of project management and network analysis. It is often associated with the Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM), which are used to plan, schedule, and manage complex projects.

Activity on Node Chart



Advantages of Activity on Node (AoN)

- **Visual Representation:** AoN diagrams provide a visual representation of the project schedule, making it easier for project managers and team members to understand the sequence of activities and their interdependencies. This visual clarity can help in better communication and coordination among team members.
- **Critical Path Identification**: AoN allows for the clear identification of the critical path, which is the longest path through the project network and determines the minimum time required to complete the project. Knowing the critical path helps in focusing resources and attention on activities that are most critical to the project's success.
- **Resource Allocation:** By having a clear understanding of activity durations and dependencies, project managers can allocate resources more effectively. They can ensure that resources are available when needed to complete critical tasks on time, avoiding bottlenecks and resource conflicts.
- **Time Management**: AoN diagrams provide a visual timeline of the project, making it easier to manage project timelines and deadlines. Project managers can track progress, monitor delays, and take corrective actions as needed to keep the project on schedule.
- **Risk Analysis**: AoN diagrams can be used to perform "what-if" analysis and assess the impact of changes or delays on the project schedule. This helps in proactive risk management and allows project managers to make informed decisions to mitigate risks.
- **Flexibility:** AoN is a flexible scheduling technique that can accommodate changes and adjustments to the project plan. If there are delays or changes in project scope, project managers can easily update the AoN diagram to reflect the new schedule.

LAB-8

AIM

To draw Workflow diagram, Slip Chart, Ball Diagram and Timeline Chart for the project: Hybrid Feature Extraction Using Transfer Learning and Machine Learning for Melanoma Diagnosis

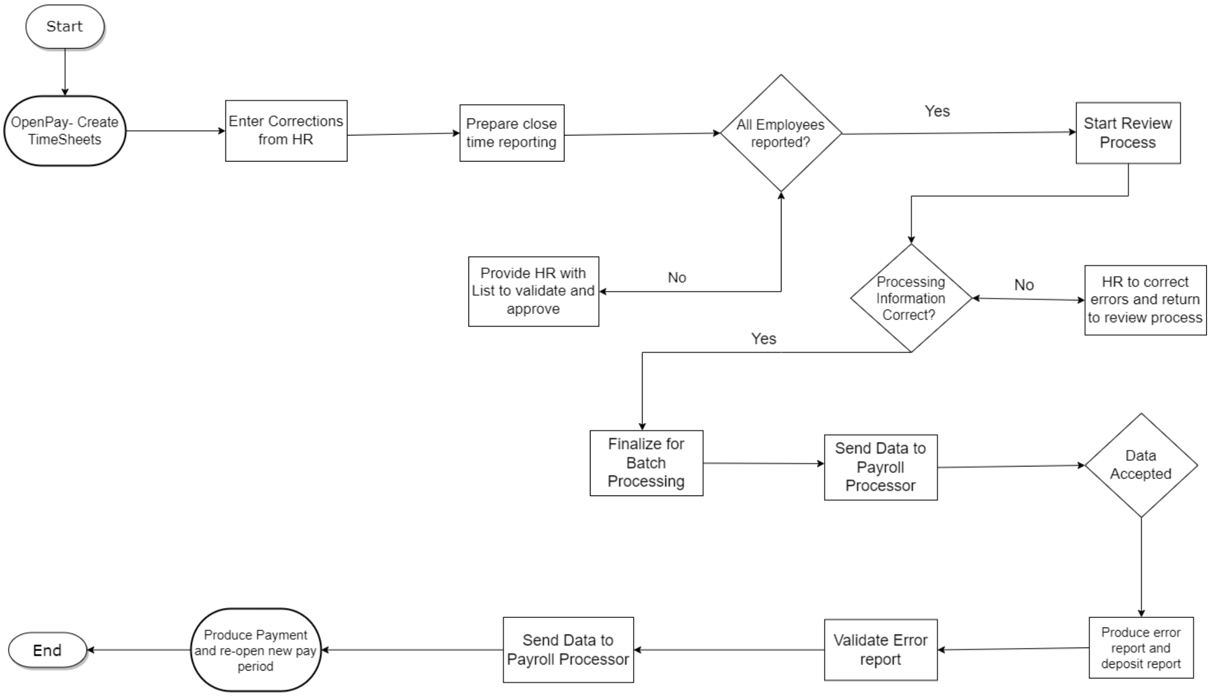
Theory

A **workflow diagram**, also known as a workflow chart, is a visual representation of a business process, project, or job in the form of a flowchart. It provides a graphic overview of the process, showing step by step how work is completed from start to finish, and who is responsible for work at each point in the process.

Workflow diagrams use standardized symbols and shapes to depict the various tasks and steps needed to complete the process. They are commonly used for project planning, business process mapping, and business process modelling, but can have other applications in industries like manufacturing and engineering.

Workflow diagrams are beneficial to project management because they help team members better understand the task sequences in which they're involved, create better communication between departments, and give teams a firm grasp on what they have to do. Workflow diagrams can also be used to identify and fix weak points within a long-standing process or workflow, and to isolate and repair inefficiencies and eliminate roadblocks.

Workflow Diagram


Start
OpenPay- Create TimeSheets
Enter Corrections from HR
Prepare close time reporting
All Employees reported?
Yes
Start Review Process
No
Provide HR with List to validate and approve
Processing Information Correct?
No
HR to correct errors and return to review process
Yes
Finalize for Batch Processing
Send Data to Payroll Processor
Data Accepted
Produce error report and deposit report
Validate Error report
Send Data to Payroll Processor
Produce Payment and re-open new pay period
End

Advantages of Workflow Diagram

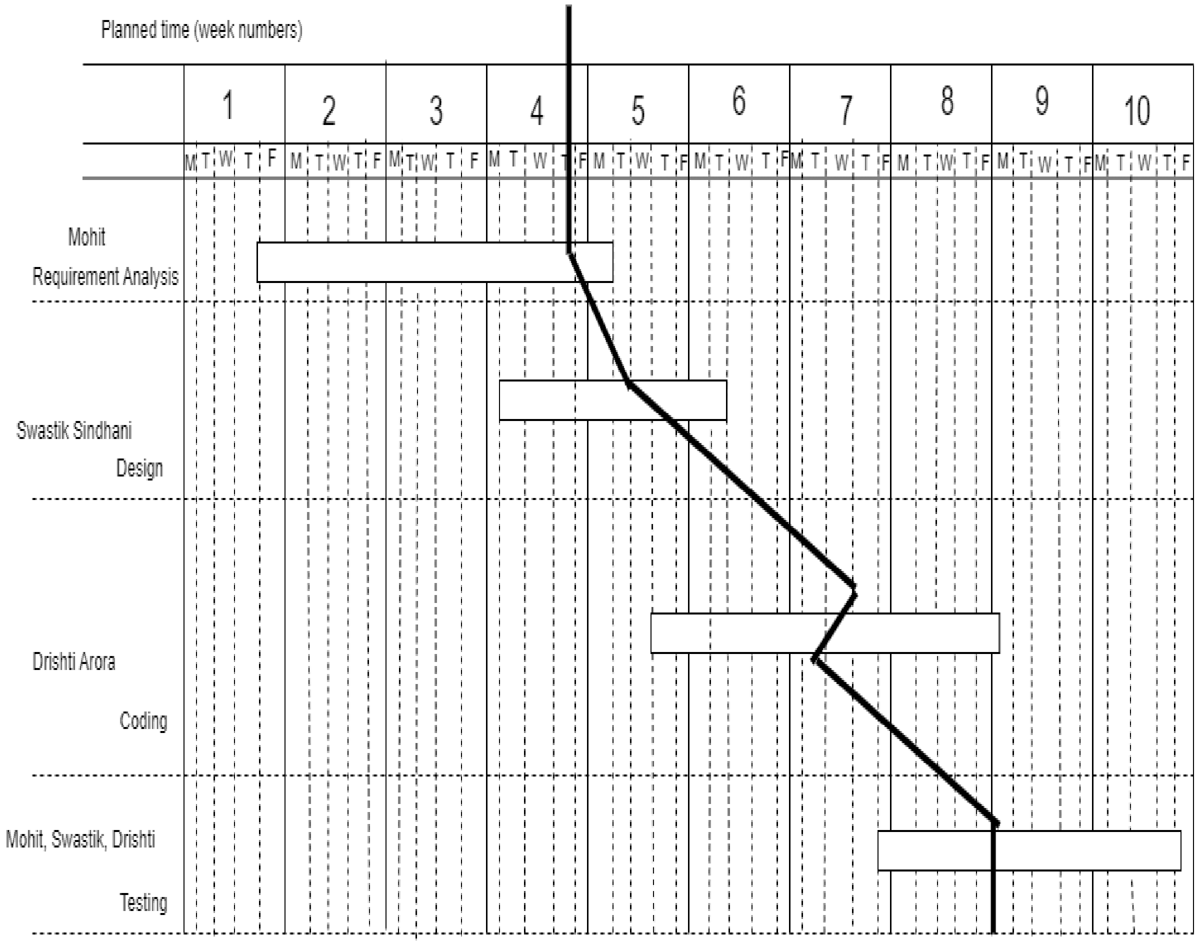
- **Visual Representation**: Workflow diagrams provide a visual representation of the project's processes and tasks. This visual clarity makes it easier for team members and stakeholders to understand the project's flow and structure.
- **Process Clarity**: They help clarify the sequence of tasks and activities required to complete the project, making it easier for team members to know what they need to do and when.
- **Identifying Dependencies**: Workflow diagrams can highlight dependencies between tasks and processes. This allows project managers to identify critical path tasks and ensure that they are on track to meet project deadlines.
- **Resource Allocation**: By mapping out the workflow, you can allocate resources (e.g., team members, equipment, software) more effectively. This helps in optimizing resource utilization and prevents overloading specific team members.
- **Risk Management**: Identifying potential bottlenecks or points of failure in the workflow helps in risk assessment and management. You can proactively address issues that may impede project progress.
- **Communication**: Workflow diagrams serve as a communication tool among project stakeholders. They provide a common language and visual reference that can be shared with clients, team members, and other relevant parties.
- **Process Improvement:** As the project progresses, workflow diagrams can be updated to reflect any changes or improvements in the processes. This allows for continuous process improvement throughout the project lifecycle.
- **Project Planning**: They are useful during the project planning phase. Project managers can use workflow diagrams to create project schedules and allocate resources effectively.
- **Quality Assurance**: Workflow diagrams can include checkpoints and quality assurance steps, ensuring that each phase of the project adheres to defined quality standards.
- **Documentation**: Workflow diagrams serve as documentation of the project's processes and can be valuable for future reference, audits, or knowledge transfer when new team members join the project.
- **Training**: They can be used for training purposes, helping new team members understand the project's workflow quickly and efficiently.
- **Integration**: Workflow diagrams can be integrated with project management software and tools, streamlining project tracking and reporting.
- **Change Management**: When changes or modifications to the project scope occur, workflow diagrams can be updated to reflect these changes, helping everyone understand the new direction of the project.

SLIP CHART

Slip chart- is a type of visual progress chart used in project management to show the overall progress of a given project over time. Slip charts are a line graph that plots out tasks, milestones, and events against time, allowing project managers to see which tasks have been completed, if any tasks are slipping behind, or if any tasks are ahead of schedule.

Slip charts are a visual indication of activities that are not progressing to schedule. They are an alternative view of a Gantt chart by providing a visual indication of those activities which are not on schedule. Slip charts are a simple but effective progress report where milestones are plotted on a grid to show when they are scheduled to occur. The more the slip line bends, the greater the variation from the plan.

Additional slip lines can be included at regular intervals, and as they build up, the project manager will gain an idea as to whether the project is improving or not. Slip charts are also used in statistical process control (SPM) to represent how often a process produces parts that fail to meet certain quality standards.



Advantages of Slip Chart

- **Visual Representation**: Slip charts provide a clear visual representation of data trends over time. This makes it easier for individuals to quickly identify patterns, variations, and trends in the data.
- **Identify Process Changes**: Slip charts are particularly useful for detecting changes or shifts in a process. Sudden shifts or changes in data patterns can indicate issues or improvements in the process, which can be addressed or capitalized upon, respectively.

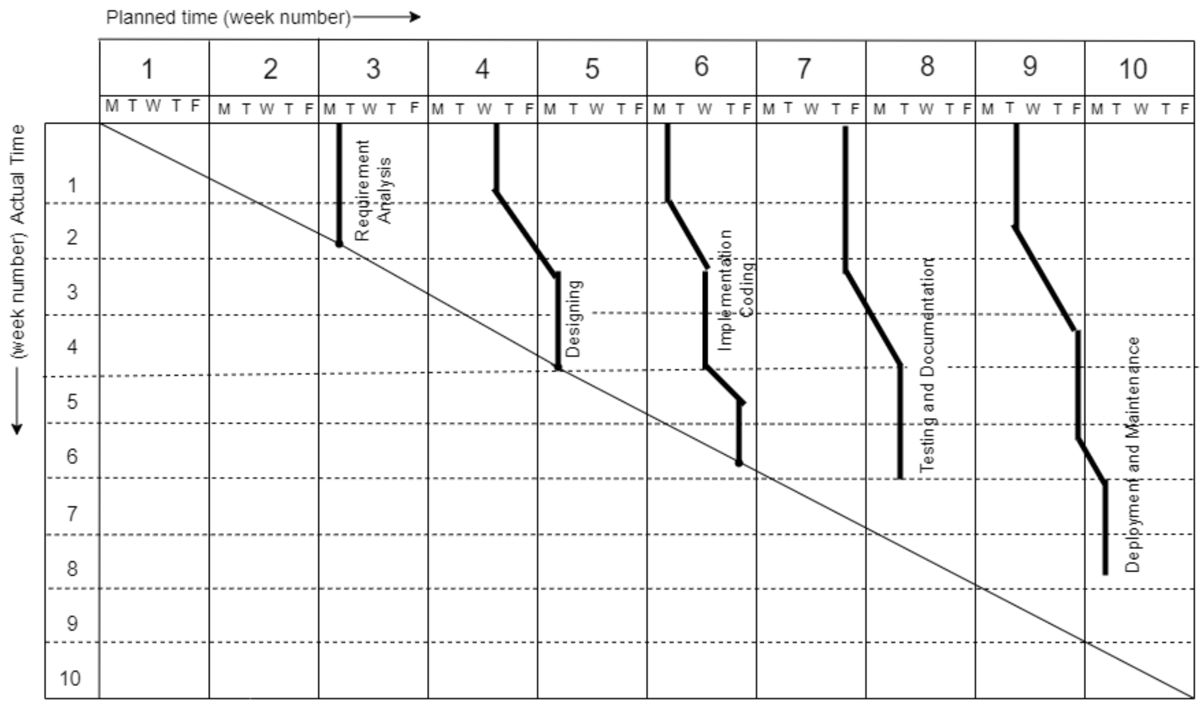
- **Anomaly Detection**: They help in identifying anomalies or outliers in the data. Anomalies may be indicative of defects or irregularities in the process that need attention.
- **Trend Analysis**: Slip charts allow for the analysis of long-term trends. By tracking data over time, it becomes easier to assess whether process improvements or changes have had a lasting impact on performance.
- **Process Control**: Slip charts are often used in statistical process control (SPC) to monitor and maintain control over a process. They help ensure that a process remains stable and within specified quality limits.
- **Data Validation**: They can be used as a tool to validate data quality. If data points consistently fall outside control limits or exhibit unusual patterns, it may indicate data collection or measurement issues.
- **Continuous Improvement**: Slip charts are integral to the continuous improvement process. By monitoring and analyzing data trends, organizations can identify opportunities for improvement and make data-driven decisions.
- **Easy Comparison**: They allow for easy comparison of multiple data sets or processes. This can be valuable when comparing the performance of different machines, shifts, or locations.
- **Decision Support**: Slip charts provide valuable information for decision-making. They help organizations decide when to take corrective actions, adjust processes, or make strategic decisions based on data trends.
- **Communication**: Slip charts can be used as a communication tool, helping team members and stakeholders understand the performance of a process at a glance.

TIMELINE CHART

It is a visual representation of a set of events or processes arranged chronologically. It is a simple yet powerful diagram that can be used to track projects to completion, illustrate historical events, or conceptualize event sequences or processes. Timeline charts typically include dates and descriptions, and some may also include images and headers.

There are several different types of timeline charts available, including event timelines, periodical history timelines, brand historical timelines, and AI growth timelines. Timeline charts are a valuable tool for organizations that need a concise way to visualize a process or event chronologically, and they can help manage complex tasks and ensure they're completed on time.

Some of the benefits of using timeline charts include providing a clear overview of what needs to be done and when, making it easier to allocate resources at the right time, and serving as a decision-making aid for current and future projects.

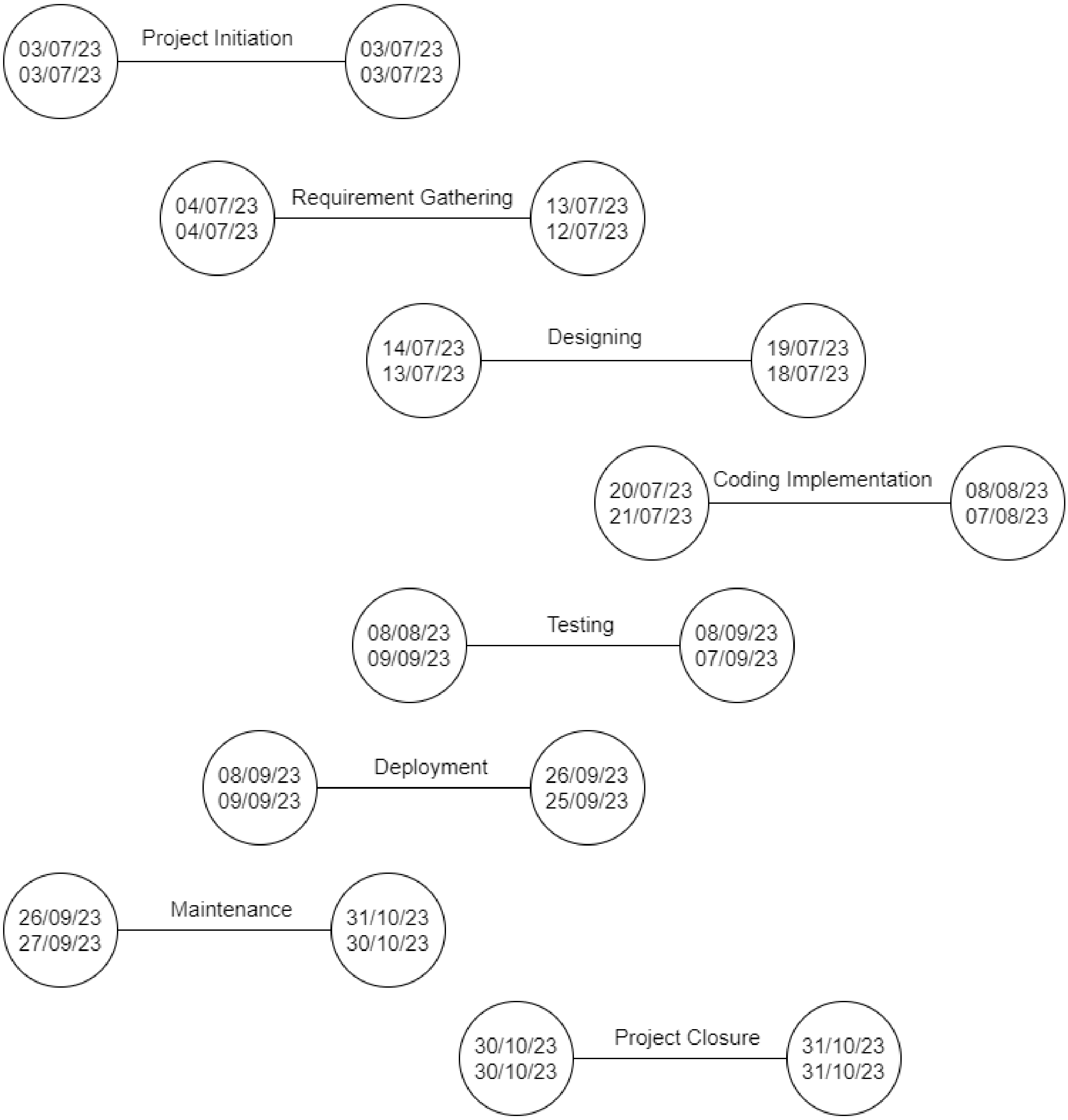


BALL CHART

Also known as a Harvey ball chart, is a type of chart that uses circles to represent data. It is a visual representation of qualitative information such as ease of use, efficiency, safety, taste, or quality. Ball charts circles contain original scheduled dates and the actual dates the activity/event took to complete.

They are also used in project management for project tracking, in lean manufacturing for value-stream mapping and continuous improvement tracking, and in business process modeling software for visualization. In a ball chart, the circles can indicate the start or end of a process, and they can be colored or filled to represent different values or categories. The chart consists of multiple circular shapes, with each colored quadrant representing a certain set of data.

The colored segment usually represents the data or criterion that has been achieved or completed, while the non-colored or blank segment represents the data or criterion to be achieved. Harvey balls are a powerful tool for visualizing different types of data, and they can be used to show different types of data in a single slide or page. However, they may not be the best choice for visualizing large amounts of data, as they can become confusing and difficult to distinguish between each set of data.


03/07/23
03/07/23
Project Initiation
03/07/23
03/07/23
04/07/23
04/07/23
Requirement Gathering
13/07/23
12/07/23
14/07/23
13/07/23
Designing
19/07/23
18/07/23
20/07/23
21/07/23
Coding Implementation
08/08/23
07/08/23
08/08/23
09/09/23
Testing
08/09/23
07/09/23
08/09/23
09/09/23
Deployment
26/09/23
25/09/23
26/09/23
27/09/23
Maintenance
31/10/23
30/10/23
30/10/23
30/10/23
Project Closure
31/10/23
31/10/23

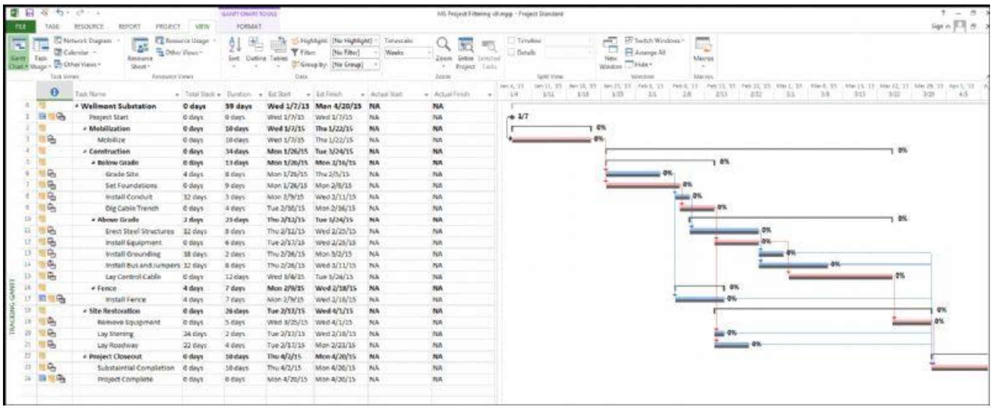
LAB-9

AIM

Introduction to Microsoft Project

Theory

Microsoft Project is a project management software product developed and sold by Microsoft. It is designed to assist project managers in developing plans, assigning resources to tasks, tracking progress, managing budgets, and analysing workloads. Microsoft Project is widely used in various industries for planning and managing projects of different sizes and complexities.



	Task Name	Total Slack	Duration	Est Start	Est Finish	Actual Start	Actual Finish
0	**Wellmont Substation**	**0 days**	**59 days**	**Wed 1/7/15**	**Mon 4/20/15**	**NA**	**NA**
1	Project Start	0 days	0 days	Wed 1/7/15	Wed 1/7/15	NA	NA
2	**Mobilization**	**0 days**	**10 days**	**Wed 1/7/15**	**Thu 1/22/15**	**NA**	**NA**
3	Mobilize	0 days	10 days	Wed 1/7/15	Thu 1/22/15	NA	NA
4	**Construction**	**0 days**	**34 days**	**Mon 1/26/15**	**Tue 3/24/15**	**NA**	**NA**
5	**Below Grade**	**0 days**	**13 days**	**Mon 1/26/15**	**Mon 2/16/15**	**NA**	**NA**
6	Grade Site	4 days	8 days	Mon 1/26/15	Thu 2/5/15	NA	NA
7	Set Foundations	0 days	9 days	Mon 1/26/15	Mon 2/9/15	NA	NA
8	Install Conduit	32 days	3 days	Mon 2/9/15	Wed 2/11/15	NA	NA
9	Dig Cable Trench	0 days	4 days	Tue 2/10/15	Mon 2/16/15	NA	NA
10	**Above Grade**	**2 days**	**23 days**	**Thu 2/12/15**	**Tue 3/24/15**	**NA**	**NA**
11	Erect Steel Structures	32 days	8 days	Thu 2/12/15	Wed 2/25/15	NA	NA
12	Install Equipment	0 days	6 days	Tue 2/17/15	Wed 2/25/15	NA	NA
13	Install Grounding	38 days	2 days	Thu 2/26/15	Mon 3/2/15	NA	NA
14	Install Bus and Jumpers	32 days	8 days	Thu 2/26/15	Wed 3/11/15	NA	NA
15	Lay Control Cable	0 days	12 days	Wed 3/4/15	Tue 3/24/15	NA	NA
16	**Fence**	**4 days**	**7 days**	**Mon 2/9/15**	**Wed 2/18/15**	**NA**	**NA**
17	Install Fence	4 days	7 days	Mon 2/9/15	Wed 2/18/15	NA	NA
18	**Site Restoration**	**0 days**	**26 days**	**Tue 2/17/15**	**Wed 4/1/15**	**NA**	**NA**
19	Remove Equipment	0 days	5 days	Wed 3/25/15	Wed 4/1/15	NA	NA
20	Lay Stoning	34 days	2 days	Tue 2/17/15	Wed 2/18/15	NA	NA
21	Lay Roadway	22 days	4 days	Tue 2/17/15	Mon 2/23/15	NA	NA
22	**Project Closeout**	**0 days**	**10 days**	**Thu 4/2/15**	**Mon 4/20/15**	**NA**	**NA**
23	Substantial Completion	0 days	10 days	Thu 4/2/15	Mon 4/20/15	NA	NA
24	Project Complete	0 days	0 days	Mon 4/20/15	Mon 4/20/15	NA	NA

Key Uses:

1. Project Planning:
    a. Task Management: Create, schedule, and manage tasks, subtasks, and milestones.
    b. Timeline View: Visualize project timelines with Gantt charts for better planning.
2. Resource Management:
    a. Resource Assignment: Assign resources to tasks and manage resource availability.
    b. Resource Leveling: Optimize resource utilization and resolve overallocations.
3. Budgeting and Cost Management:
    a. Cost Tracking: Track and manage project costs, including labor, materials, and other expenses.
    b. Budgeting: Set and monitor project budgets.
4. Reporting and Dashboards:
    a. Prebuilt Reports: Generate a variety of builtin reports for tracking project progress.
    b. Dashboards: Create custom dashboards for visualizing project data.

5. Integration with Office 365:
    a. Integration: Seamless integration with other Microsoft Office applications (Excel, Word, SharePoint).
    b. Communication: Facilitates communication and collaboration among project team members.
6. Customization:
    a. Templates: Use builtin templates for common project types or create custom templates.
    b. Custom Fields: Define custom fields to capture project-specific information.
7. Timeline View:
    a. Visual Planning: Timeline view for visually representing tasks and their dependencies.
    b. Drag and Drop Editing: Easily modify task durations and dependencies.
8. Collaboration:
    a. Online Access: With Microsoft 365 subscription, collaborate online and access projects from various devices.
    b. Sharing: Share project information with team members and stakeholders.
9. Risk Management:
    a. Risk Identification: Identify and document project risks.
    b. Risk Analysis: Analyze the impact of risks on project schedules and budgets.
10. Critical Path Analysis:
    a. Critical Path: Identify and analyze the critical path for the project.
    b. Dependency Management: Manage task dependencies and constraints.

Versions

- Microsoft Project Standard: Suitable for individual project managers and smaller teams.
- Microsoft Project Professional: Offers additional features like resource management, collaboration tools, and advanced reporting. It is designed for larger enterprises.
- Microsoft Project for the Web: A web-based version that integrates with Microsoft 365.

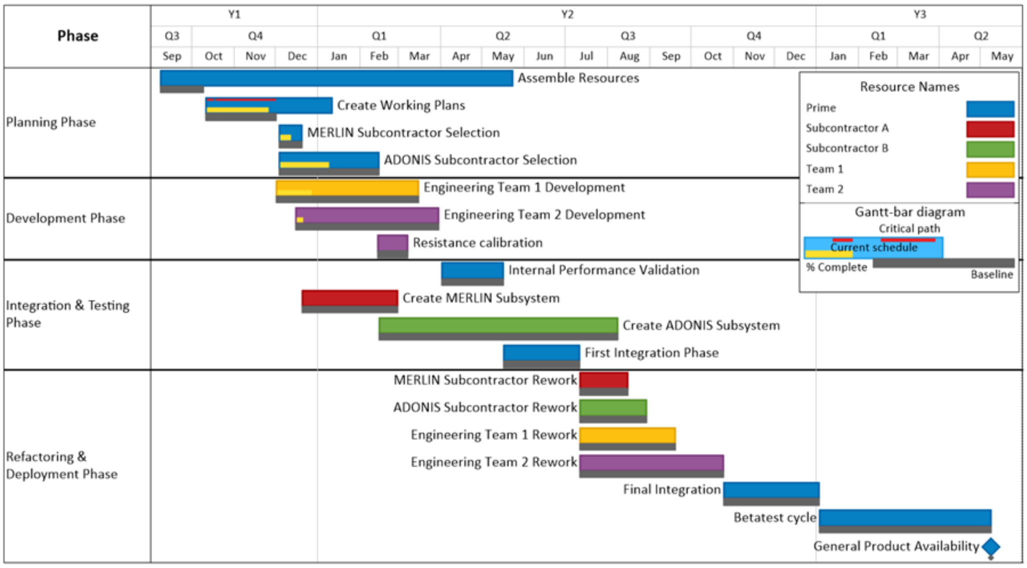
Limitations

- While Microsoft Project is a powerful tool, it may have a steeper learning curve for users new to project management software.
- Largescale projects with complex requirements may require additional integrations or a more specialized project management solution.
- It's important to note that the features and capabilities of Microsoft Project may evolve with new releases, so it's recommended to check the official Microsoft website or product documentation for the latest information.

Features

Microsoft Project offers a range of features to support project managers in planning, executing, and monitoring projects. Below are some key features of Microsoft Project:

a. Task Management:
    a. Create, organize, and manage tasks and subtasks.
    b. Define task dependencies and relationships.
b. Gantt Charts:
    a. Visualize project timelines using Gantt charts.
    b. Easily adjust task durations and dependencies.
c. Resource Management:
    a. Assign resources to tasks.
    b. Monitor resource availability and workload.



d. Timeline View:
    a. View and present project timelines using a graphical timeline view.
    b. Customize the appearance of the timeline.
e. Critical Path Analysis:
    a. Identify and analyze the critical path of the project.
    b. Manage task dependencies to optimize schedules.
f. Budgeting and Cost Management:
    a. Track and manage project costs.
    b. Set and monitor project budgets.
g. Reporting and Dashboards:
    a. Generate pre-built reports for tracking progress.
    b. Create custom dashboards for visualizing project data.
h. Customization:
    a. Use built-in templates for common project types.
    b. Define custom fields to capture project-specific information.
i. Risk Management:

  a. Identify and document project risks.
  b. Analyze the impact of risks on project schedules and budgets.

j. Task Tracking:
  a. Track task progress and completion.
  b. Update task status and mark milestones.

k. Resource Leveling:
  a. Optimize resource utilization.
  b. Resolve resource overallocations.

l. Microsoft Project for the Web:
  a. Web-based version that integrates with Microsoft 365.
  b. Access projects online and collaborate with team members.

m. Portfolio Management:
  a. Manage multiple projects within a portfolio.
  b. Analyze and prioritize projects based on organizational goals.

n. Scenarios and What-If Analysis:
  a. Perform what-if analysis to assess the impact of changes.
  b. Evaluate different project scenarios.

o. Task Calendars:
  a. Assign specific calendars to tasks for accurate scheduling.
  b. Define working and non-working hours for tasks.