Automated Loan Approval System

Group Name: CC11 – Topic 19

Course Name: INFO6007 Project Management in IT

Submitted To: Pasindu Marasinghe

Date: 18.05.2025

Contribution Summary

The project team lead should summarize the weekly contributions of each team member from Week 3 to Week 10. Each table outlines tasks completed by individual members.

Member 1 - Abraham Kuriakose

Week	Contribution Details
Week 3	Project Purpose and Project Description
Week 4	Project Justification
Week 5	Introduction & Industry Trends
Week 6	Project Overview & WBS Hierarchy
Week 7	Executive Summary & Project Objectives
Week 8	Cost Breakdown: List all Components
Week 9	Task List: Defining all Project Tasks
Week 10	Time Control Process Conclusion

Member 2 - Anyi Li

Week	Contribution Details
Week 3	Technical Decisions and Conflict Resolution
Week 4	Summary of Project Deliverables
Week 5	Gaps in Literature
Week 6	Milestones & WBS Dictionary
Week 7	-
Week 8	Contingency Fund
Week 9	Resource Allocation
Week 10	Conclude time control process

Member 3 – Nicholas Hadiwijaya

Week	Contribution Details
Week 3	Project objectives (scope, time, cost), Staffing decisions, Budget management & variance
Week 4	In scope and Out Scope
Week 5	Methodology, Comparative analysis, Conclusion
Week 6	Roles and responsibilities
Week 7	Tasks & Milestones, Timeline
Week 8	-
Week 9	Gantt chart
Week 10	Cost baseline, Variance analysis, Financial reports

Member 4 – Shawn Li

Week	Contribution Details
Week 3	High-level requirements, high-level risks
Week 4	Product Characteristics and Requirements
Week 5	Theoretical Background: Discuss relevant theories and models.
Week 6	1. Task Definition: Assign detailed descriptions to each task. 2. Task Dependencies: Identify tasks that are dependent on others.
Week 7	1. Budget Overview: Estimate costs for the project. 2. Progress Monitoring: Outline methods for tracking progress. 3. Approval & Signatures: Capture necessary approvals.
Week 8	1. Labor Costs: Estimate wages for team members. 2. Budget Approval: Define budget approval process.
Week 9	distribute resources
Week 10	Implement tracking mechanisms to monitor schedule adherence

Member 5 – Yash Sable

Week	Contribution Details
Week 3	Identifying- Stakeholders and Milestones
Week 4	Establishing Project Success Criteria
Week 5	Key Findings: Understanding key insights from previous research.
Week 6	1. Resource Allocation: Assign team members and required resources. 2. Review & Approval: Ensure stakeholders validate the structure.
Week 7	Role Assignment and Risk Management Planning
Week 8	1. Material Costs: Identify needed equipment and software 2. Overhead Costs: Including indirect expenses
Week 9	1. Estimated Durations: Calculating duration for each task 2. Dependencies: Identifying task relationships
Week 10	Time control processes and Budgeting and cost baseline

Table of Contents

1. Introduction

In today's rapidly evolving financial landscape, the integration of artificial intelligence into banking operations has become a critical enabler for efficiency, scalability, and customer satisfaction. One such application is the automated loan approval system — a technology-driven solution that enhances decision-making accuracy and speeds up loan processing workflows. Financial institutions now face increasing pressure to digitize their services, reduce manual bottlenecks, and adopt data-driven solutions that align with modern customer expectations and regulatory frameworks.

This report presents a comprehensive project plan for developing and deploying an AI-powered automated loan approval system. The aim is to streamline the traditionally time-consuming and resource-intensive loan processing lifecycle, while ensuring the system adheres to fairness, transparency, and compliance standards. The project encompasses all essential phases of IT project management, including scope definition, stakeholder analysis, scheduling, budgeting, risk assessment, and performance tracking. The proposed system will leverage machine learning algorithms to assess customer eligibility, enabling faster approvals and reducing the risk of human error.

Our project aligns closely with the core principles of the Project Management Body of Knowledge (PMBOK) and follows a structured development methodology. It incorporates agile practices to ensure adaptability and continuous feedback, alongside traditional waterfall techniques for documentation and compliance needs. Through this report, we aim to demonstrate not only the technical feasibility of the system but also the strategic value it brings to financial service operations.

2. Project Charter

2.1. Project Purpose of justification

The Automated Loan Approval System is a project which aims to help financial institutions make data-driven decisions regarding loan approvals. By analyzing historical loan data, the project can help identify risk factors, then reduce financial losses due to defaults. Therefore, it can help customers make decisions more quickly and assist some potential small businesses in obtaining more investment.

2.2. Project Description

This project follows a structured approach to analyzing SBA loan data and delivering a predictive model. Following are key phases:

- **Initiation:** Identifying project objectives, key stakeholders, and success criteria.
- **Planning:** Defining scope, deliverables, resource allocation, and risk management strategies.
- **Execution:** Conducting data collection, cleaning, exploratory analysis, and model development.
- **Monitoring & Controlling:** Evaluating model performance, validating findings, and refining based on feedback.
- **Integration & Deployment:** Integrate model to UI and deploy end-to-end system to production.
- **Maintenance & Evaluation:** Monitor system performance for feedback.

The primary deliverables include a project report detailing the analysis findings, risk assessment strategies, and a framework for decision-making support.

2.3. High-Level Requirements

In the Automated Loan Approval System project, the system needs to meet the following requirements. Firstly, all processes and required documents must be comprehensive and provide a solid basis for decision-making. Then, all collected user data must be properly hashed, securely stored, and protected from leaks. Otherwise, all loan applications should be categorized based on their risk level and amount. This system may focus solely on daily low-amount loan activities. At the same time, the system must ensure the efficiency and accuracy of the AI model to make sure that the approval results are reliable. Lastly, the system must comply with local policies and laws so that all ongoing activities are reasonable and legal.

2.4. High-Level Risks

The project meets many high-level risks. For example, data security and regulatory risks should be noticed first. Then, risks related to the reliability and bias of the AI model also need to be taken seriously. Lastly, risks associated with detecting loan scams, risks regarding the system's maintainability and performance and risks related to user trust and transparency cannot be ignored.

2.5. Project Objectives

- **Scope**
  - Create end-to-end AI system that can predict and determine bank customer loan eligibility based on customer history
  - Create separate UI for bank employees and bank clients to access the system
  - Success criteria: System achieves all functional requirements
  - Person approving: Project Sponsor
- **Time**
  - 1 month for planning and requirement collection
  - 2 weeks for data collection and processing
  - 1 month for model and UI development
  - 2 weeks for system integration, testing, and deployment
  - Success criteria: Project completed within time limit
  - Person approving: CIO
- **Cost**
  - $15000 infrastructure costs
  - $300000 labor costs
  - Success criteria: Project completed within budget
  - Person approving: Director of Finance

2.6. Milestones and Due Dates

Milestone	Due Dates
Milestone 1: Project Foundation Established	End of Week 2
Milestone 2: Data Pipeline Ready	Mid-Week 5

Milestone 3: System Core Completed	End of Week 9
Milestone 4: Deployment & Handover	End of Week 12

2.7. Stakeholders and Roles

Stakeholder(s)	Role
Project Sponsor/Client	Bank Organization
Project Manager	Nicholas Hadiwijaya
Supervisor	Shawn Li
Dev Team	Abraham Kuriakose Anyi Li Yash Sable Sam Bart John Digger Tim Williams

2.8. Project Manager Authority Level

- **Staffing decisions:** The project manager has the authority to appoint and approve staffing assignments for the project. Any changes to the staffing will need to be approved by the project manager.
- **Budget management and variance:** The project manager will be responsible for managing the budget spending. In case of any budget variance, the project manager will need to discuss this with the financial department for approval.

2.9. Technical Decisions

The Automated Loan Approval System will be developed as a microservice and integrated into the bank’s IT environment. Key technical implementations include:

- Tech stack:
    - Backend: Flask, Python
    - Frontend: Vue.js
    - Database: PostgreSQL
- Infrastructure:
    - Cloud provider: AWS
    - CI/CD: Jenkins
    - Version Control: Github

2.10. Conflict Resolution

Any conflicts arising between team members will be resolved within daily standups. A win-win solution is ideal, but if this is not possible, the project manager should intermediate between the members and choose the best available solution based on the severity of the situation.

3. Project Scope

3.1. Project Purpose

The project is essential to improving the efficiency, accuracy, and transparency of loan approval processes for small businesses. Currently, many loan applications are rejected due to a lack of clear evaluation criteria, while lenders struggle with minimizing financial risks and ensuring responsible lending.

The project provides insights into the key factors influencing loan approvals and defaults. These insights can help:

- **Increase approval accuracy** by identifying reliable indicators of business creditworthiness.
- **Enhance risk management** by reducing the chances of approving high-risk loans.
- **Improve process efficiency** by streamlining decision-making and reducing manual efforts.
- **Ensure fairness and transparency** by removing subjective biases from the approval process.

3.2. System Requirements

3.2.1. Functional Requirements

- **Data Collection & Integration:**
    - **Customer Data Ingestion:** Ability to integrate multiple data sources (credit bureaus, transaction databases, public records).
    - **Data Preprocessing:** Automated cleaning, normalization, and transformation of raw data into model-ready formats.
- **Risk Assessment Engine:**
    - **Credit Scoring**: Calculate credit scores using statistical or machine learning methods.
    - **Predictive Modelling:** Implement algorithms that predict loan default probabilities and categorize risk levels.
- **Decision Engine:**
    - **Automated Approvals/Rejections**: Based on thresholds set by risk models, the system should automatically approve, flag, or reject applications.
    - **Manual Review Interface:** Provide a user-friendly dashboard for human oversight and intervention when needed.
- R**eporting & Analytics:**

    - **Performance Metrics:** Track KPIs such as approval rates, default rates, and model accuracy.
    - Regulatory Reports: Generate compliance reports that detail the decision-making process and model performance.

3.2.2. **Non-Functional Requirements**

- **Security:**
    - **Data Encryption:** Use encryption both in transit and at rest to secure sensitive customer information.
    - **Access Controls:** Implement robust authentication and authorization mechanisms.
- **Scalability:**
    - **Horizontal Scaling:** Ensure the system can expand its capacity to process increasing numbers of loan applications.
    - **Cloud Compatibility:** Consider cloud-based deployments for flexibility and scalability.
- **Performance:**
    - **Low Latency:** Ensure real-time or near-real-time decision-making capabilities.
    - **High Availability:** Implement redundancy and failover strategies to ensure continuous operation.
- **Maintainability:**
    - **Modular Architecture:** Design the system so that components (data ingestion, model training, decision engine) can be updated or replaced independently.
    - **Monitoring and Logging:** Continuous monitoring of system performance and detailed logs for troubleshooting.

3.2.3. **Ethical and Regulatory Requirements**

- **Fair Lending Practices:**
    - **Bias Audits:** Regularly conduct audits to detect and address any bias in the model's decisions.
    - **Transparency to Customers:** Provide clear communication to customers about why a decision was made and what factors influenced their credit assessment.

- **Data Privacy:**
    - **Consent Management:** Ensure customers have provided explicit consent for their data to be used in automated decision-making.
    - **Anonymization Techniques:** Where possible, use data anonymization to protect individual privacy during model training.

3.3. Product Characteristics

3.3.1. Data-Driven Decision Making

- **Historical Data Utilization:** The system relies on customer history—including credit scores, repayment patterns, income data, and transaction records—to assess risk.
- **Predictive Analytics:** It uses machine learning models (e.g., regression, classification, and clustering algorithms) to forecast default probabilities and determine eligibility.
- **Continuous Learning:** Models can be retrained with new data to improve performance over time and adapt to changing economic conditions.

3.3.2. Automation and Efficiency

- **Real-Time Processing:** The system should process applications quickly, offering near-instant decisions through automated data ingestion and model inference.
- **Workflow Integration:** Seamless integration with existing banking systems (e.g., CRM, ERP) is essential to ensure smooth operations and minimal manual intervention.

3.3.3. Transparency and Explainability

- **Interpretability of AI Models**: Given regulatory and customer scrutiny, the system must offer explanations for its decisions. Techniques like feature importance analysis or decision trees can be used to clarify why a particular decision was made.
- **Audit Trails:** Detailed logs and traceability of decisions help in compliance and allow for independent reviews or audits.

3.3.4. Compliance and Security

- **Regulatory Compliance:** Adherence to data protection regulations (such as GDPR, CCPA) and financial standards is mandatory. The system must enforce data privacy and maintain secure data storage.
- **Bias Mitigation:** Built-in mechanisms to detect and correct biases in the training data or model outcomes are crucial to ensure fair lending practices.

3.3.5. Scalability and Reliability

- **Performance Under Load:** The system should scale to handle high volumes of applications without performance degradation.
- **Fault Tolerance:** It must have robust error handling and recovery mechanisms to maintain system availability and reliability.

3.4. Project Acceptance Criteria

3.4.1. Functional Acceptance Criteria

- **Accuracy of Loan Decisions:** The system should provide at least 90% accuracy in assessing loan eligibility compared to manual processing.
- **Processing Speed:** Loan applications should be processed within minutes, with at least 80% of applications receiving an automated decision.
- **User-Friendly Interface:** The system should be intuitive and accessible for customers and bank employees, ensuring a smooth and seamless experience.
- **Seamless Integration:** The system should successfully integrate with banking APIs, credit bureaus, and compliance databases without data inconsistencies.

3.4.2. Security & Compliance Acceptance Criteria

- **Data Security & Privacy:** The system must comply with Australian privacy laws (Privacy Act 1988 & APRA standards), ensuring encryption and secure data storage.
- **Fraud Detection & Risk Management:** The system should incorporate AI-based fraud detection with at least 95% effectiveness in identifying fraudulent applications.
- **Regulatory Compliance:** The solution must align with ASIC, APRA, and NCCP Act (National Consumer Credit Protection Act) guidelines.

3.4.3. Technical Performance Criteria

- **System Uptime & Reliability:** The system should maintain at least 99.5% uptime to ensure continuous availability for users.
- **Scalability:** The system should handle peak loads (e.g., processing 1,000+ applications per hour) without performance degradation.
- **Error Handling & Recovery:** The system should have built-in error-handling mechanisms to recover from failures without losing customer data.

3.4.4. Business & Customer Impact Criteria

- **Customer Satisfaction:** At least 85% of users should rate the system as "satisfactory" or above in usability and transparency.
- **Reduction in Operational Costs:** The system should reduce manual processing costs by at least 30%, increasing efficiency.
- **Compliance & Audit Readiness:** The system should generate comprehensive reports for audits and compliance reviews.

3.5. Project Scope

3.5.1. In Scope:

- Development of end-to-end system for predicting customer loan eligibility in scale, including:
  - Research, development, and training of AI model for predicting loan eligibility
  - End-to-end pipeline for preprocessing customer data and getting predictions from the model at scale
  - Development of system UI for bank employees and bank clients
- System documentation
- Post-development technical support for technical issues
- Training for bank employees on using the system

3.5.2. **Out Scope:**

- Any additional features beyond initial scope
- Post-launch feature updates

4. Literature Review

4.1. Introduction

Loans are an important financial service that supports businesses and individuals, but faces inefficiencies in approval due to outdated risk assessment models. AI-driven models improve efficiency, accuracy, and fairness by leveraging advanced data analytics and predictive capabilities. This literature review analyzes recent research (2022 and later) to identify theoretical foundations, recent trends and best practices for automated loan approval systems in the industry. Identifying current state-of-the-art systems, limitations, and risks will better help us in managing the scope and risks of our project to ensure success.

4.2. Theoretical Background

Automated loan approval systems rely on several theorems and models. The first is credit risk, which defines the risk lenders face when a borrower is unable to repay their debt and defaults. Traditionally, a borrower's historical financial data is condensed into a single numeric summary (such as the FICO score) that defines their predicted riskiness. This formed the basis for early automation efforts by translating human underwriting criteria into algorithmic rules.

As data availability and computational power increased, the field shifted to implement machine learning models in the process. Initially, simpler models such as logistic regression trained on large datasets were used. Later, more advanced machine learning models such as random forest, SVM, and XGBoost were adopted. These models have an advantage in being able to capture non-linear relationships and were able to make use of alternative non-traditional data sources for predictions (Mazoué, 2025).

When using AI models, ethicality and bias should also be considered. There are concerns about whether automated systems propagate human biases, especially on sensitive attributed such as race and gender. To improve transparency on the decision-making process, explainable AI techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are used, which allows regulators and applicants to understand how predictions are made.

4.3. Industry Trends

Financial institutions are increasingly using AI for lending-related tasks – 57.6%, 48.3%, and 42.4% of banks heavily use AI for fraud detection, risk assessment, and credit scoring respectively (Kingsley, 2022). These figures show the eager adoption of AI-based lending in the industry. The following are some case study examples of trends in the industry:

Alternative Credit Data: AI are reshaping traditional lending by powering the use of non-traditional alternative credit data, which allows financial institutions to serve previously "unscorable" populations.

- Kabbage, a digital-first lender from the U.S., uses a combination of traditional data (such as accounting details) and non-traditional data (such as social media profile, company location, and business industry) to assess creditworthiness for small business loans, totaling over 1.5 million data points (Nikolaienko, 2023). By using this massive amounts of data gave, Kabbage achieved unparalleled accuracy in predicting their clients creditworthiness.
- Tala, operating in emerging markets like Kenya and India, utilizes non-traditional data such as mobile device data (such as device ID, OS, and apps) and behavioral data (how the customer navigates the app) to assess credit risk (Tala, 2025). By leveraging AI models to analyze behavioral patterns, Tala has provided loan services to millions of people without formal banking histories, demonstrating how AI can bridge financial inclusion gaps.

Explainable AI (XAI): Given the black-box nature of AI models, there is increasingly more emphasis on model explainability and bias mitigation.

- Daltrics, a UK-based AI credit analytics platform, addresses these concerns by integrating SHAP values into its AI model (Daltrics, 2025). SHAP provides visualized explanations for each loan approval prediction, showing which features (e.g., total income, credit history) contributed to approval or rejection. This transparency supports compliance with GDPR and similar regulations, and builds trust with the platform’s clients.

4.4. Gaps in Literature

- Most studies are based on public datasets (such as Lending Club, German Credit) or single bank data, and lacks research on the generalization ability of models in different geographical regions and banking systems.
- Furthermore, there is little research on the transferability of models between countries with differences in laws and regulations or different credit evaluation standards.
- Most research focuses on the "automated approval model" itself, but lacks research on how the model can integrate with traditional manual approval

processes, such as semi-automatic approval systems, human-machine collaboration processes, etc.

- There is also a lack of research on tracking actual application impact (such as approval efficiency, customer satisfaction, default rate) after system deployment.

4.5. Methodology

There were several methodologies and criteria used when sourcing and evaluating literature to ensure relevance, accuracy and reliability. Many of the sources are research papers that have been peer-reviewed and are extremely reliable. For non-academic sources such as company websites and news articles, the CRAAP test was used to ensure that these sources were of good quality, and assessed based on their published date, author credentials, and purpose of writing. Keywords such as 'AI Loan approval models' and 'Automated loan approval systems' were used to filter relevant articles through google and google scholar. Furthermore, only articles published later than 2022 were considered, ensuring that the contents reflect recent advancements in the area.

4.6. Key Findings

- **Efficiency and Accuracy:** Automated systems speed up loan processing and improve accuracy, reducing costs and enhancing approval rates.
- **Cybersecurity and Compliance Risks:** Automated systems face threats like data breaches, requiring strong cybersecurity and compliance with regulations like APRA and ASIC.
- **Ethical Considerations and Bias Risks:** AI must be monitored for bias, ensuring fair lending practices and human oversight to avoid discrimination.
- **IT Governance and Data Privacy:** Data protection measures, such as encryption, are essential for safeguarding customer information and ensuring compliance.
- **Business and Customer Impact:** Automation improves customer satisfaction by providing faster approvals, while reducing operational costs for businesses.

4.7. Comparative Analysis

Traditional scoring systems rely heavily on historical repayment data, whereas modern automated loan approval systems can incorporate alternative data sources such as rental payments, spending patterns, and mobile data (Nikolaienko, 2023). These additional sources of data allow financial institutions to get more in-depth information on the customer's spending habits and ability, allowing them to better determine the customer's trustworthiness.

When comparing research of AI models for loan approval, ensemble models such as AdaBoos and XGBoost consistently achieve the best results. Muhammad et al. (2024) compared the performance of several AI models and found that XGBoost achieved the best accuracy of 99.74% on predicting loan approval, with AdaBoost trailing behind at 96.70% accuracy. Haque and Hassan (2025) did a similar test on a different dataset, achieving the best results on predicting loan approval using AdaBoost with 99.99% accuracy.

A significant drawback of ensemble models when compared to traditional scoring systems are their lack of transparency and bias. However, there are ways to mitigate this. Acharya et al. (2024) demonstrates that for XGBoost and LightGBM models, techniques such as SHAP and LIME can be used for visualizing how different attributes impact the model's prediction.

4.8. Conclusion

The adoption of AI models offers significant enhancements for the bank's loan system. Other financial institutes show a significant reduction in loan processing times after implementing AI-powered automated loan systems, and predictions have higher accuracy compared to traditional models through utilizing alternative credit data. However, bias and transparency are main points of concern, and techniques to explain model predictions such as SHAP should be used to provide model explainability.

Ensemble models provide the highest accuracy for predicting loan approval. The results show that different ensemble models can perform better depending on the dataset. Further tests should be done to determine which of these models would fit best for our use-case using our own historical data.

5. Work Breakdown Structure (WBS)

5.1. Section Overview

In this section we are going to build the work breakdown structure of this project, define tasks and the dependency relationships among them.

The table below outlines the tasks that are included or excluded from the project scope.

In-scope	Data collection from SBA loan records.
	Development of AI/ML models for approval prediction.
	Model evaluation using accuracy, fairness, and reliability metrics.
	Deployment of an end-to-end system using system for predicting loan eligibility
	Development of system UI for bank employees and bank clients
	Documentation and guideline for using the system
Out-scope	Direct integration with financial institutions.
	Real-time loan processing or decision-making.
	Handling of sensitive customer data beyond anonymized datasets.

5.2. WBS Hierarchy

The tasks required can be broken down into 5 phases: kickoff, data collection and processing, model development, end-to-end system implementation, and deployment and maintenance. The table below outlines the tasks required in each phase:

Phase	Tasks
1. Kickoff	1.1. Project initiation 1.2. Requirements gathering
2. Data Collection & Processing	1.2. Data collection 1.3. Data processing 1.4. Exploratory data analysis
3. Model Development	1.3. Model selection 1.4. Model training & tuning 1.5. Model evaluation

4. End-to-End System Implementation	1.4. UI/UX design 1.5. Backend integration 1.6. Testing & debugging
5. Deployment & Maintenance	1.5. Deployment 1.6. Documentation 1.7. Evaluation & Maintenance

5.3. Task Definition

Detailed descriptions for each task are outlined in the table below:

Task	Definition
Project Initiation	• Define objectives, assign team roles, and get stakeholder approvals.
Requirements Gathering	• Collect business and technical requirements from stakeholders.
Data Collection	• Source SBA loan dataset from data warehouse for model training
Data Preprocessing	• Clean dataset by handling missing values, feature engineering, normalization, or encoding where appropriate
Exploratory Data Analysis (EDA)	• Analyze patterns, visualize correlations, and detect anomalies in the dataset.
Model Selection	• Select appropriate ML models (e.g., Random Forest, Logistic Regression, XGBoost) according to EDA and literature review results.
Model Training & Tuning	• Train models to do loan approval prediction using evaluation metrics like accuracy, AUC, and F1-score. • Fine-tune models using methods like k-fold cross validation.
Model Evaluation	• Test models on unseen data and compare results. • Identify the best-performing model to be used.
UI/UX Design	• Create front-end mockups for user input. • Get user feedback and finalize designs. • Develop system frontend using UI design.
Backend Integration	• Connect model with UI using Flask/Django APIs to create end-to-end system.
Testing & Debugging	• Perform unit tests. • Perform integration tests. • Perform user acceptance testing.
Deployment	• Deploy system to production environment.
Documentation	• Prepare final system documentation.

	- Prepare user guide.
Evaluation & Maintenance	- Use real-time monitoring systems to track system performance. - Regularly update dataset and retrain model to maintain or improve predictive accuracy. - Collect user feedback and perform continuous enhancements for functionality and performance improvements.

5.4. Task Dependencies

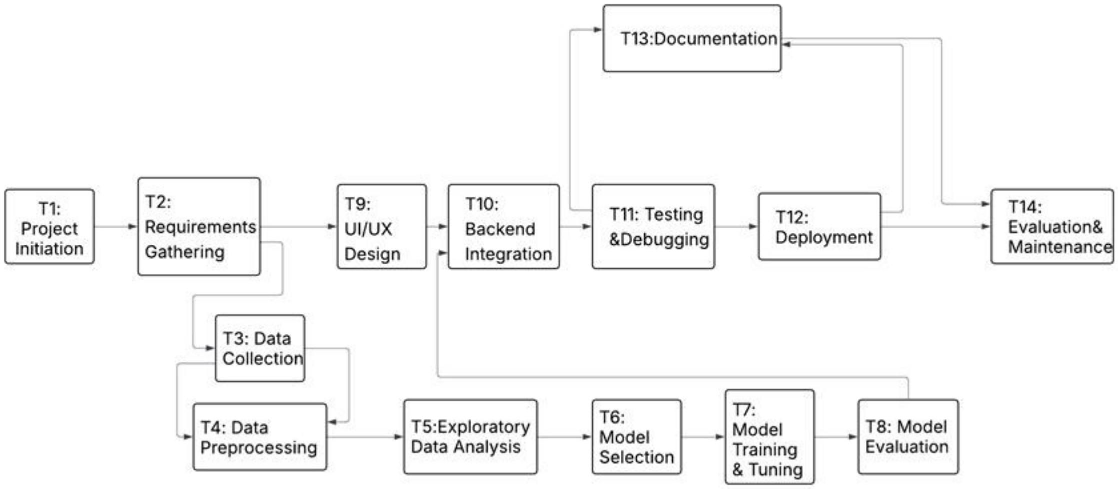


Figure 5.1: PDM Diagram

Figure 5.1 shows the dependencies among tasks.

5.5. Roles & Responsibilities

The table below describes the roles required and their responsibilities for each task:

Task	Roles	Responsibilities
Project Initiation	Project Manager	Define objectives, assign team roles, and get stakeholder approvals.
Requirements Gathering	Project Manager	Aggregate and document functional and non-functional requirements for the project.
	Business Analysts	Identify project requirements through business and market research, and inform the project manager.

	Technical Lead	Identify technical requirements based on the project requirements and inform the project manager
Data Collection	Data Engineer	Assist data scientists by preparing data for collection.
	Data Scientist	Work together with Data Engineers to source loan dataset for model training.
Data Preprocessing	Data Scientist	Clean dataset by handling missing values, feature engineering, normalization, or encoding where appropriate
Exploratory Data Analysis (EDA)	Data Scientist	Analyze patterns, visualize correlations, and detect anomalies in the dataset.
Model Selection	Data Scientist	Inform ML Engineer of EDA results.
	ML Engineer	Select appropriate ML models (e.g., Random Forest, Logistic Regression, XGBoost) according to EDA and literature review results.
Model Training & Tuning	ML Engineer	Train and fine-tune various ML models for predicting loan approval.
Model Evaluation	ML Engineer	Test models on unseen data and identify the best-performing model to be used.
UI/UX Design	UI/UX Designer	Design system UI.
	Frontend Engineer	Create system frontend based on UI design.
Backend Integration	Backend Engineer	Integrate model and UI into the backend, forming an end-to-end system.
	ML Engineer	Support Backend Engineer in integrating the model into the system.
	Frontend Engineer	Support Backend Engineer in integrating the system frontend.
Testing & Debugging	QA Engineer	Create and execute unit test, integration test, and user acceptance test.
	DevOps Engineer	Provide testing environment.
	Backend Engineer	Support QA engineer in testing the system.
Deployment	Devops Engineer	Deploy system into production environment.
Documentation	Backend Engineer	Provide documentation on the backend system APIs.

	Frontend Engineer	Provide documentation on
	ML Engineer	Provide documentation on the model training process, model evaluation, and final model used.
Evaluation & Maintenance	Backend Engineer	Monitor overall system health and performance.
	ML Engineer	Monitor model performance, and regularly update dataset to retrain model.
	Project Manager	Collect user feedback and formulate improvements.

5.6. WBS Dictionary

WBS Dictionary

Task ID	Task Name	Assigned	Dependencies
T1	Project Initiation	Project Manager	None
T2	Requirements Gathering	Project Manager, Business Analyst, Technical Lead	Start after T1 End;
T3	Data Collection	Data Engineer, Data Scientist	Start after T2 End:

T4	Data Preprocessing	Data Scientist	Start after T3 Start;
			End after T3 End;

T5	Exploratory Data Analysis	Data Scientist	Start after T4 End;
T6	Model Selection	Data Scientist, ML Engineer	Start after T5 End;
T7	Model Training & Tuning	ML Engineer	Start after T6 End;
T8	Model Evaluation	ML Engineer	Start after T7 End;
T9	UI/UX Design	UI/UX Designer, Frontend Engineer	Start after T2 End;
T10	Backend Integration	Backend Engineer, Frontend Engineer, ML Engineer	Start after T8&T9 End;
T11	Testing & Debugging	QA Engineer, Devops Engineer, Backend Engineer	Start after T10 End;

T12	Deployment	Devops Engineer, Backend Engineer, Frontend Engineer, ML Engineer	Start after T11 End;

T13	Documentation	Backend Engineer, Frontend Engineer, ML Engineer	Start after T11 Start;
			End after T12 End;

T14	Evaluation & Maintenance	Backend Engineer, ML Engineer, Project Manager	Start after T12&T13 End;

The WBS Dictionary table shows the entire task list their dependency relationships.

5.7. Review & Approval

Member	Approval
Project Manager	Approved
Technical Lead	Approved
Project Sponsor	Approved

6. Project Plan & Time Estimation

6.1. Section Overview

In this section, we are going to provide time estimation of each task, important milestones, and also allocate resources to these tasks.

6.2. Time Estimation

Based on the task list in the previous chapter, we are going to estimate the time required by each of them based on the common practices.

- **T1. Project Initiation (1 week)**
  The project initiation phase defines objectives, assigns team roles, and facilitates stakeholder communication. This week is crucial for holding kickoff meetings, developing a plan, and distributing responsibilities. All subsequent tasks depend on this step; starting too early may result in unclear direction or misallocated resources.
- **T2. Requirements Gathering (1 week)**
  This phase follows project initiation and involves thorough communication with all stakeholders to confirm business and technical requirements. One week is sufficient to complete meetings, research, and prepare initial documentation. It lays the foundation for data collection and modeling.
- **T3. Data Collection (1.5 weeks)**
  Data collection begins once requirements are clear. It includes identifying sources, obtaining permissions, downloading data, and initial cleaning. Given the time needed for data quality checks and possible delays, 1.5 weeks is a reasonable estimate.
- **T4. Data Preprocessing (1.5 weeks)**
  Data preprocessing may overlap with data collection but must wait for all data to be available before completion. This task includes handling missing values, feature engineering, normalization, and encoding. The 1.5-week duration ensures comprehensive and accurate processing.
- **T5. Exploratory Data Analysis (1 week)**
  EDA should start after preprocessing and uses visualizations and statistical analysis to understand data structure, variable relationships, and anomalies. One week is sufficient to conduct core analysis and develop an initial modeling strategy.

- **T6. Model Selection (1 week)**
  Model selection depends on EDA outcomes and aims to identify suitable algorithms (e.g., RF, LR, XGBoost). One week allows time to test different models and compare their performance—this is both reasonable and essential for informed decision-making.
- **T7. Model Training & Tuning (2 weeks)**
  This is the core stage of model development, involving multiple rounds of training, hyperparameter tuning, and validation. Considering data volume and model complexity, two weeks is an appropriate time frame to ensure practical, high-quality results.
- **T8. Model Evaluation (1 week)**
  Evaluation tests the final model on unseen data and helps select the best-performing solution. Metrics like confusion matrix, AUC, and F1-score are analyzed. One week is sufficient to complete evaluations and summarize the results.
- **T9. UI/UX Design (1.5 weeks)**
  UI/UX design can begin once requirements are finalized. It includes prototyping, flowchart development, and user experience validation. The 1.5-week period ensures clear interaction logic and visually appealing design, providing a solid foundation for further development.
- **T10. Backend Integration (1.5 weeks)**
  Backend integration begins after both the model and UI are ready. It involves connecting the components through APIs, deploying services, and conducting basic tests. 1.5 weeks is adequate to complete development and ensure system cohesion.
- **T11. Testing & Debugging (1 week)**
  This phase includes unit testing, integration testing, and user acceptance testing to ensure stable operation across scenarios. One week covers major test cases and bug fixing and is crucial for ensuring delivery quality.
- **T12. Deployment (1 week)**
  Deployment involves releasing the system to the cloud platform, configuring environments, and setting up security. One week ensures a stable production release with time to address unexpected issues during the go-live process.
- **T13. Documentation (2 weeks)**
  Documentation covers system architecture, user guides, and maintenance instructions. Since it spans from testing to deployment, allocating two weeks in parallel with deployment allows for well-structured and complete documentation.

- **T14. Evaluation & Maintenance (1 week)**
  The final phase includes preparing the project report, conducting evaluations, and planning future maintenance. One week post-delivery ensures proper review, presentation preparation, and feedback collection, helping the project transition into operation smoothly.

The table below lists out the estimated time required by each task.

Task ID	**Task Name**	**Duration (weeks)**
T1	Project Initiation	1
T2	Requirements Gathering	1
T3	Data Collection	1.5
T4	Data Preprocessing	1.5
T5	Exploratory Data Analysis	1
T6	Model Selection	1
T7	Model Training & Tuning	2
T8	Model Evaluation	1
T9	UI/UX Design	1.5
T10	Backend Integration	1.5
T11	Testing & Debugging	1
T12	Deployment	1
T13	Documentation	2
T14	Evaluation & Maintenance	1

6.3. Key milestones

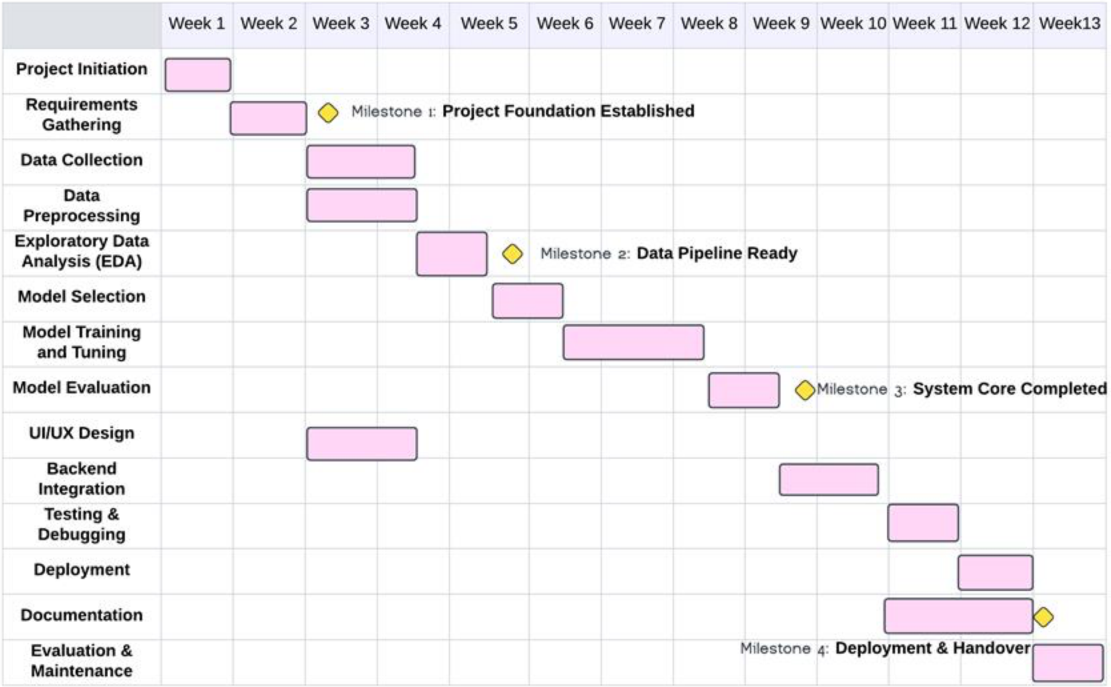
Milestone	Timeline	Task Involved
Milestone 1: Project Foundation Established	End of Week 2	T1. Project Initiation T2. Requirements Gathering
Milestone 2: Data Pipeline Ready	Mid-Week 5	T3. Data Collection T4. Data Preprocessing T5. Exploratory Data Analysis
Milestone 3: System Core Completed	End of Week 9	T6. Model Selection

		T7. Model Training & Tuning T8. Model Evaluation T9. UI/UX Design
Milestone 4: Deployment & Handover	End of Week 12	T10. Backend Integration T11. Testing & Debugging T12. Deployment T13. Documentation T14. Evaluation & Maintenance

The table above introduces 4 critical milestones in our project.

- **Milestone 1: Project Foundation Established**
  Completing project initiation and requirement gathering means the team has aligned goals, clearly defined roles, and documented technical and business needs. This milestone ensures the project is on a solid foundation before data work and development begin.
- **Milestone 2: Data Pipeline Ready**
  By this point, all necessary data has been collected, cleaned, and analyzed. The data pipeline is now prepared for model development. This milestone marks the readiness of the dataset and initial insights that guide modeling decisions.
- **Milestone 3: System Core Completed**
  The core components—machine learning models and user interface design—are completed. With models trained and evaluated and UI/UX in place, the system's technical foundation is ready for integration and testing.
- **Milestone 4: Deployment & Handover**
  The system is fully deployed, tested, and documented. Final evaluation and presentation preparations are underway. This milestone represents the formal transition from development to delivery and project closure.

The dependency relationship among these tasks can be checked on chat *5.1PDM* diagram.



6.1 Gantt Chart

Combining timeline and dependency relationship, we get *6.1 Gantt Chart.*

6.4. Resource Allocation

- T1 & T2: Project Initiation & Requirements Gathering
  - **People:** Project Manager, Business Analyst, Product Owner, Technical Lead
  - **Tools & Software:** Jira (backlog & approvals), Confluence (charters & specs), M iro/Whiteboard (process mapping)
  - **Infrastructure:** Meeting rooms or video-conferencing (MS Teams/Zoom), stakeholder interview booths
  - **Other:** Project-charter & requirements-template documents, stakeholder contact list

- T3–T5: Data Collection, Preprocessing & EDA
  - **People:** Data Engineer, Data Scientist
  - **Tools & Software:** Python (pandas, NumPy, SQLAlchemy), ETL framework (e.g. Airflow), Jupyter Notebooks，Visualization (Matplotlib/Seaborn, Tableau or Power BI)

    - **Infrastructure:** Cloud storage (AWS S3 or Azure Blob), relational database (RDS/Postgres)，Compute instances (EC2/GCP Compute with CPU/GPU as needed)
    - **Data Resources:** SBA loan dataset (or equivalent), schema documentation, data-access credentials

- T6–T8: Model Selection, Training, Tuning & Evaluation
    - **People:** ML Engineer, Data Scientist
    - **Tools & Software:** ML libraries (scikit-learn, XGBoost, TensorFlow/PyTorch), hyperparameter tuning (Optuna/Hyperopt), MLflow (experiment tracking)
    - **Infrastructure:** GPU-enabled cloud instances (AWS SageMaker/GCP AI Platform), Docker containers for reproducibility
    - **Data Resources:** Versioned training/validation/test splits, feature-store or secured data lake

- T9: UI/UX Design
    - **People:** UX/UI Designer, Product Owner (for feedback)
    - **Tools & Software:** Figma or Adobe XD, Zeplin (handoff), user-testing platform (Hotjar, UserTesting.com)
    - **Infrastructure:** Design workstations, screen-recording/test devices (mobile & desktop)

- T10–T11: Backend Integration & Testing
    - **People:** Backend Engineer, DevOps/MLOps Engineer, QA Engineer
    - **Tools & Software:** Framework (Flask or Django REST), Docker, Postman/Insomnia, pytest, Selenium/WebDriver
    - **Infrastructure:** Dev & QA environments (cloud VPC), CI/CD pipeline (GitHub Actions/Jenkins), container registry
    - **Other:** API documentation (OpenAPI/Swagger), test data & mocking services

- T12: Deployment
    - **People:** DevOps Engineer, Security Lead (for review)
    - **Tools & Software:** Terraform/CloudFormation, Kubernetes (EKS/GKE/AKS) or Docker Swarm, Helm charts
    - **Infrastructure:** Production cloud accounts (AWS/Azure/GCP), load balancer, managed database (RDS/Cloud SQL), secrets manager

- T13: Documentation
    - **People:** Technical Writer, Project Manager, Data Scientist (for model write-up)
    - **Tools & Software:** Confluence or ReadTheDocs, Markdown, GitHub repo for docs

  - **Other:** Style guide, API reference generator (Swagger), version control

- T14: Presentation & Final Review
  - **People:** Project Manager, Business Analyst, Data Scientist (for deep dive)
  - **Tools & Software:** PowerPoint or Google Slides, Zoom/MS Teams (if remote)
  - **Infrastructure:** Presentation room with projector, recording equipment (optional)
  - **Other:** Executive summary handouts, slide templates

6.5. Risk Management Plan: Identify risks and mitigation strategies

- **Data Bias in Machine Learning Model**
  - There is a risk that the machine learning model may produce biased results due to imbalanced or non-representative training data.
  - Mitigation: Use diverse and representative datasets during model training and perform regular fairness audits to identify and correct biases.
- **Cybersecurity Breaches**
  - Sensitive user data could be exposed to threats such as hacking or data leaks.
  - Mitigation: Implement strong encryption, secure authentication (e.g., OAuth2), and conduct regular security assessments and penetration testing.
- **Regulatory Non-compliance**
  - The project may face legal consequences if it fails to comply with financial regulations such as those of ASIC and the NCCP Act.
  - Mitigation: Engage with compliance and legal experts early in the development process to ensure all requirements are addressed.
- **Development Delays**
  - Technical challenges or unclear requirements could cause delays in the project timeline.
  - Mitigation: Use Agile methodology with defined sprints, regular progress reviews, and clear communication among team members.
- **External API Integration Failures**
  - Issues with third-party APIs (e.g., credit check or fraud detection) could disrupt core system functionality.
  - Mitigation: Begin integration testing early, use sandbox environments, and prepare fallback mechanisms for key services.

- **User Resistance to Automation**
  - End-users, especially internal staff, may be hesitant to trust or adopt the new automated system.
  - Mitigation: Design intuitive user interfaces and provide training, documentation, and support during rollout to encourage adoption.

6.6. Approval & Signatures: Capture necessary approvals

- **Identify Approval Gates**
  - Requirements Sign-Off: Business Lead / Product Owner
  - Design/Architecture Sign-Off: Architecture Review Board
  - Development Completion: Dev Lead / QA Lead
  - User Acceptance Testing (UAT) Sign-Off: End-User Representative
  - Production Go-Live: Operations / DevOps Lead
- **Standardized Sign-Off Form**
  - Create a simple sign-off template (electronic or paper) for each gate, capturing:
  - Deliverable name & version
  - Gate name (e.g. "Architecture Review")
  - Approver name, role, organization
  - Approval decision (Approved / Approved with Comments / Rejected)
  - Date and signature (or e-signature)
  - Comments or conditions
- **Digital Approval Workflow**
  - Use an e-signature tool (e.g. DocuSign, Adobe Sign) or built-in approval in your PM tool (Jira Cloud has a "approval" workflow step).
  - Route the form automatically to the right approver(s).
  - Require disposition (approve/reject) before allowing the project to move to the next state.

7. Project Cost Estimation

7.1. Cost Breakdown: List all cost components

The cost breakdown provides a comprehensive view of the estimated financial resources required for the successful delivery of the Automated Loan Approval System project. This breakdown helps in resource planning, financial oversight, and accountability. It includes:

- **Labor Costs:** The largest component, covering the wages of core team members such as the project manager, developers, data scientists, and UI/UX designers. These roles are crucial for system design, data processing, model training, and user experience.
- **Material & Software Costs:** Encompasses necessary equipment (e.g., hardware upgrades) and essential software tools for model development, data visualization, and API usage. These resources enable efficient development and deployment of the solution.
- **Overhead Costs:** Covers indirect costs such as electricity, internet, workspace usage, and administrative fees. These ensure smooth project operations and contribute to logistical support.
- **Contingency Fund:** Reserved to handle unforeseen expenses, such as extra cloud costs, emergency tool upgrades, or scope creep. A 10% allocation is considered standard in IT project management to manage risks.

7.2. Labor Costs: Estimate wages for team members

The table below shows estimated monthly wage for team members participating in the project:

Roles	Wage (AUD, Monthly)
Project Manager	$11,500
Business Analyst	$10,500
Technical Lead	$14,000
Data Scientist	$10,000
UI/UX Designer	$7,500
Backend Engineer	$11,500
Frontend Engineer	$8,500
ML Engineer	$10,000
Devops Engineer	$10,500

QA Engineer	$8,500

7.3. Material Costs: Identify needed equipment and software

The table below outlines estimated direct costs associated with tools and technology required for the project:

Item	Cost (AUD)	Description
Software		
Cloud storage	$600	Cloud data storage for storing model training data
CPU cloud compute instance	$3000	Cloud compute platform for running model
GPU-enabled cloud compute instance	$7000	Cloud compute platform for training model
Postman	$0	API testing platform (Free tiering)
Zeplin	$0	Design delivery platform (Free tiering)
Hardware		
Laptop	$9000	Work laptops for engineers to work on the project
Peripherals	$500	Mouse and headphones

7.4. Overhead Costs: Include indirect expenses

The table below outlines estimated ongoing operational costs that supports the project indirectly:

Item	Cost (Monthly)	Description
Collaboration tools (Slack, Jira, Confluence, Github, Zoom)	A$700	Monthly subscription for collaboration tools, estimated for 10 users
Office space	A$500	Working space provided by the bank
Office utilities	A$500	Electricity, internet, and cleaning fees

7.5. Contingency Fund: Allocate resources for unexpected costs

The contingency fund serves as a financial buffer to address unexpected challenges that may arise during the development and deployment of the Automated Loan Approval System project. Despite thorough planning, IT projects often encounter unforeseen issues such as:

- Sudden increases in cloud service consumption

- Emergency acquisition of additional software tools or licenses
- Scope changes requiring extra development efforts
- System integration challenges leading to extended testing periods

To mitigate these risks, a contingency allocation of approximately 10% of the total project budget is proposed. This reserve will be strictly managed, requiring justification and approval before use, ensuring financial discipline while maintaining flexibility to adapt to changing project needs.

Effective contingency planning enhances the project's resilience, supports smoother delivery, and helps maintain stakeholder confidence in both the budgeting process and overall project governance.

7.6. Budget Approval: Define budget approval process

1. **Budget Proposal Preparation:** Departments submit initial budget requests, including detailed cost assumptions, justifications, and supporting data
2. **Preliminary Review:** Finance conducts an initial check for completeness, compliance with guidelines, and high-level feasibility
3. **Detailed Analysis & Revision:** Requests undergo line-by-line scrutiny, scenario modeling, and comparisons to historical performance; stakeholders iterate until figures are realistic and achievable
4. **Final Approval:** The consolidated budget is routed to executive sponsors or board for sign-off; once approved, the budget is "frozen," and changes require formal change-control requests
5. **Post-Approval Monitoring:** After go-live, actuals are tracked against the approved budget, and variances trigger exception reports and potential budget amendments.

7.7. Cost Baseline

The cost baseline is a time-phased budget that serves as a benchmark for monitoring and controlling project expenditures. It consolidates all estimated costs related to labor, software, hardware, overheads, and contingency into a formal budget plan. This baseline allows project stakeholders to compare actual performance against planned expenditures and initiate corrective actions if variances arise.

The baseline was developed based on historical data, vendor quotations, and current market pricing trends. It will be used to measure financial performance throughout the project lifecycle using metrics such as **Cost Performance Index (CPI)** and **Cost Variance (CV)** under the **Earned Value Management (EVM)** framework. A control threshold of **±10%** has been established, beyond which variance analysis and formal budget revision will be triggered. Monthly financial reports will be prepared to monitor progress, ensuring transparency, accountability, and alignment with approved funding.

8. Project Schedule Development

In this section, we are going to provide an overview of the project schedule as well as resources allocations

8.1. Task Description, Timeline and Dependencies

Task List			
Task ID	**Task Name**	**Duration (weeks)**	**Dependencies**
T1	Project Initiation	1	None
T2	Requirements Gathering	1(milestone 1)	Start after T1 End;
T3	Data Collection	1.5	Start after T2 End:

T4	Data Preprocessing	1.5	Start after T3 Start;
			End after T3 End;

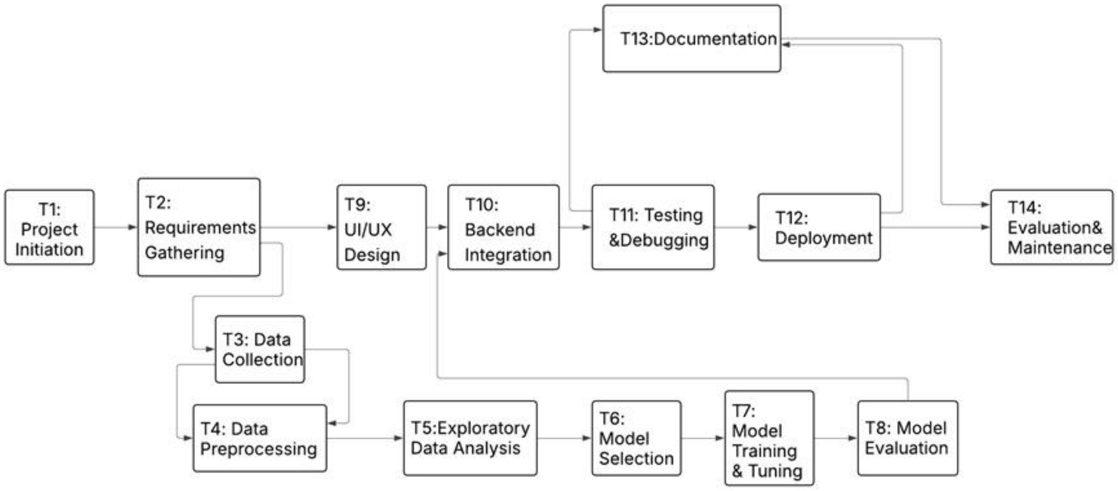
T5	Exploratory Data Analysis	1 (milestone 2)	Start after T4 End;
T6	Model Selection	1	Start after T5 End;
T7	Model Training & Tuning	2	Start after T6 End;
T8	Model Evaluation	1	Start after T7 End;
T9	UI/UX Design	1.5 (milestone 3)	Start after T2 End;
T10	Backend Integration	1.5	Start after T8&T9 End;
T11	Testing & Debugging	1	Start after T10 End;
T12	Deployment	1	Start after T11 End;

T13	Documentation	2	Start after T11 Start;
			End after T12 End;

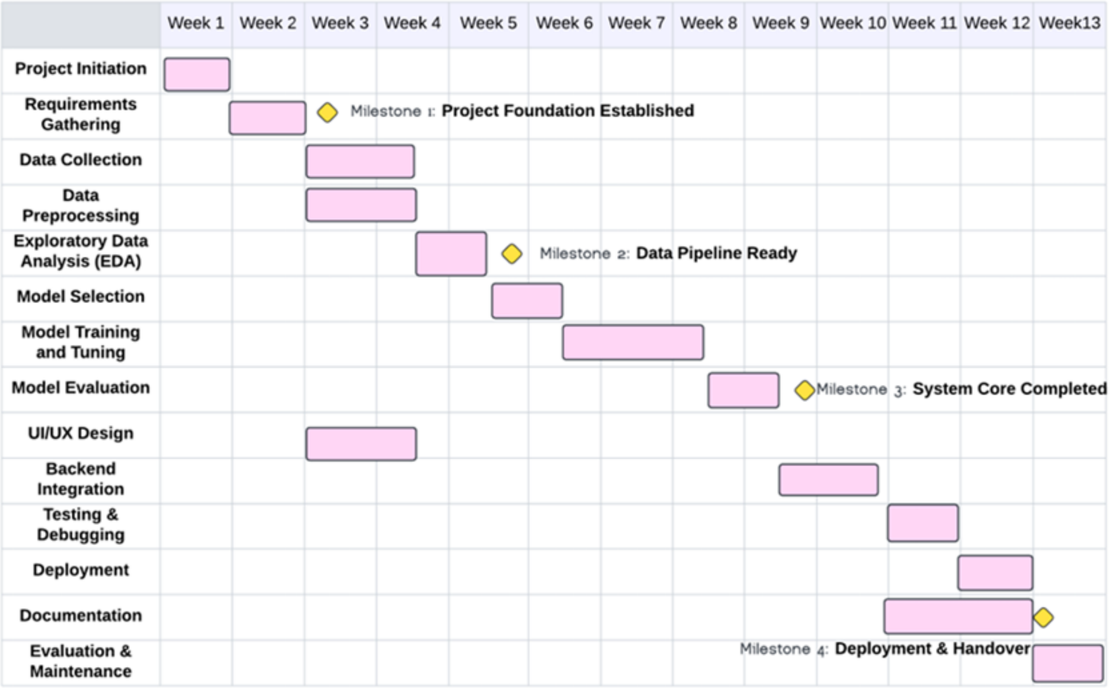
T14	Evaluation & Maintenance	1 (milestone 4)	Start after T12&T13 End;

Task table

Project could be broken down into 14 tasks, as *Task table* showed.



5.1 PDM Diagram



6.1 Gantt Chart

Gantt chart and *PDM chart* visualize the timeline and dependency relationships among these tasks.

8.2. Resources allocation

The table below provides a comprehensive view of resource allocation:

Tasks	People	Tools&Software	Infrastructure	Data
T1 T2	Project Manager, Business Analyst, Product Owner, Technical Lead	Jira, Confluence, Miro/Whiteboard	Meeting rooms or video-conferencing stakeholder interview booths	Project-charter & requirements-template documents stakeholder contact list
T3 T4 T5	Data Engineer Data Scientist	Python ETL framework Jupyter Notebooks Visualization	Cloud storage relational database Compute instances	SBA loan dataset schema documentation data-access credentials
T6 T7 T8	ML Engineer Data Scientist	ML libraries hyperparameter tuning MLflow	GPU-enabled cloud instances Docker containers for reproducibility	Versioned training/validation/test splits feature-store or secured data lake
T9	UX/UI Designer Product Owner	Figma or Adobe XD Zeplin user-testing platform	Design workstations screen-recording/test devices	
T10 T11	Backend Engineer DevOps/MLOps Engineer QA Engineer	Framework Docker Postman/Insomnia Pytest Selenium/WebDriver	Dev & QA environments CI/CD pipeline container registry	API documentation test data & mocking services
T12	DevOps Engineer Security Lead	Terraform/CloudFormation Kubernetes (EKS/GKE/AKS) or Docker Swarm Helm charts	Production cloud accounts load balancer managed database secrets manager	
T13	Technical Writer Project Manager Data Scientist	Confluence or ReadTheDocs Markdown GitHub repo for docs	Style guide API reference generator version control	
T14	Project Manager Business Analyst Data Scientist	PowerPoint or Google Slides Zoom/MS Teams	Presentation room with projector recording equipment	Executive summary handouts slide templates

9. Time Control Processes

9.1. Goals

The targets of our time control processes include following:

1. Automatically track the start and completion of each task
2. Clearly define task dependencies and automatically identify blockages and delays
3. Automatically notify task owners and project managers if a task falls behind
4. Support dynamic adjustment of schedules and timeline.

9.2. Tools

The tool set we are going to use is Jira + Jira Automation + Gantt Plugin (BigPicture):

- **Jira**: Jira is Atlassian's agile project management and issue-tracking platform that helps teams collaborate efficiently through boards, sprints, and customizable workflows. It provides built-in dashboards and real-time reports to automatically monitor project progress and visualizes schedule changes whenever task statuses update, enabling rapid timeline adjustments.

- **Jira Automation**: Jira Automation is the zero-code rules engine embedded in Jira that automates repetitive tasks via "trigger–condition–action" chains. It continuously watches for key events (e.g., overdue tasks or priority changes) and, when conditions are met, dynamically updates schedules or sends adjustment notifications to streamline calendar management.

- BigPicture is a deeply integrated Gantt-chart and project-portfolio management plugin for Jira, offering drag-and-drop scheduling, dependency mapping, and resource-load visualization.

9.3. Functions and Workflow

The tool set we introduced in the last section provides powerful functionalities including:

1. **Automation Rules Engine**: Monitors actual progress vs. plan and issues alerts for lagging tasks.

2. **Dependency Detection Module**: Detects when downstream tasks are blocked due to delays in predecessor tasks.

3. **Daily/Weekly Report Generator**: Automatically produces task status reports and delay summary tables.

4. **Adjustment Recommendation Module**: When deviations become unmanageable, proposes task compression, resource acceleration, or sequence reordering.

And these functions enable the following automatic progress tracking rules listed in *Automatic Rule Table*.

Rule ID	Trigger	Condition	Action
AR1	Planned start date passed	If the task has not been started yet	Notify assignee and project manager; label as 'Late Start'; add comment
AR2	Planned end date passed	If the task is still incomplete	Mark as 'Delayed'; notify assignee; create subtask to explain delay
AR3	A predecessor task is marked as 'Delayed'	If the task depends on another task that must finish first	Notify dependent task owner; add 'Blocked' tag; shift start date if auto-scheduling is enabled
AR4	Every day at 10 PM	No condition is required	Generate progress summary; update dashboard; email/Slack notification
AR5	Task marked as 'Delayed' for over 2 days	If the task remains idle or incomplete for more than 2 days	Notify PM; suggest adding collaborators or rescheduling

AR6	3 days before task due date	If the task is still incomplete	Notify assignee with pre-deadline alert; check if dependencies are complete
AR7	Predecessor task is completed early	If the task is linked to a predecessor that allows early start	Notify dependent task owner; suggest early start; auto-pull forward if allowed
AR8	Task is marked 'Done'	If no time has been logged for the completed task	Add 'Missing Time Log' tag; comment to prompt timesheet update
AR9	Task status changed from 'Done' to 'In Progress'	Always applies when the status change occurs	Log change reason; notify PM; update projected end date
AR10	Due date passed	If the task is overdue and not high priority	Set priority to High; tag as 'overdue'; assign to project lead

Automatic Rule Table

9.4. Reschedule rule

We set rules for adjusting the schedule based on the dependency relationships among tasks.

Light Delay (≤ 1 day)	System sends a reminder and prompts for a progress update	A variancedency adjustments required
Moderate Delay (1–3 days)	Automatically launches a “Progress Update Request” form	if the task lies on the critical path, it also triggers a Gantt-chart compression recommendation
Severe Delay (> 3 days)	Alerts the supervisor to approve additional resources or to split the task	automatically generates an “Impact Chain” report and recommends mitigation strategies such as expediting predecessor tasks or running subsequent tasks in parallel.

Reschedule Rule Table

As *Reschedule Rule Table* lists, delay triggers different actions based on its severity. Long the delay is, high priority should be set.

10. Budgeting & Cost Baseline

10.1. Cost Baseline

The table below outlines the overall cost baseline of the project:

	Months			Total
Items	1	2	3	
Labor costs				
Project Manager	$11,500	$11,500	$11,500	$34,500
Business Analyst	$10,500	$10,500	$10,500	$31,500
Technical Lead	$14,000	$14,000	$14,000	$42,000
Data Scientist	$10,000	$10,000	$10,000	$30,000
UI/UX Designer	$7,500	$7,500	$7,500	$22,500
Backend Engineer	$11,500	$11,500	$11,500	$34,500
Frontend Engineer	$8,500	$8,500	$8,500	$25,500
ML Engineer	$10,000	$10,000	$10,000	$30,000
Devops Engineer	$10,500	$10,500	$10,500	$31,500
QA Engineer	$8,500	$8,500	$8,500	$25,500
Software Costs				
Cloud storage	$200	$200	$200	$600
CPU cloud compute instance	$1000	$1000	$1000	$3000
GPU-enabled cloud compute instance	-	$3500	$3500	$7000
Hardware Costs				
Laptop	$9000	-	-	$9000
Peripherals	$500	-	-	$500
Contingency	$10000	$10000	$10000	$30000
Totals	$123200	$117200	$117200	$357,600

10.2. Tracking Methods

Throughout the project, Earned Value Management will be used to assess the project's performance with regards to the provided budget. Actual costs will be tracked periodically using a spreadsheet, and compared in a monthly basis against the baseline using metrics like Cost Performance Index (CPI) and Cost Variance (CV). We also define an overall control threshold of 10%, which will trigger if CPI goes below 0.9: exceeding this control threshold will trigger variance analysis and require formal change control approval to revise the budget. This structured approach ensures that financial resources are aligned with the project scope and schedule.

10.3. Variance Analysis

Variance analysis will be done by comparing the actual cost with the cost baseline monthly. This is done for every item to make it easy to identify the source of any cost variances. An example is given in the table below:

Item	Actual Cost	Baseline Cost	Variance	Notes
Cloud Storage	$350	$200	$150	Training data size larger than expected
GPU-enabled cloud compute instance	$3000	$3500	-$500	Model training finished earlier, instance shutdown to reduce costs

10.4. Financial Reports

Financial reports will be generated monthly for the board and executive sponsors to review. These reports will contain variance analysis for all items in the cost baseline, as well as the total CPI and CV of the project so far. This ensures transparency on budget use between the project team and stakeholders, in addition to promoting accountability. An example report is given below:

2nd Month Financial Report				
Item	Cumulative Actual Cost	Cumulative Baseline cost	Variance	Notes
Project Manager	$23,000	$23,000	$0	
Team Members	$167,000	$168,000	$1000	Engineers overtime
Cloud storage	$300	$400	-$100	Training data size smaller than expected

CPU cloud compute instance	$2,000	$2,200	$200	Requires instance with more vCPU
GPU-enabled cloud compute instance	$3,500	$3,500	$0	
Hardware	$9,500	$9,400	-$100	Less peripherals needed
Total	$205,300	$206,500	$1,000	
Cost Performance Index (CPI)			0.994	Slightly overbudget

11. Conclusion

This report has documented a complete and disciplined approach to planning the design and implementation of an **AI-driven automated loan approval system** within a structured project management framework. The initiative addresses a pressing need within financial institutions to modernize legacy processes, improve the customer experience, and reduce operational inefficiencies — all while maintaining compliance with financial regulations and ethical standards.

Over the course of this project, we applied a combination of traditional and agile project management practices to define project scope, schedule, budget, quality objectives, and risk tolerance levels. The project plan includes a detailed work breakdown structure (WBS), milestone tracking, stakeholder engagement strategies, resource allocations, and cost estimation through bottom-up and parametric methods. Crucially, we established baselines for scope, schedule, and cost, allowing for performance tracking using EVM indicators such as Cost Performance Index (CPI) and Schedule Performance Index (SPI).

The proposed loan approval system is not just technically feasible but also strategically aligned with long-term organizational goals. It is expected to reduce loan processing time by up to 60%, lower administrative overhead, and provide fairer, data-backed decision-making that is transparent to both customers and regulators. By incorporating explainable AI methods, the system promotes trust and accountability, addressing common concerns around black-box machine learning models in financial decision-making.

Furthermore, ethical and regulatory considerations have been thoroughly assessed, ensuring that the system remains compliant with frameworks such as the Australian Privacy Principles (APPs), APRA guidelines, and AI governance recommendations. We have also included a contingency reserve and change control strategy to manage uncertainty and evolving project demands.

In conclusion, this project exemplifies how AI and disciplined project management can converge to deliver meaningful digital transformation in banking. The implementation of the proposed system stands to benefit both internal stakeholders (through increased efficiency and accuracy) and external customers (through faster, fairer loan decisions). Looking ahead, the focus will shift toward model refinement, continuous monitoring, and user training to ensure sustainable adoption and long-term value realization.

References

Calomiris, W. C., Longhofer, S. D., & Jaffe, D. M. (2017). Credit Rationing. *The New Palgrave Dictionary of Economics*, 1-10.

Daltrics. (2025). *AI Credit Scoring: The Future of Credit Risk Assessment*. Retrieved from Daltrics: https://www.datrics.ai/articles/the-essentials-of-ai-based-credit-scoring

European Commission. (2021). *Proposal for a regulation laying down harmonized rules on artificial intelligence*. Retrieved from Artificial Intelligence Act: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

Haque, F. A., & Hassan, M. (2024). Bank loan prediction using machine learning techniques.

Kingsley, J. (2022, March 17). *Banking on a game-changer: AI in financial services*. Retrieved from https://impact.economist.com/perspectives/financial-services/banking-game-changer-ai-financial-services

Mazoué, R. (2025, January 23). *From credit scoring to GenAI: How modern credit decisioning has evolved—and what's next*. Retrieved from https://www.taktile.com/articles/from-credit-scoring-to-genai

Moro, S., Cortez, P., & Rita, P. (2015). Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation. *Expert Systems with Applications*, 1314-1324.

Muhammad, I., Dahlia, R., Muhammad, I. I., & Lisnawanty. (2024). erformance Analysis of Ensemble Learning and Feature Selection Methods in Loan Approval Prediction at Banks. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 557-564.

Nikolaienko, O. (2023, January 17). *Lending Re-Imagined with Alternative Credit Score Data and Automation*. Retrieved from https://www.infopulse.com/blog/digital-lending-alternative-credit-score

Nwoke, J. (2024). Digital Transformation in Financial Services and FinTech: Trends, Innovations and Emerging Technologies. *International Journal of Finance*, 1-24.

Office of the Australian Information Commissioner. (2025, May 16). *Australian Privacy Principles*. Retrieved from https://www.oaic.gov.au/privacy/australian-privacy-principles

Tala. (2025, May 16). *Data Ethics at Tala*. Retrieved from https://tala.co/data-ethics/

Acknowledgement:

We acknowledge the use of Generative AI(Claude AI, Copilot) to generate summary of discussed materials.

Prompt Used:

We entered the following prompt(s):

1) Summarize the below content

2) Create table for below content

Use: We modified the outputs, added and updated content