

1. Overall Design Document for Software Project Economic Analysis and Decision-Making Tool

1. Project Overview

1.1 Project Objectives

Develop an interactive software tool to assist software engineering teams in evaluating economic decisions throughout the Software Development Life Cycle (SDLC). The tool integrates cost estimation, budget management, risk management, and resource optimization to provide data-driven decision support for project stakeholders.

1.2 Core Functions

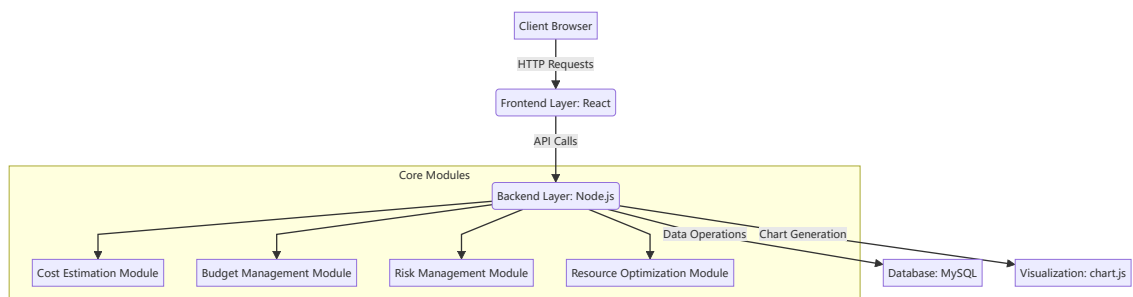
- **Cost Estimation Module:** Support multiple cost estimation methods including COCOMO, Function Point Analysis, Expert Judgment, etc.
- **Budget & Cost Management:** Calculate financial metrics (ROI, NPV, IRR, Payback Period), implement budget tracking and variance analysis.
- **Risk Management Module:** Provide sensitivity analysis, decision tree analysis, and Monte Carlo simulation functions.
- **Resource Allocation & Optimization:** Implement resource leveling and smoothing algorithms, support multi-scenario analysis.

1.3 Technology Stack

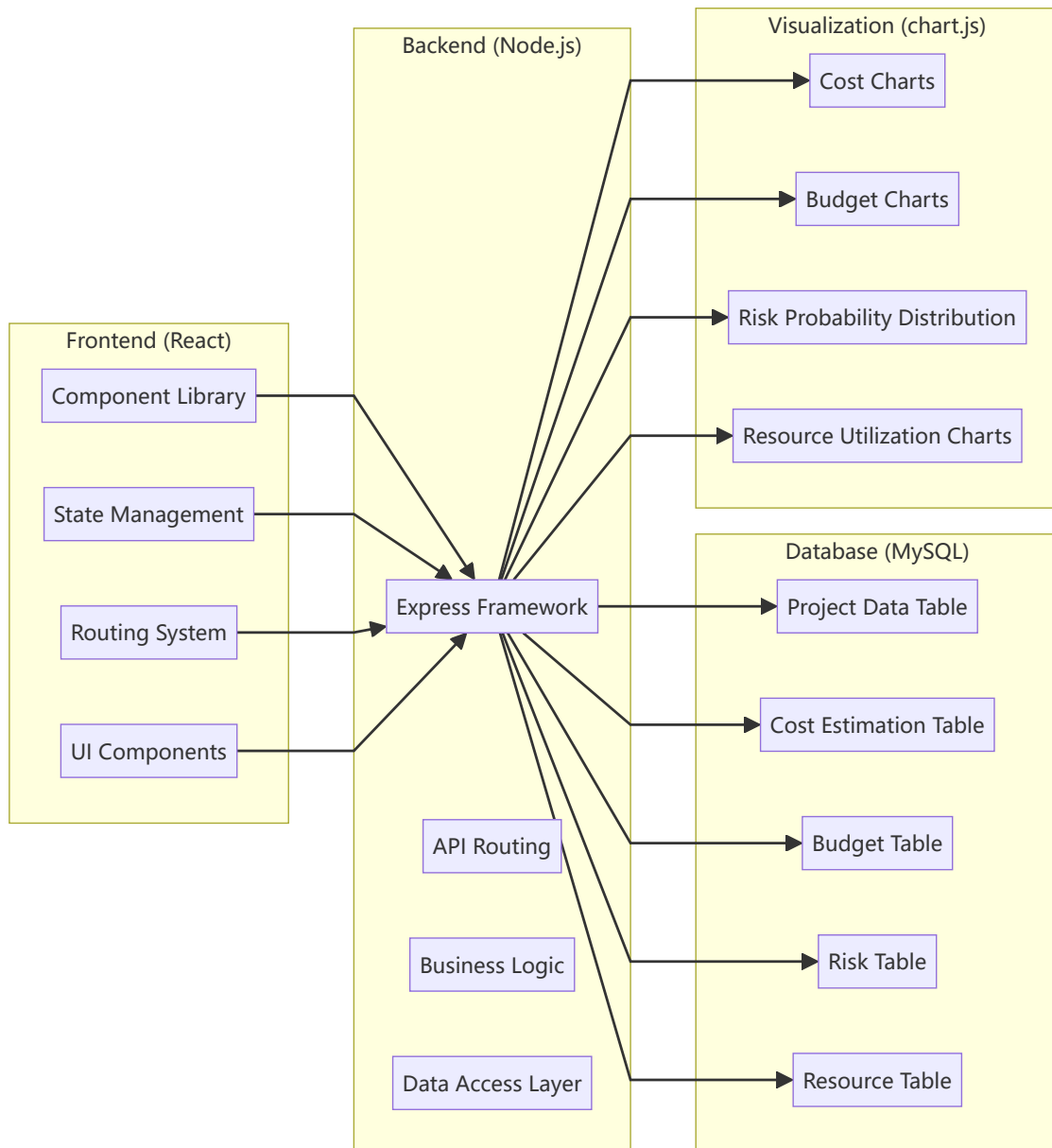
- **Frontend Framework:** React
- **Backend Framework:** Node.js
- **Visualization Technology:** chart.js
- **Database:** MySQL

2. System Architecture Design

2.1 Overall Architecture Diagram



2.2 Technical Component Architecture



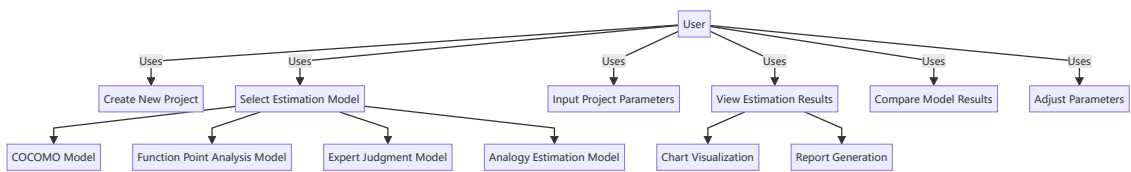
3. Module Design

3.1 Cost Estimation Module

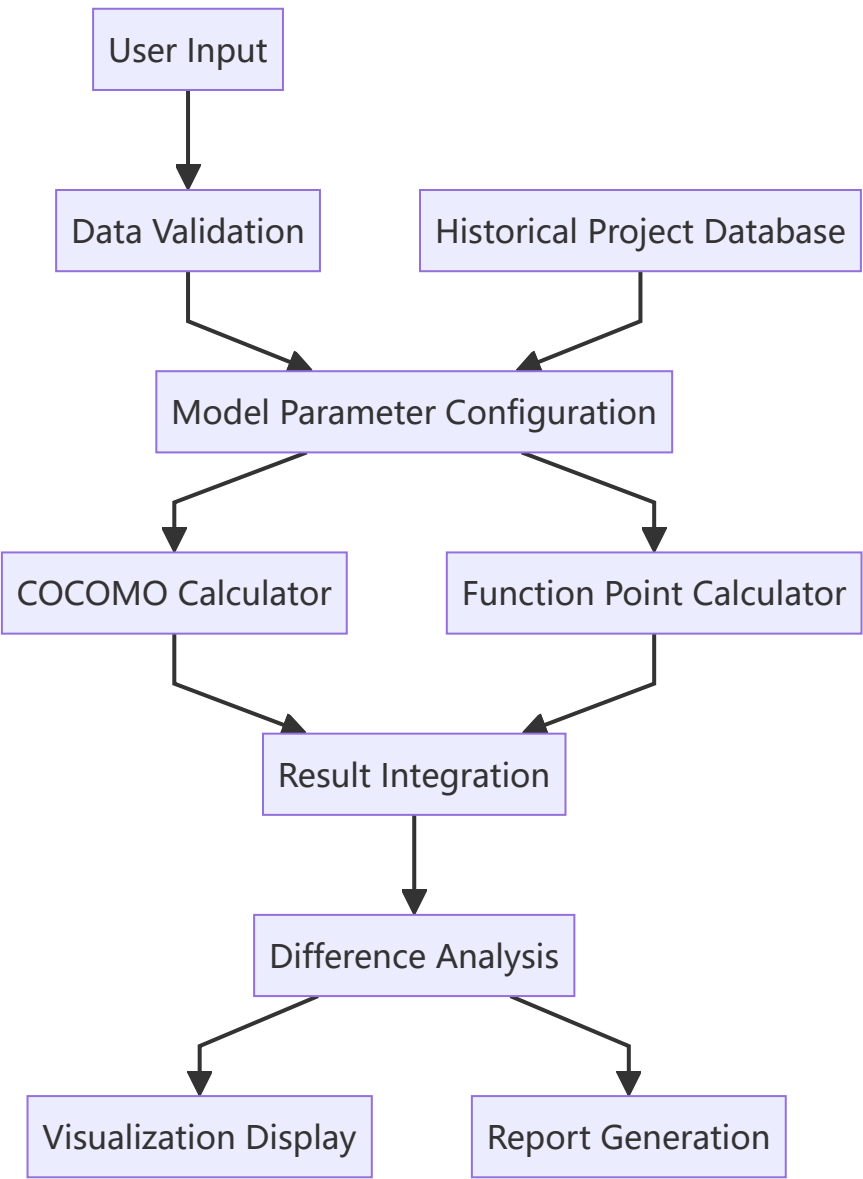
3.1.1 Functional Description

- Support multiple cost estimation methods: COCOMO model, Function Point Analysis, Expert Judgment, Delphi method
- Allow users to input project attributes (e.g., lines of code, team size)
- Provide comparative analysis of output results from different models
- Support historical data import and reuse

3.1.2 Use Case Diagram



3.1.3 Data Flow Diagram

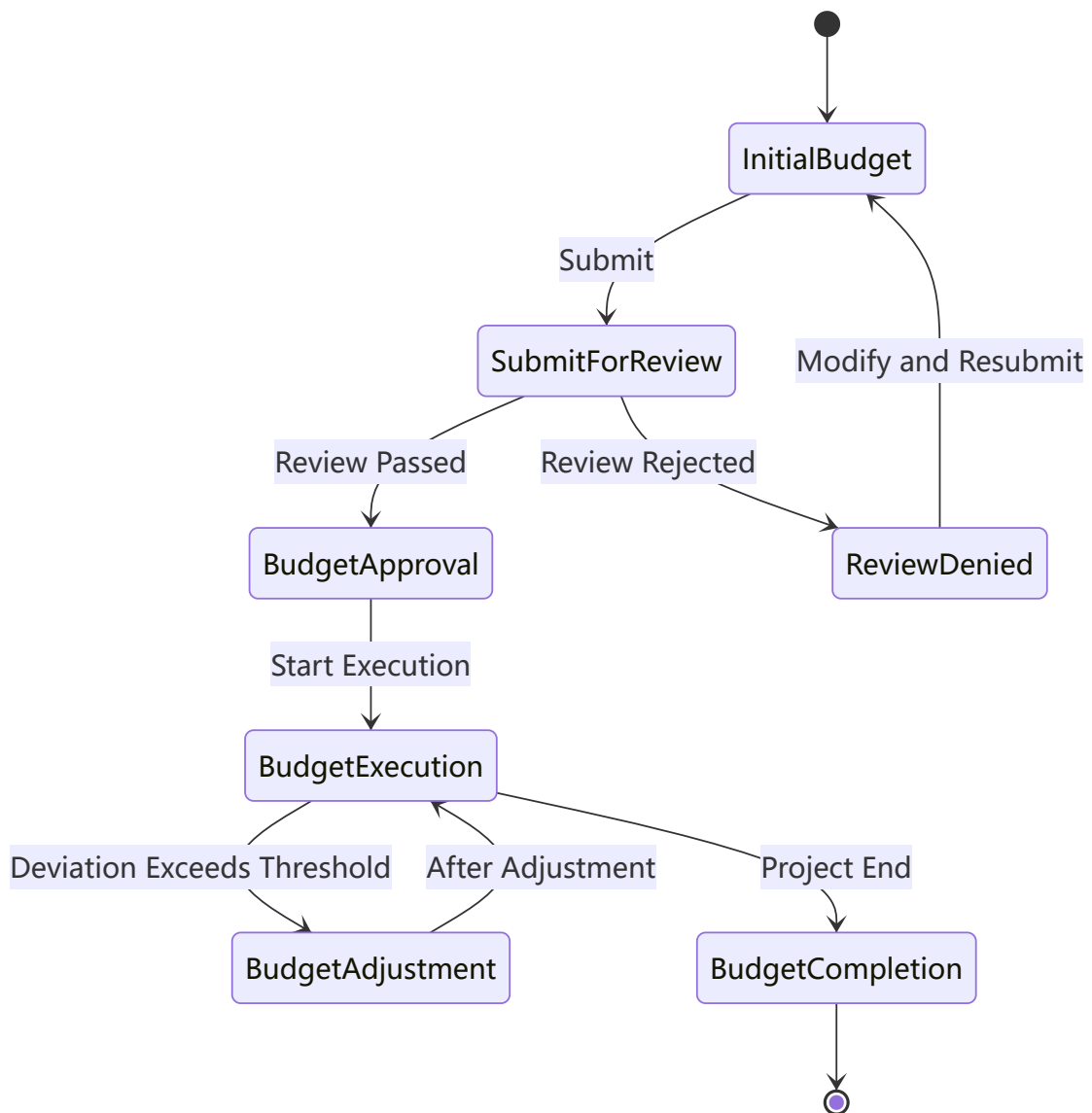


3.2 Budget Management Module

3.2.1 Functional Description

- Calculate key financial metrics: ROI, NPV, IRR, Payback Period
- Implement budget preparation, approval, execution, and adjustment processes
- Provide budget tracking and variance analysis functions
- Support budget forecasting and scenario simulation

3.2.2 State Diagram

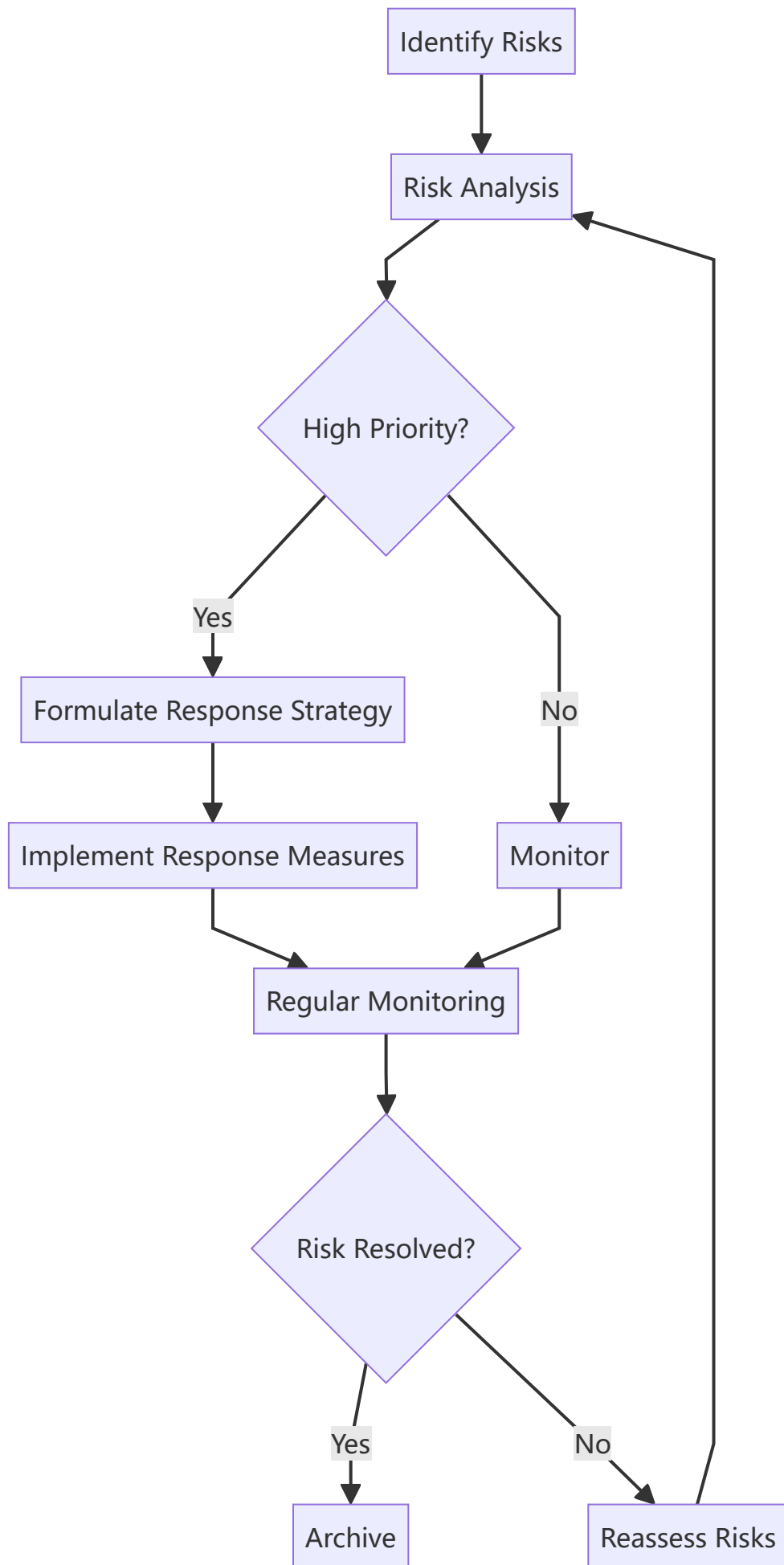


3.3 Risk Management Module

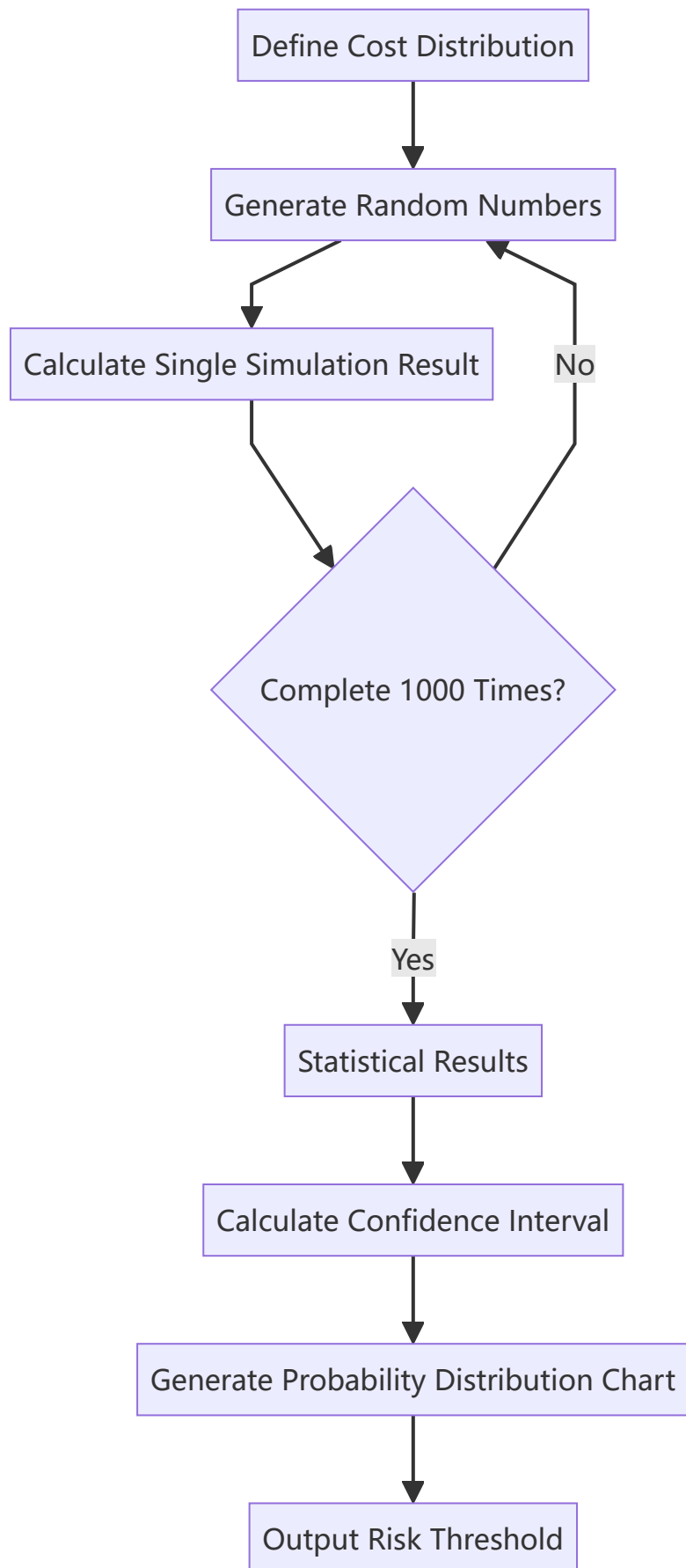
3.3.1 Functional Description

- Risk identification and priority assessment (P-I matrix)
- Sensitivity analysis: Identify key variables
- Decision tree analysis: Path-based EMV analysis
- Monte Carlo simulation: Probabilistic result range analysis
- Risk visualization and report generation

3.3.2 Activity Diagram



3.3.3 Monte Carlo Simulation Algorithm Flowchart

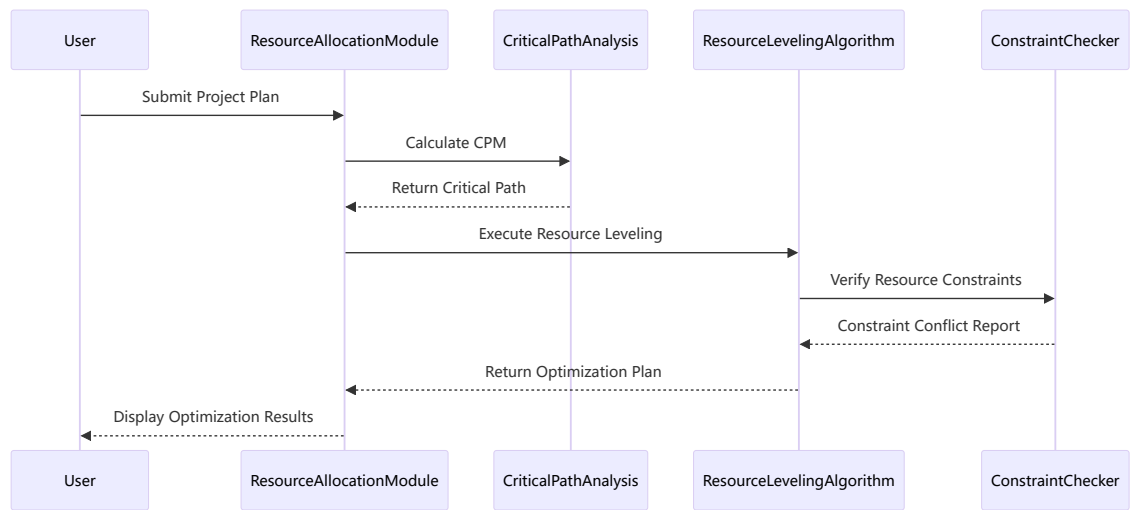


3.4 Resource Allocation & Optimization Module

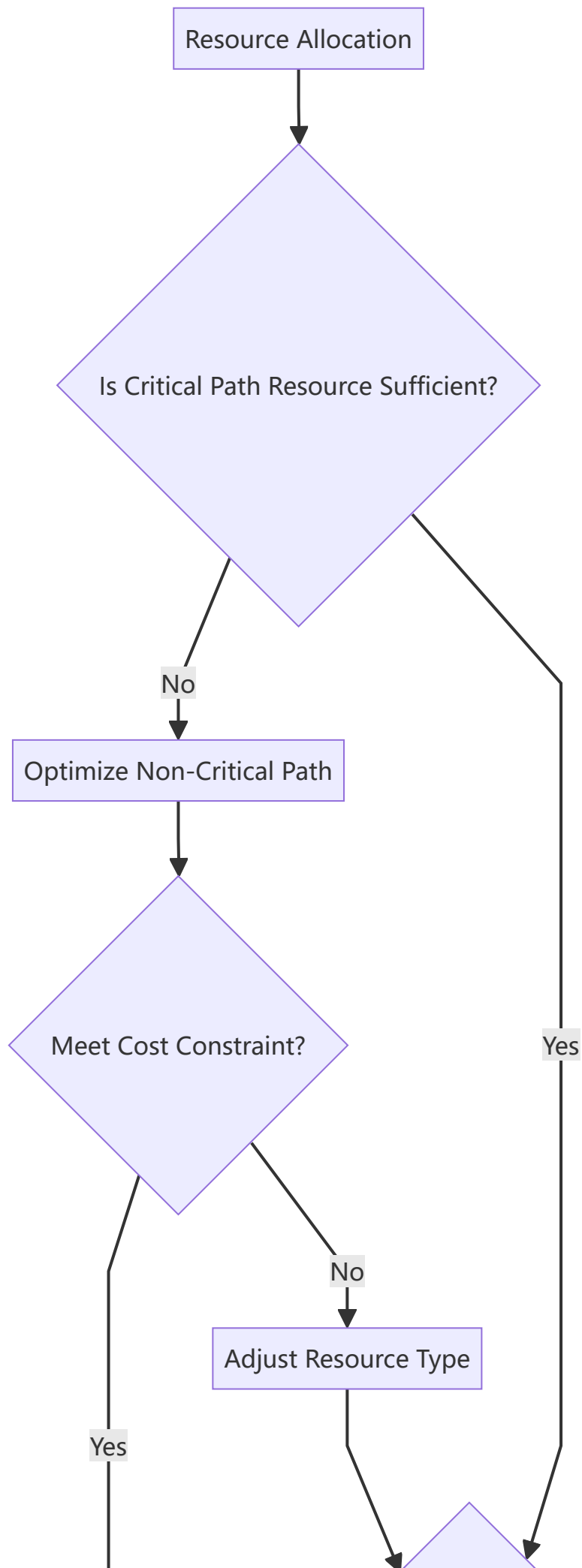
3.4.1 Functional Description

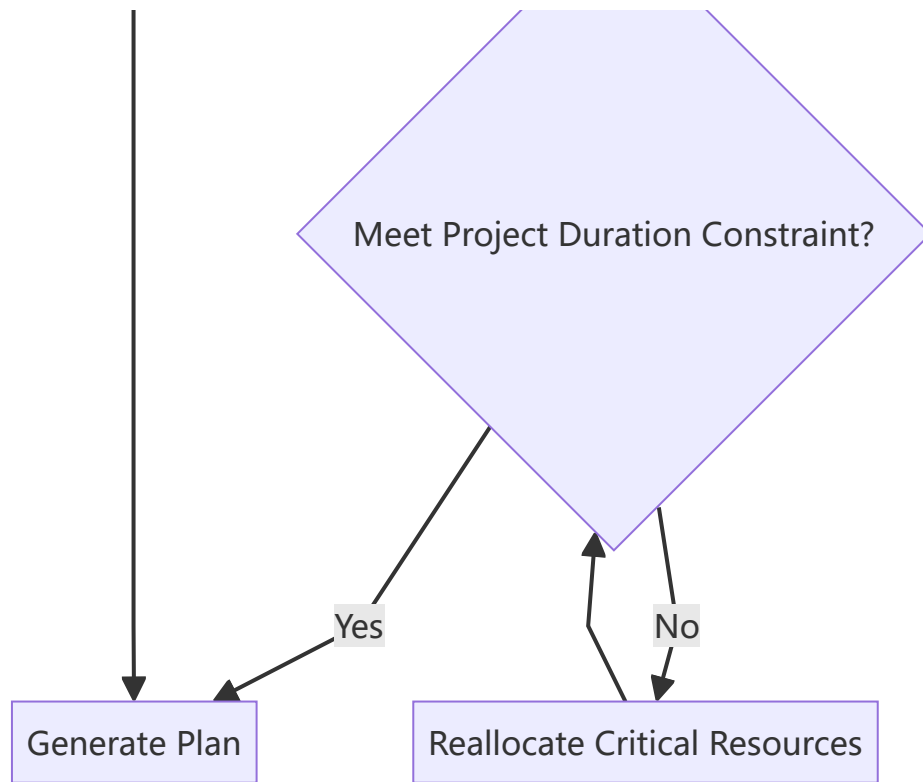
- Resource leveling algorithm: Solve over-allocation issues
- Resource smoothing algorithm: Optimize non-critical paths using machine learning
- Constraint programming: Solve multi-dimensional resource allocation problems
- Multi-scenario analysis: Trade-off analysis of cost/time/resource
- Generate Pareto-optimal solutions

3.4.2 Sequence Diagram



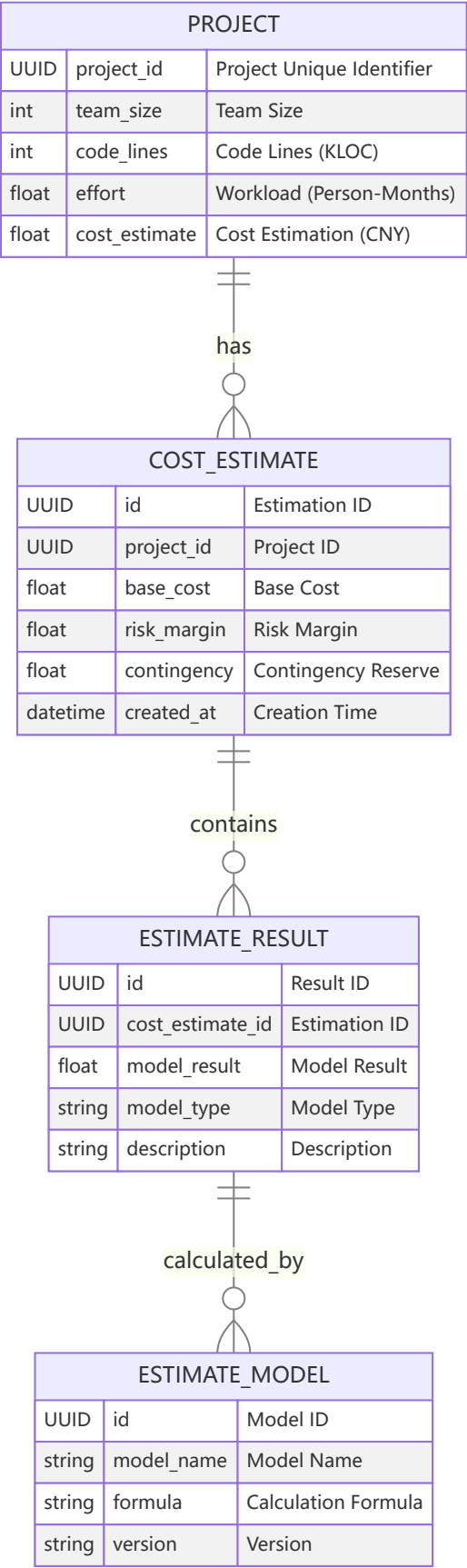
3.4.3 Decision Tree (Resource Allocation Strategy)





4. Data Model Design

4.1 Entity-Relationship Diagram (ER Diagram)



4.2 Data Dictionary (Partial)

Data Item	Type	Description	Constraints
project_id	UUID	Project unique identifier	Primary key, not null
team_size	int	Team size (number of people)	1-100
code_lines	int	Number of code lines (KLOC)	0-1000
effort	float	Workload (person-months)	>0
cost_estimate	float	Cost estimation (CNY)	>0
risk_prob	float	Risk probability	$0 \leq x \leq 1$

5. Interface Design

5.1 External Interfaces

5.1.1 Data Import Interface

- **Format:** CSV/Excel
- Functions:
 - Support historical project data import
 - Automatic or manual field mapping
 - Data integrity check and type validation

5.1.2 Report Export Interface

- **Format:** PDF/PPT/Excel
- Functions:
 - Generate cost analysis reports
 - Generate budget execution reports
 - Generate risk assessment reports
 - Support custom report templates
 - Export content permission control based on user roles

5.2 Internal Interfaces (Inter-module APIs)

Source Module	Target Module	API Path	Data Format
Cost Estimation	Budget Management	/api/v1/budget/estimate	JSON
Budget Management	Risk Management	/api/v1/risk/budget	JSON

Source Module	Target Module	API Path	Data Format
Risk Management	Resource Optimization	/api/v1/resource/risk-adjust	JSON
Resource Optimization	Budget Management	/api/v1/budget/resource-update	JSON

6. Non-functional Requirements

6.1 Performance Requirements

Indicator	Requirement	Testing Method
System Response Time	≤2s (normal operations)	Automated interface testing
Monte Carlo Simulation Time	≤15s	Performance testing tools
Concurrent Users	≥20 users	Load testing
Data Import Throughput	≥500 records/min	Batch data import testing

6.2 Usability Requirements

- **Interface Design Standards:** Comply with WCAG 2.1 AA accessibility standards
- **Operation Paths:** Key operation paths ≤3 steps
- **Responsive Design:** Support resolutions above 1024×768
- Help System:
 - Context-sensitive help embedded in each functional module
 - Complete user manual (PDF/online)
 - Tutorial videos for typical cases

6.3 Security Requirements

- Data Security:
 - Encrypted storage for sensitive data (e.g., user authentication information)
 - Data backup strategy: daily full backup + real-time incremental backup
 - Role-Based Access Control (RBAC)
- System Security:
 - Prevention of SQL injection and XSS attacks
 - Regular security vulnerability scans (at least monthly)
 - Abnormal login detection and alerting

7. Acceptance Criteria

7.1 Functional Acceptance

Function Point	Acceptance Criteria	Number of Test Cases
Multi-model cost estimation	Error rate $\leq 10\%$ (compared with standard cases)	15
Budget tracking	Deviation calculation accuracy 100%	10
Monte Carlo simulation	Generate probability distribution chart with 95% confidence interval	8
Resource optimization algorithm	Resource utilization improvement $\geq 15\%$	12

7.2 Non-functional Acceptance

Indicator	Acceptance Criteria	Testing Tools
Response time	$\leq 2s$ for 95% of requests	JMeter
Concurrent users	20 users without lag	LoadRunner
Data accuracy	Calculation error $\leq 0.1\%$	Automated test scripts
Interface usability	Average operation time $\leq 30s$	User experience testing