Software Requirements Specification (SRS) Document

Software Requirements Specification (SRS) Document

Project Title: Project Budget Analyzer

Team Members:

- Abhishek M
- Anuroop Gowda C
- Rohit Singh

Estimated Duration: 3 to 4 Weeks

---

1. Introduction

1.1 Purpose

Project Budget Analyzer aims to predict the total cost of a project by analyzing current progress, historical trends, and upcoming phases, enabling comparison with the allocated budget to ensure financial feasibility. By leveraging data-driven techniques such as machine learning models (e.g., Random Forest), it estimates future expenses and identifies potential budget overruns in specific tasks or phases. This process supports proactive decision-making, allowing project managers to adjust resources and prioritize tasks effectively. Additionally, this system provides real-time insights, helping teams adapt to changes dynamically and avoid unnecessary expenses through timely interventions.

1.2 Intended Audience

- Internal Use Only
- Primary User: Project Manager

1.3 Scope

The system will allow users to analyze a single project at a time. It will provide real-time cost predictions and alerts through a desktop application. The tool is intended to run locally on Windows machines without internet dependency. It will also include login authentication and basic data encryption for security. The system supports both budget and resource forecasting and provides visual analytics to simplify decision-making.

1.4 Definitions and Acronyms

- **ML**: Machine Learning
- **GUI**: Graphical User Interface
- **API**: Application Programming Interface
- **SRS**: Software Requirements Specification

---

2. Problem Statement of the Project

The primary challenge in modern project management is staying within budget and meeting deadlines while utilizing resources efficiently. Cost overruns, inefficient resource allocation, and poor forecasting often result in project delays and financial losses. The "Project Budget Analyzer" aims to solve this by providing an intelligent system that predicts total project costs and identifies resource gaps early. Using machine learning models and real-time data, the system assists managers in making data-driven decisions to reduce waste, redistribute workloads, and take preventive action before issues escalate.

Software Requirements Specification (SRS) Document

3. Overall Description

3.1 Product Perspective

This is a standalone software system that integrates GUI, backend processing, and ML prediction models. It is designed for internal project management teams to operate entirely offline.

3.2 Product Functions

The product includes modules for data input, real-time analysis, forecasting, visualization, and reporting. It aims to bridge the gap between raw data and actionable insights.

3.3 User Characteristics

- Basic familiarity with desktop applications
- Understanding of project management concepts

3.4 Constraints

- Windows OS
- Local execution (no cloud dependency)
- MongoDB for data storage
- Should provide fast performance for predictions

3.5 Assumptions and Dependencies

- Input project data will be available in structured format (CSV/JSON)
- ML models will be pre-trained and updated periodically via scheduled jobs

---

4. Specific Requirements

4.1 Functional Requirements

- FR1: User login system with encrypted credentials
- FR2: Upload project progress and cost data (CSV/JSON)
- FR3: Real-time prediction of project budget status
- FR4: Alert generation for possible overruns
- FR5: Dashboard with charts (cost trends, resource utilization)
- FR6: Export results and reports
- FR7: Visual indication of resource bottlenecks
- FR8: Task and phase-wise prediction of cost overrun risks

4.2 Non-Functional Requirements

- NFR1: System should run on Windows OS
- NFR2: Prediction results should appear in < 2 seconds
- NFR3: All stored data must be encrypted
- NFR4: Application should be installable as a standalone executable
- NFR5: Should not require internet access

---

5. Methodology & System Design

The system architecture is modular and built for local deployment. It consists of:

- **Frontend (GUI)**: Built using PySide/HTML,CSS to allow user-friendly interaction.
- **Backend (API Layer)**: Developed using FastAPI to handle requests between the GUI and ML models.
- **Machine Learning Models**: Trained on historical project data to predict cost overruns and resource demands.
- **Database Layer**: MongoDB is used to store user inputs, prediction results, and logs.

Workflow:

1. **Data Input**: The user uploads project progress and resource data via GUI.
2. **Processing Layer**: Data is validated and forwarded to ML models through API endpoints.
3. **Prediction Engine**: Forecasts total budget and flags resource bottlenecks.
4. **Visualization Layer**: Results are shown in dashboards using interactive charts and graphs.

This pipeline ensures real-time responsiveness, high modularity, and easy scalability in the future. The choice of FastAPI enhances performance, while the GUI simplifies user interaction, especially for non-technical managers.

---

6. System Architecture

6.1 Technology Stack

- **Programming Language:** Python
- **Frameworks:** FastAPI (backend API), PySide (GUI)
- **ML Libraries:** scikit-learn, XGBoost, TensorFlow, PyTorch
- **Database:** MongoDB
- **Visualization:** matplotlib, seaborn, plotly
- **Automation:** Bash Scripting

6.2 Tools and Utilities

- GitHub for version control
- Postman for API testing
- PyInstaller for executable packaging

---

7. Use Cases

Use Case 1: Login

- Actor: Project Manager
- Input: Username and password
- Output: Access granted/denied

Use Case 2: Upload Project Data

- Actor: Project Manager
- Input: Project file (CSV/JSON)
- Output: Data parsed and stored in DB

Use Case 3: Predict Budget Feasibility

- Actor: Project Manager
- Input: Project progress data
- Output: Predicted total cost, overrun warning

Use Case 4: Analyze Resource Feasibility

- Actor: Project Manager
- Input: Human/resource allocation data
- Output: Resource usage graphs, bottleneck alerts

---

8. Conclusion

The "Project Budget Analyzer" provides an efficient, real-time solution to predict project costs and evaluate resource feasibility. By integrating machine learning, FastAPI, and PySide GUI, it empowers project managers with actionable insights that reduce the risk of budget overruns and ensure optimal resource usage. With local deployment and strong security features like encryption and authentication, it serves as a robust internal tool. In the future, the system can be expanded with cloud deployment, multi-project support, and more advanced predictive models. Its modular design makes it easy to maintain and adapt as business needs evolve.

---

9. Appendix

9.1 References

- scikit-learn user guide
- MongoDB official docs

-