AI-Powered Automated SRS Generator

Project Proposal

Project Code

Project Team

Muhammad Talal Ahmed	*BSCS51F22S074*	*Team Lead*
Muhammad Tayyab	*BSCS51F22S070*	*Team Member*
Muhammad Ibrahim	*BSCS51F22S078*	*Team Member*

Project Supervisor

Dr. Fahad Maqbool

Project Coordinator

Dr. Saad Razzaq

Submission Date

7 October 2025

Project Supervisor Signature

_____

Table of Contents

1. Abstract

Many software projects fail due to vague or an incomplete requirement given by non-technical clients. Writing a Software Requirement Specification (SRS) requires expertise, which many clients and small teams lack.

Our proposed solution is an **AI-powered Automated SRS Generator**. Through a chat interface, clients can describe their project idea in plain English. The system uses **NLP and AI** to:

- Ask clarifying questions,
- Suggest missing requirements,
- Generate a complete **IEEE 830-compliant SRS**,
- Produce **UML diagrams** (use-case, class, sequence), and
- Provide **basic wireframes** of the system.

This will save time, reduce miscommunication, and make requirement engineering accessible to non-technical stakeholders.

2. Justification

Traditionally, **System Analysts or Requirements Engineer** gather client requirements and write the SRS with supporting diagrams. However, this is time-consuming, costly, and prone to human error.

Recent tools like Almware, Reqi's "Rex," and Blueprint.AI show early attempts to automate requirement documentation, but they lack **interactive clarification**, **missing requirement suggestions**, and **visual model generation**. Our project combines all these aspects into one assistant.

Justification: This project bridges the gap between clients and developers by acting like a **junior Requirements Engineer**. It adds academic value (NLP, AI, automation) and industrial value (time and cost savings in requirement engineering).

3. Objectives

- Develop a **chat-based interface** where clients describe requirements.
- Implement **AI-driven clarification** (chatbot asks follow-up questions).
- Add a **knowledge base/ML model** to suggest missing requirements.
- Generate **IEEE 830-compliant SRS** documents.
- Automatically create **UML diagrams** (use-case, class, sequence).
- Optionally generate **basic UI wireframes**.
- Ensure usability, scalability, and security of the platform.

Methodology / Steps:

1. Requirement analysis & architecture design.
2. Chatbot & NLP model integration.

3. SRS template design and document generation.
4. Diagram generation with PlantUML/Mermaid.
5. Frontend integration with backend APIs.
6. Testing, refinement, and evaluation.

4. Project Scope

In Scope:

- Interactive AI-based requirement gathering.
- Automated SRS generation (IEEE 830).
- UML diagrams (use-case, class, sequence).
- Wireframe prototypes.
- Exportable reports (PDF/DOCX).

Out of Scope:

- Full system implementation for client projects.
- Advanced UI/UX prototyping beyond basic wireframes.
- Domain-specific deep knowledge bases (initially general best practices only).

5. Tools and Technologies

- **Frontend:** React + Tailwind CSS (chat interface, diagram display).
- **Backend:** Python (Flask/FastAPI).
- **AI/NLP:** Hugging Face Transformers, spaCy, OpenAI API or some open-source model.
- **Database:** MongoDB Atlas.
- **Diagram Generation:** PlantUML / Mermaid.js.
- **Wireframes:** Mermaid wireframes.
- **Deployment:** Docker + Cloud (AWS/Heroku/Vercel).
- **Collaboration:** GitHub, CI/CD pipelines.

6. Potential Impact

- **Academic:** Demonstrates AI + NLP in requirements engineering, publishable research.
- **Industrial:** Saves time, reduces cost of hiring analysts, improves project success rate.
- **Social:** Makes software development accessible for non-technical founders/startups.

7. Project High Level Plan

Phase 1 – System Design & Setup

- Define project architecture, data models, and development environment.
- Design initial chat interface prototype.

Phase 2 – NLP & Chat Assistant Development

- Integrate NLP/AI model for requirement understanding.

- Implement clarification dialogue and missing requirement suggestion engine.

Phase 3 – Document & Diagram Generation

- Build SRS generation module (IEEE 830 format).
- Implement automated UML diagram generation (use case, class, sequence).

Phase 4 – Prototyping & Integration

- Add basic UI wireframe generation.
- Integrate all modules into a unified system.
- Ensure usability, security, and performance.

Phase 5 – Testing & Finalization

- Conduct user testing with sample requirements.
- Refine AI suggestions and outputs.
- Prepare final deliverables (demo, documentation, reports).

8. References

AltexSoft Editorial Team. (2023, November 30). *Functional and Nonfunctional Requirements: Specification and Types.* From AltexSoft: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/

Copilot4DevOps. (2025, March 18). *How AI is Transforming Requirements Gathering and Documentation.* From Copilot4DevOps: https://copilot4devops.com/ai-in-requirements-gathering-and-documentation

PlantUML. (n.d.). From PlantUML: https://plantuml.com/

Dr. Fahad Maqbool (fahad.maqbool@uos.edu.pk)
Mr. Farooq Javed (farooq.javed@uos.edu.pk)