SmartSDLC – AI-Enhanced Software Development Lifecycle

# 1. INTRODUCTION

## 1.1 Project Overview

SmartSDLC is a full-stack, AI-powered platform that redefines the traditional Software Development Lifecycle (SDLC) by automating key stages using advanced NLP and generative AI. It helps software teams convert unstructured requirements into structured artifacts such as code, test cases, and documentation.

## 1.2 Purpose

The purpose of SmartSDLC is to streamline and automate critical SDLC tasks, reducing manual effort, enhancing consistency, and accelerating project delivery timelines.

# 2. IDEATION PHASE

## 2.1 Problem Statement

Software teams face inefficiencies and delays due to the manual handling of requirements, code generation, testing, and documentation. SmartSDLC aims to automate and simplify these steps using AI.

## 2.2 Empathy Map Canvas

• HEAR: Complaints about manual work and delayed deliveries.  
• SEE: Fragmented tools and inconsistent outputs.  
• SAY: “We need a smarter way to handle repetitive SDLC tasks.”  
• DO: Manually write requirements, code, and test cases.  
• THINK & FEEL: Frustrated, overwhelmed, and unproductive.  
• PAINS: Lack of integration and automation.  
• GAINS: AI-assisted automation across SDLC.

## 2.3 Brainstorming

Our team explored automation opportunities across the SDLC phases: requirement gathering, code generation, testing, debugging, and documentation. We identified key pain points and envisioned AI modules to address each one.

# 3. REQUIREMENT ANALYSIS

## 3.1 Customer Journey Map

The customer uploads raw requirements or prompts to the platform. The system processes them and returns structured outputs such as user stories, code snippets, or test cases.

## 3.2 Solution Requirement

• Accept requirement files (e.g., PDFs)  
• NLP-based requirement classification  
• Prompt-based code generation  
• Test case generation  
• Bug fixing and code summarization  
• Integrated AI chatbot

## 3.3 Data Flow Diagram

User Input → Watsonx AI Model → NLP Processing → Output Generator → Frontend Interface

## 3.4 Technology Stack

• Frontend: Gradio / HTML, CSS  
• Backend: FastAPI  
• AI Models: IBM Watsonx Granite-20B  
• Tools: LangChain, PyMuPDF, Hugging Face Transformers

# 4. PROJECT DESIGN

## 4.1 Problem Solution Fit

AI can automate repetitive SDLC tasks, reducing the time and effort required by software teams while improving consistency.

## 4.2 Proposed Solution

SmartSDLC includes modules for requirement classification, AI code generation, bug fixing, test case creation, code summarization, and an AI chatbot assistant.

## 4.3 Solution Architecture

SmartSDLC architecture consists of:  
• User input interface  
• AI model service (Watsonx Granite)  
• Backend with FastAPI  
• Output renderer and export tools

# 5. PROJECT PLANNING & SCHEDULING

## 5.1 Project Planning

We divided the project into phases: ideation, design, model integration, UI development, and testing. Each module was built incrementally.

# 6. FUNCTIONAL AND PERFORMANCE TESTING

## 6.1 Performance Testing

Performance was measured in response time of AI modules, quality of generated code, and feedback from early users for usability.

# 7. RESULTS

## 7.1 Output Screenshots

Screenshots of generated code, test cases, chatbot responses, and structured requirement breakdown will be inserted here.

# 8. ADVANTAGES & DISADVANTAGES

Advantages:  
• Automation of SDLC tasks  
• Faster delivery  
• Consistent output  
  
Disadvantages:  
• Dependence on model quality  
• Requires initial setup and integration effort

# 9. CONCLUSION

SmartSDLC is a transformative tool that uses AI to modernize the software development lifecycle. It enhances productivity, reduces errors, and improves collaboration across teams.

# 10. FUTURE SCOPE

Future versions may include multilingual support, continuous learning from user feedback, advanced integrations with CI/CD tools, and support for more programming languages.

# 11. APPENDIX

• Source Code: https://github.com/lakshmi-narim/SmartSDLC-AI/blob/main/SmartSDLC-code  
• Project Demo:





