Data Flow Diagram (DFD):

DFD Level 0 – SmartSDLC Overview

End User (Client) → SmartSDLC Frontend

SmartSDLC Frontend → SmartSDLC API (FastAPI + LangChain)

SmartSDLC API (FastAPI + LangChain) →

- Requirement Classifier
- AI Code Generator
- Bug Fixer/ Debugger

AI Code Generator → Test Case and Code Summerizer

Test Case and Code Summerizer → Floating Chat Assistant
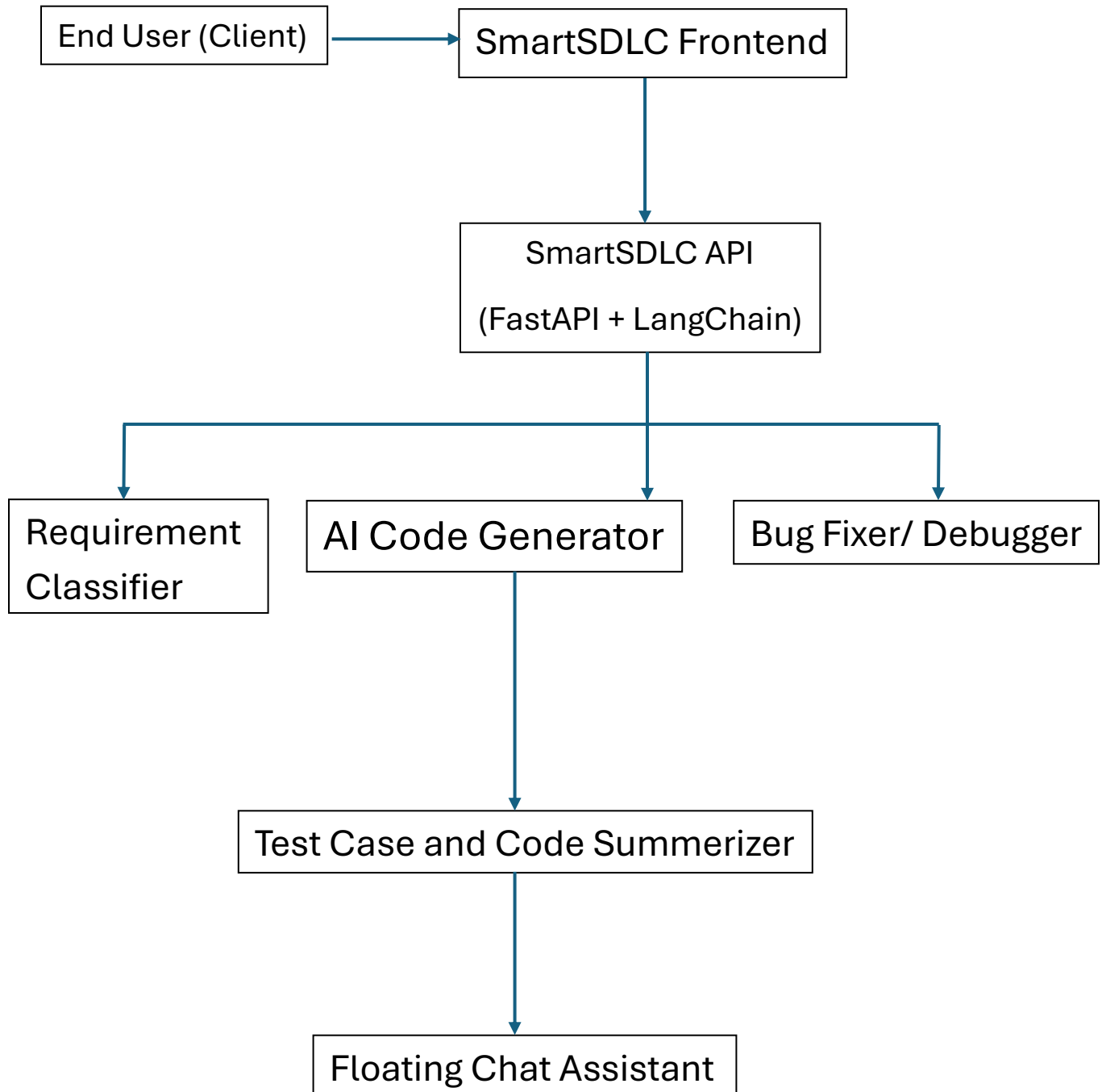
User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Developer	Requirement Classification	USN-1	As a developer, I can upload a PDF file of raw requirements and see them classified.	I can see clearly grouped requirements by SDLC phase.	High	Sprint-1
Developer	AI Code Generation	USN-2	As a developer, I can input a natural language prompt and receive production-ready code.	I receive working, syntax-highlighted code on the frontend.	High	Sprint-1
QA Engineer	Test Case Generation	USN-3	As a tester, I can input code and get auto-generated test cases.	I see test cases using unittest or pytest formats.	Medium	Sprint-2
Developer	Bug Fixer	USN-4	As a developer, I can submit buggy code and get the fixed version with explanation.	I get a corrected version of my code.	High	Sprint-2
Developer	Code Summarization	USN-5	As a developer, I can get human-readable summaries of any code snippet.	I receive a readable explanation of code functions.	Medium	Sprint-2
End User (Any Role)	Floating AI Chatbot Assistant	USN-6	As a user, I can ask SDLC-related queries to an AI chatbot.	I receive accurate, contextual SDLC answers in real-time.	High	Sprint-2