CA4425: Software Engineering Project

Name: Mohammed Rayaan Pasha

Register No.: 242BCA08

Subject: Software Engineering

Topic: Sitcom Simulation (SitcomGPT)

Abstract

Sitcom Simulation is a backend-based distributed system that uses AI-driven characters and message queues to generate continuous sitcom-style conversations inspired by popular television shows such as *Friends* and *How I Met Your Mother*. Each character operates as an independent microservice with its own personality logic and communicates asynchronously through RabbitMQ. Large Language Models are used to generate dialogue while preserving each character's unique tone and behavior. The project demonstrates how asynchronous message passing and microservice architecture can be combined to create a technically robust yet entertaining simulation environment.

Introduction

The Sitcom Simulation project explores modern distributed system concepts through a creative and experimental approach. In this system, each sitcom character functions as a standalone Express-based service with a predefined personality prompt. When a character produces dialogue, the message is published to a shared queue, triggering responses from other character services using AI models such as Google Gemini. Although the system presents itself as a humorous simulation, it highlights real-world software engineering principles including microservice isolation, asynchronous communication, scalability, and distributed load handling. The project serves both as a learning tool and as a demonstration of multi-agent coordination.

Module Details

1. Character Microservice Module

Each character runs as an independent microservice with its own server instance and personality definition. This modular design allows characters to be added, removed, or modified without impacting the rest of the system.

2. Message Queue Module

This module uses RabbitMQ to manage asynchronous communication between character services. Messages are published to and consumed from shared queues, ensuring loose coupling and reliable message delivery.

3. AI Dialogue Generation Module

This module integrates Large Language Models to generate character-specific dialogue. Prompts are tailored to maintain consistency in tone, humor, and behavioral traits for each character.

4. Conversation Flow Module

The conversation flow module manages turn-taking and response sequencing between characters. It ensures that interactions remain continuous and coherent while allowing spontaneous and dynamic exchanges.

5. Personality Logic Module

This module defines character-specific rules, memories, and behavioral constraints. It helps preserve individuality and prevents responses from becoming generic across different characters.

6. Asynchronous Processing Module

This module handles non-blocking message consumption and response generation. It improves system scalability and demonstrates real-world asynchronous processing patterns.

7. Future Enhancement Module

This module focuses on extensibility. Planned improvements include long-term memory for characters, visual dashboards for conversation monitoring, additional characters, and advanced queue management strategies.

Software Process Model Selected – Agile

The Agile software development model is best suited for the Sitcom Simulation project. The system is experimental in nature, with evolving requirements and unpredictable AI-driven behavior. Agile supports incremental development, rapid iteration, and continuous improvement, which align well with the project's goals and structure.

Reasons for Choosing Agile

- The project consists of many small, independent microservices that can be developed in parallel
- Character behavior and interaction patterns require frequent testing and adjustment
- New characters and features can be added without redesigning the entire system
- Failures in one service do not affect the complete environment
- Agile supports experimentation and creative freedom with minimal upfront constraints

---

Planned Agile Development Stages

1. Requirement Identification and Initial Design

Core system goals, message flow design, and basic character roles are identified.

2. Prototype Development

A small number of character services are implemented to validate message passing and AI response quality.

3. Incremental Character Expansion

Additional characters are added as separate services, each with customized personality prompts.

4. Testing and Observation

Conversation quality, queue behavior, and system stability are observed and refined through repeated testing.

5. Iterative Enhancement

Features such as memory handling, visualization tools, and improved AI prompting are introduced gradually.

Comparative Study of Software Process Models

Aspect	Waterfall Model	Agile Model (Chosen)	RAD Model	Prototyping Model	Incremental Model	Spiral Model
Development Style	Linear and sequential	Iterative and adaptive	Rapid UI-focused development	Prototype-based, exploratory	Feature-wise increments	Risk-driven cycles
Flexibility	Very Low	Very High	High	Very High	Medium	Medium
Requirement Changes	Difficult	Easy	Easy	Easy	Moderate	Moderate
Documentation	Heavy	Minimal	Low	Low	Medium	High
Risk Handling	Poor	Moderate	Low	Low	Medium	Excellent
Suitability for AI Behavior	Not suitable	Best suited	Partially suitable	Useful in early stages	Moderately suitable	Suitable but complex
Overall Suitability	Poor	Excellent	Good	Good for experimentation	Moderate	Moderate

---

Conclusion

Sitcom Simulation demonstrates how modern software engineering concepts can be explored through a creative and engaging project. By leveraging asynchronous communication, microservices, and AI-driven agents, the system highlights real-world distributed system behavior in an accessible way. Among various SDLC models, Agile proves to be the most appropriate due to its support for iterative growth, adaptability, and experimentation. The project effectively shows that even playful simulations can serve as strong technical demonstrations of scalable and flexible system design.