

Activity 1:

1. Ask Orientation Questions

- **Tasks:**
 1. Design chatbot interface for question input (2 days)
 2. Develop NLP model for recognizing orientation-related queries (3 days)
 3. Integrate with Student Affairs data for real-time answers (3 days)
 4. Implement predefined questions & quick replies (2 days)
 5. Test response accuracy and performance (3 days)
- **Total Estimate:** 5 Story Points (~5 Days)

2. Provide Chatbot Feedback

- **Tasks:**
 1. Design UI for feedback submission (2 days)
 2. Implement rating system (1–5 stars) (1 day)
 3. Store feedback in a database (2 days)
 4. Develop an admin dashboard for feedback analysis (2 days)
 5. Test feedback functionality (1 day)
- **Total Estimate:** 3 Story Points (~5 Days)

3. Admin Update FAQs and Orientation Data

- **Tasks:**
 1. Develop secure login system for admins (3 days)
 2. Implement role-based access control (2 days)
 3. Create an FAQ management dashboard (3 days)
 4. Integrate real-time FAQ updates with chatbot (3 days)
 5. Test admin functionalities (2 days)
- **Total Estimate:** 8 Story Points (~5 Days)

4. Scalability and Load Testing

- **Tasks:**
 1. Set up a test environment for load testing (2 days)
 2. Simulate 1,000 concurrent users with testing tools (3 days)
 3. Monitor system performance and adjust parameters (3 days)
 4. Optimize response time under 3 seconds (2 days)

- **Total Estimate:** 8 Story Points (~10 Days)

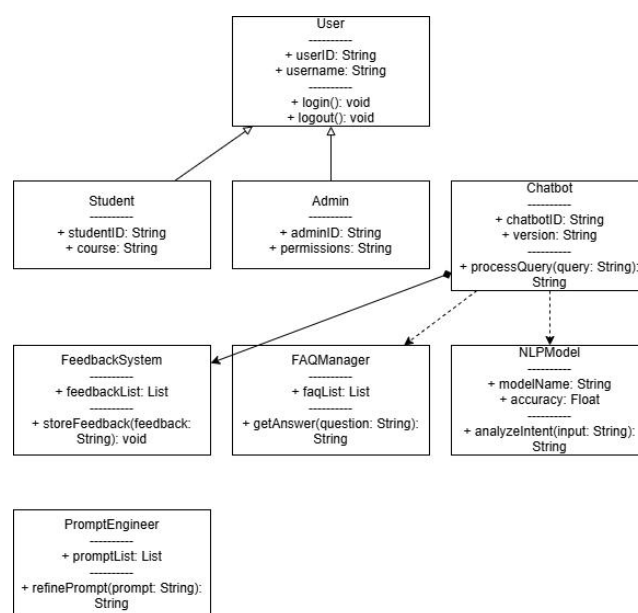
5. Accessibility Compliance

- **Tasks:**
 1. Ensure proper contrast and labeling (2 days)
 2. Implement keyboard navigation support (1 day)
 3. Verify screen reader compatibility (2 days)
 4. Conduct accessibility testing (1 day)
- **Total Estimate:** 3 Story Points (~5 Days)

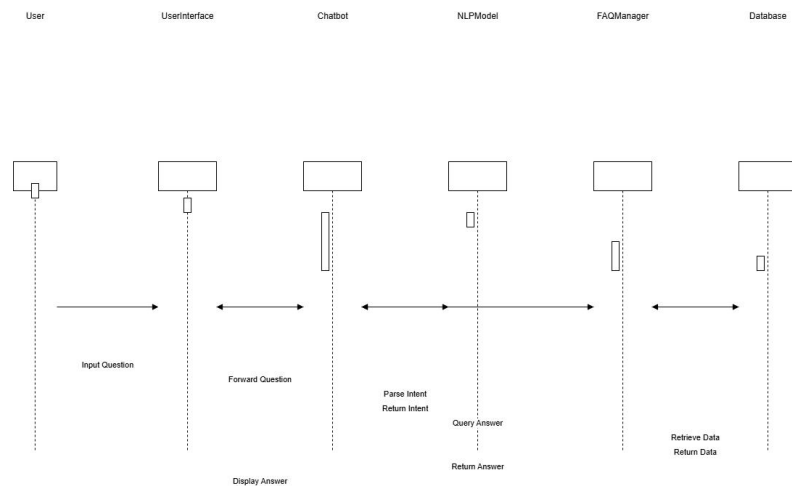
6. Prompt Engineering for Optimized Chatbot Responses

- **Tasks:**
 1. Develop structured prompt templates (3 days)
 2. Implement version control for prompts (3 days)
 3. Create a sandbox for prompt testing (4 days)
 4. Conduct test conversations for accuracy (3 days)
 5. Document best practices and guidelines (2 days)
- **Total Estimate:** 5 Story Points (~15 Days)

Activity 2:



Activity 3:



Activity 4:

Code:

```
class UserInterface:
    def get_input(self):
        return input("Ask a question: ")

    def display_response(self, response):
        print(f"Chatbot: {response}")

class Chatbot:
    def __init__(self):
        self.nlp = NLPModel()
        self.faq = FAQManager()

    def handle_query(self, question):
        intent = self.nlp.parse_intent(question)
        answer = self.faq.get_answer(intent)
        return answer

class NLPModel:
    def parse_intent(self, text):
        if "schedule" in text:
            return "orientation_schedule"
        return "general_query"
```

```
class FAQManager:
    def __init__(self):
        self.db = Database()

    def get_answer(self, intent):
        return self.db.fetch(intent)

class Database:
    def __init__(self):
        self.data = {
            "orientation_schedule": "Orientation starts on September 1st.",
            "general_query": "Please contact Student Affairs."
        }

    def fetch(self, key):
        return self.data.get(key, "No information found.")

ui = UserInterface()
chatbot = Chatbot()

question = ui.get_input()
response = chatbot.handle_query(question)
ui.display_response(response)
```