

NANYANG TECHNOLOGICAL UNIVERSITY**SEMESTER 2 EXAMINATION 2023-2024****SC2006/CZ2006 – SOFTWARE ENGINEERING**

Apr/May 2024

Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 5 pages.
 2. Answer **ALL** questions.
 3. This is an open-book examination.
 4. All questions carry equal marks.
 5. Refer to Appendix A on page 5 for the project description which is needed to answer some of the questions.
-
1. Based on the project description given in Appendix A,
 - (a) Identify the actors and use cases of the project and draw the use case diagram. Use <<include>> and <<extend>> relationships where appropriate. (19 marks)
 - (b) Write the use case description(s) for the functionality of the Reviewer agent. Your use case description must include the following elements: Actors, Pre-conditions, Flow of events and Alternative flows, if any. (6 marks)

2. (a) For the functionality of the Reviewer agent, identify the main classes and their associations and draw a conceptual class diagram. Your conceptual class diagram should clearly depict the stereotypes of each class. You do not need to identify any attributes or operations within the classes.
- (9 marks)
- (b) Draw a state machine diagram that shows the states and transitions of the QA-Checker agent.
- (7 marks)
- (c) Environments with changing requirements present additional challenges to plan-driven development methods. Discuss TWO ways to overcome these challenges.
- (4 marks)
- (d) When using the Scrum method in project development, explain why velocity is important.
- (5 marks)
3. (a) Based on the project description given in Appendix A,
- (i) Identify two agent roles which can be added into the CodeAgent platform.
- (3 marks)
- (ii) Identify the one important non-functional requirement for the CodeAgent platform.
- (3 marks)
- (iii) Propose an appropriate architecture for the CodeAgent platform and draw a detailed Class diagram with key attributes and methods in each class to reflect the architecture design. (Note: your solution should address the non-functional requirements identified in the question above)
- (8 marks)

Note: Question No. 3 continues on Page 3

- (b) Answer the following questions related to software design:
- (i) Propose a scenario in the CodeAgent platform where more than one design pattern can be applied together, and draw the class diagram to explain the scenario. (6 marks)
 - (ii) CodeAgent is an innovative approach for future software engineering activities. Please discuss whether CodeAgent can be used to replace software development teams and explain the reason. (5 marks)
4. (a) When a new software project requirement is given to the CodeAgent platform, the following information is required:
1. Project Description: the detailed project description, which must be in ASCII format.
 2. Development Time Limit (minutes): a 4-digit integer value.
 3. Launch Date: a date between 2000 and 2026.
 4. Application Type: one of “Desktop Application”, “Web Site”, “Mobile App”, “AI Application” and “All Types”.
- (i) Determine the equivalence classes for the above FOUR inputs. (4 marks)
 - (ii) Determine the boundaries of the equivalence classes identified in your answers to Q4(a)(i). For each boundary, identify a value on the boundary, a value just below the boundary, and a value just above the boundary. (4 marks)
 - (iii) You intend to perform **defensive testing** of the order information. Design a set of test cases to test the FOUR inputs based on the equivalence classes and boundary values identified in your answers to Q4(a)(i) and Q4(a)(ii). (7 marks)

Note: Question No. 4 continues on Page 4

- (b) QA-Checker in the CodeAgent platform plays a critical role in ensuring the quality and correctness of the software. Suggest how to design QA-Checker by using the testing techniques learned in this course to improve the effectiveness of the QA-Checker agent. (6 marks)
- (c) The CodeAgent platform can help to develop software. However, software continues to evolve based on new requirements. Suggest how to enhance the CodeAgent platform to support software maintenance and evolution. (4 marks)

Appendix A

CodeAgent: Automated Code Review System

Code review is an activity to identify bugs, increase code quality, and help developers learn the source code. A software agent is a computer program that performs automated tasks on behalf of a user or another program. A novel multi-agent-based system for code review, CodeAgent, is required to be developed by your team. In this multi-agent system, users configure large language model driven agents to act autonomously in the system on their behalf.

There are seven agents in CodeAgent system, including User, CEO, CPO, CTO, Reviewer, and Coder agents, representing six types of users of the system respectively and a special agent QA-Checker (short for “Question-Answer Checking”). They are defined for different specific tasks. For example, the Reviewer agent is to do the code review for given codes and files in three aspects: consistency detection between commit and commit message (CA), vulnerability injection detection (VA), and format consistency detection (FA), and then provide a detailed description of observation. Meanwhile, with the Reviewer’s assistance, Coder can process the code revision. All users need to log into the system in order to interact with their corresponding agent and the wider system.

When CodeAgent receives the request from a user to do the code review with the submitted commit, commit message, and original files, the request is processed in phases. In the first phase, CEO, CPO, and Coder will cooperate to recognize the modality of the input (e.g., document, code) and language (e.g., Python, Java, and Go). In the second phase, with the help of Coder, Reviewer will write an analysis report about consistency analysis, vulnerability analysis, format analysis, and suggestions for code revision. Then, in the third phase, according to analysis reports, Coder will align or revise the code if it finds incorrect snippets with the help of Reviewer. Coder cooperates with CTO and CEO to summarize the document and codes about the whole code review in the final phase.

All tasks are processed by the collaborative work of two agents in their multi-round conversations. These atomic conversations involve task-oriented role-playing between two agents, promoting collaborative communication. One agent works as the instructor and the other works as the assistant. Communication follows an instruction-following style, where agents interact to accomplish a specific subtask within each conversation and each conversation is under the supervision of the agent, QA-Checker.

If QA-Checker agent identifies that, the first answer is inappropriate for the initial question, it generates additional instructions attached to the original question and combines them to ask agents to further generate a different answer. The conversation between two agents is held until the generated answer is judged as appropriate by QA-Checker, it reaches the maximum dialogue times, otherwise. The QA-Checker is crucial in refining questions and answers within a conversation to ensure relevance and precision.

CodeAgent shall generate conversation logs and code review reports based on requests.

CZ2006 SOFTWARE ENGINEERING
SC2006 SOFTWARE ENGINEERING

Please read the following instructions carefully:

- 1. Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**
2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.
3. Please write your Matriculation Number on the front of the answer book.
4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.