CPE 593WS

Final Project - Milestone #1

Daniel Kang, Ryan Lee, Eric Song

Topic - Serializable Hashmap Dictionary

Our final project aims to research, design, and implement a Serializable Hashmap Dictionary, a data structure that combines the features of a hashmap and a dictionary while being serializable, for use in various software applications. The project involves literature review, algorithm design, coding, testing, and documentation.

Github Repository : https://github.com/danieljhkang/finalProject

Task Breakdown:

Individually choose and work on tasks to completion from the list below, moving on to the next task once completed.

Tasks :

- Define project scope, objectives, and deliverables.
- Conduct literature review on hashmaps, dictionaries, serialization techniques, and related data structures.
- Identify and research existing solutions to understand their strengths and limitations.
- Set up project environments and tools (IDE, version control, etc.).
- Define the data structure and API for the Serializable Hashmap Dictionary.
- Design serialization and deserialization algorithms compatible with the chosen programming language.
- Plan for error handling and exception scenarios.
- Document the design decisions and API specifications.
- Begin coding the Serializable Hashmap Dictionary based on the designed specifications.
- Implement serialization and deserialization functionalities.
- Test for individual components.
- Review code to ensure code quality and adherence to design principles.

Rough Project Planning :

March 26 - April 1: Planning and Research
- Define project goals and scope.
- Conduct initial research on existing solutions.
- Set up the project environment.

April 2 - April 8: Design
- Create data structure and API.
- Plan serialization and error handling.
- Document design decisions.

April 9 - April 15: Implementation

- Code Serializable Hashmap Dictionary.
- Implement serialization.
- Write unit tests.

April 16 - April 22: Integration and Testing
- Integrate components.
- Test thoroughly.
- Fix bugs.

April 23 - April 29: Documentation
- Write usage guides.
- Prepare documentation.
- Review and finalize.

April 30 - May 1: Optimization and Refinement
- Optimize performance.
- Refactor code.
- Test again.

May 1: Finalization and Delivery
- Prepare final deliverables.
- Conduct final tests.
- Submit project.

May 8: Presentation Day
- Prep presentation
- Actively watch other presentations
- Answer relevant questions

** Final Paper and Presentation will be continually worked on throughout each week