

Assignment Details: Design Document

Online Code Editor

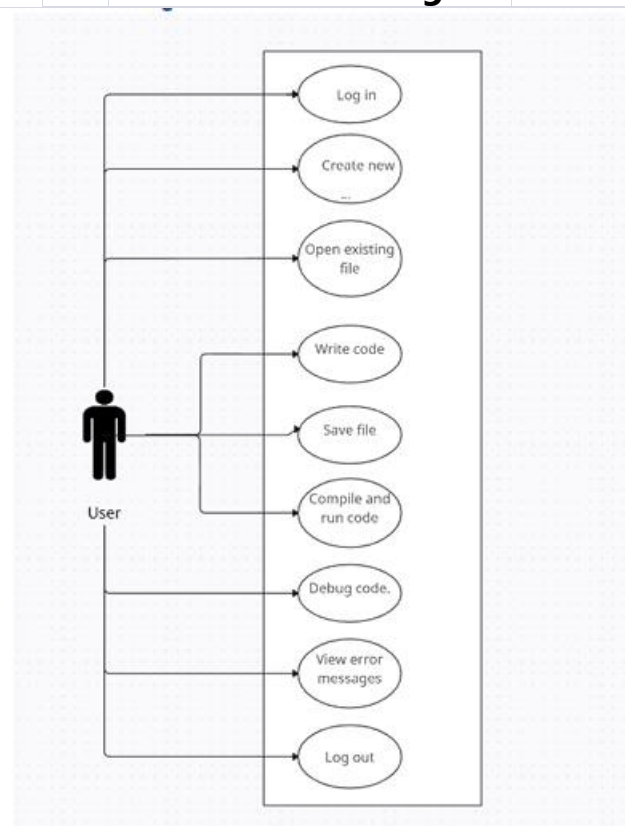
Y MANISH KUMAR - PES2UG21CS617

YASHWANTH RAO M P - PES2UG21CS623

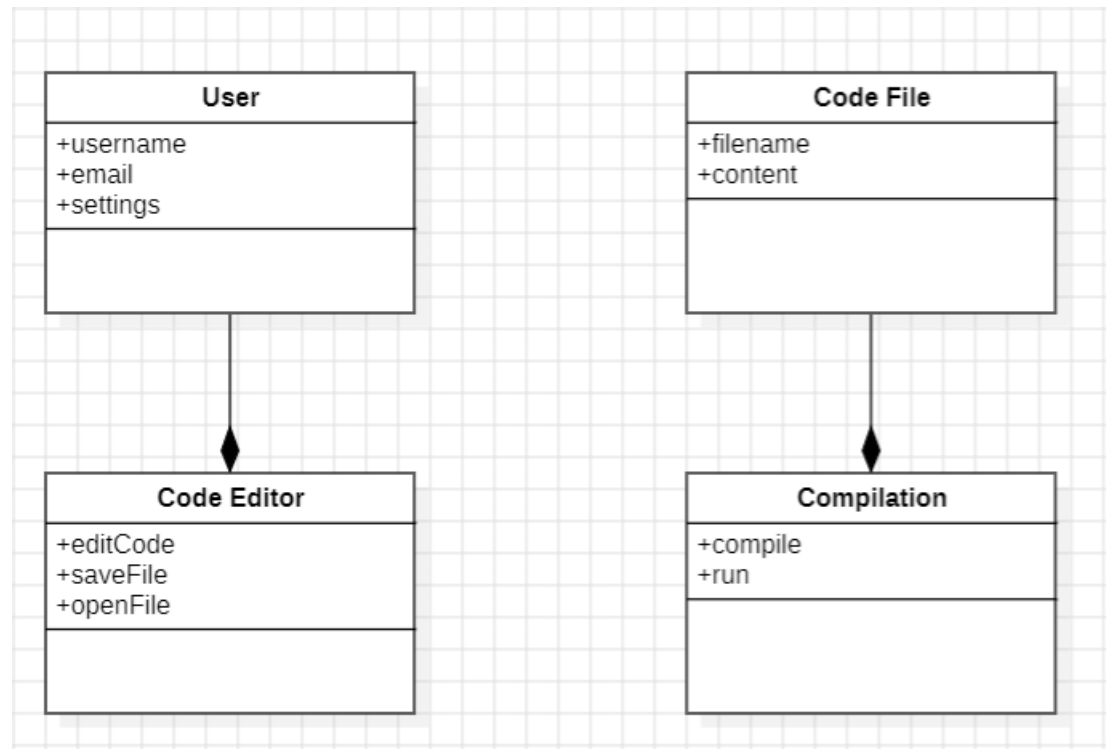
ARBAZ KHAN N R - PES2UG21CS801

RACHAN REDDY - PES2UG22CS901

1. Use case Class Diagram

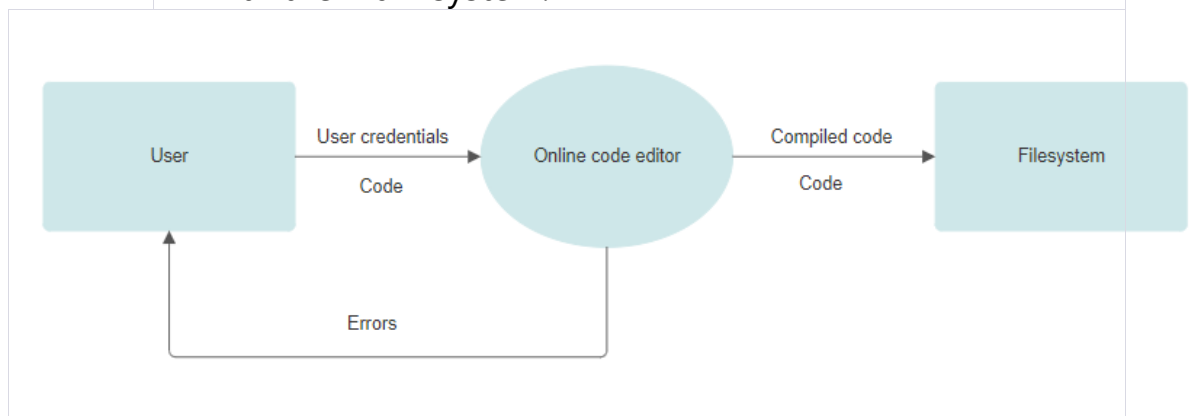


Class Diagram

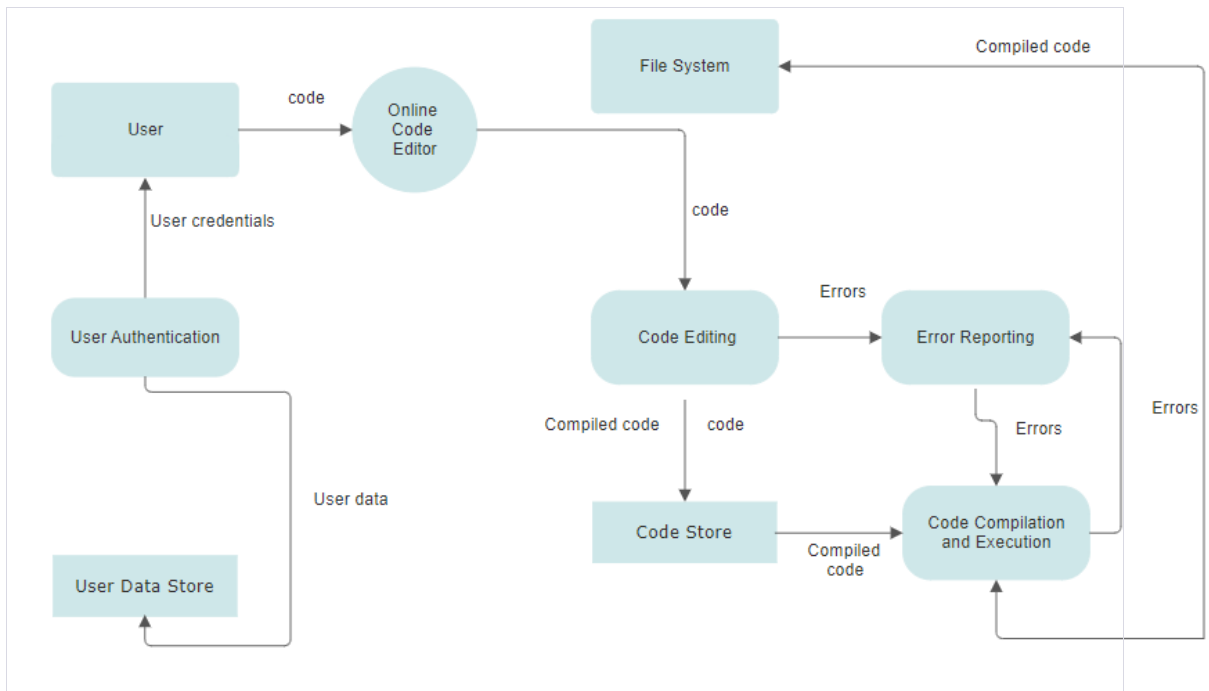


2. Incorporate DFDs:

- Develop a Data Flow Diagram (DFD) for your project:
 - **Level 0 (Context Diagram):** This is the top level of the DFD, which provides a bird's eye view of the system. It should include external entities and how they interact with the main system.



- **Level 1:** This expands the main system from Level 0 and shows its main functions. It should contain processes, data stores, and data flow among them.



3. Architectural Style Integration:

1. Modularity and Scalability:

- The Online Code Editor needs to accommodate users with varying levels of experience, from beginners to advanced coders.
- SOA allows breaking down the system into loosely coupled, independently deployable services.
- This modularity makes it easier to scale individual components, which is essential for handling a potentially large and diverse user base.

2. Maintainability:

- The platform should offer a user-friendly environment for learning and practice, and this requires frequent updates and improvements.
- SOA promotes reusability and maintainability by breaking the system into smaller, manageable services.
- This separation of concerns allows for easier updates and maintenance of individual services without disrupting the entire system.

3. Flexibility and Interoperability:

- The Online Code Editor might need to integrate with external learning resources and tools.
- SOA enables easy integration with external services and APIs.

- This facilitates access to learning resources, tutorials, and sample code by encapsulating external functionalities within well-defined service interfaces.

4. Performance:

- Collaborative coding and concurrent sessions can place varying demands on system resources.
- By distributing the workload across different services, you can optimize the performance of the system.
- Critical services, such as code editing and user authentication, can be deployed on high-performance servers, while allocating resources as needed for less critical services.

5. Security:

- Ensuring the security and privacy of user data and code is crucial for the Online Code Editor.
- SOA enables encapsulating security measures within individual services and controlling access through well-defined APIs.
- This enhances the security of the system, making it well-suited for protecting user data and code.

Justification:

The Online Code Editor is a complex web-based platform with various functionalities, including user registration, code editing, collaboration, assignment management, and more. SOA allows breaking down these functionalities into loosely coupled, independently deployable services, offering the flexibility to integrate external resources, optimize performance, and enhance security. This architectural style aligns with the requirements of the Online Code Editor and supports its scalability, maintainability, and performance needs.