***UML DİAGRAMS***

**Class Diagram:** A class diagram represents the static structure of the system, showcasing the classes, their attributes, methods, and relationships. Below is the class diagram for the Inventory Management System.

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Sequence Diagram:** The sequence diagram illustrates the interaction between objects in the system for key processes such as user authentication.

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Use Case Diagram:** The use case diagram presents the various functionalities available to the user within the system.

metin, çizgi, diyagram, taslak içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Class Diagrams**

**User Class**

* **Attributes**: username, password
* **Methods**: authenticate(), getUsername(), getPassword()
* **Encapsulation**: username and password are private attributes, ensuring data security.

**InventoryItem Class**

* **Attributes**: name, quantity, cost
* **Methods**: getQuantity(), setQuantity(), getCost(), setCost()
* **Inheritance**: Extends the Record class.
* **Polymorphism**: Overrides displayInfo() to provide specific inventory details.

**Project Class**

* **Attributes**: materials
* **Methods**: addMaterial(), listMaterials(), displayInfo()
* **Implements Interface**: Manageable

**Expense Class**

* **Attributes**: amount, description
* **Methods**: getAmount(), getDescription(), displayInfo()

**Sale Class**

* **Attributes**: item, quantity, price
* **Methods**: getQuantity(), getPrice(), displayInfo()

**Sequence Diagrams**

**Login Process**

1. The user inputs their username and password.
2. Inventory.login() is called.
3. The system fetches user credentials from the database.
4. If authentication is successful, mainMenu() is executed.
5. Otherwise, an error message is displayed.

**Design Justification**

**Encapsulation**

* Sensitive user data (username, password) is kept private.
* Getter and setter methods ensure controlled access to attributes.

**Inheritance**

* The Record class serves as a base for InventoryItem, Project, Expense, and Sale, ensuring code reusability.
* The Person class is extended by User to maintain a structured user hierarchy.

**Polymorphism**

* The displayInfo() method is overridden in multiple classes to present context-specific information dynamically.

**Abstraction**

* Record is defined as an abstract class, enforcing the implementation of displayInfo() in all derived classes.