

# Protocol: App Design and Structure

## 1. Overall Design

The app is designed with clear responsibilities for each part, making it easy to maintain and test. Each module focuses on a specific task. For example, models hold data, handlers manage logic, and interfaces define abstractions. The app uses core OOP concepts like encapsulation, inheritance, and polymorphism. Interfaces make it easier to extend the app and create mock versions for testing (later on).

## 2. Folder Structure and Purpose

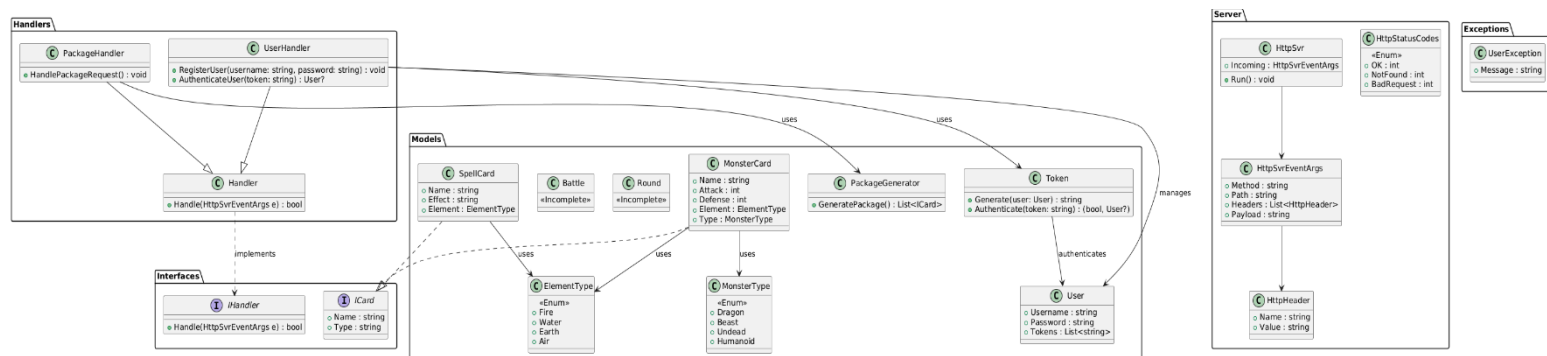
- **Handlers:**
  - `UserHandler.cs`: Manages user-related operations
  - `PackageHandler.cs`: Handles card packages
- **Interfaces:**
  - `IHandler.cs`: A base interface for handlers
  - `ICard.cs`: Interface for cards (`MonsterCard`, `SpellCard`)
- **Models:**
  - `User.cs`: Represents user data and behavior
  - `Token.cs`: Static class for generating and authenticating tokens for user sessions.
  - `MonsterCard.cs`, `SpellCard.cs`: Define card types for game
  - `Battle.cs`: Incomplete; handles the logic for card battles.
  - `Round.cs`, `Battle.cs`: Encapsulate game round and battle logic
  - `PackageGenerator.cs`: Responsible for package creation
  - `ElementType.cs`, `MonsterType.cs`: Enumerations for categorization
  -
- **Exceptions:**
  - `UserException.cs`: Handles user-specific errors

- **Server:**
  - HttpSvr.cs: Main HTTP server implementation
  - Handler.cs: Centralizes request handling
  - HttpHeader.cs, HttpStatusCode.cs: Support HTTP communication
  - HttpSvrEventArgs.cs: Parses HTTP requests into method, path, headers, and payload; allows responses with status codes.
  - HttpSvrEventHandler.cs: Delegate for handling server events.
- **Others:**
  - curlSkript.txt: Contains cURL commands for interacting with the API (e.g., registration, login, package purchase).

### 3. Class Interactions and Key Design Decisions

- **Centralized Logic via Handlers:** Handlers manage interactions between models and server components, reducing coupling.
- **Interfaces for Flexibility:** Using interfaces like IHandler and ICard ensures modules can be swapped or extended without breaking functionality.

### 4. UML-Diagram



### 5. GIT-Repository

[https://github.com/EZST99/Monster\\_Trading\\_Cards.git](https://github.com/EZST99/Monster_Trading_Cards.git)