# SWEN1 Monster Card Trading Game Protocol #1

The application is composed of two main components: a **server** and an **application**. The server operates independently and can interface with any application that implements the required interface, ensuring flexibility and reusability. The application itself is organized into several folders:

- **Controllers**: Handle incoming requests, orchestrate application logic, and return appropriate responses.
- **Entities**: Represent the core data models mapped to the database.
- **Exceptions**: Contain custom exception classes for handling errors consistently.
- **Repositories**: Manage storage operations, providing an abstraction layer for data access.
- **Routing**: Define application endpoints and map them to corresponding controllers.
- **Services**: Implement the core business logic and interact with repositories.
- **Util**: Include utility classes and helper functions used throughout the application.

### Centralized Exception Handling

A unique implementation of note is the centralized Exception handling. The ExceptionHandler class provides a unified mechanism for managing exceptions and generating consistent HTTP responses. A method takes a Supplier<Response> and executes it within a try-catch block, mapping specific exceptions (e.g., BadRequestException, NotFoundException) to corresponding HTTP status codes. A generic RuntimeException is also caught to handle unexpected errors, logging the issue and returning a 500 status. This design centralizes error handling, reducing code duplication and ensuring uniformity in error responses across the application.

https://github.com/noahhowadt/monster-card-trading-game