GAME SOLUTION

Purpose of the document

The purpose of the document is to explain the design and architecture of the solution.

Technologies Used

- JAVA 8
- Hibernate/JPA
- In memory H2 database
- Spring Boot
- Junit / Mockito
- Guava libraries

How to Run

For running the Netent-Game project just go the project folder where pom.xml is situated and execute "mvn test" OR "mvn clean install".

Betting Game can be started by running the main class- BettingGameApplication.java.

Approach

The project is split into two different modules which is
1.BettingGame- BettingGame consists of all the betting game specific code like Controller, Service and DAO.
2. Framework- All the reusable code which is generating random number and winning amount entity which can be used across different games is in this module.

Rest- Client makes sure to play Free round immediately.

Random Generation

The random number generation is done using Math.random() which gives the range from 0 to 1.

Data Model

Schema is designed according to the entities which will be required to be persisted.

1. RoundRequest:- An entity which consists of coins and roundtype(Normal or Free). Based on the coins and roundtype different validations are in place.
2. RoundResult:- An entity which is persisted into the DB after each round which has all the required data points containing round response.
3. WinningAmount:- An entity with roundId and winningamount which can be used across different games.

Code Design

Code is mainly divided in three layers. Controller layer handling all URI for betting and WinningAmount. Service is there for the business logic and DAO layer i.e. classes handling all database interaction. All classes are covered with JUNIT test cases.

Testing

All classes are covered with Junit using Mocking. For IntegrationTesting, using Rest-Assured which makes the end to end testing possible.