

# Hackermans – System specification

## Inhalt

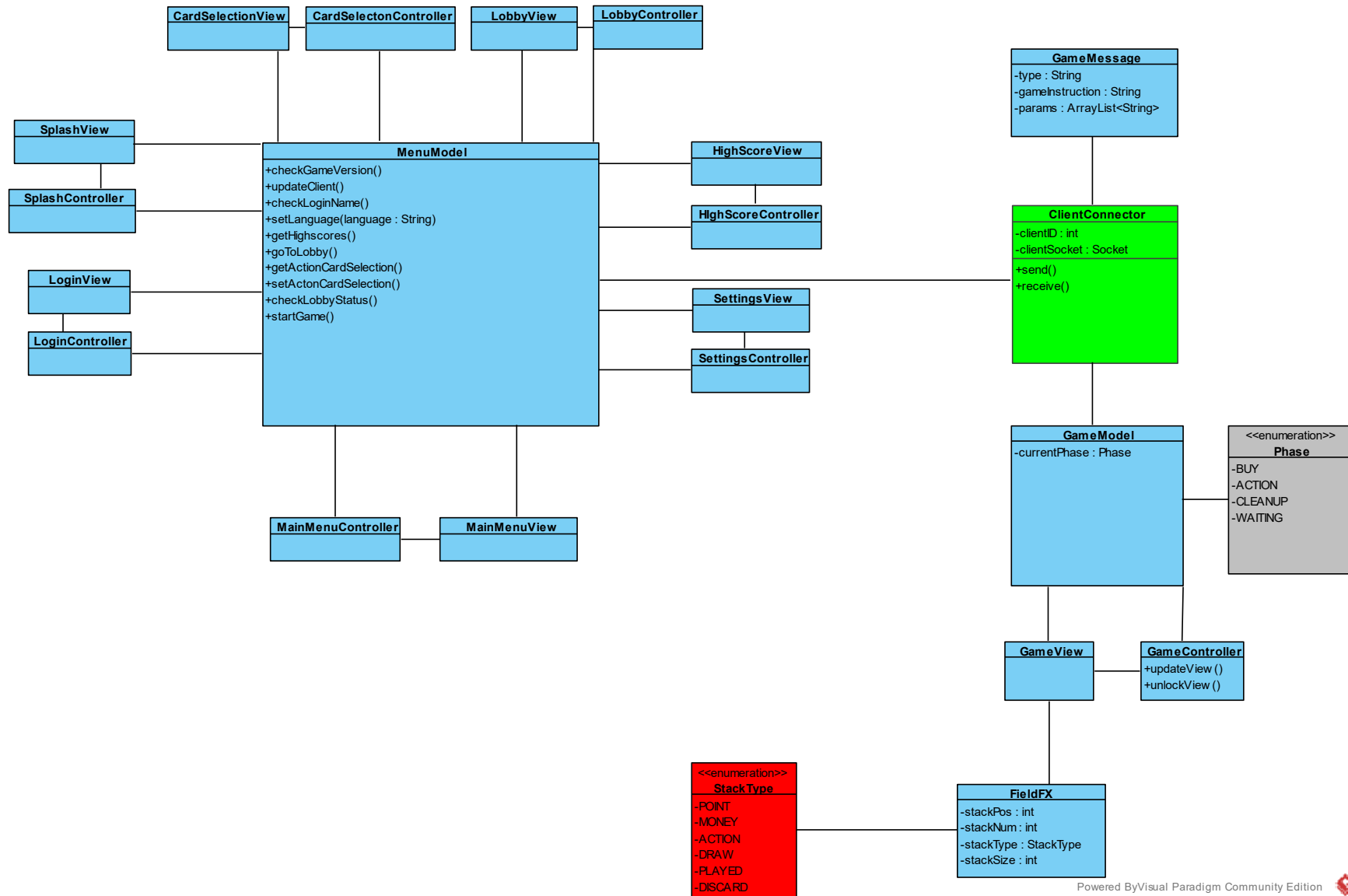
1. Generelles.....	1
2. Datenstrukturen.....	2
2.1. Klassendiagramm Client .....	2
2.2. Klassendiagramm Server .....	3
3. Client/Server Kommunikation .....	4
4. Datenbank (Pascal).....	5
4.1. ER-Diagramm .....	5
4.2. Rel. Database Shema: .....	5
4.3. Physical Database Shema: .....	5

## 1. Generelles

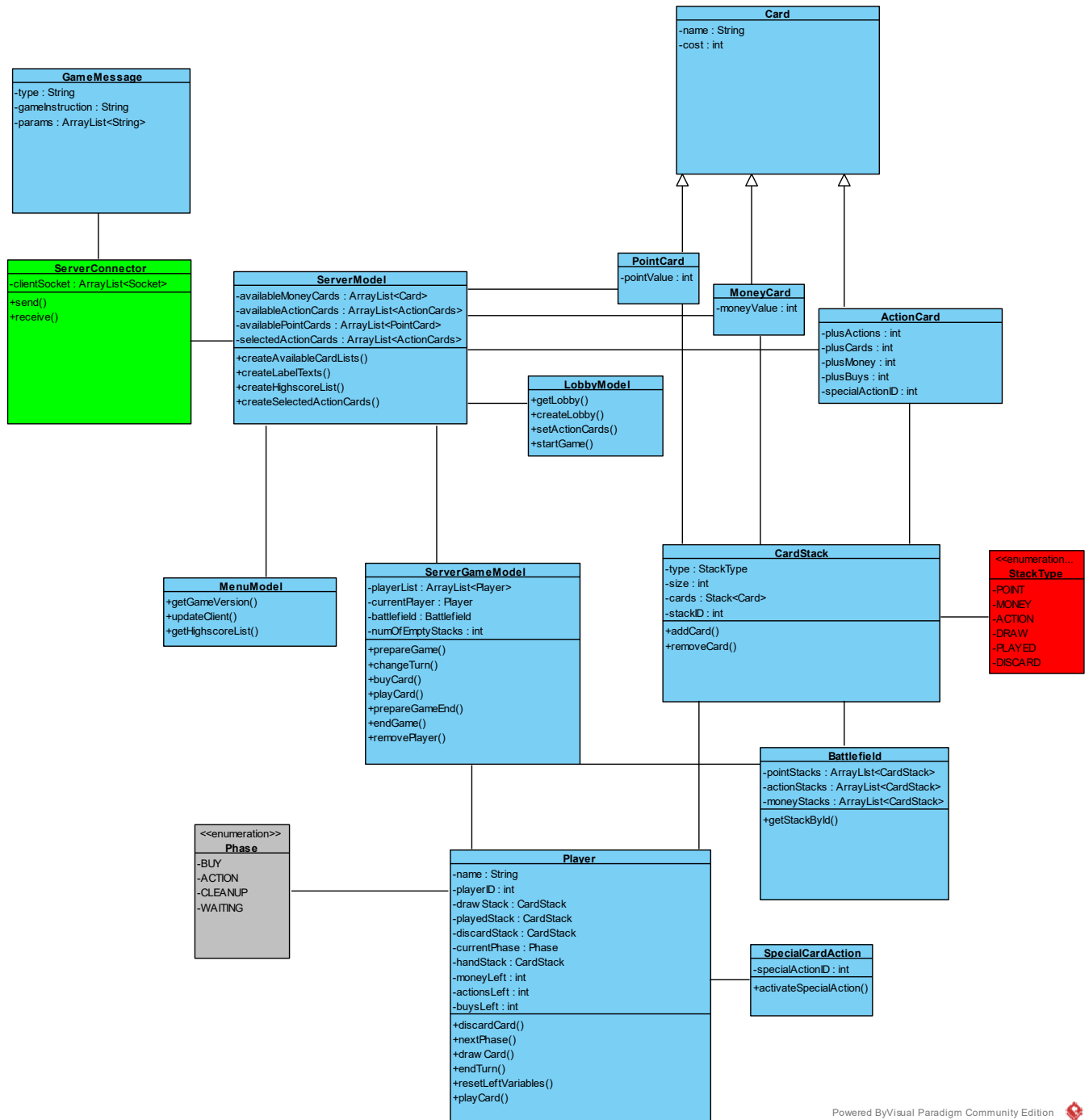
Das Spiel wird in Java mit Hilfe von JavaFX programmiert und folgt generell einer MVC-Struktur. Es läuft in einem klassischen Client/Server Modell. Der Client prüft dabei nur grundlegende Szenarien (Thin-Client). Die komplette Programmlogik läuft auf dem Server. Der Server ist mit einer MySQL-Datenbank verbunden. Die MySQL-Datenbank enthält alle Kartendaten, sowie Highscore-Listen, welche vom Client abgerufen werden.

## 2. Datenstrukturen

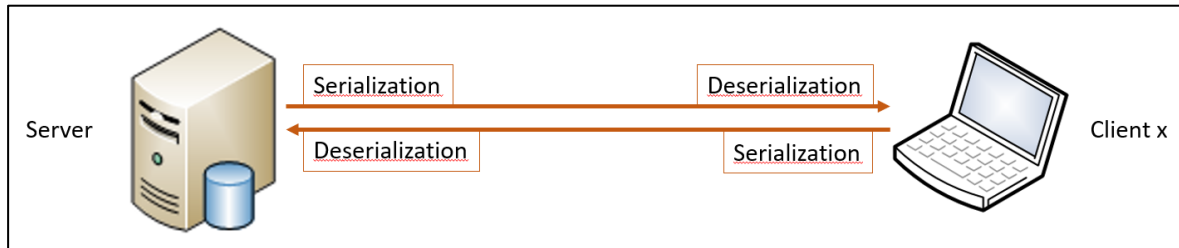
### 2.1. Klassendiagramm Client



## 2.2. Klassendiagramm Server



### 3. Client/Server Kommunikation



**Server:** Der Server läuft auf einem normalen PC im selben Netzwerk wie die Clients. Die MySQL-Datenbank läuft lokal auf dem Server.

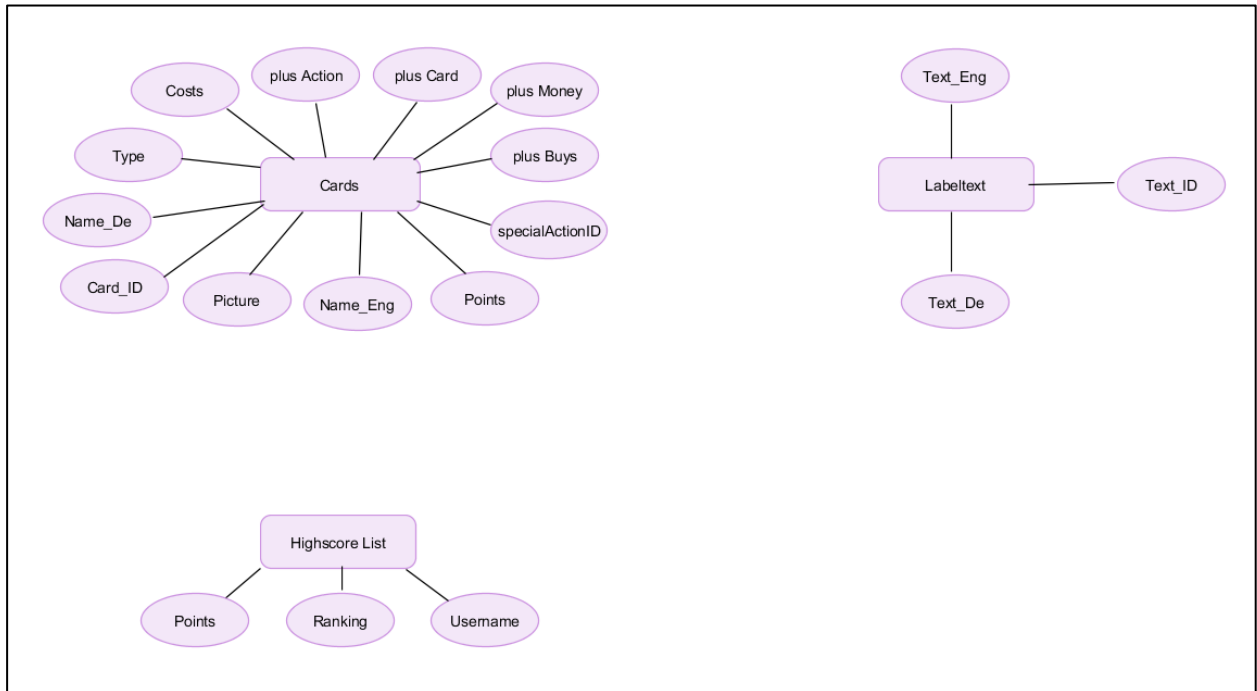
**Client:** Es wird eine Thin-Client Struktur verwendet. Der Client führt dabei nur minimale Logikprüfungen aus. Die restliche Logik wird auf dem Server geprüft. Die Kommunikation zwischen Server und Client erfolgt mithilfe von Serialisierung. Zu diesem Zweck wird eine eigene GameMessage-Klasse definiert, deren Objekte serialisiert und übermittelt werden.

**Sequenzfluss:** Der Client startet mit dem Start der Anwendung die Kommunikation mit dem Server. Diese Verbindung wird bis zum Schliessen der Anwendung aufrecht erhalten. In den Spielmenüs sendet der Client bei entsprechendem User-Input (Bsp. Aufrufen der Highscore-Liste) eine Nachricht an den Server und wartet anschliessend auf dessen Antwort. In der Lobby sowie im Spiel, empfängt der Client permanent Nachrichten des Servers.

**Vorbedingung:** Zur Sicherstellung, dass die Clients nur im richtigen Zeitpunkt Nachrichten schicken können, wird ein GUI-Sperrmechanismus verwendet.

## 4. Datenbank (Pascal)

### 4.1. ER-Diagramm



### 4.2. Rel. Database Shema:

Cards (Card\_ID, Picture, Type, Name\_Eng, Name\_DE, Points, Costs, plus\_Action, plus\_Cards, plus\_Money, plus\_Buys, specialActionID)

Highscore\_List (Ranking, Username, Points)

Labeltext(Text\_ID, Text\_Eng, Text\_DE)

### 4.3. Physical Database Shema:

