



AG Human Computer Interaction  
apl. Prof. Dr. Achim Ebert

Software-Entwicklungs-Projekt 2025

---

# Cosmic Eidex

## Anforderungsspezifikation

16. April 2025

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Ziel	3
1.2	Technologien	3
<b>2</b>	<b>Funktionen</b>	<b>4</b>
2.1	Hauptfunktionen	4
2.2	Registrierung	4
2.3	Benutzeroberfläche	4
2.4	Computerspieler	5
2.5	Spielregeln	5
<b>3</b>	<b>Qualitätsanforderungen</b>	<b>6</b>
<b>4</b>	<b>Anforderungen an den Softwareentwicklungsprozess</b>	<b>7</b>
4.1	Qualitätssicherung	7
4.2	Dokumentation	7

# 1 Einleitung

## 1.1 Ziel

Da sich Kartenspiele seit Jahrhunderten großer Beliebtheit erfreuen, soll diese Art der gesellschaftlich wertvollen Freizeitgestaltung auf das nächste Level gehoben werden. Um diesem Anspruch gerecht zu werden, soll eine Client-Server-Applikation erstellt werden. Durch dieses System soll es Nutzer:innen möglich sein, über den Server gemeinsam verschiedene Kartenspiele zu spielen und zu chatten. Durch eine Bestenliste sollen die Nutzer:innen angespornt werden, möglichst gut zu spielen und sich gegenseitig zu motivieren.

Der Client soll eine benutzerfreundliche GUI anbieten, die es den Spielenden ermöglicht, aktiv am Spiel teilzunehmen. Des weiteren sollen fehlerhafte Spielzüge des Nutzers oder der Nutzerin verhindert werden.

Der Server verwaltet alle vorhandenen Spiele. Die Clients können neue Spiele anlegen und existierenden Spielen beitreten. Mehrere Clients können an einem Spiel gleichzeitig teilnehmen, wobei immer nur so viele Clients gleichzeitig aktiv spielen können, wie es die Regeln des Spiels zulassen.

Das Spiel ist für Privatanwender:innen ausgelegt. Eine kommerzielle Verwendung – beispielsweise als Turniersoftware – ist nicht vorgesehen.

## 1.2 Technologien

Für die Implementierung soll Java 8 oder höher verwendet werden. Die Software soll ausgiebig mit JUnit 4 oder höher getestet werden.

## 2 Funktionen

Im Folgenden werden die geforderten Grund-Funktionen sowohl des Servers als auch des Clients spezifiziert.

### 2.1 Hauptfunktionen

Das Programm bietet die Möglichkeit, sich mit einem Benutzernamen anzumelden und verwaltet die Benutzer:innen.

Ebenso verwaltet das Programm sogenannte Spielräume in einer Lobby. Ein Spielraum ist ein virtueller Raum, in dem sich mehrere Benutzer:innen befinden können. Spielräume sollen aus der Lobby heraus erzeugt werden können. Aus der Lobby soll man Spielräumen beitreten können. In der Lobby sollen die vorhandenen Spielräume und die angemeldeten Benutzer:innen ersichtlich sein. Für jeden Raum soll schon aus der Lobby ersichtlich sein, welche Benutzer:innen sich bereits in ihm befinden.

Benutzer:innen in einem Raum sollen gemeinsam das Spiel spielen können. Sobald in einem Spielraum ausreichend viele Spieler:innen vorhanden sind, kann das Spiel gestartet werden.

Eine Chatfunktion ist sowohl in der Lobby als auch in den Spielräumen vorgesehen. Ebenso lässt sich eine Liste der Benutzer:innen in einem Raum anzeigen.

Eine Raumübergreifende Bestenliste soll stets aktuell anzeigbar sein.

### 2.2 Registrierung

Bevor ein(e) Benutzer:in spielen kann, muss er/sie sich am Server mit einem Nickname registrieren und ein Passwort festlegen. Sollte ein Nickname schon belegt sein, wird die Registrierung nicht akzeptiert.

### 2.3 Benutzeroberfläche

Die grafische Oberfläche soll alle spielrelevanten Operationen und Daten sowie alle Operationen und Daten, die zum Verwalten der Räume notwendig sind, einfach

zugänglich und intuitiv bedienbar gestalten. Die Spielregeln und die Verwendung der Software sollen Benutzer:innen direkt durch das Programm selbst klar werden.

Ein Spielraum soll zusätzlich zu den eigenen Spielkarten auch mindestens die Anzahl der Karten der jeweiligen Gegenspieler:innen zeigen. Alle Stapel des restlichen Spielfelds sollen jederzeit sichtbar sein. Alle Karten sollen einfach zu erkennen sein.

Die Verteilung der Gewinnpunkte soll für die Spieler:innen ersichtlich sein. Alle Informationen des Spiels sollen im Spielraum klar ersichtlich sein.

Spielaktionen sollen eindeutig und in ihrer zeitlichen Abfolge erkenntlich und nachvollziehbar sein.

## 2.4 Computerspieler

Ein Computerspieler (Bot) bietet die Möglichkeit eine(n) menschliche(n) Spieler:in zu ersetzen und in allen Spielsituationen adäquat zu reagieren. Dabei soll der Computerspieler in seinem Spiel eine möglichst gewinnbringende Strategie verfolgen. Es sollen zwei Bots implementiert werden: ein grundlegender einfacher Bot und ein komplexerer besserer Bot.

## 2.5 Spielregeln

Als Kartenspiel muss *Cosmic Eidex* realisiert werden. Die zu verwendenden Regeln sind beispielsweise zu finden unter: [https://www.spielezar.ch/modules/genzo\\_zar/views/pdf/spielregeln-cosmic-eidex.pdf](https://www.spielezar.ch/modules/genzo_zar/views/pdf/spielregeln-cosmic-eidex.pdf) [Link zu den Regeln auf [spielezar.ch](https://www.spielezar.ch)]. Die Implementierung anderer Spielvarianten ist nicht geplant.

### **3 Qualitätsanforderungen**

Um der Zielgruppe gerecht zu werden muss das System leicht zu bedienen und zu erlernen sein.

Damit das System von der Zielgruppe akzeptiert wird, muss es ausreichend robust sein, sodass ein normales Spielverhalten das Spiel nicht zum Absturz bringen kann.

## 4 Anforderungen an den Softwareentwicklungsprozess

Um ein gewisses Maß an Zuverlässigkeit sowie Wartbarkeit zu gewährleisten, werden zusätzlich zur reinen Implementierung oben genannter Funktionen folgende Anforderungen gestellt.

### 4.1 Qualitätssicherung

Mit Ausnahme der GUI gilt es, für jede Klasse Tests mittels JUnit zu erstellen. Dazu müssen Äquivalenzklassen für jede zu testende Methode erstellt und im Quelltext der Tests dokumentiert werden. Die Unit-Tests müssen eine Coverage von mehr als 70% aufweisen. Des Weiteren müssen auch Systemtests durchgeführt werden. Jede Ausführung von Systemtests muss detailliert und für beliebige Leser problemlos nachvollziehbar dokumentiert werden. Die Konsistenz der einzelnen Artefakte untereinander ist sicherzustellen.

### 4.2 Dokumentation

Der Quelltext ist mittels Javadoc vollständig zu kommentieren. Sowohl Javadoc als auch sonstige Kommentare sollen über das Gesamtprojekt hinweg *einheitlich* in Englisch oder Deutsch geschrieben werden.