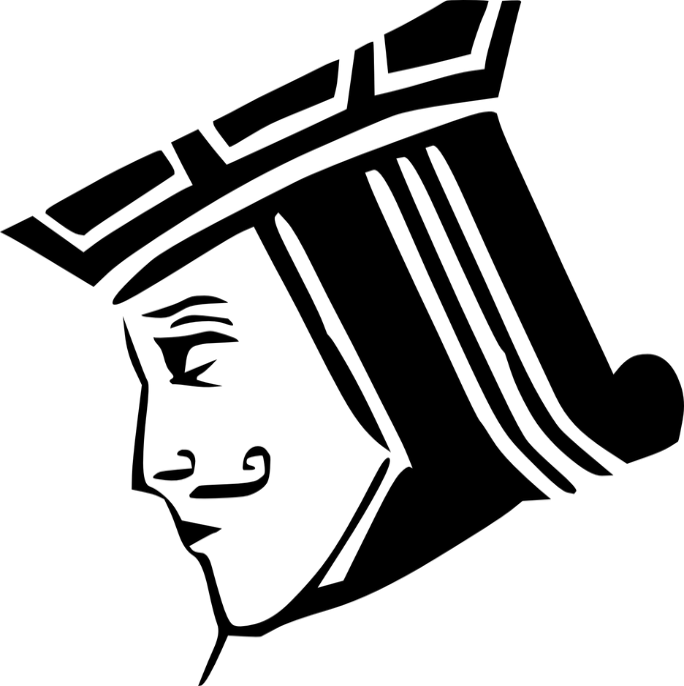


LB 426 Blackjack



https://pixabay.com/de/vectors/blackjack-schwarzer-jack-poker-303154/

**Autoren:**

Alek Cvetkovski, Elbish Bhusal,Matteo Jakob

**Klasse:**

*I21a/I21v*

Inhaltsverzeichnis

[User stories 3](#_Toc131133038)

[Design pattern 4](#_Toc131133039)

[Projekt 5](#_Toc131133040)

[Einteilung 5](#_Toc131133041)

[Branches 5](#_Toc131133042)

[SOLID 6](#_Toc131133043)

[Bewertung 6](#_Toc131133044)

[Verbesserung 6](#_Toc131133045)

[Retrospektive 6](#_Toc131133046)

# User stories

|  |  |
| --- | --- |
| ID | User story |
| 1 | Als Benutzer möchte ich mich auf der Webseite registrieren und ein Konto erstellen, um BlackJack spielen zu können. |
| 2 | Als Benutzer möchte ich eine Anleitung zum Spiel erhalten, damit ich die Regeln und Strategien verstehen kann. |
| 3 | Als Benutzer möchte ich die Möglichkeit haben, gegen einen Computer zu spielen, damit ich auch alleine spielen kann. |
| 4 | Als Benutzer möchte ich die Möglichkeit haben, verschiedene Einsätze zu tätigen, damit ich mein Spiel anpassen kann. |
| 5 | Als Benutzer möchte ich die Möglichkeit haben, meine letzten Gewinne und Verluste zu verfolgen, um mein Spielverhalten zu analysieren. |
| 6 | Als Benutzer möchte ich die Möglichkeit haben, verschiedene Zahlungsmethoden (fake) zu nutzen, um meine Einsätze zu tätigen. |
| 7 | Als Spieler möchte ich auf der Blackjack Webseite ein unterhaltsames und herausforderndes Kartenspiel erleben, bei dem ich gegen den Dealer antrete, um so nah wie möglich an die Zahl 21 heranzukommen und das Spiel zu gewinnen. |

# Design pattern

Wir benutzen Observer- und Strategypattern.

Ein Designpattern ist ein wiederkehrendes Problem in der Softwareentwicklung, das durch bewährte Lösungsansätze gelöst werden kann. Es handelt sich um eine Art Blaupause oder Vorlage, die beschreibt, wie bestimmte Softwareprobleme am besten gelöst werden können.

Das Observer-Pattern ist ein Designpattern, bei dem eine Liste von Abhängigkeiten zwischen Objekten verwaltet wird. Wenn sich ein Objekt ändert, werden alle abhängigen Objekte automatisch benachrichtigt und aktualisiert. Dieses Pattern ist nützlich, wenn Sie eine lose Kopplung zwischen Objekten erreichen möchten, um eine flexible und erweiterbare Architektur zu schaffen.

Strategy-Pattern. Dieses Pattern ermöglicht es, verschiedene Algorithmen und Strategien innerhalb einer Klasse auszutauschen, um die Funktionalität der Klasse zu ändern, ohne die Klasse selbst zu ändern.

In der Blackjack-Webseite könnte dieses Pattern genutzt werden, um verschiedene Spielstrategien bei dem Computer zu implementieren, wie zum Beispiel eine aggressive oder konservative Spielweise.

Der Einsatz des Strategy-Patterns ist sinnvoll, da es eine flexible und erweiterbare Architektur schafft, die es ermöglicht, verschiedene Strategien und Algorithmen innerhalb einer Klasse auszutauschen. Auf diese Weise kann die Blackjack-Webseite einfach erweitert und an die Bedürfnisse der Benutzer angepasst werden, ohne die Grundstruktur der Webseite zu verändern.

# Projekt

## Einteilung

US = User story

|  |  |
| --- | --- |
| Name | Beschreibung |
| Elbish | US1, US6, US4 |
| Alek | US2, Frontend |
| Matte | US3, US5, US7 Frontend |

# Branches

|  |  |  |
| --- | --- | --- |
| Userstory ID | Branch | Beschreibung |
| 1 | 1-Account | Der Benutzer soll sich ein ... |
| 2 | 2- |  |
| 3 | 3- |  |
| 4 | 4- |  |
| 5 | 5-Spielerverhalten | .. |
| 6 | 6- |  |
| 7 | 7- |  |

## SOLID

### Bewertung

123

### Verbesserung

123

# Retrospektive

Hinderniss:

Loesung:

Hinderniss:

Loesung:

Hinderniss:

Loesung:

