Casino

Thema: Project work Module 226a

Documentinformation

File name: CasinoDok

Save date: 02.11.2021

Authorinformation

Autor: Enes Spahiu

Content

[1 Introduction 3](#_Toc87195535)

[2 Plan 4](#_Toc87195536)

[3 Use Cases 5](#_Toc87195537)

[3.1 Roulette 5](#_Toc87195538)

[3.1.1 Inner Bet 5](#_Toc87195539)

[3.1.2 Outer Bet 6](#_Toc87195540)

[3.2 Blackjack 7](#_Toc87195541)

[3.3 Slot machine 8](#_Toc87195542)

[4 Sequence Diagram 9](#_Toc87195543)

[5 Class diagram 10](#_Toc87195544)

History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Details** |
| 1.0 | 21.09.2021 | Enes Spahiu | Created document |
| 1.1 | 26.10.2021 | Enes Spahiu | Added plan, class diagram and introduction |
| 1.2 | 07.11.2021 | Enes Spahiu | Finished documentation |

# Introduction

The following document describes my project work for Module 226a, how I proceeded, and what I accomplished. My project is a casino where you can easily play, it includes a slot machine, a roulette table, as well as a blackjack table.

The Project also includes a database which stores all user data and is also there, so it is more realistic. The program can register a new user but also log in a user that already exists that if a user with an already registered account can play with the amount of money he already paid in if a new user comes, he can make an account which is protected by a password chosen by the user himself.

The casino is prepared for every user it catches all exceptions so the program doesn’t crash no user can type in wrong inputs at points they are not supposed to.

# Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 21.09 | 28.09 | 26.10 | 02.11 | Free time |
| Choose project |  |  |  |  |  |
| Create GitHub Repository |  |  |  |  |  |
| Plan the project |  |  |  |  |  |
| Make Use-Cases |  |  |  |  |  |
| Create a Class diagram |  |  |  |  |  |
| Coding |  |  |  |  |  |
| Implement Exception handling |  |  |  |  |  |
| Create a Sequence diagram |  |  |  |  |  |
| JUnit testing |  |  |  |  |  |
| Work on Documentation |  |  |  |  |  |

# Use Cases

## Roulette

First you can choose which bet you want to take either an “Inner Bet” or an “Outer Bet” or you can end the game.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Inner Bet

If you choose to play an Inner Bet you can make an Straight Up Bet.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

You can then just type in numbers which are checked, so you can not type in wrong inputs.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

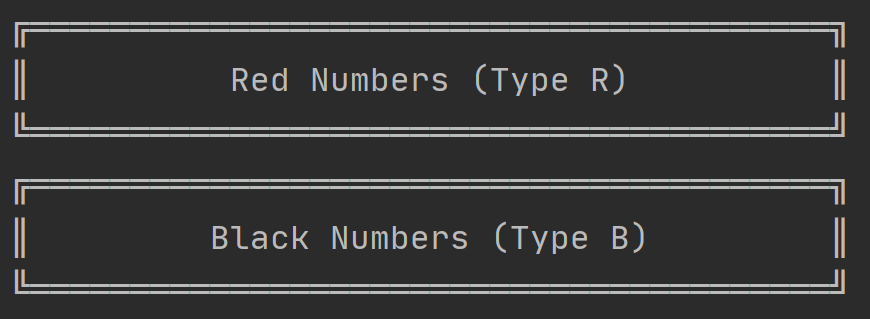
### Outer Bet

If you choose to make an outer bet you can choose between:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

In this Case I will choose “Red / Black”



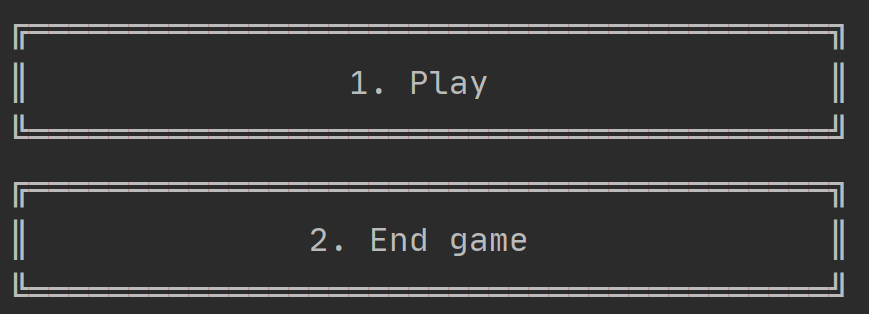
If I choose one of both, it will check which color the winning number has and then it prints out what the Color is and what the winning number is.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Blackjack

If you choose to play blackjack you can either choose to start the game or end the game.



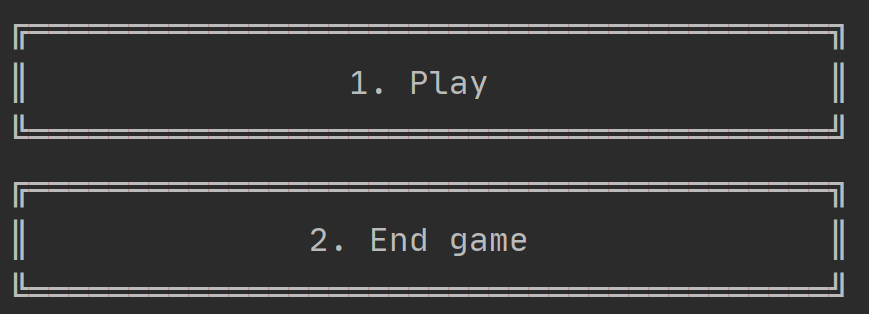
If you start the game you type in the amount you want to bet and then you have to decide if you want to draw a card or stop drawing if you stop it will compare your value with the value of the dealer.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Slot machine

If you choose to play slots you can either choose to start the game or end the game.

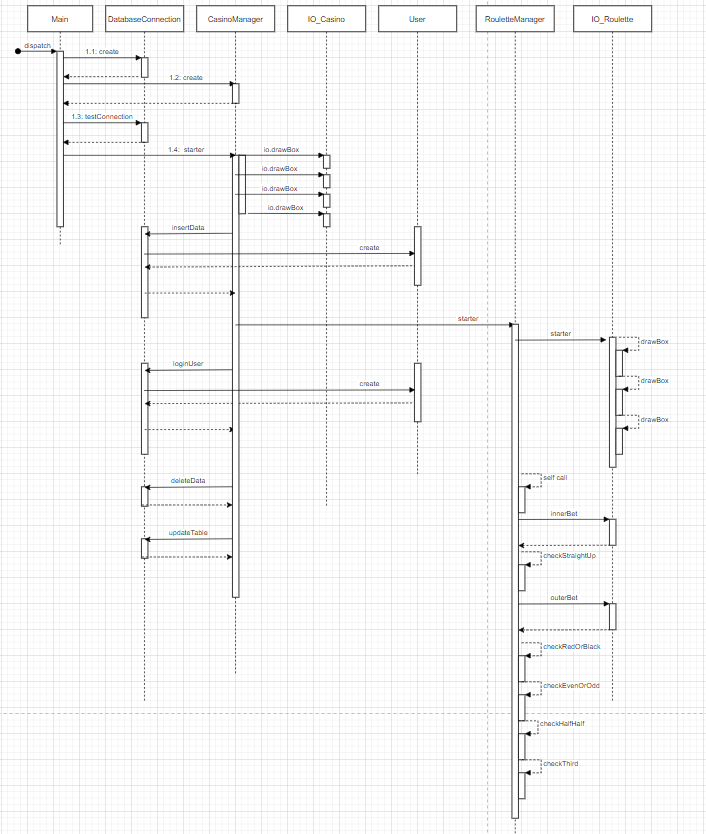


If you decide to start the game you just have to type in the amount you want to bet and then it will start spinning at the end it will be checked if you won or not.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Sequence Diagram



# Class diagram

My class diagram is based on the casino program it also implements if the relations between two classes are an aggregation or a composition.

