SCREW YOUR NEIGHBOR

Report for Milestone 3 of Sopra FS22, Group 36

Figure 1: cover (source B. Furrer)

Members

Carmen Kirchdorfer (20-720-132)
Salome Wildermuth (10-289-544)
Beat Furrer, group leader (07-542-392) Lucius Bachmann (11-060-274)
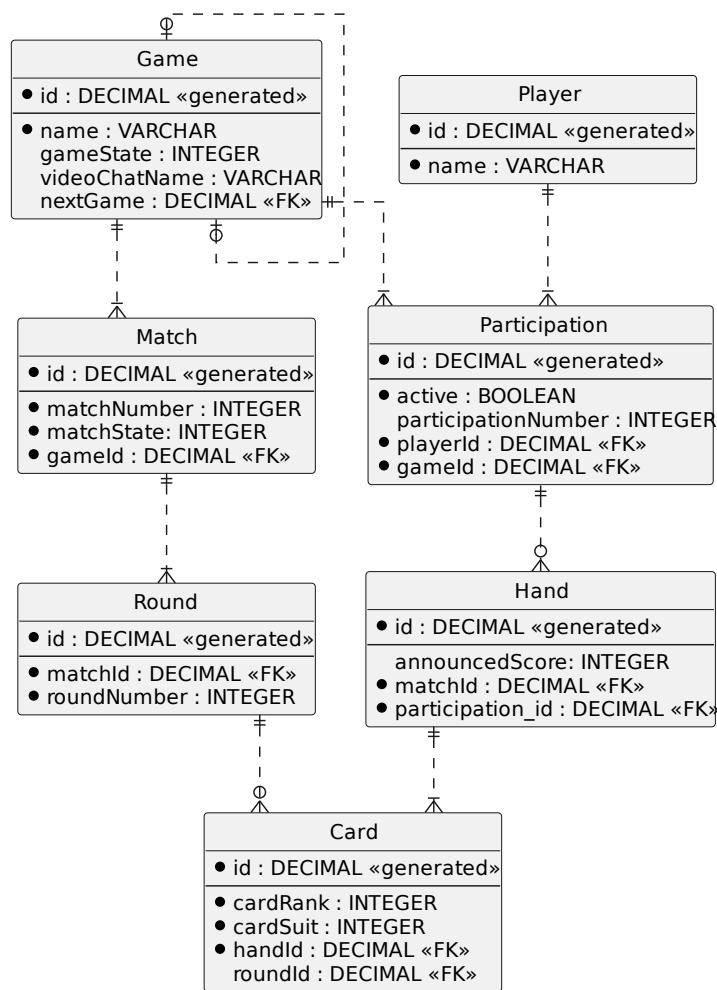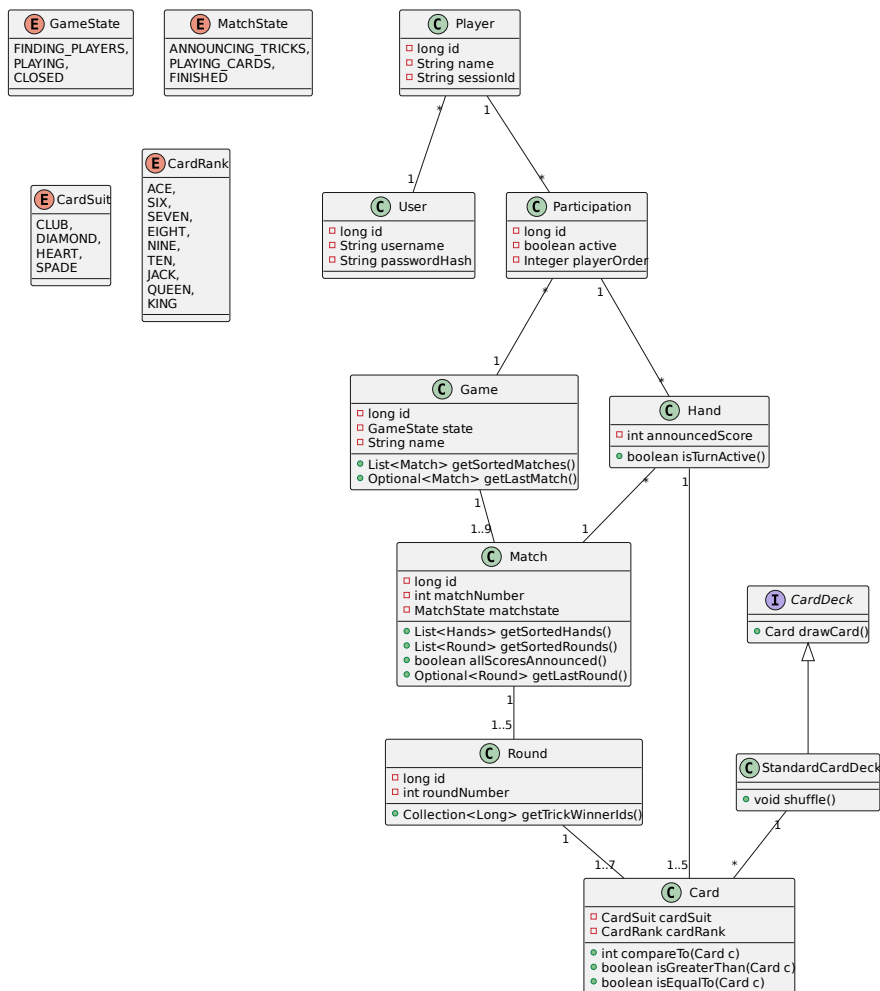Moris Camporesi (19-764-349)

Diagrams

Database Schema

Game
- id : DECIMAL «generated»
- name : VARCHAR
  gameState : INTEGER
  videoChatName : VARCHAR
  nextGame : DECIMAL «FK»

Player
- id : DECIMAL «generated»
- name : VARCHAR

Match
- id : DECIMAL «generated»
- matchNumber : INTEGER
- matchState: INTEGER
- gameId : DECIMAL «FK»

Participation
- id : DECIMAL «generated»
- active : BOOLEAN
  participationNumber : INTEGER
- playerId : DECIMAL «FK»
- gameId : DECIMAL «FK»

Round
- id : DECIMAL «generated»
- matchId : DECIMAL «FK»
- roundNumber : INTEGER

Hand
- id : DECIMAL «generated»
- announcedScore: INTEGER
- matchId : DECIMAL «FK»
- participation_id : DECIMAL «FK»

Card
- id : DECIMAL «generated»
- cardRank : INTEGER
- cardSuit : INTEGER
- handId : DECIMAL «FK»
  roundId : DECIMAL «FK»

Figure 2: db_schema

Class Diagram

The class diagram that we handed in for M2 was quite sparse what was pointed out in the feedback. It has grown significantly (and also changed a bit) meanwhile the development process and because we use it permanently as a basis for our common understanding, we decided to hand in the extended version again with the report for M3.

E GameState
FINDING_PLAYERS,
PLAYING,
CLOSED

E MatchState
ANNOUNCING_TRICKS,
PLAYING_CARDS,
FINISHED

C Player
- □ long id
- □ String name
- □ String sessionId

E CardRank
ACE,
SIX,
SEVEN,
EIGHT,
NINE,
TEN,
JACK,
QUEEN,
KING

E CardSuit
CLUB,
DIAMOND,
HEART,
SPADE

C User
- □ long id
- □ String username
- □ String passwordHash

C Participation
- □ long id
- □ boolean active
- □ Integer playerOrder

C Game
- □ long id
- □ GameState state
- □ String name
- ● List<Match> getSortedMatches()
- ● Optional<Match> getLastMatch()

C Hand
- □ int announcedScore
- ● boolean isTurnActive()

C Match
- □ long id
- □ int matchNumber
- □ MatchState matchstate
- ● List<Hands> getSortedHands()
- ● List<Round> getSortedRounds()
- ● boolean allScoresAnnounced()
- ● Optional<Round> getLastRound()

I CardDeck
- ● Card drawCard()

C Round
- □ long id
- □ int roundNumber
- ● Collection<Long> getTrickWinnerIds()

C StandardCardDeck
- ● void shuffle()

C Card
- □ CardSuit cardSuit
- □ CardRank cardRank
- ● int compareTo(Card c)
- ● boolean isGreaterThan(Card c)
- ● boolean isEqualTo(Card c)

UI Screenshots

Tests

Complex unittest

Vorschlag 1: the_first_player_must_play_a_card_when_round_starts in HandTurnActiveTest Vorschlag 2: get_multiple_trick_winner_test in RoundTest

Integrationtest

Vorschlag: play_last_card_new_round_new_match in CardEventHandlerTest

REST interface test

Vorschlag: change_gameState_to_playing in GameIntegrationTest