

---

## **Design Document for Truco Project**

---

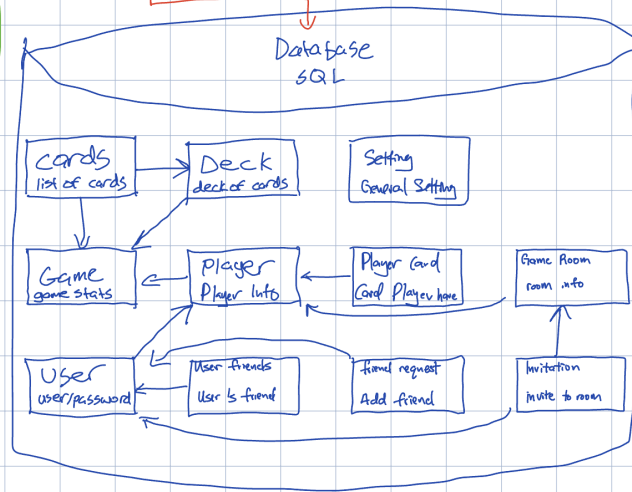
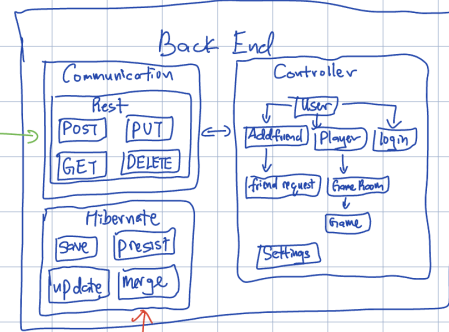
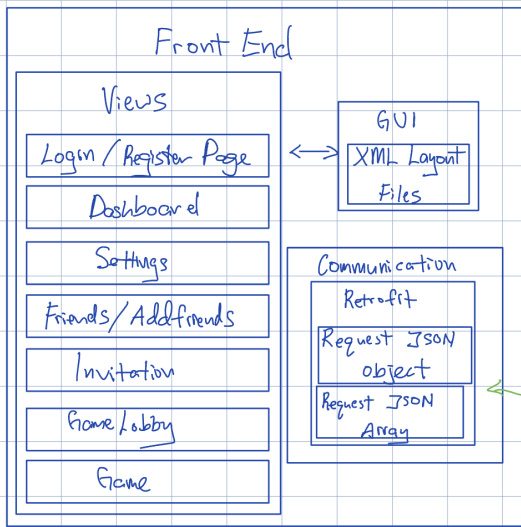
Group SD\_326

Member1: Edmund Lim

Member2 : Felipe Ely

Member3 : Dino Huric

Member4 : Adin Huric



- HTTP Connection
- JDBC Connection
- General Relationship

**Database:**

Our database holds tables Card, which store every card possibility and are associated with a deck and a game id. Each card is also given a value, which is basically how good the card is; this is used to determine which card wins after both are played. When a card is given to a player, it is removed from the deck, and they no longer have a deck ID but are given a player ID. When a card is played, the card won't have a deck ID or a player ID. Another table is the deck, which just stores the deck ID, and each game has its deck of cards. The friend request table holds the status of a friend request and the sender and receiver. The game table holds information such as round number, mode of the game, which is 1v1 or 2v2, state of the game, which player turn it is, winner ID, which is the ID of the player that won, deck ID, player one id, player two id, player 1 points, player 2 points, team 1 points, team 2 points. Eventually, I will add 2v2, so I must have four players. The game room table holds info such as if the host is ready and max players. A table for players which stores player info. A user table that stores username and password. A settings table that stores your settings, such as volume and language, and a user/friends table that stores people's friends.

**Backend:**

Our backend comprises a login, a gameroom, settings, the Truco card game logic, and friend requests. For the game logic, we currently have it, so it initializes a card game between two users that are in the database. It sets things up, such as whose turn it is, which is random, what cards the players are dealt which are from a deck, so there will be no repeated cards throughout the game, and the mode of the game, which would be 1v1 in this case. I have it set up so that in the database, each card has an attribute in the database, which is a boolean of whether it has been played or not. This is used to determine what card has been played and ensure that only 1 card is being played by each player. If a player plays a card, it becomes true. Once both cards are played, the cards can then be compared to determine who wins the round. Whoever has the better card is awarded points. Since each card is given a value when first initialized, comparing the two cards played makes it easy. We still need to implement things such as Truco being called. This would change the game's state and the amount of points awarded to the winning player of the round.

**Frontend:** The frontend comprises a lot of the backend code. We use all the model classes the backend makes and copy them for our logic. One of our more complex designs is the actual game room and its front-end designs. The backend sets the card from the deck and hands the cards out. So, the front end has to get that information from the backend and display the correct card number and suite. We connect to everything via a server and get all of our needed data from the database we have. We still have a lot of things to finish in the logic of the game and the game actually being played, like calling Truco, Envído, Flor, etc.

