ASSIGNMENT 2

UML

CSE2115 Software Engineering Methods
Team 90

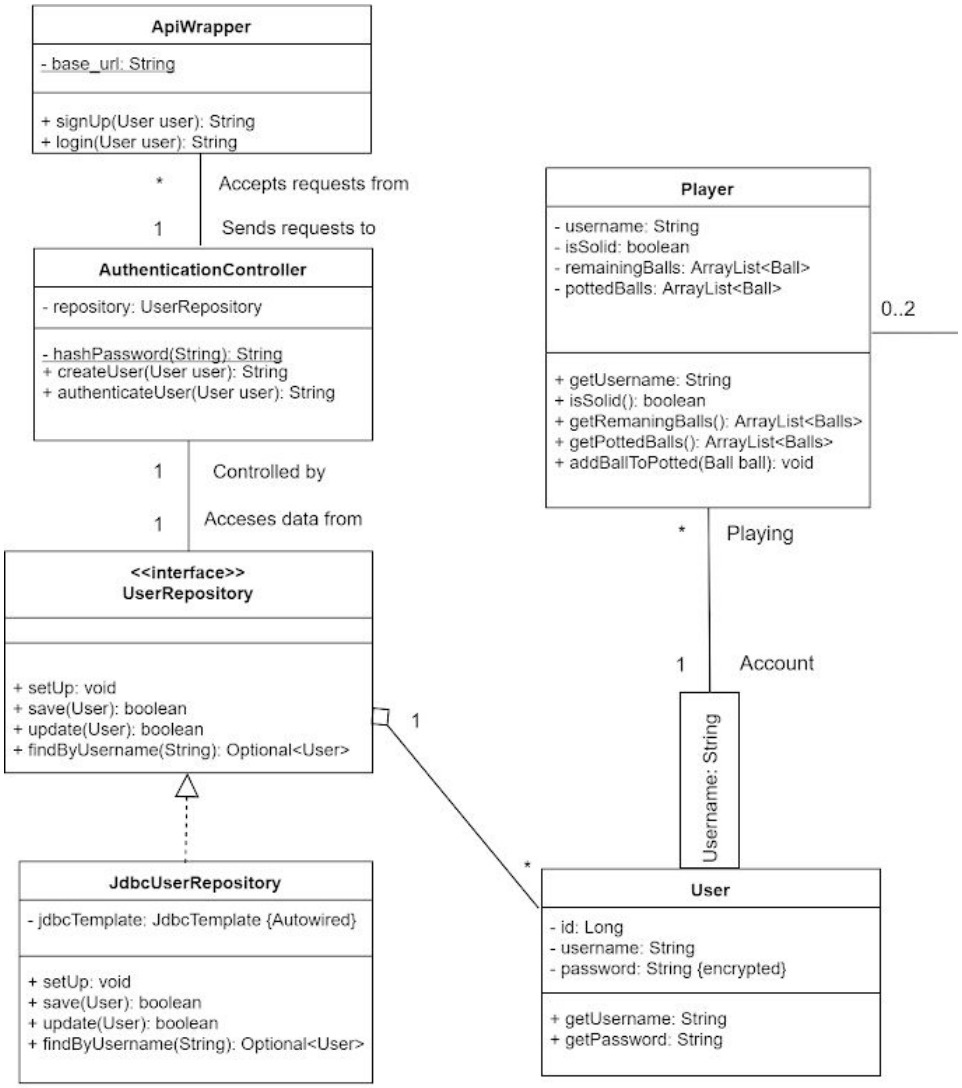
Made by:

Victor Roest
Tim Numan
Maikel Houbaer
Jesse Nieland
Eduard Filip

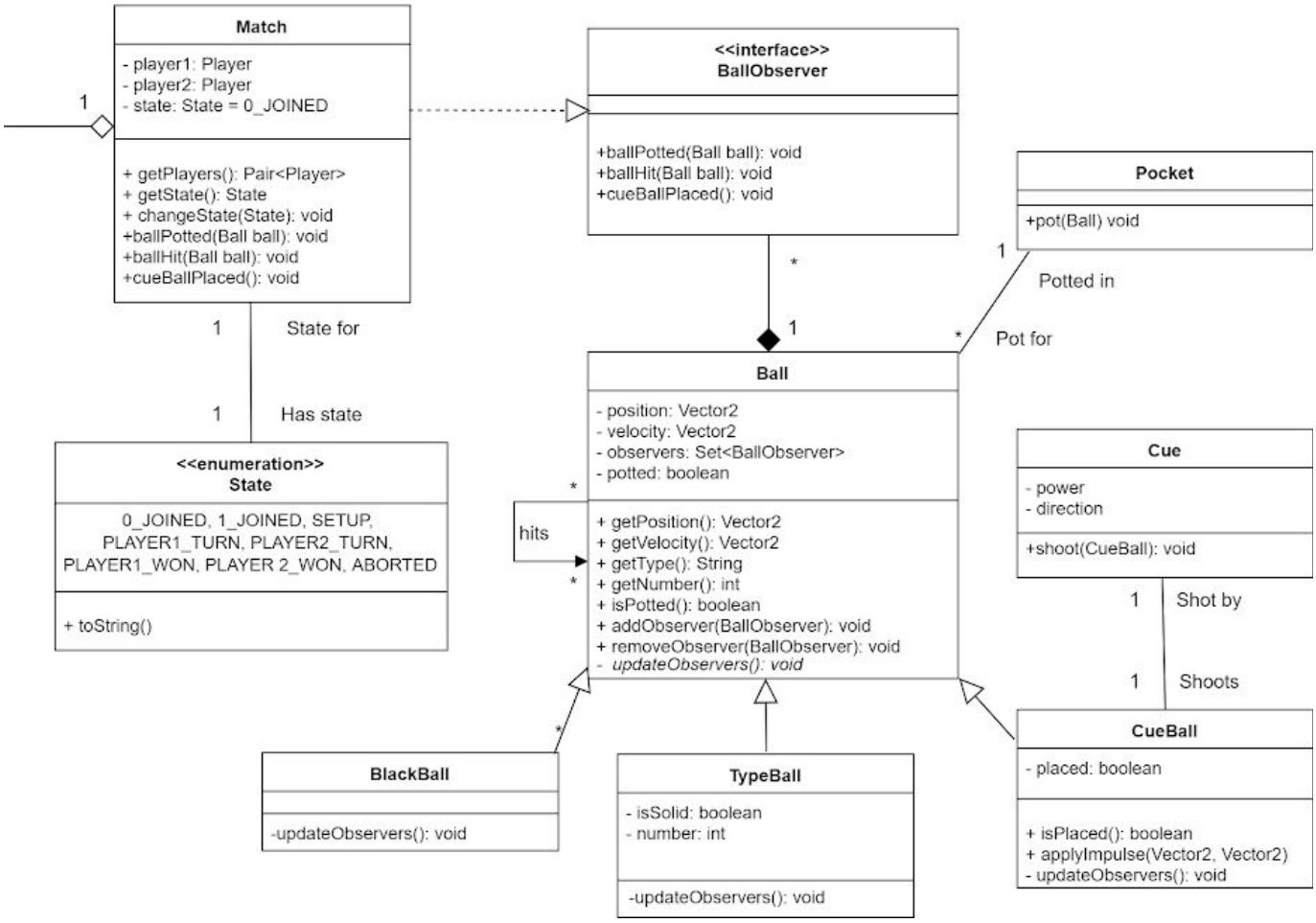
CONTENTS

Exercise 1 - Modelling class diagrams

Unfortunately the whole diagram was too large for one page, but you can 'see' the whole diagram when you place these 2 pages next to each other. You can also find the image here: https://imgur.com/a/Emi92KU



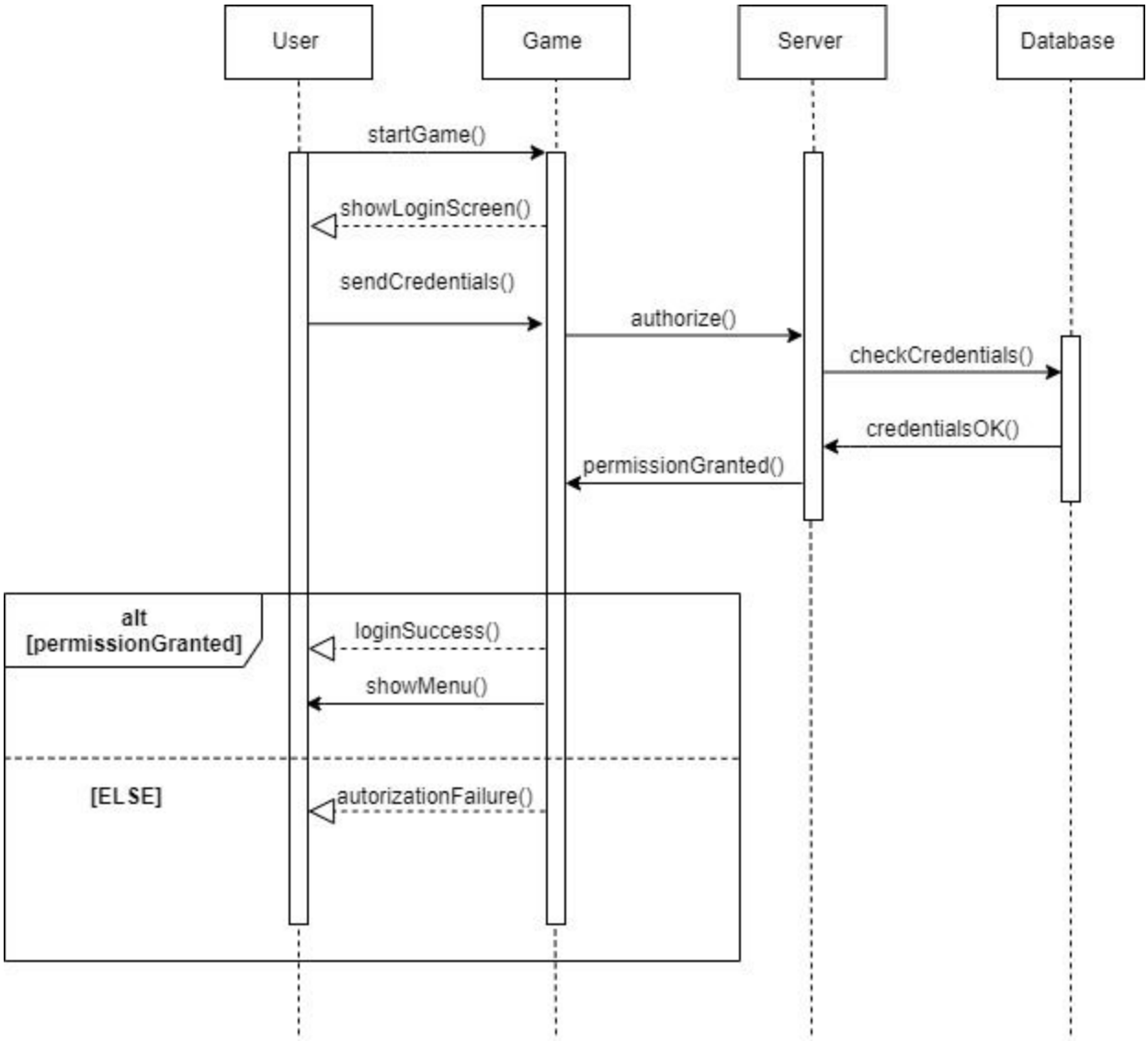
Authentication: Above you can see the classes needed for authentication. We have an ApiWrapper (client), which sends requests to the AuthenticationController (server). This controller contains a UserRepository to access user info from the database. UserRepository itself is an interface, implemented by the JdbcUserRepository. The data that can be accessed via this repository is of class User.



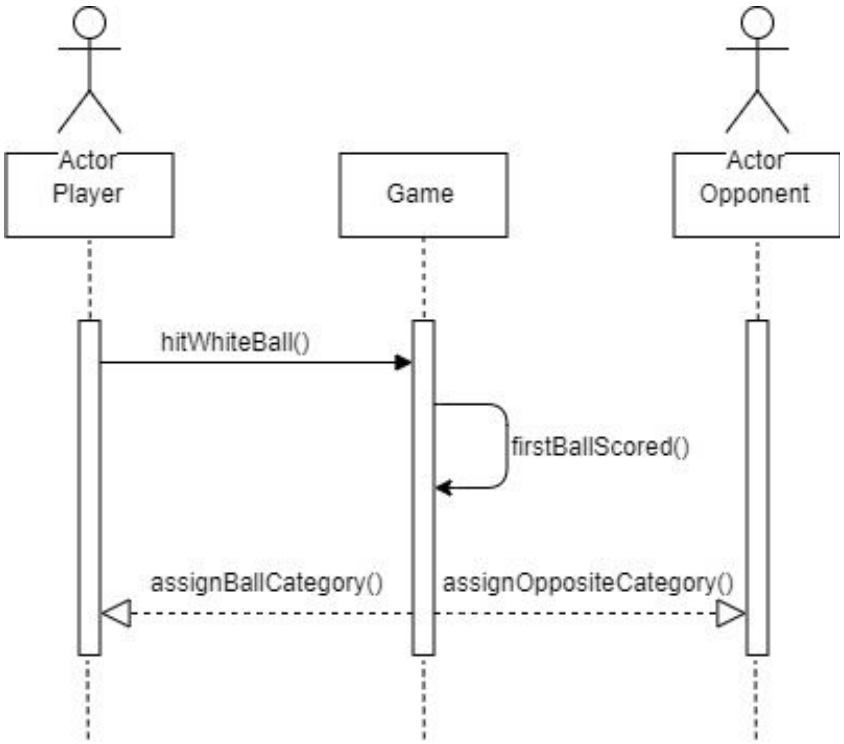
Core logic of game: The first class related to the core logic of the game is the Player class on the first image. Each time a user wants to play a match, a new Player object is made, which will store the username and all player-related game status. This player is then added to a Match object, which is the aggregation of at most 2 players + the generic game state. It implements the interface BallObserver, a nested interface of Ball (best way to visualize this was a composition), so it will be notified when a ball is potted/hit/ placed (for cue ball only). Ball is an abstract class, which will be extended by BlackBall, TypeBall, CueBall. Finally we have the Cue and Pocket classes, which are also related to the ball(s) class(es). Since we should not include GUI classes, the 'GUI parts' of classes as Ball, Pocket and Cue are also left out.

Exercise 2 - Modelling sequence diagrams

1. User Authentication



2. Potting the first ball



3. Processes after the end of the pool match