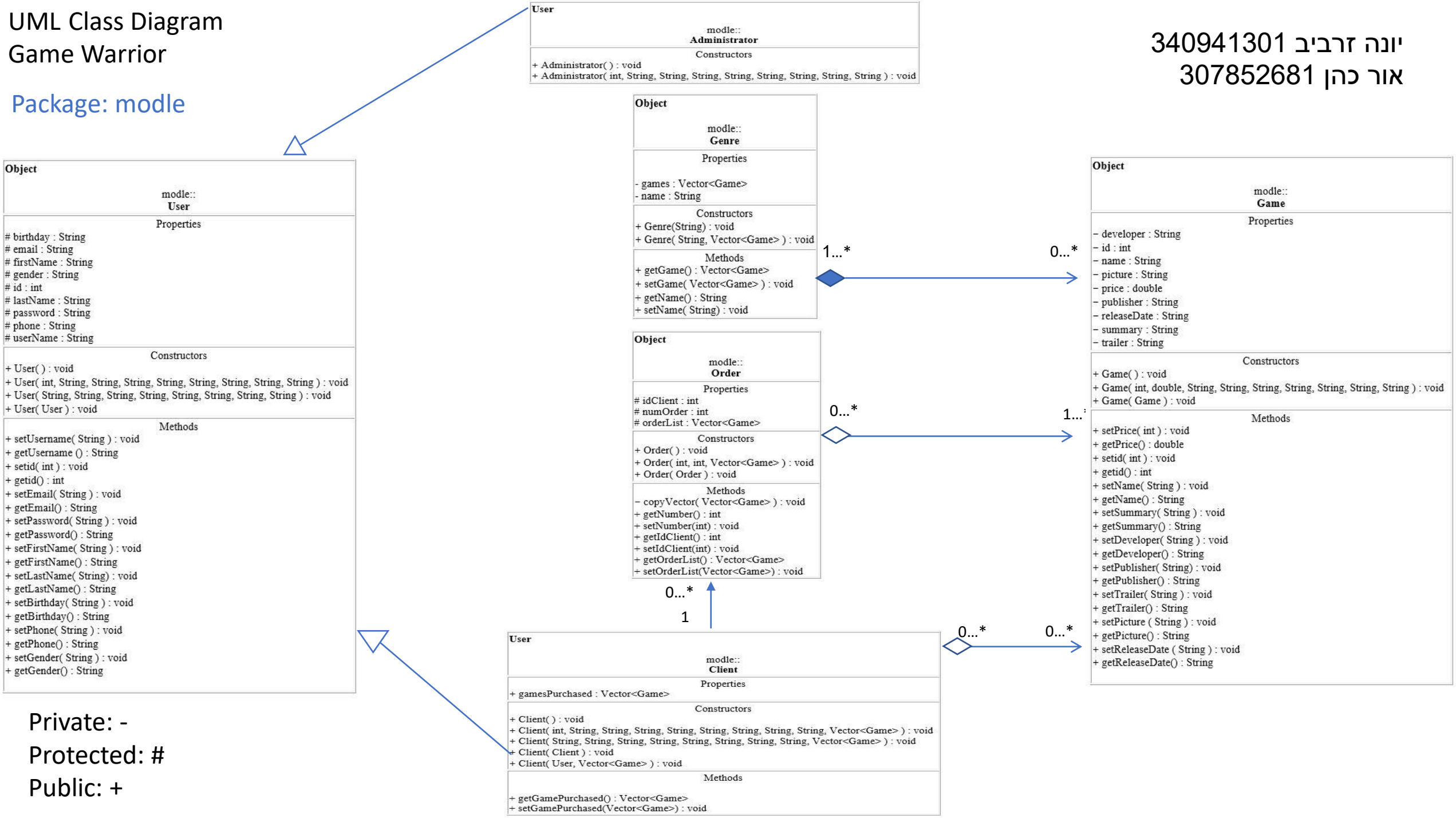
UML Class Diagram
Game Warrior

## Package: modle

יונה זרביב 340941301
אור כהן 307852681




Private: -
Protected: #
Public: +

# Package: controller

**Object**

controller::
**Control**

Constructors

+ Control( ) : void

Methods

+ addGame( String, double, String, String, String, String, String, String[] ) : boolean
+ addOrder( Vector<Game>, int ) : boolean
+ addOrderLines( Vector<Game>, int ) : boolean
+ addUser( String, String, String, String, String, String, String, String, String ) : int
+ averageRating( ) : HashMap<Integer, Double>
+ cosineCorrelation( HashMap<Integer, Double>, int ) : HashMap<Integer, Double>
+ createGenreGameVector( String ) : Vector<Game>
+ createHistoryVector( int ) : Vector<Order>
+ createOrderList( int ) : Vector<Game>
+ createOrders( ) : HashMap<Integer, Vector<Order>>
+ createVectorGames( ResultSet ) : Vector<Game>
+ deleteOrder( int ) : boolean
+ deleteUser( String[] ) : boolean
+ expectedRating( Vector<Integer>, HashMap<Integer, Double>, int, HashMap<Integer, Double> ) : HashMap<Integer, Double>
+ fiveNeighbors( HashMap<Integer, Double> ) : Vector<Integer>
+ fromRsToGame( ResultSet ) : Game
+ getClientUsers( ) : ArrayList<User>
+ getGame( int ) : Game
+ getRatingTable( ) : HashMap<Integer, HashMap<Integer, Double>>
+ getRelevantGames( ) : Vector<Game>
+ getUser( String ) : User
+ isAdmin( String ) : boolean
+ isExist( String ) : boolean
+ mostRecomeddedGames( HashMap<Integer, Double>, HashMap<Integer, Double>, int ) : Vector<Integer>
+ passwordMatch( String, String ) : boolean
+ rateGame( int, int, double ) : boolean
+ recommendedAlgo( int ) : Vector<Integer>
+ search( String ) : Vector<Game>
+ unrelevantGame( String[] ) : boolean
+ updateField( int, String, String, String, String, String, String, String, String ) : int
+ updateGame( int, String, double, String, String, String, String, String, String, String[] ) : boolean
+ userGames( int ) : Vector<Game>

# Package: database

**Object**

database::
**DatabaseFoundations**

Constructors

+ DatabaseFoundations( ) : void

Methods

+ createGamesDb( Connection ) : void
+ createGenreDb( Connection ) : void
+ createOrderLinesDb( Connection ) : void
+ createOrdersDb( Connection ) : void
+ createRatingDb( Connection ) : void
+ createTableGames( Connection ) : void
+ createTableGenre( Connection ) : void
+ createTableOrderLines( Connection ) : void
+ createTableOrders( Connection ) : void
+ createTableRating( Connection ) : void
+ createTableUsers( Connection ) : void
+ createUsersDb( Connection ) : void
+ main( String[] ) : void

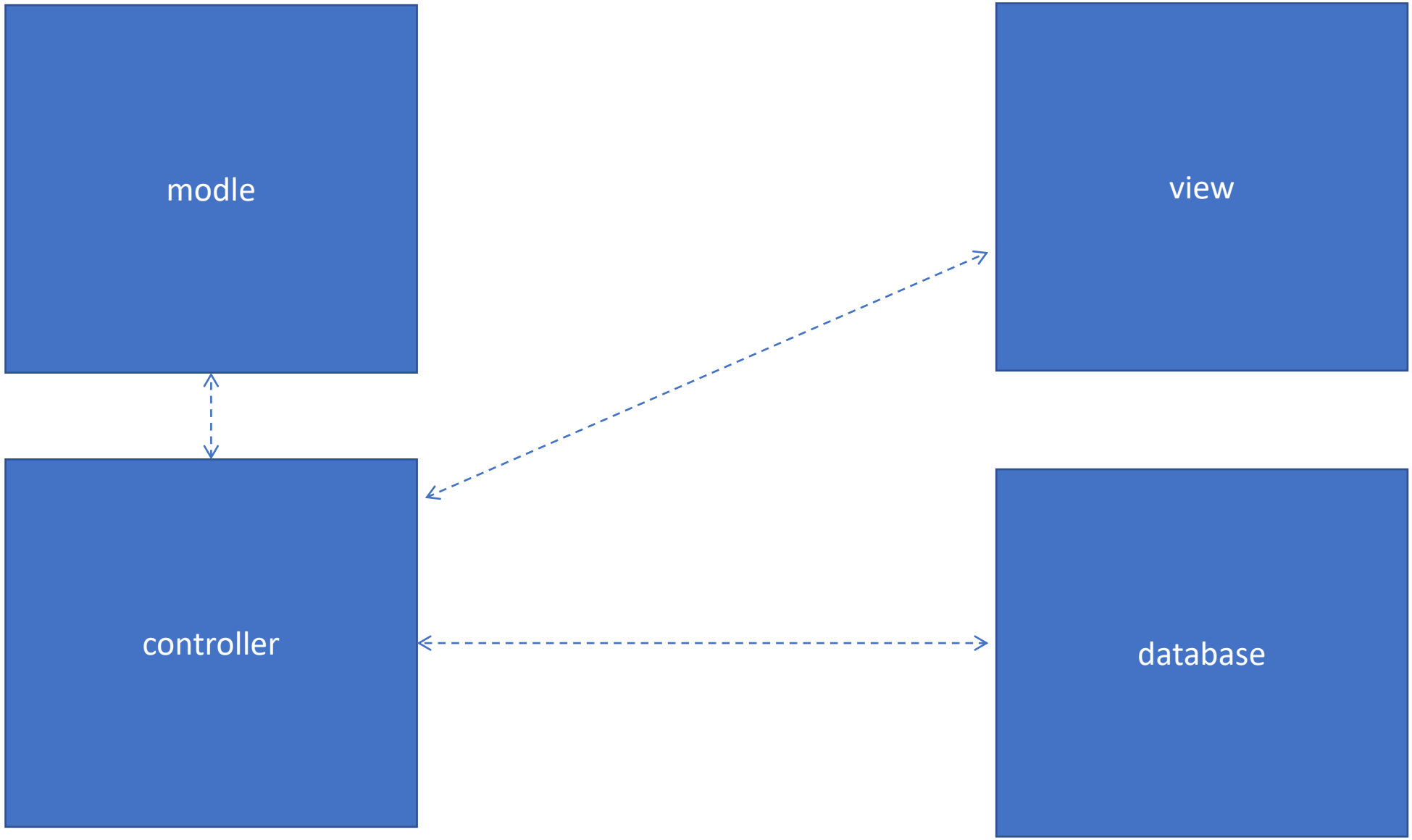
**Object**

database::
**Database**

Properties

- conn : Connection
- statement : Statement
- pStatement : PreparedStatement

Methods

+ isUserExist( String )
+ addUser( String, String, String, String, String, String, String, String )
+ updateField( int, String, String, String, String, String, String, String )
+ userGames(int)
+ getGame(int)
+ getRelevantGame()
+ getUser(String)
+ getSummary() : String
+ updateGame(int, double, String, String, String, String, String, String, String, String[]) : boolean
+ addGame(double, String, String, String, String, String, String, String, String[]) : boolean
+ addGameToGenre(String[], int) : boolean
+ getClients() : ResultSet
+ deleteUser(String) : boolean
+ deleteOrderByClientId(int) : void
+ unrelevantGame(int) : boolean
+ deleteGameFromGenre(int) : boolean
+ getGamesByGenre(String) : ResultSet
+ getGamesById(ResultSet) : ResultSet
+ getClientOrders(int) : ResulSet
+ getOrderLines(int) : ResultSet
+ addOrder(int) : ResultSet
+ deleteOrder(int) : boolean
+ addOrderLines(int, int) : boolean
+ deleteOrderLines(int, int) : boolean
+ deleteOrderLines(int) : boolean
+ getLastNumOrder() : ResulSet
+ lockTables() : void
+ unlockTables() : void
+ addRatingRowsForNewUser(int) : boolean
+ addRating(int, int, double) : boolean
+ getGameId(double, String, String, String, String, String, String, String) : int
+ deleteGame(int) : ResultSet
+ deleteRating(int, int) : boolean
+ deleteRatingByClientId(int) : boolean
+ updateRating(int, int, double) : boolean
+ getUserRating(int) : ResultSet

Connections between packages