The Game entity contains the information pertaining ShipLocation is the entity responsible for storing to gameplay. each player's ship locations. Game **ShipLocation** + MIN_SHIP_NUMBER: int {readOnly} - id: long {readOnly} - key: UUID {readOnly} + MIN X COORD: int {readOnly} - created: Instant {readOnly} + MIN_Y_COORD: int {readOnly} - userGames: List<UserGame> {readOnly} - id: long {readOnly} - userGame: UserGame {readOnly} - boardSizeX: int - boardSizeY: int - shipNumber: int - shipCoordX: int - playerCount: int {readOnly} - shipCoordY: int generateKey(): UUID UserGame is the central information hub. Any information it does not have, it gets Game is where the business by querying the other entities. The user will place all of their of playing the game happens. ships before they can join When the user opens the app, **UserGame** a game. UserGame owns the a game object is created and ShipLocation instances and then the game initializes - id: long {readOnly} updates them as the game the user's UserGame instance. progresses. - key: UUID - user: User {readOnly} - game: Game {readOnly} - locations: List<ShipLocation> - fromShots: List<Shot> UserGame is the liason between The user fires/creates - toShots: List<Shot> the user and the rest of the game. 2 a volley of shots. The All information coming to User shots are reported to goes through UserGame first. + hashCode(): int UserGame to be processed. + equals(Object obj): boolean generateKey(): UUID User - id: long {readOnly} - key: UUID {readOnly} Shot - created: Instant {readOnly} - id: long {readOnly} modified: Instant {readOnly} - fromUser: UserGame {readOnly} - displayName: String - toUser: UserGame - oauthKey: String - shotCoordX: int - userGame: List<UserGame> - shotCoordY: int - timestamp: Instant {readOnly} + hashCode(): int + equals(Object obj): boolean + isHit(): boolean generateKey(): UUID The User fires shots at other users' ships.

The User entity represents the player.