COMP 3004 - Deliverable #3
System Architecture and Design
Brackit - Mobile Tournament Bracket Creation

Metadata

Team / App Name: Brackit

Team member names

Name	Student ID
Jaime Herzog	101009321
Suohong Liu	101002340
Xiyi Liu	101004577
Alex Trostanovsky	100984702

Contents

Architecture

identify, describe, and justify the architecture of your project (architectural style, design patterns)
Outcome is a system architecture that supports the functional goals and non-functional attributes of your project
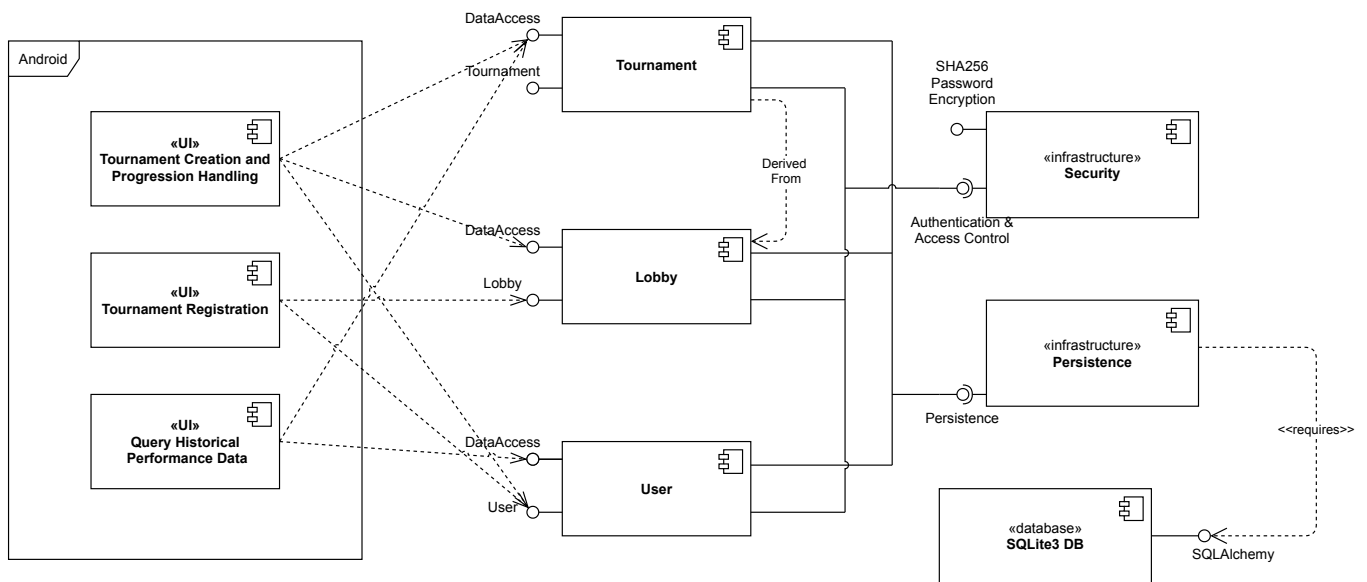
1 Description

In developing `Brackit`, we set out to address an urgent need by tournament attendants and organizers to visualize, manage, and interact with double elimination brackets on their mobile device. We committed to developing a product that will meet the following functional requirements:

1. Tournament Organizers can create, host, and maintain double elimination brackets.

2. Users of `Brackit` can use the application to join create tournaments.

2 Justification

3 Architectural Diagrams

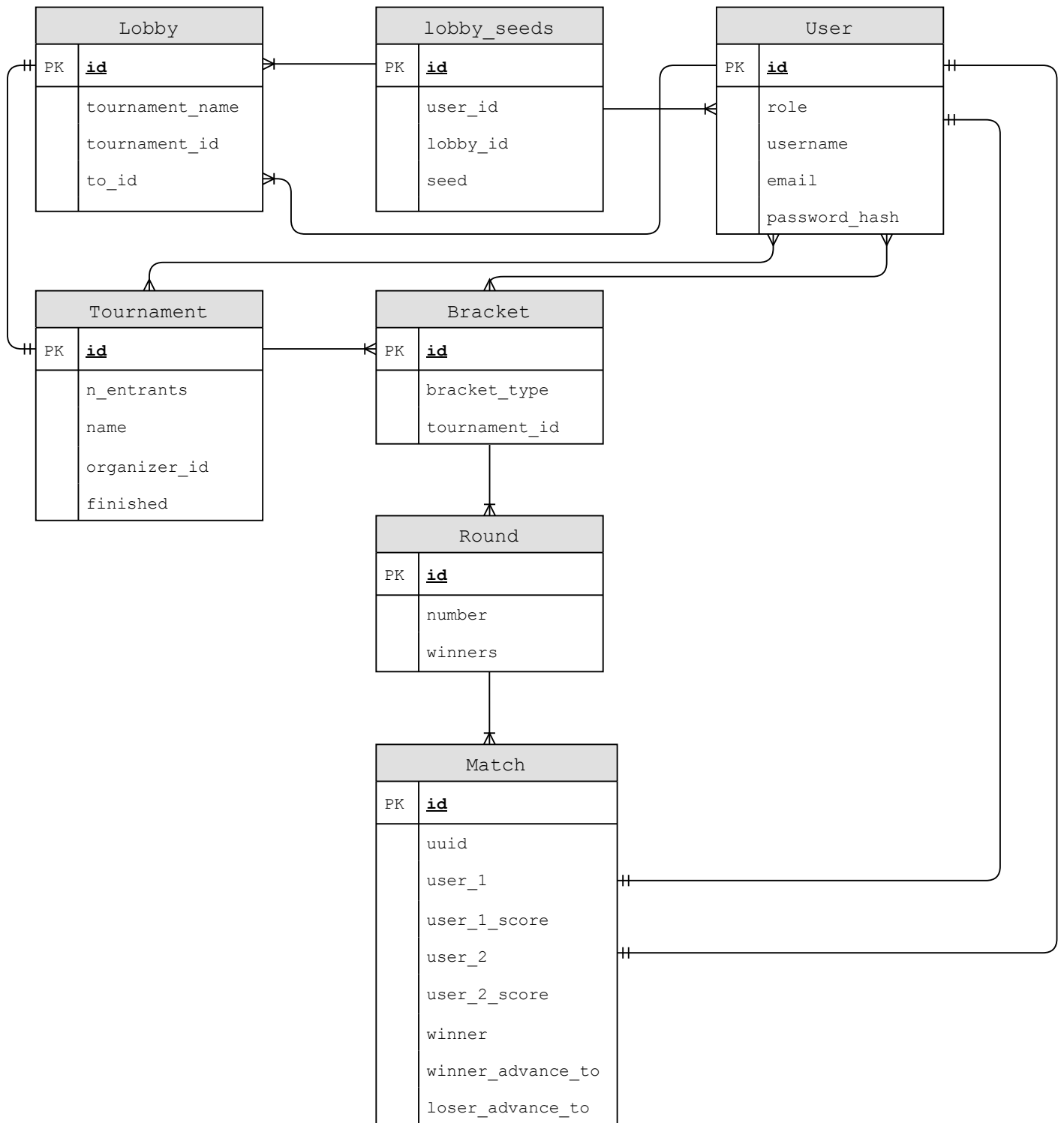Figure 1: `Brackit` - UML 2 Architectural Component Diagram

Figure 2: `Brackit` - Entity Relationship (ER) Diagram

Design

1 Description and Rationalization

- Use clear description of the structure of the components and its externally visible interfaces

- Clarify the physical location of where the classes will reside (e.g., on the client, on a server), as well as any external API

- Include references to your system's architecture (patterns, abstractions, data structures/ algorithms)

- An analysis of how your design minimizes coupling and accommodates changing requirements
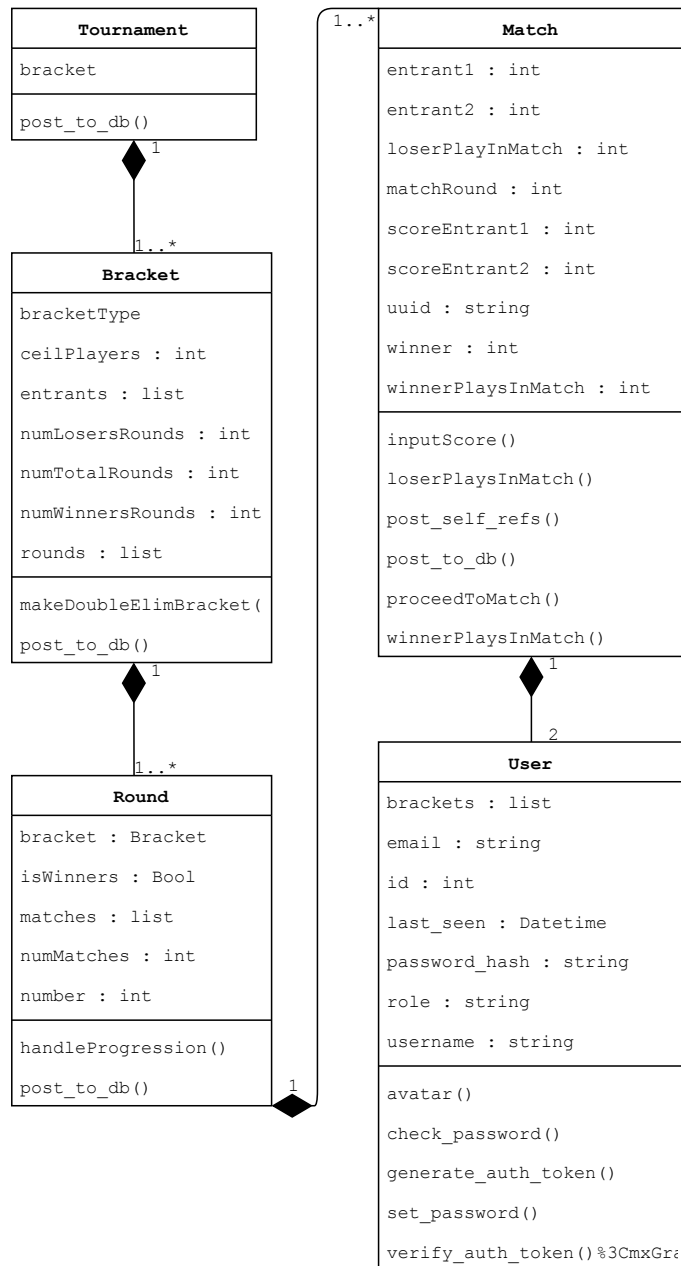
Design Diagrams

Tournament

bracket

post_to_db()

1

1..*

Bracket

bracketType

ceilPlayers : int

entrants : list

numLosersRounds : int

numTotalRounds : int

numWinnersRounds : int

rounds : list

makeDoubleElimBracket(

post_to_db()

1

1..*

Round

bracket : Bracket

isWinners : Bool

matches : list

numMatches : int

number : int

handleProgression()

post_to_db()

1..*

Match

entrant1 : int

entrant2 : int

loserPlayInMatch : int

matchRound : int

scoreEntrant1 : int

scoreEntrant2 : int

uuid : string

winner : int

winnerPlaysInMatch : int

inputScore()

loserPlaysInMatch()

post_self_refs()

post_to_db()

proceedToMatch()

winnerPlaysInMatch()

1

2

User

brackets : list

email : string

id : int

last_seen : Datetime

password_hash : string

role : string

username : string

avatar()

check_password()

generate_auth_token()

set_password()

verify_auth_token()%3CmxGr

1

Figure 3: `Brackit` - UML Class Diagram