

COMP 3004 - Deliverable #1 - Project Proposal

Brackit - Mobile Tournament Bracket Creation

Jaime Herzog, Suohong Liu, Xiyi Liu, Alex Trostanovsky

jaime.hertzog@carleton.ca, suohong.liu@carleton.ca, xiyi.liu@carleton.ca, alex.trostanovsky@carleton.ca

Metadata

Team / App Name: Brackit

Team Member Names:

Jaime Herzog: 101009321, Suohong Liu: 101002340, Xiyi Liu: 101004577, Alex Trostanovsky: 100984702

What is our project?

- **Brackit** is a tournament management and attendee platform for mobile.
- Primary focus: tournament organizers and the competitors that attend tournaments.
- **Brackit** is agnostic to the type of competition that is taking place.
- Tournament organizers can seed (rank competitors) and manage their double elimination tournament brackets on their mobile devices.
- Competitors can create their own user profiles to track their tournament history and stats.

Why is it interesting?

- Current tournament management services (e.g. **challonge**) do not offer mobile platforms, and navigating these platforms on mobile is unwieldy.
- **challonge** does not record comprehensive tournament histories:
 - tournament organizers enter competitors as a string corresponding to their team name or tag.
 - as a result, competitors can't enter brackets using a profile that tracks their history and statistics.
- **Brackit** allows competitors to evaluate their performance across tournaments in a new way by gaining access to their track record against past opponents.

As experienced tournament organizers, we've experienced firsthand the frustration stemming from the fact that competitors have limited access to information during a tournament. **Brackit** proposes to fix these problems by enabling players to interact with the tournament bracket on their phones.

As competitors, we've spent countless hours tabulating historical data from tournaments to get an idea of past results. **Brackit** solves this problem by centralizing tournaments' data collection on one platform.

Why does this project make sense in a mobile form factor?

Traditionally the only person with information about an ongoing tournament is the tournament organizer, who usually has a laptop or a tablet that every attendee crowds around to see who they have to play against.

Brackit allows players full access to the bracket in the palm of their hands.

User Types (UTs)

1. *Tournament organizers (TOs):*

Create tournaments; Seed (rank) entrants once entrants have registered in bracket; Allow entrants to register in a bracket using their **brackit** account; Register Wins/Losses/Scores of competitors using **brackit**; Monitor historical records (tournaments created/past results);

2. *Tournament competitors:*
Submit requests to participate in a TO's competition; Once accepted into a competition, can view the competition using **Brackit**; Monitor historical records (Wins/Losses/Scores);
3. *Spectators* (Registered **Brackit** users that are neither UT 1. or UT 2.):
Submit requests to view a TO's competition; Can view the competition using **Brackit**.
4. *Guest Users* (Unregistered **Brackit** users that are neither UT 1. or UT 2.)
Submit requests to view a TO's competition; Can view the competition using **Brackit**.

Functional Requirements (FR)

1. *Bracket Generation/Maintenance*

- 1.1. TOs can create joinable double elimination brackets
- 1.2. Entrants can use system to enter joinable double elimination brackets
- 1.3. Given a seeded list of entrants, output a correct double elimination bracket
- 1.4. TOs can access a setup screen where they seed (rank) the entrants dynamically and can view a preview of the bracket as seeded, then confirm the final active bracket
 - 1.4.1. If output of system is not accepted, TOs can reseed entrants

2. *Profile Creation/Maintenance*

- 2.1. Competitors/Organizers can create profiles which keep a history of:
 - 2.1.1. tournaments entered and created, placement history
 - 2.1.2. number of matches won/lost
 - 2.1.3. overall matchups against opponents
 - 2.1.4. additional user profile information
- 2.2. The system supports both user profiles and competitors who haven't created a profile (Guest Users)

3. *Data Input/Processing*

- 3.1. Tournament organizers can enter tournament results in active brackets
- 3.2. Use dynamic tournament results to render Losers brackets and subsequent Winners rounds during competitions in real time

4. *Bracket Visualization*

- 4.1. Users can access multiple views of brackets:
 - 4.1.1. Winners / Losers brackets
 - 4.1.2. Specific bracket rounds
- 4.2. Users can click on competitors in rounds to view their profile

Non-Functional Requirements (NFR)

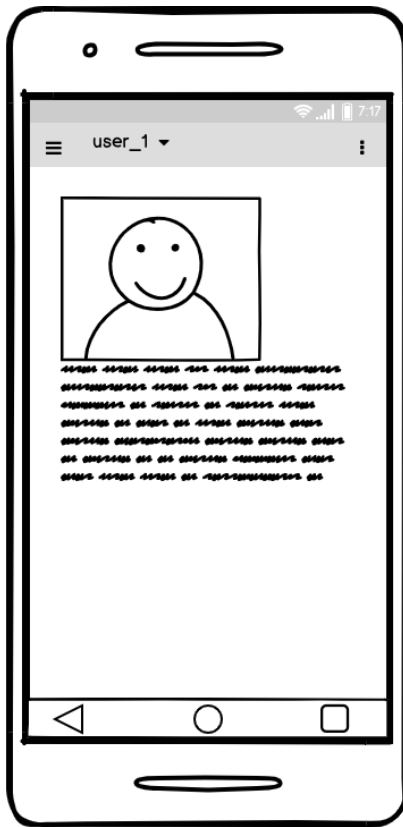
1. *Useability:* **Brackit's** main selling point - current platforms are simply not usable on mobile devices.
 - 1.1. Users can clearly view and understand bracket structure and competitors' placement within brackets
 - 1.2. Visually coherent mobile representation of a double elimination bracket
 - 1.3. Seamless competitor entry (by TOs) to brackets in the setup phase
2. *Screen Adaption*
 - 2.1. Users can visualize brackets in portrait and landscape modes
3. *Methodology*

IDE and Version Control Android Studio, Visual Studio Code, GitHub	Languages/Frameworks React-Native, Python
Platform Android	Development Process Agile

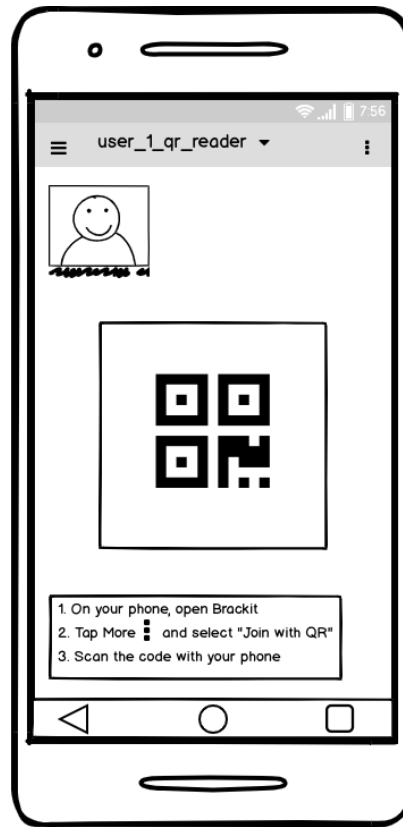
Detailed User Scenarios

1. TOURNAMENT ORGANIZER IS HOSTING A TOURNAMENT AND WANTS TO ENTER 14 PEOPLE INTO A BRACKET
 - 1.1. TO presses **Create bracket** Button (FR 1.1.)
 - 1.2. TO specifies number of competitors
 - 1.3. 14 Entrants join newly created bracket (See User Scenario 3) (FR 1.2.)
 - 1.4. TO seeds (ranks) entrants
 - 1.5. TO submits seeded list of entrants
 - 1.6. Valid seeded double-elimination bracket produced and returned for preview (FR 1.3.)
 - 1.7. TO accepts bracket preview, and creates active bracket (FR 1.4.)
2. A USER WANTS TO CREATE A USER PROFILE (FR 2.1.)
 - 2.1. User presses **Create profile** button
 - 2.2. User provides Username, password, email address, and additional information
 - 2.3. User submits profile
 - 2.4. Profile attributes are validated
 - 2.5. User profile is created
3. A COMPETITOR ENTERS A SETUP-PHASE BRACKET (FR 1.2.)
 - 3.1. QR Code associated with setup-phase bracket generated by system (Unique QR Code per bracket)
 - 3.2. TO presses **Display QR code**
 - 3.3. QR Code associated with setup-phase bracket retrieved by system
 - 3.4. TO displays QR code associated with newly created bracket
 - 3.5. Competitor presses **Join bracket**
 - 3.6. Competitor scans TO's QR code
 - 3.7. QR Code validated against system
 - 3.8. Competitor is entered into setup-phase bracket
4. A SPECTATOR WANTS TO VIEW AN ACTIVE BRACKET (FR 4.1.)
 - 4.1. QR Code associated with active bracket generated by system (Unique QR Code per bracket)
 - 4.2. TO presses **Display QR code**
 - 4.3. QR Code associated with active bracket retrieved by system
 - 4.4. TO displays QR code associated with active bracket
 - 4.5. Spectator presses **View bracket**
 - 4.6. QR Code validated against system
 - 4.7. Spectator is able to view active bracket

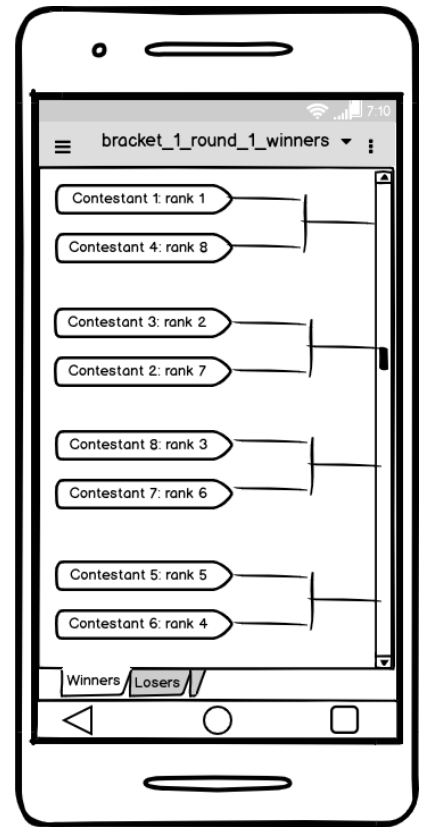
Mockups



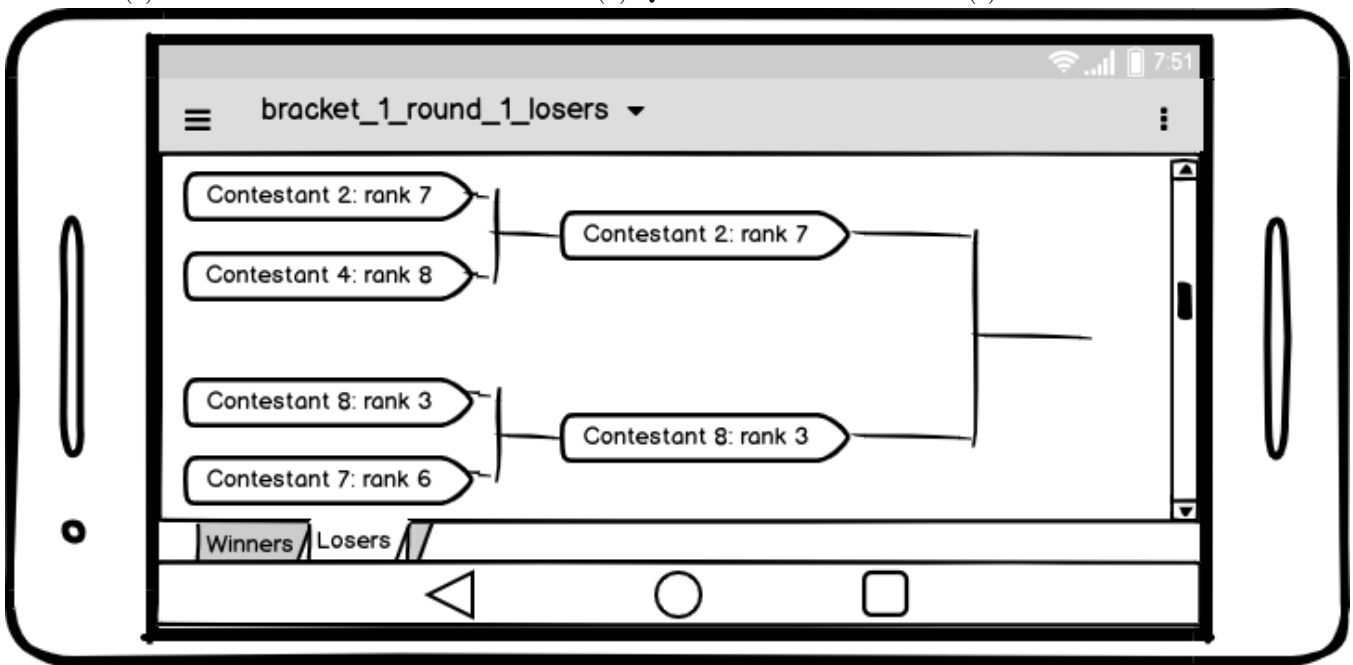
(a) User Profile



(b) QR Reader



(c) Vertical Bracket Visualization



(d) Horizontal Bracket Visualization