

UNIVERSITY OF WATERLOO

SE464

SECTION 001

Deliverable 3: Project Demonstrations

Group 25

November 21, 2016

1 Metadata

Project: ShuttleQL (Shuttle Queueing Logistics)

Team Name: Baddie Boys

Team Members:

- Cheng Dong (c9dong)
- Zhaotian Fang (z23fang)
- Clement Hoang (c8hoang)
- Di Sen Lu (dslu)

2 Demo Summary

The demo of ShuttleQL will demonstrate all of the functional and non-functional properties listed below and show a general workflow of the application.

2.1 Functional properties

1. Registering new club members
2. Allowing players to check-in/check-out of a club session
3. Match making algorithm that allocates players to open badminton courts
4. Display current players that should be on each court
5. Allow executives to override the match making algorithm
6. (Bonus) Allow executives to broadcast announcements to players

2.2 Non-functional properties

1. The user-facing interface should satisfy all functional properties on Chrome and Safari
2. (Bonus) The system should be adaptable to clubs with different number of courts and different game types for each court.
3. (Bonus) The match-making algorithm should take less than 5 seconds to complete given a pool of 100 badminton players.

There are two types of users using ShuttleQL: Players and Club Admins. The following detailed functionalities of the application for each type of user will be demonstrated.

2.3 Player functionalities

1. View personal profile information
2. View status
3. View announcements
4. View current matches

2.4 Admin functionalities

1. Registering, editing and removing players
2. Create and end club sessions
3. Check-in and check-out players of club sessions
4. Start and end match-making algorithm
5. View current players that should be on each court
6. Manually override the match making algorithm
7. Broadcast announcements to players

The following workflow shows how users will interact with ShuttleQL in a typical badminton club session.

2.5 Workflow

1. Both players and admin logins to their respective accounts
2. Admin create a club session and start checking in players
3. Admin starts match-making task to generate matches
4. Both players and admin view matches
5. Matches will be automatically generated every 15 minutes
6. Admin can manually override any matches they see fit
7. Admin can broadcast any messages to all players
8. At the end, admin ends the club session

3 Status Report

All of the required functional properties were implemented. Even one of the bonus functional properties were implemented which was real-time announcements broadcasting. The only non-functional property that was not satisfied was dependability. Our non-functional property required that the system must have an uptime of 95% or better for every 24 hours. The application, however, isn't deployed and currently only runs locally. The non-functional property was dropped because it costed money monthly to maintain the servers. We didn't account that deployment platforms such as Heroku didn't support sockets well under their free service. We need to pay in order to upgrade to support socket features. Therefore, we didn't want to pay this fee until the UW badminton club started using ShuttleQL. The next steps of ShuttleQL is to demo to the UW badminton club.

ShuttleQL

SE464 Project Demonstrations

Clement Hoang David Dong Jason Fang Tony Lu

University of Waterloo

Fall 2016

Overview

Problem

- ▶ Hundreds of student playing at UW Badminton club
- ▶ Bad club management system
 - ▶ crowded line ups between games
 - ▶ manual error prone matchmaking
 - ▶ inefficient communication between execs and members

Solution

ShuttleQL

- ▶ Mobile web optimized player dashboard
- ▶ Web based admin management dashboard

Functional Requirements

Player Dashboard

- ▶ match checker
- ▶ announcement
- ▶ player profile

Admin Management Dashboard

- ▶ remote session management
- ▶ player registration
- ▶ player checkin / checkout
- ▶ user management
- ▶ announcement broadcast
- ▶ match overrides

Non-Functional Requirements

Portability

- ▶ Player dashboard must work on both mobile chrome and safari browser
- ▶ Admin management dashboard must work on web chrome and safari browser

Efficiency

- ▶ Linear time matchmaking algorithm
- ▶ Able to compute matches for all club members in less than 1 second

Evolvability

- ▶ Court numbers can be changed through a config file
- ▶ Court can be specified as Single or Doubles game type

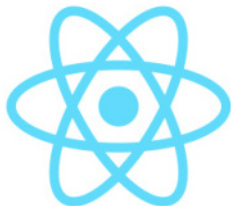
Unfinished Requirements

Dependability

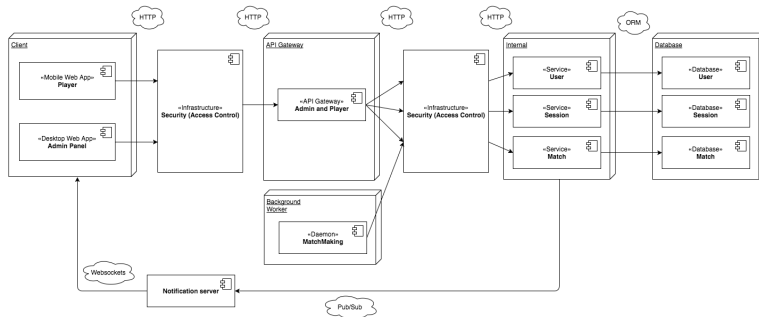
- ▶ Micro services are difficult to coordinate
- ▶ Deploying is expensive due to so many services

Architecture

Technologies



Architecture Diagram



Challenges

Challenges

- ▶ learning new technologies (scala, react)
- ▶ managing tasks
- ▶ micro services are difficult to deploy
- ▶ lack of time