TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Language Tandem Finder

Final presentation: Application Development for Mobile and Ubiquitous Computing

Salohy Miarisoa, Ljupka Titizova
Dresden, January 2017

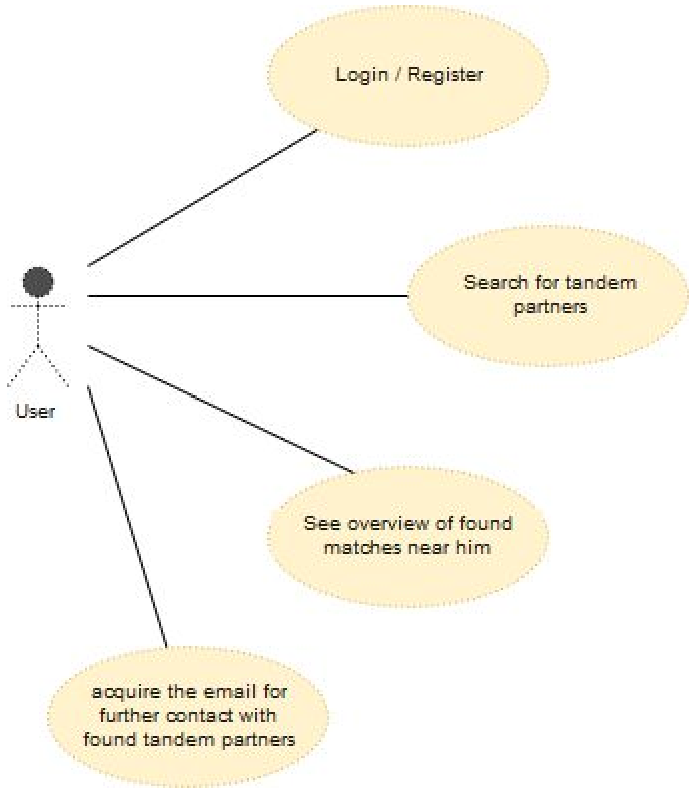
AGENDA

- Our Application
- Application Scenario
    - Use Cases and Mockups
- Architecture
- Used Technologies
- Tackled Challenges
    - Context and Adaptations
- Lessons learned and Pitfalls

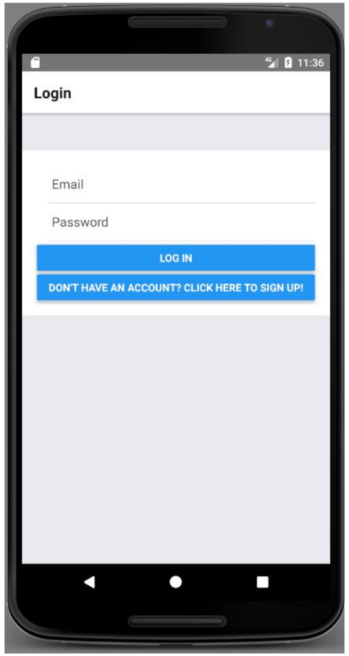
The Application

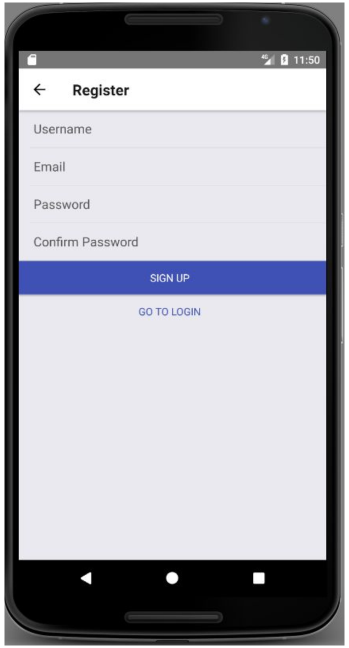
- Goal: Intermediation of Language Tandems between Students
- User:
  - e-mail address
  - offered language
  - language the user wants to study
- Matches nearby
- Overview of offered languages near the user

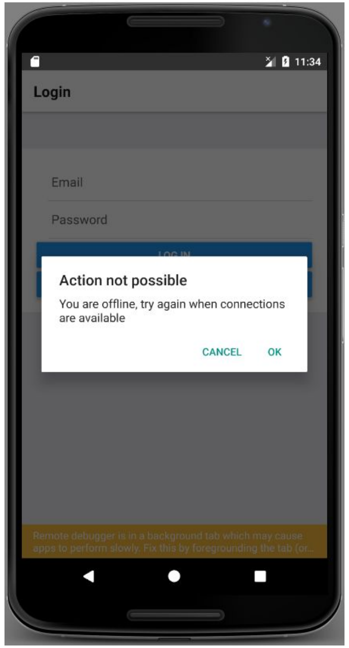
Use Cases


Login / Register
Search for tandem partners
See overview of found matches near him
acquire the email for further contact with found tandem partners
User

Mockups





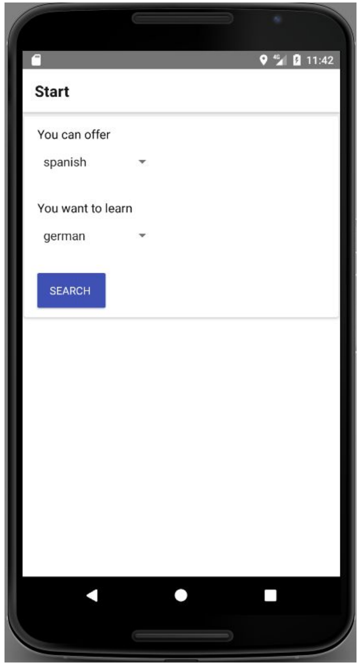


Action not possible

You are offline, try again when connections are available

Remote debugger is in a background tab which may cause apps to perform slowly. Fix this by foregrounding the tab (or...

Mockups


11:42

Start

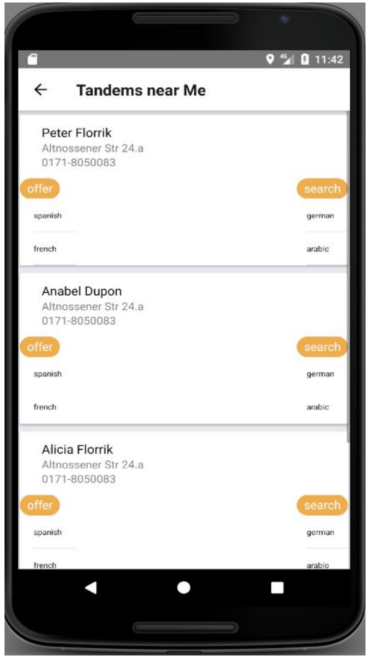
You can offer

spanish

You want to learn

german

SEARCH


11:42

Tandems near Me

Peter Florrik
Altnossener Str 24.a
0171-8050083

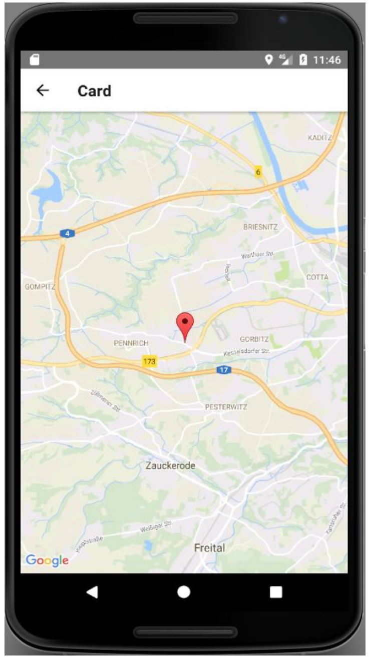
offer	search
spanish	german
french	arabic

Anabel Dupon
Altnossener Str 24.a
0171-8050083

offer	search
spanish	german
french	arabic

Alicia Florrik
Altnossener Str 24.a
0171-8050083

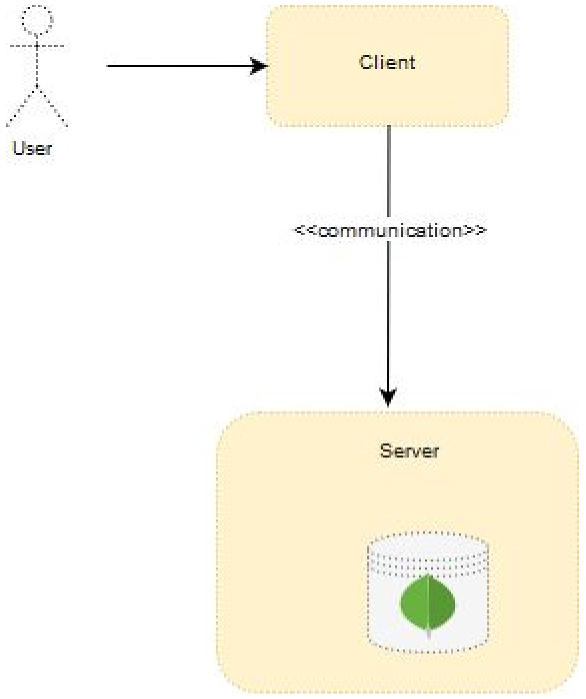
offer	search
spanish	german
french	arabic


11:46

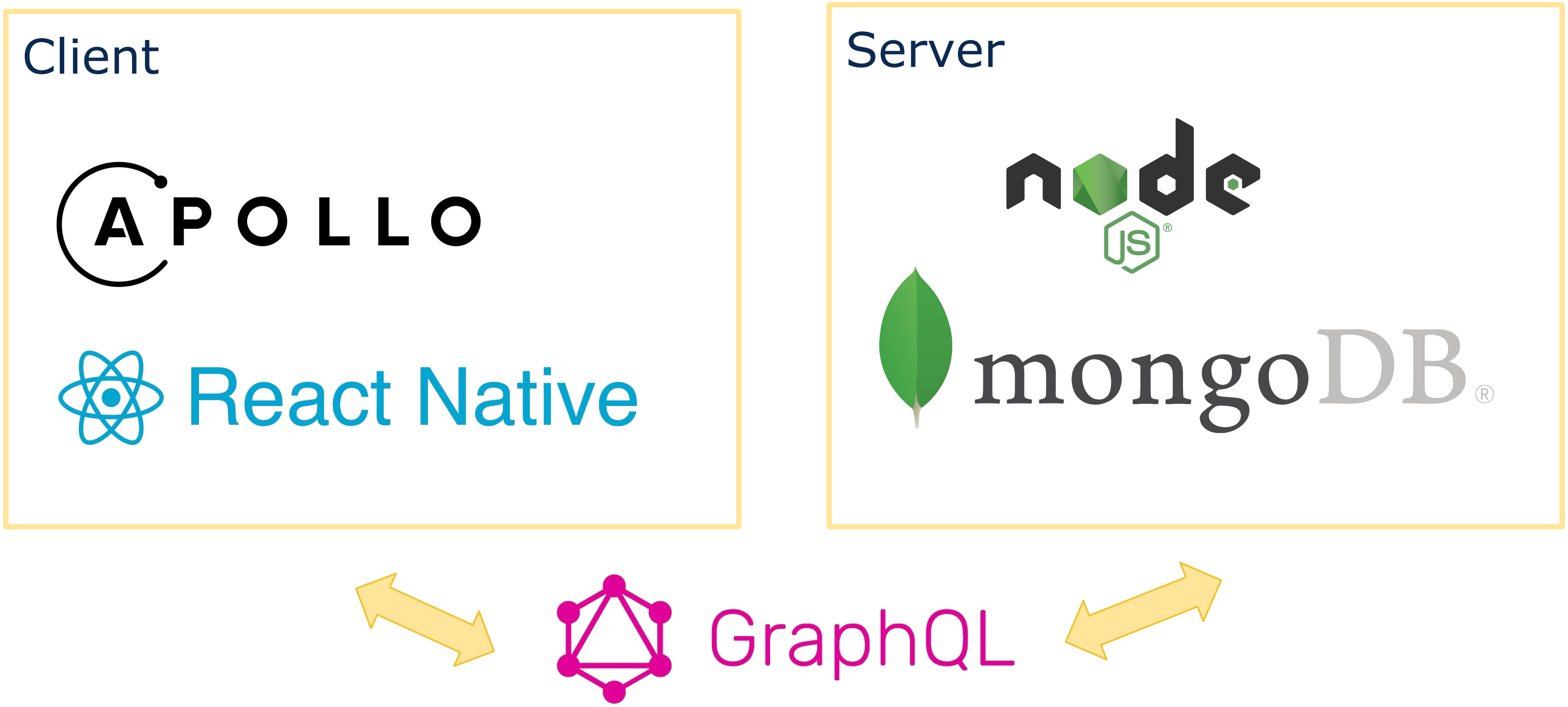
Card

Architecture

- Demand-Driven Architecture
- Client
- Server
- Database



Technologies



Tackled Challenges

Connectivity Challenge

Context: Network type, location, nearby persons

Adaptation:

- If connection is good (eg: 4G) -> Display map
- Otherwise -> Display list of nearby Tandem matches
- Client-side: detection of location of user
  - Context Source: GPS ((latitude, longitude))

```
navigator.geolocation.getCurrentPosition()
```

- Server-side: search entries near the client position

Connectivity Challenge: Map with Tandem Partners

- MongoDB
  - $near and $maxDistance for finding entries near the user

```
Tandem.find({
            "languages.offer": offer,
            "languages.search": search,
            "location": {
                $near: [latitude, longitude],
                $maxDistance: 6
            }
 })
```

Offline Challenge

- **Context**:
  - Network connection loss
  - Detection online/offline status
- **Adaptation**:
  - Client sends requests to cache and not to server
  - Use cached Data from Apollo Client (InMemoryCache)
  - Get notification about connectivity status
    - Context Source: last queries

Connectivity/Offline Challenge:

- Obtaining the connection type
    - NetInfo from React Native
    - Handle connection type changes

```
NetInfo.isConnected.addEventListener(
        'connectionChange',
handleFirstConnectivityChange
);
```

Usability Challenge

- **Context**:
  - User changes his location
- **Adaptation**:
  - Show different tandem matches based on user input
    - Context Source: database

Adaptation of Communication

- **Lazy Evaluation**:
  - first load only certain number of Tandem matches
  - on scroll: data fetched from database, Tandem matches added

- **Caching:**
  - get last loaded queries from Apollo-Client

Lessons learned and pitfalls

Learned

- Network connection
- Lazy Evaluation
- Client side cache

Challenges

- Versions of some packages
- Different methods to use frameworks of our choice

TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Thank you for your attention!