

Application Development for Mobile and Ubiquitous Computing

Travelper Final Presentation

Group 2

Cong Lian, Lisa Werkmeister
Dresden, 16.December 2016

Agenda

- 1 Application Scenario
- 2 Use Cases
- 3 Mockups
- 4 Challenges
- 5 Adaptation Concept
- 6 Architecture
- 7 Work plan

1 Application Scenario

Motivation for **Travelper**



Lots of **social projects, Backpackers** around the world
(especially in developing countries)

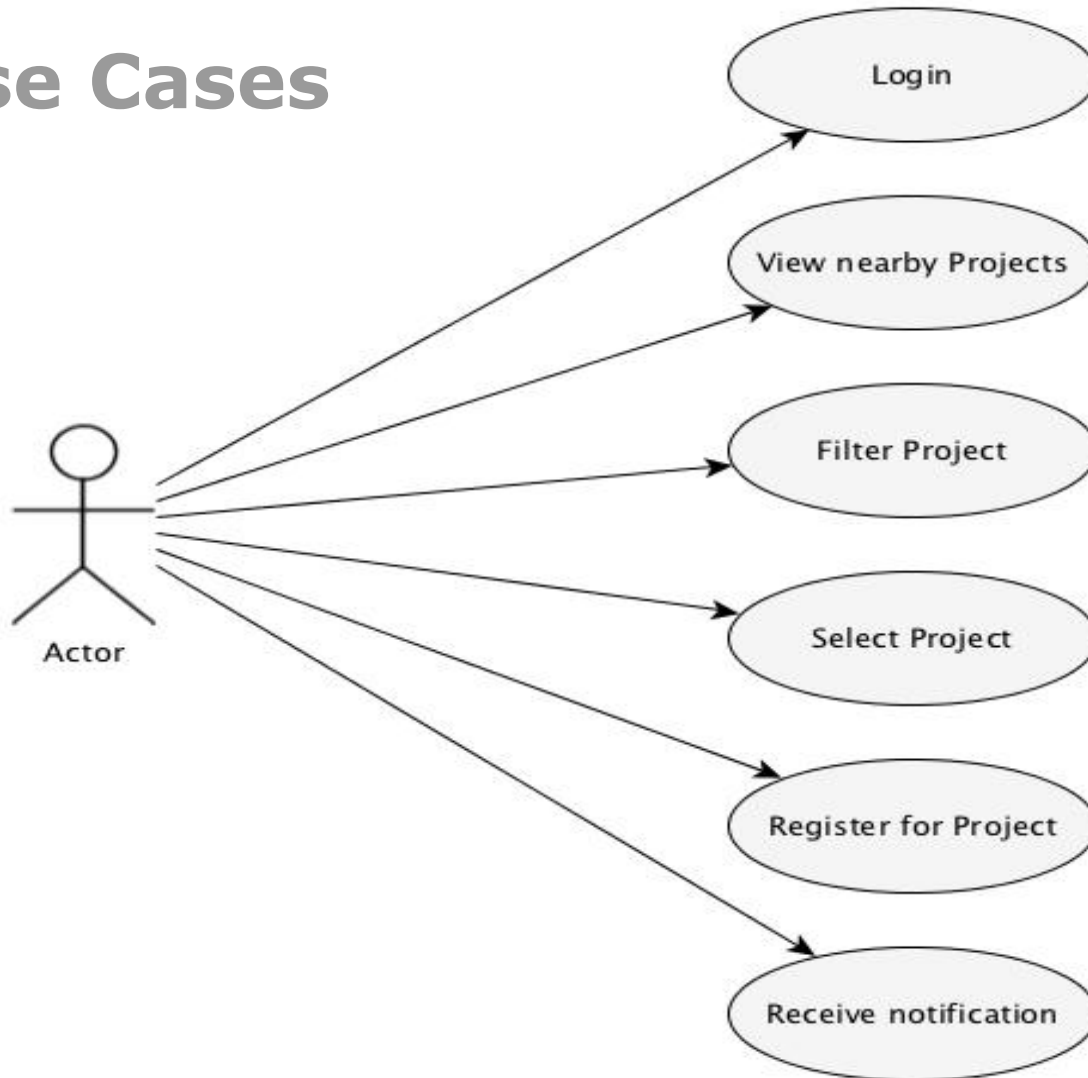
From Backpackers' perspective

- └ want to help, want to be volunteers
- └ but only have a few days on site

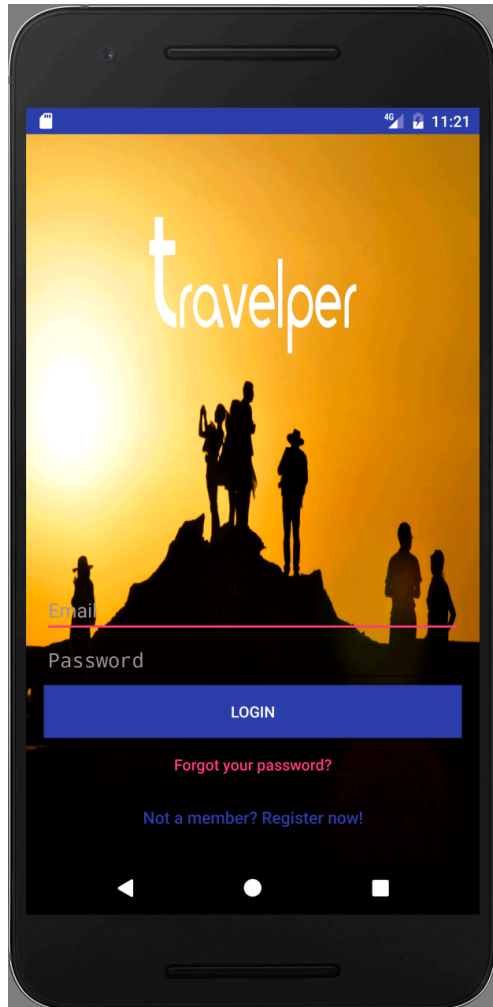
From social projects' perspective

- └ high costs (human resources) to maintain the organization
- └ high planning and administration efforts to carry out projects
- └ some projects are only held temporarily (one day/ a few days)

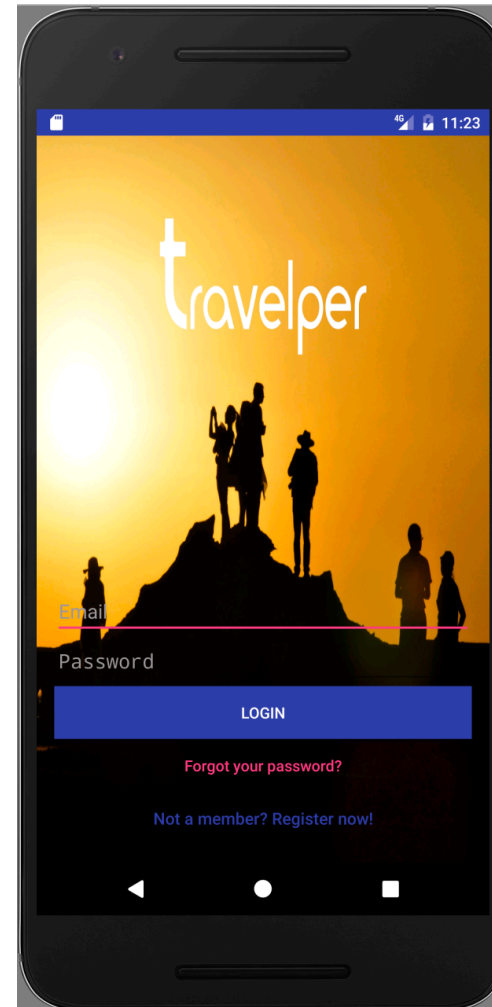
2 Use Cases



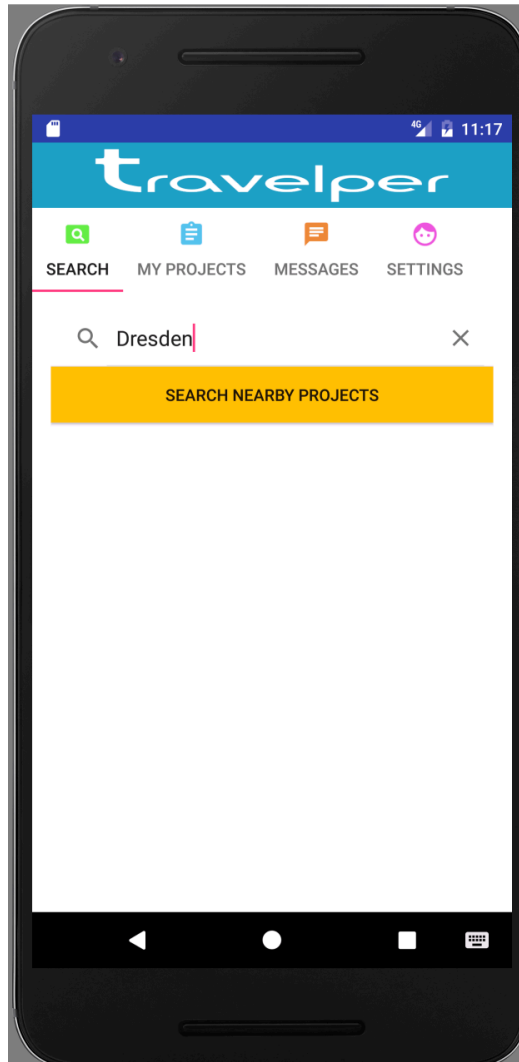
Login



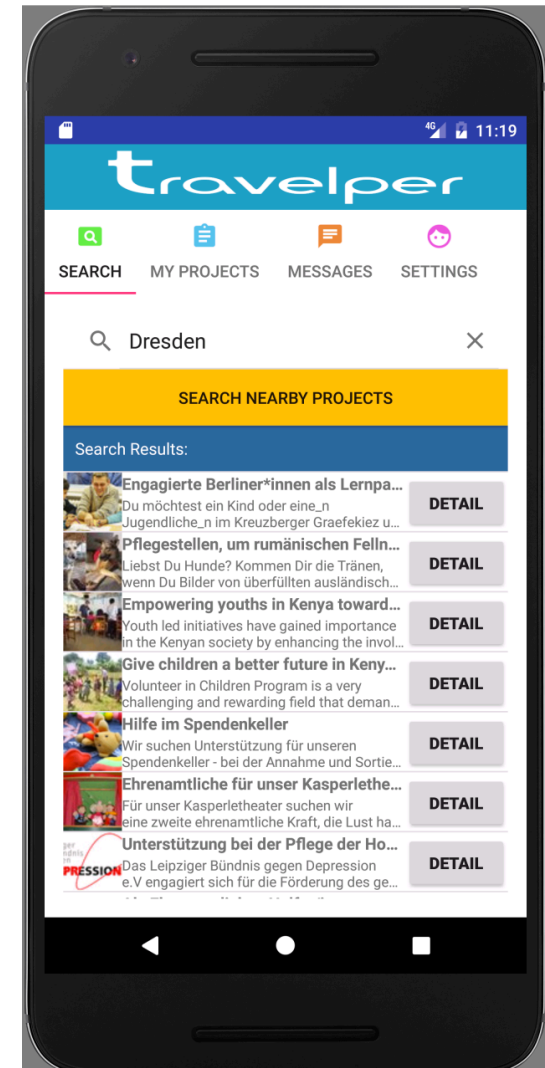
Register



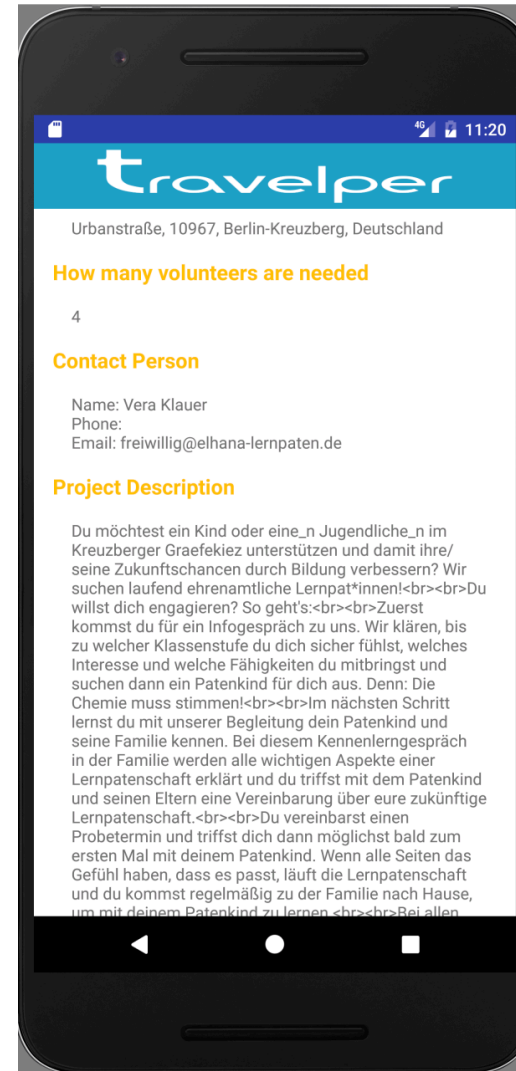
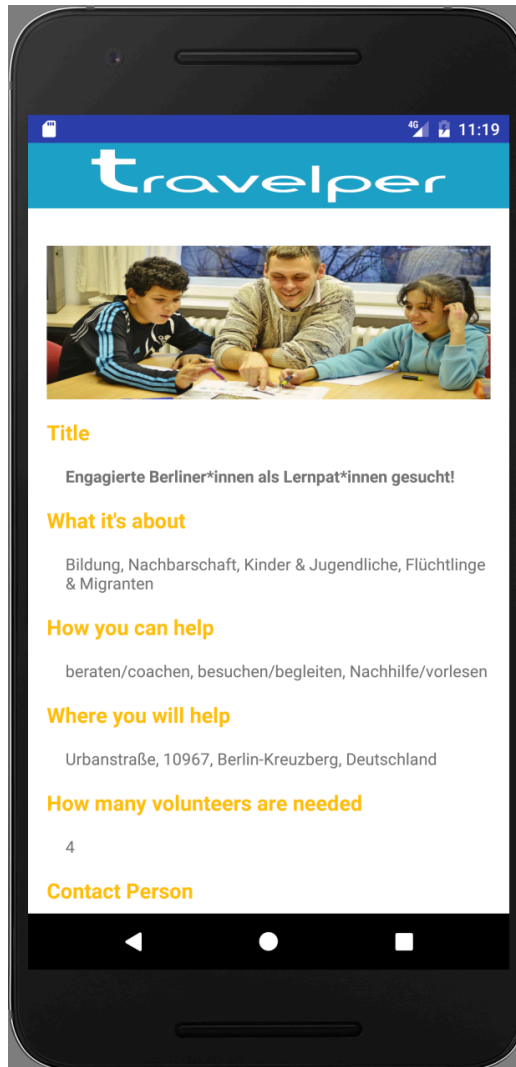
Search



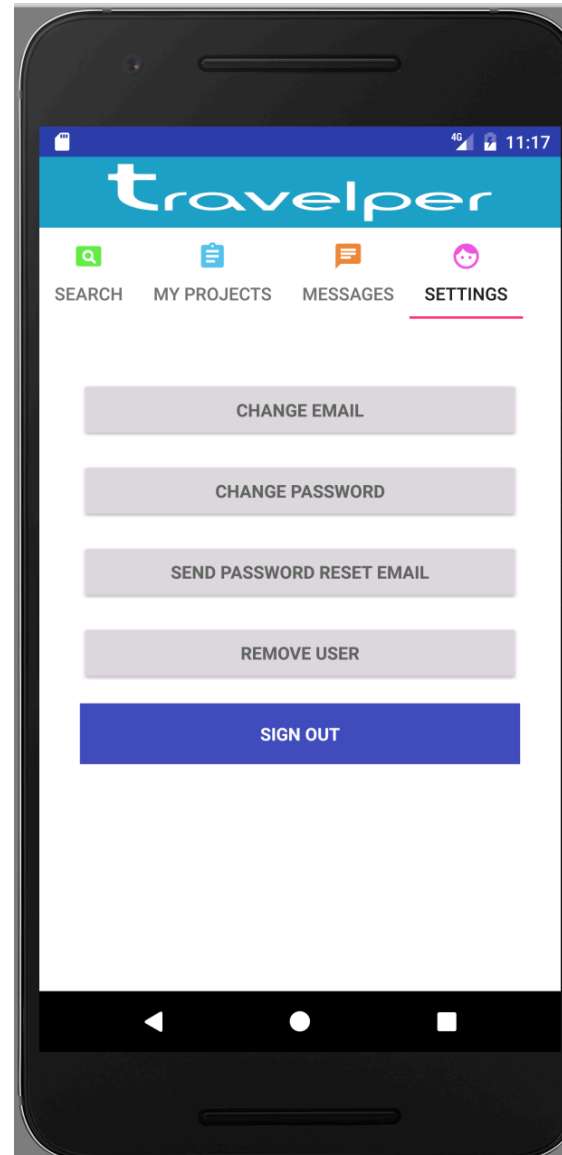
- **Search** by current location or type in City name



Detail View



Settings



4 Challenges

Usability Challenge

- adapt to different screen sizes
- adapt to user's current location to show search result
- adapt to user's profile (preference) to show search result
- share via social media, e.g. facebook

Connectivity Challenge

- provide only text type information when connection is poor/unstable

Offline Challenge

- local storage of last search result
- local storage of my projects (in which I decided to participate)

5 Adaptation Concepts

1) Location awareness

- Context: current location (physical context)
- How to capture: `android.location.LocationManager` (NETWORK_PROVIDER & GPS_PROVIDER)
- How to use: append to `HttpRequest` (`?around="currentLocation"`)
- Display results from near to far

2) Network awareness

- Context: network condition (technical context)
- How to capture: `android.net.ConnectivityManager`
- How to use:
- Download images if connection is fast enough (TYPE_WIFI II NETWORK_TYPE_LTE II ...)
- Otherwise only text

6 Architecture

Client/Server architecture

Client

- android app
- facebook APIs
- google map APIs
- GPS for location functionalities

Server

- Firebase
- Content:
betterplace.org API



Open issues

- Saving projects (Offline challenge)
 - Internal storage
- Messaging function
- Sharing on Social Media
- Save registration with email validation
- Consistent design
 - Colors, paddings
- Images scale to screen size
- Landscape mode
- Filtering, Sorting

Lessons learned

- Get basic functionality first then design
- Use dummy data before implementing API (betterorg API very complex)
- Time management essential