

Application Development for Mobile and Ubiquitous Computing

Travelper Second 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)

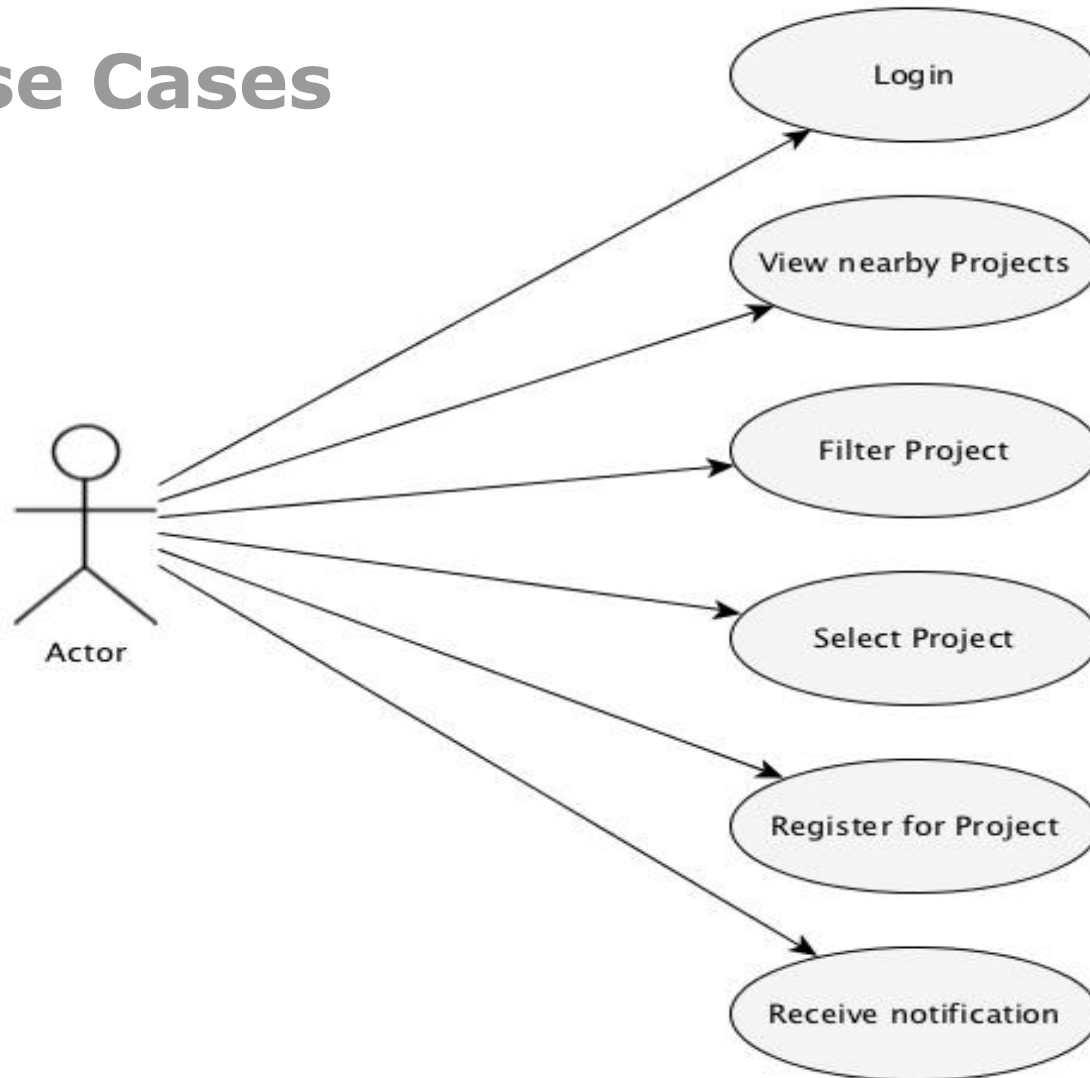
From Backpackers' perspective

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

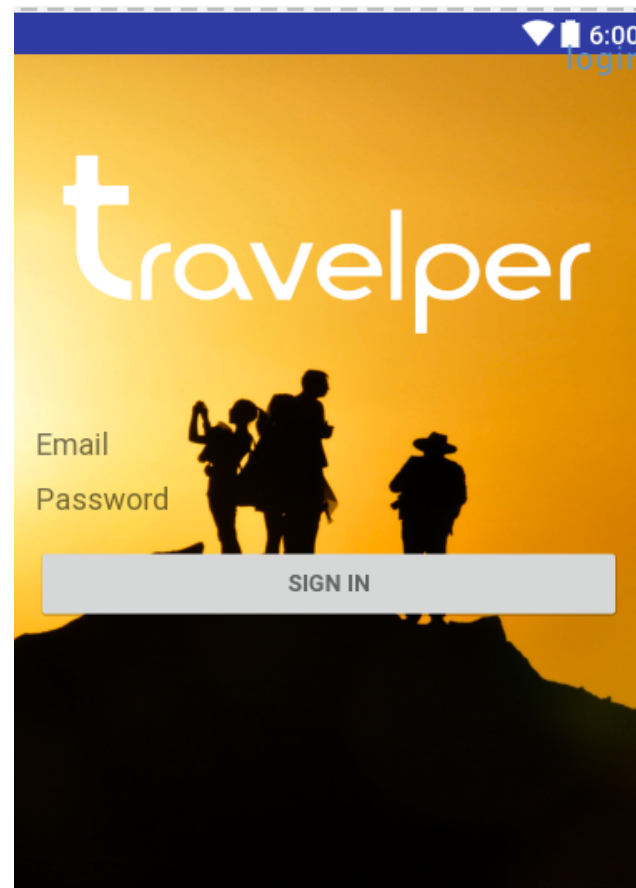
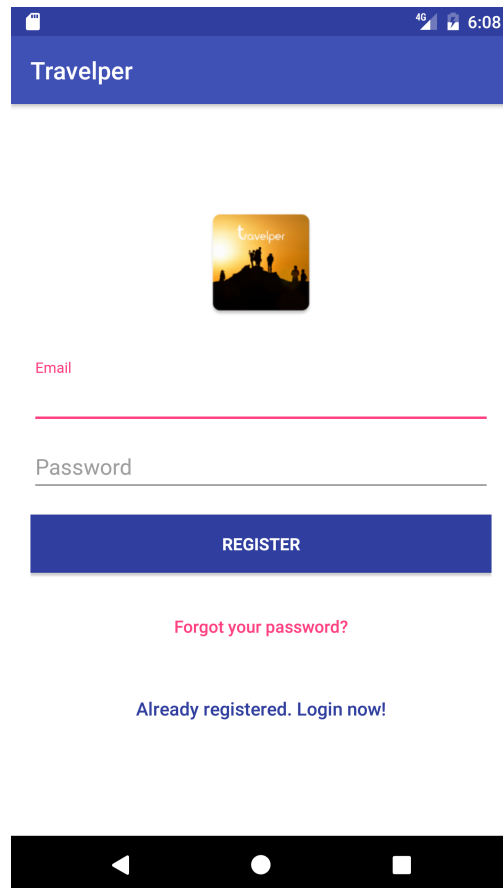
From social projects' perspective

- └ 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)

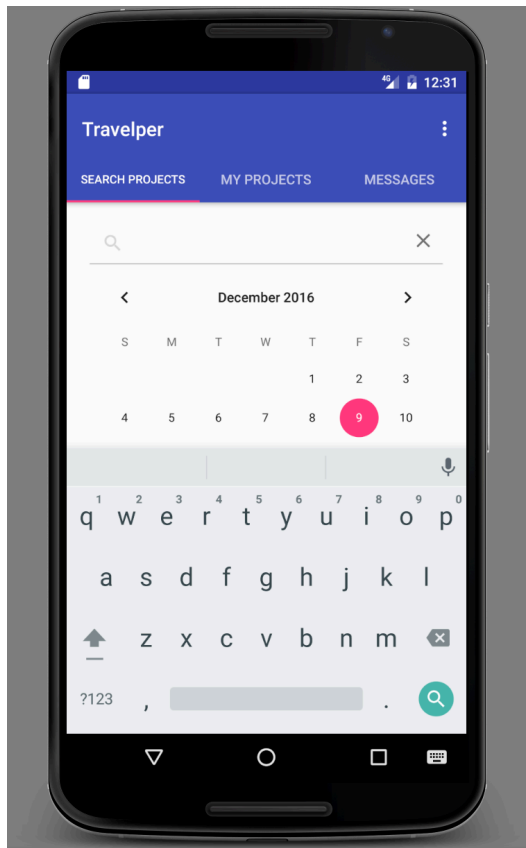
2 Use Cases



Login

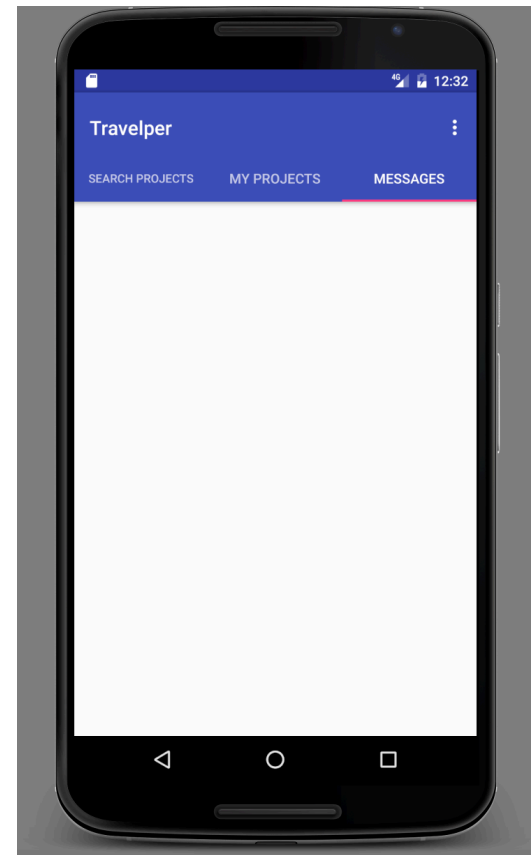
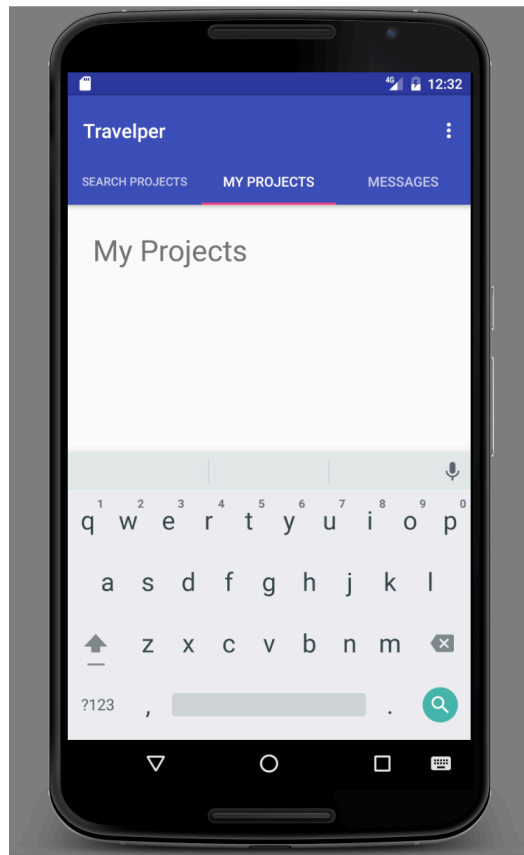


Search

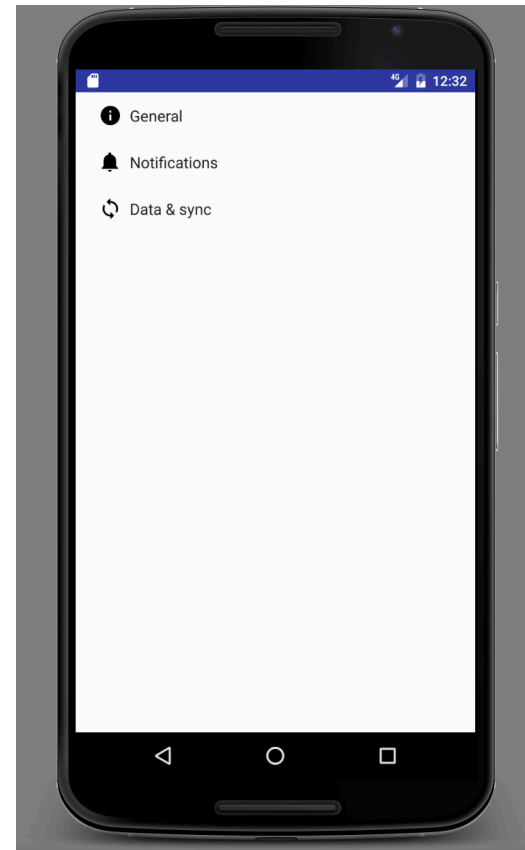
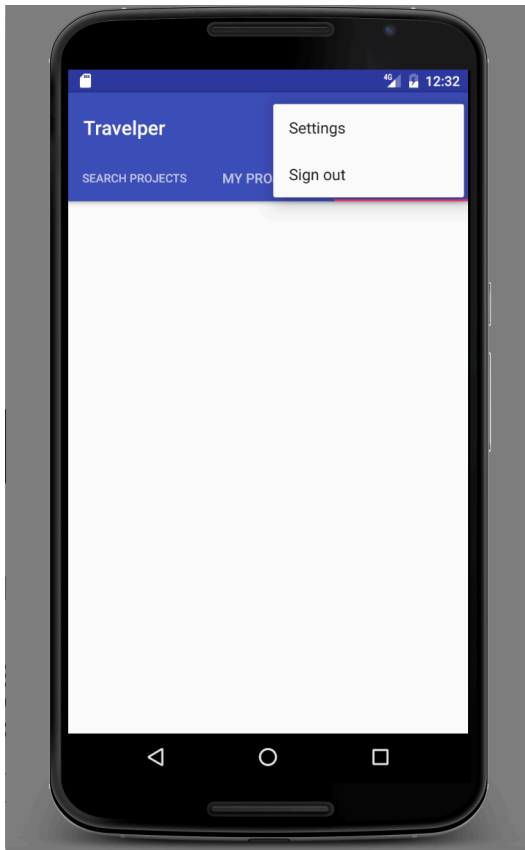


- **Search** by current location or type in City name
- **Filter** through check boxes
 - Distance:
 - 0 - 25km
 - 25 - 50 km
 - 50 - 100km
 - from 100km
 - Vacancy by Date (choose date range)
 - Price: Free
- **Sort** by Distance, Relevance, Price

3 Mockups



3 Mockups



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
- How to use: floatDistanceTo(Location dest)
- Calculate the distance to **filter out and sort** results

2) Offline usage

- Context: user input (explicitly)
- How to capture: button (one click)
- How to use:
- Get input from user to **save data on device** explicitly
- Always **cache** the last search result implicitly (Volley)
- Caching takes place whenever connection is **stable**
- Save into **XML file**

5 Adaptation Concepts

3) Network awareness

- Context: network condition (technical context)
- How to capture:
 - `android.net.ConnectivityManager`
 - `android.net.NetworkInfo`
- How to use:
 - `NetworkInfo getActiveNetworkInfo()`
 - `int getType()`
 - `int getSubtype()`
 - Get network type to decide more or less resources to transfer

6 Architecture

Client/Server architecture

Client

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

Server

- Firebase
- Content:
betterplace.org API
(adjustments needed)



7 Work Plan

So far

- UI
- Architecture
- Identifying APIs

December:

- Work on Design
- Integration of content database
- Server-side implementation

January:

- Test/Bug fixing
- Finalization
- Documentation/Review of the project

Thank you for your attention!