

Application Development for Mobile and Ubiquitous Computing

TimeTracker - Adaptation Presentation

Robin Gander - Riccardo Steffan

Technische Universität Dresden - Faculty of Computer Science

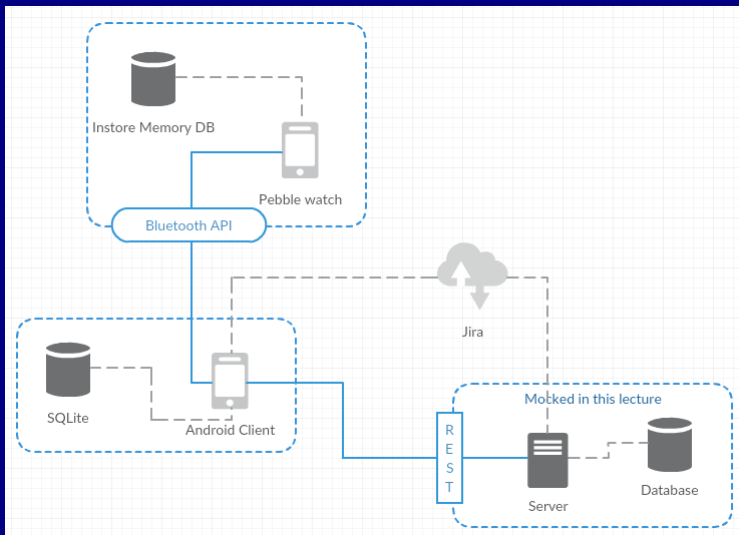
15.12.2017



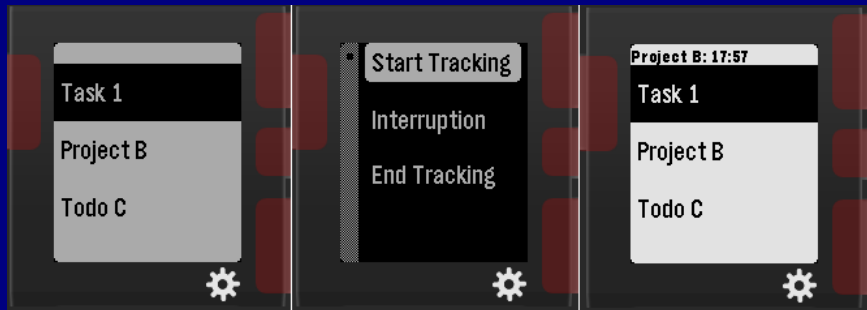
Summary

- 1 Architecture
- 2 UI Prototype
- 3 Technologies
- 4 Context Adaptation Mechanisms
- 5 Work Plan

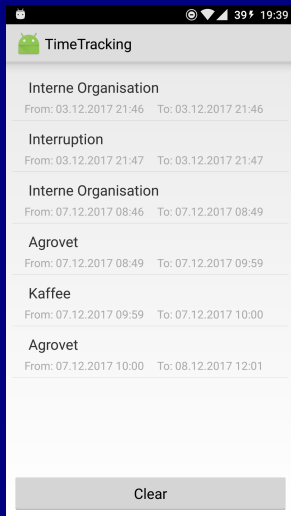
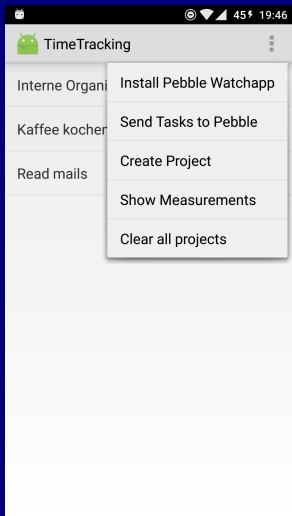
Architecture



UI Prototype



UI Prototype



Technologies

- 1 Android API 19, 4.4 KitKat (Java 7)
- 2 Room Persistence Library
- 3 Pebble (C SDK)
- 4 PebbleKit for communication (send / receiver messages)
- 5 OAuth for Trello authorization / authentication (ScribeJava library)
- 6 Volley for HTTP data transmission

Why Trello?



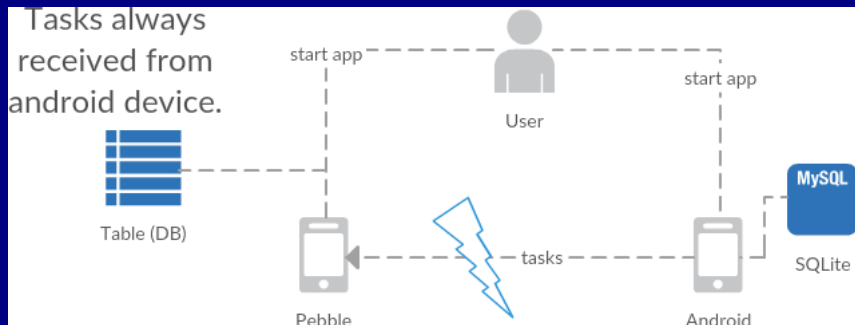
Advantages over Jira:

- No license needed for app integration – > No need to buy license or implement mock server
- Trello OAuth API supported out-of-the-box by ScribeJava – > Simple and robust implementation of authorization / authentication mechanism
- Streamlined REST API – > Easier to obtain and extract relevant project data

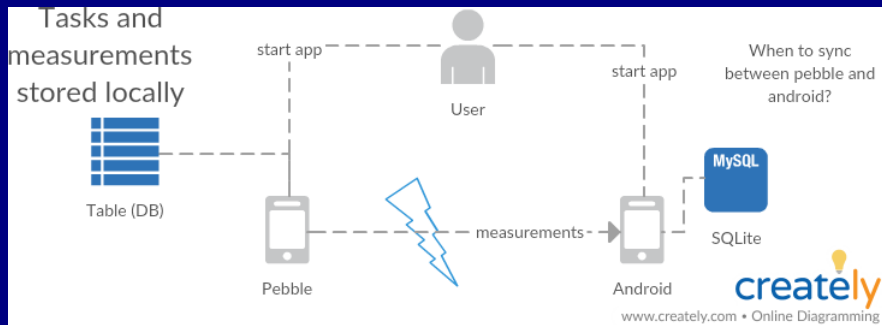
Overview

- Trello Connectivity: connect to the user's Trello account to obtain the project's data needed for tracking (i.e. tasks and their due date)

Offline Challenge 1



Offline Challenge 1



Offline Challenge 1

- Challenge: choose tasks and track time with unavailability of the smartphone.
- Context: Detect if the bluetooth connection is online
- Adaptation:
 - 1 – store tasks on pebble
 - 1 – handle inconsistencies between pebble and android with a 'rebook task'
 - 1 – for first time usage: 'rebook task' and 'unrelated'

Offline Challenge 2

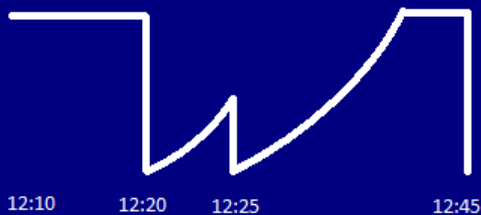
- implement a connection service to check if phone is connected
- false:
 - store the measurement on the pebble
- true:
 - check all persistent measurement on the pebble and send them (delete afterwards)
 - send the new measurement to the android phone

Persistence Challenge background

- Only 4 kB of persistent storage space on the pebble side
 - at maximum 256 keys (pebble dev suggestion is to only use 'a few' keys)
 - 0..9 for tasks
 - 10..49 puffer for tasks
 - 50..79 for measurements
 - 80..149 puffer for measurements
 - 150..169 for interruptions
 - 170..240 puffer for interruptions
 - 241 for selected task

Persistence Challenge

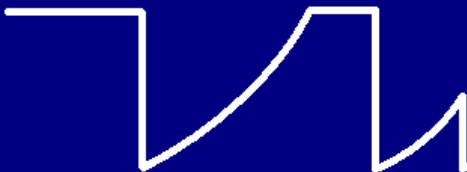
Context: Persistence storage is full



Task A	12:10	12:20
Task B	12:20	12:25
Task A	12:25	12:45

Persistence Challenge

Adaptation option: merge nearby connected tasks together



Task A	12:10	12:40	Interruption	12:20
Task B	12:40	12:45

- The curve doesn't fit, the velocity still does and we save some storage (6 Bytes).

Simplicity Challenge

- useful ordering of the tasks on the pebble
- detect movement automatically to suggest task switches
 - can implement a `HealthEventHandler` listening to `HealthEventMovementUpdates`
 - e.g. listen to steps in the past 5-10 seconds, if greater than threshold (like 5 to 10 steps) then trigger

Trello Connectivity

- **Authorize / Authenticate:** an OAuth session token is obtained using an application key and secret (stored in strings.xml), control is then passed to Trello for user's login (no need to locally store username and password)
- **Obtain project data:** query Trello through the REST API (e.g. GET /1/boards/[board-id]/lists)
- **Data selection:** before tracking starts, the user selects the projects (i.e. boards) and tasks (i.e. cards) from its account that needs to be tracked

Work Plan

November

03.11. First presentation

- Knowledge gathering about pebble development
- Knowledge gathering about the Jira API
- UI design
- First prototype

December

- Usage in working environment
- Adaptations on the application feeling

15.12. Adaptation concept presentation

January

- Bug fixing

26.01. Final presentation