Application Development for Mobile and Ubiquitous Computing

Seminar Introduction

Dr. Ing. Thomas Springer
Technische Universität Dresden
Chair of Computer Networks
Basic Idea

- To implement your first app
- To familiarize with one of the popular implementation platforms for Mobile and Ubiquitous Computing
- Design concrete mechanisms to adapt your App according to some relevant context
- It will help you to understand basic idea of this course
- These concepts are one of the basic parts of the exam
You should work in **groups of 2 students!!!**

You have to define your own app idea
- Define an application scenario from the domain of mobile and ubiquitous computing
- Select a mobile device platform for implementation
- Develop a concept for adapting your App

First presentation:
- You should present your task description and technology selection (**03.11.2017**).

Adaptation concept presentation
- You should present at least two concrete mechanisms to adapt your App and how they are controlled by particular context information (**15.12.2017**).
- **Deadline for slide submission is 08.12.2016**

Final presentation
- You should present your adaptation concept and final results (**26.01.2018**).
First Presentation

- Please prepare a presentation for 10 minutes!!!!

- First presentation should include
  - present the application scenario in detail
  - show use cases, e.g. a mockup!!!
  - identify in detail the challenges of mobile computing you want to tackle
  - define the technologies you intent to use
  - propose an architecture
  - work plan
Presentation of Adaptation concepts

- Please prepare a presentation for 10 minutes!!!!

- **Adaptation concept** presentation should include:
  - A set of *concrete context features* you capture to control your adaptation
  - The description of at least 2 *concrete adaptation mechanisms* for your App
  - The method to map your considered context features to parameters for controlling your adaptation mechanisms

  - The detailed architecture of your App
  - The technologies for implementing all components of your App
Please prepare a presentation for 10 minutes!!!!

Final presentation should include

- Present the application scenario with use cases, screenshots, a video or the running application
- Describe the final architecture, components and interaction between components
- Describe the technologies adopted for each component
- Discuss in detail the challenges of mobile computing you have tackled
  - What adaptation mechanisms have been used
  - What context information has been involved
- Discuss open issues and lessons learned
Submission and Finalization

- Submit your presentations no later than
  - First presentation – **02.11.2017**
  - Adaptation concept presentation - **08.12.2017**
  - Final presentation – **25.01.2018**

- Submission is by sending the slides by email to
  - thomas.springer@tu-dresden.de
  - Name document groupXX.pdf
  - Mail subject: [ADMUC17]

- A colloquium will be held based on individual appointments with each team at the end of the term which includes:
  - presenting the running application
  - explaining the project
  - discussing adaptation mechanisms, used context and further lecture content considered during the development of the App
You can use the whole term to fulfill the tasks
Projects are compulsory for all students
Don`t miss deadlines for slide submission

Auditorium for discussions

Email to send presentations and solution is:
  • thomas.springer@tu-dresden.de
  • Mail subject: [ADMUC17]

Project state and results are shown on lecture web page
  • please refer to „Seminar task list“
HOW TO DEFINE YOUR SEMINAR TASK?
Think your own task

- select an application scenario relevant for Mobile and Ubiquitous Computing
- scenario should tackle at least one of the challenges mentioned in the first lecture
- scenario should include concrete mechanisms for context awareness and adaptation
- could be a stand-alone App on a mobile device or a distributed application with server component
- implementation should be based on a mobile device platform
  - Android
  - iOS
  - Cross-Platform tool
EXAMPLE

Fahrtfinder, a carpooling App
- On March 27th, carpooling.com GmbH (who runs amongst others market leader mitfahrgelegenheit.de) introduced **11 % fee**
- in response, many free websites gained market share
- **problems:** many websites ⇒ user has to search each site
- **solution:** meta search engine for carpooling web sites
Mockups

Dr. Thomas Springer
Application Development for Mobile and Ubiquitous Computing - Seminar Introduction
Use Cases

- Search for lift for specific date/time and location
- Contact driver (SMS, email, telephone)
- View details for specific lift
- Add lift to bookmarks
- Add lift to calendar
- View last queries
- Share lift with friends
Challenges

- Connectivity Challenge
  - Reduce the amount of data to be transferred via the wireless link (use server component for search, processing and aggregation of lift data to reduce network traffic over wireless link)

- Context:
  - Detect type and speed of network

- Adaptation:
  - Introduce proxy on server side for data fetching and preprocessing
  - Reduce quality and size of images
  - Lazy evaluation on client side (text first, images later)
### Challenges

- **Offline Challenge**
  - Provide some functionality when offline (e.g. bookmarked lifts, last queries, booked lifts in calendar)

- **Context:**
  - Detect if your App is online or offline

- **Adaptation:**
  - Caching of Data
  - Use cached data if application is offline, Update cache if connectivity is good
  - Persistently store information from cache
Challenges

- **Usability Challenge**
  - Present lift information properly on iPhone and iPad display (iPhone 3.5 inch and iPhone 4 inch screen) and iPad screens
  - Minimize effort for user input

- **Context:**
  - Detect screen size and device type
  - Detect user location with GPS

- **Adaptation:**
  - Adapt layout for iPhone and iPad
  - Auto fill “Start field” by using GPS location
  - Cache search queries
Technologies

- **Client:**
  - iOS on iPhone
  - (some additional HTML parsing on external websites)
  - we will keep to fahrtfinder’s corporate design and style guide
  - GPS for location tracking
  - SWRevealViewController library for navigation

- **Server:**
  - provided by fahrtfinder.net, they offer a JSON API
Architecture

- Mobile Client
- Web Server with JSON API
- Web Server from Provider 1 with custom API
- Web Server from Provider 2 with custom API
- Web Server from Provider 3 with custom API
- Web Server from Provider 4 with custom API
- Web Server from Provider 5 with custom API
- Web Server from Provider 6 with custom API
- Web Server from Provider 7 with custom API
- Web Server from Provider 8 with custom API

Bookmarks saved locally

- GPS
- SWRevealViewController,
  UISplitViewController

wired connection

specific webpage

wireless connection

Caching
Filtering

metadata
Work plan

- **21.10.2013**: Begin of iOS / Objective-C tutorial (weekly until 03.02.2014)
- **24.10.2013**: Begin of implementation
- **01.11.2013**: First presentation
- **21.11.2013**: First prototype
- **20.12.2013**: Second presentation
- **20.12.2013**: Begin of testing on real devices, bugfixing on iPhone 4S / 5
- **31.01.2014**: Final presentation
- **31.01.2014**: Launch in the App Store

- think about team organisation
- assign responsibilities in team
- make yourself familiar with platform and development tools
- think about context and adaptation
- start early with implementation
- develop agile -> feature planning and periodic releases of running code
- plan time for testing and bugfixing
- deploy and test with real devices
Further Examples

- **MusIQ**
  - Simple music player with basic functionality
  - Application fetches additional information (video, biography, discography, nearby events) according to currently playing song and user location

- **CarSurfing**
  - Supports matching between hitch hikers and car drivers
  - Uses direction and GPS information
  - Visualization on Google Maps
  - Integration of Facebook to increase security
Further Examples

- **Lunchbox**
  - ultimate mensa guide providing information about the meals at the refectories including students ratings
  - data provided at server, scraped from Studentenwerk page

- **MusicMap**
  - find the places where your favourite music is played
  - integrates LastFM
  - uses location and genre to cluster users
  - visualization on Google Maps