

Mobile Services Exam Projects



F. Ricci
2009/2010

Generalities

- ❑ The project should require more or less 40 hours
- ❑ The project “deliverables” includes:
 - The code (one zip file): must be deployable in the wireless toolkit and must follow the standard structure of a WTK project
 - One text file (readme.txt) describing all the required steps to install and run the program on the WTK
 - One document (pdf file, **less than 3000 words**) describing:
 - ❑ The system functions and the human computer interaction: what is doing and how a typical interaction proceeds (use snapshots from the GUI)
 - ❑ The structure of the code (classes and files): describe the role of each class and the main methods
 - ❑ The major technical problems that have been tackled and how these have been solved.

About the grade

- I will evaluate the following elements:
 - **Usability of the system** (no errors, no strange commands, easy to learn, enjoy using the system)
 - System functions coverage – with respect to the project description
 - Quality of the presentation
 - Quality and completeness of the report
 - Deployment and running - how fast I will deploy and run the system on my PC without errors
- Present your project at the last class if you want to pass at the winter session
- The presentation must be contained in 15 mins and must illustrate the same points described in the report
- **If you reuse existing code you must clearly write what code you used and what have you done on top of that!**

Next steps

- Build a group of 2 people or work alone
- Select a project among those described here
 - Or propose some changes to one of those
 - or write down a project description in a similar way
- Send an email to me (as soon as possible) with the following data
 - Group members: names and id numbers
 - Selected project: if it is not one of those suggested include the project description, or if you propose some changes (new functions) please describe them in the email.

P1 – Decide Together a TV Program

□ **Functions**

- Two or more people (e.g., a family) should be able to search in a TV program list
- Search program by keywords or metadata (genre)
- Browse detailed information on the program (plot, pictures, etc.)
- Select one program and make a proposal to your peers
- They can reply with 'yes' or make a counter offer
- Design a process that will "converge"
- Store the selected program in a local database (all the peers)

□ **Implementation details**

- Find a list of TV programs from the web or build your own sample db
- Communicate with your peers with SMS (or sockets, if you have wifi).

P2 – News Recommendations

□ Functions

- Download a list of News from a RSS feed (e.g. BBC http://newsrss.bbc.co.uk/rss/newsonline_uk_edition/technology/rss.xml (better: a source of local news for Bolzano))
- Browse the result list
- Browse detailed information about the news
- Select a news and store it in a local store (organize the news in categories)
- Implement a function that **suggests** some items based on the similarity to some of the news in your local store

□ Implementation details

- Use an existing RSS feed
- Parse the XML using an existing library like kXML
- Exploit the RecordStore library

P3 - Transportation

□ **Functions**

- Insert your current position, the destination and eventually the time (if the current time must not be used)
- Get the timetable of the bus going to destination starting from your position

□ **Implementation**

- Position can be
 - inserted by the user with a code
 - or using the location API
 - or from a list of bus stops (the last 5 inserted)
- The timetables are stored locally
- Connect to a server to download updated timetables (optional).

P4 - Task List Alert

□ **Functions**

- Imagine a **particular** Field Force Automation situation and design the application for that scenario: e.g., fuel distribution, parcel delivery, care giver ...
- Add, delete, modify a task (and its features)
- Add, delete, modify a task category
- View tasks filtering by date, category, keyword, location (according to the application)
- Build an alert function that alert you
 1. when a task is due
 2. you are in a good situation to perform the task (e.g., approaching the right location).

□ **Implementation**

- Use the MIDP record Store, and the location API

P5 - Care Giver

□ **Functions**

- Service Request (an ill person in need of assistance)
 - User sends help request by pressing a key
 - She gets a response from the care giver (read and confirm receipt)
- Receive directions from care giver
 - User receives an alert message (e.g. "take the pills")
 - The alert is repeated if no confirmation is given
 - User acknowledges the receipt and the acceptance to the caller
 - After a timeout the caller is warned that the receiver has not acknowledge the request

□ **Implementation**

- Use SMS for exchanging information between the care giver and the user
- The MIDlet must listening for incoming SMS and make the interaction simple and easy to manage.

P6 – Geo Tagging

□ **Functions**

- Record the position and a note related to the position (picture, comment, features) during a walk (GeoNote) – imagine a specific application scenario (e.g., observation of diseases of apple plants)
- Save the GeoNote with a name in a local repository
- Upload a GeoNote to the server
- Download the GeoNotes that are closer to you and on topics that you are interested in

□ **Implementation**

- Position can be obtained using the location API
- The collections are stored locally
- Implement a server for uploading/downloading the GeoNotes

P7 – Orienteering

□ **Functions**

- Implement a game where a user get a mission on the mobile phone
- The mission consists in reaching a certain number of targets
- When you get a target you get the description of the next target (e.g., find a location where there is x)
- The mobile must log your actions
- At the end of the game the mobile debriefs the game
- Send the mission execution log to another user

□ **Implementation**

- Position can be obtained using the location API
- The missions are stored locally by the application and randomly assigned to the user.

P8 – Domotics (Home Automation)

□ **Functions**

- Control the state of your heating system (raise the temperature in one room, cover/uncover the solar panels, start-up/shut-down the furnace, ...)
- Receive an alarm from your furnace (blocked, etc.)
- Log the actions and the alarms in a record store and browse them

□ **Implementation**

- Build a nice GUI for this application (simple!)
- Use SMS or sockets connections with push
- Simulates the events in the furnace and the actuation of commands
- Imagine that in the house there is another "phone" that communicate with your mobile.

P9 – Event Remainder

□ **Functions**

- Download a list of Events from the following XML Feed (<http://wiki.ectrldev.com/ricercaEventi.xml>)
- Browse the result list
- Browse detailed information about the events
- Select an event and store it in a local store and in the calendar (using PIM API). Organize the events in categories.
- Implement a function that **suggests** some events based on the similarity to some of the events in your local store

□ **Implementation details**

- Use an existing XML feed
- Parse the XML using an existing library like kXML
- Exploit the RecordStore and PIM API