This report describes the TV-Watcher project realized during the course of Mobile Services
Contents

1. Project description .................................................................................................................. 3
2. Goal of this project .................................................................................................................. 3
3 Internal structure (important parts)........................................................................................... 4
   3.1 TV program list .................................................................................................................. 4
   3.2 Temporary storage of pending votes .................................................................................. 4
   3.3 Storage of past votes .......................................................................................................... 4
   3.4 Receiving SMS .................................................................................................................. 5
   3.5 Send SMS .......................................................................................................................... 6
4 System architecture .................................................................................................................. 7
5 The Code Structure .................................................................................................................. 8
   5.1 Default package ................................................................................................................ 8
   5.2 Logic package ................................................................................................................... 8
   5.3 XMLParser package ......................................................................................................... 8
6 Human / Computer interaction ................................................................................................. 9
7 Technical issues ....................................................................................................................... 12
8 Possible improvements for the next releases ............................................................................ 12
1. Project description

The main purpose of the TV-Watcher J2ME application is to help TV-consumers with more than one person on each television choosing together their program. This will be done via voting by using their mobile phone.

TV-Watcher is a simple peer to peer application which does not require an expensive mobile phone to run on it. There is no need of a touchscreen, no Internet connection, no Bluetooth and no GPS. The p2p connection will be established only via SMS.

This application allows a user to navigate through a TV program list by specifying the desired date and time. After this a list of all’ available programs corresponding to the specified criteria’s will appear. The user has now the possibility to get more specific information about the different TV programs. If there is a TV program, which the user likes, he can simply invite as many friends as he want from his phonebook or enter a new phone number. After this step, an invitation SMS will be sent to all chosen numbers. Now the screen of the initiator shows all the important details about the current voting progress, like the number of accepted and declined votes. In addition all the names of the peers with their corresponding answers will be displayed. The initiator can terminate the voting progress whenever he wants. After this, a message with the outcome of the voting will be sent back to all participating peers.

When a user receives an invitation SMS the proposed TV program, including all details, will appear on his screen. Now the user can accept or decline this proposal. His answer will be sent back via SMS to the initiator. When the voting is over, there will appear an alert which informs the user about the outcome of the voting progress.

Each outcome will be stored locally on every participating device.

2. Goal of this project

The goal of this application is to vote for a TV program, even if the people are at different locations. Therefor it’s possible to safe time, because they don’t have to discuss about the TV program and they can immediately start watching a TV program.

This application should be easy to use and it should run on nearly every mobile phone, which supports J2ME applications. For this reason, the application must be internally as simple as possible, without using any complex API’s, which could not be supported by cheaper mobile phones. This is a very important aspect, because families which children have only a cheaper mobile phone could also use this application.
3 Internal structure (important parts)

3.1 TV program list

Since this application needs no Internet connection, the TV program list must be stored locally. Therefore the program list will be stored in a XML file, which will be copied on the mobile phone during the installation. When the user starts the application the XML file will be parsed and converted into a vector, which facilitates the access to the data.

For this purpose we have used kXML, which is an API especially designed for MIDP devices.

The TV program XML file has the following structure:

```xml
<tv program>
  <program title = "Title1" progID="00001" startTime = "00:01" startDate ="1/1/2012" endTime = "01:00" endDate = "1/1/2012" genre = "Genre"
  img = "/pics/00001.jpg">
    <description>The Description</description>
  </program>
  <program title = "Title2" ...
</tv program>
```

The attributes startTime and startDate allows us a fast research for the corresponding date and time restrictions by the user.

3.2 Temporary storage of pending votes

When the user responds to an invitation, we must keep track to which invitation the user answered and that the outcome of the voting is pending. For this reason we store the answer temporarily into a record store until the user receives the outcome of the voting. After this, the answer will be removed from the temporarily record store. This must be done to ensure that user knows on which invitation he has respond (in the case of more invitations).

3.3 Storage of past votes

When the initiator sends the outcome of the voting, the voting is finished. Now we delete the answer of the user from the temporary storage and we store the definitive outcome into the record store for the past votes.
3.4 Receiving SMS

Receiving SMS is the key point of the whole application. Without SMS there would be no communication between the different peers.

Before we try to understand how this application handles receiving SMS, we have to understand the communication protocol between the different peers.

**Communication protocol (between two peers)**

**Step 1:**
Initiator \(\rightarrow\) Invited

The Initiator sends an invitation SMS to his friend (called Invited). This message contains the program ID.

**Step 2:**
Initiator \(\leftarrow\) Invited

The Invited sends a response SMS to the initiator, containing accept or decline.

**Step 3:**
Initiator \(\rightarrow\) Invited

The Initiator sends an outcome SMS, which the outcome of this voting.

The message contains win, lose or draw and the program title

**Description:**

Every user can receive more than one invitation at the same time (we have not to wait until a voting is over to participate on a new voting) we have to put the receiving code into a separate thread, which will listen on port number 50001.

Since we can receive more than one invitation at the same time, we have also to listen for the different invitation responses. For this reason we put the response listener into a separated thread which will listen on the port number 50002.

At the end, after each voting we have to wait for the outcome. There can be more than one outcome, and for this we put the outcome listener into a separated thread, which will listen on the port number 50003.
3.5 **Send SMS**

In this application we distinguish between 3 different types of messages, which are:

- **Invitation messages** = message sent from the initiator to the invited. This message contains only the TV program ID and will be transferred over port 50001.
- **Response messages** = message sent from the invited to the initiator. This message contains only accept or decline and will be transferred over port 50002.
- **Outcome messages** = message sent from the initiator to the invited. This message contains the outcome and the title of the TV program and will be transferred over port 50003. Example: lose,fargo
4 System architecture

As mentioned before, we are using a peer-to-peer connection between the different mobile phones. This means every mobile phone can act as a server or as a client. The initiator of a voting acts as a server, which coordinates the complete voting process. The participants act as a client, which communicate with the initiator. Every participant can become also an initiator.
5 The Code Structure

The code is divided into 3 packages

- Default package
- Logic package
- XMLParser package

5.1 Default package
This package contains

- TVWatcher MIDlet = this is the key part of the whole application, it manages the whole graphic user interface, manages the elections and starts the different threads (receive invitation thread)
- Sender class = this class manages sending of messages to the peers
- CollectVotes class = this is the thread which collects all the response messages from the participators
- VoteResponse class = this is the thread which collects the outcome messages from the initiator

5.2 Logic package
This package contains

- StorageManagement class = this class handles the methods for saving and loading the data from the record store. This class is used by the temporary storage for pending messages and for the past voting storage
- TVProgram class = this is the object definition including the getters and setters for the object TVProgram. It’s used after parsing the XML File and storing putting the data in form of a TVProgram object into the Vector

5.3 XMLParser package
This package contains

- ProgramParser class = this class parses the TV program information from the XML File and puts the TVProgram objects into the Vector.
6 Human / Computer interaction

This screenshots represents the main functionalities of the Application.

Main menu
(a) Actually there is now election...

(b) ... so [USER1] can make a new proposal (New voting) or he can view past elections (Past voting).

(c) If [USER1] click on New voting he can Select the date and time in which he wants to search for a program.

(d) [USER1] can see now the available movies for that specific time. With Details he can get more information about a movie or serial.
(e) Now the **USER1** can see all available information about his specific selection. With **Invite** he can start his proposal.

(f) If the other users which should join the proposal are not yet in the list, then the **USER1** can **ADD** them. Afterwards he **sends** them an **invitation**. E.g. at **USER2**

(g) Now **USER1** gets the window of the current voting state with the Number of accepted votes and declined votes. In addition each respond of the participatens will be listed including their phonenumbe}

(g) **USER2** gets meanwhile a proposal via SMS that he can **Accept** or **Decline**.
(h) If USER2 accept this proposal USER1 gets an actualized current voting table.

(i) USER2 gets now an actualized overview of all current votings.

(j) If the voting is finished USER1 gets the final status and can check if his proposal was accepted by the participators.

(k) Also the participator USER2 gets the final status.
7 Technical issues

One of the major technical issues was that in J2ME there are a lot of standard classes missing, which makes a simple problem becoming trickier. For example in J2ME there is no way to define the Date output format, in standard Java this would be no problem. For this reason you have to implement some methods by your own or use “tricks” to get to the same result.

Another technical issue is the usage of the PIM (Personal Information Management). We had to access the phonebook and so we thought to use PIM. The problem is that not all mobile phones support the optional JSR 75 PIM package, and so our application would not run on these devices. Therefore we solved this problem by implementing a Textfield with the “TextField.PHONENUMBER” property. For this reason, it’s not possible to access to the phonebook on the emulator, you have to install the application on your mobile phone in order to access the phonebook.

8 Possible improvements for the next releases

A possible improvement for the next releases could be to improve the performance of the code, so it would be possible to make the application faster on cheap mobile phones. The first step should be to reduce the usage of vectors to a minimum and use arrays instead of them.

Another possible improvement would be to add the possibility of sending a response combined with a comment. This means that for example a user can accept a proposal and append his own comment like “I will come 10 minutes later or buy something to drink”.

This actual release has a huge improvement to the first version. At beginning we wanted to give the user the possibility to accept, decline or to propose a counter offer. This idea was not so good, because when we allow the user to make a counter offer, we would have to create a more complex voting algorithm with a much higher data exchange between the different peers. This data exchange would cause very high costs and for this reason we should think about to plan a new option for this application, which uses an Internet connection instead of SMS. If we would realize this option, we could also solve the problem of updating the XML file in a very simple way by downloading a new XML file. This would be the best solution, because as we know a TV program could also change and in the current version there would be no possibility to handle this problem, therefore it would be good to download the updates from a web service.