Mobile Reporter

by Alexander De Luca and Michael Müller

Start: 01.03.2005 End 01.12.2005

Supervisors: Albrecht Schmidt, Enrico Rukzio

Prof. Hußmann

LFE Medieninformatik, Ludwig-Maximilians Universität München

Abstract

Mobile Reporter is a mobile blogging system containing a server application written in PHP 4.0 with a MySQL database and a mobile client as a J2ME MIDlet. Once the server is installed users can send in their news Items via different submission ways. Beside the classical submission ways over a web interface or via email, the Mobile Reporter system supports to publish Items via MMS or via HTTP post as implemented in the MIDlet. In this way the users can publish their news from nearly everywhere with their mobile phones.
The whole system was implemented as modifiable as possible and therefore new languages or templates can easily be added or the administrator of the server can configure all the settings at will.

Affirmation of Independence

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

_____ _____

Alexander De Luca Michael Müller

Table of Contents

Table of Contents

1. Introduction

This project was developed within the scope of a project thesis at the Ludwig-Maximilians University of Munich starting at the 03/01/2005. We'd like to thank all the people at the university that supported us during the work.

1.1. Motivation

In the last years the possibilities for citizens „playing an active role in the process of collecting, reporting, analyzing and disseminating news and information“ [1] grew rapidly. This phenomenon, called citizen jounalism or participatory journalism determined the report of such tragical catastrophes as the 2004 South Asian Tsunami or the 2005 London Bombings. With wide spreaded personal mobile cameras people captured these events or commented them on their personal weblogs. „The intent of this participation is to provide independent, reliable, accurate, wide-ranging and relevant information that a democracy requires“ [1].

With the phenomenon citizen journalism in mind there is definitely an increasing interest in blogging. Many people have their own weblogs and news all around the world are spread massively through this medium. For this reason an enormous amount of services, most free, is available for user to create blogs. Even big companys show big interest in this development. For example some years ago google bought the biggest blogging community of the world, blogger.com as can be read in [2]. This fact points up that blogging in common sense is getting more and more popular.

Users can create news or any private information or discussion entries at their computers and publish them in the web. To do so, they need to be near any pc that is connected to the internet. Over a web interface or a desktop application new blog entries can be composed. If something interesting or important happens while they are on the way, they probably won't be able to publish it instantely. Mobile blogging goes one step further. The idea is to enable the user to publish news whenever he wants and wherever he is.

Since existing mobile blogging systems do not fully cover this aspects, the idea to create the Mobile Reporter system was born. Our approach covers every aspect of normal desktop and mobile blogging.

In the year 2005 the worldwide sales of camera phones will reach nearly 300 million [3]. This fact will contribute to the popularity of mobile blogging. More and more people will use this technology to publish their informations and news or to stay in contact with friends and family members. The declining charges of the mobile network providers concerning data transmission will enforce this diffusion within a short time.

1.2. Goal of the work

The goal of the mobile reporter project was to create a system and publish it open source, that can be used easily by almost everyone who owns a webspace. Therefore it should fullfill some important basic requirements.

It should be easy to use. That means that every person with minimal computer knowledge should be able to install and use both. the server and the mobile application. This also includes easy modification of the system. Therefore every part was created as modifiable as possible. For example the server can be easily extended with new languages just by adding additional language files. More about modifications will be explained in chapter 5. Implementation and chapter 6. How To.

Small hard- and software requirements are just as important. Very few people would be able to install a system that needs the newest technologies. For that, some wide spreaded and well supported technologies were used.

1.3. Short description of the chapters

After introducing the aspects of blogging in common and mobile blogging the second chapter will describe the related work of the Mobile Reporter project. Some existing blogging and moblogging platforms are mentioned and their advantages and disadvantages will be explained in comparison to the Mobile Reporter project.

The third chapter explains the basic technologies that were used for the implementation. Several web technologies and also J2ME will be described shortly.

In the fourth chapter the basic architecture of the Mobile Reporter System is explained in detail. First the central component „Item“ and after that the different „senders“ as the different submission ways are described. Then a detailed overview of the overall architecture of the server as well as of the MIDlet is given.

The fifth chapter goes one step further and presents a detailed description of the implementation of the project. At the beginning the folder structure as well as the class structure is explained and illustrated by different pictures. Then the central class, the Item Parser and thereafter other important classes are described. Finally the MIDlet with all its classes and the relations with each other are explained. To modify any of the classes of the Mobile Reporter project it is recommended to read this chapter before.

The sixth chapter provides instructions how to install, administrate and use the Mobile Reporter server as well as the mobile application, the MIDlet.

The seventh chapter introduces the user study that was performed and emphasizes the results that were evaluated from the questionnaire.

In the eighth chapter different aspects of the publication of the Mobile Reporter project are described. It will be explained which licencse was chosen and where the project was published.

The ninth chapter provides a look in the future and describes the perspectives of the Mobile Reporter project.

2. Related Work

In times of wide spreaded Smartphones and PDA's the idea of Mobile Blogging as a special form of Blogging was born. As explained before, the increasing interest in Blogging and especially Mobile Blogging was the reason, that a few applications were developed. Many of them provide the functionality to publish new blog entries either online in a multifunction editor with simple tools to add images and links or even in Microsoft Word with an adequate add-on [4].

Some of them even allow the user to upload his blog entries with a mobile device running a specific software that can be used to create the new blog entry and finally to upload it to his existing blog.

The most famous related work is the blogging system Blogger.com [a] which provides an easy-to-use platform where everybody can create his own blog in just 5 minutes. After creating the blog you do not only have the possibility to create normal weblog-entries, but also a mobile blog by sending an MMS or an email containing text and/or multimedia files to go@blogger.com. The uploaded data is immediately put to a public weblog. After receiving a claim code on his phone by blogger.com the user can login and switch the post to his personal blog or edit it in a comfortable web form.

Unfortunately this mobile blogging service is at the moment only available in the USA for customers of the mobile network operators Verizon, AT&T, Cingular, Sprint or T-Mobile.

Another service provided by blogger.com is the Mail-to-Blogger function. The Mail-to-Blogger feature turns any email account into a blog-posting application. But the fact, that the user can‘t attach any kind of file to such an email is an enormous disadvantage.

Another popular platform for mobile blogging is flickr.com (“The best way to store, search, sort and share your photos”) [b].

Having uploaded their photos from camera phone or pc, registered users can then add comments notes and tags to their pictures to easily share them with friends.

To simplify this process of uploading pictures a small J2ME application for mobile devices called mobup was written [5]. It allows you to shoot photos and then it provides the flickr-specific functions to add a title, tags and a description to the photo and finally upload it.

One more mobile blogging system is called picostation [c]. Similar like all the other approaches the user can upload images, video- or audio-files to his own personal blog. The related application called PicoBlogger is available for most of the Nokia Series 60 phones and for some other smartphones.

All of these applications mentioned above allow their users to publish their comments and images.

But in most cases the process of uploading is limited to one or more specific submission ways or even to a specific cellular network. In some cases there are even restrictions concerning the files that can be attached to a post.

Entering a long text with a numeric keypad is quite exhausting. Therefore the main focus of mobile blogging is set to images. On some mobile blogging portal the user can add a title or a short description to the picture.

In all the mentioned systems the user has to register to a certain platform that has its own rules where he can‘t configure any of the settings. Furthermore these platforms don‘t provide a multi user blog where more than one user can publish to one single blog with other users. They only provide classical single user blogs where every registered user maintains a private blog.

2. Related Work

Additionaly the blogging platforms mentioned above are all hosted on a fix domain. That means you can‘t use them on your own webspace for you as a single user, but you have to share the blogging platform with other registered users.

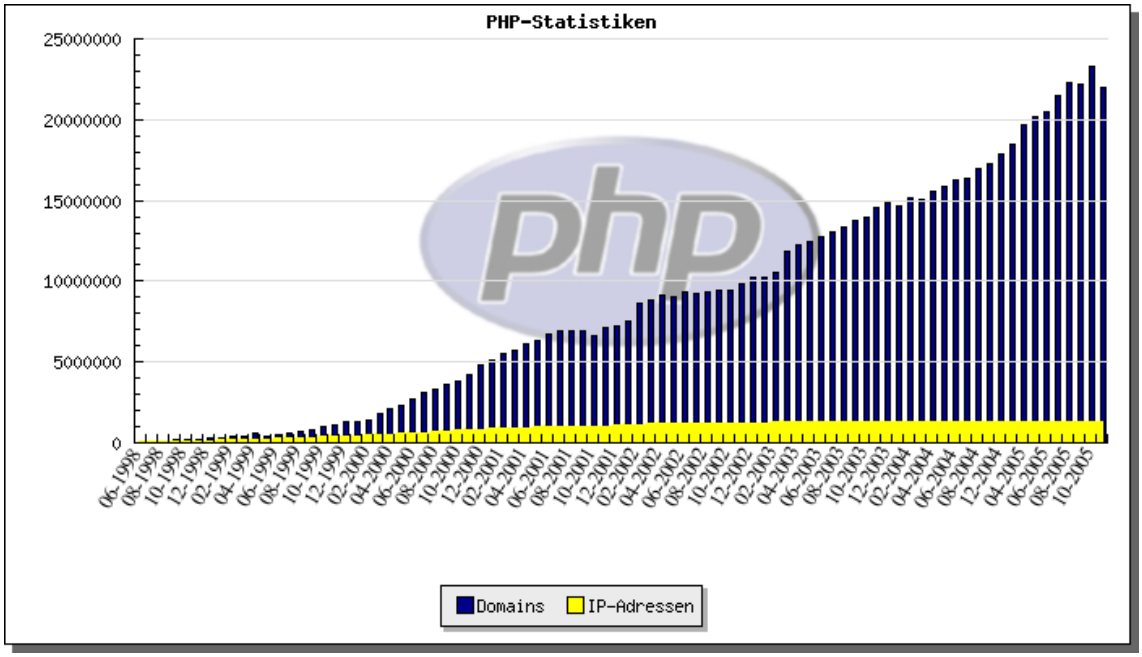
3. Technologies

This chapter describes the basic technologies that were used to create the mobile reporter system.

3.1. PHP 4

The programming language chosen for the server is PHP 4. Originally the newest version of the PHP product series PHP 5 was chosen. But at the end of the project the files were ported to the version 4 to reduce the requirements for the installation of the Mobile Reporter server. PHP is free of charge and supported by almost every web server. Picture 3.1 shows the large diffusion of PHP that proceeded over the years. As explained on the publishers webpage, „PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.“ [d]

Version 4 was chosen because it is the first one, that provides support for object oriented programming which is easier to use and maintain.



Picture 3.1: PHP statistics: The diffusion of PHP (source: http://www.dynamic-webpages.de/60.php-statistiken.php)

3.2. MySQL

All the data stored and used in the Mobile Reporter server is stored to a MySQL database. MySQL is an open source, wide spreaded Database Management System, that is free of charge. There are APIs available for Java, C, C++, Perl, PHP, Python and many more programming languages.

Just like PHP, MySQL is supported by numerous web hosts which is the main reason that it was chosen for the project. For correct use, MySQL 4.1.10. or higher is required. The official web site [e] can be used for further information and download.

3.3. Smarty

Since easy modification is one of the main goals of the Mobile Reporter project, the template engine Smarty [f] was used to create the HTML content. Using a template engine allows the strict separation of the application logic from the presentation. This not only saves time during the development process, because it eases the maintenance process, it also allows to create different views on the web sites without changing any source code.

Smarty is published under the GNU Lesser General Public License [6]. Therefore it can be included into any (non-) commercial product if it is not modified in any way and mentioned in the software's copyrights.

Smarty is included into the Mobile Reporter package and does not need to be installed by the user. Only template designers will have to busy themselves with the functionality of Smarty.

3.4. RSS

The news in the Mobile Reporter system are made available for others (for example websites) to use. Therefore News Feeds, also called RSS Feeds for every category are created. The process of making content available for other is called web syndication. [7]

„RSS is a Web content syndication format. Its name is an acronym for **R**eally **S**imple **S**yndication.“ [4] RSS is a XML dialect and can be parsed with any available XML parser. Almost every weblog uses RSS feeds. RSS readers are wide spread over the internet. Browsers like Firefox and Opera have preinstalled functions to process RSS content.

For the Mobile Reporter project the RSS version 2.0 was chosen which is described in [8].

3.5. J2ME

The mobile client application is written in Java 2 Platform, Micro Edition (J2ME) [g], an edition of the Java 2 platform, that is targeted at consumer electronics and embedded devices such as mobile phones and personal digital assistants.

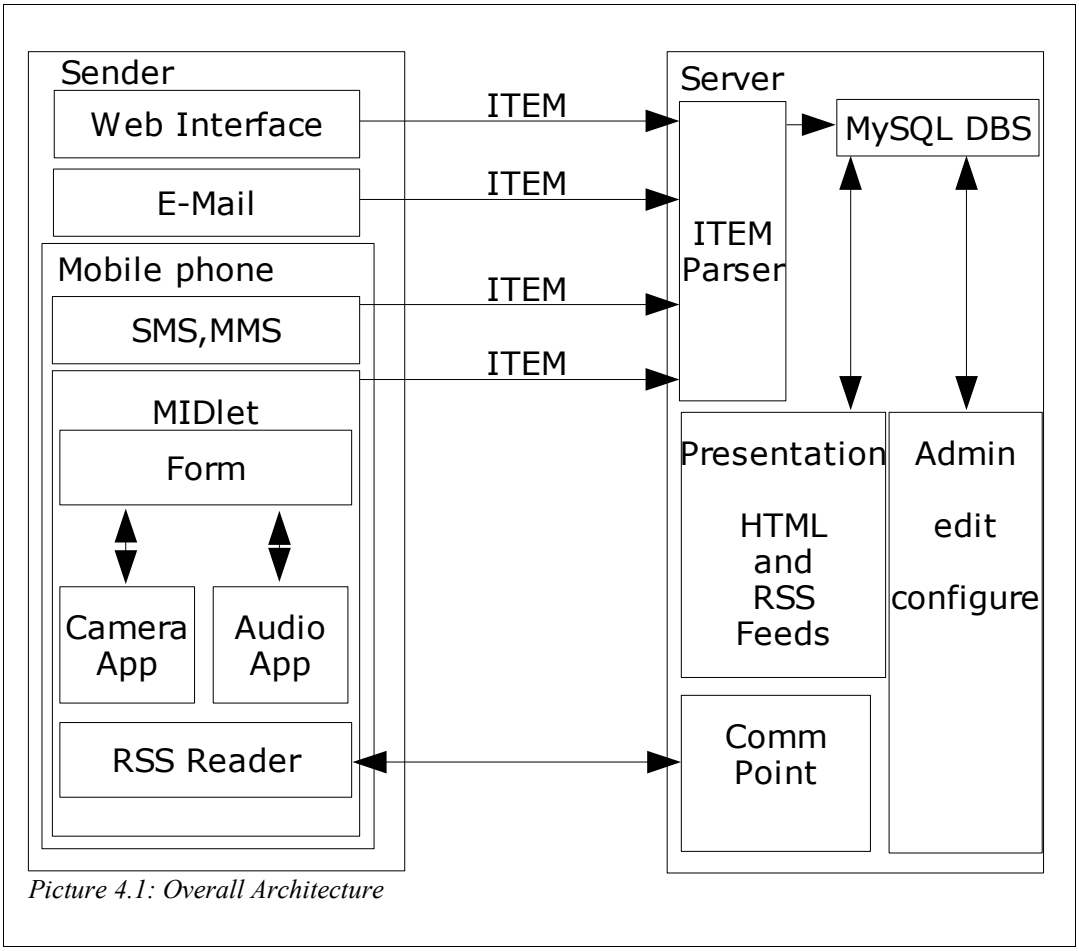
J2ME applications, also called MIDlets are supposed to work on any device that support the used configuration, profiles and additional packages. There is for example a profile called CLDC and another one called MIDP. To use the Mobile Reporter System, CLDC 1.0 or higher and MIDP 2.0 are required. Almost all new mobile phones fulfill these requirements.

4. Architecture

This chapters describes the architecture of the Mobile Reporter system. It consists of a number of components and techniques that will be described in detail. After that there will be a closer look at the server and the client.

4.1. Overall Architecture

Picture 4.1 describes the overall architecture of the Mobile Reporter Project. It covers the different sender and the transmission of the Items as well as some basic components of the server and how they interact with each other.



Picture 4.1: Overall Architecture

4.1.1. Server

Basically the server consists of five main components.

Most important the Item Parser, which is the interface for the Items sent in by any sender. The Item Parser's task is to find out how Items where sent and to recognize the single parts. For instance the part of the message, that represents the title or the Item text. It translates the Items to a specific format and saves them to the database. Files that are attached to an Item will be saved to the local file system.

The MySQL database contains all the data, that is needed for the server to work properly. It not only contains the Items, but also the userdata, categories, settings and much more.

This data is used by the presentation component to create the dynamic website and to provide the RSS feeds as well as by the admin component, which allows to edit any existing data and configure the whole system.

For direct communication with the server, it contains a component called comm point. This component answers the request to a RSS feed for a single category or the list of all categories by sending a specific XML format. This is for example used to implement the RSS reader in the MIDlet.

4.1.2. MIDlet

The mobile application that was developed for the Mobile Reporter system also consists of different components. The central component is called Item form where the user can edit and create Items. It can be used similar to a web form with different input fields. To add multimedia data, other components are used.

One of them is called camera application. This component is responsible for taking pictures and storing them on the mobile phone. It tries to connect to the mobile phones camera. If this is possible all features will be represented to the user. After saving the picture, they are available for use in the Item form. The same applies to the audio application which will record and save sounds.

Transmitting the Item is done by a small component called PostHTTP, which uses a simple HTTP Post request to connect to the server and submit the data.

The RSS Reader was already mentioned in chapter 3.4. because it is the component that communicates with the server to download and display the categories and new Items on the server on the mobile phone display.

4.2. Components

4.2.1. Item

At the beginning, a basic news data unit for the system was defined. It is called Item. An Item consists of many different parts of a news. It contains the title, a date, a place where the Item was written, the text and different media files like pictures and audio files. This is important because an Item is the kind of data, that needs to be transmitted to the server in many different ways.

The name Item is not just any arbitrary name, it is also the name of a news unit in an RSS feed, which is an important technique in the mobile reporter system.

4.2.2. Sender

Since mobility is an important aspect of the mobile reporter project, many different possible senders were checked to find out which of them should be supported. This is important because having many ways to submit Items increases the chance, that one of them is available at the users position. Therefore the analysis of the different possibilities to submit Items was a fundamental part at the beginning of the project.

Table 1 shows every sender that was analysed and the results.

4.2.2.1. SMS

SMS can be send from every mobile phone and are very cheap. The author of the Item can be easily identified by it's cell phone number and also the date can be easily identified as the time when the SMS is received. Using our Special Syntax, which will be introduced in chapter 6.1.3.2., the recognition of title, place and category can be achieved. The rest of the message, that is not marked with special syntax can be the text of the Item. But unfortunately SMS don't support any kind of multimedia data like photos, audio and videos. Another big disadvantage of SMS is the fact, that for using them, the user that administrates the server would need to install a SMS Gateway, which is very complicated to install. Therefore this was not implemented for the server because it would conflict with the goals simple to use and few requirements which are the main goals of the project.

4.2.2.2. MMS

MMS is a much more feasible technique to submit Items. It supports everything that is supported by SMS but additionally it supports also different kinds of multimedia files. It is possible to add one photo, audio file or video to it. Appending more then one file is not supported. Today almost every cell phone supports sending MMS and many people are used to this process.

The reason that receiving an MMS doesn't require any kind of Gateway is, that MMS can be send directly to an e-mail address. Therefore the decision was made, that sending an Item via MMS will be supported by the mobile reporter system.

4.2.2.3. E-Mail

Writing an e-mail got a normal part of everyday life and many different kind of terminals support it. Every PC that is connected to the internet, notebooks, handhelds, many mobile phones and other terminals offer the user the possibility to send and receive e-mails.

Not only because of the availability, but also for other reasons it was decided to implement the possibility to send Items via e-mail. There is for instance no limitation for the number of data that is attached to an e-mail. That means that the user can include as many text, pictures, audio files and videos as he wants to one Item.

The author can be identified by it's e-mail address and the title is the subject of the e-mail. Only for the place and the category of the Item the Special Syntax is needed. The date is the time, the e-mail was received by the mail server.

The implementation for e-mail and MMS support is almost identically, because an e-mail mailbox is used to receive them. The only difference is the recognition of the author. From the e-mail the senders e-mail address, for an MMS, its cell phone number is checked.

4.2.2.4. Web-Interface

A web interface is a form written in HTML. Almost every website that supports interaction by users uses forms to submit the data. Every needed part for an Item can easily be defined within such a form by adding an input field. The recognition of the date is very precise, because it is the time, the form was submited. Since the user has to be logged in to use the form, also the author data can be retrieved automatically.

Such a form can be used with every pc, that has access to the internet via a web browser like the Internet Explorer or Firefox. But also many mobile terminals like cell phones support using web forms via installed mobile browsers.

4.2.2.5. MIDlet

As mentioned in chapter 3.5., a mobile application is part of the mobile reporter system. The application is written in J2ME, the java version for small devices. Such an application is called MIDlet. Within a midlet, an application similar to a web interface can be implemented. The supported multimedia data depends on the cell phone. Actually there are none that support recording of video data. But audio files and pictures are supported by almost all camera phones.

The advantages of the midlet in contrast to MMS and e-mail is the better interface for the user. He doesn't need to know anything about the Special Syntax. The advantage to the web interface is that no browser and no connection to the internet are needed to create an Item.

The interface that is used to send the data via the midlet can be used by every application that supports http to send in Items. This means every application that can request a http connection is able to send in Items with this interface.

4.2.2.6. Result

At the end of the analysis it was decided to choose MMS, e-mail, a web interface and a midlet as senders.

	SMS	MMS	E-Mail	Web-Interface	MIDlet
Date	(XXX)	(XXX)	(XXX)	XXX	XXX
Title	X	X	XX	XX	XX
Author	XXX	XXX	XX	XXX	XXX
Text	XX	XX	XX	XX	XX
Place	X	X	X	XX	XX
Picture	-	XX	XX	XX	XX
Pictures	-	-	XX	XX	XX
Category	X	X	X	XX	XX
Audio file	-	XX	XX	XX	XX
Audio files	-	-	XX	XX	XX
Video	-	XX	XX	XX	-
Videos	-	-	XX	XX	-

X	Only with Special Syntax
XX	Easily possible
XXX	Automatic recognition
(XXX)	Maybe not exact automatic recognition
-	Not possible

Table 1: Possible Sender

5. Implementation

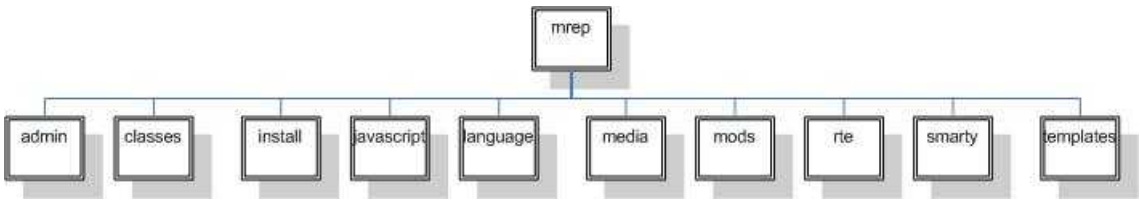
5.1. Server

5.1.1. Structure

This chapter introduces the structure of the Mobile Reporter server. It describes the folder structure to understand where the single parts of the server are located and the class structure to understand how the different classes inherit from each other.

5.1.1.1. Folder Structure

If the server is correctly installed, there will be some PHP files in the main folder as well as some subfolders.



Picture 5.1: Folder Structure

Main Folder

The PHP files contained in the main folder are for different purposes. Most of them can be directly accessed by a browser to display different web pages and functionality like adding an Item to the server or displaying categories and Items. But some of them have special functions which will be explained now.

- The file **config.php** contains the variables that are needed to access the database as well as the absolute path to the project and it's url.
- **Commpoint.php** is the file that can be accessed for direct communication with the server as mentioned in chapter 4.1.1. It represents the biggest part of the comm point component from picture 4.1.
- **Feed.php** is a displayable file. Depending on the GET parameter „id“ it creates a RSS feed for a specific category.
- **Upload.php** is an interface that can be used by any application to add Items in a specific format, using HTTP Post. For example the Mobile Reporter MIDlet submits its Items using the file upload.php.

Admin Folder

The admin folder contains files similar to the normal files of the main folder. They provide functionality to view and edit data to configure the Mobile Reporter server. For example the file options.php is used to display and edit the basic settings of the server.

In difference to the main folder, all files of the admin folder can only be accessed by users of the system having admin rights.

Classes Folder

The files in this folder are all PHP 4 class files that are used by different parts of the server. There is an inheritance between them which will be described in chapter 5.1.1.2.

Install Folder

For installing the server the files in this folder are used. It contains the installation script, the functions it uses and different kinds of SQL data. For example all the relations for the SQL database are saved in the file mysql_relations.sql.php written in SQL DML, the data modelling language. Also the templates representing the HTML web sites of the installaiton routine are saved in this folder.

After the installation this folder can be deleted, because non of it's data is used in any other part of the system.

Javascript Folder

The Javascript parts of the server are stored in this folder. The most important parts that are written in Javascript are the rich text editor, that is used for creating an Item using the web form and the tooltips that are displayed in various parts of the server.

Language Folder

As described in chapter 1.2. one of the main goals of the project was to create a system that is highly modifiable in easy ways. One part of this is the extending of the server with additional languages. The languages are stored in this folder. Adding new languages is done by copying them to the folder.

Media Folder

At first the media folder will be empty. It is filled with the files that are attached to the Items or such files that are stored temporarely while creating an Item with the web form. Temporary files are stored to the subfolder temp followed by the ID of the user. Saved files are stored in the folder „media/cat<category ID>/item<Item ID>“

Example: An user with id 5 adds a file a.jpg to the Item. It will be stored to the folder „media/temp/5/“. After saving the Item, for example the ID 4 is assigned to it and it is saved in the category with the ID 8. That means the file a.jpg will be saved to the folder „media/cat8/item4/“.

Mods Folder

This folder is supposed to be the location for additional files, mostly data files, that will be needed by any modifications for the Mobile Reporter server. For the standard installation there will be only one sub folder called „email_tmpl“ that contains the template files for the e-mail feature that was created to ease the configuration of the e-mail data during the installation and in the admin configuration menu. The reason why the „email_tmpl“ folder is located in the mods folder is that this feature was created almost at the end of the project and didn't fit anywhere else in the file structure.

rte Folder

There are several files that are opened by the Richt Text Editor to display different functionalities like inserting tables or images to the text. These files are saved in the rte Folder.

Smarty Folder

The Smarty folder contains all files of the Smarty template engine, that is used to seperate the application logic from the presentation. It's the original release version 2.6.7 of the Smarty project and hasn't been modified in any way.

Templates Folder

Just like extending the server with new languages it is also possible to extend the server with new templates to change the layout of the visible parts of the server. Every subfolder that contains a template that is valid to Smarty and to the Mobile Reporter Project will be recognized and can be chosen in the options.

5.1.1.2. Class Structure

Since the Mobile Reporter server is written in PHP 4, the object oriented programming abilities were used. Therefore a big number of classes for different purposes are located in the project folders.

Mainly, there are two different types of classes. Such, that have invisible functionality like reading something from the database and others that also have presentation purposes like showing a web page.

The functionality only classes encapsulate tasks that are used by various other classes and scripts. Every action that refers to a specific component of the project is located in one class. For example every action that can be done on an Item, like saving, reading it from the database, deleting it, are located in the class ItemManager.

The kinds that show websites are located in the main and in the admin folder. The ones in the main folder extend the class DisplayView, which is located in the folder classes. Those in the admin folder extend the class AdminView, also located in the folder classes. This is so, because there are many tasks that need to be done to show a website that do not differ from website to website, no matter what the website is about to display. These common functionality is located in those two upper classes and can be summarized in this way:

- **Objects**

  They create objects that are needed by every class that displays a website.

- **Security**

  Depending on the status (logged in/off) and the rank (admin, user ...) of the user different functionalities are displayed or denied.

- **Errors/Notice**

  Error Messages and other Notifications are always displayed in the same way. Therefore all the functionality needed for this is located in the class DisplayView and AdminView. Child classes only have to call their messages.

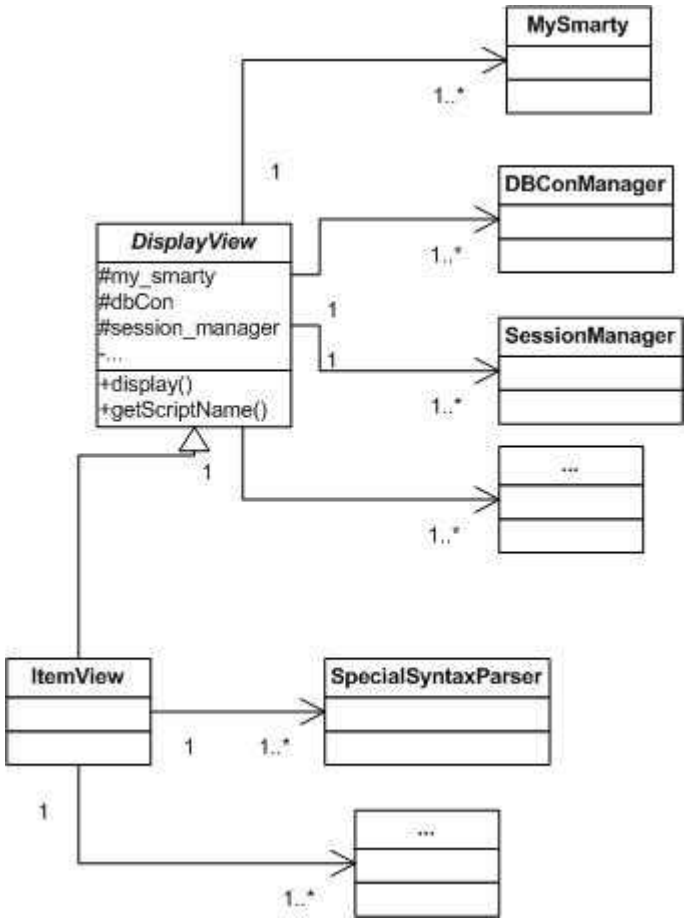
- **Header**

  Also the whole menu structure and the fast admin is created by these classes.

AdminView and DisplayView have almost the same functionality, but do also differ a lot. That

is why both are needed for the project. For example a different header is created for admin website and also security isues are not equal.

Picture 5.2 shows an UML diagram for the class ItemView which is used to show Items on the website. This structure is similar for all visible classes on the server.



Picture 5.2: Example UML structure: ItemView

5.1.2. Item Parser

The Item Parser is the central component on the Mobile Reporter Server for parsing incoming Items. You find the Item Parser in the folder classes in the file ItemParser.class.php. Every incoming Item gets parsed by this class, no matter if it was sent from the MIDlet, via email, over the web interface or via MMS. This way of a central interface allows interested people to easily add their own new submission ways. As input the Item Parser needs an array with some data like user, timestamp, title, text, location and category. Incoming Items pass the parseItem method and afterwards if they contain any media data the saveMediaData method.

In the first step the parseItem method tries to identify the user that was just uploading this Item. In case of the MIDlet or the web form this is a quite easy process, because the user is logged in to write an Item. The way of allocating the right user to an Item sent via email or MMS is more complex. If the Item was sent in an email the Item Parser verifies if the sender‘s email address is registered in the database. If it is so, the Item gets assigned to the registered user, if not the Item will be rejected for security reasons. Since at the most mobile network operators the sender of

an MMS is identified by something like <phone number>@<network operator> the Item Parser tries to parse the phone number of the sender. If the adequate number is found in the database the Item gets assigned to the user who specified exactly this number when he registered at Mobile Reporter.

The next step is to get the timestamp when the Item was written. For the web form and the MIDlet it is possible to take the actual time on the Mobile Reporter server, when the Item is received, because this is quasi the time it was created. Concerning the timestamp when writing an email or a MMS there can occur some impreciseness, because the timestamp when the Item was sent can differ from the timestamp where the Item was received on the e-mail server. But in practice this can be neglected. The timestamp that the Item Parser gets from the web form or the MIDlet should be more precisely.

After that the Item text gets processed by the Item Parser class.If special syntax was used in the text, the Special Syntax Parser will be used to translate the text. The detailed way how this happens will be described in the next chapter. If there is no occurence of special syntax the text gets stored in the database after converting all the containing HTML tags to their adequate entities. For security reasons all HTML tags are finally removed out of the Item text before saving it to the database.

Afterwards the Item Parser tries to find the title of the Item. As in the three steps above the way of getting the title depends on the chosen submission type. If the Item was sent via email then the subject of the email is supposed to be the title. Concerning the web form or the MIDlet the title can be extracted easily from the form where the user explicitly specifies the title. For setting a title in a MMS the user is forced to use the special syntax, because there‘s no other possibility to extract the title out of the MMS.

Each Item is stored in a category. If the user didn‘t declare a category the default category, which can be set in the options is applied. The web form provides the easiest way of choosing a category by offering a pull-down menu with all the categories available. Unlike the web form the email and the MMS need to contain special syntax if the Item should be stored in another category than the default category. Indeed there‘s a field for the category in the MIDlet, but for reasons of costs the complete list of categories will not be downloaded from the Mobile Reporter server. That means the user should know either the exact name or the category ID of the respective category. The task of the Item Parser is finally to check if there‘s a category in the incoming Item. If there is one, this category will be assigned to the Item, if none or a non existing category is specified then the Item will be stored in the default category of the system.

Finally in the parseItem method the Item Parser tries to get the location from where the Item has been sent. In practice it is not possible to find out the location automatically. For that it is necessary that the user fills in the location field in the web form or the MIDlet. Choosing MMS or email as submission way the user is supposed to use special syntax if he wants that a location is published with an Item.

The last step in the parseItem method is to finally store the incoming Item in the database. Here the Item Parser has to differentiate between new incoming Items and Items that already exist and should now be changed. If the incoming array contains a field with an existing ItemID then the Item Parser executes a SQL update query in the database which overwrites the existing Item with the new data. Else a new Item will be created by inserting the extracted data to the database.

Thereafter the saveMediaData method is called in case there‘s any media data attached to the incoming Item. In the next procedure the relevant directories will be created in the server‘s file system to store the attached media files. If this is done the saveFile method is called successive with every file that was sent together with the Item as parameter. This method stores each media in the right path on the server and if GDLib is available it tries to create a thumbnail of the

uploaded file provided that the file is an image. At the end the path to the media file, the filename and the mediaID are stored in the database and if marked a main image for display in the Item preview will be selected.

5.1.3. Special Syntax Parser

As explained before the Special Syntax Parser is responsible for translating the special syntax that is used to display links or attached images in the Item text into HTML tags that can be interpreted by browsers. The source code of this class can be viewed in the file SpecialSyntaxParser.class.php which is stored in the classes folder.

The Special Syntax Parser class contains two important methods called parseStaticSpecialSyntax and parseRuntimeSpecialSyntax which both expect a string as parameter. The parseStatic method is called when the incoming Item gets processed in the Item Parser for saving the Item in the database. The method tries to find the Mobile Reporter Special Syntax constructs with regular expressions. If found it decides if the object represents the title, the location or the category of the Item. For example if the following token [t:My News Item] is contained in the Item text then the Special Syntax Parser knows that he has to interpret all that follows the colon until the squared bracket close as the title of the Item (A detailed description how to use the Mobile Reporter Special Syntax can be found in the How To's in chapter 6.1.3.2.).

The so extracted title will be saved in the database and the construct with the Special Syntax is removed out of the Item text. Similar to this the location and the category can be detected if the Special Syntax is used. You should always keep in mind, that the Special Syntax is only necessary where the input form doesn't allow a separate field for data like location or category as given in a MMS or email. So the task of the parseStaticSpecialSyntax method is to extract the additional data from the Item text whereas the parseRuntimeSpecialSyntax method gets called every time this Item should be displayed. That means that every time a user visits the website with an Item on it, the parseRuntimeSpecialSyntax method is called to display the relevant images and links. More precisely the parseRuntimeSpecialSyntax method searches the Item text for a Special Syntax construct with an image or a link in it ([i:image.jpg] or [a:www.link.com]). In the next step it replaces this token with the aproppriate HTML tag. In case of an image the token [i:image.jpg] gets translated to the HTML tag <img src='<path>/image.jpg' /> provided that this image exists on the server's file system. The token for the link gets translated in a very similar manner. It will be replaced by the HTML tag www.link.com.

5.1.4. Ranking of Items

Acting as a multi user blog, the Mobile Reporter system can quickly be flooded with news Items that contain useless or not relevant articles. To avoid this situation the automatic ranking process for the Item overview on the Mobile Reporter home page was developed.

In that way depending on different aspects all the Items are ranked in a certain order. These aspects comprise for example the number of Item views for the specific Item, whereas an Item view from a registered user is ranked higher than an Item view from a guest on the website. Additionally the rating of Items is also reflected in the assessment as well as the number of comments that was composed to a certain Item. The more positive ratings and comments the higher is the ranking factor of an Item. The Items with negative ratings and no or only a few comments get a low ranking factor.

The implementation of the ranking algorithm can be found in the files index.php and in the ItemManager Class (/classes/ItemManager.class.php) that provides the relevant methods to rank the Items. In the index.php there are 3 factors item_view_user_factor, item_view_visitor_factor and comment_count_factor. These factors allow to define the composition of the total ranking factor.

If the administrator has activated the automatic Item ranking in the settings the home page will be composed of the latest Item and the top 3 of Items, i.e. the Items with the 3 highest total ranking factor. These Items are determined by the method getMostViewedItems() and if comments and rating is allowed by the methods addItemCommentCount() and addRatingCount() in the ItemManager class.

For reasons of performance the determination of the most viewed Items is done directly in the database query. The query returns a field total_factor in the result of the query that specifies how often an Item was viewed. The comment count and the rating are then taken into the computation of the total_factor in the index.php. This is done by adding the number of comments multiplied with the comment_count_factor to the total_factor. The rating factor is then added to the total factor depending on the number of positive and negative ratings for this Item.

Finally the method getTop3Items() orders all the Items by the total_factor. In this way the three top Items are determined and in the method createDisplayItems() the Item data is prepared for the display by replacing the Special Syntax with the relevant tags and by setting the desired date format and more.

5.1.5. Configurables

Most parts of the Mobile Reporter system are not static but highly modifiable. User can change almost every aspect of the website as well as create new languages or templates.

5.1.5.1. Templates

The HTML code is not directly integrated in the source code but a template mechanism was used to create web pages out of template files. This is not only done to seperate the application logic from the presentation, which is very important in a big project, but also to enable any user to create different look and feels for the website.

The system is very simple. The folder „templates“ cotnains all the available templates for the server. They consist of a number of „.tpl“ files and one file called „temp.desc.php“ that contains information about the template like its name (see text 5.1).

```
<?php
 $template_name = "Example Template 1.0";
 $template_author = "My Name";
 $template_description = "Some template.";
?>
```

Text 5.1: temp.desc.php Example

The „.tpl“ files consist of a mix of HTML, XML and the Smarty language, which is described in detail on the official website [f].

To generate a new template the whole folder should be copied and the single files edited. The only thing that has to be retained is the file structure and the specific Smarty language. The HTML part can be modified at will.

There is a function „listTemplates“, located in the file „/classes/FolderManager.class.php“, that checks the folder „templates“ for any compatible template. These will be displayed and choosable from the administration menu.

5.1.5.2. Languages

Just like the HTML code, also the language is not hard coded into the source code, but read from so called language packages. Those packages are located in the „/language“ folder and each of them consists of five files.

„lang.desc.php“ which includes important information about the language like its ISO encoding and four files that contain the language ordered by their purpose. For the main website „lang_error.php“ inlcudes the error messages and „lang_main.php“ all other langauge data. The phrases for the installation routine are saved to the file „lang_install.php“ and for the administration area they are located at „lang_admin.php“. Text 5.2 shows a small part of the english version of the „lang_main.php“.

```
// Standard form data
 $lang_main["firstname_label"] = "Firstname";
 $lang_main["lastname_label"] = "Lastname";
 $lang_main["email_label"] = "E-Mail";
 $lang_main["mobile_label"] = "Cellphone Number";
 $lang_main["username_label"] = "Username";
```

Text 5.2: Part of lang_main.php

Thus adding a new language to the system is very easy. Any of the existing language packages (folders) can be copied and reused. Splitting the languages into four files eases maintanance and enables users to selectively translate the whole system to another language. One could for example translate only the parts of the language that are visible to all the users and leave the administration part untranslated.

The function to get all available languages is also located in the file „/classes/FolderManager.class.php“ and is called „listLanguages“.

5.1.5.3. Others

Besides templates and languages there are some other parts of the server that can be easily added and modified. For example the large number of options allows extensive changes that not only affect the behaviour but also the look and feel of the server.

Another example is the email template tool, that is used to provide easy configuration of freemail accounts. The folder „/mods/email_tmpl“ includes the files that can be used to auomatically configure the e-mail settings.

5.1.6. Small Screen Version

As mention before, for the presentation part of the server, a template engine was used that creates HTML code. The server was build to support adding additional templates to the system

and one of those is already integrated in the install version of the server.

The integrated template was made in xHTML, thus it can be viewed even with browsers for mobile phones. Creators of additional templates are encouraged to do the same, because it means, that almost every functionality will be available as well on mobile phones.

Many creators of mobile phone browser introduce concepts to display websites that were build for desktop pc browsers in a good way for their users. One of these concepts is the Small Screen Rendering (SSR) from the Opera Software ASA [9]. The Mobile Reporter websites are very good looking with SSR (see Pictures 5.1 and 5.2) just because they where written in proper xHTML.

Abbildung 5.1: Website in Small Screen Rendering

Abbildung 5.2: Item Creator in Small Screen Rendering

5.1.7. Administration

The administration area is an important component of the Mobile Reporter server. It is implemented in the way that first of all only the user who installed the system can access it if he's logged in to the website. The design is kept simple and clear to ease the administrator's work. As the admin interface itself is realized with templates everyone can easily adapt the design and achieve his own idea. The administration interface can be found in the admin folder.

The implementation of this area can be dissected in three parts. First there are the options that

handle with different ranks a user can have. Thus it's possible to show all existing ranks or to choose a rank and to edit the individual attributes of this rank. The program logic of this section can be found in the file ranks.php and the presentation in the folder templates in the files rank_form.tpl, rank_list.tpl and rank_edit.tpl. Please keep in mind that the administration area has an own templates folder with all the templates for the display of the administration interface in it. It's more clearly arranged and easier to maintain if the admin area and the frontend are separated and use their own folder structure.

The second part is the section that has to do with the administration of users. Here all the users get fetched out of the database and will be shown in a list where they can be deleted or their rank can be edited. Additionally the administrator has the possibility to add a new user over a web form, what is reasonable if the option to allow new users to register is set off. The relevant files for the user administration containing the program logic are show_users.php, adduser.php and edit_user.php. The templates show_users.tpl or edit_user.tpl are responsible for displaying these functions.

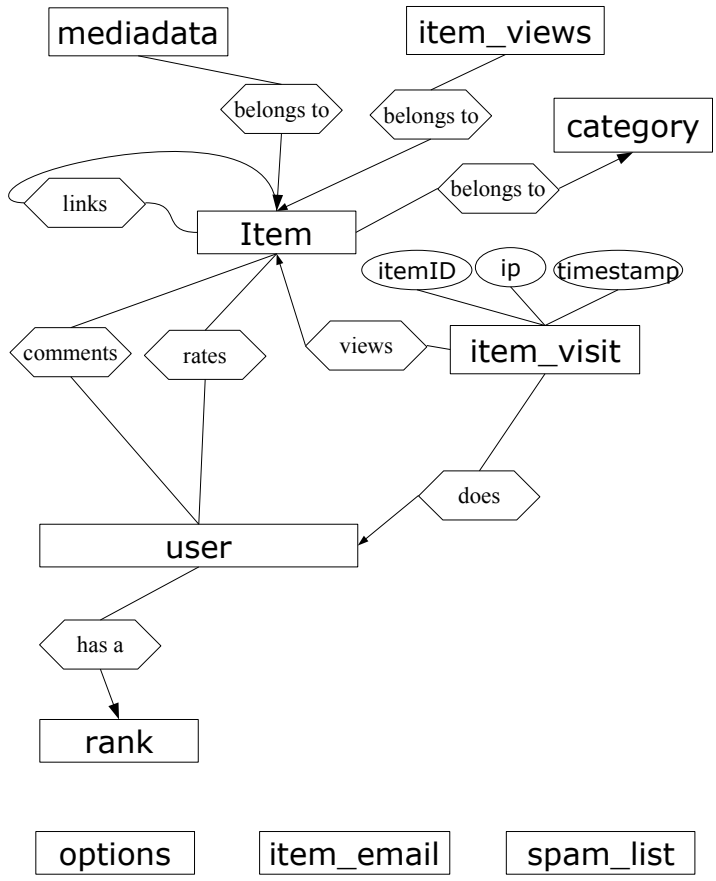
In the main part of the administration interface there are the general options like the language that is set or the template or the default category and more. All the options are fetched from the database and afterwards they will be displayed and can be edited by the aministrator. If he finally submits the changes all the settings become active. More option forms were implemented to edit the email data from the email address where the users of the system can send their Items to or to edit the data to access the database or to define a spam list of email addresses from which incoming Items should be rejected. For this area the program logic can be found at the files options.php, dbdata.php, email.php and spamlist.php. The adequate display as HTML is realized in the following templates: options.tpl, dbdata.tpl, email.tpl and spamlist.tpl

All of these administration websites require the input of the administrator to specify the settings and the functionality of the Mobile Reporter system. Basically the input of these forms is sent via HTTP POST and is then handled by the calling php script to store the data in the database if all the necessary fields were filled out.

5.1.8. Database

As already mentioned above a MySQL database was chosen to store and manage all the data that is needed for the whole Mobile Reporter system. In the next chapters the detailed structure of the tables and the relation between the tables will be described.

5.1.8.1. Database Structure



Picture 5.3: Entity Relationship Model

Picture 5.3 shows the Entity Relationship Model of the database that was created for the Mobile Reporter project during planning phase.

The central entity in the database is without a doubt the Item. The Item table contains all the relevant data that describes an Item. For example the item_timestamp, the title, the userID from the author of the Item, the category ID, the location, the item_text, a main_img flag that specifies the image that is shown in the Item preview, the creation time, which was important for the user study, and a flag if the Item is visible or not. This is necessary for archiving the Item.

Mediadata can be attached to an Item. In the relation mediadata there are the attributes media ID, filename, url, which describes the url to the media file on the server and the Item ID to what the Item belongs to (foreign key).

Each Item is stored in a category that has a unique category ID, a specific name and an attribute rss_items_count that defines the number of Items which are available in the RSS feed of this category.

A User can create, rate or comment an Item. The table user contains all the data from the user registration like the username, firstname, lastname, email, mobilephone number, password, the adequate rank ID and the unique user ID to identify the user. Users have ranks to realize different rights what they are allowed to do or not to do.

The table ranks is determined by the single rights as attributes like add_item, edit_own_items, edit_all_items, delete_own_items, delete_all_items, add_category, edit_category, delete_category, delete_all_comments, delete_own_comments and add_comment, that are all

relatively self explaining. Furthermore it contains the attribute rank_visible which permits to hide the rank of the admin and the attribute is_admin, which specifies if the user of this rank is an admin. Each rank has additionally a rank_name and a rank ID for referencing a rank with a user by setting the adequate rank ID to the data of the user.

For reasons of ranking and statistics each click on an Item by a user is counted. The entity item_view is responsible for storing the relevant data. It contains the attribute user_count which counts all the visits of registered users on this Item, the attribute visitor_count which counts all the visits of unregistered_users and the attributes day, month and year to include the aspect of the time.

To prevent users to click their own Items for a good item_view statistic there's a security aspect integrated that an Item view is counted after a configurable period of time by the same IP address. For that the entity item_visit with the attributes ip_address and visit_time and the unique identifier Item ID was created.

The three remaining entities that are all responsible for configuring the Mobile Reporter system in common sense have no relations with other entities. The table options for example contains all the data to manage the main functionality and all the settings of the server. The attributes are language, template, date_format, split_item_date_format, media_formats_allowed, max_media_size, max_files, max_text_size, html_allowed, comments_allowed, rating_allowed, registration_allowed, max_items_per_page, seconds_to_recount, email_check_on, seconds_to_email_check, last_email_check, default_cat, thumb_w, thumb_h, ranking_on, and rss_items_count.

The table item_email is needed to store the data that is relevant to establish a connection to the email account for fetching the mails if set in the options and translating them to Items. For this process the table item_email with its attributes email, server, port, protocol and password provides the required information.

As a reason of the enormous flood of spam emails it is possible to define some mail addresses from which the received email will be rejected automatically instead of parsing the spam mail and converting it to an Item. The structure for this table is kept simple. It only contains the attributes email and an unique identifier.

5.1.8.2. config.php

The file config.php is used to store the database and the project path data that was filled by the administrator during installation of the Mobile Reporter server. After the successful installation the file config.php will be written automatically on the server's file system. It contains the data that is needed to establish the connection to the database like the database host, the database username, the database password, the name of the database and the database prefix that can be specified to avoid duplicate entries concerning the table names in the database. Furthermore the location data like the project url, which is the explicit web address, where the users can reach the Mobile Reporter system as well as the absolute project path on the server. That means the path from the root folder to the folder where the Mobile Reporter system is installed on the Webserver.

5.1.8.3. Options

The options for the Mobile Reporter system provide many different settings to configure the server platform. All the options are stored in the MySQL database in the table options. Thus everybody easily has the possibility to add new options by inserting a field in the options table. For using these settings the superclass DisplayView provides a class variable as an array called options that contains all the options from the database as fields in it. By inheriting from this

super class every other class has access to this array to request the desired options by using the following construct: $this->options[“<option name>“]. For example the approach to experience if the option „HTML allowed“ is set, could be:

```
if($this->options[“html_allowed“] == 1)
```

Binary option values are stored as tinyint in the database whereas the value 1 means true and the value 0 means false.

The basical options like language and template will be checked in the super class DisplayView. This way assures that on every page that inherits from this class the language and the template specified in the options are used. The other options will be checked separately if they are needed in any class. In this manner it can be ensured, that for example only new users can registrate to the system if the administrator has the option registration_allowed set to true.

To add a new option three steps are necessary.

The first step is to integrate the new option in the database in the table option with the adequate attribute type and if required a default value. It's important to choose a clear and precise denotation for this option.

In the next step the new option should be added in the admin area for that the administrator of the website can configure this option.To do this, the files options.php for the programming logic and options.tpl for the display should be modified. They can be found in the admin folder.

Finally the newly integrated option can be applied to the frontend of the Mobile Reporter system. To check the newly created option use the programming construct from above and just replace the name of the option with the identifier of the newly created option.

5.1.9. Item Editor

The Item Editor provides a comfortable way to create or edit Items on the Mobile Reporter website. It consists of the three form fields title, location and category as well as of the main textarea with all its functions and finally of the file upload box at the bottom of the page.

For the implementation of the main textarea the Cross-Browser Rich Text Editor (RTE) by Kevin Roth [k] was modified. The RTE requires a browser like Internet Explorer version 5 or higher, Mozilla 1.3 or higher, Mozilla Firebird/Firefox version 0.6.3 or higher or Netscape 7.1 or higher to function properly. Thus every browser that supports the designMode() functionality (introduced in IE 5) is able to display the RTE. Every other browser will display a normal textarea that can be used equivalent but without all the convenient functions of the RTE.

Since the code is public domain it is permitted by the author to use this code with or without modification. To fit the RTE into the Item Editor several changes had to be done. For example to realize the mutilingual concept of the Mobile Reporter it was necessary to store the labels for the Item Editor in the specific language files and not „hardcoded“ in the source code of the RTE.

As the RTE is written in Javascript the source code of the RTE can be found in the folder javascript in the file richtext.js. Modifications concerning the whole Item Editor can be done in the file additem.php which is located in the main folder where the Mobile Reporter was installed or in the template file item_form.tpl which is located in the templates folder. In this template the richtext.js file is included and the RTE can be displayed by calling the javascript function writeRichText with several arguments.

As the file additem.php is used to create a new as well as to edit an existing Item, the class has to differentiate between those two modes. If the Item Editor should be opened to edit an existing Item the file additem.php is called with an Item ID specified as GET parameter. Therefore the

showForm() method checks first of all if an Item ID is set. If yes it proceeds with setting the adequate form fields and getting the relevant data for this Item out of the database. Already attached files will also appear in the file upload box. If the user attaches new files here they will be immediately stored in the right path contrary to the upload of files attached to new Items. If the user hits the create button after editing the changes all the Item data will be put to an array. Then the parseItem() method of the ItemParser class is called with the item_data array as parameter. The ItemParser (described in 5.1.2.) saves all the data by updating the row in the database with the adequate Item ID that was set as GET parameter.

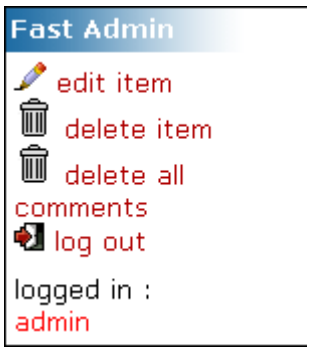
If there's no Item ID set then a new Item will be composed. Thus all the form fields are empty. Attached files will be first stored to the temp folder in the media folder. This is necessary because the correct path is among other things composed of the category ID and the Item ID of the Item where this files are attached to. But these two ID's are not yet available, because the Item is not yet created. If the user finally creates the Item then all the files will be moved to the correct path. All the data like the attached files, their path, and the title, the category and more will be sent to the Item Parser as described above. The ItemParser finally stores the data in the database.

5.1.10. Fast Admin

To make the usage of the Mobile Reporter server faster and easier, a feature called „Fast Admin“ has been developed. The Fast Admin is attached to every part of the user interface. It changes its content depending on the actual displayed part of the website and the rank of the user, which indicates his rights. Picture 5.4 shows the Fast Admin for the category overview and picture 5.5 shows it for the Item view with displayed comments.

Because the Fast Admin is needed for every page it is created through the function createFastAdmin() in the class DisplayView. To find out the rights of a user, this method uses an instance of the class RankManager, which includes functions made for this purpose. For example the method editItemAllowed() returns true if the current user is allowed to edit the Item that is currently displayed on the screen.

Picture 5.4: Fast Admin, Category Overview



Picture 5.5: Fast Admin, Items View with Comments

5.2. MIDlet

This chapter covers the implementation of the mobile phone application, the MIDlet. It introduces the most important classes and explains their connection with each other.

5.2.1. Performance Tuning

Performance tuning is an important aspect during the development of J2ME applications.

Certain aspects need to be considered, that help to speed up programs and save memory. There is a couple of guides that introduce performance tuning for MIDlets like [TODO] written by Jonathan Knudsen.

The main problem for mobile devices is the lack of memory space. Therefore developers are often confronted wit java.lang.OutOfMemoryError. This happens if to much objects are created. Mostly to much Strings or String concatenations are responsible for this. An optimization would be to use StringBuffers instead of Strings or even character arrays.

Saving memory can also be done by beeing clean. This means for example closing any kind of stream after they are not needed any more. Even such simple things as using arrays instead of Vectors or Hashtables saves not only memory but also processing time.

There are many ways to increase the speed of J2ME applications. Exemplary optimization of loops should be done. After [10] loops often contain methods that have to be called every time the loop is called. Text 5.3 shows a typical loop as explained in [10] before and after optimization.

```
Before:
for (int i = 0; i < v.size(); i++) {
Object o = v.elementAt(i);
// Process the Object o.
}


after:
int size = v.size();
for (int i = 0; i < size; i++) {
Object o = v.elementAt(i);
// Process the Object o.
}
```

Text 5.3: Loop Optimization

This is only an excerpt of things that can be done to improve the performace of applications on mobile devices but it shows the importance of programming carefully. As far as possible, these guidlines were considered while the developement of the mobile reporter midlet.

5.2.2. Manifest.mf and Configuration

After building a MIDlet, all needed files are compressed to a „.jar“ file. The MIDlets name, its Item graphic location, the author's name and a lot of other meta information are stored in a file called „MANIFEST.MF“ which is located in the folder „META-INF“. This file can be used to add aditional attributes to a MIDlet which can be used in the application. Every line of the file represents one attribute. To get the value of such an attribute, the method `getAppProperty(String property)` which is located in the class MIDlet has to be called. If an attribute represented by the method's parameter is not defined then this method will return null.

For the Mobile Reporter MIDlet, this mechanism was used to make it configurable for users without any programming experience. This means someone who is running a Mobile Reporter server can change the configuration of the MIDlet that is available for his users, just the way he

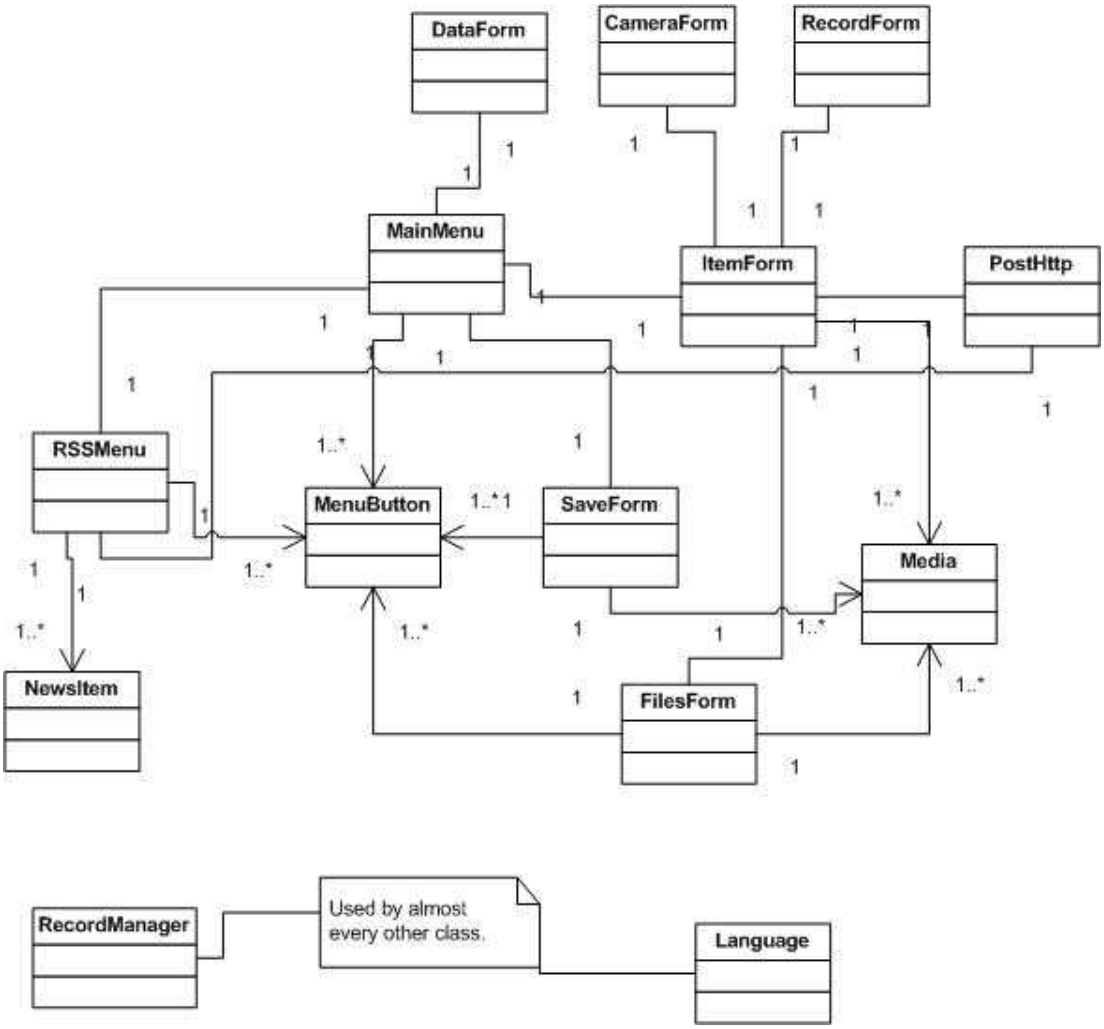
wants with only little effort.

There are exactly four attributes that can be changed, or added if not already in the file, to configure the application.

- „MY-URL“ is the url where the Mobile Reporter server is installed. If this attribute is missing or empty users will have to fill in the url at the first start up of the application. Having this attribute in the file will ease the usage for the end users.
- „MY-RSS-ON“ can only have two different values. „0“ (zero) which means the RSS feature will be disabled in the application or „1“ (one) which menas it will be enabled.
- MY-MAX-FILES can be either „-1“ or any number. The value indicates how many files a user can attach to an Item. If it is „-1“, an endless number of files can be attached to each Item.
- MY-PHP-EXT is only needed if the server application is running on a web server that demands another file extension for PHP 4 files then „.php“. If the server uses for example PHP files ending with „.php4“, then this line should look like „MY-PHP-EXT: php4“.

5.2.3. Classes

The MIDlet consist of a number of classes which interact together. In contrast to non mobile applications certain performance issues have to be considered as explained in chapter 4.2.1. Since to much objects can cause memory problems, presentation and application logic were not seperated to different classes but kept together.



Picture 5.6: UML MIDlet

5.2.3.1. MainMenu

This class is called immediatly after starting the MIDlet. Like all other classes in the project that include visible and interactive parts, it extends the MIDP class Form and implements the interfaces CommandListener and ItemCommandListener which are needed to capture key events done by the user.

The main menu represents a graphical interface to access all the main features of the MIDlet. As seen in picture 5.6 it is a central component, that allows access to the DataForm, the ItemForm, the RSSMenu and the SaveForm class that include important functionalities for the user. Some of these may not be available depending on the configuration of the application (see chapter 5.2.2.). For example if the RSS feature is disabled in the „MANIFEST.MF“, then the button for starting the RSS reader will not be available in the main menu.

5.2.3.2. ItemForm

The class ItemForm is the most important component of the MIDlet because is is the central component for creating Items and actually that's what the MIDlet is supposed to do mainly. There are two ways to create an instance of ItemForm by calling the constructor as displayed in Text 5.4.

```
ItemForm(MainMenu mainMenu, Display display, String item)
```

Text 5.4: Constructor ItemForm

The important parameter, marked red in text 5.4 is the String Item, because it can be used to load saved Items by using the unique ID of a saved Item. This will be explained in chapter 4.2.4. If the parameter is null, this means the creation of a new Item will be started and a new ID for it will be created. The parameter mainMenu contains a reference to the main menu.

If a new instance of this class is created, a variable called maxFiles is beeing created depending on the configuration as explained in chapter 5.2.2. This variable stands for the max number of files that can be attached to an Item. If it is null, specific components of the user interface will not be available to the user. To be precise, this means that the options „add photo“ „record audio“ and „show attachments“ won't be displayed in the menu. This is a big restriction for the users but could be wished by a server admin to save for example bandwidth.

The class ItemForm also extends the MIDP class Form which is used for graphical representations like displaying option menus. In the case of this class such an option menu exists but will change during the process of creating an Item. For example as long as there aren't any files attached to an Item, the option menu won't contain the option „show attachements“ because there are none. This decision wheter to display this option or not is done in the method showMe which is calling the static method RecordManager.countMediaRS to get the number of attached files for an Item.

By choosing the option „shoot photo“ or „record audio“ instances of the class CameraForm respectively RecordForm are being created and displayed and a reference to the actual ItemForm is passed to them.

To save Items either if a user is choosing this from the option menu or automatically if an error occured during the upload the static method RecordManager.saveItem is called. This saves an Item and all of its information to the record store.

An inner class of ItemForm is called UploadForm. If the user chooses to upload an Item, an instance of this inner class is created. An UploadForm uses the class PostHTTP to create HTTP Post transmission to the server to transfer all the data that belongs to an Item. To avoid loosing data during the transmission process, the Item is split into multiple parts, that are transmitted one after the other. For every part, an instance of PostHTTP is needed. At first, the title, location, category and text are transmitted. The optional parameters „time_needed“, used to count the time a user needed to write an Item, and „invisible“, which can be used to set an Item not visible until every part of it was uploade, must not be send. The server processes these information and answer with a new unique ID for the Item. After this, all the files, attached to the Item are send together with the ID. This ensures, that they will be matched to the right Item located at the server. For every transmission, the username and password will be send to authorize the user. If an error occurs or the user data is wrong, the server will send an error message which is displayed on the screen.

5.2.3.3. CameraForm / RecordForm

The two classes CameraForm and RecordForm are used by the ItemForm to display menus which allow to record audio with the mobile phone's build in microphone respectively shoot photos with the mobile phone's camera. These features won't work if the mobile phone doesn't support the parts of the multimedia API that are needed for it. For example CameraForm will only work on mobile phones that have an integrated camera and support creating player objects for it.

To gain control over either the camera or the microphone, player objects for them have to be created. On these player objects the method getControl can be used to get a control object for the current player. Text 5.5 and text 5.6 show how this is done for capturing images and recording audio.

After the user shot the photo or recorded any audio, an instance of the inner class NameChooser is created which displays a dialog to enter a name for the file. After entering a valid name with at least one character, the static method RecordManager.addMedia is used to save this file permanently to the MIDlets record store. IF the filename already exists, it will be changed by adding a number as long as there is no other file with the same name. This is also done by the method RecordManager.addMedia.

```
player = Manager.createPlayer("capture://video");
...
videoControl = (VideoControl)player.getControl("VideoControl");
```

Text 5.5: Getting Control over the Camera

```
player = Manager.createPlayer("capture://audio");
...
rc = (RecordControl)player.getControl("RecordControl");
```

Text 5.6: Getting Control over the Microphone

5.2.3.4. RSSMenu

Accessable only from the main menu, the class RSSMenu is responsible for communication with the server to download and display all available categories and Items. This class has several inner classes, that download and display specific kinds of information. Therefore they all extend the MIDP class Form. The inner class CategoryMenu download and displays all categories, the class SingleCatMenu does the same for all Items from a specified category and the class ShowItemMenu only displays an Item. Downloading an Item in this last step is not necessary, because all information regarding an Item is already downloaded for displaying a single category which is using a RSS feed to do so.

Communication is done sending a Post request to the script „commpoint.php“ which is located at the Mobile Reporter server. To do so, the class RSSMenu creates an instance of the class PostHTTP.

5.2.3.5. PostHTTP

The class PostHTTP which was mentioned already before is a non visual class that is used by instances of RSSMenu and ItemForm to submit data to the Mobile Reporter server and receive answers from it.

Post requests are part of the HTTP specification and can be received by every server. Processing data , sen wit a Post request is very simple using web programming languages like PHP or J2EE. That's why this request format was chosen for communication with the server.

Using this class is really simple. For every transmission, an instance of it must be created by calling the constructor PostHttp(String url). The parameter „url“ is any URL a connection should be established to. In case of uploading an Item, this URL will be the Mobile Reporter

server address combined with the script „upload.php“. In the constructor the host part of the url will be automatically extracted because it can be usefull for post requests. To finally submit the Post, one of the methods submit respectively submitData has to be called. The return value of this methods is the anser string of the server. Each of this two submission functions was implemented to fit specific requirements.

To explain how they differ from each other the basic idea of HTTP requests will be introduced. For HTTP requests header fields are used to tell the other communication partner, what kind of data will be transmitted and how to read them. Header fields are also used for many other information of a connection. In J2ME the class HttpConnection can be used to set up such a connection and set the headers for it. To define a header, the method setRequestProperty of the HttpConnection class is used. There are basically two different kinds of Post requests that can be identified by the „Content-Type“ header field. To send only string parameters, „application/x-www-form-urlencoded“ will be chosen, if also binary data like image files will be send „multipart/form-data; boundary=----------7d01ecf406a6“ is chosen, where boundary can be any string. Because the boundary will be used to differ different parameters and data from each other, this kind of a request creates an overhead. That's why „application/x-www-form-urlencoded“ is used to send requests that don't contain any binary data.

This is exactly the difference between the two methods. The method submit is used to for parameter only requests, whereas the method submitData is used to transfer parameters and binary data together.

Besides this two methods, there are only two other public methods. The method setParameter has only effect when using submit. It is used to add any and as much parameters to the request. The second, setData. is used to add the file that will transmitted with the requests and will only have effect if using submitData.

5.2.3.6. SaveForm

If there is at minimum one saved Item in the record store, then the main menu will contain a button to create an instance of the class SaveForm. It's constructor requires an instance of the classes MainMenu and Display. By calling the constructor, the private class createMenu will be called. This method checks by calling RecordManager.getSavedItemsCount if any Item exists in the record store. If there is none, the main menu will be shown immediately. This can occur, if an Item has been deleted using the save form. If there are saved Items, they are loaded by calling RecordManager.getSavedItems and for each of them a button will be displayed. By pressing such a button, two options are beeing displayed. An Item can be deleted or shown. Showing an Item also means having the possibility to edit and upload it. As mentioned in 5.2.3.2. the constructor of ItemForm allows to show an Item by entering a valid name. But in the case of this class the method mainMenu.showNewItemForm is used for this purpose. Actually this method will call the mentioned constructor of ItemForm but it does one more things. It changes the menu structure of the main menu which is needed to show to the user that an Item has been loaded, even if he is in the main menu.

If an Item has to be deleted, then an instance of the inner class DeleteConfirmForm is created. This will show a form that enables the user to confirm or abort the delete process. If it is not confirmed, the save form will be shown again, but if it is confirmed the method RecordManager.deleteItem is called which will delete the Item. If there are other Items, the save form will be shown, if there aren't the main menu will be displayed. As mentioned, this is decided in the method createMenu.

5.2.3.7. FilesForm

The class FileForm has similar functionality like the class SaveForm. The main difference is that it doesn't display saved Items, but files that were save for an Item. If there are files available for an Item, then the Item form will display an option called „show attachements“ which creates in instance of FilesForm.

The constructor which requrires an instance of ItemForm as a parameter, will call a method called createMenu which creates a button for every file. Depending of the type of the file which can by an image or an audio file, different options will be available for every file. For images, an option „show“ and for audio files the option „play“ will be displayed but for both the options „add to text“ and „delete“ will be available.

Choosing the option delete will create an instance of the inner class DeleteConfirmForm which will display a dialog to conform or abort the deletion of the specific file similar to the dialog displayed for Items. Confirming it will call the method RecordManager.deleteMediaByName which deletes any chosen file.

The option „add to text“ enables a user to add preformated special synta , as explained in chapter 4.1.3., to the text field of the Item form by calling the method addToText of the class ItemForm. This will also display the Item form again.

For image files, the option „play“ creates an instance of the inner class AudioPlayer that displays a form that can be used to play the specific audio file. This class implements the interface PlayerListener which requires to implement the method playerUpdate that can be used to track state of the player. This state can be for example started, stopped and end of media. The class contains also two functions to start respectively stop the playing of the audio file. To load the audio file the method RecordManager.getDataByName is called.

The last option „show“, which is only available for image files, creates an instance of the inner class ImageViewer that jsut loads the specific image by calling RecordManager.getDataByName and displays it on the screen.

5.2.3.8. Language

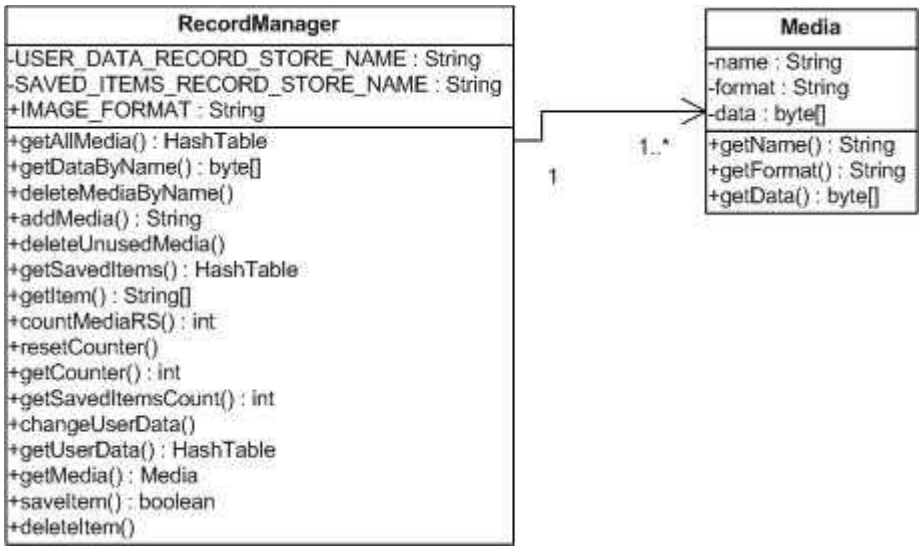
As mentioned in the server implementation chapter, no language was hard coded to the source code. This was also done for the MIDlet. Therefore the class Language was implemented. It contains only static string variables that have public accessibility as shown in the text 5.7. If there is need for some text somewhere in the source code, then the class Language is used. For instance the variable RSS_READER_BUTTON would be used like Language. RSS_READER_BUTTON. To convert the MIDlet to another language, this file has to be replaced and after that it has to be compiled again.

```
public static final String RSS_READER_BUTTON = "RSS Reader";
public static final String EDIT_DATA_BUTTON = "Edit Data";
public static final String DATA_FORM_TITLE = "Enter Data";
public static final String USER_NAME_LABEL = "User Name";
public static final String USER_PASSWORD_LABEL = "Password";
```

Text 5.7: Excerpt from the Class Language

5.2.3.9. RecordManager

This class is very important because it is the only one that directly accesses record stores to add, change and delete them. This means that every class that needs to do anything with the record store has to use the RecordManager for this purpose. All methods and variables contained in this class are static, which means no instance of this class has to be created to use it. This increases efficiency on mobile phones because no additional objects consum memory. For accessing record stores, the J2ME class RecordStore is needed which provides several basic method to work on record stores.



Picture 5.7: Class Diagram RecordManager

Picture 5.7 shows the class diagram for this class. It includes three static String variables. USER_DATA_RECORD_STORE_NAME and SAVED_ITEMS_RECORD_STORE_NAME contain the names of the record stores for the user data respectively the record store that contains the saved Items. They are private because the only class that needs to access them is the RecordManager itself. The only public variable IMAGE_FORMAT is used to predefine the image format that is used for the photos, that will be recorded with the mobile phone. It is important that this one is public because it is used in other classes as well. For example the ItemForm needs to know this format to upload images to the server.

As can be seen in picture 5.7, there are is a big amount of methods for different purposes. Most of them are self explaining like getItem or deleteItem, but there are some that need to be explained. The two methods getCounter and resetCounter are meant to create unique IDs for Items on the mobile phone. These IDs are used to connect media files with the Item they

belong to. Such an ID is internally also called „name“ or „counter“. Everytime a new Item is created, the method RecordManager.getCounter is called by the ItemForm class. This method reads the actual integer value of the counter and returns it. It also increases the counter by one and resaves it to the record store. If there is no more Item saved to the MIDlet, which can happen after deleting the last one for example, the method resetCounter is called. It sets the counter in the record store to zero to prevent an overflow of the integer value.

Every method that returns media data like images or audio files uses the class Media for this. An instance of this class contains the name, the format, the data of the file as a byte array and the methods to get them. For example the method getMedia returns exactly one media object.

The method getMedia is a good example on how to access a record store and read from it. Text 5.8 shows this method. To open a record store, the method RecordStore.openRecordStore is called. The second parameter defines if a record store with the specified name should be created if it does not exist. If erverything went well and the record store is opened, the method getRecord can be called on it. In this example a record store consits of three records whereof the second and the third are needed to create the media object. At the end, the newly created media object is returned. It is important to know, that records consist of byte arrays. This means they have to be converted to their original value.

```
public static Media getMedia(String name) {
      RecordStore rs = null;
      try {
         rs = RecordStore.openRecordStore(name,false);
         byte[] data = rs.getRecord(2);
         String itemCount = new String(data);
         Media media = new Media(name,
                  new String(rs.getRecord(3)),data);
         rs.closeRecordStore();
         return media;
      } catch (RecordStoreNotFoundException e) {
         ...
      } catch (RecordStoreException e) {
         ...
      }
      return null;
   }
```

Text 5.8: Method getMedia

The organization of the record stores for the MIDlet are explained in the next chapter.

5.2.4. Record Store Usage

Mainly for security reasons, MIDlets underlie certain limitations. Some new mobile phones support access to the mobile phone's local file system, but most don't. Instead of that, MIDlets

have some reserved space on the hard drive. These so called record stores are the only way to save data on the mobile phone. A record store is a unit that contains records which consist of byte data. The class RecordStore, that is part of the midp profile of J2ME, provides all functionality to create, delete and change record stores.

Every operation that needs to load data from or save data to the record store is done by the RecordManager class. It provides a couple of methods for this purpose. All of this methods are static which means no object of this class needs to be generated to access to them. For mobile applications, static methods have the advantage, that they save working memory.

For the Mobile Reporter MIDlet, there are a couple of things that need to be stored.

The user data is stored in a record store called „Userdata“, which includes the user name and its password, used to validate the user with the server. It also contains the url of the homepage that is used to connect to the server to uplaod Items and download RSS feeds. The last thing saved to this record store is the counter. This is a simple integer value, that is used to identify Items saved in the record store. If a new Items is beeing created then its id will be the actual counter value and the counter will be increased by 1.

As mentioned before, Items can be saved to edit or send them at another point in time. If an Item is saved, its data is stored to the „SavedItems“ record store. For each Item, five records are added to the record store. Name, title, place, category and text whereof name is the unique counter value that identifies the Item. That means the number of saved Items equals the number of records divided by 5. If there are for example 10 records in the record store, this means that 2 Items are stored in it.

At last also the media files like images or audio have to be stored. For every media file there will be its own record store. The record store is named by the name of the file. This is possible because two files can't have the same name. Each of these media record stores has 3 records. The first is called „Item name“ and contains the unique ID of the Item, the file belongs to. The second is the byte data of the file. The last is a String value for the format, which is used to recognize if the file is an audio or a video file.

Picture 5.8 shows these different record stores.

	Userdata
1	username
2	password
3	url
4	counter

	SavedItems
1	name
2	title
3	place
4	category
5	text
	...

	$media
1	Item name
2	bytes
3	format

*

Picture 5.8: MIDlet Record Stores

6. How To

The following chapters show you how to install, use and administrate the Mobile Reporter server and mobile application (MIDlet).

6.1. Server

This chapter teaches you everything that you need to know about the use of the Mobile Reporter server.

6.1.1. Installation

6.1.1.1. Quick Installation

If you have experience using FTP and PHP scripts and little knowledge about servers, you can just use these short steps to install Mobile Reporter. Otherwise just skip the quick installation and jump to the chapter 6.1.1.2. for a detailed explanation.

1. Decompress the Mobile Reporter zip archive to any directory.
2. If your server or web space requires another file extension for PHP 4 files then „.php“ you have to rename some of the files and change the extension in the file extension.inc. For a detailed description see the next chapter.
3. Upload all the files extracted to any directory on your server or web hosting account. You may not change the directory structure.
4. Set the following permisions:
   - config.php: 666.
   - /media/: 777
   - /media/temp/: 777
   - /smarty/templates_c/: 777
5. Now use your browser to navigate to the directory where you installed Mobile Reporter + „/install/install.php“ to start the installaton routine.
6. Select a language.
7. Fill out the required fields.
8. Select a start configuration for the server.
9. Delete the install folder which you can find in the main folder of Mobile Reporter.

6.1.1.2. Installation

6.1.1.2.1 Decompressing

After downloading Mobile Reporter you should have a file on your hard drive called mrep.zip or similar. Decompress this file for example using winzip [h] to any folder you like.

6.1.1.2.2 Changing File Extensions (Optional)

This step is only needed if your web server or webspace requires another PHP 4 file extension then „.php“.

Open the file „extension.inc“ from the main folder with any text editor you like and replace the value „php“ from the line „$php_extension = "php";“ with the required file extension.

Example: If your server requires PHP 4 files to be named with „.php4“ then change the line to „$php_extension = "php4";“.

Some of the files in the Mobile Reporter folder have to be renamed now. You have to replace the file extension „.php“ of these files with your extension. Only rename the files ending with „.php“ from the following list.

- Main Folder: all files except for config.php
  - /admin: all files
  - /install: install.php

6.1.1.2.3 Upload

Upload all the files and directories to your web server. Be sure to absolutely retain the file structure and all file names as they are. Image files should be uploaded in BINARY mode, all other files in ASCII mode. Usually your FTP software will do this automatically for you.

6.1.1.2.4 Permissions

Setting permissions is very important not only for security reasons. Permissions can for example be set using a FTP client like SmartFTP [i]. Just right click on the file or directory and choose permissions. A menu similar to picture 6.1 will be displayed.

Only the permissions for the file config.php and the folder „/media/“, „/media/temp/“ and the folder /smarty/templates_c/ have to be changed. For the file config.php writable for all (666) like in picture 6.1 has to be set. For the folder „/media/“ and „/media/temp/“ as well as for the folder „/smarty/templates_c/“ write and read for all (777) is needed.

Picture 6.1: Permissions using SmartFTP

6.1.1.2.5 Installation Routine

Now use your browser to visit the location where you installed and add the path „install/install.php“ to it. For example if you uploaded it to http://www.yourhost.com/mrep/ then the path should look like this http://www.yourhost.com/mrep/install/install.php. The installation routine will be started.

At first you will be asked to select a language, the websites will be shown in. Depending on the numbers of languages installed with the Mobile Reporter server, different numbers of languages will be choosable. If for example an english and german language pack are installed, the menu will look like in picture 6.2. Choose any language and press the button.

Picture 6.2: Languages

The second form consists of various input fields that you need to fill out. Depending on your server configuration, some of them may already have been filled out for you. These data are categorized and not all of them are required.

Database Data

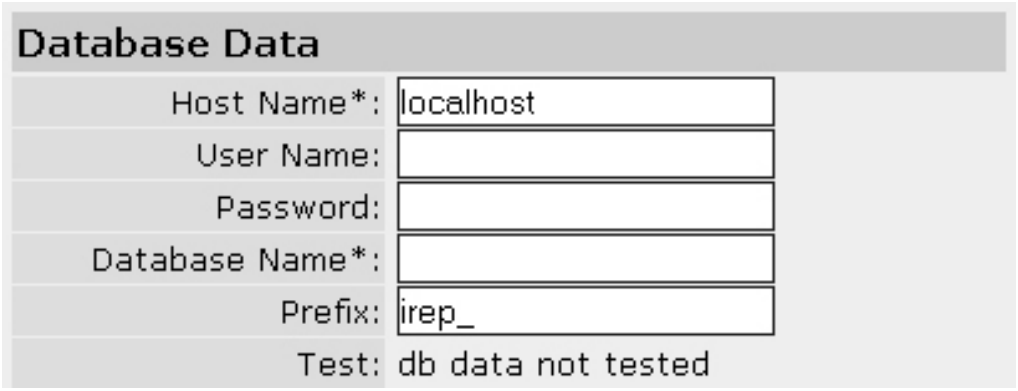
As mentioned before, the Mobile Reporter server needs a MySQL database to run properly. In the first part of the form, you need to enter the data that will be used to access this database. This part is shown in picture 6.3.

In the first field, enter the host name, which is the server, where the database is running. If you are using a web hosting account, this is normally „localhost“ but it can be any IP-address or URL. If your database is not running on the default port, add it to the host name. For example: the host name is localhost but your port is 8080, then the entry should be „localhost:8080“.

In the fields „user name“ and „password“ you need to enter the data of a user, that has access to the database. If you don't know them, contact your web hosting provider. It is possible, that the user has no password. Many MySQL databases have a root user without a password, which can be a security risk.

In the last required field enter the name of the database. This information should also be provided by your web host.

The last field, which is labeled „Prefix“ can be left empty. The prefix, which can consist of several characters is only important if there are other applications besides Mobile Reporter, which are using this database. The prefix prevents errors that can occur if a database is shared between different applications. The standard prefix „irep_“ should always be fine.



Picture 6.3: Database Data

Location Data

The next category is the location data of picture 6.4. It consists of two fields. The „Absolute Path“ and the „Project Url“. The installation routine could have filled in the fields automatically. But depending on the configuration of the server your website is running on, this automatic filling could be skipped or wrong. So don't forget to check them.

The absolute path is the path starting at your root directory and ending with the name of the directory you uplaoded Mobile Reporter to. If you don't know the absolute path please contact your web hosting provider.

The project url consits of your domain and the path to the folder where you uploaded Mobile Reporter. An example: If you uploaded Mobile Reporter to http://www.yourhost.com/mrep/, then this has to be entered to the input field. The string „http://“ must be attached to the url.

Picture 6.4: Location Data

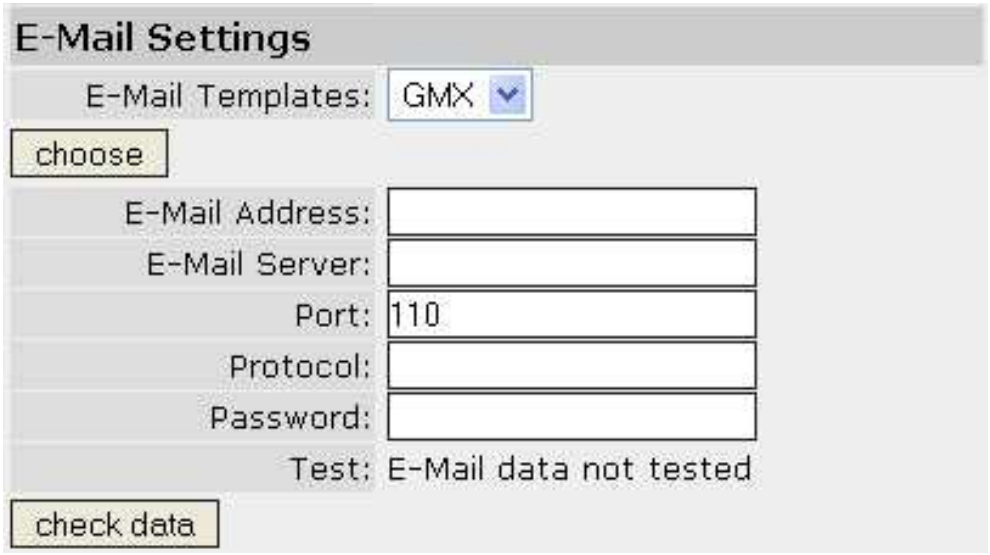
E-Mail Data

The e-mail settings category is not required. You can specify an e-mail address and its data to check a mailbox for new mails. The Mobile Reporter system supports sending Items in the e-mail format. This can be done by using any e-mail client or for example sending an MMS to this e-mail address. If you want these features to be enabled, then you have to fill out these fields.

The first field is a little tool that might help you to enter your settings. Templates for different freemail providers are available. Select the e-mail provider out of the list and press the choose button. The fields server, port and protocol will be filled out. If your provider isn't in the list, you have to specify every data by yourself. If you don't know them contact your provider read its manuals.

In the field „E-Mail Address“ just enter your exact e-mail address and it's password in the field „Password“. The „E-Mail Server“ field need to be filled out with the address of the e-mail

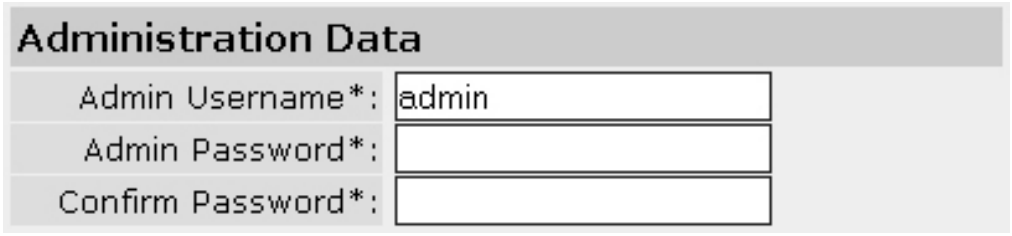
server. You will also have to enter the port, the e-mail server is running on. For pop3 servers the default value is 110. The last field „Protocol“ is either pop3 or imap.



Picture 6.5: E-Mail Settings

Administration Data

The last thing you have to fill out is the administration data. This will create the first user account for the Mobile Reporter server with full admin rights. Just choose any name and password you like.After finishing the installation you can specify or change your data when you are logged in.



Picture 6.6: Administration Data

Check Data

You can't press the „create database“ button as long as the entered data is not checked. After the e-mail data fields there is a button labelled „check data“. Press it and the data will be validated. If the check is passed, the install button will be clickable. If not, the errors will be displayed and you need to change the data. If only the E-Mail data check fails, installation can progress because these data are not required.

If everything is fine press „create database“ to save all the settings.

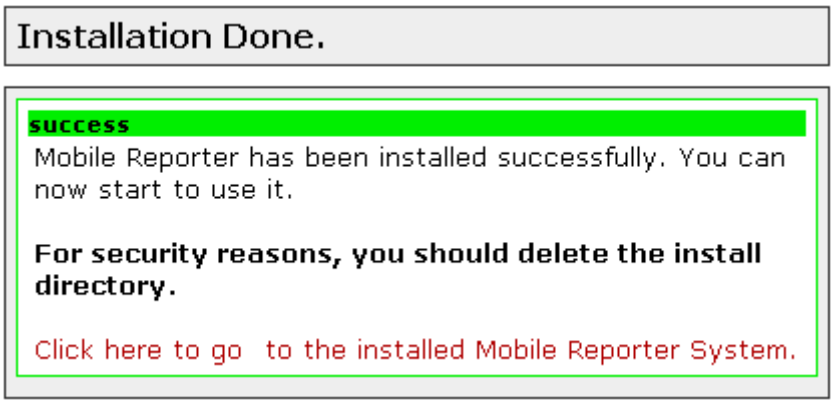
Choose Configuration

The Mobile Reporter server provides at least two start configurations. The last form let's you choose between them. The preinstalled configurations are:

- Multi User News Blog which supports unlimited users but has restrictions in the file size of the uploaded data and more.
- And Private News Blog which doesn't support registration of multiple users but provides more liberty for the user.

The configurations are just build in to ease the first start. By changing the options manually in the admin menu, you can define any configuration you like.

If you finished all the steps above, the Mobile Reporter server should be installed. If you see the message like in picture 6.7 you can be sure that the Mobile Reporter server was installed successfully and is running properly.



Picture 6.7: Installation Done

Additional Security Steps

After the installation you can increase the security of the server by the following step:

- Delete the install folder. You can find the Install folder in the main folder where you installed Mobile Reporter. This will prevent other persons from changing any settings.

6.1.2. Administration

This chapter should give an introduction to administrate the Mobile Reporter system. To administrate the server you should login with the account you specified at the installation. At the beginning only this account has the rights to enter the administration area. If necessary you can give other users the admin rank to grant access for him.

6.1.2.1. Options

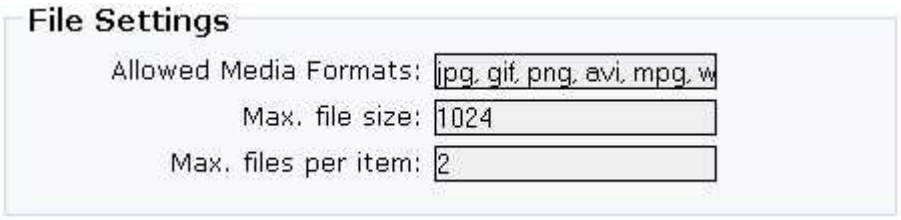
When you enter the administration area you will be directed to the basic options page. Except for some expert options all the other options should be quite self explaining. For the sake of completeness even the self explaining options will be described shortly:

Standard Settings

Picture 6.8: Standard Settings

The standard settings allow you to configure the basic behavior of the system. For example the language and the template as well as the default category can be set. The default category is the category that will be chosen if there's no category specified in an Item or if the specified category is not existing. The number in brackets behind the category name declares the number of Items that exist in this category. The next option Items per page displayed defines how many Items are displayed at one overview page. The last point of the standard settings is the setting registration allowed. As you may think this setting decides whether new users are allowed to register or not.

File Settings



Picture 6.9: File Settings

In this section you can define settings that concern the multimedia files attached to an Item. With a comma separated list you have the possibility to explicitly allow some file formats. If the user wants to upload a multimedia file format that isn't specified here, then the file will be rejected and the error will be reported to the user. To expand the list above simply go to the last entry and add a comma immediately after it. The comma should be followed by a space and then the desired file extension. For security reasons you should never add the „exe“ file extension, because that can cause users with malicious intentions to do damage to the webserver where the Mobile Reporter system is running. Of course you can also remove a file extension from the list, but please make sure, that each file extension is separated by a comma and a space.

The option maximum file size describes how large the file size of an attached multimedia file may be at most. The unit for this number is kilobyte. In the picture above like in the settings from beginning the maximum file size is set to 1024 kb which corresponds to one megabyte. Please notice that each webserver at its own has a restriction for uploading file sizes called upload_max_filesize (with root access to the server it can be defined in the php.ini file) which is set to 2 mb by default, but it depends on your web hoster.

In the field maximum files per Item you can enter a limit how much files can be attached to one single Item at most.

Item Settings

Picture 6.10: Item Settings

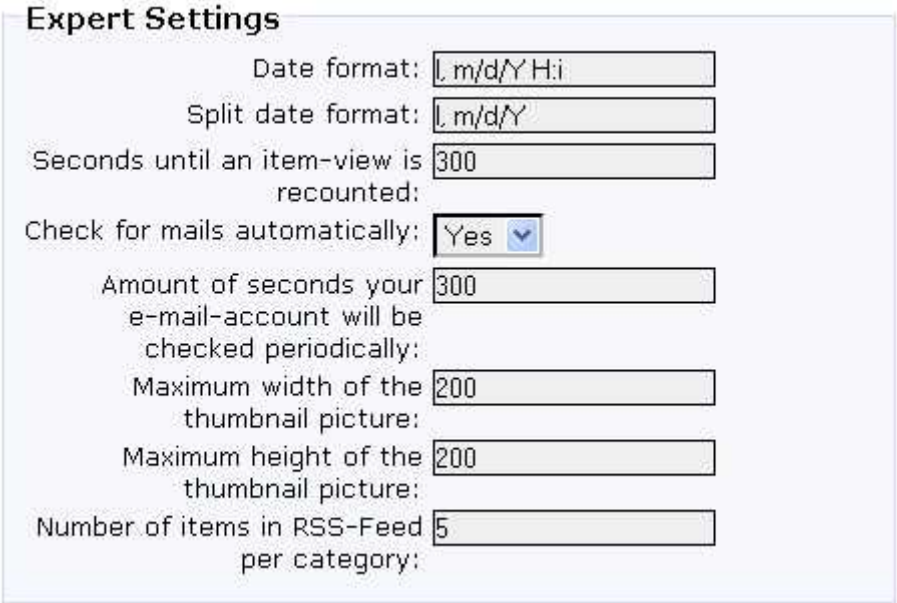
In this box you can define settings that immediately affect the Item itself. For instance here you can decide wether users can use HTML code in their Item text or not. From the technical view this option should not be active, because if you allow users to use HTLM code then they can integrate malicious code in their Item text.

The next option specifies if the user is allowed to post comments to existing Items or not.

If the setting rating allowed is active a logged in user has the possibility to rate an Item positive or negative. But he only has one vote.

The option automatic Item-ranking means to weight Items statistically with certain criteria. This way ensures that the most informative and high quality Items are placed in a higher position on overview pages like the homepage of the Mobile Reporter. So if this option is active the Mobile Reporter server is responsible for bringing the Items in a order comparable with a search engine algorithm. If this option is not active then the Items will appear on the overview page in order of their Item timestamp when they were sent in. Whereas the most actual Item is positioned on top of the page.

Expert Settings



Picture 6.11: Expert Settings

The expert settings provide options, that can be leaved unchanged, but for expert users they provide helpful functionalities.

The date format field for example specifies the date format that is used for displaying the Item timestamp when the Item was created. Just like the expected parameter of the php function date the required value in this field is a set of characters that specify how the date should be displayed. Each single character has a different meaning. For instance the first character „l“ (lowercase L) stands for the full textual representation of the day of the week, the „m“ means the month as a number with a leading zero, the „d“ displays the day of the month with a leading zero. The „Y“ specifies the 4-digit representation of the year, the „H“ stands for the 24-Hours format of an hour with a leading zero and the „i“ represents the number of minutes with leading zeros. You can find more possible characters and their meanings if you search for „php date“ in a search engine.

The split date format is a way of summarizing several Items together under one headline for example all the Items that were sent on the same day. The split date format option defines the output of this headline.

As already mentioned above for reasons of ranking and statistics every view on an Item is counted. To avoid cheaters to click many times on their own Items to improve their view statistics the IP address of the user is registered at every Item view. This field let's you specify the time in seconds after that an Item view from the same IP address is counted again.

The option check for mails automatically can be activated to enable the process of automatically fetching email from the defined email address and therafter parse the incoming mails to Items. If this option is deactivated the email address will not be checked and therefore no incoming mails are parsed to become Items.

The next option also has to do with the checking of the specified mailbox. Here you can enter a number of seconds after that the emails will be checked automatically. This value should not be to small because sometimes, depending on your freemail provider, there is a connection error if you check your mailbox every 30 seconds.

For the category overview as well as for the homepage the Mobile Reporter server will create thumbnails of the images that are attached to Items. In the options maximum width / height of the thumbnail picture you can enter the width / height in pixel that the image should not exceed.

The last entry in the expert settings specifies the default value of the number of Items in a RSS feed. Remember that this value is only the default that is preset when you create a new category, but can be changed for each category individually.

Important: Don't forget to click the button „change options“ to save your edited settings

6.1.2.2. Rank Administration

The ranks are a good concept to realize different rights for different groups of users. In the ranks section of the administration area you can view the existing ranks, edit them or create new ranks.

Picture 6.12 shows the overview of all existing ranks when you install the Mobile Reporter. Additionally the admin rank exists but it's kept invisible because you shouldn't change it. The ranks are listed vertically and the attributes of the ranks are displayed horizontally. Since the attributes are all self explaining there's no need to explain the single functions of each attribute. By clicking on the pen symbol of a rank you can edit each single attribute as well as the name of the rank.

name	senior user	user
add entry	allowed	allowed
edit own entries	allowed	allowed
edit all entries	forbidden	forbidden
delete own entries	allowed	forbidden
delete all entries	forbidden	forbidden
add category	allowed	forbidden
edit category	forbidden	forbidden
delete category	forbidden	forbidden
add comment	allowed	allowed
delete own comments	allowed	allowed
delete all comments	forbidden	forbidden
edit		

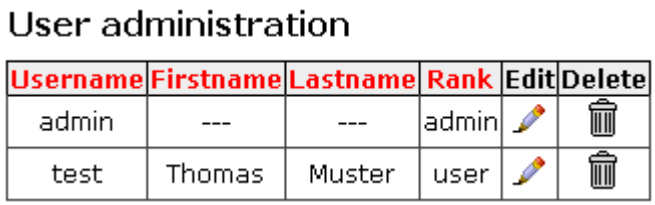
Picture 6.12: Administration, Show ranks

If you want to create a new rank just click on „Add rank“ in the menu. Then the form like in picture 6.13 will appear. Fill out the name field and set the appropriate attributes to allowed or leave them forbidden. Finally press the button „add rank“ and your newly created rank will be stored to the database and is available in the rank overview. After creating a rank you can assign it to an user in the user administration.

Picture 6.13: Administration, Create rank

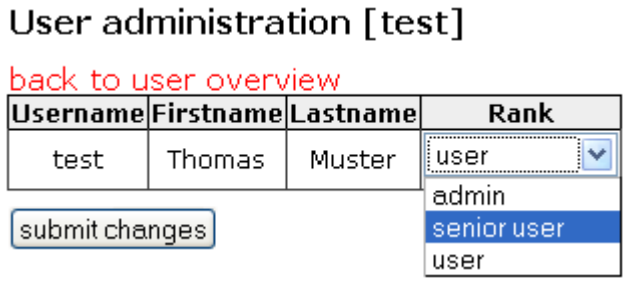
6.1.2.3. User Administration

In this section you will learn how to create, edit and delete users.



Picture 6.14: Administration, user overview

Picture 6.14 shows the user overview list with two registered users together with the rank assigned to the user. To edit the rank of an user just click on the pen symbol in his row. After that the form like in picture 6.15 will appear. There choose an existing rank from the pull down menu and submit changes to update the rank of this user.



Picture 6.15: Administration, edit user

To delete an user simply press on the trash symbol in the accordant row. If you press yes in the appearing confirm dialog the user will be deleted. As the user with the admin rank is the only user that has access to the administration area, he can't be deleted.

To create a new user simply click on „Add user“ and fill out all the fields like in the normal registration form. By default the newly created or registered users are all assigned to the rank user.

6.1.2.4. Email Settings

In the email administration you can specify or edit the data for the mailbox where users can send their Items to. If your email provider is listed in the E-Mail Templates pull down, select it and you just have to enter the email address and the adequate password. If not you have to find out the address of the email server and the used protocol. This data can often be found in the FAQ of your mail provider.

6.1.2.5. Database Settings

If you have to change your database connection data you can edit the required fields in the form of the database administration. After saving this data it will be directly written to the file config.php, because it makes no sense to save the data that is needed to connect to the database in the database itself. So, if an error occurrs you can easily open the file config.php in the main folder where you installed the Mobile Reporter server and edit the attributes of the database.

6.1.2.6. Spam list Settings

If the administrator has specified an email address and activated the option to fetch emails automatically from the email server, there's always the risk that spam mails flood the mailbox. For that reason you can define email addresses in the spam list section of the administration area. If there's an email in the mailbox from a sender that is on your spam list, the email will be rejected. Especially when you registered your email address at a freemail provider, you will receive spam mail at regular intervals. Just find out the senders email address and enter this data to your spam list.

Existing entries of your spam list can be deleted by clicking on the trash symbol beside of the email address you want to delete.

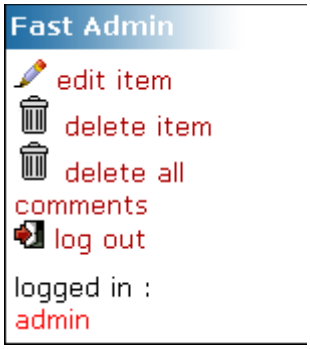
6.1.2.7. Fast Admin Tool

To administrate the basic Mobile Reporter system you have to leave the normal frontend website with all the categories and Items on it. For the administration of the Items and categories itself it would be an advantage not to leave these websites to the administration area, but to edit and delete the Items and categories directly on the relevant websites. To realizethis advantage the Fast Admin Tool was created and placed under the navigation box on the Mobile Reporter website. It allows the admin user to edit and delete Items, categories and comments straight on the overview website. Every user that is logged in can see the Fast Admin box. Depending on the displayed page the Fast Admin changes its menu Items.

On the category overview the Fast Admin, provided that you are logged in as a user with the adequate rights, let's you edit or delete the displayed categories. See picture 6.17.

For example if you are in the overview of the category „News“ then you have the possibility to delete this category or edit its name or the number of Items in the RSS feed with just a few clicks in the Fast Admin Tool. Please note that deleting a category also deletes all the Items stored in this category.

Picture 6.17: Fast Admin Tool, category overview



Picture 6.16: Fast Admin Tool, Item detail view

If you are on an Item detail page where the complete Item is shown, then the Fast Admin Tool provides the functions as you can see in picture 6.16. If you click on edit Item then you will just be directed to the Item Editor where you can edit this Items with all its attachments. The delete Item function let's you delete an Item very quickly and at the same time it deletes the attached multimedia files from the server.

Depending on the settings and the rights of the user that is logged in you can also delete any existing comments with the Fast Admin Tool. Of course this function only exists if there is at

least one comment composed. Logged in as a normal user the Fast Admin Tool makes available less functions. By default the rank user allows to delete your own comments. A single comment can be deleted by clicking on the button with the trash symbol just below the comment.

No matter what rank the logged in user has, the Fast Admin always provides the functionality to quit this session by logging out of the page.

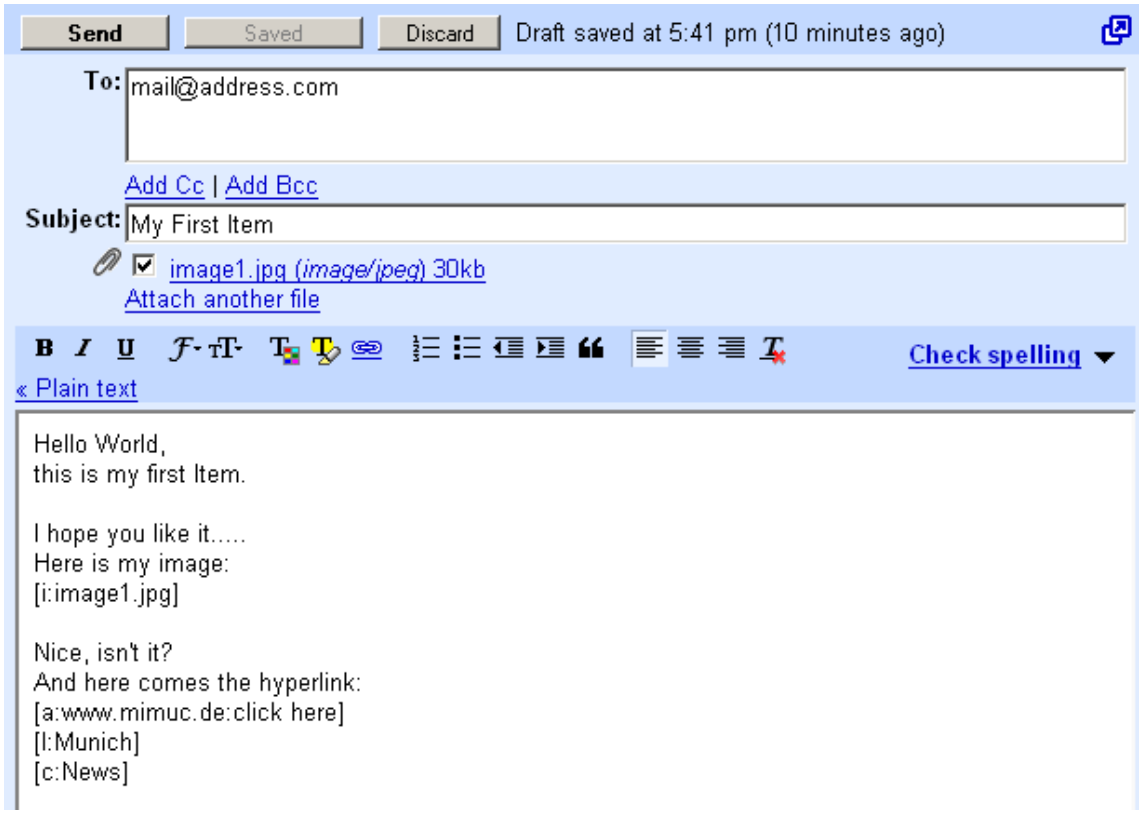
6.1.3. Usage

6.1.3.1. Compose and edit Items

As you have four different possibilities to send an Item the procedure to create a new Item differs depending on which sender you choose. For the MIDlet as sender from the mobile phone the chapter 6.2.4. will give a detailed explanation how to compose Items. But for the senders via the web form and via email and via MMS you will find a description right here.

Compose Items via email

Starting with the composing of Items via email the picture 6.18 will give you a short instruction how to create Items by sending an email to the email address that the administrator of the Mobile Reporter system specified.



Picture 6.18: Compose an Item via email

As you can see the Special Syntax is used in the email text. This is necessary, because in the email you have no specific fields for the category or the location for example. So this is a way to tell the Mobile Reporter server in which category this Item should be saved and where he has to insert the attached image in the text. To learn more about the Special Syntax read the next

chapter.

The process of composing an Item via email should be quite self explaining. You just have to compose an email with a subject that represents the title and with a text that represents the Item text and can additionally contain Special Syntax. If you want to attach an image or any other allowed multimedia file you have to attach this file to your email. Using Special Syntax you can then place the attached file in the Item text where you want.

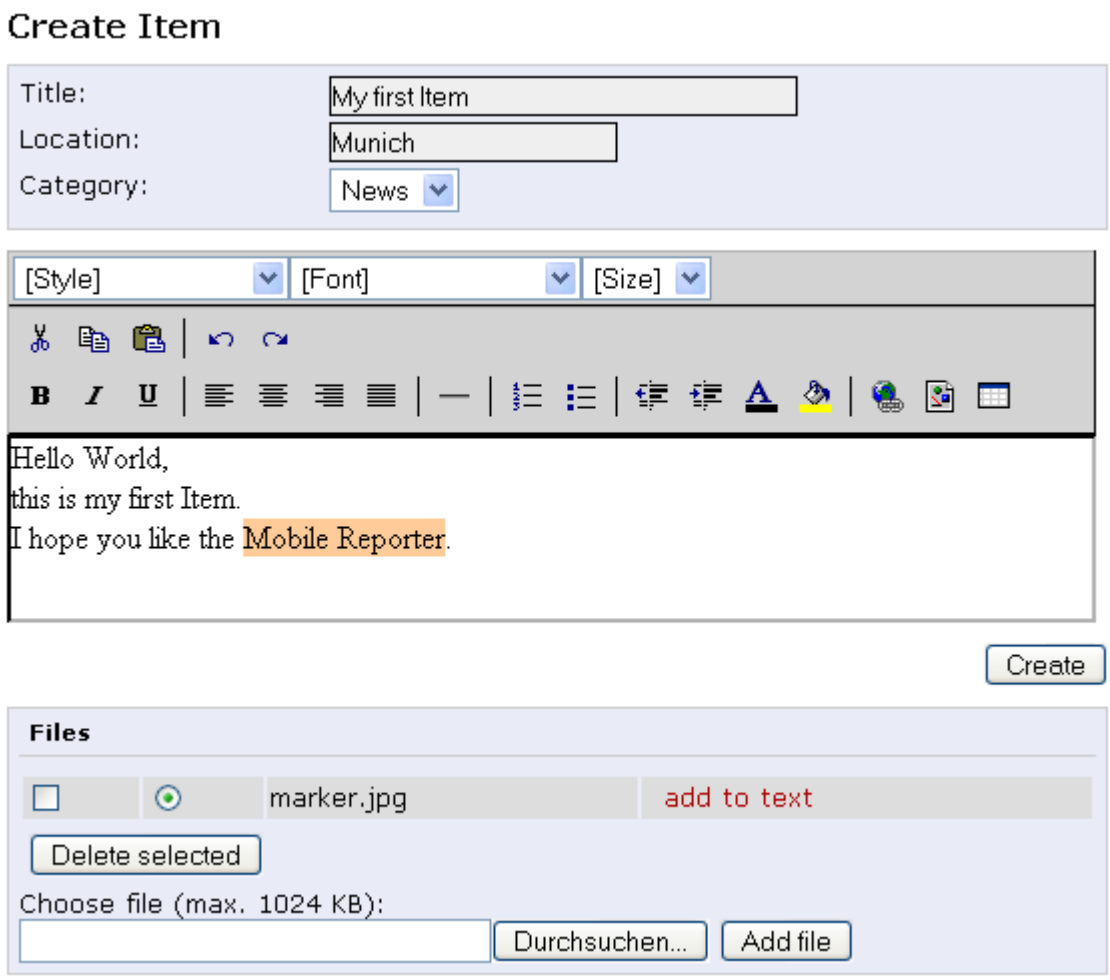
The result after sending this email can be seen in picture 6.20.

Picture 6.20: Result on the Mobile Reporter server
Picture 6.19: Detail view of the composed Item

Compose Items in the Item Editor on the website

The picture 6.21 will show you how the creation of an Item may look like if you use the Item Editor directly on the Mobile Reporter website.

On top of the Item Editor you find three input fields that should be quite easy to understand and fill. The title field represents the title of your Item and therefore this is a required field that you always have to fill. Whereas the field location is not required. You can fill it if you want with the location from where you wrote this Item. The category pull down menu specifies in what category your Item should be saved and displayed. In the beginning the default category is set to „News“, but if there exist other categories, you can select them by clicking on the pull down menu and choosing the category you want.



Picture 6.21: Create an Item over the Item Editor on the website

After doing this you can now compose your Item text in the Item Editor. The Item Editor is a comfortable way to create your Item text as it provides many functions that you should know from standard office software like OpenOffice or Microsoft Word for example. If you ever wrote a document with such an office application you should be familiar with most of the symbols shown in the menu bar of the Item Editor.

The most important functions are the ones to format text like the bold, the italic or the underline tool. But also the functions to align the text to the left, the center and to the right or to force the text to align justified are helpful to create nice Items. The three right most symbols provide really useful functions like inserting links, images or tables. Please note that in the link form, that appears when you click on the symbol with the globe, you have to enter your absolute url with an „http://“ at the beginning. The same applies to inserting an image from a url that you can enter in the form when you click on the image symbol. By the way you can also insert links or images by adding the Special Syntax that was created for the Mobile Reporter project to the Item Editor. To learn more about the Special Syntax read the next chapter. To insert a table click on the rightmost symbol and after specifying some values a table will appear in the Item Editor.

A tremendous advantage of the Item Editor is that you can upload any multimedia file format that was allowed by the administrator in the options. To upload a file just click on the search button, select the file on your hard disk and after that press the button „add file“ to upload the

file to the Item Editor. Be careful that the file doesn't exceed the maximum file size that is set by the administrator. By default this value is set to 1 mb and the actual value is always shown in the file upload box of the Item Editor.

It is very important that all the files that you upload don't have any kind of special characters in its filenames. It might work on some webservers if you upload such a file, but it's recommended to avoid special characters in filenames. Furthermore such filenames can cause problems on the client side depending on the webbrowser and the specific version.

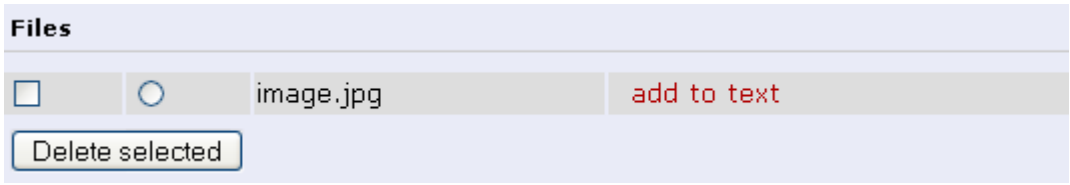
If the file format as well as the filename and the file size are all right, then the file will be uploaded to the Mobile Reporter server. If the upload fails the adequate error message will be displayed, else an entry in the file upload box of the Item Editor will show the newly uploaded file (see picture 6.22). Every uploaded file is displayed in a single line with its filenames and different buttons or functions that you can click.

If you want to delete an uploaded file you just have to select the checkbox at the left of this entry and click on the button „Delete selected“.

The radio button next to the checkbox hast the function to mark the main image that is showed in the Item overview. By default the first uploaded image is chosen for the overview page, but if you uploaded more than one image you can just set any other image as main image by activating the radio button of the adequate image. Of course the functionality to set the main image is only possible if the uploaded file is an image. If you uploaded a sound file for example the radio button will not be there.

With the function add to text you can easily insert the uploaded file to the text in the Item Editor right after the cursor's position. In case of any image file, the image will be displayed exactly at this position. The displayed image can afterwards be scaled to the desired size or be moved in the text per drag and drop. In case of any other multimedia files like sound or video files the function add to text creates a link to the file so that the user can open it quickly.

To finally upload your Item you just have to press the „create“ button and your Item will be saved to the database.



Picture 6.22: Item Editor, uploaded file

The same functionality is also available for mobile phone browser. Picture 6.1 illustrates how websites are reduced by these browser to fit on small screens.

Abbildung 6.1: Item Editor on a Mobile Phone Browser

Compose Items via MMS

Similar to composing Items via email you can compose Items by writing an MMS to the email address that the administrator specified. Picture 6.23 shows an example dialog on a Nokia mobile phone. To insert an email address into the field it is necessary to switch the input mode of your keypad from numbers to text. Then you can enter the email address. After that you can add a multimedia file to your multimedia message by selecting the adequate option on your mobile phone. Afterwards you can enter text which represents the Item text. Here you can use the Special Syntax by inserting it into your message. As you can't specify a subject in your message like in emails for setting the title of your Item it is necessary that you use the Special Syntax construct [t:My title]. All other Special Syntax constructs explained in the next chapter can be applied.

Picture 6.23: Create an Item via MMS

Edit Items

If you have the rights to edit an Item then in your Fast Admin Tool the menu option „edit item“ will appear if you enter the detail view of any Item. If you click on it, the Item Editor will open filled with all the data from the existing Item. Then you have the possibility to change the Item text, delete uploaded files or to add new files and more. In short you have all the functions that the Item Editor provides to ease the composing of Items.

6.1.3.2. Special Syntax

As already mentioned you don't have the possibility to add for example images to the text if you use the other submission ways like email or MMS. To assure this feature to every possible submission way the Special Syntax was created. With it you can display images or links or set the title or the location of an Item without having explicit fields in the form that the user has to fill out. He just writes a short special construct in his Item text and then the server interpretes this token and replaces it with the relevant HTML code.

As the Special Syntax needs to be entered not only over a normal computer keyboard but also over a mobile phone keypad it is kept as simple as possible. Basically you need an open and a closing bracket and a colon as special characters to use the Special Syntax.

With the knowledge how the Special Syntax should be used you can then control five different aspects of the Item:

- Title: [t:<title>] example: [t:My first Item]
- Category: [c:<category name>] or [c:<category ID>] example: [c:News] or [c:4]
- Location: [l:<location>] (lower case 'L') example: [l:Munich]
- Link: [a:<url>] or [a:<url>:<linktext>] example: [a:www.google.com] or [a:www.google.com:search engine Google]
- Image: [i:<filename of the uploaded image>] example: [i:image.jpg]

The title is quite easy to set. You just write down squared bracket open followed by a 't' and a colon and then your desired title followed by squared bracket close.

Setting the category for your Item you have two different options. Either you enter the category

name or to avoid long keyboard input on mobile phones you can enter the category ID. In both cases if the entered value doesn't exist in the database then the default category is chosen. For instance if you make a typing error in the category name, then this category won't be found in the database so that your Item will be stored in the default category.

You set the location of your Item by writing the squared bracket open, the letter l (lower case L) followed by a colon and then your location and finally the squared bracket close.

If you want to insert a link into your Item text you write squared bracket open the character 'a' followed by a colon and then your url where the user should be directed to if he clicks on the link. Note that you can enter the absolute url with 'http://' at the beginning or you can just write 'www.link.com'. You also have the possibility to set relative links for example to another Item or category page of the Mobile Reporter. For that you should enter the relative url (e.g. [a:item.php?itemID=2]) beginning from the path where the Mobile Reporter server is installed. To make it easier to read you can optionally specify a linktext that is displayed as link for the user. To do this you just place another colon after your url and then you write your linktext followed by squared bracket close. With this additional option you can prevent the user from the often cryptic and long url of a website.

To place an image among your Item text you first of all need to upload or attach an image to your Item. You should keep in mind the filename of your uploaded or attached file. Then you go to the position where you want your image to be displayed in the Item text and insert squared bracket open, the character 'i' followed by a colon and then the filename of the uploaded or attached file including the file extension followed by squared bracket close. If the filename that you specified here doesn't exist on the server then the Special Syntax construct is removed out of your Item text. But if the specified file exists, then the Mobile Reporter System converts the Special Syntax to HTML code and your image will be displayed.

6.1.3.3. Create and delete comments

As the Mobile Reporter system should be an interactive platform where users should give feedback to Items published by other users it is possible to comment an Item. Even unregistered users can leave their comments. To do this you simply have to click on the link „comments“ when you are in the category overview or in the detail view of a single Item. Then you'll be able to comment this Item. If you are not logged in you have to enter an username which will be displayed with your comment. The email field is optional. If you specify an email address here other users can contact you over a web form, but your email address is hidden.

In contrast if you are logged in there's just one text field where you can enter your comment and submit it.

To delelete a single comment there will be a small trash symbol below every entry. If you press on it the comment will be deleted after you confirm your intention.To delete all comments of a single Item you just have to click on the „delete all comments“ menu point of your Fast Admin Tool. Note that only administrators or users with a special rank can delete all comments of an Item and in consequence only if you have the accordant rights the menu point will appear.

6.1.3.4. Rate Items

As the Mobile Reporter system should generate a kind of automatic Item ranking for the Item overview page it is important that the user community contributes to this ranking process by rating Items. Thus every user that is logged in can rate every Item once. But to keep it simple you can just rate the Item positive or negative without any steps in between. As every user can rate an Item only once the rating form is not shown if the user already rated this Item.

To rate an Item simply go to the detail view of this Item and there you'll find a pull down menu

to rate this Item. Choose your desired rating and then press the small submit icon on the right to finally rate this Item.

6.2. Mobile Phone Application

In this chapter you will learn, how to edit, install and use the midlet, that was made to submit Items in an easy way via your mobile phone.

6.2.1. Preparation by the Administrator

For correct functionality, the mobile reporter MIDlet needs to know the url, where the mobile reporter server was installed. Normally the user will be asked to enter this url at the first start of the midlet. To ease the installation for the users, it is possible to predefine the url in the following way. Also the maximum number of files that are allowed to attach to an Item can be defined that way.

6.2.1.1. Rename the midlet

The mobile application is called mrep.jar. After installing the mobile reporter server, you will find it in the folder „./download/“. You will find this folder relative to your install directory.

If you for example installed mobile reporter into http://www.yourhost.com/mrep/ then you will find the file at http://www.yourhost.com/mrep/download/mrep.jar.

Download this file and rename it to mrep.zip.

6.2.1.2. Unzip the file

Now unzip the file to any folder you like. In the unziped folder you will find 3 subfolders. Ireporter, org and META-INF. Inside the folder META-INF, there should be only one file, named MANIFEST.MF.

6.2.1.3. Change the file

Open this file with any text editor you like. The data should look similar to text 6.1.

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.6.2
Created-By: 1.4.2_08-b03 (Sun Microsystems Inc.)
MIDlet-1: ItemMidlet,,ireporter.ItemMidlet
MY-URL: http://www.myhost.com/mrep/
MY-MAX-FILES: -1
MY-RSS-ON: 1
MY-PHP-EXT: php
MIDlet-Vendor: mrep Team
MIDlet-Version: 1.0
MIDlet-Name: mrep
MicroEdition-Configuration: CLDC-1.1
MicroEdition-Profile: MIDP-2.0
```

Text 6.1: MANIFEST.MF

In this text file there are 4 lines, that you can modify. MY-URL, MY-MAX-FILES, MY-RSS-ON and MY-PHP-EXT. If one of these lines is missing, just add it.

The line that starts with MY-URL represents the URL of your website. Replace its value with the url where you installed the mobile reporter server. If you, for example installed the mobile reporter system to http://www.yourserver.com/mrep/ then the line should look like MY-URL: http://www.yourserver.com/mrep/.

MY-MAX-FILES indicates how many files users can attach to an Item. You can choose any number you like. Choose the value -1 to define the number of attached files not limited to a maximum. That means that users can attach as many files as they want to. This parameter should be equivalent to the settings of the mobile reporter server.

The line that starts with MY-RSS-ON and enables to switch the midlets RSS reader on and off. The reader is part of the midlet and enables the user to use the midlet to view the newest Items on the webpage. If the value equals 1, then it is switched on, if it is 0, it is switched off. In case it is switched off, the midlet doesn't display the button to start the RSS reader.

The last line is labelled MY-PHP-EXT and should only be modified if you changed the file extensions on the server (see chapter 6.1.1.2.2.). Just place the original value with the new extension. If you changed it on the server to „php4“ then do the same here. If this line should be missing in the file, you can just add it.

After changing it, save the file.

6.2.1.4. Zip the folder

Now zip the folder again. The new .zip file must contain the 3 folders ireporter, META-INF and org, just like the original mrep.jar file.

At the end rename the file to mrep.jar and replace the old mrep.jar in the download folder on your server.

6.2.2. Installation

Depending on your cell phone there are different ways to install the midlet. This is only supposed to be a brief description of the posibilities. For further information please consult the instructions of your mobile phone.

6.2.2.1. Download via mobile phone

One possibility is to download the application with your mobile phones browser. You just need to enter the url as you will find it on the web page and confirm the download. After that you will be asked if you want to install the application.

The advantage of this approach is, that it is very easy. The disadvantage is that you will be charged by your mobile phone provider.

6.2.2.2. Download via Personal Computer

You can also download the application via your pc which will be much cheaper or even for free, depending on your internet access contract.

After downloading it, you have to transfer the application to your mobile phone. Most mobile phones are sold including a cable to connect the mobile phone with your pc but you can also use a bluetooth connection to transfer the data.

After the transfer the mobile phone will display an installation dialog. Follow the instructions on the screen to finish the installation.

6.2.3. Settings

On some mobile phones (e.g. Nokia 6630, Nokia 6600) untrusted java applications require a confirmation by the user for every action that needs access to external ressources. This is the case for using the build in camera, the microphone and internet connections.

For the mrep midlet, this behaviour is not wished. Therefore it is highly recommended to turn this behaviour off. The procedure differs from manufacturer to manufacturer. If you don't want to edit the settings just skip this chapter. You mobile application will also work without any modifications on it.

Here are exemplary descriptions how to proceed on specific Nokia and Siemens phones. For other phones please consult the instructions of your mobile phone.

6.2.3.1. Nokia

On Nokia series 60 mobile phones like the Nokia 6630, the Nokia N90 and the Nokia 6600 you can modify the installed MIDlet so that it will only ask your confirmation once for every action. You can do this by opening the manager. This is an application that manages every installed software. Normally you will find it ar main menu > Tools > Manager. The location may differ on some mobile phones.

In the manager search for an application called mobile reporter and open the options menu. Search the list for a command called „Suite settings“ and choose it. You will see a menu with many different options similar to picture 6.24.

Change the settings for „Network Access“, „Connectivity“ and „Multimedia“ to „ask first time“.

Picture 6.24: Settings on Nokia

6.2.3.2. Siemens

On Siemens mobile phones like the S60, you can't edit the settings, but you can edit the behaviour in a different way. Just start the mobile reporter midlet. The first time you will perform an action that requires authorisation, the mobile phone will give you multiple choices. Choose „allow for the whole session“ and you won't be asked again.

6.2.4. Usage

Depending on the settings done by the administrator, some functionalities may be enabled or disabled. If some of the features described in this chapter are missing in your installation, the administrator probabily disabled them.

6.2.4.1. First Start

The first time you start the midlet, you will see a screen similar to picture 6.25. If the administrator specified a homepage url, the third enter field won't be displayed.

In the fields „User Name“ and „Password“ enter the user name and the password, that you got either during the registration or that was send to you by the administrator.

If there is a field „Homepage“, type in the URL where the mobile reporter serveURL http://www.some-server.com/mrep/, then this is the URL you have to enter in this field.

This data will be used to verify that you have the rights to add Items to the server. You have to enter this data only once, because it will be saved on your mobile phone.

Picture 6.25: Start screen

6.2.4.2. Change User Data

If you want to change the user data, choose edit data from the main menu. This will open the same form that you've seen on the first start. Just change the data and press save.

6.2.4.3. Create an Item

To create an Item, choose the button „Write News“ from the main menu (see picture 6.26). Click it and a form will be displayed on the screen.

The form contains 4 different enter fields. You can see three of them in picture 6.27. Two of these fields are required and marked with a „*“. The first one is the title field. Enter the title for the Item there. The second is the text to be displayed as the Items text.

The optional fields are called place and category. The place stands for the location where the Item was created or the event, that is part of the Items story, happened. The category field indicates the category on the server where the Item will be saved in. It can be the name of the category or its id. If this field is left empty or there is no category with this value, then the standard category will be chosen. If there is for instance only one category on the server, this category will always be chosen.

Picture 6.26: Main Menu

Picture 6.27: Create Item

6.2.4.4. Add Files

The midlet enables you to add audio files and pictures to an Item. If the administrator set the maximum files to 0, the following features won't be available in the midlet.

Picture 6.28: Item Options

6.2.4.5. Add Photos

In the Item form, open the options menu from picture 6.28. Choose the option „Add Photo“ and click it. This will start the camera. Choose the motiv you want to record and press „Shoot Photo“ from the options menu. After that, you have to name the file. Just type in any name you like into the input field like in picture 6.29 and press „ok“.

Picture 6.29: Saving a Photo

6.2.4.6. Add Audio Files

To add an audio file, choose „Record Audio“ from the Item form's options menu from picture 6.28 and click it. This will display a timer and an option menu. The timer displays the seconds recorded. Choose „Start Recording“ from the options menu to start and „Stop Recording“ to stop the record (see picture 6.30). Also an audio file has to be named. Choose any name and press „Save Audio“.

Picture 6.30: Record Audio

6.2.4.7. Show and Delete Attachements

If one or more files are attached to an item. There will be a new option in the item form's option menu. It is called „Show Attachements“. Click it, and a new menu will be displayed. Choose a file and press options. A menu similar to picture 6.31 will be displayed.

For every file „Delete“ and „Add to Text“ will be displayed. Press „Delete“ to delete the file. „Add to Text“ will automatically add a special syntax to your text. This special syntax is interpreted by the server to display either the image at the specific position in the text, or a link to the file if it is not an image. Picture 6.32 shows an example of the special syntax. [i:Pi2.png] means, that at this position, picture 2, that will be uploaded together with this Item should be displayed.

Picture 6.31: Attachments

Picture 6.32: Special Syntax

Depending on the type of the file either „Show“ or „Play“ will be displayed in the menu. For image files there will be „Show“. If you press it, the image will be displayed. For audio files there will be „Play“ instead of „Show“. Pressing it starts to play the audio file.

6.2.4.8. Upload Items

To publish the Item you need to upload it. Choose „Upload“ from the Item's options and click it. The menu from picture 6.33 will be displayed. Press „Start“ from the options menu. If this is the

first upload in this session, you will be asked to confirm the connection to the internet. After that you have to choose the internet gateway out of your cell phones list.

You can stop the upload at any time you want by pressing „Cancel“ and restart the upload another time. If you press cancel, the upload will be stopped and the Item will be saved.

If an error should occur during the upload, you can either go back to the Item form or save the Item.

Picture 6.33: Upload

6.2.4.9. Save and Load Items

Under specific circumstances it can be necessary to save an Item. If you for example can't connect to the mobile phone network you can save the Item and send it later.

While creating an Item, you can save it at any time by pressing „Save Item“ in the options menu (see picture 6.28). The Item will be saved and the main menu will be displayed. If there are saved Items, there will be a new button in the main menu labeled „Saved Items“ like in picture 6.34. Press it and a menu will be displayed containing all saved Items. For every Item you can either choose to delete or show it (see picture 6.35). „Show“ will open the Item form where you can edit the Item and upload it. You can also just change and resave it.

Picture 6.34: Extended Main Menu

Picture 6.35: Saved Items

6.2.4.10. RSS Reader

If it is enabled by the user, you will find a button for the RSS Reader in the main menu like in picture 6.26. The RSS Reader enables you to view the newest Items that were submitted to the mobile reporter server. Press the button in the main menu. After that all the categories that exist on the server will be displayed.

Choose one of the categories and the newest Items will be displayed on the screen.

For using the RSS Reader you need to connect to the internet. Therefore your mobile phone will display all the possible gateways in a list for you. Choose one of them and the connection will be established.

7. User Study

For every software project user studies are an important part of the development process because they help to ahow and avoid errors at a time, where changing them creates less effort then after the publication. For the Mobile Reporter project two user studies were performed which are explained in this chapter.

The english version of the questionnaire that was used for this study can be found in the appendix.

7.1. Pre-Study 10/17/2005

On october 17th 2005, a pre-study was performed. The test persons were first semester students which were using our application on their first day of study.

7.1.1. Purpose

The main purpose of this first small user study was to find out how to arrange the main user study. Including a form, the time for the study and preparations. A side effect was to find out changes, or additional features that users would like to see in our mobile application.

Certainly also some first results regarding the acceptance of mobile blogging and the mobile application were expected.

7.1.2. Execution

In the last weeks of the semester break, letters were send to 20 first semester students explaining our project and asking for their participation. Out of this 20 students the test candidates were chosen.

In the morning of the test day, the test students came together and each of them received a mobile phone with the mobile reporter application. They also got a short description of the main parts of the application. They were told to create and commit as many Items as they wanted to. The costs were covered by the university. They were using the mobile phone for the complete day.

Without their knowledge the time needed to create each Item was recorded on the mobile phone and transmitted to the server, to compare it with their oppinion about how long they needed.

In the evening they gave back the mobile phones and filled out a short test form. See appendix 1 for a translated version of the form. They also were asked to speak about their experience while using the application.

7.1.3. Results

During the study the number of Items per hour, that were send to the server increased massively. This could be because of the fun factor that started to rise within the students. Another reason for this behaviour could be that the students got used to use the mobile application and therefore were able to write more. A fact that speaks for the fun theory is, that every test candidate thought it took him much less time to write an Item then it actually did.

All students said that they missed a function to save Items on the mobile phone. This was required to send Items later, if for example they where at a location where they couldn't get a connection to the mobile phone provider. Such a functionally was immediately implemented

and already integrated for the main user study.

Every user said, that the time it took him to write an Item was adequate and the user interface was easy to use. Asked with whom they would like to share such news, they prefered their school friends and fellow students. Using the system together with familiy members didn't seem to be a wished option. An additional idea for a user group who could use suche a system, clubs of any kind, like football clubs were noted.

When asked for opportunities where they would ike to use mobile reporter, holidays, parties and shopping were mentioned. One test candidate said it would be very important for him that he could use this system to immediatly show something to friends.

One question in the form was intended to find advantages compared to a digital camera. All test users said the main advantage is, that the pictures are instantely available in the inernet.

At the end, they were asked, if they would use mobile blogging, if they had a flatrate for their mobile phone, all answered with yes. That seems to match with the result of another question which asked for a „killer aspect“ of mobile reporter. Everyone answered that costs where the only factor, that would stop them from using a mobile blogging system.

The results were already better then hoped. Therefore nothing important was changed in the user form. The mobile application was only modified as mentioned above.

Picture 7.1 shows some example pictures from the pre-study, that where taken by some of the users.



Picture 7.1: Example Pictures from the Pre-Study

7.2. Main Study

Based on the results of the first study, the main study was performed starting at the 11/28/2005 and ending at the 12/04/2005. They should prove if the final version of the MIDlet now fit the requirements of every user.

7.2.1. Execution

This time no specific day was chosen but instead mobile phones were given to users for maximal one day. In contrast to the pre-study not only informatic students were chosen as test candidates but just random people no matter what they studied or where they worked. They were told to use the application as many times as they wanted to. Just creating and submitting Items whenever desired. This means there was no upper and lower border. However they were encouraged to write at least one Item.

Similar to the pre-study the users got a sketchy description of the application. Also this time, the period it took them to write Items was recorded without their knowledge. The new features, first of all saving Items, were not explained to see if these changes had the desired results.

7.2.2. Results

Not surprisingly, many results were very similar, almost equal, to the results from the pre-study. The new features, like the possibility to save Items on the mobile phone, were accepted very well and most complaints about the user interface, that occured during the first study, were not repeated.

The average time, that user thougth they needed to write an Item was 2 minutes. The real average time, measured by the MIDlet was 3 minutes.

Just like in the main study, user said the main advantage to a digital camera was the fact, that pictures could be shared with others immediatly. This time some users added, that it's good that it gives you the possibility to add other information like text to the picture. And a mobile phone is always available because most people have it with them all the time.

Some new scenarios for the usage were brought up. Including travelogues, event calenders, information sharing, documentations, commercials and announcements.

Asked for groups of persons with whom they would like to share such a system, they would use it together with friends (e.g. from school or university) but very few of them with their family.

At last when asked if they were using mobile internet before, they said no, but almost everyone would use a system like Mobile Reporter if they had some kind of flatrate for there mobile phone.



Picture 7.2: Example Pictures from the Main Study

8. Publication

Since the Mobile Reporter project was meant to be published from the start it began, the only question were under what license and where to publish them. This is described in this chapter.

8.1. License

Choosing a license under which terms a software is published is quite a difficult decision. It is important to find a license that considers not only the authors' rights and safety concerning liability. It is also important what rights are given to any third party that wants to use the software.

For the Mobile Reporter Project, GNU General Public License (GPL) Version 2 was chosen for different reasons.

From the moment the project was started, it was supposed to be published free. Whereas free doesn't mean gratis but the freedom for any third party. Charging money for it is not excluded. In the license this is described the following way:

„When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.“ [11]

The GPL places copyleft restrictions on the program. This means it forces programs based on it to be free as well. Thus this project stays free, because it demands, that distributed programs based on it which are not made by the original authors' are also distributed under the GPL or any compatible license. The changed parts must be marked so that they can be recognized. The recipients of such programs must know, that what they received is not the original „so that any problems introduced by others will not reflect on the original authors' reputations“. [11]

Authors that distribute their software under this license don't give any warranties, but they can if they want to, and can not be prosecuted for any damage that is done by it.

An additional important aspect for choosing this license was the fact that its legality has been confirmed by courts in many countries like the united states and germany.

8.2. Sourceforge

After choosing a license it is important to decide where to publish a project. One possibility is just to create an website for the project and try to bring it up on the search results of the biggest search engines and promote it by mouth to mouth propaganda. This is a hard way for several reasons. Capabilities for downloads and background technologies like CVS repositories and more have to be provided by the developers. This not only consumes much time it is also a matter of costs.

Fortunately there is Sourceforge. Browsing their website for a description this can be found. „Sourceforge is the world's largest Open Source software development web site, hosting more than 100,000 projects and over 1,000,000 registered users with a centralized resource for managing projects, issues, communications, and code. SourceForge.net has the largest repository of Open Source code and applications available on the Internet, and hosts more Open Source development products than any other site or network worldwide. SourceForge.net provides a wide variety of services to projects we host, and to the Open Source community“ [j].

This means that Sourceforge provides the users with a full environment to develop, host and administrate their projects. It provides unlimited CVS repositories, download server, webspace, databases, discussion boards, bug tracker and much more. The only requirement is, that the projects, hosted by Sourceforge must be Open Source. This means they must refer to an Open source compatible license like the GPL. A very interesting part of the conditions one agrees by choosing Sourceforge is the fact, that no CVS data may be deleted ever and every user has read access to the cvs repository. Thus every person can download every old version of the project he wants to.

Since these are very good conditions, it was decided to publish the Mobile Reporter project on Sourceforge. To be accepted there it is mandatory to submit a form that describes the whole project and its unique selling proposition (USP). This form is manually checked by a Sourceforge staff who decides if it is accepted or refused. Mobile Reporter was accepted three hours after submitting the form which indicates that it seems to be somehow interesting. In the future the Mobile Reporter project will be available at the URL http://mrep.sourceforge.net.

9. Perspectives

Even though the project thesis is finished the Mobile Reporter project lives on. With the publication on Sourceforge the base is made to get additional developers to participate on the further development of the project. This could include new features and the creation of other applications to submit Items, for all kinds of devices.

As the duration of this project is limited there are several features that couldn't be realized in this time. Among different extra features on the Mobile Reporter website the idea of a closed user blog exists, like a private room where the author of an Item can define the „visibility“ of Items. That means he can assign a category that is only viewable for logged in users or for certain user groups or ranks that can be defined. Additionally the idea to track the position of the author automatically with GPS integrated in mobile phones is a feature that could be realized in future. This information could be used to improve the automatic ranking algorithm. For example if many Items from the same location are sent it can be assumed that they all contain news about the same event. In that way redundant Items can be filtered out and so the overview of Items could be improved.

Many useful ideas for features that could be implemented in future were figured out by the user studies and many more will be discovered if the Mobile Reporter system is in use on many webspaces. Therefore:

Try it!

Appendix

Questionnaire Mobile Reporter

Personal data

Gender: □ male □ female

Age: _____

Usage of mobile phones

Do you own a mobile phone?
□ yes □ no

if yes:

Which device (manufacturer) do you have? _____

I send
□ 0 □ 1-10 □ 11-20 □ 21-30 □ 31 and more SMS per Month

I send
□ 0 □ 1-2 □ 3-4 □ 5-6 □ 7 and more MMS per Month

Did you ever use Wap / I-Mode with your mobile device?
□ yes □ no

if yes, what was your purpose?
□ Information □ Entertainment □ Communication □ Shopping

Do you know blogging / mobile blogging?
□ yes □ no

Questions to the Mobile Reporter

How much time it took you to compose an Item?
□ 30 seconds □ 1 minute □ 2 minutes □ 3 minutes □ more than 3 minutes

How do you rate this amount of time?
adequate □ □ □ □ □ too long

Appendix

Was the user guidance intuitive?

absolute □ □ □ □ □ not at all

Would you prefer the Mobile Reporter as a private blog or as a public blog or both equal?

□ private blog □ public blog □ both equal

Specify how you can imagine the usage of Mobile Reporter with the following group of people:

with family members:	absolute □ □ □ □ □ not at all
with fellow students	absolute □ □ □ □ □ not at all
with schoolmates	absolute □ □ □ □ □ not at all
other ideas:	

At what opportunity you would use the Mobile Reporter?

What are the advantages of the Mobile Reporter compared to a digital camera?

Are there any functions that you missed in the application?

Do you know other scenarios to use this software / portal?

Is there any killer-aspect for the usage / respective the non usage of such a software

If you had a flatrate for data transmission on your mobile phone would you use the Mobile Reporter system?

□ yes □ no □ maybe

Do you have any suggestions for improvement of our software?

List of Literature

[1] Shayne Bowman and Chris Willis. We Media: How Audiences are Shaping the Future of News and Information 28. Nov 2005 http://www.hypergene.net/wemedia/weblog.php?id=P36

[2] Danny Sullivan. Google Buys Blogging Company - But Why? 18. Feb 2003. http://searchenginewatch.com/sereport/article.php/2165221

[3] picturephoning.com. Exploring the new world of picture and video phones. 02. Dec 2005. http://www.textually.org/picturephoning/

[4] Blogger.com. Donwload Blogger for Word. 27.Oct 2005. http://buzz.blogger.com/bloggerforword.html

[5] Consultechnology. Mobup by Consultechnology. 05. Nov 2005 http://mobup.consultechnology.com

[6] Free Software Foundation, Inc. GNU Lesser General Public License. 05. May 2005. http://www.gnu.org/copyleft/lesser.html

[7] Wikipedia. Web Syndication. 21. Oct 2005. http://en.wikipedia.org/wiki/Web_syndication

[8] Harvard Law. RSS 2.0 Specification. 30. Jan 2005. http://blogs.law.harvard.edu/tech/rss

[9] Opera Software ASA. Small Screen Rendering. http://www.opera.com/products/mobile/smallscreen/

[10] Jonathan Knudsen. Wireles Java: Developing with Java 2 Micro Edition, Chapter 10, Performance Tuning. 06. 2001. http://developers.sun.com/techtopics/mobility/midp/chapters/j2meknudsen/Chap10.pdf

[11] Free Software Foundation. 07. Jun 2005. http://www.gnu.org/licenses/gpl.html

Web References

[a] Blogger.com - http://www.blogger.com

[b] Flickr - http://www.flickr.com

[c] Picostation - http://www.picostation.com

[d] PHP - http://www.php.net

[e] MySQL - http://www.mysql.com

[f] Smarty - http://smarty.php.net

[g] J2ME - http://java.sun.com/j2me/

[h] Winzip - http://www.winzip.com

[i] SmartFTP - http://www.smartftp.com/

[j] Sourceforge - http://www.sourceforge.net

[k] Cross-Browser Rich Text Editor by Kevin Roth - http://www.kevinroth.com/rte/demo.htm