

Functional Specification SmartMeeting

T-76.115 Software Project

Anssi Kesti-Helia Timo Mali

23rd April 2002

Version history

\$Log: functional_specification.tex,v \$

Revision 1.25 2002/03/25 09:34:58 akestihe
Just reviewing the documents.

Revision 1.24 2002/03/24 22:33:23 akestihe
Added a new function: A secretary can add tasks when a meeting is on.

Revision 1.23 2002/03/23 11:12:56 akestihe
Reduced some functions that will not be implemented.

Revision 1.22 2002/01/11 10:17:49 tmali
Adding some time attributes

Revision 1.21 2001/12/09 16:26:00 akestihe
Something for the eighth chapter. Fixed some references and the class diagram.

Revision 1.20 2001/12/07 10:07:53 tmali
Adding one more requirement with prior 5

%%Revision 1.19 2001/12/06 13:58:26 akestihe
%%Fixing errors.

%%Revision 1.18 2001/12/05 11:59:34 tmali
%%correcting some errors

%%Revision 1.17 2001/12/03 18:30:24 tmali
%%inserting framework picture and 3.3 fixing

%%Revision 1.16 2001/12/01 14:16:17 akestihe
%%Something for chapters two and seven.

%%Revision 1.15 2001/12/01 10:46:41 tmali
%%Some into 3rd chapter

%%Revision 1.14 2001/11/30 17:43:02 akestihe
%%Chapter one.

%%Revision 1.13 2001/11/30 13:05:40 akestihe
%%New functions to functional specification, typo fixed in reqspec.

%%Revision 1.12 2001/11/28 11:16:39 tmali
%%Some functions added and some into 6th chapter

%%Revision 1.11 2001/11/25 19:42:25 akestihe
%%A couple of new functions.

%%Revision 1.10 2001/11/25 18:05:43 akestihe
%%Mainly merging differences.

%%Revision 1.9 2001/11/25 15:43:24 tmali
%%Participating to a meeting added.

%%Revision 1.8 2001/11/25 14:23:43 tmali
%%changed requirements index policy

%%Revision 1.7 2001/11/25 13:16:28 akestihe
%%Some new functionalities.

%%Revision 1.5 2001/11/20 11:42:13 akestihe
%%*** empty log message ***

%%Revision 1.4 2001/11/20 07:18:20 akestihe
%%Something for 5. Commented out the instructions.

%%Revision 1.3 2001/11/18 17:29:36 tmali
%%Meeting management functions and some drafts into 6th chapter.

%%Revision 1.2 2001/11/16 14:17:27 tmali
%%Some thinking into 4th section

%%Revision 1.1 2001/11/12 21:46:31 akestihe
%%Added first draft of the functional specification.

Contents

1	Introduction	6
1.1	Purpose and Coverage	6
1.2	Product	6
1.3	Terms and definitions	6
1.4	Contents of This Document	8
2	General Description	8
2.1	Environment	8
2.2	Function	9
2.3	Users	10
2.4	General Limitations	10
2.5	Assumptions and Dependencies	11
3	Data and Database	11
3.1	Data Contents	11
3.2	Usage Intensity	13
3.3	Capacity Requirements	13
4	Functions	13
4.1	General	13
4.2	System functions	14
4.2.1	Meeting Management	14
4.2.2	User Management	16
4.2.3	Group Management	16
4.2.4	Access Rights	17
4.2.5	Participating to a Meeting	17
4.2.6	Tasks	18
4.2.7	Documents	19
5	External Interfaces	19
5.1	Hardware Interfaces	20
5.2	Software Interfaces	20

5.3	Communication Interfaces	21
6	Other Features	21
6.1	Performance and Response Times	21
6.2	Usability, Recovery, Security, Protection	22
6.2.1	Usability	22
6.2.2	Recovery	22
6.2.3	Security	22
6.2.4	Protection	22
6.3	Maintainability	22
6.4	Mobility and Portability	23
6.5	Operability	23
7	Design Limitations	23
7.1	Standards	23
7.2	Hardware limitations	23
7.3	Software Limitations	23
8	Ideas for Further Development	24

1 Introduction

1.1 Purpose and Coverage

This document is the functional specification document for a SmartMeeting software of HUT course Tik-76.115, Software Project [1]. The customer of the project is PM & RG (Product Modelling & Realisation Group) [2]. The purpose of this document is to describe the functions of the SmartMeeting software.

The user interfaces are not defined in this document. Other functions are defined here.

1.2 Product

The software product that is developed during the course is called SmartMeeting. The purpose of this project is to design and partly implement a generic architecture for SmartMeeting.

The aim of the project is to produce an application that helps having ubile meetings. With SmartMeeting one should be able to have official meetings that occur at a certain place at a certain time as well as unofficial meetings that are not bound to any particular space or time. The customer of the project has a need for a software that supports remote participants and meetings that can go on days or even weeks after the assembly. Meetings could also begin before the assembly, or have no assembly at all.

The remote participants of the meeting could attend the meeting with a web browser. With the web interface of SmartMeeting they could chat with one another and with the participants present at the assembly. They can also see practically in real time how the meeting minutes gets composed or changes.

The meeting minutes and other documents of the meetings often have importance only for one meeting. They should have, of course, but they are often forgotten because there is no obvious way to reuse them. SmartMeeting should make these documents useful for future meetings. With time passing and meetings held, the documents get a history. The documents can be linked to other documents, meetings, tasks and users. Not only the documents, also meetings and tasks get a history.

1.3 Terms and definitions

Assembly The actual physical meeting.

Meeting The actual physical meeting (assembly) with its remote participants and the time and discussion before and after the assembly until

the time when the meeting minutes is closed.

Secretary The user of the meeting that writes the decisions and outcomes of the discussion down to the meeting minutes.

Remote user A user that is not present at the assembly but participates to the meeting with her telephone, or via chat, or some other means of remote communication.

Mobile For example small, light, portable and wireless devices and services.

Ubiquitous Present everywhere

Ubile Words “mobile” and “ubiquitous” combined. The meaning is that the service can be used anytime anywhere with mobile devices.

IRC Internet Relay Chat. A multi-user chat that anyone with an IRC-client can use anywhere.

HUT Helsinki University of Technology

PDA Personal Digital Assistant is a term for any small mobile hand-held device that provides usually small and handy tools for personal and business use such as calendar and address book.

Ideapool Ideapool is a a list of various ideas. If it is used right, it helps people to remember different issues.

SMS Short Messaging Service. The transmission of up to 160 characters between mobile phones.

DBMS Database Management System. A collection of programs that enables a user to store, modify, and extract information from a database.

ODBMS Object-oriented Database Management System. Also called OODBMS. A database management system that is based on object-oriented programming.

HTTP HyperText Transfer Protocol. The protocol that is used between a web browser and a server to request for a document and to transfer it.

SSL Secure Socket Layer. A protocol for encrypted transmission over Tcp/IP networks. Developed by Netscape.

TLS Transport Layer Security. The latest version, or the successor, of SSL. A proposed Internet Standard.

HTTPS HyperText Transfer Protocol (Secure). HTTP over SSL.

SMTP Simple Mail Transfer Protocol. It is the main protocol for sending electronic mails on the Internet.

JDK Java Development Kit. A software development package for Java applications from Sun Microsystems.

1.4 Contents of This Document

The first chapter of this document is the introduction to the functional specification. The introduction includes the purpose of this document, the general description of the product, the terms used in this document and the references.

The second chapter describes the functionality of the system in a general level without going into details. The chapter includes descriptions of the environment, functions, users, general limitations and assumptions and dependencies.

Chapter three describes the data and data contents of the product and how data is stored in SmartMeeting.

In chapter four we define the functionality of the product. There is a short description for each function from the user's point of view.

Chapter five includes the external interfaces of the system. That means that hardware requirements, software requirements and communication interfaces are described there.

Chapter six describes the non-functional requirements of the system. That includes the performance and response times, usability, recovery, security, maintainability, mobility/portability, compliance and operability.

Chapter seven tells something about the limitations that affect the design of the system. Some standards as well as hardware and software limitations are described there.

In chapter eight we reveal some ideas that we have for possible future development.

2 General Description

2.1 Environment

The SmartMeeting software needs a server where it runs. It also needs an Ozone database that can be either on the same server as SmartMeeting or on another server somewhere else. Otherwise SmartMeeting is a stand-alone product. There are interfaces for external applications, such as receiving

and sending SMS. However, SmartMeeting can be used also without these external products and the features provided by them.

SmartMeeting uses the Internet to transfer data. It provides a web interface, so SmartMeeting can be used with web browsers. This of course means that the users, or their computers or other terminals, have to be connected to the Internet.

SmartMeeting is programmed in Java, so it runs on many different platforms, including Windows and Linux.

2.2 Function

The most essential part of SmartMeeting is naturally meeting management. The concepts assembly and meeting are important in SmartMeeting. An assembly is the actual gathering where users are physically present and that has pre-set place and time. Official meetings are often like this. The concept of meeting is somewhat broader in SmartMeeting. Meetings can last days or weeks and there can be lots of remote activities.

Users of SmartMeeting can add new meetings to the system. SmartMeeting has ready templates for agendas of new meetings. It should be easy to use old meeting minutes and agendas when creating new ones. SmartMeeting has an ideapool, where subjects can be placed. The secretary can take the subjects from the ideapool to the agenda of a meeting.

The meetings can be modified and deleted mainly by users with secretary rights. When a meeting is over, it can be closed. Users can browse old, current and future meetings.

It is very common that decision making takes place in meetings. Therefore SmartMeeting will have support for both voting and making and recording decisions made in meetings. Decisions can be modified until the meeting is closed.

SmartMeeting has at least one administrator. Administrators can, besides other things, manage the user accounts of SmartMeeting. User accounts can be added, modified and deleted. The users can change their passwords and contact information. They can also browse the contact information of other users. The system supports also user groups that consist of users or other user groups.

Users have different levels of authorization. Users can modify only such information that their access rights allow.

Users can participate to meetings for example by being physically present or by chatting with the SmartMeeting web client. The meetings should follow the meeting agenda. The secretary selects the subjects. The remote users

can see practically in real time how the meeting minutes and possibly other documents develop and change.

Users can have tasks that they need to prepare for example for the next meeting. One task can have many responsible users. Users can browse the tasks and volunteer for them.

There is a storage for documents in SmartMeeting. Users can add documents to the system. The documents can be modified and deleted. The documents can have history information if they have been modified and used in meetings. The documents can be linked to subjects.

2.3 Users

SmartMeeting has administrators. As said in chapter 2.2, they can manage the users of the system. The administrators can also use the system as “normal” users. In addition to the administrator of SmartMeeting, there should be some technically oriented person who is familiar with the technical foundation of the software. That means that this person knows how to restart SmartMeeting and other software required, in case of a system failure. There will probably be sufficient startup scripts for starting any needed components, so no deeper understanding of the SmartMeeting architecture is needed.

The users of SmartMeeting do not need any special education, nor do they have to be representatives of any particular occupation. The users probably need one session of instruction to get acquainted with the software and the user interface. After that, the users should be able to use the system.

Practically anyone can have meetings. The status of the participants of the meetings can vary pretty much endlessly. A company meeting may have participants from the management, marketing and workers. One could not probably tell what are the professions of the participants of a meeting of a sports club. Therefore, we cannot really limit the usage of SmartMeeting to any particular group of people. And furthermore, SmartMeeting should be easy enough to be used by practically anyone.

It is not important how often the system is used. It should be ready for use at anytime and from anywhere. It should function equally when used once a year or on a daily basis.

2.4 General Limitations

The information that is handled in meetings can be confidential, even protected by laws. Therefore the information should be protected, at least when someone starts using SmartMeeting with confidential material. Protecting

information can make the system slower, as for example encryption and decryption do. However, this effect should not be critical.

SmartMeeting cannot be too slow, otherwise no one will use it. This means that the response times should be small and the system should be stable enough so that it does not crash even under heavy load.

Although it would be convenient to have a calendar, or complete SMS functionality, integrated in SmartMeeting, it would be too big a task to implement such software. Thus we have to leave the implementation of these software to others, the future users of SmartMeeting.

2.5 Assumptions and Dependencies

SmartMeeting needs JDK 1.3 or higher. SmartMeeting will run on both Linux and Windows and possibly other platforms supporting Java. SmartMeeting will be developed on a server that has 256 MB memory, 1 GHz AMD Athlon processor and a 12 GB hard disk. SmartMeeting probably runs also with slower hardware but that will not necessarily be tested.

3 Data and Database

3.1 Data Contents

The datastorage used is Ozone-db [4]. It is object-oriented database management system. It has been explained more in chapter (5.2). Our data contents and the most important relations between data components are described with UML-diagram below.

More detailed information of data objects are represented in generated javadoc api-documentation.

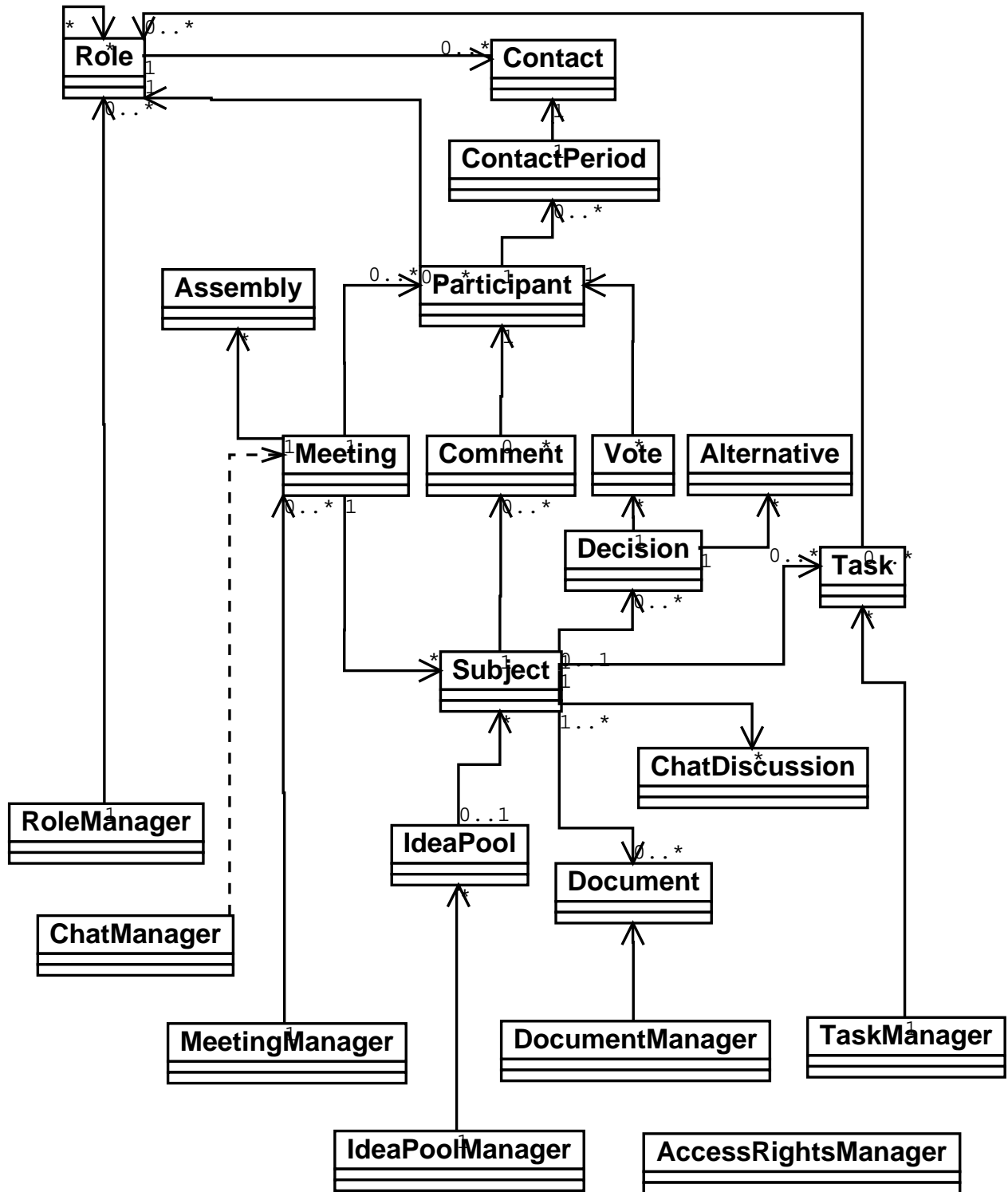


Figure 1: UML-diagram of data content

Data objects live inside the database server of the SmartMeeting server. Data is moved between cache and diskstorage by Ozone-db software. The persistence of the data objects is provided by proxy objects that are manipulated by the user. If multiple users use same data object simultaneously they all have their own proxy object of the single data object. When the proxy object is called the data object is loaded into mainmemory of the server.

3.2 Usage Intensity

Our implemented prototype will be able to serve at least 10 concurrent users. In real life use system should be able to handle hundreds of concurrent users.

Usually the most active participant during the assembly is the secretary. Also other participants can be very active and the supported maximum user intensity should be atleast 20 transaction per minute per user on the average. The system has to be able to serve 10 basic searches per minute per user. The most complex searches can take substantially longer times than basic searches but those are performed rarely. So the maximum user intensity of the system is 200 transactions per minute.

There are not any time limits to restrict the use of the system. Only possible maintenance tasks can cause some use breaks.

3.3 Capacity Requirements

According to the discussion in ozone-db news group the size of the database can be even gigabytes but the size of the cache is crucial. The cache should include the working set of data. In our use the working set is always quite small and the required size of the cache will not become too large. [8]

We do not estimate any exact numbers of documents or meetings that the system should be able to handle.

4 Functions

4.1 General

The common issues of all functions concerning the user interface are decided later and they are not documented in this document.

Our functions have priorities: 1 (must have) and 2 (useful). The requirement of priority 3 in Requirement Specification document are not processed to the functions. If later these requirements are decided to be implemented they will be documented here.

The priority of the function is written just after the bolded name of the function. The references to the Requirement Specification document are marked as (RS.XXXX) at the end of each function. References to the other functions of this document are presented like (FS.XXXX). [7]

4.2 System functions

4.2.1 Meeting Management

MM1. Adding a new meeting. (1) A user can add a new meeting to the system. When adding a new meeting the user can specify list of participants and list of participants with secretary rights. She can also specify the place of the meeting and the starttime and the endtime of the meeting. The user can also give the name and the additional description of the meeting used for searching. The user can also leave previous parameters open. (RS.MM1)

MM2. Modifying meeting (1) A secretary can modify the list of participants and the list of participants with secretary rights. The modifying secretary can not take herself out of the list of participants with secretary rights if she is the only one in the list otherwise this is possible but she can not modify meeting after completing this function.

A secretary can modify the name, description, starttime or endtime of the meeting and the place of the meeting. (RS.MM2)

MM3. Deleting a meeting (1) A secretary can delete a meeting. The Secretary is asked to confirm her decision before removing is completed. (RS.MM3)

MM4. Closing a meeting (1) A secretary can close a meeting. After closing the meeting the meeting minutes of the meeting can no more be modified. The closing time is recorded as a endtime of the meeting. Meeting is closed automatically when specified endtime is exceeded. (RS.MM4)

MM5. Browsing and searching a meeting (1) A user can browse old, current or future meetings. The user can search meetings based on given time interval, participants, name, description or subject.(RS.MM5)

MM6. Creating a decision. (1) A secretary can create a decision. Decision is created when the decision process is started. Decision process has four subsequent phases: 1. creation, 2. discussion about decision and finding alternatives of the voting, 3. voting and 4. decision making. Subject of the decision has to be specified. Deadline time of the decision can be specified.(RS.MM6, RS.MM7)

MM7. Beginning voting on a decision. (1) After discussion about the subject, the alternatives of the voting has been identified and a secretary can specify voting alternatives into SmartMeeting. After alternatives are

specified the secretary can launch the voting on a decision. Voting last until all votes are got or deadline time is exceeded.

MM8. Making a decision. (1) After decision has been made in a meeting a secretary can write down the decision into system. If the decision is based on some voting the secretary can use participants voting results to create some decision. Some statistics of votes is shown to support decision making. Decision is added with voting result to the minutes. Also the time when decision is made is shown. (RS.MM23)

Decisions can be made also without voting. The decision process has then only three phases: 1. creation, 2. discussion and 3. decision making.

Decision can be made if decision deadline is exceeded or result of voting of every participants has been got.

MM9. Modifying decision. (2) A secretary can modify alternatives of the voting before voting has started. The secretary can modify decision until the meeting is closed. (RS.MM15)

MM10. Deleting decision. (2) A secretary can delete a created decision and stop the decision process before voting has started.(RS.MM16)

MM11. Adding subjects to meeting agenda. (1) A secretary can add new subjects or subjects from ideapool to the meeting agenda. Name of the subject has to be specified. Subjects can be added to the meeting agenda before, during or after assembly.(RS.MM8)

MM12. Remove subjects from meeting agenda. (1) A secretary can remove subjects from the meeting agenda. (RS.MM9)

MM13. Adding subjects to ideapool. (1) A user can add new subjects to the ideapool. A Secretary can move subjects from a meeting to ideapool for future meetings.(RS.MM10)

MM14. Closing an assembly. (2) A secretary can close an assembly. This will change the appearance of SmartMeeting so that the remote user can easily see that the physical meeting is over.(RS.MM11)

MM15. Decision vote set by secretary. (2) A secretary of a meeting can vote for decision on behalf of a participant. (RS.MM17)

MM16. Changing order of subjects in meeting agenda. (2) A secretary can modify the order of subjects in a meeting agenda. Already handled subjects can not be moved below subjects which have not been discussed yet. (RS.MM18)

4.2.2 User Management

UM1. Adding a user. (1) An administrator can add a user to the system. (RS.UM1)

UM2. Deleting a user. (1) An administrator can delete any user account except her own. If administrators could delete their own user accounts, it would be possible that there are no administrators at all. (RS.UM2)

UM3. Changing password of a user. (1) An administrator can change the password of a user. If a user forgets her password and no one could change them, the amount of users would grow all the time because new user accounts would have to be created even for old users of the system. (RS.UM4)

UM4. Changing role of user accounts. (1) An administrator can change the role (user/administrator) of any user account except her own. If an administrator could change her own account, it would be possible that there are no administrators. (RS.UM3)

UM5. Logging in. (1) A user can log into the system by typing in her username and password in the login screen of the system. If she types an incorrect username or password, she is not allowed to log in but she is asked to log in again. The user is allowed to use the system only if she types in the correct username and password. (RS.UM5)

UM6. Changing password. (1) A user can change her password. When changing her password, she types in her current password and the new password twice. If the current password does not match with the one the user types in, the password is not changed. If the new password is not typed in twice or the two passwords the user types in do not match, the password is not changed. If the current password is correct and the user types in the same new password twice, the password is changed. (RS.UM6)

UM7. Changing contact information. (1) A user can change her contact information. (RS.UM7)

UM8. Setting notification methods for new/updated meetings. (1) A user can set the notification methods of how she is informed about new or updated meetings. An updated meeting is, for example, one whose date or time or agenda has changed. Notification methods include, for example, SMS and e-mail. (RS.UM8)

UM9. Browsing contact information. (2) A user can browse the contact information of other users. (RS.UM9)

4.2.3 Group Management

GM1. Creating, deleting and modifying groups. (2) A user can create and delete a usergroup. She can add users into the usergroup and delete

users from the group. (RS.GM1, RS.GM2, RS.GM4, RS.GM6)

GM2. Adding groups into a usergroup. (2) A user can add a usergroup into a usergroup. (RS.GM3)

GM3. Deleting group from a usergroup. (2) A user can delete a usergroup from a usergroup. (RS.GM5)

GM4. Using groups instead of a user. (2) The group can be used instead of a user anywhere in the system where it is reasonable. (RS.GM7)

4.2.4 Access Rights

AR1. Adding access rights of new information. (1) A user creating new information will have access rights to view/modify/delete the created information. She can also set access rights for other people to the information. (RS.AR1)

AR2. Access to viewing, modifying, or deleting information. (1) A user can see, modify or delete information if she is authorized to see, modify or delete the information. (RS.AR2, RS.AR3, RS.AR4)

AR3. Access to changing access rights of information. (1) A user can change access rights only for the information that she is authorized to change access rights for. (RS.AR5)

4.2.5 Participating to a Meeting

PM1. Signing into an assembly. (1) A user can sign in to an assembly that is going on. She has to specify her participation method from possible values as e.g. client, mobile phone or physical. (RS.PM1, RS.PM2)

PM2. Signing out of an assembly. (1) A user can sign out of an assembly that is going on. (RS.PM3)

PM3. Signing a user into an assembly. (1) A secretary of a meeting can sign a user into or out of the assembly of the meeting. (RS.PM4, RS.PM5)

PM4. Participating by chat. (1) A user can participate in the conversation from anywhere by chat. (RS.PM6)

PM5. Selecting next decision. (1) The secretary can also select an already made decision and the decision can be made again. (RS.PM7)

PM6. Selecting next subject. (1) A secretary can select next subject for processing. (RS.PM8)

PM7. Following current subject with client. (1) A user can see the current subject that is under discussion with her client. (RS.PM9)

PM8. Voting with a client. (1) A user can give her vote for a decision

with client during and after an assembly depending on the deadline of the decision. The user can give only one vote per decision. The misuse is informed and only the first vote is registered. (RS.PM10)

A voting result is shown to the user. The number of votes on different alternatives is shown. (RS.PM11)

PM9. Commenting subjects. (1) Subjects can have comments and comments are divided between two level. The comments of the first level can be seen in the meeting minutes.

A participant can add comments under any subject to the second comment level during and after an assembly. A user can see the comments of the first level in the minutes and she can follow link to see and browse the comments of the second level.

Only a secretary can move relevant comments from the second level into the first level. (RS.PM12)

PM10. Reading a meeting minutes. (1) A user can read meeting minutes during and after a meeting in the format that resembles traditional meeting minutes. (RS.PM13, RS.PM14)

PM11. Printing a meeting minutes. (1) A user can print a minutes in the format that resembles a traditional minutes. (RS.PM15)

PM12. Registering to an assembly in advance. (2) A user can inform beforehand if and how she is going to participate in an assembly. The user is inserted as a participant in the meeting with specified contact media. (RS.PM16)

PM13. Voting an absentee vote with client. (2) A user can give an absentee vote with client before the assembly of a meeting. The secretary can then specify one alternative to be the same as absentee vote was and the others can vote for that alternative later in the assembly. Only one vote per user is registered. (RS.PM18)

PM14. Opening documents associated with a subject. (2) A participant can easily open documents associated with a subject that is in the meeting agenda. (RS.PM19)

PM15. Adding tasks. (2) A secretary can add tasks when a meeting is on. While adding a task she can also set the responsible users for the new task.

4.2.6 Tasks

T1. Browsing tasks. (1) A user can browse her own tasks. She can see all her tasks. She can also browse other than her own tasks if she has proper authorization.(RS.T1)

T2. Creating tasks. (1) A user can create new tasks. When creating a new task, the user gives the task a name. She can also add some information concerning the task. (RS.T4)

T3. Deleting tasks. (2) If a user has sufficient authorization level, she can delete a task. (RS.T5)

T4. Closing tasks. (2) If a user has sufficient authorization level, she can set the status of her own task as finished. (RS.T6)

T5. Volunteering to a task. (2) A user can volunteer to a task that is open. (RS.T7)

T6. Responsible users for a task. (1) If a user has sufficient authorization level, she can set the list of responsible users for a task. A task can have many responsible users but it does not necessarily need to have any. (RS.T2)

4.2.7 Documents

D1. Adding documents. (1) A user can add documents. The document format is not specified, she can store documents of any format. (RS.D1)

D2. Reading documents. (1) A user can read documents that she is authorized to read. If a document is in such a format that SmartMeeting cannot show it, the user can copy it and use a proper software to read the document. (RS.D1)

D3. Subjects. (1) A user can link a document to existing subjects. Thus, the document has a context and someone browsing the subjects can see the documents that are linked to the subjects. (RS.D2)

D4. History information. (2) A user can browse the history information of the documents. When several meetings are held and a document changes and evolves, it gets a history. This history information may be important to users. (RS.D3)

D5. Browsing documents by subject. (2) A user can browse documents by subject. She can see which documents are linked to which subjects. (RS.D4)

5 External Interfaces

We have several external interfaces in our system. By using these interfaces it is possible to create external applications which communicate with the “core” server.

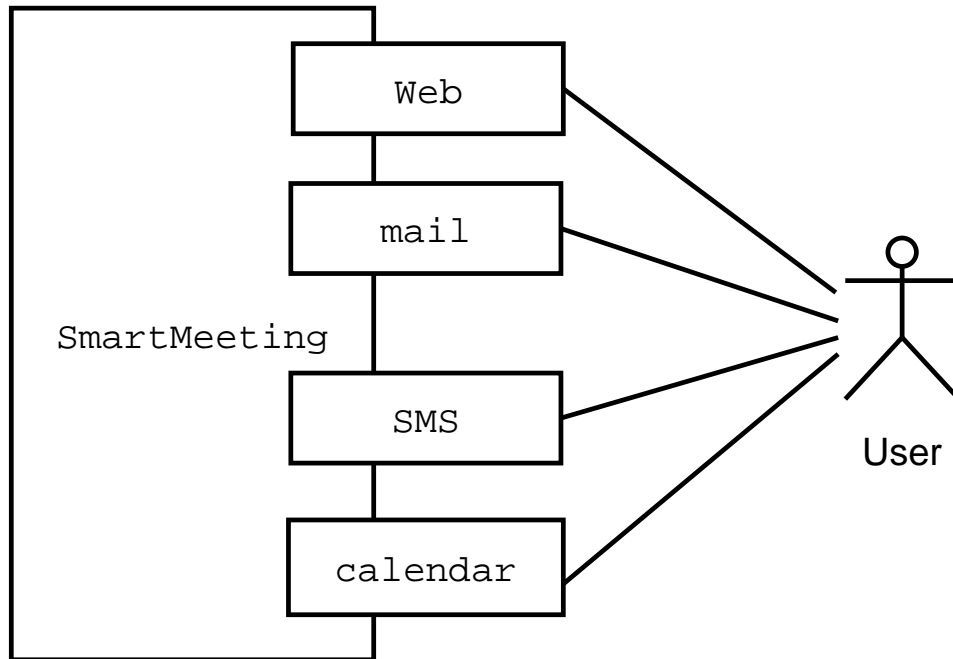


Figure 2: User interfaces

5.1 Hardware Interfaces

The system uses the printer to print meeting minutes. The minutes are stored in such a format that they print out clean. The reason for this is that the minutes are internal documents of SmartMeeting, they are not necessarily in any standard format.

Other documents can be printed using the printing features of the web browser in use. If the web browser is not capable of viewing a document, a software that can view the document is probably needed to print it.

SmartMeeting should be light enough that it could later be extended to be used with various PDA-devices. That would carry further the idea of ubile meetings. Remote participants could attend the meetings by chatting with a small hand-held device. With the same device he/she could view the meeting minutes, vote and so on.

5.2 Software Interfaces

SmartMeeting uses Ozone-db to store information. The web page for Ozone says: "The Ozone Database Project is a open initiative for the creation of an open source, Java based, object-oriented database management system." The current version of ozone is 1.0. The description of Ozone can be found

under the Ozone web page [4].

We will use Jakarta Tomcat version 4.0.1 as our servlet container. Tomcat will reside on the same server as SmartMeeting. Information about Tomcat can be found under the tomcat web page [6].

There will be several interfaces in SmartMeeting for external applications. The most important ones are those for e-mail, SMS and calendar. These may not, however, be implemented during the course due to lack of time and resources.

The SMS and e-mail interfaces are there to add usability for remote participants. SMS can be used to ask users for urgent information, for example when voting. E-mail can be used for information that is not that urgent.

It is important that the participants can agree on the date and time of future meetings. It would be a big task to add an internal calendar to SmartMeeting. That is why there should be an interface for an external calendar software.

This applies also to the SMS and e-mail interfaces. We are only going to provide the interfaces. This way SmartMeeting can be used with many different implementations and we do not dictate the choice.

5.3 Communication Interfaces

The system uses the Internet. It does not really matter whether it is a modem connection or a local area network, although from the standpoint of the efficiency of the system a faster network should rather be used.

The communication protocol will be HTTP. SmartMeeting will also support HTTPS should be used. That means using SSL for encrypting data. This is because the material of the meetings can be confidential.

The possible SMS functionality will be handled by an external application. We provide only the interfaces, the application can be chosen freely, as long as the interfaces are implemented.

Javamail could be used for sending and receiving e-mail. The protocol there is SMTP.

6 Other Features

6.1 Performance and Response Times

This is presented in chapter 4.1 of Requirement Specification. [7]

6.2 Usability, Recovery, Security, Protection

6.2.1 Usability

Usability of our prototype should be in the level that the system is usable besides the moments when an administrator is making great adjustments or the use of the system is prevented on purpose.

Strict time limits of unusability are not estimated or given.

6.2.2 Recovery

Restarting the server or client will not lose or destroy any data of the data storage.

Transactions are either committed or not committed at all.

The information lost in hardware breakings is possible but the risk is not taken care of by SmartMeeting project.

6.2.3 Security

This is presented in chapter 4.2 of Requirement Specification. [7]

6.2.4 Protection

User with no hostile purposes has to have userid and password to be able to use SmartMeeting. Access rights has to be specified when creating new information in SmartMeeting. The user is able to see only the information that she is authorized to see.

The transactions of SmartMeeting generate logs into logfiles which are useful when debugging systems function or malfunction.

Issue about encryption of communication is discussed in chapter 4.2 of Requirement Specification. [7]

6.3 Maintainability

SmartMeeting software is documented thoroughly and comprehensively which supports maintainability. The architecture is designed to be modular so that new modules should be easy to integrate in the system. Also some additional interfaces for possible future needs are supported.

Our light weight userinterface could be replaced with different ones in the future. Other userinterfaces than web-client can be inserted easily into system and the architecture is designed to support this need.

The most important decision chains are documented and this can ease the maintainability. When something has to be changed in the design of the software the old considered alternatives can be traversed easily.

6.4 Mobility and Portability

SmartMeeting will be tested in Windows and Linux environment. It will likely be portable also with other platforms because chosen implementation techniques and Java are supported quite widely.

Also mobile use is supported by possibly offering some SMS services and making architecture compatible also for integrating PDA-devices to the system in the future.

6.5 Operability

Logfiles are not removed by the system. Maybe some configuration has to be done when installing the system.

7 Design Limitations

7.1 Standards

We use Java, JDK 1.3, for developing SmartMeeting. Lower versions are not supported and it is not guaranteed that SmartMeeting will work with them.

7.2 Hardware limitations

SmartMeeting runs on a Linux server with 12 GB of hard disk space. It runs also on Windows NT 2000 server with at least the same amount of disk space. The processor is a 1 GHz AMD Athlon and there is 256 MB of memory.

7.3 Software Limitations

Jakarta Tomcat 4.0.1 is used as the servlet container [6]. The database, or data storage, is Ozone db 1.0 [4]. The kernel version of the Linux used is 2.2.16-22. Redhat 7.0 (Guinness) and Windows NT 2000 are used as the operating system.

8 Ideas for Further Development

Since meetings are common and for many people even every-day events, there are probably many features that a software such as SmartMeeting could have but that are not implemented on this course.

For example, there could be an interface for webcams. That would help the remote participants because they could then see what is going on at the assembly. SmartMeeting is not supposed to be a tool for video conferencing. Nevertheless, even that could be possible if needed.

One important thing to consider is the encryption of information. SSL-encrypted HTTP transfer will probably not be implemented on this course but it would be the next thing to do afterwards, at least as soon as anyone starts to use the system with confidential information.

References

- [1] T-76.115 Software Project
<http://www.soberit.hut.fi/T-76.115/index.html>
- [2] Product Modelling & Realisation Group
<http://www.cs.hut.fi/~pmrg/>
- [3] The description of the problem and the desired features by the customer
- [4] Ozone, the open source java ODBMS
<http://www.ozone-db.org/>
- [5] Novell's glossary of networking terms
<http://www.novell.com/info/glossary/glossary.html>
- [6] Jakarta Tomcat
<http://jakarta.apache.org/tomcat/index.html>
- [7] The requirements specification document for SmartMeeting
- [8] Discussion about ozone-db
<http://www.ozone-db.org/ozone-users/02-2001/msg00072.html>