



Messenger Design Description

Version 3.0

Messenger	Version: 3.0
Design Description	Date: 2004-12-17

Revision History

Date	Version	Description	Author
2004-11-24	1.0	Initial Draft	Jonas, Irfan, Zahid, Zdenek, Tihana, Marko
2004-12-10	2.0	Done the most of the re-doing of this doc. Some comments left.	Zahid
2004-12-11	2.1	I have done some formatting on the whole text plus making picture out of the OSD to be able to adjust the size of them to fit better to page. I have added some comments that need to be fixed.	Jonas
2004-12-11	2.2	Made some changes with the headings around heading 3	Zahid (via Jonas)
2004-12-15	2.3	Made new figure 2 in section 2.2. In section 2.3.1 added more info on shutdown program procedure. In section 2.3.3 added some explanation. In section 2.3.6 added some detailed description. Spellchecked whole document.	Tihana
2004-12-15	2.4	Changed commentary on the web pages. Add risk.	Zdenek
2004-12-15	2.5	Added pictures of predefined and search tab. Made some minor changes in the text (corrected grammatical errors in section 2.3).	Marko
2004-12-17	2.6	Grammatical check and correction. Changed some text. Copied introduction from req. doc. Rearranged positions and sizes of pictures in Web interface chapter.	Marko
2004-12-16	3.0	Did the last formatting of the doc.	Jonas

Table of Contents

1.	Introduction	3
1.1	Purpose of this document	3
1.2	Intended Audience	3
1.3	Scope	3
1.4	Definitions and acronyms	4
1.4.1	Acronyms and abbreviations	4
1.5	Related Documents	4
2.	External interfaces	5
2.1	Software Architecture	5
2.2	System Decomposition	6
2.3	Graphical User Interface for the Messenger client	8
2.3.1	Main program window	8
2.3.2	Tabs	9
2.3.3	Menu	12
2.3.4	Icons	12
2.3.5	Cursors	12
2.3.6	System Tray Menu	13
2.4	Web Page	14
2.4.1	Graphical User Interface	14
3.	Detail Software Design	21
3.1	Class Relationship diagram (CRD)	21
3.2	Object Sequence diagram	22
3.3	Database design description	26
3.4	Error handling	29
4.	Approvals	29

1. Introduction

Some communication between members in a software project leaves traces (such as emails or protocols from meetings) that make it possible to retrieve information later on. Other communication channels leave no trace. The task of this project is therefore to store relevant chat sessions in a way so that this information can be easily retrieved later. Though they sometimes do have an option to save a conversation, this option is limited to individual saving. That means the information is available only to the participant of this kind of conversation who actually saved the conversation. The other participants can't retrieve it unless they also saved it. This is however not formal enough, because every team member should have access to the same relevant information concerning the meetings, and this information should be secure. When each team member saves the conversation him/herself locally in plain text format, data manipulations are possible. Consider a case of some unscrupulous team member that is capable of changing the text of conversations. This hypothetical team member gives promises he can't deliver and then later changes the text of conversations in order to avoid responsibility for his actions. This hypothetical case can cause much harm to team work in distributed software development process. The main goal of this project is therefore to store relevant chat sessions in one place, regardless of the Instant messenger participant's use, so that all the participants can easily retrieve this information later. The information stored should be secure, because of the formal character of meetings.

1.1 Purpose of this document

The purpose of this document is to describe the design phase of the project. The document describes the system architecture, structural design, low detail design and the interfaces of the web and the messenger client. This document will be modified and updated throughout the lifespan of this project.

1.2 Intended Audience

Our intended audience is:

- Steering Group
- Project staff
- Class fellows and customer

1.3 Scope

The task of this project is to store relevant chat sessions in a way that this information can be easily retrieved later. Project outputs will be development of a small daemon or plug-in that runs locally and connects somehow to, or runs as a plug-in in any popular chat messenger program such as ICQ, AIM, MSN messenger and Yahoo messenger. Client program saves chat sessions in a central project database (e.g. MySQL or PostgreSQL database).

1.4 Definitions and acronyms

1.4.1 Acronyms and abbreviations

Acronym	Definitions
CRD	Class Relationship Diagram
OSD	Object Sequence Diagram
ID	Interaction Diagram
STD	State Transition Diagram
DB	Database

1.5 Related Documents

Document identity	Document title
Project Description	Project description document
Requirements definition	Requirement Definition document

2. External interfaces

2.1 Software Architecture

Messenger software with its web interface is actually three-tier architecture where the web pages are used to add new members/projects as well as search the chat sessions. Messenger software will connect to the centralized database residing on the server and save the conversations into related database tables. Whenever team members will start chatting the system will store the chat locally and then onto the server. Figure 1. explains the overall architecture and connectivity of the system to the server.

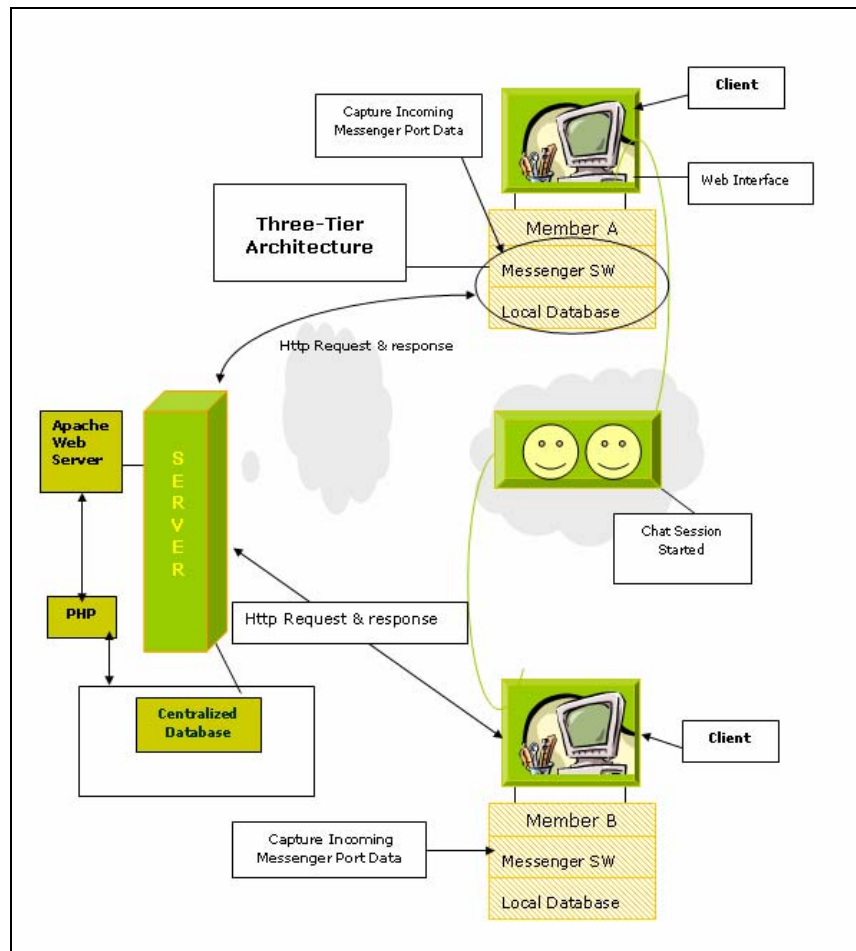


Figure 1: Architectural View

2.2 System Decomposition

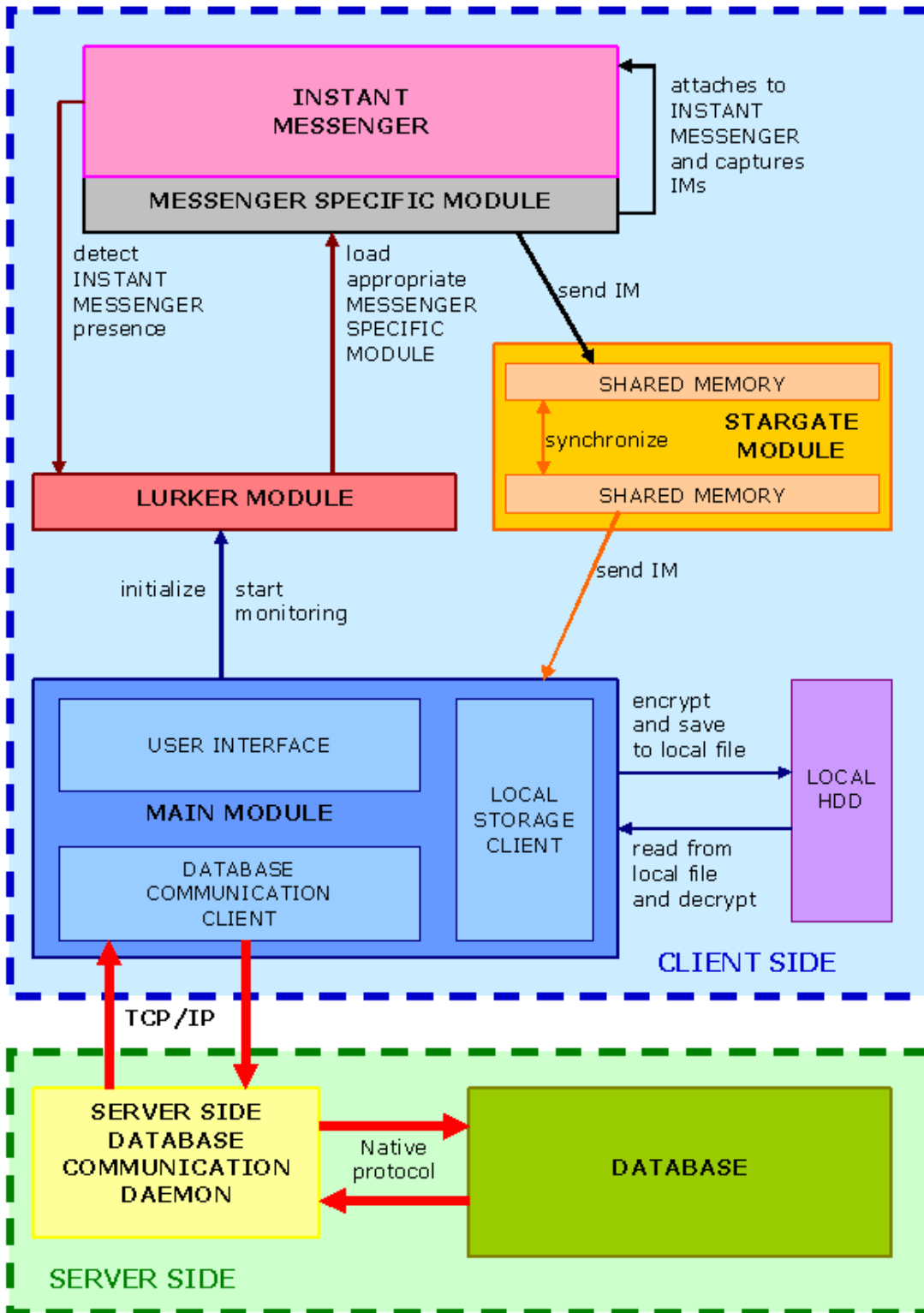


Figure 2: System decomposition

This schematic represents modules in the source code of our project, and the way these modules communicate. The module schematic has been inherited from the last year project. Green dotted line outlines light green area that represents server side of our application. Similarly, blue dotted line outlines light blue area that represents client side of our application. Each module is colored into its own color. Outline is a darker shade, and fill is somewhat lighter shade of that color. Thin arrows in outline color of a certain module represent tasks executed by that object. Significant parts of some modules are outlined in the same color as the module, but different shade of fills color. The thicker arrows in red color mark standard communication protocols. The type of the protocol is also given.

The MAIN module initializes LURKER module and initiates monitoring. LURKER module checks all running processes for new infiltration targets. When it finds one (new Messenger program) it loads defined dll file into target's process. It also keeps track of already infiltrated Instant Messenger programs. The MESSENGER SPECIFIC module when attached to an Instant messenger application (via hooks, or as a plug-in) gets messages from Instant messenger and sends them to STARGATE module via shared memory. The STARGATE module then transfers the message to MAIN module. The STARGATE module provides synchronization for accessing shared memory, and takes care that messages are not overwritten before module that they should be read by really reads them. MAIN module is concerned with communicating with the user through an USER INTERFACE, and also with Instant Message storage handling. DB COMMUNICATION client is in charge of communication with a database on a server. LOCAL STORAGE CLIENT takes care of storing the data locally. Storing is always done locally, and upon sending data to the database, the data is erased locally. User can determine which conversations he wants to be stored only locally. That way user can save all his conversations using one application; regardless if they are part of the project or private nature. The SERVER SIDE DATABASE COMMUNICATION DAEMON waits for a connection on a certain port, and then transfers all request to the database locally (as this two modules are on the same server).

2.3 Graphical User Interface for the Messenger client

When program is run the Main program window should appear by default. All buttons or any kind of other input fields, in any menu or window of this application, are grayish in color when inactive (disabled).

2.3.1 Main program window

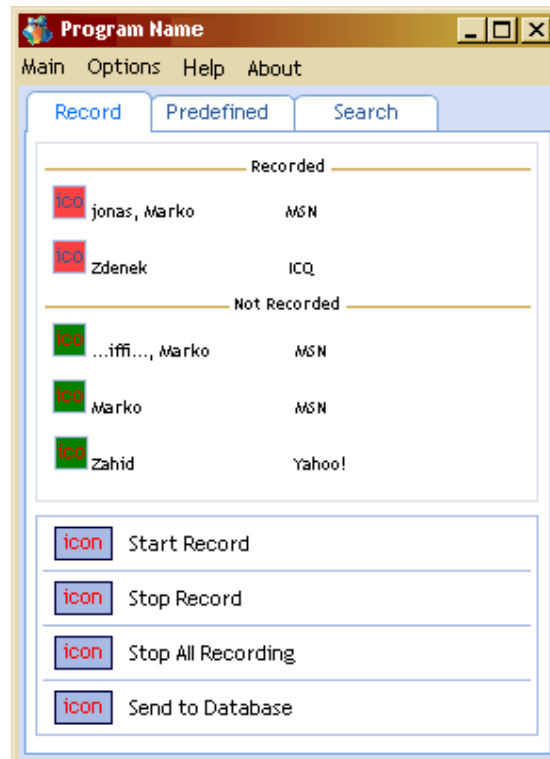


Figure 3. The Main window

Main window of the program GUI is composed of a Title Bar, Menu Bar and three Tabs in the main area: Record Tab, Predefined Tab and Search Tab. Title Bar contains Name of the program, and also three buttons: Minimize, Maximize/Restore Down and Close Button, which is windows standard. Only exception from windows standard function these perform is Close Window button, which does not close the program when clicked on the Main window. Instead it minimizes the program to System Tray, as it is the standard with Messenger Applications. There are two ways to close (shutdown) the program: through Shutdown option in the Main Menu, or through the Shutdown option in the system tray menu. The Main Menu is accessible through main program window, and the system tray menu by right-clicking on the system tray icon.

2.3.2 Tabs

Users most common action is summarized into three groups, and should therefore be implemented with Tabs in the main area of the Main window.

Record Tab is the first Tab, most common to use, and is therefore displayed by default and it is divided into two areas: Chats list and Record Functions.

Chat list is similar with ICQ Contact list. It shows Chats currently being recorded first in different color (red) with recording icon beside. Divided by the gray line from the upper part, below is list available chats which are currently not being recorded (in green with not-recording icon). The Chats in both parts of the list are selectable in the same way, as selection of the files in a directory in Windows Explorer application, which are windows standard.

Record Functions buttons: Record, Stop recording, Stop All and Send to Database Button. Buttons are wide, have both icon and text assigned to appropriate function. When one or more Chat/s from lower part of the Chat list (not-recording part) is selected Record Button becomes active. Pressing this button will start recording process for selected Chat/s. When one or more Chat/s from upper part of the Chat list (recording part) is selected Stop Recording Button becomes active (it is grayish while inactive). Pressing this button will initiate stop recording process for selected Chat/s.

This button is active when there are any Chats on a recording part of the Chat list (upper part). Pressing this button will initiate stop recording process for all Chat/s in the recording part of a Chat list. This button is always active. Pressing it will initiate sending locally stored chat sessions to the database on the server.

Record tab can be seen in figure 3. on the page 7.

Predefined Tab serves as editor of presets in recording process. Here you can add nicks from your Messenger Application Contact List, and preset the recording process for these contacts. It is also possible to choose between two options: recording conversations with multiple participants, when the contact is one of them, or recording "private" chats only (4 eyes only). This tab is also divided into two parts. First part is a list of contacts that already have the recording process preset. The other part consists of three Buttons: Add, Edit and Remove Button.

List of contacts that already have the recording process preset, is a list type already familiar from Windows environment. It has three fields: Nickname/Contact, Messenger, Description and private ("4 eyes only") field. Nickname/Contact and Messenger fields are automatically retrieved from the messenger when adding contacts. Description field can be changed by the user (Edit button), but is left blank by default. Private ("4 eyes only") is a checkbox field, which can be checked or unchecked, but is not checked by default.

Add button is always available. Pressing this button opens a new window with a checkbox list of currently active Messenger Applications that are also supported by the program. Below that simple list there is a Retrieve Button, which is active, when at least one of the Messenger Application on the list, is selected. There is also a Cancel button. Pressing the Retrieve button results in a new window with a checkbox list of contacts/messenger pairs retrieved from selected Messenger/s that are not already on a preset list. This window has also two buttons below that list: Add and Cancel. Add button is inactive until at least one contact from the list is selected. When pressed this button initiates the function which adds selected contact/messenger pairs to the preset list on the Predefined tab in the main window.

Edit button is available when a contact from the preset list is selected. Pressing this button brings out a form, where all the fields of the preset list can be changed.

Remove button is available when a contact from the preset list is selected. Pressing this button removes selected contact/s from the preset list.

Predefined tab described above can be seen in the figure 4. below.

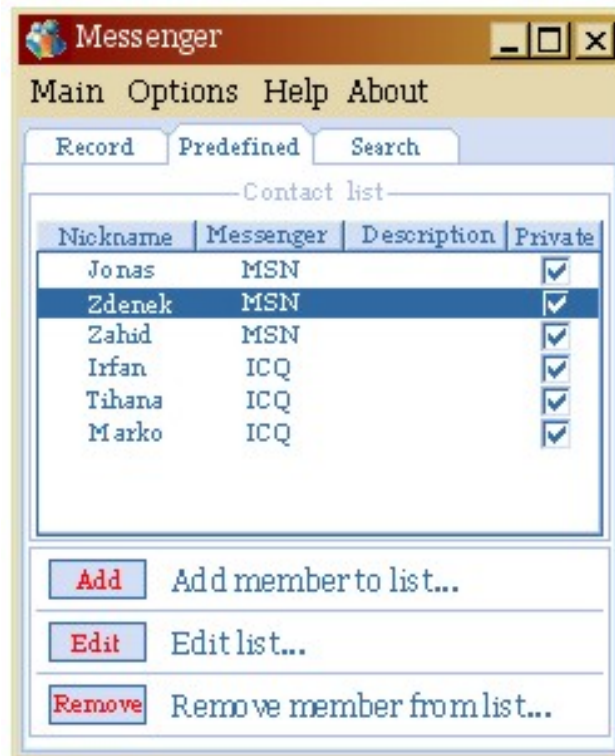


Figure 4. Predefined tab

Search tab contains a search form, used to retrieve information about stored conversations. It has two parts. On the top part there are two checkboxes: Search Local File and Search Database. These two are both checked by default (all combinations are possible). When neither of these is checked the second part of this tab, except for Cancel button, is disabled.

The second part is a form where user can enter search parameters. These are: keyword/s field (textbox), username/nickname/contact field (textbox), time period input, project drop down list, Search and Cancel buttons below. Time period input will probably be a two calendars input for a date, and simple input for hours and minutes (small textbox).

When user clicks on the Search button, new window appears containing Search results. On that window there is a list of chats found (connected to multiple files in which chat is stored - this is not viewable, but is known to the program). Also there are View and Cancel buttons. User can select one or more chat/s from the list.

When the View button is pressed, program decrypts files in which selected chat/s is/are stored, and it stores each chat into a temporary file (decrypted and merged files for each chat). Then an external viewer, which opens them, is called.

Search tab described above can be seen in the figure 5. below.



Figure 5. Search tab

2.3.3 Menu

Menu Contains Shutdown function which shutdowns the program. Opens the Options Window. Options Window has two tabs: General and Database Settings. All advanced options such as external viewer location (Notepad by default), conversation storage folder, ... Database Settings tab holds all options regarding the database like: username, password, remember (encrypted) password checkbox, server location, etc. Clicking on Help menu in this version will simply open the user manual in a default text viewer. This could be changed in later versions, in a way that clicking on this menu would open a standard Windows help window.

Standard About-Dialog is a simple message box which holds the information about the version of a program (that is version number and/or name) and about its authors (that is name of company or list of programmers).

2.3.4 Icons

Icon in system tray should change color:

- If there are some recording in progress
- If there aren't any recording in progress, but there are chats available for record
- If there aren't any chats in progress, but there is at least one Messenger

Application active

- If there aren't any Messenger Application active

2.3.5 Cursors

When some operation in progress, cursor should change to standard windows working cursor (rotating sand clock).

2.3.6 System Tray Menu

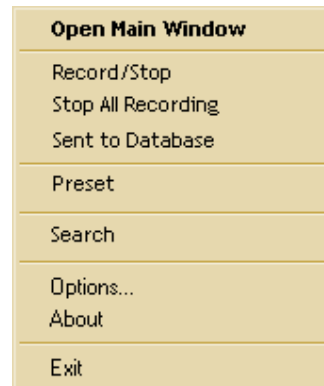


Figure 6. The System tray menu

The system tray menu is accessed by right-clicking the system tray icon. The menu floats above system tray, and its options can be accessed through a left-click. Each of the options either initiates some process or opens a relevant dialog (window). The system tray menu contains following menu options:

- Open Main program window
 - opens the Main program window
- Record/Stop
 - opens the Main program window with Record tab in front
- Stop All Recording
 - brings all currently active recordings to a stop
 - program behaves in the same way, as the "Stop All Recording" button in the Record Tab of the Main program window is pressed
 - doesn't open the Main program window
- Preset
 - opens the Main program window with Predefined tab in front
- Search
 - opens the Main program window with Search tab in front
- Options
 - opens the Options Window
- Help
 - opens the User Manual in the default text editor
- About
 - opens the Standard About Dialog (message box with program version and authors)
- Shutdown
 - initiates program shutdown

All of these options can be accessed in some other way, through the Main program window.

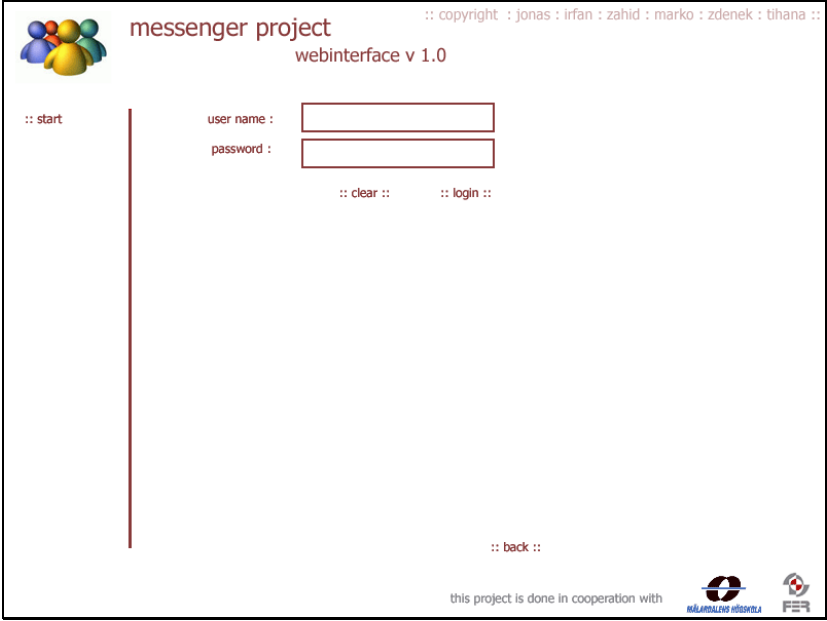
2.4 Web Page

Customer wants a homepage with information about saved conversations related to the projects and engine to search within the conversations. This document specifies the design of the product.

2.4.1 Graphical User Interface

The actor has to log in. After log in actor can search for conversation s/he would like to find. Administrator can also add new members and new projects. Super administrator moreover is responsible for deleting members and projects.

Figure 7. depicts first page visible to user. This page represents log in page. Actor has to type his/her username and password. Then click on **login**. The page will look on the Internet like in the picture:



The screenshot shows a web browser window displaying the login page for the 'messenger project'. The page title is 'messenger project webinterface v 1.0'. The page content includes a logo on the left, a 'start' link, and a login form with 'user name' and 'password' fields. Below the fields are 'clear' and 'login' buttons. At the bottom, there is a 'back' button and a footer with logos and text: 'this project is done in cooperation with' followed by logos for 'MILANSKILOVSKI' and 'FER'. The copyright notice at the top right reads: ':: copyright : jonas : irfan : zahid : marko : zdenek : tihana ::'.

Figure 7. The login page

Figure 8. depicts change password page, where actor can change his/her password. Actor is obliged to change password after first log in. New password has to be from four to sixteen characters long (cannot be same as login name). To save new password to database user has to click to **change**.

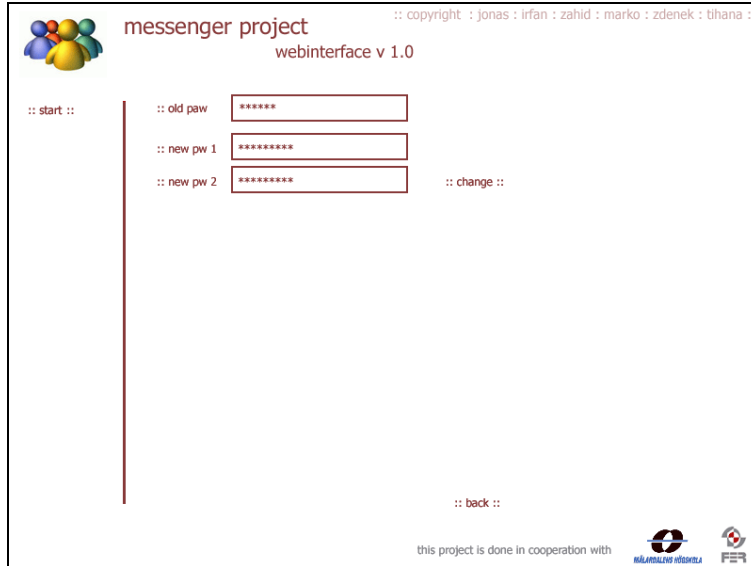


Figure 8. Change personal info page

Personal info page contains all relevant data about an actor. Actor can see all the data as it is shown on the figure 9. He can also change and update data e.g. email, telephone number, password etc. Actor is not allowed to change the project s/he is working in.

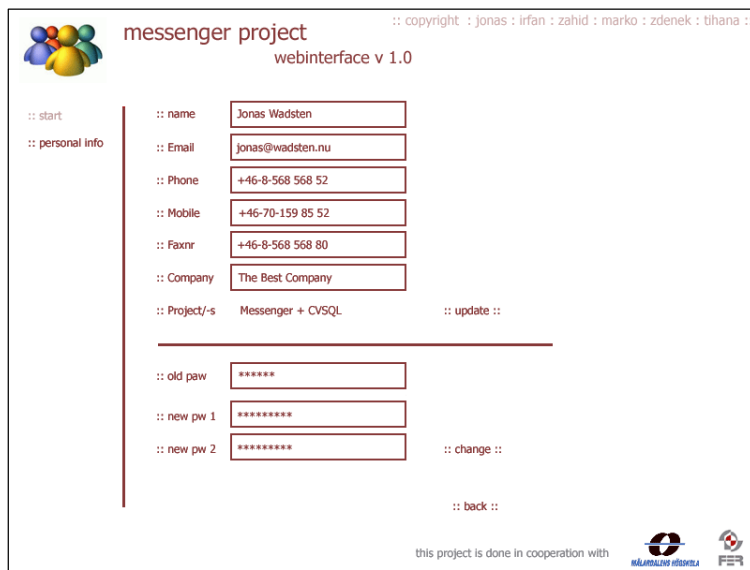


Figure 9. Personal Info page

Search page is the core of all web pages. You can see its design in figure 10. Actor can search for a conversation by criteria such as project name, project member keyword or dates of conversations. For easier search project name and project member fields are combo boxes. Also notice that format of dates is set. As you can see, on the left side of the figure there is a hyperlink that refers to personal data.

messenger project
webinterface v 1.0

copyright : jonas : irfan : zahid : marko : zdenek : tihana

start ::

personal info ::

project project a

member member names

keyword any keyword/-s you want...

date from 2004-11-05 (yyyy-mm-dd)

date to 2004-11-28 (yyyy-mm-dd)

search ::

logout ::

back ::

this project is done in cooperation with

MALAYSIAN BUREAU OF STANDARDS FER

Figure 10. Search page

After running a search, user is transferred to a search result page. Design of this page is shown in figure 11. A line divides selected conversations. To see the actual conversation, actor was looking for, s/he has to click to the date of the conversation.

messenger project
webinterface v 1.0

copyright : jonas : irfan : zahid : marko : zdenek : tihana

start ::

personal info ::

search result

project name project b

project members göran, igor

date of chat 2004-11-10
2004-11-14

project name project b

project members joans, zihad

date of chat 2004-10-23

logout ::

back ::

this project is done in cooperation with

MALAYSIAN BUREAU OF STANDARDS FER

Figure 11. Result page

Selected conversation is displayed like in figure 12. As you can see on the figure, lines separate details about the conversation. Details displayed are: name of the project, members who were in a conversation and date when conversation was saved. On the left side of the page there are start, personal info and logout links. Clicking them actor can either return to the search page, change the personal data or logout. In the figure it is visible that the members' lines are colored differently. This is done to make it easy for actor to recognize different peoples' chats.

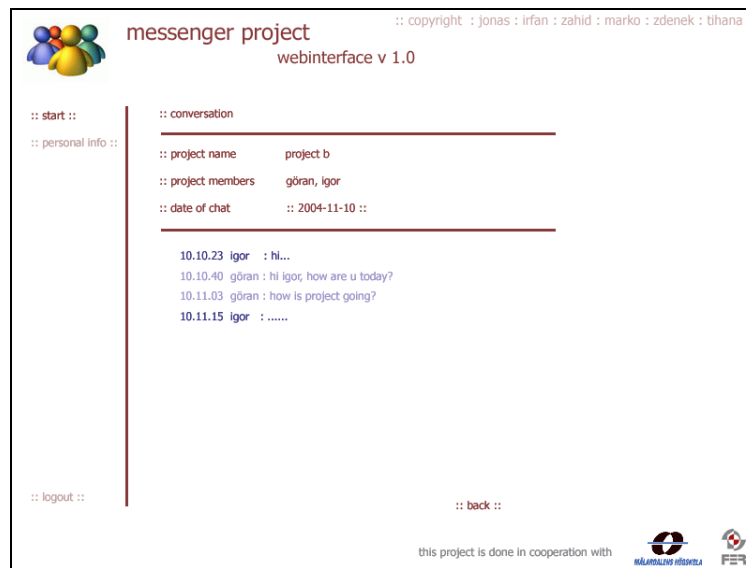


Figure 12. Conversation page

In figure 13. you can see how the left menu, when logged-in as administrator, differs from user's menu. Administrator can add and edit member and also add and edit project. Figure 13 shows add member when administrator adds new members.

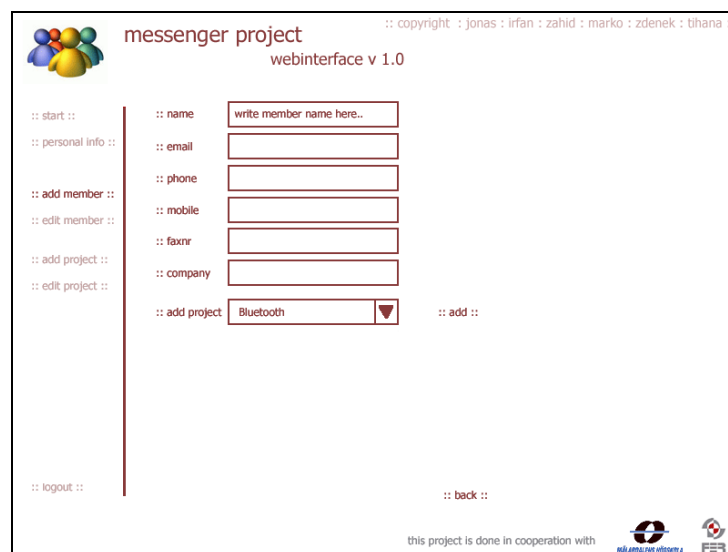


Figure 13. Add member page for administrator

Edit member page, which is shown in the figure 14, gives administrator rights to change the details about a user. Administrator can also, on this page, add a user to another project.

Figure 14. Edit member page for administrator

Figure 15 shows how administrator sees add project page. On this page administrator adds a project to the database with all the details about the project such as start and end date and the members of the project. To add more members administrator has to press CTRL + click on the name of the member that will be involved in the project. When all the details are chosen administrator saves project to the database by clicking on add.

Figure 15. Add project page for administrator

Administrator uses edit project page to change details about project or to add and remove members from project. This is depicted in figure 16. Administrator adds or removes members from project by holding CTRL and clicking on member's names. To update information s/he needs to press update when finished.

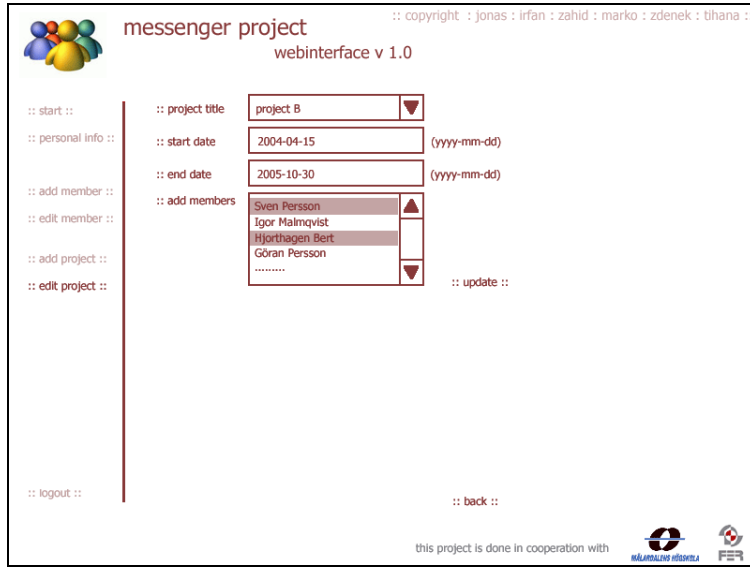


Figure 16. Edit project page for administrator

Difference between result page when logged in as user or as administrator and super administrator exists only in left panel. Administrator and super administrator can also choose from add and edit possibilities as it is displayed in the figure 17. Conversation page also differs only in the left side.

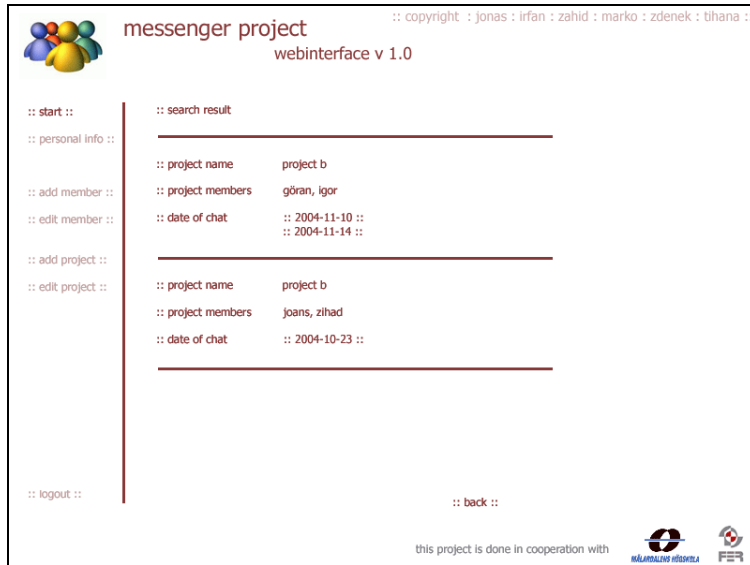


Figure 17. Result page for administrator, super administrator

Web pages differ when logged in as administrator and super administrator. Difference is visible in member page where only super administrator can delete members and all their records. As you can see in the figure 18 next to each project name delete link is placed. When super administrator clicks to this link member's account is locked. This means that only his/her name is deleted from the project table. Conversations, in which user participated, remain in database.

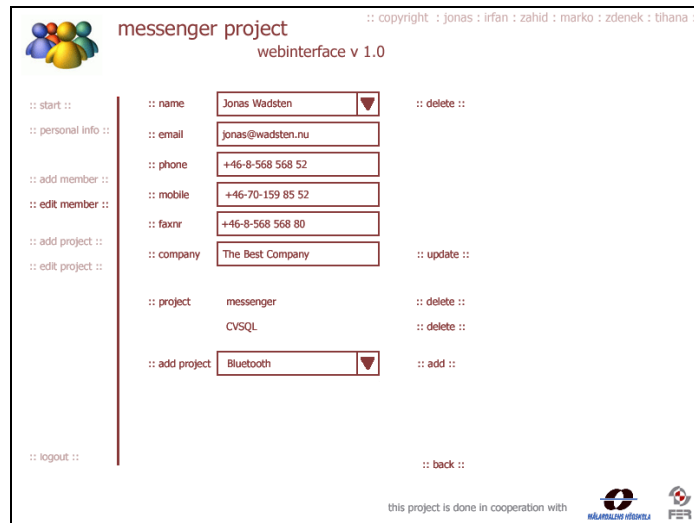


Figure 18. Edit member page for super administrator

In edit project page, depicted in figure 19, super administrator can delete the project. S/He can, from combo box, choose name of the project, edit details or lock the project with all its records and members. To lock the project super administrator presses delete, which means that no more conversations are saved to this project. Members are also locked, which means that the database cannot be changed any more. To edit details super administrator presses update hyperlink. S/he can move members from one project to another, same as administrator.

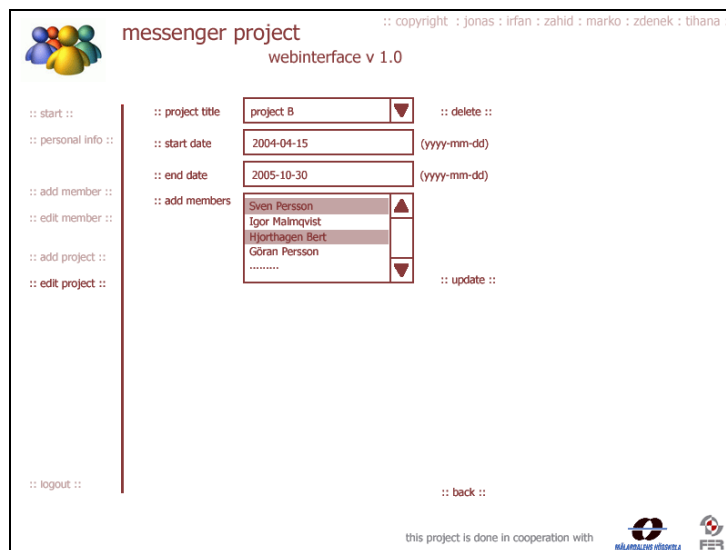


Figure 19. Edit project page for super administrator

3. Detail Software Design

3.1 Class Relationship diagram (CRD)

This part of the document describes the static behavior of the system. It explain user defined classes and the responsibility and relationships between classes.

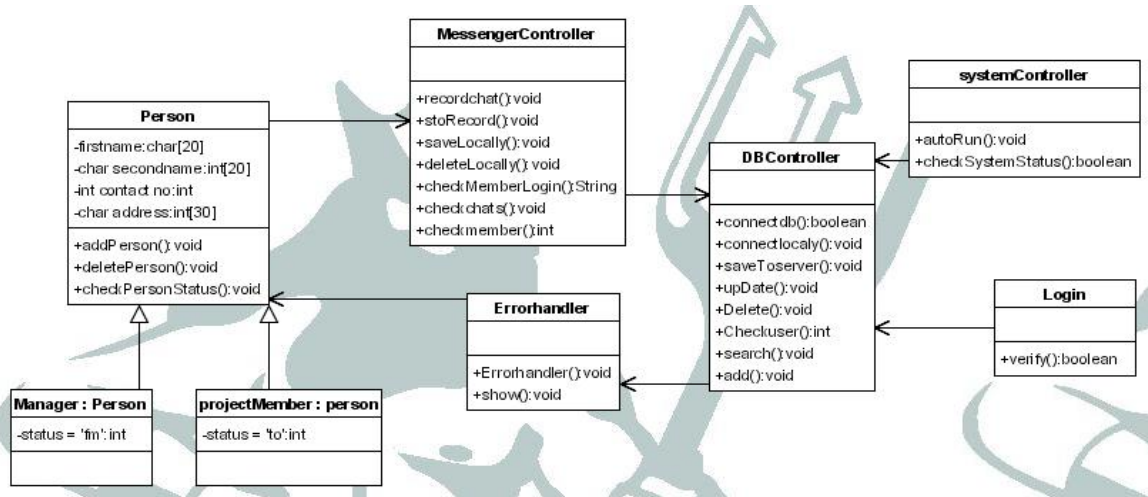


Figure 20. The Class Relationship diagram

3.2 Object Sequence diagram

Object sequence diagram is a type of interaction diagram. This model shows the dynamic behavior of the system. Main sequence diagrams of messenger client system are as follows.

The figure 21 describes the object sequence diagram of the system start up. When any machine containing the Messenger client starts, system automatically starts by calling the autoRun() method of the system class.

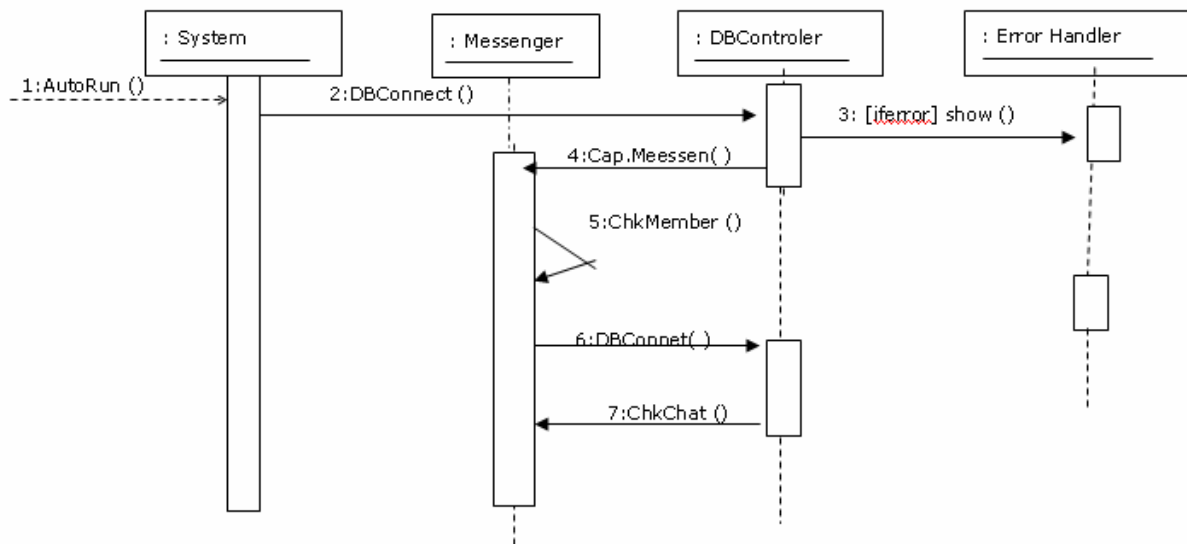


Figure 21. OSD OF System Start-up

Figure 22 describes object sequence diagram of the login session use case. To login user name and password as well as connection to database are required. If all these prerequisites are met data verification is possible. If data is wrong error is displayed otherwise login successfully.

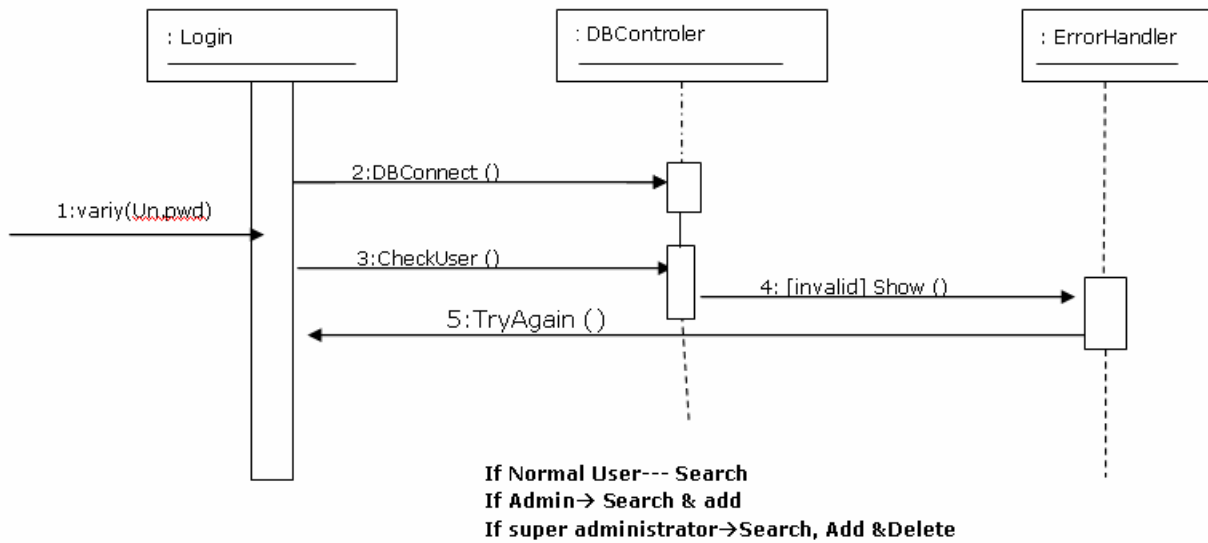


Figure 22. OSD of Login Session

Figure 23 describes object sequence diagram of multiple scenarios of the system. It includes addition, deletion and search of project members and project information by the administrator.

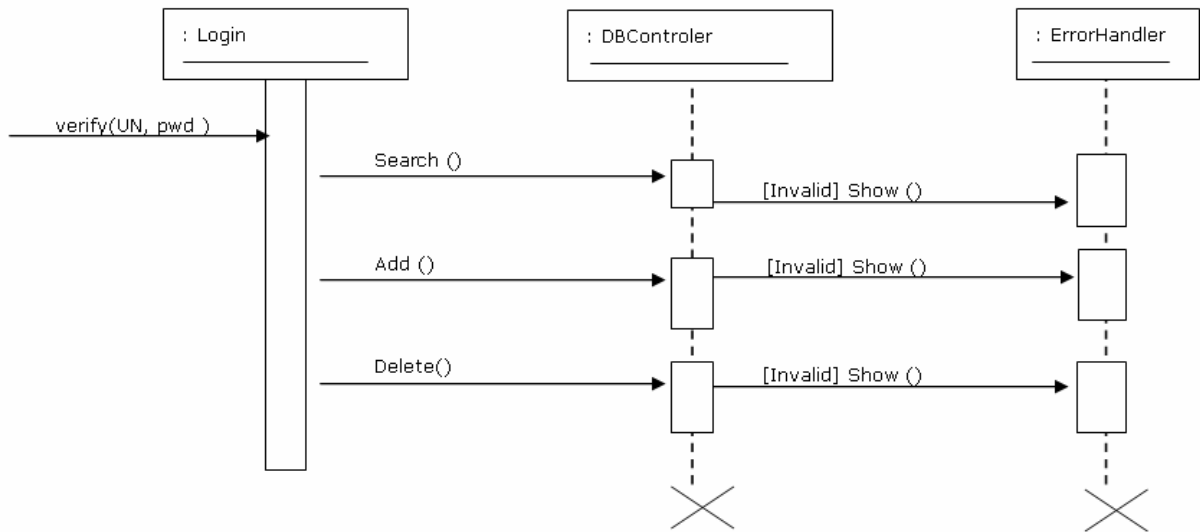


Figure 23. OSD of ADD I/M, Delete I/M and Search

Figure 24 is representation of chat scenario. Figure depicts two project members chatting. If this is the case than chat is saved automatically. If there is one guest chatting then system asks to add guest or not. Depending of the answer chat is started.

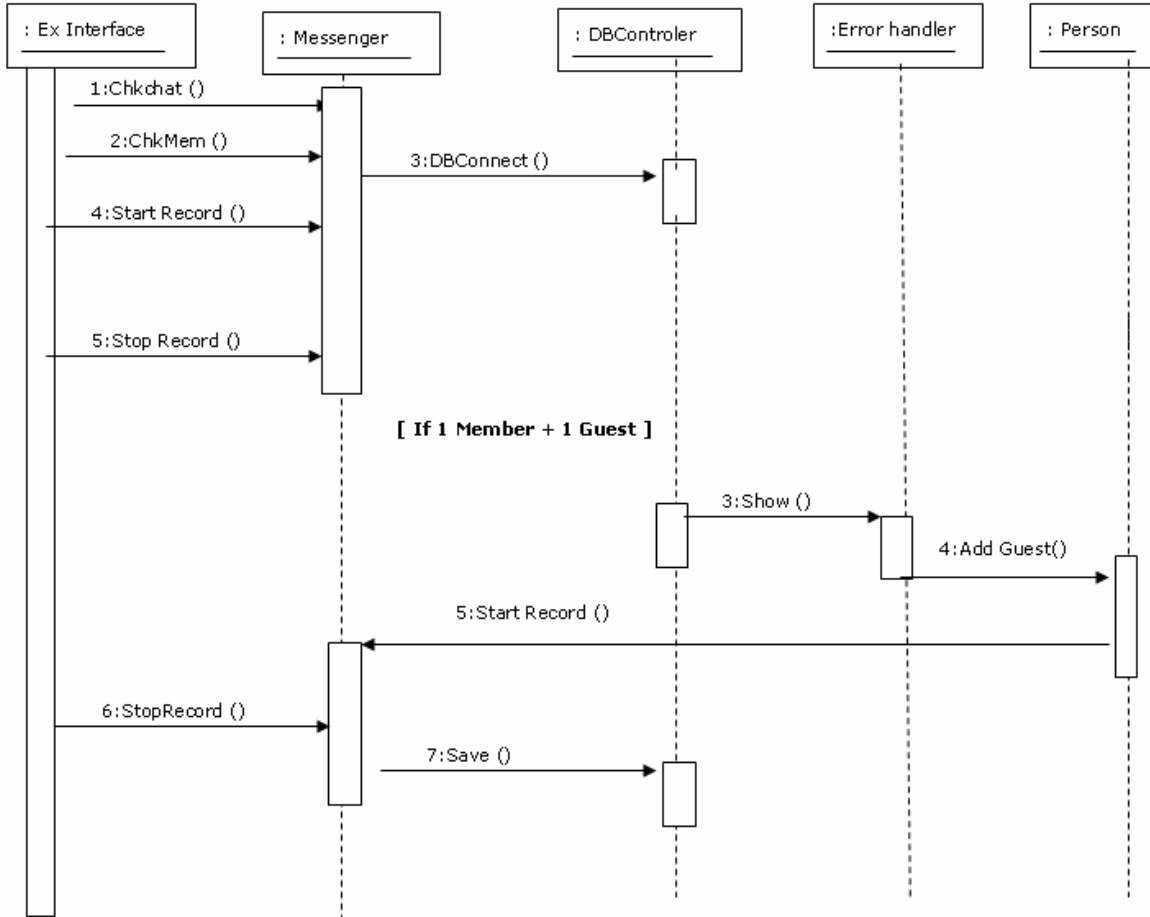


Figure 24. OSD for the Chatsessions

3.3 Database design description

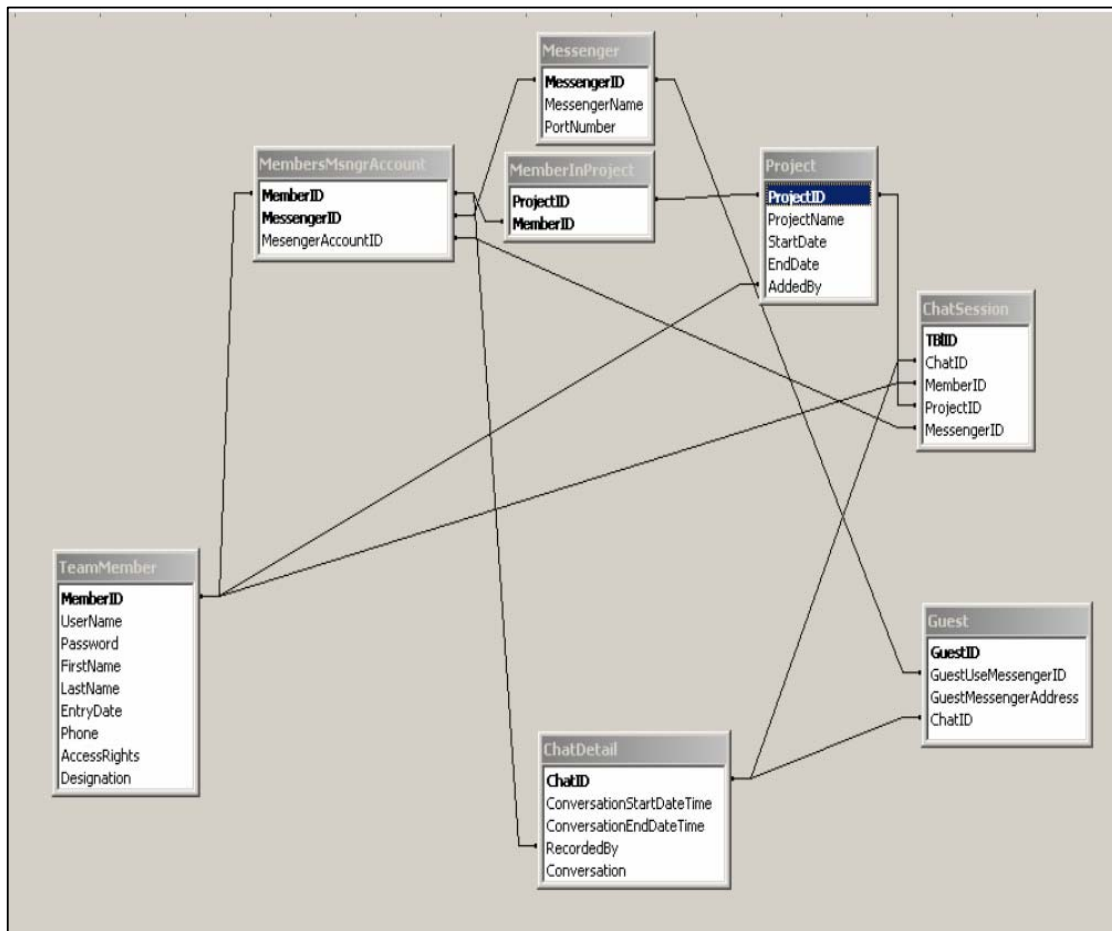


Figure 25: Entity Relationship Diagram

Entities/Tables, Description:

TeamMember: keeps the record of team members as well as keeps the record of administration member who has access rights to enter the website in order to view the conversation.

Project: The project id's and name will be stored in this table.

Relationship between: Project and Admin. One admin member can add many projects.

Messenger: Stores the information of different messengers. For example, Yahoo, Msn, Miranda messengers etc.

ChatDetail: Stores different conversation details like start and end time of conversation and conversation itself etc.

Messenger	Version: 3.0
Design Description	Date: 2004-12-17

Guest: We have separated the guest information storage from team member information. It is better to make a new table for guests for better efficiency and less disk usage. It is better to assume that guests will not join the chats so often.

ChatSession: Stores the conversation of at least two members/one guest if talking together.

Relationship among: MembersMsngAccount, TeamMember, Project, ChatDetail, and ChatSession.

Although we don't need to store members actual Id from TeamMember. The need is to store the Messenger id which he or she is using during chat. But we will store it so that it would be very easy in order to search the chats by member's UserName. For example. Irfan is using nice_pak@hotmail.com and zahid is using zahid@yahoo.com. When the system will search these Messenger ids in MemberMsngAccount table, it will be easy to get the actual Id of team member in order to proceed further. But what happens if we start to search chats of Username "irfan"? The query will get the MemberId from TeamMember table and search out the chats of member "irfan" in ChatSession and ChatDetail tables.

MemberInProject: Keeps the record, "which member(s) is involved in which project?".

Relationship between: Member and project

MemberMsngAccount: One member may have different messenger Ids. For example, Irfan has nice_pak@hotmail.com, irfan@icq.com. It means the member can use different messenger for conversation. There isn't any fixed messenger for conversation.

Relationship between: Member and Messenger with MessengerAccountID attribute.

Create Table commands and Attributes:

```
CREATE TABLE `Project` (
  `ProjectID` INT( 3 ) NOT NULL AUTO_INCREMENT ,
  `ProjectName` VARCHAR( 30 ) NOT NULL ,
  `StartDate` DATE NOT NULL ,
  `EndDate` DATE NOT NULL ,
  `AddedBy` INT( 2 ) NOT NULL ,
  PRIMARY KEY ( `ProjectID` )
) COMMENT = 'Keeps project information';
```

```
CREATE TABLE `TeamMember` (
  `MemberID` INT( 3 ) NOT NULL AUTO_INCREMENT ,
  `UserName` VARCHAR( 20 ) NOT NULL ,
  `Password` VARCHAR( 20 ) NOT NULL ,
  `FirstName` VARCHAR( 20 ) NOT NULL ,
  `LastName` VARCHAR( 20 ) NOT NULL ,
  `EntryDate` DATE NOT NULL ,
  `Phone` INT( 14 ) NOT NULL ,
  `AccessRights` VARCHAR( 20 ) NOT NULL ,
  `Designation` VARCHAR( 20 ) NOT NULL ,
  PRIMARY KEY ( `MemberID` )
);
```

```
CREATE TABLE `Messenger` (
  `MessengerID` INT( 2 ) NOT NULL AUTO_INCREMENT ,
  `MessengerName` VARCHAR( 20 ) NOT NULL ,
  `PortNumber` INT( 4 ) NOT NULL ,
  PRIMARY KEY ( `MessengerID` )
) COMMENT = 'Keeps chat messengers information';

CREATE TABLE `MembersMsngrAccount` (
  `MemberID` INT( 3 ) NOT NULL ,
  `MessengerID` INT( 2 ) NOT NULL ,
  `MesengerAccountID` VARCHAR( 30 ) NOT NULL
) COMMENT = 'Keeps record of messengers Ids(which member has which messenger id(s)) ';

CREATE TABLE `MemberInProject` (
  `ProjectID` INT( 3 ) NOT NULL ,
  `MemberID` INT( 3 ) NOT NULL
) COMMENT = 'Keeps team members record in which project they are working';

CREATE TABLE `Guest` (
  `GuestID` INT( 3 ) NOT NULL AUTO_INCREMENT ,
  `GuestUseMessengerID` INT( 2 ) NOT NULL ,
  `GuestMessengerAddress` VARCHAR( 30 ) NOT NULL ,
  `ChatID` INT( 6 ) NOT NULL ,
  PRIMARY KEY ( `GuestID` )
) COMMENT = 'Keeps guests information participating in chatting';

CREATE TABLE `FeedBack` (
  `SenderID` INT( 4 ) NOT NULL AUTO_INCREMENT ,
  `FirstName` VARCHAR( 20 ) NOT NULL ,
  `LastName` VARCHAR( 20 ) NOT NULL ,
  `Email` VARCHAR( 30 ) NOT NULL ,
  `CurrentDate` DATE NOT NULL ,
  `Comments` VARCHAR( 255 ) NOT NULL ,
  PRIMARY KEY ( `SenderID` )
) COMMENT = 'saves feedback from users';

CREATE TABLE `ChatSession` (
  `TBIID` INT( 6 ) NOT NULL AUTO_INCREMENT ,
  `ChatID` INT( 6 ) NOT NULL ,
  `MemberID` INT( 3 ) NOT NULL ,
  `ProjectID` INT( 3 ) NOT NULL ,
  `MessengerID` INT( 2 ) NOT NULL ,
  PRIMARY KEY ( `TBIID` )
) COMMENT = 'Keeps chat sessions';

CREATE TABLE `ChatDetail` (
  `ChatID` INT( 6 ) NOT NULL AUTO_INCREMENT ,
  `ConversationStartDateTime` DATETIME NOT NULL ,
  `ConversationEndDateTime` DATETIME NOT NULL ,
  `RecordedBy` INT( 3 ) NOT NULL ,
  `Conversation` TEXT NOT NULL ,
  PRIMARY KEY ( `ChatID` )
) COMMENT = 'Saves chat details and conversation';
```

3.4 Error handling

Error	Action
Database disconnected	Save the conversation encrypted locally, restore database connection when possible, send the conversation saved locally and that are finished. Start record locally the chat sessions.
System crash	Restore database connection at restart, send the conversation saved locally and that are finished. Start record locally the chat sessions.
Client can not connect to database	Save the conversations locally in encrypted. On next successful connection it will upload conversation and delete it locally.
Client crashes during chat session	Restore database connection at restart, send the conversation saved locally and that are finished. Start record locally the chat session.
Messenger program crash during recording	Treat it is as a finished chat-session and save it to central database, if the sessions starts-up again with the same users, creates a new chat-session with the same chat-id.
During chat-session user accidentally close the chat-window	Treat it is as a finished chat-session and save it to central database, if the sessions starts-up again with the same users, creates a new chat-session with the same chat-id.
Password lost or wrong set	Sent email to administrator that can check the database and see actor's details. The administrator will tell to actor or sent it to him/her. Actor should change the password after next login.

4. Approvals

Name	Title	Date (yyyy-mm-dd)	Signature