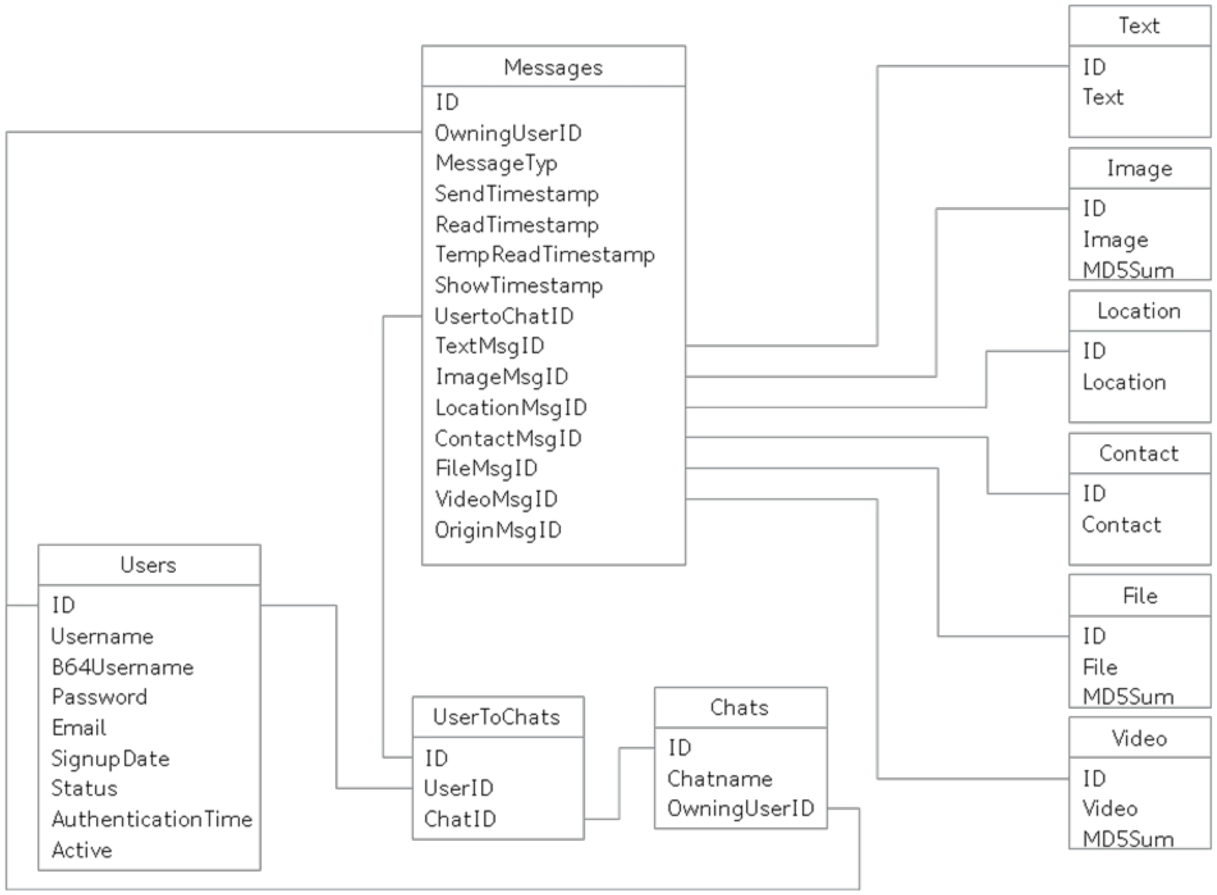
Database Schema



Tables:

Users, this table has the information of alle registered users of the service.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Username` varchar(45) NOT NULL,
`B64Username` varchar(55) NOT NULL,
`Password` char(64) NOT NULL,
`Email` varchar(100) NOT NULL,
`SignupDate` bigint(20) NOT NULL DEFAULT '0',
`Status` tinyint(3) DEFAULT NULL,
`AuthenticationTime` bigint(20) NOT NULL DEFAULT '0',
`Active` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'User activ
set by admin',
PRIMARY KEY (`ID`)
```

- ID is the unique ID of the user in the database.
- Username is the clear name of the user.
- B64Username is the Base64 encoded Username, we do not need to decode the username for comparison.
- Password is the user chosen password, the frontend must crypt the password, the backend is just storing it.
- Email is an email address, the server checks if it is a correct format, up to now the email address is not used, possible scenario in the future, the

user gets an activation code by email and can activate himself with a not yet implemented function.

- Status is not used up to now.
- AuthenticationTime is the unix timestamp when the user has made his last action against the server with an valid authentication.
- Active is check at every authentication, if it is 0 (false) the user cannot do any actions on the server, is for admins to control the users at the server.

Chats, here are all Chatrooms listed on the server.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Chatname` varchar(50) NOT NULL,
`OwningUserID` int(10) unsigned NOT NULL,
PRIMARY KEY (`ID`)
```

- ID is the unique ID of the chat in the database.
- Chatname is the clear name of the chatroom
- OwningUserID is the ID of the User who created the chat, only these person can add or remove user and can delete the chat.

UserToChats, here are the combinations of which user is in which chatroom and which chatroom has which users stored (n:m)

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`UserID` int(10) unsigned NOT NULL,
`ChatID` int(10) unsigned NOT NULL,
PRIMARY KEY (`ID`)
```

- ID is the unique ID of the combination user and chat.
- UserID is the reference to the User in the Users table.
- ChatID is the reference to the Chat in the Chats table.

Messages, here are the messages stored for all chats with references to the content of the message

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`OwningUserID` int(10) unsigned NOT NULL,
`MessageTyp` varchar(10) NOT NULL,
`SendTimestamp` bigint(20) NOT NULL,
`ReadTimestamp` bigint(20) NOT NULL DEFAULT '0',
`TempReadTimestamp` bigint(20) NOT NULL DEFAULT '0',
`ShowTimestamp` bigint(20) NOT NULL DEFAULT '0',
`UsertoChatID` int(10) unsigned NOT NULL,
`TextMsgID` int(10) unsigned DEFAULT NULL,
`ImageMsgID` int(10) unsigned DEFAULT NULL,
`LocationMsgID` int(10) unsigned DEFAULT NULL,
`ContactMsgID` int(10) unsigned DEFAULT NULL,
`FileMsgID` int(10) unsigned DEFAULT NULL,
`VideoMsgID` int(10) unsigned DEFAULT NULL,
`OriginMsgID` int(10) unsigned DEFAULT NULL,
PRIMARY KEY (`ID`)
```

- ID is the unique ID of each message stored at the server.
- OwningUserID is the ID of the user who sent the message
- MessageTyp is a text to identify which content need to be loaded, TEXT, IMAGE, …
- SendTimestamp, the unix timestamp when the message was created at the server.
- ReadTimestamp, the unix timestamp when the client has acknowledge the download.
- TempReadtimestamp, a temporary unix timestamp, download ok but not acknowledged, if the message is acknowledged the readtimestamp is set, see communications-diagram.
- ShowTimestamp, the unix timestamp when the user has seen the message (normal enter the chatroom in the frontend.
- UsertoChatID, the ID to which user in which chatroom the message belongs.
- TextMsgID, the ID of the text content message
- ImageMsgID, the ID of the image content message
- LocationMsgID, the ID of the location content message
- ContactMsgID, the ID of the contact content message
- FileMsgID, the ID of the file content message
- VideoMsgID, the ID of the video content message
- OriginMsgID, the ID of the origin message to group the status of send, read and show.

Text, here are all texts stored for each text message.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Text` varchar(10000) NOT NULL,
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the text message, independent of how often the message is used in different chats.
- Text, the text itself. It is base64 encoded text.

Image, here are the filenames and MD5sum stored for each image.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Image` varchar(1024) NOT NULL,
`MD5Sum` varchar(32) NOT NULL DEFAULT '0',
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the image message, independent of how often the message is used in different chats.
- Image, the filename of the image on disk.

- MD5Sum, the MD5 hashcode of the file for acknowledge and checks of the frontend to avoid duplicate up- and downloads

Location, this is a dummy table and not used up to now..

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Location` varchar(1024) NOT NULL,
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the location message, independent of how often the message is used in different chats.
- Location, here should be lat and Lon, but not now.

Contact, this is a dummy table and not used up to now..

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Contact` varchar(1024) NOT NULL,
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the contact message, independent of how often the message is used in different chats.
- Contact, no idea how to manage contact information for different frontends, XML might be a good solution, but not now.

File, here should be the filenames and MD5sum stored for each file, but not implemented yet.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`File` varchar(1024) NOT NULL,
`MD5Sum` varchar(32) NOT NULL DEFAULT '0',
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the file message, independent of how often the message is used in different chats.
- File, the filename of the file on disk.
- MD5Sum, the MD5 hashcode of the file for acknowledge and checks of the frontend to avoid duplicate up- and downloads

Video, here are the filenames and MD5sum stored for each video.

```
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Video` varchar(1024) NOT NULL,
`MD5Sum` varchar(32) NOT NULL DEFAULT '0',
PRIMARY KEY (`ID`)
```

- ID, the unique ID of the video message, independent of how often the message is used in different chats.
- Video, the filename of the video on disk.

- MD5Sum, the MD5 hashcode of the file for acknowledge and checks of the frontend to avoid duplicate up- and downloads