Messages and attachments in the app and backend

Attachments:
For the data-field in journal_im, we expanded it to include a JSON-object with the fields: message, attachment. A journal_im needs at least a message or an attachment to be sent to the backend. The attachment needs to be base64-encoded to be sent. It will be decoded and converted to a file in the backend. The file will be stored on the server with a unique id name in the media-folder in different sub-folders depending on the type of the file. An example could be an image sent in VICU-123: the file will be stored in media/VICU-123/unique_id123.jpg. In journal_im the data-field could look like this:

```
{
        message: “Hello”
        attachment: “/media/VICU-123/image/unique_id123.jpg”
}
```

The attachment is the file-url of the file. This will be sent back to the user to provide the attachment. At the moment an image could either be a jpeg/jpg or png. For an audio it could be wav or mp4.

Read and seen:
journal_im_readseen is a new table we added to the journal database. It has a many to one relationship to journal_im and is optitional to use. Entries to this table are not made when an entry is made to journal_im, rather they are made on the fly when users of the app see/read messages in chats. If the user is missing an entry in the readseen table for an IM it means that it is not read or seen by the user. When the user sees or reads a message a table row is appended, subsequent calls will simply update the existing row. A message is considered seen when the user sees the chat header on the chat overview, and a message is considered read when the user enters the chat with the unread message. The reason for the many to one relationship is that multiple users can receive the same message.