**Messenger fragment**

Creating a one application that provides all the services the patient can need has been a main focus since the beginning. And communication between the doctor and the patient is one these services both of them needs.

Under the communication section in the navigation drawer a messenger tab exists the purpose is to create a simple chat between the doctor and his patients to send notes, advices and instructions to his patients, also the patients can send questions or replays to the doctor. a UML class diagram is shown in figure (x).

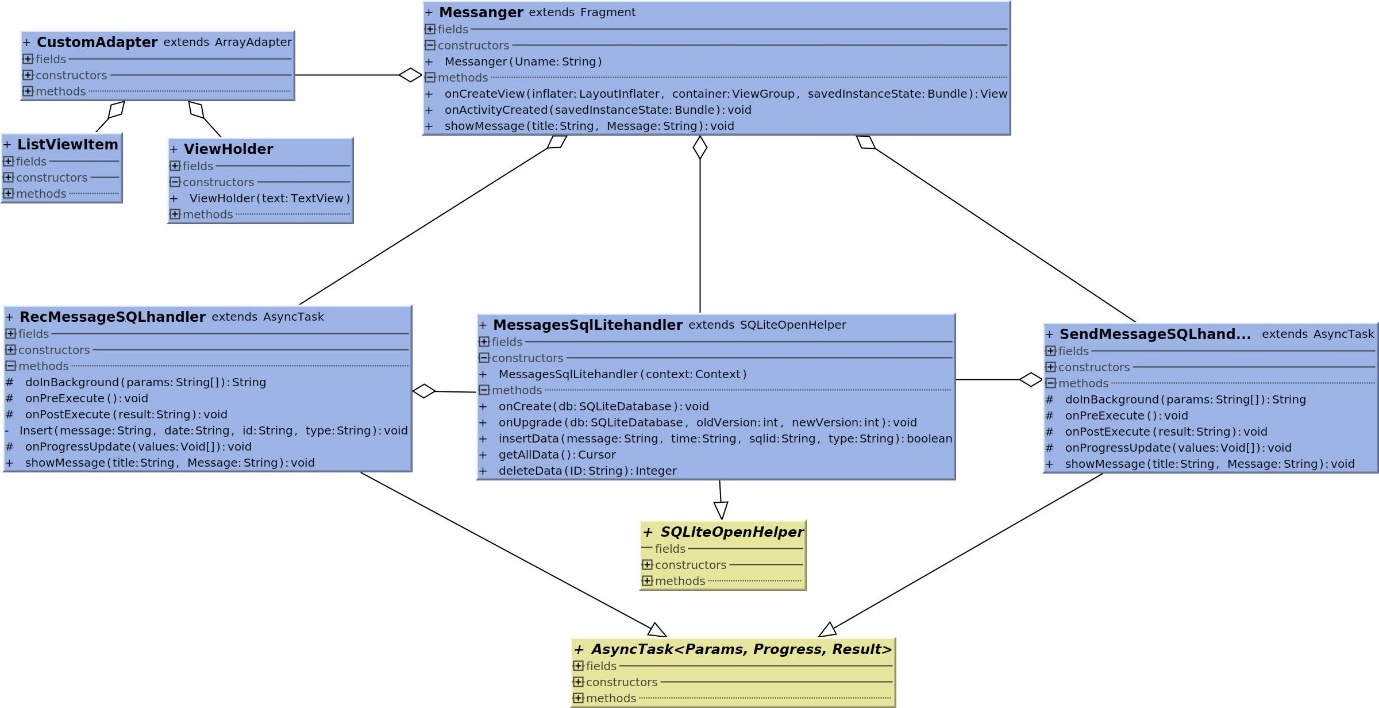


Figure (x): Class diagram for messenger.

The diagram may seem complicated but actually its very simple at the left side of Messenger in the figure the three classes CustomAdapter, ListViewItem and ViewHolder are responsible for displaying the messages in a simple way so the user can easily figure out the sent messages from the received ones based on the background colour, the text colour and the text gravity. ViewHolder loads as messages as the screen can hold only not all the existing messages and loads more when the user scrolls up or down to make the layout smoother and loads only the data can be displayed and fit in the screen.

Each message data contains the message its self and an overhead, the overhead contains the data and time of the message and the message type whether the message is sent or received. The CustomAdapter reads the data find out its type and choose the correct figure x+1 shows the way messages are displayed based on their type.

Figure x+1 to be added

There are two types of networks operation that requires connecting to the webserver the first one is when the user compose a message and send it and the class handling this operation is SendMessageSQLhandler which sends the message and the unique user id so it can be received correctly at the database the class also receive confirmation that the message was received and no issues occurred. The second class is RecMessageSQLhandler which runs in the background to find if the doctor sent any messages to the patient if so receive them, store them and display them. Both the classes mentioned above extends AsyncTask class to create a thread that allow these operations to run smoothly in the background without putting the hole application on hold.

To avoid retrieving all the messages from the webserver each time the fragment is used a local SQLite database keeps the latest messages on the devices with a unique id that was received in the response that SendMessageSQLhandler and RecMessageSQLhandler got from the server. These ids are used to find if there are new messages or not if there is receive them if not there is no need to receive the old one as they already exist. MessagesSQLiteHandler class takes care of all that, the messages Table is shown in Table y

Table y: Messages\_Table

|  |
| --- |
| Messages\_Table |

|  |  |
| --- | --- |
| MESSAGE | TEXT |
| TIME | TEXT |
| TYPE | TEXT |
| SQLID | INTEGER |

MessagesSQLiteHandler extends SQLiteOpenHelper and the functions in it create the table also insert, update, delete or retrieve the data from the table.

Messenger fragment connects all these classes and organize the operations mentioned above it also handles the click listeners and the touch or scroll events. Its also responsible for inflating the layout that shows all the messages, the message composing are and the send button. Figure x+1 shows the layout of the fragment



Figure x+1 (to be replaced)

On the website the doctor can view the messages sent to him by all his patients more on that can be found in section (chat on the website section number).

To summarize the functions of this fragment a simple diagram is shown in figure x+3.

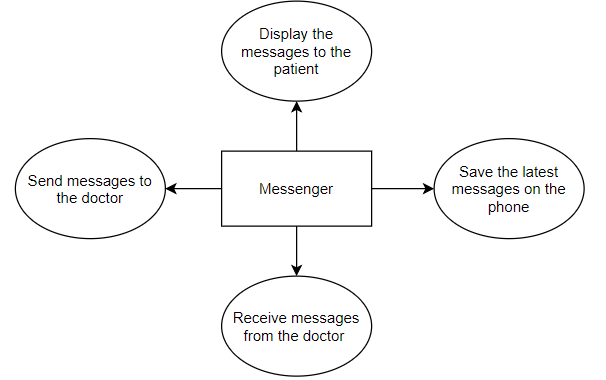


Figure x+3