Developer manual

Before everything…

You should have some notion of what IRC is.
Then, clone the repository from Github using the command

```
$ git clone git@github.com:ircSEX/ircSEX-Messenger
```

Dependencies

- JDK 7
- Android SDK
- Android support library v4
- sshj - SSHv2 library for Java

Dependencies for unit tests:

- Groovy 1.8.6
- Spock-core 0.6-groovy-1.8

The project uses the Apache Maven[1] build automation tool. The dependencies will be resolved automatically when using it.

Build

The pom.xml in ircSEX-Messenger/ircSEXMessenger defines the project object model for Maven. In this directory, Maven tasks can be executed for the project.

To build the project, run the Maven goal install with

```
$ mvn install
```

To only run the unit tests, run the Maven goal test with

```
$ mvn test
```

To deploy the application to a connected Android device, execute the task

```
$ mvn android:deploy
```

To uninstall the application from a connected Android device, execute the task

```
$ mvn android:undeploy
```

There are SNAPSHOT artifacts kept from every release. These are located in the ircSEX-Messenger/apk directory. To install an old SNAPSHOT, run

```
$ adb install <.apk name>
```

---

[1] http://maven.apache.org/

Release procedure

Refer to the Definition of done document and utilize the best practices defined in it when developing and releasing. Note that we use git-flow by Vincent Driessen.

Architecture

This serves as an introduction to the system and its architecture.

Packages

The project's top level packages and their responsibilites are:

- `irc` - IRC protocol implementation
- `model` - domain model for the client
- `ui` - contains activities, fragments, etc.
- `view` - contains extensions of Android view classes

Figure 1 shows the dependency structure in Stan.

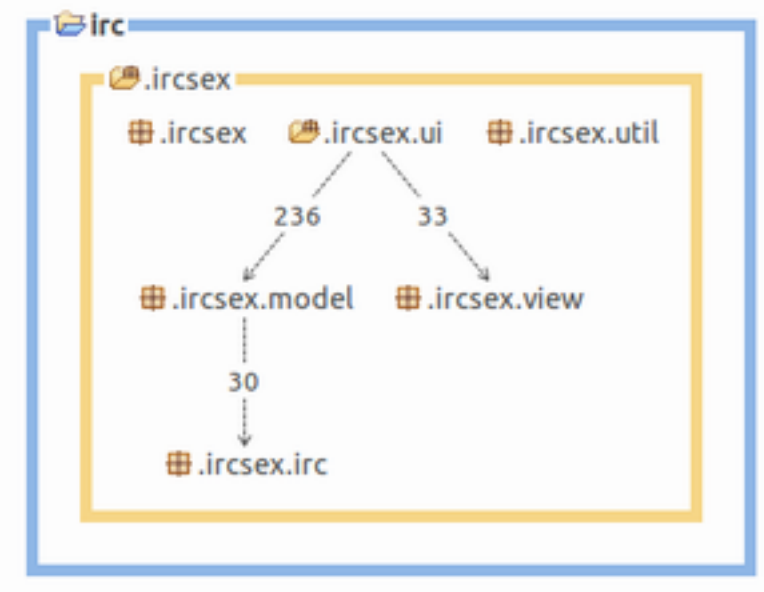

Figure 1. Package view in Stan.

The application entry point is ChannelActivity.

Database schema

The database is Android's default SQLite. It consists of three tables, which are shown in figure 2. Note that SQLite does not have a boolean data type, that is why the boolean values of useSSL and useSSH in t_servers are stored as integers as per the SQLite

best practices. If we ever migrate to a real relational database, it would be preferable to use a foreign key constraint in t_channels' server value on the t_servers' primary key.

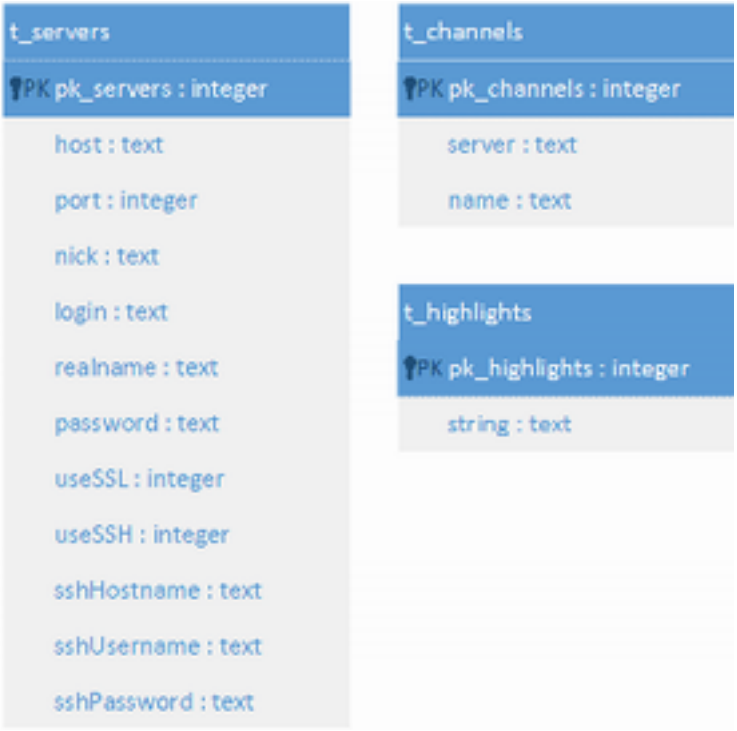

Figure 2. Database diagram.