# **<Signal-Team Incognito> Release Summary**

## **Team members**

|  |  |  |
| --- | --- | --- |
| Name and Student id | GitHub id | Number of story points that member was an **author** on. |
| Remus Mocanu | RemusCM | 40+8+1+40+20=109 |
| Daanish Rehman | daanish93 | 40+8+3+1+40+20=112 |
| Dennis Zhaolin Liu | dizidel | 40+8+1+40+20=112 |
| Ian Tablate | ian-tab | 40+8+1+40=89 |
| **Peter** Chen | petchan123 | 40+8+1+40=89 |
| Raymart De Guzman | tramyardg | 40+8+3+1+20+40=112 |

## **Mobile App summary (max one paragraph)**

***Signal*** is mobile a mobile application that allows users to message and make calls to others. This application allows users to send text, audio, video, pictures, and documents. The main benefit of using *Signal* is that is provides the user with an end-to-end encryption which makes places an emphasis on protecting the privacy of the user on like most other messaging platforms. End-to-end encryption protects the user not only from malicious hackers, but also from the private corporation that make a profit by logging user activity in order generate profit through advertisements. End-to-end encryption technology encrypts the message through the cloud and by decrypts the message only when it is received by the recipient making the messages private.

## **Velocity**

(make sure the iteration is clickable link to the milestone on github)

Only stories that have stakeholder signoff, demo steps, and tests are counted.

Please tag your released commit using “git tag release1”

Total: 7 stories, 132 points over 4 weeks

(If we only count the TA-signed stories, we only have 2 stories in the two sprints)

[Iteration 1](https://github.com/daanish93/SignalApp/milestone/1) (5 stories, 92 points)

[Iteration 2](https://github.com/daanish93/SignalApp/milestone/3) (2 stories, 40 points)

## **Plan up to next release**

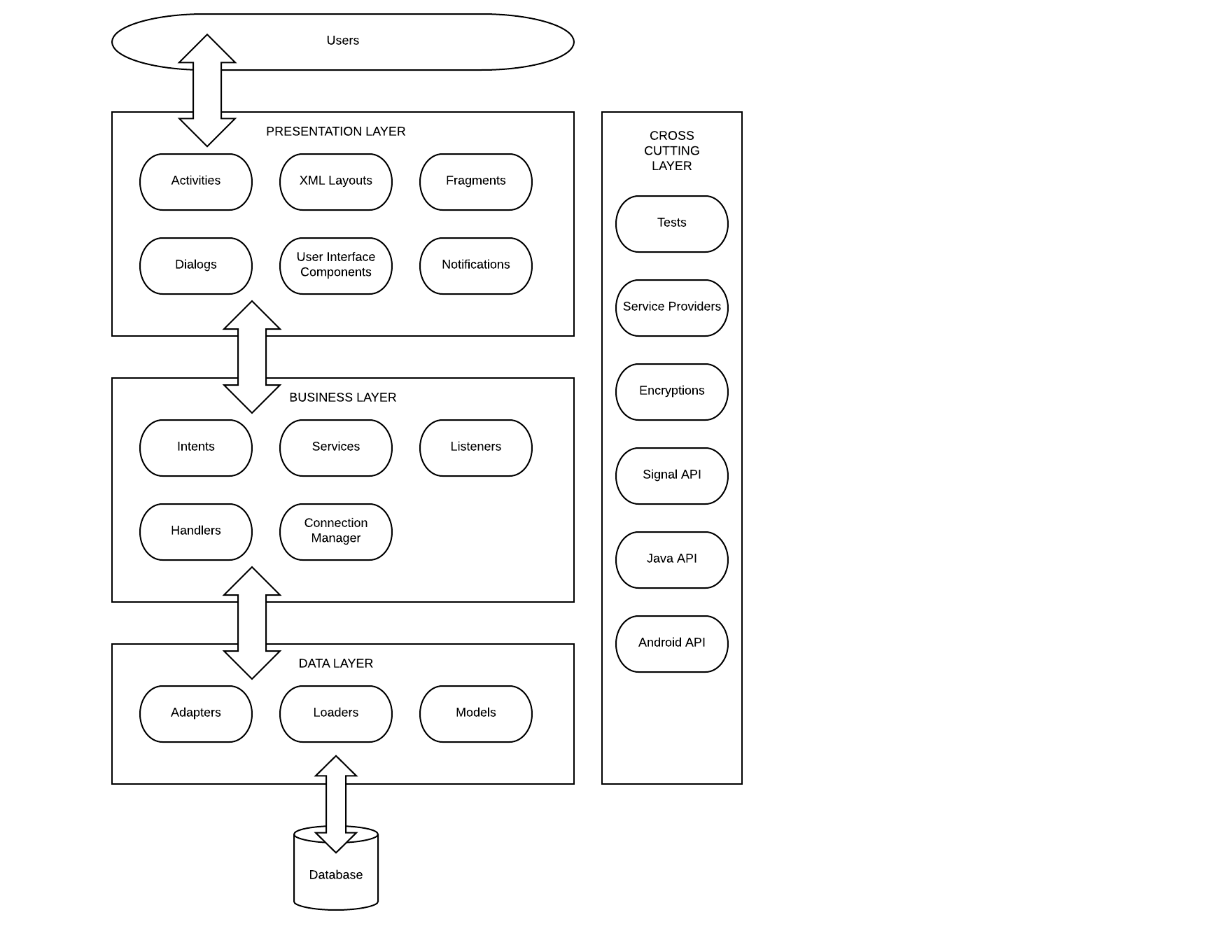
Example:

Total: 13 stories, 90 points, over 7 weeks

[Iteration](https://github.com/marouen-lamiri/Second-Soul/issues?utf8=%E2%9C%93&q=milestone%3A%22Sprint+4%22+) 3 (4 stories, 30 points)

[Iteration 4, Release 2](https://github.com/marouen-lamiri/Second-Soul/issues?utf8=%E2%9C%93&q=milestone%3A%22Sprint+6%22+) (4 stories, 27 points)

## **Overall Arch and Class diagram**



*Diagram created by Dennis*

The architectural style of this software follows that of a typical three-layered service application, it has a presentation layer, a business layer, and a data access layer.

1. The presentation layer contains the front-end components such as activities, fragments, dialogs, notifications, and other UI elements.
2. The business layer contains components that implement business logic, and services that execute the business logic.
3. The data access layer contains components that transform business layer objects into data access objects; the adapter enables queries to the database, the loader creates returning objects for the requested database classes, and the model specifies how the database is structured.

There is also a cross cutting layer which contains components that span across all of the 3 layers, these components affect the entire application.

## **Infrastructure**

For testing, we used the Mockito testing framework which allows for the mocking of objects, creating fakes, stub classes and dummy objects. For the purpose of the nickname functionality, it is used to mock objects for testing the nickname functionality.

<http://site.mockito.org/>

## **Code**

Key files: top **2** most important files that you wrote or changed (full path). We will also be randomly checking the code quality of files. Please let us know if there are parts of the system that are stubs or are a prototype so we grade these accordingly.

|  |  |
| --- | --- |
| File path with clickable GitHub link | Purpose (1 line description) |
| <https://github.com/daanish93/SignalApp/blob/master/src/org/thoughtcrime/securesms/NicknameChangeActivity.java> | The underlying logic behind the nickname functionality. |
| <https://github.com/daanish93/SignalApp/blob/master/res/layout/recipient_preference_activity.xml> | The xml file behind the nickname functionality. |

## **Testing and Continuous Integration**

Each story needs a test before it is complete. If some class/methods are missing unit tests, please describe why and how you are checking their quality. Please describe any unusually aspects of your testing approach.

List the **2** most important tests that you wrote or changed with links below.

|  |  |
| --- | --- |
| Test File path with clickable GitHub link | What is it testing (1 line description) |
| <https://github.com/daanish93/SignalApp/blob/master/test/androidTest/java/org/thoughtcrime/securesms/ChangeThemeTest.java> | This is testing the addition of the new pink theme. |
| <https://github.com/daanish93/SignalApp/blob/master/test/unitTest/java/org/thoughtcrime/securesms/database/NicknameTest.java> | This is testing the adding a nickname functionality. |

Describe your continuous integration environment. Include a link to your CI.

We are using Travis CI (Continuous Integration) to test our system. Travis CI features are consisting of automatic builds. This implies that every time a new change is pushed to a branch, a build is executed to check errors in the code. These errors are of course fixed before they are merged to the project main branch. With automated builds by Travis CI, we can focus more on coding and create useful features for clients.

<https://travis-ci.com/daanish93/SignalApp/builds>