<Signal-Team Incognito> Release 2 Summary

Team members

Name and Student id	GitHub id	Number of story points that member was an **author** on.
Peter Chen*	petchan123	13 + 13 + 40 = 66
Daanish Rehman	daanish93	13 + 13 + 40 = 66
Dennis Zhaolin Liu	dizidel	13 + 13 + 40 = 66
Ian Tablate	ian-tab	13 + 13 + 40 = 66
Remus Mocanu	RemusCM	13 + 13 + 40 = 66
Raymart De Guzman	tramyardg	13 + 13 + 40 + 10 = 76

Velocity

We completed 4 stories over 2 weeks which add to 76 story points.

The **velocity** is 38 story-points / week.

Overall Arch and Class diagram

There were no changes to the architecture on the application. In order to implement feature we added classes on top of the existing architecture. For example, to name a few classes, for the Clear Conversation feature we added ClearConversationActivity.java and for the Moderator Feature we added RecipientPrivilege.java into the root source folder src\org\thoughtcrime\securesms

Plan up to next release

To break down the current release, there are 4 completed stories: 3 stories in Sprint 4 and 1 story in Sprint 3.

This following is a link to the Sprint 3 and Sprint 4.
Sprint 3, Release 2 (13 points)
Sprint 4, Release 2 (63 points)
Release 2 sum up to 76 total story points.

The plan for the upcoming release will consist of the following features:

- voice recognition
- message lock in a chat
- message scheduler
- video compression

Changes to the aforementioned features might be done throughout the project. The distribution of these features in different sprints will be decided later on.

Infrastructure

The following APIs were used

1. TextDrawable from https://github.com/amulyakhare/TextDrawable
2. Shortcut Manager from the Android API https://developer.android.com/reference/android/content/pm/ShortcutManager.html
3. In addition, we added a column in the group database's group table in order to implement the group moderator feature..

Name Conventions

The naming convention is that we used CamelCase for the name of the classes were the first "C" in camel is lowercase and the second "C" in Case is uppercase.

For example, in the code the classes we added

- ClearConversationActivity
- RecipientPrivilege

When we are coding we are using 2 spaces instead on a tab and variable name is written in camelCase were the first "c" in camel is lowercase and the second "C" in Case is uppercase.

For example, in the code some of the variables we added

- currentUserPhoneNumber in RecipientPrivilege.java
- groupId in RecipientPrivilege.java

For constants we used all caps with underscore to separate them (eq: STUDENT_ID)

For example, in the code some of the constants we added

- EDIT_GROUP(64) in PermissionType.java
- CLEAR_GROUP_CONVERSATION(32) in PermissionType.java

Code

Key files: top **5** most important files (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)
GroupDatabase.java (commented methods are ours)	Updates the group assigning the creator as the moderator.

PermissionDatabase.java	Assign privileges to members of a group.
RecipientPrivilege.java	The class check if the recipient in a group has permissions to edit group and clear conversation.
ClearConversationActivity.java	It contains the functionality that clears the solo and group conversation history.
ShortcutCreator.java	It creates the shortcut for the most recent conversations.

Testing

Each story needs at least a tests 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 **3** most important **unit test**s with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
GroupDatabaseTest.java	It is testing if a given recipient is a moderator by checking the group database's groups table..
MessageTest.java	This file is testing the clear chat feature.
PermissionDatabaseTest.java	This file is testing if the privileges are being stored when a moderator is being created.

List the **3** most important **UI end to end (espresso) tests** with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
EditGroupConversationTest.java	UI Test for editing group and adding member.
ClearGroupConversationTest.java	UI Test for clearing messages in group conversations.
ClearSoloConversationTest.java	UI Test for clearing messages of solo conversations.

SHORT Story summaries

The following is the order of the stories by risk and priority.

Points: 40, Priority: 5, Risk: High
Story #8: Group Chat Moderator - Edit Group Restriction.
Milestone: Sprint 4 (Release 2)
Feature label: Moderator Edit and Add Feature

This story involves allowing moderator (users with elevated privileges) to be able to edit the group name or add new members to an existing group. The work involved in implementing this feature includes: creating a mockup, creating a field in the database to store the moderator's number, adding a method to restrict non moderator members, and applying the restriction in the existing layout.

Points: 13, Priority: 4, Risk: Medium
Story #9: Group Chat Moderator - Delete Group Message Restriction
Milestone: Sprint 4 (Release 2)
Feature label: Moderator Clear Chat
This story involves allowing moderator (users with elevated privileges) to be able to clear the chat. The work involved in implementing this feature includes: adding a button to the group option settings, assigning clear chat permission to the moderator, and implementing the functionality to delete message history in a group conversation.

Points: 13, Priority 4, Risk: Medium
Story #7: Clearing the chat
Milestone: Sprint 3 (Release 2)
Feature label: Clear Chat

This story involves allowing a user to be able to clear the chat. The work involved in implementing this feature includes: adding a clear chat button to the user option settings and implementing the functionality to delete message history in a one-to-one conversation.

Points: 10, Priority: 3, Risk: Low
Story #10: Create Conversation Shortcut
Milestone: Sprint (Release 2)
Feature label: Shortcut Feature
This story involves allowing user to be able to create shortcut for their most recent chats. The work required to make this feature happen includes: creating a mockup, adding an external library for creating an avatar based on a recipient's initial, fetching the most recent conversation from the database, and using the data to create shortcuts with the help of Android's App Shortcut API and Intents.

CI stages and notification

We integrated the unit test with Travis CI.

One example of notification with error is that we created a documents called report_build_error.xlsx which contains information on the error when the integration was configured.

In terms of the build stages, the Travis' build is separated into two stages, namely the compilation stage and the unit testing stage. The former uses the gradle assembleDebug script, and the latter uses the gradle test script.