

Bounty Program Agreement

Project Name

Chat v2 - Bounty

Version: 2

Team

Franklin Marcano, Rodrigo Guzmán, Richard Lozada, José Cardozo, José D. Vilchez, Eleazar Oroño, Arnaldo Santamaría

Abstract

This version extends some of the functionalities that were developed on Chat version 1. Allowing:

- Management of user's profiles
- Management of communities to establish connections among users
- Show when a remote contact is writing on the chat
- Show when a remote contact is online
- Show when was the last connection of a remote contact
- Clean the messages of a chat
- Delete all chats
- Delete an specific chat
- Copy the text of a chat bubble.

Scope

1. Management of user's profiles

Screens developments and logic structures to allow the creation and actualization of local profiles that will be used to make a connection with more profiles and as a way to establish a chat with third party contacts

Fermat Components

- Chat Profile Subapp
- Chat Profile Plugin

Size: 30%

2. Management of communities to establish connections among users

Screens developement and logic structures to allow list the profiles with which a connection is established, send a request to connect with another contact, list the community profiles, list the connections request notifications as well as block a profile and remove such connection

Fermat Components

- Chat Community Subapp
- Chat Community Plugin

Size: 35%

3. Show when a remote contact is writing on the chat

Screens developement and logic structures to show when a remote contact is writing on the chat as a way to improve the interaction.

Fermat Components

- Chat Subapp
- Chat Middleware Plugin
- Chat Network Service

Size: 5%

4. Show when a remote contact is online

Correspondent screens modifications and necessary logic structures to show when the remote contact is online

Fermat Components

- Chat Subapp
- Chat Middleware Plugin
- Chat Network Service

Size: 5%

5. Show when was the last connection of a remote contact

Screen developments and logic structures to show when was the last time a contact was online

Fermat Components

- Chat Subapp
- Chat Actor Connections
- Chat Network Service

Size: 5%

6. Clean the messages of a chat

Modifications to the appropriate screens and logical structures necessary to remove messages from a specific chat without removing the conversation. This option is available both in a popup menu when you click and let sustained in a particular chat list to be cleaned as in the menu on the toolbar messages.

Fermat Components

- Chat SubApp
- Chat Middleware Plugin

Size: 5%

7. Delete all chats

Screens modifications and logic structures to delete all created chats. This option will be displayed on the tool menu bar in the chat list

Fermat Components

- Chat Subapp
- Chat Middleware Plugin

Size: 5%

8. Delete an specific Chat

Screens modifications and logic structures to delete a selected chat. This option will be displayed on the popup menu that appears when the user taps and hold on a determined chat from the list that the user wants to delete.

Fermat Components

- Chat Subapp
- Chat SubApp Module
- Chat Middleware Plugin

Size: 5%

9. Copy the text of a chat bubble

Screen modification and logic structures to copy the text inside of a bubble from a chat.

Fermat Components

- Chat Subapp

Size: 5%

Note: This Scope does not include:

- Management of notifications by making a sound when a message is received
- Broadcasting list creation of messages to the added contacts on CHT
- Groups and members creation

- Sending group messages
- Contact Blocking
- Send a chat by email
- Messages Pagination
- Search in chats, contacts and messages.
- Chat payment management
- Transaction process management to be used from others platforms
- Emoticons management

Evaluation

The acceptance criteria will be the follow:

1. Management of user's profiles

- Create a chat profile
- Edit chat profile information
- Publish and remove a chat profile from Fermat's network

2. Management of communities to establish connections among users

- List connected profiles
- Select a profile for a connection request
- List connections
- Show which profiles are being notified
- Disconnect a profile

3. Show when a remote contact is writing on the chat

- Show label "typing..." when a remote contact is writing
- The app must send through the Network Service the information when a contact is writing on the chat with the remote contact involved



4. Show when a remote contact is online

- Show "online" label just under the contact name when the remote is online
- The app must send through the Network Service the information when the remote contact is online



5. Show when was the last connection of a remote contact

- Show label under the name of the remote contact that contains the last time the contact was online

- Save last connection of the remote contact on Actor Connections
- The app must send through the Network Service the last time a contact was online



6. Clean the messages of a chat

- Delete all messages of an specific chat from the chat list and from the messages window of the same chat

7. Delete all chats

- Delete all chats from the chat list

8. Delete an specific chat

- Delete an specific chat from chat list and associated messages

9. Copy the text of a chat bubble

- Copy the content of a bubble inside of a chat

TERMS AND CONDITIONS:

1. The team agrees that the implementation project has two stages: functionality and beta testing.
2. The team understands and accepts that the functionality will be considered done when all the features described in the scope of this agreement are completed and tested in an alpha stage.
3. The team understand and accepts that the implementation of functionality must follow the Fermat's technical guidelines.
4. Component architecture and workflows are created in the Analytics System and the Interfaces in the API library of the platform involved.
5. The team understands and accepts that implementation will be evaluated by the @bounty-program-team and will include GUI/UX design checking, functionality test and review of Fermat's technical guidelines compliance.
6. The team understands and accepts that there is only one free review for functionality and one for Fermat's technical guidelines compliance. The following reviews will cost the team 25% of the related bounty each in case the first one wasn't approved.
7. The team agrees to complete the implementation on the following conditions:

Implementation due date: All the features will be finished before **15/05/2016**. If on this date the development team does not deliver the agreed functionality, it will lose the implementation bounty.

Implementation collateral deposit: The team agrees to deposit the amount of **\$1.750** paid in tokens in favor of the @bounty-program-team, as a collateral to be lost if this part project is not approved before the due date.

Implementation margin: No more penalties are applied **7** calendar days after implementation due date.

Implementation penalty: **5%** of the implementation bounty for each calendar day that elapses after the implementation margin without formal acceptance from the @bounty-program-team. This penalty will be paid by the development team from its savings to the @bounty-program-team. If savings are not enough it will be deducted from their cash salaries.

Implementation bounty: The functionality will be **70% of the total bounty**. This bounty will be awarded to the development team when the @bounty-program-team considers that the functionality delivered is done.

8. The team understands and accepts that beta testing will be conducted by the @beta-testing-team.
9. The team understands and accepts that criteria to pass beta testing are:
 - A. No bug issues on beta testing due date,
 - B. Or no bug issues in a period of three (3) consecutive calendar days before the due date,
 - C. Or no answer from @beta-testing-team about solved bugs for (6) consecutive days before the due date.
10. The team agrees to complete the beta testing on the following conditions:

Beta testing due date: Beta testing will be passed before **30/05/2016**. If on this date the development team does not pass the beta testing, it will lose the beta testing bounty, which will be automatically awarded to the @beta-testing-team.

Beta testing collateral deposit: The team agrees to deposit the amount of **0** tokens in favor of the @bounty-program-team, as a collateral to be lost if this part of the project is not approved before the due date.

Beta testing margin: No more penalties are applied **5** calendar days after the beta testing due date.

Beta testing penalty: 5% of the beta testing bounty for each calendar day that elapses after the due date without formal passing through beta testing. This penalty will be paid by the development team from the implementation bounty previously awarded or its savings to the beta testing team. If savings are not enough it will be deducted from their cash salaries.

Beta testing bounty: The beta testing bounty will be a fixed **30% of the total bounty**. It could be awarded to the development team if it passes the beta testing on time or by @beta-testing-team if they fails. It implies that development team will not get this bounty unless it succeeds in the beta testing process.

TOTAL BOUNTY:

The total amount of the bounty in Fermat tokens for this project is **\$ 10.000 (\$7000 for implementation + \$3.000 for beta testing)**.

DISTRIBUTION OF BOUNTY BY CONTRIBUTOR

Franklin Marcano	15%
Rodrigo Guzmán	15%
Richard Lozada	15%
José Cardozo	15%
José D. Vilchez	15%
Eleazar Oroño	8%
Arnaldo Santamaría	15%
Total amount deposit: \$1.750	

SUMMARY

Implementation due date	15/05/2016
Implementation collateral	\$1.750,00
Implementation margin (days)	7,00
Implementation penalty (%/day)	5
Functionality review (attempts-25%)	1
Technical review (attempts-25%)	1
Implementation bounty (\$)	\$7.000,00
Beta testing due date	30/05/2016
Beta testing collateral (\$)	0
Beta testing margin (days)	5
Beta testing penalty (%/day)	5
Beta testing bounty (\$)	\$3.000,00
Total bounty (\$)	\$10.000,00