BOUNTY PROGRAM AGREEMENT

PROJECT NAME
Loss Protected Wallet

VERSION
1

TEAM
Natalia Cortez, Joaquin Carrasquero, Gian Barboza, Dayana Matos

ABSTRACT
Its main goal is to protect the user from losing money when attempting to spend Bitcoins at a price lower than the price when that Bitcoin amount was first purchased (or entered the wallet). The new functionalities include:

- Storing the exchange rate when a BTC amount is received in the wallet
- Checking the exchange rate when the user attempts to spend an amount of BTC and freeze the funds if they are below the purchased price

When a user tries to spend BTC the application will evaluate if current BTC price is higher than the one that it was last saved when it entered the wallet, and It will allow to spend blocks starting first with the one that has a higher price then it will keep consuming the blocks that has a lower price.

A block value is determined by the quantity of BTC that got in the wallet at a moment. The amount of BTC that enters to the wallet in a transaction are taken as a block of value which they are going to start being used when payments are made with the wallet.

The Fermat book chapter related to this bounty can be found here:
https://github.com/Fermat-ORG/fermat-book/blob/master/book-chapter-12.asciidoc

Loss Protected Wallet will be part of CCP layer and it will reuse already made components like Network Services, transactional modules and contact modules.

It will have the same functionalities as Bitcoin Wallet but also it will be added a specific functionality:

- Available balance will be calculated based on value blocks which it can be spent by the user in that moment having in consideration bitcoin dollar price.

- It will show bitcoins in a wallet screen and how much does it have consumed from each chuck value.

- At the very moment of making a payment it won't be able to exceed available balance, which it was calculated based on current dollar price.

- It will be shown the bitcoins value based on current dollar price at the moment.

- From wallet settings it is going to be able to avoid bitcoins loss protection restriction through selecting a configuration option which if turned on, it wont let the user spend BTC and if it is not, It will be possible to spend them.

For this new wallet new components will be developed:

- Loss Protected Wallet App .
- Loss Protected Wallet Module.
- Basic Loss Protected Wallet .
- Transactional module of exchanging bitcoins between wallets, It registers outgoing Loss Protected Wallet bitcoins while they are credited in the other wallet or viceversa.

SubApp Community will be used for establishing relationships between Fermat Users, who are going to be able to be added as Contacts, SubApp Identity will be used too in order to create an identity associated to the wallet.

Current dollar price will be obtained from CER-Currency Exchange Provider component.

SCOPE

1. **Wallet Home**

   Develop main wallet screen, and logical needed structures that allow to display current real balance and available along with bitcoins current dollar price at the moment, a sum of how much dollars have been earned until that moment, and a linear graphic with the total bitcoins which it was spent and with what price was done.

   Fermat Components
   - Loss Protected Wallet App
   - Loss Protected Wallet Module
   - Basic Loss Protected Wallet

   Size: 15%

2. **Send Bitcoins to a wallet contact**

Develop of form and logical structures needed for sending bitcoins to contacts. This will be done by having in consideration current available balance for validate that the user can send BTC without losing money and, in case of available balance being lower, it will be validated if it is available a configuration option named “unprotect”, for allowing to send bitcoins but warning the user first about the money loss.

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Basic Loss Protected Wallet
  - Crypto Transaction Outgoing Intra User
  - Crypto Transmission Network Services

Size: 10%

3. **Receive Bitcoins from another user wallet**

Developing logical structures necessary to receive Bitcoins from other wallets owned by the user as Bitcoin Wallet (in this first implementation), aiming to protect the bitcoins that entered these wallets. Validations when sending bitcoins and warnings about the loss of money is taken into account.

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Basic Loss Protected Wallet
  - Crypto Transaction Transfer Intra Wallet

Size: 15%

5. **Transaction listing**

Development of screens and logical structures necessary to display bitcoin transactions sent and received, with dates and associated contact information .

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Basic Loss Protected Wallet

Size: 10%

6. **Send Payment Request to a wallet contact**

Development of the form and logical structures necessary to send a payment to a contact for a specific amount of bitcoins.

Fermat Components

- Loss Protected Wallet App

- Loss Protected Wallet Module
- Crypto Payment Request

Size: 5%

7. Payment request listing

Development of the appropriate screens and logical structures needed to display the list of payment requests received and sent as well as allowing to accept or deny a request from the received screen.

Fermat Components

- Loss Protected Wallet App
- Loss Protected Wallet Module
- Crypto Payment Request
- Network Services Crypto Payment Request
- Crypto Transaction Outgoing Intra User
- Crypto Transmission Network Services

Size: 5%

8. Add Contacts

Development of the appropriate screens and logical structures to add users to the wallet, users can be Extra Users (external that do not belong to the community of Fermat) or Fermat Users (identities within Fermat with whom a connection is made) .

Fermat Components

- Loss Protected Wallet App
- Loss Protected Wallet Module
- Wallet Contact Middleware
- Crypto Address Network Services

Size: 5%

9. Contacts List

Development of the appropriate screens and logical structures that allows to see a list of contacts added to the wallet, ordered alphabetically .

Fermat Components

- Loss Protected Wallet App
- Loss Protected Wallet Module
- Wallet Contact Middleware

Size: 5%

10. Contacto detail

Development of the appropriate screens and logical structures to see the contact details, name and crypto address. From this screen you can send bitcoins or a payment request.

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Wallet Contact Middleware

Size: 5%

10. Chunck Value Listing

Development of the appropriate screens and logical structures to see the list of value blocks representing each bitcoin entry into the wallet and its exchange rate, each record will have an indicator of whether that block can be spent or not at that time.

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Basic Loss Protected Wallet

Size: 10%

11. Chunk Value detail

Development of the appropriate screens and logical structures to see the list of consumption of a value block .The date of consumption is displayed and the exchange rate that was consumed, as well as the percentage that have consumed the block so far.

Fermat Components

- Loss Protected Wallet App
  - Loss Protected Wallet Module
  - Basic Loss Protected Wallet

Size: 15%

EVALUATION

It is considered accepted the Loss Protection Wallet app if it allows the user to:

- Receive and send bitcoins to and from your contacts
- Show the current dollar value of a bitcoin
- Prevent from sending bitcoins if the current dollar rate is lower than the price at which entered the bitcoins.
- Warn about losing money if the protection setting is disabled.
- Show how many dollars have been won or lost in transactions so far.

- Show Statistics about spending bitcoins by the user and how many are left.
- Allow add external contacts and Fermat contacts .
- Allow send and receive payment requirements.
- Verify and report the available balance usable for the moment using the dollar rate to determine how many value blocks the user can spend.

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 **06/05/2016.** 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 **3500** dollars 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:

   - No bug issues on beta testing due date,
   - Or no bug issues in a period of three (3) consecutive calendar days before the due date,
   - 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 **31/05/2016.** 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 **240** 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 bounty for each calendar day that elapses after the beta testing margin 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 + **$3000** for beta testing).

DISTRIBUTION OF BOUNTY BY CONTRIBUTOR

Natalia Cortez	30%
Joaquin Carrasquero	30%
Gian Barboza	30%
Dayana Matos	10%

Total amount deposit: **$ 3.740**

SUMMARY

Implementation due date	06/05/2016
Implementation collateral	$3.500,00
Implementation margin (days)	7,00
Implementation penalty (%/day)	5
Functionality review (attempts-25%)	1
Technical review (attempts-25%)	1
Implementation bounty ($)	$7.000,00
Beta testing due date	31/05/2016
Beta testing collateral ($)	240
Beta testing margin (days)	5
Bta testing penalty (%/day)	5
Beta testing bounty ($)	$3.000,00
Total bounty ($)	**$10.000,00**