|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| Capstone Project Document | |
| **Peer-to-Peer Lending System** | |
| **Group 10 - IS** | |
| **Group members** | Nguyễn Trung Dũng – SE62576 (Leader)  Huỳnh Văn Lộc – SE62291  Lê Ngô Minh – SE62705  Ngô Hoàng Đông – SE6 |
| **Supervisor** | Nguyễn Huy Hùng |
| **Ext. Supervisor** | N/A |
| **Project Code** | PPLS |

– **Ho Chi Minh City, 13 May, 2019** –

**Table of contents**

[A. Introduction 6](#_Toc15304527)

[1. Project Information 6](#_Toc15304528)

[2. Introduction 6](#_Toc15304529)

[3. Current Situation 6](#_Toc15304530)

[4. Problem Definition 6](#_Toc15304531)

[5. Proposed Solution 7](#_Toc15304532)

[**5.1 Feature functions** 7](#_Toc15304533)

[**5.2** **Advantages and disadvantages** 7](#_Toc15304534)

[6. Functional Requirements 7](#_Toc15304535)

[7. Role & Responsibility 8](#_Toc15304536)

[B. Software Project Management Plan 8](#_Toc15304537)

[1. Problem Definition 8](#_Toc15304538)

[2. Project Organization 11](#_Toc15304539)

[3. Project Management Plan 13](#_Toc15304540)

[4. Coding Convention 14](#_Toc15304541)

[C. Software Requirement Specification 15](#_Toc15304542)

[1. User Requirement Specification 15](#_Toc15304543)

[2. System Requirement Specification 15](#_Toc15304544)

**List of tables**

[Table 1 - Hardware Requirements for Server 10](#_Toc15303355)

[Table 2 - Hardware requirement for PC 11](#_Toc15303356)

[Table 3 - Software Requirement 11](#_Toc15303357)

[Table 4 - Roles and responsibilities 13](#_Toc15303358)

[Table 5 - Tools and techniques 13](#_Toc15303359)

[Table 6 - Deliverables 14](#_Toc15303360)

**List of Figures**

[Figure 1 - Scrum Framework 12](#_Toc15844731)

[Figure 2- Guest Overview Use Case 18](#_Toc15844732)

[Figure 3Figure 3 – Guest Login Use Case 18](#_Toc15844733)

# Introduction

## Project Information

* Project name:  **Peer-to-peer Lending System**
* Project Code: **PPLS**
* Product Type: **Website Application**
* Start Date: **May 13rd, 2019**
* End Date: **--**

## Introduction

These days, borrowing money is no longer strange. Borrowing money to invest in a project or to buy something like washing machine, phone, etc … that can be seen every day around our lives. However, borrowing money from banks still exists many cumbersome procedures that make borrowers encounter many difficulties in borrowing money. Moreover, whether transactions can be changed, or even deleted to protect an individual may cause someone to be guilty.

In this document, we introduce a solution to make loan transactions transparent, a web application called Peer-to-peer Lending System. This Peer-to-peer Lending System helps people borrow money directly by making an agreement with the lender, from which all transactions will be stored on the Blockchain and viewed by anyone in the system.

## Current Situation

As mentioned in the introduction, borrowing money causes many difficulties for borrowers because many procedures need to be authenticated by the authorities. This makes many people in urgent need to earn illegal money such as interest loans.

Also, trust between people and people is still deficient to be able to lend directly between two people. That's why creating trust among people is a necessity.

## Problem Definition

Below are the disadvantages of the current situation:

* Currently, if you want to borrow money from the bank, you need a lot of documents, as well as many complicated procedures. Not to mention whether you are approved for a loan after going through a customer appraisal step.
* The current interest rate is quite high. For unsecured loans VPBank from 15.91-21% / year, TPBank from 8.28 - 17% / year, ACB up to 22% / year. As for mortgage loans, about 5-12%/year.
* The amount can be borrowed from the bank is not too much. The maximum limit of banks falls between 300 million VND and 1 billion VND.

## Proposed Solution

We will build a website using blockchain to reduce the time of borrowing money from banks and increase the borrowing limit to higher than the current banks. Features will be implemented in the system:

### **5.1 Feature functions**

* Create requirements to borrow money from people who need to borrow money.
* Create and set the necessary milestones in borrowing as well as the most appropriate payments created by borrowers and lenders.
* Store transactions using Blockchain
* Use online payment gateways for transferring loans as well as paying such as Paypal, Ethereum, Momo, ….

### **5.2** **Advantages and disadvantages**

The advantages and disadvantages of the proposed solution:

* **Advantages**:
* Optimize time to do loan procedures.
* The loan limit is not dependent on banks so the limit may be much higher.
* **Disadvantages**:
* There are still risks when lending.

## Functional Requirements

Function requirements of the system are listed below:

**6.1 Admin:**

* Admin can activate, or deactivate the account.
* Amin can view list users.
* Admin can approvedocuments for increasing loan limit of user.
* Admin can reject document of user.

**6.2 Authorized User:**

* User can create a borrowing request.
* User can detect a borrowing request.
* User can view detail of a request.
* User can accept request after view detail.
* User can make a deal with the chosen request.
* User can view their all own transactions.
* User can view their all history requests.
* User can upload identity document for verifying to get loan limit

**6.3 Guest:**

* User can sign up a new account.
* User can log in into the system.
* User can view all transactions.

## Role & Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Huy Hùng | Project Manager | Supervisor | hungnh@fpt.edu.vn |
| 2 | Nguyễn Trung Dũng | Developer | Leader | dungntse62576@fpt.edu.vn |
| 3 | Huỳnh Văn Lộc | Developer | Member | lochvse62291@fpt.edu.vn |
| 4 | Lê Ngô Minh | Developer | Member | minhlnse62705@fpt.edu.vn |
| 5 | Hoàng Nhật Đông | Developer | Member | donghnse62357@fpt.edu.vn |

# Software Project Management Plan

## Problem Definition

* 1. **Name of this Capstone Project**
* Official name: Peer-to-Peer Lending System Based on Blockchain Technology
* Vietnamese name: Ứng Dụng Vay Tiền Giữa Các Cá Nhân Trên Nền Tảng BlockChain
* Abbreviation: PPLS
  1. **Problem Abstract**
* This project is a concern for individuals who have a surplus of money and want to invest in profits, but they need transparency in transactions. We call it Peer-to-peer Lending System (PPLS). PPLS will provide users with an interactive and easy-to-use interface for making transactions.
* Individuals use the website as a reputable trading channel because every recorded transaction is saved to the Blockchain. And Blockchain is a network where all transparency is exposed, so people can only view and cannot edit any transaction after it has been done.
* The Peer-to-peer Lending System can ensure transparency in all transactions between users, thereby reducing concerns about fraud in transactions.
* Finally, PPLS will perform the transactional data control between the database and Blockchain, thereby creating more credibility for customers that the transaction history has not been changed.
  1. **Project Overview**
     1. **Current Situation**

Below are the problems encountered in this project:

• Limit time and human resource: Team has only 4 member and time for all project is about 13 weeks for writing a document, implementing the products and testing.

• Framework study: team members have a problem when applying the play framework into project. The team needs an amount of time to get familiar new techniques.

• New technique: Some team members are new to the techniques used in the project. The team needs an amount of time to get familiar with those techniques.

• Lack of UI (user interface), UX (user experience) design skill: Our team members all study IS major, and no one has studied UI, UX design. Therefore, that some UI may misunderstand or hard to use with normal users.

* + 1. **The Proposed System**

Building a web application that allows individuals lending money to others. Proof of leading will be stored in a blockchain-based system to secure lending transaction. Admin can manage user and accept users though their identity information. Individuals can create and process transaction anytime. Besides, they can process to payback money in full amount or breaking the payment in different amount.

Task will be assigned vertically to team members, so that if one member quits, the team will not lack of resources.

* + 1. **Boundaries of the System**

The system can:

* Allow Admin to manage user accounts .
* Allow Admin to accept User’s document for increasing limit loans.
* Allow Individuals to conduct borrowing request.
* Allow Individuals to view and accept by other individuals.
* Allow Individuals payback in full amount; or breaking the payment in different amounts.

The system cannot:

* Current system just support Paypal for payment.
  + 1. **Future Plans**
* The system is currently being developed on the web platform:
  + We will develop our system on Android and iOS platform
* The current system uses some user information to identify the user:
  + We will add more other user identification information to reduce risks when users join the system and increase user loan limits.
* The current system has not been able to evaluate highly reputable users
  + We will design the user review feature after each transaction. From there, users may consider lending large amounts of money.
    1. **Development environment**
       1. **Hardware requirements**

**For server**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirement** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operation System** | Ubuntu 18.04 | Ubuntu 18.04 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer memory** | 4GB RAM | 12 GB RAM or more |
| **Storage space** | 5GB | 15GB or more |

Table 1 - Hardware Requirements for Server

**For PC**

|  |  |  |
| --- | --- | --- |
| **PC** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window 7 | Window 10. |
| **Computer Processor** | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| **Computer Memory** | 2GB RAM | 4GB RAM or more |
| **Web Browser** | Firefox (v52 or higher), Chromes (v42 or higher) | Chrome latest stable version |

Table 2 - Hardware requirement for PC

* + - 1. **Software requirement**

|  |  |  |
| --- | --- | --- |
| **Software** | **Name / Version** | **Description** |
| **Operation System** | Windows Server 2014 | Operating system and platform for development |
| **Environment** | Java EE 8.0, Node v10, npm v6 | Specification for developing web application |
| **Modeling tool** | StarUML, Software Ideas Modeler | Used to design diagram |
| **IDE** | Intellij IDEA 2018.1.5, Visual Studio Code 1.35.1, A5M2 2.12.1 (SQL) | Programming tools |
| **DBMS** | MYSQL 8.0 | Used to create & manage the database for system |
| **Source control** | Git on IDE (Github) | Used for source control |
| **Web browser** | Chrome 69 or above | Testing browser |

Table 3 - Software Requirement

## Project Organization

* 1. **Software Process Model**



Figure 1 - Scrum Framework

* 1. **Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Huy Hùng | Product Owner | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Nguyễn Trung Dũng | Scrum Master | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing * Arrange Meeting * Risk Management |
| **3** | Huỳnh Văn Lộc | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create a test plan * Coding * Testing |
| **4** | Lê Ngô Minh | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create a test plan * Coding * Testing |

Table 4 - Roles and responsibilities

* 1. **Tools and Techniques**

|  |  |
| --- | --- |
| Tool/Technique | Name |
| Front-end | HTML, CSS, JavaScript, jQuery, React Semantic UI |
| Back-end | JavaEE, SpringBoot framework, JPA, Hibernate, Python, Flask framework |
| IDE | NetBeans 8.2, IntelliJ IDEA 17.2 |
| DBMS | MySQL |
| Source Control | Github |
| Modeling tool | StarUML, Software Idea Modelers |

Table 5 - Tools and techniques

## Project Management Plan

* + 1. **Product Backlog**

Product Backlog could be found here.

* + 1. **Sprint Backlog**

Sprint Backlog can be found [here](https://docs.google.com/spreadsheets/d/1DvHD5Bwmk1IL5XokeVP7r67h5yaPtwHpret4qHMtcmw/edit?usp=sharing)

* + 1. **Deliverables**

|  |  |  |
| --- | --- | --- |
| **No** | **Deliverable** | **Note** |
| 1 | Introduction, Entity Relationship Diagram, Use Case Overview, Mock UI | Sprint 1 |
| 2 | Study Spring Boot Framework, React JS, BigchainDB, Design User Interface for Web Application | Sprint 2 |
| … |  |  |
|  |  |  |

Table 6 - Deliverables

* + 1. **All Meeting Minutes**

All sprint meeting minutes could be found here.

## Coding Convention

**Summary**:

* **Naming Convention**:
* Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.
* Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.
* **Indentation**:
* One declaration per line is recommended since it encourages commenting.
* In absolutely no case should variables and functions be declared on the same line.
* Do not put different types on the same line.
* **Declarations Convention:**
  + One declaration per line is recommended since it encourages commenting.
  + Using Java Code Convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# Software Requirement Specification

## User Requirement Specification

* 1. **Guest Requirement**

*Guest is a person who doesn’t have access to the system. Guest only try some demo features. To use all functions, guest must login. These are some functions guest can use:*

* Login.
* View History Transactions.
  1. **Authorized User Requirement**

*Authorized User is an authorized user who has permission to login into the system. With authorized user role, user can use below functions:*

* Create borrowing requests.
* View list borrowing requests.
* View list pending requests.
* View list dealing requests.
* View list history requests.
* View list trading requests.
* View list own requests.
* View detail of a request.
* Accept request after view detail.
* Cancel a request after view list dealing requests.
* Make a deal with the chosen request.
* Pay for the deal borrowed.
  1. **Admin Requirement**

*Admin is an authorized user has permission to login into the system under Admin role. Admin can use below management functions:*

* Active/Deactive User account.
* View all pending documents.
* Accept user documents.
* Reject user documents.

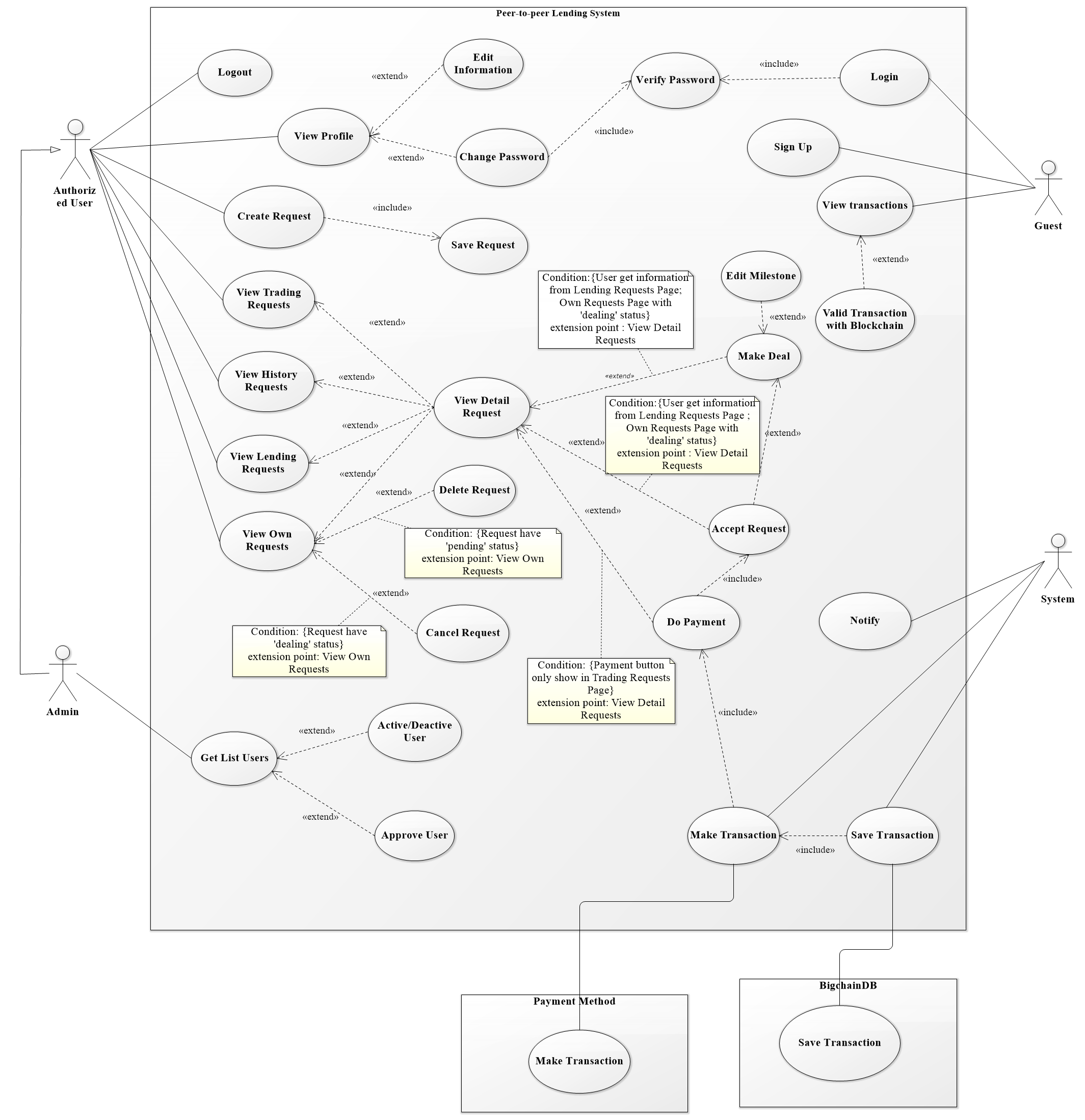
## System Requirement Specification

* 1. **External Interface Requirement**
     1. **User Interface**
* GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterate process includes design, sketching, prototyping, user assessment.
* The User Interface uses language is English.
  + 1. **Hardware Interface**
* Desktop PC.
  + 1. **Software Interface**
* Web application: work with browsers Firefox (v52 or higher), Chrome (v28 or higher), Internet Explorer (v10 or above) or with any web browser that supports HTML5 & CSS3.
  + 1. **Communication Protocol**

Use HTTP protocol 1.1 for communication between:

Web application and web server

* 1. **System Overview Use Case**



* 1. **List of use case**
     1. **<Guest> Overview Use Case**

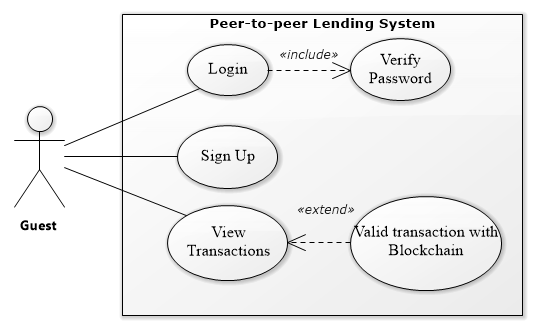
****

Figure 2- Guest Overview Use Case

* + - 1. **<Guest> Login**

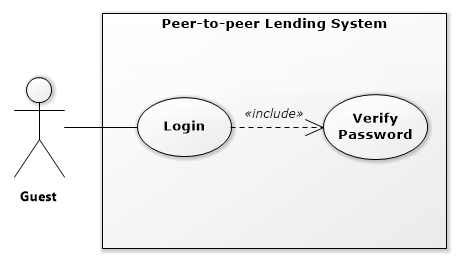
****

Figure 3 – Guest Login Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_GU\_01** | | | |
| **Use Case No.** | PPLS\_GU\_01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | DungNT | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guest to log in the system.   **Goal:**   * Guest can log in the system with a specific role.   **Triggers:**   * Guest sends login command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guest is exactly authorized with a specific role in the system. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to the login page. | The system requires identity information from Guest:  -Username: text input.  - Password: text input. | | 2 | Guest inputs information. |  | | 3 | Guest sends a command to log in to the system. | Guest logins system with their specific role. [Alternative 1] [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Guest enter wrong identity information. | The system shows error message “username or password is incorrect. Please try again.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  |  |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted before being sent to the server. * After login to the system, the guest will be redirected to view request page (user can see the other request). | | | |

* + - 1. **<Guest> View Transaction**

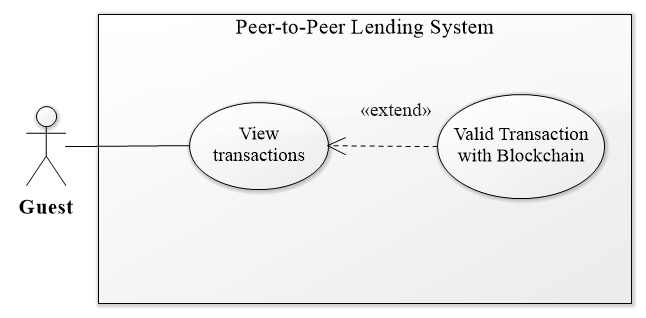
****

Figure 4 – Guest View Transaction Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_GU\_02** | | | |
| **Use Case No.** | PPLS\_GU\_02 | **Use Case Version** | 1.0 |
| **Use Case Name** | View transactions | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guests to view some transactions that completed.   **Goal:**   * Guest can see that there are successful transactions.   **Triggers:**   * Guest sends getting transactions command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guest can view transactions. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to the homepage. | Getting all present transactions with the following information:  - Id  - sender  - receiver  - amount  - createDate  - Status |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:** PPLS\_GU\_03  **Business Rules:**   * Some completed transactions must be listed. * Transactions must be ordered descending by time | | | |

* + - 1. **<Guest> Valid Transaction with blockchain**

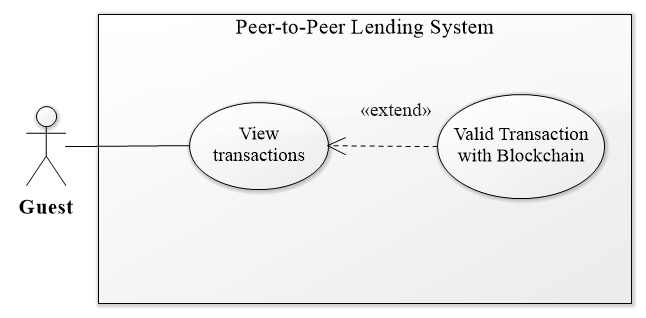
****

Figure 5 Guest Valid Transaction with Blockchain Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_GU\_03** | | | |
| **Use Case No.** | PPLS\_GU\_03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Valid transaction with blockchain | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guests to compare transaction between homepage and blockchain.   **Goal:**   * Guest can check that transaction in the homepage is real transaction that is saved in the blockchain.   **Triggers:**   * Guest sends a valid transaction with blockchain command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guest can check that transaction is valid or not. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest click on validate button | System show valid transaction dialog  with the status of the transaction |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest click on validate button | System show valid transaction dialog  with the status of the transaction |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:** PPLS\_GU\_02  **Business Rules:**   * Some completed transactions must be listed. * Transactions must be ordered descending by time | | | |

* + - 1. **<Guest> Sign Up**

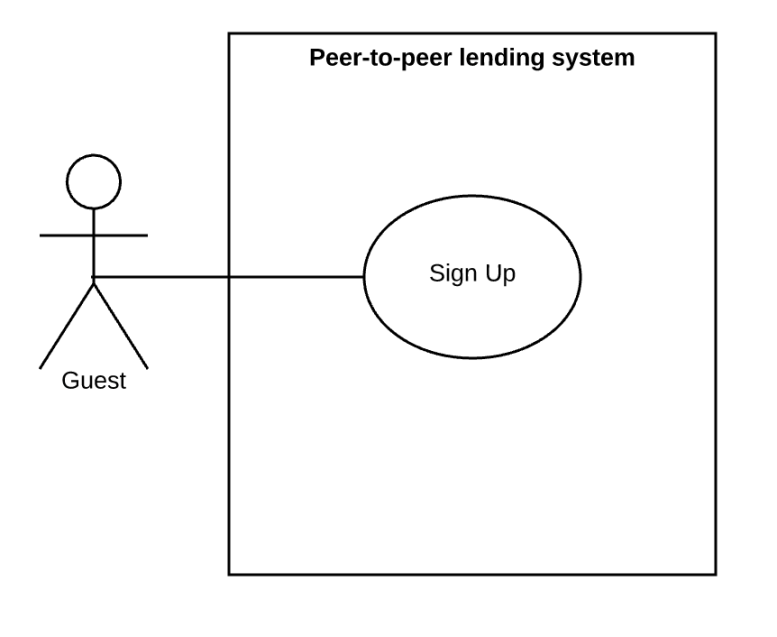
****

Figure 6 – Guest Sign Up Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_GU\_04** | | | |
| **Use Case No.** | PPLS\_GU\_04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Sign up | | |
| **Author** | LocHV | | |
| **Date** | 25/05/2019 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * Allow guest to sign up an account in the system.   **Goal:**   * Guests create an account to become a user of the system.   **Triggers:**   * Guests send sign up command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guests create a new account successfully. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest goes to register page. | The system requires identity information from Guest:  - Firstname: text input.  - Lastname: text input.  - Email: text input  - Username: text input.  - Password: text input.  - Confirm password: text input.  - Phone number: text input | | 2 | Guest inputs information. |  | | 3 | Guest sends a command to register a new account. | Guest create new account [Alternative 1] [Alternative 2] [Alternative 3] [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Duplicate username or email | The system shows an error message “This username has been used. Please choose another username.” | | 2 | Guest enter the wrong format. | The system shows the appropriate validating message. Ex: “Wrong email format,” “Password does not match,” “Phone number must be 10 numbers”, “firstname only contain alphabet character”. | | 3 | Guest does not input all information | The system shows the appropriate validating message “Please fill out this field”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | The system shows the message the "Please check your connection!" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can use the account to login in the system. | | | |

* + 1. **<Authorized User> Overview Use Case**

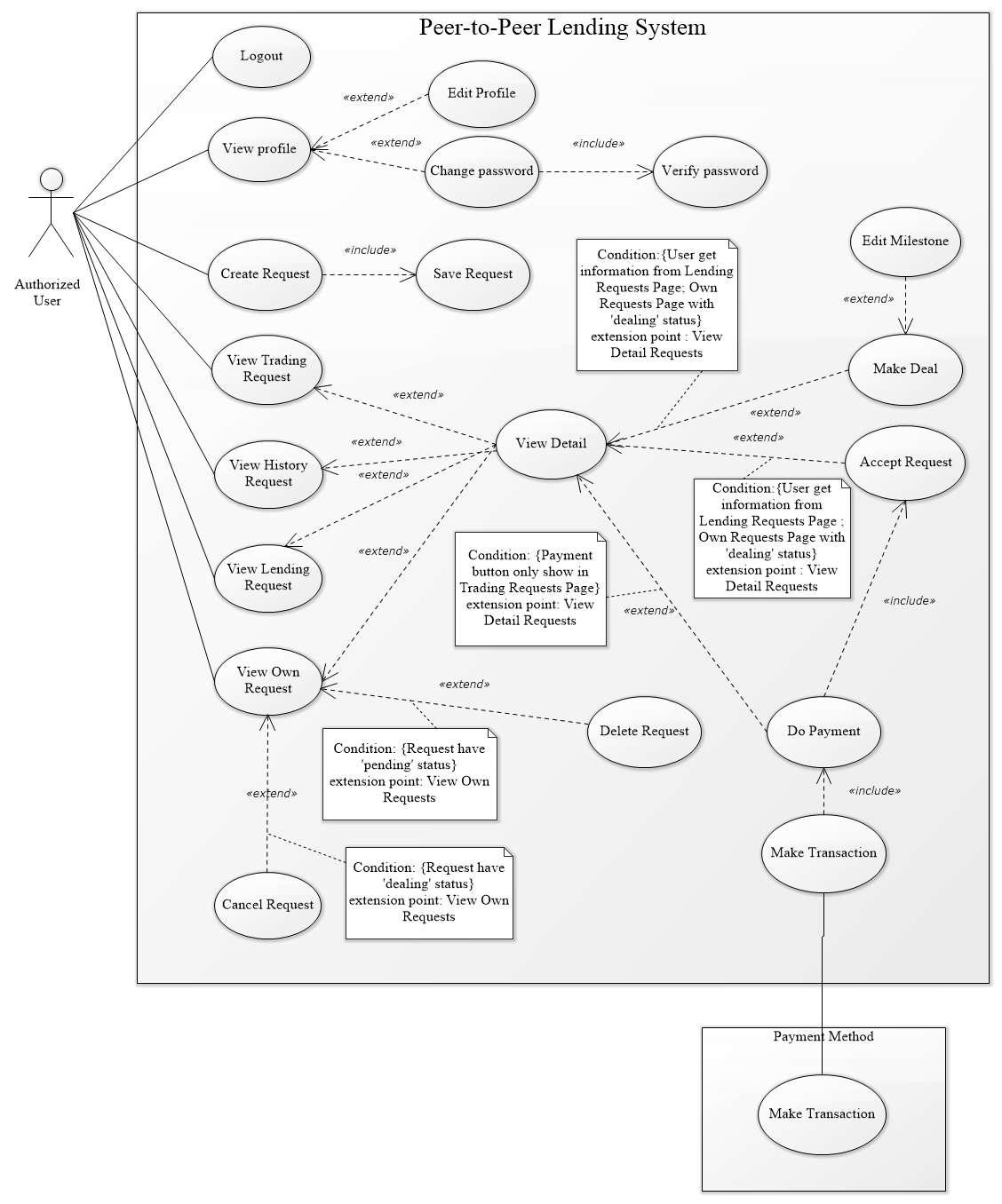


Figure 7 – Authorized User Overview Use Case

* + - 1. **<Authorized User> Logout**

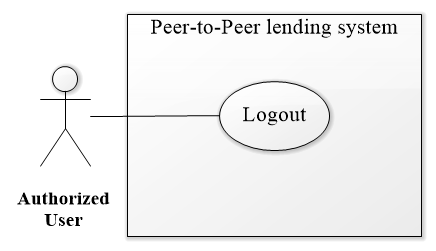


Figure 8 – <Authorized User> Logout Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_01** | | | |
| **Use Case No.** | PPLS\_AU\_01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * Authorized User can log out.   **Goal:**   * Authorized User can log out.   **Triggers:**   * Authorized User sends log out command.   **Preconditions:** N/A.  **Post conditions:**   * Success: Authorized User log out and back to view transaction page. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click logout button | System redirect to view transaction page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:** N/A  **Business Rules:**   * System will remove user data in localStorage | | | |

* + - 1. **<Authorized User> View Profile**

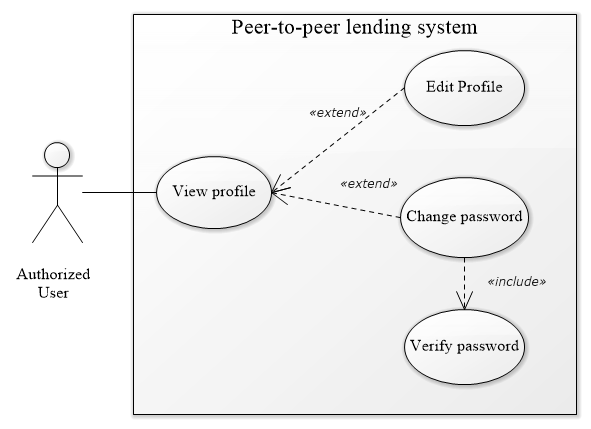
****

Figure 8 – <Authorized> View Profile Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_02** | | | |
| **Use Case No.** | PPLS\_AU\_02 | **Use Case Version** | 1.0 |
| **Use Case Name** | View profile | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User   **Summary:**   * Allow Authorized User can view their profile and their identification that uploaded.   **Goal:**   * Authorized User check information that they are uploaded in the system.   **Triggers:**   * Authorized User sends view profile command.   **Preconditions:** Must logged in as authorized user  **Post conditions:**   * Success: They can view the information in view profile page. * Fail: System shows error messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on view profile button | System redirect to view profile page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login page when token timeout |   **Relationships:** PPLS\_AU\_03, PPLS\_AU\_04  **Business Rules:**   * N/A | | | |

* + - 1. **<Authorized User> Edit Profile**

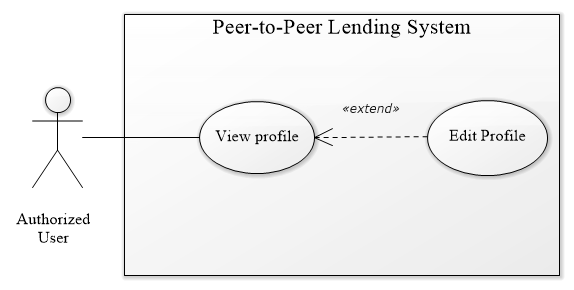
****

Figure 9 – <Authorized User> Edit Profile Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_03** | | | |
| **Use Case No.** | PPLS\_AU\_03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user edit information in their profile.   **Goal:**   * Authorized user can change their information if their information had changed.   **Triggers:**   * Authorized user sends edit profile command.   **Preconditions:**   * Must logged in as Authorized User.   **Post conditions:**   * Success: Authorized User can edit profile. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click button edit profile in view profile page. | System shows edit profile form | | 2 | Authorized User edits their profile. |  | | 3 | Authorized User click save profile | Show dialog save success  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login page when token timeout |   **Relationships:** PPLS\_AU\_02  **Business Rules:**   * N/A. | | | |

* + - 1. **<Authorized User> Change Password**

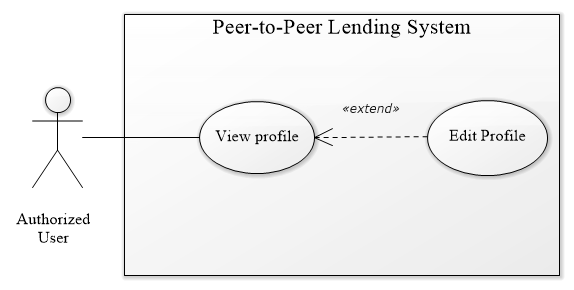
****

Figure 10 – <Authorized User> Change Password Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_04** | | | |
| **Use Case No.** | PPLS\_AU\_04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change Password | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user change password in their profile.   **Goal:**   * Authorized user can change their password.   **Triggers:**   * Authorized user sends change password command.   **Preconditions:**   * Must logged in as Authorized User.   **Post conditions:**   * Success: Authorized User can change their password. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click button change password in view profile page. | System shows change password form:  - Old password: input type password  - New password: input type password  - confirm password: input type password | | 2 | Authorized User change their password. |  | | 3 | Authorized User click save password | Show dialog save success and redirect to login page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login page when token timeout |   **Relationships:** PPLS\_AU\_02  **Business Rules:**   * N/A. | | | |

* + - 1. **<Authorized User> Create Request**

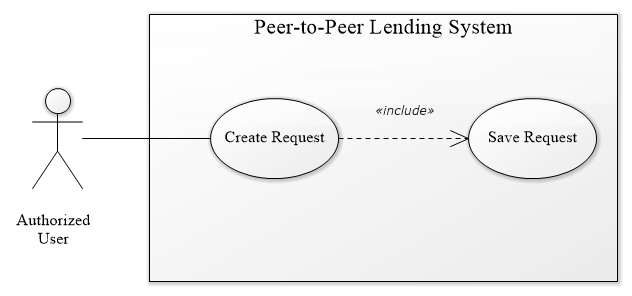


Figure 11 <Authorized User> Create Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_05** | | | |
| **Use Case No.** | PPLS\_AU\_05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Create request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user create borrow request.   **Goal:**   * Authorized User can create request to borrow money.   **Triggers:**   * Authorized User sends create request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User create a borrow request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Go to create request page. | System show create request form:  - Amount: Input type number  - Borrow duration: Input type select  - Interest rate: text fix  - Lending timeline  - Payback timeline | | 2 | Authorized user enter information into the form | [Alternative 1] [Alternative 2] | | 3 | Authorized user click create request button | System show dialog success and redirect to view own request page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Guest enter wrong format. | System shows appropriate validating message. Ex: “Amount is only contains number”. | | 2 | Guest not input all information | System shows appropriate validating message “Please fill out this field”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login page when token timeout |   **Relationships:** N/A  **Business Rules:**   * Amount less than 1.000.000.000 VND * Interest rate 18% per year | | | |

* + - 1. **<Authorized User> View Own Request**

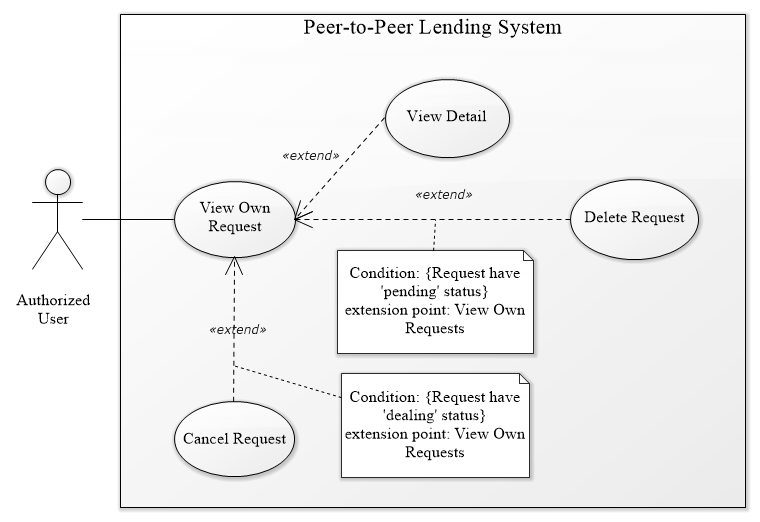
****

Figure 12 <Authorized User> View Own Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_06** | | | |
| **Use Case No.** | PPLS\_AU\_06 | **Use Case Version** | 1.0 |
| **Use Case Name** | View own request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to view their request include pending request (new request) and dealing request (request is dealing).   **Goal:**   * Authorized User can view their request.   **Triggers:**   * Authorized User sends view own request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can view their request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User go to view own request. | System show view own request page with 2 tabs include pending request and dealing request  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_07, PPLS\_AU\_08, PPLS\_AU\_15  **Business Rules:**   * N/A | | | |

* + - 1. **<Authorized User> Delete Request**

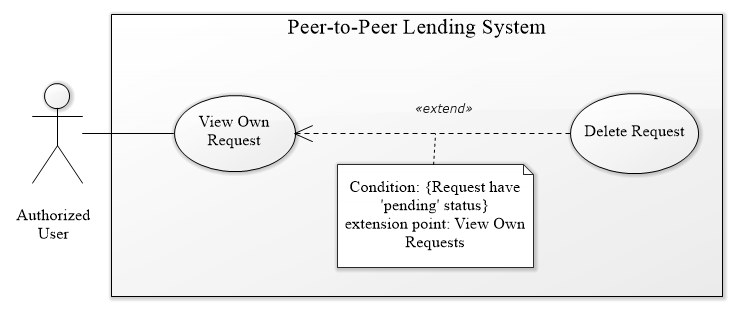
****

Figure 13 <Authorized User> Delete Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_07** | | | |
| **Use Case No.** | PPLS\_AU\_07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to delete their pending request (new request).   **Goal:**   * Authorized User can delete their request.   **Triggers:**   * Authorized User sends delete request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can delete their request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User clicks on delete button. | System show Yes/No dialog | | 2 | Authorized User click on yes button | System show success dialog  [Alternative 1][Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Authorized User click on no button | System hide Yes/No dialog |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirects to login when token timeout |   **Relationships:** PPLS\_AU\_06  **Business Rules:**   * Authorized User can delete their pending request (new request) * Authorized User cannot delete other request. | | | |

* + - 1. **<Authorized User> Cancel Request**

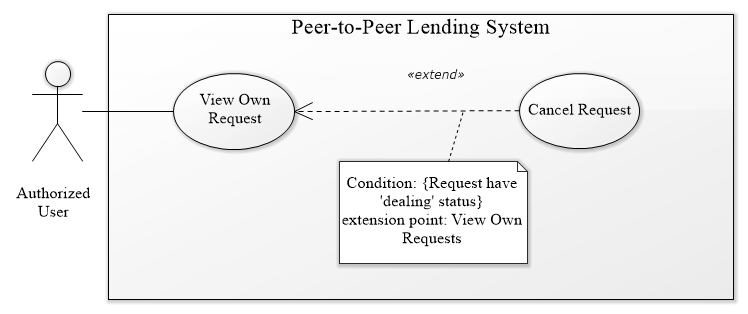


Figure 14 <Authorized User> Cancel Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_08** | | | |
| **Use Case No.** | PPLS\_AU\_08 | **Use Case Version** | 1.0 |
| **Use Case Name** | Cancel Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to cancel their dealing request (request is dealing).   **Goal:**   * Authorized User can cancel their request that is dealing.   **Triggers:**   * Authorized User sends cancel request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can cancel their request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on cancel button. | System show Yes/No dialog | | 2 | Authorized User click on yes button | System show success dialog  [Alternative 1][Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Authorized User click on no button | System hide Yes/No dialog |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_06  **Business Rules:**   * Authorized User can cancel their dealing request (request is dealing) * Authorized User cannot cancel other request. | | | |

* + - 1. **<Authorized User> View Trading Request**

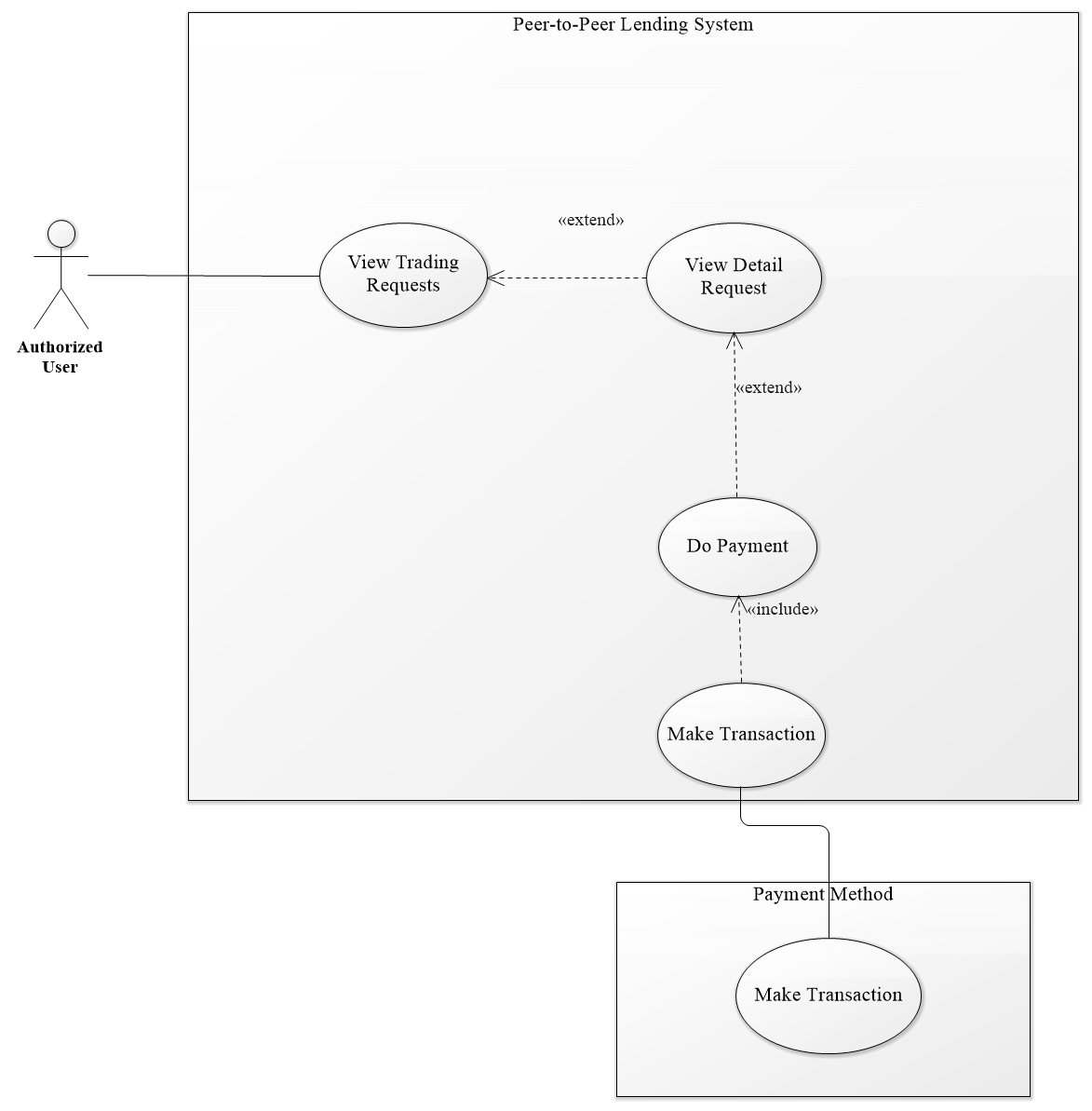
****

Figure 15 <Authorized User> View Trading Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_09** | | | |
| **Use Case No.** | PPLS\_AU\_09 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Trading Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to view their trading request (request is in process).   **Goal:**   * Authorized User can view their request that is in process.   **Triggers:**   * Authorized User sends view trading request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can view their trading request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User go to view trading request page. | System show trading request with 2 tabs include borrow and lend request  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15, PPLS\_AU\_12  **Business Rules:**   * N/A | | | |

* + - 1. **<Authorized User> View History Request**

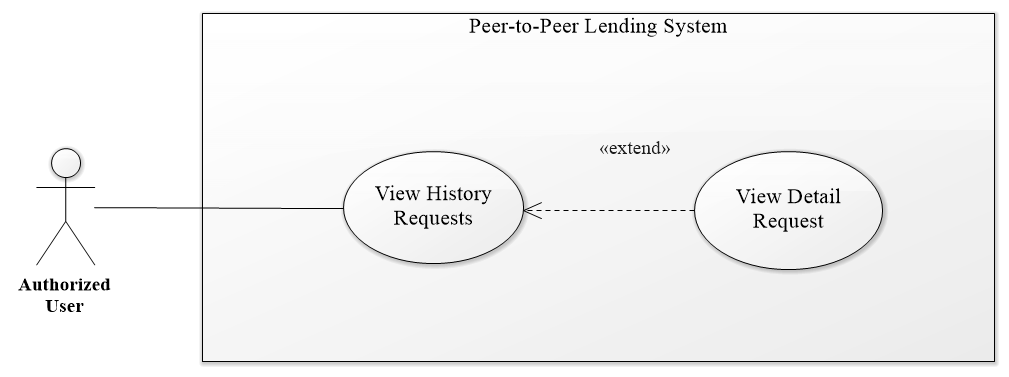
****

Figure 16 <Authorized User> View History Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_10** | | | |
| **Use Case No.** | PPLS\_AU\_10 | **Use Case Version** | 1.0 |
| **Use Case Name** | View History Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to view history request (Request had done).   **Goal:**   * Authorized User view request that had done.   **Triggers:**   * Authorized User sends view history request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can view history request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User go to view history request. | System show view history request page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15  **Business Rules:**   * N/A | | | |

* + - 1. **<Authorized User> View Lending Request**

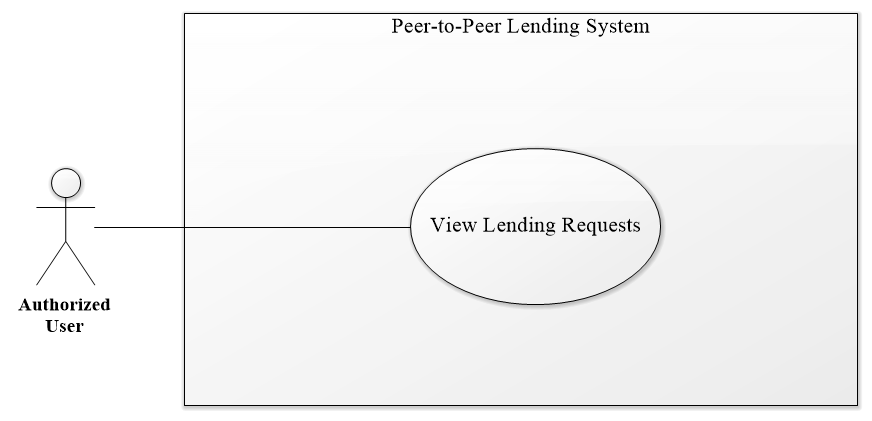
****

Figure 17 <Authorized User> View Lending Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_11** | | | |
| **Use Case No.** | PPLS\_AU\_11 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Lending Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to view the other request (borrow request from the other user).   **Goal:**   * Authorized User view borrow request.   **Triggers:**   * Authorized User sends view lending request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can view lending request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User go to view lending request. | System show view lending request page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15  **Business Rules:**   * N/A | | | |

* + - 1. **<Authorized User> Do Payment**

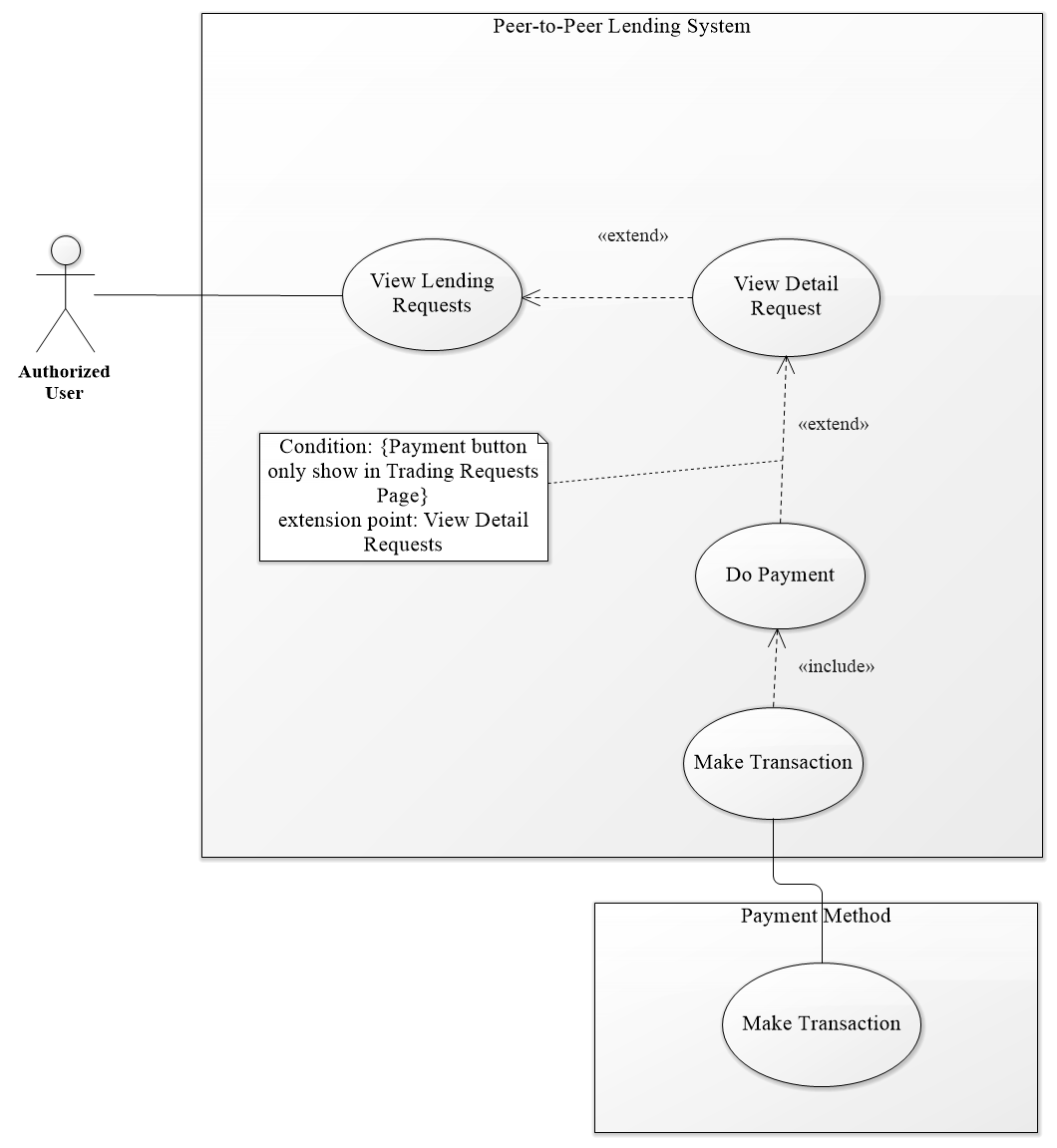
****

Figure 18 <Authorized User> Do Payment Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_12** | | | |
| **Use Case No.** | PPLS\_AU\_12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Do Payment | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to pay transaction.   **Goal:**   * Authorized User pays their transaction (lend, pay back).   **Triggers:**   * Authorized User sends do payment command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User can pay their transaction. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on payment button(paypal button) | System show paypal page. | | 2 | Authorized User login to paypal using paypal account | Paypal show payment form | | 3 | Authorized User click on pay now button | System show success dialog |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15  **Business Rules:**   * Do payment use to accept request and pay money for trading request. | | | |

* + - 1. **<Authorized User> Accept Request**

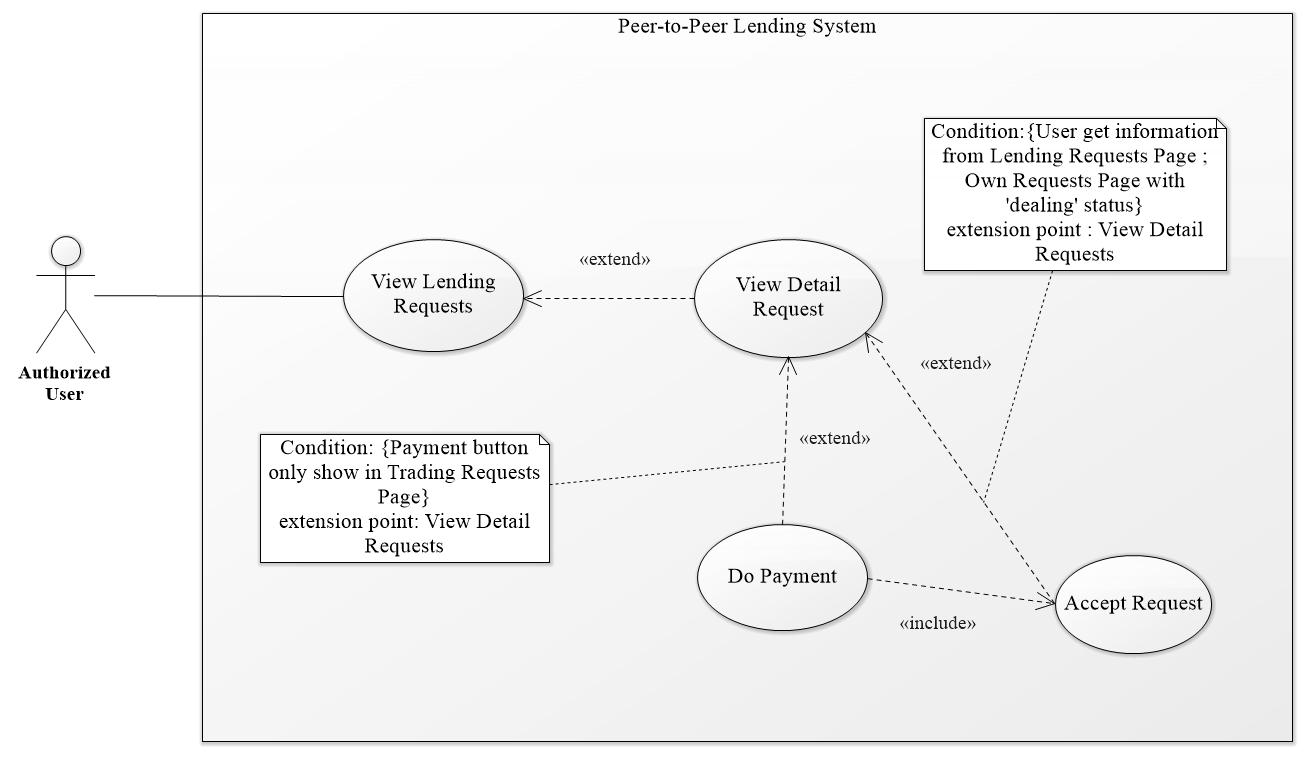
****

Figure 19 <Authorized User> Accept Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_13** | | | |
| **Use Case No.** | PPLS\_AU\_13 | **Use Case Version** | 1.0 |
| **Use Case Name** | Accept request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to accept other user request.   **Goal:**   * Authorized User accepts borrow request from other user.   **Triggers:**   * Authorized User sends accept request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User accept request and request change to trading status. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on accept request | System show accept dialog. | | 2 | Authorized User do payment | System show success dialog and redirect to view trading request page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15, PPLS\_AU\_12  **Business Rules:**   * Authorized User can accept dealing request and pending request. * Authorized User cannot accept other request. | | | |

* + - 1. **<Authorized User> Make Deal**

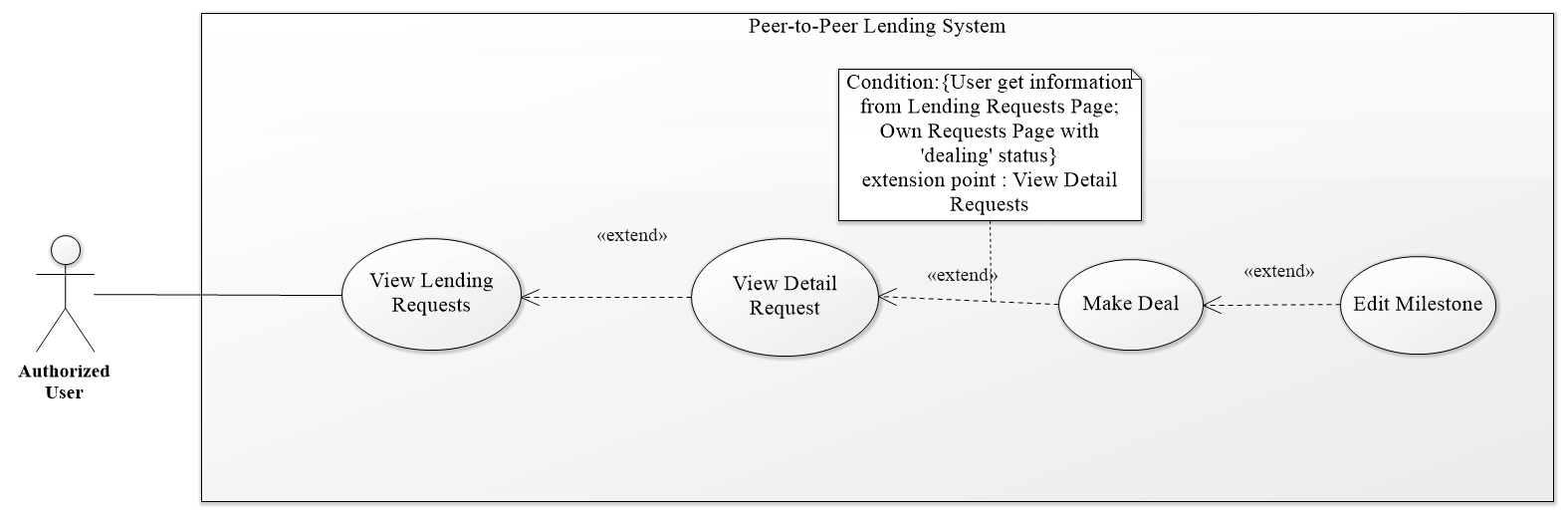
****

Figure 20 <Authorized User> Make Deal Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_14** | | | |
| **Use Case No.** | PPLS\_AU\_14 | **Use Case Version** | 1.0 |
| **Use Case Name** | Accept request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to edit borrow request to make a deal with other user.   **Goal:**   * Authorized User negotiate borrow request with other user.   **Triggers:**   * Authorized User sends make deal command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User make deal with other user. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on make deal button | System show edit borrow request form.  - Borrow name: text  - Total amount: text  - Borrow amount: text  - Borrow duration: select (Editable)  - Borrow day: text  - Create Day: text  - Payment Deadline: input date (Editable)  - Interest rate: text  - Interest Received: text  - Lending timeline (Editable)  - Payback timeline (Editable) | | 2 | Authorized User edit form |  | | 3 | Authorized User click on save deal button | System show Yes/No dialog | | 4 | Authorized User click on Yes button | System hide Yes/No dialog  [Alternative 1][Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Authorized User click on No button | System hide Yes/No dialog |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AU\_15, PPLS\_AU\_12  **Business Rules:**   * Authorized User can make a deal with pending request. * Authorized User after make deal will wait for response (other user can make deal or accept request) | | | |

* + - 1. **<Authorized User> View Detail Request**

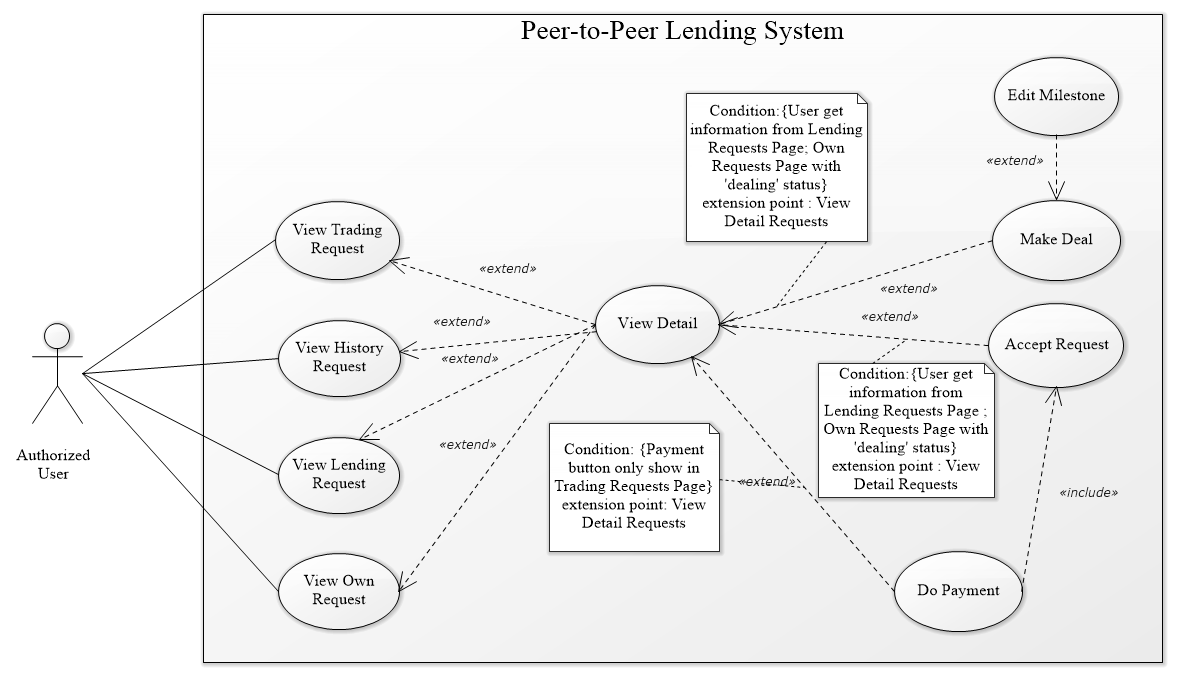
****

Figure 21 <Authorized User> View Detail Request Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AU\_15** | | | |
| **Use Case No.** | PPLS\_AU\_15 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Detail Request | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Authorized User.   **Summary:**   * Allow authorized user to view detail borrow request.   **Goal:**   * Authorized User can view detail borrow request to make decision ignore, make deal or accept this request.   **Triggers:**   * Authorized User sends view detail request command.   **Preconditions:**   * Must logged in as Authorized User   **Post conditions:**   * Success: Authorized User view detail borrow request. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Authorized User click on view detail button | System shows borrow request page.  - Borrow name: text  - Total amount: text  - Borrow amount: text  - Borrow duration: text  - Borrow day: text  - Create Day: text  - Payment Deadline: text  - Interest rate: text  - Interest Received: text  - Lending timeline  - Payback timeline  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirects to login when token timeout |   **Relationships:** PPLS\_AU\_06, PPLS\_AU\_09, PPLS\_AU\_10, PPLS\_AU\_11, PPLS\_AU\_12, PPLS\_AU\_13, PPLS\_AU\_14  **Business Rules:**   * N/A | | | |

* + 1. **<Admin> Overview Use Case**

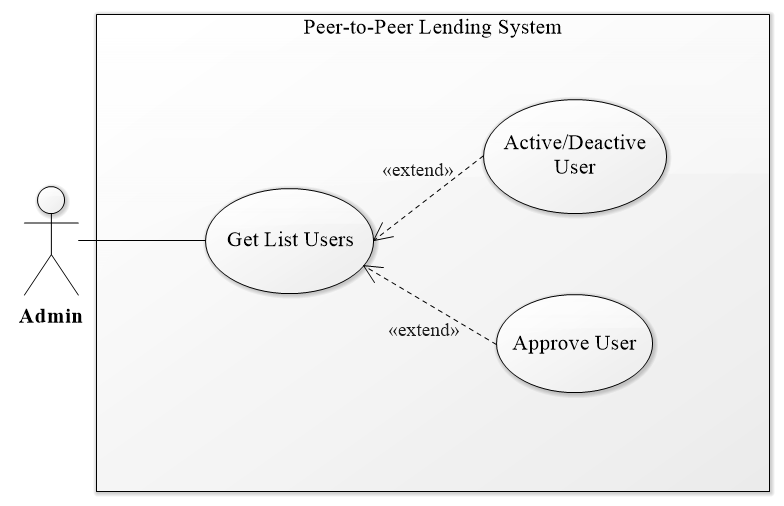
****

Figure 22 <Admin> Overview Use Case

* + - 1. **<Admin> Get List Users**

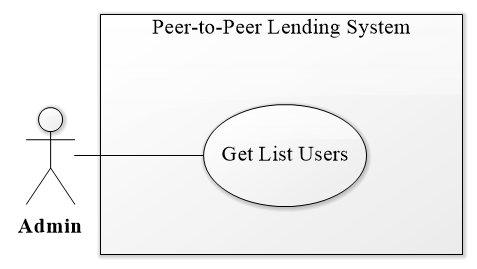
****

Figure 23 <Admin> Get List Users Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AD\_01** | | | |
| **Use Case No.** | PPLS\_AD\_01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Get list users | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * Allow admin to view list of users.   **Goal:**   * Admin can view list of users.   **Triggers:**   * Admin sends get list users command.   **Preconditions:**   * Must logged in as Admin   **Post conditions:**   * Success: Admin view list of users. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin go to view user list | System shows view user list page.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 |  | System redirect to login when token timeout |   **Relationships:** PPLS\_AD\_02, PPLS\_AD\_03  **Business Rules:**   * N/A | | | |

* + - 1. **<Admin> Activate / Deactivate User**

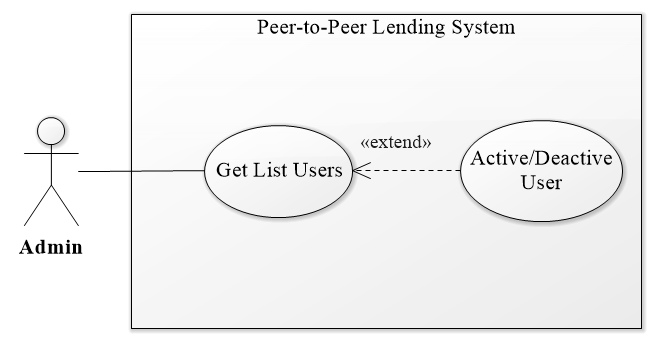
****

Figure 24 <Admin> Activate/Deactivate User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AD\_02** | | | |
| **Use Case No.** | PPLS\_AD\_02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Activate/Deactivate User | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * Allow admin to activate or deactivate user.   **Goal:**   * Admin can activate or deactivate user if had any problems.   **Triggers:**   * Admin sends activate/deactivate user command.   **Preconditions:**   * Must logged in as Admin   **Post conditions:**   * Success: Admin activate or deactivate user. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin click on toggle button (activate) | Toggle button will activate.  [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Admin click on toggle button (deactivate) | Toggle button will deactivate. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:** PPLS\_AD\_01  **Business Rules:**   * N/A | | | |

* + - 1. **<Admin> Approve User**

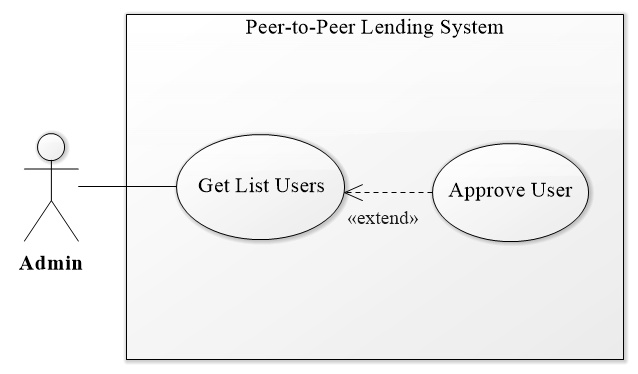
****

Figure 25 <Admin> Approve User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – PPLS\_AD\_03** | | | |
| **Use Case No.** | PPLS\_AD\_03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Approve User | | |
| **Author** | LocHV | | |
| **Date** | 22/05/2019 | **Priority** | Normal |
| **Actor:**   * Admin.   **Summary:**   * Allow admin to approve user.   **Goal:**   * Admin can approve user.   **Triggers:**   * Admin sends approve user command.   **Preconditions:**   * Must logged in as Admin   **Post conditions:**   * Success: Admin approve user. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Admin click on toggle button (activate) | Toggle button will activate.  [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Admin click on toggle button (deactivate) | Toggle button will deactivate. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | |  |  |  |   **Relationships:** PPLS\_AD\_01  **Business Rules:**   * N/A | | | |