**SOFTWARE REQUIRMENT SPECIFICATION**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**SOFTWARE REQUIRMENT SPECIFICATION**

|  |  |
| --- | --- |
| Author | Lam Nguyen |
| Date | 30-07-2018 |
| Version | 0.2 |
| Status | Reviewing |
| Project | Give and Take Application |

**TABLE OF CONTENTS**

[1 INTRODUCTION 4](#_Toc520818126)

[1.1 Purpose 4](#_Toc520818127)

[1.2 Scope 4](#_Toc520818128)

[1.3 Abbreviation 4](#_Toc520818129)

[2 REFERENCED DOCUMENTS 5](#_Toc520818130)

[2.1 Controlling Documents 5](#_Toc520818131)

[2.2 Controlled Documents 5](#_Toc520818132)

[2.3 Background Information 5](#_Toc520818133)

[3 OVERVIEW 6](#_Toc520818134)

[3.1 Actors 6](#_Toc520818135)

[4 FEATURES 7](#_Toc520818136)

[4.1 Device Orientation 7](#_Toc520818137)

[4.2 Start up 7](#_Toc520818138)

[4.2.1 User Interface 7](#_Toc520818139)

[4.2.2 Business Logic 7](#_Toc520818140)

[4.3 Login 8](#_Toc520818141)

[4.3.1 User Interface 8](#_Toc520818142)

[4.3.2 Business Logic 8](#_Toc520818143)

[4.3.2.1 Login using Facebook account 8](#_Toc520818144)

[4.3.2.2 Login using Google+ account 8](#_Toc520818145)

[4.3.2.3 Remember credentials 8](#_Toc520818146)

[4.3.3 Use Cases 9](#_Toc520818147)

[4.4 Post Management 10](#_Toc520818148)

[4.4.1 User Interface 10](#_Toc520818149)

[4.4.2 Business Logic 10](#_Toc520818150)

[4.4.3 Use Cases 11](#_Toc520818151)

[4.5 Request Management 12](#_Toc520818152)

[4.5.1 User Interface 12](#_Toc520818153)

[4.5.2 Business Logic 12](#_Toc520818154)

[4.5.3 Use Cases 12](#_Toc520818155)

[4.6 Conversation Management 14](#_Toc520818156)

[4.6.1 User Interface 14](#_Toc520818157)

[4.6.2 Business Logic 14](#_Toc520818158)

[4.7 Report management 14](#_Toc520818159)

[4.7.1 User Interface 14](#_Toc520818160)

[4.7.2 Business Logic 14](#_Toc520818161)

[4.7.2.1 User report a post 14](#_Toc520818162)

[4.7.2.2 Admin manage a list of reported posts 14](#_Toc520818163)

[4.7.3 Use cases 15](#_Toc520818164)

[4.8 Category Management 16](#_Toc520818165)

[4.8.1 User Interface 16](#_Toc520818166)

[4.8.2 Business Logic 16](#_Toc520818167)

[4.8.2.1 Admin manage category 16](#_Toc520818168)

[4.8.2.2 User choose a category to filtrate the posts 16](#_Toc520818169)

[4.9 Feedbacks Management 17](#_Toc520818170)

[4.9.1 User Interface 17](#_Toc520818171)

[4.9.2 Business Logic 17](#_Toc520818172)

[4.9.2.1 Quest create feedbacks 17](#_Toc520818173)

[4.9.2.2 Admin read feedbacks 17](#_Toc520818174)

[4.9.3 Use cases 17](#_Toc520818175)

[4.10 Ranking system 18](#_Toc520818176)

[4.10.1 User Interface 18](#_Toc520818177)

[4.10.2 Business Logic 18](#_Toc520818178)

[4.10.2.1 Ranking system for Giver 18](#_Toc520818179)

[4.10.2.2 Ranking system for Post 18](#_Toc520818180)

[5 Non-functional Requirements 19](#_Toc520818181)

[5.1 Operating System Support 19](#_Toc520818182)

[6 ANALYZE AND DESIGN 20](#_Toc520818183)

[6.1 Activity Diagram 20](#_Toc520818184)

[6.1.1 Login 20](#_Toc520818185)

[6.1.2 Post Management 23](#_Toc520818186)

[6.1.3 Request Management 26](#_Toc520818187)

[6.1.4 Report Management 29](#_Toc520818188)

[6.1.5 Category Management 30](#_Toc520818189)

[6.1.6 Feedbacks Management 33](#_Toc520818190)

[6.2 Class Diagram 34](#_Toc520818191)

[6.3 Database 34](#_Toc520818192)

[7 APPENDIX A: [TBD] 34](#_Toc520818193)

**CHANGE HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 09-07-2018 | Thanh Vo | Initial version |
| 0.2 | 30-07-2018 | Lam Nguyen | Update SRS |

1. **INTRODUCTION**
   1. **Purpose**

The purpose of this document is to describe the requirements and specification for the Give and Take application on multiplatform.

* 1. **Scope**

This document is intended for anyone in direct relation to the Give and Take application.

* 1. **Abbreviation**

|  |  |
| --- | --- |
| **Abbreviation** | **Explanation** |
| SRS | Software Requirement Specification |
| CMS | Content Management System |
| SQS | Sioux Quality System |

1. **REFERENCED DOCUMENTS**
   1. **Controlling Documents**

No dependencies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Document | Author | Version | Date |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. **Controlled Documents**

No controlled documents

|  |  |  |
| --- | --- | --- |
| Id | Document | Author |
|  |  |  |
|  |  |  |

* 1. **Background Information**

The following documents are relevant to the context of the document but do not affect the contents in a direct way:

[SQS]: Sioux Quality System

Version 3.5  
August 7, 2012

1. **OVERVIEW**

“Give and Take” is an application where people can share their old items to other people, and also can find items they need. Especially, it’s also a place where charity organizations and philanthropists can meet together and deliver useful items to poor people.



In “Give and Take”, there are **Givers** who posts and gives their items to other people, and **Takers** who receives items from Givers. Givers upload their items (names, images) into **public posts** (posts can be seen by anyone who installed “Give and Take” app), and Takers will send requests to receive them.

Besides that, there is an **admin** taking responsibility to manage the content of the app including: Posts, Categories, and Users; and receive report and feedback to improve the app.

“Give and Take” application will make giving items and doing charity more easily, more controllable and more fun. “Let’s give a lot and receive more than this, with Give and Take”.

* 1. **Actors**

The following table gives an overview of the different human and non-human actors that interact with the system. The name between parentheses is the formal name further used in this document.

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Admin | Who control CMS administration page |
| User | The user who can be both Giver and Taker in system |
| Guest | Who use the system but are not login. |

1. **FEATURES**
   1. **Device Orientation**

**Give and Take App** supports Portrait mode for supported devices:

iPhone with the ***minimum*** ***iOS 11.0***, and android phones with the ***minimum android 5.1****,*multi-touch display with LED backlight and IPS technology.

For ***iPhone***: the minimum model supported is **iPhone 5s** ***(Performance issues are acceptable)*** and the ***Standard Testing Device*** is **iPhone 6s**.

* 1. **Start up**
     1. **User Interface**

For iPhone

*[TBD]*

For Android:

*[TBD]*

* + 1. **Business Logic**

Splash screen is the first screen to show whenever user opens the application.

Splash screen remains during loading progress and close after the loading finished.

Once the app is finished loading, it will show the new feeds (the screen showing newest public posts).

* 1. **Login**
     1. **User Interface**

For iPhone

*[TBD]*

For Android:

*[TBD]*

* + 1. **Business Logic**

By default, guests can only view public posts. When they want to create a new post/item or send a request to Givers, they must login first.

* + - 1. **Login using Facebook account**

The user login to “Give and Take” app using their Facebook account

* If the device is not yet log in to Facebook, the user must login to Facebook and they must confirm the privacy problem of Facebook. If the user agrees, redirect back to the main screen.
* If the device is logged in to Facebook, skip the “log in to Facebook” step, redirect to the main screen.
* If the user account already synchronized with “Give and Take” app, skip “Confirm privacy” step.
  + - 1. **Login using Google+ account**

The user login to “Give and Take” app using their Google+ account

* They must confirm privacy problem of Google, which will display by the user’s phone browser. If the user agrees, redirect back to the main screen.
* If the user wants to use another Google+ account, they must choose “Login to another account”, then fill all the mandatory fields needed, and use that account to login. After that, redirect to confirm privacy step.
* If the user account already synchronized with “Give and Take” app, skip “Confirm privacy” step.
  + - 1. **Remember credentials**

It is possible to remember the credentials. In this case the credentials information (Facebook/Google account that linked with app) will be used for next login purpose. User will not need to “Sign in with Facebook” or “Sign in with Google+” again, but the App will be logged in automatically.

* + 1. **Use Cases**

|  |  |  |
| --- | --- | --- |
| Use Case | UC1: Login using Facebook account | |
| Actors | **Guest** |  |
| Pre-conditions | N/A | |
| Post-conditions | Application successfully synchronize with the user Facebook account | |
| Main success scenario | 1. The user’s phone have logged in to Facebook. 2. The user confirms privacy. 3. The application receive access token of the user’s Facebook account from Facebook services. 4. The application gets all the information of the user. 5. Redirect to the main screen. | |
| Alternative scenario 1 | Step 1: The user’s phone not yet logged in to Facebook.   * + 1. The user must login to Facebook. Then continue from main step 2. | |
| Alternative scenario 2 | Step 2: The user don’t confirm privacy.  2.2.1) Redirect to the login screen and start from main step 1. | |
| Alternative scenario 3 | Step 2: The user’s Facebook account has already synchronize with “Give and Take” application  2.3.1) The application updates all the user information through their Facebook account and continues from main step 4. | |
| Business Rules |  | |
| Other requirements | Time of launching app is as short as possible (3 - 5s) | |
| Open issues |  | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC2: Login using Google+ account | |
| Actors | **Guest** |  |
| Pre-conditions | N/A | |
| Post-conditions | Application successfully synchronize with the user Google+ account | |
| Main success scenario | 1. The user uses their existing Google+ account. 2. The user confirms privacy. 3. The application receive access token of the user’s Google+ account from Google services. 4. The application gets all the information of the user. 5. Redirect to the main screen. | |
| Alternative scenario 1 | Step 1: The user want to use different Google+ account.   * + 1. The user must login to Google+ using their different account, continues to main step 2. | |
| Alternative scenario 2 | Step 2: The user don’t confirm privacy.  2.2.1) Redirect to the login screen, and start from main step 1. | |
| Alternative scenario 3 | Step 2: The user’s Google+ account has already synchronize with “Give and Take” application  2.3.1) The application update all the user information through their Google+ account and continue from main step 4. | |
| Business Rules |  | |
| Other requirements | Time of launching app is as short as possible (3 - 5s) | |
| Open issues |  | |

* 1. **Post Management**
     1. **User Interface**
     2. **Business Logic**

Public posts screen is loaded after the application is loaded successfully. Users/guests can see all the posts that givers have posted. Besides that, users/guests can use filters to classify the posts to see what they concern. After users/guests choose a filter, the posts will be filtrated and sorted. There are 3 filters:

* Category (default is “All category”)
* Sort by (Like number or Time)
* Location (default is “Da Nang”) (In the first version, we don’t allow users to select this filter).

Givers can see all their posts.

Givers can create a new post by filling information: title, description, location and photos about the items they want to give. They can import the photos from their gallery or take a new picture. After they submit, this post will be create and update to system so all users/guests can see.

Givers modify their posts by changing post’s fields: title, description, location, state, add or remove photos. Givers can update description, image, location, and category to inform the status of items in the post to Takers. Besides that, after Givers send all their giving items to takers, Givers can update the status of their post to “Closed”, so Takers can aware that the post is no longer opening to send a request. After they submit the changes, a notification will be sent to all takers who sent the request to the post.

A post has 2 statuses:

* “Opening” means that this post is giving and Takers can make request to it.
* “Closed” means that all items in this post is gave and Takers can’t make request to it. If the post has been delivered to Takers, the Giver can change status of the post into “Closed”. As a result, all requests, not accepted yet, will be refused.

Users can delete their posts. If users want to delete a post which is in “Opening” status and has requests, they must confirm a warning: “This post has been requested by takers. Are you sure you want to delete this post? ”.

If users find a post containing useful items, they can like it, give a comment about it or even share it on both Facebook and Google+, so their friends on Facebook/ Google+ can be able to know about it and can open the post in our application if they have already installed app in their phone. Or requiring install app to see the post if they have not installed app yet.

* + 1. **Use Cases**

|  |  |  |
| --- | --- | --- |
| Use Case | UC1: Giver Create post | |
| Actors | **Giver** |  |
| Pre-conditions | The user have logged in successfully | |
| Post-conditions | The post has been created successfully | |
| Main success scenario | 1. Givers “Create new post”. 2. Givers fill the post’s information: title, description, location, categories. 3. Givers “Submit”. 4. The post is created and changes into “Opening” status, so all people can see it. | |
| Alternative scenario 1 | Step2) Givers let one of mandatory fields empty   1. “Give and Take” app shows error signal for each field and user can’t “Submit” before filling all mandatory fields. | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC2: Giver modify post | |
| Actors | **Giver** |  |
| Pre-conditions | The user have logged in successfully | |
| Post-conditions | The post has been modified successfully and notification is sent to all requesters. | |
| Main success scenario | 1. Givers select a post to modify 2. Givers choose “Edit”. 3. Givers edit information of the post. 4. Givers “Save”. 5. The post is updated. 6. The Give and Take server sends notifications to all requesters | |
| Alternative scenario 1 | Step6) The updated post doesn’t have any request, so there is no notification sent. | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC3: Giver delete post | |
| Actors | **Giver** |  |
| Pre-conditions | The user have logged in successfully | |
| Post-conditions | The post has been deleted successfully | |
| Main success scenario | 1. Giver select a post to delete 2. Giver commit to “Delete”. 3. The post is deletes in system, and notification is sent to Takers | |
| Alternative scenario 1 | Step 3) The Give and Take app display an error message: “An error occurred while deleting this post.” | |
| Alternative scenario 2 | Step 3) The deleted post doesn’t have any request, so there is no notification sent. | |

* 1. **Request Management**
     1. **User Interface**
     2. **Business Logic**

A giver can accept many requests for his/ her post.

Guests can only see posts. In order to create requests, they must login in advanced.

When usersfind a post containing useful items for them, they can send a request containing their message to the giver, and the message is not allowed to be empty. User can only send the request to the posts in Opening status. After they send the request, a notification will be sent to the giver.

Givers can see all the requests of their posts including the request’s message and requester’s profile. Givers decide who they want to give. Givers accept the request by sending contact information including: telephone number (optional), address and description to the requester. Then they will negotiate by phone, by social network, or by conversation in system.

Requesters will receive notification that their request is denied when givers decline their requests or givers change the status of the post to “Closed” but didn’t accept their requests.

* + 1. **Use Cases**

|  |  |  |
| --- | --- | --- |
| Use Case | UC1: Create request to a post | |
| Actors | **Taker** |  |
| Pre-conditions | The user have logged in successfully | |
| Post-conditions | The request is created successfully | |
| Main success scenario | 1. Takers request a post they wanted 2. Takers fill in the message they want to send to the Giver 3. Takers submit the form. 4. A notification is sent to the Giver | |
| Alternative scenario 1 | Step 2) Takers let the message empty.  Takers submit the form, and receive an error message : ”Message cannot be empty” | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC2: Delete (Cancel) a request | |
| Actors | **Taker** |  |
| Pre-conditions | The user have logged in successfully | |
| Post-conditions | The request is deleted successfully | |
| Main success scenario | 1. Takers launch request management. 2. Takers delete a particular request. 3. Takers receive a warning:” Are you sure you want to delete this request?”, and confirm the warning. 4. The request is deleted in system. | |
| Alternative scenario 1 | Step 3) Takers receive a warning, and cancel the warning.  Return to the main step 1. | |

|  |  |
| --- | --- |
| Use Case | UC3: Giver claim item to Taker (Accept or Decline Requests) |
| Actors | **Giver** |
| Pre-conditions | The user have logged in successfully |
| Post-conditions | One request is accepted or declined. The reply message (notification) is sent to the chosen one successfully |
| Main success scenario | 1. Giver opens his post and check the request. 2. Giver sees the detail of a request. 3. Giver accepts the request with contact information like: telephone number (optional), address and description. 4. "Give and Take" displays list of requests again to user 5. "Give and Take" app sends notification to the Taker. |
| Alternative scenario 1 | 1. Givers can go to the post based on notification.    1. Giver continues main step 2. |
| Alternative scenario 2 | 1. Giver accepts the request without see the detail. 2. Giver continues main step 3. |
| Alternative scenario 3 | 1. Giver declines the request. 2. The request is deleted and notification is sent to the Taker. |

* 1. **Conversation Management**
     1. **User Interface**
     2. **Business Logic**

Giver and Taker can negotiate each other via conversation on system.

A user can go to another user’s profile start conversation with him/ her. Besides that, there is a quick way for a user to send a message to another user by opening his/ her small information pop up.

Message send to the user (and there is notification for this message), and the user Choose Reply or not.

User can block other user if the message is spam. If user A are blocked by user B, A still can send message to B but no notification is sent to B. Blocked Messages will be located in separate place with messages from other users.

Users can delete their conversations.

* 1. **Report management**
     1. **User Interface**
     2. **Business Logic**
        1. **User report a post**

Users can report a post that violated the terms of the App by filling the message field and submit the form. The report will be sent to admin to handle.

* + - 1. **Admin manage a list of reported posts**

In CMS, there is a list of reported posts including necessary information:

* Index
* Post’s link
* Number of report
* Detail of reports (a link)
* Name of owner
* State of report (solved or unsolved)

After admins see details of the reported posts and the message that users submitted. By basing on the level of the violation they will decide to send a warning message to the reported Giver or delete the post.

Admin can delete violating- term posts.

Admin can block a user if he post violated post many time. So a user cannot login the app anymore until admin unblock this user.

* + 1. **Use cases**

|  |  |  |
| --- | --- | --- |
| Use Case | UC1: Send warning notification to user | |
| Actors | **Admin** |  |
| Pre-conditions | Login to CMS successfully | |
| Post-conditions | N/A | |
| Main success scenario | 1. Admin Warn user 2. System displays confirm box 3. Admin confirm that he want to warn user 4. System sends a warning notification to user’s phone and updates number of warnings of that user 5. Unsolved state changes to Solved | |
| Alternative scenario 1 | Step 3: Admin don’t confirm that he want to warn the user.  3.1.1 Close confirm box. Redirect to main manage screen. | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC2: Block user | |
| Actors | **Admin** |  |
| Pre-conditions | Login to CMS successfully | |
| Post-conditions | N/A | |
| Main success scenario | 1. Admin Block User 2. System displays confirm box 3. Admin confirm that he want to block a user 4. System updates user’s state to blocked 5. Unsolved state changes to Solved | |
| Alternative scenario 1 | Step 3: Admin don’t confirm that he want to block the user.  3.1.1 Close confirm box. Redirect to main manage screen. | |

|  |  |  |
| --- | --- | --- |
| Use Case | UC3: Delete post | |
| Actors | **Admin** |  |
| Pre-conditions | Login to CMS successfully | |
| Post-conditions | N/A | |
| Main success scenario | 1. Admin Delete a post 2. System displays confirm box 3. Admin confirm that he want to delete that post 4. System removes the post and all items was attached to this post 5. System sends notification to user (information of post and all items deleted) 6. Unsolved state changes to Solved | |
| Alternative scenario 1 | Step 3. Admin don’t confirm that he want to delete that post  3.1.1 Close confirm box. Redirect to the main manage screen. | |

* 1. **Category Management**
     1. **User Interface**

[TBD]

* + 1. **Business Logic**
       1. **Admin manage category**

In CMS, categories are displayed as a list. Each row includes some necessary information like:

* Index
* Category’s name

Admin can create a new category in CMS and this category will be available in app.

Admin can modify, delete categories. After deleting, the category will be removed in system, and be invisible in app.

Admin can only delete categories which don’t have any posts.

With categories having posts, admin can only disable them. When admin disable a category having posts, all opening posts in this category are not visible in new feeds (the screen showing newest public posts) anymore.

So other people cannot request these posts. However, the owner of these post can see them it his/ her post list.

The owner can edit by moving to another category to make it visible for everyone.

If the post has requests before admin disables the category, giver still can accept requests and taker still can see that post although the category is disable.

* + - 1. **User choose a category to** **filtrate the posts**
* Category values are defined by admin.
* User chooses a particular category.
* The result will be shown with filter for this category in result screen.
  1. **Feedbacks Management**
     1. **User Interface**

[TBD]

* + 1. **Business Logic**
       1. **Quest create feedbacks**

User can give comments, feedback by filling what they think about the app, what is the advantages and disadvantages of the app, what do they think to improve the app better… to the shout box and submit it.

Feedbacks which were submitted are shown to Admin.

* + - 1. **Admin read feedbacks**

Feedbacks which were submitted are shown to Admin. Admin read feedbacks and have a plan to improve the app in the future.

* + 1. **Use cases**

|  |  |  |
| --- | --- | --- |
| Use Case | UC1: Create feedback | |
| Actors | **Quest** |  |
| Pre-conditions | N/A | |
| Post-conditions | The feedback has sent successfully | |
| Main success scenario | 1. User request to give a comment/feedback 2. User fill in comment/feedback 3. User submit to send | |
| Alternative scenario 1 | Step 3. The Give and Take app notifies message field are missing. | |

* 1. **Ranking system**
     1. **User Interface**

[TBD]

* + 1. **Business Logic**
       1. **Ranking system for Giver**

The rank of Givers is defined base on the number of appreciation from Takers.

When he receive items from giver, he can thank to giver by ranking system for giver in the post which taker requested.

The ranking system is just available for takers who are accepted from giver.

The levels of ranking can be:

|  |  |
| --- | --- |
| Name | Number of appreciations from Takers (x) |
| Thành viên | 0 < x < 50 |
| Tấm lòng bạc | 50 <= x < 150 |
| Tấm lòng vàng | 150 <= x < 300 |
| Tấm lòng bạch kim | 300 <= x < 600 |
| Tấm lòng kim cương | x >=600 |

* + - 1. **Ranking system for Post**

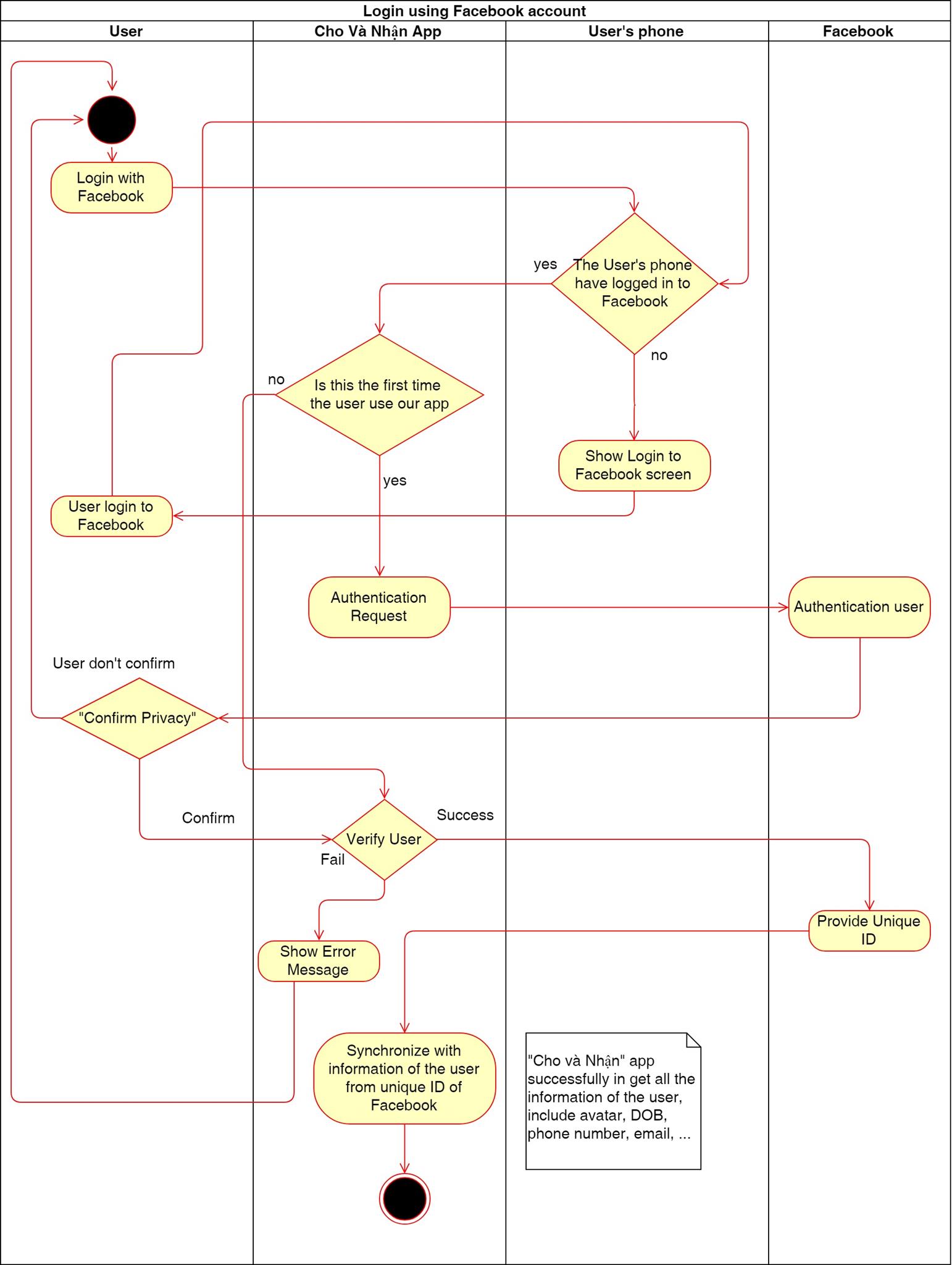
User can click Like (Facebook) for the posts they think interesting and helpful.

The post which has the most number of Likes, will be on top.

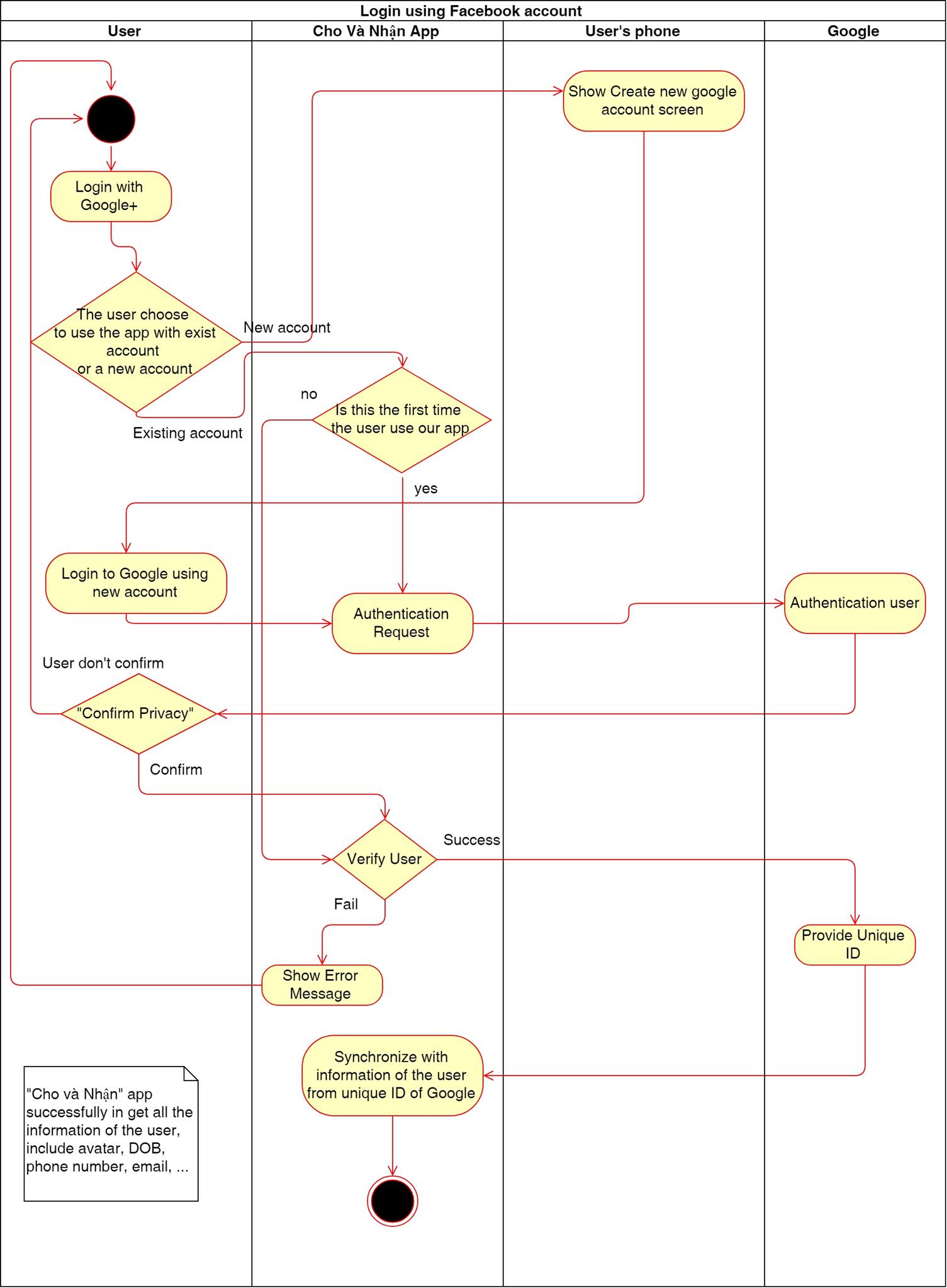
1. **Non-functional Requirements**
   1. **Operating System Support**

Give and Take App supports minimum iOS 11, Android 5.0.

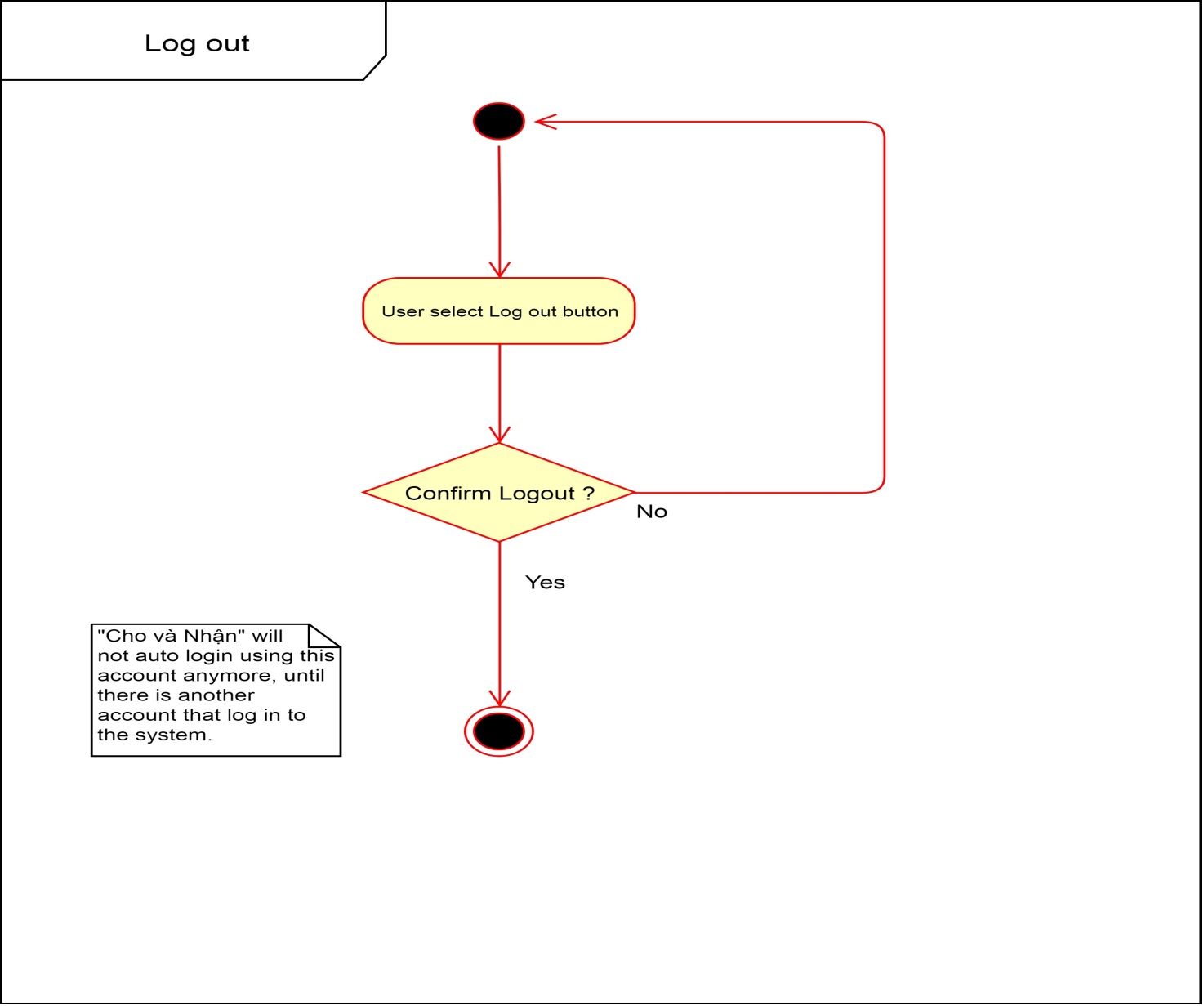
1. **ANALYZE AND DESIGN**
   1. **Activity Diagram**
      1. **Login**



*Figure 8‑1 Activity Diagram\_ Login Using Facebook*

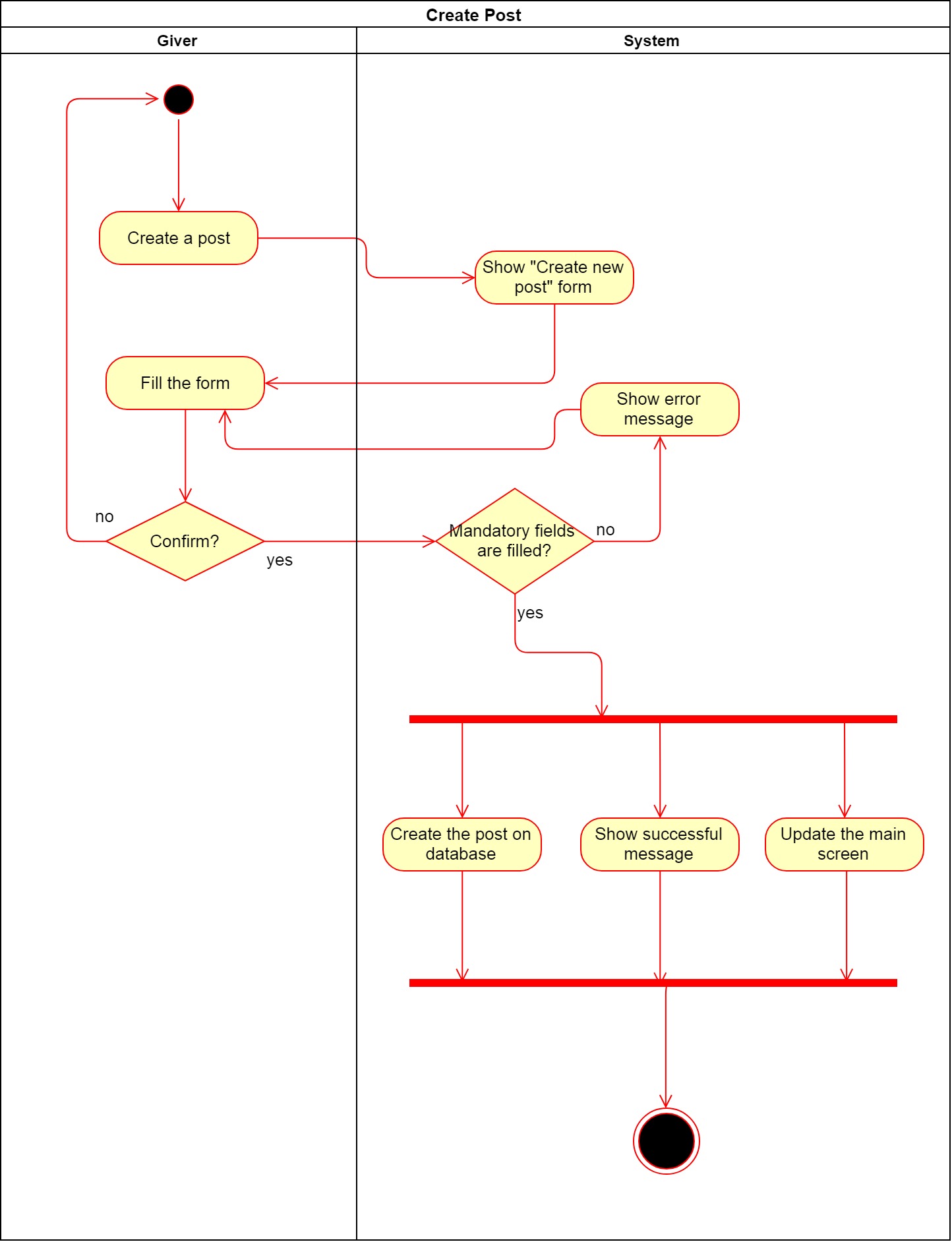


*Figure 8‑2 Activity Diagram\_ Login Using Google*

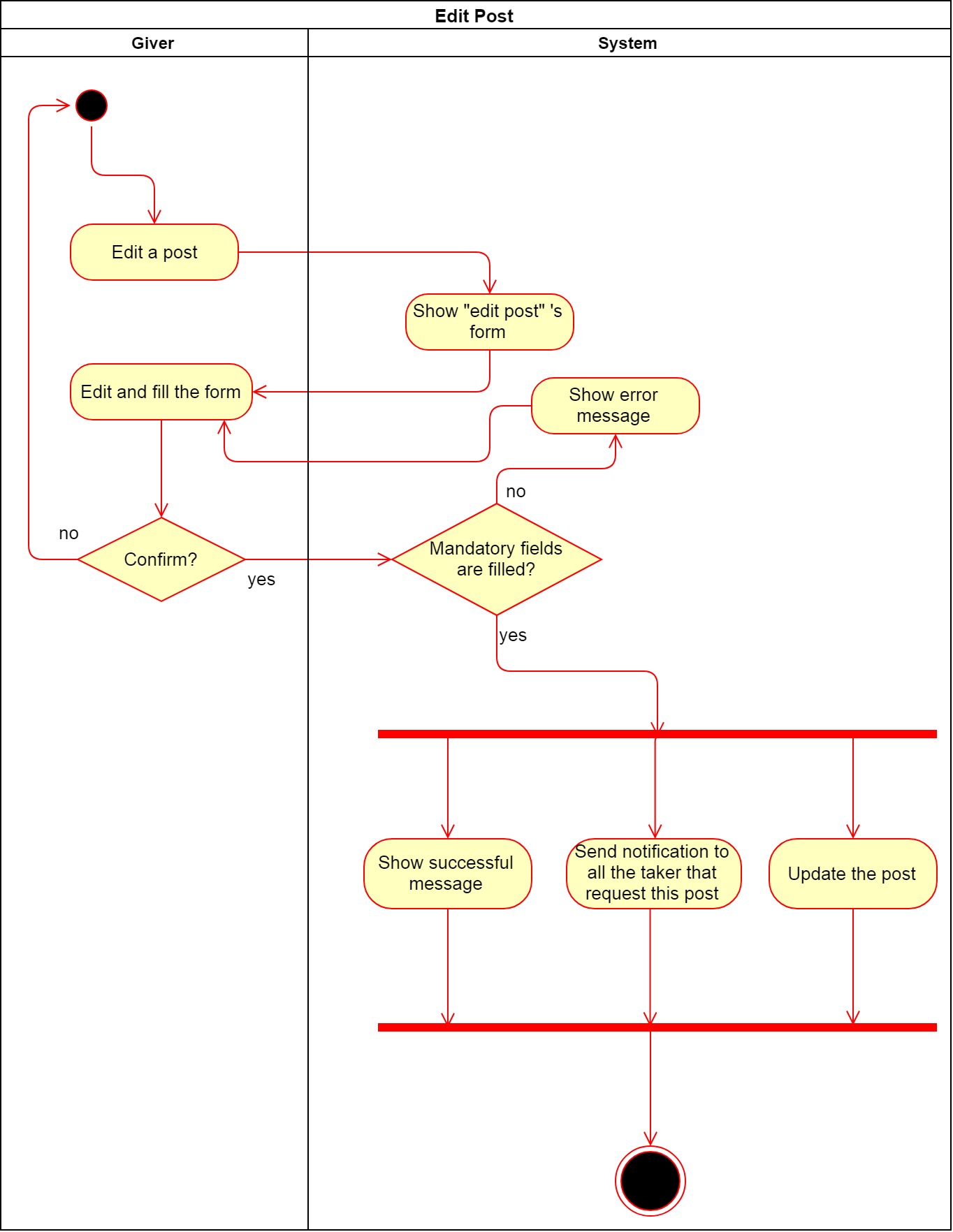


*Figure 8‑3 Activity Diagram\_ Logout*

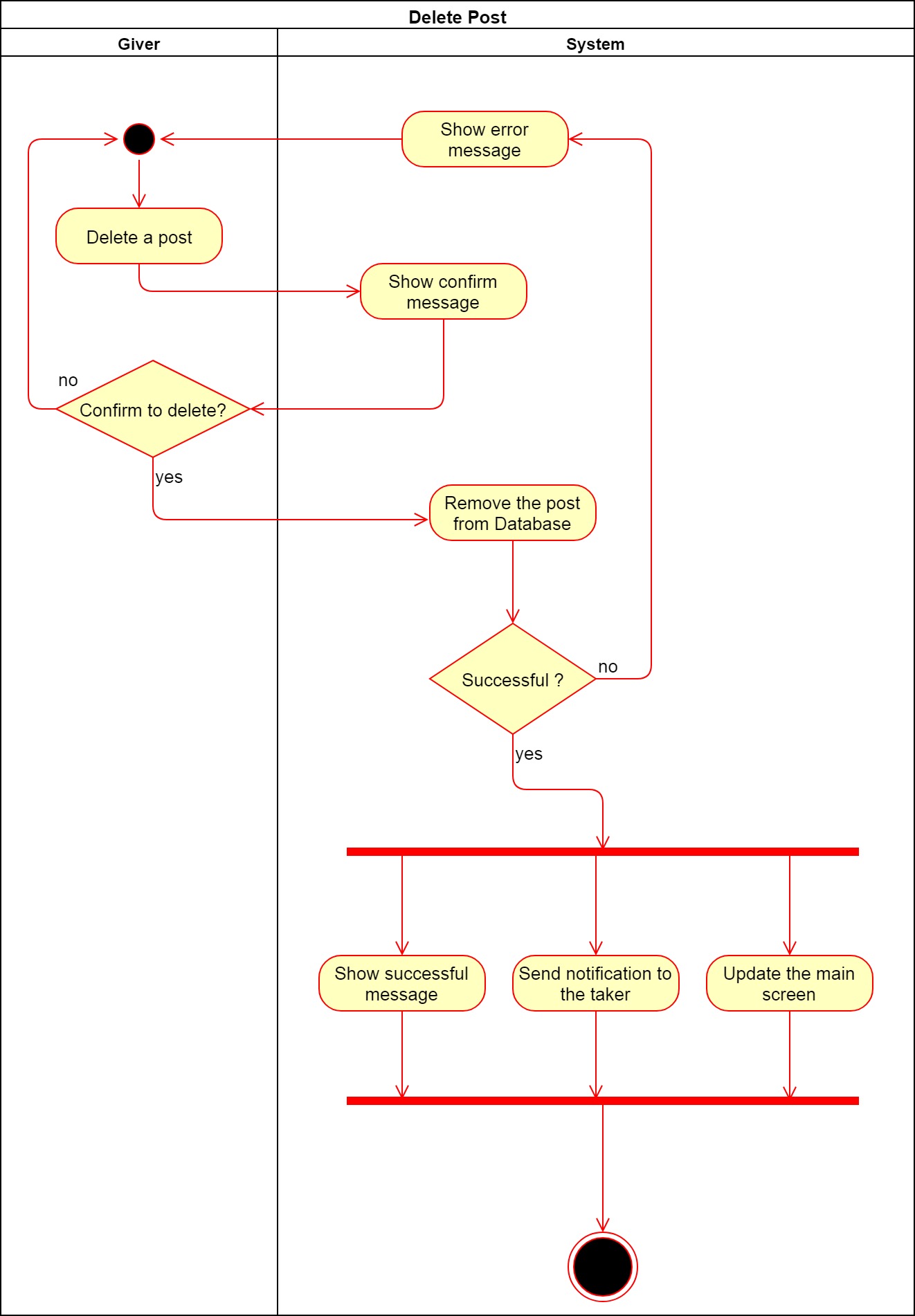
* + 1. **Post Management**



*Figure 8‑4 Activity Diagram\_ Create Post*

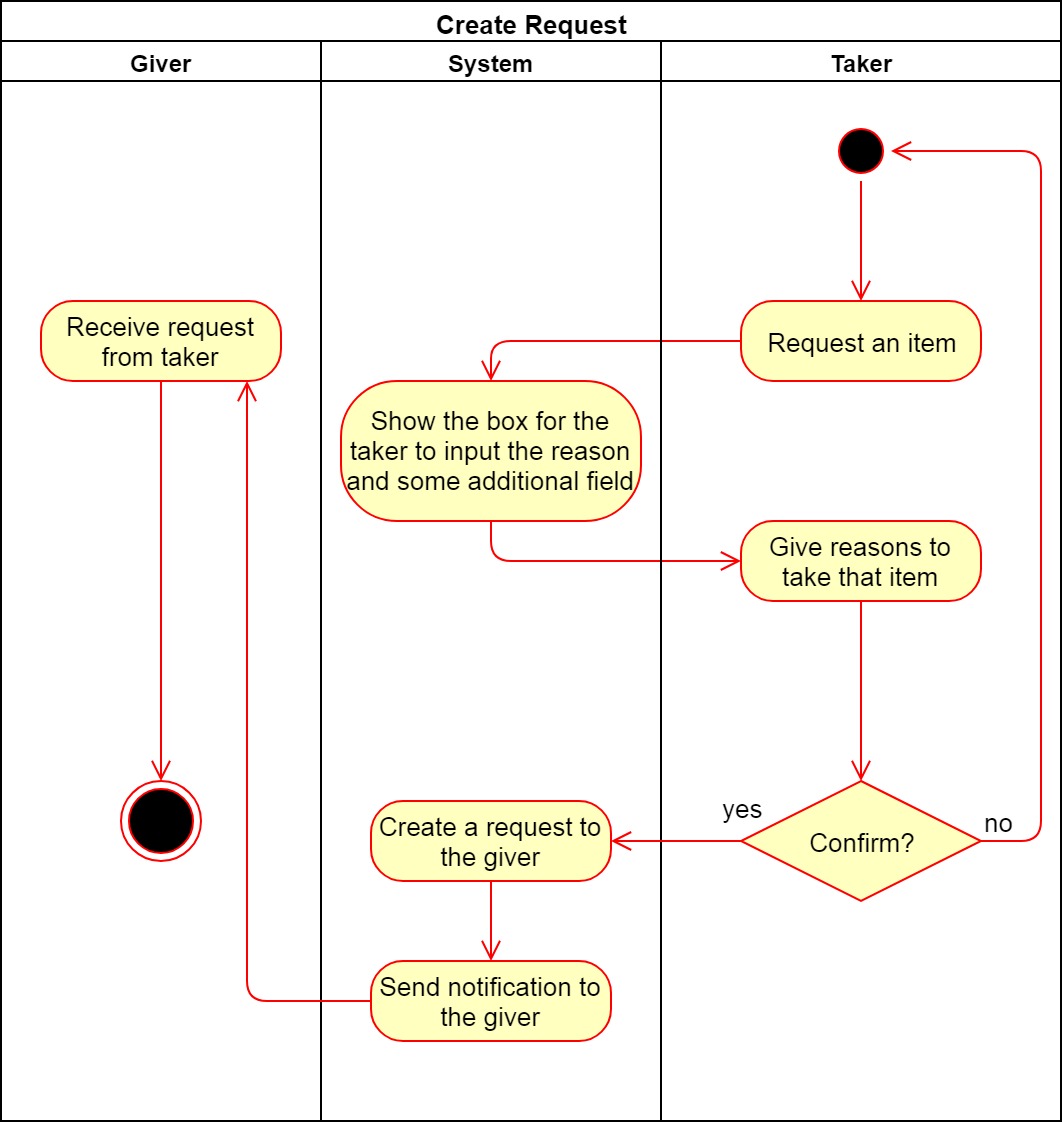


*Figure 8‑5 Activity Diagram\_ Edit Post*

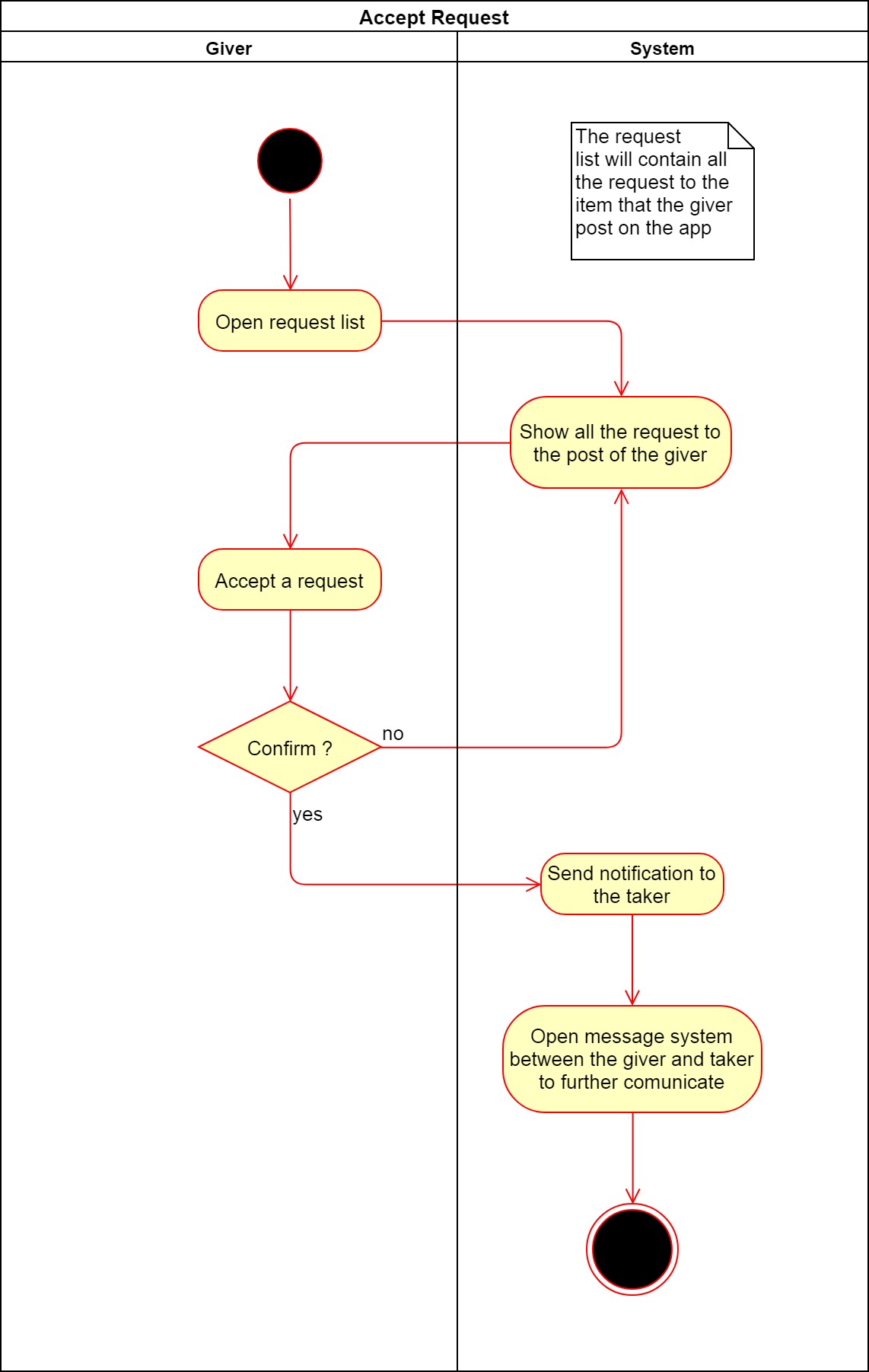


*Figure 8‑6 Activity Diagram\_ Delete Post*

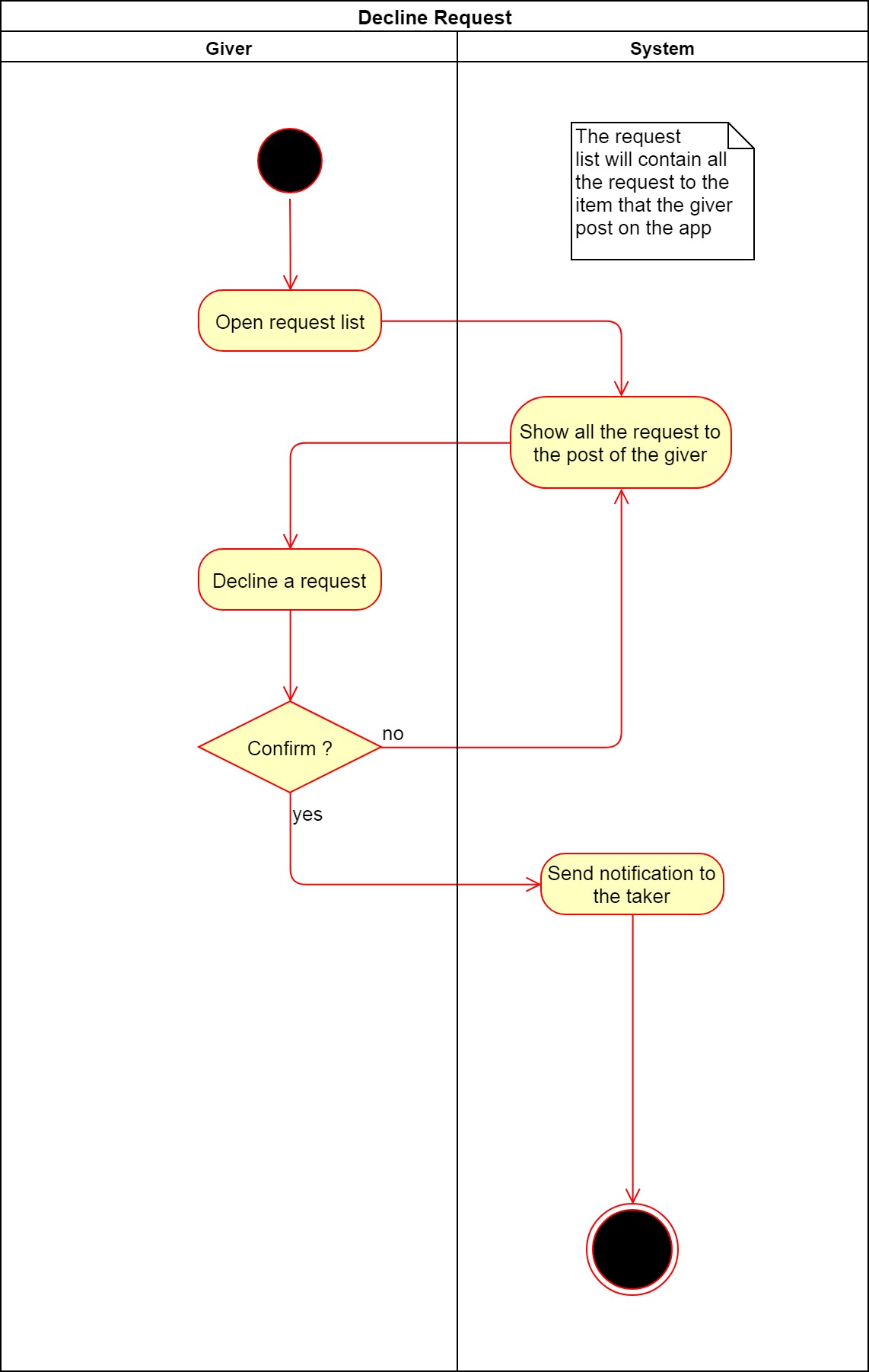
* + 1. **Request Management**



*Figure 8‑7 Activity Diagram\_ Create Request*

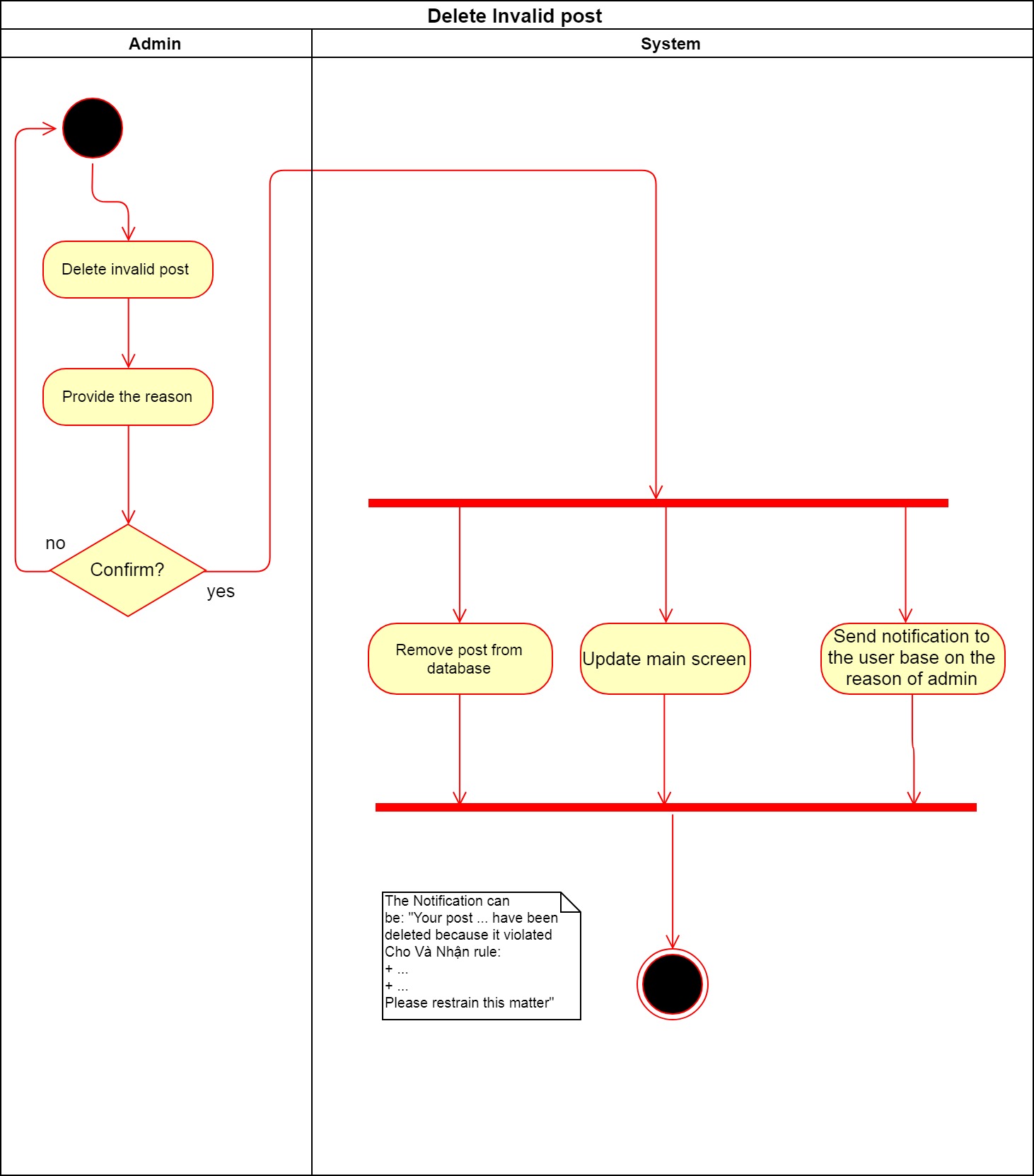


*Figure 8‑8 Activity Diagram\_ Accept Request*



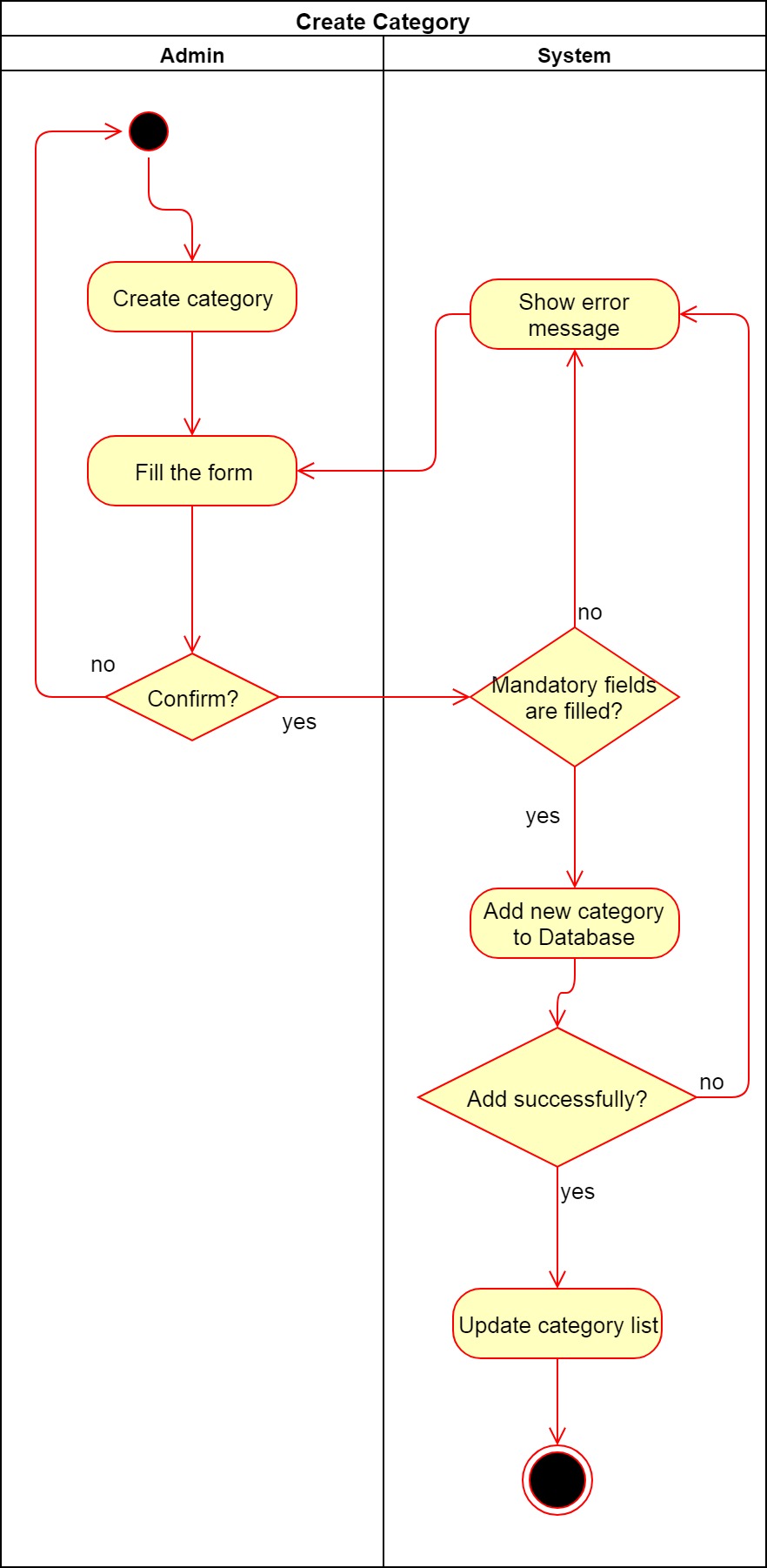
*Figure 8‑9 Activity Diagram\_ Decline Request*

* + 1. **Report Management**

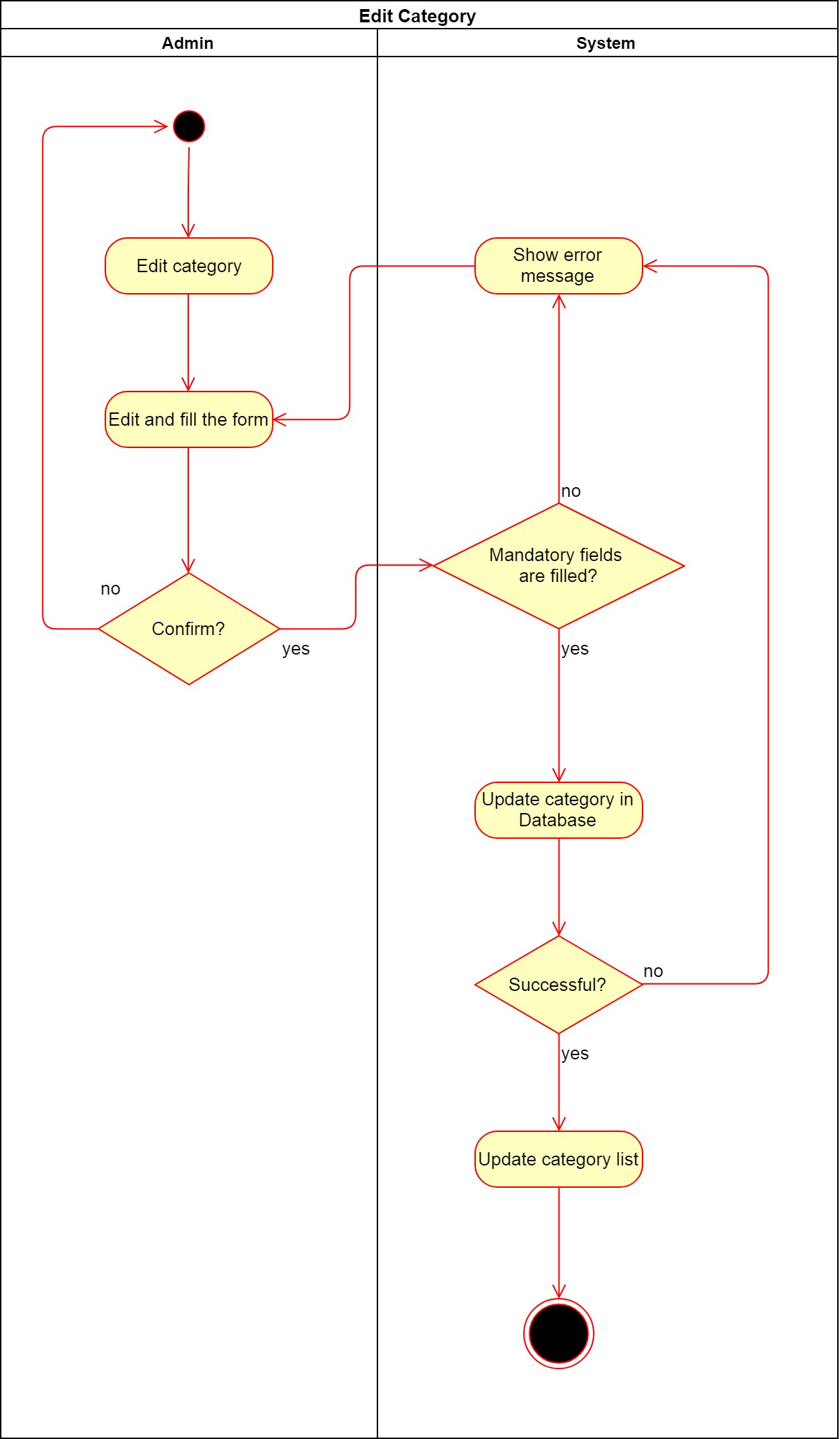


*Figure 8‑10 Activity Diagram\_ Delete Invalid Post*

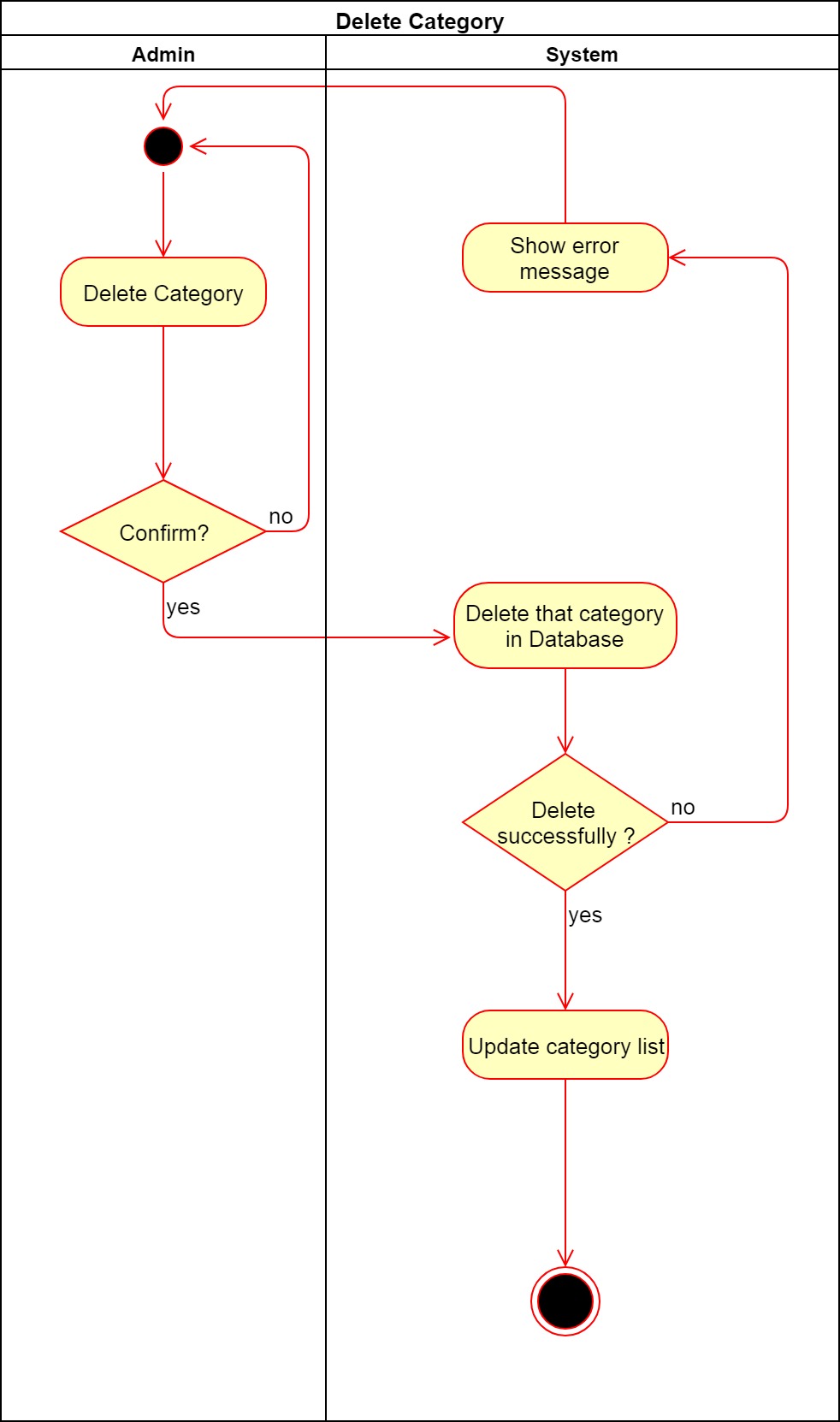
* + 1. **Category Management**



*Figure 8‑11 Activity Diagram\_ Create Category*

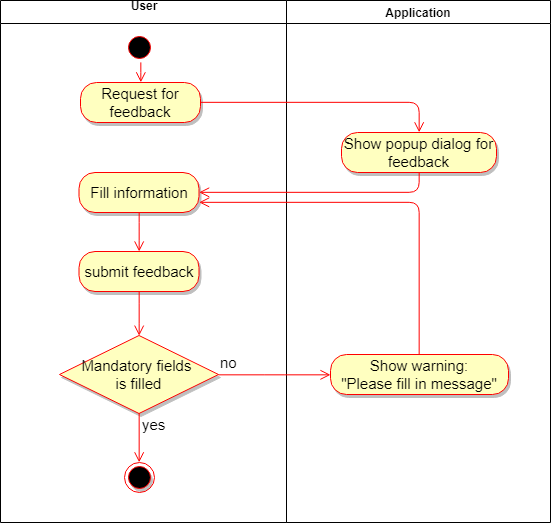


*Figure 8‑12 Activity Diagram\_ Edit Category*



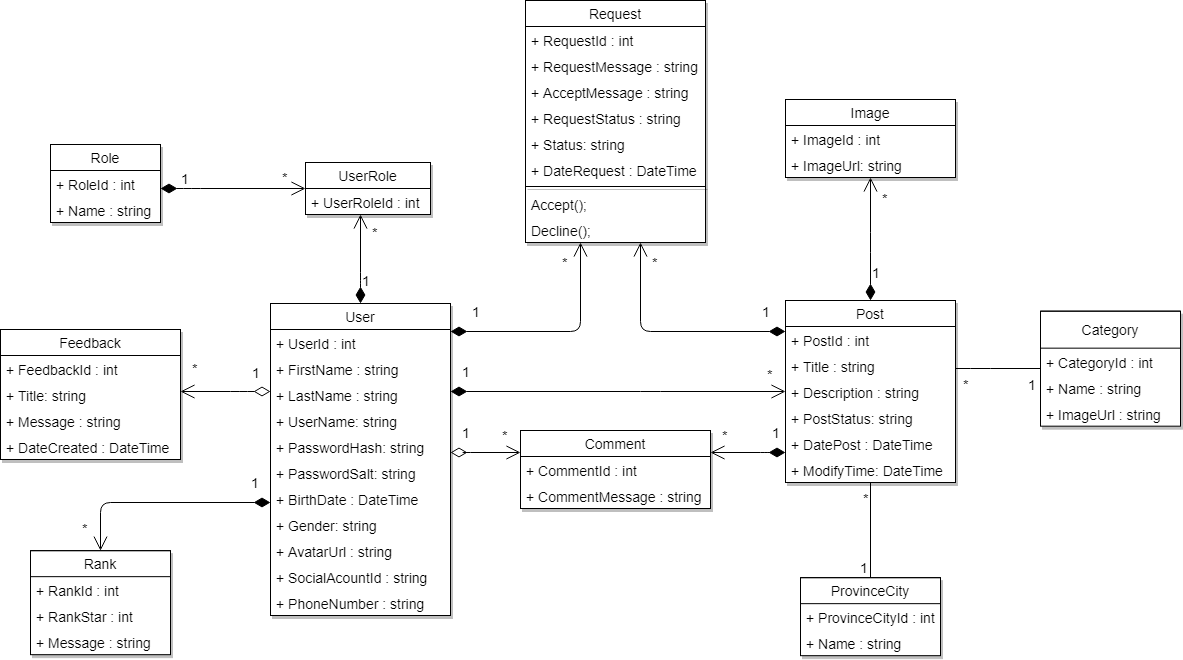
*Figure 8‑13 Activity Diagram\_ Delete Category*

* + 1. **Feedbacks Management**



*Figure 8‑14 Activity Diagram\_ Create Feedbacks*

* 1. **Class Diagram**



*Figure 8‑15 Class Diagram*

* 1. **Database**

…

1. **APPENDIX A: [TBD]**