

**Thuedi**

**Software Design Document**

– Hanoi, January 2022 –

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) **3**

[1. Code Packages/Namespaces](#_heading=h.30j0zll)

2. User Case 3

3[. Coding Conventions](#_heading=h.1fob9te) 3

3[.1. File Names](#_heading=h.undryh9jby6x) 3

3[.2. File Organization](#_heading=h.nmgm1maaemvj) 4

[Java Source Files](#_heading=h.b553ehf5e9t2) 4

3[.3. Indentation](#_heading=h.b5bwbc13soqp) 4

3[.4. Comments](#_heading=h.kj2sg1mtdrwk) 4

3[.5. Declarations](#_heading=h.wop2wnqtuufm) 4

3[.6. Statements](#_heading=h.wj2f82718car) 5

3[.7. White Space](#_heading=h.u71pshro1i28) 5

3[.8. Naming Conventions](#_heading=h.1kpt4649omed) 5

[II. Code Designs](#_heading=h.3znysh7) **5**

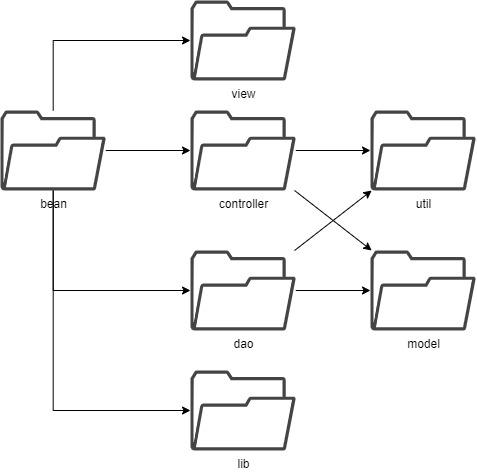
[III. Database Design](#_heading=h.17dp8vu) **7**

[1. Database Schema](#_heading=h.b4yd1h5bfruv) 7

[2. Table Description](#_heading=h.bn2tbo9ty3wl) 7

# I. Overview

## 1. Code Packages/Namespaces

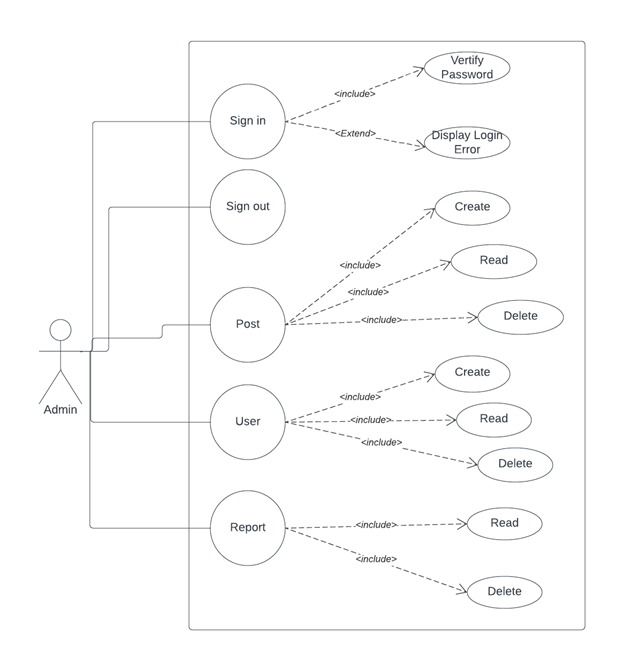


***Package descriptions & package class naming conventions***

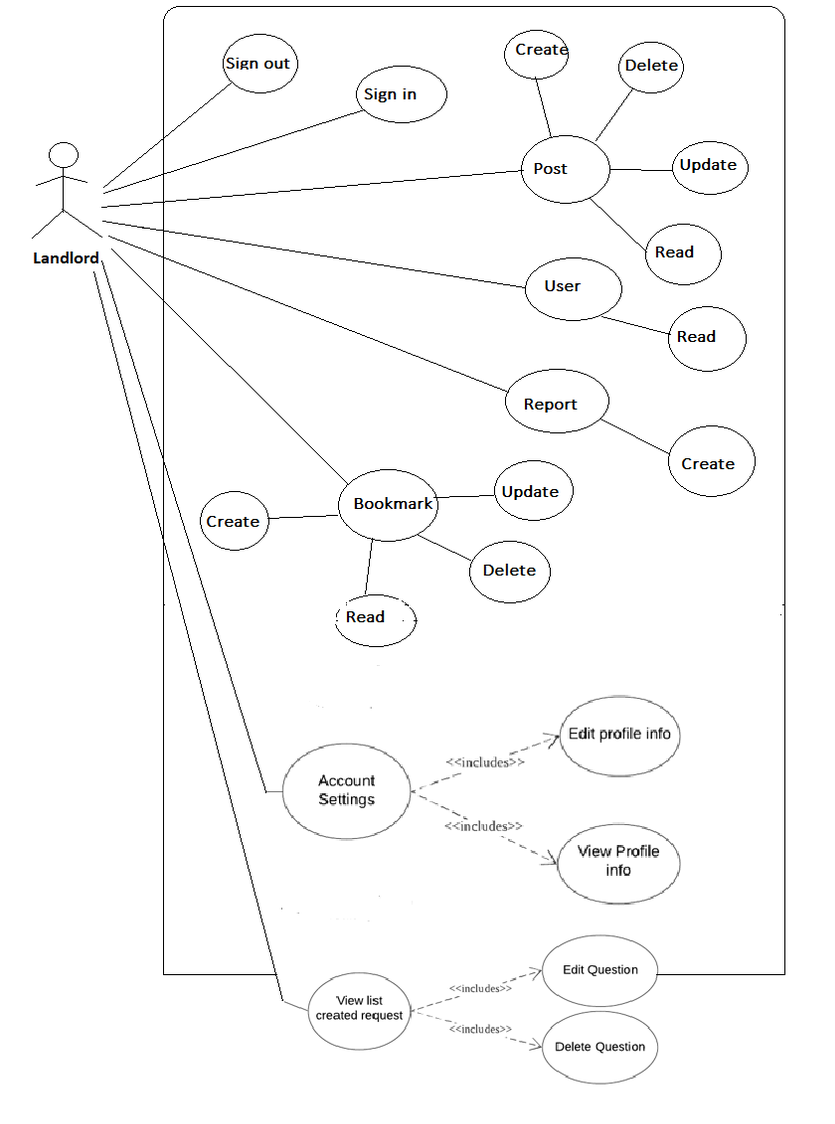
| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *bean* | *The highest package hierarchically that will contain the whole project.* |
| *02* | *view* | *Containing views for the MVC model.* |
| *03* | *controller* | *Containing controllers for the MVC model.* |
| *04* | *model* | *Containing models for the MVC model.* |
| *05* | *dao* | *Contains database related classes.* |
| *06* | *util* | *Contains utility classes/functions for other classes to implement/consumn.* |
| *07* | *lib* | *Contains third-party libraries.* |

## 2. User case

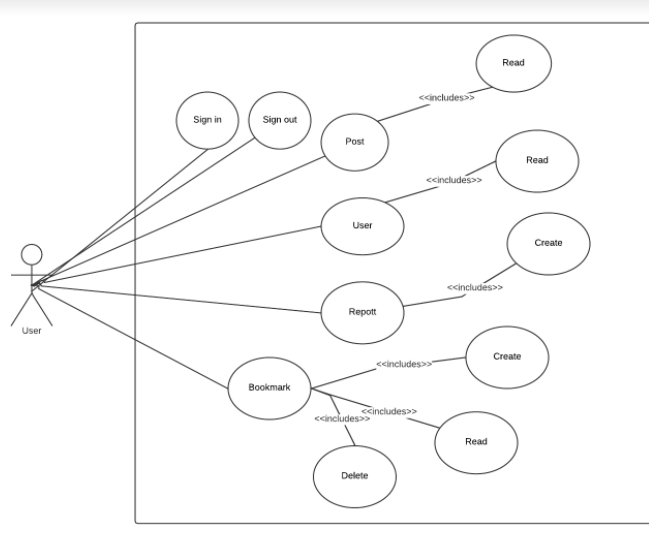
### 2.1. Admin



### 2.2. Landlord



### 2.3. User



## 3. Coding Conventions

### 3.1. File Names

| **File Type** | **Suffix** |
| --- | --- |
| Java source | .java |
| Java bytecode | .class |

### 3.2. File Organization

* A file consists of sections that should be separated by blank lines and an optional comment identifying each section.
* Files longer than 2000 lines are cumbersome and should be avoided.

#### Java Source Files

* Each Java source file contains a single public class or interface. When private classes and interfaces are associated with a public class, you can put them in the same source file as the public class. The public class should be the first class or interface in the file.
* Java source files have the following ordering:
* Beginning comments.
* Package and Import statements.
* Class and interface declarations.

### 3.3. Indentation

* Four spaces should be used as the unit of indentation.
* Line Length: Avoid lines longer than 80 characters, since they’re not handled well by many terminals and tools.
* Wrapping Lines: When an expression will not fit on a single line, break it according to these general principles:
* Break after a comma.
* Break before an operator.
* Prefer higher-level breaks to lower-level breaks.
* Align the new line with the beginning of the expression at the same level on the previous line.
* If the above rules lead to confusing code or to code that’s squished up against the right margin, just indent 8 spaces instead.

### 3.4. Comments

* Implementation comment formats: Programs have four styles of implementation comments: block, single-line, trailing and end-of-line.
* Documentation comments: Doc comments describe Java classes, interfaces, constructors, methods, and fields. Each doc comment is set inside the comment delimiters /\*\*...\*/, with one comment per API. Doc comments should not be positioned inside a method or constructor definition block, because Java associates documentation comments with the first declaration after the comment.

### 3.5. Declarations

* Number per line: One declaration per line is recommended since it encourages commenting.
* Placement: Put declarations only at the beginning of blocks.
* Initialization: Try to initialize local variables where they’re declared. The only reason not to initialize a variable where it’s declared is if the initial value depends on some computation occurring first.
* Class and interface declarations: When coding Java classes and interfaces, the following formatting rules should be followed:
* No space between a method name and the parenthesis “(“ starting its parameter list.
* No space between a method name and the parenthesis “(“ starting its parameter list.
* Closing brace “}” starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the “}” should appear immediately after the “{“ .
* Methods are separated by a blank line.

### 3.6. Statements

* Simple statements: Each line should contain at most one statement.
* Compound statements: Compound statements are statements that contain lists of statements enclosed in braces “{ statements }”.

### 3.7. White Space

* Blank lines: improve readability by setting off sections of code that are logically related.
* Blank space.

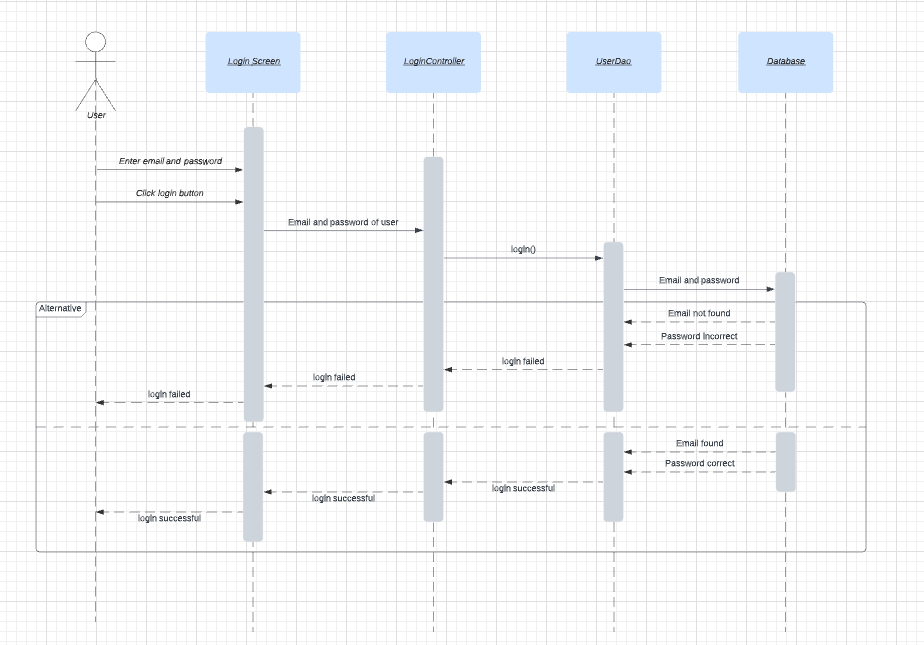
### 3.8. Naming Conventions

* Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier—for example, whether it’s a constant, package, or class—which can be helpful in understanding the code.

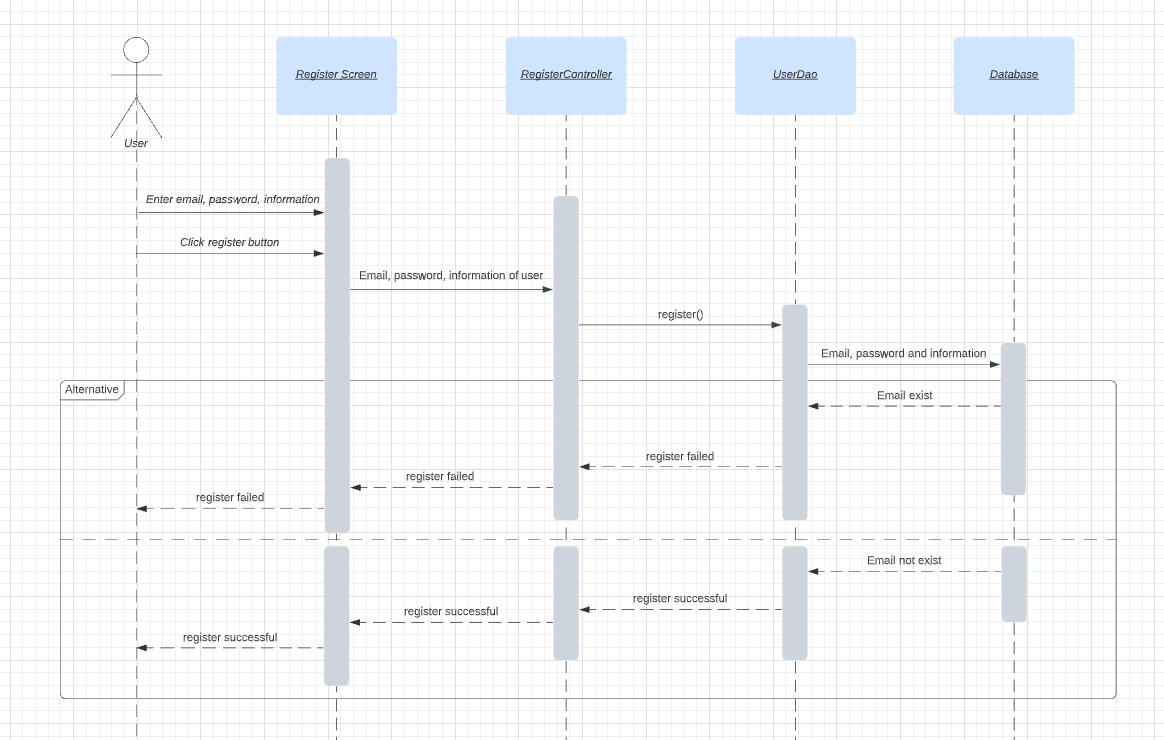
# II. Code Designs

## 1. Login

* Sequence diagram:

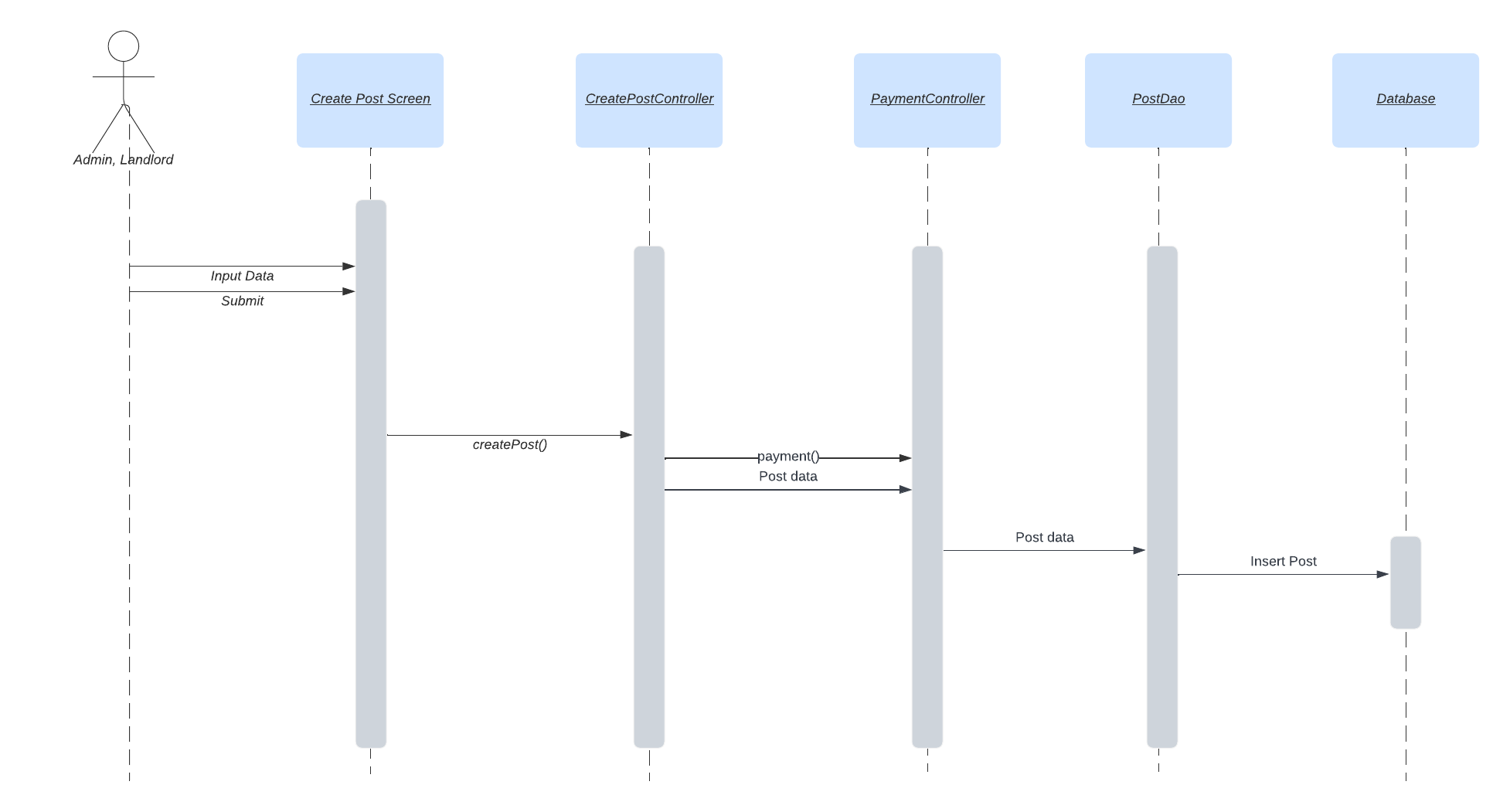


## 2. Register

* Sequence diagram: 

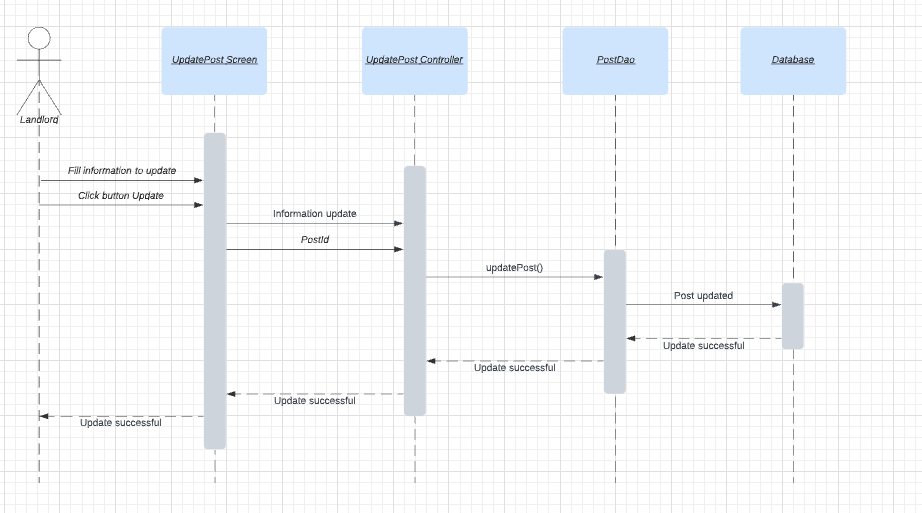
## 3. Create Post & 4. Pay to create post

* Sequence diagram:



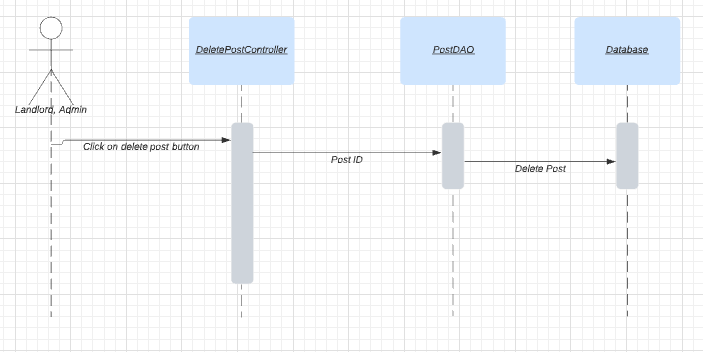
## 5. Update post

* Sequence diagram:



## 6. Delete post

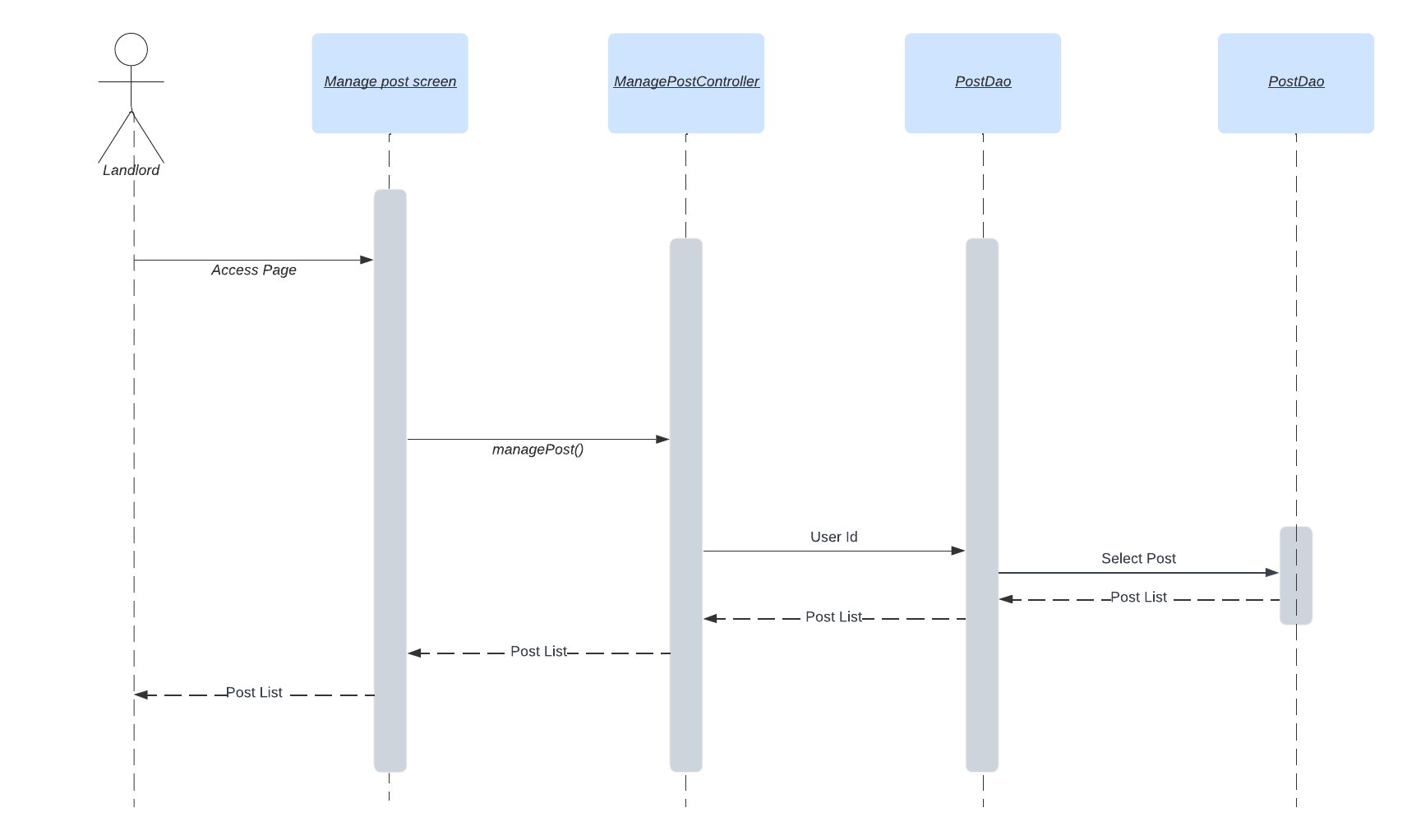
* Sequence diagram:



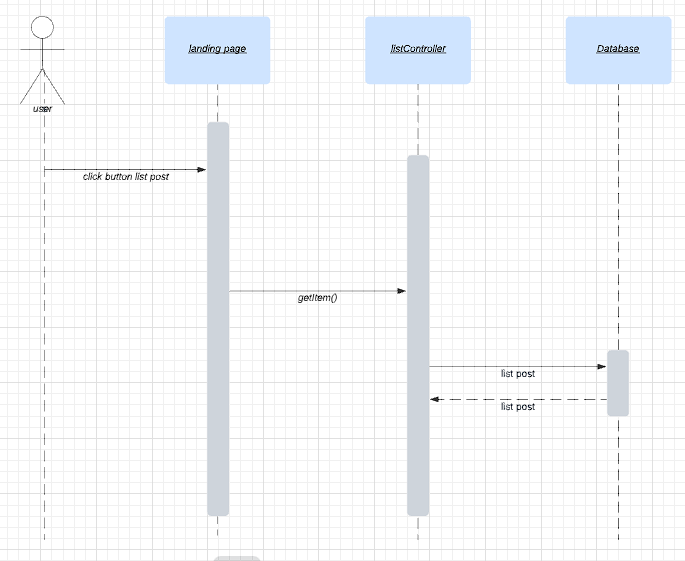
## 

## 7. Manage post

* Sequence diagram:

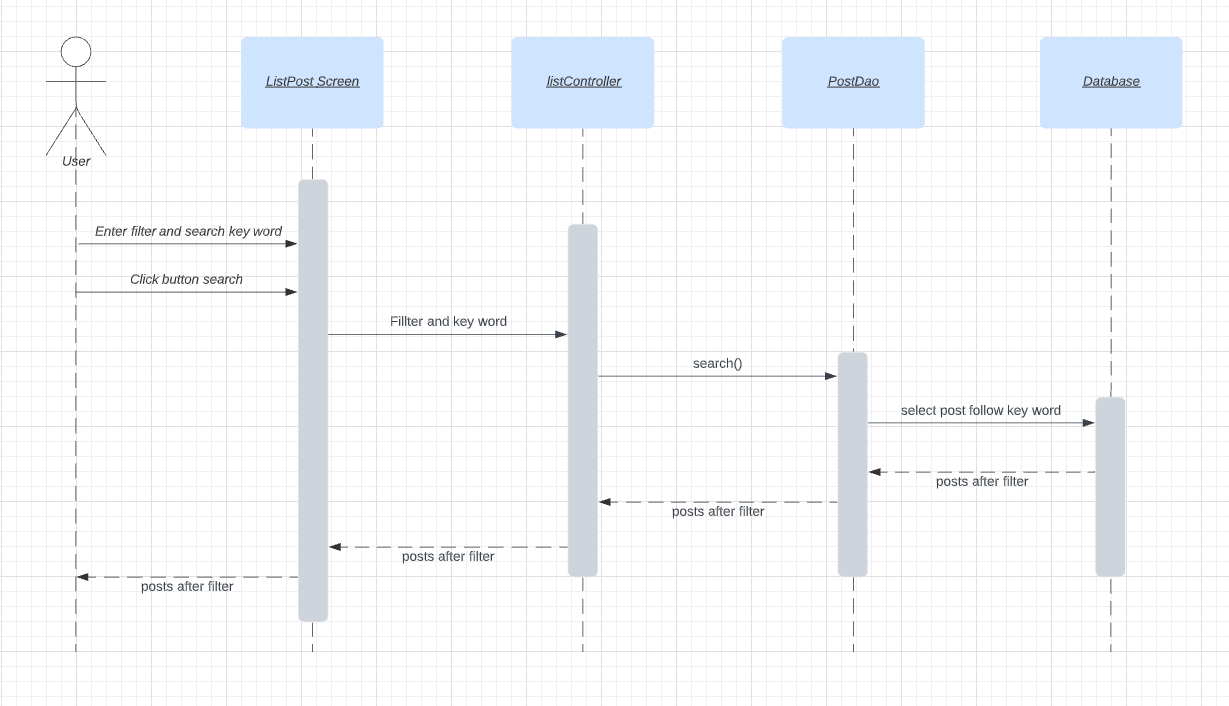


## 8. Post list & 9. Paginate posts

* Sequence diagram: 

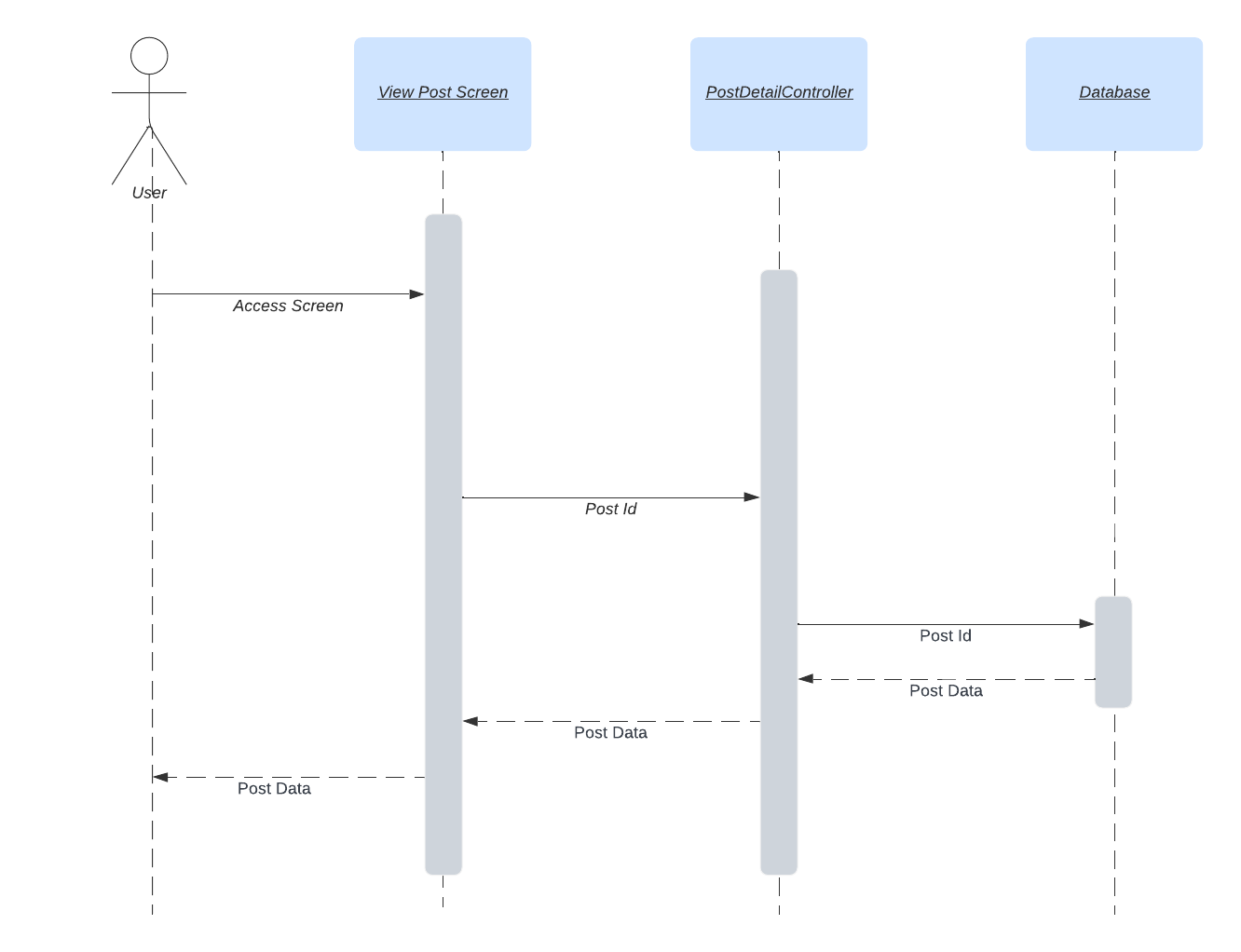
## 10. Filter post

* Sequence diagram:



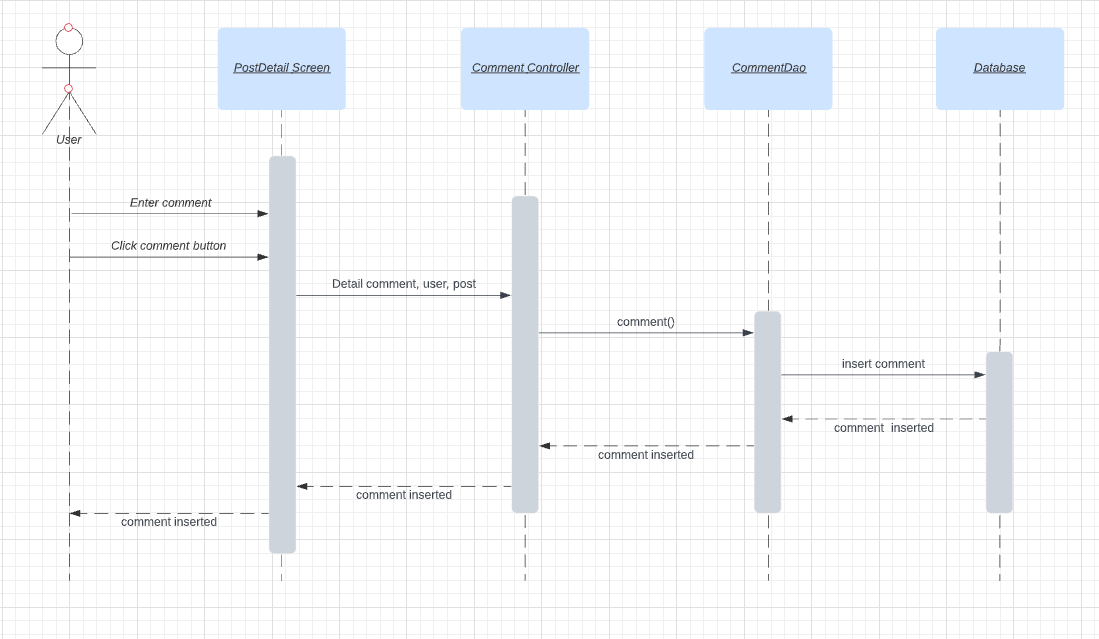
## 11. View post

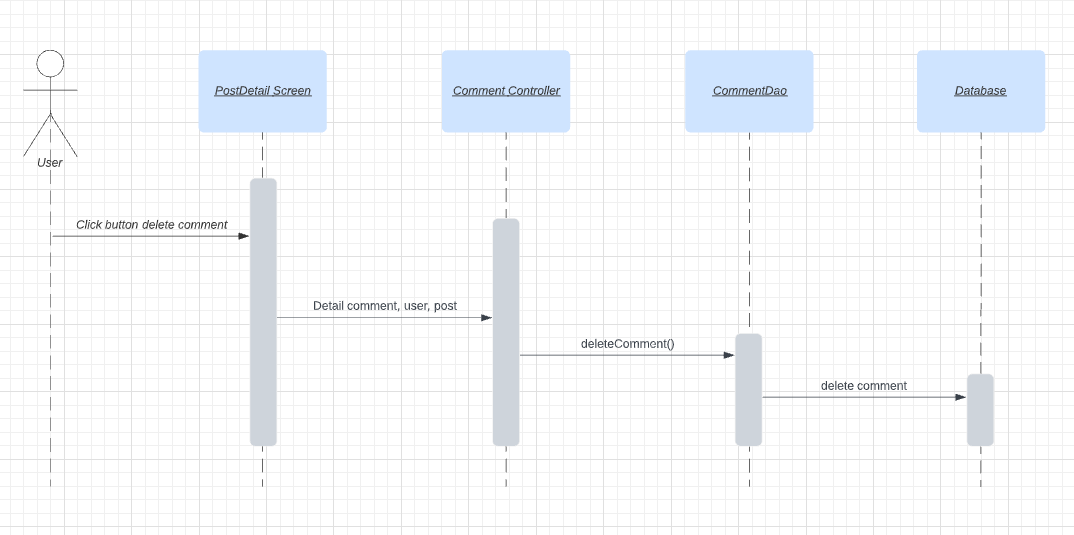
* Sequence diagram:



## 12. Comment on post

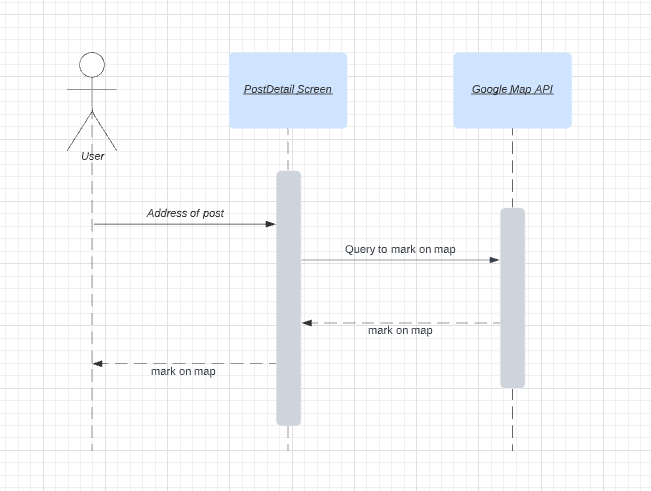
* Sequence diagram:



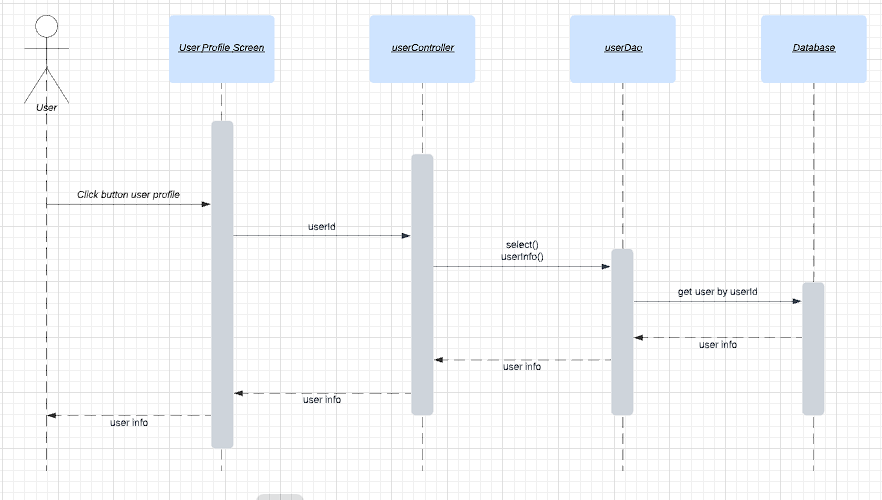


## 13. View services near address

* Sequence diagram:

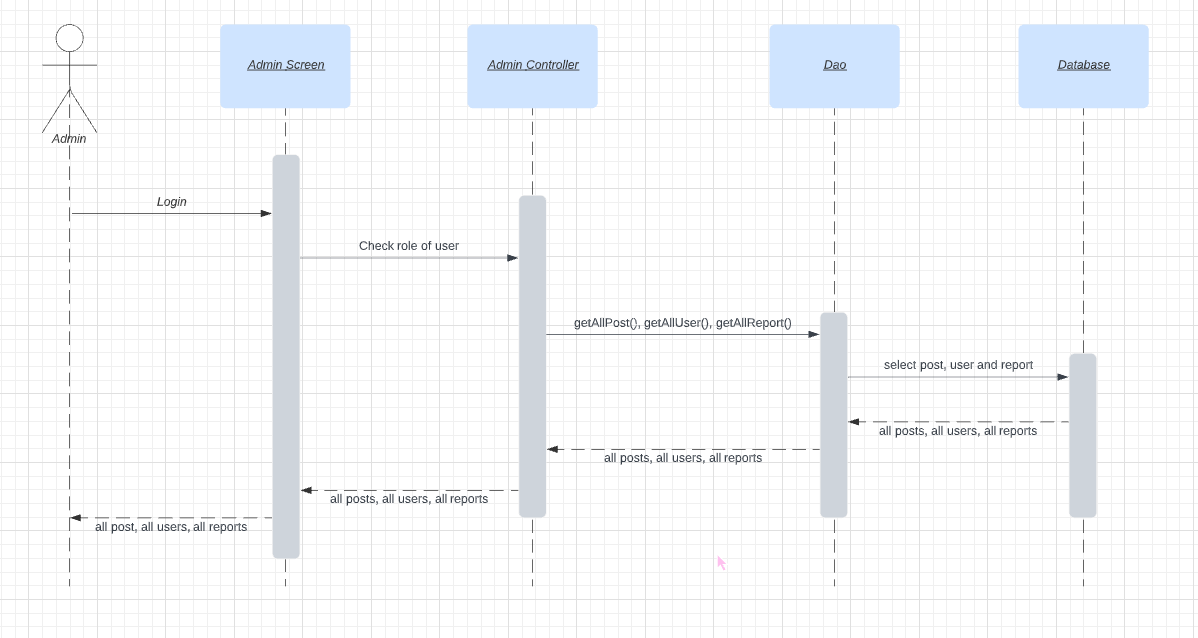


## 14. User profile

* Sequence diagram: 

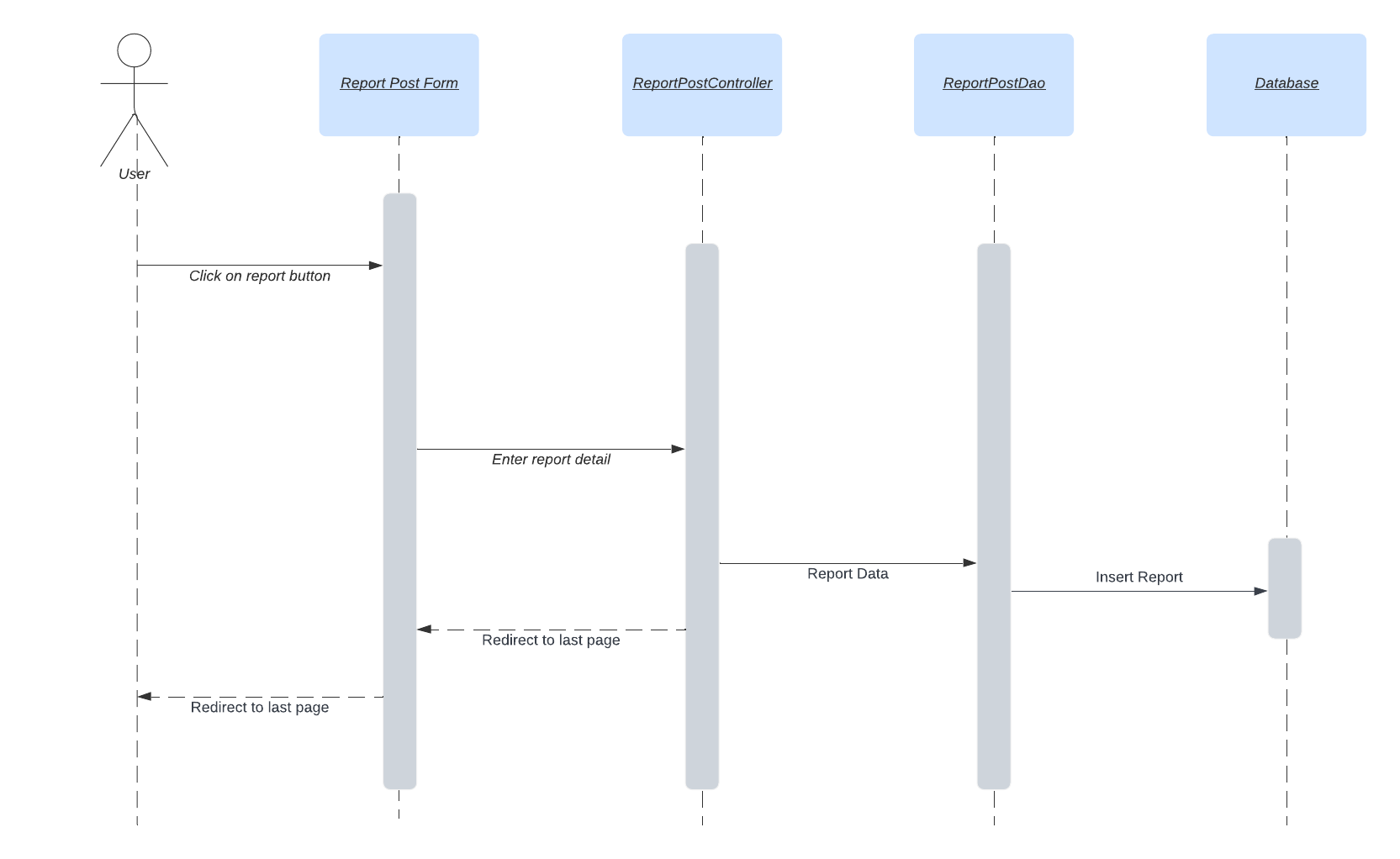
## 15. Admin page

* Sequence diagram:



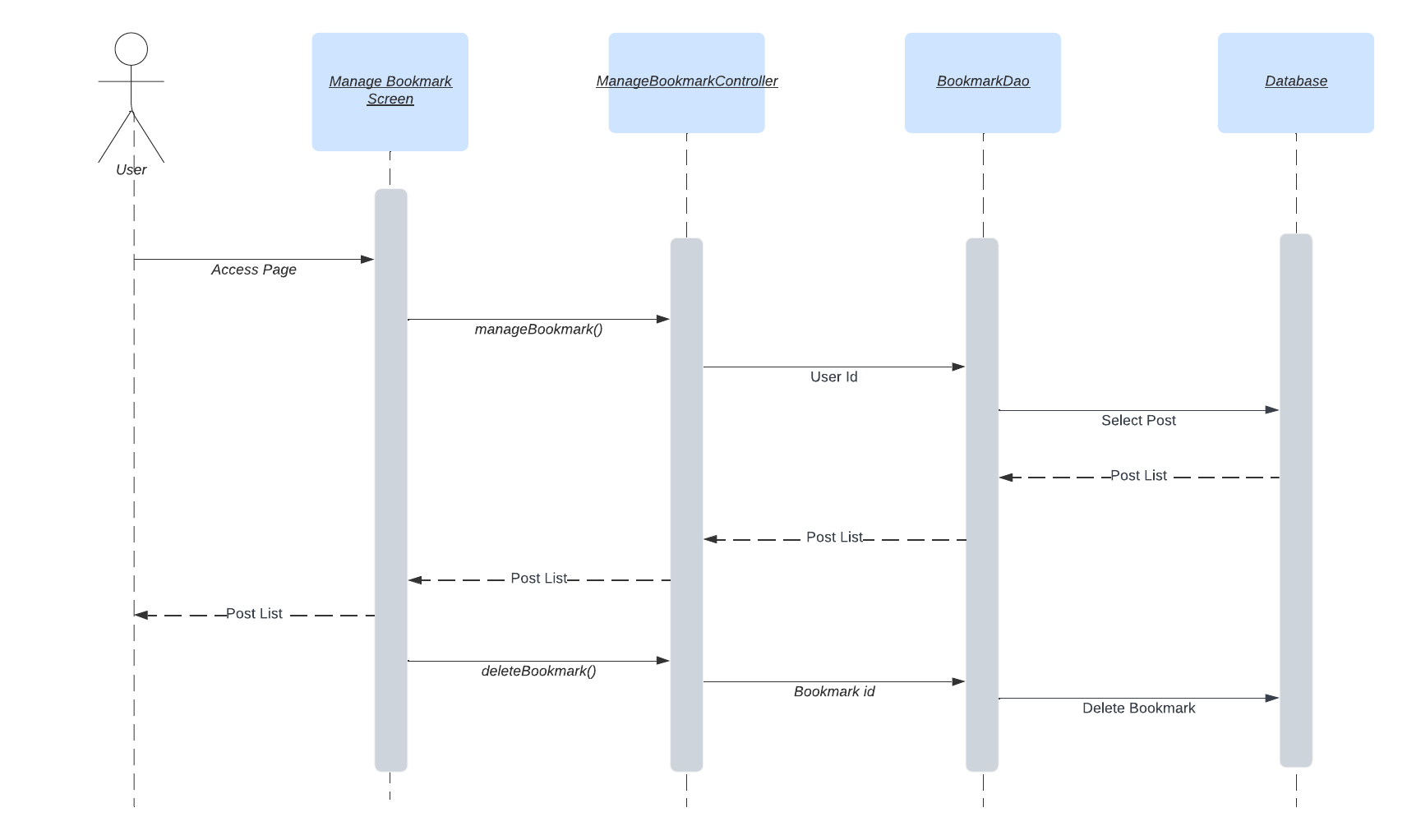
## 16. Report post

* Sequence diagram:



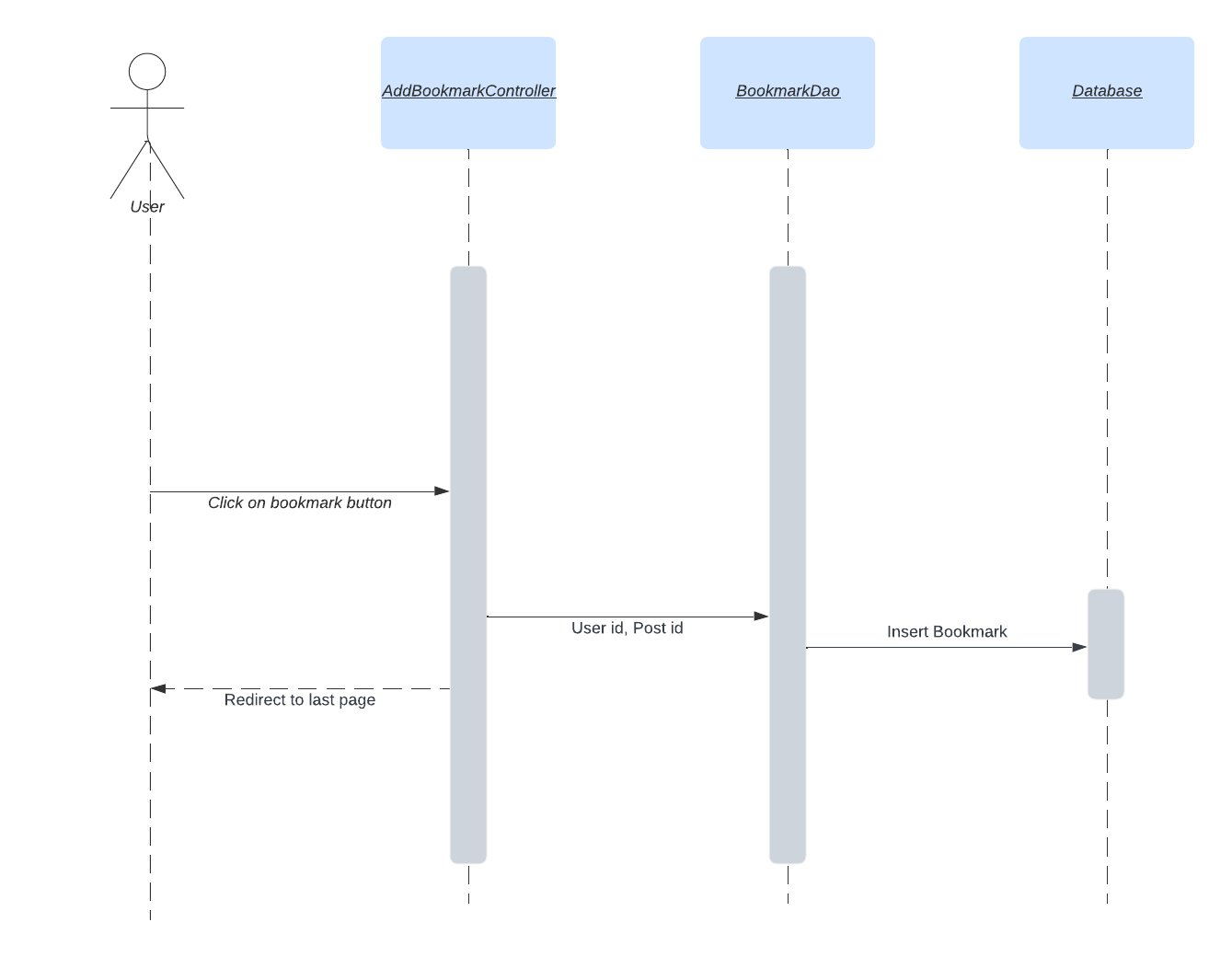
## 17. Review area

* Sequence diagram:

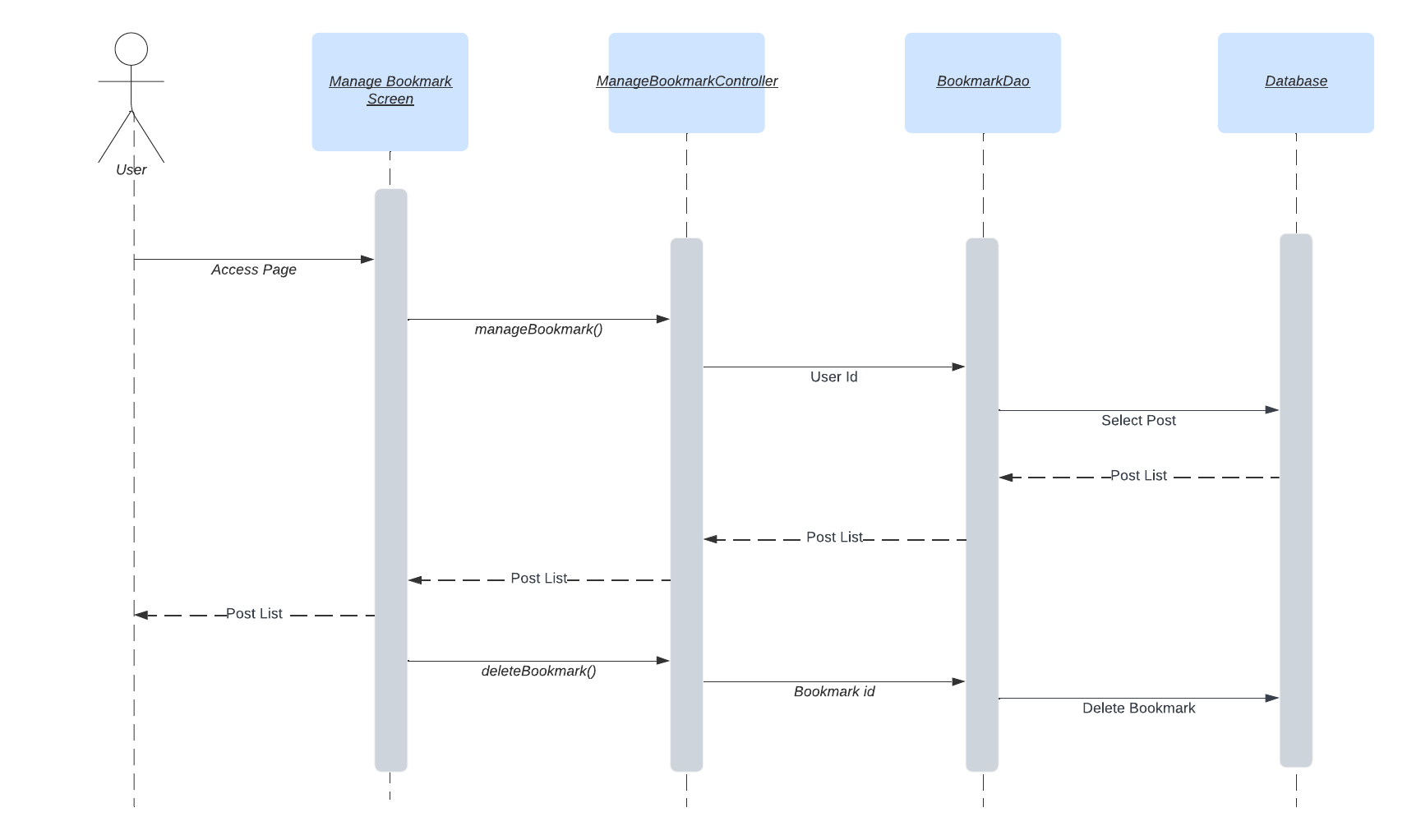


## 18. Bookmark post

* Sequence diagram:

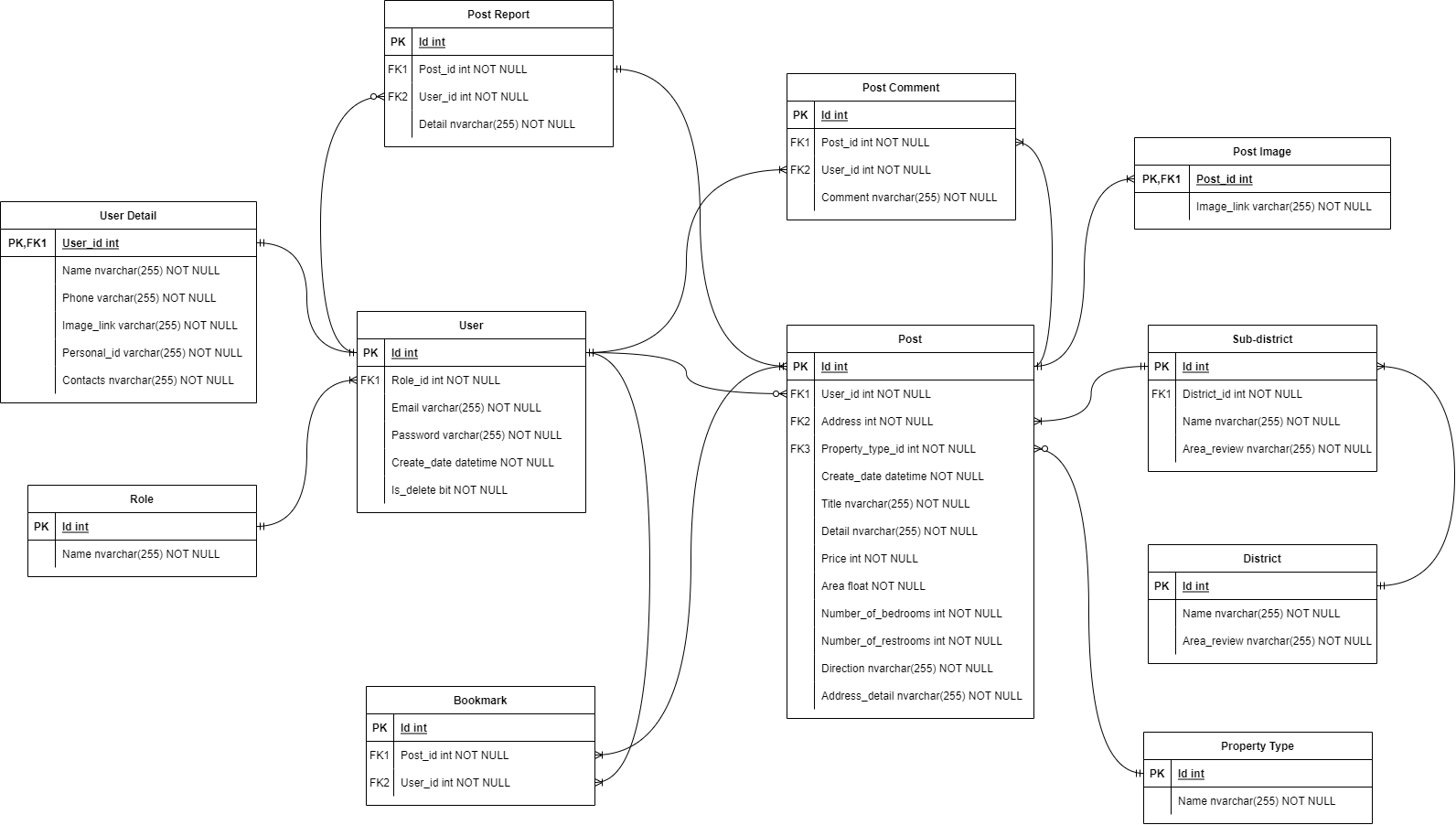


## 19. Manage bookmarks

* Sequence diagram:
* 

# III. Database Design

## 1. Database Schema



## 2. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *User* | *Include the user's information to log into the system: username, password*  *- Primary keys: id*  *- Foreign keys: role\_id* |
| *02* | *User detail* | *Include the user's information such as name, phone,...*  *- Primary keys: user\_id*  *- Foreign keys: user\_id* |
| *03* | *Role* | *Categorize user roles to be able to delegate different functions*  *- Primary keys: id* |
| *04* | *Post* | *Include the post's information such as title, detail,...*  *- Primary keys: id*  *- Foreign keys: user\_id, address, property\_type\_id* |
| *05* | *Property type* | *Categorize post for users to easily filter and search information*  *- Primary keys: id* |
| *06* | *Post image* | *Include the post's images so that users can easily visualize the room*  *- Primary keys: post\_id*  *- Foreign keys: post\_id* |
| *07* | *Post comment* | *Include the comment of user in a post*  *- Primary keys: id*  *- Foreign keys: post\_id, user\_id* |
| *08* | *Post report* | *Include the information of the post when the user reports a violation*  *- Primary keys: id*  *- Foreign keys: post\_id, user\_id* |
| *09* | *Bookmark* | *Includes information about the posts the user has bookmarked*  *- Primary keys: id*  *- Foreign keys: post\_id, user\_id* |
| *10* | *District* | *Includes information of all districts in Hanoi*  *- Primary keys: id* |
| *11* | *Sub-district* | *Includes information of all sub-districts in Hanoi*  *- Primary keys: id*  *- Foreign keys: district\_id* |