

Contents:

1. Introduction…………………………………………………………………………………….

2. Functional Requirements……………………………………………………………………

3. Non- functional Requirements……………………………………………………………

4. Use-Case Diagram…………………………………………………………………………….

4.1 Use-Case…………………………………………………………………………………..

4.2 Use-Case Description…………………………………………………………………

5. System Architecture…………………………………………………………………………

5.1 System Architecture…………………………………………………………………

5.2 System Architecture Description……………………………………………..

6. Activity Diagrams……………………………………………………………………………..

7. Database Specification……………………………………………………………………

7.1 ERD………………………………………………………………………………………….

7.2 Tables………………………………………………………………………………………

8. Class Diagram………………………………………………………………………………….

9. Object Diagram……………………………………………………………………………….

10. Package Diagram…………………………………………………………………………..

11. Sequence Diagram…………………………………………………………………………

3.0 Non-Functional Requirements

3.1 Look and Feel Requirements

-the page have listening mood to help all people

-the page absorb facts fast and experience stories in new way with Innovative graphics and data visualization

3.2 Performance Requirements

-the page bring you moment-by-moment updates on develop

stories from experts in the field

-the page works immediately when the user open it

-refresh process takes less than max 5 sec

3.3 Usability & Humanity Requirements

-the system is easy to use and understand without training

-the system is easy with payments

-the page can translate articles into 16 languages to be easy to understand

3.4 Operational & Environmental Requirements

-the page uses Dark Mode so it can be used in variable lighting conditions

3.5 Cultural Requirements

-the page language is formal and polite

-the system is sensitive to the cultural background

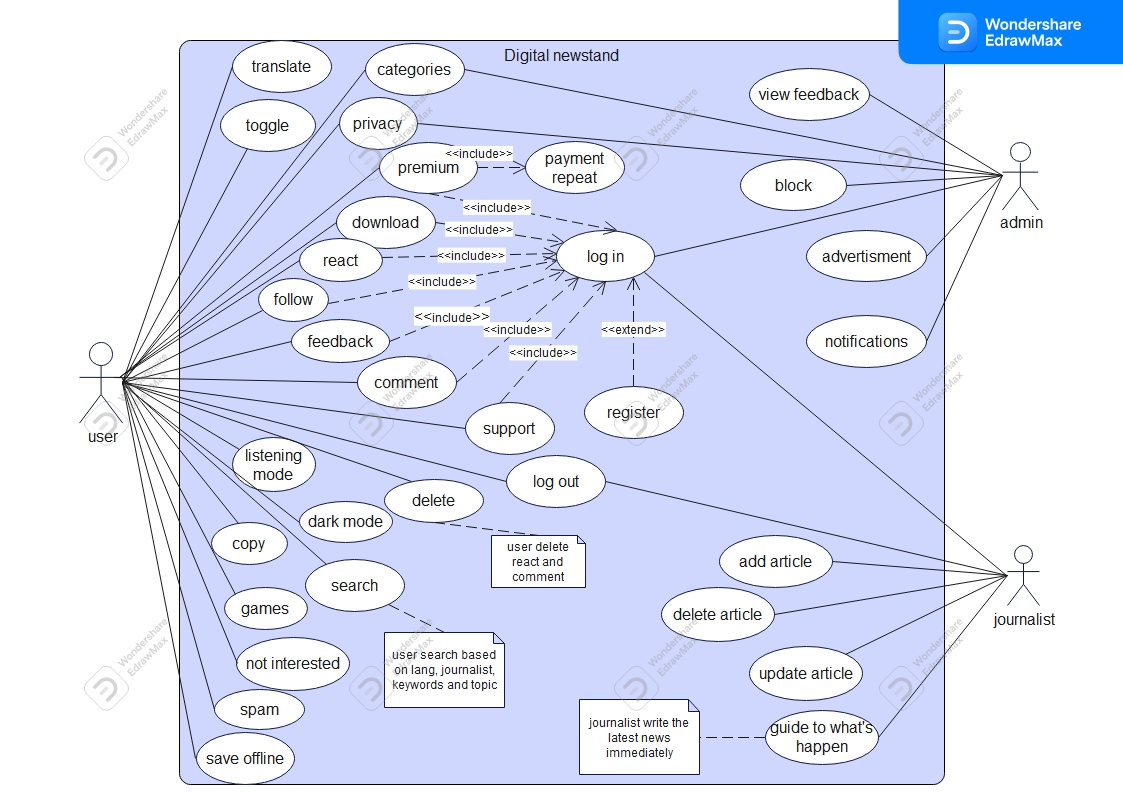
3.6 Security Requirements

-the system saves the information of every user

-the system protects your payment information from theft

4. Use-Case Diagram

4.1 Use-Case



4.2 Use-Case Description

* + 1. log in

|  |  |
| --- | --- |
| ID Number-Name: | -uc-1 |
| Goal: | -User can log in to the system to browsing the website |
| Pre-conditions: | -The user should have an available username and password  -the admin and journalist should have an ID and a password |
| Post-conditions: | -the user, admin and journalist accessed in the system |
| Error Situations: | -the username and password are not allowed |
| Actors: | -User, admin and journalist |
| Trigger: | -once user clicks on the log in button |
| Main Success Scenario: | -steps:  1-enter username and password for user and ID for admin and journalist  2-validate(username, ID ,password)  3-enter for his account |
| Alternative: | -steps:  2’-invalid(username, ID ,password)  3’-there is an error |

* + 1. Toggle

|  |  |
| --- | --- |
| ID Number-Name: | -uc-2 |
| Goal: | -user can see the original print view and the new one |
| Pre-conditions: | -there is an original print view and web-optimized text view |
| Post-conditions: | -user can read the two articles |
| Error Situations: | -the original article not found or the copy article |
| Actors: | -user |
| Trigger: | -when user click on button to see the articles |
| Main Success Scenario: | -steps:  1-user search for the article  2-it appears two articles (one original and another copy)  3-user choose which one he wants to read |
| Alternative: | -steps:  2’-there isn’t an original article |

* + 1. Subscribe to Premium

|  |  |
| --- | --- |
| ID Number-Name: | -uc-3 |
| Goal: | -user can subscribe to premium to get unlimited access to every newspaper and magazine |
| Pre-conditions: | -user should have an account on the website |
| Post-conditions: | -user get unlimited access to every newspaper and magazine |
| Error Situations: | - User cannot subscribe to the website |
| Actors: | -user |
| Trigger: | -when user subscribe to premium he get unlimited access to the website |
| Main Success Scenario: | -steps:  1-user enter his account information  2-choose to subscribe premium  3-user should pay to access it  4-user have unlimited access |
| Alternative: | -steps:  1’- there is an error  3’-there is a problem with payment |

* + 1. Auto Renew to Payment of Premium

|  |  |
| --- | --- |
| ID Number-Name: | -uc-4 |
| Goal: | -the user manage the renew of the premium if it was turned off/on |
| Pre-conditions: | -user choose if he want to do auto-renew mode  -user should pay for renewing |
| Post-conditions: | -user can continue using premium if he renew it |
| Error Situations: | -user didn’t renew the subscription |
| Actors: | -user |
| Trigger: | -user do the auto-renew mode |
| Main Success Scenario: | -steps:  1-user can turn off/on the auto renew |
| Alternative: | -steps:  1’-there is an error and user can’t renew the premium |

* + 1. Support

|  |  |
| --- | --- |
| ID Number-Name: | -uc-5 |
| Goal: | -function supports user if there is a problem  -user can edit his profile |
| Pre-conditions: | -user should have an account |
| Post-conditions: | -the problem should be solved  -User data has been modified |
| Error Situations: | -none |
| Actors: | -user |
| Trigger: | -when user ask for support |
| Main Success Scenario: | -steps:  1-user should open settings  2- user should choose support  3- user should write his problem and system will solve it |
| Alternative: | -steps:  3’-system couldn’t solve the problem |

* + 1. Feedback

|  |  |
| --- | --- |
| ID Number-Name: | -uc-6 |
| Goal: | -giving feedback on the system |
| Pre-conditions: | -user has already account in the system |
| Post-conditions: | -user will be able to send his feedback |
| Error Situations: | -user doesn’t have an account |
| Actors: | -user |
| Trigger: | -user can give feedback on the system |
| Main Success Scenario: | -steps:  1-user will send his feedback about the whole system |
| Alternative: | -none |

* + 1. View Feedback

|  |  |
| --- | --- |
| ID Number-Name: | -uc-7 |
| Goal: | -view feedback that comes from the user for admin |
| Pre-conditions: | -the system has already feedback from users |
| Post-conditions: | -admin will can see user’s feedback with there username |
| Error Situations: | -none |
| Actors: | -admin |
| Trigger: | -reading feedbacks of users |
| Main Success Scenario: | -steps:  1-admin will see user’s feedback |
| Alternative: | -none |

* + 1. Advertisement

|  |  |
| --- | --- |
| ID Number-Name: | -uc-8 |
| Goal: | -showing some advertisements while browsing the website |
| Pre-conditions: | -user should open the website |
| Post-conditions: | -user can see the advertisements |
| Error Situations: | -none |
| Actors: | -admin |
| Trigger: | -when the user opens the website he can read these advertisements |
| Main Success Scenario: | -steps:  1-user should open the website |
| Alternative: | -none |

* + 1. Add Article

|  |  |
| --- | --- |
| ID Number-Name: | -uc-9 |
| Goal: | -journalist adds articles and news on the website |
| Pre-conditions: | -journalist should have an account on the system |
| Post-conditions: | - journalist can add articles |
| Error Situations: | -none |
| Actors: | - journalist |
| Trigger: | - journalist can write articles and news |
| Main Success Scenario: | -steps:  1- journalist should have an account  2- journalist should write and add articles |
| Alternative: | -steps:  1’- journalist doesn’t have an account |

* + 1. Delete Article

|  |  |
| --- | --- |
| ID Number-Name: | -uc-10 |
| Goal: | -journalist should be able to delete articles an news on the website |
| Pre-conditions: | - journalist should have an account on the system |
| Post-conditions: | - journalist can add articles |
| Error Situations: | -none |
| Actors: | - journalist |
| Trigger: | - journalist can delete articles and news that he wrote |
| Main Success Scenario: | - steps:  1- journalist should have an account  2- journalist should delete articles |
| Alternative: | - steps:  1’- journalist doesn’t have an account |

* + 1. Download

|  |  |
| --- | --- |
| ID Number-Name: | -uc-11 |
| Goal: | -User can download any article to the device |
| Pre-conditions: | -User should have an account at the system |
| Post-conditions: | -The article will be loaded in the data base |
| Error Situations: | -User has no account in the system. |
| Actors: | -User. |
| Trigger: | -User clicked on download button. |
| Main Success Scenario: | -Steps:  1-User login the system  2- click to download |
| Alternative: | -Steps:   1. User open his account 2. Asked the user to login the system by his account |

* + 1. log out

|  |  |
| --- | --- |
| User case ID: | -uc-12 |
| Goal: | -User leaves the system. |
| Pre-conditions: | -The user shall be able to log in to the system. |
| Post-conditions: | -The user will be able to leave the system. |
| Error Situations: | -No error situations. |
| Actors: | -User |
| Trigger: | -User request to leave the website. |
| Main Success Scenario: | -Steps:   1. User presses the log out button. 2. User leave the system. 3. The system’s main page will be loaded. |
| Alternative: | -Steps:   1. Leave the website. 2. There is no errors |

* + 1. search:

|  |  |
| --- | --- |
| User case ID: | -uc-13 |
| Goal: | -Enable user to find an article, journalist, page and catigories. |
| Pre-conditions: | -The user shall log in the system |
| Post-conditions: | -User will be able to select rout from the available route list. |
| Error Situations: | -User is not registered in the system. |
| Actors: | -User. |
| Trigger: | -When the user presses on search button. |
| Main Success Scenario: | -Steps:   1. User fills “from” input field. 2. User fills “to” input field. 3. User presses the search button. |
| Alternative: | -Flow 1:   1. User forgets to fill “from” or “to” input field. 2. The related warning message is shown to the user to fill the input fields properly.   -Flow 2:  1. User fills the input fields properly.  2. The available routes will be  listed. |

4.2.14 block :

|  |  |
| --- | --- |
| User case ID: | -uc-14 |
| Goal: | -Enable admin to prevent the user from being on the system. |
| Pre-conditions: | -The admin shall enter the intended to block user’s profile page. |
| Post-conditions: | -Admin will be able to prevent user’s communication with the site. |
| Error Situations: | -Username is not exist or not correct. |
| Actors: | -Admin. |
| Trigger: | -When the admin presses on block button. |
| Main Success Scenario: | -steps:   1. admin enters the profile page of the user who is intended to be blocked.   2- admin presses the block user button.  3- The same page will be reloaded as marked as blocked. |
| Alternative: | steps:   1. admin open his account. 2. Reload the user page . 3. Press on block button.  * Errors:  1. Wrong username or not existing. |

4.2.15 translate :

|  |  |
| --- | --- |
| User case ID: | -uc-15 |
| Goal: | -Translate all the article for the user with 16 language. |
| Pre-conditions: | -No precondition. |
| Post-conditions: | -User shall be able to see the article in selected language. |
| Error Situations: | -There is no error |
| Actors: | -User. |
| Trigger: | -Once the user click on OK button. |
| Main Success Scenario: | - steps:  1. User clicks the translate  language button.  2. The system converts its  content in the selected  language. |
| Alternative: | -Steps:  1. The user clicks the “Turkçe” from the language list .  2. The article content will be  converted to English  language.  Flow 2:  1. The user clicks the “English”  from the language.  2. The article content will be  converted to Turkish  language. |

4.2.16 update:

|  |  |
| --- | --- |
| User case ID: | -uc-16 |
| Goal: | -Enable journalist to update any article  -Enable admin to update any thing from system requirements. |
| Pre-conditions: | -When updating system shall requires admin and journalist to enter password for verification |
| Post-conditions: | -Admin and journalist both are able to update any article or and system needs. |
| Error Situations: | -User name of journalist or admin is not true. |
| Actors: | -journalist and admin. |
| Trigger: | -when any of journalist or admin press on update button. |
| Main Success Scenario: | -Steps:   1. Admin and journalist log in they account 2. Admin and journalist do what they want to update 3. Admin and journalist click on update button |
| Alternative: | Flow 1:   1. If admin try to access a part of the system is not permitted 2. If journalist try to access a part of the system is not permitted |

4.2.17 live streaming :

|  |  |
| --- | --- |
| User case ID: | -uc-17 |
| Goal: | -Enable admin to share live from an accident or special event on the site. |
| Pre-conditions: | -Admin has an account on the site. |
| Post-conditions: | -Admin shall share a live on the website and every user can see it. |
| Error Situations: | -Admin log in with wrong password and username. |
| Actors: | -Admin. |
| Trigger: | -When admin press on share button. |
| Main Success Scenario: | -Steps:   1. Admin access his account 2. Admin start use camera from the site. 3. Admin share the live. |
| Alternative: | -Flow 1:  Admin access with wrong password |

4.2.18 comment :

|  |  |
| --- | --- |
| User case ID: | -uc-18 |
| Goal: | -Enable user to interact with an article by commenting. |
| Pre-conditions: | -user shall sign in to the system by his account. |
| Post-conditions: | -user shall comment on any article. |
| Error Situations: | -user does not access the system with his account. |
| Actors: | -user. |
| Trigger: | -when user click on button post. |
| Main Success Scenario: | -steps:   1. user access his account 2. user write his comment 3. user press on post button. |
| Alternative: | -user open his account  -user write his comment. |

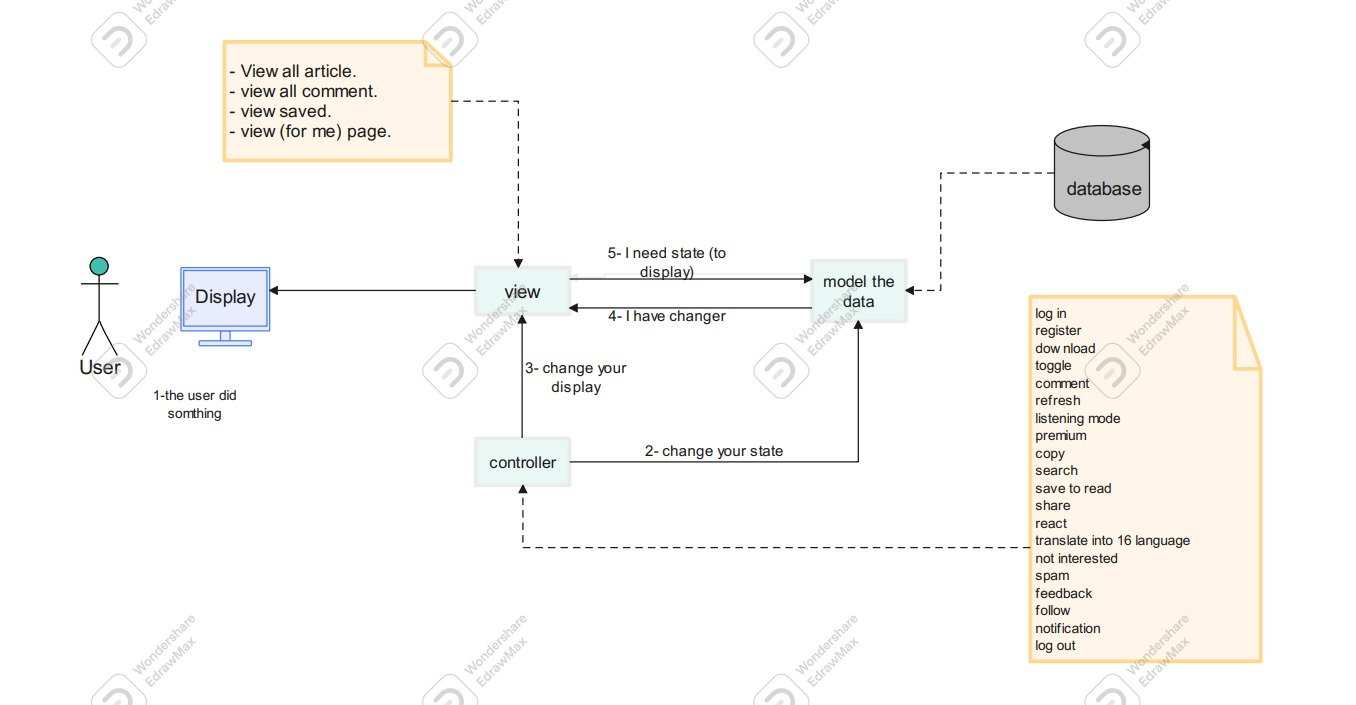
4.2.19 listening mode :

|  |  |
| --- | --- |
| User case ID: | -uc-19 |
| Goal: | -enable user to listen the article. |
| Pre-conditions: | -user shall sign in to the system by his account. |
| Post-conditions: | -user shall listen any article. |
| Error Situations: | -user does not access the system with his account. |
| Actors: | -user. |
| Trigger: | -when user click on button post. |
| Main Success Scenario: | -steps:   1. user access his account 2. user select the article. 3. user press on listen button. |
| Alternative: | -user open his account  -user listen to the artical . |

5.0 System Architecture

5.1 System Architecture

Model-View-Controller Architecture



Layered Architecture Pattern



5.2 System Architecture Description

System architecture is a model used to define the behavior and structure of the system. It is a formal way to describe and represent the system.

The software architecture consists of components, and it compares between these components, its visible properties and the relation between them. it also separates the overall structure of the system from the internal details. When we use the software architecture system, we will be able to solve complexity, reuse some parts and support flexibility and future evolution.

The software architecture design starts from Highest Abstraction level to lowest levels. We must be sure that the design achieves the functional and non-functional requirements. Nonfunctional requirements may cause a change in the architecture design, and this make it very important

# MODEL-VIEW-CONTROLLER ARCHITECTURE:

*Model:*

Holds all the data, state and application logic. Oblivious to the View and Controller. Provides API to retrieve state and send notifications of state changes to “observer”..

*View:*

Gives user a presentation of the Model. Gets data directly from the Model.

*Controller:*

5.3 System Architecture Description Takes user input and figures out what it means to the Model

# B) Layered Architecture:

*The presentation layer:*

it has the user interface for patient, admin and doctor. The business logic layer and Data access

*layer:*

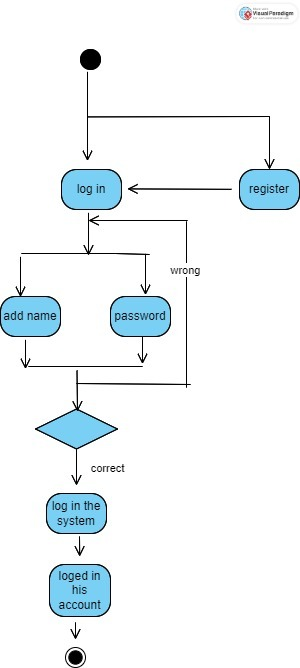
It contains function of each user in the system and handling function.

*The database layer:*

This is where all the data is stored.

6.0 Activity Diagrams

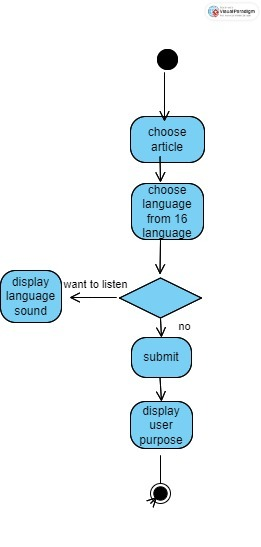
6.1 log in



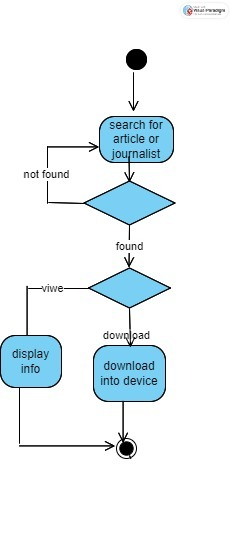
6.2 Delete



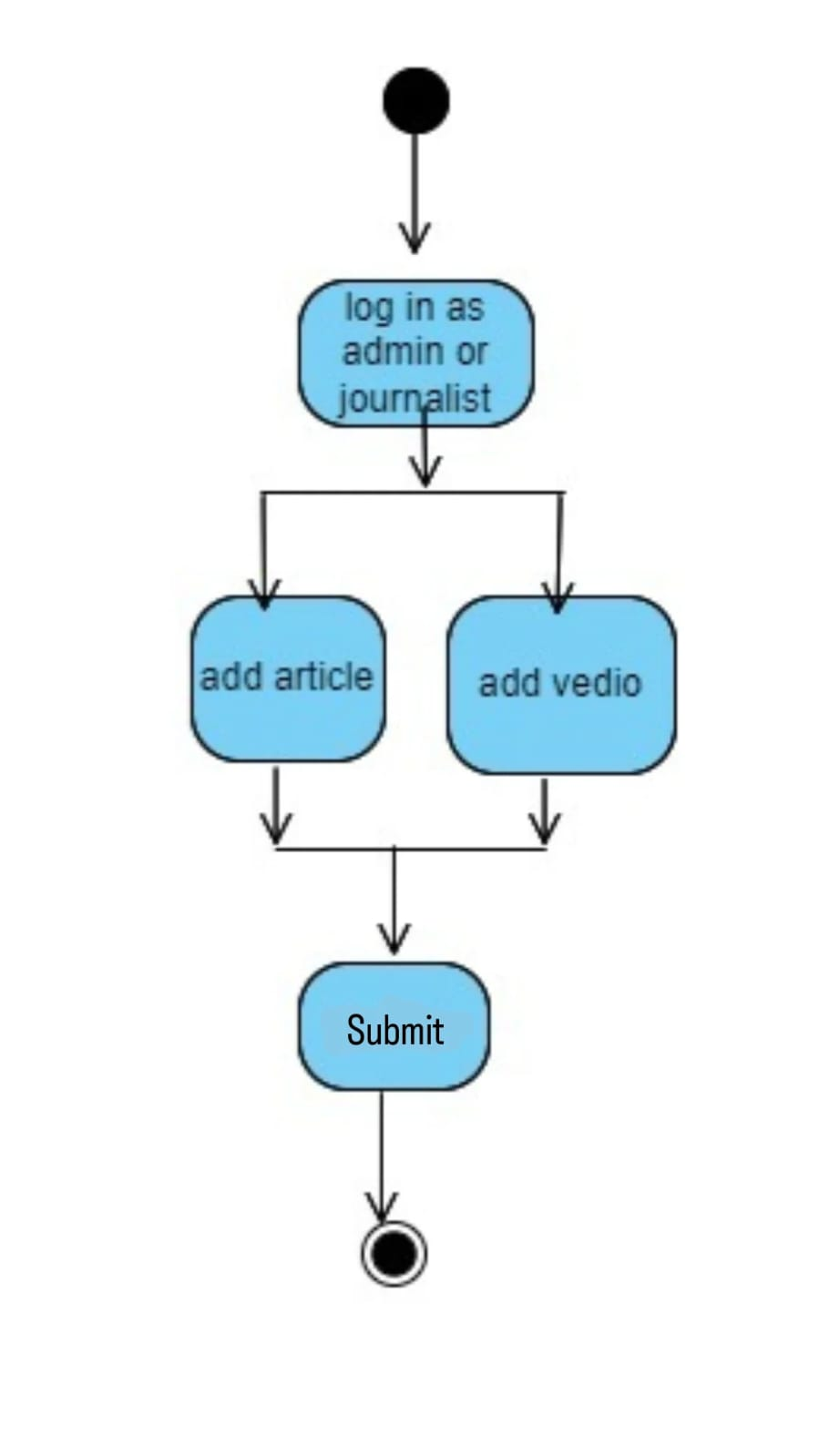
6.3 Translate and Listen



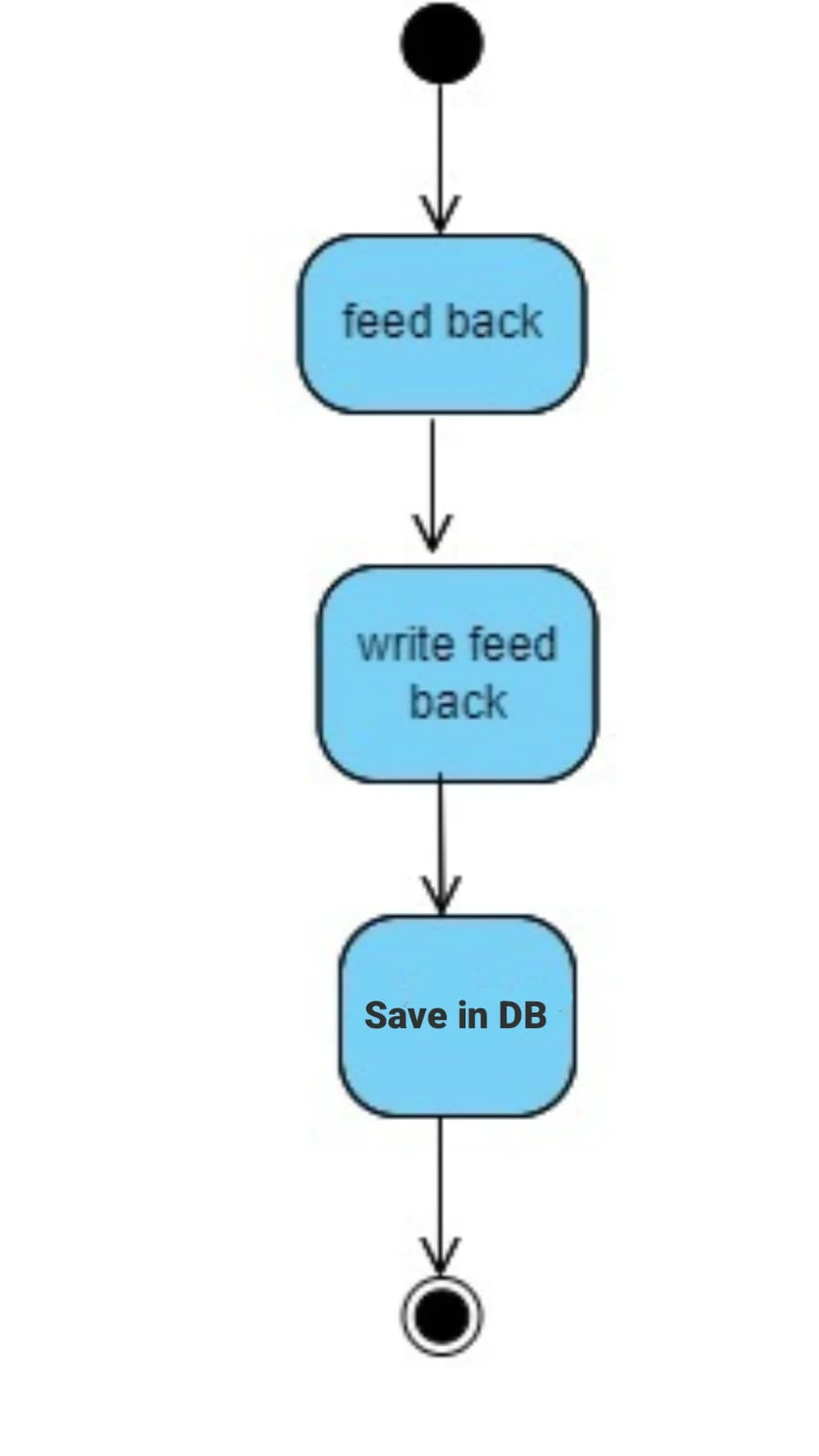
6.4 Search



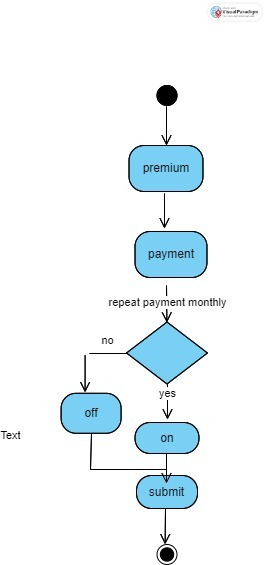
6.5 Add article



6.6 Feedback

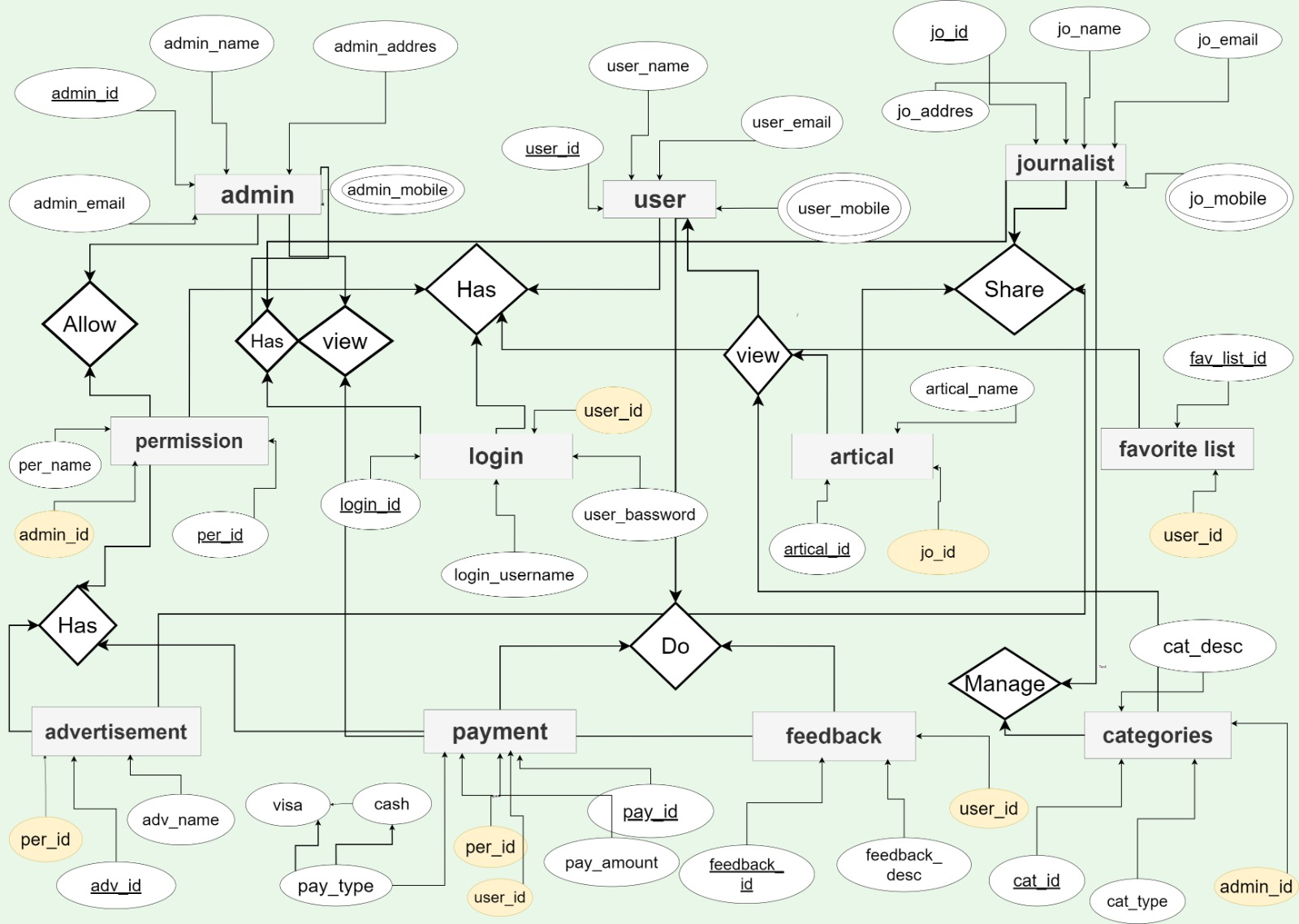


6.7 Premium

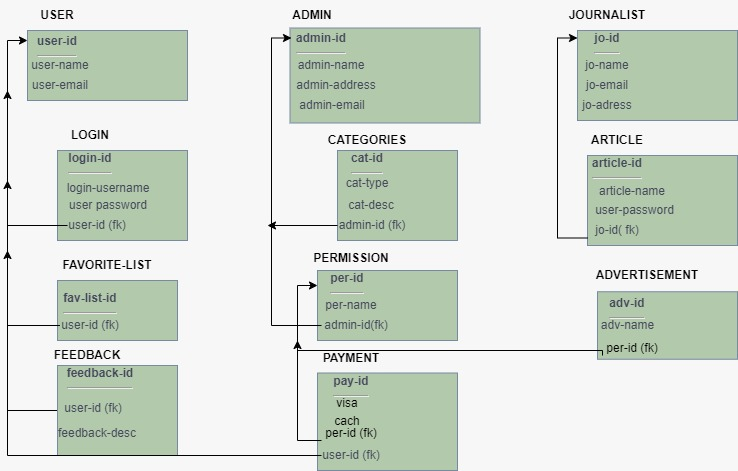


7.0 Database Specification

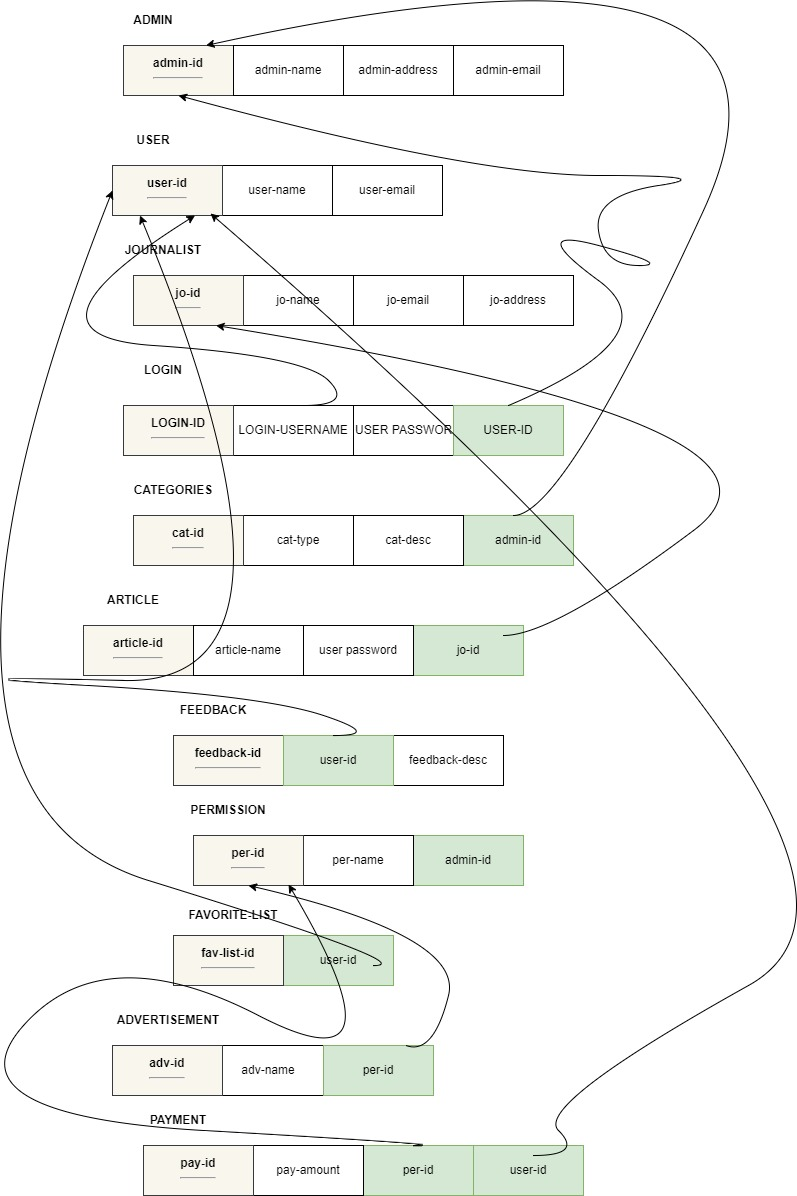
7.1 ERD



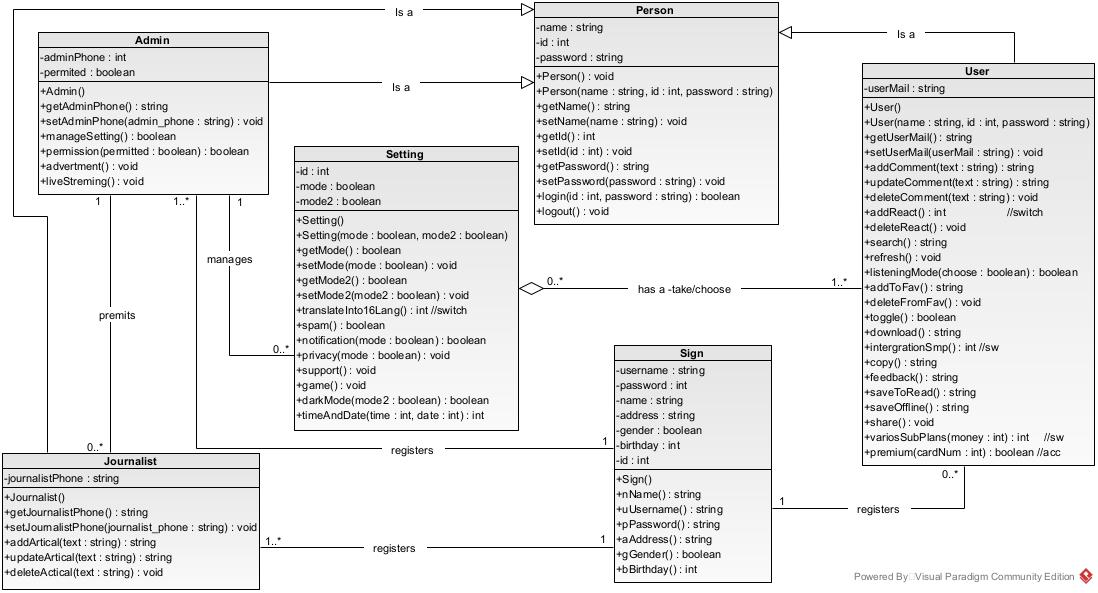
7.2.1 Tables



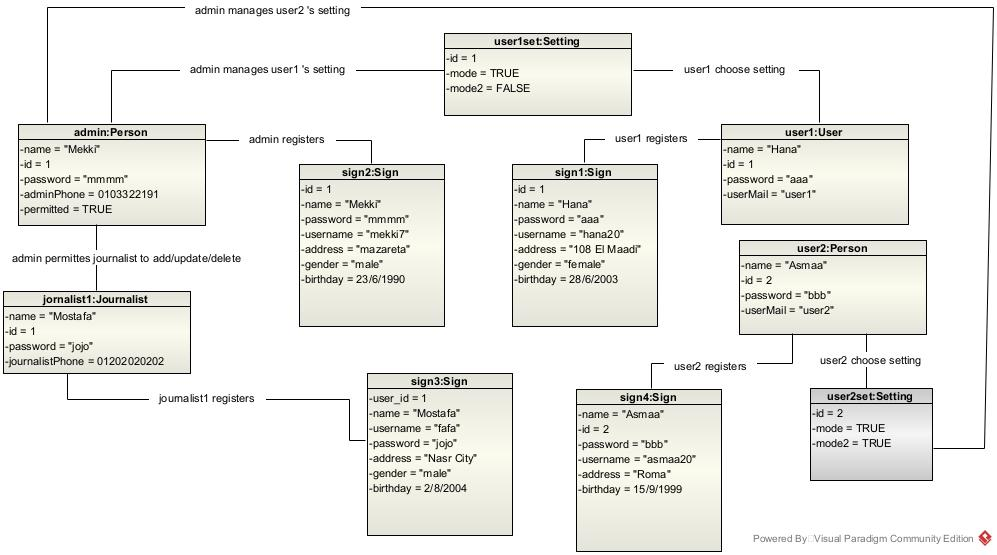
7.2.2 Tables



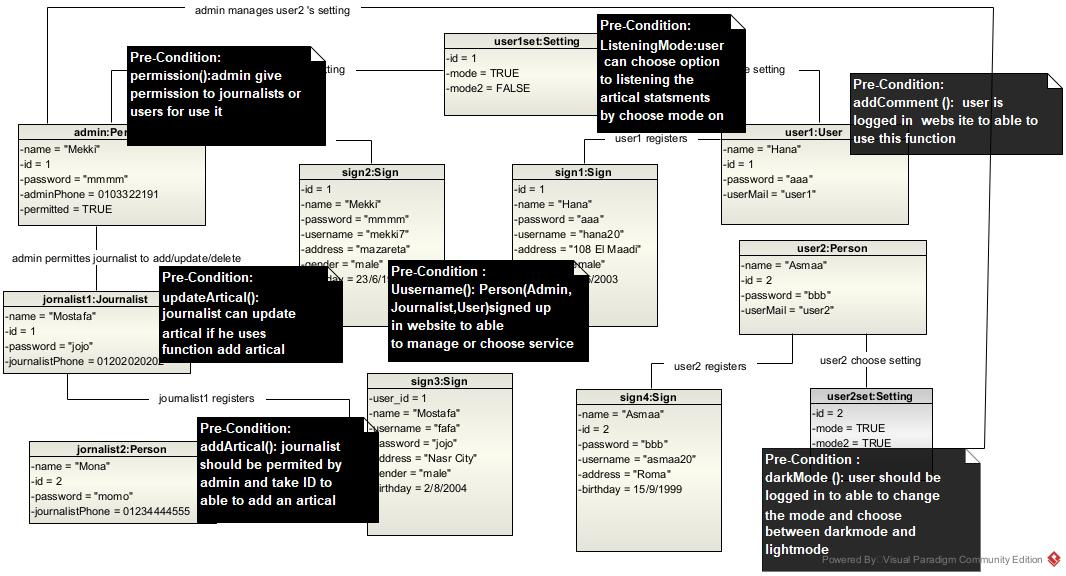
8.0 Class Diagram



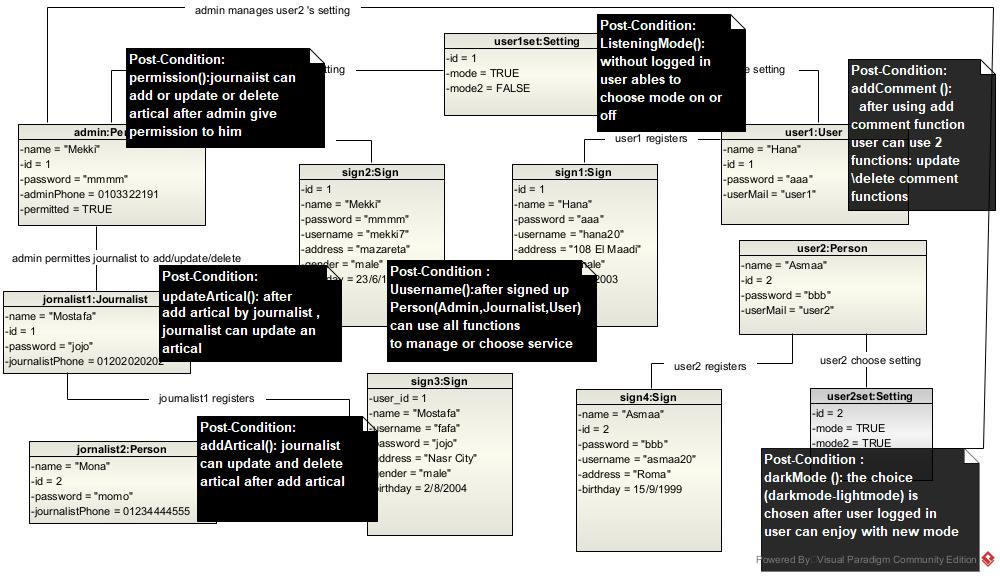
9.0 Object Diagram



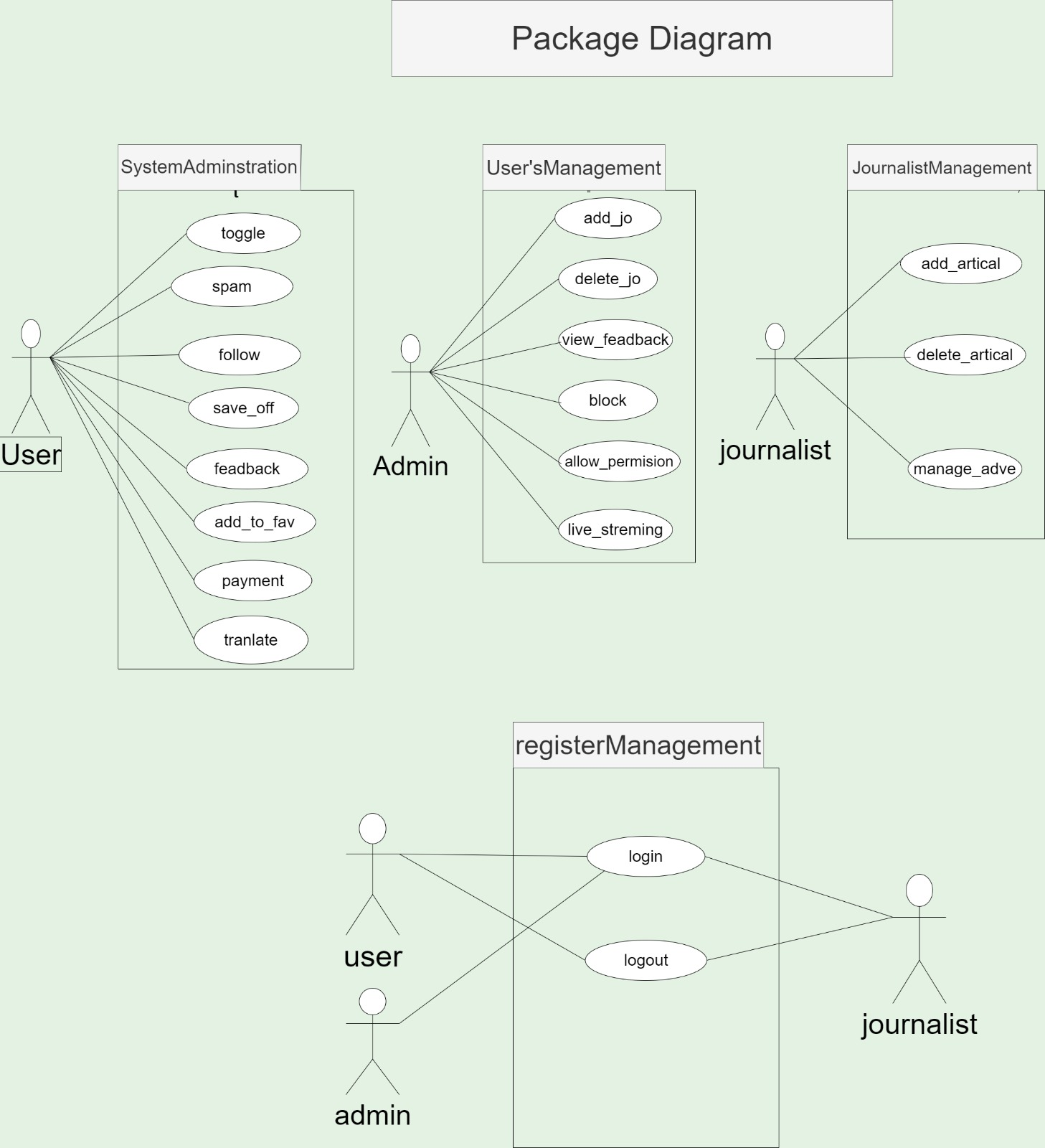
9.1 Pre-Condition



9.2 Post-Condition

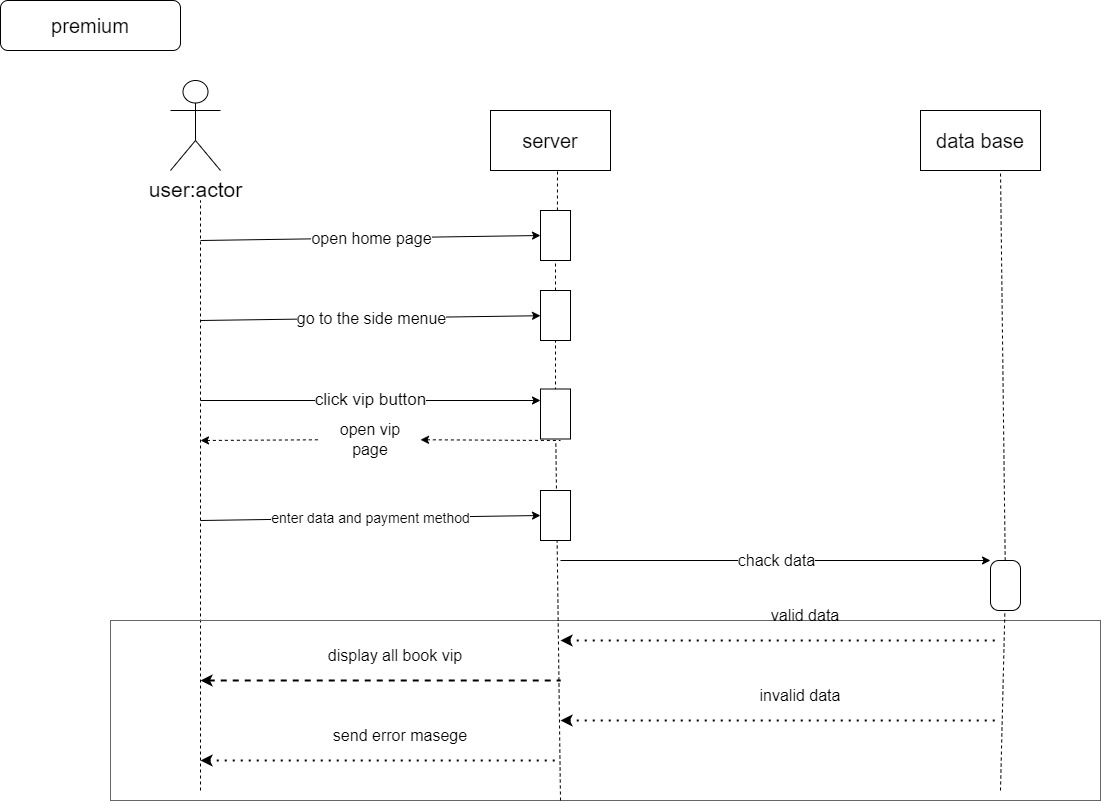


10.0 Package Diagram

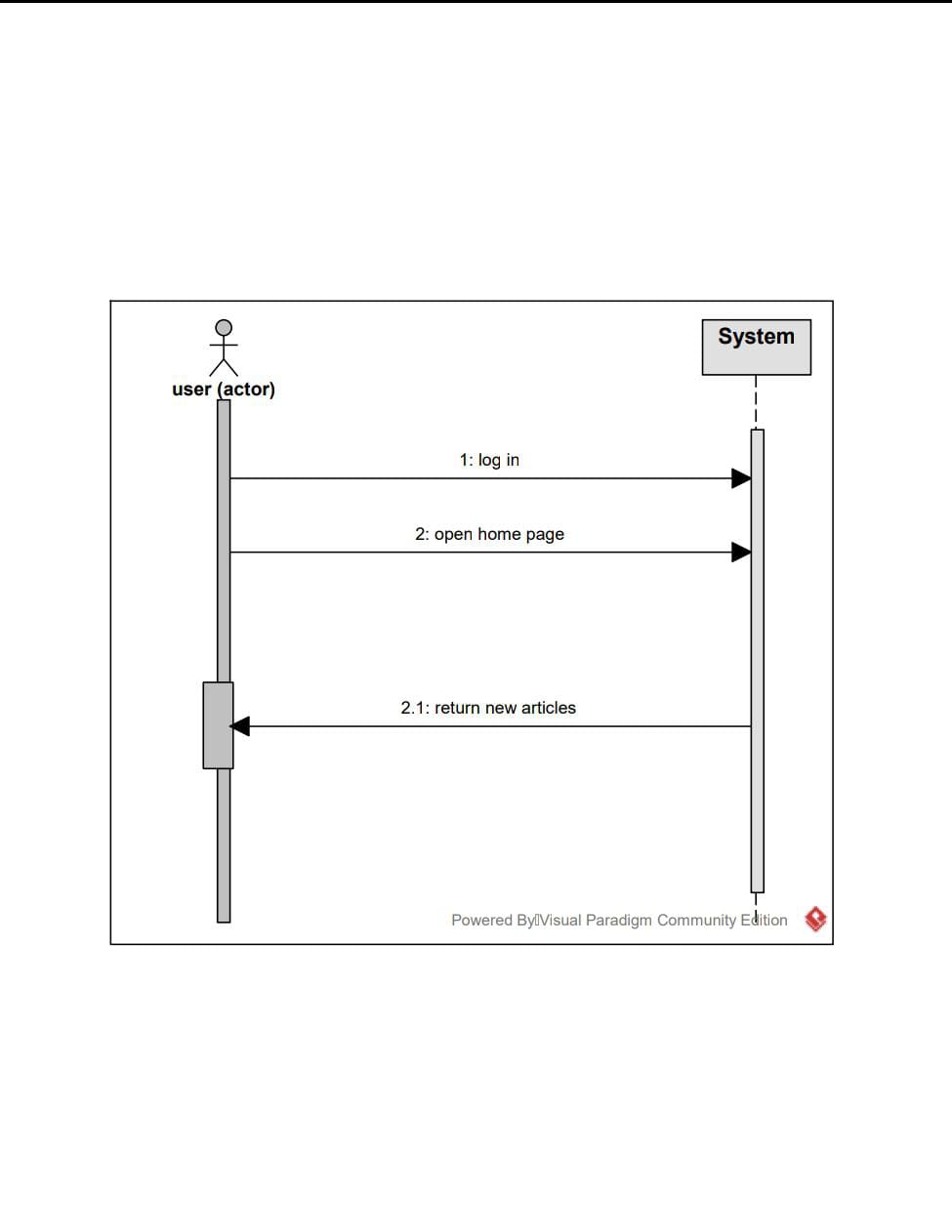


11.0 Sequence Diagram

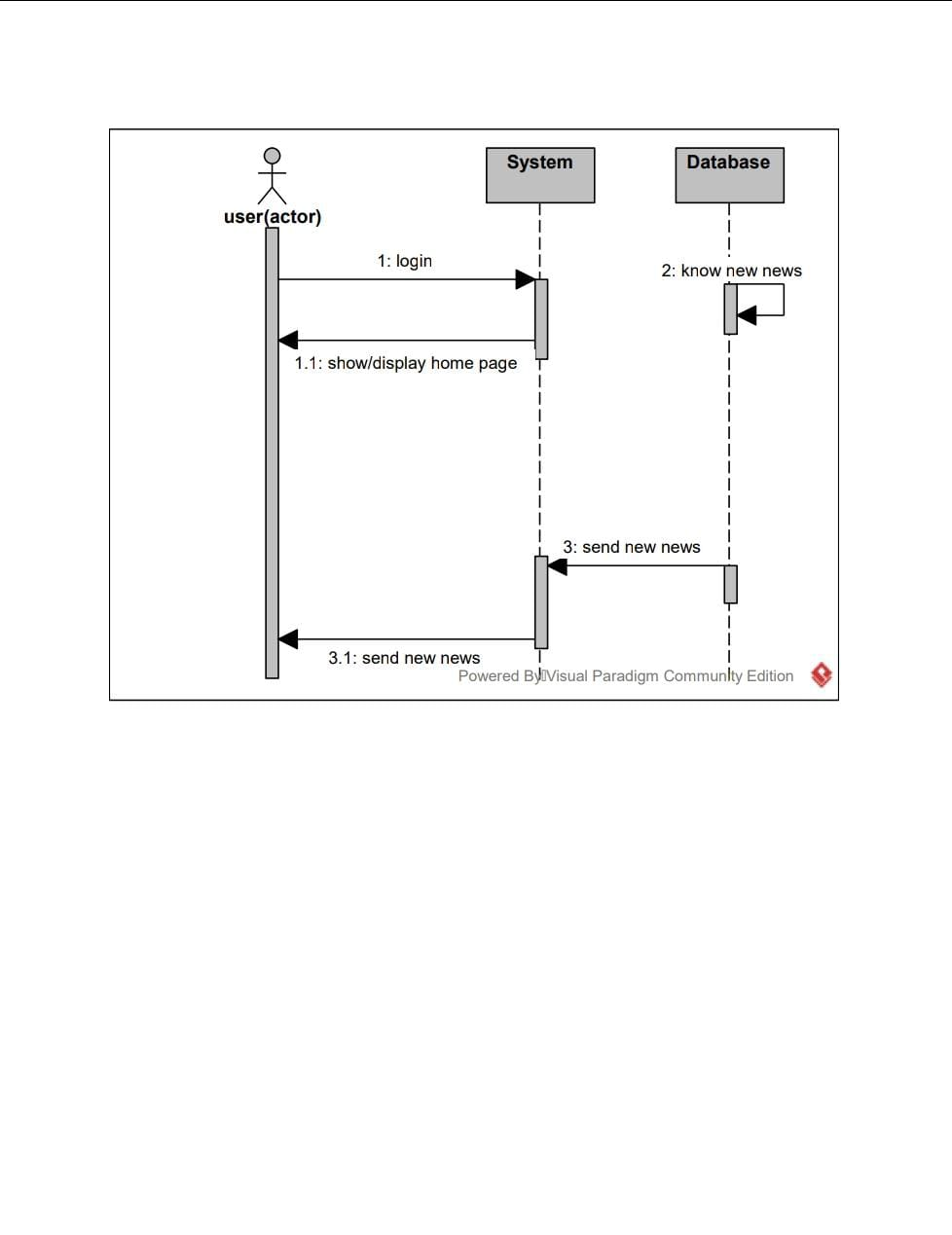
11.1 premium



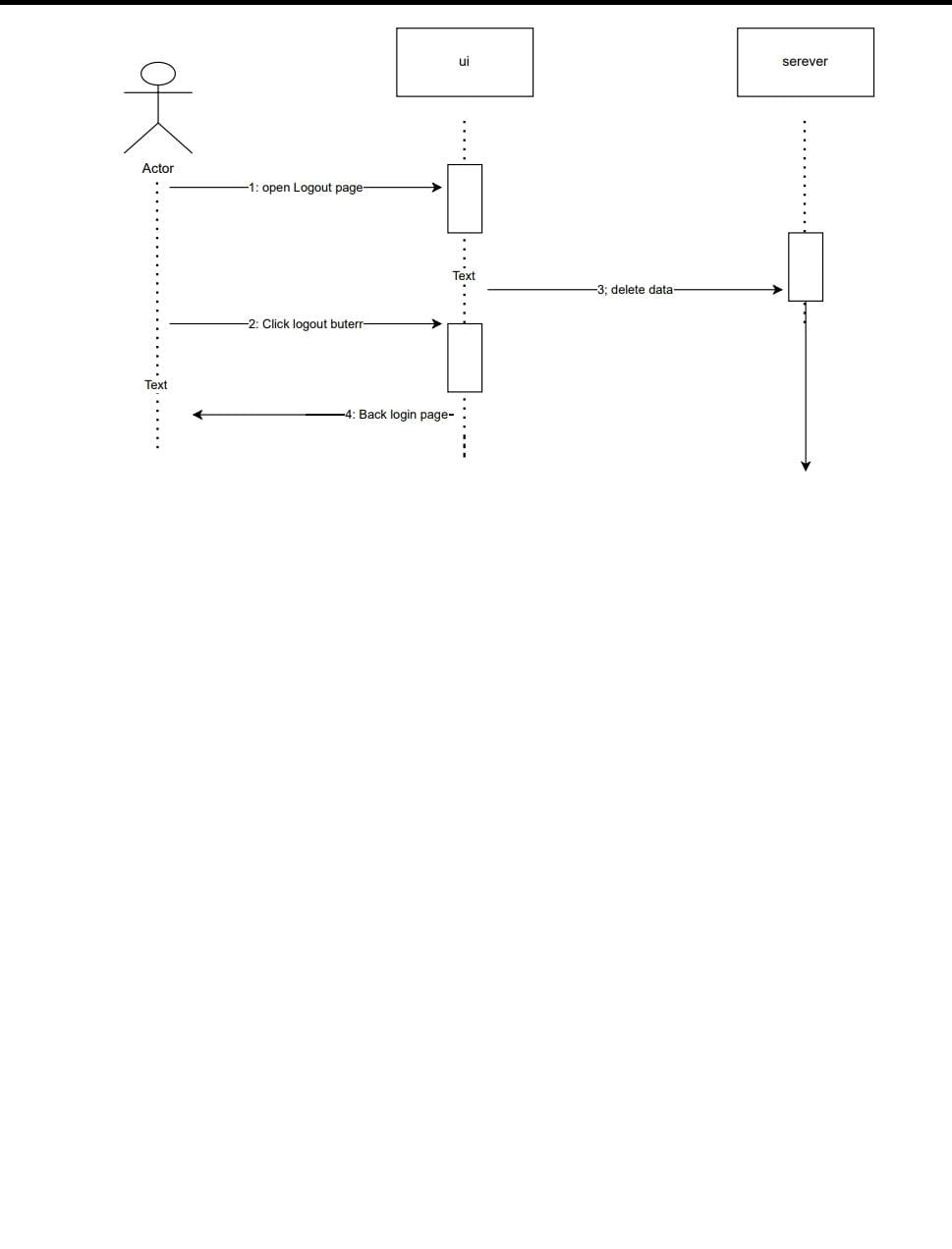
11.2 refresh



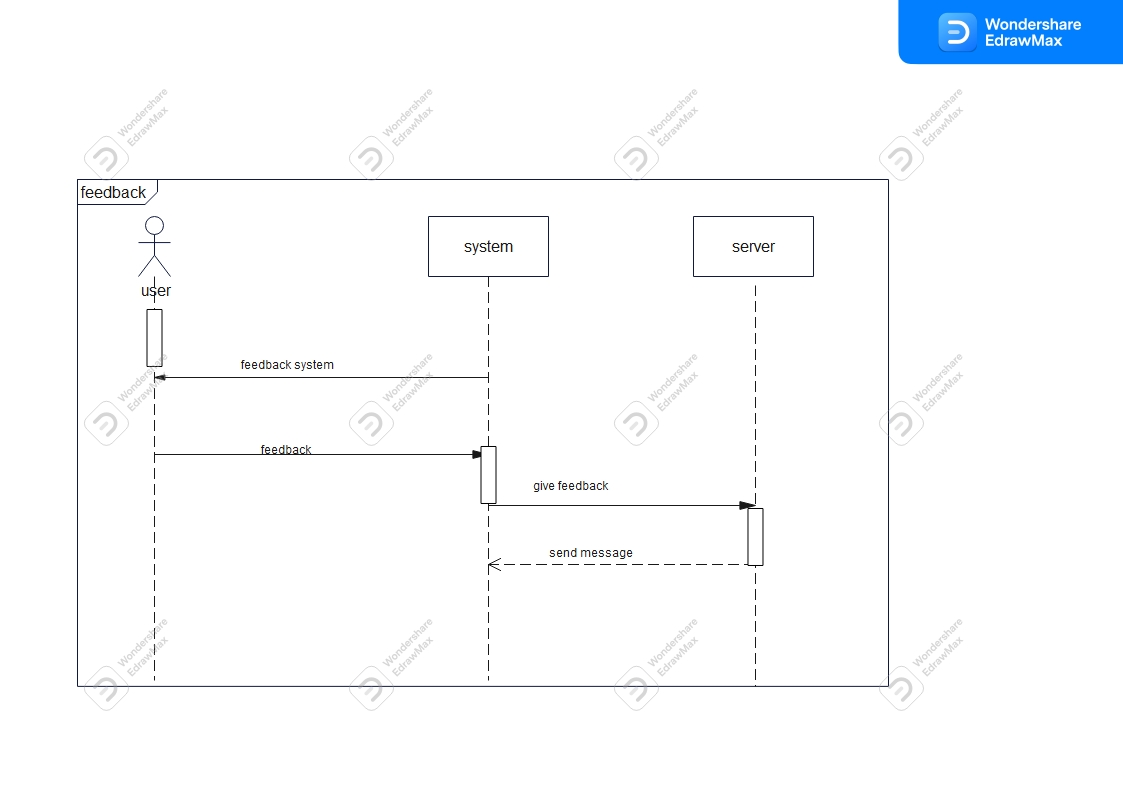
11.3 notification



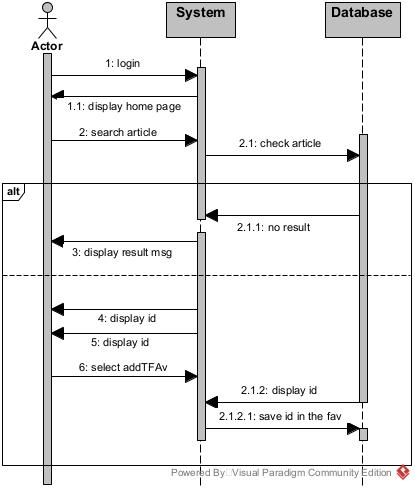
11.4 log out



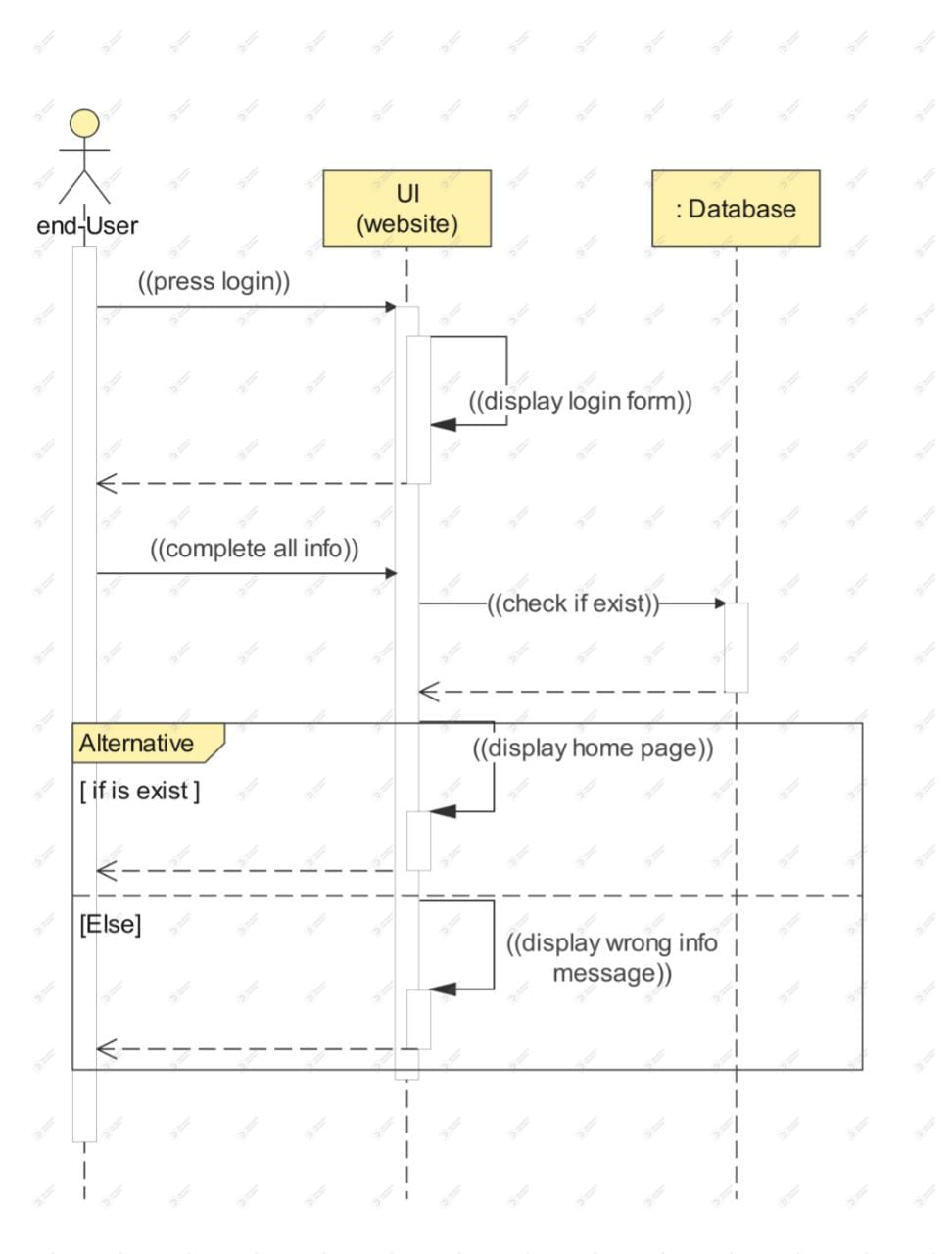
11.5 feedback



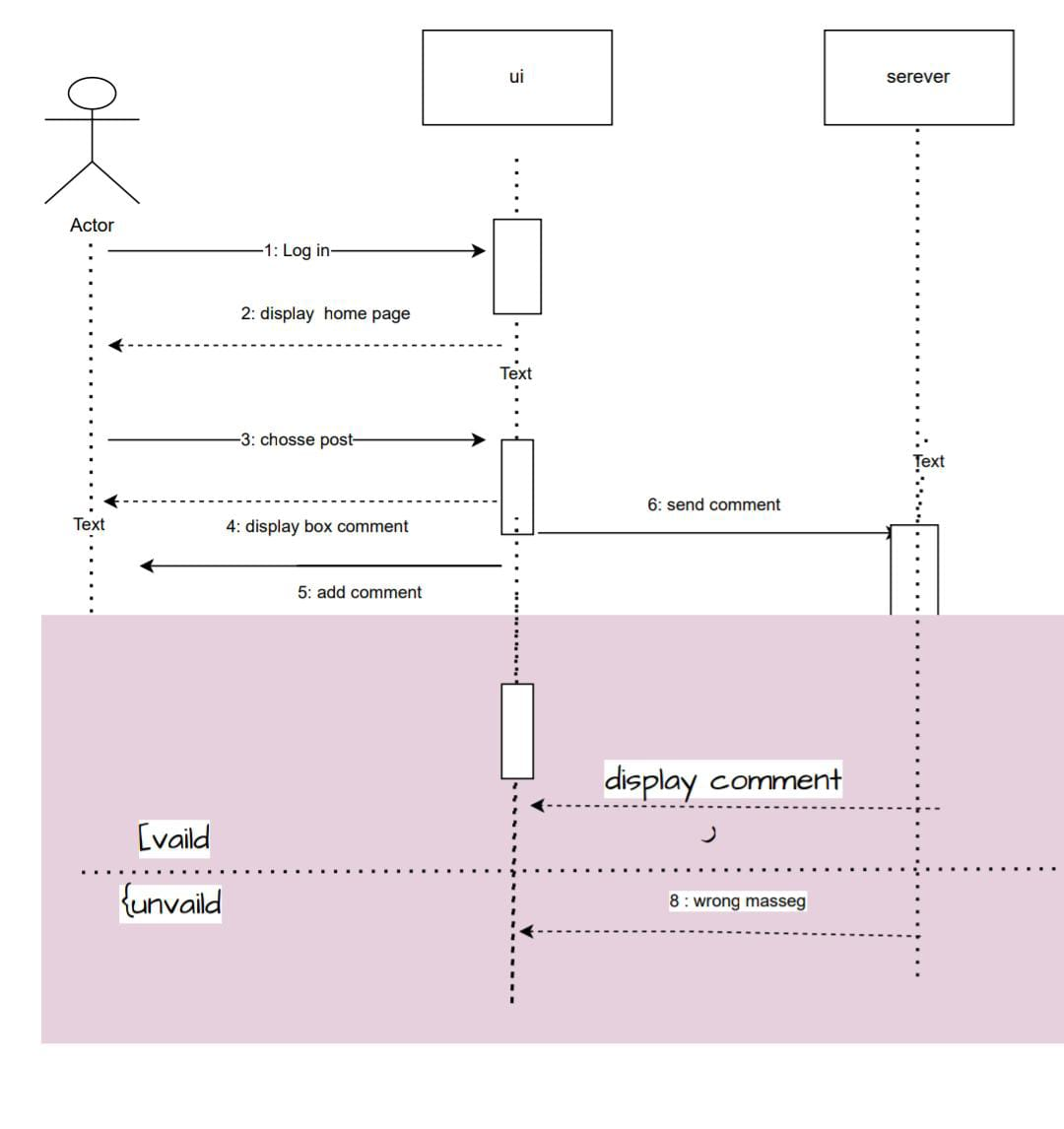
11.6 Add to favorite



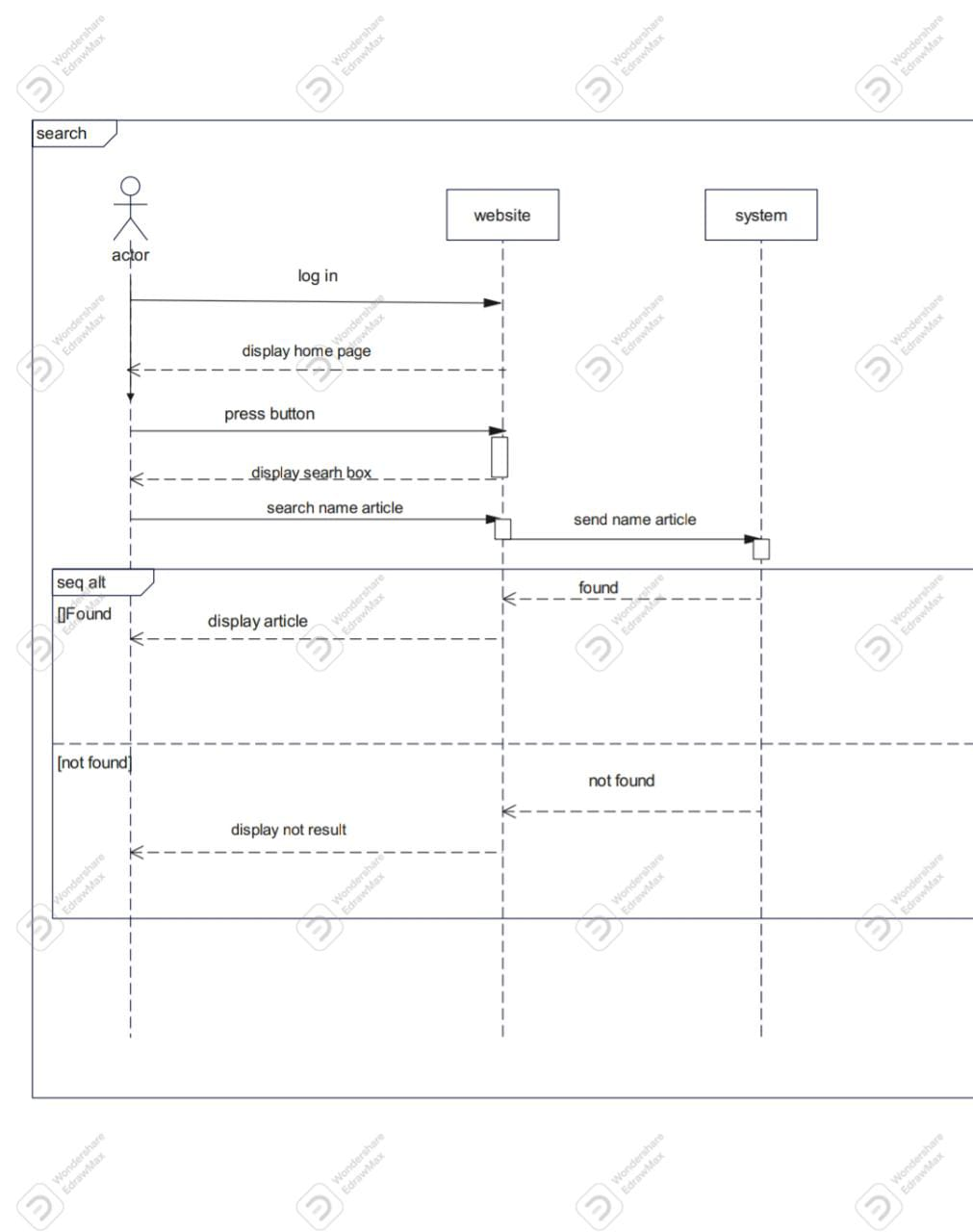
11.7 log in



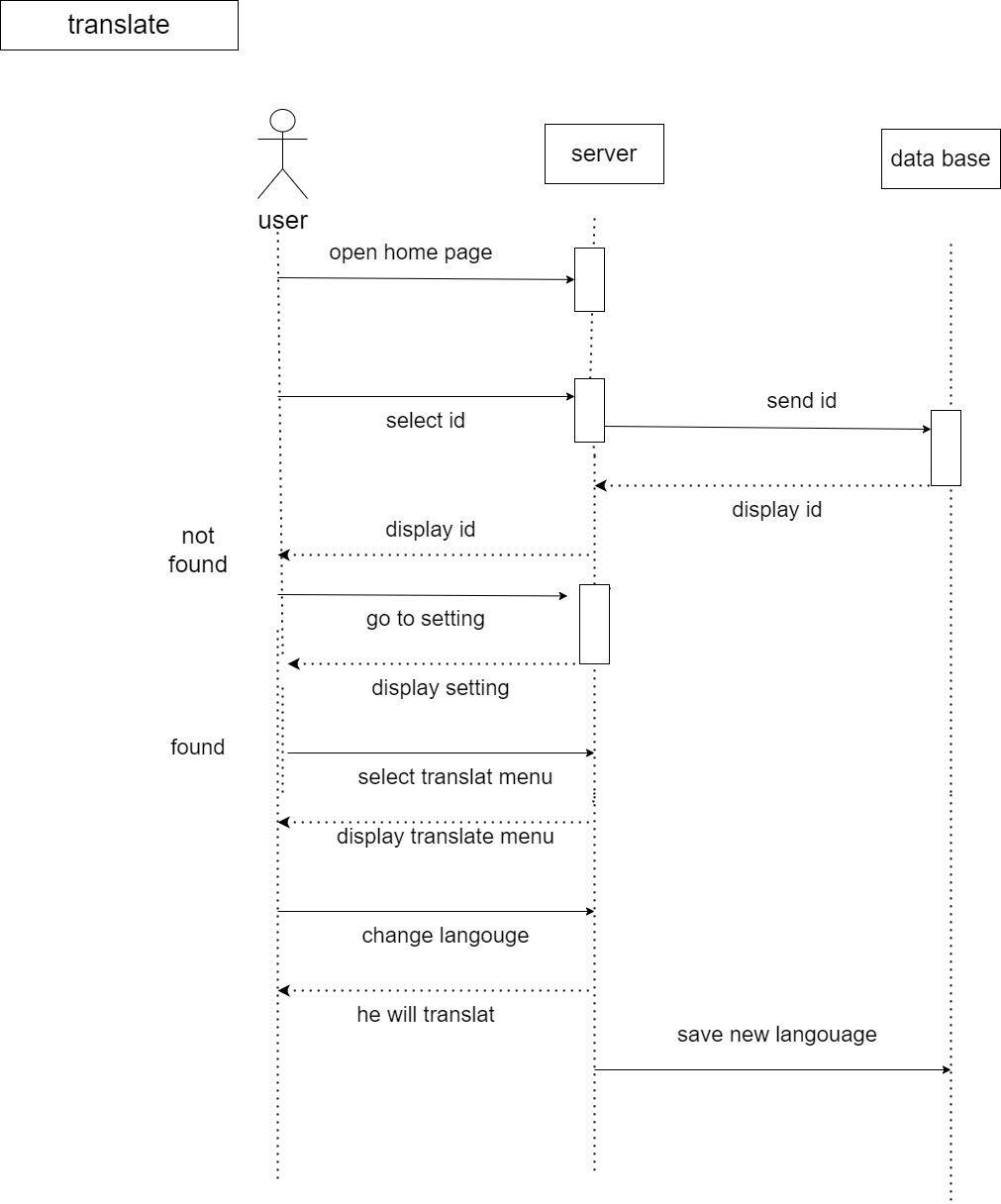
11.8 Comment



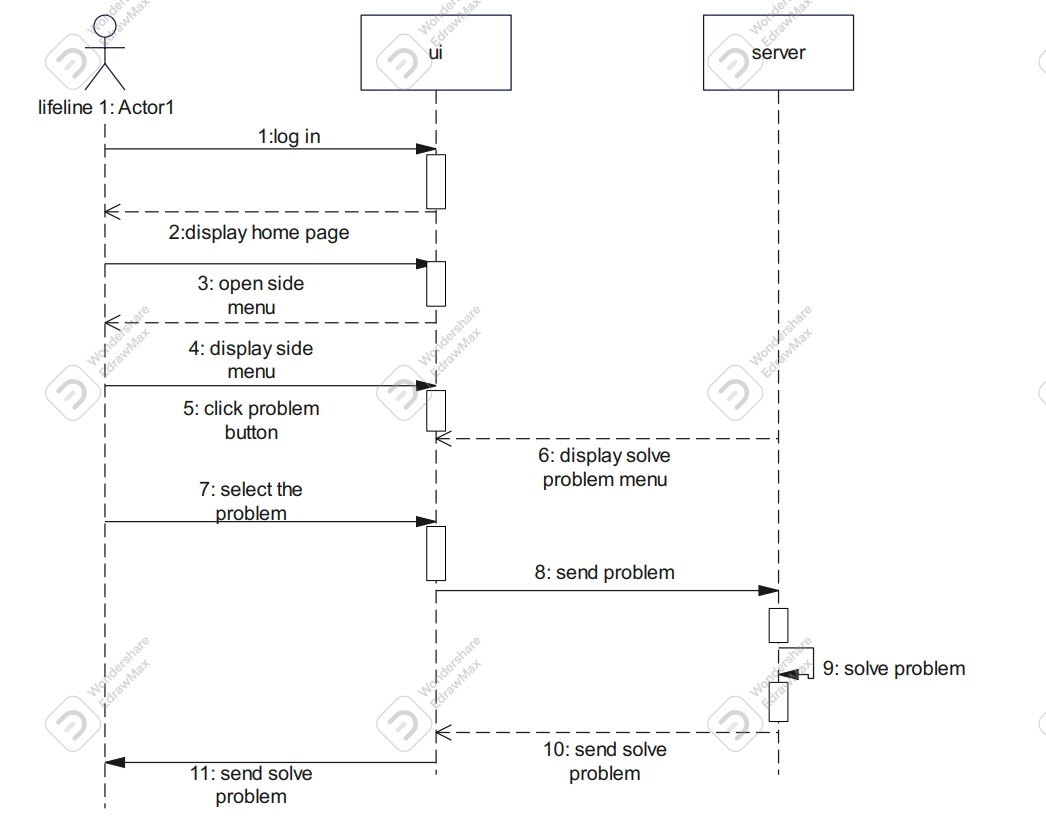
11.9 search



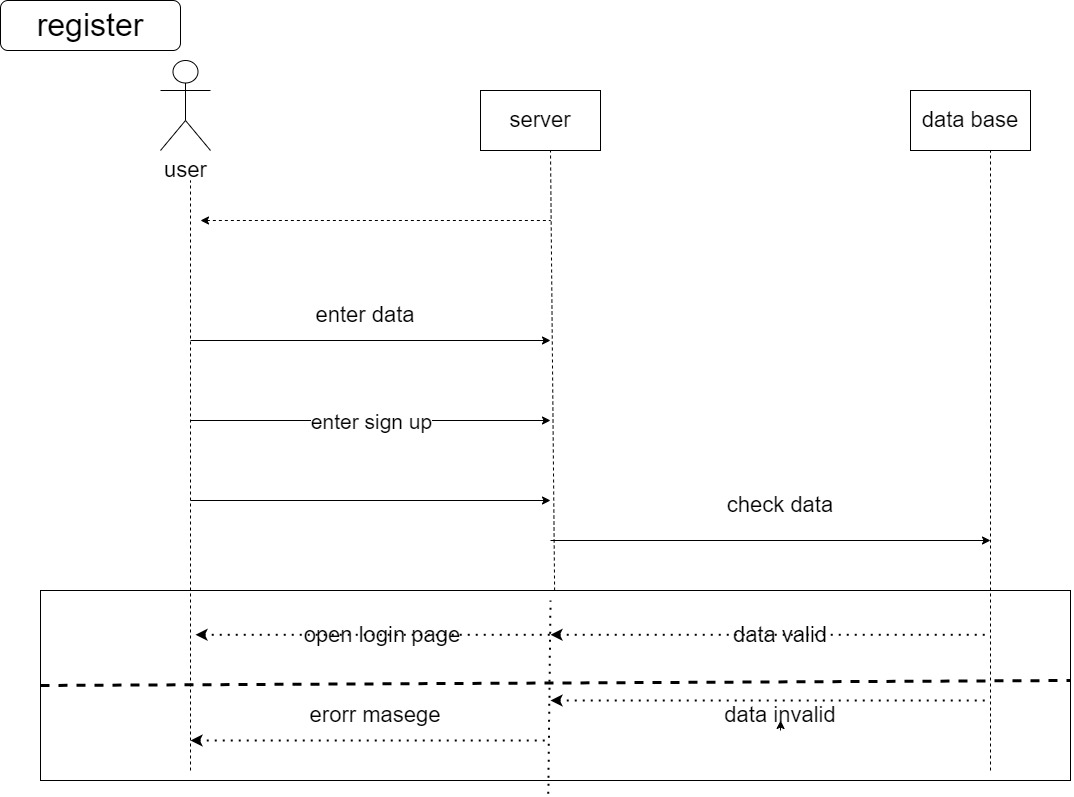
11.10 Translate

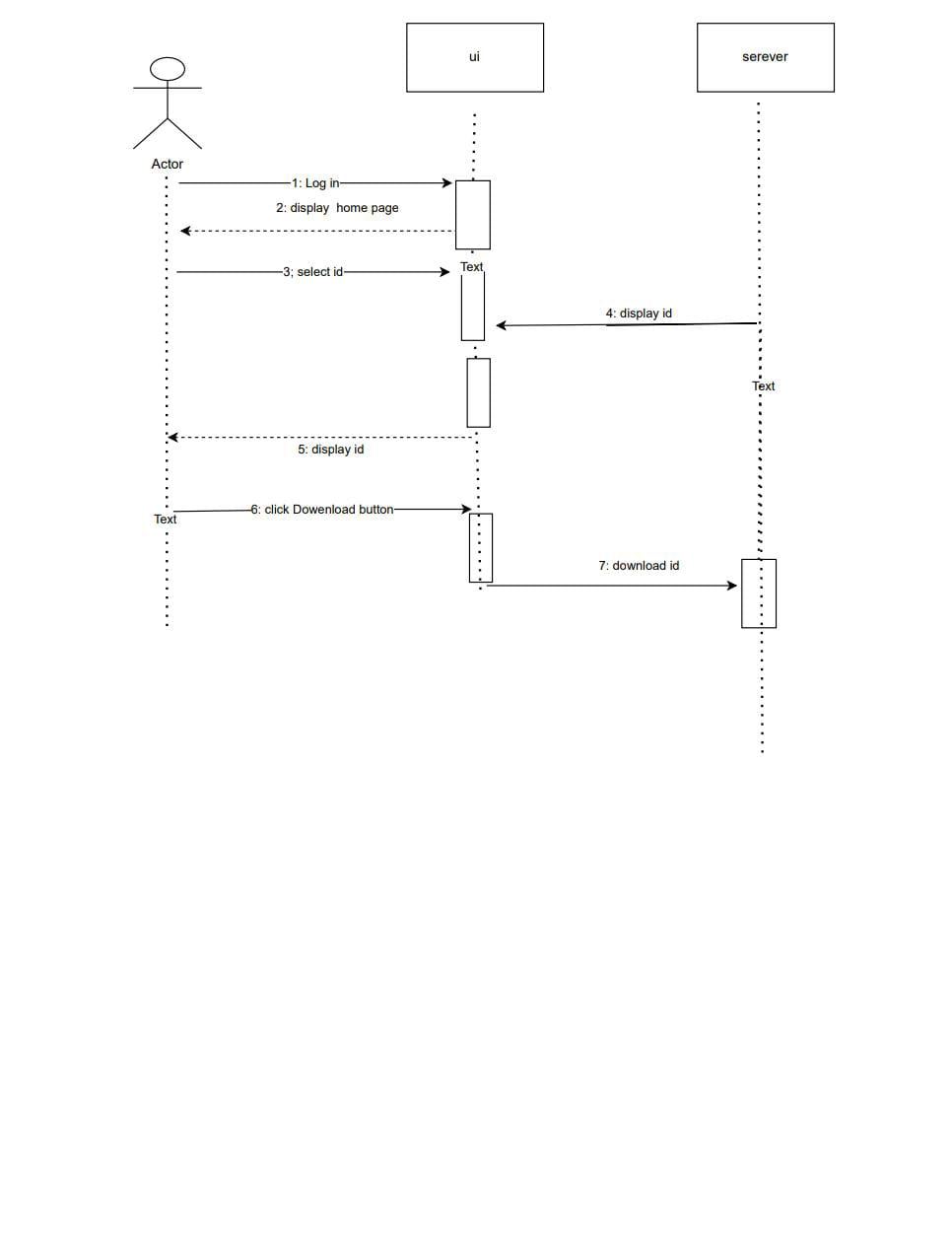


11.11 Support



11.12 Register



11.13 Download

11.14 payment

