**Bia**

1. **Business.**

Customer transactions at the bank or internet banking system customer transactions at the bank will be supported.

**Business requirements.**

***Customer*:**

View Profile, Edit profile, change password, view Transaction.

***Support:***

Add/modify/disable and remove accounts when it is approved or marked as removable.

Search accounts based on ID Card number, Name, Account type/number, state, phone & address, add funds to account, withdraw funds from account, transfer funds from one account to another.

Search accounts based on ID Card number to view account balance amount and transactions in a given period. Only active account can perform transactions.

Change state: New 🡪 active 🡪 disable 🡪 removable, disable 🡪 active

***Admin:***

Verify account: New 🡪 active, disable 🡪 removable.

Verify and approve transactions.

1. **Applying technology.**
   1. **Introducing the Spring Framework and Spring MVC and JPA.**
      1. **Spring Framework.**

Open-source framework used to build programs for applications on the Java platform.

* J2EE applications simpler.
* The system is capable of maintaining good .
* System reliability and ease of control.
  + 1. **Spring MVC**

Spring MVC is one of the important modules of the Spring framework.

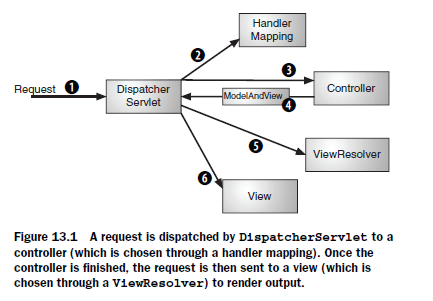
It Inheriting the advantages of Spring and it has become one of the famous framework for developing web applications .

Spring MVC makes building web applications closely, flexibility and compliance with the Model -View -Controller.

* There is a division between the Controller , Models and Views.
* Having the ability to integrate other frameworks.
* Spring MVC is very flexible it allows user interface for configuration, process and lugin.
* Provide concealment mechanism technology.
  + 1. **JPA**

The Java Persistence API (JPA) is a Java application programming interface specification that describes the management of relational data in applications using Java Platform, Standard Edition and Java Platform, Enterprise Edition.

* It helps to work with the database easily.
* Simplify technology for persistence layer (data layer).
* Do not depend on the ORM framework.
  + 1. **Spring MVC**



DispatcherServlet is sending a request to Spring MVC Controller (1).

DispatcherServlet will consider one or more handler mappings (step 2) to find out where is the destination of the next request.

DispatcherServlet requests to the controller is selected (3).

Controller will be packaged data model and the view to the object name ModelAndView (4).

Once ModelAndView been sent to DispatcherServlet, DispatcherServlet find JSP (5).

Sends data model, the model data is kept going to view, requirements have been completed (6).

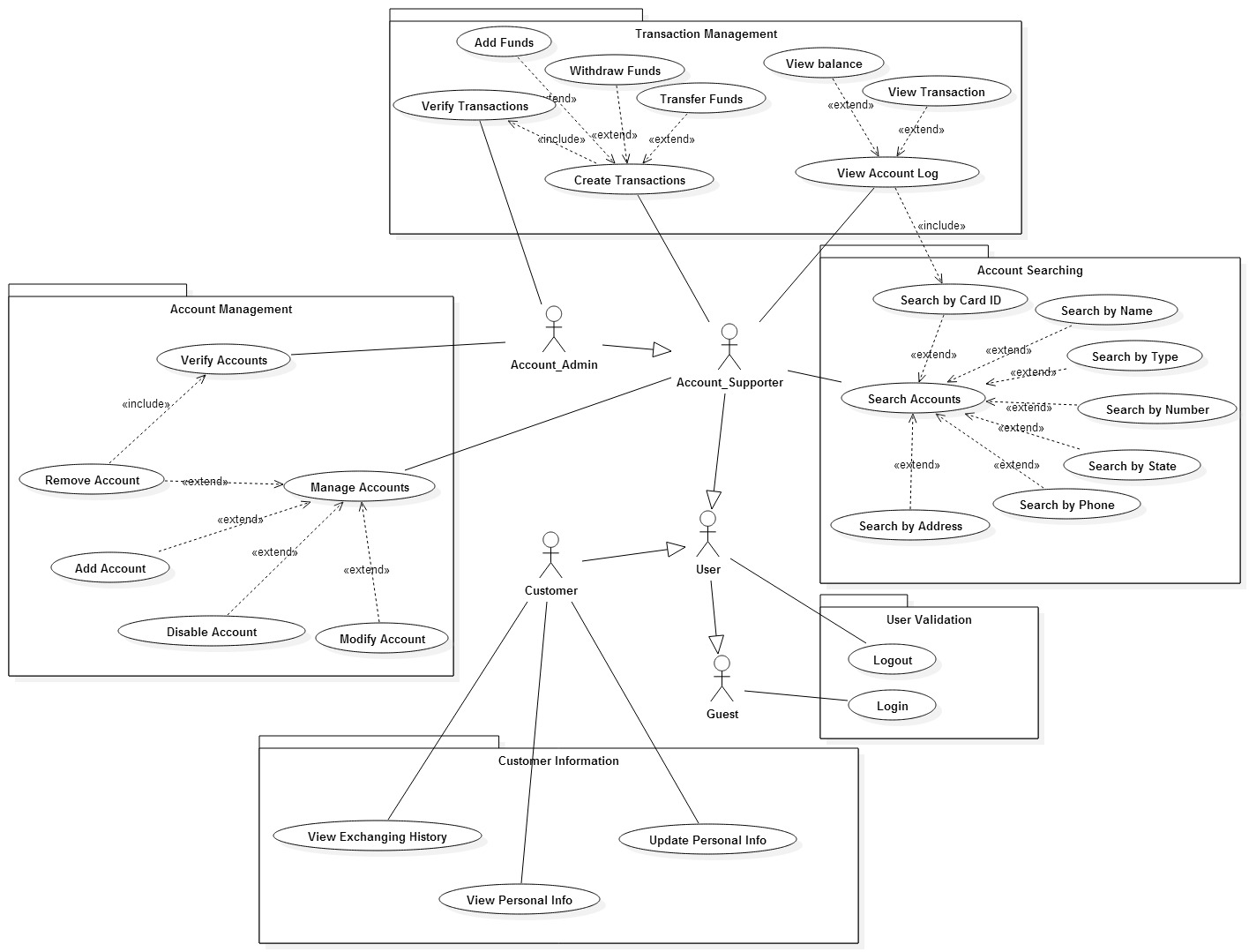
* + 1. **Annotation.**

It helps to configure Spring easily

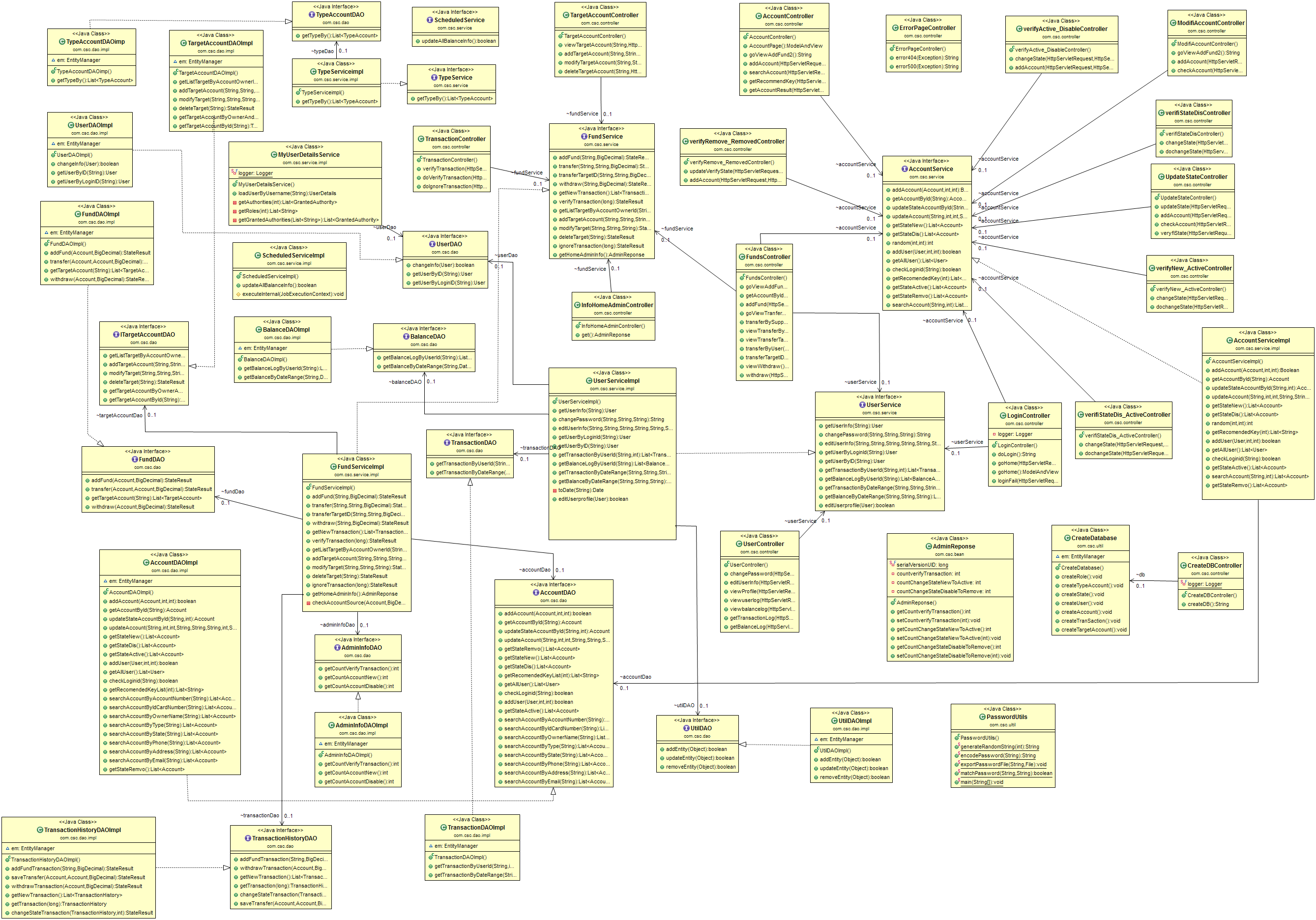
* @Controller
* @SessionAttributes
* @Autowired
* @RequestMapping
* @PersistenceContext
* @Transactional

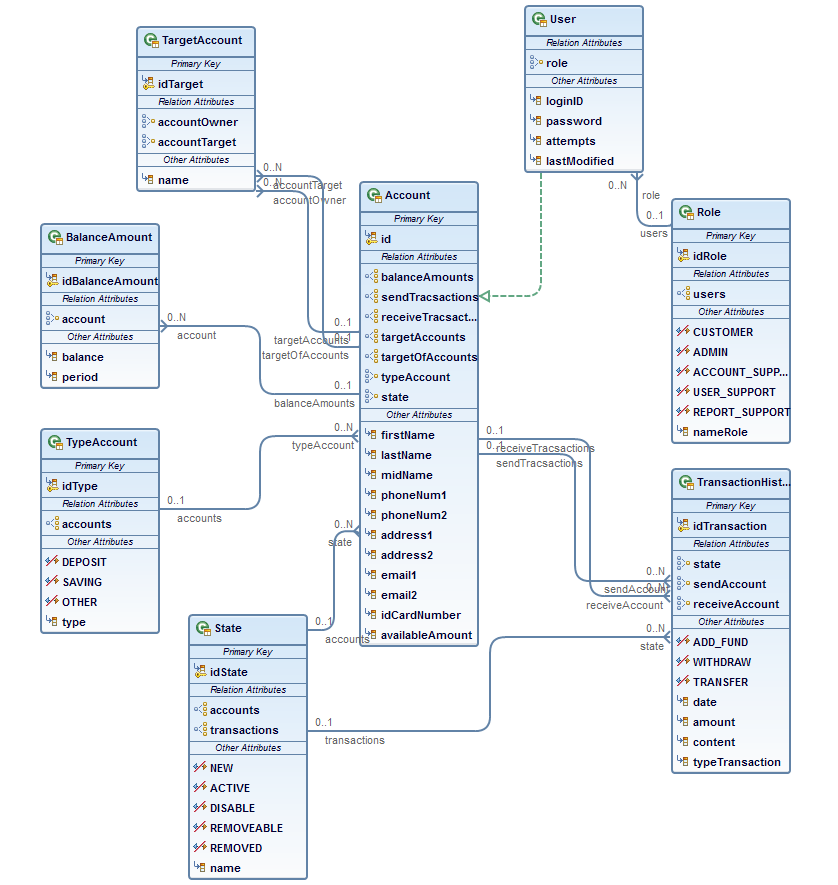
|  |  |  |
| --- | --- | --- |
| **Annotation** | **Use** | **Description** |
| @Autowired | Constructor, Field,  Method | Provides more fine-grained control over where and how autowiring should be accomplished. The @Autowired annotation can be used to autowire bean on the setter method just like @Required annotation, constructor, a property or methods with arbitrary names and/or multiple arguments. |
| @Configurable | Type | Used with <context:springconfigured> to declare the type that its properties can be injected |
| @Controller | Type | Configuring it is Spring MVC controller. |
| @Service | Type | Configuring it is Service. |
| @RequestMapping | Method ( | Mapping a URL pattern and / or the ending types HTTP to Controller |
| @SessionAttributes | Type | the burners stored in session. |

1. **Analyze and design systems**
   1. **Usecase diagram.**

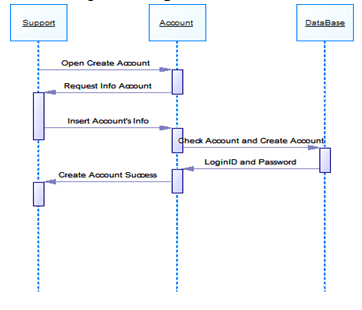


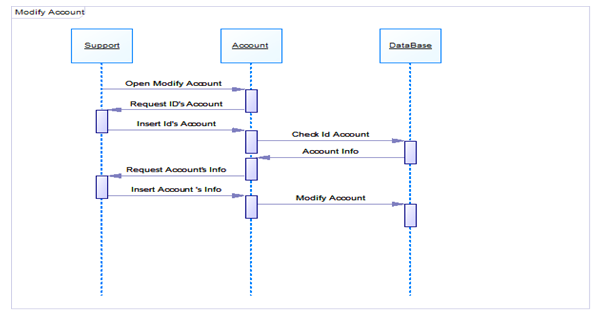
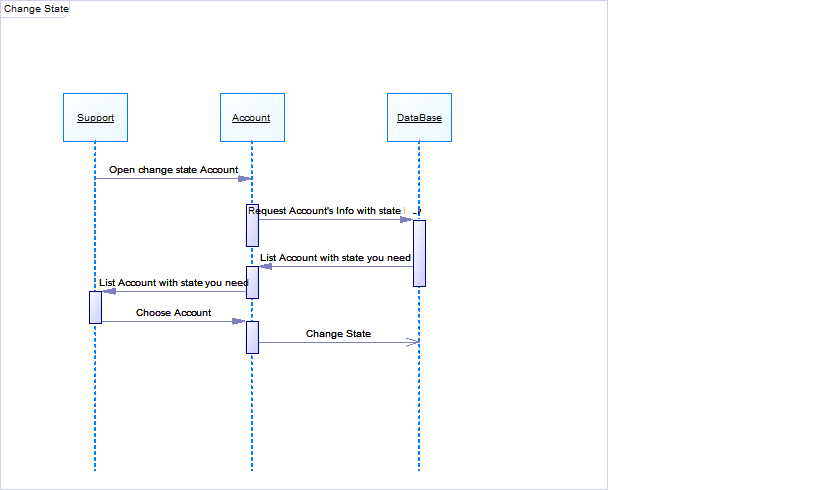
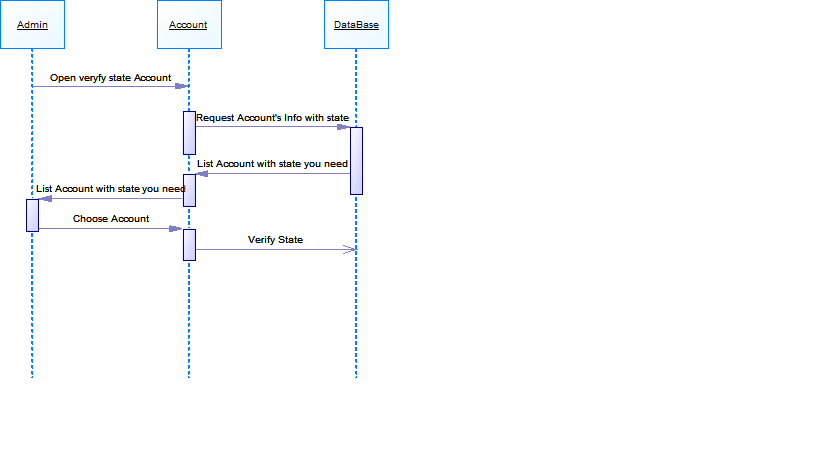
* 1. **Class Diagram**

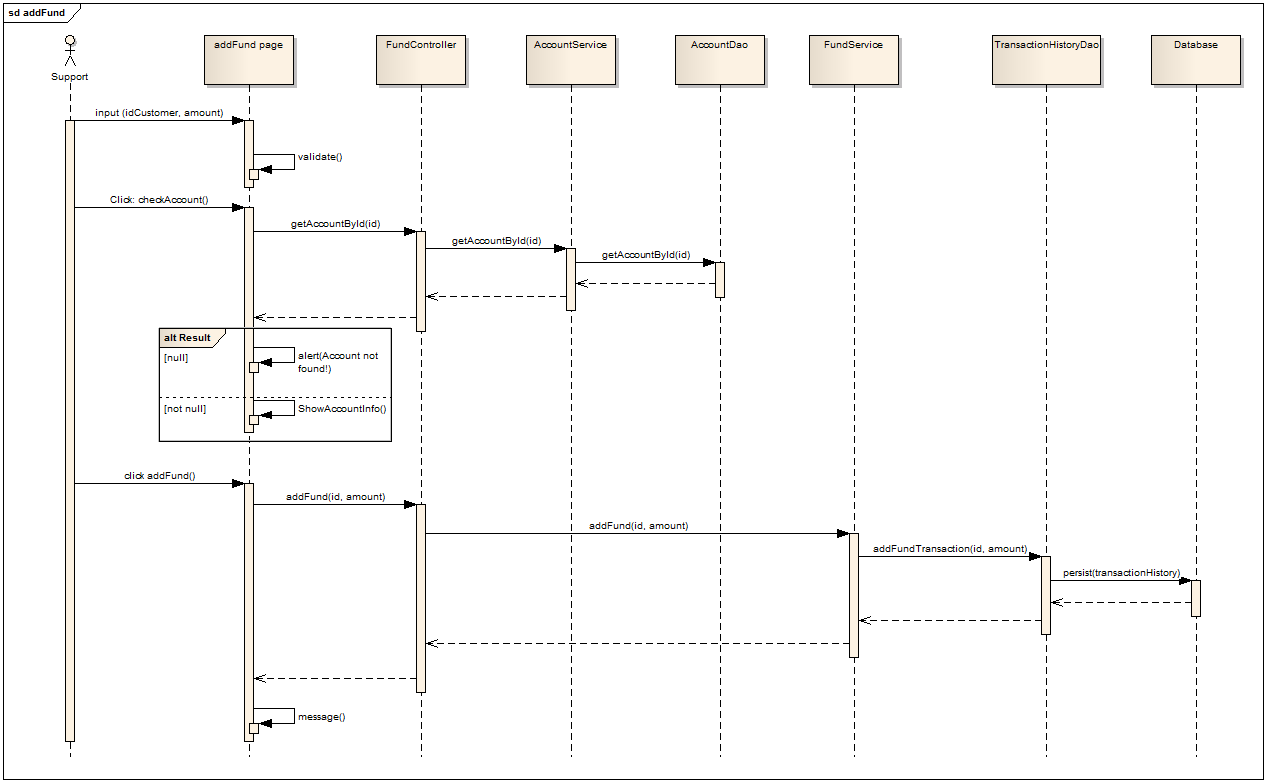


**Entity class diagram.**

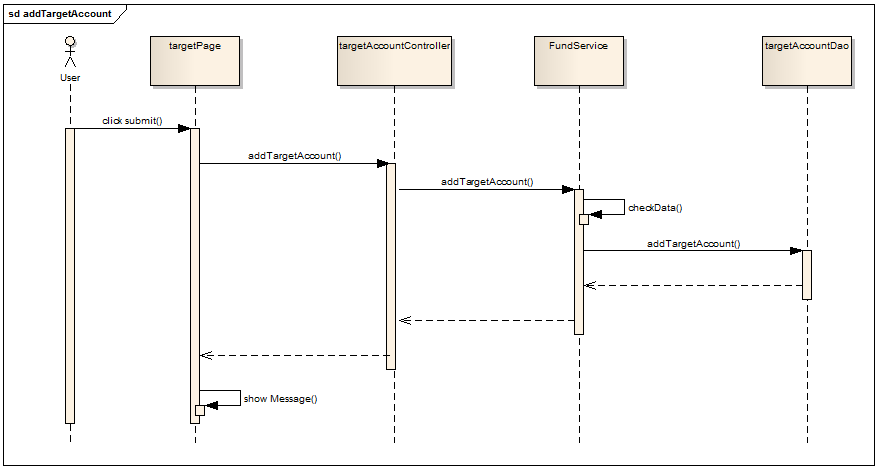
* 1. **Sequence diagramAdd Account:**
     1. **Add account.**

****

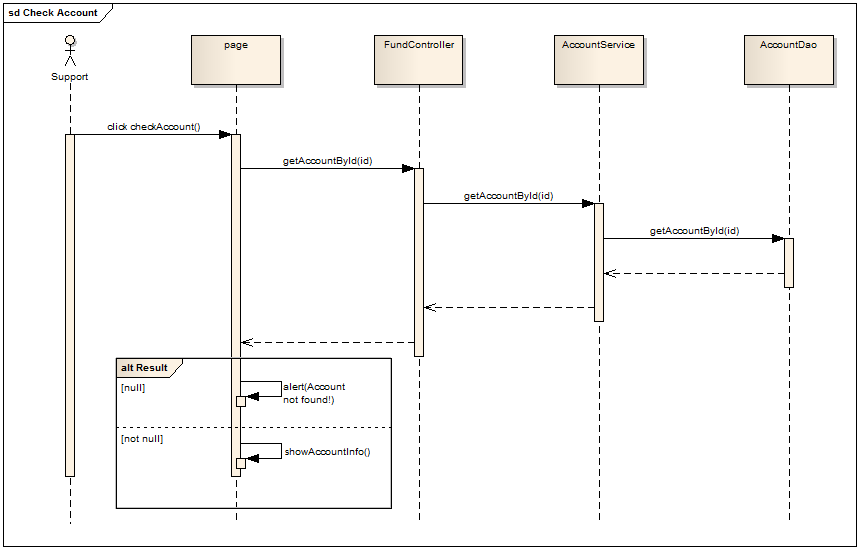
* + 1. **Modify Account.**
    2. **Change State**
    3. **Verify account.**
    4. **Add Fund**



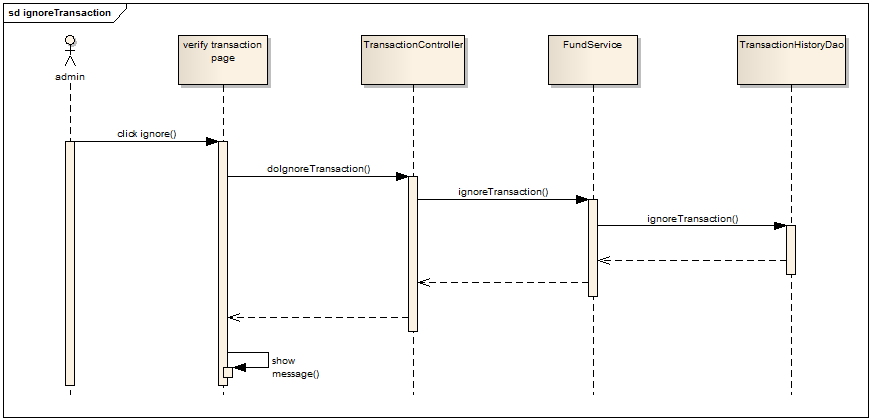
* + 1. **Add TargetAccount**



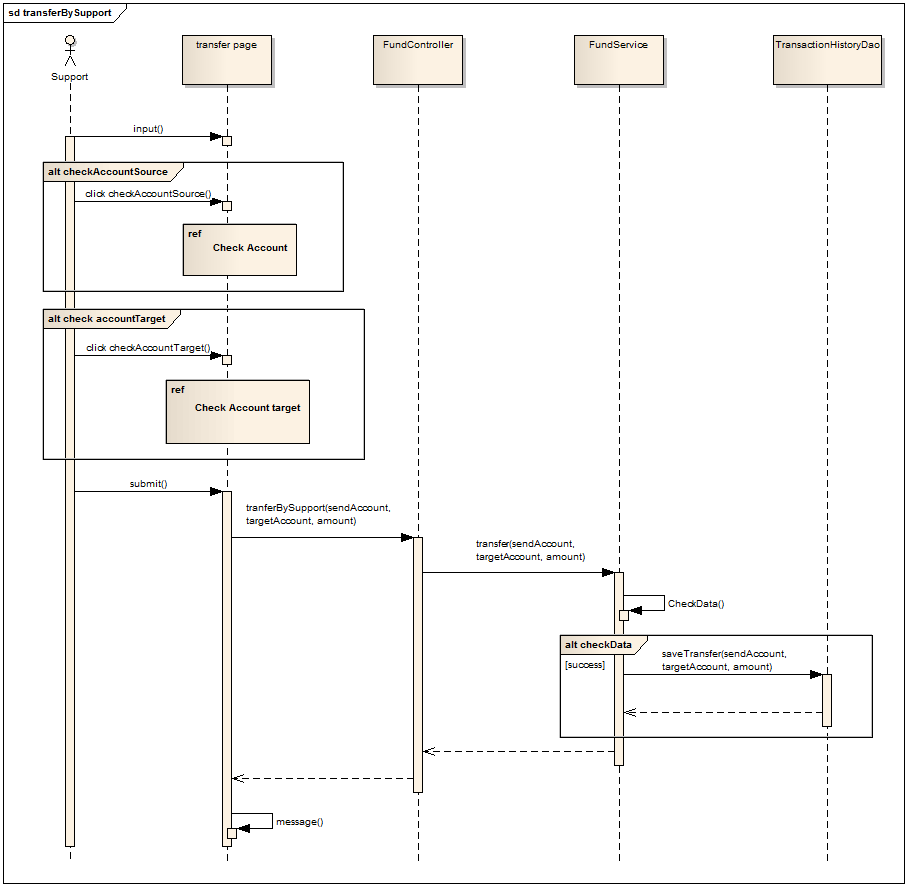
* + 1. **Check Account**



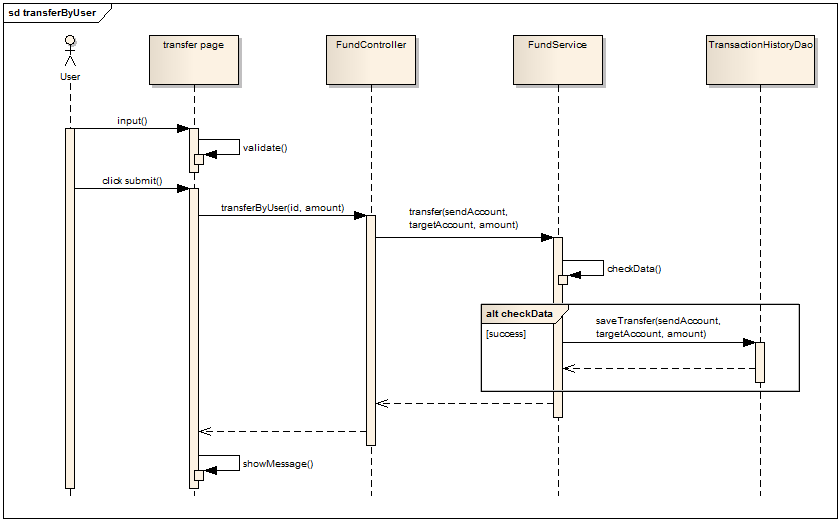
* + 1. **Ignore Transaction**



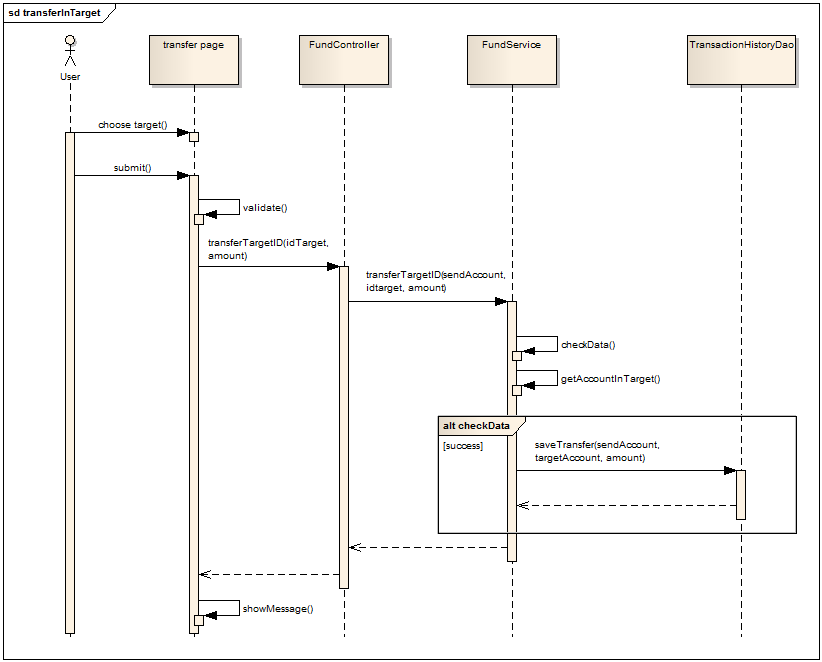
* + 1. **Tranfer by support**



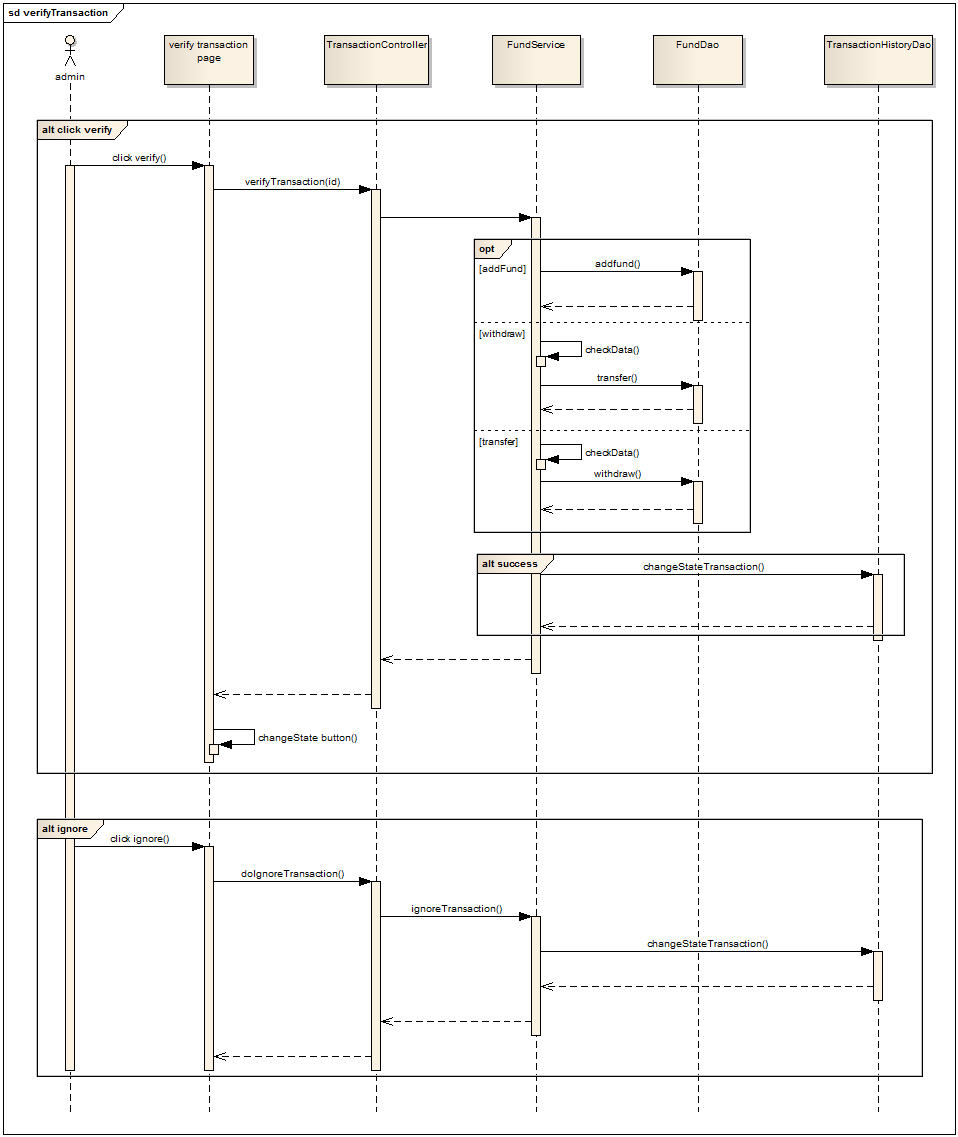
* + 1. **Tranfer by user**



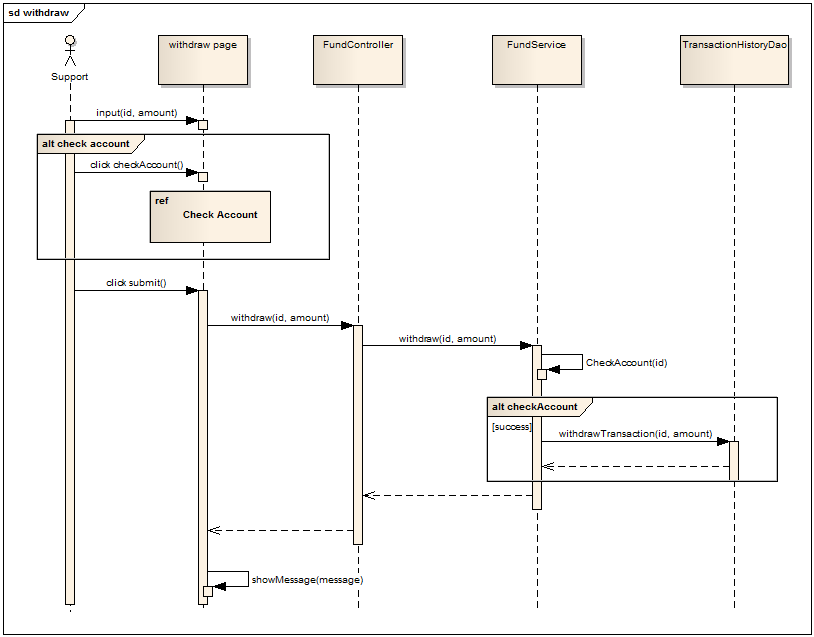
* + 1. **Tranfer in target**



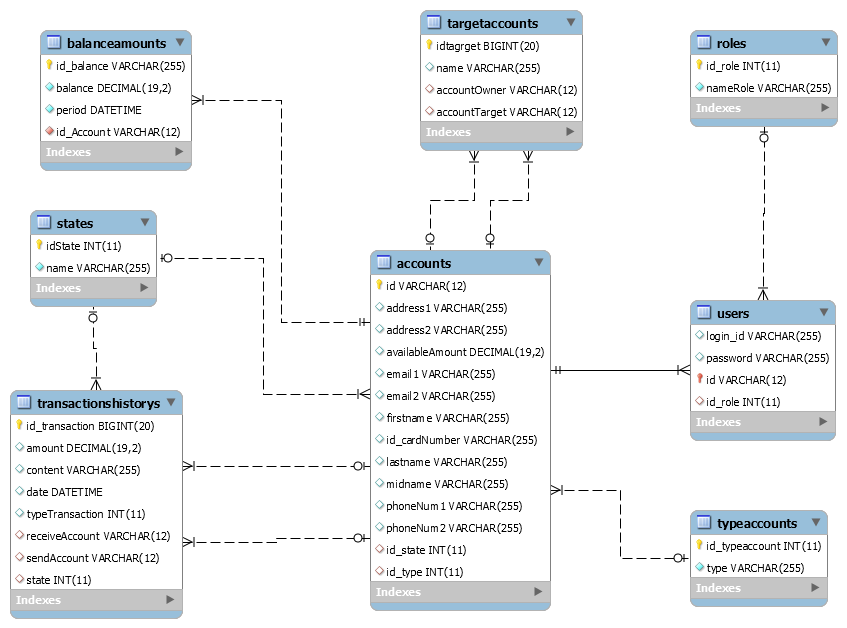
* + 1. **Verify Transaction**



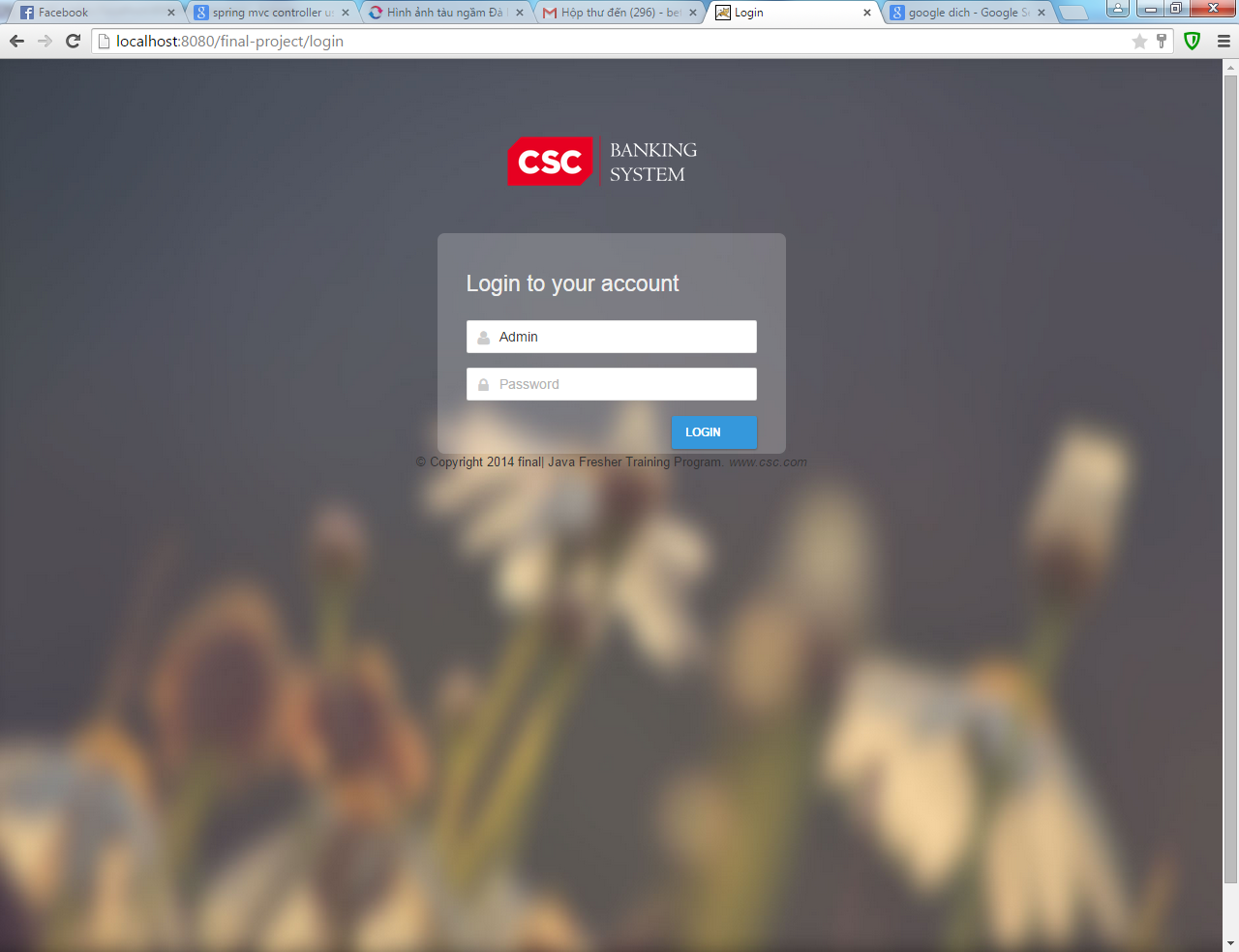
* + 1. **Withdraw**



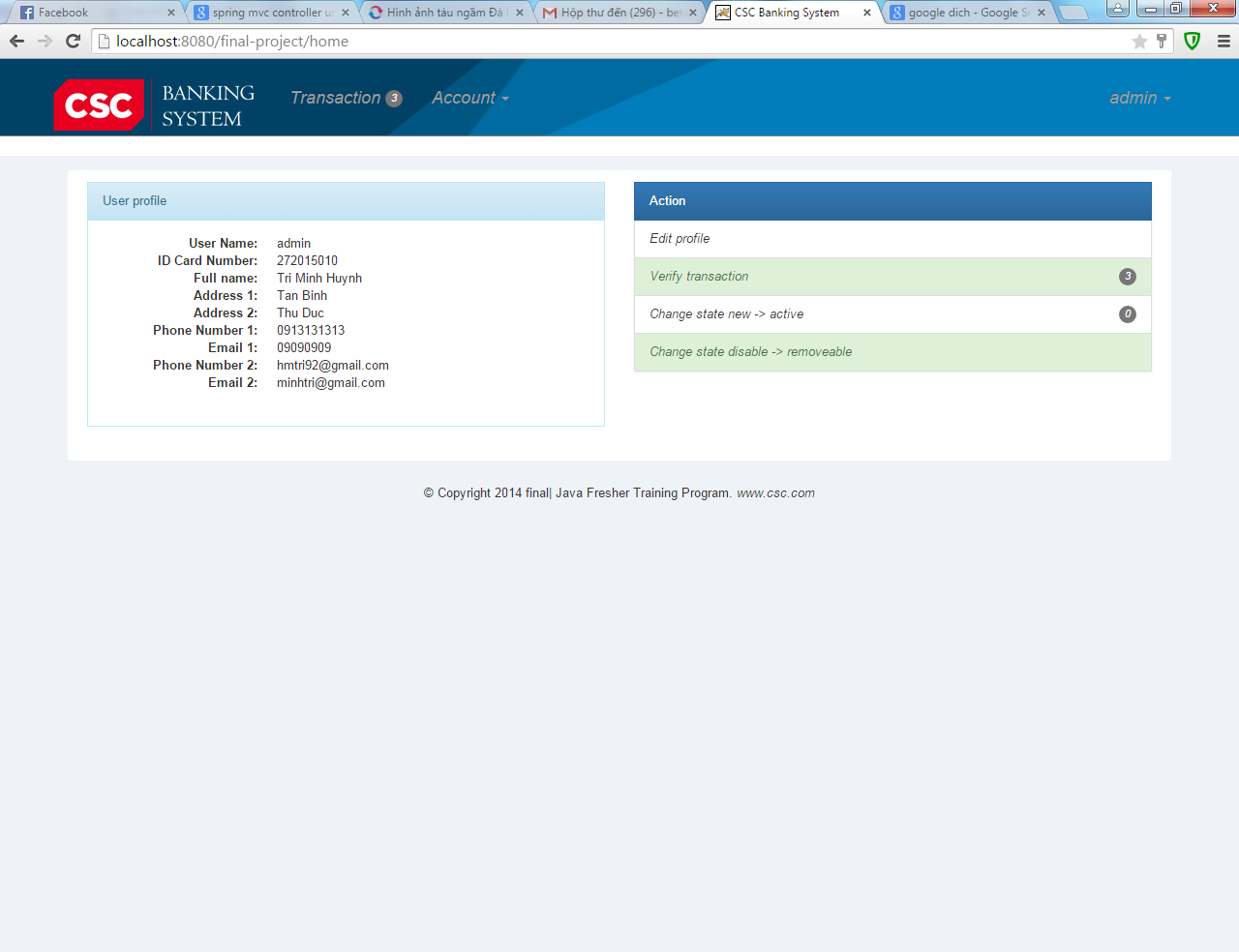
* 1. **DataBase**

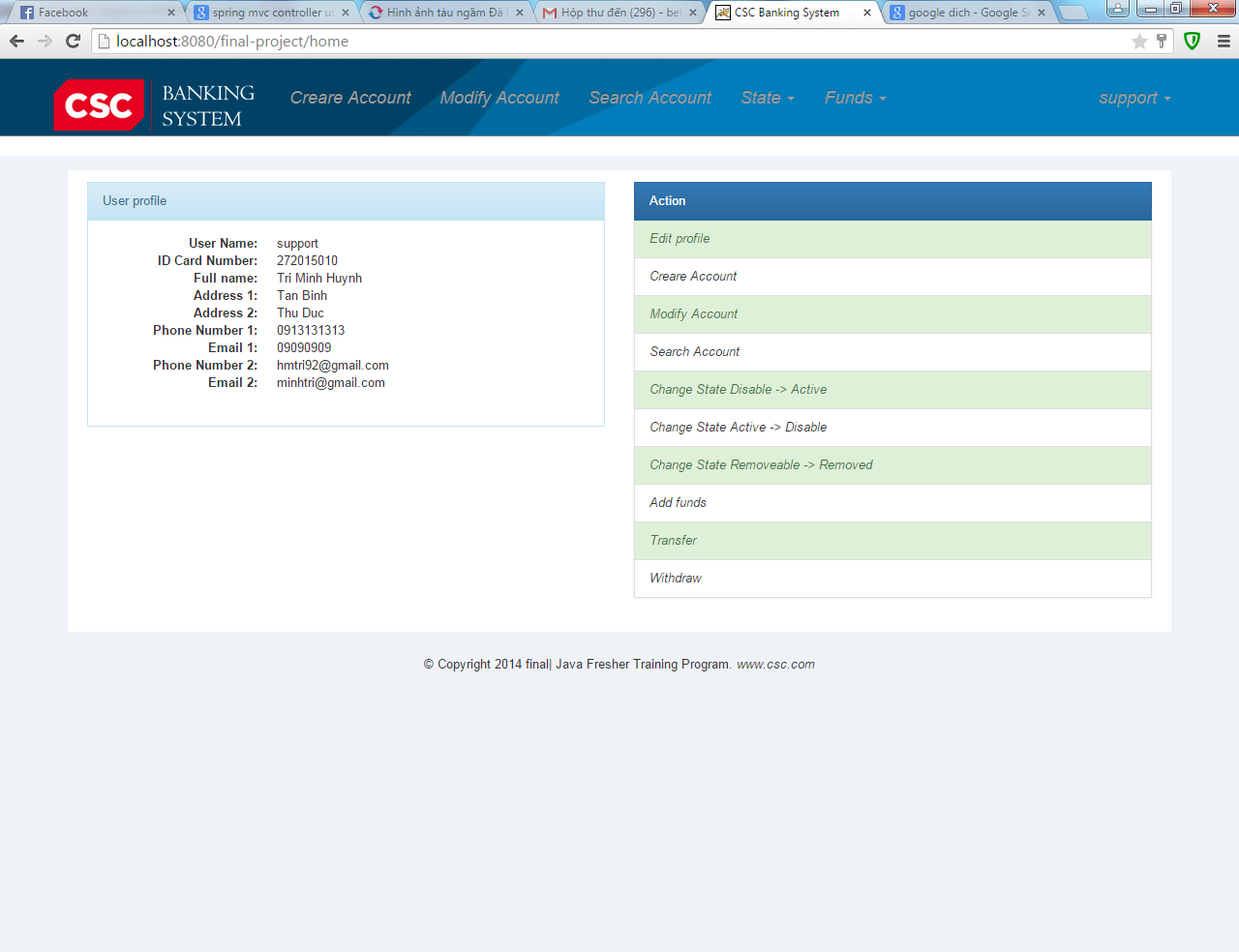


* 1. **Interface Design.**
     1. **Login Page**



* + 1. **Page Admin.**



* + 1. **Page Support.**
    2. **Page Customer.**

