

# Design Model Document: Bird Sung Gulley

---



Bird Sung Gulley (鸟鸣涧)

- Group Number: 11
- Group Members (Ascending by student number) :
  - 2051278 Fengming Tong
  - 2052635 Ze Sun
  - 2052637 Wenchao Liu
- Instructor: Yan Liu

## Design Model Document: Bird Sung Gulley

1. Introduction
  - 1.1 Purpose
  - 1.2 Project scope
  - 1.3 Glossary
  - 1.4 Progress on System Design
  - 1.5 Description of Implementation Platforms and Frameworks
  - 1.6 Architectural Styles
2. Architectural Refinement
3. Design Mechanism
  - 3.1 Data Persistence Mechanism
  - 3.2 Security Mechanism
    - 3.2.1 Sa-Token
    - 3.2.2 Secure Transmission Protocol
4. Design of Subsystems and Interfaces
  - 4.1 View layer
  - 4.2 Control layer
  - 4.3 Model layer
  - 4.4 Third-party interface Specification
    - 4.4.1 View the Artwork API Specification
  - 4.5 Query User Information API Specification
  - 4.6 Subsystem Interface Specification
    - 4.6.1 Painting System Management
    - 4.6.2 Info Uploading Management

4.6.3 Affair Management
4.7 Use Case Realization
4.7.1 Forum interaction
4.7.2 Painting Upload
5. Progress on Prototyping
5.1 Front-end Prototyping
5.2 Back-end Prototyping
6. Project Self-Reflection
Wenchao Liu
Tong Fengming
Ze Sun
7. Participation
8. Acknowledgement

# 1. Introduction

---

## 1.1 Purpose

The purpose of this System Design Model Document (SDM document) is to present detailed design model of the our project Bird Sung Gulley. This document will cover the platform-independent architecture and related subsystems with specific interfaces. Some mechanisms and samples of system prototyping will be given as well.

## 1.2 Project scope

The Bird Sung Gulley is a web-based club management system, our aim is to build a so called "digital museum" or "e-gallery", which will be able to offer both amateurs and pros an easy-to use platform to communicate and exchange their ideas freely. The users can access the system by PC or mobile device through a Website.

The potential scenarios of this system mainly include are as follows:

- searching for a specific work or artist;
- writing comments for a topic or a specific work;
- browsing brief introduction of traditional Chinese art's history or something else that worth a look at;
- posting your own work and interacting with others who wrote comments for your work;
- sharing your thoughts and then "lead to" a friendly debate;
- .....

Besides, this online system also has the listed features:

- running in a networked environment;
- handling the user authentication for security;
- owning a centralized database and etc.

## 1.3 Glossary

TERMS	DEFINITION
<b>Bird Sung Gulley</b>	The name of our system, quoted from Wang Wei's work (translation's not official).
<b>Amateur</b>	This title will be given to every new user of this system. By participating in all kinds of activities, users can earn more points to show their "status" in this system. For example, if your comments on some specific topics or paintings are liked by many other users, you will be able to get lots of points for being helpful to this system.
<b>Pro</b>	This identity will only be authorized to an user after he/she succeeded in passing identity authentication. For example, if you are a registered member at local painter association, or you were officially acknowledged as a painter at a certain rank.
<b>Column</b>	A collection of the same type of topic posted in the forum.
<b>Post</b>	Something you would like to discuss with others or simply you want to share your ideas on something, etc. No matter what your purpose is (as long as it is not hostile), you can write a post to express them.
<b>Activities</b>	The online activity held by the platform, which every user can take part in without any cost. Well, forget it if you value your time as a precious and non-renewable source.

TERMS	DEFINITION
<b>Publicize Activities</b>	The moderator of the column can make it a sticky post if the activity is permitted by the admin, so that more users will be able to participate in it easily.
<b>Report</b>	If any user noticed some inappropriate comments or posts, he/she can report it to the admin or moderator of this column.
<b>Profile</b>	This means basic info of any user of this system, including personal info along with posts and comments they ever made. all users can decide whether other users can browse their profile or not. However, some basic info will be required to remain public, for example, your username, sex, etc.

## 1.4 Progress on System Design

In the previous report, we initially completed the design of the system structure and analysis model; created a complete user interface; showing the relationship of classes in different subsystems through class diagrams; reflecting the timing relationship of function implementation through timing diagrams.

Based on the previous job, this time, we determined the microservice architecture of the project. We will provide specific technical architecture and logical architecture model, and design specific interface for our project, detailed description of part of the third party API, subsystem interface, analysis mechanism, use case realization and system prototype design. We focus on the specific design of the project, prepare to explain the main technology needed to implement the project, and determine how the project will be implemented.

The major refinements of our system are listed as follows:

- The specific logical architecture, development architecture and deployment architecture of the system are completely determined. What are the actual technologies that may be used are analyzed. The logical architecture is presented in a package diagram.
- Improved the interface implementation and interface specification of our subsystem, and provided the details and usage specification of the third-party API to be used.
- Data persistence mechanism, security mechanism and other design mechanisms are adopted.
  - The data persistence mechanism helps our system services to avoid direct operation on the database and improve the system efficiency and availability;
  - the security mechanism contains a series of security measures such as data transmission security to improve the system security.
- We personally practiced the construction of part of the project content and present the prototyping.

## 1.5 Description of Implementation Platforms and Frameworks

In the practical development, we will use microservice architecture.

In microservice architecture, each microservice in the system will be independently deployed, and each microservice is loosely coupled. which means each microservice focuses on just one specific task . This allows for low coupling, greater flexibility, targeted problem solving, easier independent development, and high availability and stability.

So we need a series of proven techniques and frameworks to build the system. For each sub-system and component in the whole system, the overview of practical implementation is shown as follows:

- The front-end of the system uses vue.js and mpVue frameworks depending on the web side and the applet side respectively. We use Node.js, NPM, and Webpack tools to build complete front-end pages, and use AJAX and WebSocket to build interactive pages with real-time updates to give users a better browsing experience. Meanwhile, all APIs are based on Rest style specification, data is uniformly converted into JSON format and sent to the interface layer.
- In the interface layer, to keep the system stable and efficient, we used Nginx clusters as reverse proxies to protect the raw resources in the back-end and speed up the transmission. At the same time, Spring Cloud Gateway is used as a gateway to provide filtered data, unified interfaces, and The Spring Cloud Gateway is used as a gateway to provide filtered data, unified interfaces, and triggering of the fusion mechanism to further ensure the security of the back-end services. It also provides SSO based on Web JSON Token SSO single sign-on mechanism based on Web JSON Token standard, so that all trusted systems can be accessed with one login.
- In the service layer, we used Spring Security to identify access user privileges to further enhance security, and use the Nacos, Sentinel, Dubbo, and Seata components provided by Spring Cloud Alibaba to build microservices and provide service configuration, traffic control, distributed system communication and distributed system transaction processing, respectively. The Aliyun logging service framework is capable of providing the following functions AliCloud's logging service framework enables logging of services. SkyWalking injects monitoring probes into the above services, invoking the link for monitoring and tracking and providing visual notifications, enabling developers to clearly know the system's historical operation status for better system maintenance and error location. The components in this layer and the gateway in the upper layer are deployed in containers and managed and maintained by Kubernetes.
- The data persistence layer is mainly for data storage and access, and MySQL is chosen to work with Hibernate as the database and persistence framework. MySQL has the characteristics of light weight and high performance, and with Elasticsearch can be more convenient to achieve MySQL is lightweight, high performance, and with Elasticsearch can more easily achieve read-write separation, while Hibernate is more advantageous in object maintenance and access.
- The server layer mainly uses SpringBoot as the development framework, which encapsulates some common packages and can achieve rapid development.

We will present all these with a figure in the following part.

## 1.6 Architectural Styles

Our system mainly adopts the object-oriented style in developing the whole framework of this project.

Data abstraction and object-oriented architecture. The components of data abstraction and object-oriented architecture style are objects, and objects are instances of abstract data types. In abstract data types, the representation of data and their corresponding operations are encapsulated.

The behavior of the object is reflected in its acceptance and request actions. Connectors are the way of interaction between objects. Objects interact through the calls of functions and procedures. Objects are encapsulated, and the change of one object will not affect other objects. Objects have state and operations, and are also responsible for maintaining state.

This structural style includes features such as encapsulation, interaction, polymorphism, integration, and reuse. To achieve a low-coupling, highly reusable and maintainable system that is easy to upgrade and protects internal integrity, this system uses IOC (Inversion of Control) to build containers to achieve decoupling between objects and AOP (Tangent Oriented Programming) to extract public behaviors encapsulated as common services to reduce system redundancy and coupling between modules and improve system operability and maintainability.

## 2. Architectural Refinement

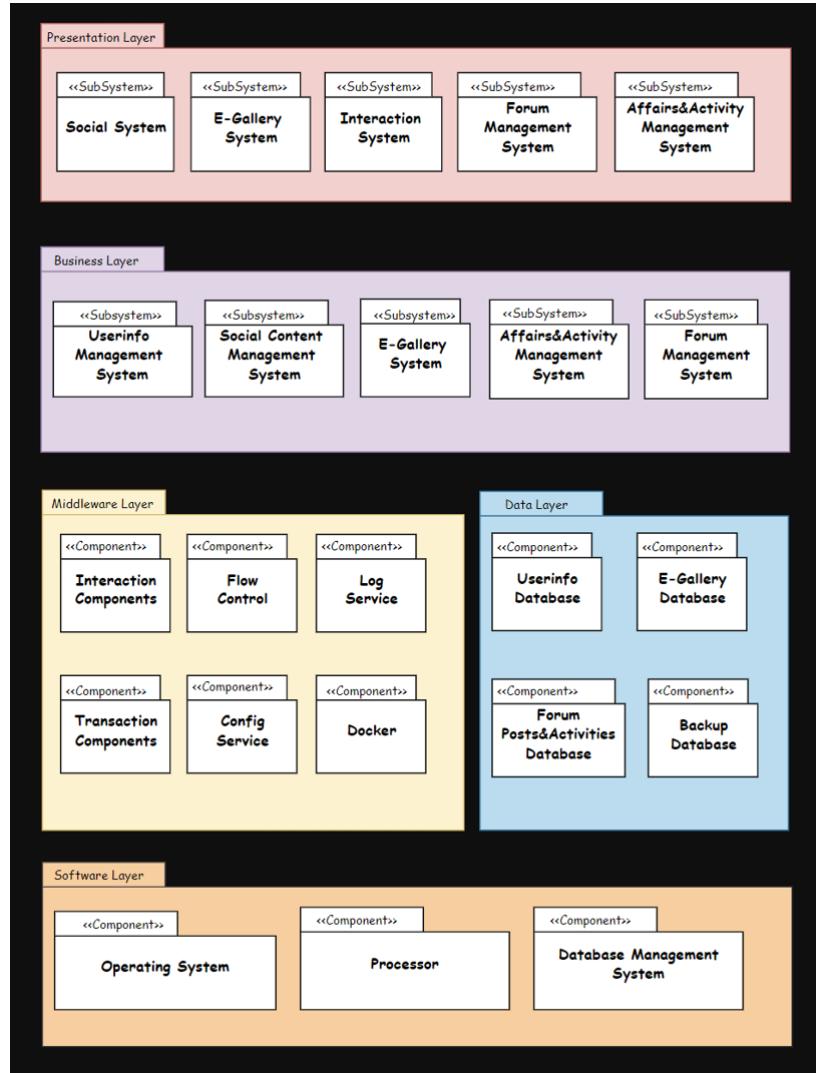
---

We mainly focused on the integrity of the system's internal functions and the security of data transmission, and combined the design of the actual technical and physical models, we chose to use a microservice architecture system based on domain-driven design to complete the improvement and upgrade of the system's logical architecture.

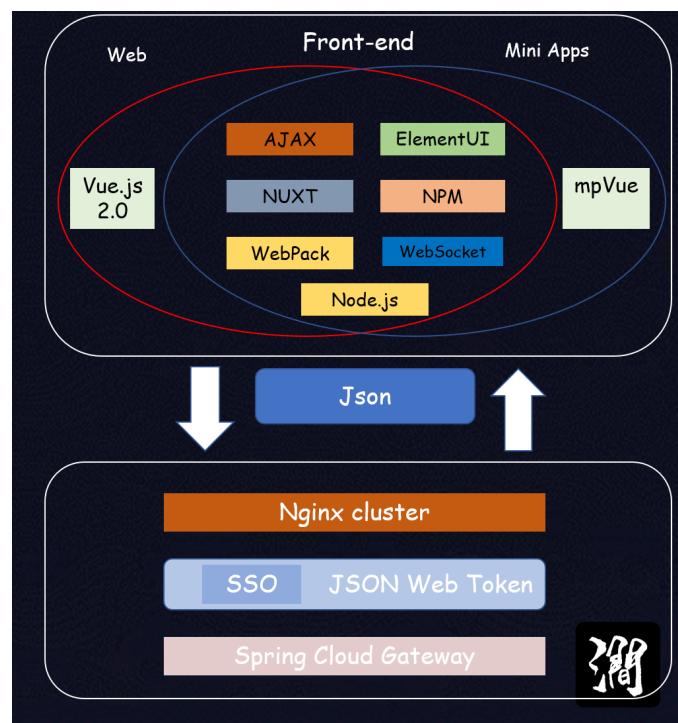
The reasons for choosing the transition from layered architecture to microservice architecture in this project are:

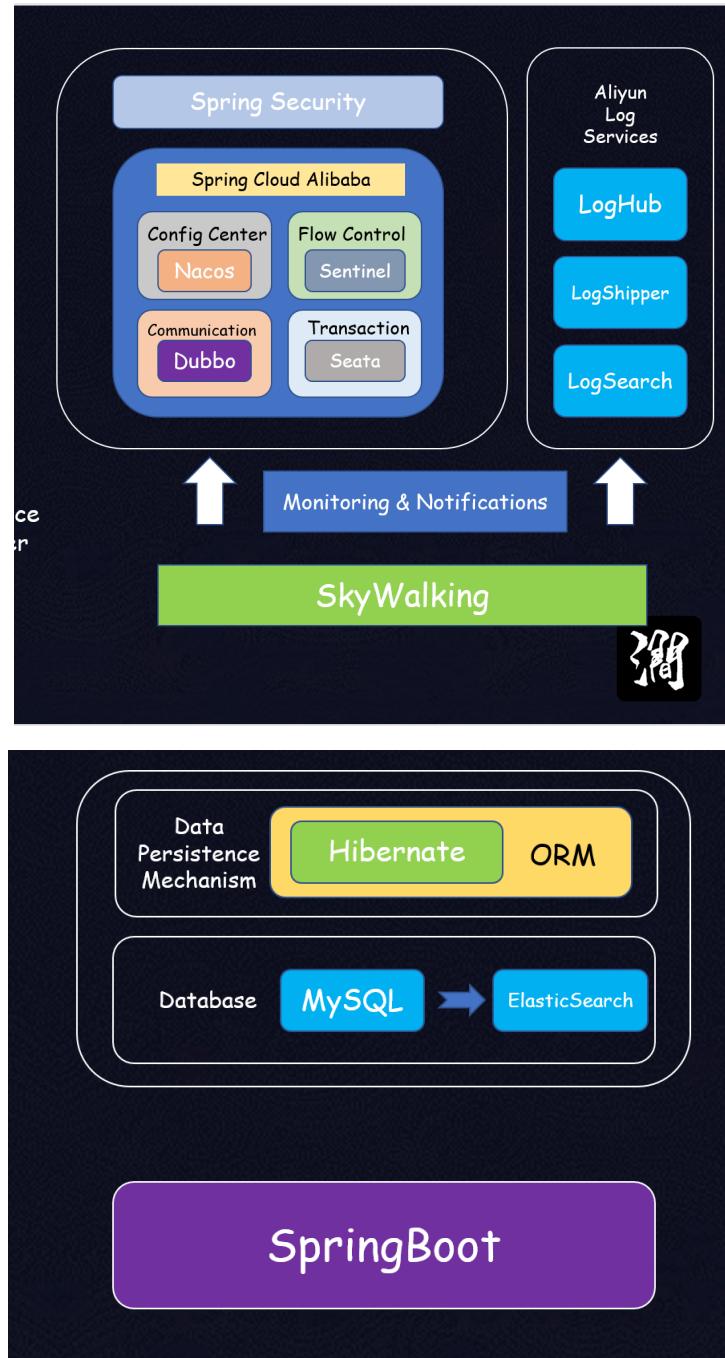
- The services provided are relatively complete and independent, and can be packaged into different microservice clusters for other systems to use.
- Microservice development is flexible, fast iterative and supports rapid updates.
- Different services are deployed independently, with high cohesion in services and loose coupling between service.

Based on the above characteristics of microservice architecture, and after weighing the pros and cons of microservice architecture and traditional layered architecture, in order to design a system architecture with complete internal functionality, high security and light weight, we choose to generalize and refine the previous architecture between layers appropriately.



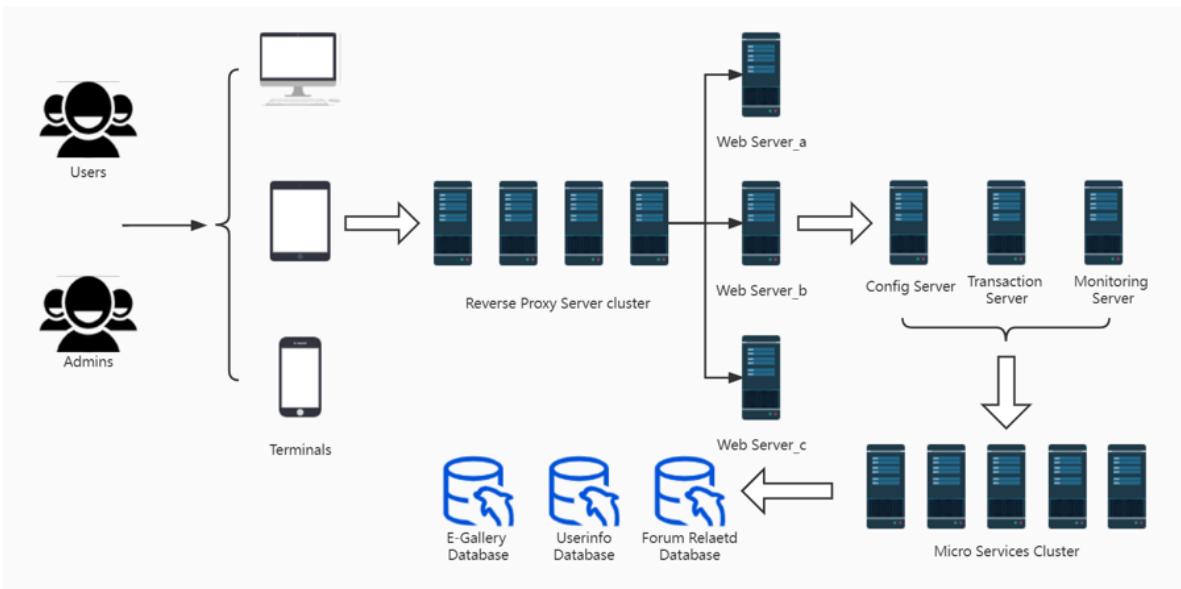
This figure mainly showed what we've talked about in chapter 1.5, which is the developing architecture figure.





Apart from deciding on what develop routine we shall follow and how should we organize all these microservices together, the implementation and distribution of our system shall also be thought of.

This figure below demonstrated how we planned to organize our server and offer services to users.



## 3.Design Mechanism

### 3.1 Data Persistence Mechanism

A lot of data information will be stored in our system, which need to be preserved permanently for the realization of system functions. As an object-oriented programming language, Java objects can only be saved in memory. Therefore, special databases need to be used for data storage, which is called "Data Persistence".

According to the requirements of our system, the relational databases are selected for data storage. Relational data and objects are two forms of business entities. Business entities are represented as objects in memory and relational data in databases. There are many association and inheritance relationships among objects in memory, which cannot be expressed by relational data.

Therefore, Object Relational Mapping (ORM) is needed for data persistence. In order to achieve the goal, we implement a persistence layer between Business Layer and Database. Persistence layer is a kind of middleware, which can automatically store objects in databases, as well as load data from databases to memory. Under such an abstraction layer, the programs will indirectly access databases via session instead.

Hibernate is a high-performance object-relational persistence storage and query service.

Hibernate not only focuses on mapping from Java classes to database tables, but also Java data types to SQL data types, in addition to providing data query and retrieval services. It also serves as a bridge between the database and the interface, and requires object-oriented thinking to manipulate objects.



Hibernate directly provides related support, and the underlying driver can switch databases at will, fast and simple. Separate the business layer from the specific database, and develop only for Hibernate to complete the persistence of data and objects. Data persistence has several advantages:

- High reusability of source code, even if you change the database, you only need to change the configuration file and do not have to rewrite the source code.
- The business logic code is readable, and there won't be a lot of SQL language in the code, which improves the readability of the program.
- Persistence technology can be automatically optimized to reduce the amount of access to the database and improve the efficiency of program operation.

In the following part, we are going to introduce the 6 key APIs of Hibernate.

- Configuration Object

The Configuration object is the first Hibernate object created in any Hibernate application. It is usually created only once during application initialization, which represents a configuration or properties file required by the Hibernate.

- SessionFactory Object

Created by Configuration object, SessionFactory object configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated. The SessionFactory is a thread safe object and used by all threads of an application.

- Session Object

A Session is used to get a physical connection with a database, it provides an interface between the application and data stored in the database. The Session object is lightweight, short-lived and wraps the JDBC connection. Persistent objects are saved and retrieved through a Session object. The Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.

- Transaction Object

A Transaction represents a unit of work with the database. Transactions in Hibernate are handled by an underlying transaction manager and transaction (from JDBC or JTA). Transaction is optional, Hibernate applications may manage transactions in their own application code instead.

- Query Object

Query objects use SQL or Hibernate Query Language (HQL) string to retrieve data from the database and create objects. A Query instance is used to bind query parameters, limit the number of results returned by the query, and finally to execute the query.

- Criteria Objects

Criteria objects are used to create and execute object oriented criteria queries to retrieve objects. Criteria API is designed to build up criteria query objects programmatically where you can apply filtration rules and logical conditions.

Between our business application and the database, the Hibernate framework uses Session sessions to accomplish data delivery, update, delete, query, etc.

When we need to commit data to the database, such as adding registered users, we save the data to the Session session, and then decide automatically or manually when to commit that save to the database, depending on the framework's configuration file.

When we need to query data from the database, we need to clear the cache (either manually, or automatically through the configuration file framework) so that the data in the Session session is consistent with the data in the database. This way, the program only needs to query the data in the Session.

In summary, data persistence is a solution to the problem of interactive access between the program and the database, so that the program does not access the database directly, but accesses the Session session directly, as long as it ensures that the program is consistent with the data in the database every time it accesses the data in the session, no errors will occur.

## 3.2 Security Mechanism

### 3.2.1 Sa-Token

Sa-Token is chosen as the authorization and authentication framework, which is a lightweight Java authority authentication framework.

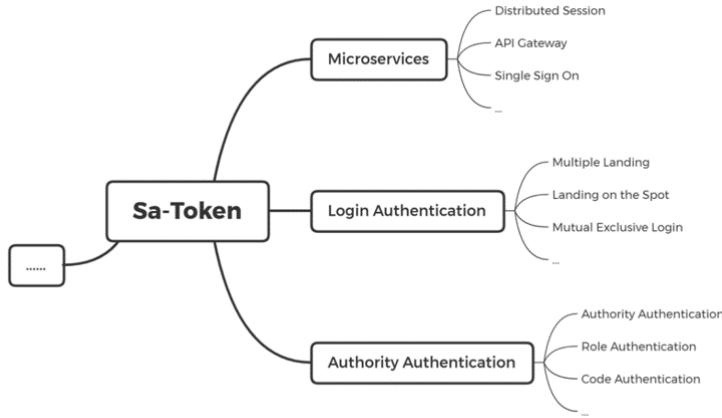
Compared with other certification frameworks, Sa-Token has the following advantages:

- Lightweight

Sa-Token is a lightweight and zero configuration start-up framework, while Spring Security mainly provides security access control solutions for enterprise application systems therefore is difficult to learn.

- Separation of front and rear platforms

The design concept of separating front and back of traditional frameworks such as Apache Shiro, Spring Security lags behind, which makes them difficult to fit with the project; while Sa-Token is designed for the separation of front and back office architecture, which is more suitable for the system.



### 3.2.2 Secure Transmission Protocol

- SSL protocol (Secure Socket Layer)

Another important point of web security for the user is the security of the data exchanged between the user and the server. For general web pages, these data are transmitted in plaintext, meaning they are visible to anyone, which is sufficient for most use cases.

However, for private data, such as passwords, bank account numbers, etc., there is a risk if the data is transmitted in plaintext. HTTPS is a secure data transfer protocol that uses SSL (Secure Socket Layer) technology on top of the HTTP protocol to encrypt the transmitted data, thus ensuring the security of the data. The SSL protocol is built on top of the TCP protocol, under the application layer protocol HTTP, and the main process of SSL is server authentication, followed by user authentication.

- TLS protocol (Transport Layer Security)

TLS (Transport Layer Security) is based on SSL 3.0, which uses a new encryption algorithm and is therefore not compatible with HTTPS.

TLS is used between two communicating applications to provide confidentiality and data integrity. The protocol is composed of two sub-protocol layers, including TLS Record and TLS Handshake, with the lower layer being TLS Record, the higher layer being TCP protocol.

Three major properties of TLS are as follows:

- Asymmetric cryptography can be used to authenticate the identity of the peers.
- Negotiation of shared encryption keys is secure, negotiated encryption is difficult for a thief to obtain, and in addition an authenticated connection cannot obtain encryption, even by an attacker entering the middle of the connection.
- The negotiation is reliable and no attacker can modify the communication negotiation without being detected by the members of the communicating party.

## 4. Design of Subsystems and Interfaces

---



## 4.1 View layer

上传画作系统

接口类：PaintingUpload System

接口函数：

- 1.sendUpload():void //上传画作
- 2.updatePaintingdatabase():void //更新画作数据库

社交系统

接口类：Social System

接口函数：

- 1.editMessage():void
- 2.editComment():void
- 3.leaveComment(Content new commnet):bool  
//通过审核成功发布后返回true

事物管理系统

接口类：AffairManagement System

接口函数：1.submitLogFile(file):void

- 2.editNotification():void
- 3.authorize(string UID):void//授予用户管理员权限

画作信息系统

接口类：IPaintingsInfoSystem

接口函数：

- 1.displayPaintingsInfo(string paintingType):PaintingInfo list
- 2.sharePainting():void

## 4.2 Control layer

个人信息管理系统

接口类：IUserInfoSystem

接口函数：

- 1.getUserInfo(string UID):userInfo

2.updateUserInfo(string UID,UserInfo newUserInfo):void  
3.getFilteredPersonalInfo(FiteredCondition condition):PersonInfo List

#### 内容管理系统

接口类： ISocialContentManagetSystem

接口函数：

- 1.auditContent(Content newContent):bool
- 2.editMessage():void
- 3.editComment():void

#### 平台系统

接口类： Platform Management System

接口函数：

- 1.organizeActivity():void
- 2.reportInappropriateContent():void
- 3.deleteInappropriateContent():void

#### 专栏管理系统

接口类：

Column Management System

接口函数：

- 1.createColumn():void
- 2.authorizeModerator(UID):void//授权成为专栏版主

## 4.3 Model layer

#### 用户信息数据库

接口类： UserInfoDatabaseSystem

接口函数：

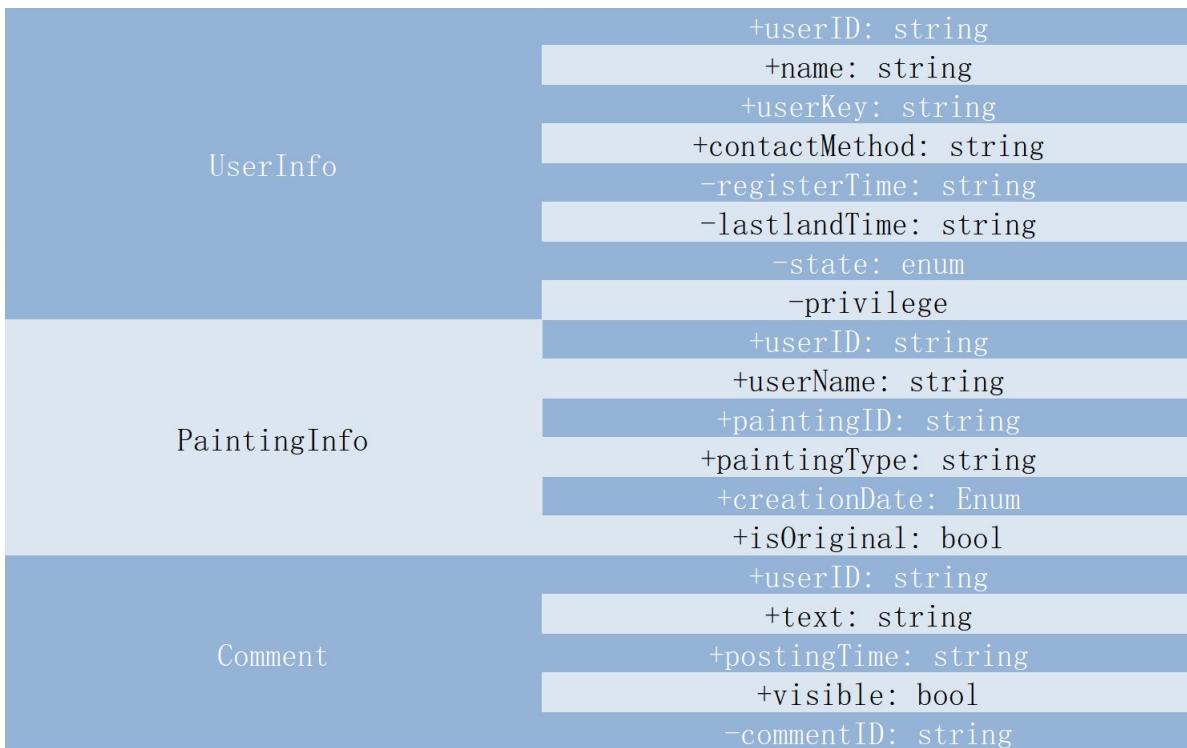
- 1.queryUserInfo(string UID):UserInfo
- 2.updateUserInfo(string UID,UserInfo newUserInfo):void
- 3.queryUserreputation(string UID):Repu
- 4.updateUserreputation(string UID,Repu newRepu):void

#### 画作信息数据库

接口类： paintingInfoDatabaseSystem

接口函数：

- 1.queryPaintingInfo(string VID):void
- 2.storePaintingInfo(PaintingStoration PaintingInfo):void
- 3.updatePaintingInfo(PaintingStoration PaintingInfo):void



## 4.4 Third-party interface Specification

### 4.4.1 View the Artwork API Specification

When a user calls the painting viewing system and enters the key information of the painting they want to query, the interface provided by the third-party website will be called, and the data will be processed at the same time. In the design of this system, we decided to use the interface provided on the aggregated data. According to the interface documentation, we need to provide the following parameters

Request Method: GET

Request Parameters:

参数名称	类型	默认值	示例值	必要性	描述
pic	string		pic	是	固定值pic; 代表数据为画作
picurl	string		<a href="http://test.nmj.com/1a8b1.pg3">http://test.nmj.com/1a8b1.pg3</a>	是	图片的网络地址
user_id	string		10272	是	请求查看用户id
token	string			是	用于验证当前请求发起者的登录状态

After sending the request, the interface will return a JSON file that contains the painting information we need. The return parameters are as follows:

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为上传成功，其他值为失败错误代码
msg	string		"pic不存在"	否	失败的原因，code不为0时，才有此字段
pic_id	string		20345	是	图片的id
pic_info	string		作者：xx 创作时间:xx ...	是	图片的基本信息
add_time	string		2022-6-1	是	上传时间
isDeleted	bool	false	false	是	图片是否被删除
altText	string	图片无法显示	图片不存在	否	在图片无法正常加载时显示的替换文字
altPic_url	string	<a href="http://xxxxxx.pg3">http://xx xxxxx.pg3</a>	<a href="http://test.nmj.com/1a8b1.pg3">http://test.nmj.com/1a8b1.pg3</a>	否	在图片无法正常加载时显示的替换图片

## 4.5 Query User Information API Specification

When the user calls the Query User Information system and enters the account password of his own account, this interface will be called and the data will be processed at the same time. According to the interface documentation, we need to provide the following parameters

Request Method: GET

Request Parameters:

参数名称	类型	默认值	示例值	必要性	描述
act	string	query	query	是	固定值: query; 代表查询请求
user_id	string		10272	是	用户id
user_key	string		12345abc	是	用户的账号密码

The return parameters are as follows:

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为提交成功，其他值为失败错误代码
msg	string		"用户信息查询失败"	否	失败的原因，code不为0时，才有此字段
user_id	string		10272	是	用户id
user_key	string		12345abc	是	用户的账号密码
name	string		test123	是	用户昵称
contactMethod	string		<a href="mailto:123@126.com">123@126.com</a>	否	绑定的联系方式
registerTime	string		2022-06-04 22:08:10	是	注册时间
lastlandtime	string		2022-06-04 22:08:10	是	最近登录时间
state	enum	1	1	是	账号状态；1：正常；2：限制；3：封禁
privilege	bool	0	0	是	账号权限；0：普通用户；1：版主

Submit Application API Specification

Request Method: GET

Request Parameters:

参数名称	类型	默认值	示例值	必要性	描述
act	string	submit	submit	是	固定值submit；代表提交申请操作
type	enum		1	是	申请的类型；1：创建专栏；2：举办活动；3：人员变动
submission_info	string		本申请简要信息为...	是	申请的简介
fileitem	File		activity.docx	是	项目企划文件

uploader_id	string		10272	必 要 性	申请者id
参数名称 token	类型 string	默认值	示例值	描述	用于用户登录状态的验证

The return parameters are as follows:

参数 名称	类型	默认 值	示例值	必要 性	描述
code	int		0	是	0为上传成功，其他值为失败错误代码
msg	string		"pic_address不存在"	否	失败的原因，code不为0时，才有此字段

## 4.6 Subsystem Interface Specification

画作系统API

接口类： IPaintingInfo

接口函数： getPaintingInfo(string list tid):Json

社交系统API

接口类： ISocialSystem

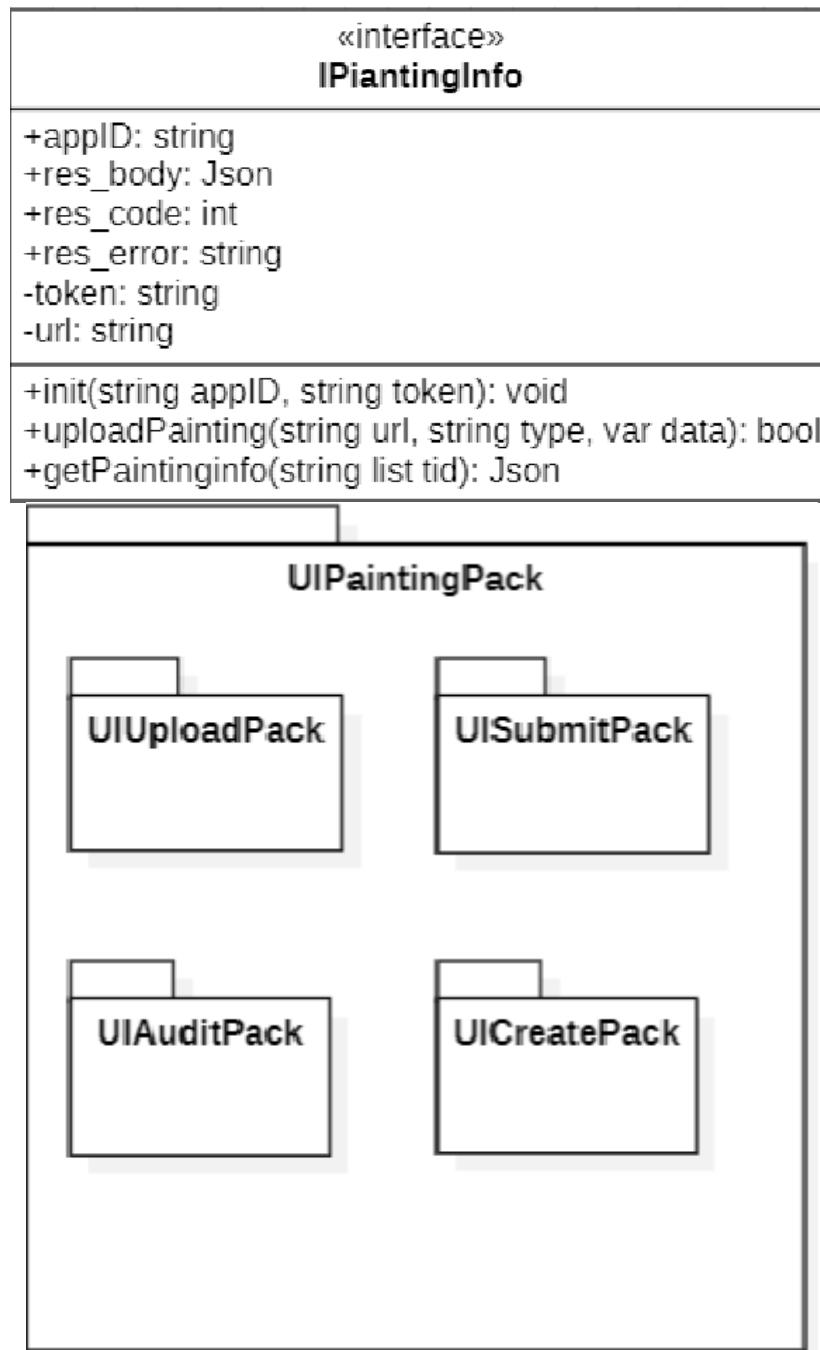
接口函数： leaveComment(in Comment comment, in ColumnRegion col): bool

事物管理API

接口类： ISocialSystem

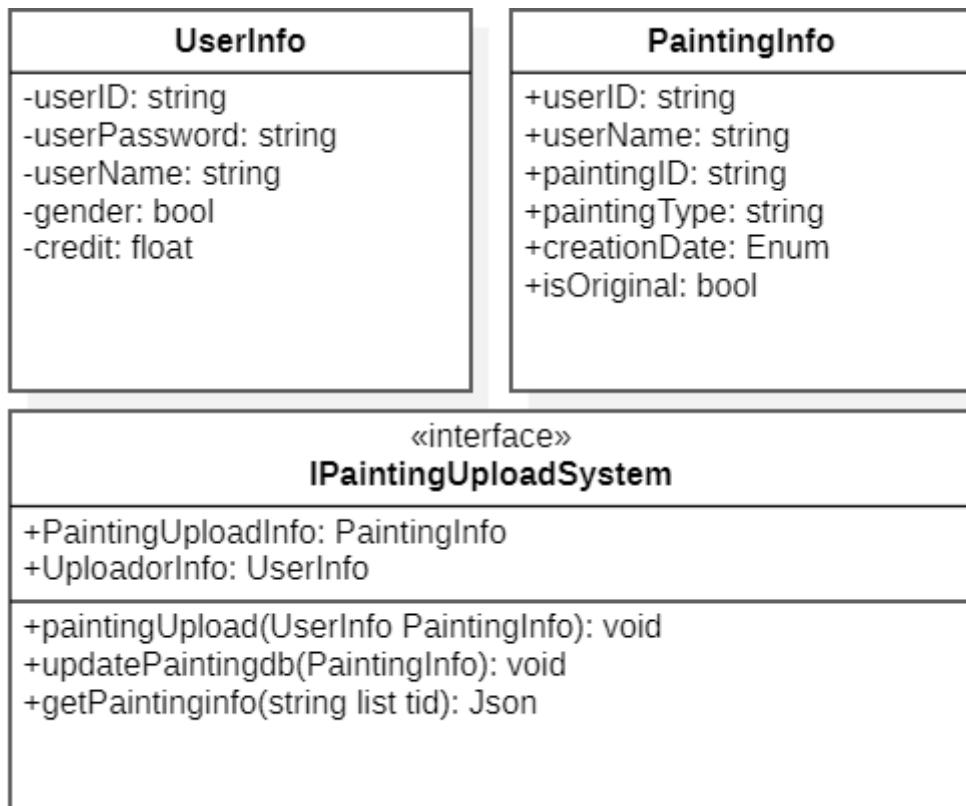
接口函数： leaveComment(in Comment comment, in ColumnRegion col): bool

### 4.6.1 Painting System Management



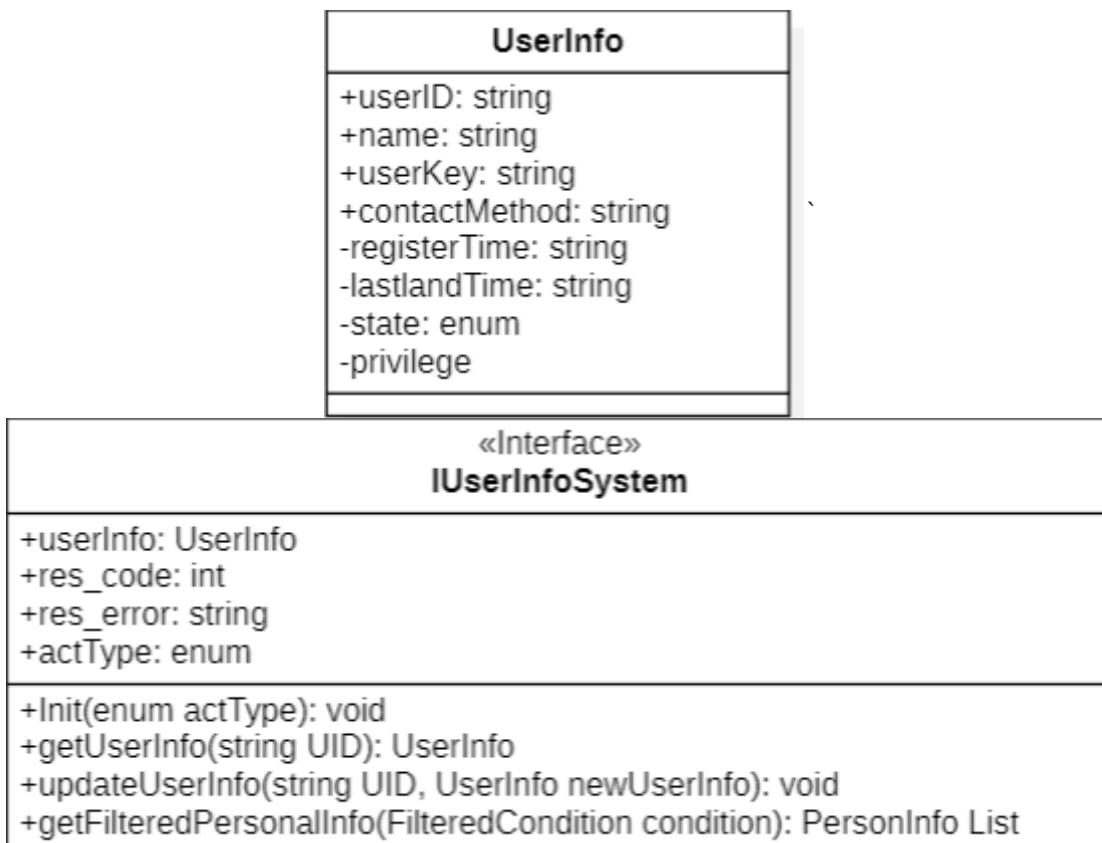
Next, we will show several interfaces for information management. The first one is **IPaintingInfo**, which contains data classes such as `uploadPainting` and `getPaintinginfo`. The interface functions include initialization, query, and deletion.

#### 4.6.2 Info Uploading Management

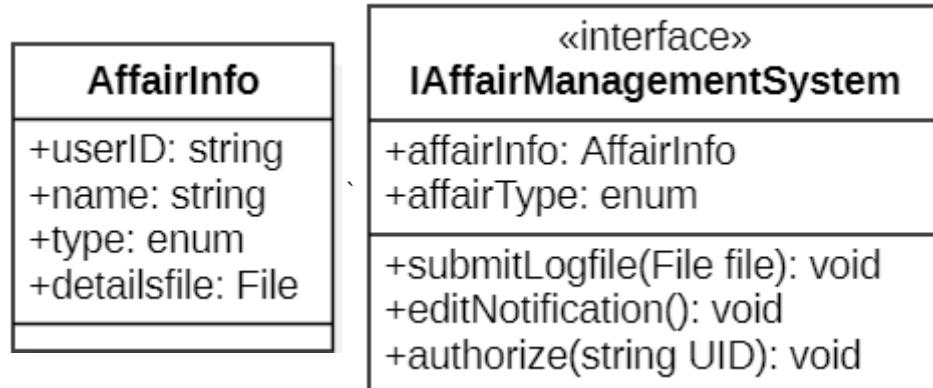


This is IPaintingUploadSystem, including UserInfo and PaintingInfo data classes, paintingUpload data class, updatePaintingdb and getPaintinginfo interface functions, which are called by the view layer painting upload system.

It should be noted that in the data class PaintingInfo, isOriginal is a bool variable, which means that the uploaded paintings are all original by the uploader. If it is true, it means it is original, and if it is false, it means it is not original.



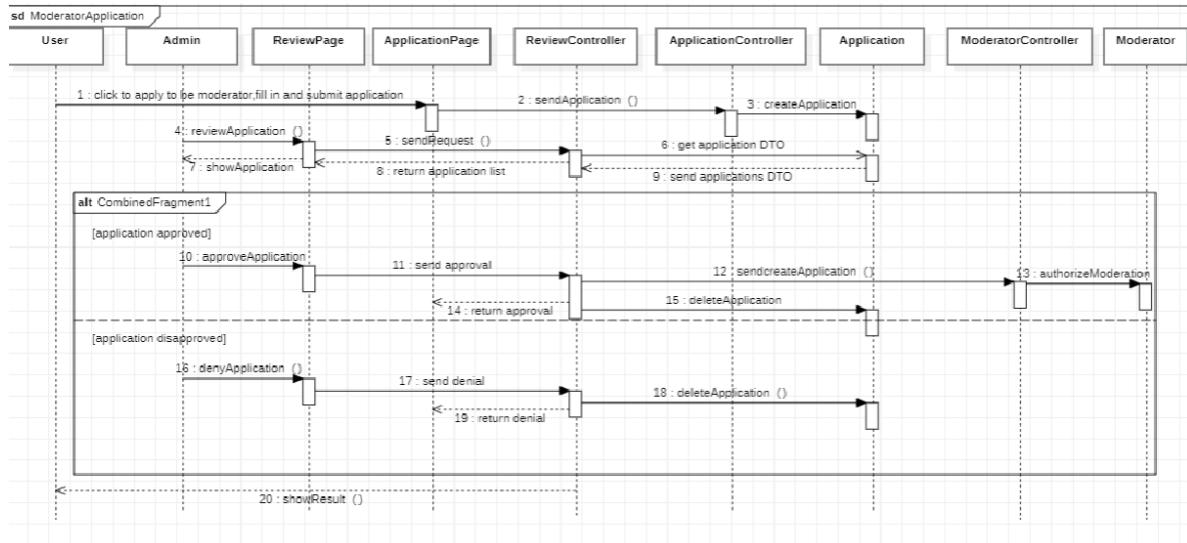
### 4.6.3 Affair Management



The above figure shows IAffairManagementSystem, which contains AffairInfo, and the interface functions include submitLogFile and editNotification. authorize() is to authorize the user to become a moderator or administrator.

## 4.7 Use Case Realization

### 4.7.1 Forum interaction



Our system provides an application function for some affair requests. Now select the use case where the moderator is requested for detailed description. The process of the use case is demonstrated in the above sequence diagram.

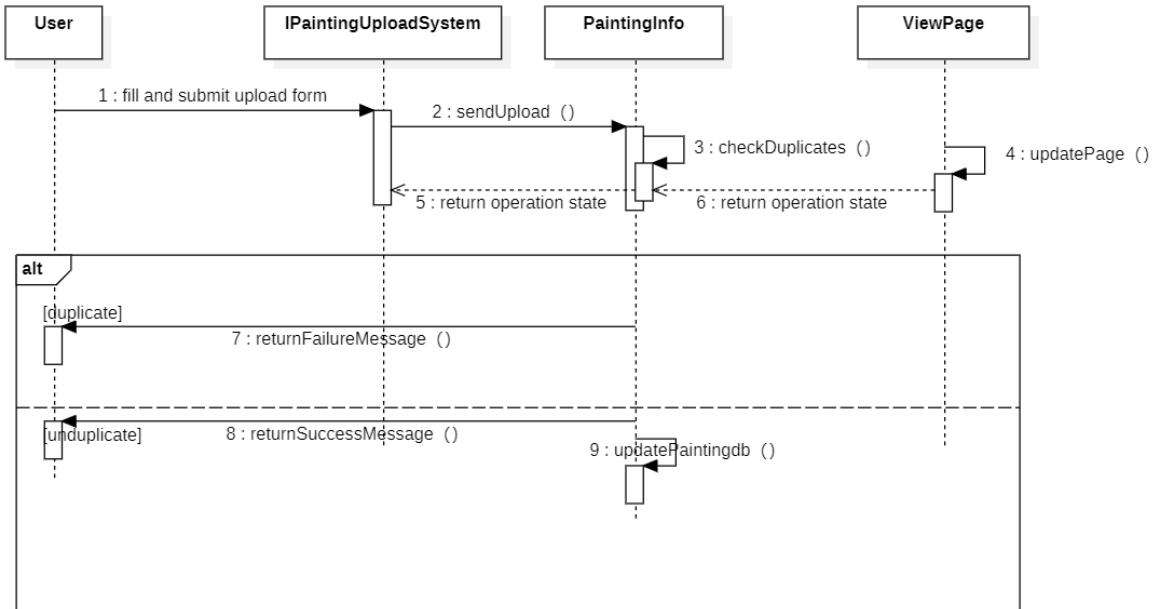
According to our system design, we use AJAX for data interaction. The Spring MVC framework, as is used in our previous use case realization, will be used here for handling request as well as validation process. The approved contents will be submitted to the database through REST API.

Firstly, the user types contents he wants to submit. After clicking the submit button, the corresponding HTML form data is generated, which is converted by AJAX techniques into the Java object. The corresponding request of this object is sent to Spring MVC.

For validation process, Spring MVC will examine the correctness of the content format. Check the database for duplicates after uploading. If the request is judged to be invalid, then it will be rejected.

After passing the validation, the content to be submitted will be posted (POST command powered by REST API) to the content database, which will return the result in JSON format, showing that the content is successfully loaded to the database. (If the operation goes wrong, an empty JSON will be returned and caught by error processing methods) Finally, the system refreshes the webpage to display the forum with the new content.

#### 4.7.2 Painting Upload



Our software relies heavily on image data. Therefore, we have perfected the process of use cases for image uploading, so that the processing of image data can be traced. The specific process is as follows:

- Live image preview

Use [html5] FileReader Object. FileReader is primarily used to read file contents into memory, and through a series of asynchronous interfaces, local files can be accessed in the main thread. FileReader provides four different ways to read files, such as `readAsArrayBuffer` reading the contents of the file as an `ArrayBuffer` object, and `readAsBinaryString` reading the file as a binary string. We use `readAsDataURL` method to encode the contents of the file to base64 and output. And store these contents directly in urls to optimize the loading speed and execution efficiency of the website.

the related js script is as follows:

```

1  function getFileContent(){
2    // file[0]就是上传的图片本身
3    // FileReader的readAsDataURL方法可以将图片转换为base64格式
4    // 1.上传单个文件时：
5      var reader = new FileReader();
6      var file = document.querySelector("#myFile").files;
7      reader.readAsDataURL(file[0]);
8      // 一定要在文件读取 成功完成时 再进行相应的操作：
9      reader.onload = function(){
10        document.querySelector("img").src = reader.result;
11      }
12  }

```

The file contains the name, size, and type of file uploaded by the user, and the size and type of the image can be limited by adding the judgment conditions.

- Upload the image to the server

Use [html5]FormData object to submit the form asynchronously with ajax  
(asynchronous file submission is also possible)

```

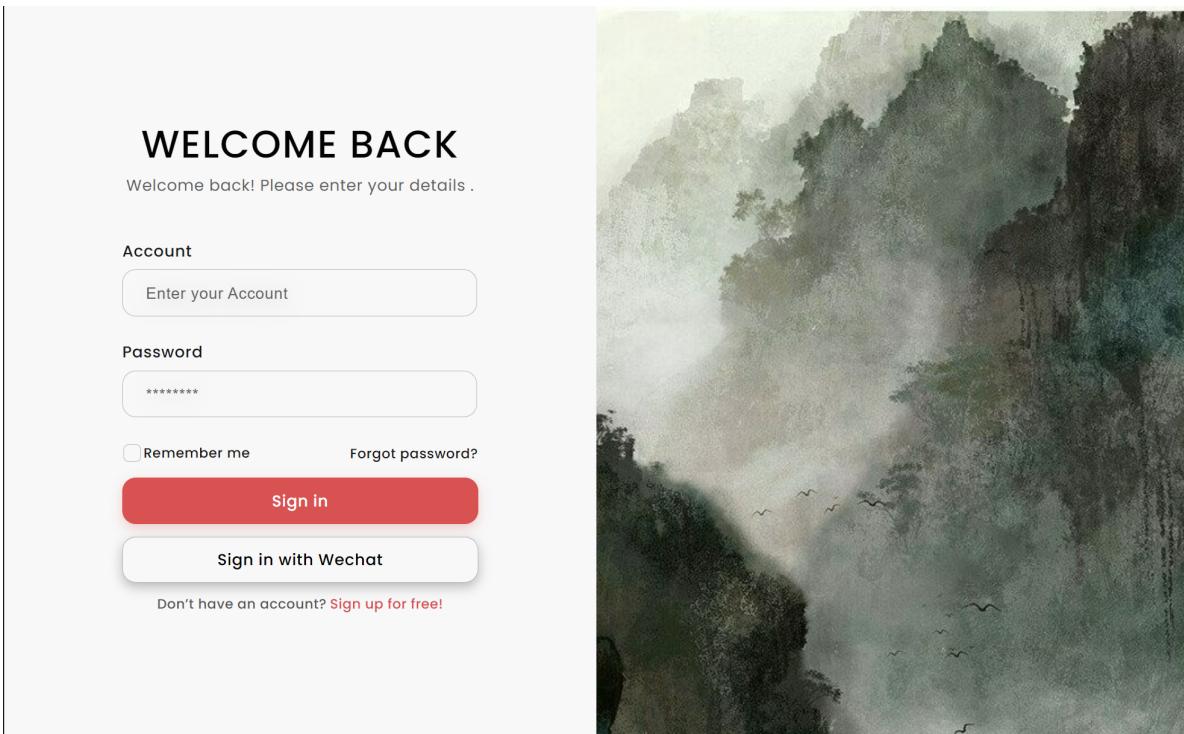
1 var formData = new FormData($('#form1'));
2 // formData.append('images', $('#myFile')[0].files);
3 //ajax请求
4 $.ajax({
5   type: "post",
6   url: "接口地址",
7   data: formdata,
8   dataType: 'json',
9   processData: false, // 告诉jQuery不要去处理发送的数据
10  contentType: false, // 告诉jQuery不要去设置Content-Type请求头
11  xhrFields:{withCredentials:true},
12  async: true, //默认是true: 异步, false: 同步。
13  success: function (data) {
14    callback(data);
15  },
16  error: function (data) {
17    layer.msg('请求异常');
18  },
19 });

```

When the server get the image data information, it will read the file information in byte form, and the byte is encrypted byte to base64 and stored in the database.

## 5. Progress on Prototyping

## 5.1 Front-end Prototyping



- This is the login page.

The image shows a user interface for a social media or forum application. At the top, there is a header with a logo, the title "Bird Sung Gulley", a search bar, and buttons for "Register" and "Login". Below the header, there is a sidebar with navigation links: "Search", "MENU", "Posts" (which is highlighted), "Columns", and "Ranking". The main content area displays three posts in a grid format. Each post includes a user profile picture, the user's name, the time since posting, the post content, and interaction metrics like upvotes and downvotes. To the right of the posts, there are two sidebar sections: "Must-read posts" and "Featured links". The "Must-read posts" section lists "Posting notes for every column! Must read before you post!" and "Why Baishi Qi was regarded as goat ?". The "Featured links" section lists "Wikipedia", "Pixabay", and "Tongji University". At the bottom of the screen, there are icons for messaging, notifications, and sharing.

- This is where you can have a rough view of all the posts in a certain column.

The screenshot shows a user profile for "Bird Sung Guley". The main navigation bar includes "Post What You Want...", "New Post", and a user icon. On the left, a sidebar titled "PERSONAL NAVIGATOR" lists "Your Posts" (highlighted in orange), "Your Replies", and "Your likes". The main content area displays a post by "@J. Cole" from May 15, 2022, at 12:05. The post content is "How to find some tutorials for learning how to paint? XXXXXXXXXX XXXXXXXXX (pics are allowed here)". Below the post is a "Comments" section with a text input field "Type here your comments..." and a "Comments" button. A comment by "@Elizabeth" from May 14, 2022, at 19:35, with content "XXXXXXX XXXXXXXX", has 12 likes and 3 replies. The replies are: "@Elizabeth, XXXXXXX by @Robot\_01" and "@Robot\_01, Thanks! by @Elizabeth". Another comment by "@A. Mohamed" from May 14, 2022, at 19:58, with content "XXXXXXXXXXXXXX", has 256 likes and 43 replies. The right side of the screen features a large red swan logo and the text "@J. Cole 101".

- Where you can have more detailed view of a post.

The screenshot shows the "New Post" creation interface for "Bird Sung Guley". The top bar includes "New Post" and a user icon. The sidebar "PERSONAL NAVIGATOR" shows "Your Posts" (highlighted in orange) and other options like "Your Replies" and "Your Likes". The main area is titled "New Post" and contains fields for "Choose categories", "Type catching attention title", and "Type your post!". It also includes buttons for "Add Image", "Save as draft", and "Publish". To the right, there are two sidebar sections: "Must-read posts" (with notes about reading notes before posting) and "Featured links" (listing Wikipedia, Pixabay, and Tongji University).

- Write your post here...

## 5.2 Back-end Prototyping

```
events
{
    worker_connections 1024;
}

http
{
    include mime.types
    default type application/octet-stream;

    log_format main '$remote_addr - $remote_user[$time_local] "request" '
                    '$status $body_bytes_sent "$http_referer"'
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log logs/access.log main;
    sendfile on;
    tcp_nopush on;

    gzip on;
    upstream tomcat_proxy
    {
        server 192.168.3.63:8080 max_fails = 3 weight = 1 fail_timeout = 60;
        server 192.168.3.64:8080 max_fails = 3 weight = 1 fail_timeout = 60;
    }
}
```

- This is how we intend to balance the workload of all the servers, we'd like to offer a convenient using environment for all the users.

```
@Service
public class LoginServiceImpl implements LoginService
{
    @Autowired
    private UserMapper UserMapper;

    @Override
    public Result login(LoginDTO loginDTO)
    {
        if (StringUtils.IsEmpty(loginDTO.GetLoginName()))
            return new Result(400,"You haven't enter your account!","");
        if (StringUtils.IsEmpty(loginDTO.GetPassword()))
            return new Result(400,"You haven't enter your password!","");
        QueryWrapper<User> wrapper = new QueryWrapper();
        wrapper.eq("login_name",loginDTO.GetLoginName());
        User user = UserMapper.SelectOne(wrapper);
        if (user != null && user.GetPassword().equals (loginDTO.GetPassword()))
        {
            LoginVO loginVO = new LoginVO();
            loginVO.SetID(user.GetID());
            loginVO.SetToken(UUID.randomUUID().toString());
            loginVO.SetUser(user);
            return new Result(200,"", loginVO);
        }
        return new Result(401,"Login Failed","");
    }
}
```

- This is an example of implementing the login service.

## 6.Project Self-Reflection

## **Wenchao Liu**

This course offered me a different perspective to build a real software system, through this semester I learned how to analyze a system from functional analysis to architectural analysis. However, I had to confess that before taking this class, actually I had no idea about all the things i wrote down in this semester's work. The process of learning all these diagrams and mechanisms is beneficial for me.

the main problem I encountered in the course was the lack of knowledge of the architecture level and the shallow understanding of the corresponding technologies and implementations.

Besides, what is also of importance is that I have get accustomed to working with my partners as a team. We used some applications to collaborate with each other, this is very helpful in the further study.

## **Tong Fengming**

In this project, our group designed a painting forum platform. For me personally, I have no development experience before taking this course and doing this project.

This time, although I didn't write all the code to implement it, it really made me feel what kind of process software development is, and how complicated and difficult it is to develop an excellent software system. Our group went from the first definite function to the second From the production of subclass diagrams, sequence diagrams to the final interface specification, I can learn a lot in each team communication. What impressed me the most was the understanding of the development architecture in the third assignment. The real framework and technology choices made us really difficult, which also made me realize the importance of teamwork in system analysis and design.

## **Ze Sun**

Through the learning in class, I have a snapshot understanding of the principles of system analysis and design. But it was through the practice of group projects that I was able to find a combination of theory and practice. I realize that as tools, various diagrams are only the externalized embodiment of our thoughts.

All designs must be based on actual functionality and feasibility. As the design level continues to refine, it is possible that the previous framework will no longer be applicable, and the interface of our system needs to be changed differently depending on external conditions. From my perspective, the design and division of the system and the design of the interface are very difficult and important. We need to be proactive and have an understanding of the current state of the related application field. Teamwork is also very important, a reasonable and clear division of labor is the guarantee of work efficiency.

## **7. Participation**

---

All group members actively took part in every aspect of every assignment.

Every group members' contribution should be 100 percent.

## **8. Acknowledgement**

---

All the group members want to give our sincerest gratitude to Mrs. Liu for all the lectures offered and the advice on potential refinement.