DressingUp Design Specification

Ying Sun  
40115153  
Electronical EngineeringConcondir University  
rayelegy1004@gmmail.com

Shaoxia Yang  
40155499  
Electronical EngineeringConcondir University  
[shaoxiay@gmail.com](mailto:shaoxiay@gmail.com) Yanhe Lan  
40116293  
Electronical EngineeringConcondir University  
[yilia.lan@gmail.com](mailto:yilia.lan@gmail.com)

Sijie Min  
40152234  
Electronical EngineeringConcondir University  
[ellian98765@gmail.com](mailto:ellian98765@gmail.com)

*Abstract*—This document is to be used by the software engineering team to design one software application DressingUp. It will also be used by other design groups to understand the functions of the design in detail. It includes user stories, requirements, task distribution, design timeline and detailed function design etc.

Keywords—software engineering, ICDE system, system requirements, user requirements, database, capture pictures

# **Introduction**

This document will be used by the software engineering team to design one software application DressingUp. It will also be used by other design groups to understand the function of the design in detail. It includes user stories, requirements, task distribution, design timeline and detailed function design etc.

## **Project timeline and milestone**

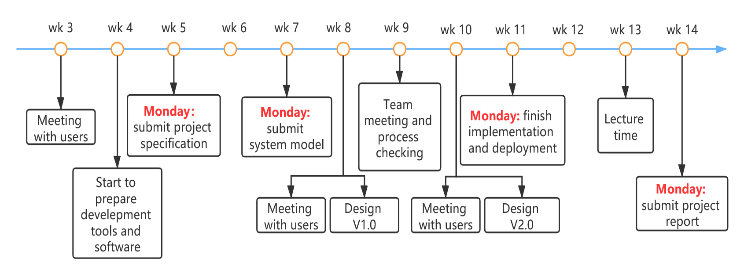


Fig. 1 project timeline

Milestone 1: week 10 Friday, deliver V1.0 implementation

Milestone 2: week 12 Friday, finish project report

## **scrum roles**

Product owner: Yanhe Lan → Sijie Min → Shaoxie Yang (Scrum sprint span: 3 weeks)

Master: Ying Sun

## **Working time**

48 hours per week/person.

## **Software processing model**

We use the Agile processing method in our project for it acts well when changes happen.

## **Skills and tools**

* Myeclipse and Mysql for database development
* Eclipse for Android development
* Java for coding
* Trello for activities recording
* Zoom for team meeting twice per week to sync our project development status
* GitHub for project data repository, Java code and documents
* WhatsApp and WeChat for group communication

## **Reversion control**

Table 1: version table

|  | Revision control | | |
| --- | --- | --- | --- |
| revision | date | description |
| 1 | V1.0.0 | 29/01/2020 | Draft version |
| 2 | V1.0.1 | 21/02/2020 | 1. Revise requirements description  2. Add model diagrams |
| 3 | V1.0.2 | 31/03/2020 | 1.Revise architecture designs  2. adopt MVC and Layered Architecture Pattern  3.Add a statistical count of software metrics |

# **icde system and system and user requirements**

## **problem statement**

Many people don't know how to dress themselves up when encounter some special occasions, for example dating, party and other activities etc. They need one method to help them to solve their clothes matching problems.

## **System architecture(ICDE system)**

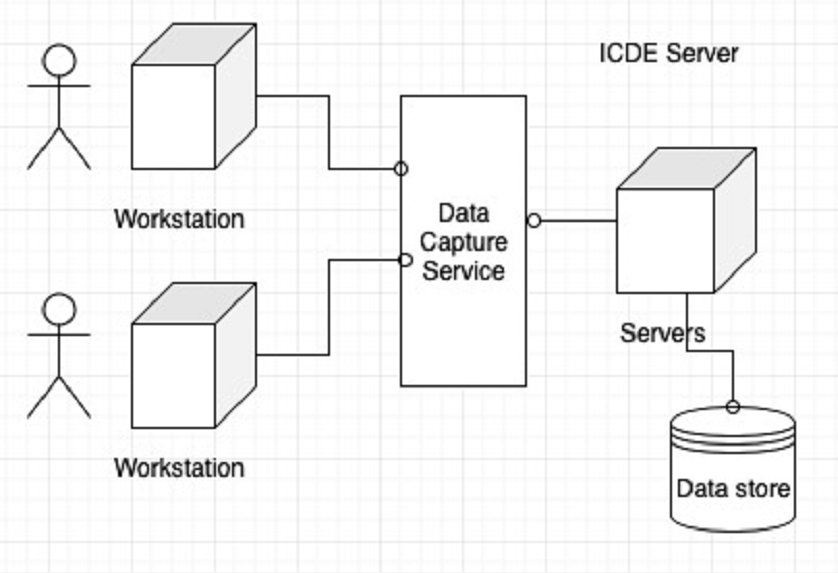


Fig. 2 ICDE system model

## **Project goal and scrum diagram**

This project is to design one software application which can help users match their clothes matching and give them some useful recommendations. This software application mainly contains user register and login, capture pictures from other websites, save captured pictures and user data into database and users like and favorites etc.

The project is planned by scrum processing method, and has 4 sprints for implementation, below diagram is the first sprint.

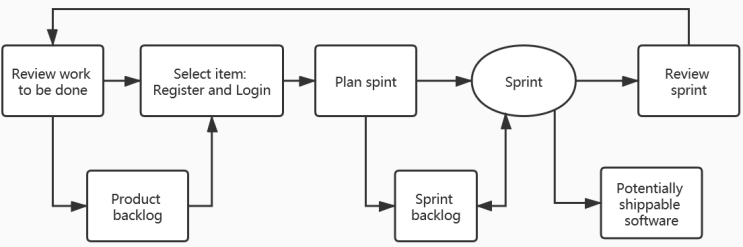


Fig. 4 sprint 1 of the implement

## **User stories**

Lily has a date in the weekend and she wants some help on her dressing and match.

First, she registers her account on the DressingUp software application and enters her basic information.

Second, she can choose the appropriate occasions: "date", "party", "company" "school", and so on. If she chooses a scene, she will be asked if she needs a "whole match recommendation" or "only accessories recommendation".

Third, if she chooses "overall match recommendation", the system will recommend a few suitable combinations for her based on the information stored in the database or some matches uploaded by other users. She can like the match ingested and will automatically add it to her favorites. If she has a favorite item, clicking on the picture system will jump directly to the product website to help her get product information.

Forth, if she chooses "only recommend accessories", she will be asked to get a photo of her clothes, and she can choose to take a photo or choose to upload an image. She will be recommended for appropriate accessories based on the clothing information she uploads.

She can also click on the details of the product.

This satisfactory result will be saved after the end of the selection.

## **system level architecture**

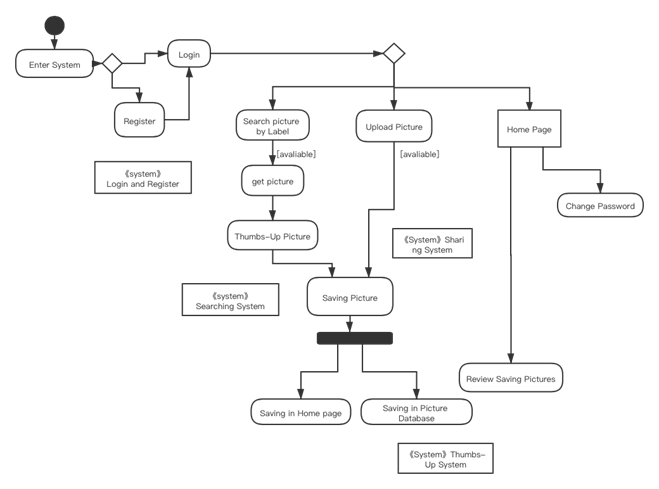
******

Fig. 3 system level architecture

## F**.user requirements**

1. Users shall be able to access the application with their email addresses to register and login
2. Users shall be able to access the application’s main page to search their desired matching recommendations by using some scenes key words like working, party and hang-out etc.
3. Users shall be able to access the application to upload their pictures and share them
4. Users could access the application to thumb up some pictures that they like

## G**.system requirements**

1. Build an interface for users to register and login
2. Build an interface to verify users’ identifications
3. Build a main page for users to search the matching recommendation pictures
4. Get pictures from picture database and show them on the main page
5. The searching result should be shown in order according to the number of thumbs-up.
6. Provide an interface for users to upload their pictures from phone’s camera or photo album
7. save users’ pictures into database with certain tags
8. Build a button for users to thumb up the pictures they like
9. record users’ favorites into user database

## H**.Task divide (sub-task)and distribution**

1. **Task divided into sub-tasks from the requirements**
   1. **Task 1 for user requirement 1 and system requirement 1.1 and 1.2**
2. Build a sign-in page for users
3. Build a log-in page
4. Build a function to send e-mail to users to verify their identification
5. Save user information to database (user database)
   1. **Task 2 for user requirement 2 and system requirement 2.1, 2.2 and 2.3**
6. Build a search page for users to search their desired pictures
7. Build a show page with sorting pictures
8. Build the second database (picture database) to store pictures
9. Get some matching pictures from internet
10. Add labels to pictures in database
11. Read number of thumbs-up for each chosen picture from the picture database
12. Exhibit the result pictures according to number of thumbs-up, from larger number to smaller number
    1. **Task 3 for user requirement 3 and system requirement 3.1 and 3.2**
13. provide an interface for users to upload pictures from camera or photo album
14. provide a path to visit users’ photo album
15. build a database for users pictures and information
16. save users’ upload pictures to database with tags
    1. **Task 4 for user requirement 4 and system requirement 4.1 and 4.2**
17. provide a up button for users to thumb up the pictures they like
18. add a tag to users thumb up pictures in database
19. provide a save button for users to save some pictures they like
20. save users favorites into database with tags
21. **sub-tasks distribution**

Table 2: sub-task and requirement

| User requirement | | System requirement | | Designer |
| --- | --- | --- | --- | --- |
| Task 1 | Subtask 1.1.1 | Task 1.1 | Subtask 1.1.1 | Yanhe Lan |
| Subtask 1.1.2 | Subtask 1.1.2 | Yanhe Lan |
| Subtask 1.2.1 | Task 1.2 | Subtask 1.2.1 | Yanhe Lan |
| Subtask 1.2.2 | Subtask 1.2.2 | Yanhe Lan  Ying Sun |
| Task2 | Subtask 2.1.1 | Task2.1 | Subtask 2.1.1 | Sijie Min |
| Subtask 2.1.2 | Subtask 2.1.2 | Sijie Min |
| Subtask 2.2.1 | Task2.2 | Subtask 2.2.1 | Shaoxia Yang  Ying Sun |
| Subtask 2.2.2 | Subtask 2.2.2 | Shaoxia Yang  Ying Sun |
| Subtask 2.2.3 | Subtask 2.2.3 | Shaoxia Yang  Ying Sun |
| Subtask2.3.1 | Task2.3 | Subtask2.3.1 | Ying sun |
| Subtask2.3.2 | Subtask2.3.2 | Ying Sun |
| Task3 | Subtask 3.1.1 | Task3.1 | Subtask 3.1.1 | Shaoxia Yang |
| Subtask 3.1.2 | Subtask 3.1.2 | Shaoxia Yang |
| Subtask 3.2.1 | Task3.2 | Subtask 3.2.1 | Ying Sun |
| Subtask 3.2.2 | Subtask 3.2.2 | Sijie Min |
| Task4 | Subtask 4.1.1 | Task4.1 | Subtask 4.1.1 | Sijie Min |
| Subtask 4.1.2 | Subtask 4.1.2 | Sijie Min |
| Subtask 4.2.1 | Task4.2 | Subtask 4.2.1 | Yanhe Lan |
| Subtask 4.2.2 | Subtask 4.2.2 | Yanhe Lan |

# **system models and detail task’s process model**

## **context description (use case diagrams)**

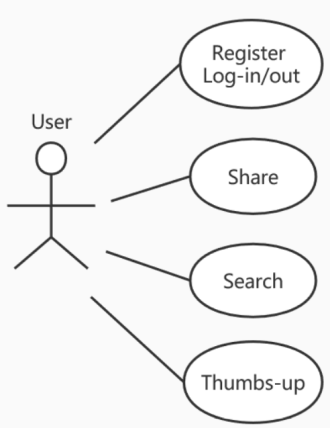


Fig. 5 use case diagram (overall level)

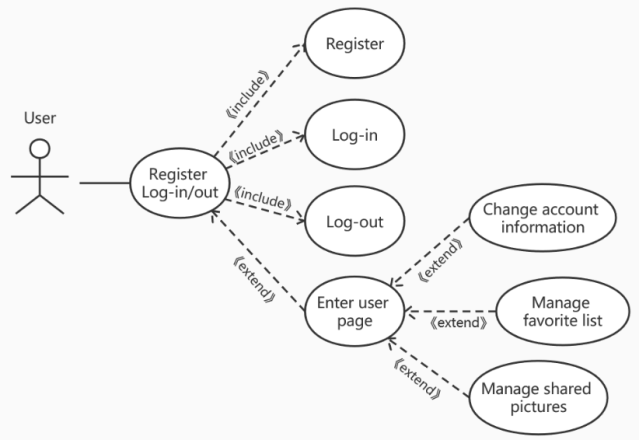


Fig. 6 use case diagram (user requirement 1)

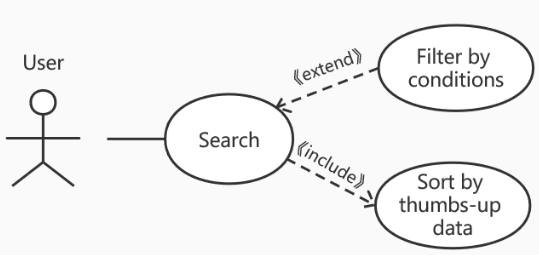


Fig. 7 use case diagram (user requirement 2)

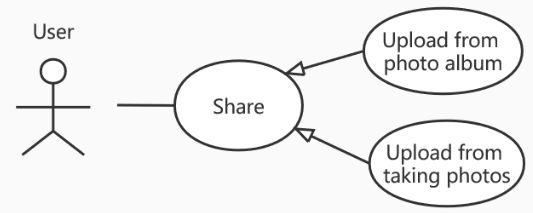


Fig. 8 use case diagram (user requirement 3)

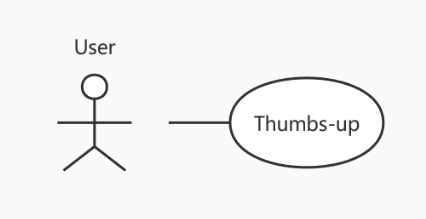
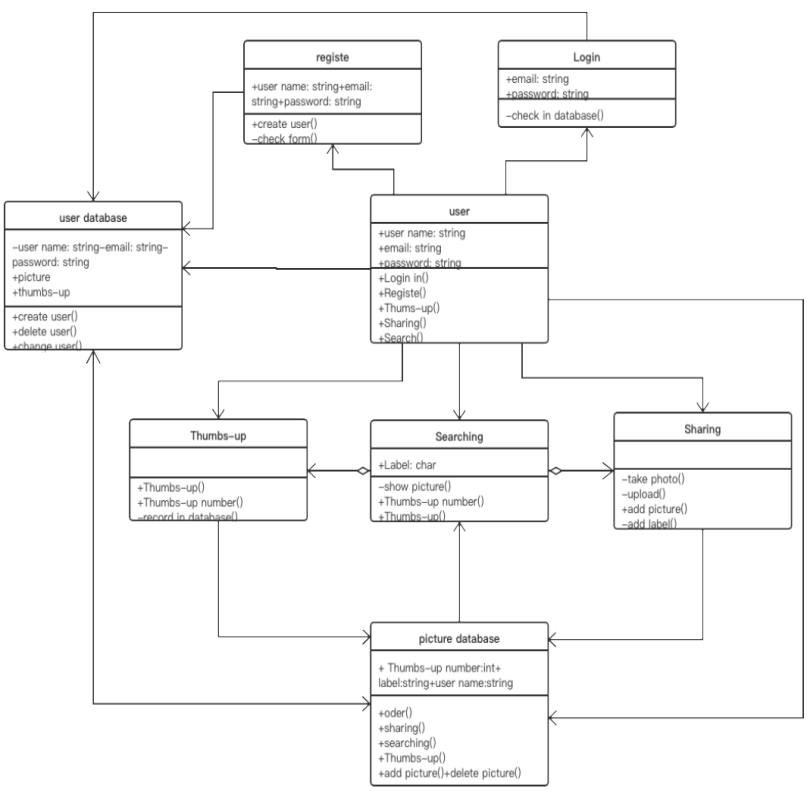


Fig. 9 use case diagram (user requirement 4)

## **Architecture diagram(class diagram)**



## Fig. 10 high level class diagram

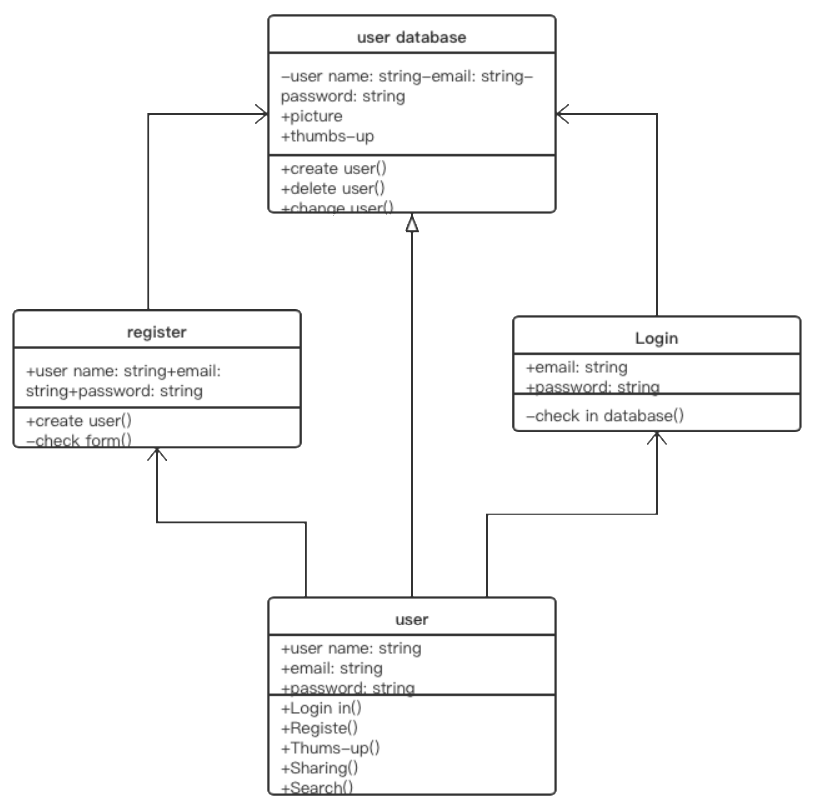


Fig. 11 class diagram (system requirement 1)

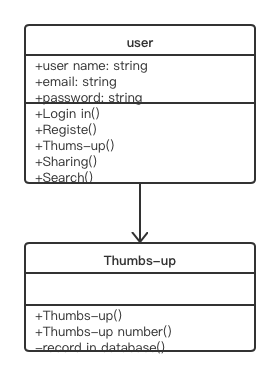


Fig. 12 class diagram (system requirement 2)

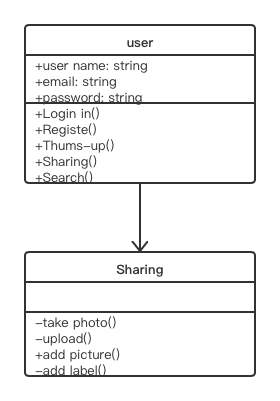


Fig. 13 class diagram (system requirement 3)

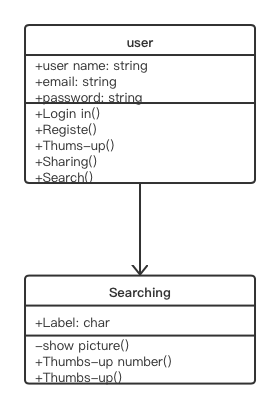


Fig. 14 class diagram (system requirement 4)

## **Task implement**

* + 1. Task 1
       1. Structural context and data flow of task 1

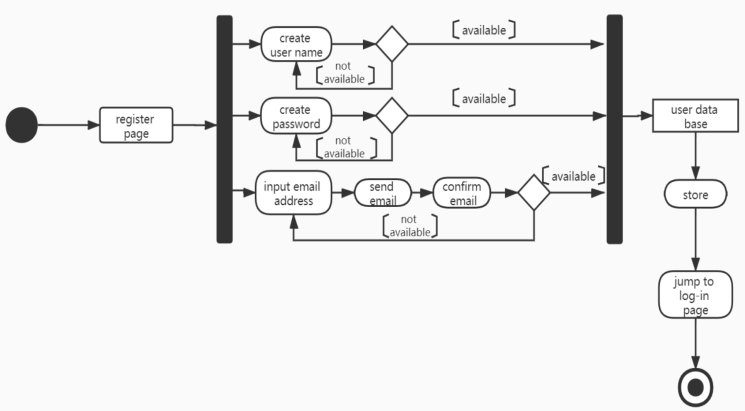


Fig. 15 Activity of Register in task 1

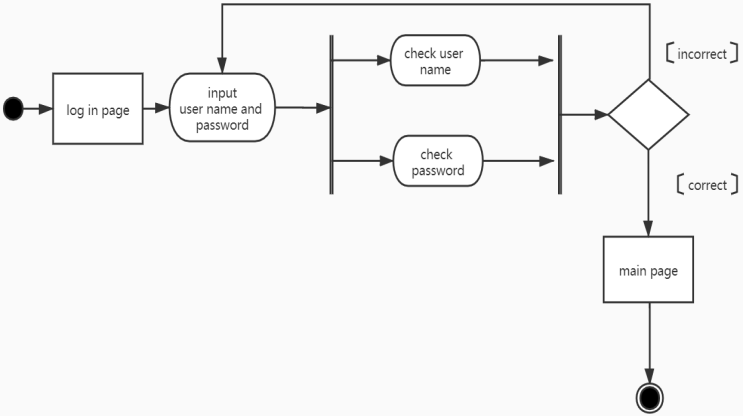


Fig. 16 Activity of Login in task 1

* + - 1. Interaction context with other components and system

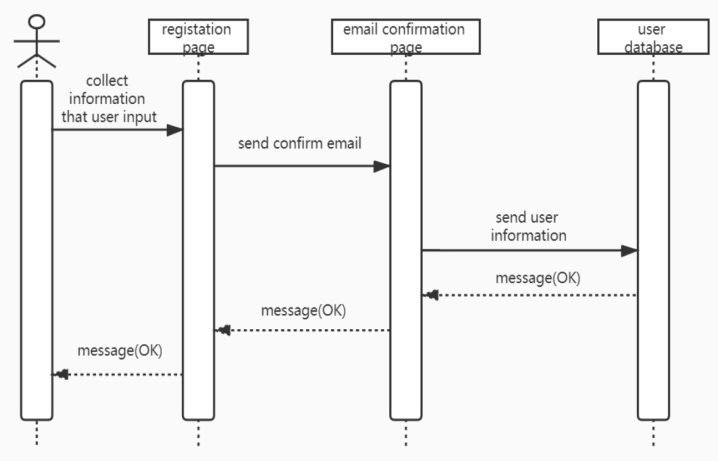


Fig. 17 Sequence diagram of Register in task 1

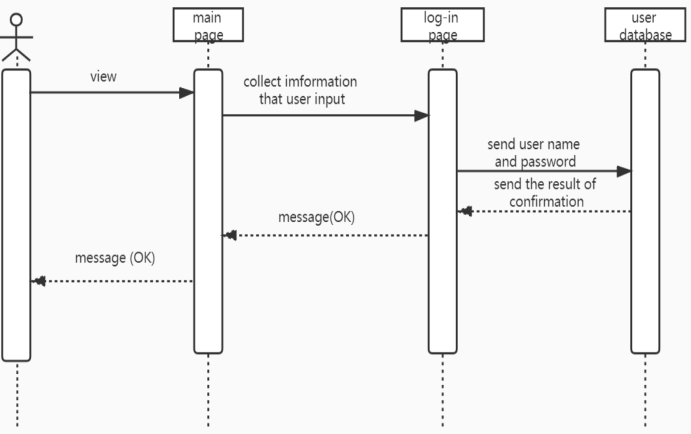


Fig. 18 Sequence diagram of Login in task 1

* + - 1. Behavior context with response to events

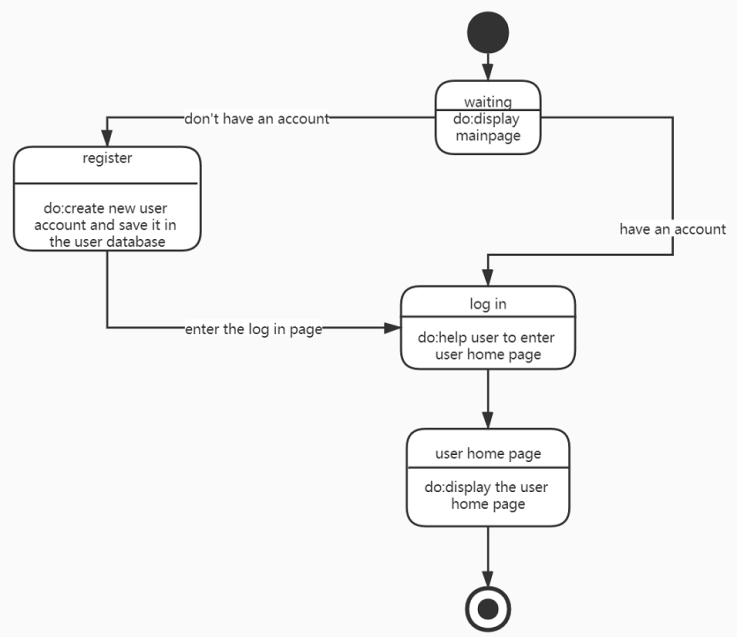


Fig. 19 State diagram of Register in task 1

* + 1. Task 2
       1. Structural context and data flow of task 2

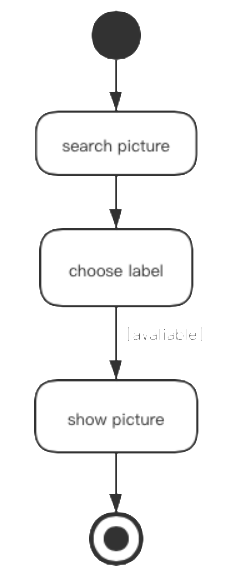


Fig. 20 Activity diagram of task 2

* + - 1. Interaction context with other components and system

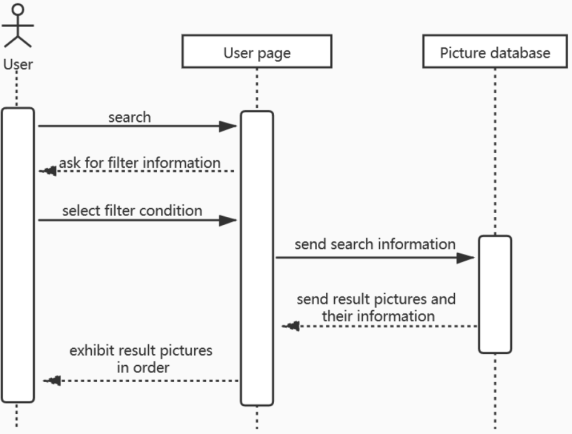


Fig. 21 Sequence diagram of task 2

* + - 1. Behavior context with response to events

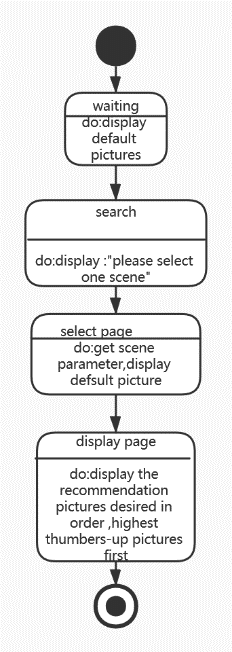
****

Fig. 22 State diagram of task 2

# **architectural designs and software engineering process**

## Identify and articulate architectural designs

Android with the recommended APP mainly implements the query and management of related information. This system has the advantages of simple interface, easy to understand, convenient management, and complete functions, so it has high application value, allowing Android mobile phone users to It is very convenient to view the matching information. This system has great advantages, has many potential customers, and has broad application prospects.

This system adopts Android and JSP dynamic webpage development technology, based on JSP operating system , eclipse system, MYSQL database and myeclipse database, and designs and implements a complete function recommendation app. This system has been tested with stable operation results and convenient and fast operation. It has realized an information platform with comprehensive functions, good practicability, high security, and good scalability and maintainability.

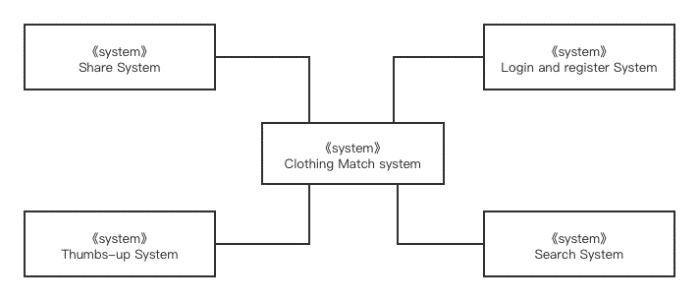


Fig. 23 The context of overall system

## Identify and articulate the associated software engineering process

This project uses myeclipse and eclipse to design a software application DressingUp. This software application mainly contains four main functions: user register and login, user share pictures, user thumb up picture, user search pictures. So we develop our system based on user requirements. In this case, this software application mainly contains database set up for dressing recommendation, user information and user thumb up. Below figures are the scrum of main functions.

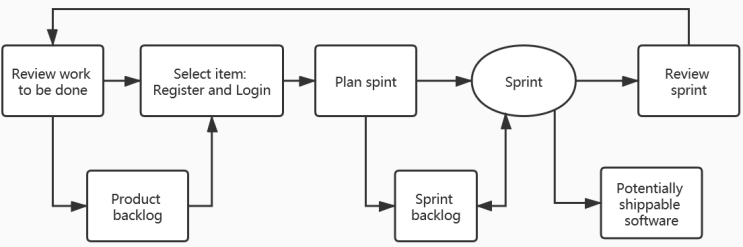


Fig. 24 The scrum of register and login

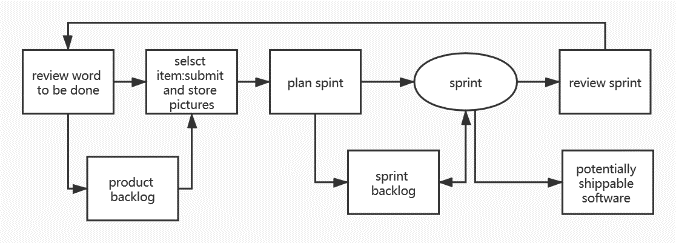


Fig. 25 The scrum of store and submit picture

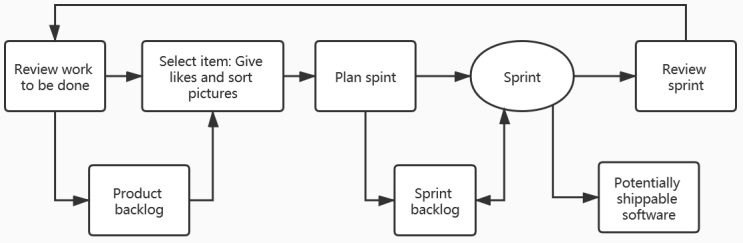


Fig. 26 The scrum of thumb up

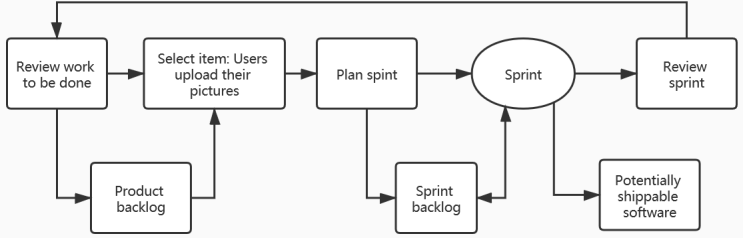


Fig. 27 The scrum of upload picture

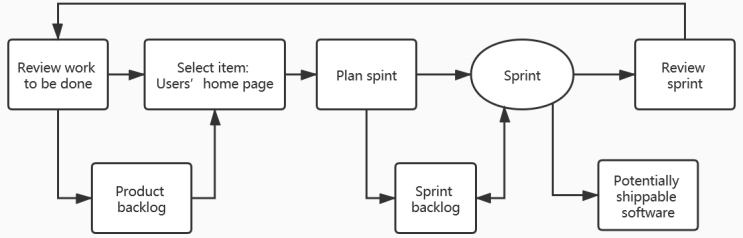


Fig. 28 The scrum of home page

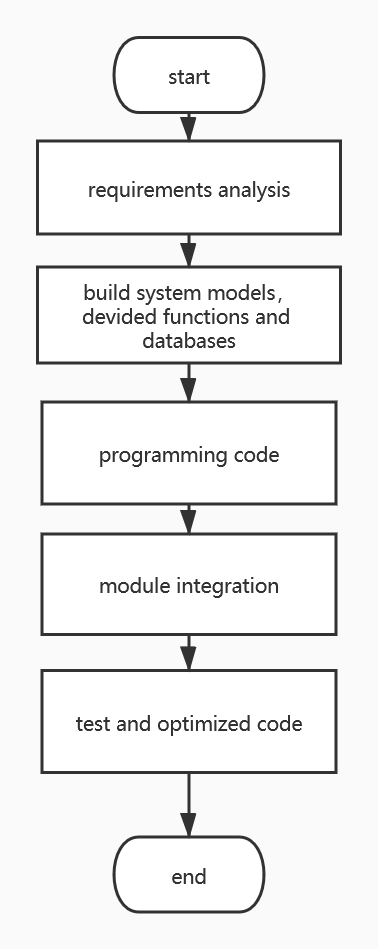


Fig. 29 The design flow chart

# **MVC and Layered Architecture Pattern to ICDE-App**

*A. Design decision:*

1. Why do we choose to design a mobile application instead of a website?

Considering of the situation that users will use our product, they always use it when they are going outside, such as hanging out, going to the party, going to work and so on. A mobile application can offer our users convenience so that they can quickly open and use it.

1. Why do we use Java language?

The main reason we choose Java is the independency. It means the Java source code can run on any platform. It’s portable and if a java code is written on one computer system then it can run on another PC. Also, Java is a simple Language without complex features like a pointer, operator overloading, etc. So, it is easy to learn.

1. Why do we design such a user interface?

As a daily used application, it should be very easy to use. Users don’t need to spend much time to learn how to use it but know how to operate as long as they see the user interface. The interface is beautiful and concise and everyone can easily start to use.

1. Why do we use three databases?

When design a software, it should come with a clear structure and low risks. Therefore, we use three databases, user information database, pictures information database and user’s favorite database. The first two databases contain a large amount of important information and they need frequently interaction when users want to check their own “likes”. So, the user’s favorite database provides the users’ “likes” information and the software only need to call a relatively simple database, which can reduce the running time. Besides, it can reduce the risk for they are independent.

*B. Conclusion:*

1)Advantages:

Our product meets the requirements of users and can provide a smooth and pleasant user experience. Users can use it at any time and place with their phones, which is very convenient. Also, the operation is very simple to learn. It’s a low risk application which meets the security requirement.

2)Disadvantages:

Till now, the application has a limited number of functions and cannot provide customization service. Besides, it has no external application program interfaces so it cannot interact with third party applications.

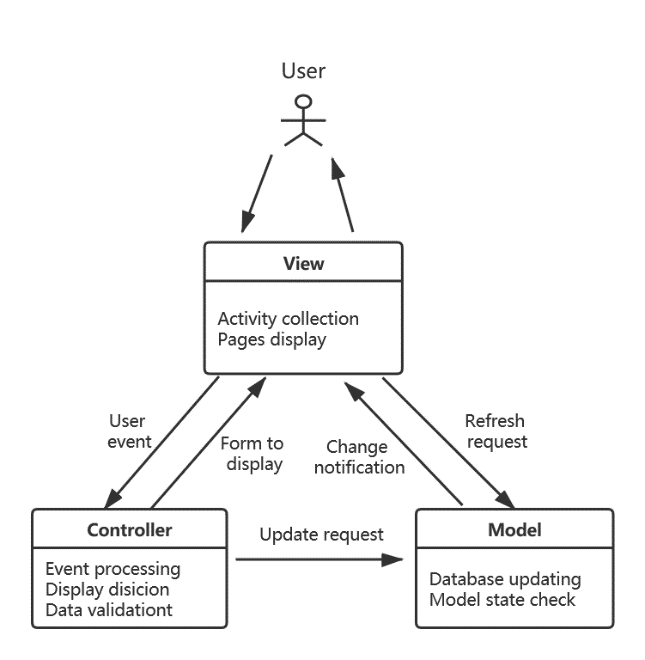


Fig. 30 architecture using MVC pattern

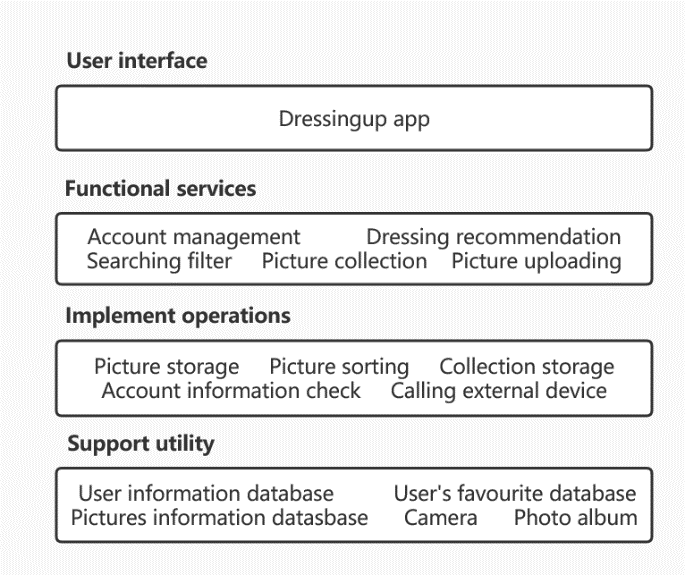


Fig. 31 layered architecture of Dressing Up app

# **Software metrics and granularity ofcomponents**

Table 3: statistical count of software metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Task name | loc | component Granularity Level | Numbers of Unit |
| 1.1 sing-in | 301 | Java class | 1 |
| 1.2 log-in | 138 | Java class | 1 |
| 1.3 user database | NA | Framework | 1 |
| 1.4 user home | 800 | Libraries | 7 |
| 2.1 search page | 456 | Package | 7 |
| 2.2 picture database | NA | Framework | 1 |
| 2.3 thumbs-up | 892 | Libraries | 5 |
| 3 upload picture | 606 | Package | 4 |
| 4 exhibit thumbs-up picture  Thumbs-up database | NA | Framework | 1 |
| Overall code | 4134 |  |  |

Table 4: statistical count of weekly software metrics

|  | Week  4 | week 5 | Week 6 | week 7 | week 8 | week 9 | week 10 | Week 11 | week 12 | week 13 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Line of Code | 128 | 365 | 566 | 435 | 389 | 680 | 699 | 667 | 120 | 0 |

##### **References**

1. Ian Sommerville, Software Engineering, 10th ed., Pearson, March 24 2015.
2. Roger S Pressman and Dr. Bruce R. Maxim, Software Engineering: A Practitioner’s Approach, 8th Ed., Mcgraw-Hill Education, Jan 23 2014.
3. Ian Gorton, Essential Software Architecture, 2nd Ed., May 17 2014.
4. OMG Unified Modeling Language specification V2.5, June 2015
5. Programming Android: Java Programming For The New Generation Of Mobile Devices, 2nd ed., O'Reilly Media, October 22, 2012