2022

Conference Manage System

Joseph 1930026109
Oliver 1930026012
Matt 1930031130
Barry 1930026064

CONTENS

Specification

Main features

Benefit, User Friendliness

Demonstration

01

PART ONE

Specification Documents & Design

-- Methods & Diagrams & Models

Matt

□ Overview

Software Process Model :

The Waterfall Model

Design Pattern :

Model-View-Controller Model (MVC)

4 documents, 15 diagrams and many techniques applied.

Software Requirements Specification (SRS)

- Use case diagram
- State-transitional diagram
- Class diagram
- Sequence diagram

Detailed Design

- Reconstrued Class diagram

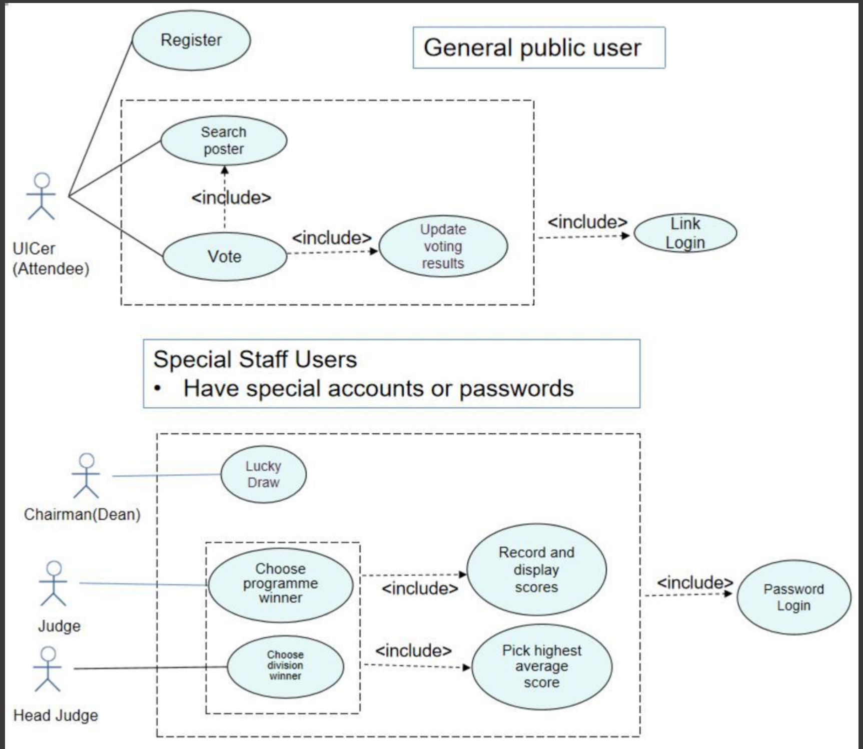
Architecture Design

- System overview

Software Testing

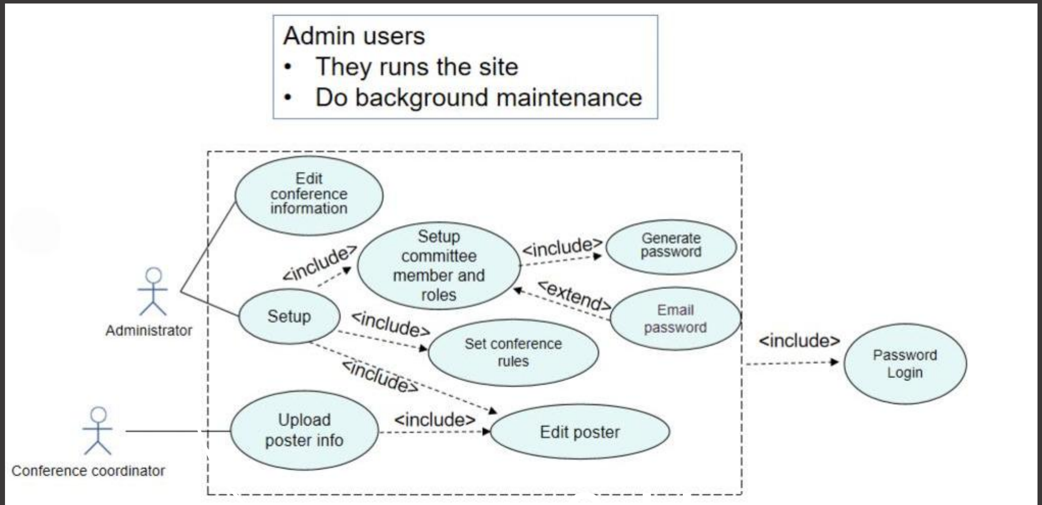
- Testing plan
- Testing report

Use Case Diagram



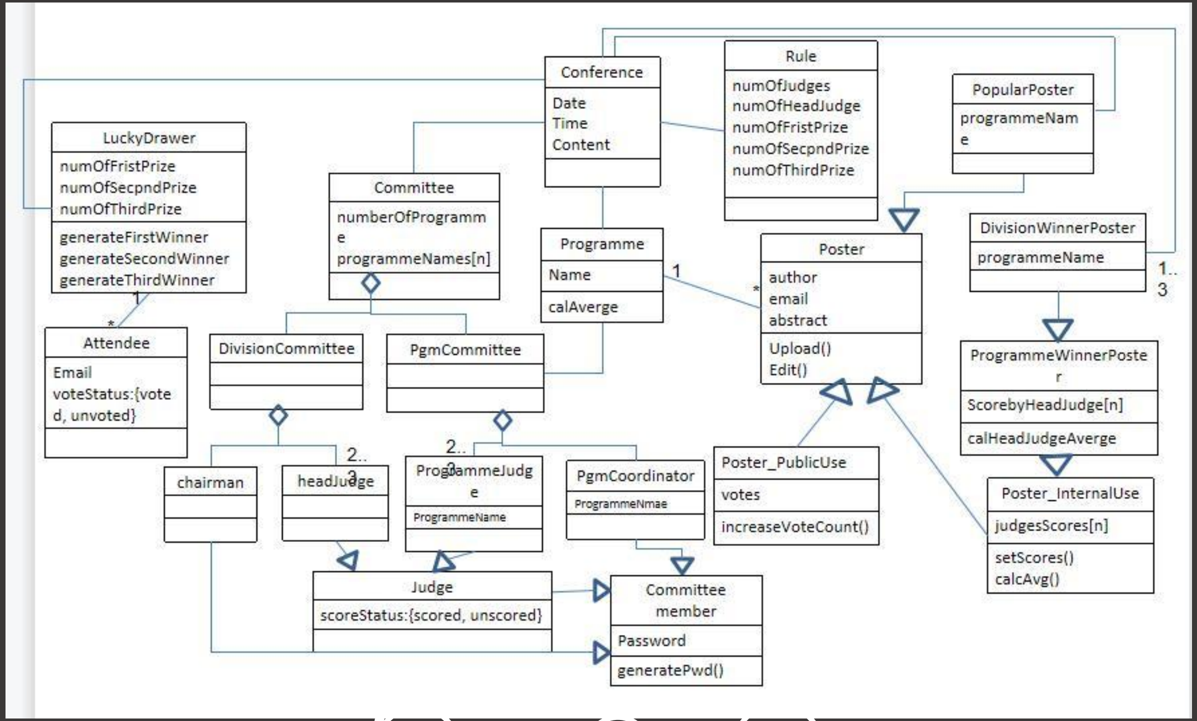
General public user

Special Staff Users
- Have special accounts or passwords

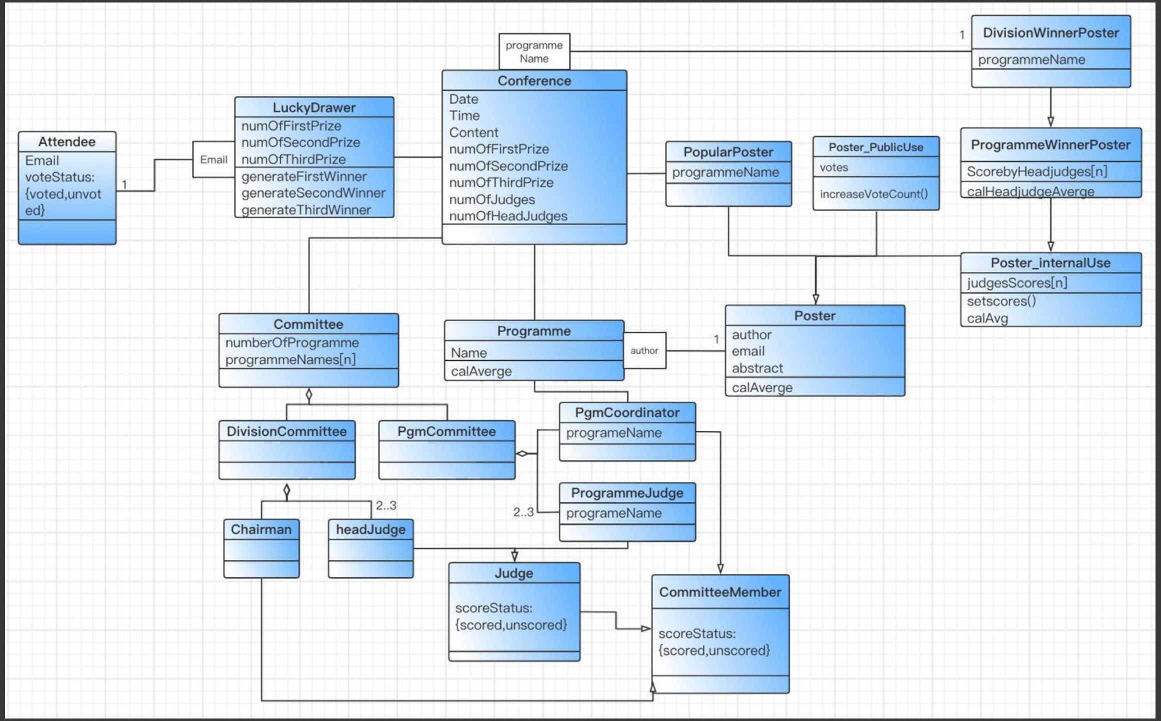


Admin users
- They runs the site
- Do background maintenance

□ Class Diagrams



- [illegible]

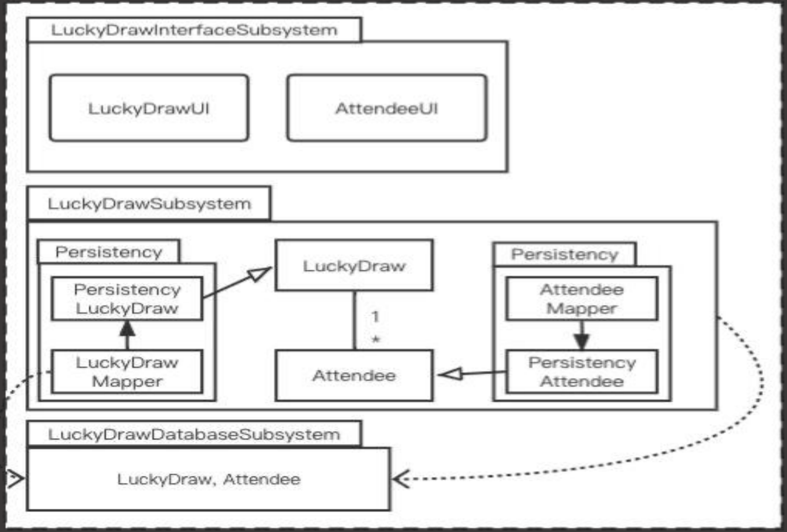


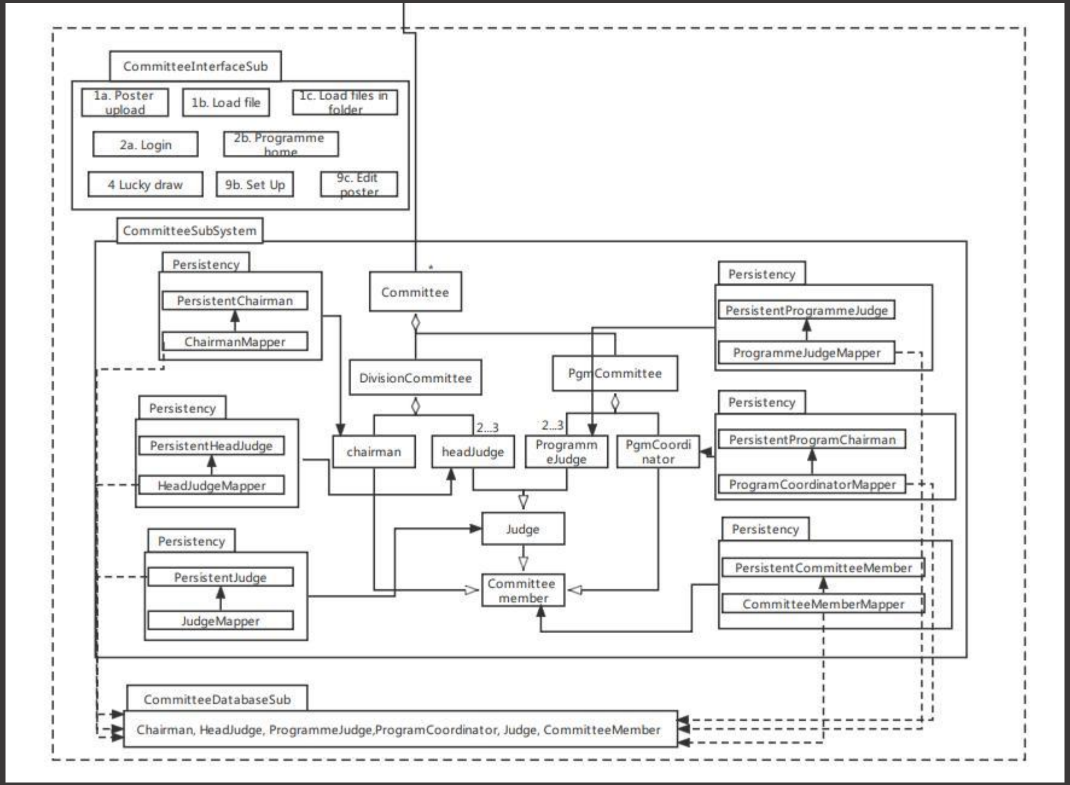
- Reconstructed Class diagram

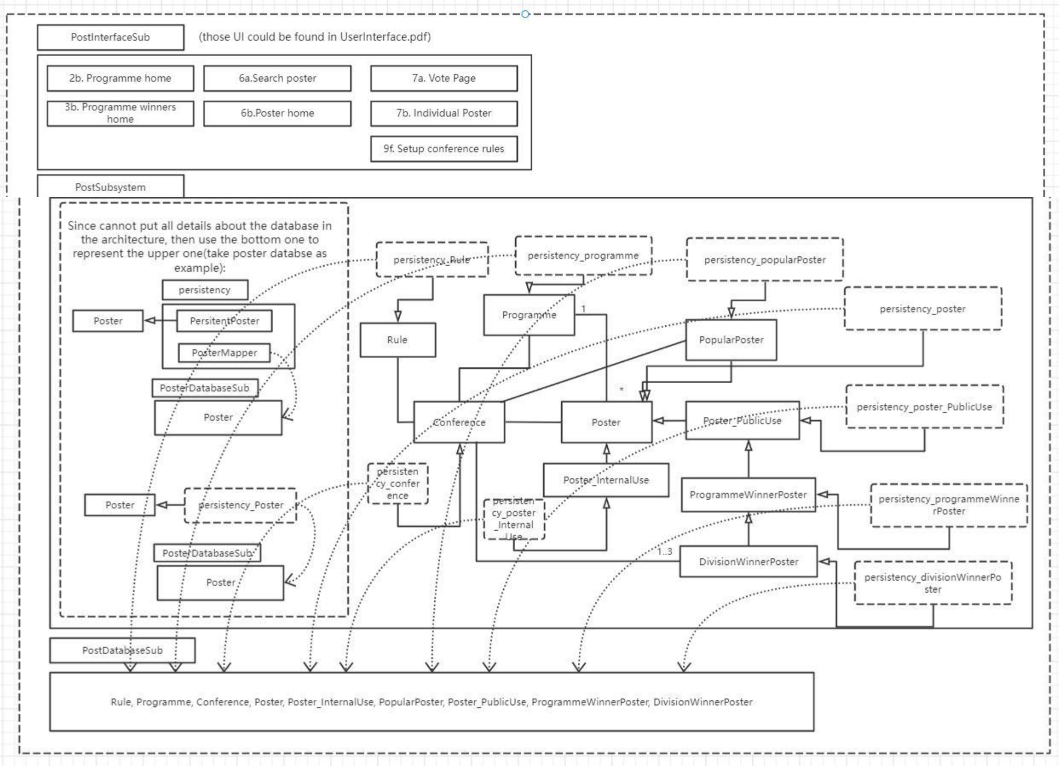
- Adding Qualifier
- Increasing Inheritance

□ System Architecture

- The MVC model
- Three Sub-systems






PostInterfaceSub
(those UI could be found in UserInterface.pdf)
2b. Programme home
6a.Search poster
7a. Vote Page
3b. Programme winners home
6b.Poster home
7b. Individual Poster
9f. Setup conference rules
PostSubsystem
Since cannot put all details about the database in the architecture, then use the bottom one to represent the upper one(take poster databse as example):
persistency
Poster
PersitentPoster
PosterMapper
PosterDatabaseSub
Poster
Poster
persistency_Poster
PosterDatabaseSub
Poster
persistency_Rule
persistency_programme
persistency_popularPoster
persistency_poster
Rule
Programme
1
PopularPoster
*
Conference
Poster
Poster_PublicUse
persistency_poster_PublicUse
persistency_conference
Poster_InternalUse
persistency_poster_Internal_Use
ProgrammeWinnerPoster
persistency_programmeWinnerPoster
1..3
DivisionWinnerPoster
persistency_divisionWinnerPoster
PostDatabaseSub
Rule, Programme, Conference, Poster, Poster_InternalUse, PopularPoster, Poster_PublicUse, ProgrammeWinnerPoster, DivisionWinnerPoster

□ Software Testing & Design Independence

T[illegible] L[illegible]:

Integration Testing

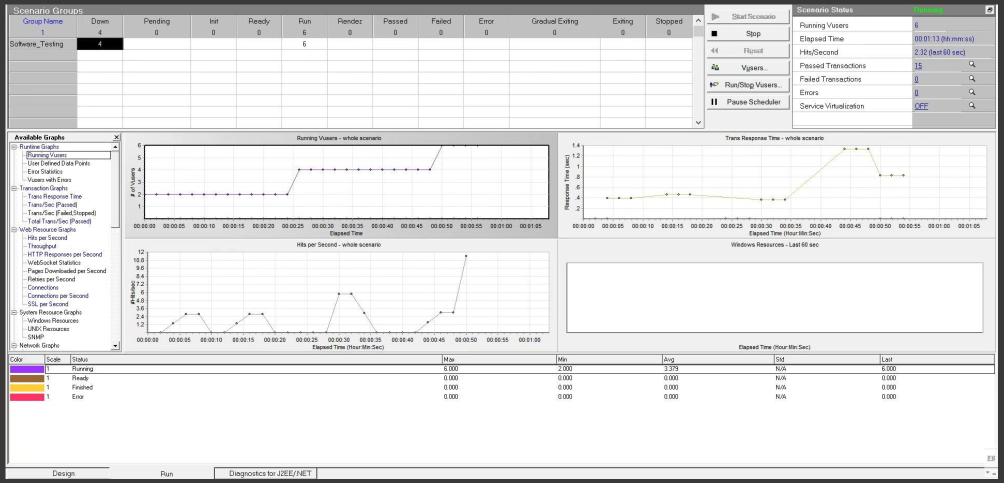
T[illegible] M[illegible]:

Black-box testing

(E[illegible])

Lo[illegible] T[illegible]

(Lo[illegible] R[illegible])



Scenario Groups

Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
1	4	0	0	0	6	0	0	0	0	0	0	0
Software_Testing	4				6							

Scenario Status Running

Running Vusers	6
Elapsed Time	00:01:13 (hh:mm:ss)
Hits/Second	2.32 (last 60 sec)
Passed Transactions	15
Failed Transactions	0
Errors	0
Service Virtualization	OFF

Color	Scale	Status	Max	Min	Avg	Std	Last
▇	1	Running	6.000	2.000	3.379	N/A	6.000
▇	1	Ready	0.000	0.000	0.000	N/A	0.000
▇	1	Finished	0.000	0.000	0.000	N/A	0.000
▇	1	Error	0.000	0.000	0.000	N/A	0.000

Optimizing the Cohesion and Coupling

02

PART TWO

Main features

Oliver

What tools do we use?

□ Concise

Python-Flask **Why?**

Comparation--Java : Python is more concise (time cost)

Comparation--Django : full-stack web framework

Flask : Light-weight framework (consider base on size)

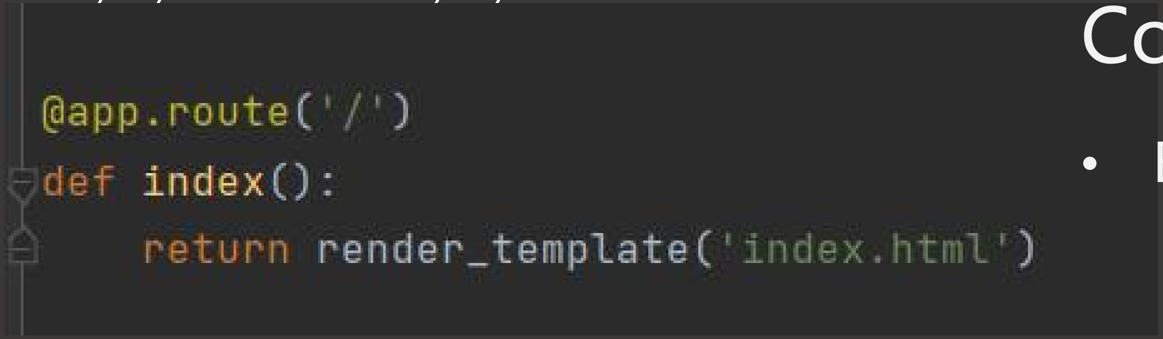
- Only two main dependencies : Werkzeug & Jinja2
- Flexible : Full control of decisions to build
- Coherent and neat API
- Easy and flexible configurations

What problems do we meet?

Connection

- Between HTML(templates) and controller

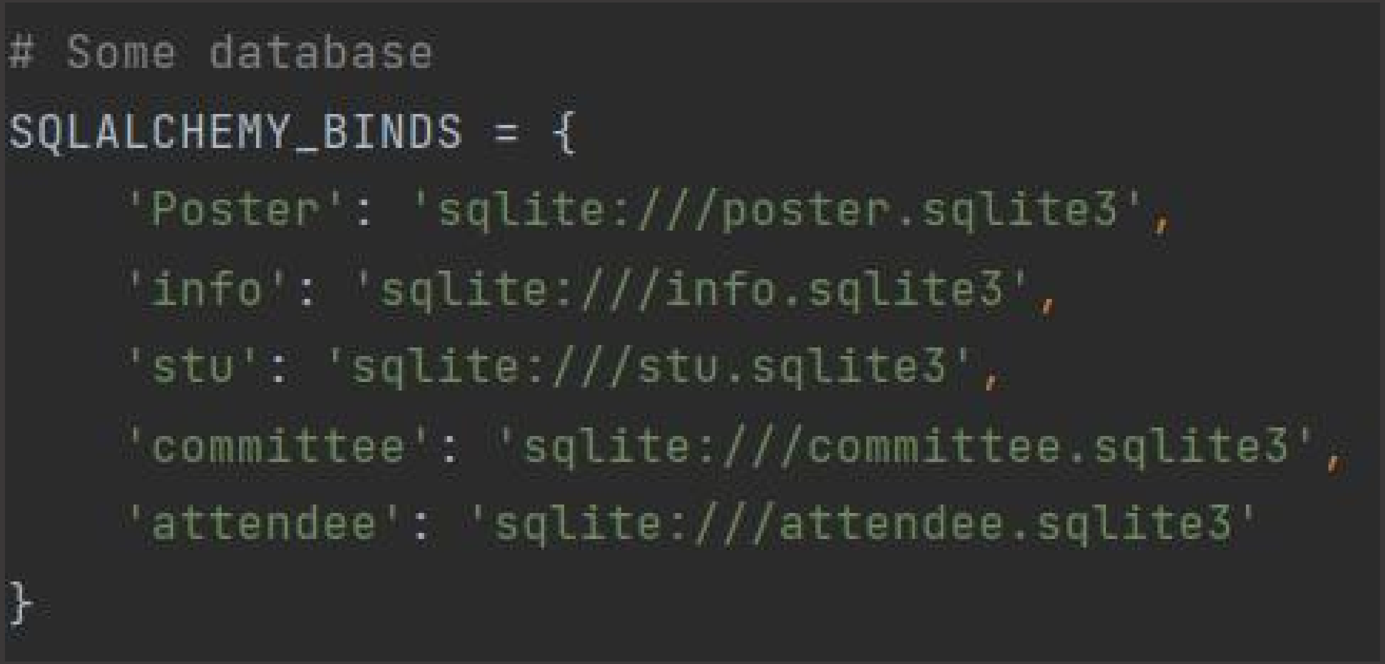
Route



```
@app.route('/')
def index():
    return render_template('index.html')
```

- Between model and database

Flask-sqlalchemy



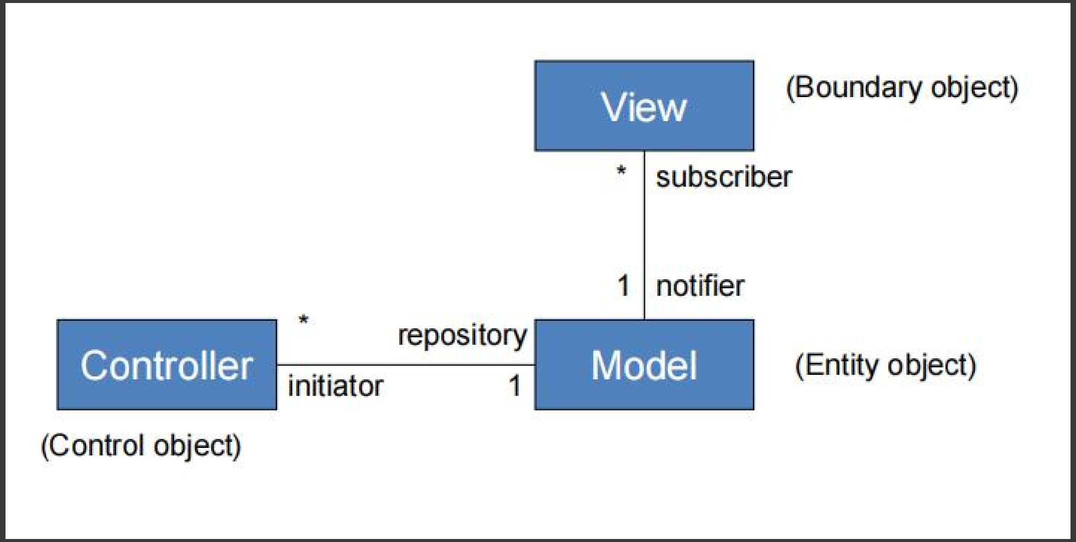
```
# Some database
SQLALCHEMY_BINDS = {
    'Poster': 'sqlite:///poster.sqlite3',
    'info': 'sqlite:///info.sqlite3',
    'stu': 'sqlite:///stu.sqlite3',
    'committee': 'sqlite:///committee.sqlite3',
    'attendee': 'sqlite:///attendee.sqlite3'
}
```

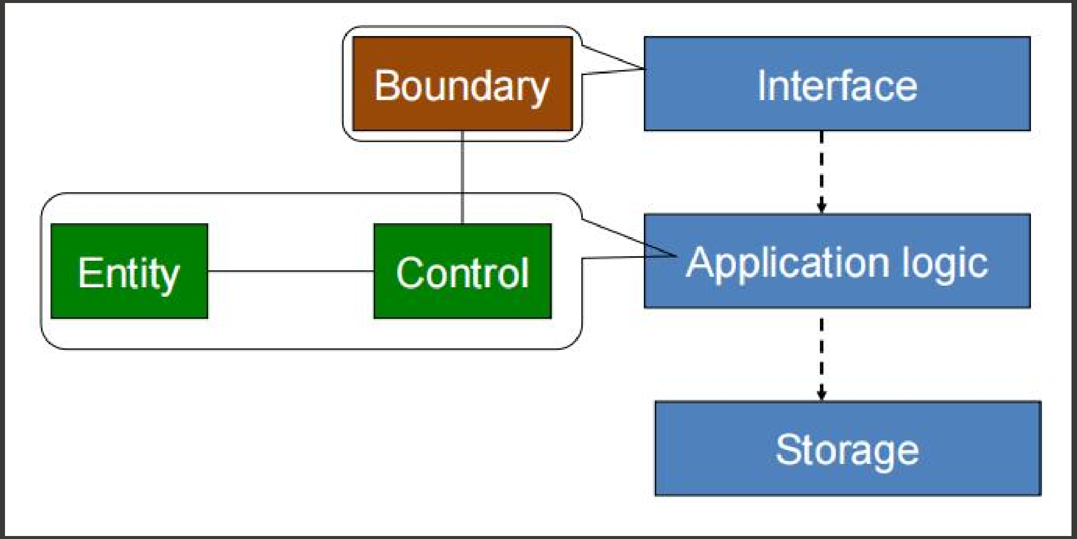
- □ **Easy for maintenance**
- □ **Easy to comprehend**

What do we changed during coding?

1. Model

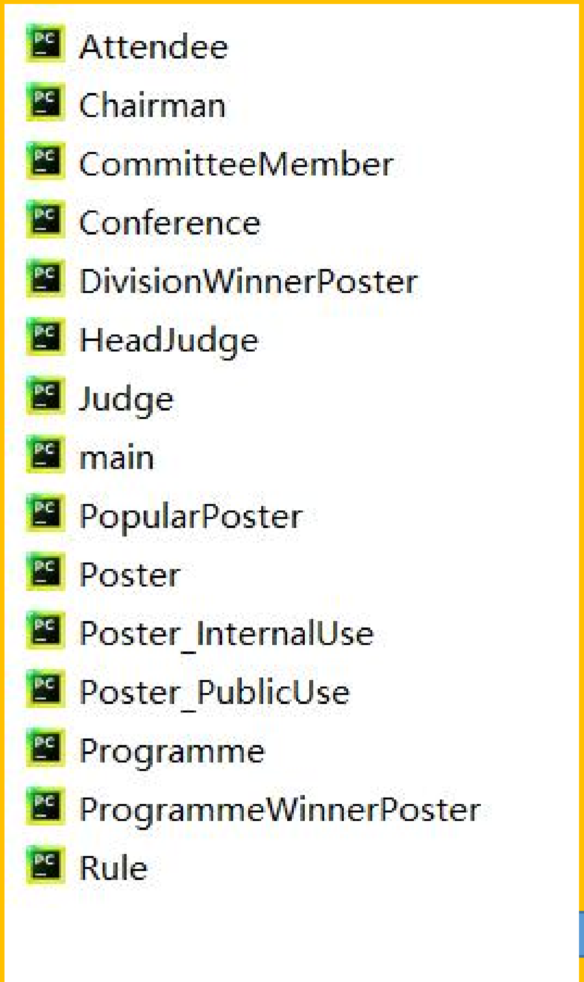
- MVC
- The Layered Model – Three Tier



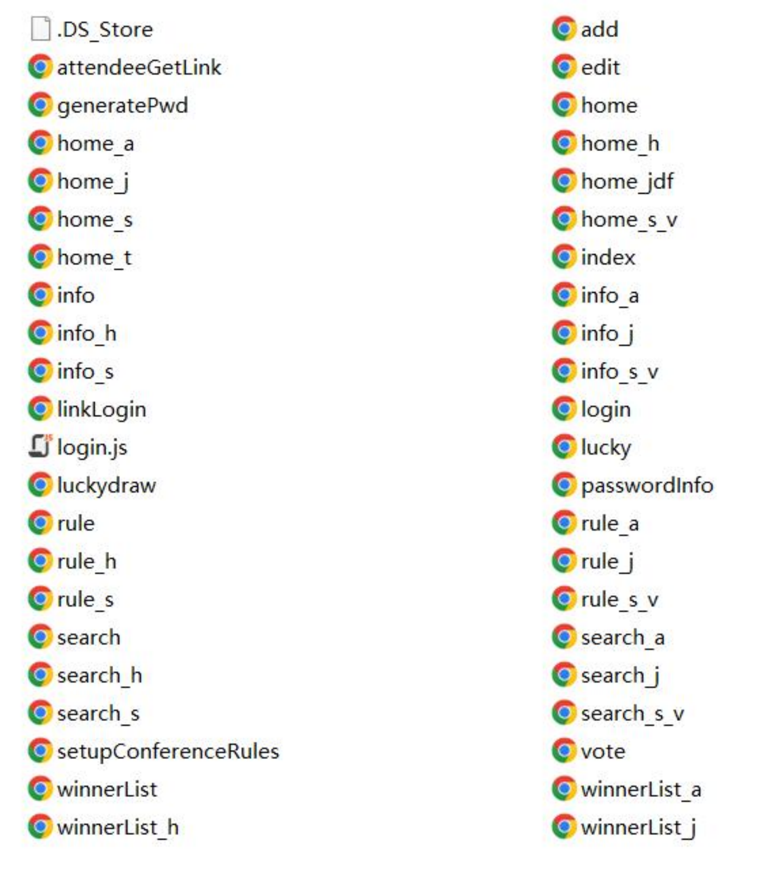


What do we changed during coding?

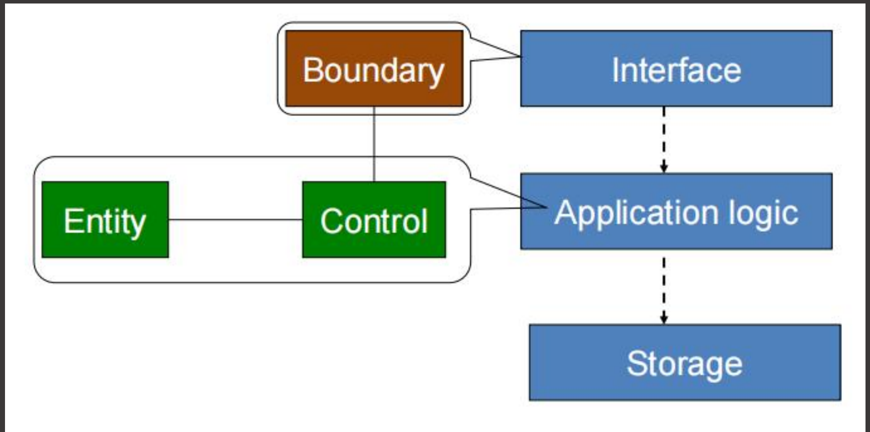
1. Model



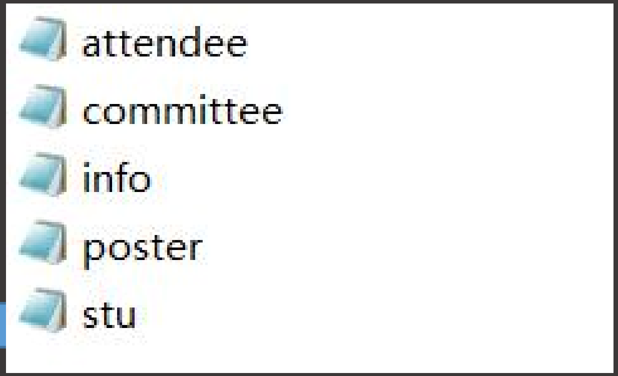
Attendee
Chairman
CommitteeMember
Conference
DivisionWinnerPoster
HeadJudge
Judge
main
PopularPoster
Poster
Poster_InternalUse
Poster_PublicUse
Programme
ProgrammeWinnerPoster
Rule



.DS_Store
attendeeGetLink
generatePwd
home_a
home_j
home_s
home_t
info
info_h
info_s
linkLogin
login.js
luckydraw
rule
rule_h
rule_s
search
search_h
search_s
setupConferenceRules
winnerList
winnerList_h
add
edit
home
home_h
home_jdf
home_s_v
index
info_a
info_j
info_s_v
login
lucky
passwordInfo
rule_a
rule_j
rule_s_v
search_a
search_j
search_s_v
vote
winnerList_a
winnerList_j



main

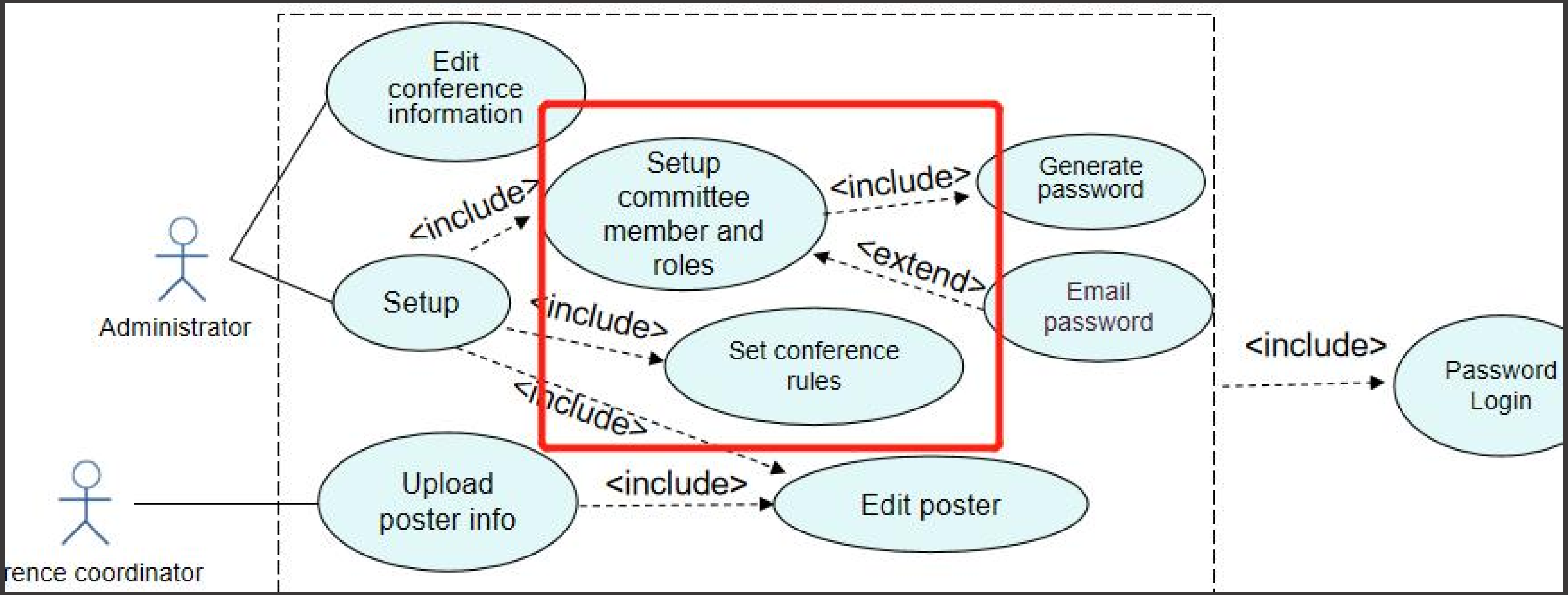


attendee
committee
info
poster
stu

What do we changed during coding?

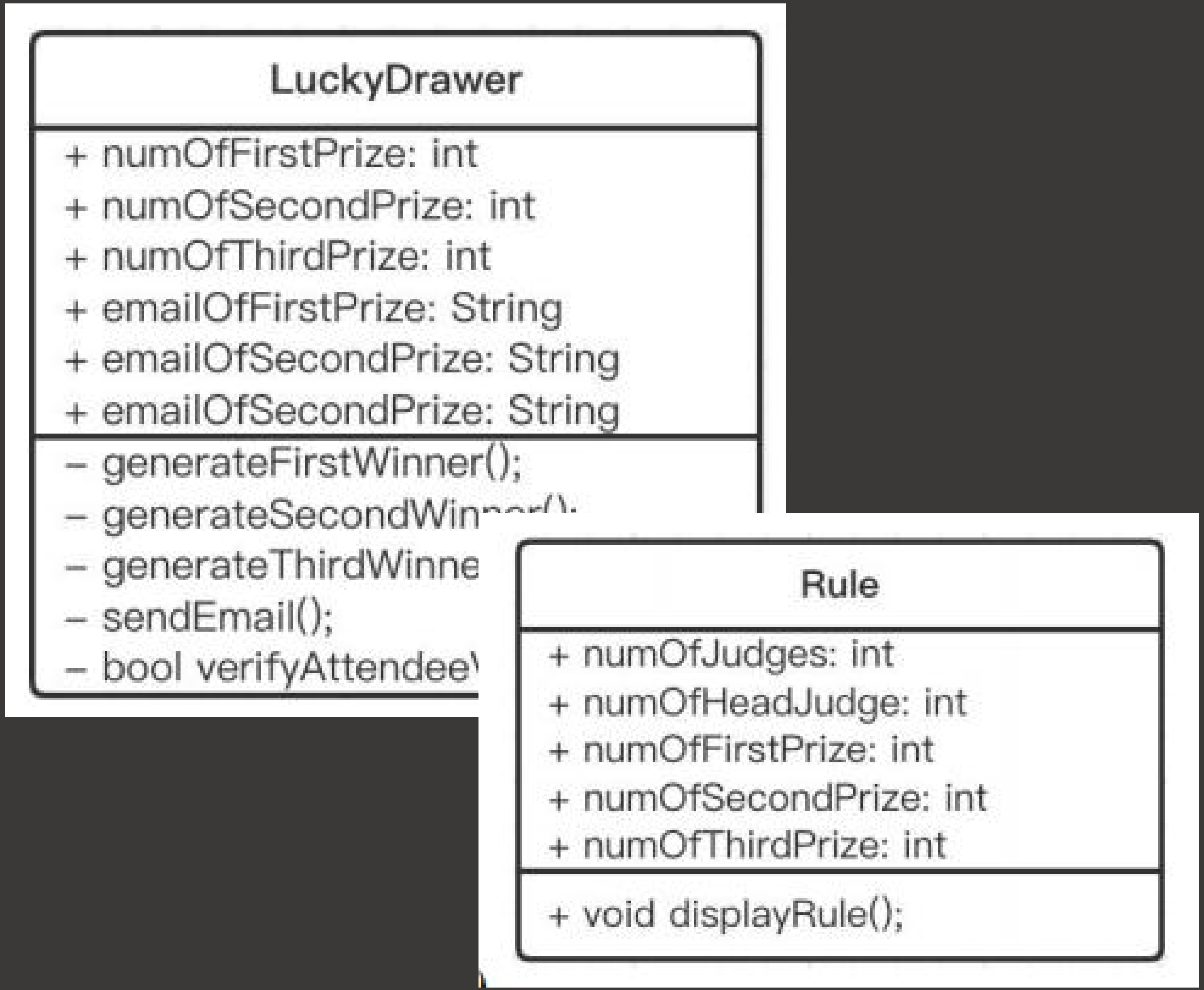
2. Assumption --
To make product concise & meet all requirements

- Rule



What do we changed during coding?

2. Assumption



LuckyDrawer

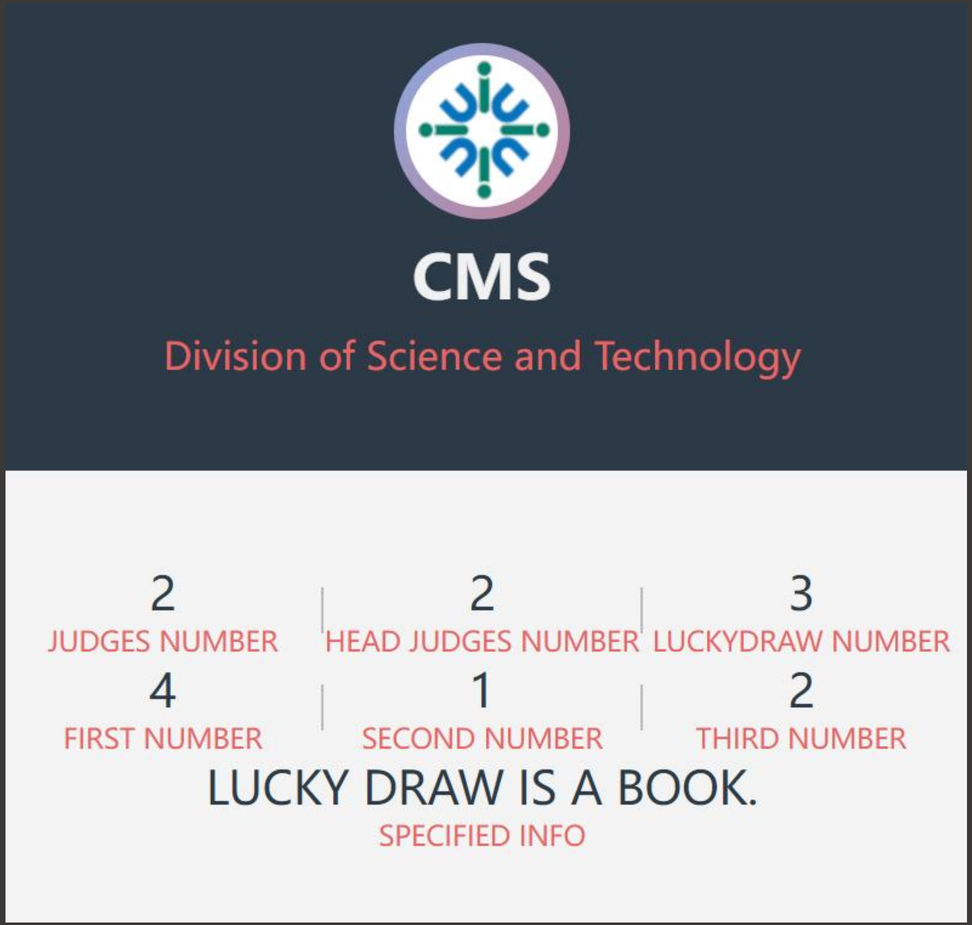
- + numOfFirstPrize: int
- + numOfSecondPrize: int
- + numOfThirdPrize: int
- + emailOfFirstPrize: String
- + emailOfSecondPrize: String
- + emailOfSecondPrize: String

- – generateFirstWinner();
- – generateSecondWin
- – generateThirdWinne
- – sendEmail();
- – bool verifyAttendee

Rule

- + numOfJudges: int
- + numOfHeadJudge: int
- + numOfFirstPrize: int
- + numOfSecondPrize: int
- + numOfThirdPrize: int

- + void displayRule();



CMS

Division of Science and Technology

2	2	3
JUDGES NUMBER	HEAD JUDGES NUMBER	LUCKYDRAW NUMBER
4	1	2
FIRST NUMBER	SECOND NUMBER	THIRD NUMBER

LUCKY DRAW IS A BOOK.

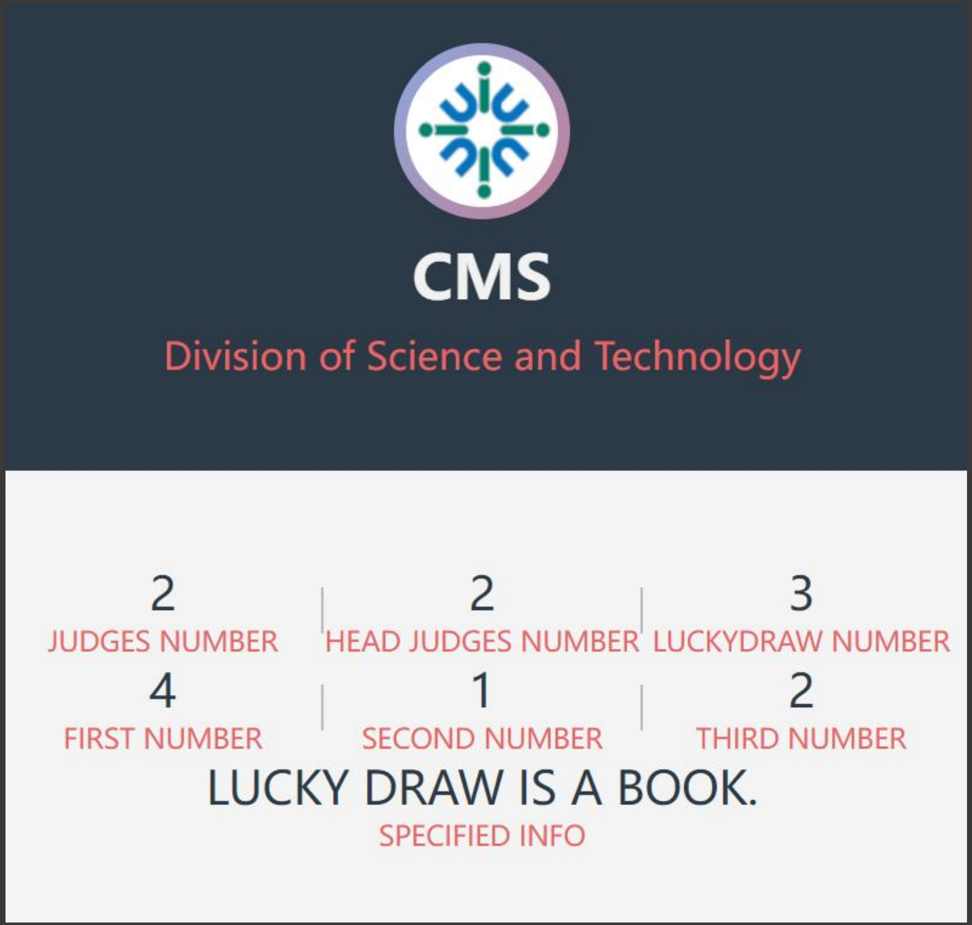
SPECIFIED INFO

What do we changed during coding?

2. Assumption

IDEA:

- □ **User Friendly**
- □ **Meet the requiements**



CMS

Division of Science and Technology

2	2	3
JUDGES NUMBER	HEAD JUDGES NUMBER	LUCKYDRAW NUMBER
4	1	2
FIRST NUMBER	SECOND NUMBER	THIRD NUMBER

LUCKY DRAW IS A BOOK.

SPECIFIED INFO

After solving those problems…

We done a nice work!

User Interface

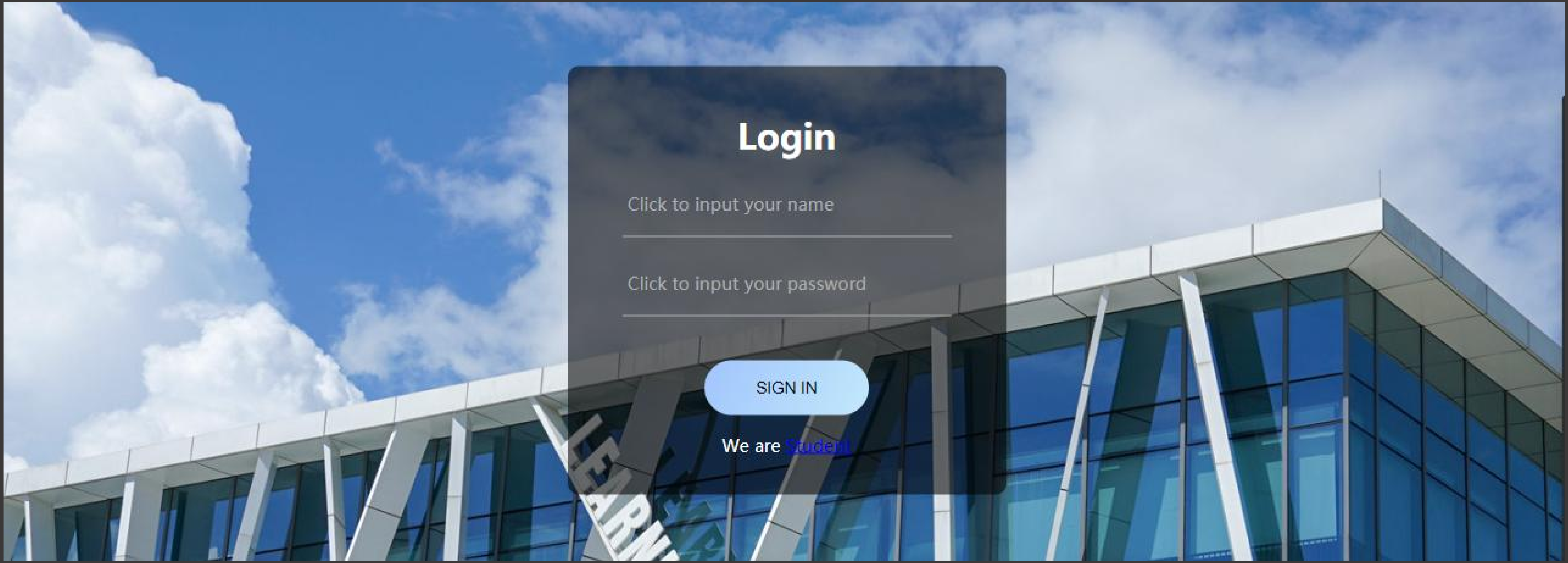
03

PART THREE

Benefit and User Friendliness

Joseph

Multiple login methods



Login

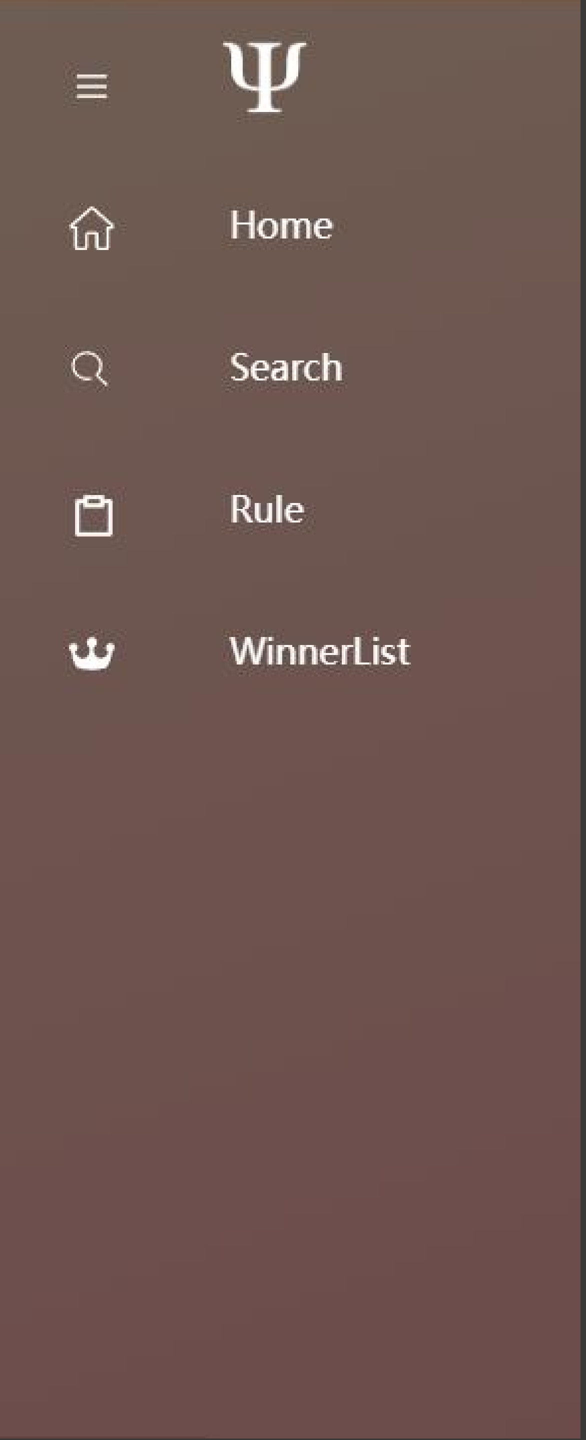
Click to input your name

Click to input your password

SIGN IN

We are Student

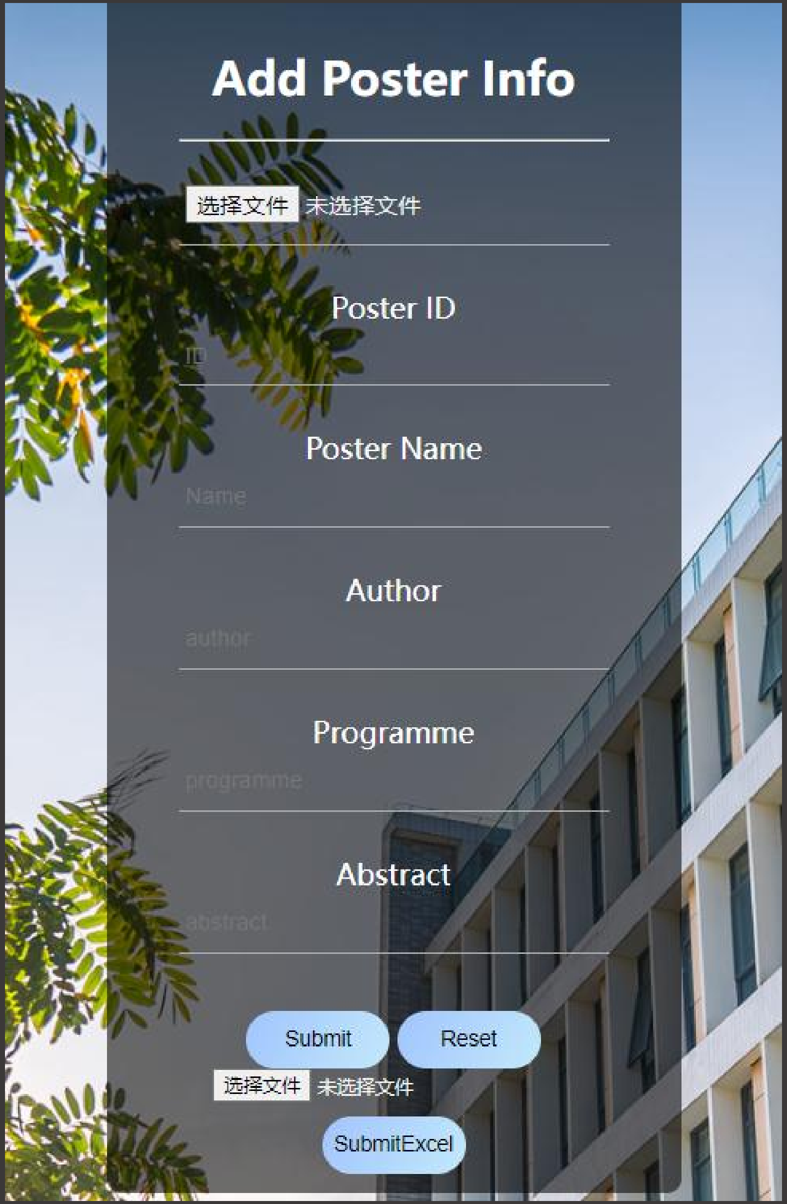
We provide different login methods for attendee and students.   Committee needs to enter the password while entering the account to log in.



Hover

Different users see different initial interfaces.
Here, we control the permissions of different users through the hover.

Userfriendly-Administrator



Add Poster Info

选择文件 未选择文件

Poster ID

ID

Poster Name

Name

Author

author

Programme

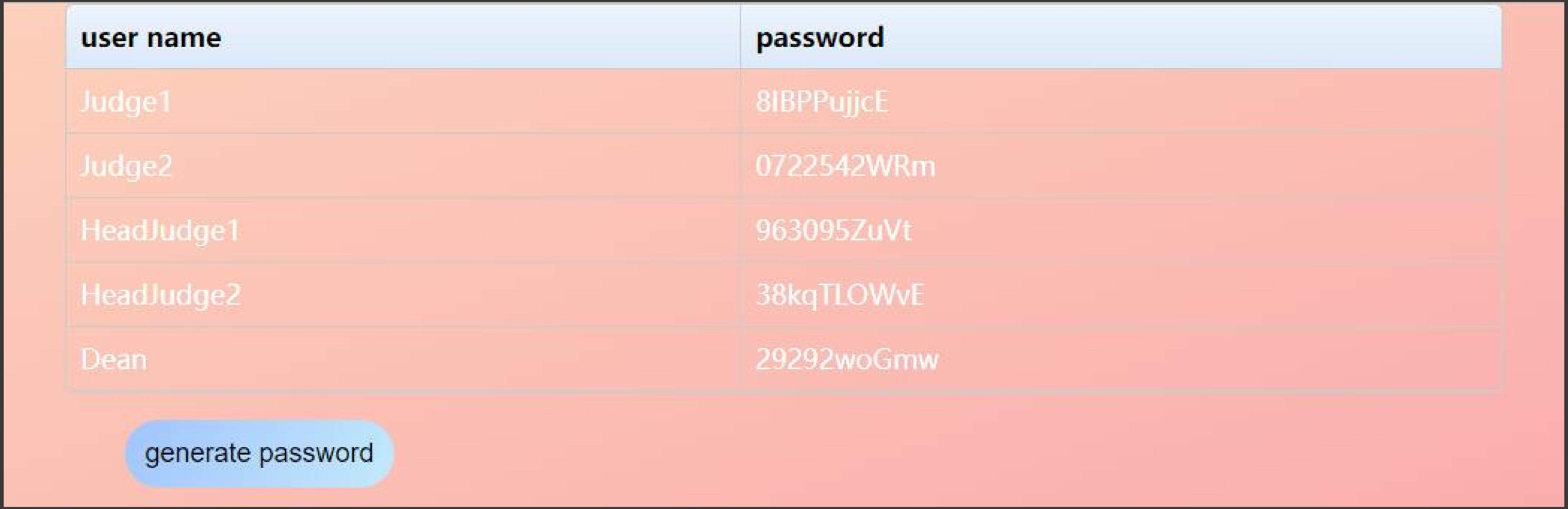
programme

Abstract

abstract

We provide a function of recording table data. The administrator can directly upload an excel table containing a large number of posters.

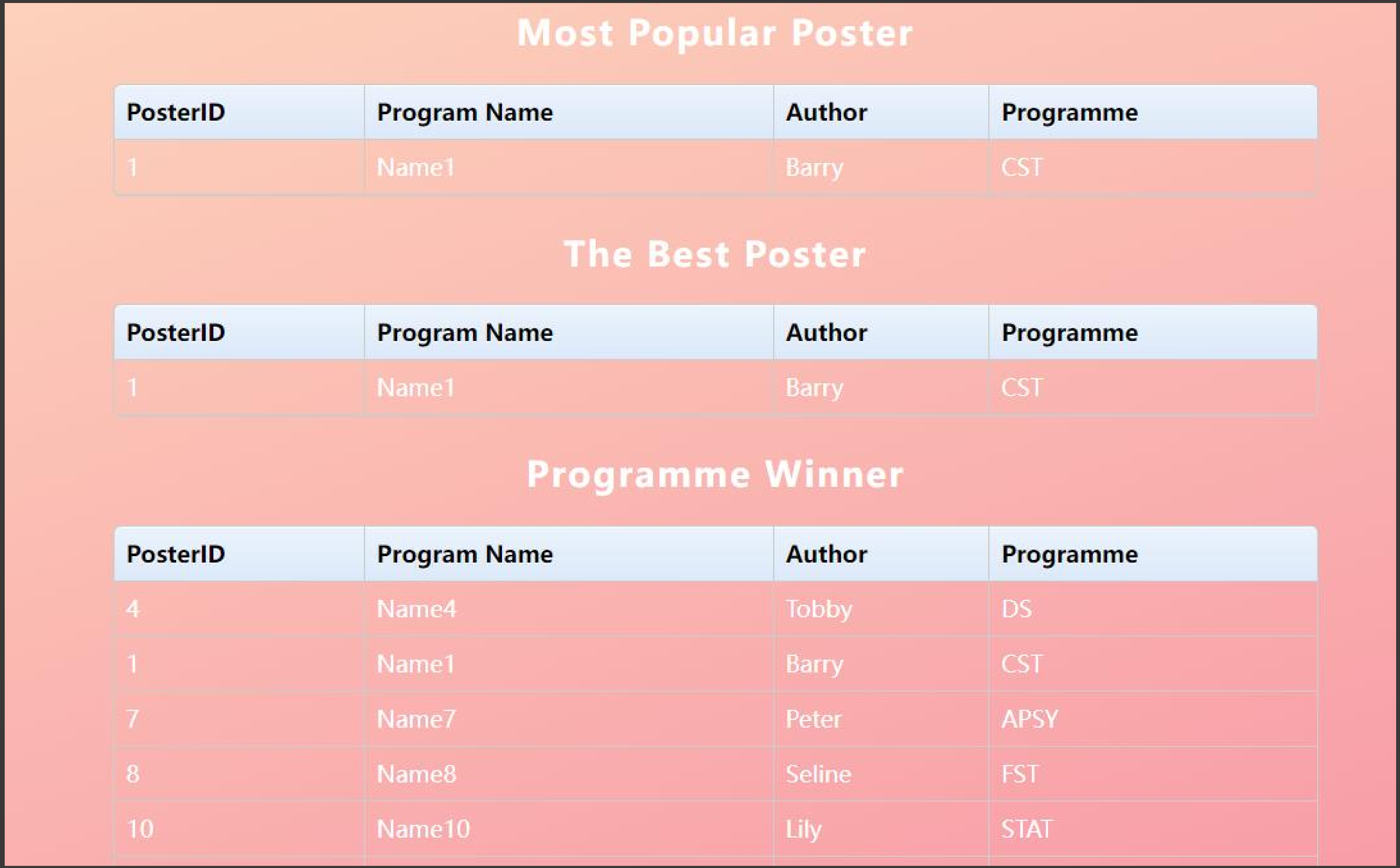
Userfriendly-Administrator



user name	password
Judge1	8IBPPujjcE
Judge2	0722542WRm
HeadJudge1	963095ZuVt
HeadJudge2	38kqTLOWvE
Dean	29292woGmw

The administrator can randomly generate passwords for all advanced users with one click.

Userfriendly-Judge



Most Popular Poster

PosterID	Program Name	Author	Programme
1	Name1	Barry	CST

The Best Poster

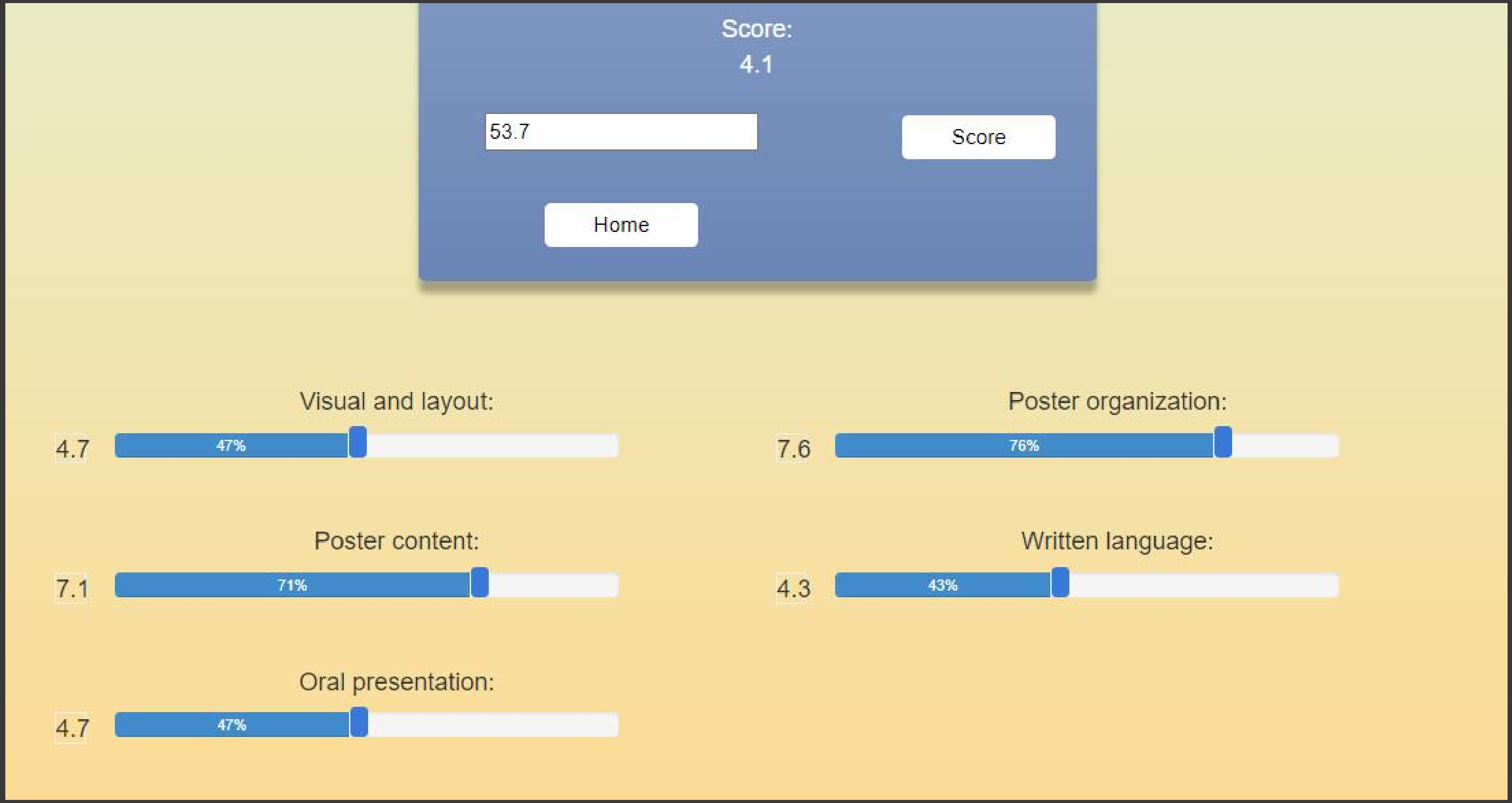
PosterID	Program Name	Author	Programme
1	Name1	Barry	CST

Programme Winner

PosterID	Program Name	Author	Programme
4	Name4	Tobby	DS
1	Name1	Barry	CST
7	Name7	Peter	APSY
8	Name8	Seline	FST
10	Name10	Lily	STAT

For two classes with similar permissions, judge and headjudge, we let them all see all the posters.

Userfriendly-Judge



Score:
4.1

53.7

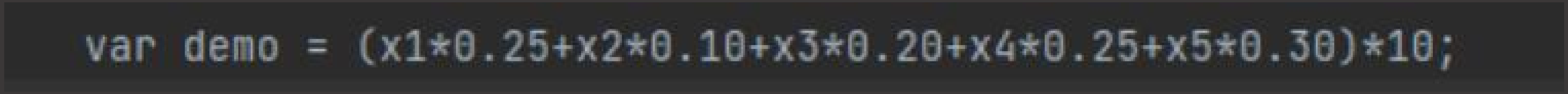
Visual and layout:
4.7 47%

Poster organization:
7.6 76%

Poster content:
7.1 71%

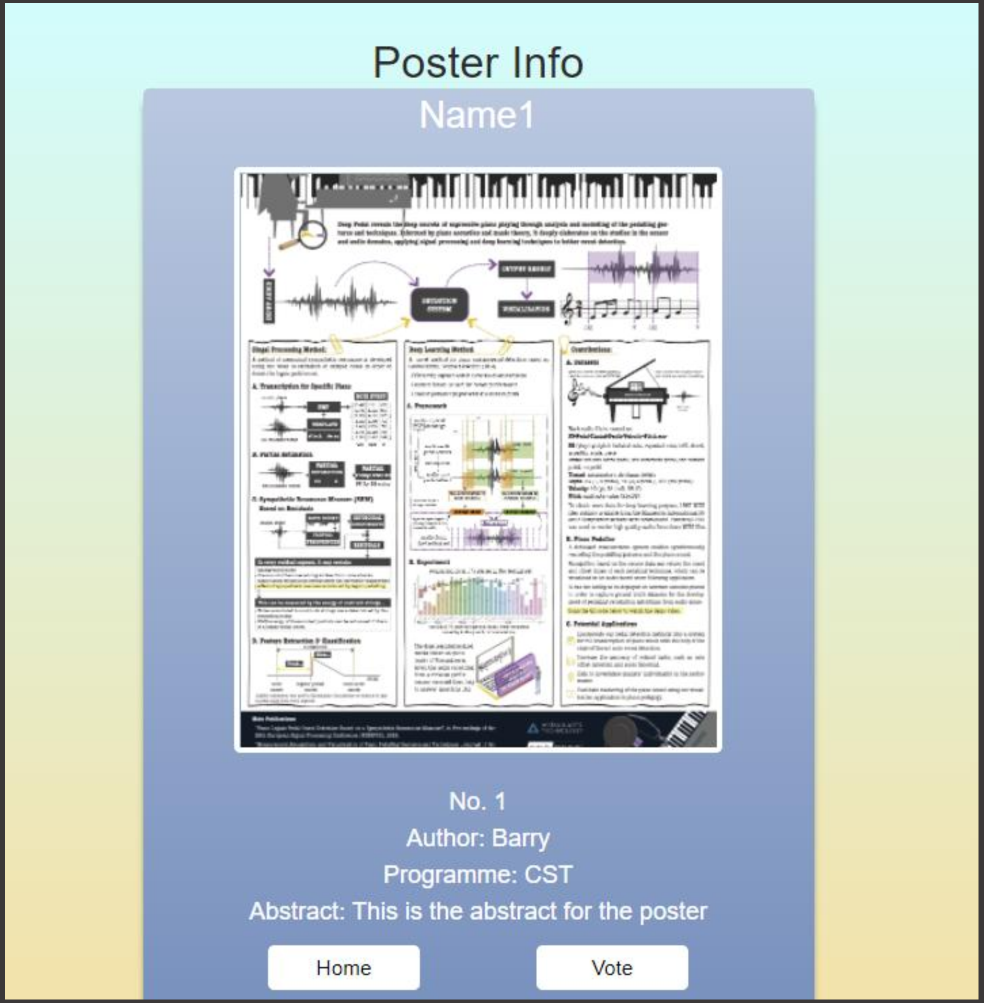
Written language:
4.3 43%

Oral presentation:
4.7 47%



```
var demo = (x1*0.25+x2*0.10+x3*0.20+x4*0.25+x5*0.30)*10;
```

Feature-Student



Poster Info

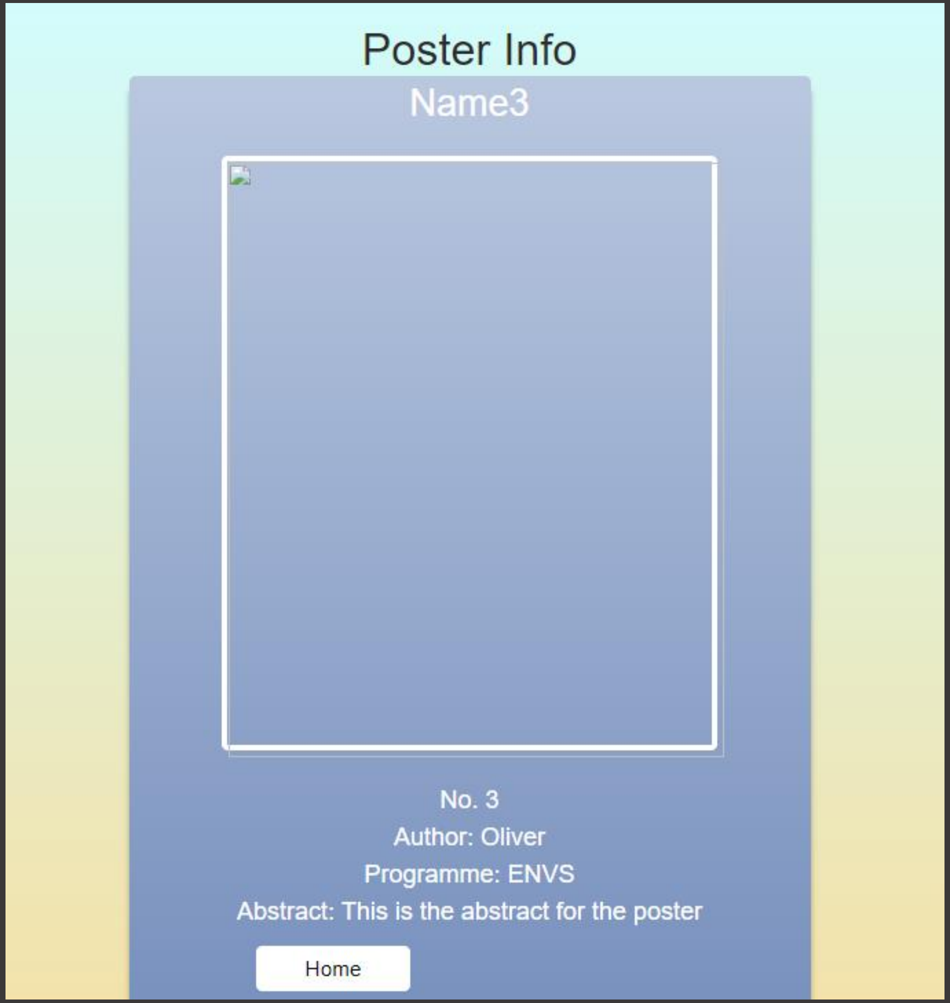
Name1

No. 1

Author: Barry

Programme: CST

Abstract: This is the abstract for the poster



Poster Info

Name3

No. 3

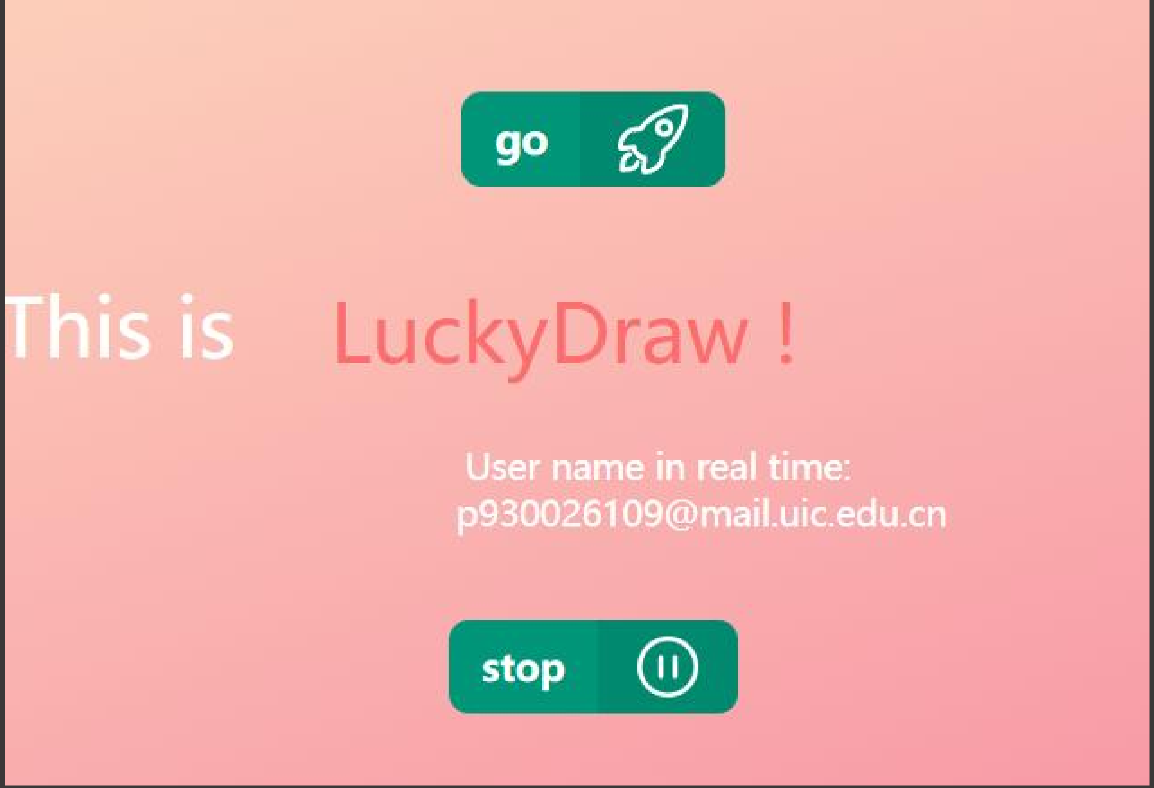
Author: Oliver

Programme: ENVS

Abstract: This is the abstract for the poster

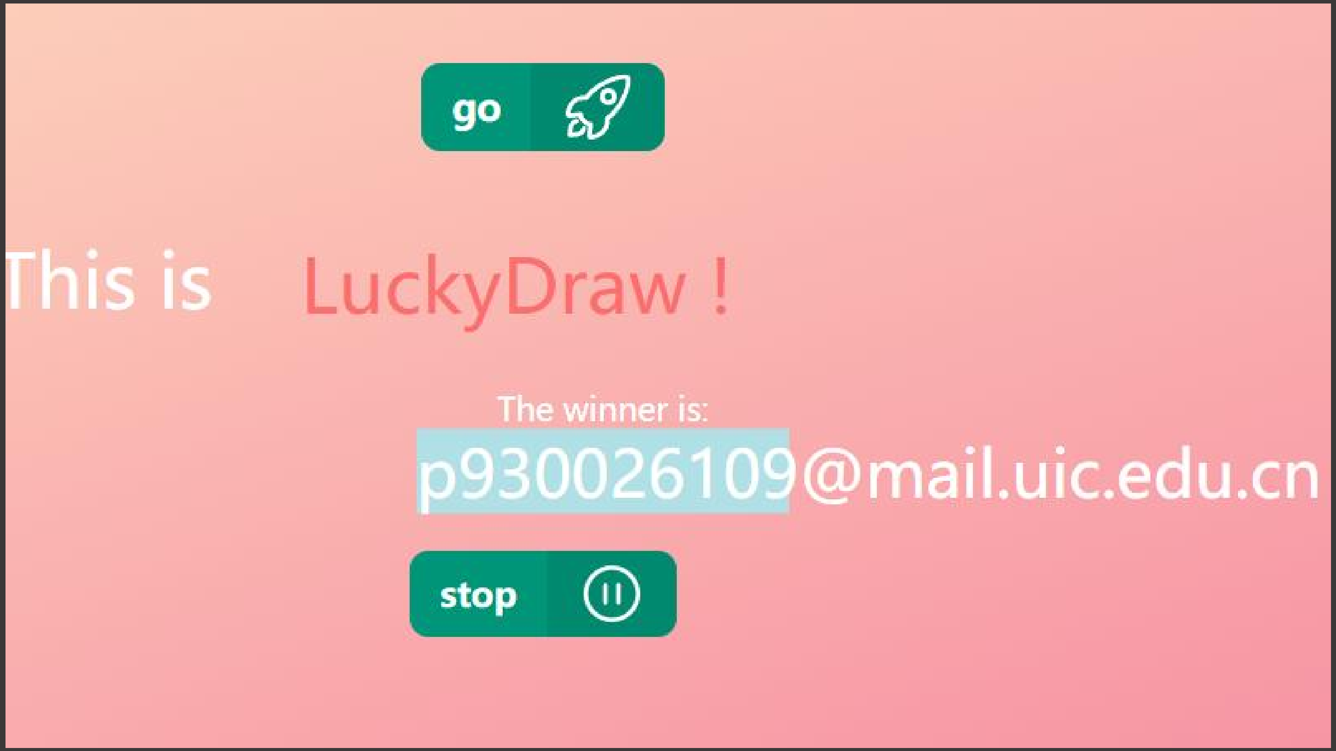
When students vote, our system will detect whether the students have finished voting. If they have already voted, they can't vote for others.

Feature-Luckdraw



This is LuckyDraw !

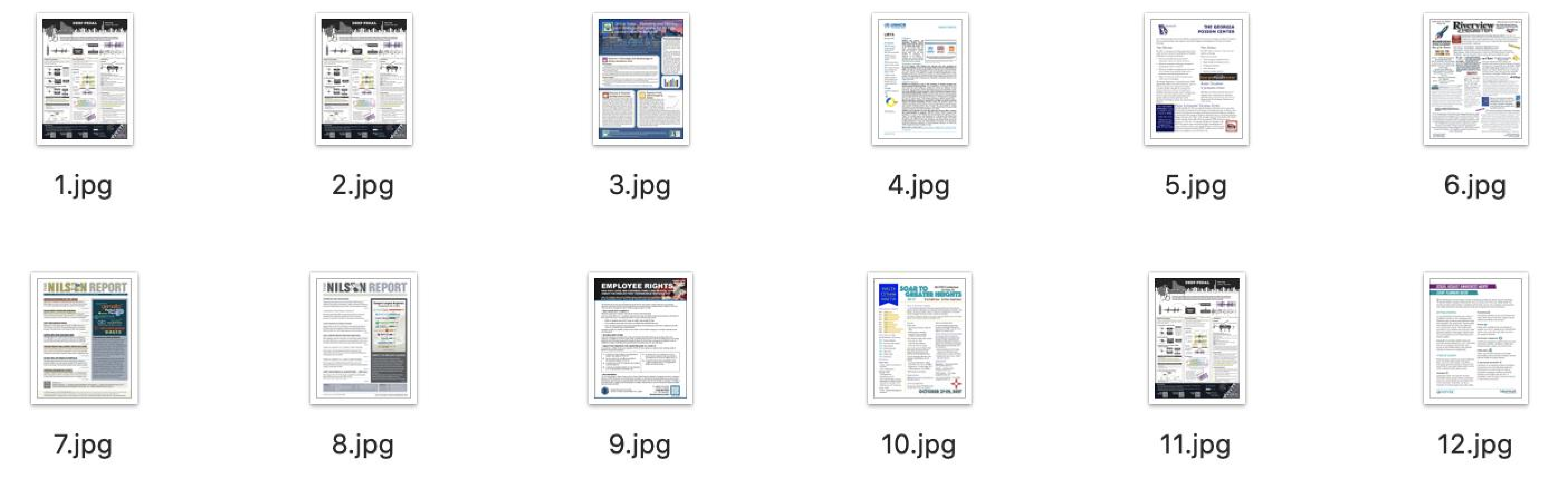
User name in real time:
p930026109@mail.uic.edu.cn



This is LuckyDraw !

The winner is:
p930026109@mail.uic.edu.cn

Feature-Poster



The poster photos we upload to the website will be files with various names, but the photos saved in the folder of our program after uploading will record the corresponding poster ID

04

PART FOUR

Demo

Barry

THANKS