**CS673 Software Engineering**

**Team 6 - IGroup**

**Software Design Description**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Shawn Hu | Team leader | *Shawn Hu* | 09/19/2022 |
| Dawei Yin | Design leader | *Dawei Yin* | 10/18/2022 |
| Siming Qian | Req leader | *Siming Qian* | 09/20/2022 |
| Alex Wang | Security leader | *Alex Wang* | 10/19/2022 |
| Haiyang Wang | QA leader | *Haiyang Wang* | 10/18/2022 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **Iteration 1** | **Team 6** | **10/20/2022** |  |
| **Iteration 2** | **Shawn,Dawei,Alex** | **11/11/2022** | 1,database design  2,system architecture  2,description |
| **Iteration 3** | **Shawn,Dawei** | **12/10/2022** | Remove Redundant Information |

[Introduction](#_87t9hln2vjz0)

[Software Architecture](#_3ipvmjgn6clp)

[Class Diagram](#_ky60nv8suxxm)

[UI Design (if applicable)](#_7ucksmkf6rzx)

[Database Design (if applicable)](#_tcmuor4nl1kz)

[Security Design](#_x18fj36s1121)

[Business Logic and/or Key Algorithms](#_mtfbusfb0eq3)

[Design Patterns](#_9zvwkmc4luo5)

[Any Additional Topics you would like to include.](#_15tmymhipvdv)

[References](#_50ojo9i46ytq)

[Glossary](#_8n34lvocupub)

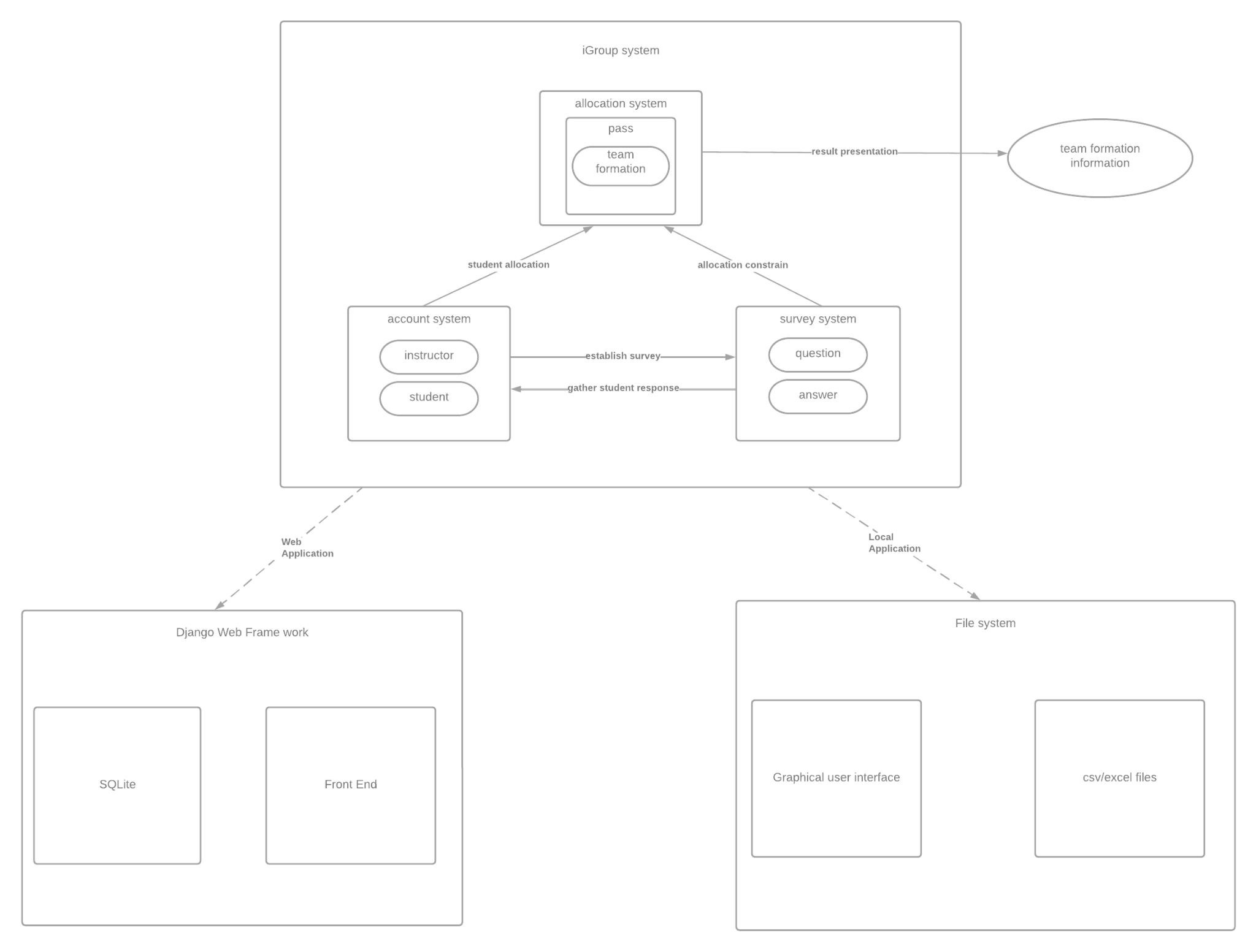
# Introduction

In this section, give an overview of this document, and also address the design goals of your software system.

The following is the design plan of our software. We want to create a professional web application focused on making perfect teams, both in terms of running speed and UI feeling, we want to create a relaxed and efficient appearance. Algorithms are the main goal to be overcome in our project. Based on the literature "team-maker" we found, we improved it, so that the original survey questions can be designed more flexibly, and each parameter can be perfectly coordinated together to produce more valuable information.

# Software Architecture

In this section, you will describe the decomposition of your software system, which includes each component (which may be in terms of package or folder) and the relationship between components. You shall have at least one diagram to show the whole architecture of . The interface of each component and dependency between components should also be described. If any framework is used, it shall be defined here too.

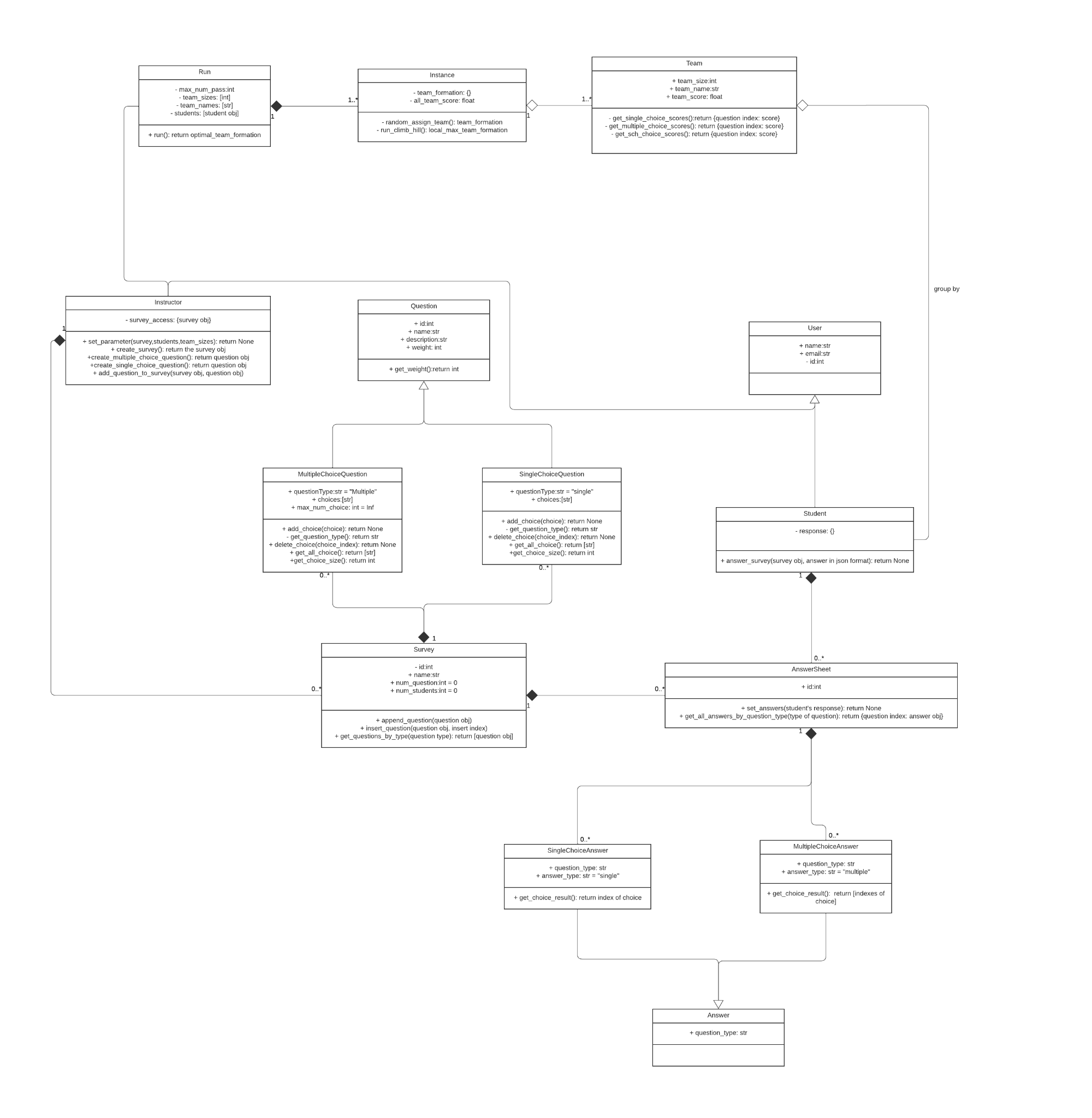


In our iGroup application, Since we have adopted the agile methodology as the development process, we mainly focus on developing the local application at this iteration, with using local file systems as our persistent data storage.

Within the iGroup system, the user needs to use the account system to gain access to create surveys and send the survey to students, which the student responds to and creates the student account in the system. Then the instructor can gather the response from the students and call the allocation system to do the team allocation corresponding to the parameter that instructor assigned before the allocation.

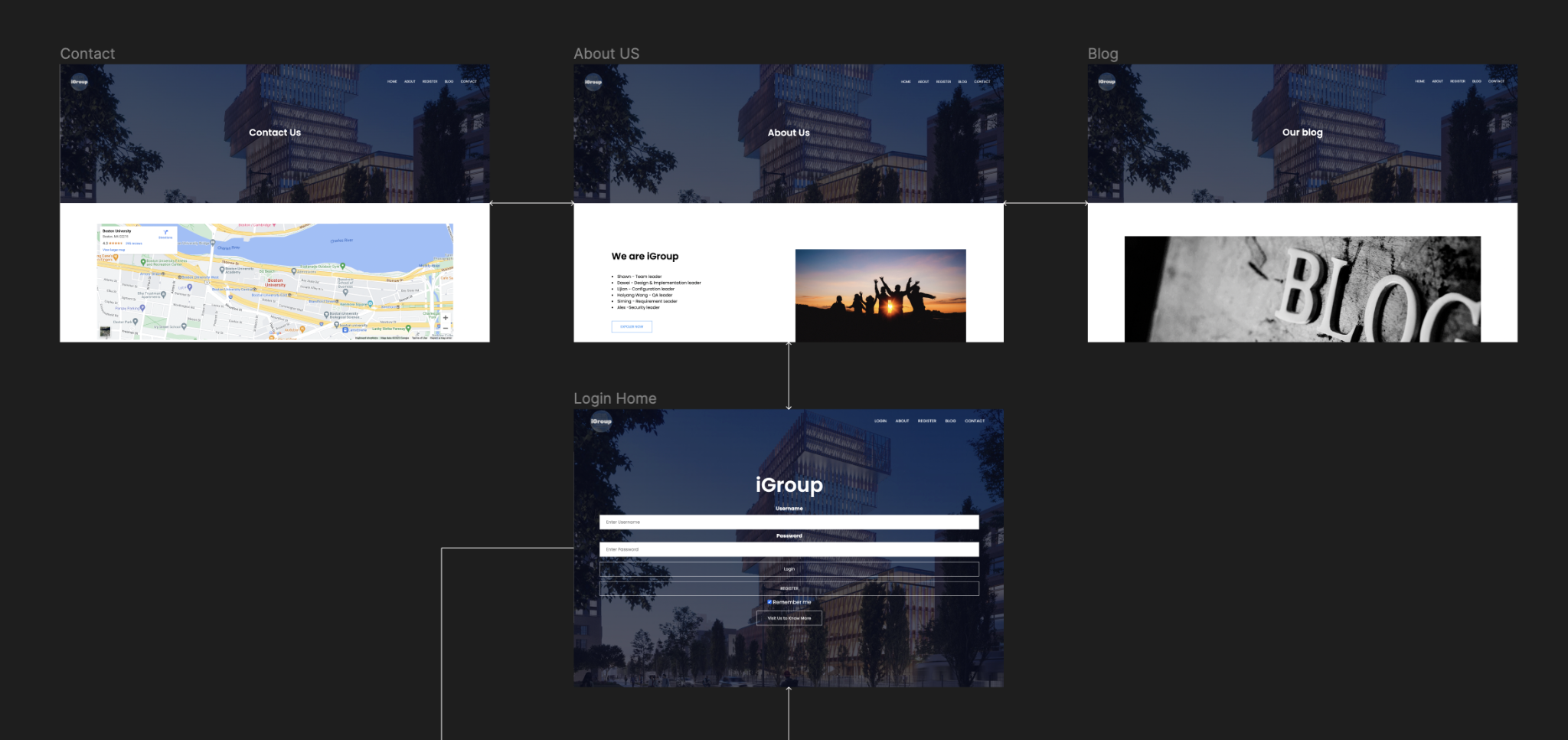
* Class Diagram

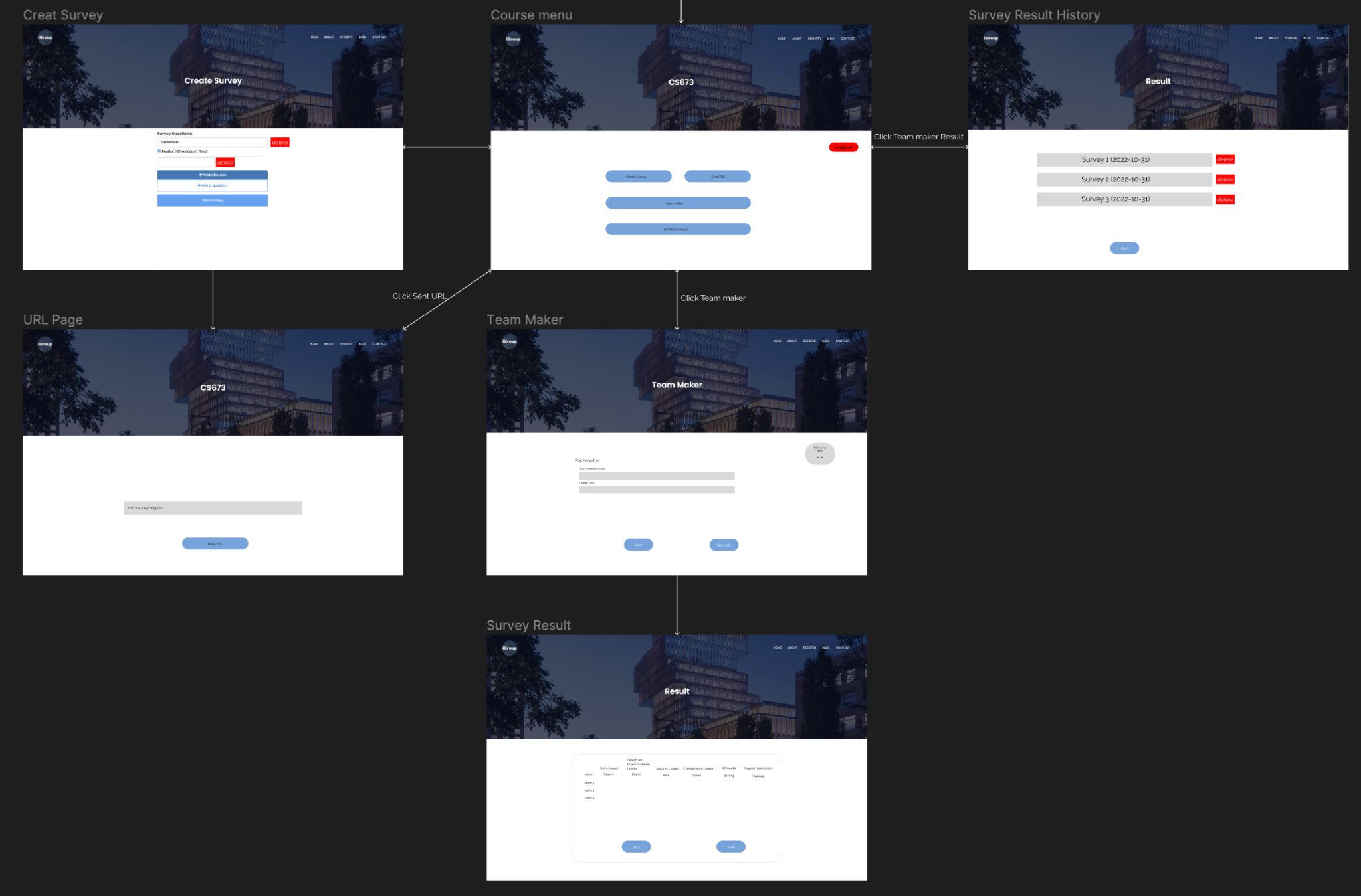
In this section, you will provide a detailed description of each component (or package) and use one or multiple class diagrams to show the main classes and their relationships in each component.



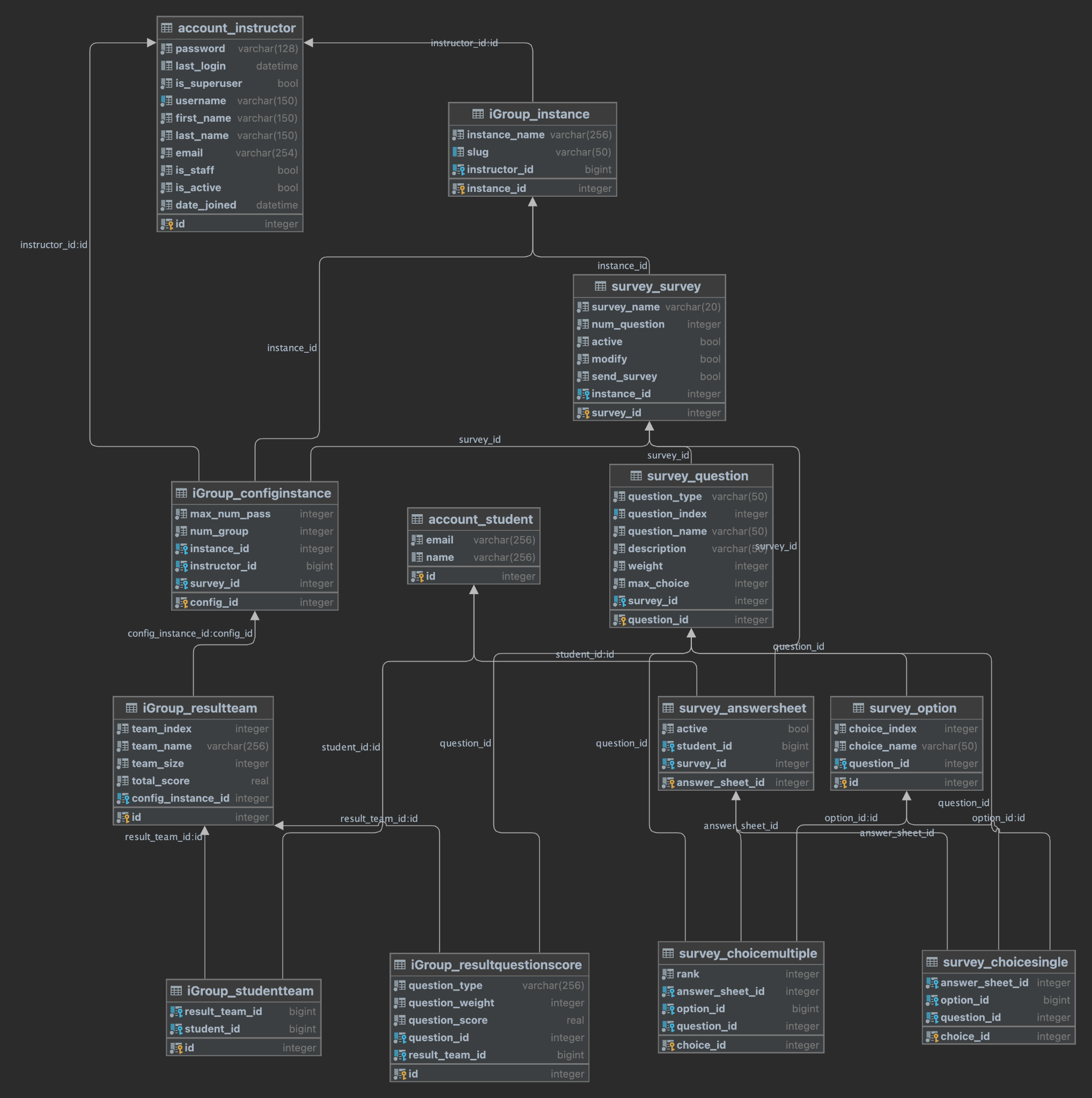
# UI Design (if applicable)

In this section, you can describe your UI design



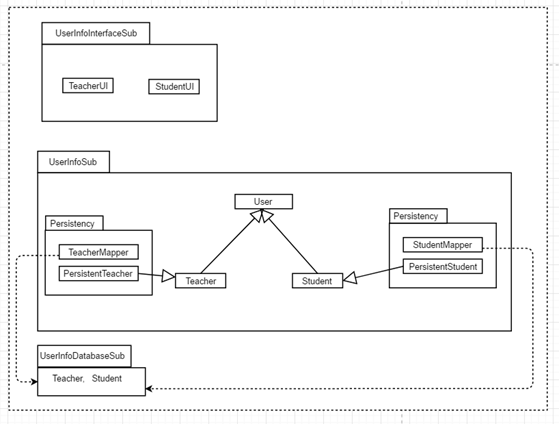


# Database Design (if applicable)

The updated Database:

We divide the system into 3 subsystems, which are UserInfo Subsystem, FormGroup Subsystem and Survey system. Userinfo Subsystem is to record two kinds of user information, FormGroup Subsystem is to form a group,calculate group survey answer score, and Survey system is to import survey answers and export score for a group.

* UserInfo

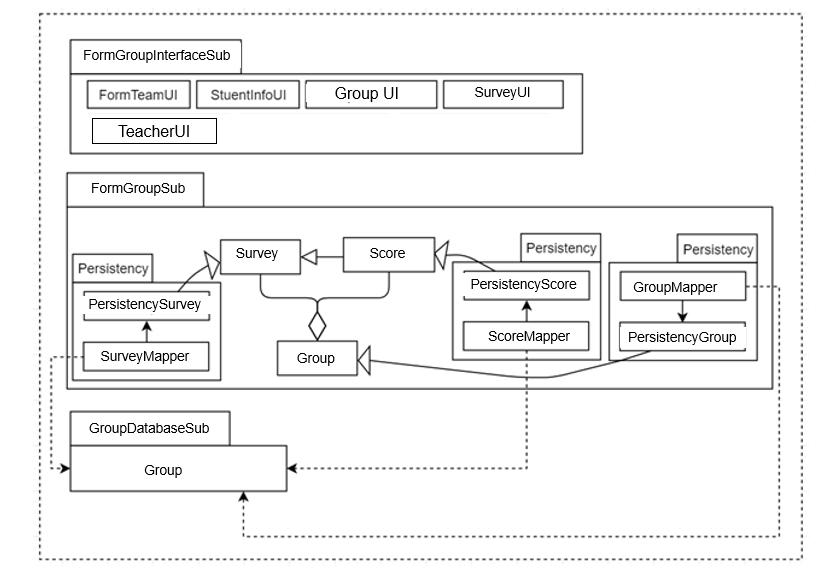


* Database

| Teacher | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Teacher\_Name | Teacher\_ID | Course\_Name | Course\_ID | Password | Email | Survey\_Question |
|  |  |  |  |  |  |  |

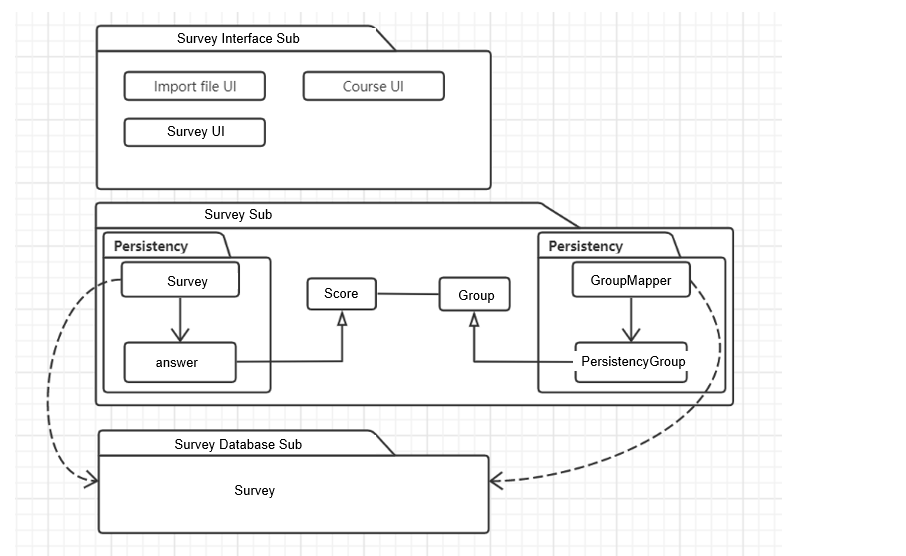
| Student | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student\_Name | Student\_ID | Course\_Name | Course\_ID | Survey\_Anwser | Group\_ID | Password | Email |
|  |  |  |  |  |  |  |  |

* Form Group



| Group | | |
| --- | --- | --- |
| Course\_ID | Group\_ID | Group\_Score |
|  |  |  |

* Survey system



| Survey | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Survey\_ID | Teacher\_Name | Survey\_Answer | Survey\_Question | Course\_ID | Course\_Name | Student\_ID | Student\_Name |
|  |  |  |  |  |  |  |  |

One of the subsystems is called UserInfo Subsystem which includes User, Teacher and Student.

The relationships among each entity are based on our class diagram which we have shown at section 2.1. The categories and number of UI based on the UI which we have shown at SRS document.

# 

# Security Design

# Security Test plan

1. Objectives

1.1 Modules under test

Python unittest

1.2 References

2. Testing levels and methods

2.1 Testing levels

Unit test

2.2 Testing methods

Assert base testing

3. Test cases

| Test ID | Test case description | Expected result |
| --- | --- | --- |
| 1 | Answer sheet init | Successfully init |
| 2 | Question init | Successfully init |
| 3 | Survey init | Successfully init |
| 4 | Team init | Successfully init |

# 

# 

# Business Logic and/or Key Algorithms

In this section, you shall describe any key algorithms used in your software system, either in terms of pseudocode or flowchart, or sequence diagrams.

For the process of the algorithm, the first step is to randomly assign all the students into different groups, and then calculate the score of each team (the part of calculating the score here uses the team-maker algorithm). After the score of each group is obtained, the group with the second highest score and the group with the lowest score will swap the team members and calculate the score again, until the new score of both groups has improved, and then confirm to switch the team members. After 20 rounds of this, all the groups were formed. (The Gale-Shapley Algorithm used here)

**Detail of Algorithm**

* **Score Calculate**

Team-Maker supports four types of questions:

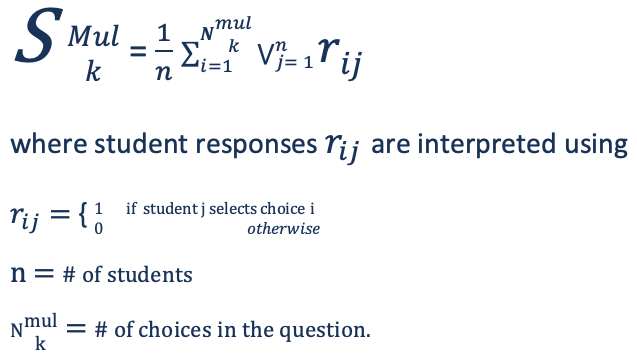
1. Multiple-choice

2. Choose-any-or-all-of

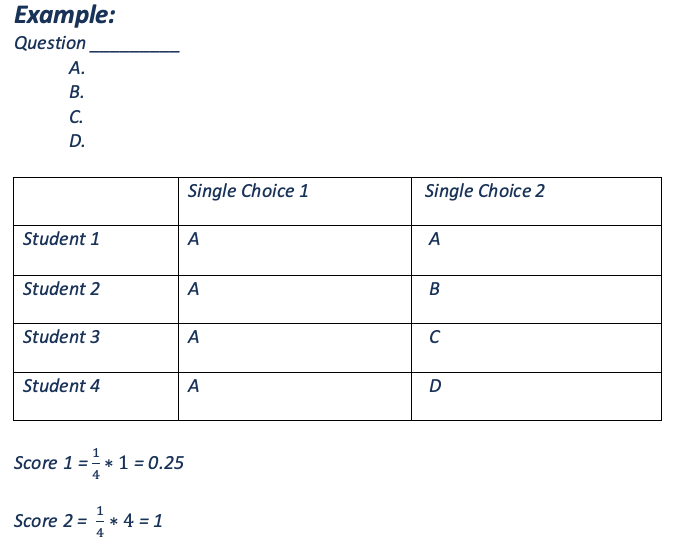
3. Schedule- compatibility

4. Underrepresented member

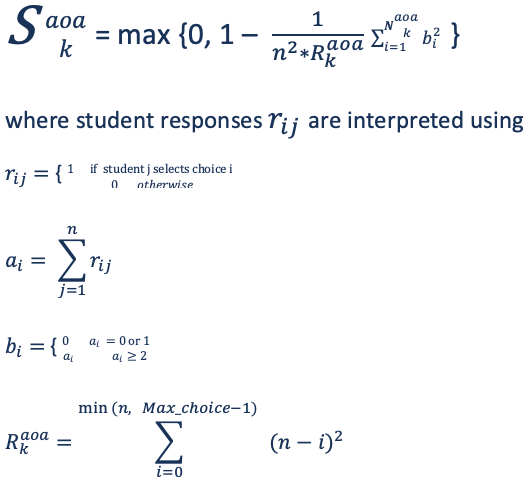
**#1 Multiple - choice Question**

****

The greater the number of options that the team members select in common, the closer the score approaches zero, indicating team homogeneity on this question.

****

**#2 Choose-any-or-all-of Question**

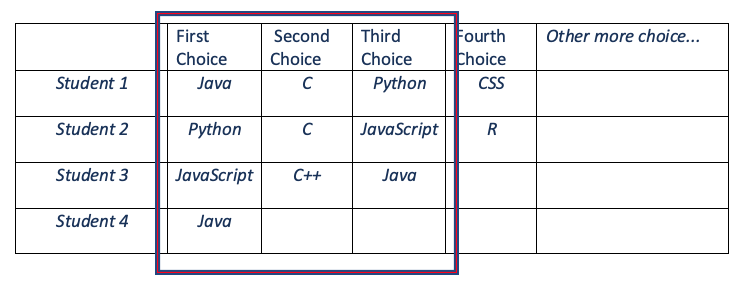
****

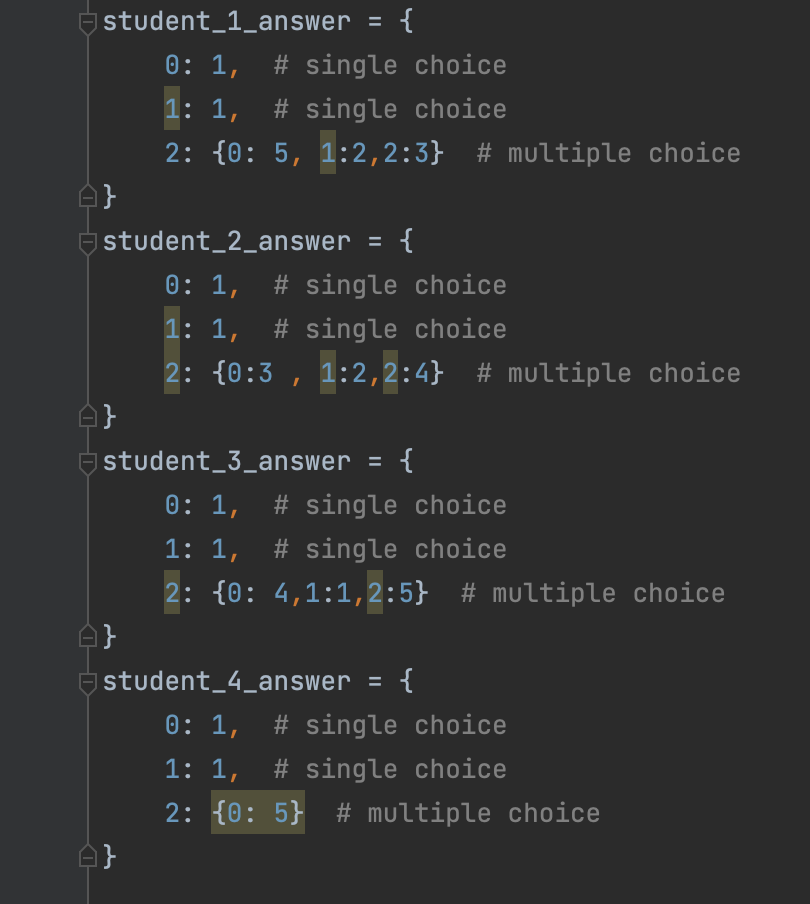
***Example #1 :***

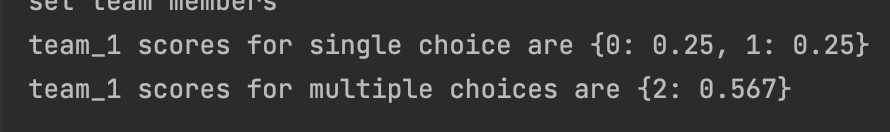
**Question: Please select the programming language you are best at in order**

**【 Java, Python, R, C, C++, JS, CSS, etc.】**

**Answer:**

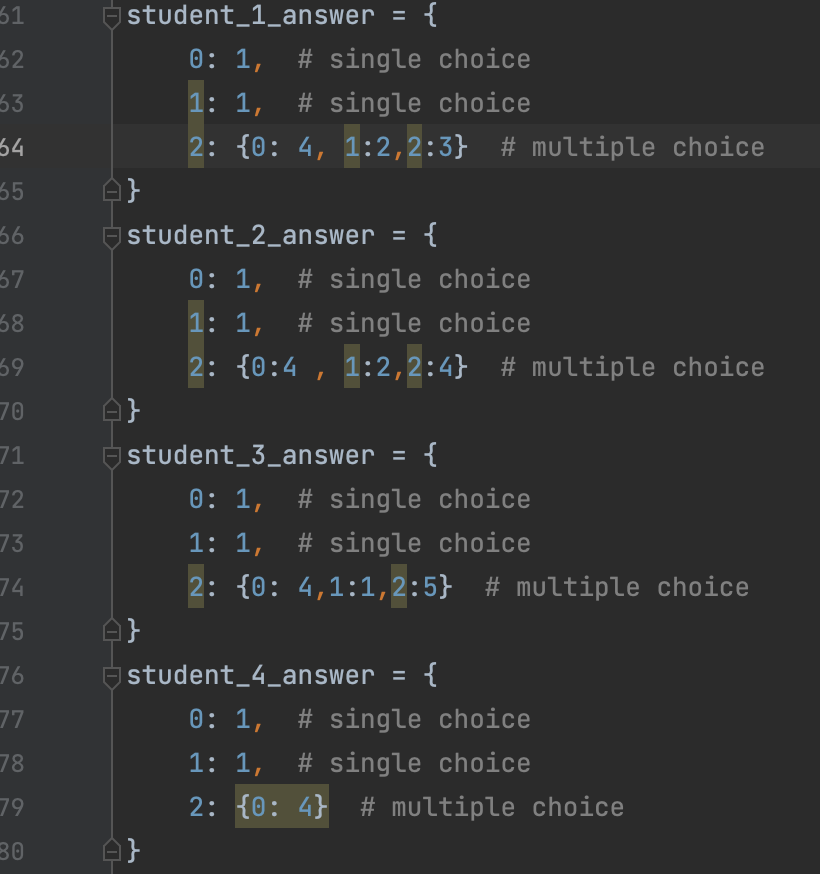
****

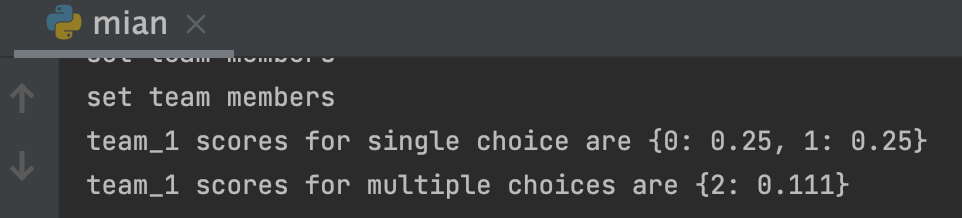
****

****

***Example #2 :***

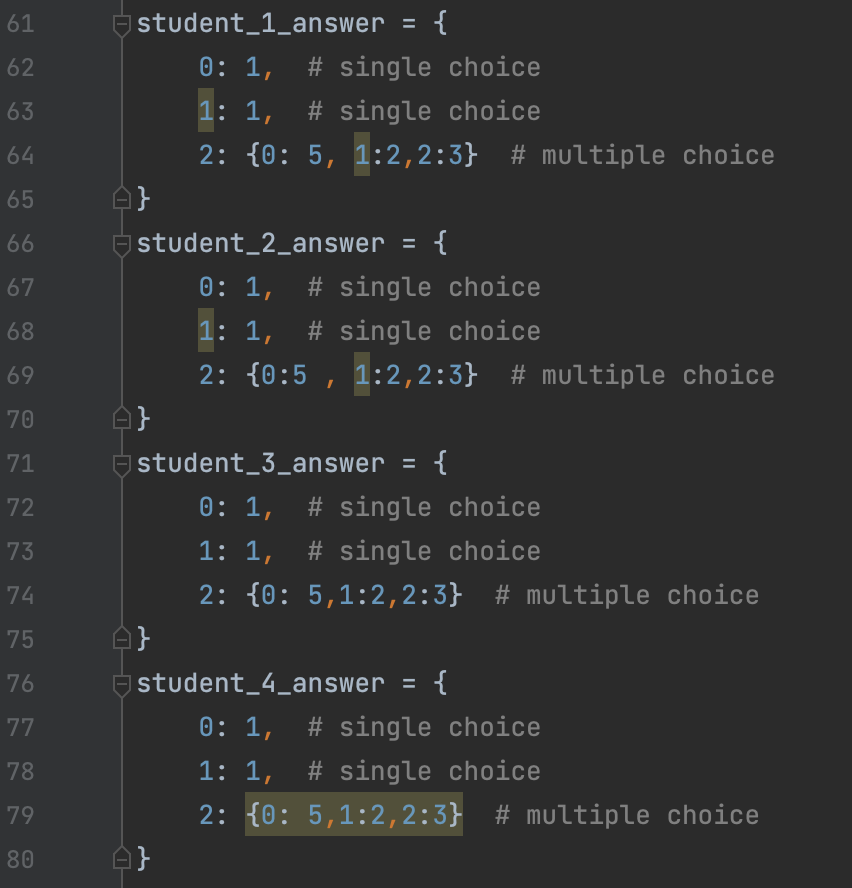
* **More similarity, since all student’s first preference is 4**

******

******

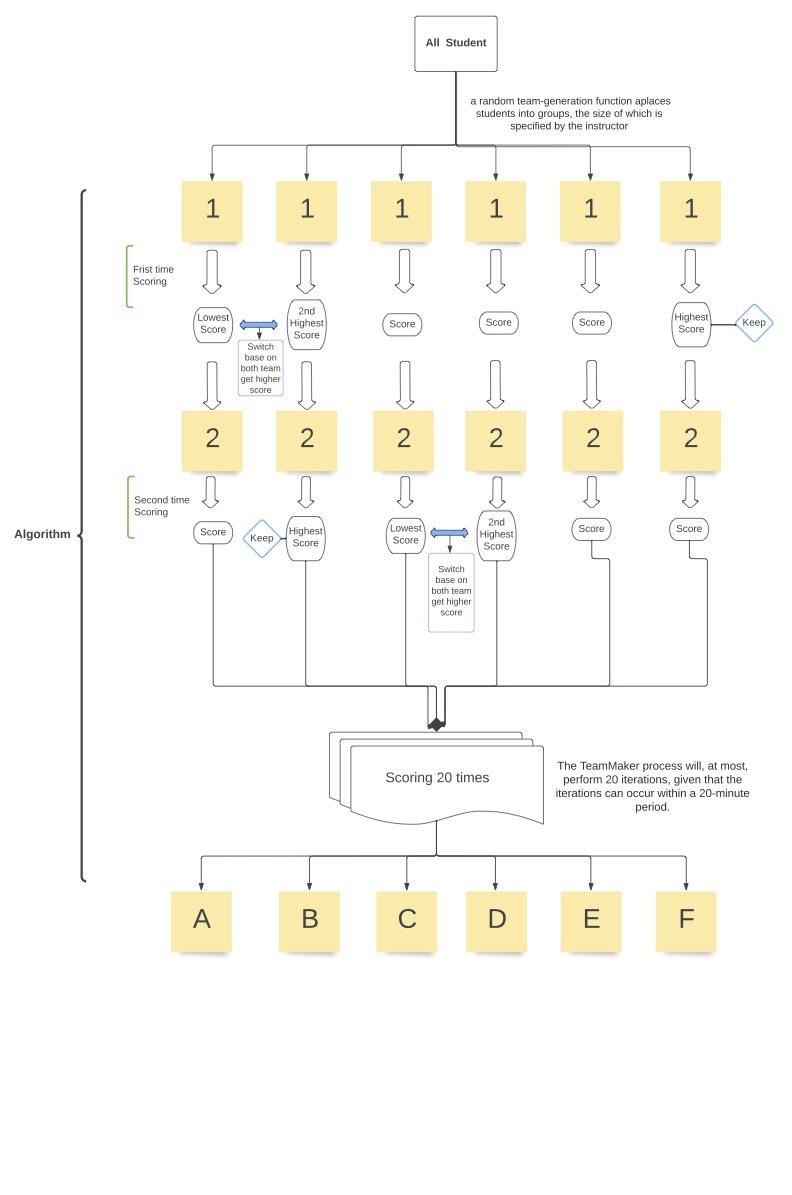
***Example #3 :***

* **max similarity, everyone in the team has the same preference**

****

****

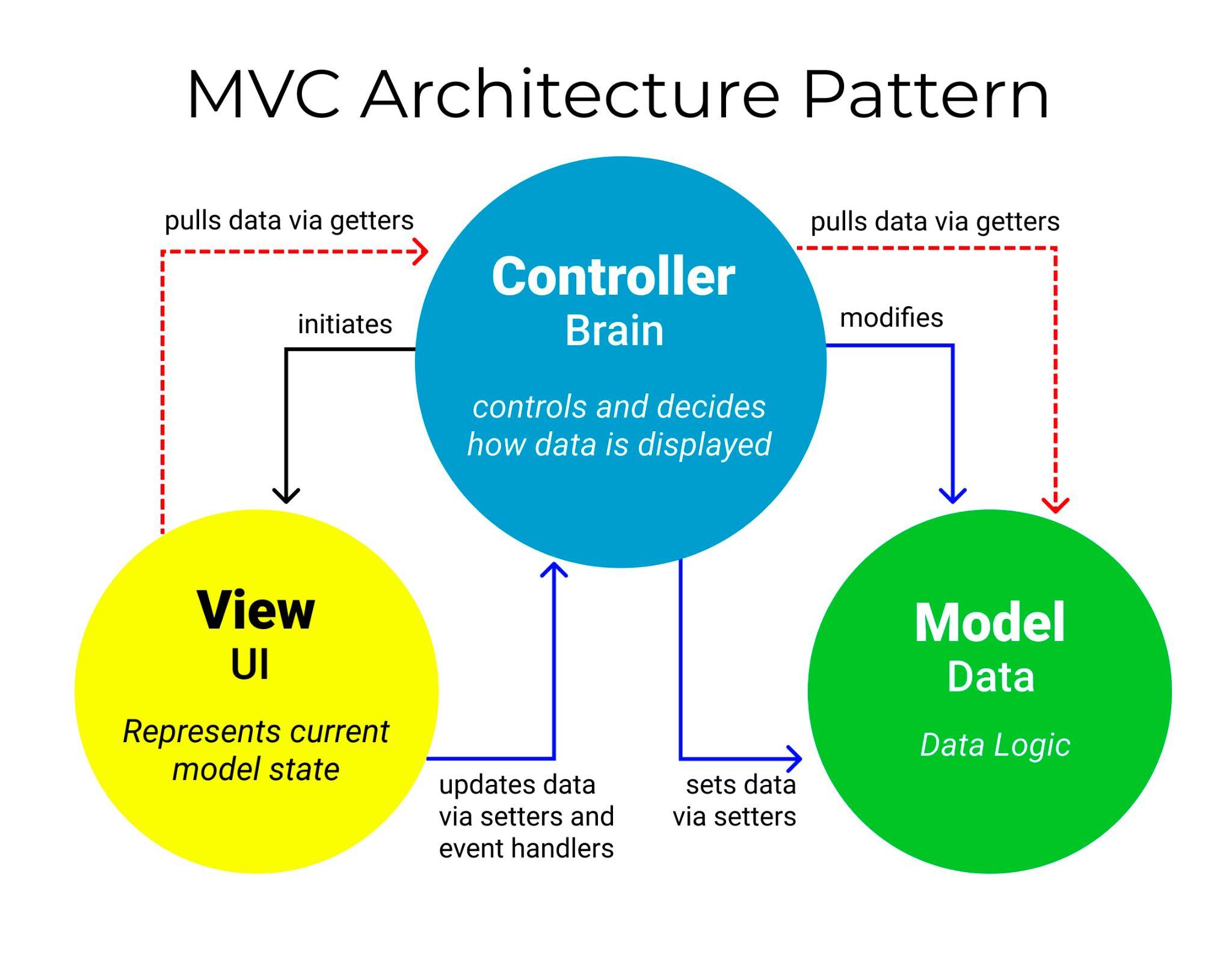
* **Swap Method**



* **Pseudo code**

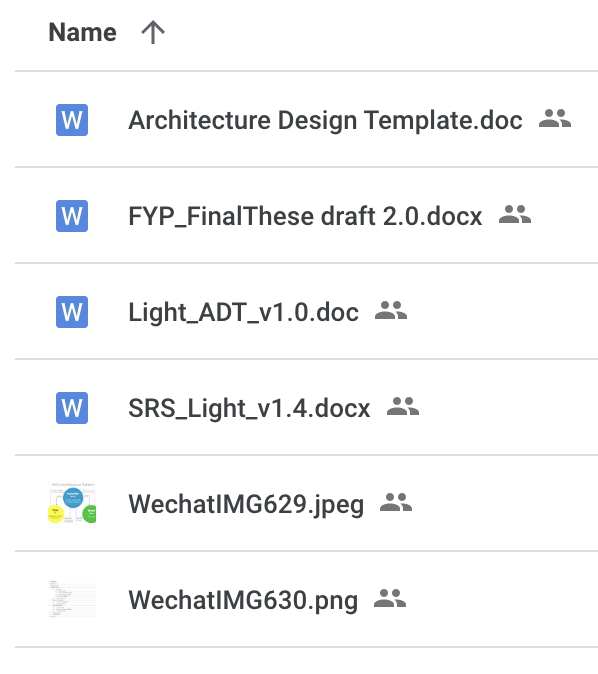


# Design Patterns



# References

In our team 6 google drive (reference)



# 