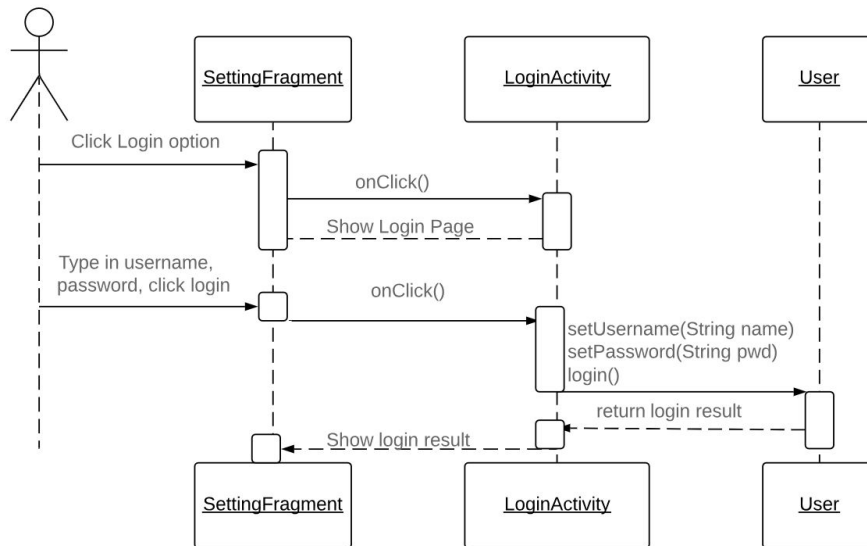


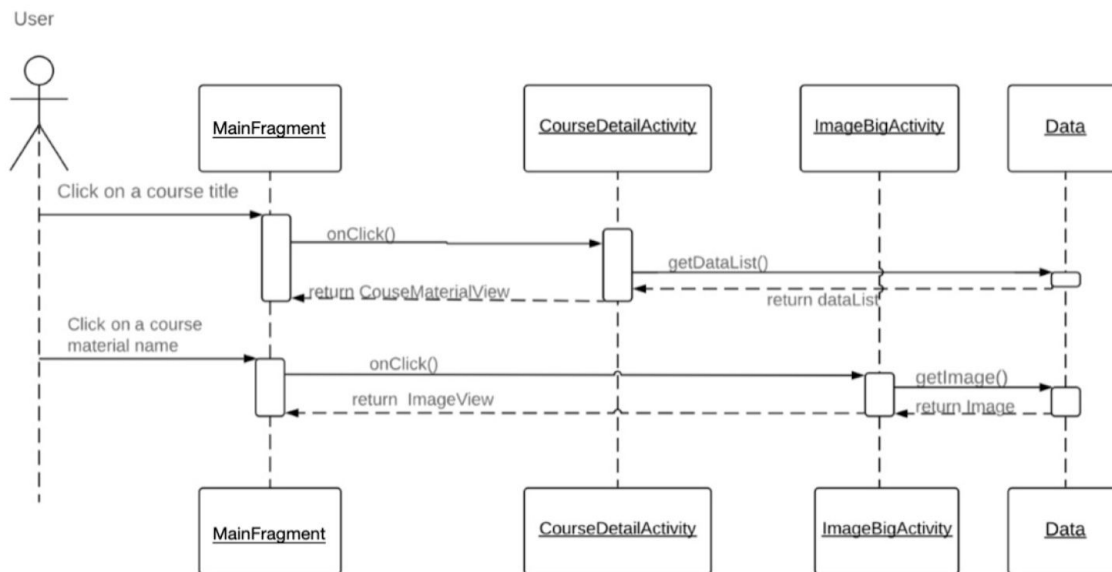
Interim Release - Five Girls

Sequence Diagrams

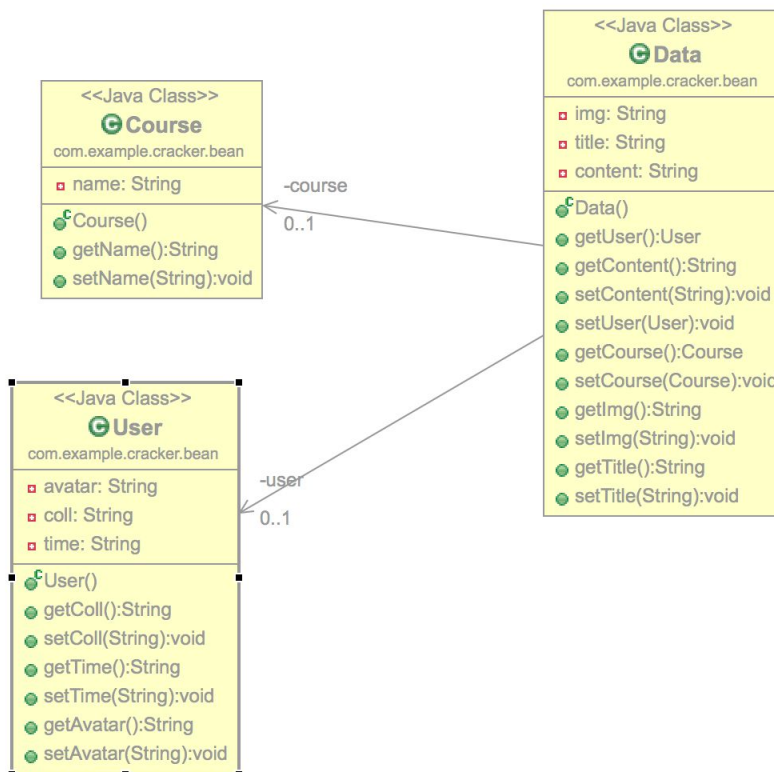
Sequence diagram 1 : UserLogin



Sequence diagram 2: ViewCourseMaterial



Static Class Diagrams



CRC Cards

ReleaseActivity(4.Procedural cohesion) BaseActivity	
<ul style="list-style-type: none"> • allow User to upload course-related image • allow User to update the image 	<ul style="list-style-type: none"> • Course (5 data coupling) • CourseDetailActivity (2 common coupling)

RegisterActivity(7. Informational cohesion) BaseActivity	
<ul style="list-style-type: none"> • hold the name and password for the User • allow User to enter the Name, password and set the Name, password for User 	<ul style="list-style-type: none"> • User(5.Data coupling)

LoginActivity(7.Informational cohesion) BaseActivity	
<ul style="list-style-type: none"> • Ask User for name and password • Check the name and password, if match, login success 	<ul style="list-style-type: none"> • User(5.Data Coupling) • RegisterActivity (5.Data Coupling)

CourseDetailActivity(5. Communication cohesion) BaseActivity	
<ul style="list-style-type: none"> • able login user to see the courses stored • store course data • user can open course related image • user can upload course image 	<ul style="list-style-type: none"> • Data(4 stamp coupling) • User(4 stamp coupling) • Course(5 data coupling) • LoginActivity(3.control coupling) • ImageBigActivity(1. content coupling) • ReleaseActivity(1. content coupling)

ImageBigActivity(7. Informational cohesion)		BaseActivity
<ul style="list-style-type: none"> allow the user to open the images list in CourseDetail 	<ul style="list-style-type: none"> Data(5. data coupling) 	

Abstract	(7.Informational Cohesion)	BaseActivity	App
		ImageBig CourseDetail Register Search Main Login Release	
<ul style="list-style-type: none"> Set the name of the application 			

	(6.Functional Cohesion)	SearchActivity	BaseActivity
<ul style="list-style-type: none"> Knows what text user put in Search for particular course 		<ul style="list-style-type: none"> Course 5.data CourseDetailActivity 2.common 	

	(6.Functional Cohesion)	MainActivity	BaseActivity
<ul style="list-style-type: none"> Control tab fragment 		<ul style="list-style-type: none"> HomeFragment 4.stamp SettingFragment 3.control 	

Design Approach

The design approach is based on the objects of our application. Our application has three main objects - courses, calendars and users.

In the design, we used two design patterns, model-view-presenter and adapter. The model-view-presenter pattern is used to build user interfaces. We had three classes corresponding to the model, view and presenter, which helping us implement the user interface. We used the adapter design pattern to allow two existing classes together without changing the source code.

And, our implement adheres to SOLID principles.

Single Responsibility Principle – SRP

All the class, function, module in our design only focus on one concern. We implemented class Image big, which only focus on open the material image. Also, we implemented the welcome page class, which only focus on the app turn on a moment.

Open-Closed Principle – OCP

All the modules are well defined and allow extension. For instance, the base activity class, we do not need to change the source code, but still can extend new classes from it.

Liskov Substitution Principle – LSP

Derived types must be completely substitutable for their base types. In our design, the search activity is the subtype of the base activity. The search activity doesn't change the behavior of the base activity, it just extends the base activity with more functions.

Interface Segregation Principle – ISP

We divided our interface into many small sections: calendar, my courses, courses material, and login, logout interfaces, which allow the client only need to know the method they interest about. For example, if the client wants to refactor, change or redeploy the calendar interface, they only need to learn the method of calendar section.

Dependency Inversion Principle – DIP

In our design, both lower-level modules and higher-level modules are depend on abstractions. The base activity, which is the higher-level class, and the Course Detail Activity, which is the lower-level class, are both depend on abstraction.

Since our application is an android application, so we are coding in Java. The IDE we used is Android Studio, which is a primary IDE for native Android application development. For storing the pictures of the backtests and user information, we use the Bmob cloud service(<https://www.bmob.cn/>), which is a cloud service platform. We have used this platform before and found it quite easy to use and suitable for our application. Using Bmob can virtually eliminate the majority of server-side coding workloads. It provides a visual cloud data table design interface to easily build a library which is useful for storing user information. Its file services can store information such as pictures, videos, audio, and documents which fits our requirement of storing backtest as pictures.

Contribution Summary and Status Report 5%

Contribution Summary:

YiRan Zheng(Feli)

- Draw the static class diagram
- Organize CRC cards
- Read over the other work and suggest improvement/modification.
- Implement status report

Zhou Lu(Winnie)

- Draw the static class diagram
- Organize CRC cards
- Read over the other work and suggest improvement/modification.

Sylvia Hua

- Work on the upload feature
- Make code more readable
- Draw the sequence diagram
- Read over the other work and suggest improvement/modification.

Chenhao Pan

- Write the design approach - figure out the design approach, design pattern and the technologies used
- Read over the other work and suggest improvement/modification.
- Help with the test of the implemented function

Jennifer Fu

- Work on the design approach
- Read over and organize the work

Status Report:

We are finishing sprint 6 and moving forward to sprint 7. We have completed all the features we promised in sprint 4 deliverables, though some of them still need improvement. The mobile-end pages are done, with all the necessary UIs on it. We will focus on the calendar feature in the next two sprints. Currently, our code does not have many comments, so we are going to add more in order to make the code more understandable.