StudyPal

STUDYPAL

Final Group Report

Yanbo Zhao (yzhb505)
Siqing Zhang (szha490)
Shuhao Gao (gshu666)
Zhuoyang Li (zli316)
Ruiqi Xu (rxu385)

Final Vision
Prepared by*: Team18 – "Huh?"*

2023 S2
COMPSCT399 CAPSTONE PROTECT

OUR TEAMS

Yanbo Zhao
yzhb505@aucklanduni.ac.nz

Back-end developer, Quality assurance

Siqing Zhang
szha490@aucklanduni.ac.nz

Front-end developer, Quality assurance

Shuhao Gao
gshu666@aucklanduni.ac.nz

Back-end developer, Quality assurance

Zhuoyang Li
zli316@aucklanduni.ac.nz

Team Leader, Full-stack developer

Ruiqi Xu
rxu385@aucklanduni.ac.nz

Front-end developer, Researcher, Resource Manager

1. Executive Summary

With the rapid development of education and the continuous advancement of technology, students are facing more and more academic challenges and pressures. Traditional learning methods and resources, such as library self-study, may no longer be able to meet the learning needs of most students. Therefore, students urgently need a more convenient and personalised platform to find academic help that suits them. At the same time, some students who have advantages in certain fields are eager to share their professional knowledge and help those who encounter academic difficulties. However, traditional educational environments do not provide sufficient opportunities for these two groups to effectively connect.

Our project "StudyPal" is developed to meet the needs of users in the context of the rapid development of modern education, and aims to provide an innovative Android application. The purpose of this application is to provide a bridge between most students who are determined to have an academic advantage and who are willing to help (which will be called tutor in subsequent reports) with students seeking academic help.

1.1 Technical implementation:

1. **Front-end development:** in the front-end development part, the front-end team chose Flutter as the core framework of front-end development. Based on the premise of this open-source UI software development kit provided by Google, "StudyPal" has excellent cross-platform performance and a high degree of consistency in the use experience on different devices. In addition, the high-quality plug-ins and flexible design mechanisms provided by Flutter allow us to provide users with a highly functional and aesthetic interface.

2. **Back-end development:** the back-end development choice is developed in Java language and based on the Spring Boot framework. This framework allows us to quickly build an efficient and stable back-end service. In addition,

our back-end team also chose MySQL as the database system to ensure that data storage, query and management can run efficiently and stably.

3. **Human-computer interaction design:** in the UI design process of "StudyPal", we strictly abide by the design principles of gestalt principles, balance, emphasis, unity and so on. The functionality and design logic of each page have undergone rigorous testing to ensure that users can do so intuitively, providing an excellent user experience.

1.2 Core values of the application:

StudyPal not only implements a common academic help platform but also provides a two-way connection between students and mentors. Students can get help on the platform according to their own actual needs, while other students who have advantages in the academic field can also become mentors to provide help and knowledge sharing. The design of this mechanism not only promotes the dissemination of knowledge but also creates an environment and atmosphere for students to learn from each other.

1.3 Conclusion:

After considering the related technology implementation and functional value, we firmly believe that our whole team has the software development capability of the project, as well as experience in deploying relevant application scenarios. At the same time, we have assigned roles to different team members that will help speed up the entire project development process. I believe that through our team's technical experience and thirst for unknown knowledge, we will be able to do the research and development of this project.

2. Table of Contents

3. Introduction

3.1 Aims and objectives of our project

The project aims to provide an App for students at the University of Auckland, so that students who need academic help can find mentors who meet their needs, and so that top students who are willing to share their talents can help those who are in trouble. The program allows students to take what they need and creates a platform for students to progress and grow together.

The specific goals of the project are divided into three categories:

1. Through Flutter, a beautiful interface layout that conforms to human-computer interaction logic is provided.
2. Good and stable back-end API functions ensure the operation of the core functions of the project.
3. Powerful database storage ensures the timeliness and stability of information released by all users.

3.2 The target audience of the project

The target audience of the project is very wide, from secondary school students who are proficient in using mobile phones to graduate students and doctoral students in university institutions. While the current app settings are limited to Auckland University students, they change over time and our APP will keep updating at this high rate, adding more different learning stages to the curriculum settings to cater to more students in different stages or fields.

3.3 Scope of the project

Front end：

1. **UI design:** Using Figma to design high-fidelity prototypes to ensure that the interface and user interaction process are optimized.
2. **RESTful API integration:** The CRUD operation of data is realized by interacting with the backend through HTTP requests.
3. **Error and exception handling**: Use try-catch statements to handle potential errors in asynchronous operations gracefully.
4. **Front-end performance optimization**: Use code segmentation and caching policies to reduce loading time.

Back end:

1. **RESTful API design**: Design and implement API according to REST architecture style to meet front-end requirements.
2. **Authentication and authorization**: Use tokens for user authentication and secure access to API.
3. **Server performance monitoring:** Monitor server performance and system health.
4. **Error logging**: Log back-end errors and exceptions.

Database:

1. **Relational data model:** Use ERD (entity relation Diagram) to design appropriate data structures and associations.
2. **Transaction processing and ACID principles:** ensure atomicity, consistency, isolation, and persistence of database operations.
3. **Index strategy:** Create and adjust indexes reasonably according to query frequency and performance requirements.
4. **Data backup and recovery:** Automatic backup strategy and ensure that data can be quickly restored in case of emergency.

3.4 Approach Used in

In our project planning, we used Flutter to build the front end and Spring Boot to build the back end to build the entire App. It allows us to create more efficient code and faster user front-end interfaces. The entire project is developed based on Dart and Java, and we also decided to use Mysql as our database system.

The programming cycle of the entire project is about 10 weeks. We chose to use the project management software Monday.com to track the progress of the project in real time, assign tasks, and monitor the status of each task. At the same time, we adopted the agile development methodology scrum to promote rapid feedback loops and continuous delivery. After each sprint, we will conduct a retrospective meeting to summarize lessons learned and adjust the strategy and direction for the next phase. Finally, a GitHub repository is used for version control and continuous feature integration.

3.5 Important outcomes

"StudyPal" has been carefully designed and developed to achieve the overall goal of finding ideal learning partners and improving communication through the functions mentioned above. Collaborative learning is more convenient and effective because the user-friendly design is easy to use and search for students and teachers. With the help of the software, users will be able to identify ideal learning partners regardless of their progress in professional courses.

4. Background

4.1 Preliminary background

Today, with the continuous change in the global educational background, the learning guidance mode of case-by-case has gradually become a key demand. So far, the solution to this demand has mainly focused on establishing a connection between students and appropriate educational resources. According to statistics, about 60% of students have encountered difficulties in their student life (Demirbilek, N). However, in the context of traditional educational resources and systems, it is impossible to provide enough support for every student. Therefore, many students are easy to fall behind academically.

As the Internet model evolves, the usage of applications in the Internet context increases exponentially (Fettweis, G). However, most of the educational content available today is not innovative in nature, such as simply uploading records that students can watch repeatedly. At the same time, these changes are not innovative enough to provide students with personalized learning needs and difficult solutions, resulting in many students feeling disappointed.

For this project, we hope that the "StudyPal" application will become an useful part of students' daily learning in the popularity of smartphones. Let StudyPal match students with the right mentors and provide students with customized learning advice, which will create great value for students and help them achieve better academic results.

4.2 Research

In some studies conducted by Akyuz on the impact of online personalized tutoring on students' academic performance, they explored how personalized tutoring can improve academic performance (Akyuz, Y.). Traditionally, when students encounter academic difficulties, they usually choose to consult the instructor or consult the course lecturer in the limited class time. The study explores the benefits of personalized tutoring provided by online education platforms such as "StudyPal" to improve academic performance and efficiency.

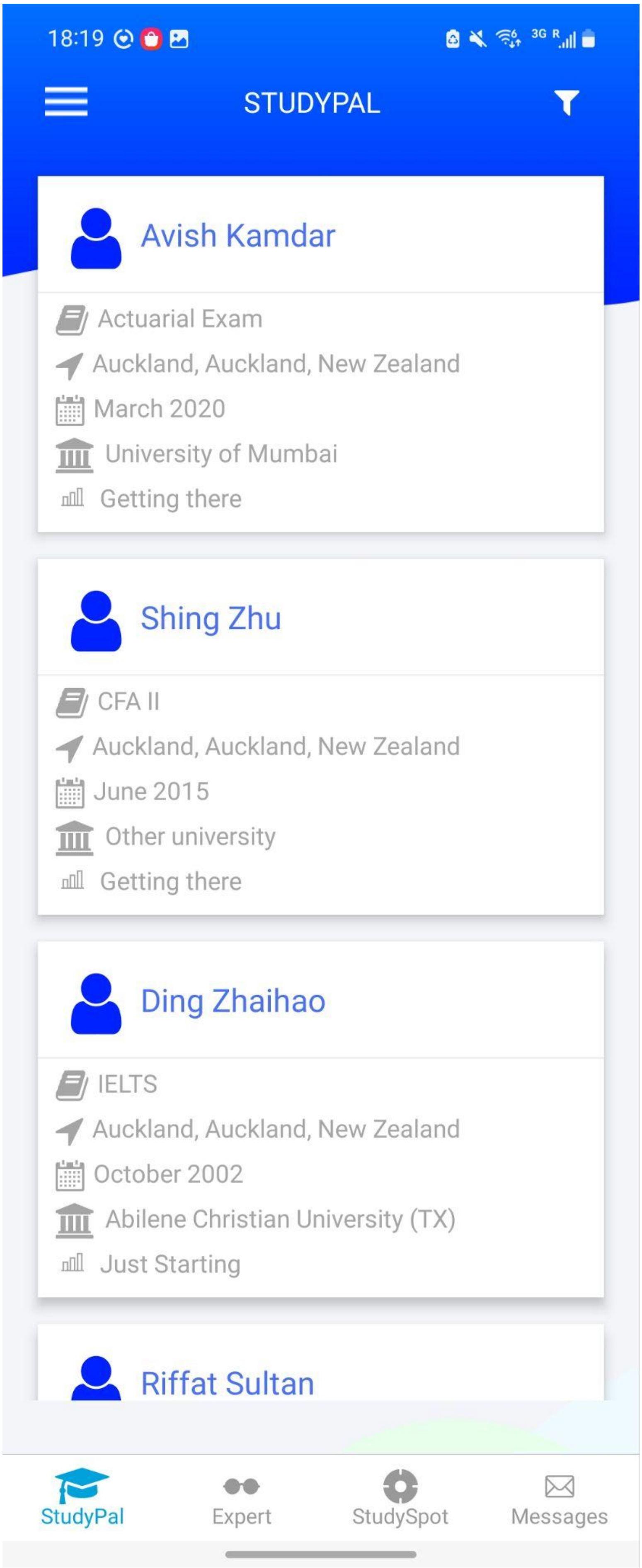
4.3 Existing solutions

At present, there are many applications on the market trying to solve similar tasks, but there is still a huge gap in the market when it comes to the two-way connection between mentors and students.

The StudyPal tool allows users to find learning partners who are taking similar courses in the same region. However, it lacks functions and tools designed for "mentors", which means that students cannot accurately match their learning partners for a particular course, nor can they identify their expertise or strengths in that area of the course. This situation may not address the need for personalized learning guidance for students.

The software aims to find students in a series of courses such as IELTS and TOEFL who are also taking this series of courses.

(Screenshot of StudyPal below)



STUDYPAL

Avish Kamdar

Actuarial Exam

Auckland, Auckland, New Zealand

March 2020

University of Mumbai

Getting there

Shing Zhu

CFA II

Auckland, Auckland, New Zealand

June 2015

Other university

Getting there

Ding Zhaihao

IELTS

Auckland, Auckland, New Zealand

October 2002

Abilene Christian University (TX)

Just Starting

Riffat Sultan

4.4 Methods and tools:

	APP front-end development	Flutter is designed to develop high-performance, beautiful applications to provide native performance for Android. Through a single code base, developers can quickly build and test cross-platform applications, greatly saving development time and resources.
	Prototype design (UI)	Figma is a cloud-based design tool that allows users to collaborate to create, test, and share user interface designs. It provides a complete set of design tools that can be used to create interactive prototypes. In fact, the real-time collaboration feature enables the team to edit and comment on the design at the same time, greatly improving the efficiency and coordination of the project.
	Version control	GitHub is a Git-based version control platform that allows developers to collaborate on projects.GitHub provides code hosting, version control, and collaboration tools that enable developers to track changes to projects, discuss issues, and merge code changes. It is an important infrastructure for both open-source and private software projects.

	Code Development Tools	It offers code highlighting, smart code completion, debugging, and extension features, and supports a wide range of programming languages and frameworks.VS Code's lightweight and highly customizable nature made it our first choice.
 	Back-end development	Java is a widely used programming language designed to have as few implementation dependencies as possible. Due to its cross-platform nature and strong community support, java has become a popular choice for mobile applications. Its stability and performance have made it widely used in many fields.
 	Database system	A relational database management system that provides a system for managing data organized through a relational model. It is widely used in website databases, business applications, and data warehouses, etc. MySQL is open source and offers high performance, reliability, and ease of use

5. Specification & Design

5.1 SPECIFICATION

Studypal provides end-to-end functionality between students and tutors on mobile applications.

View operation:

1. Check out all the helpful course information posted by the tutor.
2. Check out all the information posted by students about courses they want help with
3. View the tutor's Profile
4. Browse favorite posts
5. View all posts by yourself

Editing operation:

1. The tutor and editor introduced themselves.
2. Users manage their own posts

5.2 User requirements

Students:

1. Jump to the registration and login screen when opening the application
2. A rich course listing home page displaying information on numerous courses with the help of multiple tutors.
3. Excellent Search and Filter Course List Features
4. Post personal help information and ask for help from others
5. Favourite courses showing personal preferences
6. Manage your own requests for help.

Tutors:

1. Jump to the registration and login interface when you open the application
2. Intuitive, brief interface design
3. Can browse the help information of all students
4. Release courses that can help according to personal advantages
5. Write a self-introduction that highlights your personal advantages
6. Manage the course information released by yourself

5.3 Use cases

Use case 1: Tutor registers applications, publishes courses that it can help, and expects to get in touch with students.

1. When the tutor opens the app interface, a login registration page is displayed asking for an account number and password.
2. After filling out the basic account information, the teacher was asked to write a self-introduction that highlighted his or her strengths.
3. After entering the APP, the tutor can go to the "I can help" interface to create and publish course information that they can help. Then wait for interested students to view the published course information and contact themselves.

Use case 2: The tutor logs in to the application and hopes to find a course that he is qualified for.

1. When the tutor opens the application interface, it displays a login page asking for an account number and password.
2. When directed to the "I can help" list, the tutor can see a large number of student requests for help.
3. On this interface, the tutor can locate relevant courses through search or filtering functions.

4. After selecting the course you are interested in, you will go to the course details page, where you can view the details of the student's need for help, and then decide whether you want to contact the student for help.

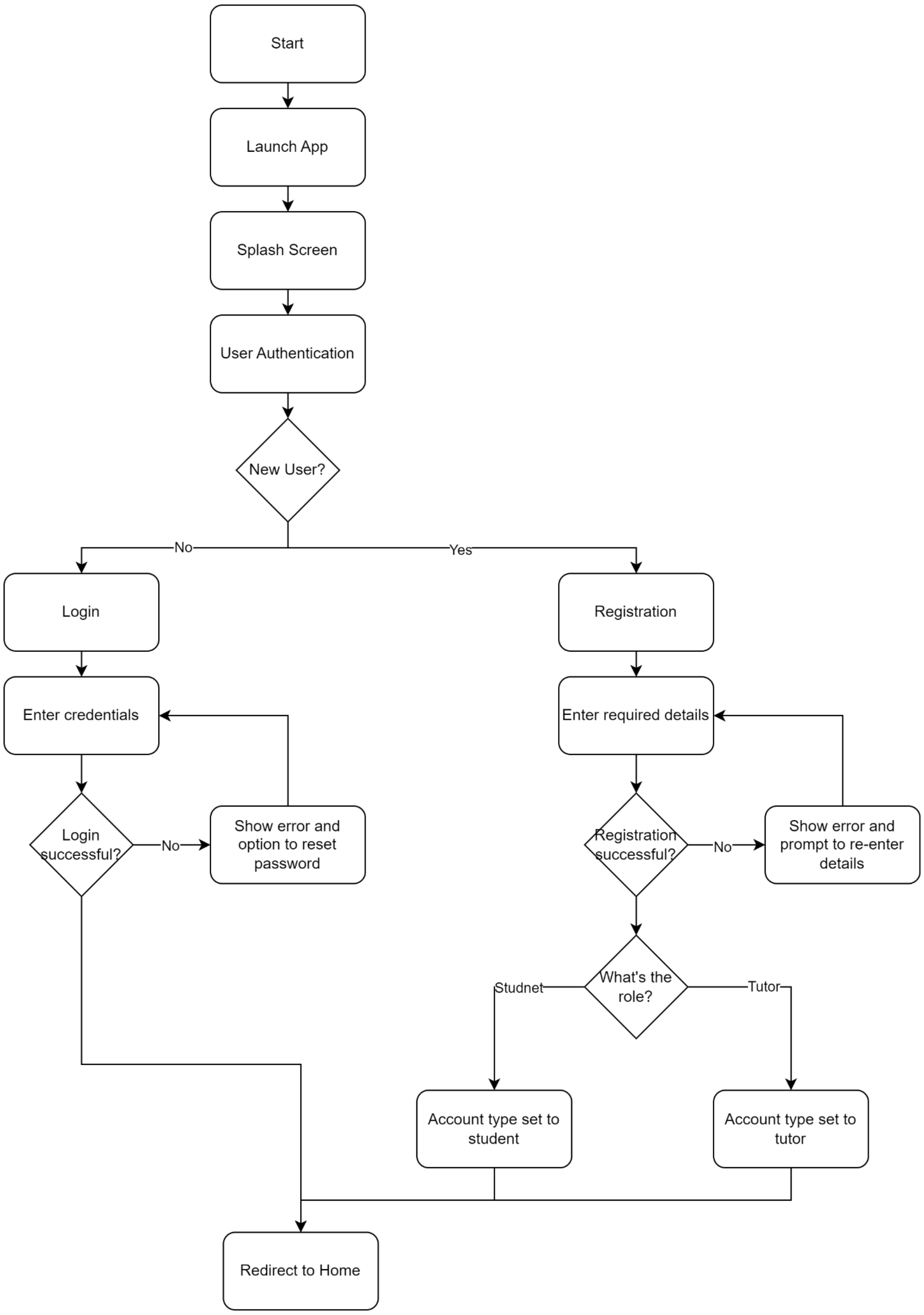
Use case 3: When students are looking for a tutor who can provide the help they need if they do not find suitable course information, they hope to post their own request so that the tutor who feels that they can provide help can contact them.

1. When you open the application interface, a login registration page is displayed, asking for an account number and password
2. After entering the APP, users will be directed to the course list home page, where they can search and filter to find courses that meet their needs.
3. When students are unable to find a course that meets their needs, they can choose the "I need help" interface, publish the details of the courses they need help and wait for a capable tutor to contact them.

5.4 Design

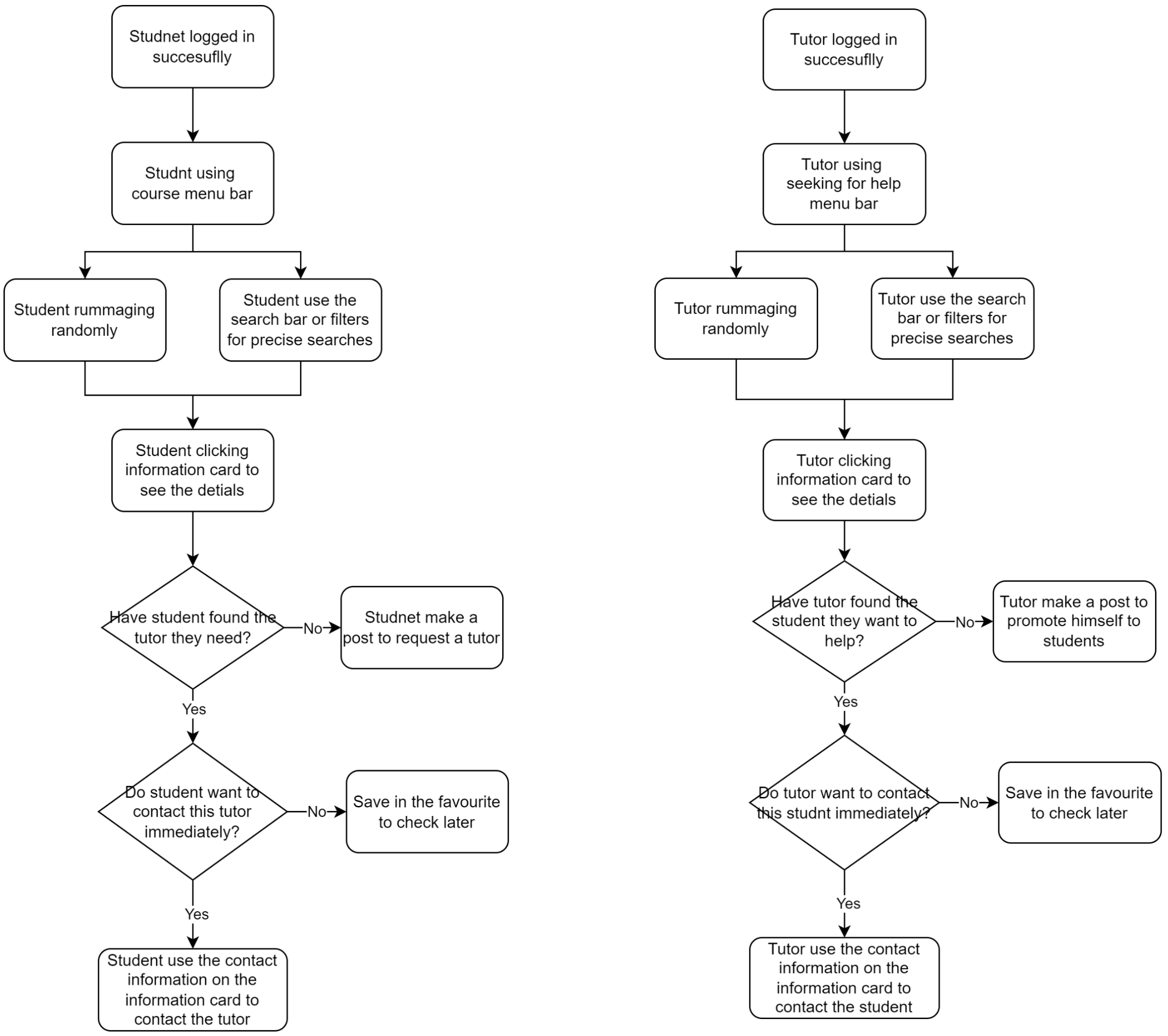
1. User registration log in data stream

When you open the user login interface, you will be asked to enter your own account information if the user record is not in our database, the user's login failed. When the user's account is successfully logged in, the database will automatically identify the user identity selected by the user when registering, and guide the students and mentors to different app versions.

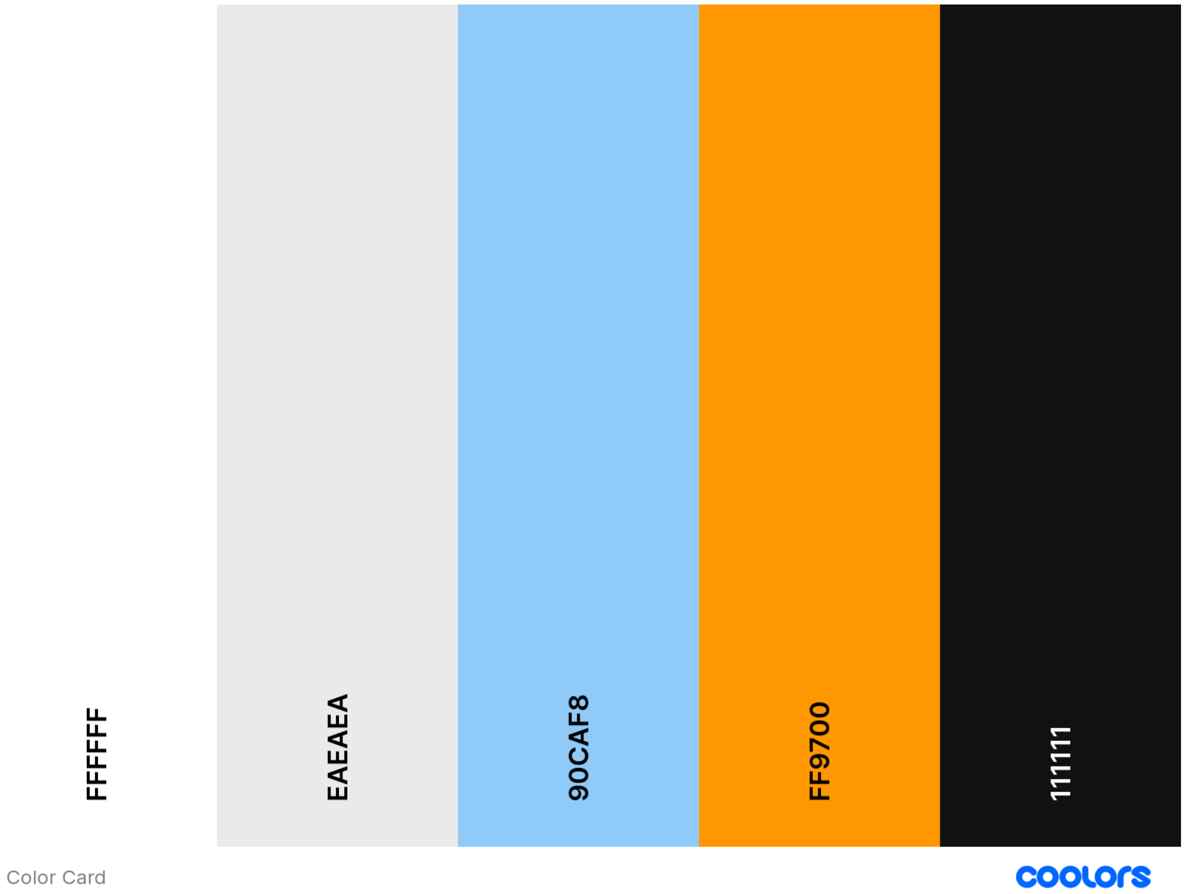


2. Student and tutor version information data stream

Once users successfully log in, they will be taken to the Crouse list. If you are a teacher, your Crouse list will be full of student requests for help, and if you are a student, it will be full of course information that can help. Users can browse the course schedule manually, or use the search box and filter to query the course. When the user clicks to select the course, the system will show the user detailed help or help information. Users can also use the snakebar under the app to switch to the favourites, publishing and profile interfaces. Also, in the profile interface, the teacher version will need to fill in more abundant account information, in turn, to help students better understand the advantages of relevant mentors and facilitate other student users to choose.


Studnet logged in succesuflly
Studnt using course menu bar
Student rummaging randomly
Student use the search bar or filters for precise searches
Student clicking information card to see the detials
Have student found the tutor they need?
No
Studnet make a post to request a tutor
Yes
Do student want to contact this tutor immediately?
No
Save in the favourite to check later
Yes
Student use the contact information on the information card to contact the tutor
Tutor logged in succesuflly
Tutor using seeking for help menu bar
Tutor rummaging randomly
Tutor use the search bar or filters for precise searches
Tutor clicking information card to see the detials
Have tutor found the student they want to help?
No
Tutor make a post to promote himself to students
Yes
Do tutor want to contact this studnt immediately?
No
Save in the favourite to check later
Yes
Tutor use the contact information on the information card to contact the student

5.5 Colour palette



In the process of APP development, our entire application colour plate plays an important role in the aesthetic appeal of the entire user experience. First of all, we take white as our basic background to make the whole page more readable. Then use orange as our tag design color, especially for the outstanding content needed by the application. This kind of colour design can not only attract the attention of users but also stimulate their enthusiasm. The text and icon in the application use the most common black as our dominant tone, and combination it with a white background always ensures that the content is in focus. Finally, the top of the app is decorated with light blue and forms a refreshing contrast with white. This choice not only helps to split the interface of the application but also enables better user navigation.

5.6 Design choices

The design of the project follows the principles of control, consistency, tolerance, directness and simplicity in human-computer interaction design.

Control:

Users can click and use the snackbar below the app to jump between different app interfaces. And the widgets displayed to the user on each page have actual functions. For example, the user clicks on the course card to jump to the details page. At the same time, except for the four main pages displayed below the snackbar , and all other interfaces Each detail page has a return button in the upper left corner to ensure that users can return to the previous operation.

Directness:

Following the design principle of human-computer interaction, all our buttons have names, not only using tags. The design idea here is to tell users exactly what to do when this button is disabled. At the same time, there will be a description of the interface function in the app session, so that the design can tell the user more directly where the user is currently at and what the user is looking at.

Tolerance:

When users edit their profile or published information, they can cancel the content they do not want at any time. When a user publishes a course and finds that he is not suitable for the content of the course, he can go to the management publishing interface to view all his publications or delete certain publications. . Similarly, even if the user confirms his new profile content, student can still go to the profile interface again to modify it to ensure that the user can ensure that the content is what he wants to display.

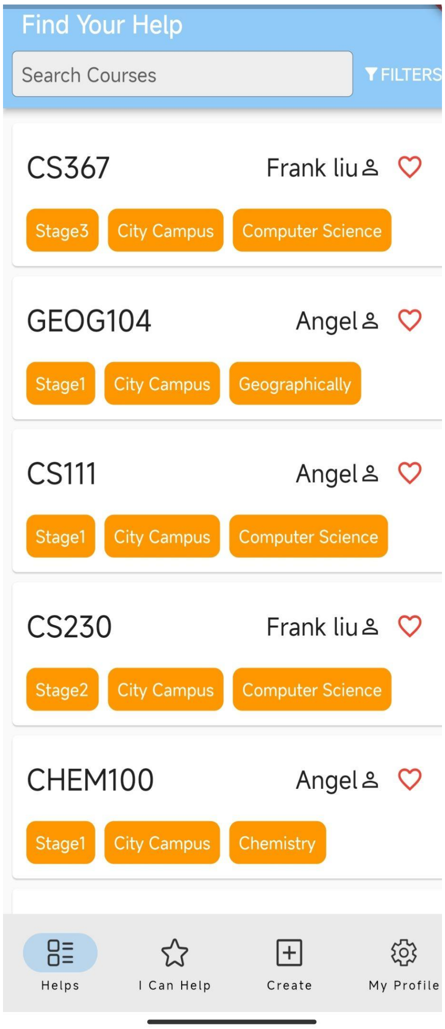
Simplicity:

All course cards in the course list displayed in the system are designed to have no more than three tags and only contain key information such as the course name, publisher name, and course price. This prevents users from being overwhelmed by a large amount of content displayed at once, and allows users to quickly configure courses that suit their general direction through important information such as course names or prices. When the user clicks on the course card, the user can read more related content of the course.

5.7 Page design

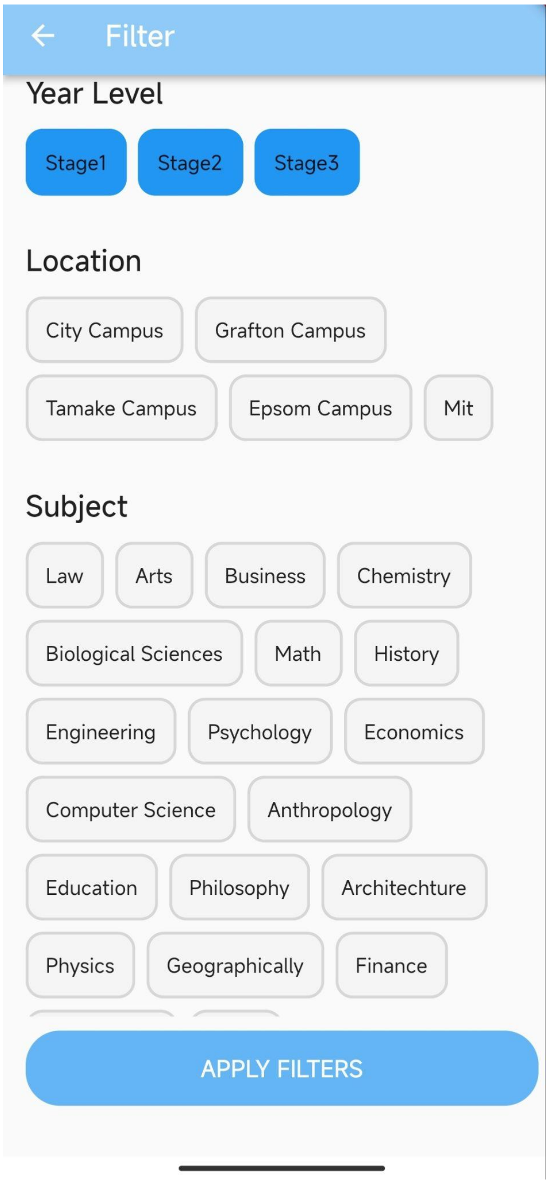
Home Page

Course list, as the core function of the Studypal application, shows users all courses posted by students or tutors. The course list here also integrates the search and filter functions, providing students with a highly integrated interface that allows users to quickly filter courses that match their needs.



Filter

Given that students have diverse academic needs, filtering allows them to find suitable tutors by specific grade level, location or subject. This way, students can more easily find the specific help they need and avoid aimless browsing and searching. Vice versa, tutors can use the filter function to more easily find the students they can help. The right filtering capabilities can ensure a higher matching rate between students and tutors who can meet their academic needs, thereby improving the effectiveness and user satisfaction of the entire platform.



Filter

Year Level

Stage1 Stage2 Stage3

Location

City Campus Grafton Campus

Tamake Campus Epsom Campus Mit

Subject

Law Arts Business Chemistry

Biological Sciences Math History

Engineering Psychology Economics

Computer Science Anthropology

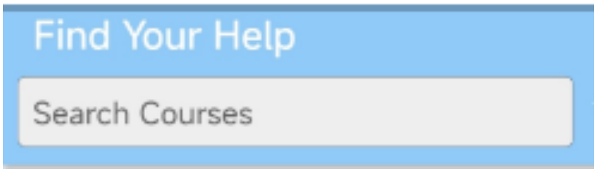
Education Philosophy Architechture

Physics Geographically Finance

APPLY FILTERS

Direct search

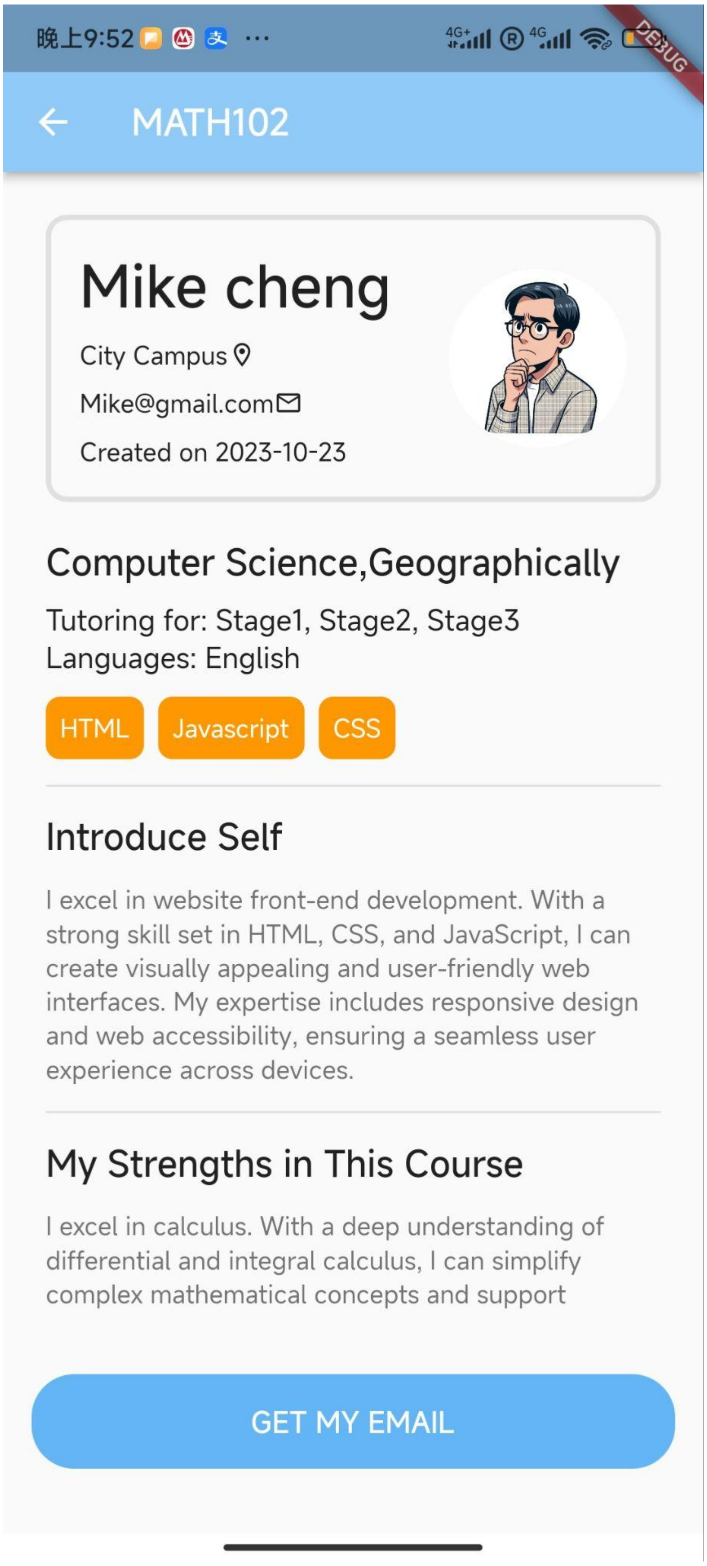
The direct search function allows users to quickly enter course keywords or other information they are interested in to immediately find relevant tutors. Conversely, tutors can also enter course keywords or other information they are good at to immediately find students they can help. This is the most direct and fastest method for users who already know what they need. While filtering and sorting are useful, direct search gives users a more flexible way to search based on their specific needs or keywords. New users or those who are not familiar with all the available filtering options may find direct search an easier way to find the resources they need.



Information card

All course cards in the course list displayed in the system are carefully designed, giving users sufficient key information to quickly prepare courses that suit their own direction. After clicking on the course card, more detailed information will be displayed, which gives students or teachers more reference content. Since the content expected by tutors and students is different, we have differentiated designs to meet the different needs of the two groups of people.

- For student: When the course card is not opened, it will display: the course code, course stage, location of the tutor, subject of the course, tutoring fee expected by the tutor, name of the tutor and the number of times the current course card has been collected. The "red heart" shaped button in the upper right corner of the Source card can put the current course card into favourites. When the user clicks on the course card and opens it, the course card will show the student the specific information of the tutor, which includes: the tutor avatar, tutor name, tutor location, tutor contact information, and the time when the tutor released the current course card. The tutor’s major, the stage at which the tutor can teach, the language in which the tutor can communicate, the tutor’s strong skills, the tutor’s self-introduction, and the tutor’s Strengths in This Course.



Mike cheng

City Campus

Mike@gmail.com

Created on 2023-10-23

Computer Science,Geographically

Tutoring for: Stage1, Stage2, Stage3
Languages: English

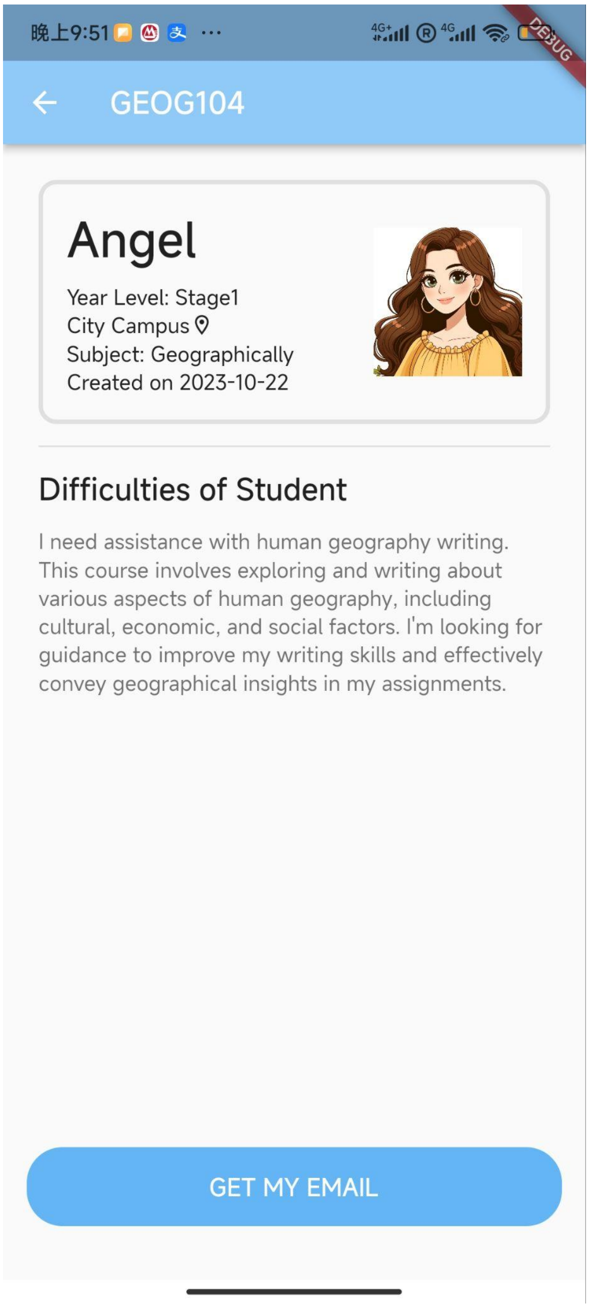
Introduce Self

I excel in website front-end development. With a strong skill set in HTML, CSS, and JavaScript, I can create visually appealing and user-friendly web interfaces. My expertise includes responsive design and web accessibility, ensuring a seamless user experience across devices.

My Strengths in This Course

I excel in calculus. With a deep understanding of differential and integral calculus, I can simplify complex mathematical concepts and support

- For tutor: When the help card is not opened, it will display: the course code, course stage, student location, subject studied by the student, and student name. The "red heart" shaped button in the upper right corner of the help card can put the current help card into favourites. Different from the Student interface, the help card will only show the tutor the difficulties that the student encounters in the course. The reason for this design is that it is more important for students to explain the problems they encounter, rather than to promote themselves to the tutor.



Angel

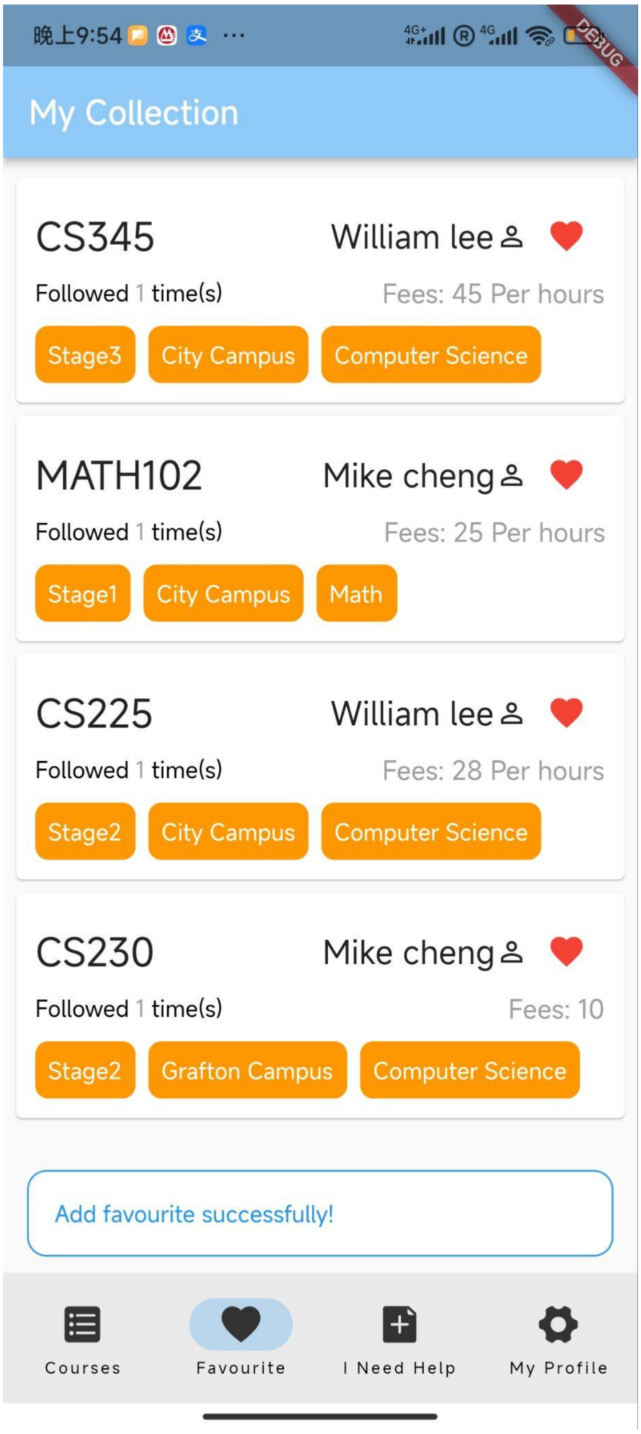
Year Level: Stage1
City Campus
Subject: Geographically
Created on 2023-10-22

Difficulties of Student

I need assistance with human geography writing. This course involves exploring and writing about various aspects of human geography, including cultural, economic, and social factors. I'm looking for guidance to improve my writing skills and effectively convey geographical insights in my assignments.

Favourite

Students may encounter tutors while browsing that they find useful but currently do not have the time to study in depth or require further comparison. By using the "favourite" feature, they can easily save this information for later review. This feature provides a more personalized user experience, giving users the flexibility to manage and access content. according to their own time and needs.



晚上9:54

My Collection

CS345 William lee ♥
Followed 1 time(s) Fees: 45 Per hours
Stage3 City Campus Computer Science

MATH102 Mike cheng ♥
Followed 1 time(s) Fees: 25 Per hours
Stage1 City Campus Math

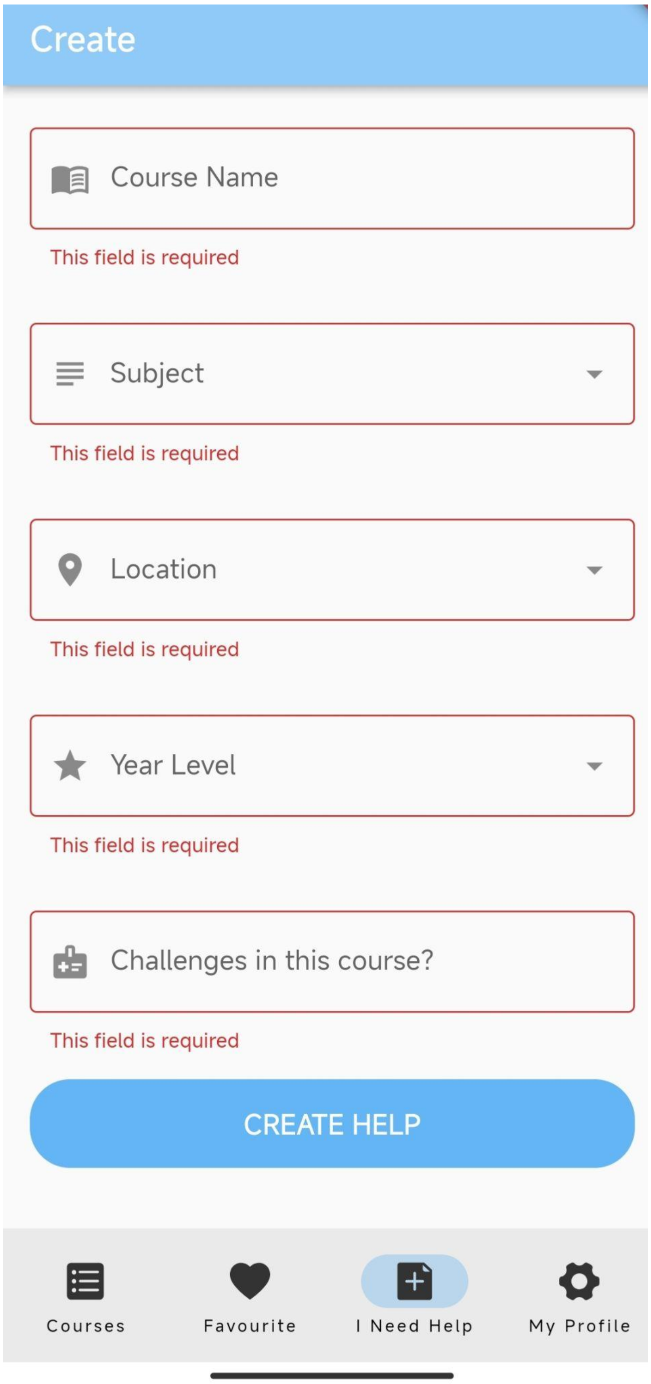
CS225 William lee ♥
Followed 1 time(s) Fees: 28 Per hours
Stage2 City Campus Computer Science

CS230 Mike cheng ♥
Followed 1 time(s) Fees: 10
Stage2 Grafton Campus Computer Science

Add favourite successfully!

I Need help / I can help

Allow users to post a message or request on the app describing their academic needs or problems they encounter so that tutors can see and respond. Vice versa, tutors can publish information to promote themselves in the app for students to find. Both students and teachers can express their needs, rather than just searching or filtering to find help. At the same time, it can also make up for the pain of not being able to find the resources you want directly on the home page.



Create

Course Name

This field is required

Subject

This field is required

Location

This field is required

Year Level

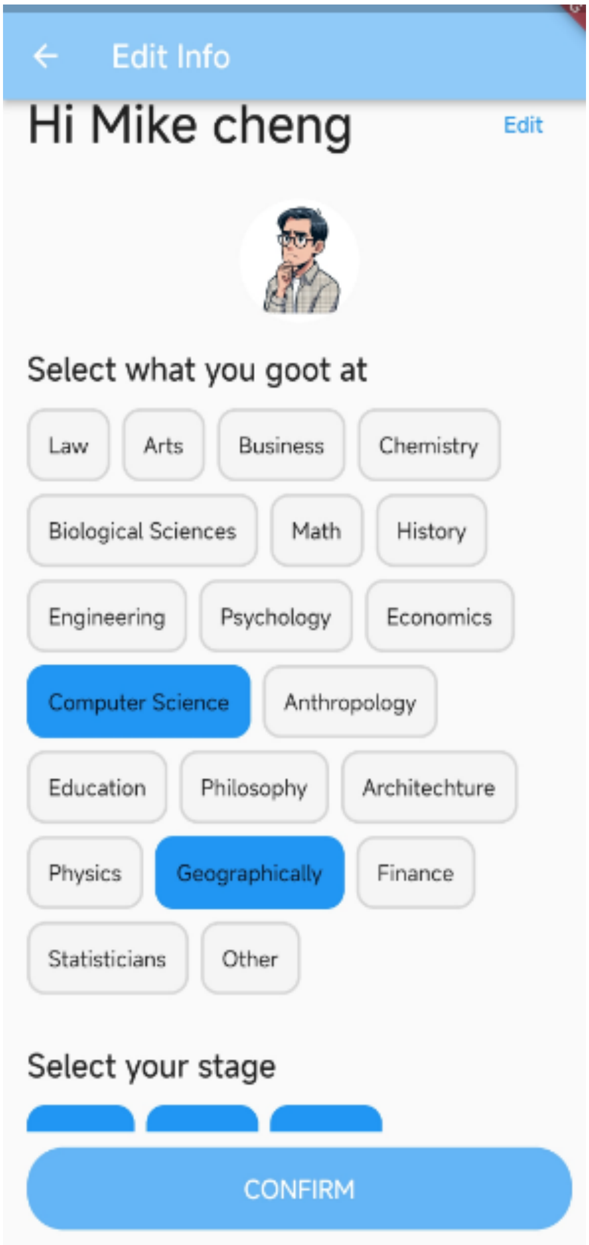
This field is required

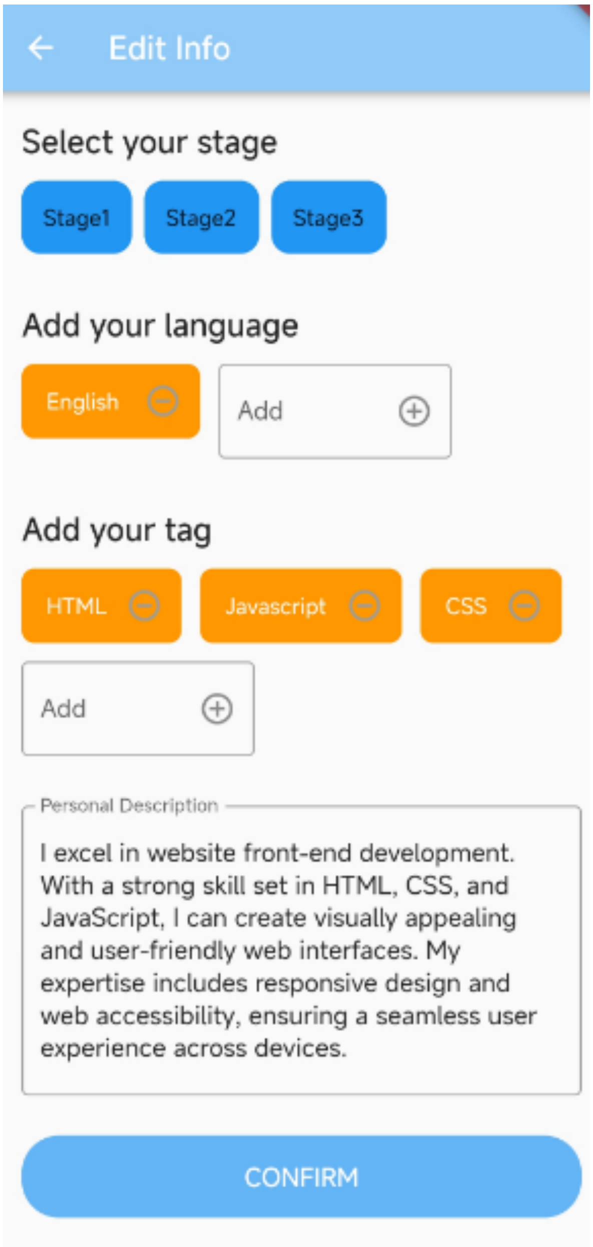
Challenges in this course?

This field is required

My profile

Modify profile allows users to change their personal information in the application, such as name, avatar, and tutors can modify their academic background. In this way, users can update their information according to their own changes and preferences to ensure the accuracy and timeliness of the information.

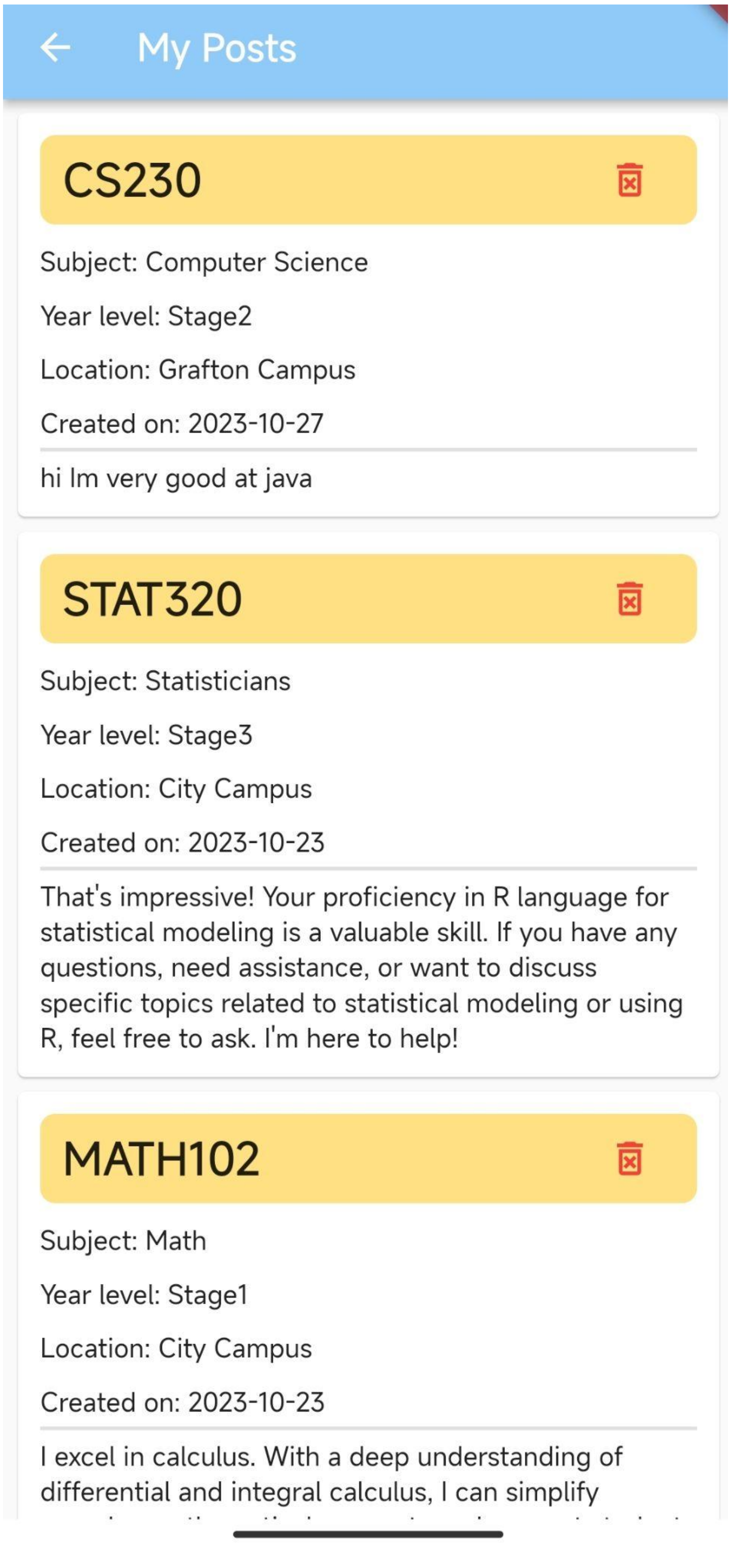




I excel in website front-end development. With a strong skill set in HTML, CSS, and JavaScript, I can create visually appealing and user-friendly web interfaces. My expertise includes responsive design and web accessibility, ensuring a seamless user experience across devices.

Manage your own post

Allow users to view and delete all promotions or requests they have posted on the application. Ensure that users can control what they post and update or delete posts that are outdated or no longer relevant.



CS230

Subject: Computer Science

Year level: Stage2

Location: Grafton Campus

Created on: 2023-10-27

hi Im very good at java

STAT320

Subject: Statisticians

Year level: Stage3

Location: City Campus

Created on: 2023-10-23

That's impressive! Your proficiency in R language for statistical modeling is a valuable skill. If you have any questions, need assistance, or want to discuss specific topics related to statistical modeling or using R, feel free to ask. I'm here to help!

MATH102

Subject: Math

Year level: Stage1

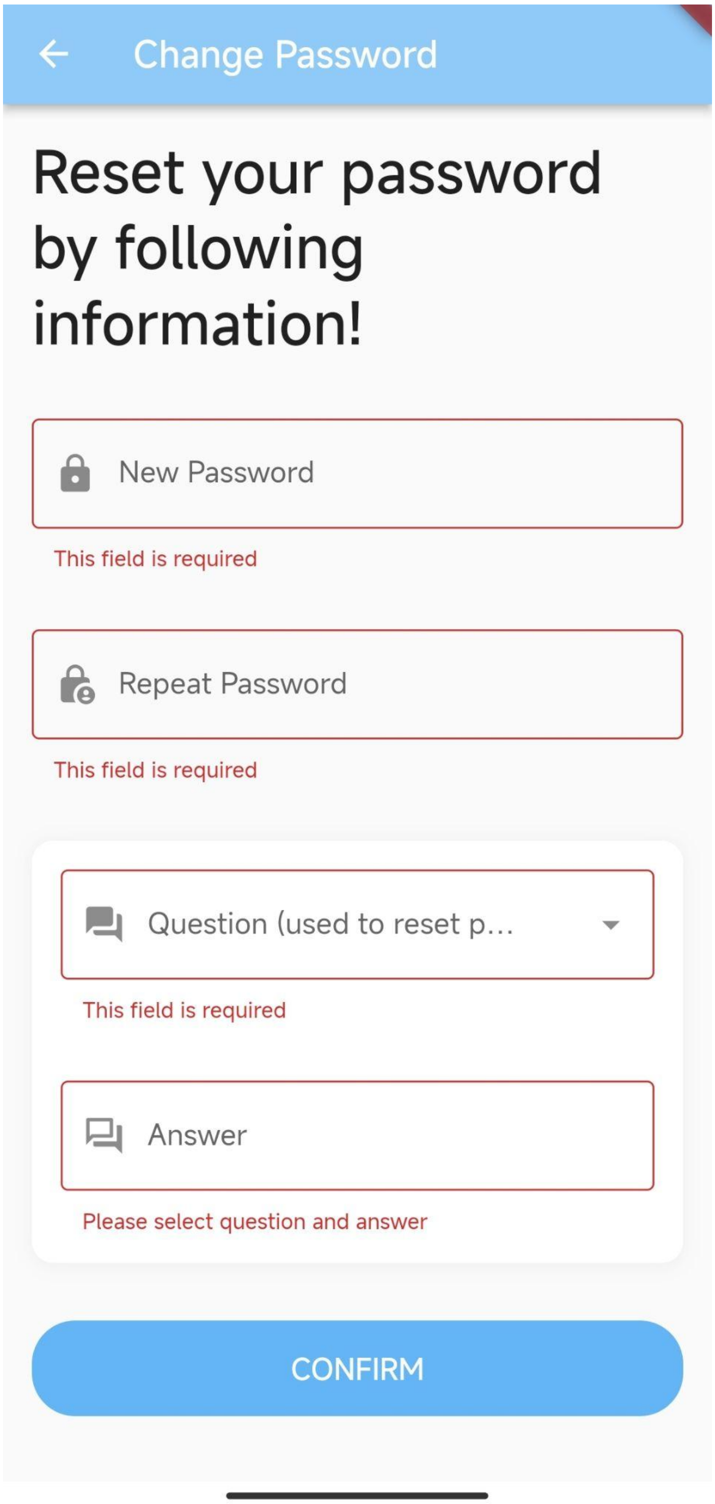
Location: City Campus

Created on: 2023-10-23

I excel in calculus. With a deep understanding of differential and integral calculus, I can simplify

Change the password

Allow users to change their login password. If a user forgets their password or thinks their password has been known to someone else, they can easily change it through this feature, avoiding the frustration of not being able to access their account.



Change Password

Reset your password by following information!

New Password

This field is required

Repeat Password

This field is required

Question (used to reset p…

This field is required

Answer

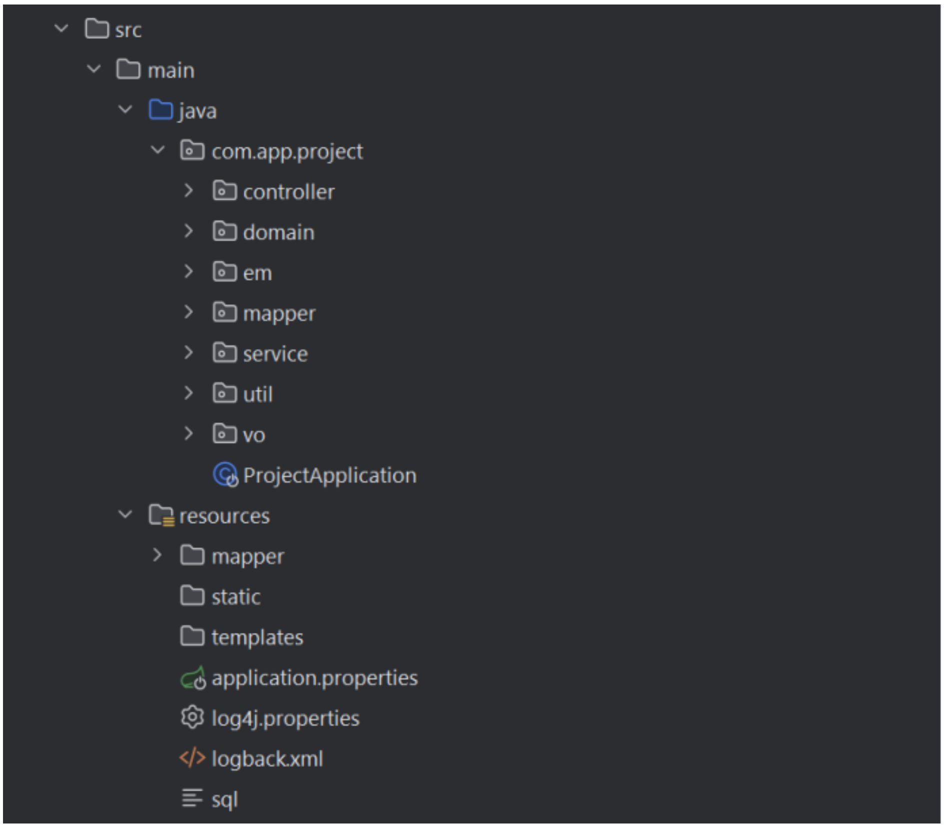
Please select question and answer

CONFIRM

6. Implementation

6.1 Back-end implementation

6.1.1 mvc Organization



```
src
  main
    java
      com.app.project
        controller
        domain
        em
        mapper
        service
        util
        vo
        ProjectApplication
    resources
      mapper
      static
      templates
      application.properties
      log4j.properties
      logback.xml
      sql
```

The entire application follows the MVC architecture (Model-View-Controller), which provides a simple and efficient way to connect the front-end user interface to the back-end data store and reduces the bidirectional coupling of the view with the business logic.

Model-> domain and vo

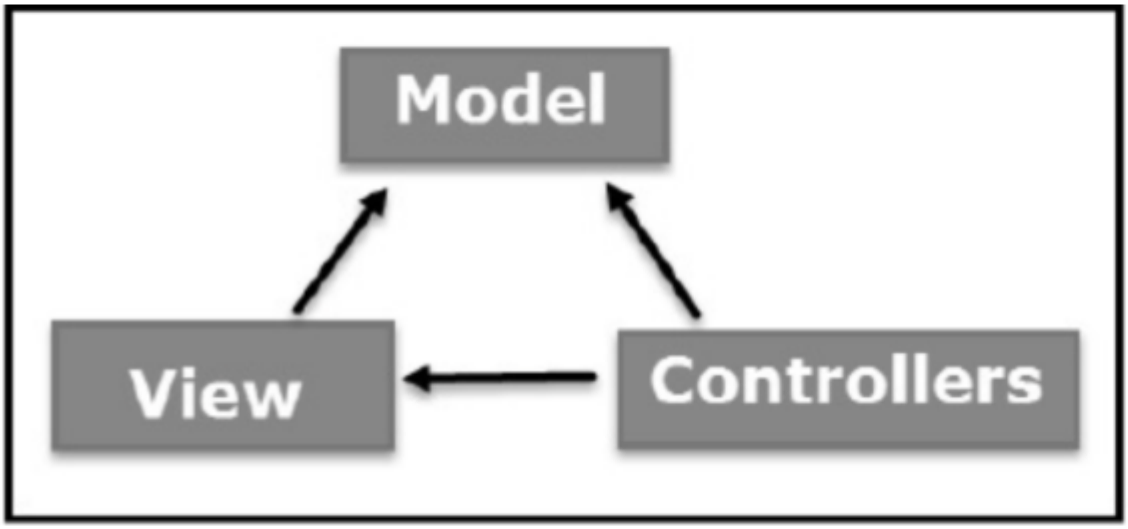
Controller -> Controller

View ->flutter

The model (domain and vo) provides the data to be displayed and therefore contains the data and behaviour.

VIEW (flutter) is responsible for the presentation of the model, generally known as the project's app interface.

The controller is responsible for accepting user requests, delegating them to the model for processing, and then returning the model data after processing to the VIEW (flutter) for display by the VIEW (flutter), which means that the controller plays the role of scheduling in the project.



6.1.2 Spring Boot

The whole project is created based on Springboot, and the various levels are based on Spring-Ioc can complete the data source, conversation factory, conversation will be and Dao object creation. And in Spring Boot, we can use annotations to define and identify beans. these beans will be automatically registered through the ConfigurationClassPostProcessor.

6.1.3 Log management

Logging is implemented to track application behaviour, errors and events. There are different logging levels specified in our code, such as DEBUG, INFO, and WARN.These levels allow you to control the information logged and are usually categorized by importance. For example, the debug level is used for detailed debugging information, while the WARN level is used for warning information. For example, in our code, logback.rootLogger defines a root logger that logs WARN and higher level log messages and outputs them to the console and log files.

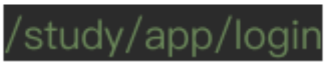
6.1.4 Security and Optimization

The login service is mainly responsible for generating login tokens and validating user passwords. The registration service generates a token when it detects that the account password is legitimate and stores the encrypted password in the database. To support these functions, a base user service is provided that generates random tokens, encrypts passwords, sets SALT values, and returns UserDO objects. For other interfaces on the backend, you can access them directly by simply holding an authentication token.

6.1.5 API design

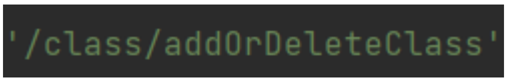
The Spring Framework and associated annotations are used to create a RESTful API endpoint to perform various operations via HTTP requests to support communication between clients and servers.

Main API

1 

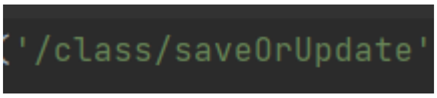
```
/study/app/login
```

Realization of the user registration function

2 

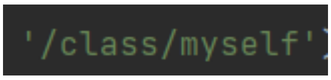
```
'/class/addOrDeleteClass'
```

Add user favorite courses or cancel them.

3, 

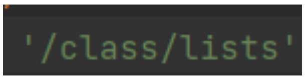
```
('/class/saveOrUpdate'
```

Enables users to create and modify courses

4 

```
'/class/myself')
```

View user-posted courses

5 

'/class/lists'

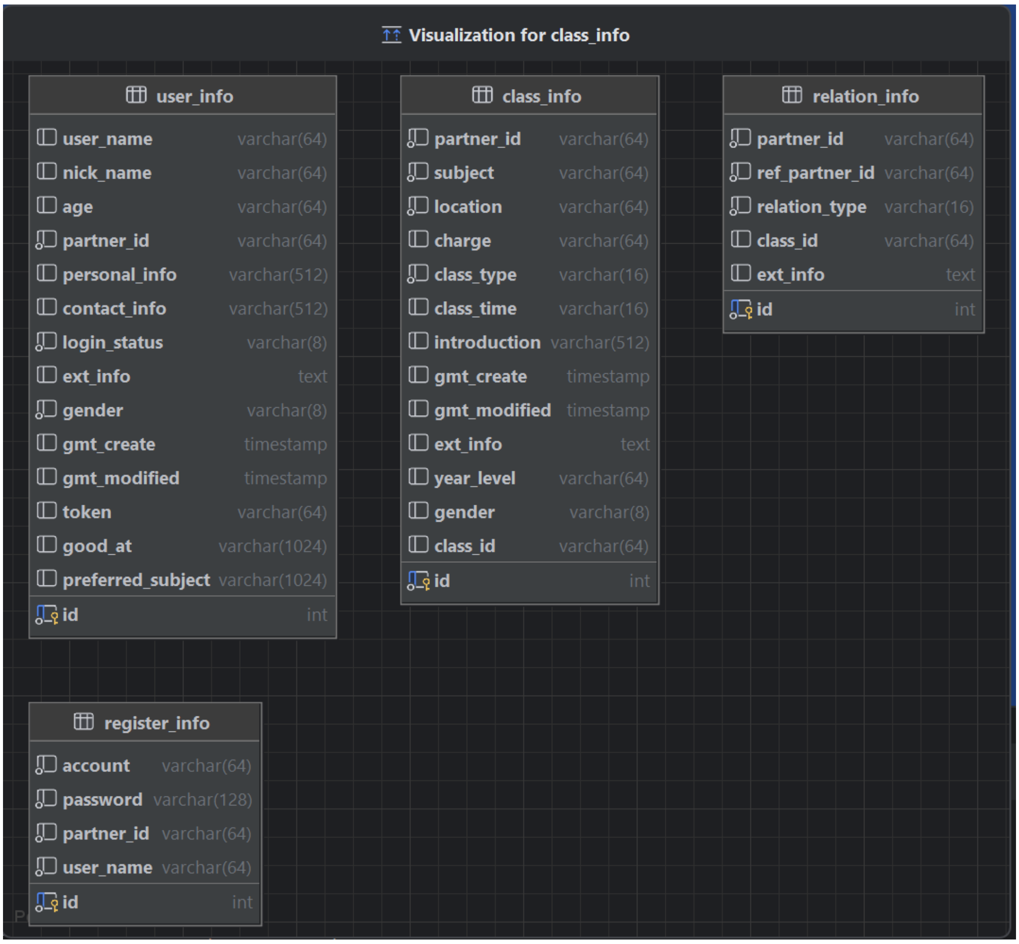
Course list query in user interface, including class name fuzzy query

6

/study/app/register

Realization of user registration app

6.1.6 Database design



Visualization for class_info

user_info	
user_name	varchar(64)
nick_name	varchar(64)
age	varchar(64)
partner_id	varchar(64)
personal_info	varchar(512)
contact_info	varchar(512)
login_status	varchar(8)
ext_info	text
gender	varchar(8)
gmt_create	timestamp
gmt_modified	timestamp
token	varchar(64)
good_at	varchar(1024)
preferred_subject	varchar(1024)
id	int

class_info	
partner_id	varchar(64)
subject	varchar(64)
location	varchar(64)
charge	varchar(64)
class_type	varchar(16)
class_time	varchar(16)
introduction	varchar(512)
gmt_create	timestamp
gmt_modified	timestamp
ext_info	text
year_level	varchar(64)
gender	varchar(8)
class_id	varchar(64)
id	int

relation_info	
partner_id	varchar(64)
ref_partner_id	varchar(64)
relation_type	varchar(16)
class_id	varchar(64)
ext_info	text
id	int

register_info	
account	varchar(64)
password	varchar(128)
partner_id	varchar(64)
user_name	varchar(64)
id	int

Our database contains four separate tables, which means that there are no foreign key associations between these tables. Despite this, we still use Java to retrieve the required data across multiple tables. This database primarily manages user profiles, courses and their information, user registrations, and relationships between various entities.

Using MyBatis as our database access tool, we generated several mapping files (UserMapper.xml, ClassMapper.xml, RelationMapper.xml, and RegisterMapper.xml) for various database tables. The mapping relationships between Java objects and database tables, as well as SQL operations like insert, update, query, and delete, are defined by these mapping files.

To obtain the UserDO object, for instance, we can import UserMapper and use its selectByCondition function to perform a database query. Every DO (data object) class has fields and associated get and set methods, just like a database table. The "PartnerDO" class, for instance, is used to represent user data in fields like "nickName," "name," "partnerId," and "personalInfo."

6.1.7 Service layer and controller

The main logic of our application—that is, the logic pertaining to users, usability, relationships, etc.—is written at the service layer. The logic needed to process and work with the data is contained in it. It handles the logic as a go-between for the controller and the mappers, utilizing MyBatis Mappers to access the database. includes features for managing courses, logging in, and registering. To receive requests from clients, process their HTTP requests, and map those requests to handler methods, we'll utilize controllers. Ultimately, the client receives their outcome back.

6.1.8 Database retrieval

We construct the RelationService, beginning with the addOrDeleteClass method, because we must manage the relationship between a user and a class: This procedure is used to modify the connection between a user and a class. After confirming the user's identity and current login state, it builds a RelationDO object to symbolize the user-class relationship and determines whether to add or remove it based on the value of the select field. The relation is added to the database if the select value is "add," and eliminated otherwise. To find out more about the relationship between the user and the class, there is also ClassRelationQuery. After confirming the user's identity and login status, it builds a RelationDO object based on the user type (student or teacher) in order to obtain the necessary relationship data. It then iterates through the relationships that were retrieved, creating a ClassInfoVO object with class and user data for each relationship. Ultimately, a list with details on every relationship is returned.

6.2 Front-end implementation:

6.2.1 Navigation bar

In order to provide users with a more efficient and intuitive interface experience, we adopt an icon combined with the bottom label text as an interactive feedback element. Specifically, we utilize the NavigationDestination component to construct the bottom navigation bar. By carefully designing the icon and configuring the guidance text in the label attribute, we realize the purpose of allowing users to seamlessly switch between multiple functional modules or pages.

6.2.2 Course list loads multiple courses and refreshes the course list

When there are more than 4-5 course cards in the program database, the home page will not display additional cards to maintain user experience and avoid interface complexity caused by too much information. Users can refresh or browse more courses by scrolling down or sliding. I defined a generic Flutter component called SpList to implement the slideable interface, which includes the ListView that comes with flutter and integrates reset and fetchMore methods. The reset is triggered when scrolling to the top and the fetchMore is triggered when scrolling to the bottom to load new data.

6.2.3 Use the system security

In our project, we have introduced the Validations static class, which is mainly used for formatting and validating various inputs. This class mainly specifies the format of account numbers, passwords and security questions. Specifically, it ensures that account numbers are between 6 and 12 characters long, and passwords need to include both letters and numbers.

To further enhance the security of the user account, the system also requires the user to set a password question during the registration phase. In order to do so, we design a QuestionInput component to collect user's questions and answers. Utilizing Flutter's TextEditingController, we use questionController and answerController to manage questions and answers and ensure that they meet our security standards through Validations.questionAndAnswer.

6.2.4 Tutor profile

In the tutor interface, the tutor is able to update his resume, e.g., if he excels in a Java course, he can add a Java tag in order to notify students seeking help with Java. In the info interface, we define the _TutorInfoState class and design several references here, including subjectViewModel (derived from components/sp_item.dart), yearLevelViewModel (also derived from components/sp_item .dart), languageViewModel and tagsViewModel (both from components/sp_custom_items.dart), and personalInfoViewModel (from components/sp_input.dart). These data models are used to display and capture user input. By defining the asynchronous function onApplyPressed, we can update the mentor's information based on the modifications. A prompt is displayed when the update succeeds, while an error message is displayed when it fails.

6.3 Unforeseen problems

1. During the testing phase of the project, we did not have any relevant experience in quality testing, so we spent a lot of time exploring a test methodology that could be repeated many times.
2. The client was very dissatisfied with our first project prototype because the functionality of our initial design was very different from the client's expectations in terms of direction, so we had to make a lot of changes to a lot of the underlying logic.
3. In the design of the project filter design planning, we underestimated its difficulty and spent a lot of time on the development of the filter. As a result, we were a bit short on time for the whole project.

7. Results & Evaluation

7.1 Project objectives

The project has achieved its stated objectives to a large extent. Although time constraints and some obstacles to the project's feasibility caused delays in the development schedule, most of the project's targeted features have been realise. This mobile application was built from scratch and successfully met the requirements set at the beginning of the project.

The application has successfully implemented the "help posting" function, providing a two-way interaction between tutors and students. Now, not only can students look for tutors who are willing to help, but tutors can also offer help to students who need it. This not only meets the needs of both parties but also builds a bridge for communication between them. Through this feature, we have not only facilitated the transfer and exchange of knowledge but also increased the usefulness and value of the program. Despite some delays, the main goal of the project has been well achieved, providing users with a practical and efficient interaction platform.

7.2 Testing

For system testing, we adopt a multi-layered and multi-perspective testing strategy to ensure that the functionality and performance of the application meet the expected requirements. Each project iteration phase is accompanied by a detailed testing plan, and the following is our testing process and approach：

1. Acceptance test：
    - We first performed acceptance tests to verify that the application's functionality meets the project requirements. The test results show that the application's functionality performs as expected and so far we have not found any bugs.
2. Integration test：

    - During the integration testing phase, we paid attention to possible errors caused by interactions and operations between components. Once we found the errors, we immediately fixed them to ensure the stability and correctness of the system. As of the last test, we did not find any functional errors.

3. Back-end API and database testing：
    - While testing the back-end API and database, we encountered a serious dirty read issue which could cause the application to crash. To address this issue, we invested a lot of time in redesigning and modifying the database tables to eliminate the possibility of dirty read errors.
4. Test records and problem-solving：
    - Although we did not record all the test results due to project practices, all the issues identified during the testing sessions were effectively resolved.

Through the above testing process, we have ensured the reliability and stability of the system and laid a solid foundation for the successful delivery of the project. The test plan for each iteration is an important part of our project quality assurance, which helps us identify and solve problems in a timely manner to ensure the smooth progress of the project.

7.3 Critical evaluation

- Advantage
    - Our mobile apps have an efficient user flow that makes it intuitive and easy for users to access and use the app's features.
    - The application's user interface is designed to be clean and simple, providing a great visual experience that allows users to navigate and operate with ease.
    - Most of the features meet the basic needs of the client and are in line with the original project design and the client's expectations.

    - Our page design adapts to different screen sizes in Android, ensuring the usability and user experience of the app on different devices.

- Disadvantages
    - Not all predefined features have been implemented, for example, communication technology between users within the application has not yet been implemented, which may affect the user interaction experience.
    - The application may have some instability due to some issues in code management. Also, some insufficiently tested or uncertain modifications may have led to some bugs, which may affect the performance of the application and user satisfaction.
    - The lack of 2-step authentication in the App may pose a security risk.

The above strengths and weaknesses analysis provides us with valuable feedback on the current state of the project and possible areas for improvement. In future development, we will strive to improve the code management process, refine the unimplemented features, and continue to optimize the user experience in order to push the project to a higher level of completion.

8. Future Work

Due to time constraints and technical issues, we were unable to implement all the features we intended to add to our app.

Real-time messaging system:

At present, the only way to establish substantive communication between students and tutors is to click on "Get My Email" to get the other party's email, and then jump to an external app to send information. This makes the process more cumbersome for users. Therefore, in the future, we would like to add a real-time messaging system to the app, so that users can communicate with each other within the app, which ensures the efficiency and simplicity of communication.

Evaluation and feedback system:

In the enhanced system, students can not only see how many people have bookmarked the course posted by the instructor but also allow students to evaluate and leave comments on the instructor's teaching. Students can make better choices and judgments through other users' comments on the course.

Payment system:

Courses offered by lecturers usually indicate their fees, so it's also crucial to integrate payment functionality into the app. Such a design also provides some protection for students.

Schedule and reservation system:

In the add-on system, tutors can set up a daily schedule of their available time to be displayed to students. Once the student has decided on a tutor, he or she can use the information in the schedule to choose a time when the tutor will be available for tutoring.

Community discussion area:

In the enhanced system, the app will have the interface to set up a community discussion, where students can post some of their questions in the community, and if a capable tutor sees the question they can answer it and help. This will make it possible to ask small questions without having to spend a lot of time with a tutor for one-on-one counselling.

Multilingual support:

Provide localized content and interfaces to accommodate students and tutors from different countries and regions.

9. Conclusion

9.1 Purpose

The purpose of this project is to provide a two-way connection between students and tutors with an app that will help students who want help with their academics as well as give a platform and resources to tutors who are willing to help. We truly hope that this program will be beneficial to all users.

9.2 Key findings

We've found that Flutter offers excellent cross-platform support in front-end development. Compared to traditional dual-platform development, Flutter can significantly improve development efficiency and ensure a consistent user experience for apps on the Android platform. From a back-end perspective, Java excels in building server logic due to its stability and mature ecosystem. Combined with the MySQL database, we can ensure reliability and efficiency in data storage. We also found that MySQL offers powerful queries and flexible data structures compared to other popular database systems, making it particularly suitable for medium to large applications. Overall, the combination of Flutter, Java, and MySQL provided us with a powerful, efficient, and reliable development toolset that meets the needs of modern mobile applications.

9.3 Main achievements

In the Studypal project, we have successfully connected students and tutors to create a two-way communication platform for education. The core of the project is to allow students to find a tutor and post their difficulties in the coursework, and also to provide a platform for tutors to educate and offer help to others, this two-way connection allows for a high level of activity and usability.

In order to make the user's finding process more accurate, we have included a highly optimized search function in the project, equipped with multiple types of filters. This way students can find the tutor who best meets their requirements by the number of favourites in the course, as well as the qualifications of the tutor. Tutors can also use keywords to find students they can help.

Considering that users may be interested in more than one course at the same time, we also added a favourites feature to the project so that users can save their favourite courses and view them later. Secondly, the publish feature allows teachers and students to share their needs and offers simply and directly.

We also paid special attention to the personalization of the tutor side. Therefore, we have set up a detailed profile page for tutors, which not only allows users to customize their information but also ensures that tutors can more accurately present their strengths to students.

In short, studypal is more than just a platform for students to communicate with each other, it is a comprehensive user-learning community that aims to provide an optimal teaching and learning experience for both students and tutors.

10. References

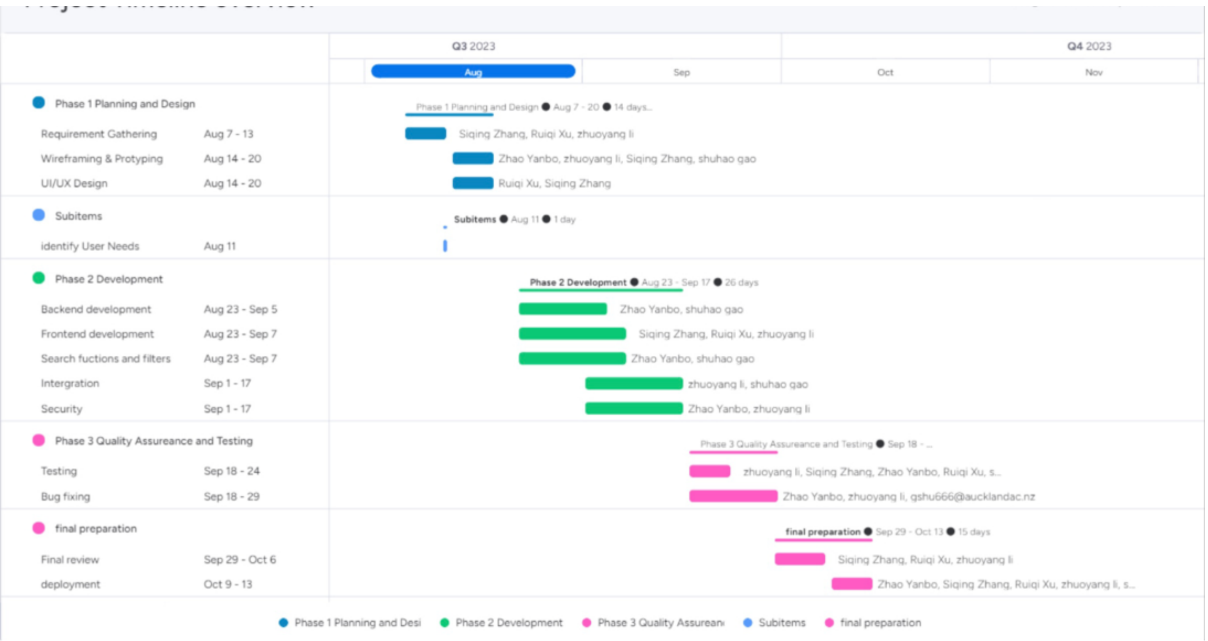
1：Akyuz, Y. (2020). Effects of intelligent tutoring systems (ITS) on personalized learning (PL). *Creative Education*, *11*(6), 953-978.

2：Demirbilek, N. (2023). Satisfaction levels, communication situations, and difficulties encountered by university students regarding Distance Education. *Education and Urban Society*, *55*(5), 593-616.

3：Fettweis, G. P. (2014). The tactile internet: Applications and challenges. *IEEE vehicular technology magazine*, *9*(1), 64-70.

11. Appendices

A. Gantt Chart



B. Link to final build

https://github.com/uoa-compsci399-s2-2023/capstone-project-team-18

C. Link to video

https://i.ii.immo/team18projectdemovideo

D. Authorship table specifying the sections of the report each team member worked on

section	Author(s)
1. Title Page	Ruiqi Xu
2. Executive Summary or Abstract	Zhuoyang Li
3. Table of Contents	Zhuoyang Li
4. Introduction	Siqing Zhang
5. Background	Siqing Zhang
6. Specification & Design	Shuhao Gao, Yanbo Zhao
7. Implementation	Yanbo Zhao,Shuhao Gao
8. Results & Evaluation	Ruiqi Xu
9. Future Work	Ruiqi Xu,Zhuoyang Li
10. Conclusion	Shuhao Gao,Yanbo Zhao
11. References	Zhuoyang Li
12. Appendices	Zhuoyang Li