INTERIM REPORT OF AI STUDENT ADVISOR

COMP4801 Final year project

20/01/2019

Chan Lap Kiu 3035269784
Kwok Cheuk Lum 3035269992
Woo Chong Yiu 3035271012
Chau Chun Wing 3035281275
Abstract

Which major or course to choose is a question always faced by university students. Currently, there are several online tools and databases providing information and advices on choosing majors or courses. However, these databases just display the data without processing them. This project aims to develop a mobile application that consists of an AI grade prediction engine and an academic advising community in order to help university students in course selection.

AI technologies, including machine learning or text mining, can find out students’ strengths by utilizing a huge amount of data. Tailor-made feedbacks and suggestions on study paths can then be provided to students according to their strengths. Moreover, the academic advising community provides an online platform that allows students to discuss problems on academic planning and course selection. The data generated in the academic advising community can also serve as learning data to improve the accuracy of the AI service.

This report talks about the design, methodology and current progress of the project. Currently, the academic advising community function of the mobile application is developed. This MVP is introduced to gain feedbacks for future development. This report also formulates the development of the neural network for machine learning and the testing of the system in the next stage of the development.
Acknowledgement

We would like to express our greatest gratitude to the following people for their help. Without their help, this project might not be able to be completed.

Dr. T. W. Chim: We would like to thank Dr. Chim for being our project supervisor. He was always supportive and gave us timely and helpful advice.

The staff of the department: We would like to thank the staff, especially staff of student office for providing past students’ academic performances to us. Without the data, our AI model could not be trained.

HKU CS students: We would like to thank every HKU CS student who had contributed in our project by providing valuable real-world data of past academic performance and user feedback. Without the data, our project is not achievable.
# Table of Contents

Abstract ........................................................................................................................................ 1  
Acknowledgement ....................................................................................................................... 2  
List of Figures .............................................................................................................................. 5  
List of Tables ............................................................................................................................... 5  
Abbreviations .............................................................................................................................. 6  
1. Introduction ............................................................................................................................. 7  
   1.1. Background ....................................................................................................................... 7  
   1.2. Previous works ............................................................................................................... 7  
   1.3. Objectives ....................................................................................................................... 8  
   1.4. Scope ............................................................................................................................. 8  
   1.5. Deliverables .................................................................................................................... 8  
   1.6. Outline of report ............................................................................................................ 9  
2. Methodology .......................................................................................................................... 10  
   2.1. Development Approach ............................................................................................... 10  
   2.2. System Architecture ..................................................................................................... 11  
      2.2.1. iOS .......................................................................................................................... 11  
      2.2.2. Android .................................................................................................................. 11  
      2.2.3. Web service ............................................................................................................ 13  
   2.3. Programming languages, frameworks and services ......................................................... 13  
   2.4. Functions Design ........................................................................................................... 15  
      2.4.1. Grade Prediction ...................................................................................................... 15  
      2.4.2. View posts in order and filter (My Posts/ Bookmarked/ Hot/ Chronological) ....... 15  
      2.4.3. Create post ............................................................................................................. 16  
      2.4.4. Bookmark post ....................................................................................................... 16  
      2.4.5. Add comment ......................................................................................................... 17  
      2.4.6. Authentication ....................................................................................................... 17  
   2.5. Data collection methods ................................................................................................. 19  
   2.6. Project schedule ............................................................................................................. 20  
3. User Interface ......................................................................................................................... 21
3.1. View posts in order and filter (My Posts/ Bookmarked/ Hot/ Chronological) ........ 23
3.2. Create post ........................................................................................................ 25
3.3. Add/ view comments .......................................................................................... 26
3.4. Edit Post ........................................................................................................... 27
3.5. Login/ Register ................................................................................................. 28
4. Difficulties and Risks .......................................................................................... 30
  4.1. Training the neural network .......................................................................... 30
5. Conclusion ........................................................................................................... 31
References .............................................................................................................. 32
List of Figures

Figure 1. An agile software development cycle diagram, showing the 6 major steps to go through during development with “MEET” as the first step to start with [6].................................................. 10
Figure 2. A MVC architecture diagram showing the interactions between the three different types of objects, namely model, view and controller [7]................................................................. 11
Figure 3. A MVVM architecture diagram showing the dependencies and separation of concerns in this design [11].......................................................................................... 12
Figure 4. A REST architecture diagram showing the connection between components [12]..... 13
Figure 5. Algorithm of calculating hot posts [13]........................................................................... 16
Figure 6. Sequence diagram of registration process. ....................................................................... 17
Figure 7. Navigation drawer (left) & bottom navigation (right) layout............................................. 21
Figure 8. Layout of iOS version Facebook app [16]. ........................................................................ 22
Figure 9. Navigation drawer of AI Student Advisor......................................................................... 22
Figure 10. Home screen of the app showing the newest posts. ....................................................... 23
Figure 11. User interface of creating a post....................................................................................... 25
Figure 12. User interface of adding a tag.......................................................................................... 25
Figure 13. A tag “COMP9999” being added................................................................................... 25
Figure 14. User interfaces of viewing comments and adding comments................................. 26
Figure 15. User interface of editing a post...................................................................................... 27
Figure 16. User interface of logging in........................................................................................... 28
Figure 17. User interfaces of registration. ....................................................................................... 28

List of Tables

Table 1. Schedule for this project. ........................................................................................................ 20
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CGPA</td>
<td>Cumulative Grade Point Average</td>
</tr>
<tr>
<td>CS</td>
<td>Computer Science</td>
</tr>
<tr>
<td>GPA</td>
<td>Grade Point Average</td>
</tr>
<tr>
<td>HKU</td>
<td>The University of Hong Kong</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>iOS</td>
<td>iPhone Operating System</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
<tr>
<td>MVP</td>
<td>Minimum Viable Product</td>
</tr>
<tr>
<td>MVVM</td>
<td>Model-View-ViewModel</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>NoSQL</td>
<td>Non-Structured Query Language</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
</tbody>
</table>
1. Introduction

1.1. Background

Every year, students studying university face questions like which major to choose or what course to take. When students are planning their study path and choosing courses, they often make wrong decisions due to the lack of understanding of oneself or incapability to estimate their own strength. According to the report from the University of La Vern, around half of the university freshmen have no idea what to choose as their major when they get admitted to college [1]. Moreover, around one-third of the students do not complete the first subject they are admitted to because they do not think fully before deciding the subject suggested by the Minister for Education and Training of Australia, Simon Birmingham [2].

The current methods for students to get advice in the university like referencing statistics from unofficial websites or listening to seniors’ advice cannot really help students make correct decisions. The reason behind is because those online statistics are not reliable enough and the seniors’ advice cannot fit their own scenarios. These result in students ending up with unsuitable courses and having an undesirable result. As such, this project aims to build an online forum that utilizes AI technologies and give suggestion to students according to their own strength. The engagement of students will generate questions and answers in the forum, thus it can train the AI to increase its accuracy.

1.2. Previous works

There are existing works with similar ideas of producing an online academic advisor, but the technologies used and the main areas of concern are different. These works include UM-IBM SAPPHIRE PROJECT at the University of Michigan and A Proposed Model for a Web-Based Academic Advising System. For UM-IBM SAPPHIRE PROJECT, it aims at using AI technology to process natural language and change natural text into SQL [3] and apply it in an academic advisor so that it can reply the users’ natural language queries. Although this project also integrates the technologies of AI, the focus of this project is to provide a more convenient way for students to get advices but not giving them better advices. It also only improves the accessibility by resolving the users’ natural language queries but not through other means. The Proposed Model for a Web-Based Academic Advising System focus on reviewing what model should be used for an e-Academic Advising System in the modern day so that staff and advisors can follow-up the suggestions given and feedbacks from the students [4]. That project does not investigate on using any AI technologies to give better advices but only give suggestions on how web-based technologies can facilitate human advisors to follow-up on students’ feedbacks. Thus, it does not solve the problem of the suggestions not being accurate enough.
1.3. Objectives

Two objectives were set in order to tackle the aforementioned difficulties.

First, a mobile application that provides appropriate guidance to students when they are deciding on their major or course selections needs to be built by employing existing AI technologies like ML and text mining.

Second, a mobile application with alluring social platform and gamified elements should be introduced to attract university students.

1.4. Scope

The mobile application is only developed to support iOS and Android platforms but not other platforms like Blackberry and Windows because these two platforms have already taken over 96% of the mobile OS market share [5]. Thus, only developing on these two platforms is enough to cover most of the targeted users and meet the industrial standard. Owing to the difficulty in collecting past students’ grades from other departments and faculties, the mobile application only considers the courses that are held by the Computer Science Department in HKU during the current development stage.

1.5. Deliverables

A mobile application with both Android and iOS versions is the final products of this project. Feature of this application includes course grade prediction, course suggestions and an online community for course selection advice.

The course grade prediction feature allows the user to predict his course grade for a particular course. The mobile application predicts the grade by using the data given by the user including his past course grades and current GPA. The course suggestion is a feature similar to the course grade prediction. The AI student advisor predicts the grades that the student will get for all courses and sort the list in descending order of the predicted grades.

A user can also seek for advices and ask questions in the online community if he feels more comfortable in asking his peers or advisors. On the other hand, the AI student advisor will also give advice to the question asked so the user can compare the AI’s answer and other users’ answers. If the user is satisfied with the answers, he or she can choose the one that helps the most as a “Best Answer”. Other users may also upvote or downvote one another’s answers to indicate they are useful or useless respectively. Combining the mechanism of best answer, upvoting and downvoting from all users, the application feed these data back into the AI’s neural network so it can improve on its future predictions.
1.6. Outline of report

The remainder of this report proceeds as follow. The methods on how to build a mobile application and the user interface design are provided and discussed in chapter 2. The whole project’s development approach will be covered first. After that, the choices of system architectures on Android, iOS and web service are discussed and justified. Furthermore, the reasons for choosing Node.js and MongoDB as the backend and database are also provided in this chapter. Data collection methods and project schedule are also discussed. In chapter 3, the current user interfaces of the project are discussed. Then the difficulty we encountered during the development will be offered and discussed in chapter 4. Finally, it ended with a summary and future work in chapter 5.
2. Methodology

The technological choices and how the application is built are covered in this chapter. The agile software development cycle was chosen due to its short and fast iteration cycles which are needed in this project. Software architectures, programming languages and frameworks were chosen carefully with the criteria of fitting the purpose of this project, compatibility with the respective platforms, stability and community size.

2.1. Development Approach

![Agile software development cycle](image)

*Figure 1. An agile software development cycle diagram, showing the 6 major steps to go through during development with “MEET” as the first step to start with [6].*

In this project, agile development method is used (see Figure 1) so that the project is more flexible and adaptive to changes comparing with traditional waterfall approach. It works by breaking the project into smaller parts and then plan, design, build and test for each of these smaller parts rapidly. Each of these smaller parts is called an iteration. After each iteration is completed, that part is deployed to get feedbacks and then improved in the next iteration according to the feedbacks.

This kind of flexibility is needed in this project because a lot of feedbacks will be given by our advisor. The users may also give us feedbacks on whether the application features are attractive and useful enough. The agile development approach allows us to adopt the feedbacks and make necessary changes effortlessly. It also allows us to mitigate potential risks earlier as feedbacks are given earlier.
2.2. System Architecture

2.2.1. iOS

For the iOS mobile application, the MVC architecture is used such that the objects in the application is a model, view or controller (see Figure 2). This architecture is officially suggested in the Apple Inc. documentation as a good design for a Cocoa application. In fact, many of the Cocoa Touch APIs provided by Apple Inc. were built using the MVC architecture [8]. Therefore, this architecture’s performance and compatibility with iOS is the best.

Other than the performance and compatibility advantages that are specific on iOS, the codes written in MVC architecture is generally more reusable, thus making the application more extensible. Moreover, the MVC architecture was introduced in Smalltalk-80 version of Smalltalk that was released in 1980 [9]. It proves that this architecture was widely tested out and so stable that it is still used nowadays.

However, it is commonly known that MVC architecture might cause the problem of having fat or massive ViewControllers in iOS applications. This is because the ViewController classes sometimes consist of all the codes not that does not belong to Model or View. In this project, service layer, categories, subclassing and Interface Builder to leverage this problem.

Despite the fact that fat ViewControllers might exist in the project, the MVC architecture was chosen due to its stability, extensibility and compatibility which outweighs the downside.

2.2.2. Android

The Android version is built using the MVVM architecture. The MVVM architecture is comparatively new to the MVC architecture that was employed on the iOS version as it was only
proposed in 2005 [10]. With the help of the ViewModel component, the architecture prevents fat Activity, which is similar to fat ViewController in iOS, to happen. The ViewModel component can handle the presentational codes that originally exist in an Activity with MVC architecture, so the Activity depends on it but not handling the presentation logic by itself (see Figure 3).

Figure 3. A MVVM architecture diagram showing the dependencies and separation of concerns in this design [11].

Moreover, this architecture is used due to its compatibility with the Android OS so that it can be implemented at ease. The Android Architecture Components released in Android Jetpack also suggests a MVVM architecture and provide features like data binding to facilitate the implementation.
2.2.3. Web service

The web services that are implemented on the backend conforms to REST architecture (see Figure 4) and HTTP. REST architecture provides data in specified format, JSON in this project, which lowers the coupling between the client and the server so that it is more scalable. It was chosen because of this scalability which allows the system not to break as it grows. The REST architecture is also simpler to use so the project can be developed faster.

2.3. Programming languages, frameworks and services

For the frontend mobile application, Objective-C and Java were the chosen languages to develop for iOS and Android platforms respectively. New mobile frameworks including React Native and Flutter were reviewed but they were not chosen because the communities of these technologies are relatively small comparing to the native codes’ communities. It means that there are not enough existing frameworks and support to suit the project’s rapid development. Important flaws were also spotted when applications are ejected from React Native like crashing the app unexpectedly, React Native and Flutter also do not support some native features which might be needed. Moreover, the major benefit of React Native and Flutter, that is the reusability of the codes, cannot be fully utilized in this project because the UI design of the Android and iOS versions might be different to facilitate the habit of different OS users. After these thorough considerations, the decision made was to fall back to using the widely tested and robust native code development. The more modern language, Swift, was not chosen to develop the iOS application because our team is more familiar with Objective-C while Swift does not offer a significant improvement.
Some existing third-party libraries and frameworks will be used to increase our development pace in frontend development. For example, AFNetworking and OkHttp are used on the iOS and Android platforms respectively to make HTTP requests also handle the network errors effortlessly. DragRefreshAndLoadMoreTable and Ultra Pull to Refresh with Load More are used on the iOS and Android platforms respectively to implement the features of “pull to refresh” and “pull up to load more”.

For backend, Node.js is used together with the MongoDB database just as the common combination. Node.js was chosen because it supports async functions which help in improving the backend’s performance. Traditional SQL database was not selected not only because we are more familiar with MongoDB but also because NoSQL databases are generally more scalable. Considering the fact that the project might scale up in the future and the rapid development required, MongoDB was chosen. Since settings up servers is a cumbersome job, a PaaS called Heroku and MongoDB Cloud are used in this project for setting up the backend environment. The backend environment can be set up conveniently with this service which shortens the development time a lot. This is particularly important because the development time span of this project is quite short.

Gitlab is used for managing the source codes of this project, no matter the frontend or the backend codes. It was chosen because it is free to create a private project on GitLab. The branching tools of GitLab also allows the team to develop and write codes simultaneously and combine them in the future seamlessly. GitLab was also designed in a way to facilitate continuous integration and continuous development that fits the agile development approach well.

The choice of AI services is still under consideration, but a few potential choices are spotted including Amazon Comprehend, Alibaba Cloud and Microsoft Azure. The selection criteria include cost, stability and community size. Machine learning will be used by feeding back the results of the suggestions from the online community and the final grades the students received when the semester finishes back to the neural network for training. Text mining will be used to find keywords and generate tags for the student’s questions in the communities so that each question will be stored together with the tags in the database. These tags can be used to facilitate the searching of other students.
2.4. Functions Design

The mobile application consists of six major functions: Grade Prediction, View posts in order, Create post, Bookmark post, Add comment, Authentication. This section proceeds with detailed breakdown of the user flow for each function and data flow of the mobile application for each function.

2.4.1. Grade Prediction

The grade prediction function allows users to get a grade prediction from the application after he inputs the course that needs to be predicted and his past courses’ grades. Then the client application will send a request with the data to the server and the server will use the trained AI model to analyze the data. After a result is generated, the server will return the result to the client and the client will show the grade prediction to the user.

2.4.2. View posts in order and filter (My Posts/ Bookmarked/ Hot/ Chronological)

The view posts function allows users to view posts at the home page of the application, it contains four types of view orders and filters. They are viewing self-created posts, viewing bookmarked posts, viewing hot posts and viewing posts in chronological order.

The client sends a request to the server that specifies how the posts should be ordered first. Then the server sends a response of the first ten posts in the corresponding order (or less if there are less than ten posts in the database). When the user scrolls to the bottom of the mobile application, the client will send a request for the subsequent ten posts. The user can continue to scroll until there are no more posts to load.

Hot posts are ranked by the formula (see Figure 5) which references the algorithm from Reddit, a popular open source forum in the world. The variable x is modified from the number of upvotes and downvotes to the number of comments in this mobile application because there are no upvotes and downvotes in this mobile application. In this algorithm, submission time plays an important role where newer stories will be ranked higher than older ones. The newest posts will almost always be ranked at the top. Another feature of the algorithm is that the first ten upvotes, downvotes or number of comments is as high as the next hundred as logarithm is used.
where \( y \in \{-1, 0, 1\} \)

\[
y = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{if } x = 0 \\
-1 & \text{if } x < 0 
\end{cases}
\]

and \( z \) as the maximal value, of the absolute value of \( x \) and 1

\[
z = \begin{cases} 
|x| & \text{if } |x| \geq 1 \\
1 & \text{if } |x| < 1 
\end{cases}
\]

we have the rating as a function \( f(t, y, z) \)

\[
f(t, y, z) = \log_{10} z + \frac{yt_s}{45000}
\]

Figure 5. Algorithm of calculating hot posts [13].

2.4.3. Create post

The create post function allows users to create a post in the community. The user clicks the create post button first, and then enter the title, content, and tags. The client will send a create post request with title, content, tags and user id to the server, then the server will add the document to the database and return a successful message if it successfully adds the document to the database. If the client receives the successful message, the mobile application will return to the home page which is the page that shows the latest posts, so the user is able to see his post is on the top of the page. Otherwise, the client-side will pop up an error message according to the error message received from the server.

2.4.4. Bookmark post

The bookmark function allows users to bookmark a post that he wants to bookmark when the user clicks the bookmark button in the client-side application, the application will send the bookmark request with the post id. Then the backend server will add the user id to the post document in the database based on the post id received and return a successful message if it successfully adds the document to the database. If the client-side receive the successful message, the client-side application will change the bookmark button to black to show the post is bookmarked. Otherwise, the client-side will pop up an error message according to the error message received from the backend server.
2.4.5. Add comment

The add comment function allows users to reply a post when they enter a post in the client-side application. The user only needs to fill in the content of the reply, then the client-side application will send a reply post request to the backend server with user id and post id. After the backend server receives the reply post request, it will insert the document into the comment database, and return a successful message if record inserted successfully. If the client-side receive the successful message, the client-side application will append the comments on the page directly, while not request the backend server to send all the comments in the post again as it will waste some performance of the backend server. Otherwise, the client-side will pop up an error message according to the error message received from the backend server.

2.4.6. Authentication

The authentication function allows users to register or log in the application. The user cannot use other functions before logging in.

Figure 6. Sequence diagram of registration process.
For the registration process, the user needs to fill in his username, password, and other personal information in the mobile application. The entered username and password are first validated by the client-side application, so that they are not empty, and the password is longer than six characters. Then the client sends the registration request to the backend server. The server first checks whether there is a duplicate username in the database, then hash and encrypt the password and insert the document into the database. The server then sends an email containing a one-time token to the user’s email to verify his or her identity. In order to prevent hackers from guessing the one-time token, this unauthenticated account and one-time token are only stored in the database for 15 minutes. If the user does not enter and submit the correct token within the time period, he or she needs to start the registration process again. If the one-time token verification is successful, the user is automatically logged in and can start using the application directly.

Passwords are hashed before saving into the database. By hashing a password, a hacker cannot recover the original password even if he or she has access to the value stored in the database.

Bcrypt is used to hash and encrypt the passwords rather than other alternatives in the market as it was specifically designed for passwords and thus have better performance. Furthermore, a 128-bit salt [14] is included and the iteration count can be modified in the algorithm of bcrypt, so it is immune to rainbow attacks and brute force attacks.

To login, the user needs to enter the username and password that he or she used to register previously. The application first validates the input and then send the login request to the server. Then backend checks if both the username and password are same as the database by hashing and encrypting the received password using the bcrypt algorithm again. Finally, the server will return a successful message if both username and password are same as the database. If the client-side receive the successful message, the client-side application will login and redirect user to the home page of the application which shows the latest posts. Otherwise, the client-side will pop up an error message according to the error message received from the backend server.
2.5. Data collection methods

The application adopts ML models in the grade prediction function. Data of students’ past performance need to be collected to train the artificial neural network and to verify it. The quota for undergraduate place in the Department of Computer Science of HKU is 120 [13]. Since the program is changing quite fast, only 10 years of undergraduates are considered. Therefore, the population is around 1200. The sample size that is needed to attain a confidence interval of 5 and a confidence level of 95% is 291. An additional 10% of samples are needed to verify the accuracy after ML training. Therefore, the total amount of samples needed for building up the database to cater machine learning and verification is $291 + 291 \times 10\% \approx 320$.

Two methods will be used to collect the initial data. First, questionnaires will be distributed to ask students about their GPA and some individual course grades. Second, grades, including the GPA and individual course grades, of past graduates are collected from the Department of Computer Science of HKU.

More data in the artificial neural network gives more accurate predicted grades. This application is designed to learn from users’ data continuously after it has been launched. The data collection method adopted for this purpose is to obtains students’ previous grades from HKU Portal directly with the access right granted by them when they log in to the system. The application will be linked up with the HKU Authentication Service, which supports Single Sign On (SSO) authentication, allowing users to login with their HKU Portal accounts. Once signed in successfully, HKU Authentication Service generates a token and sends back to the application. Hence the user can be identified meanwhile their previous grades can be obtained. In addition, the application will automatically update the record when grades of the semester have been released. Users must agree on the privacy policy that subscribing our service means they agree to grant permission to the system to access to their grades.

Users who do not prefer using their HKU Portal account can input their previous grades manually. Users are required to fill in their past course grades when they register with emails. Yet, this registration method is not recommended because data error can easily be made with human process. The average number of records of 10 courses in a year required to be inputted by hand is huge and the frequency of at least twice a year required to update the record is high. Input errors may cause chain effects in the artificial neural network resulting in garbage-in-garbage-out. Data validation methods will be adopted to minimize the influence of invalid data to the artificial neural network.

The content in the online community will be collected as a feedback to the AI. Selected comments of users will be adopted to train the artificial neural network to give more accurate answers.
2.6. Project schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept</td>
<td>Deliverables of Phase 1</td>
</tr>
<tr>
<td>Oct</td>
<td>Development of mobile app (Basic UI, GPA calculator, Personality test)</td>
</tr>
<tr>
<td>Nov-Dec</td>
<td>Development of mobile app (Forum feature)</td>
</tr>
<tr>
<td>Jan</td>
<td>Deliverables of Phase 2</td>
</tr>
<tr>
<td>Jan-Feb</td>
<td>Data Collection (Questionnaire) + AI model research + training + mobile app testing</td>
</tr>
<tr>
<td>Feb-April</td>
<td>AI model research + training + migration with mobile app</td>
</tr>
<tr>
<td>April</td>
<td>Deliverables of Phase 3</td>
</tr>
</tbody>
</table>

*Table 1. Schedule for this project.*
3. User Interface

![Figure 7. Navigation drawer (left) & bottom navigation (right) layout.](image)

Layout design is crucial to a mobile application for there is no user manual nor tutorial provided to users. Users use the application according to what they perceive from the UI. There are norms in UI design, such that people are used to the meaning of those representations [15]. For example, a button with three horizontal stripes means navigation drawer, and a bar at the bottom means switching functions (see Figure 7). Both representations are widely used for switching between functions and exhausted the choices of representation for such purpose. The application implemented the navigation drawer layout.

Navigation drawer layout is a layout which shows all of its functions in the navigation drawer, typically on the left side of the screen. This layout is commonly seen in mobile applications developed by Google. The development strategy of Google in developing mobile application is to scale down the scope of each application so that the focus of the applications is clearly defined. Although there are different functions listed in the navigation drawer, they share similar layout.
Bottom navigation layout is a layout which shows the application’s major functions in the bottom bar which allows users to switch to different functions by one simple click. Most mobile applications adopt this layout style. This layout is suitable for mobile applications with multiple featured functions and the users are always switching between functions. The functions usually have very different layout, which users can identify the significant differences when switching between functions. Taking the iOS version of Facebook application as an example, the bottom bar enables users to switch to different modules, including homepage, friend requests, massages and notifications (see Figure 8), which have completely different layout designs and nature of function.

Navigation drawer layout is implemented in AI Student Advisor to simplify the UI of the application. The application has a single focusing function. It features the online academic advising community where the grade prediction engine will be integrated to. Although there are functions for switching the methods of reading posts in the online community, all functions share similar layout. Users can be more focus in the main function with the current UI design.

AI Student Advisor implements the navigation drawer layout (See Figure 9). All functions are presented in the navigation drawer in the left. The top part of the navigation drawer shows users’ information including their icon (had not been set by the user in Figure 9), nickname (“Ben”), and cumulative grade point average (CGPA) (“4.3”). The menu in the navigation drawer is first followed by sorting methods of the online community, then followed by other functions of the mobile app. Functions can be launched by selecting the respective links in the application drawer. Details of the layout of each specific function will be discussed in the following sections.
Users can close the navigation drawer by 3 ways: pressing the back button (<) on the top left corner, pressing at the diminished area on the right of the navigation drawer and swiping to the left of the screen. In addition, Android version supports the fourth way to close the navigation drawer, which is by pressing the back button on mobile devices.

3.1. View posts in order and filter (My Posts/ Bookmarked/ Hot/ Chronological)

![Figure 10. Home screen of the app showing the newest posts.](image)

Figure 10 demonstrated the basic interface of a post. Taking the top post as an example, each post includes a title (“Mobile app”) in the first row, content (“Is the mobile app course a good grade course?”) in the second row, tags (“Mobile App”) in the third row and number of comments (“0” at the bottom right corner next to the comment icon (评论)). There is also a bookmark button (Bookmark) and comment icon (评论) in the bottom row.

Pressing the bookmark button (Bookmark) performs the “adding to bookmarked” function to posts which have not been bookmarked but “removing from bookmarked” function to bookmarked posts. Bookmarked posts show a highlighted bookmark (Highighted bookmark) (“Which course should I take?” in Figure 10) while posts which have not been bookmarked show a normal bookmark (Normal bookmark) (“Mobile app” and “Which common core course to take?” in Figure 10). All bookmarked posts are shown in the list of “bookmarked” automatically when choosing “Bookmarked” in the navigation drawer.
Pressing on any area (except the bookmark button) of a post shows the details of the post. The app will start a new screen showing the details of the post, all comments and allowing users to add new comments. Details of the comments screen will be discussed in Section 3.3.

The user interface provides four types of view orders and filters, including self-created posts, viewing bookmarked posts, viewing hot posts and viewing posts in chronological order. The default screen is showing the newest posts in the online community. User can select other sorting methods of the posts in the navigation drawer (see Figure 9), which can be opened by clicking the menu button (≡) on the top left corner of the screen (see Figure 10). In the navigation drawer (see Figure 9), “My Posts” means to display posts in the online community which are created (see Section 3.2) by the user in chronological order. “Bookmarked” means to display posts which are marked as bookmarked by the user. “Hot” means to display posts in descending order of hotness. “Forum” means to display posts in the default order, which is in chronological order of the creation the posts.

Viewing posts in different orders or filters does not affect the layout of the online community, but only the content and order of the posts. Viewing orders and filtering methods can be changed in the same way as in the home screen in any other screens listed above.
3.2. Create post

When a user clicks on the add post button (+) at the top right corner on any page of the online community (see Figure 10), the app will create a new screen of creating a post (see Figure 11).

Figure 11 shows the user interface of creating a post. Users need to enter a post title in the top text box (with the placeholder “Post Title”). The larger text box is for entering the content of a post. A tag can be added by clicking the “Add” button in blue next to “Tags”, then a dialog box (see Figure 12) will pop up. Users can set the text to be displayed in a single tag by typing in the text box of the dialog box (with the placeholder “Please enter a tag”). Users can complete adding a tag by clicking “Add” or return to the previous screen without creating a tag by clicking “Cancel” in the dialog box. Tags successfully created will be listed below “Tags” (see Figure 13). Tags can be deleted by clicking on the created tags. Multiple tags can be created by repeating the above procedures. A post can also be created without any tag. After filling all the required fields, users can create the post by clicking “Post” in the top right corner or cancel the creation and return to the community screen by clicking the return button (<) at the top left corner on the creating post screen (see Figure 11).
3.3. Add/ view comments

![Figure 14. User interfaces of viewing comments and adding comments.]

After a user pressed on any area of a post in the home screen, the app shows the screens of viewing and adding comments (see Figure 14). This page shows the details of the post. Details include creator of the post (“ben”), title of the post (“Mobile App”), content of the post (“Is the mobile app course a good grade course?”) and number of comments (“1” in the bottom right corner next to the comment icon (-square) (see Figure 14a)). There is also a bookmark button (square) which performs the same function as the bookmark button (square) in Figure 10.

Under the details of the post are the comments (see Figure 14a). No comments are shown in this area if there are no comments. Each comment shows its creator (“Victor”) and content (“Yes, best course ever!”) (see Figure 14a). New comments can be created by tapping at the edit text area which is docked at the bottom of the screen. The keyboard pops up after tapping the edit text box and push it up on top of the keyboard (see Figure 14b). Users can enter the content of the comment in the edit text box with the virtual keyboard and submit the comment using the send button (square). The send button is dimmer in color and disabled until the user enters a string that contains characters other than spaces or newlines. After submitting the comment, the virtual keyboard is hidden, and the newly created comment is added at the bottom of the list (see Figure 14a). The screen will automatically scroll to the bottom to display the newest comment if it was not.
If the user is the creator of the post, an edit button (📝) is shown in the top right corner (see Figure 14a). Users can edit the post by pressing the edit button. An edit post screen will be launched. Details of the edit post screen will be discussed in the next section.

3.4. Edit Post

![Figure 15. User interface of editing a post.](image)

Users can edit a post in the edit post screen (see Figure 15) navigated from the comments screen. The screen is nearly the same as the create post screen so that users do not need to adapt to the UI again. The only difference is that it automatically fills in the original post title (“Mobile app”) in the smaller text box, the original post content (“Is the mobile app course a good grade course?”) in the bigger text box, and the original tag(s) at the bottom. Users can edit the post title, content and tags just like how they create a post (see Section 3.2). After editing the post, users can post the amendment by clicking “Post” at the top right corner to overwrite the original post or cancel the amendment by clicking the return button (←) at the top left corner.
3.5. Login/ Register

![User interface of logging in.](image)

*Figure 16. User interface of logging in.*

![User interfaces of registration.](image)

*Figure 17. User interfaces of registration.*

The app shows the login screen (see Figure 16) when users launch the app for the first time or after logging out. Users can login by pressing “Login” button after entering their username and password.
If a user does not have an account, he or she can click “Register” in blue at the bottom of the screen (see Figure 16). Then a registration screen is shown (See Figure 17a). After filling in his or her email and password, he or she can click “Continue” in blue at the bottom of the screen to navigate to the next step (see Figure 17b). The server generates a one-time token via email after receiving the create-account request to verify that the email address the user provided is a valid email address. The user has to enter the token into the text box and click “Continue” in blue under the text box. They can click “Resend token” in blue at the bottom of the screen if they have not received the token. After verification, a personality test is shown (see Figure 17c). Users can complete the test and login to the app or skip it for now by pressing “skip” in blue at the top. Then the registration process is completed, and the application redirects the user to the home screen.
4. Difficulties and Risks

There are several Difficulties faced during the development of the application, including training the neural network and collecting useful data.

4.1. Training the neural network

The major difficulty faced in this project is training the neural network. First, the team is not familiar enough with the machine technologies and services, so it is difficult for the team to implement an ML model. There are currently two mitigation strategies for this problem. The first is to do more research on ML to increase the knowledge of it and understand the method behind to build it. Another one is to find the professors with relevant background for help. The last method is to use AI technologies other than ML to do the task.

On the other hand, it is hard to collect useful data to train the neural network as only students that graduated have the data that is needed in this project. It is typically hard to contact them. It is partly solved by asking the Computer Science department for giving the team valuable data for training the neural network, but a lot more data is needed to improve the AI student advisor’s accuracy.
5. Conclusion

The problem of students not being able to choose suitable courses for themselves is significant. This project aims at using AI technologies to provide tailor-made and accurate advice to students in choosing the right course. Another objective is to introduce intriguing social platform and gamified elements to it so that universities students are willing to use it.

The overall progress of the project is on schedule. The screen designs are completed. The frontend mobile applications are nearly completed, and it can work with the backend APIs, but testing is needed in the future to ensure the reliability of both frontend and backend.

However, the future development pace of the project should be sped up due to the unfamiliarity to machine learning technologies. Investigations on machine learning services and models can be started earlier so as to mitigate the potential risk. The team may need to ask for professional advices on machine learning services to resolve the potential risk stated above.
References


[2] Sing Tao Daily, “大學生選科後中途變卦 僅三分一在六年內畢業 ,” 大學生選科後中途變卦 僅三分一在六年內畢業, Jan. 20, 2017. [Online]. Available: http://std.stheadline.com/daily/article/detail/1537743%E6%B5%B7%E5%A4%96%E8%8F%AF%E4%BA%BA-%E5%A4%A7%E5%AD%B8%E7%94%9F%E9%81%B8%E7%A7%91%E5%BE%8C%E4%B8%AD%E9%80%94%E8%AE%8A%E5%8D%A6+%E5%83%85%E4%B8%89%E5%88%86%E4%B8%80%E5%9C%A8%E5%85%AD%E5%B9%B4%E5%85%A7%E7%95%A2%E6%A5%AD. [Accessed: Oct. 16, 2018]


