THE UNIVERSITY OF HONG KONG

DEPARTMENT OF COMPUTER SCIENCE

COMP 4801 - Final Year Project

A Mobile and Intelligent Student Interview System for HKUCS

Project Plan

September 2015

Supervisor: Reynold Cheng
Lo Ming Fai
Vinny Wong
Lau Pui Lam(Michael)
YOU Dongguang(Nick)

Content To be corrected

Abstract(finished)

Project Title: A Mobile and Intelligent Student Interview System for HKUCS

This Project plan is a comprehensive preview for our Final Year Project. HKUCS stands for the Computer Science Department of the University of Hong Kong.

Our major objective is to add the online interview function to the webpage, and more importantly to build two new components, i.e. the iOS interview assistant application and a data mining tool. The iOS application and will be written in Swift will communicate with the backend through a Restful web service. Potential challenges will also be addressed and a number of possible solutions will be proposed. Data mining tool is to predict the result of interview applicants based on the current data provided by HKUCS department.

In this project plan, we include the methodology, related works,

deliverable, challenges to explain how we implement the project. In addition, schedule and work division is included to describe different phase of the project and how we collaborate with each other.

To wrap it up, we have every reason to believe that this new interview system will not only outperform the previous one but also significantly improve the efficiency of the admission process and the quality of admitted students.

Introduction(finished)

Every year since HKU was founded, a huge number of students have come to HKU to pursue their various degrees. As a result, the application and admission process has caused too much administrative work for HKUCS employees. In particular, the process for postgraduate students has to involve much engagement of CS professors, who could otherwise spend the time on teaching and research instead. Below is a chart illustrating the admission process of postgraduate students in HKUCS.

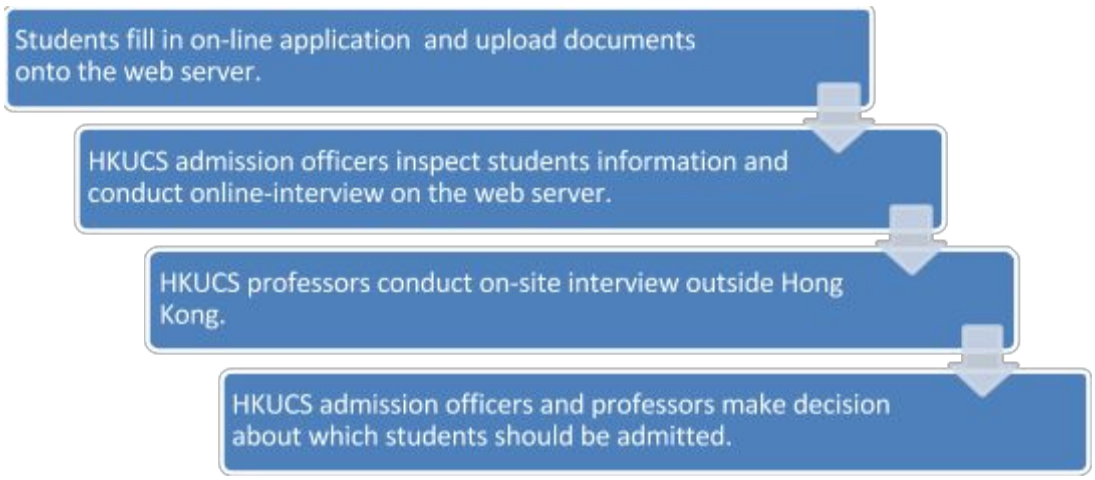

Figure 1

Hence a high-performance system is demanded to facilitate the interview process, and select the best students as well. In such context, a team of CS students created an interview system last year to aid for the efficiency. Although it turned out to be a big success, many flaws and defectives were left, and there remains much room for improvement.

Therefore this year our team is devoted to developing a more user-friendly system on top of the previous version, and more importantly, add new features that better satisfies the use of professors and applicants.

Previous Works Evaluation

Since our system will be based on the existing one, we will have to do a

comprehensive study of the existing system. As far as what we know, the current system has the following flaws.

First of all, the current system doesn't provide effective ways of making a comprehensive comparison between two students and this has to be done manually. Besides, the current system doesn't do well in matching students with professors either. Each professor has to look for suitable students all by themselves.

Second, on-site interviews always take place outside Hong Kong. The current interview system only provides web-based application which has many requirements for users in terms of web environment. To make it worse, accessing to HKU server is prohibited in places like Mainland China.

Third, the online interview module is not being used because many bugs have been reported in that part. Below is a graph showing a typical flow of application and admission process.

We will study the existing project further in early October and find as more defects as possible.

Related Studies

In addition to the existing project, solid background in data mining and machine learning will be a prerequisite to analyze enormous past applicants information and make predictions accordingly. Recommender system[1], a sub-field of data mining which seeks to predict the 'rating' or 'preference' that a user would give to an item, is one of the previous studies our project is closely related to. Besides, the logistic regression model[2], where we yield a Boolean value from a series of factors, is also relevant.

Scope(finished)

This Project is aimed to facilitate the admission process by enhancing the usability of the current interview system in many different ways. We will make it significantly easier for professors to compare candidate students and make decisions about which to choose. The website will interact with the user more intelligently and sufficient assistance will be provided to the on-site interview outside Hong Kong.

In order to accomplish these goals, a list of features has been devised.

1. A mobile version will be implemented to support iOS devices having

[1] see https://en.wikipedia.org/wiki/Recommender_system
[2] see https://en.wikipedia.org/wiki/Logistic_regression

IOS 7 or higher in order to serves the need of on-site interview. In particular, through the IOS app, the user will be able to access interviewee information as well as the interview question bank, and put down his assessment of each interviewee.

2. We will make use of all accessible data and perform data mining procedures to build a powerful prediction engine of student performance based on their experiences and past performance. The system should be able to deduce a standard score for applicants according to a particular input of requirements.
3. The web server will be further refined and the online video conference function(supporting Chrome, Firefox, iOS) will be implemented. The prediction engine will also be integrated into the server to recommend suitable students to professors. In addition, when a user hovers mouse on a student label, the information of that student will be shown.
4. The function of comparing two students will be added to the web server. Specifically, the user will be able to drag the labels of two students to the comparison interface and a comprehensive summary of both will be displayed side by side.

Currently we already have access to many useful resources including the applicant data in the past and the source code of the existing system.

Methodology(to be modified, this part is supposed to talk about how we are going to implement the aforementioned features, not why)

Web:

Online question bank – The back-end of the system should be implemented by PHP which align the programming language with the existing system. The database of the online question bank will be combined with the existing database. This function provides an online application test for the applicant when they make an application. The test results will be stored in the database for further data analysis.

Video conference for pre-site interview – the function implemented by WebRTC. The reason we choose this because it is an open source project that supports Real-time communication across a lot of popular browsers(Chrome, Firefox and Opera) and mobiles devices(Android and IOS) (webrtc.org).

Features developed from existing data - Mouse-hovering displaying summary of the applicants, dragging two students and displaying their summary side-by-side, displaying data analysis result. (briefly explain how?)
Web Service:

An API will be implemented for data communication between the iOS application and database of the interview system. PHP and Flight framework will be used in the implementation, because we want to have consistent programming language with the previous web applications and Flight framework is a fast, simple, extensible framework for PHP. Flight enables you to quickly and easily build RESTful web applications. In order to sync with the web application, we will make use of the existing database from the previous web application.

The API will provide a number of URLs for the IOS devices to communicate to. The iOS will make HTTP requests on the URLs and get/post data in JSON format. The API will provide the services for getting list of students according to the sorting and filtering parameters, getting the basic info and the url to the CV of a specific student and getting interview schedule of the user. It will also provide channels for users to post comments and rate students and request suggested questions for on-site interview.

IOS:

Language:

There are currently two languages at disposal for writing native IOS application. One is Objective C which has been the major language for writing IOS application since iPhones came out. Another is Apple's newly developed programming language Swift. Primarily, Swift will be the language used to write this project's IOS application.

The foremost reason for choosing Swift over Objective-C is that Swift is cleaner and easier to understand, as illustrated by the simplest "hello-world" example (Thomas). Another reason is that Craig Federighi, Apple's senior vice president of Software Engineering, believes that Swift will be the programming language IOS application developers would use in the upcoming two decades (Wall). These aspects are utterly vital as our system would have a relatively long lifetime and our team would not be in charge of maintenance.

One possible downside of using Swift is the loss of many powerful third party libraries written in Objective-C cumulated over the years. However, this problem is addressed by the interoperability of Swift and Objective-C. Simply by including the Objective-C headers in the bridging header file, the Objective-C codes will be exposed to Swift without any modifications (Apple Inc.). Xcode would even prompt you to create a bridging header file upon inclusion of Objective-C Codes. Therefore, in sight of its cleanness, popularity and interoperability with Objective-C, we have decided to use Swift as the primary language of our application.

Implementation

The application will be using NSJsonSerialization to obtain Json data from the Web Service Backend. Iis built into iOS and would support devices having iOS 5 or higher, which is sufficient for the project's scope. Once the user has logged in and obtained the authorization token for the web service, list of users will then be displayed at the front view of the application through the use of UITableView and data controller (Figure 1(a)).

By using the SWRevealViewController, a slide-out navigation should be provided for users to filter the list. Upon filtering the list, a request should be made to the web service based on the parameters set. The returned data will then be passed to the data controller to update the UITableView.

For each student item of the UITableView, users should have the option to start interview with the student. Users will then be directed to another view in which there will be a UIPageViewController. From that, professors can navigate through student's profile, his CV (implemented using UIWebView), his question banks' answers and others' comments on the student. Beneath the UIPageView, users should be able to flip through suggested questions generated by the backend and leave down comments and ratings(Figure 1(b),(c)).

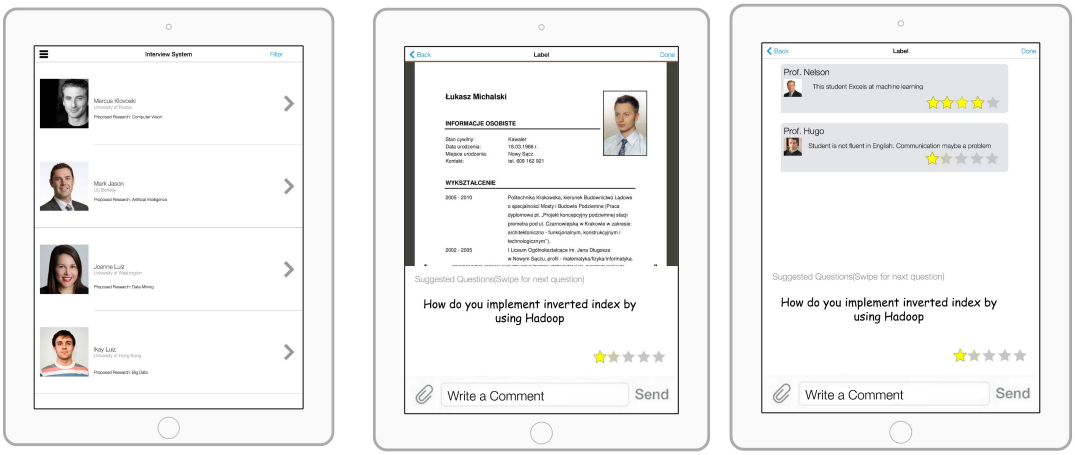

Figure 1(a) UITableView for displaying list of students. Figure 1(b)(c) For use during interview. the top part can be swiped for different info.

Data mining

The objective of data mining is to find out interesting patterns among admitted students and gain insights on what professors are really looking for.

Logistic regression is a model where we can study how a list of factors can affect a boolean variable. In our case the boolean variable would be whether a past applicant was admitted and the factors would be data of applicants such as their gpa. The model helps us to find the weight of each

factor and thus indicate how these factors affect the decisions of professors, which fits our needs.

We will first collect all information of past applicants and retrieve auxiliary data from sources such as university ranking. we will reorganize all the information and store them into our database. Data of applicants for different research area will be separated into different tables. Next we will construct logistic regression models for applicants in different research area. Then we will keep calculating the goodness of fit of the model and perform backward and forward elimination to get rid of insignificant factors, until we find the model where the accuracy is maximized. We will measure the accuracy using C_p criterion[3].

We will use the final model as the engine to predict whether a given applicant should be recommended to a professor. We simply substitute the applicant's data into the model and compute the boolean value.

The limitation of this model is that it can't provide an accurate rank among recommended students. In comparison, linear regression doesn't have that limitation but it requires a numerical representation of the dependent variable when constructing the model. However, since we only know whether an applicant was admitted but cannot compute a score for each admitted students, linear regression is not feasible here.

On the other hand, we will make use of the market basket model because it can find interesting common characteristics among admitted students and matches our objective. Specifically, we will treat each applicant as a basket which stores all his/her information as individual items. We will add an extra item named "admitted" to all admitted applicants. Next, we will employ the PCY algorithm to find all frequent itemsets containing the "admitted" item and deduce association rules. A good example of such rules would be that students who have went abroad for exchange has a bigger chance of being admitted.

Note that this method differs from logistic regression in the sense that this method emphasizes on non-scalar properties such as the aforementioned exchange experience. In addition, there could be other useful tools and we will study further to identify alternatives.

Evaluation

To assess the performance of our system, we will test the three components separately. For the web and IOS application respectively, we will recruit 20 volunteers and ask them to go through all the test cases we designed. Afterwards we will ask them to select a number in the range from one to ten to indicate how satisfied they are.

In particular, for the web, we will ask the volunteers to go through the

[3] see https://en.wikipedia.org/wiki/Mallows%27s_Cp

same test cases on the old system before they give their feedback. Let U1 denote how a random user would like the old system and U2 denote that for our new system. We will consider our project successful if we can reject U1>=U2 at 90% significance level.

Another criterion is the comments and evaluation from the professors and admission officers. Similarly we will ask them to select a number in the range from one to ten to indicate how satisfied they are with our system compared with the previous version.

Deliverables(use pictures to illustrate)

A new version of web-based applications that

- filters(?) and makes comparisons between students with ease.
- convenient to gather applicants' data and view student's profile.(? previously supported already?)
- a big data tool should provide a standard scale for comparison on applicants from different background.
- supports online video interview

An iOS application which

- links with the backend of the interview system
- allows viewing of student's profile
- assist professors in interview by providing functions like commenting, rating and providing suggested question.

Challenges

The major challenge of the system lies in deducing a general score for applicants. The reason is that, so far, there is only a limited amount of admitted student record. The sample size may not be large enough to generate an accurate score. The result of standard score model may contain errors.

To tackle the challenges, we will try to combine the use other sources of rules deducing from other similar student performance prediction research with the historical student history which minimize the errors inducing by the limit amount of data.

WebRTC only supports limited number of browsers such as Chrome, Firefox.Some function of supported browser maybe blocked in the Mainland China.To tackle the problem, Flash may be used when WebRTC is not working in the video conference. We will also ask people from the Mainland China to help testing the video conference function before launching the

system.
What about web and IOS?

Schedule

30 Sep 2015	**Gather User Requirements** **Hosting Set Up** **Draft project plan**
4 Oct 2015	**Deliverables of Phase 1 (Inception)** • **Detailed project plan** • **Project web page**
Oct	**Study the use of data mining algorithms and tools** **Design the database structure and UI of applications** **Study the WebRTC plug-in**
Nov - Jan	**Development of Beta Version** • **Implement the database and back-end service** • **Implement a new version of web-based applications** • **Implement the mobile applications** • **Implement the standard score model**
24 Jan 2016	**Deliverables of Phase 2 (Elaboration)** • **Preliminary implementation** • **Detailed interim report**
Feb - Apr	**Optimization and Testing**
17 Apr 2016	**Deliverables of Phase 3 (Construction)** • **Finalized tested implementation** • **Final report**
18-22 Apr 2016	**Final presentation**
3 May 2016	**Project exhibition**

Work Division

Mobile application development	Vinny Wong
Web-based applications	Michael Lau
Big data analyzing tools	Nick You
Back-end services for other functional parts	Lo Ming Fai

Conclusion

This project is believed to provide an easy to use interview system across all platforms and facilitate the interview process. In order to provide the department a high performance and useful system, the project makes use of the technology like real-time video communication and adopts data analysis from previous students. By all these, we hope to make it more efficient and useful for professors or department to admit the best Mphil and Phd students.

Appendix

Webrtc – Webrtc offers the advantage of using videos without the need of plugins like flash. However, that would also mean that it requires html 5 and would not be supported in older browser. This is a thing to note during development of the project.

References

Thomas, K. (2015, June 16). Getting Started with iOS: Objective-C vs. Swift. Retrieved September 22, 2015, from *http://www.codenewbie.org/blogs/getting-started-with-ios-objective-c-vs-swift*

Wall, Q. (2015, June 10). The Future Of Enterprise App Development Is Swift. Retrieved September 22, 2015, from *http://techcrunch.com/2015/06/10/the-future-of-enterprise-app-development-is q-swift/*

Study for student performance prediction:
http://www.lon-capa.org/papers/v5-FIE-paper.pdf
http://www.researchgate.net/publication/242341193_DATA_MINING_APPROACH_FOR_PREDICTING_STUDENT_PERFORMANCE
http://arxiv.org/ftp/arxiv/papers/1203/1203.3832.pdf
http://arxiv.org/ftp/arxiv/papers/1201/1201.3418.pdf

WebRTC plug-in:
http://www.webrtc.org/

C_p criterion
https://en.wikipedia.org/wiki/Mallows%27s_Cp

Recommender System
https://en.wikipedia.org/wiki/Recommender_system

Logistic Regression
https://en.wikipedia.org/wiki/Logistic_regression