COMP4801 Final Year Project
Final Report

A Cloud-based Mobile App for Tutors and Parents
EasySchedule

Created by
CHAN Ka Lok 3035229502

with
LAM Siu Wai 3035xxxxxx
HO Hilary 3035xxxxxx

Supervised by
Dr. WU Chuan

Date of Submission: 4/15/2018
ABSTRACT

Making appointment between parent and tutor can be complicated when there are multiple parents and tutors. Such problem can be solved by an automatic scheduling system. This project solves the problem by developing an Android client application, implementing data storing through cloud services, and design a scheduling algorithm for class scheduling. The whole system, including the backend scheduling algorithm, cloud functions, and the front-end Android application have been implemented throughout the year. There is performance analysis after the scheduling algorithms are developed. Also stress test of our system have been performed. The next step of the project is to refine the UI and UX of the client application and research for scheduling classes using artificial intelligence approach, such as using genetic algorithms. This final report presents the background, the approach, the result, the problems encountered, and the future works of the project.
ACKNOWLEDGEMENT

We would like to express our appreciation to Dr. Wu for her advice and guideline.
TABLE OF CONTENTS

Abstract ................................................................................................................................................. 1

Acknowledgement .................................................................................................................................. 2

List of Tables .......................................................................................................................................... 5

Abbreviations ......................................................................................................................................... 6

1. Introduction ......................................................................................................................................... 7

2. Related Works Review .................................................................................................................... 9

3. Methodology ....................................................................................................................................... 10
   3.1 System Architecture ..................................................................................................................... 10
   3.2 Android Application ..................................................................................................................... 11
   3.3 Cloud backend .............................................................................................................................. 11
   3.4 Scheduling Algorithm .................................................................................................................. 11
   3.5 Project Management .................................................................................................................... 12

4. Final Result ......................................................................................................................................... 13
   4.1 Android Application ..................................................................................................................... 14
       4.1.1 Specify available time slot for tutors and parents ............................................................... 14
       4.1.2 View personal class schedule for tutors and parents ....................................................... 17
       4.1.3 Receive class notifications for tutors and parents ............................................................. 18
       4.1.4 Send class announcement for tutors ............................................................................... 20
       4.1.5 Receive class announcement for parents ......................................................................... 21
       4.1.6 Class enrollment ............................................................................................................... 22
       4.1.7 Create class ....................................................................................................................... 23
       4.1.8 Register ............................................................................................................................. 24
       4.1.9 Log-in ................................................................................................................................ 25
   4.2 Firebase Data Modeling and Cloud Function .............................................................................. 26
   4.3 Scheduling Algorithm .................................................................................................................. 32
4.4 Performance Analysis

5. Difficulties
   5.1 Algorithm design
   5.2 Instant messengers integration

6. Conclusion
   6.1 Future Work

7. References
LIST OF TABLES

1. Other related applications and their key features 9
ABBREVIATIONS

Here are some abbreviations used in report.

AVD: Android Virtual Device

CPU: Central Processing

GCP: Google Cloud Platform

IDE: Integrated Development Environment

NDK: Native Development Kit

OS: Operating System

SDK: Software Development Kit

VCS: Version Control System
1. **INTRODUCTION**

Hiring private tutor for playgroups or skills teaching are becoming popular among young children’s parents. To make an appointment, both parties, the parent and the tutor, must arrange it through face-to-face communication, phone calls, or instant messengers. They must suggest some available dates and time to each other and determine the best date and time for both parties. Considering more children, more parents, and more tutors, making an appointment needs more time and effort. Managing multiple appointments for parent and tutors is also became undoubtedly complicated. Therefore, we propose a cloud-based application called EasySchedule to assist both parties in arranging and managing multiple appointments with ease.

To arrange and manage multiple appointments between multiple parties manually through communication and schedule checking is time consuming. There exist some event scheduling or parent-to-tutor matching application in the market. They all can help with the appointment scheduling and managing process in some degree. Nonetheless, they lack an automatic scheduling feature for tutors and parents, which is a crucial feature in our project.

A client-server system and a scheduling algorithm are required for solving the problem. The client is a mobile application connected with cloud platform. It provides users - parents and tutors, the following features: automatic class scheduling, class calendar, class notification, class announcement, and private messaging.

This project targets the Android platform, as all our team members have the hardware and software for developing Android application. Besides, the administrator control panel for managing the users and classes details will not be developed in this project. The focus of this project is on the client application rather than designing the optimal algorithm for the scheduling problem.

The backend is implemented on cloud technology. In which the database is setup on Firebase, and the scheduling algorithm is implement on the Google Cloud Platform. Modeling the data in a non-relational structure is important in this project as to utilize the benefits of the real-time NOSQL database provided by Firebase. Also, the scheduling
algorithm must be efficient and correct, such that it can generate schedules that fit for most users within a reasonable amount of time.

The following sections of this report will cover the methodology of this project, the work that we have done currently, the problems we encountered, and our following actions.
2. RELATED WORKS REVIEW

There is some existing solution to the problem in Google Play Store, which contain the keywords “private tutor”, “tutoring”, “tutor scheduling”, etc. There are three most related applications to this project. They are itzTutor Partner, Synkers, and TuLi. Their focus is on matching parents and tutor. The first two application only provide a one-to-one appointment function, that function allows user to make an appointment to the tutor, without optimizing the tutor’s free time and the tutee’s free time. For TuLi, it comes with a scheduling function, but their website is not accessible. We suspect their application is obsoleted.

<table>
<thead>
<tr>
<th>Name</th>
<th>Platform</th>
<th>Scheduling</th>
<th>Communication</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doodle</td>
<td>Web+app</td>
<td>Manually</td>
<td>N/A</td>
<td>Any activities</td>
</tr>
<tr>
<td>Calendly</td>
<td>Web</td>
<td>Manually</td>
<td>N/A</td>
<td>Meetings</td>
</tr>
<tr>
<td>Bloomz</td>
<td>Web+app</td>
<td>Manually</td>
<td>Message/feed</td>
<td>School</td>
</tr>
<tr>
<td>Remind</td>
<td>Web+app</td>
<td>N/A</td>
<td>Message</td>
<td>School</td>
</tr>
<tr>
<td>SignUp.com</td>
<td>Web</td>
<td>N/A</td>
<td>N/A</td>
<td>Any activities</td>
</tr>
</tbody>
</table>

Table 1 - Other related applications and their key features.

In addition, the applications shown in table 1 are some event scheduling or school-oriented communication application. Most of them have some kind of event enroll feature, which eventually requires the organizer or the participant to determine the date and time of the event.

In contrast, this project aims to provide an automatic scheduling service. The users only required to input their available time slots. Then the system will schedule the classes for parents and tutors.
3. METHODOLOGY

This project focuses on implementing the Android client of EasySchedule, the cloud backend, and the class scheduling algorithm. The cloud platform selected for the project is Google Cloud Platform and Firebase, as they provide easy to use API, scalable services, and stability.

The implementation of this project is divided into two parts: developing the client application and implementing the backend logic. The details for the system architecture, the android application, the cloud backend, and the scheduling algorithm will be discussed in the following sections:

3.1 SYSTEM ARCHITECTURE

The architecture of the system is designed as follows:

Figure 1- The overview of the 2-tier architecture system
The system is implemented with a 2-tier architecture design (Figure 1). The clients are the users of our Android application. The cloud backend consists of two components - Firebase and Google Cloud Platform (GCP). Firebase is used as the database of this system. Since Firebase does not support implementing complex algorithm on its Cloud Function feature, GCP is used for running the scheduling algorithm. By choosing this 2-tier architecture approach, our team members can focus more on the Android application development as there is much less setup needed for the backend.

3.2 ANDROID APPLICATION

The client application is a native Android application written in Java. Our team members all have Java programming and internship experience in the past. And more importantly, developing iOS application requires macOS, which is a huge limitation for our members. Therefore, the deliverable of this project is an Android application.

3.3 CLOUD BACKEND

The cloud backend is Firebase. And this project use GCP for running the scheduling algorithm. GCP and Firebase are hosted by Google. The server stability and reliability can be ensured. Firebase and GCP are well known cloud services for years, they have well documented APIs and support documents. By implementing the backend on these cloud services, the system can be developed, deployed, and maintained in lower costs.

3.4 SCHEDULING ALGORITHM

The scheduling algorithm is run on the GCP instead on Firebase. The reason behind this is that Firebase does not support running complex functions in its Cloud Function feature. Running the scheduling algorithm on GCP can ensure that there are enough resources for this task. The CPU cores, memory, and storage on GCP can be scale up after setup. This ensure scalability for the system. As a result, the current setup is most effective and efficient.

The scheduling algorithm utilized the bin-counting approach which will fits the class slot that indicated by most parents. This problem resembles of the 3-dimensional matching problem, which is a NP-hard problem. Hence, we are not able to have an optimal solution to solve such problem.
3.5 Project Management

We use Bitbucket as our online version control repository (Figure 2), which allows us to access, modify, and update the code more easily across team members. Also, we put the distributed tasks on Trello, which helps us keep track of the progress of the project.

![Figure 2 – Screenshot of the Bitbucket page.](image-url)
4. **Final Result**

The result of this project are as follows: scope identification, background research, project website, system design, use cases design, implementation of the application, implementation of the data structure, implementation of the scheduling algorithm, and performance analysis. And this section will be focusing on the following major result – the Android application, the data structure on Firebase, the scheduling algorithm, and the performance analysis.
4.1 **Android Application**

After the research on related-works, we have defined several use cases that will be included in our first release. We have implemented all of them, the details are as follows:

4.1.1 **Specify available time slot for tutors and parents**

Users can input their available time slot from the user by long clicking the empty space on the screen and input the date and time in the second screen(Figure 3).

![UI flow of adding a timeslot.](image-url)

*Figure 3– UI flow of adding a timeslot.*
Users also can add the timeslot from syncing with Google Calendar. First click the sync button on the top right corner in Schedule view (Figure 4).

Then after getting the required permission from users, a dialog will pop up asking for which timeslot should be added. As seen in the right figure, there is an event in user’s calendar on Wednesday from 10am to 11am. Therefore, the dialog will not include this timeslot for user to choose from (Figure 5).
After selecting the option “Wednesday from 08:00 to 10:00”, the timeslot will be indicated in the Schedule view (Figure 6).

![Schedule view with one available timeslot shown.](image)
4.1.2 View personal class schedule for tutors and parents

User can view a list of confirmed classes which have been scheduled automatically. By clicking the scheduled class in the Home view, the class schedule is listed under the “Class schedule” section (Figure 7).

![UI flow for viewing class schedule](image)

Figure 7 – UI flow for viewing class schedule.
4.1.3 Receive class notifications for tutors and parents

User will be notified for some special events, such as reminder for indicating timeslot, a class has been scheduled, and class reminder. The notification option can be turned on or turned off in Setting View. After a class has been created, the participants of the class will receive an email. Also, there will be a notification before a scheduled class is started (Figure 8).

Figure 8 – UI designs for the notifications.
If the user hasn’t indicated the available timeslot for class scheduling, a notification will be sent to the user as a reminder (Figure 9).

![Figure 9 – UI for indicate timeslot reminder.](image)

After a class has been scheduled, the system will notify the user about the class schedule(Figure 10).

![Figure 10 – UI for class schedule reminder.](image)
4.1.4 Send class announcement for tutors

Tutor can send class announcement too all class members which the parents are enrolled the class. The tutor first click the “message” button in Class Detail page. Then the user input the title and message to be sent. After clicking the “send” button on the top right corner, the message will be sent to all participants by push notification (Figure 11).

Figure 11 – UI flow for sending class announcement.
4.1.5 Receive class announcement for parents

Parents can view the class announcement from tutors and parents will receive a notification about the class announcement (Figure 12).

![Figure 12 – UI class announcement notification.](image)
4.1.6 Class enrollment

The parents can request for joining a class from the Search view. After clicking a class in the Search view, the parents can send a request to the tutor by click the “JOIN THIS CLASS” button (Figure 13).

Figure 13 – UI flow for class enrollment.
Then the tutor of that class will receive a push notification about it. The tutor can accept or reject the request in the Class Detail view (Figure 14).

![figure 14 - UI for accept/reject request.](image)

### 4.1.7 Create class

The tutor can create a new class by clicking the “add” floating action button on the Home view. Then input the class information and add the participants. The class will be listed on the Home view after (Figure 15).

![figure 15 - UI flow creating new class.](image)
4.1.8 Register

Any users need to register an account to use our service. A user need to indicate their role and input some of their personal particulars. Then they need to verify their email after clicking the register button (Figure 16).

![UI flow for registering a account.](image)

Figure 16 – UI flow for registering a account.
4.1.9 Log-in

User need to login to use our service after registration (Figure 17).

Figure 17 – UI for log-in screen.
4.2 Firebase Data Modeling and Cloud Function

With Firebase’s authentication feature, the user name, email, and password are stored with the built-in Firebase database without defining the database model. Except basic users’ information, the database also need to store the class information, users’ available time slots, announcement, and chat messages. To utilize the speed of Firebase, the structure of those data is flattened, i.e. de-normalized. The structure of our database is shown as follows in JSON format:

![Structure for Users data](Figure 18)

Figure 18 – Structure for Users data.
Figure 19 – Structure for UserSchedule data.
Figure 20 – Structure for Class data.
Figure 21 – Structure for ClassSchedule data.
Figure 22 – Structure for Category data.
Figure 23 – Structure for Notification data.
4.3 SCHEDULING ALGORITHM

The scheduling algorithm is implemented in JavaScript with NodeJS which will be hosted on Google Cloud Platform. After a class has been created, the scheduling deadline date will be saved to the cloud platform. The scheduling algorithm will then be triggered by a https URL endpoint from the cloud platform to perform class scheduling.

The scheduling algorithm design we adopted are the bin counting approach with heuristics. They are by greatest availability of the timeslot, sequentially, and least conflicts among time slots.

The performance analysis of the algorithm is in the following section.
4.4 PERFORMANCE ANALYSIS

After running the algorithms 20 times with different number of classes. The number of classes successfully scheduled from 3 algorithms with different heuristics are almost identical.

![Distribution of Numbers of Successful Classes](image1)

Figure 24 – Distribution of numbers of successful classes.

But the running time for the algorithm based on greatest availability heuristic are significantly faster.

![Running time analysis](image2)

Figure 25 – Running time analysis.

Therefore we will choose the one with greatest availability heuristic as out scheduling algorithm.
5. DIFFICULTIES

After the use case design, user interface design, and data modeling, the following aspect of the project require extra attention:

5.1 ALGORITHM DESIGN

The algorithms for solving a scheduling problem is complicated. If we choose to design an algorithm to solve our problem with the optimal result, this task itself will probably consume a year. But if we only implement a trivial state space search for searching a feasible timetable for all the parents and tutors, then the running time of the algorithm will be extremely slow and probably unable to complete.

And in this project we only implemented algorithms that based on trivial bin-counting technique, which are not able to reflect the truth performance of our algorithms when comparing them to other advanced algorithms.

Therefore, we are going to research about different types of solution for scheduling problem, for example using genetic algorithm, in order to come up with a good enough solution for the scheduling problem in this project.

5.2 INSTANT MESSENGERS INTEGRATION

In our project proposal, we planned to integrate with one of the instant messenger, WhatsApp or WeChat. Due to the lack of API, we decided to remove such feature. We will implement an in-app chat feature instead.
6. CONCLUSION

An automated system to schedule classes for parents and tutors allows them to plan their time more easily. To achieve this goal, our team have to design a system with a scheduling algorithm.

Six months after submitting the project plan, we have determined the system architecture, the basic features of the application, and basic user interface of the application. And finally implemented the whole system.

Although there are some setbacks, such as the algorithm is yet far from perfect, our team will continue to research more on this topic.

6.1 FUTURE WORK

We are going to refine every UI in the application to provide a better and more intuitive experience for users. Especially the input available timeslot procedures are considered as slightly complicated. Also, we need to implement the edit class function for tutor, in case they need to modify the class information.

As aforementioned, we will keep researching on different advanced solution on scheduling problem. Such that we can have a boarder understanding on how to solve such problem in a cost-effective and efficient way.
7. REFERENCES


