Course Schedule Planning:

Sabancı Üniversitesi

PURE
PROGRAM FOR UNDERGRADUATE RESEARCH

Student(s)
MERT ERTÖRER
GÜRHAN ELÇİÇEK
M. KEREM KAHRAMAN

Faculty Member(s)
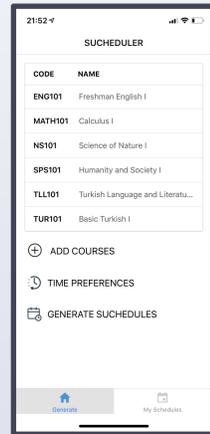ESRA ERDEM

ABSTRACT

Figure 1

Figure 2

This poster introduces mobile application which makes course schedule planning based on user's preferences about sections and time with React Native. The algorithm that creates schedule with user's preferences already constructed for web based personalized course schedule planning program *SUcheduler*, the main purpose of this project is enhanced the accessibility of *SUcheduler* by switching the program to a mobile platform. As a result *SUcheduler* being more universalize and support to concept of simplicity.

OBJECTIVES

Preparing a schedule is an important problem for students since there are lots of preferences, sections and consequences between classes students want to take. *SUcheduler* is a web based program which may be seen as complicated for students. This project is about designing a mobile application for *SUcheduler* with React Native to ease the usage of program for students. Our objective is helping the students for their program creation with the simplicity. As a result they would create their most satisfied program only with their mobile phones , tablets...
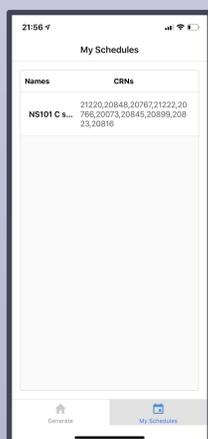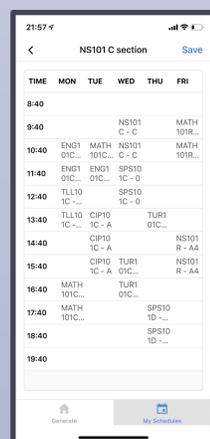
SOLUTION APPROACH

Figure 3

Figure 4

Firstly we did survey to students to understand what qualifications did students desire. Then we started with designing interface of the program to make it useful and easily understandable by hand. Then we examined the libraries and components that we could use and learned how to create components and implement them into specific page. We started to create our application by downloading React Native which is a cross-platform mobile application development platform using JavaScript code (https://facebook.github.io/react-native/). By using tab options we created two main screens which named Generate (seen in Figure 2) and My Schedules (seen in Figure 3) and navigate to sub screens from these screens such as preferences about scheduling. Since we have not worked on JavaScript before, we struggled to get used to the language and implementing the designs and ideas we had. Therefore , we still have some missing parts which we wanted to include.

DETAILS

We can easily portrait *SUcheduler* with 2 tab navigation screens, Generate and My Schedules and 2 sub screens, Add Courses and Time Preferences.
Firstly, **Generate** is the main page that will be shown after the loading screen ends. This screen stands for generating the schedule with selected preferences and courses. User can navigate to Add Courses page to select courses and Time Preferences to arrange possible free times with the related buttons on the page. Finally , Generate Schedules button navigate user to the list of possible schedules with arranged preferences (Figure 7). If there is no possible schedule with chosen classes because of time conflicts, program shows an alert that explain this conflict.

Time Preferences page (Figure 5) displays empty schedule to let users choose their free times. To do this we needed to implement table cells as toggle switches. Application will create optimal schedules with considering arranged freetimes but does not give a guarantee for %100 accurate free times that user set since the algorithm relies on a fixed course schedule, they may not be possible.
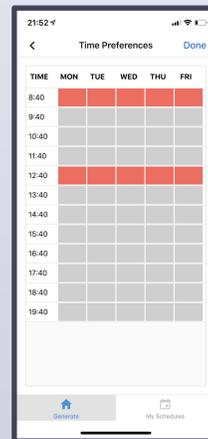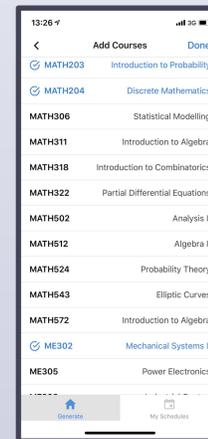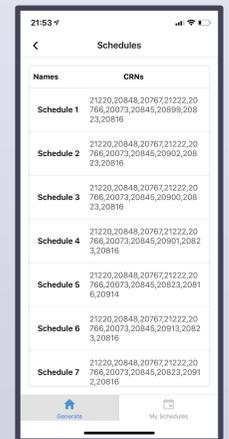

Figure 5

Figure 6

Figure 7

My Schedules (Figure 3) is the second tab navigation screen that users can see their saved schedules with CRN codes of selected classes in a table. And they can view the detailed schedule by day and time by clicking on the name of saved schedule.We implemented the table component of React native to create table of schedule names and CRN codes for "My Schedules" screen and assigned the rows as a button which navigates through its detailed version also we implemented similar table for detailed schedule by day and time.

Added Courses page (Figure 6) is basically made with select able list of available classes in current term. For implementing this property we used select able list library and arranged its several properties and appearance. User specifies the classes that s/he wanted to sign in this page and after clicking done button at right top corner, selected courses will be shown in the box above the buttons at Generate screen.

CONCLUSIONS

We couldn't manage to finish all the properties of the project as we wanted in 3 months because this project needs strong and deep knowledge for coding in JavaScript. React Native provides us semi-ready components which provides us to create template. We should improve our mobile coding skills and upgrade the template with advanced implementations.On the other hand we tried to build the simplest design for students because we believe that simplicity is helping the decision-making mechanism.Also we managed to create most of the pages we planned , and we thank **Saim Emre Şahiner** for his helps to enhance and beautify the UI and UX and also for publishing the application on his App Store and Google Play Store accounts, you can find it by searching *SUcheduler*. We will add some missing features to Generate such as adding **Section Preference** for selecting specific section of chosen class or relevant lab, recitation or discussion.
In the long term, *SUcheduler* will be a very good precursor for an intelligent scheduler for universities and maybe in another branch.

REFERENCES

Components and APIs. (2019) retrieved from https://facebook.github.io/react-native/docs/components-and-apis.html

Kahraman M.K., Erdem E. (2019) Personalized Course Schedule Planning Using Answer Set Programming. In: Alferes J., Johansson M. (eds) Practical Aspects of Declarative Languages. PADL 2019. Lecture Notes in Computer Science, vol 11372. Springer, Cham, p(37-45).

React Native.(2019) retrieved from https://facebook.github.io/react-native/

React Native Modules and Installation.(2019) retrieved from http://www.reactnative.com/

React Native Table Component.(2019) retrieved from https://www.npmjs.com/package/react-native-table-component

The fastest way to build an app.(2019) retrieved from https://expo.io/