Development of Mobile Application for Faculty Time out Information
Telkapalli.V.S Ram Kashyap, Kishore Kumar Kamarajugadda

Abstract: In the trend of mobile applications, the world is surging through many applications for various personal and professional purposes. In every domain including the cutting-edge technology such as Machine learning, IoT (Internet of Things), representing the data to the user in a proper and understanding manner is very important. This is where mobile applications come to use. Mobile applications can be used to resolve many communication issues especially when communication is between low level to high level and vice versa. This application is made to serve as one of the best ways of communication between faculty and students especially when the faculty is not available in the cabin and the student is willing to meet the faculty at the same time. The mobile application uses Dart Language with Flutter UI Software Development.

Keywords: Dart, Flutter, IoT (Internet of Things), Machine Learning, UI (User Interface).

I. INTRODUCTION

The evolution of mobile application is increasing rapidly in the 20th century. In any sector like bank, industrial, private and public sectors, the use of mobile application changed the phase of data representation completely and the ease of transparency in a sector. Not only in the representation of data, but also it bought the revolution in the process of work getting done especially in the bookings for railways and other means of transport. Showing the way of reduced number of levels in the conventional process in the sectors easy for both the users on both sides to get the required job done. From centuries, there has been a continuous change in the world of internet. During this change, many platforms such as websites were introduced. But occupation of smart phones changed the phase and mobile applications became necessary as they are efficient and as compatible as websites in the computer. The designed and developed mobile application is built for students and for faculty in the university where the faculty sets the reason, day and timings when they are leaving out of the cabin in the application and the students can check if the faculty is available in the cabin if they want to meet the concerned faculty.

In the university setup [1], if the student wants to discuss with the faculty members any student related issues or related with the course/subject, student is unaware of the fact the concerned faculty is in/out of his cabin for the discuss. If the students have information about the faculty availability in his/her cabin, it will save a lot of time for the student. This miscommunication between students and faculty in the university about the availability of the concerned faculty in their respective cabin is leading to some problems to both students and faculty especially at important times such as examinations, admissions. The necessity to solve this problem led to building a mobile application.

Generally, a mobile application [2] can be built using many languages such as android, Dart etc. The language used for building the proposed application is Dart and flutter SDK. Flutter is a user interface software development created and introduced by Google. It is an open source. In flutter, everything is considered as a widget which are divided into two types namely Stateless Widget and Stateful Widget. It uses Dart language which is very easy to adapt for any software developer as it is one of the best high-level language ever created.

A mobile application [3] demands a lot of space, fast, accurate and efficient data retrieval as the number of users increases. Hence, the database used here is Cloud Firestore which was created by Google.

II. REVIEW OF APPLICATION

The existing systems, Learning Management Systems (LMS) provide various facilities for the students such as representing the attendances, the number of courses registered by a student, respective faculty for the registered courses and the corresponding grades achieved by a student [4]. But it does not provide any facility of showing if the faculty is available in the cabin under different situations. Different situation includes the faculty having a sudden meeting, applying for an emergency leave due to any serious reason.

In order to overcome the described problem, the mobile application has been developed so that the students can know about the faculty time out information from the cabin with the reason to maintain transparency.

III. PROPOSED METHODOLOGY

The activity diagram (Fig.1) shows the architecture and the following process that a faculty member and a student will be undergoing in the mobile application [5]. As shown, both the faculty and student will be registered by giving the details and then they are navigated to login where they will be given proper and correct credentials. If the wrong credentials are given, they remain in the Login module.

On successful login, they are navigated to their respective home modules where faculty gives date, to and from timings and students can check their respective faculty timeout in their home module.

Revised Manuscript Received on February 2, 2020.
T.V.S Ram Kashyap, Department of ECE, Faculty of Science and Technology, ICFAI Foundation for Higher Education, Hyderabad, India.
Email: kashyap.ram17@gmail.com
K. Kishore Kumar, Department of ECE, Faculty of Science and Technology, ICFAI Foundation for Higher Education, Hyderabad, India.
Email: kkishore@ifheindia.org

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication
The mobile application is divided into two domains namely Student domain and Faculty domain [6]. As the names represent, the student domain is for enrolled students and the faculty domain is for faculty of the university.

A. Starting Module

This is the module where the application gets divided as discussed. This is where the user decides which domain he belongs to and continues to the further modules. The advantage by doing this is that one application will be enough to both students as well as faculty instead of different applications for both domains. Here, to change the default font, Google fonts are used. By downloading and adding the fonts in the pubsec.yaml file enables the developer to change the default font to the custom font. This brings a beautiful representation of the application. This comes under Stateless Widget. To differentiate between student domain and faculty domain, Hero Widget are used as shown in Fig. 2. The major advantage of Hero Widget on other button widgets like Raised Button is that Hero Widget gives a facility to add a logo to it and also acts as a button which helps to navigate to another domain. Here, two Hero Widgets are used as there are two domains. The Hero Widget comes under Stateful Widget. Additionally, an image of university is also added. This Widget comes under Stateless Widget.

B. Registration Module

In this module, the user registers by filling the required credentials. The data will be further used in the Home domain and in the Login domain. Here, TextField Widget is used as shown in Fig. 3.1. and Fig 3.2. TextField Widget gives the facility to have hint text which will be very useful for the user what to provide in each text field. Further, it also provides box decoration facility for rounded edges.
IV. RESULT ANALYSIS

Cloud Firebase which is fast and simple to understand as the data is stored in the form of the document fields in a document which ID can be auto generated in a collection for which the desired name can be given by the developer. Cloud Firestore accepts many data types. Here the data is stored in 3 collections which are

1. Faculty_info
2. Student_info
3. Faculty_timeout

Faculty_info consists of the details given by the faculty at the faculty registration module as shown in Fig. 4. 2.

Student_info consists of the details given by the student at the student registration module as shown in Fig. 4. 1.

Faculty_timeout consists of the details given by the faculty at the faculty home module as shown in Fig. 4. 3.

A. Login Module

In this module, the user is provided with two TextFormField Widgets which are email and password. Using the respective query of Cloud Firestore, the data given in the test field is compared with the data in database [8]. If the data is matched, the user will be directed to their respective Home modules as shown in Fig. 5. 1. and Fig. 5. 2.

Raised Button Widgets are also used which are Stateful Widgets. The Login criteria is given in the Table- I.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Criteria</th>
<th>System Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Email and Password fields are empty.</td>
<td>System should not accept</td>
</tr>
<tr>
<td>2</td>
<td>Only Email field is empty.</td>
<td>System should not accept</td>
</tr>
<tr>
<td>3</td>
<td>Only Password Field is empty.</td>
<td>System should not accept</td>
</tr>
<tr>
<td>4</td>
<td>Only Email input is incorrect.</td>
<td>System should not accept</td>
</tr>
<tr>
<td>5</td>
<td>Only Password input is incorrect.</td>
<td>System should not accept</td>
</tr>
<tr>
<td>6</td>
<td>Email and Password input are correct.</td>
<td>System should accept</td>
</tr>
</tbody>
</table>

Table- I: Login Criteria for Login Module
B. Faculty Home Module

This is the module where the faculty gives the date, from and to timings of their timeout. It is important to make the inputs for them to be easy because a text field for the reason input will be a conventional way as the faculty cannot type the description under emergency conditions. Hence, a dropdown button is added in the home module having Meeting, Leave, Break and Others as the list in the dropdown menu items as shown in Fig 6.1.

**Fig. 6. 1. Drop down for Faculty Domain**

For the input of Date, a raised button with the calendar logo, the set date and change text are given at the trailing of the button in a row as shown in Fig 6.2. The specialty of this raised button is that it takes the current date using the function `DateTime.now()` which gives the date and time by `DateTimePicker` package added in `pubsec.yaml` file. But, here only date is needed. Hence, it can be achieved in the following manner:

```dart
_date = '${date.year}-${date.month}-${date.day}';
```

“cancel” and “Done” buttons are also provided in case the user wants to cancel the timeout.

**Fig. 6.2. Date button for Faculty Domain**

Similarly, For the timing’s sections, Raised Button widget is used and `DateTime.now()` function is used to take the current time as shown in Fig. 6.2. As time is the only requirement here, it can be achieved as follow:

```dart
_time1 = '${time.hour}:${time.minute}:${time.second}';
```
Fig. 6. 3. From Time button for Faculty Domain
Same goes for time which takes “To”. Raised button with current time as shown in Fig. 6. 3. But the faculty needs to set the time when their time out is going to end. Done and cancel buttons are provided in case they want to change the data regarding the reason in the drop down or “from” timings in the faculty home module and come back to time when their time out is going to end. These features make it simple for the faculty to give the inputs in a less span of time and works efficient with the accurate data and it is very simple to understand about the widgets provided in the module without any explanation or guide giving to the users before usage.

An AppBar is provided which comes under the Stateless Widget. Stateless Widgets are the widgets which do not contain any state or any purpose like button widgets.

Fig. 6. 4. To Time button for Faculty Domain
After giving the date and timings, the following image shows the overall look of the home module previewing the data that has been entered by the faculty such that they can have a look on the data that has been chosen by them.

Fig. 6. 5. Preview
A Raised Button is provided which on clicking access the function which triggers the query that is responsible for storing the data in the Cloud Firestore as shown in Fig. 6. 4.

In order to give confirmation that the data is stored in the database successfully, a notification is sent to the faculty mobile as soon as the data is stored in the database as shown. For this feature, flutter_local_notification package has been added to the pubspec.yaml file. After adding packages, showNotification() which is an async function because the notification has to wait until the data is stored in the cloud Firestore successfully.

If the data is stored, flutterLocalNotificationsPlugin/show() gets activated showing the following result in the faculty mobile notification center. This notification ensures the faculty that the data has been added successfully to the database as shown in Fig. 6. 5.

Fig. 6. 6. Notification received after successful storage of data in Cloud Firestore.
On clicking on the notification, a dialog box which shows the absolute confirmation that the respective timeout is arranged.
When the notification is clicked, it triggers the onSelectedNotification() which is future and async function as it has to wait until the user clicks on the notification. Fig. 6. 6. is the following is the result of the dialog box for another confirmation. The dialog box pops up when the user clicks anywhere in the screen.

Fig. 6. 7. Dialog box when Notification is clicked.
In the AppBar, and additional widget is also added which is the side drawer. Here, UserAccountsDrawerrHeader() which consists of accountName, accountEmail and currentAccountId() is added to show the details of user in the side menu as shown. The background color also can be changed with the help of BoxDecoration() function. Here, the list tiles “Home”, “History” is provided as shown in Fig. 6. 7.
Home represents the module where the faculty gives the date and time inputs and history is the module which shows the number of timeouts that has been taken by the concerned faculty which includes the reason as the title, Date as the subtitle and timings as the trailing.

Fig. 6. 8. Side Bar for Faculty Domain

C. History Module
For History module, Future Builder is used. There are two types of build functions which are stream builder and future builder. These build functions are used to make a function wait until another function completes its task. Here, retrieving the data is the task. Hence, Future Builder is used to make sure that the list tile function has to wait until the query which is responsible for retrieving the data is executed first. The main point is that the data has to be retrieved with respect to the specific faculty that is, the faculty user. To retrieve such kind of data, the following query is used:

`await Firestore. Instance. collection("faculty_timeout"). where ("Username", isEqualTo: username). getDocuments() ;`

Here, on basis of the username, the data is being received in the form of snapshots and the data can be represented by using these snapshots as shown in Fig. 6. 8.
On clicking the “home” list tile in the side menu, the user is navigated back to the main Faculty Home Module. This History module is so useful for the faculty as they can have a track of the timeouts that they are taking and this maintains transparency.

Fig. 6. 9. History module for Faculty Domain

D. Student Home Module
In Student Home Module, the Raised buttons and a drop down which shows the usernames of only registered faculty members in the dropdown menu items are provided for the ease of choosing and avoiding any kind of errors in the spellings of faculty names [9].
An AppBar is also given to the student home module to give the confirmation to the student that he is in the right domain to check if the concerned faculty is available or not. The drown button shows the list of registered usernames only which takes the corresponding data from the collection “faculty info” from Cloud Firestore.

Fig. 6. 10. Home module for Student Domain
A confirm button is given and on clicking it, it triggers the database query which retrieves the data from the collection faculty_timeout from cloud Firestore.

If no data is present which matches the data with respect to the given date as input by the student, a dialog box is shown by the faculty_timeout from cloud Firestore.

If no data is present which matches the data with respect to the given date, a Dialog box is shown saying that faculty is still in the cabin.

Else, it shows a Dialog box showing the date and timing of the faculty with the reason saying faculty time out.

ACKNOWLEDGEMENT

This paper was initiated by Prof. K. Kishore Kumar, Chairperson - Academic Instruction, Faculty of Science & Technology and would not have been completed without his valuable guidance for the completion of the mobile application. Author would like to extend his gratitude towards Dr. M. Srinivasa Reddy, the Director- Faculty of Science and Technology for providing a vision that enables the students to explore in every technical aspect in the real time environment.

REFERENCES


AUTHORS PROFILE

Telkapalli V.S. Ram Kashyap, graduated from ICFAI Foundation of Higher Education with a Bachelor of Technology degree in the field of Electronics and Communication Engineering.

Kamarajugadda Kishore Kumar is associated with Department of ECE, Faculty of Science and Technology, ICFAI Foundation of Higher Education, Hyderabad, India.