Proceedings of the Second International Conference on Smart Electronics and Communication (ICOSEC).
IEEE Xplore Part Number: CFP21V90-ART; ISBN: 978-1-6654-3368-6

2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) | 978-1-6654-3368-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOSEC51865.2021.9591786

MeetUp: An Appointment Booking System using Flutter and Django Framework

MeetUp: Meetings Made Easy

Prof. Pradnya Bhangale
Department of Computer Engineering
KJSCE, Vidyavihar
Mumbai, India
p.bhangale@somaiya.edu

Bhavik Bhatt
Department of Information Technology
KJSCE, Vidyavihar
Mumbai, India
bhavik.bhatt@somaiya.edu

Muskaan Nandu
Department of Information Technology
KJSCE, Vidyavihar
Mumbai, India
muskaan.n@somaiya.edu

Piyush Chavda
Department of Information Technology
KJSCE, Vidyavihar
Mumbai, India
piyush.chavda@somaiya.edu

***Abstract*—A ubiquitous problem faced by students and teachers of an educational institute is to arrange for meetings in an effortless manner and at a time which suits both. Teachers usually have a fixed time table for their regular classes, but they often get busy with impromptu meetings, extra classes, official work, etc. Since these situations are frequent and unpredictable, students willing to meet the teachers outside the regular class time often find it difficult to do so. To reduce the hassle of waiting for teachers and searching for them around the campus, this paper proposes a teacher-student appointment system named 'MeetUp'. This application allows a teacher member to upload their timetable from which free slots of the teacher will be extracted and shown against the respective teacher on the student portal, which will help students request for an appointment time that is viable for both the members. This system not only benefits the students but also helps the teachers manage their time as students would only show up with prior appointments and the teachers have the authority to accept an appointment request at their convenience. Meetup is built for native android users using Dart language based Flutter framework and python based Django Framework.**

Keywords— Appointment System, Flutter, Dart, Python, Django, Camelot, Mobile Application, Android Application, API

I. Introduction

Teacher members of an educational institution have a lot of responsibilities to manage and thus have a weekly timetable to manage these responsibilities which consist of different activities and timings on different days of the week. The above scenario manifests into a problem when students are unaware of the teacher's free time and thus struggle to set up a meeting in an effortless manner. A student may have different kinds of doubts and queries which he/she wants to clear with the respective teacher, however, due to the busy schedule of teacher members, a student has to roam about in the campus looking for the respective teacher. Even after locating the teacher, it might be a case where the teacher is busy with another task and cannot address the student. For example, we once wanted to set up a meeting with our Professor to propose an idea for a project but we weren't aware of the teacher's schedule and so we visited her office 2-3 times in a day to find her without any luck. When we found her she was busy with some other work and thus couldn't meet us. The above mentioned struggles are faced by plenty others, which results in a disoriented meeting system between the teacher and the student.

It is important to address the problem discussed above because in the current scenario, time of faculties and students is wasted in setting up a meeting which otherwise could be used in doing productive activities. Another problem is a lack of logs and records of the meetings that do take place. Even if students and teacher members would meet, there would not be a log of that meeting which might be of potential data interest to the institute.

The problems discussed above have always existed which were solved by quick fixes and temporary solutions, however, the current pandemic situation of COVID-19 worsens the problem. This is because, currently one cannot meet the teacher in person which leaves emails or phone calls as the only means of communication. Teachers might receive dozens of emails in a day and thus they wouldn't be able to answer all of them, making the process slower. On the other hand, some teachers are not comfortable sharing their private phone numbers with students for personal reasons. Hence, it becomes imperative to have a system that is time saving, fixes the above issues, is easily manageable and one which keeps history of the meetings in a manner that suits all.

To manage these problems we faced as students, we came up with an idea to build an application - “MeetUp” which aims to solve these traditional problems where users of the application have to first sign up into their role, student or teacher. For a teacher role, the teacher has to upload his/her institutional details along with their Timetable in PDF format. This pdf is sent to a server for processing and after processing the servers returns the free slots to the application client. These free slots are stored in a database in the respective teacher’s document.

After registration, the student can view all faculties of the educational institute and search for the teacher he/she wishes to book an appointment for. On the appointment page, the student can select his preferred date and day, after which the free slots of the selected teacher for that day are displayed to the student. The student then simply has to select a time from the displayed free time slots and mention the reason for appointment. The appointment details are sent to the respective teacher on MeetUp application and email upon which the teacher can accept the appointment request or reject it by mentioning the reason of rejection. The status of the appointment is shown to both the teacher and student at all times. This way all the previous appointments can be tracked, students can set an appointment with the teacher without struggling to find them and the teacher can manage the appointment requests from the students without having to share their private phone number.

II. LITERATURE REVIEW

In the system ‘Book The Room’[1] authors have developed an android application using flutter to make the process of booking conference rooms easier. The booking process involves 2 individuals, the user who books a conference room and an admin who grants permission to the user for the respective booking. This application thereby reduces clashes in the process of booking a conference room..

Another system called ‘Mwa3edk’ was developed by the authors[2] with the aim of adding new concepts in the process of booking appointments with doctors in hospitals and medical clinics by transferring this process into the online world technology.

"Bookazor" [8] is an online web-based application which is used for booking and scheduling appointments for parlor, hospitals and architects, etc. Technologies used in this application are CSS, HTML, JavaScript and Firebase. The system also uses NodeJS for storing the number of requests. The motive of this study was to identify the benefits and barriers faced while implementing an online appointment booking system.

The paper [9] highlights the development process of a web based Online Appointment Booking System which allows students to book and schedule an appointment with their advisor during their available time. It ‘s main purpose is to allow students to meet the lecturer for the purpose of working on a project with them. The lecturer registers their project description on the application and interested students can schedule an appointment with them.

The paper “e-Vaccine: An Immunization App” [10] introduces a mobile application that facilitates the vaccination process and helps parents and caregivers keep better track of children's vaccination schedules and enables them to make vaccination appointments. The result of developing the application demonstrated that the application was working as per expectations and that the users of this application would prefer using such an app if it were to be integrated into the healthcare system by the government.

“Medicus: A Doctor Appointment Booking System” [7] is also an appointment booking system in the healthcare domain. It is an on-the-go appointment booking system which helps patients schedule an appointment with their preferred doctor. The app was successful in eliminating long queues that the patients stand in and instead provides a way for them to book appointments of their choice in a convenient manner.

The paper [5] proposes an idea of how a web based appointment booking systemcan be used to let both teachers and students be acquainted with the time of appointment. It allows students to send a message to a teacher along with the appointment time, which is emailed to the teacher. The teacher must then reply to the email and the student will be notified.

The paper [6] talks about how online booking systems are useful because of their flexibility in planning and time efficiency. The paper aimed at assessing the online booking systems in Iranian hospitals and concluded that an enhanced quality of online booking systems in hospitals and clinics would help increase their efficiency and would be a good solution for the physicians and patients in setting clinical appointments.

The above-mentioned systems are diverse but have been developed for the fundamental purpose of facilitating and easing the process of booking appointments between two groups. However, every system has its own way of functioning, a few of which do not address all the problems faced by the users. MeetUp solves these problems and includes features that are not available in any of the previous appointment booking systems.

As discussed earlier, one of the problems faced by teachers is that they tend to receive dozens of emails everyday and thus might miss on responding to an email from a student requesting to meet them. MeetUp solves this problem by reducing the load on teacher's inbox and providing a different platform, customised for appointment requests. Secondly, certain available online appointment systems require the teachers to fill in their free slots manually. However, MeetUp has automated this process by extracting the free slots of a teacher from a pdf document of their timetable.

Apart from the features described above, MeetUp allows students to send an appointment request with a short message describing the purpose of the meet. Teachers may accept or reject a request based on this purpose and may also send a message to the students describing the reason for rejection, if applicable. Hence, MeetUp is a distinct concept that provides customised features for campus use, provided by no other existing system.

The MeetUp app is developed with the intention of smoothing the process of teacher student interaction where a scanned copy of the teacher's timetable is uploaded and the free slots of the teacher are extracted from that timetable. To achieve this we had to look for ways to scan a PDF document which include timetables in tabular format and then extract the Day and Time of the free slots. There are a variety of tools available online to extract tables from a PDF file. The two approaches for the same were proposed in the paper by Aditya Kekare, Abhishek Jachak, Atharva Gosavi and Prof. P. S. Hanwate, published in IRJET [3] :

1. Open source tools - Tabula, Camelot and PyPDF2.
2. Deep learning techniques - Using TableNet.

Since the requirement of the MeetUp application was customized for our college use, we achieved the best results by using the Camelot Package. Camelot is a package available in python for extracting tables. It takes a PDF file as an input and returns a DataFrame which can be later converted into CSV, Excel or JSON format. To make use of this python code in the flutter application 2 approaches are possible:

1. Using Starflut Plugin [4]
2. Using Django as a backend Server

Since the Starflut package is a new plugin, still under development phase, we opted to use django as a backend server.

III. Method

We have developed this application on Flutter which is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac and Windows. Flutter supports fast code development with increased time-to-market speed along with great potential of UI customization.Flutter applications are written in the Dart programming language, which is a simple to understand language. Dart is a garbage-collected, object-oriented, class-based language with C-style syntax and has an optimized approach for building user interfaces. Developers are able to iteratively make modifications to the source code and using the hot reload feature can view the result/modifications in the running app. Flutter offers other benefits such as:

1. Developers write just one codebase for your 2 apps – covering both Android and iOS platforms. Flutter doesn't depend on the platform, because it has its own widgets and designs
2. The same app UI can run on older devices without additional costs.Flutter runs on Android Jelly Bean or newer, as well as iOS 8 or newer.
3. Less testing: Being one codebase, the developers write automatic tests only once and hence the quality assurance process can be faster

Flutter has been used in this application to make User-Interface pages known as "Widgets" in flutter. Each widget is a page that is either displayed as an app screen or used for performing other functionalities.

Django, which is a Python-based free and open-source web framework has also been used in the implementation of this application. Authors chose Django primarily for the following reasons:

1. It is built on python which is simple to learn and use
2. Wide range of libraries available
3. Reliable and official documentation along with tutorials are available for django
4. As Django is an open-source framework and available for free, it is supported by active volunteers who constantly provide updates and resources on djangoproject.com and on Github
5. It offers Customizable Framework

Flutter is used to create the user interfaces of this application. The application uses the Google Firebase Firestore as the database to store and retrieve information.

Django has been used in this application as a server that takes a TimeTable PDF as input and returns the extracted free slots from the PDF as a json-object. The returned json-object is then converted back to the desired data type in flutter for further processing of the appointment request. This is essentially a client-server architecture wherein the Flutter app is a client requesting for service from the Django server and the Django server performs the designed functions and returns the end result back to the client.

IV. Implementation

MeetUp is an easy to use android application which allows registered students to book appointments with faculties who have registered and uploaded their timetable to the application. When a user downloads the MeetUp application, the user is first brought to a page to select their Role in the institution which is namely a student or a teacher which is shown in Fig I. The workflow of the application is different for both Student and Teacher roles. The architecture of the application is shown in Figure V below.

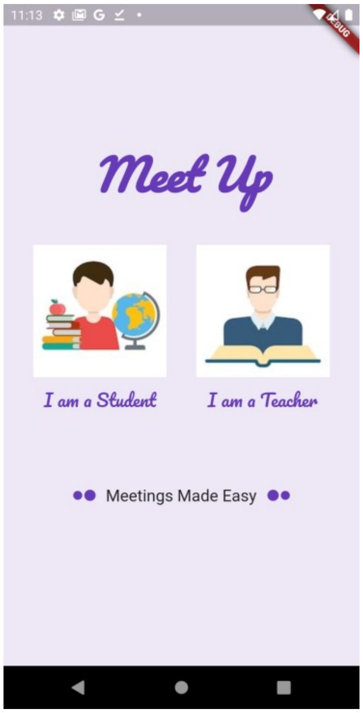


Fig I. Role Title of the User

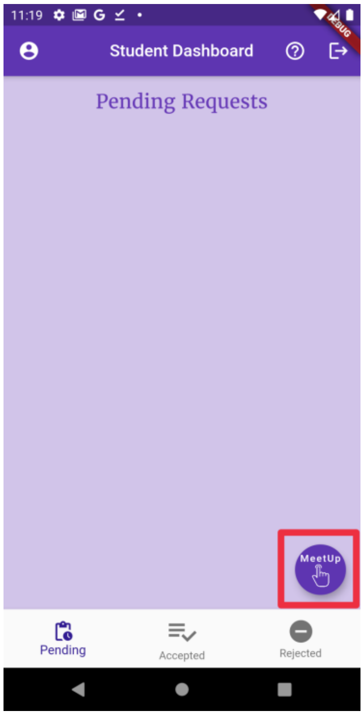
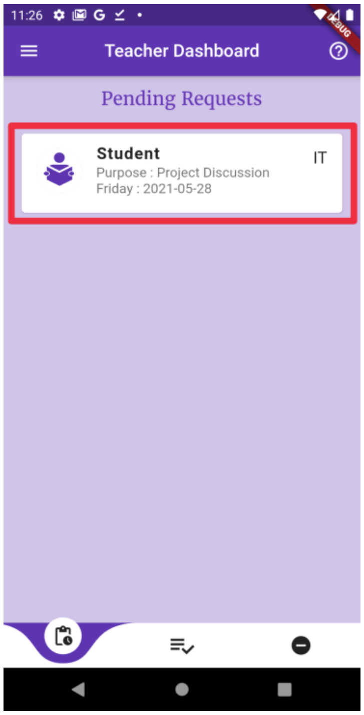


Fig II. Student Dashboard



Student
Purpose : Project Discussion
Friday : 2021-05-28
IT

Fig III. Teacher Dashboard

In the first scenario, the user being a student, he/she is taken to the student login page. If the Student is a new user they are required to register to the MeetUp application using their institutional Email ID and by filling in personal details. The details entered by the student are validated using regular expressions. On successful registration, the student is sent a verification link on their email ID. Next, using Firebase's "isEmailVerified" functionality, on every login attempt, it is checked if the student's email address is verified or not. On 3 unsuccessful login attempts due to missing verification of user account, the registration of the user is nullified and deleted from the database.Once the email is verified, the student can successfully login to the application and is taken to the Student dashboard from where the student can access his/her profile and make new appointments as shown in Fig II. To make an appointment, the student must select the teacher with whom they want to make an appointment from the displayed teacher list. The student user is also provided with a search option where they can search for a particular teacher by her name or employee code. We check if the search text is present in the list of registered teachers using the contains() function provided by Dart, and only those values are added to the dynamic list which return true value for the search condition.

Next the student must select a date and a time slot from the free time slots of the teachers displayed with respect to a particular teacher which are obtained when the teacher uploads their time table. The student can select a date starting from the next day to the upcoming 5 days. Then the student must describe the purpose of the meeting concisely and click confirm. On clicking confirm, the request is sent to the teacher and the teachers are notified via an email. If the student wishes to cancel an appointment, the student can click on the particular request and cancel the appointment by clicking on the cancel button (Only future requests in the calendar can be cancelled.)

For the second scenario,users being a teacher member are taken to the teacher login page, where the teacher can login using their Email ID and Password. If the teacher is a new user, they are required to register to the MeetUp application using their institutional email id and by filling in institutional details. Similar to a student's verification, the teacher is verified against the inputted details and email id. Once the email is verified the teacher is taken to the Teacher Dashboard from where the teacher can access his/her profile, access the requests from students as shown in Fig III

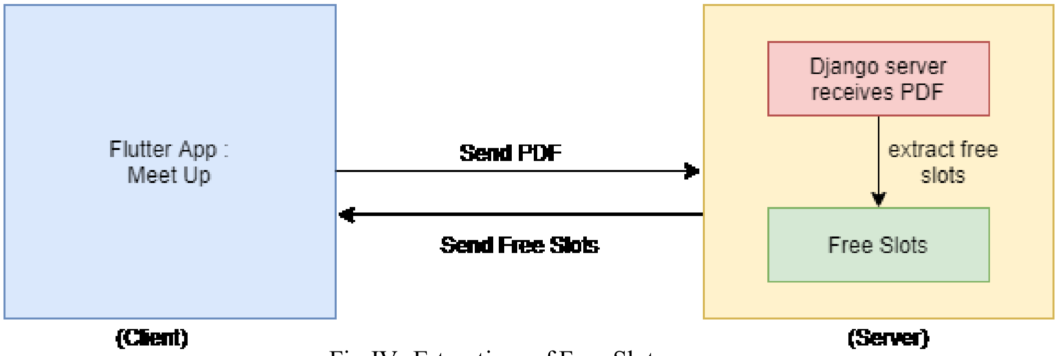


Fig IV. Extraction of Free Slots

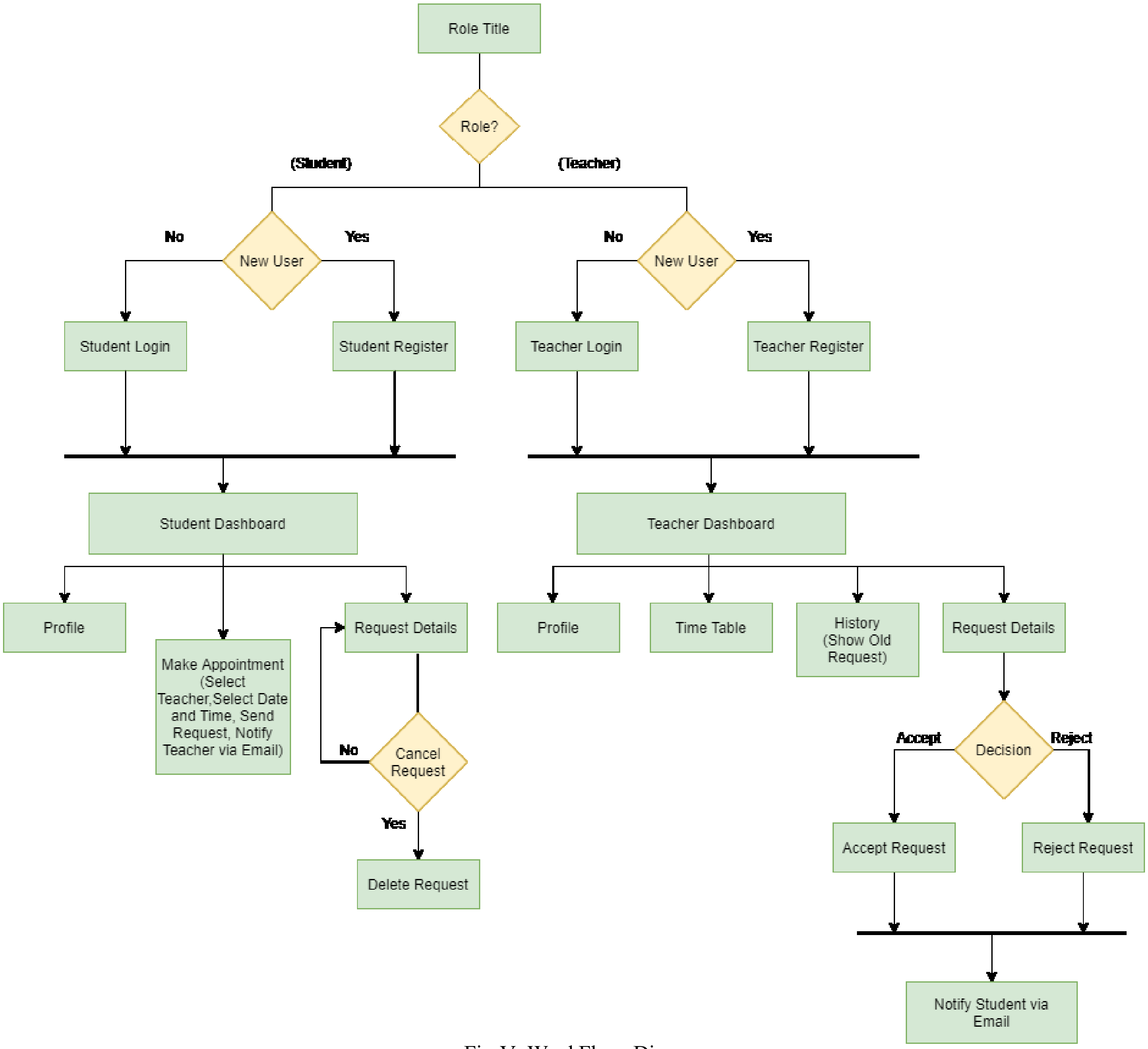


Fig V. WorkFlow Diagram

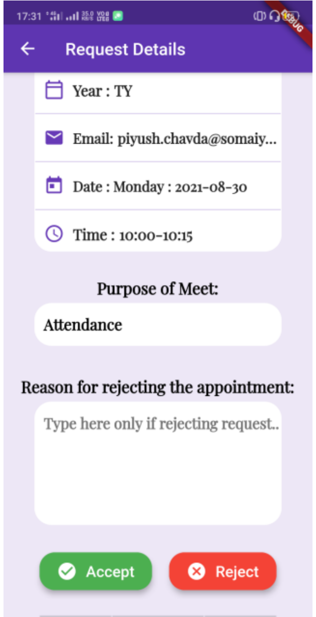
and upload his/her time table so as to make time slots available for students to book an appointment, it is mandatory for the teacher to upload their time table.

The teacher uploads his/her timetable in PDF format on the flutter app which is then passed onto the Django Server as a service call as explained in the Fig IV above. The Django server stores the PDF on a local file system which is then accessed by pointing to the path of the stored file. The file is then processed upon by using Camelot, a python library, to extract tables from PDF files. Using the Camelot library, teacher's free slots are extracted and stored in a dictionary. The dictionary is in a key-value pair where the key is day of the week and values are time slots for which the teacher is free for that particular week day. This dictionary is then returned to the flutter client as a json-object. This json-object is converted back to a dictionary in the flutter application and is stored in the database along with the teacher's other information and all this is used for displaying teacher data on the student's side. Once a teacher receives an appointment request, he/she is notified via an email. The teacher dashboard displays requests for the upcoming week, hence older requests can be accessed from the History section. The teacher can click on any request from the dashboard and check the details of the student. The teacher may then accept or reject the request based on the purpose of the meeting. In case of rejection, the teacher must specify the reason for denial. Also, if the requested appointment date by student has passed then the teacher cannot respond to the request. Once the teacher responds to the request, the student is notified via email. The already booked slots are not available to other students who wish to take appointments for the same date and time. The booked slots of a particular day are freed on the next day and made available for students seeking appointments in the coming week. For example, the booked slots on Wednesday are reset and freed on Thursday. This is achieved by using background tasks with the help of the "Workmanager" library in flutter. First, the current day is retrieved and next the day prior to the current day is reset. In this way the decision management of the time slots is handled and ensures there are no clashes.

V. RESULTS

In this system, we used Dart language and Flutter development kit to make the appointment application which smoothens teacher and student interaction in educational institutes. After completion of the development and testing phase, the application was found to be working perfectly as per expectations. The whole process of developing the application including planning and testing took around 90 days. Classmates were invited to take part in the testing phase of the application and they expressed their liking for the application. They found the application to be useful and how the appointment system would experience an overhaul and give rise to a much hassle free system.

The users said that they would personally use the application if it were to launch for institutional use. The teachers liked the application as all the information regarding meeting appointments was present at a single place and made it easier for them to track appointments made by students. The teachers could get an overview about the student and the purpose of the meeting before accepting or rejecting a request as shown in Fig VI. The requests are segregated based on their status, i.e accepted, rejected and pending requests which are displayed separately making it is easier for the teacher to keep follow up on the appointments. A history of all the requests is maintained separately as well on the teacher's account.
The app made it easier for students to locate and meet a teacher . A student can search for a teacher from a list of faculties registered on the application. Also, they can search for a teacher based on both their name and employee ID. The student can then view all the details of the teacher including their office location in the campus thus making it easier to find the teacher. The student can then select a date after which all free slots for the selected day will be shown to him/her. The student can select a time from the available slots and state the purpose of the meet. After sending the request, a student can track the status of their appointment as shown in Fig VII. In case the request is rejected, the student will be shown the reason for rejection in the request details page.



Request Details

Year : TY

Email: piyush.chavda@somaiy...

Date : Monday : 2021-08-30

Time : 10:00-10:15

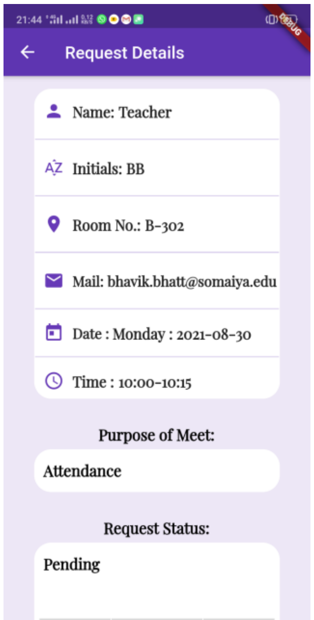
Purpose of Meet:

Attendance

Reason for rejecting the appointment:

Type here only if rejecting request..

Fig VI. Teacher's View of a request details



Request Details

Name: Teacher

Initials: BB

Room No.: B-302

Mail: bhavik.bhatt@somaiya.edu

Date : Monday : 2021-08-30

Time : 10:00-10:15

Purpose of Meet:

Attendance

Request Status:

Pending

Fig VII. Student's View of a request details

In MeetUp we built five collections for processing the appointment requests using the Firebase database. The collections and their document fields are shown in Table I.

Table I. Database Information

Collection Name	Faculty information	Student information	Pending requests	Accepted requests	Rejected requests
Document Fields	1. Time table 2. Time table status 4. Email 5. Employee code 6. Name 7. Name initials 8. Role (faculty or student) 9. Room number 10. Teacher id 11. Display picture URL	1. Branch 2. Year 3. Email 4. Name 5. Password 6. Role (student or faculty) 7. Roll no. 8. Student ID 9. Display picture URL	1. Date 2. Time 3. Purpose 4.Purpose Details 5.Status (Pending) 6. Request ID 7.Employee code 8. Teacher ID 9. Teacher Name 10. Teacher Initials 11. Teacher Mail 12. Teacher Room 13. Teacher Display URL 14. Student ID 15. Student Mail 16. Student Name 17. Student Roll Number 18. Student Branch 19. Student Year 20. Student Display URL	1. Date 2. Time 3. Purpose 4.Purpose Details 5.Status (Accepted) 6. Request ID 7.Employee code 8. Teacher ID 9. Teacher Name 10. Teacher Initials 11. Teacher Mail 12. Teacher Room 13. Teacher Display URL 14. Student ID 15. Student Mail 16. Student Name 17. Student Roll Number 18. Student Branch 19. Student Year 20. Student Display URL	1. Date 2. Time 3. Purpose 4.Purpose Details 5.Status (Rejected) 6. Request ID 7.Employee code 8. Teacher ID 9. Teacher Name 10. Teacher Initials 11. Teacher Mail 12. Teacher Room 13. Teacher Display URL 14. Student ID 15. Student Mail 16. Student Name 17. Student Roll Number 18. Student Branch 19. Student Year 20. Student Display URL

V. Conclusion

In this paper, we present an appointment system, customized for campus use : “MeetUp - Meetings Made Easy”. The application is built for native android users using the Flutter framework and python based Django Framework. Through this application we have tried to solve the most prevalent problem on campus by easing the process of booking an appointment with a teacher at a time suitable for both.

The application aims to save time and avoid miscommunication between the students and teacher members, as the app requires minimal work on users end. Setting up meeting appointments, tracking and managing them, keeping a log of all requests has been made convenient and straightforward for members of an educational institute.

VI. Future Scope

The application can be further scaled and improved by including the following:

1. iOS customisation of the application
2. Push Notifications for new requests and sent request status updates
3. Web application implementation for the same

VII. References

[1] Praveen, Anagha, et al. "Conference Room Booking Application using Flutter." 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2020.

[2] Odeh, Ayman, Raghad Abdelhadi, and Hussien Odeh. "Medical patient appointments management using smart software systems in UAE." 2019 International Arab Conference on Information Technology (ACIT). IEEE, 2019.

[3] Kekare, Aditya, et al. "Techniques for Detecting and Extracting Tabular Data from PDFs and Scanned Documents: A Survey." Tabula 7.01 (2020).

[4] Arb, Ghusoon Idan, and Kadhum Al-Majdi. "A Freights Status Management System Based on Dart and Flutter Programming Language." Journal of Physics: Conference Series. Vol. 1530. No. 1. IOP Publishing, 2020.

[5] Bello, Ridwan O., et al. "Student-teacher online booking appointment system in academic institutions." (2016)

[6] Samadbeik, Mahnaz, et al. "Assessing the online outpatient booking system." Shiraz E-Medical Journal 19.4 (2018).

[7] Khalid, M., et al. "Medicus: A Doctor Appointment Booking System." International Journal of Computing and Technology 5.4 (2018): 48-52.

[8] Akshay, V., et al. "BOOKAZOR-an Online Appointment Booking System." 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, 2019.

[9] Qaffas, Alaa, and Trevor Barker. "Online Appointment Management System." Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011

[10] Shirin, et al. "e-Vaccine: An Immunization App." 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM). IEEE, 2021

[11] Google Firebase, ”Firebase Documentation”, Google Firebase, 2012 [online]. Available: https://firebase.google.com/docs [Accessed: Dec 20, 2020]

[12] Naveen Kumar, “How to run the python code in the Flutter app?”, Smazee, Oct 02 , 2020 [online]. Available: https://smazee.com/blog/how-to-run-the-python-code-in-the-flutter-app [Accessed: Feb 5, 2021]

[13] Google LLC, “Flutter packages”, Google LLC, 2017 [online]. Available:https://pub.dev/packages [Accessed: Dec. 1, 2020]

[14] Google LLC, “Dart Documentation”, Google LLC, 2011 [online]. Available: https://dart.dev/guides [Accessed: Dec. 1, 2020]

[15] Django Software Foundation, “Django Documentation”, Django Software Foundation, 2005 [online]. Available: https://docs.djangoproject.com/en/3.2/ [Accessed: Feb. 10, 2021]