

Final Report of Tutron App Project

SEG2105 Z - Introduction to Software Engineering

Summer 2023

School of Electrical Engineering and Computer Science University of Ottawa

Course Coordinator: Prof. Wassim El Ahmar

Group 2

Abhay Ariyappillil, 300322826

Pierre Georges, 300283706

Nalan Kurnaz, 300245521

Sum Yan Wan, 300316371

Submission Date: 2023-07-16

Short Introduction

Tutron is a Ottawa-based tutoring application that helps students find local tutors for personalised support in their studies. It is a platform where students can register, search for tutors based on their needs, view detailed tutor profiles and schedule lessons. Administrators handle issues and maintain a secure learning environment, while tutors can register, manage teaching topics, handle purchase requests and monitor their profile ratings. By focusing on one-on-one learning methods, Tutron aims to enhance students' performance, and leading them to have a successful educational journey.

Updated UML Class Diagram

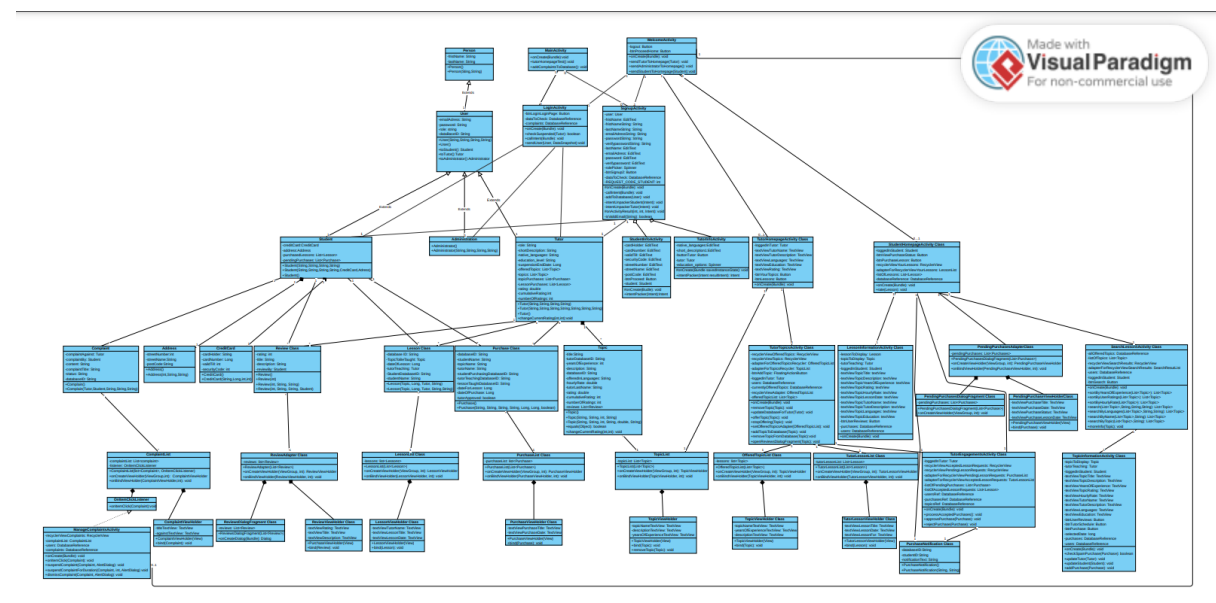


Figure 1: UML Diagram of Tutron Application

More detailed version of UML Diagram can be found in the Github Repository.

Tables specifying the contributions of team members for each deliverable

Feature or Task	Team Member
The team created in Github contains all members of the group	Abhay Ariyappillil
UML Class diagram of your domain model	Pierre Georges, Abhay Ariyappillil
The APK is submitted	Nalan Kurnaz
A user can create a Student or Tutor account	Abhay Ariyappillil
The Administrator, Tutor, or Student user	Abhay Ariyappillil

can see the “welcome screen” after successful authentication. The welcome screen specifies the user role.	
The user can log off.	Nalan Kurnaz
All fields are validated. There are appropriate error messages for incorrect inputs. (-1 for each field in which the user input is not validated)	Sum Yan Wan
Optional – The group uses a DB (e.g., Firebase, SQLITE, or another similar technology)	Abhay Ariyappillil

Table 1: Distribution of tasks specified in Deliverable 1

Feature or Task	Name of the Team Member
Updated UML Class diagram of your domain model	Pierre Georges, Nalan Kurnaz
The APK is submitted	Nalan Kurnaz
You implemented 4 Unit test cases (simple local tests). There is no need to include instrumentation or Espresso Tests (UI).	Nalan Kurnaz
When a user registers, their account information is stored in the DB	Abhay Ariyappillil
The Administrator can view the list of complaints	Abhay Ariyappillil
The Administrator can action the complaint (dismiss complaint or suspend tutor). For a temporary suspension, the Administrator the date when the suspension can be lifted The complaint disappears from the Administrator’s list once it is actioned	Sum Yan Wan
The complaints list is stored in the DB	Abhay Ariyappillil
The Tutor can see a message informing them that they have been suspended For temporary suspension, the message informs	Sum Yan Wan

them when the suspension will be lifted For permanent suspension, the message informs them that they can no longer use the application	
Optional – The group integrates with CircleCI to see the automated builds and automated test units execution	Nalan Kurnaz

Table 2: Distribution of Tasks in Deliverable 2

Features or Tasks	Name of the Team Member
Updated UML Class diagram of your domain model	Pierre Georges, Sum Yan Wan
The APK is submitted	Nalan Kurnaz
You implemented 4 Unit test cases (simple local tests). There is no need to include instrumentation or Espresso Tests (UI).	Nalan Kurnaz
The Tutor can add a topic to the profile	Abhay Ariyappillil
The Tutor can add a topic to the offered topics list	Abhay Ariyappillil
The Tutor can delete a topic from the profile	Sum Yan Wan
The Tutor can remove a topic from the offered topics list	Sum Yan Wan
When a suspended Tutor logs on, they can only see the suspension message and log off. They cannot perform any other action.	Abhay Ariyappillil
All fields are validated. There are appropriate error messages for incorrect inputs.	Nalan Kurnaz

Table 3: Distribution of Tasks in Deliverable 3

Features or Tasks	Name of the Team Member
Updated UML Class diagram of your domain model	Pierre Georges, Nalan Kurnaz
The APK is submitted	Nalan Kurnaz

You implemented 4 Unit test cases (simple local tests). There is no need to include instrumentation or Espresso Tests (UI).	Nalan Kurnaz
Final report	Nalan Kurnaz
The Student can search for a lesson The Student can see search results for lessons offered by non-suspended Tutors	Abhay Ariyappillil
The Student can view the Tutor's information and rating for each lesson in the search result The Student can view the lesson's information for each lesson	Abhay Ariyappillil
The Student can submit a purchase request The Tutor can receive the purchase request submitted by the Student	Abhay Ariyappillil
The Student can view the status of their purchase (pending, approved, or rejected)	Abhay Ariyappillil
The Student can rate the Tutor from which they have purchased a lesson	Sum Yan Wan
The Student can submit a complaint about a Tutor to the administrator	Abhay Ariyappillil
The Tutor can view and approve/reject purchase requests received from Students.	Sum Yan Wan
The Tutor can view their profile and rating.	Abhay Ariyappillil
All fields should be validated. There should be appropriate error messages for incorrect inputs.	Nalan Kurnaz
Optional – Student receives a notification when their purchase request is approved/rejected	Abhay Ariyappillil, Sum Yan Wan

Table 4: Distribution of Tasks in Deliverable 4

All the screenshots of our application

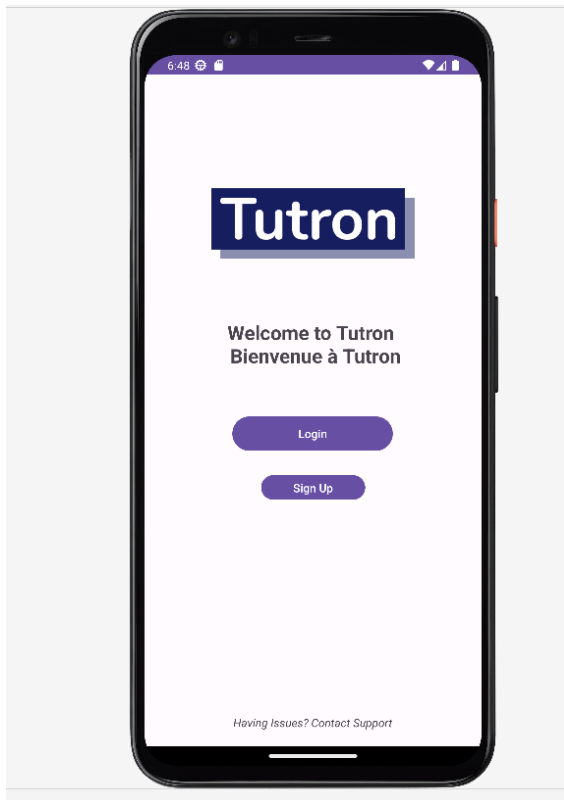


Figure 2: Welcome Page for Tutron App

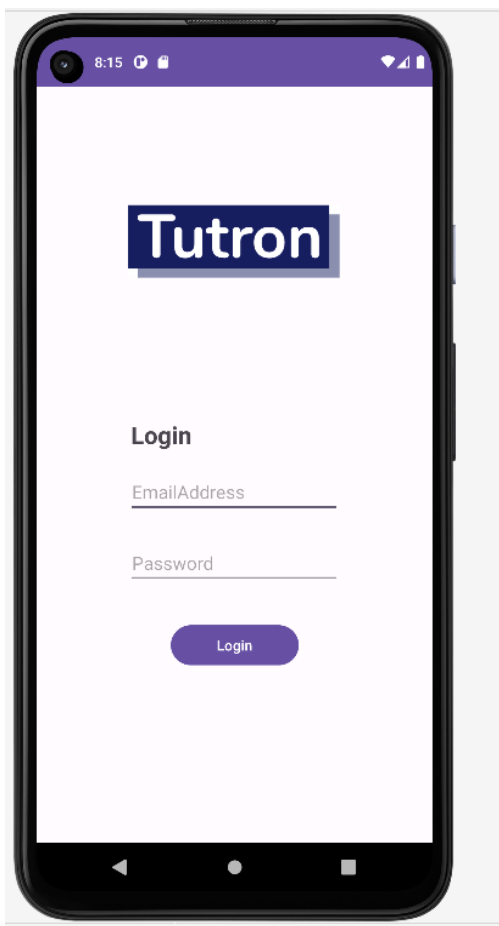


Figure 3: Login Page Display

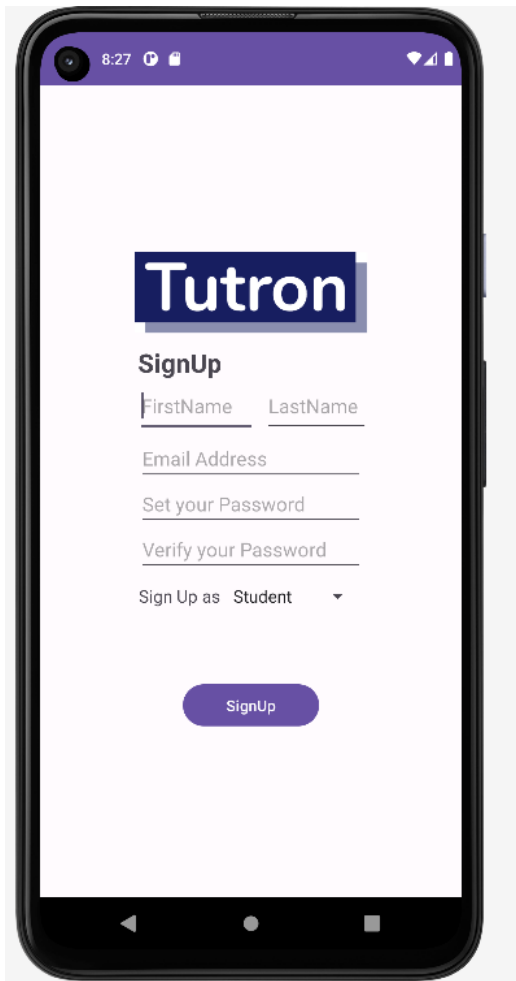


Figure 4: Sign Up page of Tutron Application

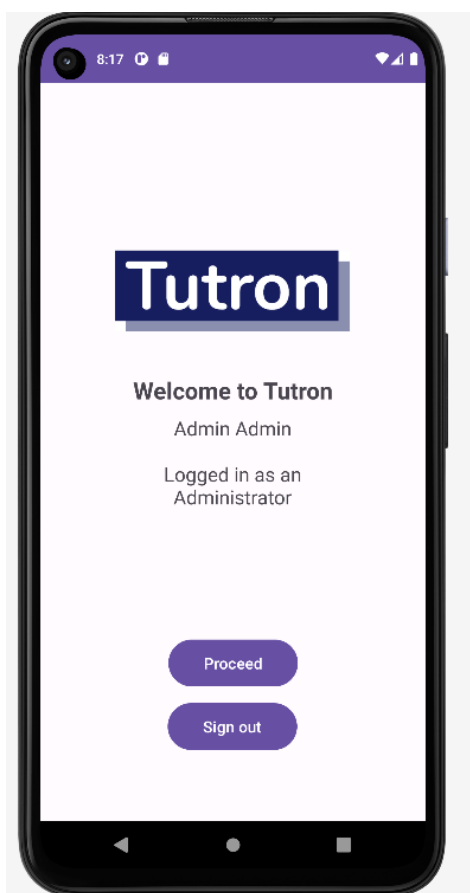
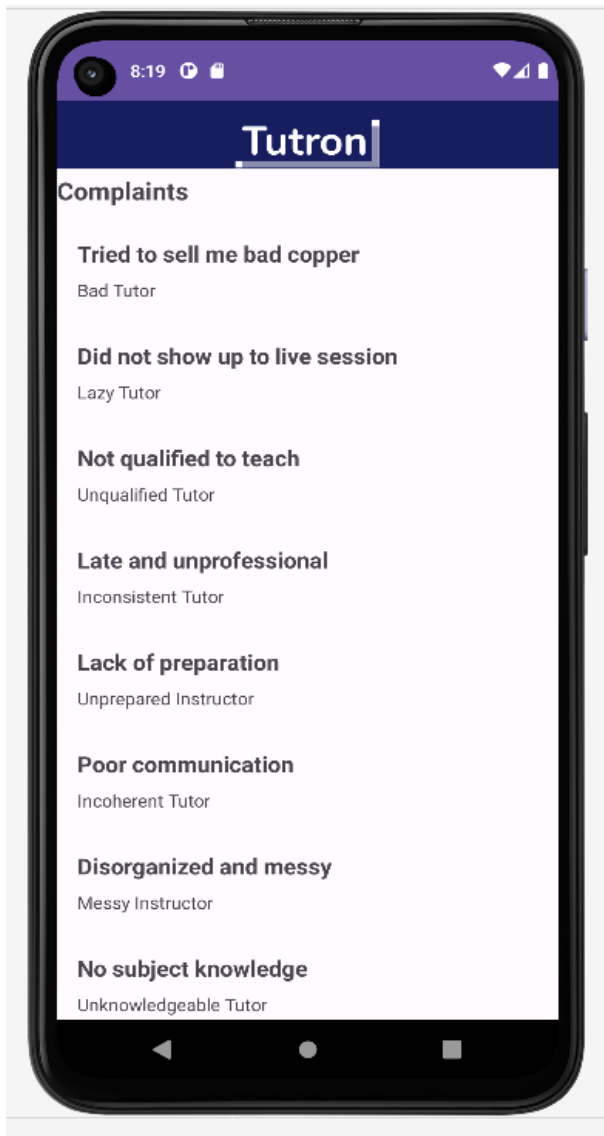


Figure 5: Administrator Homepage



Administrator can see all the complaints in a list

Figure 6: List of Tutor Complaints

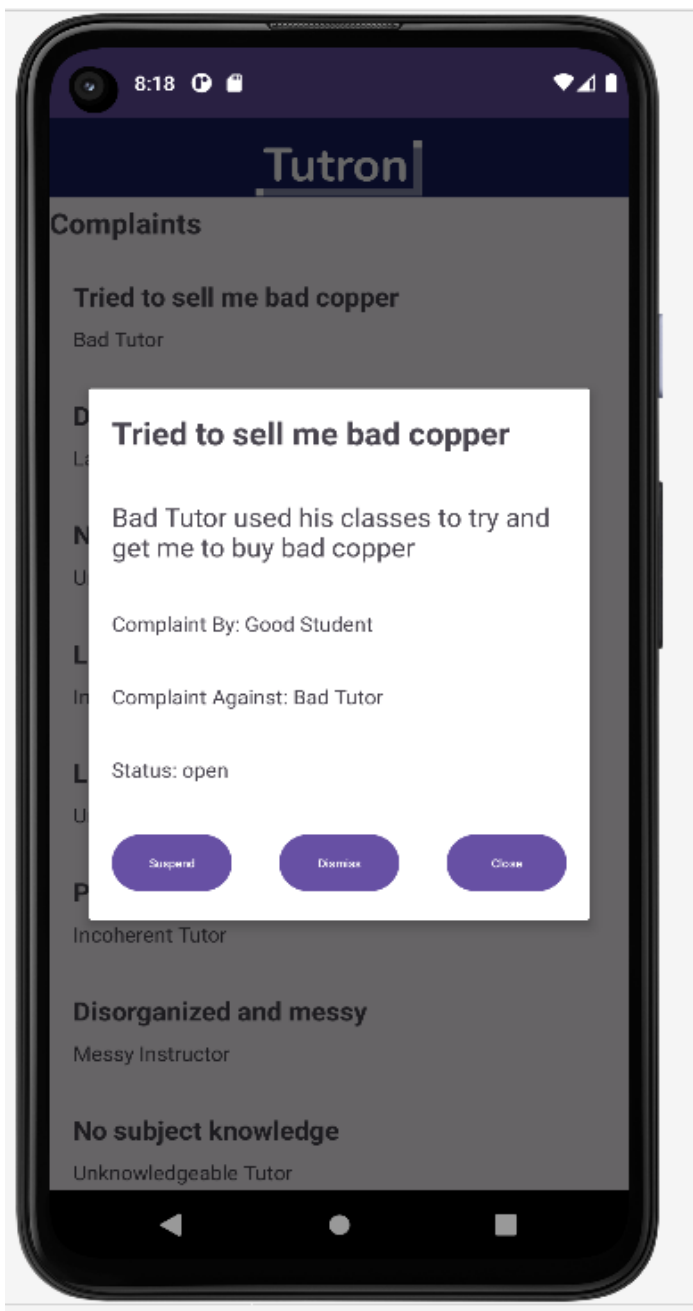


Figure 7: Options of Complaint Proceeding

Administrator can decide to suspend the tutor based on the complaint or the administrator can also dismiss the complaint if there is no problem.

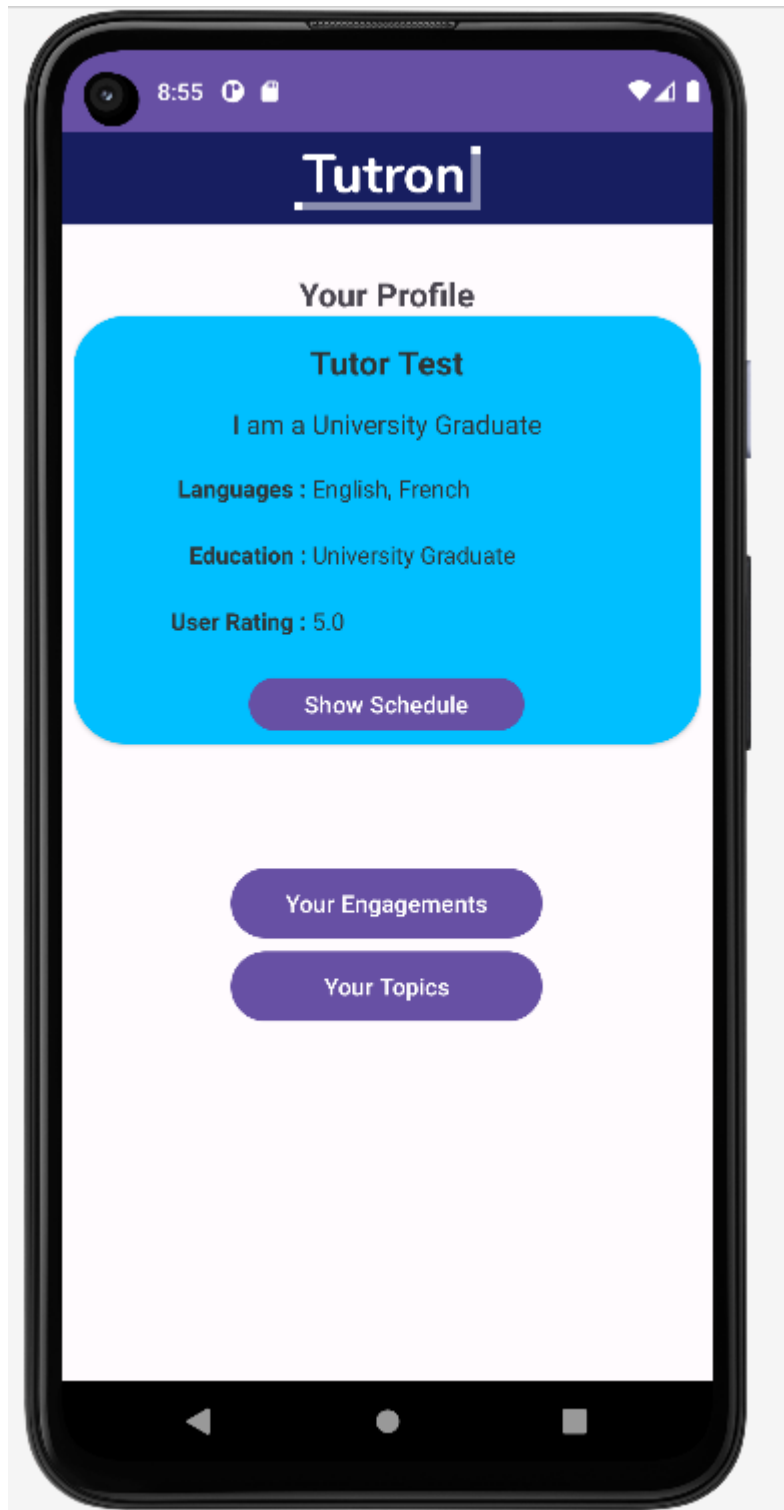


Figure 8: Tutor Homepage Display

Tutor is able to reach to Engagements where it leads to Pending Lesson Requests List and Accepted Lesson Request List. Tutor is also able to modify his/her offered Topics by clicking on Your Topics button.

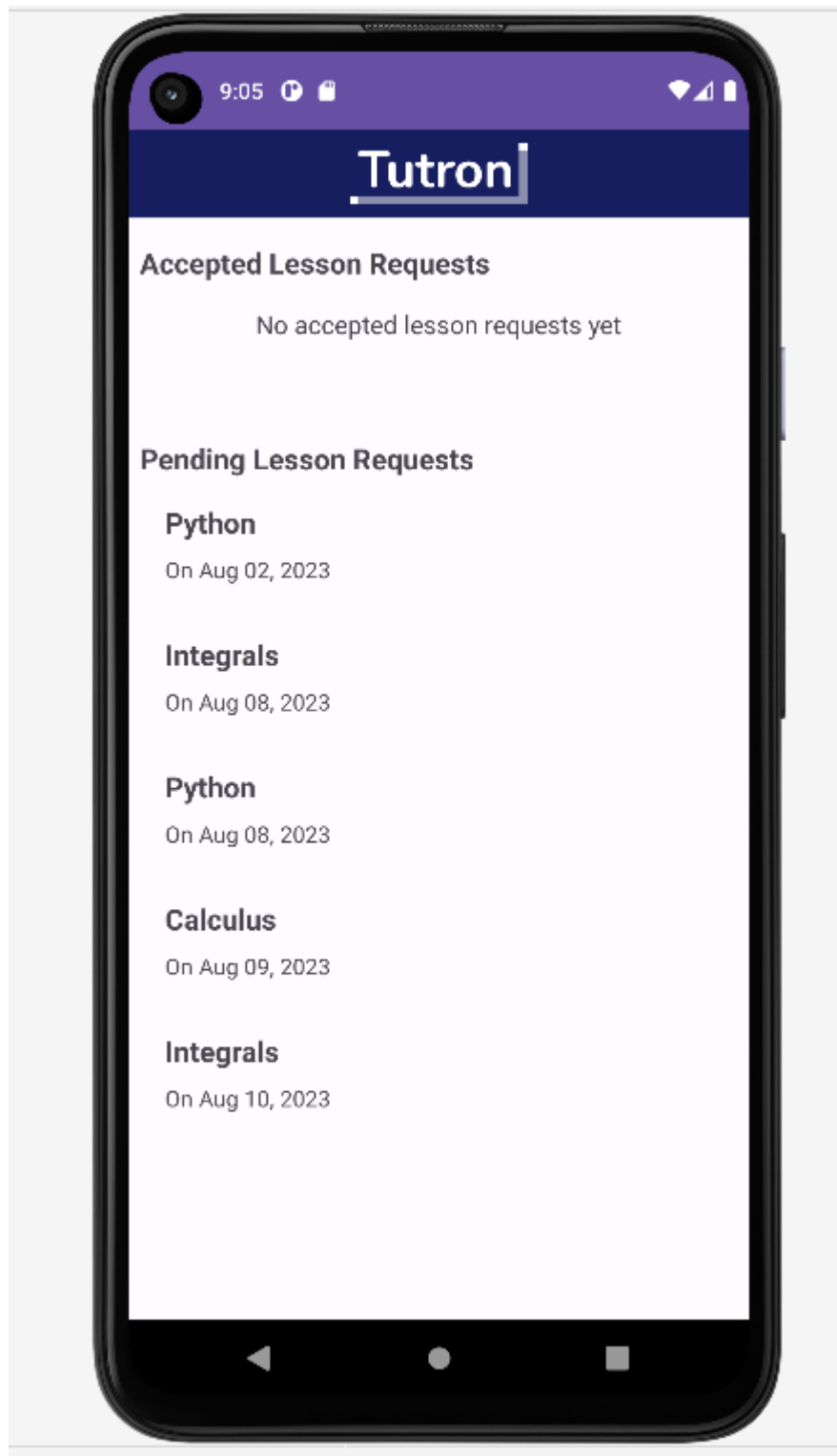


Figure 9: List of Pending Lesson Requests and Approved Lesson Requests

Procedures after clicking on Your Engagements button.

As described previously, the tutor is able to manage the requests from the student. When clicked on the topics on the Pending Lesson Request List, tutor can choose to approve or reject the purchase.

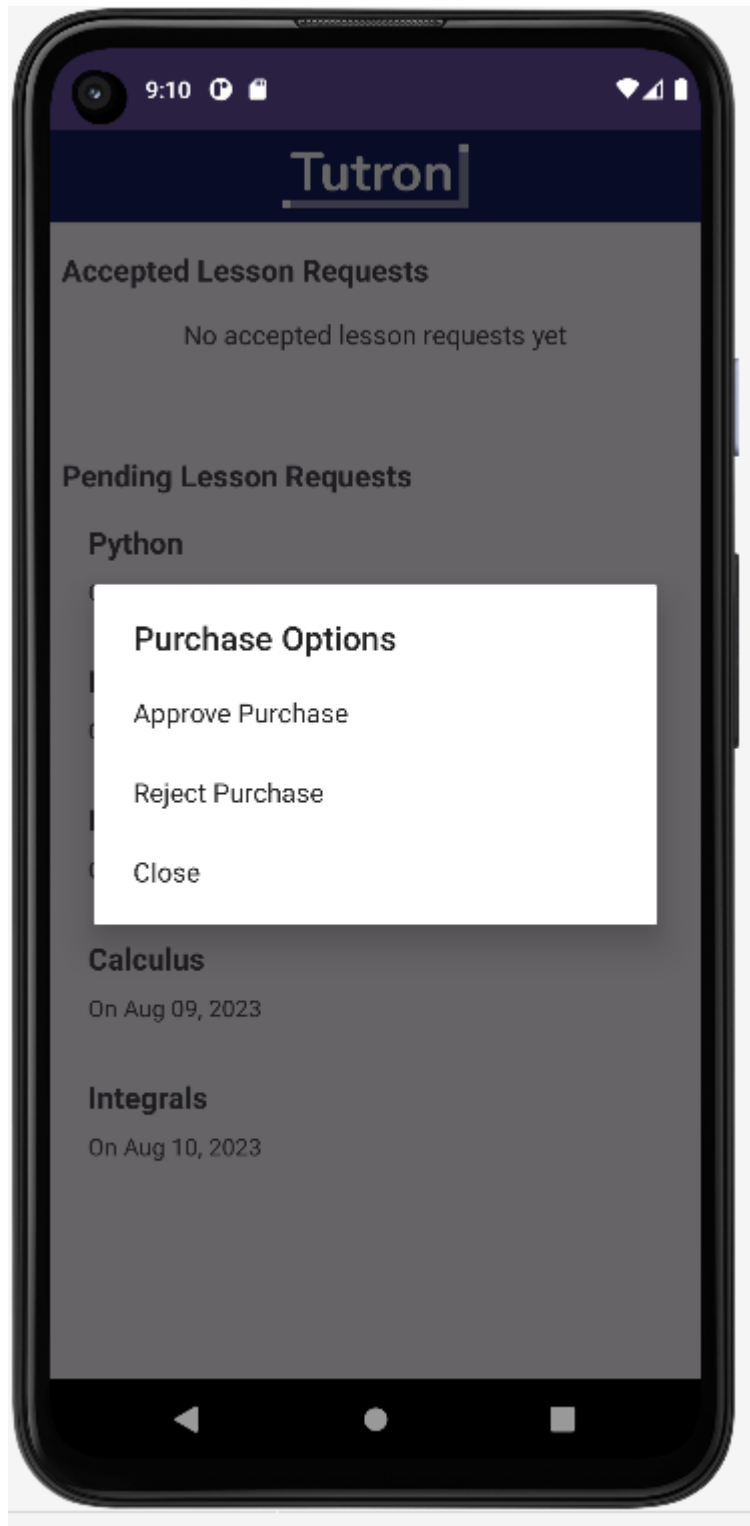


Figure 10: Options of Tutor when a student requests a lesson

As seen in the photo, the tutor has the authority to approve the purchase or reject the purchase of the lesson request made by the student.

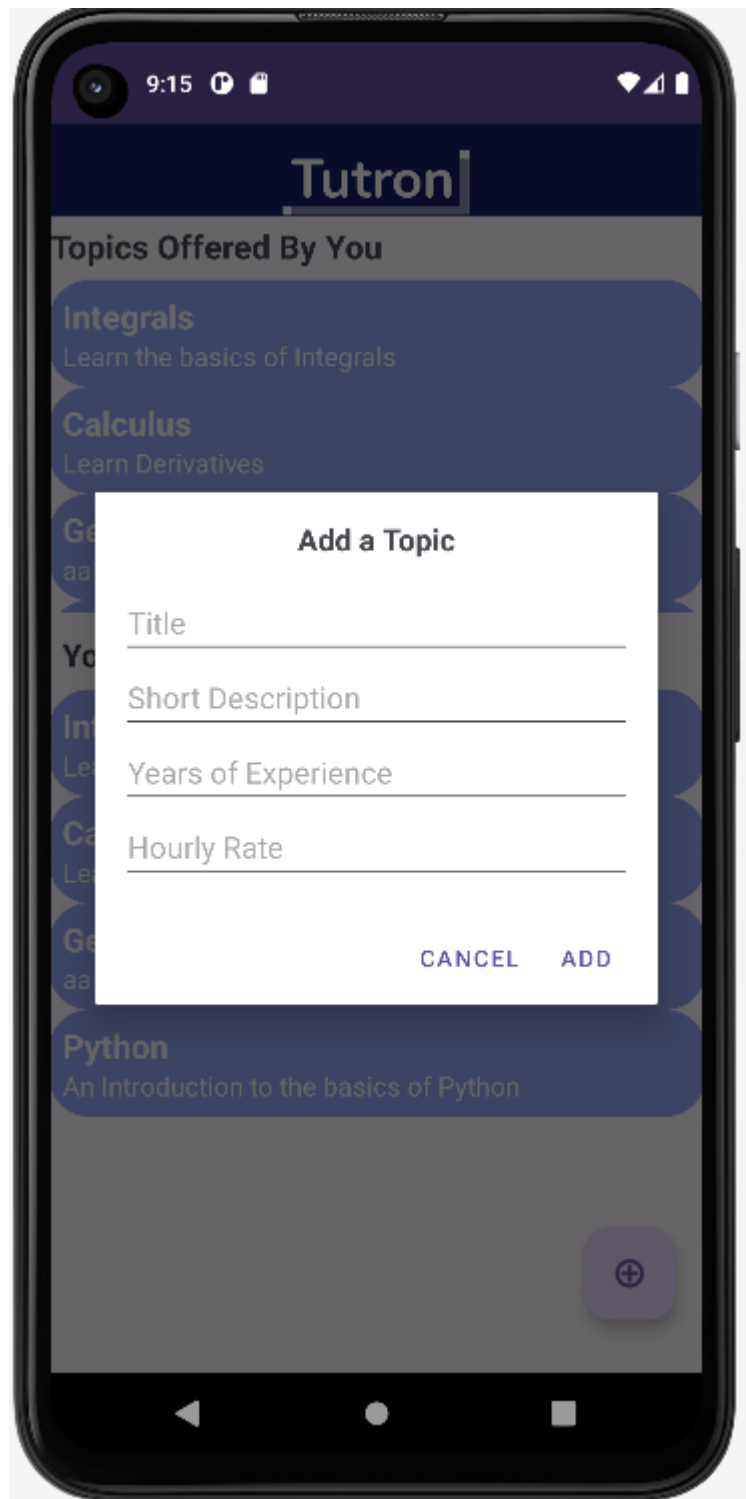


Figure 11: Adding a Topic

Tutor is able to add a topic through the application by clicking on the button on right bottom hand.

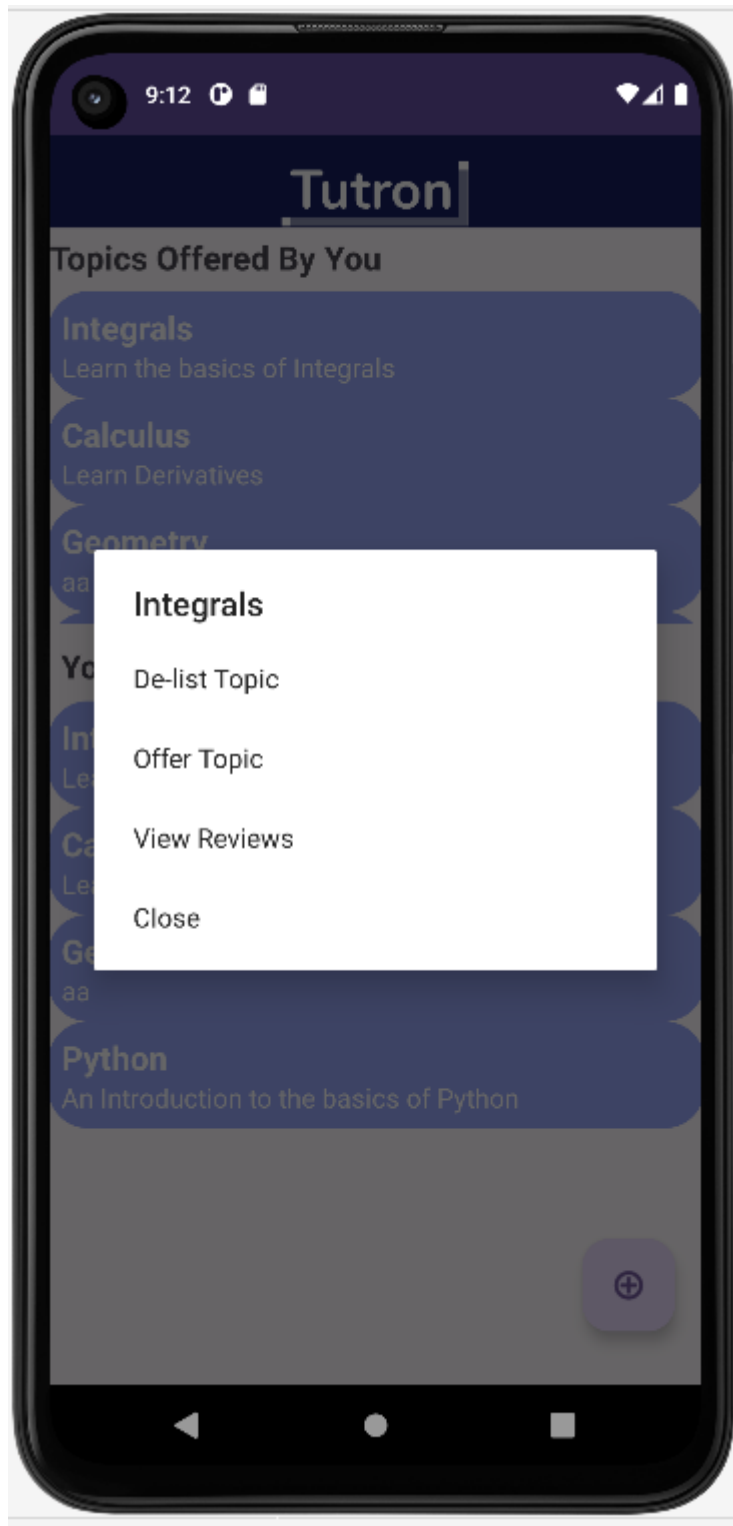


Figure 12: Display of Offered Topics List

Tutor can offer the topic for tutoring, s/he can view reviews and de-list topic if s/he doesn't want to tutor that topic.

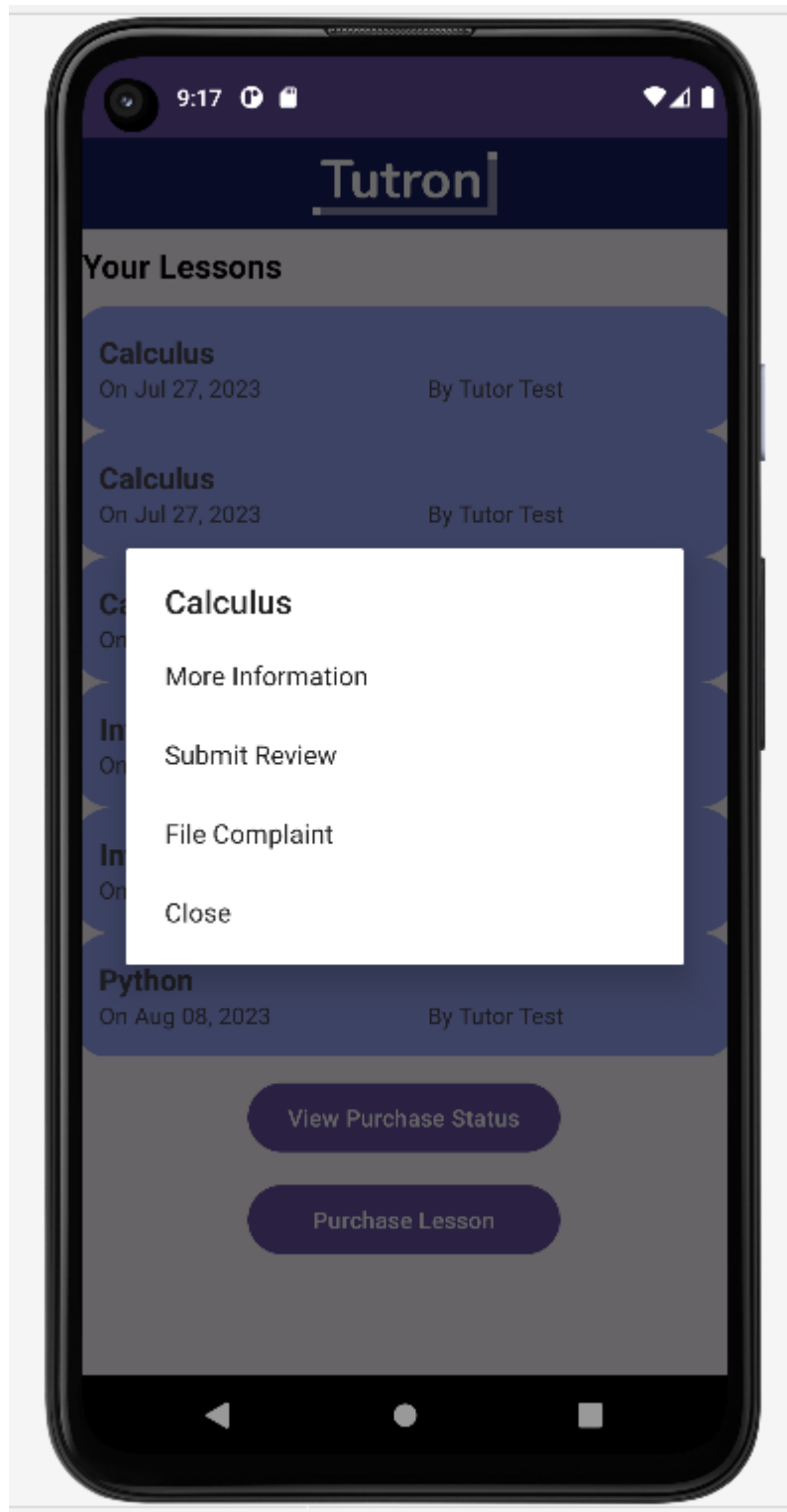


Figure 13: The student can choose to Submit Review, or can see More Information and can file a complaint.

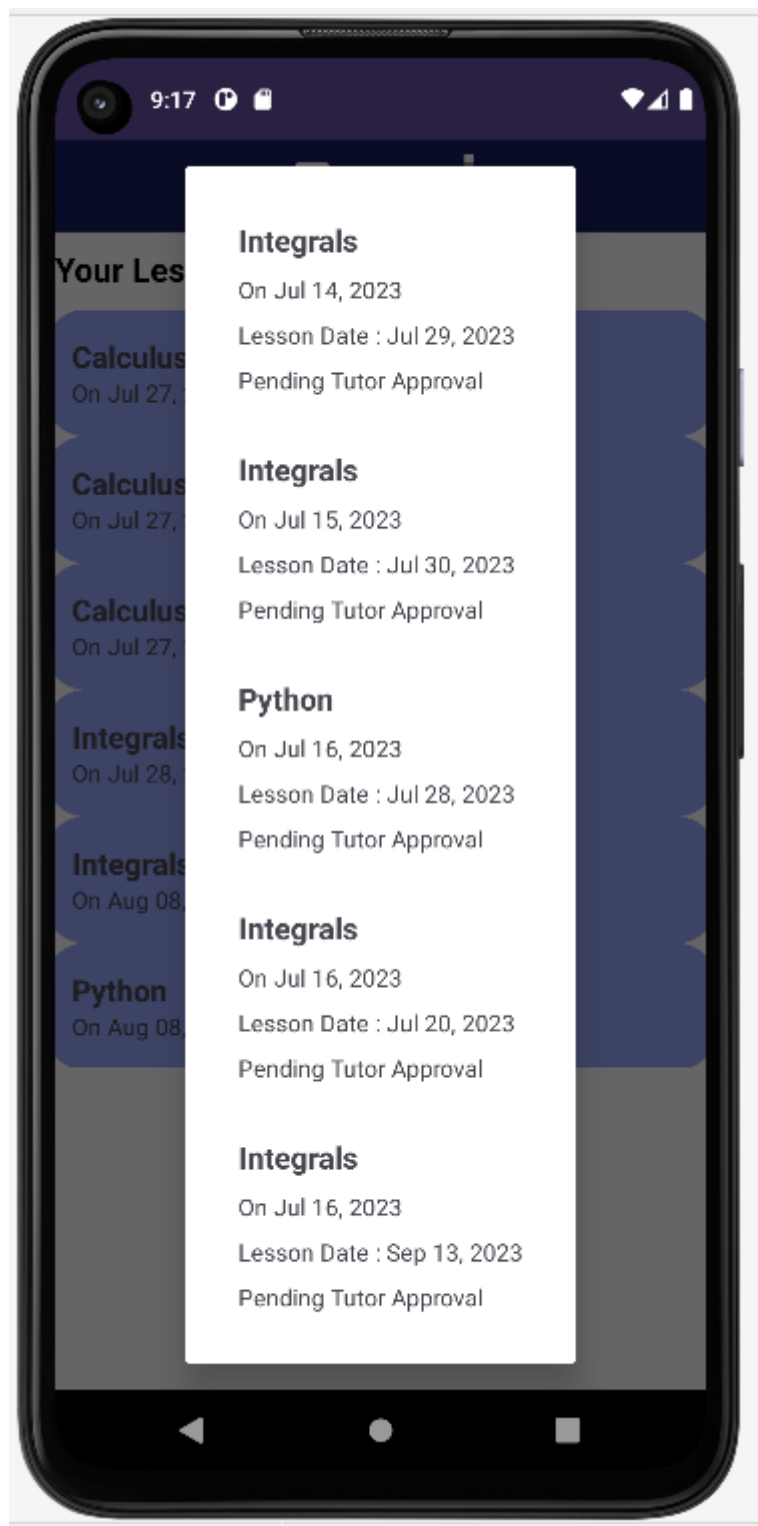


Figure 14: The student is able to see the pending topics and approved topics.

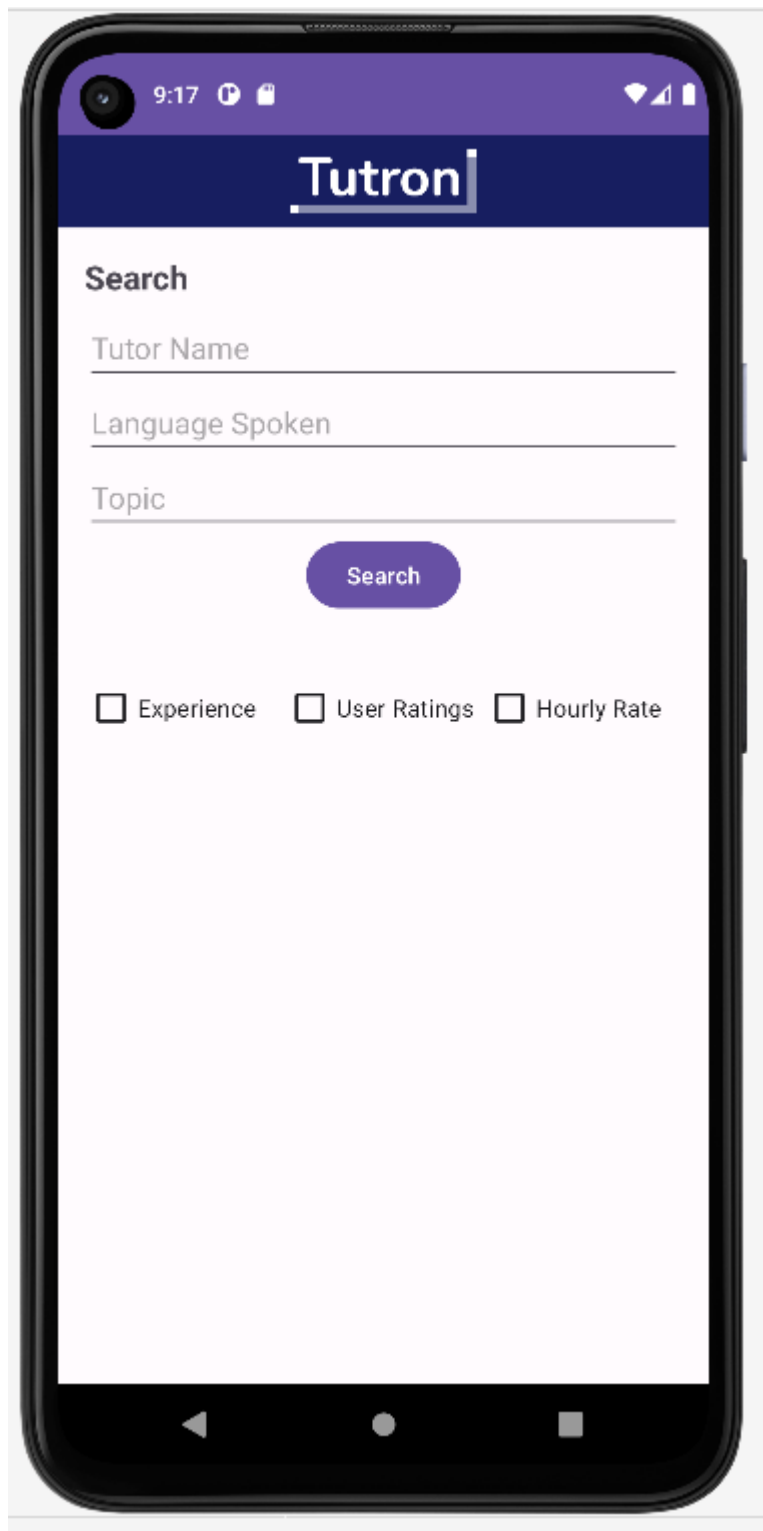


Figure 15: The student can choose and search topics by filtering in Tutor Name, Language Spoken and Topic Name.

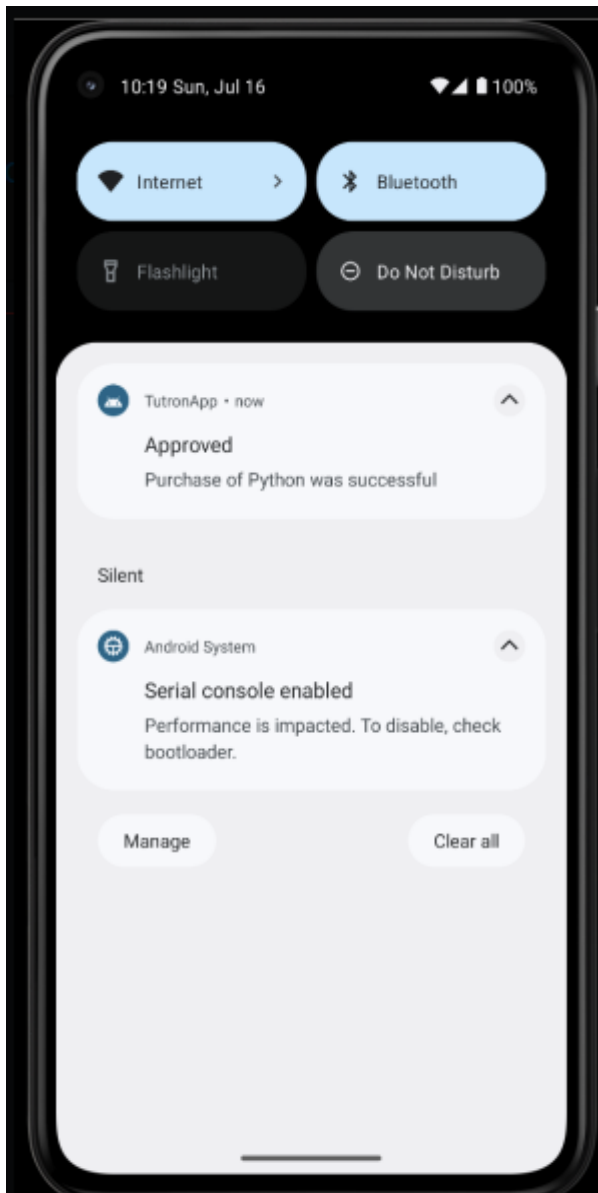


Figure 16: The notification that the student receives when the purchase is accepted or rejected

Student receives a notification when their purchase request is approved/rejected

Lessons learned

During the development of the Tutron application using Android Studio and Firebase database, we gained several valuable technical insights. Firstly, integrating Firebase as our backend solution provided us with a reliable and scalable infrastructure for storing and managing user data. We learned the importance of properly structuring and securing the database, implementing user authentication, and utilising Firebase's real-time features for data synchronisation. Additionally, we utilised various Android SDK components and libraries to enhance the app's functionality, including RecyclerView for efficient data presentation and Espresso for reliable UI testing.

During the development journey of the Tutron application, we acquired valuable knowledge in crafting a user-centric environment. By conducting extensive user research and incorporating iterative improvements based on user feedback, we deepened our understanding of our target audience's desires and requirements. This awareness played a pivotal role in shaping our design choices, resulting in user interfaces that are both intuitive and visually captivating. Our focus on simplicity and user-friendliness became paramount, enabling seamless navigation and effortless access to the app's fundamental functionalities.