Polimi Reviews
Design document

Vincenzo Santomarco
914688
Leonardo Romano
920337

Design and implementation of mobile applications
September 7th 2020

github.com/TheVinz/DIMA

Contents

1 Introduction

1.1 Idea

The idea of this application is to provide to Polimi's students a platform for exchanging information about courses, exams and professors, in order to help them in compiling the study plan and to find this information in a fast and useful way.
The idea was born from students' experience: the study plan could become difficult to be compiled when a student has almost total freedom in picking the exams and the choice of the courses is wide. Generally, the tactic is to read the information of all the possible courses and then ask some colleagues which exam they appreciated the most, found more interesting and why, in order to find out the perfect balance between interesting and challenging exams - and maybe the easier ones.
The problem is that sometimes there are some courses that on the paper are very interesting but there is no feedback on these, also the average student wants to find out how to compile in the optimal way his study plan but it is not easy at all to accomplish this task using only the few information we can retrieve from the Polimi's website.
So the question is: why shall we not use the knowledge gathered by sharing experience among Polimi colleagues?
Thus the development of this application is based on a specific need: to give students of Politecnico di Milano a platform for exchanging all this experience, gather it and give each other a powerful tool where they could find the knowledge for compiling the optimal study plan.

1.2 Functionalities

The functionalities currently implemented in the app are:

- **Course of study selection**: the user can select the courses inserting a specific school and degree.
- **Exams research**: If the user is searching for a particular exam or some keywords he can use the search feature for looking for exams whose name contains the given input
- **Exams selection**: The user can browse among all exams, or look only for the filtered ones.
- **Courses information**: The app shows the user some basic info of the exam, like name, CFU, teaching professor and a brief description of the contents (that is exactly the same found on Polimi official website), and then a rating given by averaging all the ratings students gave to that exam.
- **Rating**: The user can interact with the exams' reviews by leaving his personal rating and comment on the exams and also like the other users' review.
- **Reviews sorting**: Each exam page shows the students' ratings and comments for that exam and those ratings can be sorted according to four different orders: latest first, most liked first, ascending ratings, descending ratings.
- **Favorites list**: The user can mark some exams as 'favourite', saving these exams in the favourite tab and allowing faster access to the exam's page.
- **Review modification**: The user can delete and modify his own reviews for a specific exam.

1.3 Implementation choices

While implementing the app functionalities, the choice was to focus more on the usability and on making the app usage as much intuitive as possible, more than implementing a lot of features.
Thus the key feature of the whole application is to share knowledge gathered from students' experience in such a way that it wont get lost into the chaos generated by a complex app.
A lot of importance has been given at making a responsive and gladly to see interface: for example a lot of care has been taken in refreshing the exam page as soon as its global rating is updated, in order to have an instant feedback when a rating is added or removed, while avoiding useless and annoying reloading of the reviews list, and in animating some widgets for giving the app a more fluid look.

2 Design

2.1 Architecture

This app uses Google Firebase for managing the users' login and registration and for storing and retrieving all the data relative to exams and reviews.

2.1.1 Data model

This section will describe which information is stored into the database and how it is represented.

User A user object is created every time a new user registers via Firebase authentication system.

- **UserID**: a unique ID that identifies the user, created by Firebase.
- **UserName**: a nickname created by the user when it sign up to the app, it is showed at the top of every review submitted by the user.
-

School The school object identifies a school of the Politecnico di Milano.

- **Name**: the name of the school that uniquely identifies it.
- **Degrees**: the list of the degrees available for this school.

Degree The degree object identifies a degree available at Politecnico di Milano.

- **Name**: the name of the degree that uniquely identifies it.
- **Exams**: the list of the exams available for this degree.

Exam The exam object identifies a precise course held at Politecnico di Milano, there exist one exam for each pair name, professor.

- **ExamID**: id of the exam generated by Firestore.
- **Name**: name of the exam.
- **Professor**: teaching professor for this exam.

- **CFU**: CFU value for the exam.
- **Description**: a brief description of the arguments of the exam, taken from Politecnico di Milano official website.
- **NumReviews**: the number of reviews currently submitted for this exam.
- **Score**: the average rating of the exam, expressed as a double with a minimum of 0 and a maximum of 5. In case the NumReviews field is 0 this filed may not be present on the database, and is by default initialized at 0 on new instances.
- **Reviews**: list of the reviews submitted for this exam.

Review The review object defines a single review submitted by a user for a given exam. There can only be at most one review per user for each exam.

- **ID**: the ID of the review given by Firebase.
- **UserID**: the ID of the review's author.
- **Author**: the username of the review's author. This information, even if it could easily be retrieved via the UserID, is stored for performance purposes.
- **Comment**: a comment leaved by the author.
- **Timestamp**: timestamp telling when was the review submitted.

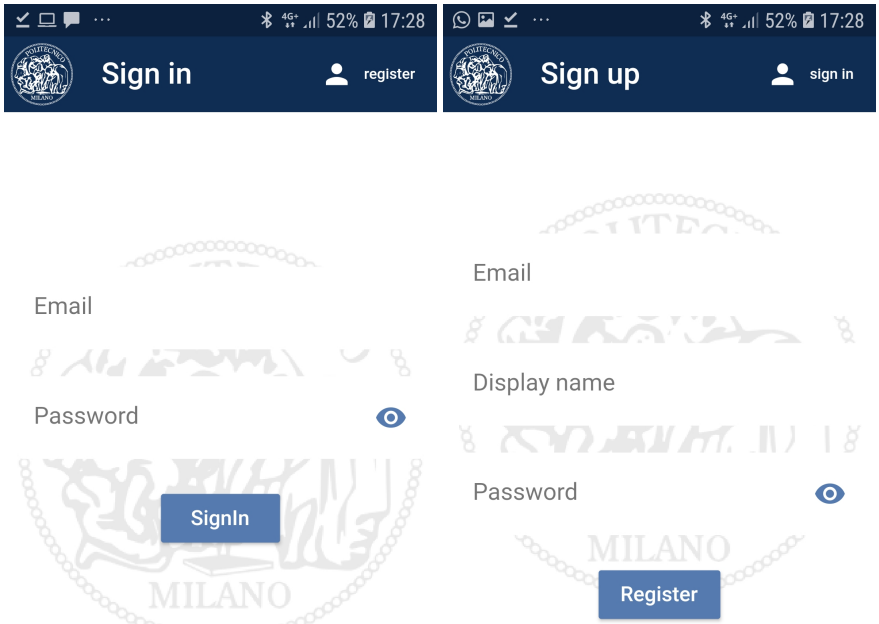
2.1.2 Data management

The data is stored on Google Firestore, and is retrieved by the application via streams. This means that every change of the data into the database is immediately notified to the application that update the user interface accordingly. It is important to point out that changes to the reviews present into an exam page are notified via DocumentChanges: this means that when a review is submitted, removed or updated the whole collection is not downloaded again, but the single change is received by the app instead, resulting in less computation by the device and in rendering a smooth animation representing the event, without rebuilding again the whole list widget.
In particular the functions implemented via Firebase are:

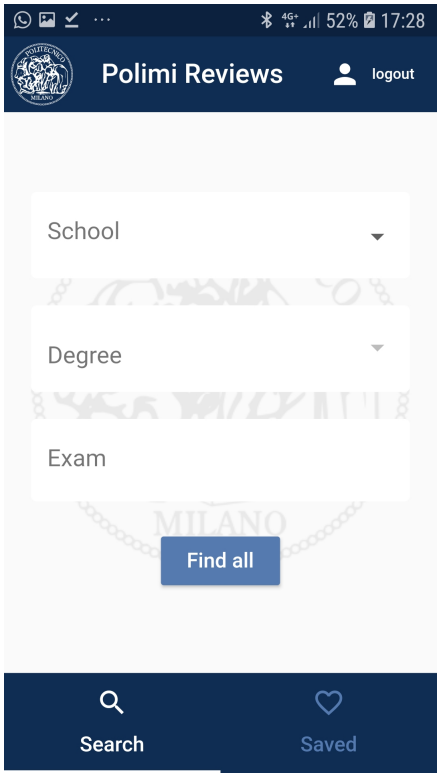
- Registering and logging in via email and password
- Storing and retrieving user data, like the username and the list of saved exams, via Firebase's user id
- Retrieving schools' name and degrees
- Retrieving degrees' name and exams
- Retrieving exams' name, teaching professor, cfu and description
- Listening for updates on exams' score
- Updating exams' score
- Storing and updating users' reviews
- Listening for single changes on reviews

2.2 User Interface

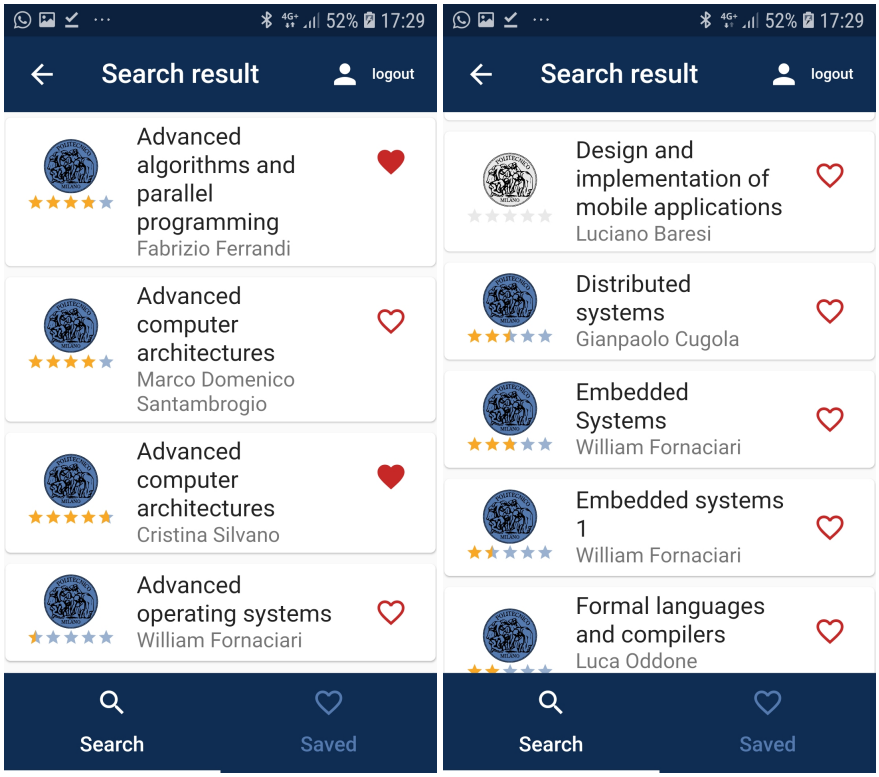
The main screen of the app is based on a TabView that contains the main `search page`, for searching exams, and the `Favourites List page`, for accessing the saved ones. Those pages have their own navigation stack that keeps its state whenever the user navigates from one tab to the other, allowing a more intuitive and fluid navigation.



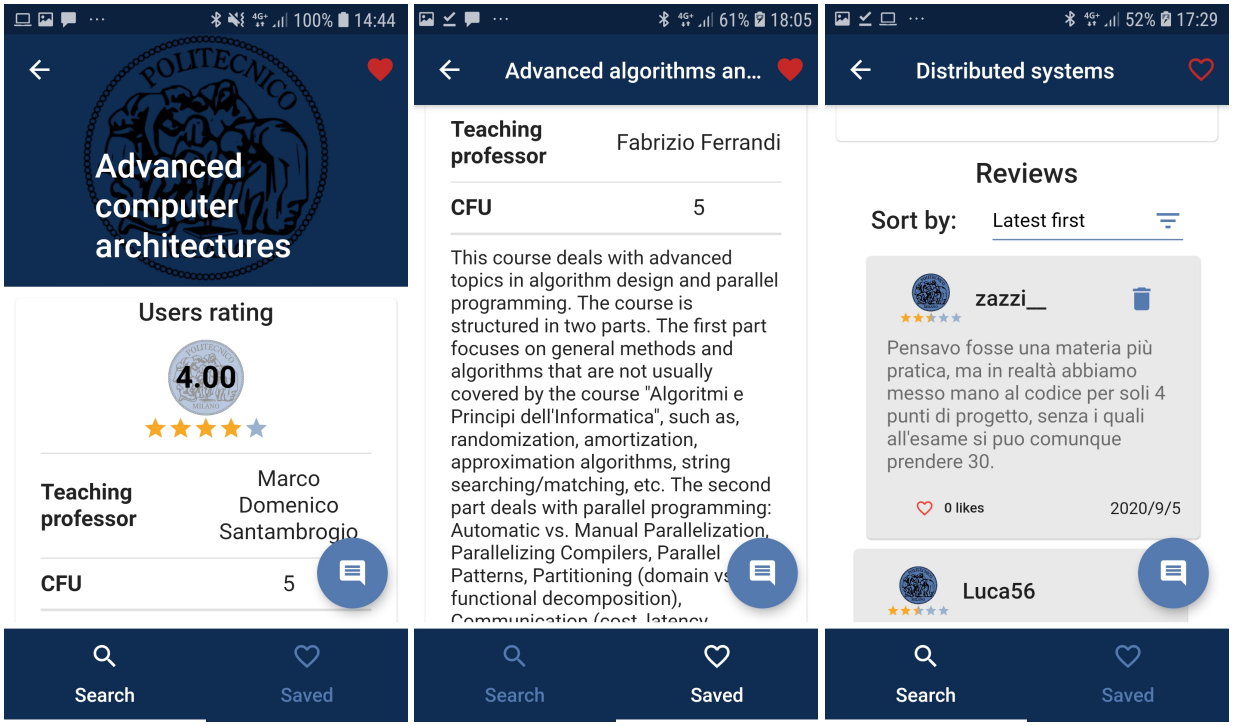
Authentication page The authentication mechanism is composed by two screens: one for registering a new user and the other one for logging in with an already existing one. For switching between the two screens a button on the right of the AppBar indicates the page the user is going to switch to by clicking it. The form fields present in this page supports the focus mechanism: once a field has been filled the user can click 'next' on his keyboard to focus the next field, until the last has the 'done' option that removes the focus. The password field has an icon button for trigger the password visibility.
For logging in or register it is necessary to press the button at the bottom. Once the user is logged in the the `Search page` is shown.



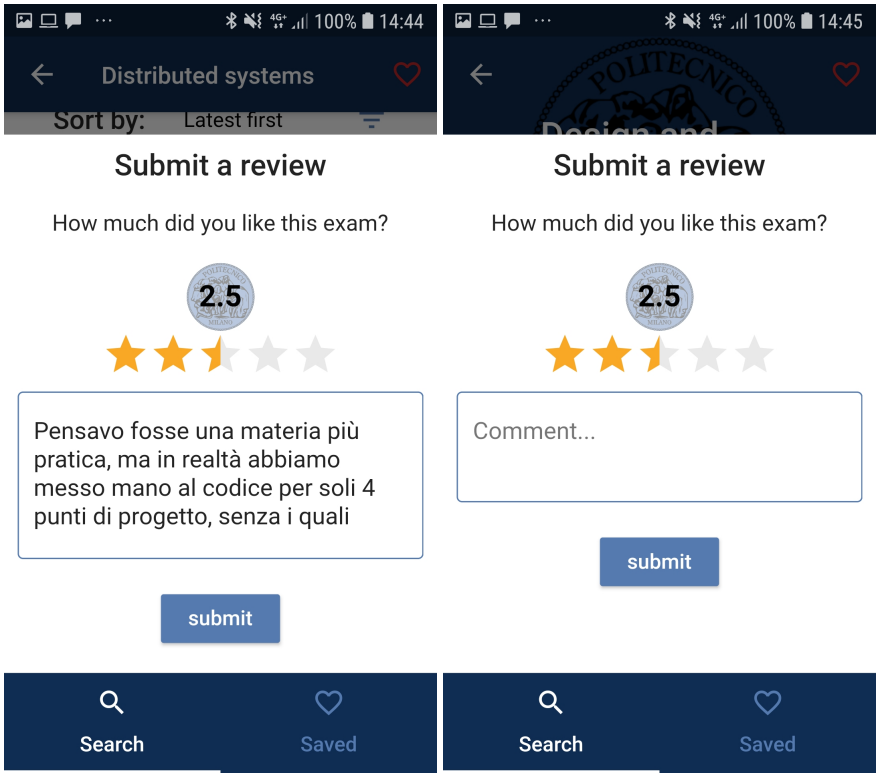
Search page Main page for searching an exam. On the right of the AppBar a button allows the user to log out, while on the center there is the form to be compiled in order to perform the search. This form is composed by two dropdown menus: one for selecting the School and the other one for the Degree, the latter changes according to the degrees fetched by querying the database with the School data. Compiling those fields is mandatory in order to perform the research. A text form allows the research of a particular exam belonging to the given degree: the result of the research will be every exam whose name contains the prompted sequence of chars, if this field is left empty every exam of that degree will be shown. Finally at the bottom there is the button that allows the user to perform the search: in order to help the user understand the real role of the text field, once a char is prompted into this field the text of the button switches from 'Find All' to 'Search'. Pressing the button the user will be forwarded to the `Search Result page`.



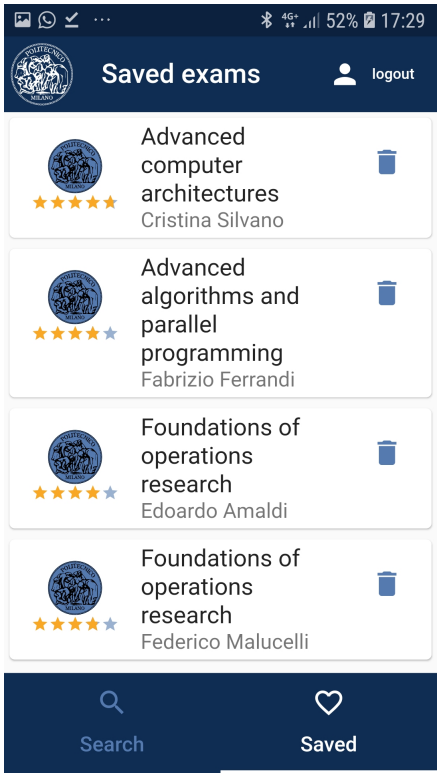
Search Result page This page shows a list of every exams found via the search form. Each tile shows name and teaching professor of the exam, together with a star indicator of the total score given to that exam by averaging users review. If no review is found the theme of this header will be grey, indicating to the user that there is no information on the server for calculating a real score for that exam. On the left of each tile a heart shaped button permit the user to add or remove the given exam from its favourites list: a filled heart means that exam is already saved, so a click would remove it from the favourite list, while an empty heart means the exam is not saved yet, so by clicking the 'save' action is performed. By clicking on the tile the `Exam Detail page` for that exam is opened.



Exam Detail page This page shows the details of a given exam. The name of the exam is shown on the app bar, while the overall rating, the CFUs, the teaching professor and the description are in the body. Also here the overall rating of the exam is expressed with a stars representation, together with a number that represents the exact averaged value, rounded up to the second decimal. At the bottom of this page all the reviews submitted by the users are listed.
On top of this list a dropdown menu shows the sorting options for the reviews, then each review tile shows review's author, rate, comment, number of likes and date, together with a trash icon for deleting it in case the author is the current logged user.
Whenever a new review is added it is placed in the right position according the current sorting.
The logged user can also like the reviews of the exam both pressing the heart icon or double-tapping the liked review. The reviews are aligned in a Whatsapp-message like mode: other users' review are aligned on the left side of the screen, while the current user's review is aligned on the right side of it, this is due to make easier for the user to find its own review.
An heart shaped button on the AppBar allows to add/remove the exam from the favourites list, while the Floating Action Button shows the `Submit Review form`.



Submit Review form This Modal Bottom Sheet allows the user to rate and comment the exam. For rating it a star-shaped slider allows the user to insert a rating from 0 to 5, with steps of 0.5, while the current selected value is showed on top of it. On the bottom a text field requires to be filled with at least 6 characters for submitting the user's personal opinion about the exam.
This form is by default set to 2.5 rating and empty comment, in case the user did not submit any review yet for the current exam, otherwise the form is initialized with the review the user already submitted for this exam. This allows the user to understand that if a review is already present into the list, every new submit will lead into in an update of that review, with a new timestamp.



Favourites List page This list is structured like the `Search Result page`, with the difference that each tile here does not have a hart button as trailing, but instead a trash icon: this icon allows the user to remove that exam from the favourites list. Clicking on a tile will open the corresponding `Exam Detail page`.

2.3 Navigation

As already mentioned, the main page of the app is divided into two tabs: search and favorites. The user can navigate among those tabs by swiping right and left on the screen or by clicking on the corresponding icon on the bottom bar. Each tab has its own navigation stack, so when pressing the 'back' button on the phone the activity at the top of the current stack is popped. If the user is on the `Favourite List page`, pressing the 'back' button will drive the user to the search tab.

2.4 Assumptions

Since we could not deal with a real database containing all the possible exam present into the official Politecnico di Milano website, for data managing purposes some assumptions on the size of the data have been done:

- The app filters locally the exams resulting from a 'search' operation: since for the user it is mandatory to select a degree while searching an exam, in the worst case scenario the application will have to deal with around 300 different exams. Since the exam object is just a set of strings the user should not issue any lag, but clearly in order to asses that the app should be tested with a real and fully populated database.
- The app sorts the users' review locally every time a new review sorting filter is selected: again it is supposed that the number of reviews will be around 200 for each exam, also it is expected that every 2-3 years the exam may change contents/professor so the old one should be deleted and a new one added to the database, also it is possible, in some worst case scenarios, to mark as 'too old' all those reviews done more than a couple of years before, and thus deleted via a database trigger.

3 Testing

The app has been tested taking particular attention in avoiding graphic glitches. It has been stressed by performing several unusual actions by the user and checking that the interface kept working as intended. Also the consistency of the data has been tested when those are accessed and modified concurrently from different devices.
Then the app has been used for a couple of days by a set of students: they have been asked to perform some tasks and to give then some impressions on how hard was to accomplish to those tasks and how intuitive it was to understanding what they were doing and which was the final purpose of the applications. Typical tasks were:

- Create an account
- Log in
- Look for all this exam for a particular degree
- Look for a specific exam
- Submit a review
- Update a review
- Add and delete some exams from the favourites list

Then the questions they were asked to were:

- Among this list of exam, which of them do you think you will probably add to your study plan?
- How useful has the review sorting mechanism been in evaluating if an exam was good for you?
- How fluid was the navigation all around the app?
- Did you always had a clear idea of where you were into the app?

Thanks to their advice the app changed look several times, in particular for finding an optimal way, like the star-bar score, for showing intuitively the exam's evaluations.

4 Conclusions

In conclusion this app wants to meet the needs of an average Polimi students, giving them an easy but powerful tool for making decisions about their academic career. The strengths of this app are its easy, elegant and smooth user interface and the immediacy by which all the knowledge gathered on the platform can be retrieved. The app is very simple, tough it gives the user everything he would like to get from an app like that, and achieves one simple but crucial objective: give him the best user experience possible.