**FCI E-Campus**

App’s Functionalities:

1. The app will allow each student, professor and TA to own an account but everyone has special permissions and capabilities in the system. The student does not need to be verified he can sign up normally providing basic information about himself, then he can register in the courses by a providing a course code (similar to Acadox) and also a group number (each student can be in only one group in each course). The professor and TA must be verified because they have more options like upload file, add materials, add tasks and more.

There are two types of students, normal student and moderator student. Each course has its moderator students that are chosen by the course’s professor (and maybe TAs also). This moderator student is just a normal student but he has higher permissions in the course. Moderator students can upload materials in the extra resources section of the course and also manages the course’s forum.

The verification of professors and TAs will be an automated process (similar to WhatsApp’s signup) and it will be as follows:

In the app’s sign up page screen will be an option to sign up as a student or teacher. The professors and TAs will choose to sign up as teachers. It will be mandatory for them to sign up with their FCI academic email (ex. [m.nassef@fci-cu.edu.eg](mailto:m.nassef@fci-cu.edu.eg)). The app will verify that the provided email exists in its list of the faculty’s current doctors and TAs academic emails and then the app will automatically send an email to the specified email containing a secret code and the doctor/TA can open that email and put the secret code in our app’s sign up screen and continue creating the account.

1. A Schedule with courses lectures, labs and sections each student has according to his registered courses. This schedule shows the user what he/she has at every slot of the day (like lecture, lab, section or delivering assignments and tasks or exams). Each student has his own schedule which include only stuff related to his registered courses.
2. Most of new students don’t know the location of their lecture/section halls which makes a big problem to them. They spend a lot of time searching for their lecture/section hall or they miss the lecture/section, we can solve this issue by adding a static map for the faculty.

This map is a static graphical map which guides the students by showing the buildings and the location of each lecture/section hall as well as professors’ offices, TAs’ office, the library, the mosque, the restaurant, etc…

As well as the students can search for a location by typing it into the search bar and the map will refer to their needs if any.

1. For each course the student is registered in there will be a resources screen which includes the materials for this course. The materials can be files of any type (PDF, Word, PowerPoint presentation) as well as any links to external resources such as videos or websites and also notes from the material uploader.

Students can view the materials and download them. Professors and TAs can upload new materials for the courses they teach.

In the resources screen there will be extra materials section which will include the materials uploaded by the moderator students. These resources can be also files or links such as summaries or links to useful websites or videos.

This material will be added in the server side, which can be requested to download by the frontend users (Students). Course resources and materials can only be viewed by the course’s registered students only.

1. Allow Professors & TAs to add materials which can be files, of any type, and “useful links to video, audio, text files or websites” to the course they assigned to, then a notification is sent to all the students assigned to this course informing them that there is a new material added to this course and when they click on the notification the resources screen for this course is opened, we also then can add highlighting to the new materials added.
2. Allow Professors & TAs to add tasks to the courses they are assigned to. These tasks can be assignments, lab tasks, etc... Each task will have a deadline and a description which can be a separate file or just a text viewed within the assignment’s screen. Also each task can be set as a task that requires students to upload their solutions or not. Only professors and TAs can add tasks and only for the courses they are assigned to. Students registered to that course will get a notification for the new tasks and they can view it in the course’s tasks screen and later they can upload their solutions, if a solution is needed.
3. The app will have an Announcements Screen to inform students with any new announcements made by the Admin (similar to announcements in FCI E-com). The announcements screen contains announcements that does not relate to a specific course. The announcements will be information related to the faculty as a whole (ex. informing students that a day is an official holiday, informing students about the date of the first day of the term, etc…). All the students signed up to the app can see this announcements screen even if they are not registered in any courses.

When a new announcement is added, Notification will be sent to all users of the app (including students and teachers).

1. Each course will have a Q/A Forum. Course’s students can post in the course’s forum to ask a question (related to this course) and other students, professors and TAs can answer.  
   Moderator students can delete any forum post if they find that it was inappropriate such as unrelated to the course “or asked before (extra)”.

The forum post will be Q/A-based and each post, very similar to stack overflow, will have a “solved” flag that the student who asked the question can set and it will indicate that this post is fulfilled and it doesn’t require more solutions (however more can be added). This will help to focus on the unanswered questions.  
The app saves all this forum posts and allows students to view them and search for a specific question by “tags”.

1. The Notification system in the application will be divided into two subsystems.

**The first subsystem** is responsible for notifying the users for any **new** incoming changes on the application. This will include notifying the users for new announcements, new assignments, new tasks added, new quizzes and new exams.

The notification system will not be a background service, but it will collaborate with the synchronization service on the client app. This sync service is responsible for listening to events on the backend. These events will be such as the arrival of a new announcement or the addition of a new assignment on the courses that the user is registered in. The sync service will update the application and calls the notification system to show a notification to the user.

**The second subsystem** is a reminder background service that is responsible for showing reminders notifications. Reminder notifications will be managed locally without the need for any backend sync (very similar to an alarm application). Take this example: If a user has an assignment that will be delivered on Thursday the reminder service will send a notification before a week from the deadline, if the user hasn’t marked this assignment as done, the reminder will continue and sends a notification to the user before 4 days from the deadline, then again if it’s not marked as done it will send a notification to the user before 2 days from the deadline and so on it will continue to send notification before a day, then before 6 hours, … and so on until either the user either marks the assignment as done or the deadline is passed.

***Extra Stuff:***

1. The app will allow users (professors, students or TAs) to send direct private messages to any other user. This is **not a chat service**. The messages will be similar to emails with a title, body & attachments such as images or raw files. The user can search for another user by his name or his ID in our system, then he can send a message to him as described above. Messages will be grouped in an inbox page and categorized and organized by the user it was sent to or received from (similar to chat rooms) so that all messages sent to or received from the same user appear in a single specific place inside the inbox.

~~Then we will develop on top of that a Profanity and Spam Filtering System. This will filter the messages before they’re received to ensure it doesn’t contain any spam, curses, inappropriate words or sexual references. If a message is detected as spam, it’ll not be received and the user who sent it will be reported to Admin and warned. This filtering system can be developed either by using a static dictionary that contains all the spam words and curses as far as we can, then when a new message is being sent the system will scan this message and compare each word in it to all the words in the dictionary by searching in the dictionary. The scanning and searching phase can be greatly improved by using a data structure such as a~~ [**~~Trie~~**](https://en.wikipedia.org/wiki/Trie)~~. This filtering system will exist on the backend server not on the client devices, so resources is not a problem. Also the filtering system can be even more extended to use machine learning & AI to discover new curse statements and spam situations by itself.~~

1. The app supports that students can upload their solution for a task “if the task needs a submission” and the doctors or TAs that are assigned to this course only will have a full access to all the submitted assignments and can assign a grade to each one of them on the app. Each student can see his grade only; they will not be public.
2. The app will have a history for each course which contains the course materials from previous years and schedule report. Users can view and download the materials for this course for a user-specified previous year.

When the course ends and the doctor sends to the admin that the course has ended, so the admin will archive this course and save its material by year like “concept Spring-2017”.

And make this course back to its free state “can put new doctor, new TAS and new students can register in it”, materials will be saved for view only and anyone can download it but notifications and forum will be deleted.

Example scenario is when the doctor flags the course as finished, the admin approves that the course is ended and return it to its free state, then saves materials and schedule report by its year then this course goes to old courses then save it and anyone can see this course materials and schedule report.

Note: The current version of the course will be private and only students who are registered to the course can view it. But the old version of the courses which are archive, can be seen by any of the app user even if not registered to the course.

Users Roles:

1. **Student**: student can register to courses and view its schedule, materials, forum, tasks, etc…
2. **Moderator Student**: This is just a normal student but with more permissions in the course. These permissions are that he can add materials in the extra materials section of the course (moderator student’s materials won’t be mixed with the official materials sent from doctors or TAs) and he also manages the forum helping deleting inappropriate posts.
3. **Doctor/TA**: both of them can be assigned to the courses and they can add materials, forum posts, tasks or assignments there. Each course can have one or more doctors. The course’s doctors are responsible for choosing and approving the moderator students of this course. Also each course can have many TAs.
4. **Admin**: this user represents the faculty and can be a specific person hired by the faculty (one of the student’s affairs employees for example). He maintains the app, maintains and update the list of the faculty’s professors and TAs’ FCI academic emails and also creates courses. He assigns doctors to the courses and assigns TAs to the courses specifying the group of each TA that he/she teaches in the course (TA can have more than one group and doctors have all groups). When the year ends he archives the current courses (saves its materials + schedule report) and resets them so they are ready to be assigned to new professors and TAs and registered by new students and lastly he can add schedules for the courses (schedules will be added manually or from an excel sheet).

The admin also can receive reported problems from any app user. He will receive them and be notified for them.

App’s Views:

This is a minimal imagination of what the client apps UI should contain and provide.

1. The student’s view contains:
2. **Dashboard** containing the stuff he has in his schedule today. It will just display today’s schedule in a more focusing way.
3. **Calendar** which will be similar to the mobile phone’s calendar containing all his lectures, labs, sections and tasks from his registered courses (it will be the merge of the registered courses schedules).
4. **Tasks** (all tasks done + in progress).
5. **Registered courses** which will be a list of all the courses he is registered in.
6. The teacher’s view contains:
7. **Dashboard** (same as student).
8. **Calendar** (same as student).
9. **Assigned courses** which will be a list of the courses that the doctor/TA teaches currently.
10. The admin’s view contains:

The admin will have a separate website allowing him to do the roles specified above.

1. The course’s view contains:
2. **Tasks** view containing all the tasks that was added to the course (either finished or in progress). The teachers can add more tasks and students can upload and submit for each task or mark them as “Done”.
3. **Schedule** of this course. (The same for students and teachers and when tasks finish it’s not deleted from the schedule).
4. **Resources** containing two sections the first is for materials uploaded by the course’s doctor or TAs and the other contains materials uploaded by moderator students.
5. **Forum**.