

Guidelines

Design: (If you follow them, you will create a revolution. Again.)

1. Design should be aesthetic, easy to use and elegant.
2. Selected colours in the app should look beautiful and unified.
3. Colours can also be used for guiding focus to particular section on the screen.
4. Colour in text (if used) should not decrease its visibility.
5. Every screen should have minimal but sufficient number of Menu Options and/or Buttons.
6. Every screen should show only required Options/Buttons.
7. Menu Options should be grouped logically to reduce confusion.
8. User should be able to navigate through the app effortlessly and smoothly.
9. User should get appropriate response from the app on every apt touch.
10. User should be able to reach any screen from anywhere in the app in not more than two button presses.
11. Design should be intuitive and should look familiar even if using for the first time.
12. Toasts should be concise, clear and should correctly respond to a specific action of the user.

(For further reading, please check out the following two links:

- [Android Design Principles](#).
- [Material Design](#).)

Coding: (Make your code better and life simpler.)

1. Always add comments to your code. (Very Important.)
2. Always keep a backup of the last working code.
3. Use camelcase for naming variables and functions. For eg. Instead of naming variable as `user_slots`, name it as `userSlots`. And naming a function as `Sort_Subjects()`, name it as `SortSubjects()`.
4. Name of variables should convey their role in the code clearly. For eg. If you want to make a variable for list of teachers, name the variable as `listOfTeachers` and not as 'list' or 'lot' or 't' etc.
5. Make variable and method names as proper as an unknown person can also come to know why it has been created for.
6. Keep your code modular and group same functions under same classes.
7. Even if you write a sample to test a particular feature, write it using the coding guidelines.
8. Remove unused variables and methods. Keep your code clean and organized.
9. Don't make unnecessary threads.
10. Don't write lengthy methods, break it into chunks and use them.
11. Free any allocated memory before exiting the main thread.
12. Put spaces wisely.
13. Test your code in every way possible, before releasing.
14. Write a code which as beautiful as Candice Swanepoel ;-)

Suggested Future Developments:

(Great ideas and great actions change the world forever.)

1. 'Attendance' section can be included in the app, with downloading course material, where students can check their updated attendance and check modify it to check percentage change if they attend or miss classes in future. (Similar to VIT Academics app.)
 - *This can be done using the tabbed structure. For eg. in Whatsapp, tabbed structure is used to separate Calls, Chats and Contacts.*
2. 'Holiday' Section can be added, in which all the holidays' for the current semester can be viewed in an attractive way.
3. 'Pull to Refresh' can be added, like in Gmail app.
4. 'Assignment, References' can be included with the course material.
5. A small separation can be added between CAT-I, CAT-II, Term End material list, so that user can clearly distinguish between the three categories.
6. 'Events' section can be created, in which User will get the updated information about the events happening in VIT on that day, their timings and venue.
7. 'Alerts' section can be created, in which, user can add their Quiz dates for the subjects and the app will notify them about the quiz a day before.
8. User-Interface enhancements could be done according to the latest material design guidelines.
9. iOS version of Slate.

