**Guidelines for Final Term Project**

1. Term Project will be done in group of two members.
2. Groups can be made across sections.
3. Students can start project after approval only.
4. Working on project will be started from May 7, 2021. Github Repositories should be created from May 7, 2021 and add [samyan.uet@gmail.com](mailto:samyan.uet@gmail.com) as collaborator.
5. Progress on project will be tracked from Github accounts
6. Project will be evaluated based on rubrics described in the document(all the rubrics are based on basic requirement).
7. List of Group Ids are given on last page.
8. Name of repository should be in this format: CS162S21GIDxx e.g. if group id is 9 then repository name should be CS162S21GID09
9. Bonus tasks will be compulsory for the groups having 3 members.
10. Deadlines will be as followed:

|  |  |
| --- | --- |
| **Milestone** | **Deadline** |
| Selection of Project | May 5, 2021 |
| Detailed requirement document | May 20, 2021 |
| GUI Prototype | May 23, 2021 |
| Object Oriented Analysis and Design | May 30, 2021 |
| Implementation of Design | June 10, 2021 |
| Project Documentation | June 15, 2021 |
| Project Testing | June 20, 2021 |
| Resolution of Issues | June 25, 2021 |
| Project Viva | Any time after June 15, 2021  (One member can be chosen only) |

* Basic requirement are being shared, you are required to add more features on your own and write detailed requirements to remove any ambiguity. Detailed requirement should mention the UI design using pencil tool.
* In all of these projects, basic requirement to save data in txt/csv files. For bonus credit, students can save data in word, excel or pdf format
* Tools to be used for project
  + Github
  + StarUML
  + MS Word
  + NetBeans/Visual Studio
  + Pencil Tool

**Basic Requirements to be implemented in the project.**

1. Minimum 10 domain classes
2. Minimum 8 software classes
3. At least 5 classes overall should contain the parametrized constructors as well.
4. Multi Level Inheritance (2 examples)
5. Multiple Inheritance (2 examples)
6. Composition (3 examples)
7. Abstract Classes (at least 3)
8. Polymorphism (at least 3 examples)
9. Use of Upcasting
10. Use of Down casting
11. Singleton Pattern
12. Strategy Pattern
13. Validators
14. Use of collection classes of ArrayList, LinkedList, Queue, Stack, HashSet and TreeSet
15. Handle exceptions
16. UI should contain text boxes, password fields, radio button, check boxes, dropdowns, date selector fields, text areas, scroll bar, tables, buttons inside table, use of panels for at least 3 forms, file menu on each screen, use of same form for add and edit.
17. Data storage and load from at least 5 files with at least one file having complex format.
18. Email sending
19. Creation of two generic classes