**Analysis of Algorithms**

Spring 2020

**Members Details**

| Group ID | CS311S20PID30 |
| --- | --- |
| Registration Number of Group Members | 2018-CS-22  2018-CS-33 |
| Section | A |

**Project Details**

|  |  |
| --- | --- |
| ***Project*** |  |
| Project Title | Activity Scheduling Tool |
| Executive Summary | Our project is an Activity Scheduling Tool. It is a desktop application that generates the timetable automatically after providing it with certain information. The input given to make timetable will be the list of teachers, classroom in which class will be conducted, contact hour list for each subject in a week, list of courses of a session, and the user has also to tell if it is a lab or theory subject. We are using an “Activity Scheduling” algorithm to make an automatic timetable. Our algorithm will select the activity to be scheduled and will place it in the available slot keeping in the view the subject's contact hours and will manage timetable accordingly. We have an option to choose either a web or desktop application for our project. We made a Desktop application for our project. We worked in .Net Framework with C# (C Sharp) language for the frontend and used SQL server 2014 Management Studio for database and backend. In SQL Server 2014 Management Studio we used DDL (Data Definition Language) and DML (Data Manipulation Language). In the activity scheduling tool, the program generates a timetable depending upon the number of contact hours of the subject. It arranges the subjects such that it takes minimum days of the week and cover all subjects according to its contact hours. It has 5 working days; Monday, Tuesday, Wednesday, Thursday, and Friday. The algorithm checks where slots are available. It also checks the contact hour of the subject and keeping in the view all its dependencies it allocates the lecture slot to the subject. The room will be selected by the user itself. After generating the timetable the user can log in again and can see its generated timetable whenever he/she wants. For this, the user needs to remember the userID he entered while generating the timetable. |
| ***Business Case*** |  |
| Outline the business need for the project | *[This section to contain a clear articulation of the business need in the form of a statement that addresses the problem or opportunity. This statement should be no more than three or four sentences]* |
| End user of the product | *[Clearly identify the real life domain and user that you are going to target]* |
| Motivation for Project | Timetable generator was a different project than the other two that were file compression and plagiarism checker. These both were boring projects than the timetable one in our view. Moreover, in the timetable generator, we just need data from the user and we will present data to the user. It means that there is not much use of file handling.  *[This section to contain a clear statement of motivation which drives you to this project]* |
| Description of the project objective(s) | *[Identify the key objectives of the project]* |
| State the level of impact expected should the project proceed and implications of not proceeding | *[State whether the implementation would have an impact at an operational level and/or strategic level and state the impact(s) in 2-3 lines]* |
| Functional Requirements | *[State list of features/services that you have impemented* |
| ***Benefits*** |  |
| What benefits are expected/ anticipated? | Using this program the user can schedule all the activities in minimum days. It means that maximum activities will be arranged in early weekdays. |
| ***Implementation Details*** |  |
| Link to Github Repository | https://github.com/ayeshatahirme/CS311S20PID30.git |
| Total Number of commits in repository before 8th December 2019 | No commits before 8th December. |
| Exact contribution of each member | **2018-CS-22**   * Drew interfaces in Milestone 1. * Wrote correctness and complexity analysis of the algorithm. * Created all the interfaces of the frontend of the project. * Implemented algorithm. * Created all database tables. * Integrated database with the project. (Made connections) * Added all the functionality except log in and signup in the project. * Stored entered data from user input textboxes to the database. * Wrote project configuration document. * Report writing.   **2018-CS-33**   * Wrote about algorithm selection in Milestone 1 * Wrote pseudoCode and calculated the cost of the algorithm. * Added log in and sign up functionality in project. |
| ***Commits in github repository by each member*** | |
| |  |  | | --- | --- | | **Member Registration No.** | **Total Commits** | | 2018-CS-22 | 80 | | 2018-CS-33 | 11 | |  |  | | |
| **Details of commits** | |
| |  |  |  |  | | --- | --- | --- | --- | | **Sr. No.** | **Details of commit** | **Date** | **Member Reg No.** | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| Have you used built in algorithms or you have implemented yourself? |  |
| Formats of input | *In which format, input will be given to your system? Provide complete details on input formats.* |
| Validations | *List the validations that you have applied on input with complete details* |
| Format of output | *In which format, output will be expected?* |
| Deployment | *Have you deployed your project in any format? If yes, provide the details* |
| ***Details of algorithms*** | |
| *In this section, you are required to provided details of algorithms used in the project with the pseudo code, one paragraph description of each algorithm, complexity and correctness.* | |
| ***Interfaces for your project*** | |
| *[Add actual interfaces of your project with description of each control in the UI]* | |
| ***Integration*** | |
| *What type of difficulties were faced by you while integration of UI and algorithms. What was your strategy in this regard.* | |
| ***Change Requests*** | |
| We made only one change that was of our algorithm. We first decided to work in the Activity scheduling algorithm and submitted it in the “decision of algorithm” on its respective deadline. We could not implement our algorithm according to what we chose. Then we decided to change our algorithm to the Genetic Algorithm and tried to understand it. We submitted the pseudo-code of the algorithm but its cost was too much that we had to make many changes in the algorithm and finally we implemented our algorithm. It is the greedy choice algorithm. It has a linear cost. It means it is an efficient one. Its cost is θ(n) and it seems to be a good and efficient one, | |
| ***Testing*** | |
| *In this section, you are required to mention the issues report and solution proposed.* | |
| ***Technology*** |  |
| Programming Language | C Sharp (C#) |
| Platform | Desktop Application |