University Of Central Punjab, Lahore
Project Deliverable

<u>Data Structures and Algorithm</u>

Group Members	
Name	**Roll Number**
Sheroz Akram	L1F20BSSE0454
Sikander Ali	L1F20BSSE0452
Talha Hameed	L1F20BSSE0462

Introduction:

Social Networking Tool is an app which manages different events of UCP. Students create new accounts in the application. Already existed users can login with their email and password. User can add new Events to their account. The app will automatically manages the events in a Heap System.

By Using heap students will get an events that will occur before all the other events. The app also make use of time library to remove those events that are passed away. Students can Post to their timeline and it will also be stored in the database. Students will be able to add new friends and remove already existing friends. They can list all their friends from the menu.

Functionalities:

1. Create Account.
2. Login
3. Read Data from File
4. Display Next Event
5. Display Recent Timeline Posts
6. Add a New Event
7. Heap bases Event Management System
8. Add new Post to Timeline
9. Automatically Remove Passed Events
10. Write the updated data to File
11. Add or Remove Friends
12. AVL based Friend Management System

Class Diagram:

AVL

- NodeAVL * Root

+ AVL()
+ Delete(string) : void
+ Display() : void
+ DisplayAll() : void
+ getRoot() : NodeAVL *
+ Insert(string) : void

NodeAVL

- string * Data
- int Height
- NodeAVL * Left
- NodeAVL * Right

+ NodeAVL(string)
+ getData() : string
+ getHeight() : int
+ getLeft() : NodeAVL *
+ getRight() : NodeAVL *
+ setData(string) : void
+ setHeight(int) : void
+ setLeft(NodeAVL *) : void
+ setRight(NodeAVL *) : void

DoublyLinkedList

- Node<T>* Head
- Node<T>* Tail

+ DoublyLinkedList()
+ Insert(T Data) : void
+ Display () : void
+ getHead() : Node<T> *

Node<T>

+ T Data
+ Node<T>* Next
+ Node<T> * Previous

+ Node(T Data) : void
+ getData() : T
+ getNext() : Node<T> *
+ getPrevious() : Node<T> *
+ setData(T) : void
+ setNext(Node<T> *) : void
+ setPrevious(Node<T> *) : void

Event

- Int Date
- Int Month
- Int Year
- int Hour
- int Minute
- string Description

+ Event(*)
+ Read() : void
+ Display() : void
+ DisplaySmall() : void
+ setDate(int) : void
+ setMonth(int) : void
+ setYear(int) : void
+ setHour(int) : void
+ setMinute(int) : void
+ setDescription(string): void
+ getDate() : int
+ getMonth() : int
+ getYear() : int
+ getHour() : int
+ getMinute() : int
+ getDescription() : int

MaxHeap

- Event * arr
- Int Current
- int Size

+ MaxHeap(int)
+ getEvents() : Events *
+ getSize() : int
+ getTotal() : int
+ Insert(Event) : void
+ Display() : void
+ IsHigher(Event , Event) : int
+ Swap(int , int) : void
+ Top() : Event

HashTable

- int Capacity
- DoublyLinkedList<Student> *Table

+ AddNewFreind(string , string) : void
+ AddPost(string , string) : void
+ Display(string) : void
+ DisplayFreinds(string) : void
+ getNextEvent(string) : Event
+ hashFunction(string) : int
+ HashTable() : void
+ Insert(Student) : void
+ isHigher(Event , Event) : int
+ Login(string , string) : int
+ newEvent() : void
+ Rear() : void
+ RecentTimeLine(string) : void
+ RemoveFreind(string , string) : void
+ Search(string) : int
+ Write() : void

Student

- string Class
- string Email
- Maxheap * Events
- AVL * Freinds
- string Name
- string Password
- string * Timeline
- int TimeNo
- int Year

+ Student (*)
+ Display() : void
+ DisplayFreind() : void
+ DisplayTimeline() : void
+ InsertNewTimeline(string *) : void
+ NewEvent(Event) : void
+ AddPost(string) : void
+ NewStudent() : void
+ RemoveFreind(string) : void
Getter and Setters