Group 6: Library Management System

In the project, we have implemented a basic library management system using Linked Lists. It includes an interactive user interface (implemented using a while loop.

Functions used, and their complexity:

|  |  |  |
| --- | --- | --- |
| ***Class*** | ***Function Name*** | ***Complexity*** |
| LinkedList | insert\_data(self, data) | O(1) |
| LinkedList | delete(self, title) | O(n) [n is number of books in book list] |
| LinkedList | display(self) | O(n) [n is number of books in book list] |
| Person | get\_user\_id(self) | O(1) |
| Person | get\_name(self) | O(1) |
| Student | get\_borrowed\_books(self ) | O(1) |
| Student | borrow\_book(self, book, transactions\_list) | O(1) |
| Student | return\_book(self, book, transactions\_list) | O(n) [n is number of books in borrowed books list, since remove function is O(n)] |
| Librarian | add\_book(self, book\_list, book) | O(1) |
| Librarian | remove\_book(self, book\_list, book\_title) | O(n) [n is number of books in book list] |
| Book, Hardcopy, EBook | book\_type(self) | O(1) |
| Book | is\_available(self) | O(1) |
| Book | set\_availability(self, value): | O(1) |
| Book | get\_title(self) | O(1) |
| Book | get\_author(self) | O(1) |
| LMS | check\_id(self, uid) | O(n) [n is number of users in users list] |
| LMS | add\_person(self, uid) | O(1) |
| LMS | make\_book(self, typ) | O(1) |
| LMS | student\_borrow(self, user, book\_title) | O(n) [n is number of books in book list] |
| LMS | student\_return(self, user, book\_title) | O(n) [n is number of books in borrowed book list; default max 3] |
| LMS | borrows(self, user) | O(n) [n is number of books in borrowed book list; default max 3] |

Team members:

#### **K. V. Meghana – cb.sc.u4aie24022**

#### **K Supriya – cb.sc.u4aie24025**

#### **K Sivakumar – cb.sc.u4aie24027**

#### **M Mahendra – cb.sc.u4aie24033**