**LEARNING MANAGEMENT SYSTEM**

****

**ALI NADIR 20K0325**

**IMTIAZ ALI 20K0313**

**AHMED ABDULLAH 20K0470**

**Computer Organization and Assembly Language**

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES – FAST**

**ABSTRACT:**

The aim of the project is to enable students in university to effectively manage their academic records, which would primarily consist of records of past assignments, their mid terms and their finals. The application would allow them to see a breakdown of their results for every subject chosen and also will calculate their grade for the chosen subjects. Additionally, students can set and view their own personalized timetable.

**INTRODUCTION:**

The application allows a student to view and manage academic records for a particular term. It will enable the user to select a certain number of subjects they are currently pursuing, manage academic records for those subjects, and also generate fee challan and transcript for those subjects. Also allows for setting and viewing timetable.

**LITERATURE REVIEW:**

LMSs play an important role in enhancing and facilitating teaching and learning. LMSs not only enable the delivery of instructions and electronic resources to improve and augment student learning in a collaborative environment, but also allow instructors to focus on designing meaningful pedagogical activities (Kattoua, Al-Lozi, and Alrowwad [2016](https://www.tandfonline.com/doi/full/10.1080/1743727X.2020.1737002)). It should be noted that most LMS software available relies on online connectivity, and for good reason as well. But this should not negate the fact that an offline, almost instantaneous solution to every students’ academic record keeping should exist, and this project is driven primarily by the motivation. LMS software such as those based on Sakai, such as SLATE and FLEX offer students to create their own timetables and study plans which is why this application models the same.

**PROBLEM DEFINITION:**

Set, manage, view, and extract academic records as efficiently and intuitively as possible, and allow timetable functionality.

**METHODOLOGY:**

Procedures have been greatly used to reduce structural complexity of the program and to enhance readability. Due to the scope of the language used, imperative programming approach has been adopted. Since user input is key to the application, data validation has been implemented in as many areas as possible. Conditional statements, loop structures and conditional breaks play an integral part in the functionality of the program. Nested function calls have been used. Conditional jumps have been implemented as well. Since most of the loops used in the code exceed the maximum memory that can be allocated, flag jumps have been used as well. In addition, the program is heavily reliant on local variables in most of the procedures used.

**DETAILED DESIGN AND ARCHITECTURE:**

Program is divided into various subsystems or routines that drive the program. The flow is as follows:

* Login with valid student ID
* Once, redirect to function that sets your marks
* Once validated marks successfully registered in system, display menu
* Menu consists of a total of 8 procedures that is continuously displayed and clears the screen.
* Procedures within Menu procedure:
  + Display subjects
  + Set Marks
  + View academic records
  + Generate fee challan
  + View transcript
  + Set timetable
  + View timetable
  + Logout
* These functions contain further subroutines such as grade subject, calculate total, display grade, check marks etc. which are utilized in nested function calls by the menu procedures

**IMPLEMENTATION, TESTING, AND PROGRAMMING CODE:**

**RESULTS/SOFTWARE SIMULATION AND DISCUSSION:**

**CONCLUSION, COST, AND FUTURE WORK:**

The solution solves the problem of computational complexity in that it solely uses runtime stack and memory for the storage, extraction and manipulation of record data without straining secondary storage with this data, as filing has not been used. File processing will cause unnecessarily slow read and write speeds, thus defeating the ultimate purpose of the application. Since users can readily view and pull the records they want to view individually instead of sifting through endless piles of multiple records of any number of subjects, the user friendliness and efficiency of the code can be asserted as well, as tabular data was observed to be less resourceful as individual record data of choosing. Also, the code relies heavily on the use of the runtime stack to manage memory as efficiently as possible throughout runtime. In the future, the system should be improved to allow multiple academic terms to be handled with the same efficiency as described before. GUI interface implementation should be focused on in the future as well to improve the UI/UX aspects of the application as well.

**REFERENCES:**

[What is an LMS? Definition, Features, and Use Cases (ispringsolutions.com)](https://www.ispringsolutions.com/blog/what-is-lms)

[Full article: Learning management systems: a review of the research methodology literature in Australia and China (tandfonline.com)](https://www.tandfonline.com/doi/full/10.1080/1743727X.2020.1737002)

[Full article: Learning management systems: a review of the research methodology literature in Australia and China (tandfonline.com)](https://www.tandfonline.com/doi/full/10.1080/1743727X.2020.1737002)