Examination Management System

INTRODUCTION

DESCRIPTION

An examination management system is a software that's been developed to manage the entire exam process. It covers all the activities that are related to examination management, right from receiving enrollment and examination forms, to the processing of exams, printing, distribution, and statistical reports.

OBJECTIVE

This project explains about examination management system which includes the below features

- Add/Delete the Details of the Students
- Check Eligible Students for Exams
- monitoring attendence for the students
- Printing the records of the students
- Set/Edit Eligibility criteria for exams

SPACE AND TIME COMPLEXITY :

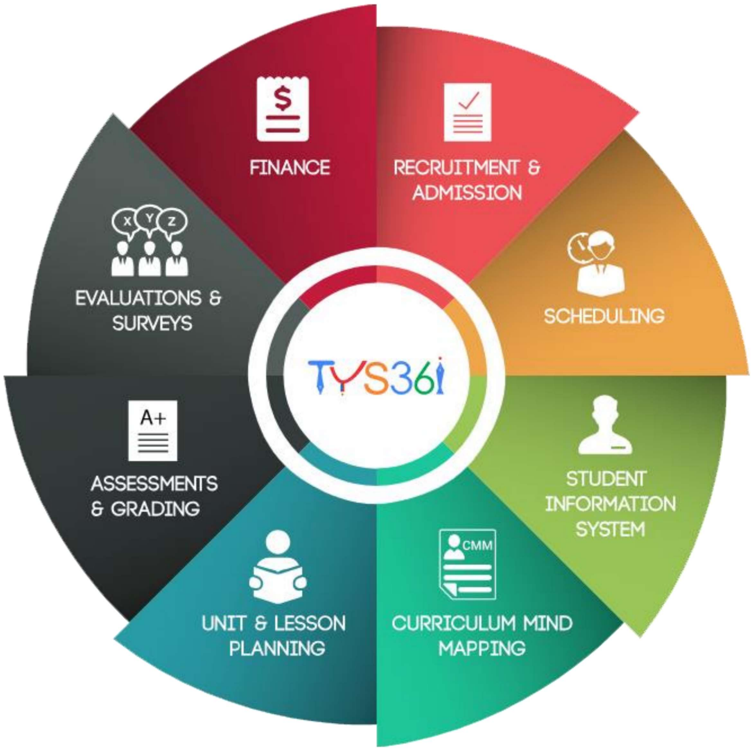
TIME COMPLEXITY :

O(1) time complexity that requires O(n) space complexity. it is simpler to write, and debug and more importantly, there are no loops!

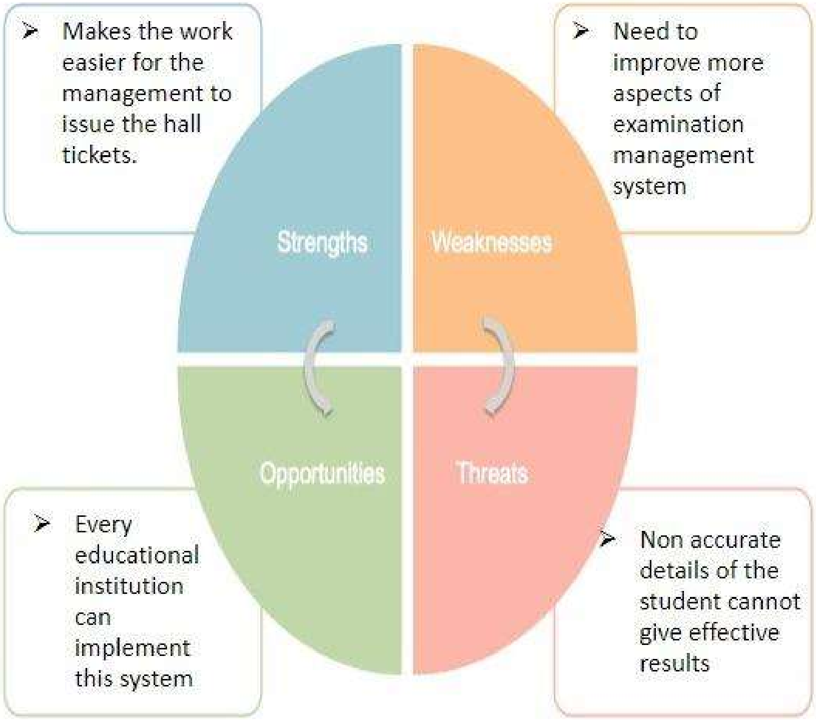
SPACE COMPLEXITY :

O(n) in space complexity.

SYSTEM DESIGN



SWOT-ANALYSIS



- Makes the work easier for the management to issue the hall tickets.

- Need to improve more aspects of examination management system

- Every educational institution can implement this system

- Non accurate details of the student cannot give effective results

4W's and 1'H

WHO :

- Educational Institutions can use this system.

WHAT :

- Examination Management System can be used to evaluate the eligibility of the students for the exam.

WHEN :

- Examination Management System can be used at the time of examination.

WHERE :

- Every Educational Institution can use this system in their office to evaluate the results.

HOW :

- They can download this software into the system and use it for evaluation.

HIGH LEVEL REQUIREMENTS

ID	Description	Status
HR01	Number of working days should be mentioned	Implemented
HR02	Number of students need to mentioned	Implemented
HR03	Details of the student should be mentioned	Implemented
HR04	Fee status of the students should be mentioned	Implemented
HR05	Attendence of the student should be calculated	Implemented
HR06	Deleting the student record option should be available	Implemented
HR07	Data should not be lost in case of faliure	Implemented

LOW LEVEL REQUIREMENTS

ID	Description	Status
LR01	Hall ticket should be generated on sucessful eligibility of the candidate	Future
LR02	Date of examination should be available once the hall tickets are generated	Future
LR03	Amount of Additional attendence should be displayed to get the eligibility for examination	Future
LR04	Amount of fees to be paid should be shown	Future

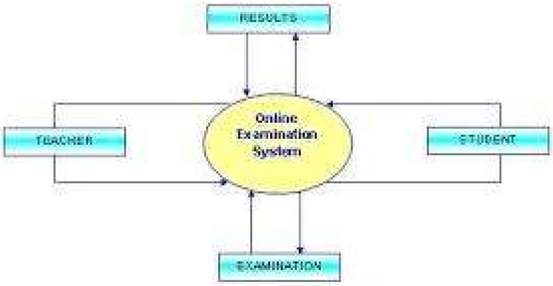
Best Methods To Be Followed

- Used functions to decrease dependency on main function
- Used structures and arrays to accept the inputs from user and store the values which helped in creating easy design of Employee management system.
- Printf statements have been placed only wherever necessary to avoid confusions
- Created header file so that the fuctions can be used else where ever required without any difficulty
- Unit testing is done to avoid any computational errors.

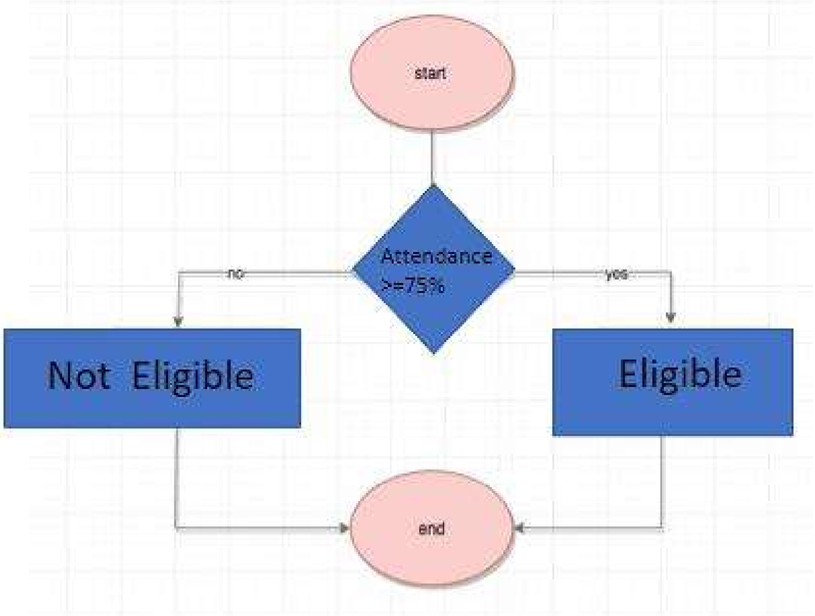
Architecture

USE CASE DIAGRAM

LOW LEVEL DESIGN



FLOWCHART OF EXAMINATION ELIGIBILTY SYSTEM



Here are the below steps to run the code

- For Running and Building the code

  - Type make run in terminal
- For Running the Building tests
  - Type make run_test in terminal
- For static analysis
  - Type make static_analysis in terminal
- For dynamic analysis
  - Type make dynamic_analysis in terminal
- For code coverage
  - Type make coverage in terminal
- For cleaning
  - Type make clean in terminal

Usage of Pointers

Pointers:

- The Pointer in C, is a variable that stores address of another variable.
- A pointer can also be used to refer to another pointer function.
- A pointer can be incremented/decremented, i.e., to point to the next/ previous memory location. The purpose of pointer is to save memory space and achieve faster execution time. Like variables, pointers in C programming have to be declared before they can be used in your program. Pointers can be named anything you want as long as they obey C's naming rules. A pointer declaration has the following form.

syntax: data_type * pointer_variable_name;

- Types of Pointers in C: 1.Null Pointer 2.Void Pointer 3.Wild pointer 4.Dangling pointer

Usage of Preprocessor :

- The C preprocessor is a macro processor that is used automatically by the C compiler to transform your program before actual compilation. It is called a macro processor because it allows you to define macros, which are brief abbreviations for longer constructs.

*The C preprocessor provides four separate facilities that you can use as you see fit:

- Inclusion of header files. These are files of declarations that can be substituted into your program.

Function Pointer

- In the C function pointer is used to resolve the run time-binding.
- A function pointer is a pointer that stores the address of the function and invokes the function whenever required.
- In C, we can use function pointers to avoid code redundancy.

Struct

- A structure is a key word that create user defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type.
- 'struct' keyword is used to create a structure.

- A structure variable can either be declared with structure declaration or as a separate declaration like basic types.
- Structure members cannot be initialized with declaration.

Typedef

- typedef, which we can use to give a type a new name. Following is an example to define a term BYTE for one-byte numbers.After this type definition, the identifier BYTE can be used as an abbreviation for the type unsigned char
- Syntax: typedef data_type new_name

TEST PLAN

HIGH LEVEL TEST PLAN

ID	Description	Expected Input	Expected Output	Actual Output	Type of Test
HR01	Number of working days should be mentioned	Integer	SUCCESS	SUCCESS	Technical Based
HR02	Number of students need to mentioned	Integer	SUCCESS	SUCCESS	Technical Based
HR03	Details of the student should be mentioned	char	SUCCESS	SUCCESS	Scenario Based
HR04	Fee status of the students should be mentioned	char	SUCCESS	SUCCESS	Technical Based

LOW LEVEL TEST PLAN

ID	Description	Expected Input	Expected Output	Actual Output	Type of Test
LR01	Attendance of the student should be calculated	Integer	SUCCESS	SUCCESS	Technical Based
LR02	Deleting the student record option should be available	char	SUCCESS	SUCCESS	Scenario Based
LR03	Data should not be lost in case of failure	char	SUCCESS	SUCCESS	Scenario Based