

**EXAM CENTRE ASSIGNMENT APPLICATION**

**Software Requirement Specification (SRS) Document**

**Sprint 1 Implementation**

**Project Timeline: 02.10.2022 to 10.10.2022**

**Index**

1. Introduction
   1. Intended Audience
   2. Project Purpose
   3. Key Project Objective
   4. Project scope
2. Design Overview
   1. Design Objective
   2. Design Alternative
   3. User Interface Paradigms
   4. Validation
3. System Features and Requirements
   1. System Features
   2. System Requirements
   3. System Features
4. Data Flow Diagrams
   1. Level 0 DFD
   2. Level 1 DFD
5. **INTRODUCTION**

The exam centre application is a system that allows users to automatically assign exam centre to a list of candidates stored in multiple files based upon the exam they are appearing for and also assigns room number and seat number to the candidates in their respected exam centre. The exam centre details are stored in separate file. After assigning the exam centre to all the individual candidates, the candidate’s details along with details assigned exam centre are stored in separate files for each exam and exam centres. An option is provided to print the hall tickets where all the details can be obtained by entering the Candidate id.

* 1. **Intended Audience**

The target audience set for this project can be any authority that is willing to conduct various examinations and to store the candidate’s data in efficient manner so it can be easily accessed.

* 1. **Project Purpose**

The exam centre assignment project can help us in understanding the basic concepts of functions and file handling using system calls. Dynamic memory allocation and Structure data types is used mostly for efficient use of memory spaces. Once the splitting up of candidates is done in separate files based upon their exam centre, user can view their hall ticket using their candidate id. Error message is displayed in case if invalid candidate id is entered by the user.

Separate threads are created for reading individual input candidate list files and proper mutex locks are created and released for writing into separate exam centre files and for writing invalid candidates details in one separate file.

* 1. **Key Project Objective**

1. Allow the user to input multiple candidate list files through command line.
2. Separate candidates based upon their exams
3. Assign Exam centre, room and seat number to each candidate.
4. Allow candidates to view their hall ticket using their candidate id.
5. Print error message in case of invalid candidate id entered by the user.
   1. **Project Scope**

The project aims to develop an application to automatically assign exam centre to a list of candidates where the details of candidates are stored in text files in a very specified manner and accordingly the project code is written.

1. **Design Overview**

Exam centre assignment project comprises of only one main source file under which separate function calls and library functions are used.

* 1. **Design Objective**

1. Pass multiple candidate list files through command line.
2. Create and populate structure for each candidate from all the input files using one thread created for each input file.
3. Create and populate the structure for exam centre details taking input from exam centre text file.
4. Exam id present in both candidate and exam structure acts as the link between both the structure.
5. Use pointers for referring and mapping candidate from candidate list structure to their respective exam centre structure.
6. Write candidate details along with their exam centre details in the respected exam centre files.
7. Display hall ticket using candidate id.
   1. **Design Alternative**

We have used linked list data structure for storing Candidates data i.e. Candidates name, Candidates exam id, candidate’s id, start and end date of exam and similarly for storing exam centre details i.e. Exam id, Exam centre name and address of exam centre.

* 1. **User Interface Paradigms**

The exam centre assignment project gives an option to view the hall ticket containing all details of the candidate along with storing the list of students for various exams separately in text files.

* 1. **Validation**
* Candidate list files is to be passed through command line while executing the application and exact file path has to be provided.
* In case of invalid candidate id entered while displaying hall ticket, error message is displayed.
* The candidate list files and exam centre list files should have the details within them strictly separated by a **comma (,)** .

1. **SYSTEM FEATURES AND REQUIREMENTS**
   1. **SYSTEM FUNCTIONS**
2. Exam centre file is opened and the data is allocated to dynamically created memory using Linked list structure.
3. Candidate list files are opened using separate threads and data is again allocated in dynamically allocated memory using Linked list structure and this structure is linked to exam centre structure using a struct pointer variable.
4. Invalid candidates’ details are written in Invaid\_candidate\_list.txt files and for this purpose mutex are used to avoid deadlock.
5. Valid candidate data are written to separate files based on the exam they are appearing for.
6. After separating candidates based on exams, user is asked to enter candidate id to display.
7. If he enters valid candidate id from the candidate list, all the details are displayed else error message is displayed.
   1. **SYSTEM REQUIREMENTS** 
      1. **TOOLS TO BE USED**

* Pthread Library
* C File Handling
* C Language
* System Programming
  1. **SYSTEM FEATURES**

### Supportability: The system is easy to use.

* Design Constraints: The system is built using only C language.
* Usability:The Exam centre assignment application can be used to replace the old means. Assigning Exam centres to the candidates just by entering the path of file with candidate list makes it much easier than individually one by one though the details of each student manually.
* Reliability & Availability**:** The system is available 24/7 that is whenever the user would like to use the system, they can use it up to its functionalities.
* Performance: The system will work on the user’s terminal**.**

1. **DATA FLOW DIAGRAMS :**

**4.1 LEVEL 0 DFD :**

HALL TICKET (DETAILS ABOUT CANDIDATE ID ,EXAM ID EXAM CENTRE NAME AND ADDRESS,EXAM START AND END DATE )

CANDIDATE ID

CANDIDATE

CANDIDATE ID

CANDIDATE

Fig 4.1

**4.2 LEVEL 1 DFD:**

CANDIDATE ID

CANDIDATES LIST FILES THROUGH COMMAND LINE

INVALID CANDIDATES

VALID CANDIDATES

INVALID CANDIDATE LIST FILE

HALL TICKET WITH ALL THE INFORMATION

USER

SEPARATE CANDIDATE LIST FILES FOR EACH EXAM CENTRES

USER

Fig 4.2