EXAM CENTER ALLOTMENT SYSTEM

High Level Design & Low Level Design

The purpose of this document is to provide a template for documenting both HLD & LLD.

**Document Control:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 07/01/2023 | HLD\_LLD DESIGN DOCUMENT-Version 0.1 | M. Bindhu | Design | | | | Prasanth | |
| 07/01/2023 | HLD\_LLD DESIGN DOCUMENT-Version 0.2 | B. Sai sireesha | SRS briefing | | | | Prasanth | |
| 14/01/2023 | HLD\_LLD DESIGN DOCUMENT-Version 0.3 | Anusha Challapureddi | Coding | | | | Prasanth | |
|  | HLD\_LLD DESIGN DOCUMENT-Version 0.4 | Siraparapu Rupika sri | Testcases | | | | Prasanth | |
| 14/01/2023 | HLD\_LLD DESIGN DOCUMENT-Version 0.5 | Talabattula Sai Santhoshini Reethika | Coding | | | | Prasanth | |

[**1. Introduction**](#_heading=h.3znysh7) **5**

[1.1. Intended Audience](#_heading=h.2et92p0) 5

[1.2. Acronyms/Abbreviations](#_heading=h.tyjcwt) 5

[1.3. Project Purpose](#_heading=h.3dy6vkm) 5

[1.4. Key Project Objectives](#_heading=h.1t3h5sf) 5

[1.5. Project Scope and Limitation](#_heading=h.4d34og8) 5

[*1.5.1. In Scope*](#_heading=h.2s8eyo1) 6

[*1.5.2. Out of scope*](#_heading=h.17dp8vu) 6

[1.6. Functional Overview](#_heading=h.3rdcrjn) 6

[1.7. Assumptions, Dependencies & Constraints](#_heading=h.26in1rg) 6

[1.8. Risks](#_heading=h.lnxbz9) 6

[**2. Design Overview**](#_heading=h.35nkun2) **7**

2[.1 Design Objectives](#_heading=h.1ksv4uv) 8

[*2.1.1. Recommended Architecture*](#_heading=h.44sinio) 8

[2.2. Architectural Strategies](#_heading=h.2jxsxqh) 8

[*2.2.1. Design Alternative*](#_heading=h.z337ya) 8

[*2.2.2. Reuse of Existing Common Services/Utilities*](#_heading=h.3j2qqm3) 8

[*2.2.3. Creation of New Common Services/Utilities*](#_heading=h.1y810tw) 9

[*2.2.4. User Interface Paradigms*](#_heading=h.4i7ojhp) 9

[*2.2.5. System Interface Paradigms*](#_heading=h.2xcytpi) 9

[*2.2.6. Error Detection / Exceptional Handling*](#_heading=h.1ci93xb) 9

[*2.2.7. Memory Management*](#_heading=h.3whwml4) 9

[*2.2.8. Performance*](#_heading=h.2bn6wsx) 9

[*2.2.9. Security*](#_heading=h.qsh70q) 9

[*2.2.10. Concurrency and Synchronization*](#_heading=h.3as4poj) 9

[*2.2.11. Housekeeping and Maintenance*](#_heading=h.1pxezwc) 9

[**3. System Architecture**](#_heading=h.49x2ik5) **10**

[3.1. System Architecture Diagram. (Not Necessary)](#_heading=h.2p2csry) 11

[3.2. Flowchart](#_heading=h.147n2zr) 12

[3.3. Subsystem Architecture](#_heading=h.3o7alnk) 13

[3.4. System Interfaces](#_heading=h.23ckvvd) 14

[*3.4.1. Internal Interfaces*](#_heading=h.ihv636) 14

[*3.4.2. External Interfaces*](#_heading=h.32hioqz) 14

[**4. Detailed System Design**](#_heading=h.1hmsyys) **14**

[4.1. Key Entities](#_heading=h.41mghml) 14

[4.2. Detailed-Level Database Design](#_heading=h.2grqrue) 14

[*4.2.1. Data Mapping Information 1*](#_heading=h.vx1227)4

[*4.2.2. Data Conversion 1*](#_heading=h.3fwokq0)4

[4.3. Archival and retention requirements](#_heading=h.1v1yuxt) 14

[4.4. Disaster and Failure Recovery](#_heading=h.4f1mdlm) 15

[4.5. Business Process workflow](#_heading=h.2u6wntf) 15

[4.6. Business Process Modeling and Management (as applicable)](#_heading=h.19c6y18) 15

[4.7. Business Logic](#_heading=h.3tbugp1) 15

[4.8. Variables](#_heading=h.28h4qwu) 15

[4.9. Activity / Class Diagrams (as applicable)](#_heading=h.nmf14n) 15

[4.10. Data Migration](#_heading=h.37m2jsg) 15

[*4.10.1. Architectural Representation*](#_heading=h.1mrcu09) 15

[*4.10.2. Architectural Goals and Constraints*](#_heading=h.46r0co2) 15

[*4.10.3. Logical View*](#_heading=h.2lwamvv) 15

[*4.10.4. Architecturally Significant Design Packages*](#_heading=h.111kx3o) 15

[*4.10.5. Data model*](#_heading=h.3l18frh) 15

[*4.10.6. Deployment View*](#_heading=h.1egqt2p) 16

[**5. Environment Description**](#_heading=h.3ygebqi) **16**

[5.1. Time Zone Support](#_heading=h.2dlolyb) 16

[5.2. Language Support](#_heading=h.sqyw64) 16

[5.3. User Desktop Requirements](#_heading=h.3cqmetx) 16

[5.4. Server-Side Requirements](#_heading=h.1rvwp1q) 16

[*5.4.1. Deployment Considerations*](#_heading=h.4bvk7pj) 16

[*5.4.2. Application Server Disk Space*](#_heading=h.2r0uhxc) 16

[*5.4.3. Database Server Disk Space*](#_heading=h.1664s55) 17

[*5.4.4. Integration Requirements*](#_heading=h.3q5sasy) 17

[*5.4.5. Jobs*](#_heading=h.25b2l0r) 17

[*5.4.6. Network*](#_heading=h.kgcv8k) 17

[*5.4.7. Others*](#_heading=h.34g0dwd) 17

[5.5. Configuration](#_heading=h.1jlao46) 17

[*5.5.1. Operating System*](#_heading=h.43ky6rz) 17

[*5.5.2. Database*](#_heading=h.2iq8gzs) 17

[*5.5.3. Network*](#_heading=h.xvir7l) 17

[*5.5.4. Desktop*](#_heading=h.3hv69ve) 17

[**6. References**](#_heading=h.1x0gk37) **17**

# 

# 1. Introduction

* Exam Centre Allotment System is an online process developed for colleges to simplify examination hall allotment.
* Mostly students are facing many problems for finding the exam hall. A newly invented concept can aid for the students for checking their exam halls.
* Admin have information about all the students who attend the examination.
* It contains the name of the student, ID of student, exam name, exam center details
* This project keeps track of various details in modules such as Candidate Details, Exam Details, and Exam center Details with the proper descriptions..

## 1.1 Intended Audience

|  |  |
| --- | --- |
| BU Authority |  |
|  |  |

## 1.2 Acronyms/Abbreviations

|  |  |
| --- | --- |
| UT | Unit Test |
| IT | Integrated Test |
|  |  |
|  |  |

## 1.3 Project Purpose

The purpose of this project is Exam Centre Allotment based on the candidate exam.The procedure of assigning candidates to a certain exam centre is known as exam centre allotment. Exam centres are assigned to candidates based on their candidate and exam IDs. This information is stored in text files.

## 1.4 Key Project Objectives

To schedule without interference between exams by maintaining time management.

**1.5 Project Scope and Limitation**

* Scheduling without interference and making all exams done for given candidates without leaving any candidates given in list of file. There will be some interference if candidate gets allocated two exam centers.

### 1.5.1 In Scope

This Project allocates exam centers to candidates based on the candidate id, exam id.

### 1.5.2 Out of scope

Required to involve techniques such as concurrency, inheritance to make the project done completely.

## 1.6 Functional Overview

CPP STL enables to code the job specifications, FileIO operations helps to read file from Command line arguments and write schedule into different text files. Valgrind captures the data of memory leak.

## 1.7 Assumptions, Dependencies & Constraints

Schedule for different exams is done by assuming the candidate info given in the candidate file is of correct format. Modified application will remove the exams that are not in correct format.

## 1.8 Risks

All assumptions, functional overview and design parameters are documented without evaluation which are to be implemented without missing.

# 2.Design Overview

START:

This is the start block which indicates the start of the program.

1.CANDIDATE:

Candidate has to login by providing the required credentials.Candidates has to give the valid details of ID, name, exam to be appeared.

2.LOGIN:

Candidate has to login by providing the required credentials and get apply for the exam i.e, UPSC, NEET.

3.ADMIN:

Admin will generate the hall ticket and allot the exam centers according to the requirement.

4.EXAM CENTER:

After assignments are done admin display total number of candidates and remaining capacity of exam center.

5.Invalid candidates entires from all input files should be written to invalid Candidates.txt.file.

6.Candidates appear for the exam based on the allotment.

## 2.1 Design Objectives

Design objectives include different parameters like exams given to candidate and scheduling corresponding exam centers and providing interference less design while assigning exam centers to candidates.

**2.1.1 Recommended Architecture**

Flow chart

## 2.2 Architectural Strategies

No architectural strategies have been employed.

### 2.2.1 Design Alternative

Designed flowchart as design paradigm used in application.

### 2.2.2 Reuse of Existing Common Services/Utilities

Design and development is done from scratch using existing sources star UML for design and VI for development.

**2.2.3 Creation of New Common Services/Utilities**

Used existing resources to develop the application specific services.

### 2.2.4 User Interface Paradigms

NA

### 2.2.5 System Interface Paradigms

NA

### 2.2.6 Error Detection / Exceptional Handling

The files that are sent through command line argument are checked for exception and catches exception if it occurs and at the conditions where the source code may terminate gets checked for exceptions by placing that code in try block if any exceptions it cached by catch block.

### 2.2.7 Memory Management

This application involves Dynamic memory allocation which allows us to use only required memory without reserving extra memory. Valgrind application used to check for memory leaks if any and rectified if any memory leaks occur.

**2.2.8 Performance**

System performance needs to be accurate while scheduling for each machine. Each machine schedule is to be done in different files as given without interference.

**2.2.9 Security**

The source code is available only in read only mode for others who are accessing file and final executable file is only available for further modifications.

**2.2.10 Concurrency and Synchronization**

Application is designed to work concurrently without interfering with the function of other part of code and also work simultaneously

### 2.2.11 Housekeeping and Maintenance

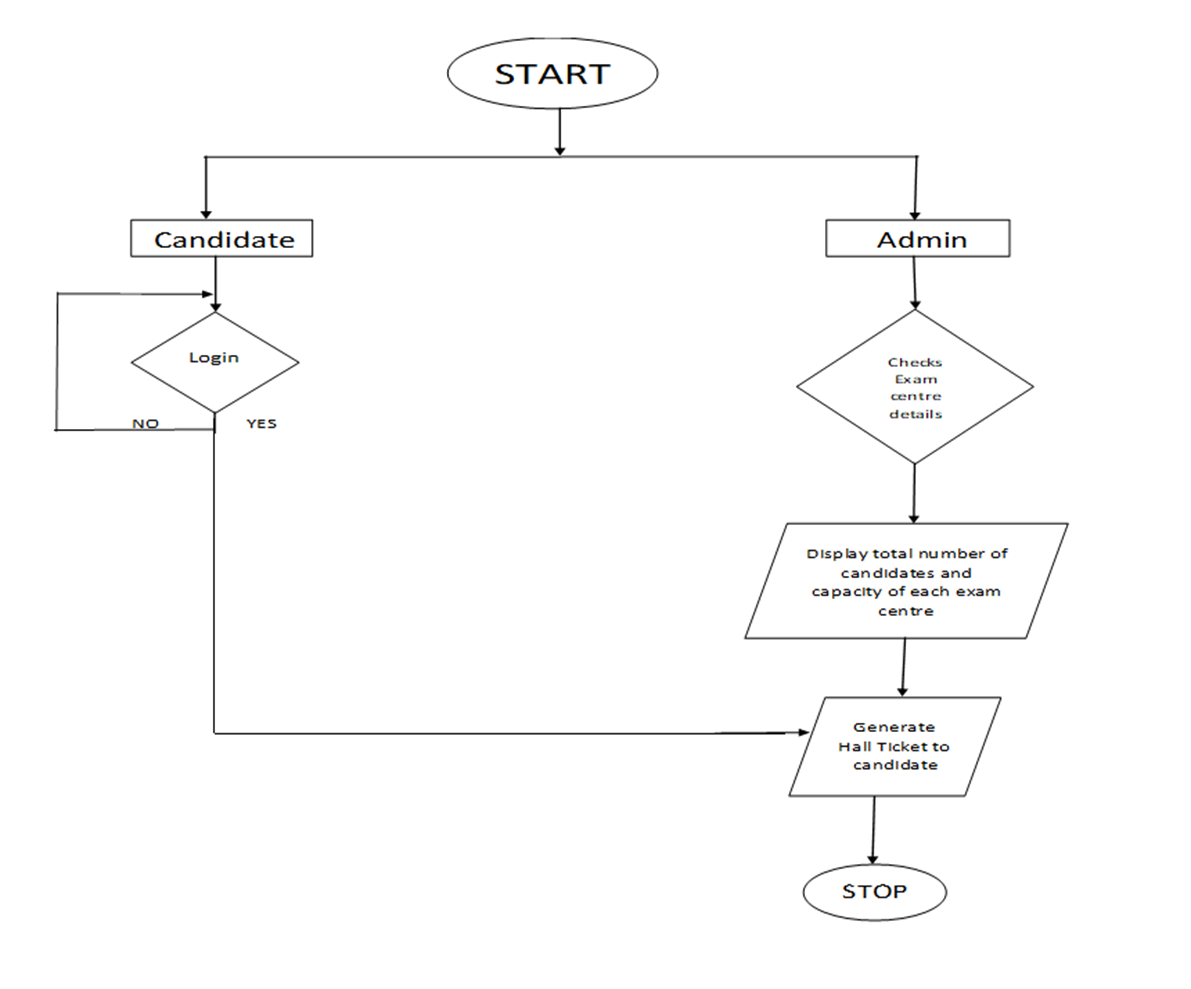
Makefile is cleared first to remove the previous cache and text files are used to manage the database

# 3. System Architecture

# 3.1 System Architecture Diagram

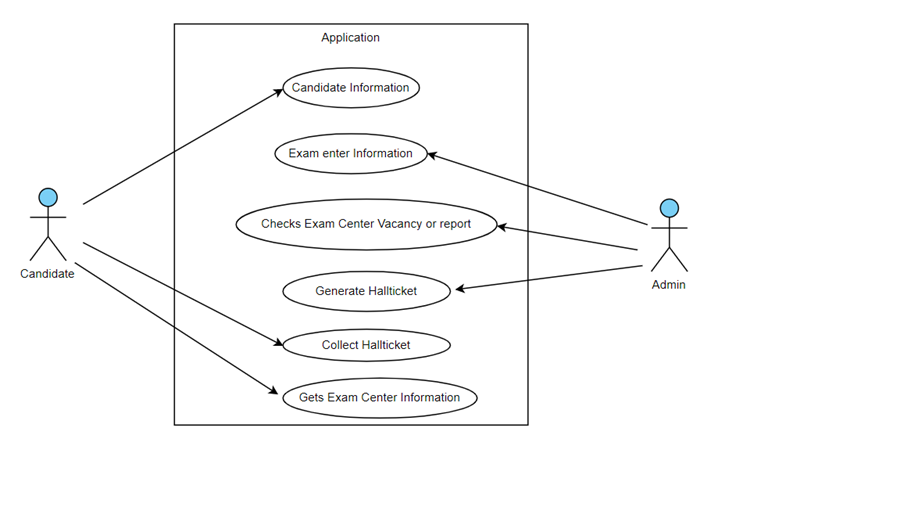
NA

**3.2 FLOWCHART**



## 

**3.2.1.USE CASE**



**3.2.2 SEQUENCE DIAGRAM**



## 3.3 Subsystem Architecture

NA

## 3.4 System Interfaces

NA

### 3.4.1 Internal Interfaces

NA

### 3.4.2 External Interfaces

NA

# Detailed System Design

NA

## Key Entities

* We use key entities as valgrind and gdb.

## 4.2 Detailed-Level Database Design

NA

### 4.2.1 Data Mapping Information

NA

### 4.2.2 Data Conversion

NA

## 4.3 Archival and retention requirements

NA

## 4.4 Disaster and Failure Recovery

NA

## 4.5 Business Process workflow

NA

## 4.6Business Process Modeling and Management (as applicable)

NA

## 4.7 Business Logic

NA

## 4.8 Variables

NA

## 4.9 Activity / Class Diagrams (as applicable)

NA

**PSEUDOCODE**

1.Candidate details

Set Candidates to zero

Set Exams to zero

Set Exam center to zero

Input the Candidate to apply for exam

integer id

Display "id"

Input id

string name

Display "name"

Input name

if id not equal to zero and equal to 10 integers

print id

else

print invalid id

IF name string not equal to null

Print valid name

End

Else

Print invalid name

End

Function Candidate(arguments: id, name)

Get the id and name

For I =candidatebegin To candidateEnd

output i

Display id and name

While the candidate has not given proper login details

Not to be apply for Exam

If the Candidate enter correct details apply for exam

2.Apply for exam

//exam 1:NEET

Function NEET (argument: Year of passing)

For I = begin To end

output i

EndFor

Display year of passing

//Exam 2:UPSC

Function UPSC (arguments: StartDate, EndDate, State)

For i = UPSC.begin To UPSC.end

Output startDate

Output endDate

EndFor

if setstartDate=getstartDate

print startdate

if setendDate= getendDate

print enddate

Display startdate, enddate

//Exam 3:Eamcet

Function EAMCET(arguments:sub1,sub2,sub3)

For I =EAMCET.begin To EAMCET.end

Output sub1

Output sub2

Output sub3

Display sub1, sub2, sub3

3.Exam centre

FunctionCandidate(arguments: ExamId, CenterName, Address,capacity)

For I = CANDIDATE.begin To CANDIDATE.end

Output ExamId

Output CenterName

Output Address

Output Capacity

END

//Data Validation

Candidate Id Validation Function:

The Roll No length should be 10. If Roll No length is 10, it should return True else false.

1. Declare int Candidate\_Id\_Validation(int Candiate Id)

2. if(Candidate Id.length() == 10)

return true

else

return false

Exam Validation Function:

If the Exam is NEET or EAMCET or UPSC, it should return true else false.

1. Declare char Exam\_Validation(char exam)

2. if(Exam == "NEET" || Exam == "EAMCET" || Exam == "UPSC")

return true

else

return false

**4.10 Data Migration**

NA

### 4.10.1 Architectural Representation

NA

### 4.10.2 Architectural Goals and Constraints

* The project is just for educational purposes.

### 4.10.3 Logical View

### NA

### 4.10.4 Architecturally Significant Design Packages

NA

### 4.10.5 Data model

NA

### 4.10.6 Deployment View

NA

# 5. Environment Description

GCC: In Linux, the GCC stands for GNU Compiler Collection. It is a compiler system for the various programming languages. It is mainly used to compile the C and C++ programs.

UBUNTU: Ubuntu is an open-source operating system (OS) based on the Debian GNU/Linux distribution. Ubuntu incorporates all the features of a Unix OS with an added customizable GUI, which makes it popular in universities and research organizations. Ubuntu is primarily designed to be used on personal computers, although a server edition does also exist.

GITHUB: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests.

## 5.1 Time Zone Support

NA

## 5.2 Language Support

NA

## 5.3 User Desktop Requirements

Linux, Ubuntu

## 5.4 Server-Side Requirements

Linux, Ubuntu

### 5.4.1 Deployment Considerations

NA

### 5.4.2 Application Server Disk Space

NA

### 5.4.3 Database Server Disk Space

NA

### 5.4.4 Integration Requirements

NA

### 5.4.5 Jobs

NA

### 5.4.6 Network

NA

### 5.4.7 Others

NA

## 5.5 Configuration

* Operating system, Processor.

### 5.5.1 Operating System

4GB RAM, Processor Intel (R) Core (TM) I3-7020U CPU @, 64 bit operating system, x 64-based processor.

### 5.5.2 Database

NA

### 5.5.3 Network

Network is a process of assigning network settings, policies, flows and controls. In a virtual network, its easier to make network configuration changes because physical network devices appliances are replaced by software removing the need for extensive manual configuration.

### 5.5.4 Desktop

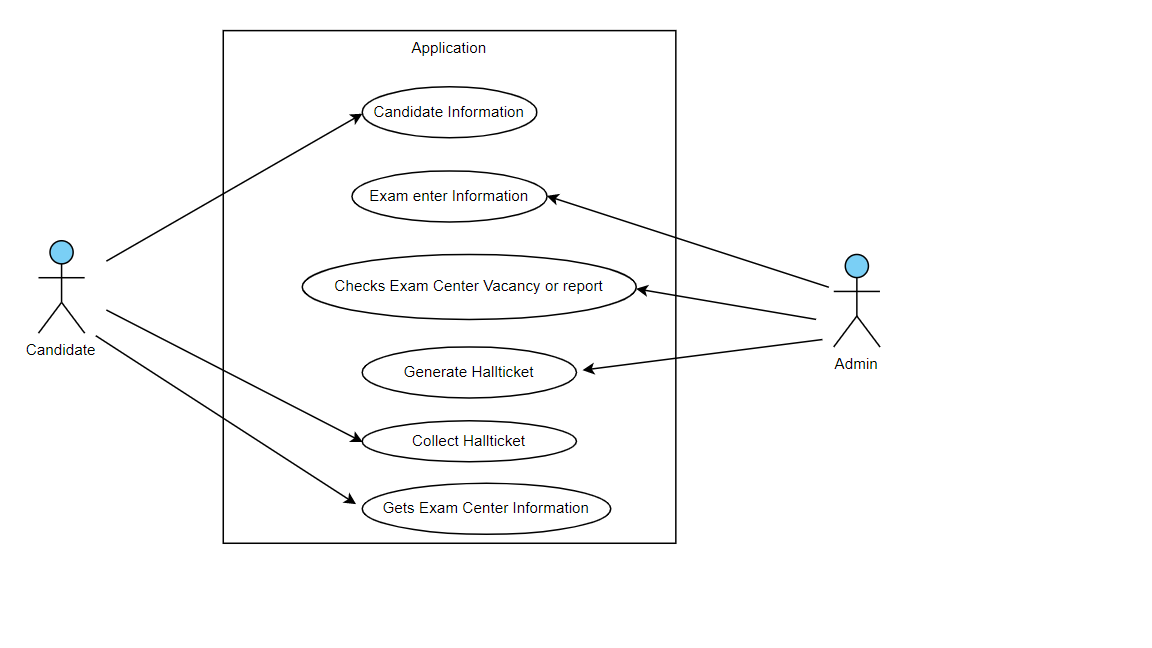
* RAM: 4GB or greater - For optimal performance, 6GB or 8GB
* Linux OS

# 

# 6. References

<https://jpinfotech.org/exam-hall-seating-allotment-system/>

<https://www.geeksforgeeks.org/examination-management-system-in-c/>

**