Capgemini

[redacted]

High Level Design & Low Level Design

Document Control :

Project Revision History

Date	Version	Author	Brief Description of Changes	Approver Signature
26.09.2022	1.0	Group 5		

Index

1. Introduction

The introduction of the High Level Design (HLD) and Low Level Design (LLD) provides an overview of the entire document with purpose, intended audience, scope and key objectives. The purpose of this High Level Design (HLD) Document is to add the necessary detail to the **Model City Vaccination Drive Application's** description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level. This application will manage vaccine registrations and the data involved in the vaccination drive.

1.1. Project Purpose

This online application would track all the new users, schedule vaccination slots for registered users and update all the data related to registered users. Secondly, the application also allows only the authorized city government employees to update the registered user's database, track vaccine units in various centre and view specific required information related to vaccination drive.

1.2. Intended Audience

This document is intended to be read by clients, designers , program and solution teams.
The intended audience for this application is the citizen that want to get vaccinated whereas the application is to be supervised by government personnel.

1.3. Project Scope

The Model City Vaccination drive is specifically developed for the vaccination of the beneficiaries. This application allows the beneficiaries to register for the vaccination drive at any of the 3 centres available and also book their slots. Secondly, the application allows only authorised personnel to delete the fully vaccinated beneficiaries record. Also, he can create a vaccine file containing all the 3 centre codes and number of units of vaccine available across the centres just before the vaccination begins. He can also get the list of all the beneficiaries who got vaccinated today.

1.4. Key Project Objectives

- Allow Beneficiaries to apply for vaccination drive.
- Create account for new beneficiaries.
- Beneficiaries can access the application using aadhar number.
- Maintain the data related to beneficiary and vaccination.
- Schedule slot timing for registered beneficiaries.
- Government personnel are given the role of admin to access the specific functions.
- Only Admin can update or modify the databases present in this application.

2. Design Overview

Model City Vaccination Drive comprises of following modules:

- Menu

Name of the Module	Main Menu
Handled by	Aboli Pawar
Description	This module contains the general user login

Name of the Module	Beneficiary Login
Handled by	Aboli Pawar
Description	This module contains menu restricted for beneficiaries only.

Name of the Module	Check Password
Handled by	Anuja Nikam
Description	This module verifies the password that are restricted for admin role only.

- **User Corner**

Name of the Module	New User Menu
Handled by	Aboli Pawar
Description	Consists menu for users that are new to the application.

Name of the Module	Create Account
Handled by	Nitika Mhatre
Description	This module lets user create account by entering valid details.

Name of the Module	View User Details
Handled by	Aboli Pawar
Description	This module will display the details entered by the user while creating the account.

Name of the Module	Registered User Menu
Handled by	Vaastav Talwar
Description	Consists menu for users that are already existing in database of the application.

Name of the Module	Select Centres & Book Slot
Handled by	Anuja Nikam
Description	Consists of centre names from which user can select centre upon vaccine availability. Lets user book a slot according to specific timing .

Name of the Module	Get Vaccination
Handled by	Vaastav Talwar
Description	This module lets user complete the vaccination process by verifying a token.

- **Admin Corner**

Name of the Module	Admin Menu
Handled by	Vaastav Talwar
Description	This module displays all the functions available to admin role.

Name of the Module	Display Beneficiary Records
Handled by	Nitika Mhatre
Description	This module shows all the beneficiary records.

Name of the Module	Edit & Display Vaccine Units
Handled by	Nitika Mhatre
Description	This module lets admin edit and display the vaccine units available in centres.

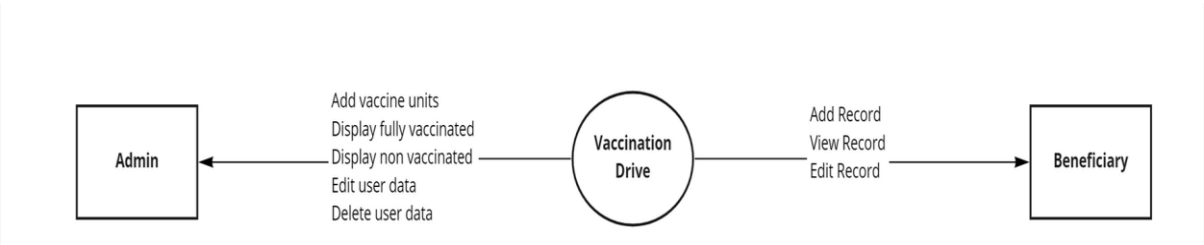
Name of the Module	Display & Delete Fully Vaccinated
Handled by	Vaastav Talwar ,Nitika Mhatre
Description	This module shows the fully vaccinated records and delete them in the database.

Name of the Module	Display Not Vaccinated & Today Vaccinated
Handled by	Anuja Nikam
Description	This module shows the beneficiary records.

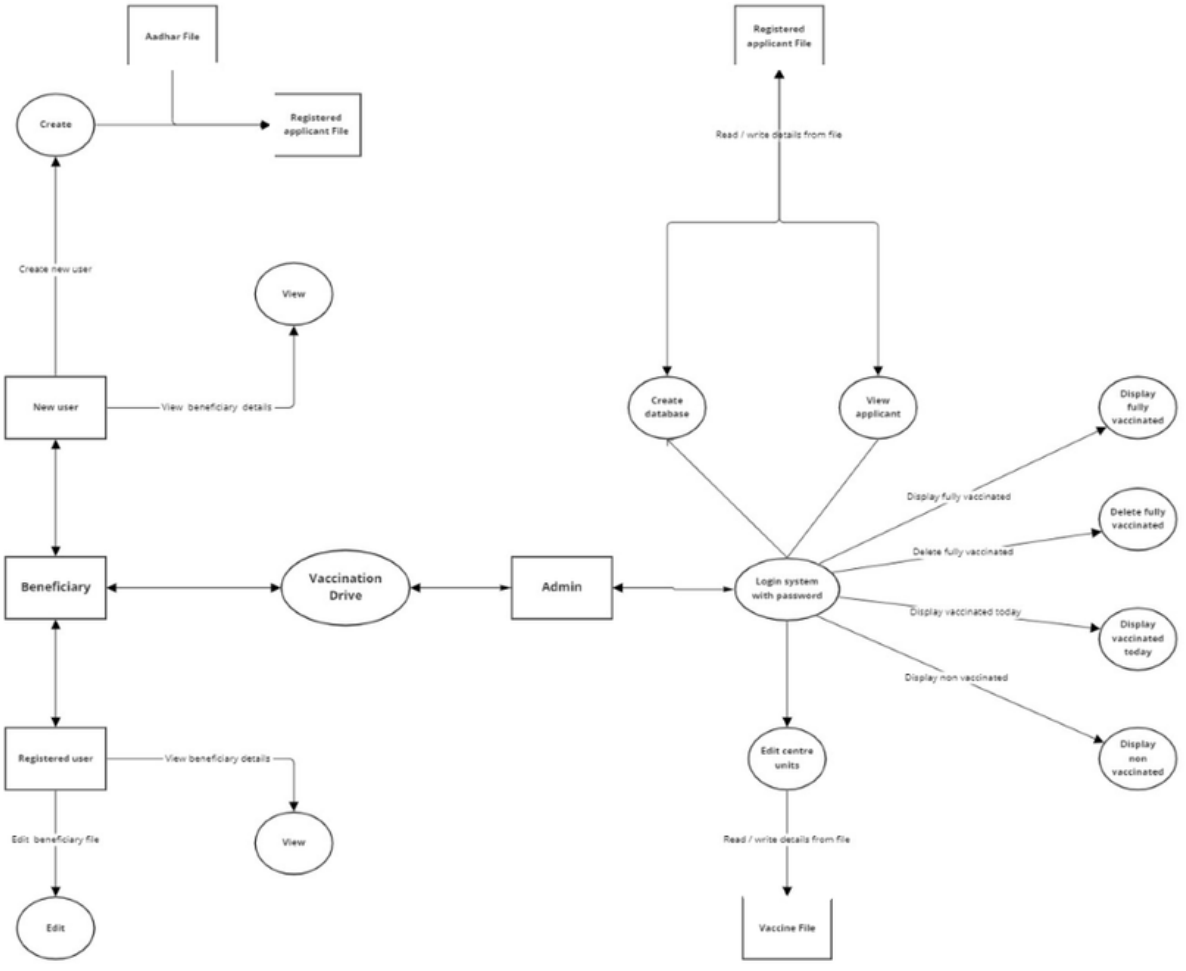
3. Detailed System Design

3.1. Data Flow Diagram

- Level 0

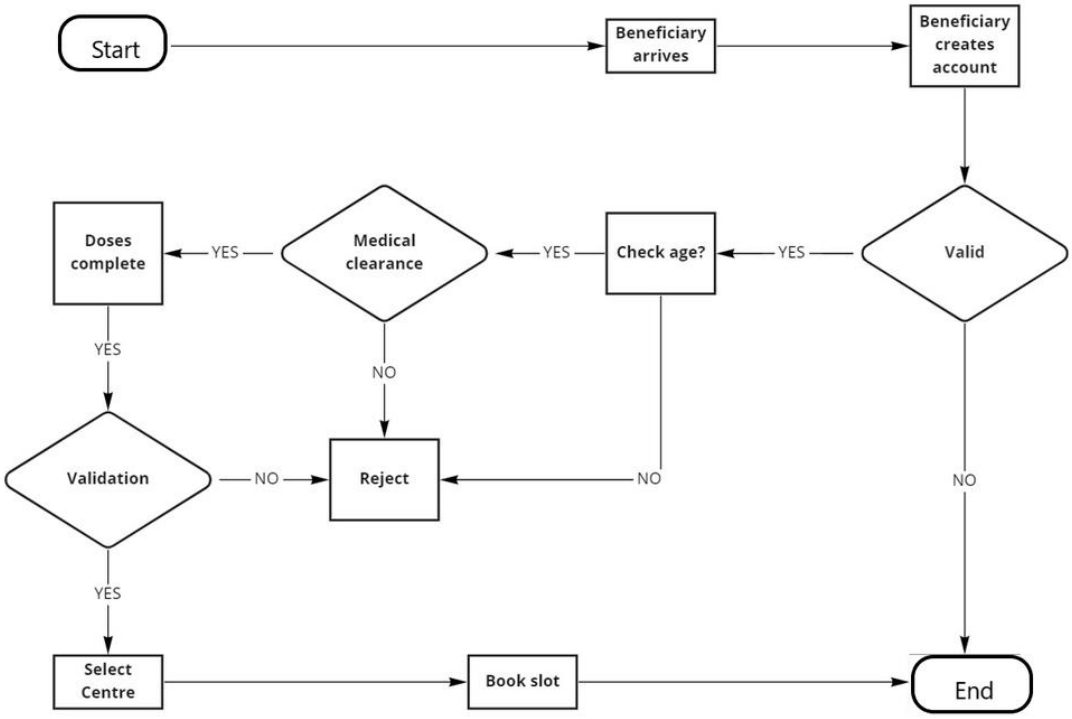


- Level 1

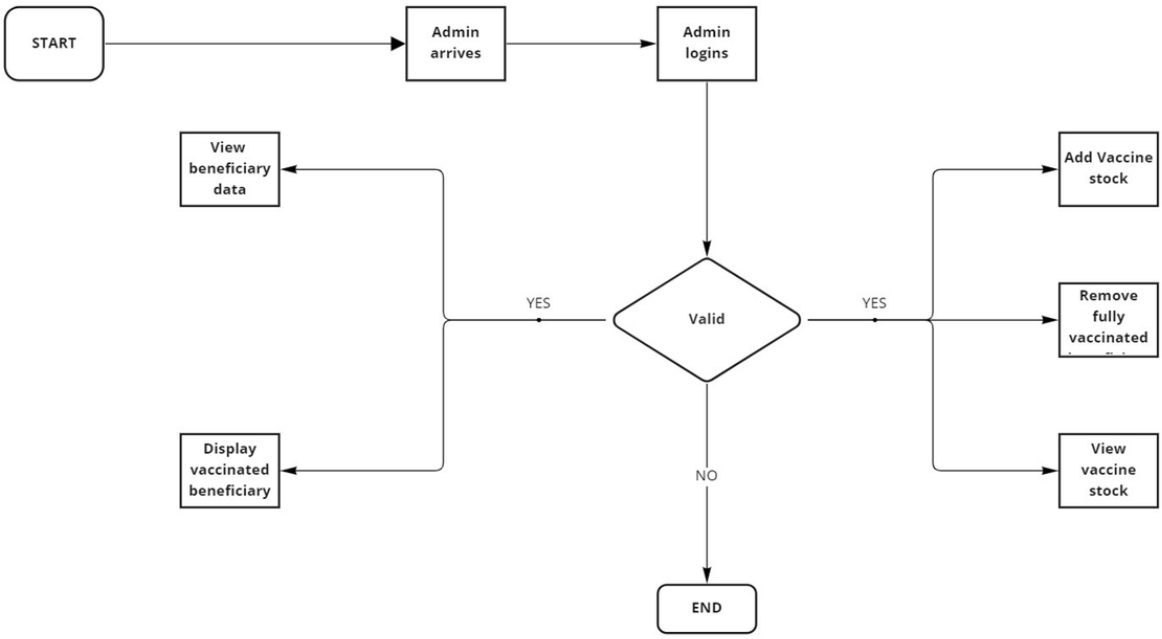


3.2. Flowcharts

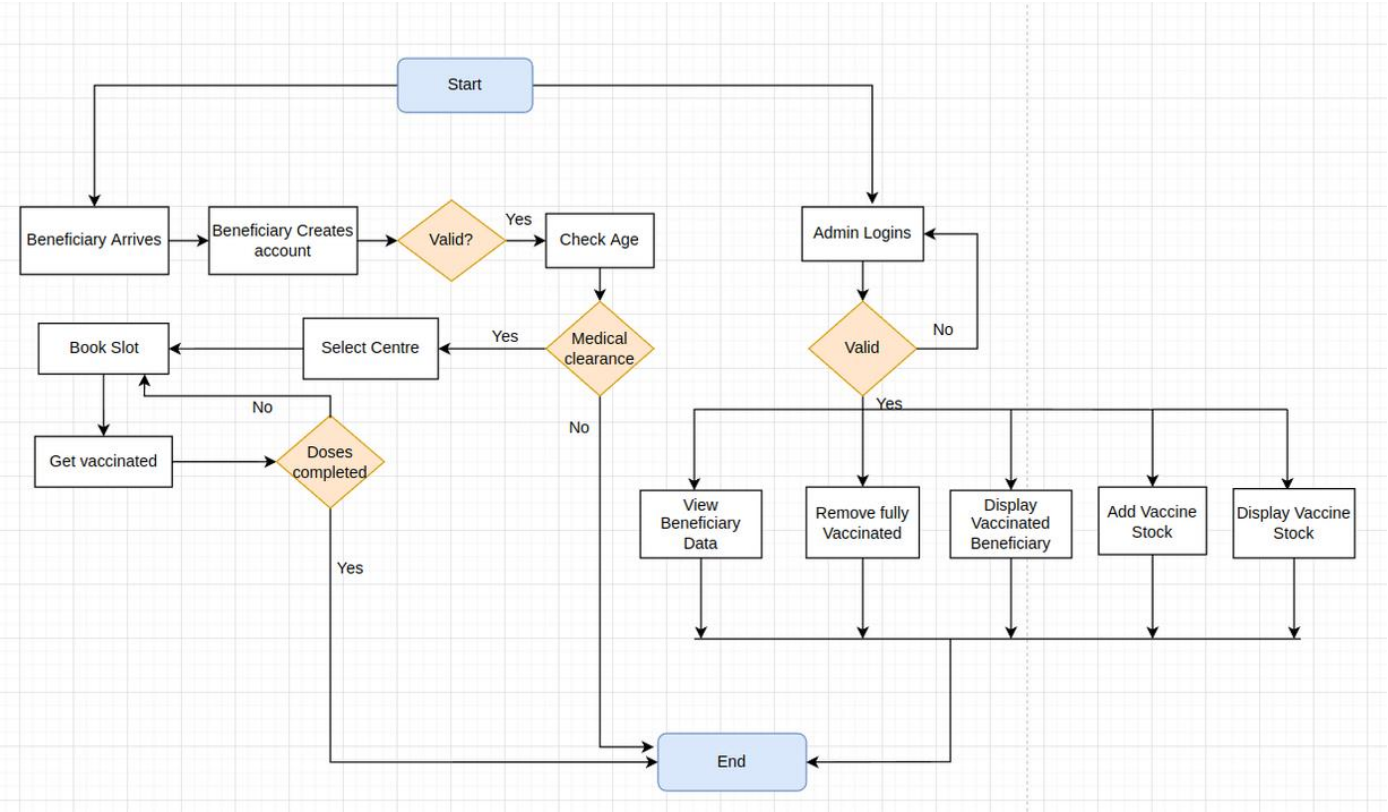
- **Beneficiary Flowchart**



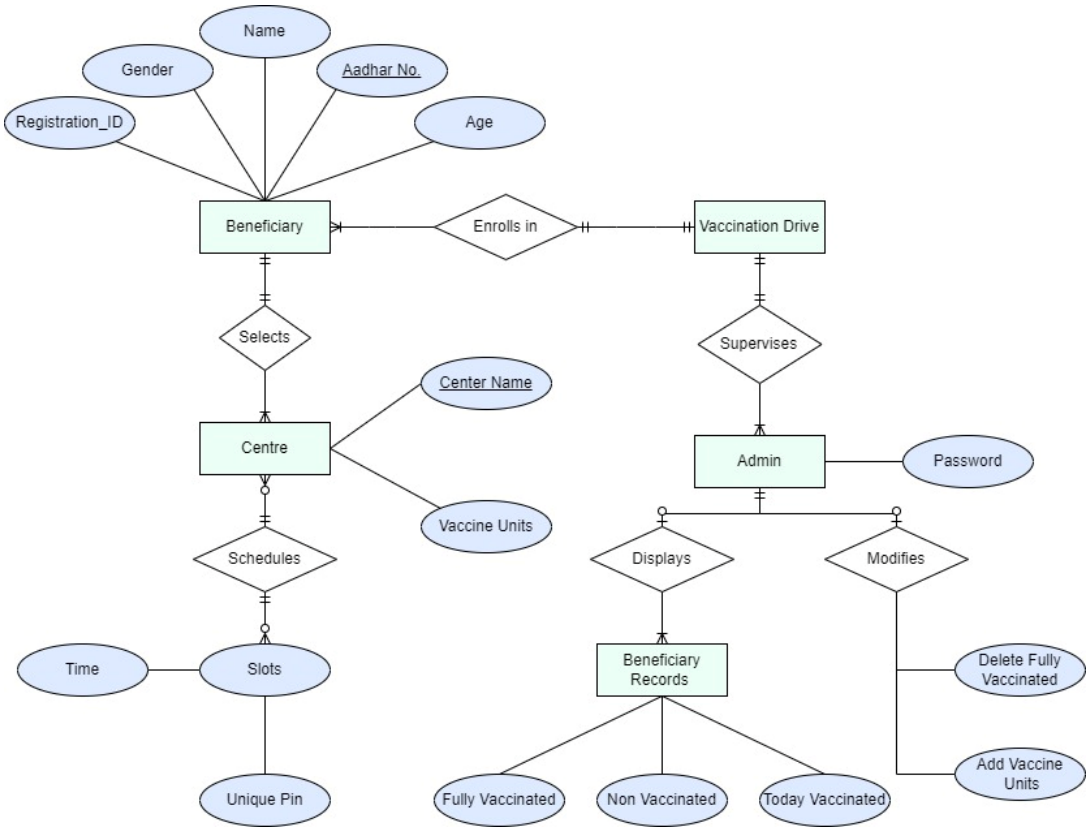
- **Admin Flowchart**



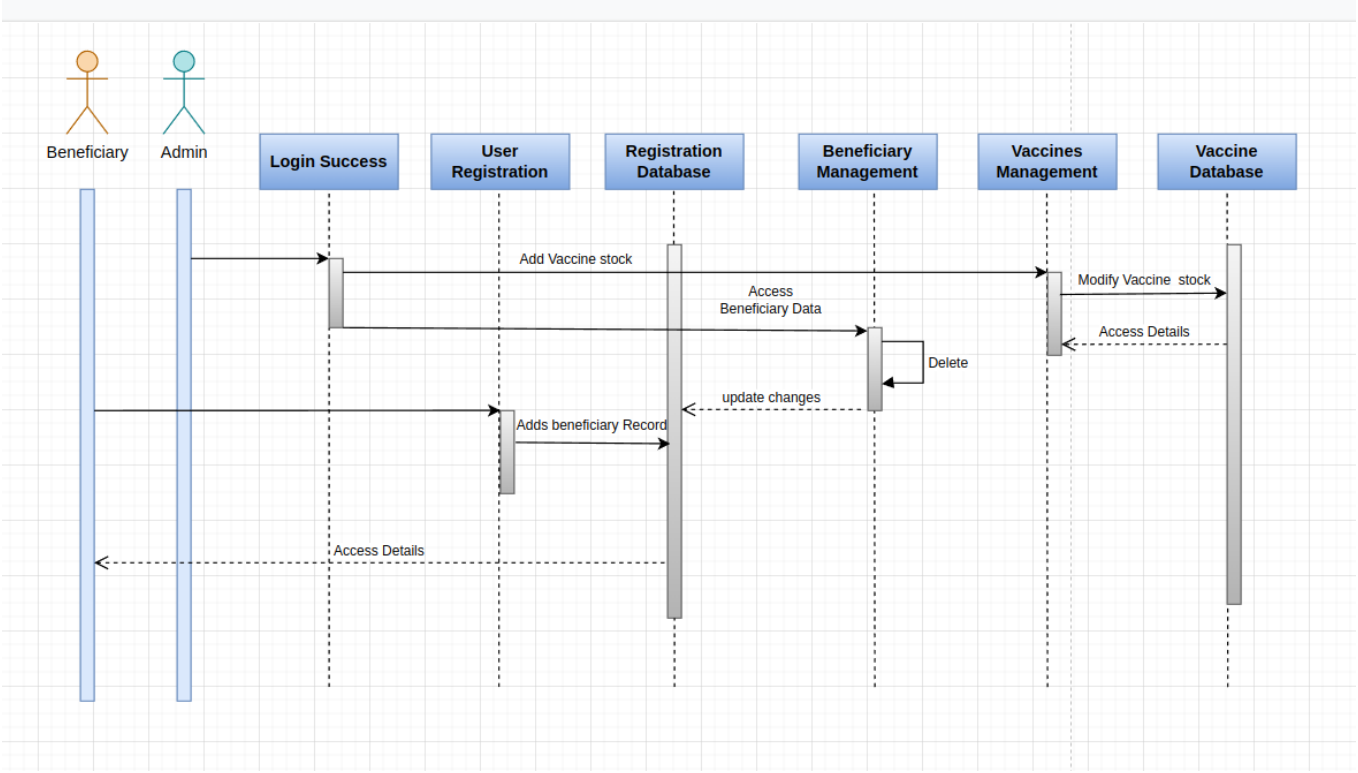
- **Combined Flowchart**



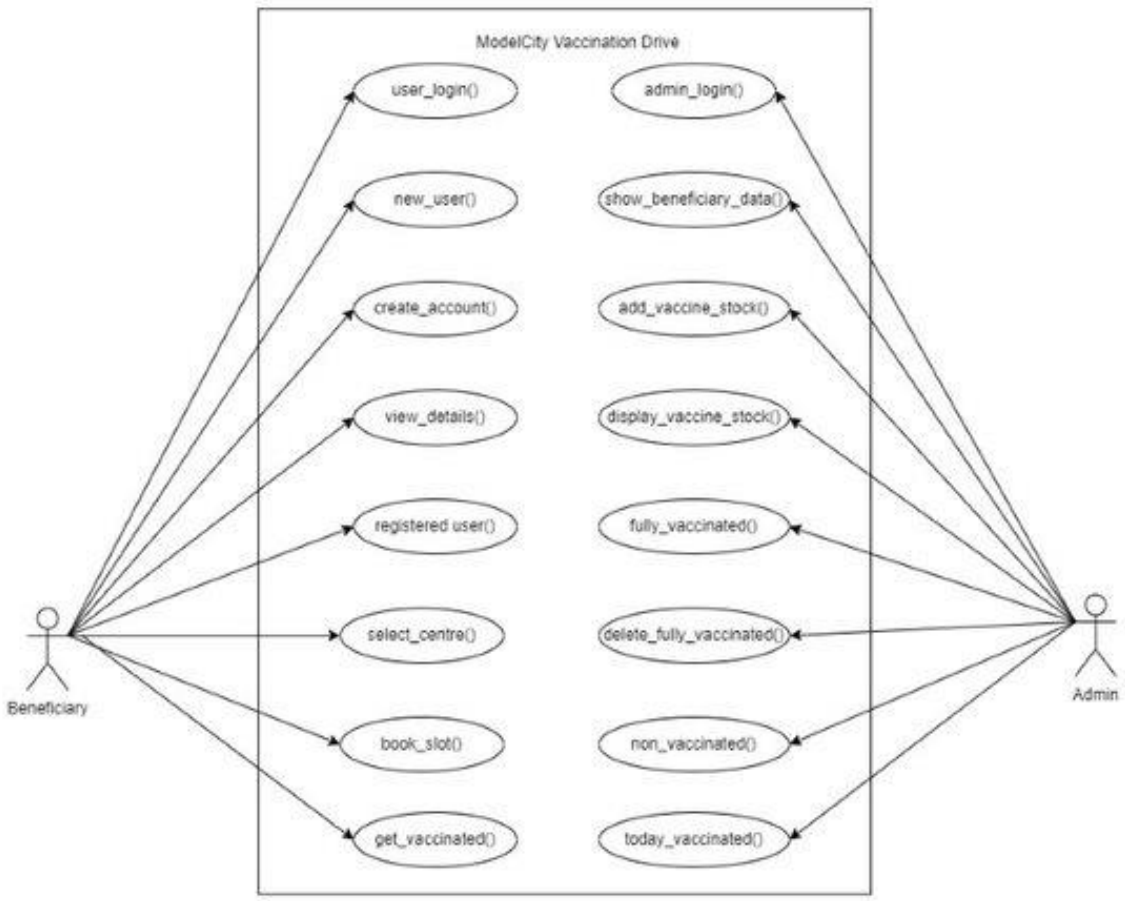
3.3. ER Diagram



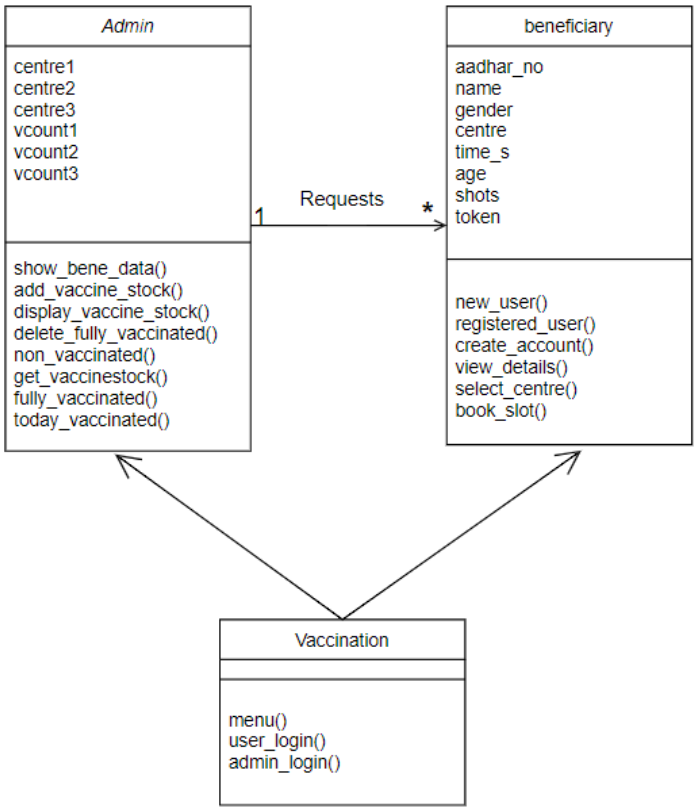
3.4. Sequence Diagram



3.5. Use Case Diagram



3.6. Class Diagram



3.7. Pseudocode

- Aadhar_len()

Aadhar length = INPUT : " Enter aadhar number:"
IF aadhar length is equal to 6
    PRINT "Valid Length"
ELSE
    PRINT "Invalid Length"
ENDIF

- Select_centre():
  Aadhar number = INPUT: "Enter your aadhar number:"
  IF aadhar number is valid
    THEN
      INPUT: "Please select your centre to book a slot:"
    IF center is equal to 1
      PRINT "Hawkins"
    ELSE IF center is equal to 2
      PRINT "Riverdale"
    ELSE IF center is equal to 3
      PRINT " Central Perk"

ELSE
PRINT “Invalid Centre”
ENDIF
ELSE “Invalid aadhar number”
ENDIF

- book_slot():
  Aadhar number= INPUT: “Enter your aadhar number:”
  IF aadhar number is valid
  THEN
  INPUT: ”Book your slot:”
  IF centre is equal to 1
  THEN
  PRINT “1. MORNING”
  INPUT: “Confirm your time between 10 to 12”
  IF time slot is between 10 to 12
  PRINT “Valid time slot”
  ELSE
  PRINT ”Invalid time slot”
  ELSEIF centre is equal to 2
  THEN
  PRINT “2. EVENING”
  INPUT: “Confirm your time between 1 to 4”
  IF time slot is between 1 to 4
  PRINT “Valid time slot”
  ELSE
  PRINT ”Invalid time slot”

3.8. Validations

- A unique 6-digits aadhar has to be entered by the beneficiary while creating an account. Also, an existing account holder cannot create an account as the aadhar number already exists in Registered_applicant_file.
- This application allows only beneficiaries above 18 years to apply for vaccination drive. Beneficiaries above age 35 need medical clearance in order to apply for vaccination.
- Name
- Gender
- passwd
- The number of vaccination shots for each beneficiary is restricted to 3.
- The vaccine file as well as the registered_applicant_file can be only modified by the user having the role of Admin.

4. Detailed Features and Requirements

4.1. Functional Requirements

- menu():The user will be encountered with two options after starting the application: 1.Beneficiary Login 2. Admin Login. Beneficiaries are the ones who want to apply for vaccination drive whereas the admin login is for government personnel's who would supervise the vaccination drive.

- user_login():
  There would be two choices for beneficiary login: 1.New 2.Registered. The registered beneficiaries can access the application by entering the allocated register id. Whereas the new beneficiaries have to enter the required details and is redirected to Registered choice.

- new_user():
  After choosing the new option in the previous login, user will be given two options: 1. Create Account 2.View Details.

- create_account():
  Details such as name, aadhar number, gender and age are to be entered for creating a new account.
  These details are first crosschecked with Aadhar File and only then they will be stored in Registered Applicant file.

- view_details():
  Details of the created account are viewed

- registered_user():
  Menu for registered beneficiary will be displayed.

- select_centre():
  Vaccination centres are selected by beneficiaries based on location and availability.

- book_slot():
  Slots are selected according to beneficiary's' convenience.

- get_vaccinated():
  The beneficiary has to verify the token provided while filling the slot, only after this the vaccination is said to be successful. After completion of this stage, the count of the vaccination unit is decremented for that particular centre in the vaccination file.

- checkpassword():
  A government personnel can login by entering the specified password so as to access all the functions related to the vaccination application. Only after the password is matched, the screen is directed to Admin menu.

- admin_login():
  Menu for admin role will be displayed.

- show_bene_data():
  Data of all the beneficiary records is displayed.

- add_vaccine_stock():
  The user would be able to edit the of number of vaccine units allocated to each centre. These details are than updated in Vaccine file.

- display_vaccine_stock():
  Displays the details of number of vaccine units in each allocated centres. These details are fetched from Vaccine file.

- fully_vaccinated():
  Displays the list of beneficiary that have vaccine count equal to 3. These details are fetched from Registered Applicant file.

- delete_fully_vaccinated():
  The user would be able to delete the record of the beneficiary that have reached the vaccine count to 3. These details are than updated in Registered Applicant file.

- non_vaccinated():
  Displays the list of beneficiaries that have vaccine count equal to 0. These details are fetched from Registered Applicant file.

- today_vaccinated():
  Displays the list of beneficiary that have been vaccinated today. These details are fetched from Registered Applicant file.

5. Tools Report

5.1. Cpp Check:

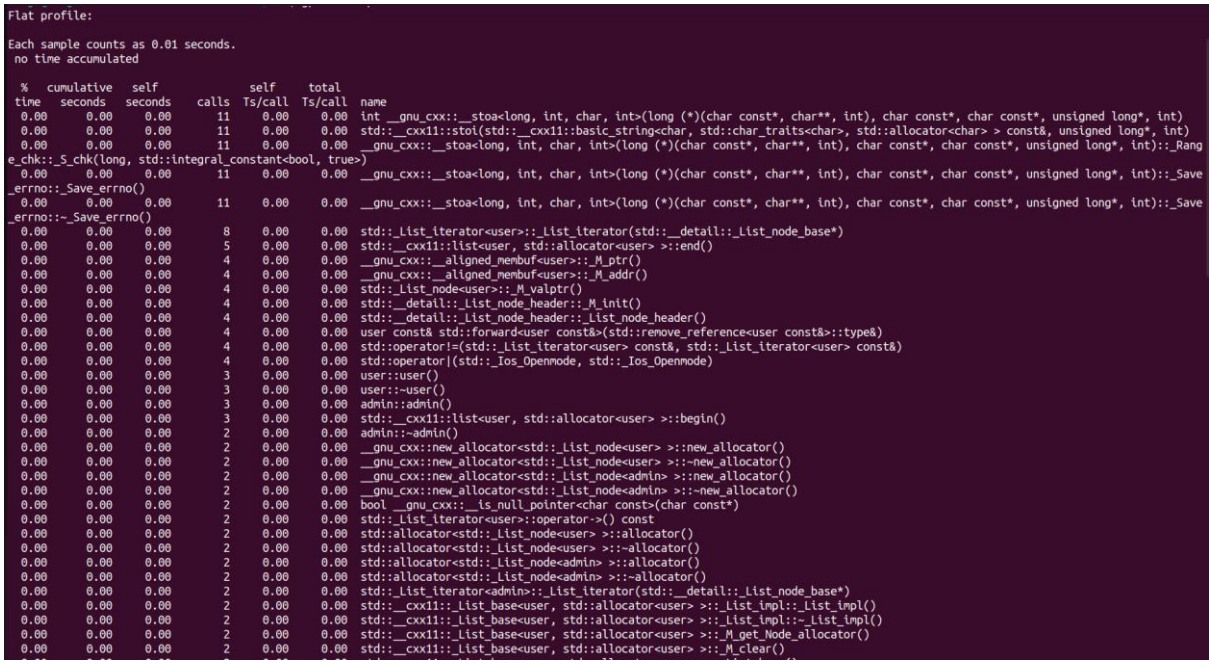
```
header.hpp:59:4: performance: Variable 'centre' is assigned in constructor body. Consider performing initialization in initialization list. [useInitializationList]
   centre="NA";
   ^
header.hpp:85:4: performance: Variable 'centre1' is assigned in constructor body. Consider performing initialization in initialization list. [useInitializationList]
   centre1="Hawkins";
   ^
header.hpp:86:4: performance: Variable 'centre2' is assigned in constructor body. Consider performing initialization in initialization list. [useInitializationList]
   centre2="Riverdale";
   ^
header.hpp:87:4: performance: Variable 'centre3' is assigned in constructor body. Consider performing initialization in initialization list. [useInitializationList]
   centre3="Central Perk";
   ^
vaccine3.cpp:712:9: style: Condition 'flag==1' is always false [knownConditionTrueFalse]
 if(flag==1)
       ^
vaccine3.cpp:703:11: note: Assignment 'flag=0', assigned value is 0
 int flag=0;
          ^
vaccine3.cpp:712:9: note: Condition 'flag==1' is always false
 if(flag==1)
       ^
password.cpp:95:6: style: Variable 'nchr' is reassigned a value before the old one has been used. [redundantAssignment]
 nchr=password(&p,MAXPW,'*',fp);
     ^
password.cpp:91:6: note: nchr is assigned
 nchr=password(&p,MAXPW,'*',fp);
     ^
password.cpp:95:6: note: nchr is overwritten
 nchr=password(&p,MAXPW,'*',fp);
     ^
validation.cpp:35:6: style: The scope of the variable 'y' can be reduced. [variableScope]
 int y;
     ^
vaccine3.cpp:147:6: style: The scope of the variable 'choice' can be reduced. [variableScope]
 int choice=0;
     ^
file.cpp:25:7: style: Local variable 'temp' shadows outer variable [shadowVariable]
  int temp = sizeof(u);
      ^
vaccine3.cpp:10:8: note: Shadowed declaration
string temp,input;
       ^
file.cpp:25:7: note: Shadow variable
  int temp = sizeof(u);
      ^
```

5.2. Gcov:

```
        -:    0:Source:main.cpp
        -:    0:Graph:main.gcno
        -:    0:Data:main.gcda
        -:    0:Runs:1
        -:    1:/*****************************************************************************
        -:    2: *  FILENAME     : main.c
        -:    3: *
        -:    4: *  DESCRIPTION : This file is used as Main menu of the vaccination drive which calls
        -:    5: *                user login and admin login options.
        -:    6: *
        -:    7: *  REVISION HISTORY
        -:    8: *
        -:    9: *  DATE            NAME          REASON
        -:   10: *
        -:   11: * -------------------------------------------------
        -:   12: *
        -:   13: *                   Username      Menu
        -:   14: *
        -:   15: * ****************************************************************************/
        -:   16:#include "password.cpp" // Required header file
        -:   17:#include "header.hpp"
        -:   18:#include "functions.cpp"
        -:   19:using namespace std;
        -:   20:
function _ZN11vaccination4menuEv called 1 returned 100% blocks executed 75%
        1:   21:void vaccination :: menu()
        -:   22:{
        1:   23:        system("clear");
call    0 returned 1
        -:   24:        int choice;
        1:   25:        int admin_pass=0;
        4:   26:        while(choice!=3)
branch  0 taken 3
branch  1 taken 1 (fallthrough)
        -:   27:        {
        3:   28:                cout<<endl<<"--------------------------------"<<endl;
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   29:                cout<<endl<<"\t MAIN MENU"<<endl;
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   30:                cout<<endl<<"\t1. User Login"<<endl;
call    0 returned 3
```

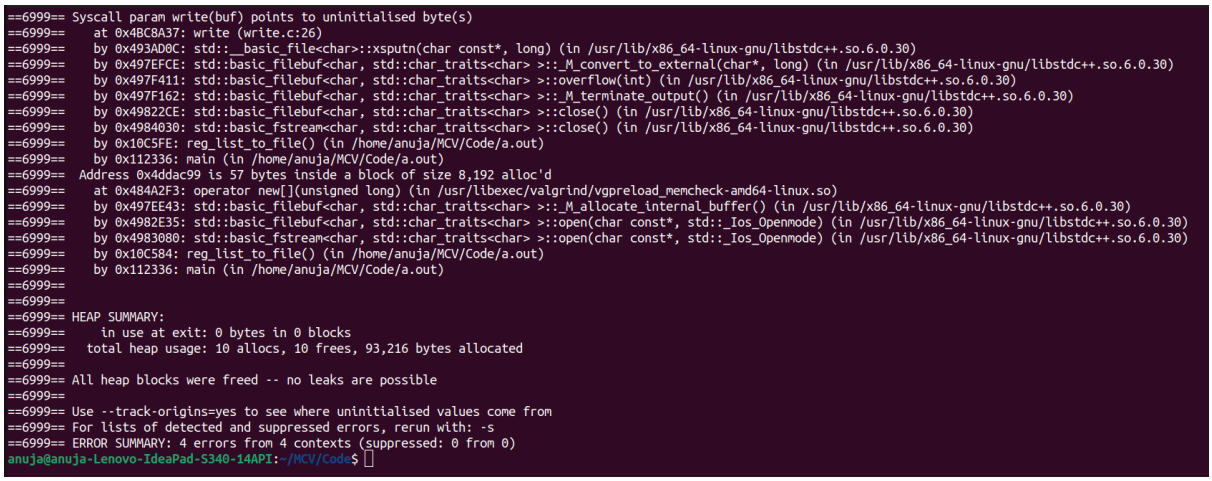
```
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   31:                cout<<endl<<"\t2. Admin Login"<<endl;
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   32:                cout<<endl<<"\t3. Exit"<<endl;
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   33:                cout<<endl<<"--------------------------------"<<endl;
call    0 returned 3
call    1 returned 3
call    2 returned 3
        3:   34:                cin>>temp;
call    0 returned 3
        -:   35:                try
        -:   36:                {
        3:   37:                        choice=stoi(temp);
call    0 returned 3
branch  1 taken 3 (fallthrough)
branch  2 taken 0 (throw)
        -:   38:                }
    =====:   39:                catch(exception& ex)
branch  0 never executed
branch  1 never executed
call    2 never executed
        -:   40:                {
    =====:   41:                        cout<<endl<<"Exception Occured...Enter only digits"<<endl;
call    0 never executed
branch  1 never executed
branch  2 never executed
call    3 never executed
branch  4 never executed
branch  5 never executed
call    6 never executed
branch  7 never executed
branch  8 never executed
    =====:   42:                }
call    0 never executed
call    1 never executed
        3:   43:                switch(choice)
branch  0 taken 1
branch  1 taken 1
branch  2 taken 1
branch  3 taken 0
        -:   44:                {
        1:   45:                        case 1:
        1:   46:                                system("clear");
call    0 returned 1
        1:   47:                                user_login();   // calling user_login function
call    0 returned 1
        1:   48:                                break;
        1:   49:                        case 2: system("clear");
call    0 returned 1
        1:   50:                                cout<<endl<<"\t\tAdmin Login"<<endl<<endl;
call    0 returned 1
call    1 returned 1
call    2 returned 1
call    3 returned 1
        1:   51:                                admin_pass=checkpassword();     // check admin's password
call    0 returned 1
       1*:   52:                                if(admin_pass==1)
branch  0 taken 0 (fallthrough)
branch  1 taken 1
        -:   53:                                {
    #####:   54:                                        printf("\nINVALID PASSWORD!!\n");
call    0 never executed
    #####:   55:                                        continue;
        -:   56:                                }
        -:   57:                                else
        -:   58:                                {
        1:   59:                                        admin_login();  // calling admin_login function
call    0 returned 1
        -:   60:                                }
        1:   61:                                break;
        1:   62:                        case 3:
        1:   63:                                system("clear");
call    0 returned 1
        1:   64:                                break;
    #####:   65:                        default:system("clear");
call    0 never executed
    #####:   66:                                cout<<endl<<"Invalid Choice"<<endl;
call    0 never executed
call    1 never executed
call    2 never executed
```

5.3. Gprof



```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00       11     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned long*, int)
  0.00      0.00     0.00       11     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, unsigned long*, int)
  0.00      0.00     0.00       11     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned long*, int)::_Rang
e_chk::_S_chk(long, std::integral_constant<bool, true>)
  0.00      0.00     0.00       11     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned long*, int)::_Save
_errno::_Save_errno()
  0.00      0.00     0.00       11     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned long*, int)::_Save
_errno::~_Save_errno()
  0.00      0.00     0.00        8     0.00     0.00  std::_List_iterator<user>::_List_iterator(std::__detail::_List_node_base*)
  0.00      0.00     0.00        5     0.00     0.00  std::__cxx11::list<user, std::allocator<user> >::end()
  0.00      0.00     0.00        4     0.00     0.00  __gnu_cxx::__aligned_membuf<user>::_M_ptr()
  0.00      0.00     0.00        4     0.00     0.00  __gnu_cxx::__aligned_membuf<user>::_M_addr()
  0.00      0.00     0.00        4     0.00     0.00  std::_List_node<user>::_M_valptr()
  0.00      0.00     0.00        4     0.00     0.00  std::__detail::_List_node_header::_M_init()
  0.00      0.00     0.00        4     0.00     0.00  std::__detail::_List_node_header::_List_node_header()
  0.00      0.00     0.00        4     0.00     0.00  user const& std::forward<user const&>(std::remove_reference<user const&>::type&)
  0.00      0.00     0.00        4     0.00     0.00  std::operator!=(std::_List_iterator<user> const&, std::_List_iterator<user> const&)
  0.00      0.00     0.00        4     0.00     0.00  std::operator|(std::_Ios_Openmode, std::_Ios_Openmode)
  0.00      0.00     0.00        3     0.00     0.00  user::user()
  0.00      0.00     0.00        3     0.00     0.00  user::~user()
  0.00      0.00     0.00        3     0.00     0.00  admin::admin()
  0.00      0.00     0.00        3     0.00     0.00  std::__cxx11::list<user, std::allocator<user> >::begin()
  0.00      0.00     0.00        2     0.00     0.00  admin::~admin()
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::new_allocator<std::_List_node<user> >::new_allocator()
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::new_allocator<std::_List_node<user> >::~new_allocator()
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::new_allocator<std::_List_node<admin> >::new_allocator()
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::new_allocator<std::_List_node<admin> >::~new_allocator()
  0.00      0.00     0.00        2     0.00     0.00  bool __gnu_cxx::__is_null_pointer<char const>(char const*)
  0.00      0.00     0.00        2     0.00     0.00  std::_List_iterator<user>::operator->() const
  0.00      0.00     0.00        2     0.00     0.00  std::allocator<std::_List_node<user> >::allocator()
  0.00      0.00     0.00        2     0.00     0.00  std::allocator<std::_List_node<user> >::~allocator()
  0.00      0.00     0.00        2     0.00     0.00  std::allocator<std::_List_node<admin> >::allocator()
  0.00      0.00     0.00        2     0.00     0.00  std::allocator<std::_List_node<admin> >::~allocator()
  0.00      0.00     0.00        2     0.00     0.00  std::_List_iterator<admin>::_List_iterator(std::__detail::_List_node_base*)
  0.00      0.00     0.00        2     0.00     0.00  std::__cxx11::_List_base<user, std::allocator<user> >::_List_impl::_List_impl()
  0.00      0.00     0.00        2     0.00     0.00  std::__cxx11::_List_base<user, std::allocator<user> >::_List_impl::~_List_impl()
  0.00      0.00     0.00        2     0.00     0.00  std::__cxx11::_List_base<user, std::allocator<user> >::_M_get_Node_allocator()
  0.00      0.00     0.00        2     0.00     0.00  std::__cxx11::_List_base<user, std::allocator<user> >::_M_clear()
```

5.4. Valgrind



```
==6999== Syscall param write(buf) points to uninitialised byte(s)
==6999==    at 0x4BC8A37: write (write.c:26)
==6999==    by 0x493AD0C: std::__basic_file<char>::xsputn(char const*, long) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x497EFCE: std::basic_filebuf<char, std::char_traits<char> >::_M_convert_to_external(char*, long) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x497F411: std::basic_filebuf<char, std::char_traits<char> >::overflow(int) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x497F162: std::basic_filebuf<char, std::char_traits<char> >::_M_terminate_output() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x49822CE: std::basic_filebuf<char, std::char_traits<char> >::close() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x4984030: std::basic_fstream<char, std::char_traits<char> >::close() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x10C5FE: reg_list_to_file() (in /home/anuja/MCV/Code/a.out)
==6999==    by 0x112336: main (in /home/anuja/MCV/Code/a.out)
==6999==  Address 0x4ddac99 is 57 bytes inside a block of size 8,192 alloc'd
==6999==    at 0x484A2F3: operator new[](unsigned long) (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==6999==    by 0x497EE43: std::basic_filebuf<char, std::char_traits<char> >::_M_allocate_internal_buffer() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x4982E35: std::basic_filebuf<char, std::char_traits<char> >::open(char const*, std::_Ios_Openmode) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x4983080: std::basic_fstream<char, std::char_traits<char> >::open(char const*, std::_Ios_Openmode) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30)
==6999==    by 0x10C584: reg_list_to_file() (in /home/anuja/MCV/Code/a.out)
==6999==    by 0x112336: main (in /home/anuja/MCV/Code/a.out)
==6999==
==6999==
==6999== HEAP SUMMARY:
==6999==     in use at exit: 0 bytes in 0 blocks
==6999==   total heap usage: 10 allocs, 10 frees, 93,216 bytes allocated
==6999==
==6999== All heap blocks were freed -- no leaks are possible
==6999==
==6999== Use --track-origins=yes to see where uninitialised values come from
==6999== For lists of detected and suppressed errors, rerun with: -s
==6999== ERROR SUMMARY: 4 errors from 4 contexts (suppressed: 0 from 0)
anuja@anuja-Lenovo-IdeaPad-S340-14API:~/MCV/Code$
```

6. Testing

6.1. Unit Testing

1.Create Account

```
anuja@anuja-Lenovo-IdeaPad-S340-14API:~/MCV/Code$ g++ -o cpptest cppUnit_testing.cpp -lcppunit
anuja@anuja-Lenovo-IdeaPad-S340-14API:~/MCV/Code$ ./cpptest
testuser::test_create_account
Enter Aadhar Number : 123456

Enter your Age : 22

Enter your name : anuja

Your Registeration ID is : an1234

Enter Gender (F/M/O) : F
```

2.View Details

```
testuser::test_view_detailsEnter your Aadhar number to view your details: 123456

----------------View Details-----------------

Aadhar  Name    Age     Gender  Shots
123456  anuja   22      F       0
```

3.Select Centre

```
testuser::test_select_centreEnter your aadhar number: 123456


1. Hawkins
2. Riverdale
3. Central Perk

Select one centre
2

Your centre for vaccination drive is succesfully selected

_____________________________________________________________
```

4.Book Slot

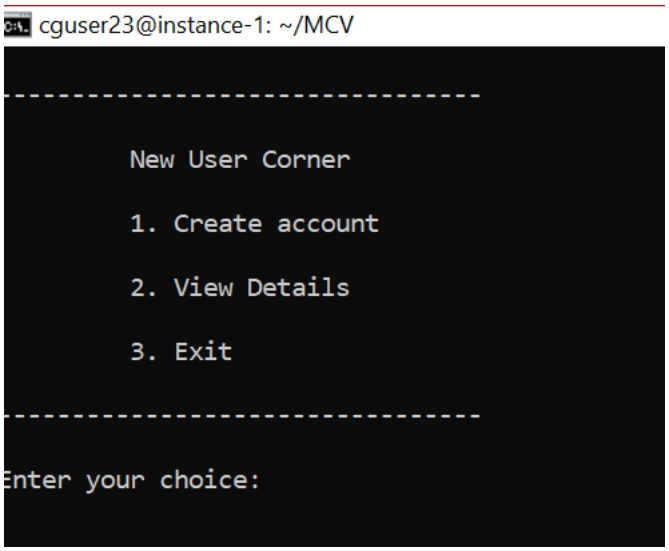
```
testuser::test_book_slotEnter your aadhar number: 123456



**************Book your slot**************

1. Morning
2. Afternoon


****************************************


Select one slot:
1

YOUR TOKEN FOR BOOKING IS: **32**

PLEASE CONFIRM YOUR TOKEN
32
Confirm your time between 10 to 12
11

Your Vaccination slot is selected successfully

_____
```

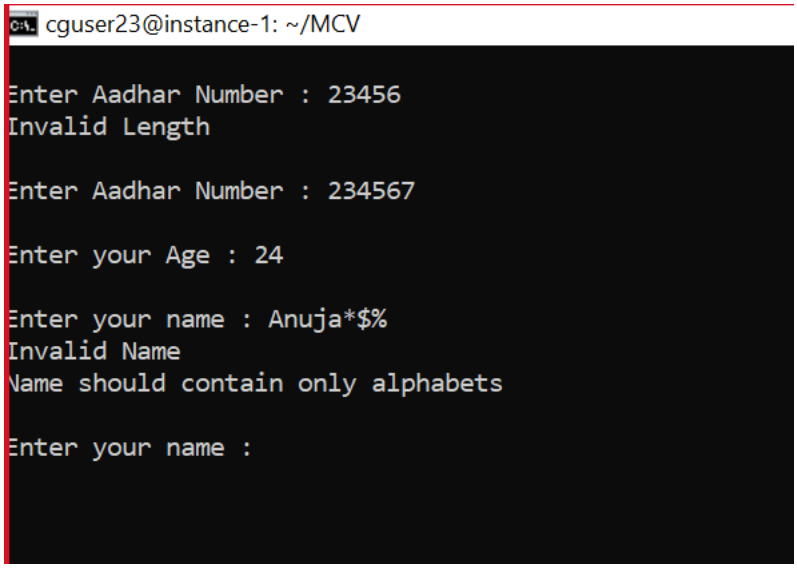
5.Get Vaccinated

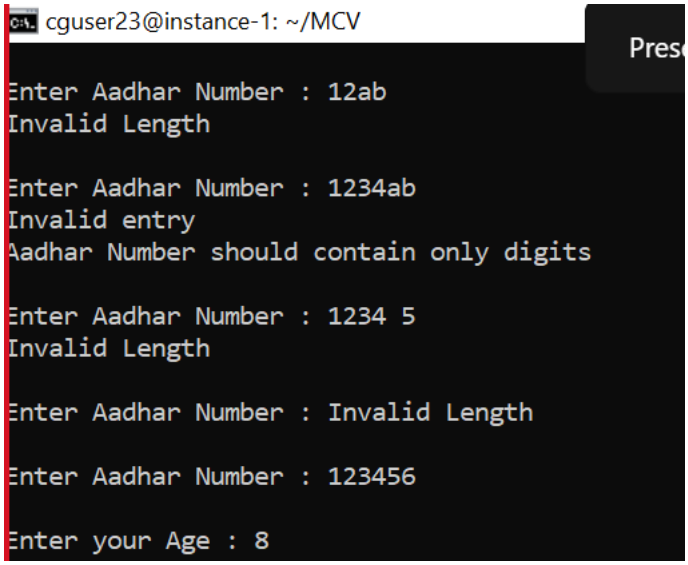
```
testuser::test_get_vaccinatedEnter your Aadhar number: 123456

PLEASE CONFIRM YOUR TOKEN
42

Entered token is wrong
 : OK
OK (5)
```

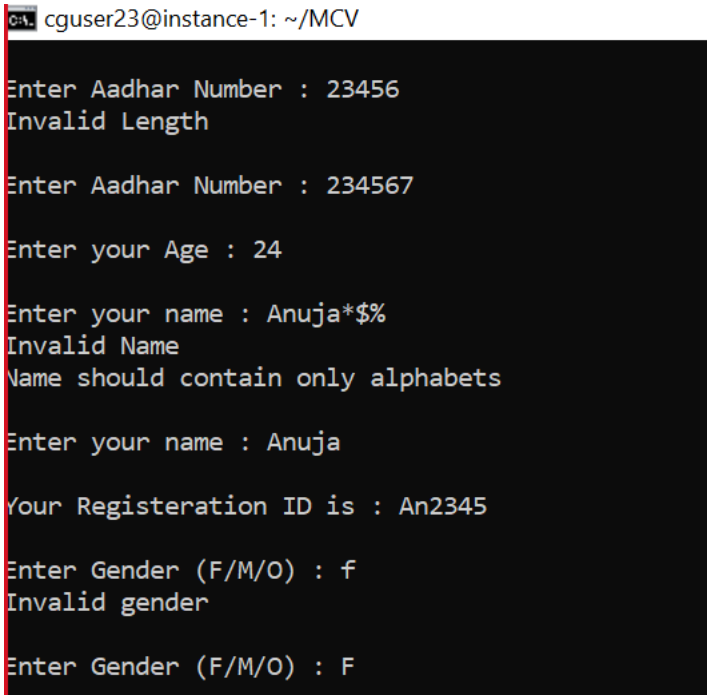
6.2. Integration Testing

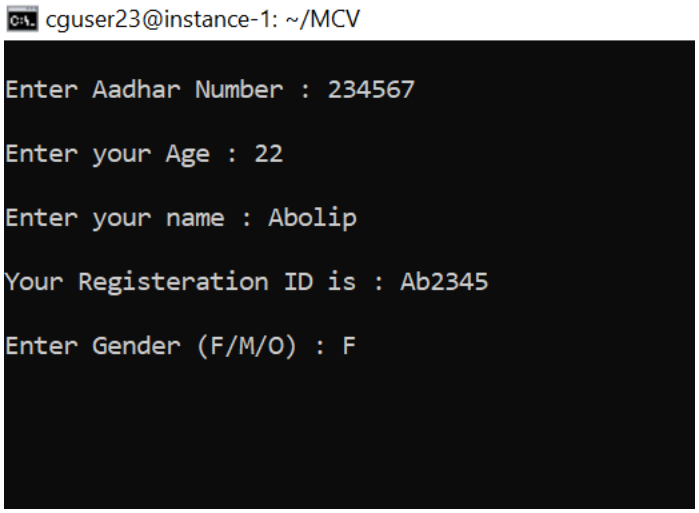
IT_CASE 1: For creating account






cguser23@instance-1: ~/MCV
Enter Aadhar Number : 12ab
Invalid Length
Enter Aadhar Number : 1234ab
Invalid entry
Aadhar Number should contain only digits
Enter Aadhar Number : 1234 5
Invalid Length
Enter Aadhar Number : Invalid Length
Enter Aadhar Number : 123456
Enter your Age : 8


cguser23@instance-1: ~/MCV
Enter Aadhar Number : 23456
Invalid Length
Enter Aadhar Number : 234567
Enter your Age : 24
Enter your name : Anuja*$%
Invalid Name
Name should contain only alphabets
Enter your name : Anuja
Your Registeration ID is : An2345
Enter Gender (F/M/O) : f
Invalid gender
Enter Gender (F/M/O) : F


cguser23@instance-1: ~/MCV
Enter Aadhar Number : 234567
Enter your Age : 22
Enter your name : Abolip
Your Registeration ID is : Ab2345
Enter Gender (F/M/O) : F

IT_CASE 2 : View Details

```
cguser23@instance-1: ~/MCV
Enter your Aadhar number to view your details: 123456

----------------View Details-----------------

Aadhar  Name    Age     Gender  Shots
123456  Aboli   22      F       0
```

IT_CASE 3 : Select Centre

```
cguser23@instance-1: ~/MCV

---------------------------------

         Registered User Corner

         1. Select centre

         2. Book Vaccine Slot

         3. View Details

         4. Get Vaccinated

         5. Exit

---------------------------------

Enter your choice:
```

```
Select cguser23@instance-1: ~/MCV
Enter your aadhar number: 123456


1. Hawkins
2. Riverdale
3. Central Perk

Select one centre
1

Your centre for vaccination drive is succesfully selected

_____________________________________________________________
```

IT_CASE 4 : Book Vaccine Slot

```
cguser23@instance-1: ~/MCV
Enter your aadhar number: 123456



**************Book your slot**************

1. Morning
2. Afternoon


****************************************


Select one slot:
1

YOUR TOKEN FOR BOOKING IS: **30**

PLEASE CONFIRM YOUR TOKEN
30
Confirm your time between 10 to 12
11

Your Vaccination slot is selected successfully
```

```
cguser23@instance-1: ~/MCV
Enter your aadhar number: 123456



**************Book your slot**************

1. Morning
2. Afternoon


****************************************


Select one slot:
1

YOUR TOKEN FOR BOOKING IS: **33**

PLEASE CONFIRM YOUR TOKEN
31

Entered token is wrong. Please recheck your entered token
```

IT_CASE 5 : Get Vaccinated

```
cguser23@instance-1: ~/MCV
Enter your Aadhar number: 123456

PLEASE CONFIRM YOUR TOKEN
30

**********Congratulations on getting vaccinated**********

You have completed 1 doses

---------------------------------

         Registered User Corner

         1. Select centre

         2. Book Vaccine Slot

         3. View Details

         4. Get Vaccinated

         5. Exit

---------------------------------

Enter your choice:
```

IT_CASE 6: Show Beneficiary

```
cguser23@instance-1: ~/MCV
---------------------------------

         Admin Corner

         1. Show Beneficiary Data

         2. Add Vaccine stock

         3. Display Vaccine Stock

         4. Delete Fully Vaccinated Beneficiaries

         5. Display non vaccinated beneficiaries

         6. Display Fully Vaccinated Beneficiaries

         7. Display Today Vaccinated Beneficiaries

         8. Exit

---------------------------------

Enter your choice:
```

```
cguser23@instance-1: ~/MCV

Aadhar  Name    Age     Gender  Centre          Time slot       Shots

123456  Aboli   22      F       Riverdale               11              3
234567  Anuja   21      F       NA              0               0
345678  Nitika  24      F       NA              0               0
```

IT_CASE 7: Add Vaccine Stock

```
cguser23@instance-1: ~/MCV
1. Hawkins
2. Riverdale
3. Central Perk

Select the slot in which you want to add vaccine stock:
3

                -->>For Center 3<<--

Vaccine in Center *Central Perk* are: 15

        Enter Number of vaccines you want to add: 10
```

IT_CASE 8 : Display Vaccine Stock

```
cguser23@instance-1: ~/MCV


_____________________________


Centre          Vaccine Count
Hawkins                 25
Riverdale               4
Central Perk            15

_____________________________
```

IT_CASE 9 : Delete Fully Vaccinated Beneficiaries

```
cguser23@instance-1: ~/MCV

Enter the Aadhar number whose record you want to delete:234567
This beneficiary has not completed his all doses.
----You cannot delete this record.----

--------------------------------

        Admin Corner

        1. Show Beneficiary Data

        2. Add Vaccine stock

        3. Display Vaccine Stock

        4. Delete Fully Vaccinated Beneficiaries

        5. Display non vaccinated beneficiaries

        6. Display Fully Vaccinated Beneficiaries

        7. Display Today Vaccinated Beneficiaries

        8. Exit

--------------------------------

Enter your choice:
```

IT_CASE 10: Display non Vaccinated Beneficiaries

```
Aadhar  Name    Age     Gender  Centre          Time slot       Shots

234567  Vaastav 27      M       NA              0               0
345678  Nitika  19      F       NA              0               0
```

IT_CASE 11 : Display Fully Vaccinated Beneficiaries

```
Aadhar  Name    Age     Gender  Centre          Time slot       Shots

123456  anuja   22      F       Riverdale               11              3
```

IT_CASE 12 : Display Today Vaccinated Beneficiaries

```
Aadhar   Name    Age     Gender  Centre          Time slot       Shots
123456   anuja   22      F       Riverdale               11              3
```

7. Requirement Traceability Matrix

Req	Design Mapping	Code Mapping	UT Mapping	IT Mapping
MVD_01	3.1.1	menu()		
MVD_02	3.1.2	user_login()		
MVD_03	3.1.3	new_user()		
MVD_04	3.1.4	create account()	Test_case_1	IT_CASE 1
MVD_05	3.1.5	view_details()	Test_case_2	IT_CASE 2
MVD_06	3.1.6	registered_user()		
MVD_07	3.1.7	select_centre()	Test_case_3	IT_CASE 3
MVD_08	3.1.8	book_slot()	Test_case_4	IT_CASE 4
MVD_09	3.1.9	get_vaccinated()	Test_case_5	IT_CASE 5
MVD_10	3.1.10	checkpassword()		
MVD_11	3.1.11	admin_login()		
MVD_12	3.1.12	show_bene_data()		IT_CASE 6
MVD_13	3.1.13	add_vaccine_stock()		IT_CASE 7
MVD_14	3.1.14	display_vaccine_stock()		IT_CASE 8
MVD_15	3.1.15	fully_vaccinated()		IT_CASE 11
MVD_16	3.1.16	delete_fully_vaccinated()		IT_CASE 9
MVD_17	3.1.17	non_vaccinated()		IT_CASE 10
MVD_18	3.1.18	today_vaccinated()		IT_CASE 12

8. Reference

- https://www.softwaretestinghelp.com/linked-list/
- http://www.trytoprogram.com/cplusplus-programming/multiple-inheritance/
- https://www.simplilearn.com/tutorials/cpp-tutorial/classes-in-cpp