Capgemini

[redacted]

High Level Design & Low Level Design

Index

1. Introduction

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyse and give an in-depth .so in Dynaland is a country with population of around 20 crore.Earlier they used to have a population census every 4 years which has completely manual.The reports were taking a lot of time to be compiled and generated.Also that the govt.since they have the census coming up after 6 months they have decided to automate the system.

1.1 Intended Audience: -

The intended audience for this application can be any country that wants to hold a population census for their citizens by replacing the manual traditional method to a fully automated method. Which will not only help to reduce the manual labour but also be more fast and efficient in giving timely much more accurate reports based on the data.

1.2 Project Purpose: -

The purpose of this document is to show the requirements for the Dynaland Automated Population Census Application , which helps in the census of population Automated because of more population in Dynaland country they decided to census the population for every 4 years which was completely manual. The reports were taking a lot of time to generate so decided to things become easier to census population fastly so decided to census coming up after 6 months to automate the system.which will help them in Getting timely and correct reports. save time and money for the process of planning for the welfare of people through the correct data.

1.3 Key Project Objectives: -

a. Allow the Zone Representative to add Citizen's Information

b. Creating population Census Application

c. Allows the Central Representative of the country to update the changes in Citizen's information

d. Displays all the reports and queries required.

e. Modify/Update the Citizen Data.

f. Necessary Calculations for generating the reports.

1.4 Project scope : -

This project aims to create the Reports of Dynaland Automated Population Census Application. In which the Government should prepare reports about the census population Automatically . This saves time and money in the census population. so introduced an Automated concept to easily census the population.

2. Design Overview: -

- **Zone Representative comprises the following modules in Population Census Application :**

Name of the Module	Add Module
Handled by	
Description	The Zone Representative adds the record in the database

- **Central Representative comprises of the following modules in Population Census Application :**

Name of the Module	Change Admin ID and password module
Handled by	
Description	Changing the Admin details and password for admin

Name of the Module	Maintaining Citizen database module
Handled by	
Description	This module contains various options like view and edit the changes in citizen database

Name of the Module	Reports Module
Handled by	
Description	Contains reports like % of age group , % of male and female population , farming population, Below Poverty Line population.

Name of the Module	Queries Module
Handled by	
Description	Consists of queries like Literacy Rate, highest Literacy rate , % of rural and urban population and highest male and female population zone.

- **Population Census comprises of the following modules in reports:**

Name of the Module	Age Group Module
Handled by	
Description	It will show the percentage of citizen from different age group like <18, 18-40 , >40 for each zone and entire country

Name of the Module	Gender Group Module
Handled by	
Description	It will show the percentage of population of male ,female and transgender from each zone and whole country

Name of the Module	Farming Population Module
Handled by	
Description	It will show the percentage of citizen having occupation as farming from different zone and entire country

Name of the Module	Below Poverty Line Group Module
Handled by	
Description	Citizens having income less than 100000 are categorized as below poverty line hence the percentage of BPL over different zones and the whole country is calculated.

- **Population Census comprises of the following modules in Queries:**

Name of the Module	Literacy Rate Module
Handled by	
Description	By using this query the representative of the central government will be able to view the percentage of the literate population of the country.

Name of the Module	Zone with highest Literacy Rate Module
Handled by	
Description	By using this query the representative of the central government will be able to know which zone has the highest number of literacy rate.

Name of the Module	BPL Average income Module
Handled by	
Description	By using this query the representative of the central government will be able to know the average annual income of the below poverty level people.

Name of the Module	Rural and urban Population module
Handled by	
Description	By using this query the representative of the central government will be able to know the percentage of people living in the urban and rural areas.

Name of the Module	Zone with highest male and female population Module
Handled by	
Description	By using this query the representative of the central government will be able to know which are the zones having the highest number of males and females.

2.1 Design Objectives:

1. Creating an easy to use Population Census Application.
2. Categorize citizens based on their zone location.
3. Merging all zone data into a central database.
4. Central admin monitoring and updating over citizen data.
5. Generate accurate reports based on citizen data available.
6. Responsive to Queries.

2.1 Design Alternative: -

We have used classes for different functionalities and also we have used STL libraries for storing and manipulating data effectively , linked list structure is used to store data i.e.. SSID, Name, Address, Age, Gender, Annual income, Occupation, Qualification, House type , No of dependents and area type like rural or urban.

2.3 User Interface Paradigms: -

The Population Census Application provides an option to Central and zone representatives to manage and add citizens data respectively and generate various reports and queries from the database .

2.4 Validation: -

- SSIS should not be blank and Duplication is not allowed and characters aren't allowed in the SSID.
- In case of integer validation, if the entered SSID is not Integer it displays the message SSID must contain Integer only and should not contain more than 9 digits.
- We check for the validity of the name,Citizen name should not be empty and only alphabets.
- In case of Address validation we check that it should not be blank.
- Age should only contain integers and characters are not allowed,Also should be in a particular range between 1-120.
- Gender should also consist of character alphabets only and particularly gender should be entered as “m” for male , “f” for female and “t”for transgender only.
- Annual income shall be integers only and the minimum limit for annual income is set as 1000 .
- Occupation shall be entered as alphabets strictly .
- For checking the validity for qualification, the qualification of the citizen should contain characters and if there is no qualification for citizen then “NA” will be accepted , it cannot be blank.
- We check the validity for house type as it should accept only single characters such as “o” for own and “r” for rented strictly no other characters or integers are allowed.

- No dependents for citizens shall not be greater than 20 dependents hence only integer values less than 20 and no character values are allowed.

- We check the validity for area type as it should accept only single characters such as “r” for rural areas and “u” for urban areas.

3. SYSTEM ARCHITECTURE: -

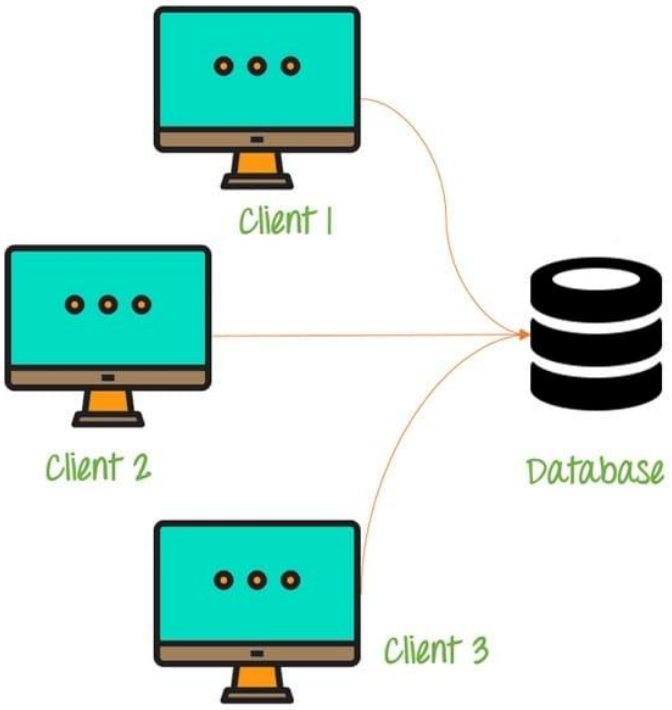
3.1. Database Architecture

The architecture used in this system comprises of the database architecture. It is a representation of the database management system design, wherein you can design, develop, implement and maintain the database. This architecture allows dividing the database into different components that can be independently modified, changed, replaced and altered as required for the system.

The database architecture is divided into three tiers namely:-

- 1 - Tier Architecture
- 2 - Tier Architecture
- 3 - Tier Architecture

Our system is based on the Tier 1 model of the database architecture. In this type of model the database is directly available to the user, the user can directly access the database and all of its contents. Which enables the user to directly interact and execute operations.



Some of the characteristics of Database Architecture are:

Self-Describing Nature of a Database System :

- One of the most fundamental characteristics of the database approach is that the database system contains not only the database itself but also an entire definition or description of the database structure and constraints also known as metadata of the database.

Isolation between Data, Programs and Data Abstraction:

- In a traditional file processing system, the structure of database knowledge files is embedded within the application programs, so any changes to the structure of a file may require changing all programs that access that file.

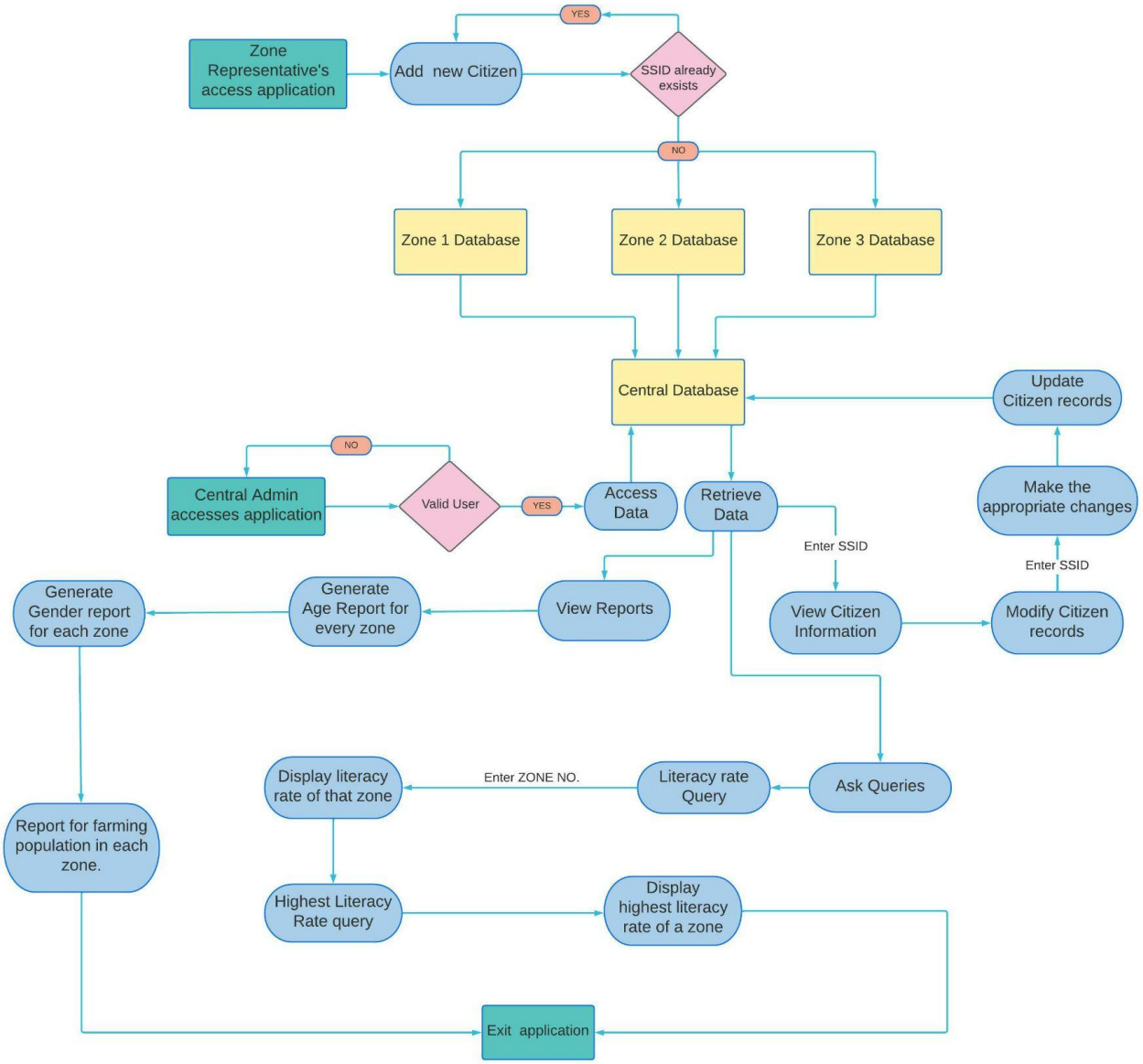
Support for Multiple Views of the Data :

- A database sometimes has many users, each of whom may require a special perspective or view of the database.

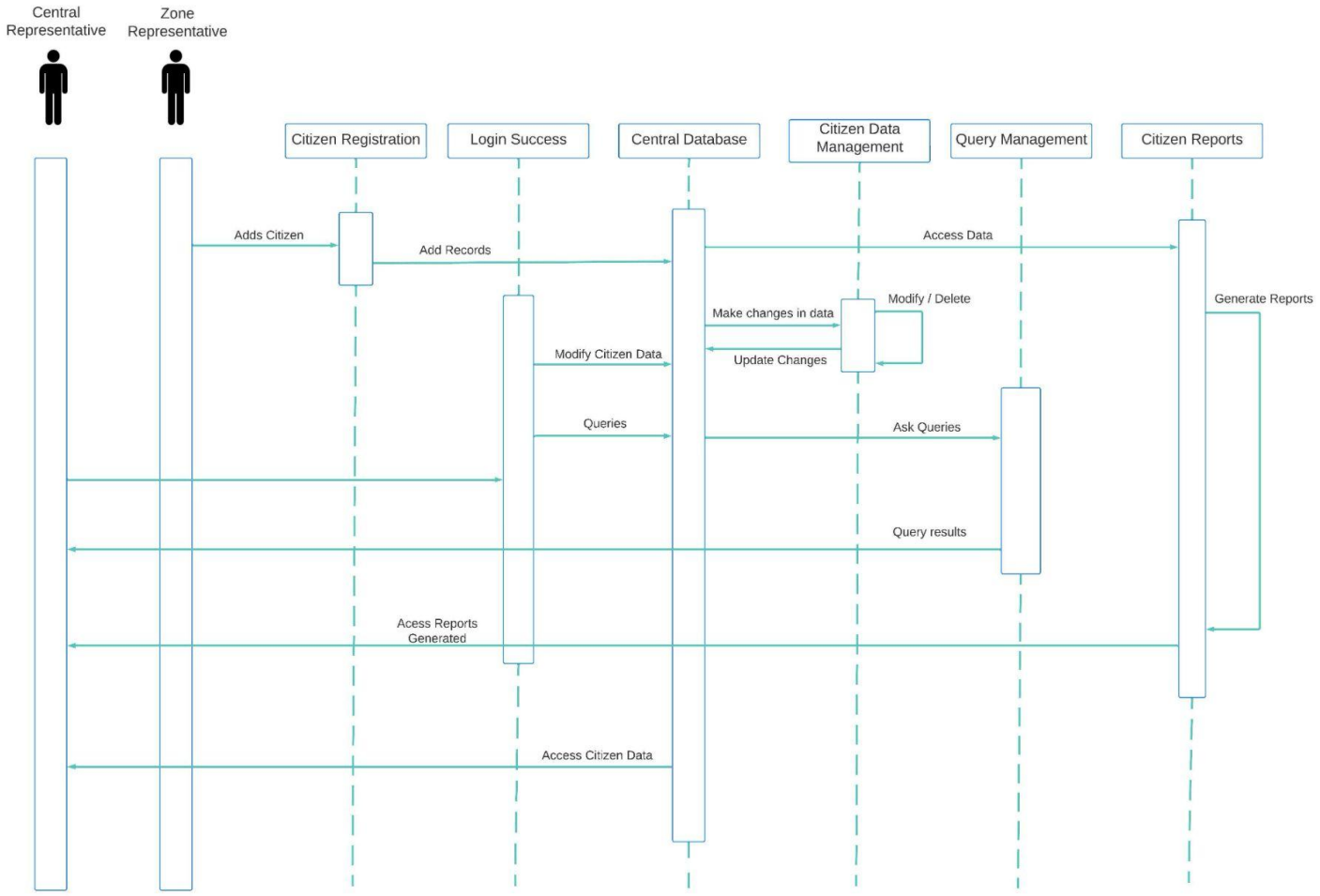
Sharing of knowledge and Multi-user Transaction Processing :

- A multi-user DBMS, as its name implies, must allow multiple users to access the database at an equivalent time or concurrently.

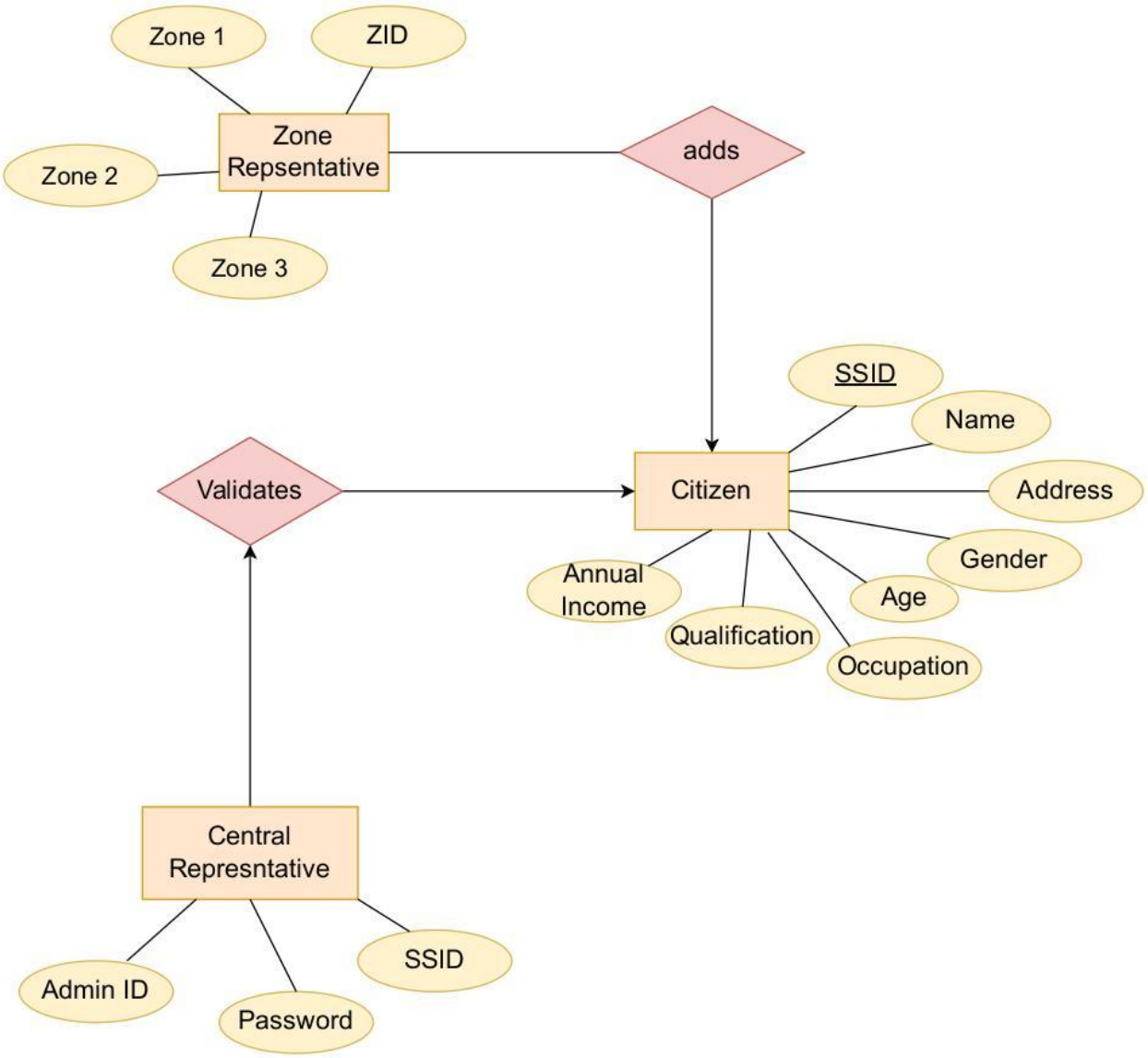
4. DETAILED SYSTEM DESIGN:



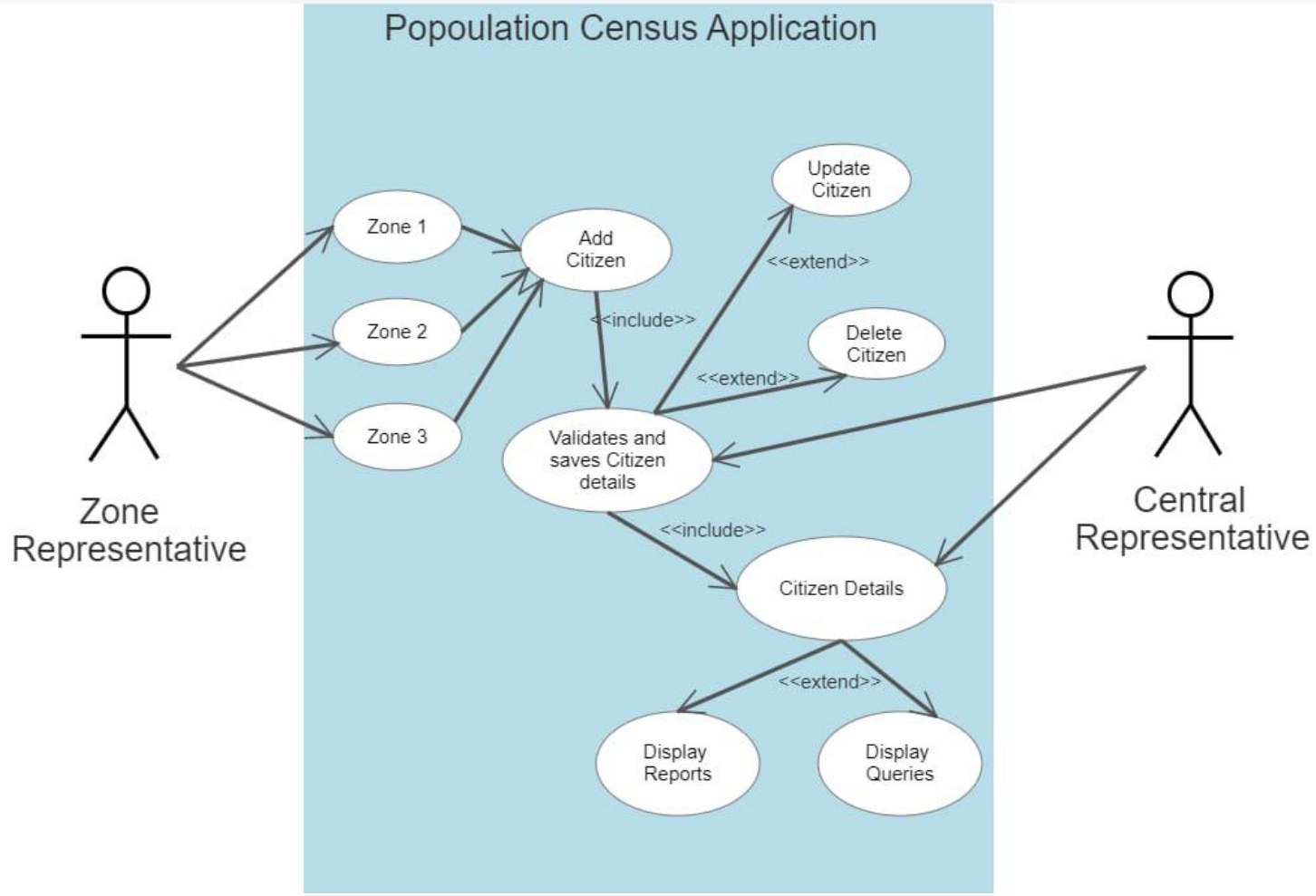
4.1 Flowchart of the Application.



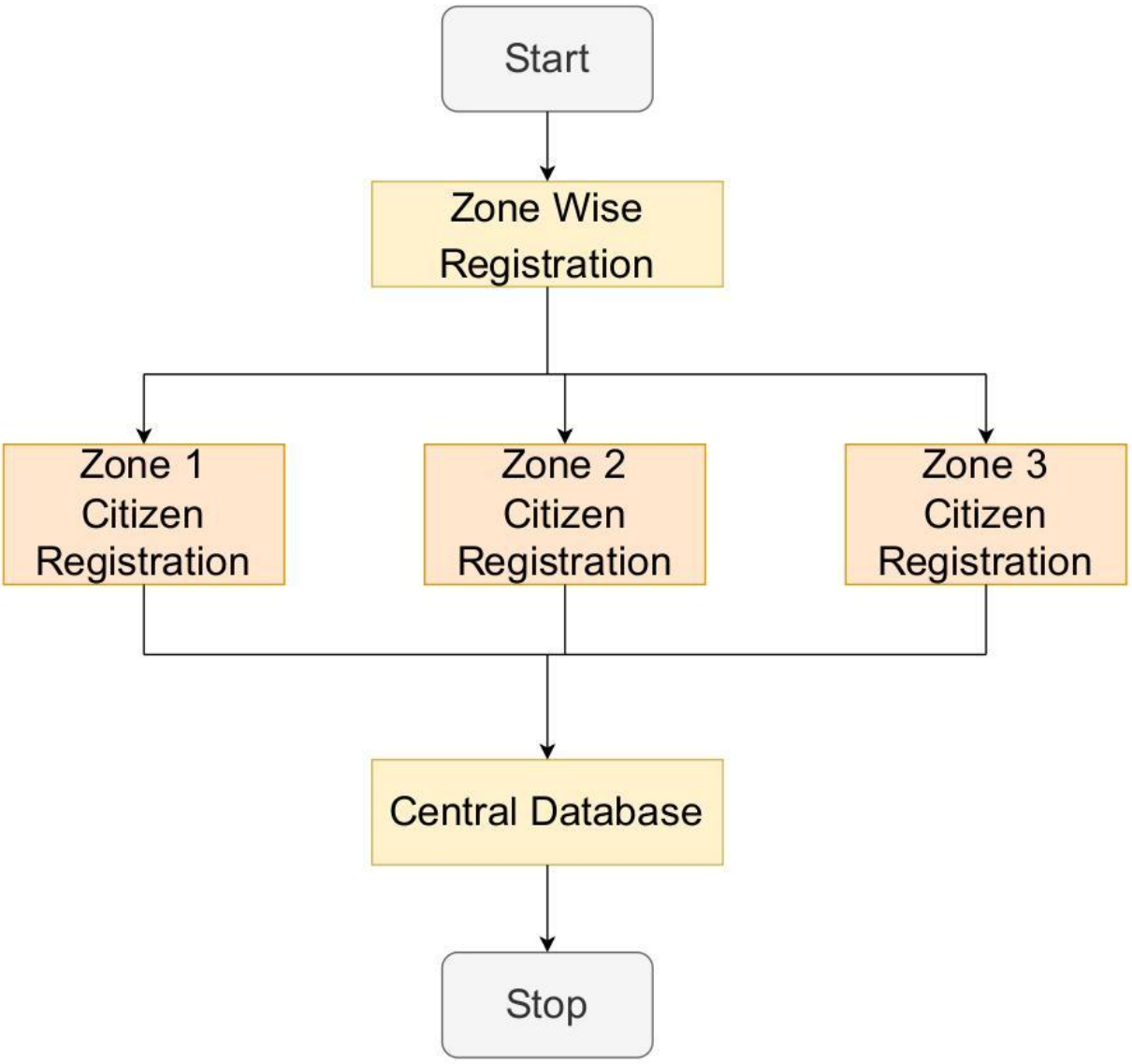
4.2 Sequence Diagram of the application.



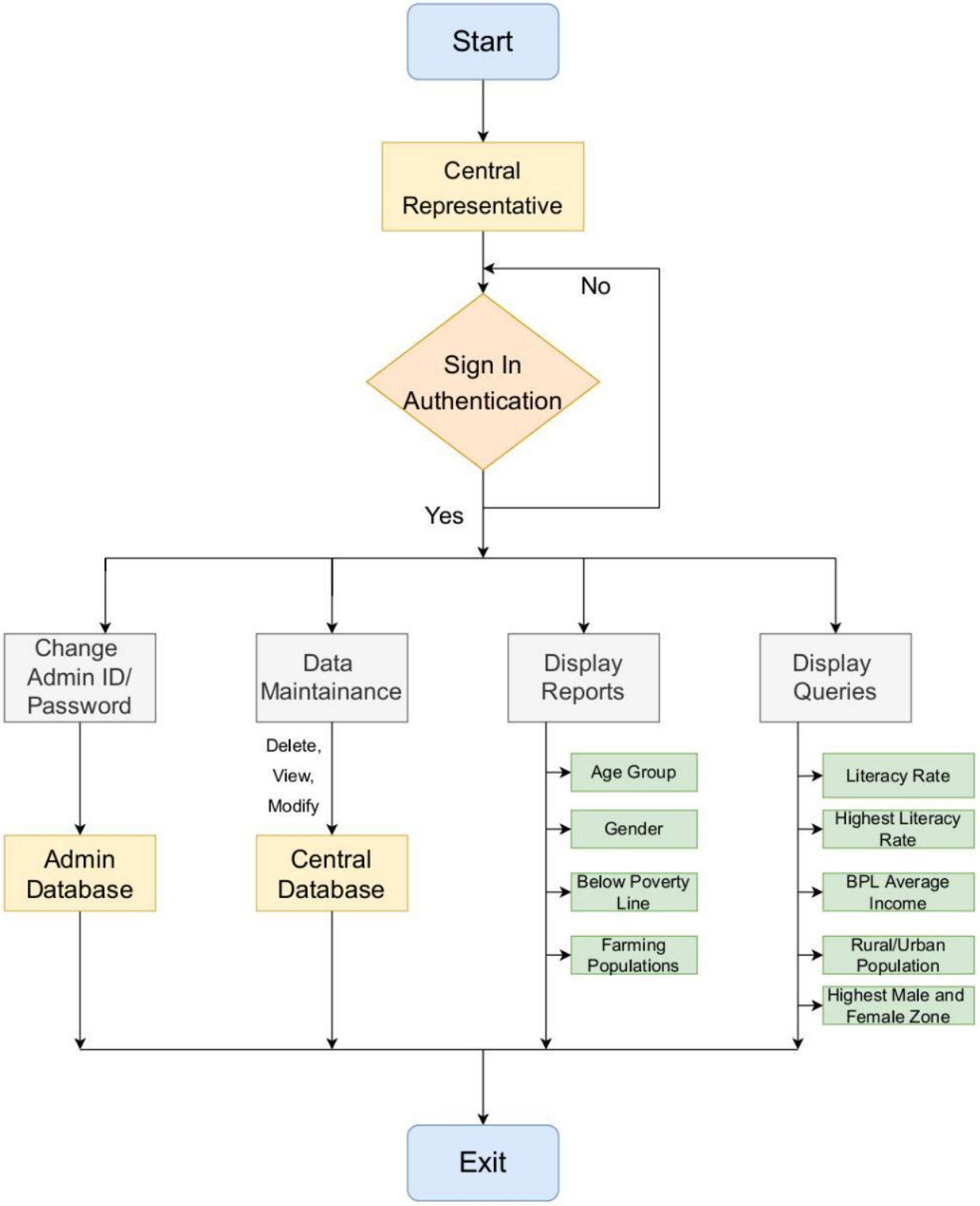
4.3 Entity Relationship Diagram



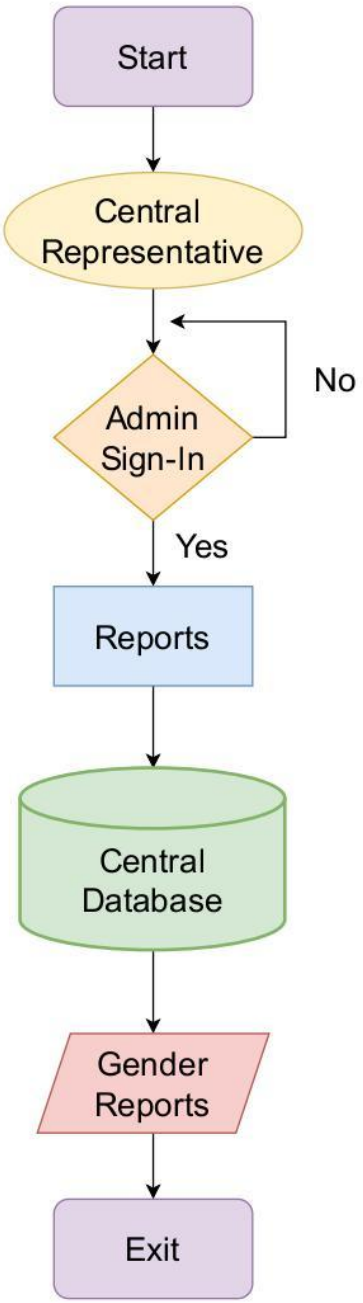
4.4 Use Case Diagram of the Application



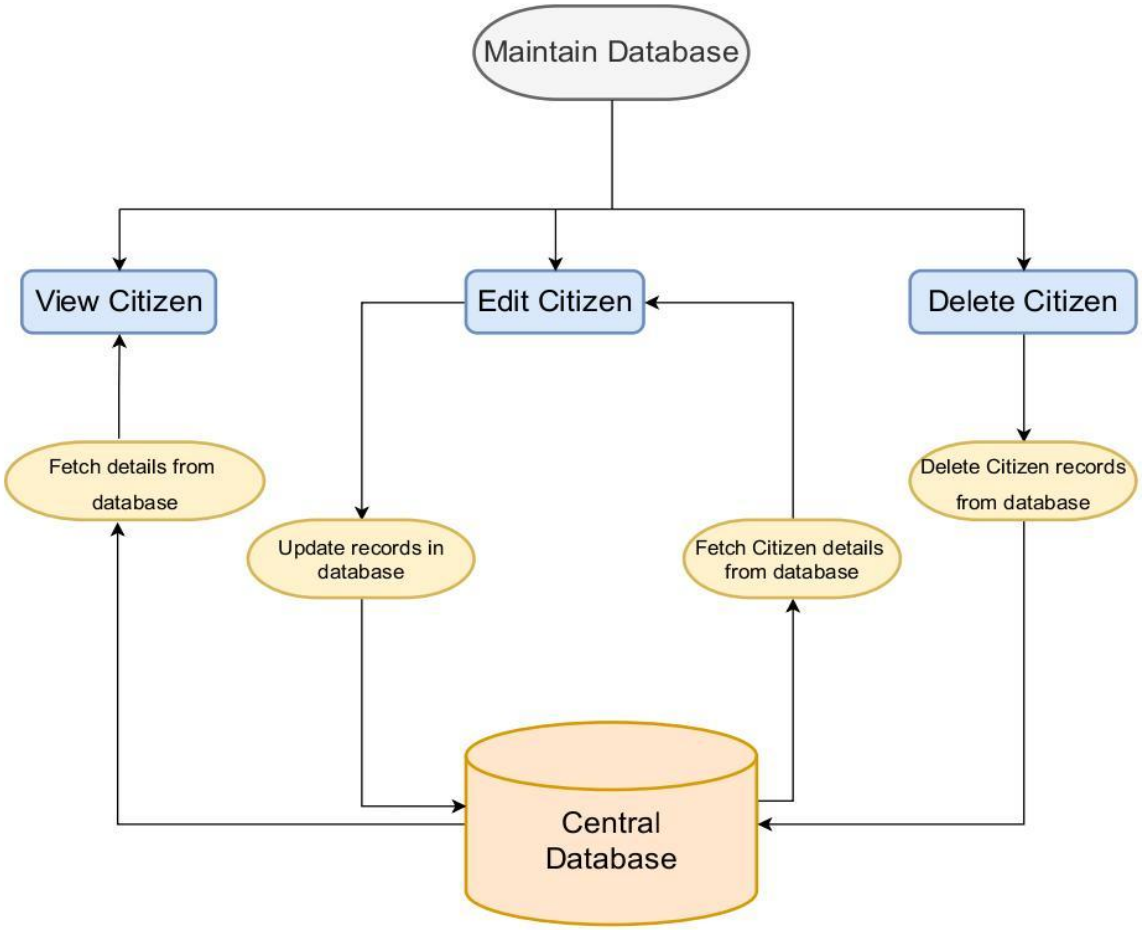
4.5 Flow Chart for Zone Registration function



4.6 Flow chart for Central command of the application



4.7 Flow chart for Gender report function of the application



4.8 Flow chart for Maintain database function of the application

5. TOOLS REPORT

5.1 Strace

% time	seconds	usecs/call	calls	errors	syscall
45.82	0.001048	1048	1		execve
16.62	0.000380	42	9		mmap
6.91	0.000158	52	3		openat
5.51	0.000126	42	3		mprotect
4.55	0.000104	104	1		munmap
3.63	0.000083	16	5		close
3.41	0.000078	19	4		pread64
3.41	0.000078	26	3		newfstatat
2.45	0.000056	18	3		brk
1.66	0.000038	38	1	1	access
1.36	0.000031	15	2	1	arch_prctl
1.05	0.000024	24	1		read
0.92	0.000021	21	1		getrandom
0.74	0.000017	17	1		prlimit64
0.70	0.000016	16	1		rseq
0.66	0.000015	15	1		set_tid_address
0.61	0.000014	14	1		set_robust_list
100.00	0.002287	55	41	2	total

5.2 Valgrind

```
==196968== Memcheck, a memory error detector
==196968== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==196968== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==196968== Command: ./test
==196968==
./test: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./test)
./test: /lib/x86_64-linux-gnu/libstdc++.so.6: version `GLIBCXX_3.4.29' not found (required by ./test)
==196968== Jump to the invalid address stated on the next line
==196968==    at 0x1036: ???
==196968==    by 0x401ACBD: _dl_receive_error (dl-error-skeleton.c:246)
==196968==    by 0x40040CA: dl_main (rtld.c:1912)
==196968==    by 0x4019B3E: _dl_sysdep_start (dl-sysdep.c:252)
==196968==    by 0x4002033: _dl_start_final (rtld.c:485)
==196968==    by 0x4002033: _dl_start (rtld.c:575)
==196968==    by 0x4001097: ??? (in /usr/lib/x86_64-linux-gnu/ld-2.31.so)
==196968==  Address 0x1036 is not stack'd, malloc'd or (recently) free'd
==196968==
==196968==
==196968== Process terminating with default action of signal 11 (SIGSEGV)
==196968==  Bad permissions for mapped region at address 0x1036
==196968==    at 0x1036: ???
==196968==    by 0x401ACBD: _dl_receive_error (dl-error-skeleton.c:246)
==196968==    by 0x40040CA: dl_main (rtld.c:1912)
==196968==    by 0x4019B3E: _dl_sysdep_start (dl-sysdep.c:252)
==196968==    by 0x4002033: _dl_start_final (rtld.c:485)
==196968==    by 0x4002033: _dl_start (rtld.c:575)
==196968==    by 0x4001097: ??? (in /usr/lib/x86_64-linux-gnu/ld-2.31.so)
==196968==
==196968== HEAP SUMMARY:
==196968==     in use at exit: 0 bytes in 0 blocks
==196968==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==196968==
==196968== All heap blocks were freed -- no leaks are possible
==196968==
==196968== For lists of detected and suppressed errors, rerun with: -s
==196968== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

5.3 CPP Check

```
cguser1@instance-1:~/updated_PCA_Project$ cat analysis.out
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      602     0.00     0.00  bool __gnu_cxx::__is_null_pointer<char const>(char const*)
  0.00      0.00     0.00      602     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00      0.00     0.00      602     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::alloca
tor<char> >::_M_construct<char const*>(char const*, char const*, std::forward_iterator_tag)
  0.00      0.00     0.00      602     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<c
har> >::basic_string<std::allocator<char> >(char const*, std::allocator<char> const&)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::difference_type std::__distance<char c
onst*>(char const*, char const*, std::random_access_iterator_tag)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::iterator_category std::__iterator_cate
gory<char const*>(char const* const&)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::difference_type std::distance<char con
st*>(char const*, char const*)
  0.00      0.00     0.00      307     0.00     0.00  __gnu_cxx::__aligned_membuf<Citizen>::_M_ptr()
  0.00      0.00     0.00      307     0.00     0.00  __gnu_cxx::__aligned_membuf<Citizen>::_M_addr()
  0.00      0.00     0.00      307     0.00     0.00  std::_List_node<Citizen>::_M_valptr()
  0.00      0.00     0.00      190     0.00     0.00  std::_List_iterator<Citizen>::_List_iterator(std::__detail::_List_node_ba
se*)
  0.00      0.00     0.00      186     0.00     0.00  std::__cxx11::list<Citizen, std::allocator<Citizen> >::end()
  0.00      0.00     0.00      185     0.00     0.00  std::operator!=(std::_List_iterator<Citizen> const&, std::_List_iterator<
Citizen> const&)
  0.00      0.00     0.00      184     0.00     0.00  Citizen const& std::forward<Citizen const&>(std::remove_reference<Citizen
```

```
 ^
main_menus.cpp:137:17: style: Exception should be caught by reference. [catchExceptionByValue]
                catch(string a)
                ^
main_menus.cpp:151:17: style: Exception should be caught by reference. [catchExceptionByValue]
                catch(string b)
                ^
main_menus.cpp:181:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(string ch)
  ^
main_menus.cpp:254:17: style: Exception should be caught by reference. [catchExceptionByValue]
                catch(string a)
                ^
main_menus.cpp:269:17: style: Exception should be caught by reference. [catchExceptionByValue]
                catch(string b)
                ^
main_menus.cpp:285:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(string ch)
  ^
main.cpp:33:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(string a)
  ^
main.cpp:47:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(string b)
  ^
main.cpp:75:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(string ch)
  ^
class_functions.cpp:14:7: style: The scope of the variable 'flag' can be reduced. [variableScope]
```

```
   BPL++;
   ^
queries_menu.cpp:138:4: error: Uninitialized variable: BPL_total_income [uninitvar]
   BPL_total_income += stod(c.annual_income);
   ^
data_maintainance_menu.cpp:53:18: style: Variable 'li' is assigned a value that is never used. [unreadVariable]
  list<Citizen>li(temp);
                ^
queries_menu.cpp:129:9: style: Variable 'BPL' is not assigned a value. [unassignedVariable]
 double BPL;
        ^
change_menu.cpp:18:71: performance: Prefer prefix ++/-- operators for non-primitive types. [postfixOperator]
   for(list<Authenticate> :: iterator a = aut.begin(); a!= aut.end(); a++)
                                                                      ^
change_menu.cpp:59:71: performance: Prefer prefix ++/-- operators for non-primitive types. [postfixOperator]
   for(list<Authenticate> :: iterator a = aut.begin(); a!= aut.end(); a++)
                                                                      ^
data_maintainance_menu.cpp:12:59: performance: Prefer prefix ++/-- operators for non-primitive types. [postfixOperator]
 for(list<Citizen> :: iterator c=li.begin(); c!=li.end(); c++)
                                                          ^
data_maintainance_menu.cpp:72:59: performance: Prefer prefix ++/-- operators for non-primitive types. [postfixOperator]
 for(list<Citizen> :: iterator c=li.begin(); c!=li.end(); c++)
                                                          ^
file_handling.cpp:156:61: performance: Prefer prefix ++/-- operators for non-primitive types. [postfixOperator]
   for(list<Citizen> :: iterator c=li.begin(); c!=li.end(); c++)
                                                            ^
nofile:0:0: information: Cppcheck cannot find all the include files (use --check-config for details) [missingIncludeSystem]

cguser1@instance-1:~/updated_PCA_Project$
```

5.4 Gcov

Main.cpp.gcov

```
cguser1@instance-1:~/gcov_rel$ cat main.cpp.gcov
        -:    0:Source:main.cpp
        -:    0:Graph:main.gcno
        -:    0:Data:main.gcda
        -:    0:Runs:1
        -:    1:#include<iostream>
        -:    2:#include<string>
        -:    3:#include "main_menus.cpp"
        -:    4:#include "file_handling.cpp"
        -:    5:using namespace std;
        -:    6:
        -:    7:
function main called 1 returned 100% blocks executed 43%
        1:    8:int main()
        -:    9:{
        -:   10:
        -:   11:        //puting the login details like admin and password into list;
        1:   12:        system("clear");
call    0 returned 1
branch  1 taken 1 (fallthrough)
branch  2 taken 0 (throw)
        1:   13:        file_to_list();
call    0 returned 1
branch  1 taken 1 (fallthrough)
branch  2 taken 0 (throw)
        1:   14:        string ch = "a";
call    0 returned 1
call    1 returned 1
branch  2 taken 1 (fallthrough)
branch  3 taken 0 (throw)
        1:   15:        void (*p[2])(){Dynaland_Zone_Wise_Registration, Dynaland_Central_Representative};
        3:   16:        while(ch != "3")
call    0 returned 3
branch  1 taken 3 (fallthrough)
branch  2 taken 0 (throw)
branch  3 taken 3
branch  4 taken 0 (fallthrough)
        -:   17:        {
```

```
branch  8 taken 0 (throw)
call    9 returned 3
branch 10 taken 3 (fallthrough)
branch 11 taken 0 (throw)
        -:   21:
        3:   22:                cin.clear();
call    0 returned 3
branch  1 taken 3 (fallthrough)
branch  2 taken 0 (throw)
        3:   23:                cin.sync();
call    0 returned 3
branch  1 taken 3 (fallthrough)
branch  2 taken 0 (throw)
        3:   24:                getline(cin,ch);
call    0 returned 3
branch  1 taken 3 (fallthrough)
branch  2 taken 0 (throw)
        -:   25:
        -:   26:                try
        -:   27:                {
        3:   28:                        if(ch.length() != 1)
call    0 returned 3
branch  1 taken 0 (fallthrough)
branch  2 taken 3
        -:   29:                        {
    #####:   30:                                throw ch;
call    0 never executed
call    1 never executed
branch  2 never executed
branch  3 never executed
call    4 never executed
call    5 never executed
        -:   31:                        }
        -:   32:                }
    =====:   33:                catch(string a)
branch  0 never executed
branch  1 never executed
call    2 never executed
call    3 never executed
        -:   34:                {
    =====:   35:                                cout<<"Enter a single digit only"<<endl;
```

```
branch  5 taken 0 (throw)
        -:   73:               }
        -:   74:               }
    =====:   75:               catch(string ch)
branch  0 never executed
branch  1 never executed
call    2 never executed
call    3 never executed
        -:   76:               {
    =====:   77:                       cout<<"Only number is allowed!"<<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
    =====:   78:               }
call    0 never executed
call    1 never executed
branch  2 never executed
branch  3 never executed
call    4 never executed
call    5 never executed
        -:   79:
        -:   80:       }
        -:   81:
        1:   82:       list_to_file();
call    0 returned 1
branch  1 taken 1 (fallthrough)
branch  2 taken 0 (throw)
        1:   83:       system("clear");
call    0 returned 1
branch  1 taken 1 (fallthrough)
branch  2 taken 0 (throw)
        -:   84:
        1:   85:}
call    0 returned 1
call    1 never executed
cguser1@instance-1:~/gcov_rel$
```

5.5 Gprof

```
cguser1@instance-1:~/updated_PCA_Project$ cat analysis.out
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      602     0.00     0.00  bool __gnu_cxx::__is_null_pointer<char const>(char const*)
  0.00      0.00     0.00      602     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00      0.00     0.00      602     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::alloca
tor<char> >::_M_construct<char const*>(char const*, char const*, std::forward_iterator_tag)
  0.00      0.00     0.00      602     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<c
har> >::basic_string<std::allocator<char> >(char const*, std::allocator<char> const&)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::difference_type std::__distance<char c
onst*>(char const*, char const*, std::random_access_iterator_tag)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::iterator_category std::__iterator_cate
gory<char const*>(char const* const&)
  0.00      0.00     0.00      602     0.00     0.00  std::iterator_traits<char const*>::difference_type std::distance<char con
st*>(char const*, char const*)
  0.00      0.00     0.00      307     0.00     0.00  __gnu_cxx::__aligned_membuf<Citizen>::_M_ptr()
  0.00      0.00     0.00      307     0.00     0.00  __gnu_cxx::__aligned_membuf<Citizen>::_M_addr()
  0.00      0.00     0.00      307     0.00     0.00  std::_List_node<Citizen>::_M_valptr()
  0.00      0.00     0.00      190     0.00     0.00  std::_List_iterator<Citizen>::_List_iterator(std::__detail::_List_node_ba
se*)
  0.00      0.00     0.00      186     0.00     0.00  std::__cxx11::list<Citizen, std::allocator<Citizen> >::end()
  0.00      0.00     0.00      185     0.00     0.00  std::operator!=(std::_List_iterator<Citizen> const&, std::_List_iterator<
Citizen> const&)
  0.00      0.00     0.00      184     0.00     0.00  Citizen const& std::forward<Citizen const&>(std::remove_reference<Citizen
```

6. Testing

6.1 Unit Testing

```
cguser1@instance-1:~/updated_PCA_Project$ ./unit_test
testdr::test_check_ssid
SSID number length should be 9.
 : assertion
testdr::test_check_genderEnter a valid gender!  ( m : male / f : female / t : transgender )
Enter a valid gender!  ( m : male/ f : female / t : transgender )
 : assertion
testdr::test_check_nameName cannot be blank
Name cannot start with blank space
 : assertion
cppUnit_testing.cpp:39:Assertion
Test name: testdr::test_check_ssid
assertion failed
- Expression: 1==check_ssid("afdfdf54")

cppUnit_testing.cpp:49:Assertion
Test name: testdr::test_check_gender
assertion failed
- Expression: 1==check_gender("4654")

cppUnit_testing.cpp:54:Assertion
Test name: testdr::test_check_name
assertion failed
- Expression: 1==check_name("   ")

Failures !!!
Run: 3   Failure total: 3   Failures: 3   Errors: 0
cguser1@instance-1:~/updated_PCA_Project$
```

6.2 Integration Testing

6.2.1 Add

```
---------  Please select the Zone of Citizen Registration ----------


1. Zone 1 citizen registration
2. Zone 2 citizen registration
3. Zone 3 citizen registration
4. <-Back to Main Menu
1
Enter your ssid (It will be a 9 digit number):
123456789
Enter your name :
shyam
Enter your age :
22
Enter your gender (m : male,  f : female, t : transgender) :
m
Enter your address :
sdf dfsf
Enter your qualification (If none please enter NA):
NA
Enter your occupation :
farmer
Enter your annual income :
546546546
```

6.2.2 Modify

```
Enter the option you want to edit

1. Zone id
2. Name
3. Age
4. Address
5. Qualification
6. Occupation
7. Annual Income
8. Number of Dependants
9. House Type
10. Area Type
11. <-Back
Enter your choice
```

```
Modify Citizen Details
Enter new name :
king

Name is updated successful
```

6.2.3 : View

```
************ Details of the Citizen ***************


Zone id : ZID100
SSID : 123456789
Name : shyam
Age : 22
Gender : male
Address : sdfd df
Qualification : NA
Literate : NO
Occupation : farmer
Annual Income : 46546
Number of Dependents : 4
House Type : own
Area Type : rural
**********************************************
```

6.2.4 Query

```
---------  Select a Query  ----------


1. Literacy Rate of a Dynaland
2. Zone with highest Literacy Rate
3. Below Poverty Line average Income
4. Rural and Urban Population
5. Zone with highest male and female population
6. <-Back
1


Literacy Rate of Dynaland is : 66.6667 %
```

```
---------  Select a Query  ----------


1. Literacy Rate of a Dynaland
2. Zone with highest Literacy Rate
3. Below Poverty Line average Income
4. Rural and Urban Population
5. Zone with highest male and female population
6. <-Back
5
The Zone With Highest percentage of males 100 % is Zone 1
The Zone With Highest percentage of females 100 % is Zone 3
The Zone With Highest percentage of transgender 0 % is Zone 1
```

6.2.5Gender wise report

```
*****************  Gender Wise  Report  *****************

--------------  Zone 1 ----------------

Percentage of male population   : 100 %
Percentage of female population  : 0 %
Percentage of transgender population : 0 %
--------------  Zone 2 ----------------

Percentage of male population   : 100 %
Percentage of female population  : 0 %
Percentage of transgender population : 0 %
--------------  Zone 3 ----------------

Percentage of male  population   : 0 %
Percentage of female  population  : 100 %
Percentage of transgender population : 0 %



--------------  Country Dynaland Gender Wise Report  ----------------

Percentage of male population across country  : 66.6667 %
Percentage of female population across country  : 33.3333 %
Percentage of transgender population across country  : 0 %


**********************************************************************
```

```
****************  Below Poverty line Zone Wise  Report  ****************

--------------  Zone 1 ---------------

Percentage of below poverty line population   : 100 %
--------------  Zone 2 ---------------

Percentage of below poverty line population   : 0 %
--------------  Zone 3 ---------------

Percentage of below poverty line population   : 0 %


--------------  Country Dynaland Below Poverty Line(BPL)  Report  ---------------

Percentage of BPL population across country  : 33.3333 %


*********************************************************************
```

```
*****************  Age Group Report  *****************

--------------  Zone 1 ----------------

Percentage of Population whose age is below 18  : 0 %
Percentage of Population whose age inbetween 18 to 40  : 100 %
Percentage of Population whose age is more than 40  : 0 %



--------------  Zone 2 ----------------

Percentage of Population whose age is below 18  : 0 %
Percentage of Population whose age inbetween 18 to 40  : 100 %
Percentage of Population whose age is more than 40  : 0 %



--------------  Zone 3 ----------------

Percentage of Population whose age is below 18  : 0 %
Percentage of Population whose age inbetween 18 to 40  : 100 %
Percentage of Population whose age is more than 40  : 0 %



--------------  Country Dynaland Age Group Report  ----------------

Percentage of Population whose age is below 18  : 0 %
Percentage of Population whose age inbetween 18 to 40  : 100 %
Percentage of Population whose age is more than 40  : 0 %


**************************************************************************
```

6.2.

```
****************** Changing Admin ID ******************

Enter Admin ID :
shyam
Enter Password :
000000
Enter New Admin ID to change
ram

Admin ID changed Successfully!
```

6.2.7 Validation

```
---------  Please select the Zone of Citizen Registration ----------


1. Zone 1 citizen registration
2. Zone 2 citizen registration
3. Zone 3 citizen registration
4. <-Back to Main Menu
1
Enter your ssid (It will be a 9 digit number):
dfsdfs

SSID number length should be 9.
Enter your ssid (It will be a 9 digit number):

ssid cannot be blank
Enter your ssid (It will be a 9 digit number):
556435215115165

SSID number length should be 9.
Enter your ssid (It will be a 9 digit number):
shyamgoli
SSID should have only numbers.
Enter your ssid (It will be a 9 digit number):
```

7. Requirements Traceability Matrix(RTM)

Req	Design mapping	Code Mapping	UT mapping	IT Mapping
PCA_01	3.1.1.1	Zone1_Citizen_Register()		IT_CASE_03
PCA_02	3.1.1.2	Zone2_Citizen_Register()		IT_CASE_02
PCA_03	3.1.1.3	Zone3_Citizen_Register()		
PCA_04	3.1.2.1	Citizen_Data_Modify()		IT_CASE_05
PCA_05	3.1.2.2	Citizen_Data_Delete()		IT_CASE_04
PCA_06	3.1.2.3	Citizen_Data_View()		IT_CASE_08
PCA_07	3.1.2.4	Age_Group_Report()		
PCA_08	3.1.2.5	Gender_Report()		IT_CASE_09
PCA_09	3.1.2.6	BPL_Zone_Wise_Report()		
PCA_10	3.1.2.7	Farming_Population_Report()		
PCA_11	3.1.2.8	Literacy_Rate()		IT_CASE_10
PCA_12	3.1.2.9	Zone_with_highest_literacy()		
PCA_13	3.1.2.10	BPL_average_income()		
PCA_14	3.1.2.11	Rural_and_Urban_Population()		
PCA_15	3.1.2.12	Zone_with_highest_male_female()		
PCA_16	3.1.3.1	check_ssid()	Test_Case_1	IT_CASE_07
PCA_17	3.1.3.2	check_gender()	Test_Case_2	
PCA_18	3.1.3.4	check_name()	Test_Case_3	
PCA_19	3.1.3.5	check_age()		
PCA_20	3.1.3.6	create_account()		
PCA_21	3.1.4.1	change_Admin_ID()		
PCA_22	3.1.4.2	change_password()		