|  |  |  |
| --- | --- | --- |
| **المملكة العربية السعودية**  **وزارة التعليم**  **جامعة الإمام محمد بن سعود الإسلامية**  **كلية علوم الحاسب والمعلومات** | A description...  **Second term 1441/2020** | **KINGDOM OF SAUDI ARABIA**  **Ministry of Education**  **Al-Imam Mohammad University**  **College of Computer & Information Sciences** |
| **Software Engineering (CS- 310)**  **BSCS- Section: 171**  **Project-Phase No: 2**  **DB Center**  **(Design Document)**  Submitted By | | |
| Ali Khalifaha Alhawas (439015852) – Coordinator  Abdulrahman Mubasher ALghurman (438009996)  Nasser Hamad Alkhurayji (439011631)  Anas Eissa Alzhrani (438011286)  **Nawaf Khalid Aldaham (439011864)**  Supervisor  Dr. Sultan Alqahtani  Date: 12/11/2020 | | |

< DB Center >

Software Development Plan

Version <1.2>

Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| <24/9/20> | <1.0> | <Proposal> |
| <20/10/20> | <1.1> | < Specification > |
| <12/11/20> | <1.2> | <Design > |
|  |  |  |

# Table of Contents

Contents

[1. Table of Contents 3](#_Toc55919679)

[2. Abstract 3](#_Toc55919680)

[3. Introduction 3](#_Toc55919681)

[4. High & medium level Design 4](#_Toc55919682)

[4.1 Use-Case Diagram 4](#_Toc55919683)

[4.2 System architecture 4](#_Toc55919684)

[4.3 Class Diagram 5](#_Toc55919685)

[4.4 Class Method Descriptions 5](#_Toc55919686)

[4.4.1 User Controller 5](#_Toc55919687)

[4.4.2 View 6](#_Toc55919688)

[4.4.3 CitizenView 6](#_Toc55919689)

[4.4.4 EmployeeView 7](#_Toc55919690)

[4.4.5 DBModel 7](#_Toc55919691)

[4.4.6 Citizen 7](#_Toc55919692)

[4.4.7 Employee 7](#_Toc55919693)

[4.4.8 Administrator 8](#_Toc55919694)

[4.5 Flow Chart 9](#_Toc55919695)

[4.6 Sequence diagram 10](#_Toc55919696)

[5. User interface design 12](#_Toc55919697)

[6. Team Members Contributions 14](#_Toc55919698)

[7. Conclusion 15](#_Toc55919699)

# **Abstract**

This is a design document that will describe our project in detail of design architecture, design model, class hierarchy, and how the sub-system will communicate with each other as a whole system. It will include three chapters introduction, design include system architecture and system design and the last chapter will be the conclusion. This document will give a good understanding of how the system will be like.

# Introduction

This section gives a scope description and overview of everything included in this DD document,

Also, the purpose of this document is described and a list of abbreviations and definitions is

Provided.The purpose of this document is to provide a complete description of the requirements for the ”DataBase Center ”(DB-Center), This document is primarily intended to be submitted to a customer for its approval  
and a source for developing the first version of the system for the development team, The development team thought it will speed up the process and save time and money for the government to have an online system that saves, store and keep track of the Citizens information online. the DB Center will allow the following Functionality online: To Search a citizen, to display a report about a citizen, To update citizen information, To create a DataBase. the DB Center is expected to help the government by lowering the average time for a citizen to do a single task and to reduce the number of employees and the budget.

# High & medium level Design

## Use-Case Diagram

The use case diagram is a representation of user interaction with the system.

Diagram

Description automatically generated

Figure 1: Use Case Diagram

## System architecture

The system architecture that we found fit our project inconvenient way is the MVC pattern. Is Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user.



Figure 2: System Architecture

## Class Diagram

The class diagram shows the structure of SATS in program units.

Diagram

Description automatically generated

Figure 3:Class Diagram

## Class Method Descriptions

In this section, each class method is briefly described with respect to visibility, parameters, and return type.

### User Controller

The Controller class interacts with the Module class and the View class. The Controller class controls all actions performed on the system and passes these interactions to the View and the Module.

|  |  |
| --- | --- |
| **Class** | UserController |
| **Method** | getID():int |
| **Visibility** | Public |
| **Return Type** | Int |
| **Parameters, Type** | N/a |
| **Description** | This method returns the Id of the user. |

|  |  |
| --- | --- |
| **Class** | UserController |
| **Method** | srtAddress |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | Address: String |
| **Description** | This method applies the user to change the address. |

|  |  |
| --- | --- |
| **Class** | UserController |
| **Method** | getAddress |
| **Visibility** | Public |
| **Return Type** | String |
| **Parameters, Type** | N/a |
| **Description** | This method returns the address. |

### View

The View class is the superclass of the two classes CiitzenView and EmployeeView, the view class purpose is to just view or display .

|  |  |
| --- | --- |
| **Class** | View |
| **Method** | ViewRelatedCitizenInformatio |
| **Visibility** | public |
| **Return Type** | Void |
| **Parameters, Type** | N/a |
| **Description** | This method display related information about the citizen |

### CitizenView

The CitizenView class will display the citizen's information to them self’s.

|  |  |
| --- | --- |
| **Class** | CitizenView |
| **Method** | ViewOwnProfile |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | N/a |
| **Description** | This method View citizen information about his/her self. |

### EmployeeView

The CitizenView class will display the citizen's information to the employee.

|  |  |
| --- | --- |
| **Class** | EmployeeView |
| **Method** | ViewCitizenProfile |
| **Visibility** | public |
| **Return Type** | Void |
| **Parameters, Type** | NationalIdNumber:int |
| **Description** | This method applies the employee to see information about the citizen’s |

### DBModel

The DBModule class is used to add or check citizen's data. The class DBModule is manipulated by the class Controller who has one attribute of type Module. The class DBModule has methods to update citizen’s information, and methods to check certain information.

|  |  |
| --- | --- |
| **Class** | DBModel |
| **Method** | VerifyLogIn |
| **Visibility** | public |
| **Return Type** | Bool |
| **Parameters, Type** | N/a |
| **Description** | This method returns the user status. |

### Citizen

The Citizen class maintains the states and attributes of a citizen, Every citizen is saved on the database and controlled by the UserController class

|  |  |
| --- | --- |
| **Class** | Citizen |
| **Method** | UpdateMinorINformation |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | Address: String |
| **Description** | This method applies the citizen to change minor information. |

### Employee

The employee class maintains the states and attributes of a citizen, Every Employee can update some information about the Citizens.

|  |  |
| --- | --- |
| **Class** | Employee |
| **Method** | UpdateHealthRecord |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | HealthRecord:String ,NationalIdNumber:int |
| **Description** | This method applies the employee to update the health Record of a citizen. |

|  |  |
| --- | --- |
| **Class** | Employee |
| **Method** | UpdateAddress |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | Address:String ,NationalIdNumber:int |
| **Description** | This method applies the employee to update the Address of a citizen. |

|  |  |
| --- | --- |
| **Class** | Employee |
| **Method** | UpdateMedicine |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | Medicine:String ,NationalIdNumber:int |
| **Description** | This method applies the employee to update the Medicine of a citizen. |

|  |  |
| --- | --- |
| **Class** | Employee |
| **Method** | UpdateMedicineDosage |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | MedicineDosage:String ,NationalIdNumber:int |
| **Description** | This method applies the employee to update the Medicine Dosage of a citizen. |

|  |  |
| --- | --- |
| **Class** | Employee |
| **Method** | UpdateBalance |
| **Visibility** | Public |
| **Return Type** | Void |
| **Parameters, Type** | Balance:int ,NationalIdNumber:int |
| **Description** | This method applies the employee to update the Balance of a citizen. |

|  |  |
| --- | --- |
| Class | Employee |
| Method | UpdateCriminalRecord |
| Visibility | Public |
| Return Type | Void |
| Parameters, Type | CriminalRecord:bool ,NationalIdNumber:int |
| Description | This method applies the employee to update the Criminal Record status of a citizen. |

|  |  |
| --- | --- |
| Class | Employee |
| Method | UpdateDrivingRecord |
| Visibility | Public |
| Return Type | Void |
| Parameters, Type | DrivingRecord:String ,NationalIdNumber:int |
| Description | This method applies the employee to update the Driving Record of a citizen. |

### Administrator

The Administrator class maintains the states and attributes of an Administrator, Every Administrator can update and change all information about the Citizens it got entered wrong.

|  |  |
| --- | --- |
| Class | Administrator |
| Method | UpdateBloodType |
| Visibility | Public |
| Return Type | void |
| Parameters, Type | BloodType:String , NationalIdNumber:int |
| Description | This method applies the employee to update the Blood Type of a citizen. |

|  |  |
| --- | --- |
| Class | Administrator |
| Method | UpdateHealthRecord |
| Visibility | Public |
| Return Type | Void |
| Parameters, Type | age:String ,NationalIdNumber:int |
| Description | This method applies the employee to update the Age of a citizen. |

## Flow Chart

This flow chart explains the system in detail.

Diagram

Description automatically generated

Figure 4: FlowChart

## Sequence diagram

 simply depicts interaction between objects in a sequential order the order in which these interactions take place.

Table

Description automatically generated

Figure : Sequence diagram

# User interface design

When a user enters our website they will choose ether which type of user are they Citizen, Executive, or Citizen Services figure 6 User Type page then they will go to the login page figure 7 login page after they enter the username and password the website will move them to the next page like the following,

if they are a citizen the website will move them to their page and they can just edit general information about them figure 9Citizen profile for the Citizen.

If they are Executive the web-site will move them to the search page and they can search by a citizen ID number and they will see information about the citizen

If they are citizen service employees the web-site will move them to the search page and they can search by a citizen ID number and they will see Specific information depend on where they work in figure 8 Search function & Citizen profile for citizen service.

Graphical user interface, diagram, application

Description automatically generated

Figure 6: Uesr Type Page

Graphical user interface, application

Description automatically generated

Figure 7: LogIN Page

Graphical user interface, text, application

Description automatically generated

Figure 8: citizen profile for the citizen

Graphical user interface, text, application, email

Description automatically generated

Figure 9: search function & citizen profile citizen service

# Team Members Contributions

|  |  |
| --- | --- |
| **Person** | **Roles and Responsibilities** |
| Ali Alhawas, Coordinator | Class diagram & Class Method Descriptions  Use case Diagram & Abstract |
| Abdulrahman Mubasher Alghurm | User interface & Flow Chart |
| Nasser Hamad Alkhurayji | User interface & Class diagram |
| Anas Eissa Alzhrani | system’s architectural pattern |
| Nawaf Khalid Aldaham | sequence diagram |

# Conclusion

Conclusion In conclusion, our DD gives full knowledge of the software in Medium-level, high-level, class hierarchy, and the system architecture. We provide an abstraction, introduction that helps to understand the DD and the DBCenter. We also include class diagram, use cases, sequence diagram, methods, and classes description to approach the software correctly and as requirement needs