Chester Descallar

Unit 15: Object Oriented Programming  
Task 2

Contents

[Purpose 2](#_Toc495944953)

[Requirement Specification 3](#_Toc495944954)

[Class Responsibility Collaboration Card 4](#_Toc495944955)

[Class Diagram 6](#_Toc495944956)

[Prison Management System Pseudocode 11](#_Toc495944957)

# Purpose

HM Prison Stafford has asked me to design an application – this application is a prison management system that will manage prisoners and wardens within the prison. The system must store specific details of each prisoner including unique identifier. This application must also be able to do the following:

* Appoint a governor
* Appoint wardens
* Add prisoners

The application could also be improved by implementing some of these features:

* Assigning wardens to a wing
* Assigning prisoners to a wing
* Allowing prisoners to be released
* Listing all wardens for each wing
* Listing all prisoners for each wing

# Requirement Specification

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirement | Description |
| 1 | Create a prison management application | The prison management application must allow the user to manage wardens and prisoners within the prison. |
| 2 | Create a menu for the application | The menu must allow the user to select an option by inputting something such as pressing 1 to appoint a governor. |
| 3 | Provide an option to appoint a governor | In the main menu, when the user inputs the key for this option, it should send them to the area that allows them to appoint a governor. |
| 4 | Provide an option to appoint a head warden | In the main menu, when the user inputs the key for this option, it should send them to the area that allows them to appoint a head warden. |
| 5 | Provide an option to appoint wardens | In the main menu, when the user inputs the key for this option, it should send them to the area that allows them to appoint a warden. |
| 6 | Provide an option for the governor’s area (allows assignment of warden to a wing) | In the main menu, when the user inputs the key for this option, it should send them to the governor’s area – in this area, the user will be able to assign the head warden to a wing |
| 7 | Provide an option for the head warden’s area | In the main menu, when the user inputs the key for this option, it should send them to the head warden’s area – in this area, the user will be able to assign the other wardens to a wing. In here, you can also display / list all the wardens for each wing and release a prisoner. |
| 8 | Provide an option for the warden’s area | In the main menu, when the user inputs the key for this option, it should send them to the warden’s area – in this area, the user will be able to assign the prisoners to a wing. In here, you can also display / list all of the prisoners for each wing |
| 9 | Provide an option for help | When the user presses the designated key for ‘Help’ it should send them to a help area e.g. texts that answers the most frequently asked questions. |

|  |  |
| --- | --- |
| Class: Governor | |
| *Responsibilities* | *Collaboration* |
|  |  |
|  |  |
|  |  |
|  |  |

# Class Responsibility Collaboration Card

|  |  |
| --- | --- |
| Class: Warden | |
| *Responsibilities* | *Collaboration* |
| Know the prisoner details | Prisoners , person profile |
| Carry out cell inspections | Cells |
| Sets out a regime | Prisoners |

|  |  |
| --- | --- |
| Class: Prisoner | |
| *Responsibilities* | *Collaboration* |
| Know their work duty | Wardens |
| Know what their exercise plan is | Wardens |
| Know their dining times | Wardens |
| Follow their regime | Wardens |
| Know their conviction, prison sentence and release date | Wardens, Head Wardens, Governor |
| Know their ID | Wardens, Head Wardens, Governor |
| Know which wing they are in | Wings, Wardens, Head Wardens, Governor |
| Know which cell they are in | Cells, Wardens, Head Wardens, Governor |

|  |  |
| --- | --- |
| Class: Wing | |
| *Responsibilities* | *Collaboration* |
| Meets health and safety standard | Warden, Head Warden |
| List the amount of prisoners in that wing | Prisoners, Head Warden |
| Know how many cells in a wing | Cells |
| Know the limit of prisoners in a wing | Prisoners, Head Warden |
| Know the available cells | Cells |

|  |  |
| --- | --- |
| Class: Governor | |
| *Responsibilities* | *Collaboration* |
| Know the warden details | Warden, Person Profile |
| Know the prisoner details | Prisoner, person profile |
| Supervise the wings and wardens | Wings and wardens |
| Carry out ‘wing’ inspections | Wings |
| Know head warden details | Head Warden, Person profile |
| Appoint head wardens | Head Warden |
| Make sure prisoner limit is met | Wings |
|  |  |

|  |  |
| --- | --- |
| Class: Cell | |
| *Responsibilities* | *Collaboration* |
| Ensure that the prisoners remain in their cells | Prisoners |
| List of each prisoner in that cell | Prisoner |
| Know how many prisoners can fit in a cell | Prisoner, Head Warden |
| Know how many prisoners are currently in the cell | Prisoner |

|  |  |
| --- | --- |
| Class: Head Warden | |
| *Responsibilities* | *Collaboration* |
| Appoints how long a prisoner stays in the cell | Prisoners, Governor |
| Allocate release time | Governor |
| Assigns a prisoner to a wing | Prisoner, Wing |
| Assigns a prisoner to a cell | Prisoner, Cell |
| Know the prisoner details | Prisoner, Personal Detail |
| Allocate time for parole and visitations | Prisoners, Governor |

|  |  |
| --- | --- |
| Class: Personal Profile | |
| *Responsibilities* | *Collaboration* |
| Add the first name and surname | Prisoners  Wardens  Governor  Wing manager |
| Add the age and date of birth |
| add a gender |
| Add a role in prison |
| Add address |
| Add contact details |
| Add medical history |

# Class Diagram

|  |
| --- |
| Class: Governor |
| Properties:  Prisoner details string  Wing manager details string  Warden details string  Prison Budget integer |
| Methods:  setGovernor()  removeGovernor()  assignHeadWarden()  removeHeadWarden()  viewWardenDetails()  viewHeadManagerDetail()  viewPrisonerDetail()  viewBudget() |

|  |
| --- |
| Class: Head Warden |
| Properties:  Prisoner Cell Room integer  Parole Time time  Warden Wing Boolean |
| Methods:  setPrisonerRecord()  getPrisonerRecord()  assignWardenWing()  assignPrisonertoWing()  getPrisonerWing()  assignPrisonertoCell()  getPrisonerCell()  setPrisonerParole()  getPrisonerParole()  releasePrisoner()  transferPrisoner()  setHeadWardenDetail()  getHeadWardenDetail() |

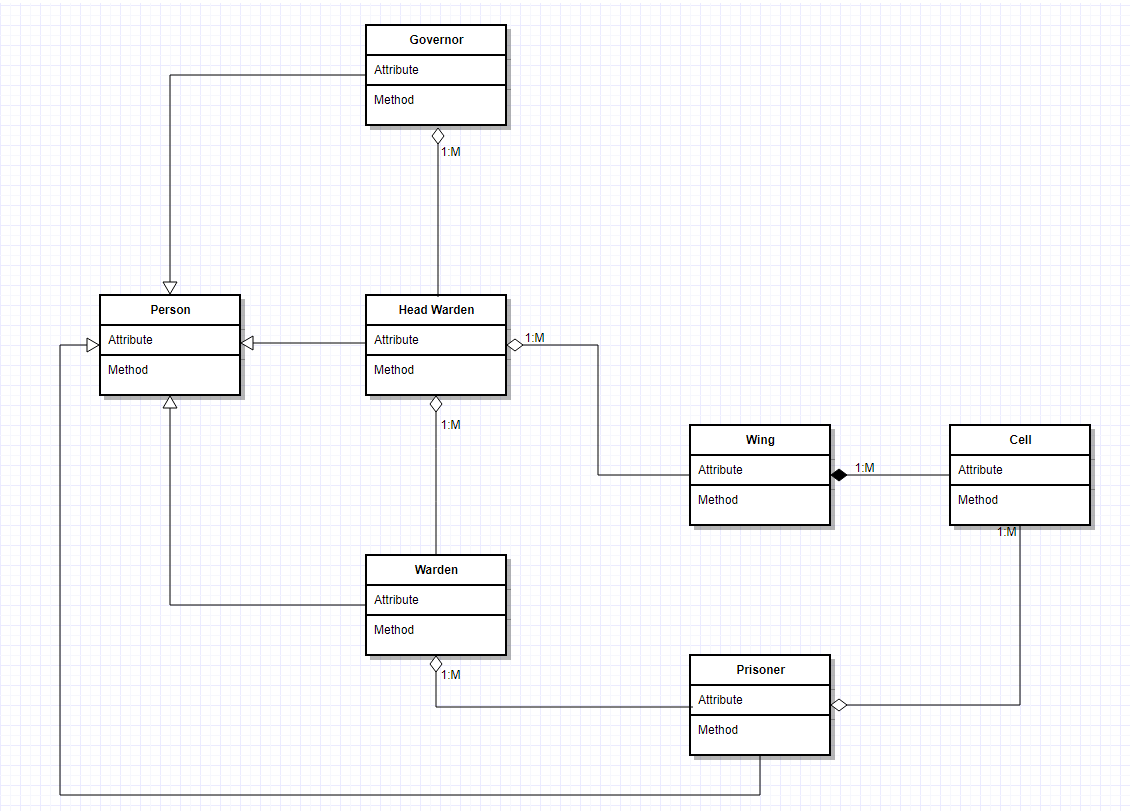
|  |
| --- |
| Class: Warden |
| Properties:  Work Duty Regimen string  Exercise Regimen string  Dining Regimen string  Inmates infractions string |
| Methods:  setWorkDutyRegimen()  getWorkDutyRegimen()  setExericeRegimen()  getExerciseRegimen()  setDiningRegimen()  getDiningRegimen()  setWardenDetails()  getWardenDetails()  setInmateInfractions()  getInmateInfractions() |

|  |
| --- |
| Class: Wing |
| Properties:  Wing area (A/B) Boolean  Number of cells integer  Prisoner Limit integer  Available cells Boolean |
| Methods:  setWingName()  getWingName()  setNumberofCells()  viewNumbersofCells()  setPrisonerLimit()  getPrisonerLimit()  viewAvailableCells() |

|  |
| --- |
| Class: Cell |
| Properties:  Prisoners in cell integer  Cell number integer  Cell capacity integer |
| Methods:  viewPrisonersinCell()  setCellNumber()  viewCellNumber()  setCellCapacity()  viewCellCapacity() |

|  |
| --- |
| Class: Prisoner |
| Properties:  Unique ID string  Prison Sentence integer  Crime or conviction string  Criminal Record string  Release Date date/time |
| Methods:  setPrisonerID()  getPrisonerID()  setPrisonerConviction()  getPrisonerConviction()  setPrisonerSentence()  getPrisonerSentence()  setCriminalRecord()  getCriminalRecord()  setReleaseDate()  getReleaseDate() |

|  |
| --- |
| Class: Person Profile |
| Properties:  Name string  Surname string  Gender string  Date of Birth date/time  Address string  Contact Number integer  Email address string  Role in prison string |
| Methods:  addName()  removeName()  addSurname()  removeSurname()  addGender()  removeGender()  addDateOfBirth()  removeDateOfBirth()  addAge()  removeAge()  addAddress()  removeAddress()  addContactNumber()  removeContactNumber()  addEmailAddress()  removeEmailAddress ()  addRoleInPrison()  removeRoleInPrison() |



The person class is a class that can be used by both the employee and the prisoners of prison. This class contains properties such as name, address, gender, etc. hence why it has the association relationship ‘People’ (Governor, Head Warden, Warden, Prisoner) class as it is where they get their personal details from.

The governor, head warden and warden are in an aggregation relationship because all three can exist without one of the other.

The wing class has an aggregation relationship with the governor and head warden because the wing can still exist even without the head warden and the governor. Similarly, the prisoner class and the cell class also has an aggregation relationship because there will still be prisoners even without a cell.

However, the wing class and the cell class has a composition relationship – this is because without a wing the cells would not exist.

# Prison Management System Pseudocode

START

//Class Person

setPersonalDetails()

{

Create list (List<person> newPerson= new List <Person>)

newPerson.Add {

Set First Name

Set surname

Set gender

Set DOB

Set address

Set contactno

Set emailadd

Set Role

}

}

//Class Governor : Person

setGovDetails()

{

Create list( List<governor> newGovernor = new List <governor>)

newGovernor.Add {

Ask for user input (Governor ID)

setPersonalDetails()

}

//Class Head Warden : Person

setHeadWardDetails()

{

Create list (List<headW> newHeadWar = new List <headW>)

newHeadWar.Add {

Set Head Warden ID

setPersonalDetails()

}

}

//Class Warden : Person

setWardenDetails()

{

Create list (List<Warden> newWarden = new List <Warden>)

newWarden.Add {

Set Warden ID

SetPersonalDetails()

}

//Class Prisoner : Person

setPrisonerDetails()

Create list (List<Prisoner> newPrisoner = new List <Prisoner>)

newPrisoner.Add{

Set Prison ID

setPersonalDetails()

//Class Wing

Create list for Wing A ((List<Wing> WingA = new List <Wing>)

Create list for Wing B ((List<Wing> WingB = new List <Wing>)

Int wingACells = 40

Int wingBCells = 30

Int wingMaxA = 120

Int wingMaxB = 60

//Class Cell : Wing

cellAMax = 3

cellBMax = 2

//Main Menu

Output (“Press 1 to Appoint Governor”)

Output (“Press 2 to Appoint Head Warden”)

Output (“Press 3 to Appoint Warden”)

Output (“Press 4 to go to Governor’s area”)

Output (“Press 5 to go to Head Warden’s area”)

Output (“Press 6 to Warden’s area”)

Output (“Press 7 for Help”)

Output (“Press 8 to Exit”)

userchoice = “ ”

running = 1

while running == 1

if userchoice == “1”;

Get setGovDetails() to run

(This should ask the user to input the governor’s details)

End if

Else if userchoice == “2”;

Get setHeadWardDetails() to run

(This should ask the user to input the head warden’s details)

End if

Else if userchoice == “3”;

Get setWardenDetails() to run

(This should ask the user to input the warden’s details)

EndIF

Else if userchoice == “4”;

Output (“Press 1 to Assign Head Warden to a wing”)

Output (“Press 2 to Display Head Warden in wing”)

Userchoice2 = “”

If userchoice2 == “1”;

Ask for head warden id

Set Head Warden Wing

End if

Else if userchoice2 == “2”;

For each headwarden in newHeadWar

Output headwarden

End if

Else if userchoice == “5”;

Output (“Press 1 to Assign Warden to a wing”)

Output (“Press 2 to Display Warden in wing”)

Output (“Press 3 to Release Prisoner”)

Userchoice3 = “”

If userchoice2 == “1”;

Ask for warden id

Set Head Warden Wing

End if

Else if userchoice2 == “2”;

For each warden in newWar

Output warden

End if

Else if userchoice4 == “3”;

Ask for prisoner id

Remove data from list

End if

End If

Else if userchoice == “6”;

Output (“Press 1 to Display prisoners in Wing”)

Output (“Press 2 to Add a Prisoner”)

Output (“Press 3 to Assign Prisoner to Wing”)

Userchoice5 = “”

If userchoice5 == “1”;

For each prisoner in newPrisoner

Output prisoner

End if

If userchoice3 == “2”;

Get setPrisonerDetails() to run

End if

Else if userchoice4 == “3”;

Ask for prisoner id

Set Prisoner Wing

End if

Else if userchoice == “7”;

Output Help text

End IF

Else if userchoice == “8”;

Exit Program

End IF

END