

AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

International Credit Hours Engineering Programs (i.CHEP)



AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING- ICHEP

Agile Software Engineering

CSE233

Submitted to

Dr Salwa Osama Selim

Eng Rana Sobhy

Submitted by

Team 14

Team Members

Name	ID
Alaa Mohamed Mohamed Hamdy	19P6621
Anas Salah Abdelrazek SaadEldin	19P9033
Ahmed Amr MohyEldin ElGayar	19P8349
Nada Amr Attia Abd El Hamed	19P1621
Jannah Ayman	19P1728

Table of Contents

1.0	INTRODUCTION.....	4
2.0	CUSTOMER MEETING REPORT	4
2.1	First Meeting.....	4
2.1.1	Organization overview	4
2.1.2	Objective	4
2.1.3	Project Overview	4
2.1.4	Project Requirements	4
2.2	Second Meeting.....	5
3.0	OUR JOURNEY	8
3.1	Prototype.....	8
3.2	Database	21
3.2.1	4 Core Key Classes.....	21
3.2.2	Dynamic Messages	22
3.2.3	Singleton	23
3.2.4	Schema.....	24
3.3	MVC	25
3.4	Jira.....	30
3.5	Good User Story.....	36
3.6	Multilevel Planning	39
3.7	Story Mapping.....	43
3.8	Trello.....	45
3.9	Workflow.....	47
3.10	GitHub Project	51
3.11	Clean Code.....	54
3.12	Additional Part	60

1.0 INTRODUCTION

We are students in the faculty of engineering; we decided to make an application for a nonprofit charity to help them get benefit of new technologies in helping others. We used agile method specially the scrum approach in our journey to finish this application. In this report we will go through each phase in our journey.

2.0 CUSTOMER MEETING REPORT

2.1 First Meeting

Our Client is a member in organization "معانا لإنقاذ الإنسان", our client need to apply shelter for homeless and he needed the booking system for this shelter to be automated. The following sections clarified the main points which we discussed.

2.1.1 Organization overview

نشأت فكرة مؤسسة معانا من رحم الشارع المصري بأيدي شباب مصرى حينما رأى المهندس محمود وحيد رئيس مجلس الإدارة مسن مشرد يتحلل وهو على قيد الحياة ومن هنا بدأ الفكرة بشباب لديهم فكر واعي للنهوض بمصرنا الحبيبة بأجمل اشكال التقدم الحضاري، هؤلاء الشباب وضعوا نصب اعينهم ظاهره لم يسلط عليها أي ضوء من قبل وهى ظاهرة المشردين من المسنين كبار السن وجاء ذلك من تفاقم عدد المشردين في الشوارع خصوصاً في السنوات الأخيرة فقاموا على الفور بالتعاون مع الجهات المعنية لتأسيس كيان قانوني ينبع بـإشراف الحكومة المصرية والمعنية به وزارة التضامن الاجتماعي

2.1.2 Objective

Homelessness in Egypt is a significant social issue affecting almost 12 million people in the country. According to UNICEF, there are 1 million children living on the streets in Egypt. Other researchers estimate the number to be some 3 million. Homelessness NGOs assisting street children include those such as Hope Village Society, and NAFAS. Other NGOs, such as Plan International Egypt, work to reintegrate street children back into their families.

2.1.3 Project Overview

The main aim of this project is to provide temporary homes for the homeless people in Egypt, providing them with the basic human needs and give them the feeling of worthiness, support and care. This project will be able to deliver a decent living condition, aiding them to take a step forward in their lives.

2.1.4 Project Requirements

Design a desktop application that manages booking system for rooms provided by Ministry of Housing, Utilities and Urban Communities. The system allocates guests to rooms according to the availability while recording the information of both room conditions and residents for future bookings. The booking process initiates when a guest requests to stay, An employee request the national ID of the guest and assigns bed, in the case where there is no available beds the guest is relocated to the nearest building containing available bed. A

bed is assigned to the guest for a fixed period of time. The system allows the building admins to asses room conditions and guest leave. The general admin can add buildings , rooms , beds , employees to the system database

2.2 Second Meeting

In this meeting we talk more deeply about the idea of our application

Our main 3 Modules

- Authentication
- Admin Dashboards, Building Admin Dashboards, Floor Admin Dashboard, Receptionist Dashboard
- Reservation

Key project stakeholders

- Admin
- Building Admins
- Floor Admins
- Receptionists

Functional Requirements (Admin)

- Add, Delete, Modify buildings.
 - o Location
 - o Floors
 - o Rooms
 - o Beds
- Add, Delete, Modify Building Admins
 - o Name
 - o Date of birth
 - o Address
 - o Salary
 - o Contact Info.
 - o Building Assigned
 - o Shifts

- Add, Delete, Modify Floor Admins

- o Name
- o Date of birth
- o Address
- o Salary
- o Contact Info.
- o Building Assigned
- o Floor Assigned
- o Shifts

- Add, Delete, Modify Receptionists

- o Name
- o Date of birth
- o Address
- o Salary
- o Contact Info.
- o Building Assigned
- o Shifts

Functional Requirements (Building Admin)

- Add, Delete, Modify Floor Admins

- o Name
- o Date of birth
- o Address
- o Salary
- o Contact Info.
- o Floor Assigned
- o Shifts

- Add, Delete, Modify Receptionists

- o Name
- o Date of birth
- o Address

- Salary
 - Contact Info.
 - Shifts
- View Reservations

Functional requirements (Floor Admins)

- View Reservations
- Rate Resident's Stay
- Update Room Status

Functional Requirements (Receptionists)

- Collect Resident Info.
 - National ID
- Reserve Beds
 - Choose room / bed
 - Confirm reservation
- View alternative building occupancy

General Functions

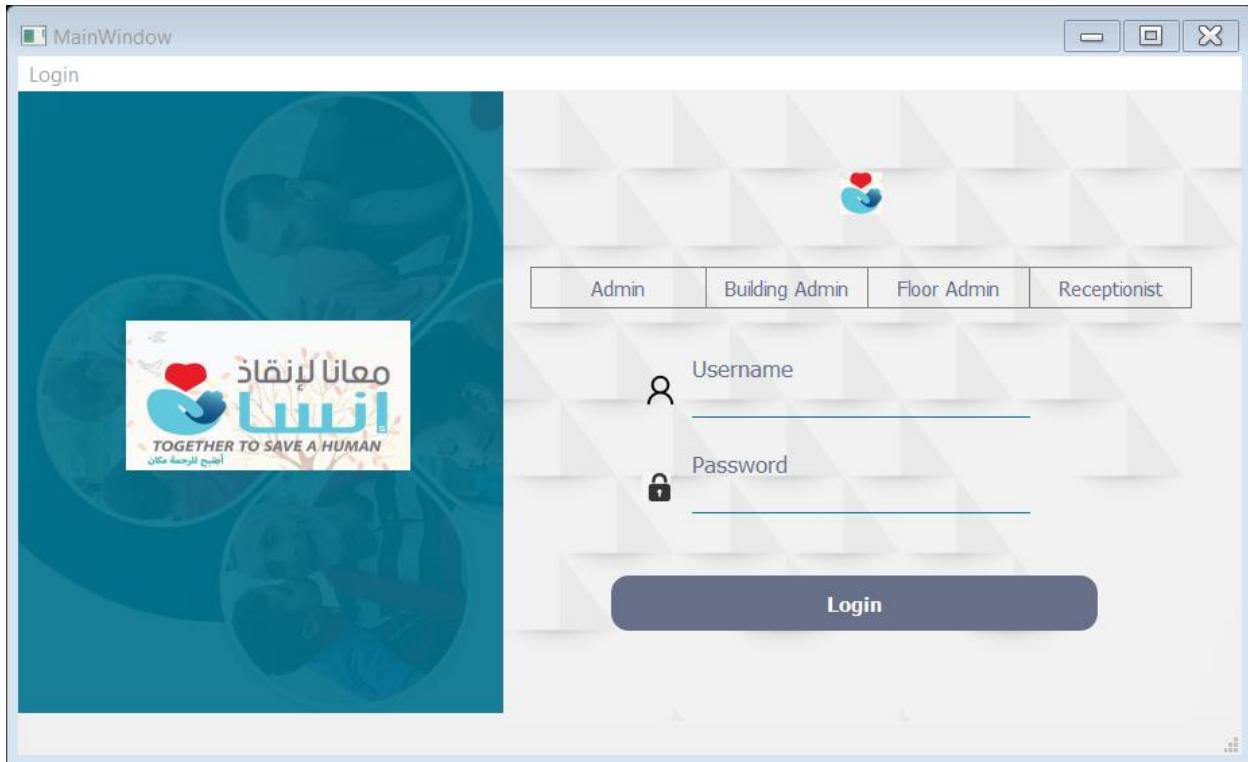
- Login
- Log out

3.0 OUR JOURNEY

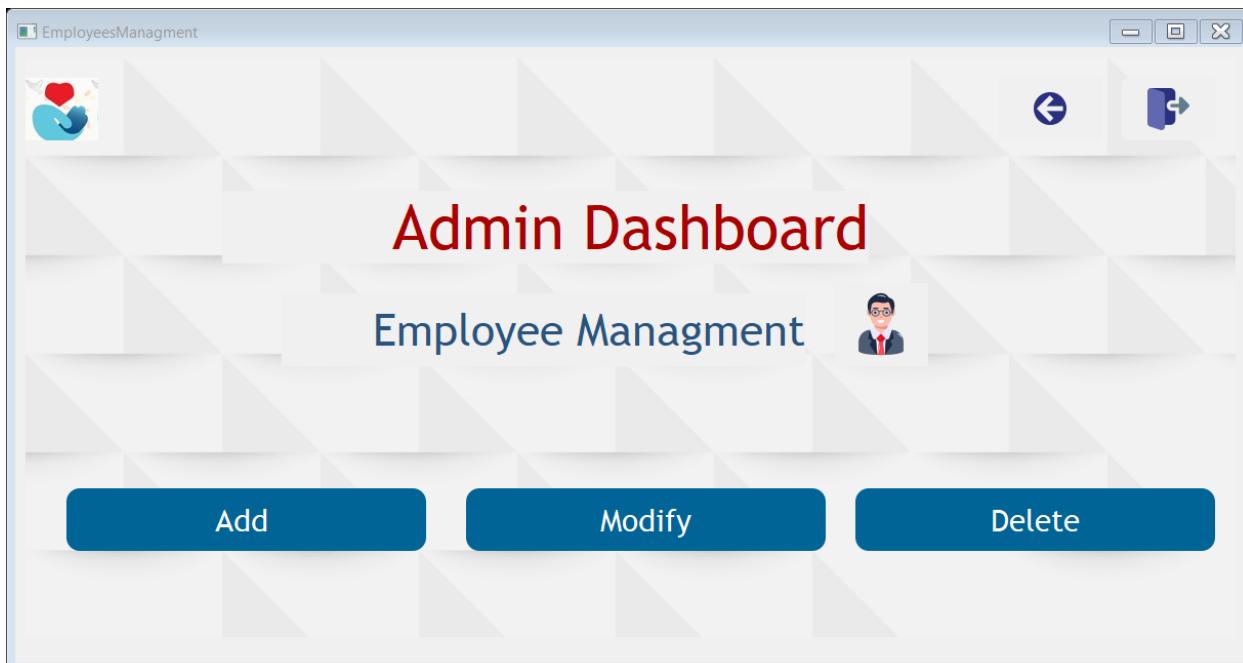
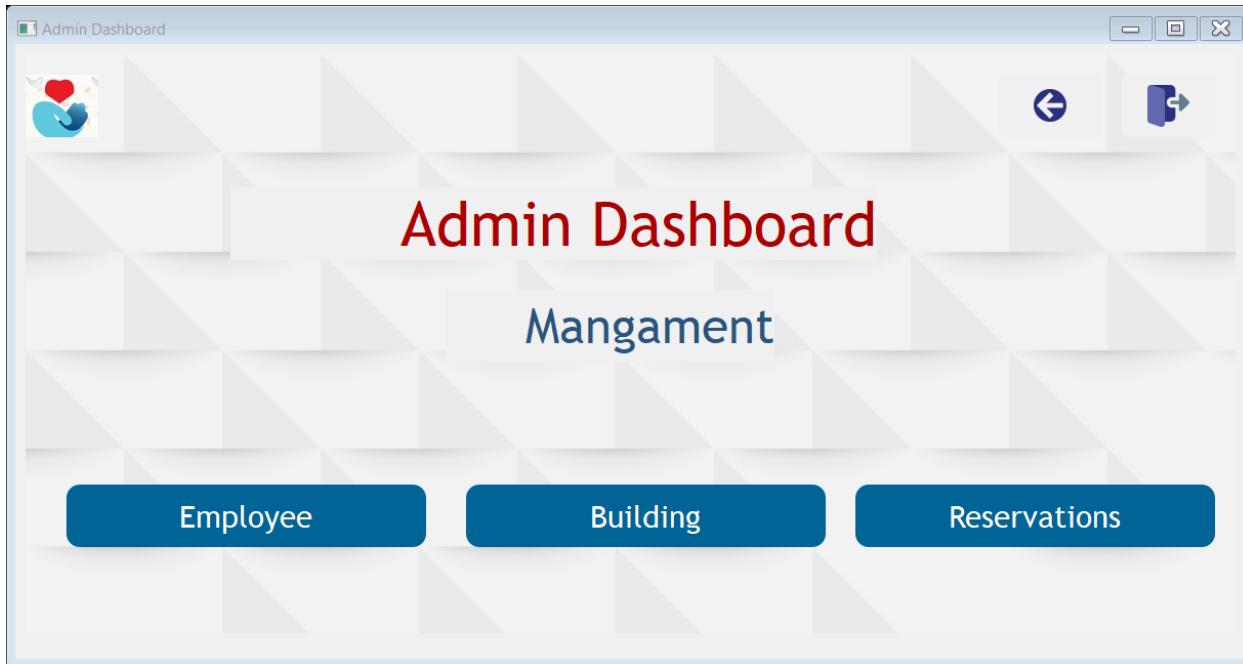
3.1 Prototype

The client asks that he needs the GUI to be plain and very clear and easy to use.

- Authentication



- Dashboards
 - o Admin Dashboard



addNewEmployee



Add Employee



Id Salary **Admin Id**

Name Contact Info Building Admin Id

Address Date of Deployment Building Id

Data of Birth

Add

modifyEmployee



Modify Employee



Show **Id** Salary **Admin Id**

Name Contact Info Building Admin Id

Address Date of Deployment Building Id

Data of Birth

Modify

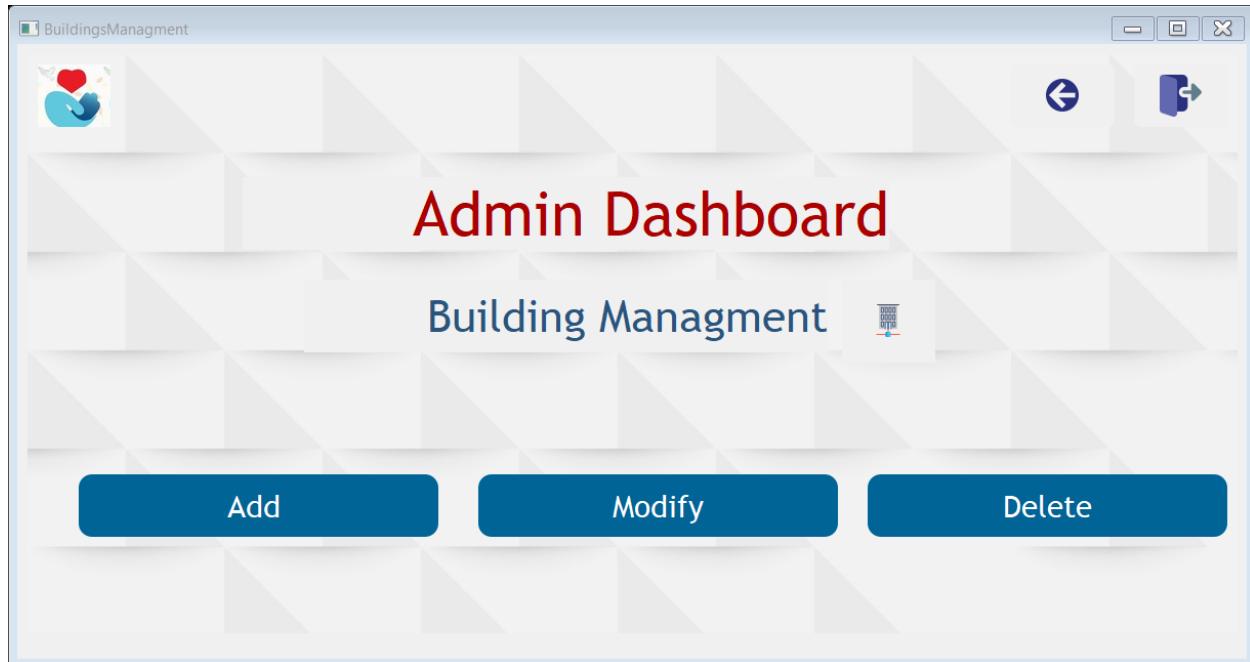
deleteEmployee



Delete Employee



Id <input type="text"/> 	Salary <input type="text"/>	Admin Id <input type="text"/> 
Name <input type="text"/>	Contact Info <input type="text"/>	Building Admin Id <input type="text"/> 
Address <input type="text"/>	Date of Deployment <input type="text"/>	Building Id <input type="text"/> 
Data of Birth <input type="text"/>	Delete	

The screenshot shows the "Add New Building" form. It has two columns of input fields. The left column contains: "Id" with a numeric input field, "Location" with a location pin icon, "Admin Id" with a numeric input field, and "Total Floors" with a numeric input field. The right column contains: "Total Rooms" with a room icon, "Total Beds" with a bed icon, and "Floors Gender" with a gender icon. At the bottom right is a large blue "Add" button.

 modifyBuilding

Modify Building

 Id Show

 Location

 Admin Id

 Total Floors

Total Rooms 

Total Beds 

Floors Gender 

Modify

 deleteBuilding

Delete Building

 Id Show

 Location

 Admin Id

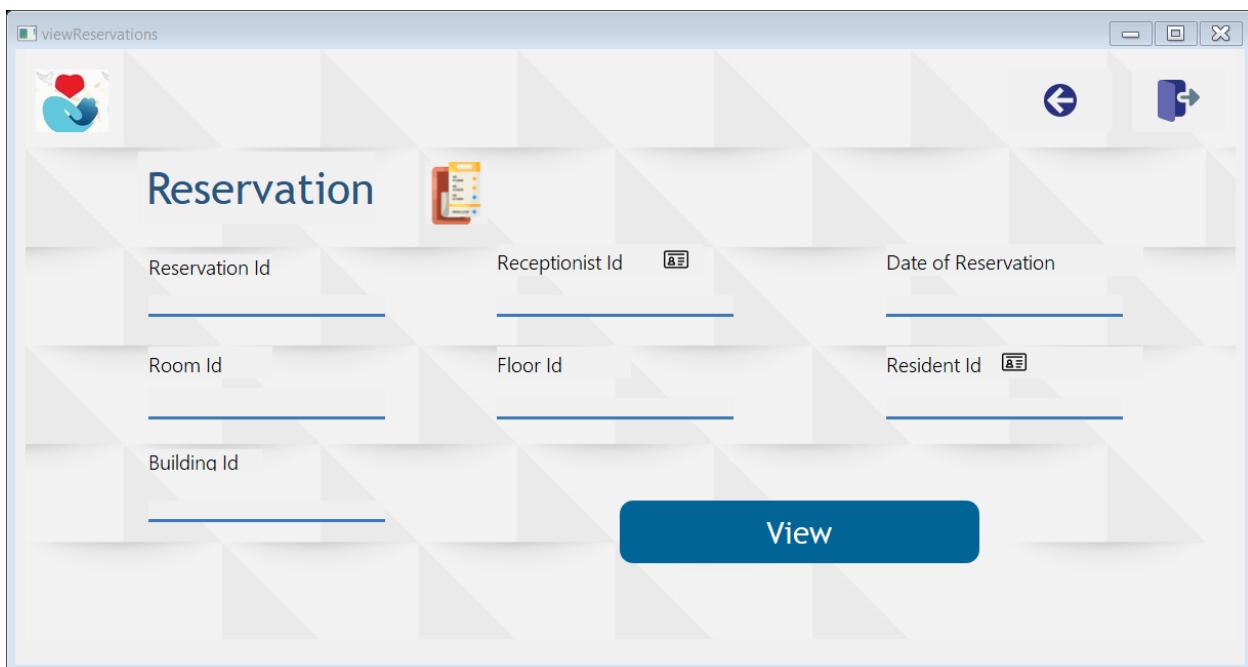
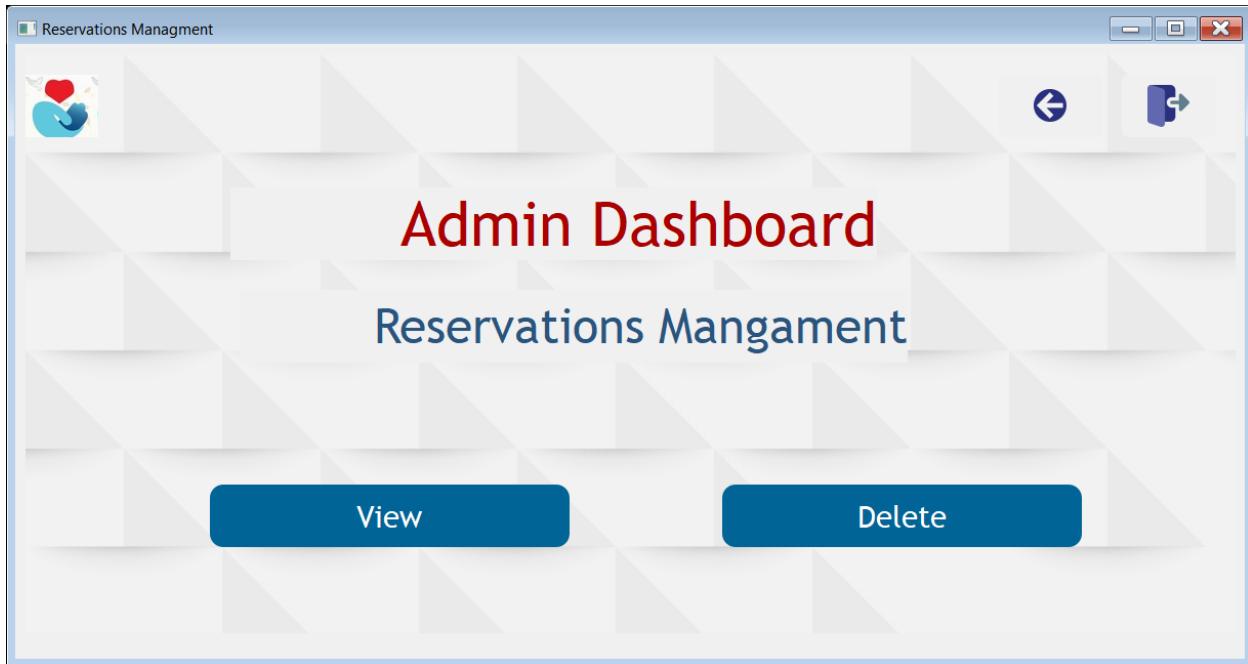
 Total Floors

Total Rooms 

Total Beds 

Floors Gender 

Delete



viewReservations



Reservation

Reservation Id Show Receptionist Id Date of Reservation

Room Id Floor Id Resident Id

Building Id

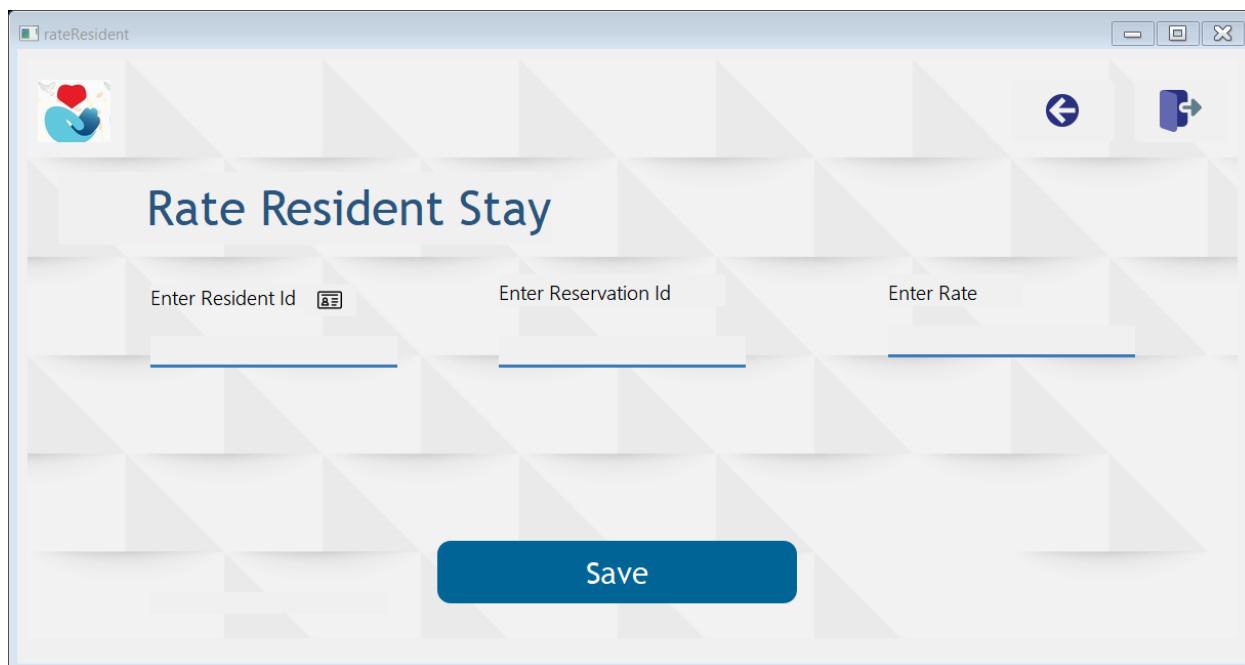
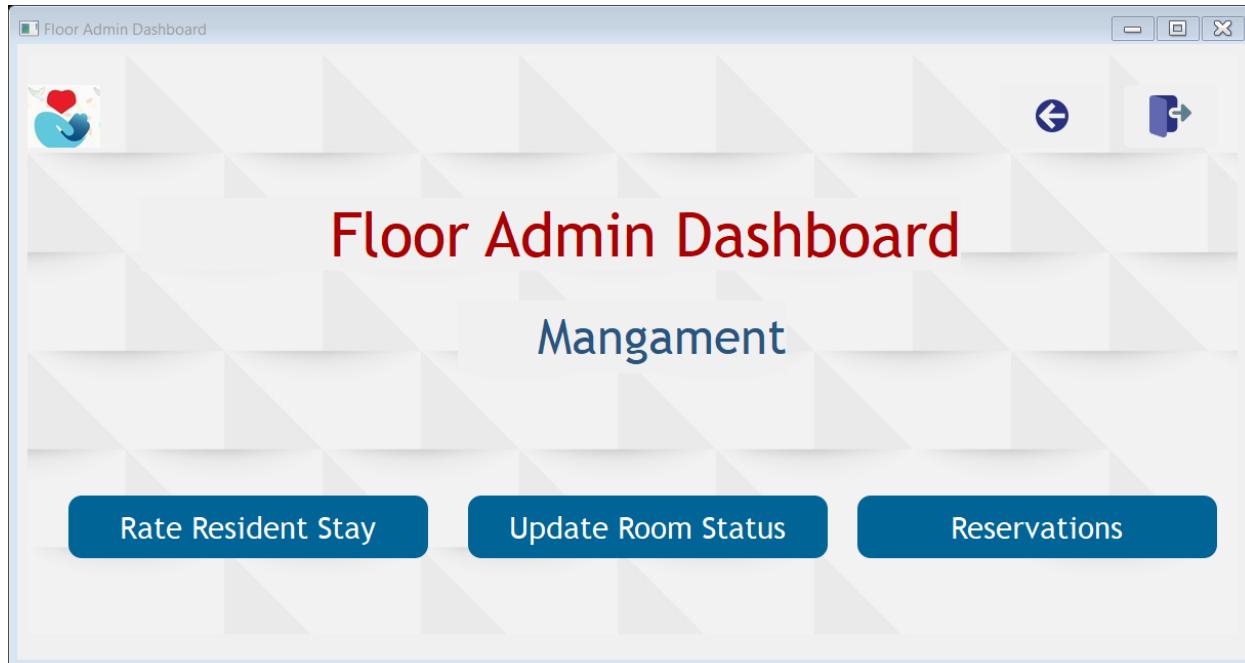
Delete

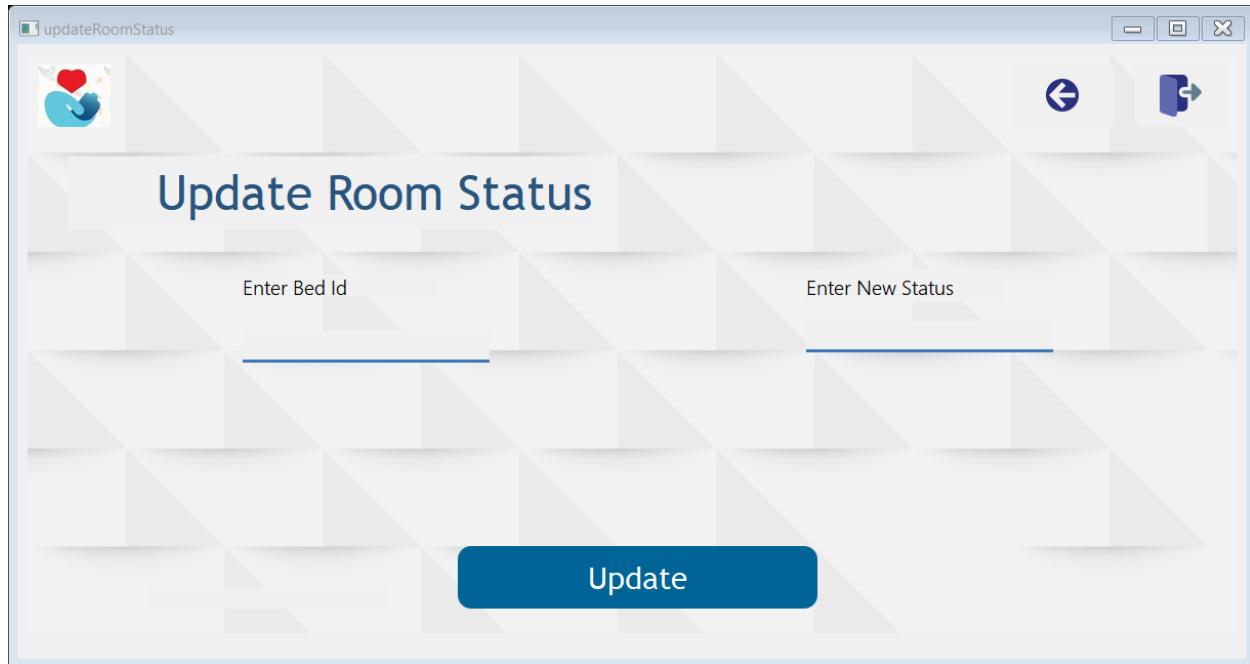
- o Building Admin Dashboard



Note: The 3 pages for add, modify, delete the same as the admin ones, Also The reservation.

- Floor Admin Dashboard





Note: The reservation page the same as the admin ones

- Reservation “Receptionist Dashboard”

Resident Info Success



Resident Info

Enter the resident's National ID

National ID: Anas Salah

Show

Name	Gender
Anas Salah	Male

Age: 21

Eligible: Yes

Last Stay: 5 days ago

Book

Building Selection



Building Occupance

Building ID: D

Location: الكورنٰ، مصر الجديدة 21

Beds Available: D

Book Bed

Name	Gender
Anas Salah	Male

Age: 21

Eligible: Yes

Last Stay: 5 days ago

Find Alternative

Reservation

Bed Reservation

Building Id

Choose Floor

Gender ♂

Available Beds 🛌

Confirm

This form is titled "Bed Reservation". It contains four input fields: "Building Id", "Choose Floor", "Gender" (with a male icon), and "Available Beds" (with a bed icon). A large blue "Confirm" button is centered at the bottom.

viewReservations

Reservation

Reservation Id

Receptionist Id

Date of Reservation

Room Id

Floor Id

Resident Id

Building Id

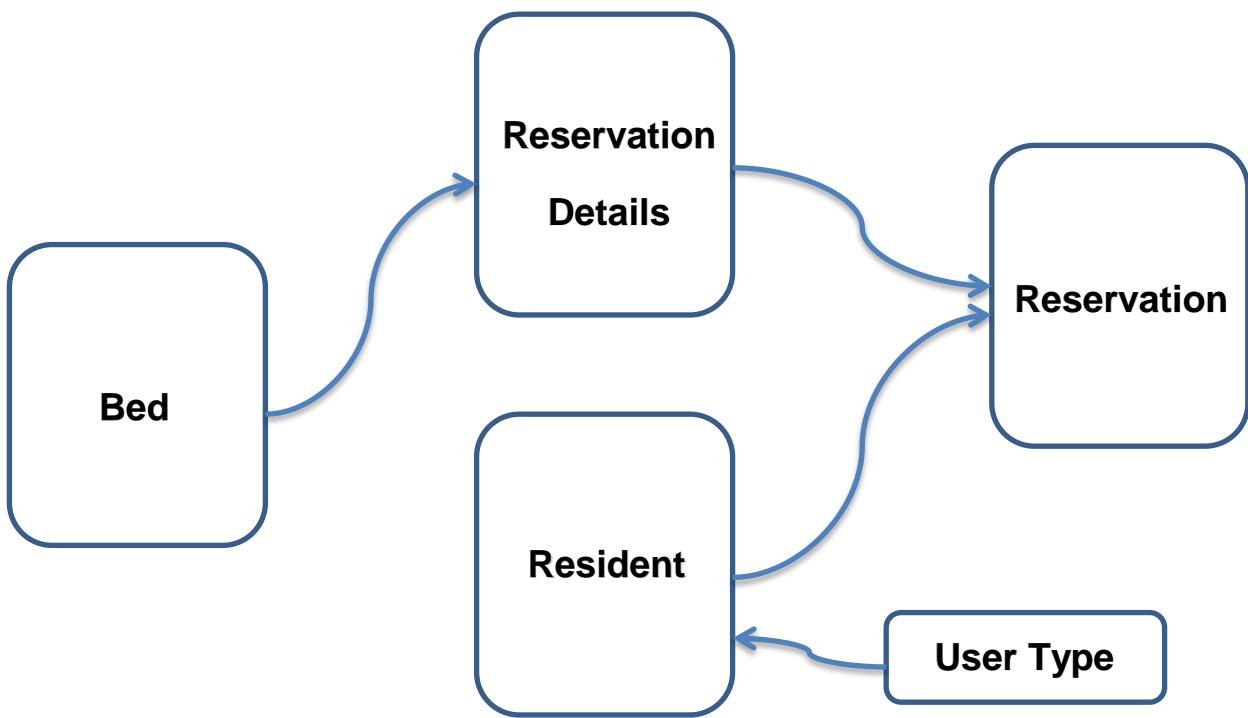
Done

This form is titled "Reservation". It contains six input fields: "Reservation Id", "Receptionist Id" (with a clipboard icon), "Date of Reservation", "Room Id", "Floor Id", and "Resident Id" (with a clipboard icon). There is also a separate input field for "Building Id". A large blue "Done" button is centered at the bottom.

3.2 Database

3.2.1 4 Core Key Classes

- Resident
- Reservation
- Bed
- reservation details



3.2.2 Dynamic Messages

We created a table in our database to save the used messages of our code in it, just in case we need to modify the message we do it in the database without affecting the code.

- Example of these messages:

- 1 Invalid login, please choose user type and try again
- 2 Invalid login, please try again with correct credentials
- 3 Please enter both username and password
- 4 Failed to add, check that information is complete and try again
- 5 added succesfully!
- 6 failed to add, error:
- 7 Please enter an id
- 8 Updates succesfully!
- 9 id does not match, no such record
- 10 user is not eligible to stay

gui_msgs	
gui_msgs_id	INT
msg	VARCHAR(45)
Indexes	

- Example in Code:

```
def deleteEmployee(self):  
    mc = main_control()  
    result = mc.getGuiMsgsControl()  
    if (self.idLineEdit.text() == ''):  
        messagebox.showwarning('Warning', str(result[6]))  
    return  
    isacceptable = mc.deleteEMployeeControl(self.idLineEdit.text())  
    if str(isacceptable).lower() == "none":  
        messagebox.showinfo('information', str(result[7]))
```

3.2.3 Singleton

We override the `__new__(cls)` function which is a built in function for creating new instances for a class, in our overridden function, we check over a variable called `_instances` which tracks the instances of the objects for the class, if there are no instances in a class it creates one using the `__init__` functions which calls the `singleton_connection()` function which creates the connection with the database and returns, if there were already any single instance of the class, no new instance is created and instead references the first and only instance which stores the connection making it a singleton connection

```
def __init__(self):
    self.connection = singleton_connection()

#we override the __new__() method that is called internally by Python when you create an object of a class
#then check the instances, if an object already exists do not create a new instance, hence singleton
def __new__(cls):
    if cls._instance is None:
        print('Creating the object')
        cls._instance = super(main_control, cls).__new__(cls)
    return cls._instance

#call model to get building ID
def getBuildingIDControl(self,id):...

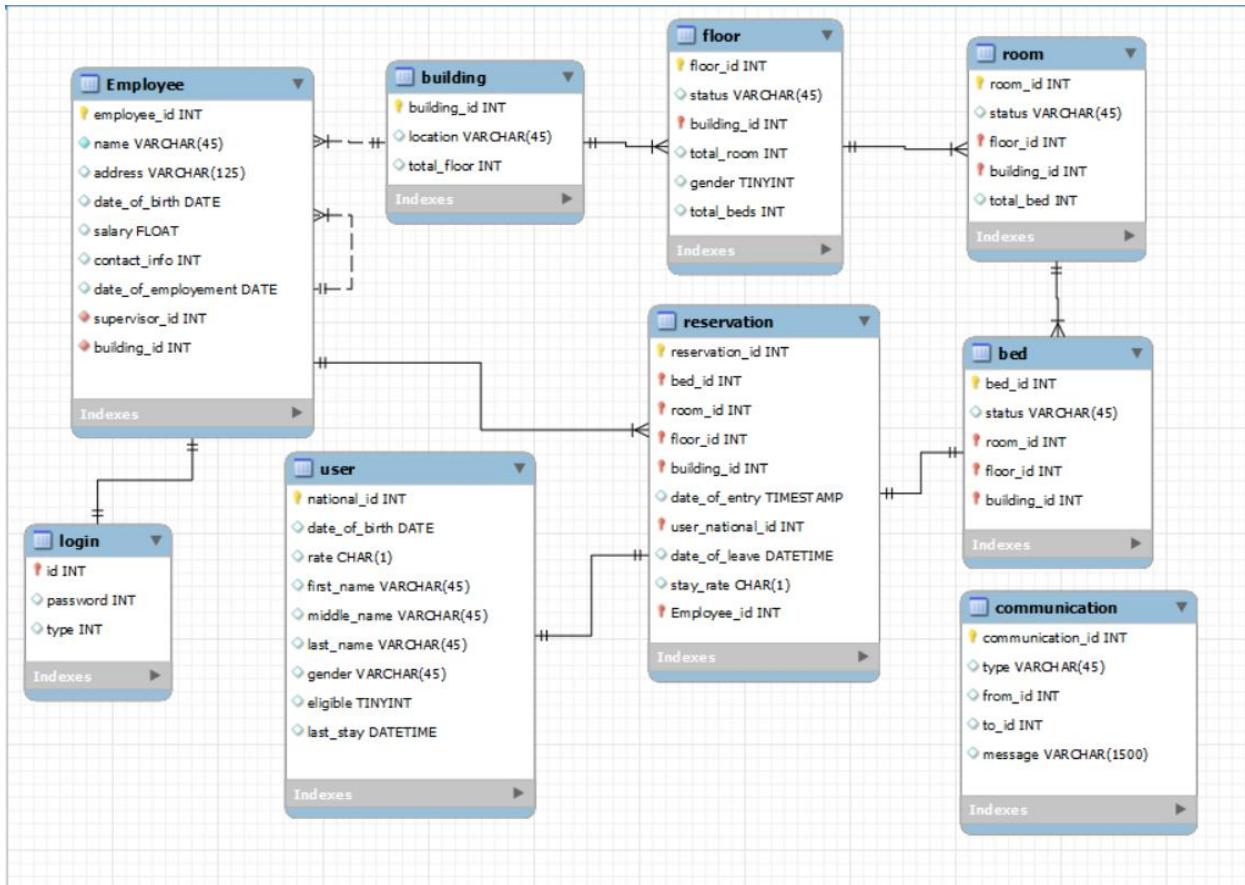
#call model to make sure building id exists in db buildings
def buildingIDCheckControl(self):...

# call MODEL to query login info and authenticate info
def checkloginControl(self,type , usertuple):...

# call MODEL to query adding info and authenticate insertion
def addEmployeeControl(self, usertuple):...
```

```
def singleton_connection():
    try:
        print('conn')
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="1234",
            database="agile_project")
        print(connection.is_connected())
        return connection
    except Error as e:
        print(e)
```

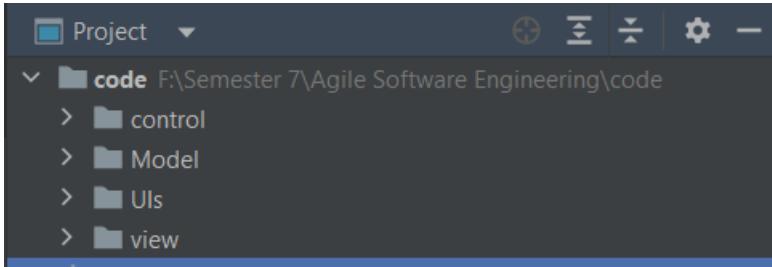
3.2.4 Schema



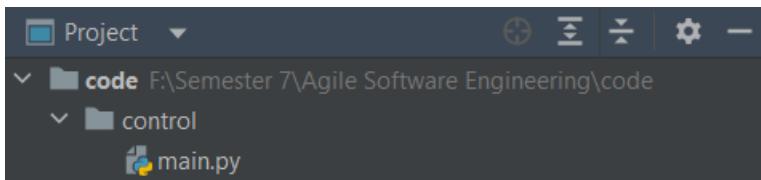
3.3 MVC

In our code we use MVC Architecture in our code

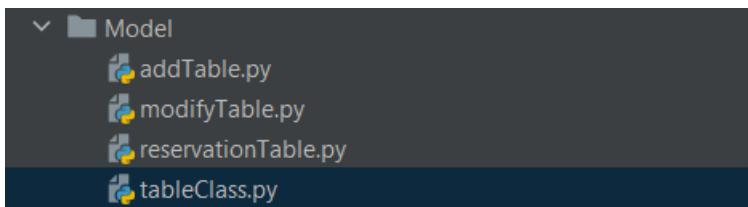
- MVC Architecture



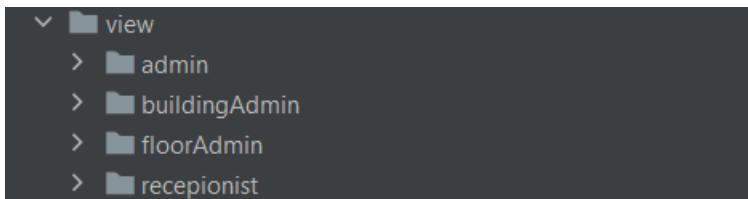
- Control: contains **main** the functions that control our app

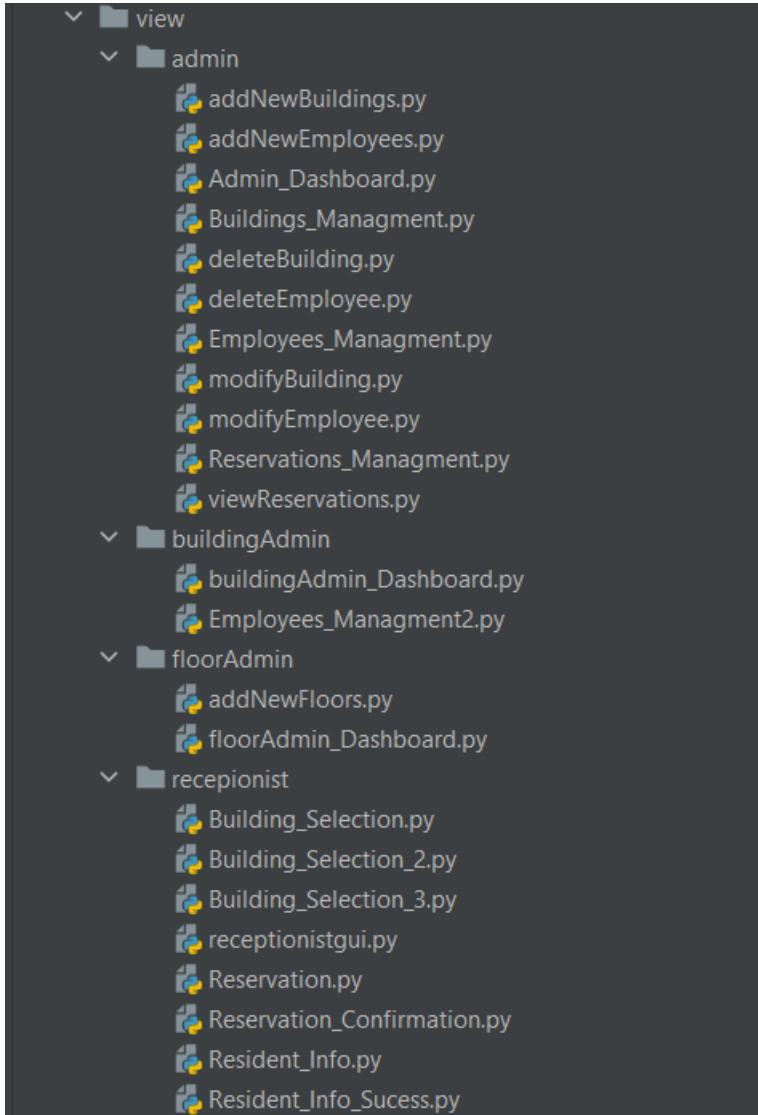


- Model: contain functions which deals with the database



- View: contains GUIs pages





Example of MVC in the functionalities

In the view:

```
66     def showUserInfo(self):
67
68         result = self.mc.getGuiMsgsControl()
69         if (self.nationalIdlineEdit.text() == ''):
70             messagebox.showwarning('Warning', str(result[6]))
71             return
72         result = self.mc.getUserInfoControl(int(self.nationalIdLineEdit.text()))
73         print(result)
74         if (not result):
75             result = self.mc.getGuiMsgsControl()
76             messagebox.showError('Error', result[8])
77             return
78         main_control.userInfo = result
79         _translate = QtCore.QCoreApplication.translate
80         self.showNameLabel.setText(_translate("Resident_Info_Sucess", str(result[0][3])))
81         self.showAgeLabel.setText(_translate("Resident_Info_Sucess", str(result[0][2])))
82         self.showGenderLabel.setText(_translate("Resident_Info_Sucess", str(result[0][6])))
83         self.showLastStayLabel.setText(_translate("Resident_Info_Sucess", str(result[0][7])))
84         self.showEligibleLabel.setText(_translate("Resident_Info_Sucess", str(result[0][8])))
85
86         return
```

In the control:

```
151     #call model to get gui msgs
152     def getGuiMsgsControl(self):
153         queryobj=logitable(type,self.connection)
154         result = queryobj.gui_msgs()
155         return result
156
157     #call model to get employee info
158     def getEmployeeInfoControl(self,id):...
159
160     #call model to get user info
161     def getUserInfoControl(self,id):
162         queryobj=reservationTable(self.connection)
163         result = queryobj.showUser(id)
164         return result
165
166
167
168
```

In the model:

```
    self.connection.cursor()
9
10    def showUser(self,id):
11        try:
12            cursor = self.connection.cursor()
13            query = """
14                SELECT *
15                FROM agile_project.user
16                WHERE national_id = {} ;""".format(id)
17            with self.connection.cursor() as cursor:
18                cursor.execute(query)
19                result = cursor.fetchall()
20                return result
21        except Error as e:
22            print(e)
23            return e
24
```

```
27
28    def gui_msgs(self):
29        try:
30            cursor = self.connection.cursor()
31            get_query = """
32                SELECT msg FROM agile_project.gui_msgs;"""
33            with self.connection.cursor() as cursor:
34                cursor.execute(get_query)
35                result = cursor.fetchall()
36                return result
37        except Error as e:
38            print(e)
39            return e
```

Another example of a more complex functionality:

In the view:

```

41     def showConfirm(self):
42         restuple = []
43         dateTime = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
44         restuple.append(dateTime)
45         restuple.append(main_control.userInfo[0][0])
46         restuple.append(self.floorComboBox.currentText())
47         restuple.append(self.buildingIdLabel.text())
48         result = self.mc.getGuiMsgsControl()
49         if (str(self.mc.confirmRes(restuple)).lower() != 'none'):
50             messagebox.showwarning('Warning', str(result[3]))
51             return
52         self.confirmRes = QtWidgets.QMainWindow()
53         self.ui = CR.Ui_viewReservations()
54         self.ui.setupUi(self.confirmRes)
55         self.confirmRes.show()
56

```

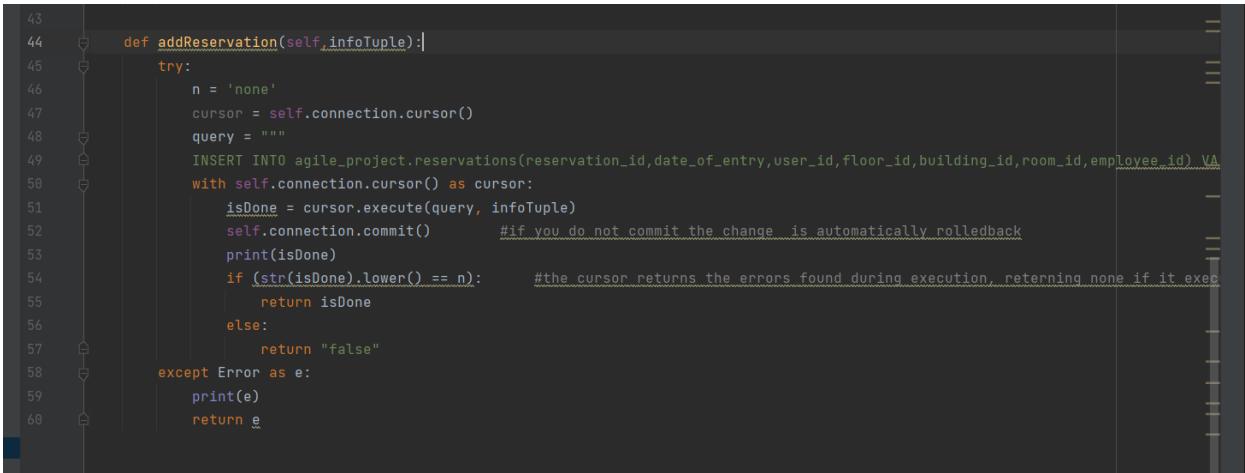
In the control:

```

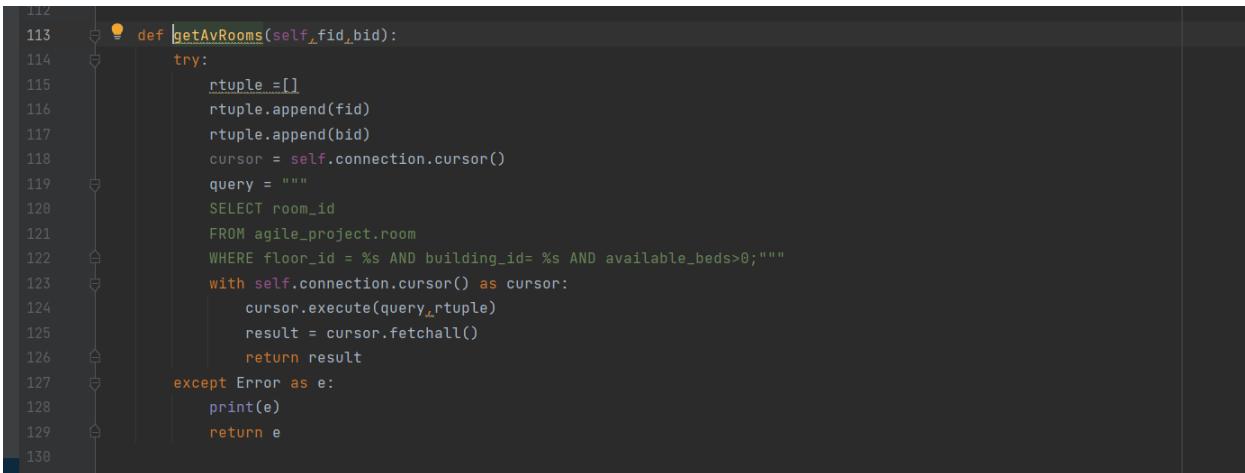
71
72     #function to set the eligibility of the reserved user to no so that he/she cannot reserve another bed
73     def setEligibleNO(self,infotuple):...
74
75
76     #function to decrease the available beds in a room
77     def decRoom(self,infotuple):...
78
79
80     #function to add the reservation to the db
81     def confirmRes(self,infotuple):
82         queryobj=reservationTable(self.connection)
83         res = queryobj1.genResID()
84         infotuple.insert(0,int(str(res[0])[1:2])+1)
85         self.resID = infotuple[0]
86         result = queryobj1.getAvRooms(infotuple[3],infotuple[4])
87         rooms = []
88         for item in result:
89             rooms.append(int(str(item)[1:2]))
90         if len(rooms) == 1:
91             room = rooms[0]
92         else:
93             room = random.choice(rooms)
94         infotuple.append(self.empID)
95         queryobj= addStakeholder(self.connection)
96         result = queryobj.addReservation(infotuple)
97         self.setEligibleNO(infotuple)
98         self.decRoom(infotuple)
99
100        return result
101
102
103
104
105
106
107
108
109

```

In the model:



```
43
44     def addReservation(self, infoTuple):
45         try:
46             n = 'none'
47             cursor = self.connection.cursor()
48             query = """
49             INSERT INTO agile_project.reservations(reservation_id, date_of_entry, user_id, floor_id, building_id, room_id, employee_id) VA
50             with self.connection.cursor() as cursor:
51                 isDone = cursor.execute(query, infoTuple)
52                 self.connection.commit()          #if you do not commit the change is automatically rolledback
53                 print(isDone)
54                 if (str(isDone).lower() == n):    #the cursor returns the errors found during execution, returning none if it exec
55                     return isDone
56                 else:
57                     return "false"
58             except Error as e:
59                 print(e)
60                 return e
```



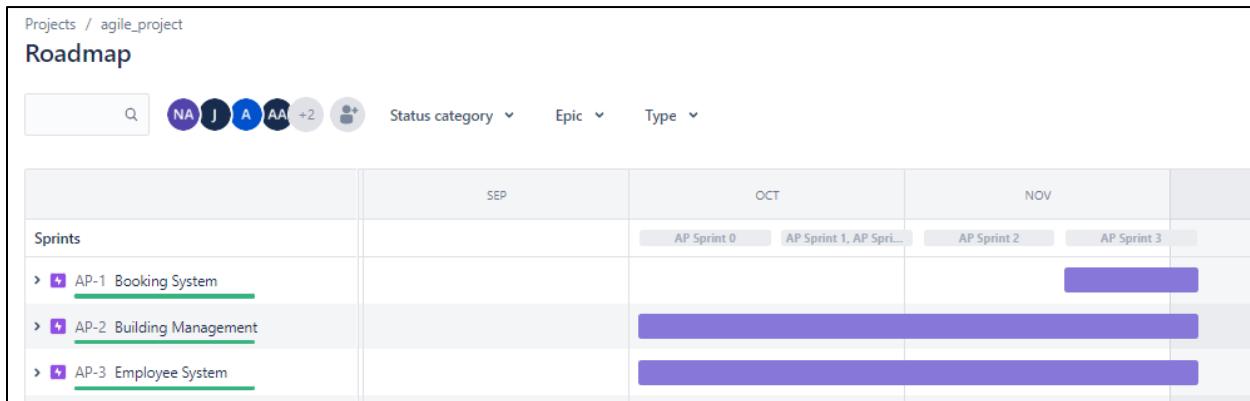
```
112
113     def getAvRooms(self, fid, bid):
114         try:
115             rtuple = []
116             rtuple.append(fid)
117             rtuple.append(bid)
118             cursor = self.connection.cursor()
119             query = """
120             SELECT room_id
121             FROM agile_project.room
122             WHERE floor_id = %s AND building_id= %s AND available_beds>0;"""
123             with self.connection.cursor() as cursor:
124                 cursor.execute(query, rtuple)
125                 result = cursor.fetchall()
126                 return result
127             except Error as e:
128                 print(e)
129                 return e
130             
```



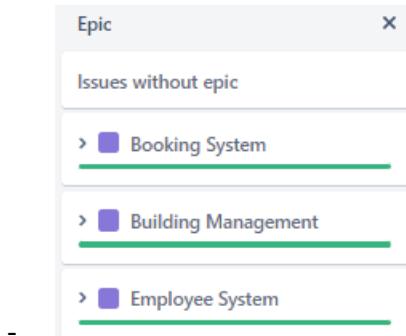
```
82
83     def genResID(self):
84         try:
85             cursor = self.connection.cursor()
86             query = """
87             SELECT Max(reservation_id)
88             FROM agile_project.reservations;"""
89             with self.connection.cursor() as cursor:
90                 cursor.execute(query)
91                 result = cursor.fetchall()
92                 print(result)
93                 return result
94             except Error as e:
95                 print(e)
96                 return e
97             
```

3.4 Jira

- **Account link:**
<https://nada20.atlassian.net/jira/software/projects/AP/boards/1/backlog>
- Our Road map: we have created 4 sprints [from sprint 0 to sprint 3]
This screen shot clarify the timeline of our sprints and how each epic contribute in each sprint



- Our Epics



Projects / agile_project / AP-3

Employee System

Attach Add a child issue Link issue ...

Description
Key project stakeholders

- Admin
- Building Admins
- Floor Admins
- Receptionists

Child issues Order by ... + 100% Done

Issue ID	Description	Status
AP-7	login page	DONE
AP-13	Database Creation	DONE
AP-44	Database Modification	DONE
AP-21	As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data	DONE
AP-38	As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data	DONE
AP-62	Apply Error handling in Login Page	DONE

Projects / agile_project / AP-1

Booking System

Attach Add a child issue Link issue ...

Description
Design a desktop application that manages booking system for rooms provided by Ministry of Housing, Utilities and Urban Communities.

Child issues Order by ... + 100% Done

Issue ID	Description	Status
AP-28	As a Admin I want to view, modify reservations so that I can track all reservations in the system.	DONE
AP-31	As a Building Admin I want to view, modify reservations so that I can track all reservations in the building.	DONE
AP-32	As a Floor Admin I want to view reservations so that I will be aware of the reservations log	DONE
AP-33	As a Floor Admin I want to Rate Resident's Stay so that I will have the knowledge if he deserve the opportunity to be reserved in our s...	DONE
AP-35	As a Receptionist I want to be able to register a bed to the client so that I will be able to make successful reservations to clients	DONE
AP-34	As a Floor Admin I want to Update Room Status so that I will be able to decide to assign this room to new clients or not	DONE
AP-36	As a Receptionist I want to retrieve client info from his ID so that I will check his identity to be able to make a decision about make a r...	DONE
AP-37	As a Receptionist I want to view alternative building occupancy so that I will be able to assign beds to clients in other buildings if the c...	DONE
AP-39	As a user I want to login to Log out from my page so that I get out of the application	DONE

Projects / agile_project / AP-2

Building Management

Attach Add a child issue Link issue ...

Description - Unsaved changes

The system allows the building admins, and floor admins to asses room conditions and guest leave.

Child issues

	Order by	100% Done
<input checked="" type="checkbox"/> AP-19 Uploading our Project on Github	AS	DONE
<input checked="" type="checkbox"/> AP-18 MVC Structure Creation	A	DONE
<input checked="" type="checkbox"/> AP-25 As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data	AS	DONE
<input checked="" type="checkbox"/> AP-49 As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need	A	DONE

- Random Screenshots for our jira account while working
- Sprint 2: Backlog

Projects / agile_project

Backlog

Epic Issues without epic + Create Epic

AP Sprint 2 3 Nov – 17 Nov (4 issues)

<input checked="" type="checkbox"/> AP-41 Database Modification	EMPLOYEE SYSTEM	0 0 0	Complete sprint
<input checked="" type="checkbox"/> AP-21 As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data	EMPLOYEE SYSTEM	0 0 0	DONE
<input checked="" type="checkbox"/> AP-30 As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data	EMPLOYEE SYSTEM	0 0 0	IN PROGRESS
<input checked="" type="checkbox"/> AP-25 As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data	BUILDING MANAGEMENT	0 0 0	IN PROGRESS

+ Create issue

Backlog (13 issues)

<input checked="" type="checkbox"/> AP-28 As a Admin I want to view, modify reservations so that I can track all reservations in the system.	BOOKING SYSTEM	0 0 0	Create sprint
<input checked="" type="checkbox"/> AP-31 As a Building Admin I want to view, modify reservations so that I can track all reservations in the building.	BOOKING SYSTEM	0 0 0	NA
<input checked="" type="checkbox"/> AP-32 As a Floor Admin I want to view reservations so that I will be aware of the reservations log	BOOKING SYSTEM	0 0 0	AS
<input checked="" type="checkbox"/> AP-33 As a Floor Admin I want to Rate Resident's Stay so that I will have the knowledge if he deserve the opportunity to be reserved in our shelter	BOOKING SYSTEM	0 0 0	NA
<input checked="" type="checkbox"/> AP-34 As a Floor Admin I want to Update Room Status so that I will be able to decide to assign this room to new clients or not	BOOKING SYSTEM	0 0 0	TO DO
<input checked="" type="checkbox"/> AP-35 As a Receptionist I want to be able to register a bed to the client so that I will be able to make successful reservations to clients	BOOKING SYSTEM	0 0 0	A
<input checked="" type="checkbox"/> AP-36 As a Receptionist I want to retrieve client info from his ID so that I will check his identity to be able to make a decision about make a reservation for him or not	BOOKING SYSTEM	0 0 0	J
<input checked="" type="checkbox"/> AP-37 As a Receptionist I want to view alternative building occupancy so that I will be able to assign beds to clients in other buildings if the capacity in my building is full	BOOKING SYSTEM	0 0 0	A
<input checked="" type="checkbox"/> AP-38 As a user I want to communicate with other users so that I can send requests to them to ask for tasks or to complain about something	BOOKING SYSTEM	0 0 0	NA
<input checked="" type="checkbox"/> AP-39 As a user I want to login to Log out from my page so that I get out of the application	BOOKING SYSTEM	0 0 0	NA
<input checked="" type="checkbox"/> AP-40 As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need	BOOKING SYSTEM	0 0 0	A

Sprint 2: Board

Projects / agile_project

AP Sprint 2

The scrum board displays the following tasks:

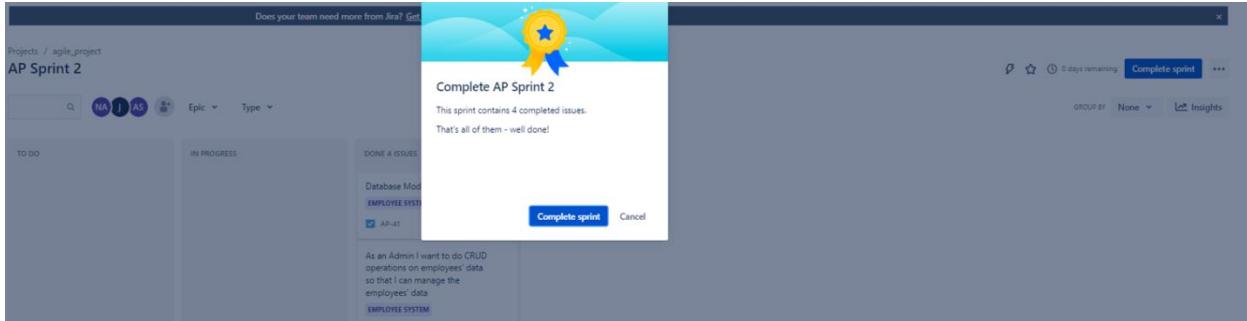
- TO DO 1 ISSUE**:
 - As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data
BUILDING MANAGEMENT
AP-25
- IN PROGRESS 2 ISSUES**:
 - As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-21
 - As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-30
- DONE 1 ISSUE**:
 - Database Modification
EMPLOYEE SYSTEM
AP-41

Projects / agile_project

AP Sprint 2

The scrum board displays the following tasks:

- TO DO**:
 - As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-21
 - As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-30
 - As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data
BUILDING MANAGEMENT
AP-25
- IN PROGRESS**:
 - As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-41
- DONE 4 ISSUES**:
 - Database Modification
EMPLOYEE SYSTEM
AP-41
 - As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-21
 - As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data
EMPLOYEE SYSTEM
AP-30
 - As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data
BUILDING MANAGEMENT
AP-25



Sprint 3: Backlog

Issue ID	Description	Type	Status
AP-28	As a Admin I want to view, modify reservations so that I can track all reservations in the system.	BOOKING SYSTEM	DONE ✓ NA
AP-31	As a Building Admin I want to view, modify reservations so that I can track all reservations in the building.	BOOKING SYSTEM	DONE ✓ AS
AP-32	As a Floor Admin I want to view reservations so that I will be aware of the reservations log	BOOKING SYSTEM	DONE ✓ AA
AP-40	As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need	BOOKING SYSTEM	IN PROGRESS ▾ NA
AP-33	As a Floor Admin I want to Rate Resident's Stay so that I will have the knowledge if he deserve the opportunity to be reserved in our shelter	BOOKING SYSTEM	IN PROGRESS ▾ NA
AP-34	As a Floor Admin I want to Update Room Status so that I will be able to decide to assign this room to new clients or not	BOOKING SYSTEM	IN PROGRESS ▾ NA
AP-35	As a Receptionist I want to be able to register a bed to the client so that I will be able to make successful reservations to clients	BOOKING SYSTEM	IN PROGRESS ▾ NA
AP-36	As a Receptionist I want to retrieve client info from his ID so that I will check his identity to be able to make a decision about make a reservation for him or not	BOOKING SYSTEM	TO DO ▾ J
AP-37	As a Receptionist I want to view alternative building occupancy so that I will be able to assign beds to clients in other buildings if the capacity in my building is full	BOOKING SYSTEM	TO DO ▾ A
AP-39	As a user I want to login to Log out from my page so that I get out of the application	BOOKING SYSTEM	TO DO ▾ NA
AP-62	Apply Error handling in Login Page	BOOKING SYSTEM	TO DO ▾ J

Sprint 3: Board

Column	Issue ID	Description	Type	Status
TO DO 4 ISSUES	AP-36	As a Receptionist I want to retrieve client info from his ID so that I will check his identity to be able to make a decision about make a reservation for him or not	BOOKING SYSTEM	TO DO ▾ J
IN PROGRESS 3 ISSUES	AP-33	As a Floor Admin I want to Rate Resident's Stay so that I will have the knowledge if he deserve the opportunity to be reserved in our shelter	BOOKING SYSTEM	IN PROGRESS ▾ NA
IN PROGRESS 3 ISSUES	AP-37	As a Receptionist I want to view alternative building occupancy so that I will be able to assign beds to clients in other buildings if the capacity in my building is full	BOOKING SYSTEM	IN PROGRESS ▾ A
IN PROGRESS 3 ISSUES	AP-39	As a user I want to login to Log out from my page so that I get out of the application	BOOKING SYSTEM	IN PROGRESS ▾ NA
DONE 4 ISSUES	AP-28	As a Admin I want to view, modify reservations so that I can track all reservations in the system.	BOOKING SYSTEM	DONE ✓ NA
DONE 4 ISSUES	AP-31	As a Building Admin I want to view, modify reservations so that I can track all reservations in the building.	BOOKING SYSTEM	DONE ✓ AS
DONE 4 ISSUES	AP-32	As a Floor Admin I want to view reservations so that I will be aware of the reservations log	BOOKING SYSTEM	DONE ✓ AA
DONE 4 ISSUES	AP-40	As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need	BOOKING SYSTEM	DONE ✓ J

User Story Tracking

Projects / agile_project / AP-1 / AP-39

As a user I want to login to Log out from my page so that I get out of the application

Attachment Add a child issue Link issue ...

Description
In this user story the receptionist need to verify the client info
0.75 points equivalent to quarter a day.

Acceptance Criteria
Scenario 1: Log out Successfully ✓
[Given]: I am on Dashboard
[When]: I click on Log out
[Then]: User is Logged out successfully

Scenario 2: Cannot Log out Successfully ✗
[Given]: I am on Dashboard
[When]: I click on Log out
[Then]: User isn't Logged out successfully

Estimated Time 0.75 points

Assignee Team None

Child issues Order by 66% Done

- AP-94 Implementing the Functionality DONE
- AP-95 Testing DONE
- AP-96 GUI Creation TO DO

Projects / agile_project AP Sprint 3

TO DO IN PROGRESS DONE IN ISSUES

As a Admin I want to see, modify reservations so that I can track all reservations in the system.
BOOKING SYSTEM

As a Building Admin I want to view modifications so that I can track all reservations in the Building.
BOOKING SYSTEM

As a Floor Admin I want to view modifications so that I will be aware of the reservations log.
BOOKING SYSTEM

As a user I want to clear the page that have issued items so that I can easily have access on only on the page that I need.
BOOKING SYSTEM

As a Floor Admin I want to have the ability to say that I will take the responsibility if the customer has reserved in our shelter.
BOOKING SYSTEM

As a Receptionist I want to be able to register a guest to the client so that I will be able to create successful reservations to clients.
BOOKING SYSTEM

As a Floor Admin I want to Update Room Status so that I will be able to decide to assign this room to many clients or not.
BOOKING SYSTEM

As a Receptionist I want to receive a card info from his ID so that I will check his identity to be able to make a decision about making a reservation for him or not.
BOOKING SYSTEM

As a Receptionist I want to view alternative building occupancy so that I will be able to assign books to clients in other buildings if the capacity in my building is full.
BOOKING SYSTEM

Apply Error handling in Login Page
BOOKING SYSTEM

Does your team need more from Jira? Get started!

Projects / agile_project AP Sprint 3

Epic Type

TO DO IN PROGRESS DONE 11 ISSUES

As a Admin I want to see, modify reservations so that I can track all reservations in the system.
BOOKING SYSTEM

Complete AP Sprint 3
This sprint contains 11 completed issues.
That's all of them - well done!

Complete sprint Cancel

0 days remaining Complete sprint

GROUP BY None Insights

Apply Error handling in Login Page
BOOKING SYSTEM

3.5 Good User Story

Example 1:

The screenshot shows a user story card and its detailed view in a project management tool.

User Story

Description

As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data

Description:

- data: Name, Date of birth, Address, Salary, Contact Info, Building assigned, shifts
- CRUD operation are Create, Read, Modify, and delete.
- This Employee can be a Building Admin, a Floor Admin, or a Receptionist.
- 1 day and half equivalent to 5x points

Acceptance Criteria

Scenario 1: Add employee Successfully ✓
[Given]: I am on Dashboard
[When]: I enter valid Name, Date of birth, Address, Salary, Contact Info, Building assigned, shifts
AND I click on Add
[Then]: Employee is added successfully

Scenario 2: Cannot add employee Successfully ✗
[Given]: I am on Dashboard
[When]: I enter an invalid Name, or Date of birth, or Address, Salary, or Contact Info, or Building assigned, or shifts, or all.
AND I click on Add
[Then]: Employee isn't added successfully

Scenario 3: Modify employee Successfully ✓
[Given]: I am on Dashboard
[When]: I enter a valid Id

Owner

Owner: Nada Amr

Details

Assignee: Nada Amr

Time tracking: No time logged

Labels: None

Sprint: None

Story point estimate: None

Development: 1 branch, 10 commits, 3 pull requests, MERGED

Reporter: Nada Amr

Automation: Rule executions

Link to GitHub

Acceptance

Critiria

Epic

Epic: Employee

[Given]: I am on Dashboard
[When]: I enter a valid Id
AND I click on Delete
[Then]: Employee is deleted successfully

Scenario 6: Cannot delete employee Successfully ✗
[Given]: I am on Dashboard
[When]: I enter an invalid Id
AND I click on Delete
[Then]: Employee isn't deleted successfully

Estimated Time

Estimated Time: 5 points

Assignee Team: None

Tasks

Child issues:

- AP-42 Implementing the Functionality of the creation operation page
- AP-43 Implementing the Functionality of the modify and delete operation page
- AP-44 Implementing the Functionality of read operation page
- AP-45 Creating GUI
- AP-46 Testing

Order by: 100% Done

Example 2:

The screenshot shows a user story backlog and its details in a project management tool.

User Story Details:

- Title:** As a user I want to login to Log out from my page so that I get out of the application
- Description:** In this user story the receptionist need to verify the client info
- Acceptance Criteria:**
 - Scenario 1:** Log out Successfully ✓
 - [Given]: I am on Dashboard
 - [When]: I click on Log out
 - [Then]: User is Logged out successfully
 - Scenario 2:** Cannot Log out Successfully ✗
 - [Given]: I am on Dashboard
 - [When]: I click on Log out
 - [Then]: User isn't Logged out successfully
- Estimated Time:** 0.75 points
- Assignee Team:** None

Child Issues:

- AP-64 Implementing the Functionality (Status: DONE)
- AP-65 Testing (Status: DONE)
- AP-66 GUI Creation (Status: TO DO)

Owner Details:

- Assignee:** Nada Amr
- Time tracking:** No time logged
- Labels:** None
- Sprint:** AP Sprint 3
- Story point estimate:** None
- Development:** Create branch, Create commit
- Reporter:** Nada Amr
- Automation:** Rule executions

Created: December 13, 2022 at 5:03 PM
Updated: 28 seconds ago
Resolved: 28 seconds ago

Annotations:

- Owner:** Red text above the owner details.
- Acceptance:** Red text next to the acceptance criteria section.
- Critiria:** Red text next to the acceptance criteria section.
- Estimated Time:** Red text next to the estimated time field.
- Tasks:** Red text next to the tasks section.
- Sprint:** Red text next to the sprint field.

Example 3:

Projects / agile_project / AP-3 / AP-41 Task

Database Modification

Attach Add a child issue Link issue ...

Description

- Database modification: reworking and removing redundant tables for easier database management
- Singleton: applying singleton connection to database in all queries

Original Estimate	2h	Estimated Time
-------------------	----	----------------

Assignee Team None

Attachments (1)

The screenshot shows a Jira task card for 'Database Modification'. The title is 'Database Modification'. The 'Task' tab is selected. The 'Description' section contains two bullet points: 'Database modification: reworking and removing redundant tables for easier database management' and 'Singleton: applying singleton connection to database in all queries'. Below the description is a table with 'Original Estimate' and 'Estimated Time' columns, both showing '2h'. The 'Assignee Team' field is set to 'None'. A 'Attachments' section shows one file named 'updated schema.png' from '29 Dec 2022, 03:58 PM'.

Example 4:

Projects / agile_project / AP-2 / AP-40 Bug

As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need

Done ✓ Done

Details Owner

Assignee ahmed02amr Assign to me

Labels None

Sprint None +1

Story point estimate 1 completed sprint

AP Sprint 3 Dec 03, 2022 Development

Estimated Time

Description

The aim is to close the pages which the user is moved from

Original Estimate 2h

Activity

Show: All Comments History

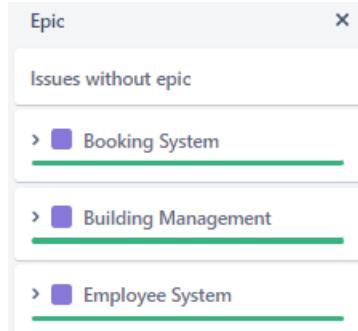
Newest first 4

The screenshot shows a Jira bug card for the issue 'close the pages that have moved from'. The title is 'close the pages that have moved from'. The 'Bug' tab is selected. The 'Description' section contains the text 'The aim is to close the pages which the user is moved from'. Below the description is a table with 'Original Estimate' and 'Estimated Time' columns, both showing '2h'. To the right of the card, there is a sidebar with sections for 'Details' (Owner: ahmed02amr, Assignee: ahmed02amr, Assign to me), 'Labels' (None), 'Sprint' (None +1), 'Story point estimate' (1 completed sprint, AP Sprint 3, Dec 03, 2022, Development), and activity history (Comments, History).

3.6 Multilevel Planning

1- Firstly we have high level requirements which we represent it as **Epics**

- We have three main epics



2- Firstly we divide each one of our epics in **user stories**

The screenshot shows a software interface for managing user stories. At the top, it displays the path: Projects / agile_project / AP-1. Below this, it shows a section for the 'Booking System' epic. It includes buttons for 'Attach', 'Add a child issue', 'Link issue', and a '...' button. The main area is titled 'Description' and contains the text: 'Design a desktop application that manages booking system for rooms provided by Ministry of Housing, Utilities and Urban Communities.' Below this, there is a table titled 'Child issues' with the following data:

Issue ID	Description	Order by	Status
AP-28	As a Admin I want to view, modify reservations so that I can track all reservations in the system.	NA	DONE
AP-31	As a Building Admin I want to view, modify reservations so that I can track all reservations in the building.	AS	DONE
AP-32	As a Floor Admin I want to view reservations so that I will be aware of the reservations log	AA	DONE
AP-33	As a Floor Admin I want to Rate Resident's Stay so that I will have the knowledge if he deserve the opportunity to be reserved in our s...	NA	DONE
AP-35	As a Receptionist I want to be able to register a bed to the client so that I will be able to make succesful reservations to clients	NA	DONE
AP-34	As a Floor Admin I want to Update Room Status so that I will be able to decide to assign this room to new clients or not	NA	DONE
AP-36	As a Receptionist I want to retrieve client info from his ID so that I will check his identity to be able to make a decision about make a r...	J	DONE
AP-37	As a Receptionist I want to view alternative building occupancy so that I will be able to assign beds to clients in other buildings if the c...	A	DONE
AP-39	As a user I want to login to Log out from my page so that I get out of the application	NA	DONE

Projects / agile_project / AP-3

Employee System

Attach Add a child issue Link issue ...

Description
Key project stakeholders

- Admin
- Building Admins
- Floor Admins
- Receptionists

Child issues Order by 100% Done

Issue ID	Description	Status
AP-7	login page	DONE
AP-13	Database Creation	DONE
AP-14	Database Modification	DONE
AP-21	As an Admin I want to do CRUD operations on employees' data so that I can manage the employees' data	DONE
AP-30	As a Building Admin I want to do CRUD operations on employees' Data so that I can manage the employees' data	DONE
AP-62	Apply Error handling in Login Page	DONE

Projects / agile_project / AP-2

Building Management

Attach Add a child issue Link issue ...

Description - Unsaved changes
The system allows the building admins, and floor admins to asses room conditions and guest leave.

Child issues Order by 100% Done

Issue ID	Description	Status
AP-19	Uploading our Project on Github	DONE
AP-18	MVC Structure Creation	DONE
AP-25	As an Admin I want to do CRUD operations on Buildings' Data so that I can manage the Buildings' data	DONE
AP-40	As a user I want to close the pages that have moved from so that I can easily have work on only on the page that I need	DONE

3- Each user story we divide it into tasks

We will take random samples:

login page

Attach Add a child issue Link issue ...

Description

- This page is our desktop application page for admin, building admin, floor admin, and receptionist
- 1 day "1d" equivalent to 3x points

Acceptance Criteria

Scenario 1: User Log in Successfully ✓

[Given]: I am on log-in page
[When]: I give valid username and , password
AND I click on Login
[Then]: User logs in successfully

Scenario 2: User Log in unsuccessfully ✗

[Given]: I am on log-in page
[When]: I give invalid username or , password or both
AND I click on Login
[Then]: User cannot log in

Estimated Time 3 points

Assignee Team Anas Salah

Attachments (1)

122.jpg 15 Nov 2022, 05:42 PM

...

Child issues

	Order by	...	+
AP-10	Implementing the Functionality	AS	DONE
AP-11	Testing	J	DONE
AP-12	GUI Creation	NA	DONE

Database Creation

[Attach](#) [Add a child issue](#) [Link issue](#) [...](#)

Description

In this task we need to create database schema for our project

3points equivalent to 1 day

Acceptance Criteria

None

Estimated Time 3 points

Assignee Team None

Attachments (1)



database schema.jpg

02 Dec 2022, 01:23 PM

... +

Child issues

Order by ... +

100% Done

[AP-14](#) EER Diagram Creation

AS DONE

[AP-16](#) Tables Creation

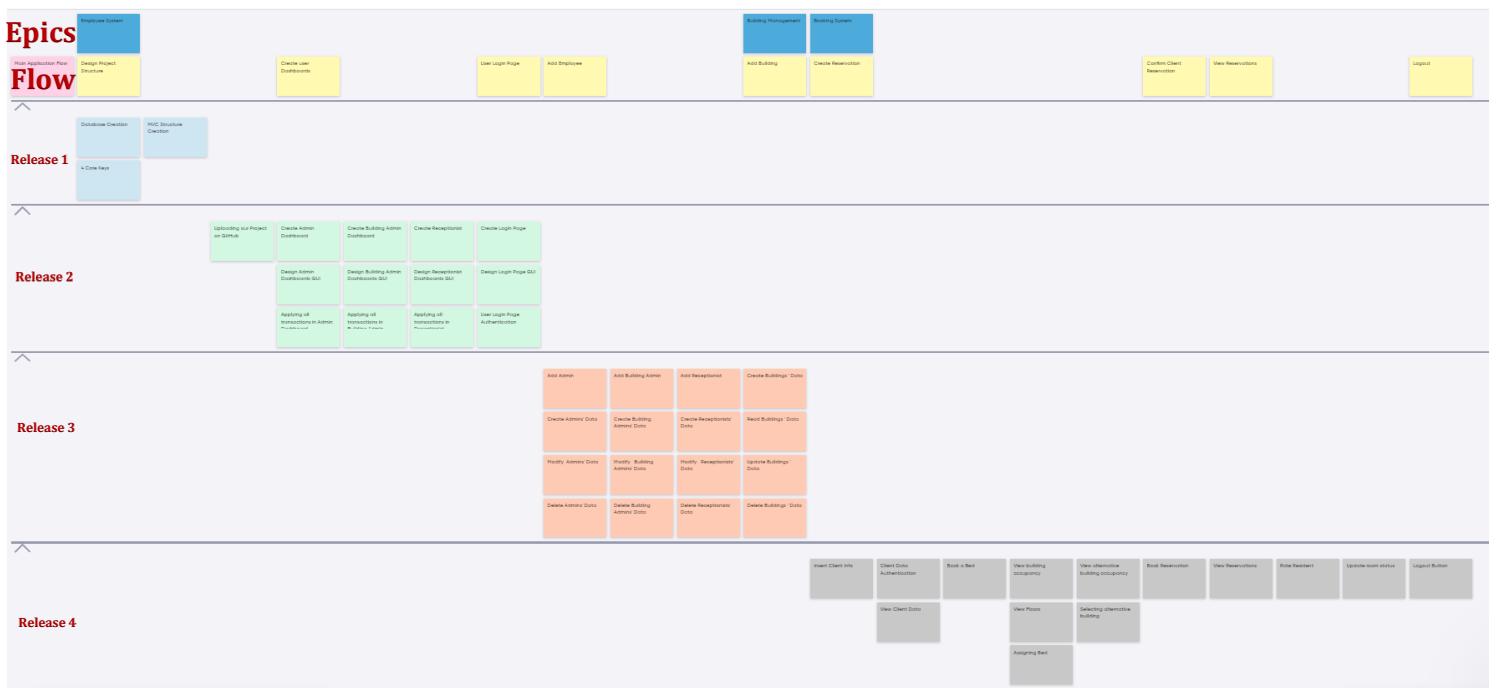
MM DONE

[AP-17](#) Schema Generation

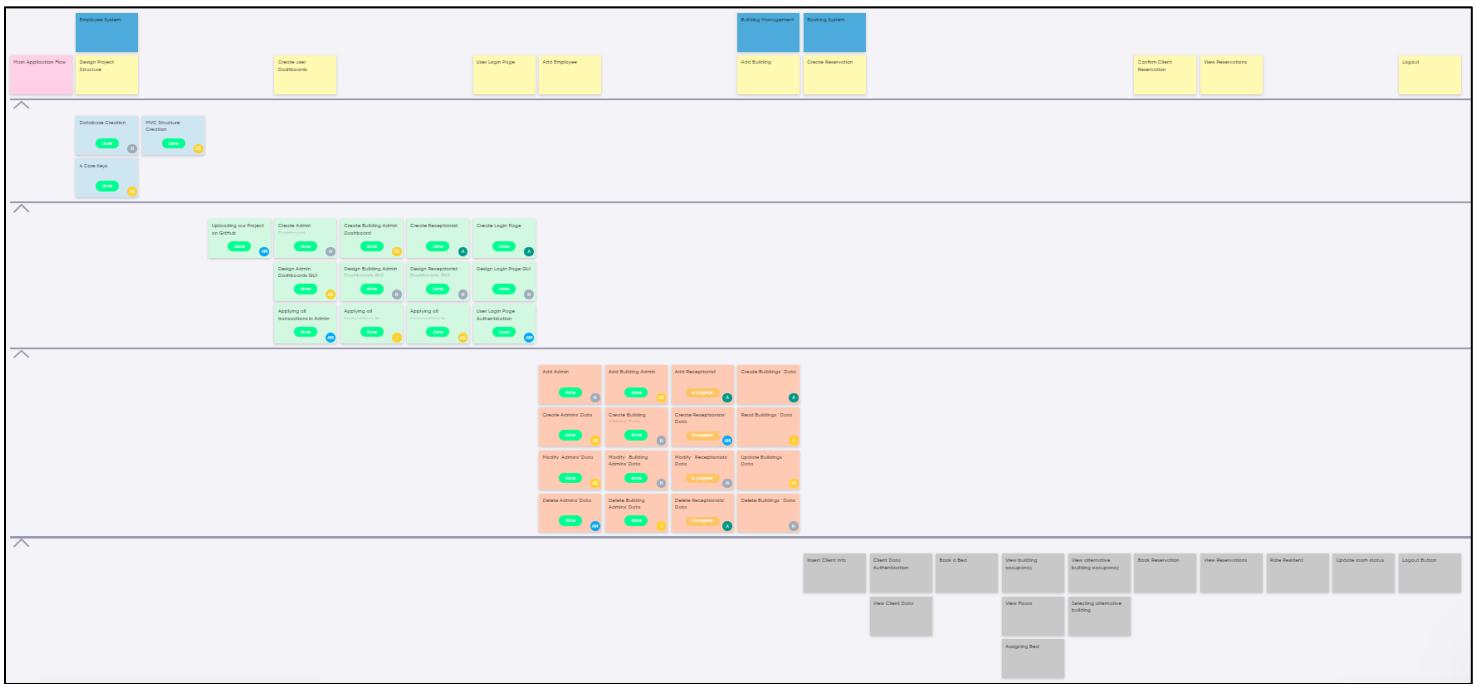
A DONE

3.7 Story Mapping

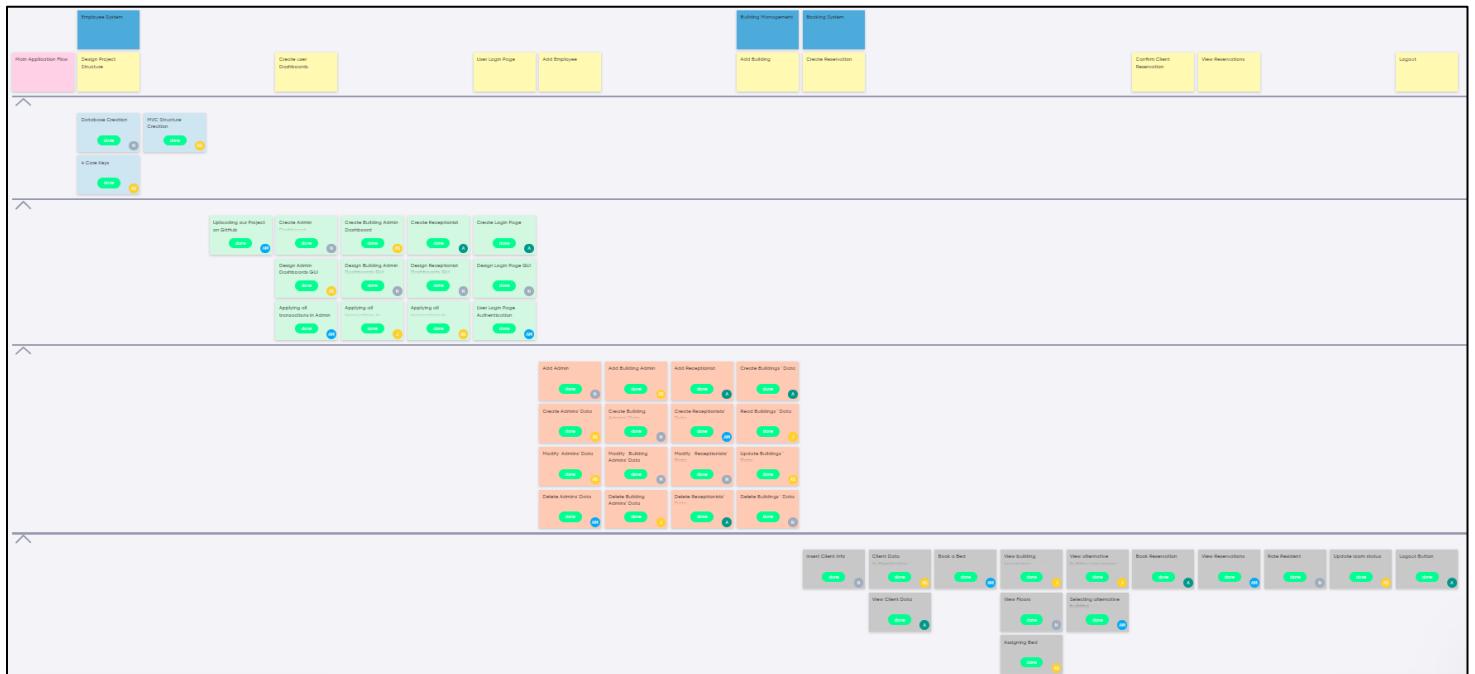
- **Account link:** <https://app.cardboardit.com/maps/214667>
- 1- Story Mapping structure is 2D: Horizontally is the flow of the application “walking skeleton”, vertically is the tasks for each card, also the releases “and it is ordered by priority”.
- 2- Each release has different color for clear visualization



- Screen shot while working



- Screenshot after we are finished



3.8 Trello

We used this project management tool to track the story cards, **we put each card and show its task has been done on how much releases.**

- We created lists for our main cards
- We create our labels:

Green label: Epic 1

Pink label: Epic 2

Red label: Epic 3

Purple label: means release one

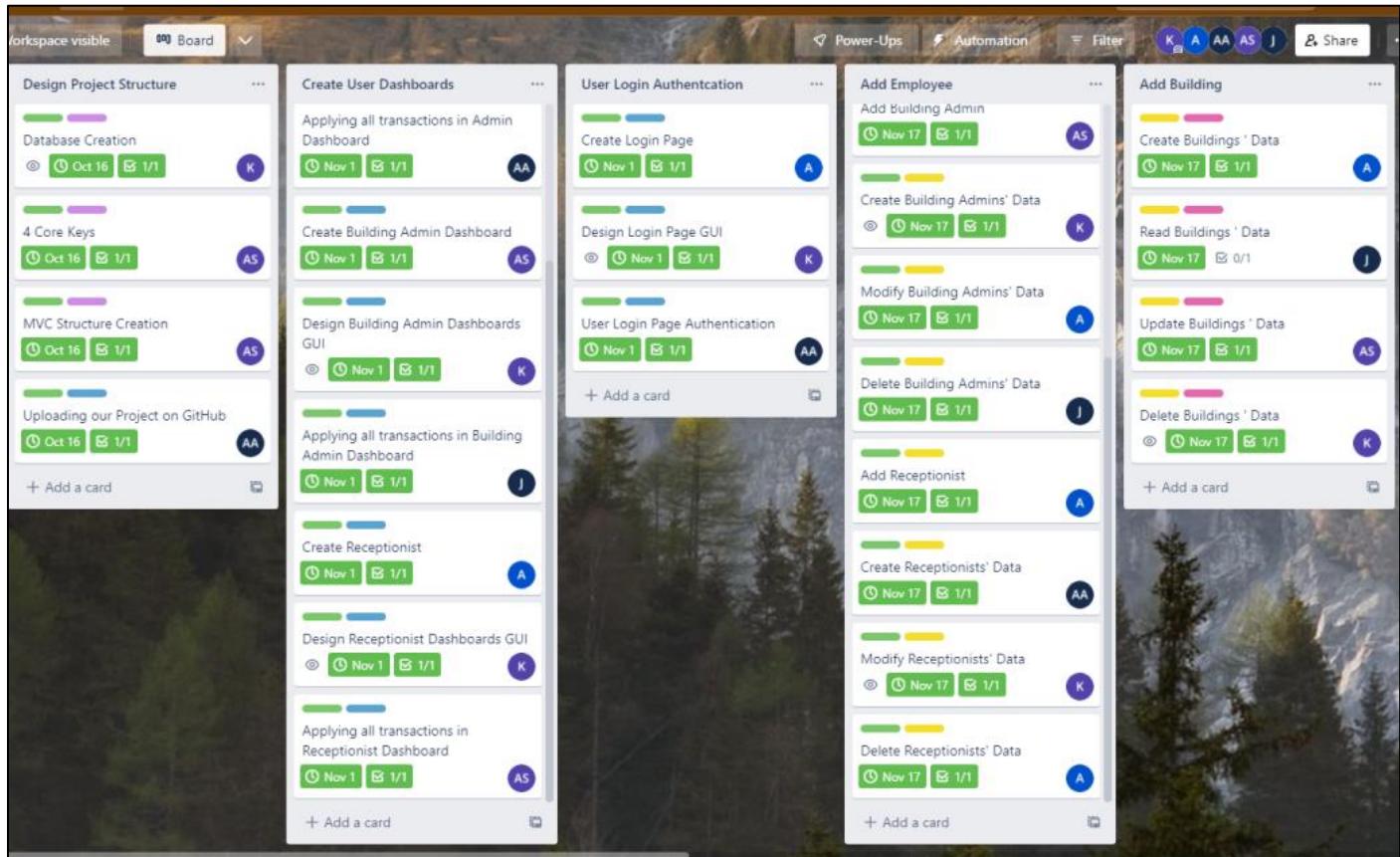
Blue label: release two

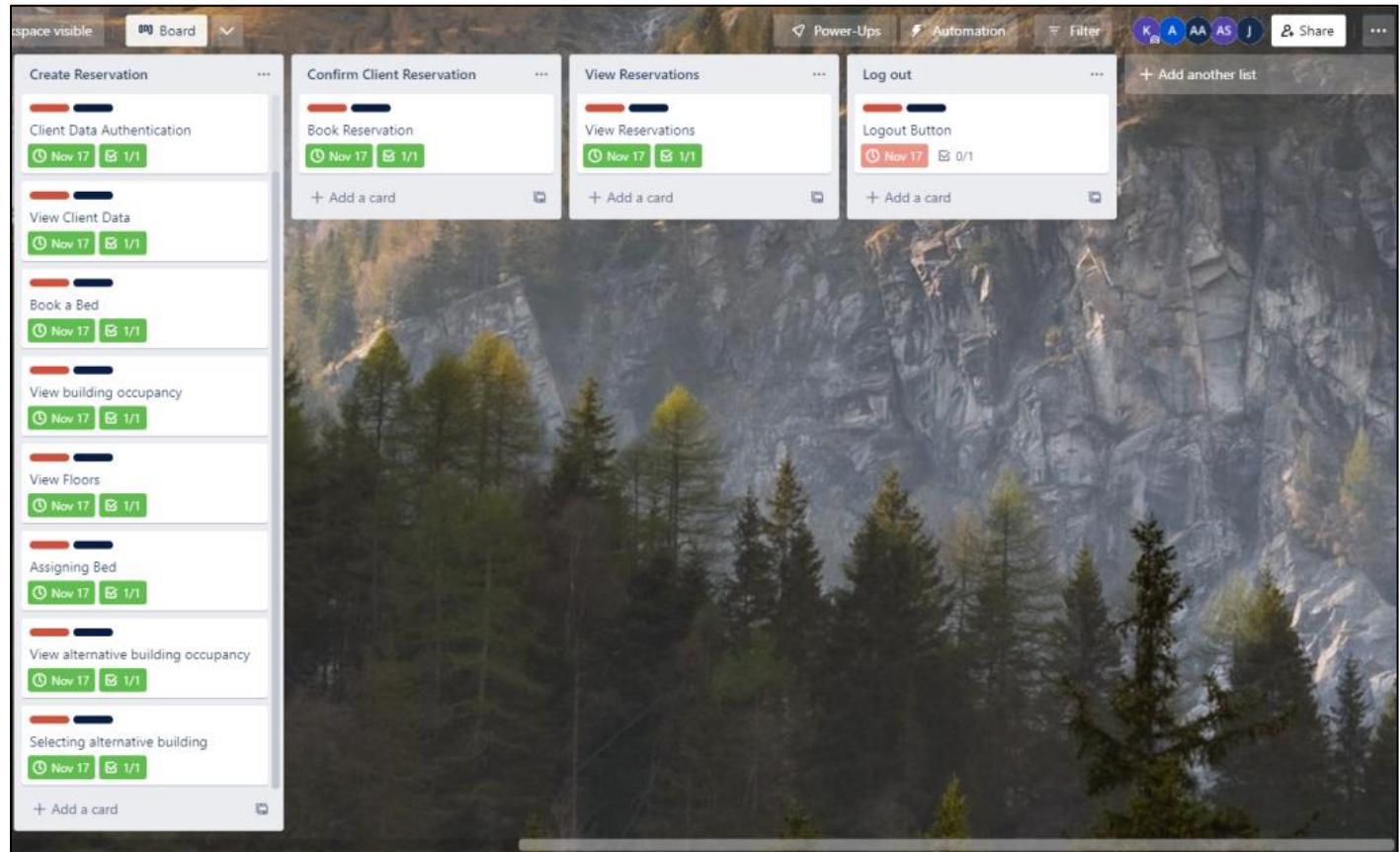
Yellow label: release three

Black label: release four

- We create added cards
- we assign due date for each card
- we assign every card to a member

Final Result as shown in the figure below:





3.9 Workflow

- 1- We made the first 2 meetings with the client, which we talked about before, to understand the idea of our project.
- 2- We prepared the project management tools which we will use: Jira, GitHub, Trello, Cardio, then We design the main structure of our project
- 3- Before each **sprint** we do a planning meeting with maximum 2 hours: in this meeting we assign members for user stories and tasks.

2022-1-10 "Sprint0 Planning"

Created by Alaa Mohamed Mohamed Hamdy Ahmed, with a template
Last updated: just a moment ago by Nada Amr • 1 min read

Date
Oct 1, 2022

Participants

- @Alaa Mohamed Mohamed Hamdy Ahmed
- @Anas Salah
- @Nada Amr**
- @Jannahayman
- @ahmed02amr

Goals

- Identify Database structure and 4CC
- Define MVC structure of the project

Discussion topics

Time	Item	Presenter	Notes
10 minutes	MVC	Alaa	<ul style="list-style-type: none">Which components will be placed in Model, View, ControlAny additional requirements needed from the team
5 minutes	4 CC	Gayar	<ul style="list-style-type: none">Which 4CC will be implemented in our databasedefining database structure

Action items

- @ahmed02amr Database Creation
- @ahmed02amr add 4CC in database
- @Alaa Mohamed Mohamed Hamdy Ahmed Implement the MVC structure

4- Stand up meeting: we do it offline, or online it depends on the flow of the day.

5- Sprint review: We do it after our sprint to review our release with our client.

6- Sprint Retrospective meeting: we do it as a team to check our strength points and do it again and also check our weakness to avoid it next time.

First Retrospective

Created by Alaa Mohamed Mohamed Hamdy Ahmed, with a template
Last updated: Dec 20, 2022 • 1 min read

Overview

Reflect on past work and identify opportunities for improvement by following the instructions for the Retrospective Play.

Date	17/10/22
Team	
Participants	@Nada Amr @Alaa Mohamed Mohamed Hamdy Ahmed @Anas Salah @ahmed02amr @Jannahayman

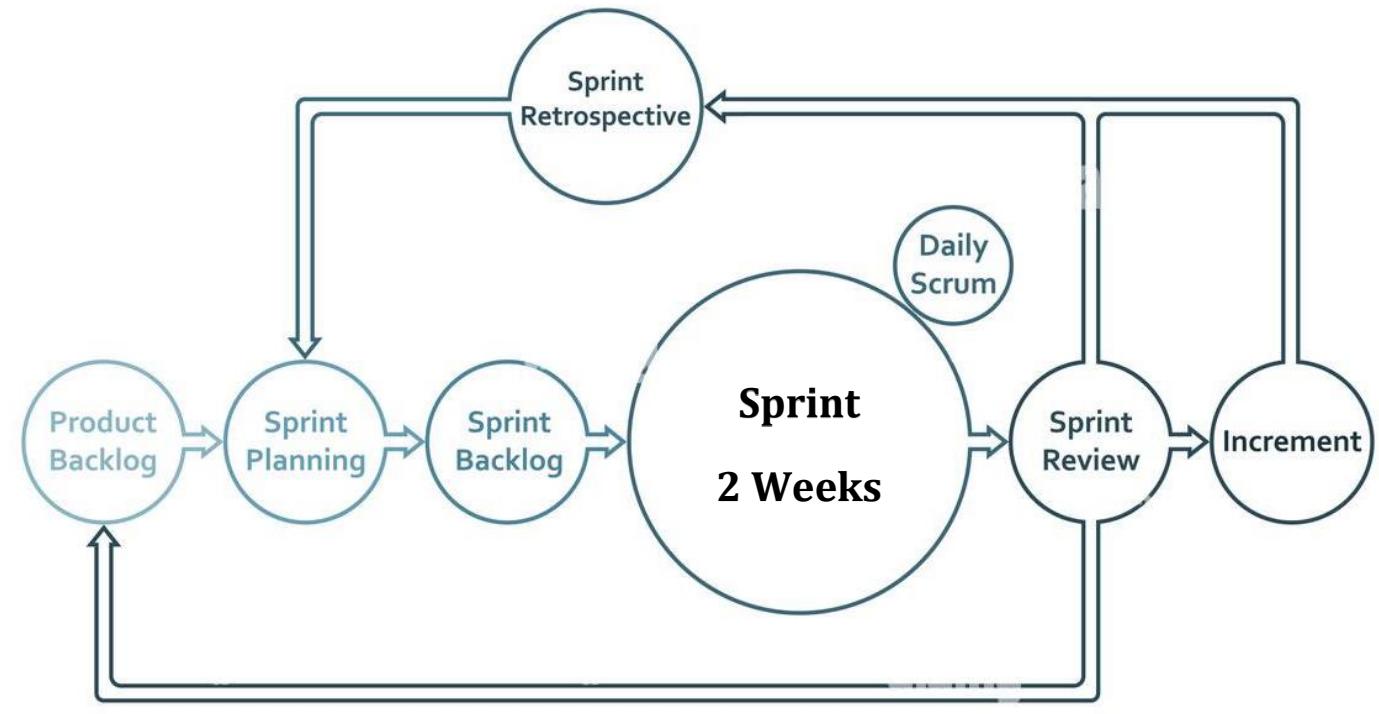
Retrospective

Add your Start doing, Stop doing, and Keep doing items to the table below. We'll use these to talk about how we can improve our process going forward.

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none">Sparring work earlier in the development cycleSharing progress at daily standups	<ul style="list-style-type: none">Scheduling meetings without a clear agenda or meeting goalsWorking more than agreed upon between the team	<ul style="list-style-type: none">Team members helping each other in their tasksWrite notes daily about problems faced or occurred to alert the team withFocusing on each PBI and don't rush the workStart applying work constantly on github

Action items

Daily Stand ups progress discussion
 Shared Notes between the team
 Update github regulalry



7- After we finished our project we calculated our velocity for upcoming projects.

Sprint 1: 3 user stories x 5 story points = 15

Sprint 2: 4 user stories x 9 story points = 36

Sprint 3: 5 user stories x 7 story points = 35

Total = 86

So, your average sprint velocity is $86 \div 3$ is approximately 29

Additional Part:

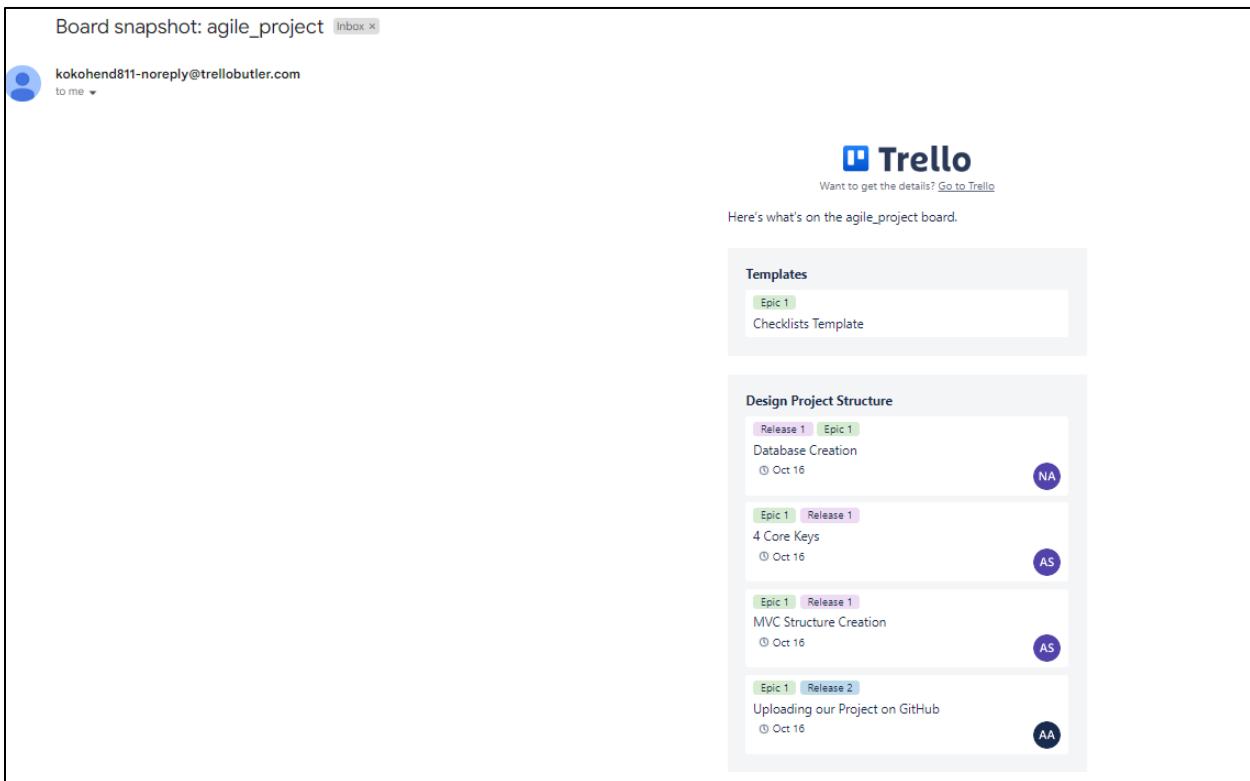
We do more automation on the project tool managements which we used to work more efficient and in a flexible way.

8- We connected jira to our Github to link the user stories as an branches on our commits.



The screenshot shows the Jira interface for the project 'agile_project'. The left sidebar has sections for PLANNING (Roadmap, Backlog), DEVELOPMENT (Code, Deployments), and OPERATIONS (Project page, Add shortcut, Project settings). The 'Code' section is selected. The main area shows 'Latest repositories' with one entry: 'Jira-Team-14/agile-project' (last updated about 16 hours ago). Below it is a 'Linked pull requests' section with a single item: 'git commit -m "AP-03 tmp" → main' (from repository: Jira-Team-14/agile-project). The pull request status is MERGED, with no reviewers or comments, updated 6 days ago.

9- On trello we send a report of our board at the end of each week



The screenshot shows a Trello board snapshot for the project 'agile_project'. It includes an inbox message from 'kokohend811-noreply@trellobutler.com' to 'me'. The main content area shows the Trello logo and a message: 'Here's what's on the agile_project board.' It displays two sections: 'Templates' (Checklists Template) and 'Design Project Structure' (Database Creation, 4 Core Keys, MVC Structure Creation, Uploading our Project on GitHub). Each card in the 'Design Project Structure' section has a status indicator: NA, AS, AS, and AA respectively.

3.10 GitHub Project

Account link: <https://github.com/Jira-Team-14/agile-project.git>

We used GitHub Project to be able as a team to pull the project, commit and push our changes and to be all aware of our code.

- Screen shots for our contribution on github

The screenshot shows the GitHub repository page for 'agile-project'. At the top, there's a navigation bar with 'Overview', 'Repositories 1', 'Projects', 'Packages', 'Teams', 'People 5', and 'Settings'. Below the navigation is a search bar with 'Find a repository...', filters for 'Type', 'Language', and 'Sort', and a green 'New repository' button. The main content area displays the repository 'agile-project' which is public and written in Python. It has 2 commits, 0 stars, 0 forks, and was updated 18 hours ago. A green line graph is shown to the right of the repository details.

The screenshot shows the GitHub repository page for 'agile-project' with a focus on recent activity. At the top, it shows 'main' branch, 12 branches, and 0 tags. There are buttons for 'Go to file', 'Add file', and 'Code'. A message says 'Your main branch isn't protected' with a 'Protect this branch' button. Below this, a list of recent commits is shown:

Author	Commit Message	Date
aGayar30	Merge pull request #5 from aGayar30/main	ba541f8 last week 28 commits
	.idea database connected commit 1	2 months ago
	Agile AP-63	last week
	trialDB project increment login page commit 1	2 months ago

At the bottom, there's a callout for 'Help people interested in this repository understand your project by adding a README.' with a 'Add a README' button.

Branches

Default branch

main Updated last week by aGayar30 Default

Your main branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. [Learn more](#)

Dismiss Protect this branch

Your branches

AP-25-as-an-admin-i-want-to-do-crud-operations-on-buildings-data-so-that-i-can-manage...	0 21	New pull request
AP-34-As-a-Floor-Admin-I-want-to-Update-Room-Status-so-that-I-will-be-able-to-decide...	0 0	New pull request
AP-39-As-a-user-I-want-to-login-to-Log-out-from-my-page-so-that-I-get-out-of-the-appl...	0 0	New pull request
AP-32-As-a-Floor-Admin-I-want-to-view-reservations-so-that-I-will-be-aware-of-the-res...	0 0	New pull request
AP-35-As-a-Receptionist-I-want-to-be-able-to-register-a-bed-to-the-client-so-that-I-w...	0 0	New pull request

[View more of your branches >](#)

Active branches

AP-25-as-an-admin-i-want-to-do-crud-operations-on-buildings-data-so-that-i-can-manage...	0 21	New pull request
AP-39-As-a-user-I-want-to-login-to-Log-out-from-my-page-so-that-I-get-out-of-the-appl...	0 0	New pull request
AP-37-As-a-Receptionist-I-want-to-view-alternative-building-occupancy-so-that-I-will...	0 0	New pull request
AP-36-As-a-Receptionist-I-want-to-retrieve-client-info-from-his-ID-so-that-I-will-che...	0 0	New pull request
AP-35-As-a-Receptionist-I-want-to-be-able-to-register-a-bed-to-the-client-so-that-I-w...	0 0	New pull request

[View more active branches >](#)

Commits, Pulls, and Pushes			
AP-33 Nada119 committed 18 hours ago		5dabdf2	
AP-34 Nada119 committed 18 hours ago		13e990a	
AP-35 Nada119 committed 18 hours ago		b2600bc	
AP-37 Nada119 committed 18 hours ago		4e25176	
AP-37 Nada119 committed 18 hours ago		e23a058	
AP-37 Nada119 committed 18 hours ago		6fed578	
AP-36 Nada119 committed 18 hours ago		0b74680	
Merge branch 'main' of https://github.com/agile-project-team-14/agile... ... Nada119 committed 18 hours ago		51ff246	
AP-36 Nada119 committed 18 hours ago		a6c34ac	
AP-36 final Nada119 committed 18 hours ago		b6f357a	
Commits on Dec 24, 2022			
Merge pull request #5 from aGayar30/main ... aGayar30 committed last week		ba541f8	
AP-63 aGayar30 committed last week		349fecc	
Merge pull request #4 from agile-project-team-14/AP-63 ... anassalah24 committed last week		7507e38	
git commit -m "AP-63 tmp" anassalah24 committed last week		cb55935	

Commits, Pulls, and Pushes			
Merge pull request #3 from agile-project-team-14/AP-21 ... anassalah24 committed 2 weeks ago		501c3d9	
AP-21 anassalah24 committed 2 weeks ago		6f415ce	
Commits on Dec 17, 2022			
AP-25 Nada119 committed 2 weeks ago		74b2be5	
AP-25 Nada119 committed 2 weeks ago		697f550	
AP-25 Nada119 committed 2 weeks ago		0ea48d7	
AP-25 Nada119 committed 2 weeks ago		367cae8	
AP-25 Nada119 committed 2 weeks ago		472c5bf	
Merge branch 'main' of https://github.com/agile-project-team-14/agile... ... Nada119 committed 2 weeks ago		323208f	
AP-25 worked? Nada119 committed 2 weeks ago		ad8e4b8	
Merge pull request #2 from agile-project-team-14/AP-21 ... anassalah24 committed 2 weeks ago		4c5a571	
Merge remote-tracking branch 'origin/AP-21' into AP-21 anassalah24 committed 2 weeks ago		05698f4	
AP-21 anassalah24 committed 2 weeks ago		8855329	
Merge pull request #1 from agile-project-team-14/AP-21 ... anassalah24 committed 2 weeks ago		f326820	
AP-21 trial commit anassalah24 committed 2 weeks ago		f52c907	

3.11 Clean Code

Writing cleaning code allows us to maintain and develop a piece of software fast in the long term.

- Give meaningful names to variables, functions, classes, and other entities in the code.

```
self.adminDashboardLabel = QtWidgets.QLabel(self.centralwidget)
self.adminDashboardLabel.setGeometry(QtCore.QRect(120, 100, 431, 71))
font = QtGui.QFont()
font.setFamily("Lucida Calligraphy")
font.setPointSize(26)
font.setBold(False)
font.setWeight(50)
self.adminDashboardLabel.setFont(font)
self.adminDashboardLabel.setStyleSheet("color: rgb(41, 84, 125);\n"
align: left;\n")

self.adminDashboardLabel.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtCore.Qt.AlignVCenter)
self.adminDashboardLabel.setObjectName("adminDashboardLabel")
self.ageLabel = QtWidgets.QLabel(self.centralwidget)
self.ageLabel.setGeometry(QtCore.QRect(130, 300, 91, 31))
font = QtGui.QFont()
font.setFamily("Lucida Calligraphy")
font.setPointSize(12)
font.setBold(False)
font.setWeight(50)
self.ageLabel.setFont(font)
self.ageLabel.setObjectName("ageLabel")
self.showNameLabel = QtWidgets.QLabel(self.centralwidget)
```

```
class addStakeholder:
    def __init__(self,connection):
        self.connection=connection

    def addEmployee(self,infoTuple):
        try:
            n = 'none'
            print(self.connection)
            cursor = self.connection.cursor()
            addBAquery = """
                INSERT INTO agile_project.employee VALUES (%s, %s, %s, %s, %s,
                """
            with self.connection.cursor() as cursor:
                isDone = cursor.execute(addBAquery, infoTuple)
                self.connection.commit()          #if you do not commit
                if str(isDone).lower() == n:      #the cursor return
                    return isDone
                else:
                    return "false"
        except Error as e:
            return e

    def addBuilding(self,infoTuple):
        try:
            n = 'none'
            print(self.connection)
            cursor = self.connection.cursor()
```

- Create functions that are small and do a single thing.

```
def back(self):
    self.Buildings_Managment = QtWidgets.QMainWindow()
    self.ui = BM.Ui_Buildings_Managment()
    self.ui.setupUi(self.Buildings_Managment)
    self.Buildings_Managment.show()
    close()
```

This Function to go the previous page

```
def logOut(self):
    self.LoginWindow = QtWidgets.QMainWindow()
    self.ui = G.Ui_LoginWindow()
    self.ui.setupUi(self.LoginWindow)
    self.LoginWindow.show()
```

This Function to go the logout only

```
def addBuilding(self):
    mc = main_control()
    result = mc.getGuiMsgsControl()
    if (self.idLineEdit.text() == ''):
        messagebox.showwarning('Warning', str(result[3]))
        return
    emptuple = (int(self.idLineEdit.text()), self.locationLineEdit.text(), self.totalFloorsLineEdit.text())
    isacceptable = mc.addBuildingControl(emptuple)
    if str(isacceptable).lower() == "none":
        messagebox.showinfo('information', str(result[4]))

    else:
        ermsg2 = str(result[5])
        ermsg1 = str(isacceptable)
        ermsg = ermsg2 + " " + ermsg1
        messagebox.showError('Error', ermsg)
```

This Function to add building

```
# functions to switch windows after login authentication

def showAdmin_Dashboard(self):
    self.Admin_Dashboard = QtWidgets.QMainWindow()
    self.ui = AD.Ui_Admin_Dashboard()
    self.ui.setupUi(self.Admin_Dashboard)
    self.Admin_Dashboard.show()

def showBuildingAdmin_Dashboard(self):
    self.buildingAdmin_Dashboard = QtWidgets.QMainWindow()
    self.ui = BD.Ui_buildingAdmin_Dashboard()
    self.ui.setupUi(self.buildingAdmin_Dashboard)
    self.buildingAdmin_Dashboard.show()

def showFloorAdmin_Dashboard(self):
    self.floorAdmin_Dashboard = QtWidgets.QMainWindow()
    self.ui = Ui_floorAdmin_Dashboard()
    self.ui.setupUi(self.floorAdmin_Dashboard)
    self.floorAdmin_Dashboard.show()

def showReceptionist_Dashboard(self):

    self.Resident_Info_Sucess = QtWidgets.QMainWindow()
    self.ui = RS.Ui_Resident_Info_Sucess()
    self.ui.setupUi(self.Resident_Info_Sucess)
    self.Resident_Info_Sucess.show()
    # Resident_Info.close()
```

This Function to show Admin Dashboard

This Function to show Building Admin Dashboard

This Function to show Floor Admin Dashboard

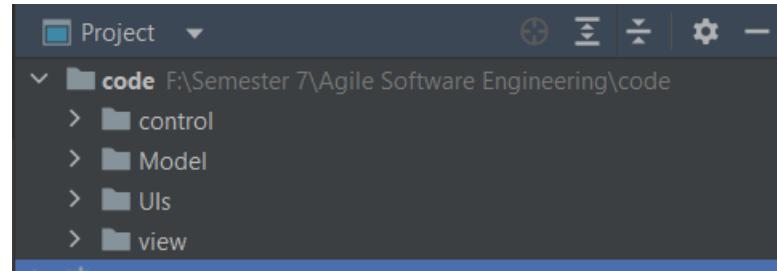
This Function to show Receptionist Dashboard

- Encapsulate related data and functions into small independent classes, Keep related code together and keep the lines smaller.

```
# This is a class for DATABASE TABLE "LOGIN" query methods
class logintable:
    def __init__(self,type,connection):
        self.type = type
        self.connection=connection
    def getloginquery(self):
        try:
            print(self.connection)
            cursor = self.connection.cursor()
            get_login_query = """
                SELECT id,password
                FROM agile_project.login
                WHERE type = {} ;""".format(self.type)
            with self.connection.cursor() as cursor:
                cursor.execute(get_login_query)
                result = cursor.fetchall()
                for row in result:
                    print(row)
            return result
        except Error as e:
            print(e)
            return e

    def gui_msgs(self):
        try:
            print(self.connection)
            cursor = self.connection.cursor()
            get_query = """
                SELECT msg FROM agile_project.gui_msgs;"""
            with self.connection.cursor() as cursor:
```

- Structure the code for better readability: As we said before we used MVC Structure



- Enhance readability with proper comments.

```
from getpass import getpass
from mysql.connector import connect, Error

# This is a class for adding employees and buildings into the db
class addStakeholder:
    def __init__(self, connection):
        self.connection = connection

    def addEmployee(self, infoTuple):
        try:
            n = 'none'
            print(self.connection)
            cursor = self.connection.cursor()
            addBQuery = """
                INSERT INTO agile_project.employee VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s);
            """
            with self.connection.cursor() as cursor:
                isDone = cursor.execute(addBQuery, infoTuple)
                self.connection.commit()           #if you do not commit the change is automatically rolledback
                if (str(isDone).lower() == n):      #the cursor returns the errors found during execution, returning none if it executed successfully
                    return isDone
                else:
                    return "false"
        except Error as e:
            return e
```

```
#call model to get building ID
def getBuildingIDControl(self, id):...

#call model to make sure building id exists in db buildings
def buildingIDCheckControl(self):...

# call MODEL to query login info and authenticate info
def checkloginControl(self, type, usertuple):...

# call MODEL to query adding info and authenticate insertion
def addEmployeeControl(self, usertuple):...

# call MODEL to query adding info and authenticate insertion
def addBuildingControl(self, usertuple):...

# call MODEL to query editing info and authenticate update
def editEmployeeControl(self, usertuple):...
```

- Write readable, fast, independent tests.

```
1 ►  def test_checklogin():...
3 ►  def test_add_building():...
5 ►  def test_add_employee():...
7 ►  def test_back():...
9 ►  def test_log_out():...
11 ► def test_add_building():...
```

- **Additional Points**

Avoid code repetitions.

Keep the code simple: Each module should do only one thing.

Hide implementation details.

Keep the code extensible.

Avoid unnecessary functionalities.

Stay consistent and avoid surprises in the code.

3.12 Additional Part

After we finished our three main models we are still communicating with customer to upgrade our project.

Review Meeting of Sprint 3:

2023-12-23 Sprint3 Review

 Created by Nada Amr, with a template
Last updated: just a moment ago • 1 min read

 Date
Dec 23, 2022

 Participants

- [@Nada Amr](#) [@Alaa Mohamed Mohamed Hamdy Ahmed](#) [@Jannahayman](#) [@Anas Salah](#)
[@ahmed02amr](#)
-

 Goals

- product backlog grooming
- Get to know feedback of the customer on the release

 Discussion topics

Time	Item	Presenter	Notes
12:00 PM	Review	Team 14	Sprint 3 release is done successfully
1:00 PM	Add new Features	Customer	The Customer asked for adding new feature to the app "Communication between Employees", this Communication types are: requests and complains.

Jira:

The Jira Backlog interface for the 'agile_project' project. The sidebar shows 'Issues without epic' under 'Booking System', 'Building Management', and 'Employee System'. The main area displays the 'AP Sprint 4' backlog (5 Dec – 19 Dec) with 3 issues: AP-68 (Database New Feature Modification), AP-87 (As a user I want to communicate with other users so that I can send to them requests or complains), and AP-94 (As a user I want to check the messages which have been sent to me so that I will be aware of the requests that I have). Below the sprint is the 'Backlog' section with 4 issues: AP-91, AP-93, AP-97, and AP-98. Each issue has a status bar indicating 'DONE', 'IN PROGRESS', 'TO DO', and 'AS'. A 'Create sprint' button is also present.

Story mapping:



Portfolio:

