



**COEN6311 Software Engineering**  
**Department: Gina Cody school of engineering and computer science**

**Mini Project**  
**Employee Task Management**

*Submitted by*  
**Ragul Nivash Rangasamy Sekar (40169564)**

*Submitted to*  
**Prof. Tariq Daradkeh**  
**TA Salah Harb**

## Content

Sr. No.	Topic	Page No.
1	Project description	3
2	User Stories	4
3	Business flow diagram	5
4	Use case diagram	6
5	Sequence diagram	7
6	Class diagram	8
7	Time report	9
8	Relation between code and diagram	10

# 1. Project Description

[https://github.com/RagulNivash/Employee\\_tast\\_management](https://github.com/RagulNivash/Employee_tast_management)

## 1.1.Overview.

For this project, I have built console-based application for employee task management in Python, which is then parsed according to task, employee and notification in the database. The company's hierarchy is as follows General manger >> Manager >> Worker. For instance, I have created one general manager (Tariq), two managers (Salah for backend and Ted for frontend) and four and three works in each department

## 1.2. General Manager

General manager can assign task to any of the two departmental managers or to any workers. They have access to full company report. General manager can create task. General manager has access to the notifications of the newly assigned task and have authorization to cancel the task.

## 1.3. Manager

Manager can assign task to their department workers, but limited to maximum of three task per worker. They can assign only within their department. In case if manager wants to cancel task, they need to get permission from General manager.

## 1.4.Worker

Worker can assign unassigned task to themselves of their department. They cannot work more than three task at once. Worker cannot cancel their task.

## 1.5. Project Goal

To build a console-based application to a company, allowing employees to manage their task according to their role.

## 2. User stories

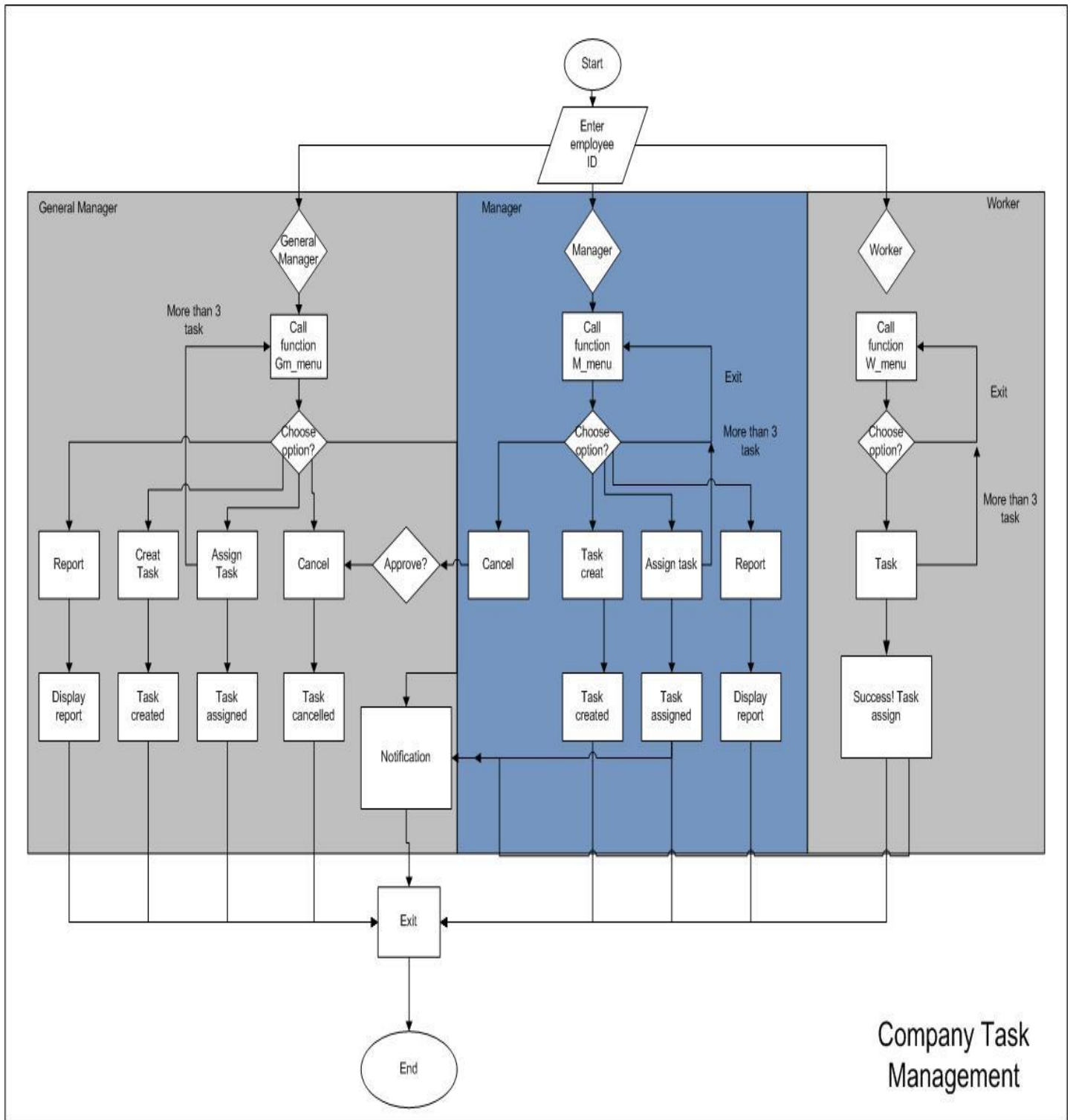
### 2.1.Epic: Task.

Role	User Story	Importance
General Manager	As a general manager, they can assign task to workers, so that they can accomplish the task.	High
	As a general manager, they can cancel the task, so that tasks can be managed much better.	High
	As a general manager, they can view the newly assigned task, so that they can construe the status of the task.	Low
	As a general manager, they can view the entire company's task report, so that they can understand the company's load.	Moderate
Manager	As a manger, they can assign task to their department workers, so that unassigned task can be completed.	High
	As a general manager, they can view their depart report, so that they can understand the department load.	Moderate
Worker	As a worker, they can assign unassigned task to themselves, so that company's task load can be reduced.	High
	As a worker, they can view unassigned task of their department, so that they can assign to themselves.	High

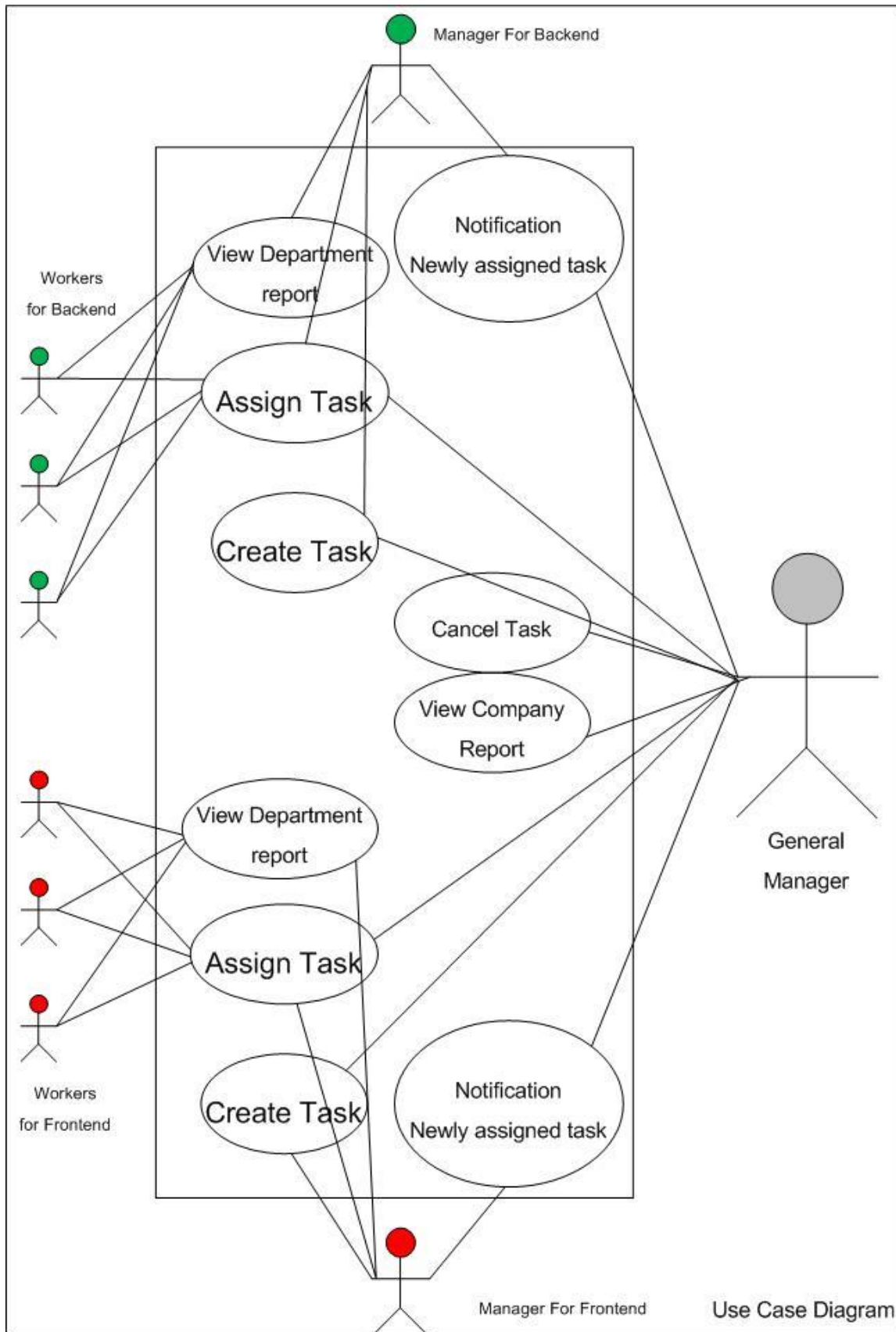
### 2.2. Epic: Limits

Feature	User Story	Importance
General Manager	As a general manager, they can assign only maximum of three task per worker, so that workers efficiency is maintained.	High
Manager	As a manger, they can assign only maximum of three task per worker of their department, so that workers efficiency is maintained.	High
	As a manger, they can assign task to workers only within their department, so that task with required skilled worker is assigned.	High
	As a manger, they need to get permission from the general manager to cancel the task, so that role's responsibility is maintained.	High
Worker	As a worker, they cannot assign more than three tasks to themselves, so that their efficiency is maintained.	High
	As a worker, they cannot cancel their task, so that role's responsibility is maintained.	High

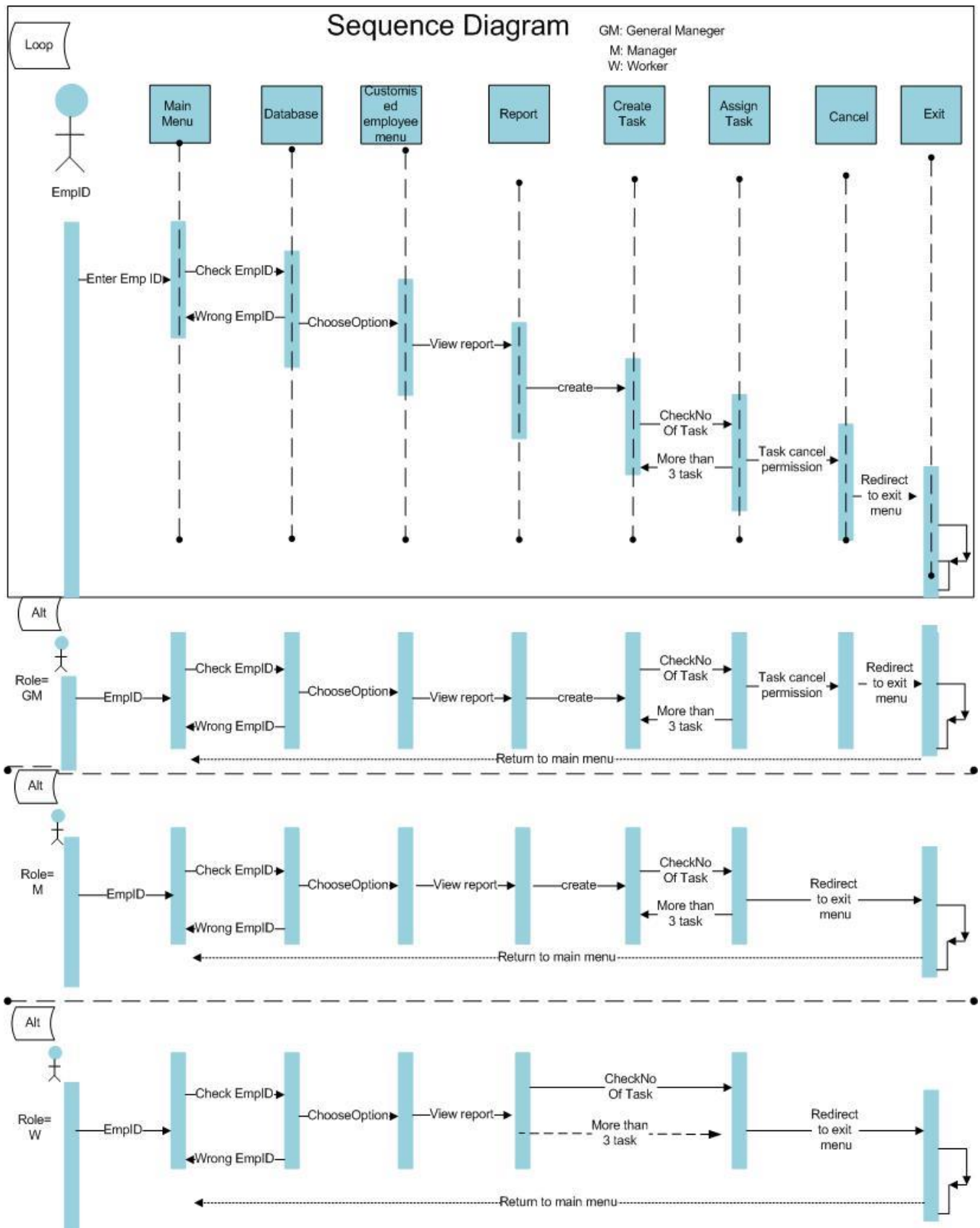
### 3. Business model diagram:



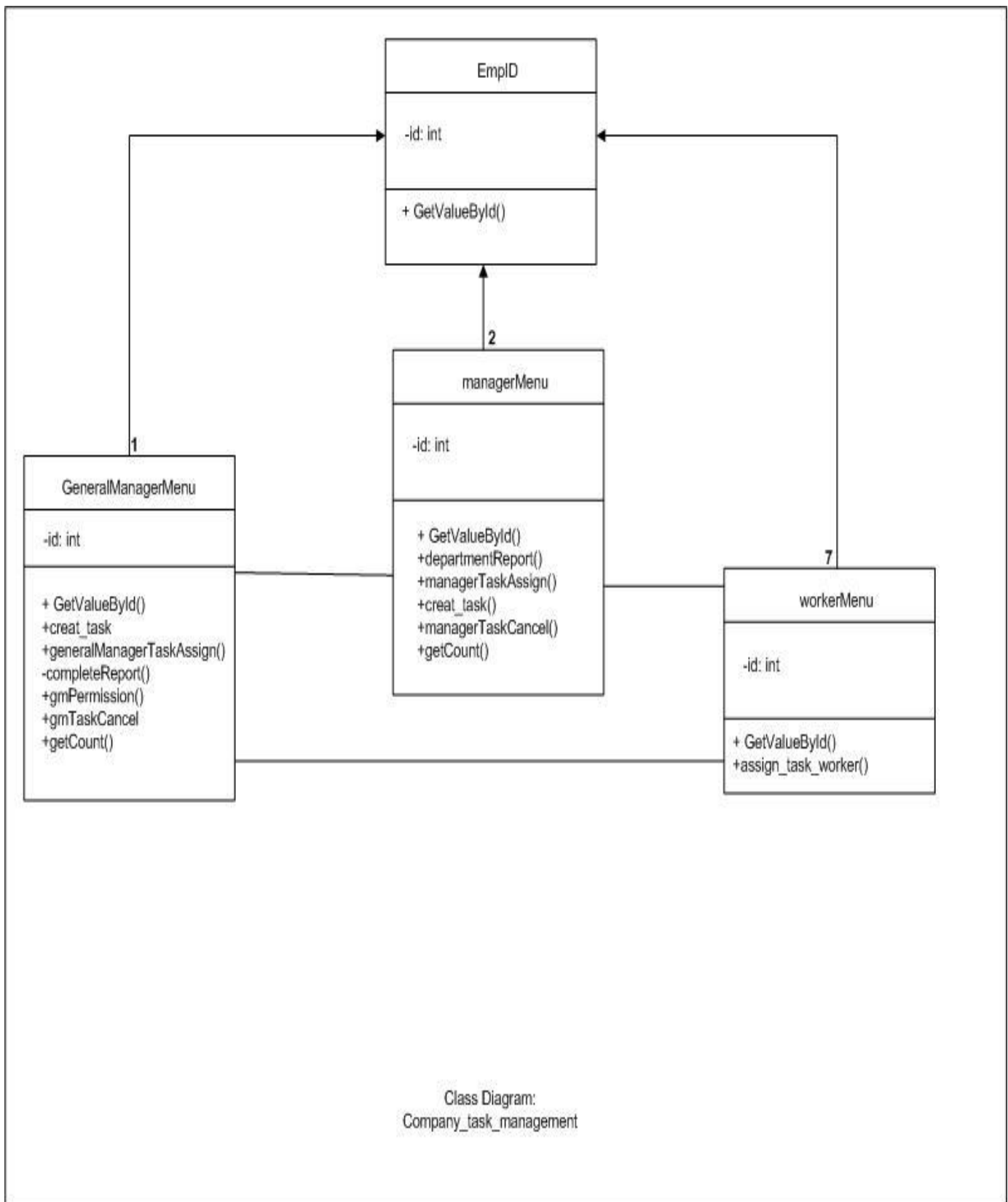
#### 4. Use case diagram



## 5. Sequence diagram



## 6. Class diagram



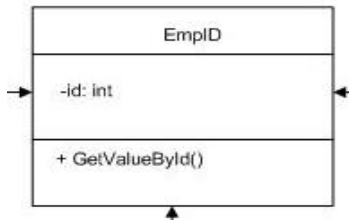


## 7. Time Report

Date	Agenda	Estimated Time	Actual Time
October 02, 2022	<ul style="list-style-type: none"> <li>Project Description Understanding</li> </ul>	25 Minutes	20 Minutes
October 03, 2022	<ul style="list-style-type: none"> <li>User Story</li> </ul>	30 Mintutes	15 Minutes
October 04, 2022	<ul style="list-style-type: none"> <li>User Story</li> </ul>	30 Minutes	30 Minutes
October 06, 2022	<ul style="list-style-type: none"> <li>Use case Diagram</li> </ul>	20 minutes	25 minutes
October 07, 2022	<ul style="list-style-type: none"> <li>My SQL setup and learning</li> </ul>	2 hours	4 hours
October 08, 2022	<ul style="list-style-type: none"> <li>DB for employees and tasks</li> </ul>	1 hour	30 minutes
October 09, 2022	<ul style="list-style-type: none"> <li>Coding for worker functionality</li> </ul>	2 hours	2 hours
October 12, 2022	<ul style="list-style-type: none"> <li>Coding for General manager and Manager functionality</li> </ul>	1 hour	30 minutes
October 13, 2022	<ul style="list-style-type: none"> <li>Coding for General manager and Manager functionality</li> </ul>	1 hour	2 hours 30 minutes
October 14, 2022	<ul style="list-style-type: none"> <li>Altering the code and checking all the modules</li> </ul>	1 hour	1 hour

October 16, 2022	<ul style="list-style-type: none"> <li>Business model diagram, sequence diagram, class diagram</li> </ul>	2 hours	2hours
October 17, 2022	<ul style="list-style-type: none"> <li>Report making</li> </ul>	1 hour	30 minutes
October 20, 2022	<ul style="list-style-type: none"> <li>Code altering</li> </ul>	45 minutes	1 hour 30 minutes
October 21, 2022	<ul style="list-style-type: none"> <li>Pushing the code in Github, Project submission.</li> </ul>	1 hour	1 hour

## 8. Relation between code and diagram

Diagram	Code
	<pre>emp_id = input("Enter your Employee id ") role = getValueById("employee", "role", emp_id) flag = True while(flag == True):     clear_output(wait=True)     if role == 3:         flag = generalManagerMenu(emp_id)     elif role == 2:         flag = managerMenu(emp_id)     elif role == 1:         flag = workerMenu(emp_id)     if flag != False:         flag = True</pre> <p>Methods:</p> <pre>def getValueById(table, column, emp_id):     conn = pymysql.connect(         host='localhost',         user='root',         password = "rootpassword@1",         db='employee',     )      cur = conn.cursor()     query = "select %s from %s where id = '%s'"%(column, table, emp_id)     cur.execute(query)     res = cur.fetchall()     value = res[0][0]     conn.close()     return value</pre>



```

generalManagerMenu(emp_id):
    print("General Manager Options:")
    print("1: view complete report")
    print("2: Create task")
    print("3: assign task to employee")
    print("4: View notifications")
    print("5:cancel task")
    print("6: exit")

    option = int(input())
    if option == 1:
        completeReport(emp_id)
    elif option == 2:
        create_task(emp_id)
    elif option == 3:
        generalManagerTaskAssign(emp_id)
    elif option == 4:
        gmPermissions(emp_id)
    elif option == 5:
        gmTaskCancel(emp_id)
    elif option == 6:
        return False

```

#### Methods:

```

def completeReport(emp_id):
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()
    query = "select * from tasks"
    cur.execute(query)
    res = cur.fetchall()
    print("All Tasks")
    print("-----")
    print("id | assigned by | department | status | assigned to | name ")
    for i in res:
        print("%s | %s | %s | %s | %s | %s"%(i[0],i[1],i[2],i[3],i[4],i[5]))
    print("-----")
    time.sleep(5)

def create_task(emp_id): #common for all employees
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()

    name = input("Enter task Name ")
    department = input("Enter department ")
    status = "unassigned"
    assigned_by = emp_id

    query = "insert into tasks(name,department,status,assigned_by) values"
    cur.execute(query)
    conn.commit()
    query = "insert into notifications(taskId ,type, description, requ"
    cur.execute(query)
    conn.commit()
    conn.close()
    print("task created successfully!")
    time.sleep(3)

```

```

def generalManagerTaskAssign(emp_id):
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )
    cur = conn.cursor()
    query = "select * from employee where role < 3"
    cur.execute(query)
    res = cur.fetchall()
    print("All Employees")
    print("-----")
    print("id | name | role | department | ")
    for i in res:
        print("%s | %s | %s | %s |"%(i[0],i[1],i[2],i[3]))
    print("-----")

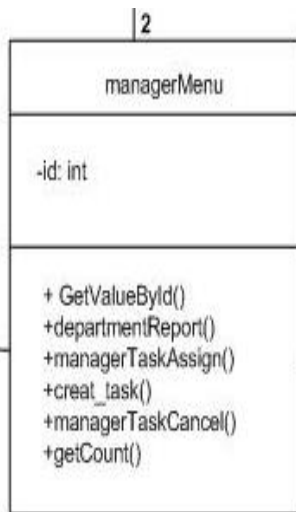
    departmentReport(emp_id)
    task_id = input("select task id to be assigned")
    id2 = input("enter employee id to assign task")
    if getCount(id2) > 2:
        print("Employee cannot have more than 3 tasks!")
        return 0
    query = "update tasks set status = 'assigned', assigned_to = %s where id = %s"
    cur.execute(query)
    conn.commit()
    print("Success!")
    time.sleep(3)

def gmPermissions(emp_id):
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )
    cur = conn.cursor()
    query = "select * from notifications "
    cur.execute(query)
    res = cur.fetchall()
    print("Notifications")
    print("-----")
    print("id | taskId | type | description | request | raisedBy | assignedTo ")
    for i in res:
        print("%s | %s | %s | %s | %s | %s | %s |"%(i[0], i[6],i[1],i[2],i[3],i[4],i[5],i[7]))
    print("-----")
    noti_id = input("enter id to take action")
    noti_type = getValueById("notifications","type",noti_id)

    if noti_type == "permission":
        status = input("Please enter A to accept and R to reject")
        query = "update notifications set request = '%s'"%(status)
        cur.execute(query)
        conn.commit()
        if status == 'A':
            task_id = getValueById("notifications", "taskId",noti_id)
            query = "delete from tasks where id = '%s'"%(task_id)
            cur.execute(query)
            conn.commit()

def gmTaskCancel(emp_id):
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )
    cur = conn.cursor()
    query = "select * from tasks"
    cur.execute(query)
    res = cur.fetchall()
    print("All Tasks")
    print("-----")
    print("id | assigned by | department | status | assigned to | name ")
    for i in res:
        print("%s | %s | %s | %s | %s | %s | %s |"%(i[0],i[1],i[2],i[3],i[4],i[5],i[6]))
    print("-----")
    # query = "select id from employee where role = 3"
    # cur.execute(query)
    # res = cur.fetchall()
    # gm_id = res[0][0]
    task_id = input("Enter id of task to be cancelled")
    query = "delete from tasks where id = '%s'"%(task_id)
    cur.execute(query)
    conn.commit()
    print("Success!")
    time.sleep(3)

```



```

managerMenu(emp_id):
    print("Manager Options:")
    print("1: view department report")
    print("2: assign task to department employee")
    print("3:Create task")
    print("4:cancel task")
    print("5: Exit")

    option = int(input())
    if option == 1:
        departmentReport(emp_id)
    elif option == 2:
        managerTaskAssign(emp_id)
    elif option == 3:
        create_task(emp_id)
    elif option == 4:
        managerTaskCancel(emp_id)
    elif option == 5:
        return False
  
```

## Methods

```

def departmentReport(emp_id):
    department = getValueById("employee", "department", emp_id)
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()
    query = "select * from tasks where department = '%s'"%(department)
    cur.execute(query)
    res = cur.fetchall()
    print("Department Tasks")
    print("-----")
    print("id | assigned by | department | status | assigned to | name")
    for i in res:
        print("%s | %s | %s | %s | %s | %s"%(i[0],i[1],i[2],i[3],i[4],i[5]))
    print("-----")
    # wait = input("press enter to continue")
    time.sleep(5)
  
```

```

def managerTaskAssign(emp_id):
    department = getValueById("employee", "department", emp_id)
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()
    query = "select * from employee where department = '%s'"%(department)
    cur.execute(query)
    res = cur.fetchall()
    print("Department Employees")
    print("-----")
    print("id | name | role | department | ")
    for i in res:
        if int(i[2]) == 1:
            print("%s | %s | %s | %s |"%(i[0],i[1],i[2],i[3]))
    print("-----")

    departmentReport(emp_id)
    task_id = input("select task id to be assigned")
    id2 = input("enter employee id to assign task")
    department_worker = getValueById("employee", "department", id2)
    if department_worker != department:
        print("Employee id entered doesnt belong to your department!!!!")
        return 0
    if getCount(id2) >2:
        print("Employee cannot have more than 3 tasks!")
  
```

```

def create_task(emp_id): #common for all employees
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

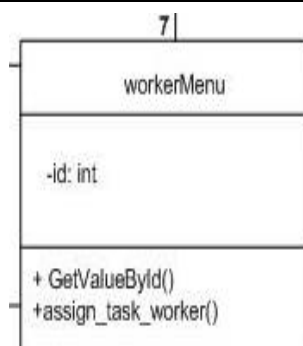
    cur = conn.cursor()

    name = input("Enter task Name ")
    department = input("Enter department ")
    status = "unassigned"
    assigned_by = emp_id

    query = "insert into tasks(name,department,status,assigned_by) values(%s,%s,%s,%s)"
    cur.execute(query)
    conn.commit()
    query = "insert into notifications(taskId,type,description,request,assigned_by) values(%s,%s,%s,%s,%s)"
    cur.execute(query)
    conn.commit()
    conn.close()
    print("task created successfully!")
    time.sleep(3)

def managerTaskCancel(emp_id):
    department = getValueById("employee","department",emp_id)
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )
    cur = conn.cursor()
    query = "select * from tasks where department = '%s'"%(department)
    cur.execute(query)
    res = cur.fetchall()
    print("Department Tasks")
    print("-----")
    print("id | assigned by | department | status | assigned to | name")
    for i in res:
        print("%s | %s | %s | %s | %s | %s"%(i[0],i[1],i[2],i[3],i[4],i[5]))
    print("-----")
    query = "select id from employee where role = 3"
    cur.execute(query)
    res = cur.fetchall()
    gm_id = res[0][0]
    task_id = input("Enter id of task to be cancelled")
    query = "select * from tasks where department = '%s' and id = '%s'"%(department,task_id)
    cur.execute(query)
    res = cur.fetchall()
    description = input("please enter reason for cancellation")
    query = "insert into notifications(taskId,type,description,request,assigned_by) values(%s,%s,%s,%s,%s)"

```



```

def workerMenu(emp_id):
    print("Worker Options:")
    print("1: assign task to self")
    print("2: Exit")

    option = int(input())
    if option == 1:
        assign_task_worker(emp_id)
    elif option == 2:
        return False

```

## Methods

```
def getValueById(table, column, emp_id):
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()
    query = "select %s from %s where id = '%s'"%(column,table,emp_id)
    cur.execute(query)
    res = cur.fetchall()
    value =res[0][0]
    conn.close()
    return value
```

```
def assign_task_worker(emp_id):
    if (getcount(emp_id)>2):
        print("you cannot undertake more than three tasks!")
        time.sleep(3)
        return 0
    conn = pymysql.connect(
        host='localhost',
        user='root',
        password = "rootpassword@1",
        db='employee',
    )

    cur = conn.cursor()
    query = "select * from tasks where status = 'unassigned' and assigned"
    cur.execute(query)
    res = cur.fetchall()
    print("-----")
    print("id | assigned by | department | status | assigned to | name")
    for i in res:
        print(i)
    print("-----")
    task_id = input("Enter task id to undertake")
    if task_id != None:
        query = "update tasks set status = 'assigned', assigned_to = %s"
        cur.execute(query)
        conn.commit()
        print ("Success!")
        time.sleep(3)
    conn.close()
```