

**Ho Chi Minh City University of Technology and Education**

**FACULTY FOR HIGH QUALITY TRAINING**

**INFORMATION TECHNOLOGY**

**PROJECT REPORT**

**TOPIC: LAB MEMBER MANAGEMENT**

**SUBJECT: WINDOW PROGRAMMING**





**Instructor : Huynh Xuan Phung**

**Group : Nguyen Van Dan - 17110010**

 **Nguyen Tan Phat - 17110059**

**Doan Viet Thuan - 17110080**

**Ho Chi Minh City, July 3rd, 2020**

SCORE

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Content | Present | Total |
| Score |  |  |  |

Instructor’s comments

**………………………………………………………………………………………………..………………………………………………………………….………………………….……………………………………………………………………………………………………………..……………………………………………………………………………..……………………..………………………………………………………………………..……………………………..………………………………………………………………..……………………………………..………………………………………………………..……………………………………………..………………………………………………..……………………………………………………..………………………………………..……………………………………………………………..………………………………..……………………………………………………………………..…………………………..**

Instructor

(***Sign and Fullname***)

**THANK YOU!**

In order to successfully complete this topic and report, we would like to express our sincere thanks to the lecturer, Huynh Xuan Phung, who has directly supported us throughout the process of making the topic. We feel that the teacher gave us advice to guide us in accordance with the requirements of the selected topic, always answer questions and give suggestions and corrections in time to help us overcome weaknesses. and complete well.

Due to limited time and limited knowledge, the implementation of the project certainly cannot avoid certain shortcomings. We'd love to hear from you so we can have a lot of experience implementing it to better complete our project.

Best regards!

*Ho Chi Minh city, July 3rd, 2020*

**Leader**

Phat

Nguyen Tan Phat

**CONTENTS**

[**List of images** 6](#_Toc44604321)

[**List of tables** 6](#_Toc44604322)

[**I. Introduction** 7](#_Toc44604323)

[**1. Purpose** 7](#_Toc44604324)

[**2. Input Information** 7](#_Toc44604325)

[**3. Feature** 8](#_Toc44604326)

[**4. Expected Interface** 8](#_Toc44604327)

[**II. Assignment for work** 9](#_Toc44604328)

[**III. Design** 9](#_Toc44604329)

[**1. Layer design** 9](#_Toc44604330)

[**A. List of entities used in the program** 9](#_Toc44604331)

[**B. Methods in one entities:** 10](#_Toc44604332)

[**2. Database Design:** 12](#_Toc44604333)

[**3. Field:** 12](#_Toc44604334)

[**3.1. tbUser entity:** 12](#_Toc44604335)

[**3.2. PersonalInformation entity:** 13](#_Toc44604336)

[**3.3. BasicInformation entity:** 13](#_Toc44604337)

[**3.4. Salary entity:** 13](#_Toc44604338)

[**3.5. Project entity:** 14](#_Toc44604339)

[**3.6. Task entity:** 14](#_Toc44604340)

[**3.7. Department:** 14](#_Toc44604341)

[**4. Interface Design:** 16](#_Toc44604342)

[**4.1. Login Form** 16](#_Toc44604343)

[**4.2. Account Form** 16](#_Toc44604344)

[**4.3. Department Form** 17](#_Toc44604345)

[**4.4. Project Form** 18](#_Toc44604346)

[**4.5. Task Form** 18](#_Toc44604347)

[**4.6. Salary Form** 19](#_Toc44604348)

[**4.7. Search Form** 19](#_Toc44604349)

[**IV. Testing** 20](#_Toc44604350)

[**1. Add account** 20](#_Toc44604351)

[**2. Delete account** 21](#_Toc44604352)

[**3. Update account** 22](#_Toc44604353)

[**4. Reload account** 23](#_Toc44604354)

[**5. Add department** 24](#_Toc44604355)

[**6. Delete department** 25](#_Toc44604356)

[**7. Update department** 26](#_Toc44604357)

[**8. Reload department** 27](#_Toc44604358)

[**9. Add project** 28](#_Toc44604359)

[**10. Delete project** 29](#_Toc44604360)

[**11. Update project** 30](#_Toc44604361)

[**12. Reload project** 31](#_Toc44604362)

[**13. Add task** 32](#_Toc44604363)

[**14. Delete task** 33](#_Toc44604364)

[**15. Update task** 34](#_Toc44604365)

[**16. Reload task** 35](#_Toc44604366)

[**17. New salary** 36](#_Toc44604367)

[**18. Edit salary** 37](#_Toc44604368)

[**19. Delete salary** 38](#_Toc44604369)

[**20. Search by ID staff** 39](#_Toc44604370)

[**21. Search by Fullname** 40](#_Toc44604371)

[**22. Search by IC** 41](#_Toc44604372)

[**23. Search by ProjectID** 42](#_Toc44604373)

[**V. Conclusion** 42](#_Toc44604374)

[**1. Rate of completion** 42](#_Toc44604375)

[**2. Problem** 42](#_Toc44604376)

[**3. Advantage and disadvantage** 42](#_Toc44604377)

[**a. Advantage** 42](#_Toc44604378)

[**b. Disadvantage** 42](#_Toc44604379)

[**4. Development direction** 43](#_Toc44604380)

List of images

[1. Login Form 16](#_Toc44604343)

[2. Account Form 16](#_Toc44604344)

[3. Department Form 17](#_Toc44604345)

[4. Project Form 18](#_Toc44604346)

[5. Task Form 18](#_Toc44604347)

[6. Salary Form 19](#_Toc44604348)

[7. Search Form 19](#_Toc44604349)

[8. Add account 20](#_Toc44604351)

[9. Delete account 21](#_Toc44604352)

[10. Update account 22](#_Toc44604353)

[11. Reload account 23](#_Toc44604354)

[12. Add department 24](#_Toc44604355)

[13. Delete department 25](#_Toc44604356)

[14. Update department 26](#_Toc44604357)

[15. Reload department 27](#_Toc44604358)

[16. Add project 28](#_Toc44604359)

[17. Delete project 29](#_Toc44604360)

[18. Update project 30](#_Toc44604361)

[19. Reload project 31](#_Toc44604362)

[20. Add task 32](#_Toc44604363)

[21. Delete task 33](#_Toc44604364)

[22. Update task 34](#_Toc44604365)

[23. Reload task 35](#_Toc44604366)

[24. New salary 36](#_Toc44604367)

[25. Edit salary 37](#_Toc44604368)

[26. Delete salary 38](#_Toc44604369)

[28. Search by ID staff 39](#_Toc44604370)

[29. Search by Fullname 40](#_Toc44604371)

[30. Search by IC 41](#_Toc44604372)

[31. Search by ProjectID 42](#_Toc44604373)

List of tables

[1. tbUser entity: 12](#_Toc44604335)

[2. PersonalInformation entity: 13](#_Toc44604336)

[3. BasicInformation entity: 13](#_Toc44604337)

[4. Salary entity: 13](#_Toc44604338)

[5. Project entity: 14](#_Toc44604339)

[6. Task entity: 14](#_Toc44604340)

[7. Department: 14](#_Toc44604341)

I. Introduction

1. Purpose

In order to facilitate the management of employees in a department of a company, we have built a lab member management system.

To build an optimal Lab member management system to manage the tasks in one system.

Admin can easily manage employees, projects, salaries, ... with the functions already integrated in the application.

2. Input Information

|  |  |
| --- | --- |
| Object | Input Information |
| User (admin) | Username  Pass  Role  Email  Birthday |
| Employee | ProjectID  DepartmentID  ID  Fullname  Birthday  Gender  IC  Position  Note |
| Project | ProjectID  ProjectName  Deadline  StartDate  Note |
| Task | ProjectID  TaskName  ID  Description  TaskID |
| Salary | ID  DepartmentName  FullName  BasicSalary  Reward  WorkingDay  DayOff  Salary  Note |
| Department | ProjectID  DepartmentID  DepartmentName  EstablishDate  Note |

3. Feature

|  |  |
| --- | --- |
| Main function | Oject |
| Reload | Employee, Project, Task, Salary, Department |
| Add | Employee, Project, Task, Salary, Department |
| Update | Employee, Project, Task, Salary, Department |
| Delete | Employee, Project, Task, Salary, Department |
| Calculate | Salary |
| Search | Employee, Project |

4. Expected Interface

- Login form

- Personal information form

- Project form

- Task form

- Salary form

- Department form

II. Assignment for work

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Full Name | Task description | Percent contribution |
| 17110010 | Nguyen Van Dan | - Design database - Design application interface  - Code functions in Personal information form, Task form, Department form | 33% |
| 17110059 | Nguyen Tan Phat | -Design database -Design application interface  -Code functions in Project form, Basic Information form, Salary form | 34% |
| 17110080 | Doan Viet Thuan | -Design database -Design application interface  -Code functions in Login form, Search form, Project form | 33% |

III. Design

1. Layer design

### A. List of entities used in the program

|  |  |  |  |
| --- | --- | --- | --- |
| Oder | Entity’s name | Purpose | Student |
| 1 | User | Using as an account to access into program.  Interact with User table | Thuan |
| 2 | PersonalInformaton | Interact with  PersonalInformation table | Dan |
| 3 | BasicInformation | Interact with BasicInformation table | Phat |
| 4 | Project | Interact with Project table | Phat, Thuan |
| 5 | Task | Interact with Task table | Dan |
| 6 | Salary | Interact with Salary table | Phat |
| 7 | Department | Interact with Department table | Dan |
| 8 | Search | Find out project information by project ID | Thuan |

### B. Methods in one entities:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order | Method | Purpose | File’s name | Student |
| 1 | Add | Insert a new account to User table | frmSignup.cs | Thuan |
| 2 | Update | Modify account information of the given User table | frmSignup.cs | Thuan |
| 3 | Delete | Remove account from User table | frmSignup.cs | Thuan |
| 4 | Reload | Refresh account | frmSignup.cs | Thuan |
| 5 | Add | Insert a personal information to PersonalInformation table | frmPersonalInformation.cs | Dan |
| 6 | Update | Modify personal information of the given PersonalInformation table | frmPersonalInformation.cs | Dan |
| 7 | Delete | Remove personal information from PersonalInformation | frmPersonalInformation.cs | Dan |
| 8 | Reload | Refresh personal information | frmPersonalInformation.cs | Dan |
| 9 | New | Refresh fill status | frmInformation.cs | Phat |
| 10 | Add | Insert employee’s information to BasicInformation table | frmInformation.cs | Phat |
| 11 | Update | Modify employee’s information of the given BasicInformation table | frmInformation.cs | Phat |
| 12 | Delete | Remove employee’s information from BasicInformation table | frmInformation.cs | Phat |
| 13 | Reload | Refresh project information | frmProject.cs | Phat |
| 14 | Add | Insert project information to Project table | frmProject.cs | Thuan |
| 15 | Update | Modify project information of the given Project table | frmProject.cs | Phat |
| 16 | Delete | Remove project infromation from Project table | frmProject.cs | Thuan |
| 17 | Reload | Refresh task information | frmTask.cs | Dan |
| 18 | Add | Insert task information to Task table | frmTask.cs | Dan |
| 19 | Update | Modify task information of the given Task table | frmTask.cs | Dan |
| 20 | Delete | Remove task infromation from Task table | frmTask.cs | Dan |
| 21 | New | Refresh fill status | frmSalary.cs | Phat |
| 22 | Edit | Modify salary information from Salary table | frmSalary.cs | Phat |
| 23 | Delete | Remove salary information from Salary table | frmSalary.cs | Phat |
| 24 | Calculate | Calculate salary for each employee | frmSalary.cs | Phat |
| 25 | Reload | Refresh department information | frmDepartment.cs | Dan |
| 26 | Add | Insert department information to Department table | frmDepartment.cs | Dan |
| 27 | Update | Modify department information of the given Department table | frmDepartment.cs | Dan |
| 28 | Delete | Remove department information from  Department table | frmDepartment.cs | Dan |
| 29 | openconnection | Connect to database | Clsdatabase.cs | Phat |
| 30 | closeconnection | Close the connection to database | Clsdatabase.cs | Phat |
| 31 | getconnectionString | Get the connection string | Clsdatabase.cs | Phat |

2. Database Design:

|  |  |  |
| --- | --- | --- |
| Order | Table’s name | Purpose |
| 1 | tbUser | Save and show account information |
| 2 | PersonalInformation | Save and show personal information |
| 3 | BasicInformation | Save and show employee information |
| 4 | Salary | Save and show salary for each employee |
| 5 | Project | Save and show project information |
| 6 | Task | Save and show task information |
| 7 | Department | Save and show department information |

3. Field:

### 3.1. tbUser entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| Username | nvarchar(50) |  | User name |
| Pass | nvarchar(50) |  | Password name |
| Role | nchar(10) |  | Role for each account |
| Email | nvarchar(50) |  | Email information |
| Birthday | date |  | Birthday information |

### 3.2. PersonalInformation entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ID | nvarchar(50) | x | Person Code |
| Fullname | nvarchar(50) |  | Person name |
| PlaceOfBirth | nvachar(50) |  | Place of birth |
| Address | nvarchar(50) |  | Address information |
| Phone | char(12) |  | Phonenumber information |
| Nation | nvarchar(30) |  | Ethnic information |
| Religion | nvarchar(20) |  | Religion information |
| Education | nvarchar(30) |  | Education information |
| Note | nvarchar(50) |  | Some specific notes. |

### 3.3. BasicInformation entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ProjectID | char(10) |  | Project code |
| DepartmentID | char(10) |  | Department name |
| ID | nvachar(50) | x | Employee code |
| Fullname | nvarchar(50) |  | Employee’s name |
| Birthday | Date |  | Birthday information |
| Gender | nvarchar(3) |  | Geder information |
| IC | navarchar(50) |  | Indentity Card |
| Position | nvarchar(50) |  | Position information |
| Note | nvarchar(100) |  | Some specific notes. |

### 3.4. Salary entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ID | nvarchar(50) | x | Employee Code |
| DepartmentName | nvarchar(50) |  | Department name |
| FullName | nvachar(50) |  | Employee name |
| BasicSalary | int |  | Basic salary information |
| Reward | nvarchar(50) |  | Reward salary information |
| WorkingDay | int |  | Working day information |
| DayOff | int |  | The number of Dayoff information |
| Salary | int |  | Salary information |
| Note | nvarchar(50) |  | Some specific notes. |

### 3.5. Project entity:

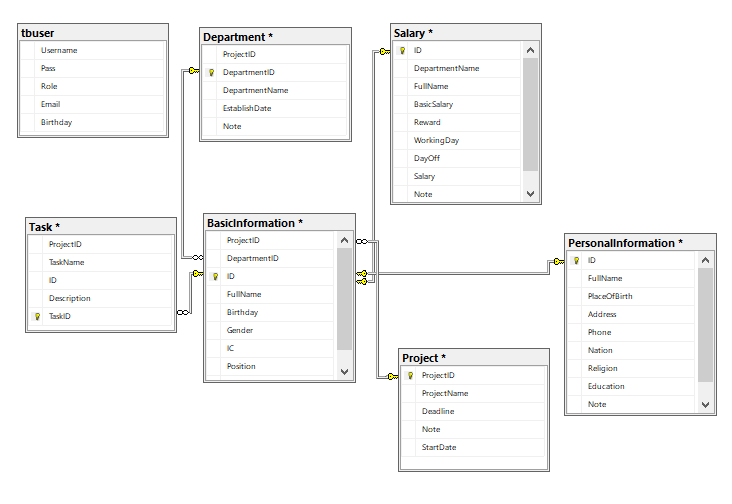
|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ProjectID | char(10) |  | Project code |
| ProjectName | nchar(10) |  | Project name |
| Deadline | datetime |  | Project end date |
| StartDate | datetime |  | Project start date |
| Note | nchar(10) |  | Some specific notes |

### 3.6. Task entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ProjectID | char(10) |  | Project code |
| TaskName | nchar(10) |  | Task name |
| ID | nvachar(50) |  | Employee code |
| Description | nvarchar(50) |  | Some specific notes |
| TaskID | nchar(10) |  | Task code |

### 3.7. Department:

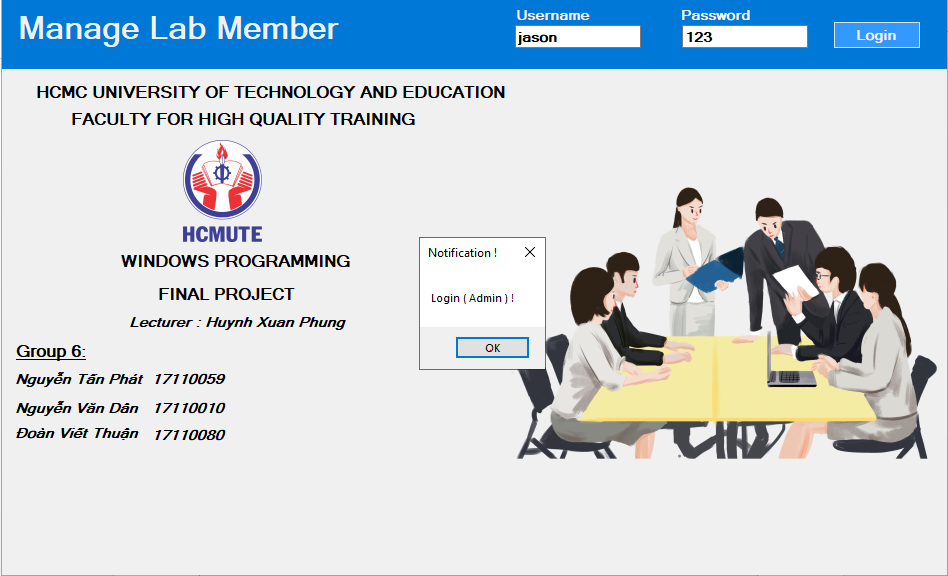
|  |  |  |  |
| --- | --- | --- | --- |
| Entity name | Data type | Primary key | Decription |
| ProjectID | char(10) |  | Project code |
| DepartmentID | char(10) |  | Department code |
| DepartmentName | nvachar(50) |  | Department name |
| EstablishDate | datetime |  | Establish day |
| Note | nvarchar(50) |  | Some specific notes |



Database Diagram Image

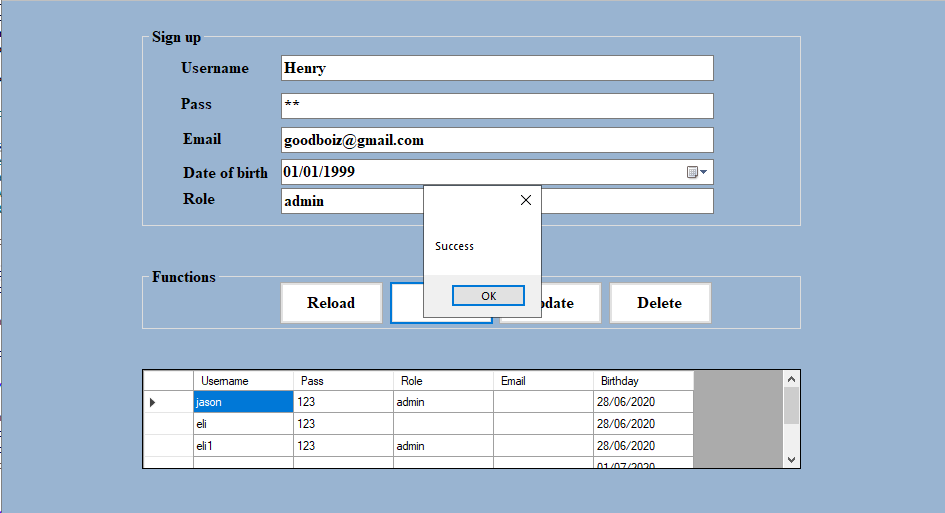
4. Interface Design:

### 4.1. Login Form



Sign in by admin authorities at the first time.

### 4.2. Account Form



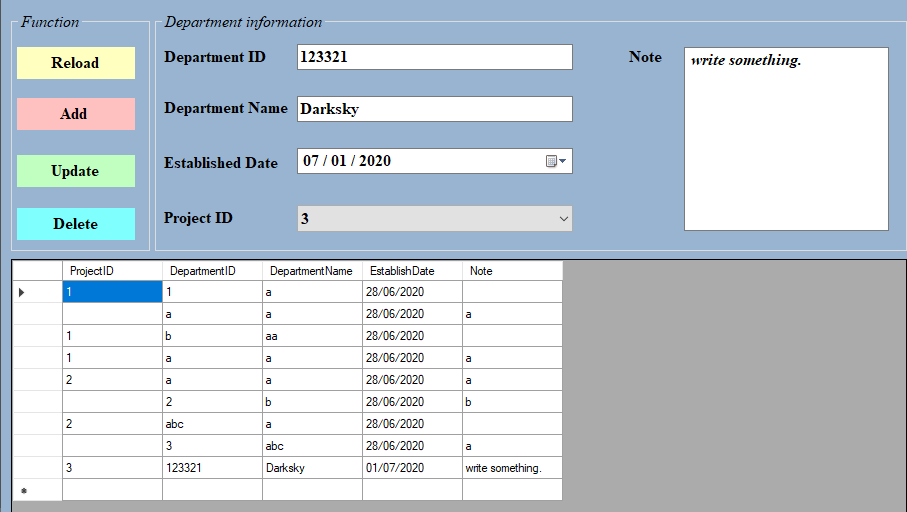
- The account form allows user to register account.

- This form displays accounts information such as username, password, email, date of birth ,role.

- Function: Reload, Add, Upadate, Delete

After sign up sucessfully, the program will allows user accesses to it.

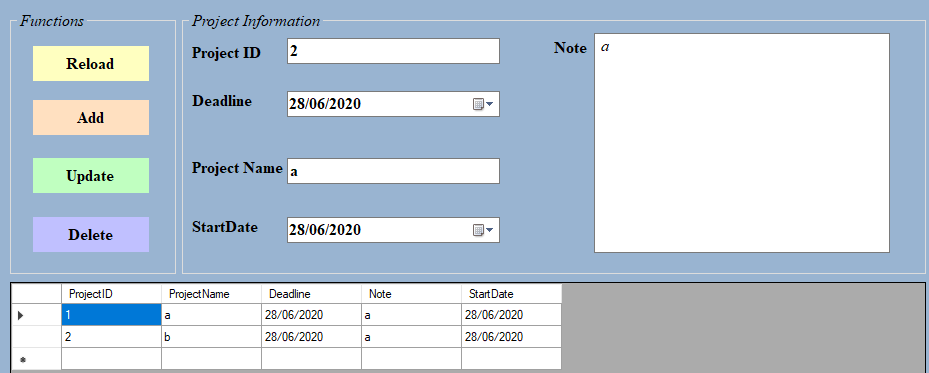
### 4.3. Department Form



- The department form shows department information such as Department ID, Department Name, Established Date, Project ID, Note.

- Function: Reload, Add, Update, Delete.

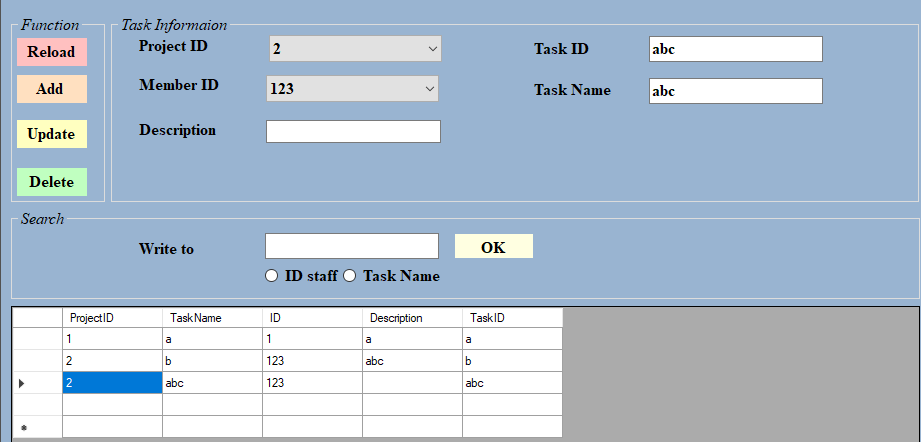
### 4.4. Project Form



- The project form shows the project information such as Project ID, Deadline, Project Name, Start Date, Note.

- Function: Reload, Add, Update, Delete.

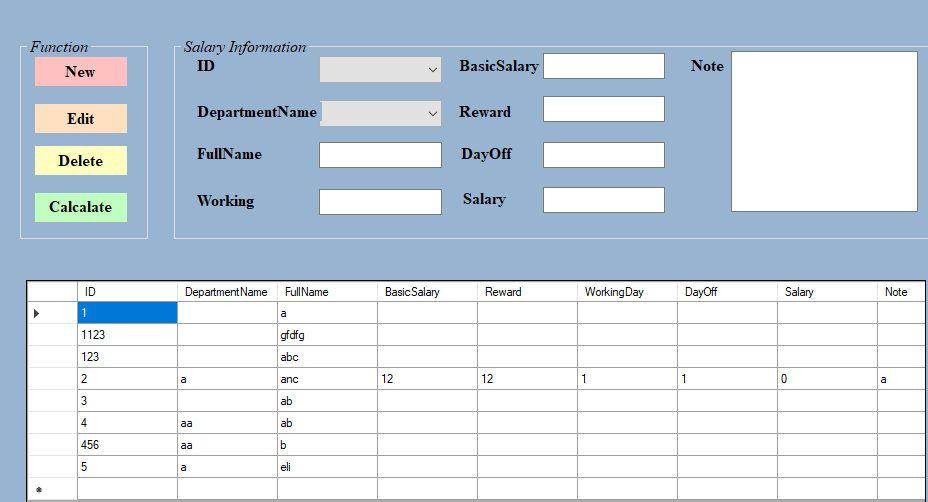
### 4.5. Task Form



- The task form displays the task information about Project ID, Member ID, Task ID, Task Name, Description.

- Function: Reload, Add, Update, Delete.

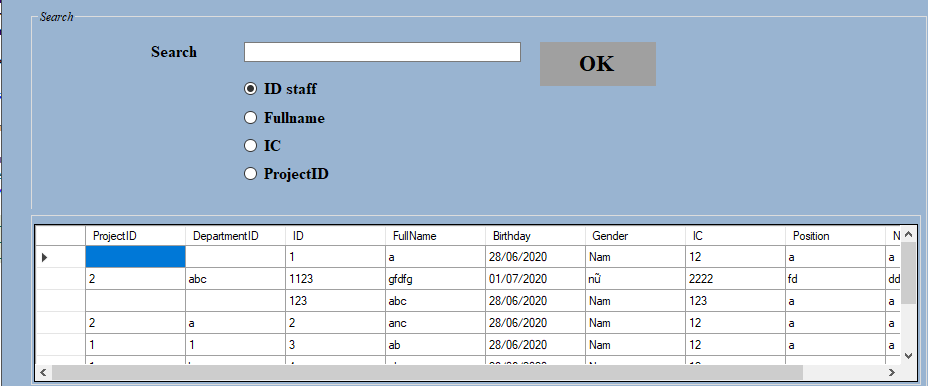
### 4.6. Salary Form



- The salary form displays salary information for each employee such as employee ID, employee name, department name, working, basic salary, reward, day off, salary, note.

- Function: New, Edit, Delete, Calculate.

### 4.7. Search Form

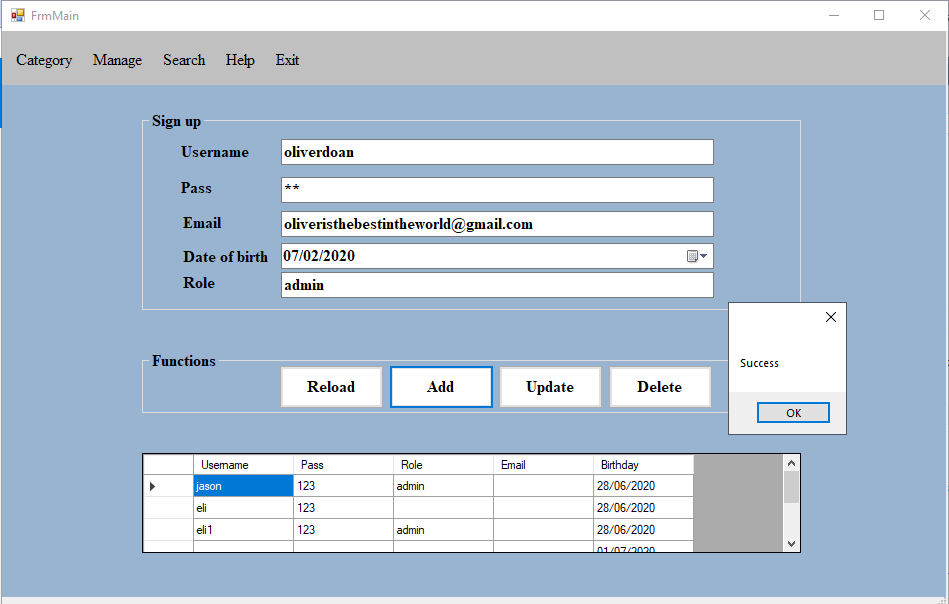


- The search form shows search information. Searching by ID staff, Fullname, IC (Indentity Card), ProjectID.

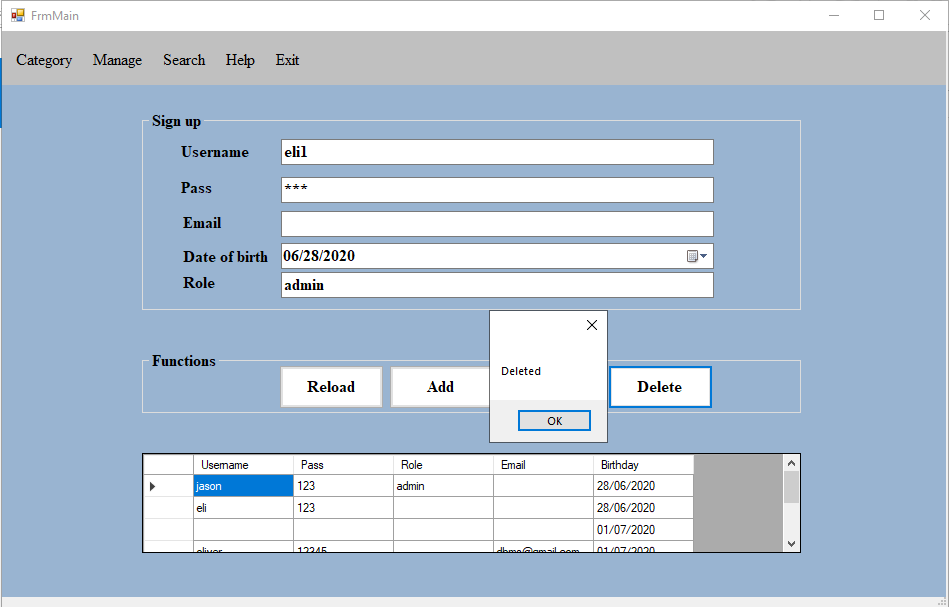
- This form will displayes information specifically for each type of search.

IV. Testing

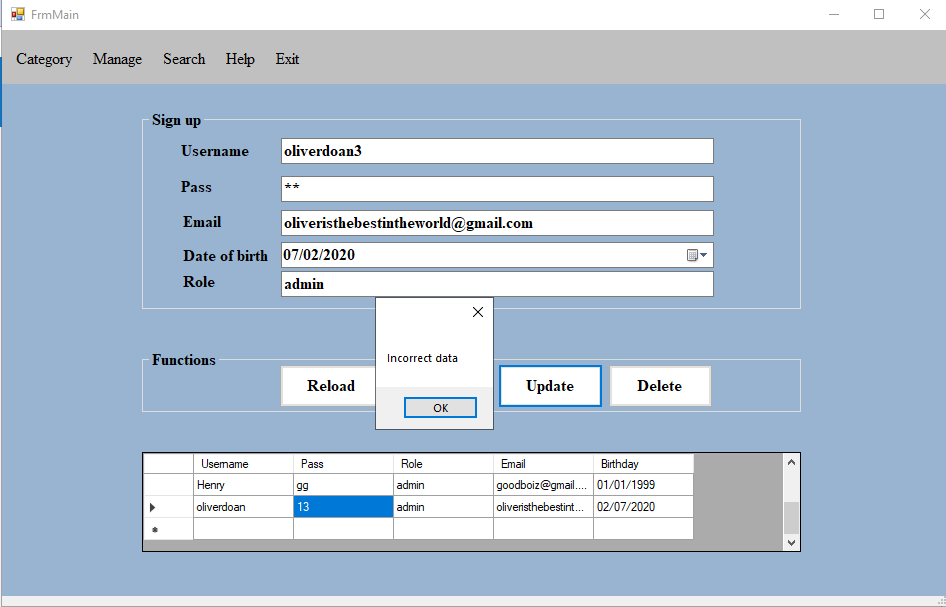
1. Add account



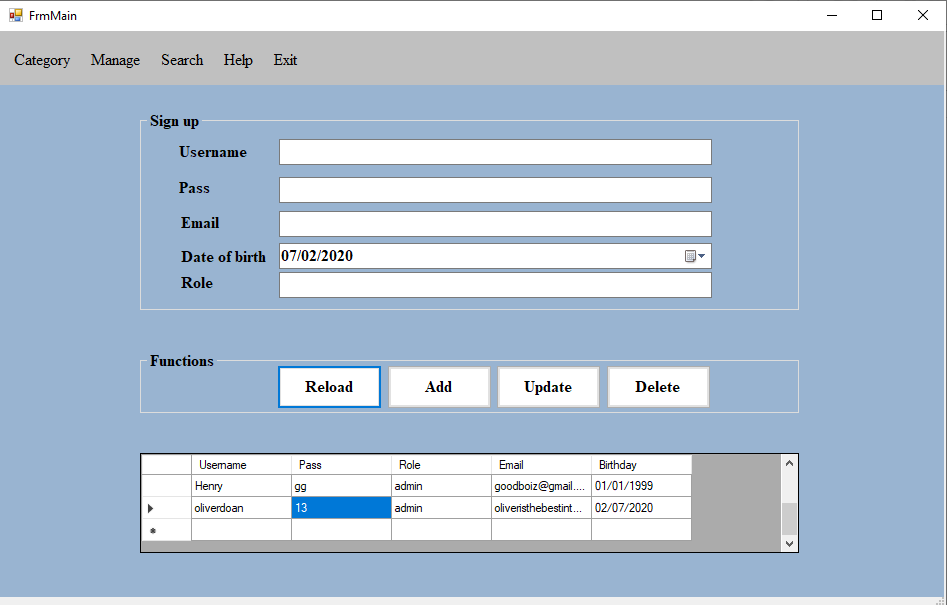
2. Delete account



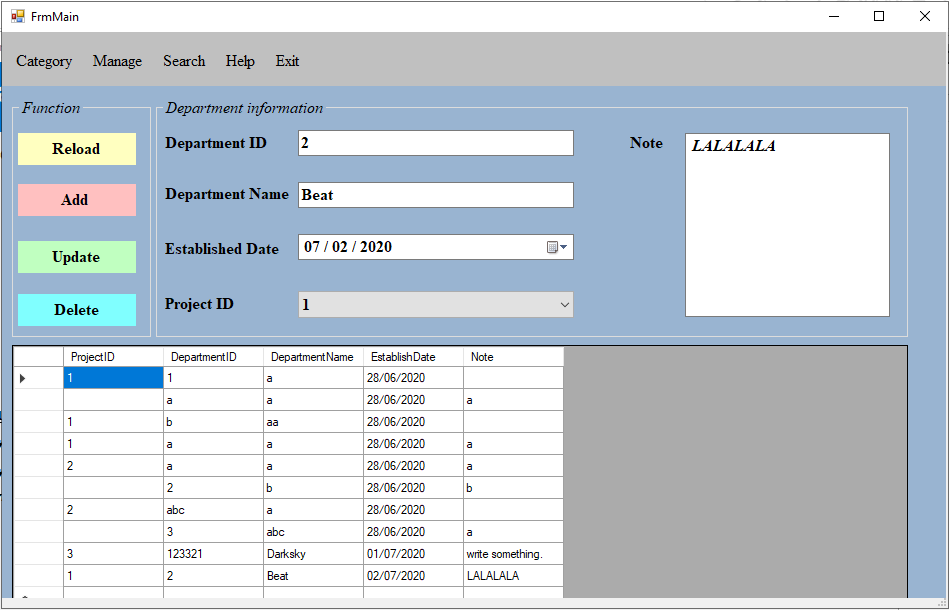
3. Update account



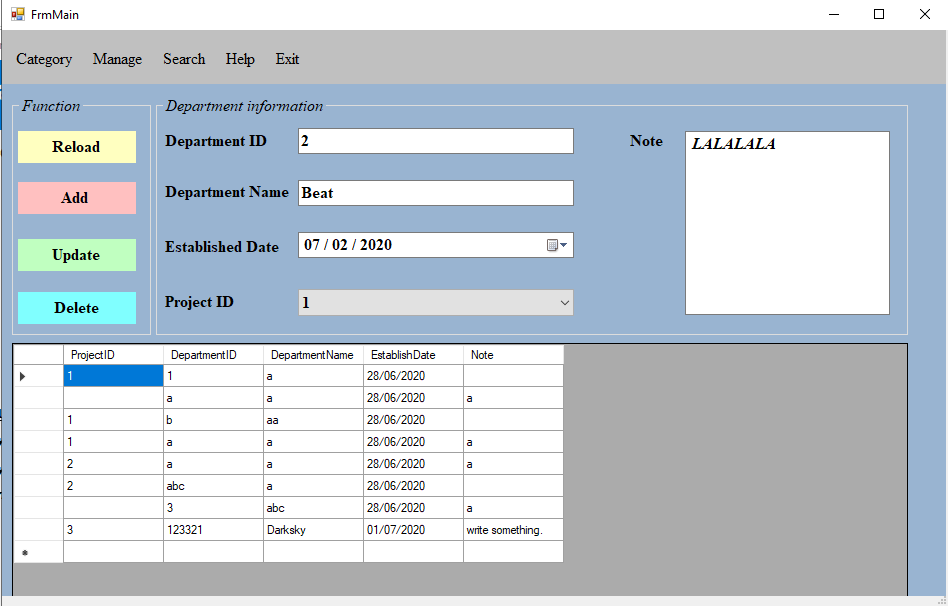
4. Reload account



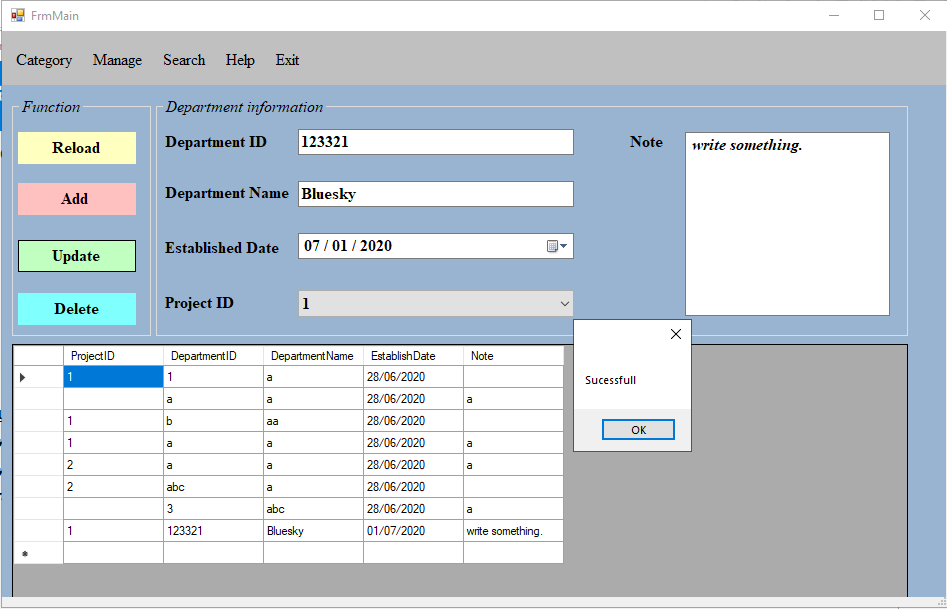
5. Add department



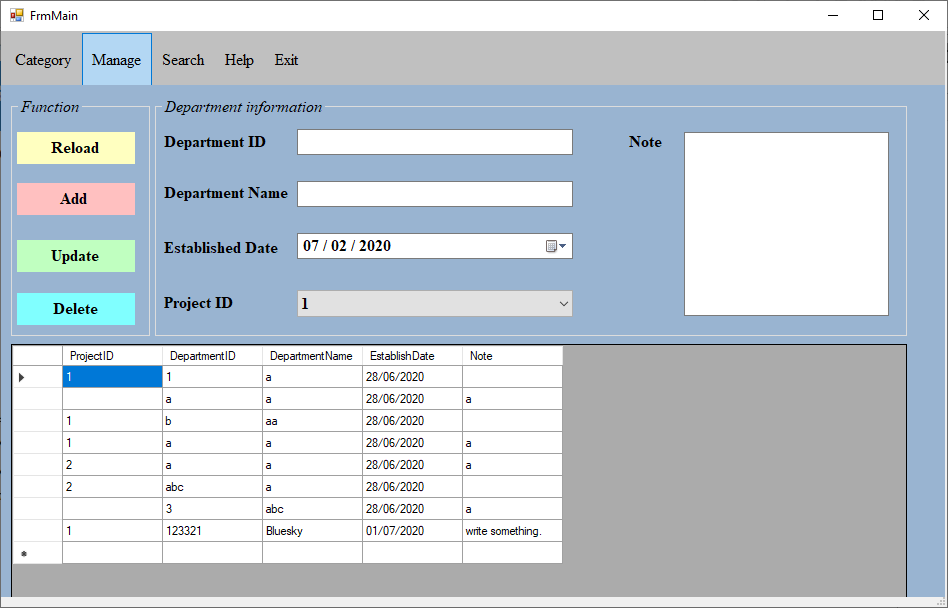
6. Delete department



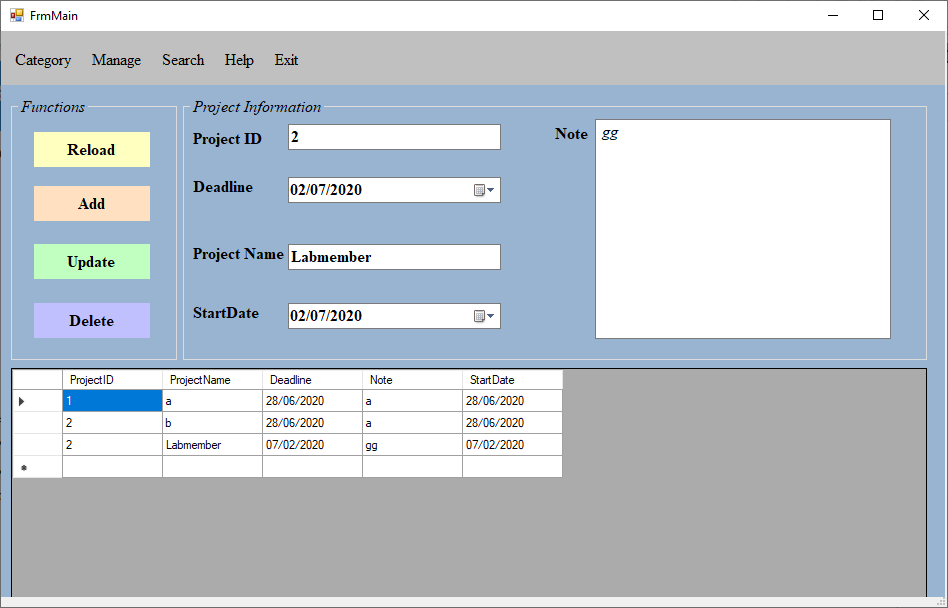
7. Update department



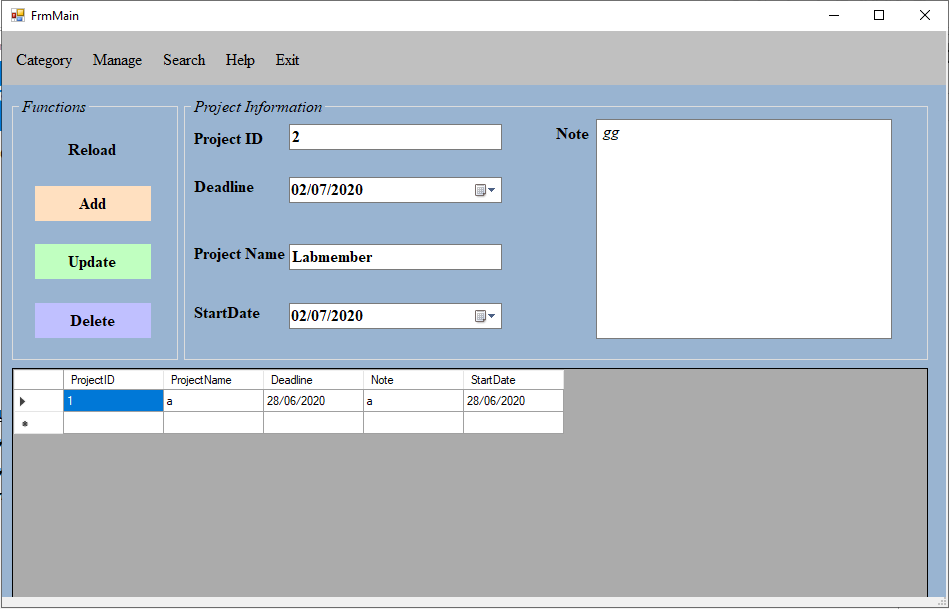
8. Reload department



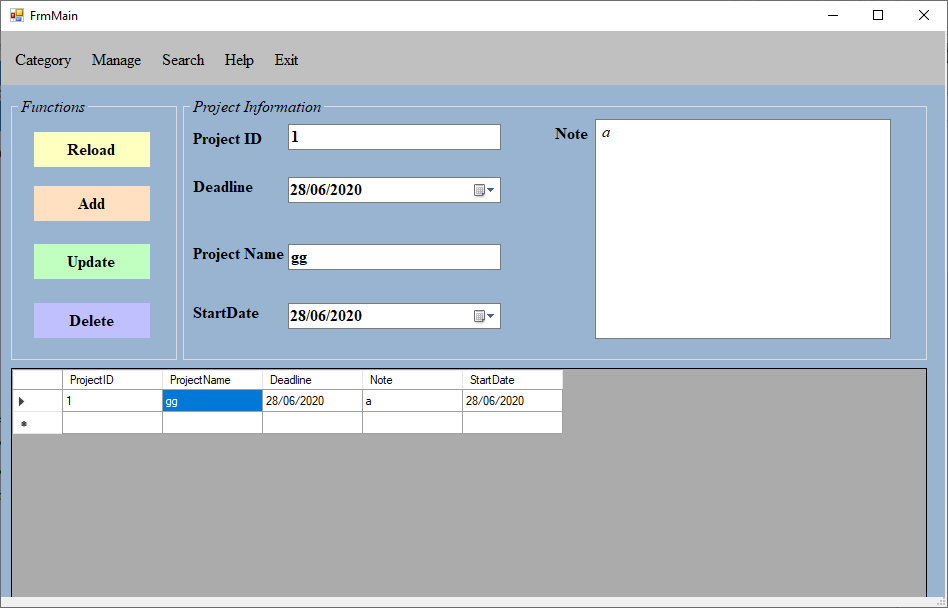
9. Add project



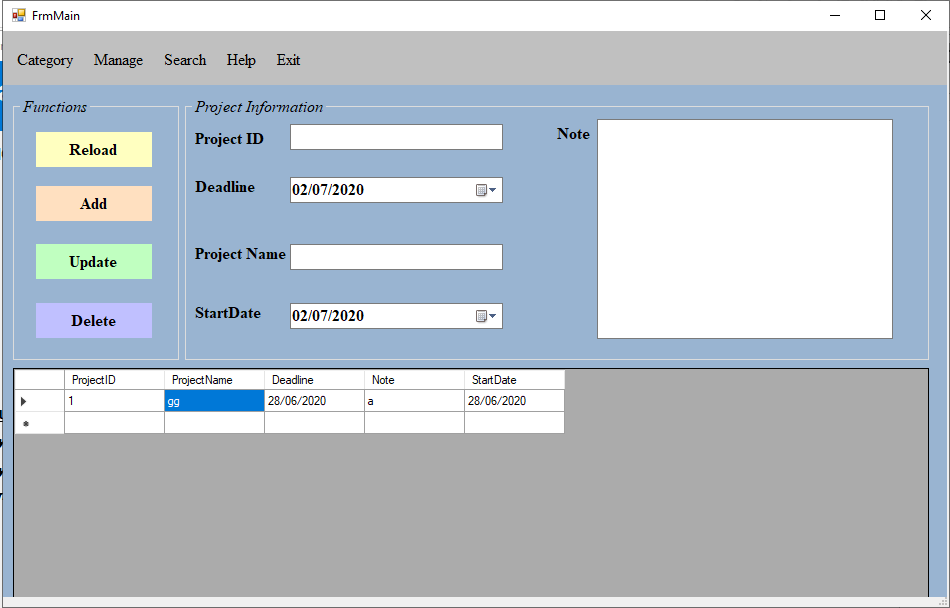
10. Delete project



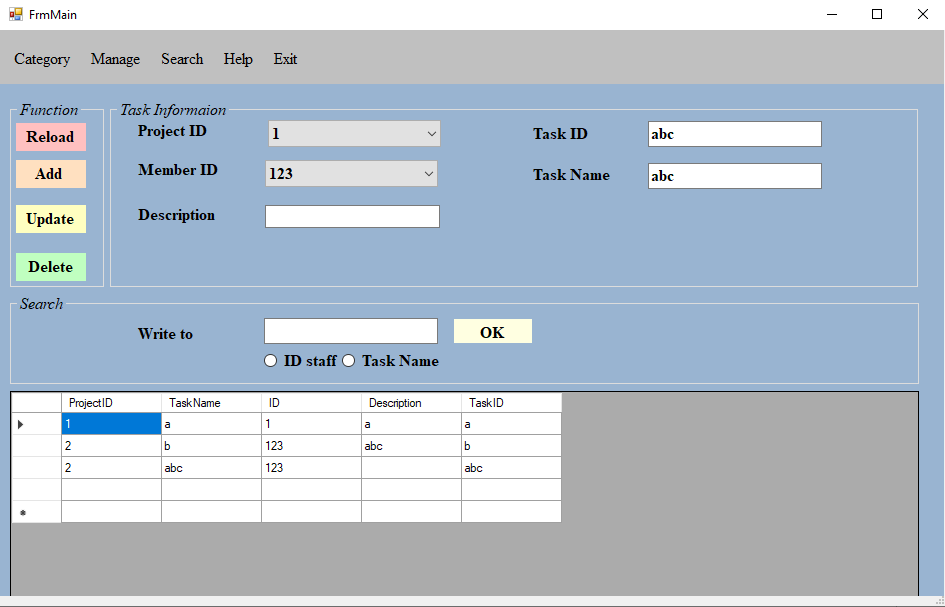
11. Update project



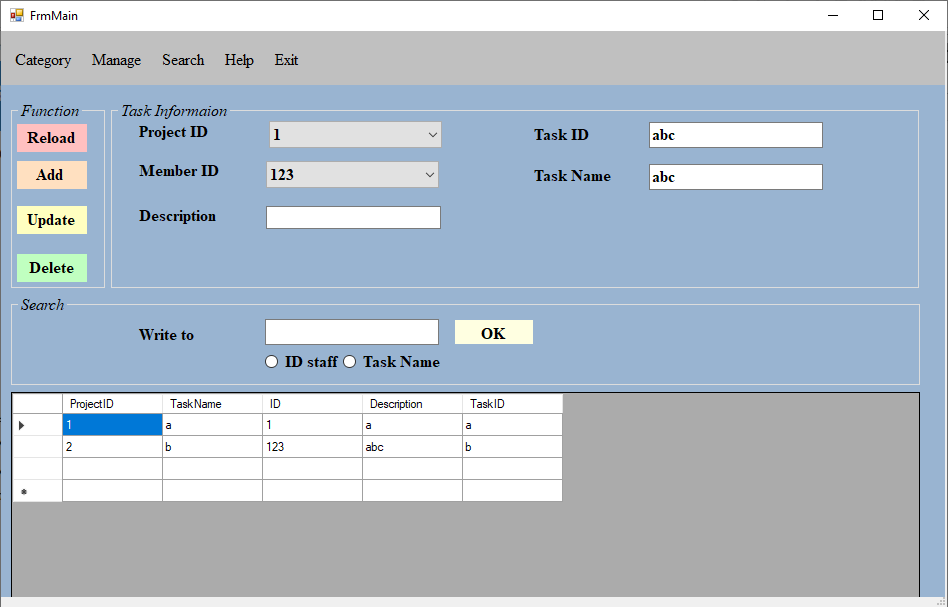
12. Reload project



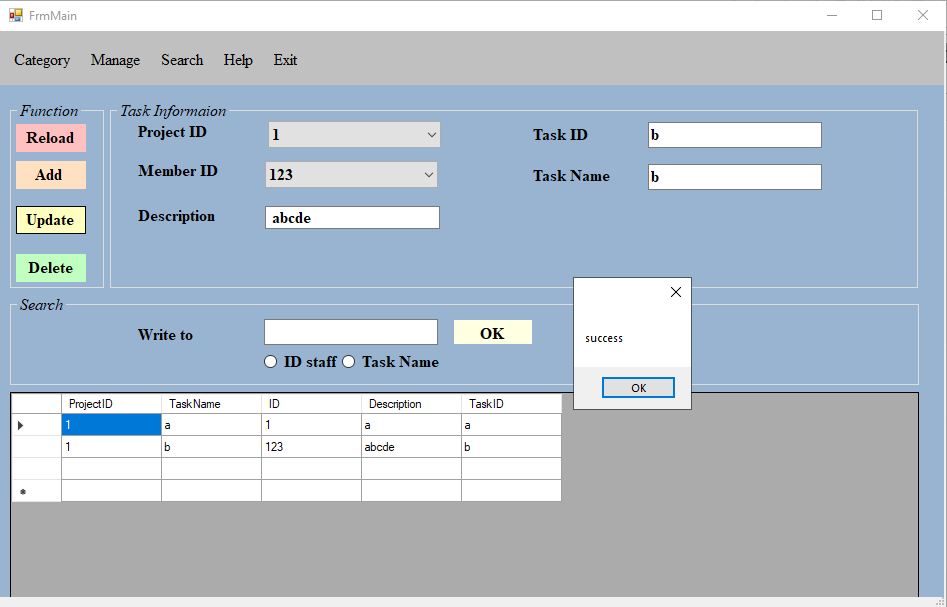
13. Add task



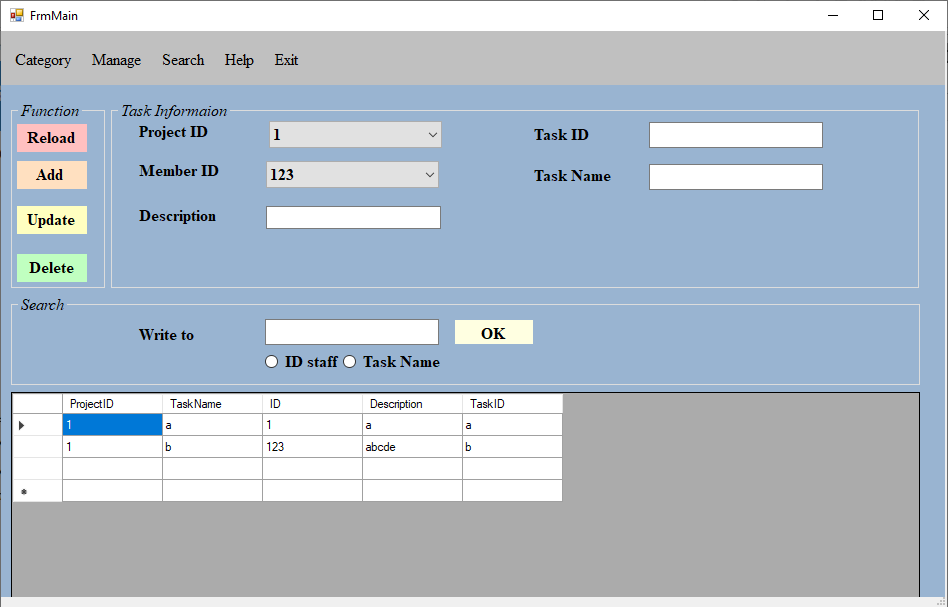
14. Delete task



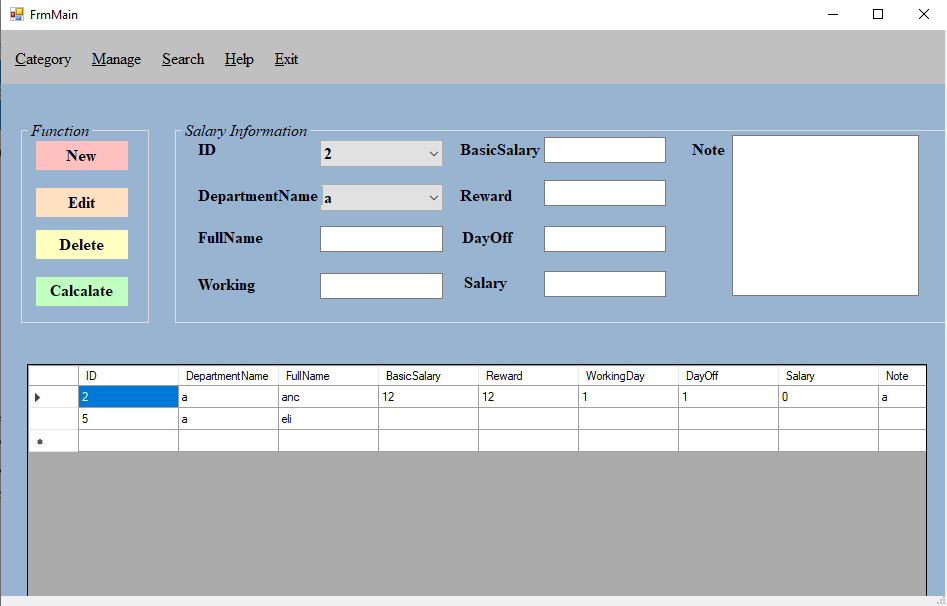
15. Update task



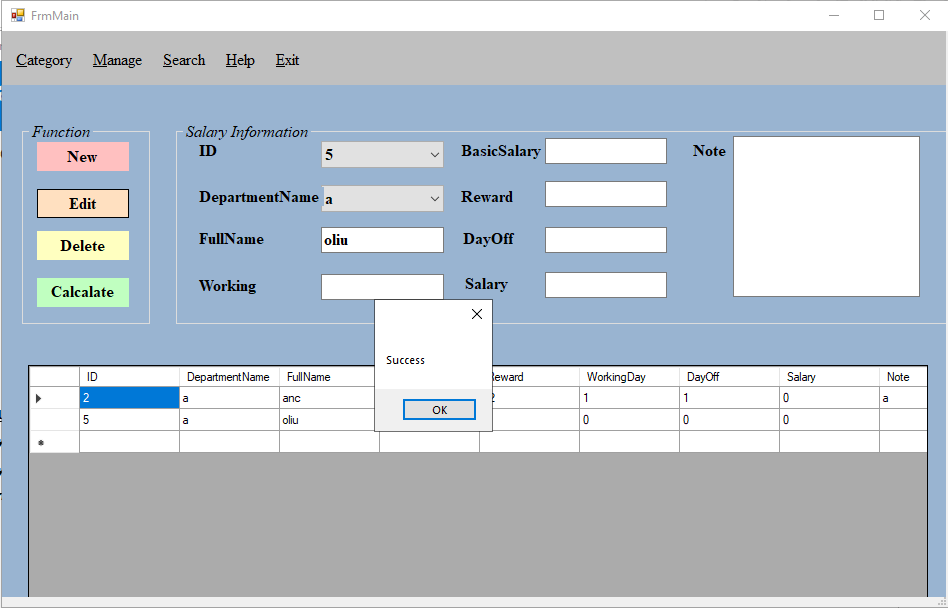
16. Reload task



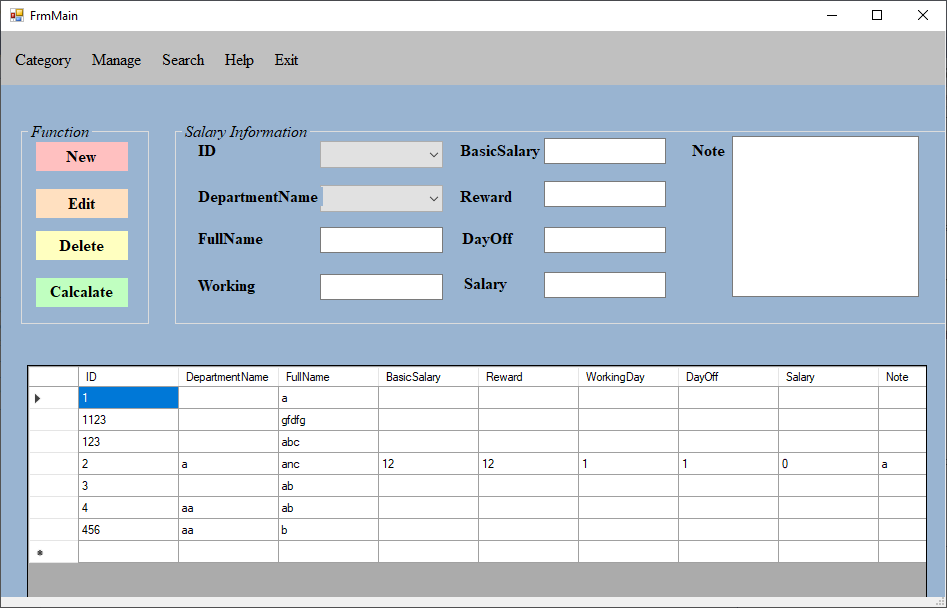
17. New salary



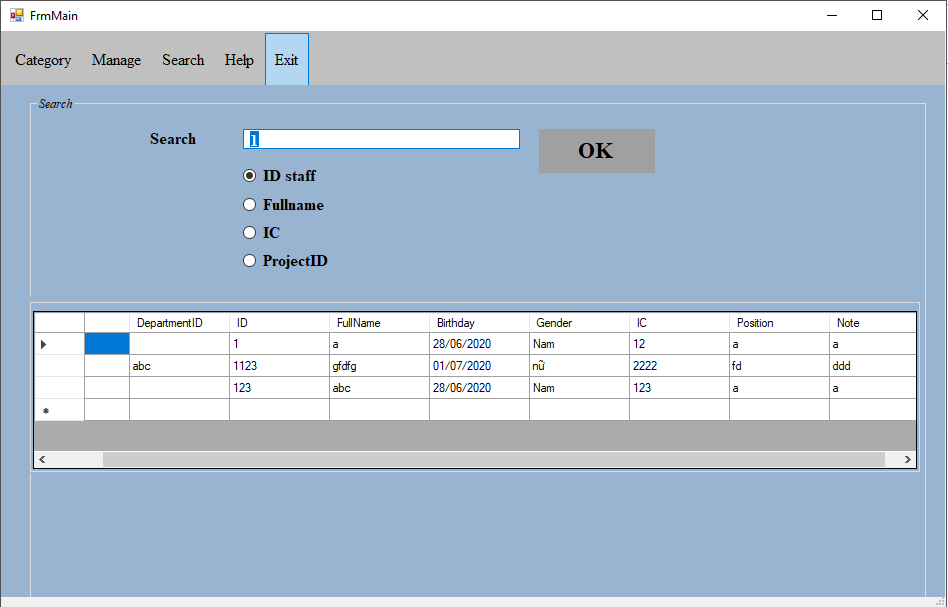
18. Edit salary



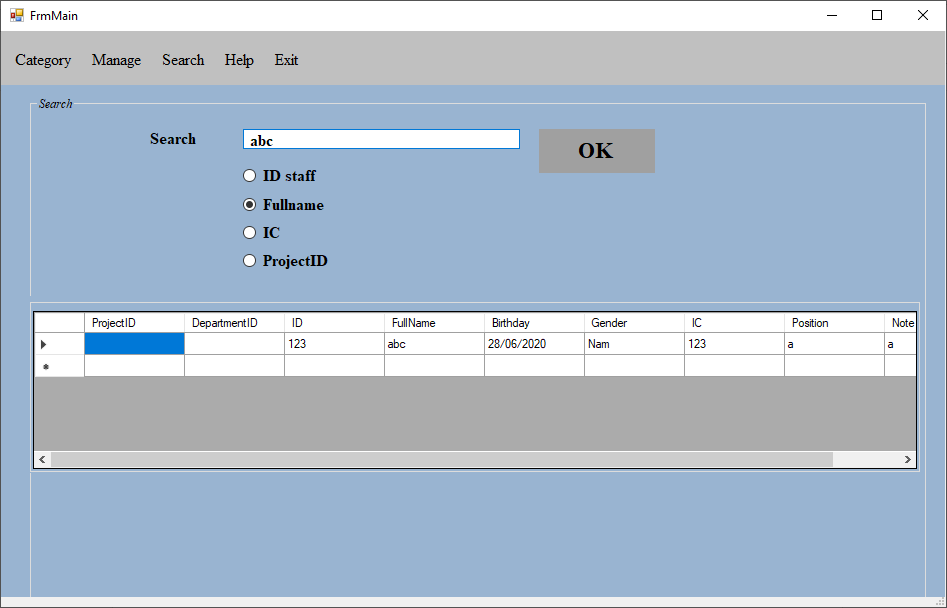
19. Delete salary



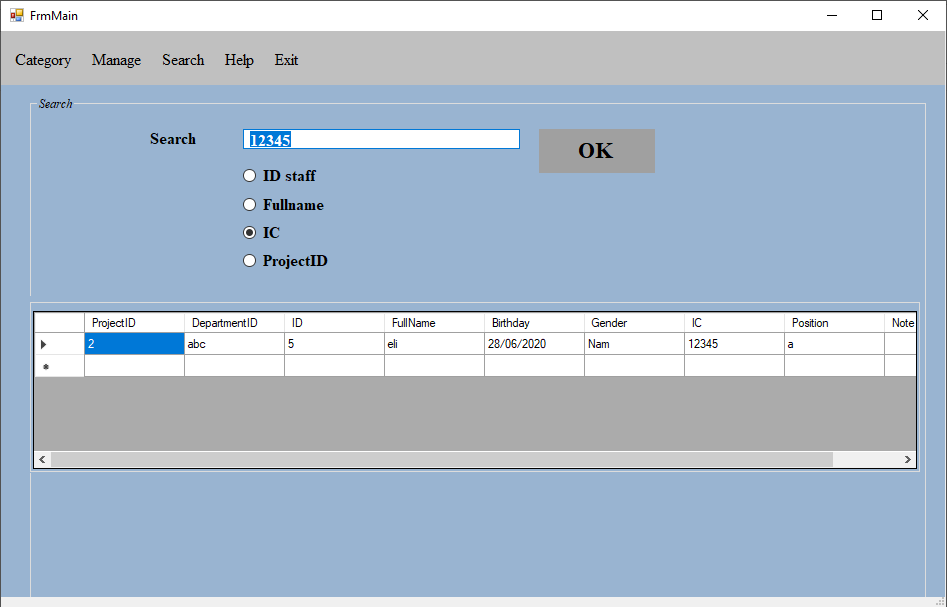
20. Search by ID staff



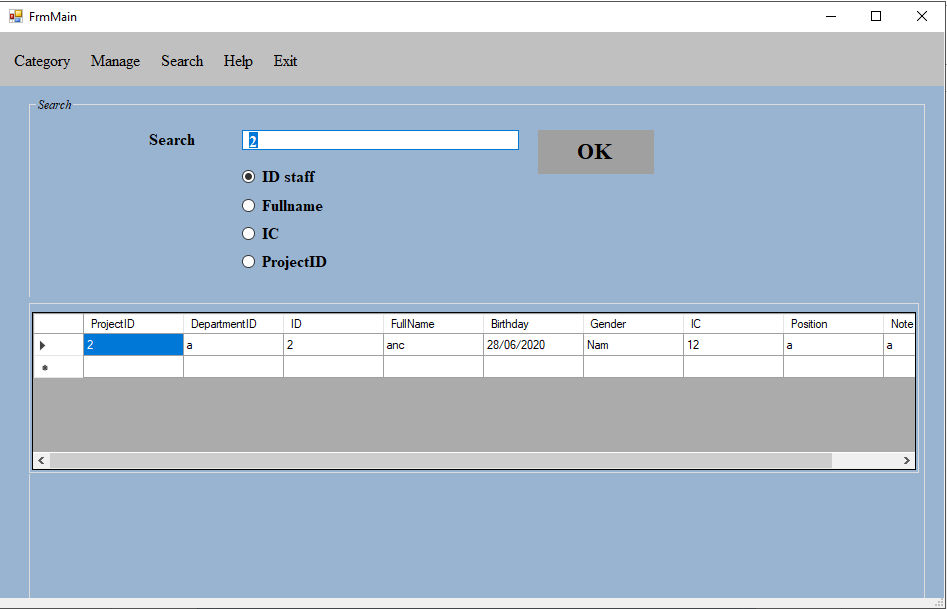
21. Search by Fullname



22. Search by IC



23. Search by ProjectID



V. Conclusion

1. Rate of completion

My team self-assesses that 90% of the project has been completed.

2. Problem

- There are some errors about database such as the relationship between entities.

- The program has not built workfow function to track the completion of the project.

3. Advantage and disadvantage

### a. Advantage

- Easy to use program.

- The program runs fine.

- Accurate results.

### b. Disadvantage

The interface is pretty simple.

4. Development direction

- Improve user interface.

- Complete some functions that has not completed such as workflow function.