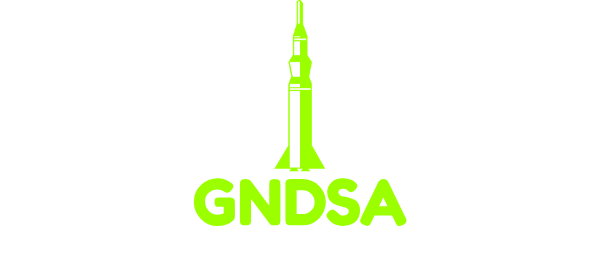
GURU NANAK DEV SPACE AGENCY

**PROJECT REPORT GNDSA DATABASE**

**DILAWAR SINGH (100350551)**

**TAVLEEN JIND ()**

**PAWAN DHALIWAL ()**

**SIMRANDEEP SINGH (100354812)**

**Overview**

As employees are the biggest assets to the company, ensuring they are having maximum time is crucial to any business ‘s success. One way of doing that is giving them easy access to resources they need to work. Creating an application that can save employees time as they can access their accounts.

GNDSA Database is a standalone Database management Software for space agency GNDSA. It gives employees, access to information according to their roles. It provides workforce and management, tools right in their pocket. Permissions for employees to access any of self-service features are according to their roles in agency. The users are assigned with unique IDs and password by HR, to get logged into the system.

Different features provided to employees according to their roles are:

**HR Features:**

* HR is given interface to add new employees to the database of the agency and provide them with unique password. Whereas, unique IDs are automatically assigned by the system as per their hiring date.
* HR have the permission to terminate employees, alongside deleting their record in the database.
* HR also has access to edit the employee information.

**Aerospace Manager Features:**

* Aerospace Manager can create the team with employees and has access to delete it.
* Aerospace Manager has permission to create the Mission and can also delete it.
* As Aerospace Manager has permission to create Mission, he or she can also assign team for any specific Mission.
* Launch Vehicles to the Mission are assigned by Aerospace Manager.

**Aerospace Engineer Features:**

* Aerospace engineers have access to their profiles but in read-only mode.
* Aerospace engineer can message to their team members regarding their missions.
* Aerospace engineer can view status of Missions they are assigned.
* Aerospace engineers have one inbox where they get messages regarding their activities in the agency.

Entity Types

1. Employees- The main role of this entity is store the personal details of employees of this organization
2. Aerospace Team- This entity holds the number of employees that are assigned to Missions.
3. Mission- This entity has main role of storing information such as name, category and subcategory about all the missions of this organization.
4. Launch Vehicle- This entity holds different types of vehicles that are assigned to missions.

Attributes

Employees: employee\_ID, employee\_FName, employee\_LName, department, password, DOB, position, isAssigned

Aerospace Team: Team\_ID, member1, member2, member3, member4, memberID

Mission: mission\_ID, mission\_Name, mission\_Category, mission\_SubCategory, mission\_Type, mission\_Planet, mission\_Status

Launch Vehicle: vehicle\_ID, vehicle\_Name, vehicle\_Mass, vehicle\_Stage

Primary Key

Employees: Team\_ID

Aerospace Team: mission\_ID

Mission: mission\_ID

Launch Vehicle: vehicle\_ID

Foreign Key

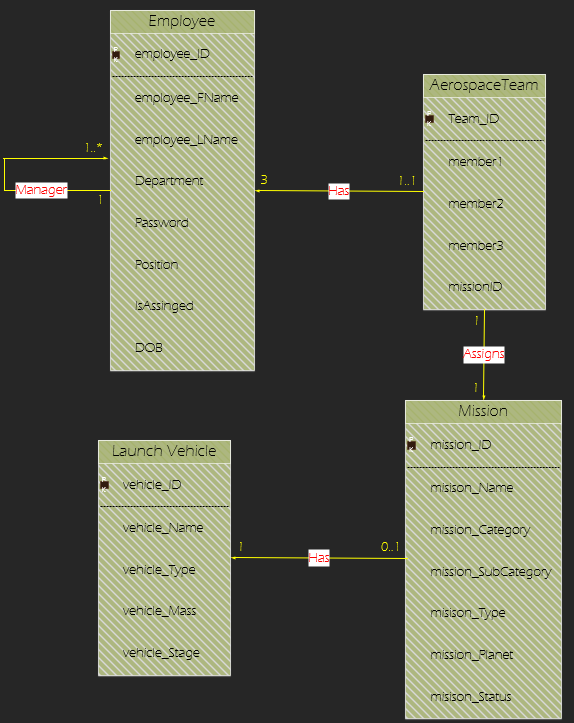
Employee: N/A

Aerospace Team: mission\_ID

Mission: N/A

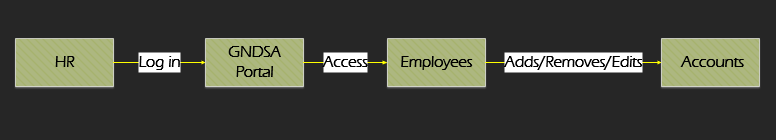
Launch Vehicle: N/A

ER Diagram

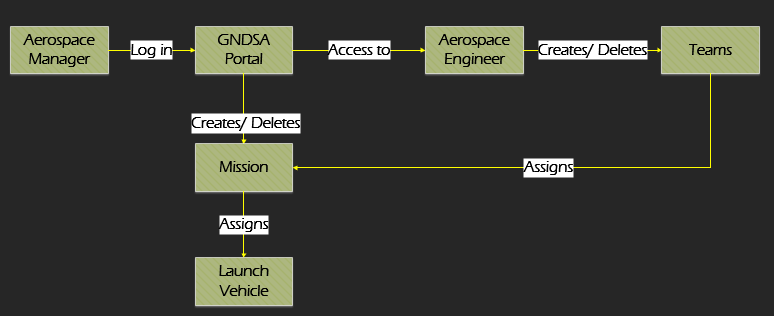


Data Flow Diagrams

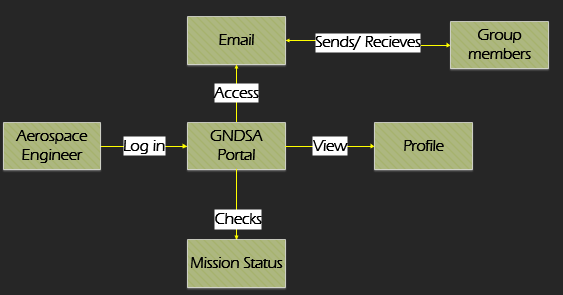
HR schema



Aerospace Schema



Aerospace Engineer Schema



Primary Tables

**Employee (employee\_ID {PK}, employee\_FName, employee\_Lname, department, password, DOB, position, isAssigned)**

**Mission (missionID {PK}, missionName, missionCat, missionSubCat, missionType, missionPlanet, missionStatus)**

**AerospaceTeam (teamID {PK}, member1 {FK}, member2 {FK}, member3 {FK},**

**missionID {FK})**

**t(MEMBER ID) ( SID, SendDate,Subject,Message)**

**vmass (teamID {FK}, VehicleID {FK})**

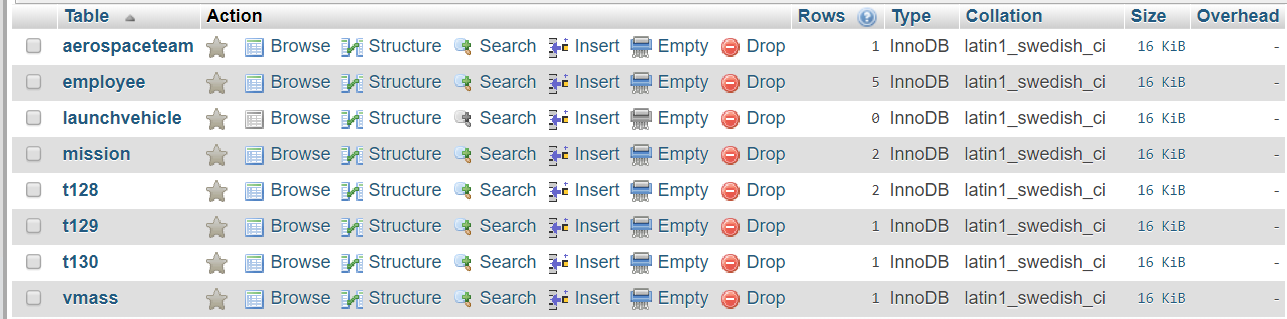


Figure 1: Tables used

DESIGNING APROACH

This database is designed primarily in Java language, specifically using JavaFX and uses JDBC drivers and XAMPP Server Control panel to bridge front end and backend SQL database. The GUI elements are designed using Scene-Builder open-source software integrated with eclipse IDE. The programme uses different SQL queries to perform functions expected from fully functional programme. The UI is configured using CSS.

Software

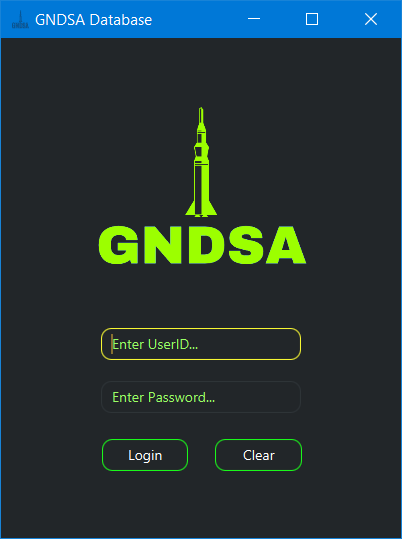
Since database contains confidential information, it is mandatory to prevent un-authorised use of database. Therefore, it is Password protected. The User identification system works on the principle of filtering where username of user trying to log in is compared and based on his department he is presented with his dashboard. For example, HR manager will automatically be directed to HR Dashboard where as Aerospace manager will be directed to Aerospace Portal Dashboard.

Figure 2:Login Screen

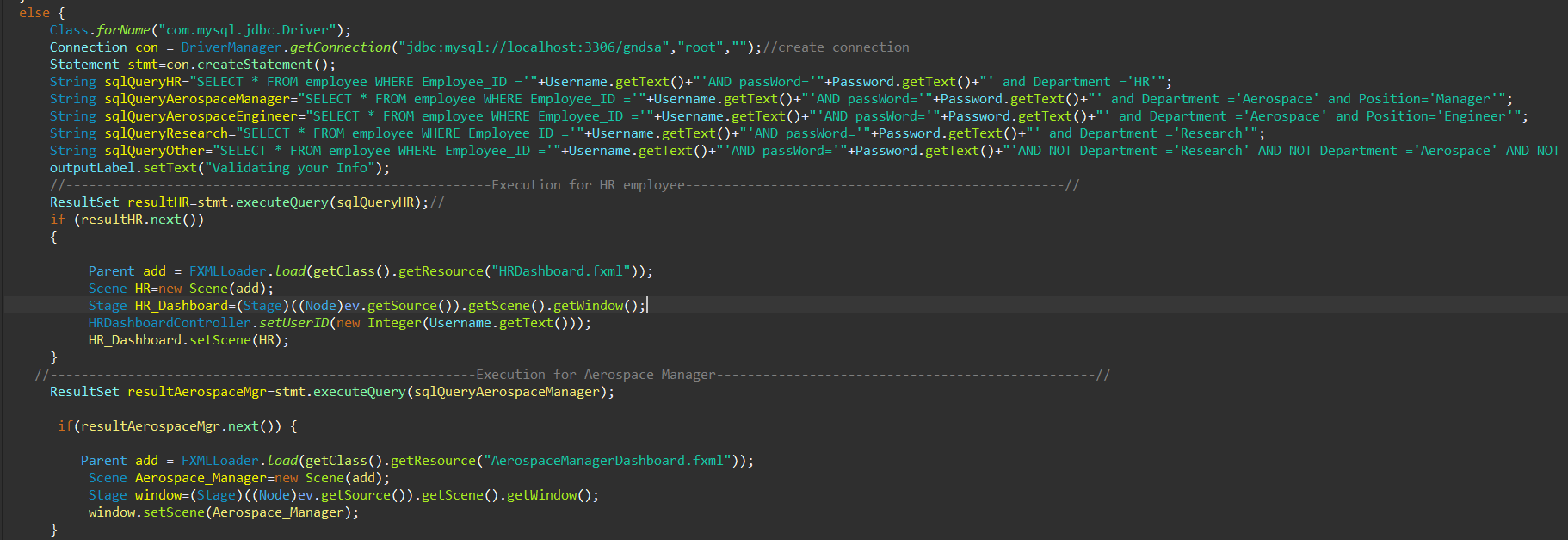


Figure 3:SQL user identification query in java

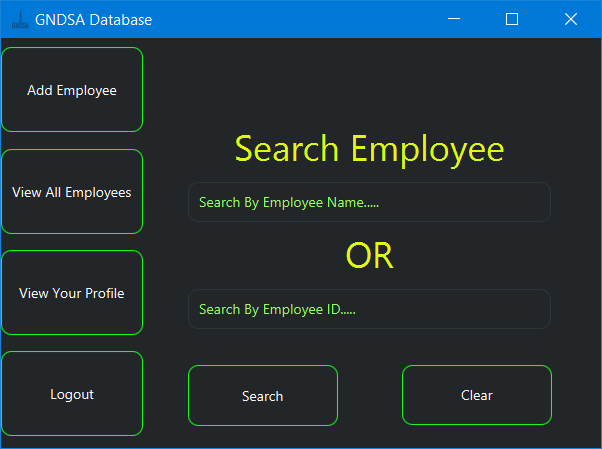


Figure 4:HR Manager Dashboard

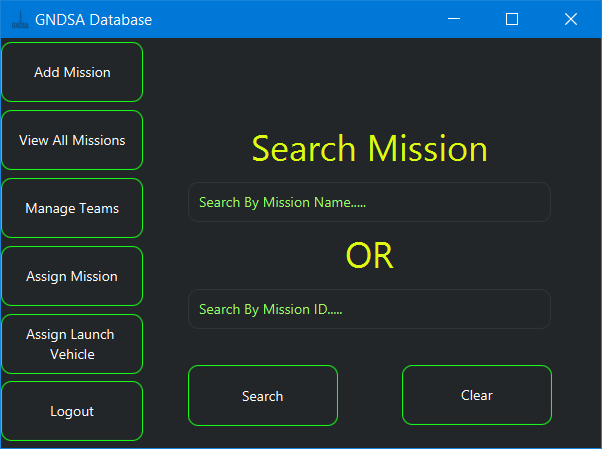


Figure 5: Aerospace Manager Dashboard

In their Dashboard, they can perform all the functions which a person at their position has to do. For example, HR Manager can hire, or fire people and Aerospace manager can Initiate new missions and manage teams of aerospace engineers.

HIRING OF EMPLOYEE (Inserting Data) AND SQL

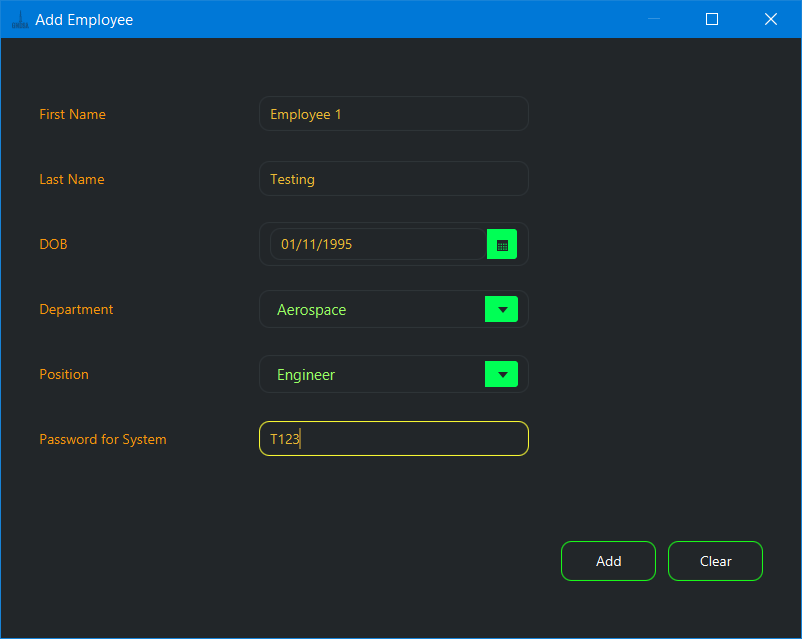
Hiring process uses “INSERT” command to add details of the person in database.

Figure 6: Adding Details of new Employee



Figure 7: SQL INSERT query used in java code

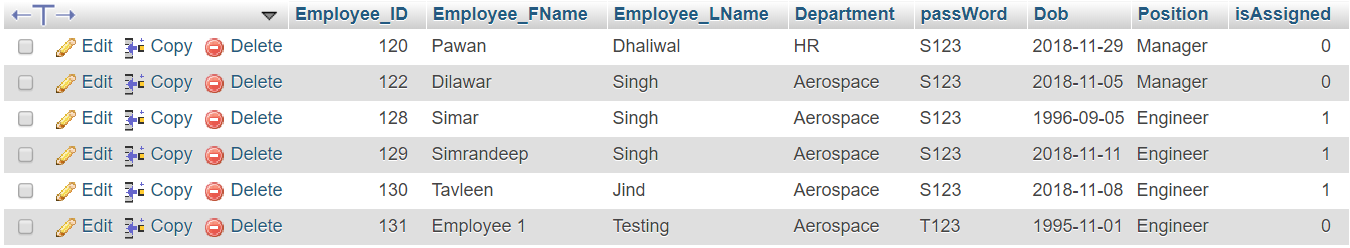


Figure 8: Data Insertion in SQL Table

DELETING AN EMPLOYEE(Deleting Data) AND SQL

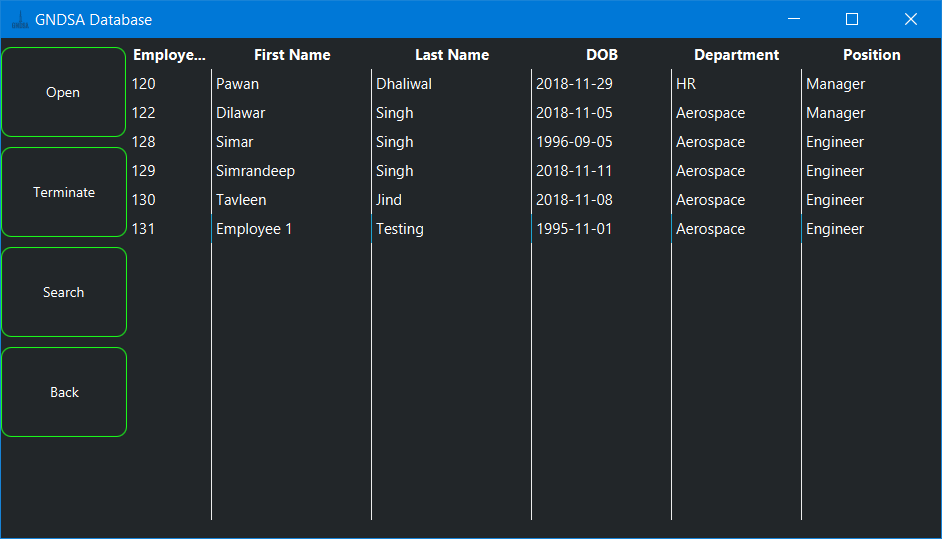
Deleting is one of the most important aspects of a database since unwanted information should be removed from database. Deletion uses “DELETE” command to delete an employee and his records.

Figure 9: Termination (Deletion) of Employee

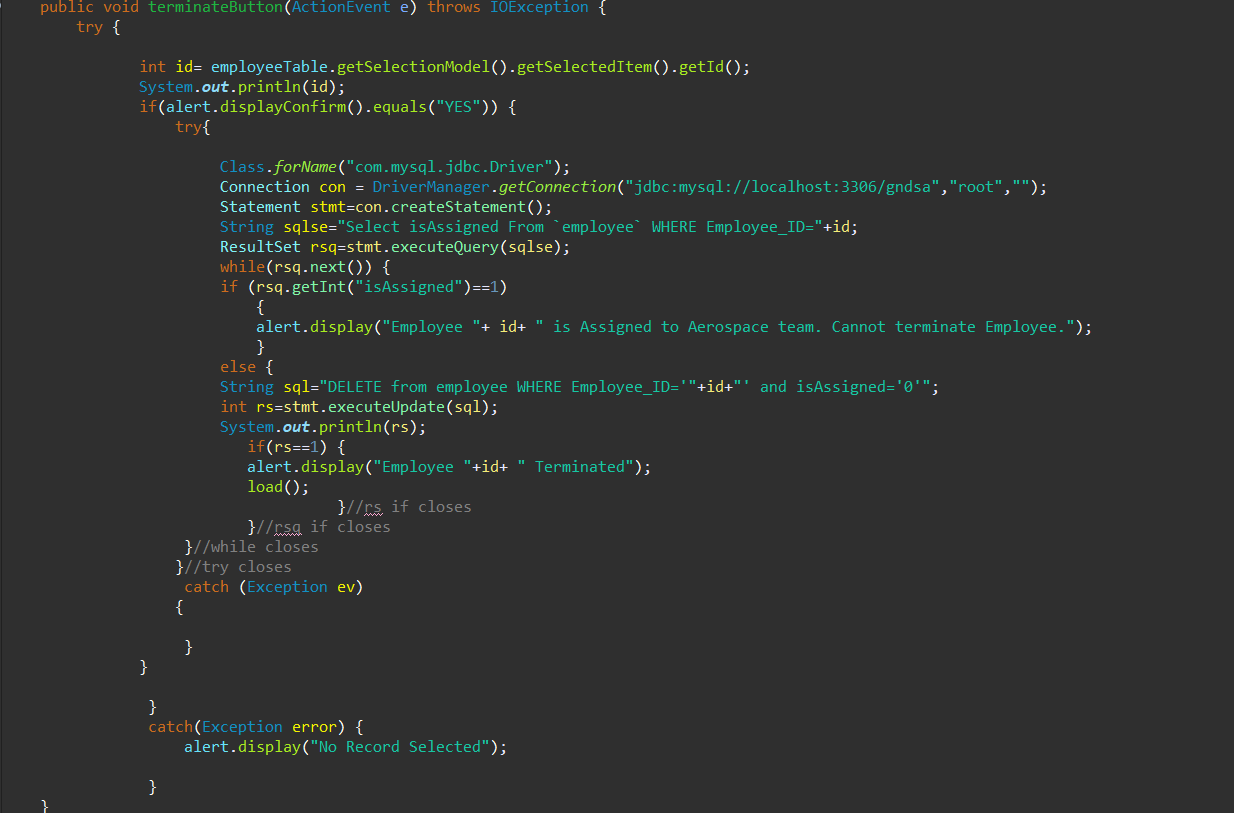


Figure 10: Terminate method in Java which uses DELETE query

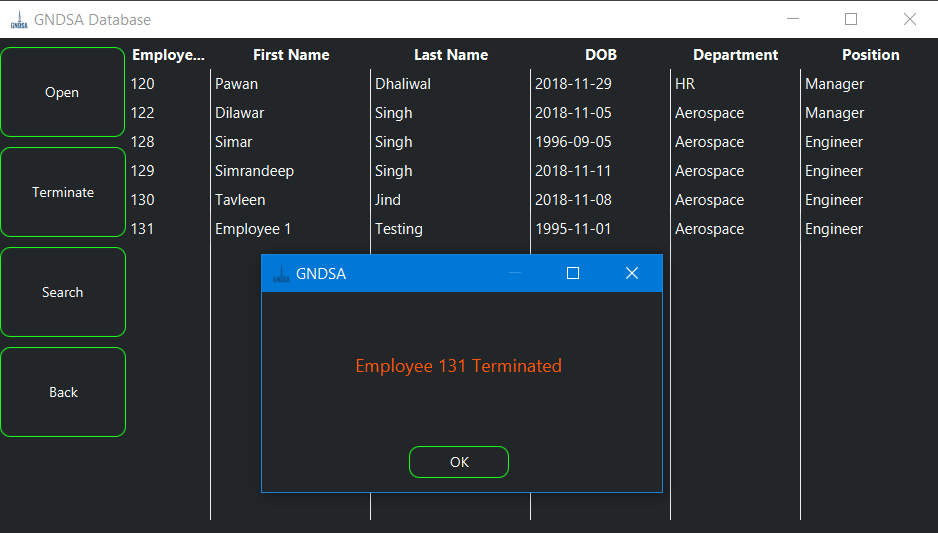
To terminate an employee, database first checks if an employee whom HR is trying to terminate is assigned to any team. If the Employee is assigned to team then hr cannot terminate an Employee. After checking, code deletes the record from database.

Figure 11: Employee terminated

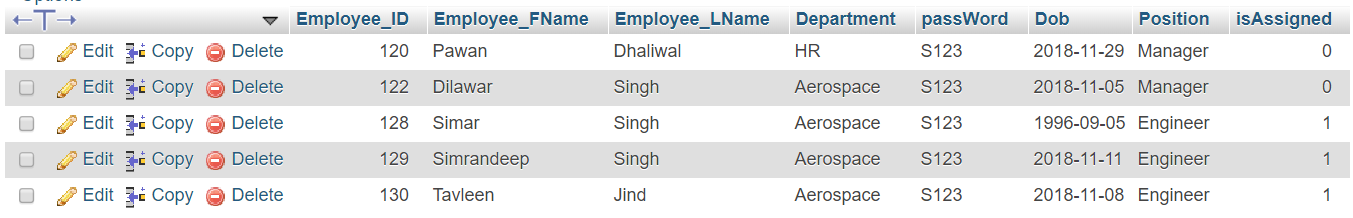


Figure 12: Terminated Employee 131 is not in Table Anymore

SEARCHING AND SQL

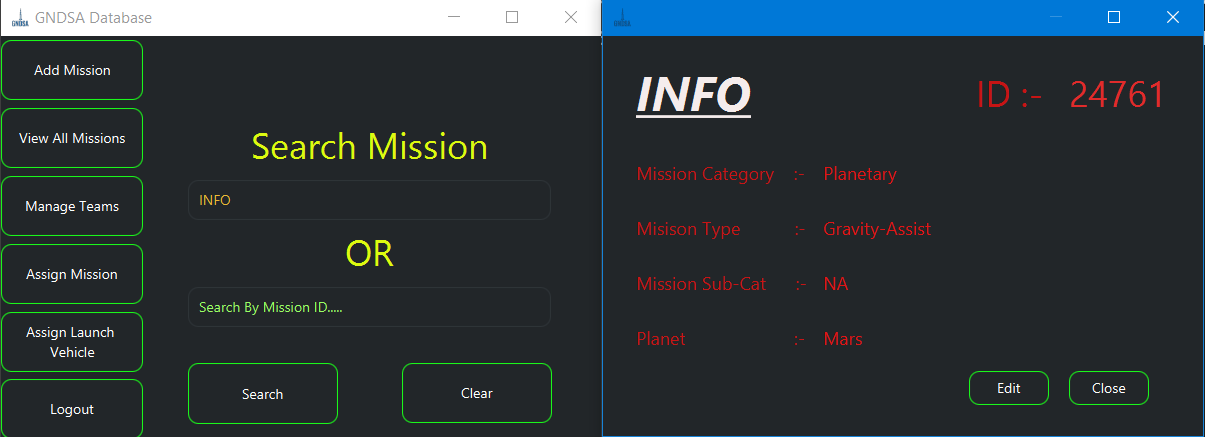
Ability to search for specific details is another important requirement for database management software. GNDSA Database is no less. HR and Aerospace Dashboard provide managers with direct ability to search without going into additional menus. Concept Is same in both the cases. HR can search for employee with Employee’s name or employee ID and Aerospace Manager can search mission with mission name or mission ID.

Figure 13: Searching in Aerospace Manager Dashboard

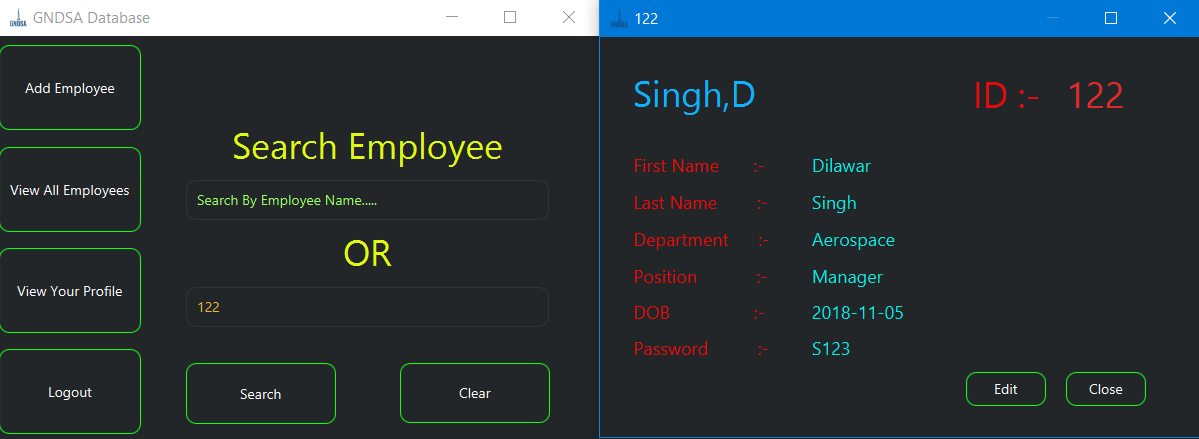


Figure 14: Searching in HR Manager Dashboard

Searching uses “SELECT” command to perform the required functions. It shows proper error message if no result is found.

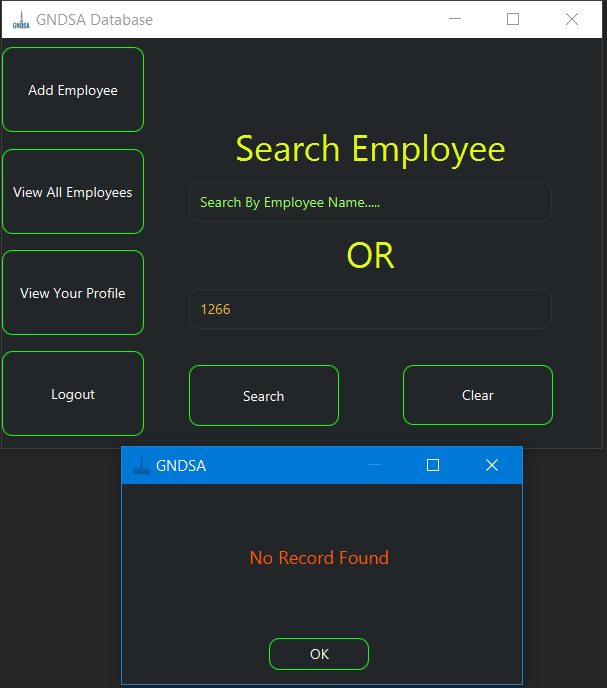


Figure 15: Searching with no record in database

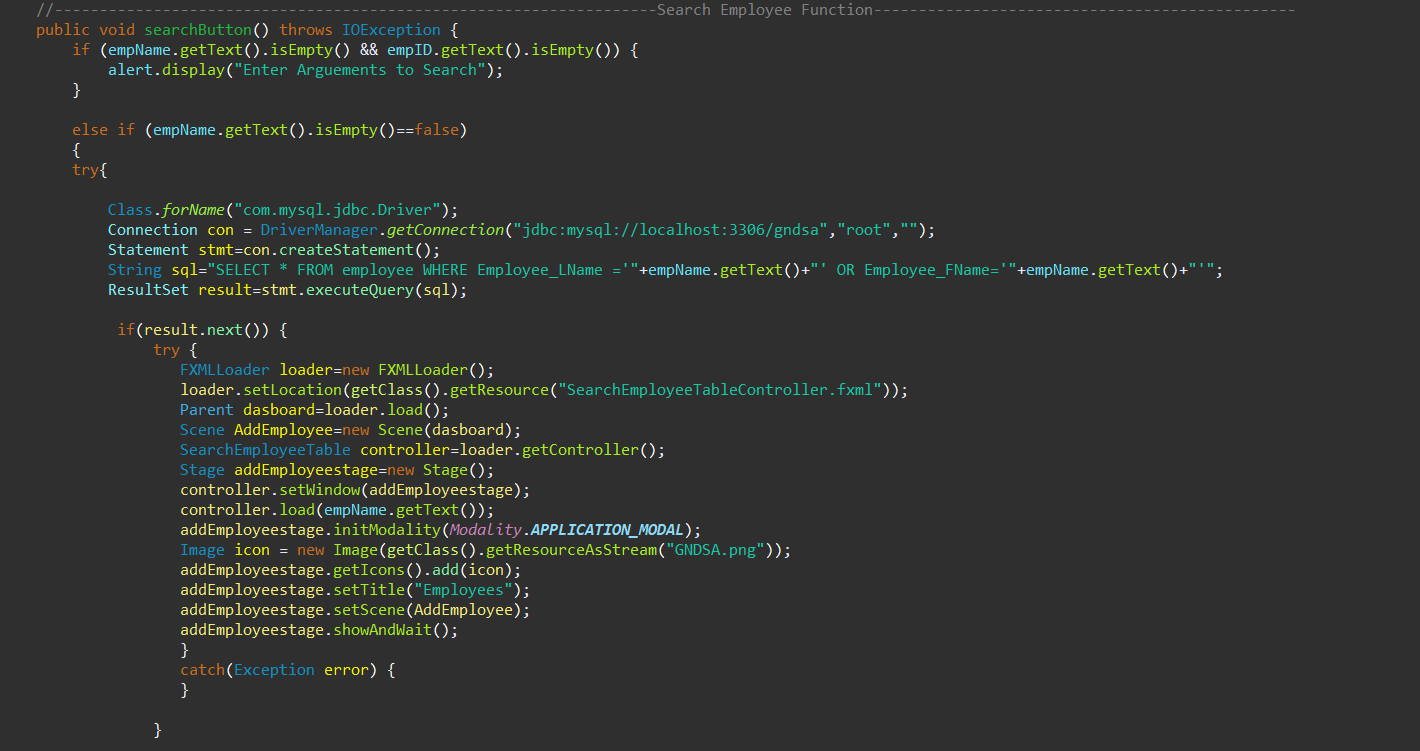


Figure 16: Use of SELECT command to search

EDITING INFORMATION AND SQL

No database management software is complete without giving user the ability to update the information which added previously. GNDSA users can edit the information without much hassle.

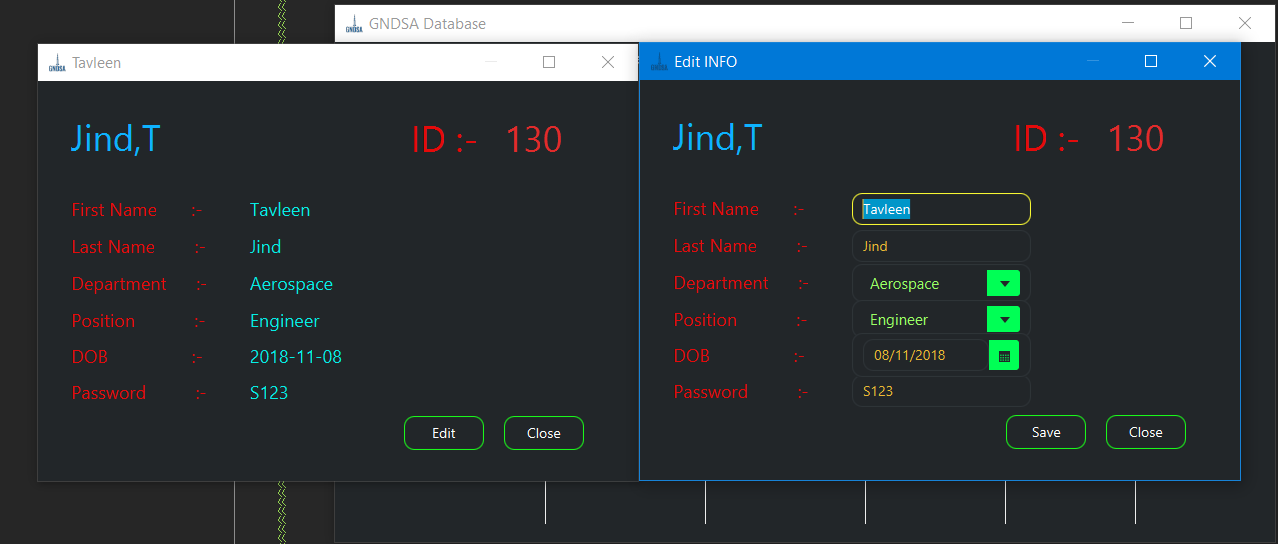


Figure 17: Editing info of employee

User can Simply make changes and hit save. The record will be updated in SQL table.



Figure 18: Updated information

Editing the information uses “UPDATE” and “SET” Commands.

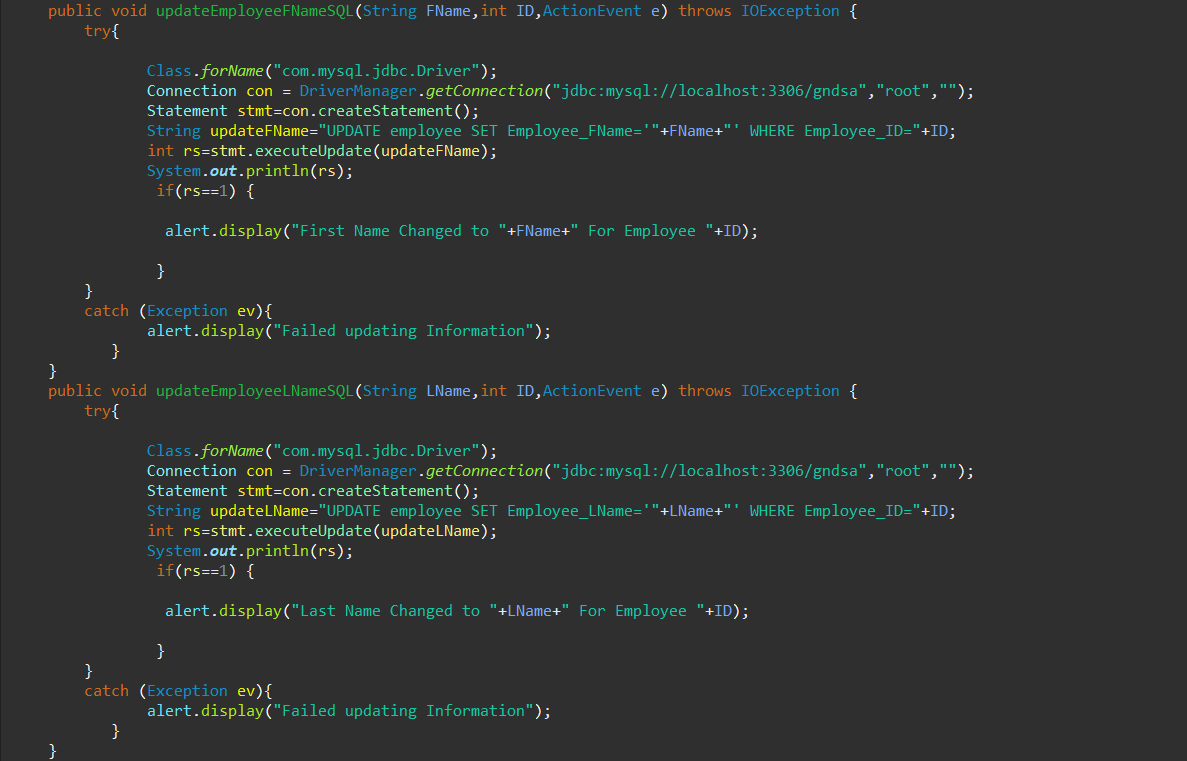


Figure 19: Updating Information. Use of Set and update Commands in java

SENDING EMAILS

One of the features of GNDSA Database is the ability of Aerospace team members to communicate among teammates. Whenever any activity takes place, like engineers are assigned to teams or teams are assigned to missions, team members get automatic system generated emails to inform them about the activities. Below are the screenshots of the full process.

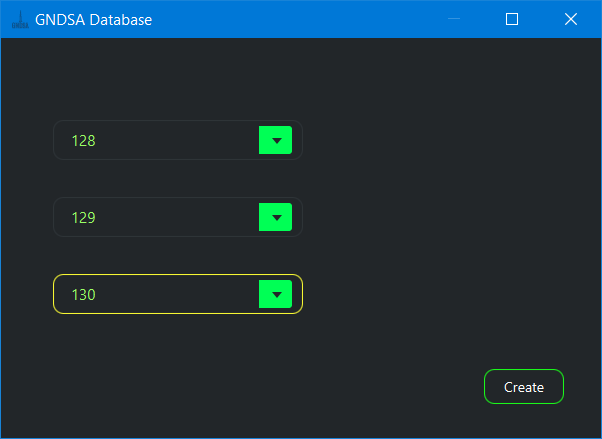


Figure 20: Creating team

When Aerospace manager Creates the Team, in SQL, the new table is made with name t(team-member ID). This created table acts as the mailbox which keeps the information related to message (Sender’s ID, Date of Message, Subject and actual message).

t(MEMBER ID) ( SID, SendDate,Subject,Message)

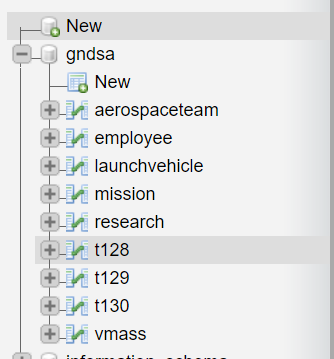


Figure 21: Employee mailbox table

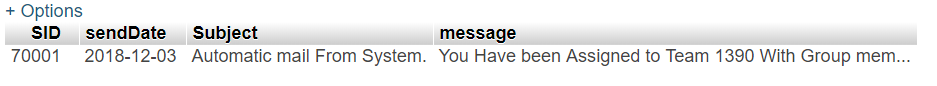


Figure 22: Storing message in SQL table

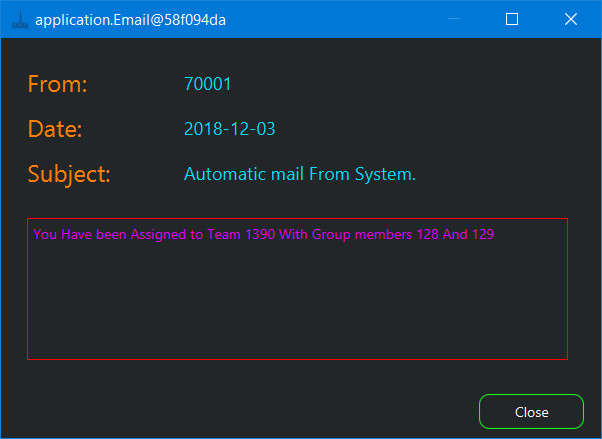


Figure 23: Message Inbox view

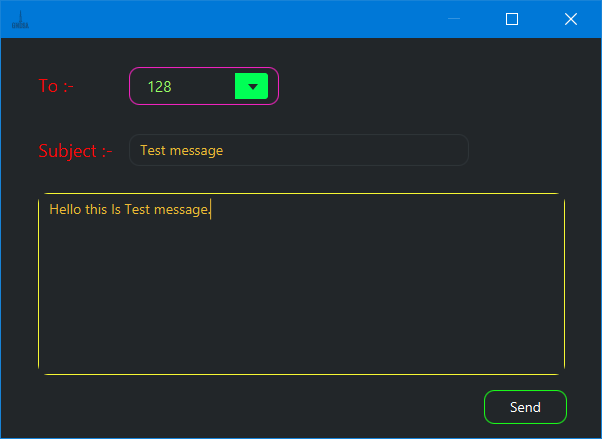


Figure 24: New Message

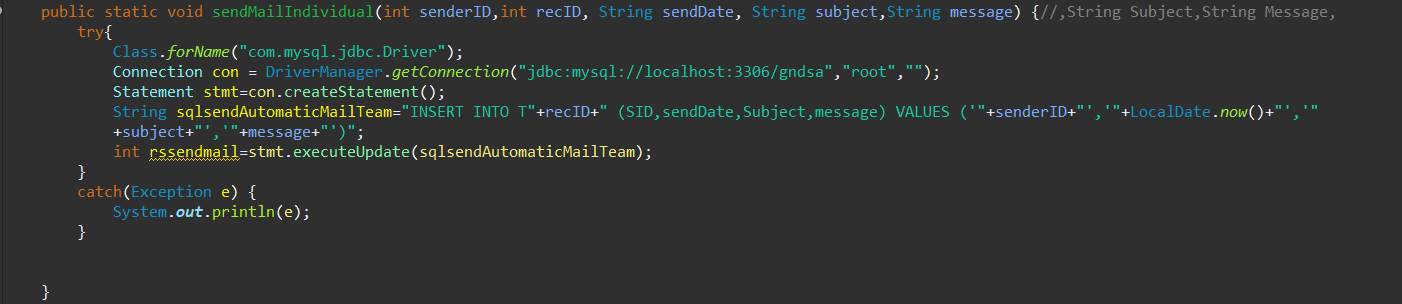


Figure 25: java function uses "INSERT" command to send message

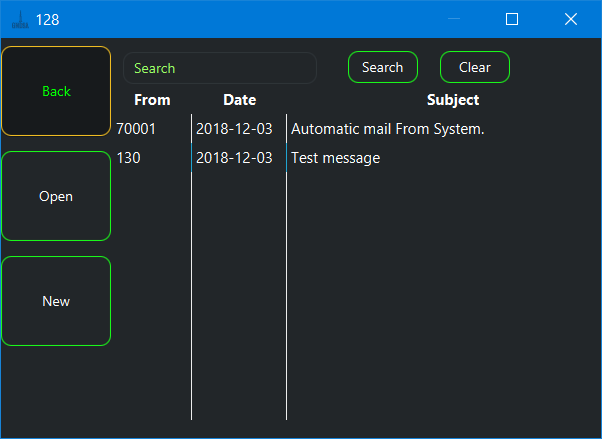


Figure 26: Inbox of Receiver

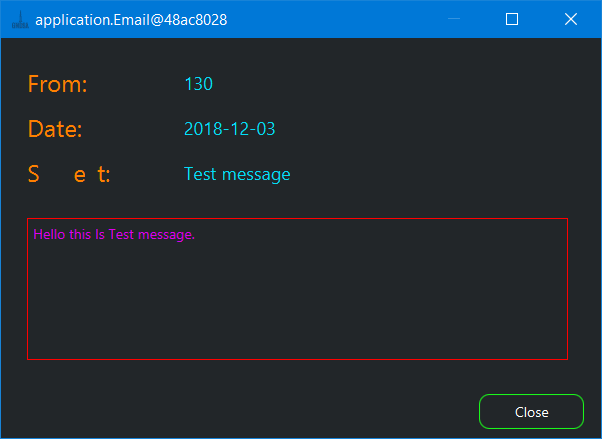


Figure 27: Message View in Receiver Inbox

To read and Show the Message, “SELECT” command is used.



Figure 28: Code for reading inbox in java (id is user-ID who is currently logged in).

Advance Searching and joining of Tables

Our Database Software tries to use the full functionality of SQL. Table Joining is one of the advanced concepts which helped our project provide proper User experience. One Part where table joining is used is Aerospace manager getting names of team members from Employee table using team-member ID as Foreign key.

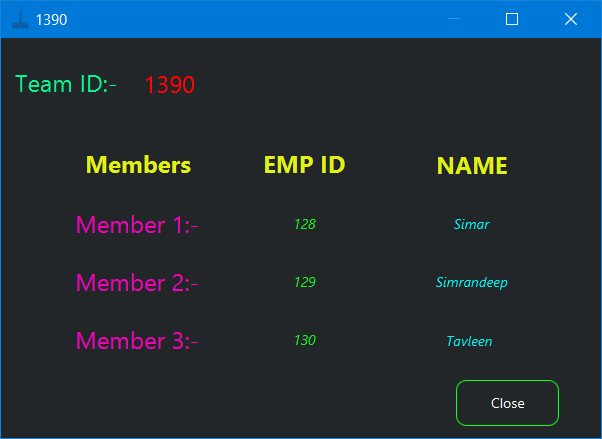


Figure 29: Team Details In Aerospace Manager Dashboard

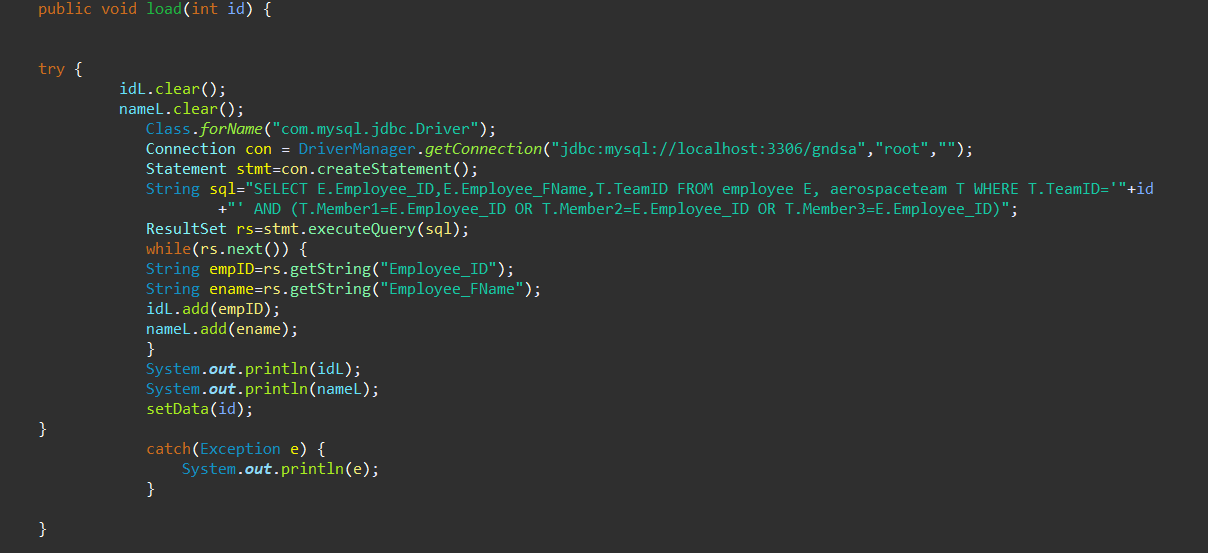


Figure 30: SQL table joining query

Conclusion

Our team would like to thank Dr. Mandeep Pannu to provide us with an opportunity to learn and implement real life things which will help us in nurturing our career in computer IT field. We learned many new things especially team working skills. We, as a team, enjoyed a lot while working on this project.