THE ASCY AIRPLANE DATABASE

Table of Contents

Acknowledgements

Firstly, we would like to express my gratitude to our parents for making available the materials and the knowledge necessary for the completion of this project.

We would also like to thank our Python teacher, Ms. Anuja Mittal, for equipping us with the right tools and mindset to bring this project to life.

In addition, we thank the school and its administration, including but not limited to our Principal Ms. Richa Agnihotri and Senior School Incharge Ms. Poorni Rajesh as well as the Central Board of Secondary Education for giving us the opportunity to write my own research paper.

Introduction

ASCY stands for an application created by **A**dvay, **C**hirag and **S**rijan which would go on to store **Y**ottabytes of data.
ASCY is a database system tailored for the usage of a commercial airline company. It boasts multiple tables with ease of access, with a 2-factor secured admin account that can assign security levels to the viewing and editing of different tables. The application can also automatically update a lot of data values if connected to its respective source, for example, an input to the projected fuel costs will help it to automatically generate ticket prices for all the flights on that particular day, depending on the number of stops, distance travelled, payload, profit margin etc. Similarly an input on the weather conditions is helpful to predict the Estimated Time of Arrival.
The software was built with a small to medium scale company in mind, but has been coded in a way so as to not hinder scalability. A new table according to company demands can easily be added, assigned a security level and integrated as if it were there from the very beginning.

The program will help reduce man hours spent on manually entering data values, projecting prices and other menial tasks, drastically helping the company to reduce expenses as well as focus its monetary aid towards the safety and advancement of air travel.

Applications

Each and every successful company over the last years, strives to achieve a level of excellence by being extremely efficient across all fields of work. Our program uses the simple applications of a Database management system and allows seamless and smooth connectivity among company employees, executives and infrastructure. In order to make things even simpler, we use python to connect a DBMS to the user interface that is very easy to understand.

The database management system (DBMS) handles the data; the database engine allows data to be viewed, locked, and updated; and the database schema specifies the logical structure of the database. These three fundamental features contribute to concurrency, security, data integrity, and consistent data administration methods. Many common database administration functions are supported by the DBMS, including change management, performance monitoring and tuning, security, and backup and recovery. Most database management systems are also in charge of automatic rollbacks and restarts, as well as recording and auditing of database and application activities.

The DBMS provides a centralised view of data that may be accessed in a controlled way by numerous users from different places.A database management system (DBMS) can limit what data end users see and how they see the data by offering many views of a single database schema. Because the DBMS handles all requests, end users and software programmes are not required to understand where the data is physically housed or on what sort of storage media it resides.

The DBMS can provide both logical and physical data independence, shielding users and applications from needing to know where data is kept or being worried about changes to data's physical form. Developers will not have to adjust programmes simply because the database has changed if applications use the application programming interface (API) provided by the DBMS.

Python has bindings for many database systems including MySQL, Postregsql, Oracle, Microsoft SQL Server and Maria DB.

Some other examples of DBMS with Python connectivity are:

- Railway stations services
- Human Resource services in any company
- Social Media experiences
- Credit Card management (banks)
- E-Commerce
- Manufacturing

Logic

<u>Login</u>

A multi-level login system with the following classifications, based on usage and ease of access. This can be customized to the company's needs as well:

- <u>A-Level Clearance</u>
  This would grant the user access to the entire company database.
- <u>B-Level Clearance</u>
  A clearance tailored to the needs of the ticket **booking staff** and websites, it grants access solely to the ticket prices and availability records.
- <u>C-Level Clearance</u>
  A special clearance for the **crew** (pilots and cabin crew), it would grant access to the Flight and Plane Details.
- <u>E-Level Clearance</u>
  The E-Level Clearance grants access to the Plane Maintenance database for **engineers** working on snags.
- <u>F-Level Clearance</u>
  The user would have access to all **Financial** details of the company. Ideal for employees in the financial department.

<u>Plane Management/Maintenance</u>

A database that would record the planes owned by the company, their specific details, abilities and engineers will approve the plane for take off in the Status column.

ID	Model	YoM	Passenger Capacity	Fuel Consumption per HO	Endurance	Hours flown	Status (Airworthy/Check/ Grounded/Retired)

ID - An identification code unique to each plane to be used for efficient communication (integer)
Model - All general and specific repairs will be done keeping the plane model in mind (string)
YoM - Year of Manufacture helps determine the extent of repair and retirement date (string)
Passenger capacity - Helps to determine the demand-supply ratio (integer)
Fuel Consumption - The average fuel consumption per Hour of Operation (HO) (integer (liters))
Endurance - The maximum distance that the plane is capable of flying on a full fuel tank and at full capacity (integer (nautical miles))

Hours flown - The number of hours it has been in use would help determine the extent of repairs (integer)
Status - Indicates what state the plane is in; it will be filled out by the Chief Engineer
Airworthy means that the plan is ready for its next flight
Grounded means that the plane is currently unavailable for flights. This could be for repairs, tests, inspections, reserves or any other reason
Retired means that the plane is now useless to the company due to outdated equipment, the cost of updating which will be more than buying a new plane

---

Passenger Database

Passenger Code	Flight Number	Economy or First Class	Name	Sex	Phone	Email

Passenger Code - This will be a unique 3 digit code assigned to each individual passenger
Flight Number - Linked to the flight details table, it would help to identify which flight the passenger is on
Class - Economy or First depending on personal preference
Passenger Details (Name, Sex, Phone, Email) - All required details and will be necessary when contacting the person

---

Flight Details + Ticket Costs

Flight ID	Destination	Date	Boarding Time	Boarding Gate	Ticket Cost

A database which will contain all ticket and departure details
Flight ID - An identification code unique to each plane to be used for efficient communication (integer)
Destination - The place to which the flight is flying, the starting point is always XYZ terminal
Date - The date of departure of the flight
Boarding Time- The boarding time, passengers must be seated, one hour before the flight departure
Boarding Gate - The gate at which passengers will board the aircraft
Ticket Cost - The total pricing of a singular ticket on the flight

---

Financials

Flight ID	Date	Taxes	Fuel Cost	No. of Pax	Ticket Cost	Service Charge

Would basically include all details about the company's financials

Menu driven

- **Input taxes**
- **Revenue**
- **Calculate profit/loss**

Taxes to be of two kinds primarily, but to be inputted as sum

- Government tax
- Tax for parking of aircraft

Revenue collected will be solely from the customers:

- Government taxes
- Service Charge

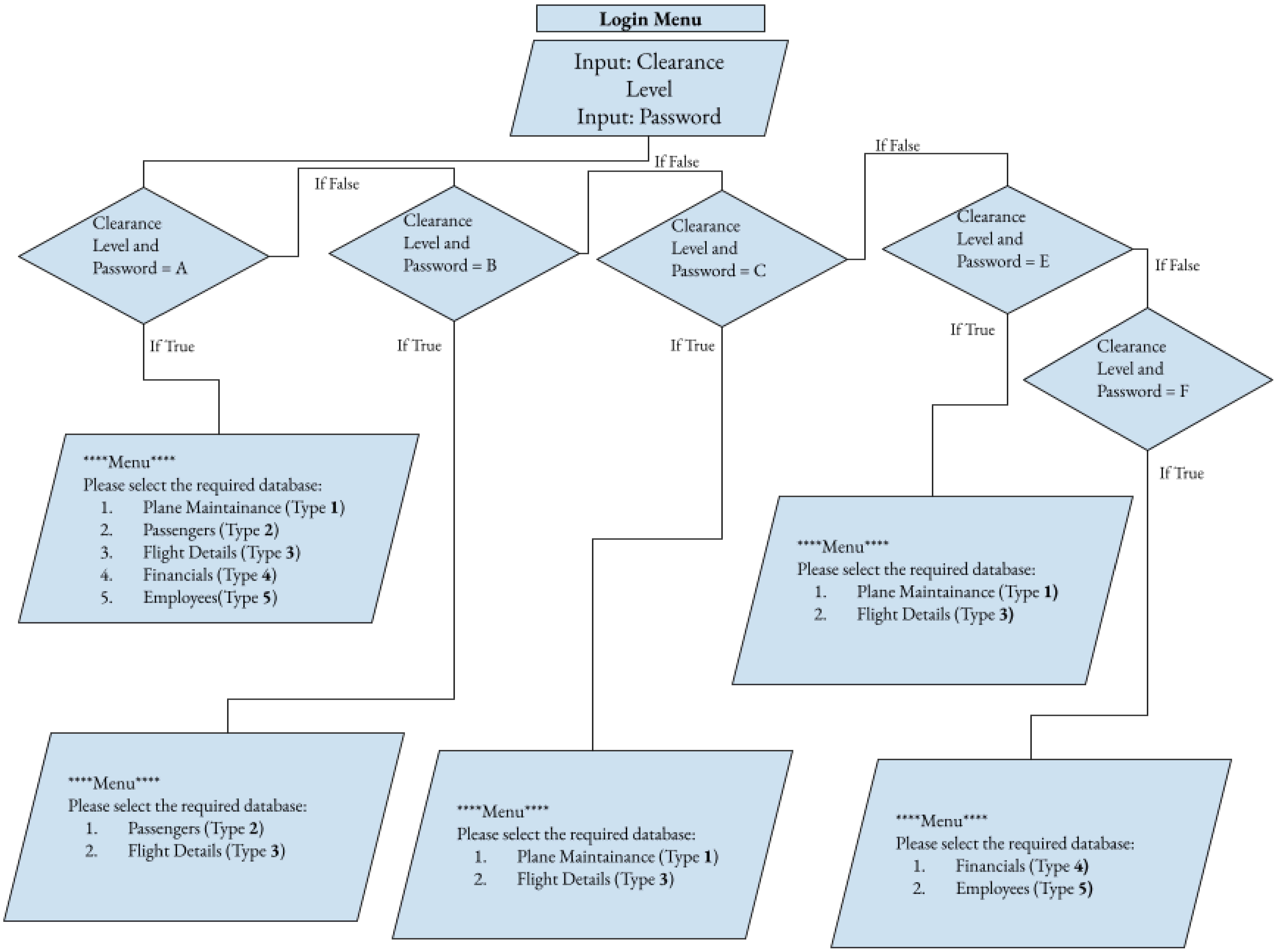
Profit/ loss will be calculated on the basis of revenue collected

Employee Salaries

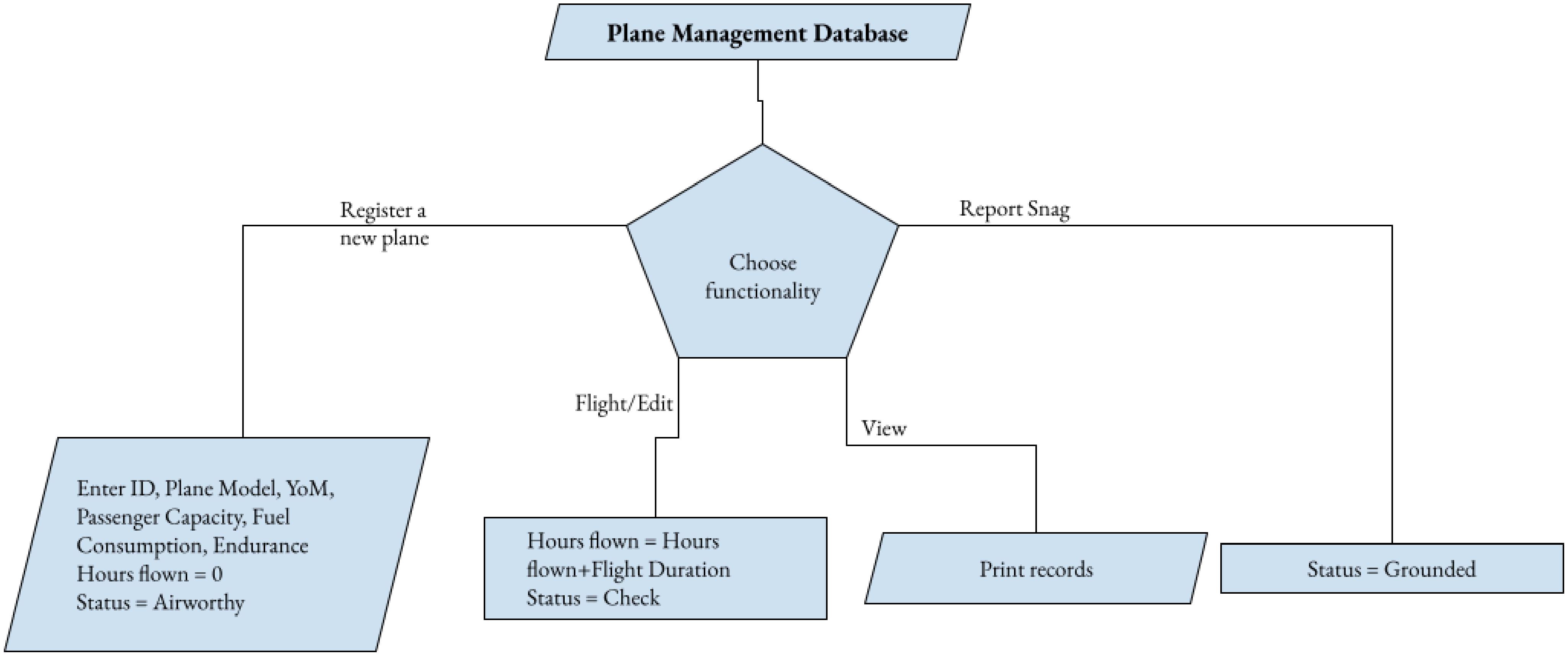
EmpID	Name	Age	Date of Joining	Designation	Salary

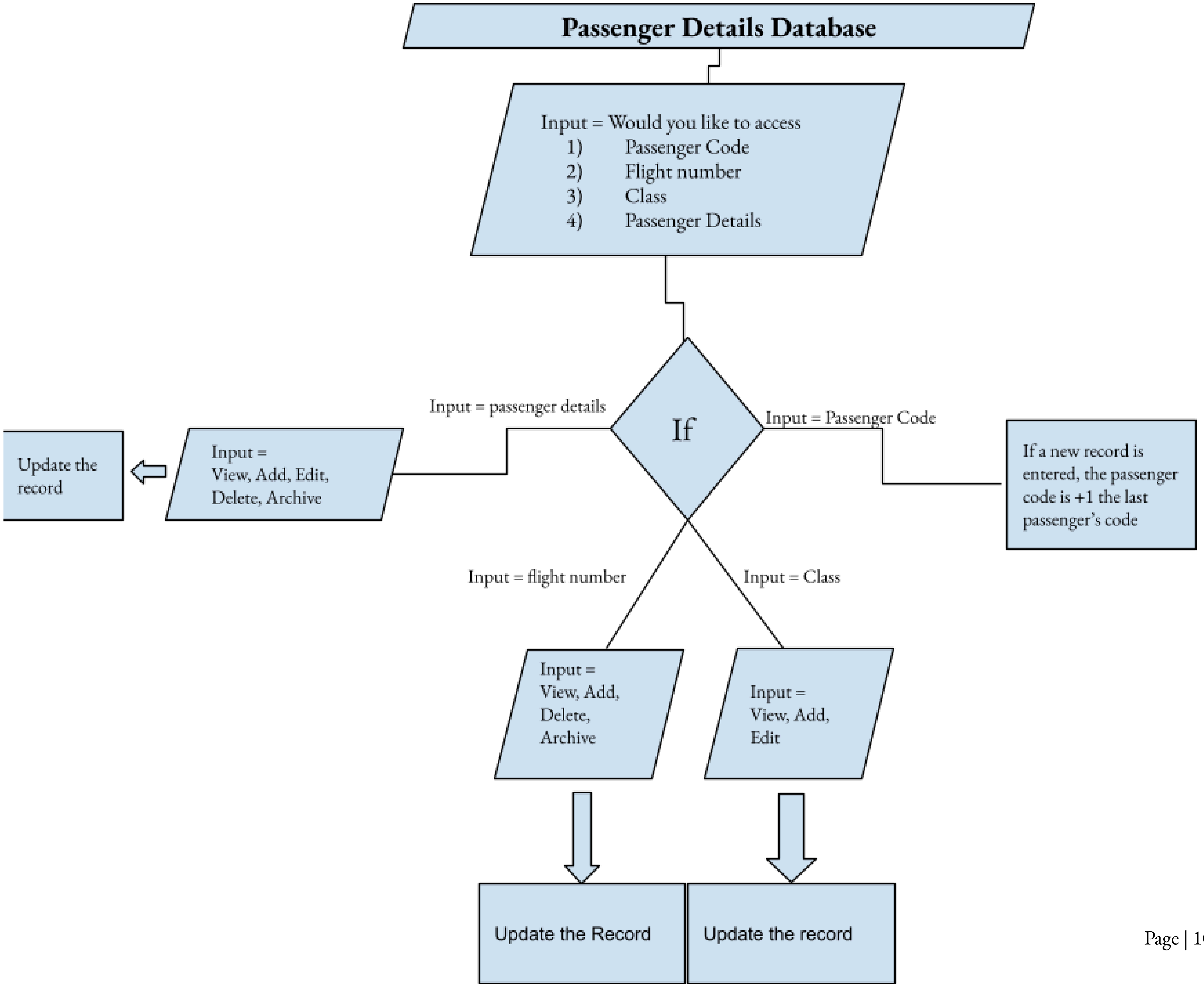
Will have an option to select category of employee (high executive or a normal employee)
Input salary of the person
Can have a formula for an increment of salaries by a particular percentage

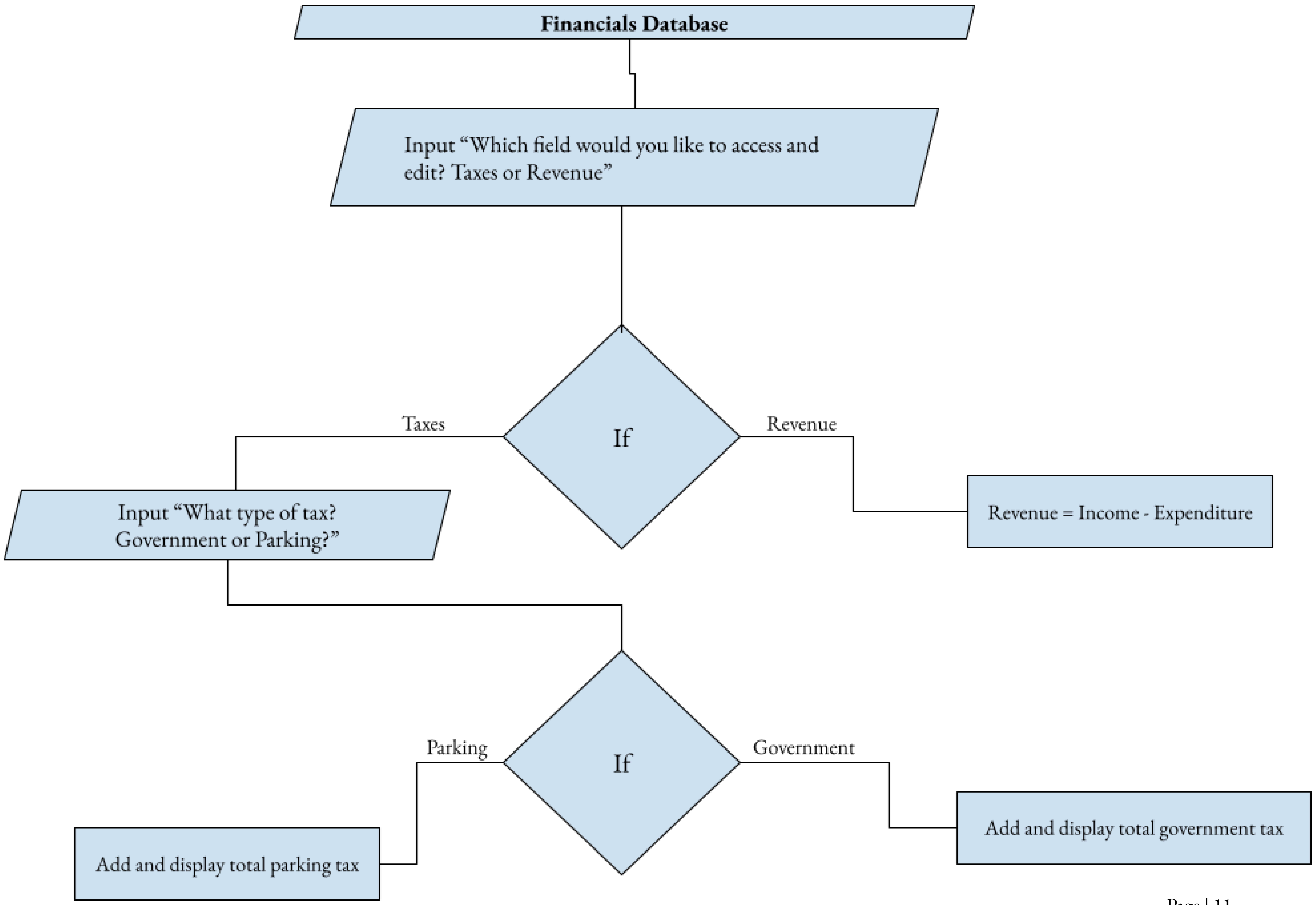
Flowcharts

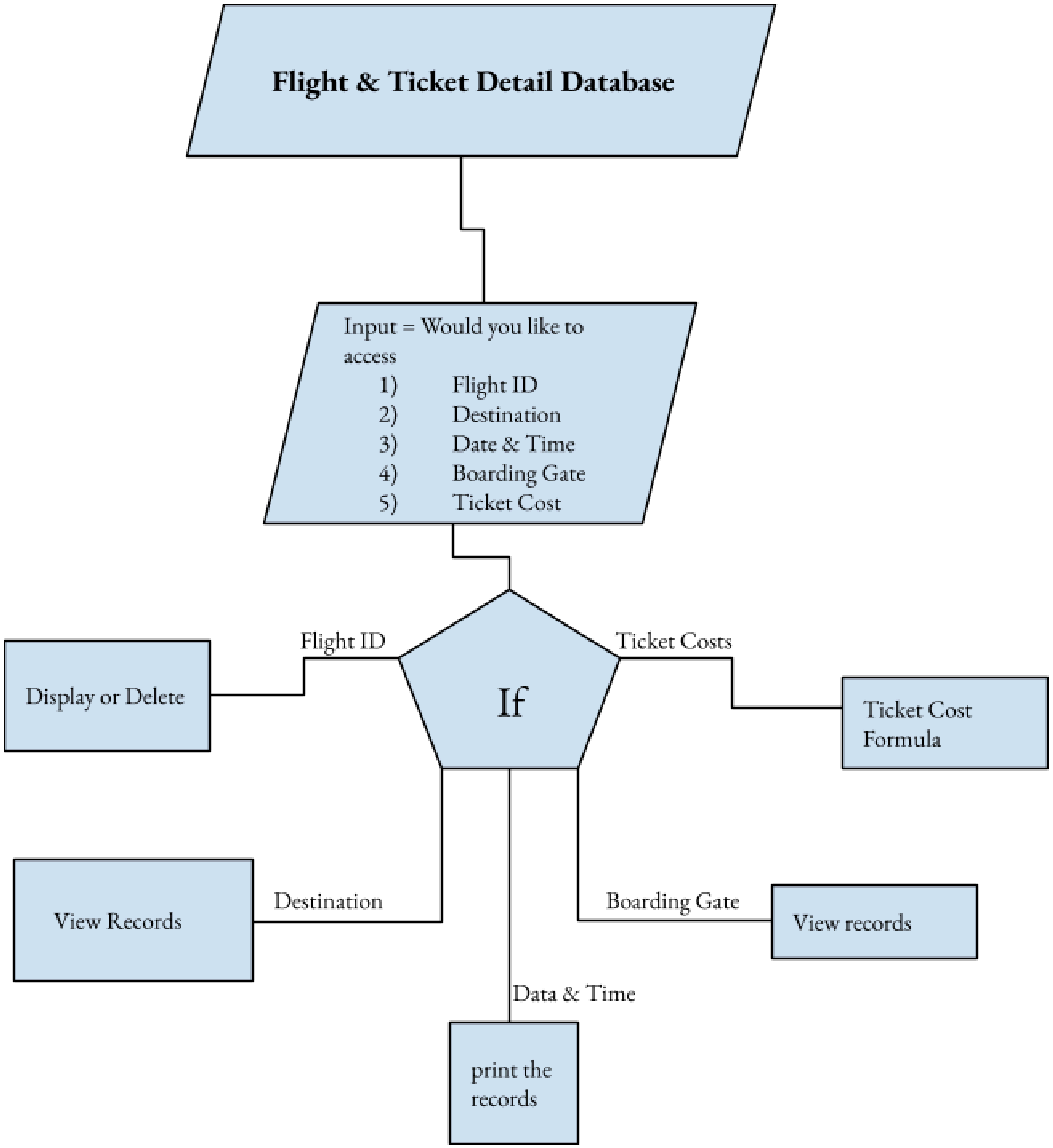


Login Menu
Input: Clearance Level
Input: Password
If False
If False
If False
If False
Clearance Level and Password = A
Clearance Level and Password = B
Clearance Level and Password = C
Clearance Level and Password = E
Clearance Level and Password = F
If True
If True
If True
If True
If True
****Menu****
Please select the required database:
1. Plane Maintainance (Type 1)
2. Passengers (Type 2)
3. Flight Details (Type 3)
4. Financials (Type 4)
5. Employees(Type 5)
****Menu****
Please select the required database:
1. Plane Maintainance (Type 1)
2. Flight Details (Type 3)
****Menu****
Please select the required database:
1. Passengers (Type 2)
2. Flight Details (Type 3)
****Menu****
Please select the required database:
1. Plane Maintainance (Type 1)
2. Flight Details (Type 3)
****Menu****
Please select the required database:
1. Financials (Type 4)
2. Employees (Type 5)


Plane Management Database
Choose functionality
Register a new plane
Report Snag
Flight/Edit
View
Enter ID, Plane Model, YoM, Passenger Capacity, Fuel Consumption, Endurance Hours flown = 0 Status = Airworthy
Hours flown = Hours flown+Flight Duration Status = Check
Print records
Status = Grounded


Passenger Details Database
Input = Would you like to access
1) Passenger Code
2) Flight number
3) Class
4) Passenger Details
If
Input = passenger details
Input = Passenger Code
Input = flight number
Input = Class
Input = View, Add, Edit, Delete, Archive
Update the record
If a new record is entered, the passenger code is +1 the last passenger's code
Input = View, Add, Delete, Archive
Input = View, Add, Edit
Update the Record
Update the record


Financials Database
Input "Which field would you like to access and edit? Taxes or Revenue"
If
Taxes
Revenue
Revenue = Income - Expenditure
Input "What type of tax? Government or Parking?"
If
Parking
Government
Add and display total parking tax
Add and display total government tax


Flight & Ticket Detail Database
Input = Would you like to access
1) Flight ID
2) Destination
3) Date & Time
4) Boarding Gate
5) Ticket Cost
If
Flight ID
Display or Delete
Ticket Costs
Ticket Cost Formula
Destination
View Records
Boarding Gate
View records
Data & Time
print the records