Air Traffic Ground Based Management System

A Project Report

Submitted by

Shreyash Palodkar N036
Shubham Patel N038
Shlok Sambre N042

Under the Guidance of
Prof. Kamal Mistry

in partial fulfillment for the award of the degree of

MBA in Technology Management

IN BRANCH OF STUDY
Computer Engineering

At

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING, MUMBAI

MARCH, 2021

DECLARATION

We , Shreyash Palodkar, Shubham Patel and Shlok Sambre, Roll No. N036, N038 and N042 MBA Tech (Computer Engineering), IV semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Students: _____, _____, _____, _____

Names: Shreyash Palodkar, Shubham Patel, Shlok Sambre

Roll Nos. : N036, N038, N042

Place: Mumbai

Date: 05-04-2021

ACKNOWLEDGEMENT

We would like to thank Prof. Kamal Mistry , Prof. Ameyaa Biwalkar and Prof. Sabina Tandon for their valuable guidance for the project, providing us an environment in which we were able to complete the project and most importantly, helping us whenever we were stuck somewhere in any situation, rather it be coding or document preparation without any fail. We also thank them for providing unwavering support, both morally and technically. At last, we would like to thank our college for providing us an opportunity where we can apply all our knowledge on a project and gain some extra knowledge.

CERTIFICATE

This is to certify that the project entitled “Air Traffic Ground-based Management System” is the bonafide work carried out by Shreyash Palodkar, Shubham Patel and Shlok Sambre of MBA Tech, MPSTME (NMIMS), Mumbai, during the IV semester of the academic year 2020-21, in partial fulfillment of the requirements for the Course Programming Language.

_____

Prof. Kamal Mistry

Internal Mentor

_____ _____

Examiner 1 Examiner 2

Table of contents

Chapter 1: Introduction to the System

1.1 Introduction

Air traffic in this world has been increasing day by day and so as the demand for air travel. Increased number of flights mean increased number of air traffic, which in turn means increase in ruckus created at airports as many flights come and go from airport, so each flight needs to be given a route, parking allocation, permissions for takeoff and landing and several other ground-based tasks. This all makes the job of staff of Air Traffic Control difficult and time consuming. Hence, for better execution of all the tasks at airport, a proper management system is necessary. So, we came up with the Air traffic ground-based management system for better running of airport as well as less load on Air Traffic Control.

The purpose of Air traffic ground-based management system is to allow the staff of air traffic control to track the movements of all the flights which are coming from some place and which may or may not want to go to other place. There is provision for staff of ATC to enter flight details which are coming towards airplane and may or may not want to leave from airport for a specific time. It will also facilitate the staff of ATC to give permission to airplane to land at the airport, allocation of parking or hardstands where boarding/deboarding of passengers or loading/unloading of cargo will take place, guide airplanes to their allocated hardstands, again give them permission to move towards runway for takeoff, guide airplanes towards runway by telling directions and giving permissions to airplanes to takeoff. Also, there is provision for emergency landing of airplanes as well as provision to vie history of all planes which departed from the airport.

1.2 Problem Statement

Problem in tracking of flights: There is a lot of problem for staff of ATC to track current position of different flights as one person can check for only one flight at a time.

Allocation of Parking by checking availability manually: Parking allocation manually can be difficult because of potential delays of flights, so it is better to automate parking allocation for airplanes.

Tracking of flight history: The retrieval of information about flight or airplanes is difficult, the immediate storage of flight details is difficult and also, it is difficult to update flight details.

1.3 Users of System

Database Administrator: The database administrator adds new users in the database who are the staff of ATC which will have access to the application and will provide with important tasks for flights.

Application Programmers and Database designers: Both of these updates the application's backend/frontend and the updation in database, if any.

Naïve Users: The whole staff of the ATC are naive users and they don't care about all is going on the back they just use the application for several tasks like, they all add new flights, give permissions, route, view the current status of terminals and do all the tasks which are necessary for ground based tasks at the airport for airplanes.

Chapter 2: System Design and Constraints

2.1 ER Model

There are total 6 entities in our ER Diagram out of which 5 are strong entities and 1 is week entity.

5 strong entities are: Admin, Status, Flttype, Terminal and Flights.

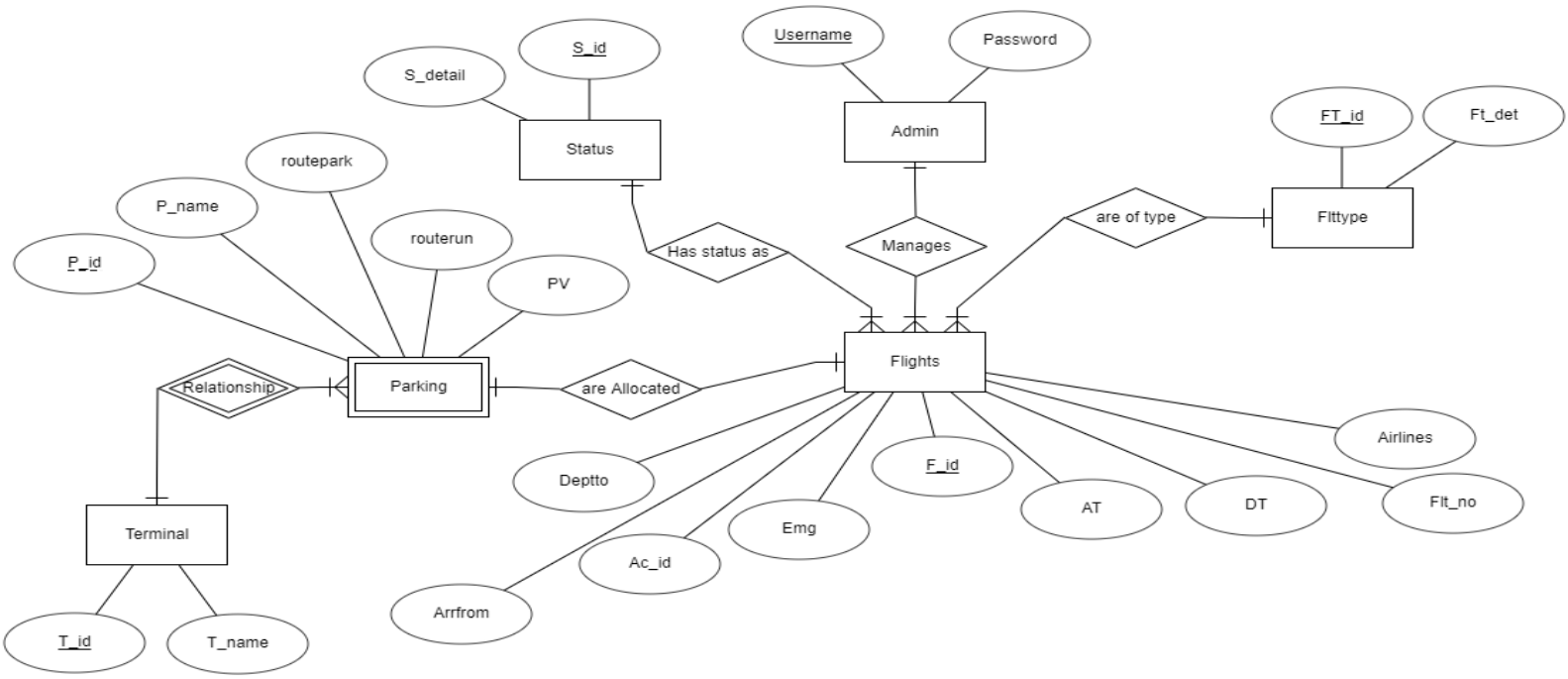
1 weak entity is parking which is dependent on terminal.

Attributes for respective entities are:

- Admin: username, password
- Status: S_id, S_detail
- Flttype: FT_id, Ft_det
- Terminal: T_id, T_name
- Parking: P_id, P_name, Routepark, Route Run, PV(parking vacancy)
- Flights: F_id, AT(arrival time), DT(departure time), Flt_no, Airlines, Emg, Ac_id, Arrfrom, Deptto

Relations between entities are:

- Admin and Flights: one to many relationship, one on admin side and many on flights side with total participation on many side.
- Status and Flights: one to many relationship, one on status side and many on flights side with total participation on many side.
- Flttype and Flights: one to many relationship, one on Flttype side and many on flights side with total participation on many side.
- Parking and Flights: one to one relationship without total participation
- Parking and Terminal: one to many relationship, one on Terminal side and many on Parking side, with total participation on many side.

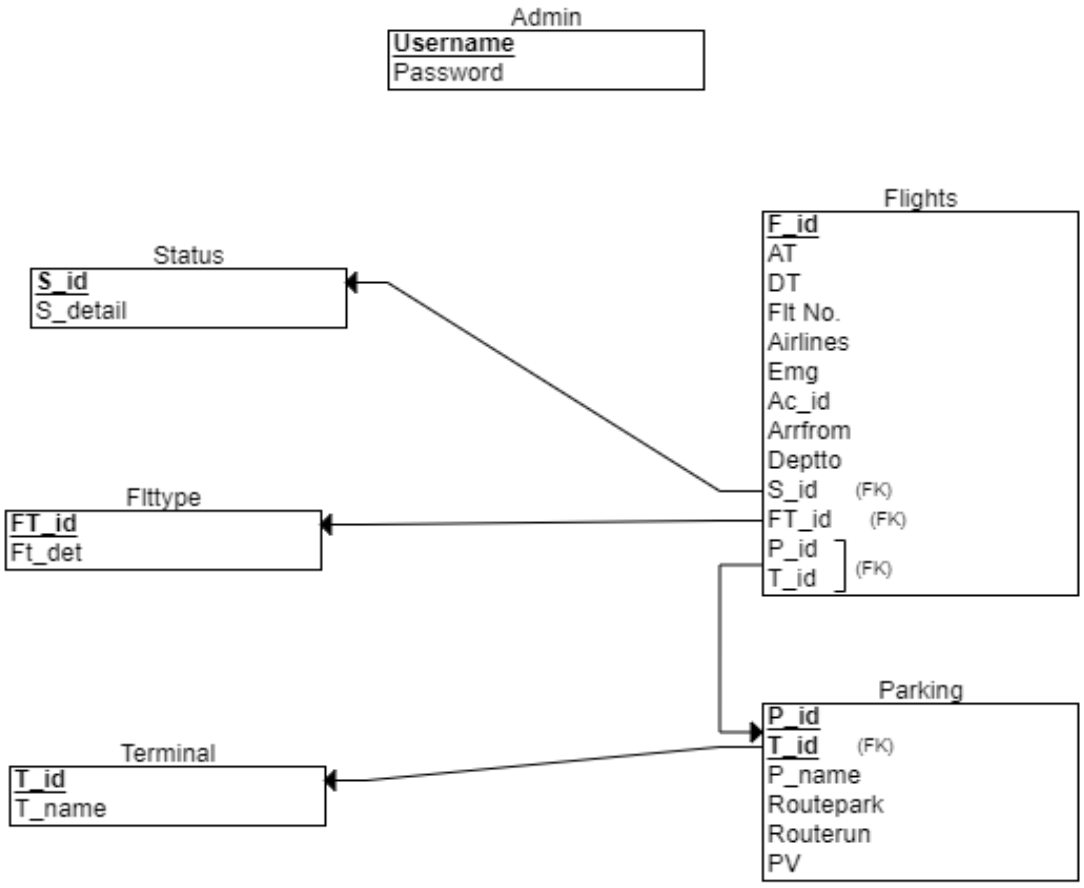


2.2 Reduction of ER Model to relational model

Steps in reduction of ER model to relational model are:

1) Admin Table consists of username and password columns.
2) Since, there is one to many relationship between flttype and flights, and total participation is on flight side, so we will add one extra column on flight side and include primary key of one side. Hence flights now consists of F_id, AT(arrival time), DT(departure time), Flt_no, Airlines, Emg, Ac_id, Arrfrom, Deptto and FT_id.
3) Since, there is one to many relationship between status and flights, and total participation is on flight side, so we will add one extra column on flight side and include primary key of one side. Hence flights now consists of F_id, AT(arrival time), DT(departure time), Flt_no, Airlines, Emg, Ac_id, Arrfrom, Deptto, FT_id and S_id.
4) Now, parking is a weak entity set and it depends on Terminal Table, so adding primary key of terminal in Parking, and the primary key combine becomes foreign key and discriminator, i.e. T_id and P_id.
5) Since, there is one to one relationship between parking and flights, so we will add one extra column on flight side and include primary key of Parking side. Now, primary key of Parking is T_id and P_id, but to prevent partial dependency, we see that P_id is enough as a foreign key for flights. Hence flights now consists of F_id, AT(arrival time), DT(departure time), Flt_no, Airlines, Emg, Ac_id, Arrfrom, Deptto, FT_id, S_id and P_id.

2.3 Schema Diagram



2.4 Constraints used:

- Constraints in Table Admin: Username is primary key and password not null.
- Constraint in Flttpye: FT_id is primary key and Ft_det is notnull.
- Constraint in Status: S_id is primary key and S_detail is not null
- Constraint in Terminal: T_id is primary key and T_name is not null
- Constraints used in Table Parking:
    - CONSTRAINT "F3" FOREIGN KEY("t_id") REFERENCES "terminal"("t_id") ON UPDATE CASCADE ON DELETE CASCADE
    - Primary key is P_id
    - P_name,PV,routerun,routepark,T_id not null

- Constraints used in Table Flights:
    - CONSTRAINT "F1" FOREIGN KEY("p_id") REFERENCES "parking"("p_id") ON UPDATE CASCADE ON DELETE CASCADE,
    - CONSTRAINT "F2" FOREIGN KEY("s_id") REFERENCES "status"("s_id") ON UPDATE CASCADE ON DELETE CASCADE
    - Primary key is F_id
    - S_id has default value as 0.
    - FT_id, Arrfrom, AT,Flt_no and Airlines Is not null.

2.4 Normalization Techniques applied on Relational Model:

- For relation between Flttype and Flight, there is no multivalued attribute or composite attribute, hence relation is in one normal form. Now, since there is also no partial dependency in the relation, it is in two normal form. Now, there is also no transitive dependency as we have already created two different tables for flight type and flights, so Ft_name cant be directly derived from F_id, hence relation is in three normal form also. The relation is also in BCNF form as in functional dependencies, LHS is a super key.
- For relation between Status and Flight, there is no multivalued attribute or composite attribute, hence relation is in one normal form. Now, since there is also no partial dependency in the relation, it is in two normal form. Now, there is also no transitive dependency as we have already created two different tables for Status and flights, so S_detail cant be directly derived from F_id, hence relation is in three normal form also. The relation is also in BCNF form as in functional dependencies, LHS is a super key.
- For relation between Parking and Terminal, there is no multivalued attribute or composite attribute, hence relation is in one normal form. Now, since there is also no partial dependency in the relation, it is in two normal form. Now, since there is a transitive dependency in which T_id derives P_id and P_id derives some other attributes, hence highest normal form for this table is two normal form.
- For relation between Parking and Flights, there is no multivalued attribute or composite attribute, hence relation is in one normal form. Now, there existed a partial dependency In which P_id and T_id were used as foreign key in table, while P_id was enough to be used as a foreign key, so we only use P_id as a foreign key in Flights so that two normal form can be achieved. Now, there is no transitive dependency in the relation, hence it is in three normal form also. Also, relation is in BCNF also as all the functional dependencies' LHS has a super key only.

Chapter 3: Implementation

3.1 Hardware and Software details

- VS Code: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.

- DB Browser: DB Browser for SQLite (DB4S) is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.
  DB4S is for users and developers who want to create, search, and edit databases. DB4S uses a familiar spreadsheet-like interface, and complicated SQL commands do not have to be learned. Controls and wizards are available for users: to Create and compact database files; Create, define, modify and delete tables; Create, define, and delete indexes; Browse, edit, add, and delete records; Search records; Import and export records as text; Import and export tables from/to CSV files; Import and export databases from/to SQL dump files; Issue SQL queries and inspect the results; Examine a log of all SQL commands issued by the application; Plot simple graphs based on table or query data.

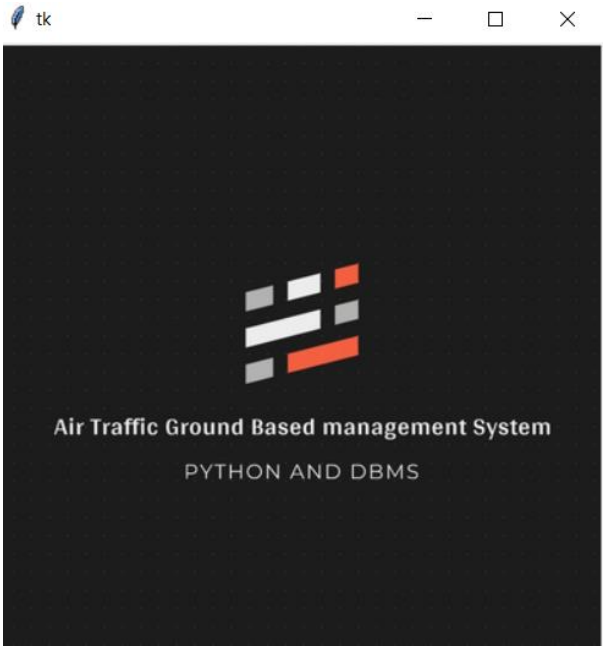
- Microsoft Photos: Microsoft Photos is Microsoft's modern image organizer, graphics editor, and video editor. It was first included in Windows 8 as a functional replacement for Windows Photo Viewer. Photos has Microsoft Sway integration and can use selected

photos as a source for creating a Sway project. Users can also upload photos to OneDrive, Facebook, Twitter , Instagram and GroupMe for sharing or saving on cloud. We are using photos in our project to show pictures of routes towards various hardstands present at the airport.

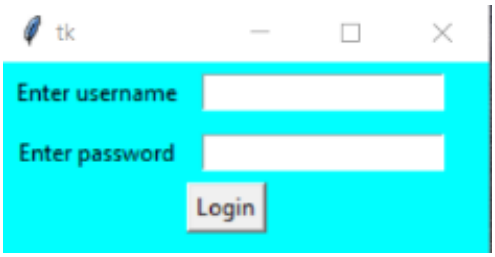
3.2 Tools and modules used:

- Tkinter: Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python. The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh Tkinter is free software released under a Python license. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application. Tkinter is not the only GuiProgramming toolkit for Python. It is however the most commonly used one.
- Datetime: In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.Datetime module supplies classes to work with date and time.These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
- PIL: Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux.
- IO: This module is quite useful when you want to perform file-related I/O operations (eg. File reading/writing). While you can use normal read() and write methods to read/write a file, this module gives us a lot more flexibility regarding these operations.

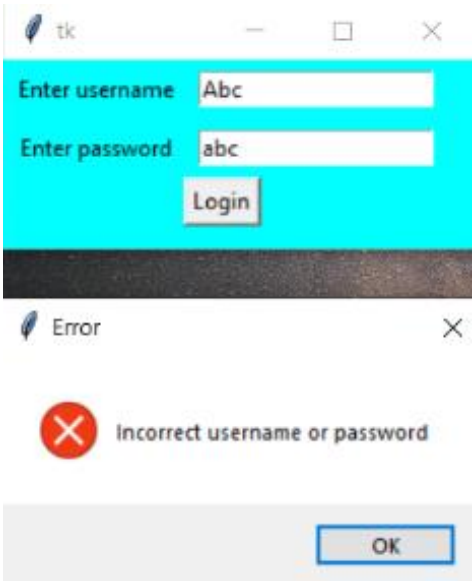
3.2 Screenshots and Descriptions:



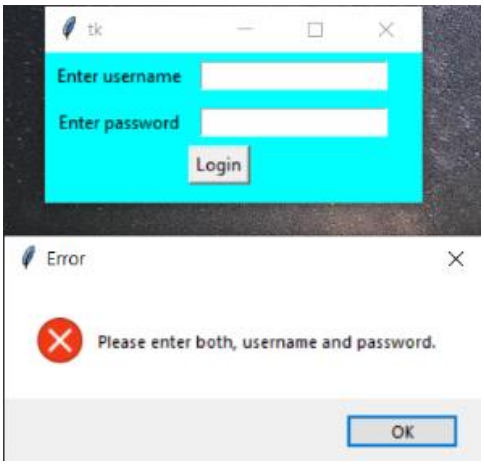
Splash Screen



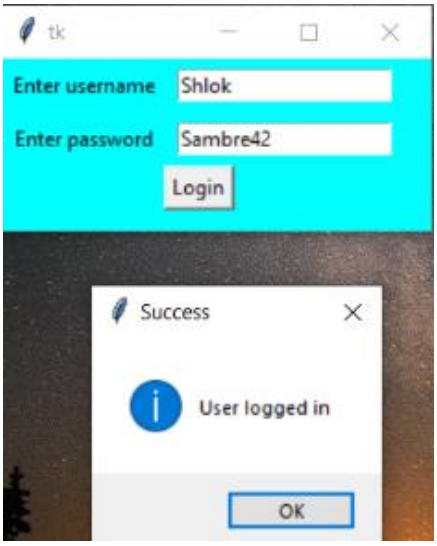
Login Screen



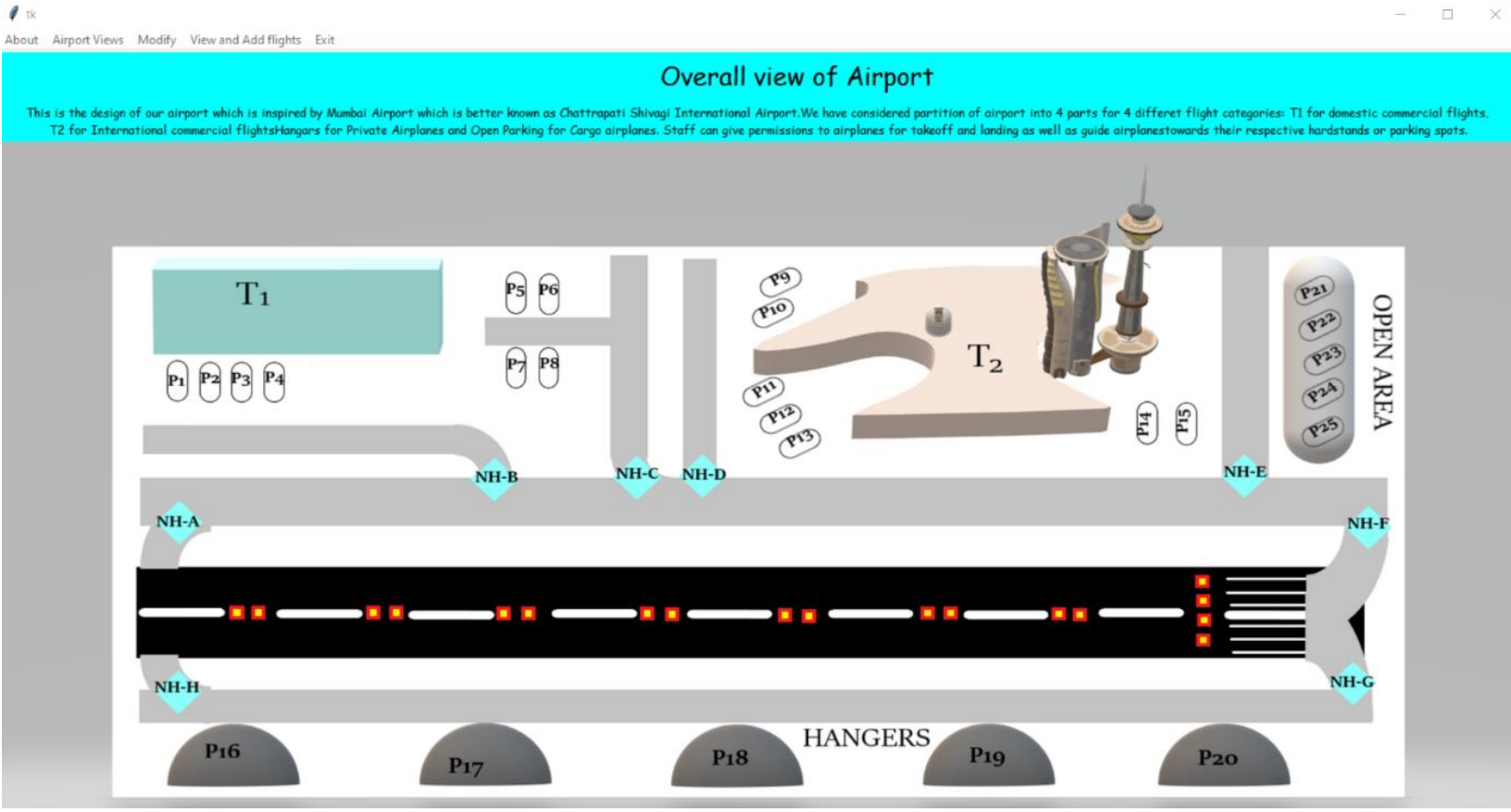
Error handling for incorrect entry details



Error handling for no entry fillled



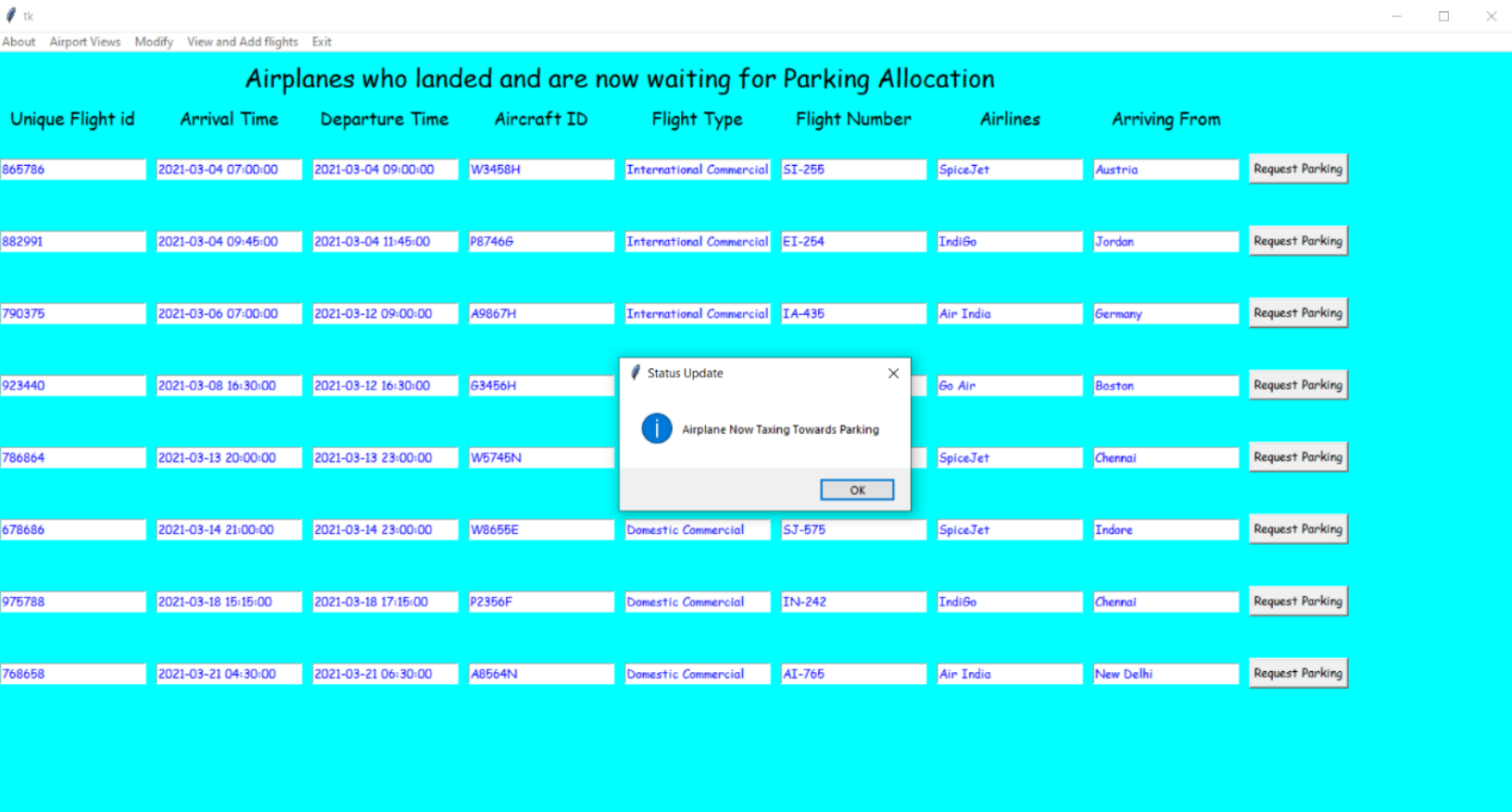
Login Successful



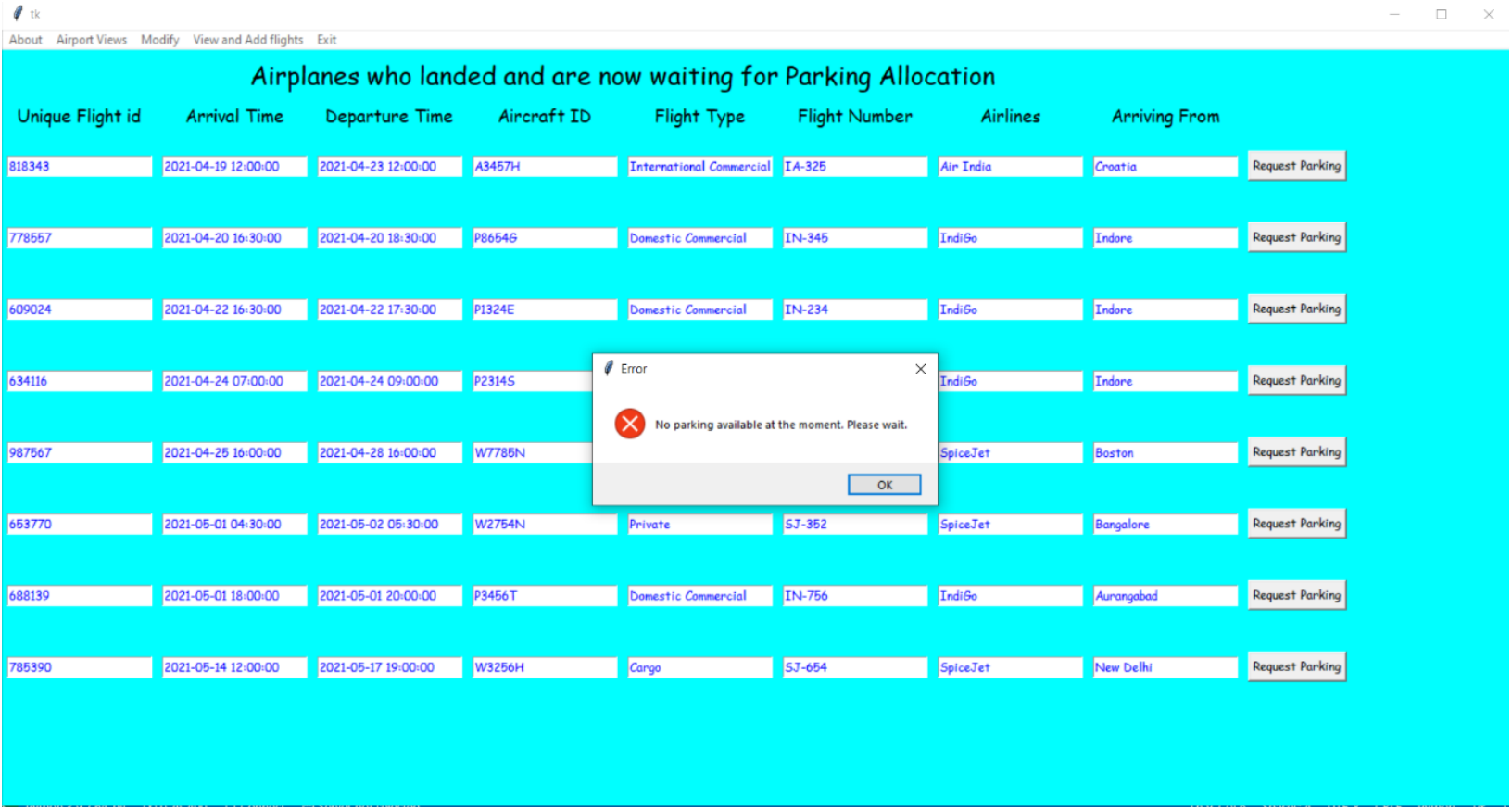
About Page

Airplanes coming to land

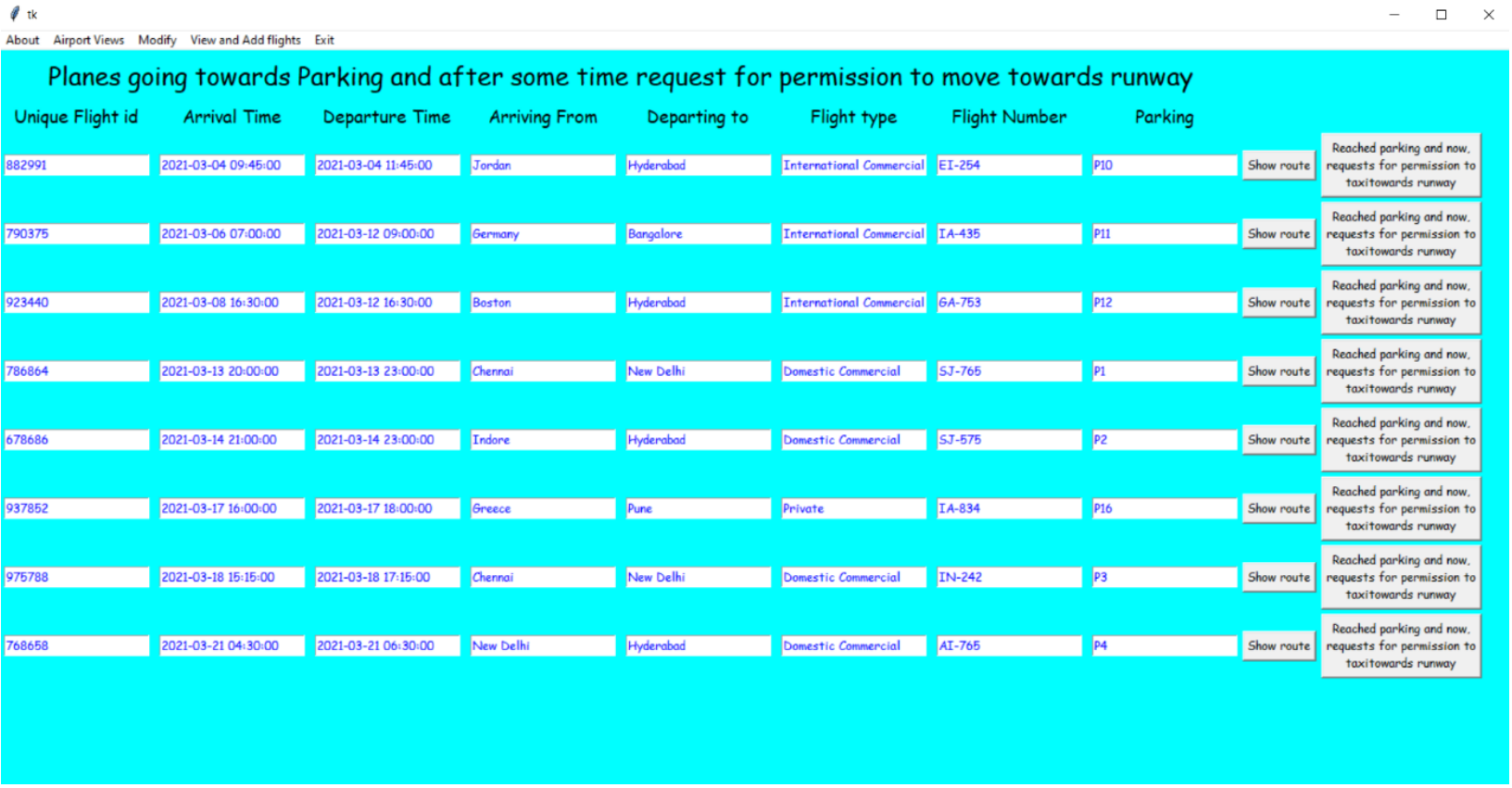
Plane landed updatation dialog



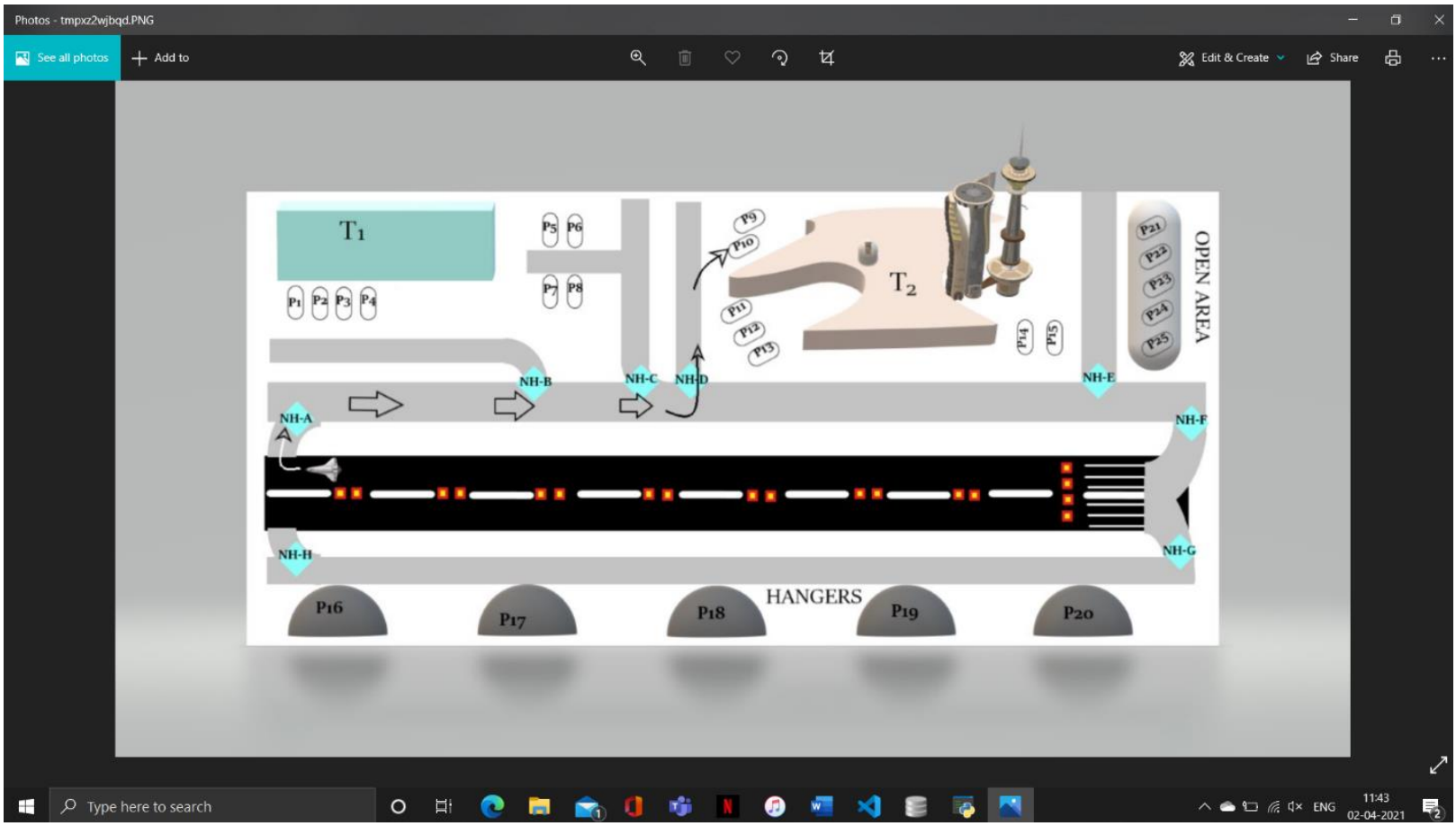
Airplanes who landed and are now requesting for parking allocation along with updation diaolog



Error handling while allocation of parking while no parking is available.

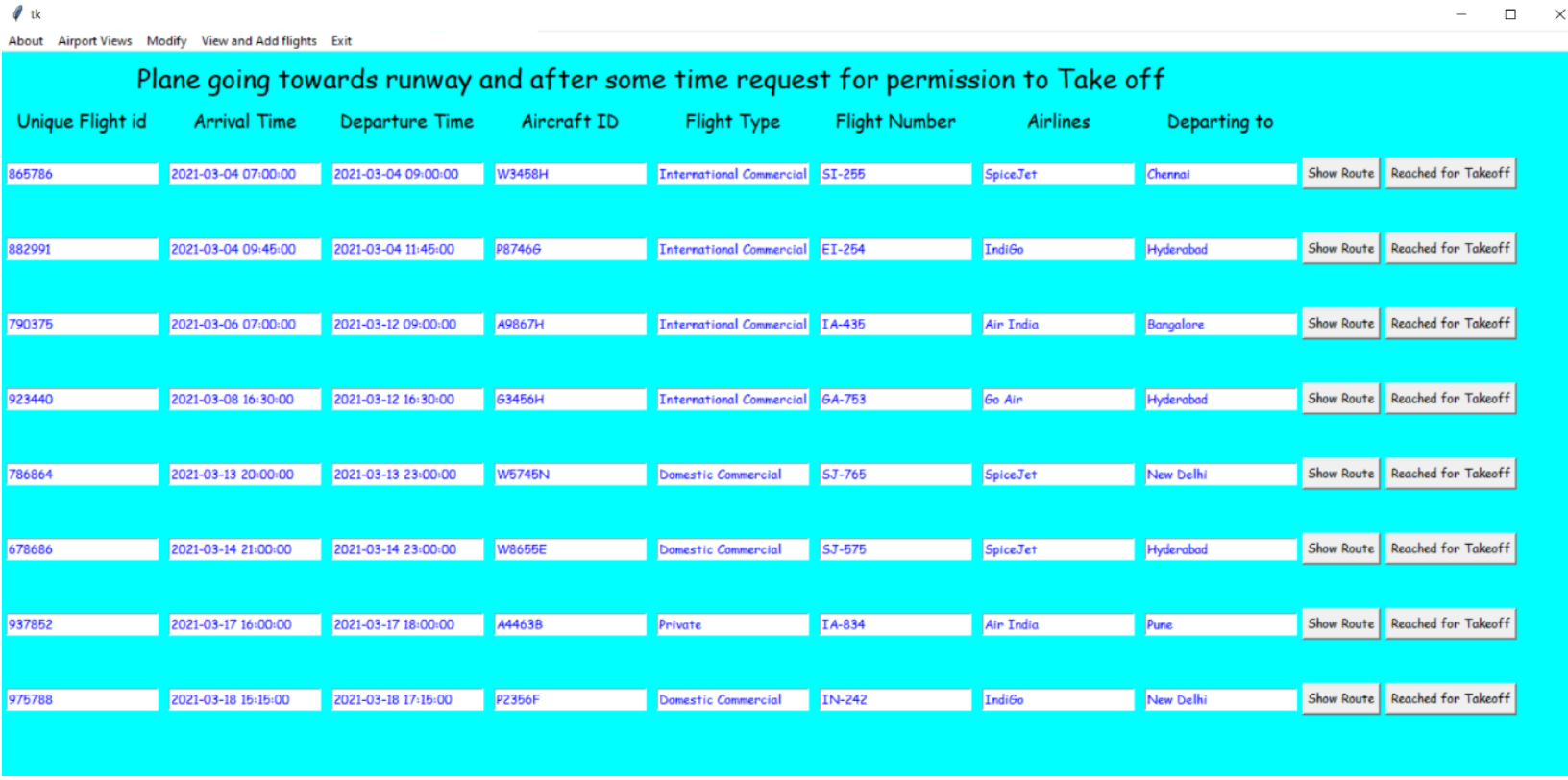


Airplanes who landed and are taxing towards parking or are parked.

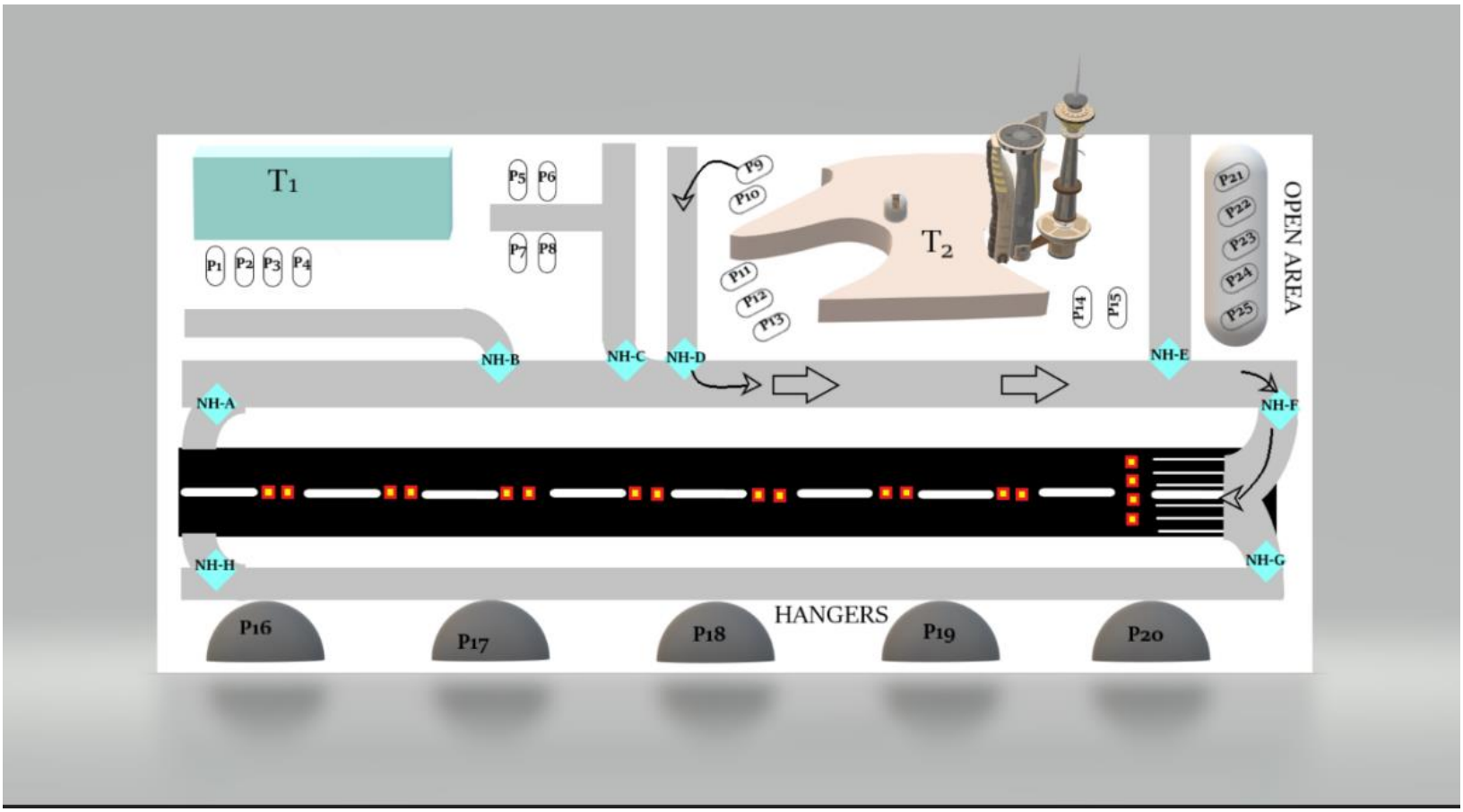


Route from runway to parking for airplanes

Plane taxing towards runway with updation diaolog

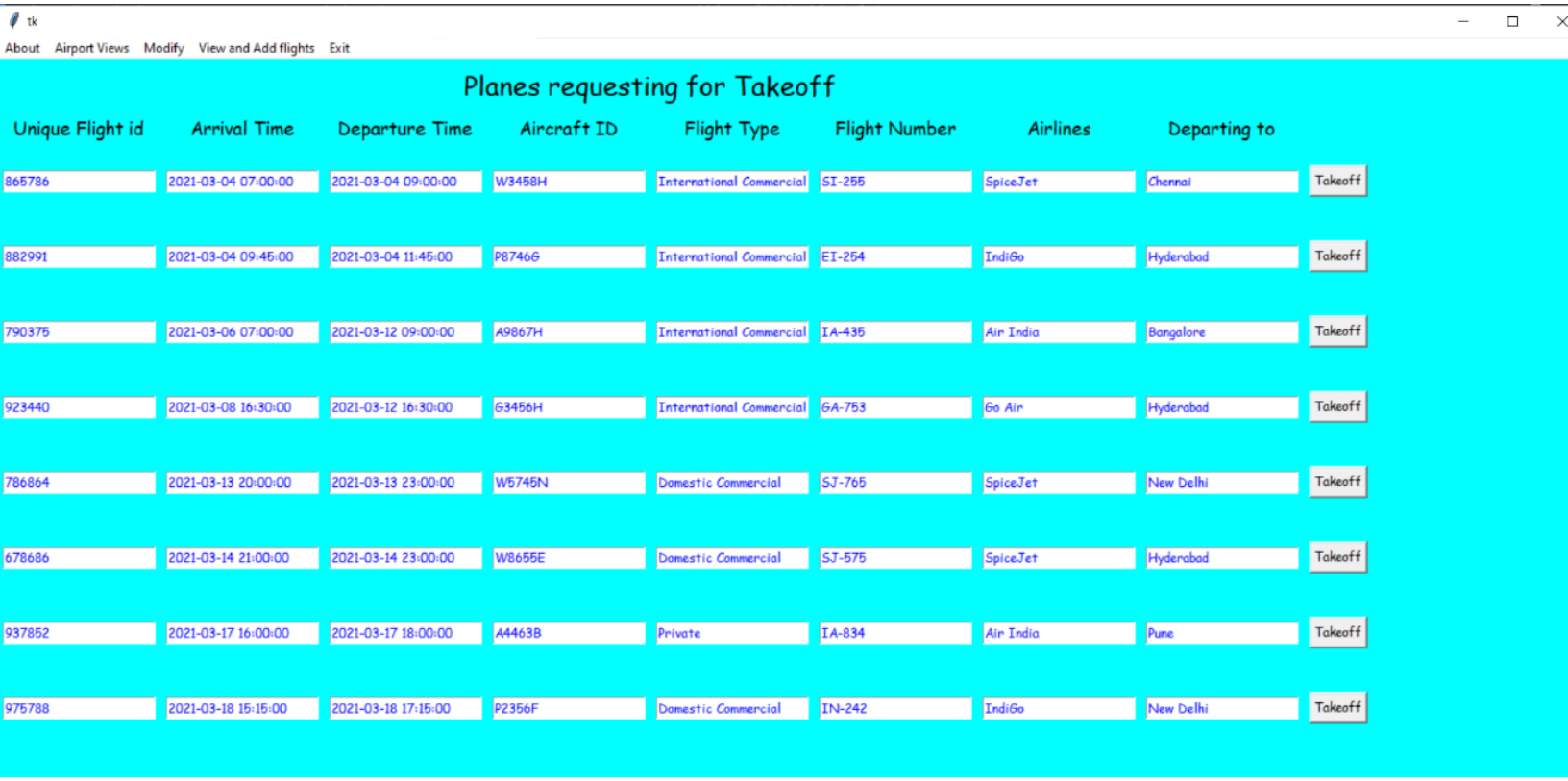


Planes taxing towards runway

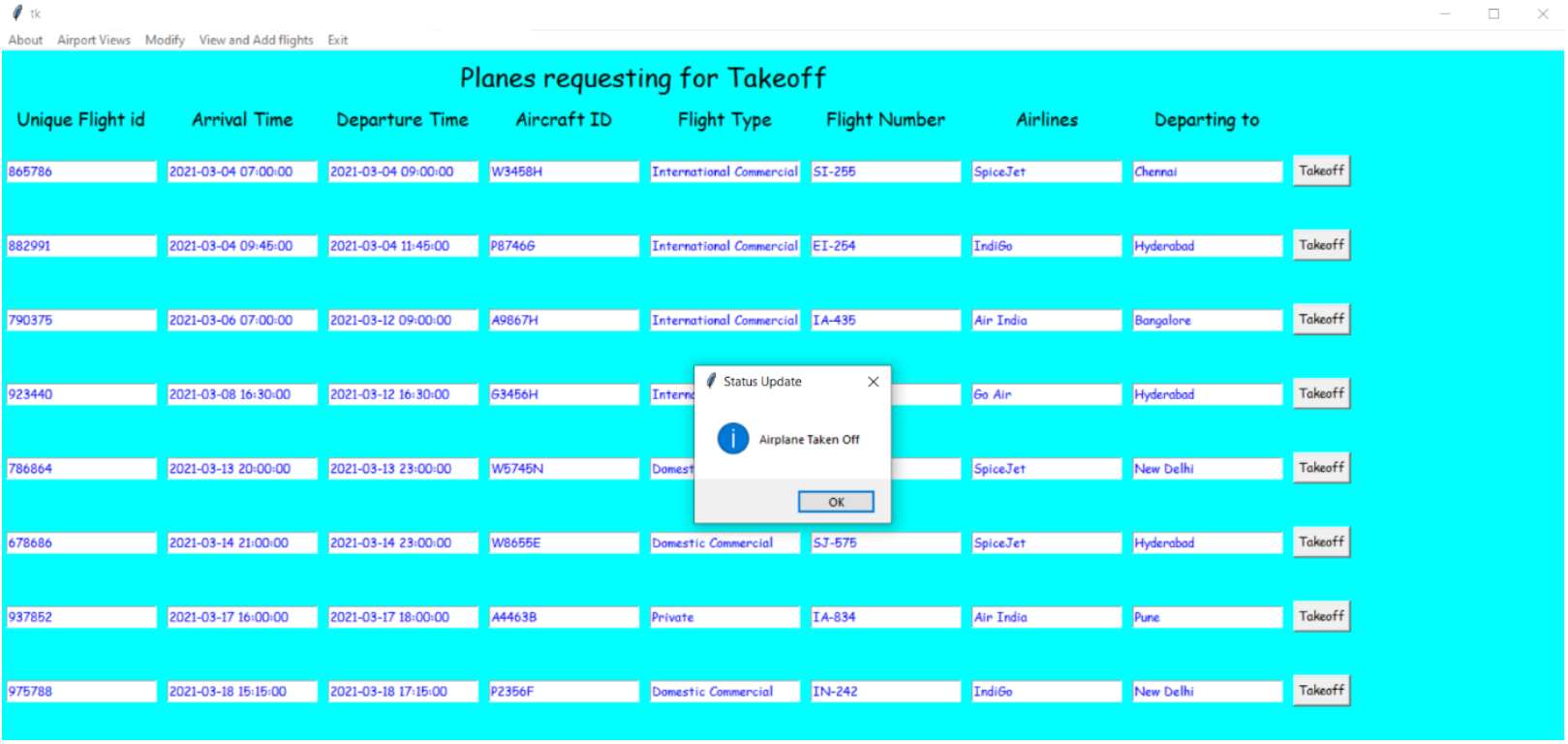


Route for airplanes from parking to runway

Airplane ready to takeoff dialog box

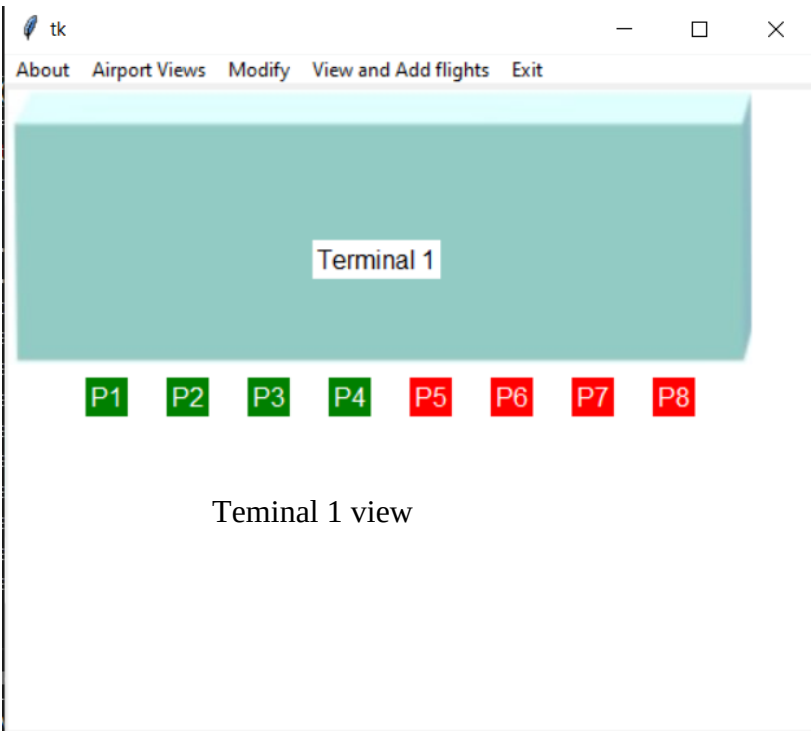


Airplanes which are requesting for takeoff permission

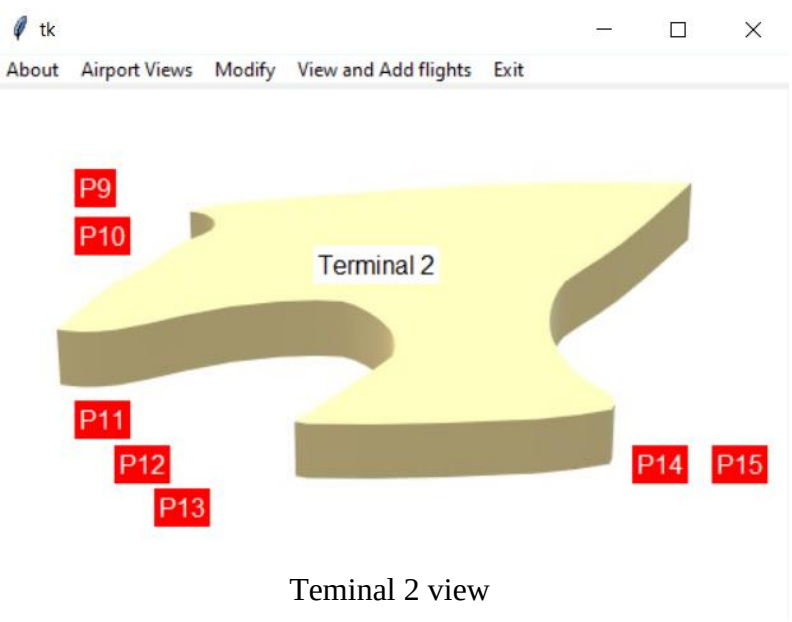


Airplane taken off updation dialog

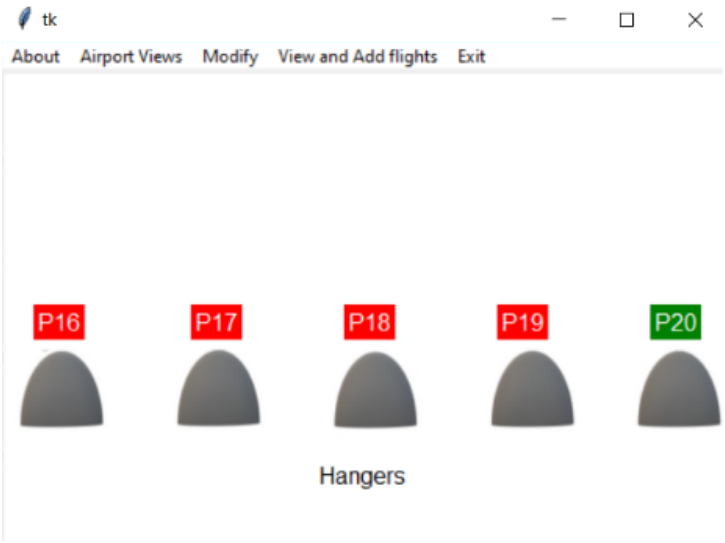
Flight history screen



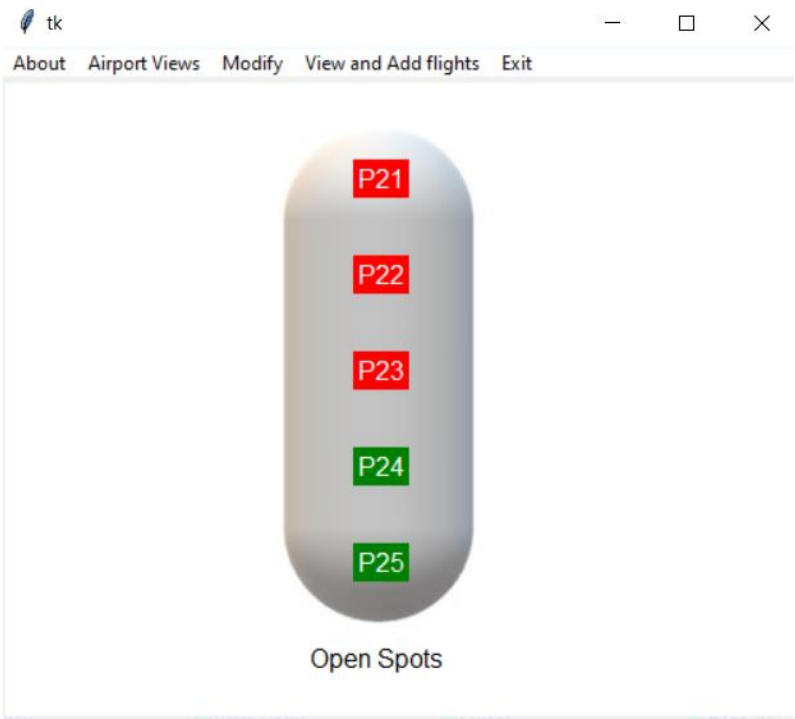
Teminal 1 view



Teminal 2 view



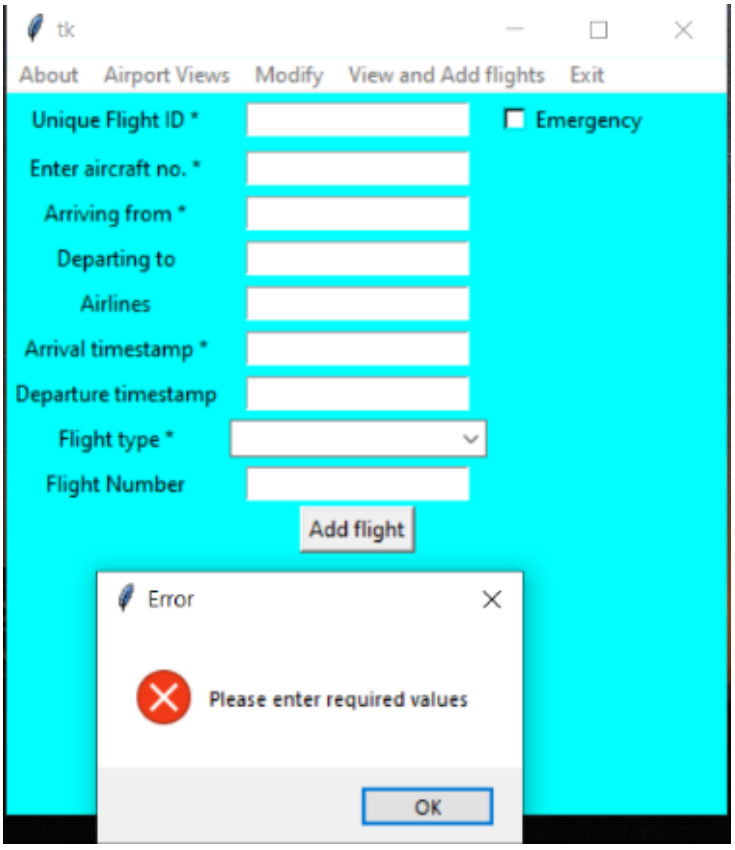
Hangars view



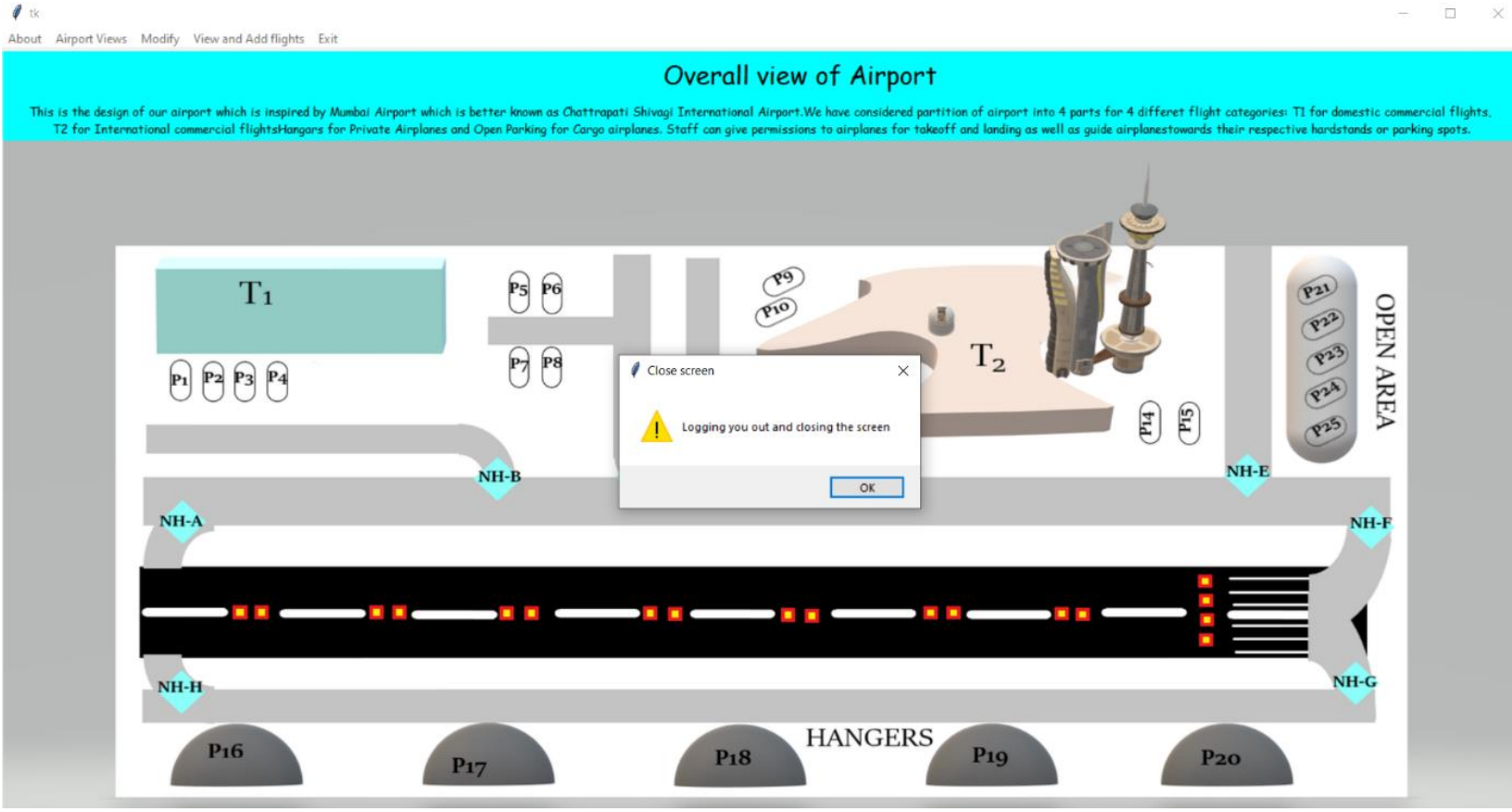
Open spots view

Add flight screen with success dialog

Add emergency flight screen with success dialog



Error handling for incomplete details



Logging out dialog box on click of exit button

Database structure:

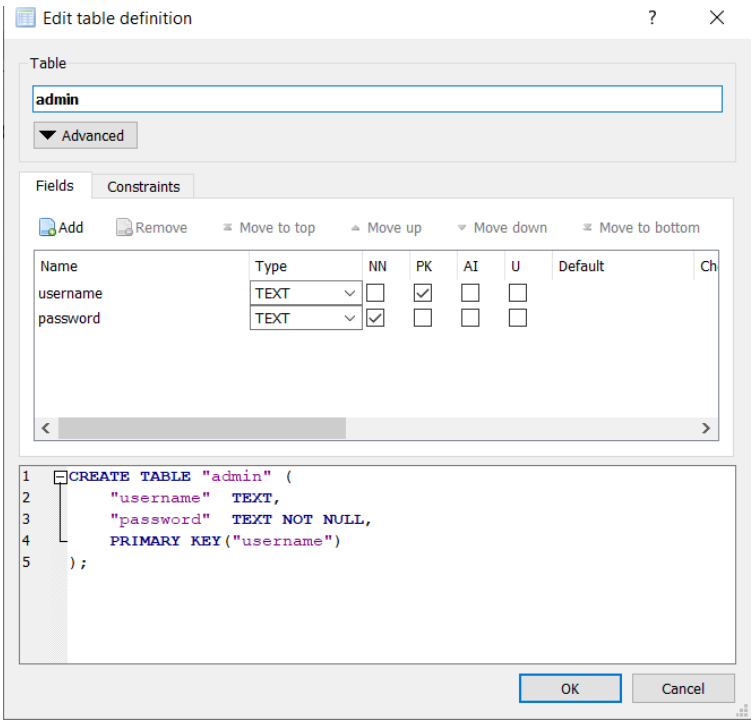


Table Admin

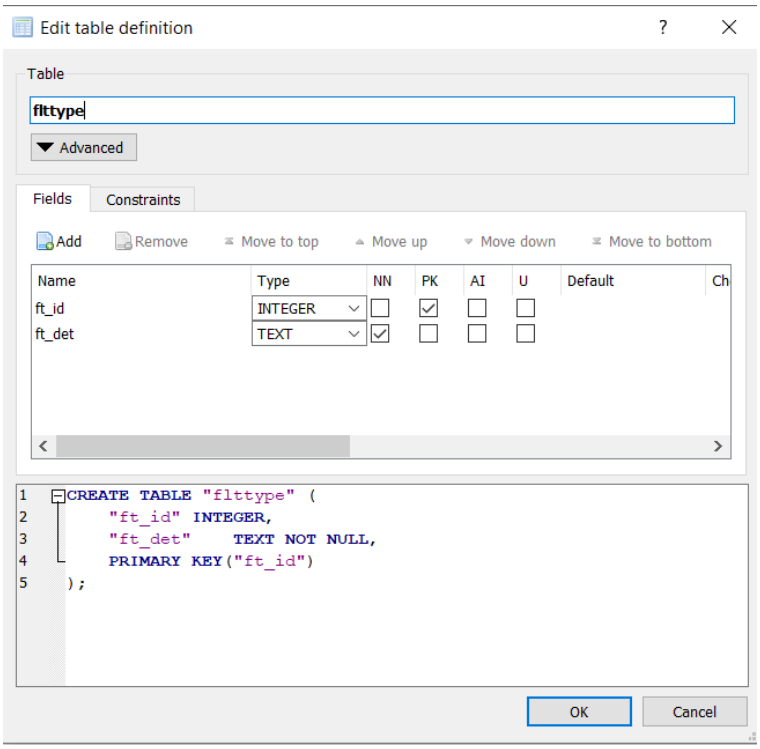


Table flttype

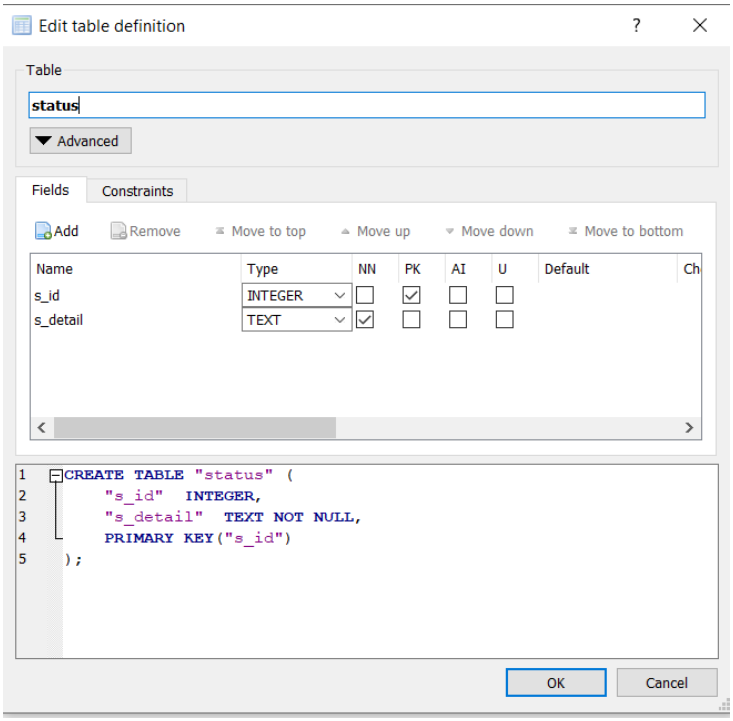


Table flttype

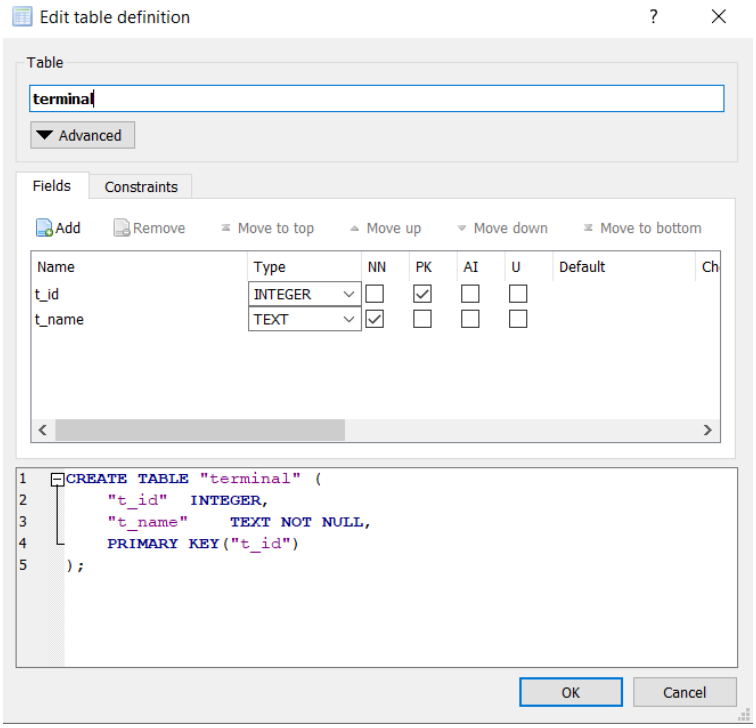


Table terminal

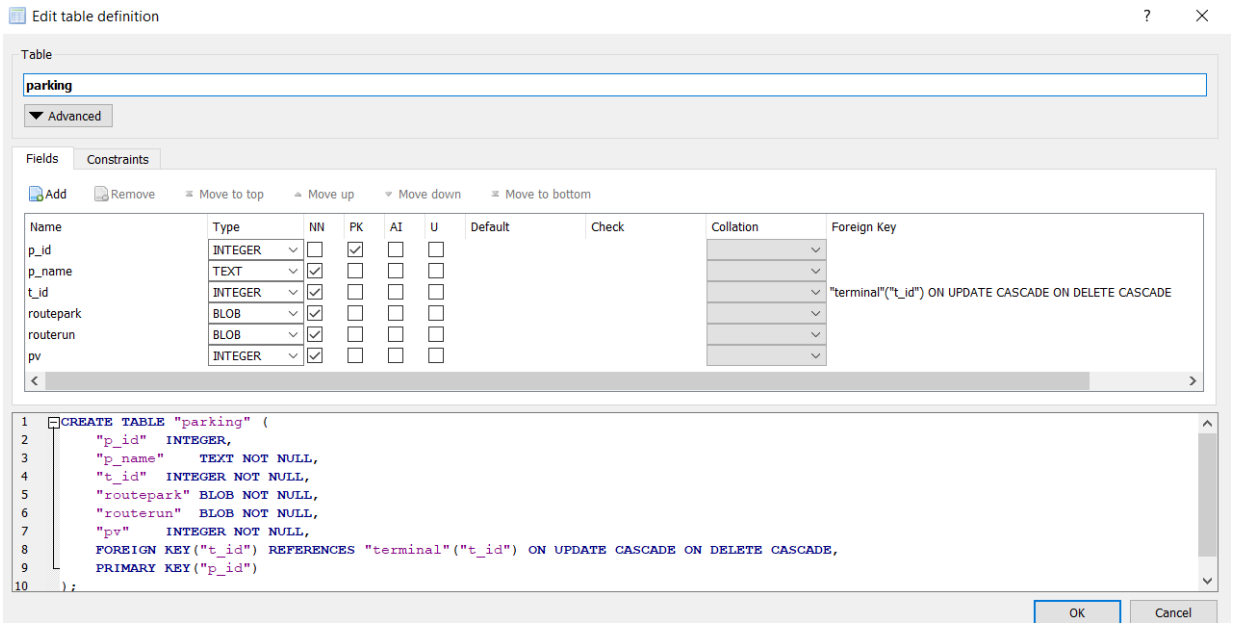


Table parking

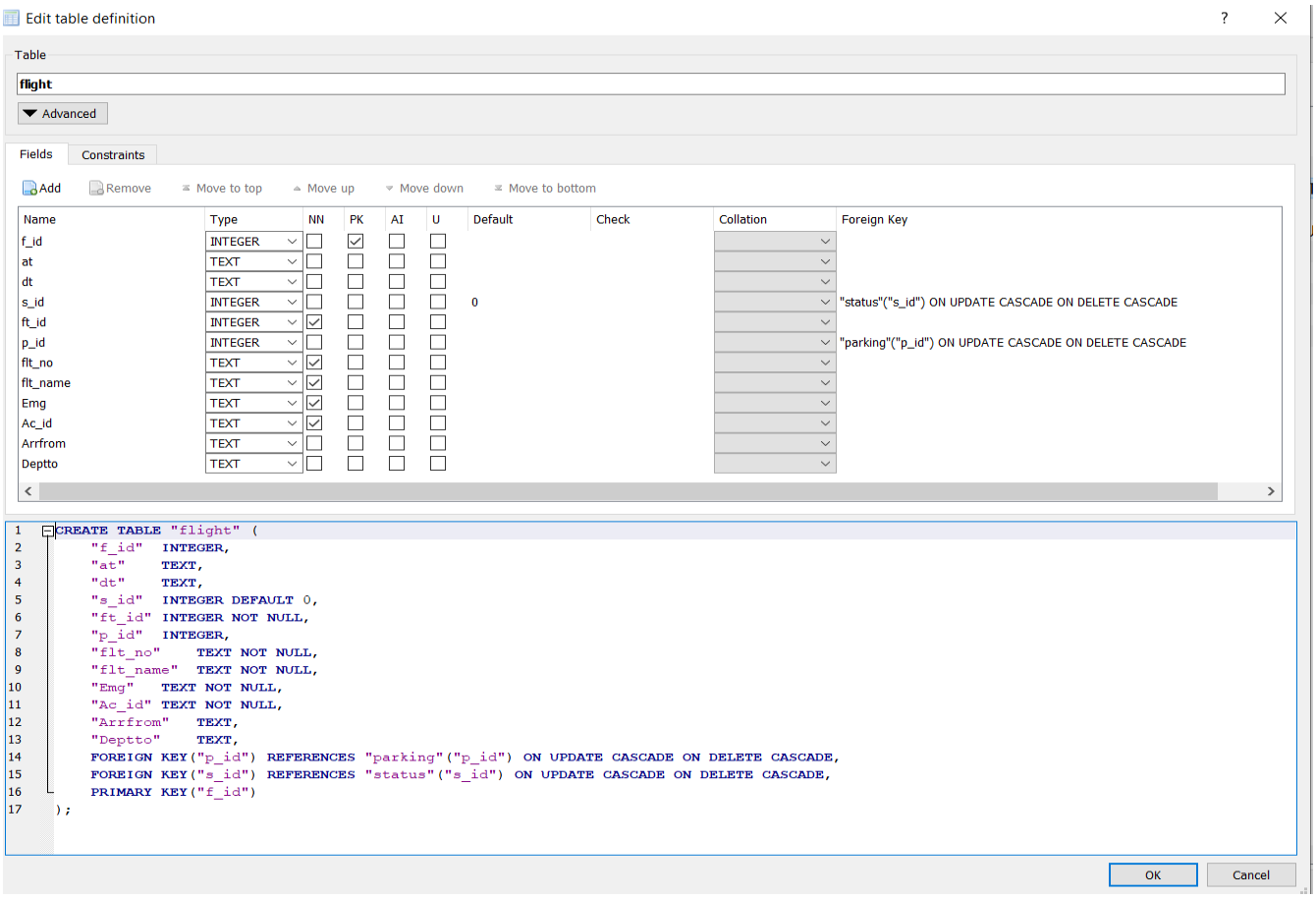


Table flight

Chapter 4: Conclusion and Future Work

Air traffic in this world has been increasing day by day and so as the demand for air travel. Increased number of flights mean increased number of air traffic, which in turn means increase in ruckus created at airports as many flights come and go from airport, so each flight needs to be given a route, parking allocation, permissions for takeoff and landing and several other ground-based tasks.

This all makes the job of staff of Air Traffic Control difficult and time consuming. Hence, for better execution of all the tasks at airport, a proper management system is necessary. So, we came up with the Air traffic ground-based management system for better running of airport as well as less load on Air Traffic Control.

Our application provides a simple UI which enables the staff of ATC to smoothly run all ground based air traffic commands so that the job of them is easier, as well as the movement of all the flights are smooth and fast to increase the effectiveness of the airport. We hope that the airport can be as most effective as it can so that the number of flights going from and coming to airport increases which will also help in raising economy.

We know that there are some other various high tech software which the ATC uses at the airport for performing all of these, but those software also include some other features which makes it difficult to manage ground based operations, so we propose this for only ground based operations.

There are some features that we would like to work upon in our program so we can try to make it into an application which all staff of the ATC can use all together and will share the resources of the database and UI. Also, we would like to enable the share of the route in airplane cockpit as soon as the plane has landed so that plane can move by itself to its allocated parking spot and the staff of ATC can manage other flights. Also, we would also want that before giving permission to airplanes to take-off and land, there should be a check to see it the runway is empty or not. And, at last, there should be an option for multiple runways which is present in various airports of the world which increases effectiveness of the airports.