# Flight Reservation

## CS 157A

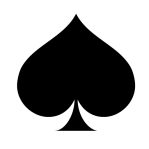
Introduction to Databases

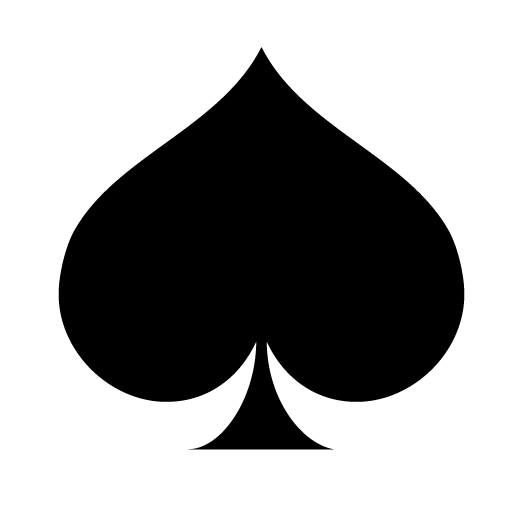
## Team Members

Amir Radman

Sophia Tan

David Ruvalcaba

TEAM ACE



Database Schema

drop database if exists flightReservation;

create database flightReservation;

use flightReservation;

drop table if exists user;

create table user

(uID INT AUTO\_INCREMENT Primary Key,

uNAME VARCHAR(30) NOT NULL,

age INT);

ALTER Table user AUTO\_INCREMENT = 100;

drop table if exists flightList;

create table flightList

(fID INT AUTO\_INCREMENT Primary Key,

aName VARCHAR(30) NOT NULL,

numSeats INT NOT NULL);

ALTER Table flightList AUTO\_INCREMENT = 200;

create table reservation

(uID INT,

fID INT,

reservedDate Date NOT NULL Default '0000-00-00' ,

updatedAt timestamp NOT NULL on update current\_timestamp default current\_timestamp,

PRIMARY KEY (uID,fID,reservedDate),

FOREIGN KEY (uID) references user (uID) on delete cascade,

FOREIGN KEY (fID) references flightList (fID) on delete cascade);

drop table if exists seat;

create table seat

(uID INT,

fID INT,

seatNumber INT NOT NULL auto\_increment UNIQUE,

PRIMARY KEY (uID,fID,seatNumber),

FOREIGN KEY (uID) references user (uID) on delete cascade,

FOREIGN KEY (fID) references flightList (fID) on delete cascade);

ALTER Table seat AUTO\_INCREMENT = 300;

drop table if exists canceledReservation;

create table canceledReservation

(uID INT,

fID INT,

canceledDate Date NOT NULL Default '0000-00-00',

reservedDate Date NOT NULL Default '0000-00-00',

PRIMARY KEY (uID,fID,canceledDate)

);

drop table if exists Archive;

create table Archive

(uID INT,

fID INT,

updatedAt timestamp NOT NULL);

DROP TRIGGER IF EXISTS INSERTTORES;

delimiter //

CREATE TRIGGER INSERTTORES

AFTER INSERT ON RESERVATION

FOR EACH ROW

BEGIN

insert into seat(uID,fID) values(new.uID,new.fID);

END;//

delimiter ;

DROP TRIGGER IF EXISTS INSERTTOCANCEL;

delimiter //

CREATE TRIGGER INSERTTOCANCEL

AFTER delete ON reservation

FOR EACH ROW

BEGIN

insert into canceledreservation(uid,fid,canceledDate,reservedDate) values (old.uid,old.fid,current\_date(),old.reservedDate);

delete from seat where uID = uID and fID = fID;

END;//

delimiter ;

drop PROCEDURE if exists archivedReservation;

delimiter //

create PROCEDURE archivedReservation(IN value varchar(55))

BEGIN

insert into Archive select uid,fid,updatedAt from reservation where date(updatedAt) < value ;

delete from reservation where date(updatedAt) < value;

END; //

delimiter ;

Main GUI Application (LoginPopUp.java)

## C:\Users\amirr\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (11).png

All Select Statements

General User Commands

1. Select uid,fid,reservedDate from reservation where uid = ?
2. Select \* from flightList where aName = ?
3. select FL.fid,aName,numSeats,uid,reservedDate from flightList FL left Outer JOIN reservation R on FL.fid = R.fid
4. select distinct MonthName(reservedDate) as 'High Season Month(s)' from reservation group by Month(reservedDate) having(count(\*)>=3)
5. Select \* from flightList where numSeats = ?

Administrator Commands

1. select \* from user JOIN reservation using(uID) group by uid having(age >= ?)
2. select \* from user JOIN canceledReservation using(uID) group by uid having(age >= ?)
3. Select uid,uname,age from user u1 where (select count(\*) from canceledreservation where u1.uid = uid group by uid)>= ?
4. Select \* from user
5. Select \* from flightList
6. select distinct uid from reservation where uid NOT IN (select uid from canceledreservation)
7. select avg(age) as 'averageAge' from user U Natural join (select distinct uid from reservation group by uid) T

All Update Statements

\*\*\* There were no update statements \*\*\*

All Delete Statements

General User Commands

1. delete from Reservation where uid= ? and fid=? and reservedDate=?

Administrator Commands

1. delete from user where uid = ?
2. Delete from flightList where fid = ?

All Insert Statements

General User Commands

1. insert into user(uName,age) values(?,?)
2. insert into reservation(uid,fid,reservedDate) values(?,?,current\_Date())

Administrator Commands

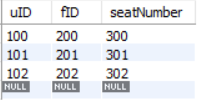
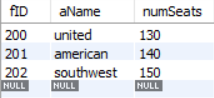
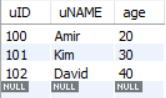
1. insert into flightList(aName,numSeats) values(?,?)

15 Functional Requirements

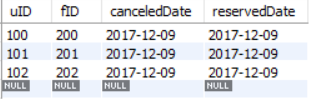
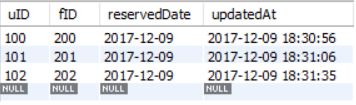
|  |  |
| --- | --- |
| **Functional Requirements** | **Corresponding SQL Query** |
| Adding Users | Insert into user(uName,age) values(?,?) |
| Deleting Users | Delete from user where uID = ? |
| Adding Reservation | Insert into reservation (uID , fID,reservedDate ) values(?,?,current\_Date()) |
| Canceling Reservation | Delete from reservation where uid = ? and fid = ? |
| Users with reservation who have not canceled any flights AT ALL  (LOYAL CUSTOMERS) | select distinct uid  from reservation where uid NOT IN (select uid from canceledreservation); |
| Adding Flights | Insert into flightList(aName, numSeats)  values(?,?) |
| Deleting Flights | Delete from flightList where fid = ? |
| User’s reservation with age >= ? | select \* from user JOIN reservation using(uID) group by uid having(age >= ?); |
| User’s cancellation with age >= ? | select \* from user JOIN canceledReservation using(uID) group by uid having(age >= ?); |
| Finding flights using airline name | Select aName from flightList group by aName having (aName = ?) |
| Finding flights with ? # seats | Select fid,aName from flightList where numSeats = ? |
| Flights that have not been reserved at all | select \* from flightList FL left Outer JOIN reservation R on FL.fid = R.fid; |
| High Season Month(s): Months with more than 3 reservations | select distinct MonthName(reservedDate) as 'High Season Month(s)' from reservation group by Month(reservedDate) having(count(\*)>=3); |
| Users who have canceled more than ? reservation | Select uid,uname from user u1 where ? < (select count(\*) from canceledreservation where u1.uid = uid group by uid); |
| Average age of users who reserved | Select avg(age) as ‘averageAge’ from user U NATURAL JOIN  (select distinct uid from reservation group by uid) T; |

Relations Screenshots with Populated Data

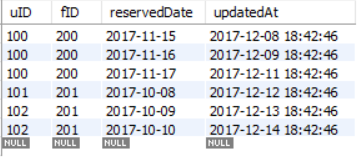
## User FlightList Seat



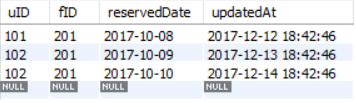
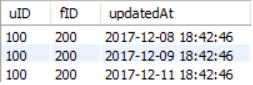
**Reservation** **CanceledReservation**



Reservation Before ArchiveArchive(Empty)



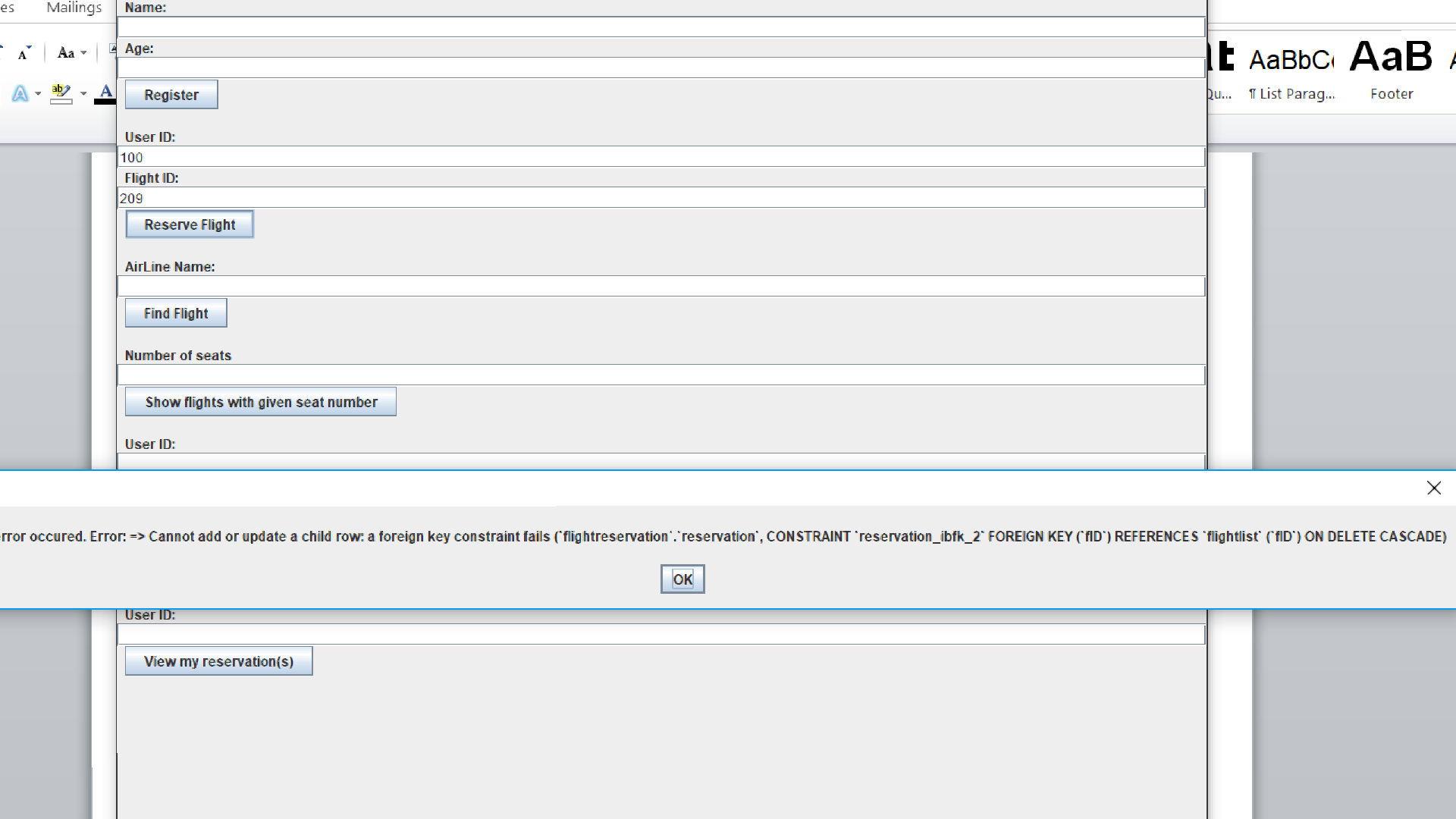
Reservation After ArchiveArchive

Key Constraints & Foreign Key Constraint

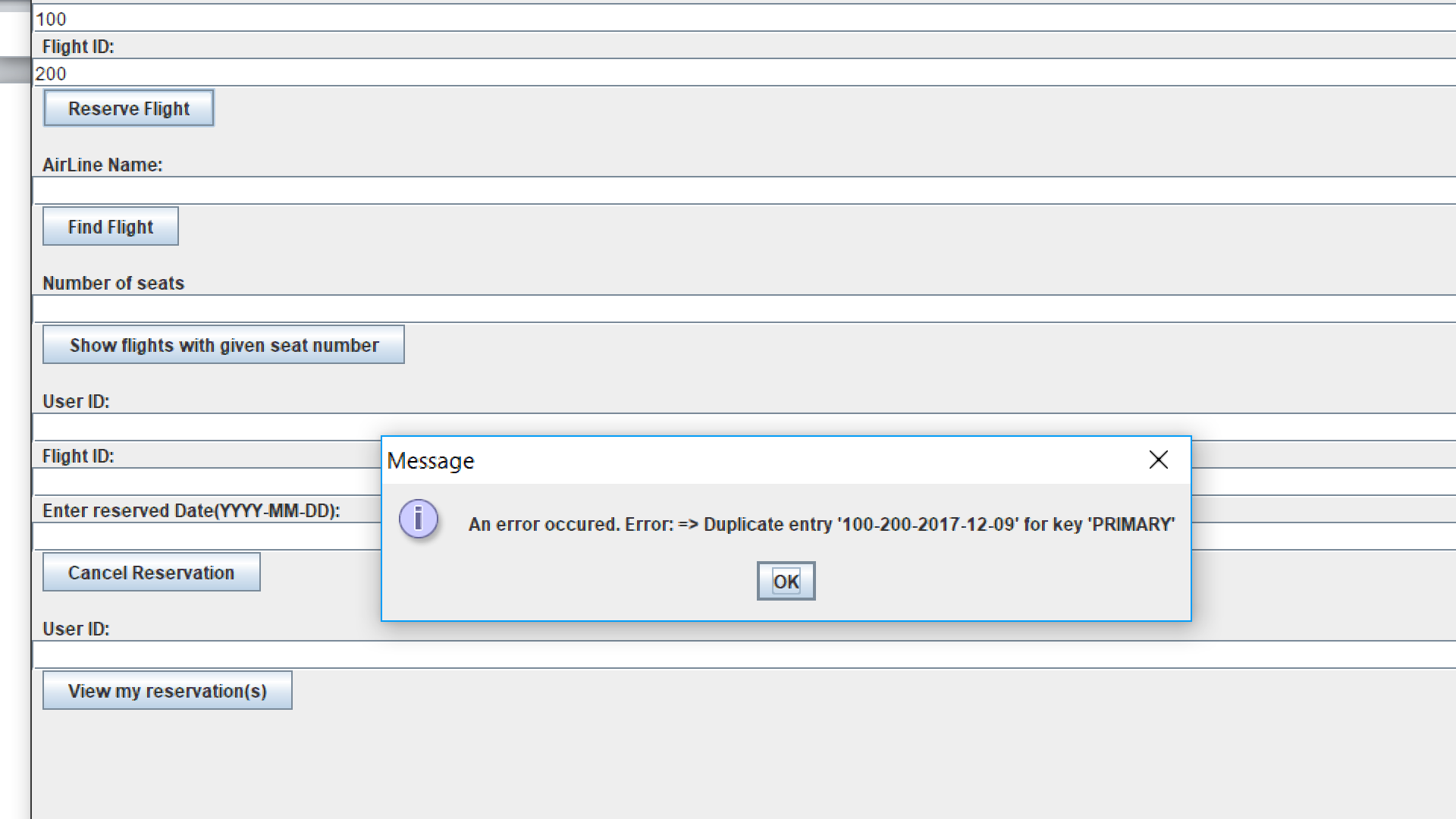
Foreign Key Violation

Reservation Using a Non-Existent Flight ID



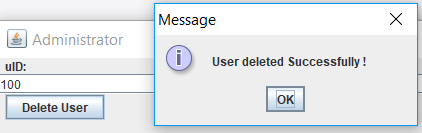
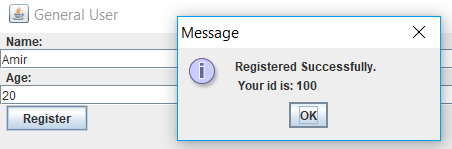
Key Constraint Violation

Same Day Reservation Using Same User ID & Flight ID

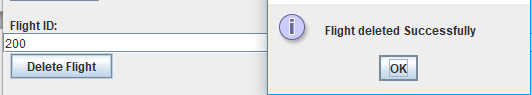


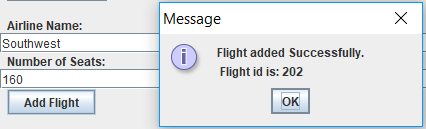
15 Functionality Screenshots

**Adding User Delete User**

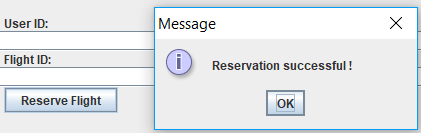
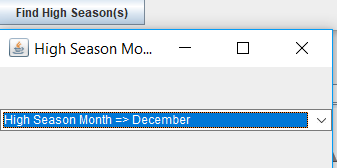


**Adding Flight Delete Flight**

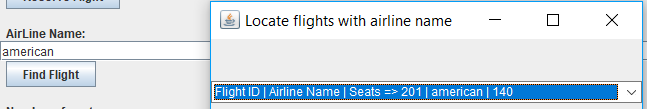




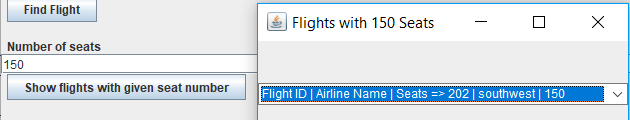
**Flight Reservation High Season**

** **

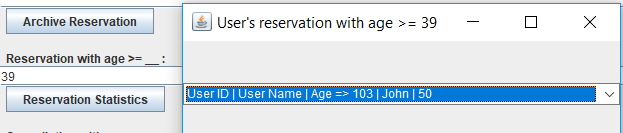
**Airline Name Search**



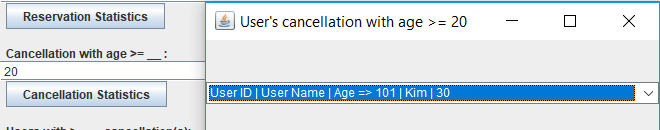
**Airline Search by Seat #**



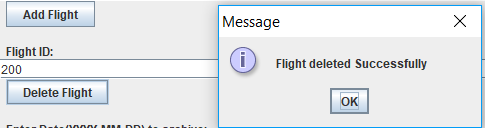
**User Reservations with Age >= \_\_**



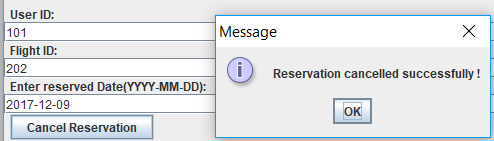
**User Cancellation with Age >= \_\_**



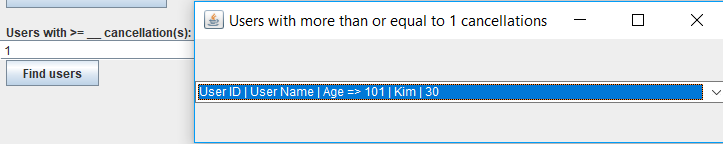
**Delete Flight**



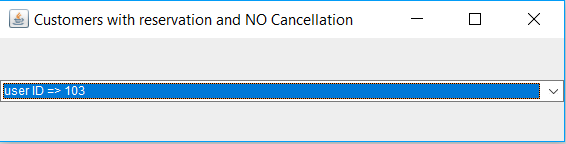
**Flight Cancellation**



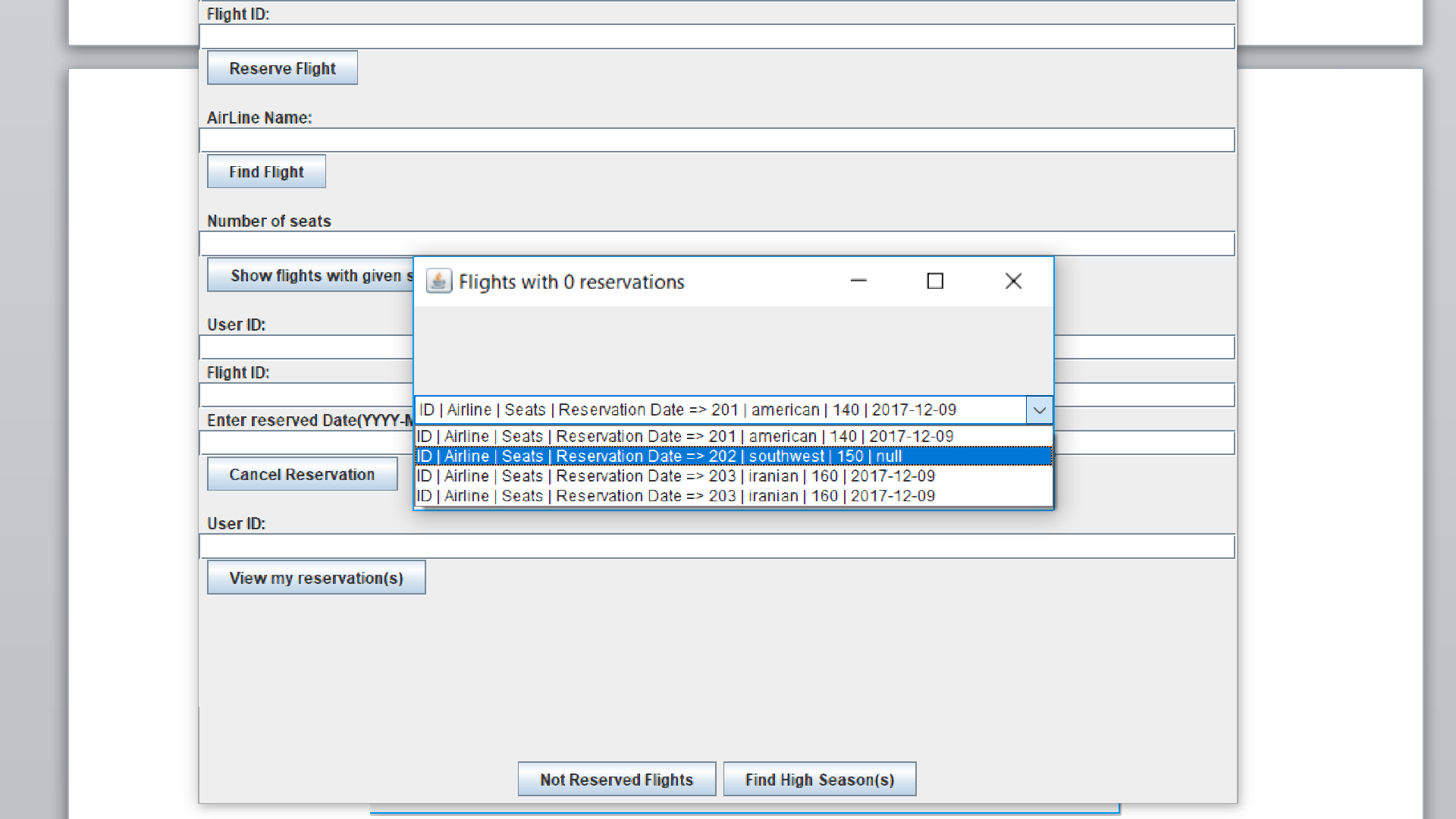
**Users with >= \_\_ Cancellations**



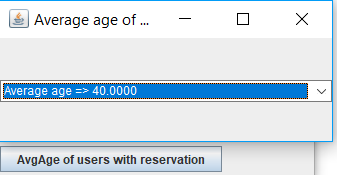
**Loyal Customers**



**Not Reserved Flights (Outer Join)**



**Average Age of Users with Reservation**



END OF REPORT