Home Page Use Cases

* View public info
  + SELECT \* FROM flight WHERE (departure\_airport = %s OR arrival\_airport = %s OR CAST(departure\_date AS DATE) = %s OR CAST(arrival\_date AS DATE) = %s) AND departure\_date > CURRENT\_DATE
    - Gets future flight information based on a search for an airport or a date
* Register
  + SELECT \* FROM customer WHERE email = %s
  + INSERT INTO customer VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    - Checks if a user with that email already exists, if not it inserts the new user into the database
  + SELECT \* FROM staff WHERE username = %s
  + INSERT INTO staff VALUES(%s, %s, %s, %s, %s, %s, %s)
    - Checks if a staff member with that email already exists, if not it inserts the new user into the database
* Login
  + SELECT \* FROM customer WHERE email = %s AND password = %s
    - Makes sure a user exists with the info provided
  + SELECT \* FROM staff WHERE username = %s AND password = %s
    - Makes sure a staff member exists with the info provided

Customer Use Cases

* View my flights
  + SELECT \* FROM ticket WHERE %s = customer\_email AND departure\_date > CURRENT\_DATE
    - Gets future flight information for a customer
* Search for flights
  + SELECT \* FROM flight WHERE (departure\_airport = %s OR arrival\_airport = %s OR CAST(departure\_date AS DATE) = %s OR CAST(arrival\_date AS DATE) = %s) AND departure\_date > CURRENT\_DATE
    - Gets search information for future flights based on date or airport
* Purchase tickets
  + SELECT \* FROM flight WHERE %s = airline\_name AND %s = departure\_date AND %s = number
    - Checks that the flight does exist
  + SELECT id FROM ticket WHERE %s = airline\_name AND %s = departure\_date AND %s = flight\_number
    - Checks if a ticket id value is already in the system, if it is then it gets reassigned and the query reran
  + INSERT INTO ticket VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    - If not, inserts new ticket data into database
* Cancel trip
  + DELETE FROM ticket WHERE id = %s AND departure\_date > (SELECT ADDDATE(CURRENT\_DATE, 1) AS DateAdd)
    - Removes the ticket from the users data as well as the database
* Give ratings and comment on previous flights
  + SELECT airline\_name, departure\_date, flight\_number, class, rating, comment FROM ticket NATURAL LEFT OUTER JOIN rating WHERE %s = customer\_email AND departure\_date < CURRENT\_DATE
    - Gets all the flight info from a flight that a user has taken in the past
  + INSERT INTO rating VALUES(%s, %s, %s, %s, %s, CURRENT\_TIMESTAMP)
    - Adds comment and rating to the selected flight
* Track my spending
  + SELECT SUM(sold\_price) as spent FROM ticket WHERE customer\_email = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 365) as DATESUB) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1) as DATEADD)
    - Gets spent money from the past year of tickets for the user
* Logout
  + No sql, pops session and logs user out

Staff Use Cases

* View flights
  + SELECT \* FROM flight WHERE airline\_name = %s AND departure\_date > CURRENT\_DATE AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 30) AS DateAdd)
    - Gets staff flights for the upcoming month
* Create new flights
  + SELECT \* FROM flight WHERE departure\_date = %s AND airline\_name = %s AND number = %s
    - Checks if a flight that is being created already exists
  + INSERT INTO flight VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    - If not, creates a new flight in the database
* Change status of flights
  + SELECT \* FROM flight WHERE departure\_date = %s AND airline\_name = %s AND number = %s
    - Checks that the flight exists in the system
  + UPDATE flight SET status = %s WHERE departure\_date = %s AND airline\_name = %s AND number = %s
    - If it does updates the status
* Add airplane in the system
  + SELECT airline\_name, id FROM airplane WHERE airline\_name = %s AND id = %s
    - Checks if an airplane is already in the system
  + INSERT INTO airplane VALUES(%s, %s, %s, %s, %s)
    - If it does not it is created
* Add new airport in the system
  + SELECT code FROM airport WHERE code = %s
    - Checks if an airport already exists in the system
  + INSERT INTO airport VALUES(%s, %s, %s, %s, %s)
    - If it does not it is created
* View flight ratings
  + SELECT DISTINCT AVG(rating) AS rating, departure\_date, flight\_number FROM rating WHERE airline\_name = %s GROUP BY departure\_date, flight\_number
    - Gets the average rating for each individual flight from the staff members airline
  + SELECT rating, comment, departure\_date, flight\_number FROM rating WHERE airline\_name = %s
    - Gets all the comments and ratings from the flights on the airline
* View frequent customers
  + SELECT COUNT(customer\_email) AS count, customer\_email FROM ticket WHERE airline\_name = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 365)) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1)) GROUP BY customer\_email
    - Gets the most frequents customers on the airline within the last year
  + SELECT DISTINCT \* FROM ticket WHERE airline\_name = %s
    - Gets a list of all flights the customers have taken on the staff members airline
* View earned revenue
  + SELECT SUM(sold\_price) as revenue FROM ticket WHERE airline\_name = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 31) as DATESUB) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1) as DATEADD)
    - Gets the total revenue for the airline from the past year
  + SELECT SUM(sold\_price) as revenue FROM ticket WHERE airline\_name = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 365) as DATESUB) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1) as DATEADD)
    - Gets the total revenue for the airline from the past month
  + SELECT SUM(sold\_price) as revenue, class FROM ticket WHERE airline\_name = %s AND departure\_date <= CURRENT\_DATE GROUP BY class
    - Gets the total revenue split by flight class
* View top destinations
  + SELECT COUNT(arrival\_airport) as top, arrival\_airport FROM ticket NATURAL JOIN flight WHERE airline\_name = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 365) as DATESUB) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1) as DATEADD) GROUP BY arrival\_airport ORDER BY top DESC
    - Gets the yearly top three destinations that were traveled to
  + SELECT COUNT(arrival\_airport) as top, arrival\_airport FROM ticket NATURAL JOIN flight WHERE airline\_name = %s AND departure\_date > (SELECT SUBDATE(departure\_date, 90) as DATESUB) AND departure\_date < (SELECT ADDDATE(CURRENT\_DATE, 1) as DATEADD) GROUP BY arrival\_airport ORDER BY top DESC
    - Gets the three month top three destinations that were traveled to
* Logout
  + No sql, pops session and logs user out