Course Project
Phase 3
Philadelphia, PA to Cleveland, OH  Fri Mar 19 2010 – Mon Mar 22 2010

Promo for Cleveland Flights. Today Only! Tickets from $49.
FareSpotter.net/Cleveland Flights

358 of 358 roundtrips shown show matrix

<table>
<thead>
<tr>
<th>Price*</th>
<th>Airlines</th>
<th>Takeoff</th>
<th>Landing</th>
<th>Stops</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$294</td>
<td>Continental</td>
<td>PHL: 1:15p</td>
<td>CLE: 7:05p</td>
<td>0 (1h 25m)</td>
<td></td>
</tr>
<tr>
<td>$294</td>
<td>Continental</td>
<td>PHL: 1:15p</td>
<td>CLE: 12:05p</td>
<td>0 (1h 25m)</td>
<td></td>
</tr>
<tr>
<td>$294</td>
<td>Continental</td>
<td>PHL: 1:15p</td>
<td>CLE: 11:20a</td>
<td>0 (1h 25m)</td>
<td></td>
</tr>
</tbody>
</table>

Select

© Ben Carterette
Phase 3

• Complete application-level programming
• A user should be able to:
  – Create an account
  – Store information in the account (credit cards, travelers)
  – Search for trips on a certain date
  – Purchase tickets/fares

Searching for Flights

• All flights in the data are one-way and direct
• Users should be able to see trips that consist of multiple direct flights with layovers
  – E.g. someone wants to go from PHL to ABQ, but there’s no direct flight. There may be a flight from PHL to IAD, then IAD to ABQ, though.
  – For simplicity, price of trip = sum of price of individual fares
    • Price of round trip = sum of prices of all fares
• Users should be able to indicate preferences as noted in the requirements
  – Airport type, cabin, one-way, etc
Interface

• There should be an interface for users to interact with the database
• It does not need to be fancy—a simple command-line interface will do
• Users should not be inputting SQL—the program should take user input and translate it into the necessary SQL queries

Interface Suggestions

• Three simple approaches:
  1. Allow users to input commands such as “create account [name] [password]”, “search flights [origin] [destination] on [date]”; parse and process
  2. Ask users a series of questions to determine what they want to do
  3. Use command-line flags to pass information to the program
Save State

- This is a database application, so it should not rely on information being available in memory
- If the user quits the program, or the program crashes, all information should still be accessible on restart
  - This includes account information, ticket purchases, etc