Καρατζάς Δημήτρης **icsd13072**
Λάζαρος Απόστολος **icsd13096**

2η ΟΜΑΔΙΚΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΥΛΟΠΟΙΗΣΗ

Α) ΒΑΣΙΚΕΣ ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΟΦΑΣΕΙΣ

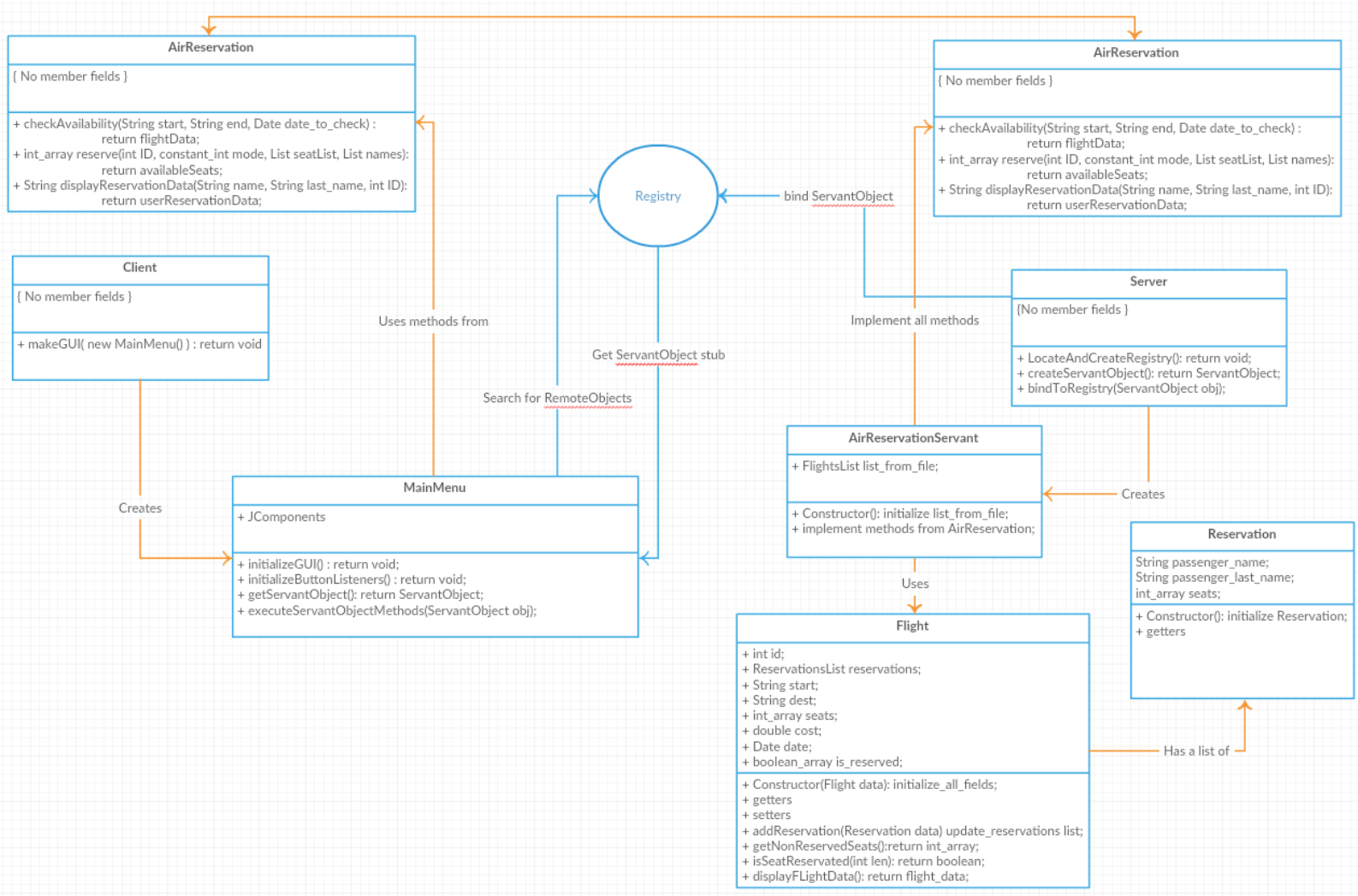
Την εφαρμογή την υλοποιήσαμε χρησιμοποιόντας το πρόγραμμα NetBeans. Για την υλοποίηση της, έχουμε δημιουργήσει 2 projects, ένα για τον **client** και ένα για τον **server**. Ο server και ο client επικοινωνούν μέσω **Java RMI.** Συγκεκριμένα ο Client πρέπει να επικοινωνήσει στη διεύθυνση **localhost/ReservationService** για να μπορέσει να έχει πρόσβαση στις υλοποιημένες μεθόδους της απομακρυσμένης διεπαφής. Αυτό σημαίνει πως ο Server έχει εγγράψει στο μητρώο του (ξεκινάμε την υπηρεσία rmiregistry προγραμματιστικά και όχι από το cmd αν και δεν υπάρχει διαφορά) ένα αντικείμενο υλοποίησης του interface.
Επίσης σχετικά με τα αρχεία, στον φάκελο αρχείων .java του Server, χρησιμοποιήσαμε το εργαλείο **javac** για να παραχθούν τα **.class** αρχεία και να δουλέψει η εφαρμογή. Στον Client προσθέσαμε μόνο το .java αρχείο του interface (όχι της κλάσης υλοποίησης του). Δηλαδή η εφαρμογή παράγει *δυναμικά* τα απαραίτητα **stub** και **skeleton** αρχεία, οπότε δε χρειάστηκε να χρησιμοποιήσουμε το εργαλείο **rmic.**

Β) RMI INTERFACE

Το interface είναι πολύ απλό, ορίζει τρείς ενέργειες, όσες και οι ενέργειες που ζητούνται να υλοποιηθούν, δηλαδή:

- Έλεγχος διαθεσιμότητας συγκεκριμένης πτήσης
- Εισαγωγή κράτησης σε μια πτήση
- Εμφάνιση στοιχείων για μια κράτηση

Ο server αρχικά ξεκινάει το registry καθώς επίσης ξεκινά μια RMI υπηρεσία και της δίνει και ένα όνομα (στη συγκεκριμένη άσκηση ReservationService). Στη συνέχεια δημιουργεί ένα αντικείμενο υπηρέτης, το οποίο υλοποιεί τις παραπάνω μεθόδους. Τέλος το αντικείμενο αυτό το εγγράφει στο μητρώο (Bind) ώστε να μπορεί κάποιος Client να χρησιμοποιήσει τις μεθόδους του.
Παρακάτω φαίνεται το UML διάγραμμα (θέλει λίγο zoom για να φανεί καλά). Στην αριστερή μεριά είναι οι τρεις κλάσεις του Client, και δεξιά οι πέντε κλάσεις του Server. Η Client κλάση απλώς δημιουργεί ένα αντικείμενο MainMenu το οποίο είναι και όλος ο Client ουσιαστικά, δηλαδή ψάχνει στο μητρώο για Remote Objects και εκτελεί τις μεθόδους του καθώς επίσης δημιουργεί το GUI και τους listeners. Στη πλευρά του Server, ο Server απλώς δημιουργεί το αντικείμενο υπηρέτη και το εγγράφει στο μητρώο. Οπότε στον server όλη την δουλειά την κάνει αυτό το αντικείμενο, το οποίο υλοποιεί τις μεθόδους, το πως υλοποιούνται φαίνεται και στον κώδικα. Τέλος οι κλάσεις Flight/Reservation χρησιμοποιούνται από τον Server (υπηρέτη) για να ελέγξει τις πτήσεις και κρατήσεις για κάθε πτήση.



Γ) ΠΑΡΑΔΟΧΕΣ

Μια παραδοχή που έχουμε κάνει αφορά τον αριθμό των θέσεων, συγκεκριμένα έχουμε θέσει μια τιμή στις 250 θέσεις για κάθε πτήση. Το πιο λογικό θα ήταν να είχαμε μια παραπάνω κλάση για ένα αεροπλάνο και διάφορα αντικείμενα αεροπλάνου (με πιθανών διαφορετικό αριθμό θέσεων) οπότε θα συσχετίζαμε κάθε πτήση με κάποιο αεροπλάνο, αλλά υποθέσαμε πως κάτι τέτοιο είναι εκτός σκοπού της εφαρμογής (χρήση Java RMI).
Επίσης δεν θέσαμε κανέναν SecurityManager ώστε να τρέξει η εφαρμογή , αλλιώς έπρεπε να δημιουργήσουμε διάφορα policy files.
Τέλος, σχετικά με τα αρχεία, έτσι όπως είναι ορισμένος ο σερβερ δουλεύει μόνο με τις αρχικές τιμές, διότι όταν ενημερώνει κάποια πτήση (προσθέτει κρατήσεις) δεν γράφεται κάπου στο αρχείο με τις κρατήσεις, το λογικό θα ήταν μετά από ένα ορισμένο χρονικό διάστημα να κάνει overwrite το αρχείο και θα ξαναγράφει όλες τις πτήσεις ώστε να το ενημερώνει αλλά και εδώ θεωρήσαμε πως δεν είναι απαραίτητο για τη συγκεκριμένη άσκηση.

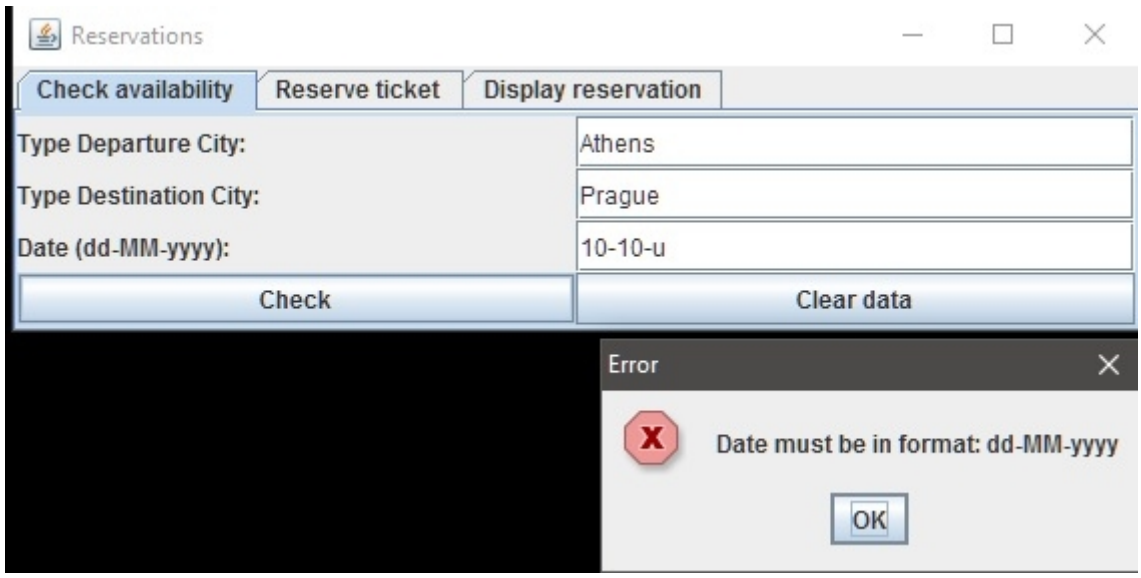
Δ) ΟΔΗΓΙΕΣ ΓΙΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

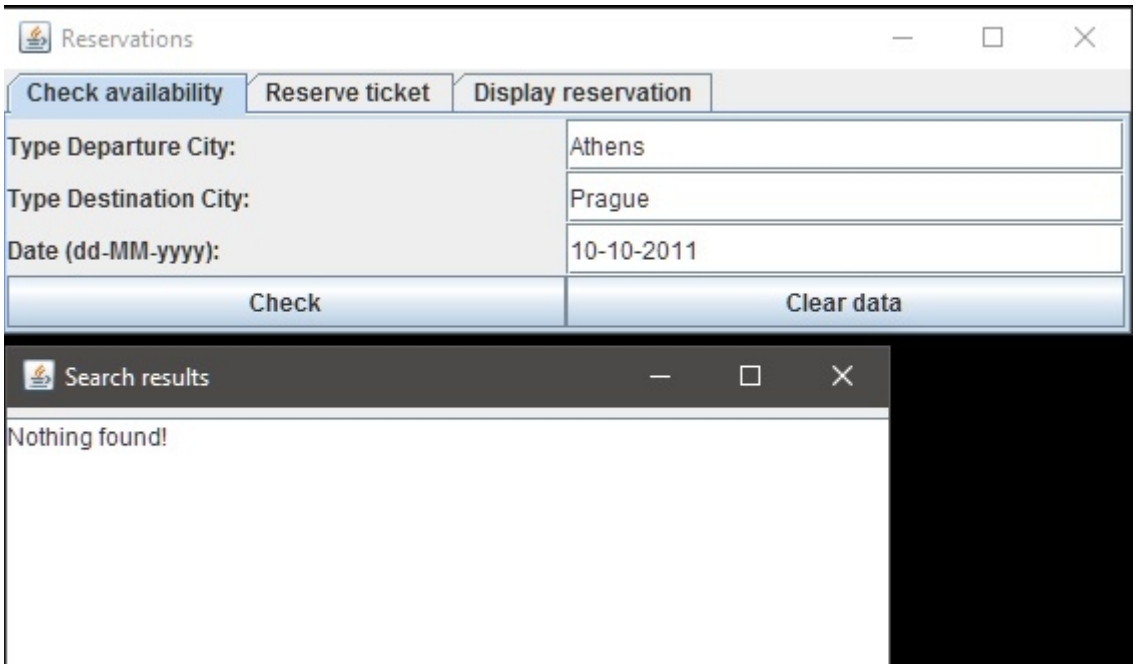
Server: Η εφαρμογή του server ξεκινάει και λειτουργεί αυτόματα, δεν χρειάζεται κάποια συγκεκριμένη διευκρίνιση.

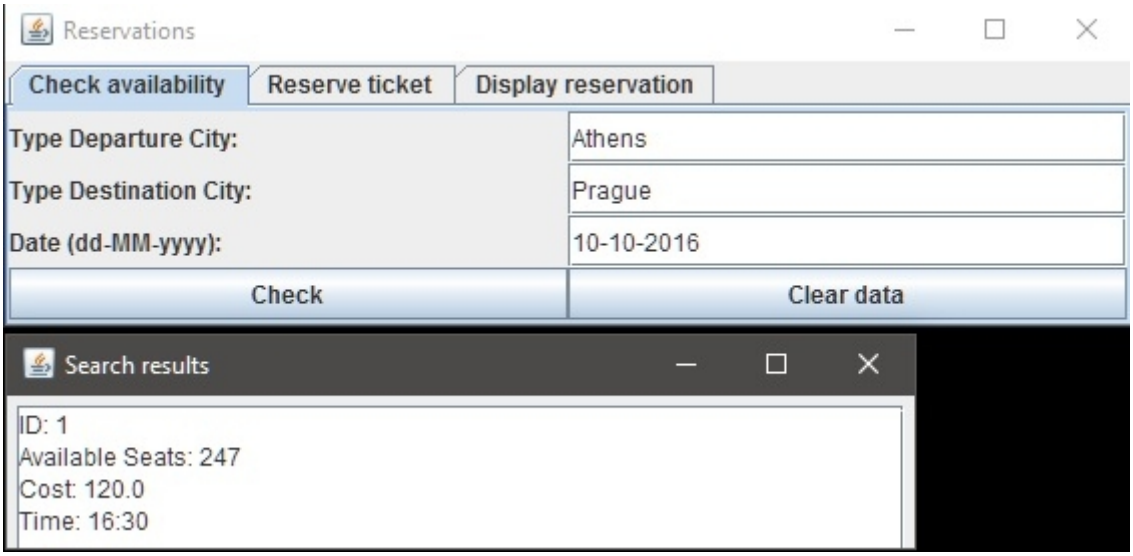
Client: Το GUI είναι αρκετά λιτό και είναι πολύ εύκολο κάποιος να χρησιμοποιήσει την εφαρμογή, συγκεκριμένα η εφαρμογή χωρίζεται σε 3 μέρη (tabs), το κάθε ένα εφαρμόζει μια ενέργεια της διεπαφής. Ο χρήστης εισάγει τα απαραίτητα στοιχεία στα text fields, αν κάνει κάποιο λάθος η εφαρμογή το εντοπίζει αυτόματα και στέλνει μήνυμα. Οδηγίες για την εκτέλεση των ενεργειών:

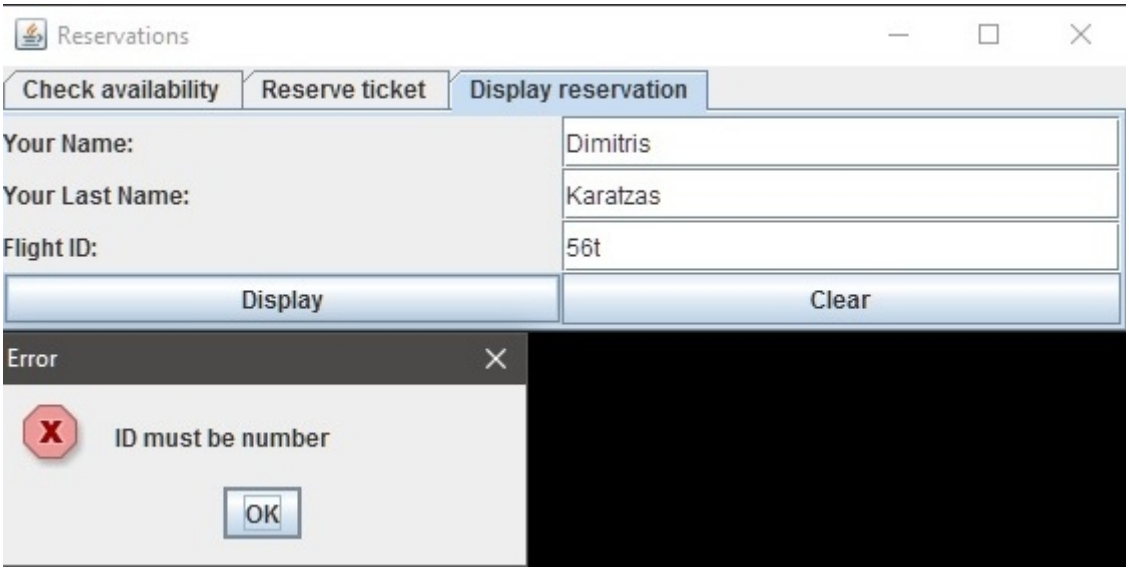
- 1ο tab: Έλεγχος διαθεσιμότητας πτήσης, ο χρήστης εισάγει το όνομα της πόλης προορισμού και αναχώρησης καθώς και την ημερομηνία της πτήσης και η εφαρμογή του επιστρέφει τις πτήσεις που πληρούν τα κριτήρια
- 2ο tab: Κράτηση, ο χρήστης εισάγει πρώτα τον κωδικό της πτήσης. Η εφαρμογή του επιστρέφει μια λίστα με διαθέσιμες θέσεις (δηλαδή δεν τις έχουν δεσμεύσει άλλοι πελάτες) για τη συγκεκριμένη πτήση, ο πελάτης επιλέγει τους αριθμούς και πατάει το κουμπί για συνέχεια, τώρα μπορεί να γράψει τα στοιχεία του (ονοματεπώνυμο) και να ολοκληρώσει τη διαδικασία. (Υπάρχει όριο 2 λεπτών στο να γράψει το ονοματεπώνυμο του)
- 3ο tab: Εμφάνιση στοιχείων κράτησης: Ο χρήστης απλώς γράφει το ονοματεπώνυμο του καθώς και τον κωδικό της πτήσης. Η εφαρμογή θα ψάξει για τις κρατήσεις που έχει κάνει ο συγκεκριμένος χρήστης για τη συγκεκριμένη πτήση και θα εμφανίσει τα στοιχεία της, αλλιώς δε θα εμφανίσει κάποια κράτηση.

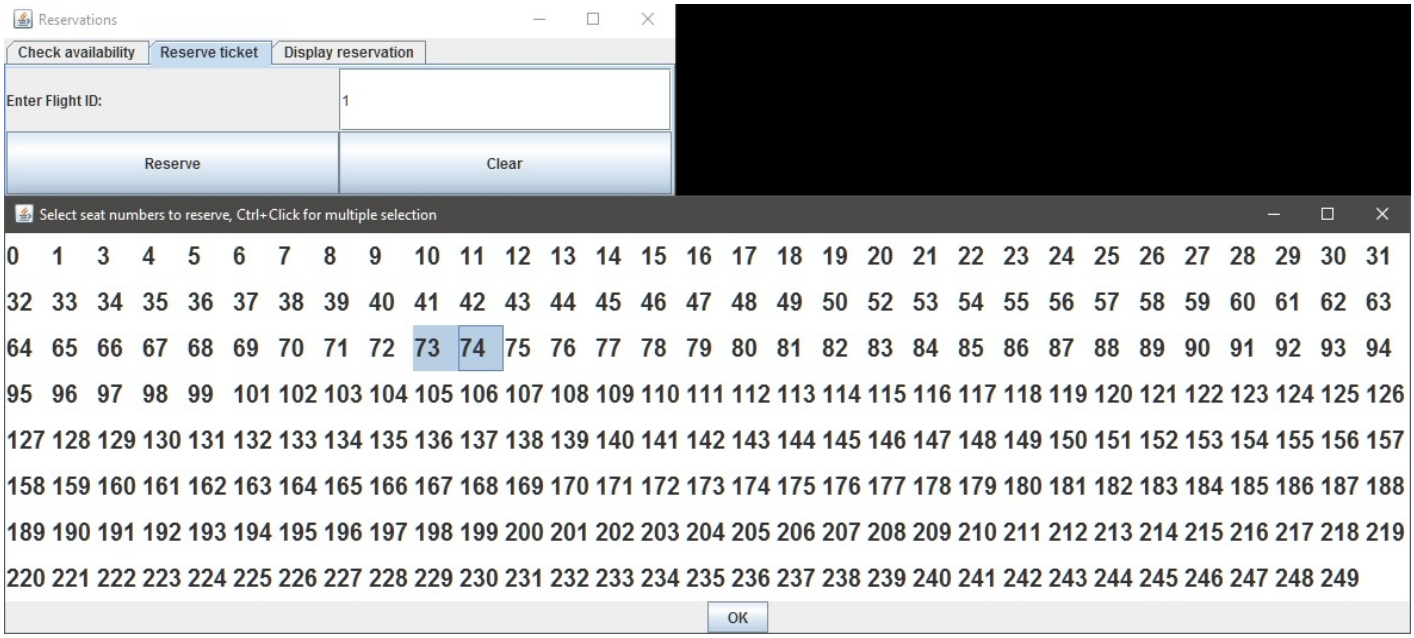
Ε) ΟΘΟΝΕΣ ΕΚΤΕΛΕΣΗΣ

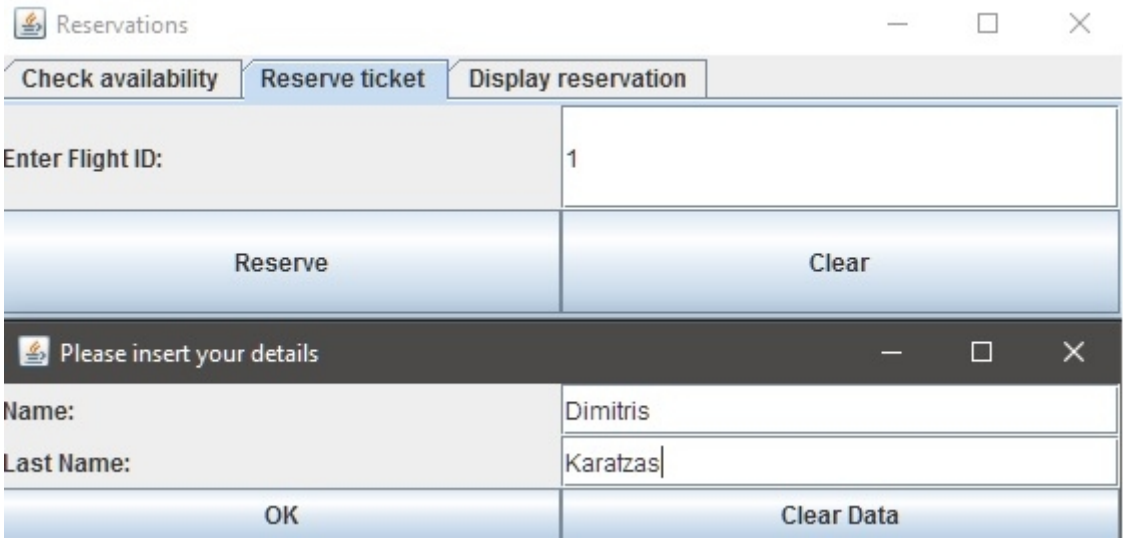


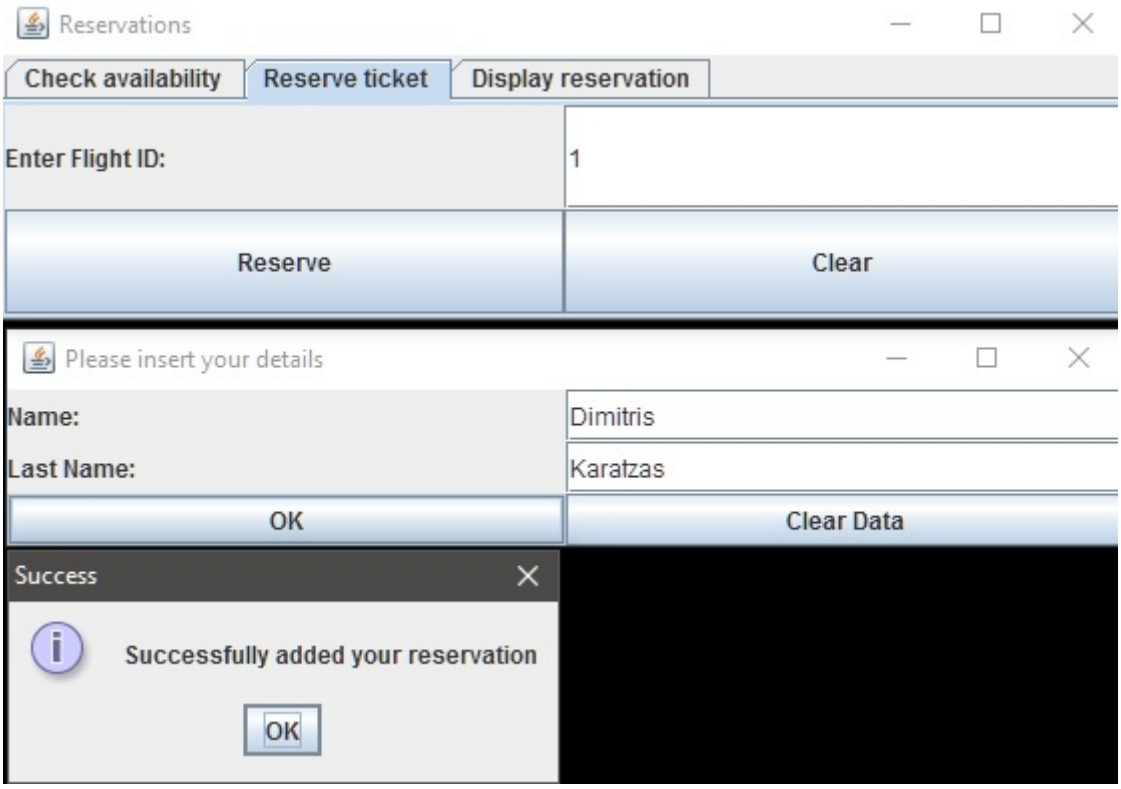

Reservations
Check availability
Reserve ticket
Display reservation
Type Departure City:
Athens
Type Destination City:
Prague
Date (dd-MM-yyyy):
10-10-2011
Check
Clear data
Search results
Nothing found!

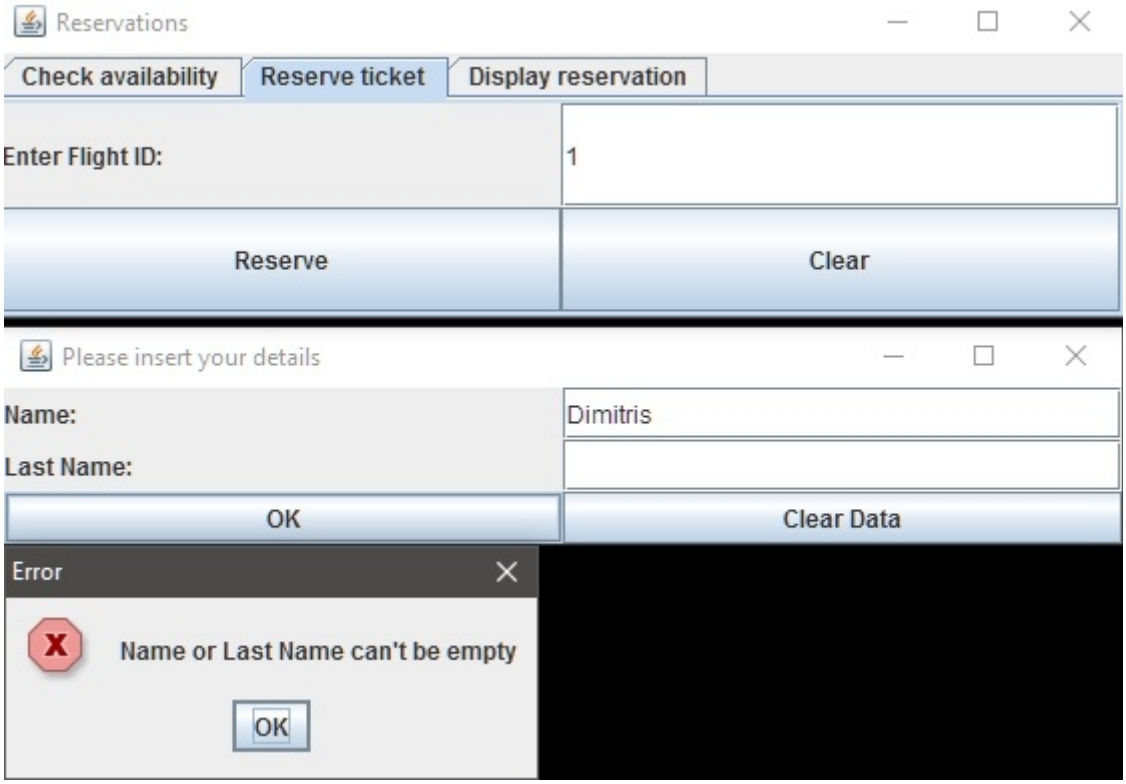

Reservations
Check availability
Reserve ticket
Display reservation
Type Departure City:
Athens
Type Destination City:
Prague
Date (dd-MM-yyyy):
10-10-2016
Check
Clear data
Search results
ID: 1
Available Seats: 247
Cost: 120.0
Time: 16:30

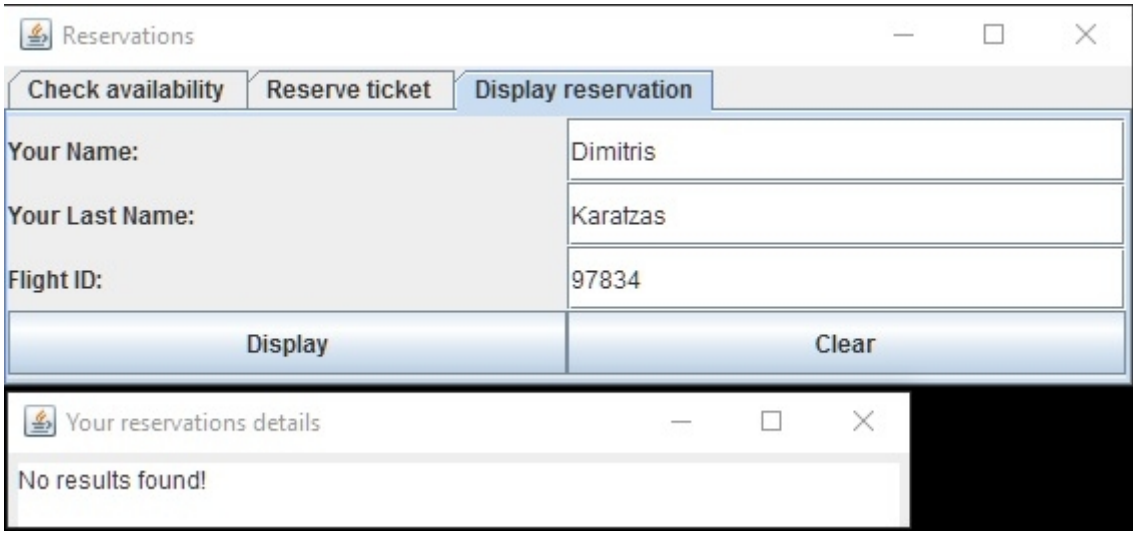

Reservations
Check availability
Reserve ticket
Display reservation
Your Name:
Dimitris
Your Last Name:
Karatzas
Flight ID:
56t
Display
Clear
Error
ID must be number
OK

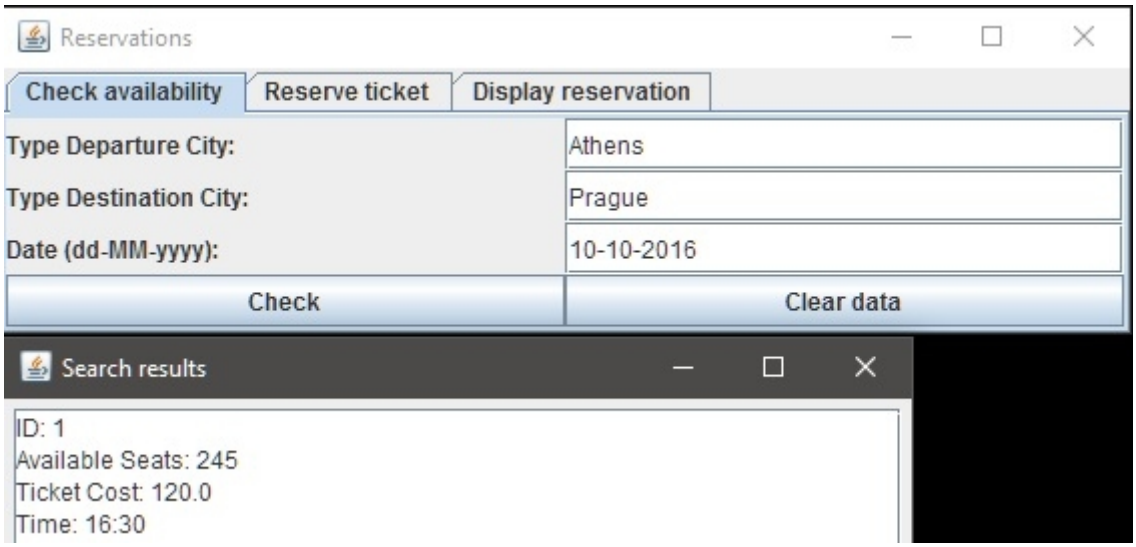

Reservations
Check availability
Reserve ticket
Display reservation
Enter Flight ID:
1
Reserve
Clear
Select seat numbers to reserve, Ctrl+Click for multiple selection
0 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
95 96 97 98 99 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188
189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219
220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249
OK

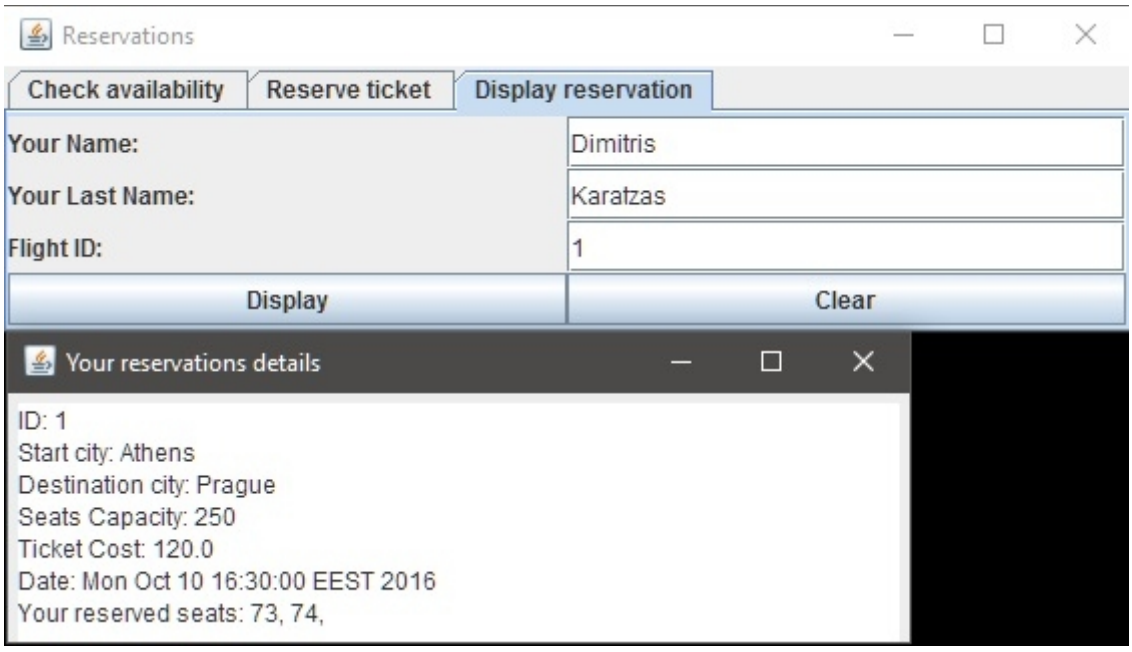

Reservations
Check availability
Reserve ticket
Display reservation
Enter Flight ID:
1
Reserve
Clear
Please insert your details
Name:
Dimitris
Last Name:
Karatzas
OK
Clear Data


Reservations
Check availability
Reserve ticket
Display reservation
Enter Flight ID:
1
Reserve
Clear
Please insert your details
Name:
Dimitris
Last Name:
Karatzas
OK
Clear Data
Success
Successfully added your reservation
OK


Reservations
Check availability
Reserve ticket
Display reservation
Enter Flight ID:
1
Reserve
Clear
Please insert your details
Name:
Dimitris
Last Name:
OK
Clear Data
Error
Name or Last Name can't be empty
OK


Reservations
Check availability
Reserve ticket
Display reservation
Your Name:
Dimitris
Your Last Name:
Karatzas
Flight ID:
97834
Display
Clear
Your reservations details
No results found!


Reservations
Check availability
Reserve ticket
Display reservation
Type Departure City:
Athens
Type Destination City:
Prague
Date (dd-MM-yyyy):
10-10-2016
Check
Clear data
Search results
ID: 1
Available Seats: 245
Ticket Cost: 120.0
Time: 16:30


Reservations
Check availability
Reserve ticket
Display reservation
Your Name:
Dimitris
Your Last Name:
Karatzas
Flight ID:
1
Display
Clear
Your reservations details
ID: 1
Start city: Athens
Destination city: Prague
Seats Capacity: 250
Ticket Cost: 120.0
Date: Mon Oct 10 16:30:00 EEST 2016
Your reserved seats: 73, 74,

Παρατηρούμε πως μετά την εισαγωγή της νέας κράτησης για τη συγκεκριμένη πτήση, οι διαθέσιμες θέσεις μειώθηκαν από 247 σε 245, που σημαίνει πως η κράτηση που προσθέσαμε πριν έχει μπεί στο σύστημα.

ΣΥΝΘΗΚΕΣ ΑΝΤΑΓΩΝΙΣΜΟΥ

Παραθέτουμε το κείμενο που είχαμε γράψει στη 1η ομαδική εργασία σχετικά με τον συγχρονισμό:
"Συνθήκες ανταγωνισμού εντοπίζονται μόνο όταν χρησιμοποιούμε νήματα. Συνθήκη ανταγωνισμού είναι όταν δύο ή παραπάνω νήματα προσπαθούν να γράψουν στο ίδιο στοιχείο ταυτόχρονα και έτσι έχουμε απρόσμενα αποτελέσματα, διότι το λειτουργικό σύστημα θα αποφασίσει τον χρονοπρογραμματισμό, οπότε δεν είναι προβλέψιμο το αποτέλεσμα. Για να το αντιμετωπίσουμε αυτό πρέπει να συγχρονίσουμε τα νήματα. Στη Java ο συγχρονισμός γίνεται με τη λέξη κλειδί synchronized. Όταν συγχρονίζουμε ένα κομμάτι κώδικα, αυτό σημαίνει πως κανένα άλλο νήμα της κλάσης αυτής δε μπορεί να χρησιμοποιήσει το τμήμα αυτό, παρα μόνο το νήμα που τρέχει τώρα (δηλαδή δεν επιτρέπουμε παραλληλία, το αντίθετο από αυτό που πετυχαίνουμε με νήματα, οπότε πρέπει να το εφαρμόζουμε μόνο εκεί που πρέπει). Έτσι, πρέπει να δηλώσουμε τα τμήματα κώδικα ως synchronized που υπάρχει πιθανότητα δυο νήματα να κάνουν ταυτόχρονα write σε έναν κοινό πόρο. "

Στη συγκεκριμένη εργασία συνθήκη ανταγωνισμού μπορεί να εντοπιστεί σε ένα κομμάτι μόνο, και αυτό είναι στη **δημιουργία κράτησης**, διότι όταν γίνεται μια κράτηση, ενημερώνεται η λίστα του σερβερ σχετικά με κάποια πτήση (μια πτήση κρατάει μια λίστα με τις κρατήσεις που ανήκουν σε αυτή). Οπότε είναι αναγκαία η χρήση του synchronized όταν γίνεται ενημέρωση της λίστας. Πέρα από αυτό, πάλι στην ίδια ενέργεια (δημιουργία κράτησης) ο server δεσμεύει για 2 λεπτά τις κρατήσεις που επέλεξε ο Client. Άρα δε μπορεί κάποιος άλλος (κάποιο άλλο νήμα) να εισάγει κράτηση στις ίδιες θέσεις οπότε και εδώ υπάρχει συνθήκη ανταγωνισμού.

Δεν υπάρχει κάποια άλλη συνθήκη ανταγωνισμού, διότι οι άλλες δυο ενέργειες απλώς **διαβάζουν** δεδομένα από τον σέρβερ και δε γράφουν τίποτα οπότε δε χρειάζεται συγχρονισμός.

ΕΞΩΤΕΡΙΚΕΣ ΠΗΓΕΣ

Σχετικά με τα .class αρχεία
https://docs.oracle.com/javase/tutorial/rmi/compiling.html

Σχετικά με το μητρώο
https://docs.oracle.com/javase/tutorial/rmi/implementing.html

Το διάγραμμα το δημιουργήσαμε στον ιστότοπο
https://creately.com/