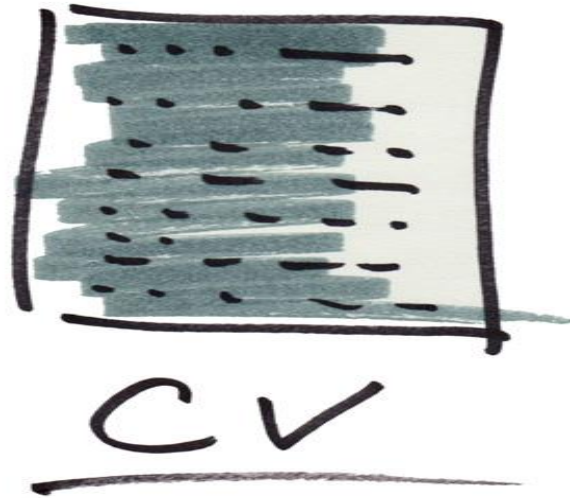


Ανάπτυξη συστήματος διαχείρισης Βιογραφικών υποψηφίων μελών ΕΠ

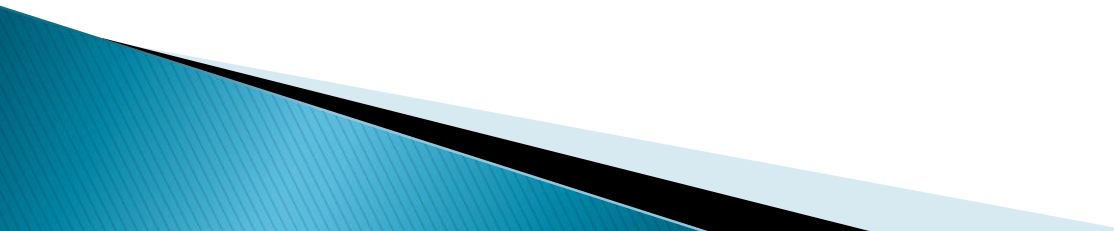


Μιχάλης Μασούρας
Επιβλέπων Καθηγητής: Ιγνάτιος Δεληγιάννης

Στόχος Πτυχιακής

- ▶ Η ανάπτυξη ενός αντικειμενοστρεφούς συστήματος λογισμικού, το οποίο θα διαχειρίζεται δεδομένα των βιογραφικών σημειωμάτων των υποψηφίων.

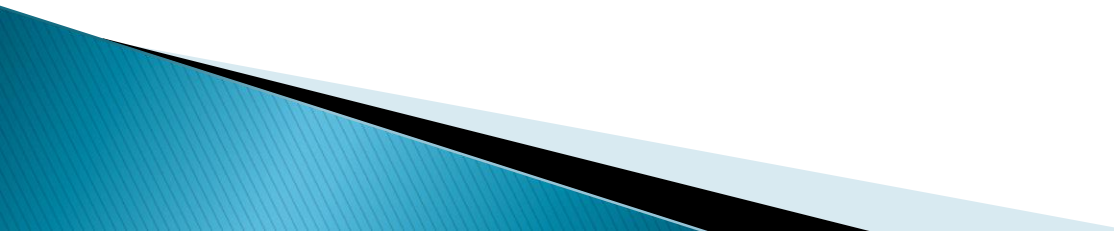
Κυρίως Σημεία

- Δυνατότητα αποθήκευσης και επεξεργασίας στοιχείων
 - Πολλαπλές επιλογές αναζήτησης και ταξινόμησης με διάφορα κριτήρια
 - Υπολογισμός προϋπηρεσίας
 - Δημιουργία και αποστολή κωδικού στο email ενός νέου χρήστη
- 

Τεχνολογίες

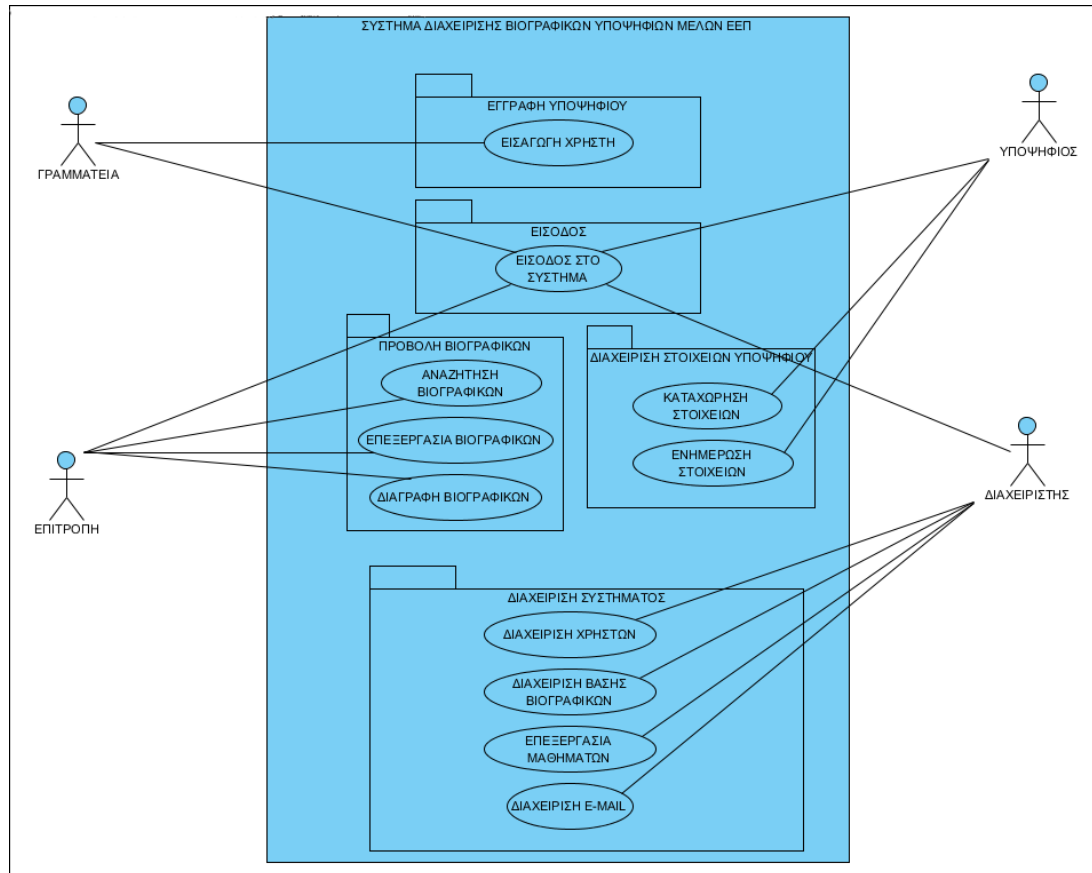
- ▶ UML
 - ▶ Java
 - ▶ Java Swing
 - ▶ JDBC
 - ▶ SQLite
 - ▶ JavaMail API
- 

Εργαλεία

- ▶ Visual Paradigm
 - ▶ NetBeans
 - ▶ SQLite Administrator
 - ▶ hMailServer
- 

ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ

Διάγραμμα περίπτωσης χρήσης: Δείχνει τους χρήστες και τη σχέση τους με την εφαρμογή.



Περιγραφή Χρηστών

- ▶ Υποψήφιος
 - Καταχώρηση, επεξεργασία και αποθήκευση προσωπικών βιογραφικών στοιχείων
- ▶ Γραμματεία
 - Εγγραφή υποψηφίων χρηστών
 - Επεξεργασία πίνακα ωρών διδασκαλίας και επιθυμητών μαθημάτων

Περιγραφή Χρηστών

- ▶ Επιτροπή
 - Εμφάνιση, επεξεργασία και αποθήκευση βιογραφικών υποψηφίων
 - Αναζήτηση, ταξινόμηση βιογραφικών υποψηφίων
 - Διαγραφή βιογραφικών

Περιγραφή Χρηστών

- ▶ Διαχειριστής
 - Προσθήκη ή διαγραφή χρηστών
 - Επεξεργασία πίνακα διαθέσιμων μαθημάτων
 - Επιλογές email SMTP server

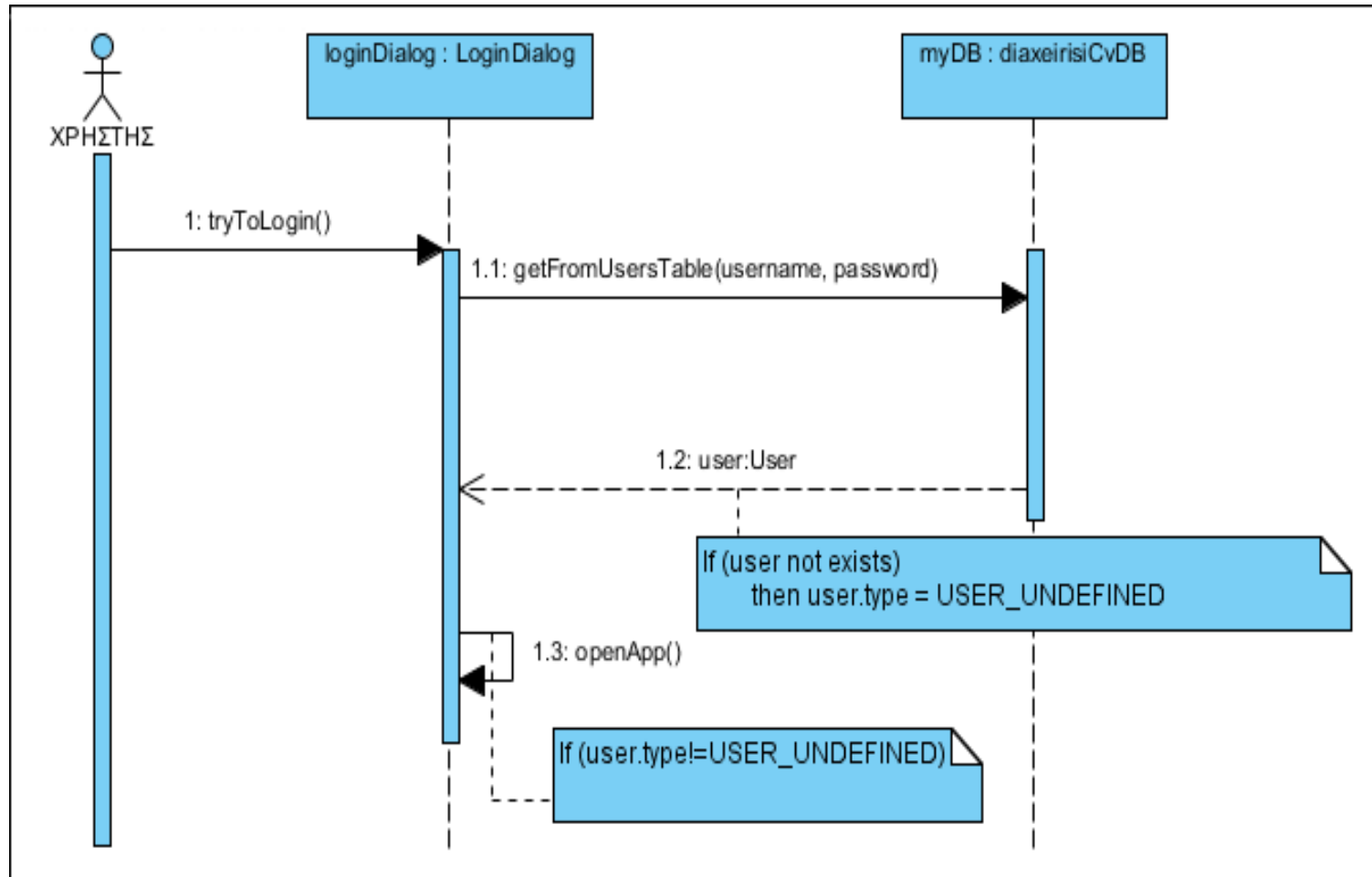
Η κλάση User

- ▶ Περιέχει πληροφορίες του χρήστη που προσπαθεί να μπει στο σύστημα μέσω ενός παραθύρου εισαγωγής στο σύστημα (LoginDialog).
 - String username
 - String password
 - int id
 - int type

Τύποι χρηστών

- ▶ Type definitions
 - USER_UNDEFINED = 0
 - USER_CANDIDATE = 1
 - USER_SECRETARY = 2
 - USER_COMMISSION = 3
 - USER_ADMINISTRATOR = 4

Σχεδιασμός-Διάγραμμα Ακολουθίας για την Π.Χ. Είσοδος στο σύστημα μέσω του LoginDialog



Περιγραφή βασικών παραθύρων

- ▶ LoginDialog : Dialog που περιμένει εισαγωγή username και password και αναλόγως με τον τύπο του χρήστη ανοίγει το αντίστοιχο Frame
- ▶ CvEditFrame : παράθυρο επεξεργασίας βιογραφικού υποψηφίου
- ▶ SecretaryFrame: παράθυρο γραμματείας
- ▶ CommisionFrame: παράθυρο επιτροπής
- ▶ AdministratorFrame: παράθυρο διαχειριστή

Παράθυρο Βιογραφικού

Διαχείριση Βιογραφικών Υποψηφίων Εκπαιδευτικών - Επεξεργασία Βιογραφικού

Προσωπικά Στοιχεία

Εκπαίδευση

Δημοσιεύσεις

Προμηθευτές

Υπόθεση Εταιρίας

Mike Mixos

Ημερομηνία Γέννησης: 23 Νοεμβρίου 1981
Διεύθυνση: Ελευθερίου 16 - Αρκαδία 53000
Πόλη: Θεσσαλονίκη
Τηλέφωνο: 2310300400
Email: mike@hotmail.com

Εκπαίδευση

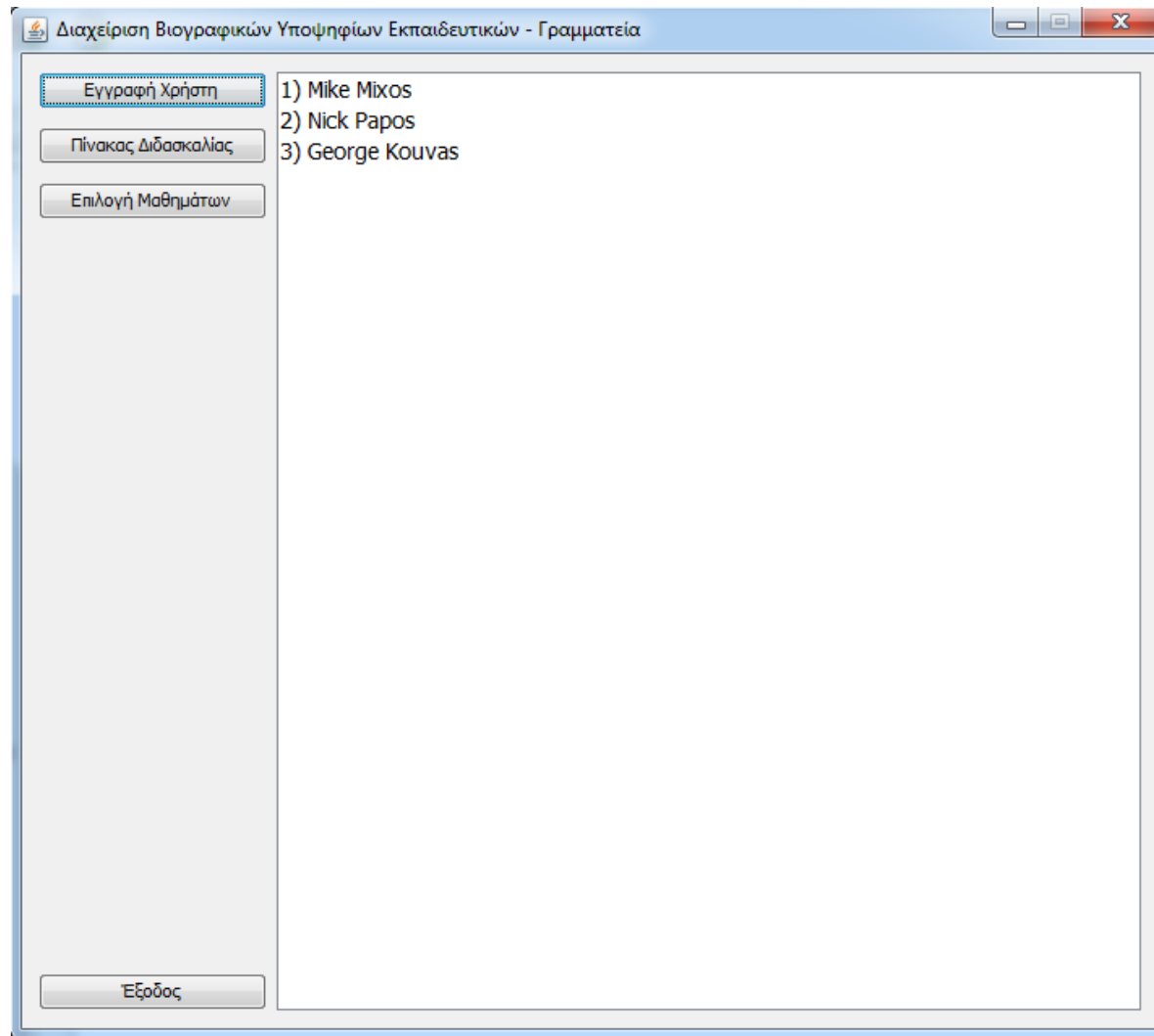
- *Ιανουάριος 1990 - Δεκέμβριος 1993*
 - Test123 University - BSc in Theoretical Theories
- *Σεπτέμβριος 1994 - Σεπτέμβριος 1996*
 - Theical University - MSc in Theical Theistry

Δημοσιεύσεις

- *Σεπτέμβριος 1999*
 - Data Reconstructor Technologies in Tabular Data Approximating Telematic Devices Error Logs - PHd Unreal University

Εξοδος

Παράθυρο Γραμματείας



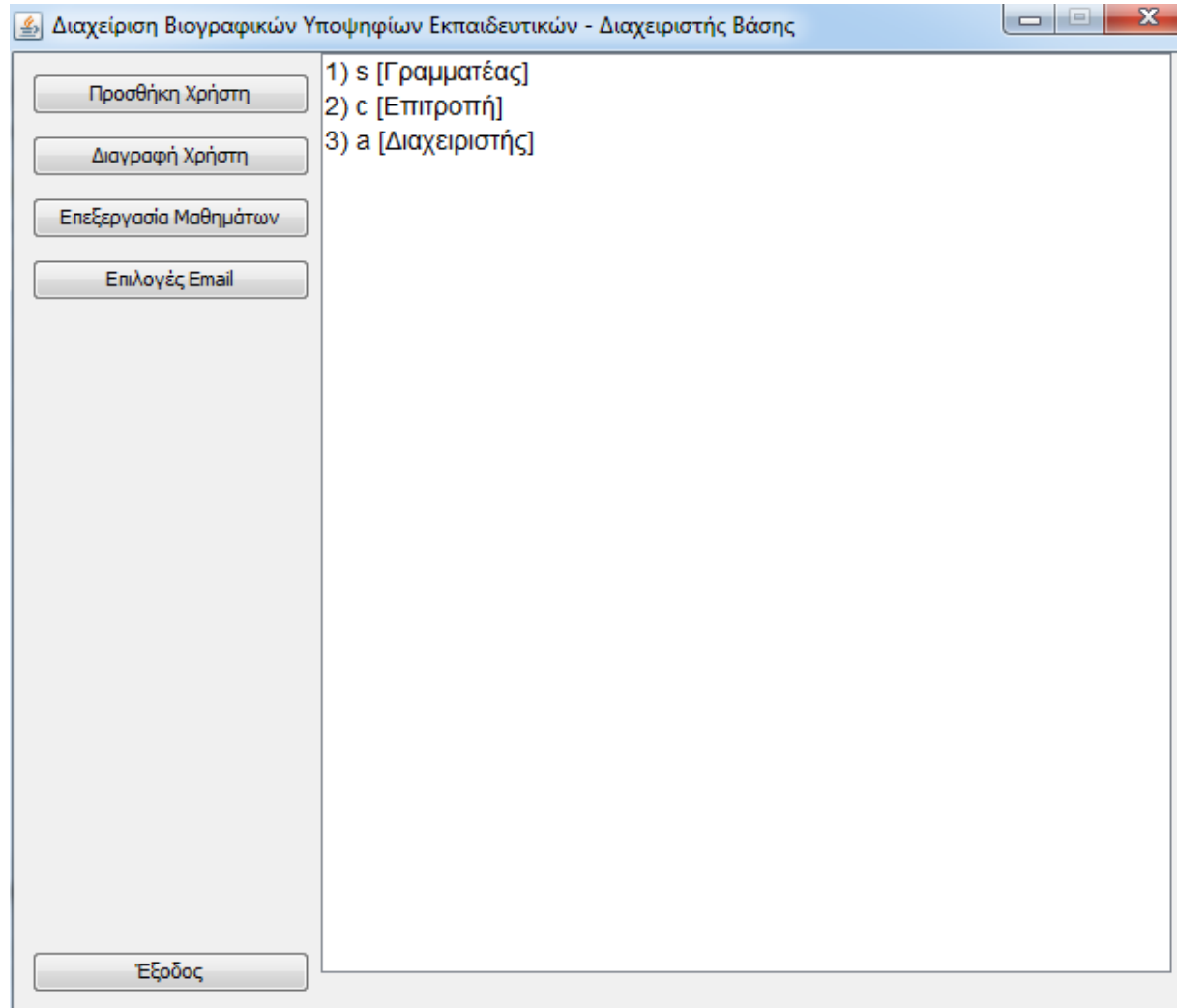
Παράθυρο Επιτροπής

Διαχείριση Βιογραφικών Υποψηφίων Εκπαιδευτικών - Επιτροπή

Επεξεργασία	Όνοματεπώνυμο	Ηλικία	Τόπος Διαμονής	Αριθμός Δημοσιεύσεων	Αριθμός Εταιριών	Μέρες Προϋπηρεσίας
Διαγραφή	Mike Mixos	29	Θεσσαλονίκη	1	1	827
Αναζήτηση	Nick Papos	28	Ιωάννινα	3	1	633
Εμφάνιση Όλων	George Kouvas	42	Αθήνα	2	1	305

Εξοδος

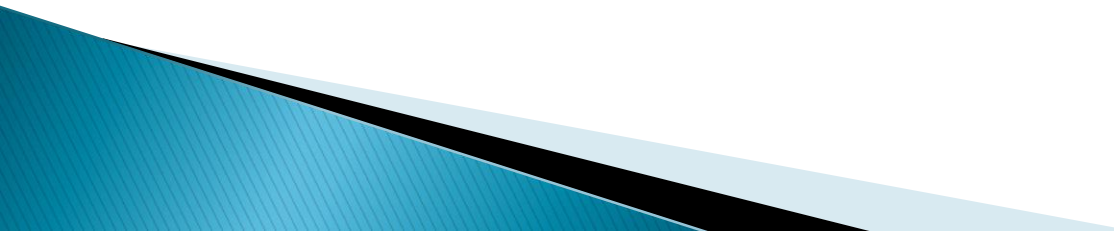
Παράθυρο Διαχειριστή



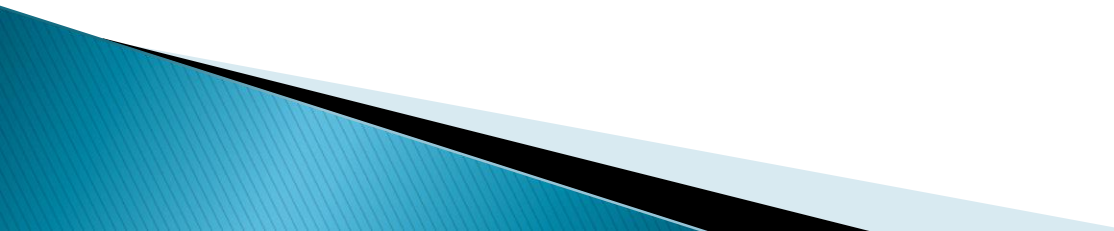
Η κλάση CV

- ▶ Περιέχει όλες τις πληροφορίες για την περιγραφή ενός βιογραφικού. Αποτελεί σύνθεση από άλλες κλάσεις.
 - int id
 - PersonalData personalData
 - ArrayList<EducationData> educationList
 - ArrayList<PublicationData> publicationList
 - ArrayList<WorkExperienceData> workExperienceList
 - ArrayList<CompanyData> companyList

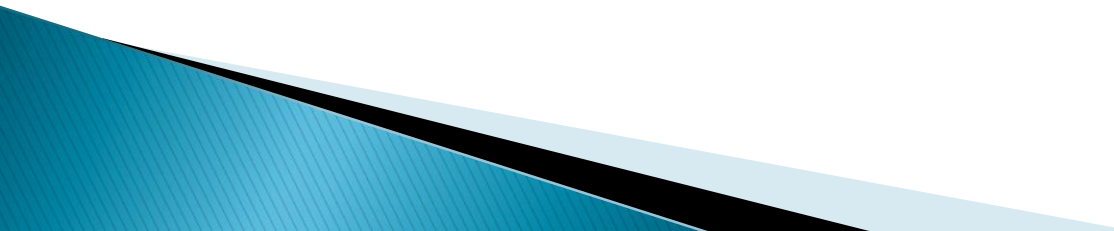
Η κλάση PersonalData

- ▶ String name
 - ▶ String surname
 - ▶ java.util.Date birthdate
 - ▶ String email
 - ▶ String address
 - ▶ String city
 - ▶ String phone
- 

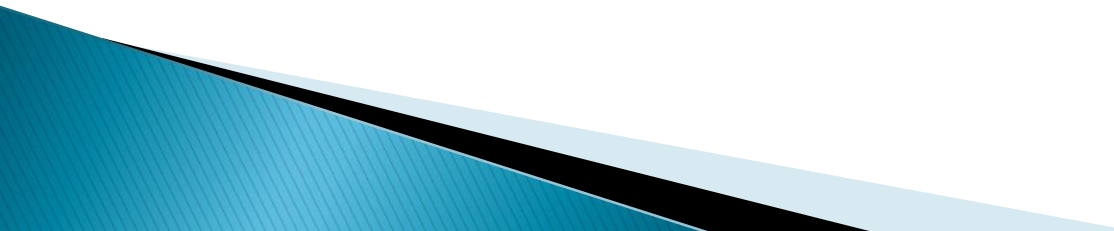
Η κλάση EducationData

- ▶ int id
 - ▶ java.util.Date dateFrom
 - ▶ java.util.Date dateTo
 - ▶ String description
- 

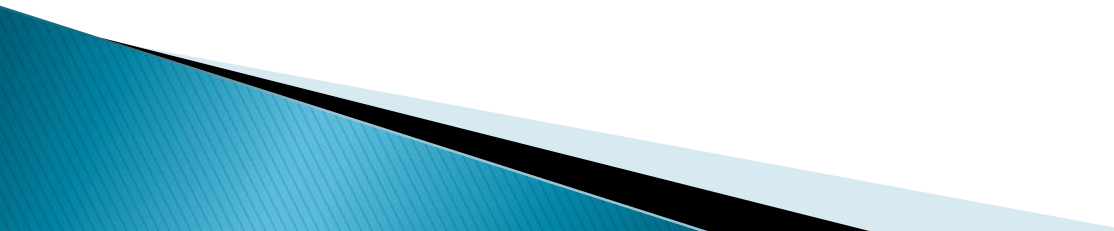
Η κλάση WorkExperienceData

- ▶ int id
 - ▶ java.util.Date dateFrom
 - ▶ java.util.Date dateTo
 - ▶ String description
- 

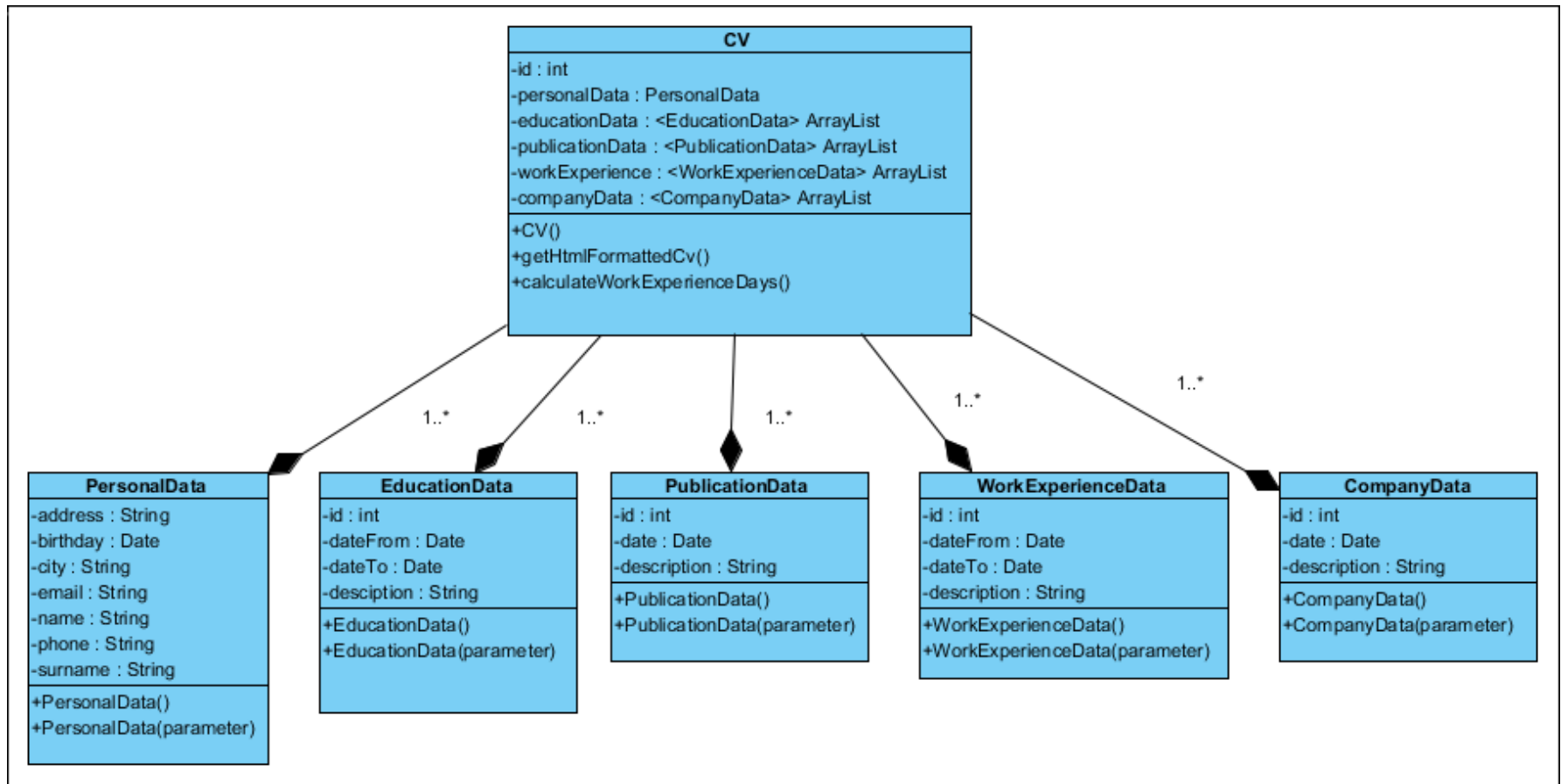
Η κλάση PublicationData

- ▶ int id
 - ▶ java.util.Date date
 - ▶ String description
- 

Η κλάση `CompanyData`

- ▶ `int id`
 - ▶ `java.util.Date date`
 - ▶ `String description`
- 

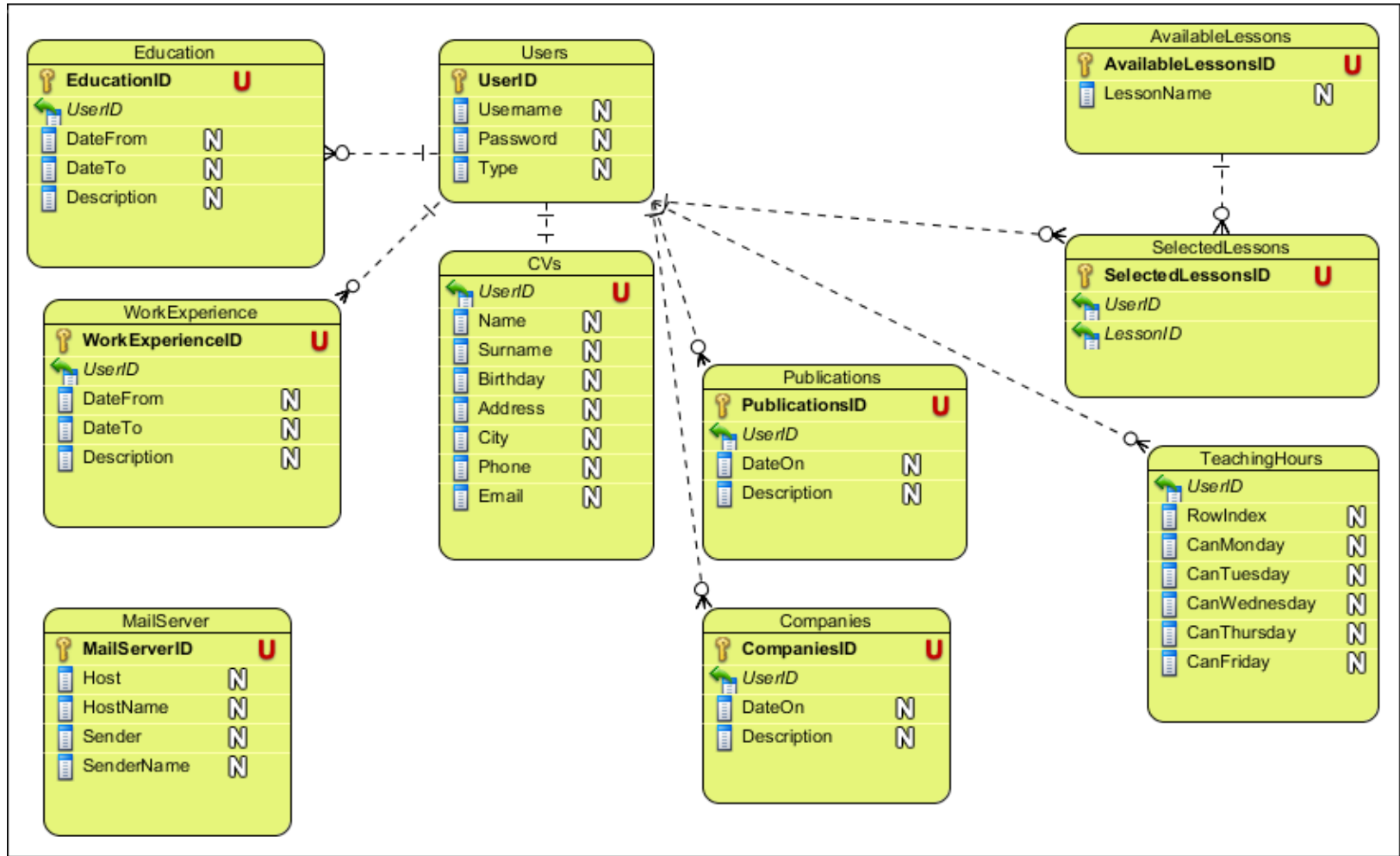
Διάγραμμα κλάσης CV



Η βάση δεδομένων

- ▶ Γενικώς χρησιμοποιώ την διεπαφή προγραμματισμού εφαρμογών JDBC για τη σύνδεση και την αποστολή ερωτημάτων στην SQLite.
- ▶ Την κατασκευή και αποστολή ερωτημάτων για τη συγκεκριμένη βάση την πραγματοποιεί η κλάση `DiacheirisiCvDB` η οποία κληρονομεί από μια abstract class, την `DBConnection`.
- ▶ Η `DBConnection` αναλαμβάνει διάφορες γενικές διεργασίες όπως την επιλογή του συγκεκριμένου DBMS (SQLite) κατά τη σύνδεση και ορισμένες εντολές για το κλείσιμο `Statement` ή `ResultSet` αντικειμένων.

Διάγραμμα σχεσιακής βάσης



Δημιουργία SQL query

- ▶ Για γενικότερη αναφορά στα ονόματα πινάκων και στηλών έχω προετοιμάσει κάποια Strings στα οποία αναφέρομαι καθώς κατασκευάζω τα SQL queries.
Π.χ. για τον πίνακα χρηστών:
 - `protected static final String USERS_TABLE_NAME = "Users";`
 - `protected static final String[] USERS_TABLE_COLUMN_NAMES = { "UserID", "Username", "Password", "Type" };`

Δημιουργία SQL query

Παράδειγμα:

```
String query = "CREATE TABLE " + USERS_TABLE_NAME + " ("
    + USERS_TABLE_COLUMN_NAMES[0] + " INTEGER
    PRIMARY KEY AUTOINCREMENT,"
    + USERS_TABLE_COLUMN_NAMES[1] + " VARCHAR (20),"
    + USERS_TABLE_COLUMN_NAMES[2] + " VARCHAR (20),"
    + USERS_TABLE_COLUMN_NAMES[3] + " INTEGER )";
```

Αποστολή SQL query μέσω JDBC

```
public boolean doSomethingToAtable(possible arguments)
{
    boolean success = true;
    String query = null;
    try
    {
        if (conn != null)
        {
            Statement stmt = conn.createStatement();
            query = ... // δημιουργούμε το query με τον παραπάνω τρόπο που περιγράψαμε
            stmt.executeUpdate(query);
            close(stmt);
        }
    }
    catch (SQLException e)
    {
        System.out.println("Problem in doing something to the " + OUR_TABLE_NAME + " table\n" + e.getMessage());
        if (!query.isEmpty()) System.out.println(query);
        success = false;
    }
    return success;
}
```

Επιστροφή αποτελεσμάτων με SQL query μέσω JDBC

```
User user = new User();  
int userId = .... // pirame to userId
```

```
try  
{  
    if (conn != null)  
    {  
        Statement stmt = conn.createStatement();  
        ResultSet rsIt = null;  
  
        query = "SELECT * FROM " + USERS_TABLE_NAME + " WHERE " + USERS_TABLE_COLUMN_NAMES[0] + "=" + userId;  
  
        rsIt = stmt.executeQuery(query);  
  
        while (rsIt.next()) {  
            user.id = rsIt.getInt(USERS_TABLE_COLUMN_NAMES[0]);  
            user.username = rsIt.getString(USERS_TABLE_COLUMN_NAMES[1]);  
            user.type = rsIt.getInt(USERS_TABLE_COLUMN_NAMES[3]);  
        }  
        close(stmt);  
        close(rsIt);  
    }  
}
```

Αναζήτηση βιογραφικών

- ▶ Περιλαμβάνει τα εξής βήματα
 - Ενημέρωση μιας ArrayList με το σύνολο των βιογραφικών από τη βάση (μια φορά).
 - Κάθε ερώτημα αναζήτησης διαγράφει από την ArrayList τα στοιχεία που δεν ταιριάζουν στα κριτήρια.
 - Ανανέωση του Graphical User Interface (GUI) με τα υπολειπόμενα αποτελέσματα στην ArrayList.

Ταξινόμηση

- ▶ Η ταξινόμηση ξεκινάει όταν ο χρήστης στο παράθυρο της επιτροπής πατήσει πάνω στο header μιας στήλης του πίνακα και εκτελεστεί το αντίστοιχο event.
- ▶ Από το event παίρνουμε τον αριθμό της στήλης και αναλόγως καλούμε την μέθοδο sort της java με στοιχεία από την ArrayList που θέλουμε να ταξινομήσουμε και ένα Comparator object όπου ορίζουμε τον τρόπο σύγκρισης μεταξύ δύο στοιχείων της λίστας.
- ▶ Ανανεώνουμε το GUI του πίνακα με την ανανεωμένη ArrayList.

Υπολογισμός προϋπηρεσίας

- ▶ Θεωρούμε πως μέρες προϋπηρεσίας είναι όλες οι μέρες της βδομάδας εκτός από Σάββατο και Κυριακή.
- ▶ Έχει δημιουργηθεί ειδική κλάση `DateTools` με διάφορα εργαλεία για τον υπολογισμό των εργασιμων ημερών από μια αρχική ημερομηνία έως μια τελική.
- ▶ Για κάθε ξεχωριστή δήλωση προϋπηρεσίας μέσα σε ένα βιογραφικό καλείται η μέθοδος `getWorkingDays(java.util.Date dateFrom, java.util.Date dateTo)` από την `DateTools` και επιστρέφεται το αποτέλεσμα το οποίο προστίθεται σε ένα συνολικό άθροισμα.

Υπολογισμός προϋπηρεσίας

- ▶ Επιλέχθηκε πιο αποδοτικός τρόπος για την εύρεση από ότι μια απλή ανάγνωση κάθε μέρας ξεχωριστά σε όλο το χρονικό διάστημα με την `Java.utils.Date.getDay()` .
- ▶ Μέσα στο διάστημα χρόνου που ζητούνται οι μέρες, ξεκαθαρίζουμε από πού ξεκινάνε και τελειώνουν οι πλήρεις βδομάδες και με απλό υπολογισμό πέντε εργασίμων ανά βδομάδα βρίσκουμε κομμάτι του αποτελέσματος χωρίς να κάνουμε μεγάλο βρόγχο.
- ▶ Συμπληρώνουμε και τις υπόλοιπες εργάσιμες μέρες από τα δύο κομμάτια μη ολοκληρωμένων βδομαδών στην αρχή και στο τέλος του χρονικού διαστήματος.

Υπολογισμός προϋπηρεσίας



Ημέρες προϋπηρεσίας: (πλήρεις εβδομάδες) * 5 εργάσιμες + ακρινές

Αποστολή του password με email

- ▶ Ο χρήστης δίνει το email του μαζί με τα υπόλοιπα στοιχεία στη γραμματεία και αυτή εισάγει τον χρήστη στο σύστημα.
- ▶ Εφόσον η εγγραφή έγινε επιτυχώς, ένα αυτοματοποιημένο σύστημα αποστολής email στέλνει τον δημιουργημένο κώδικο στην διεύθυνση του χρήστη.
- ▶ Για τον σκοπό αυτό χρησιμοποιούμε το JavaMail API.

Αποστολή του password με email

- ▶ Χρειάζονται τέσσερις πληροφορίες που τροφοδοτούνται στο API:
 - host
 - hostName
 - sender
 - senderName
- ▶ Δίνουμε τη δυνατότητα στον διαχειριστή της βάσης να αλλάξει αυτές τις τέσσερις πληροφορίες.
- ▶ Για τη δοκιμή του email API δημιουργήσαμε έναν local mail server με την εφαρμογή hMailServer.

ΣΥΜΠΕΡΑΣΜΑΤΑ & ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ

- ▶ Αρχικός Στόχος: η δημιουργία ενός συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ
- ▶ Λειτουργίες καταχώρησης και ενημέρωσης του βιογραφικού από τους υποψηφίους, προβολής και ταξινόμησης βιογραφικών από τη επιτροπή και δημιουργίας και αποστολής κωδικών από τη γραμματεία.
- ▶ Τελική Υλοποίηση: Υλοποιήθηκαν όλες οι λειτουργίες που τέθηκαν στον αρχικό στόχο της εφαρμογής, αφού έγινε αρχικά η ανάλυση και ο σχεδιασμός τους με αντικειμενοστραφείς μεθόδους (περιπτώσεις χρήσεις, διαγράμματα κλάσεων, διαγράμματα αλληλεπίδρασης κ.τ.λ.).
- ▶ Μελλοντικές επεκτάσεις και βελτιώσεις:
 - Η ανάρτηση του συστήματος σε ιστοσελίδα
 - Επιπλέον επιλογές αναζήτησης και ταξινόμησης κατά την προβολή των βιογραφικών, (π.χ. με βάση τις ώρες μη διαθεσιμότητας)

Ευχαριστώ για την προσοχή σας!

