



Ανάπτυξη Full-Stack Εφαρμογών με Java Swing

Θ. Ανδρούτσος



Αρχιτεκτονική τριών επιπέδων

Swing Desktop Full Stack Apps

- Μία Full Stack εφαρμογή, στο πλαίσιο του MVC, αποτελείται από τρία επίπεδα:
 - **View.** Γραφική Διεπαφή, Η γραφική διεπαφή στο MVC πρέπει να είναι passive (HTML, XML)
 - **Controller.** Ενδιάμεση οντότητα που διαχειρίζεται τα events και προωθεί κατάλληλα calls στο Model. Ουσιαστικά οι controllers είναι event handlers
 - **Model.** Οι κλάσεις του Συστήματος με δεδομένα και λειτουργίες. Το Model μπορεί να είναι μονολιθικό ή να διαχωριστεί περαιτέρω σε Layers όπως DAO, Service Layer



Swing MVP

- Το Swing δεν είναι MVC γιατί το View δεν είναι passive αλλά το γράφουμε με κώδικα Java
- Το μοντέλο εδώ είναι το MVP (Model View Presenter), δηλαδή View και Controller είναι από φυσική άποψη μαζί στην ίδια κλάση και υπάρχει μόνο λογικός διαχωρισμός



Αρχιτεκτονικές προσεγγίσεις

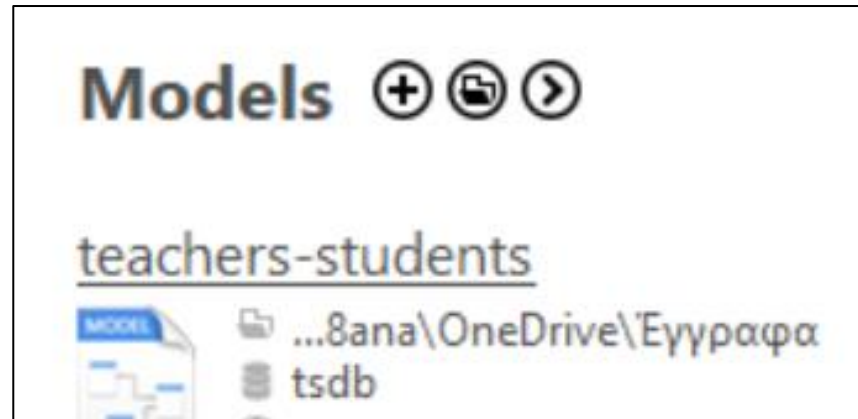
Swing Desktop Full Stack Apps

- Μπορούμε ωστόσο όσο αφορά το Model να έχουμε διάφορες προσεγγίσεις, μονολιθικές ή Layered Architectures:
 1. Monolithic: Όλος ο κώδικας, business methods + state management στο Model
 2. N-Tier Architectures – DAO, Service Layers και Model & DTO



Δημιουργία ΒΔ στην MySQL

Swing Desktop Full Stack Apps




- Θα δημιουργήσουμε μία ΒΔ, έστω **tsdb** με τρεις πίνακες: TEACHERS, STUDENTS και TEACHERS_STUDENTS
- Πρώτα δημιουργούμε νέο Model με το MySQL Workbench



Δημιουργία ΒΔ

tsdb - Schema

Name: Specify the name of

 Refactor model, changing all references

Charset/Collation: Default Collation: The character set and

Comments:

- Μέσα στο Model διπλό κλικ πάνω στο όνομα και δίνουμε νέο όνομα στη ΒΔ: το όνομα **tsdb** (ή ότι όνομα νομίζετε ότι αντιπροσωπεύει καλύτερα τη ΒΔ) και κωδικοποίηση χαρακτήρων UTF8mb4 ώστε να διαχειριζόμαστε όλα το φάσμα των Unicode χαρακτήρων



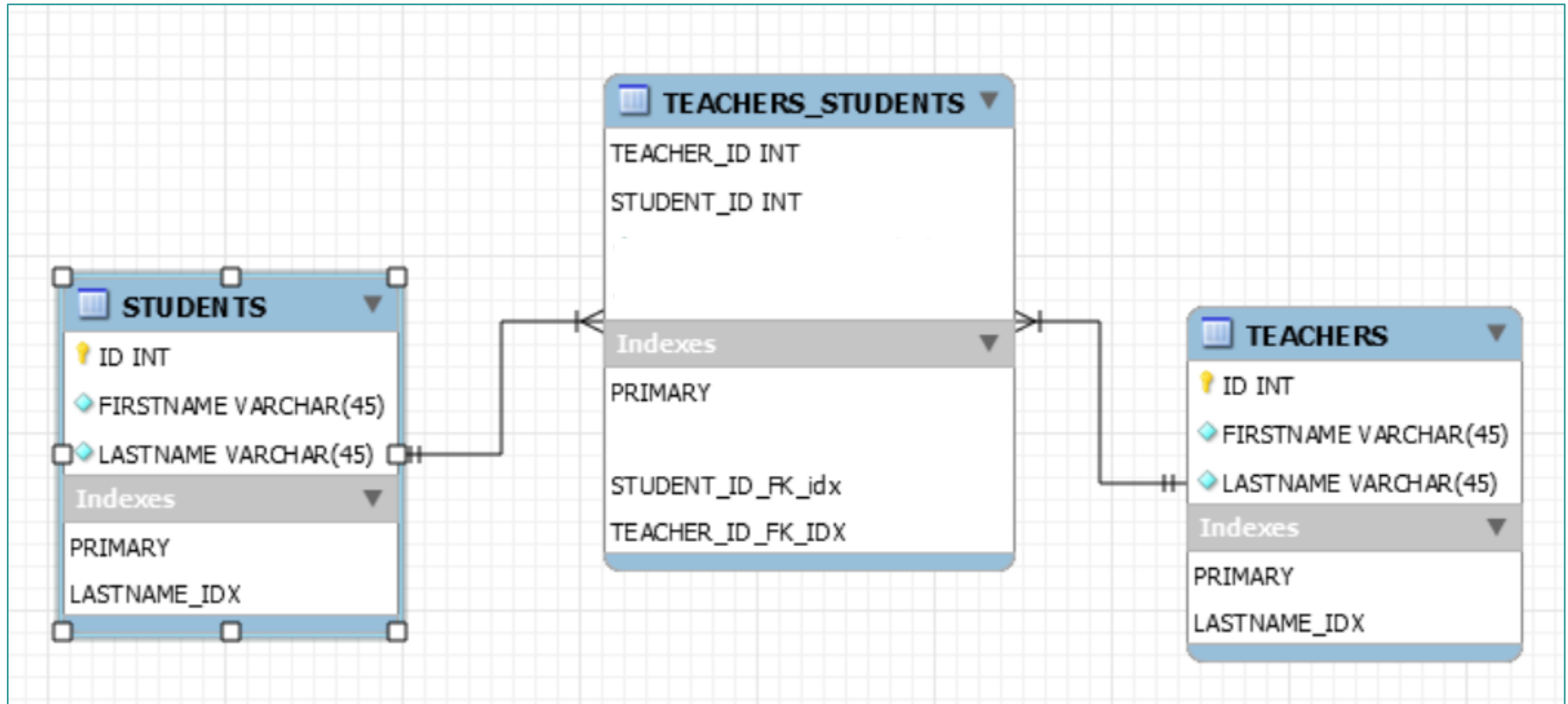
Charset / Collation

Swing Desktop Full Stack Apps

- Η MySQL δεν παρέχει τύπο `nvarchar`, αλλά μόνο `varchar`
- Αλλά εφόσον το charset είναι `utf8mb4` τότε όλα τα `character types` είναι UTF8 με 4 bytes, οπότε υποστηρίζουν το πλήρες σετ `codepoints` του UTF8



Σχήμα ΒΔ



- Παραπάνω είναι το E-R Model της Βάσης Δεδομένων



Πίνακας TEACHERS

Swing Desktop Full Stack Apps

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FIRSTNAME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LASTNAME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Δημιουργούμε ένα πίνακα από το **Add Table** και δίνουμε όνομα πίνακα **TEACHERS**
- Δίνουμε πεδία: **ID, FIRSTNAME, LASTNAME** καθώς και **τύπους δεδομένων** για κάθε πεδίο. Το range του int είναι από -2147483648 έως 2147483647 (4 bytes). Στα varchar το (45) είναι το max πλήθος των χαρακτήρων του πεδίου.
- Επίσης δίνουμε για κάθε πεδίο ιδιότητες. Για το ID δίνουμε PK (primary Key - πρωτεύον κλειδί) και NN (Not Null). Για τα υπόλοιπα αρκεί Not Null αν θεωρούμε ότι είναι υποχρεωτικά



Ευρετήριο

TEACHERS - Table x

Table Name: Schema: **tsdb**

Index Name	Type
PRIMARY	PRIMARY
LASTNAME_IDX	INDEX

Index Columns

Column	#	Order	Length
<input type="checkbox"/> ID		ASC	
<input type="checkbox"/> FIRSTNAME		ASC	
<input checked="" type="checkbox"/> LASTNAME	1	ASC	

Index Options

Storage Type:

Key Block Size:

Parser:

Visible: ☒

- Επειδή οι αναζητήσεις στον πίνακα **TEACHERS** θα γίνονται με το Επώνυμο του Καθηγητή, δημιουργούμε **ευρετήριο (index)** με βάση το αντίστοιχο πεδίο
- Αν δεν δώσουμε index και πάλι θα μπορούμε να κάνουμε αναζητήσεις με βάση το Επώνυμο, αλλά θα γίνονται αργά (το index δημιουργεί ένα b-tree με αποτέλεσμα οι αναζητήσεις με αυτό το πεδίο να γίνονται σε χρόνο $\log n$)



Πίνακας Students

Swing Desktop Full Stack Apps

STUDENTS - Table x

Table Name: STUDENTS Schema: tsdb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FIRSTNAME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LASTNAME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

STUDENTS - Table x

Table Name: STUDENTS Schema: tsdb

Index Name	Type
PRIMARY	PRIMARY
LASTNAME_IDX	INDEX

Index Columns

Column	#	Order	Length
<input type="checkbox"/> ID		ASC	
<input type="checkbox"/> FIRSTNAME		ASC	
<input checked="" type="checkbox"/> LASTNAME	1	ASC	

Index Options

Storage Type:

Key Block Size: 0

Parser:




Visible: ☒

- Το ίδιο όπως πριν, τώρα για τον πίνακα STUDENTS



Πίνακας TEACHERS_STUDENTS

Swing Desktop Full Stack Apps

TEACHERS_STUDENTS - Table ×										
		Table Name:	TEACHERS_STUDENTS						Schema:	tsdb
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	
 TEACHER_ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 STUDENT_ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Ξένα Κλειδιά

TEACHERS_STUDENTS - Table x

Table Name: Schema: **tsdb**

Foreign Key Name	Referenced Table
TEACHER_ID_FK	`tsdb`.`TEACHERS`
STUDENT_ID_FK	`tsdb`.`STUDENTS`

Column	Referenced Column
<input type="checkbox"/> TEACHER_ID	
<input checked="" type="checkbox"/> STUDENT_ID	ID
<input type="checkbox"/> CLASSROOM	
<input type="checkbox"/> MEETING_DATE	

Foreign Key Options

On Update:

On Delete:

☐ Skip in SQL generation

- Ο Πίνακας TEACHERS_STUDENTS είναι ενδιάμεσος



Ευρετήρια (για αναζητήσεις)

Swing Desktop Full Stack Apps

TEACHERS_STUDENTS - Table x

Table Name: Schema: **tsdb**

Index Name	Type
PRIMARY	PRIMARY
STUDENT_ID_FK_idx	INDEX
TEACHER_ID_FK_IDX	INDEX

Index Columns

Column	#	Order	Length
<input checked="" type="checkbox"/> TEACHER_ID	1	ASC	
<input type="checkbox"/> STUDENT_ID		ASC	
<input type="checkbox"/> CLASSROOM		ASC	
<input type="checkbox"/> MEETING_DATE		ASC	

Index Options

Storage Type:

Key Block Size:

Parser:


Visible: ☒

- Πάντα τα ευρετήρια δημιουργούνται και χρησιμοποιούνται για αναζητήσεις
- Όταν έχουμε πρωτεύοντα κλειδιά, ξένα κλειδιά για joins ή αναζητήσεις και πεδία που κάνουμε αναζητήσεις, τότε χρειαζόμαστε ευρετήρια



Χρήστης

thanos3 - User ×

 Name: The name of the user. It is recommended to use only alpha-characters. Spaces should be avoided and be replaced by _

Password: The password used for this user account. Make sure to use s to prevent brute force attacks.

Roles:

< >

- table.readonly
- table.insert
- table.modify
- routine.execute

- Δημιουργούμε και ένα χρήστη για να συνδεόμαστε με τη ΒΔ (ποτέ δεν συνδεόμαστε ως root, πάντα με χρήστη που έχει δικαιώματα owner)



Forward Engineering

Swing Desktop Full Stack Apps

This script will now be executed on the DB server to create your databases.
You may make changes before executing.

```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES';
6
7  -----
8  -- Schema tsdb
9  -----
10
11  -----
12  -- Schema tsdb
13  -----
14  CREATE SCHEMA IF NOT EXISTS `tsdb` DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci ;
15  USE `tsdb` ;
16
17  -----
18  -- Table `tsdb`.`TEACHERS`
19  -----
20  CREATE TABLE IF NOT EXISTS `tsdb`.`TEACHERS` (
21    `ID` INT NOT NULL AUTO_INCREMENT,
22    `FIRSTNAME` VARCHAR(45) NOT NULL,
23    `LASTNAME` VARCHAR(45) NOT NULL,
24    PRIMARY KEY (`ID`),
```

Save to File...

Copy to Clipboard

Back

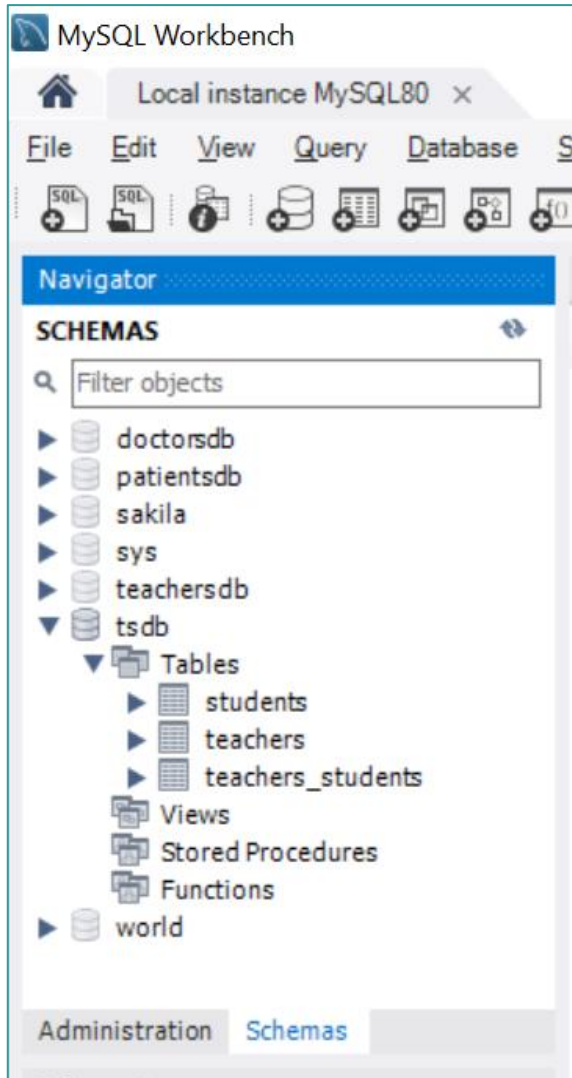
Next

Cancel

- Με Forward Engineering δημιουργούμε το SQL Script και το εκτελούμε ώστε να δημιουργηθεί η ΒΔ στην πλευρά του Server



MySQL Server



- Παρατηρούμε ότι δημιουργήθηκε η ΒΔ στον Server
- Μπορούμε πλέον να συνδεθούμε με οποιονδήποτε πρόσφορο τρόπο, όπως για παράδειγμα μέσω JDBC



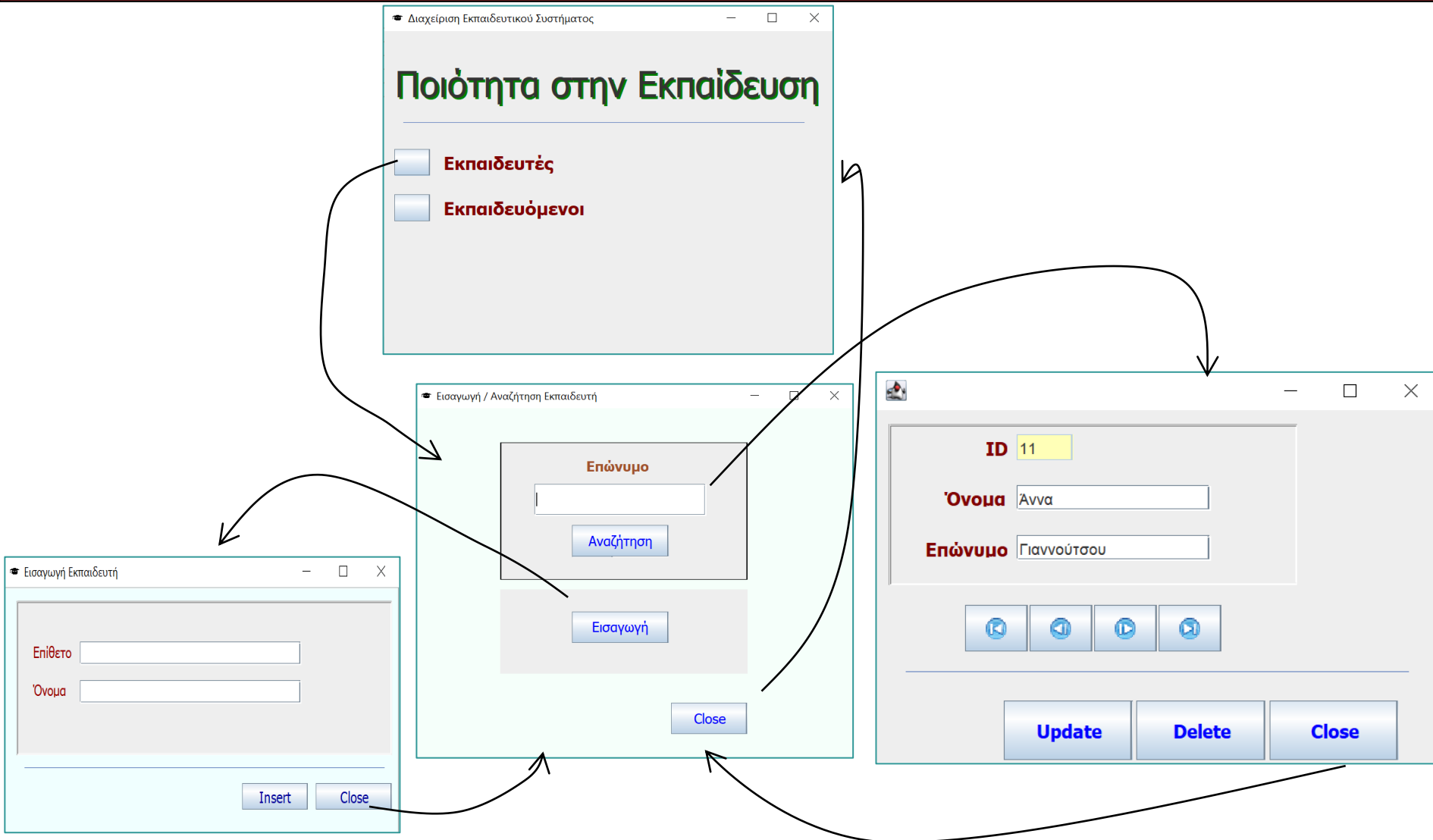
View

- Ένα από τα σημαντικότερα θέματα στην ανάπτυξη μιας εφαρμογής είναι ο σχεδιασμός και η ανάπτυξη του View, δηλαδή των **οθονών** που θα βλέπει ο χρήστης
- Μπορούμε στο Figma ή σε άλλα εργαλεία να σχεδιάσουμε τις **οθόνες** και το **navigational model**



Οθόνες/Mockups Εφαρμογής

Swing Desktop Full Stack Apps





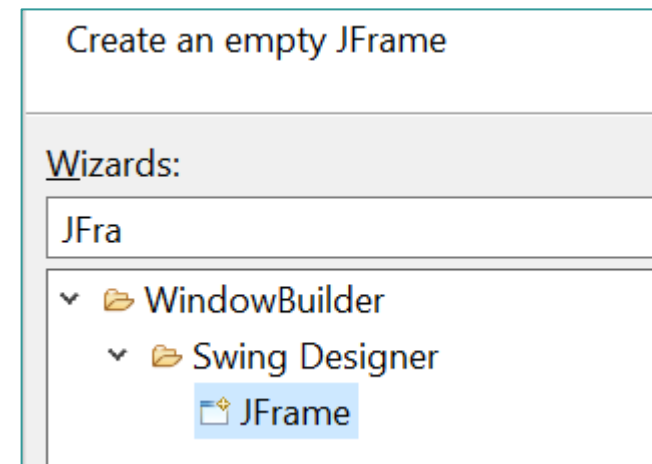
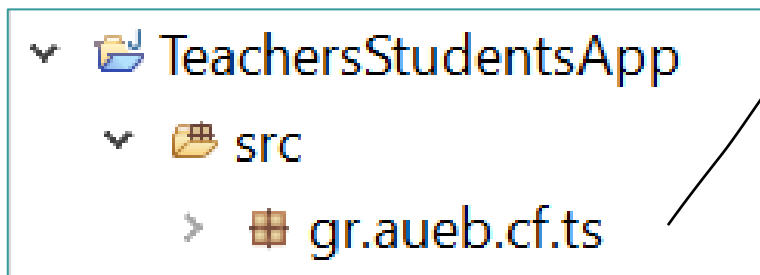
View (1)

- Το View αποτελείται από 4 Οθόνες:
 - Μία οθόνη με το **Κύριο Μενού**
 - Μία οθόνη για το **Υπομενού αναζήτησης / Εισαγωγής**
 - Δύο ακόμα οθόνες. 1) Μία οθόνη **Εισαγωγής** και 2) μία οθόνη **Αναζήτησης**
Μέσα από την οθόνη Αναζήτησης κάνουμε **Ενημέρωση** και **Διαγραφή**




Κύριο Μενού (1)


- Αφού δημιουργήσουμε ένα νέο project, **TeachersStudentsApp** και ένα package *gr.aueb.cf.ts* στη συνέχεια δημιουργούμε μία κλάση *JFrame* (*new / Other / JFrame*) για να αναπτύξουμε την οθόνη του Μενού





Κύριο Μενού (2)

 New JFrame — □ ×

Create JFrame 

Create an empty JFrame.


Source folder:

Package:

Name:

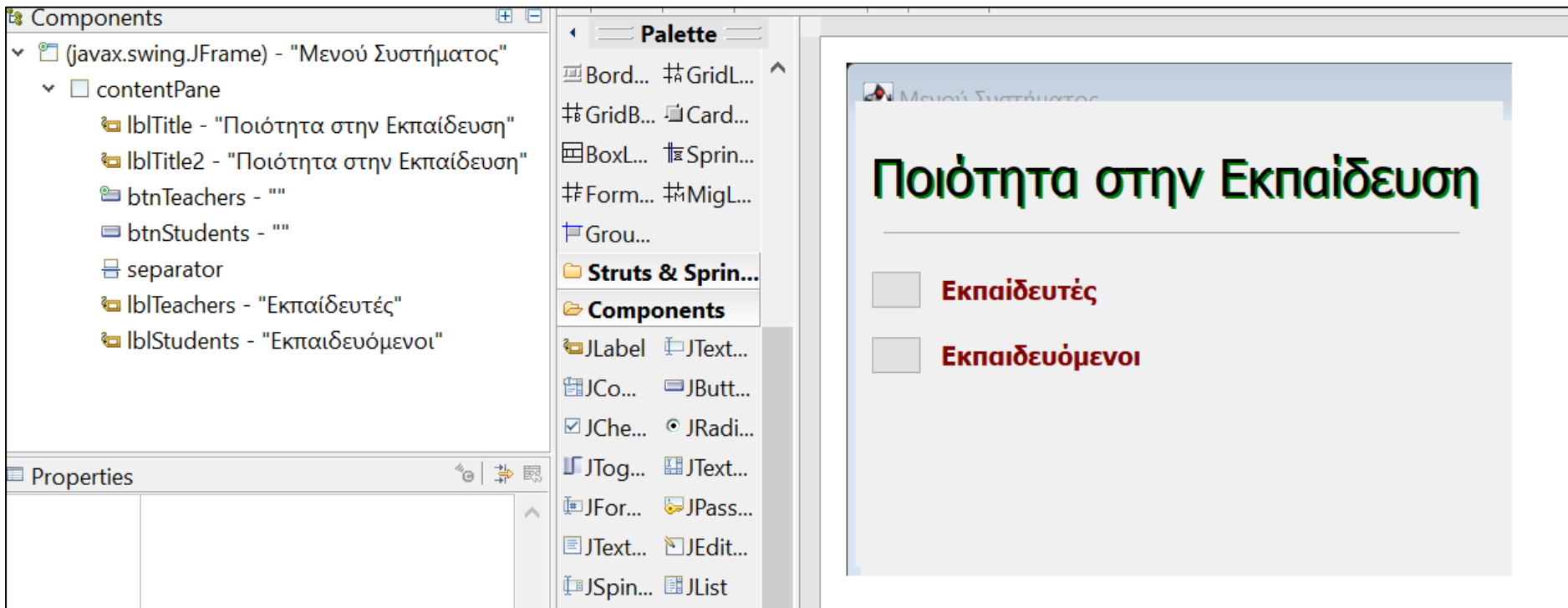
Superclass:

☒ Use advanced template for generate JFrame





Οθόνη Μενού



- Εισάγουμε τα στοιχεία της οθόνης μέσα σε ένα JPanel (container με σύνηθες όνομα *contentPane*) που είναι μέσα στο JFrame
 - Ένα *JLabel* για τον τίτλο (+ ένα για σκιά)
 - Δύο *JButton* και δύο *JLabel* για το Μενού



JDBC Driver

The screenshot shows the MySQL Product Archives page for Connector/J. The browser address bar shows the URL: downloads.mysql.com/archives/c-j/. The page title is "MySQL Product Archives" and the breadcrumb is "MySQL Connector/J (Archived Versions)". A yellow warning box states: "Please note that these are old versions. New releases will have recent bug fixes and features! To download the latest release of MySQL Connector/J, please visit [MySQL Downloads](#)." Below the warning, there are two dropdown menus: "Product Version" set to "8.0.33" and "Operating System" set to "Platform Independent". A table lists two download options:

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-j-8.0.33.tar.gz)	Mar 7, 2023	4.0M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-j-8.0.33.zip)	Mar 7, 2023	4.8M	Download

MD5 hashes and signatures are provided for each download option.

- <https://downloads.mysql.com/archives/c-j/>
- Κατεβάζουμε τον JDBC Driver (Connector/J 8.0.33 ή παλαιότερη έκδοση από τα Archives). Σημειωτέον, αν έχουμε κάνει full installation του MySQL Server, έχει κατέβει ο J-Connector στο Program Files (x86) / MySQL



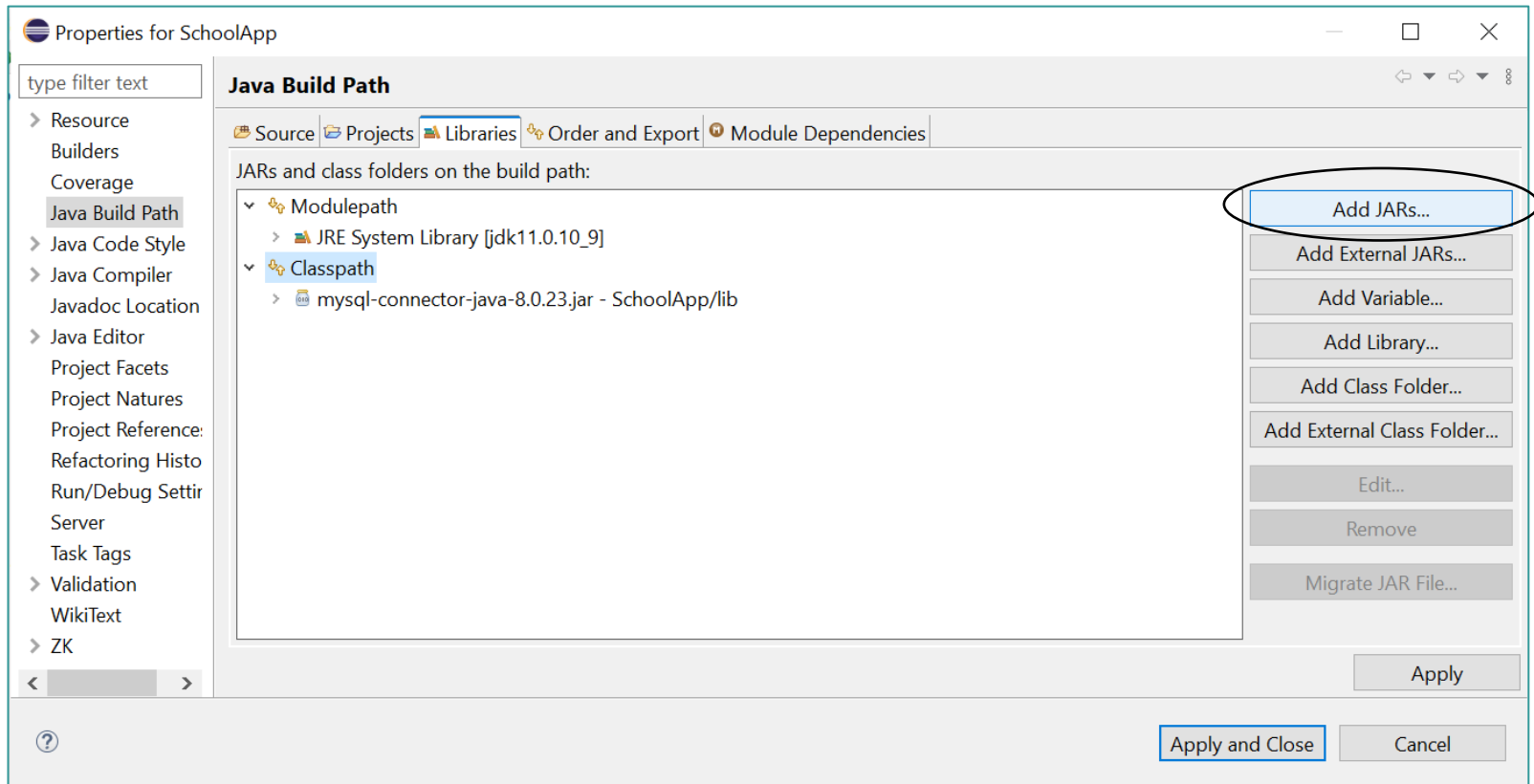
Φάκελος lib

```
▼ TeachersStudentsApp
  > src
  > JRE System Library [jdk11.0.10_9]
  > Referenced Libraries
  > lib
      mysql-connector-java-8.0.23.jar
```

- Δημιουργούμε ένα φάκελο lib στο project και εισάγουμε τον connector ώστε να είναι μέρος του project και να είναι μεταφέρσιμο



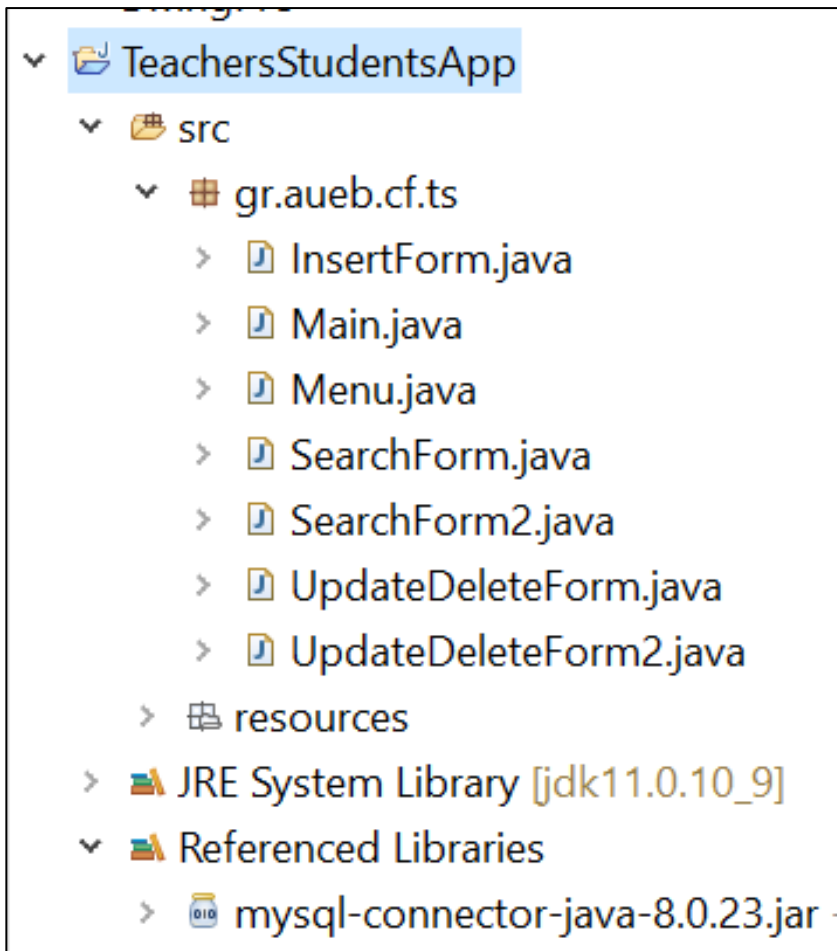
Add JARs



- Στα Properties του Project και στο Java Build Path εισάγουμε στο *Classpath* με το Add JARs τον JDBC Driver (σσ. τον Connector/J)



Dependencies



- Τα third-party libraries τα οποία χρειάζεται η εφαρμογή μας για να λειτουργήσει ονομάζονται dependencies
- Τα dependencies μπορεί να έχουν είτε compile scope ή runtime scope ή και άλλα scopes (system, provides, test)
- Στο Classpath του Eclipse εισάγουμε compile-time και runtime dependencies
- Ο MySQL connector είναι runtime dependency



Σύνδεση με τη ΒΔ

```
this.addWindowListener(new WindowAdapter() {
    @Override
    public void windowOpened(WindowEvent e) {
        String url = "jdbc:mysql://localhost:3306/tsdb?serverTimeZone=UTC";
        String username = "thanos3";
        String password = "Thanos3";
        //String password = System.getenv("TS_USER_PASSWORD");

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, username, password);
            System.out.println("Connection Established");
        } catch (SQLException e1) {
            e1.printStackTrace();
        } catch (ClassNotFoundException e1) {
            e1.printStackTrace();
        }
    }
});
```

- Όταν ανοίγει η εφαρμογή (WindowOpened event) συνδεόμαστε με τη ΒΔ, ορίζοντας το connection string (url, username, password) και καλώντας την static μέθοδο getConnection() του DriverManager
- Χρειαζόμαστε επομένως το interface *Connection*, την κλάση *DriverManager* και την κλάση *SQLException*. Για λόγους ασφάλειας το password με το οποίο συνδεόμαστε δεν πρέπει να βρίσκεται μέσα στον κώδικα



Αποθήκευση password

Swing Desktop Full Stack Apps

Ιδιότητες συστήματος

Προστασία συστήματος

Απομακρυσμένη σύνδεση

Όνομα υπολογιστή

Υλικό

Για προχωρημένους

Πρέπει να έχετε συνδεθεί ως Administrator για την πραγματοποίηση των περισσότερων από αυτές τις αλλαγές.

Επιδόσεις

Οπτικά εφέ, χρονοδιάγραμμα επεξεργαστή, χρήση μνήμης και εικονική μνήμη

Ρυθμίσεις...

Προφίλ χρηστών

Ρυθμίσεις επιφάνειας εργασίας σχετικές με την είσοδο

Ρυθμίσεις...

Εκκίνηση και αποκατάσταση

Εκκίνηση συστήματος, αποτυχία συστήματος και πληροφορίες αντιμετώπισης σφαλμάτων

Ρυθμίσεις...

Μεταβλητές περιβάλλοντος...

Μεταβλητή	Τιμή
OneDrive	C:\Users\asana\OneDrive - aueb.gr
OneDriveCommercial	C:\Users\asana\OneDrive - aueb.gr
OneDriveConsumer	C:\Users\asana\OneDrive
Path	C:\Users\asana\cargo\bin;C:\Program Files\MySQL\MySQL Shell...
TEMP	C:\Users\asana\AppData\Local\Temp
TMP	C:\Users\asana\AppData\Local\Temp

Δημιουργία...

Επεξεργασία...

Διαγραφή...

Μεταβλητές συστήματος

Μεταβλητή	Τιμή
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
TS_USER_PASSWORD	Thanos3
USERNAME	SYSTEM
VS160COMCOMNTOOLS	C:\Program Files (x86)\Microsoft Visual Studio\2019\Community...
windir	C:\WINDOWS
ZES_ENABLE_SYSMAN	1

- Ένα σημείο που μπορούμε να αποθηκεύσουμε ειδικά passwords είναι στις μεταβλητές του Συστήματος. Άλλο μέρος θα ήταν σε κάποιο Directory Service όπως το LDAP.



Αποθήκευση Password

Μεταβλητές συστήματος	
Μεταβλητή	Τιμή
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
TS_USER_PASSWORD	Thanos3
USERNAME	SYSTEM
VS160COMCOMNTOOLS	C:\Program Files (x86)
windir	C:\WINDOWS
ZES_ENABLE_SYSMAN	1

- Μπορούμε να αποθηκεύσουμε το password του Connection string σε μία global μεταβλητή του Συστήματος όπως π.χ. TS_USER_PASSWORD



Listeners

- Ορίζουμε listeners για τα events, όπως events σε buttons.
- Στην εφαρμογή αυτή θα ορίσουμε και ένα Listener σε ένα event του Συστήματος, όπως το window opened, που γίνεται raise όταν ανοίξει ένα παράθυρο



Listeners / Event Handlers

```
this.addWindowListener(new WindowAdapter() {
    @Override
    public void windowOpened(WindowEvent e) {
        String url = "jdbc:mysql://localhost:3306/tsdb?serverTimeZone=UTC";
        String username = "thanos3";
        //String password = "Thanos3";
        String password = System.getenv("TS_USER_PASSWORD");

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, username, password);
            System.out.println("Connection Established");
        } catch (SQLException e1) {
            e1.printStackTrace();
        } catch (ClassNotFoundException e1) {
            e1.printStackTrace();
        }
    }
});
```

- Στον window opened listener αντικαταστήσαμε το password με την System.getenv("TS_USER_PASSWORD") που επιστρέφει την τιμή (το password) της μεταβλητής TS_USER_PASSWORD



Open another window

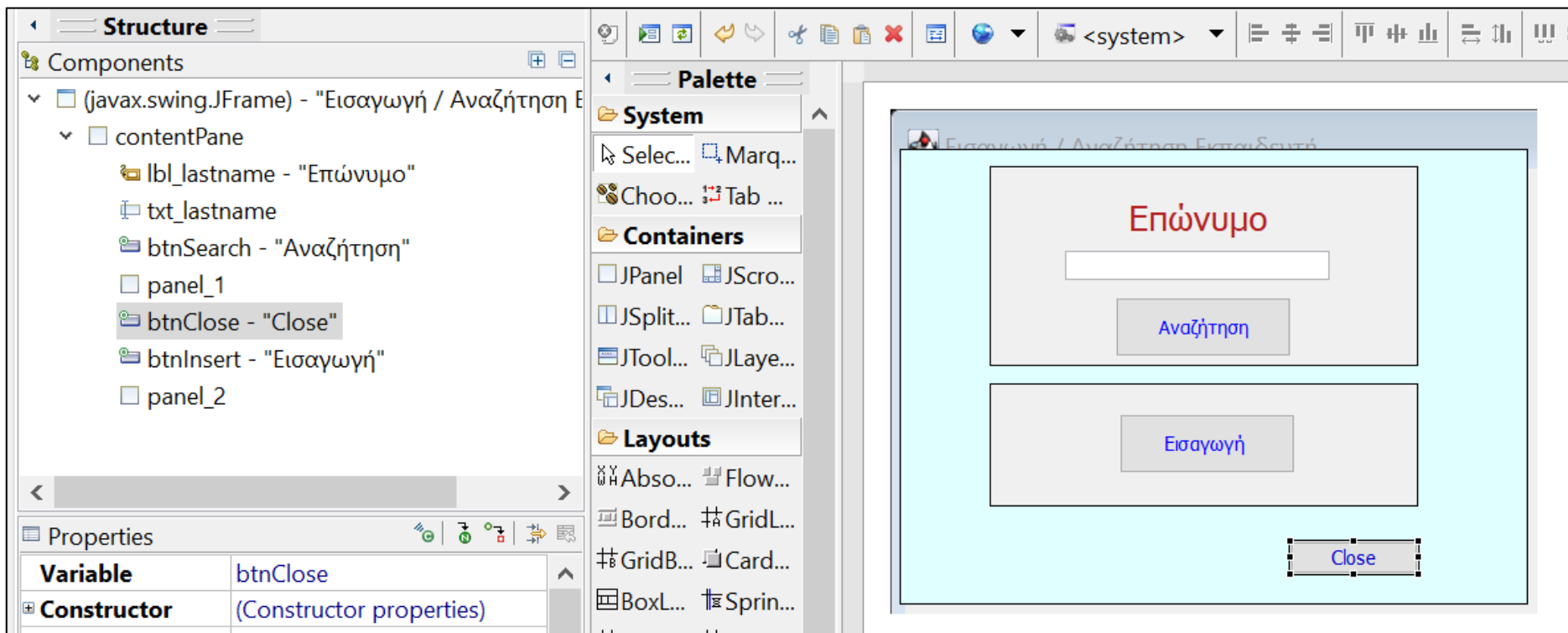
```
JButton btnTeachers = new JButton("");
btnTeachers.setBounds(12, 132, 40, 30);
btnTeachers.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Main.getSearchForm().setVisible(true);
        Main.getMenu().setEnabled(false);
    }
});
contentPane.add(btnTeachers);
```

- Στο button btnTeachers στο onclick (ActionEvent) ανοίγει η φόρμα του Υπομενού (γίνεται **visible**) και το Μενού γίνεται **disabled** (modal)



Υπομενού Εισαγωγής / Αναζήτησης

Swing Desktop Full Stack Apps



- Στο υπομενού Αναζήτησης / Εισαγωγής Εκπαιδευτή έχουμε τρία buttons, άρα θα χρειαστούμε τρεις Listeners



Search Form Listeners

```
btnSearch.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        inputLastname = txt_lastname.getText();  
        Main.getSearchForm().setEnabled(false);  
        Main.getUpdateDeleteForm().setVisible(true);  
    }  
});
```

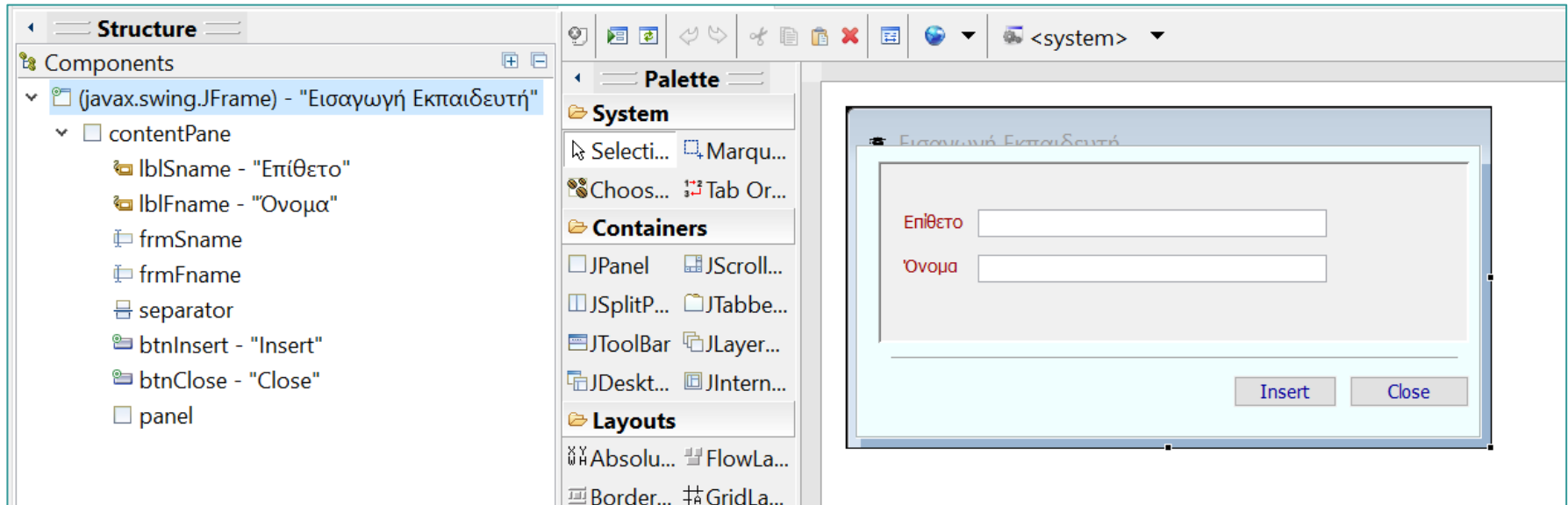
```
JButton btnInsert = new JButton("Εισαγωγή");  
btnInsert.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        Main.getSearchForm().setEnabled(false);  
        Main.getInsertForm().setVisible(true);  
    }  
});
```

```
btnClose.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        Main.getMenu().setEnabled(true);  
        Main.getSearchForm().setVisible(false);  
    }  
});
```

- Ο Listener για το Search διαβάζει το input πεδίο αναζήτησης, το κάνει set και και ανοίγει τη φόρμα update-delete
- Ο Listener για την Εισαγωγή απλά ανοίγει την φόρμα Εισαγωγής κάνοντάς την visible και επίσης disable την παρούσα φόρμα
- Ο Close Listener κλείνει (invisible) την παρούσα φόρμα και κάνει enable την φόρμα του Menu



Insert Form



- Όπως παρατηρούμε έχουμε δύο events *onclick* (Action events στο Swing).
 1. Η φόρμα εισαγωγής 'πατάει' στον πίνακα. Δηλαδή είναι μία γραφική αναπαράσταση προς τον χρήστη του ίδιου του πίνακα και επιλέγοντας *Insert* θα γίνεται INSERT INTO TEACHERS...
 2. Το *close* που κλείνει (invisible) την παρούσα φόρμα και ενεργοποιεί την προηγούμενη φόρμα (Search Menu)



Insert Form

```
btnInsert = new JButton("Insert");
btnInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String inputLastname = frmSname.getText().trim();
            String inputFirstname = frmFname.getText().trim();
            int n;

            if (inputLastname.equals("") || inputFirstname.equals("")) {
                return;
            }
            p = Menu.getConn().prepareStatement("INSERT INTO TEACHERS (FIRSTNAME, LASTNAME ) VALUES (?, ?)");
            p.setString(1, inputFirstname);
            p.setString(2, inputLastname);
            n = p.executeUpdate();
            JOptionPane.showMessageDialog(null, n + " recordes inserted", "INSERT", JOptionPane.PLAIN_MESSAGE);
        } catch (SQLException e1) {
            e1.printStackTrace();
            JOptionPane.showMessageDialog(null, "Something went wrong with Insertion, Please try again", "Error",
                JOptionPane.WARNING_MESSAGE);
        } finally {
            try {
                p.close();
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
    }
});
```

- *PreparedStatement* με *positioned placeholders* (?, ?) και μετά με setters (*setString()*) εισάγουμε τα user-supplied data



Reset πεδία Φόρμας

```
addWindowListener(new WindowAdapter() {  
    @Override  
    public void windowActivated(WindowEvent e) {  
        frmSname.setText("");  
        frmFname.setText("");  
    }  
});
```

- Deactivated γίνεται η φόρμα όταν πετάγεται το JOptionPane από τον ActionListener
- Τότε γίνονται reset τα πεδία της Φόρμας



SQL Injection

```
// SQLInjection attacks may happen when inserting user-supplied data directly into the query
// The query may be altered by the attacker. For instance:
// p = Menu.getConnection().prepareStatement("INSERT INTO TEACHERS (FIRSTNAME, LASTNAME) VALUES ('"
//                                     + inputFirstname + "', '" + inputLastname + "')");
// Safely use place-holders (?) - So, the query is precompiled and the the user-supplied data are inserted literally.
```

- Ποτέ δεν εισάγουμε user-supplied data απευθείας στο Query αν και έχουμε τεχνικά τη δυνατότητα (όπως παραπάνω μέσα στα σχόλια)
- Αν είχαμε, για παράδειγμα, ένα query με WHERE PASSWORD = 'inputPassword' και ο attacker έδινε ως input password όχι ένα String αλλά ένα expression: a' or '1' ='1, τότε θα μπορούσε να αλλάξει structurally το query και να εισέλθει στο σύστημα

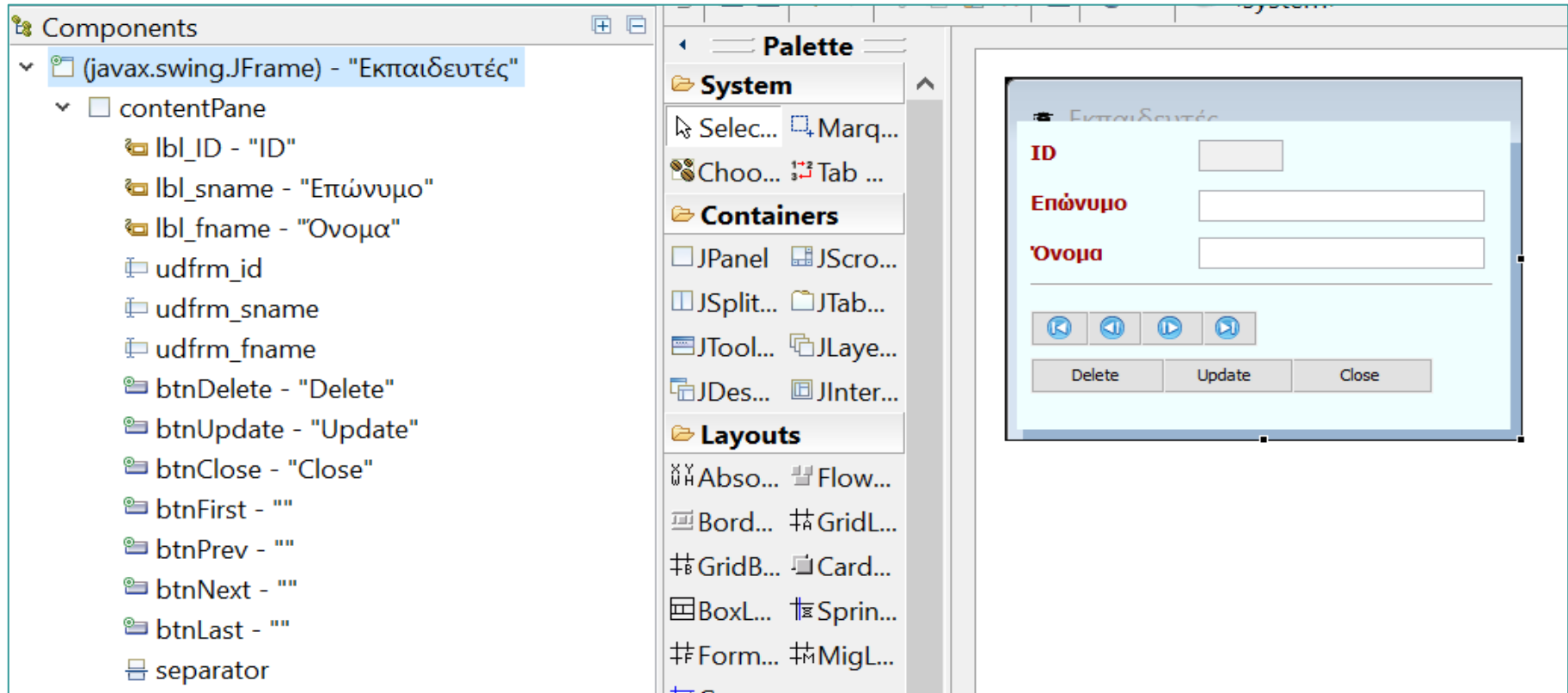


Prepared Statement

- Τα prepared statements δίνουν λύση στο πρόβλημα του SQL Injection γιατί το query μεταγλωττίζεται με τους placeholders οι οποίοι αντικαθίστανται @runtime όποτε και διερμηνεύονται πάντα ως Strings, όχι ως expressions, οπότε δεν μπορεί να μεταβληθεί structurally το query



Φόρμα Update / Delete



- Το id είναι disabled.
- Η φόρμα αυτή 'πατάει' σε Query (SELECT). Επιστρέφει *ResultSet*. Υπάρχουν και τέσσερα κουμπιά περιήγησης και τρία buttons, οπότε χρειαζόμαστε ισάριθμους Listeners.



Select - Result Set (1)

```
addWindowListener(new WindowAdapter() {
    @Override
    public void windowActivated(WindowEvent e) {
        String sql = "SELECT ID, FIRSTNAME, LASTNAME FROM TEACHERS WHERE LASTNAME LIKE ?";

        try {
            conn = Menu.getConn();
            p = conn.prepareStatement(sql, ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE);

            p.setString(1, Main.getSearchForm().getInputLastname() + '%');
            rs = p.executeQuery();

            if (rs.next()) {
                txtId.setText(Integer.toString(rs.getInt("ID")));
                txtFirstname.setText(rs.getString("FIRSTNAME"));
                txtLastname.setText(rs.getString("LASTNAME"));
            } else {
                txtId.setText("");
                txtFirstname.setText("");
                txtLastname.setText("");
            }
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
})
```

- Το *WindowActivated* γίνεται raise όταν το window ενεργοποιείται. Όταν για παράδειγμα δημιουργείται ένα popup window από το JOptionPane, τότε το παράθυρό μας γίνεται deactivated και μόλις κλείσει το popup window, ξαναγίνεται activated



Select - Result Set (2)

```
@Override
public void windowClosed(WindowEvent e) {
    try {
        if (rs != null) rs.close();
        if (pst != null) pst.close();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
}
});
```

- Το event window closed ενεργοποιείται όταν το παράθυρο κλείσει. Κάνουμε close πρώτα το result set και μετά το prepared statement μιας και αποτελούν resources και είναι security leak να παραμένουν ενεργά (το close prepared statement κάνει close και το rs αλλά ωστόσο κάνουμε τα close διακριτά)



Select - Result Set (3)

- Το ResultSet είναι ένα interface, που λειτουργεί ως in-memory δομή πίνακα (Object[][]), όπου η κάθε row είναι μία εγγραφή του πίνακα της Βάσης
- Με `pst.executeQuery()` εκτελούμε το `PreparedStatement` και τα αποτελέσματα πάνε στο `ResultSet`



Select - Result Set (3)

- Επειδή ο cursor (pointer) στο ResultSet δείχνει πριν την 1^η εγγραφή, πρέπει πάντα να ξεκινάμε με `if (rs.next())` ή `while (rs.next())`. Με **`rs.getInt()`** και **`rs.getString()`** (για κάθε τύπο δεδομένων υπάρχει `get*`) με βάση το όνομα του πεδίου ή τη θέση του column στον πίνακα μπορούμε και διαβάζουμε πεδία του Result Set και τα αντιστοιχούμε στην φόρμα με `setText()`



Delete Listener

```

JButton btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String query = "DELETE from TEACHERS where ID = ?";
        int response;

        try (PreparedStatement preparedStmt = Menu.getConn().prepareStatement(query)) {
            preparedStmt.setInt(1, Integer.parseInt(udfrm_id.getText()));

            response = JOptionPane.showConfirmDialog (null, "Είστε σίγουρος;",
                "Warning", JOptionPane.YES_NO_OPTION);

            if (response == JOptionPane.YES_OPTION){
                int numberOfRowsAffected = preparedStmt.executeUpdate();
                JOptionPane.showMessageDialog (null, numberOfRowsAffected + " rows deleted successfully",
                    "DELETE", JOptionPane.INFORMATION_MESSAGE);
            }
        } catch (SQLException e6) {
            e6.printStackTrace();
        }
    }
});
```

- Πρώτα εμφανίζουμε ένα confirm dialog και μετά αν είναι OK, κάνουμε executeUpdate(). Δίνουμε επιβεβαίωση με message dialog



Update Listener

```
JButton btnUpdate = new JButton("Update");
btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String query = "UPDATE TEACHERS set FIRSTNAME = ?, LASTNAME = ? where ID = ?";
        int numberOfRowsAffected;

        try (PreparedStatement preparedStmt = Menu.getConn().prepareStatement(query)) {
            preparedStmt.setString(1, udfm_fname.getText());
            preparedStmt.setString(2, udfm_sname.getText());
            preparedStmt.setInt(3, Integer.parseInt(udfm_id.getText()));

            numberOfRowsAffected = preparedStmt.executeUpdate();

            JOptionPane.showMessageDialog(null, numberOfRowsAffected
                + " rows affected", "UPDATE", JOptionPane.PLAIN_MESSAGE);
            preparedStmt.close();
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
});
```

Κάνουμε update με `executeUpdate()`



First Record Button

```
btnFirst.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
            if (rs.first()){  
                udfrm_id.setText(Integer.toString(rs.getInt("ID")));  
                udfrm_sname.setText(rs.getString("LASTNAME"));  
                udfrm_fname.setText(rs.getString("FIRSTNAME"));  
            }  
        } catch (SQLException e2) {  
            e2.printStackTrace();  
        }  
    }  
});
```

- Listener για το κουμπί που μας πάει στην 1^η εγγραφή του Result Set (αν υπάρχει)



Previous Record Button

```
btnPrev.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
            if (rs.previous()) {  
                udfrm_id.setText(Integer.toString(rs.getInt("ID")));  
                udfrm_sname.setText(rs.getString("LASTNAME"));  
                udfrm_fname.setText(rs.getString("FIRSTNAME"));  
            }  
            else  
            {  
                rs.first();  
            }  
        } catch (SQLException e3) {  
            e3.printStackTrace();  
        }  
    }  
});
```

- Listener για το κουμπί που μας πάει στην προηγούμενη εγγραφή του Result Set, αν υπάρχει, αλλιώς στην 1η



Next Record Button

```
btnNext.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
            if (rs.next()){  
                udfrm_id.setText(Integer.toString(rs.getInt("ID")));  
                udfrm_sname.setText(rs.getString("LASTNAME"));  
                udfrm_fname.setText(rs.getString("FIRSTNAME"));  
            }  
            else  
                rs.last();  
        } catch (SQLException e4) {  
            e4.printStackTrace();  
        }  
    }  
});
```

- Listener για το κουμπί που μας πάει στην επόμενη εγγραφή του Result Set (αν υπάρχει)



Last Record Button

```
btnNext.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
            if (rs.next()){  
                udfrm_id.setText(Integer.toString(rs.getInt("ID")));  
                udfrm_sname.setText(rs.getString("LASTNAME"));  
                udfrm_fname.setText(rs.getString("FIRSTNAME"));  
            }  
            else  
                rs.last();  
        } catch (SQLException e1) {  
            e1.printStackTrace();  
        }  
    }  
});
```

- Listener για το κουμπί που μας πάει στην τελευταία εγγραφή του Result Set (αν υπάρχει)



Login Page

Username

Password

Login

- Μπορούμε να κάνουμε login ως admin ή ως απλός χρήστης. Το password του admin είναι σε μεταβλητή περιβάλλοντος, ενώ τα passwords των χρηστών είναι στη ΒΔ



Διαχείριση Χρηστών

Swing Desktop Full Stack Apps

- Μετά το Login του admin οδηγούμαστε σε αυτή την οθόνη



Εισαγωγή User

Swing Desktop Full Stack Apps

Εισαγωγή Χρήστη

Username

Password

Insert Close



- Όταν αποθηκεύουμε passwords δεν αποθηκεύουμε plain text, αλλά hashed passwords δηλαδή κρυπτογραφημένα passwords
- Ο βασικός αλγόριθμος βασίζεται στο Blowfish (symmetric block cipher)
- Ο αλγόριθμος τρέχει επαναληπτικά και όσο περισσότερες επαναλήψεις, τόσο πιο αργός είναι και τόσο πιο δύσκολο να σπάσει
- Το cost factor είναι ένας εκθέτης του 2 που ορίζει πόσες επαναλήψεις να κάνει το bCrypt. Το ελάχιστο cost factor είναι 10 και ιδανικά 12 (ή περισσότερο)



jBCrypt dependency

Java Build Path

Source Projects Libraries Order and Export Module Dependencies

JARs and class folders on the build path:

- ▼ Modulepath
 - > JRE System Library [jdk11.0.10_9]
- ▼ Classpath
 - > commons-dbcp2-2.9.0.jar - C:\Program Files\commons-dbcp2-2.9.0
 - > commons-logging-1.2.jar - C:\Program Files\commons-logging-1.2
 - > commons-pool2-2.11.1.jar - C:\Program Files\commons-pool2-2.11.1
 - > jBCrypt-0.4.3.jar - C:\Users\A8ana\Downloads
 - > mysql-connector-java-8.0.23.jar - C:\Program Files\Java



Admin Login

```
btnLogin.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        String inputUsername = txtUsername.getText().trim();  
        String inputPassword = String.valueOf(txtPassword.getPassword()).trim();  
        String password = System.getenv("TS_ADMIN_PASSWORD");  
  
        if (inputUsername.equals("") || inputPassword.equals("")) {  
            return;  
        }  
  
        if ((inputUsername.equals("admin")) && (inputPassword.equals(password)))  
            Main.getLoginPage().setVisible(false);  
            Main.getSearchUsers().setVisible(true);  
        } else {
```

- Το admin password είναι στη μεταβλητή περιβάλλοντος TS_ADMIN_PASSWORD



User Login

```
} else {
    String sql = "SELECT PASSWORD FROM USERS WHERE USERNAME = ?";

    try (Connection conn = DBUtil.getConnection();
        PreparedStatement p = conn.prepareStatement(sql)) {

        p.setString(1, inputUsername);

        ResultSet rs = p.executeQuery();
        if (rs.next()) {
            hashedPassword = rs.getString("PASSWORD");
        } else {
            JOptionPane.showMessageDialog(null,
                "User not found", "Error", JOptionPane.WARNING_MESSAGE);
            return;
        }

        if (BCrypt.checkpw(inputPassword, hashedPassword)) {
            Main.getMenu().setVisible(true);
            Main.getLoginPage().setVisible(false);
        } else {
            JOptionPane.showMessageDialog(null, "Error in password", "Login", JOptionPane.PLAIN_MESSAGE);
        }
    } catch (SQLException e1) {
        e1.printStackTrace();
        JOptionPane.showMessageDialog(null, "Something went wrong with Insertion, Please try again",
            "Error", JOptionPane.WARNING_MESSAGE);
    }
}

});
```



Insert User

```
btnInsert.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String sql = "INSERT INTO USERS (USERNAME, PASSWORD) VALUES (?, ?)";
        int n;

        try (Connection conn = DBUtil.getConnection();
            PreparedStatement p = conn.prepareStatement(sql);) {

            String inputUsername = txtUsername.getText().trim();
            String inputPassword = String.valueOf(txtPassword.getPassword()).trim();

            if (inputUsername.equals("") || inputPassword.equals("")) {
                return;
            }

            int workload = 12;
            String salt = BCrypt.gensalt(workload);
            String hashedPassword = BCrypt.hashpw(inputPassword, salt);

            //return(hashed_password);
            p.setString(1, inputUsername);
            //p.setString(2, inputPassword);
            p.setString(2, hashedPassword);

            n = p.executeUpdate();

            JOptionPane.showMessageDialog(null, n + " recordes inserted", "INSERT", JOptionPane.PLAIN_MESSAGE);
        } catch (SQLException e1) {
            e1.printStackTrace();
            JOptionPane.showMessageDialog(null, "Something went wrong with Insertion, Please try again",
                "Error", JOptionPane.WARNING_MESSAGE);
        }
    }
});
```



Connection Pool

Swing Desktop Full Stack Apps

- Η δημιουργία connections είναι μία ακριβή σε όρους χρόνου διαδικασία
- Ιδανικά θα θέλαμε σε κάθε thread να δίνουμε ένα νέο connection προς την ΒΔ χωρίς την καθυστέρηση της δημιουργία του Connection
- Το Connection Pool είναι ένα database connection cache implementation



Apache Commons DBCP - Pooling framework

Swing Desktop Full Stack Apps

Java Build Path

Source Projects Libraries Order and Export Module Dependencies

JARs and class folders on the build path:

- ▼ Modulepath
 - > JRE System Library [jdk11.0.10_9]
- ▼ Classpath
 - > commons-dbc2-2.9.0.jar - C:\Program Files\commons-dbc2-2.9.0
 - > commons-logging-1.2.jar - C:\Program Files\commons-logging-1.2
 - > commons-pool2-2.11.1.jar - C:\Program Files\commons-pool2-2.11.1
 - > jBCrypt-0.4.3.jar - C:\Users\a8ana\Downloads
 - > mysql-connector-java-8.0.23.jar - C:\Program Files\Java



DBUtil – Utility class (1)

Swing Desktop Full Stack Apps

```
1 package gr.aueb.cf.ts.util;
2
3 import java.sql.Connection;
4 import java.sql.SQLException;
5
6 import org.apache.commons.dbcp2.BasicDataSource;
7
8 public class DBUtil {
9
10     private static BasicDataSource ds = new BasicDataSource();
11     private static Connection conn;
12
13     /**
14      * No instances of this class should be available
15      */
16     private DBUtil(){ }
17
18     static {
19         ds.setUrl("jdbc:mysql://localhost:3306/tsdb?serverTimeZone=UTC");
20         ds.setUsername("thanos3");
21         ds.setPassword("Thanos3");
22         ds.setInitialSize(8);
23         ds.setMaxTotal(32);
24         ds.setMinIdle(8);
25         ds.setMaxIdle(10);
26         ds.setMaxOpenPreparedStatements(100);
27     }
28 }
```



DBUtil – Utility class (2)

Swing Desktop Full Stack Apps

```
29 public static Connection getConnection() throws SQLException {  
30     conn = ds.getConnection();  
31     //return ds.getConnection();  
32     return conn;  
33 }  
34  
35 public static void closeConnection() {  
36     try {  
37         if (conn != null) conn.close();  
38     } catch (SQLException e) {  
39         e.printStackTrace();  
40     }  
41 }  
42 }
```

- Με try with resources μπορεί να γίνει και auto-close



Ιδιότητες BasicDataSource

```
ds.setUrl("jdbc:mysql://localhost:3306/tsdb?serverTimeZone=UTC");  
ds.setUsername("thanos3");  
ds.setPassword("Thanos3");  
ds.setInitialSize(8);  
ds.setMaxTotal(32);  
ds.setMinIdle(8);  
ds.setMaxIdle(10);  
ds.setMaxOpenPreparedStatements(100);
```

- Ο αριθμός των αρχικών connections είναι 8 καθώς και τα ελάχιστα idle connections (δεν έχουν εκχωρηθεί σε thread).



Εργασία

- Υλοποιήστε μία αντίστοιχη εφαρμογή Swing αλλά **με δικό σας domain model**, π.χ. Γιατροί – Ασθενείς, Προμηθευτές-Προϊόντα, Προϊόντα – Πελάτες, κλπ.
- **Χωρίς Users και χωρίς Login Page**, υποθέστε ότι είστε Admin
- Χρησιμοποιήστε **Connection Pooling**
- Αναφερθείτε στην υλοποίηση που θα ανέβει στο GitHub