ΤΕΙ ΑΘΗΝΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΣΚΗΣΗ ΕΞΑΜΗΝΟΥ – ΔΙΑΦΗΜΙΣΗ ΜΑΘΗΣΙΑΚΩΝ ΕΥΚΑΙΡΙΩΝ

ΓΙΑ ΤΟ ΜΑΘΗΜΑ

«ΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ»

<<131037 ΔΗΜΗΤΡΗΣ ΖΕΪΜΠΕΚΑΚΗΣ>>

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2016-2017

1. Βάση δεδομένων

Κάθε μαθητής θα πρέπει να μπορεί να εισέλθει στο σύστημα, οπότε χρειάζεται αναγνωριστικά εισόδου. Για τους σκοπούς της άσκησης δε χρειαζόμαστε περισσότερες πληροφορίες για τον μαθητή οπότε αυτά τα πεδία αποτελούν τον πίνακα **students**.

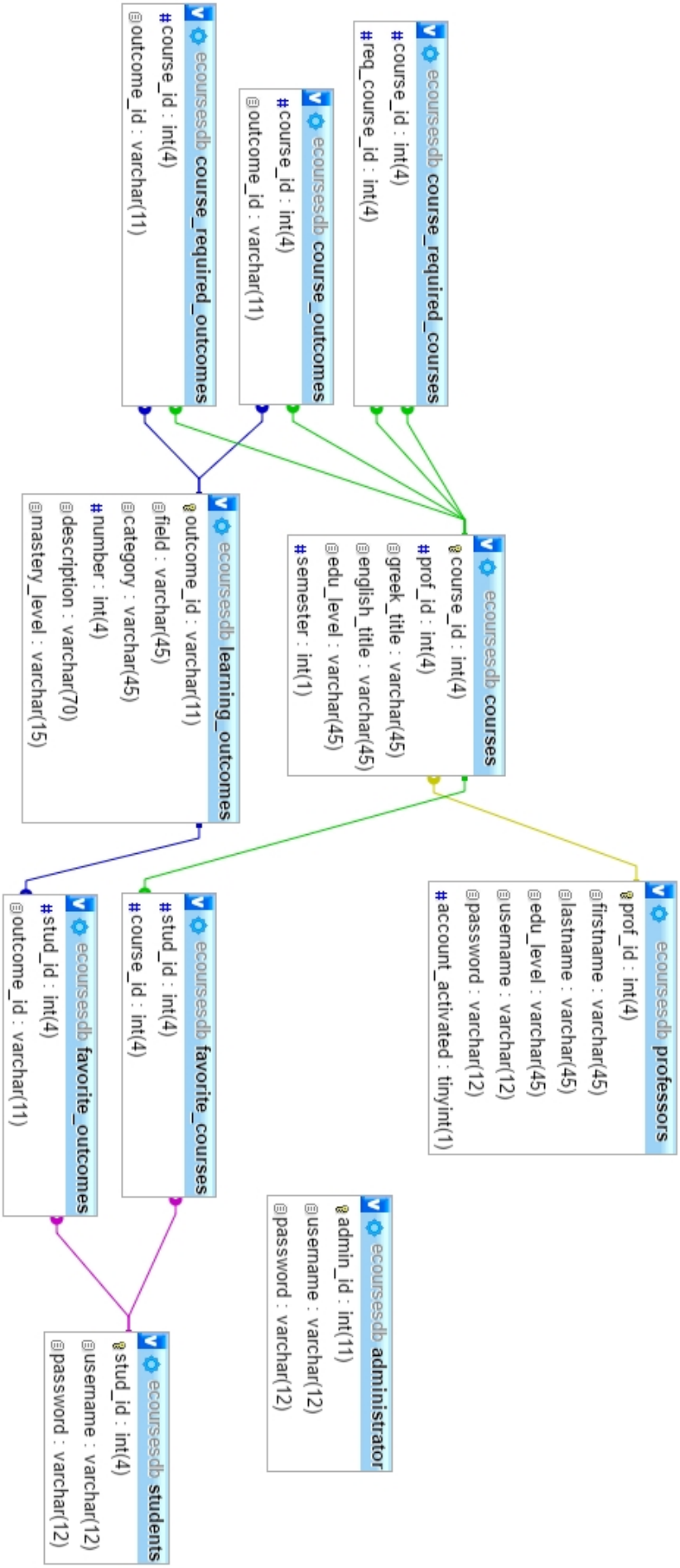
Επίσης θα πρέπει να μπορεί να αποθηκεύει αγαπημένους μαθησιακούς στόχους και αγαπημένες μαθησιακές ευκαιρίες, χρειαζόμαστε λοιπόν δύο πίνακες που να αποθηκεύουν την αντιστοίχηση των κύριων κλειδιών του μαθητή με τα αγαπημένα, οι **favorite_courses** και **favorite_outcomes**.

Τα πεδία του πίνακα μαθησιακών στόχων **learning_outcomes** θα είναι το αναγνωριστικό του στόχου είναι τα αρχικά της κωδικοποίησης ACM, για παράδειγμα, για τον στόχο Network and Communication / Introduction είναι το NC.I.1 .Τα πεδία field, category, number είναι ανάλυση της συντομογραφίας του αναγνωριστικού άρα σε αυτό το παράδειγμα Network and Communication, Introduction, 1 ενώ τα υπόλοιπα πεδία περισσότερες πληροφορίες για τον μαθησιακό στόχο.

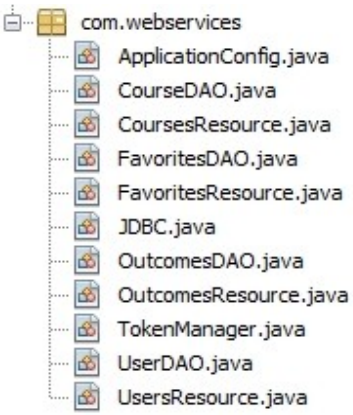
Κάθε καθηγητής **professor** θα πρέπει να εισέρχεται στο σύστημα αν ο λογαριασμός του είναι ενεργοποιημένος άρα χρειάζεται αναγνωριστικά εισόδου και ένα πεδίο ελέγχο ενεργού λογαριασμού. Από την εκφώνηση προκύπτουν τα υπόλοιπα πεδία.

Τα μαθήματα **courses** δημιουργούνται από τους καθηγητές οπότε υπάρχει μια σύνδεση μαθημάτων – καθηγητών μέσω του αναγνωριστικού του καθηγητή. Τα υπόλοιπα πεδία προκύπτουν από την εκφώνηση. Πεδία στα οποία μπορούν να υπάρχουν πολλαπλές επιλογές έχουν χωριστεί σε ξεχωριστούς πίνακες για καλύτερη κανονικοποίηση της βάσης και λειτουργούν ως σύνεσμοι ανάμεσα στα κύρια κλειδιά των πινάκων.

Ο διαχειριστής **administrator** θα πρέπει να μπορεί να εισέλθει στο σύστημα οπότε χρειάζεται αναγνωριστικά εισόδου.


coursesdb course_required_courses
course_id : int(4)
req_course_id : int(4)
coursesdb course_outcomes
course_id : int(4)
outcome_id : varchar(11)
coursesdb course_required_outcomes
course_id : int(4)
outcome_id : varchar(11)
coursesdb courses
course_id : int(4)
prof_id : int(4)
greek_title : varchar(45)
english_title : varchar(45)
edu_level : varchar(45)
semester : int(1)
coursesdb learning_outcomes
outcome_id : varchar(11)
field : varchar(45)
category : varchar(45)
number : int(4)
description : varchar(70)
mastery_level : varchar(15)
coursesdb professors
prof_id : int(4)
firstname : varchar(45)
lastname : varchar(45)
edu_level : varchar(45)
username : varchar(12)
password : varchar(12)
account_activated : tinyint(1)
coursesdb administrator
admin_id : int(11)
username : varchar(12)
password : varchar(12)
coursesdb favorite_courses
stud_id : int(4)
course_id : int(4)
coursesdb favorite_outcomes
stud_id : int(4)
outcome_id : varchar(11)
coursesdb students
stud_id : int(4)
username : varchar(12)
password : varchar(12)

2. Webservices



Για τα webservices χρησιμοποιείται XML για την ανταλλαγή δεδομένων με τις εφαρμογές.

Τα απεικονιζόμενα Resources με τη βοήθεια της βιβλιοθήκης MySQL JDBC Driver και τα αντίστοιχα Data Access Objects καλύπτουν τις παρακάτω λειτουργίες:

Outcomes Resource:

- επιστροφή λίστας μαθησιακών στόχων

Favorites Resource:

- αποθήκευση αγαπημένου μαθήματος
- αφαίρεση αγαπημένου μαθήματος
- επιστροφή λίστας με τα αγαπημένα μαθήματα
- αποθήκευση αγαπημένου μαθησιακού στόχου
- αφαίρεση αγαπημένου μαθησιακού στόχου
- επιστροφή λίστας με τους αγαπημένους μαθησιακούς στόχους

Courses Resource:

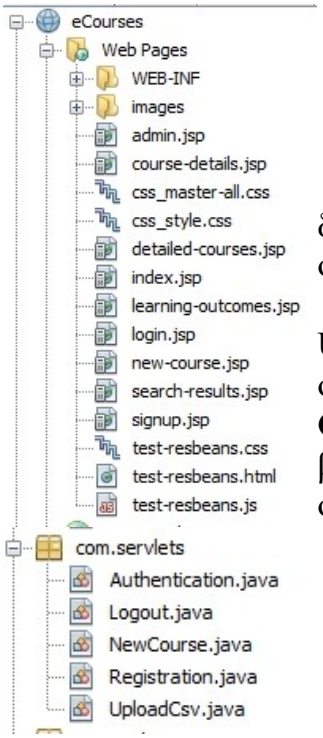
- επιστροφή λίστας μαθημάτων καθηγητή με βάση το id
- επιστροφή λεπτομέριων μαθήματος με βάση το όνομα του
- εισαγωγή μαθήματος
- εισαγωγή μαθημάτων από λίστα
- διαγραφή μαθήματος
- επιστροφή απλής λίστας μαθημάτων(τίτλοι, id)
- επιστροφή λεπτομερούς λίστας μαθημάτων
- επιστροφή απλής λίστας μαθημάτων με βάση τον τίτλο
- επιστροφή απλής λίστας μαθημάτων με βάση τον καθηγητή
- επιστροφή απλής λίστας μαθημάτων με βάση μαθησιακό στόχο

Users Resource:

- εισαγωγή νέου μαθητή
- εισαγωγή νέου καθηγητή
- ενεργοποίηση λογαριασμού καθηγητή
- απενεργοποίηση λογαριασμού καθηγητή
- επιστροφή στοιχείων καθηγητή με βάση το id
- επιστροφή λίστας όλων των καθηγητών
- αυθεντικοποίηση στο σύστημα*
- έξοδος από το σύστημα*

*Για την αυθεντικοποίηση και την έξοδο χρησιμοποιούνται Tokens που τα διαχειρίζεται ο **TokenManager** τα οποία περιέχουν 3 πεδία τύπο χρήστη(μαθητής ,δάσκαλος, διαχειριστής), id χρήστη και το token που είναι ένας universally unique identifier(UUID) το οποίο διανέμεται στον χρήστη για την αυθεντικοποίηση του και αποθηκεύεται σε έναν πίνακα στον server όπου αφαιρείται κατά την έξοδο.

3. Εφαρμογή Web



Για την web εφαρμογή χρησιμοποιούνται jsp pages για δυναμικό περιεχόμενο σε συνδυασμό με servlets για ορισμένες λειτουργίες.

Για τις κλήσεις στα webservices επιλέχθηκε η χρήση URL connections με το αντίστοιχο request method, τα οποία ομαδοποιήθηκαν κάτω από την βοηθητική κλάση **CommunicationHandler** η οποία αξιοποιείται από την βοηθητική κλάση που ορίστηκε με όνομα **XMLHandler** όπου διαχειρίζεται τους marshallers και unmarshallers για τα XML δεδομένα. Με εξαίρεση ελάχιστες επικοινωνίες όλες οι λήψεις και αποστολές δεδομένων γίνονται σε XML.

index.jsp - Μη συνδεδεμένος χρήστης

- Μπορεί να δει λίστα μαθημάτων(τίτλοι μαθημάτων).
- Μπορεί να δει λίστα μαθημάτων(καθηγητής, τίτλοι, εξάμηνο σπουδών) με περισσότερες πληροφορίες (μεταβαίνει στη **detailed-courses.jsp**).
- Μπορεί να δει όλες τις λεπτομέρειες του μαθήματος(μετάβαση σε **course-details.jsp**).
- Μπορεί να κάνει αναζήτηση μαθημάτων με τον τίτλο του μαθήματος ή τον διδάσκοντα του (μετάβαση σε **search-results.jsp**).

index.jsp

Courses | Learning Outcomes | Log In

Find: By Course Title | By Professor | Search

All Available Online Courses		Detailed Courses
English Title	**Greek Title**	
NetworksI	DiktiaI	See Details
NetworksII	DiktyaII	See Details

detailed-courses.jsp

Courses | Learning Outcomes | Log Out

All Available Online Courses				
Professor	**English Title**	**Greek Title**	**Semester**	
Profius	NetworksI	DiktiaI	1	More Details
Profius	NetworksII	DiktyaII	2	More Details

course-details.jsp

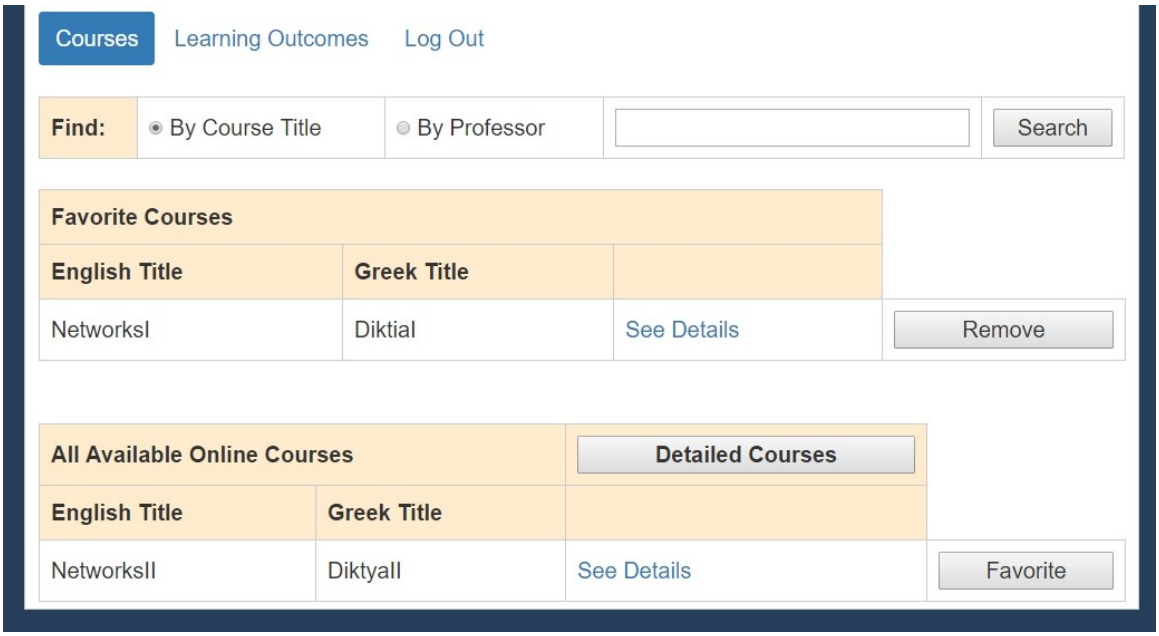
Courses | Learning Outcomes | Log Out

Course Details	
English Title	NetworksII
Greek Title	DiktyaII
Educational Level	undergraduate
Semester	2
Professor	Profius
Learning Outcomes	• NC.I.2
Required Courses	• NetworksI
Required Learning Outcomes	• NC.I.1

index.jsp - Συνδεδεμένος ενδιαφερόμενος

- Μπορεί να κάνει τα παραπάνω
- Τη προσθήκη ή αφαίρεση αγαπημένων μαθημάτων σε πίνακα αγαπημένων. Τα requests προς τα webservices γίνονται σε μορφή application/form-urlencoded για τη διαχείρηση αγαπημένων.

-

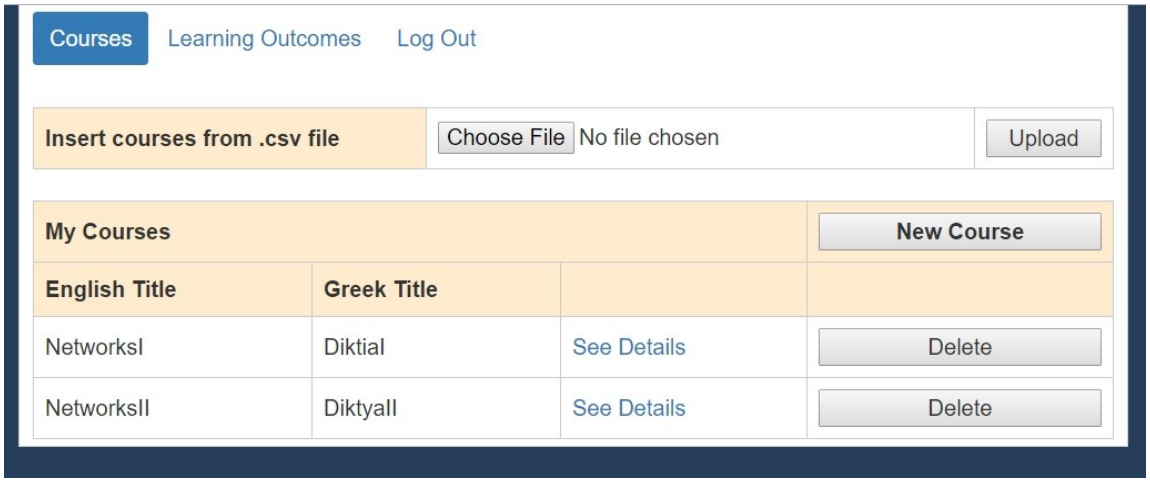


index.jsp - Συνδεδεμένος εκπαιδευτικός

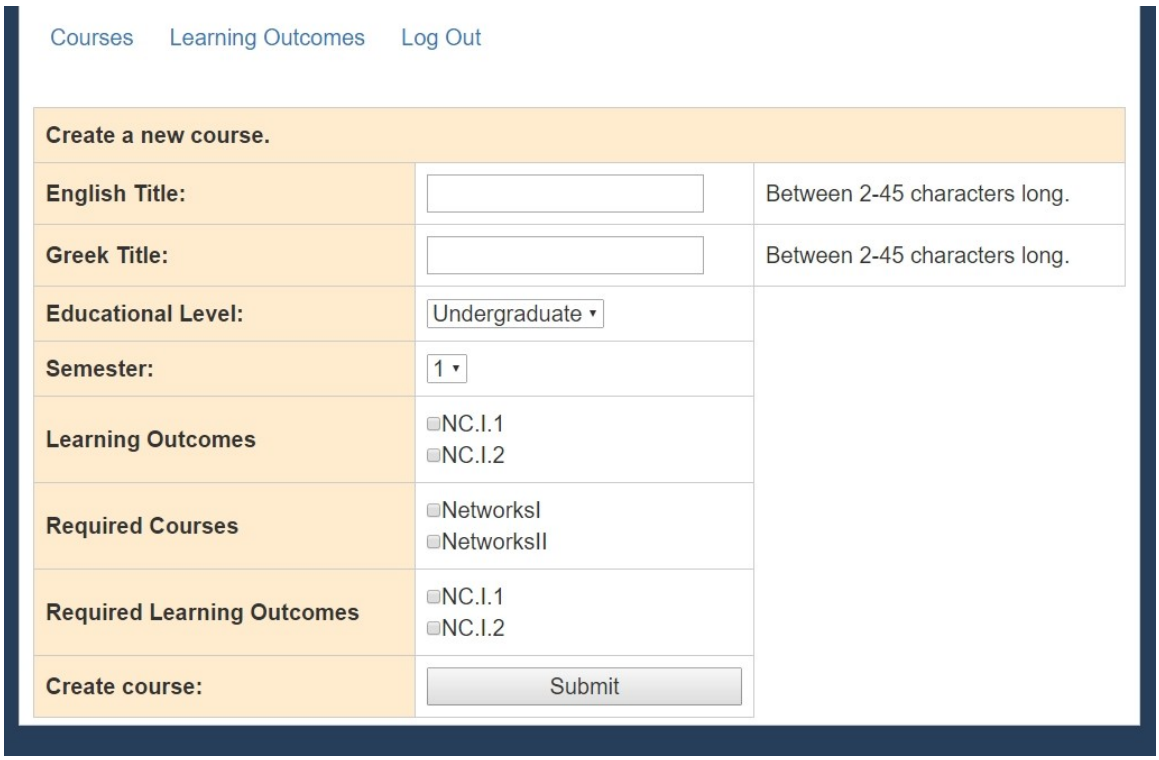
- Δε μπορεί να κάνει τα παραπάνω.
- Μπορεί να δει τα μαθήματα του.
- Να προσθέσει νέο μάθημα (μετάβαση σε **new-course.jsp**) όπου συμπληρώνει τον τίτλο του μαθήματος και επιλέγει τα υπολειπόμενα δεδομένα από dropdowns και checklists προκαθορισμένων επιλογών ώστε να αποφευχθούν λάθη. Γίνεται υποβολή των δεδομένων στο **servlet NewCourse.java** το οποίο λαμβάνει το post και προσθέτει τα δεδομένα σε form το οποίο μετατρέπεται σε XML και αποστέλεται με τη βοήθεια των βοηθητικών κλάσεων στο webservice.
- Να διαγράψει κάποιο από τα μαθήματα του, ξανά με επικοινωνία τύπου application/form-urlencoded.
- Να εισάγει μαθησιακές ευκαιρίες από αρχείο .csv. Αφού επιλέξει το αρχείο στη συνέχεια αποστέλλεται στο **servlet UploadCsv.java** όπου αναλύει και εισάγει το αρχείο σε form και στη συνέχεια αποστέλλει το form σε μορφή XML στο webservice ως λίστα μαθημάτων. Χρησιμοποιούνται οι βιβλιοθήκες **commons-fileupload-1.3.2.jar και commons-io-2.5.jar** για την εκτέλεση του. Οι λάθος εγγραφές αγνοούνται. Το αρχείο πρέπει να είναι την μορφής “,” για αλλαγή πεδίου και “#” για διαχωρισμό σε πολλαπλά πεδία.

Για παράδειγμα: Networks II, Diktia II, Undergraduate, 2, NC.I.2#NC.I.3, Networks II, NC.I.1

index.jsp

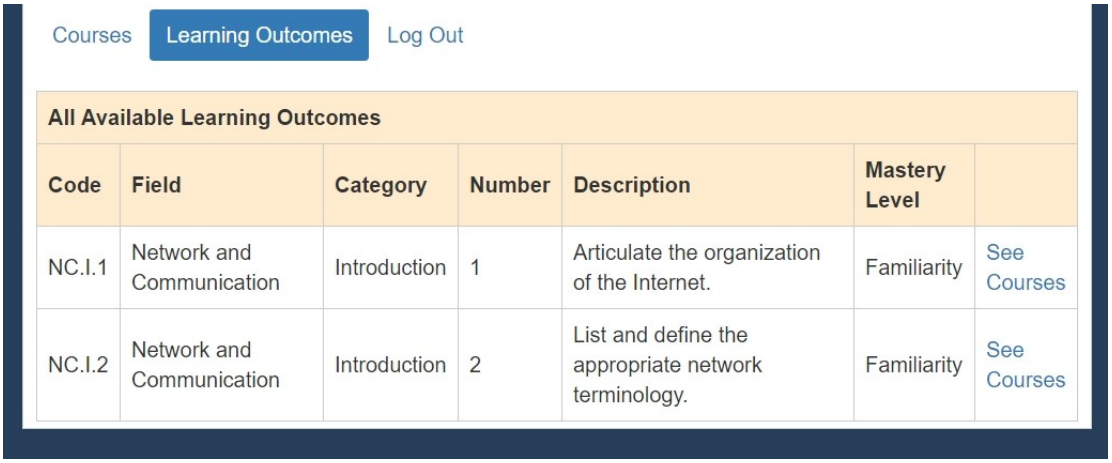


new-course.jsp



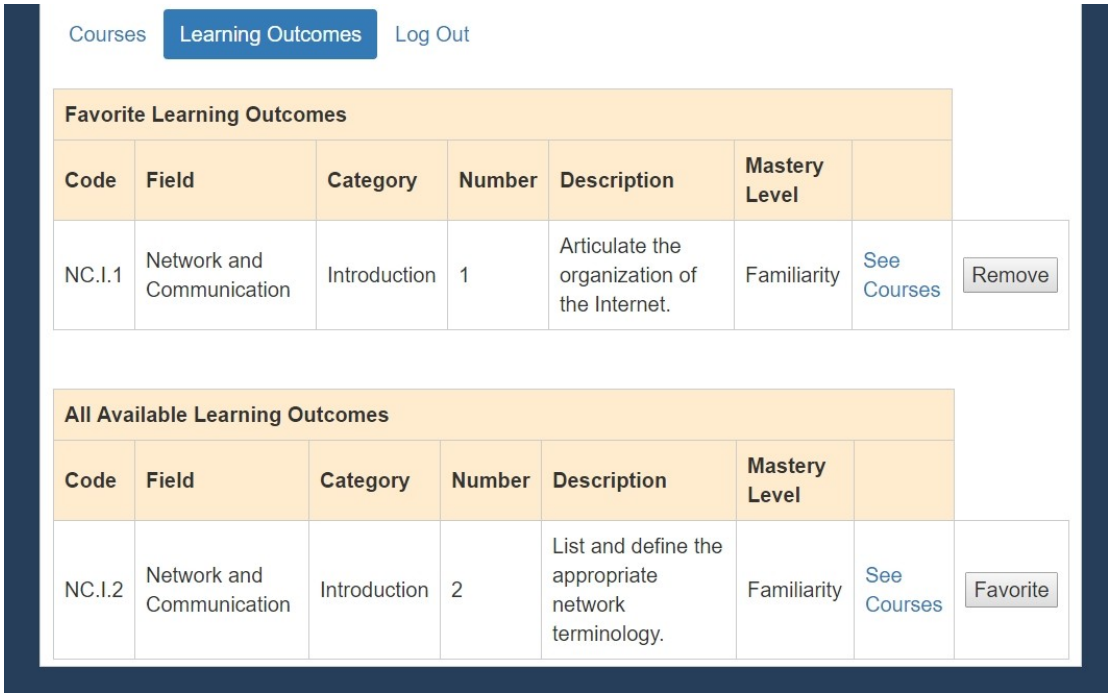
learning-outcomes.jsp – Μη συνδεδεμένος χρήστης

- Μπορεί να προβάλει μαθησιακούς στόχους.
- Μπορεί να κάνει αναζήτηση για μαθήματα που ικανοποιούν τον μαθησιακό στόχο μέσα από το See Courses δίπλα από τον κάθε μαθησιακό στόχο.



learning-outcomes – Συνδεδεμένος ενδιαφερόμενος

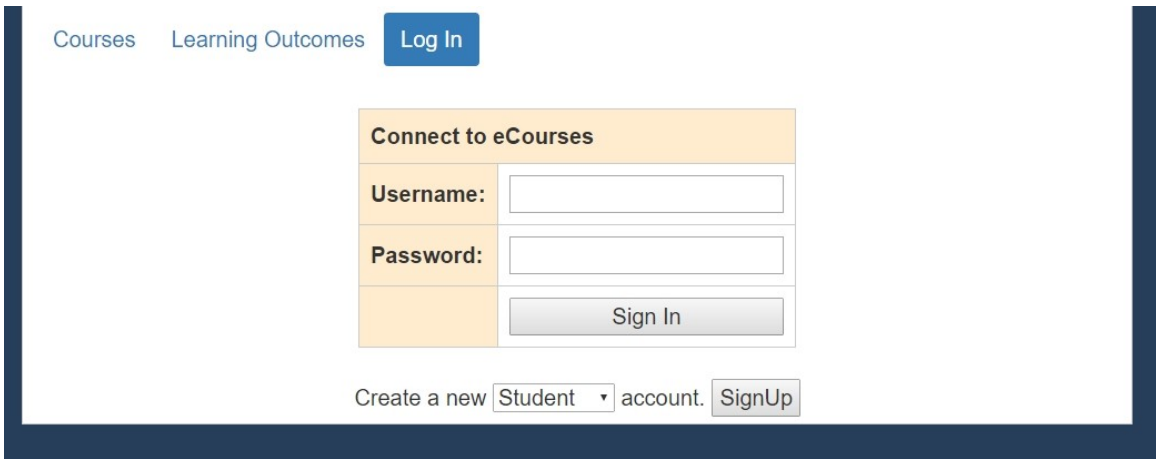
- Μπορεί να κάνει τα παραπάνω και να αποθηκεύει ή να αφαιρεί αγαπημένους μαθησιακούς στόχους από τον πίνακα αγαπημένων του (ίδια μεθολογία με τη διαχείριση αγαπημένων μαθημάτων).



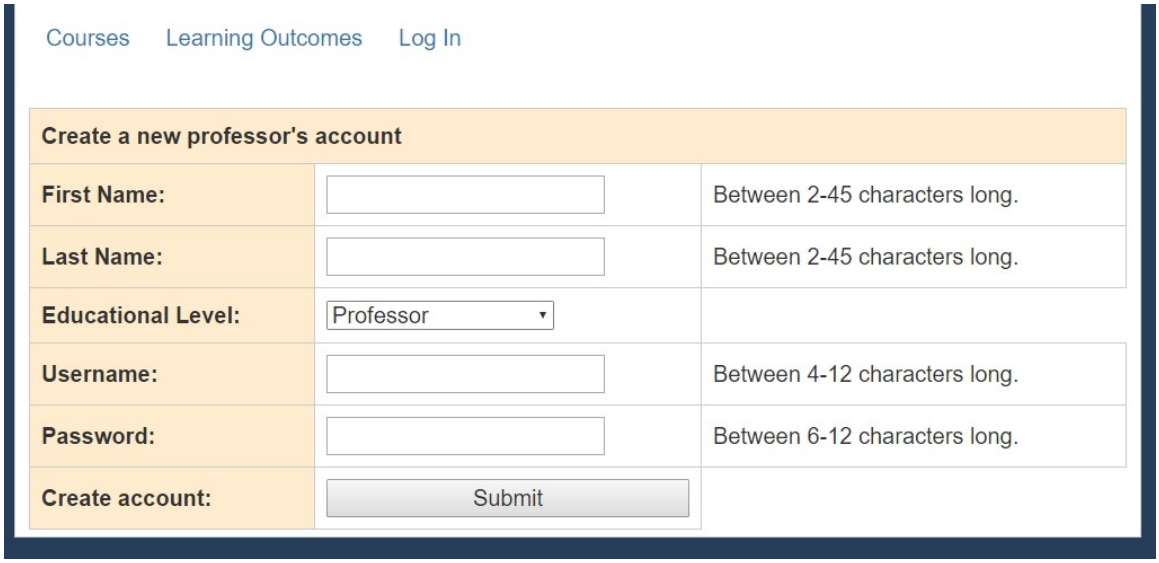
login.jsp – Μη συνδεδεμένοι χρήστες

- Μπορούν να συνδεθούν στο σύστημα, με τη βοήθεια του **servlet Authentication.java** το οποίο αφότου επικοινωνήσει με το αντίστοιχο webservice και λάβει το token δημιουργεί ένα session με τα περιεχόμενά του.
- Μπορούν να δημιουργήσουν νέο λογαριασμό ενδιαφερόμενου ή εκπαιδευτικού με επιλογή από dropdown(μετάβαση στο **signup.jsp**) και συμπλήρωση της αντίστοιχης φόρμας. Τα δεδομένα αποστέλλονται στο **servlet Registration.java** που αναλαμβάνει την επικοινωνία με το webservice. Οι λογαριασμοί των εκπαιδευτικών είναι αρχικά απενεργοποιημένοι μέχρι να ενεργοποιηθούν από τον διαχειριστή.

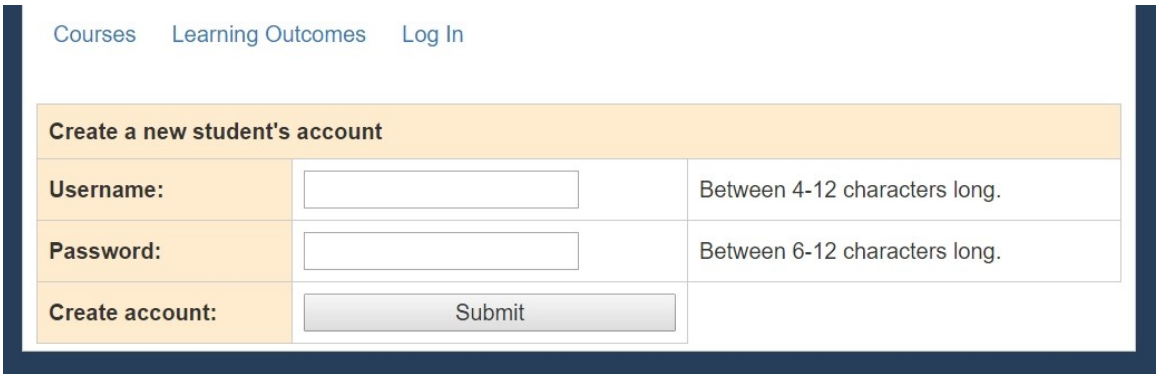
login.jsp



signup.jsp - Εκπαιδευτικός



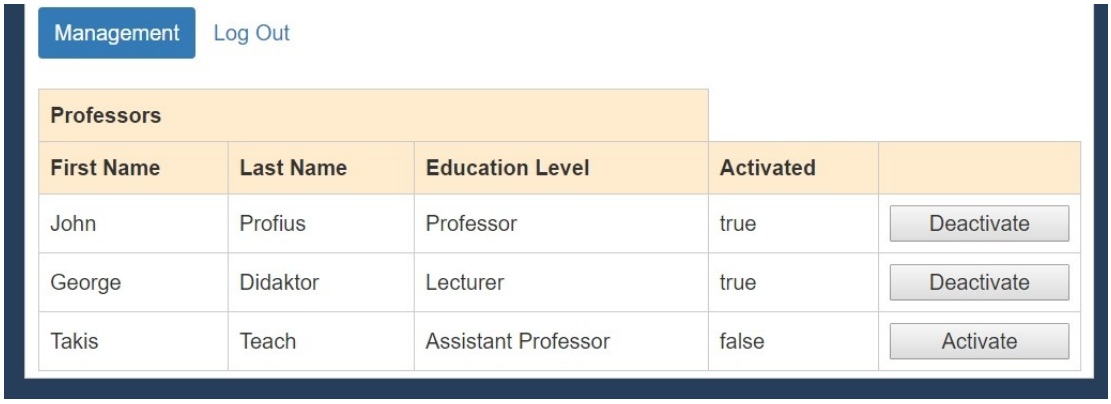
signup.jsp - Ενδιαφερόμενος



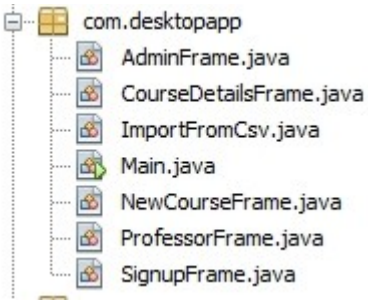
Όταν ένας χρήστης είναι συνδεδεμένος η επιλογή Log In στα nav pills αλλάζει σε Log Out όπου οδηγεί στο **servlet Logout.java,** το οποίο τερματίζει το session και αποστέλλει τα δεδομένα χρήστη στο αντίστοιχο webservice ώστε να αφαιρεθεί το token του από τα ενεργά tokens.

admin.jsp – Συνδεδεμένος Διαχειριστής

- Ανακατευθύνεται αυτόματα ο διαχειριστής.
- Μπορεί να ενεργοποιήσει η να απενεργοποιήσει λογαριασμούς εκπαιδευτικών(ίδια μεθοδολογία με αγαπημένα και διαγραφή μαθημάτων).



4. Desktop Εφαρμογή



Η desktop εφαρμογή προορίζεται για χρήση από εκπαιδευτικούς και διαχειριστές. Χρησιμοποιεί τις ίδιες βοηθητικές κλάσεις που χρησιμοποιούνταν από τα jsp/servlets στην αντίστοιχη web εφαρμογή.

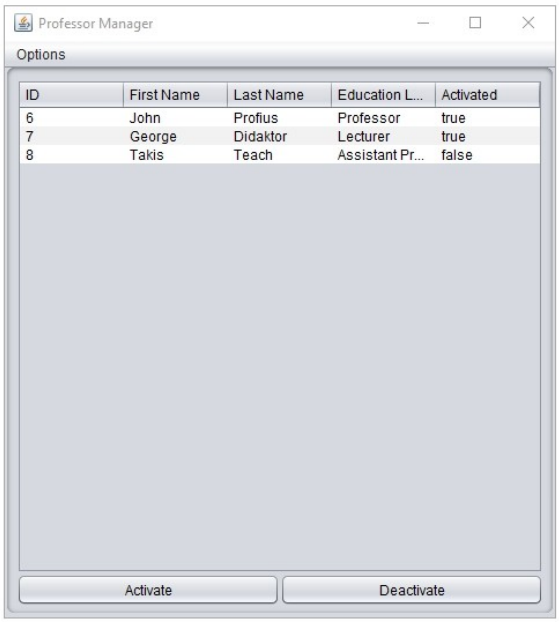
Main.java

- Εκτελείται κατά την εκκίνηση της εφαρμογής και δίνει επιλογές για είσοδο και εγγραφή στο σύστημα.
- Περιέχει static μέθοδο αποσύνδεσης από το σύστημα.
- Περιέχει σε static field το token του χρήστη.
- Εκτελείται συνεχώς στο background, απλά γίνεται invisible το frame όταν γίνει εισαγωγή στο σύστημα.
- Επικοινωνεί με το αντίστοιχο webservice ώστε να γίνει έκδοση ή διαγραφή του token.

SignupFrame.java

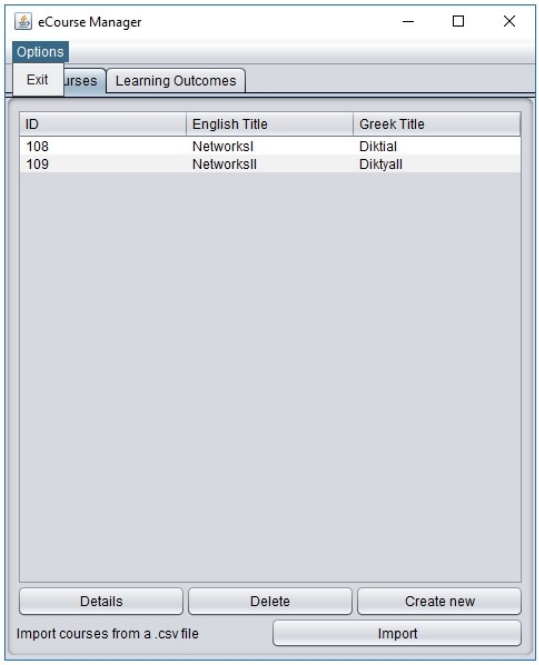
- Singleton object.
- Περιέχει εγγραφή μόνο για εκπαιδευτικό αφού η εφαρμογή δε προορίζεται για ενδιαφερόμενους.
- Επικοινωνεί με το αντίστοιχο webservice για την εισαγωγή εκπαιδευτικού.

AdminFrame.java



- Singleton object.
- Επιλογή από τη λίστα του εκπαιδευτικού προς ενεργοποίηση/απενεργοποίηση.
- Εκτελεί μέθοδο post τύπου application/x-www-form-urlencoded για την αποστολή δεδομένων στο αντίστοιχο webservice για ενεργοποίηση/απενεργοποίηση εκπαιδευτικού.
- Κάθε λειτουργία ανανεώνει το frame για να εμφανιστούν οι αλλαγές.
- Options→Exit για αποσύνδεση και έξοδο από την εφαρμογή.

ProfessorFrame.java

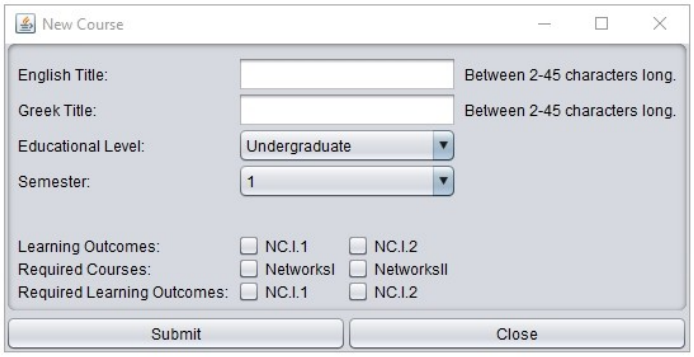


- Singleton object.
- Εκτελεί μέθοδο post τύπου application/x-www-form-urlencoded για την αποστολή δεδομένων στο αντίστοιχο webservice για διαγραφή μαθήματος.
- Κάθε λειτουργία ανανεώνει το frame για να εμφανιστούν οι αλλαγές.
- Options→Exit για αποσύνδεση και έξοδο από την εφαρμογή.
- Επιλογή από τη λίστα του μαθήματος προς διαγραφή/ εμφάνιση λεπτομερειών.

CourseDetailsFrame.java

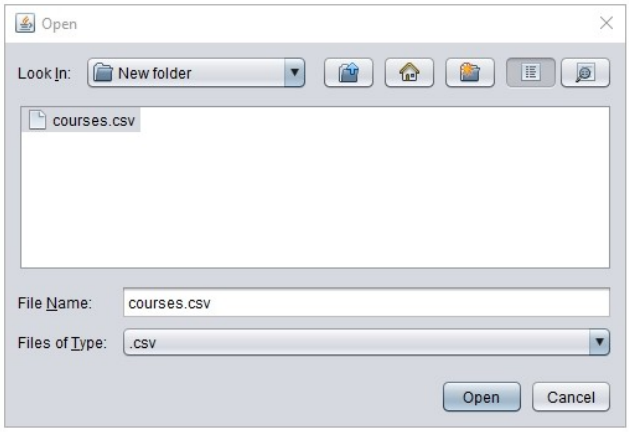
- Μπορούν να υπάρχουν πολλαπλά instances της κλάσης.
- Dispose κατά το κλείσιμο για να μην επιβαρύνει το σύστημα.
- Επικοινωνεί με τα αντίστοιχα webservices για την λήψη λεπτομερειών μαθήματος με βάση τον αγγλικό τίτλο του.

NewCourseFrame.java



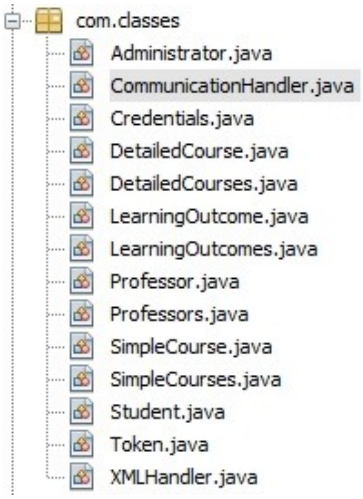
- Singleton object.
- Dispose κατά το κλείσιμο για να μην επιβαρύνει το σύστημα.
- Εμφανίζει δυναμικά τα υπάρχοντα μαθήματα και μαθησιακούς στόχους σε γραμμές των δύο στηλών.
- Εισάγει το μάθημα σε form και επικοινωνεί με το αντίστοιχο webservice για την εισαγωγή του.
- Καλεί νέο instance του ProfessorFrame.java.

ImportFromCsv.java



- Singleton object.
- Dispose κατά το κλείσιμο για να μην επιβαρύνει το σύστημα.
- Φίλτρο για csv αρχεία.
- Εισαγωγή csv με τρόπο ίδιο με του web application.
- Επεξεργάζεται το αρχείο και εισάγει τα δεδομένα σε form για αποστολή στο αντίστοιχο webservice.
- Καλεί νέο instance του ProfessorFrame.java.

5. Βοηθητικές κλάσεις



Χρησιμοποιούνται για την εισαγωγή των δεδομένων και τη μετατροπή τους σε XML. Όταν θέλουμε να αποστείλουμε ένα XML με πολλά DetailedCourse χρειαζόμαστε μια κλάση DetailedCourses για wrapper που αποτελείται από μια λίστα DetailedCourse.

Οι CommunicationHandler και XMLHandler όπως αναφέρθηκε και προηγουμένως είναι μια ομαδοποίηση συχνά χρησιμοποιούμενων μεθόδων με σκοπό την αποστολή request για επικοινωνία με τα webservices μέσω HttpURLConnections και το handling των XML με τους marshallers και unmarshallers. Εμπεριέχονται(οι περισσότερες) σε όλα τα projects (server, webapp, desktopapp) αφού χρησιμοποιούνται από όλες.