CourseNext

Course Management Solution using SpringBoot & MySQL

Biswaraj Kar
Bishwajeet Dey

Spring 2018: CS 5200: Database Management System

Introduction

There is a constant need with educators to have a good software/web-based solution to deliver course content easily and effectively to all students. While there are some software in this domain, like Blackboard, most of them are too complex to manage and make simple tasks complex and time taking.

Hence, in discussion with Prof. Jose Annunziato, we started out this semester to build a simple course-management platform with a programmable web interface. The intention was to allow Instructors to be able host/add various courses on a central online repository, which would further have Pages, Tabs and Widgets to categorize and organize teaching content.

Problem Statement

CourseNext is a next generation course management system for use by faculty and students. The project is proposed and mentored by Prof. Jose Annunziato. A somewhat basic web-based system for a similar purpose was made in the past using the MEAN stack. However, the existing system was fairly basic, web-driven and lacked user/role management, authentication, advanced data-persistence or any kind of relational-schema to interpret and process data quickly.

We had been tasked with replacing major parts of the project, as well as enhancing it with new features using technologies Spring Boot, JPA, Angular.js and MySQL; to be run on a wholly cloud-based infrastructure with Amazon RDS & AWS Elastic Beanstalk. We also added advanced role/user management and stateful data-persistence.

Given the syllabus of CS 5200 which covers most of the above technologies, we are able to deliver a robust and fast web application.

Solution Statement

The proposed solution is a cloud-based solution with data stored at Amazon RDS. The following are the key features of the project, based on client discussions:

- The system has advanced User/Role management which requires users to authenticate their credentials first to use any part of the system.
- Based on the authenticated user's role, the User has varied level of access to the different components of the system, which can be customized as per individual need. E.g. only Faculty or Administrators are able to create Courses and Students will only be able to consume the data.
- The data for the application, including all content and layout, is stored in a relational database (MySQL), which is optimized for quick queries, data-retrievals and proposed data-analytics in the future.
- The data-persistence infrastructure is cloud-based to ensure high-availability (Amazon RDS) and 24x7 access.
- The platform is also available 24x7 on a reliable cloud infrastructure supported by AWS Elastic Beanstalk.

Architecture & Technology Stack

We used the highly flexible and maintainable **M**odel **V**iew **C**ontroller approach to design the solution ground up. Since we wanted a scalable, maintainable solution, we chose SpringBoot as our primary tool. In summary the following was our technology stack:

Front End/UI: Angular.js, Javascript, HTML 5, CSS, XML

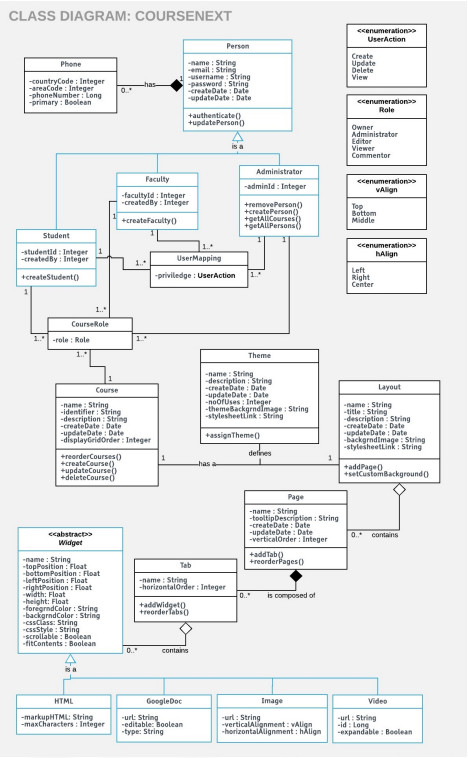
Backend/Application: Java 8, Spring Tool Suite

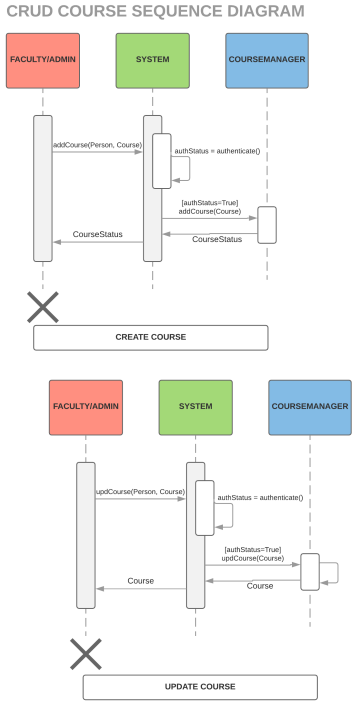
Database: MySQL 5.6.39 hosted on Amazon RDS

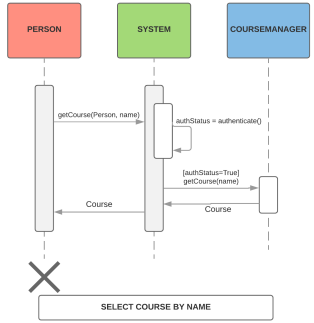
Web Server: Tomcat 8 with Java 8 running on 64bit Amazon Linux/2.7.7 hosted on Amazon Elastic Beanstalk

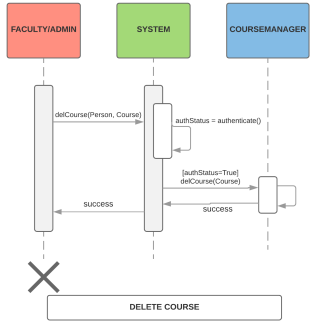
Continuous Integration, Version & Build Management: Circle CI and Github

Data Model


CLASS DIAGRAM: COURSENEXT
Phone
-countryCode : Integer
-areaCode : Integer
-phoneNumber : Long
-primary : Boolean
has
0..*
1
Person
-name : String
+email : String
-username : String
-password : String
-createDate : Date
-updateDate : Date
+authenticate()
+updatePerson()
is a
<<enumeration>>
UserAction
Create
Update
Delete
View
<<enumeration>>
Role
Owner
Administrator
Editor
Viewer
Commentor
<<enumeration>>
vAlign
Top
Bottom
Middle
<<enumeration>>
hAlign
Left
Right
Center
Faculty
-facultyId : Integer
-createdBy : Integer
+createFaculty()
Administrator
-adminId : Integer
+removePerson()
+createPerson()
+getAllCourses()
+getAllPersons()
Student
-studentId : Integer
-createdBy : Integer
+createStudent()
UserMapping
-priviledge : UserAction
CourseRole
-role : Role
Theme
-name : String
-description : String
-createDate : Date
-updateDate : Date
-noOfUses : Integer
-themeBackgrndImage : String
-stylesheetLink : String
+assignTheme()
defines
Layout
-name : String
-title : String
-description : String
-createDate : Date
-updateDate : Date
-backgrndImage : String
-stylesheetLink : String
+addPage()
+setCustomBackground()
Course
-name : String
-identifier : String
-description : String
-createDate : Date
-updateDate : Date
-displayGridOrder : Integer
+reorderCourses()
+createCourse()
+updateCourse()
+deleteCourse()
has a
Page
-name : String
-tooltipDescription : String
-createDate : Date
-updateDate : Date
-verticalOrder : Integer
+addTab()
+reorderPages()
contains
is composed of
<<abstract>>
Widget
-name : String
-topPosition : Float
-bottomPosition : Float
-leftPosition : Float
-rightPosition : Float
-width: Float
-height: Float
-foregrndColor : String
-backgrndColor : String
-cssClass: String
-cssStyle : String
-scrollable : Boolean
-fitContents : Boolean
Tab
-name : String
-horizontalOrder : Integer
+addWidget()
+reorderTabs()
contains
is a
HTML
-markupHTML: String
-maxCharacters : Integer
GoogleDoc
-url: String
-editable: Boolean
-type: String
Image
-url : String
-verticalAlignment : vAlign
-horizontalAlignment : hAlign
Video
-url : String
-id : Long
-expandable : Boolean


CRUD COURSE SEQUENCE DIAGRAM
FACULTY/ADMIN
SYSTEM
COURSEMANAGER
addCourse(Person, Course)
authStatus = authenticate()
[authStatus=True]
addCourse(Course)
CourseStatus
CourseStatus
CREATE COURSE
FACULTY/ADMIN
SYSTEM
COURSEMANAGER
updCourse(Person, Course)
authStatus = authenticate()
[authStatus=True]
updCourse(Course)
Course
Course
UPDATE COURSE


PERSON
SYSTEM
COURSEMANAGER
getCourse(Person, name)
authStatus = authenticate()
[authStatus=True]
getCourse(name)
Course
Course
SELECT COURSE BY NAME


FACULTY/ADMIN
SYSTEM
COURSEMANAGER
delCourse(Person, Course)
authStatus = authenticate()
[authStatus=True]
delCourse(Course)
success
success
DELETE COURSE

Demo Time!