Dreamtitz

created by Dream-girlz

Derek Klatt, Mick Tarsel, Sean Lyons and Palmer Lao

Introduction

This project will be about creating a MOOC. For clarification of terms check the glossary at the end of the documentaaaaation. For more information or to see the code check the site https://github.com/mtarsel/dream-girlz.

Stakeholders

The four main stakeholders are the Students, Instructors, the Administration and the University or Academic Institution.

Primary Actors

The primary actors will be Students and Instructors.

Vision

Our vision for this project is to create a web app for academic institutions. We want to help instructors organize their courses and efficiently distribute course material to large amounts of students.

Current Features

For our first iteration we currently have implemented:

- Admin Panel - Allows Admins to edit the database
- Teacher Panel - The site for teachers to add courses and assignments.
- Backend - The database logic and templates that will be reused
- Student Dashboard - The site for students to enroll, attend classes and submit assignments
- Adding and Dropping Courses - Allows students to enroll and drop classes
- Edit Profile - Users can change their information about their account
- Authentication - Allows users to be distinguished from each other and checks to seee if they are logged in or not

Future Plans

For our second iteration we wish to implement the following:

- Send email activation link
- Instructor dashboard
- Specific course page
- View file uploaded
- Update design

Supplementary Specification

Functionality:

- Logging and Error Handling - Log all errors to persistent storage.
- Security - Admin module and the registration module requires user authentication.

Open Source Project:

- Entire project is Open Source and can be viewed at https://github.com/mtarsel/dream-girlz

Interfaces:

- As a web-based application, Dreamtitz should run on common browsers

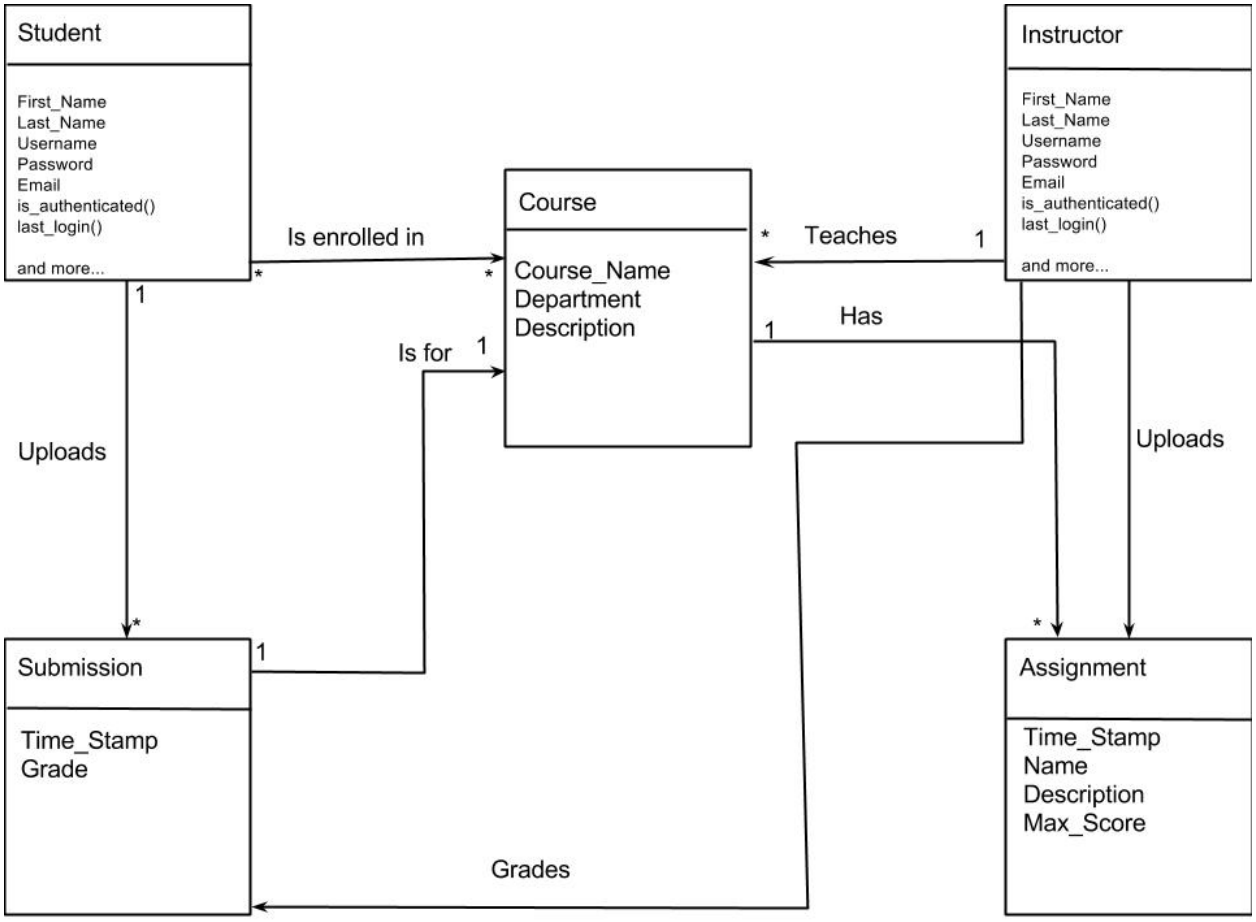
Reliability:

- Hours of Use - Usable 24 hours a day, 7 days a week. Current implementation requires the site to be down during maintenance.

Design Model

The project is broken up into three parts. The first part is the basic MOOC. This includes the basic homepage where users can see a list of all the courses, register and login. The second part is the Student App. This is where students get a unique dashboard where they can enroll in courses. When a student is enrolled in a course they will have the ability to attend class by watching lectures, uploading completed assignment and viewing and interacting with other course material. The third and last part is the Instructor App. This is the dashboard for the Instructors where they can create courses. Once the Instructor has created a course, they can upload material and assignments for the students to view and give them grades.

Domain Model



Use Cases

Student Class Registration

Scope: MOOC Project for Dream Girlz

User Goal Level: Student successfully registers for a class

Sub-function Level:

a. Student User goes to domain name, website

b. If Student is a new user – register for an account

c. If Student is a returning user – they select which class they want to enroll

Primary Actor: Website user

Stakeholders:

-Student wants to enroll for a class

-Instructor needs students to enroll to have a class

Preconditions: User must be logged in

Success guarantees: Class is saved in student account database. Instructor has one more student enrolled in class

Main Success Scenario: Student registers for a class and is enrolled in a new class

Extensions: Student enrolls in several classes, or student enters invalid information upon registration

Special Requirements: User authorization

Technology and Data Verification: New user must enter valid information in order tonj jjenroll

Attending Class

Scope: MOOC Project for Dream Girlz

User Goal Level: Student successfully registers for a class

Sub-function Level:

A. The student will click on the video lectures and watch the latest one
B. The student will click the homework section and submit the homework is due
C. The student will take the quiz on the current lecture
D. The student will download and start the new homework assignment

Primary actor: Students

Stakeholders and interests:

- The Academic Institution that owns the site – wants the students to be able to take classes online as though they were in the classroom
- Students – get the classroom experience while having the convenience of being anywhere
- Instructors – give the classroom experience while not being in a classroom

Preconditions: The Student must be logged in

Success guarantees: The instructor will give a great education to the student at his own convenience

Main success scenario: The Student logs on and takes the class at his or her convenience

Extensions (or alternative flows):

- The Student attends another class
- The Student can't watch the movie
    - The Teacher reloads the lecture
    - The Student relogs
    - The Teacher emails the video to the Student
- The Student can't upload the homework
    - The Teacher reopens the link
    - The Student relogs
    - The Student emails the homework to the Teacher
- The Student can't take the quiz or test
    - The Student relogs
    - The Teacher reactivates the quiz or exam

Instructor Registration

Scope: MOOC Project for Dream Girlz

User Goal Level: Instructor successfully registers for the site

Sub-function Level:

a. Instructor goes to domain name, website.

b. If Instructor is a new user – instructor will be forced to register before logging **in.**

c. Instructor is redirected to registration form

d. Instructor fills out and submits registration form

e. Instructor is redirected to his/her dashboard

Primary Actor: Website instructor

Stakeholders:

-Instructor needs to register in order to teach a class

-Students require an instructor to teach the class

Preconditions: User (Instructor) is not already registered

Success guarantees: Instructor profile is saved in the database

Main Success Scenario: Instructor registers for the site and can create/administrate classes

Extensions: N/A

Special Requirements: Site admin permission

Technology and Data Verification: New user must enter valid information in order to register.

Use Case UC1: Create Course

Scope: MOOC Project for Dream Girlz

Primary Actor: Instructor

Stakeholders and Interests:

● Instructor: Wants easy to use application for creating courses, time wasted dealing with the system's shortcomings takes away from other projects/research

● Student: Needs to be able to see and begin work on courses in a timely manner

Preconditions: Instructor is registered, identified, and authenticated

Success Guarantees / postconditions: Video lectures, syllabi, notes, slides, and any other teaching materials are uploaded and available to the student.

Main Success Scenario:

1. Instructor creates a new course on the MOOC web application.
2. Instructor chooses the course material to be uploaded
3. System accepts course material
4. Instructor configures when course material becomes available
5. System saves all configuration and course material

Extensions:

a.) If internet connection is lost or System fails:

Ensure that all course materials or other information have been saved so that the Instructor can easily pick up where he/she left off later.

1a. Instructor uploads a list of students that have preregistered

2a. Instructor attempts to upload invalid filetype or file that is too large

1. System notifies students once the course is created and course materials have been uploaded

1. System rejects the file in question and notifies the Instructor appropriately

Special Requirements: None

Technology and Data Variations: None

Test Case

All test cases first require starting the site and logging into the admin panel and opening another tab with the homepage.

Registering as a new user: Click the register button. Fill the required fields and hit submit. Check the admin panel to make sure the user was created and the follow the test case for logging in.

Logging In: Either create a user through the admin panel or first follow the test case for registering a new user. Click the login button. Fill in the username followed by the appropriate password and click submit. Make sure the url changed and it shows the user's dashboard.

Enrolling in a course: Create a course and user in the admin panel and sign in as the user. Click courses and click the enroll button next to the course. Make sure the user is listed with the course on the admin panel.

Submitting an assignment: Create a course, assignment and user and have the user enroll in the course through the admin panel. Login as the user and go to courses and click the course. Go to submissions and choose file and click submit. Make sure the file appears in the list of submissions on both the url and in the admin panel.

Glossary

- MOOC - Massive Online Open Course
- Dreamtitz - DREAMing of Teachers Incorporating TechnologieZ
- Django - Python framework to design web applications
- App - Group related functionality focused on one section of the site
- Git - Version control
- Course - An online class
- Student - A user who takes classes
- Instructor - The teacher or professor of a course
- Grade - A number marking in a course that a student receives
- Dashboard - User interface that caters for the type of user
- Assignments - Material Instructors assign to the students of a course
- Submission - Turned in completed assignments from a student