Coursemeleon

Design Document

Bilal Areeb Baig

Zachary Bryant

Mitchell Demerly

Rujula Kapoor

Lauren Lum

Table of Contents

- **Purpose**
  - Non-Functional Requirements
  - Functional Requirements
- **Design Outline**
  - System Components & Purpose
  - Component Interactions
  - High-level UML diagram
- **Design Issues**
  - Non-Functional Design Issues
  - Functional Design Issues
- **Design Details**
  - Class Diagrams
  - Class Interactions
  - Sequence Diagrams
  - UI mockups

Purpose

A course webpage is an important resource for both students and the professor, providing students with direct access to class information and professors with an important means of communication. These websites, however, tend to be hosted on the platform an individual instructor is comfortable with; users are expected to keep track of many different websites for each class they are taking in order to stay up-to-date with information. WIth no standard existing, site functionality and organization can vary.

Coursemeleon offers an efficient and user-friendly way for professors to create course webpages. By providing a web application that takes course information input by a professor and organizes it intuitively on a webpage, professors will be able to follow a standard design without investing more time and technical knowledge. The application's customizable features gives professors flexibility with the information they choose to publish. Students, on the other hand, will be able to access information like course calendars, announcements, and grades through a single service.

Functional Requirements

As a student...

1. I would like to view course information on a course site so I can stay up to date about the course
2. I would like to view past course information so that I can gain an understanding of the course background
3. I would like to see course information in an organized way so that I can effectively make use of the course website in a timely manner
4. I would like to view professor announcements on a single page so that I can know when an important announcement is made regarding the course
5. I would like to have access to a class calendar so that I can see information about upcoming assignments and exams
6. I would like to be able to export a course calendar onto a calendar service I use so that I can combine the course information with other activities
7. I would like to have access to course resources so that I can have all of the necessary files and information for the course.
8. I would like to view information of instructors so that I can contact them if I need assistance.
9. I would like to see course statistics so that I can determine my standing in the course and approximate my final grade.
10. (If time allows) I would like to be able to communicate with others in the course so that I can speak with peers about course issues or assignments.

11. I would like to see a list of course information I recently viewed so that I can navigate the website more effectively.
12. I would like to be notified of recently updated course information so that I can be aware of any changes.
13. I would like to download course resources so that I can access resources offline.
14. I would like to see all relevant course links so that I do not have to navigate many websites.

As a professor…

1. I would like to be able to make quick changes to the course site so that students will see the most updated information.
2. I would like to create an organized course site without worrying about formatting so that I can spend more time on course content.
3. I would like to use a simple, step-by-step process to insert information onto my course site so that I can make a course site efficiently.
4. I would like to display information about the course instructors so that students are available to contact them.
5. I would like to import a course calendar for my students to see so that students will always be aware of assignments or important dates in my class.
6. I would like to be able to share resources for my students to use so that they will have course information readily available.
7. I would like to upload course statistics for my students so that sharing information is fast and intuitive.
8. I would like to graphically display course statistics for my students so that they have a better grasp of my course content..
9. I would like to upload course resources for users to download so that they can access the materials offline.
10. I would like a way to announce information to my students so they are up-to-date with my course.
11. I would like to decide what kind of information is included in my course site so that I can control what my students have access to.
12. I would like to decide who can modify my course site so that course integrity is maintained.
13. I would like to customize the layout of my course site as I see fit so that I can decide how information is displayed.
14. I would like to add custom pages to my course site so that I have more flexibility with information and design.
15. I would like to add custom functionality to my course site so that my course is not restricted by the template.
16. I would like to answer questions and communicate with my students with a course chat so that my students are better informed about changes in the course.
17. I would like to be able to reuse the website for the next year so that I can save time and not have to redo my work.

18. I would like to theme my website so that it matches my school brand guidelines.
19. I would like to hide or publish my course page as I see fit so that I can finish customization at a later date.

Non-Functional Requirements

Usability

The interface should be easy to navigate and simple enough to enable quick start-up time for course pages. It should be flexible enough to seamlessly integrate additional functionality. The result generated from the Coursemeleon should have standardized navigation so that students are are not confused about where relevant information is. We also want to maintain accessibility for various devices and user abilities. There are around 2400 instructional employees at Purdue; at expected adoption rates, it can be predicted that around half will create accounts within three years. Students will not have accounts but are assumed to be checking in to these course pages frequently; Coursemeleon will accommodate traffic from universities with over 40,000 students.

Scalability

Coursemeleon should be able to store all customization of regular templates. Our biggest challenge will be handling the data uploaded such as PDFs, pictures, and other data. Most of the workload will be on the frontend (i.e. rendering is done by the user) using Vue.js. We will store the data in a NoSQL database hosted by Firebase to take advantage of the loosely enforced structure of data. Our backend will be written using Express.js, and will enable our RESTful API. The frontend will request data from this API. The server should perform its necessary duties and respond with data in under 1 second, and should be able to handle 10,000 simultaneous requests. Server uptime should be 24 hours, 7 days a week.

Security

The login system will be based on sending a random string of words as a password to the email. This reduces our risk of storing and encrypting passwords and relies on the security of the user's email service. Integrity of course pages will be maintained through this login by via a permissions system. Unless otherwise whitelisted, students will be able to access any course page.

Our backend will be secured using Helmet for Express, which blocks common vulnerabilities such as cross-site scripting. Our backend will also take care of sanitization to prevent SQL injection. All modify requests are subject to authentication by the server, and courses with whitelisting enabled will check GET requests as well.

Design Outline

Coursemeleon is a web application that generates formatted web pages based on user input. It will use a client-server model in order to effectively receive and display data and will use a single server to handle numerous clients.

System Components & Purpose

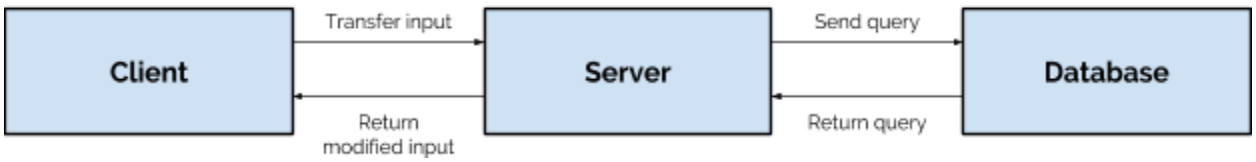
1. Client
    - All user interactions deal with the client
    - The client will allow users to input information
    - The client will display the styled webpage
    - The client sends login information to server for validation

2. Server
    - The server will transfer course information to the database
    - The server will send queries to modify or add information in the database when requested by client
    - Server will return requested information to client
    - The server verifies login information with database and returns results to client

3. Database
    - The database will store user information so users have exclusive and secure access to their course page
    - The database will hold all course information that will be displayed on the final webpage
    - The database will deliver information when requested by the server

System Overview and Component Interactions

The client must first enter any desired information which will be collected by the server. Requests include adding, deleting, or modifying existing data. The server will then send a query to the database in order to implement the requested change. After performing the query, the database returns the results to the server. Finally, the server delivers the finished request to the client.



Design Issues

In this section we will cover design issues that we have encountered while planning as well as justifications for our solutions.

Non-Functional

1. Which front end framework should we use if any?
    - React.js - A beginner can get started with React in a couple hours. Documentation is thorough but not as clear as Vue. Uses JSX and components to simplify the development process. Although react has a special syntax that requires more getting used to. React is maintained by Facebook though, so long term support is not a worry.
    - **Vue.js** - Can be loaded as a resource, meaning the whole library can be used without building. In general, Vue is a no-nonsense framework. Questions in the documentation are answered directly. Vue allows for JSX, which lets us write html/css/js in one single file. This will greatly reduce the complexity of our src folder. Vue is also a bit faster than other options. Vue is maintained by 23 developers but is only growing.
    - None - This will make web development much slower since we will have to do a lot more work to make code consistent and general.
2. Which back end framework should we use if any?
    - **Express.js** - Quick to learn and get a product working. Express is well documented with examples and tutorials, and is flexible for our needs. There is a lot of support available from the community in the form of NPM packages. However, Express doesn't hold hands for new developers and often leaves things like security as optional.
    - PhP - Php is simple to get running and makes use of OOP. It also has a lot of history and support from big companies. However, the language is very inconsistent and at times confusing to use.
3. Which CSS framework should we use if any?
    - Bootstrap - One of the most popular CSS frameworks out there. We would have no problem getting started with this.
    - **Vuetify** - Has a lot of support from companies and community. This framework is actively maintained and follows material design standards. It is also designed specifically for Vue projects, so it integrates with the framework very easily.
4. Which database management system should we use?
    - **Firebase** - Firebase is NoSQL which allows for loose database structure using JSON. It is also in the cloud, so issues and uptime are managed by a different company.

    - MySQL - Great because it would be hosted and have no network latency issues. However the rigid structure of relational databases will slow us down during development. Any issues are on us to fix.
5. How should professor pages be hosted?
    - Centralized - This means everyone accesses the same website across all universities, and all data is managed by us. This introduces scalability and data integrity concerns for us. The likely number of connections at any time could be in the hundreds of thousands.
    - **Per university** - Each university can host their own "instance" of Coursemeleon, and all data is managed by the university. This reduces the likely range of concurrent connections from hundreds to tens of thousands,

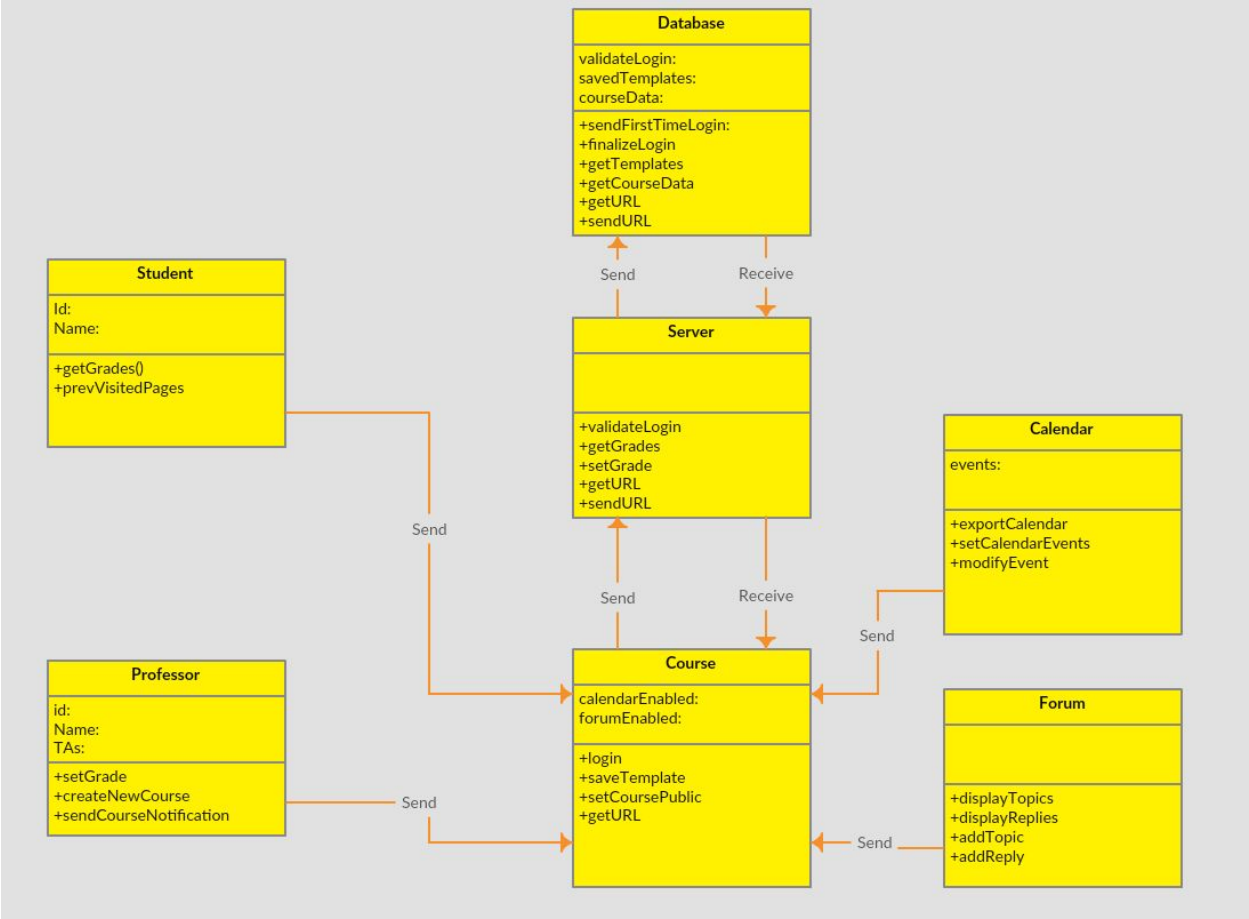
Functional

1. Should we restrict file-type uploads?
    - Yes - In order to develop a secure website, potentially harmful files such as executables should not be allowed to be uploaded. Though this would restrict some ability of the website, it would increase security.
    - **No** - Coursemeleon caters to professors of all subjects, and therefore we need to support all file types. For example, executable file types may need to be uploaded by computer science professors. Security measures can be taken to ensure that no file is executed on the server.
2. How should users authenticate?
    - Username & password - This method is the standard and traditional way to authenticate users into a website. Users would be able to choose a custom username and password. However, in this method we would be responsible for securely storing and validating login information.
    - **Username and random key sent to email**- We choose this so that we don't have to worry about securely storing user passwords, and also so that our users don't need to remember yet another password. They request a login, and then paste a randomly generated string of words from an email as the password.
3. What degree of customization should we allow while building pages?
    - No customization - While this would force professors to adhere to a standard, it would also take away their flexibility with their own website. Different classes require different information; by catering to all professors of all subjects, Coursemeleon should allow enough customization to properly represent all course content.
    - **Basic customization** - This would give professors the option to insert limited custom elements into the site and choose select preset elements to disable or enable. This option allows for limited customizability of the predesigned template, as well as the freedom to implement their own custom additions.

- Full customization - Full customization would allow for professors to have full control over their website and add any additional features they may wish. However, this would be counterintuitive to the platform; if a professor has the technical knowledge to add new elements and fully customize their site then they are outside of the target audience.

Design Details

Class Diagrams



Class Interactions

The classes in our class design are based on different objects used by the web application; the diagram also includes some of the functions and values each object will have.

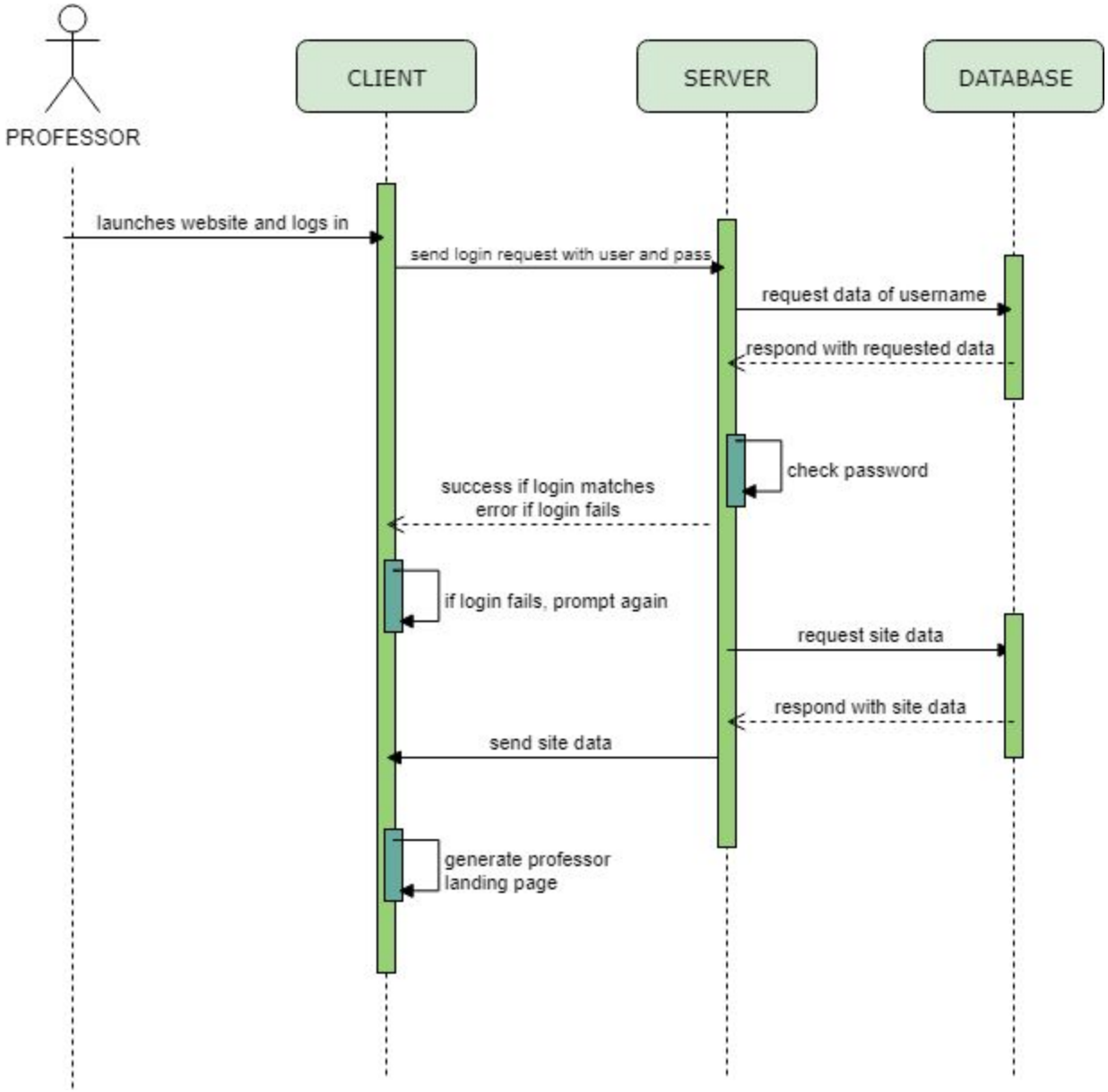
- Course
    - A professor can:
        - Enable a calendar

        - Enable a forum
        - Receive requests from professors
        - Send request to server
        - Get server response
        - Receive requests from students
        - Send request to server
        - Get server response
- Professor
    - A professor can:
        - Set grades
        - send course notifications
        - create new template
        - login
- Student
    - A student can:
        - request grades
        - navigate pages
        - login
- Calendar
    - a student can:
        - view calendar
        - export calendar

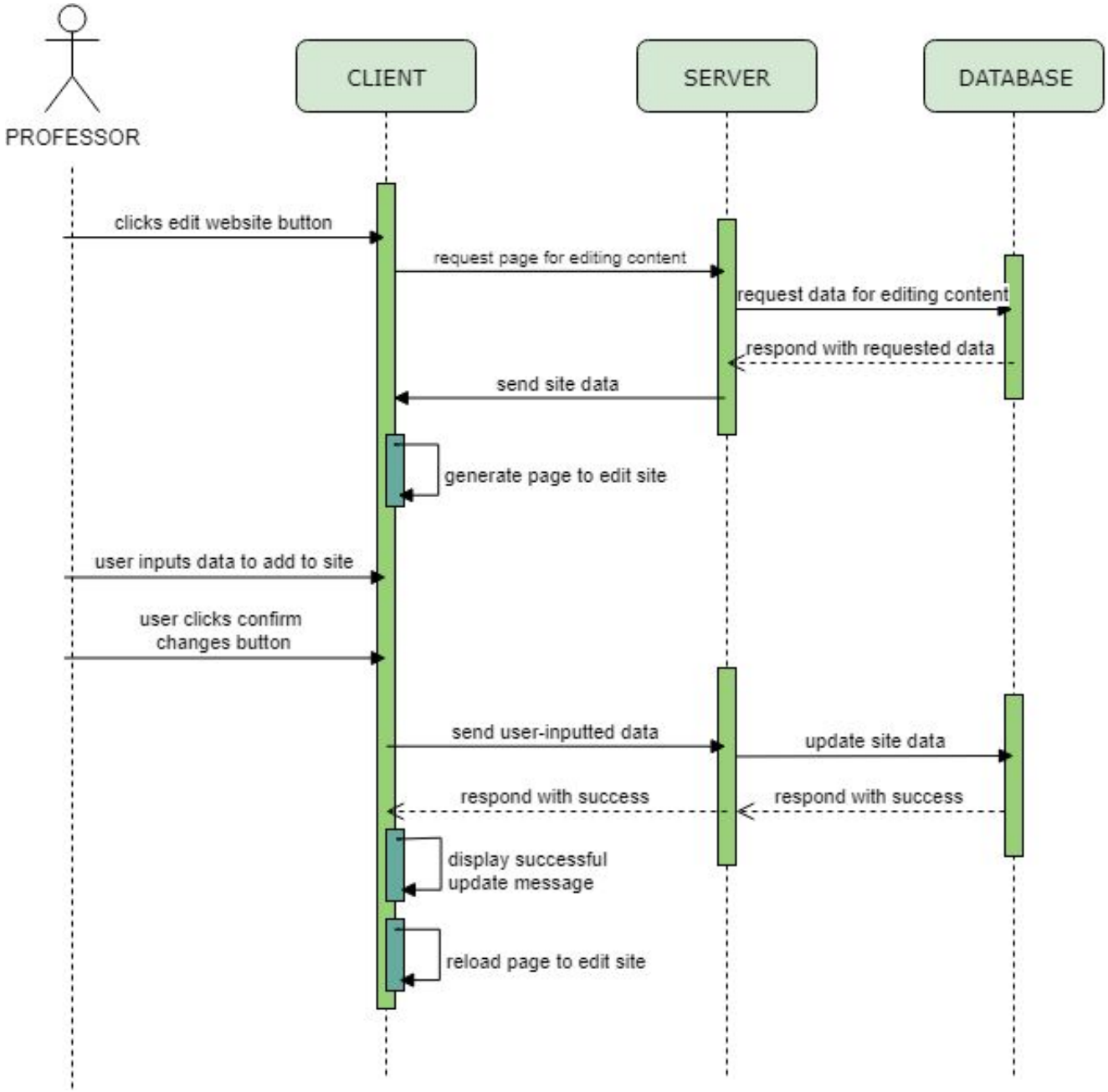
    - a professor can:
        - view calendar
        - set calendar events
        - modify events
- Forum
    - a student can:
        - view forum topics
        - view forum replies
        - add forum topics
        - add forum replies
        - a professor can:
        - view forum topics
        - view forum replies
        - add forum topics
        - add forum replies

Sequence Diagrams

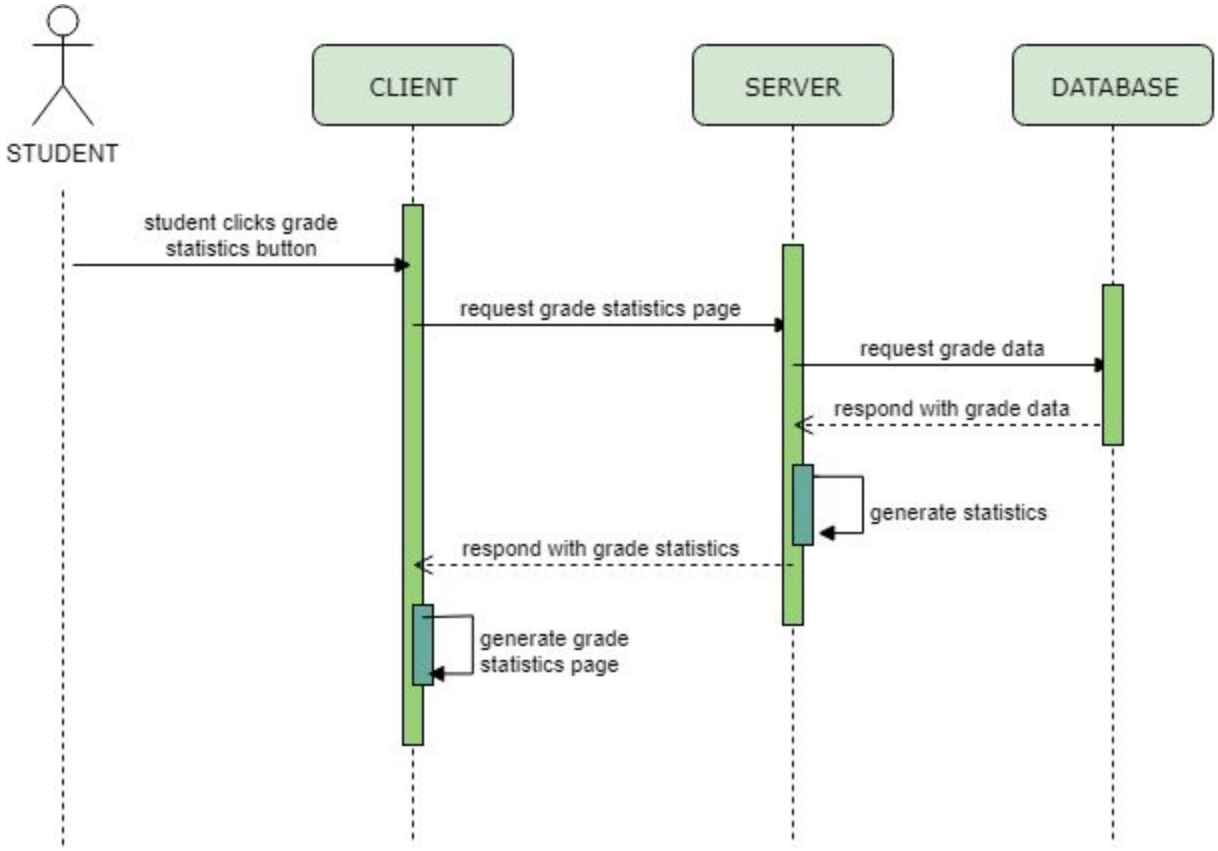
Sequence of events when a professor logs in to the site



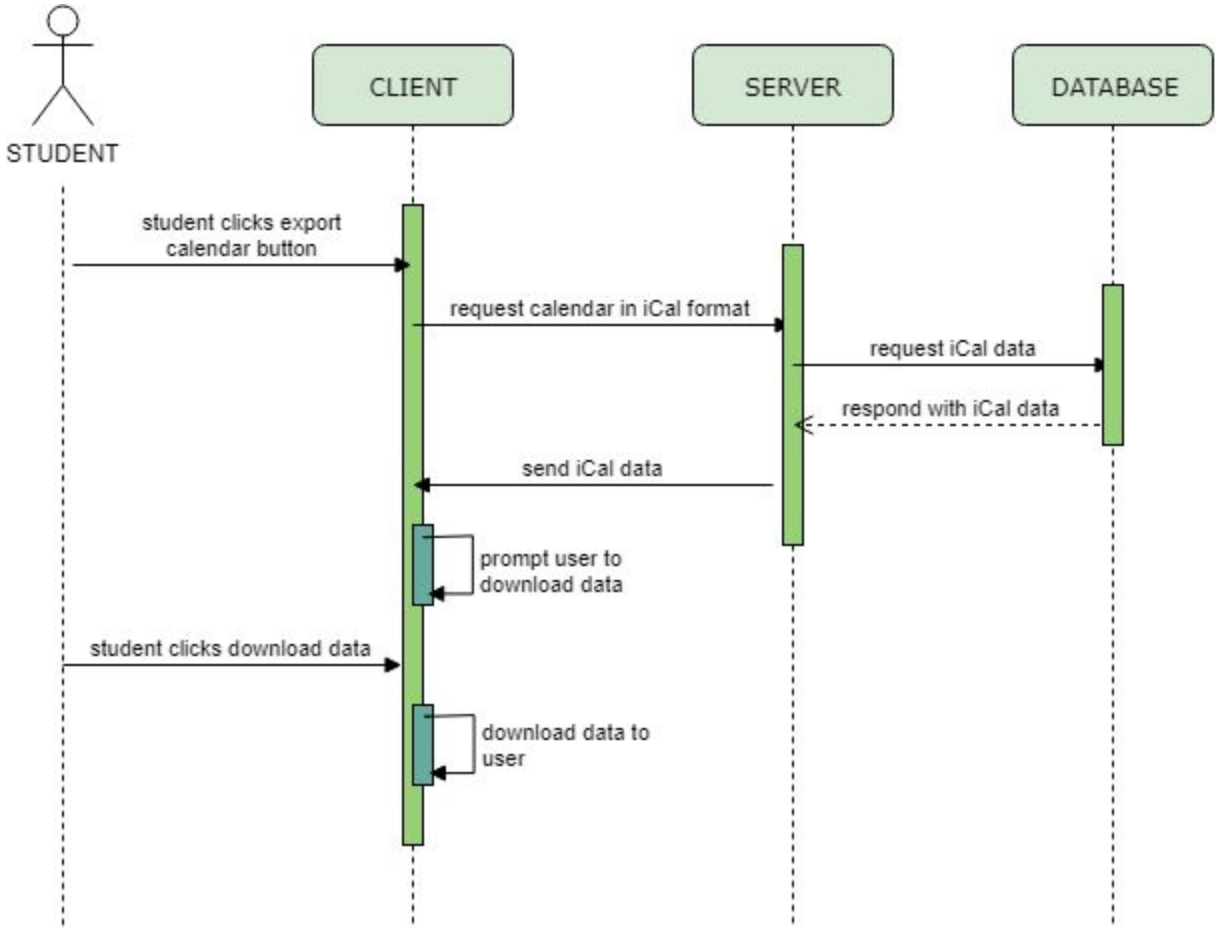
Sequence of events when a professor requests to edit their site



Sequence of events when student loads the grade statistics page



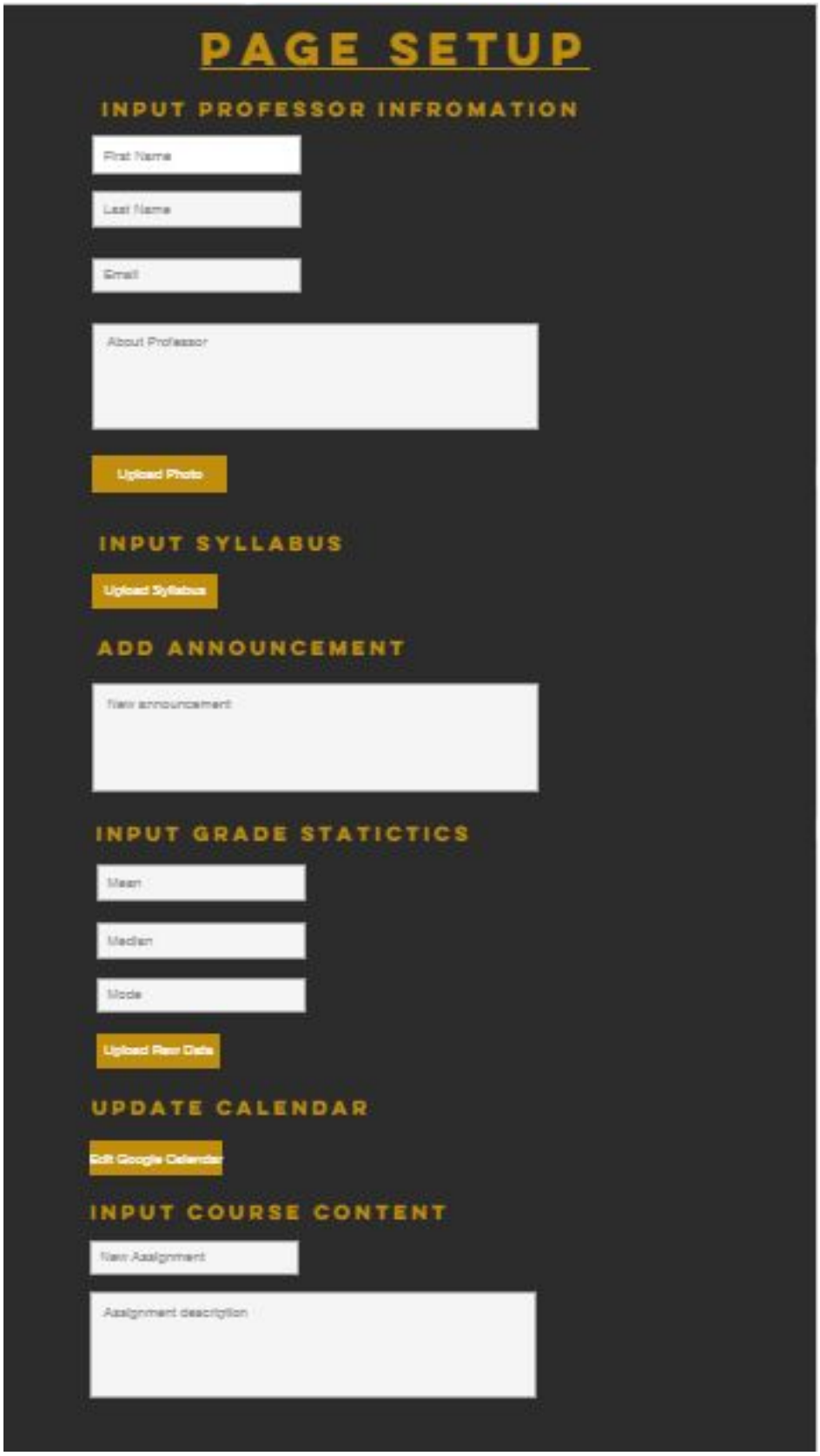
Sequence of events when student exports calendar information



UI Mockups

Homepage

Page Setup


PAGE SETUP
INPUT PROFESSOR INFROMATION
First Name
Last Name
Email
About Professor
Upload Photo
INPUT SYLLABUS
Upload Syllabus
ADD ANNOUNCEMENT
New announcement
INPUT GRADE STATICTICS
Mean
Median
Mode
Upload Raw Data
UPDATE CALENDAR
Edit Google Calendar
INPUT COURSE CONTENT
New Assignment
Assignment description

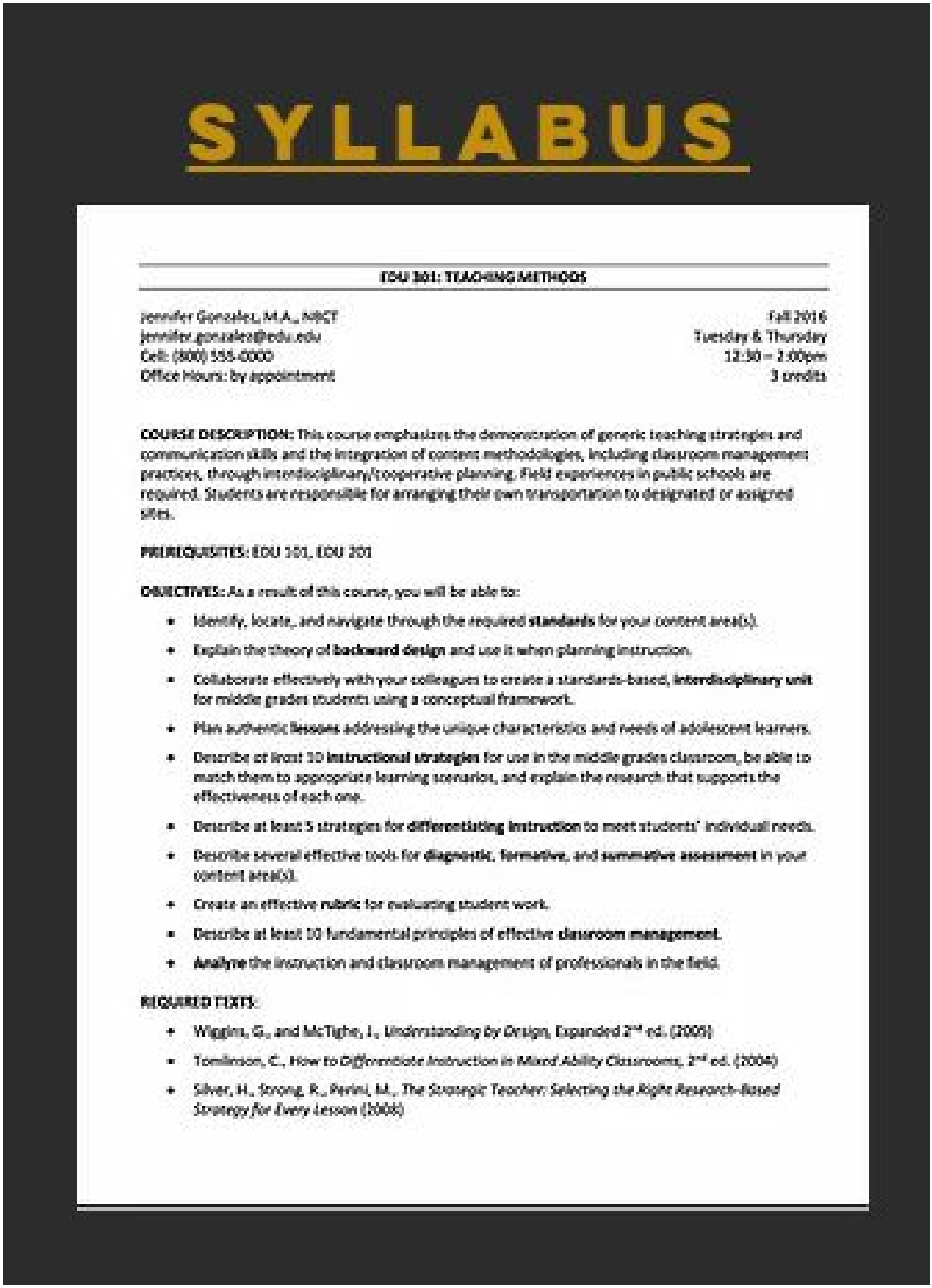
Grade Analytics

Course Content

Calendar


CALENDAR
May 2016
today
month
week
day
SUN
MON
TUE
WED
THU
FRI
SAT

Syllabus



SYLLABUS

EDU 301: TEACHING METHODS

Jennifer Gonzalez, M.A., NBCT
jennifer.gonzalez@edu.edu
Cell: (800) 555-0000
Office Hours: by appointment

Fall 2016
Tuesday & Thursday
12:30 – 2:00pm
3 credits

COURSE DESCRIPTION: This course emphasizes the demonstration of generic teaching strategies and communication skills and the integration of content methodologies, including classroom management practices, through interdisciplinary/cooperative planning. Field experiences in public schools are required. Students are responsible for arranging their own transportation to designated or assigned sites.

PREREQUISITES: EDU 101, EDU 201

OBJECTIVES: As a result of this course, you will be able to:

- Identify, locate, and navigate through the required **standards** for your content area(s).
- Explain the theory of **backward design** and use it when planning instruction.
- Collaborate effectively with your colleagues to create a standards-based, **interdisciplinary unit** for middle grades students using a conceptual framework.
- Plan authentic **lessons** addressing the unique characteristics and needs of adolescent learners.
- Describe at least 10 **instructional strategies** for use in the middle grades classroom, be able to match them to appropriate learning scenarios, and explain the research that supports the effectiveness of each one.
- Describe at least 5 strategies for **differentiating instruction** to meet students' individual needs.
- Describe several effective tools for **diagnostic, formative,** and **summative assessment** in your content area(s).
- Create an effective **rubric** for evaluating student work.
- Describe at least 10 fundamental principles of effective **classroom management.**
- **Analyze** the instruction and classroom management of professionals in the field.

REQUIRED TEXTS:

- Wiggins, G., and McTighe, J., *Understanding by Design*, Expanded 2nd ed. (2005)
- Tomlinson, C., *How to Differentiate Instruction in Mixed Ability Classrooms*, 2nd ed. (2004)
- Silver, H., Strong, R., Perini, M., *The Strategic Teacher: Selecting the Right Research-Based Strategy for Every Lesson* (2008)