Design Document - Open Alexandria

Team 4

Xu Han, Brian Chan, Andy Chen, Hongda Zeng

Purpose

Many file sharing and education sites cost money and are close source. We want to provide a free, open-source, secure, in-depth education sharing platform where users can help each other in academia or even contribute code to the site.

Almost every college student has a class they will search up for help with their homework and project. Often students will find the information they need online but access to those information is not open. Many file sharing and education sites cost money. Websites like Course Hero or Chegg charge a premium or require subscription and often provide free users with a limited preview of relevant documents or discussions. This often results in locked features and hindered usability for students unable to afford payment. There is also a concern that these websites are profiting from student's contributions and that student information are being sold and traded behind their back.

Open Alexandria will be a free and open solution what will allow users to share documents and have discussions without charging any user for access. Our open sourced nature will allow inspection of the code to ensure security and that private information is not being kept. Open Alexandria can also be self-hosted to eliminate dependency on a particular host.

User Stories

As a user:

I want to post questions
I want to answer questions
I want to add a course
I want to upvote/downvote questions/answers
I want to upload documents
I want to access shared documents
I want to delete my uploaded documents
I want to review documents uploaded
I want to be able to contact the developer to report any issues
I want to automatically retrieve a list of classes as I type the course name
I want to register an account by entering basic user information
I want to log into my account
I want to be able to edit basic user information

Design Outline

Open Alexandria uses uses a Client-Server architecture and consists of 3 main components:

Web Client:

The web client is built with HTML, CSS, and JavaScript. This is where information is displayed to the user.

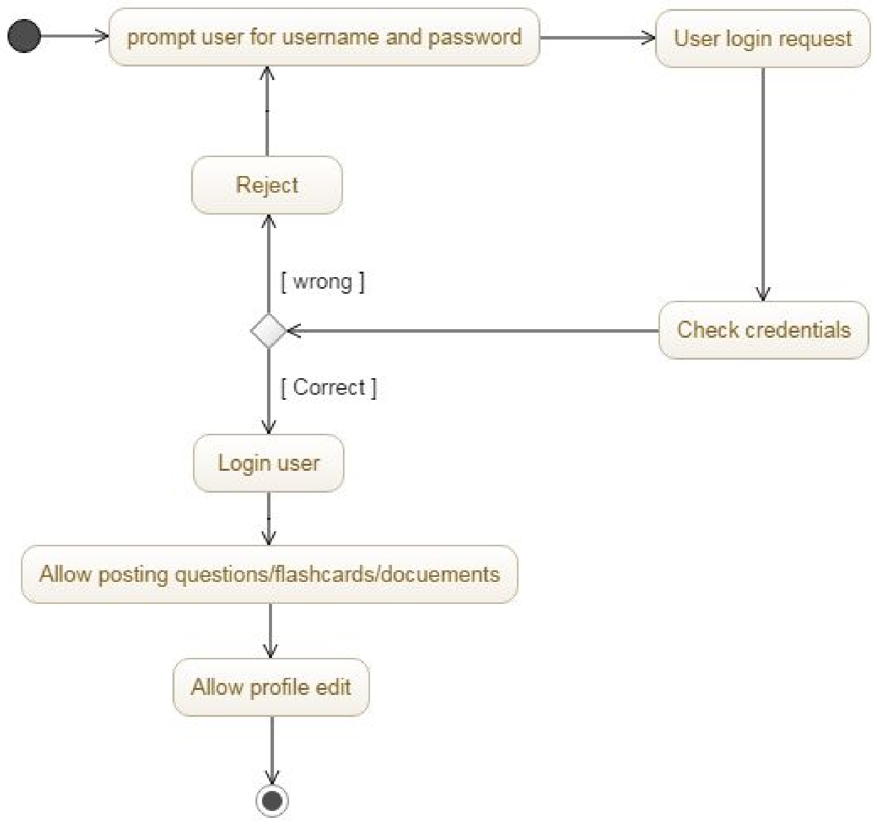
Server:

The server is built using Node.js. The current responsibilities of the server is to handle account information and file information requests from the web client. The server makes all access from and changes to the database on behalf of the client.

Database:

The database is built with PostgreSQL, an open source database that can will store core data concerning the users, groups, and documents.

The following UML diagram explains the login structure of the Open Alexandria system.



Design Issues

Design Issue #1 - Where do we host our backend services?

Option 1: Amazon AWS
We can take advantage of AWS's scalability and performance. However, it suffer from high learning curve as well as complicated Amazon specific documentation.

Option 2: Self host
We can also host our web application on our own hardware. This will all us to have total control over every part of our application. However, it would be hard for us to ensure uptime as well as security.

Option 3: Digital Ocean
Digitalocean have an intuitive UI to start and stop instances. However, it is pricy compared to AWS and self hosting.

We choose option 1 because we already have an account hosting other projects. The AWS will be capable of hosting more future projects. It is relatively easy to use and delivers all the service we need.

Design Issue #2 - What database will we use?

Option 1: MongoDB
Mongo is a popular database. It works well with Node.js. It also is in JavaScript, which is the language that we want to work in. It also provides high model flexibility, elastic scalability and high performance.

Option 2: MySQL
MySQL is easy to use, yet extremely powerful, secure, and scalable. And because of its small size and speed, it is the ideal database solution for Web sites. It is also free. Its main downfall is its rigid formatting requirements for database schemas.

Option 3: PostgresDB
PostgresDB is widely regarded as one of the highest performing as well as one of the most mature database available. We will be able to take advantage of Postgres' unique features since we have had the chance to use them before.

We choose option 3 because we are more familiar with Postgres. Compared to MySQL, it is more feature rich and better documented. Compared to MongoDB, its relational root is better suited for our use case.

Design Issue #3 - How will we build our client-side applications?

Option 1: Java.

Using Java would give us a multi-platform application. The team also has plenty of experience coding in Java. However, Java does not offer great visuals and Java applications can be slow and take up memory.

Option 2: C/C++/C#.

Using these languages would give our application the potential to be very fast. However, the development of the software would take a lot longer due to a lack of experience in working with APIs in C/C++.

Option 3: Web Application/HTML/CSS/JavaScript

This would allow us the most flexibility in development and we would not have to worry about memory management/consumption.

We choose option 3 because web design with HTML, CSS, and JavaScript is more flexible than with object-oriented programming languages. CSS has great frameworks, such as Bootstrap, that allow us to create great reusable content as well as diverse customization options. Both JavaScript and JQuery are integrated with CSS and HTML to achieve many effects of modern day website designs. Web design is something Java or C++ cannot easily achieve.

Design Issue #4 - How will we develop our server-side applications?

Option 1: PHP

PHP is fairly easy to use has a large amount of documentation. It is also simple to learn and implement. However, it is slower than Node.js.

Option 2: Node.js

Node applications are written in JavaScript arguably have as much documentation and help as PHP.

Option 3: Java.

Java has a strong foundation and all of our team knows it well. Java also has great debuggers and many IDE options.

We chose option 2 because Node.js is written in JavaScript. The client side of our application is also written primarily with JavaScript. This allows us to use a single, cohesive programming language to handle requests and responses between the clients and the server. And since both languages natively support JSON notation, it allows us to work with data objects rather than SQL-like data.

Design Issue #4 - How will the user log into the device?

Option 1: Google Account
Using Google Account would mean that that user does not have to spend time creating a new account. However, this will mean that the website must integrate Google.

Option 2: Local account
A local Open Alexandria account would allow us to have the easier implementation but sacrificing integration. We will store account information in our own database.

Both of these options have benefits, but the design will use option 2. We will create our own user authentication. Core functionality such as changing user information, upload or delete documents, and comments or reviews would be easier to implement when we control how user accounts are stored.

Design Issue #5 - What framework for the front-end do we want to use?

Option 1: Bootstrap
Bootstrap is a well-documented framework for quickly creating polished front end GUI's in the web browser.

Option 2: No framework, build from scratch
This would allow us to have complete control over the design of the GUI, but would require a significantly larger amount of the team's time to implement.

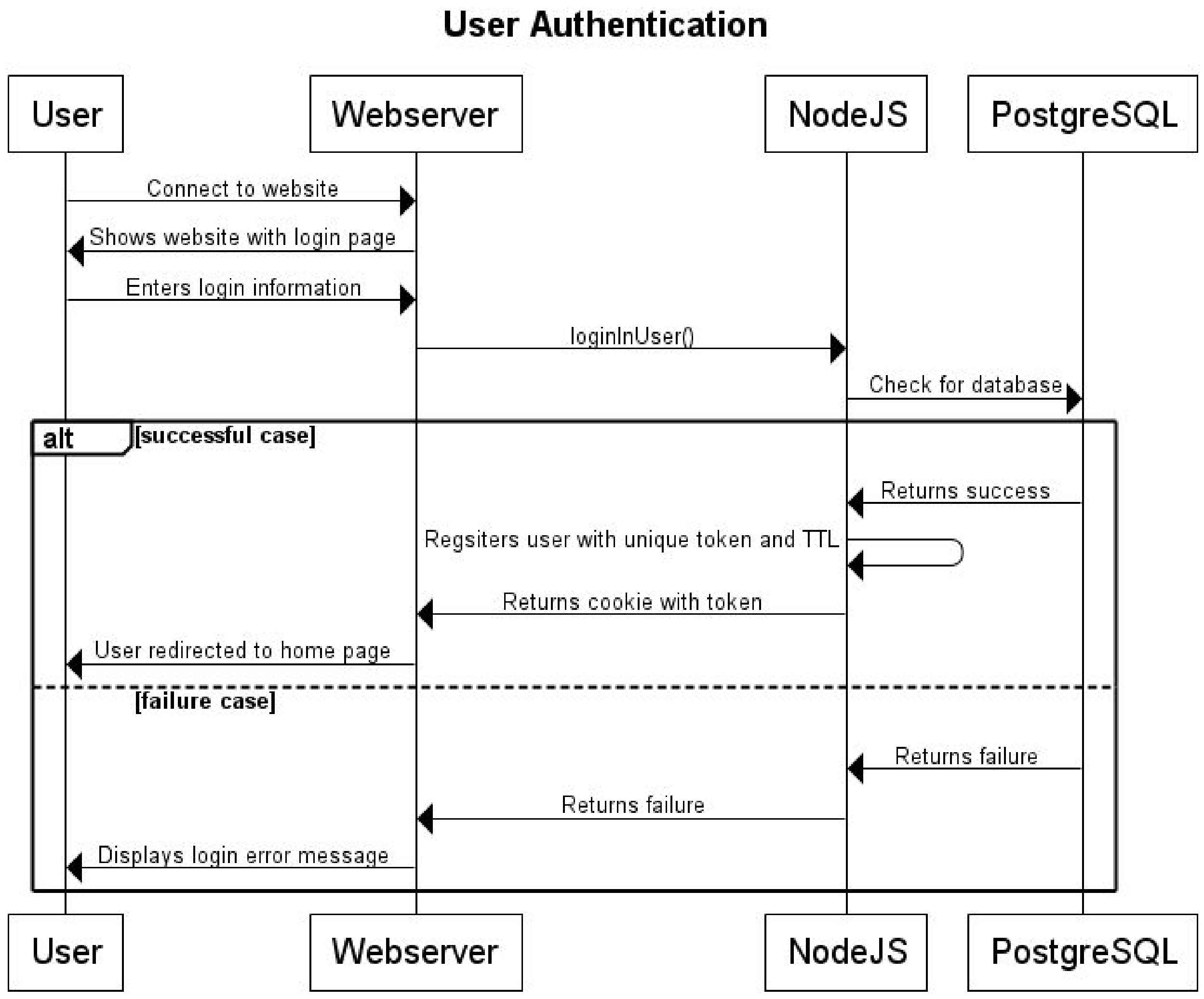
Option 3: Google Polymer
Google polymer would allow our team to create highly customizable, reusable web components for our front-end, but is poorly documented and still early in development compared to Bootstrap and Materialize.

Option 4: Materialize.css
Materialize is a web framework similar to Bootstrap that has less comprehensive documentation but provides a different, flatter design for front-end development using Google's Material Design principles.

We choose option 1 because of the high volume and quality of support documentation for Bootstrap. Additionally, 2 members of the team have previously had great results using Bootstrap to make highly usable web applications in relatively short amounts of time.

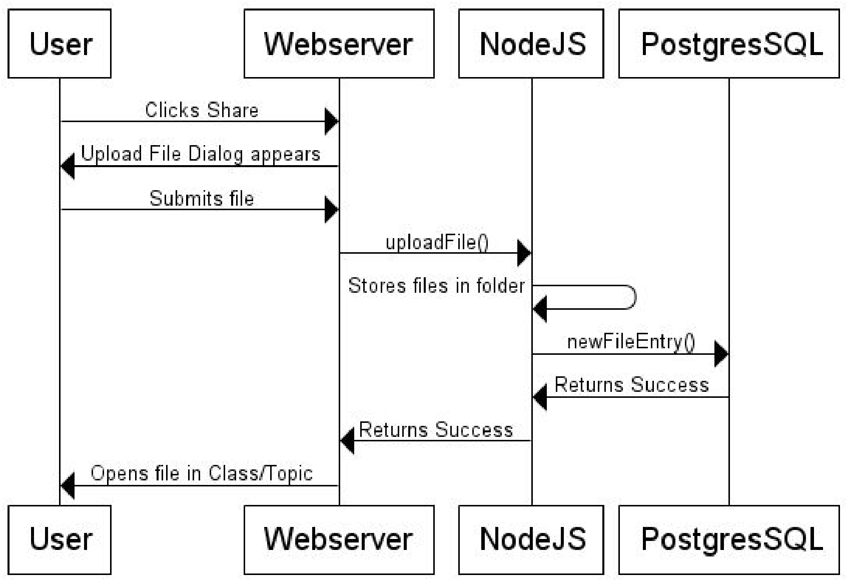
Design Details

Sequence Diagrams



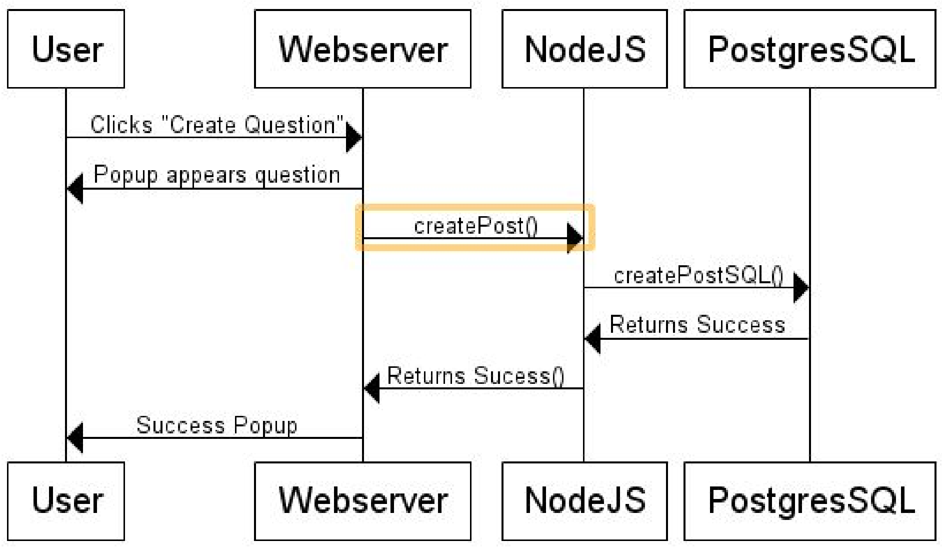
User authentication is important in our project since we will be constantly verifying the token given by the machine at any request to our NodeJS. At every RESTful API or “action” the user calls, our Node server will verify the login information against the database. All of the user information will be storage as cookie session information. If the data is found in the database, Node server will get a success response and generate access token. Node server will register the user as logged in with a default time-to-live of one day. User will be able to access the server if the cookie wasn’t cleared. Each RESTful API action will require the access to token.

Uploading Files



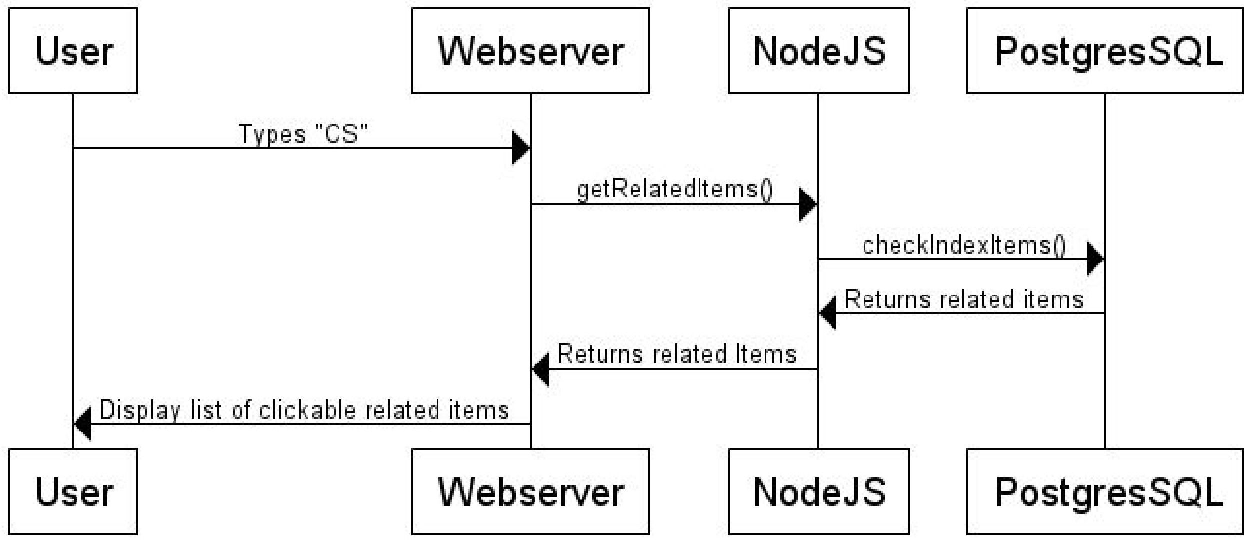
The diagram above describes the procedure to uploading files to Open Alexandria. With our platform, we are allowing users to share files with other users. The file will be stored on the server and indexed in the database for faster retrieval. Physical files will be stored in filesystem, but in the database, tags and a link to the file will be stored.

Create Question (Assume user sucessfully authenticated)



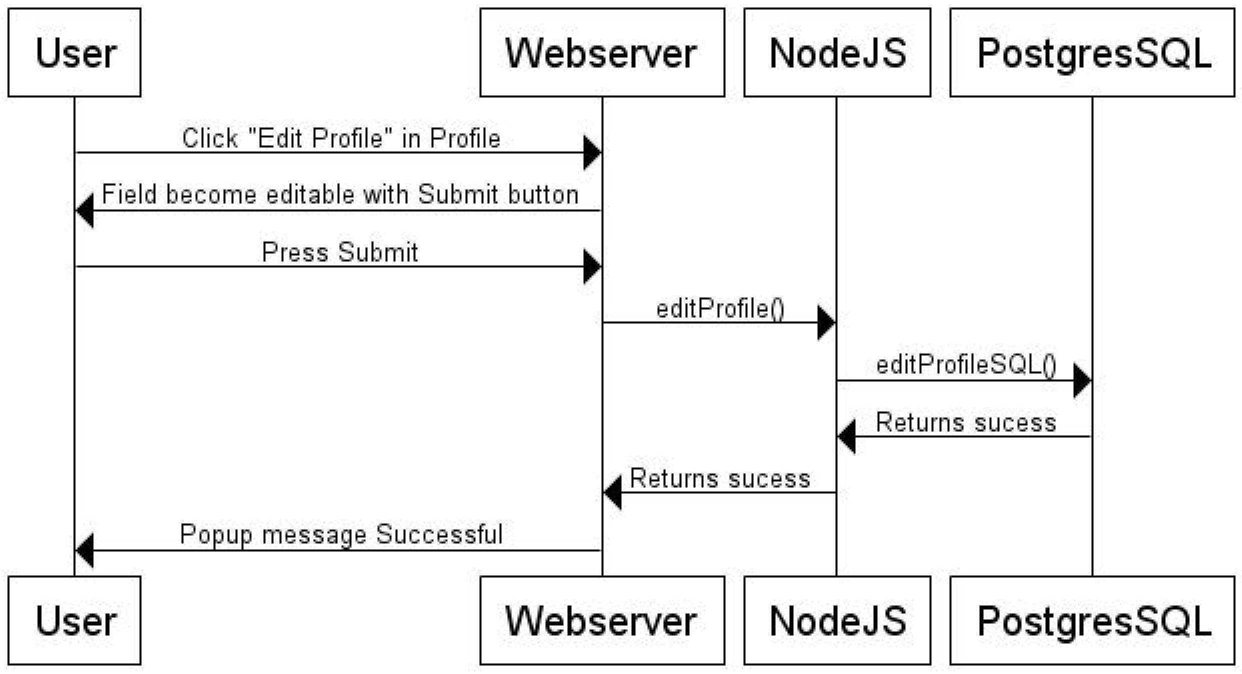
This scenario shows a student is creating a question when they are already inside a topic or class. A simple data entry into the database assuming user have already logged in.

Searching for Class/Topic



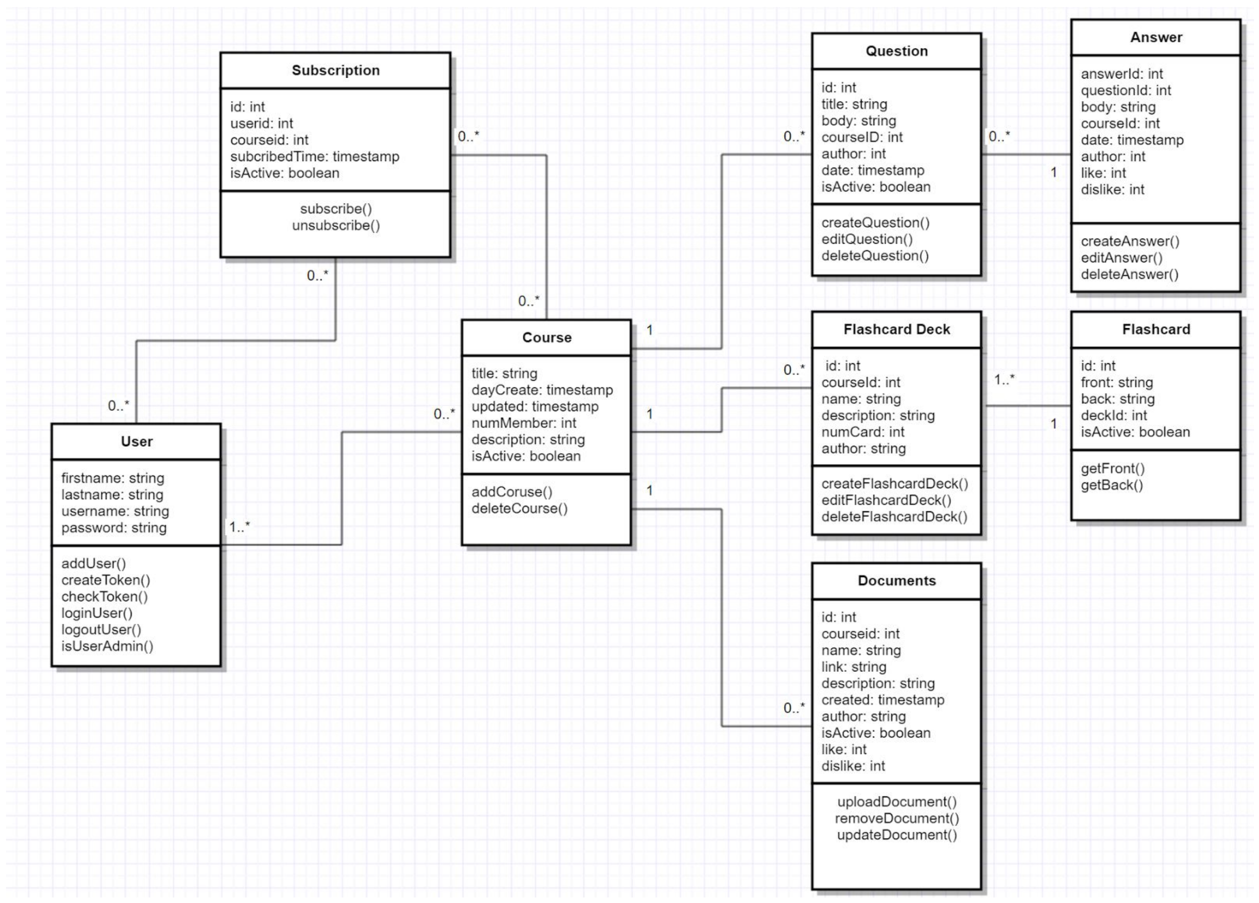
This diagram shows a scenario where the user is searching for a class or topic and a dropdown list of related items appear. This is a simple data retrieval from PostgresSQL. Again we will have to verify the user’s authentication.

Editing Profile

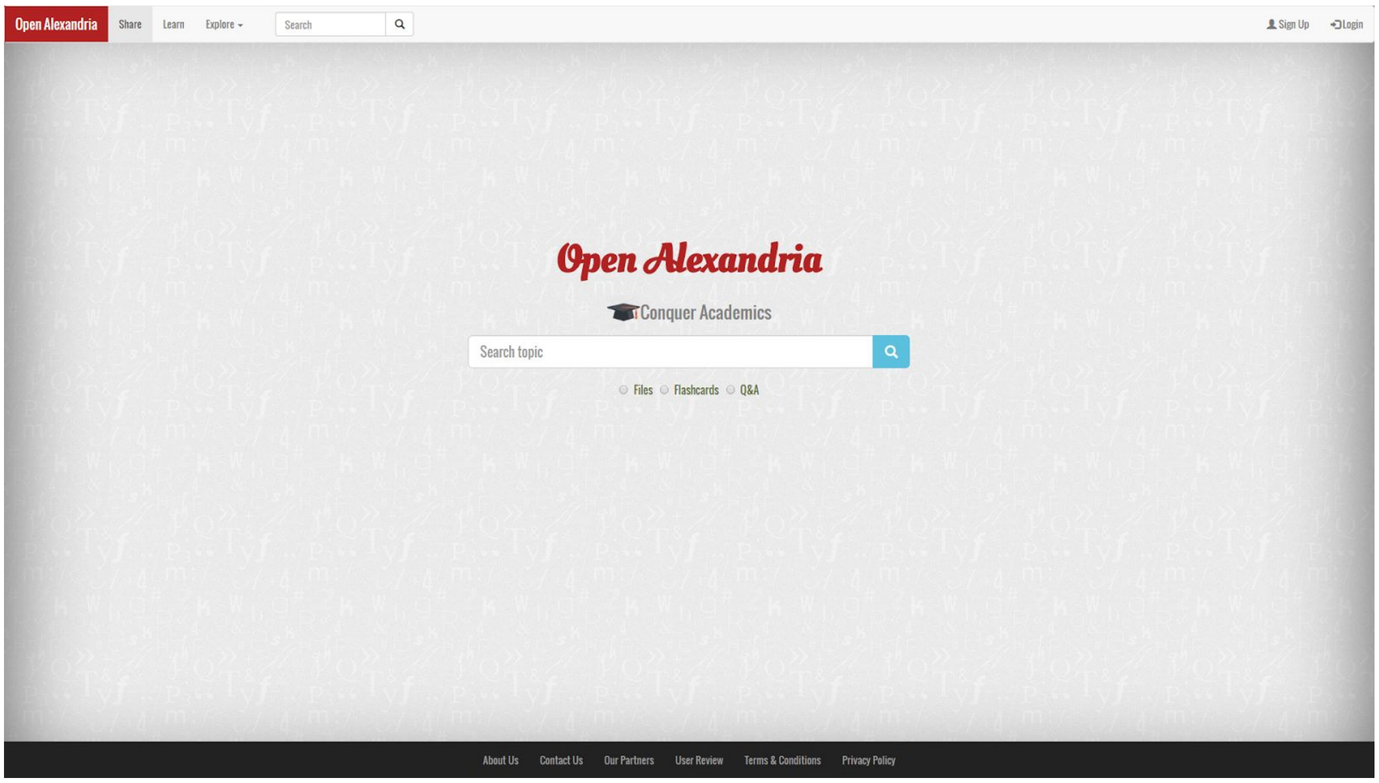


This sequence diagram shows the process of editing the user’s own profile. This would including changing passwords, name, email, or other information. It is a simple edit to the database entry given that the users have already authenticated correct.

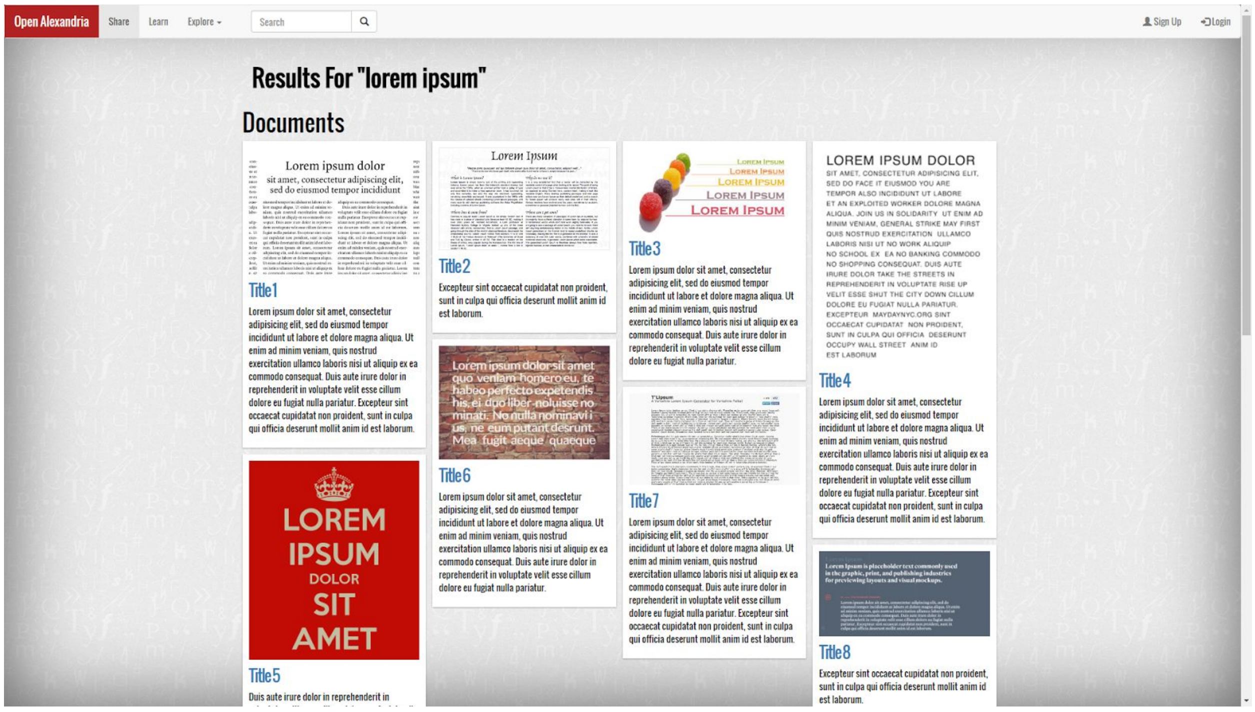
Class Diagram



User Interface Mockup



The theme of our client website will have a simple, but functional and polished design. This allows for a more focused and productive interface. The landing page is an uncluttered page with a search bar, since the main function is to find academic course and topic material.



Results For "lorem ipsum"

Documents

Title1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Title2

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Title3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Title4

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Title5

Duis aute irure dolor in reprehenderit in

Title6

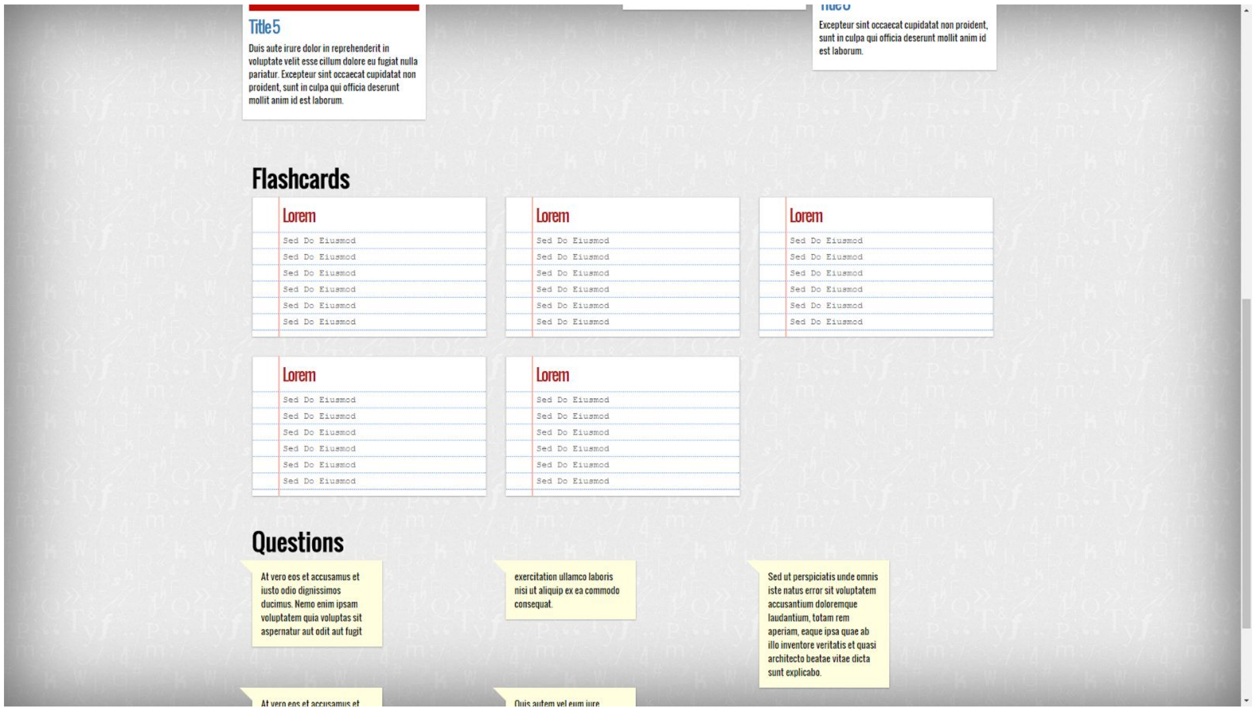
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Title7

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Title8

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Title5

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Flashcards

Lorem
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod

Lorem
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod

Lorem
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod

Lorem
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod

Lorem
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod
Sed Do Eiusmod

Questions

At vero eos et accusamus et iusto odio dignissimos ducimus. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit

exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

At vero eos et accusamus et

Quis autem vel eum iure

The search results page consists of a tiling layout of documents, flashcards, and questions. Users can click on a question to see its answer.