**CS 307 – Software Engineering**

**Design Document**

**Team 6: PU Network**

**Jaime Ortega**

**Stephen Kercher**

**Zhixuan Yu**

**Fangfang Sun**

**Di Pan**

**Table of Contents**

|  |  |
| --- | --- |
| **Title** | **Page** |
| Purpose | 3 |
| Design Outline | 4 |
| Design Issues | 5 |
| Design Details | 7 |

**Purpose**

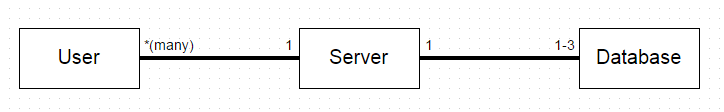
It’s very difficult for students on campus to connect with other students and faculty for help on assignments and studying in their courses. The PU network offers an online solution for Purdue students where they can connect with students and faculty involved in the same course allowing students to find help and support on school assignments. As a result, the PU Network is a student centric social network specifically tailored to college students with the sole purpose of helping them excel in their academic endeavors by allowing them to connect, share, and collaborate relevant information about their courses.

The following describes our functional requirements in detail:

* **Course Pages:** The course pages contain all the content relevant to each course. On these course pages is where students and faculty will interact, share, and connect.
  + As a student I would like to:
    - Send messages to students and faculty
    - View information about the course
    - Rate the professors of the course
    - Find students taking the same course
    - Find tutors for the course
    - Post questions relevant to the course
    - Answer questions posted by other students
    - Upload and share relevant documents about the course
    - Download documents from the course page
  + As a Professor or TA I would like to:
    - View information about the course
    - Post and answer questions for students
    - Upload and share relevant documents about the course
    - Share course announcements
    - Send messages to students and faculty
* **Account Management:** Students and faculty can manage their account settings and account activities such as messages and course page following.
  + As a student I would like to:
    - Register to the PU Network as a student
    - Manage my account settings such as name, email, and username
    - Manage my account passwords
    - Request the creation of a class page
    - Follow the class pages I’m enrolled in
    - Follow other students to see what they posted and answered on course pages
    - Invite other friends to join the PU Network
    - Create and manage study groups
    - Send and receive messages
  + As a Professor or TA I would like to:
    - Register to the PU Network as a Professor or TA
    - Manage my account settings such as name, email, and username
    - Manage my account password
    - Authorize the creation of the class page I teach in
    - Follow the class pages I teaching
    - Send and receive messages

**Design Outline**

The aim of our project is to build a website which serves as a platform for Purdue students from the same course to interact with each other. Our project will use the client-server model to set up a website that users could view the data about the course and interconnect with other users. In order to achieve this objective, we will connect the users to the website and store their account information and the information about each course on a MySQL database. The server serves as a mediator between the website and the database.



1. Website
   1. The website will be the place where the users can interact with other users in public (such as Q&A forum) as well as send personal message to other users.
   2. The website will display the course information for each course, and will display the information of each user as it applies to the courses they are taking. What each user allows other users to see of their profile is determined by the preferences in their account management.
   3. The user could also request for the setup of a course page if it doesn’t exist.
2. Server
   1. The server will serve as a middleman between the website and the database, whenever a request for database information is made.
   2. The server is also responsible for storing the code for displaying web pages and evaluating requests from users and acting accordingly.
3. Database
   1. We will create three databases for three different kinds of information.
   2. The first database is used to store information about the each user such as email, name, username, list of courses linked to, etc.
   3. The second database is used to store any information about the course such as the CRN, course info, students linked to the course, the identity of the professor, etc.
   4. The third database could be used to store any other information such as archives of forums, Q&As, or anything else needed.

**Overview of website**

We will have many users who are the clients and they are connected to the server by registering at the website, which will create an account for them. Then the server stores all the user information into our first database. Users could also request information from the website and the server will fetch any appropriate information from the database and display the information to the users.

**Overview of the database**

We are going to create 3 instances of databases using MySQL. Each of these databases will have different uses such as User information, Course information, and Forum archives. The reasons for doing so are to make it easier to keep track of the information and manage the security of the database. Using a database is preferable for security as accessing the information needs to be done through MySQL instead of just finding the plaintext file on the server.

**Design Issues**

Issue 1: Should we use several languages or one language to design webpage?

Option 1: Use several different languages

Option 2: Use only one language

Decision:

We chose Option 1, because we have to write several webpages and each group member can choose their own language to build their part.

Issue 2: Website Implementation Selection

Option 1: HTML

Option 2: PHP

Option 3: JSP

Option 4: Python

Option 5: JavaScript

Decision:

Most of our webpage will use Option 4, because it is easier to learn and one of team members has experience with it.

Issue 3: Database type

Option 1:MySQL

Option 2:MSSQL

Option 3:Microsoft Access

Option 4:SQLite

Decision:

We choose Option 1 because team members are more familiar with MYSQL and MYSQL is included in Python library. Also, MySQL is more suitable for small/medium size website. It is faster and easier to implement and is low cost.

Issue 4: How should the website be designed

Option 1: Design website from scratch

Option 2: Use existing webpage framework

Decision:

Because of the limited time we have, Option 2 seems to be the better choice for us. Option 1 can have better flexibility but it requires much more time to develop.

Issue 5: How do we distinguish the identity of user (for example: professors, teaching assistants, students...)?

Option 1: We can create user permission fields to make them different, which can be stored in database with other user information fields.

Option 2: When users log in, they can choose whether they are a professor, TA, or student

Decision:

We choose option 1 because different types of users will have different levels of permissions, so we want to make sure students cannot pretend to be professors or TAs.

Issue 6: How are the course pages going to be created?

Option 1: The course pages will automatically be created by having them supplied through myPurdue

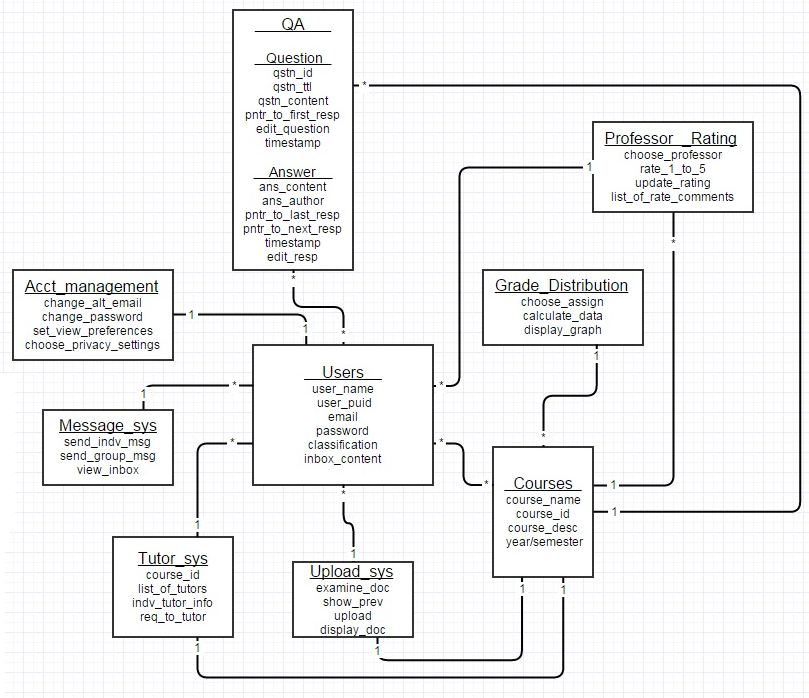
Option 2: The course pages will be created and supplied by users

Decision:

We choose option 2 because it’s difficult to access Purdue’s servers and have the course pages supplied.

**Design Details**

*Class Design*

****

**Description of classes:**

*User*

* The entity that encapsulates PU Network’s main users, which are the students, TAs, and Professors.
* The entity is classified by course name, id, and course description

*Courses*

* The entity classifies and organizes the course by course name, number, and id, and by year.
* It contains general information of the course through a course description

*QA*

* The entity that represents the Q&A forum on the course page
* Each question will have an ID, title, and time it was posted.

*Message System*

* The entity represents a messaging system very similar to an email that can send and receive messages between users or group of users.

*Tutor System*

* The entity is a list of tutors for each course that is organized by course ID
* The list contains the contact information of the individual tutor
* Tutors request to be added to the list with their contact information

*Upload System*

* The entity represents the system to upload relevant documents of the course to the course page. This is done through user interaction, used by professor and TAs to make relevant documents available

*Professor Rating*

* The entity is a classification system where the quality professor is rated from 1 to 5.
* Comments can be posted by users

*Grade Distribution*

* This entity represents the grade distribution of the class which will be in the form of a bar graph

*Account Management*

* The entity is the account management system where user can update and manage their account
* Users can change email and password, and set preferences and privacy settings.

**Sequence Diagram**

Figure 1.

Overall Sequence Diagram. It is a basic flow of what a user generally can do on the website.

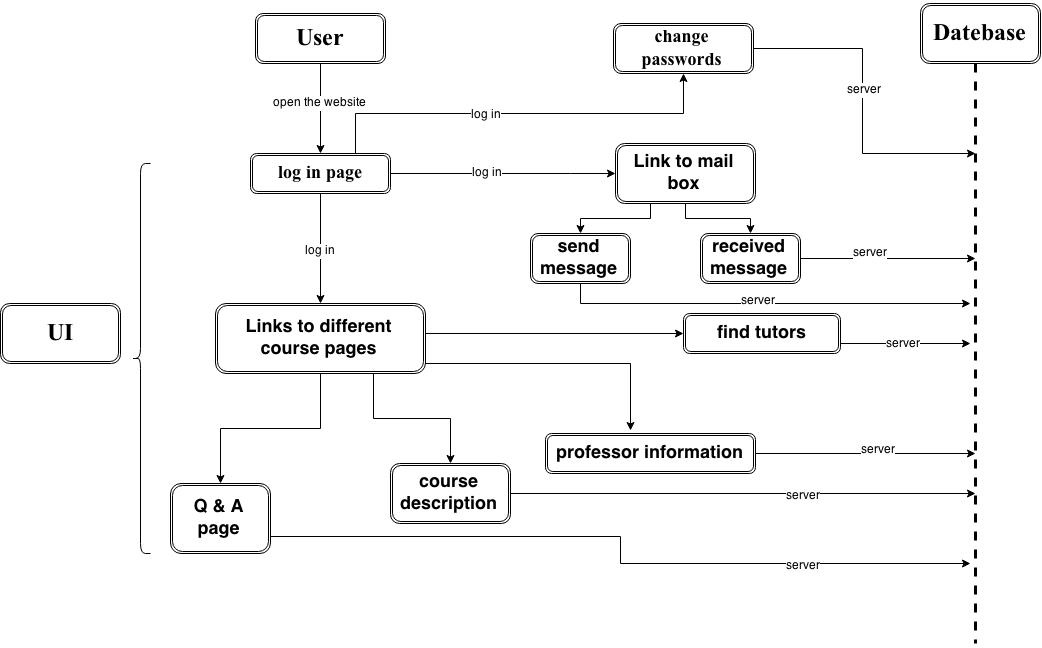


Figure 2.

Password Changing Diagram. In some cases, a user needs to change password, our website allows him to change it.

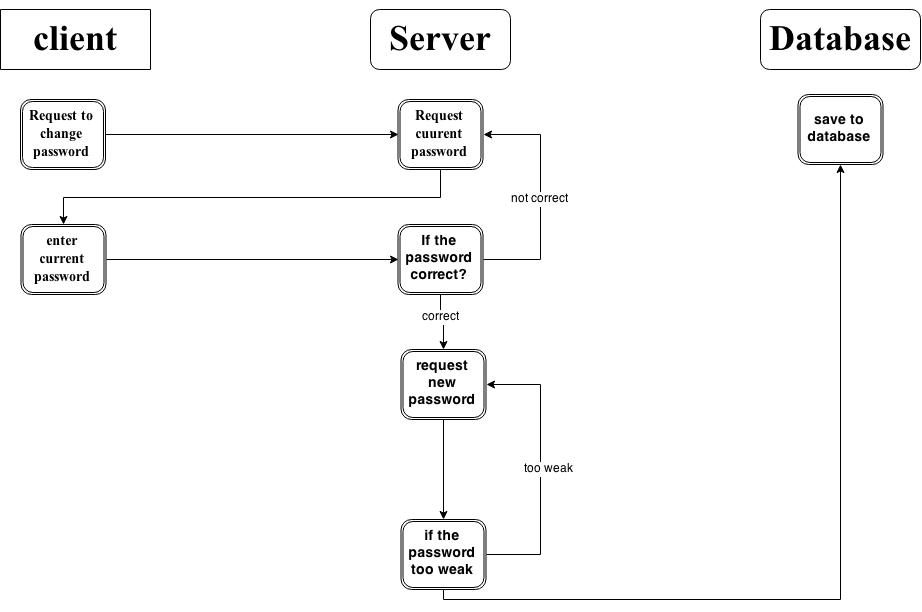


Figure 3.

Message sending/receiving Diagram. Users can send messages to someone else on our website.

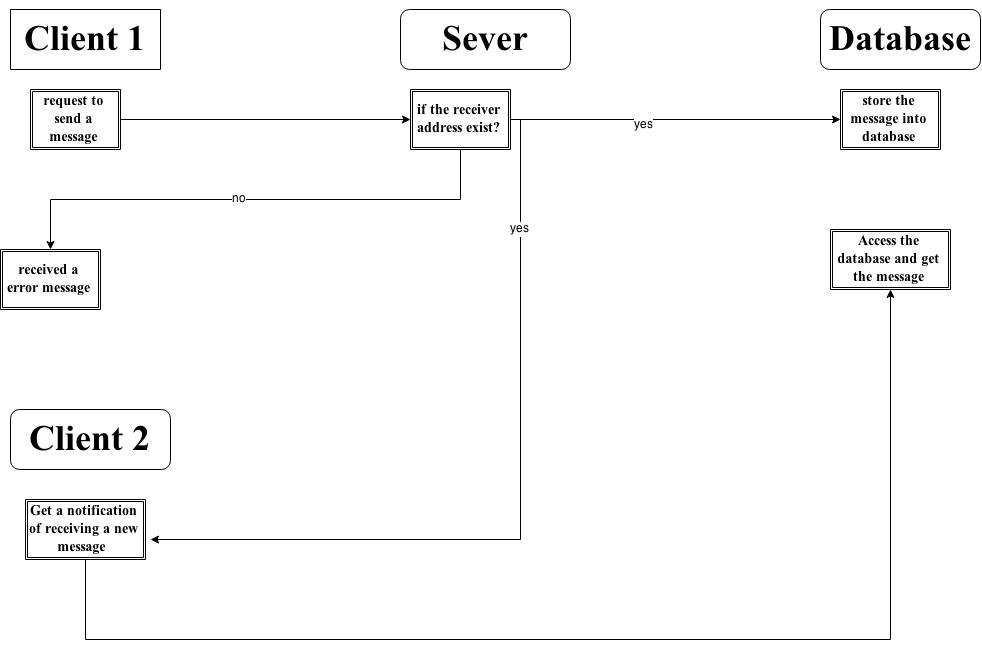


Figure 4

Q & A Diagram. Users could post questions online and professors and other classmates could answer their questions or discuss about the topic.

