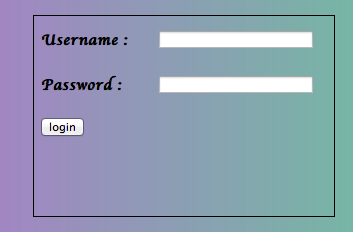
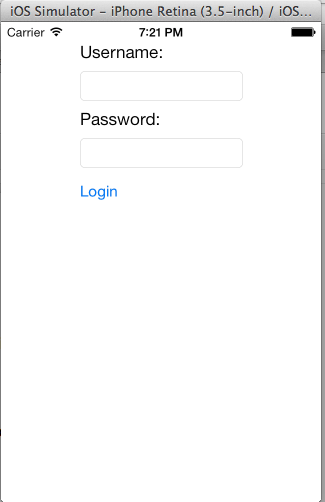
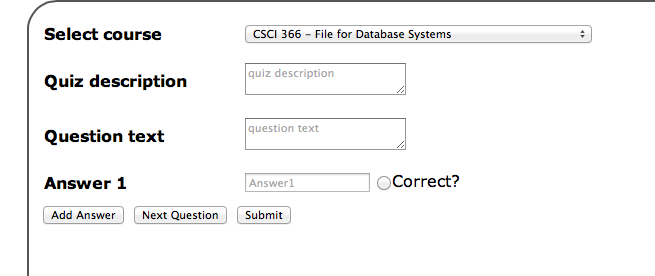
**OVERVIEW OF THE SYSTEM**

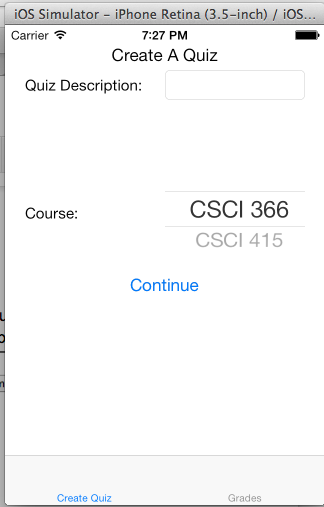
* Students and teachers can log in to the application using the interactive interface given below. The iOS device can be used by both students and teachers, and the system knows whether a student or teacher is logging in and forward the user to the appropriate screen. The web interface can only be used by the teacher.



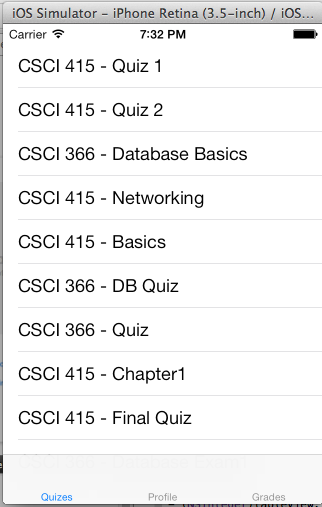
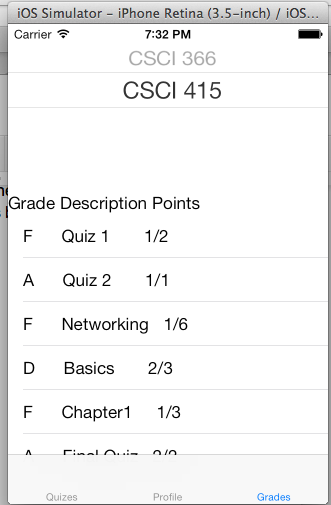


* Teachers can manage his/her classes and post the questions for a particular class. They can use either their iOS device or a web browser to create a quiz.

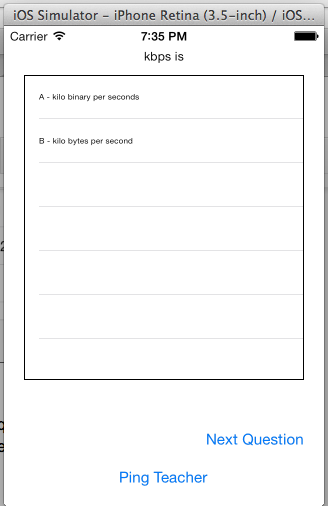




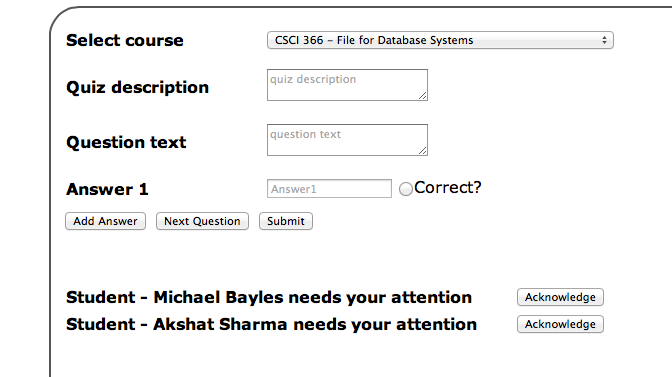
* Students can select which class he/she has to take the quiz. It also allows getting the immediate result to the students and the teachers because the quiz is graded instantly.

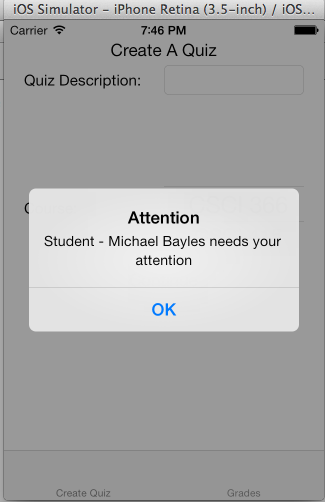


* If a student needs help clarifying a question during a quiz or has a question, he/she can press the Ping Teacher button to get the teachers attention.

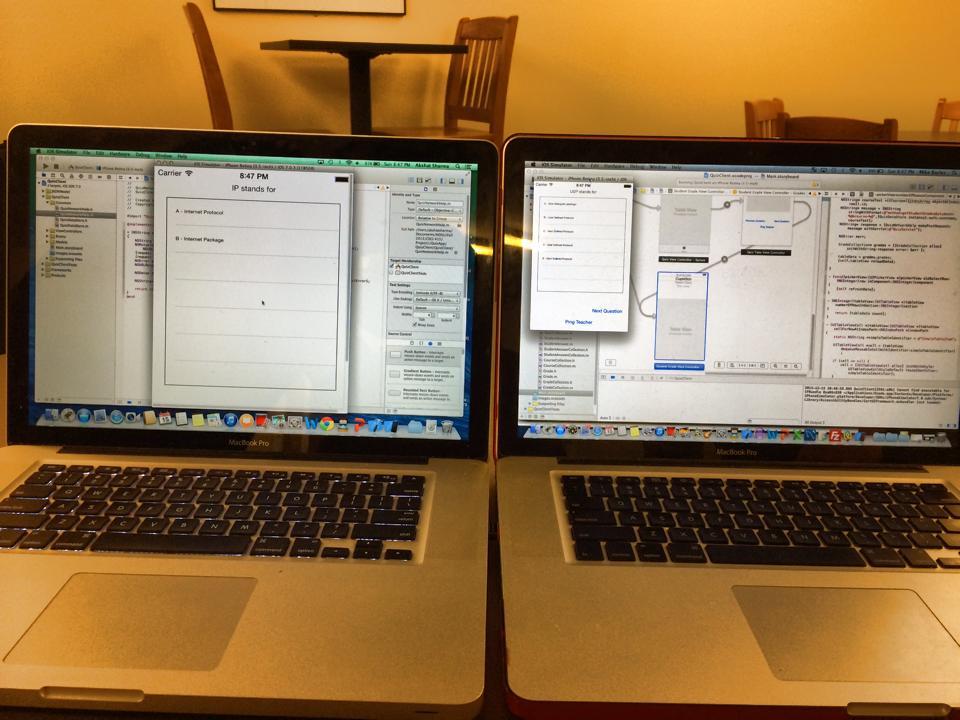


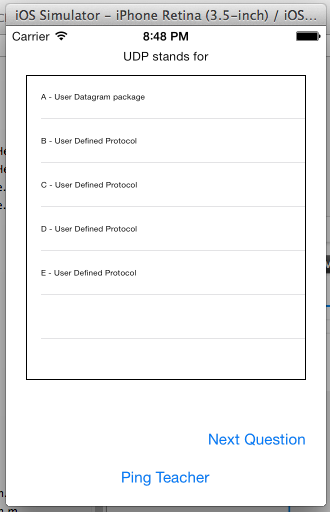
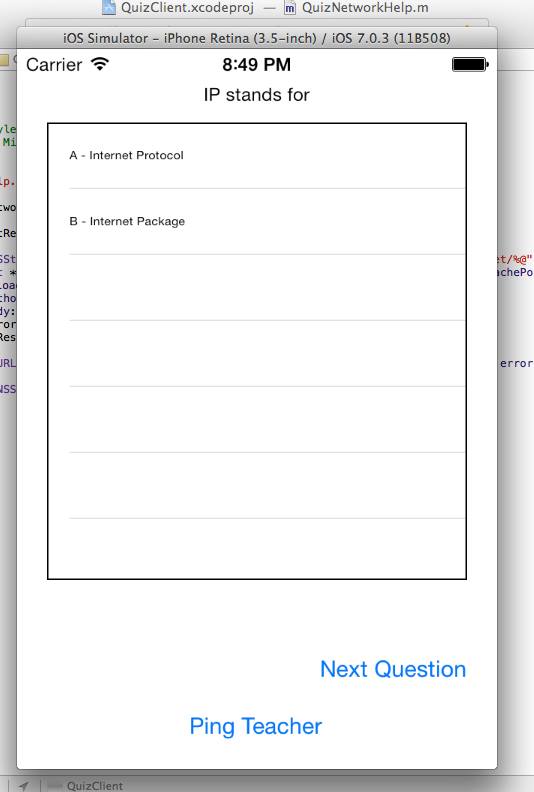
* Teachers will receive an alert on their device (web or iOS) if a student needs help.



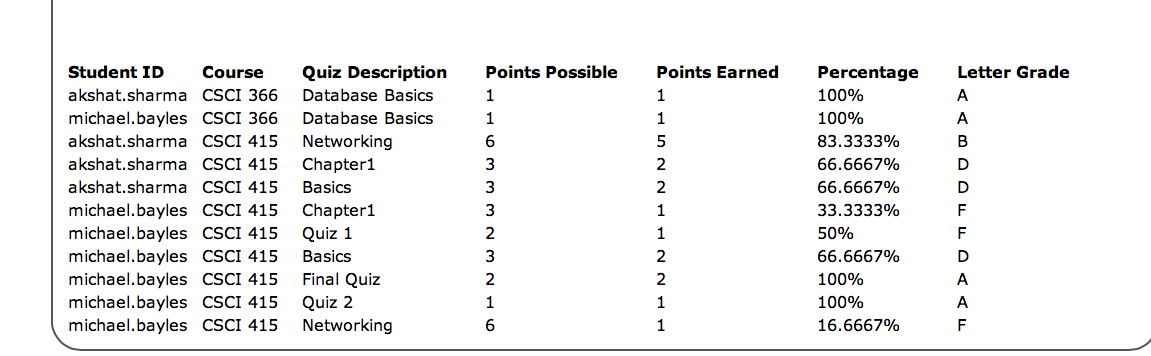
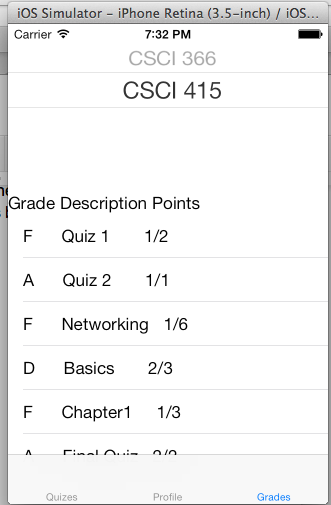


* Multiple users can take the quiz at the same time. Also notice that the questions have been shuffled to prevent cheating.



* Students can get the immediate feedback of the quiz on their iOS device as well as the teachers. The Teacher gets the update of the score on the web browser too.

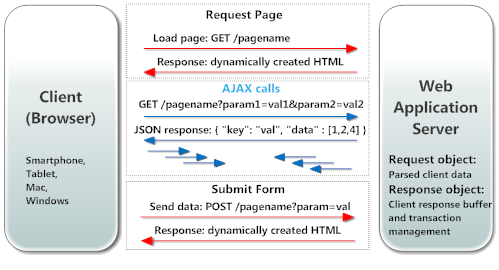


**Related existing application/System**

* Blackboard application is an existing Web-based server software which features course management
* It may be installed on local servers or hosted by Blackboard ASP Solutions
* Its main purposes are to add online elements to courses traditionally delivered face-to-face and to develop completely online courses with few or no face-to-face meetings.
* The application is an effective way of helping students organize their assignments and prioritize on the basis of the deadlines.
* However, the blackboard application that we have don’t allow the user to ping the teacher right away while taking the quiz,
* As a result, students have to tell it to the teachers in person or by email.
* Our system effectively used the concept of networking and students can ping the teachers right away as they are stuck while doing the quiz.
* The existing Blackboard application is there for different platforms on mobile but it takes quite a lot of time in order to authorize the user.
* Although our system is strictly designed for the iOS, but since our database is a lot smaller, it is much faster and easier to retrieve any data.

**Detailed Explanation**

* We have extended the basic concept of socket programming in this application. Our application acts as proxy server and sends the request for connection to the server, which makes connection to the database to get all the necessary data from the tables.
* We have implemented multiple threads to support parallel programming allowing multiple users to take the quiz at the same time, and at the same time we have taken care of thread synchronization preventing any classes in the critical section. Database access has been synchronized to enforce mutual exclusion on data.
* All the results of the query to fetch data is passed on using JSON from Java environment to the Objective C. JSON acts as a language-neutral layer that both Java and Objective C can understand. Because of this, it would be very easy to implement clients using any language without having to modify our server.
* The basic flow of events is
  + User selects action on client
  + Client serializes necessary data to json
  + Client device sends post message to server
  + Server analyzes post message and decides what to do
  + Server deserialize json to necessary class
  + Server does appropriate action accordion to the message received
  + Server serializes content to json
  + Server responds to the client
  + Client receives response and deserializes json to appropriate class
  + Client displays content



**Conclusion**

* We have effectively used the concept of parallel computing, because different users can take the same quiz at the same time. This is implemented creating multiple threads and synchronizing them when necessary
* We have effectively used the concept of networking by making use of socket programming, TCP/UDP, HTTP, and making our application lightweight by posting all the questions on the server and requesting the data as per the request from the database.
* Our system is extremely scalable and dynamic, as it is very easy to implement clients on different platforms, languages, etc.
* We want to make our system more user-friendly and convenient to sell it at the app store at some point later.
* We want to focus more on the design aspect of the interface, because our design is fairly simple and not pretty.
* Signing up for an account should be done through some sort of administration page that only certain people have access to. Right now, we have to manually edit the database tables to do such a task.
* We’d like to make an Android client as well as a C# client.
* Add some extra features to the grading to sort, filter, etc, as well as see stats such as mean, median, and mode.
* Use the native iOS notification system to allow teachers to receive notifications even if he/she isn’t logged in.
* We could randomly choose from a question bank to allow for different questions in different quizzes
* Our application assumes there are up to 5 answer choices that are only multiple choice, and every question is worth exactly 1 point.
* There isn’t a lot of error checking, so we’d like to prevent things such as forgetting to select ‘Correct Answer’ or not adding any answers
* We need to enforce that a student can only take a quiz once