How Question Asking During Lectures Could Be Improved Using Digital Technology

Alexander Lind
Andy Ishak
Gustav Dyrssen

Civilingenjörsprogrammet i informationsteknologi
Computer and Information Engineering Programme
Abstract

How Question Asking During Lectures Could Be Improved Using Digital Technology

Alexander Lind
Andy Ishak
Gustav Dyrssen

In order for students to learn a lecture’s subject, question asking is important. Although this, students tend to not always ask the questions that they come up with. One reason for this could be that students think that they are the only one in the room that have this question. Another reason might be that they are shy of talking in front of a crowd. In this project, a web application, called Hum, was developed in order to help students ask questions both by letting them be anonymous and not interrupting the lecturer. Moreover, in order to make Hum available for as many students as possible, the system was implemented as a web application. Using Hum, students are be able to post questions during lectures, either anonymously or not. Students can also view and upvote questions asked by other students. Following this, teachers can answer these questions, both in the application and during the lecture. All questions and answers entered in the application are saved to a library, and in this way, students can also delve deeper in previously unanswered questions.

Handledare: Sofia Cassel and Björn Victor
Examinator: Björn Victor
Sammanfattning

Contents

1 Introduction 1

2 Background 2
   2.1 The Flipped Classroom Model 2
   2.2 Question Asking During Lectures Today 2
   2.3 Phone Use During Lectures 3

3 Purpose, Aims and Motivation 4
   3.1 Usage During Lectures 4
   3.2 Scope and Relevance 5

4 Delimitations 5

5 Related Work 6
   5.1 Studies and Projects During the Last Decade 6
   5.2 Similar Systems 6

6 Method 7
   6.1 Site Management 7
   6.2 Site Enhancement 8

7 System Structure 8
   7.1 Client and Log In Page 9
   7.2 Students’ Usage 9
   7.3 Teachers’ Usage 10
   7.4 Database and Communication 10
1 Introduction

In today’s lectures, students may face large amounts of difficult material covered in a short time span. Consequently, students may come up with questions regarding the covered material. In our experience, many of these questions seldom get answered partly due to the fact that many students do not ask them. All of this is both based on our own experiences and can also be read about in an article by Carnegie Mellon University[21]. One contributing factor to students not asking questions may be that they do not want to interrupt the lecturer, or possibly since they do not dare to ask questions when the crowd is listening[22]. In order to help them ask these questions, an idea of a system was developed; a web application called Hum. Since this system was implemented as a web application, it is available on any device that has a web browser.

Hum is an application where students can create accounts and register to courses created by teachers. After this, students can both ask questions and view and upvote other students’ questions during lectures. All questions are saved in a library where both students and teachers can view and discuss them after the lecture. As Hum was developed as a web application, it is accessible on all devices with a web browser.

By asking questions using Hum instead of asking questions directly in class, students can choose to be anonymous. Following this, the lecturer can chose to instead of answering published questions during the lecture, write a response later in the web application. By using Hum, the questions are gathered in a library where the teacher can view and answer questions from previous lectures. Moreover, the questions are published in a live feed, and thus, other students using the application have the opportunity to upvote a question if they are wondering the same thing. Using this upvoting system, questions that a large part of the students are wondering are highlighted in the application. Furthermore, all questions and answers entered in the system are saved to a library. This library will be accessible to users of the web application so that students will be able to go through previous questions and explanations.

The different users of Hum was divided by two roles: teachers and students. Depending on a user’s role, he or she is directed to specific pages in Hum with different properties and available functions. A teacher has for example the ability to create a course that students can join. Furthermore, teachers’ answers to questions are distinguished from students’: teachers’ answers will be emphasized more compared to the students’ answers, which will be more of a discussion.
2 Background

Today, technology is influencing a growing part of our community, and this project focused on technology as an aid when teaching and lecturing. Here, different means of technology are present, e.g. in lectures where computer slides viewed using projectors play an important part in many classes. Furthermore, even before the introduction of digital projectors and computers in class, overhead projectors was used in similar ways.

2.1 The Flipped Classroom Model

Even when using the technology discussed above, lecturing could often be seen as old-fashioned in the sense that it does not engage students enough[25]. This does not only affect each and every student’s education, but also the overall course quality. A study[32] shows that active learning in lectures increases student performance, and the interest for a enhanced lecture structure has therefore contributed in the development of new methods and platforms based on the Flipped Classroom model[34].

When using the Flipped Classroom model, students watch online lectures at home and solve exercises with help from the teacher during class. This method has proved to increase the students’ results in various schools, and one example is the Spartan College of Aeronautics. Journalist Casey Smith writes in her article[33] about the college that the total test pass rates increased from 83.9% to 96% when using the Flipped Classroom method. This is one example of technology being an important tool for improving lecturing.

As can be seen in both the study and the article above, teaching methods continue to evolve in order to engage students more. However, another aspect of lectures, where usage of technology is not as common, is when students are given the opportunity to ask questions to the teacher. In this situation, technology is commonly not playing as big of a part as the previous examples. Instead, questions are asked in the same way as they have been for a long time.

2.2 Question Asking During Lectures Today

Students sometimes hesitate to ask questions in today’s lecture format[19]. According to discussions on the forum Quora[30], one reason for this could be that the material covered during a lecture is difficult to grasp. This could lead to questions being raised by the students, questions that may be asked out directly or just raised in the mind of
the students. When the latter happens, a real problem could occur as these questions would neither be answered nor noticed. In addition, another reason could be that some students may be insecure when asking questions in front of a crowd[24], this could lead to uncertainty of the pertinence of a question. Consequently, these questions may never be asked, and therefore, the questions may never receive an answer.

Besides this, a subject can for one student be comprehended with ease, yet for another student it raises more questions. A student can then feel anxious to ask a question as the answer may be obvious to others, thereby risking exposure to oneself of appearing unlettered[30, 24]. This does however not lessen the importance of being able to ask questions, and a way to anonymously ask and receive an answer could help students with this anxiety.

Sometimes lecturers may get into a rhythm of tempo and flow, and some teachers may not want to be interrupted during such a flow. Even though a question might be of real importance and truly be helpful for the entire lecture, students often may stop from asking the question in order to not disturb or hinder the rest of the class[30]. Likewise, a certain amount of material is often meant to be covered during a lecture. This, combined with the limited time span for a lecture, could lead to interrupts not being appreciated by the rest of the class[30]. This could be because the lecture might get delayed or other material will not be covered during the limited time.

Consequently, a way for students to anonymously ask questions and receive answers without disturbing or interrupting the rest of the class or lecture would be of aid. To help students further, it would be beneficial if questions raised during lectures also could be pondered about when time is more allowing. Therefore, a way to discuss the answers and understand the problems outside of the lecture halls (or in the end of a lecture when there is time left) could enhance lectures. This would increase the possibilities for students to understand the material covered during a lecture.

### 2.3 Phone Use During Lectures

According to an article[23] written by Dr. D. Duncan of the university of Colorado, smart phone use during lectures has increased dramatically during the last decade. Besides this, the article also states that students regularly get distracted from the ongoing lecture due to their frequent smart phone usage. Moreover, a study[29] shows that students’ multitasking capabilities are not nearly as effective as they think themselves, and furthermore, a habit of checking ones smart phone regularly has surged, were users frequently check their devices for updates and notifications. However, the same study also shows that this spurs users to simultaneously check for other things on their phones.
As students may look at their screens frequently (e.g. for checking social media like Facebook), our project can grant them with the opportunity to instead use their phones for academical purposes, e.g. question asking.

### 3 Purpose, Aims and Motivation

The purpose of creating Hum was to provide a tool that could be available for all devices and that could encourage students to ask questions during lectures. This was done by using our web application as a teaching aid. In addition, this project intended to enhance the overall quality of lectures by providing a system for teachers to answer the questions directly in the application. This could be beneficial to all kinds of lectures since it can improve the quality of learning for the audience. Also, when using the system, lecturers would be able to directly get feedback of which parts of a lecture that students find difficult to understand.

#### 3.1 Usage During Lectures

One of the aims of Hum was to increase the audience participation during lectures. This was done by making question asking through a live feed available in the new way discussed earlier. By designing using our web application, students are given the possibility to write and upload a question to a live feed of questions. Following this, other students using the app will be able to upvote already asked question, or naturally, ask questions of their own. Likewise, the ability to let students be anonymous when both asking and upvoting questions was another aim, and by implementing the application as intended, this was fulfilled. The purpose of this was to get students to ask questions they otherwise would not. Furthermore, another goal was to narrow down the gap between the teachers and students, and this could be realized by letting teachers use the question data from the application as feedback in order to improve future lectures.

Providing students with questions from previous lectures along with discussions and answers to the questions was another of the projects goals. This was done by letting students have access to the *Library* of asked questions. In the library, students can view questions and write discussions about them. All of this could possibly increase students’ ability to learn more using Hum.
3.2 Scope and Relevance

When looking at the examples in 2.2, one can see that the traditional method of teaching still dominates lecturing in universities today, and therefore, new supplements may be a welcome addition in order to improve students’ learning during lectures. On the other hand, there is nothing that says that this tool can not be used in other lecture formats, e.g. the Flipped Classroom model. Our vision was to build a supplementary tool that could improve the communication during lectures and simultaneously bring relevant feedback to both teachers and students. Providing students with a digital application that could be integrated into the curriculum was what motivated us to delve deeper into this project.

Evolving and modernizing the lecture format is important; today, one can watch lectures online for free. If lectures at universities do not encourage the use of modern technology, lectures online could be as effective as the traditional classroom model. As discussed above, technology has not influenced question asking during lectures as much as other parts of lectures. One tool building on the Flipped Classroom model is Scalable Learning[17]. Here, students have the opportunity to ask questions during online lectures. Yet, Scalable Learning cannot be used with traditional live lectures; although, this is a future implementation that the foundation are working on according to Dr. Black-Schaffer[16]. This is where Hum complements this tool along with the Flipped Classroom model and other teaching methods. In addition, Hum could also be used during lectures and presentations outside of universities.

This project could be interesting to different kinds of lecturers and lecture takers, and especially in universities, where lectures are part of most courses. We have discussed this project with Dr. Black-Schaffer at Uppsala University, and as Hum has common goals with his project mentioned before, Scalable Learning, he has been a guidance counselor for Hum.

4 Delimitations

The application was developed to be cross platform accessible since it was implemented as a web application. Thus, the system could be usable for any device that uses a web browser. This limited our design, but on the other hand, it also made it more available. As contradictory as this may sound, the system is widely available, however not by specific applications for Android, iOS, Windows, OSX or other operating systems. Notwithstanding the possibility, this is further discussed below in the section Future Work 14, but these specific apps were not implemented during this project.
Furthermore, the application was limited by the possible amount of active users. As the server used for the project may not handle the possible large amount of clients, the application’s effectiveness was limited. For further development, the requirements of the server have to be re-evaluated if the application would be used in a larger scale.

5 Related Work

5.1 Studies and Projects During the Last Decade

This project regarded the implementation of an application that would not only work as a communicative tool in lectures, but also as a study aid for students outside of the classroom. Finding a technology that can improve students’ curriculum is a field of study that has been relevant for many years.

Alison King’s article, "From Sage on the Stage to Guide on the Side"[25], is now known as the pioneer of the lecture format that we today call Flipped or Inverted Classroom. Research[31] in this field aims to increase student engagement in lectures. Though this is shown to be a preferred format with positive results[33], it does not emphasize enough on reaching those individuals that does not ask questions, e.g. in fear of showing low level of knowledge.

For a long period of time, increasing students’ level of engagement during lectures has been a problem of great concern. Studies concerning the use of "classroom clickers” have been done for at least 15 years[18]. The main problem with this technology is that the questions asked are still dependent on the teacher, which will have to select questions that students can answer using the clickers. This method do help students to participate in lectures while being anonymous, but the only way in which they can participate anonymously is by answering the questions asked by the teacher. Additionally, studies[28] has shown that using technology, like mobile devices, in order to enable audience members to ask questions anonymously is a good method to communicate with the lecturer.

5.2 Similar Systems

Mentimeter[10] is, like Hum, a service that focuses on enhancing the audience participation through a web application. The difference from Hum is that the responsibility lies on the lecturer to provide questions. The audience members are then given alternatives specified by the lecturer to choose between. The results of the voting is shown through
the website, or Microsoft PowerPoint, as a plugin. Instead of letting the lecturer have
the responsibility, Hum focused on giving the audience the opportunity to ask questions
without troubling the lecturer.

The Scalable Learning project[17] is an example of the Flipped Classroom model, and
it is currently used by more than 12,000 students across many universities in the world.
Scalable Learning uses technology to further enhance the benefits of Flipped Classroom
teaching by providing teachers with the ability to ask students questions during online
lectures using questions and quizzes. Furthermore, it also provides the possibility for
students to give feedback about parts of lectures that they did find hard to understand.
In particular, when watching an online lecture, students can pause and write feedback
or questions that the teacher will be able to see, along with data of exactly where in the
lecture the question or the feedback was given. This is similar to our project; Hum tried
to achieve similar features for live lectures by engaging students to ask questions during
lectures using their mobile devices.

6 Method

6.1 Site Management

Hum was developed using Wordpress[37], a content manager system with an embedded
database. Wordpress provides a web site structure as well as a database that were used
for storing the questions and answers entered in Hum. Wordpress content manager
system were used for handling the different users. It was used for creating new users,
managing the log in functionality and distinguishing between teachers and students.
Apart from the content management system, Wordpress` database structure was also
used when implementing Hum. These two parts of the system are the only things used
taken from Wordpress.

There are few services that provide as much implementation guidance for developing a
web application as Wordpress does. Particularly, in a study by W3Techs[36], Wordpress
was used by 26.3% of the web sites showing its immense versatility and practicality.
Compared to other alternatives such as Joomla![7] and Drupal[3], Wordpress is both
better suited for beginners and more widely used[27]. As we did not have any experi-
ence with web development before this project, Wordpress were chosen as the Content
Management System for Hum.
6.2 Site Enhancement

Hum was also implemented using the framework *Bootstrap*[1], which provides a front-end web framework. Basically, Bootstrap adjusts the layout dynamically of web pages, independently of which device the web page is used on, e.g. computers, phones or similar. According to website builder Laurence Bradford’s article[20], Bootstrap, compared to another framework (*Foundation*[4]), has a greater community support and is more widely used. It also supports more browsers, and thus, Bootstrap was chosen for this project.

In order to make Hum interactive, a JavaScript library called *jQuery*[8] was used. *jQuery* provides functions for animations, manipulations and more that was used in Hum. Similar libraries exists, e.g. AngularJS[2], which provides comparable functions, but *jQuery* was chosen due to its widespread usage among top sites[9] and also due to the fact that it was provided in Wordpress.

7 System Structure

![System Structure Diagram](image)

Figure 1: A flow chart of Hums’ system structure. Circles represent different views of the system, rectangles represent main parts of the system and hexagons represent sub parts of the system. The wider connection between Usage and Database emphasizes that the main communication is between the database and Usage.
7 System Structure

7.1 Client and Log In Page

In Figure 1, two different kinds of attributes are connected to Client: teachers and students. These two are the user roles available when signing up for the system. The main difference between them is that teachers can create new courses and lectures while students only can join existing ones. When a user enters the web application, the user is directed to the Log In page. Here, the user can either log in to an existing account or create a new one, and when doing so, the page communicates with the database. The Log In page is the only page available to users not logged in to the system.

7.2 Students’ Usage

When logged in, students are redirected to one of two possible pages, either the Live Feed page or the Library page. Users are redirected to the Live Feed page if there is a live lecture going on in a course that the user either has created or is registered to. On the other hand, if there are no live lectures when the user logs in, the user will be redirected to Library. Logged in users have access to the Settings page and the menu (see Figure 2), where users can navigate to different pages of the application.

Figure 2: The menu of Hum, accessible for users logged in to the application. The menu is hidden before the user clicks on Menu. By clicking on the different texts, users are redirected to the corresponding pages.
7.3 Teachers’ Usage

When teachers log in, they are instead redirected to the course settings page where they can manage and add courses and lectures. This is one of two parts where teachers’ usage differs from students’ usage. The other part that differs is the Library. Here, teachers’ answers to questions are distinguished from students’ discussions.

7.4 Database and Communication

The Database is the only part of the system which was not designed from scratch. Instead, the default Wordpress’ database was used and additional data fields and tables were added. All of Hum’s information is stored in the database, from user information to question data. The information in the database is both accessed and edited through PHP on the different sites. The communication between the pages, i.e. sending users from one page to another, uses HTML.

7.5 Alternative Design With No User Roles

One alternative design was to completely remove the user roles, i.e. not having users grouped as students and teachers. In this way, anyone could have created a course for other users to join. However, this could have made the system more complicated to use as users could not have been guided in the way as they are now: teachers are directly sent to course management when logging in while students are sent to the live feed. Thus, a new user that quickly wants to start asking and upvoting questions would have needed to navigate the menu to reach the Live feed instead of being directly redirected to it.

8 Requirements and Evaluation Methods

Hum was meant to satisfy a number of important specifications; one of them is that the web application needs to be fast and simple to use. In other words, the time it takes for a student to log in and start writing and viewing questions must be minimal (for this project, minimal time was said to be less than five seconds). Thus, studies were performed on students that used the system for the first time. In these test, the time between logging in and viewing questions on a live lecture was measured.
Moreover, the system would need to be simultaneously functional for just a couple of students if used in just a small class, as well as for a large group if used on a crowded lecture. Nonetheless, the system would probably not be required to handle more than 300 students at once per lecture as most lecture halls seldom exceed that amount of seats[35]. This was according to our own experiences as the application was developed on Uppsala University. If the system were to grow and expand in popularity and usage, the system would need handle more users connected at the same time.

In order to measure the usability of the system for multiple simultaneous users, we performed tests where twelve users used the system to enter and upvote questions concurrently. These users used different devices and web browsers, including Safari for iOS, Safari for Mac OSX, Chrome for Android, Chrome for Windows, Chrome for Mac OSX, Firefox for Windows and Edge for Windows. For this project, we did not have the possibility to test the system in a lecture with a larger audience. In addition, an automated test (using the web site http://www.webpagetest.org[14]) was used to test the Live Feed page.

The Library part of the system would need to have a higher capacity for active users as a larger number of students might simultaneously use it. However, its responsiveness is not of as high importance, mostly due to the fact that the database does not have the same requirements of instant live updates as the live question feed during lectures. As the Library was not fully developed, we could not test this part of the site.

9 System Overview

Hum consists of three main parts: the database, the back-end part and the front end part. The database stores all information used on the website, the back-end part communicates with the database by sending and receiving information and the front-end part consists of what the user actually sees: animations, text and similar. All code in the application are written by us, but however, some Wordpress functions (described in section 9.1.1) and some frameworks and libraries (see sections 9.3.1 and 9.3.2) were used.

9.1 Database

The database was created using Wordpress own database system, leaving the focus of the development on creating the communication module and the front-end system.
9 System Overview

Figure 3: ER diagram of Hum’s database. The rectangles represent the different tables, and the topmost texts in bold are the names of them. The text inside the rectangles are the names and types of the different fields. The lines between the tables represent the connections between them.

Looking at Figure database, most of the tables are from the included database in Wordpress. However, some fields and tables were added. Starting with the table wp_posts, which is the data for the questions, two fields were added: post_rating and post_lecture. The first is used for keeping track of questions’ upvotes, and the second is used for keeping track of which lecture the question belongs to.

The table wp_users is used for keeping track of users’ information, and no new fields were added to this table. The tables wp_users and wp_posts are connected to the table wp_upvoteusers which is used for keeping track of which users that have upvoted which questions. In this way, a user can only upvote a question once.

The tables wp_lectures and wp_courses were created specifically for Hum. A teacher can create a new course which students can join, and the information about the course is stored in wp_courses. In addition, the information about which users that has joined a course is stored in the table wp_student_course. In the same way as the course creation, teachers can create lectures belonging to a course, and the information about these are stored in wp_lectures. Questions are connected to lectures as can be seen in the Figure 3.
9.1.1 Wordpress

Wordpress[37] was a useful system when implementing Hum. This was because it yields plugin architectures and template systems for web site construction. Wordpress provides an open source content management system based on PHP, a script language described below, and database management using MySQL[11]. Wordpress’ internal database structure could therefore be used for storing both user and question data.

When registering a new user in the system, Wordpress’ PHP function `wp_create_user()` was used. A description of both this and the following Wordpress functions can be found in the Wordpress Codex[15]. In the same way, `wp_login_form()` was used for the log in page. Furthermore, at various page of the application, the function `get_userdata()` was used in order to retrieve information about the current user via Wordpress’ content management system.

9.2 Back-end

PHP-methods, e.g GET and POST, was used to connect the client with the database. The back-end’s responsibility was to identify the user, post its messages and work as a overall information gatherer for all clients.

9.2.1 PHP and HTML

PHP[13] is a back-end script language designed for web development. This language was used for communication between the client and the database server. PHP is often used together with HTML, the standard markup language for web pages. PHP generates and processes content on a server and returns it to the clients’ web browsers where it is displayed using HTML. Benefits of using PHP are that it can open up many interactive features that would not be implementable only by HTML.

9.3 Front-end

This part of the system is used to give the user a interface for orientation. The front-end system was built using HTML, and also JavaScript, which can enhance the clients’ interaction with the application and add animations. CSS framework called Bootstrap was also included to make the web application applicable for all devices and for a uniform application design.
9.3.1 jQuery

jQuery[8] is a JavaScript library that was implemented in order to simplify HTML modifications and animations. This library was used for animations in web application, granting us the ability to develop a more interactive site. Additionally, one part of the jQuery library that was used is AJAX (Asynchronous JavaScript and XML). Using this library of functions, a data from our database can be loaded and updated in the background without reloading the whole page.

9.3.2 Bootstrap

Bootstrap[1] is a Front-end CSS framework tool for websites and web applications. The framework helps in making websites dynamic, i.e. adapts them for devices with different screen sizes. This was a crucial feature when developing Hum as it should be used on various devices with various screen sizes, and thus, Bootstrap was included in the project.

10 Pages

When the user browses the web application, several pages are available; a flow chart of the different pages and the navigation between them can be seen in Figure 4. In addition to the navigation lines in the figure, users can log out from the system and be redirected to the Log In page from anywhere in the application. Most of the pages are available to all users. However, teachers will have more functionality in some pages as opposed to students as well as one page only available to teachers.
Figure 4: A flow chart of Hums’ different pages. The rectangles represent different pages and the navigation lines between them represent how different kinds of user can navigate between the pages.

10.1 Log In

Users not logged in to Hum are redirected to the Log In page when accessing the site. This page consists of two text fields and two buttons as can be seen in Figure 5. The text fields are used for entering the log in credentials. After entering a correct log in, pressing Enter or clicking the Log In button will log in the user to the application. If the user did not write a correct combination of username and password, the Log In page is reloaded with one of three different error messages indicating either that the log in credentials was incorrect or that the user did not enter a password or a username.
Figure 5: The Log In page of Hum. Users’ username and password are entered in the respective fields. The Log in button is used to log in to the system using the entered information, and the Register button is used to register a new user for the system.

10.2 Registration

The other part of the page (which can be seen in Figure 6), used for registering new users, is accessed by clicking the Registration button on the Log In page. Here, the user can register to the site by entering the desired username, password and e-mail address, and afterwards, pressing the Sign Up button. This button is only available when the user has entered valid information (a unique username, a password, and a valid e-mail address), and this can be seen in Figure 7. Also, after pressing the Registration button, the user can navigate back to the Log In view by pressing the Back to Log In text. The log in and registration functions were implemented with the help of Wordpress’ built in functions `wp_login_form()` [6] and `wp_insert_user()` [5], respectively.
Figure 6: The Registration page. Users’ desired username and password are entered in the respective fields. The Back to log in text is used to take users back to the login page.

Figure 7: The Registration page, when valid information has been entered. The text under the first field indicates that the entered username is available. The Sign Up button shows up after the user has entered valid information. Clicking this button creates a user account with the entered credentials and redirects the user to the Log in page.

10.3 Live Feed

The Live Feed page is only available when a lecture is live, and the page is unique for each course. In other words, if a user is registered to two different courses that each has a live lecture ongoing, he must choose to view one of the two courses’ Live Feed pages. Yet, as the Live Feed page is meant to be used while students are participating in a lecture, viewing two Live Feed pages simultaneously will not be necessary as a student can not participate in two lectures concurrently.
The page consists of a text field where students can enter new questions along with previously asked questions from all users on the lecture, as can be seen in Figure 8. Each question will have an upvote button that users can push in order to upvote a question. Users can only upvote each question once, and the questions in the list will be arranged by the number of upvotes each question has gotten, from most to fewest. The question list is retrieved from the database, and new questions and upvotes are sent to the database.

![Figure 8: The Live Feed page. Users can enter questions in the text field and submit them using the Submit button. The texts under the Submit button is previously entered questions along with a number indicating the number of upvotes the question has received and a upvote button.](image)

### 10.4 Library

On the Library page, users can view questions from previous lectures, and the questions are first grouped by courses. By clicking on a course, the course’s lectures are shown, and by clicking on a lecture, the user is redirected to that lecture’s Question page. However, users can only view courses that they are registered to, and thus, user can only access Question pages belonging to courses they are registered to.

### 10.5 Question page

On the Question page, the questions for the chosen (on the Library page) lecture is viewed together with both teacher’s answers and student’s discussions. When a teacher views a question page, he or she can write an answer that will show up under Teacher’s
Answers. On the other hand, when a student visits the page, he or she can write a discussion post that will show up under Student’s Discussion. The questions, answers and discussions are all both stored and gathered from the database.

10.6 Settings and Course Registration

From the menu, users can access both the Settings page and register for new courses. The Settings page can be seen in Figure 9, and here, users can change their user information. Again, this information is stored in the database. Course registration is simple, when clicking the Course Registration button in the menu, users are prompted with a text field where they can enter a course code provided by the course teacher as can be seen in Figure 10.

Figure 9: The Settings page, used for changing users’ settings. By typing in values to the desired text fields, the corresponding user information are changed to the entered value when clicking the Update button. However, in order to change any user data, the field Old password must be filled out with the user’s current password before clicking the Update button.
10.7 Course and Lecture Creation

An additional page, only accessible by teachers, is the Course and Lecture Creation page. Here, teachers can create and manage courses and lectures. A course is created simply by entering a course name and a course code and the code is used for letting students join a course. When a course has been created, the teacher can create lectures in that course. A lecture has a name, number, a start time and an end time. The start and end times are used for deciding when the lecture should be live, i.e. when students should be able to enter and upvote questions.

11 Evaluation Results

By discussing with Dr. Black-Schaffer[16], the lecturer’s point of view was taken in mind when designing the part of Hum which lecturers would use. As many lecturers on university level’s prime focus is research, there is not always enough time for improving lectures. However, this do not mean that lecturers do not want to improve lecturers; the lecturers that do research on a subject may be devoted in teaching about that topic. The problem here is primarily that research often tends to take the overhead of the lecturer’s time.

When designing Hum, we wanted users to be able to post questions quickly after logging in, and thus, this was focused on during the design process. Naturally, the easiest way of implementing this was found to be by simply redirecting users to the Live Feed page and the question asking form immediately after logging in. Regarding the tests performed
on the systems usability, were a user should be able to efficiently reach the Live Feed page, the results met our requirements. Three first time users were all able to reach the question asking form and the upvoting system within a few (less than five) seconds after logging in.

### 11.1 Site Speed and Efficiency

Tests were performed with students who were using the application for the first time, and the results were overall positive although some problems occurred. The system was able to manage the workload as it did not crash and kept on executing the tasks given by the users. However, Hum did experience slower response time. This was when the users, during the tests, tried to stress the system as much as possible on the Live Feed page. In these tests, the application was successfully able to handle up to twelve users that entered and upvoted questions at the same time. There were some bugs that occurred. For instance, at one time, a user could not enter a question without reloading the page. Another bug that occurred was that some users were able to upvote questions twice. Apart from these bugs, the system was fully functional for all users during the test, and thus, Hum successfully handled the test.

In order to improve the efficiency of Hum, i.e. to make it both fast and easy to use, different ways of updating the web application was evaluated. For instance, using JavaScript in order to update specific information on our site instead of refreshing the whole page makes the website both faster and easier to use. First, it makes it faster simply because less information is downloaded to the client, and second, the user can continue to use some parts of the site while other parts are updated. One example where this is necessary is the Live Feed page. Here, the text field where users enter questions must be available while new questions are fetched from the database and viewed on the page.

Furthermore, in our speed test using [webpagetest.org](http://webpagetest.org), the primary page of Hum, Live feed, was tested in order to get statistics of the web application. The results shown in Figure 11 confirmed that our site had a good response time from various locations around the world by passing in connection setup time, connection duration and file compressing. Cache static content and CDN (content delivery network) was not implemented meaning that the users do not cache any data while browsing the website and that no data is saved on any DNS-servers. Although these are examples good optimization, Hum’s performance would not benefit significantly from this due to the application’s small amount of data usage. A further globalization of Hum would probably require these features to be implemented.
11.2 Server Capacity

Satisfying the requirement of having many simultaneous users connected at the same time was discussed in two different parts; one of them was viewing questions and answers in the library. Here, a bigger number of users are updating and viewing the database during a wider time span. The other part is the usage of Hum during live lectures, when the database is updated frequently by a smaller number of users over a shorter period of time. In addition, there could possibly be multiple live lectures for different courses going on at the same time, making the requirements of the server even larger.

In our test of the Live Feed, the application successfully handled twelve users that entered and upvoted questions at the same time. There were some bugs that occurred. For instance, at one time, a user could not enter a question without reloading the page. Another bug that occurred was that some users were able to upvote questions twice. Apart from these bugs, the system was fully functional for all users during the test, and thus, Hum successfully handled the test.

12 Results and Discussion

Hum was developed as a web application, and thus, it was made to be available to most devices with a web browser. However, the application was not tested with all possible web browsers, and therefore, we cannot exclude that there are a web browser where our system is not fully functional. Nevertheless, the system was fully functional on most modern browsers (including Safari for both Mac and iOS, Google Chrome for Mac, Windows and Android and both Mozilla Firefox and Edge for Windows), and therefore, the result is satisfactory.

When posting questions and comments or upvoting questions using Hum, the user’s identity is never shown, and hence, Hum successfully lets users be anonymous while
using the application. However, this do not include the database where the user’s identity is stored together with the questions and comments that the user has written. This database is only visible by the developers of Hum, and consequently, the anonymity of users of the system is preserved.

Due to various reasons, students may not always ask the questions that arise during lectures. However, the creation of Hum provided a tool that let students raise their thoughts anonymously without being exposed in front of the crowd. Through the use of technology, students can now ask questions they otherwise may not have dared to ask.

When using the Library, both students and teachers can browse previous lectures and show questions. Further, they can write discussions and answers to the questions, and hence, the Library can also help in improving lectures. This part of the site will also be one way in which lecturers can get feedback about what students find difficult in lectures. Teachers can also get live feedback during lectures by viewing the Live Feed page. Although the Library has not been tested properly in actual courses, Hum has provided the ability to give lecturers feedback and let teachers and students view and discuss questions after lectures. Thus, these results are satisfying.

Hum was not tested during an actual lecture, and thus, we can not see if our project possibly could enhance the overall quality of lectures. Similarly, we can not see if lecturers could benefit from using the app by getting feedback of which part of a lecture students found hard to understand.

13 Conclusions

Hum was developed as a web application available for most modern web browsers which allows users to be anonymous while using the app. In this way, the application can be used as a tool for both modernizing lectures and letting students be anonymous while asking questions. This is a crucial step when improving traditional lectures as it helps them to keep up with modern alternatives, such as online lectures.

By choosing to create Hum as a web application, Hum’s users did not have to use a specific operating system. This helped Hum to broaden its target group and enabled most students to use the Hum through most modern browsers and operating systems.

The Library page was successfully created in order to grant both teachers and students with the ability to browse through previous lecture questions. For example, students can use this feature in order to read up before an exam. Also, teachers can use the Library in order to get feedback of what the students found difficult during the course.
Although Hum provides the features mentioned above, the system was never properly tested during an actual live lecture. This prevented us from confirming whether the web application could enhance lectures in general. However, the system did provide the functionality to strengthen lecturing and encourage question asking among students. In conclusion, Hum could possibly be an important step when both modernizing lecturing and improving education in general by helping students to ask questions in a new way.

14 Future Work

As Hum was meant to be accessible for every student, teacher, lecturer and alike while simultaneously being an easy to use and fast system, the main focus for future developments is to create specific applications for each operating systems. This would provide users with a quicker and more personalized way to use Hum.

In addition to the upvoting functionality, a future development would be to add some kind of Natural Language Processing[12] plug-in that could group similar questions with each other. In this way, different questions regarding the same subject could be emphasized in the live feed, and teachers could answer similar questions at the same time.

The usability of Hum was always a main focus during development. This was since we wanted users to always be able to interact with the application without being distracted from the actual ongoing lecture. Therefore, it would be helpful to seek assistance from both experienced web application designers and human computer interaction specialists. These people would be consulted in order to improve Hum’s general user interaction as well as its usability, and together, this would increase Hums’ overall effectiveness.

Furthermore, an additional way of logging in to the system has been discussed. The website Socrative[26] has a function which lets users log in without having to create an account to the website. In contrary, when using Hum, students have to create an account with a personalized login before using the application. Socrative’s implementation has positive effects as well as drawbacks; users will get a more accessible way of interacting with the system, but however, users might also exploit this function to create numerous logins per user. Hence, users could possibly sabotage the system. Consequently, a function like this has to be thoroughly examined and discussed, and in this way, it may be finalized in a way that will be suitable for Hum.
References


References


