A mobile app for submitting questions during a lecture

Ioannis Mimtsoudis
H00141571

August 2013

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for
the award of the degree of MSc in Advanced Internet
Applications
Declaration

I, Ioannis Mimtsoudis confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:
Date:
Dedicated to my parents.
Abstract

Achieving active learning and increasing interactivity in traditional classrooms is not an easy process. Several electronic educational systems have been developed and used in order to achieve it. The most recent achievements in this area are mobile and web based systems. These systems mainly allow students to answer multiple choice questions posed by the lecturer. None of them allows a student to initialize an interaction by posing a question.

Identifying this gap in existing technologies, an application for asking questions in a classroom was developed, taking advantage of mobile and web/cloud based technologies. The purpose of this application is to facilitate interactivity in a classroom. An Android application was developed in order to enable students to submit questions during a lecture. Questions are displayed on a screen visible by everyone in the classroom. Furthermore, questions are processed in order to avoid duplicated questions on the screen. Last, a web application was developed in order to trace questions and allow a lecturer to get feedback.

The evaluation of the questioning system has revealed a number of interesting results. To begin with, the application would motivate students to ask questions and thus increase interactivity. Also, particular student groups would be benefited most, such as shy, unconfident and foreign students. Furthermore, results indicate that a balance between personal and electronic interaction is necessary. Additionally, reputation points could motivate students to use the application and increase classroom interactivity and participation as physical presence is required. The majority of evaluators reported that they would have used the application if it was available in a classroom. Last, results indicate that males are more skeptical than females about the usefulness of the application.

Furthermore, this project includes a literature review, explanation of professional, legal, ethical and social issues, requirements analysis and future recommendations.
Acknowledgements

I would like to express my great appreciation to my supervisor Nick Taylor for his valuable and constructive suggestions during the planning and development of this dissertation. His willingness to give his time so generously has been very much appreciated. Furthermore, his patient guidance, enthusiastic encouragement and useful critiques of this research work were of great importance in order to keep my progress on schedule and lead me to the completion of this dissertation.

My grateful thanks are also extended to people of PUMA lab for their help and support in technical issues in the beginning of the project. Also, their help in offering me the resources for conducting the evaluation was of great importance. Furthermore, I would like to thank the second reader of this dissertation Judy Robertson for her valuable feedback on the research report and for reading this dissertation.

My special thanks are extended to my family for their unconditional love and support throughout my studies. Last but not least, I would like to thank my friends and especially my girlfriend for their psychological support as well as for their grammar overview and spelling.
# Table of Contents

Abstract ........................................................................................................................................... 3
Acknowledgements .......................................................................................................................... 4
List of Figures .................................................................................................................................. 7
List of Tables .................................................................................................................................... 8

1. Introduction .................................................................................................................................. 9
   1.1 Introduction of the problem....................................................................................................... 9
   1.2 Aim and Objectives ............................................................................................................... 10
   1.3 Project’s Outline .................................................................................................................... 10

2. Literature Review .......................................................................................................................... 12
   2.1 Traditional Classrooms .......................................................................................................... 12
   2.2 The Importance of Asking Questions ..................................................................................... 13
   2.3 Active learning and technology ............................................................................................ 14
   2.4 Audience Response Systems ................................................................................................ 15
      2.4.1 Hardware based response systems .................................................................................. 15
      2.4.2 SMS based response systems ......................................................................................... 16
      2.4.3 Web/cloud based response systems ............................................................................... 17
      2.4.4 Benefits of using Response Systems .............................................................................. 20
   2.5 Real Time Web ....................................................................................................................... 21
      2.5.1 The web and real time issues ........................................................................................... 21
      2.5.2 Server Push Techniques ................................................................................................ 22
      2.5.3 WebSocket ..................................................................................................................... 25
   2.6 Critical Analysis of Literature Review .................................................................................... 34

3. Professional Legal, Ethical Issues and Social Issues ..................................................................... 36
   3.1 Professional Issues ................................................................................................................ 36
   3.2 Legal Issues ............................................................................................................................ 36
   3.3 Ethical/Social Issues .............................................................................................................. 36

4. Requirements Analysis ................................................................................................................ 37
   4.1 Mandatory and optional functional requirements .................................................................... 37
   4.2 User requirements .................................................................................................................. 38

5. Design .......................................................................................................................................... 39
   5.1 System Design ....................................................................................................................... 39
5.2 Database Design........................................................................................................42
5.3 Android Application Class Diagram.......................................................................43
5.4 User Interface Design...............................................................................................44
  5.4.1 Android application............................................................................................44
  5.4.2 Question Screen ...............................................................................................46
  5.4.3 Lecturer Management Screen ..........................................................................47
6. Implementation ........................................................................................................50
  6.1 Client – Side ..........................................................................................................50
    6.1.1 Android application .......................................................................................50
    6.1.2 Web application ............................................................................................58
  6.2 Server – Side .........................................................................................................61
    6.2.1 WebSocket Server .......................................................................................62
    6.2.2 Question - filtering algorithm .....................................................................63
7. Evaluation ..................................................................................................................67
  7.1 Evaluation Method .................................................................................................67
  7.2 Findings ................................................................................................................68
  7.3 Discussion of results .............................................................................................84
8. Further Work .............................................................................................................87
9. Conclusion ................................................................................................................88
References ......................................................................................................................90
Appendices ....................................................................................................................93
  Appendix A : Questionnaire ......................................................................................93
  Appendix B: Student responses in question 11 ........................................................96
  Appendix C: Student responses in question 13 .........................................................97
List of Figures

Figure 1: Clickers ................................................................. 16
Figure 2: HTTP request/response protocol ....................................... 21
Figure 3: Semi busy polling ..................................................... 23
Figure 4: Long polling .......................................................... 23
Figure 5: WebSocket clients connected to a server .............................. 26
Figure 6: Polling vs WebSockets ............................................... 26
Figure 7: WebSocket upgrade handshake ...................................... 27
Figure 8: WebSocket frame header ............................................. 28
Figure 9: Execution Order of a Ratchet script ................................. 32
Figure 10: Deployment Diagram ............................................... 41
Figure 11: Database diagram ................................................... 43
Figure 12: Android application class diagram ................................ 43
Figure 13: MainActivity screen states ........................................ 44
Figure 14: RepPointsActivity screen ......................................... 45
Figure 15: LoginActivity screen states ........................................ 45
Figure 16: RegistrationActivity screen ....................................... 46
Figure 17: Question screen ..................................................... 47
Figure 18: Lecturer Management Screen - Search results .................... 48
Figure 19: Lecturer Management Screen - Question Screen options ........ 48
Figure 20: Android, PHP, MySQL Interaction ................................ 51
Figure 21: WebSocket Connectivity .......................................... 54
Figure 22: Question feedback .................................................. 56
Figure 23: Question Screen - Question Wall ................................ 59
Figure 24: Use of class/audience response system ........................... 69
Figure 25: Shows if AskFree would motivate students to ask questions .... 69
Figure 26: Shows if AskFree would motivate students to ask questions by gender .... 70
Figure 27: Shows the frequency of students who think that AskFree would support students that are not confident to raise their hand and ask a question .......... 71
Figure 28: Shows the frequency of students who think that AskFree would support students that are not confident to raise their hand and ask a question by gender ...... 71
Figure 29: Shows if students think that AskFree would help shy and unconfident students integrate in the class and become more sociable .......................................................... 72
Figure 30: Shows if students think that AskFree would help shy and unconfident students integrate in the class and become more sociable by gender .............................. 72
Figure 31: Shows the frequency of students who believe that AskFree would improve the student classroom experience .......................................................... 73
Figure 32: Shows the frequency of students who believe that AskFree would improve the student classroom experience by gender .......................................................... 73
Figure 33: Shows the number of students who believe that AskFree would improve the student-teacher interaction ............................................................................. 74
Figure 34: Shows the number of students who believe that AskFree would improve the student-teacher interaction by gender ............................................................................. 74
Figure 35: Shows the number of students who believe that AskFree would improve the capability of teachers to identify and support students with learning difficulties.

Figure 36: Shows the number of students who believe that AskFree would improve the capability of teachers to identify and support students with learning difficulties by gender.

Figure 37: Shows if reputation points can motivate students to use AskFree.

Figure 38: Shows if reputation points can motivate students to use AskFree by gender.

Figure 39: Shows the ease of use of AskFree.

Figure 40: Shows the ease of use of AskFree by gender.

Figure 41: Shows if question anonymity encourages student participation.

Figure 42: Shows if question anonymity encourages student participation by gender.

Figure 43: Shows the number of students who would have used AskFree and those who would not.

Figure 44: Shows the number of students who would have used AskFree and those who would not by gender.

Figure 45: Shows how helpful evaluators found the question feedback and if it was appropriate for the questions they asked.

Figure 46: Shows how helpful evaluators found the question feedback and if it was appropriate for the questions they asked by gender.

List of Tables

Table 1: Positive responses

Table 2: Negative responses
1. Introduction

1.1 Introduction of the problem

The traditional lecture theatre where a lecturer delivers the teaching materials to a large group of students is the current state at most universities. This is a cost efficient and scalable way of teaching; however it comes with a price. Interactivity has become difficult and inefficient, leading to decreased student motivation, participation, engagement and learning. Student’s participation is often limited to passively listening and taking notes. Furthermore, the capability of a lecturer to get feedback from students and hence identify and help those with learning difficulties is also reduced (Draganova, 2009; Scornavacca, et al., 2009).

Active participation in the learning process helps students to absorb more knowledge. This can be achieved using interactive classrooms where students ask and answer questions, give feedback and participate in class activities. It has been proven that classroom interaction improves student’s attention and allows them to reflect on the content presented, thus enhancing learning (Draganova, 2009). However, it is difficult to apply these practices in traditional classrooms and the main reason is the large number of students.

During the past years in order to increase interactivity in traditional classrooms several technologies have been used, with class response systems (clickers) being the most known. On the down-side, these systems have a number of limitations (Kay & LeSage, 2009). The use of technology in classrooms provides a number of benefits for both students and lecturers (McDonald & Bulchandani, 2011).

The widespread use of mobile devices such as smart phones, tablets and laptops in classrooms provide an opportunity to design a new generation of systems for providing active learning in traditional classrooms. Combining mobile internet enabled devices with web and cloud technologies allows the development of ubiquitous classroom systems.
1.2 Aim and Objectives

The aim of the project is to develop a classroom questioning system that will facilitate interaction in a classroom setting.

The objectives of the system are:

- Develop an Android application that enables students to type and send questions.
- Develop a screen to display questions once they are sent.
- Organize the way multiple questions are displayed on the screen when they are sent in a specific time of each other in order to avoid duplicate questions.
- Develop an application that will enable a lecturer to trace questions and hence receive feedback from students, increasing his/her capability to identify students with learning difficulties.
- Investigate whether the questioning system would be beneficial for learning in a university classroom.

1.3 Project’s Outline

The project’s structure is formed in the following chapters:

Chapter 2

Chapter 2 contains a literature review and critical analysis of classroom issues, audience response systems and real time web.

Chapter 3

Chapter 3 discusses the professional, legal, ethical and social issues of the project

Chapter 4

Chapter 4 contains the requirements analysis of the system.

Chapter 5
Chapter 5 demonstrates the overall design decisions taken for the implementation of the system. The Android class diagram, database and user interface design decisions are discussed.

Chapter 6

Chapter 6 describes the implementation of the system. Specifically, development decisions, problems that were faced and addressed are explained in detail.

Chapter 7

Chapter 7 discusses the evaluation of the system. The evaluation method, findings and discussion of results are provided.

Chapter 8

In chapter 8 recommendations for future work are made.

Chapter 9

Chapter 9 discusses the conclusions and achievements of this research project.
2. Literature Review

2.1 Traditional Classrooms

Nowadays, the dominant way of teaching at universities is one lecturer delivering the teaching materials to a large group of students (hundreds in some classes). This is a cost effective way of teaching, however, this results in making interaction between students and lecturer difficult and inefficient. The possibility for students and lecturer getting engaged in a meaningful conversation vanishes; students usually participate passively by taking notes or listening. This leads to decrease of student engagement, motivation and learning (Draganova, 2009; Jones, et al., 2009).

Boyle & Nicol (2003) state that asking and answering questions, discussing and debating are critical in the learning process. However, as the number of students in a classroom increases, the amount of time available for meaningful interaction decreases. It is impossible for all students in a big class to ask a question or for a lecturer to get everyone’s answer or comment. Moreover, the capability of a lecturer to identify and support students with learning difficulties is limited due to zero or limited feedback a lecturer gets.

As Stowell, Oldham, & Bennet (2010) argue, there is less participation in larger classes and the most important reason is that students don’t want to appear less intelligent to other students. Furthermore, students hesitate to openly express their diverse opinion in discussions of controversial issues when they feel negative reactions from their peers and are afraid of being stereotyped and come into conflict. Shyness and anxiety are also responsible for not actively participating in a classroom; many students feel uncomfortable raising their hand and ask or answer a question. Furthermore, shy students experience higher levels of social anxiety, inhibition, fear of judgment and interpersonal rejection; these effects lead to low academic performance (Ulbig & Notman, 2012) and social exclusion. Shy students experience difficulties to absorb lecture material presented in a more social environment as big lecture theatres and are less enthusiastic about classroom activities.
2.2 The Importance of Asking Questions

Marshal Brain (Brain, n.d.) argues that the process of questioning is a fundamental concept on which research relies on. A student during his or her studies constantly asks basic questions concerning facts and techniques of his or her discipline. Then these questions are answered by the student in a unique and original way.

Moreover, Marshal Brain acknowledges the importance of questions in the research process and asserts that students at a university should be taught how to ask good questions and how to answer them in the right manner.

The previous assertion, points out the importance of the questioning process in every class taught at a university. There are two reasons that questions are important to students in a class:

1. Asking questions helps students to learn how to ask questions. Furthermore, in order to learn how to ask good questions students need to get feedback on questions they have previously asked. By learning to ask good questions students learn to become scholars.

2. At the moment a student asks a question he or she becomes a self-motivated learner, a researcher and this is the behavior the universities want to nurture.

Questions are also important to a teacher:

1. Questions help teachers to understand if their students understand and are thinking about the covered material. Questions also allow a teacher to understand if the class is awake or asleep.

2. If students ask questions about concepts they don’t understand, the teacher gets immediate feedback and can spend more time explaining these concepts.

3. Students attend classes in order to participate in a dialog between them and the teacher. This is the essence of education and asking questions is an important part of this dialog.
2.3 Active learning and technology

The widespread use of technology at universities and especially the Internet enables many lecturers to provide students with electronic learning material, assignment descriptions, podcasts and even discussion forums or chat. As more material is provided electronically students don’t go to the classroom expecting to keep pages of notes from the lecturer’s monologue while being passive listeners. Today’s students are a generation that has grown up surrounded by technology, including laptops, mobile phones, social network sites, etc. in which they are creating and engaging their social world, they feel comfortable to use it in every context of their life and expect to use it in the classroom too. This technology is highly interactive and students are used to be active learners motivated by doing and getting immediate feedback (Guthrie & Carlin, 2004). The way of learning has changed over the years; focus and concentration tend to be replaced by multitasking and simultaneous flow of information (Stav, et al., 2010).

The use of portable devices such as smart phones, tablets and laptops in classrooms is becoming a norm, enhancing the learning experience. Taking advantage of the free Wi-Fi provided by universities, students use mobile computing devices to search for information on the Internet, communicate and collaborate.

As we discussed above, the dominant approach of teaching at universities is large-scale lectures. The growing demand of society for higher education will not reverse this trend that has been proven to promote surface learning and is the bane of active learning, engagement and involvement (Stav, et al., 2010).

Students absorb knowledge when they actively participate in the learning process by asking, giving answers or doing an activity. Classroom interactivity benefit both students and lecturers, it promotes an active learning environment, provides feedback for the lecturer, increases student motivation and enables a learning community (Scornavacca, et al., 2009). Classroom activities and especially multimedia presentations help students acquire and retain more information (Ulbig & Notman, 2012). However, interactivity in large classes is not-trivial. One way to overcome this challenge is the introduction of technology in order to allow students to be more involved in class and engage in active learning (Hunsinger, et al., 2008).
The question is how it is possible to combine modern technology and traditional teaching methods in order to engage students and promote active participation.

During the past years, several educational systems based on the audience response systems model have been developed and used in higher institutions to create interactivity between the teacher and students. Specifically to engage students, facilitate classroom interaction and enhance the learning experience.

2.4 Audience Response Systems

Audience response systems enable a group of people to give instant feedback that is displayed in real time, either anonymously or associated with each individual. The purpose of these systems is to create interactivity between a presenter and the audience (Oxford University Computer Services, n.d.). They have been mostly used in TV shows, political debates and education. Bing pulse has been used in the biggest live online poll in history. During president’s Obama speech 12.9 million votes have been done showing the reactions of millions of people in real-time according to gender and political party (Bing, 2013).

2.4.1 Hardware based response systems

The most common of these systems which is widely used in large classes to increase student’s participation and learning are known as Student Response Systems (SRS), Class Response Systems (CRS) or just Clickers. Student Response Systems are handheld devices that allow students to answer displayed multiple choice or yes/no questions. All answers are instantly displayed in chart forms enabling all students to see them. Responses are anonymous to their peers but the lecturer can associate every device with an answer. This technology allows a lecturer to get instant feedback from the class through collecting and analyzing large amount of data. Moreover, SRS’s have been used also for assessment purposes (Stav, et al., 2010; Kay & LeSage, 2009).
A SRS is usually composed by a receiver for lecturers, keypads (clickers) used by students to submit their answers and a software application. The application is installed on the lecturer’s computer to analyze, save responses or create interactive presentations. Several commercial systems exist with variations in functionality and communication means such as infrared or radio frequencies (Stav, et al., 2010).

![Figure 1: Clickers](image)

Hardware based SRSs have several technical and teaching limitations. Technical difficulties that accompany SRS technology involve the cost of purchasing the infrastructure devices, software and cost of installation. Moreover, students are required to purchase or rent the handheld devices which have no other practical use and remember to bring them in the classroom. The communication faces limitations in big classes where the signal of student devices may not be received by the receiver (Scornavacca, et al., 2009).

In addition, students can answer only to multiple choice questions, there is no option for open-ended answers or comments.

### 2.4.2 SMS based response systems

Active learning and student participation does not involve only answering questions but also asking or commenting, which is encouraged but not very common among students. A two way communication should be promoted between the learner and the lecturer.
Due to cost and technical limitations of SRSs, mobile phone solutions have been developed in the past. Systems like “txt-2-LRN” or “Edutxt” were developed with the assumption that every student owns a mobile phone and carry it in the classroom (Scornavacca, et al., 2009; Draganova, 2009). These systems are composed from a sms gateway that receives text messages connected to the instructor’s computer and a software management tool allowing the instructor to read incoming messages and automatically analyze the results of polls. Every student is able to submit a question, comment and answer any time during a lecture without interrupting the class. Questions that have not been answered during the class can be answered afterwards through sms or a discussion forum (Scornavacca, et al., 2009). Moreover, students can also answer on multiple choice questions and participate in quiz games.

According to the research of Scornavacca & Marshall (2009) over 90% of students who have used a sms based system (txt-2-LRN) in a big lecture theatre reported that the ability to send sms – based questions or comments during the class was useful. The lecturer of the class had also a positive perception of the technology, commenting that the quality and quantity of student feedback had increased.

The biggest disadvantage of sms-based systems is the cost of sms messages which burdens the students and the necessary, difficult to install and costly equipment on the instructor’s side.

2.4.3 Web/cloud based response systems

Today’s widespread adoption of mobile computing, applications and everywhere Internet access make the aforementioned systems look obsolete.

The new generation of audience response systems (Poll Everywhere, 2013; AnswerQwik, 2013; Promethean, 2013; mqlicker, 2013; TurningPoint, 2013) relies only on Internet access. A lecturer using a web or cloud based software system on a laptop, a tablet or a smart phone can create questions which may be open-ended or multiple choice. The questions are displayed during or after a presentation, within the presentation program, in a web browser or only on student mobile devices. Students connect to the software system via the Internet either from a web browser or an
application. They can respond on questions, comment or respond with a new question. Responses are transmitted via the Internet and displayed on a screen in real time; the provision of statistics is also an option. Responses are usually anonymously displayed, however, depending on lecturer’s needs students may be required to login into the system with their account. Moreover, responses are saved in a database and can be associated with each student, allowing the lecturer to analyze the data and extract useful outcomes.

The advantages of web based response systems are:

- Students don’t need to buy a specific hardware device (clicker), they can use any web enabled device such as laptop, smart phone or tablet.
- Responses are free, students don’t need to pay.
- There is no need for communication infrastructure such as receivers. This eliminates the cost of purchasing, installing and maintaining them for the university.
- Internet in classrooms can be accessed via the university’s Wi-Fi or 3G network coverage.
- There is no additional cost for question/response management software. A lecturer can access and analyze responses through a web page.

On the other hand, after reviewing web/cloud based solutions it is clear that they don’t allow a student to initialize an interaction with the lecturer by autonomously pose a question or a comment during a lecture.

The following chapters describe two of the most famous general purpose web/cloud based commercial audience response systems and one dedicated for educational purposes.

2.4.3.1 Poll Everywhere

Poll Everywhere is a commercial web-based audience response system. It enables a presenter to ask the audience a question using an online application. The audience answers in real-time using mobile phones, twitter or web browsers. In particular, the audience can answer the question in four different ways: 1.) Send a SMS text message
to a unique event id number, 2.) Use a smart phone’s web browser to access the application 3.) Send a tweet on twitter 4.) Use a computer or tablet web browser. The audience and the presenter can see the responses live on the web or in a presentation tool. Questions are in form of polls, but the presenter can choose what type of question to ask: 1.) Multiple choice question, 2.) Open-ended text reply. Responses for multiple choice questions are displayed in charts that are updated in real-time as people respond. Open-ended text replies are displayed on a text wall that is updated in real-time as people answer (Poll Everywhere, 2013). Poll Everywhere is a system that is built around a web application that can be accessed by a variety of means. It combines web-based with sms-based and social network technologies eliminating the need of receivers or any additional hardware or software. It allows anyone who has access to the web to set-up an audience response system at any place fast and easy and allows every participant of an event regardless of what device he/she carries to respond.

2.4.3.2 AnswerQwik

AnswerQwik is a cloud based audience response system. A presenter uses AnswerQwik online platform to ask a question to the audience that appears on their smart phone, tablet or laptop. The audience participates in the presentation in real-time and is able to answer on multiple-choice, true/false or scalar based questions. Multiple choice questions allow open-ended answers too. No hardware equipment is necessary. Native applications operate as clients and are available for iPhone, Android and Blackberry smart phones in order to respond. Smart phone apps are free for the audience and can be downloaded by the respective app store. In addition, responses through a web browser are also possible using a laptop or a tablet. The presenter can choose whether responses are displayed in real-time, after the poll has finished or just gather data for later analysis. Polls can be run at multiple sites simultaneously, or via webinar, since all polling works via the cloud. AnswerQwik can be used in a variety of events including conference, education, and business (AnswerQwik, 2013). It allows easy and fast set-up of polls using the web and a variety of means to respond,
including sophisticated smart phone apps, however, only web based responses are available.

**2.4.3.3 Socrative**

Socrative is an audience response system dedicated for classroom interaction. It aims to engage the class using any web enabled mobile device such as smart phone, tablet and laptop. Teachers use the online application to initiate an event and students connect to the specific event using their device in order to respond to a multiple choice, true/false, open-ended question or a quiz with instant feedback. Student Responses are instantly displayed on teacher’s screen, however they are not visible by students and no graphs are provided. Socrative is mostly used for educational exercises and games in order to engage the class or for assessment purpose. Student responses are logged and sent to teacher’s email (Socrative, 2013).

**2.4.4 Benefits of using Response Systems**

Response systems have been used in a variety of disciplines including engineering, medicine, finance, social science, physics, etc. and are well regarded by students and lecturers. Researchers have revealed a number of benefits.

Using response systems improves attendance. When students have a motivation such as responses accounting in the total grade, the attendance is even higher. Students reported that anonymity of their responses encouraged their participation because they could answer without being judged by their peers. This feature helps also the shy and unconfident students to participate more. Moreover, students reported that RS creates a classroom environment where they are more interested or engaged in concepts presented and discussed (Kay & LeSage, 2009), (Guthrie & Carlin, 2004). The quantity and quality of class discussion is increased, especially when used with “peer instruction” strategy (Kay & LeSage, 2009). An instructor gets immediate feedback from student responses which are traced to each student and saved in a
database which facilitates **data gathering and analysis**. From the received feedback the instructor can identify if students understood the concepts of the lecture and point out parts that are not clear or even modify the style of teaching. It is also easier to identify students with learning difficulties if they constantly respond wrong. Responses are **displayed in real time** and presented in chart forms allowing both students and lecturer to know how well the class understands the concepts or start a discussion. Learning **performance** is increased in classes using response systems as a result of active learning (Kay & LeSage, 2009). The ability to **ask questions** during the class is well regarded by students and has a positive impact resulting in more and better formulated questions (Scornavacca, et al., 2009). Furthermore, students argue that comparing their answers with their peers is positive as it creates competition (Kay & LeSage, 2009).

### 2.5 Real Time Web

#### 2.5.1 The web and real time issues

The web was perceived as a collection of HTML pages linking to each other to create a conceptual web of information. It was not build as a service to implement synchronous communication.

The web is based on the HTTP request – response protocol (figure 2) using the client-server model. Web clients initiate an interaction with web servers by establishing a connection and sending HTTP requests. Servers accept the connection and reply with HTTP responses (Loreto, et al., 2011).

![Figure 2: HTTP request/response protocol](image-url)
Web clients can easily pass messages to servers by posting the messages using POST or GET methods. On the other hand, servers are passive components, their purpose is to respond on HTTP requests and they are not able to initiate an active interaction with clients or send an unrequested HTTP response. Servers can not initiate server push because HTTP was not created to be used for bi-directional communication (Deitel, et al., 2012; Taylor, 2013; Loreto, et al., 2011).

Nowadays, most of the web applications require real-time communication of data between a client which can be a browser, an application or a mobile application and the server. Some examples are, chat applications like the facebook chat, live sport result applications, financial data such as stock prices, medical device readings, gaming, etc.

However, many problems arise in the context of real time web applications. Some of them are:

- At the time the user sees the data on the webpage, it may already be outdated.
- Refreshing the page continuously and manually to get new data is not a good solution; it results in poor user experience. (Lubbers, n.d.)

### 2.5.2 Server Push Techniques

To overcome the bi-directional communication problem, several techniques that emulate server push have been developed. These techniques are known as Comet and primarily use AJAX to provide real-time bidirectional communication between a client and a server. The most common are semi-busy polling, long polling and streaming.

- In semi busy polling (figure 3) the client frequently asks the server via AJAX if there is new data. The server replies immediately whether there is new data or not. However, frequent polling causes inefficient use of resources when no data is transferred between request and replies. Moreover, the responsiveness of the application can be reduced. The reason is that until the client makes a new poll to the server, the data is queued.
In long polling (figure 4) the client makes a long lived call to the server via AJAX keeping the connection open until it times out. The server responds only when there are new data available, and then the connection is renewed with a new HTTP request. This technique minimizes the latency in message delivery because there is always a pending request to which the server can reply once a new event occurs. On the other hand, when a high volume of data is transferred there is no substantial performance improvement compared to semi-busy polling (Loreto, et al., 2011).
proxy servers and firewalls, causing an increase of the message delivery. This is the biggest disadvantage of streaming compared to long polling. An alternative solution is the use of TLS connections to shield the response from being buffered, but this increases the resource consumption (Lubbers & Greco, 2010).

All methods have advantages when compared with each other; however none of them provides full duplex bi-directional real-time communication, where both client and server can exchange messages at exactly the same time.

Furthermore, Comet methods result in several problems:

- It is necessary for the server to use two TCP connections for each client. One for upstream and one for downstream. The price to maintain and coordinate these two connections is resource consumption and complexity.
- There is a high overhead because each client request has an HTTP header.
- A mapping from the outgoing connections to the incoming connections should be maintained by the client-side script in order to track replies (Fette & Melnikov, 2011).

The WebSocket protocol together with the WebSocket API provides an alternative to HTTP polling for bi-directional communication between a client and a server. They address the above issues using a single TCP connection for traffic in both directions (Fette & Melnikov, 2011).
2.5.3 WebSocket

2.5.3.1 Introduction to WebSocket

The WebSocket protocol is designed to replace existing bi-directional technologies that use the HTTP protocol. WebSocket is a web technology that allows clients to communicate with servers and vice versa over a single TCP connection, eliminating the overhead of the HTTP protocol (Shwetan, 2012).

It uses its own protocol which is standardized as RFC6455 by IETF in 2011. Moreover, it also has an API that is called WebSocket API and it is standardized by W3C. Web applications can use the API to open and close connections and send messages (Fette & Melnikov, 2011; Hickson, 2013). It was initially designed as part of HTML5 to be implemented by browsers and servers; however, it can be used by any client server application.

WebSocket is currently supported by the latest versions of all major browsers, including Chrome, Firefox, Opera, IE 10, Safari and mobile browsers for Android, iOS and Blackberry (Can I use, 2013). If a browser does not support WebSocket it can use Flash polyfill (Ratchet, n.d.). The communication is done over TCP port number 80, which allows data to bypass firewalls.

It provides natural full duplex bi-directional communication between a server and clients. It allows the development of faster, more scalable and more robust high performance real-time applications on the web (Shwetan, 2012).

With WebSocket an HTTP request is used as a single request to establish a WebSocket connection (WebSocket ws:// or WebSocket over TLS wss:// ) and the same connection is re-used for client-server and server-client communication. Servers can push asynchronous data to clients only if there is an open WebSocket connection. A client initiates an interaction and establishes a connection with the server. A WebSocket server can also be used as a WebSocket client. Nevertheless, clients like browsers cannot accept a connection if they did not initiate it (Wang, et al., 2013). Figure 5 illustrates WebSocket clients connected to a WebSocket server, where either the server or clients can send real time data any time.
The utilization of WebSocket reduces latency, the reason is that after the connection is opened, the server can initialize an interaction and send messages once they are available. In contrast to polling, as figure 6 depicts, WebSocket makes a single request. There is no need for the server to wait for a request from the client in order to send data. Likewise, messages can be sent from the client to the server at any time. This single request, in addition to the natural server push, is one of the reasons WebSocket excels polling and reduces latency (Wang, et al., 2013). The key point here is that the server is able to push messages to clients once they are available instead of requiring the clients to check periodically for updates. This constitutes an evolution of the HTTP protocol, especially for real time event driven web applications.

Figure 5: WebSocket clients connected to a server

Figure 6: Polling vs WebSockets
2.5.3.2 How does WebSocket work

The WebSocket Protocol

The WebSocket protocol was designed to work efficiently with the existing web infrastructure. A WebSocket connection starts with a HTTP connection in order to verify full backwards compatibility with the non WebSocket world. This connection is called a ‘handshake’, in which a switch in protocols takes place, from HTTP to WebSocket (Wang, et al., 2013).

The client sends a HTTP request to the server. This request is similar to any other, except that it includes a special header called “Upgrade”. Sending this header, the client indicates that it wishes to upgrade the connection to another protocol. The figure 7 illustrates the opening handshake, including the required and optional headers.

![WebSocket upgrade handshake](image)

If the server “speaks” the WebSocket protocol, it agrees to the protocol switch through the upgrade header. At this point it is important to note that the server must support web sockets, it must be a WebSocket server. The HTTP connection stops to
exist and the WebSocket connection replace it over the same underlying TCP/IP connection. The same ports as HTTP (80) and HTTPS (443) are used by default (Wang, et al., 2013).

After the connection is established, server and client can communicate in full-duplex mode sending WebSocket data frames (figure 8). Text and binary frames are supported and can be sent in either direction at the same time. A frame of 2 bytes is added to the data. Each text frame starts with a 0x00 byte, ends with a 0xFF byte, and UTF-8 data is contained in between. A terminator is used by WebSocket text frames, while a length prefix is used by binary frames (Wang, et al., 2013).

![WebSocket frame header](image)

**Figure 8: WebSocket frame header**

Except of the opening handshake, WebSocket has also a closing handshake which indicates whether the underlying TCP socket closed intentionally or accidentally. A numerical code and a reason string that indicates why the socket has been closed are sent by the endpoint that is terminating the connection (Wang, et al., 2013).

**The WebSocket API**

The WebSocket API which is a JavaScript based API can be used to control the WebSocket protocol and create applications. Basically, it enables applications to use the WebSocket protocol and control a full-duplex communication channel through which they can send and receive messages.

**The WebSocket Constructor**
A connection to a server is established by instantiating a WebSocket object and pointing it to a URL that denotes the endpoint to which the client wants to connect. Two URI schemes are defined by the WebSocket protocol, ws:// and wss:// for the unencrypted and encrypted traffic between the client and the server respectively. The WebSocket constructor accepts two arguments, one required and one optional. The first argument is the URL to which the client wants to connect and the second is the protocol (or array of protocols) over which the client and server will communicate. Examples of protocols are SOAP, XMPP or a custom protocol. The following listing illustrates a WebSocket object instantiation (Wang, et al., 2013).

```javascript
var myWebSocket = new WebSocket("ws://www.websockets.org");
```

**WebSocket Events**

The WebSocket API is totally event-driven. WebSocket objects have event handlers which handle incoming data and changes in connection status. The application code listens for events. In addition, the WebSocket protocol is also event-driven, which means that the client application does not have to poll the server for updates. Events and messages arrive asynchronously one the server sends them.

Compared to the HTTP request/response model, where the server listens for incoming requests, a reversed asynchronous programming model is followed by WebSocket programming. With WebSocket, the client is the one that is listening for events as long as a WebSocket connection is open. To enable the application to start listening for events a callback function must be added to the WebSocket object. Four different events are available: Open, Message, Error and Close.

The open event is triggered after the server responds to a connection request. The open event indicates a successful connection, after that the WebSocket is ready to receive and send data. The callback function is onopen.

The data sent from the server is contained in messages. The message event is triggered when messages are received. WebSocket messages can handle text and binary data. Binary data are handled as ArrayBuffer or Blob messages. The corresponding callback function is onmessage.
When unexpected failures happen, the error event fires. The error event is associated with the onerror callback function. When an error happens the connection closes.

When the connection is closed, the close event is triggered and no more messages can be transferred between the client and the server. The associated callback function with the close event is onclose. The reasons why this event is triggered might be an error or a successful closing handshake (Wang, et al., 2013).

**WebSocket Methods**

There are two methods that can be invoked on a WebSocket object: send() and close().

Once a connection is established between a server and a client and after the open event is triggered, the send method can be called. The send() method can be used in order to send messages from the client to the server. The send() method sends data only when the connection is open, else an exception is thrown. The WebSocket API allows string (text messages), and binary data to be sent. Binary data is useful for implementing binary protocols such as standard Internet protocols layered on top of TCP, where the payload can be either an ArrayBuffer or Blob. Blob objects can be combined with the JavaScript File API for sending and receiving files, such as images, video and audio (Wang, et al., 2013).

The close() method is used to close the WebSocket connection or to stop an attempt to connect. After this method is called, no more data can be transferred along the WebSocket. Two optional arguments (a numerical status code and reason (String)) can be passed in the method to inform the server why the client has closed the connection (Wang, et al., 2013).

### 2.5.3.3 WebSocket server

A WebSocket server is an application listening at a specific port and transfers data to the clients from the server-end. It has the ability to hold many connections open at the
same time, being aware of every single connection and able to communicate with them at any time. In fact, WebSocket is a stateful protocol that beats the statelessness of the HTTP protocol (Mozilla Developer Network, 2013; Wang, et al., 2013).

A WebSocket server can be written in any server side programming language, with the restriction that it should implement HTTP protocol rules. On the other hand, there are plenty open-source and commercially existing implementations that can be used in order to implement a WebSocket application. The advantage of using an existing WebSocket server is that it is already tested, supported by a community of developers and offers additional features (Wang, et al., 2013).

2.5.3.4 Benefits of using WebSocket

- Performance: Using WebSocket, real-time communication becomes more efficient. Older AJAX based techniques such as polling and streaming can still be used over HTTP for a number of applications, however, WebSocket reduces bandwidth usage, saves CPU power and latency.
- Simplicity: Communication between client and server over the web becomes much simpler to implement. Implementing real-time applications with Comet techniques was complicated and the maintenance of session state across stateless requests increases complexity.
- Standards: Standard protocols can be built on top of WebSocket because it is an underlying network protocol. Thanks to these higher level protocols the development of re-usable components is possible.
- HTML5: WebSocket is a native HTML5 component enabling web browser to become fully capable application platforms. It provides TCP style networking for HTML5 applications without compromising security (Wang, et al., 2013).
Ratchet is a PHP component library that enables one to create real time, bi-directional applications between clients and servers over WebSockets. It allows the development of applications through simple interfaces using the decorator and command pattern. Specifically, Ratchet components can assemble a full stack server. A stack of Ratchet I/O Server Component, Ratchet WebSocket protocol handler and the business logic of the application can be built together in order to support WebSockets on the server side. Other components are the Session Provider, WAMP protocol handler, IpBlackList and FlashPolicy. Every component is built up on top of another similar to Lego bricks. (Ratchet, n.d.).

Compared to traditional PHP scripts using PHP in Ratchet differs. A connection remains open after the execution of only one script. Figure 9 illustrates the execution order of a client requesting a webpage. Once the page has been rendered it makes a new WebSocket connection with the Ratchet application and if there is no error a connection is established and remains open where either the server or the client can exchange data at any time (Ratchet, n.d.).

![Figure 9: Execution Order of a Ratchet script](image)

IoServer – This component is the base of a server application. It receives and handles new connections, reads/writes to those connections, closes the connections and handles all errors from the application. IoServer triggers the following events, onOpen, onMessage, onError and onClose. Also it invokes two methods, send and close on the connection object.
Ws-Server – This component allows the server to communicate with web browsers and applications that use the WebSocket API. It is compatible every WebSocket protocol version: RFC6455, HyBi, and Hixie76. Ws-Server triggers the following events: onOpen – when a new client connection has been opened, onClose: when a client connection is about to or has closed, onMessage – when a data message has been received and onError - when an error has occurred with a connection. Moreover, this component has two functions send and close. The first one is used to send a message to a client and the second one to close a connection with a client (Ratchet, n.d.).

2.5.3.6 WebSocket for Android

As it is mentioned above, all major web browsers for Android support WebSocket connection, but the Java JDK so far supports only HTTP. So, to develop a native Android application using WebSocket, one must implement the entire protocol in Java or use an existing WebSocket library, such as jWebSocket, wasync or Autobahn Android.
2.6 Critical Analysis of Literature Review

The large number of students at university classrooms results in less interactivity between lecturer and students. Student participation, engagement, motivation and learning are decreased and students become passive listeners. Moreover, it reduces the capability of a lecturer to get feedback, identify and support students with learning difficulties.

The active learning paradigm promotes active participation of all students in the learning process by asking, giving answers or doing an activity. Especially asking questions is of high importance for students, it is the base of research and self-motivated learning. Active learning has positive effects for both students and lecturer. However, it is easy to apply active learning in large classes. A solution to this problem is the combination of active learning and technology.

During the past years several electronic educational systems have been developed in order to create an interactive classroom. They are known as audience response systems and enable students to give instant feedback to the lecturer and the result is displayed in real time on a screen visible by all.

The literature shows that existing audience response systems add value in the learning process and enhance the student and lecturer experience, providing numerous benefits and enabling an active learning environment.

However, after reviewing all types of audience response systems it is clear that they are focused on answering multiple choice questions in the form of a poll. Except the sms-based, none of the existing systems allows a student to take the initiative to pose a question any time during a lecture. It is obvious that there is a gap in this area. The existing systems are not student-centric; they are designed to satisfy the needs of the lecturer for feedback, assessment and class activities.

The great availability and use of Internet-capable mobile devices among students can be used as a base to design interfaces that are cheap, user-friendly and can be accommodated in any classroom for any kind of student group. Mobile technologies
provide capabilities to support classroom interaction via web and cloud-based solutions eliminating infrastructure costs and limitation.

A critical element for the efficient operation of audience response systems is real time communication. WebSocket is a new technology that provides real time communication over the web between a client and a server. It supersedes Comet techniques based on Ajax which are difficult to implement and expensive in terms of resources. WebSocket enables a server to push information to clients on its own initiative, provides higher performance, makes the web stateful and it is simple to implement. It is a technology that can be used in order to create powerful web based audience response systems.
3. Professional Legal, Ethical Issues and Social Issues

3.1 Professional Issues

The work of other authors in any form is carefully cited and acknowledged. Moreover, the work of other developers is appropriately cited and credits are given to them. Professional conduct was demonstrated when interacting with project’s supervisor, university stuff and students taking part in evaluation. The use of university resources was compliant with the computer misuse law of the university.

3.2 Legal Issues

Before conducting the evaluation, the supervisor’s consent was taken. No sensitive data was collected from the interaction of users with the questioning system or the questionnaire.

3.3 Ethical/Social Issues

A question that arises is how far technology should go in replacing natural human behaviors. The intention of this application is not to replace the personal and physical interaction in a classroom, but enhance it by means of improving interactivity in large classroom where it is difficult to achieve it. The importance of human behavior in a classroom between students and lecturer and its advantages in the general social behavior are fully acknowledged. Students using this system must be informed that their questions are saved and associated with them.

Before the start of the evaluation process, the group of students that took part was informed that questions appearing on the screen are anonymous, but the application will keep track of questions and questioners and after the end of the evaluation the database will be deleted.

Furthermore, the questionnaire was anonymously completed by volunteers and data will be kept confidential, by the end of the project data will be deleted. (Mimtsoudis, 2013)
4. Requirements Analysis

4.1 Mandatory and optional functional requirements

Mandatory requirements

- **Client – server connection**: Create a client-server connection that will allow Android applications to connect to the server in order to organize questions and then display them.

- **Android application**
  - **Login**: Enable a user to login to the system once he has started the application by providing username and password.
  - **Registration**: Enable a user to register to the system.
  - **Type and send a question**: A should be able to type and send a question.
  - **Question feedback**: Allow a user to choose whether his question should be displayed or not after the filtering algorithm identified the user’s question as duplicate.

- **Simple algorithm for organizing displayed questions**: Before a question is displayed it must be processed by a simple algorithm in order to avoid duplicated questions being displayed.

- **Questioning screen**: The questioning screen will be a HTML5 web page for displaying questions sent by users. Every question will have a number indicating the number of students that have asked this question. This number will be increased every time a student asks this question. A question that has been asked by many students will be highlighted.

- **Lecturer management screen**: A lecturer will be able to browse between submitted questions. Every question will be associated with the student who has sent it.

Optional requirements

- Allow a lecturer to reward students with reputation points
- Enable students to see the list of reputation points
- Allow a lecturer to answer by email
- Integrate the questioning system with Societies platform
- Sophisticated algorithm for organizing displayed questions

4.2 User requirements

User requirements concern three types of users:

a) Students possessing an Android smartphone or tablet that can register, login and send a question. Students may get feedback from the lecturer in the classroom or later by email.

b) Students that do not possess a smartphone but can be benefited from the questions displayed on the screen.

c) Lecturer:
   - A lecturer will be able to see questions displayed on the screen and answer during the lecture.
   - Using the lecturer management screen a lecturer will be able to browse between submitted questions, answer by email and give reputation points.

User Requirements:

1. Easy to use Android interfaces that can fit in all types of screens, for both smartphones and tablets.
2. Allow a user to change the orientation of the application screen.
3. Provide anonymity for questions displayed on the questioning screen.
4. The questioning screen should be clear and well designed in order to allow every student in a classroom to read the displayed questions.
5. Lecturer must be able to trace students who send the questions.
5. Design

This chapter describes the design decisions that were taken for the implementation of the system. Specifically, the overall system design decisions are described and illustrated in a deployment diagram. Additionally, user interface design of the Android application and HTML5 webpages for the question screen and the lecturer management screen are illustrated and described. Last, the database design is described and depicted in a diagram.

5.1 System Design

This section provides an overview of the architecture and describes the components that constitute the system. Furthermore, the system’s functionality is described.

The system follows the client-server architecture and is an integrated questioning system that is composed from both a server side and a client side as illustrated in figure 10.

Server side

The Server side is hosted on a virtual server of Amazon Elastic Compute Cloud (Amazon EC2). It is composed of

- Apache Web server
- MySQL Database server
- WebSocket server - Ratchet

The web server hosts PHP web services that are used to connect the Android application and the lecturer management screen with the database. It also hosts the HTML5 webpages (lecturer management screen and question screen).

The WebSocket server is used to push asynchronously data to clients over a persistent connection in real time. It also hosts the question – filtering algorithm.

The database contains data concerning students, lecturers, submitted questions and data that is used by the question – filtering algorithm in order to display questions.
Client side

The client-side is composed of 3 major components which the user interacts with. These are:

- Student Android application
- Questioning screen
- Lecturer management screen.

The student application is installed on Android mobile devices. It contains all the required functionality that allows students to register to the system, login, type a question that will be anonymously displayed on the questioning screen and also to check the reputation points list of the class.

The question screen is a HTML5 web page that displays submitted questions in real time during a lecture, allowing both students and lecturer to see them. The questioning screen is designed to be projected in a classroom.

The lecturer management screen is a HTML5 web page that enables a lecturer after he/she logs in to initialize a new questioning screen, search for all previously submitted questions, answer each question by email and reward students with reputation points.

Communication

As it can be seen in figure 10, PHP web services can be accessed by different URL’s using the HTTP protocol.

The WebSocket technology is used in order to establish a persistent bi-directional real time communication between the clients and the server. The WebSocket server acts as a proxy between the Android application and the question screen. An alternative technology that could be used is Comet techniques. As described in the literature review, WebSocket outmatches Comet techniques in many ways; most important are its simple implementation, reduced resource consumption and natural bi-directional communication. Moreover, the continuous polling of a mobile device would drain the
battery very fast. Another reason for using WebSocket is to get involved and investigate this technology further.

Figure 10 depicts a deployment diagram of the system; it shows all software and hardware components of the distributed application, how they are connected and how they interact.
5.2 Database Design

The design and development of the database started just after the initial design of the system. The database is a fundamental part of the system’s design and functionality. The chosen database management system is MySQL.

The database which is illustrated on figure 11 consists of the following tables:

- students: A student entry is inserted into the table after a successful registration using the Android application. It includes a username which is each student’s unique email address (primary key), an encrypted password that is hashed with the md5 function, first name, last name and student reputation points.

- questions: A question entry is inserted every time a student sends a question. Each question entry gets a unique id (primary key), it includes the student’s question, the time submitted and the student’s username which is a foreign key of students table. There is a “1 to many” relationship between these two tables.

- lecturers: This table includes the lecturers email (primary key), name and password. It allows a lecturer of the university to log in into the lecturer question management page and initialize a question screen.

- keywordslist: A keyword (primary key) with a timestamp is added if after the processing of a submitted question there are new keywords. This table is emptied depending on the time the lecturer has set from the question management screen or a default value.

- filterwords: This table includes a set of words that are used in order to filter out words that do not give context in a question.

- displayed_questions: A new entry is inserted into this table every time a question is displayed on the questioning screen. Each question has a counter which indicates how many times this question was asked. The table is emptied at the same time as the keywordslist table.
Figure 11: Database diagram

5.3 Android Application Class Diagram

Figure 12: Android application class diagram
5.4 User Interface Design

This Chapter describes with the aid of figures the user interface design of the Android application, the questioning screen and the lecturer management screen.

5.4.1 Android application

After the application is launched users encounter five screens, but only four are visible. Three of these are Activities, one is a TabActivity and one is a ListActivity.

**BaseActivity** extends the TabActivity class, it is not visible but acts as the base Activity for MainActivity and RepPointsActivity which are represented as tabs. This is the launch Activity of the application; if users are already logged in, the first screen they see is MainActivity. Users can click on tabs to switch between the two Activities.

**MainActivity** extends Activity class; it is displayed after users successfully login. If the device is connected to the WebServer, a user is able to type a question and send it. If there is no Internet or WebSocket connection, a status messages is displayed. Moreover, by clicking the menu button of the device a user can log out or refresh the connection to the WebSocket server. The different states of the MainActivity are depicted in figure 13.

![Figure 13: MainActivity screen states](image-url)
**RepPointsActivity** (figure 14) extends ListActivity class. It displays the reputation points of all registered users in a ListView. The reputation points are fetched from the database through a web service.

![RepPointsActivity screen](image)

**Figure 14: RepPointsActivity screen**

**LoginActivity** (figure 15) extends Activity, if users are not logged in, the MainActivity is stopped and the LoginActivity is loaded. This screen enables users to login with their credentials or navigate to the RegistrationActivity. After a successful login a new user session is started. If the device is not connected to the Internet a status message is displayed indicating the absence of Internet and users are not able to click any button.

![LoginActivity screen](image)

**Figure 15: LoginActivity screen states**
**RegistrationActivity** (figure 16) extends Activity class and allows users to register into the system. After a successful registration user details are added to the database.

![RegistrationActivity screen](image)

**Figure 16: RegistrationActivity screen**

### 5.4.2 Question Screen

The question screen as illustrated in figure 17 is a HTML5 web page that allows the students and the lecturer of a classroom to see all submitted questions in real time. Every new question is displayed on top of the other and also contains a counter that indicates how many students have asked this or a similar question. When more than a number of students have asked the same question, this question is highlighted in green to indicate that it is important. The screen also contains a status that shows whether it is connected to the WebSocket server or not. If the connection is stopped for any reason, by clicking the connect button a new connection will be attempted. The question screen is intended to be projected in the classroom during a lecture session. Questions are dynamically generated using the JQuery library. Moreover, a CSS file is used to style and place elements on the page.
5.4.3 Lecturer Management Screen

The lecturer management screen is a web application that manages the functionality of the system from the lecturer’s side.

It allows a lecturer to search (by date, student or both) and browse between submitted questions (figure 18). A lecturer can answer by email a question that has not been answered in the class and reward students with reputation points for smart questions. Also it enables a lecturer to get feedback, identify students with learning difficulties and provide them personalized help.
Furthermore, one can start a new question screen and set the time (in minutes) in order to empty the list of keywords that are held in the database for identifying similar or same questions (figure 19). This time indicates the change of topic, so when the topic changes, the new questions should not be recognized as similar or same with questions posed in the previous topic. The time can be set before or after question screen has started. The default time set is 10 minutes.
JQuery UI facilitated the implementation of the UI. The accordion widget is used to display collapsible control panels for presenting the content of three option menus, Search, Question Screen, and Class Analytics. Furthermore, the Datepicker widget allows a user to select a date from a popup calendar. Additionally, a CSS file is used to style and place HTML elements on the page.
6. Implementation

This chapter describes the implementation of the system. Throughout the development phase small parts of each component were developed, tested and integrated to the system until the full functionality was achieved. This approach enabled the developer to always have a working client - server architecture which allowed easy code refactoring with the least effects to other components.

Because of the complexity of the system it is not possible to describe the exact flow of development. So, it felt reasonable to divide this chapter into two sections, Client - Side and Server – Side. Each section will describe the development of the most important parts of the involved components.

6.1 Client – Side

6.1.1 Android application

This section describes the development of the Android application. At the beginning, the Android manifest file was modified to allow Internet connectivity and network information by adding INTERNET and ACCESS_NETWORK_STATE permissions.

6.1.1.1 Registration and Login

The application allows students to register to AskFree questioning system and subsequently login to the application.

In order to register, the application must connect to the remote database. However, Android does not provide a means to directly connect to a remote server. Therefore, a PHP web service was developed to query the database and return the response. The response is returned in JSON format which is parsed into a JSON object in order to extract the data.

The inserted user details (email, password, first and last name) are passed to the web service using the HTTP POST method. The PHP script checks if the username already
exist in the database and if not a new user is inserted. The password is hashed with the md5 function and takes the place of the plain text before it is inserted in the database. A response from the server is returned for each case respectively and parsed with a custom JSON parser that was written in Java. Figure 20 illustrates the interaction between the Android application, the PHP script and MySQL database.

<table>
<thead>
<tr>
<th>Android Device</th>
<th>Server Side</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Android Device" /></td>
<td><img src="image" alt="Server Side" /></td>
</tr>
<tr>
<td>HTTP Request</td>
<td>PHP</td>
</tr>
<tr>
<td>JSON object</td>
<td>MySQL</td>
</tr>
</tbody>
</table>

Figure 20: Android, PHP, MySQL Interaction

Once a user has registered successfully, he is able to login into the application. The login process is similar to the registration; the application passes user credentials (username and password) with a HTTP POST request to a web service that queries the database in order to check whether the user is registered or not. The inserted password is hashed with md5 to match a registered has value. The JSON response is parsed and displayed.

After a successful login, the LoginActivity is stopped, a new user session is created with user’s email and password and the BaseActivity is started.

**Background Services**

Network operations should not be done in the main UI thread; otherwise a NetworkOnMainThreadException will be thrown. Any code that tries to connect with the outside world should be executed in a background thread.

One of the options that Android provides to handle this issue is to create an inner class that extends the AsyncTask class. AsyncTask enables an operation to be executed in
the background and provides event handlers to display progress and post results on the main GUI thread.

Hence, all interactions with web services and the database are performed using AsyncTasks. Specifically, they are responsible for:

- Registering users to the database.
- Logging in users to the application.
- Displaying responses on the UI screen.
- Displaying a progress dialog.

It should be mentioned that AsyncTasks are not persistent across Activity restarts, which means that if the orientation of the device changes, the AsyncTask will be canceled. However, they are a good and easy solution for short tasks such as posting and fetching data from the database and displaying the results on the UI.

### 6.1.1.2 Session Manager

Once a user has successfully logged in the application, a session is created with his username and password. The session is persistent across all activities and application closure until the user decides to terminate it by logging out.

The Session Manager class uses Shared Preferences to create user sessions. Shared Preferences is a name/value pair mechanism to save application data. The Session Manager class contains methods to

- Create a session
- Kill a session (log out)
- Retrieve session data (username, password)
- Check if a session exists (if user is logged in)

A session can be killed by choosing log out from the options menu. Every time the application is started it is checked if a session exist. When the session details are retrieved they are stored in a Hash table.
I would like to thank Ravi Tamada (2013) for his online tutorial in session management which was inspiring and allowed me to develop the session manager.

6.1.1.3 Create a WebSocket Client

MainActivity class is the most important part of the application, it enables a user to type and send a question. As mentioned in the previous chapter, a question can be sent only if the Android application is connected to the WebSocket server. The MainActivity enables the application to become a WebSocket client.

As discussed in the literature review, Android does not support yet the WebSocket protocol and API for native applications. Therefore, the Autobahn Android library was chosen to enhance the capabilities of the application and create a WebSocket client.

**Autobahn Android library**

The Autobahn Android (Tavendo, n.d.) is a WebSocket / WAMP client side library that enables one to develop native Java Android applications with WebSocket connectivity that talk to any WebSocket server or WAMP server. It is an open-source networking library and implements the WebSocket protocol and the WebSocket Application Messaging Protocol (WAMP). It supports WebSocket RFC6455, Draft Hybi-10+ and WAMP v1.

The library provides three callback methods to handle a WebSocket connection:

- `onOpen()` : It is fired when the connection has been established.
- `onTextMessage()` : It is fired when a text message has been received.
- `onClose()` : It is fired when the connection has been closed or could not been established at the first place.

It also provide a method to send text messages, `sendTextMessages()`.

**Establish a connection**

To begin with, when the activity is created, a new WebSocket connection object is instantiated.
After that the application checks if a session is set and if the device is connected to the Internet. If both conditions are true, the application tries to connect to the WebSocket server. Otherwise, a toast prompts the user to connect to the Internet, a status is displayed on the screen and the send button is disabled.

When the application establishes a connection to the WebSocket URI `ws://54.218.113.176:8080`, it creates a persistent bi-directional communication with the server and causes the onOpen method to be fired.

If the connection is lost, the user can refresh the connection by choosing the “refresh” option from the Android options menu that appears after a user clicks the options button that exist on every Android device.

**Send a question**

After a connection is established, users are able to type and send questions that are displayed in real time on the question screen that is also connected to the WebSocket server. The connection of all engaged parties is depicted in figure 21.

![WebSocket Connectivity Diagram](image)

**Figure 21: WebSocket Connectivity**

JSON format was chosen for sending messages between WebSocket clients and the WebSocket server. The reason for using JSON and not just string messages is to allow the WebSocket server to distinguish between different types of messages that come
from different types of clients. JSON format was chosen instead of XML documents because it has a smaller size and it is easier to be processed. The representation of information in JSON is more efficient than XML. Furthermore, it is easy for humans to write and read and for machines to parse and generate.

Thus, before a question is sent over the WebSocket protocol it is first transformed into a JSON object and then passed to the sendTextMessage() method that is invoked on the connection object.

Additionally, the question is entered in the database table questions. This enables a lecturer to use the lecturer management screen to browse between all submitted questions associated with a student. An AsyncTask was implemented for making a HTTP POST request and getting the response in a background service. Along with the question, the user email (username) is retrieved from the current session and is posted to the PHP script using theHttpPost method

6.1.1.4 Question Feedback

Before a question is displayed on the question screen, it is processed on the server by the question – filtering algorithm (will be explained in section 6.2.2) in order to identify if this or a similar question has been previously displayed or if it has no meaningful context e.g. “I would like to”.

If this is the case, the WebSocket server sends back a JSON object with an identification type. There are two types “0” and “1”. Type ”0” specifies that the question has no meaningful context and type “1” specifies that the same or a similar question has been previously displayed. The type “1” JSON object contains the question id and the question text.

When the message is received by the Android app, the onTextMessage() method id fires. Then, the JSON message is parsed and the data is extracted. If the type is “0”, a toast is shown indicating that the question the user has sent has no meaningful context. On the other hand, if the type is “1” the question text and the question id are extracted from the JSON object and an alert dialog is shown on the UI.
The alert dialog asks the student if the received question is what he/she means, as illustrated in figure 22. The student can answer either “Yes” or “No”.

- If the student answers “Yes”, his or her question will not be displayed and the counter that accompanies every question on the screen will be increased. This counter shows how many students have asked this particular question. The AsyncTask increaseQuestionCounter is executed. This task takes as parameter the question id and makes an HTTP POST request to a PHP script in order to increase the question’s counter in the displayed_question table in the database. At the same time a new JSON object that contains the question id is generated and sent to the WebSocket server in order to update the counter on the question screen.

- If the student clicks the “No” button, his or her question will be displayed on the question screen. A JSON object that contains the question is generated and the question is resent to the WebSocket server that immediately sends it to the question screen.

The purpose of the question feedback alert dialog is to give students an option to choose whether the question that the algorithm has found matches their question or not. It allows students to ask a question even if the matched question is correct. If students feel that they want to express the same question in a different way, they are free to do it. In this way students will not feel censored by the algorithm or the lecturer; they are able to express themselves in any way they want. Therefore, it was
decided not to censor rude words or sentences. Every question is saved in the database and associated with the student who asked it, so this will work as a natural way to avoid rude or frivolous submissions. The advantage of this approach is that students will feel free to send whatever they want, even rude words, but they will not do it because they know that submissions are associated with them.

6.1.1.5 Reputation Points

Students can be rewarded with reputation points for their questions by the lecturer. Reputation points aim to increase student participation and motivation to ask smart and well thought question. It could also create competence between students. The reputation points for every registered member are displayed in the Reputation Points tab.

The RepPointsActivity activity consists of a ListView that is populated by results returned by database queries that are performed by an AsyncTask through a PHP web service. The results are returned in JSON format and are parsed into a JSON object from which a list of objects is created. Every object in the list contains the name and reputation points of every registered student.

The reputation points list can be refreshed by clicking the “refresh” button in the option menu.
6.1.2 Web application

6.1.2.1 Question Screen

The question screen is a HTML5 web page that displays questions asked by students using the AskFree Android application that was described in chapter 5.1.1. The WebSocket server pushes student questions to the question screen that is a WebSocket client.

Create a WebSocket client

When the web page is loaded, a WebSocket connection object is instantiated and a connection to the WebSocket URI ws://54.218.113.176:8080 is performed. After a connection is established, the onopen() event handler is fired and a message is sent to the WebSocket server containing the client id “*sc*” specifying that a screen has connected. Also a green connection status is shown on the screen. After that, the screen is able to receive messages.

Handle incoming messages

When a message is received, the onmessage() event handler is fired. At this point it should be mentioned that the communication between the server and the HTML5 webpage is done by exchanging JSON objects. The received message is parsed and the type of the message is checked. Depending on the type of the message, either a new question is displayed on the screen or the student counter of an already displayed question is increased.

Create a dynamically generated screen

When a new question screen is loaded, it contains only a question wall, as illustrated in figure 23. JQuery is used to dynamically generate new questions on the question wall.
Incoming JSON objects that are new questions contain the question id, the question text and the number “1”, which is the number of students that have asked this question. Using JQuery, every new question is placed inside a HTML fieldset and the question id is set as the fieldset id. Additionally, a HTML div that contains the number of students along with the sentence “…students ask this.” is appended to the fieldset. Every new question is placed on top of the previous. The maximum number of questions hold on the question wall is 8; when this number is reached, for every new question the last one is removed.

Incoming JSON objects that are positive responses on the question feedback, meaning that a student answered “Yes”, contain the question id which will be updated. Using JQuery, this question id is used to find the fieldset that contains the corresponding question and increase the number of students that have asked this question.

When a question has been asked by 3 or more students, the question is highlighted in green in order to show that it is an important question. The number 3 was randomly chosen for the purpose of development and testing.

Last, when the connection with the WebSocket server is closed the onclose() method is triggered and the status on the screen changes. A user can attempt a new connection by clicking the connect button.
6.1.2.2 Lecturer Management Screen

The lecturer management screen is a HTML5 webpage that allows a lecturer to search for questions, answer a question by email, reward a student with reputation points, and initialize a new question screen.

Establish a WebSocket connection

When the page is initially loaded it establishes a connection with the WebSocket server on the URI \( ws://54.218.113.176:8080 \). After that, the onopen handler is fired; a JSON object is generated and sent to the server in order to specify that a lecturer has connected. Furthermore, the default time value to empty the list of keywords in the database is set. The connection to the WebSocket server is not essential for the operation of this page; it is useful only during a lecture session in order to set a time value.

Search questions

A lecturer is able to search in the database for submitted questions using a variety of search criteria, such as student name/surname, date, date and student name/surname, and last get all questions. The JQuery library is used to get student’s search criteria and then perform AJAX requests with the $.ajax() function in order to get results and display them on the existing web page, without reloading the whole page. The requests are asynchronous, allowing a user to continue the interaction with the web page while the server processes the request concurrently. Specifically, each AJAX request is performed after the user clicks the search button. User’s search criteria are posted to a PHP web service that fetches the data from the database and returns results in JSON format. Results are extracted from the incoming JSON object and a table is dynamically generated containing the results. A partial page update takes place. Additionally, each row contains two buttons, “Email” and “+1RP”. The first one allows a lecturer to answer on a specific question and the second one to reward a student with a reputation point. Every row contains the student’s id, so when a user clicks the +1RP button this id is retrieved in order to perform a HTTP post request to a PHP web service in order to update the student’s reputation points in the database.
Initialize a question screen

This page allows a lecturer to start a question screen and also set the time to empty the keywords list in the database. When a user sets the time in minutes a JSON object is generated and sent over the WebSocket connection to the server. The JSON object contains an indicator that the message comes from a lecturer management screen and the minutes. The time can be changed any time during a session. Furthermore, there is no limit for the number of question screens that can be started and operate concurrently.

Class Analytics

A lecturer is able to get a list of registered students and see the student’s reputation points list. The action is performed as AJAX request using the JQuery $.getJSON() function. This function makes a GET request to a PHP web service that returns a JSON object. After that, similar with the search results, a HTML table that contains the results is dynamically generated using JQuery function $.append().

6.2 Server – Side

Server - Side Problems

At the beginning of this project it was intended to implement the server-side using the current infrastructure of Heriot-Watt University. However, after installing two different WebSocket servers, naming node-js/socket.io and ratchet it was not possible to access any of them remotely. After consulting the computer manager of the School of Mathematical and Computer Sciences, it was comprehended that the university’s firewall was blocking remote connections on the required ports. An application to the central IT office of the university was required in order to open these ports. To avoid this time consuming process an initiative was taken to set up a virtual cloud server on the Amazon Web Services. The Amazon Elastic Compute Cloud (EC2) provided a free (for 1 year) and quick way to set up and manage a server. Using the EC2 management console a new virtual machine was instantiated with Ubuntu OS pre-installed. After that, the required ports and security settings were configured. SSH
was used to log in and install Apache Server, MySQL Server, Ratchet PHP WebSocket Server and phpMyAdmin.

This initiative was essential for the developer to experiment and gain experience with cloud computing and Amazon Web Services, server technologies as well as operating systems, which was out of the scope of this research project. Furthermore, a better understanding of WebSocket technology was achieved with the installation and use of two different WebSocket servers, JavaScript based and PHP based.

### 6.2.1 WebSocket Server

The WebSocket server enables Android and web application clients to establish a real time bi-directional persistent connection with the server. The server pushes messages from the Android application to the question screen. It also pushes messages to the Android application after they are processed with the question filtering algorithm.

Ratchet PHP WebSocket server was chosen because it is PHP based which allows easy interaction with MySQL and it is compatible with AutobahnAndroid, making the interaction between all parts transparent.

**Server Initialization**

The WebSocket server was installed on the virtual machine and after it is initialized, it listens on incoming connections on port 8080. After the initialization, the keywordslist and displayed_questions tables in the database are emptied and the default value of 10 minutes is set for the next empty. Also a map (\SplObjectStorage) is created to store connected clients as object.

**Accept and Close a connection**

When a new client connects to the WebSocket server the onOpen() method is fired and a new connection object is stored to the map. After that, messages can be sent to this client. When a question screen is connected it is identified by the server thanks to an indicator that it carries, so before this client is saved, an id value is set to the
connection object. This allows the server to direct messages to connected question screens.

When a connection is closed the onClose() method is fired and the corresponding connection object is deleted from the map.

**Receive a message**

When a new message is received the onMessage() method is fired having two parameters, message and sender.

A received message is checked if it is a JSON object or a text message. After a JSON object is identified, it is decoded and analyzed. There are four types of objects that can be received that correspond to different cases:

- When a user sends a new question it is processed by the question – filtering algorithm and depending on the process an action is performed.
- After a user has pressed “No” on the question feedback alert dialog of the Android application, the question is re-send and received by the server. After that, the question is inserted in the database (displayed_questions table) and the question id is acquired. Next, a JSON object that contains the question, the question id and the number “1” is generated and sent to every connected question screen in order to display the question.
- After a user has pressed “Yes” on the question feedback alert dialog of the Android application, the server receives the question id of the question that is going to be updated on the screen. The question id is wrapped again into a JSON object and sent to every connected question screen.
- A lecturer management screen is connects to the server or a user sets a new time value for emptying the keywordslist and displayed_questions tables in the database.

**6.2.2 Question - filtering algorithm**

Every question that a student sends is received by the server and processed in order to organize questions displayed on the screen. The primary goal is to avoid duplicated
question that have been sent in a particular time. The term duplicated questions is defined as questions that have the same or similar context, which is specified by keywords. A word is treated as keyword if it is not a filter word.

The database table filterwords contains common English words such as “I, you, a, could, have, and, is, etc” that do not give any specific meaning to a question and cannot be treated as keywords. These words are specified as filter words and are indicative, it is not implied that the list of words is complete.

The operation of the question – filtering algorithm is briefly explained below:

When a new question is received, the words that have no meaning (filter words) are filtered out. After that, if no words exist, it is assumed that the question has no meaningful context, the question is not sent to the question screen and a message is returned to the user.

If words exist, it is assumed that the question has a meaningful context and the words are specified as keywords.

- After that, if the keywordslist table in the database is empty, each keyword is inserted in the table and the whole question is inserted in the displayed_questions table. Then, the question together with the question id is sent to every connected question screen.
- If the keywordslist table is not empty, the new keywords that have been extracted from the question are compared against keywords in the keywordslist (will be called old keywords) in order to find similar or same keywords.
  - First, a word comparison is performed to find words that are equal. These words are considered perfect matches and are saved in the $perfectMatches array.
  - Then, in order to find similar words, the length of new and old keywords is compared. For every new keyword that is smaller than an old keyword, the last character is trimmed and a check is performed to find out if the trimmed keyword is inside any of the old keywords. The purpose of this process is to identify most of the plural and singular words as similar, as well as words such as “race” and “racial”.
- After that, a similarity check is performed with the similar_text() PHP function between the trimmed word and the keyword. If the similarity percentage is higher than 60%, the two words are considered similar. The similarity percentage is used to avoid word occurrences inside other words that are not relevant with each other. For instance, “design” exists in “designated”; however these words are not related with each other and should not be considered as similar. The percentage is arbitrarily chosen for the purpose of development and testing. There has not been any research that indicates that this assumption is correct.

- The same process occurs also for every old keyword that is smaller than a new one. Keywords that are considered as similar are saved in the $similarWords array.

- Keywords that are not equal or similar are considered new keywords and are saved in the $newKeywords array.

- So, keywords in these three arrays form the question.

- After that, if the question is specified as new it is sent to the question screen to be displayed. Else, it is specified as same/equal with a question that has already been displayed, meaning that these questions have similar or equal context. Next, the question with most similar or same context is found and returned to the user as question feedback. After that, according to user’s action cases 2 or 3 from section “Receive a message” applies.

- The process of recognizing the type of question and finding the question with most similar or equal context is briefly discussed below.

  - The three arrays ($perfectMatches, $similarWords, $newKeywords) that have been previously explained are used to recognize the type of question.

  - Questions that have been already displayed in a specified time period are fetched from the database and analyzed. Keywords from $perfectMatches, $similarWords arrays and a combination of them are compared against all words from displayed questions. If matches are found in a question, the question’s id is saved with a counter that is
increased with every new match. Last, the question id with most matches is returned in order to fetch the question text.

- The question text and the question id are sent to the user as the question feedback.
- Keywords from the $newKeywords array that exist in a question are added to the keywordslist in the database.
- If only $newKeywords array has values it means that this question is new and it is sent to the question screen to be displayed.
7. Evaluation

The evaluation process primarily aims to discover if the developed system would be beneficial for learning in a university classroom. Furthermore, it aims to observe reactions of participants during the evaluation.

7.1 Evaluation Method

The evaluation was conducted in a lecture room of Heriot-Watt University at the Department of Computer Science in order to simulate a lecture session and create a more realistic feeling and experience. Participants were MSc students of a variety of disciplines of Heriot-Watt University. 16 participants were recruited, 8 males and 8 females in the age group of 22-27. One of the requirements of the evaluation was the possession of an Android smart phone. Participants who did not possess one were given one for the sake of the evaluation from the PUMA lab staff.

Before evaluation started the Android application – AskFree was installed on all devices. Furthermore, evaluators were informed that their questions will be displayed on a screen visible by everyone in the room and also saved in a database for the purpose of this research. After that, evaluators were asked to start the application, register to the system and then login. As part of the evaluation a presentation was given to participants in order to explain the functionalities and the purpose of the system. Participants were encouraged to use the application during the presentation in order to ask questions and observe them on the question screen.

Unfortunately, a connectivity problem between the projector and the laptop did not allow the projection of the question screen on the white board. To overcome this problem, the laptop was placed on a desk in the middle of the class, so all students gathered around it and were able to see the questions that were being sent.

After the presentation, evaluators were asked to:

- Type and send more questions
- Give different feedback (Yes/No) on the feedback question
- Check the reputation points list
- Observe questions on the screen

During the evaluation period reputation points were given to all participants and their reactions were observed.

After that, evaluators had the opportunity to assess the lecturer management screen. They were allowed to search for questions, give reputation points to other participants, view the reputation points list and start a new screen. The purpose was to give evaluators the opportunity to see how the system works from the lecturer’s side, to put themselves in the position of a lecturer and how it would affect the interaction with students.

Then, evaluators were asked to fill in a questionnaire in order to gather data about their experience with the questioning system and if they feel that it could add value to the learning process. The questionnaire included Likert scale questions, true or false (yes/no) questions and open ended questions.

### 7.2 Findings

Below are the findings from the questionnaire (refer to Appendix A) which was filled in just after the evaluation:

**Question 1:** Have you ever used a class/audience response system (e.g. Clicker) before? If yes in which context have you used it?

**Rationale for the question:** In this question it is intended to find out if evaluators have had a previous experience with these systems and how they have used them. It is important to know if participants had a previous experience in a real environment because this can lead to a different level of appreciation of the evaluated system.

**Results:**
The results show that only 2 out of 16 people had used an audience or class response system before. Both have used it at a university. The overwhelming majority of evaluators have no previous experience.

**Question 2:** On a scale 1-5, do you think that AskFree would motivate students to ask questions in a classroom? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** Student participation in classrooms is very low, students are primarily passive listeners. The application aims to increase student participation through asking. This question tries to find out if using this application, students would be willing to ask questions and hence participate more actively in the classroom.

**Results:**

Figure 25: Shows if AskFree would motivate students to ask questions
Figure 26: Shows if AskFree would motivate students to ask questions by gender

The results show that the majority of respondents think that students would be motivated to ask questions. Specifically, 5 answered agree and 6 strongly agree. Furthermore, 4 are undecided and 1 responded disagrees. From the second graph of this question (figure 26) it can be seen that female evaluators have a slightly more positive attitude than male, as 6 female participants agree or strongly agree compared to 5 male participants.

**Question 3:** On a scale 1-5, do you think that AskFree would support students that are not confident to raise their hand and ask a question? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** Shy and unconfident students are likely not to ask questions during a lecture and participate less. AskFree aims to allow these students to actively participate in the classroom.

**Results:**
Figure 27: Shows the frequency of students who think that AskFree would support students that are not confident to raise their hand and ask a question. The overwhelming majority of respondents agree or strongly agree that AskFree would support students that are not confident to raise their hand and ask a question. Specifically, 15 students agree or strongly agree compared to only 1 who is undecided. From the second chart (figure 28) showing the different responses between males and females it can be derived that the majority of females strongly agree compared to the majority of males who agree. Additionally, the undecided student is a male.

Figure 28: Shows the frequency of students who think that AskFree would support students that are not confident to raise their hand and ask a question by gender.

Question 4: On a scale 1-5, do you think that it would help these students to integrate in the class and become more sociable? (1 means strongly disagree and 5 means strongly agree).
**Rationale for the question:** As described in chapter 2.1 anxiety which is a feature of shy and unconfident students may lead to low academic performance and social exclusion. This question is included in order to identify whether the application could help these students.

**Results:**

![Figure 29](image1.png)

*Figure 29: Shows if students think that AskFree would help shy and unconfident students integrate in the class and become more sociable.*

![Figure 30](image2.png)

*Figure 30: Shows if students think that AskFree would help shy and unconfident students integrate in the class and become more sociable by gender.*

Responses in this question are surprising, whereas in the previous question the majority of students responded that AskFree could support shy and unconfident students, it seems that they are more skeptical concerning if it can help them to integrate in the class and become more sociable. In the above charts (figures 29, 30) it is depicted that 6 respondents were undecided, 5 agree, 2 strongly agree and 3 disagree. Observing the gender differences it is clear that males are more skeptical as
4 are undecided, 2 disagree and 2 agree compared to females where 3 agree, 2 strongly agree, 2 are undecided and 1 disagrees.

**Question 5:** On a scale 1-5, do you think that AskFree would improve the student classroom experience? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** The success of the application is based on the improvement of current classroom experience which is considered poor according to the literature. This question is included in order to find out if students see any value in the system.

**Results:**

![Figure 31](image1.png)

Figure 31: Shows the frequency of students who believe that AskFree would improve the student classroom experience.

![Figure 32](image2.png)

Figure 32: Shows the frequency of students who believe that AskFree would improve the student classroom experience by gender.
Results in this question show that the majority of students (10), agree or strongly agree that the questioning system would improve the student classroom experience, whereas 1 student disagrees and 5 express no opinion. Observing gender differences in figure 32 it can be seen that 5 males and 5 females agree or strongly agree. On the other hand, more females are undecided, whereas the one who disagrees is a male.

**Question 6:** On a scale 1-5, do you think that AskFree would improve student – teacher interaction? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** Teaching in large classroom is the current state at universities. However, this style of teaching decreases the student – teacher interaction. With this question it is intended to discover if students believe that AskFree could improve this situation.

**Results:**

![Figure 33](image1.png) Shows the number of students who believe that AskFree would improve the student-teacher interaction

![Figure 34](image2.png) Shows the number of students who believe that AskFree would improve the student-teacher interaction by gender
The above diagram (figures 33) illustrates that the majority of respondents (10) agree or strongly agree that AskFree would improve the interaction between students and teacher. However, it is not surprising that 6 respondents are undecided; student-teacher interaction is related with classroom experience. Looking back to the previous question it is observed that 5 respondents were undecided too. Furthermore, females express a slightly more positive attitude towards AskFree as depicted in figure 34.

**Question 7:** On a scale 1-5, do you think that AskFree would improve the capability of teachers to identify and support students with learning difficulties? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** Because of low student engagement and teaching in large classrooms it is difficult for a lecturer to get feedback from students and identify those with learning difficulties.

**Results:**

![Bar chart](image)

*Figure 35: Shows the number of students who believe that AskFree would improve the capability of teachers to identify and support students with learning difficulties.*
Results of this question are very positive as 14 respondents agree or strongly agree that the questioning system would improve the capability of teachers to identify and support students with learning difficulties. On the other hand, 1 respondent agrees and 1 is neutral. The majority of female respondents strongly agree while the majority of male agree.

**Question 8:** On a scale 1-5, do you think that reputation points would motivate students to use AskFree? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** A lecturer can reward students with reputation points for a smart question or for any other reason. Reputation points can create competition between students and increase participation and engagement. The purpose of this question is to discover if reputation points can motivate students to use the application.

**Results:**
Figure 37: Shows if reputation points can motivate students to use AskFree

![Bar chart showing the distribution of responses to the question: On a scale 1-5, do you think that reputation points would motivate students to use AskFree? (1 means strongly disagree and 5 means strongly agree).]

Figure 38: Shows if reputation points can motivate students to use AskFree by gender

![Bar chart showing the distribution of responses to the question: On a scale 1-5, do you think that reputation points would motivate students to use AskFree? (1 means strongly disagree and 5 means strongly agree) by gender.]

The results above (figure 37) show that the majority of students think that reputation points would motivate students to use AskFree. Specifically, 8 students agree and 4 strongly agree. On the other hand, there are 3 respondents who disagree and one who is undecided. Looking at the second chart (figure 38) it is observed that the 3 respondents who disagree are males while females once again are more positive.

**Question 9:** On a scale 1-5, do you think that it is easy to use the application? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** This question is included in order to identify the usability of the application.

**Results:**
The majority of students (13) think that it is easy to use the application. Specifically, 8 students responded strongly agree and 5 agree. On the other hand 2 responded neutral and 1 student thinks that it is difficult to use the application as he or she answered disagree. Observing the different responses between males and females (figure 40) it can be seen that males seem to find the application usage easier than females.

**Question 10:** On a scale 1-5, do you think that question anonymity encourages student participation? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question:** One of the reasons that students do not participate actively in a classroom is fear of judgment or humiliation from their peers.
Anonymity provides a shield to the students in order to ask questions without being afraid. Anonymity is a key feature of AskFree, so it is important to find out whether students appreciate it or not.

**Results:**

![Figure 41](image1.png)

Figure 41: Shows if question anonymity encourages student participation

![Figure 42](image2.png)

Figure 42: Shows if question anonymity encourages student participation by gender

The results of this question indicate that the majority of respondents appreciate the anonymity feature. Specifically, 10 students strongly agree in addition to 3 who agree. On the other hand 2 respondents are undecided whereas 1 disagrees. No specific gender differences can be derived from this question as the number of males and females who strongly agree is equal, as well as the number of undecided respondents. However, the number of females who agree is higher by 1, as the respondent who disagrees is a male.
**Question 11:** If AskFree was available in classrooms during your master’s course, would you have used it?

If yes, please explain your motivation for doing this. If no, please explain why you wouldn’t.

**Rationale for the question:** This question aims to find out if students would have used AskFree if it was available in classrooms. It is important to know why they would have used it and why they would not. Useful results can be obtained from this question regarding system’s usefulness and student’s point of view.

**Results:**

![Bar chart showing the number of students who would have used AskFree and those who would not](chart1.png)

Figure 43: Shows the number of students who would have used AskFree and those who would not.

![Bar chart showing the number of students who would have used AskFree and those who would not by gender](chart2.png)

Figure 44: Shows the number of students who would have used AskFree and those who would not by gender.

The chart of figure 43 illustrates that the majority of students (10) would have used AskFree if it was available during their Master’s course. On the other hand, 5 respondents answered negative and 1 response (male) is missing. It is interesting to
observe gender differences (figure 44) in this question, as the majority of females (7) answered positive while the majority of males (4) negative.

The number of male respondents who answered negative is surprisingly high because in general previous male responses were positive. In order to understand the reason for answering either positive or negative, one should read the reasons that were given by respondents. Reasons are grouped by Yes or No in the following two tables: (all responses can be seen in Appendix A)

<table>
<thead>
<tr>
<th>Yes</th>
<th>Reason</th>
<th>Number of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shy or unconfident person</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Anonymity</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Cannot express myself in a foreign language</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Not interrupting the lecturer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Not disturbing the class</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Ask something for a previous point in the lecture</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Positive responses

From people who answered that they would have used the application, the majority (6) consider themselves as shy or unconfident person, which is the main reason that they would have used it. Furthermore, 3 of them stated that a reason for using it is anonymity, because they would not be afraid of other people’s judgment. Additionally, 2 students said that they would use the application because they are not confident and cannot express themselves easily in a foreign language. Another student would have used it to ask something for a previous point in the lecture without interrupting the lecturer. Last, another respondent would have used it for not disturbing the class.
<table>
<thead>
<tr>
<th>Reason</th>
<th>Number of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefer personal interaction</td>
<td>3</td>
</tr>
<tr>
<td>Application decreases personal interaction</td>
<td>1</td>
</tr>
<tr>
<td>Difficult to use the application while</td>
<td>1</td>
</tr>
<tr>
<td>following the lecture</td>
<td></td>
</tr>
<tr>
<td>Not shy</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Negative responses

Most of the people (3) who would not have used AskFree prefer a personal interaction with the teacher. One of them stated that the application would decrease personal interaction. Another respondent thinks that it would be difficult to type and send a question and at the same time follow the lecture. Also there is a consideration that by the time the teacher sees the question he or she would be already in the next topic. Last, a student said that the reason he would not have used the application is that he is not shy.

**Question 12**: On a scale 1-5, how helpful did you find the similar-same question feedback (“do you mean: …”)? Did this question always seem to be appropriate for the questions you asked? (1 means strongly disagree and 5 means strongly agree).

**Rationale for the question**: This question is included in order to identify the usability of the same-similar question feedback and if the feedback is helpful.

**Results:**
Results of this question indicate that the majority (10) of students agree or strongly agree that the question feedback is useful and also that it always seem to be appropriate for the questions they asked. However, 6 respondents are undecided.

There is a concern about the results of this question as there is a feeling that the majority of respondents answered based on the first part of the question. Hence, the results of this question will not be considered for further analysis.

**Question 13:** Any other comment or recommendation you would like to make about the system?

**Rationale for the question:** This question is done in order to get a general feedback about the system and give the opportunity to students to comment.

**Results:** Unfortunately only two respondents answered this question (responses can be seen in Appendix C). The first said that the system would be probably successful
among shy and unconfident students so they would rely on it. The second respondent commented that cross platform implementation would be necessary as the system assumes that students have certain device (Android) and software.

7.3 Discussion of results

This chapter provides a discussion of the questions and responses undertaken by students of Heriot-Watt University. The methods of evaluation and data gathering have been explained in chapter 6.1 along with the results in chapter 6.2. Data has been analyzed in detail in order to form conclusions which will contribute in deeper understanding of the usefulness of the developed questioning system for learning in a university classroom.

To begin with, the majority of evaluators were inexperienced with the use of class or audience response systems. Therefore, a certain amount of time was required in order to become familiar with the concept of using an electronic means inside the classroom to interact with the teacher.

The results clearly show that the majority of respondents have a positive attitude towards the system. They believe that it would motivate students to ask more questions and thus increase active participation in the classroom.

Shy and unconfident students are more likely to be benefited as it allows them to overcome anxiety and fear of judgment by sending questions anonymously. This opinion is shared by the overwhelming majority of students. However, when evaluators were asked if AskFree could help these students become more sociable and integrate in the classroom responses were mixed. This could be explained by taking into consideration that there is a difference in the way people perceive social. Nowadays, with the tremendous use of social networks and technology there are people who meet, talk and create their social environment within electronic media. Hence, for some people asking and taking part in classroom activities by using an application could be perceived as social. On the other hand, there are people who perceive as social mainly personal interactions.
Similar results were obtained about student classroom experience and student-teacher interaction. The majority of evaluators responded that the use of AskFree would improve both; however there is also a lot of skepticism, as a significant number of respondents is undecided. Results of those two questions indicate a relation between student classroom experience and student-teacher interaction. The large number of undecided students could be partially explained when considering that the main reason that some students would not have used the application is a preference in personal interaction. It seems that personal interaction plays a significant role for students. It is a positive sign that they distinguish personal and electronic interaction; this means that they seek to connect with people in person which is not easy in large classrooms. A balance between personal and electronic interaction would be suggested. Furthermore, it is crucial to highlight that an application likes this should not replace personal interaction but enhance it when it is difficult to achieve it.

The capability of a teacher to identify and support students with learning difficulties is an aspect that is highly rated by respondents. It is obvious that a teacher can get more feedback through the questioning system and is able to answer questions during the class or later.

The majority of evaluators believe that reputation points can motivate students to use the application. A reason for this trend could be competition, as students like to compete with each other. Also as pointed out in the literature review in chapter 2.2.4, student’s attendance in a classroom is increased when there is a reward. Furthermore, during the evaluation process reputation points were given to several students. It was observed that this had a positive impact as students were competing with each other, discussing about their points and asking for more. On the other hand, there were students who responded negatively in this question. This could be possibly explained by the fact that the physical presence of students in a classroom is required in order to ask questions and get reputation points; however, some students may prefer not to attend classes.

An encouraging fact about the usefulness of the system is that the majority of students stated that they would have used the application if it was available during their Master’s course. It is interesting that 60% of those students consider themselves as shy or unconfident and stated that anonymity is a reason for using the application, as
they would not be afraid to ask questions. Furthermore, foreign students seem to appreciate the application as they feel that it could help them to express themselves and be more confident without feeling embarrassed for their grammatical mistakes or their accent. The findings suggest that the application could be more valuable to some groups of students, mainly shy, unconfident and foreign. However, this does not mean that it would not be useful to other groups of students.

Results indicate that generally males are more skeptical than females. Furthermore, the majority of males reported that they would not have used the application and the reason is that they prefer personal interaction. This could be explained by the fact that males by their nature are more confident than females. The fact that females expressed more positive feelings about the application and are willing to use it could be explained by considering that female usage of social media is higher than male. Especially the use of twitter which provides similar functionalities as AskFree is dominated by women (Allen, 2013).
8. Further Work

This MSc project introduced an application to facilitate classroom interaction. Furthermore, an evaluation has been achieved in order to discover if this system would be beneficial for learning in a classroom environment. This project successfully met the research aim and fulfilled the objectives. However, AskFree deserves further research and expansion as mobile and web based class systems constitute a new research field.

The Android application could be expanded in order to allow students to receive and answer questions anonymously posted by the lecturer. This would also require an expansion in the lecturer management screen in order to post questions. This extra feature would enable a two way electronic dialog in a classroom. Furthermore, the lecturer management screen could be enhanced with an email interface to facilitate answering questions as the current version delegates the task to user’s mail system.

The question organizer could be replaced or enhanced by a more sophisticated algorithm using Natural Language Processing. The semantic of questions could be identified and grouped accordingly in order to provide optimal feedback to students and display question in a more efficient way.

As the project was conducted over the summer months it was not possible to evaluate the questioning system in a real classroom environment. Further evaluation could be conducted over an extended period such an academic semester in order to discover whether the application increases student classroom experience, student-teacher interaction and how it affects student’s performance and participation.
9. Conclusion

This dissertation achieved to develop an application that facilitates classroom interaction. Particularly, an Android application has been developed that enables students to type and send a question during a lecture session. The questions are displayed in real time on a screen which is intended to be projected in a classroom. An algorithm was designed and implemented in order to avoid duplicated questions being displayed on the screen when they are submitted in a specified time period. Furthermore, a web application was developed in order to enable a lecturer to browse between questions and students, answer a question by email and reward students with reputation points. This web application intends to provide a lecturer with feedback and enhance his capability to support students.

After the implementation, an evaluation was conducted to investigate if the developed questioning system would be beneficial for learning in a university classroom. It has been found that students are positive in the use of the application in a classroom, as they believe that it would motivate students to ask more questions and thus increase participation. The results showed that shy, unconfident and foreign students would be benefitted most, as question anonymity allows them to overcome anxiety, fear of judgment and embarrassment because of accent or poor use of language. However, the results are mixed whether AskFree could help shy and unconfident students to become more sociable because of different perception people have about social. It was found that there is a relation between student classroom experience and student-teacher interaction. While the majority of evaluators believe that the application would improve both, there is also significant consideration. The findings indicate that personal interaction with a lecturer plays an important role for students. A balance between personal and electronic interaction would be suggested. Furthermore, an application should not replace personal interaction but enhance it when it is difficult to achieve it. The results show that reputation points would motivate students to use the application and increase classroom participation. On the other hand, in order to acquire reputation points the physical presence of students is required in the classroom; however some students may prefer not to attend classes. The most encouraging fact about the usefulness of the system is that the majority of evaluators would have used it if it was available during their master’s course. Last, the results
show that males are more skeptical than females about using the application and its usefulness. The main reason is that they prefer personal interaction.


Appendices

Appendix A : Questionnaire

AskFree is a class questioning (response) system that aims to facilitate student – lecturer interaction and improve student participation in a classroom setting. This questionnaire aims to discover if this system would be beneficial for learning in a university classroom.

Gender: Female □ Male □

Have you ever used a class/audience response system (e.g. Clicker) before?
Yes □ No □

If yes, in which context have you used it?

On a scale 1-5, do you think that AskFree would motivate students to ask questions in a classroom? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that AskFree would support students that are not confident to raise their hand and ask a question? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that it would help these students to integrate in the class and become more sociable? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

[93]
On a scale 1-5, do you think that AskFree would improve the student classroom experience? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that AskFree would improve the student – teacher interaction? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that AskFree would improve the capability of teachers to identify and support students with learning difficulties? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that reputation points would motivate students to use AskFree? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that it is easy to use the application? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

On a scale 1-5, do you think that question anonymity encourages student participation? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

If AskFree was available in classrooms during your master’s course, would you have used it?

Yes No

If yes, please explain your motivation for doing this. If no, please explain why you wouldn’t.
On a scale 1-5, how helpful did you find the similar-same question feedback ("do you mean: …")? Did this question always seem to be appropriate for the questions you asked? (1 means strongly disagree and 5 means strongly agree).

1 2 3 4 5

Any other comment or recommendation you would like to make about the system?
### Appendix B: Student responses in question 11

<table>
<thead>
<tr>
<th>Yes</th>
<th>1. <strong>Yes.</strong> In order not to disturb the class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. <strong>Yes.</strong> It enables shy people to ask questions easily, therefore I would have used it</td>
</tr>
<tr>
<td></td>
<td>3. <strong>Yes.</strong> 1) for fun, 2) to ask something for a previous point in the lecture without interrupting the lecturer</td>
</tr>
<tr>
<td></td>
<td>4. <strong>Yes.</strong> I am not a confident person, so an application like this could help me ask questions and keep me anonymous. As a result my questions would have an answer.</td>
</tr>
<tr>
<td></td>
<td>5. <strong>Yes.</strong> I am a foreign student so it is difficult for me to express myself. I think that an app like that would help people like me to be more confident</td>
</tr>
<tr>
<td></td>
<td>6. <strong>Yes.</strong> As a person who usually doesn’t ask questions, that would be very useful as nobody would see that asked it and I would not be afraid about other people judgment. Overall great idea.</td>
</tr>
<tr>
<td></td>
<td>7. <strong>Yes.</strong> Not confident asking questions in a foreign language</td>
</tr>
<tr>
<td></td>
<td>8. <strong>Yes.</strong> Ask question which I would have during the lecture</td>
</tr>
<tr>
<td></td>
<td>9. <strong>Yes.</strong> Because of the anonymity and because of that I'm not confident in order to raise my hand</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>1. <strong>No.</strong> I would raise my hand and ask personally</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. <strong>No.</strong> I can always ask the professor directly</td>
</tr>
<tr>
<td></td>
<td>3. <strong>No.</strong> Lectures are made to interact with teacher personally. This app would lose the personal interaction</td>
</tr>
<tr>
<td></td>
<td>4. <strong>No.</strong> Because I'm not shy</td>
</tr>
<tr>
<td></td>
<td>5. <strong>No.</strong> It would take me time to write the question and at the same time I have to follow class. Teacher possibly would be already in the next topic by the time he gets my question</td>
</tr>
</tbody>
</table>
## Appendix C: Student responses in question 13

<table>
<thead>
<tr>
<th></th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>It will be probably successful among the shy students and unconfident</td>
</tr>
<tr>
<td></td>
<td>students, so they would rely on it</td>
</tr>
<tr>
<td>2</td>
<td>Cross-Platform implementation. Only bad is that there is an assumption of</td>
</tr>
<tr>
<td></td>
<td>owing a certain device/software</td>
</tr>
<tr>
<td>3</td>
<td>None! Great work!</td>
</tr>
</tbody>
</table>