6-13-2013

Mobile educational forum

Suzanne Lien
Santa Clara University

Alexandria Shearer
Santa Clara University

Santa Clara University
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 13, 2013

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Suzanne Lien and Alexandria Shearer

ENTITLED

Mobile Educational Forum

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THESIS ADVISOR

DEPARTMENT CHAIR

MOBILE EDUCATIONAL FORUM

by

Suzanne Lien and Alexandria Shearer

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California

June 13, 2013

Mobile Educational Forum

Suzanne Lien and Alexandria Shearer

Department of Computer Engineering
Santa Clara University
June 13, 2013

ABSTRACT

Good education is essential for youth around the world. However, many communities in the developing world do not have access to basic knowledge and resources, like HIV prevention, women's rights, and life skills. As a result, the youth in these communities grow up and face difficult substandard lives. The Mobile Educational Forum addresses these issues by enabling a central party to distribute educational information to youth in emerging markets by making use of resources they already have on hand—mobile phones. Through our mobile web application, young people not only receive valuable information, but are also able to engage in discourse with their peers in geographically remote areas. Users with basic phones without Wi-Fi may still benefit from our application, which allows them to be able to send and receive text messages from the central party. In this way, the Educational Forum hopes to create positive social change by bringing knowledge and empowerment to youth who are otherwise destined to continue to live in poverty and inequity.

ACKNOWLEDGMENTS

We would like to express our sincerest gratitude to Professor Silvia Figueira, our advisor, for her helpful insights and spirited support throughout the process of Senior Design.

In addition, we would like to thank our collaborators, Equal Access, for providing us with a project that has given us the opportunity to make a larger impact than we could have conceived.

And finally, we would like to thank the School of Engineering, in particular, the Computer Science and Engineering department. We attribute our successes to this wonderful institution filled with faculty and staff dedicated to our education. You have inspired us and helped shape our future.

This thesis is dedicated to those we acknowledged above, in addition to the countless family and friends that have supported us throughout our four years at Santa Clara University.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Background

1.1.1 SCU Frugal Innovation Lab

The frugal innovation lab at SCU aims to "develop and apply novel technologies that underserved communities can afford and sustainably use over time." The hope is that these technologies "improve the quality of life for individuals while simultaneously catalyzing economic transformation" (Frugal Innovation Lab). Our advisor, Professor Silvia Figueira, is actively involved in the Frugal Innovation Lab and has provided us with the knowledge and resources that inspired our project.

1.1.2 Equal Access

Equal Access is an organization that has partnered with the SCU Frugal Innovation Lab and is a collaborator in our project. From their website:

> "Equal Access is an international not for profit organization (501c3), headquartered in San Francisco and working throughout Asia, Africa and the Middle East. A communications for social change organization that combines the power of media with community mobilization, Equal Access creates customized communications strategies and outreach solutions that address the most critical challenges affecting people in the developing world such as women's & girls' empowerment, youth life skills & livelihoods, human rights, health and civic participation & governance."

They have partnered with us in the hope of creating a purposeful, impactful application that touches the lives of the communities they work with by utilizing mobile devices, a plentiful resource in these emerging communities.

1.2 Problem Statement

In emerging markets, or areas of the world undergoing rapid development and industrialization, sales of mobile devices are eclipsing the sales of laptops and other computing platforms. Estimates say that by 2020, every person in Africa will own a mobile phone, and in 2011 census data showed that households in India were more likely to have a mobile phone than a toilet.

Because of the pervasiveness of mobile devices in emerging markets, there is tremendous opportunity for mobile developers to create purposeful applications with a positive impact. For example, physical and geographic barriers like distance in rural areas disconnect communities, preventing youth from connecting with their peers. In addition, access to information and resources is limited in these remote areas, which limits prospects for youth in their careers and education.

This is why we present our mobile learning application—we aim to address these issues by creating a mobile "forum" that allows for a central party to facilitate conversations or provide helpful resources to youth in developing areas, as well as allow youth to stay connected with each other. We will make our system accessible to this group by ensuring that our application will work atop SMS technology, which is supported by the most basic of phones. In addition, we will create a web application that will work for users with a phone that supports Wi-Fi connectivity. In this way, we will reach a wider user base, thereby increasing our potential impact.

Our application will be backed by Equal Access, a radio broadcasting company with a presence in nine countries. Equal Access has a broadcast reach of 90 million people, which will be leveraged as a user-base for our application. The primary issues Equal Access addresses are the empowerment of women, educating youth with life skills and improving their career prospects, human rights, health, and civic participation in emerging democracies. In addition, the SMS broadcasting feature may be useful in

disaster-response scenarios. Equal Access will give our application the presence to give the youth in emerging markets access to the education and resources they need to succeed.

1.3 Related Work

There are many web applications in existence that have a similar premise of supporting messaging across mobile platforms, including support for SMS. Ours, however, is meant to cater specifically for the emerging world by providing a forum for education. Our emphasis is placed on discourse to bring these communities together, and for education to improve their prospects for a strong future.

1.4 Objectives

We hope to accomplish the following objectives within the time span of our project:

1. Create a fully featured dashboard for Equal Access to scan, filter, and broadcast messages to subscribers.
2. Bring the web application component for smartphone users to completion.
3. Simulate the sending and receiving of text messages by implementing modules that are easily replaceable with an actual SMS framework.

2. Requirements

Requirements describe the functionality of the system (functional), the manner in which the functional requirements must be achieved (non-functional), and the requirements that constrained our implementation (design constraints). Below we list the functional and non-functional requirements as well as the design constraints.

2.1 Functional Requirements

The system will:

- Allow Equal Access to broadcast information to subscribers
- Allow subscribers to comment on the broadcasts they receive

2.2 Non-functional Requirements

The system will be:

- Usable
    - Easy to visually scan content
    - Easy to navigate
    - Easy to send messages
- Maintainable
    - Easy to scale
    - Easy to update
    - Easy to add more features, etc.
- Portable
    - Work across different mobile devices and browsers
- Secure
    - Provide users with the option to remain anonymous
    - Prevent unauthorized access to the database
    - Prevent common web-hacking exploits

2.3 Design Constraints

- Must work on all mobile phones
- Work with SMS technology (texting) and the web for Wi-Fi enabled phones

3. Conceptual Model

This section presents the basic concept of our system. We are modeling our system with two classes of users, namely the broadcaster (Equal Access) who sends out the content, and the subscribers who receive the broadcasts via SMS or the web application.

3.1 Broadcaster/Subscriber Model

During our first implementation stage, the subscribers will be able to send messages to the broadcaster and the broadcaster may broadcast to all subscribers, either via SMS or by starting a new thread on the web application. This concept is visualized below in Figure 1, where the Equal Access figure represents the broadcaster, and the subscribers are the group of people with phones.

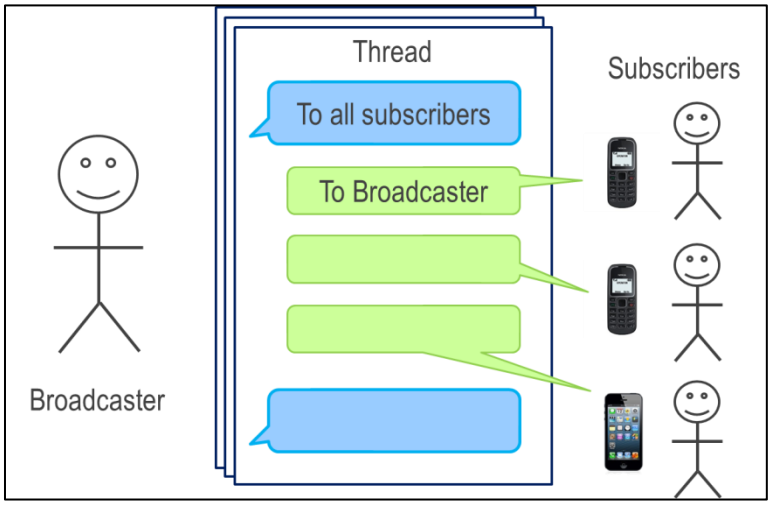


Figure 1: Broadcaster/Subscriber Model

Equal Access will be able to organize the system information into threads, which store the broadcasted messages and the responses from the subscribers. Subscribers will either receive the broadcasts via SMS or be able to view the content of the entire thread through the web application if they have a Wi-Fi enabled phone.

3.2 Broadcaster Interface

The broadcaster for Equal Access may access this information in the page shown in Figure 2, which is a basic web interface for managing the content.

Figure 2: Equal Access Dashboard

The interface in Figure 2 is a dashboard for all of the use cases that the broadcaster may perform. For example, should the broadcaster wish to create a new thread, he/she should use the add button in the threads pane on the left as seen in Figure 3. To search through threads, there is a similarly located magnifying glass button. The threads pane lists past threads in order of most-recently created, containing the name of the thread, the time of its creation, and the first few characters of the first broadcast for context.

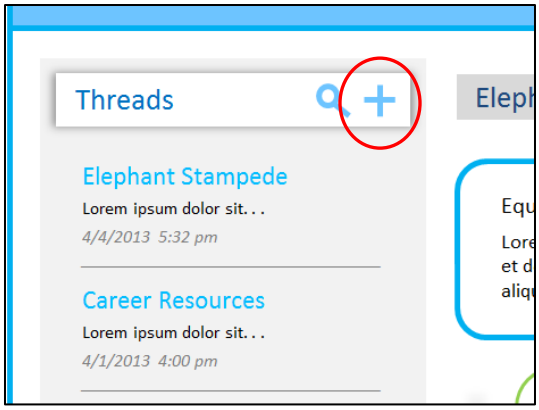


Figure 3: Create New Thread Icon

The central/main content pane contains the information about the current thread selected, including the title, time of creation, etc. There is a search button for the specific thread to allow the broadcaster to search for specific keywords, users, etc. There is also a "create new message" icon on the right as shown in Figure 4.

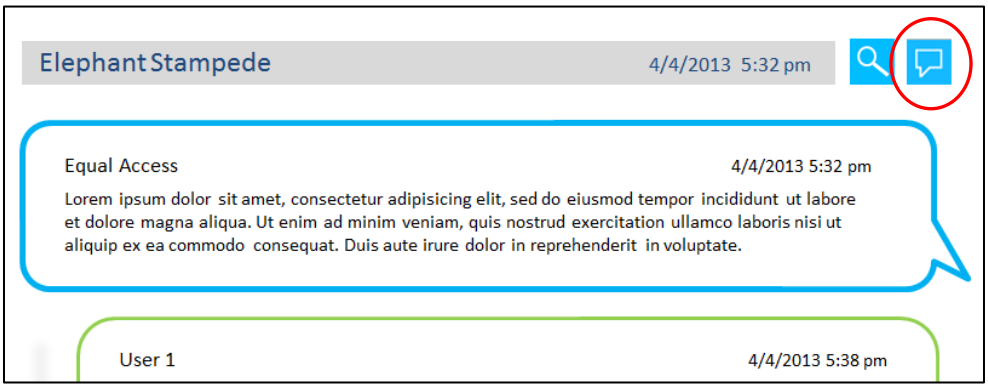


Figure 4: Create New Message Icon

The messages listed in the main pane are also color coded. Equal Access has blue coded message bubbles while the subscribers are green. The bubbles include the content of the messages, as well as the time sent/received and a username.

3.3 Subscriber Interface

3.3.1 SMS interface

The SMS interface for the subscribers is very simple. They will receive broadcasted messages through text, as shown in Figure 5. They can reply similarly, which Equal Access will receive in the interface shown in Figure 2.

Figure 5: Subscriber Interface

To subscribe to the system, users would simply text "ADD <username>" to the number provided by Equal Access. The username would be optional should they wish to remain anonymous. To unsubscribe, they would simply text "STOP." This is visualized in Figure 6.

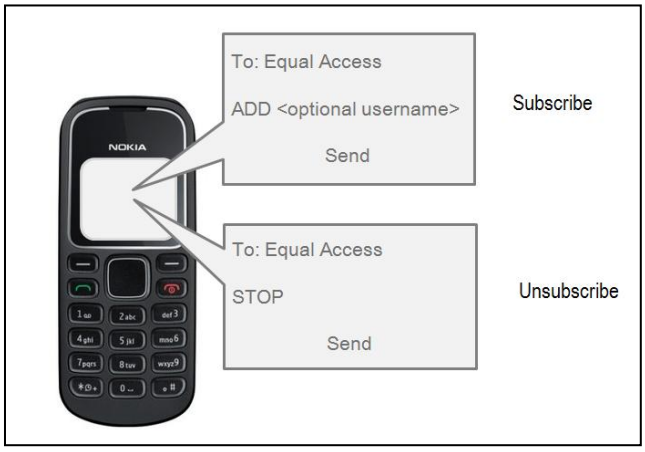


Figure 6: Subscription/Unsubscription Example

3.3.2 Web Application Interface

The Web Application Interface would be reminiscent of the Equal Access dashboard, except users would experience a mobile-optimized site. An example of the messages for a single thread displayed through the mobile interface would be similar to Figure 7.

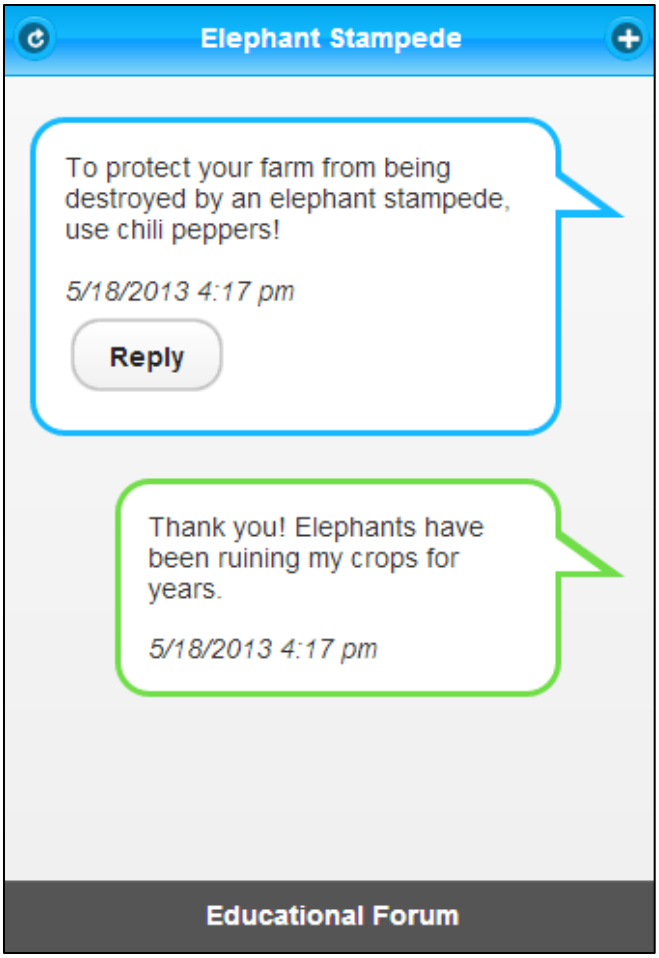


Figure 7: Web Application Interface

4. Use Cases

Our system will be used by two primary users: broadcasters and subscribers. Broadcasters are responsible for sending information to the subscribers. Subscribers will

receive the information and can reply with comments. For each of these users, we list the functionalities they can perform below.

Broadcaster:

- Login
- Create thread
- Broadcast
- Search
- Reply to individual comments
- Logout

Subscriber:

- Subscribe
- Comment
- Unsubscribe

A more in-depth narrative of each use case is shown in Table 1.

Use Case	**Actor**	**Pre-condition**	**Post-condition**	**Scenario**
Login	Broadcaster	Admin has previously created an account	Broadcasters are logged into the system and are taken to the dashboard page	1. Enter username and password 2. Login *Exception:* Incorrect username/password combination. Users will be prompted to double check their username and to re-enter their password.
Broadcast information	Broadcaster	The broadcaster is logged in to the system	A broadcast is stored in the database and is sent to all subscribers; user is taken back to the dashboard page	1. Click the broadcast icon 2. Type a message and submit
Search	Broadcaster	The broadcaster is logged in to the system	The page displays only threads, broadcasts, and comments that contain the associated search string	1. Click on the search box and type a string to search for 2. Hit enter or click submit

Use Case	Actor	Pre-condition	Post-condition	Scenario
Reply to individual comments	Broadcaster	A broadcast post and comment must exist	Text message is sent to an individual user	1. Click on a comment's associated 'Reply' button 2. Type a message to send to that individual and submit
Logout	Broadcaster	Logged in to the system	Logged out of the system and taken back to the login page	1. Click the 'logout' button
Subscribe	Subscriber	None	User is added to the database as a subscriber	1. Send a text message that contains the command "ADD <username>" to Equal Access; the username is optional and can be omitted
Comment	Subscriber	User must have received a broadcast to comment on	Comment is sent to the system and is stored in the database	1. Click on a broadcast's associated 'Reply' button 2. Type a message and submit
Unsubscribe	Subscriber	User is subscribed to the system	User will not receive any text messages from the system	1. Send a text message that contains the command "STOP" to Equal Access

Table 1: Narrative Description of Each Use Case

5. Architectural Diagram

Although our goal is to create an application that even basic phone users can use, our system is not currently making use of SMS technology for several reasons. For one, it is not possible for our server to receive a text message directly from a phone; additional software needs to be integrated. Secondly, in order for the system to be fully functional with SMS, service with a mobile provider is needed, which we don't have easy access to.

Lastly, we simply do not have enough time to incorporate SMS into the system. For these reasons, we are building a functional web application that works using the internet. In this section, we provide a description of the architectural models for our web application and the system once SMS is integrated.

5.1 Web Application Architecture

Our system is based on the standard client server architecture for web-based systems as shown in Figure 8. We are using a large database to store and archive broadcasts, comments, and information about the users. With the web application that we have built, the broadcaster (i.e. Equal Access) and the subscribers are directly communicating with the server.

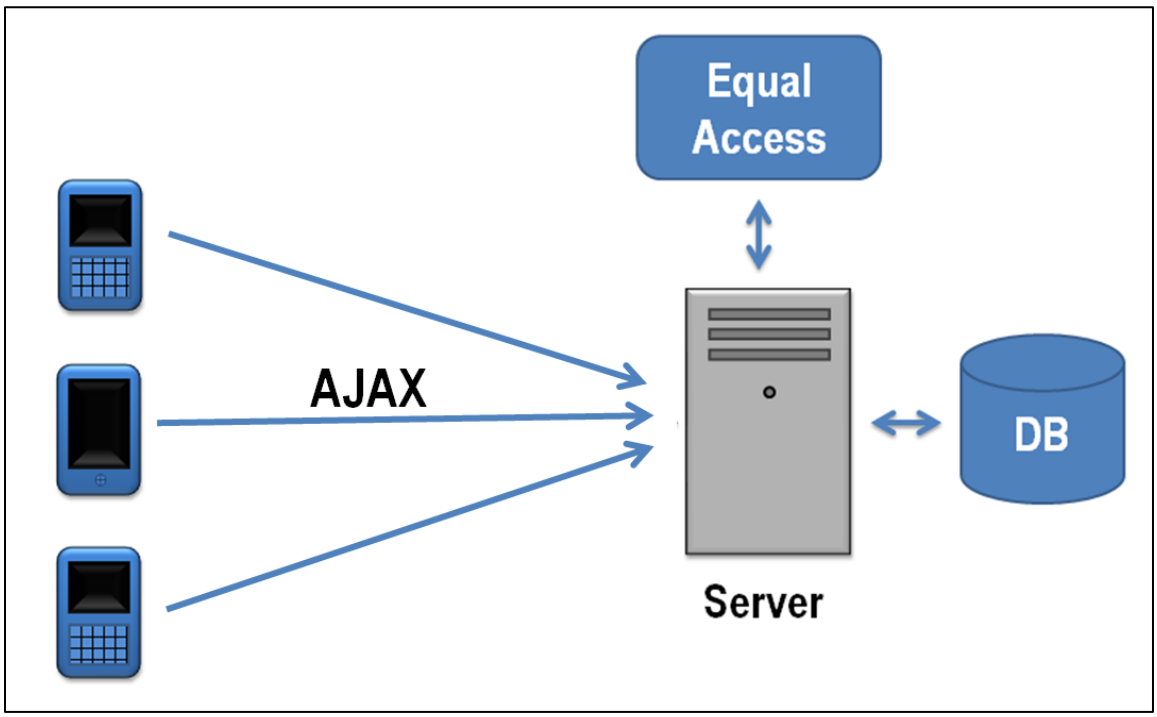


Figure 8: Architectural Diagram of the Web Application

When the broadcaster creates a new thread or broadcast, that information is sent to the server to be stored in the database. New broadcasts, however, are not sent to all our subscribers because of a limitation with our architecture; the server has no way to identify

the clients in the system. Instead, once clients are subscribed to the system, they will send AJAX requests to the server to either comment on a broadcast or to retrieve new information.

5.2 SMS Architecture

The SMS system is very similar to our web application. The broadcaster's interaction with the server remains the same. However, instead of having subscribers retrieve new content as it becomes available, text messages will actually be sent to subscribers using an SMS framework such as Frontline SMS. Frontline SMS is software that puts a computer on a mobile network and enables the computer to send and receive text messages much like a phone. As seen in Figure 9, a computer running Frontline SMS will intercept messages being sent between the server and the subscribers.

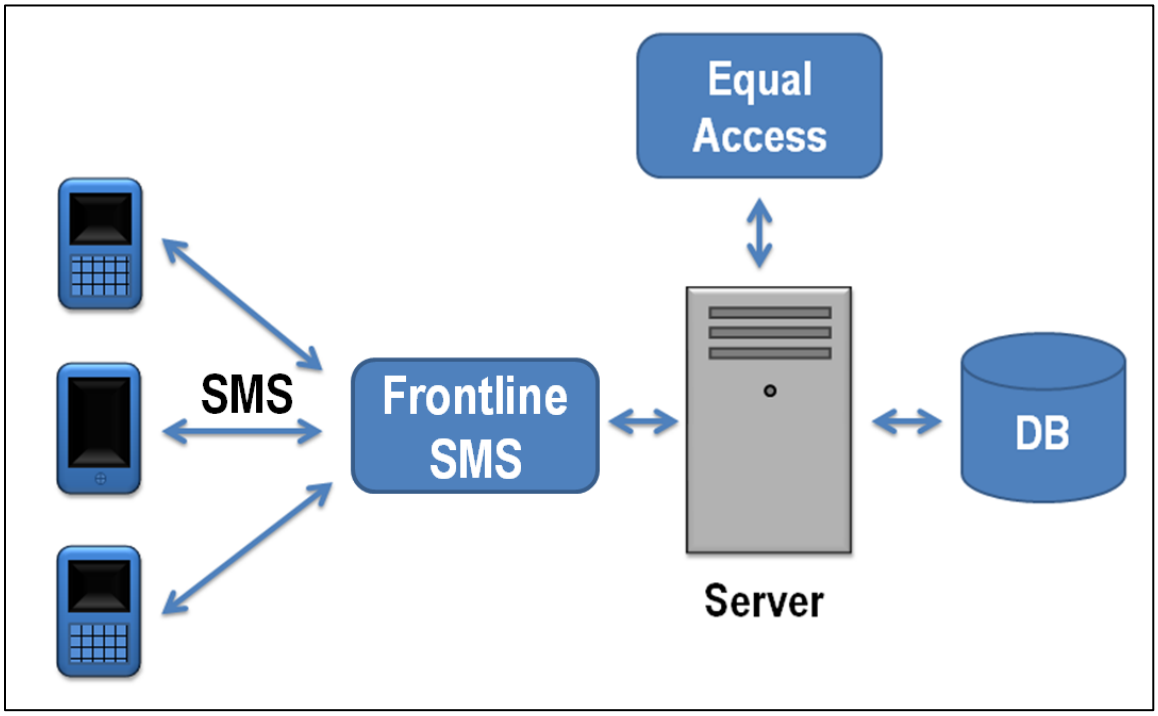


Figure 9: Architectural Diagram of the System with SMS

6. Technologies Used

To build our system, we are using the technologies listed in Table 2. All of the technologies we are using are open source. If there are software updates in the future, new developers can easily download the updates online and continue development.

Technology	**Description**
Hypertext Markup Language (HTML)	The main markup language used for displaying web pages and other information displayed in a web browser.
Cascading Style Sheets (CSS)	A style sheet language used for describing a web page's looks and formatting.
Hypertext Processor (PHP)	A general purpose server-side language designed for producing dynamic web pages.
JavaScript	A scripting language that is commonly used for creating dynamic forms and enhanced user interfaces.
JQuery	A scripting language that is commonly used for creating dynamic forms and enhanced user interfaces, a Javascript library that is often used for its ease and dynamism.
JQuery Mobile	JQuery for mobile devices.
MySQL	A database for storing our posts and user data.
PhoneGap	A mobile development framework that enables mobile applications to be built using web technologies instead of device-specific languages.

Table 2: List of Our Technologies

7. Design Rationale

After meeting with Equal Access and learning about the type of application they wanted, we came up with a design—largely driven by the requirements listed in section 2—for the system and implemented it. In this section, we discuss the design decisions we made and explain why we made them.

7.1 Broadcaster Interface

Like other existing forums on the web, we decided to group related broadcasts and their associated comments into threads for better organization and ease of search. The layout of the broadcaster's dashboard was centered around this concept of threads. To make scanning the forum content easier, we color-coded the posts by users: content posted by the broadcaster is outlined in blue while comments submitted by subscribers are outlined in green. Additionally, replies to messages are indented out to display conversation between users.

Because the number of comments for each broadcast can potentially grow to hundreds—if not thousands—of messages, we added a feature that allows Equal Access to collapse and hide content within each broadcast to more quickly find and view the comments of the particular broadcast of interest. With a colossal number of messages, we are also providing a feature to search through and filter threads, broadcasts, and comments that contain the specified keywords.

7.2 Subscriber Commands

In order to start receiving broadcast text messages, users have to subscribe to the system to indicate that intent. Having built our system so that SMS can later be easily integrated, the subscriber commands were designed around the constraints of SMS. When a text message is sent, we only know three pieces of information: who the message is addressed to, whom the message came from, and the message itself. Out of these three, the only field that is not constrained is the message, so this is where users have to indicate their intent to subscribe. But because the message field can contain a variety of things, users have to follow a very specific format—like spelling and capitalization—to successfully subscribe. Due to this strict specification, we try to keep subscriber commands short and simple to reduce the number of errors users may run into as they attempt to subscribe. An unsubscribe feature is also provided if users want to opt out of receiving messages from the forum.

7.3 Web Application

Our original intent was to build the Mobile Forum using SMS so that the forum can be used even by basic phone users. However, we were not able to actually use SMS to build our system for several reasons:

1. The server in the client-server architecture cannot receive a text message from an actual phone; additional software needs to be set up.
2. Service with a mobile provider is needed to make SMS fully functional and we do not have access to this.
3. We simply do not have the time to integrate SMS.

Due to these limitations, the objective of our project is to build the infrastructure for SMS to later be integrated. In the meantime, as we developed our system to simulate the sending of SMS messages, we have created the beginnings of a web application that works using the internet to send and receive information.

Although a web application is currently not widely accessible by many in the developing world because of limited internet access, smartphones will become more ubiquitous in the future so that even people in the developing world will own them.

7.4 Technologies

Since we are building a web application, we chose standard, widely-used web technologies to build our system. We also chose the technologies listed in section 6 because we can easily access them in the Santa Clara University's Design Center and we also have experience working with most of these programming languages. Additionally, all of our chosen technologies are open source, so if there are software updates in the future, new developers can simply download the new updates and continue working on the project.

8. Test Plan

Our test plan is a simple unit test for each use case we presented previously, in addition to a subset of common web-exploits that will fail with our application. We will try to prevent against as many of the top ten common web exploits listed below (Category: OWASP Top Ten Project):

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross-Site Request Forgery (CSRF)
- Using Known Vulnerable Components
- Unvalidated Redirects and Forwards

Since performance in a web application is critical, we will also test the responsiveness of the site in seconds.

9. Development Timeline

Having changed our project part way through Winter quarter, we spent the majority of our time that quarter creating a new design document and revising it. We were only able to begin the implementation phase of our system at the beginning of Spring quarter. Each of our responsibilities for the project's development is shown in Figure 10.

SPRING Quarter											
	Week										
	SB	1	2	3	4	5	6	7	8	9	10
Implementation											
Database											
Broadcaster:											
Login/Logout											
UI											
Create threads											
Send broadcasts											
Reply to comments											
Search											
Subscriber:											
UI											
Receive broadcast											
Send comment											
Subscription											
Maintenance											
Testing											

Legend:	Team	Alex	Suzanne	Deadline

Figure 10: Gantt Chart of Our Development

10. Societal Issues

We have examined the ways in which our project addresses the following societal issues. Note that the content that Equal Access will broadcast to their subscribers may also address these issues.

10.1 Ethical

We believe we are acting in accordance with moral conduct by intentionally creating an application that helps the underserved. The primary ethical issues surrounding our project are those of equity and the basic rights of every human to have access to information

critical to their rights as human beings. The Universal Declaration of Human rights asserts the right of all persons to have access to such information, no matter their social location. As an extension, any problem that involves impediments to one's access should be resolved. This was the primary motivation for our project. In addition, we have considered other ethical responsibilities we have to address below.

10.1.1 Socio-Ethical Responsibility

Our ethical responsibilities have informed the way we ensured the quality of our project. We are transferring ownership of our system to Equal Access and therefore we have made every consideration and effort to provide a usable, maintainable, and secure system. We have ensured our system is reliable and secure through testing and designing for scalability as part of fulfilling our ethical requirements.

10.1.2 Project Development Ethics

Neither of us are web developers and we have had plenty to research to build our system. From this research, we have gained a lot of new information that has helped us during the process of implementation. To make sure we don't infringe on the intellectual property of others, we have properly cited all our sources. We have not yet used any proprietary software, so we have had no need to obtain licenses.

10.1.3 Team and Organizational Ethics

In addition to the ethical issues we have encountered from the development of our project, we have also made sure that we considered the ethical issues within our team, the university, and our professional community.

As a team

We treated each other fairly by splitting up all work evenly. To ensure that we would do that, we created a Gantt chart of the tasks each team member has done and will do. If we saw that one person is doing more work than another person, we re-delegated tasks so that they are more evenly divided.

Within the university

As a religious university, Santa Clara embraces the Jesuit value of service to others. To ensure that we were acting ethically within the university, we aligned our project's goals with the university's goals.

Within our professional community

As software engineers, we have made sure that we abided by the software engineering code of ethics. To accomplish this we have tried to make the highest quality product possible within our available time span. We have undergone a constant process of reconsidering or redesigning our project to fit within our ethical bounds.

10.2 Usability and Aesthetics

10.2.1 Formal Considerations

The SMS interface literally could not be simpler. The dashboard neatly and compactly organizes the content. We avoided over-complicating the content by using colors and shapes to signify meaning rather than having to explain it. As for the web application interface, we have not yet fully implemented it, but the same simplicity of the dashboard will inspire our design.

10.2.2 Usability

Behavior and Knowledge Characteristics of Our Users

The users of our system are youth in emerging markets, although anyone with a mobile device in those areas may use it. Our users will possess the following characteristics:

- Desire to connect with their peers
- Value for education
- Limited technical knowledge
- Likely to pursue activities where there is a large perceived value

The first two characteristics we presented are likely to increase the chances of user loyalty. We knew if we do not make the process of subscription and the general use cases as easy as possible, the users will likely choose to spend their time in more lucrative exploits, like working to earn money for their families. So we made sure that the use cases for subscription were intuitive.

We still have more work to do to ensure that the UI is more responsive and performs more efficiently to prevent us from losing user interest.

How We Will Ensure our Documentation and Product is Usable

- Testing: we have tested for performance and performed alpha testing to ensure our project meets basic usability requirements.
- Document as we go (comment our code): We continually document any changes in our design and reevaluate our system periodically, thereby making it more usable for ourselves and future people working on this project.
- Seek feedback from our presentation audience: The questions and feedback we receive from the audience during the design review inform us how usable and understandable our product is. We have incorporated that feedback and adjusted our design as necessary.

10.3 Others

10.3.1 Social

Our application, as a forum designed for educational discourse, is inherently social. It will bring together disparate communities in the emerging world.

10.3.2 Political

Education emboldens the public with knowledge. Equal Access certainly will be addressing politics in the use of this application, so we expect that we are taking a part in transferring political power to underserved communities in the emerging world through education.

10.3.3 Economic

So far, we have had little to no costs in building and hosting our application. Once we transfer ownership of our application to Equal Access, they will bear the costs of maintaining it, including the costs of buying servers and modems. Users will bear the cost of text messaging and Wi-Fi.

10.3.4 Health and Safety

There are little to no health and safety risks associated with our project. In fact, we expect that our application provides an opportunity to improve public health through education about health and safety risks.

10.3.5 Manufacturability

We have built a working prototype for our system using basic web technologies; however we are currently aware that our application does not utilize newer technologies in modern browsers that would enable our application to be more responsive with less work. We will transition our implementation to incorporate these technologies, which are easier to build with.

10.3.6 Sustainability

As long as the upward trend of sales of mobile devices continues in the emerging world, our application will continue to be viable if it is reasonably maintained.

10.3.7 Environmental Impact

The environmental issue associated with large mobile/web frameworks and services is that they consume a lot of power, especially when the engineers make poor software design choices. Energy is valuable in the emerging world. We have made considerations to make sure our web application is designed to limit the amount of battery life consumed when it is being used by a mobile device.

10.3.8 Accessibility

We have made considerations to make sure our application is accessible to our target user base by designing for both SMS and web technologies. Also, mobile devices are commonly owned by a whole family in emerging markets, so if a user has a disability or special needs that inhibit their use of our application, their family members are able to assist them.

10.3.9 Lifelong learning

We have begun by building our initial prototype with basic web technologies. However, this project has challenged us to become more aware of modern frameworks and web technologies that would make our implementation easier and make the application more robust. We will continue to seek out new web technologies.

10.3.10 Compassion

We hope that our application will relieve suffering that is caused from lack of education about civil and human rights, health, and career prospects, for example.

11. Future Work

With limited time to complete our project, we are not able to incorporate all the features we would like to have in the Mobile Educational Forum. In this section, we discuss future enhancements to our system for prospective students if they would like to continue our project.

11.1 Integrate SMS

With the system we have built thus far, basic phone users, who make up the majority of our targeted users, will not be able to participate in conversation through the forum. For this reason, we would like the integration of SMS into our system to be a high priority in the future.

Before starting the implementation of SMS, however, we first suggest doing some research on an alternative technology called NSS. We have heard that NSS is a texting service that allows the user to send a text message without incurring any costs. NSS also does not require internet usage and sends text messages to a short code instead of a phone number.

11.2 Integrate Third Party Notifications

Currently, in order for subscribers to receive a broadcast, the broadcaster has to explicitly create and send out a message. In reality, however, a broadcaster is not always going to be available to send information. If the Educational Forum will be used to distribute time-sensitive information (such as warnings about tornados, tsunamis, or perhaps viral outbreaks), then it would be a good idea to integrate third party systems that can automatically send notifications to subscribers so that the information is received as soon as possible.

11.3 Enable Keyword Tagging for Threads

Not all of our subscribers may want to receive every broadcast that is sent. Some may only be interested in health information and others may only be looking for human rights related information. We propose adding a feature where every thread can be tagged with a set of keywords. Using these keywords, subscribers can request subscriptions to receive only a set of particular broadcasts.

11.4 Scale for Multiple Countries

Our system is currently set up to work for one location. However, because Equal Access is an organization that reaches nine countries around the world, they would like to bring our application to all nine countries. This would require an administrator to manually set up the application with a specified database for each location to use. Rather than a manual set-up, we propose using a subscriber's location or phone number to identify their location's or country's associated forum. Scaling the application may also require making

changes to the database to handle different languages, depending on the type of phones subscribers have in certain locations.

11.5 Enhance Web Application with MMS

With the use of internet, the web application is powerful enough to transfer more complex types of media, like images and videos. For this reason, the web application version of the forum could be enhanced to distribute and receive these types of media with the use of MMS (Multimedia Messaging Service). If this feature is implemented in the future, however, basic phone users and web applications users should be distinguished; people with simple phones should not be sent images nor videos since this may incur unwanted extra costs for the subscriber.

12. Conclusion

12.1 Accomplishments

- We built a fully functional dashboard that allows Equal Access to:
    - Broadcast messages
    - Receive comments to broadcasts
    - Reply to comments
    - Create and manage threads
- We fully simulated the sending and receiving of text messages to subscribers and modularized this portion of our system so it could be easily modified to work with an actual SMS framework.
- In the process of creating our simulation for sending and receiving text messages, we created the backbone of our mobile web application because our simulation was built using mobile web technologies (HTML5, jQuery Mobile, etc.).

12.2 Learning Outcomes

12.2.1 Design for Adaptability

When we first began implementing our system, we had narrow-mindedly decided to design only for basic phones with SMS. This is why our system is fairly limited. For example, some features we do not support because of our initial design approach are:

- Subscriber-to-subscriber communication in web application
- Ability to send images and video content for smartphone application
- Keyword tagging for threads/broadcasts, etc.

If we had foreseen that a web application for smartphone users was valuable initially, we would have designed our system to easily support these features. Now that creating a fully functional web application is one of our objectives, we are worse off for not having designed for adaptability.

12.2.2 Design for Performance

Performance is critical in web applications in terms of usability, especially for mobile platforms. Our initial implementation was not optimized for performance. So our subsequent implementations added features on top of our first implementation, which degraded performance even further. Now that we recognize that performance is critical for usability, we have to re-implement our site to work within acceptable performance benchmarks, which will cost us much more time than if we had optimized for performance in the first place.

12.2.3 Modularize the Front-end System from the Back-end System

Our database was initially implemented using INI files since we did not get MySQL server accounts from the Design Center until week 2 of Spring quarter. When the back-end had to change to the MySQL database, we also had to change some of our front-end code. If we had properly modularized the code and used abstraction in the front-end, we

would not have had to change the front-end code when the back-end changed. This would have also allowed us to work more efficiently.

12.3 Advantages

12.3.1 Appropriate Technology

Our solution is designed for a target user base of youth in emerging markets and was built according to the needs of a particular organization, Equal Access.

12.3.2 Frugal Innovation

Our system relies on an already ubiquitous resource in emerging markets, mobile devices. This makes our solution cost-effective and maximizes its potential to positively impact our users.

12.4 Disadvantages

12.4.1 No SMS Integration

Most of the mobile phone owners in emerging markets have very basic phones with no web browsing capability. Due to limited resources, like the lack of a powerful modem to handle many incoming messages to a server, our system does not use SMS but simulates it. Therefore, with the current state of our system we are not capable of serving the majority of our target user base.

12.4.2 No Real Time Communication

Our system as we implemented it makes the clients, or mobile devices, query the server to check if there is any new content. This means that for our application to work in real time we have to continually poll the server for new data rather than the server sending new messages to the client whenever they are available. Real time communication is a must for an application like this and currently we do not implement it.

12.5 Closing Thoughts

Our culture takes access to information for granted. Many communities in the developing world do not have access to the internet, television, or even things we consider basic like books. Lack of information or education means that the youth in these communities do not have the skills or resources to reach their full potential in life, which could result in a very discouraged group of people who are susceptible to "extremist influences and violence" (Equal Access). Through the Mobile Educational Forum, we hope to create positive social change by distributing important information to these people and engage them in a conversation using the technologies they already have. With Equal Access, which already has a radio broadcasting station that reaches an audience of 90 million people around the world, we hope to be able to reach even more people with the Educational Forum and ultimately change more lives.

APPENDIX A: User Manual – Broadcaster

1. Overview

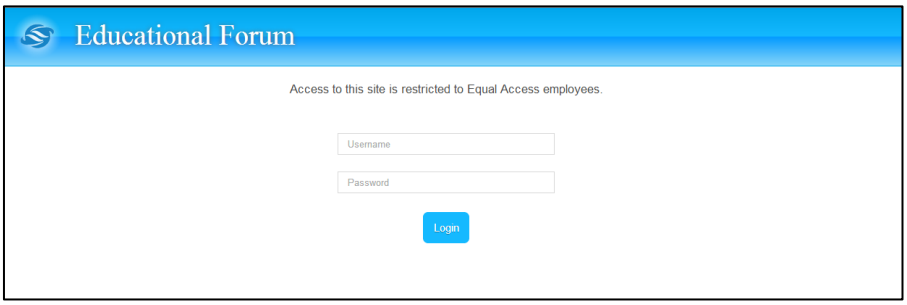
a. This part of the user manual is for the broadcaster, or the person who is working with the dashboard and has the role of broadcasting content to subscribers.

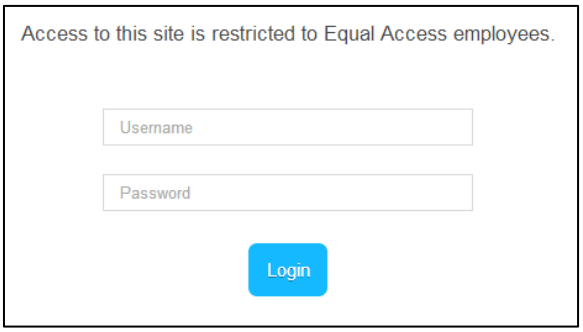
b. The subsequent sections for this manual will detail how to perform the various use cases for the broadcaster.

2. Login

a. The system administrator should have provided the broadcaster with a specific username and password.

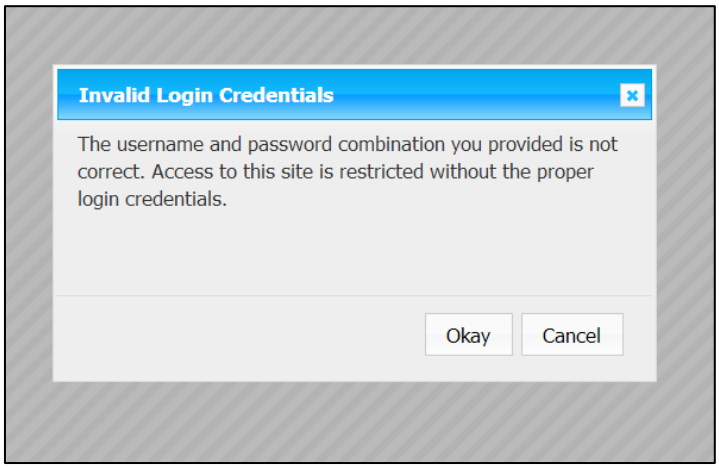
b. Below is the login page. Notice the text fields labeled “Username” and “Password”:





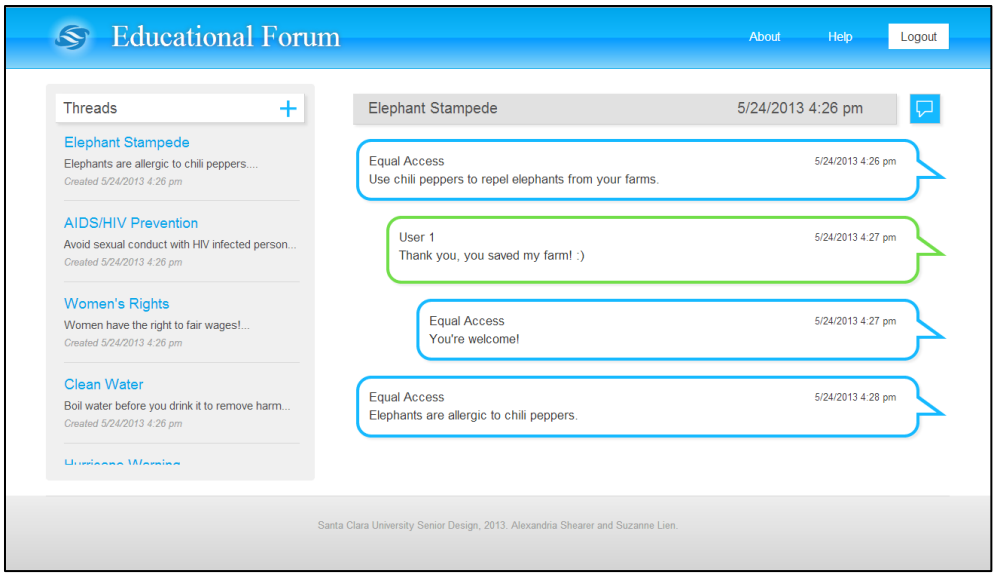
c. The broadcaster should enter the appropriate username and password and then press the “Login” button to enter the site.

d. Entering invalid login credentials will result in the following error dialog:



3. Dashboard - Basics

a. Below is the dashboard, the main interface for the broadcaster.



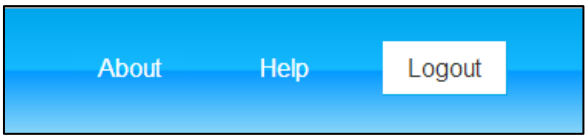
b. The thread pane is on the left side, and when it is completely filled it is scrollable with the mouse and the mousewheel.

c. The right pane contains the information for a specific thread, including the title, the time it was created, and the broadcasts and user comments associated with each broadcast. When it is filled it is also scrollable with the mouse and the mousewheel.

d. Broadcasts show up as blue speech bubbles while comments to broadcasts show up as green speech bubbles.

4. Dashboard - About

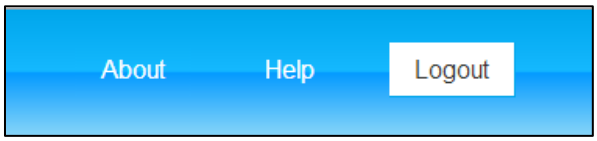
a. Notice the “About” link in the navigation bar



Clicking this link brings the user to a page that details the basic information and purpose of the site.

5. Dashboard - Help

a. Notice the “Help” link in the navigation bar



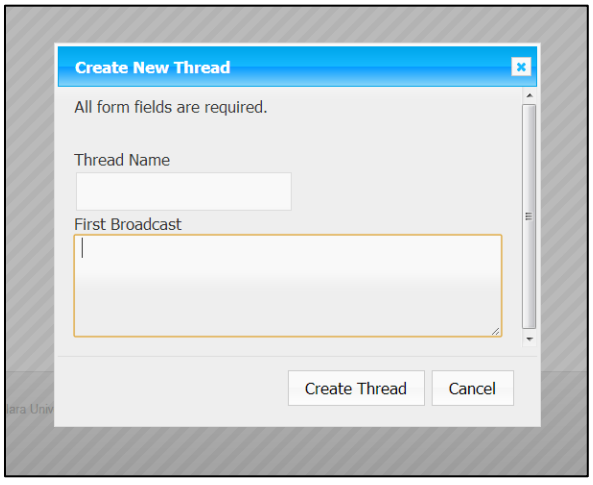
Clicking this link brings the user to a page that reiterates the information in this user manual and provides additional resources to assist the user if he or she should have any problems.

6. Dashboard - Create Thread

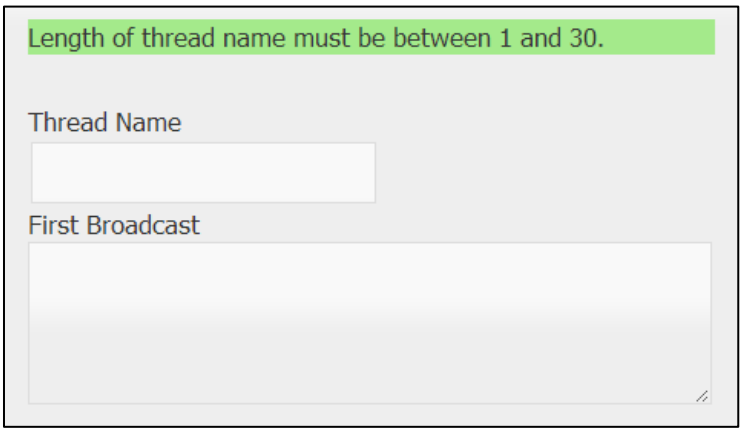
a. Every forum is populated with threads of discussion. Ours is no different.

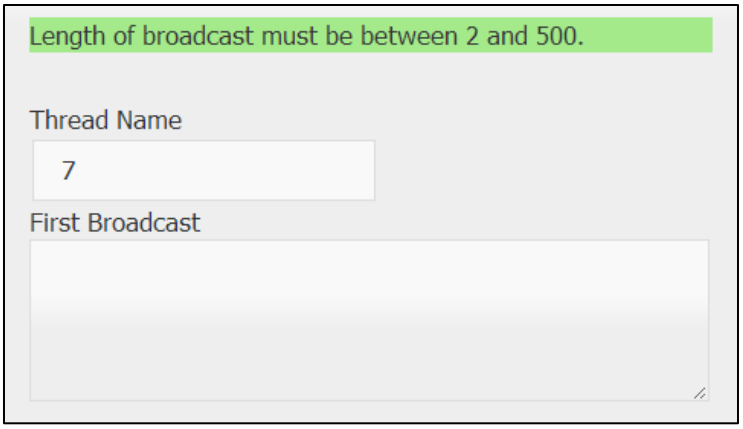
b. The create thread button is the plus icon in the title bar of the thread pane:

c. Clicking this button prompts a dialog to open, which has fields for creating a new thread:



d. The length of the thread name must be between 1 and 30 and the length of the broadcast must be between 2 and 500. Otherwise, the following error messages show in the dialog:





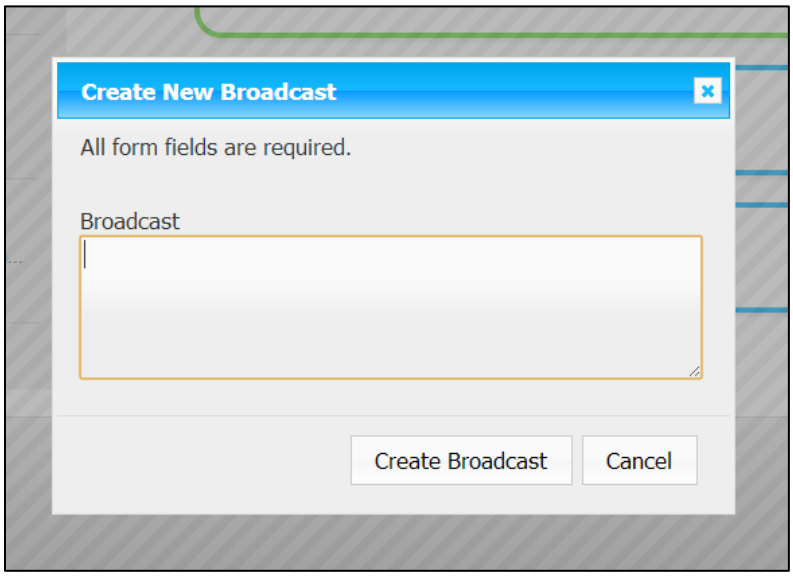
e. Once a thread is created, it will be added to the end of the thread-pane list.

7. Dashboard - Create Broadcast

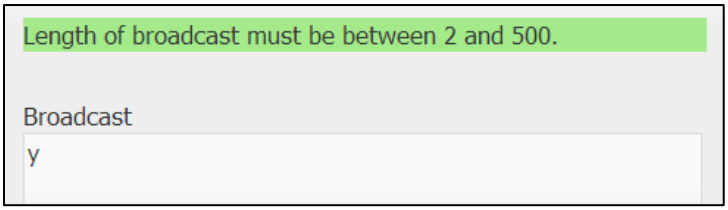
a. Creating a broadcast means adding a broadcast to a thread, as well as sending out a text message to all subscribers in the system.

b. The create broadcast button is blue button with the message icon on the top right hand side of the main pane:

c. Clicking this button prompts a dialog to open, which has a single field for creating a new broadcast message.



d. A broadcast must be more than 2 characters, otherwise it cannot be created:



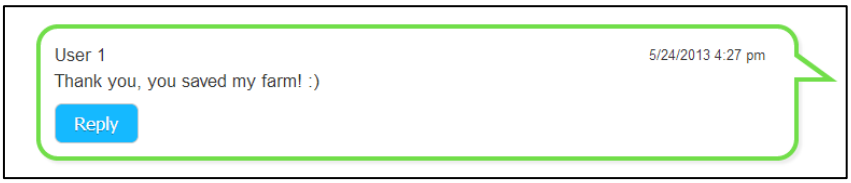
e. Once a broadcast is created, it will be added to the end of the list of content for that thread.

8. Dashboard - Reply to Comment

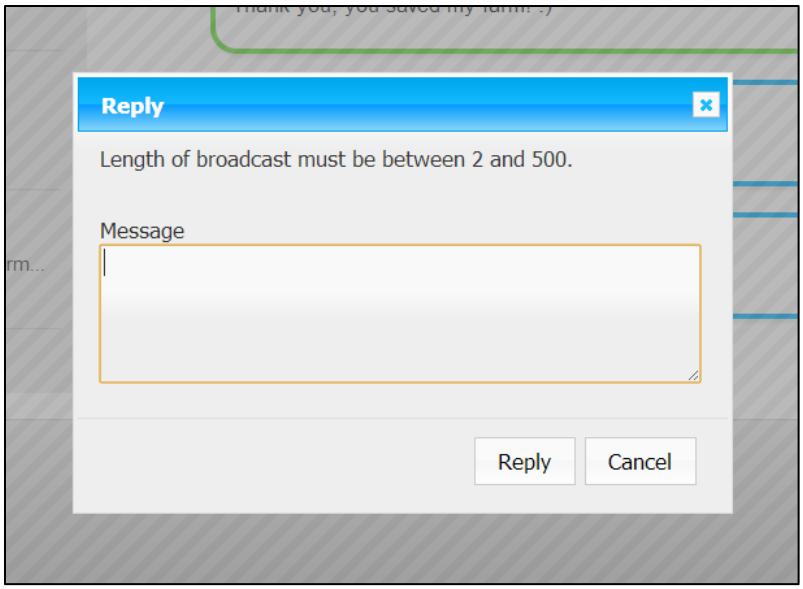
a. User comments to specific broadcasts show up as green speech bubbles. The first broadcast above it in the list of messages in the thread is the broadcast the user replied to.

b. Replying to a user comment causes it to display in the web application and results in sending a reply text message to that subscriber if that user has subscribed for the SMS component.

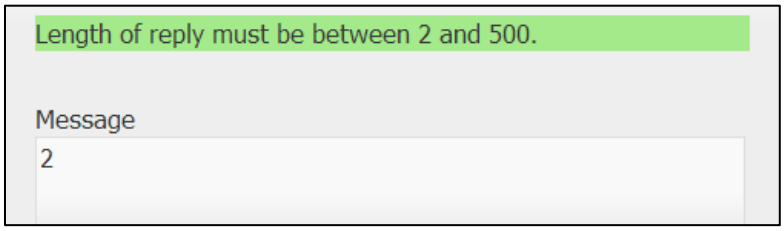
c. Clicking the user comment causes a blue reply button to appear:



d. Clicking this reply button causes a dialog to open, which has a single text field for the content of the broadcaster's reply message:



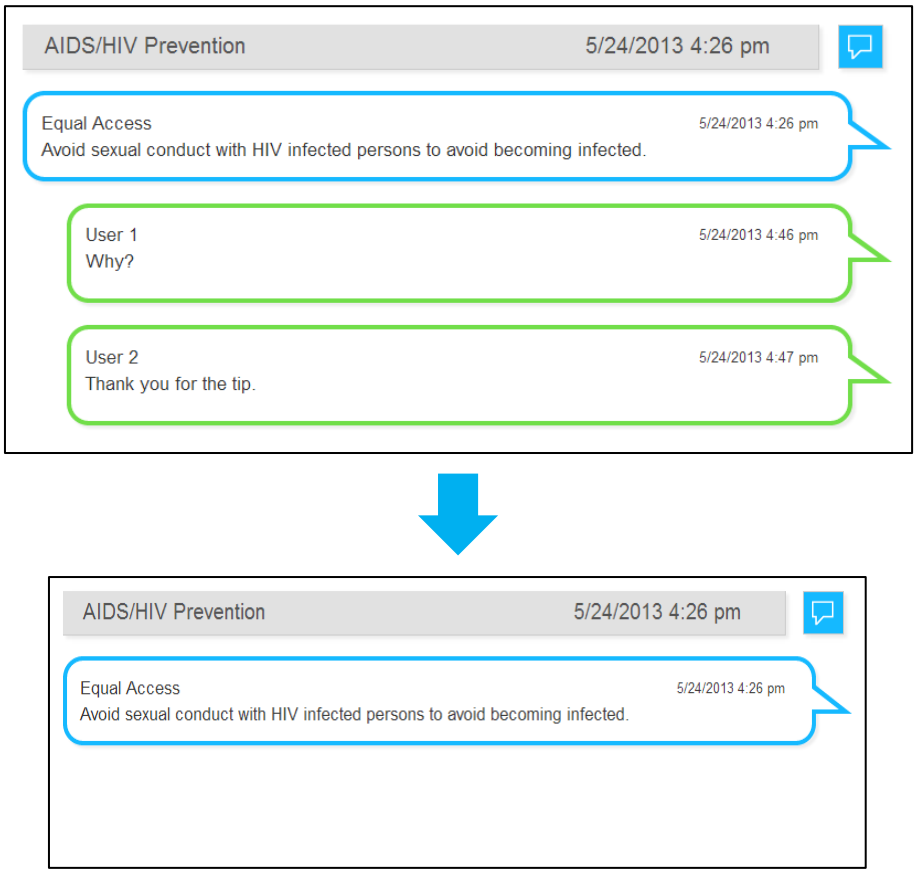
e. A reply must be more than 2 characters, otherwise it will not be sent:



9. Dashboard - Collapse Comments

a. For ease of use, the broadcaster may choose to collapse all of the comments associated with a specific broadcast.

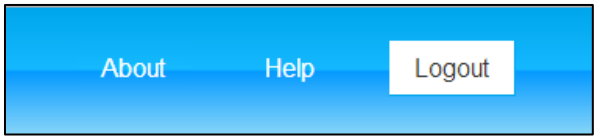
b. This can be done by clicking the broadcast:



c. This action can be undone by clicking the collapsed broadcast again.

10. Dashboard - Logout

a. Notice the "Logout" button in the navigation bar:



Clicking the logout button securely ends the broadcaster's session and redirects the broadcaster to the login page.

APPENDIX B: User Manual – Subscriber

1. Overview

a. This part of the user manual is for the subscriber.

b. The subsequent sections for this manual will detail how to perform the various use cases for the subscriber.

2. Subscribe

a. To subscribe, first click the plus icon located in the top right corner:

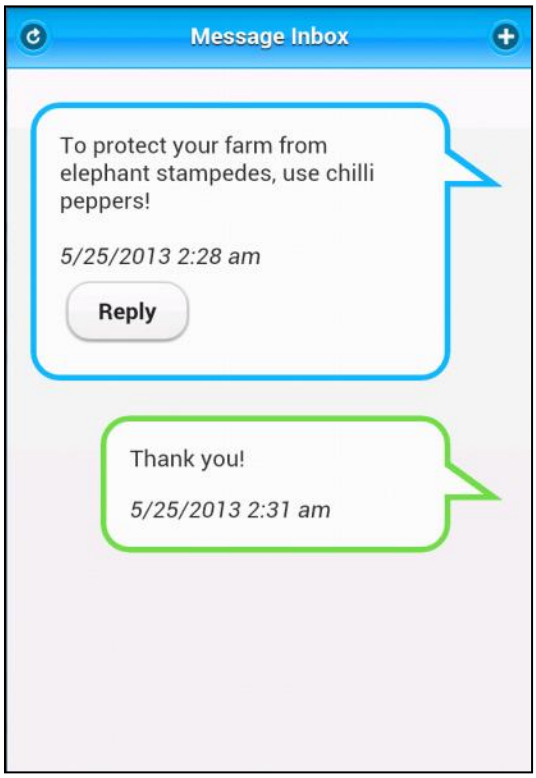
b. Type “ADD” followed by an optional username. If a username is not specified, the user is automatically given the default username, “Anonymous”.

c. Upon successful subscription, the user will begin to receive broadcasts.

3. Comment

a. Once subscribed and a broadcast has been sent, a subscriber can comment on the broadcast by clicking the broadcast’s associated reply button:

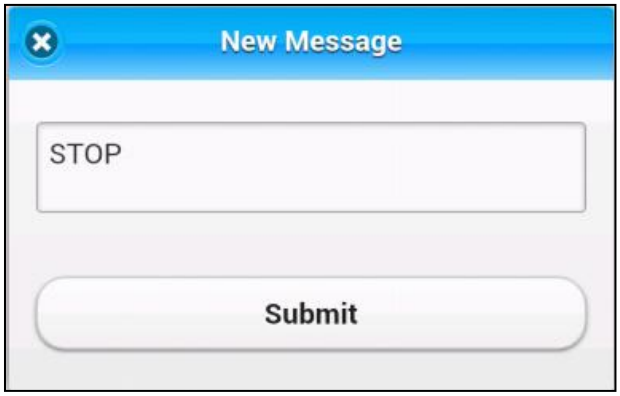
b. Type a comment and submit. Upon successfully comment, the comment will appear in the message inbox.



4. Unsubscribe

a. First click the plus icon located in the top right corner:

b. Type “STOP”. The subscriber will no longer receive any broadcast or replies from the forum.



APPENDIX C: Database Tables

We list our database tables, their associated column fields, and a description of each column in Table C.1. All the names listed here are shown exactly as they are in the actual database, including case sensitivity.

Table	Column	Description
Subscribers	subscriberID	A unique ID associated with each subscriber.
	phoneNum	A subscriber's phone number.
	signature	A subscriber's chosen username alias. If no username was specified, "Anonymous" will be the default username.
	isActive	Denotes whether the subscriber is "active". A 0 indicates not active while a 1 indicates active. An active subscriber is defined as a person who wishes to receive broadcasts or replies from the forum.
	timestamp	The subscriber's time of subscription.
Threads	threadID	A unique ID associated with each thread.
	topic	The title of a thread.
	timestamp	The time the thread was created.
Broadcasts	broadcastID	A unique ID associated with each broadcast.
	threadID	The ID of the associated thread.
	message	The broadcast message.
	timestamp	The time the broadcast was submitted.
Comments	commentID	A unique ID associated with each subscriber comment.
	broadcastID	The ID of the broadcast the comment is responding to.
	subscriberID	The ID of the subscriber who made the comment.
	message	The message content of the subscriber's comment.
	timestamp	The time the message was submitted.
Replies	replyID	A unique ID associated with each broadcaster's reply to a comment.
	commentID	The ID of the comment the reply is responding to.
	message	The broadcaster's reply message.
	timestamp	The time the message was submitted.

Table	Column	Description
SubscriberReplies	subReplyID	A unique ID associated with each subscriber reply to a broadcaster's reply.
	subscriberID	The ID of the subscriber who made the reply.
	replyID	The ID of the broadcaster reply the subscriber is responding to.
	message	The message content of the subscriber's reply.
	timestamp	The time the message was submitted.

Table C.1: Database Description

REFERENCES

"Category: OWASP Top Ten Project." - *OWASP*. Wikipedia, n.d. Web. 19 May 2013. <https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project>.

"Frugal Innovation Lab." *Santa Clara University - School of Engineering*. N.p., n.d. Web. 19 May 2013. <http://www.scu.edu/engineering/frugal/context.cfm>.

Equal Access, n.d. Web. 19 May 2013. <http://www.equalaccess.org/>.