6-11-2015

Beacon pack

Aidan Barbari
Santa Clara University

James Mack
Santa Clara University

James Terry
Santa Clara University

Recommended Citation

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 9, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY
SUPERVISION BY

Aidan Barbari, James Mack, and James Terry,

ENTITLED

Beacon Pack

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING
&
BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING

_____
THESIS ADVISOR

_____
DEPARTMENT CHAIR

Beacon Pack

by

Aidan Barbari, James Mack, and James Terry

SENIOR DESIGN PROJECT REPORT

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

Santa Clara, California

June 11, 2015

Beacon Pack

Silvia Figuiera

Department of Computer Engineering
Santa Clara University
2015

ABSTRACT

STEM technologies have the ability to spread information to those corners of the world where Internet access has yet to reach. However, there are very few technologies that can bring this necessity to developing countries, most of which do not have the capital to pursue these technologies. In this paper, we discuss Beacon Pack, a solar powered database in a traveler's pack that can bring information acquired from the Internet to these individuals in developing countries with low-level cellular phones or "feature phones." Our product tackles this issue of providing this info, specifically news articles, to developing countries through an existing cellular signal. Beacon Pack connects to any cell signal and can thus provide educational content to surrounding individuals through SMS text, one of the few communication tools that most third world citizens have access to today. We found that this project is not only incredibly beneficial, but it is also pragmatic in its approach to bridging the information gap. Through simple coding techniques, our team was able to establish this technology to support multiple users at a cost of $150 per pack. Each pack, when connected to a single cellular service signal, can supply thousands of users with information in the area, making this project highly cost efficient. Consequently, we plan for this device to be expanded into outside countries by a San Jose based non-profit, Community Technology Alliance. Through their efforts, more teams of engineers will be able to further implement Beacon Pack to include several different types of educational tools for users, all of which will be SMS text enabled.

Acknowledgements

This thesis is dedicated to our families and our sponsor, Community Technology Alliance. They have supported us during the pinnacle of our struggles and the highs of our triumphs. CTA in particular made this project possible through their encouragement and funding.

We owe our greatest thanks to our advisor, Prof. Silvia Figueira, whose insights and belief in us enabled our efforts to be doubled during our times of hardship. She has been a strong mentor over the past four years of our college career and her helpfulness cannot be understated.

We would also like to thank Thomas Martin, provided the unique skills to help us implement our MySQL database. His knowledge and expertise were vital to the success of our project and we cannot thank him enough for his time and dedication.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

I: Introduction

Developing countries lack the infrastructure of the developed world. Resources like smartphones and consistent wireless networks provide readily available information to first world. People in the third world do not have the same level of access to current and reliable information. While some third world countries have large cities with more accessibility, a majority of rural areas are only able to connect to cellular networks through feature, SMS-based cellphones. While they may call each other through this network, they cannot access the Internet's plethora of information, thus increasing the knowledge gap between certain areas of developing countries.

Due to these issues, citizens in rural areas must rely on less accurate media, such as hearsay, which can be further complicated by the government propaganda that is prominent in many developing countries. Furthermore, information is often exaggerated or simplified through communication, causing important details to be convoluted. For any direct access to information, people in remote areas must travel great distances on unreliable and sometimes dangerous roads to reach cities. However, there still remains the ability to connect to a cellular network in these remote areas, which, for most, primarily provides phone calls to be made or SMS texts to be sent.

We plan to improve access to information using a device called Beacon Pack. The Beacon Pack is a portable, solar-powered device, which can be accessed with simple text-based cell phones. Travelers trekking with this Beacon Pack can provide access to its database, which supplies educational content, to those connected to the same cellular signal as the Beacon Pack. Any SMS-based cellphone will be able to access this repository of information stored on a small and lightweight computer, called a Raspberry Pi, located inside the Beacon Pack. Users would then be able to receive a list of the content available and request more detailed information by selecting items from this list. Whenever a traveler accesses the Beacon Pack via their smartphone, the device's informational content can be updated.

With a Beacon Pack, those cut off from any direct communication infrastructure will be able to access an exponentially larger repository of information. More people will be able to share a readily available source of accurate and current information. Currently, our team has shown with a prototype that Beacon Pack can successfully translate news content from the *New York Times* into deliverable text-based content as a proof of concept for Beacon Pack's viability. There is also great potential for Beacon Pack beyond the delivery of news content, including the possibility of delivering educational and informational content.

II: Requirements

Requirements for the system, as we described in the introduction, can be divided into two categories: functional and non-functional requirements. Functional requirements define what must be done by the system, while non-functional requirements describe the manner in which the functional requirements need to be achieved. The requirements we have identified are listed below.

II.a) Functional requirements

We have identified nine functional requirements to be met by the final system in order to solve the problem explained in our introduction.

1. The system will interact with mobile phones via SMS to a portable server.
2. The system will interact with different mobile devices.
3. The system will provide informational and educational content.
4. The system will receive user input.
5. The system will allow users to input voluntary or suggested information.
6. The system will update content through an administrator web application.
7. The system may be capable of sending pictures to smart phones.
8. The system may have the capability to switch between specific languages.
9. The system may have a menu option for the user to change the settings of the application.

II.b) Non-functional Requirements

We have identified five non-functional requirements aimed at improving the user's experience. These will be verified through a dialog with the client throughout the system's development.

1. The system will be easy to maintain and update by outside programmers.
2. The system will allow multiple users to retrieve information at any time.
3. The system will be adaptable to various cellular service providers.
4. The system should have an easily accessible and defined UI for accessing the data through SMS for the user's benefit.
5. The system should update with data from a country's top news sites, allowing each user to have news in their country's language.

II.c) Design Constraints

1. The system must work on the Raspberry Pi.
2. The system must be able to display content in SMS form.

III: Design

III.a) Technologies Used

In order to best meet the requirements, we identified several technologies and devices needed to implement our design. We decided to use the following programming languages: HTML, CSS, PHP, JavaScript, Python, and MySQL. We also decided to use the following technologies: a small solar powered backpack, a Raspberry Pi, a USB or Ethernet ported Wireless or Cellular modem, and a Cellular service card.

- *HTML:* Used to formulate and construct a webpage to sign up a cellular phone for content.
- *CSS:* Used to make the website aesthetically pleasing to the admin.
- *PHP:* Used to communicate between the webpage and cellular phones with the MySQL database.
- *JavaScript:* Used to construct the user and administrator application for web access.
- *Python:* Used to communicate between multiple system files.
- *MySQL:* Used to hold and store both user data and content for all cellular phones.
- *Tropo*: Allows connection to the Internet in order to beta test our system.
- *Solar powered backpack:* Used to house and provide power to the system and devices.
- *Raspberry Pi:* Used to host all files and content.
- *Wireless or Cellular modem:* Used to broadcast and communicate with cellular devices.
- *Cellular service card:* Used to connect to and create a cellular when needed and to track service charges.

III.b) Design Rationale

We have made several design decisions that allowed us to meet all functional and non-functional requirements and enabled us to address how the Beacon Pack can be used. We decided it would be easier to explain our current design choices by splitting them by the functionalities the Beacon Pack provides.

First, we considered the communication our device will be performing. We decided that, because travelers in rural areas will use our device, it would be best to broadcast on a cellular network or to provide our own through a cellular service card. This decision was made to simplify the features of our device so the device would only need to communicate on a cellular network. Considering the Raspberry Pi has no built-in Wireless modem or connectivity, we required a wireless or cellular modem for us to incorporate the cellular service card.

Second, we constructed multiple user interfaces to go with all possible connecting devices. JavaScript was used to construct and create the applications and the user interface for the web application to communicate with the Raspberry Pi. A web application for the administrator was used because it is accessible on multiple devices, i.e. a laptop via Wi-Fi or a smartphone via the web. The programming for the application was also made easier due to the fact that one of our teammates is a web engineer. The app can then be translated as an app later on if our customer decides to do so. The connections made to the Raspberry Pi were through system files, which were coded in PHP and Python.

Third, we needed lightweight and energy efficient storage, which is why we decided to go with a solar pack and Raspberry Pi. Solar packs can be made to be powerful and compact enough to house all the devices we are storing in it. The Raspberry Pi is also compact and it can house a lot of information and data for a small energy cost.

Our group worked with our customer at Community Technology Alliance to choose these technologies and devices. We all have familiarity with a majority of the technologies and devices being used, but there are some coding gaps that had to be made. Considering we have group members with a background in web design and implementation, JavaScript was chosen because of its formatting and its similarity to certain web programming languages.

We have also left room for change within the system. We decided this flexibility was prudent because it allows outside programmers to easily update and modify the system based on the parameters and environment it might need to be adapted for. It also helped in our testing protocols, where we were able to change the implementation based on the bugs we encountered within our code.

IV: Conceptual Model

Based on our design decisions and requirements listed on the previous pages, we designed a conceptual model of our system, as seen in Figure 1.
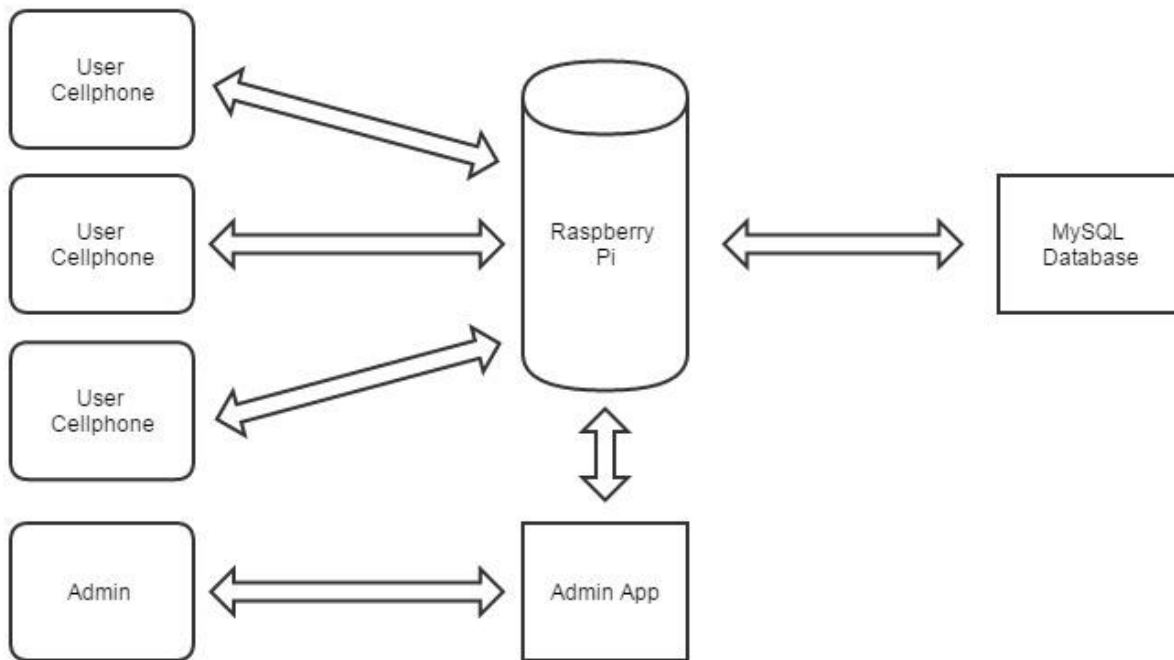

Figure 1. A low-level architectural model of the Beacon Pack's system

In this diagram, we can see a data-centric architectural mock-up of the Beacon Pack's functionality. A central MySQL database, located within the Raspberry Pi, is updated by an admin through the administrative application. Multiple users, in turn, can then access this database through their personal cellular devices.

V: User Flow

Our user flow, seen below in Figures 2 and 3, is based on whether the user or administrator will be accessing the system respectively. For the user, we initially display the database's content, and then provide them with the option to request or suggest content through their input. Afterward, they will see the system's content once more, and have the opportunity to exit the system. For the administrator, we also initially list the database's content. From there, the administrator will either add content suggested by the user or edit the existing content. Finally, they will be able to exit the system or review the system's modified content.

Figure 2. User flow diagram.
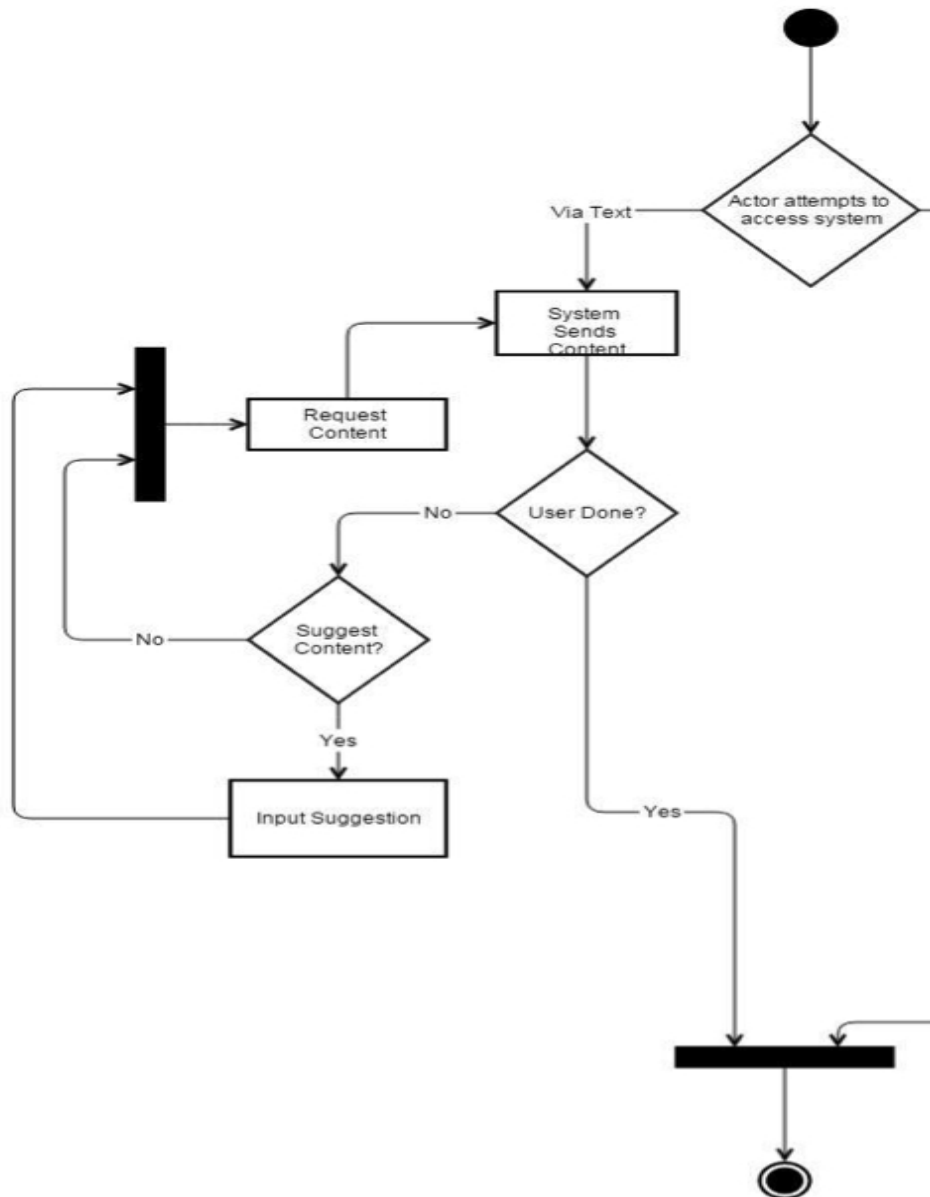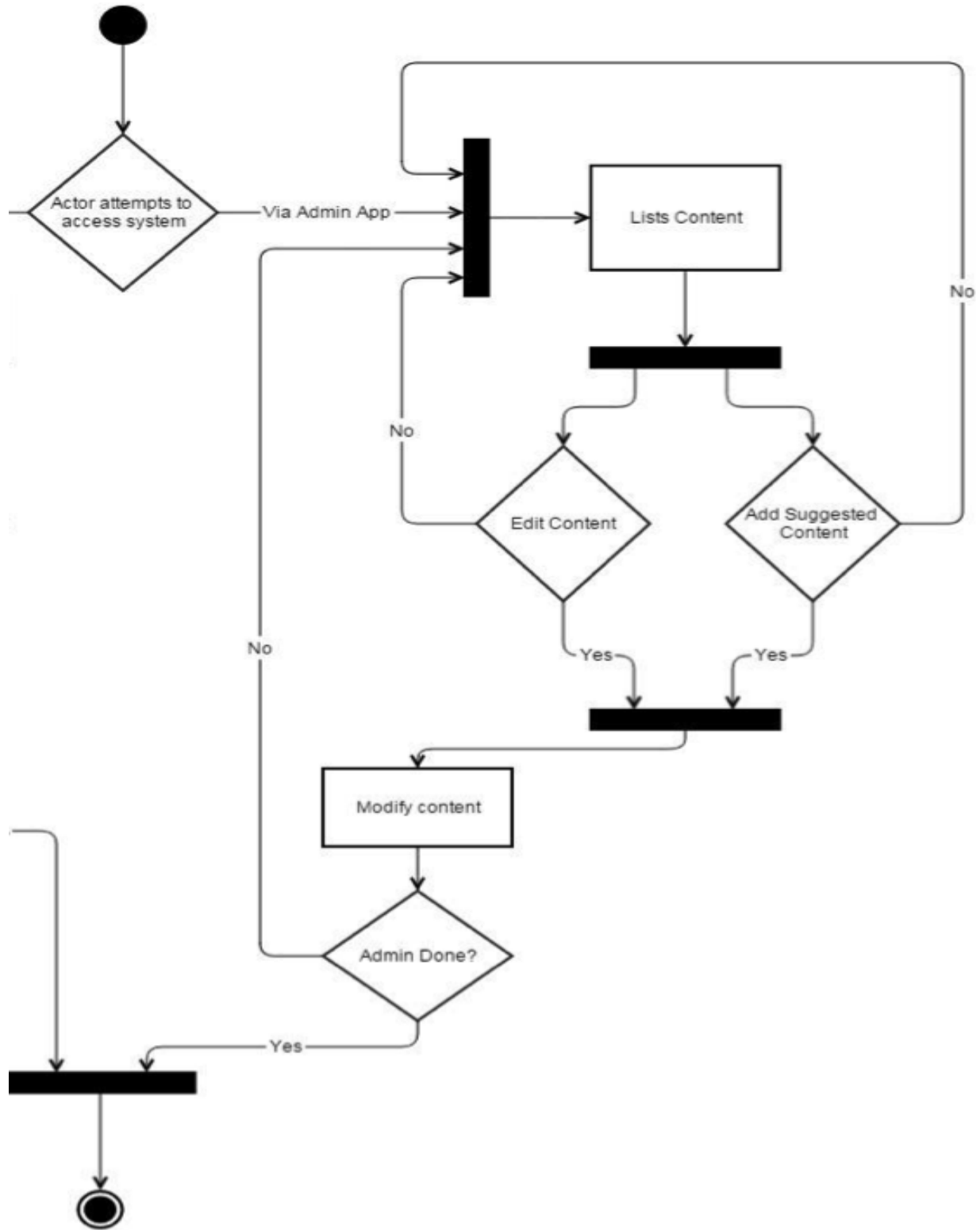
Figure 3. Admin flow diagram.

VI: Use Cases

For our system, use cases describe schedule customization functionality to the user. We have identified seven use cases that define the steps to accomplish our functional requirements, which are enumerated on in Section II. Figure 3 gives a high-level view of all use cases provided by our system.
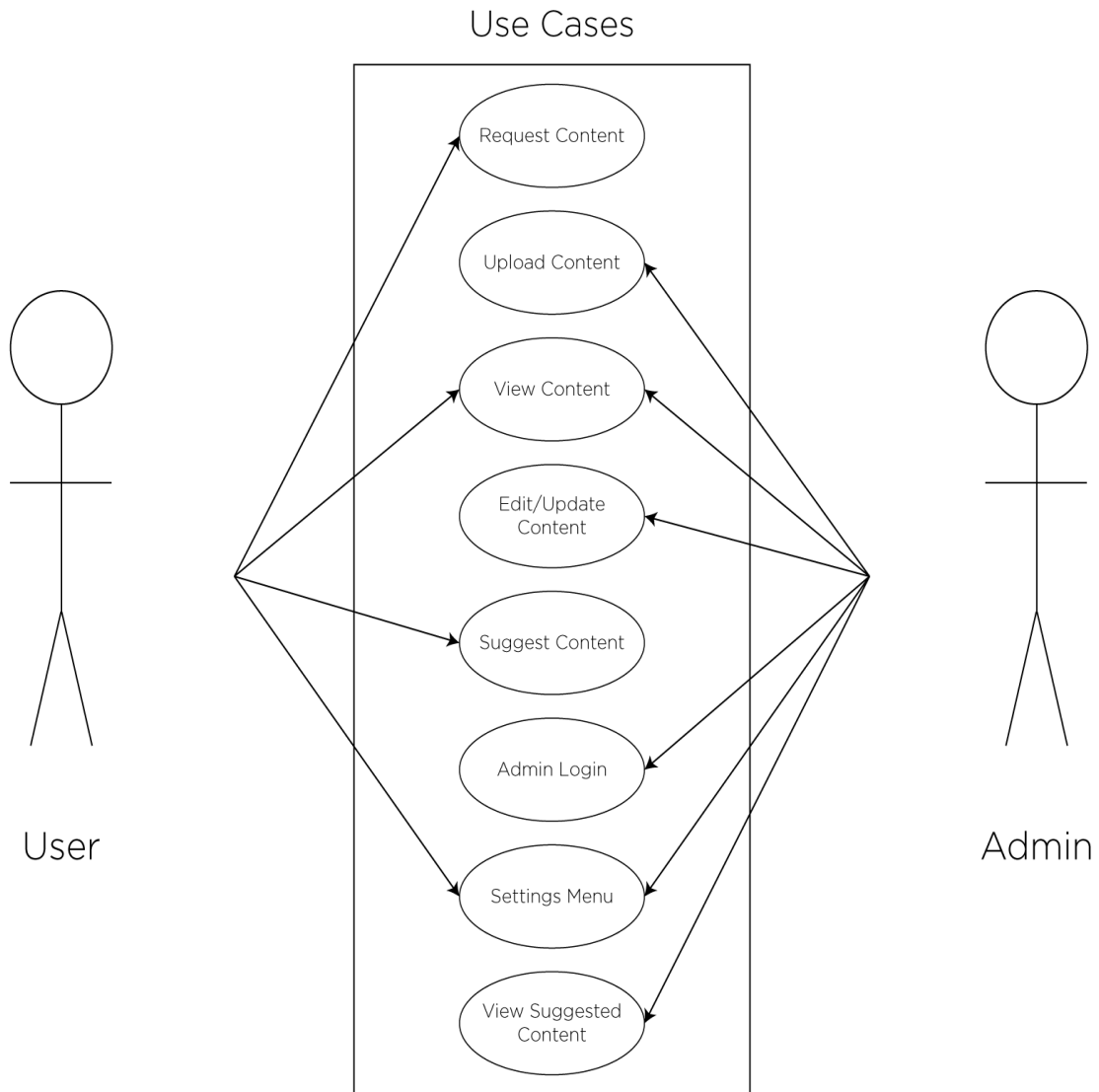

Figure 4. Beacon Pack use cases.

The following are more detailed descriptions of the use cases our system supports:

1: Request Content

    Goal: System reads request from the user and attempts to display it.

    Actors: User

    Preconditions: Content the user requests has been uploaded to the system.

    Postconditions: The system attempts to display the content in an SMS form.

    Steps:

        1. User texts system's mobile number to request main menu.
        2. User chooses to display content that has been uploaded.
        3. User selects which content they would like to see through SMS text.

    Exceptions: If the user's request is not valid, display error message.

2: Upload Content

    Goal: System's database contains content uploaded by the admin.

    Actors: Admin

    Preconditions: Admin must be logged into the system.

    Postconditions: System's database has been properly updated with content uploaded.

    Steps:

        1. Admin logs into system.
        2. Admin chooses the upload content option from the main menu.
        3. Admin chooses content to upload and types corresponding information into the controller.

    Exceptions: If content is not in the correct format, display message to the admin to change content to correct format.

3: View Content

    Goal: System displays content on user and admin's screen.

    Actors: User and admin

    Preconditions: The request for the specific content must be made by the user or admin.

    Postconditions: The system displays content in SMS form.

    Steps:

        1. User or admin requests content.
        2. Request is sent to the display, and shows content in SMS form.

    Exceptions: None.

4: Edit/Update Content.

    Goal: Correctly edit system's database of informational content.

Actors: Admin

Preconditions: Content has been uploaded to system and admin logged into system.

Postconditions: Content and system has been correctly updated based on admin's input.

Steps:

1. Admin logs into system.
2. Admin chooses which content to manipulate or update.
3. Admin makes edits accordingly.

Exceptions: The edited or updated content is not in the correct format. Display error message.

5: Suggest Content

Goal: The user inputs content that they would like the system to display in the future.

Actors: User

Preconditions: The user has accessed the system through SMS text.

Postconditions: The system holds the information the user has suggested.

Steps:

1. The user has accessed the system through SMS text.
2. The user chooses to suggest content from the main menu.
3. User inputs content they would to suggest to the admin and system.

Exceptions: The input of suggested content is not in the correct format, display message to change format of input.

6: Admin Login

Goal: The admin has successfully accessed the system.

Actors: Admin

Preconditions: The admin's login information has been correctly uploaded into the system's database.

Postconditions: The admin is able to access the main menu of the system.

Steps:

1. The admin accesses the login page for the system.
2. The admin inputs their login information.
3. The system accepts the correct login information and displays the main menu.

Exceptions: The admin's login information is incorrectly given, and must be changed to the correct information.

7: Settings Menu

Goal: The admin or user has the ability to change the settings of the system.

Actors: Admin or user

Preconditions: The admin has successfully logged into the system and the user has accessed the main menu of the system.

Postconditions: The admin or user has successfully changed the system's settings.
Steps:
1. The admin or user accesses the system.
2. The admin selects the settings menu from the smartphone app or the user navigates to the settings menu by texting the system.

Exceptions: If the user's request is not valid, display error message.

8: View Suggested Content

Goal: The admin has the ability to view content that the user has suggested.
Actors: Admin
Preconditions: The admin has successfully logged into the system and the user has accessed the main menu of the system.
Postconditions: The admin has viewed the suggested content.
Steps:
1. The admin logs in to the application.
2. The admin selects "View suggested content" on the smartphone application.

Exceptions: If there is no suggested content to view, then the application will state "There is no suggested content at this time"

VII: Implementation

During our implementation phase of the project, we focused on two main areas: the front end of the system and the back end. The front end mainly consisted of implementing the administrator's web application via HTML, CSS, PHP and JavaScript. The back end of the system used several different scripting languages used to implement the database and communication to our web application. We used Python, MySQL and Tropo for this portion.

VII.b) Front End

- HTML
    - This language was used as the basic framework for the administrative website. This highly useful language is widely used by programmers and is one of the few logical choices in building a web application because of its ease of implementation and simplicity.
- CSS
    - This is another widely used language often paired with HTML. CSS is used in many web scenarios in order to bring the framework that HTML provides into an easy functioning and accessible UI.
- PHP
    - PHP is a powerful scripting language, used in our system to connect our back end and front end. Although JavaScript allows us to manipulate the content on the database itself, PHP provides the connection necessary modify the content easily.
- JavaScript
    - We used this particular language because of its powerful ability to integrate content and functions within a web application. The connection between the database and the administrative application is built upon being able to modify the data and view suggested content from the user, both of which were made possible by the use of JavaScript.
    - In addition to JavaScript, we used AngularJS, an open-source web application framework. This addition to JavaScript was necessary because it allowed us to read our HTML page, which contains some custom tag attributes, and interprets those as directives to bind I/O parts of the page to a model represented by standard JavaScript variables. This enables powerful functions to be translated and used by our system. An example of implementation in AngularJS can be viewed on the next page in Figure 4.

```
<div class="bort">
    <div class="email" ng-repeat="news in news">
        <a href="#/main/{{$index}}">
            <span class="article">{{ news.title }}</span><span class="date">{{ news.pubdate | date }}</span>
        </a>
    </div>
</div>
```

Figure 5. AngularJS code example used in our system.

VII.a) Back End

- Python
  - This particular language is a general-purpose, high-level scripting tool used for enhanced readability and ease of use. The use of this language allows future programmers to readily understand and change portions of the system far more easily than other languages due to the benefits mentioned above. In our project, we used Python briefly to cull through our multiple PHP files to make sure that our PHP script is being called properly by the system. This helped in our testing protocols and allowed us to identify bugs more efficiently.
- MySQL
  - MySQL is primarily a database centric language that allowed us to create a table capable of holding all of our news articles. It all holds all the information of the users, making it easy for us to keep track of this necessary information. Many of our responses to the user were based on what was kept within the MySQL database, so it was important that we implemented this portion perfectly. Using this language accomplished our task of keeping information in a single place very doable.
- Tropo
  - This script activates upon a one-time use so the database and the interaction between the user and Beacon Pack are consistent. This heavily logical code uses matched keywords to understand what the user is trying to retrieve. Its connection through Wi-Fi also made it easy for us to test our system because of Wi-Fi's similarities to cellular service.

VIII: Test Plan

In order to alleviate several of the risks listed below, we imposed a test plan. Our test plan involved acceptance testing through consistently communicating with the client, as well as unit testing to ensure that our functional requirements were met.

VIII.a) Unit Testing

- Test to see if the program provides its required functions with the following test cases:
  - The system will interact with mobile phones via SMS to a portable server.
    - Test: Use SMS enabled mobile phone to connect to server.
    - Result: Successfully connect to the server.
  - The system will update content through an administrator application.
    - Test: Use administrative app to update content.
    - Result: Server database reflects content updated from administrative app.
  - The system will be adaptable to various cellular service providers.
    - Test: Use several phones on different networks to connect to server.
    - Result: All phones will connect to server regardless of network.
  - The system will be easy to fix and set up by outside engineers.
    - Test: Have third party engineers attempt to set up and/or fix the system
    - Result: Third party engineers successfully set up/fix the system without our help
  - The system will interact with different mobile devices
    - Test: Use several different types of mobile devices to connect to the server.
    - Result: Successfully connect to the server regardless of device type.
  - The system will be able to receive user input to navigate.
    - Test: SMS communication from mobile device to server and server to phone allows navigation of the structure of the system.
    - Result: Mobile device user is able to navigate the system to their heart's content.
  - The system will allow users to input voluntary or suggested information.
    - Test: Use mobile device to submit information.
    - Result: The server shows the submitted information.
  - The system will be able to allow multiple users to retrieve information at any time.
    - Test: Use multiple mobile devices to access the server simultaneously.
    - Result: The system successfully handles the multiple connections.

VIII.b) Acceptance Testing

- Sat down with the customer and allowed her to use our system. She was confident that it fulfilled all specified functional and non-functional requirements. Acceptance testing was then complete upon verifying our system with the customer.

IX: Risks

In our collaborations, we discovered our system's requirements and our design decisions posed risks that we needed to address. As a team, we decided which risks would be most applicable to our project and were able to list them as follows from highest to lowest probability. We measured the following risks by calculating the probability and severity of a risk to recognize its impact on our implementation:

Table 1. Table of Risks arranged by projected probability of occurrence. Impact and severity are estimates. Each risk contains two mitigation strategies.

Risks	Consequences	Probability	Severity	Impact	Mitigation
Bugs	The system fails to work correctly.	1.0	5	5	● Follow unit test protocols to effectively handle runtime errors and isolate faulty code. ● Have team members test the system frequently.
Misunderstood or missing requirements	The system fails acceptance testing.	0.6	7	4.2	● Start acceptance testing early so if failure does occur or requirements change, then we have time to implement them. ● In-depth conversations with the customer as to exactly what they need the system to do and why.
Time	Not completing our system by the specified deadline.	0.3	8	2.4	● Follow our development timeline closely and intermittently work on small portions of the system. ● Create deadlines for individual portions of development.

Table 1 continued.

Risks	Consequences	Probability	Severity	Impact	Mitigation
Lack of knowledge of programming language being used	Certain sections of the system take a vast amount of time to implement or are unable to be completed.	0.25	5	1.25	● Team members who are not familiar with a specific language will take online courses or consult team members with the appropriate knowledge. ● Split implementation up between team members experienced in the language and those not in order to capitalize on our team's strengths.
Personnel disability/ illness	A team member must cover for the disabled personnel, taking more time to complete the project.	0.1	7	0.7	● Use predefined backup team members who will take over jobs of the disabled personnel. ● Practice good hygiene and safety precautions.

X: Development Timeline

In order to complete the sections of the project in a timely fashion and by the given deadline, our team devised the following general Gantt chart:

Table 2. Development timeline arranged to conform to Santa Clara University's and our employer's (Community Technology Alliance) deadlines for our project.

Legend	Team	Aidan	James M.	James T.	Deadline

	Fall Weeks 1 - 2	Fall Week 3	Fall Weeks 4 - 9	Fall Week 10	Winter Weeks 1 - 2	Winter Week 3	Winter Week 4	Winter Weeks 5 - 9	Winter Week 10	Spring Week 1	Spring Weeks 2 - 9	Spring Week 10
Problem Statement	Team	Deadline										
Design Report		James M.	James M.	Deadline								
Design Review				James M.	James M.	Deadline						
Revised Design Report					James M.	James M.	Deadline					
Initial Operation System			Team	Team	James T.	Aidan	Team	Team	Deadline			
Server					James T.	Aidan	Team	Team	Deadline			
Smartphone App					James T.	Aidan	Team	Team	Deadline			
Communication Implementation					James T.	Aidan	Team	Team	Deadline			
Webpage					James T.	Aidan	Team	Team	Deadline			
Testing					Aidan		Team	Team	Team	Team	Team	
Final Presentation					James M.	James M.	James M.	Team	Team	Deadline		
Final Report					James M.	James M.	James M.	James M.	James M.	Team	Team	Deadline
Final System					James T.	Aidan	Team	Team	Team	Team	Team	Deadline

18

XI: Societal Issues

- **Ethical**
  - o Our main concern with this project was to make sure that our users were not exposed to any dangerous information that broke any country laws or ethical guidelines. With this in mind, we decided to use *New York Times* as our primary news source to retrieve news articles because of its reliability and general objective nature.
- **Social**
  - o As a team, we believe our system has a positive impact on the general public. We aim to make our users' lives easier and more informative. The system is easily adaptable into any country because of its simplicity and cheap cost.
- **Political**
  - o Although we plan to deploy this in several different countries, the political issues will need to be addressed by our successors on this project. Currently, we are unsure where we will be deploying this system.
- **Economic**
  - o This system allows users to save plenty of time in their daily lives as well as increase their access to information without the need for a smartphone or computer. Thus, this economically affects our users in a positive way.
- **Health and Safety**
  - o Our next main concern with this project was data security. Our users' information and our databases contents need to be kept safe in order to comply with our goals and our users' privacy. We accounted for this in the protocols we used in our implementation.
- **Manufacturability**
  - o Several outside companies have donated the specific hardware used for this project, which ensures plenty of our devices to be deployed easily. We also found it easy to duplicate our code from one device to another.
- **Sustainability**
  - o Our system runs wholly on solar energy harnessed by the pack itself, making the system as a whole completely sustainable.
- **Environmental Impact**
  - o As stated above, our project has a neutral impact on the environment because of its use of solar technology, providing power to the database without outside sources.
- **Usability**
  - o Due to our constraint to make this device work through SMS enabled devices and a user-friendly administrative web-application, we decided to make our designs as

minimalistic as possible. This created a highly usable product that can be understood in a very small amount of time.

- **Lifelong Learning**
  - o This project, in particular, enabled us to explore new coding languages and techniques now primarily used in today's world. For example, although one of our team members had extensive experience in JavaScript, he did not altogether have a concise understanding of AngularJS. However, through the lifelong learning process, we developed a highly thorough understanding of this addition to JavaScript, making it far easier to implement our project.

- **Compassion**
  - o One of the central themes of this project is to supply educational content to those in need of understanding of the outside world, primarily those citizens living in third world countries. It is important to understand that education is one of the few tools that help bring impoverished individuals out of there economic struggles and herein lies Beacon Pack's greatest strength. Through education, the Beacon Pack enables individuals to be more aware of the outside world and the effects it has on their daily lives.

XII: Conclusion

The Beacon Pack system is an excellent way to advance educational content in developing countries due to its portability, ease of use, and all around cost effective strategy. The ability to access *New York Times* articles via SMS is nothing short of groundbreaking because it makes our users' lives undoubtedly more simple and efficient. Using the Raspberry Pi as our primary computer system creates a sustainable system not only capable of being powered by a solar pack, but also powerful enough to provide hundreds of users with educational content on the go.

Our project offers innumerable advantages to traditional methods of communication. As our system connects to a cellular service signal, we have managed to inform our users on the go without any reliance on other sources other than their own simplistic cellular device. It also provides the necessary security a user might be concerned with as well as adheres to the many scenarios in which ethical issues might be a factor. Since our project is centered on mobility, the traveler's pack and Raspberry Pi provides our trekkers with a lightweight, compact pack that can be attached to any existing backpack for long journeys. Altogether, our system creates an enjoyable experience for both administrators and users through our simplified UI design.

Although our project provides countless benefits to the outside world, there are some distinct disadvantages that come with this system. As of yet, we have found it difficult to shift between the many languages that might be encountered in the system's deployment in several different countries. We have also been unable to integrate the ability for users to send pictures to the administrator in order to communicate what they would like to see in future content. This, we believe, is essential for future updates of this system because it allows a breakdown of the language barrier that might exist in several countries encountered. However, most of the disadvantages the system currently has are extraneous features and thus have little impact on the overall experience of the essential functionalities.

The future of this project is kept alive within the non-profit Community Technology Alliance, who will be hiring programmers to integrate this system into several different countries. As such, we have designed it with the crucial fact that other programmers will need to modify our system in the future. Many differences may need to be implemented from country to country depending on the political nature, ethical issues, and overall data security that is required for the region. However, we are confident that our efforts have made this process easy for our successors.

Overall, this system provides the necessary basis for educational growth that so many countries so desperately need in today's world. Thankfully, due to our cost effective design, careful implementation, and CTA's efforts to bring this into reality, we have high hopes for the future of this system and the positive impact it will have on the world around us.

Section XIII: Bibliography

[1] Abrahamsson, P., Helmer, S., Phaphoom, N., Nicolodi, L., Preda, N., Miori, L., Angriman, M., Rikkila, J., Wang X., Hamily, K., & Bugoloni, S. (2013). Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment. IEEE 5th International Conference, 2, 170-175.

[2] Chigona, Wallace, Darry Beukes, Junaid Vally, and Maureen Tanner. "Can Mobile Internet Help Alleviate Social Exclusion in Developing Countries?" The Electronic Journal of Information Systems in Developing Countries. 2009. Web. 27 Oct. 2014.

[3] Donner, Jonathan, and Shikoh Gitau. "New Paths: Exploring Mobile-Only and Mobile-Primary Internet Use in South Africa." World Wide Web Consortium. W3C, 1 Apr. 2009. Web. 27 Oct. 2014.

[4] Gitau, Shikoh. "After Access – Challenges Facing Mobile-only Internet Users in the Developing World." CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2010): 2603-606. ACM. Web. 21 Oct. 2014.

[5] Meetoo-Appavoo, A., Chutoo, A., Appavoo, P., & Durgahee, B. (2013). Open Source ICT Framework for Developing Countries. IEEE Technology & Society Magazine, 32(3), 39-47.

[6] Moturi, Christopher and Charles, Nyota. "Filling The Digital Divide In Rural Connectivity: Case For Last Mile Mobile Broadband Subs." International Journal Of Computing & ICT Research 6.2 (2012): 46-63. Applied Science & Technology Source. Web. 21 Oct. 2014.

[7] Olla, P., & Choudrie, J. (2014). Mobile technology utilization for social development in developing countries: An ethnographic futures research study. Information Systems Frontiers, 16(3), 369-382.

[8] Park, E., Nojun Kwak, N., Kyu Lee, S., Kim, J., & Kim, H. (2011). Provisioning QoS for WiFi-enabled Portable Devices in Home Networks. KSII Transactions On Internet & Information Systems, 5(4), 720-740.

[9] Unknown, "Africa Increases Its Internet Usage." Communications Of The ACM 55.4 (2012): 12. Applied Science & Technology Source. Web. 21 Oct. 2014.

[10] Younis, M., Sardesai, A., & Yesha, Y. (2007). Optimal dynamic transport selection for wireless portable devices. Communications & Mobile Computing, 7(1), 9-21.

Appendix A: User Manual

We have listed below the instructions for using our system in order to provide the smoothest experience for the user. In this manual, we provide several iPhone screenshots that enhance the visibility of the user's experience.

Access Application: If you have access to an SMS-text capable, cellular device, then you can connect via the Beacon Pack by texting the following number: 801-430-9661. Any text message input will allow access to the system.

Main Menu: Upon texting the system, you will see the following menu:
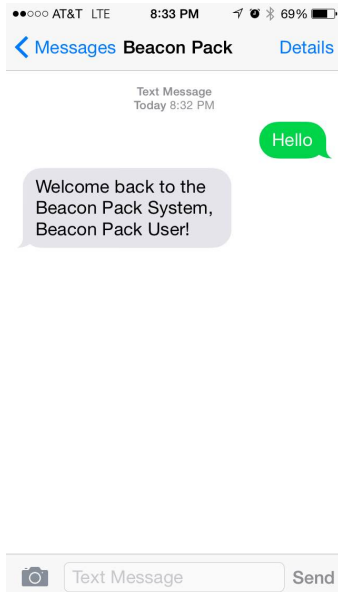
The main menu has the following options:
- Send "List" to request the database of informational content from the system.
- Press "Suggest XXXX" to suggest informational content to the admin that you would like the database to display in the future, where "XXXX" is what you'd like to suggest.
- Send "HELP" to retrieve any help you might need in navigating the system.

Request Content: The request content menu will list the headlines of informational content that you can access in greater detail. Simply text the system the number corresponding to each headline you select. For example, you will see the following menu located on the next page:
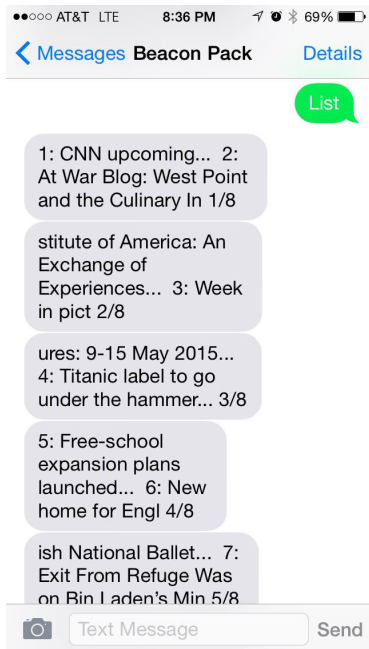
The system also provides several different pages of contents. In order to access the next or previous page of listed content, simply text "Next" or "Previous" to the system as such:
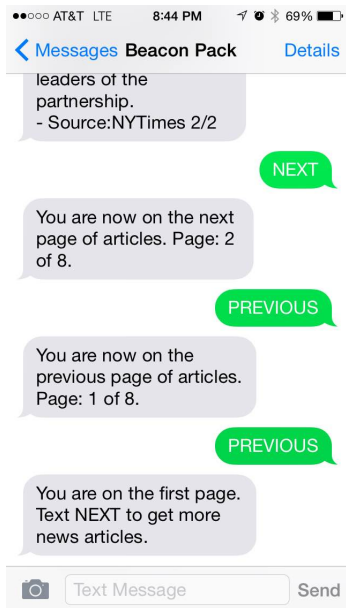
In order to access "2: At War Blog: West Point…." in greater detail, text 2 to the system. You will then see the following message (located on the next page):
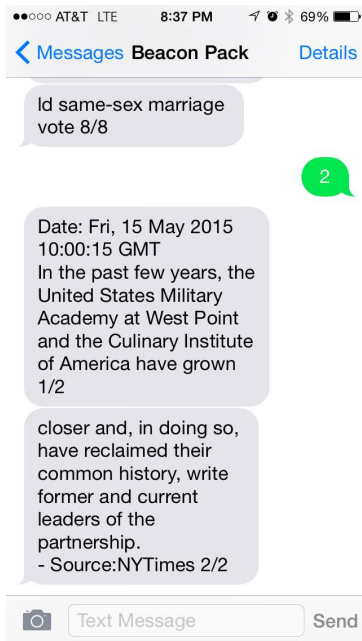
Suggest Content: In order to suggest content, simply text "Suggest XXXX". For example, you can input the following:

The admin will be able to see your suggestion in his admin app, and will make an update accordingly.

Exit System: To exit the system, simply stop texting the system. It will not text you further unless prompted to.

Appendix B: Admin Manual

We have listed below the instructions for using our system in order to provide the smoothest experience for the administrator. In this manual, we provide several iPhone screenshots that enhance the visibility of the admin's experience.

Download Application: If you have access to a cellular or WiFi connection, you can access the Beacon Pack app by going to "beaconpack1.engr.scu.edu" from your cellular or laptop device and following the instructions to use the application from your system.

Access Application: To access the application, simply open the web crawler from your phone and input your appropriate admin login information.

Main Menu: Upon logging into the system, you will either see the main menu, or a message asking you to connect to the cellular network that the Beacon Pack is on, which ensures access to the Beacon Pack system. If you are having issues connecting to the Beacon Pack, you may need to reconnect to your Wi-Fi network.
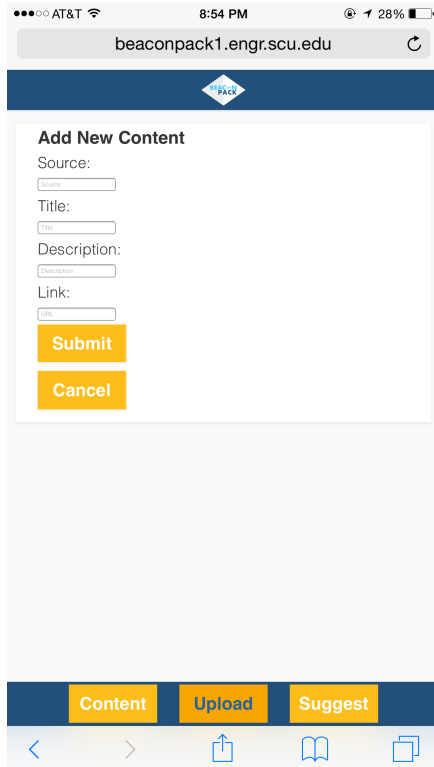
This is the main menu of the admin app:
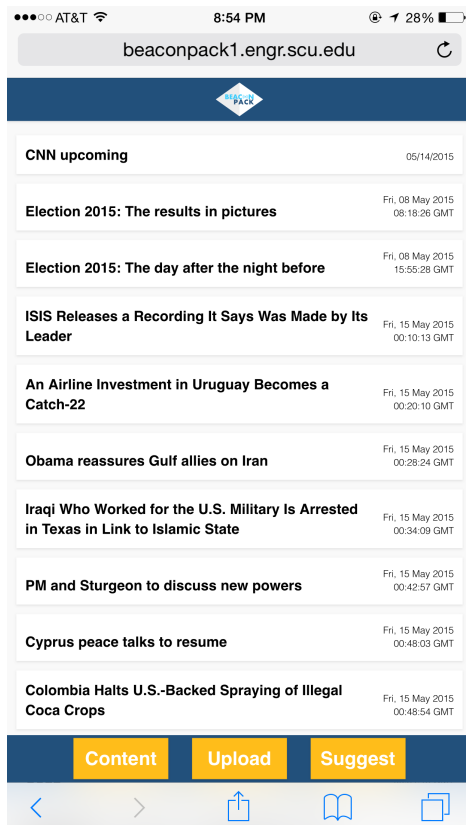
The main menu has the following options:
- Press "Upload" to upload content via your smartphone.
- Press "Content" to view the specific content that is currently in the database.
- Press "Suggest" to view a list of content that specific users are requesting.

Upload Content: Upon choosing this option, you will be given the opportunity to add a news article to the database. An example can be seen here:

Content: Upon choosing this option, you will be given a list of content, which is the main menu of the application.

You can then press on individual content in the list in order to see the content's information in the following way (image on next page):
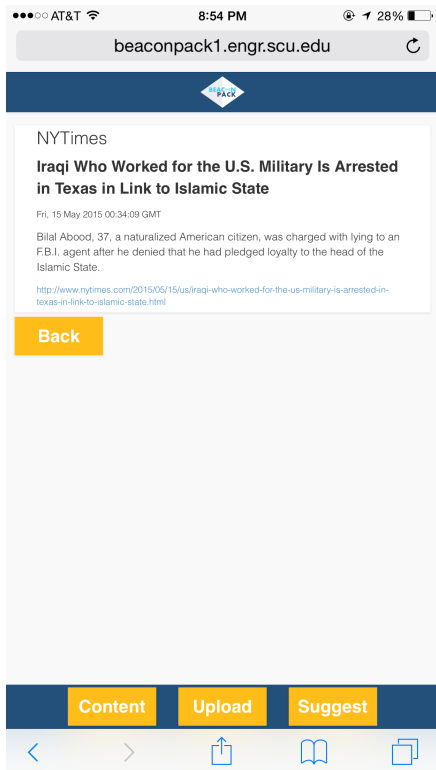
View Suggested Content: Upon choosing "View suggested content" you will see a list of content: