CSE 454 Capstone Project

Add to Your Tools

Project Proposal
Catalyst Tools
Files

Project Proposal

Problem

What is the problem we are trying to solve?

In developing countries, farming is one of main part of the economy. However, without good communication infrastructure such as Internet, there is no practical ways for farmers to seek for more potential buyers. Depending on one dominated supply company as only route to end consumers does not create a fair competition for farmers.

Team Members

Our team members.

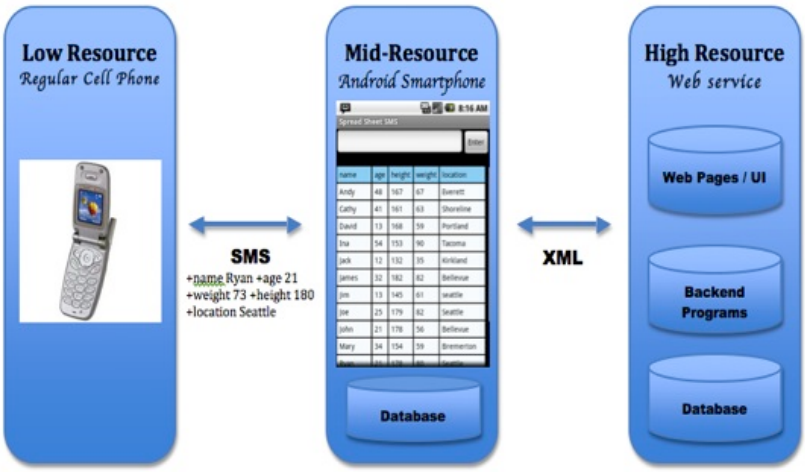
Name	Email	Concentration
YoonSung Hong	hys235@cs	Project ideas and management, Smartphone application
Matthew Lauren	matt.lauren@gmail.com	Server side scripting and user interface.
Steve Wang	tingyenuw@gmail.com	Backend programing

Description

A system for connecting suppliers and consumers in developing regions.

Goal
Our final goal will be creating a seamless and cohesive system that can provide services that helps stimulate local enterprises and improve health care deliveries in developing regions. However, with the time constrain in this class, we will aim to demonstrate one use-case and show a light of extensibility. Primarily, we will build web service that connects urban buyers to farmers at low-resource areas.

System Architecture



In this class, we are interested in building web service stack. Mid-resource and Low-resource solutions are built by YoonSung under supervision of professor Borriello(More can be found in http://code.google.com/p/opendatakit/).

Responsible Functions

1. Low Resource - Farmers must be able to send and receive SMS to/from Android nodes.
2. Mid Resource - Takes SMS messages sent by farmers and saves in local databse. The Android application serves as data visualization and managment tool. Also, it can simply be a router and data center between the low resource stack and the high resource stack. This stack takes care of the communication between the low and high resources using SMS and XML.

Announcements
No announcements

Send questions about this workspace to Yoon Sung Hong.

3. High Resource - Web services for buyers.

Components

- **Web Pages / UI**

The primary audience of web service will be buyers or service recipients. Thus, it's important to design a website that will be easy and clear to buyers. We find that Craiglist's location and service oriented model fits on our project. Most of buyers in developing regions are interested in local products and specific items or services. This model allows us to provide only essential information to users.

Just as Craigslit, users won't be able to make transaction on the website. However, we will keep track of information/message transactions between buyers and sellers. And, the messages will be private to each buyer and seller. This allows buyers to bid on an item and sellers to pick the best deal. Of course, item details such as prices and contact information will be shared to the public.

We will keep the user interface simple and look-and-feelings less exciting since our target domain is enterprises and we want to extend the service beyond just supply chain case.

Components:

- User accounts
- Supplier Portal (similar to home page)
- Buyer Portal (Possibly a single merged portal)
- Messaging between users (private and public)
- Service/Product listing by location
- Data searching (type/location/cost/supplier)
- Abstraction of written text (for translation purposes)

Extensions

- Module based, supply chain (primary), medical need, missionary need coordinaiton (public message that a certain group will be fulfilling a specific need).

- **Backend Programing**

The back-end will include the Java application code and will consist of two primary interfaces:

1) Interfacing with the Android smartphone: The Android smartphone will be the primary means of inputting supply data into the system, particularly to the database (see section 3 for more about the database). The back-end applications will provide an interface for Android smartphones to receive products from suppliers - which will be in XML format - and subsequently input the data into the database through the database interface. For example, if a supplier with an Android phone exports the addition of y quantity of product x by supplier s to the system, the back-end applications will read the XML data, parse it for the desired attributes, and update the database accordingly.

2) Interfacing with the Database: All of the raw data will be housed in a database. Examples of "raw data" can be supplier information, purchaser information, and actual product data. The back-end applications will interface with the database to read and write to the database. For example, adding product x by a supplier s (such as through an Android smartphone) will cause the back-end application to write a product product x tuple for supplier s. On the other hand, if someone has purchased product x from supplier s (such as through the web interface), the back-end application to remove the product x tuple from supplier s.

- **Databse**

MySql will be our database. We will have tables such as user list, messages and listings.

Milestone

Detailed schedule for the project.

- Oct 13 - proposal due

1. Further discussions.
2. Meeting with professor Weld.
3. Discuss about user interface and functionalities of the web service.

- Oct 29 - milestone 1 due

1. Setup Amazon EC3 development environment. Ruby on Rails/PHP, Java, and MySQL needs to be ready to go.
2. Design user interfaces and finalize on functionalities that will be supported.
3. Have a simple web pages running and XML parser ready on the server.

- Nov 17 - milestone 2 due

1. Declare XML rules and finish on pareser and enconder.
2. Able to show items updated by XML on the web pages.
3. Structure the overall web sites simllar to Craigslist. At this point, we must able to see the list of items in a structured web pages.
4. Finish on user system and message system between users. At this point, practically it's possible to make real transactions.

- Dec 3 - code complete; start experiments and writing

1. Review codes and polish UI.
2. Do unit tests and simulate the common tasks(Evaluation)
3. First draft of report completed.
4. Finalize the written reports and prepare for the presenatation.
5. If time allows, go beyond the 'farming supply chain' case.

- Dec 14 - Oral presentations, written reports, and code.

Evaluation

How do we evaluate this project?

Given the time and resources in the class, it's not practically possible to have real deployments. Many aspects such as security and privacy issues must be addressed carefully before it gets deployed. Rather, we will be adopting unit-test and case-test paradigm.

Unit-test involves testing individual components such as information retrieval system precison and performance.

We are interested in test and make sue the most common tasks work correctly. The common tasks includes:

1. Can farmers make a posting on the our web page?
2. Can buyers make transactions and and bid on items?
3. The messages between buyers and sellers kept tracked? Public and private messages handled correctly?

For the purpose of the class, we will create mock XML data and pretend they are comming from the Android application. However, if the time allows, we are interested in demonstrating the entire process at once. We would also like to perform load testing on the system and see the extent of the scale of the system.