



# AKN Public Services Project Software Design Document

**Prepared By:**

Ali Jaouhari

Kristina Wegener

11/29/2014  
Northern Kentucky University

## Contents

1. Introduction	
i. Purpose.....	3
ii. Scope.....	3
iii. Overview.....	3
2. Project Plan	
i. High Level Objectives.....	3
ii. Evaluation Criteria.....	3
3. System Architecture	
i. Architectural Design.....	4
ii. Decomposition Description.....	5
4. Data Design	
i. Data Dictionary.....	6
ii. GRASP Patterns.....	7
5. Human Interface Design	
i. Overview of User Interface.....	8
ii. Screen Images.....	9

## Attachments

- Project Vision
- Use Case Models
- Domain Model
- System Sequence Diagrams
- Test Plan Document
- Security Document
- Class Diagram

## 1. Introduction

---

### 1.1 Purpose

This design document contains all documentation for the final iteration of the AKN Public Services System, describing all of the classes and major interactions that are implemented. This document serves as the final technical deliverable for the project to the customer.

### 1.2 Scope

AKN Public Services System is a free, web-based system developed by NKU students Ali Jaouhari and Kristina Wegener to help individuals and organizations visualize open source data from the Open Source Cincy project.

### 1.3 Overview

This document will cover the final product of the AKN Public Services System. It will include all technical documentation either in this document or in its attachments.

## 2. Project Plan

---

### 2.1 High Level Objectives

We will be creating a web-based system that allows users to view open source data from the Open Source Cincy project. We are specifically focusing on City of Cincinnati Service Requests, Vacant Building Registrations and Neighborhood Information, including area.

This information is open source but it is available only in a .csv or .xlsx file, which prohibits many citizens from fully conceptualizing and mapping the data. Our goal is to create a visual representation of this data in a functional, easy to use web interface.

### 2.2 Evaluation Criteria

This project will be evaluated by Dr. James Walden, NKU faculty on the following criteria:

- Coding Style
- Traceability (design to code and requirements to design)
- Effectiveness of Design Models and Analysis
- Issue Tracking
- Correctness of Notation
- Clarity of Diagrams

- Professionalism
- Project Demonstrations

### 3. System Architecture

---

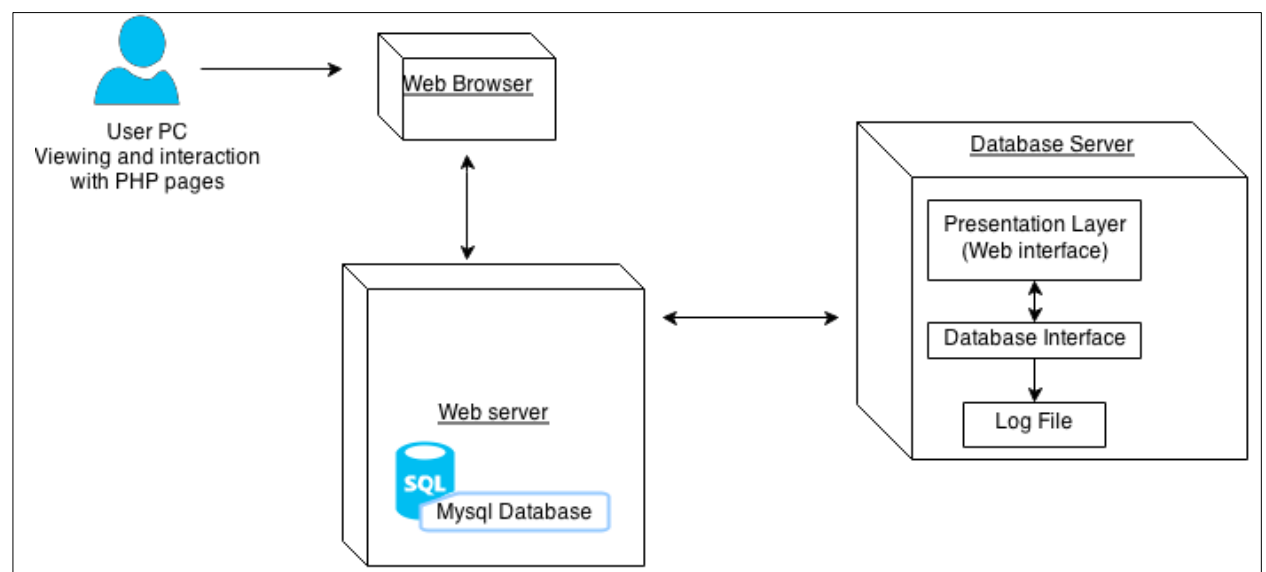
#### 3.1 Architectural Design

The AKN Public Services System consists of a website hosted on Amazon ec2 that is visible to all users who can connect to the internet. The system allows visitors to look up and access data stored on the server. The visitors will send requests for data to the server and receive requested data from the server and view information about the requested item.

Google Maps Javascript API will be utilized to provide mapping of addresses. The system will embed Google Maps into the page to allow for simple mapping of addresses returned in the user's search.

The majority of the system will be written in Object Oriented PHP. Using PHP, the site will communicate to a Mysql Database with information that will be populated from Open Data Cincy by a script.

Once a user requests information, the system will convert the user data entry into PHP requests and parse the PHP responses back into usable data which will be displayed to the user through their browser.



### 3.2 Decomposition Design

The visitor subsystem is divided up into a three layered architecture; it has a user interface, web browser, and visitor's device layer. Each layer has its own interface that other layers can use to interact with it.

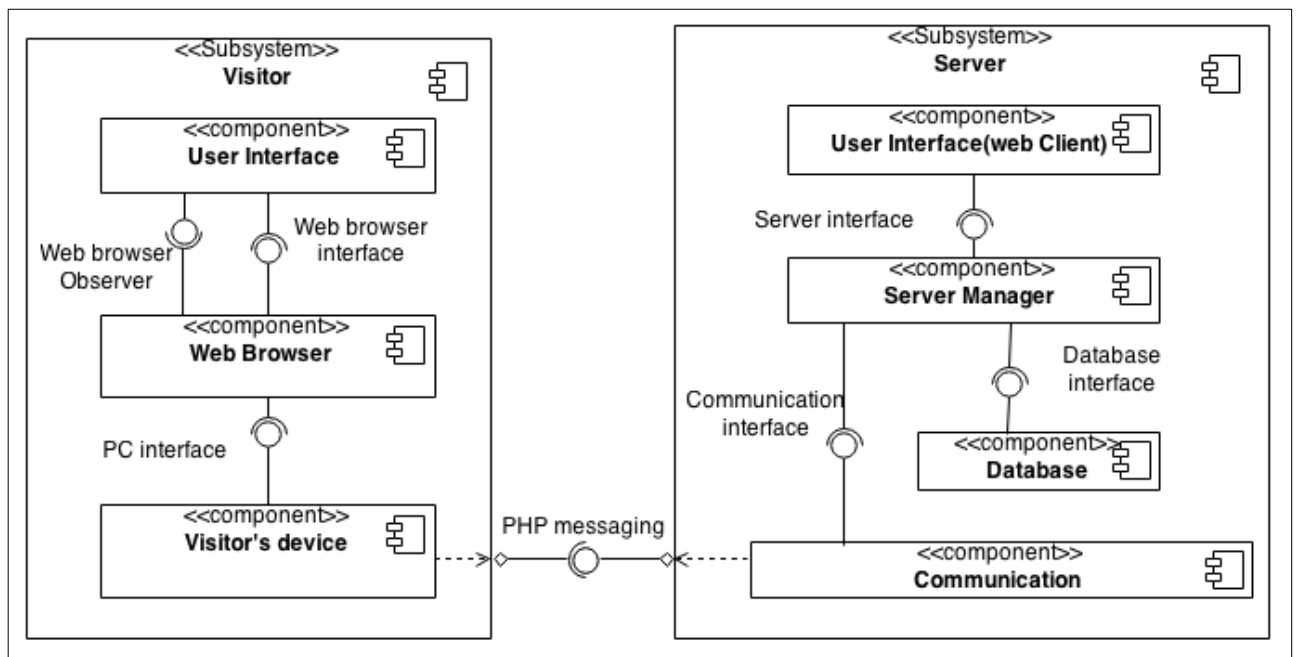
The user interface layer contains a web browser observer object and updates its data, using data from the observable application layer, via the observer pattern.

The web browser layer handles threads, logs, and converts messages from the user interface layer into PHP messages and sends them to the device layer.

The device layer handles the interactions with the hardware. Each layer is also a component that can be individually updated or replaced as long as the interface remains the same.

The server subsystem contains a web client component, server manager component, database component, and communication component. The web client component is a simple desktop application that connects to the server and follows the same rules for communication as the visitor system; it sends and receives PHP messages.

The server manager component handles the threads, message parsing, and database queries. The database component stores all of the data for the system. The communication component handles the ports and the sending and receiving of messages.



## 4. Data Design

---

### 4.1 Data Dictionary

The information domain of our system consists of the different types of information about Service Request, and Vacant Property records. The data stored in our central database is queried by the server which receives messages from the web-app. The data is organized by tables in a relational database.

While there are many classes and tables used in this project for the structure and functionality of the site, below are the classes that represent items that exist in the physical world and are most relevant to the user. For a full list of classes and variables, please see the Domain Model (attachment).

- SR (Service Request Record):
  - csr\_num: the number of the Service Request.
  - desc: the Service Request's written description.
  - rcvd\_dt: the date the Service Request was placed.
  - pln\_comp\_dt: the date the Service Request should be completed.
  - comp\_dt: the date the Service Request was completed.
  - stat\_id: an id number for the status of the Service Request.
  - priority\_id: an id number for the priority of the Service Request.
  - type\_id: an id number for the type of Service Request.
  - parcel: the parcel number for the location of the Service Request.
- SR Type (Service Request type):
  - type\_id: an id number for the type of the Service Request record.
  - type\_name: The type of the Service Requested.
- SR Priority (Service Request priority):
  - priority\_id: id number for the priority of the requested service.
  - priority\_name: The priority of the Service Requested.
- SR Status (Service Request Status)
  - stat\_id: an id number for the status of the Service Request record.
  - stat\_name: The status of the Service Request record.
- VP (Vacant Property Record):
  - stat\_id: an id number for the Vacant Property record
  - type\_id: an id number for the Vacant Property type.
  - parcel: the parcel number for the location of the Vacant Property.
- VP status (Vacant Property Status):
  - status\_id: an id number for status of the Vacant Property.
  - status\_name: status of the Vacant Property.
- VP Type (Vacant Property Status):
  - type\_id: an id number for the Vacant Property type.
  - type\_name: The type of Vacant Property.

- Address:
  - parcel: the parcel number of the property.
  - nhood\_id: an id number for the neighborhood .
  - st\_no: the number of the street.
  - st\_name: the name of the street.
  - st\_dir: the street direction (ex: North, South, East West. Optional)
  - xcoord: x coordinate for the address on the city of Cincinnati master map
  - ycoord: y coordinate for the address on the city of Cincinnati master map
- Nhood ( Neighborhood)
  - name: The name of the neighborhood
  - nhood\_id: an id number for the neighborhood.
  - area: the area for the neighborhood
  - perimeter: the perimeter of the neighborhood in miles.
  - acres: amount of acreage in a neighborhood.
- Administrator
  - login: The name of the administrator.
  - password: The password for the administrator login.

## 5.2 GRASP Patterns

### Information Expert

The Service Request's Address, Priority, Type and Status and the Vacant Property Record's Address, Type and Status are held as individual classes within either the Service Request or Vacant Property record. This is done to follow the Information Expert GRASP pattern. The Service Request or Vacant Property tables in the database only hold an ID number and the classes in the PHP code hold a class, rather than store the information with the name.

For Address, the sr and vp classes only have a parcel number. The address class is responsible for taking that parcel number and providing the sr and vp classes with information pertaining to that address. Extensive changes can be made to the address class without affecting either the sr or vp classes.

This is done so that changes can be made of any of these classes that will not affect the sr or vp class. It is the responsibility of these classes/objects to know how the ID number corresponds to other information about the Address, Priority, Type or Status.

### High Cohesion

Because the SRs and the VPs do not contain the Address information, there is a High Cohesion with strongly focused responsibilities. The address class is responsible for all the location information for a SR or VP record. The SR or VP record links back to address through a unique parcel number, but there is no knowledge of what this parcel number does. The address class uses the parcel number for mapping and to return a physical address for the SR or VP independent of the SR or VP functions.

Creator/Low Coupling

In this project, there is a general principle for the assignment of creation responsibilities. Because only the sr and vp classes can create the Priorities, Types and Statuses, there is low coupling. Sr and vp classes do not rely on the aforementioned classes and these classes can change without affecting the sr and vp.

## 6. Human Interface Design

---

### 5.1 Overview of the User Interface

The flow of the interface has been designed around the idea of searching for records, and displaying the returned. After displaying the information returned an option will be available to display the location of those records on a map. After selecting display on map, the system will load another page with map view.

The user can also choose to export the filtered results to an XML file opened in Xcel or other relevant program on the user's computer.

The user can also search for all records for a particular address or street. These results will also be displayed in list form.

Lastly, the user will be able to view information about a particular Neighborhood in Cincinnati, including its area, size and information pertaining to the Service Requests submitted for that Neighborhood.



## 5.2 Screen Images

### Main Page

[Service Requests](#)
[Vacant Properties](#)
[Neighborhoods](#)
[Address Search](#)





# Connecting People with Data

OpenDataCincinnati is a portal that provides access to various data sets, applications, and APIs related to the Greater Cincinnati region. OpenDataCincinnati is based on the idea that providing free and easy access to data information encourages better and more transparent government and a more engaged and knowledgeable citizenry. By connecting people with data, OpenDataCincinnati is hoping to encourage users to take the data and transform it into creative applications, projects, and visualizations that demonstrate the power that data can have in understanding and shaping our communities.

This website was created with Data from OpenDataCincinnati, primarily the Non-Emergency Service Requests by year and the Vacant Property Records, by Ali Jaouhari and Kristina Wegener for Northern Kentucky University's Software Engineering class, Fall 2014. All data is assumed accurate as given.

[Administrator Login Page](#)
2014 Ali & Kristi

### Service Requests – Unfiltered

[Service Requests](#)
[Vacant Properties](#)
[Neighborhoods](#)
[Address Search](#)



## Service Requests

Filter, Search and Map all Service Request Records


Status:  Type:  Type:  Neighborhood:

Filter to Display Results

[Administrator Login Page](#) 2014 Ali & Kristi

### Service Requests – Filtered

[Service Requests](#)
[Vacant Properties](#)
[Neighborhoods](#)
[Address Search](#)



## Service Requests

Filter, Search and Map all Service Request Records

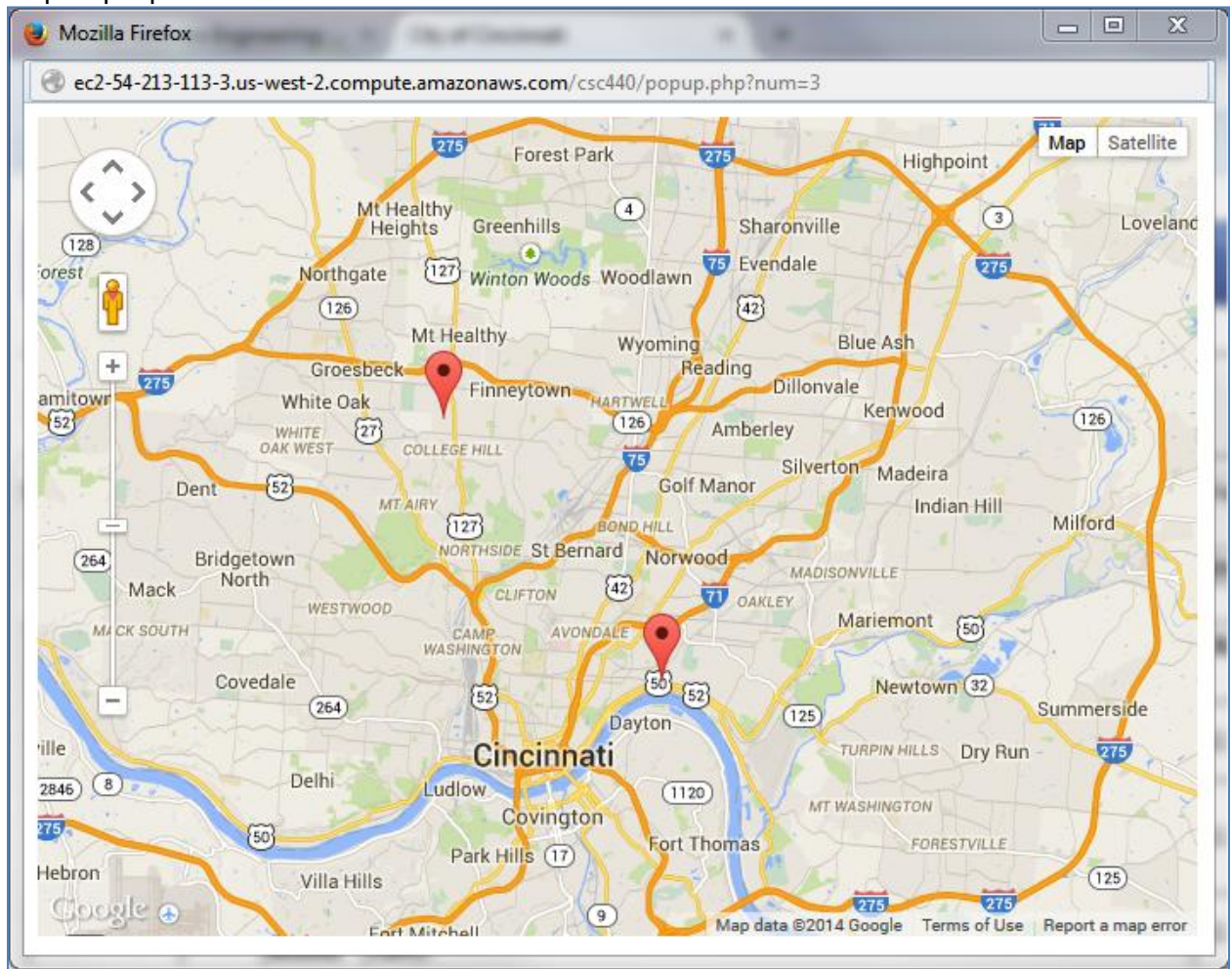
Status:  Type:  Type:  Neighborhood:

How many Addresses would you like to map:

CSR No.	Status	Type	Description	Received Dt.	Priority	Neighborhood	Address	Planned Comp.	Completed
SR13085222	NEW	Dead animal	RABBIT	11/26/2013	STANDARD	College Hill	1810 HARBESON AV	11/27/2013	
SR13085235	NEW	Dead animal	dead deer on Columbia Pkwy between William H Taft & Delta- currently on double yellow line.11/27/2013 1:13 AM/DCOTTRELL - The deer is now on curb side in bouind close to Delta There is also 2 cones to mark it.	11/27/2013	STANDARD	East End	1946 WILLIAM H TAFT RD	11/28/2013	
SR13085493	NEW	Dead animal	There is a dead possum in the grass area in front of our home (next to the curb).	11/29/2013	STANDARD	Roselawn	7636 GREENLAND PL	11/30/2013	

[Administrator Login Page](#) 2014 Ali & Kristi

## Map Pop-Up



## Vacant Properties – Unfiltered

[Service Requests](#) [Vacant Properties](#) [Neighborhoods](#) [Address Search](#)



---

## Vacant Properties

Filter, Search and Map all Vacant Property Records


Status: All Statuses Type: All Types [Filter](#)

Filter to Display Results

[Administrator Login Page](#)2014 Ali & Kristi

## Vacant Properties – Filtered

[Service Requests](#)
[Vacant Properties](#)
[Neighborhoods](#)
[Address Search](#)



---

## Vacant Properties

Filter, Search and Map all Vacant Property Records

Status: All Statuses Type: All Types [Filter](#)


[Export to Excel](#)
[Map Addresses](#)

Address	Status	Type
3511 BEVIS AV	VACATE THE BUILDING	ORDERS
1214 CHAPEL ST	VACATE THE BUILDING	ORDERS
3637 READING RD	VACATE THE BUILDING	ORDERS
1547 CORVALLIS AV	VACATE THE BUILDING	ORDERS
5211 GRAFTON AV	VACATE THE BUILDING	ORDERS
1735 ELM ST	VACATE THE BUILDING	ORDERS
4449 HILLSIDE AV	VACATE THE BUILDING	ORDERS
6671 PARKLAND AV	VACATE THE BUILDING	ORDERS
6665 PARKLAND AV	VACATE THE BUILDING	ORDERS
1104 CARSON AV	VACATE THE BUILDING	ORDERS
765 WELLS ST	VACATE THE BUILDING	ORDERS
1800 DENHAM ST	VACATE THE BUILDING	ORDERS
2623 KIPLING AV	VACATE THE BUILDING	ORDERS

[Administrator Login Page](#)
2014 Ali & Kristi

## Neighborhood Statistics

[Service Requests](#)
[Vacant Properties](#)
[Neighborhoods](#)
[Address Search](#)



---

## Neighborhoods

Select a Neighborhood and see a Record Overview

Neighborhood: Choose A Neighborhood [Filter](#)

### Fay Apartments

Perimeter: 2.97 Miles

Area: 0.41 Square Miles

Average: 261.70 Acres

This Neighborhood has 29 Service Requests

That is 0.06% of all Service Requests

5 Addresses in this Neighborhood have multiple Service Requests

3764 PRESIDENT DR had the most at 13 Service Requests

Number of Service Requests By Type:

Type	Count	Percentage
Annual - c annual insp:	5	17.24%
Dead animal:	1	3.45%
Light loader infested collectn:	1	3.45%
Building residential:	3	10.34%
Unsanitary living conditions:	1	3.45%
Roaches - bldg or apartment:	6	20.69%
Trash request for collection:	1	3.45%
Sewage surfacing priv prop:	1	3.45%
Sign down/missing:	1	3.45%
Mold building or apartment:	3	10.34%
Sign down/missing stop sign:	1	3.45%
Mice building has mice:	2	6.90%
Grass cutting city inspection:	3	10.34%

[Administrator Login Page](#)
2014 Ali & Kristi

## Address Search – Pre-Search


Service Requests

Vacant Properties

Neighborhoods

Address Search

city of CINCINNATI



---

Address Search

Search all Service Request Records and Vacant Property Records for an address

Street Number:

Street Direction:

Street Name:

Filter

You must enter at least a street name to continue.

Administrator Login Page

2014 Ali & Kristi

## Address Search – With Results

Service Requests

Vacant Properties

Neighborhoods

Address Search

city of CINCINNATI



---

Address Search

Search all Service Request Records and Vacant Property Records for an address

Street Number:

Street Direction:

Street Name:

Filter

Vacant Property Records

Address	Status	Type
28 W 13TH ST	KEEP THE BUILDING VACANT	ORDERS
20 W 13TH ST	KEEP THE BUILDING VACANT	ORDERS

Service Request Records

CSR No.	Status	Type	Description	Received Dt.	Priority	Neighborhood	Address	Planned Comp.	Completed
SR11027732	CLOSED	Default parks	06/10/2011 2:48 PM/DTHOMASBENCHES FROM THE PARK 6/10/2011 2:46 PM/DCOTTRELLplease remove benches and place in a park 05/31/2011 8:10 AM/CALLENThere is a small greenspace on the NE corner of 13th and Republic that has become a living space for the homeless	05/31/2011	STANDARD	Over-the-Rhine	18 W 13TH ST	07/10/2011	06/13/2011
SR11063711	CLOSED	Graffiti removal	This building is owned by 3CDC but i am planning to develop into condos early next year.	12/29/2011	STANDARD	Over-the-Rhine	18 W 13TH ST	01/28/2012	01/12/2012
SR11063785	INPROGRS	Recycling new 96 gallon cart	Request entered through the Web. Refer to Intake Questions for further description.	12/30/2011	STANDARD	Over-the-Rhine	24 W 13TH ST	01/9/2012	

Administrator Login Page

2014 Ali & Kristi

## Administrator Login Page

[Service Requests](#) [Vacant Properties](#) [Neighborhoods](#) [Address Search](#)




Email or Username   
Password

Administrator Login Page 2014 Ali & Kristi

## Administrator Functions Page

[Service Requests](#) [Vacant Properties](#) [Neighborhoods](#) [Address Search](#)



**Hello Kristina Wegener!**

Administrator Functions

[Add Administrator](#)  
[Download Service Request and Vacant Property Files From ODC](#)  
[Import 2008 SRs From Downloaded File](#)  
[Import 2009 SRs From Downloaded File](#)  
[Import 2010 SRs From Downloaded File](#)  
[Import 2011 SRs From Downloaded File](#)  
[Import 2012 SRs From Downloaded File](#)  
[Import 2013 SRs From Downloaded File](#)  
[Import Vacant Property Records From Downloaded File](#)

Name	Username	E-mail	
admin	admin	admin@website.com	<input type="button" value="Delete Admin"/>
ali jaouhari	jaouharia1	jaouharia1@nku.edu	<input type="button" value="Delete Admin"/>
Kristina Wegener	kristi	kristi@satcen.com	
Robert Oprisko	rob	roprisko@rob.com	<input type="button" value="Delete Admin"/>
Test Admin	dummyadmin	a.slkfnae;lfv	<input type="button" value="Delete Admin"/>

[LOGOUT](#)

Administrator Login Page 2014 Ali & Kristi