**Chicago Crime Alert Application**

**Design and Functional Specification**

**Project Requirements and Architecture**

11/23/2019

Written by Jack, Vismark

# Owners and List of Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Email** | **Phone** | **Role** |

Jack Broncato [jbroncato@luc.edu](mailto:jbroncato@luc.edu) 331-6425727 Developer

Vismark Juarez. [vjuarez1@luc.edu](mailto:vjuarez1@luc.edu) 773-710-6749 Developer

# Signoffs

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Name** | **Date** | **Signature** |

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Reason for change(s)** | **Author(s)** |

**Table of Contents**

[Owners and List of Contacts 2](#_Toc246152908)

[Signoffs 2](#_Toc246152909)

[Revision History 2](#_Toc246152910)

[About this Document 4](#_Toc246152911)

[Purpose and Contents 4](#_Toc246152912)

[What this specification does 4](#_Toc246152913)

[What this specification does not do 4](#_Toc246152914)

[A ‘Living Document’ 4](#_Toc246152915)

[Assumptions 4](#_Toc246152916)

[Questions and Comments 4](#_Toc246152917)

[Software Overview 5](#_Toc246152918)

[Product description 5](#_Toc246152919)

[Product functional capabilities 5](#_Toc246152920)

[User characteristics 5](#_Toc246152921)

[User operations and practices 5](#_Toc246152922)

[General constraints 5](#_Toc246152923)

[Assumptions 5](#_Toc246152924)

[Other software 5](#_Toc246152925)

[System Architecture 6](#_Toc246152926)

[Description 6](#_Toc246152927)

[Logical View 7](#_Toc246152928)

[Physical View 8](#_Toc246152929)

[Class Diagram 8](#_Toc246152930)

[Sequence Diagram 8](#_Toc246152931)

[Database Diagram 9](#_Toc246152932)

[Specific Function Description 10](#_Toc246152933)

[1. Product 10](#_Toc246152934)

[Description 10](#_Toc246152935)

[User Interface 10](#_Toc246152936)

[Inputs 10](#_Toc246152937)

[Processing 10](#_Toc246152938)

[Reports 11](#_Toc246152939)

[Report 1: 11](#_Toc246152940)

[Actions 12](#_Toc246152941)

[1. Select 12](#_Toc246152942)

[2. Update 12](#_Toc246152943)

[3. Delete 12](#_Toc246152944)

[4. Calculate 12](#_Toc246152945)

[Output Screen: 13](#_Toc246152946)

[Security 14](#_Toc246152947)

# About this Document

## Purpose and Contents

### What this specification does

### What this specification does not do

This is not a project plan. It is a guide for system architecture and development, not for phasing, timelines or deliverables.

### A ‘Living Document’

Finally, this specification will change, continuously, as the project proceeds. We will add details and edit existing information as the database structure and functionality evolve in the course of the project.

### Assumptions

### Questions and Comments

If you have questions or comments regarding this document, contact:

* Vismark Juarez ([vjuarez1@luc.edu](mailto:vjuarez1@luc.edu))
* Jack Broncato ([jbroncato@luc.edu](mailto:jbroncato@luc.edu))

|  |  |
| --- | --- |
|  |  |

# Software Overview

## Product description

The Chicago Crime Application is exactly what it sounds like, it supplies a user with crime statistics and info through a user friendly interface.

## Product functional capabilities

The application can: give up to date crime statistics, alert a users mobile phone of important update on either local or relevant crimes, and be used as a browser of curated news articles.

## User characteristics

Users would be anyone in the Chicago area who are concerned about crime and how it may affect their daily lives, or anyone who is curious about the subject.

## User operations and practices

Users can set up alerts to their mobile phone for relevant news, and query an API for crime info, having it returned in a user readable graph.

## General constraints

Time, Resources(we lost a team member we believe), and Technological, as one branch of the project had to be abandoned because of a lack of chart capability.

## Assumptions

The Crime API will not fail, and neither will the Twilio, also that our users will be using windows machines.

## Other software

No other software was included in this project besides what was used to write the code, that being visual studio.

# System Architecture

## Description

A close up of a logo

Description automatically generated

A close up of text on a white background

Description automatically generated

## Logical View

**Figure 1: High Level Logical Architecture**

The architecture is divided into three logical layers to better meet the requirements of reusability, ease of code maintenance, and scalability.

* **Web Layer/Presentation**

Winforms application to allow users to easily navigate/utilize the application

* **Business Logic Layer (BLL)**

Twilio/Chicago Crime API, handle processing of user data and fetching of relevant crime data for display on the presentation layer.

* **Data Access Layer**

MongoDB database for holding user info safely and in mass

## Physical View

N/A – The application does not require the deployment of any physical components.

## A screenshot of a cell phone Description automatically generatedClass Diagram

## 

## Sequence Diagram

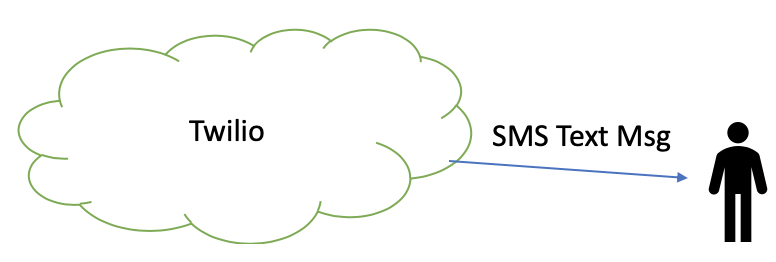
1. User enters their contact information into the application’s Alert Signup Page. The data is then stored into a MongoDB Atlas datastore:

A close up of a logo

Description automatically generated

## 

## Upon successful registration, the user will receive an SMS text message confirming that their information has been successfully stored:



1. The user will then receive daily alerts when crimes are reported in their specified area.

A close up of a device

Description automatically generated

## Database Diagram

A close up of text on a white background

Description automatically generated

# Specific Function Description

## 1. Product

### Description

The application gives up to date crime statistics, alerts a users mobile phone of important update on either local or relevant crimes, and is used as a browser of curated news articles.

### User Interface

User interface is comprised of a WinForms application that handles all user data input, displays graphs, news, etc. It is the main and only user interface at this time, and all features are present in it.

### Inputs

Mouse and keyboard, clicking a top menu for navigation through the application. There are also several textboxes used for user data input, as well as selection of data from the crime statistics API.

### Processing

Processing includes fetching data from APIs and displaying it in the graph, storing user data in a database, and notifications to a user’s mobile phone through the Twilio API.

# Reports

## Report 1:

WPF problem: WPF was chosen at an early point in development as a potential front end, but as it has come to be seen there is no easy way to implement a graph in WPF, at least not one that time will allow to be implemented. Because of this shortness of time and Graph difficulty development on a WPF front end has halted, and from now on development will solely be on WinForms, where a graph is already implemented and most functionality is already present, thanks to Vismark.

Actions

## 1. Select: After the user has added their contact information into the Chicago Crime Alert Application database, their information will be selected when sending SMS alerts to the user. The application will also select certain data from the Chicago Crime Portal API in order to deliver it to the user (via a graphical UI).

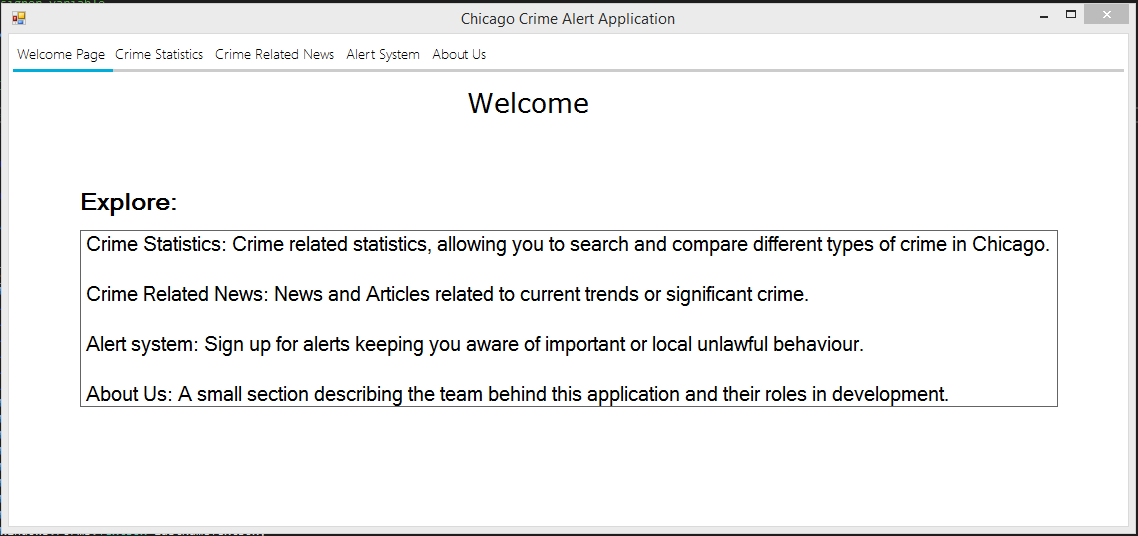
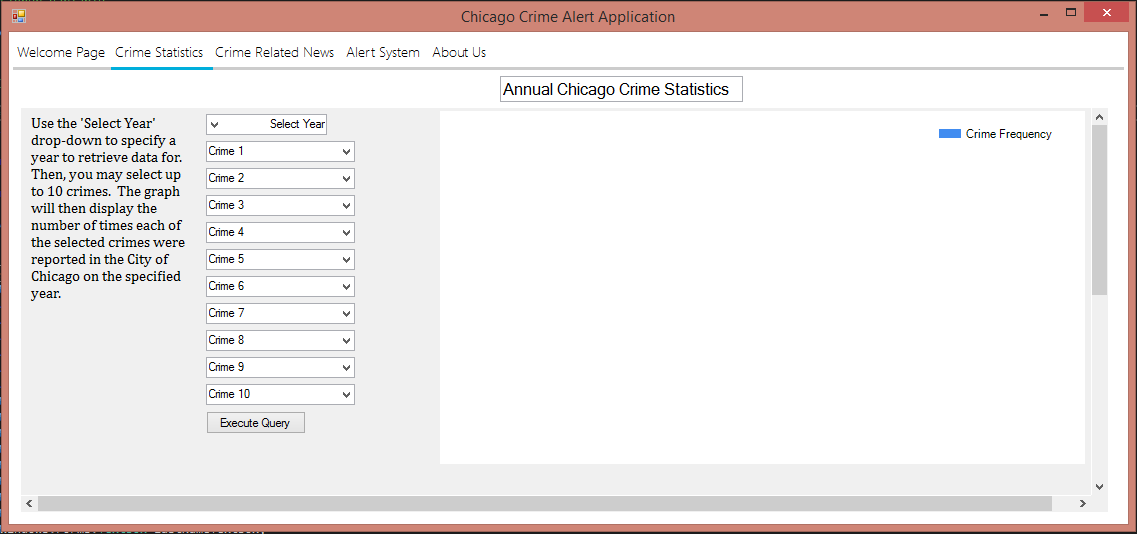
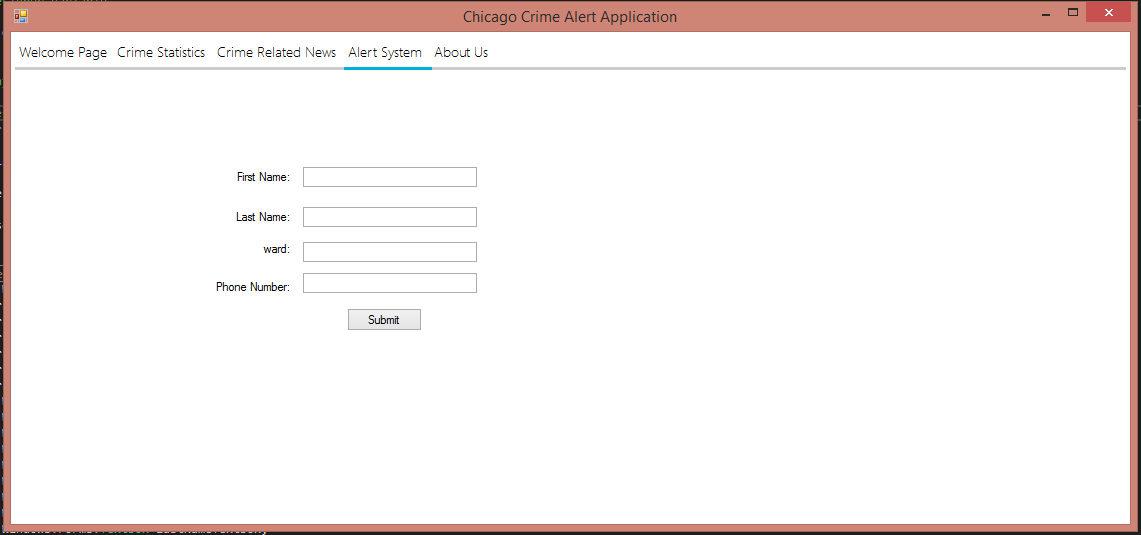
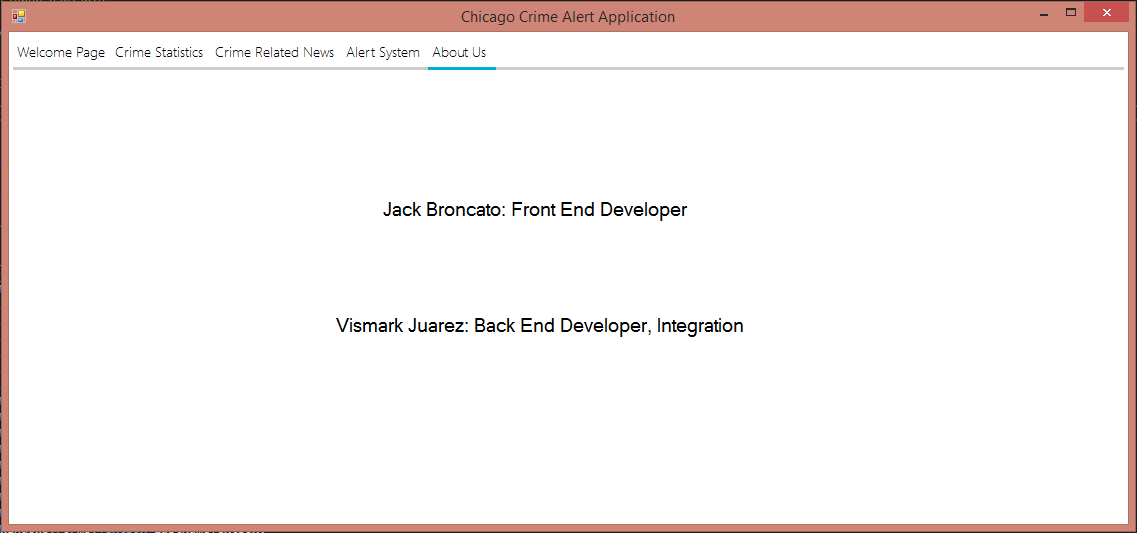
## 2. Update: A user’s information will not need to be updated, nor will the Chicago Crime Alert Application update any data via a third-party API.

## 3. Delete: No data need be deleted. This application will remove all user data from its databases manually.

## 4. Calculate: Statistics will be calculated after the applications fetches relevant information and has all of the data it requires. These statistics include: number of times a crime has occurred in a certain ward, the number of times a specific crime has occurred in a specific year, etc.

# Output Screen:

Welcome Screen:

Crime Statistics:Alerts:About Us:

# Security

**Figure 11: Proposed security architecture.**

Access to the mongoDB database is only through a privately shared credentials file, thus keeping user data safe against anyone without proper authorization.