# 

Final Deliverable

CIS 4911 Senior Project

## BOLO Flyer Creator

Spring 2015

Team Members:

Danae I. Perez Tillan

Icxe Vidal

Copyright (C) Mobile Clinic-Electronic Medical Records

Permission is granted to copy, distribute and/or modify this document

under the terms of the GNU Free Documentation License, Version 1.3

or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

# Executive Summary

BOLOS, or Be On the Look Outs, is a law enforcement term that describes the way information is shared amongst police officers and between police agencies. A BOLO is distributed in several ways: over the police radio, via Computer Aided Dispatch (CAD) message or as BOLO flyer. When a crime is committed and the responding patrol officer completes a police report with the details. The report is submitted for approval and is validated by the records function of the department. The report is then distributed to the investigations division for review. The investigations supervisor reads the report and assigns it to a detective. The detective or analyst creates the flyer and sends it via email, usually to one or a few representatives at other departments, not to the patrol officers. The recipient would then have to forward it to all personnel within his or her agency. This process can take from a few days to a few weeks depending on the agency.

This is where the idea for the BOLO Flyer Creator came from, the need of having a system that would allow police officers to share BOLOs in real time and automatically, at the moment a crime is being reported, with as many police agencies as possible.

[BOLO Flyer Creator](#h.ednvu44xhw51)

[Executive Summary](#h.gxfysl6mwl22)

[1. Introduction](#h.8ap07o1afz15)

[1.1. Problem definition](#h.71bxkrp5cwwx)

[1.2. Scope of system.](#h.6ree2cnr5ad)

[1.3. Over all development methodology.](#h.tb556wm02hjv)

[1.4. Definitions, acronyms, and abbreviations (at most one page).](#h.dhy47fpgmi6b)

[1.5. Overview of document](#h.nh8d2272ugpn)

[2. Feasibility Study](#h.d2b14q52qdc7)

[2.1. Description of current system. Identify limitations and constraints](#h.wtpcnm7xcrko)

[2.2. Description of alternative solutions considered.](#h.yrzvzfhnwfw)

[2.3. Recommendation with explanation of why the solution was selected.](#h.25ybd382giiv)

[3. Project Plan](#h.f85vg8v7uouf)

[3.1. Project Organization](#h.8qju4xxl9ate)

[3.1.1. Project Personnel](#h.f1d796davit2)

[3.1.2. Hardware and Software Resources](#h.l28ty5o48qcn)

[3.2. Identification of Tasks, Milestones and Deliverables](#h.mx5cgn8sxpnr)

[3.3. Cost of the Project](#h.tlletx2c17jc)

[4. System Requirements](#h.jmhficq3rmfa)

[4.1. Functional and Nonfunctional Requirements – similar to RD](#h.rjy9z7d0k9gn)

[4.2. Requirements Analysis - similar to RD](#h.7cauykrq9cd)

[4.2.1. Use case model](#h.z3yec3iu3ah)

[4.2.2. Static model](#h.nyn7o6y7xadp)

[4.2.3. Dynamic model](#h.unu03wcagn99)

[5. System Design (i.e., overall system design)](#h.pcpmljsocu10)

[5.1. Overview](#h.ggbl2ixaus4a)

[5.2. Subsystem Decomposition](#h.m6slq7ioeela)

[5.3. Hardware and Software Mapping](#h.98rd9ylm24ur)

[5.4. Persistent Data Management](#h.kjvuitozf5pz)

[5.5. Security/Privacy](#h.ne7kuym69zpm)

[6. Detailed Design](#h.s8o5dmejyje)

[6.1. Overview](#h.z3vney2nqe6o)

[6.2. Static model](#h.exgmti8iys19)

[6.3. Dynamic model](#h.qetm4ca4c9xv)

[6.4. Code Specification](#h.awzjo4u9pr80)

[7. System Validation](#h.obcduza2h25)

[7.1. System Tests](#h.69ch16rnrltd)

[7.2. Evaluation of Tests – evaluate how successful the tests were. Use a tabular form.](#h.pcbbwudic6c0)

[8. Glossary - define terms used in document, especially domain specific terms.](#h.g3rr4rpgp532)

[9. Appendix](#h.4hlni41sn69d)

[9.1. Appendix A - Project schedule (Gantt chart or PERT chart).](#h.rg9kq0ax7662)

[9.2. Appendix B – All use cases with nonfunctional requirements.](#h.9ekwa4fbk3am)

[9.3. Appendix C – User Interface designs.](#h.2mpj3qm4yfnx)

[9.4. Appendix D – Analysis models (static and dynamic)](#h.a6yaswu4108i)

[9.5. Appendix E – Design models (static and dynamic)](#h.6jgb5z7bs48h)

[9.6. Appendix F – Documented Class interfaces (code) and constraints.](#h.2m8rdy352prh)

[9.7. Appendix G – Documented code for test drivers and stubs.](#h.p7yjfwyljw5t)

[9.8. Appendix H – Diary of meeting and tasks for the entire semester.](#h.tkp818gdth5z)

# 1. Introduction

The Police Department Of the Pinecrest Village wants to develop an application that will assist first responder officers creating BOLO flyers when they arrive on the crime scene and allows them to upload them to a server; so that when a detective is assigned to the case, he is able to retrieve it from the system

## 1.1. Problem definition

A crime is committed and the responding patrol officer completes a police report with the details. The report is submitted for approval and is validated by the records function of the department. The report is then distributed to the investigations division for review. The investigations supervisor reads the report and assigns it to a detective.The detective or analyst creates the flyer and sends it via email, usually to one or a few representatives at other departments, not to the patrol officers. The recipient would then have to forward it to all personnel within his or her agency. This process can take from a few days to a few weeks depending on the agency.

## 1.2. Scope of system.

The BOLO flyer creator will provide police officers with a tool to create and distribute BOLO alerts. In this first phase The intended users of the system are police officers belonging to police agencies enrolled in it. The website will grant users different levels of functionality depending on the tier they belong to. All user will have access to the same basic functionalities like create a bolo and either submit it to the system or download it as a pdf document, they will be able to search for BOLOs created by other users, they will be able to edit and archive the BOLOs they authored and opt out of receiving email notifications (NOTE that the email notifications is a system function performed automatically when a new BOLO is created, users don’t actually send the emails themselves).

In addition to these functions, police supervisors can edit BOLOs created by any police officer in their agency, and have an option to mass archive BOLOs by a date range.

Only the site administrators can enroll new users and police agencies in the system and have access to the database and backend of the website

## 

## 1.3. Over all development methodology.

The software development methodology used in the creation of the system was Agile model. We wanted to emphasize adaptability in order to accommodate our product owner's changing needs, and be able to change directions, if needed, midway the development process.

## 1.4. Definitions, acronyms, and abbreviations (at most one page).

BOLO: Be On the Lock Out

## 1.5. Overview of document

Section 1 is the introduction to the BOLO Flyer Creator Project. It states the problem the project aims to solve, defines the methodology used during the development of the project and presents the rest of the document

Section 2 describes the feasibility analysis. It explains the limitations of the current system, alternative solutions considered and the reasons the proposed solution was chosen

Section 3 states the project plan. It describes how the project is organized, the human and material resources needed, and identifies milestones, tasks and deliverables.

Section 4 introduces the proposed system. It describes the functional and non functional system requirements

Section 5 covers the overall system system design. It explains the architecture used, each of the major subsystems implemented, mapps the software to hardware, states what kind of data needs to be stored and the way it’s going to be accomplished, and lastly privacy and security are discussed

Section 6 covers the detail class design. It describes the behaviour and structure of each subsystem and design patterns used. Next is a detailed description of the structure of each subsystem; followed by the class interfaces

Section 7 covers the validation of the system. It includes the test cases used on system and subsystem testing. For system testing each test has 2 sunny and 1 rainy cases

Section 8 is glossary of terms used throughout the document

Section 9 includes all the appendices: project schedule, all use cases, analysis and design models and class interfaces,

Section 10 includes references to other documents and sources

# 

# 

# 2. Feasibility Study

## 2.1. Description of current system. Identify limitations and constraints

BOLOS, or Be On the Look Outs, is a law enforcement term that describes the way information is shared amongst police officers and between police agencies. A BOLO is distributed in several ways:

*Over the police radio*: When an incident occurs the responding officer will “BOLO” pertinent

information over the police radio for other units to hear. Limitations: If you are not on duty or

missed the radio transmission you did not hear the information.

*Via Computer Aided Dispatch (CAD) message*: The officer will send a text only message via the CAD to other police units that are currently on duty. Limitations: Text only and units from

other police agencies do not get the message. The dispatcher would then have to contact via

phone or police radio the other agencies and relay the BOLO information.

*BOLO flyer*: A BOLO Flyer (Document) with pertinent information, to include pictures, is created, usually via MS WORD and then converted to PDF and distributed to other officers or agencies via Email or FAX. Limitations: First responding police officer does not usually have the capability or time to create such flyer. Flyers are commonly created by detectives or analysts several days after the incident was reported. The distribution is via email so there is no central,

searchable database. If you are not on the email distribution list, you don’t get the BOLO flyer.

## 2.2. Description of alternative solutions considered.

-A possible alternative solution for this system, is having a Google Drive folder shared among police officers where they could create BOLOs and view the BOLOs created by other officers

-Other solution is creating a master-email account where officers would send the BOLOs they create, and then they would be distributed to the rest of the police force from that master-email

## 2.3. Recommendation with explanation of why the solution was selected.

BOLO Flyer Creator was designed to streamline this process and give the first responding officers the ability to quickly, from the field, create a BOLO flyer and distribute it to all users. This can be done at the time the crime is reported. This application not only distributes the flyer but posts it to a central site that is searchable. All users have the ability to see via a graphical interface, all BOLOs posted as well as search and sort the BOLOs by a variety of different criteria. BOLO flyer creator also give the user the ability to attach a video link should the a video of the incident be available. This solution allows all registered users, regardless of whether they were on duty or not, the ability to view the flyers.

# 

# 

# 3. Project Plan

## 3.1. Project Organization

### 3.1.1. Project Personnel

Danae Perez Tillan and Icxe will be working on all the front-end website aspects of the application.

Danae Perez Tillan and Icxe Vidal will work jointly between the backend aspects of the application.

Danae Perez Tillan and Icxe Vidal will work jointly on the documentation of the application.

### 3.1.2. Hardware and Software Resources

Hardware

Processor: x86 / x64 2.0 GHz or faster

Memory: 4 GB or higher

Disk Space: 500 GB

Software:

Wordpress

PHP 5.1.6

HTML

MySQL 3.23.23 or greater

JavaScript

CSS

## 3.2. Identification of Tasks, Milestones and Deliverables

|  |  |  |
| --- | --- | --- |
| Task Name | Start | End |
| Login screen  Create Bolo Form  Set user permissions  Create user management tools | 2/2/15 | 2/13/15 |
| Search function  Navigation menu  Homepage thumbnails  Bolo flyer | 2/16/15 | 2/27/15 |
| Edit function  Add agency  Send notifications  Delete bolo | 3/2/15 | 3/13/15 |
| Purge option  Extended user groups  Mobile version  Full size image | 3/16/15 | 3/27/15 |
| Opt out of notifications  Save as pdf  Improved user management  Edit history tracking  Improved homepage | 3/30/15 | 4/10/15 |

## 3.3. Cost of the Project

The cost of this project depends on the server chosen and the data plan to go with it. There is no cost associated with the software, since all of it is free and open source

# 

# 

# 4. System Requirements

This chapter describes the system requirements of the BOLO Flyer Creator Project

## 4.1. Functional and Nonfunctional Requirements – similar to RD

Functional Requirements:

-The system shall allow registered users to log in(Bolo-001 Log in)

-The system shall logged in users to log out(Bolo-002-Log out)

-The system shall provide a tool for officers to edit a BOLO (Bolo-004-Edit Bolo)

-The system shall provide a tool for officers to create new BOLO (Bolo-005-Create Bolo)

-The system shall display the 24 more recent BOLOs in the homepage as a thumbnail preview(Bolo-006-Thumbnails)

-The system shall provide a tool for officers to create and download a private BOLO in pdf format (Bolo-007\_save as pdf)

-The system shall allow users to search BOLOs by any of their fields (Bolo-008\_Search)

-The system shall allow users to navigate thru all BOLOs from the homepage (Bolo-009-Next Button on Homepage)

-The system shall allow administrators to create, delete, promote and manage users (Bolo-013-User Management)

-The system shall provide a flier view of the BOLO (Bolo-014-Bolo flier view)

-The system shall provide a navigation menu that allows users access to the functionality they can use (Bolo-015-Navigation menu)

-The system shall allow administrators the ability to add new police agencies to the system (Bolo-016-User groups)

-The system shall send an email notification to all registered users whenever a new BOLO is created (Bolo-017-Email Notifications)

-The system shall allow users to purge BOLOs from the system (Bolo-018-Purge Bolos)

-The system shall allow users to archive BOLOs in the system (Bolo-019-Archive)

-The system shall upload a default image when users don't upload any when creating a BOLO (Bolo-020-No image available)

-The system shall allow users to view the image associated with a BOLO in full size (Bolo-021-View image full size)

-The system shall allow users to filter the thumbnails in the preview by the agency that created them (Bolo-022-Filter thumbnails)

-The system shall allow users to opt out of receiving email alerts when new BOLOs are created (Bolo-023-Opt out alerts)

-The system shall keep track of all changes history when BOLOs are edited (Bolo-024-Edit tracking)

Non-functional requirements:

Usability:

- The system shall require no training

- It should take no more than 2 minutes to create a BOLO

Reliability:

- The system shall inform its users of scheduled maintenance at least a week before

Performance:

- The system shall load any page in less than five seconds

- A message will only be presented once on the same page

Supportability:

- After the system is deployment, it should be to change

Implementation:

- Every page presented to the users should look the same in the supported browsers

Security:

- The system will authenticate each user, and depending on his credential, the system will provide the proper user rights, thus no user will access anything beyond his limits

## 4.2. Requirements Analysis - similar to RD

### 4.2.1. Use case model

In Appendix B, each use case is described. It identifies which user has access to each use case. It describes the relationship and interaction among the use cases and actors involved in the system. The users who are the actors are classified in Administrators, Tier2 (for police supervisors) and Tier1 (for the rest of police officers)

### 4.2.2. Static model

Appendix C shows the different subsystems the Model-View-Controller architecture will be split into and the classes that compose them

### 4.2.3. Dynamic model

Appendix D include the sequence diagrams creating during the system design

# 

# 

# 5. System Design (i.e., overall system design)

This section defines the architecture employed in the implementation of the project

## 5.1. Overview

Our application will use a server-client architecture and all of the implementation will be on the server side, where we will use a MVC architecture

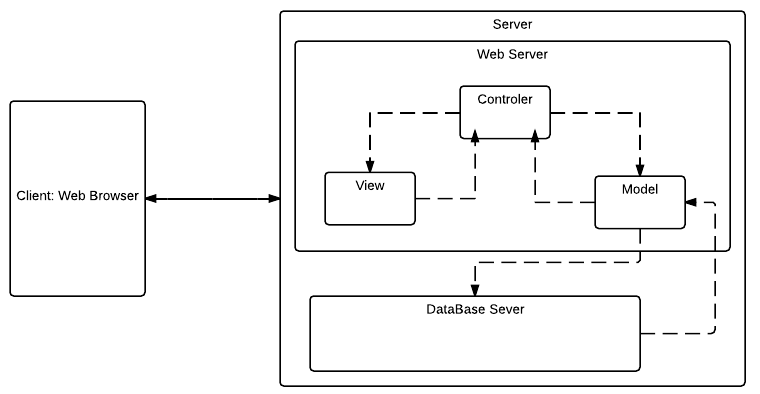
## 5.2. Subsystem Decomposition

The server subsystem will be implemented using a Model-View-Controller (MVC) architecture.

The Controller subsystem will handle the logic of the system and the interactions between the other Model and View subsystems. The Controller will receive the requests from the web browser, send that request for data to the Model subsystem, the data received from the Model is used to create the View which is presented in HTML via the web browser the user is using. In essence the Controller subsystem is responsible for behaving like an intermediary between the user and the data storage

The View subsystem contains and formats the information presented to the user. This information is received from the Controller subsystem and in turn sends user input to the Controller.

The Model subsystem handles the data storage, which will be done via a MySQL database

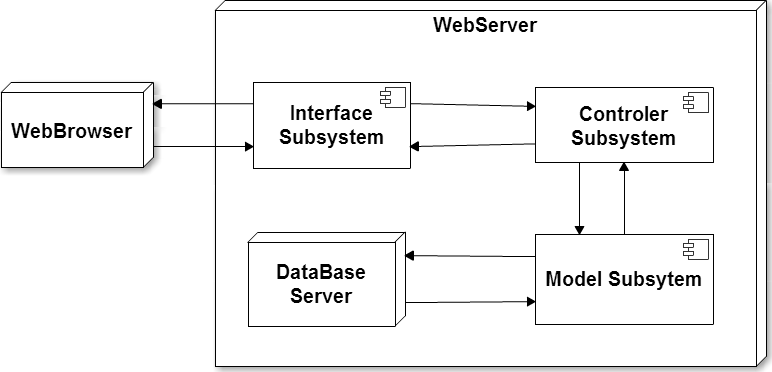


## 5.3. Hardware and Software Mapping

The BOLO Flier Creator uses two architectural patterns. On the top level there is a server-client architecture. On the Server side there is a Model View Controller (MVC) architecture with the core functionality.

The client is any modern web browser, which renders and displays the HTML code the server returns

The server is a computer running Windows, WampServer, MySQL server and PHP runtime



## 5.4. Persistent Data Management

The system will store two different kinds of data: user profile data and the BOLOs in a relational database system powered by MySQL

Furthermore, BOLO information is also kept in two separate tables: all BOLO marked as an archived will be saved in another table and the state of each BOLO before it is edited is also saved in another table

## 5.5. Security/Privacy

SQL Injection The system relies on the Wordpress framework for protection against SQL Injection.

Data Encryption: the system encrypts user passwords in the database by using the class phpass

Cross-site scripting:Wordpress comes with a Cross Site Scripting Hack prevention filter that filters all POST and COOKIE data

Cross-site request forgery: CSRF is also addressed by enabling the Cross Site Scripting Hack prevention filter

# 6. Detailed Design

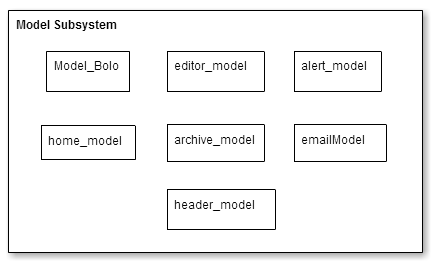
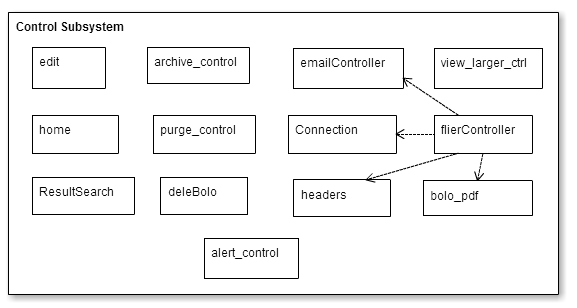
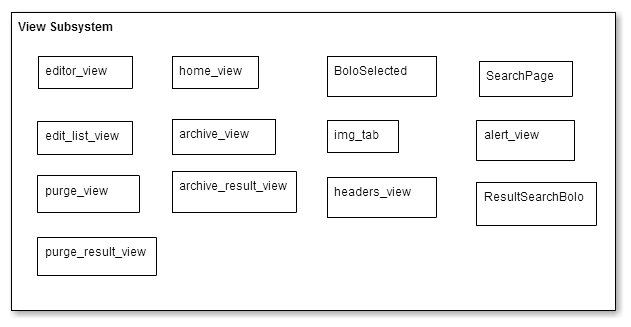
This chapter includes a detailed description of the implementation of the project. Static and dynamic models are analyzed in depth and main algorithms are described

## 6.1. Overview

The system is decomposed into 3 subsystems: Model, Controller and View Subsystems. The Model Subsystem is responsible for the controlling access to the system’s database; the View Subsystem is responsible for drawing the user interface and handling the user’s interactions; the Controller Subsystem handles all the logic, it acts as an intermediary between the View and the Model and processes and validates the data the user inputs

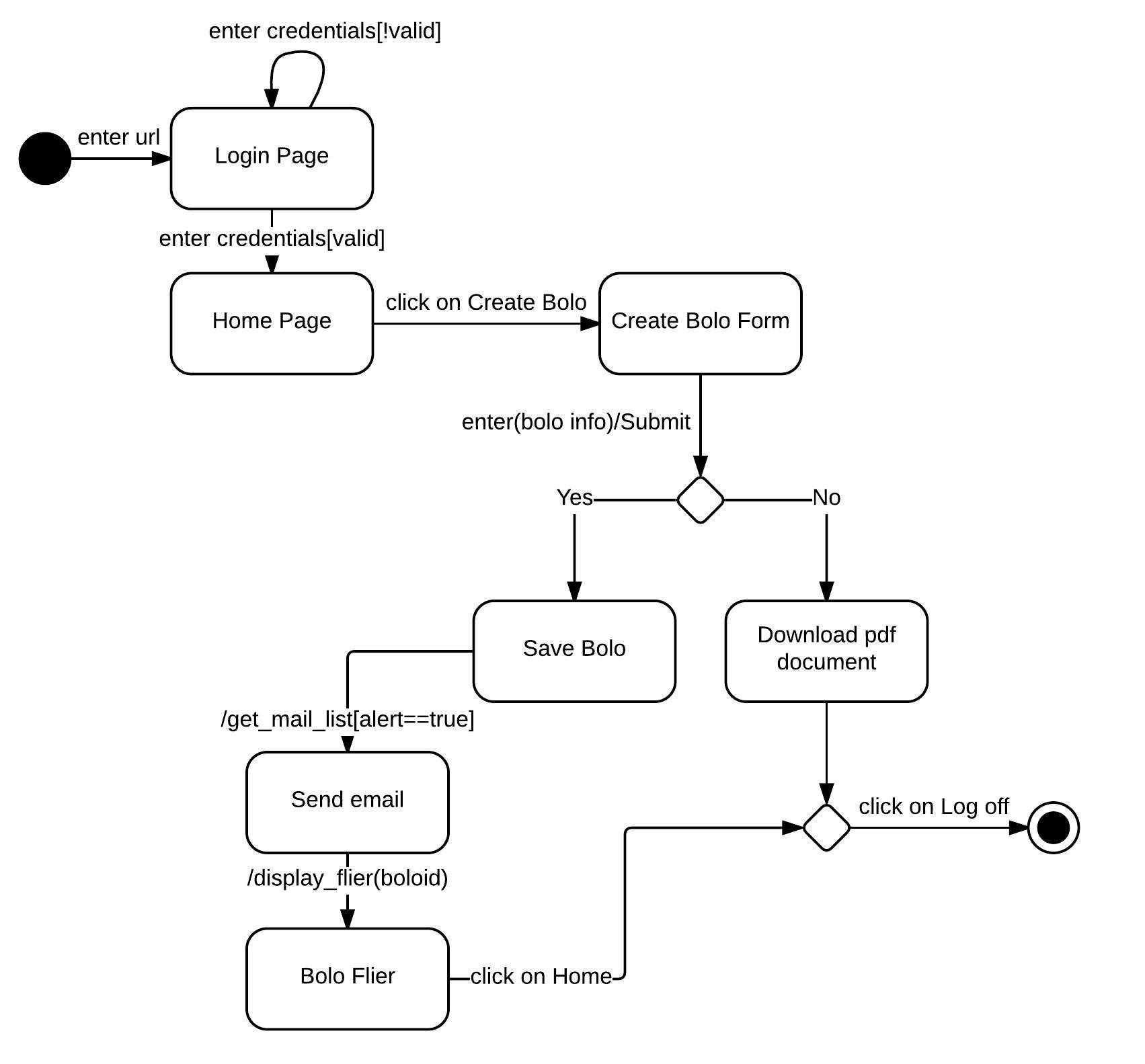
## 6.2. Static model

The following images list the classes that compose each subsystem. The classes are organized by functionality: for example, all the model, view and control classes needed to implement the Edit functionality are grouped in one folder; the ones needed for the homepage are in another folder, and so on. In the case of the Model subsystem, some of the most commonly used functions are grouped into one big class called Model\_Bolo.php. *NOTE:* classes in a subsystem are not always related to each other, but to classes in the other subsystems



## 6.3. Dynamic model

The following state machine shows the process of login in and out, submitting a BOLO and creating BOLO as a pdf document



## 6.4. Code Specification

The BOLO Flyer Creator is implemented using Wordpress as a platform , and while it doesn’t implement a MVC architecture, our code does.

The 3 MVC subsystems are further organized into smaller subsystems that handle specific functionality within the website; for instance the edit bolo functionality, it in itself a subsystem with its own model, view and controller classes, and so is the case with the rest of the features of the site.

The control tier was implemented in its majority in PHP, with a few elements in Javascript. The model subsystem was done in PHP and SQL. And the view, in HTML, CSS and PHP

# 7. System Validation

This section describes the testing performed during the implementation of the BOLO project. Testing involved dynamic and static, both automated and manual. Each test case has a test case id, purpose, test setup, input and expected output. Testing was performed on each functionality as its implementation was finished, first independently, and then integrated to the whole system

## 7.1. System Tests

|  |  |
| --- | --- |
| Test Case ID | BOLO 001 SUNNY1 |
| Purpose | To validate the system login |
| Preconditions | The user have typed in the website url |
| Input | id: tier1  pw: tier1 |
| Expected Result | Successful login as a tier1 user |

|  |  |
| --- | --- |
| Test Case ID | BOLO 001 SUNNY1 |
| Purpose | To validate the system login |
| Preconditions | The user have typed in the website url |
| Input | id: tier2  pw: tier2 |
| Expected Result | Successful login as a tier2 user |

|  |  |
| --- | --- |
| Test Case ID | BOLO 001 RAINY |
| Purpose | To validate the system login |
| Preconditions | The user have typed in the website url |
| Input | Enter a specific path in the url |
| Expected Result | The system redirects to the login screen |

|  |  |
| --- | --- |
| Test Case ID | BOLO 002 SUNNY1 |
| Purpose | To validate the logout functionality |
| Preconditions | The user has already logged into the system |
| Input | The user clicks on the logout button |
| Expected Result | The system logs the user out, and redirects to the login screen |

|  |  |
| --- | --- |
| Test Case ID | BOLO 002 RAINY |
| Purpose | To validate the logout functionality |
| Preconditions | The user has already logged into the system |
| Input | The user closes the browser |
| Expected Result | The system logs the user out |

|  |  |
| --- | --- |
| Test Case ID | BOLO 004 SUNNY1 |
| Purpose | To validate the edit functionality |
| Preconditions | The user has logged into the system  The user has clicked on Edit in the navigation bar |
| Input | User clicks on BOLO of id = PC000000792  address = “test address”  user clicks on “save” |
| Expected Result | The current state of BOLO PC000000792 has been saved in the database  BOLO number PC000000792 has been updated with a new address: “test address” |

|  |  |
| --- | --- |
| Test Case ID | BOLO 004 SUNNY2 |
| Purpose | To validate the edit functionality |
| Preconditions | The user has logged into the system  The user has selected the BOLO he wants to edit |
| Input | click on BOLO of ID: PC0000013  enter new category: “ASSAULT”  user clicks on Save |
| Expected Result | The current state of BOLO PC0000013 has been saved in the database  BOLO number PC0000013 has been updated with a new category: “ASSAULT |

|  |  |
| --- | --- |
| Test Case ID | BOLO 004 RAINY |
| Purpose | To validate the edit functionality |
| Preconditions | The user has logged into the system  The user has selected the BOLO he wants to edit |
| Input | The user closes the web browser |
| Expected Result | The BOLO is not edited nor its state is saved |

|  |  |
| --- | --- |
| Test Case ID | BOLO 005 SUNNY1 |
| Purpose | To validate the create BOLO functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | category = “BURGLARY”  User clicks on Submit |
| Expected Result | A BOLO of id = PC000000790 has been created with the current date |

|  |  |
| --- | --- |
| Test Case ID | BOLO 005 SUNNY2 |
| Purpose | To validate the create BOLO functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | category = “WANTED”  User clicks on Submit |
| Expected Result | A BOLO of id = PC000000792 has been created with the current date |

|  |  |
| --- | --- |
| Test Case ID | BOLO 005 RAINY |
| Purpose | To validate the create BOLO functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | The user closes the browser |
| Expected Result | No new BOLO is created |

|  |  |
| --- | --- |
| Test Case ID | BOLO 006 SUNNY1 |
| Purpose | To validate the thumbnails preview in the homepage |
| Preconditions | The user has logged into the system  A new BOLO of id = PC000000792 has been created |
| Input | user clicks on the Home link in the navigation bar |
| Expected Result | The first thumbnail in the list is that of BOLO PC000000792 |

|  |  |
| --- | --- |
| Test Case ID | BOLO 006 SUNNY2 |
| Purpose | To validate the thumbnails preview in the homepage |
| Preconditions | The user has logged into the system  BOLO of id = PC000000792 has been archived |
| Input | user clicks on the Home link in the navigation bar |
| Expected Result | Thumbnail of BOLO PC000000792 no longer appears in the homepage |

|  |  |
| --- | --- |
| Test Case ID | BOLO 006 RAINY |
| Purpose | To validate the thumbnails preview in the homepage |
| Preconditions | The user has logged into the system  A new BOLO of id = PC000000793 has been created |
| Input | The user closes the browser |
| Expected Result | The thumbnail corresponding to BOLO of id = PC000000793 shows in the homepage |

|  |  |
| --- | --- |
| Test Case ID | BOLO 007 SUNNY1 |
| Purpose | To validate the Save as pdf functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | category = “BURGLARY”  User clicks on Save as PDF |
| Expected Result | A flier for with the information entered in the form has been downloaded to the computer |

|  |  |
| --- | --- |
| Test Case ID | BOLO 007 SUNNY2 |
| Purpose | To validate the Save as pdf functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | category = “WANTED”  User clicks on Save as PDF |
| Expected Result | A flier for with the information entered in the form has been downloaded to the computer |

|  |  |
| --- | --- |
| Test Case ID | BOLO 007 RAINY |
| Purpose | To validate the Save as pdf functionality |
| Preconditions | The user has logged into the system  The user has clicked on Create Bolo in the navigation bar |
| Input | category = “WANTED”  User closes the browser |
| Expected Result | No pdf document is downloaded |

|  |  |
| --- | --- |
| Test Case ID | BOLO 008 SUNNY1 |
| Purpose | To validate the search bolo functionality |
| Preconditions | The user has logged into the system  The user has clicked on Search in the navigation bar |
| Input | category = “WANTED”  user clicks on Search |
| Expected Result | The system displays all BOLOs of the WANTED category |

|  |  |
| --- | --- |
| Test Case ID | BOLO 008 SUNNY2 |
| Purpose | To validate the search bolo functionality |
| Preconditions | The user has logged into the system  The user has clicked on Search in the navigation bar |
| Input | category = “WANTED”  agency = “Pinecrest”  user clicks on Search |
| Expected Result | The system displays all BOLOs of the WANTED category created by the Pinecrest Police Department |

|  |  |
| --- | --- |
| Test Case ID | BOLO 008 RAINY |
| Purpose | To validate the search bolo functionality |
| Preconditions | The user has logged into the system  The user has clicked on Search in the navigation bar |
| Input | user clicks on Search |
| Expected Result | The system displays a “no matches found” message |

|  |  |
| --- | --- |
| Test Case ID | BOLO 009 SUNNY1 |
| Purpose | To validate the browsing functionality of the homepage |
| Preconditions | The user has logged into the system  The user is in the homepage  The homepage is displaying BOLOs 1 thru 24 |
| Input | The user clicks on the Next button |
| Expected Result | The homepage displays BOLOs 25 thru 48 in the system |

|  |  |
| --- | --- |
| Test Case ID | BOLO 009 SUNNY2 |
| Purpose | To validate the browsing functionality of the homepage |
| Preconditions | The user has logged into the system  The user is in the homepage  The homepage is displaying BOLOs 49 thru 72 |
| Input | The user clicks on the Prev button |
| Expected Result | The homepage displays BOLOs 25 thru 48 |

|  |  |
| --- | --- |
| Test Case ID | BOLO 009 RAINY |
| Purpose | To validate the browsing functionality of the homepage |
| Preconditions | The user has logged into the system  The user is in the homepage |
| Input | The user clicks the Next button repeatedly in order to look at all BOLOs |
| Expected Result | The Next button is not displayed when the last BOLO is shown |

|  |  |
| --- | --- |
| Test Case ID | BOLO 013 SUNNY1 |
| Purpose | To validate the user management functionality |
| Preconditions | An administrator has logged into the system  The admin has navigated to the backend of the website  The admin has accessed the Create User Form |
| Input | name = “john smith”  user id = “jsmith”  password = “mrsmith”  agency = “Pinecrest”  click on Save |
| Expected Result | A new user of account has been created for John Smith, with id: jsmith and password: mrsmith, belonging to the Pinecrest Police Department |

|  |  |
| --- | --- |
| Test Case ID | BOLO 013 SUNNY2 |
| Purpose | To validate the user management functionality |
| Preconditions | An administrator has logged into the system  The admin has navigated to the backend of the website  The admin has accessed the Create User Form |
| Input | name = “john smith”  user id = “jsmith”  password = “mrsmith”  click on Save |
| Expected Result | The system prompts the user to enter the agency name |

|  |  |
| --- | --- |
| Test Case ID | BOLO 013 RAINY |
| Purpose | To validate the user management functionality |
| Preconditions | A non administrator has logged into the system |
| Input | The non admin enters the url of the backend of the website |
| Expected Result | The system redirects the user homepage |

|  |  |
| --- | --- |
| Test Case ID | BOLO 014 SUNNY1 |
| Purpose | To validate the BOLO Flyer View |
| Preconditions | The user has logged into the system  The user has navigated to the homepage |
| Input | The user clicks on the Details link on the thumbnail of BOLO PC000000787 |
| Expected Result | The system displays the flyer of BOLO PC000000787 |

|  |  |
| --- | --- |
| Test Case ID | BOLO 014 SUNNY2 |
| Purpose | To validate the BOLO Flyer View |
| Preconditions | The user has logged into the system  The user filled the create bolo form |
| Input | The user clicks on Submit |
| Expected Result | The flyer of the BOLO just created is diplayed |

|  |  |
| --- | --- |
| Test Case ID | BOLO 014 RAINY |
| Purpose | To validate the BOLO Flyer View |
| Preconditions | The user has logged into the system  The user has navigated to the create bolo form |
| Input | The user clicks on Submit |
| Expected Result | An empty flier is displayed |

|  |  |
| --- | --- |
| Test Case ID | BOLO 015 SUNNY1 |
| Purpose | To validate the navigation menu function |
| Preconditions | The user is in the login page of the website |
| Input | id= tier1  pw= tier1  click on log in |
| Expected Result | The navigation menu provides access to: Home, Create Bolo, Search, Edit, Alerts, Logout |

|  |  |
| --- | --- |
| Test Case ID | BOLO 015 SUNNY2 |
| Purpose | To validate the navigation menu function |
| Preconditions | The user is in the login page of the website |
| Input | id= admin  pw= admin  click on log in |
| Expected Result | The navigation menu provides access to: Home, Create Bolo, Search, Edit, Agencies, Purge Option, Archive, Alerts, Logout |

|  |  |
| --- | --- |
| Test Case ID | BOLO 016 SUNNY1 |
| Purpose | To validate the user groups functionality |
| Preconditions | An administrator has logged into the system  The admin has navigated to the Agencies section |
| Input | Admin clicks on New Agency  Agency Name = “Coral Gables”  Address = “Testing address”  Zip = “33000”  Admin clicks on Save |
| Expected Result | A new user group called Coral Gables has been created |

|  |  |
| --- | --- |
| Test Case ID | BOLO 016 RAINY |
| Purpose | To validate the user groups functionality |
| Preconditions | An administrator has logged into the system  The admin has navigated to the Agencies section |
| Input | Admin clicks on New Agency  Agency Name = “Key Biscayne”  Address = “Testing address”  Zip = “33000”  Admin clicks on Agencies in the navigation menu |
| Expected Result | The system displays the list of agencies currently enrolled and no agency named Key Biscayne is shown |

|  |  |
| --- | --- |
| Test Case ID | BOLO 017 SUNNY1 |
| Purpose | To validate the email notifications functionality |
| Preconditions | A user has logged in to the system  The user has filled a bolo form with the available information |
| Input | The user clicks on submit |
| Expected Result | All registered users receive an email with a link to the flier view of the bolo the user just created |

|  |  |
| --- | --- |
| Test Case ID | BOLO 017 RAINY |
| Purpose | To validate the email notifications functionality |
| Preconditions | A user has logged in to the system  The user has filled a bolo form with the available information |
| Input | The user clicks on Save as pdf |
| Expected Result | The system downloads a psd version of the bolo, and no email notification is sent |

|  |  |
| --- | --- |
| Test Case ID | BOLO 018 SUNNY1 |
| Purpose | To validate the purge bolo functionality |
| Preconditions | Bolo of id = PC0000001 and date created: 05/01/15 has been archived  An administrator has logged in  The admin has navigated to the Purge section |
| Input | Date from: 04/20/15  Date to: 05/02/15  The admin clicks on search to delete |
| Expected Result | Bolo of id = PC0000001 has been purged |

|  |  |
| --- | --- |
| Test Case ID | BOLO 018 SUNNY2 |
| Purpose | To validate the purge bolo functionality |
| Preconditions | Bolo of id = PC0000001 and date created: 05/01/15 has been archived  Bolo of id = PC0000002 and date created: 05/01/15 has been archived  An administrator has logged in  The admin has navigated to the Purge section |
| Input | Date from: 04/20/15  Date to: 05/02/15  The admin clicks on search to delete |
| Expected Result | Bolo of id = PC0000001 has been purged  Bolo of id = PC0000002 has been purged |

|  |  |
| --- | --- |
| Test Case ID | BOLO 018 RAINY |
| Purpose | To validate the purge bolo functionality |
| Preconditions | An administrator has logged in  The admin has navigated to the Purge section |
| Input | Date from: 04/20/15  Date to: 05/02/15  The admin clicks on search to delete |
| Expected Result | No Bolo is purged |

|  |  |
| --- | --- |
| Test Case ID | BOLO 020 SUNNY1 |
| Purpose | To validate the select default image functionality |
| Preconditions | A user has logged in to the system  The user has navigated to the create bolo page |
| Input | Category = “ARSON”  the user clicks on Submit |
| Expected Result | The predetermined default image is selected and associated with the Bolo |

|  |  |
| --- | --- |
| Test Case ID | BOLO 021 SUNNY1 |
| Purpose | To validate the view image in full size functionality |
| Preconditions | A user has logged into the system  The user has accessed the flyer view of bolo PC00000023 |
| Input | The user clicks on the image of the flier view |
| Expected Result | The image associated with the bolo is displayed in full size in a new tab |

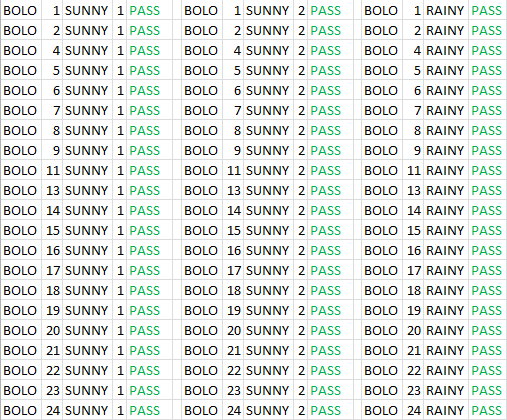
|  |  |
| --- | --- |
| Test Case ID | BOLO 022 SUNNY1 |
| Purpose | To validate the filter thumbnails of the homepage functionality |
| Preconditions | A user has logged into the system  The user has navigated to the homepage |
| Input | The user selects “West Miami” from the dropdown menu on the top right corner  The user clicks on Go |
| Expected Result | The homepage displays only thumbnails of BOLOs created by the West Miami agency |

|  |  |
| --- | --- |
| Test Case ID | BOLO 022 SUNNY2 |
| Purpose | To validate the filter thumbnails of the homepage functionality |
| Preconditions | A user has logged into the system  The user has navigated to the homepage |
| Input | The user selects “South Miami” from the dropdown menu on the top right corner  The user clicks on Go |
| Expected Result | The homepage displays only thumbnails of BOLOs created by the South Miami agency |

|  |  |
| --- | --- |
| Test Case ID | BOLO 023 SUNNY1 |
| Purpose | To validate the opt out of alerts functionality |
| Preconditions | User of id = 12 currently receives email notifications  User of id = 12 is logged into the system  The user navigates to the Alerts section of the site |
| Input | The user clicks on No  The user clicks on Save  The user creates a new bolo |
| Expected Result | The user doesn’t receive an email notification |

|  |  |
| --- | --- |
| Test Case ID | BOLO 023 SUNNY2 |
| Purpose | To validate the opt out of alerts functionality |
| Preconditions | User of id = 12 currently does not receive email notifications  User of id = 12 is logged into the system  The user navigates to the Alerts section of the site |
| Input | The user clicks on yes  The user clicks on Save  The user creates a new bolo |
| Expected Result | The user receives an email notification |

## 7.2. Evaluation of Tests – evaluate how successful the tests were. Use a tabular form.



# 

# 

# 8. Glossary - define terms used in document, especially domain specific terms.

BOLO: Be On the Look Out, the flier created for the officers as a notification to other police agencies, kinda like those wanted posters from old western movies

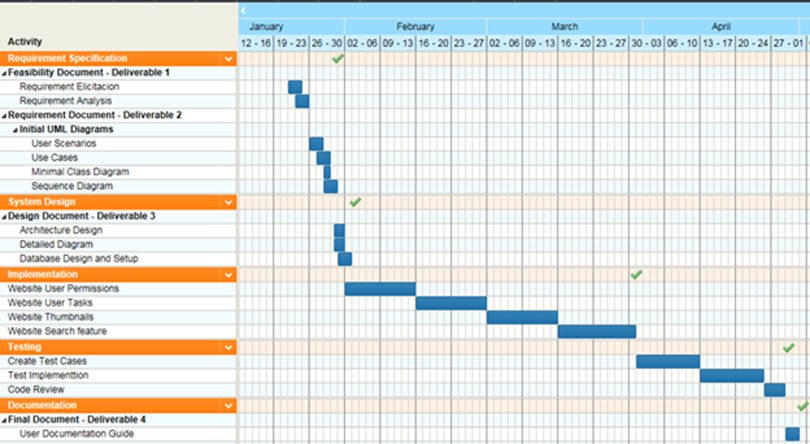
First Responder: first police officer to arrive on the crime scene who collects information on the suspect

# 

# 

# 9. Appendix

## 9.1. Appendix A - Project schedule (Gantt chart or PERT chart).



## 9.2. Appendix B – All use cases with nonfunctional requirements.

|  |  |
| --- | --- |
| Use Case ID | Bolo-001 |
| Description | Log In |
| Actor | Registered user |
| Pre-conditions | The user is on the Log in page of the system |
| Flow of Events | The user clicks on Log in link  The system will display to the user the new user form  The user will fill in all fields: username, password. |
| Alternate Flow | The user cancels the request by clicking on cancel or closing the browser. |
| Post-Conditions | The user is able to log in |
| Exceptions | The user forgot his/her password, so they will not be able to login in the system. |

|  |  |
| --- | --- |
| Use Case ID | Bolo-002 |
| Description | Log Out |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system |
| Flow of Events | The user clicks Log Out from the system. The system logs out the user from the application. |
| Alternate Flow |  |
| Post-Conditions | The system will not allowed the user to have access to the application without logging in again |
| Exceptions | The system fails logging out the user |

|  |  |
| --- | --- |
| Use Case ID | Bolo-004 |
| Description | Edit Bolo |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system  The user has clicked on the Edit link in the header |
| Flow of Events | The user clicks on the Edit option  The user add or change information  The user click on Save button |
| Alternate Flow |  |
| Post-Conditions | The new Bolo information has been saved in the database |
| Exceptions | The user do not change any field in the Bolo form |

|  |  |
| --- | --- |
| Use Case ID | Bolo-005 |
| Description | Create Bolo |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system  The user is in the Home page |
| Flow of Events | The user clicks on the “Create Bolo” button. The system will display the Bolo form. |
| Alternate Flow | User cancels the request by clicking on the Cancel button at the end of the form. |
| Post-Conditions | The user is able to see the Bolo form to start fill out all the fields needed. |
| Exceptions | The Bolo form is not displayed. |

|  |  |
| --- | --- |
| Use Case ID | Bolo-006 |
| Description | Thumbnails Page |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system  The user is in the Home page |
| Flow of Events | The system display the most recent BOLOs created  The user click on any of the Bolo to see all description |
| Alternate Flow | The user click on any other link of the page. At that time he will be redirect to another page of the website. |
| Post-Conditions | The user is able to see the thumbnails of the most recent Bolos created in the last 24 hours. |
| Exceptions | The user do not see any Bolo in the Home page because in there are not Bolos created in the last 24 hours. |

|  |  |
| --- | --- |
| Use Case ID | Bolo-007 |
| Description | Save Bolo as pdf |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system |
| Flow of Events | The user clicks on Create BOLO  the user fills the BOLO information  The user click on the “Save as pdf” button |
| Alternate Flow |  |
| Post-Conditions | The Bolo is saved as a pdf file in the user’s computer. |
| Exceptions |  |

|  |  |
| --- | --- |
| Use Case ID | Bolo-008 |
| Description | Search page |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system  The user is in the Search page |
| Flow of Events | The user select the filters need it  The user click on the search button  The system display the Bolos requested as thumbnails (same style as the homepage) |
| Alternate Flow | The user click on any other link on the page |
| Post-Conditions | The system display the Bolos requested by the user using the filters selected. |
| Exceptions | The user does not select any filter when doing the search. |

|  |  |
| --- | --- |
| Use Case ID | Bolo-009 |
| Description | Browse BOLO from homepage |
| Actor | Registered user |
| Pre-conditions | The user is Logged in the system  The user is in the homepage |
| Flow of Events | the user clicks on the next button at the end of the page to view the next 24 bolos  the user can either keep clicking next to pull the next 24 or click on prev to view the previous 24 |
| Alternate Flow | The user selects and agency before clicking on next |
| Post-Conditions |  |
| Exceptions |  |

|  |  |
| --- | --- |
| Use Case ID | Bolo-012 |
| Description | Edit BOLO |
| Actor | Tier2, Administrator, and Tier1 who created the BOLO |
| Pre-conditions | The user has clicked on the edit link on the header |
| Flow of Events | The case begins when the user clicks on the “Edit” button link next to the button he wants to edit  The system will display the bolo form prepopulated with the bolo information  The user will edit the bolo information as needed  The user will click on save |
| Alternate Flow |  |
| Post-Conditions | The changes made to the BOLO were saved on the database |

|  |  |
| --- | --- |
| Use Case ID | Bolo-013 |
| Description | User Management |
| Actor | Administrator |
| Pre-conditions | The admin has logged into the system  The admin has accessed the backend side of the website |
| Flow of Events | The use case begins when the admin goes to the user management section  The admin clicks on create user  The admin fills the form with the required data  The use case ends when the admin clicks on the save button |
| Alternate Flow |  |
| Post-Conditions | The new user account has been created and the new user can access the system |

|  |  |
| --- | --- |
| Use Case ID | Bolo-014 |
| Description | View Bolo Flier |
| Actor | All users |
| Pre-conditions | The user is in the homepage |
| Flow of Events | The use case begins when the clicks on the Details link on the thumbnail  The use case ends when system displays the b flier of the bolo |
| Alternate Flow |  |
| Post-Conditions |  |

|  |  |
| --- | --- |
| Use Case ID | Bolo-015 |
| Description | Navigation Menu |
| Actor | All users |
| Pre-conditions | The user has logged into the system |
| Flow of Events | The use case begin when the system displays a navigation bar at the top of the page  The use case ends when the user clicks in the link of the functionality he wants to access |
| Alternate Flow |  |
| Post-Conditions | The system displays the page the user clicked on |

|  |  |
| --- | --- |
| Use Case ID | Bolo-016 |
| Description | Create User Groups |
| Actor | Administrator |
| Pre-conditions | The admin has logged into the system |
| Flow of Events | The use case begins when the admin clicks on the Agencies link in the navigation bar  The admin clicks on the New Agency button  The admin fills the required information  The use case ends when the admin clicks on Save |
| Alternate Flow |  |
| Post-Conditions | The new agency has been added to the system, users can be assigned to it, bolos in the homepage can be sorted by it, and it shows in the list of the agencies enrolled in the system |

|  |  |
| --- | --- |
| Use Case ID | Bolo-017 |
| Description | Send email notifications |
| Actor | System |
| Pre-conditions | The user has filled the bolo form and |
| Flow of Events | The use case begins when the user clicks on the submit button  The system shall pull the email addresses of the users that want to receive email notification  The system shall create an email with a link to the bolo that was created  The use case ends when the system sends the emails to the addresses pulled |
| Alternate Flow |  |
| Post-Conditions | The emails have been sent |

|  |  |
| --- | --- |
| Use Case ID | Bolo-018 |
| Description | Purge Bolos |
| Actor | Administrator |
| Pre-conditions | The administrator has logged into the system |
| Flow of Events | The use case begins when the admin clicks on the Purge Option link in the header  The admin enter the date range he wishes to purge  The admin clicks on the Delete All link at the end of the list  The use case ends when the system displays a confirmation message |
| Alternate Flow |  |
| Post-Conditions | The bolos in the range have been purged from the database |

|  |  |
| --- | --- |
| Use Case ID | Bolo-019 |
| Description | Archive Bolo |
| Actor | Administrator |
| Pre-conditions | The admin has logged in the system  The admin has clicked on the Archive link on the navigation menu |
| Flow of Events | The use case begins when the admin clicks on the Purge Option link in the header  The admin enter the date range he wishes to archive  The admin clicks on the Archive All link at the end of the list  The use case ends when the system displays a confirmation message |
| Alternate Flow |  |
| Post-Conditions | The bolos in the range have been archived and no longer appear in the homepage nor on the search |

|  |  |
| --- | --- |
| Use Case ID | Bolo-020 |
| Description | No image available |
| Actor | All users |
| Pre-conditions | The user has navigated to the Create Bolo form |
| Flow of Events | The use case begins when the user fills the form, but leaves the image field empty (ie uploads no picture)  The systems shall select a default picture of a black silhouette with a “no image available” message  The use case ends when the systems saves the default image in the database |
| Alternate Flow |  |
| Post-Conditions | The bolo is assigned a default image |

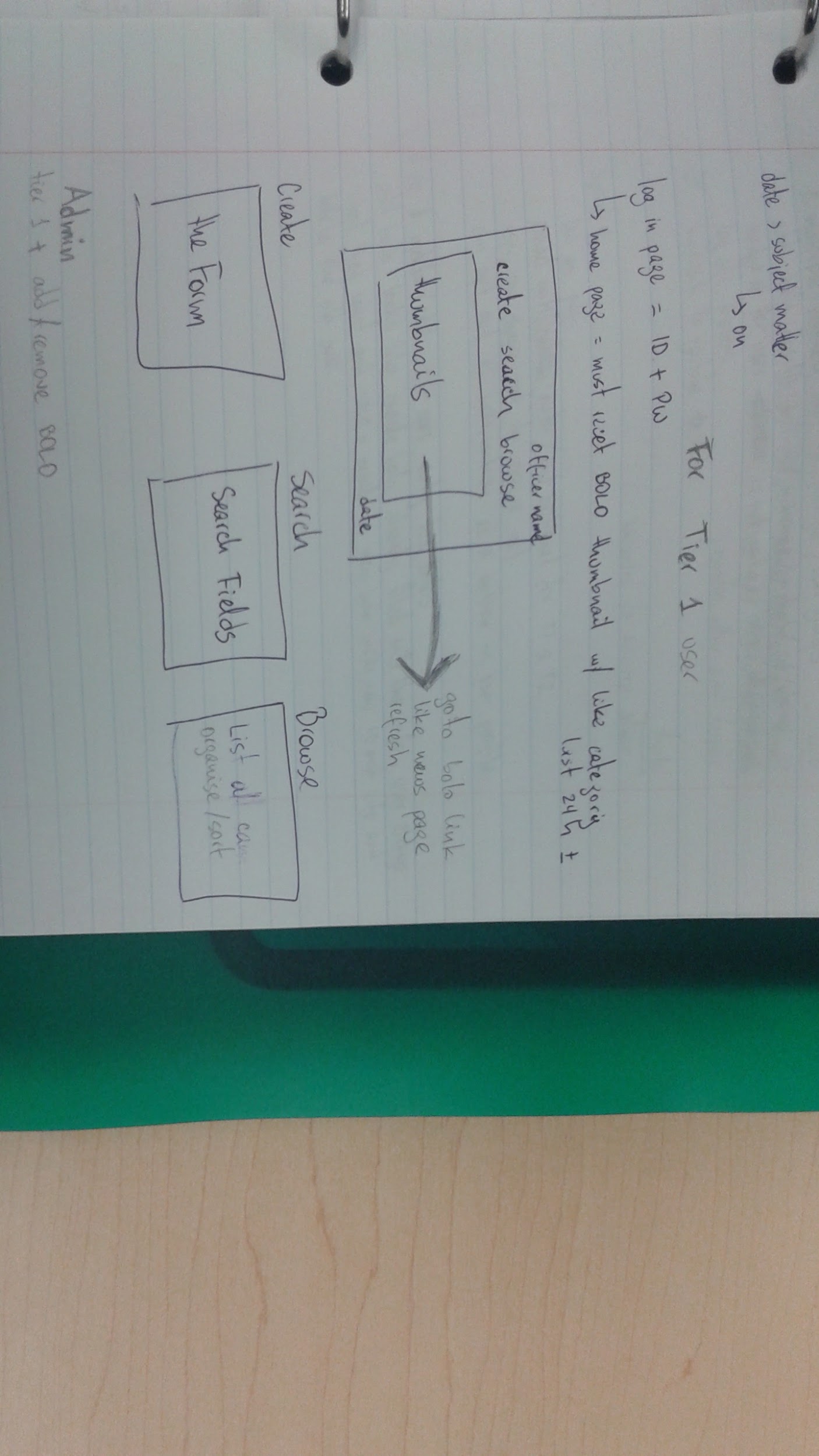
|  |  |
| --- | --- |
| Use Case ID | Bolo-021 |
| Description | View image in full size |
| Actor | All users |
| Pre-conditions | The user has pulled the flier of a bolo |
| Flow of Events | The use case begins when the user clicks on the picture in the bolo flier  The use case ends the systems displays the image in full size in a new tab |
| Alternate Flow |  |
| Post-Conditions | A new tab has been opened with the image in full size |

|  |  |
| --- | --- |
| Use Case ID | Bolo-022 |
| Description | Filter the thumbnails in the homepage by agency |
| Actor | All users |
| Pre-conditions | The user has logged into the system and has navigated to the homepage |
| Flow of Events | The use case begins when the user selects and agency from the dropdown menu in the top right corner of the homepage  The user then clicks on the Go button next to the dropdown menu  The system shall filter the bolos shown in the homepage according to the selected agency  The use case ends when the systems refreshes the homepage with the new thumbnails |
| Alternate Flow |  |
| Post-Conditions | The homepage contains only the bolo form the selected agency |

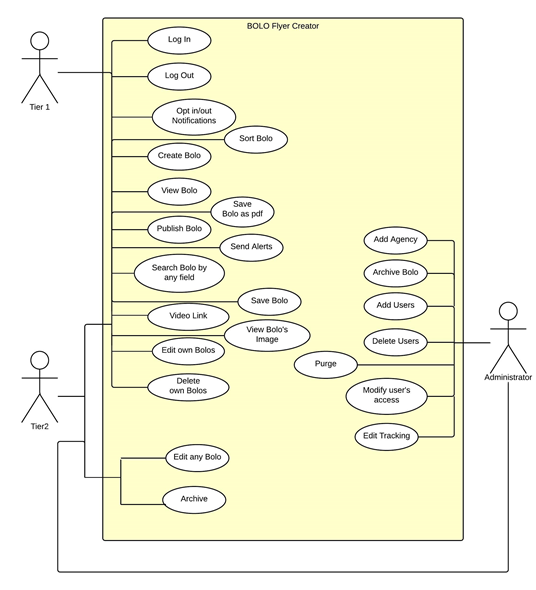
|  |  |
| --- | --- |
| Use Case ID | Bolo-023 |
| Description | Opt in / out of receiving email notifications |
| Actor | All users |
| Pre-conditions | The user has logged into the system |
| Flow of Events | The use case begins when the user clicks on the Alerts link in the navigation menu  The user then chooses whether he wants to receive email notification when a new bolo is created or not  The user clicks on the Save button  The use case ends when the system refreshes the page showing the new option chosen |
| Alternate Flow |  |
| Post-Conditions | This user won't receive emails notification until he changes this setting again |

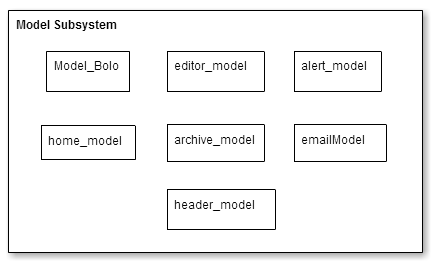
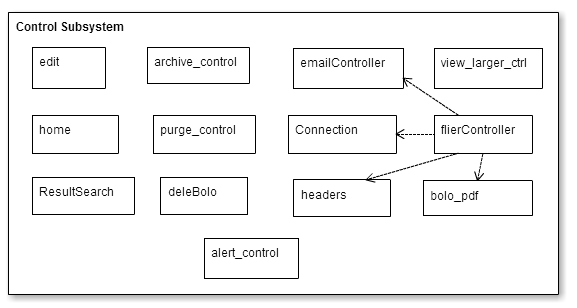
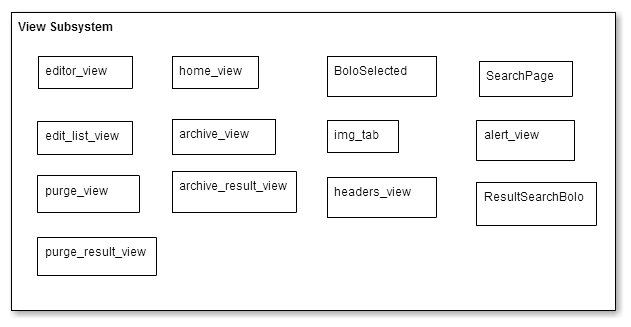
|  |  |
| --- | --- |
| Use Case ID | Bolo-024 |
| Description | Keep an edit history |
| Actor | System |
| Pre-conditions | The user has made the necessary changes to the bolo he wants to edit |
| Flow of Events | The use case begins when the user clicks on the Submit button on the end of the form  The system shall save the state of the bolo before saving the changes made in the form  The system shall save the changes made on the form  The use case ends when the system has displayed the homepage |
| Alternate Flow |  |
| Post-Conditions | The bolo that was just edited appears first in the thumbnail view of the homepage |

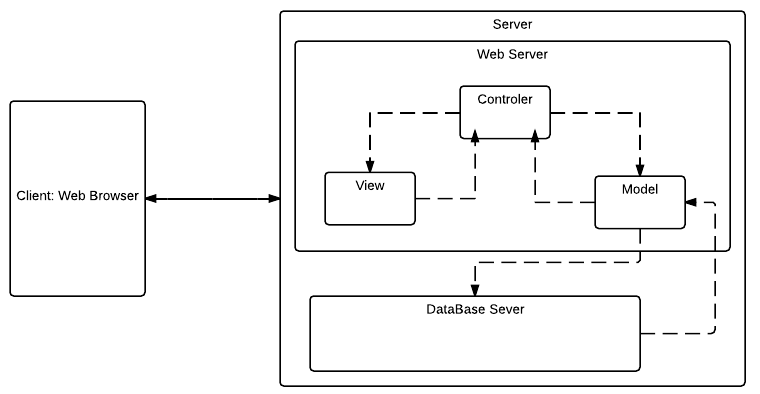
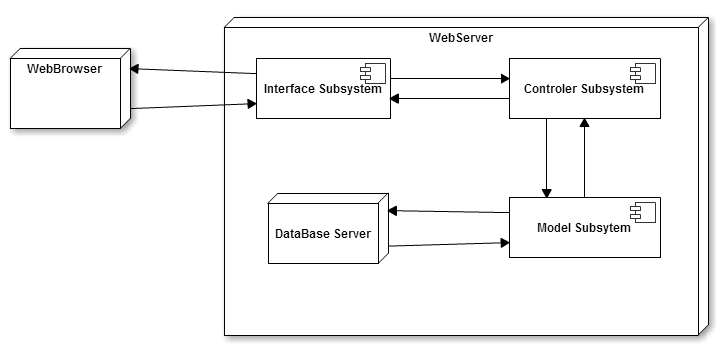
## 9.3. Appendix C – User Interface designs.



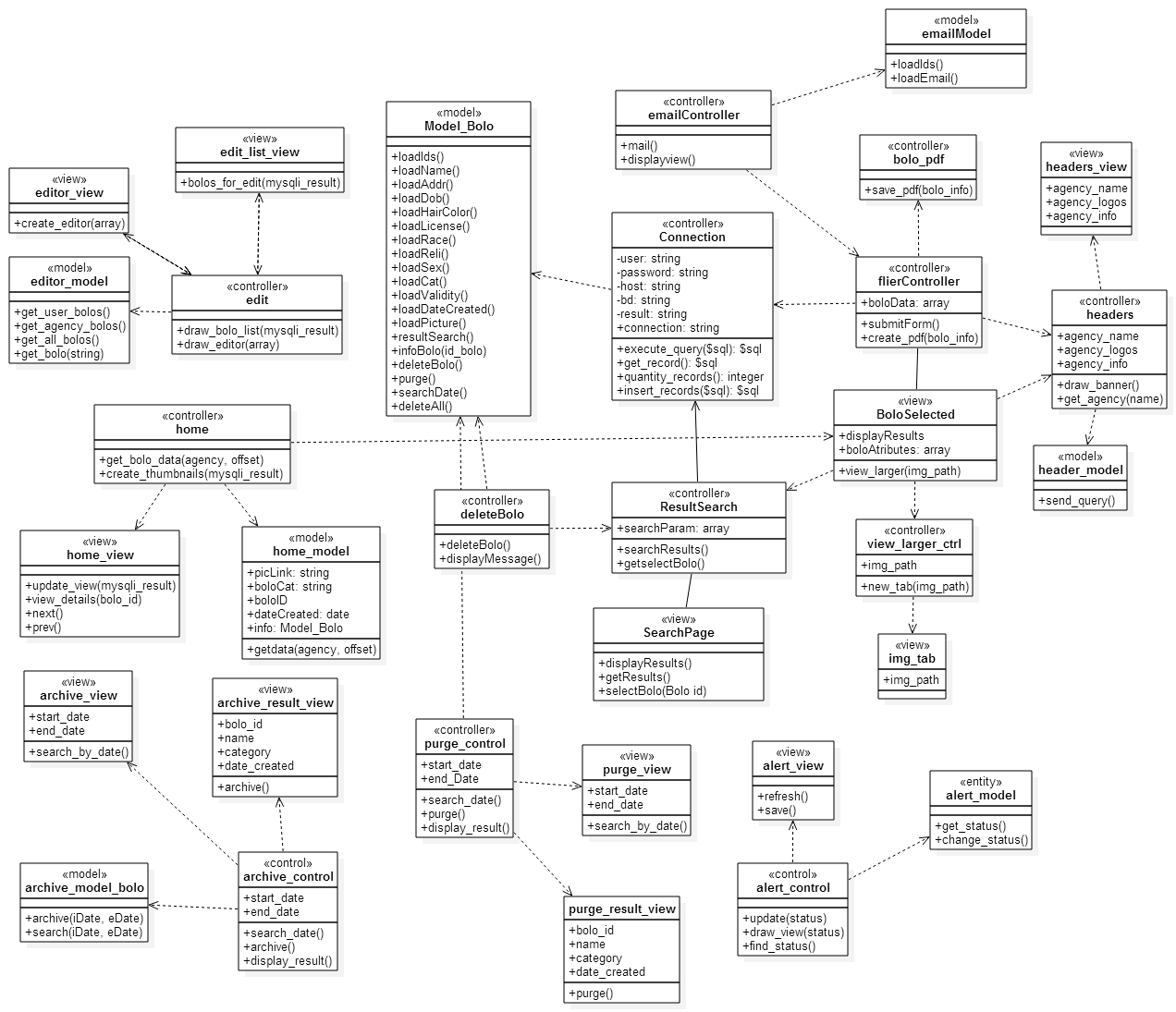
## 9.4. Appendix D – Analysis models (static and dynamic)

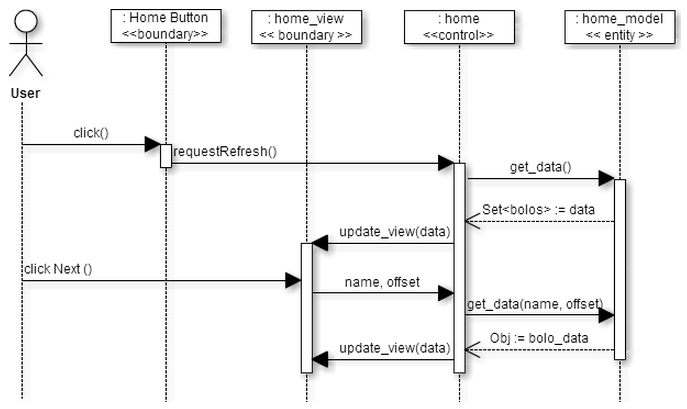


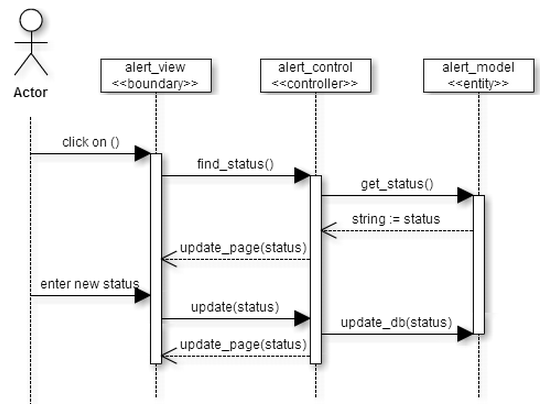


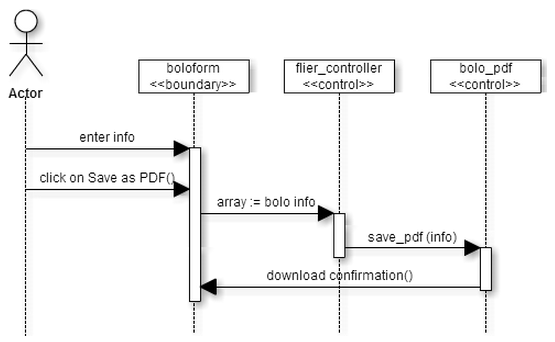


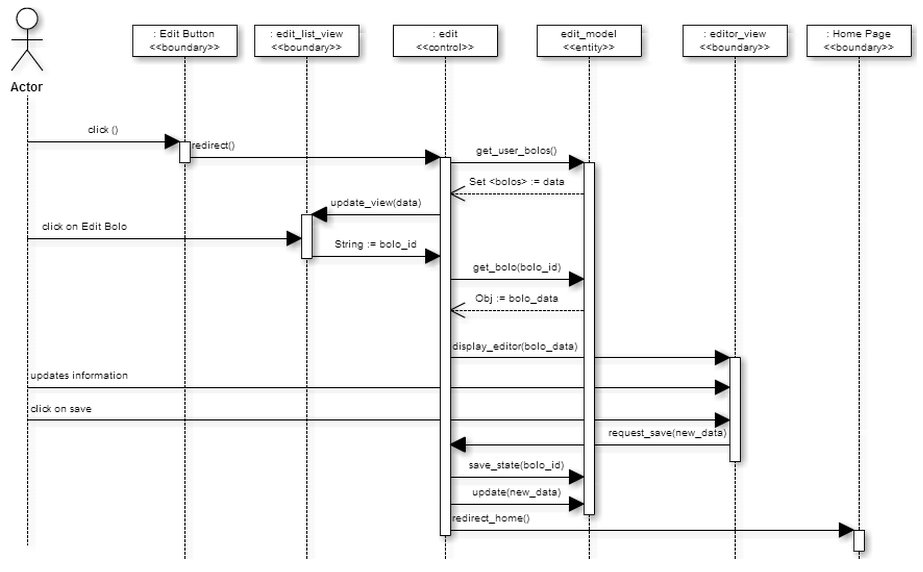
## 9.5. Appendix E – Design models (static and dynamic)

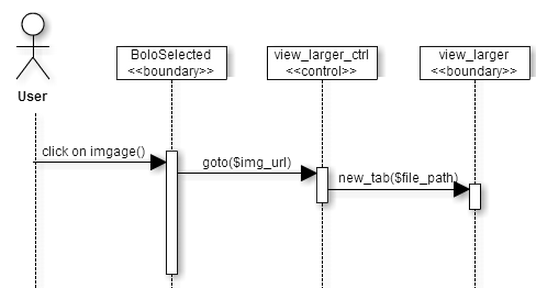
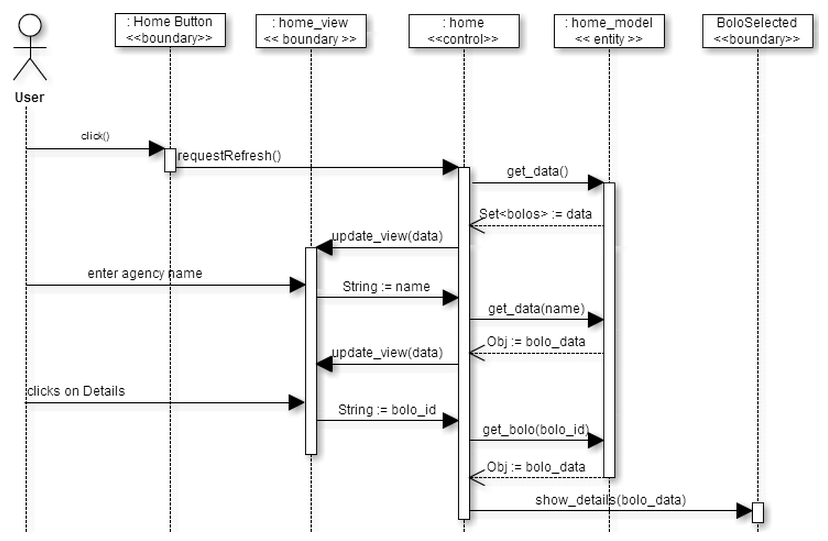


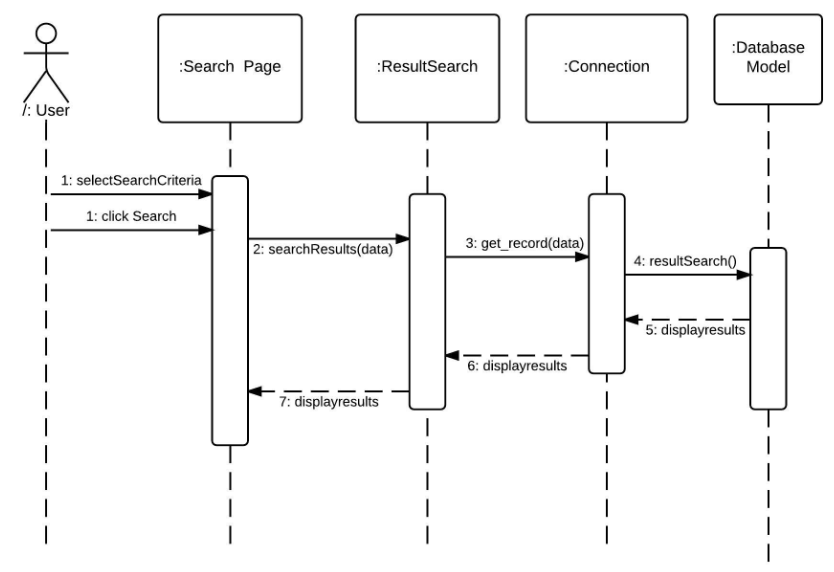
homepage next and prev buttons

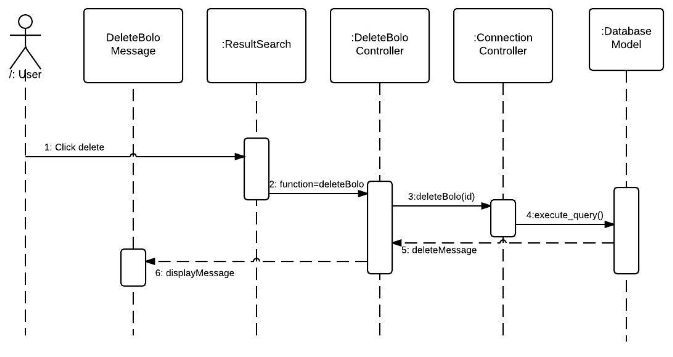
opt out/in of notifications

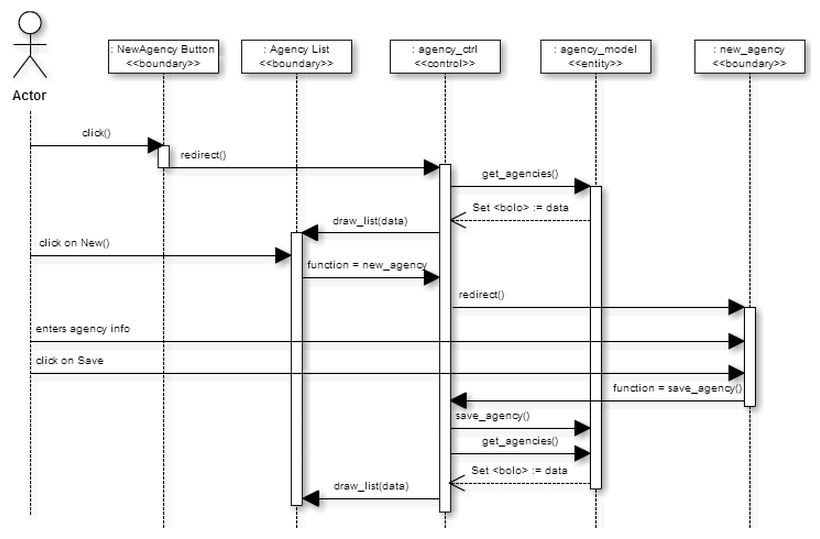
save as pdf

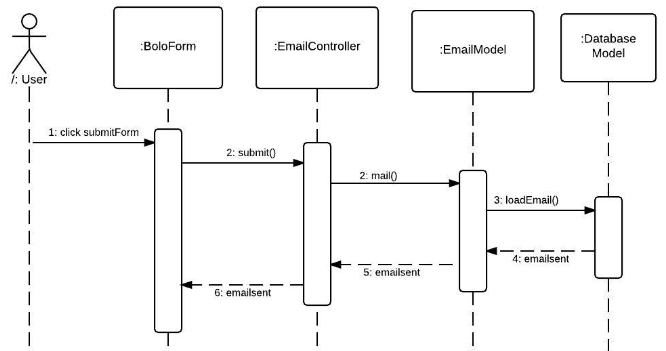
edit bolo

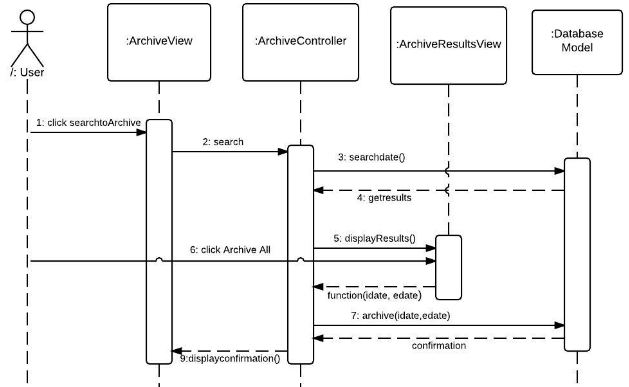
open image full sizehomepage

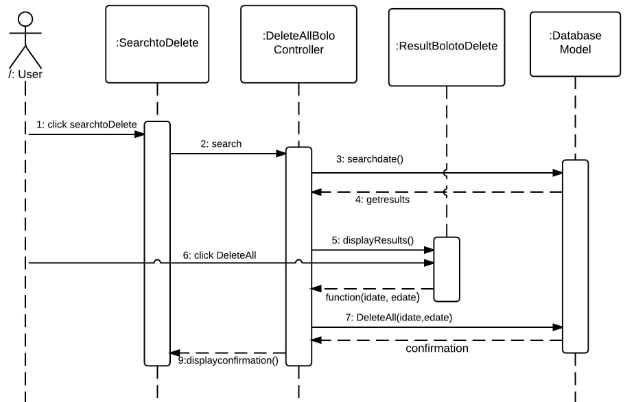
search bolo

delete bolo

create new agency

email alerts

archive bolo

purge option

## 9.6. Appendix F – Documented Class interfaces (code) and constraints.

alert\_control.php

/\*\*

\* controller for the opt out of notifications functions

\*/

class alert\_control{

public function \_construc()

// gets if the user wants notifications

public function get\_info()

}

/\*

\* This method request and update of the bolo

\*/

function update()

alerts\_model.php

class alert\_model{

public function \_construct()

/\*\*

\* gets the status of whether the user wants to receive alerts

\* @param the id of the user

\* @return true or false depending on whether the user wants to receive alerts

\*/

public function get\_alert($user)

/\*\*

\* updates the status of if the user wants to receive alerts

\* @param $user the id of the current user (is a number)

\* @param $status if the user wants to receive alartes (true or false)

\*/

public function update\_alert($user, $status)

}

archive\_model.php

class archive\_model{

public function \_construct(){ }

//marks a bolo as archived

//@param $bolo\_id for the id of the bolo to be archived

public function archive($bolo\_id)

bolo\_pdf.php

class bolo\_pdf{

public function \_construct(){ }

/\*\*

\* creates a pdf document version of the bolo passed as a parameter.

\* the bolo fits in one single a4 size paper

\* the pdf document is downloaded automatically or prompted to open or save; depends \*on the browser being used

\* @param $bolo this must be a mysqli\_result object containing the bolo

\* you want to create a pdf of

\*/

public function save\_pdf($bolo)

}

edit\_list\_view.php

class edit\_list\_view{

public function \_\_construct()

/\*

\* displays all the bolos the current user can edit

\* @param $result a mysqli\_result object with the data to be displayed

\*/

public function bolos\_for\_edit($result)

}

edit\_model.php

class edit\_model{

public function \_construc()

/\*

\* this function returns all the bolos the current user can edit

\* (ie the user currently loged in)

\* @return a mysqli\_result object with the bolos

\*/

public function get\_user\_bolos()

/\*

\* this function returns all the info of the bolo matching the id

\* @param $boloid the id of the bolo you want the info of

\* @return the mysqli\_result object

\*/

public function get\_bolo($boloid)

/\*\*

\* updates a bolo given all the parameters

\*/

public function update\_bolo($boloid, $selectcat, $myName, $lastName, $dob, $DLnumber, $race, $sex, $height, $weight, $haircolor, $address, $tattoos,

$adtnlinfo, $summary, $newfilename, $val, $rel, $clas, $update, $author\_id, $link)

/\*\*

\* archives the bolo matching the given id from the database

\* That is: copies the desired bolo into the bolo\_archive table

\* and marks it as archived in the wp\_flierform table

\*

\* There is some redundancy in this function: a bolo is marked as archived in the wp\_formflier

\* and then copied into bolo\_archive. It could be one of the two ways: either marked as archived

\* and kept in wp\_flierform or copied into bolo\_archive and deleted from wp\_flierform. It was done this

\* because BOLOs must not be deleted ever. So it servers as a backup/failsafe

\*/

public function archive($boloid)

}

editor\_controller.php

class editor\_controller{

public function \_contruct()

/\*

\* request the model object for the desired bolo

\* @param the id of the bolo

\* @returns an associative array with the info of the bolo

\*/

public function get\_bolo($boloId)

/\*

\* This method request an update of the bolo

\*/

function update()

home\_model.php

class home\_model{

public function \_construct()

/\*\*

\* gets 24 bolos for the specified agency

\* @param $agency\_name for the name of the agency, to select the last 24 bolos overal enter $agency\_name = 'Show ALL'

\* @param offset for offset from the most recent bolo created: 0 will get the most recent 24, offset = 24 would

\* get bolos 24 to 48, and so on

\*/

public function get\_data($agency\_name, $offset)

/\*\*

\* gets the list of all the agencies enrolled

\* @return a mysqli\_result object with all the agencies

\*/

public function get\_agencies()

home\_view.php

/\*\*

\* method to draw the the homepage view: thumbnails, dropdown box and buttons needed

\* @param $result a mysqli\_result object containing the information of the bolos to displayed

\* @param $agencies a mysqli\_result object containing the information of the agencies enroled in the system

\* @param $current\_agency a string with the name of the agency by which the bolos are filtered

\* @param $offset for the current offset

\*/

public function update\_view($result, $agencies, $current\_agency, $offset)

new\_agency\_control.php

/\*\*

\* saves the new agency to the database

\*/

function save\_agency()

new\_agency\_model.php

class agency\_model{

public function \_construc()

/\*

\* saves the new agency to the database

\*/

public function save\_agency($name, $address, $city, $zip, $phone)

public function get\_agencies()

emailModel.php

class EmailModel {

public function \_\_construct()

//returns all the user ids in a mysqli\_result object

public function loadIds()

//returns the emails of all the users who want to receive notifications

public function loadEmail()

//returns the bolo last created

public function getlast()

## 9.7. Appendix G – Documented code for test drivers and stubs.

## 9.8. Appendix H – Diary of meeting and tasks for the **entire semester**.

10. References

**User’s Guide:**

1. Cover Page
2. Copyright and trademark information.
3. Introduction – brief description of application.
4. Hardware and Software Requirements.
5. Installation and setup
6. Getting Started - How to run application. Include user screens where appropriate and the description to run the two most important use cases.
7. Quick reference
8. Accessing online help
9. References

**You should hand in two (2) separate documents (System Documentation and User’s Guide) both of them properly formatted and bound.**

**You are also required to hand in a CD with ALL the artifacts for the semester, including: all presentations, documents, uml diagrams and source code.**

**Also include contact information (Tel #, address) in a text file in case we need to contact you.**