Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title:
BOLO Flier Creator Version 7

Team Members: Mario Siu, Rachel Hill, Victor Estopinan

Product Owner(s): Jason Cohen, Frank Alvarado

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) *2017 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

The Pinecrest Police Department(PPD) sought help from FIU's college of engineering and computer science to digitalize the current process used to fill and distribute criminal fliers. BOLO Flier Creator, a web application created on 2015, was FIU's response to tackle PPD's difficulties. The previous BOLO iteration, BOLO - Version 6.0, was faulty and buggy. On the other hand, the current BOLO iteration, Version 7.0, fixed most bugs and added new features. This document presents the information necessary to gain a good understanding of BOLO Flier Creator - Version 7.0.

Table of Contents

INTRODUCTION
……..4
CURRENT SYSTEM
……..4
PURPOSE OF NEW SYSTEM
……4
USER STORIES
IMPLEMENTED USER STORIES
……..5
PENDING USER STORIES
……..140
PROJECT PLAN
HARDWARE AND SOFTWARE RESOURCES
……..6
SPRINTS PLAN
……..7
Sprint 1
……..7
Sprint 2
……..8
Sprint 3
…… 13
Sprint 4
…… 17
Sprint 5
…… 22
Sprint 6
…… 26
SYSTEM DESIGN

ARCHITECTURAL PATTERNS
…….. 32
SYSTEM AND SUBSYSTEM DECOMPOSITION ……..…
33
DEPLOYMENT DIAGRAM
……..…... 34
DESIGN PATTERNS
……..….... 35

SYSTEM VALIDATION
……...……...35

GLOSSARY
……...……
..57

APPENDIX
……..……
….58
APPENDIX A - UML DIAGRAMS
…….. 58
Static UML Diagrams
…….. 58
Dynamic UML Diagrams
…….. 74
APPENDIX B - USER INTERFACE DESIGN ……..….
100
APPENDIX C - SPRINT REVIEW REPORTS
……...…..125
APPENDIX D - OTHER
DOCUMENTS……..134
USER
GUIDE…….
….134 INSTALLATION/MAINTENANCE DOCUMENT
……..…..138
SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS ……..139

REFERENCES
……..……..……..……..14
2

INTRODUCTION

BOLO Flier Creator is a web application developed in order to convert the current flier distribution process, used by the Pinecrest Police Department (PPD), into a digital one. Today's PPD process consists of police officers manually filling out BOLO (Be on the Lookout) flier reports and then sending them out to law-enforcement agencies. This process tends to be time-consuming, inefficient, and costly. On the other hand, BOLO improves all of these difficulties by securely dispersing BOLOs via the web, ensuring a swifter response and widening the situational awareness of other police departments.

Current System

The current system, BOLO-6.0, is not in a stable state for deployment. It was discovered that the system has several fatal bugs that would cause the server to shutdown during production. Too add to these problems, many of the system features, such as searching and filtering BOLOs, resetting a password, generating PDFs, and subscribing to agencies, do not meet their expected functionality. Other system features are not user-friendly to the non-technical staff. This requires assistance from programmers just to complete simple tasks,such as updating the login page and making changes to the user guide.

Purpose of New System

BOLO-7.0 fixes most bugs, improves older implementations, and adds new features to the system, such as the option to keep a BOLO strictly accessible within the agency, the functionality to subscribe to and unsubscribe from agencies, and the ability to delete archived BOLOs within a given date range. Customization of the login page and the user guide was modified for ease of use, lifting the burden imposed on the root users of having to write markdown code. More features implemented into the system will be discussed further in the following sections.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the BOLO-7.0 project. These user stories served as the basis for implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

Sprint 1
- #1215 - Set up the BOLO-6.0 environment

Sprint 2
- #1217 - Fix BOLO pdf
- #1218 - Search Issue in the System
- #1222 - Password Validation
- #1246 - Prevent server from crashing when creating a new bolo without selecting a category

Sprint 3
- #1219 - Prevent BOLO from Disappearing when Updated
- #1265 - Subscribing to Agency
- #1223 - BOLO Filter not Working

Sprint 4
- #1286 - Notifying Users Subscribed to Agencies
- #1287 - Edit User Guide
- #1288 - Edit Login Page

Sprint 5
- #1315 - Forgot Password
- #1316 - Add Delete Options To Categories
- #1320 - Edit login page fix bugs

Sprint 6
- #1328 - Add internal option to create BOLO form
- #1336 - Reduce size of Officer's badge panel
- #1337 - Change Date to US Standard
- #1338 - Add a purge date to delete archive BOLOs
- #1357 - Impede users from navigating page after entering temporary password

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in this section. This section also describes the components, both software and hardware, chosen for the project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware Specs Recommended:

- 30 Gb of free hard drive space
- 8 Gb of Memory RAM
- 2 GHz of processor speed or faster

Hardware Specs Used:

- 1 Tb hard drive space
- 16 Gb RAM
- 3.7 GHz processor

Software Technologies Used:

- Windows 10 OS
- Node.js 6.10
- HTML 5 / CSS 3
- Pug
- Javascript
- NPM 3.10.10
- Visual Studio Code(for Windows)
- MongoDB
- Github / Gitbash
- Sendgrid
- Notepad++
- Apache HTTP Server

Sprint Plan

The main target of the BOLO 7.0 project was to fix the current system bugs and errors in order to make it functional. After fixing most bugs, some features were remodeled and new ones were added to the application.

Sprint 1

User Story #1215 - Set up the BOLO-6.0 environment

- As a User, I would like to have a running BOLO-6.0 environment setup on my local server so that I could develop and test code.

Acceptance Criteria

- Users can log in to the system.
- Users can receive emails from the system

Tasks

- Create a SendGrid account (ID #1264)

Use Case

- Name: Access BOLO Server
- Actors:
    - Administrator
    - Officer
    - Supervisor
    - Root

- Preconditions:
    - MongoDB server is running
    - Node JS is running
- Description:
    1. The user will navigate to http://localhost:3000
    2. The user will log into the system.

Modeling

N/A

Sprint 2

User Story #1217 - Fix BOLO PDF

- As an user, I can create Bolo PDFs that show only filled attributes.

Acceptance Criteria

- When user clicks Preview when creating a BOLO, or Details, or Generate PDF only the fields that were filled will be shown on.

Tasks

- Take out attributes in PDF generation (ID #1230)
- Take out N/A attributes while creating BOLO (ID #1229)
- Take out N/A attributes while in details (ID #1228)

Use Case

- Name: ID #1230
- Actor: User
- Preconditions:
    - User must be signed in.

- User must be looking at a BOLO's details

- Description :
  1 - User clicks on generate pdf.
  2 - System generates pdf without N/A attributes.

Use Case

- Name: ID #1229
- Actor: User
- Preconditions:
  - User must be signed in.
  - User must be creating a BOLO
- Description :
  1 - User clicks on details.
  2 - System renders BOLO preview page without N/A attributes.

Use Case

- Name: ID #1228
- Actor: User
- Preconditions:
  - User must be signed in.
  - User must be at archive or at home
  - There must be at least one BOLO archived or active
- Description:
  1 - User clicks on details.
  2 - System renders BOLO Details page without N/A attributes.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 3a-3c and Dynamic UML Diagrams, Figure 21a-21c.

User Story #1218 - Search Issue in the System

- As a User I would like to be able to search any BOLO in the system from any agency

Acceptance Criteria

- Any user (Officer, Admin, Supervisor) can search any BOLO without the system scratch
- Any user can search any BOLO by category by setting the category in the drop down select box
- Any user can search any BOLO by agency by setting the agency in the drop down select box
- Once the user search under any category, user can enter any keyword in the text boxes to search BOLOs

Use Case

- Name: ID#1226 Search BOLO should works even if not input are provided
- Actor: User
- Preconditions:
    - User must be logged In
    - There should be any BOLO archived or active
    - User should click of the Search button in the menu
- Description :
    - User should click on Search button on the menu
    - User click on Search button without choosing any agency or category

Use Case

- Name: ID#1245 Search BOLOs by category and agency should work combined

- Actor: User
- Preconditions:
    - User must be logged In
    - There should be any BOLO archived or active
    - User should click of the Search button in the menu
- Description :
    - User should click on Search button on the menu
    - User should set any agency

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 4 and Dynamic UML Diagrams, Figure 22.

User Story #1222 - Password Validation

- As an user, I would like that when creating a new password, the system matches if password and confirmation fields. Also system must not crash if no password is submitted.

Acceptance Criteria

- “password” and “repeat password” input boxes match.
- Must happen when creating a new password or changing password.
- System must not crash if password or confirmation fields are empty when user clicks submit.

Use Case

- Name: Password Validation
- Actor: User

- Preconditions:
    - User must have an account.
    - User must be changing password or creating password for first time.
- Description:

1 - User fills Password and Repeat Password fields.

2 - If password and confirmation match, and password has a valid format, then the system updates password and informs the user.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 5 and Dynamic UML Diagrams, Figure 23.

User Story #1246 - Prevent server from crashing when creating a new bolo without selecting a category

- As a User, I would like to view an error message after attempting to create a new bolo without selecting a category first so that I am aware of the error in the form.

Acceptance Criteria

- An error message is displayed if the user inputs a date and a time but forgets to select a category.
- A new bolo is not created if the user submits the form without selecting a category
- The server is still running if the user submits the form without selecting a category.

Use Case

- Name: Creating a New BOLO
- Actors

- Administrator
- Officer
- Supervisor
- Root

- Preconditions:
  - The user has BOLO categories created.
  - The user invokes the option to create a new BOLO.
- Description:
  1. The user will fill out the form and submit.
  2. The system will validate the form and create a new BOLO.
- Exceptions

  1A: The user wants to preview the BOLO before submitting

  1. The user will select the option to preview.
  2. The system will validate the form and create a preview.
  3. The user will select the option to go back and repeat step 1.

  2A: The system finds an error in the form.

  4. The system will notify the user of invalid data.
  5. The user will correct the error in the form.
  6. The use case will repeat step 2.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 6 and Dynamic UML Diagrams, Figure 24.

Sprint 3

User Story #1221 - Setup bolo.cs.fiu.edu server

- As a User, I should be able to deploy the BOLO system to a dedicated production VM so that it could be used for demos and testing.

Acceptance Criteria

- Users can navigate to http://bolo.cs.fiu.edu in order to access the BOLO system.
- Users can log in to the system.
- Users can receive emails from the system.

Tasks

- Update SendGrid Key (ID #1268)
- Investigate why server keeps shutting down (ID #1267)
- Investigate why server keeps redirecting to http://localhost:3000 (ID #1266)
- Run MongoDB and Node js as background processes (ID #1259)
- Allow access to the site via the domain name (ID #1258)

Use Case

- Name: Access BOLO Server
- Actors:
    - Administrator
    - Officer
    - Supervisor
    - Root
- Preconditions:
    - MongoDB server is running on bolo.cs.fiu.edu
    - Node JS is running on bolo.cs.fiu.edu
    - Apache2 is running on bolo.cs.fiu.edu
- Description:
    1. The user will navigate to http://bolo.cs.fiu.edu
    2. The user will log into the system.

Modeling

N/A

User Story #1219 - Prevent BOLO from Disappearing when Updated

- As a User, I would like the original BOLO to be visible if the BOLO has been updated and not confirmed.

Acceptance Criteria

- The original BOLO should be visible on the BOLO page until the updated BOLO is confirmed.
- The original BOLO should not be updated if the updated BOLO is not confirmed.
- The updated BOLO appears with the “Updated” title.
- If multiple updates were performed on the same BOLO, only the update that was confirmed first will be visible on the BOLO page while the other updates will be ignored.

Use Case

- Name: Editing a BOLO
- Actors
    - Administrator
    - Officer
    - Supervisor
    - Root
- Preconditions:
    - The user has already created a BOLO.
    - The user has invoked the option to edit the BOLO.
- Description:
    1. The system will prefill the form with existing data.
    2. The user will make changes to the form and submit.
    3. The system will validate the form and send an email confirmation.

4. The user will confirm the updated BOLO.
5. The system will display the updated BOLO.

- Exceptions

2A: The user selects the option to “Go Back”.

1. The system will display the Bolo page.
2. The use case will terminate.

3A: The system finds an error in the form.

1. The system will notify the user of invalid data.
2. The user will correct the error in the form.
3. The use case will repeat step 2.

5A: The system finds that another update overrode the user’s update.

1. The system will notify the user that the BOLO to confirm could not be found.
2. The use case will terminate.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 7 and Dynamic UML Diagrams, Figure 25.

User Story #1265 - Subscribing to Agency

- As an user, I would like to subscribe and unsubscribe to/from multiple agencies.

Acceptance Criteria

- Able to subscribe/unsubscribe to multiple agencies
- When an user subscribes to an agency, its subscribed agencies are updated
- When an user unsubscribes from an agency, its subscribed agencies are updated
- If user tries to subscribe/unsubscribe while no marking any agency, page will refresh itself.

Use Case

- Name: ID #1265
- Actor: User
- Preconditions:
    - User must be signed in.
    - User must have clicked on account.
- Description :

1- User clicks the “Notifications” button.

2- System redirects to a new page and shows subscribed agencies.

3- If user marks any agency and clicks unsubscribed, the system will update the page.

4- If user clicks on “More Agencies”, the system will redirect to a new page and shows unsubscribed agencies.

5 - If user marks any agency and clicks submit, the system will update the page.

6 - If user clicks Go back to your notifications, the system redirects to previous page and shows subscribed agencies.

- Exceptions :
    - If user clicks submit/unsubscribe when no agency have been marked, then the system will refresh the page.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 8 and Dynamic UML Diagrams, Figure 26a-26d.

Sprint 4

User Story #1286 - Notifying Users Subscribed to Agencies

- As an user, I would like to be notified for any created/updated BOLO belonging to agencies I subscribed to.

Acceptance Criteria

1. After a BOLO is created/updated, emails will be sent to each user that subscribed to the agency of that BOLO
2. Notification email must include a link to the created/updated BOLO details page for user to click it.
3. Notification email must include a pdf showing a picture of the BOLO, its status, and its agency.

Use Case

- Name: Notifying Users Subscribed To Agencies
- Actor: User
- Preconditions:
    - User must be signed in.
    - User must have created/updated a BOLO
- Description:

1- User receives confirmation email and clicks confirmation link.

2- System receives confirmation.

3- System send a notification email to the BOLO author and to any user that is subscribed to the BOLO or to the BOLO's agency.

4- Users receive notification email.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 9 and Dynamic UML Diagrams, Figure 27.

User Story #1287 - Edit User Guide

- As a Root User, I would like to edit the Root, Administrator, Supervisor, and Officer User Guides.

Acceptance Criteria

Root user could:

- Edit the user guide
- Create and edit new sections of the User Guide
- Specify who has access to each section
- Delete a section of the User Guide
- Preview the Root, Administrator, Supervisor, and Officer User Guides

Use Case

- Name: Edit User Guide Title
- Actors
    - Root
- Preconditions:
    - The user has invoked the option to edit the User Guide.
- Description:
    1. The user will edit the title and submit.
    2. The system will validate the title and update the User Guide.
- Exceptions

  2A: The system finds that the title is invalid.
    1. The system will notify the user of invalid data.
    2. The user will repeat step 1.

Use Case

- Name: Add User Guide Section

- Actors
    - Root
- Preconditions:
    - The user has invoked the option to edit the User Guide.
- Description:
    1. The user will select the option to add a new section.
    2. The system will display a new form.
    3. The user will fill out the the form and submit.
    4. The system will validate the form and create a new section.
- Exceptions

  3A: The user selects the option to return.
    1. The system will display the User Guide page.
    2. The use case will terminate.

  4A: The system finds an error in the form.
    1. The system will notify the user of invalid data.
    2. The user will correct the error in the form.
    3. The use case will repeat step 3.

Use Case

- Name: Edit User Guide Section
- Actors
    - Root
- Preconditions:
    - The user has invoked the option to edit the User Guide.
- Description:
    1. The user will invoke the option to edit the section.
    2. The system will prefill the form.
    3. The user will make changes to the the form and submit.
    4. The system will validate the form and update the section.
- Exceptions

  3A: The user selects the option to return.

1. The system will display the User Guide page.
2. The use case will terminate.

3B: The user selects the option to delete the section.

1. The system will remove the section from the User Guide and display the User Guide page.
2. The use case will terminate.

4A: The system finds an error in the form.

1. The system will notify the user of invalid data.
2. The user will correct the error in the form.
3. The use case will repeat step 3.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 10 and Dynamic UML Diagrams, Figure 28a-28c.

User Story #1288 - Edit Login Page

- As a Root Admin, I would like to be able to edit the login page.

Acceptance Criteria

- User with root access could change the image on the login page.
- User with root access could change the titles and subtitles on the login page.

Use Case

- Name:ID # 1302 Edit Title and Subtitle of LogIn Page
- Actor:Root
- Preconditions:
    - The user should has root access
    - The root should be Logged In
    - The root should get into the Admin menu

- Description:
    - The root should log In
    - The root should get into the Admin menu
    - The root should click on the Edit Log In Page
    - The root should change the title and subtitle by enter them in the given text fields
    - The root should save the changes by clicking in the SAVE button

Use Case

- Name:ID # 1303 Edit Logo of Login Page
- Actor:Root
- Preconditions:
    - The user should has root access
    - The root should be Logged In
    - The root should get into the Admin menu
- Description:
    - The root should log In
    - The root should get into the Admin menu
    - The root should click on the Edit Log In Page
    - The root should upload the image for logo and by clicking the change logo button
    - The root should save the changes by clicking in the SAVE button

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 11 and Dynamic UML Diagrams, Figure 29.

Sprint 5

User Story #1315 - Forgot Password

- As a User, I would like to reset my password if it is forgotten.

Acceptance Criteria

- User receives an email with instructions to reset a forgotten password.
- User could reset password after clicking the link found on the email.

Use Case

- Name: Forgot Password
- Actors
    - Root
    - Administrator
    - Supervisor
    - Officer
- Preconditions:
    - The user has invoked the Forgot Password option.
- Description:
    1. The user will fill out the Forgot Password Form and submit.
    2. The system will validate the form and send the user an email.
    3. The user will click the link in the email to invoke the Reset Password Use Case.
- Exceptions

  2A: The system finds that the form data is invalid.
    1. The system will notify the user of any invalid data.
    2. The user will repeat step 1.

Use Case

- Name: Reset Password
- Actors
    - Root
    - Administrator
    - Supervisor

- Officer

- Preconditions:
  - The user has invoked the option to Reset Password.
- Description:
  1. The user will fill out the Reset Password Form.
  2. The system will validate the form and update the user's password.
- Exceptions

  2A: The system finds that the form data is invalid.

  1. The system will notify the user of any invalid data.
  2. The user will repeat step 1.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 12 and Dynamic UML Diagrams, Figure 30a-30b.

User Story #1316 - Add Delete Options To Categories

- As an user, I would like to be able to delete a BOLO category in the category management tab.

Acceptance Criteria:

- If category is being used by any BOLO, it will not be deleted.
- Proper error/success message must be shown when deleting category.
- Add a Delete Category button in the Category Management tab.
- Categories must be deleted from database after user clicked on Delete button.

Use Case

- Name: Add Delete Option to Categories
- Actor: User
- Preconditions:

- User must be log in.
- User must be

- Description:

1- User clicks on category.

2- System loads edit category page.

3- User clicks on Delete Category button.

4- System searches if category is being used by a BOLO. If it is, reports error message, else deletes category and reports success message.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 13 and Dynamic UML Diagrams, Figure 31.

User Story #1320 - Edit login page fix bugs

- As a root admin, I would be able to edit the logo, title or subtitle individually

Acceptance Criteria

- The root admin can edit just the logo of the login page
- The root admin can edit just the title of the login page
- The root admin can edit just the subtitle of the login page
- The root admin can edit any feature in the login page and not all of them at the same time

Use Case

- Name:ID # 1323 Edit Logo or Title or Subtitle of LogIn Page
- Actor:Root
- Preconditions:

    - The user should has root access
    - The root should be Logged In
    - The root should get into the Admin menu
- Description:
    - The root should log In
    - The root should get into the Admin menu
    - The root should click on the Edit Log In Page
    - The root should change the title, subtitle, logo or any combination of them
    - The root should save the changes by clicking in the SAVE button

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 14 and Dynamic UML Diagrams, Figure 32.

Sprint 6

User Story #1223 - BOLO Filter not Working

- As an user, I would like to filter archived BOLOs by agency, BOLOs created by me, and all BOLOs

Acceptance Criteria

- Must show archived BOLOs once the filter is changed according to filter criteria.
- By default the filter should show all BOLOs

Use Case

- Name: ID #1373 The filter Bolos' archived should filter all archived bolos by the agency's user, the user's bolos and and all archived bolos in the system
- Actor: Any user
- Preconditions:
    - The user should be logged in

- The user should click in the Archive button

- Description :
  - The user should be logged in
  - The user should click in the Archive button
  - The user should choose any criteria to filter the list of archived bolos

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 15 and Dynamic UML Diagrams, Figure 33.

User Story #1328 - Add internal option to create BOLO form

- As a User, I would like the option for a BOLO to only be available within the agency.

Acceptance Criteria

- If the internal option is selected, the BOLO is only available within the agency.
- If the internal option is not selected, the BOLO is available in all agencies.

Use Case

- Name: Creating a New BOLO
- Actors
  - Administrator
  - Officer
  - Supervisor
  - Root
- Preconditions:
  - The user has BOLO categories created.
  - The user invokes the option to create a new BOLO.

- Description:
  1. The user will fill out the form and submit.
  2. The system will validate the form and create a new BOLO.
- Exceptions

  1A: The user wants to preview the BOLO before submitting

  1. The user will select the option to preview.
  2. The system will validate the form and create a preview.
  3. The user will select the option to go back and repeat step 1.

  2A: The system finds an error in the form.

  4. The system will notify the user of invalid data.
  5. The user will correct the error in the form.
  6. The use case will repeat step 2.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 16 and Dynamic UML Diagrams, Figure 34.

User Story #1336 - Reduce size of Officer's badge panel

- As an admin/root, I would like that the Officer's badge panel was of the same size as the email's panel.

Acceptance Criteria

- Officer's badge panel must be of same size as the email's panel.

Use Case

- Name: Reduce size of Officer's badge panel
- Actor: Admin/Root
- Preconditions:
  - Must be signed in.

- Must be creating/editing an user

- Description:

1- User fills badge. User submits.

2- System validates data and saves it on database.

- Exceptions:

2- User did not enter username. System shows correspondent message.

2- User did not enter email. System shows correspondent message.

2- User email is incorrect. System shows correspondent message.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 17 and Dynamic UML Diagrams, Figure 35.

User Story #1337 - Change Date to US Standard

- As an user, I would like the date while creating/updating a BOLO to be in US standard format.

Acceptance Criteria

- Date must appear as MM/DD/YYYY
- Must be validated

Use Case

- Name: Change Date to US Standard
- Actor: User
- Preconditions:
  - Must be signed in.
  - Must be creating/validating a BOLO

- Description:

  1- User types date.

  2- System processes date after user clicks on other panel.

  3- User submits BOLO.

- Exceptions:

  2- If date is in wrong format or invalid, system shows correspondent message.

  3- User cannot create BOLO with incorrect date, system shows correspondent message.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 18 and Dynamic UML Diagrams, Figure 36.

User Story #1338 - Add a purge date to delete archive BOLOs

- As a User, I would like to be able to delete archived BOLOs within a given date range.

Acceptance Criteria

- The User must be able to select a date range in order to delete archived BOLOs.
- The User must be given the option to delete all archived BOLOs.
- The User must confirm that archived BOLOs set for deletion are correct.

Use Case

- Name: Purge Archived BOLOs
- Actors
    - Root
- Preconditions:
    - The user has a root account.

    - There are several BOLOs already archived.

- Description:
    1. The user will select the date range to purge archived BOLOs and submit.
    2. The system will display all BOLOs within the given date range.
    3. The user will enter his/her password to confirm that the BOLOs to delete are correct.
    4. The system will delete all archived BOLOs within the date range.
- Exceptions

  3A: The user enters an incorrect password.

    1. The system will notify the user that an invalid password was entered.
    2. The user will repeat step 3.

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 19 and Dynamic UML Diagrams, Figure 37.

User Story #1357 - Impede users from navigating page after entering temporary password

- As an user, I would like that when a new user logs in for the first time with its temporal password, it cannot traverse to other tabs after creating a new password.

Acceptance Criteria

- After user logs in with temporal password, he/she cannot traverse to other tabs before it completes creating its password.
- Button to go back must be added

Use Case

- Name: Impede new users from navigating page after entering temporary password
- Actor: User

- Preconditions:
    - User must log for the first time with temporal password
    - User must be at password reset page
- Description :

  1- User logs in with temporary password

  2- System redirects user to Password Reset page

  3- User creates fills password and repeat password panels and presses submit.

- Exceptions :

  3- If password and confirmation does not match, system warns user with appropriate message.

  3- If password is in incorrect format, system warns user with appropriate message

Modeling

Refer to UML diagrams in Appendix A: Static UML Diagrams, Figure 20 and Dynamic UML Diagrams, Figure 38.

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. To conclude, the design patterns used in the project are discussed.

Architectural Patterns

BOLO-7.0 uses a combination of the three-tier and the Model-View-Controller (MVC) architecture.

The entire system is divided into three independent modules, also known as tiers. The database tier includes the MongoDB server, which is where the application's data is kept. At the application tier, the Cloud Foundry server and all programs that access the database reside. End-users operate at the user tier. How the user views the data stored in the database tier is generated by programs that reside in the application tier.

Inside the application tier is where the MVC architectural pattern exists. The application is separated into three components: Model, View, and Controller. In the model layer, data, logic, and the rules of the BOLO application are kept. The view layer is responsible for displaying data for end-users based on changes to the model. The controller layer is where input is accepted from the user and converted into commands for the model or view layer.

By choosing these architectural patterns, it was possible to create a secure, testable, and scalable system. It also made it practical to add new features to the system without having to update the core functionality.

System and Subsystem Decomposition

BOLO-7.0 is divided into six major subsystems: agencies, authentication, edit, users, bolos. All subsystems are composed by views, a router, a service, and a repository. The proper functionality of the system relies on the interaction among all 6 subsystems:

Agency Controller: It handles agency creation, updating, deactivation, and deletion. Without an agency, users can not exist in the system.

Authentication Controller: It handles the login cookies of different users accessing the system. Every call to the server will pass though this controller to ensure that the user is currently logged in. It also handles password expiration.

Bolo Controller: Being the most extensive one, it manages all bolo-related functionality such as BOLO creation, previewing, search, PDF, email, deletion, archiving and others.

Category Controller: It handles the categories and form data for new bolos.

Edit Controller: It allows the root user to edit the website. Editable pages include the user guide, the login page, and the about us page.

User Controller: It handles user accounts and user management. It allows users to be created, edited, and deleted. It also oversees account functionality such as passwords, login, and logout.

Deployment Diagram

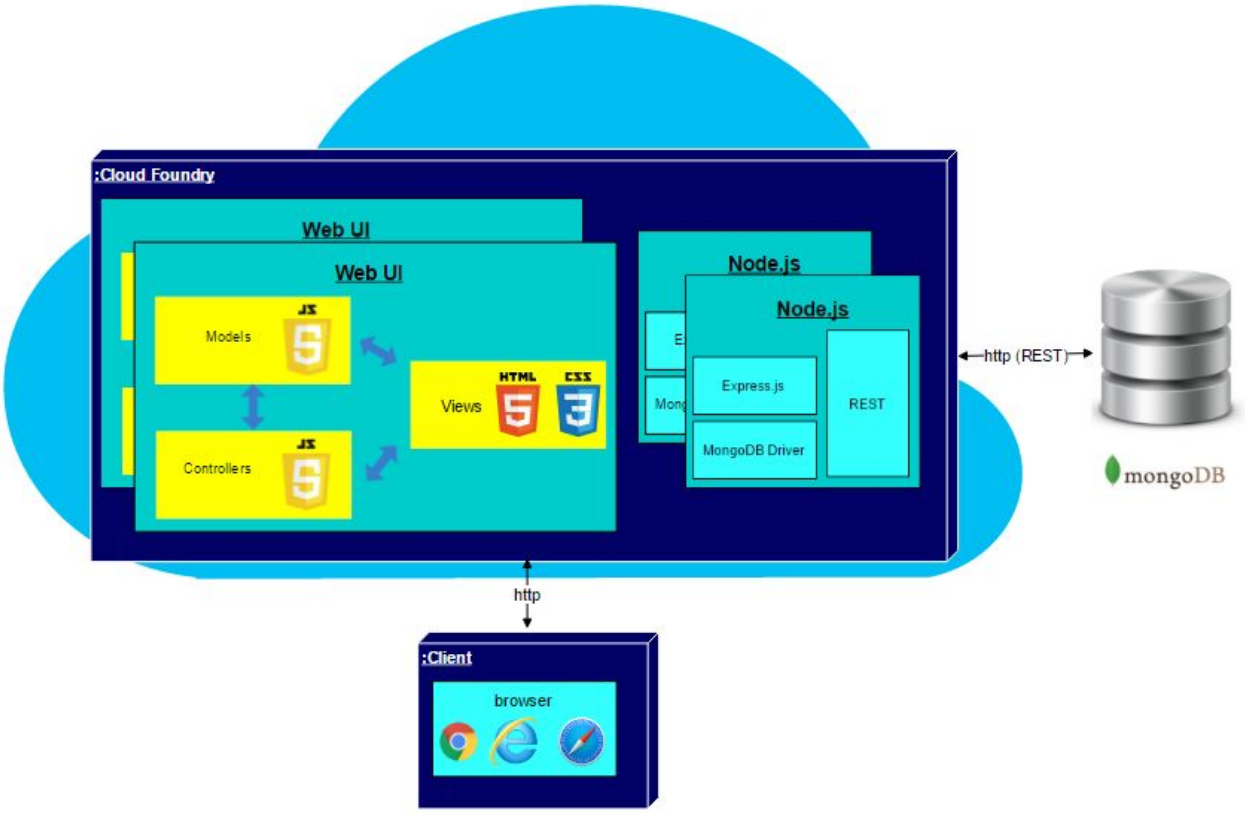


Figure 1: The deployment diagram for the BOLO-7.0 system uses the three-tier and MVC architectures.

Design Patterns

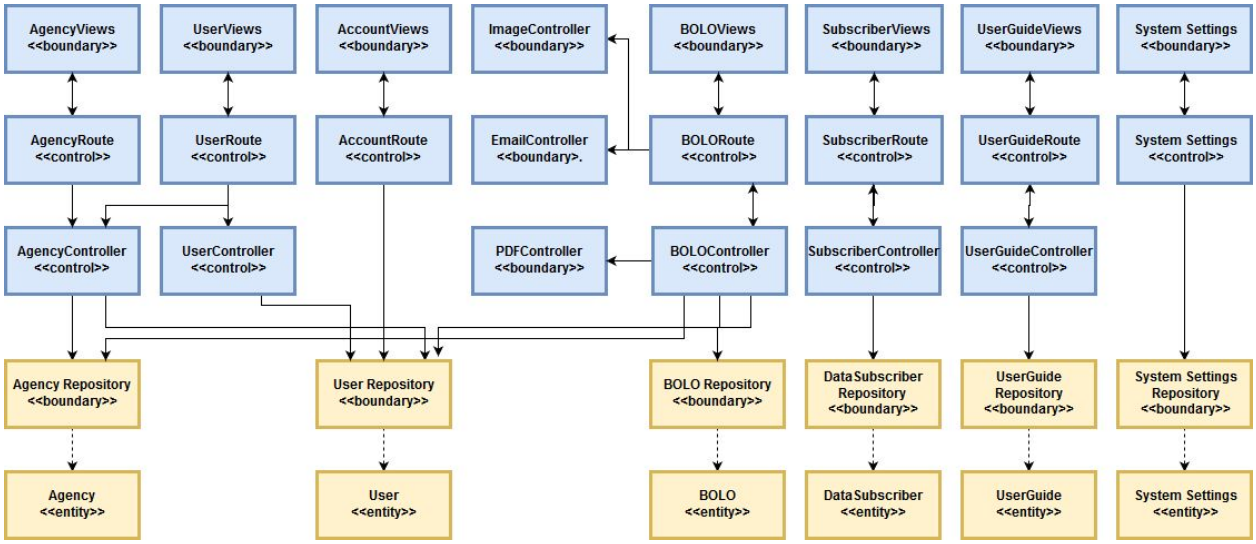


Figure 2: The minimal class diagram represents the static view of the BOLO application. It is not only used to visualizing and describing aspects of the system but also for constructing executable code of the software application.

System Validation

Unit tests are essential to test the functionality within the application. All test cases in this section were used to validate the code for each User Story.

User Story #1215 Setup the BOLO-6.0 environment

Test ID: BOLO_Setup_1
Purpose: To test that the BOLO system is functioning on the my local server.
Preconditions: MongoDB server and Node JS are all running.
Input: Navigate to http://localhost:3000 Login to the root account Create a new user
Expected Output: The login page is displayed The bolo page is displayed An email is sent

User Story #1217 Fix BOLO PDF

Test ID: BOLO_PDF_1
Purpose: Take out N/A attributes while in details
Preconditions: User must be signed in. User must be looking at a BOLO's details User must be at archive or at home There must be at least one BOLO archived or active
Input: User clicks on details
Expected Output: System renders BOLO Details page without N/A attributes

Test ID: BOLO_PDF_2
Purpose: Take out N/A attributes while creating BOLO
Preconditions: User must be signed in User must be creating a BOLO
Input: User clicks on details
Expected Output: System renders BOLO preview page without N/A attributes

Test ID: BOLO_PDF_3
Purpose: Take out attributes in PDF generation
Preconditions: User must be signed in User must be looking at a BOLO's details
Input: User clicks on generate pdf
Expected Output: System generates pdf without N/A attributes

User Story #1218 Search Issue in the system

Test ID: BOLO_Search_1
Purpose: Search BOLO should works even if not input are provided

Preconditions: User must be logged In There should be any BOLO archived or active User should click of the Search button in the menu
Input: User click on Search button on the menu User click on Search button without choosing any agency or category
Expected Output: Display all BOLOs actives and archived

Test ID: BOLO_Search_2
Purpose: Search BOLOs by category and agency should work combined
Preconditions: User must be logged In There should be any BOLO archived or active User should click of the Search button in the menu
Input: User click on Search button on the menu User should set any agency User should set any category and any token on the given category
Expected Output: Display all BOLOs that match with the given agency and category

User Story #1222 Password Validation

Test ID: BOLO_PasswordValidation_1
Purpose: Not writing a password and pressing Submit button should not be accepted

Preconditions: Must be creating new password
Input: Not writing a password and pressing Submit button when creating new password
Expected Output: Password Format Is Not Valid!!! warning

Test ID: BOLO_PasswordValidation_2
Purpose: Writing a valid password
Preconditions: Must be changing password
Input: Input password and repeat password
Expected Output: Password Must Match!!! Warning (if passwords do not match)

User Story #1246 Prevent server from crashing when creating a new BOLO without selecting a category

Test ID: BOLO_CreateBolo_1
Purpose: To verify that a BOLO is not being created if the user does not select a category
Preconditions: The user already has an account The list of categories is not empty. The user has selected the option to Create a BOLO.
Input: Date Reported: 10/2/2017 Time Reported: 13:00

Expected Output: The following error message appears: “Please select a category”

Test ID: BOLO_CreateBolo_2
Purpose: To verify that a BOLO is not being created if the user does not select a category
Preconditions: The user already has an account The list of categories is not empty. The user has selected the option to Create a BOLO.
Input: Date Reported: 10/2/2017 Time Reported: 13:00 Category: Boat Theft
Expected Output: An email to confirm the BOLO is sent to the user.

User Story #1219 Prevent BOLO from disappearing when updated

Test ID: BOLO_UpdateBolo_1
Purpose: To verify that the original BOLO is still visible on the Bolo page if it has been updated and not confirmed.
Preconditions: The user already has an account. The user already created and confirmed a BOLO.
Input: Change Date Reported to “23/2/2017” Change Year to “2012”

Change License Plate to “TEST00”
Expected Output: An email to confirm the BOLO was received. The details of the BOLO displayed reflects the new changes. The title “Updated” is visible.

Test ID: BOLO_UpdateBolo_2
Purpose: To verify that the Edit BOLO form stores previously entered data.
Preconditions: The user already has an account. The user already created and confirmed a BOLO.
Input: Delete the entry for Date Reported Change Zip Code to “55555”
Expected Output: The form is not submitted. An error message is displayed: “Please enter a date”. The form saved the value for the zip code

Test ID: BOLO_UpdateBolo_3
Purpose: To verify that conflicting updates are handled appropriately.
Preconditions: The user already has an account. The user has not confirmed a BOLO. The user is editing the same BOLO that was not confirmed.
Input:

Change the status from ACTIVE to RESOLVED.
Expected Output: First confirmation email: The BOLO is updated with status FOUND. Second confirmation email: The following error message is displayed: “Bolo to confirm was not found on the database”. The BOLO was not updated with the status RESOLVED

User Story #1221 Setup the bolo.cs.fiu.edu server

Test ID: BOLO_Setup_2
Purpose: To test that the BOLO system is functioning on the dedicated production VM.
Preconditions: MongoDB server, Node JS, and Apache2 are all running on bolo.cs.fiu.edu.
Input: Navigate to http://bolo.cs.fiu.edu. Log in to the root account. Select “Admin”. Select “User Management”. Add a new user. Fill out the form and submit.
Expected Output: Login page is displayed, BOLO home page is displayed, an email is received

User Story #1361 Create Logo path

Test ID: BOLO_EditLogin_1
Purpose: The logo image should be get from any directory by the root

Preconditions: The user should has root access The root should be Logged In The root should get into the Admin menu
Input: The root should log In The root should get into the Admin menu The root should click on the Edit Log In Page The root should change the logo of the app by uploading the image from any directory The root should save the changes by clicking in the SAVE button
Expected Output: The root should change the logo in the login page

User Story #1265 Subscribing to agency

Test ID: BOLO_Notifications_1
Purpose: Clicking on Notifications to be redirected to the Unsubscription page.
Preconditions: Must be signed in. Must be at account.
Input: Click on Notification
Expected Output: Should load the Unsubscription page and show subscribed agencies, if any.

Test ID: BOLO_Notifications_2
Purpose:

Clicking on More Agencies to be redirected to the Subscription page.
Preconditions: Must be signed in. Must be at account.
Input: Click on More Agencies
Expected Output: Should load the Subscription page and show unsubscribed agencies, if any.

Test ID: BOLO_Notifications_3
Purpose: Clicking on Submit after marking some agencies should show same page but updated.
Preconditions: Must be signed in. Must be at account.
Input: Clicking on Submit
Expected Output: Should load the Subscription page and show unsubscribed agencies, if any.

User Story #1287 Edit User Guide

Test ID: BOLO_EditUserGuide_1
Purpose: To verify that a new section could be added to the User Guide.
Preconditions:

The user already has a root account The user has invoked the option to add a new section.
Input: Level of Access: ROOT Section Heading: Data Analysis Instructions: 1) Click on “Admin”. 2) Click on “Data Analysis”. 3) Select one or more agencies. 4) Click “Download”.
Expected Output: The section is added to the section table. The section is added to the Root User Guide.

Test ID: BOLO_UserGuide_2
Purpose: To verify that a section could be edited in the User Guide.
Preconditions: The user already has a root account The user has selected to edit the “Agency Directory” section of the User Guide.
Input: Change the Level of Access from “ROOT” to “ROOT, ADMINISTRATOR” Select “Save”.
Expected Output: The section is updated in the section table The section is added to both the the Root and Administrator User Guides.

Test ID: BOLO_UserGuide_3

Purpose: To verify a section could be deleted from the User Guide.
Preconditions: The user already has a root account. The user has selected the option to edit the User Guide.
Input: Click on “Agency Directory” in the section table. Click on “Delete”.
Expected Output: The section is removed from the section table The section is removed from both the the Root and Administrator User Guides.

Test ID: BOLO_UserGuide_4
Purpose: To verify that the title could be edited in the User Guide.
Preconditions: The user already has a root account. The user has selected the option to edit the User Guide.
Input: Change the title from “Welcome to the User Guide” to “Welcome to the BOLO User Guide”. Click on “Save”
Expected Output: The title is updated in the Root, Administrator, Supervisor, and Officer User Guides.

Test ID: BOLO_UserGuide_5
Purpose: To verify that form validation is working in the Edit User Guide forms.

Preconditions: The user already has a root account. The user has selected the option edit the User Guide.
Input: Delete the title. Select “Save”
Expected Output: The following error message is displayed: “A title for the User Guide is required” The title of the User Guide is not updated.

Test ID: BOLO_UserGuide_6
Purpose: To verify that form validation is working in the Edit User Guide forms.
Preconditions: The user already has a root account. The user has selected the option to add a new section to the User Guide.
Input: Select “Save”
Expected Output: The following error messages are displayed: ”Title is required” and “Instructions are required”. An empty section was not added to the User Guide

User Story #1286 Notifying users subscribed to agencies

Test ID: BOLO_Notifications_4

Purpose: Create a BOLO and click confirmation email to check if users subscribed receive notification emails.
Preconditions: User must be log in. At least one user must be subscribed to the BOLO's agency.
Input: User receives confirmation email and clicks confirmation link. System receives confirmation. System send a notification email to the BOLO author and to any user that is subscribed to the BOLO or to the BOLO's agency. Users receive notification email
Expected Output: Notifications emails are received for each subscribed user plus the author.

Test ID: BOLO_Notifications_5
Purpose: Update a BOLO and click confirmation email to check if users subscribed receive notification emails.
Preconditions: User must be log in. At least one user must be subscribed to the BOLO's agency.
Input: User receives confirmation email and clicks confirmation link. System receives confirmation. System send a notification email to the BOLO author and to any user that is subscribed to the BOLO or to the BOLO's agency. Users receive notification email
Expected Output: Notifications emails are received for each subscribed user plus the author.

User Story #1288 Edit Login Page

Test ID: BOLO_EditLogin_2
Purpose: Edit Title and Subtitle of LogIn Page
Preconditions: The user should has root access The root should be Logged In The root should get into the Admin menu
Input: The root should log In The root should get into the Admin menu The root should click on the Edit Log In Page The root should change the title and subtitle by enter them in the given text fields The root should save the changes by clicking in the SAVE button
Expected Output: The root should be able to edit the Title and Subtitle of the login page

Test ID: BOLO_EditLogin_3
Purpose: Edit Logo of Login Page
Preconditions: The user should has root access The root should be Logged In The root should get into the Admin menu

Input: The root should LogIn The root should get into the Admin menu The root should click on the Edit Log In Page The root should upload the image for logo and by clicking the change logo button The root should save the changes by clicking in the SAVE button
Expected Output: The root should be able to change the logo of the login page

User Story #1315 Forgot password

Test ID: BOLO_ResetPassword_1
Purpose: To verify that the user could reset their password if it is forgotten.
Preconditions: The user already has an account.
Input: Select “Forgot Password”. Enter the email, first name, and last name used in the registered account. Select “Reset Password”. Click the link in the Reset Password email. Enter a new password and submit.
Expected Output: The following message is displayed: “Reset information successfully sent” The user receives an email with instructions to reset their password.

Test ID: BOLO_ResetPassword_2
Purpose: To verify that the error messages are displayed in the Forgot Password form.
Preconditions: The user already has an account.
Input: Select “Forgot Password”. Enter the following and select “Reset Password”: ● Invalid email, valid first name, and valid last name ● Valid email, invalid first name, and invalid last name ● Valid email, missing first name, and missing last name. Select “Reset Password”.
Expected Output: If invalid email, the following message is displayed: “Email address is not registered” If invalid name and invalid last name is entered, the following message is displayed: “Credentials do not match”. If missing first name and last name, the following messages are displayed: “First name is required” and “Last name is required”.

User Story #1316 Add Delete Option to Categories

Test ID: BOLO_EditCategory_1
Purpose: Clicking on delete button on a category that is being used by a BOLO.
Preconditions: Must be signed in. Must be at Edit Category page.
Input:

User clicks on category. System loads edit category page. User clicks on Delete Category button. System searches if category is being used by a BOLO. If it is, reports error message, else deletes category and reports success message.
Expected Output: Category is not deleted and error message appears on top

Test ID: BOLO_EditCategory_2
Purpose: Clicking on delete button on a category that is not being used by a BOLO
Preconditions: Must be signed in. Must be at Edit Category page.
Input: User clicks on category. System loads edit category page. User clicks on Delete Category button. System searches if category is being used by a BOLO. If it is, reports error message, else deletes category and reports success message.
Expected Output: Category is deleted and success message appears on top

User Story #1320 Edit Login Page Fix Bugs

Test ID: BOLO_EditLogin_4
Purpose:

Edit Logo or Title or Subtitle of LogIn Page
Preconditions: The user should has root access The root should be Logged In The root should get into the Admin menu
Input: The root should log In The root should get into the Admin menu The root should click on the Edit Log In Page The root should change the title, subtitle, logo or any combination of them The root should save the changes by clicking in the SAVE button
Expected Output: The root should be able to edit the Title, Subtitle or logo or any combination of them of the login page

User Story #1328 Add Internal Option to Create BOLO Form

Test ID: BOLO_CreateBOLO_3
Purpose: To verify that the User could create an internal BOLO
Preconditions: The user already has an account. The list of categories is not empty. The user has selected the option to Create a BOLO.
Input: <Check> Internal Date Reported: 4/7/2017

Time Reported: 12:00 Category: Auto Theft
Expected Output: An email to confirm the BOLO is sent to the user. The title “Internal” is visible on the BOLO thumbnail.

User Story #1338 Add a Purge Date to Delete Archived BOLOs

Test ID: BOLO_DeleteBolo_1
Purpose: To verify that the User could delete all archived BOLOs.
Preconditions: The user has a root account. There are several BOLOs already archived. The user has selected the option to view archived BOLOs.
Input: Select “All Archived Bolos”. Select Purge. Enter password and select Delete.
Expected Output: All archived BOLOs are displayed when user selects “Purge” The following message is displayed: “All archived BOLOs have been deleted” after user confirms the page.

Test ID: BOLO_DeleteBolo_2
Purpose:

To verify that the User could delete all archived BOLOs within a date range.
Preconditions: The user has a root account. There are several BOLOs already archived. The user has selected the option to view archived BOLOs.
Input: Select “1 Year”. Select Purge. Enter password and select Delete.
Expected Output: All archived BOLOs older than one year are displayed when the user selects “Purge”. The following message is displayed: “All BOLOs older than one year has been deleted” after the page is confirmed.

User Story #1336 Reduce Size of Officer’s Badge Panel

Test ID: BOLO_EditUser_1
Purpose: Edit user badge as root to 4
Preconditions: Must be signed in as root
Input: User fills badge. User submits. System validates data and saves it on database.
Expected Output: User’s badge was changed. Success message appears on screen

User Story #1337 Change Date to US Standard

Test ID: BOLO_CreateBolo_4
Purpose: Typing 13/12/2017 as date to check if it is validated.
Preconditions: Must be signed in. Must be creating/validating a BOLO
Input: User types date. System processes date after user clicks on other panel. User submits BOLO.
Expected Output: After clicking on other option, the page should reflect a “Please enter a valid date (MM/DD/YYYY)” message

User Story #1357 Impede New User from Navigating Page After Entering Temporary Password

Test ID: BOLO_ResetPassword_3
Purpose: Create new user. Log in new user with temporary password.
Preconditions: User must log for the first time with temporary password User must be at password reset page
Input:

User logs in with temporary password System redirects user to Password Reset page User creates fills password and repeat password panels and presses submit.
Expected Output: Cannot move to other tab after creating password

GLOSSARY

Administrator: A user that has access to all options supervisor have plus it can create BOLO categories, agencies, and users.

Agency: A Police department.

BOLO: A "Be On the Lookout" flyer that contains information about a specific crime.

Officer: A basic user. Has access to basic functions such as creating, editing, and searching for BOLOs.

Supervisor: Same as officer, except it can archive BOLOs.

Root: A user that has access to all of the system. It can edit the login page, user guide, create agencies, etc.

APPENDIX

Appendix A - UML Diagrams

Static UML Diagrams

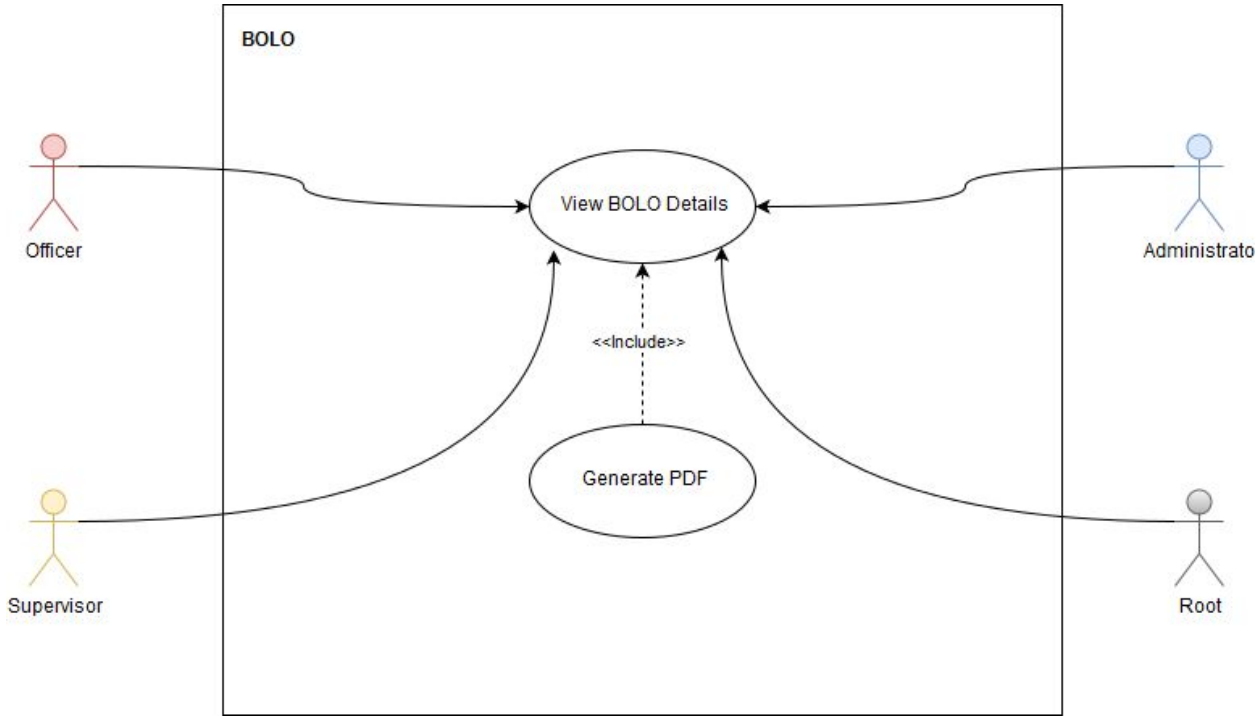


Figure 3a: UML diagram for Generate PDF, part of User Story #1217 - Fix BOLO PDF.

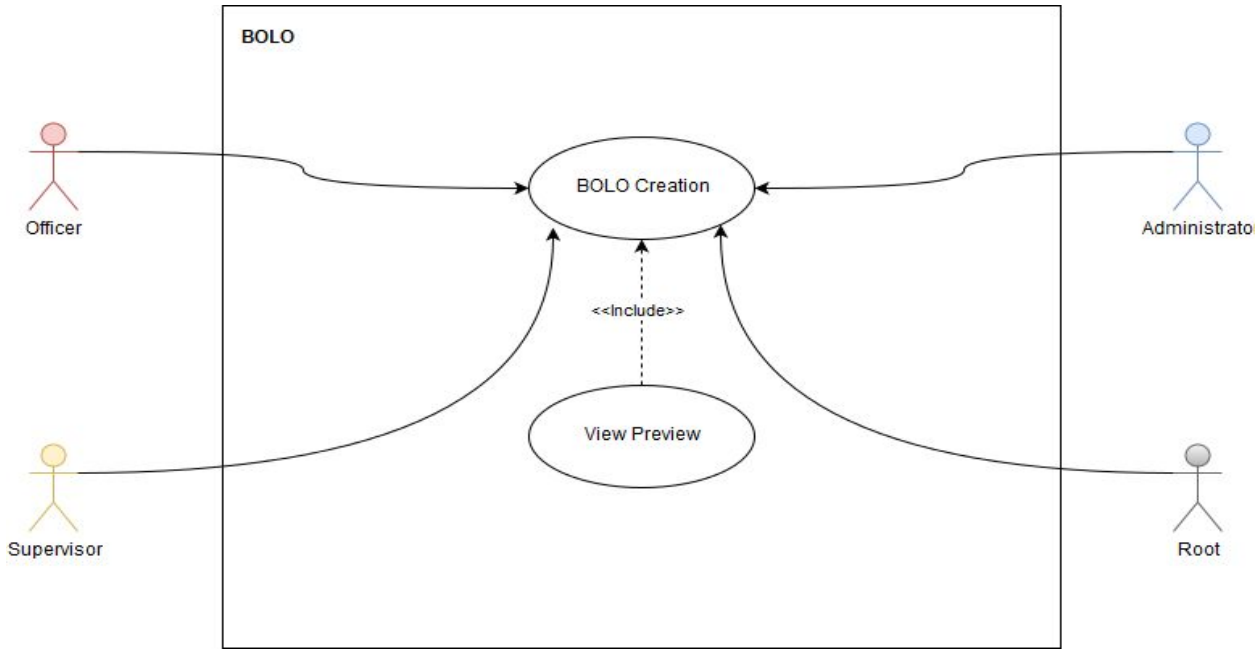


Figure 3b: UML diagram for View Preview, part of User Story #1217 - Fix BOLO PDF.

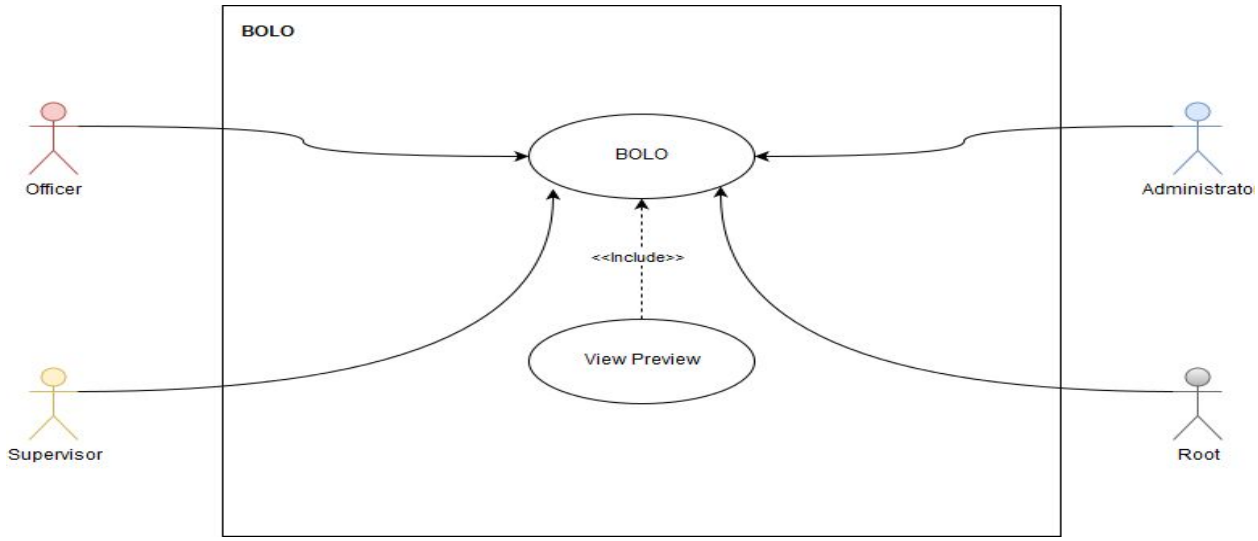


Figure 3c: UML diagram for View Preview, part of User Story #1217 - Fix BOLO PDF.

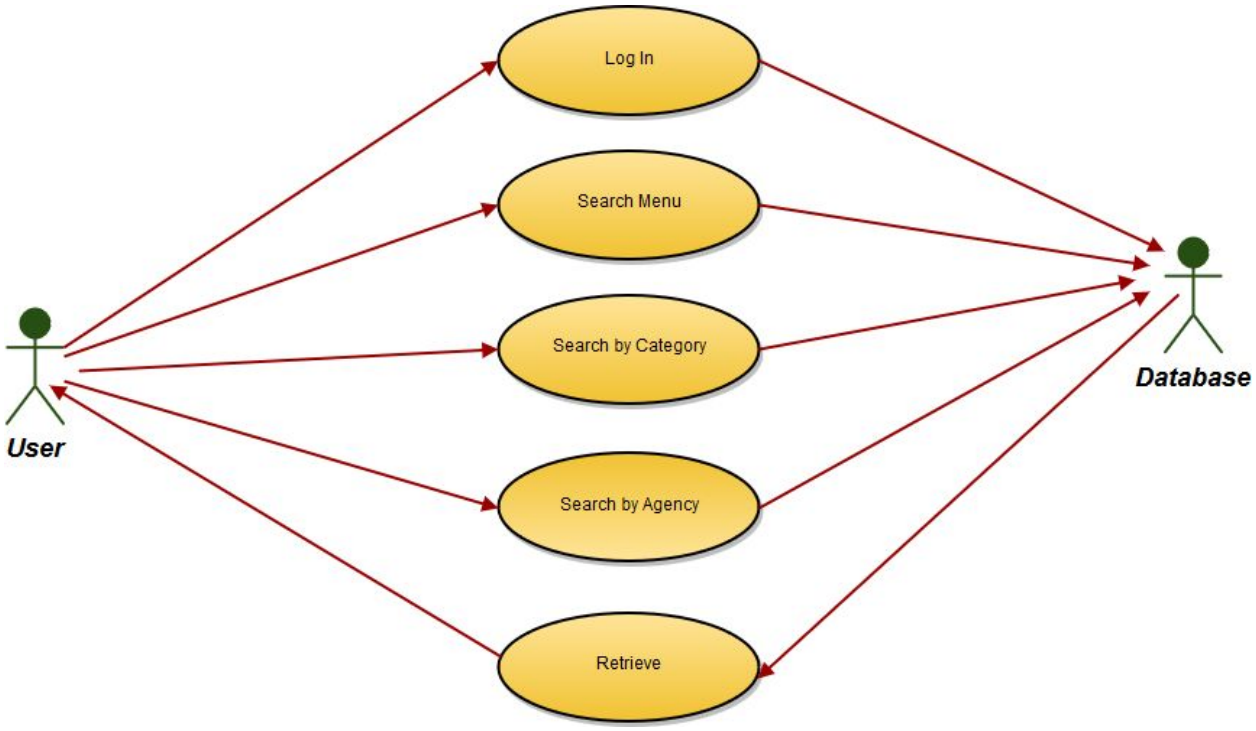


Figure 4: UML diagram for Searching BOLOs, part of User Story #1218 - Search Issue in the System.

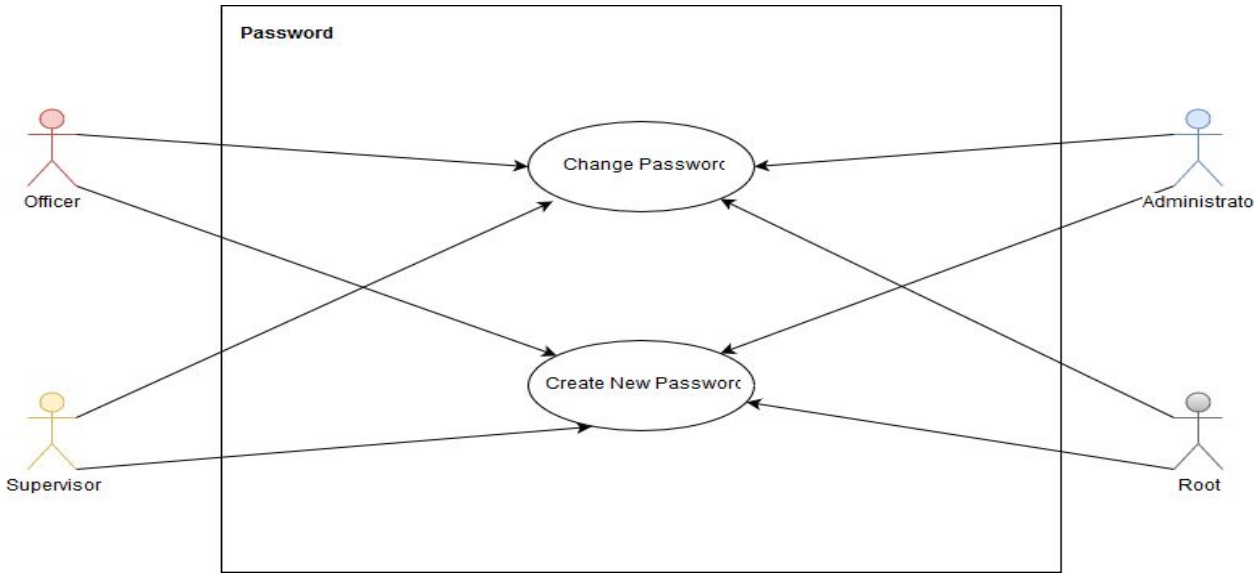


Figure 5: UML diagram for Resetting Password, part of User Story #1222 - Password Validation.

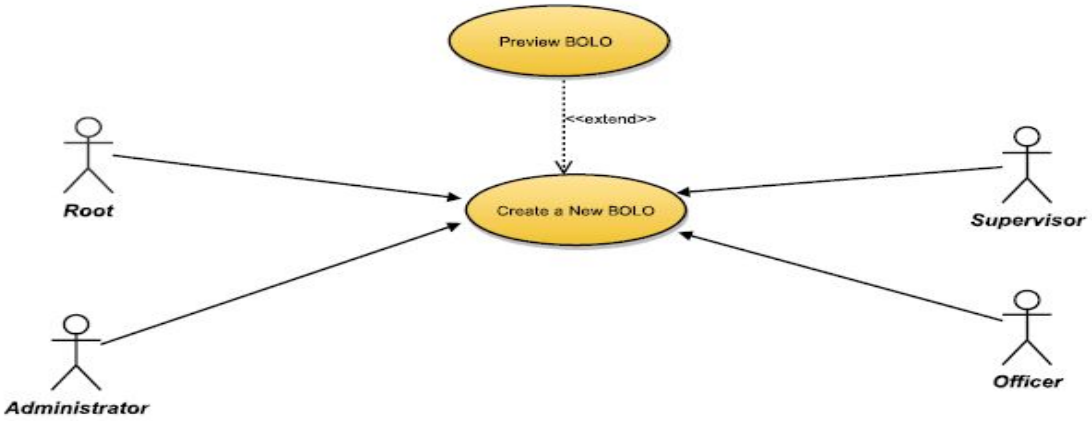


Figure 6: UML diagram for Creating a New BOLO, part of User Story #1246 - Prevent server from crashing when creating a new bolo without selecting a category

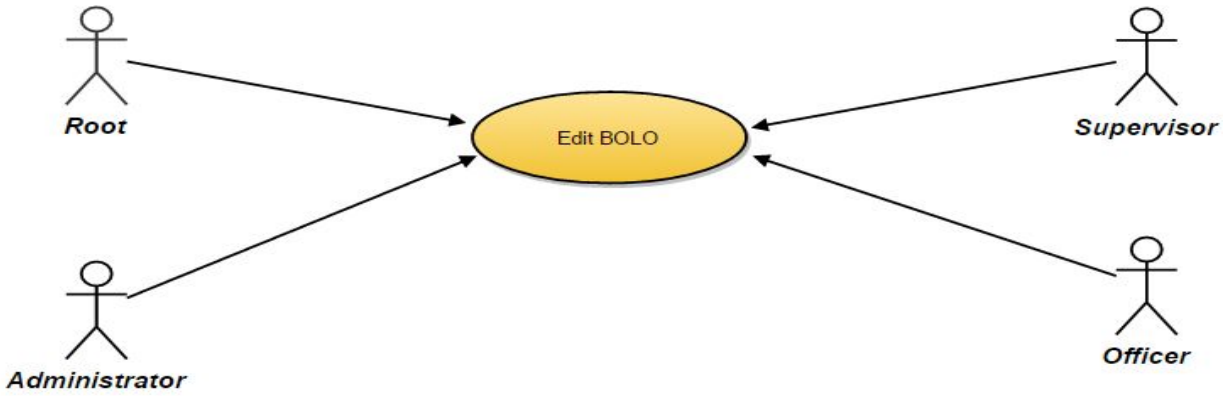


Figure 7: UML diagram for Editing a BOLO, part of User Story #1219 - Prevent BOLO from Disappearing when Updated.

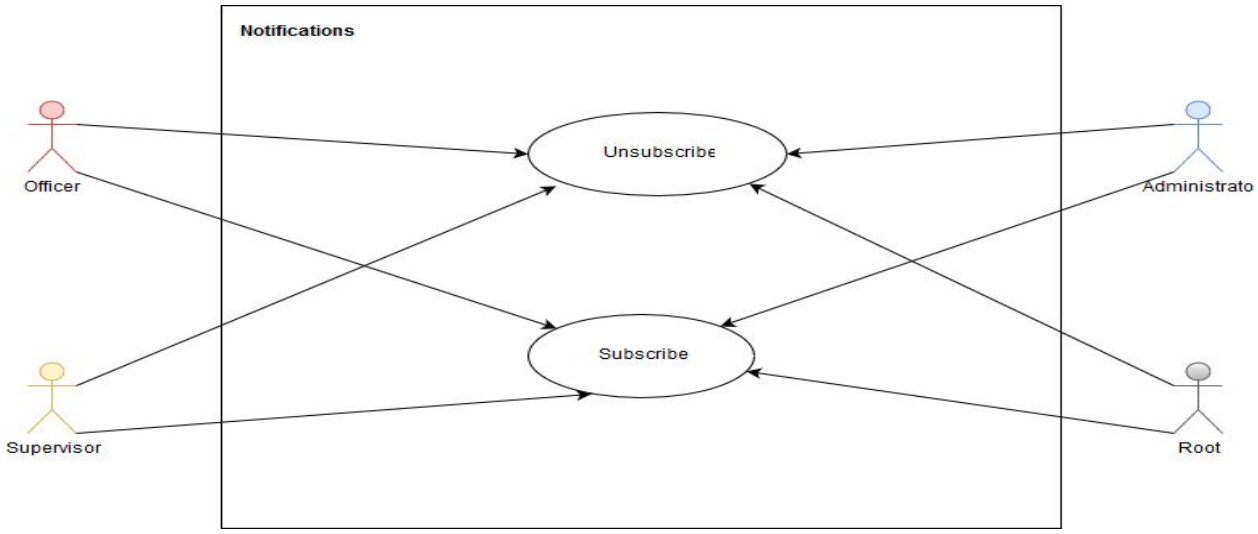


Figure 8: UML diagram for Subscribing to an Agency, part of User Story #1265 - Subscribing to Agency.

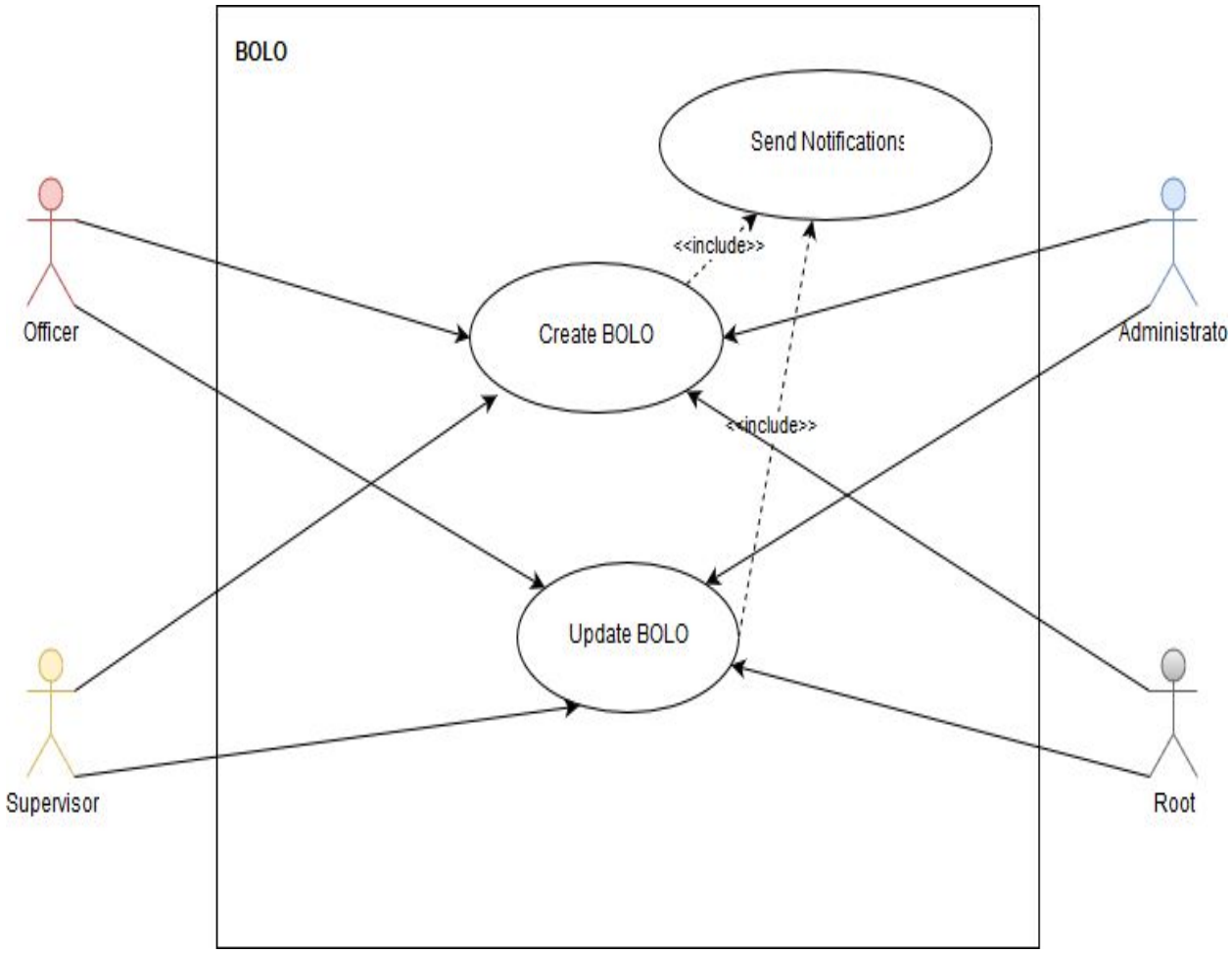


Figure 9: UML diagram for Notifying User Subscribed to an Agency, part of User Story #1286 - Notifying User Subscribed to Agencies.

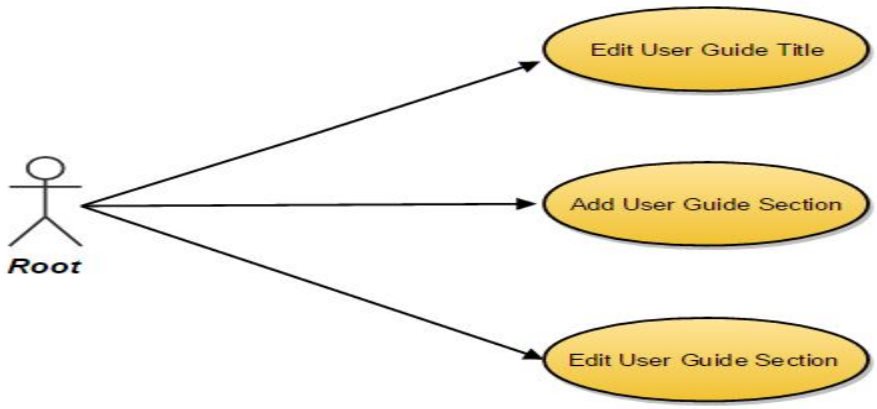


Figure 10: UML diagram for Editing the User Guide, part of User Story #1287 - Edit User Guide.

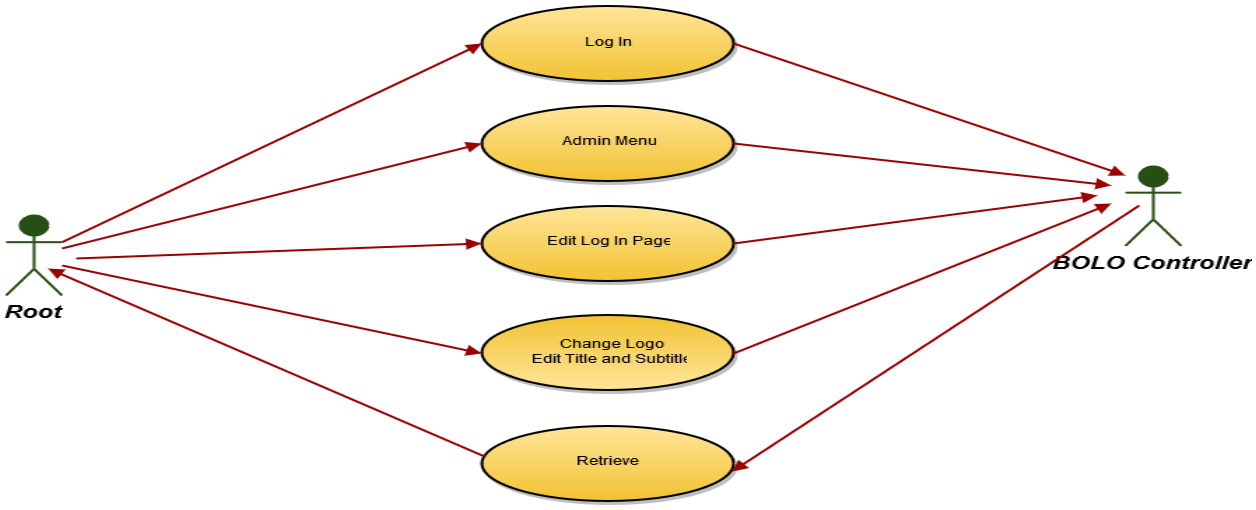


Figure 11: UML diagram for Editing the Login Page, part of User Story #1288 - Edit Login Page.

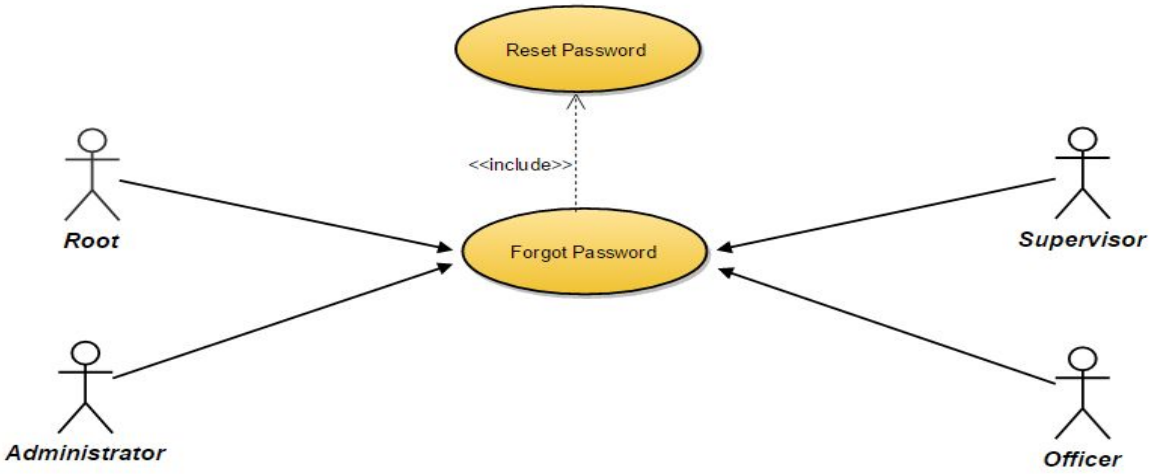


Figure 12: UML diagram for Resetting a Forgotten Password, part of User Story #1315 - Forgot Password

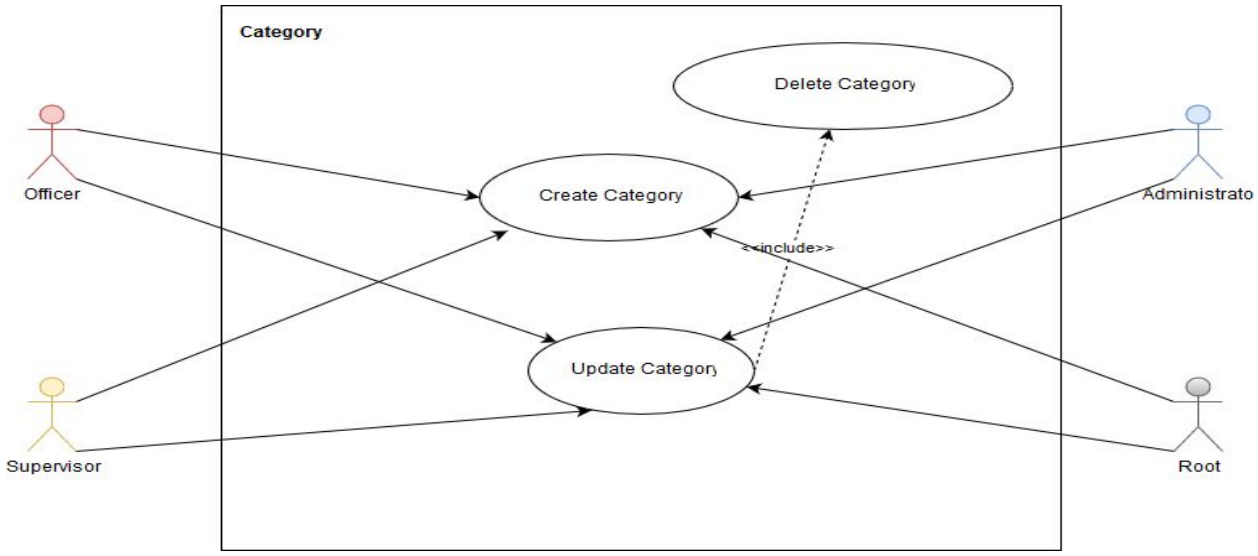


Figure 13: UML diagram for Deleting a Category, part of User Story #1316 - Add Delete Options To Categories

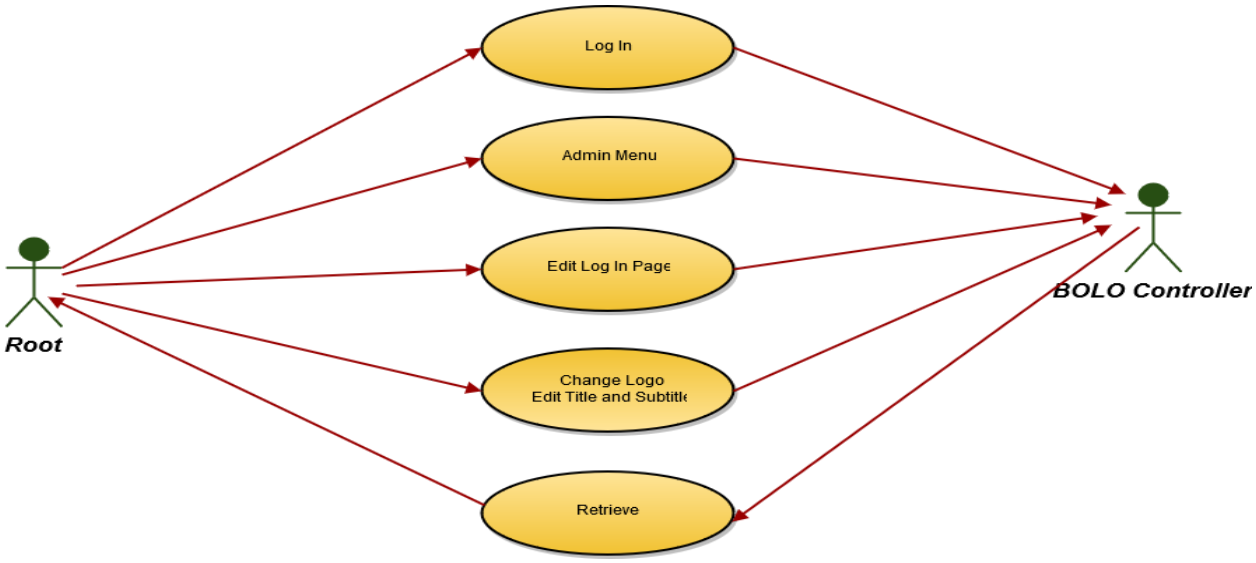


Figure 14: UML diagram for Editing the Login Page, part of User Story #1320 - Edit Login Page Fix Bugs

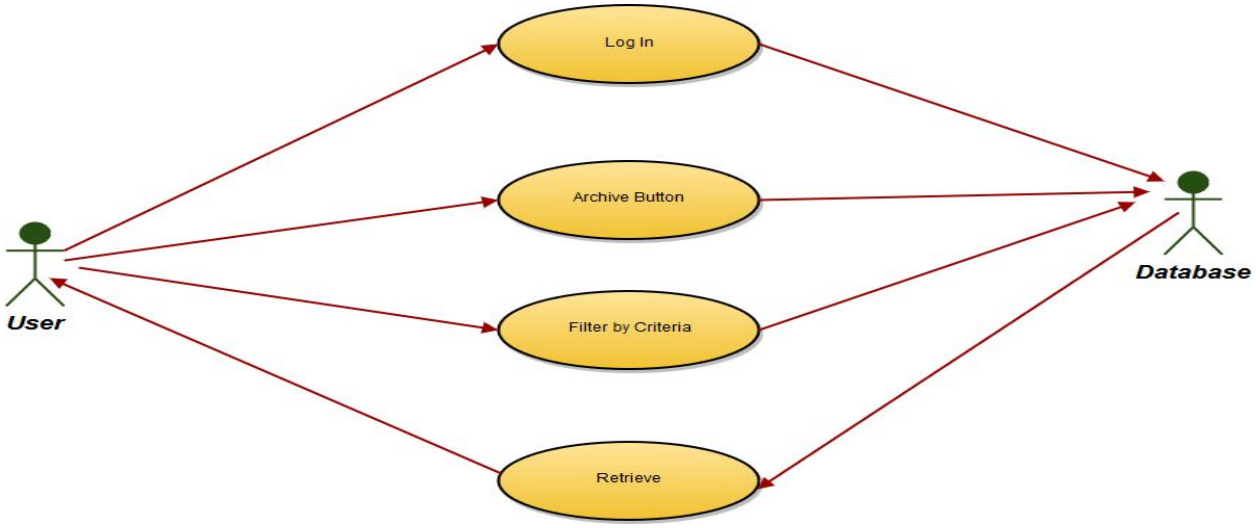


Figure 15: UML diagram for Filtering BOLOs, part of User Story #1223 - BOLO Filter not Working

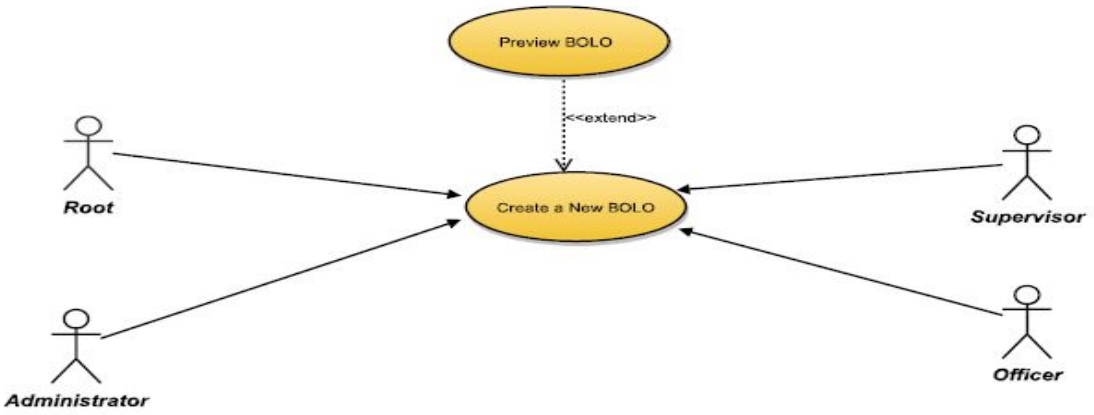


Figure 16: UML diagram for Creating a New BOLO, part of User Story #1328 - Add internal option to create BOLO form

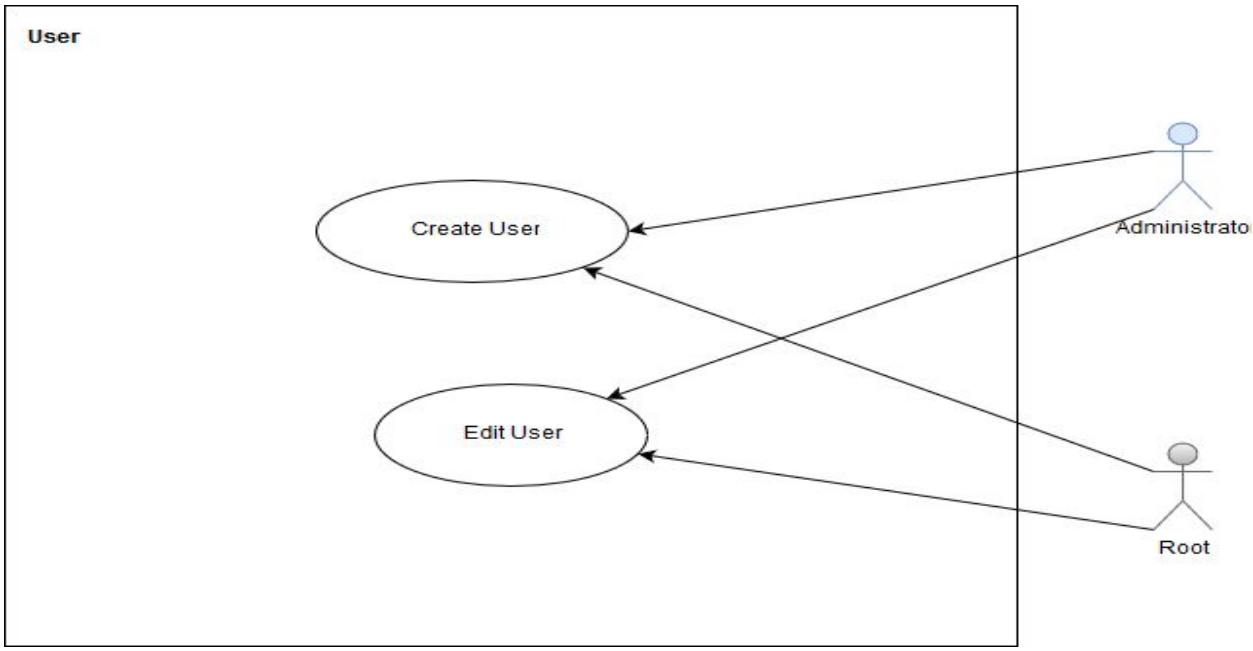


Figure 17: UML diagram for Creating a New User and Editing an Existing User, part of User Story #1336 - Reduce size of Officer's badge panel

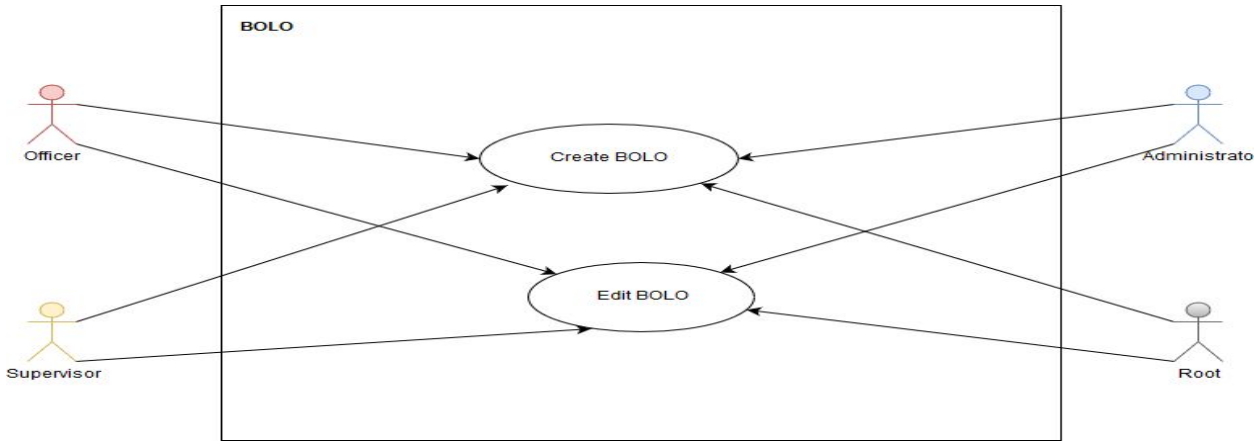


Figure 18: UML diagram for Creating a New BOLO and Editing an Existing BOLO, part of User Story #1337 - Change Date to US Standard

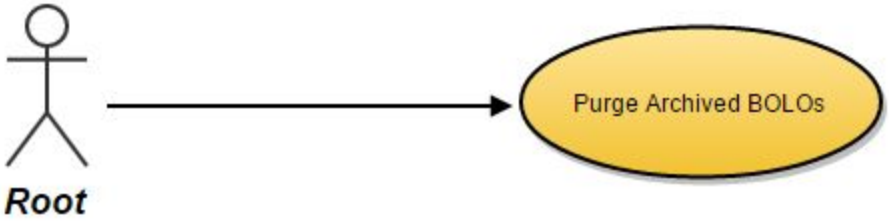


Figure 19: UML diagram for Deleting Archived BOLOs, part of User Story #1338 - Add a purge date to delete archive BOLOs.

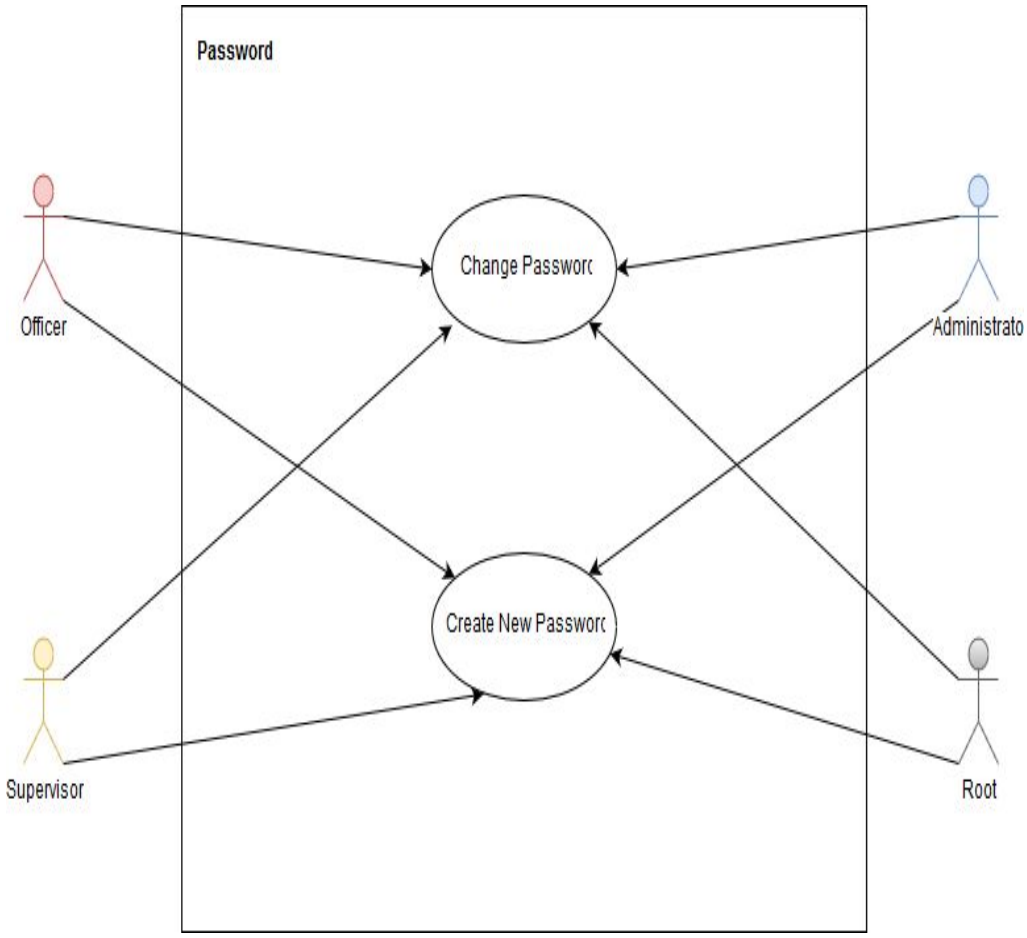


Figure 20: UML diagram for Resetting a Password, part of User Story #1357 - Impede users from navigating page after entering temporary password.

Dynamic UML Diagrams

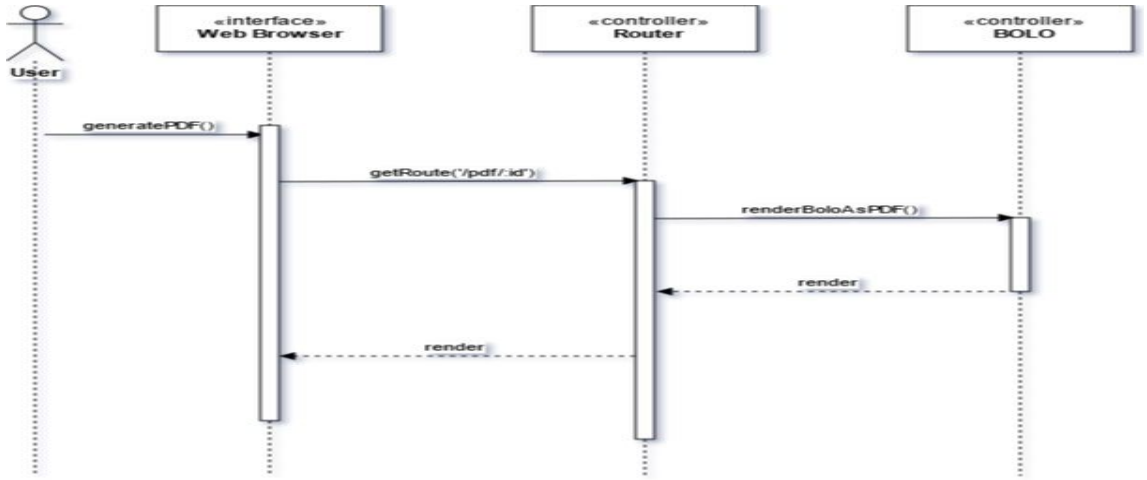


Figure 21a: Sequence diagram for Generating a PDF, part of User Story #1217 - Fix BOLO pdf

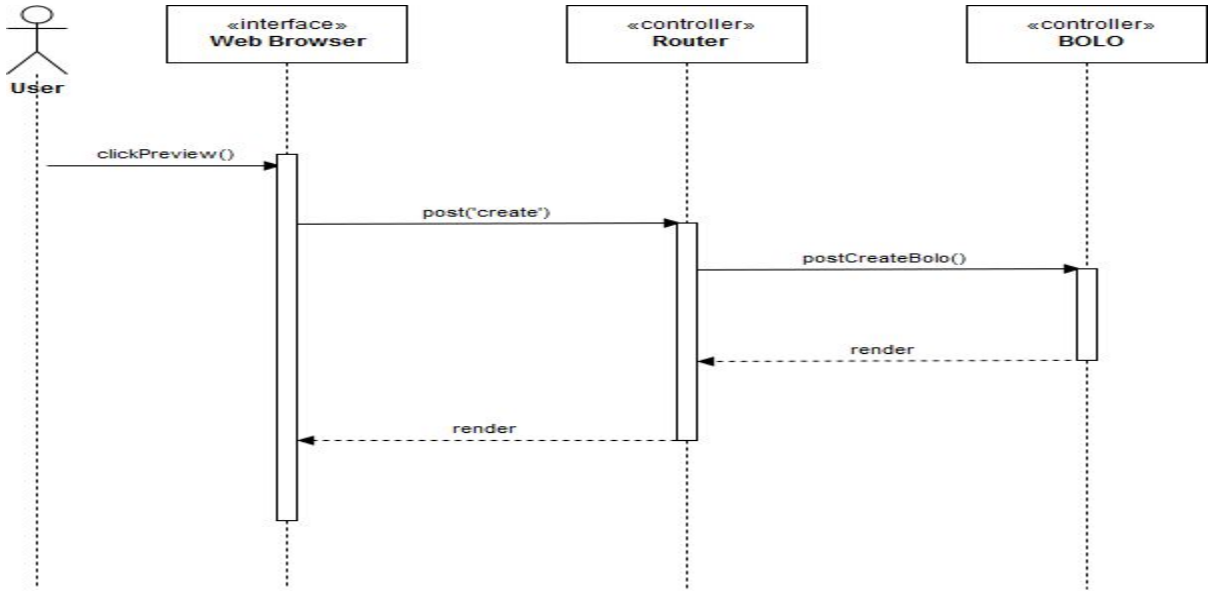


Figure 21b: Sequence diagram for Previewing a PDF, part of User Story #1217 - Fix BOLO pdf

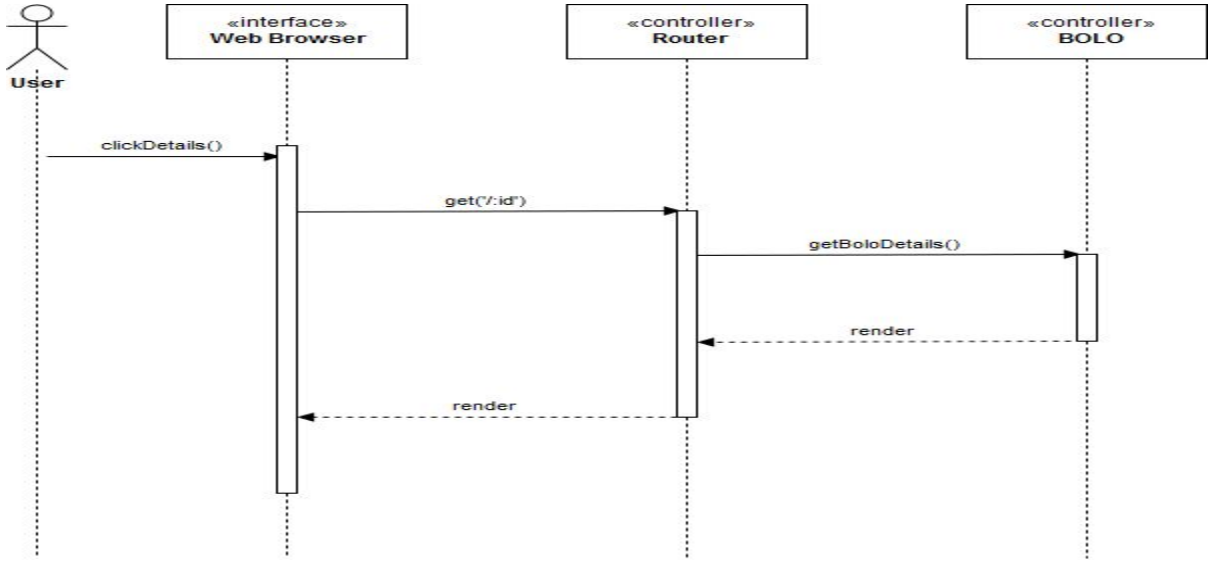


Figure 21c: Sequence diagram for Viewing the Details of a PDF, part of User Story #1217 - Fix BOLO pdf

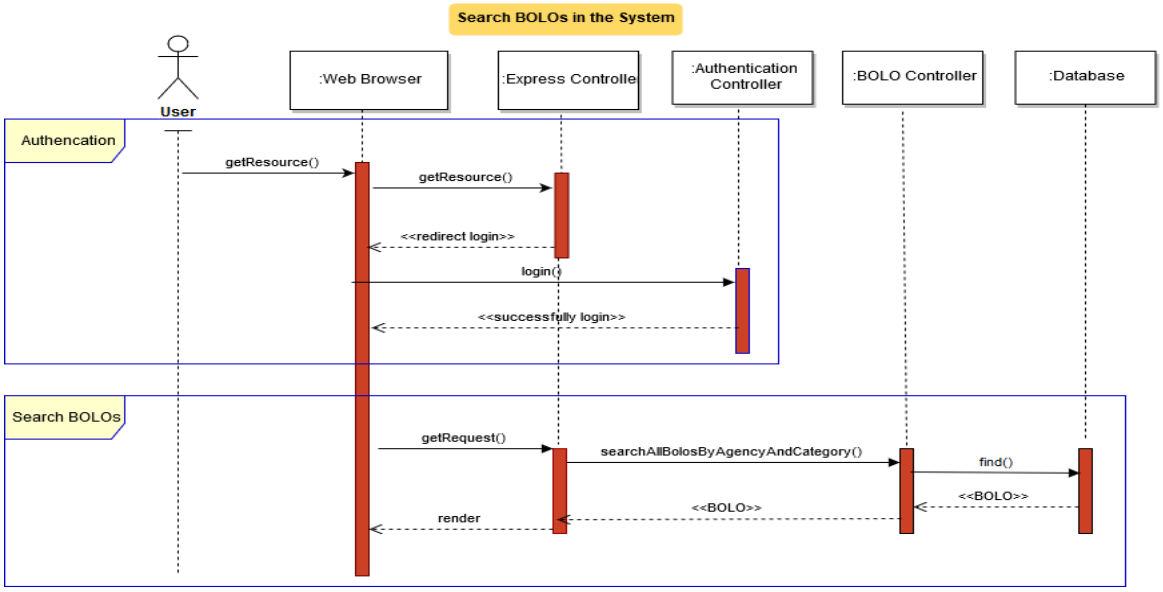


Figure 22: Sequence diagram for Searching for a BOLO, part of User Story #1218 - Search Issue in the System

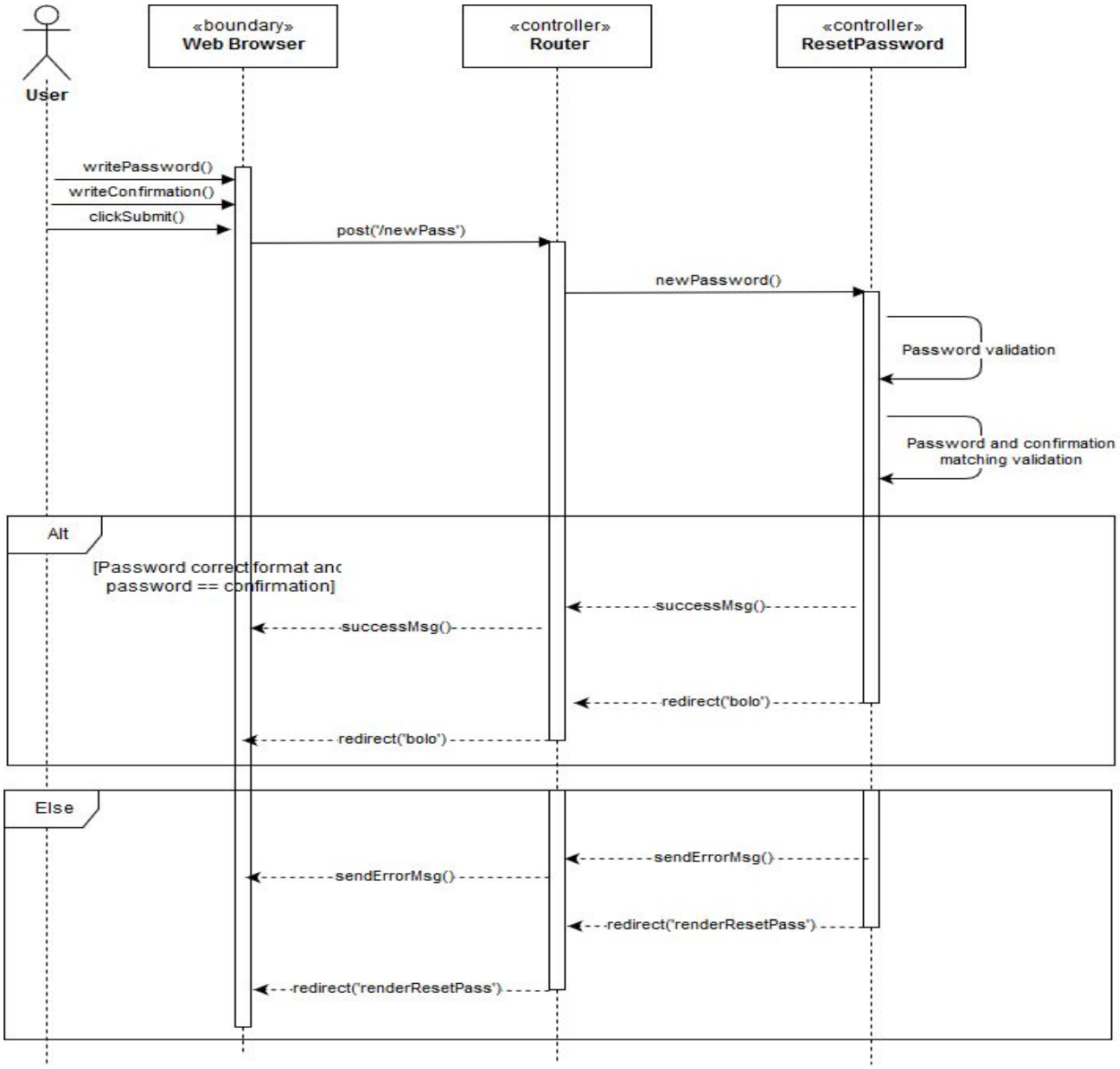


Figure 23: Sequence diagram for Resetting Password, part of User Story #1222 -Password Validation

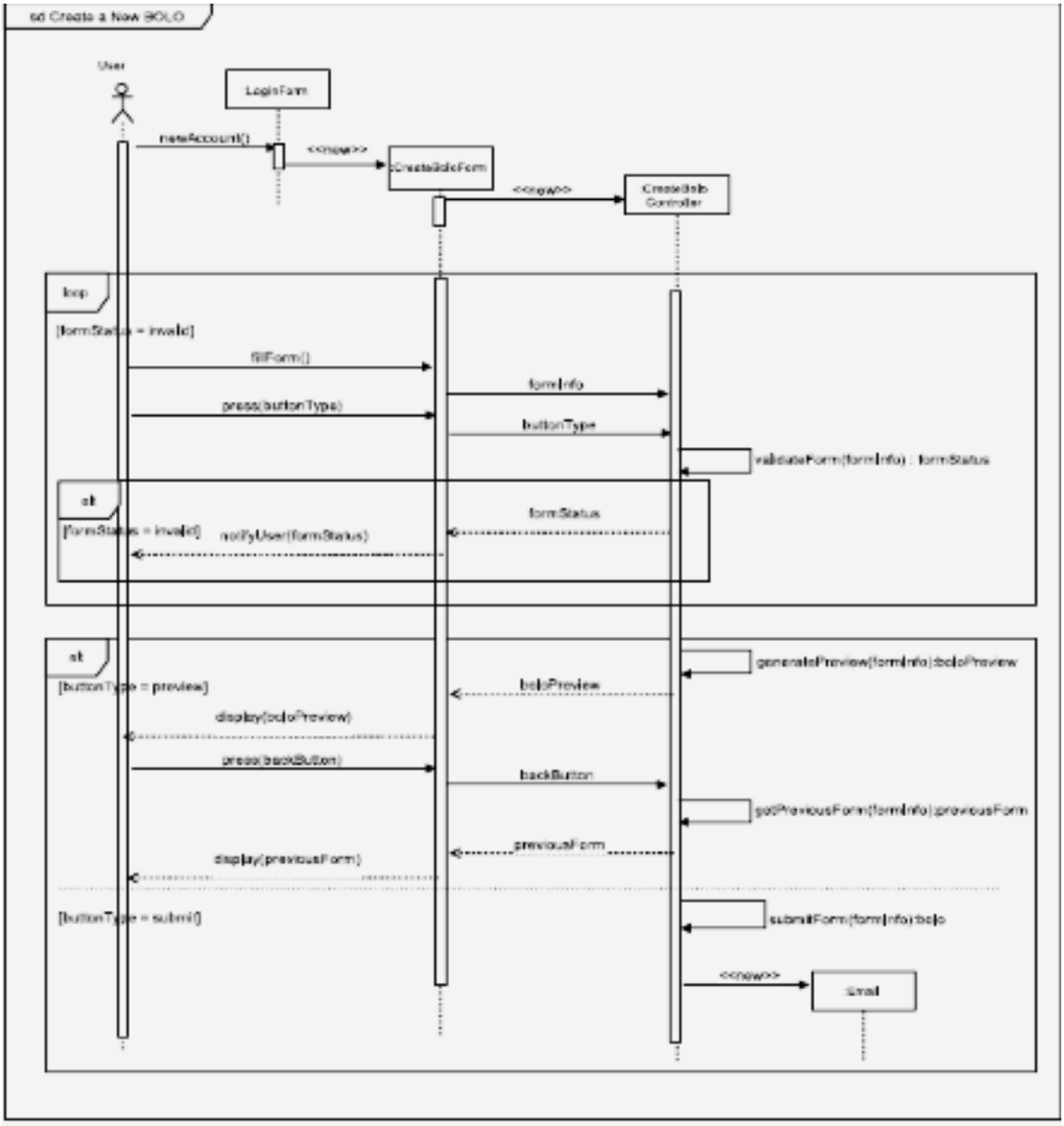


Figure 24: Sequence diagram for Creating a New BOLO, part of User Story #1246 - Prevent server from crashing when creating a new bolo without selecting a category

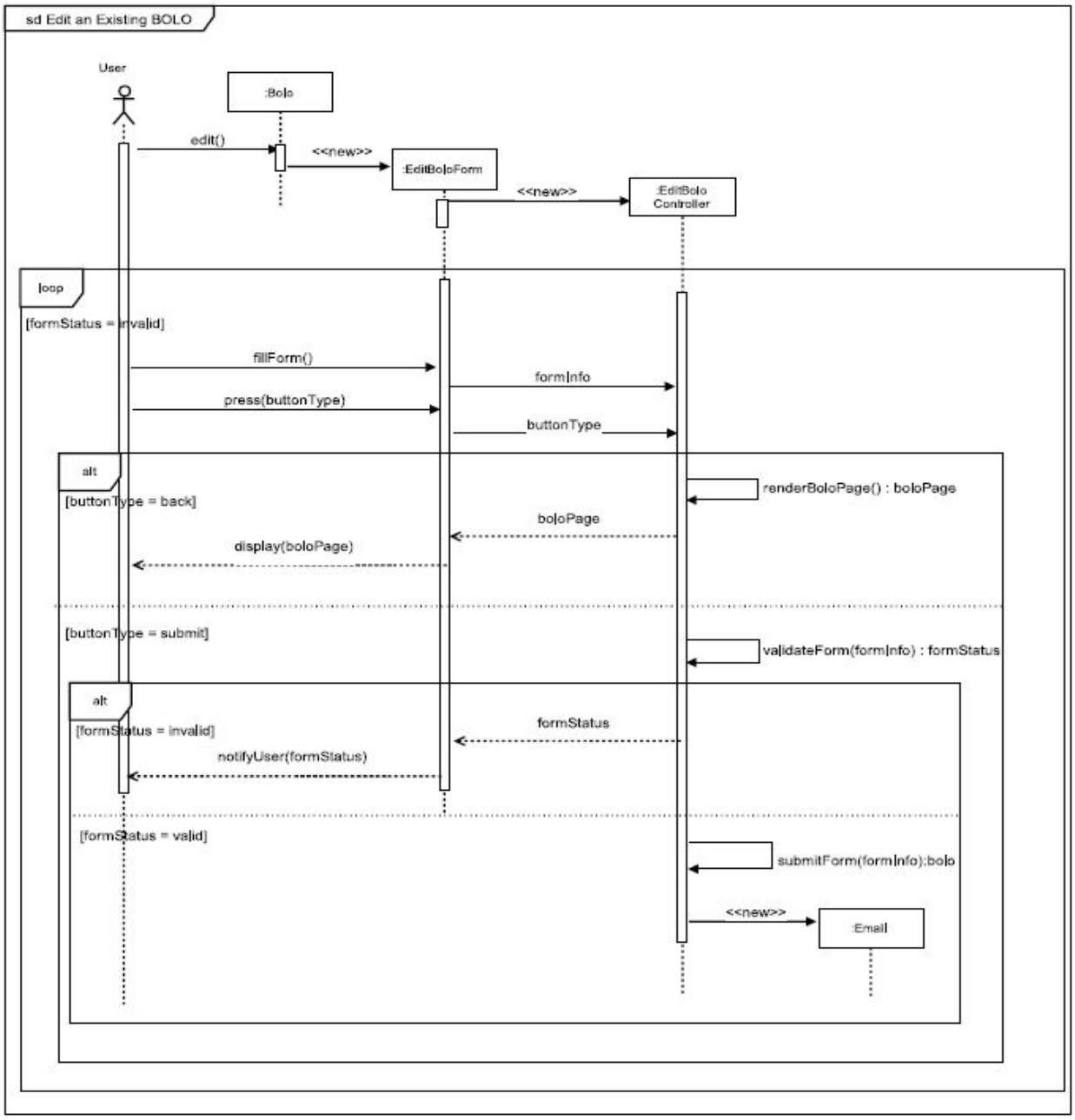


Figure 25: Sequence diagram for Editing an Existing BOLO, part of User Story #1219 - Prevent BOLO from Disappearing when Updated

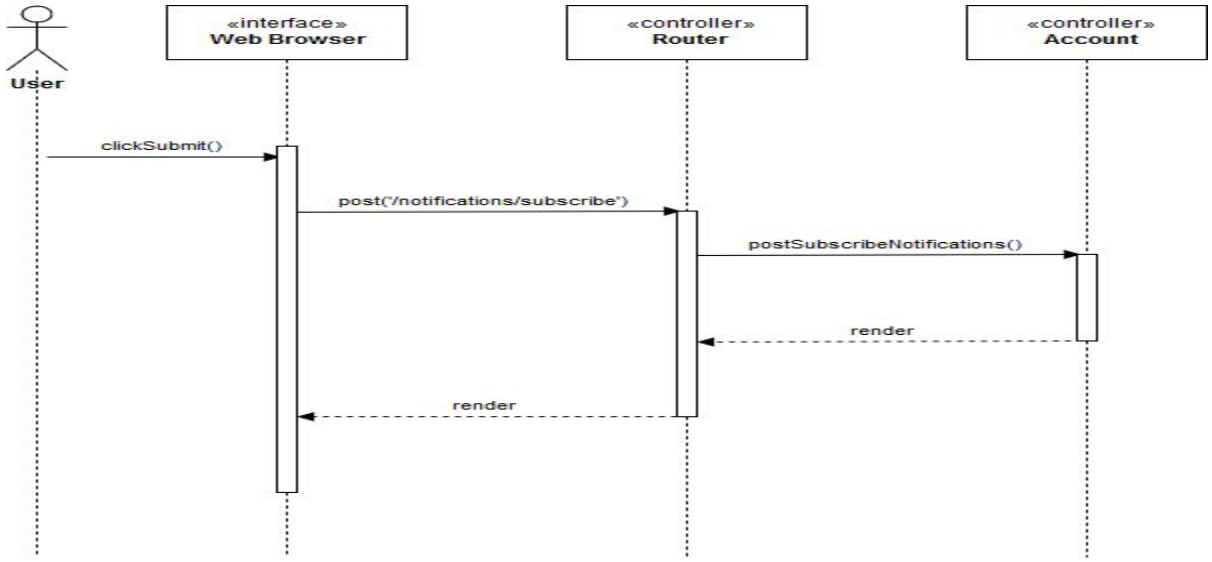


Figure 26a: Sequence diagram for Subscribing to an Agency, part of User Story #1265 - Subscribing to Agency

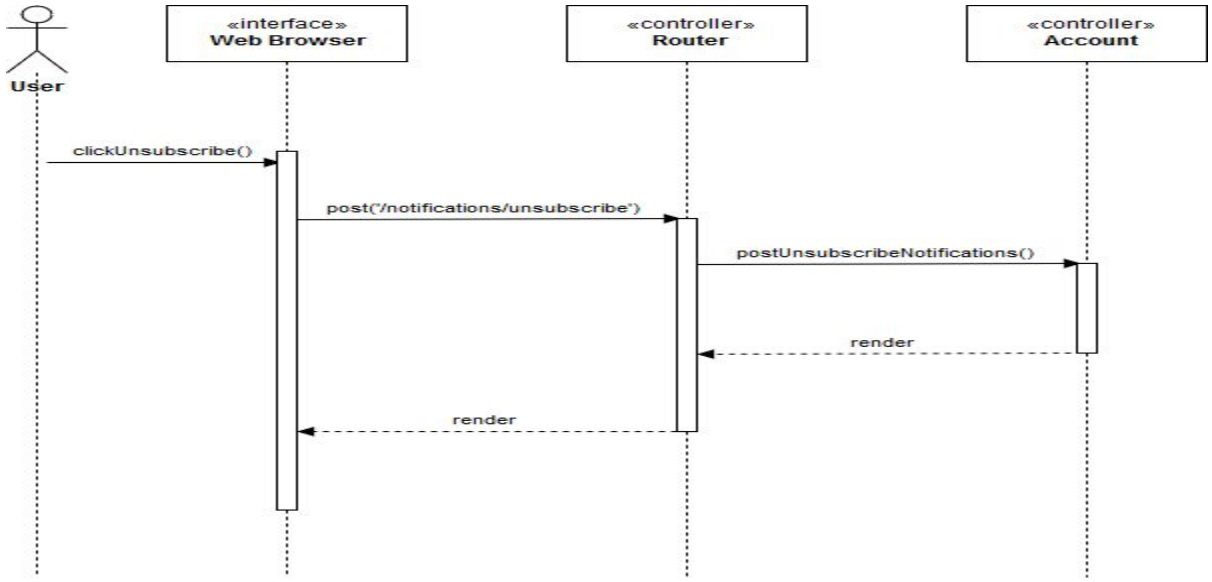


Figure 26b: Sequence diagram for Unsubscribing from an Agency, part of User Story #1265 - Subscribing to Agency

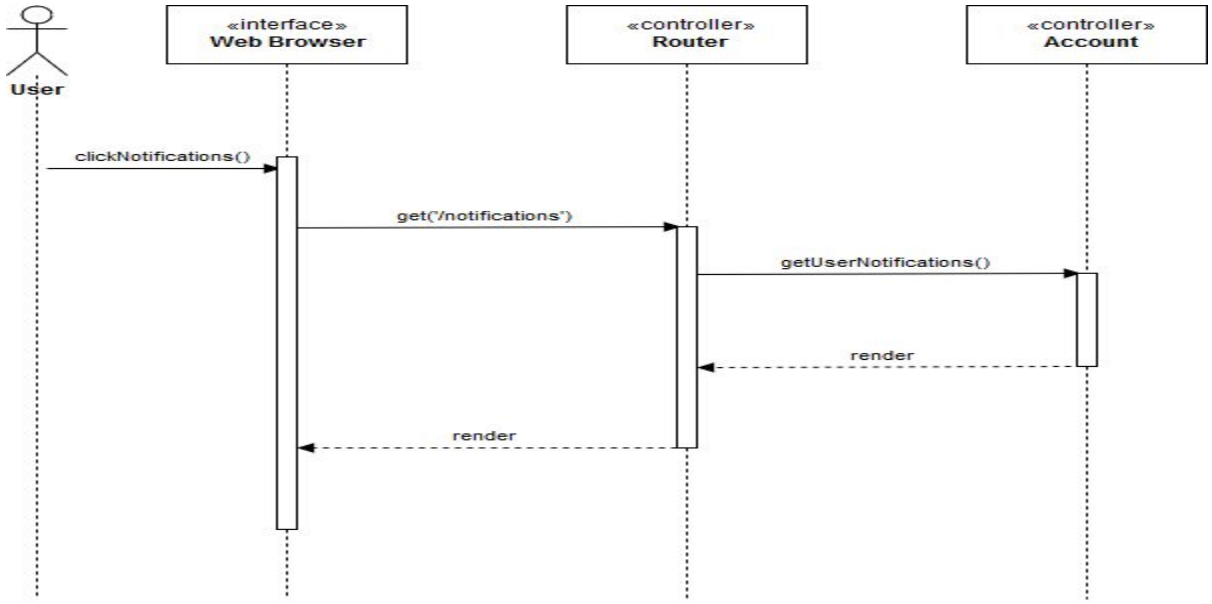


Figure 26c: Sequence diagram for View Notifications, part of User Story #1265 - Subscribing to Agency

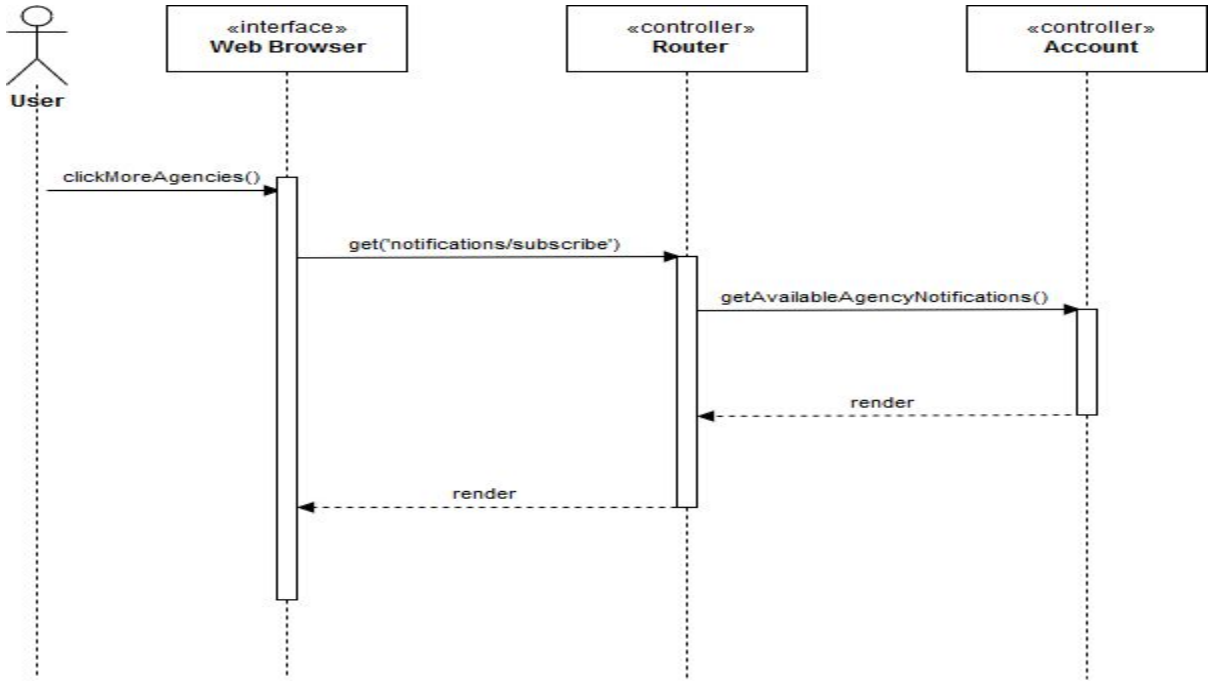


Figure 26d: Sequence diagram for View More Agencies, part of User Story #1265 - Subscribing to Agency

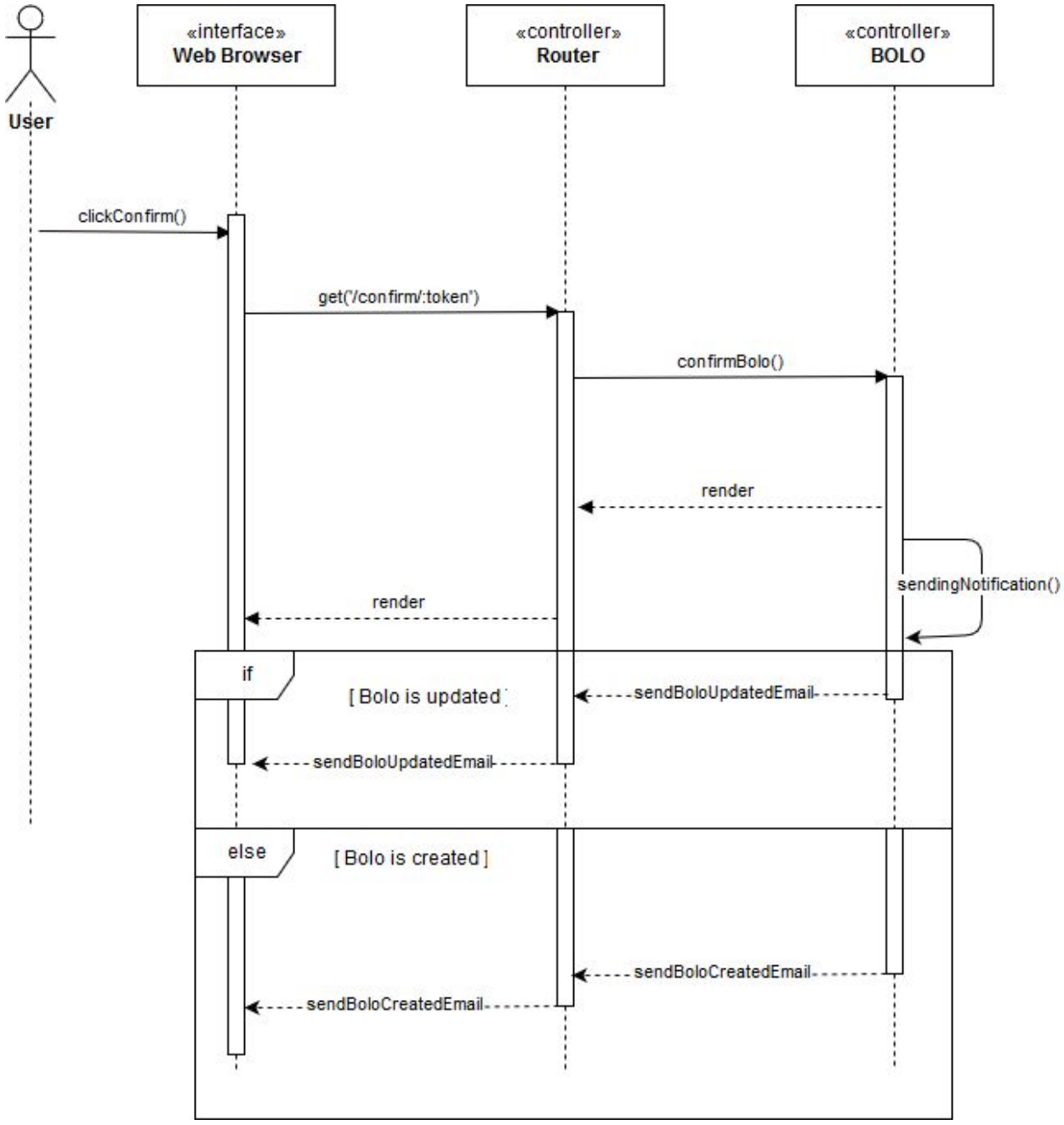


Figure 27: Sequence diagram for Notifying Subscribed Users, part of User Story #1286 - Notifying Users Subscribed to Agencies

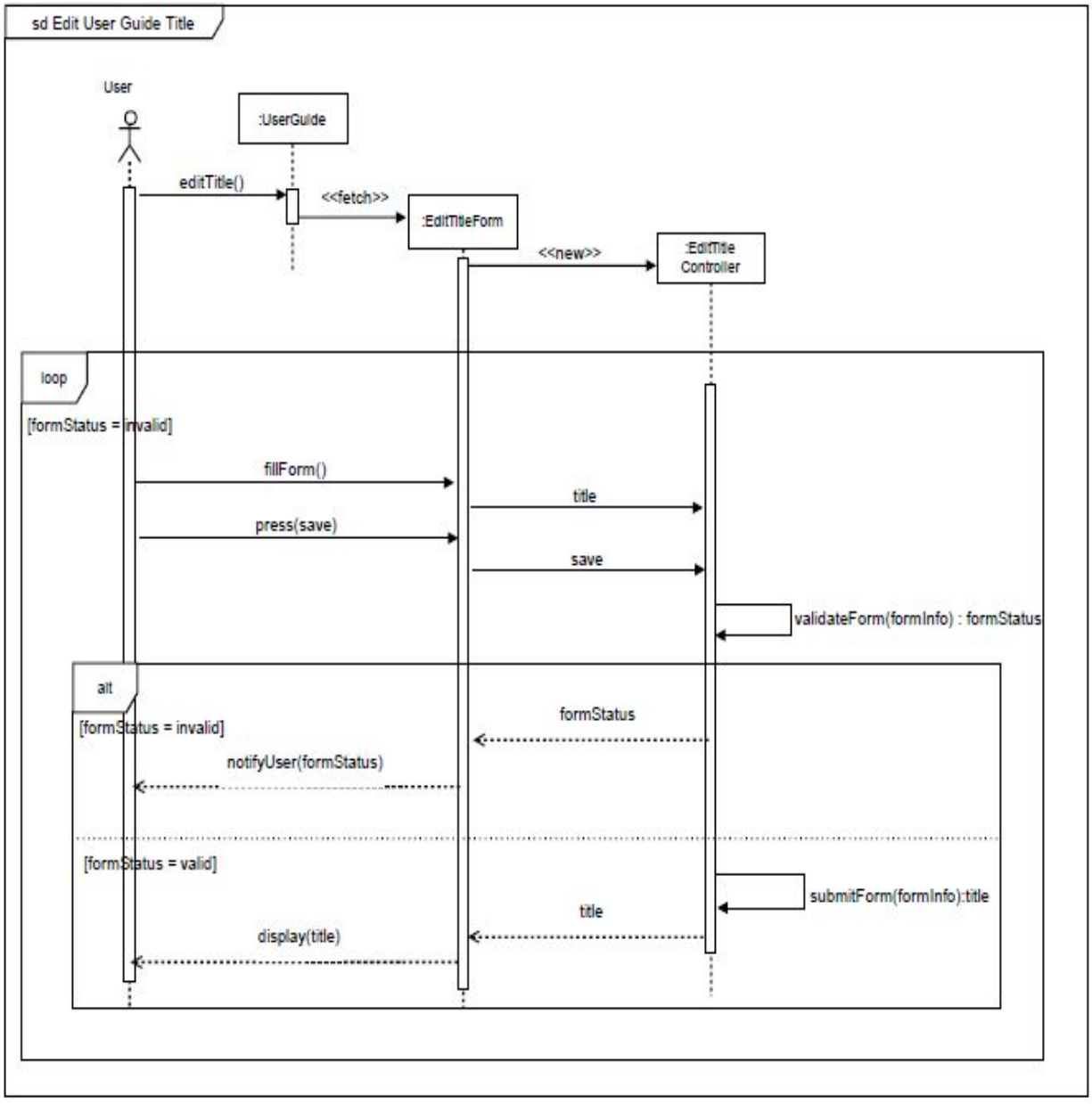


Figure 28a: Sequence diagram for Editing the Title of the User Guide, part of User Story #1287 - Edit User Guide.

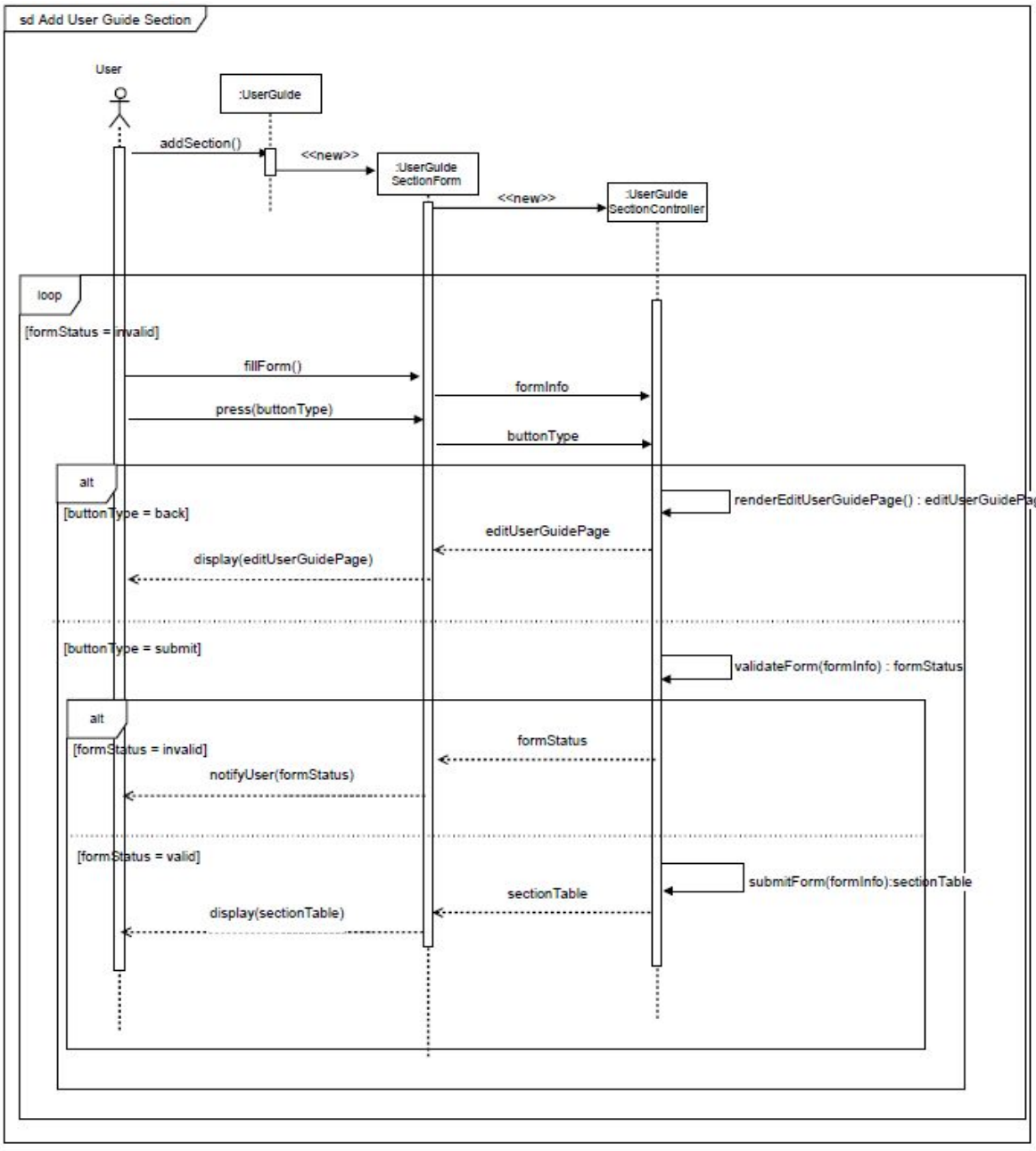


Figure 28b: Sequence diagram for Adding a Section to the User Guide, part of User Story #1287 - Edit User Guide.

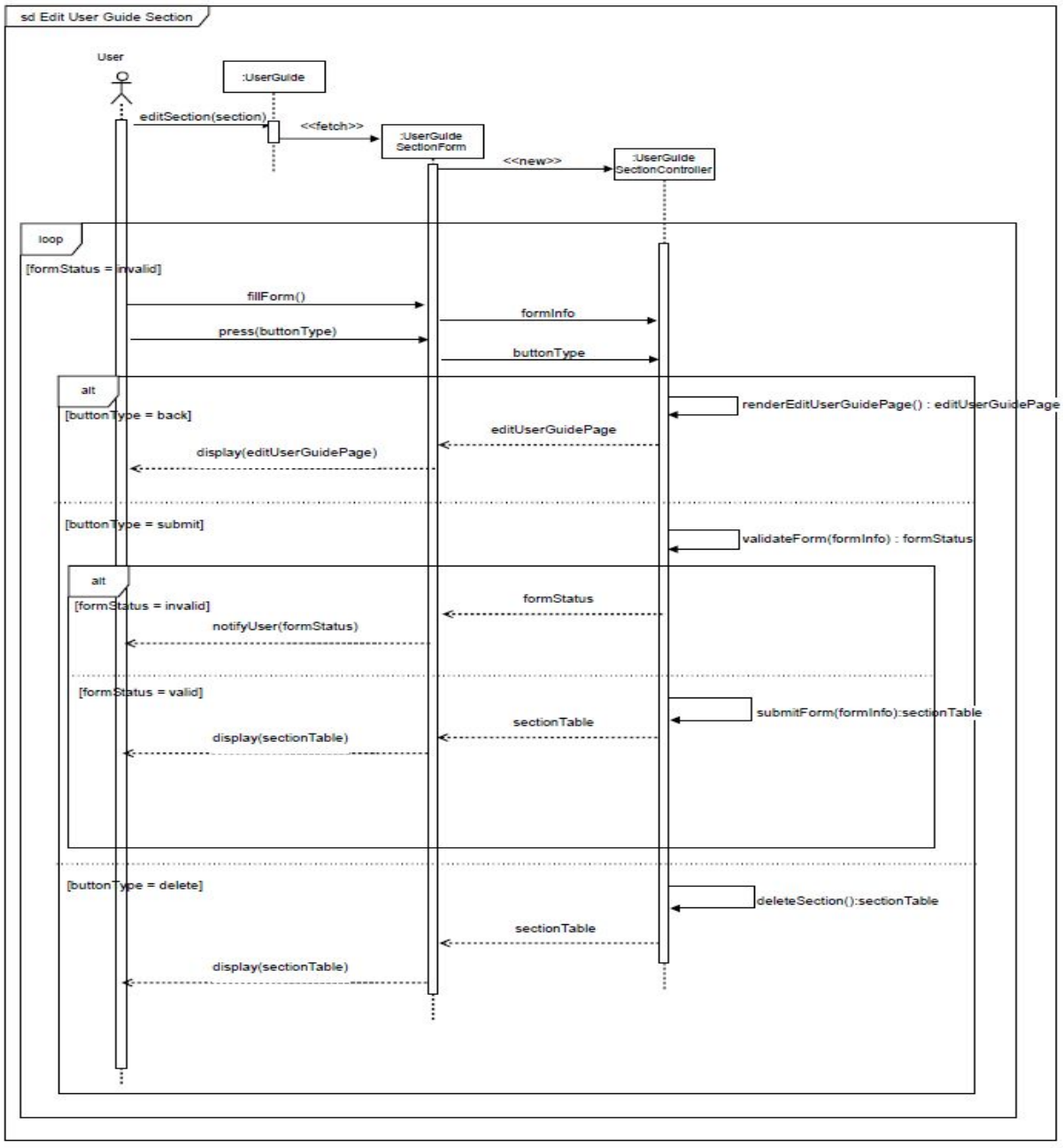


Figure 28c: Sequence diagram for Editing a Section of the User Guide, part of User Story #1287 - Edit User Guide.

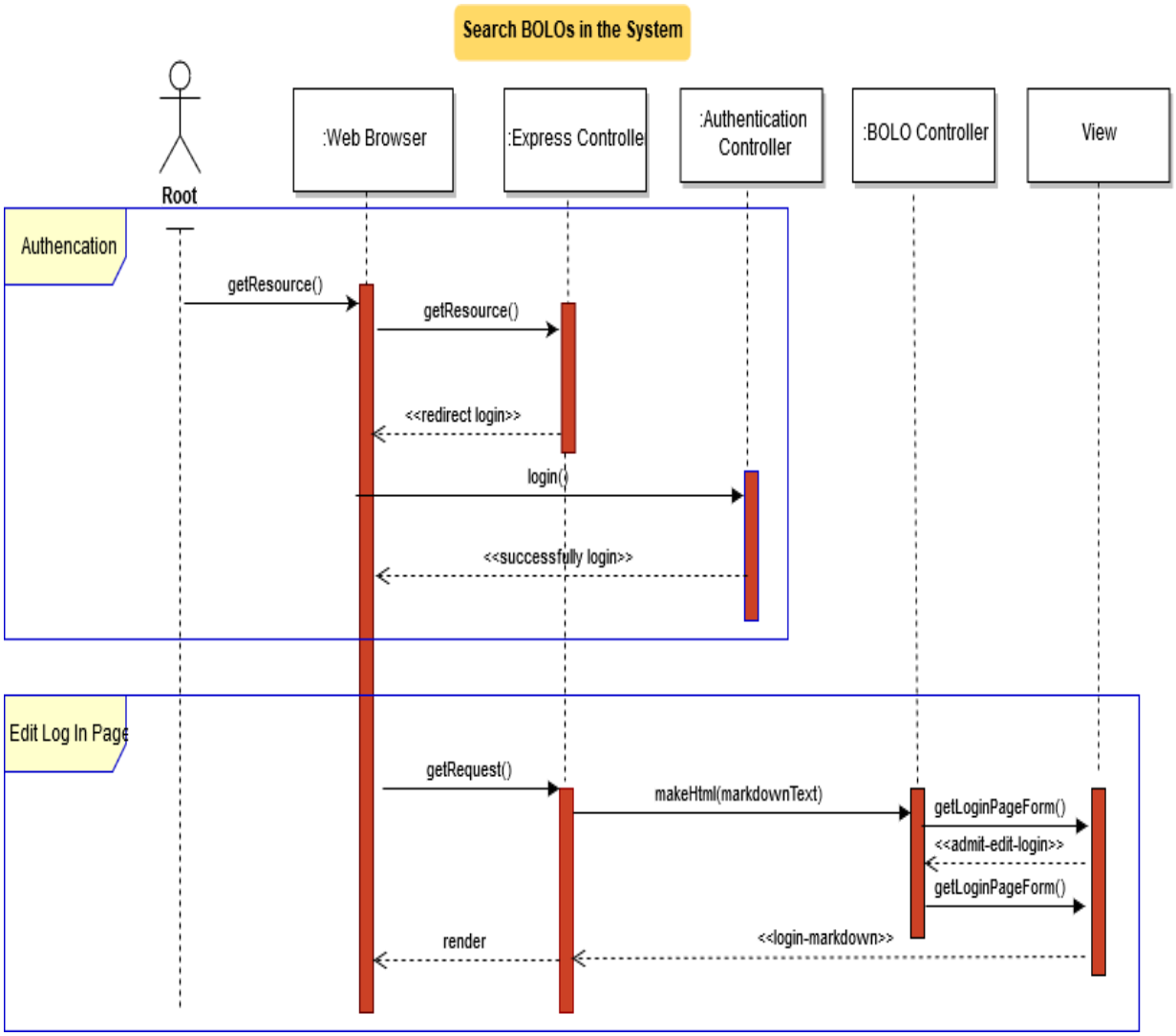


Figure 29: Sequence diagram for Editing a the Login Page, part of User Story #1288 - Edit Login Page

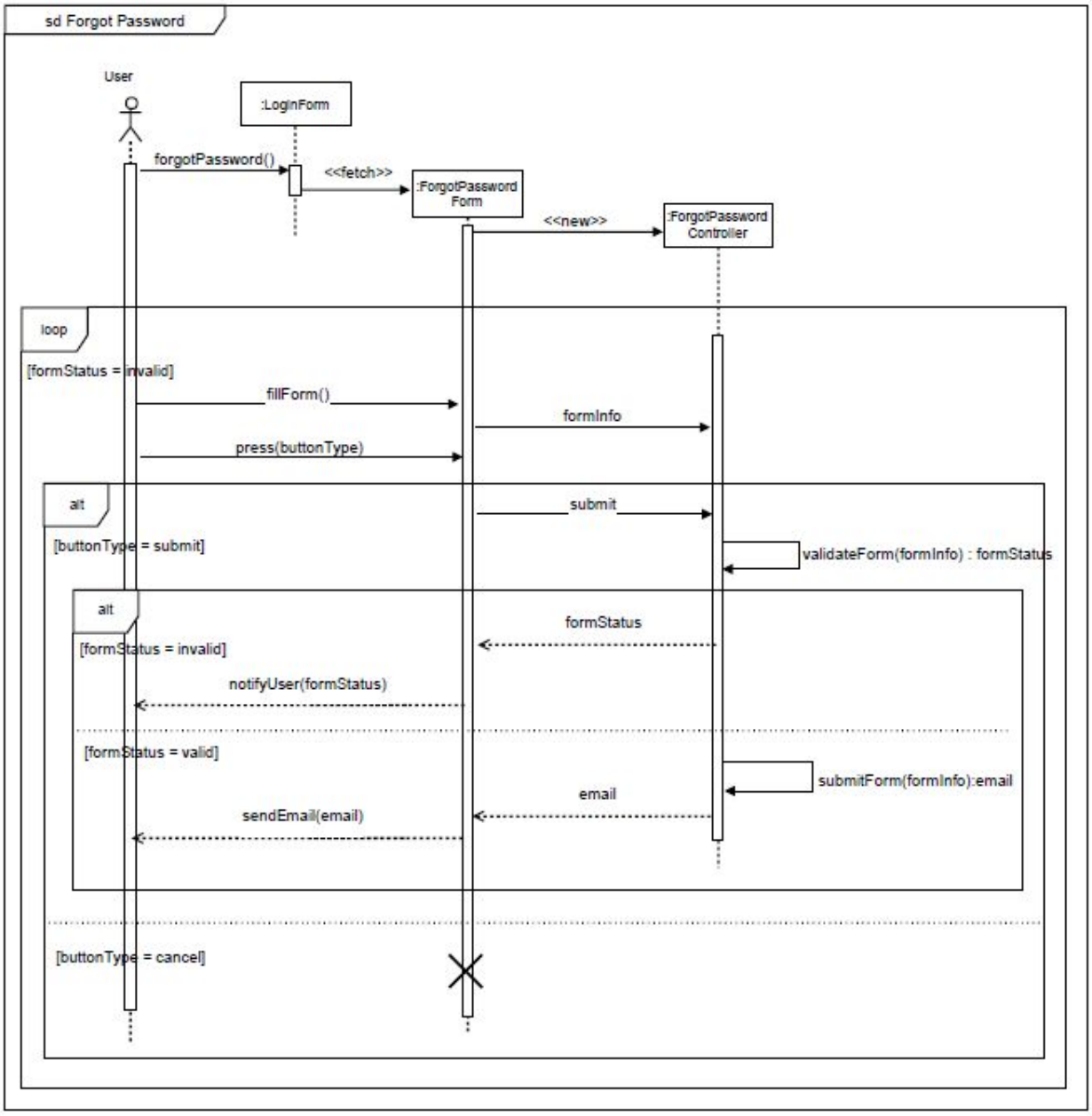


Figure 30a: Sequence diagram for Forgot Password, part of User Story #1315- Forgot Password.

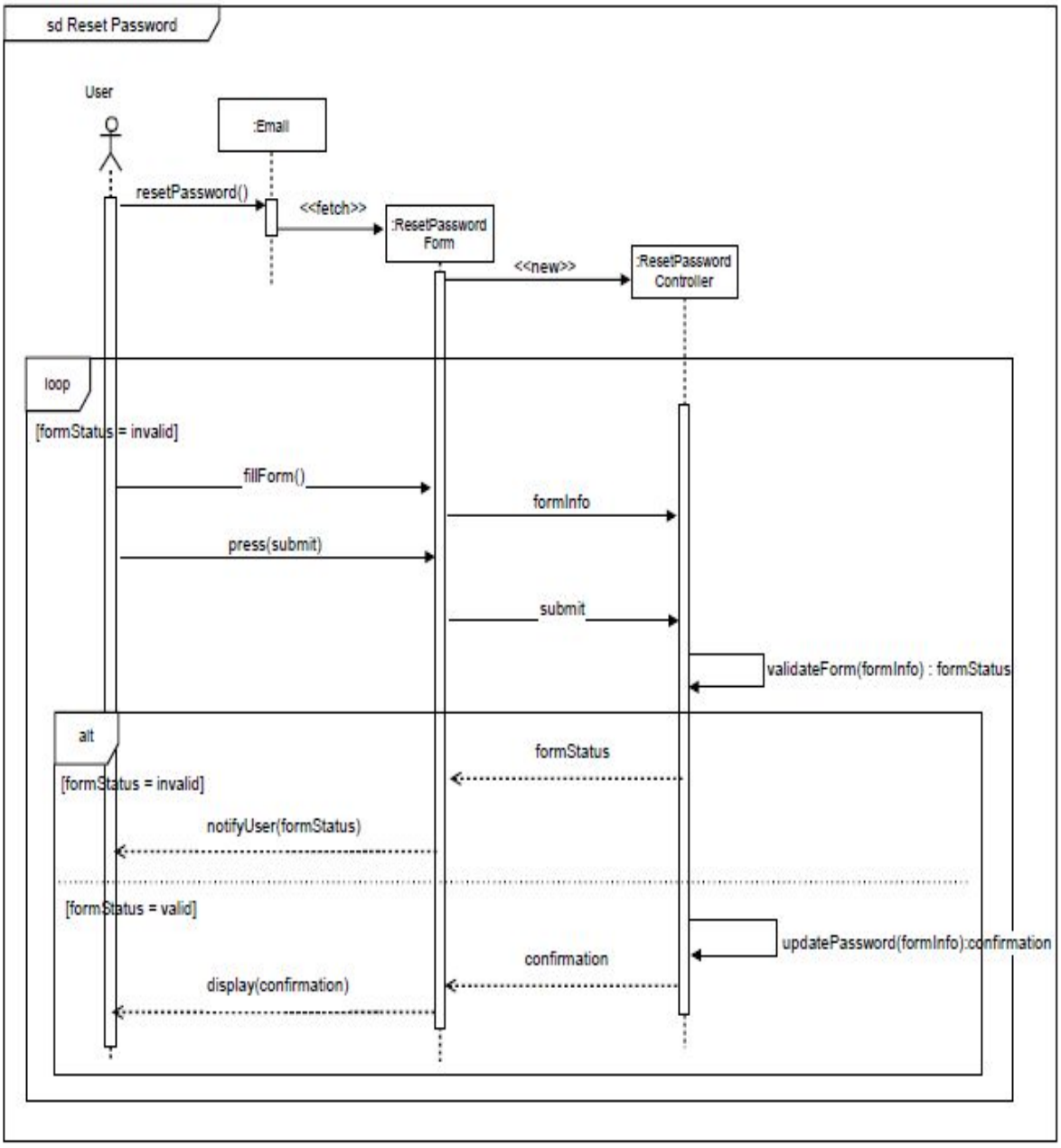


Figure 30b: Sequence diagram for Reset Password, part of User Story #1315- Forgot Password.

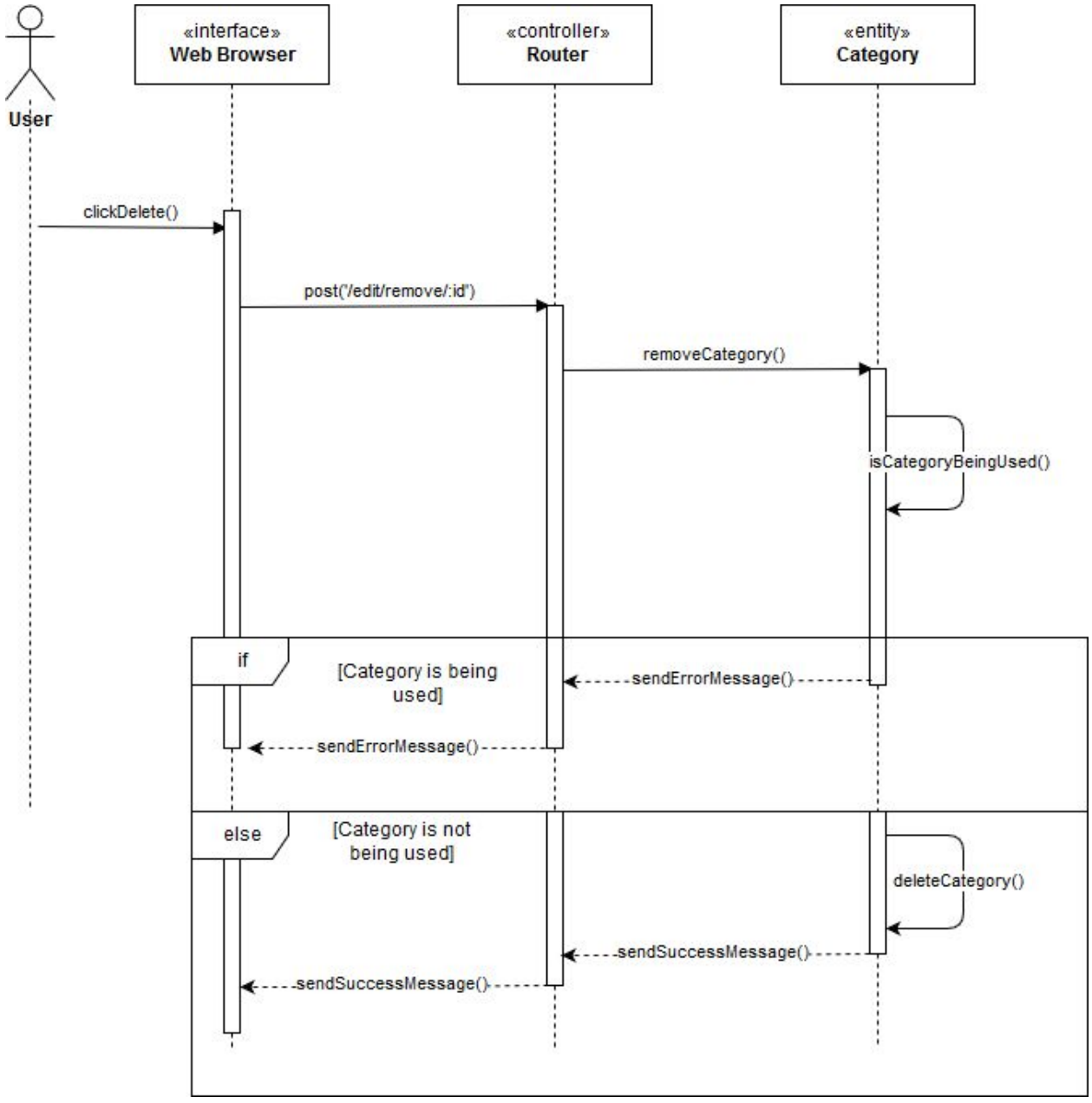


Figure 31: Sequence diagram for Delete Categories, part of User Story #1316 - Add Delete Options To Categories.

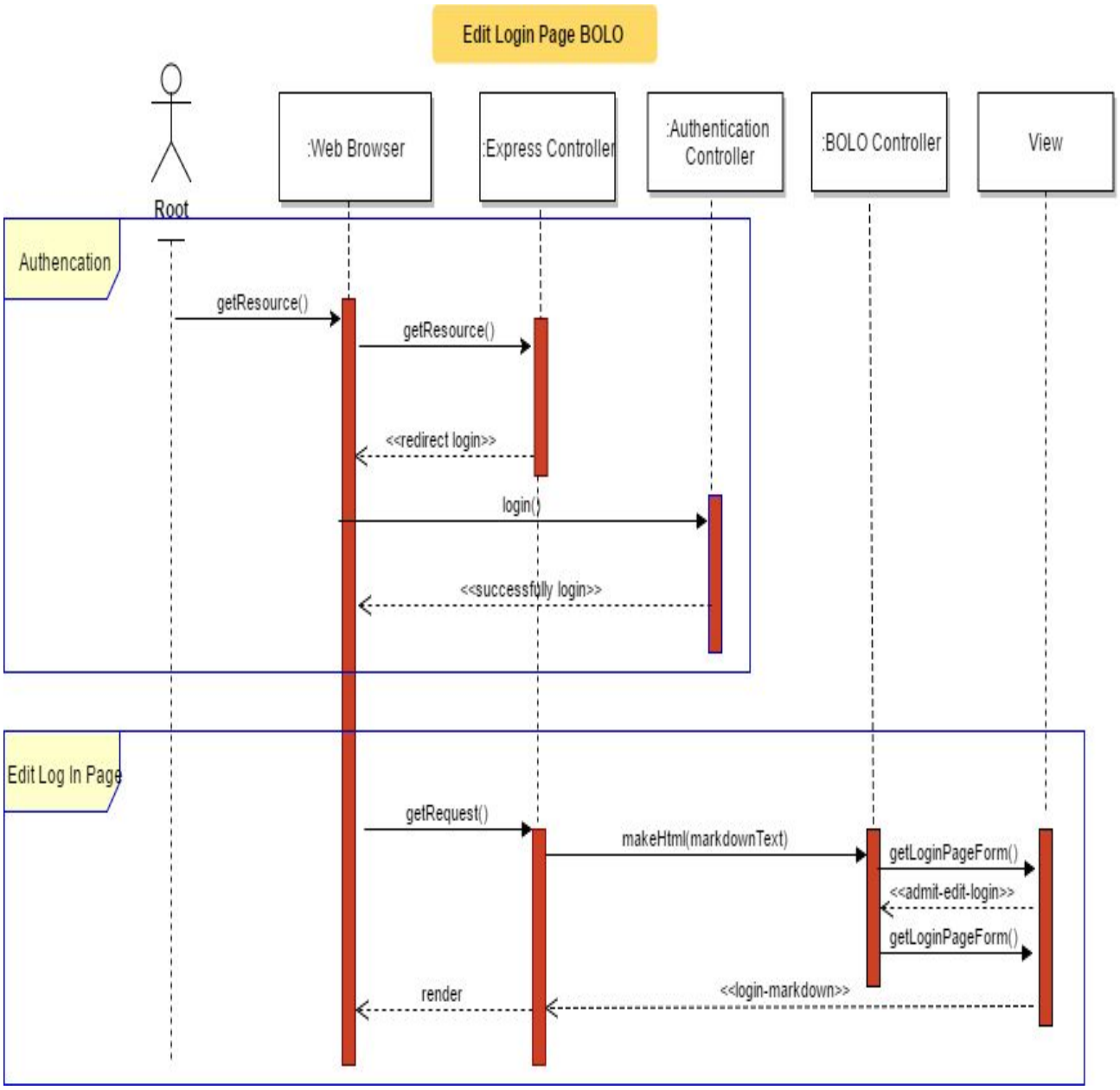


Figure 32: Sequence diagram for Edit Login Page, part of User Story #1320 - Edit login page fix bugs.

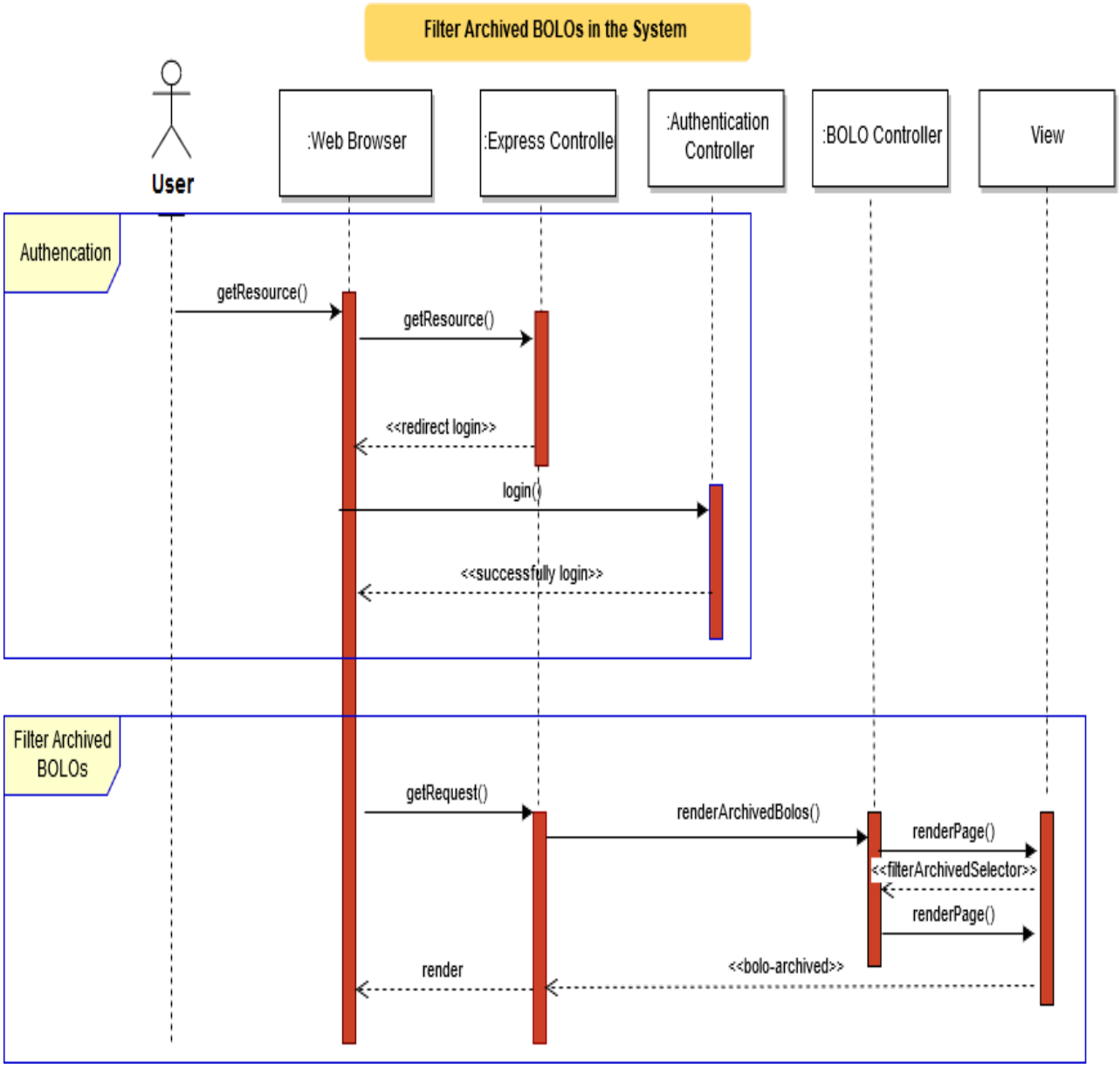


Figure 33: Sequence diagram for Filtering BOLOs, part of User Story #1223 - BOLO Filter not Working.

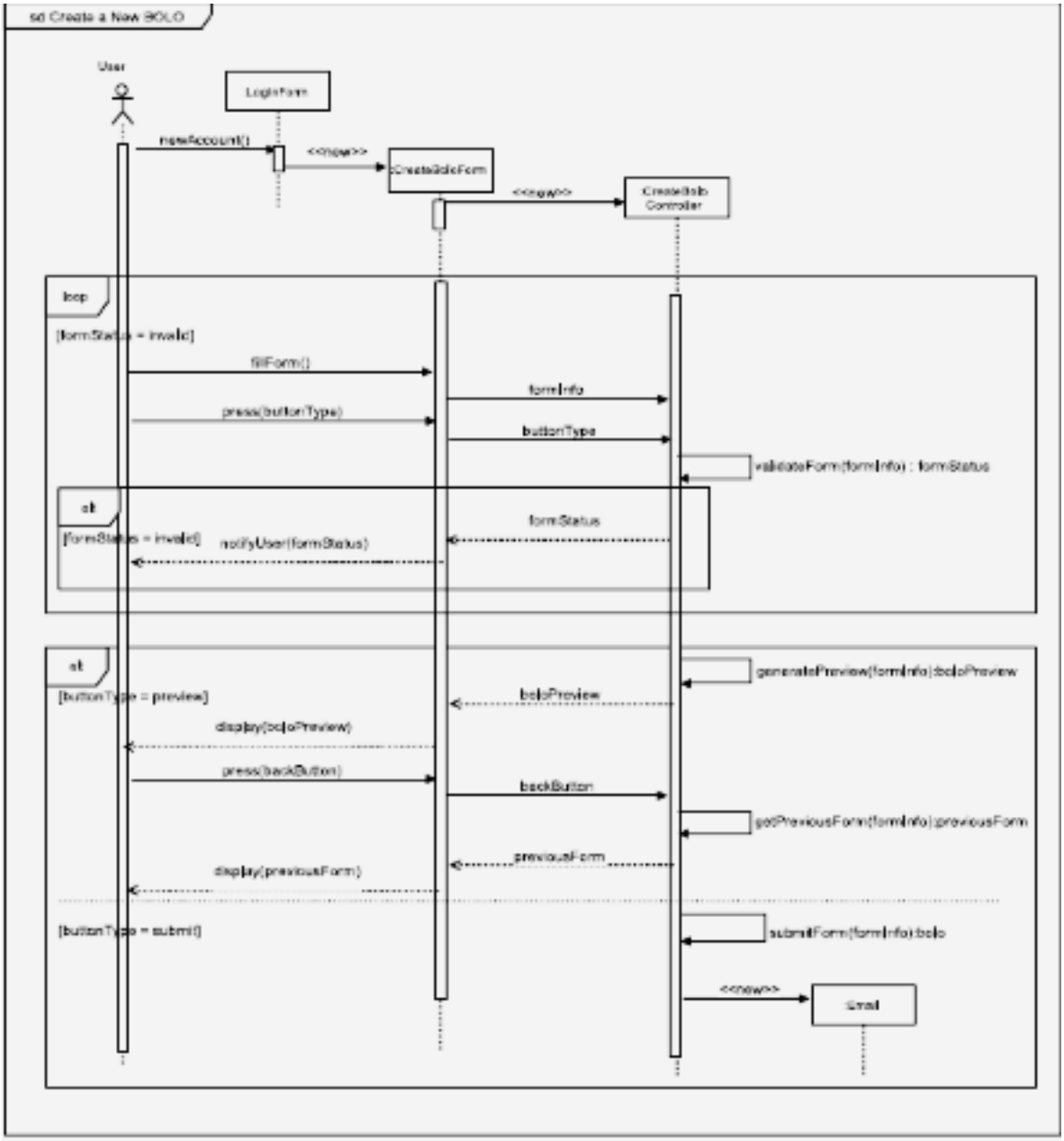


Figure 34: Sequence diagram for Creating a New BOLO, part of User Story #1328 - Add internal option to create BOLO form

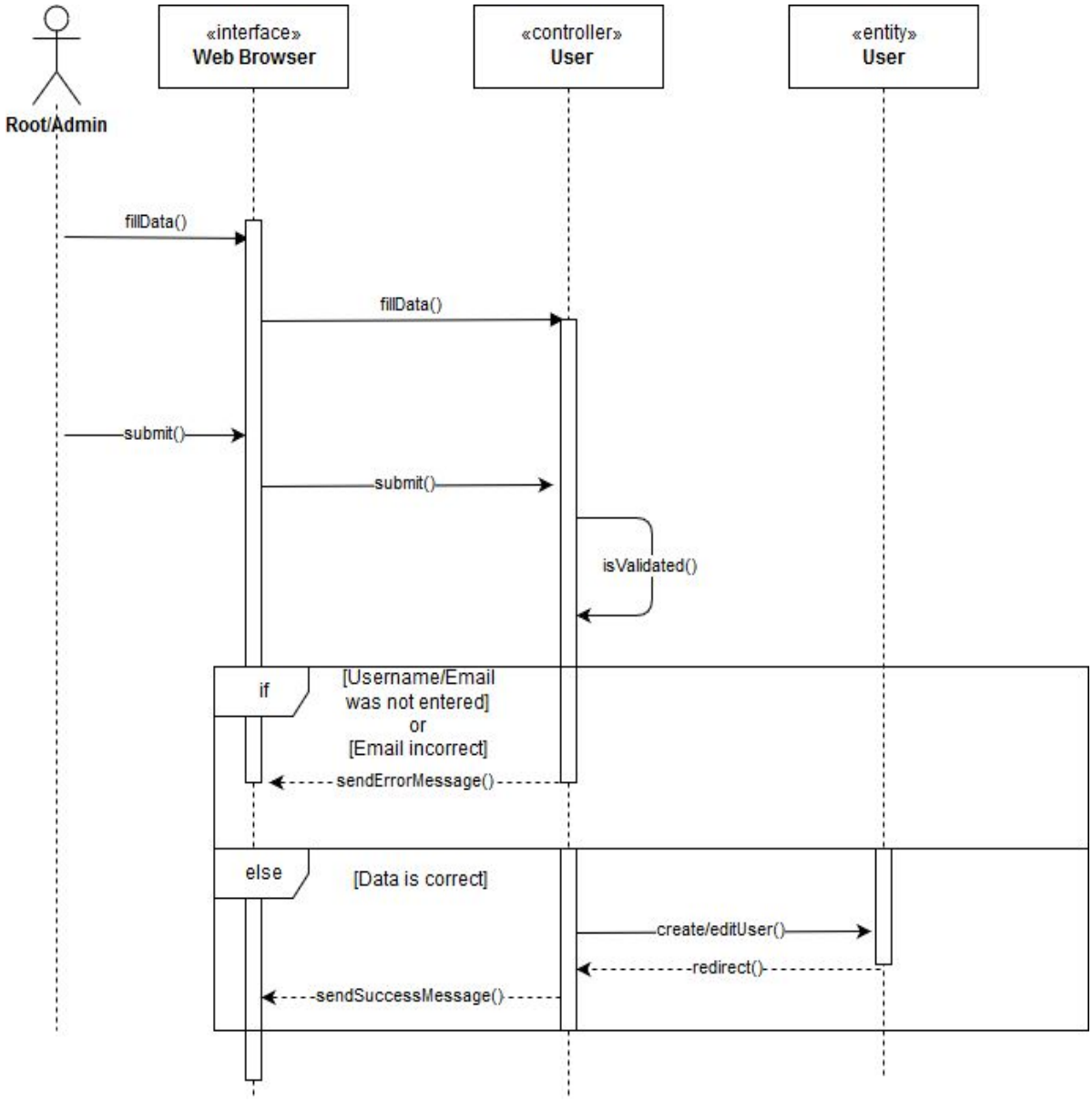


Figure 35: Sequence diagram for Creating a New User or Editing an Existing User, part of User Story #1336 - Reduce size of Officer's badge panel.

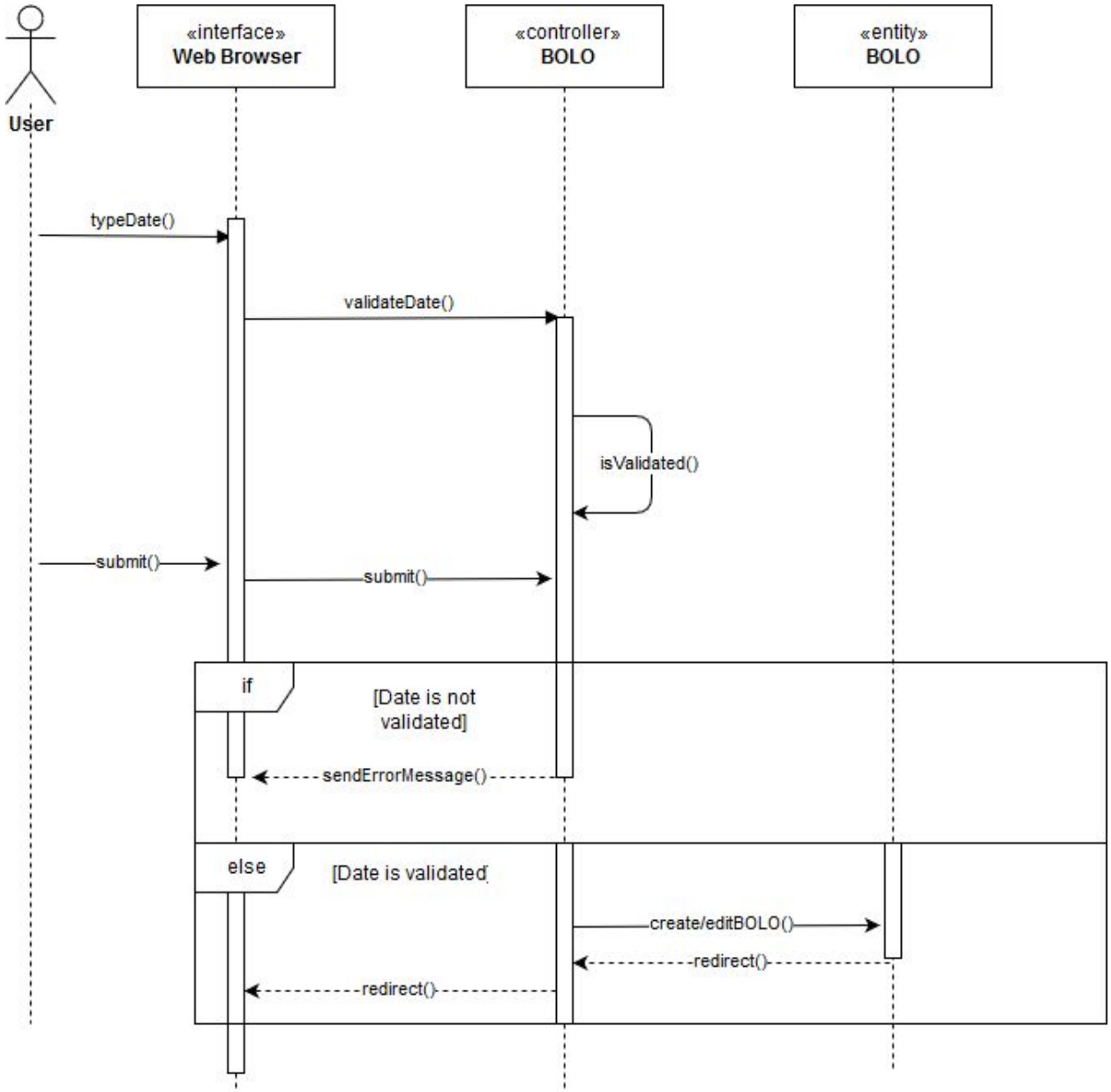


Figure 36: Sequence diagram for Creating a New BOLO or Editing an Existing BOLO, part of User Story #1337 - Change Date to US Standard.

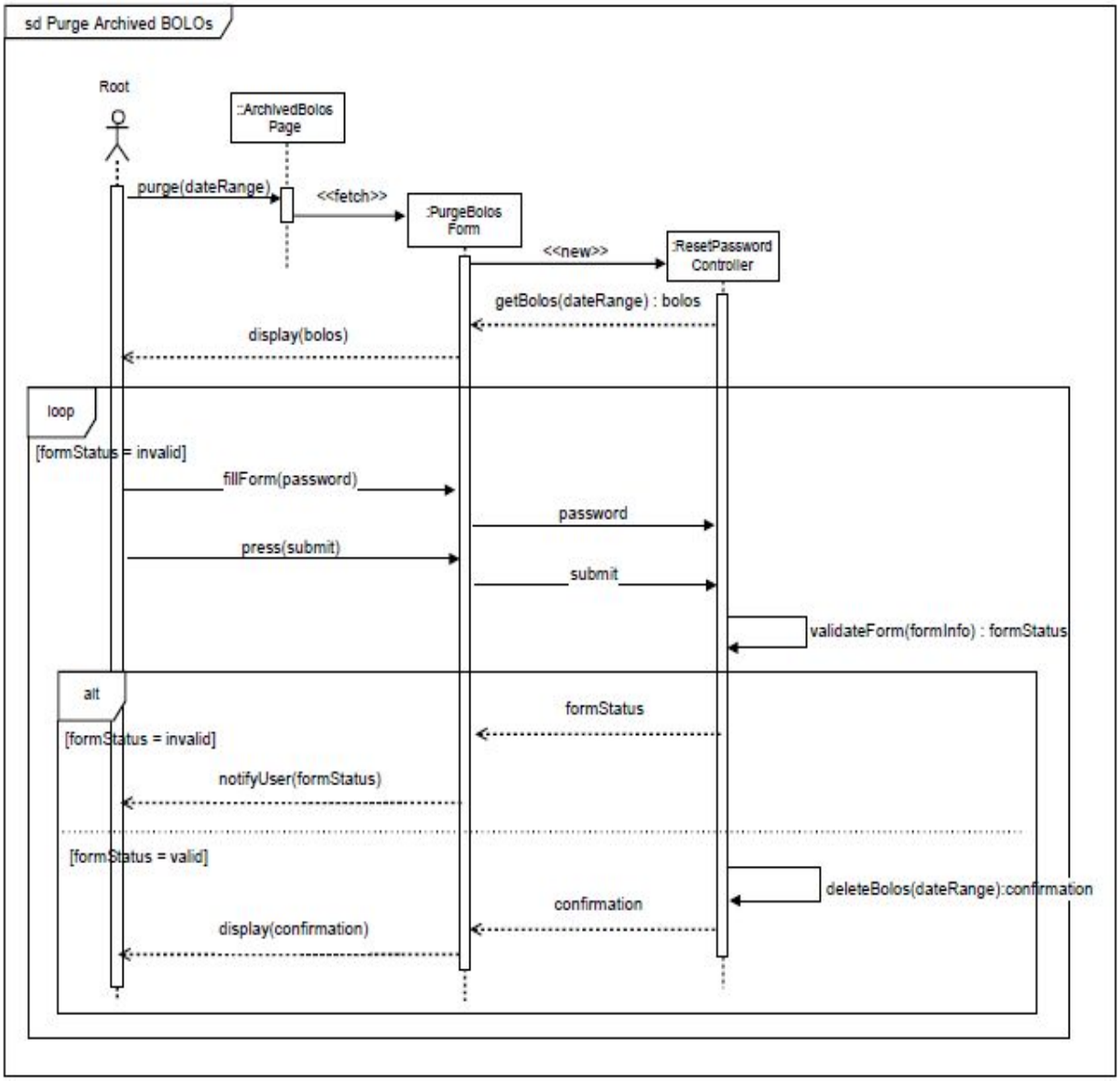


Figure 37: Sequence diagram for Deleting BOLOs, part of User Story #1338 - Add a purge date to delete archive BOLOs.

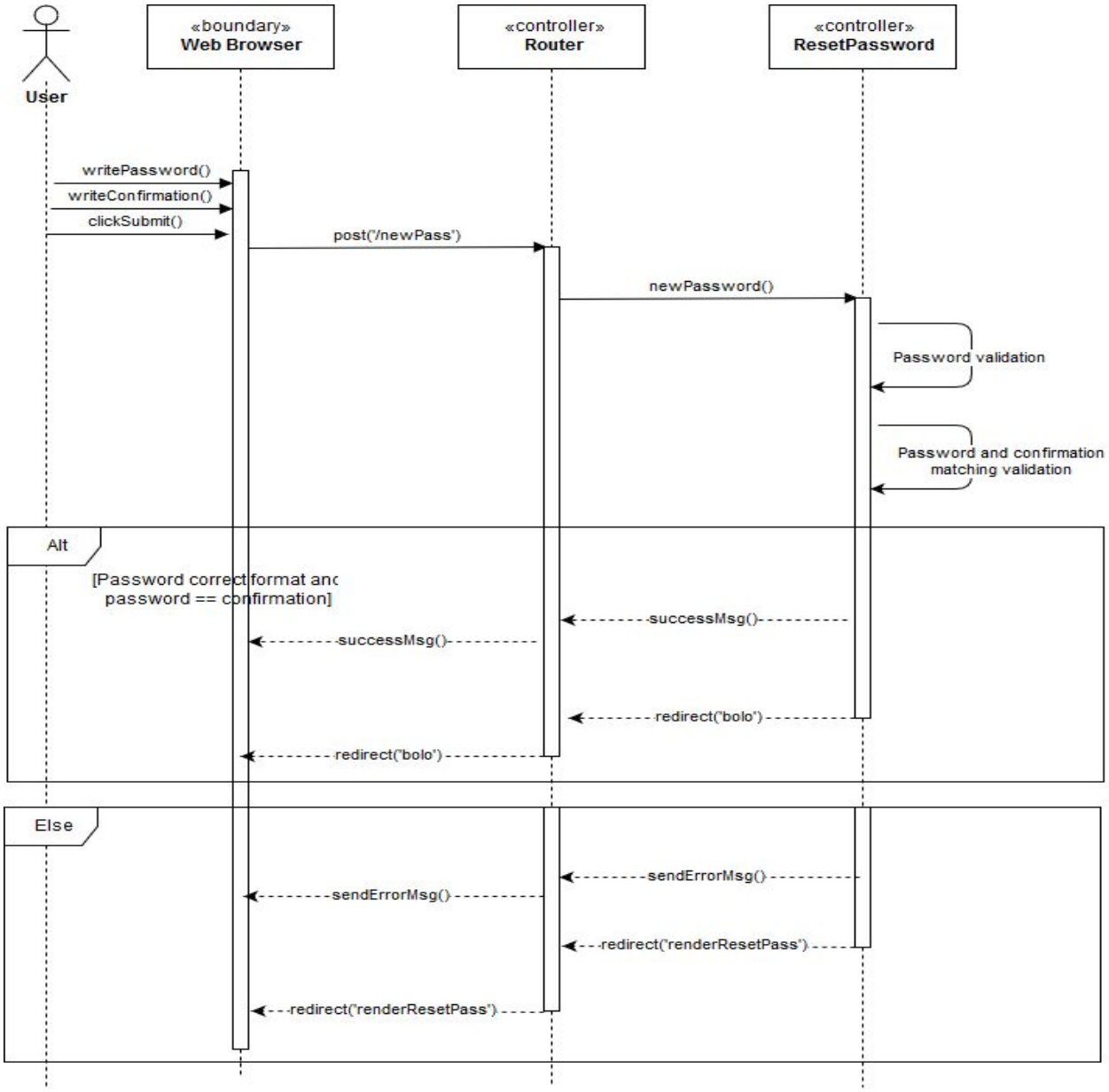


Figure 38: Sequence diagram for Resetting Password, part of User Story #1357 - Impede users from navigating page after entering temporal password.

Appendix B - User Interface Design

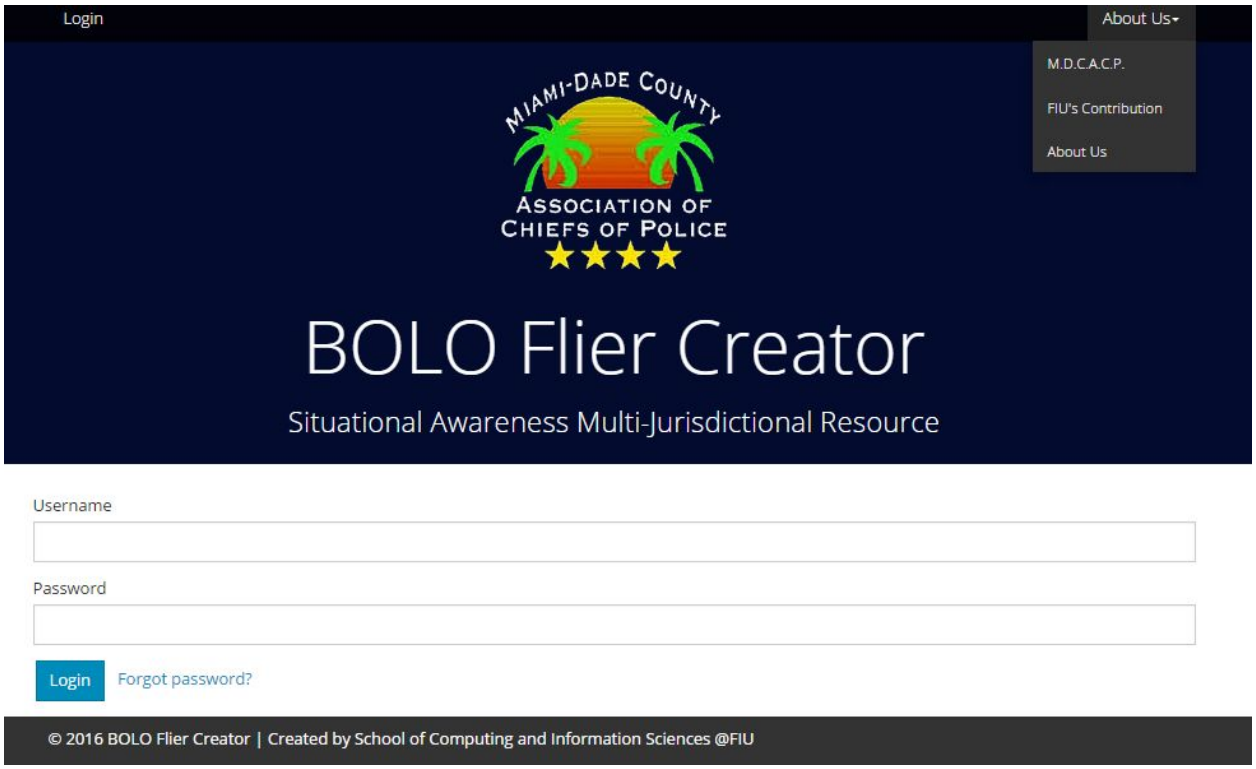
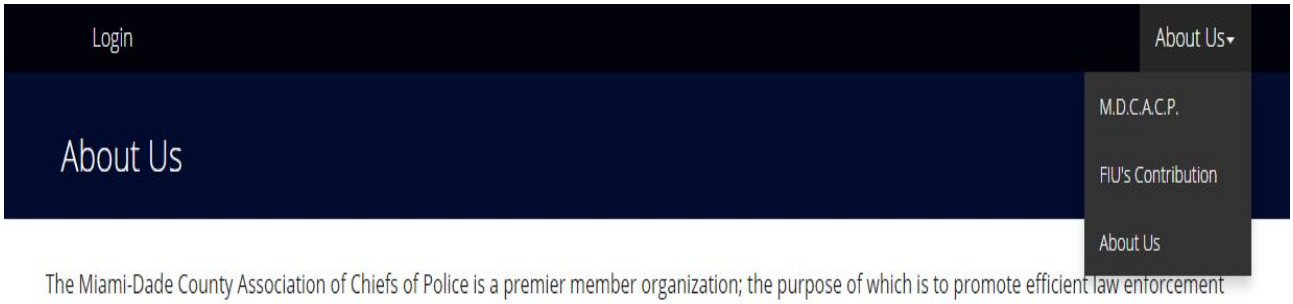


Figure 39: About us



The Miami-Dade County Association of Chiefs of Police is a premier member organization; the purpose of which is to promote efficient law enforcement services through the cooperation and coordination of all law enforcement agencies within Miami-Dade County.

The Miami-Dade County Association of Chiefs of Police is the second largest Police Chiefs Association in the State of Florida. It is comprised of more than 250 of the County's top law enforcement executives and another 300 associate members. The Association was originally organized in 1937 to promote the administration and professionalism of law enforcement that would enhance public security by providing superior police protection for the residents of Miami-Dade County and its many visitors. Today, with the same ultimate goal, its role has expanded to provide better communication, education, and training for the county's various police agencies. MDCACP facilitates the dissemination of information, provides continuing police training, and promotes a better understanding of the police profession in general. Membership in the Miami Dade County Association Chiefs of Police supports and maintains the highest standards of public safety and police protection for the millions of people who live, work and play in Miami-Dade County.

Figure 40: About us Miami Dade Association of Chiefs of Police

For the seniors in FIU's Department of Electrical and ComputerEngineering, everything they've learned in four years culminates with a single project. Working together in teams, they take on projects that test their knowledge, ingenuity, patience and imagination. Finally, on Senior Design Day, the students put on their best suits and showcase their projects to a host of industry advisers who pick their brains and give them feedback on their work. The Senior Project course was created to test a potential graduates ability to work in a real software development environment. Potential graduates' success in this course will enable them to provide a significant software project that may help them land a job in the real-world.

Masoud Sadjadi, PhD

Dr. Sadjadi holds a B.S. degree in Hardware Engineering, an M.S.degree in Software Engineering, and a Ph.D. degree in Computer Science. He is currently an Associate Professor in the School of Computing and Information Sciences at Florida International University, where he has been on the faculty since 2004. He serves as the Director of the VIP project at FIU, the Director of the Professional Master in Science for Information Technology at FIU, and is in charge of Computer Science Senior Projects. He has extensive experience in software development and leading large-scale software engineering projects both in industry and in academia. During the past three years, Dr. Sadjadi has successfully transformed senior projects at FIU from a traditional waterfall approach to an agile/scrum approach. Dr. Sadjadis research is focused on Agile Software Development in Distributed Systems, Autonomic Computing, High-Performance Computing, Grid Computing, Cloud Computing, Pervasive Systems, and Mobile Computing. He is the Director of the Autonomic Cloud Research Laboratory (ACRL) at FIU and leads several projects under the Latin American Grid initiative. Currently, he is collaborating with top researchers in 8 countries and is leading several

Figure 41: About us FIU

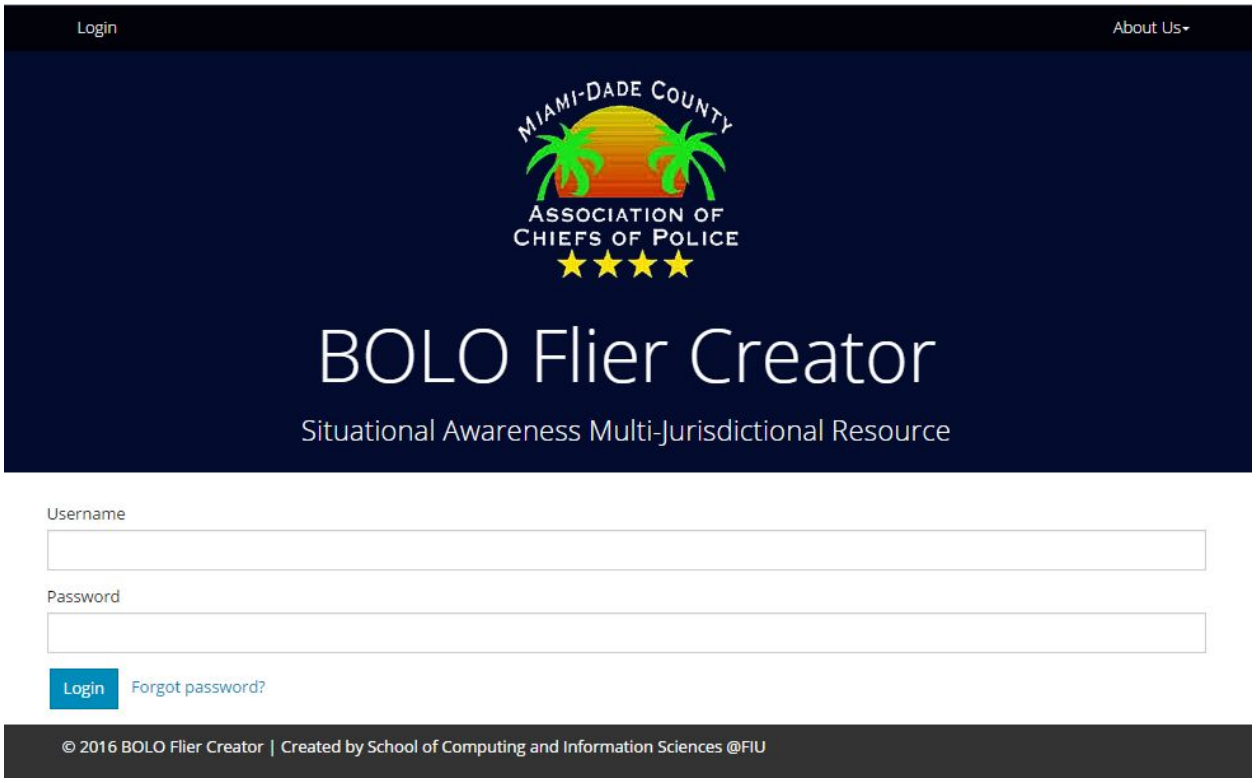


Figure 42: Login

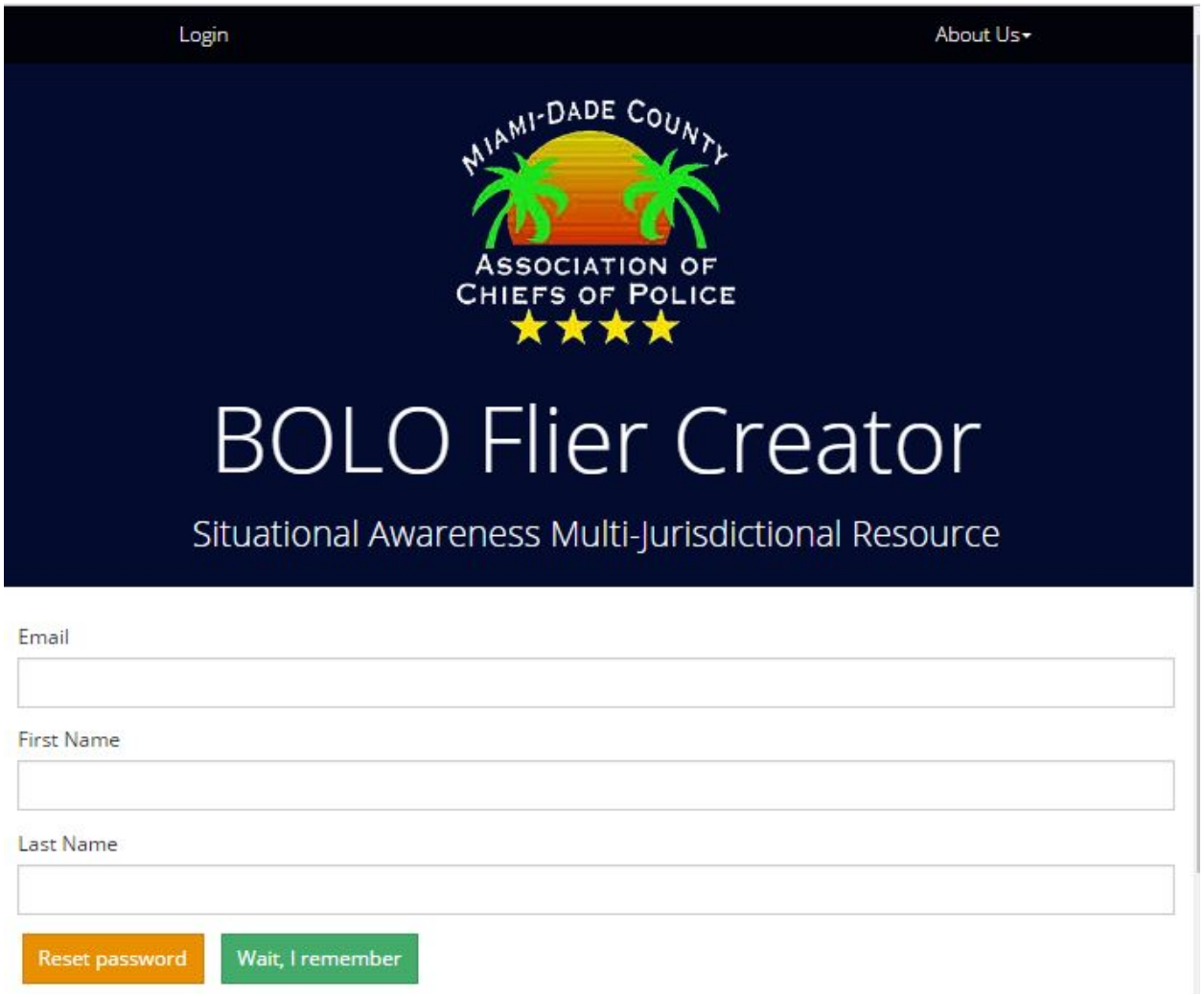


Figure 43: Forgot Password

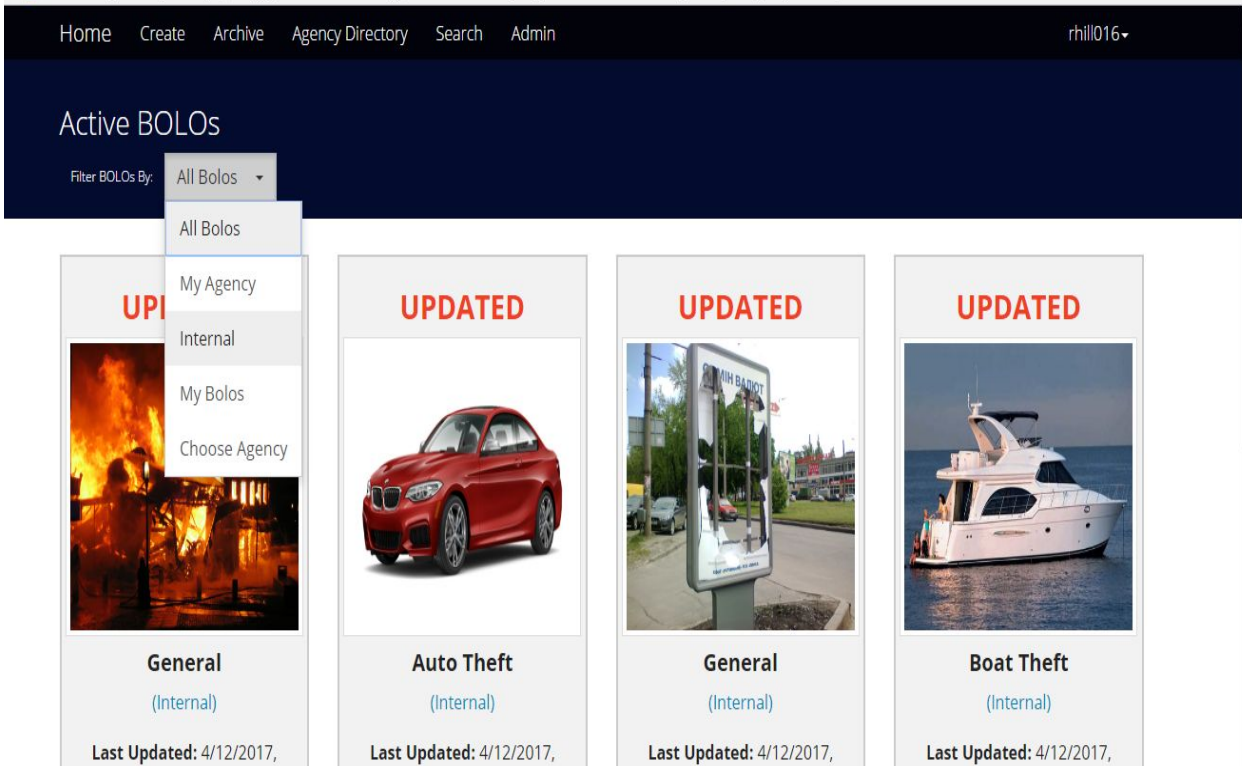


Figure 44: Filter Bolos

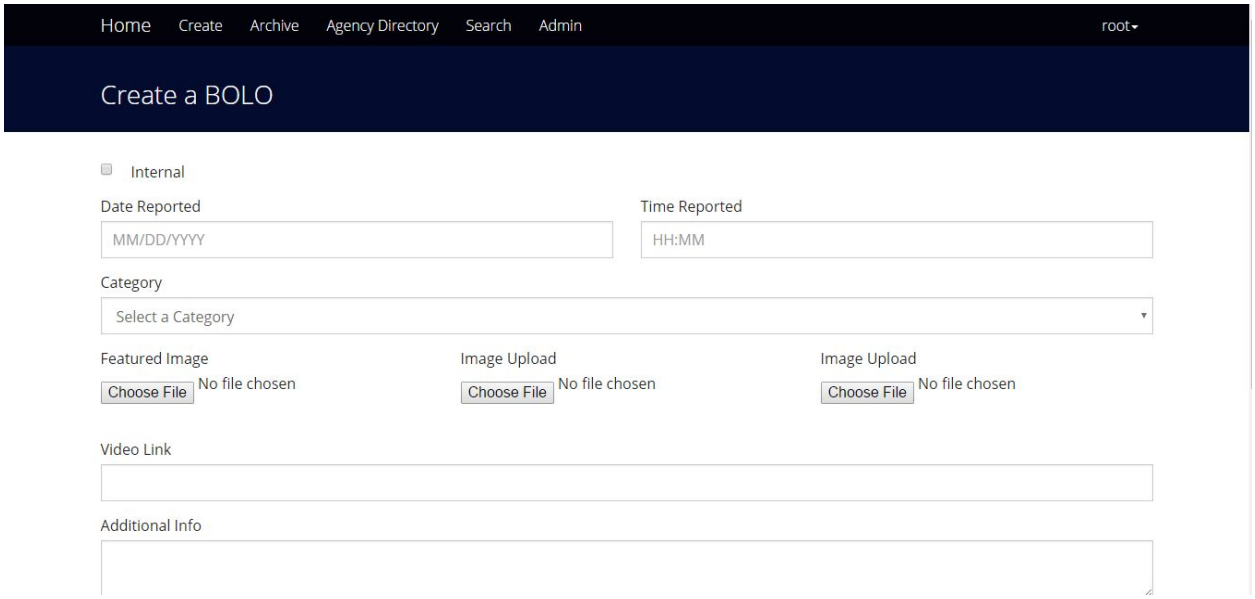


Figure 45: Create Bolo Form

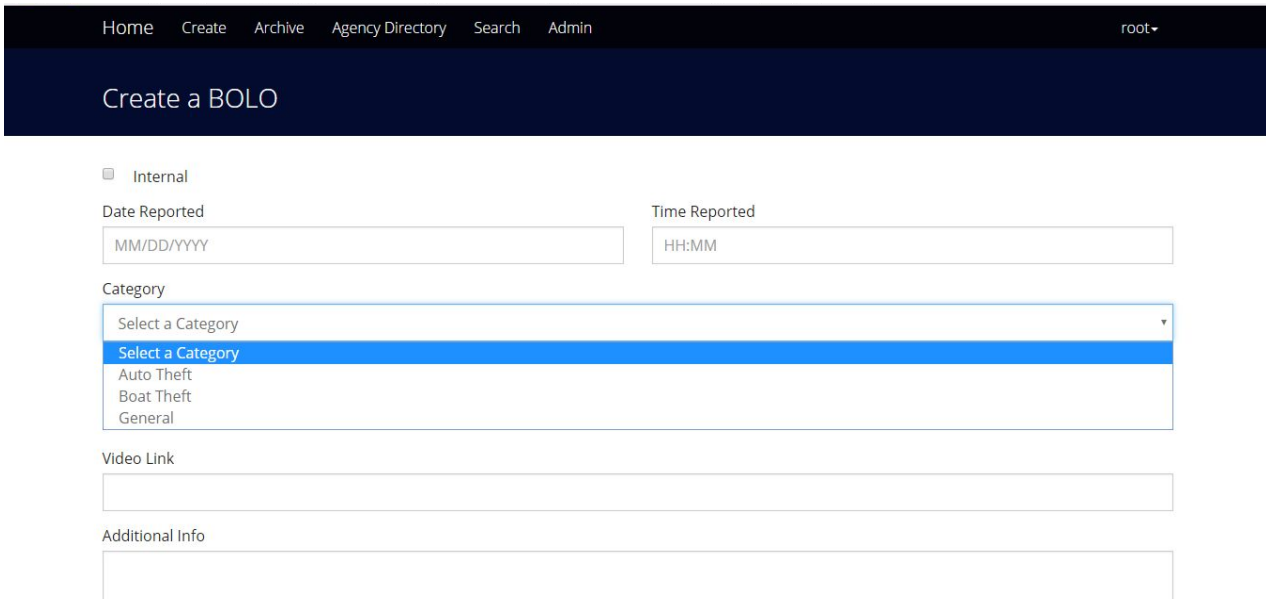


Figure 46: BOLO Category Rendering

UNCLASSIFIED// FOR OFFICIAL USE ONLY// LAW ENFORCEMENT SENSITIVE

FIU Police Department

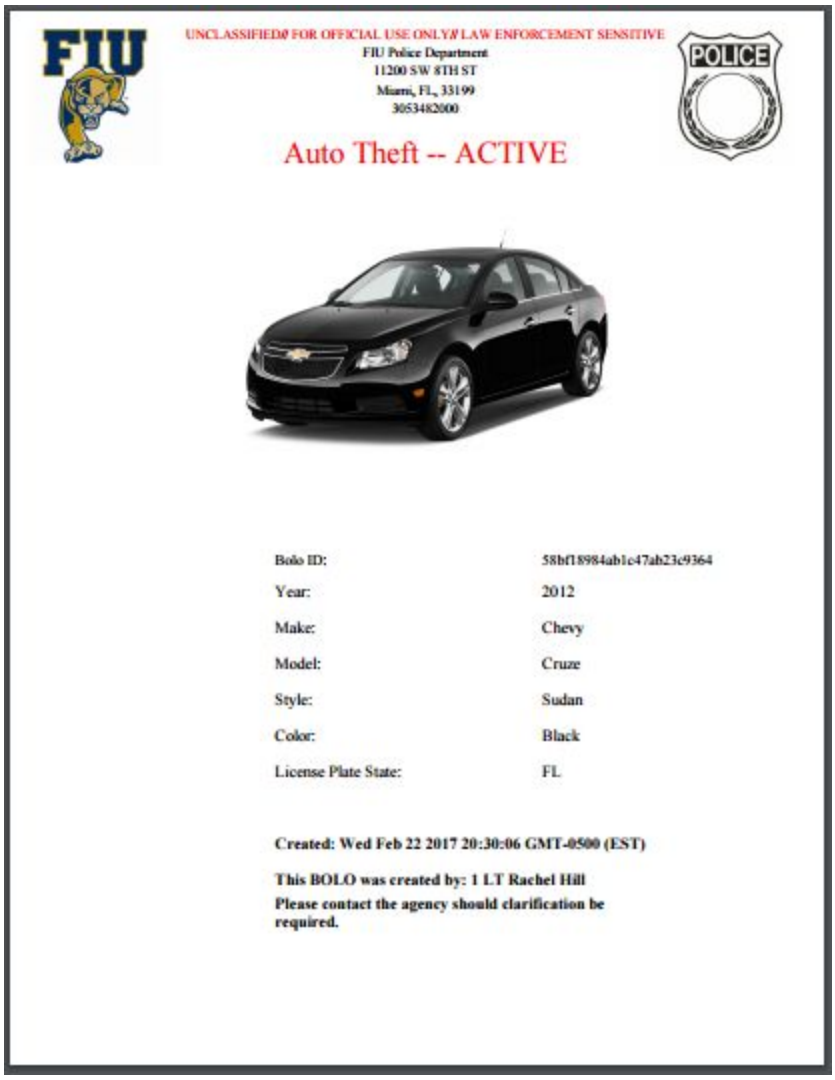
11200 SW 8TH ST

Miami, FL.

3053482000

Auto Theft -- ACTIVE

Figure 47: BOLO Details View


UNCLASSIFIED// FOR OFFICIAL USE ONLY// LAW ENFORCEMENT SENSITIVE

FIU Police Department
11200 SW 8TH ST
Miami, FL, 33199
3053482000

POLICE

Auto Theft -- ACTIVE

Bolo ID:	58bf18984ab1c47ab23c9364
Year:	2012
Make:	Chevy
Model:	Cruze
Style:	Sudan
Color:	Black
License Plate State:	FL

Created: Wed Feb 22 2017 20:30:06 GMT-0500 (EST)

This BOLO was created by: 1 LT Rachel Hill
Please contact the agency should clarification be required.

Figure 48: BOLO PDF Rendering

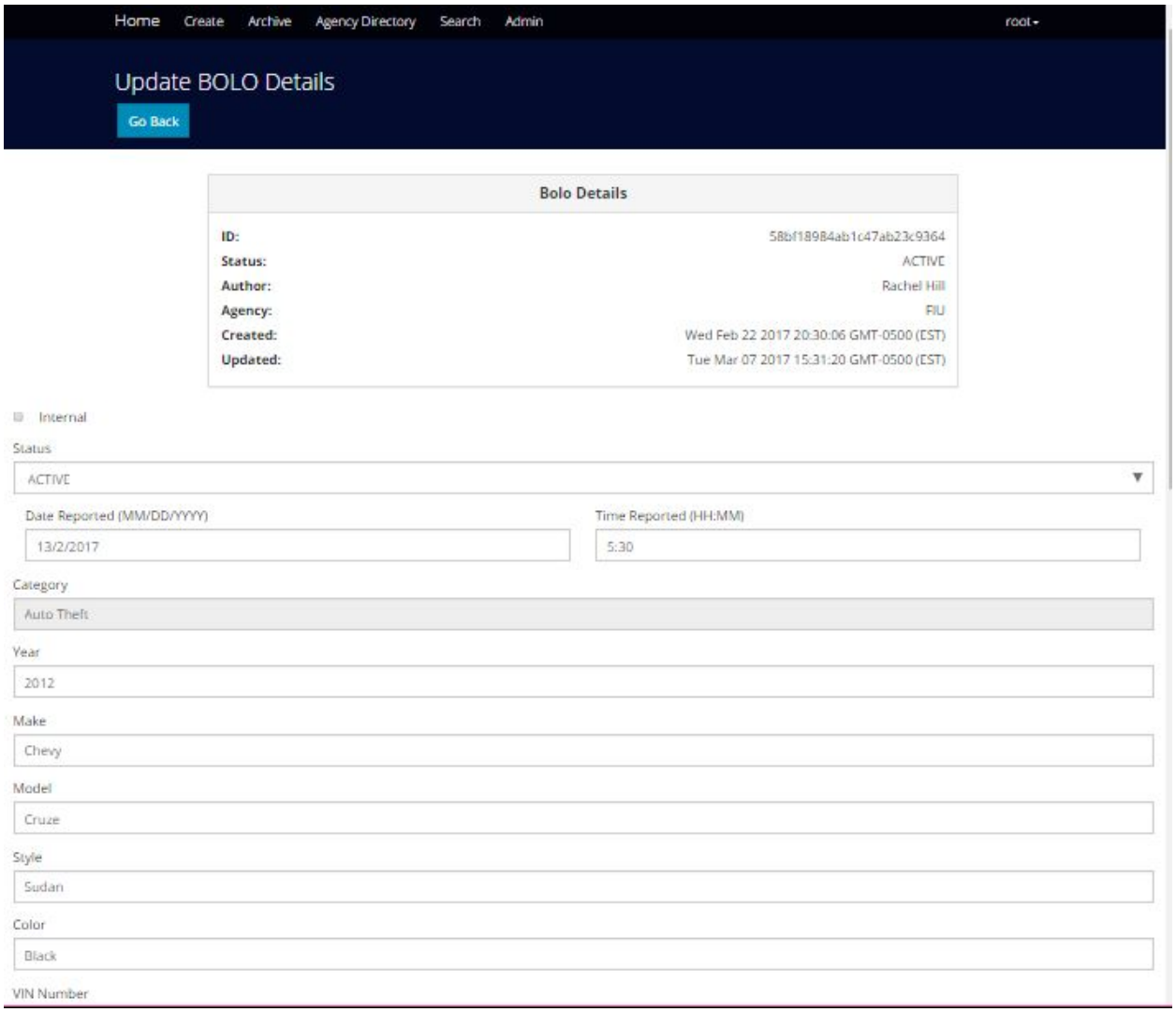


Figure 49: Edit Bolo

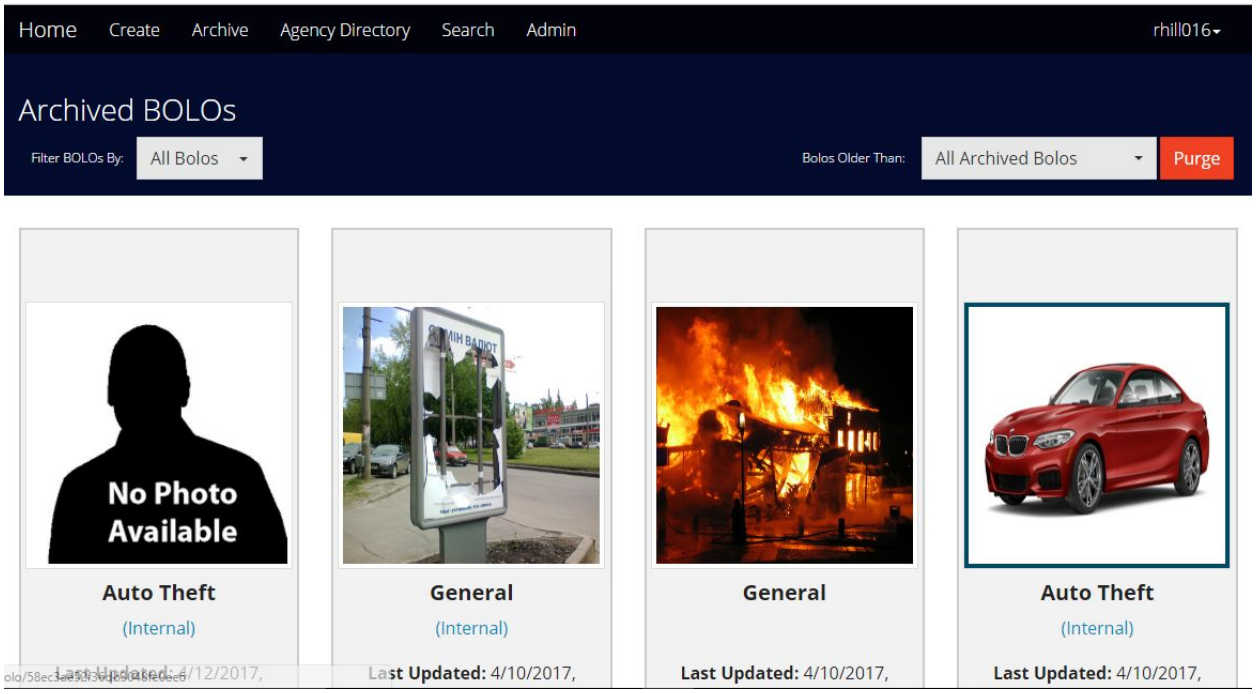


Figure 50: Archived Bolos

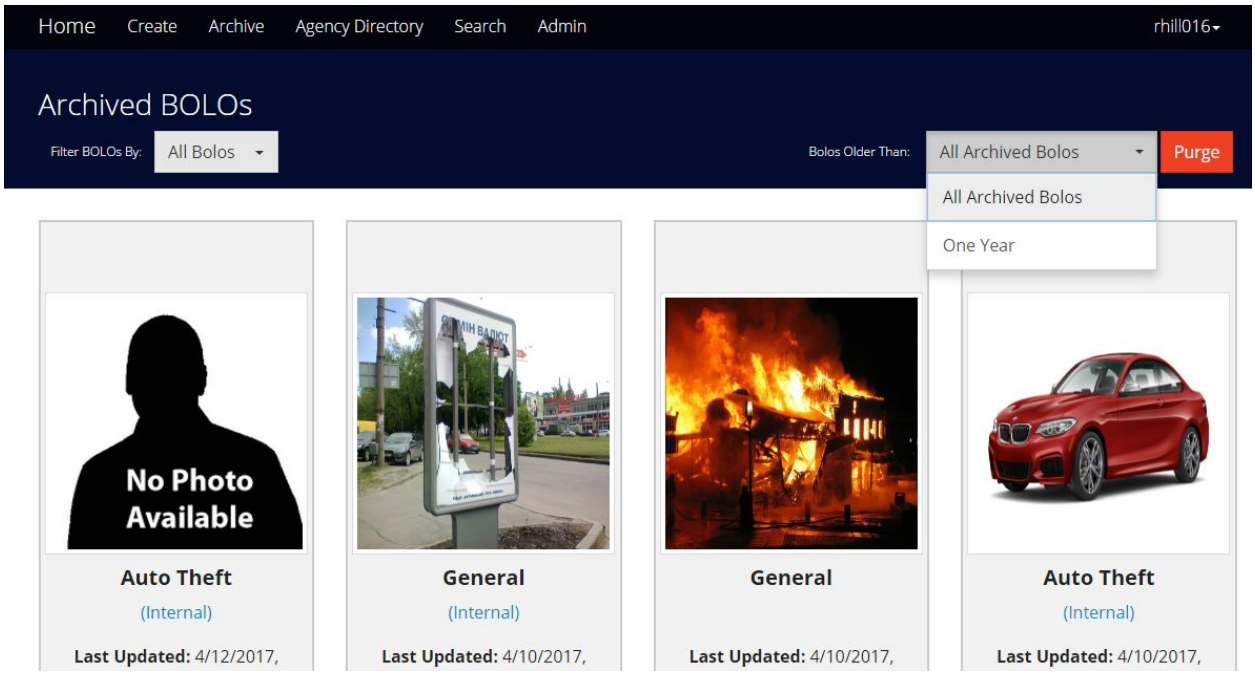


Figure 51: Purge Archived BOLOs

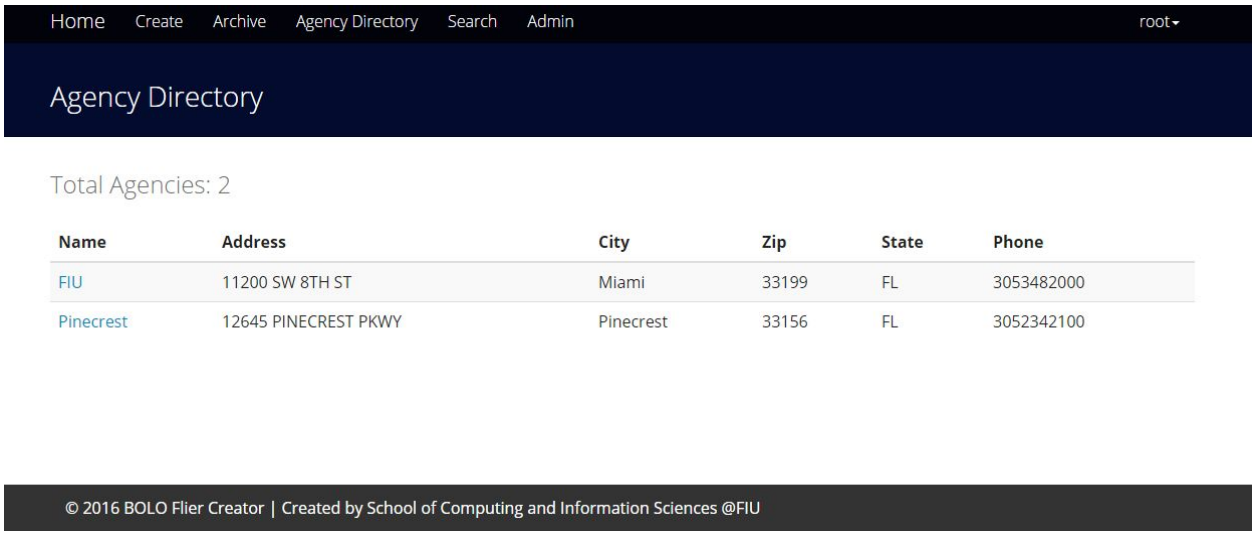


Figure 52: Agency Directory

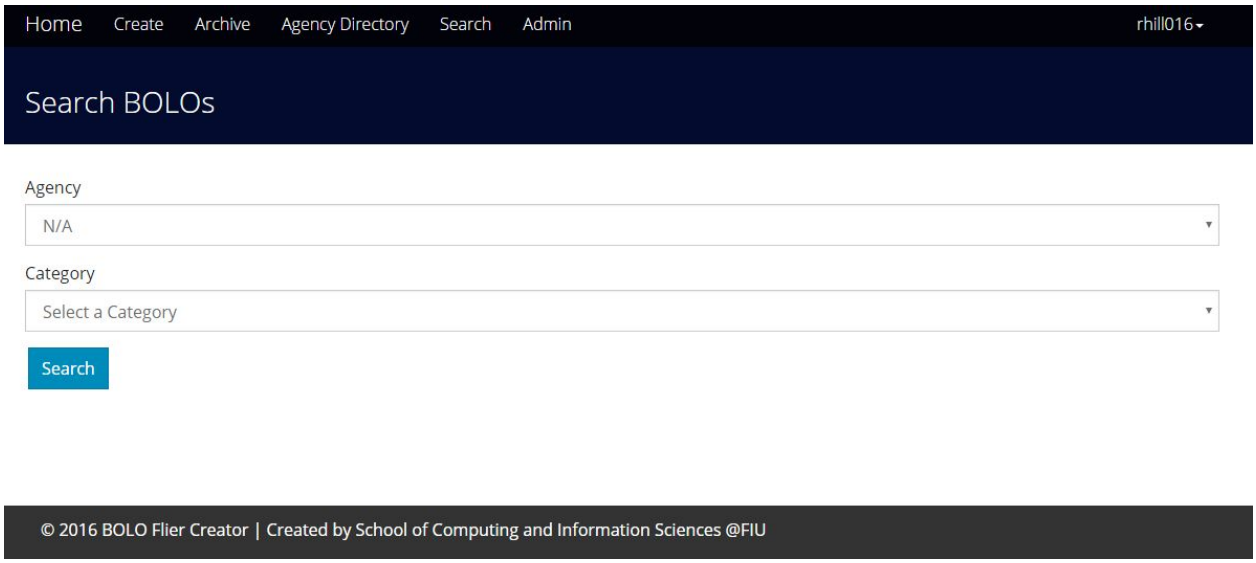


Figure 53: Search Bolo

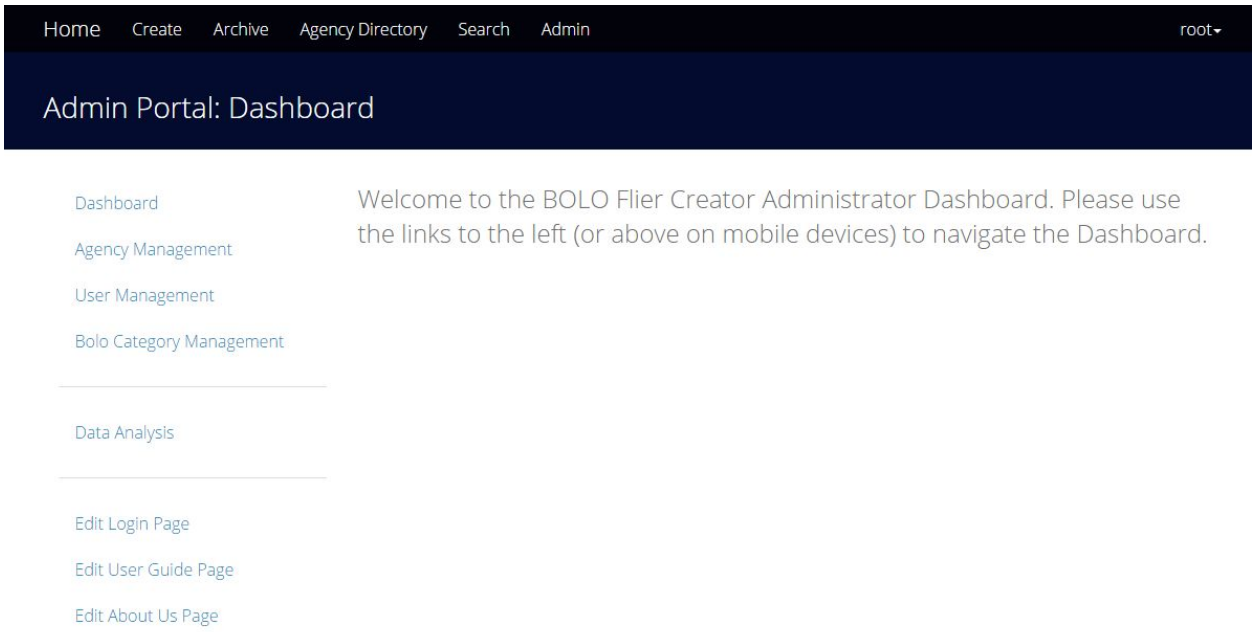


Figure 54: Admin Dashboard

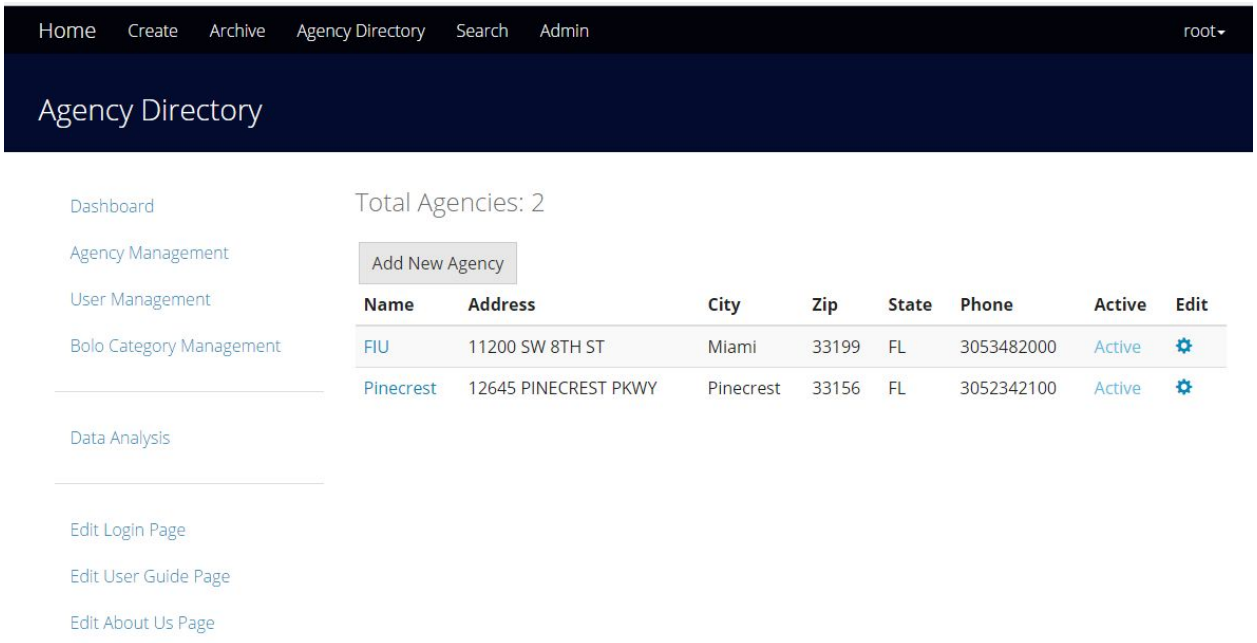


Figure 55: Agency Management

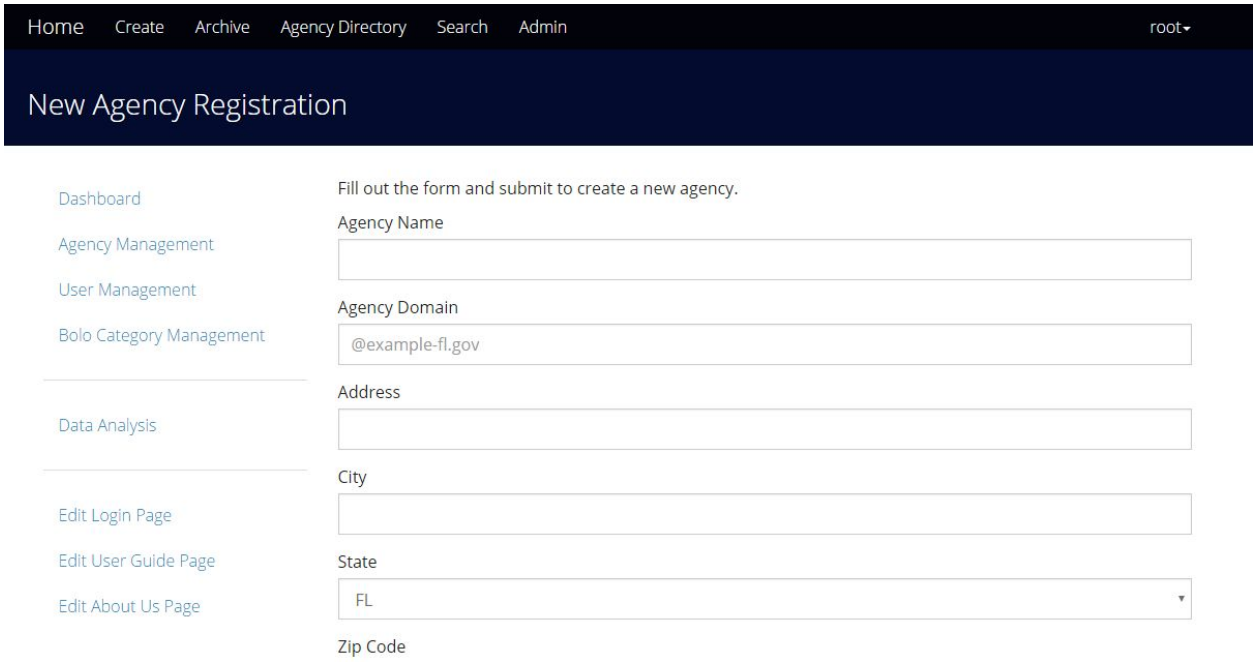


Figure 56: Create New Agency

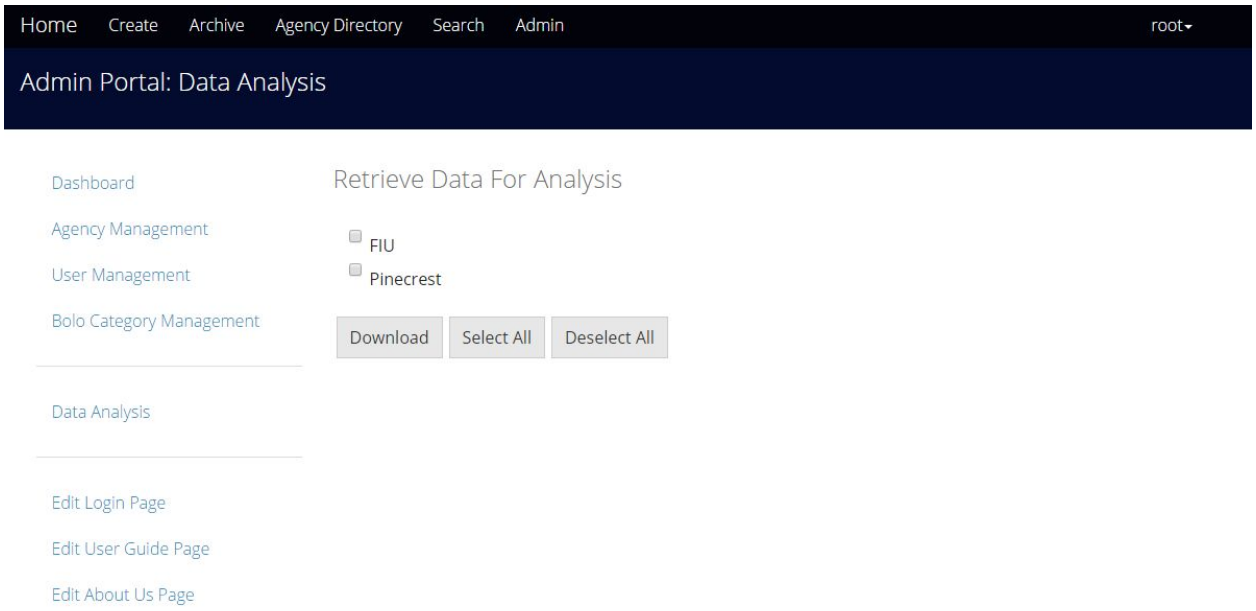


Figure 57: Data Analysis

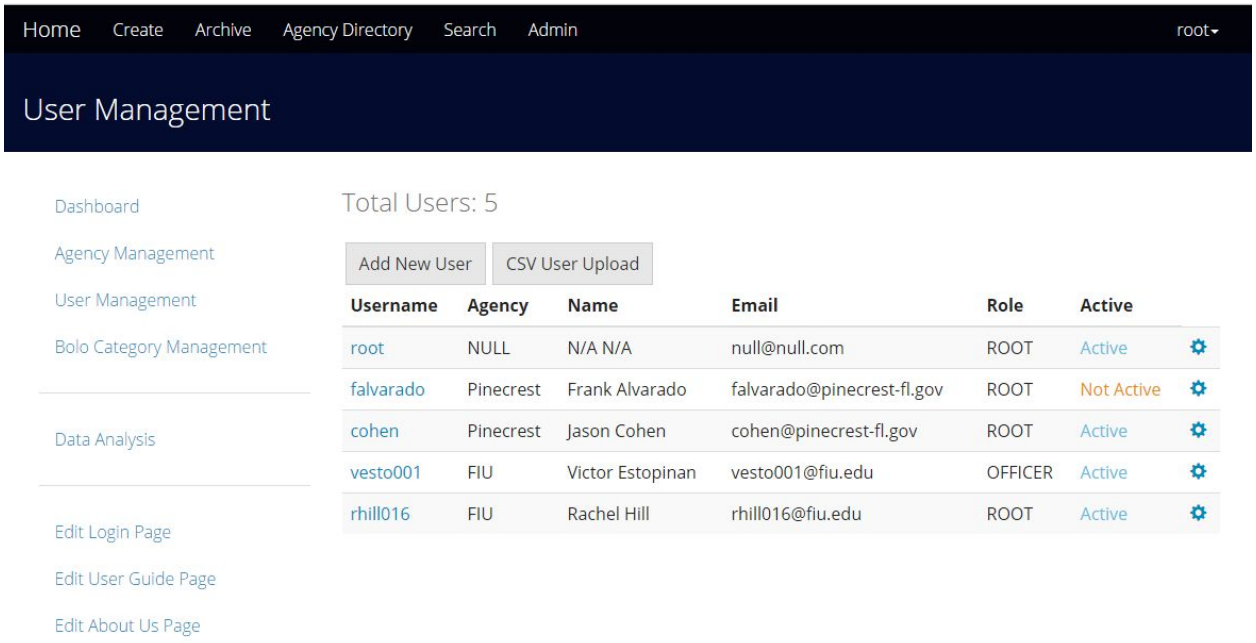


Figure 58: User Management

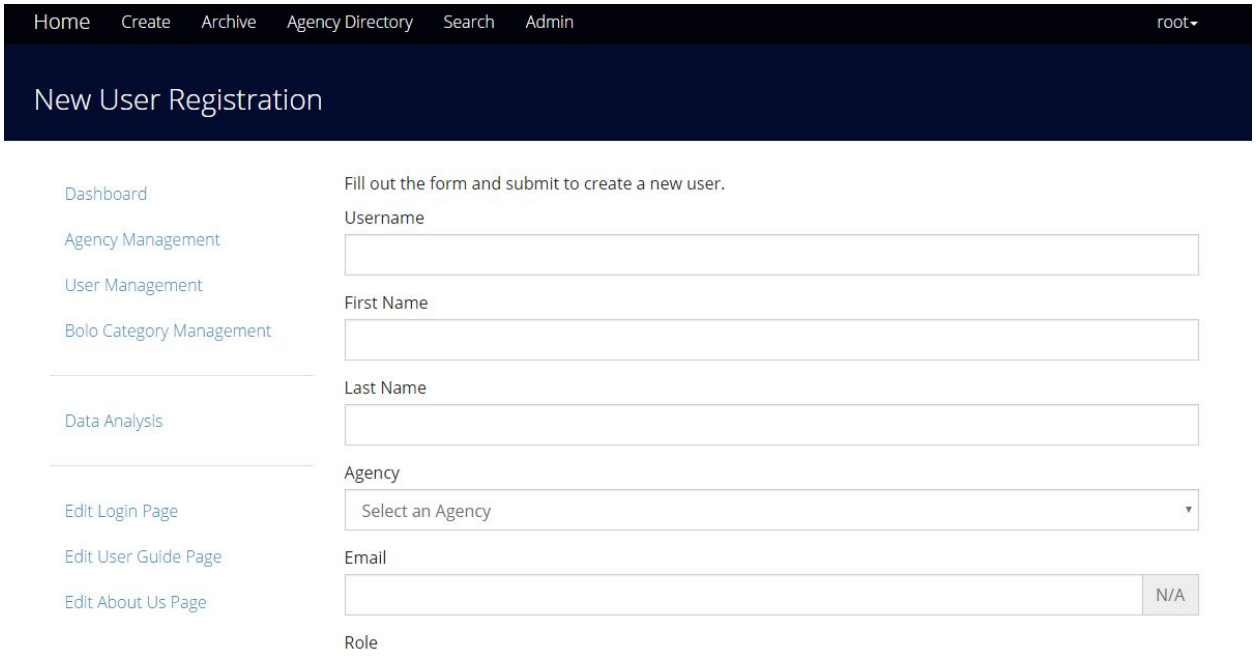


Figure 59: Create New User

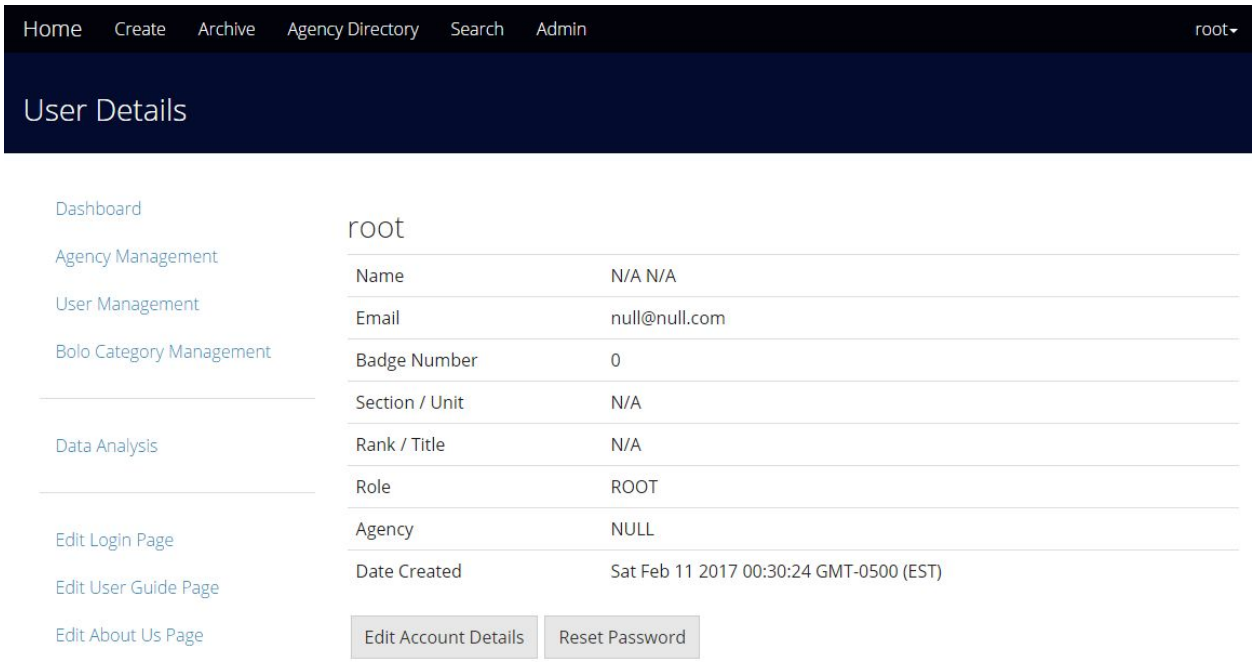


Figure 60: Account Details

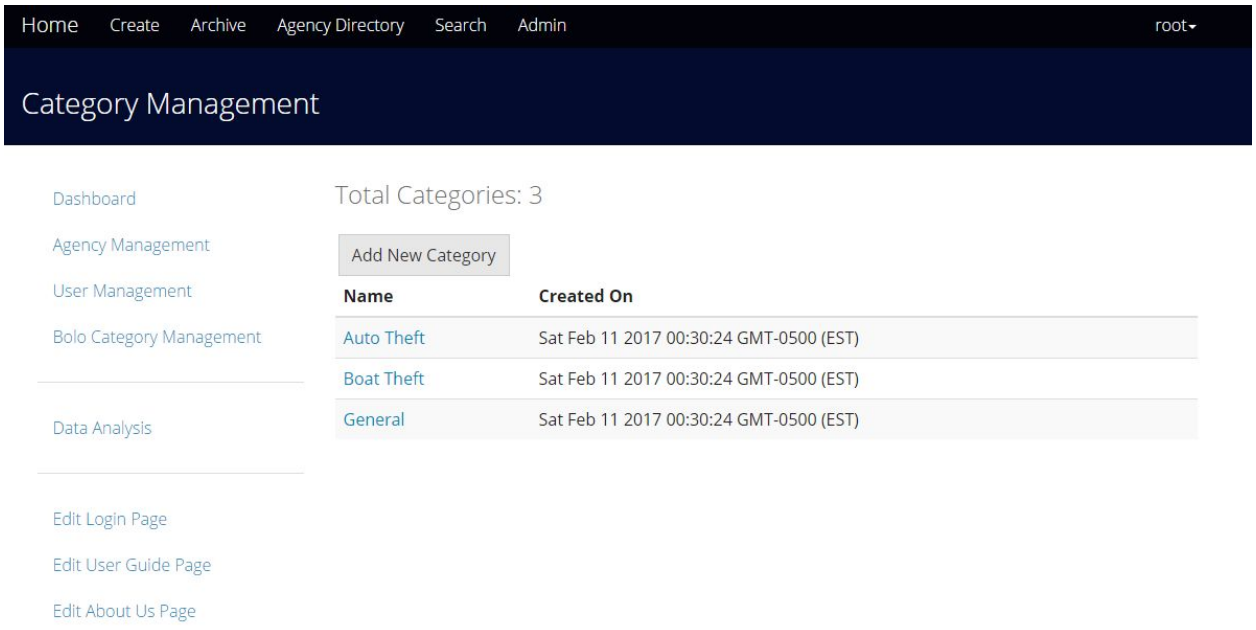


Figure 61: Category Management

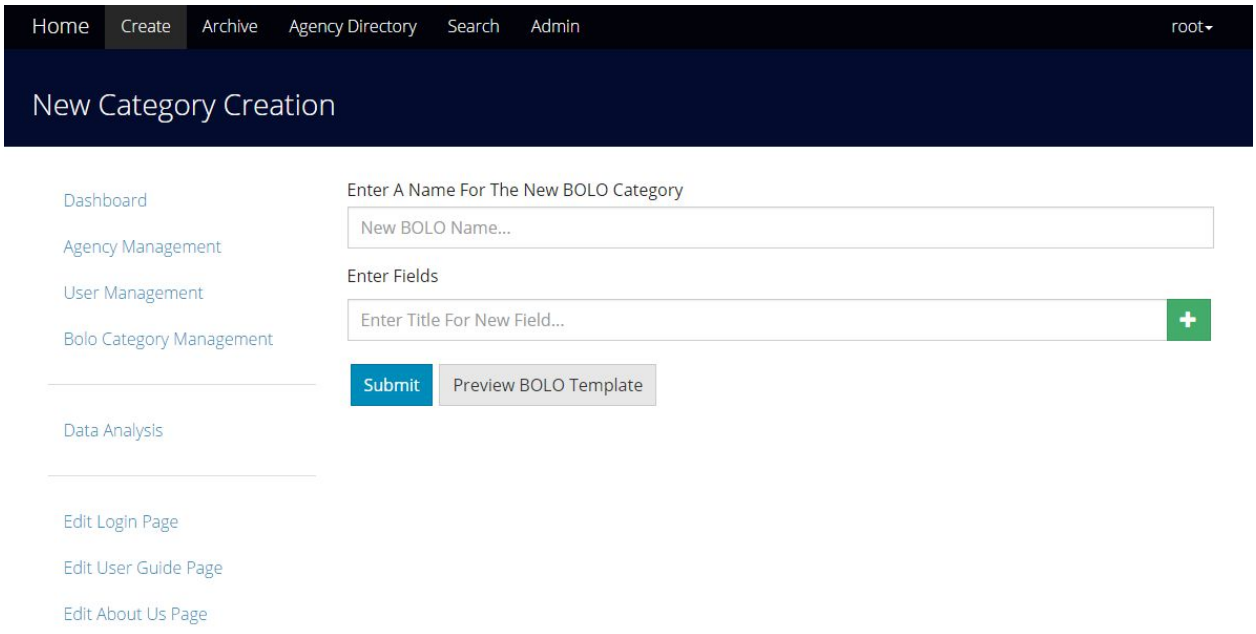


Figure 62: Category Creation

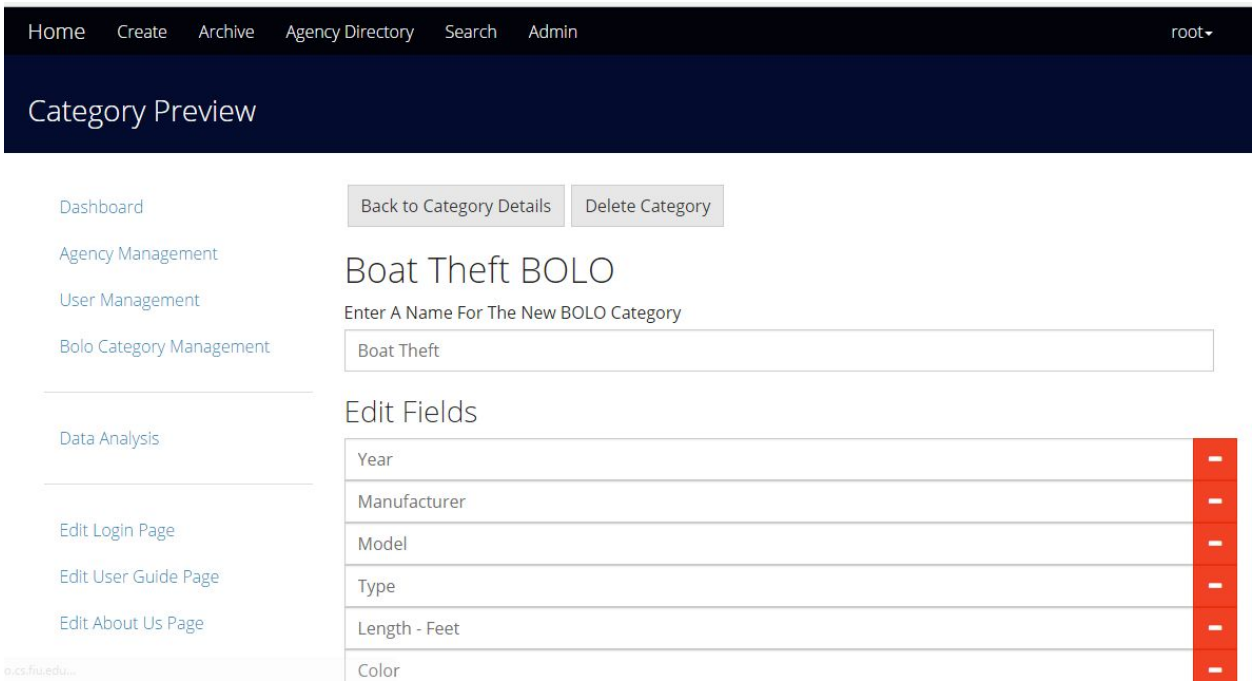


Figure 63: Category Preview

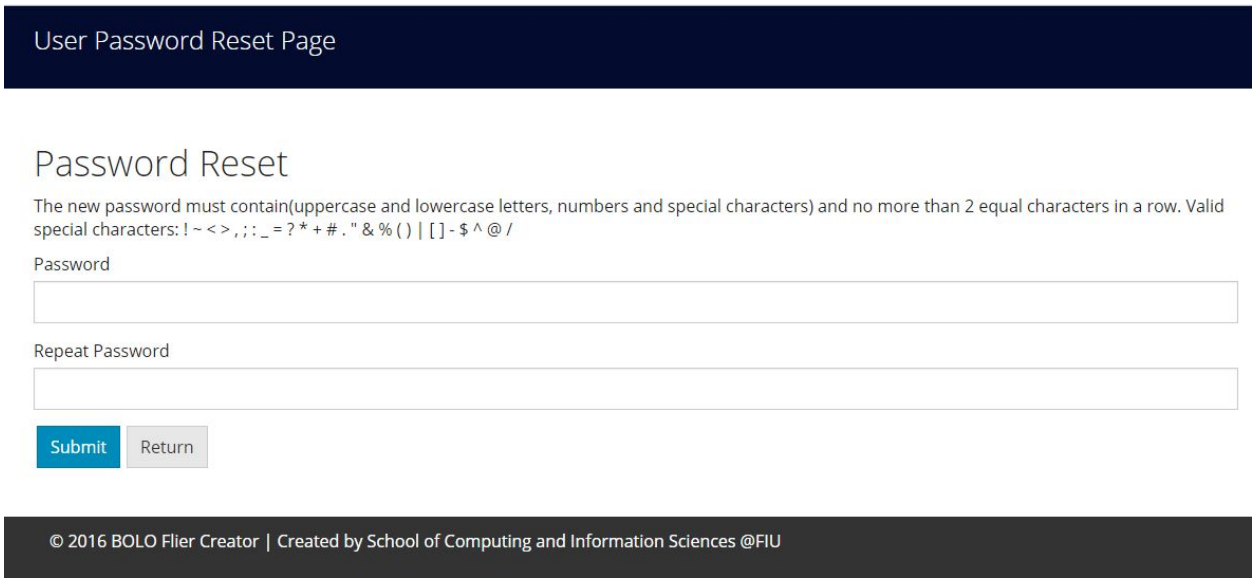


Figure 64: Change Password

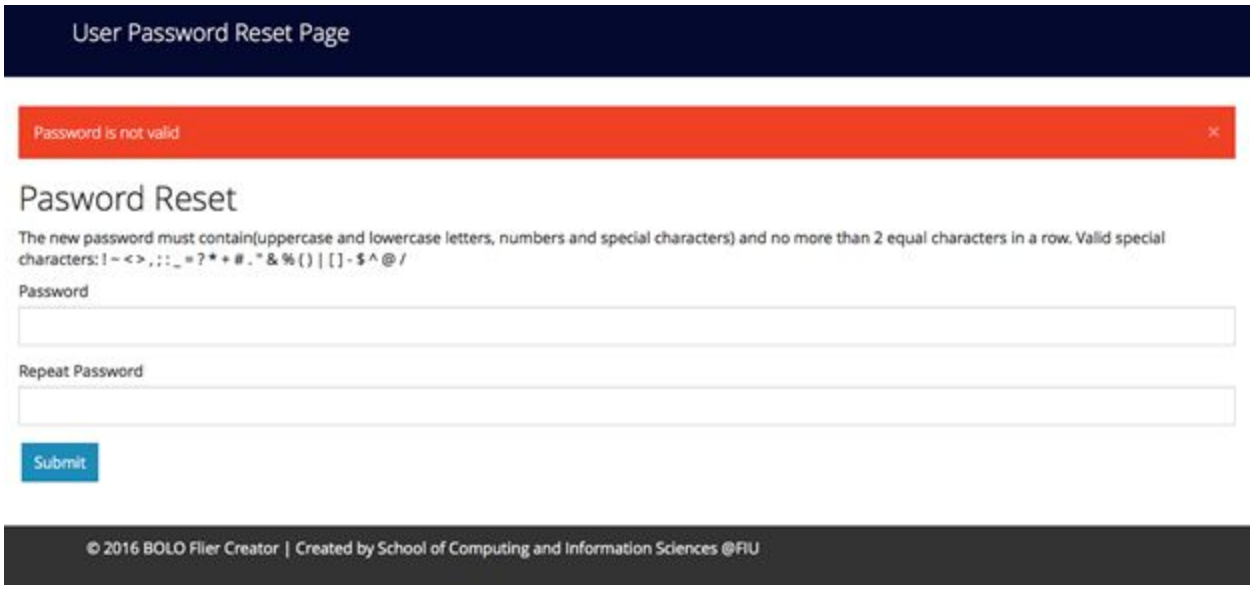


Figure 65: Password Verification

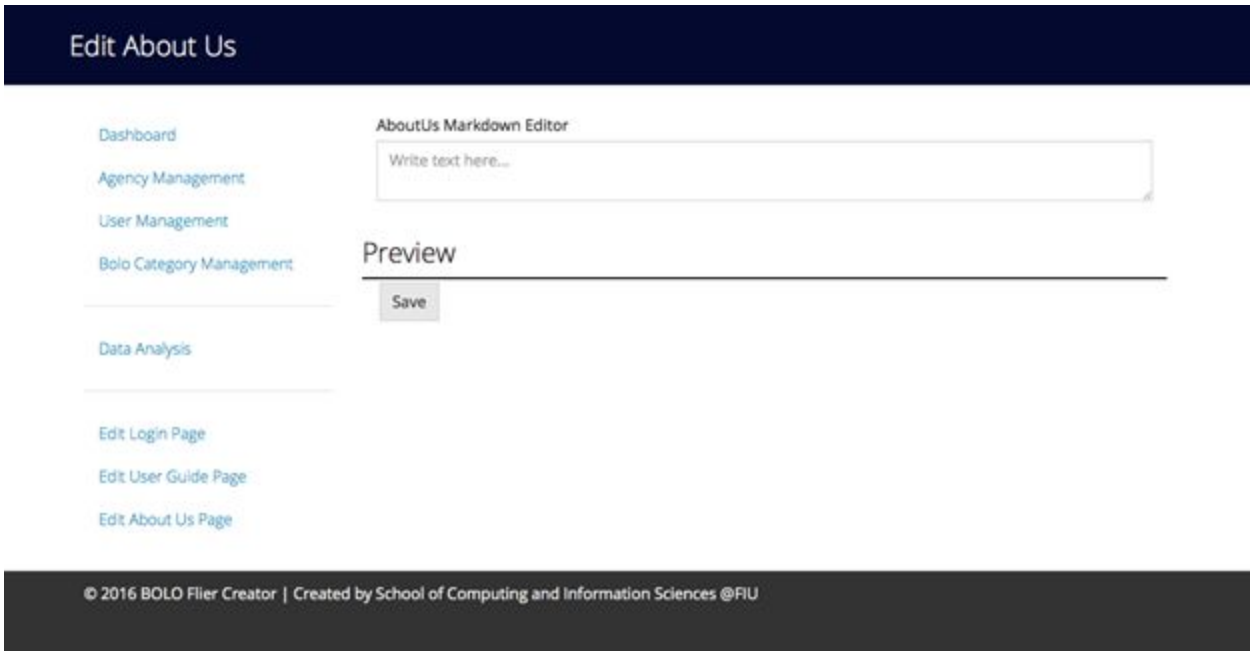


Figure 66: Edit About Us

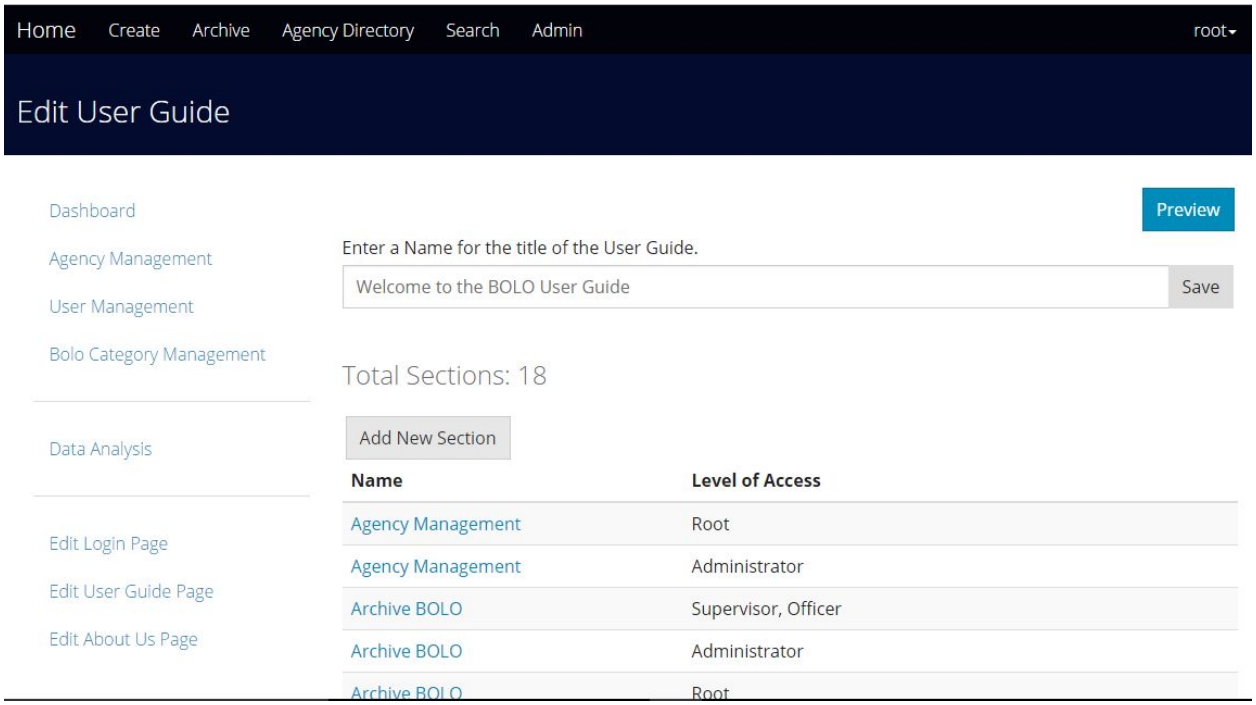


Figure 67: Edit User Guide

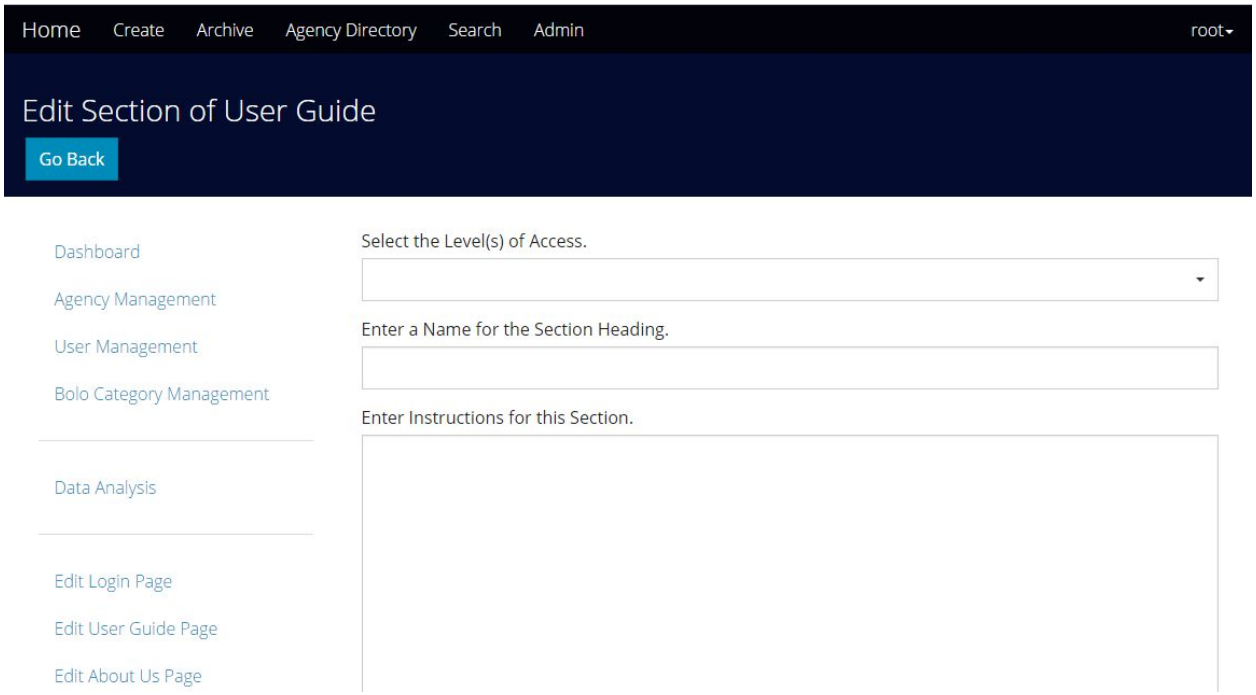


Figure 68: Add User Guide Section

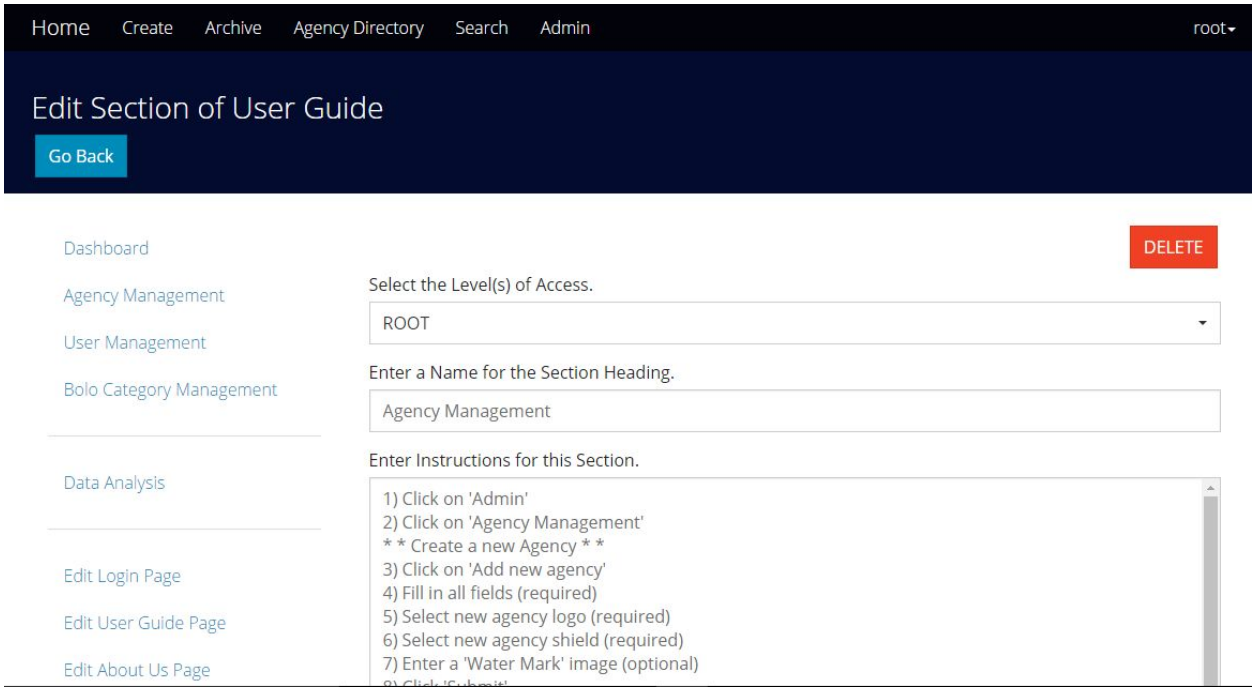


Figure 69: Edit User Guide Section

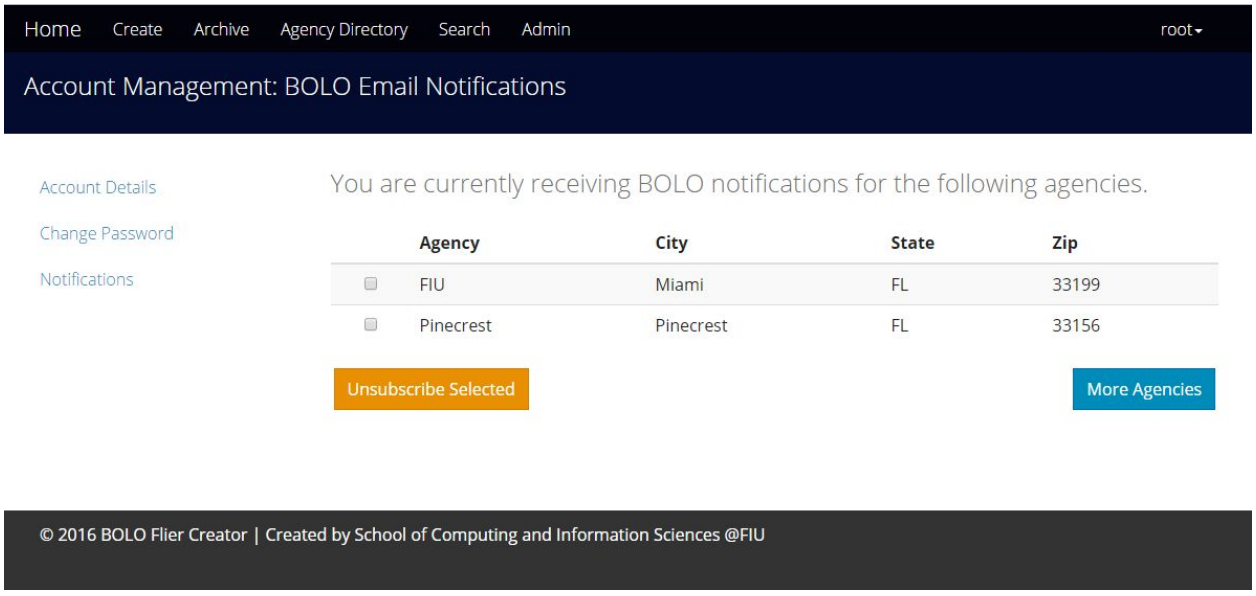


Figure 70: Subscribe/Unsubscribe to Notifications

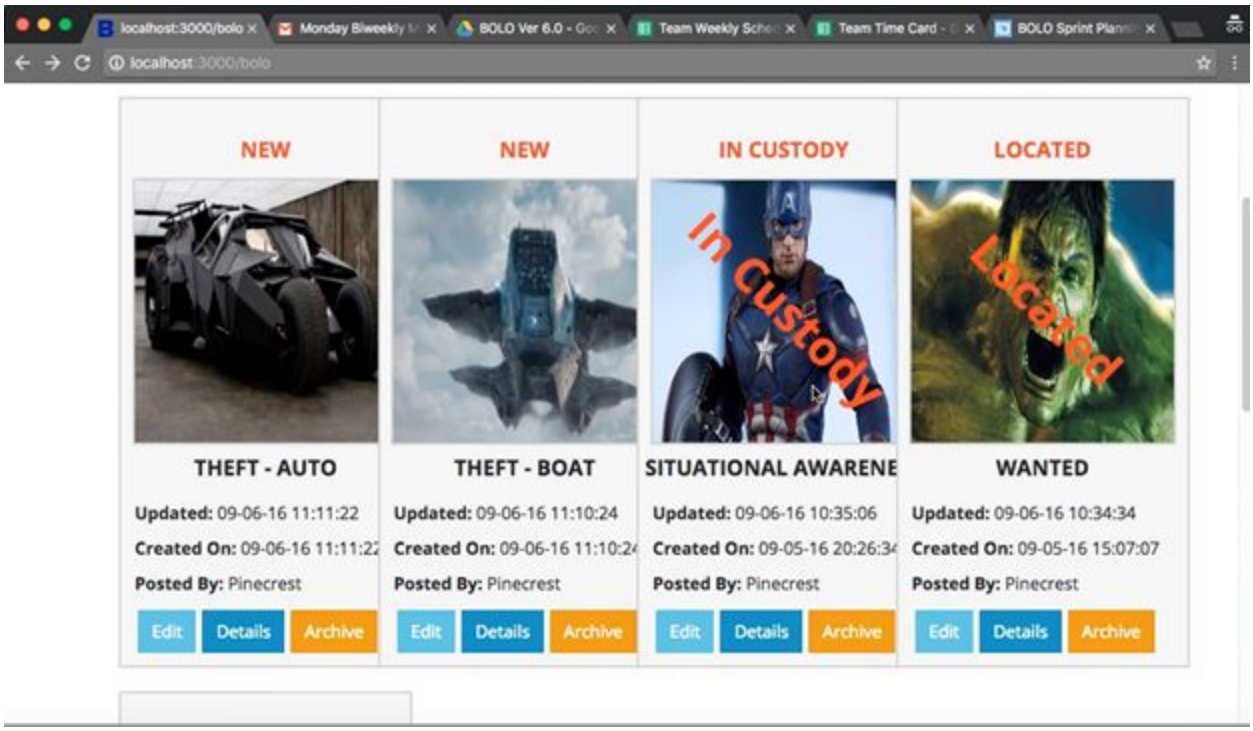


Figure 71: BOLO Watermark

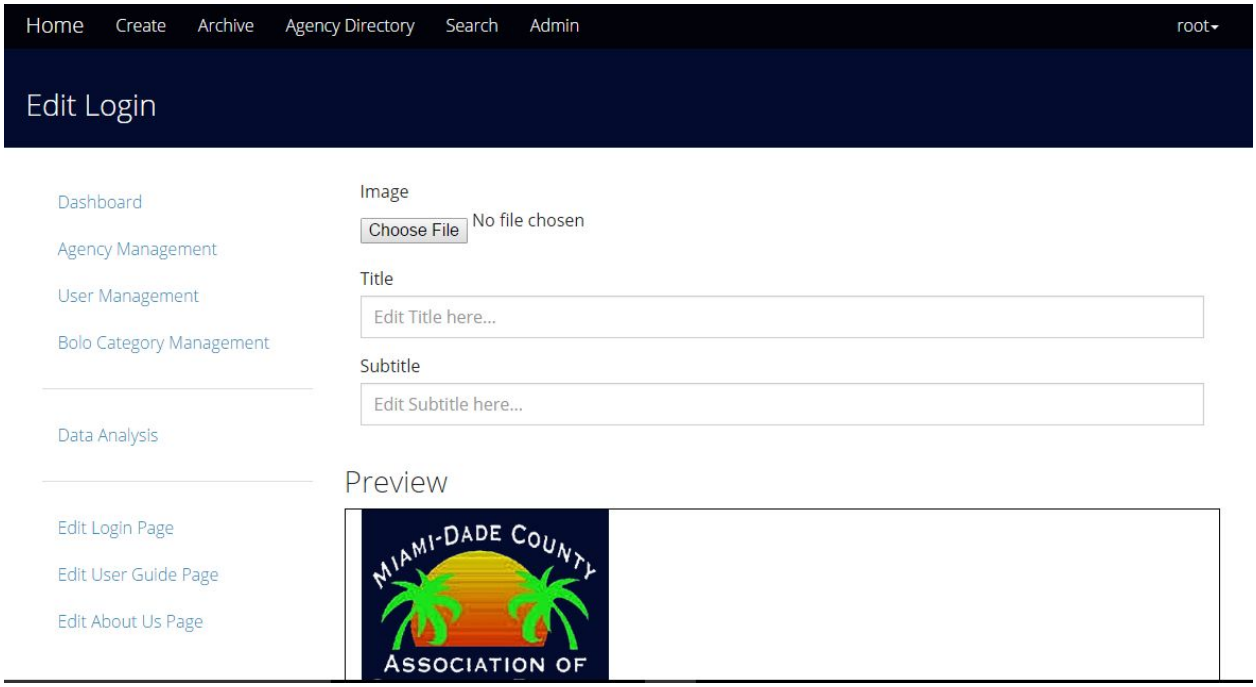


Figure 72: Edit Login Page

Appendix C - Sprint Reports

Sprint Review

Sprint 1

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 3:30 PM
End time: 4:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #1215 Setup the BOLO-6.0 environment.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- No implementations were rejected

Sprint 2

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 3:30 PM
End time: 4:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #1217 Fix BOLO PDF
- User Story #1218 Search Issue in the system
- User Story #1222 Password validation
- User Story #1246 Prevent server from crashing when creating a new bolo without selecting a category.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #1221 Setup bolo.cs.fiu.edu server.
    - The server keeps redirecting to http://localhost:3000.

Sprint 3

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 3:30 PM
End time: 4:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #1219 - Prevent BOLO from disappearing when updated
- User Story #1221 Setup bolo.cs.fiu.edu server.
- User Story #1265 Subscribing to Agency

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- No implementations were rejected

Sprint 4

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 2:30 pm
End time: 3:00 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #1286 Notifying Users Subscribed to Agencies
- User Story #1287 Edit User Guide Page

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #1288 Edit Login Page
    - Bugs were found during demo.

Sprint 5

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 2:00 PM

End time: 2:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #1315 Forgot Password
- User Story #1316 Add Delete Option to Categories
- User Story #1288 Edit Login Page

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- No implementations were rejected

Sprint 6

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 2:00 PM
End time: 2:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story # 1328 Add internal option to create BOLO form
- User Story # 1337 Change date to US standard
- User Story # 1338 Add a purge date range to delete archived bolos
- User Story #1357 Impede new users from navigating away from page after entering temporary password
- User Story #1336 Reduce size of Officer's badge panel

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- No implementations were rejected

Sprint Retrospective

Sprint 1

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 11:00 AM
End time: 11:30 AM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - N/A
- Did we do a good job estimating the points (time required) for each user story?
    - Yes
- Did each team member work as scheduled?
    - Yes

What went right?

- Each team member was able to have a working BOLO system on their local computers with SendGrid configured properly.

How to address the issues in the next sprint?

- How to improve the process?
    - We should have met with the product owners earlier.
    - We should have reported issues to our product owners as well as our professor earlier.
- How to improve the product?
    - We can improve on the product by creating an instruction manual with clear steps on how to solve the certain issues we ran into.

Sprint 2

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 11:00 AM
End time: 11:30 AM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - Yes
- Did we do a good job estimating the points (time required) for each user story?
    - Our estimation was below the actual time it took us to complete an user story.
- Did each team member work as scheduled?
    - Yes

What went right?

- We completed most user stories planned for this sprint.

How to address the issues in the next sprint?

- How to improve the process?
    - Working on documentation earlier
- How to improve the product?
    - Searching for bugs and doing constant testing
    - Leaving behind detailed notes and comments

Sprint 3

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 3:00 PM
End time: 3:30 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - Yes
- Did we do a good job estimating the points (time required) for each user story?
    - Our estimation fell below the actual time it took us to complete an user story.
- Did each team member work as scheduled?
    - Yes

What went right?

- We completed most user stories planned for this sprint.

- We were able to get the BOLO system running on the FIU server.
- We were able to fix most bugs found on the system.

How to address the issues in the next sprint?

- How to improve the process?
    - We could up try to communicate better when help is needed on a User Story.
- How to improve the product?
    - Searching for bugs and doing constant testing
    - Leaving behind detailed notes and comments

Sprint 4

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Start time: 3:00 PM
End time: 3:30 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - Yes
- Did we do a good job estimating the points (time required) for each user story?
    - Yes; however, some stories required overtime.
- Did each team member work as scheduled?
    - Yes

What went right?

- We were able to find and fix small bugs
- We agreed that the system is stable enough to be migrated into production
- We improved on the previous design of the system
- We added functionality to allow the root user to edit the User Guide

How to address the issues in the next sprint?

- How to improve the process?
    - We could improve on our test cases to find more bugs
    - We could divide huge user stories into multiple ones in Mingle
- How to improve the product?

- We could discuss possible features to add
- We could add a new feature to allow users to contact the developers when any bugs or undesirable functionalities are found

Sprint 5

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Product Owners: Jason Cohen, Frank Alvarado
Start time: 2:30 PM
End time: 3:30 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
  - Yes
- Did we do a good job estimating the points (time required) for each user story?
  - Yes
- Did each team member work as scheduled?
  - Yes

What went right?

- We were able to find and fix small bugs
- We installed BOLO7 on the Pinecrest Police Station server
- We improved on the previous design of the system
- We added functionality to allow the user to delete a bolo category in the category management

How to address the issues in the next sprint?

- How to improve the process?
  - We could improve on our test cases to find more bugs
- How to improve the product?
  - We could add a new feature to allow users to create a BOLO that is only visible within the agency.
  - We could add better error messages to the forms.

Sprint 6

Attendees: Rachel Hill, Victor Estopinan, Mario Siu
Product Owners: Jason Cohen, Frank Alvarado
Start time: 2:30 PM
End time: 3:30 PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - Yes
- Did we do a good job estimating the points (time required) for each user story?
    - Yes
- Did each team member work as scheduled?
    - Yes

What went right?

- The internal option for a BOLO was created
- The Filter Archived BOLO was set in the system
- The Officer’s badge panel was set to the same size of the email’s panel
- The date while creating or updating a BOLO was set to the US standard MM/DD/YYYY
- New users were prevented from navigating away from the page after entering a temporary password.
- Root users now have the option to select a date range to delete archived BOLOs.

How to address the issues in the next sprint?

- How to improve the process?
    - We could demo the product during in-person meetings.
- How to improve the product?
    - We could perform daily testing to discover hidden bugs in the system.

Appendix D - Other Documents

User Guide

Agency management

1) Click on 'Admin'
2) Click on 'Agency Management'
3) Click on the 'Blue Asterisk' to the right on the agency
4) Enter or Modify any fields (required)
5) Enter or update 'Logo' image (required)
6) Enter or update 'Shield' image (required)
7) Enter or update 'Watermark' image(optional)
8)Choose to allow officers to opt out of viewing BOLOs from their agency
9) Click 'Submit'

Category management

1) Click on 'Admin'
2) Click on “BOLO Category Management”
3) To create a new Category:
    1. Click on “Add New Category”
    2. Fill out the Category Name
    3. Add as many fields as you need by clicking on the green “+” symbol
    4. Delete fields by clicking on the red “-“ symbol
    5. Once you have finished creating the fields of the new Category click Preview
    6. Review the Category and go back if you want to make any changes
    7. Once you are satisfied with the Category click Submit and the Category will be created
4) To delete a category:
    1. Select the category you want to delete
    2. Click on “Delete Category”

3. Category is now deleted

5) Please Note that once a Category is being used by any BOLO, it can't be deleted.
6) Editing Categories should only be used if no BOLOs are using the category
    1. Select on the Category you want to edit
    2. Make changes as you see fit and then click submit
    3. The Category is now updated

Editing Login Page

1) Click on 'Admin'
2) Click on "Edit Login Page"
3) Fill 'Title' and 'Subtitle'
4) Copy desired image on C:\BOLO-7.0\Code\src\public\img folder
5) Click browse and select copied image
6) Click 'Save'

Editing User Guide Page

1) Click on 'Admin'
2) Click on "Edit User Guide Page"
3) Fill 'Name' of user guide
4) Click on 'Add New Section'
5) Select 'Level Access' and fill 'Name Heading' and 'Instructions'
6) Click 'Save'

User Management

1) Click on 'Admin'
2) Click on 'User Management'

* * Add a new user's profile * *

3) Click on 'Add New User'
4) Fill in the empty fields(required)
5) Click 'Submit'

* * Add a new user's profile * *

3) Click on the 'Blue Asterisk' on the right of the user
4) Update required fields
5) Click 'Submit'

Create a BOLO

Click 'Create'
Fill In the Date Reported and Time Reported Fields

1) Select a Category
2) The Form will now be updated with the proper fields
3) Fill out all the required fields for that category
4) Upload any images that you might have, max is 3
5) Fill out the Video Link, Additional Info and Summary fields, if applicable
6) Select Preview
7) Go over the BOLO to ensure it looks exactly how you want it to
8) Repeat steps 1-8 until you are satisfied with the BOLO
9) Click Submit
10) Go to your registered email and click on the confirmation to finish creating your BOLO flier

Edit a BOLO

* * You can only edit your BOLOs * *
1) Click 'Edit'
2) Select any field you want to edit
3) Enter any information required
4) Click 'Submit'

View Details of a BOLO

1) Click 'Details'
2) Review details of a selected BOLO
3) Select 'Generate PDF' to download
* * View a detailed record of who has edit this BOLO * *
5) Select 'View Record'
6) Click 'Back' to go back to the BOLO details

Filter by (Me)

1) Click 'Filter By'
2) Select filter:
   * My BOLO
   * My Agency
   * Internal
   * All BOLOs
3) Results BOLOs will display

Filter by Agency

1) Click 'Select Agency'
2) Select one or more agencies
3) Click 'Submit'
4) Results BOLOs will display

Archive

1) Click 'Archive
* * At the top of the page* *
2) Select a BOLO
3) Click:
  * 'View' to view details
  * 'Restore' to restore the BOLO
  * 'Delete' to delete a BOLO
* * On the BOLO * *
4) Select filter to purge BOLOs by date:
  * All Archived Bolos
  * One Year
  * Two Years
  * Three Years
  * Four Years
  * Five Years
  * Six Years
  * Seven Years
5) Click ‘Purge’ to delete BOLOs from archive by selected range

Agency Directory

1) Click on 'Agency Directory' to view a list of partnering agencies.
2) Select an agency by clicking their name
  * View contact information
  * Verify an agency's 'Logo' and 'Shield

Search BOLO by field(s)

1) Click 'Search'
2) Select any field(s) you wish to search.
3) Enter information value of field(s).

* *Wildcard search is for something you can not search with an optional search * *
4) Click 'Search'

Get Notifications
1) Click on your 'Username'
2) Click on 'Account' from the dropdown
3) Click on 'Notifications'
4) Click on 'More Agencies' at the bottom left to subscribe to other agencies
5) Select agencies to receive notifications
* *You may also unsubscribe from an agency * *
6) Select agencies to remove from your list of Agencies
7) Click 'Unsubscribe Selected'

Reset Password
1) Click on your 'Username'
2) Click on 'Settings' from the dropdown
3) Click 'Change Password'
4) Enter new password
5) Enter confirmation password
6) Click 'Submit'

Installation / Maintenance

1. You are going to need a GitHub account, NodeJS, MongoDB, Git tools, a Sendgrid API key and an IDE.
2. Install NodeJs, MongoDB, your IDE and Git tools.
3. Go to GitHub and go to the project repository and get the clone link
4. Open up a bash shell, and type in "*git clone*" followed by the link from the repository.
5. While the project is downloading, set up MongoDB so that it points to the local host.
6. Once the repository is cloned, use a CLI, either a command prompt or a bash shell, to navigate to the project folder. Once there navigate to the "Code" folder, and type in "*npm install*". This will install all the dependencies onto your machine.
7. While that is installing you need to create your Sendgrid API key. Go to the Sendgrid website and make an account, it is free.

8. Once your account is setup, go to settings and find the option that says “Generate API Key”.
9. Follow the steps and generate the API Key. DO NOT NAVIGATE AWAY FROM THE WEBPAGE, unless you have copied the API key to a text document or your clipboard. It will disappear and you won’t be able to view it again.
10. Open up your IDE and then open up the project folder. Once it is loaded up, you are going to look for a file named .env.example. Make a copy of this file and name it .env. This file has a line that dedicated to the Sendgrid API Key, locate that line and past your Sendgrid API key into it.
11. Your dependencies should be finished installing by now, but if they haven’t just wait. Use either a bash shell or a command prompt and navigate to the MongoDB folder.
12. Ensure that you have root privileges or administrative rights, then navigate to the bin folder. Here type the command “*mongod*”, depending on your operating system you might need to use the command *“./mongod*” instead. This will start the database.
13. Open a new bash shell or command prompt and navigate to the project folder. Once there navigate to the “Code/src” folder and start the web application by running “node app.js”. Your application should start with no issues and now you are ready to begin using and or coding for the BOLO Flier Creator web application.
14. If you get an error please ensure that MongoDB is running, that no other service is using port 3000, and also that all your dependencies are installed correctly.

Shortcomings and Wish list

While working on the 7th iteration of the BOLO Flier Creator, we were able to add few new features as well as improve on the functionality to an already usable piece of software, however there were some desired features that were left unimplemented.

In future iterations of the BOLO Flier Creator web application we would like to see the following additions/improvements:

- Transition away from using Sendgrid or any other third party services. This is due to security risks and policies that are being observed by the product owners.
- When rendering the BOLO as a PDF the document looks a little lackluster, although a watermark can be uploaded, maybe a border will help the document look like a proper law enforcement document.

- It would also be very useful to add system settings that the root or administrator can utilize.
- A system wide log that can be seen by the root users would also be useful. It would contain information such as which users have been deleted/deactivated and by whom.
- A system wide data collection portal would also be useful for collecting statistics. For example, it can show how many BOLOs each user has created, which agency/department creates the most BOLOs, and when is the most common time that BOLOs are created.
- By separating the users belonging to different agencies, it would look more organized to the root user.
- It might be useful to create a new feature to allow users to leave comments about any issues that were discovered while utilizing the system.

Pending User Stories:

User Story #1361 - Create Logo Path

- As a root, I would like to change the logo image for the app.

Acceptance Criteria

- The root should get the logo image from any directory in the device
- The system should accept an image with any size

Use Case

- Name: ID # 1374 The logo image should be get from any directory by the root
- Actor: root
- Preconditions:
    - The user should has root access
    - The root should be Logged In
    - The root should get into the Admin menu
- Description :
    - The root should log In
    - The root should get into the Admin menu

- The root should click on the Edit Log In Page
- The root should change the logo of the app by uploading the image from any directory
- The root should save the changes by clicking in the SAVE button

Use Case

- Name:ID # 1303 Edit Logo of Login Page
- Actor:Root
- Preconditions:
  - The user should has root access
  - The root should be Logged In
  - The root should get into the Admin menu
- Description:
  - The root should log In
  - The root should get into the Admin menu
  - The root should click on the Edit Log In Page
  - The root should upload the image for logo and by clicking the change logo button
  - The root should save the changes by clicking in the SAVE button

Unit Test

- Test case ID: # 1374 The log image should be get from any directory by the root
- Description/Summary of Test: The log image should be get from any directory by the root
- Pre-condition:
  - The user should has root access
  - The root should be Logged In
  - The root should get into the Admin menu
- Expected Results: The root should change the logo in the login page
- Actual Result: The logo image did not change
- Status (Fail/Pass): Fail

References

Govett, Devon (2016) pdfkit source code (0.7.2) [Source code].
https://www.npmjs.com/package/pdfkit ,

Mueller, Patrick (2016) cfenv source code (Version 1.0.3) [Source code].
https://www.npmjs.com/package/body-parser ,

Smith, J (2016) cloudant source code (Version 1.4.1) [Source code].
https://www.npmjs.com/package/cloudant ,

Smith, J (2016) cfenv source code (Version 1.0.3) [Source code].
https://www.npmjs.com/package/cfenv ,

Wilson, Doug (2016) body-parser source code (Version 1.13.2) [Source code].
https://www.npmjs.com/package/body-parser

Libraries:

"bcrypt-nodejs": "^0.0.3",
"bluebird": "latest",
"body-parser": "^1.15.2",
"brfs": "^1.4.3",
"browserify": "^13.1.0",
"cfenv": "^1.0.3",
"connect-flash": "^0.1.1",
"cookie-parser": "^1.4.3",
"csurf": "^1.9.0",
"csv": "^1.1.0",
"csvtojson": "^1.0.3",
"dotenv": "^1.2.0",
"errorhandler": "1.4.1",
"express": "4.12.x",
"express-messages": "*",

```
  "express-session": "1.7.6",
  "express-validator": "*",
  "fast-csv": "^2.1.0",
  "gm": "^1.21.1",
  "hat": "0.0.3",
  "iframe": "^0.3.1",
  "imagemin": "^4.0.0",
  "j-i-c": "^2.0.2",
  "jimp": "^0.2.9",
  "json2csv": "^3.6.2",
  "jstransformer-markdown-it": "^1.0.0",
  "lodash": "^3.10.1",
  "marked": "latest",
  "method-override": "2.3.3",
  "moment": "*",
  "moment-timezone": "^0.5.5",
  "mongodb": "latest",
  "mongoose": "^4.6.3",
  "morgan": "1.6.1",
  "multer": "^1.2.0",
  "multiparty": "^4.1.2",
  "node-markdown": "^0.1.1",
  "node-phantom": "^0.2.5",
  "node-uuid": "^1.4.3",
  "node-wkhtml": "^0.1.0-alpha1",
  "nodepdf": "1.2.1",
  "number-to-words" : "latest",
  "passport": "^0.3.0",
  "passport-local": "^1.0.0",
  "pdfkit": "^0.7.2",
  "phantomjs": "^2.1.3",
  "promise": "^7.0.4",
  "pug": "^2.0.0-beta6",
  "request": "^2.64.0",
  "sendgrid": "^2.0.0",
  "serve-favicon": "^2.3.0",
  "socket.io": "^1.0.0",
  "socket.io-express-session": "^0.1.3",
```

```
  "validate.js": "^0.9.0",
  "wkhtmltopdf": "^0.1.5",
  "node": "0.12.x",
  "chai": "^3.3.0",
  "gulp": "^3.9.1",
  "gulp-confirm": "latest",
  "gulp-exec": "latest",
  "istanbul": "^0.3.22",
  "jsdoc": "^3.3.3",
  "jshint": "^2.8.0",
  "mocha": "^2.3.3",
  "mockery": "^1.4.0",
  "nock": "^2.13.0",
 "read-multiple-files" : "latest",
  "sinon": "^1.17.0"
```