Michigan State University

Team Auto-Owners

Catastrophic Claims Unit Mobilization

Project Plan

Spring 2016

Auto-Owners Staff:

Scott Lake

Sherry McKenzie

Dominic Mersino

Michigan State University Capstone Members:

Jason Steele

Nicole Lawrence

Nick Reuter

Matt Wiechec

Table of Contents

1. Executive Summary

Auto-Owners Insurance is a Fortune 500 company founded in 1916 and is the 15th largest insurer in the country. They have consistently received the highest rankings in growth, financial stability and claims service. From the customer's perspective, Auto-Owners claims service is often the most important service that can be provided to them. This service allows customers to file claims on damaged properties after a catastrophic weather event. Auto-Owners' goal is to provide efficient, timely service to their customers when such events occur. At times, Auto-Owners sends a Mobile Claims Unit to a location after a catastrophic weather event to manage a cluster of claims in the region.

Auto-Owners is interested in stream-lined automation to help manage the large amount of claims these catastrophic events produce. There is a need for spatial mapping tools that can be used to plot potential as well as actual claims associated with a catastrophic event to help with the following:

1. claim assignment to internal and external claim handlers
2. providing the user a visual understanding of the loss locations
3. categorizing areas as high or low probability for severe damage

Auto-Owners is interested in providing these Mobile Claims Units an application that allows them the ability to gather large weather related data from external sources and combine it with existing policy location data to be displayed in the map view. To capture all the policyholder's properties that could produce potential claims as a result of the catastrophic weather event, the claims adjuster will need the ability to draw closed polygon on the map that would represent the selected area. Once the shape is drawn, the adjuster should be able to identify locations within that area (i.e. claims, policies) based on their address. The claims adjusters should be able to zoom in and out on the map, move the map to center a desired location on the screen and have the ability to select filters for the claims/policies being displayed on the map. These filters would be displayed on the map and the visible plotted items should refresh as the filters change. The application should have the ability to show weather information on the map as well as an area on the map and have the application provide the optimal positioning of the mobile unit based on the weather and policy data. The application will need to leverage location services to provide suggested routing for the mobile unit to the desired location(s).

Auto-Owners associates have access to a web system that allows for administrative reporting features. The administrators have access to reports on specific claims and prospective claim details as well as overall statistics of usage by the claim handles (reports and graphs etc.).

2. Functional Specification

The primary focus of this project for Auto-Owners is establish an interactive webpage and mobile application to determine the locations of policyholders effected by a catastrophic weather event. This application is utilized by Catastrophic Claims team members who will be on-site after the weather event to help policyholders file their claims. Through the use of a web and mobile application, the claims team will be able to locate policies, determine a centralized point between all the potential claims in an area effected by a catastrophic event to efficiently park their Mobile Claims Unit Bus, and create claims reports using the mobile app even while offline.

2.1. Web Application

The web application is the start of the Catastrophic Claims process. The first function of the web application is displaying all the policyholder's properties in an area based on postal address on a Google map in the middle of the page. On this map, there will be a weather overlay that can be toggled using the data gathered from Aeris Weather to visualize where a catastrophic event is happening and what policies could potentially be in a high damage area. Each policyholder property will have a color coding to determine the claim status on the property. These colors are used to easily identify if a claim has been started, been assigned to an adjuster or if the claim has been closed. Once gathered, the claims unit can then use this information to determine a centralized location to park the Mobile Claims Unit Bus to maximize efficiency in assisting policyholders with their claims.

The second function of the web application is to narrow down policies based on a variety of filters. A unique type of filter is to draw a closed polygon on the map, capturing all the policyholder property in that area. From this captured policy information, the claims unit can then pull down statistics from these policies as well as go in and see what individual polices are in this set of data. This includes all policy information as well as claims that have already been filed if applicable. Polices are also searchable incase a claims unit user wants to look up a specific policy. Statistical data includes claim count over a period of time, claim count by agency, and a breakdown of claims by certain categories.

The third function of the web application will be to save all the polices captured together into a specific catastrophe event. This way a claims unit user can look up policies related to a certain catastrophe and can add or delete polices in that particular dataset as needed. This catastrophe set is what the mobile application will be pulling down so that the mobile device already has all the preset policy information before going out into the field to collect claims as they are being filed.

In the web application, each claims adjuster will be assigned an authority level. All levels can read and search policy data and see statistics based on a set of policies captured in an area. A level 2 authority user can edit some of the claims filed where as a level 3 authority user can edit any claim filed in the system.

2.2. Mobile Application

The mobile application is used once the claims unit user has collected all the policies in an area that they desire and arrive on the scene of a catastrophic event. The claims unit user would then respond to policyholders as they file claims and the user would already have the policyholder's information and a preset claims questionnaire to fill out on the application. By using a mobile application form, this reduces the amount of paper that the claims unit would have to carry around as it has the ability to sync up with the main dataset when they are done with an area.

There are two different functionality parts with the mobile application. If the mobile application has access to internet services, then the claims unit user will be able to login via their Auto-Owners account credentials to pull down new disaster data sets to their device or sync up data they have of an existing dataset. If the mobile device does not have access to internet services, then the claims unit user logs into the application via a password for a specific disaster to go and see policy information and file claims using a preprinted loss notice on the device. The Mobile Claims Unit will be able to quickly and efficiently search for the insured person's policy information based on common data found on the form and begin the claims process.

3. Design Specification

3.1. Overview – Web Application

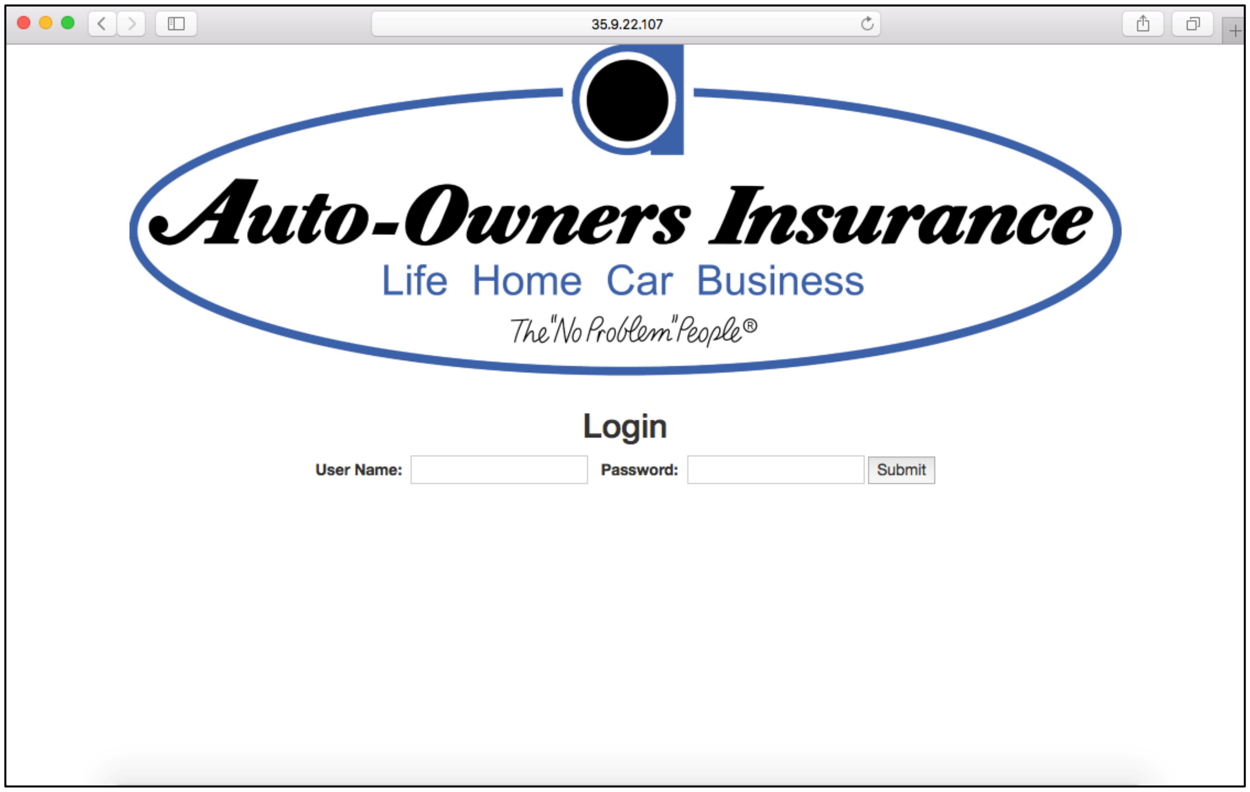
The web application can be accessed on the internet by logging into the site using Auto-Owners credentials. The web application is used to help organize the Mobile Claims Unit before they are deployed down to a disaster area by capturing and organizing potential claims policies in that area. The web application will allow the Mobile Claims Unit to create a disaster dataset and filter through to find specific policies in that dataset to maximize their efficiency. The web application will also allow the claims unit to determine the centralized location of the potential claims policies in order to park the Mobile Claims Unit Bus effectively.

3.2. User Interface – Web Application

The user interface of the web application consists of four web pages to display, filter, capture and generate statistics of desired policy sets. The home page lists all the catastrophic datasets and allows users to make new ones. The map page is where the user will be able to visualize weather data, capture a set of policies in an area on the map by drawing a closed polygon on the map, and filter through those policies. The policy page is where the user will be able to display a catastrophic event policy set and see individual policyholder information and claims information. The statistics page is where the user will be able to view statistics from a given policy set.

3.2.1. Login Page

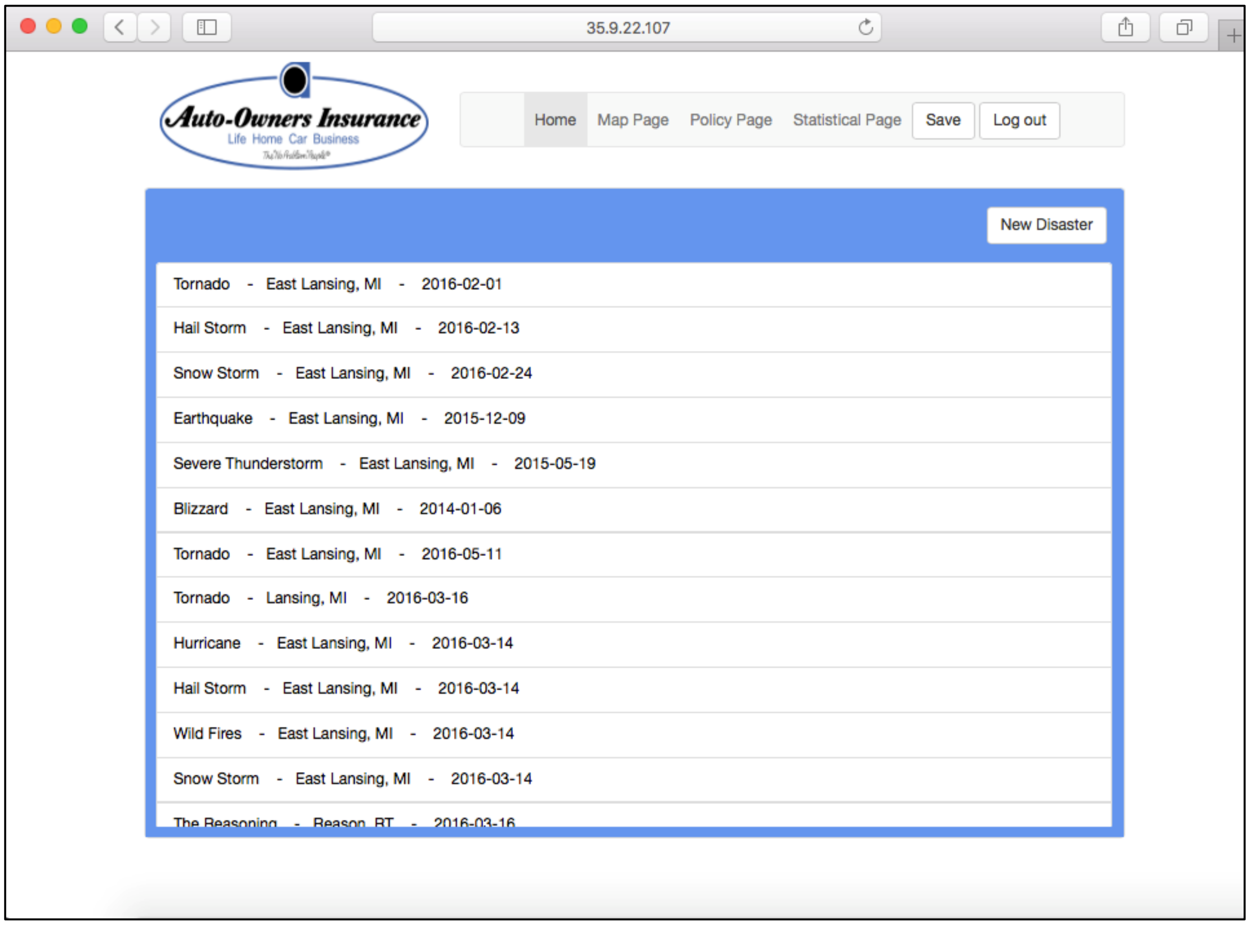
Figure 1 shows login page for the web application. The claims unit user will log in via their Auto-Owners login information to gain access to the web application.



[Figure 1 – Login page]

3.2.2. Home Page

Figure 2 displays the landing page for the web application. A claims unit user will be able to select an existing disaster event and display that event's information or it allows the user to create a new disaster event and be redirected to the map page.

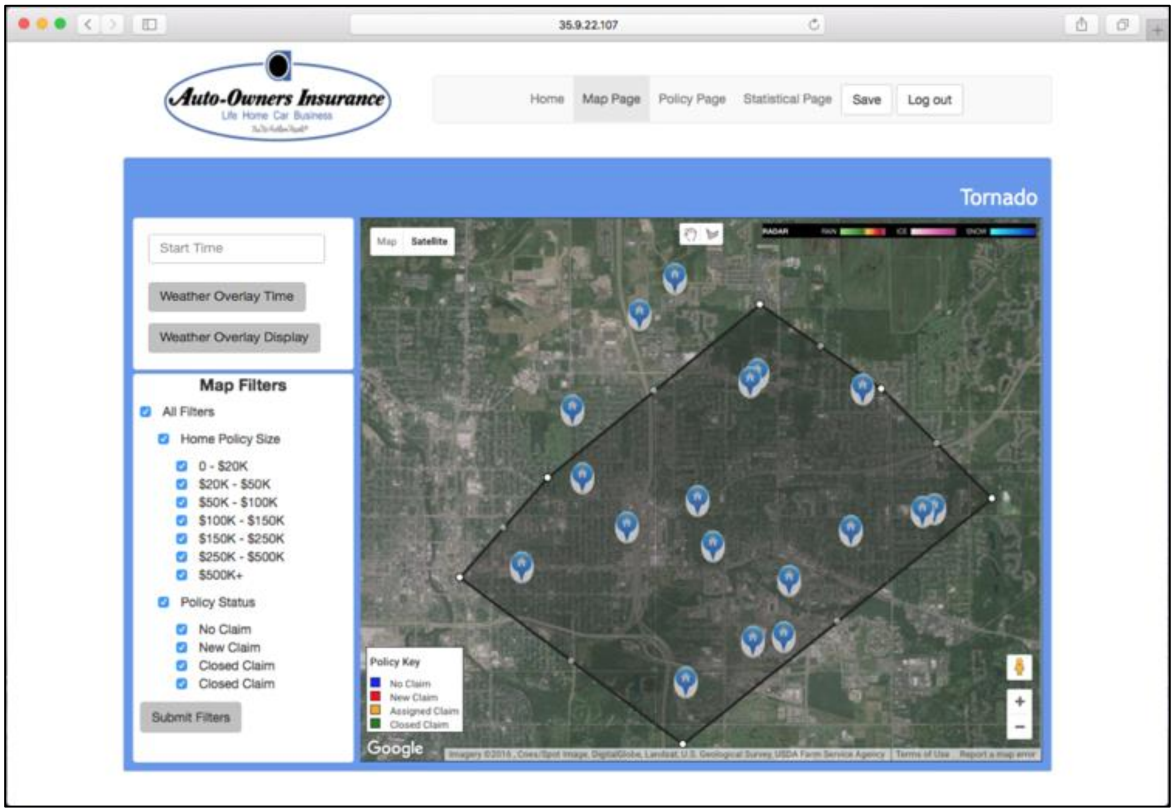


Tornado - East Lansing, MI - 2016-02-01
Hail Storm - East Lansing, MI - 2016-02-13
Snow Storm - East Lansing, MI - 2016-02-24
Earthquake - East Lansing, MI - 2015-12-09
Severe Thunderstorm - East Lansing, MI - 2015-05-19
Blizzard - East Lansing, MI - 2014-01-06
Tornado - East Lansing, MI - 2016-05-11
Tornado - Lansing, MI - 2016-03-16
Hurricane - East Lansing, MI - 2016-03-14
Hail Storm - East Lansing, MI - 2016-03-14
Wild Fires - East Lansing, MI - 2016-03-14
Snow Storm - East Lansing, MI - 2016-03-14
The Reasoning - Reason, RT - 2016-03-16

[Figure 2 – Home Page]

3.2.3. Map Page

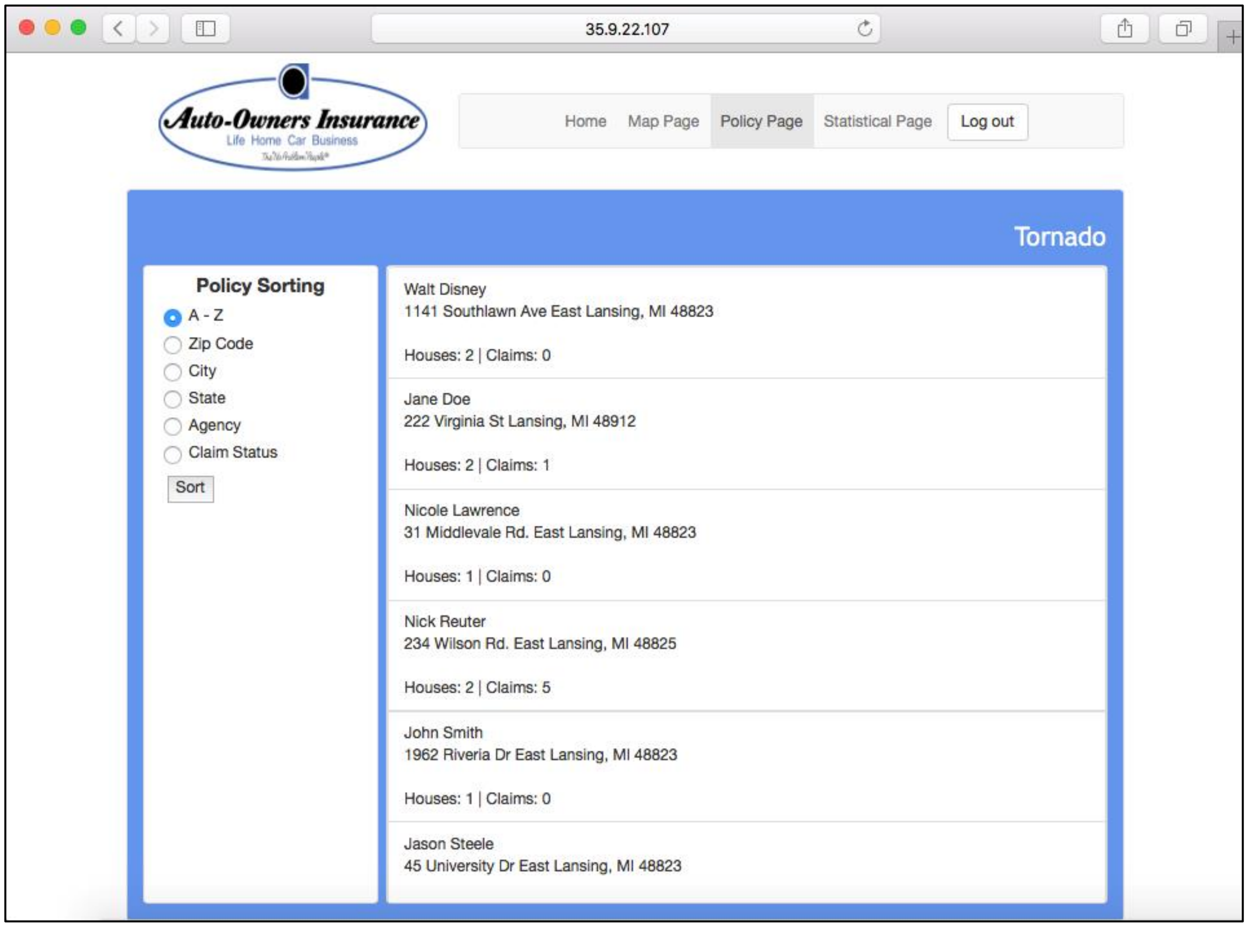
Figure 3 shows where the user can see all the policyholder's properties on a map indicated by different color markers. On this map, the markers can be filtered by checkboxes in the left navigation pane where the user can select multiple filters in each category. The user will also be able to zoom in and out, move, rotate and center the map as well as draw a closed polygon on the map by click on the corners of the polygon they wish to create. The user can then hit the Weather Overlay Display button in the left corner to toggle the weather overlay on and off. Hitting the Weather Overlay Time button after selecting a date brings up the weather image from that timeframe so claims adjusters can see past weather occurrences. A weather map overlay will be placed on top of the map to visual the weather and locate polices in high damage areas. Once the user is satisfied all the desired polices have been captured, they can hit the save button to then create a new catastrophe dataset or sync up to an existing dataset.



[Figure 3 – Map page]

3.2.4. Policy Page

Figure 4 represents the policy page where a claims unit user can see all the policyholders associated with a disaster dataset. They will be shown in a scrollable panel and display high level information such as a policyholder's name address as well as the number of houses they have insured. This list can be filtered using the radio buttons on the left navigation pane. A user can click on a policyholder and it will bring up that particular policyholder's information and any claims associated with their properties.



Tornado

Policy Sorting

- [✓ A - Z]
- Zip Code
- City
- State
- Agency
- Claim Status

Sort

Walt Disney
1141 Southlawn Ave East Lansing, MI 48823

Houses: 2 | Claims: 0

Jane Doe
222 Virginia St Lansing, MI 48912

Houses: 2 | Claims: 1

Nicole Lawrence
31 Middlevale Rd. East Lansing, MI 48823

Houses: 1 | Claims: 0

Nick Reuter
234 Wilson Rd. East Lansing, MI 48825

Houses: 2 | Claims: 5

John Smith
1962 Riveria Dr East Lansing, MI 48823

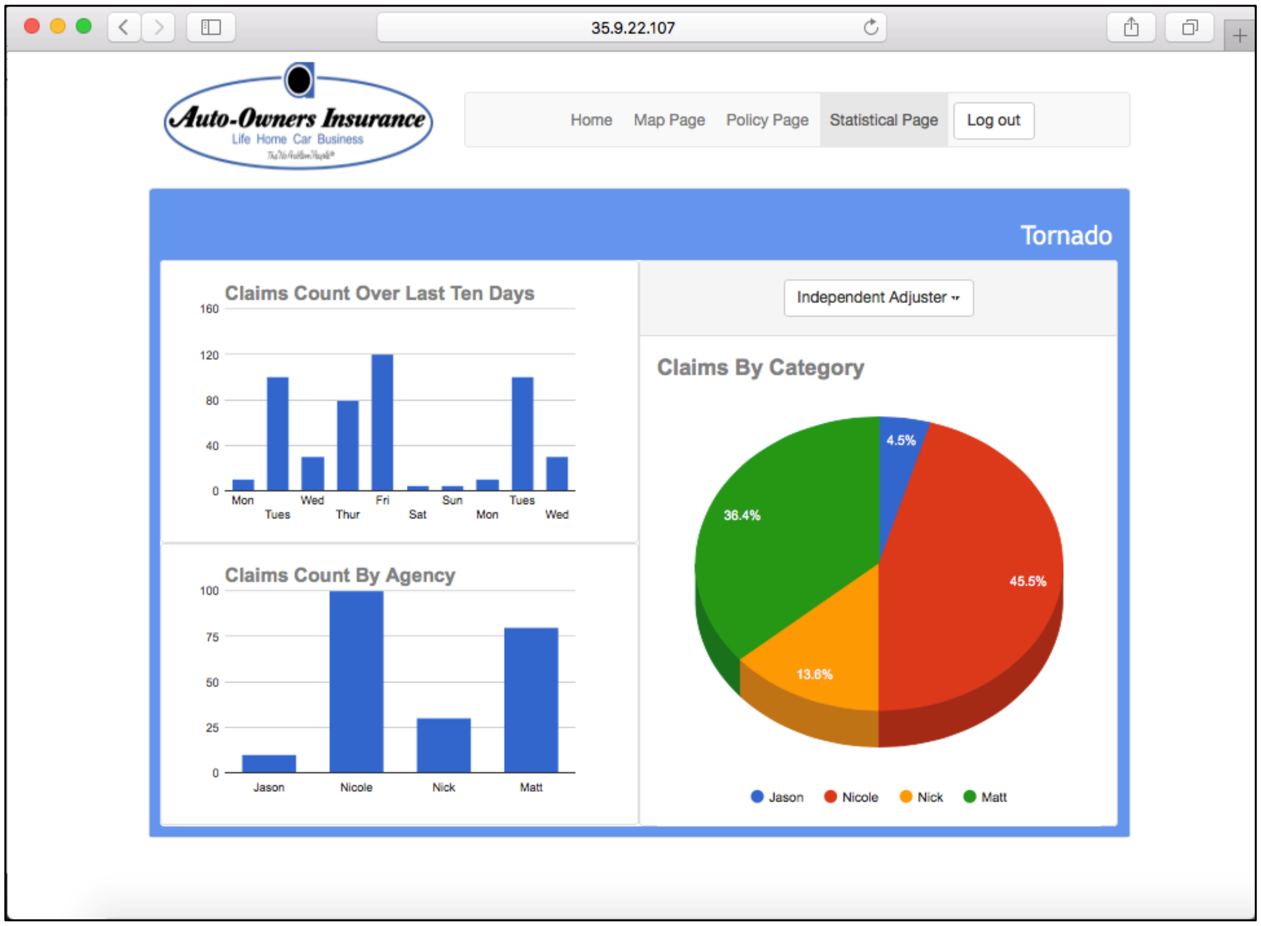
Houses: 1 | Claims: 0

Jason Steele
45 University Dr East Lansing, MI 48823

[Figure 4 – Policy Page]

3.2.5. Statistics Page

Figure 5 displays the statistics page where a claims unit user is shown statistics on a disaster dataset. The bar charts display the claims count over time and by agency while the bar chart information will change based on the drop down selection. For example, claims unit user can see what percentage of claims each adjuster has filed during this disaster.



[Figure 5 – Statistics Page]

3.3. Overview – Mobile Application

The iOS application will be downloadable by Auto-Owners Insurance employees on the Mobile Claims Unit disaster team. The application is built using the Swift Programming language with the Cocoa Touch framework and is targeted at Apple iPad devices. The application is designed to replace a paper form claims agents currently use and will expedite the claims process during a disaster when Auto-Owners is receiving a high volume of claims. The application is designed to enable the claims agent to process claims without an internet connection and upload completed claims when they connect to the internet due to the high probability of network outages in the disaster area.

3.4. User Interface – Mobile Application

When the user starts the iOS application they are presented with a screen where they can select a disaster, publish the disaster to the web application, or get a new disaster. If they decide to download a disaster they must login with their Auto-Owners username and password. If they successfully login, they are presented with a screen listing disasters stored on the server.

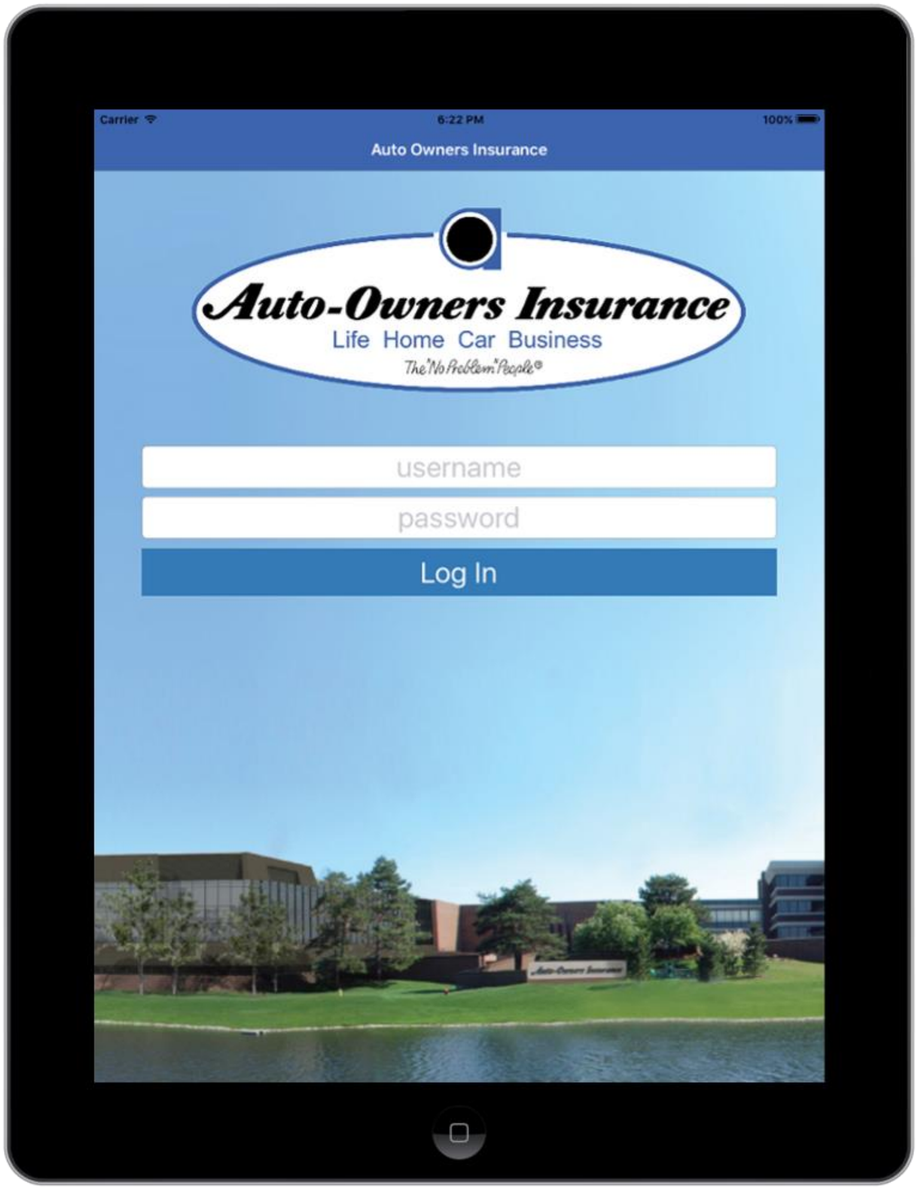
The user will then select a disaster to download all policies associated with it. After the policies for that disaster have been downloaded the user is returned to the start screen and can select the newly downloaded disaster. When the user clicks the publish to server button all claims stored on the mobile device will be published to the web application. When the user selects a disaster on the start screen they must login with a password that is stored on the device.

If they successfully login they are taken to a screen displaying a searchable list of policyholders' names and addresses that are associated with the disaster. When the user finds a policyholder they are looking for, they can click on the person's name and view detailed policy information and start a claim.

After the user completes the claim they click save and the claim is saved on the local device. Once the user connects to the internet they will log into the server through the iOS application and publish claims stored on the device to the database.

3.4.1. Login Page

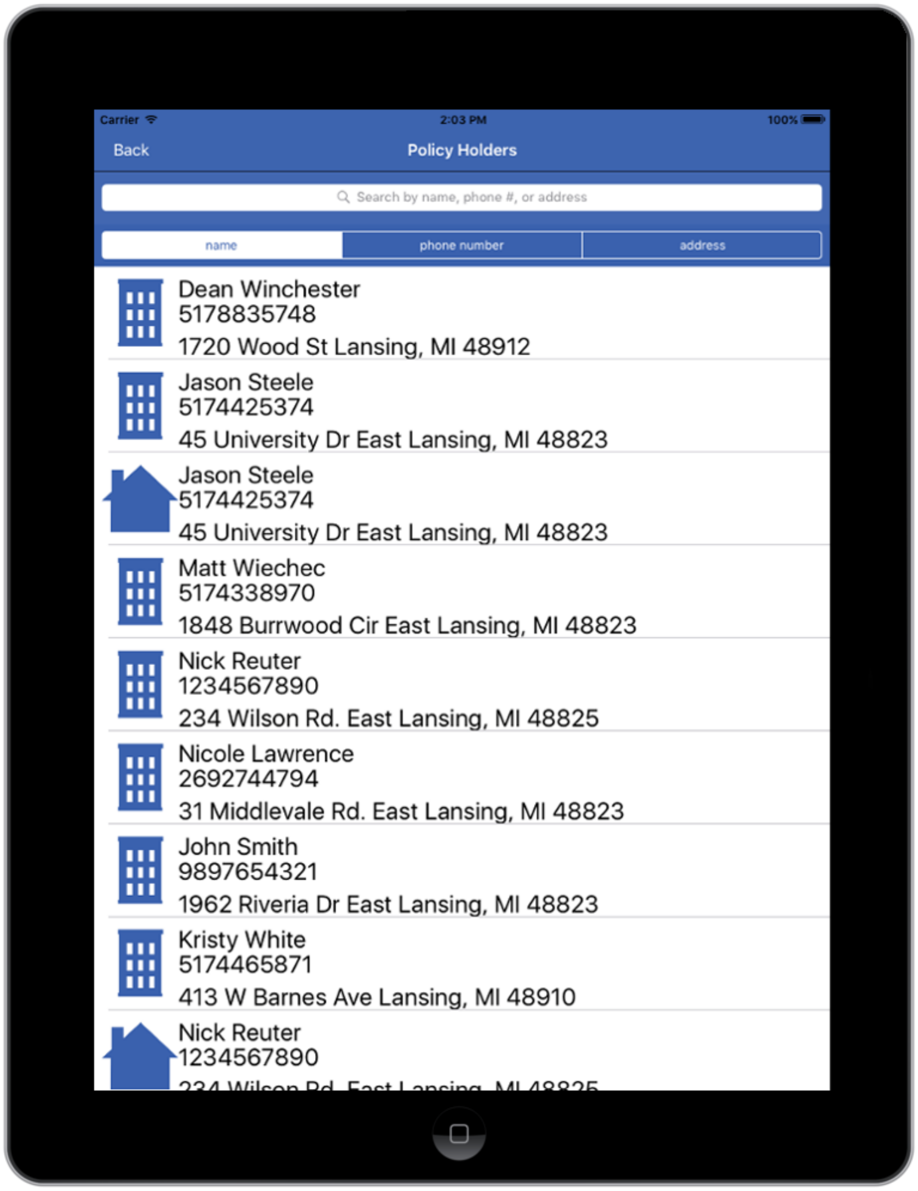
Figure 6 displays the login screen which prevents unauthorized users from using the mobile application and viewing policies currently stored on the device. When there is no internet connection the user supplied credentials will be verified against an internal database.



[Figure 6 – Login screen]

3.4.2. Policies Page

Figure 7 lists the policyholder name and address for all policies associated with a disaster. The search bar at the top of the screen can search policies based on a policyholder's name, address, or phone number. When a user clicks on the policyholder they are taken to a new screen to view more detailed policy information.



Dean Winchester
5178835748
1720 Wood St Lansing, MI 48912

Jason Steele
5174425374
45 University Dr East Lansing, MI 48823

Jason Steele
5174425374
45 University Dr East Lansing, MI 48823

Matt Wiechec
5174338970
1848 Burrwood Cir East Lansing, MI 48823

Nick Reuter
1234567890
234 Wilson Rd. East Lansing, MI 48825

Nicole Lawrence
2692744794
31 Middlevale Rd. East Lansing, MI 48823

John Smith
9897654321
1962 Riveria Dr East Lansing, MI 48823

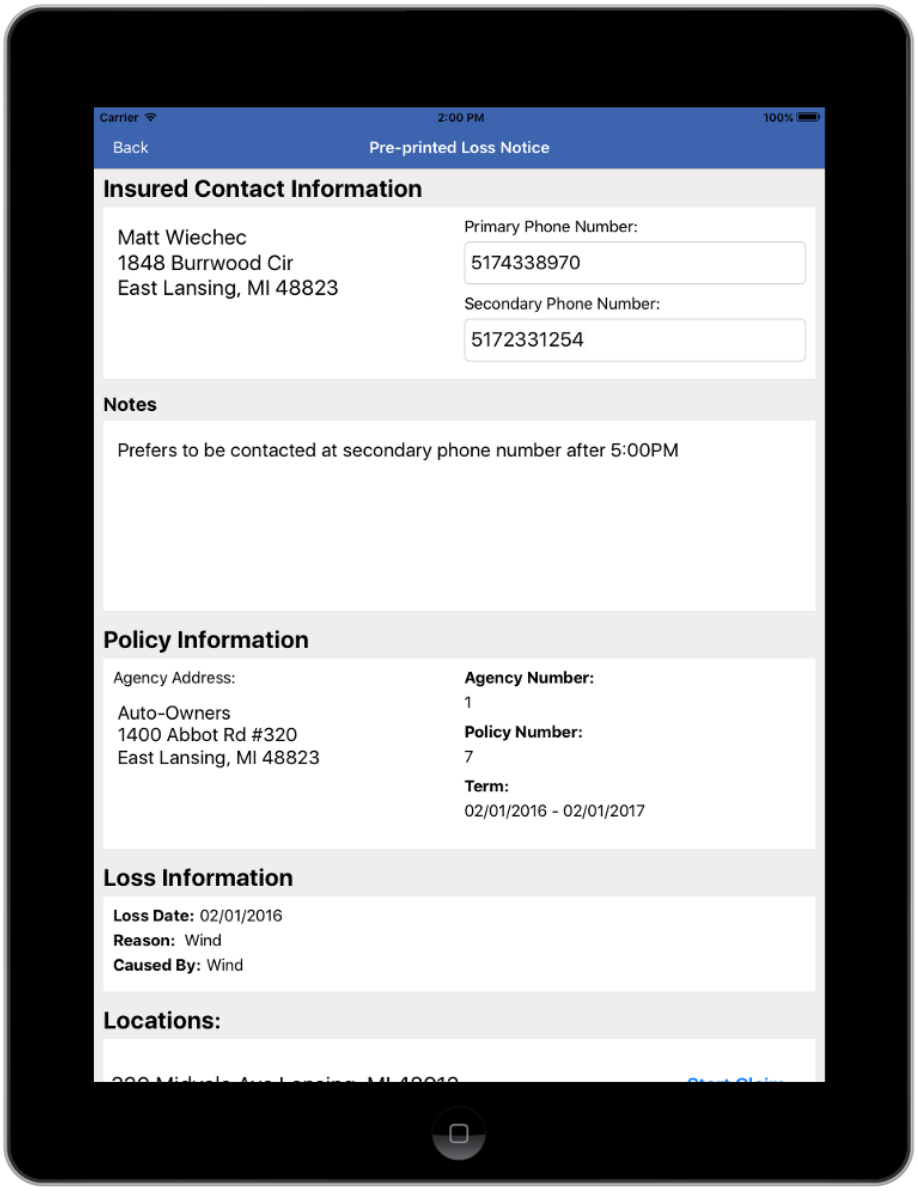
Kristy White
5174465871
413 W Barnes Ave Lansing, MI 48910

Nick Reuter
1234567890

[Figure 7 – Policy list screen]

3.4.3. Policyholder Information Page

Figure 8 lists all available information regarding a policy. The user can gather contact information for the policyholder and view loss information associated with the disaster. When the user clicks the Start Claim button for a location they are taken to a new screen to begin the claims process.



Insured Contact Information

Matt Wiechec
1848 Burrwood Cir
East Lansing, MI 48823

Primary Phone Number: 5174338970

Secondary Phone Number: 5172331254

Notes

Prefers to be contacted at secondary phone number after 5:00PM

Policy Information

Agency Address:
Auto-Owners
1400 Abbot Rd #320
East Lansing, MI 48823

Agency Number: 1
Policy Number: 7
Term: 02/01/2016 - 02/01/2017

Loss Information

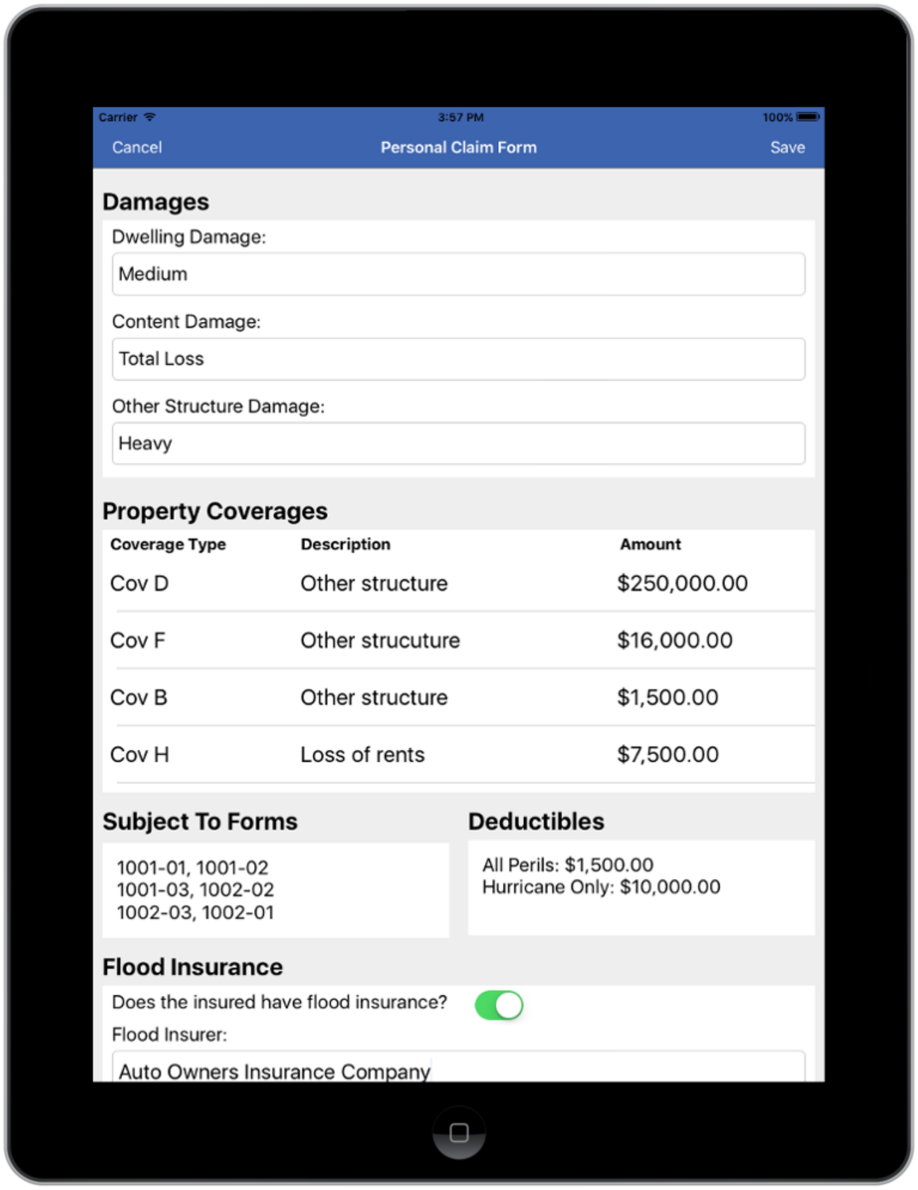
Loss Date: 02/01/2016
Reason: Wind
Caused By: Wind

Locations:

[Figure 8 – Policyholder screen]

3.4.4. Personal Claim Page

Figure 9 allows the claims unit user to process a personal claim associated with a policy. The user determines the amount of damage to the insured structure and if the insured has flood insurance. When the user clicks save, the application stores the added data in the internal database until it can be published to the web application.



Damages

Dwelling Damage: Medium

Content Damage: Total Loss

Other Structure Damage: Heavy

Property Coverages

Coverage Type	Description	Amount
Cov D	Other structure	$250,000.00
Cov F	Other strucuture	$16,000.00
Cov B	Other structure	$1,500.00
Cov H	Loss of rents	$7,500.00

Subject To Forms

1001-01, 1001-02
1001-03, 1002-02
1002-03, 1002-01

Deductibles

All Perils: $1,500.00
Hurricane Only: $10,000.00

Flood Insurance

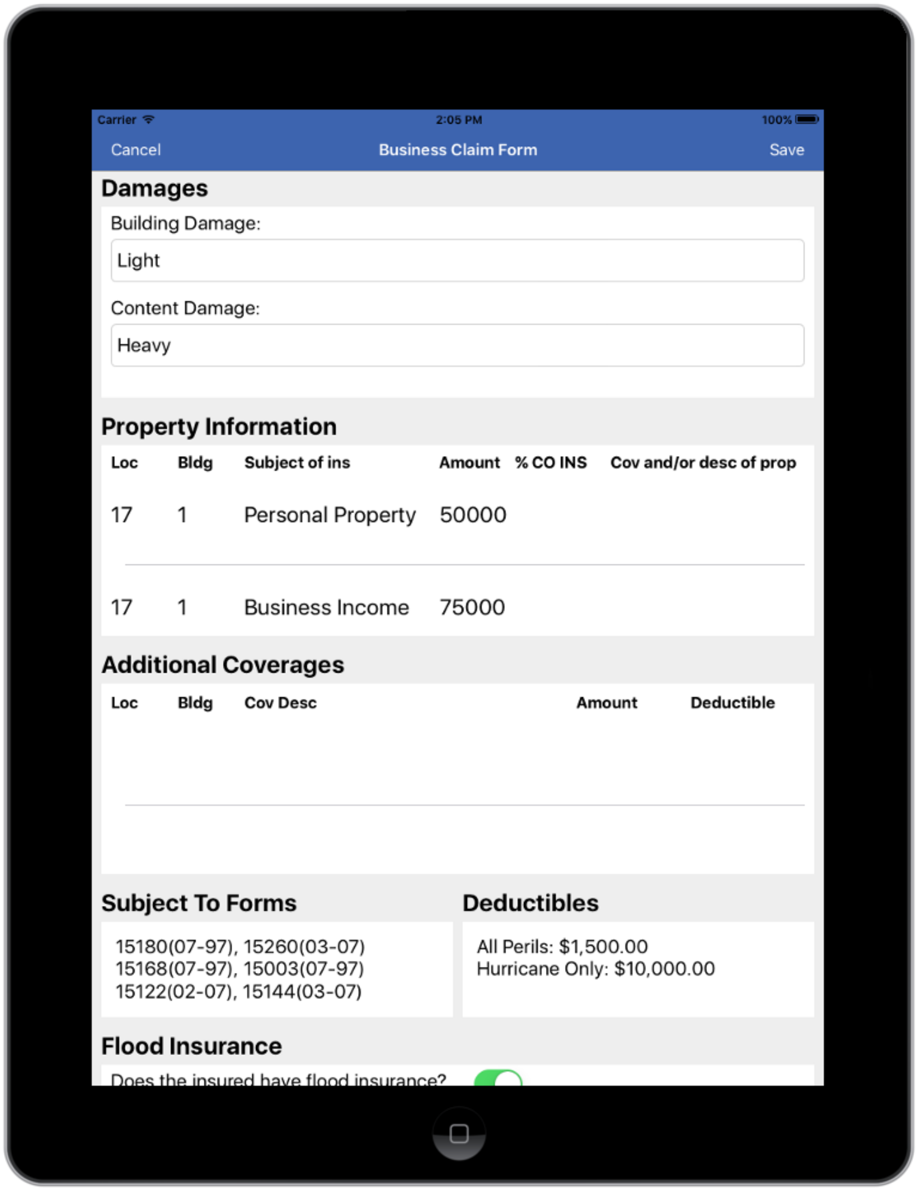
Does the insured have flood insurance? ✓

Flood Insurer: Auto Owners Insurance Company

[Figure 9 – Personal Claim Form]

3.4.5. Business Claim Page

Figure 10 allows the claims unit user to process a business claim associated with a policy. The user determines the amount of damage to the insured structure and additional buildings and if the insured has flood insurance. When the user clicks save, the application stores the added data in the internal database until it can be published to the web application.



Damages

Building Damage: Light

Content Damage: Heavy

Property Information

Loc	Bldg	Subject of ins	Amount	% CO INS	Cov and/or desc of prop
17	1	Personal Property	50000		
17	1	Business Income	75000		

Additional Coverages

Loc	Bldg	Cov Desc	Amount	Deductible

Subject To Forms

15180(07-97), 15260(03-07)
15168(07-97), 15003(07-97)
15122(02-07), 15144(03-07)

Deductibles

All Perils: $1,500.00
Hurricane Only: $10,000.00

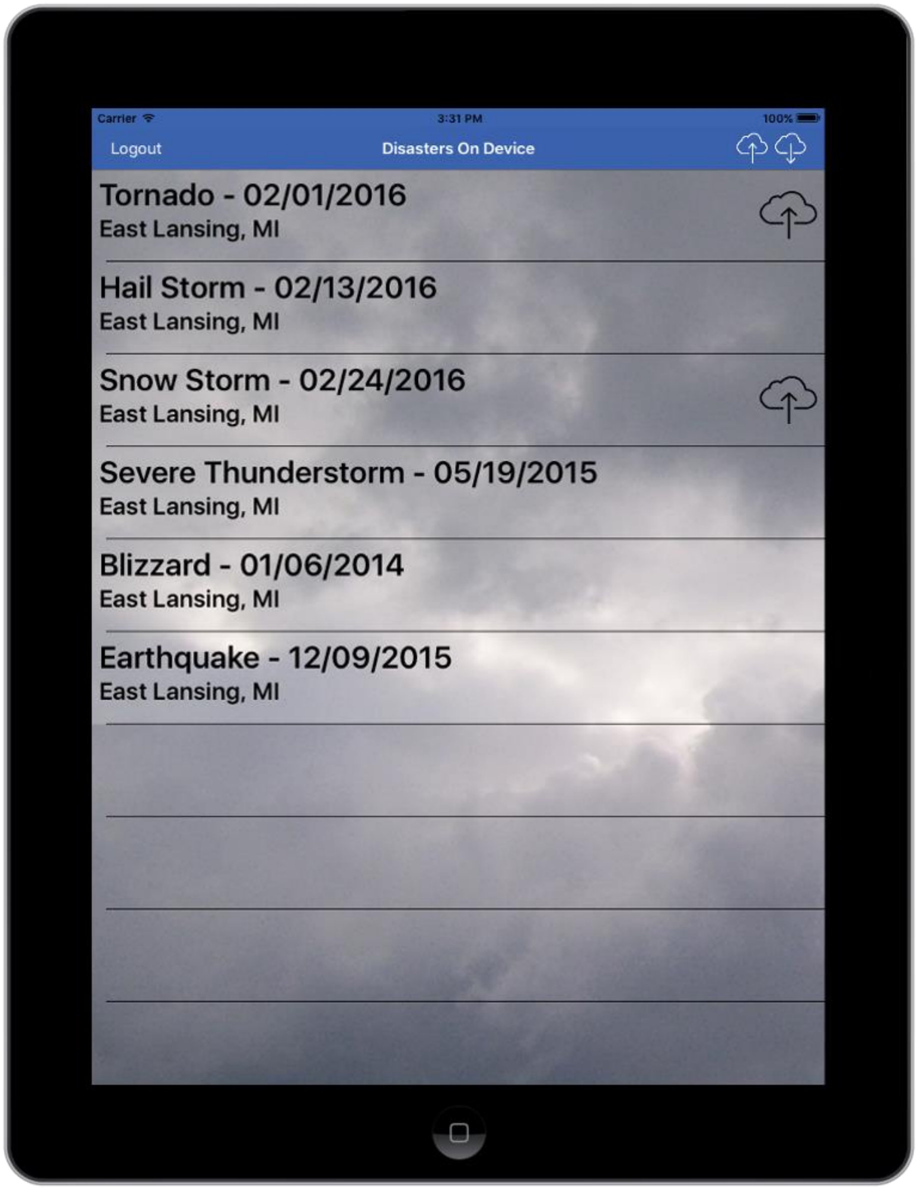
Flood Insurance

Does the insured have flood insurance?

[Figure 10 – Business Claim Form]

3.4.6. Disaster List on Client

Figure 11 displays the list of disasters on stored on the device. If the user clicks on one of the disasters, they are taken to a screen to view all policyholders affected by that disaster. When the upload button is clicked the user will upload all the claims they have made or modified to the web application. When the user clicks the download button they will be taken to a screen to download a disaster from the web application



Tornado - 02/01/2016
East Lansing, MI

Hail Storm - 02/13/2016
East Lansing, MI

Snow Storm - 02/24/2016
East Lansing, MI

Severe Thunderstorm - 05/19/2015
East Lansing, MI

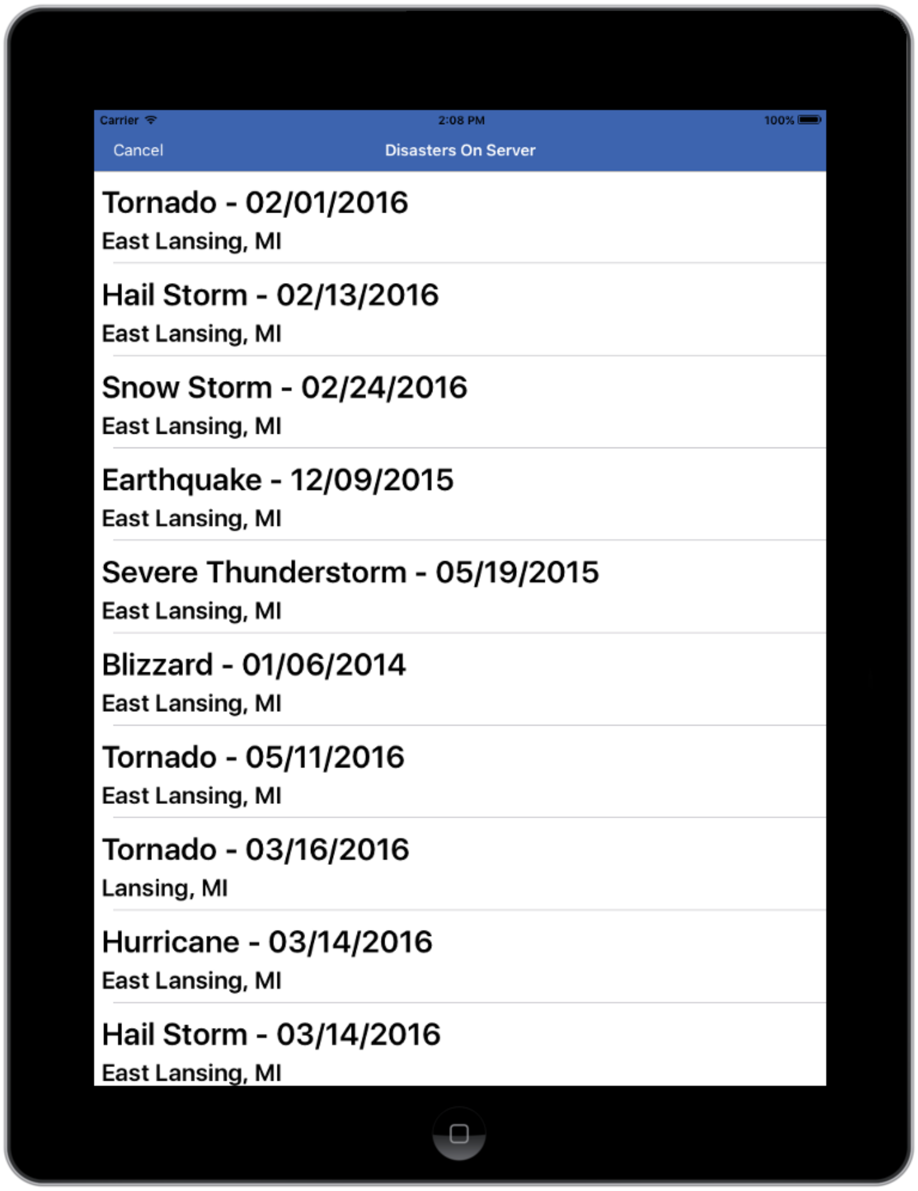
Blizzard - 01/06/2014
East Lansing, MI

Earthquake - 12/09/2015
East Lansing, MI

[Figure 11 – Local Disaster List screen]

3.4.7. Disaster List on Server

Figure 12 allows a user to download a disaster stored on the main database. Once the user selects a disaster they are returned to the disasters on device screen and have the ability to select the newly downloaded disaster.



Tornado - 02/01/2016
East Lansing, MI

Hail Storm - 02/13/2016
East Lansing, MI

Snow Storm - 02/24/2016
East Lansing, MI

Earthquake - 12/09/2015
East Lansing, MI

Severe Thunderstorm - 05/19/2015
East Lansing, MI

Blizzard - 01/06/2014
East Lansing, MI

Tornado - 05/11/2016
East Lansing, MI

Tornado - 03/16/2016
Lansing, MI

Hurricane - 03/14/2016
East Lansing, MI

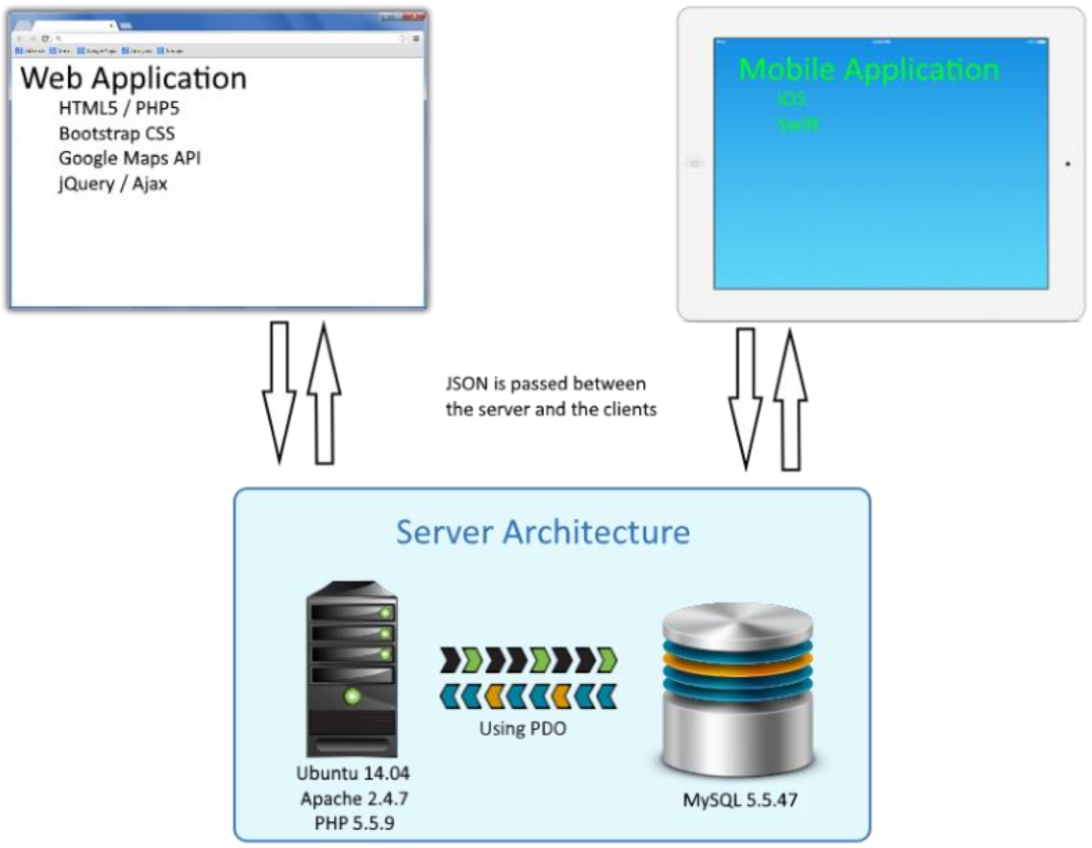
Hail Storm - 03/14/2016
East Lansing, MI

[Figure 12 – Complete Disaster List screen]

4. Technical Specification

4.1. System Architecture

The Catastrophic Claims Unit Mobilization Application consists of two user-facing applications and one central server. Comprised of a web application and an iOS mobile application, both communicate to the same central server to share information and generate the correct set of policies and claims. The client applications, the server, and the database form a cohesive unit allowing for both applications to be extended with the common point of interest remaining unchanged.



[Figure 13 - diagram of system architecture]

As seen above in Figure 13, the Database instance is hosted locally within the server and is only accessible by the applications through the Apache instance and utilizing PDO. The separation of the client-facing applications allows for additional features to be added to each without disrupting the overall workings of the system.

4.2. The Database

The database is an instance of MySQL 5.5.47 and is hosted locally on the web server.

4.3. The Server

The server itself is running Ubuntu 14.04 and utilizes Apache 2.4.7 as the web hosting software. The pages themselves are constructed with PHP 5.5.9. PHP is being used instead of Java because the recommended system for deploying Java web applications, GlassFish, has a bug that prevents it from establishing any JDBC's. As a result, we had to swap it out for PHP. The server is able to communicate with its instance of MySQL using PDO objects.

4.4. Web Application

The web application is constructed using basic HTML5, JavaScript, and Twitter's Bootstrap CSS library. To communicate with the server, the web application will make AJAX calls using jQuery and will send and receive requests through JSON. These calls are done using the HTTP methods GET, POST, PUT, and DELETE. The web application also makes use of Google's APIs for Maps, Geo-lookup, Aeris Weather Overlays, and freeform drawing on maps. These APIs allow the policy data to be properly displayed and selected on the map. Each webpage is only displayed if there is a record of a user having previously successfully completed signing in with the server.

When policies are selected in a closed polygon drawing on the map, those locations are added into an array of geolocations. The web application will call the server and query for basic policy information such as name and address and store that into a temporary dataset table on the server. When the user goes to the policy page and looks at a specific policy, that geolocation calls the server for the full download on that policy including any claims that are associated and displays it on the screen. The statistics page will take the array of geolocations and will pull down any statistical information on that geolocation set and display it on the screen. If a user syncs the selected geolocations with an existing disaster dataset, then the web page will call the server for all the information of those geolocations and add them to the existing table for the dataset. If the user syncs a new disaster dataset, the geolocations in the array pull all the policy information and create a new table specifically for that disaster dataset with the name of the major disaster. A new claims table is created and linked to the new disaster table.

4.5. Mobile Application

The mobile application is being written in Swift for iOS 9 on apple's iPad devices. Swift has built in libraries for talking to external sources such as servers, but since the server responds with JSON instead of xml so the app will be utilizing the Gloss library which makes the parsing of JSON much easier. The mobile app will also have a Core Data instance running on it to keep a copy of the policies from the main Disaster table on the server as well as an empty claims table to store any claims created using the app.

4.6. Development Environment

To create the web application, PHPStorm on an Apple desktop running El Capitan was used and the deployment option was linked to the directory on the server where Apache serves its files from. OpenSSH was also installed on Ubuntu to allows the files to be transferred over SFTP and so that the server could be remotely accessed through SSH. The MySQL instance is accessible for testing and observation through the phpMyAdmin plugin. The mobile application is getting developed on an Apple laptop running El Capitan using Xcode.

5. Risks

Map overlays from NOAA onto Google Maps

- Finding and pulling past and present NOAA maps and merging them onto Google Maps
- Mitigation: Researching map overlays and synchronization
- Status: Displaying Aeris Weather data and are able to pull down past date weather information

iOS Swift programming fundamentals

- No practical knowledge base on swift development
- Mitigation: Research swift and iOS development
- Status: Able to use Swift to make a custom mobile application

Synching database to mobile app and ability to run offline

- Creating a mobile app that can run offline without cell service
- Mitigation: Researching self-contained databases
- Status: Running with offline capabilities and syncing to database when connected to internet

6. Timeline

Week 1 (1/11 – 1/17)

1/15 – In person meeting with Auto-Owners team
Began Project Plan and started a ruff layout
Installed VMware, Windows, and Chrome

Week 2 (1/18 – 1/24)

Created Git account and Trello for project management.
Installed NetBeans, Sublime, Xcode and PHPStorm
Set up server for Windows, installed Apache, MySQL and GlassFish

Week 3 (1/25 – 1/31)

1/25 – Status Report Presentation
Presented 1st Mock-ups to Auto-Owners
Completed Project Plan
Working on 2nd Mock-ups
Reset server for Linux, installing Apache, MySQL and PHP

Week 4 (2/01 – 2/07)

2/01 – Project Plan Presentations
Mobile - iOS login functional, iOS screen layouts nearly complete, iOS can query Core Data database
Web - Finalize all filters and page layouts for web application

Week 5 (2/08 – 2/14)

Mobile - iOS screen layouts complete, application can communicate with web server and download and view policies
Web – Login functionality and database design. Basic page layout complete for all pages

Week 6 (2/15 – 2/21)

Mobile - iOS application can process claim and store in database
Web – Test connections between web and database using JSON. Capture geolocations using closed polygons drawn on map

Week 7 (2/22 – 2/28)

2/22 – Alpha Presentations
Mobile - User can push claims to web application/error messages display when no internet connection
Web –Weather overlays toggled and updated overall design of web application

Week 8 (2/29 – 3/06)

Web – Be able to create and manage disaster datasets as well as sync new policies to an existing dataset. Create and display policyholder information via modal.

Week 9 (3/07 – 3/13)

Spring Break

Week 10 (3/14 – 3/20)

Mobile - User can delete disaster from iPad only if all claims have been uploaded to server
Web – Statistical Google APIs set up waiting on claims from mobile application

Week 11 (3/21 – 3/27)

Adding test data

Week 12 (3/28 – 4/03)
- Start Video
- Mobile – Testing offline and online functionality
- Web – Testing functionality of features on web application. Generate statistics

Week 13 (4/04 – 4/10)
- 4/04 – Beta Presentations

Week 14 (4/11 – 4/17)
- Initial aim at completion of mobile app and webpage
- Debugging and polishing

Week 15 (4/18 – 4/24)
- Final testing and last minute modification.
- Wrapping up in preparation for Design Day

Week 16 (4/25 – 5/01)
- 4/25 – Project Videos
- 4/27 – All Deliverables
- 4/28 – Design Day Setup
- 4/29 – Design Day