Lab 1 - Pest Patrol Product Description

Bryan Burton

Old Dominion University

CS411W

Professor J. Brunelle

September 18, 2022

Version 2

Table of Contents

List of Figures

1 Introduction

Pests are everywhere, from nuisance mosquitos, voracious termites, and deadly bobcats, pest encounters in communities are a common and sometimes daily occurrence. However, despite pest encounters being so common in the community, rarely does the community become aware of these pests and work together to mitigate them, as most pest encounters are isolated and there are few ways to report them. This is troublesome, as it has been shown from the prevention of the spread of Spongy Moth in the United States (Coyle, 2021), 90% reduction of invasive sea lamprey in the Great Lakes (Bugg et al., 2020), and eradication of invasive Japanese beetles in parts of California ("Japanese beetle repeatedly eradicated from California," n.d.), that awareness, communication, and coordination are crucial for the successful mitigation of pests. None of which are possible without the means to report pests to the community and collaborate with community members.

A means for reporting pests and collaborating with community members is necessary now for mitigating pets, and may become more necessary as the occurrence of pest encounters is likely to increase in the near future. Due to warmer winters, the prevalence of pests will increase ("How warmer winters affect pests - Varment guard wildlife services," 2018); for example, without a cold winter to encourage hibernation, pests like mosquitoes and termites will continue to reproduce and spread throughout the year, potentially leading to more encounters with these pests. As well, as more habitats are encroached on for construction, more wildlife is displaced, which may lead to nuisance pests moving into populated areas (Abell, 2021); notably, adaptable and opportunistic animals such as raccoons and bears may remain in the same area despite construction, leading to potential encounters with them.

Another key problem is that most pest encounters are preventable, and yet they still occur due to the lack of real-time community awareness. For example, if a community member has a rat infestation in his house, but does not make the community aware of a potential rat problem, then these rats will find it very easy to migrate to his neighbor's house and cause another infestation. Pests can and do spread, and the lack of preparation and strategies to prevent this will just make their spread easier.

Members of the community are not the only people who are affected by the lack of pest awareness. Cities, researchers, and outdoor businesses all have an interest in knowing what pests are in their areas. Cities, for instance, need to know when an invasive species enters their area, as this pest could harm local wildlife or negatively affect local agriculture. Researchers need to know the locations of pests in order to predict their spread and create ways to mitigate it. Outdoor businesses have to know what pests are in their work area and how hazardous they may be; a mosquito swarm may not be a problem, but a bobcat in the area could be a liability.

There are many applications that can serve as potential solutions to this problem, but none of them fully address the need for collective awareness and collaboration within a community to manage pests. Nextdoor and Facebook, for example, are community oriented applications that could be used as a solution, however, their focus is not on pests and they lack the ability for tracking pests and alerting users of nearby pests. Similarly, iNaturalist, an application for crowdsourcing animal and plant identification and locations, does not focus on pests specifically, and does not alert the user of nearby pests or allow users to collaborate with each other to manage pests.

Pest Patrol addresses the lack of community pest awareness and collaboration by giving community members the means to report their pest encounters, find consolidated information about local pest sightings, and collaborate with neighbors to manage pests effectively. By using pest alerts and an interactive map containing pest incidents, members of the community can stay informed about the pests in their area and be able to track pest outbreaks. As well, with the ability to communicate with other Pest Patrol users through the application, community members can collaborate with each other to share effective pest control strategies and manage pests together.

2 Pest Patrol Description

Pest Patrol is a Web-based application that enables members of a community to report sightings of pests and see the locations of pests in their area. Pest Patrol makes communities safer and minimizes the severity of pest problems by giving its users an enhanced awareness of potential pests that may be nearby. With heat maps, predictive modeling, and the knowledge collected from prior pest sightings, members of the community can more easily recognize, prevent, and manage any pests that they may find in their community.

2.1 Key Product Features and Capabilities

Pest Patrol is a cross-platform application that will work on any machine equipped with a web browser that conforms with Word Wide Web Consortium (W3C) standards, such as Mozilla Firefox or Google Chrome. This means that Pest Patrol can be used on Windows, Macintosh, and Linux, as well as mobile operating systems such as Android.

Upon logging in to Pest Patrol, users will be greeted with a dashboard where they can access all other features in the application. Here the user can select his preferred view mode, where he can see the incident map or see discussion threads regarding recent pest incidents. The user can also combine both the incident map and the discussion view in a hybrid view mode, where he can interact with both the map and a discussion thread simultaneously.

The incident map displays reported pest incidents near the user’s location, which is defined as the user’s community location when using Pest Patrol on a desktop or the user’s physical location when using Pest Patrol on a mobile device. Which pest incidents are displayed can be customized by the user. For instance, a maximum distance can be set that will only show pests spotted within a certain radius from the user’s location. The user can also display incidents that were reported within a certain time frame, as well as display only specific types of pests, such as just insects or all pests but birds. From the incident map, the user can also report a new incident. The incident map can also display a heat map that can show the user, at a glance, the recency and prevalence of certain pests in an area.

Users can also receive pest alerts to their mobile device that will notify them about recent pest incidents that have occured in their vicinity. Like the incident map, users can customize which pests the app should prioritize when sending alerts.

From the dashboard, the user can access discussion threads that he has participated in or created. These threads can be related to a specific pest incident or they may stand alone. Threads which the user has subscribed to can be accessed here in one place.

A tab containing recent activity in the user’s community can also be accessed from the dashboard. Here the user can view new incidents and discussion threads that were created since the last time Pest Patrol was used. With machine learning and analyzing the history of prior incidents, preemptive alerts can be sent to users to warn of potential pest incidents, such as the emergence of mosquitos and other insects at different times during the year.

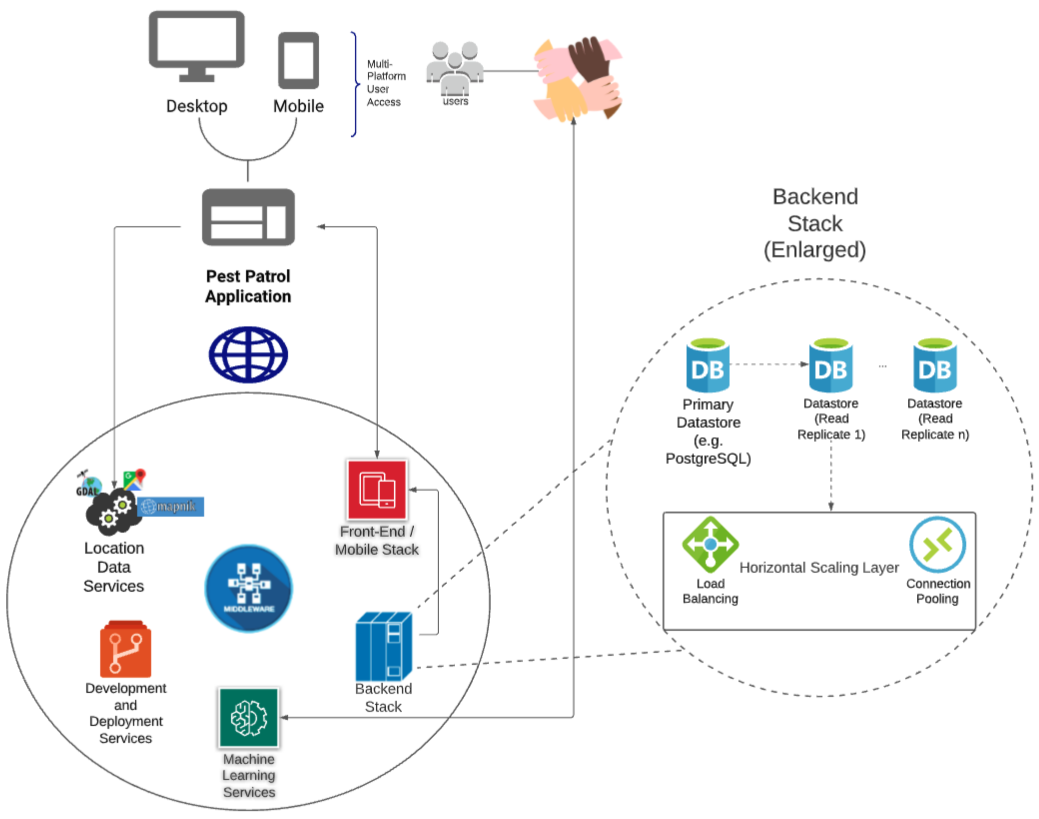
Pest Patrol also allows users to interact with each other through the community tab in the dashboard. Here the user can follow others, report users for malicious behavior, or, when learning to manage a certain pest, audit users for the validity of their suggestions based on the content of their prior discussion posts. Outside of the community tab, users can also customize their own profile with a name and password, as well as send a direct message to other users.

2.2 Major Components (Hardware / Software)

Pest Patrol is produced for a Web browser, so that it may be used on any device so long as it is connected to the internet. This application consists of a front-end and backend, with the former being the user interface which is presented to the user and the latter which exchanges data from the front-end to the database, and vice versa. A complete diagram of Pest Patrol's components can be seen in Figure 1, which displays the interaction between the front-end and the backend.

Figure 1

Pest Patrol Major Functional Component Diagram



The front-end is designed using Angular.js in order to make the Pest Patrol application reactive. This makes it possible for dynamic Web pages within the app, so that new pest incidents and thread posts can be displayed to the user in real-time. Using Angular.js also makes it simple to translate the desktop app UI to a mobile device.

Pest Patrol uses Node.js to manage interactions between the front-end and the database. Node.js asynchronous design will allow Pest Patrol to handle multiple connections at once. As well, the application can be scaled by using multiple servers and setting up a Node.js load balancer, to keep the application available during times of heavy traffic.

Pest Patrol uses GitHub for code management. Using GitHub to manage Pest Patrol's code base allows for the use of continuous integration, continuous delivery, and a platform for publishing its documentation to the Web. Microsoft Visual Studio Code is used for developing Pest Patrol and interacting with Git issues and repo management directly.

A PostgreSQL database is used to store transactions and user data, such as profile settings, login information, and authorization. The database will also store information about pests, incidents, and discussions; this information can be used to track the movements of pests, determine the best ways for managing them, and predict potential occurrences of pest incidents in the future.

Pest Patrol utilizes machine learning for bot moderation and pest identification. Discussion thread posts that have been reported by a user and have been reviewed by a human moderator can be used to train the machine learning artificial intelligence (AI) to recognize potentially malicious content. As well, images of pests with their correct identification can be used to train the AI to automatically recognize the species of pest when a user uploads a pest incident report.

Pest Patrol uses the Google Cloud Platform to manage virtual private servers, cloud SQL instances, and interact with its map API. Google's cloud storage is also used for storing blob data, such as images. Google Cloud's SQL instances can be used to host Pest Patrol's PostgreSQL database.

3 Identification of Case Study

Pest Patrol's main users are members of a community and outdoor enthusiasts, such as hikers and campers. These users, due to being involved in the community and the outdoors, are more likely to notice and encounter pests in their daily lives. Because of their ties with the community, they would be more likely to report these pest incidences to other members of the community or seek assistance from others in the community to manage these pests.

Cities, outdoor businesses, and pest control companies are secondary customers of Pest Patrol, as these customers are less likely to make direct contributions to Pest Patrol, but may be inclined to use it. These secondary customers can use the same options as Pest Patrol's main users to view the locations of pests in their area.

The purpose of Pest Patrol is to give members of the community the means to report, monitor, and communicate pest activity in their community. With this information, community members can collaborate with each other to manage pests that are affecting them. In this case, Pest Patrol's secondary customers can work with community members to help manage pests; for instance, a pest control company can see a heat map of pest activity in a community, reach out to its community members, and offer its pest control service. As well, outdoor businesses can use the information from pest incidents to plan their activities and mitigate potentially hazardous pests. Cities can set pest alerts to be notified when an invasive species is spotted in their area, and plan accordingly to prevent its spread.

In order to determine if Pest Patrol has fulfilled its purpose, a case study group of its primary customers, community members and outdoor enthusiasts, will be selected for a beta test of the Pest Patrol prototype. This selected group will be given access to the application and will receive a walkthrough on how to use it, explaining how to make a pest report, how to start discussions, how to find pests, and how to interact with other users. Throughout the duration of this beta test, the users will relay feedback to the developers on how intuitive the user interface is, any bugs that need to be fixed, which features need to be added or tweaked, and most importantly, how well Pest Patrol assists them in becoming aware and spreading awareness of pests in their community.

Pest Patrol may also benefit a wider audience. Homeowners' associations may use Pest Patrol to observe the pest activity in their area, who can then act as representatives of their community to create plans to mitigate pests. Researchers can use the data collected from pest reports to estimate a pests population size, range, and time of year when the pest is most active. Similarly, government agencies such as the United States Fish and Wildlife Service can use Pest Patrol to track invasive species.

4 Pest Patrol Product Prototype Description

Pest Patrol's main goal is to give community members the means to report pests in their community and view the locations where pests were spotted in their area. The Pest Patrol prototype will have many of the features of the real world product, such as pest reporting, discussion threads, and partially implemented pest alerts, which will enable the prototype to achieve the goals of the real world product.

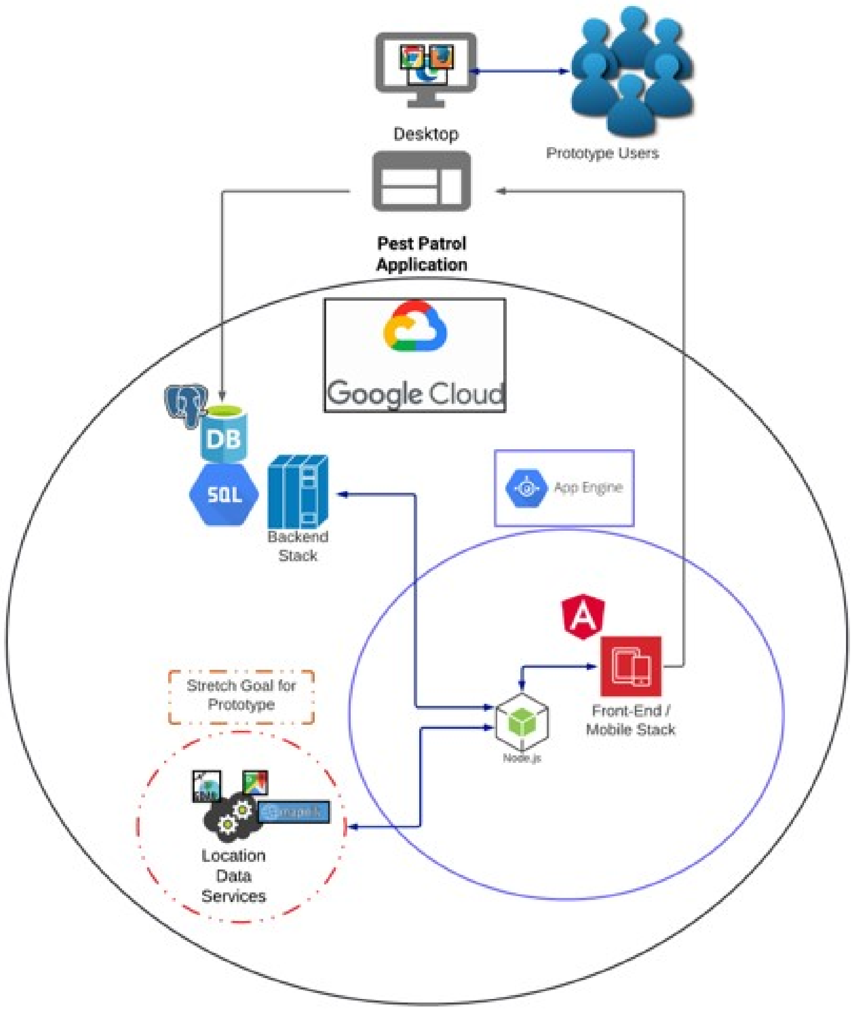
4.1 Prototype Architecture (Hardware/Software)

The architecture of the prototype for Pest Patrol will closely resemble the real world product, however, some components such as machine learning services will not be present in the prototype. With the prototype architecture, Pest Patrol will provide the essential features necessary for community members to report pests, view pest sightings, discuss local pests, and receive notifications about new pests in the area. The Pest Patrol prototype application will be a web application, allowing it to be used on any platform such as desktop or mobile through the use of a Web browser. A complete diagram of the Pest Patrol prototype's architecture can be seen in Figure 2.

(This Space Intentionally Left Blank)

Figure 2

Prototype Major Functional Component Diagram



The prototype MFCD remains similar to the real world product MFCD with the exception of the machine learning component. A front-end Angular Web application that utilizes Node.js as a middleware will run on a virtual private server within Google Cloud Platform. For the backend, a PostgreSQL database will be hosted on a Google Cloud SQL instance and will contain data on pests, discussions, and user information. If time permits, Google's location data services will be integrated into the middleware and Google's Big Query will be used to analyze large sets of incident and user discussion data that can be used to train Pest Patrol's machine learning artificial intelligence.

4.2 Prototype Features and Capabilities

Most of Pest Patrol's real world product features will be present in the prototype; however, some features will be partially implemented or have been removed for the sake of time. Table 1 shows the current status of the prototype's features compared to the real world product. Some features that will be partially implemented include: desktop and mobile compatibility, incident reporting, and pest alerts. Compatibility on desktop and mobile devices will be reliant on reactive front-end design using angular; it is possible that user interface features on desktop may not translate well to the mobile version. Incident reporting will be functional, but location tagging may be absent, which prevents community members from seeing on a map where pests have been found in their area. Pest alerts will be present, however, their use will be limited to JavaScript alerts and emails. Other features such as UI hybrid mode and user reputation have been removed from the prototype as these are extraneous features that do not immediately help Pest Patrol achieve its goals. The Pest Patrol prototype will be able to achieve the goal of giving community members the means to report pests, view local pests, and discuss pest sightings with the features currently present.

Table 1

Pest Patrol RWP vs Prototype

Function	Real World	Prototype
General		
Web and mobile compatibility	Fully Functional	Partially Functional
Dashboard	Fully Functional	Fully Functional
Hybrid Mode	Fully Functional	Eliminated
Authentication and Identification	Fully Functional	Eliminated
Password Recovery	Fully Functional	Eliminated
Incident Map		
Incident Map	Fully Functional	Fully Functional
Incident Reporting	Fully Functional	Partially Functional

Ad hoc Incident Filtering	Fully Functional	Fully Functional
Heat Mapping	Fully Functional	Partially Functional
Discussion View		
Discussion Thread View	Fully Functional	Fully Functional
Expanded discussion view	Fully Functional	Fully Functional
Follow/Subscribe to discussion thread	Fully Functional	Fully Functional
Discussion thread creation	Fully Functional	Fully Functional
Reply to discussion thread	Fully Functional	Fully Functional
Provide positive/negative feedback to threads	Fully Functional	Fully Functional
Pest Alerts		
Pest Alerts	Fully Functional	Partially Functional
Alert customization	Fully Functional	Partially Functional
Community		
Search for user	Fully Functional	Fully Functional
Add friends	Fully Functional	Fully Functional
Report Users	Fully Functional	Fully Functional
User reputation system	Fully Functional	Eliminated
Automated Moderation (ML)	Fully Functional	Eliminated
Hide flagged content	Fully Functional	Fully Functional
Account suspension	Fully Functional	Fully Functional
Flag inappropriate content	Fully Functional	Fully Functional
Content removal	Fully Functional	Fully Functional
View flagged content	Fully Functional	Fully Functional
Block user	Fully Functional	Fully Functional
Content search	Fully Functional	Fully Functional
Recent Neighborhood Activity	Fully Functional	Fully Functional
Direct Messaging	Fully Functional	Fully Functional
New thread activity notification	Fully Functional	Fully Functional
New direct message activity notification	Fully Functional	Fully Functional
New incident notification	Fully Functional	Fully Functional
AI generated notifications (ML)	Fully Functional	Eliminated
Notification customization	Fully Functional	Fully Functional
Predictive Modeling (ML)	Fully Functional	Eliminated

The implementations of these features in the prototype will help to mitigate potential risks in the real world product. Using Angular as a front-end for the web application will lessen the chance that the application is not compatible with a user's device, as Angular will make the web application reactive so that it will conform to the dimensions of the user's screen. The issue of users making false or otherwise fictitious pest reports will be mitigated by giving the users the ability to block these users, comment on these pest reports, and report them so that they may be reviewed by an administrator. The risk of the prototype's data being retrieved or modified by unauthorized users will be mitigated by using Google's data at rest encryption on the database and file storage, as well as using transport layer security for users who interact with the web application.

4.3 Prototype Development Challenges

Some of the challenges with developing the prototype include ensuring that every component of the application, such as the front-end, middleware, and back-end can all operate with each other without failure. Another challenge would be effectively deploying these components onto a server or cloud infrastructure so that they can serve the web application publicly with very little downtime and few errors.

One of the main challenges with developing the prototype is the time constraint and the learning curve that the team must overcome. As there is only one semester to produce the prototype, the team must prioritize certain features over others. This may result in features that are not fully implemented due to lack of time. For most of the team, programming with Angular or Node.js and using cloud services is new and unfamiliar, so this poses another challenge that must be overcome. Dividing work and learning how to use these tools as one works on the prototype will be integral to creating a working proof of concept by the end of the semester.

5 Glossary

Angular: a Typescript-based open-source web development framework

Back-end: The database which responds to the application's middleware

Bot Moderation: The automatic screening of user content to ensure proper user behavior

Community: The people with common interests living in a particular area broadly the area itself

Community Member: A member of a community, see Community definition

Discussion Thread: running commentary of messages between members within a community

Feedback: a positive or negative rating that can be applied to any user-created discussion thread or response

Front-end: Software element that determines how the user interface is displayed in the Web browser

Git: a distributed version control system for software development

GitHub: an internet hosting service for software development and version control using Git

Heat Map: a data visualization technique that shows the magnitude of a phenomenon as a color in two dimensions

Incident Map: a graphical map that displays the locations of all reported pest incidents for an area

Incident: An occurrence or sighting of a pest reported by a user

Infestation: A specially designated incident involving a high concentration of pests in a given area

Middleware: The application layer that queries the database, manipulates data, and relays the data to the front-end

Node.js: an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser

Pest: Any animal or plant harmful to humans or human concerns

PostgreSQL: an open-source relational database management system emphasizing extensibility and SQL compliance

User: Any individual interacting with the Pest Patrol application. This includes: community members, hikers/campers, pest control companies, homeowner's associations and administrators

W3C: The World Wide Web Consortium; an organization that develops standards for the World Wide Web

6 References

Abell, J. (2021, February 12). *Nuisance wildlife encounters on the rise*. Homestead.org. https://www.homestead.org/lifestyle/homesteading-life/nuisance-wildlife-encounters-on-the-rise/

Bugg, S., Colborne, S., & Haerther, D. (2020, March). *Great Lakes Invasives: Sea Lampreys*. Shedd Aquarium. https://www.sheddaquarium.org/stories/great-lakes-invasives-sea-lampreys

Coyle, D. (2021, September 10). *International biosecurity program prevents spread of invasive moth*. Entomology Today. https://entomologytoday.org/2021/09/10/international-biosecurity-program-succeeds-preventing-spread-invasive-moth-lymantria-dispar-asiatica/

Herring, D. (2012, March 6). *Climate change: Global temperature projections*. NOAA Climate.gov. https://www.climate.gov/news-features/understanding-climate/climate-change-global-temperature-projections

How warmer winters affect pests - Varment guard wildlife services. (2018, January 15). Humane Wildlife Removal & Control - Varment Guard Wildlife Services. https://varmentguard.com/blog/warmer-winters-affect-pests

Japanese beetle repeatedly eradicated from California. (n.d.). UC Statewide IPM Program (UC IPM). https://www2.ipm.ucanr.edu/Invasive-and-Exotic-Pests/Japanese-Beetle/

Lyme disease costs up to $1.3 billion per year to treat, study finds. (2015, February 5). Johns Hopkins Bloomberg School of Public Health. https://publichealth.jhu.edu/2015/lyme-disease-costs-more-than-one-billion-dollars-per-year-to-treat-study-finds

Parkman, K. (2021, September 23). *Latest pest control statistics for 2019*. ConsumerAffairs. https://www.consumeraffairs.com/homeowners/pest-control-statistics.html

U.S. Census Bureau. (2021, April 21). *Residents of 14 million housing units reported seeing roaches, 14.8 million saw rodents in last 12 months*. Census.gov. https://www.census.gov/library/stories/2021/04/how-many-american-homes-have-pests.html