Lab 1 - Pest Patrol Product Description

Andrew Undercoffer

Old Dominion University

CS 410

Professor James Brunelle

Apr 10, 2022

Final

Table of Contents

List of Figures

1. Introduction

Pest encounters are a common occurrence in most communities. These encounters are often dealt with on an individual basis. However, a pest encounter is rarely an isolated incident. Many pests, such as ticks, mosquitoes, and rats, are often a problem for the whole community and not just for an individual (Parkman, 2021). A problem that affects a community requires a community based solution.

Unfortunately, communities often lack the necessary tools needed to deal with pests holistically (Community Approach to Managing Pests in Homes and Schools, 2017). Members of a community have no reliable way to stay informed on local pest encounters, they often don’t share information on how to deal effectively with pests, and what information that does exist isn’t consolidated. While there are tools available to an individual, these tools treat the problem as an isolated incident.

Pest Patrol is an application designed to protect communities from pests by allowing communities to consolidate and share information about pest encounters. The application keeps community members informed on all reported pest encounters in their community. It also allows community members to track pest outbreaks with customizable Heat Mapping. Additionally, the application canl aggregate already existing knowledge on specific pests. All of these functions are implemented in real time, so that community members are always kept up to date.

2. Pest Patrol Product Description

Pest Patrol provides real-time location information on local pests, based on information submitted by local community members. This location information is displayed as a heat map that shows the concentration of various pests. Additionally, users of the application can post pest related questions to the application, to which other users can respond. The overall goal of the application is to consolidate all local pest related information in one easy and convenient location.

2.1 Key Product Features and Capabilities

Key Features of Pest Patrol include, Incident Mapping, Heat Mapping, Incident Alerts, and Discussion Board. Incident Mapping and Heat Mapping provide the user with an overview of pest activity in their local area, as reported from fellow local users. The Incident Alert is a mobile device notification that alerts users to particularly hazardous pests that have been reported near the users proximity. The Discussion Board allows for posting and commenting on local pest incident reports.

The incident map provides the user with an overview of pest encounters in their local community. Pest encounters are pinned to a map of the local area and display an icon indicating what type of pest was encountered. A question mark indicates that the species of pest is unknown. This map can be filtered by incident date, pest type, and the incident reporter.

This incident map also has an alternative heat map view. The heat map view is designed to showcase emerging pest trends for specific pests. Cooler colors on the alternative view indicate a lower probable concentration of specific pests, while warmer cooler indicate a higher probable concentration of a specific pest. These predictions are based on user reports, local historical data, and known pest behavior.

Closely associated with Incident Mapping is Incident Alerts. Incident alerts are mobile popup notifications. These notifications alert users to dangerous and persistent pest encounters that are in their vicinity. An incident report of a rabid dog, rattlesnake nest, or a bear would trigger an automatic Incident Alert for all local users. A user may opt out of this service, if they so choose.

The application also has a basic Discussion Board. This Discussion Board has a comment section for each pest incident report for a local community. This comment section displays the initial incident report at the top. Blue colored icons near the incident picture indicate that the report has been verified by geo-location. Associated comments are displayed under the “Replies” section. Thumbs up/down icons allow the user to rate the initial post and the subsequent comments. Each post and comment is tagged with the username and a date/time stamp. A quality rating is also associated with each post and comment. This shows their overall quality rating.

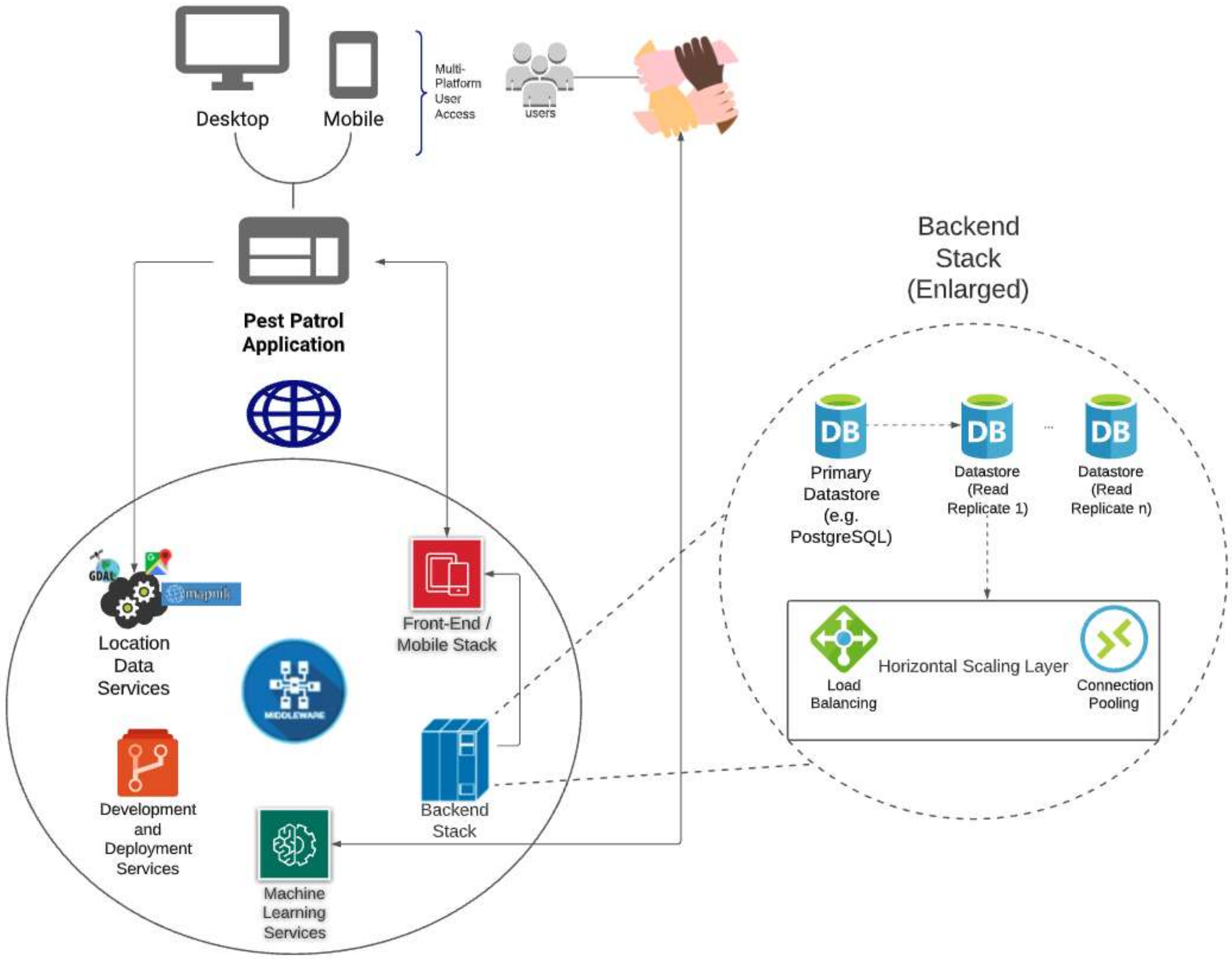
2.2 Major Components (Hardware/Software)

The application is web based. No specialty hardware is required to access the application beyond a camera and an internet enabled device. The frontend of the application is created through the Angular web framework. This Angular framework connects to the middleware Node.js. Middleware services handle a variety of tasks including, geo-location, peer messaging, and machine learning. This middleware section is also an interface between the frontend and backend. Figure 1 provides an overview of the application’s major components.

Intentionally left blank

Figure 1

Pest Patrol Major Functional Component Diagram



The frontend desktop/mobile sections connect the user to the pest patrol application, which connects to the middleware section. The middleware interfaces with the backend stack. This backend stack consists of several datastores, which implement load balancing and connection pooling. The backend also contains the application databases. All of this then connects back to the frontend/mobile stack, which connects back to the frontend.

3. Identification of Case Study

Pest Patrol is intended for use by community members, hikers, campers, and local governments. These groups all have a vested interest in avoiding and/or eliminating pests. Community members wish to protect themselves and their property. Hikers and Campers also wish to protect themselves and their property, but they also spend a great deal more time outside and this time is often spent in unfamiliar locations. The application will allow these outdoors people to better avoid pests, which will greatly improve their outdoor experience. Local governments have a duty to protect their citizens and their property from pests as much as possible. The application will allow them to better monitor and control pests that negatively affect community welfare.

The case study group for this application will be a simulated suburban neighborhood consisting of several families. This simulated community will demonstrate how pest patrol can be used to safeguard communities from pests by running through a simple use case scenario.

The simple use case scenario will have each member of this simulated group create a Pest Patrol account. Members will then log in to Pest Patrol via their account. Once logged on, the members will navigate through the application and create Incident Reports of simulated pest encounters. After these reports have been created, other members will browse the reports and leave comments. These comments will then be viewed and responded to by other members of the simulated group. This simple use case scenario will provide the development team insights regarding the difficulties of using the Pest Patrol application.

Pest control companies, homeowners associations, government agencies, and researchers will all benefit indirectly from the Pest Patrol application. The application will give exterminators a better understanding of how to target a neighborhood's pest problems and better pest management will allow homeowners associations to more effectively maintain neighborhood home prices and the data gathered by the application will prove invaluable to both government agencies and researchers.

4. Product Prototype Description

The prototype demonstrates the same basic features as the real world product. This will provide a proof of concept. The prototype will provide users with real-time location information on local pests, based on information submitted by other users. This location information will be displayed as a heat map that shows the concentration of various pests. Additionally, users of the prototype can post pest related questions to the application, to which other users can respond.

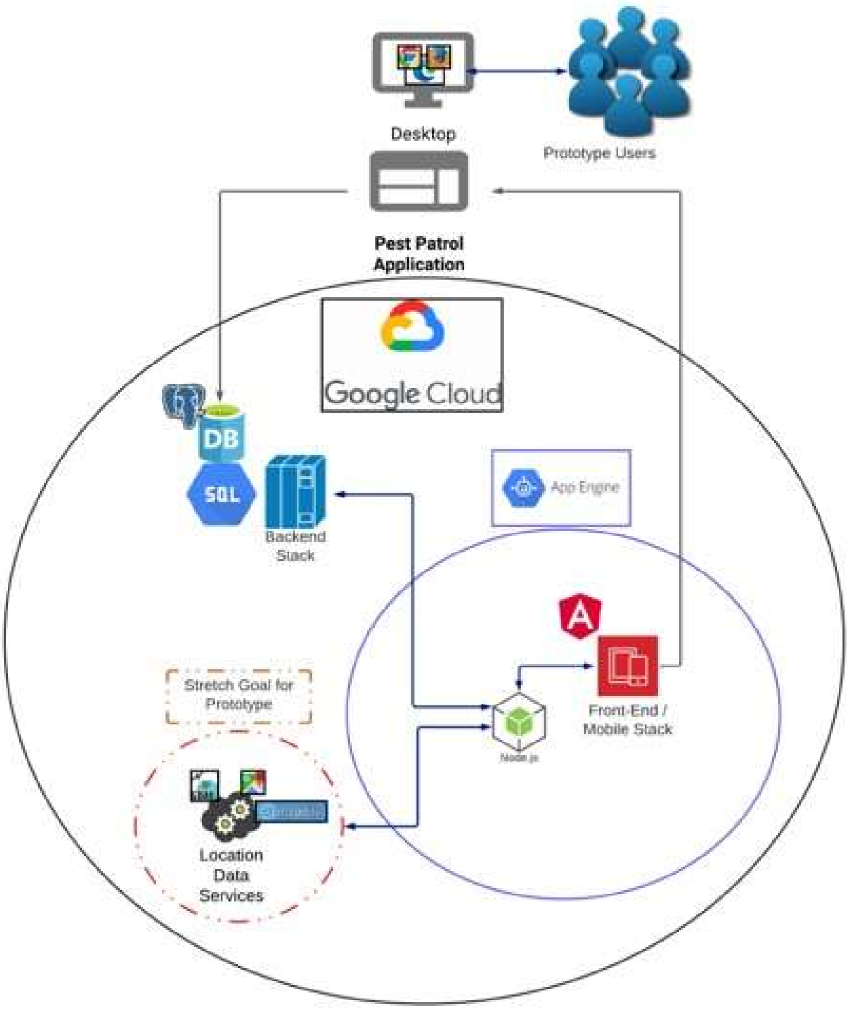
4.1 Prototype Architecture (Hardware/Software)

The prototype application is web based. The frontend of the application is created through the Angular web framework. This Angular framework connects to the middleware Node.js. The middleware section is the interface between the frontend, Location Data Services, and backend. Figure 2 provides an overview of the application’s major components.

Intentionally left blank.

Figure 2

Prototype MFCD



The frontend desktop sections connect the prototype users to the Pest Patrol application, which connects to the middleware section. Both the front end and middleware section run on the Google Application Engine.

The middleware interfaces with the backend stack and the Location Data Services. This backend stack consists of PostgreSQL instance of a Google Cloud SQL. The Location Data Services is a stretch goal for the prototype. All of these components are contained in the Google Cloud Platform.

4.2 Prototype Features and Capabilities

In order to minimize prototype development risks, as identified in CS 410, some functions have either been eliminated entirely or only partially implemented. This will leave us with a stripped down version of the real world product that will demonstrate basic feasibility.

The main features of Incident Map, Incident Reporting, and Discussion Board will still be present, however Hybrid Mode, Password Authentication, Password Recovery, Reputation System, Automated Moderation, AI Notifications, and Predictive Modeling will not be present in the prototype. Figure 3 provides an overview of the difference between the final product and the prototype.

The various functions of the application are listed in the first column, while the second and third column list to what level they are implemented in either the real world product or the prototype respectively. A dark green cell color indicates that the function is fully implemented, a blue cell color indicates that the function is partially implemented, and a red cell color indicates that the function will not be implemented.

Figure 3

Real World Product 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 incident map will be more limited than the real world product, but it still provides the user with an overview of pest encounters in their local community. Pest encounters are pinned to a map of the local area and display an icon indicating what type of pest was encountered. A question mark indicates that the species of pest is unknown. This map can be filtered by incident date, pest type, and the incident reporter.

This incident map will also be more limited than its real world counterpart. However, it will still feature the same major functions. The heat map view remains present and is designed to showcase emerging pest trends for specific pests. Cooler colors on the alternative view indicate a lower probable concentration of specific pests, while warmer cooler indicate a higher probable concentration of a specific pest. These predictions will be limited in the prototype.

The prototype also has a basic Discussion Board. This Discussion Board has a comment section for each pest incident report for a local community, it is identical to the discussion board as described in section 2.1.

4.3 Prototype Development Challenges

The vast majority of the team has little to no experience in developing a web based application. Most members will have to learn and implement some combination of Angular, Google Cloud, SQL, and NodeJS within approximately four months. This limited time frame is further compressed by the established commitments that each team member has to attend to.

The large team size may help mitigate this time crunch, but it may prove to be its own challenge. Organizing a remote team of eight without any formal structure can be incredibly difficult. The difficulties of organization further compound the challenges already presented by limited time and inexperience.

5. Glossary

Administrator: Individuals responsible for keeping an application running. For Pest Patrol these responsibilities include the moderation of user interactions and posted content.

Angular: a typescript based open-source web development framework

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

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

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

CRUD: Create, Read, Update, and Delete. They are the four basic operations of persistent data storage.

Density: The number of incidents linked to a certain pest in a given area

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

Docker: an application that uses operating system level virtualization to deliver software in packages called containers

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

Geo-tagging: The process of appending geographic coordinates based on the location of a mobile device

Geo-targeting: Method of determining the geolocation of an application user and delivering different content to that visitor based on their location

Git: a distributed version control system for software development

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

Google App Engine: a cloud computing platform for the development and hosting of web applications in Google-managed data centers.

Google Location Services: A service from Google that aims to provide a more accurate device location and generally improve location accuracy.

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

Hiker/Camper: Any individuals that engage in prolonged leisure activities outdoor

Homeowner's Association (HOA): A membership organization formed by a real estate developer to own and maintain common green areas, streets, and sidewalks and to enforce covenants to preserve the appearance of the development. It is operated for the benefit of all the residents of the community.

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

Instance: a single copy of the software running on a single physical or virtual server

Interface: frontend graphical displays of the Pest Patrol application that users interact with

Major Functional Component Diagram: a high-level visualization of the main system resources and their dependencies with one another

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 Control Company: any business entity that specializes in the regulation or management of pest species

Pest Hotspot: A number of reported incidents of a specific pest that exceed a certain threshold determined by the user

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

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

Predictive Modeling: the use of statistics to predict outcomes

SMS Messaging: Short Message Service. Is a text messaging service that uses standard communication protocols to enable the exchange of short text messages

TypeScript: Open source programming language that is a syntactical superset of JavaScript.

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

Virtual Machine: a compute resource that uses software instead of a physical computer to run programs and deploy apps

VSCode: a source-code editor made by Microsoft that includes features for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

6. References

Abell, J. (n.d.). Nuisance Wildlife Encounters on the Rise. https://www.homestead.org/lifestyle/homesteading-life/nuisance-wildlife-encounters-on-the-rise/

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

Coyle, D. (2021, September 10). Not So Fast! International Biosecurity Program Succeeds in Preventing Spread of Invasive Moth. https://entomologytoday.org/2021/09/10/international-biosecurity-program-succeeds-preventing-spread-invasive-moth-lymantria-dispar-asiatica/

Community Approach to Managing Pests in Homes and Schools. (2017, September 14). https://extension.psu.edu/community-approach-to-managing-pests-in-homes-and-schools

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

How Warmer Winters Affect Pests. (2018, January 15). https://varmentguard.com/blog/warmer-winters-affect-pests

Japanese Beetle Repeatedly Eradicated from California. (n.d.). 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). https://publichealth.jhu.edu/2015/lyme-disease-costs-more-than-one-billion-dollars-per-year-to-treat-study-finds

Parkman, K. (2021, September 23). Pest control statistics and trends. 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. https://www.census.gov/library/stories/2021/04/how-many-american-homes-have-pests.html