Lab 1- Pest Patrol Product Description

Lukas McKennedy

Department of Computer Science, Old Dominion University

CS 411W: Professional Workforce Development II

Professor James Brunelle

October 12, 2022

Final Version

Table of Contents

1 Introduction.. 3

2 Product Description.. 4

2.1 Key Features and Capabilities.. 4

2.2 Major Functional Components (Hardware/Software).. 6

3 Identification of Case Study.. 8

4 Prototype Description.. 9

4.1 Architecture (Hardware/Software).. 9

4.2 Features and Capabilities...11

4.3 Developmental Challenges...14

5 Glossary.. 15

6 References... 18

List of Figures

Figure 1: Pest Patrol Major Functional Components Diagram.. 7

Figure 2: Pest Patrol Prototype Major Functional Components Diagram.. 10

List of Tables

Table 1: Pest Patrol Features Table...12

1 Introduction

Run-ins with pests are common in most communities; as of April 2021, approximately 14,000,000 households in the United States reported seeing roaches in their home within the past year, while approximately 14,800,000 reported seeing rodents, according to the U.S. Census Bureau (Sellner & Wicht, 2021). However, these experiences are often isolated and means of reporting are scarce and not centralized, so there is rarely community awareness of, or an informed collective response to, these encounters. There is no reliable way for members of a given community to stay informed on local pest encounters; due to the lack of real-time awareness of pest presence and location, there are many encounters that could be prevented. Even though many people have pest infestations and learn from experience how best to deal with them, the guidance that they could give to people who are dealing with these problems for the first time is not consolidated or is inaccessible. The lack of awareness of current pest infestations in an area can make it difficult to effectively plan for outdoor activities and be able to mitigate pest-related problems. If someone is planning fireworks at night, but does not know that an area has recently become infested with mosquitos, they may not bring the proper mosquito repellant, which could cause them to be miserable.

A solution to this lack of communication and need for consolidation is Pest Patrol. Pest Patrol is an application designed to protect communities from pests by making it possible to proactively respond to their presence. The application works by keeping users informed of all pest encounters in their area based on reports from other users, as well as allowing communities to track outbreaks with customizable heat mapping. It also aggregates knowledge and experience gained by people who have encountered pests in the past and allows that knowledge to be utilized by community members in future encounters. As a result, users can safely plan outdoor activities with access to up-to-date information about pest problems at specific locations.

2 Product Description

Pest Patrol is a web-based, cross-platform supported application. It uses community crowdsourcing to provide real-time awareness of pest issues and locations, as well as heatmapping and predictive modeling to keep members of a community informed of pest-related situations in that community. It also contains a knowledgebase of past incidents and strategies and allows users to directly offer suggestions to other community members on how to deal with pest problems. Its goal is to make communities safer by minimizing the frequency of unwanted pest encounters as well as to diminish the severity of the pest encounters that do occur.

2.1 Key Features and Capabilities

The objectives of Pest Patrol are to provide a streamlined interface for reporting and exchanging information on pest encounters in a community, to enable users to tag reported incidents with their exact location, to provide a means for users to communicate about their encountered incidents, and to consolidate reported incidents and related discussions. The key features and capabilities that Pest Patrol provides in support of this objective include: the Dashboard, the Incident Map, Pest Alerts, Thread Activity and Recent Activity, Direct Messaging, and the Community feature. The application also allows customization of the profile through Profile Settings, as well as the ability to customize a search with the Search Options function.

Pest Patrol can be used in one of two ways: it can be used in view-only mode without registering, or an account can be created using a valid email address. Registered accounts can be customized in the Profile Settings, allowing for input of user information and photo, updating of login credentials, and customization of incident display and alert settings. The Dashboard is the first page that is loaded when logging in with an account; it provides access to all of the other features in the application. It has three view modes, including the Incident Map mode which displays the Incident Map, the Discussions mode which shows a list of discussion threads related to reported incidents, and the Hybrid mode which simultaneously displays the Incident Map, the Discussion threads, and the currently selected thread. The Incident Map lists reported incidents based on the location of the user; the incidents are customizable based on the location, age of the incident, type of pest, and user. The Incident Map allows users to report new incidents on the map, and it also displays heatmapping with alternate map views based on the age of pest activity and the type of pest. The Pest Alert function allows the user to receive either SMS or email alerts about recent incidents in the vicinity of their location. The type of alert can be changed in the Profile Settings.

The Thread Activity feature of the application displays activity related to discussion threads that the user has either created or in some way participated in. It provides quick and easy access to relevant discussions, and it also allows users to subscribe to threads they want to monitor. The Recent Activity module includes all recent events in a user’s community such as new incidents and new threads or thread activity related to the community. The Recent Activity module also includes anticipatory alerts about potential incidents based on machine learning analysis of historical data. The application’s Direct Messaging feature allows communication between users without needing to use discussion threads, and the Community aspect lets users follow or friend each other, as well as report inappropriate behavior or use of the application. The Search Options are an ad hoc way of filtering items on the incident map without needing to create specific customized settings.

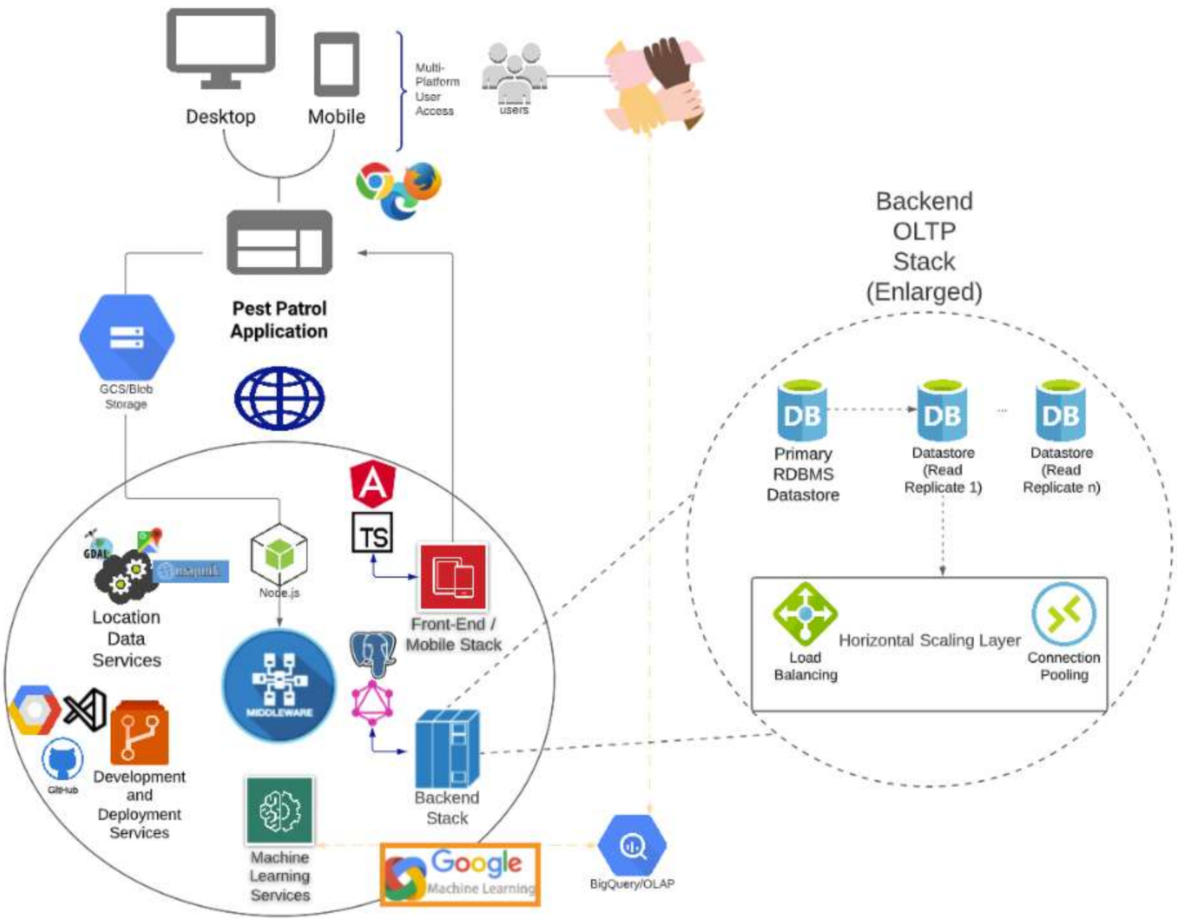
2.2 Major Functional Components (Hardware/Software)

Figure 1 is the Major Functional Components Diagram (MFCD) for the Pest Patrol application. The MFCD maps the functional parts of Pest Patrol and breaks them down into individual components from the front-end, which the user interacts with, to the backend, which stores and accesses the application data.

[This space intentionally left blank.]

Figure 1

Pest Patrol Major Functional Components Diagram



The application can either be utilized via desktop or mobile device. If used on a mobile device, the mobile application can be used; otherwise, it can be accessed through a web browser on any mobile or desktop device that has internet access, camera access, and a keyboard and mouse or touch screen. The front-end of Pest Patrol is developed in JavaScript using Angular with TypeScript; the backend Online Transactional Processing (OLTP) stack is built with PostgreSQL and Google Cloud SQL, while Google Cloud Storage is used for blob and image data. Node.JS is used for the middleware, which also includes the Location Data Services, as well as the Google

Machine Learning Services. The code is maintained in a GitHub repository and utilizes CI/CD services. The development is done through Microsoft Visual Studio Code.

3 Identification of Case Study

Pest Patrol will be useful to anyone who lives or has any activity in an area that has the potential to be inhabited by pests and has other users of the application. The main users will be people who are members of a community that experiences pest incidents. Other users include people who are planning outdoor activities, as well as outdoor businesses and even large communities or cities.

In order to determine the efficacy of Pest Patrol, there can be multiple case studies. One case study can be comprised of individual members of a community who will look for and report pests within their communities, and view alerts of other incidents reported within the communities. Another case study can consist of people planning and executing a camping trip; they will view the map of their destination, look for areas with pest sightings, and determine what they need to deal with or avoid the pests. There can also be a case study with a city government as the user: they can create plans to address large pest problems in their communities and implement them using the application as a guide.

Individuals in communities, those planning outdoor trips, outdoor businesses, and cities are customers for Pest Patrol; however, there are some other stakeholders as well. These include pest control companies who need to know where pest infestations are occurring and homeowner associations who want to know where and how many pests are being seen in their neighborhoods. Government agencies such as the United States Fish and Wildlife Service could also be considered stakeholders, as well as researchers who are interested in the data for pest population tracking or studies.

4 Prototype Description

The objective of Pest Patrol’s prototype is to demonstrate: a streamlined interface for reporting and exchanging information on pest encounters in a community, the ability to tag reported incidents with their exact location, a means for users to communicate about their encountered incidents, and a way to consolidate reported incidents and related discussions. Pest Patrol’s prototype will be implemented in a web application that will be accessible via mobile device as well as desktop.

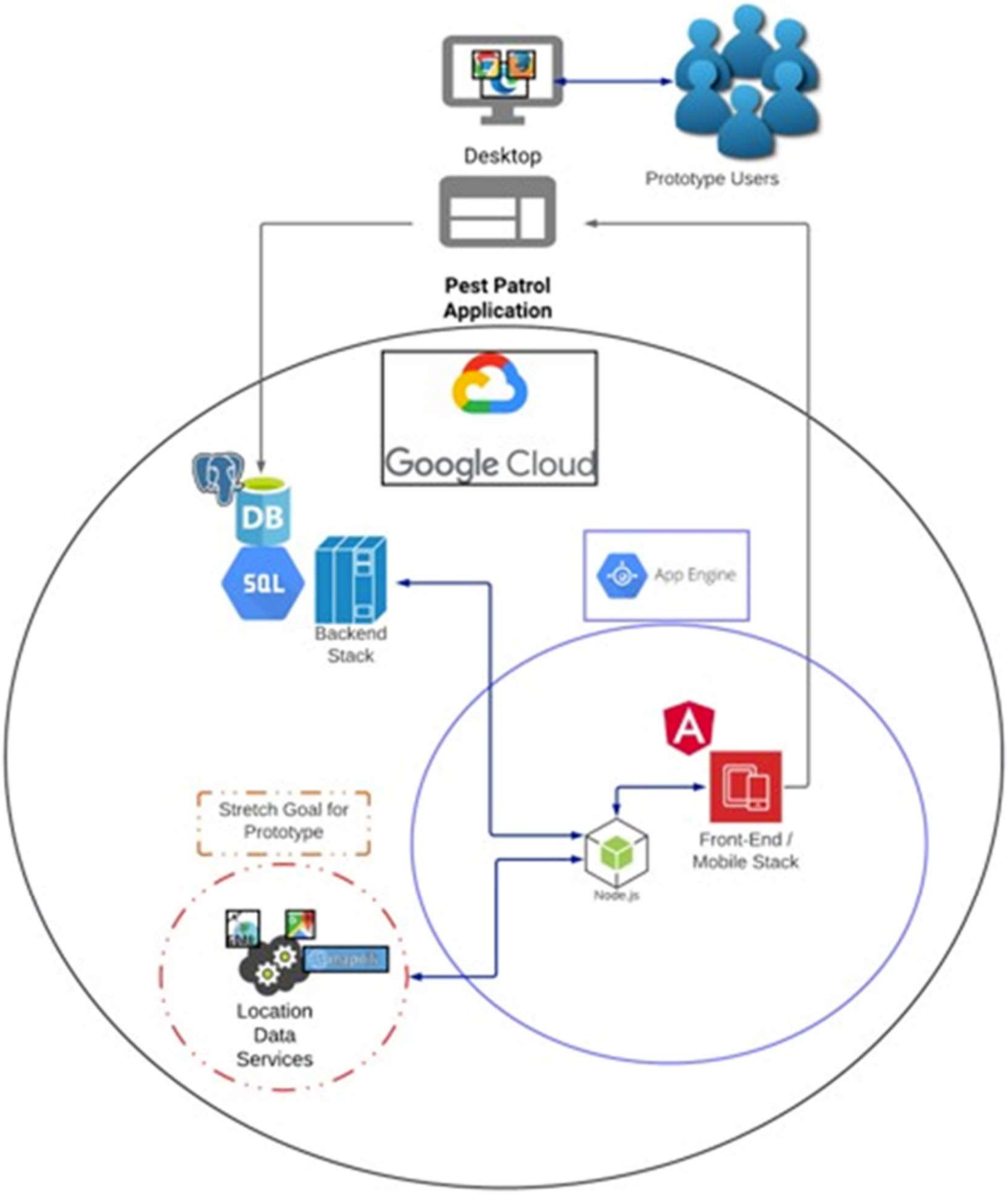
4.1 Prototype Architecture (Hardware/Software)

Figure 2 is the Major Functional Components Diagram (MFCD) for the Pest Patrol prototype. The MFCD maps the individual components of Pest Patrol that will be implemented in the prototype .

[This space intentionally left blank.]

Figure 2

Pest Patrol Prototype Major Functional Components Diagram



The application will still be accessed by either a mobile device or desktop, but only through the web browser. The front-end will still be built on Angular and will run on Google App Engine in Google Cloud. The Node.JS middleware will run on Google App Engine in Google Cloud as well. If time permits, Location Data Services will also be integrated with the middleware. The backend data of the prototype will be held in a Google Cloud SQL instance with Postgre SQL.

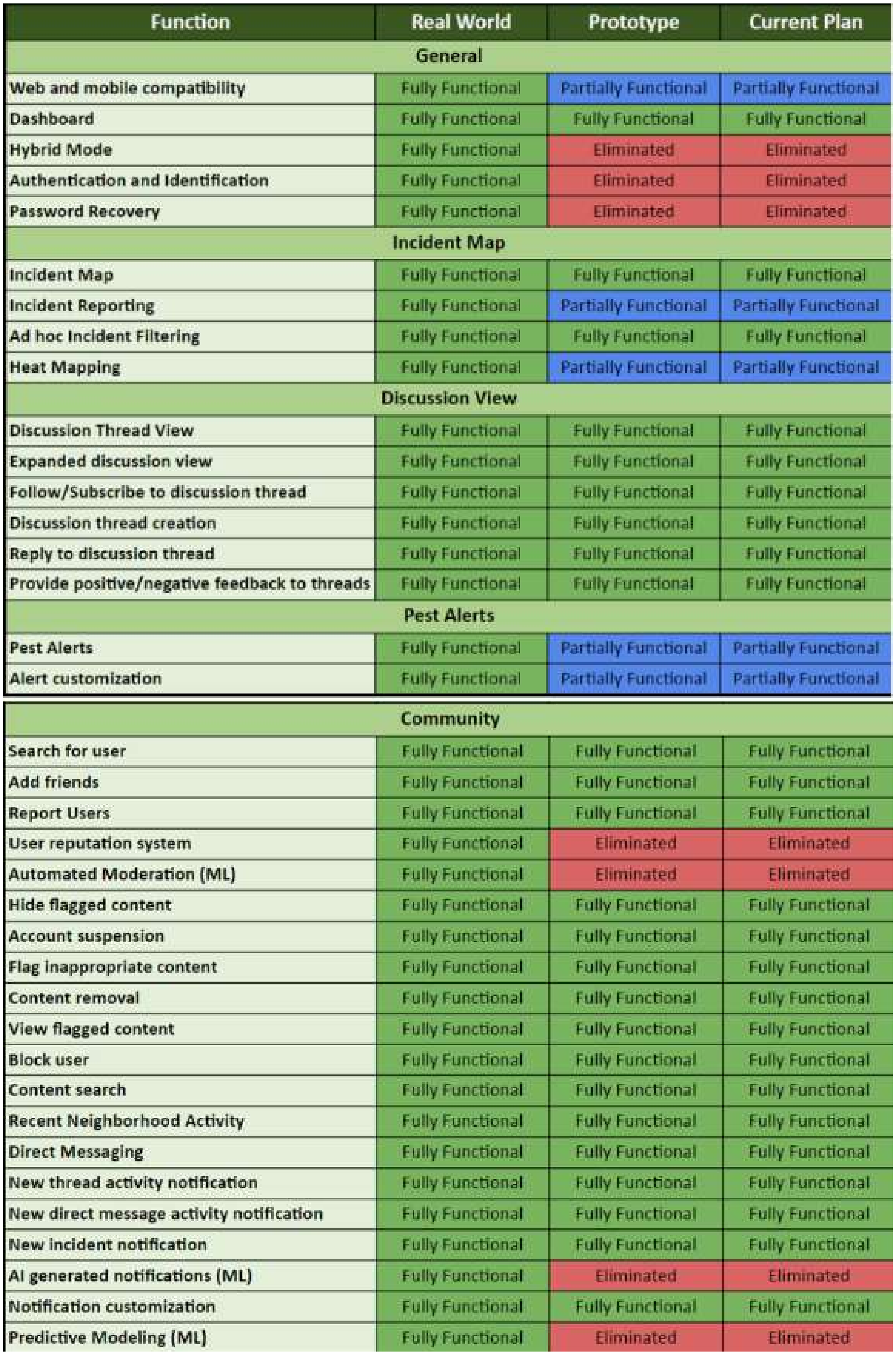
4.2 Prototype Features and Capabilities

The prototype for Pest Patrol will demonstrate the application's functionalities, including opening the web application, demonstrating the Dashboard and the ability to access the other modules, viewing the Incident Map and reporting, accessing Discussion threads, viewing Pest Alerts, and the Community feature as shown in Table 1. Through these functionalities, the prototype will demonstrate solving the problem that communities have with staying up-to-date and sharing information on pest incidents in a given area.

[This space intentionally left blank.]

Table 1

Pest Patrol Features Table



Function	Real World	Prototype	Current Plan
General			
Web and mobile compatibility	Fully Functional	Partially Functional	Partially Functional
Dashboard	Fully Functional	Fully Functional	Fully Functional
Hybrid Mode	Fully Functional	Eliminated	Eliminated
Authentication and Identification	Fully Functional	Eliminated	Eliminated
Password Recovery	Fully Functional	Eliminated	Eliminated
Incident Map			
Incident Map	Fully Functional	Fully Functional	Fully Functional
Incident Reporting	Fully Functional	Partially Functional	Partially Functional
Ad hoc Incident Filtering	Fully Functional	Fully Functional	Fully Functional
Heat Mapping	Fully Functional	Partially Functional	Partially Functional
Discussion View			
Discussion Thread View	Fully Functional	Fully Functional	Fully Functional
Expanded discussion view	Fully Functional	Fully Functional	Fully Functional
Follow/Subscribe to discussion thread	Fully Functional	Fully Functional	Fully Functional
Discussion thread creation	Fully Functional	Fully Functional	Fully Functional
Reply to discussion thread	Fully Functional	Fully Functional	Fully Functional
Provide positive/negative feedback to threads	Fully Functional	Fully Functional	Fully Functional
Pest Alerts			
Pest Alerts	Fully Functional	Partially Functional	Partially Functional
Alert customization	Fully Functional	Partially Functional	Partially Functional
Community			
Search for user	Fully Functional	Fully Functional	Fully Functional
Add friends	Fully Functional	Fully Functional	Fully Functional
Report Users	Fully Functional	Fully Functional	Fully Functional
User reputation system	Fully Functional	Eliminated	Eliminated
Automated Moderation (ML)	Fully Functional	Eliminated	Eliminated
Hide flagged content	Fully Functional	Fully Functional	Fully Functional
Account suspension	Fully Functional	Fully Functional	Fully Functional
Flag inappropriate content	Fully Functional	Fully Functional	Fully Functional
Content removal	Fully Functional	Fully Functional	Fully Functional
View flagged content	Fully Functional	Fully Functional	Fully Functional
Block user	Fully Functional	Fully Functional	Fully Functional
Content search	Fully Functional	Fully Functional	Fully Functional
Recent Neighborhood Activity	Fully Functional	Fully Functional	Fully Functional
Direct Messaging	Fully Functional	Fully Functional	Fully Functional
New thread activity notification	Fully Functional	Fully Functional	Fully Functional
New direct message activity notification	Fully Functional	Fully Functional	Fully Functional
New incident notification	Fully Functional	Fully Functional	Fully Functional
AI generated notifications (ML)	Fully Functional	Eliminated	Eliminated
Notification customization	Fully Functional	Fully Functional	Fully Functional
Predictive Modeling (ML)	Fully Functional	Eliminated	Eliminated

The Dashboard will be fully implemented, but Hybrid Mode, Authentication and Identification, and Password Recovery will not be implemented in the prototype, due to not implementing account features. The Incident Map itself will be fully implemented along with Ad Hoc Incident Filtering, but the Incident Reporting and Heat Mapping will only be partially implemented, just enough to be able to demonstrate proof of concept. The Discussion View will be completely included in the prototype, including the Discussion Thread View and ability to expand as well as the abilities to follow, create, reply to, and provide feedback on discussion threads. Pest Alerts will also only be partially implemented. Most of the Community Function will be usable in the prototype, including the abilities to search for, friend, and report users. The ability to deal with inappropriate content will also be implemented, such as being able to hide content, suspend accounts, flag or remove inappropriate content, and block users. The prototype will allow for searching for content, looking at recent neighborhood activity, and direct messaging. Notifications will be implemented for New Thread Activity, New Direct Message Activity, and New Incidents. AI Generated Notifications and Predictive Modeling will not be implemented, and neither will the user reputation system or machine-learning-based Automated Moderation.

The features that are being included in the prototype will sufficiently achieve the prototype's objectives. The Incident Reporting feature that is being implemented will demonstrate the objective of having a streamlined interface to report and exchange information on pest encounters in a community, along with the implementation of the Incident Map. The Incident Map will also give the ability to tag reported incidents with an exact location. The implementation of the Discussion Thread view and the capability to create and reply to threads, along with the Community features, will demonstrate the ability for users to communicate about encountered incidents and consolidate reports and related discussions.

4.3 Prototype Development Challenges

The development of this prototype will come with many challenges. There will be technical challenges due to unfamiliarity with the many different platforms and frameworks that will be used. Inexperience with tools such as Angular and JavaScript on the front-end, Node.JS for middleware, or PostgreSQL on the backend can cause challenges. This project has several interconnected parts that need to work together perfectly in order for the application to work correctly, which could prove challenging. The use of containerization through Docker also adds another layer of complexity to the project. Another development challenge will be version control and collaboration, due to the number of team members working on components that rely on each other. This will make communication and proper use of Github critical to maintaining the integrity of the project.

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: 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: A 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, backend 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: 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/

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

Sellner, M. & Wicht, J. (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