

Tram Nguyen

REINDEER TRACKING APPLICATION

REINDEER TRACKING APPLICATION

Tram Nguyen
Bachelor's thesis
Autumn 2019
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Development

Author: Tram Nguyen

Title of the bachelor's thesis: Reindeer Tracking Application

Supervisor: Veijo Väisänen

Term and year of completion: Autumn 2019

Number of pages: 62

Anicare is a foundation which provides a smart solution solving reindeer herding challenges. It was born in Finland with keeping reindeer herding beneficial and meaningful source as a main goal. By bringing the modern tools to the reindeer herding, Anicare offer help with the latest technology to this traditional source of livelihood.

The main aim of this bachelor's thesis was to introduce the Anicare mobile application that help herders track in real-time and locate whereabouts of the reindeer remotely. Anicare application will go with a health-tag which is a GPS device attached to the earflap of the reindeer. Together, the owner can apply the rescue and treatment for an injured or sick reindeer quickly with the exact GPS location. The Ionic Framework was chosen to develop this hybrid application. The process consisted of analysing, designing, developing and testing. HTML5, SCSS, Angular, Typescript, PHP and MySQL were manipulated during the design and implementation. A workflow management tool Trello was indispensable to be utilized through the creation cycle. The process consisted of three stages: a server written with the Lumen Framework, database backends and a hybrid mobile application development.

The result of this thesis mainly focused on the Android mobile application with a simple and intuitive User Interface, providing accurate information stably. Exploring Anicare application development methods includes the mobile application architecture overview, concept preparation, development guidelines, testing and execution techniques. The described process theory and methodology have theoretical systemic and engineering practice foundations. The full documentation of the application as well as the project implementation were explained entirely in the thesis.

Keywords: reindeer, Ionic, Angular, hybrid application

PREFACE

First of all, I would like to express my deepest and profound gratitude to Mr. Veijo Väisänen, the instructing professor and thesis advisor, for guiding and advising me during this work. The professor's rigorous academic attitude, profound knowledge, and selfless dedication have inspired me. His kind support and valuable resources went through the process of my thesis. Without his instructions, the thesis could have been unable to be done successfully.

I desire to acknowledge my gratitude to Mr. Aki Marttila, the founder of Anicare company. He has given me the opportunity to accost and flourish ideas, then blown up my motivation in developing the application. His professional working methods, understanding, sophistication and sincere feedback, helped me to constantly improve myself and advance my skills.

My deeply thanks to my parents, for their endless love, unconditional support, motivation and guidance on every walk of life. Thanks to them, I had many opportunities to interact with the advanced modern technology in the world. They have given me an ideal environment to learn and explore and experience the techniques in my writing.

I would like to send my appreciation to my heartening friends to emphasize their kind encouragement and support during my thesis writing. To my colleagues at Oulu University of Applied Sciences and Kyunghee University, I would like to thank you for your assistance and patience while studying the technologies with me.

At last, I wish to thank the respondents who enthusiastically participated in answering the usability test questions to help me complete this thesis.

Oulu, 23.07.2018
Tram Nguyen

CONTENTS

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
ABBREVIATIONS	7
1 INTRODUCTION	9
1.1 Purpose	9
1.2 The Methodology and Idea	10
2 PREPARATION FOR DEVELOPMENT	13
2.1 Tech Stack Choice	13
2.2 Application Architecture	14
2.2.1 Front-end	14
2.2.2 Back-end	15
2.3 Development Tools and Working Environment	16
3 THEORETICAL BACKGROUND	22
3.1 Ionic	22
3.2 Angular 7 and TypeScript	23
3.3 Lumen	25
3.4 MySQL	26
4 IMPLEMENTATION	28
4.1 UI and Logo Reform	28
4.2 Features	30
4.2.1 Map	31
4.2.2.1 Plugins	31
4.2.2.2 Functionality	32
4.2.2 Push Notification	35
4.2.3 Registration	38
4.2.4 Share	40
4.2.5 Report	42
4.3 Server	44

4.3.1 RESTful API	44
4.3.2 Local Testing Environment	46
4.3.2.1 Homestead, Vagrant and Virtual Box	46
4.3.2.2 Postman	47
4.3.3 API	49
4.3.4 Database	54
5 VISUAL VIEW	56
6 CONCLUSION	58
7 REFERENCES	60

ABBREVIATIONS

API Application Programming Interface

CSS Cascading Style Sheets

ES ECMAScript

FAB Floating Action Button

JSON JavaScript Object Notation

GPS Global Positioning System

HTML Hyper Text Markup Language

IDE Integrated Development Environment

NB-IoT Narrowband Internet of Things

MVC	Model - View – Control
MySQL	Authentication, Authorize, Accounting
ORM	Object Relational Mapping
PHP	Hypertext Preprocessor
SCSS	Sassy CSS
REST	Representational State Transfer

1 INTRODUCTION

1.1 Purpose

A reindeer is an integral part of nature and culture in northern Finland. The main challenges facing reindeer husbandry are the increasing pressure from the industrial land use and the damage caused by large carnivores and other human-induced damage. According to the Finnish Ministry of Agriculture and Forestry's researches, reindeer husbandry practitioners believe that large carnivores are the most significant economic threat to reindeer husbandry (1.). The reindeer herders would be paid compensation for a reindeer killed by a large carnivore due to the supported policy from The Finnish Wildlife Agency. To get compensation, the killed reindeer must be found. Previously, herders would have a hard time to find the whereabouts of their reindeer, in case, unfortunately, the reindeer was lost, ill or dead. Dogs had been trained to find the tracks, but they were not enough potent and productive. (2.)

The Anicare Company has been cognitive that reindeer tracking is important for the reindeer industry, as most of the dead reindeer in the forest are not found. A great demand for an effective monitoring technology in reindeer husbandry throughout Finland was identified and developed. The main aim was helping reindeer husbandries take a step in the most modern direction with the help of a new NB-IoT network method and mobile technology. Therefore, the Anicare mobile application was designed and built. A reindeer can now be tracked in real time to find lost animals. The HealthTag device will monitor the movement and temperature of the reindeer, as well as record and update this into the system via

Anicare's server, which is relying on information including the increase or decrease of temperature, movement and position. A reindeer will be considered dead if the information has not changed for one to two days. With the Anicare application on hand, the reindeer herders will be allowed to control the unexplored disappearances of reindeer by some simple taps via their smartphone. They can access real-time and reliable information which is available for reindeer movements and disappearances.

The steps leading to the usage of the Anicare mobile application can be explained as following:

- Reindeer owners have a smartphone with Android.
- Reindeer owners buy a Anicare HealthTag device from the Anicare web shop.
- Reindeer owners get a HealthTag product in the sales packet.
- In the sales packet, there is a guide for a new Anicare user, who needs to create an account via website
- A tracker serial number registration in the application can be found under the QR-code
- Downloading the application and registering for the serial number

Currently, the application has both Finnish and English languages and it is suitable for Finns and foreigners.

1.2 The Methodology and Idea

With the rapid development of the mobile Internet technology, there are many operating systems and multiple types of mobile intelligent terminals which have rapidly become popular. The traditional native development model can no longer meet the cost and efficiency requirements of enterprises and developers, while the hybrid model has taken the advantage. Hybrid applications are independent on the platform. They have the advantages of Web App across platforms and terminals, and at the same time they look like a Native App. This concept uses a variety of system level services to provide a good user experience. It is possible to create an application which supports both iOS and Android platforms and reduces the total development cost. An Ionic framework was implemented using Apache Cordova, which bridged the gap between JavaScript and device APIs. Apache Cordova helped network and mobile developers build world-class hybrid mobile applications for multiple platforms with only one code base. (3.)

In this thesis, a context to create an Anicare mobile application for Android is going to be explored. This is a hybrid mobile application. The application design is easy to be utilized covering for all the functions. The user interface was set to be as minimal and convenient as possible. The main screens took the map and the navigation of objects as a pivotal point. The UI design circled around the concept called WYSIWYG. This abbreviation stands for What You See Is What You Get.

Apache Cordova is a core open source project where the development and contributions take place. Ionic is an idea choice for building the application, working with Angular. To start a hybrid application that uses Apache Cordova's mobile platform, the needed tools are Java IDE for Android, Android SDK, mobile devices and the Android virtual device simulator. The next part proposes how to use Ionic, Angular and other web technologies and the Cordova package to deal with cross-platform mobile application development issues. (3.)

In the server-side, a shortened modern version of Laravel with a major focus on developing powerful API and microphone services with an extraordinary speed, Lumen was considered as a potential choice for this project. The well-known power quartet LAMP (Linux, Apache, MySQL, PHP) were implemented to build the full application. The process involved designing, development, testing and modifications. During this application process, Trello Boards was used to organize, reflect tasks and follow workflows. Most tasks from the feedback document are moved to a Trello task card and it is easy to track the status of every task.

2 PREPARATION FOR DEVELOPMENT

2.1 Tech Stack Choice

Hybrid mobile apps are like any other mobile app. They can be installed on your phone. You can find them on the app market. With them, you can play games, connect friends through social networks, take photos and track your health. Similar to other websites on the Internet, hybrid mobile apps are also built using a combination of web technologies, such as HTML, CSS and JavaScript. Then, they are covered with an external container to become like the native mobile app and downloadable on the Appstore. The only difference is that mobile hybrid applications are stored inside a WebView native application (Imagine that a WebView as a browser window is configured to run a full screen inside an application). It is because of this, they can gain access to the hardware-based functions of the phone through the API of plugins, such as an accelerometer sensor, a camera, contacts. These are often restricted capabilities when accessed on mobile browsers. (4.)

Furthermore, hybrid mobile applications may also include Native UI components in necessary cases. They not only bring a good user experience, but also the application development though not simple but less complicated than the native mobile app. (4.) Hybrid mobile application development allows developers to target more than one platform and save coding costs multiple times. Plus, updating the content will be easier for both developers and users. A new version will come immediately and transparently to the user. Just a change at the server and users who are allowed to access the application will get it updated.

2.2 Application Architecture

Designing an architecture assure project must not only work well but it must also be well organized. The proper architecture has saved the number of times, effect and cost. The architecture is considered as a process of developing a sustainable structured application acting according to a set of certain patterns and techniques. With the help of reasonable architecture components, the proceeding of development, maintenance, debugging as well as understanding becomes smoother and more efficient. As mentioned earlier, one of the advantages is the ease of change. Adding new features and modifications becomes easy and fast if components are rationalized and communicated in an organized manner. This separation leads to many other advantages, such as the ability to test, giving the possibility to check components independently. (5.)

2.2.1 Front-end

The aim was to create an application that works on both IOS and Android platforms, therefore Anicare started with cross-platform development. Ionic, Angular and Cordova were chosen for creating this particular application. A mobile front-end display, based on Ionic's UI framework, rich libraries and components, HTML and CSS mix, can handle the cross-platform and the screen adaptation of Android and IOS problem. Logic was written in Angular and native JavaScript. Angular followed the MVC pattern of software engineering and provided a solution for web applications like traditional server-side programming. (4.)

2.2.2 Back-end

The Back-end side, included a web API, a server and a database. The server part is responsible for the performance, scalability and connecting to the Internet of the application. Web servers respond to data transmission to and from the application. Taking the idea of microservices and interacting with APIs that focus on the server, the microframework Lumen has met the necessary criteria. During the process, the modern attribute of Lumen helped to maximize performance. Seeing photos from Lumen's homepage, it can be seen that this microframework has a superior speed compared to other similar types of microframework in the market. See figure 1 below.

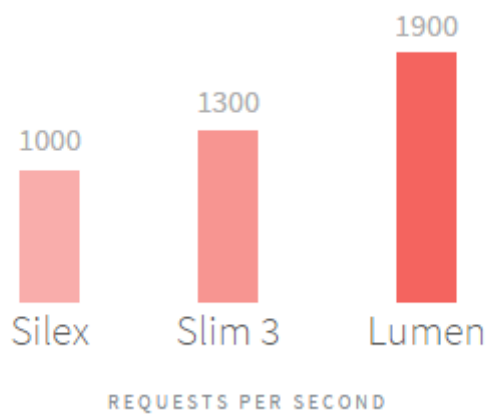


FIGURE 1. Benchmark Breaking Speed (6.)

In the future, if the Anicare company wants to upgrade the application to a higher level, switching from Lumen to Laravel is also very easy. Mostly, Lumen is already configured. In fact, there is no setup path (config) when installing Lumen. Instead, there is a file with the .env extension to configure most of applications. Lumen does not use the Symfony router as its predecessor Laravel. This is probably the

greatest difference and also why it can be so fast. Instead, Lumen uses Fast Route, a faster route implementation library developed by Nikita Po-pov. (6.)

Relational database management system (RDBMS), MySQL store data in the form of "relational": tables with rows and columns where all data information is stored as a value of a specific cell. Data in a RDBMS is managed through well-known SQL protocol - structured query language. It is useful in handling well-structured data and supports 4 attributes which were identified as vital for Anicare development: Atomicity, Consistency, Isolation, Durability. Data structures can also be expanded quickly, adding new data does not affect the data available. (7.)

2.3 Development Tools and Working Environment

Java SE Development Kit (JDK) and Android SDK need to be installed to Cordova. After the JDK installation is complete, the system would require adding new JAVA_HOME system variables and JDK paths. The Android SDK tools has been using Android Studio because it is currently the best way to quickly and easily install all the necessary things for the Android development. Similar to JDK, new ANDROID_HOME system variables and Android SDK paths are required to be add. The list of tools includes:

- Android development kit (Android SDK, Android SDK Manager, Android SDK platform tool, Android SDK build tool)
- Android Emulator with a lot of Android configuration
- IDE
- gradle

Visual Studio Code was applied to develop and debug the Ionic project. Visual Studio Code (VSC) is undoubtedly the most popular lightweight code editor. The VSC plugins are already quite popular and essential to JavaScript developers. Its supports are diverse, including code snippets for specific technologies, syntax highlighting, Emmet, and smart reminders. Furthermore, VSC provides a wealth of debugging tools. Plugins have been added to assist the coding process:

- Ionic Extension Pack: This package contains plugins for Ionic, Angular, Cordova and HTML.
- JavaScript Snippets: Provides a collection of ES6 code snippets.
- ESLint: This plugin integrates ESLint into VSC in order to avoid common mistakes in the code. Its rules are configured in `.eslintrc.json`.
- JavaScript Booster: An amazing code refactoring tool which is satisfying the need for code operations, such as turning `var` to `const` or `let`, removing extra `else` statements, merging declarations and initialization.
- Rest Client: Instead of testing your REST API endpoints with a browser or a CURL program, this tool sends HTTP requests directly to each other in the editor.
- Cordova Tools: Supports the Cordova plugin and the Ionic framework, providing intelligent hints for Cordova-based projects, debugging support for other features.
- ES Mocha Snippets: A Mocha code snippet that provides ES6 syntax. The focus of this plugin is to use the arrow function to minimize the use of curly braces and keep the code compact. A semicolon can be allowed by setting. (4.)

API is a bridge between the client and the server, a bridge between two objects. The tool for testing API is Postman. Postman was mainly used to test the params transfer or see the results returned from a Rest API in this project. It is easy to use and support by running both the UI and the non-UI.

For managing the MySQL database through phpMyAdmin, a tool was written by PHP, administered via a web browser instead of insipid command line interface. Using phpMyAdmin helped increase the database management performance. When working with phpMyAdmin, a significant increase in efficiency compared to using the command line window was recorded and confirmed. (6.) phpMyAdmin is designed to help accomplish common tasks, such as viewing a list of databases on servers, viewing the structure of a table, inserting data into tables, and changing table structures quickly. This can be seen when looking at a specific interface. See the figure 2 below.

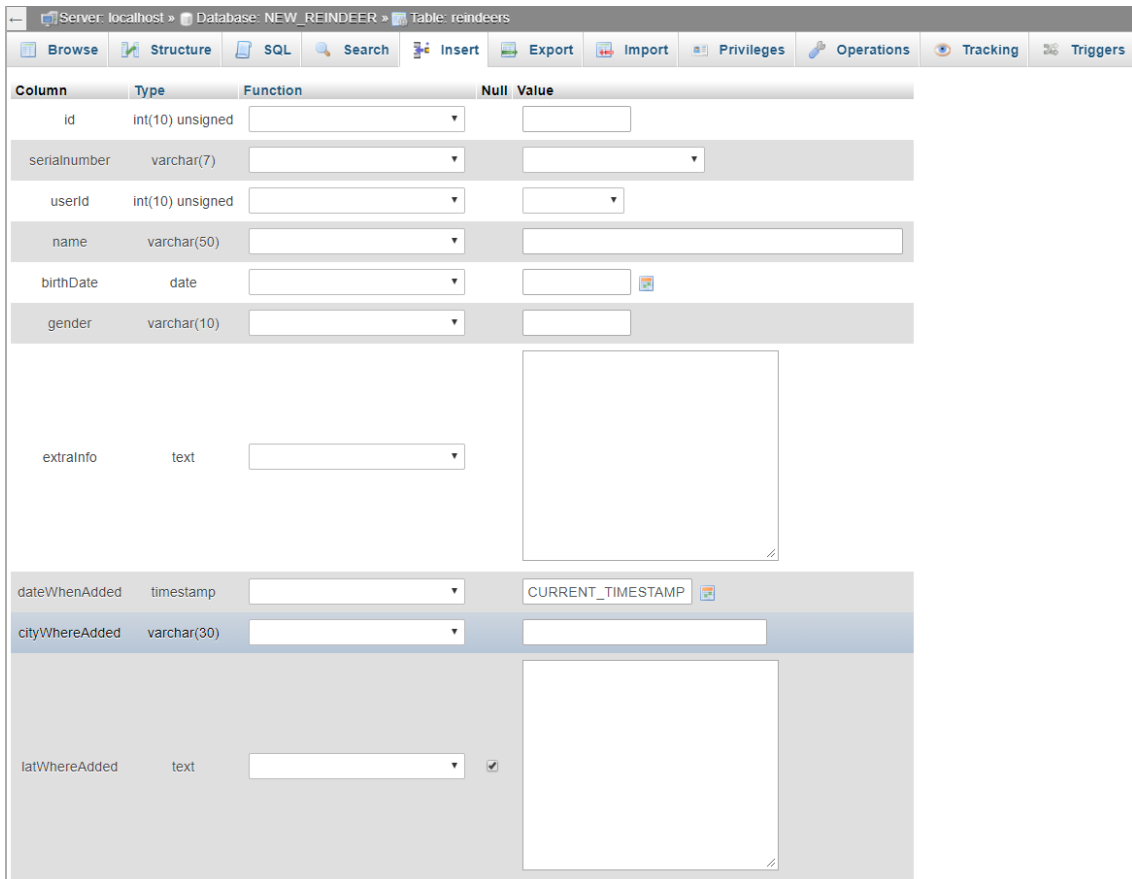


FIGURE 2. phpMyAdmin Insert Interface

With the above interface, users will quickly know how many databases are created on the server by looking at the left-hand frame. When entering data, users can also know the type of data that must be entered for each field to avoid the incorrect data entry. However, not only the common tasks that have been accomplished with the command line window, phpMyAdmin also helps developer during the scheduling process by the design view in an intuitive way. See the figure 3 below.

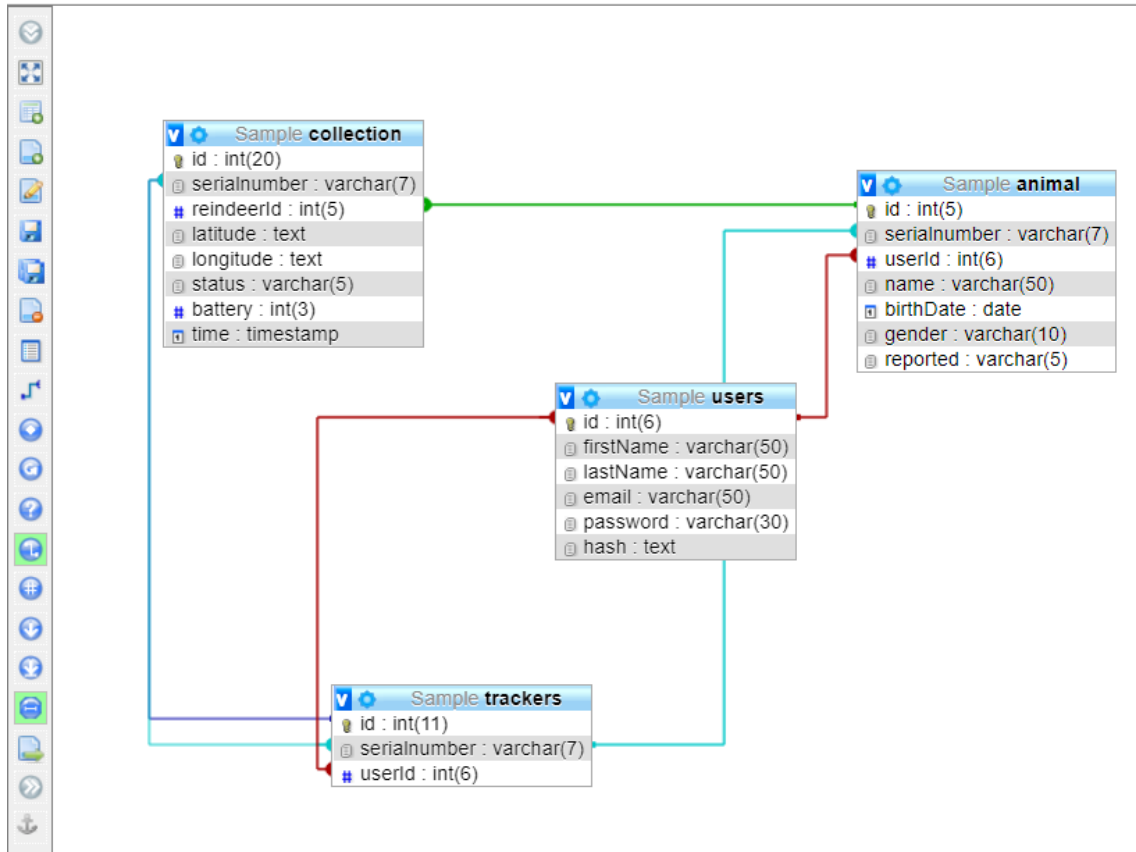


FIGURE 3. Example of phpMyAdmin Designer View

While working with the database, there were unpredictable situations, such as SQL injection, user errors, and other database corruption cases. Those situations could have been handled through phpMyAdmin whilst working with an object. In particular, it has the ability to back up MySQL automatically. This ability has helped the author at the time amending and redesigning a new database structure which is the official current one. For the security, Anicare restricted access to the URL of phpMyAdmin for the certain IP to avoid possible risks to the database.

GitLab was decided as a Git code management. GitLab has the advantage of viable long term, based on a solid community with hundreds of thousands of organizations that use and contribute to software. (7.) This means that GitLab is more usable because it will not be reliable if there is only one company supporting the software. GitLab managed sharing code between the author and the company, provided bug tracking, Wiki space and other tools for "social coding". Laravel Homestead, Vagrant and Virtual Box are the combo for running the demo and testing in local before pushing the codes and commit to Git. Those will be discussed in detail later.

3 THEORETICAL BACKGROUND

3.1 Ionic

Ionic is a framework which is utilized to develop a Hybrid application for the mobile. Hybrid can be understood as a hybrid between native and mobile web applications. The nature of hybrid is an interface written by HTML, CSS and JavaScript but it allows to call the system's native API. (3.) So, it can manipulate the mobile operating system as other native applications. Ionic is currently the most promising HTML5 mobile application development framework. The purpose of the Ionic framework is to develop mobile applications from a web perspective. Based on the PhoneGap compilation platform, applications compiled into various platforms can be implemented. The great Ionic characteristics are:

- Ionic is based on Angular grammar and it is easy to learn
- Ionic is a lightweight framework
- Ionic provides the UI design for mobile apps
- Ionic performance is superior and runs fast

Ionic provides a rich library of CSS, including navigation bars, footers, buttons, lists, menus, toggles, tabs, and grid layouts. Ionic also offers a rich library of colour styles and icons that look like icons. Like the native app, they adapt to Android and IOS. Ionic uses TypeScript which is similar to normal JavaScript for its codes. (8.) Many components require TypeScript to produce magical effects, although usually components do not require encoding and can be easily used through framework extensions. Moreover, Ionic provides, for example, Slide Box, Modal, Action Sheet, Popup, Loading, Platform, Gesture, Utility and the UI components

of the Keyboard. Ionic integrates Cordova's build environment, so the developer can use Ionic command-line tools instead of Cordova commands to package and run projects. (9.)

Ionic is not an alternative to Cordova, however, it is a UI library to make the Cordova project better. Ionic is a framework built on Angular and Cordova Ionic can compare with frameworks such as Bootstrap or Foundation, but only for Mobile and not for Web. Ionic and Angular is responsible for the implementation of the page, and Cordova is responsible for wrapping the implemented pages into native applications. Like peanuts, the innermost peanut kernel is Angular, the peanut kernel is Ionic, and the outermost peanut shell is Cordova. (9.) Here, in this Anicare application, Ionic version 3 and Angular 7 were implemented.

3.2 Angular 7 and TypeScript

On the grounds that the primitive project was sketched with scribbled ideas and started working from the end of 2018, Angular 7, the latest framework version of the year was chosen to implement the project. Angular 7 simplifies APIs, it has fewer directives, making it easier to remember, easier to use, having simple data binding, and being less complicated in overall concept.

Angular 7 had a simple solution to the many pain points which had been encountered on the project. There were TS (TypeScript) and RXJS blessings to make the development efficiency fly fast, and the robustness is guaranteed. It allowed the developer to jump directly from the HTML template to the method name in TS, then break the debugging directly in the VSC, quickly find the definition and reference of a method or property. With the dependency injection system, it was

convenient to use the service class for communication between components and various other things. (10.)

TypeScript is an open source project developed by Microsoft. It can be considered an enhanced version of JavaScript by adding the static and object-oriented class options. With static typing, the TypeScript code is easier to predict, and easier to debug. TypeScript can be used to develop applications that run on client-side (Angular) and server-side (NodeJS). TypeScript uses all the features of the latest JavaScript (ES6, ES7 and ES8), such as classes or modules. TypeScript makes the vision that JavaScript playing in Java style. (11.)

While interacting with this language, the conclusion is deduced that the nature of TypeScript is to compile JavaScript scripts so that the developer can run it anywhere as long as it supports the JavaScript compilation. Alternatively, developers can use the JavaScript syntax mixing into TypeScript, which makes it easier for learners to approach TypeScript. Ease of developing large projects is one of the advantageous points for this project in the future. (11.) See the figure 4 below to see the popularity of TypeScript.

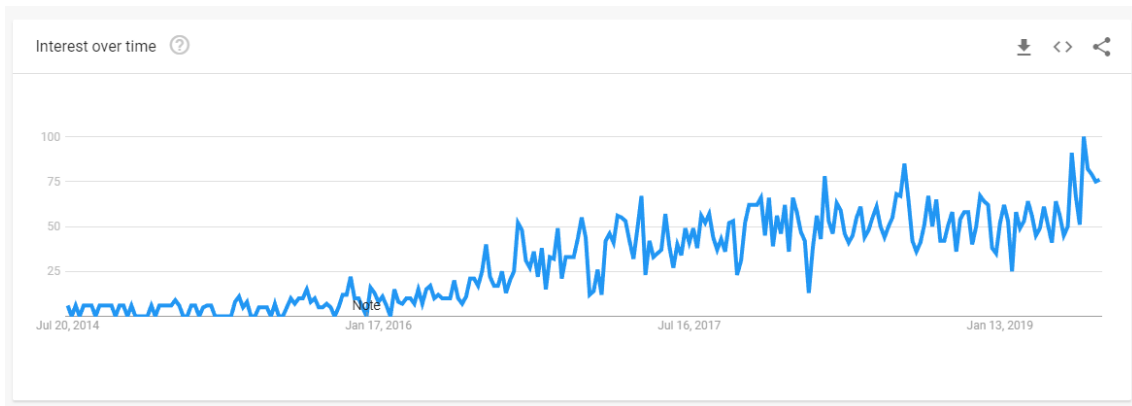


FIGURE 4. The recently increasing popularity of TypeScript from Google Trends

3.3 Lumen

Lumen is a wonderfully fast microservice PHP framework. The project aimed to fluently operate Microservices in order to automate the infrastructure to reduce the deployment and operational complexity. Microservices is an architecture that links multiple small services by the API which achieves the processing speed improvement. Also, when a failure occurs, it is painless to check the impact of it for each service. Lumen is designed for microservices and APIs. Lumen's strength is that it was built on Laravel to make it powerful and the having an incredible speed. (12.)

Lumen is a loading mechanism which optimizes the framework for Laravel and has much less resource requirements. The Artisan command under Lumen, which is responsible for the ease to create RESTful APIs, is less complicated than Laravel. It seems that Lumen is giving up and throwing away the creation of a feature-rich app from where the commands are being truncated. The command tool only plays a role in assisting the development and improving efficiency. In-

stead of the config files, the necessary settings are done in the app file of Bootstrap directory or .env. This seems to be the result of tuning the boot process, throwing away flexibility and extensibility to speed up the application. (12.)

The server used some basic features, such as routing, data-base, authentication, eloquent and middleware, so Lumen would have not been discussed deeply. In the forward time, if the project grows larger and requires more complicated structure and functions as full-stack frame, Lumen's migration to Laravel is very easy, and the code changes are not large.

3.4 MySQL

The database is stored and managed in database management systems, in which MySQL is one of the database management systems - Database Management System (DBMS). The SQL statement (Transact-SQL) is the MySQL language used to exchange data between the Client and the MySQL Server. MySQL is used simultaneously with PHP and the Apache Web Server on a Linux distribution. In particular, PHP specifically relates closely to MySQL to form a cross-platform, closely supporting each other in building applications. The following paragraphs in figure 5 will explain the workflow of Database briefly. (13.)

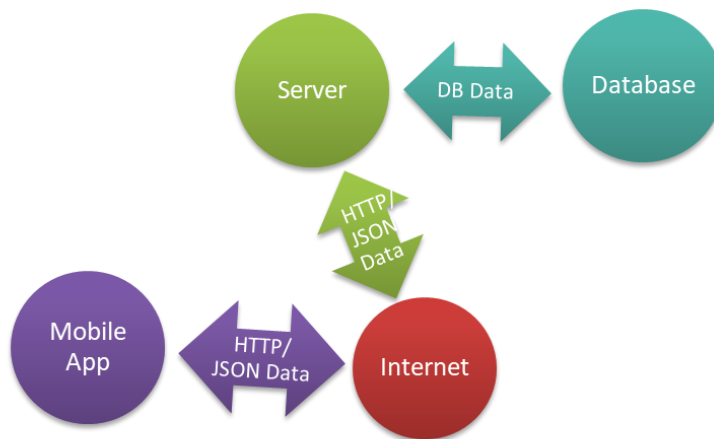


FIGURE 5. MySQL Client-Server Example

When the client requests the server to download files to the Anicare application, the following steps will follow:

- Step 1: The server uses PHP to perform file transfers
- Step 2: PHP transmits the signal to MySQL and then stores the file
- Step 3: MySQL accepts the PHP request and then proceeds.
- Step 4: PHP responds to the Server that MySQL has accepted the request.
The command sent to the Server can print the file contents.

Data safety is always the most important issue when choosing RDBMS software. With the access control system and account management, MySQL set very high security standards. The encryption of login information and authentication from hosts are available. (14.)

4 IMPLEMENTATION

In this section, Anicare proceeded from ideas to reality. Currently the application has come through code, build and test stages. Three completed stages would be analysed elaborately. They are self-experiences, self-perspective, self-obstacles as well as self-research gathered for solution. Functionalities worth mentioning are tracker registration, navigation, notification, auto-fill report form, and sharing. Coming to the development, the workflow would be step by step theme and logo reform, application functions, database creation and server connection.

4.1 UI and Logo Reform

In the scratch and wireframe period, the application had makeshift design based on the default colour variables from Ionic. With the desire to build professionalism and be aware that colour effects can affect the user's emotions as well as impact the meaning of interactive objects in the application, the Anicare colour scheme was evaluated. The colour palette is the combination of Nature and Cheerful-bright tones. The main colours are Pencil Yellow #F0A202, Vermillion #E36414, Tomato #9A031E, Dark Ocean #0F4C5C and Tyrian Purple #5F0F40. A pair of complementary yellow and purple colours could increase the contrast and visual strength of the composition. See the figure 6 below.

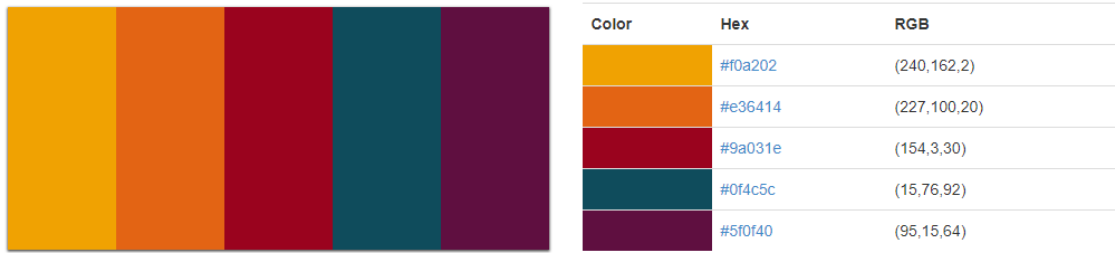


FIGURE 6. Aniccare Color Palette

Meanwhile, the pencil yellow became the dominant colour, which appeared in navbars and main functionality buttons. The reason for choosing yellow was that it comes from the official logo of Aniccare company. The logo revolution turned from the basic plain animation reindeer into the delicate and professional with the blended yellow and black colour scheme. See the figure 7 below.

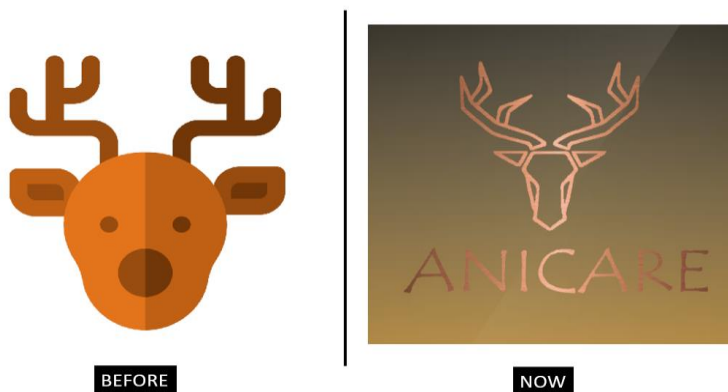


FIGURE 7. Aniccare Logo Reform

The UI design applied the black text among the white background and pencil yellow buttons. The content used the card to separate from the header and distinguish different kinds of content. Each function landed on individual page. This minimized the cognitive load. The human brain has a memory limit of key information called Cognitive load (15.). So, the mobile application needed to minimize

the information displayed, or else it would create a negative feeling when being used for a while. The overwhelmed feeling would reduce the positive emotions forwards the application.

Focusing only on one essential task on one page, bringing out the simplicity and ease of use, were the purpose of Anicare, especially when the target groups who are in the era of receiving modernization from traditional solutions. Using Field Masking to help users enter input more quickly and conveniently, such as if the information field is a "serial number", then the format will be numerical, and limit the number of characters. The layout was redesigned into the simpler and more natural style.

The interface with simple curves, geometry and fab buttons can be more effective and let users to focus on the main functions. Those made areas must be clear and readable and also maintained the consistence in layout. The splash screen is the first image users can see when learning about a mobile application. Here, Splash was designed to be minimalistic with the display of the name and logo of the reindeer with 3 displayed seconds. In addition, it also showed the progress of loading, so that impatient users could know when the application would be presented. Information must be displayed clearly, and friendly, and it must be easy to understand. These are one of the critically important criteria that determine the success of the user experience for UI design. Using jargon and terms within the application context was avoided to apply.

4.2 Features

4.2.1 Map

The core of the application is a map function built from Google Maps data and Anicare data. The map is more minimalist than regular maps. It contains key features, such as reindeer positioning, providing distance measurement, and coordinate information. In addition, it has the specific locations update each time line and geographical position changes when reindeers move for a long time. The path from the user's GPS position to the destination (reindeer) will be highlighted on the map via the red line. The pinpoint point pinned is the location Google Maps icon and the end point is the reindeer icon. The reindeer will have three icon types that represent three states. This is an overview of the Anicare road map.

4.2.2.1 Plugins

With accuracy across the globe, well-known features and various optimization, Google Maps was a wise choice among many factors to consider for an integration plan. Google Maps is the extremely popular application which the vast of majority mobile users have it on their device. This will benefit Anicare to make it easier for users to get familiar with the map interface. Not only that, from the Anicare application, users can also connect and redirect straight to the Google Maps if they wish to use the advanced privilege features from this Google application.

Fortunately, Cordova has released the powerful plugin for embedding native Google Maps, a thin wrapper using libraries of Google Maps Android API, Google Maps SDK for iOS and Google Maps JavaScript API v3. (16.) Although its features are less than Google Maps JS v3, the rendering performance is faster and

smoother in mobile platforms. The special issue while working with this plugin was that Map View was not the HTML element but manipulating as an HTML element. The map module perceives all interaction before the browser, ascertains who ought to get the touch event, at that point to pass it. This means that when you turn on components e.g. the side bar menu, the overlap in the touch event will not appear with the map. The plugin was installed into the application root with the Google API key generated from the Google Cloud website to enable the APIs. Then, updating config.xml with API keys in order that the Ionic project would recognize and access the API.

Cordova provides a plug-in for device positioning, through which the user can perform location processing to obtain GPS location data, such as longitude, latitude, altitude information. It is called Cordova Plugin Geolocation. The geolocation plugin offers returning the position information of the specified coordinate system and supporting the position tracking. (17.)

4.2.2.2 Functionality

This service will hold most of the logic of map function. The table 1 below expresses the vital implemented functions:

TABLE 1. Main functions in Map

Name	Meaning
loadReindeers()	Get reindeer from server
loadMap()	Create a map after the view and platform is ready

getMyLocation()	Get user's current location provided by GPS
initMarkers()	Initialize markers
filter()	Filter data by the status of reindeers
animateCamera()	Center location of the camera view to specific object

Firstly, getting reindeer from the server and calling the map content later were started in **ionViewDidLoad()** life cycle events after the Platform service was ready. The usage of platform service in Ionic here is to make sure that the native code will be resolved successfully. The loading of reindeers from the server function was ran by a service provider which made an HTTP request to an API. This will be discussed in detail in the server part. When fetching the reindeer data, the storage service provided Ionic handled data storing. See the figure 8 below:

```
/**
 * gets reindeer from the server and then calls loadGoogleMaps()
 */
loadReindeer() {
  this.reindeerProvider.getReindeer(this.hash)
    .then(data => {
      this.reindeer = data;
      // console.log(this.reindeer)
      this.storage.set("ReindeerLength", data.length);
      this.loadGoogleMaps();
    });
}
```

FIGURE 8. Getting reindeer data by Service Provider and storing by Ionic Storage

When it was ready, the next function would be fired. For initializing the map, the pre-written protocols in the Native Google Map plugin were utilized. They were nested in a conditional statement to ensure no duplication if the map already existed. Within the function, event listener code for the location button and creating

reindeer markers happened. The benefit of map plugin was MVC status change events. MVC could be written separately. See the figure 9 below:

```
loadMap() {
  if (this.map == null) {
    this.initMap();
    // After the MAP_READY event, you can modify the map.
    this.map.one(GoogleMapsEvent.MAP_READY).then(() => {
      if(this.reindeer.length > 0){
        this.initMarkers();
      }
    });
  } else {
    this.map.animateCamera(
      {
        target: {
          lat: this
            .myLat,
          lng: this
            .myLong
        },
        zoom: 5,
        duration: 1000
      }
    );
  }
  //button for going to your current location provided by GPS
  this.map.on(GoogleMapsEvent.MY_LOCATION_BUTTON_CLICK).subscribe(() => {
    LocationService.getMyLocation(this.options)
      .then((location: MyLocation) => {
        this.myLat =
          location.latLng.lat;
        this.myLong =
          location.latLng.lng;
        this.storage.set('myLat', location.latLng.lat);
        this.storage.set('myLng', location.latLng.lng);
        this.map.animateCamera(
          {
            target: {
              lat: this
                .myLat,
              lng: this
                .myLong
            },
            zoom: 15,
            duration: 1000
          }
        );
      });
  });
  this.updateMapType();
});
} else {
  this.initMarkers();
}
}
```

FIGURE 9. Snapshot of LoadMap() function

It should be noted that while initializing the map, if there are no reindeer markers, the camera view would automatically jump to the current location of use. Otherwise, the view will move to the latest dead reindeer. There are three icons

for three status of reindeers, including alive, dead and reported. Now, when all the objects had been displayed, some plus features were added in. While touching the reindeer icon, the user will activate a polyline from the current location to the selected reindeer. The algorithm runs and gives out the straight distance result via the marker window in order to let the user know how far he or she is from the reindeer. The navigation link to open reindeer's detail would be inserted into the marker window.

4.2.2 Push Notification

There are two kinds of notifications, involving announcing whenever the reindeer is dead and reminding the user to check the reindeer status every two hours when the application is closed completely. The first notification type appeared within the application and the other triggered on the mobile status bar. Both used the OneSignal service.

For a mobile application, the push notification is indispensable. The push notification are messages sent by the server down to the client via a certain cloud message port, and then from this port they will be notified down to user's computer that there is a new notification and it is displayed. Therefore, there must be the internet for a new push notification. Through researches and comparisons between services, the OneSignal was selected, because of the following reasons: (18.)

- A full-featured Ionic SDK
- Unlimited device, unlimited instant messaging and statistics
- Support for API
- Segment user

- A / B test message
- Flexibility like Parse
- Support a large amount of platform, iOS, Android, Unity, Amazon Alexa and so on
- Send Push and User DB services. User DB can easily be viewed
- Send 5000 Push per second feature. Currently speed is still improving (18.)

The server key and the caller ID were set in Firebase. Once the subscription was complete, linking the push notification feature was the next step. In the OneSignal web interface, registering for the application, selecting the platform and the target SDK were done gradually. A reusable template of message with a symbolic icon from Anicare would be created for the sake of the synchronization in the design of the application. OneSignal was initial in the project as a Cordova plugin. Subsequently, the code was modified in project resources.

First, the application ran OneSignal Cordova plugin, which means that the web browser would not be able to implement this plugin, only a real device or an emulator. The code checked if Cordova is available, otherwise a prompting message would be sent as shown in the figure 10:

```

/* ** check if Cordova is available or not
 * that means if the user is using a web browser or real device
 * Because push notifications will not work in the browser
 */

export let isCordovaAvailable = () => {
  if (!(<any>window).cordova) {
    alert('This is a native feature. Please use a device');
    return false;
  }
  return true;
};

```

FIGURE 10. Checking Cordova availability code

Secondly, at the application startup, initialized codes were added into the module and components directory to register the device. The implementing OneSignal lines appeared as shown in the figure 11:

```

if (isCordovaAvailable()) {
  let notificationOpenedCallback = function (jsonData) {
    console.log('notificationOpenedCallback: ' + JSON.stringify(jsonData));
    console.log('Did I receive a notification: ' + JSON.stringify(jsonData));
  };

  // Will show popup dialogs when the devices registers with Apple/Google and with OneSignal.
  // Errors will be shown as popups too if there are any issues.
  window["plugins"].OneSignal
    .startInit(oneSignalAppId, sender_id)
    .handleNotificationOpened(notificationOpenedCallback)
    .endInit();
}

```

FIGURE 11. OneSignal plugin code at application startup

Finally, the push function which informs the reindeer status change from live to death, was attached directly on the home page. There is an extra function inside, for the purpose of checking and measuring the condition of reindeer. It returns the value of reindeer information, so that when a reindeer is dead, the push notification popped up from the notification drawer. On the other hand, when the application is inactive, the user will receive a notification in the status bar to remind

them about the reindeer. When touching the notification, the application becomes active and the user can go straightly into it. See the figure 12 below.

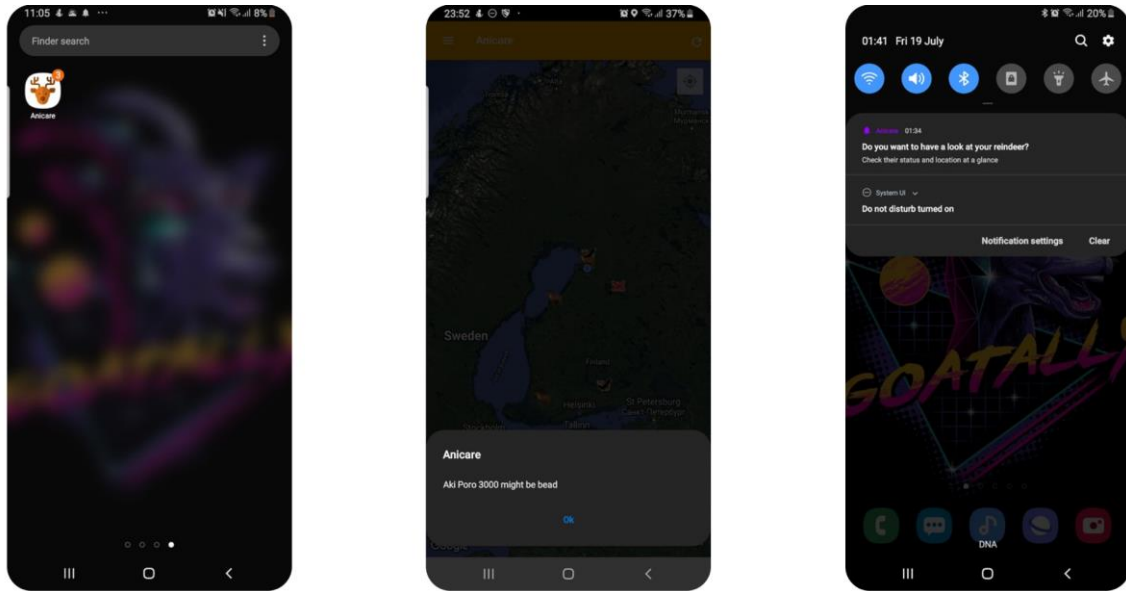
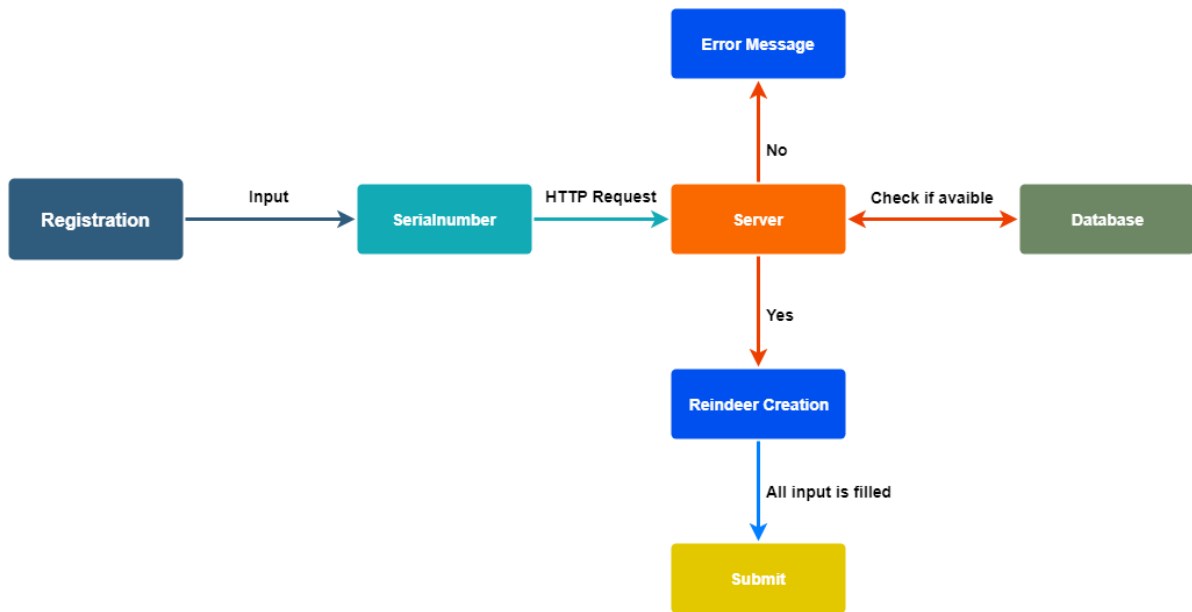


FIGURE 12 Push Notification in real device

4.2.3 Registration

This section will focus on tracker and reindeer registration. The authorization for users as to login and signup will be discussed in the server part. The registration involves two scenarios: tracker clarification and reindeer profile creation. See the table 2 below for the Tracker Clarification process.

TABLE 2. Tracker Clarification process



Tracker Clarification: The user's product code will be verified by comparing the product number information table on the company's database. The product code can be found when the user buy the HealthTag device from the store. It is shown as a serial number. The input bar will be assigned with a function that automatically recognizes whenever the user enters a code, with a length and special characters limitation. After the input, a button for checking the code will be activated. A script written as a Service Provider will execute the work by using the GET in HTTP Request method. The method has to be passed by the API call to send instructions to the server. The return data endpoint is the JSON data which will be transformed into the object later in order to show and compute with the algorithm in codes. The server will check the validation of product number and the authentic association with the user ID (which has been hashed) together. If the product number is valid and it has not been activated elsewhere yet, the reindeer profile form will appear.

Reindeer profile creation: There are name, age, gender and extra info (optional) to be filled. The name input bar will prefill the next available name with the number. The syntax is the user's first name + Poro + numerical order. E.g. last time the user created the fourth reindeer, this time the prefill will be the User's first name Poro 05. Name, age and gender are mandatory, the submission will not success if one of them is missing or incorrect. The API call here is opposite to the above step, it calls for the POST request, so that the server can transfer the data into the database.

The last step of the process, all the entered information includes reindeer profile information, product number and location data where the user registers for the reindeer tracker (automatically retrieved by a GPS data log function in the background) is going to be uploaded to the server. The server will receive and update into the database. The Callback function shows a toast – a subtle notification on the top of the app's content, as a successful message. Then the user can officially activate the HealthTag device and track the reindeer through the app. The added reindeer will appear in the general map and its detail will display on the reindeer page.

4.2.4 Share

This feature was created with the aim of helping users to share troubled reindeer information for other application users located near the reindeer. In this feature, the main user will have the power to grant data access to other application users to the selected reindeer through their email import and verification. Moreover, the main user can easily cut off the rights of those people from the list with a simple remove action.

On the share page, the sharing button appears as a FAB button. The FAB button is a special button, represented by a circle with a share symbol, floating on the surface of the design interface. (8.) By clicking on it, the user triggers the share corresponding action. In addition, the share option can also be accessed straight from the side menu or an action sheet on the living reindeer page.

There are two types of share functionality. First, the main user can give the access to other Anicare user for seeing and identifying his or her reindeer. One reindeer can be shared with multiple users. Therefore, the main user can interact and remove any users who already received the share allowance. Second, the application, also displays who has sent the share allowance to and which reindeer has been shared with the main user.

When working with the API, the author only works with two main API types: GET and POST. GET requests the server to give back the resource and POST requests the server to create a new resource. Both API scripts are in the service provider file. In the share method, an object needs to pass, including the main target shared user email and selected reindeer ID. To return an array list of feasilily reindeer to select, a GET method calls by a function, then filters which are the living reindeers. See the figure 13 below.

```

share(){
  for (let i = 0; i < this.selectedArray.length; i++) {
    this.reindeerProvider.sendShare('{"email":"' + this.email + '", "reindeerId":"' + this.selectedArray[i].reindeerId + '"}').then(data => {
      if (data) {
        this.nav.pop();
        let toast = this.toastCtrl.create({
          message: 'share done',
          duration: 3000,
          position: 'top'
        });
        toast.present();
      }
      else {
        this.nav.pop();
        let toast = this.toastCtrl.create({
          message: 'something wrong',
          duration: 3000,
          position: 'top'
        });
        toast.present();
      }
    });
  }
}

```

FIGURE 13. Snapshot of Share function

Since the Share function was built into the application, this took away the need to rely on any third-party scripts. That indicated the less running code, and fewer HTTP requests. So, it is a huge win for performance. Furthermore, fewer third-party scripts mean that fewer scripts are likely to steal or lose user data, so it is safer and more secure for the user. In consideration of the independence from the third-party, it means one less script which goes beyond of the author's control, in the end, it is more protected for the application. In addition, the share function can only be triggered through user interaction, so even the page load cannot trigger the function on the page.

4.2.5 Report

The application's autofill form feature was born for the purpose of saving time and maximizing convenience. In the future, Anicare is going to negotiate with authorities for API access to the Animal Damages Act. If successful, the application in-app report will be treated as equal as an e-form which people have to go to the site and through the steps to verify the identity to complete. At that time, sending a report will be faster and more accurate in the automated and modernized way.

In the database, a data table was designed to store and record the position information of reindeer through each update of the electronic necklace. After the reindeer stops moving, the application sends the notification to the user. And the system will automatically understand that the final flow of data recorded by the system will become information allowing the user to locate the carcasses. The idea of this report form comes from the claim compensation procedure. People have to fill out a form to report to the authorities about the loss of reindeer farming with a specific reporting form, either through the traditional handwriting form or the e-form.

An HTTP request will be executed and the latest data of the reindeer will be retrieved. Then this data cluster will be smashed into specific pieces. They will be allocated depending on the content shown in the form. The data to be displayed in the application includes: reindeer ID, product number, its name, gender, birth date, date when added and found, place where added and found (latitude, longitude and city name). Information must be processed as well as type conversion, reverse geolocation code for example, before displaying on the UI. There is an extra function to calculate the reindeer age based on the birth date. Single lines were designed whenever filled with information. Each line would display the green bottom border line. When pointing to edit, it turns yellow, which makes it easy to identify the area the user is paying attention to within multiple lines of information in the form. See the figure 14 below.

```

loadDetails(reindeerId: string) {
  this.reindeerProvider.getDetails(reindeerId, 5, this.hash)
    .then(data => {
      this.reportForm.serialnumber = data[0].serialnumber;
      this.reportForm.name = data[0].name;
      this.reportForm.gender = data[0].gender;
      this.reportForm.birthDate = data[0].birthDate.toString();
      this.age = this.calculateAge(data[0].birthDate);
      var currentdate = new Date();
      this.reportForm.dateWhenAdded = data[0].time.toString();
      this.reportForm.dateWhenFound = currentdate.getFullYear() + "-" + currentdate.getMonth() + "-" +
        currentdate.getDay() + " " + currentdate.getHours() + ":" + currentdate.getMinutes()
        + ":" + currentdate.getSeconds();
      this.reportForm.latWhereAdded = data[0].firstLocationLat.toString()
      this.reportForm.longWhereAdded = data[0].firstLocationLong.toString()
      this.nativeGeocoder.reverseGeocode(parseFloat(this.reportForm.latWhereAdded), parseFloat(this.reportForm.longWhereAdded))
        .then((result: NativeGeocoderReverseResult[]) => {
          return this.reportForm.cityWhereAdded = JSON.stringify(result[0].locality)
        })
        .catch((error: any) => console.log(error));
      this.reportForm.latWhereFound = data[0].locations[0].lat.toString()
      this.reportForm.longWhereFound = data[0].locations[0].long.toString()
      this.nativeGeocoder.reverseGeocode(parseFloat(this.reportForm.latWhereFound), parseFloat(this.reportForm.longWhereFound))
        .then((result: NativeGeocoderReverseResult[]) => {
          this.reportForm.cityWhereFound = JSON.stringify(result[0].locality)
        })
        .catch((error: any) => console.log(error));
    })
  });
}

```

FIGURE 14. Snapshot of Auto-fill report form function

4.3 Server

Within this thesis, for security reasons and the copyright of the Anicare company, the instructions with detailed codes are not be discussed and displayed. The author only made a quick introduction, going through the implemented functions to write Restful, and how the testing stage happens. With frameworks, APIs are built quickly, the developer can also use APIs to create complex multi-tier applications. Each API needs to be connected to the server to return data, writing code to do that and formatting the returned data are necessary. Laravel has a stripped-down framework just to make RESTful. It is Lumen.

4.3.1 RESTful API

REST stands for the phrase: Representational State Transfer. REST relies on the verbs of HTTP (GET, POST, PUT, DELETE, PATCH, OPTIONS) to transmit and receive data between the Client and the Server. In particular, data becomes the target of actions in the system, and in this article the data will be reindeer and user information.

The same URL (for example, <http://my-domain-here.com/api/hello/bonny>) is also called an endpoint, but with GET there is an endpoint, with POST we have another endpoint (different here means the way to handle and return data to the client). This makes users more convenient in managing these endpoints. It is easier to modify and change them than using SOAP (Simple Object Access Protocol). (19.) The term API stands for the Application Programming Interface, which is a method for connecting libraries or applications (20). HTTP life revolves around a vicious circle: Request and Response. The client sends the request, the server responds the response as to whether the server can do the client wants. And the API is built on two main components: Request and Response. The figure 15 can explain about that.

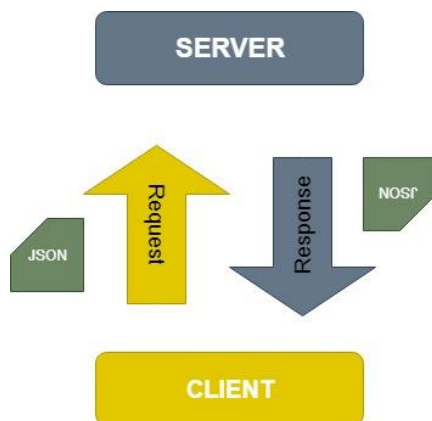


FIGURE 15. REST workflow

REST is a simple rule system that effectively transfers data over HTTP, which is why REST becomes popular and is the best choice for developing APIs. The RESTful API makes it easy for users to use and the main purpose is to enhance the user experience. The client can be the Web Browser, or it can also be a Mobile App, even a different Web Service. The data transmitted is mainly JSON with 2 components: keys and values. The relationship between the Client and the Server is Stateless, meaning that the Server does not need to know who the Client is. (19.)

4.3.2 Local Testing Environment

4.3.2.1 Homestead, Vagrant and Virtual Box

Laravel strives to make the whole PHP development experience interesting, including the development environment. Vagrant offers a simple way to provide and manage virtual machines. Laravel Homestead is the official version. A pre-packaged version of the Vagrant box provides a great development environment without requiring the user to install PHP, HHVM, web server, or any other software on the local workstation.

After VirtualBox / VMWare and Vagrant have been installed, adding the Laravel / homestead box to Vagrant uses the following command in the terminal:

```
vagrant box add laravel/homestead
```

Homestead is installed by simply copying the Homestead folder archive. After copying Homestead, run the **bash init.sh** command is run to create the Homestead.yaml configuration file. File Homestead.yaml will be placed in the hidden `~/ .homestead` folder. To set up the Homestead configuration, providers, configure, the shared folder, Nginx and File Hosts need to be adjusted in Homestead.yaml to suit the project. Specifically, the provider keyword indicates which Vagrant provider will be used (VirtualBox). The keywords folder lists all the folders that will be shared with the Homestead environment. If any file in this folder changes, it will be synchronized between user's workstation and the Homestead environment. Site properties easily map a "domain" to a folder in the Homestead environment. The Homestead.yaml file includes an example configuration. Homestead works as a convenient virtualization environment for the current Laravel project.

After the Homestead.yaml has been edited, the **vagrant up** command is run from the Homestead folder. Vagrant will boot the virtual machine and automatically share the shared folder and Ngix page. To cancel a virtual machine, the **vagrant destroy --force** command is executed. Connecting SSH to the virtual machine uses the **vagrant ssh** command from the Homestead folder. However, to connect SSH to the virtual machine frequently, adding the function described below helps in connecting SSH to the Homestead box quickly. To connect to the database MySQL, 127.0.0.1 and the port 33060 are assigned to the executable.

```
function homestead() {( cd ~/Homestead && vagrant $* )}
```

4.3.2.2 Postman

During the process of deploying the project, the server and the client parts work independently so there are many places where the client has not finished, the

developer cannot wait for the client to finish testing the data but they must test the API with another tool. Now, the testing stage is completely independent of the client. Even when the client is done, if the developer tests on the client and finds errors related to logic and data, it is also necessary to test the API to know exactly whether the error is caused by the server or the client. This way, fixing errors will be faster. (20.)

Postman allows working with APIs, especially REST, which is very helpful for testing. It supports all HTTP methods. Sending requests to servers is manipulated and test data in an easy way. Postman is available on both the Google Chrome Packaged App and on the Google Chrome browser integrated application. Postman is an API development environment that allows to build, test, document, monitor and publish API documentation. In addition, Postman supports RESTful providing the history with auto-suggestions. Another important feature is that the feedback display (Response Viewer) is very intuitive and useful. (20.) See the figure 16 below.

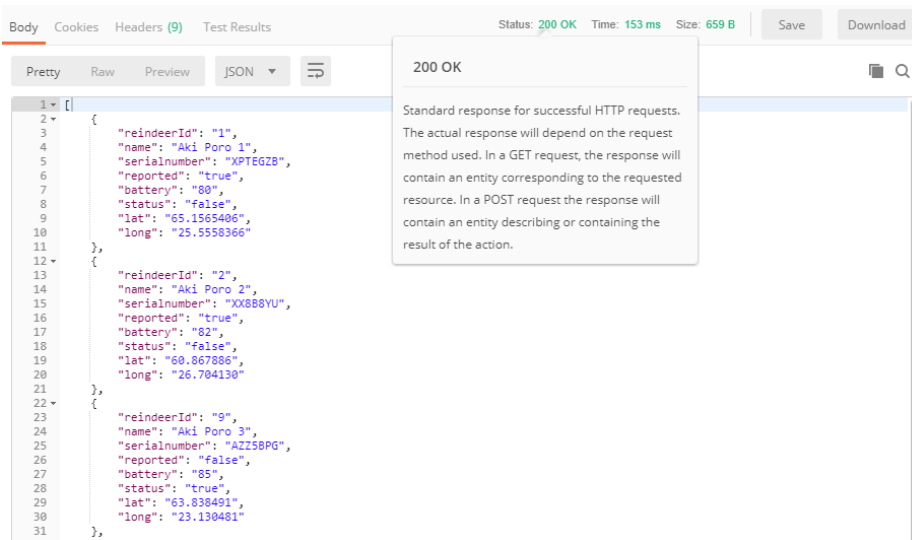


FIGURE 16. Example of Response Viewer in Postman

4.3.3 API

Lumen makes it easy to connect to databases and execute queries. The HTTP Request used with the high frequency in the project are GET and POST. When the server receives an HTTP request, the framework tries to match it to a route registered in the route files. All route files are in the route directory. routes/web.php contains routes for the web interface, while routes/api.php contains routes for APIs. Routers allow declaring routes for resources along with HTTP methods that target resources. When validating stable routes, Postman tool shines for this assignment. A sample route file used to return some predefined data is represented in the figure 17.

```
/**
** routes/api.php
**/

Route::get('reindeers', function () {
    return response(['Reindeer 100', 'Reindeer 200', 'Reindeer 300'], 200);
});

Route::get('reindeers/{reindeer}', function ($reindeerId) {
    return response()->json(['reindeerId' => "{$reindeerId}"], 200);
});

Route::post('reindeers', function() {
    return response()->json([
        'message' => 'Create success'
    ], 201);
});

Route::put('reindeers/{reindeer}', function() {
    return response()->json([
        'message' => 'Update success'
    ], 200);
});

Route::delete('reindeers/{reindeer}', function() {
    return response()->json(null, 204);
});
```

FIGURE 17. Basic Routes for a RESTful service.

```

<?php
/**
 ** Retrieve JSON data from Server using HTTP GET method
 **/

Route::get('/reindeers', function(Request $request){
    $entries = [

        [
            "name" => "Anicare Poro 9283",
            "gender" => "male"
        ],[
            "name" => " Anicare Poro 375",
            "gender" => "female"
        ] ];
    return response()->json($entries, 200);
}

/**
 ** Send JSON data to Server using HTTP POST method
 **/

Route::post('/reindeers', function(Request $request){

    $entries = [
        [
            "name" => "Anicare Poro 9283",
            "gender" => "male"
        ],[
            "name" => " Anicare Poro 375",
            "gender" => "female"
        ] ];

    $reindeer = [
        "name" => $request->input('name'),
        "gender" => $request->input('gender')
    ];

    // Append news animal into current list.
    $entries[] = $reindeer;

    return response()->json($entries, 200);
});

```

FIGURE 18. GET and POST methods sample

In the figure 18, JSON data is sent to the Server using HTTP POST and JSON data from is retrieved the Server using the HTTP GET method events. **\$entries** is an array containing child arrays, reindeer, and the **json()** function of the Response object, which does the task of converting this array to JSON and returns it to the Client. In PHP, switching from Array to JSON uses **json_encode()** and **json_decode()**. However, in Laravel this is pre-packaged, so the developer can apply it directly.

The POST method sends the new server a reindeer profile, the Server receives and appends to the current list and returns this new list to the Client. If writing PHP normally, the **json_decode()** function is called to retrieve the word data in JSON, but in Laravel it integrates that thing in the Request object. So then, getting each value through the corresponding key is like taking the data from the POST form as a normal way via the **input()** function of the Request. In PHP, switching from Array to JSON uses **json_encode()** and **json_decode()**. In Laravel, this has been pre-packaged, so the only thing to do is to use it. After that, Postman can be used to check the returned data or response. See the figure 19 below.

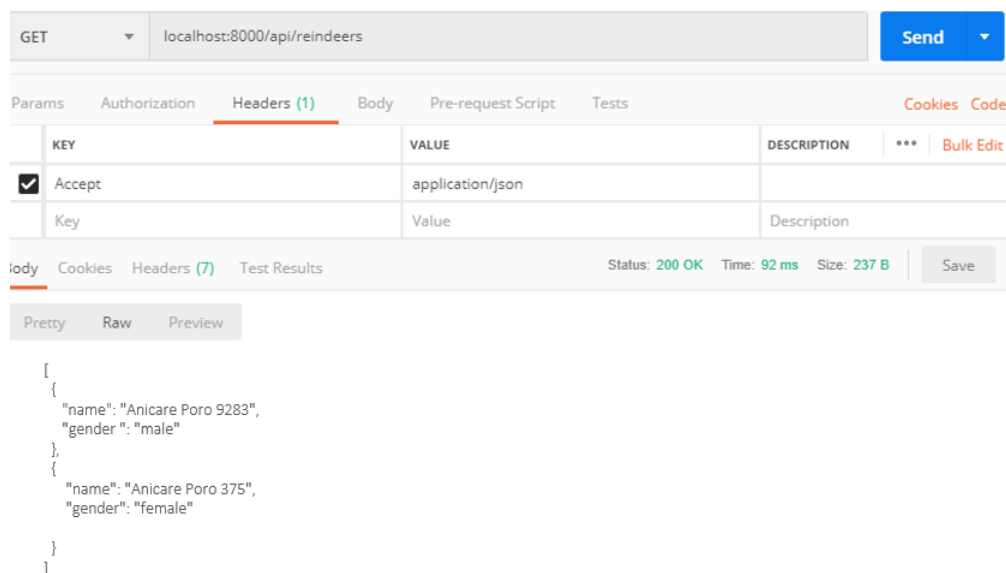


FIGURE 19. Test GET Method in Postman

There are two types of database common queries, Eloquent ORM and Query Builder. Based on the advantage to operate with databases easily, ORM or Eloquent ORM techniques were chosen. In this section, the knowledge field of Eloquent Model (Model) - as part of the MVC model, will be mentioned frequently. These models will directly manipulate the DB, handle major logic and return data

to the controller. Through ORM, each table in the database will be a Model. Interacting with the database actions will be done on the Model, such as querying data or inserting records into tables. The model can be saved anywhere to load automatically according to the composer.js file but usually by default all models will be saved in the app / folder. Models can be created manually by creating files with PHP extensions or automatically via the artisan commands. To make sure that model is mapping correctly with the table in the database, the testing can be done by Tinker Artisan. If true returns, then Eloquent has already mapped the correct model to the table in the database. (21.)

Data queries are written to retrieve data, queries are made through the Model. When retrieving data from the database, to get all the records in 1 table, execute **all()**. Likewise, the find or first function can also get the model instance as **all()**. In addition, when executing queries, in case of having to handle large amounts of results from Eloquent, using chunk will save memory when manipulating large result data sets. This function takes each chunk of Eloquent models, provides them via Closure to handle. The example is showed in the figure 20.

```
<?php
use App\Product;

$products = Product::all();
foreach ($products as $p) {
    echo 'product: ' . $p->name . ' price ' . number_format($p->price) . 'euro' ;
}
}
```

FIGURE 20. Example of get all the record in product table

If the model instance is not found, an exception will be desired to fire out. The findOrFail and firstOrFail functions return the first result of the query, but if there is no result, then ModelNotFoundException will be fired for the user. These lines

of code are placed in try catch to bring the better performance. Since the exception is not captured, then an HTTP response 404 will automatically be sent back to the user. Collections provides a lot of good and short methods to save working time, especially making the API connect to the database because the data returned from the database is available as Collection. Some methods are as simple as list (returns an array with key and value pairs), Filter, Count, Take, Sum and SortBy (21.)

Next, inserting and updating a database will be handled. For the insert section, to create a new record in the table, the model instance has to be created, then the attribute value must be set and the **save()** method must be used. The **save()** function is also used for updating an existing model in the database. The model instance needs to be retrieved first, then the properties in the model instance are changed, and then the **save()** is fired. (22.)

The server system needs to have authentication when performing one or more actions that the system allows. Authenticate, Auth, is a support class for managing login or logout, checking whether the user is logged in or checking the user's login information. For example, with the raw PHP, when checking an account's login, it must initialize the variable, then write the query statement with the condition, then execute and test again. But when using Auth, the user's login information is just inserted and Auth will return the result. See the example in the codes below.

```
$data = ['username'=> $username, 'password' => $password];  
  
if(Auth::attempt($data)) => login
```

When successfully logging into the system, users can access authenticated information anywhere. In the past, when working with features or changing directions to a new page, authentication had to start from beginning again. The Auth has shortened the lines of codes and execution time. The helpers used during the server construction includes: hashing, errors and logging (to handle errors and logs). Hashing is a data encryption function, using the Blowfish algorithm. This helps to encrypt the password.

4.3.4 Database

Similar to the server, for security and copyright reasons, this section will not delve into the official design of this application's database. Instead, the chosen model and contemplation during the database design process will be given out. Database design is the process of transferring real-world data into a logical model. Then, this model will be implemented. This project goes with relational database. Although relational databases have a solid logical and mathematical set, and the scientific rigor of the database design process is also related to aesthetics and intuition, of course, it also includes the designer's subjective tendencies.

To be able to program with PHP and MySQL, the first step to do is to design the database. This is through two steps of creating the database and adding a table to the database to store data. With the phpMyAdmin tool, initialization and installation for the database took place quickly and easily. What to do was the interact and click on the available options in the interface, no need to type every SQL command line as the past. While creating the database, the storage engine was InnoDB which can check for the data integrity and high constraints. This attribute avoids crashing the index. This type of Storage Engine works according to the Row Level Locking mechanism, so when updating (adding, deleting, editing) one

table, only the record which is being manipulated is blocked, other activities on this table still occur. In the future, the Anicare website will also link directly to this database, so the transaction event will also be supported by InnoDB. For Charset and Collation, the utf8 charset with utf8_unicode_ci collation was chosen.

The database is standardized in the form of 3NF because it is the best model to represent data entities and balance query performance and to insert, update, and delete records. Further, it reduces redundancy. Usually, the root username of MySQL is used directly to connect to the database, but for security reasons, creating a separate username for each database is carried out.

5 VISUAL VIEW

The application has been fully functional. By acquiring comments and selecting the appropriate plan to deploy to the application, the UI version was finally fixed, by correction the problems discovered during the testing phase. Feedback is always the key to development the usability of this application. Finally, the application development and design meetings with the company determine valuable changes to the application. There are some of the screen captures of the application. See the figure 21 for Home Page, figure 22 for Reindeer pages, figure 23 for Share, Register and Report Page.

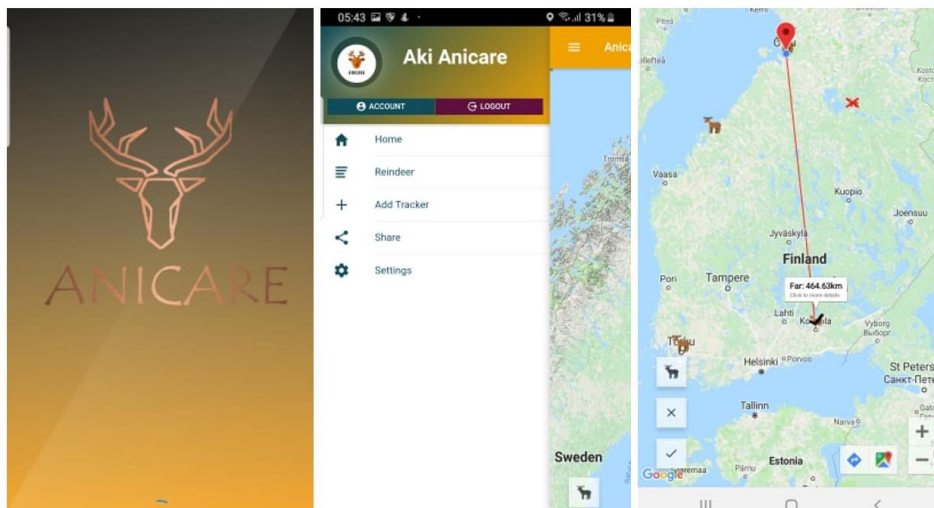


FIGURE 21. Home Page

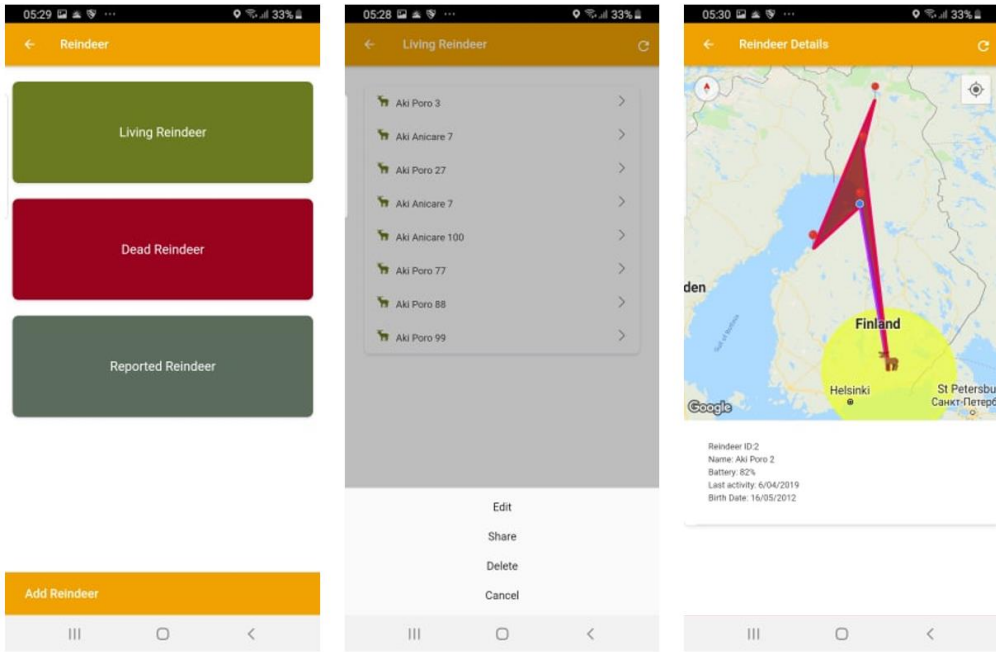


FIGURE 22. Reindeer Pages

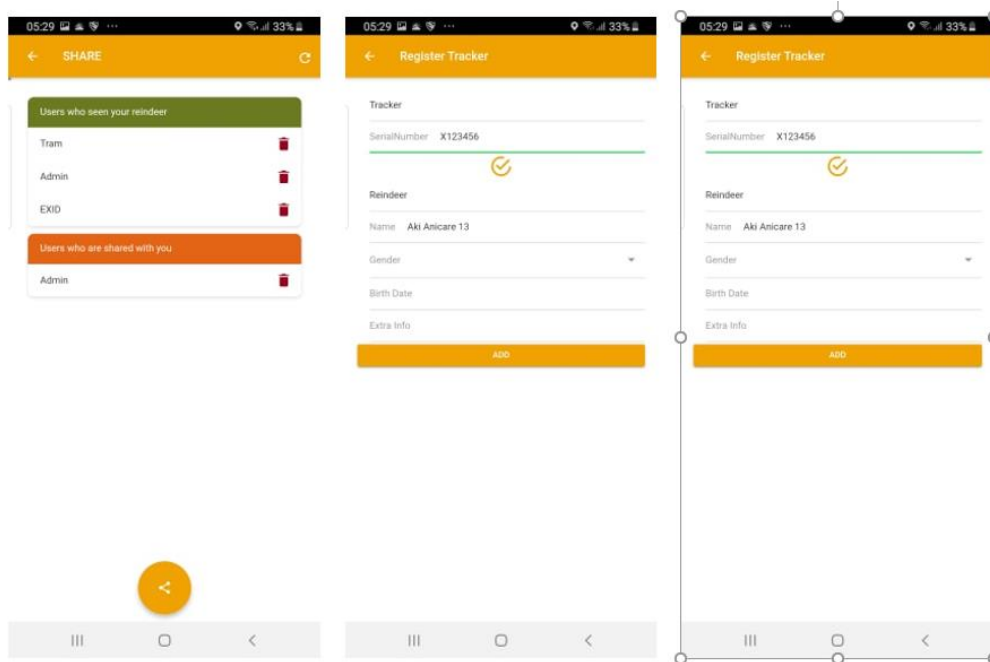


FIGURE 23. Share, Register and Report Page

6 CONCLUSION

The purpose of this thesis was to create a mobile application using a hybrid mobile development framework. This application serves as the client application for monitoring the service provided by Anicare Company. The objective of this project was to propose a mobile application to facilitate the track and search for reindeers, which incorporates a geographical component to improve the visualization of the information and, in turn, allows the main user to selectively notify users close to the place of occurrence of an event, expediting and optimizing the diffusion process by the sharing feature.

At the end of the project, all major start requirements were met, and mobile application development processes were explicated in a structured order. In the process of designing and implementing, the application is constantly being modified and optimized. This makes the final result much better than it was originally expected. This is mainly determined by the company's pursuit of perfect personality. The structural simplicity is considered a favourable aspect, not only allowing the user to use the application from anywhere through their smartphone, but also in stricter terms to the architecture, it is expected that if necessary, the components of the server and database can be migrated quickly and easily.

The project developed serves as an example to give rise to future works that involve geographic information in a practical way to solve everyday problems. The added value generated by the geographic data is significant and it is worth including it in the developments carried out. The production process of this graduation thesis is a process of re-learning and improving. All of this not only enriches the knowledge structure, expands the knowledge, enhances the potential

for analysis and problem solving, but also enhances the potential of self-study and lays the foundation for the future work.

7 REFERENCES

1. Suurpedot-sivusto, 2018. Poronhoito Ja Suurpedot. Date of retrieval: 04.06.2019. <http://www.suurpedot.fi>.
2. Kojola Ilpo, Heikkinen, Samuli, Kaartinen, Salla, 2018. Suurpetojen vaikutus poronhoitoon: Makera-hankkeen loppuraportti, Date of retrieval 01.07.2018, <http://jukuri.luke.fi/handle/10024/543254>.
3. Bharadwaj Ashesh. December 07, 2018. 10 things every Android app developer should know about Architecture, Date of retrieval 02.07.2018, <https://proandroiddev.com/10-things-every-android-app-developer-should-know-about-architecture-75ffc37df556>.
4. Ionic Framework. 2019. Ionic documentation. Date of retrieval: 11.07.2019. <https://ionicframework.com/docs>.
5. Pisuwala Ubaid. Everything you need to know about mobile app architecture. Date of retrieval: 18.07. <https://www.peerbits.com/blog/all-about-app-architecture-for-efficient-mobile-app-development.html>.
6. Lumen Framework. 2018. Lumen documentation. Date of retrieval: 19.07.2019. <https://lumen.laravel.com/docs/5.8>.
7. Dutton Jude. 2018. It's Gonna Be Okay: My Intro to MySQL. Date of retrieval: 19.07.2017. <https://medium.com/@judedutton/its-gonna-be-okay-my-intro-to-mysql-1677c3f4901f>.

8. INKONIQ. 2017. Top 7 Reasons to Pick IONIC Framework for Mobile Apps. Date of retrieval: 18.07.2017. <https://medium.com/inkoniq-blog/top-7-reasons-to-pick-ionic-framework-for-mobile-apps-30169275fb58>.
9. Overview. 2018. Cordova documentation. Date of retrieval: 19.07.2017. <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>.
10. Angular. 2018. Wikipedia. Date of retrieval: 17.06.2019. [https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework)).
11. Lasn Indrek. 2018. TypeScript — JavaScript With Superpowers. Date of retrieval: 27.06.2019. <https://medium.com/better-programming/typescript-javascript-with-super-powers-a333b0fcabc9>.
12. PoweredLocal. 2016 Developing a simple API gateway in PHP and Lumen. Date of retrieval: 11.06.2017. <https://medium.com/@poweredlocal/developing-a-simple-api-gateway-in-php-and-lumen-f84756cce043>.
13. Nitesh Mishra. 2016. How to create MySQL Database for your Backend. Date of retrieval: 07.07.2019. <https://medium.com/configure-it/how-to-mysql-database-for-your-backend-9befa599a10>.
14. Linode. 2017. 5 Essential Steps to Hardening Your MySQL Database. Date of retrieval: 19.07.2019. <https://medium.com/linode-cube/5-essential-steps-to-hardening-your-mysql-database-591e477bbbd7>.
15. Arhipova Alina. 2019. Psychology in Design. Principles Helping to Understand Users. Date of retrieval: 11.07.2019. <https://tubikstudio.com/psychology-in-design-principles-helping-to-understand-users/>.

16. Measure the performance of a Cordova app. 2015. Microsoft Documentation. Date of retrieval: 11.07.2019. <https://docs.microsoft.com/en-us/visualstudio/cross-platform/tools-for-cordova/tips-workarounds/measuring-performance?view=toolsforcordova-2017>.
17. Ionic 3 | Get Device Latitude, Longitude and Address using Geolocation and Native Geocoder Services in Ionic 3 Native Application. 2019. FreakyJolly documentation. Date of retrieval: 14.07.2019 <https://www.freakyjolly.com/ionic-3-get-device-latitude-longitude-and-address-using-geolocation-and-native-geocoder-services-in-ionic-3-native-application/>.
18. Sapkota Sanjeeb. 2018. Integrate OneSignal Push Notifications Into An Ionic 3 Application. Date of retrieval: 27.06.2019. <https://www.c-sharp-corner.com/blogs/integrate-onesignal-push-notifications-into-an-ionic-3-application>.
19. Representational state transfer. 2019. Wikipedia, the free encyclopedia. Date of retrieval: 11.07.2019. https://en.wikipedia.org/wiki/Representational_state_transfer.
20. Using Postman to access the API. 2017. Online Article. Date of retrieval: 15.07.2019. <https://www.jomres.net/manual/developers-guide-2/63-jomres-api/341-using-postman-to-access-the-api>.
21. Dayle Rees. 2016. Code Bright: Eloquent ORM. Date of retrieval: 15.07.2019. <https://daylerees.com/codebright-eloquent-orm/>.
22. Mhetre Ramesh. 2019. Using Laravel's Eloquent ORM Efficiently. Date of retrieval: 20.07.2019. <https://fullstackworld.com/post/using-laravels-eloquent-orm-efficiently>.