

Duck Tracker 2.0

Software Requirements Specification

Group 6 -

**Nick Henderson(nh), Wentian Lin(wl), Christopher Wycoff(cw), Hongtao Yang(hy),
Dennis Zipprich(dz)**

6/1/2020

Table of Contents

- 1. SRS Revision History**
- 2. The Concept of Operations**
 - 2.1 Current System or Situation**
 - 2.2 Justification for a New System**
 - 2.3 Operational Features of the Proposed System**
 - 2.4 User Classes**
 - 2.5 Modes of Operation**
 - 2.6 Operational Scenarios**
- 3. Specific Requirements**
 - 3.1 External Interfaces**
 - 3.2 Functions**
 - 3.3 Useability Requirements**
 - 3.4 Performance Requirements**
 - 3.5 Software System Attributes**
- 4. References**
- 5. Acknowledgements**

1. SRS Revision History

This section will provide a change for this document. Changes made should be reviewed by all team members. If any change is made it will be documented here with the following information:

Date - This is the specific day that the change was made to this document.

Author - This is the person responsible for making that change in the form of initials.

Description - This is a brief overview of what changes are made in the entry.

Date	Author	Description
5/14/2020	dz,cw	Created initial document.

5/15/2020	cw	Edited and revised document.
6/1/2020	dz,cw,nh	Edited and revised document.

2. The Concept of Operations (ConOps)

Duck Tracker 2.0 will be an application that gathers real world geolocation data from users.

2.1 Current System or Situation

Data regarding human location and movement is valuable. Geolocation may be defined as “the process of determining the location of an electronic device, as a computer, cell phone, satellite, etc.” (Dictionary.com). Many systems that collect geolocation data often do so without clear consent from the users (Dans). Geolocation systems often are complicated, opaque, and operated by massive tech companies (Google Policies). Duck Tracker 2.0 seeks to be a simple and maintainable system where users may anonymously provide geolocation data.

2.2 Justification for a New System

One of the most significant events of 2020 and perhaps the decade, is the global pandemic known as the Novel Coronavirus or “COVID-19.” Novel geolocation services may ease the difficulty of social distancing. For example, knowing the most popular times of a hiking trail might allow for people to make higher quality decisions about when to visit.

It was in our teams interest to recreate Project 1, which aids in data gathering for social distancing purposes, as we found multiple items worth reworking and reimplementing to make modules more transparent and easy to use. Our team had black box dependencies and the installation process was complicated with much room for error. Therefore we chose to create Project 2.0 with more stringent requirements, outlined below.

The Duck Tracker 2.0 system aims to:

- Collect geolocation data from users’ mobile devices. This data can theoretically be used to create additional systems to increase the quality of people's lives.
- Be easier to install than Duck Tracker 1.0, by providing an installation script for the user and detailed install instructions.
- Allow for maintainable and long-lasting modules by eliminating “black box” dependencies.
- Provide more robust documentation alongside the system.

2.3 Operational Features of the Proposed System

Since this will be a rework of Project 1, the operation features will be mostly the same as Project 1. The requirements of the system dictate two major operational features: data collection and data aggregation.

The geolocation data must be gathered at frequent intervals. Given the ubiquity of smartphones, Duck Tracker 2.0 will leverage geolocation services built in to smartphones in order to accomplish this requirement.

Once data has been collected, it must be sent to persistent storage and be formatted in a useful manner. A database technology can assist in achieving this requirement. There specific tab delimited formatting requirement that the data ultimately must be delivered in.

The format is as follows:

User I.D.\tDate\tTime\tLatitude\tLongitude\tTime at Location\n

2.4 User Classes

The data set gathered by Duck Tracker 2.0 may be useful to other developers for many uses, as it is an entirely real-world data set. This section will give a few examples of different user classes that can utilize Duck Tracker 2.0's data.

There are two main user classes that our team is considering:

1. A phone application user. This could be essentially anyone willing to provide geolocation data to the Duck Tracker 2.0 system.
2. A system installer and maintainer. This user will install and deploy the system given only the source code and documentation.

User classes are not limited to the two presented, and can include developers of any auxiliary application examining mobility in the Eugene area.

2.5 Modes of Operation

Duck Tracker 2.0 will have one primary mode of operation: data collection mode. This mode will be minimally invasive to the user as it will collect data in the background and send small

packets of data to a database for secure storage.

This data can be viewed in Duck Tracker 2.0's secondary mode of operation: data aggregation mode. Here, users will be able to view the data by visiting a link to download a copy of the database.

2.6 Operational Scenarios (Also Known as “Use Cases”)

Duck Tracker 2.0 will be designed to be minimal, and will be able to run intermittently on a user's phone to collect geolocation data. On many phones the application will allow mostly hands-off use. See README for optimal usage to accomodate for most phones/settings.

As the system operates in the background it will update position every 5 minutes via trilateration (Wilson). As the mobile device updates position, it will send a small geolocation data packet to a MySQL database. The user will be able to access this geolocation data via a LAMP (Linux, Apache, MySQL, PHP) server web application. This will be simple deliverable measurable data as per system requirement six.

Use Case: Location tracking

Description: This use case describes how a user would use their mobile device to track and store geolocation data.

Actors: Mobile device user.

Preconditions:

1. The user has access to an Android mobile device that they will be able to download software onto.
2. Users have access to cellular data plans that enables them to access the Internet and geolocation services.
3. The mobile device is configured to allow location services and to run the application in the background.

Steps to Complete the task:

1. User opens Duck Tracker 2.0 application on device.
2. User may now update location manually or application will update every five minutes.

Postconditions:

1. Duck Tracker 2.0 will now send data every five minutes so long as they have the application open (even in the background).

Use Case: Viewing geolocation data

Description: This use case describes how a user would view all the data stored on the

MySQL database.

Actors: Duck Tracker 2.0 user.

Preconditions:

1. Users have access to ix-dev servers at the University of Oregon.
2. Users have gone through the installation steps outlined in Installation Instructions to set up the database to receive data.

Steps to Complete the task:

1. Navigate to
https://ix.cs.uoregon.edu/~<YOURUSERNAME>/duck_tracker_server/duck_tracker_gate.php
2. Open the file that is automatically downloaded.

Postconditions:

1. The user may now view all the data that has been sent over the database.

3. Specific Requirements

The following subsections will outline the different functional and nonfunctional requirements of Duck Tracker 2.0. For a full set of project requirements please see <https://classes.cs.uoregon.edu/20S/cis422/P1/> for more information. Our team will build upon these requirements through the lens of simplicity and maintainability.

The primary goal of Duck Tracker 2.0 will be to reimplement Duck Tracker 1.0. Duck Tracker will be improved to be easier to install, maintain and test. This will allow major stakeholders to have more control over Duck Tracker 2.0. The technologies used in Duck Tracker 2.0 are chosen primarily for their simplicity and portability. Some ‘black box’ dependencies will be shed from Duck Tracker 1.0.

Duck Tracker 2.0 requirements:

1. Simple and transparent dependencies
 - Reduce or remove the degree of reliance on Google services and API key
 - If an API key is determined to be necessary, make it easy for a user to know what if anything to modify for a fresh installation.
 - This will be achieved by implementing proven open source software that is easily customizable to meet the needs of our system and replace existing dependencies.
2. Ease of installation requirements - script or DIY
 - Cater to command line usage
 - Inline comments and documentation on how to link dependencies.

- Use a pass/fail install script to install needed dependencies. A pass is the system installs with no problems following, a fail is the system did not install for any reason.

3. Error reporting system

- Basic error reporting system to aid maintainers and testers of quality of system
- This requirement was not implemented in full due to limited resources.

4. Notification of connectivity

- Mobile app interface will show state of connectivity
- All other user interfaces will display connectivity or provide a method with which to verify connectivity.
- This will be implemented as a simple status indicator with a pass/fail criterion.
Pass - the status will be green. Fail - the status will be red.

5. User privacy and anonymity

- Users data will be anonymous and the home obscuring functionality will be simplified
- App will use a secure pipeline (SSH) for transmission to database
- User data stored in the database will be anonymous, i.e. Duck Tracker will not keep any identifying information, only geolocation data.

3.1 External Interfaces (Inputs and Outputs)

This section will outline some requirements that are needed by external devices. These requirements mostly depend on users' hardware available.

External requirements:

- The major input to the system will be latitude and longitude
- Users must be able to submit geolocation data, via a smartphone, to use Duck Tracker 2.0.
- Users must be able to access web applications from any modern web browser.
- Users' mobile devices must have a SIM card installed for accurate device geolocation tracking.
- Duck Tracker 2.0 must be able to triangulate position with users' mobile service carrier or Internet signal.

The primary means of using Duck Tracker 2.0 is via mobile device, therefore the only input required is user accessing mobile application from a mobile device. The output will be the collected geolocation data sent to Duck Tracker 2.0's database, represented as a .txt file.

3.2 Functions

This section will detail required actions in the software that must take place in order for users to use Duck Tracker 2.0.

(needs more)

Transmits location data given that the user has correctly configured their mobile device to utilize geolocation services.

Post request must receive status OK (200) response for the server. This will be indicated via a status image and text in the mobile application.

If the mobile device connection to the database fails, it will be indicated as an implemented binary status indicator on mobile application.

Output of formatted data will be accessible via the duck_track_gate.php on server. Accessing this file with no request parameters will result in a download of formatted data from the database.

The post request parameters sent from the app to the PHP file are strings with the following identifiers:

- ID
- time
- latitude
- longitude
- tal (time at location)
- date

3.3 Usability Requirements

This section will define usability requirements for software systems implemented in Duck Tracker 2.0.

Users must have access to at least one Internet connected and GPS-enabled smartphone device that is able to determine location within a reasonable degree of accuracy. Inaccurate devices will still transmit location data, but the aggregate dataset may be affected by poor quality of GPS data.

The user's device must be kept charged and on. Though the app does not need to be open in order for the user's location to be logged, it must be running in the background.

The web app must run on major web browsers such as Google Chrome, Apple Safari, Microsoft Edge, and Brave web browser.

3.4 Performance Requirements

The user's device must be able to acquire the date, time, and geolocation data in less than one second. This is essential to keep the interval between data collection synchronized.

Duck Tracker 2.0 must also be required to automatically post to a database component at least every five minutes. This data posting to the database should happen in under five seconds.

In addition to keeping data on a server, our team will do a daily local backup of our database for increased redundancy.

End users should also be able to parse data collected in a specified tab delimited format. This data should be easily accessible, potentially via a URL.

3.5 Software System Attributes

The following software attributes are required for Duck Tracker 2.0:

- All software modules must be reliable and work as described.
- Software modules must not be black box modules, i.e. our team must have full control and understanding of every aspect of the software.
- Installation and configuration of software modules will be streamlined and a script will provide functionality to setup modules automatically.

4. References

Dans, Enrique. 2018. "They Wouldn't Sell Your Geolocation Data Without Your Permission...", Forbes

<https://www.forbes.com/sites/enriquedans/2018/12/11/they-wouldnt-sell-your-geolocation-data-without-your-permission/#282905c77f08>

Dictionary.com. <https://www.dictionary.com/browse/geolocation>

Google Policies. <https://policies.google.com/technologies/location-data>

Vliet, Hans. (2008). Software Engineering: Principles and Practice, 3rd edition, John Wiley & Sons.

Wilson, Tracy. "How GPS Phones Work" 24 October 2005. HowStuffWorks.com.
<https://electronics.howstuffworks.com/gps-phone.htm> 14 April 2020

5. Acknowledgements

SRS template - 5-6-2020 - v1.02 provided by A. Hornof, University of Oregon, CIS422,
<https://classes.cs.uoregon.edu/20S/cis422/Handouts/Template-SRS.pdf>