

Duck Tracker
Software Requirements Specification
Dennis Zipprich (dbz) - 4-10-2020 - v2.0

Table of Contents

1. SRS Revision History
2. The Concept of Operations (ConOps)
 - 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 (Inputs and Outputs)
 - 3.2 Functions
 - 3.3 Useability Requirements
 - 3.4 Performance Requirements
 - 3.5 Software System Attributes
4. References
5. Acknowledgements

1. SRS Revision History

Date	Author	Description
4-10-2020	dbz	Created the initial document outline.
4-15-2020	nbh	Made typo and other minor corrections to sections 2 and 3
4-23-2020	dbz	Removed diagrams, formatted file for .txt format for git hub transfer, small changes to section 3.3, 3.4
4-25-2020	dbz	made corrections to documents based off of instructor feedback
4-26-2020	clw	proofread and edited

2. The Concept of Operations (ConOps)

2.1 Current Situation

Many different systems use geolocation data. Such a system might assist in managing traffic or to notify consumers about the busiest times at a store. A major obstacle to implement one of these systems is obtaining said geolocation data. Duck Tracker will provide specific functionality

that will allow users to track, store, and view aggregate users geolocation data.

Duck Tracker can provide stakeholders with real world collected data, from many different users anonymously. This real world data can be implemented in many different ways, however it is our team's only goal to collect this raw data and provide deliverable and parsable format for future use.

2.2 Justification for a New System

The major story of this year (2020), and perhaps this decade, is the global pandemic known as the novel coronavirus or "COVID-19." Social distancing remains the primary tool utilized by human societies in the fight against COVID-19. Novel (pun intended) 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.

The Duck Tracker system aims to collect geolocation data from users' mobile devices. This data can be theoretically used to create additional systems to increase the quality of people's lives.

Though there are likely established large companies developing (or have developed) similar systems, "group 6" acknowledges the value of working on a current and dire issue. Even if this system is never used, the knowledge and experience regarding software methodology on a real problem, will be invaluable for the members of "group 6."

2.3 Operational Features of the Proposed System

A Duck Tracker user can be anybody with a mobile Android device. Users can also download the "APK" file per install/README instructions. Geolocation permissions are requested from a user in an attempt to ensure a user consents to their data being collected.

A more technical user hoping to use Duck Tracker data may access the data via being granted direct database access. Additionally public access can be made available via a web portal (javascript/html) or a python program. (see README and additional documentation for details)

There will be two major operational features to Duck Tracker: data collection and data aggregation.

Data collection will begin once the user downloads and installs the app on their mobile Android device. Duck Tracker is meant to be mostly hands off and will work in the background of their mobile device after they've opened the app for the first time. Users may also update their location manually. Unfortunately, depending on the type of Android device, it may be necessary to keep the program in the foreground to ensure data is collected correctly. In a future update to the system this may be rectified.

Duck Tracker will depend on the users mobile service provider as most smartphones must use a data plan to access geolocation data via trilateration (Wilson). Duck Tracker will not be able to update location if the mobile device is out of range of a mobile service provider coverage area.

Data aggregation happens on the users mobile android device but on any modern day internet web browser. The user must navigate Duck Tracker's website and view data gathered by all users anonymously. Users may also download data in a tab delimited format. The specific 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 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's data.

The primary user class is the University of Oregon CIS 422 students who will use this data to possibly create further systems which depend on a real data set.

The secondary user class is the University of Oregon CIS 422 instructors that can use this data to assign future projects to students.

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

2.5 Modes of Operation

Duck Tracker 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 a small set of data to a database for secure storage.

While in this data collection mode, the application can be used to aid in movement tracking and recording locations visited. This data can also be viewed in Duck Tracker's secondary mode of operation, data aggregation mode. Here, users will be able to view the total data set in real time via web application or python program.

2.6 Operational Scenarios

Duck Tracker will be designed to be minimal, and will be able to run intermittently on a user's phone. 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 it updates position, it will send a small data packet of geolocation data via the mobile user's service provider data plan to a Google Firebase database. The user will be able to access this geolocation data via a Firebase-hosted web application. This will be simple deliverable measurable data as per system requirement six.

3. Specific Requirements

Duck Tracker will have a specific set of requirements, however is it designed to be accessible to many different users. The following subsections will outline the different functional and nonfunctional requirements of Duck Tracker. For a full set of project requirements please see <https://classes.cs.uoregon.edu/20S/cis422/P1/> for more information.

3.1 External Interfaces (Inputs and Outputs)

Duck Tracker is an Android based application, and thus requires an Android operating system to run the software. Duck Tracker sends the geolocation data and information about the measurement to the database for storage. Below are Duck Tracker software elements that require external inputs needed from the user to run the mobile application.

Android app component (1) - Privacy/permissions

Inputs - gain permission for geolocation from user

Outputs - na

Description - This software component of Duck Tracker enables the mobile application to access built in technology from the phone to acquire geolocation data. Mobile devices may require further permission to run in the background and the application was designed to do.

Android app component (2) - Hardware Identification

Inputs- first time user of application

Outputs- unique user identification code specific to device

Description - This software component of Duck Tracker will acquire a unique user identification code generated using the user's hardware identification number. This will ensure that data is aggregated correctly and that a user's location data is properly grouped by a single anonymous ID. This step requires no action of the user besides starting the app for the first time.

3.2 Functions

This subsection will include functions that Duck Tracker will employ once there are no more external inputs needed from the user. As Duck Tracker is meant to run in the background, these functions are meant to be lightweight and non-taxing to the user's mobile device performance. Each software component will have its own task as outlined below.

Android app component (3) - Device Trilateration

Inputs - access user location via network service provider and Google Play services-maps api

Outputs - data set sent to server, update map displayed on android view, drop user pin on map

Description - This software component will utilize the mobile device's built in geolocation hardware and get a specific latitude and longitude, with an accuracy down to 4 decimal places, and other information such as time, date, time spent at location. Then this software component will then send this data packet to android component (4), outlined below. Finally this software component will drop on an in-screen map to show location. This process will be done via Google play services-maps api for simply data transfer

Android app component (4) - Data Transfer Module

Inputs - latitude, longitude, date, time from Android app component (3)

Outputs - post lat long to firebase real time database via Google (firebase-database, firebase-analytics api)

Description - The Data Transfer Module will take the data received from Android app component (3), outlined above, and post the data to our Google Firebase database for safe storage in real time. This is meant to be light weight only posting what is required in the project technical and system specifications.

Firebase - database component (1) - Receive Data

Inputs - database post from android app component (4)

Outputs - update to realtime database with applicable data set components as per project specification system requirement 6.

Description - This component will receive the database post from the Android app component (4), outlined above, and update to a realtime database. This will be done via Google Firebase platform for ease of use and scalability.

Web app component (1) - Data Review

Inputs - database post from android app component (4)

Outputs - update to realtime database with applicable data set components as per project spec sys. req's 6.

Description - this will be the final component of Duck Tracker, a web application, that allows the user to review data being stored in the database.

Web app - generate file

Inputs - JSON

Outputs - Database .txt file

Description - This component is used to generate a .txt file of the information in the database in the format described by the system requirements.

3.3 Useability Requirements

Users must have access to at least one internet connected Android OS based device, that has a runnable .apk file to run Duck Tracker. This is essential for the user to employ Duck Tracker. The user may have more than one device, but this data will be recorded under a different device ID unless the user configures their device immediately.

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

The web app must run on major web browsers such as Google Chrome, Microsoft Edge, Apple's Safari, 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 must also be required to automatically post to the Firebase database component every five minutes, as required by the project specification. This data posting to the database should happen in under five seconds.

Database backup should happen daily using a local machine. In addition to the extremely reliable nature of the Google Firebase platform, 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 format from our team's web application. Users can use the application designed to view the database to also download a copy of the database in .txt format.

3.5 Software System Attributes

Required attributes:

1. Geolocation data collect- Achieved (via android app)
2. Storage of geolocation data - Achieved (via firebase database)
3. Backups - Achieved (via python script, and via webportal)
4. Viewing of data - Achieved (via python script, and via webportal)
5. Masking of home location - Achieved (via anonymous ID android app component, scrambling of most common location python app component)
6. Tab delimited format - Achieved (via python script)
7. TSI - Achieved (via python script, and via Android app design)

Nonfunctional/Not absolutely required functionality that can be implement for future use can also include:

- Multiple user view and activities on android app
- Additional security for user authentication
- Cross platform compatibility
- Ability to report problems with application
- User support via problem reporting and backend dev portal
- Added bluetooth functionality to mimic functionality for cooperation of outside systems

4. References

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

Uses of Class java.util.UUID. (n.d.). Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/util/class-use/UUID.html>

Best practices for working with Android identifiers. (2020, March 19)
Retrieved from <https://developer.android.com/training/articles/user-data-ids#best-practices-android-identifiers>

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