Potty-Poll Software Plan

1.0 Scope

Potty-Poll is an Android application used for finding nearby public restrooms as well as sharing reviews and comments on those restrooms. The motivation for this application was the common problem of trying to find the closest public restroom during an "emergency situation." This idea expanded passed just a restroom locator to a reviewing app to highlight an area's cleanest and most pleasant restroom experiences. At the same time, it should warn users of any unpleasant restroom experiences they may unexpectedly find themselves in. This app will allow users to find the restroom that fits their needs without needing to check themselves and to share their opinions on restrooms through a 5-star review system. The hope for this application is to improve our users' day-to-day lives by improving their average restroom experience.

1.1 Functions

This application will allow users to select locations that have public restrooms on a map and view relevant information about those restrooms. Searching for restrooms will be done using nearby locations, taking advantages of location services, or using a user-inputted location. The search radius and search filters will also be adjustable by the user before the search begins. Users will be able to view information about these restrooms including size, hours, and cleanliness and overall ratings. Data for restrooms will be provided by the users through reviews that can be submitted through a form on the app. Users will have the option to submit both a text review and a star review. The form will include sections to input information as well as optional comments. Users will also be able to suggest new restroom locations through a similar form. These suggested locations should be restrooms that currently exist, but are not currently in the database.

Possible functions that may be added to the application include a login feature that will allow users to keep track and edit their rating and comments they have made in the past. A flagging function may also be added to allow users to report mistakes and omissions in the data.

1.2 Performance

The performance of this application will rely heavily on the performance of the server handling the requests. The server will have to run an algorithm that determines what data is sent to each user, so the amount of requests the server can handle and how fast it can produce responses will determine the application's performance as this will likely be the bottleneck. Displaying the data once it is received should be relatively quick. Another factor

that may impact performance is the local density of data points in a specific area. The more restrooms within a specified area means that there is more data that needs to be sent from the server and stored locally in memory. Performance will also be affected by the distance and regions included within the map. If the map is large and there are many locations, this may negatively affect the performance. However, keeping the locations within a small range should allow the application to run fairly quickly. Filtered searches may also help increase performance by reducing the amount of data that is needed to be sent by the server. Settings such as search radius and filters may be changed by the user locally on their device and will affect the server requests made by user devices.

1.3 Limitations

Data is gathered from users' subjective reviews, so the data may be inaccurate or missing. There may not be an automated system to check for errors or the validity of information provided. Error checking will rely on user flagging which may be unreliable. This application will also not provide directions to a selected restroom either. It will only provide information about the restrooms and their locations visually on a map. However, links to other services that provide directions may be added.

This application is also only used to review current restroom locations and should not be used to discuss new physical locations of a new public restroom. Reviews are only meant to discuss the restrooms themselves and do not directly reflect the conditions of the building where the restroom is located or the behaviors of the people who use the restroom and should not be used to discuss such aspects of restroom locations.

In order to interact with the application, users must have a stable internet connection. Without a connection, data will not be able to be viewed and/or uploaded to the app.

2.0 Tasks

There will be two primary tasks required for the basic functionality of this application. The two primary tasks of for this application will be designing how data will be stored as well as how that data is displayed/gathered. In terms of data storage, each restroom will represented in a database as a class with various fields such as location, ratings, and other information about that specific restroom that will be needed to be displayed to the user. There will also be an algorithm designed to determine which restrooms to show and how many. In addition, there must be a method to handle restroom locations that were recently added to the map, as well as a way to avoid duplicate review threads for the same location.

The other primary task required for this application will be to design the user interface (UI) to display the data in an Android app. The UI will consist of displaying the data for each restroom such as ratings and comments, a map showing icons representing nearby restrooms, and any forms needed for data retrieval. This map will take advantage of location services or a user inputted location to determine which restrooms to show. These icons will be interactable and

when clicked, the related data for that restroom will be displayed on-screen. Included in the UI development is designing the forms that will be used to enter reviews and suggesting new restroom locations. After the form is filled, the user-inputted data should be automatically stored into the database.

There are a number of extra tasks/features that may be added as well. One of these features is a login system where users can create accounts where they can keep a record of reviews and comments they have made. An account/login system will also require storing usernames and a password authentication system. Also, a system that will take advantage of sessions and create, send, and authenticate cookies could be created if needed. Another feature that could be added is a flagging system that will allow users to report incorrect information. These flags can be dealt with manually or an algorithm can be designed to handle these flags autonomously. Given the limited timespan to complete the application, these extra features will only be included if the benefits of the additions will outweigh the risks and any inconvenient circumstances that may arise.

After implementation of these features, testing of the application will also be required. Performance standards will be tested by running the application on a smartphone running a currently supported version of Android. This will be done by a tester who will not take part in the development of the application code.

3.0 Resources

3.1 Hardware

A server will be needed to store data for each restroom such as ratings and comments. This server will be responsible for handling requests for restrooms near the location provided by the user. The algorithm for determining which restroom data to send will also be performed on this server. User information may also need to be stored on the server or on a separate server. If a separate authentication server is used, it will store all usernames of users as well as their corresponding password hashes. The authentication server will also handle all login requests. For testing purposes, a device capable of running a currently supported version of the Android Operating System as well as having internet connectivity will be required. As long as the device meets the two requirements stated, the specific hardware of the devices should not matter. However, a wide range of devices may be used to test performance on a wide range of device specifications.

3.2 Software

The data for this application will be displayed in an Android app so the UI will be written primarily using Android Studio. The functionality of the application will be written using Java. This includes any algorithms that need to be designed for determining what data is sent as a response to user requests to the server. PL/SQL will be used for database management needs. A currently supported version of Android OS will also be needed for

testing the application. Various versions of Android may be used during testing to ensure the application's performance meets standards on all supported versions of Android.

3.3 Team Members

Name	Role(s)
Ezra Taimuty-Loomis	Project Manager, Tester
Christian Armistead	Database Management, Sales
Andrew Tran	Document Specialist, Application Developer
Jacob Musone	Application Developer
Michael Jones	Document Specialist, GUI Developer