Common Sense

Android Application

Design Document

Joe Hammer, Darron Herbert, Jason Renna

Tyler Robinson, Dominic Nolt, Nicholas Scaramuzzi, Dylan Shapiro

Table of Contents

High Level Design

The purpose of this application is to to promote Common Sense, a persistent antimicrobial solution. Users will be required to create a login so that their name, address, location, etc. may be stored. The app will contain features that potential users of Common Sense may be interested in. Access to important health information in the form of news from the CDC and FDA will help keep users up to date and safe. It will also allow users to view their local weather and what effects it could have on their health. The app can be used to find where the product is sold and where it has been used to treat a location. A store is available to users would prefer to order the product through the application. The user will also be able to change their settings in the profile page.

Problem solving approaches

So far, there haven't been too many problems that were difficult to solve. The only one that really stands out is that we had a lot of ideas for what we could put into this app and had to narrow it down so that the app didn't become to cumbersome for a user to interact with and because we're limited on development time.

Another problem we had was how the store would be handled. Originally the idea was to make the store in the app and to handle everything there, however, the company that we're developing this app for already has a store website. After talking with the company, it was decided that we would simplify the app by linking the store to the website, rather than have an in-app store. With this simplification, we would then devote more time into accomplishing some of the stretch goals that the company would like to see implemented.

Screens

When the app is opened the user will be asked to login. First-time users will create an account using their Google Account. Once they have logged in they will be on the News screen which features up to date news from the CDC and FDC. The other screens of the app include the Weather, Map, Store, and Profile.

The Weather is obtained using Yahoo's API. This allows us to determine the weather based on a user's location and obtain a variety of information related to it. This information is also used to determine different health risks related to the current weather conditions.

The Map is obtained using Google Maps' API. This will show users where they can purchase Common Sense as well as what locations have been treated by the product in the area around the user's location.

The Store will allow users to access the website's store where they can view and purchase the products.

The Profile will contain all the user's settings and details which they can access and change.

Screen Navigation

We designed the app to be easy to navigate. The app opens with the login screen, once the user has logged in they are brought to the News screen. On every screen, the bottom part of the app shows an array of screen choices: News, Weather, Map, Store, and Profile. Tapping on the bottom navigation will transfer the user to the corresponding screen.

Flow Diagram

Technology Stack

The front end for the app is being developed in Android Studio using Java. The user interface uses XML for its layout and design. Since the app is only being developed for Android, we're using a combination of emulated hardware and Android devices for testing the software.

The Weather for the app is obtained using Yahoo's API. This allows us to determine the weather based on a user's location and obtain a variety of information related to it. This information is also used to determine different health risks related to the current weather conditions.

The Map is obtained using Google Maps' API. This will show users where they can purchase Common Sense as well as what locations have been treated by the product.

The backend for the application is using a database developed in MySQL and being linked to the app using PHP code.

Backend Information

The backend of this app consists of PHP code that will communicate with a remote MySQL database hosted on an AWS. This backend will be responsible for querying the data for user information, news and weather updates pulled from the relevant web crawlers, and the locations of areas related to Common Sense Hand Sanitizer. The results of these queries will then be processed and presented to the user through the front end application. A web crawler will search for health related news articles from sources such as the CDC and FDA, which will be supplied to the user. The backend will also be responsible for taking user entered information, such as names, email addresses, and physical addresses, and storing them in the MySQL database appropriately.

Authentication and Security

When users create an account with us by using their Google Account, we create an accountID for them. As developers, we know that privacy is an important ethic to maintain with our users so we only take their details that the app's features will use. The details that we will store in our database include the user's first and last name, their accountID, their ZIP Code, and the amount of points they have earned. The user's password will also be stored in the database as a hash value, so their password will be kept private and protected, without the actual

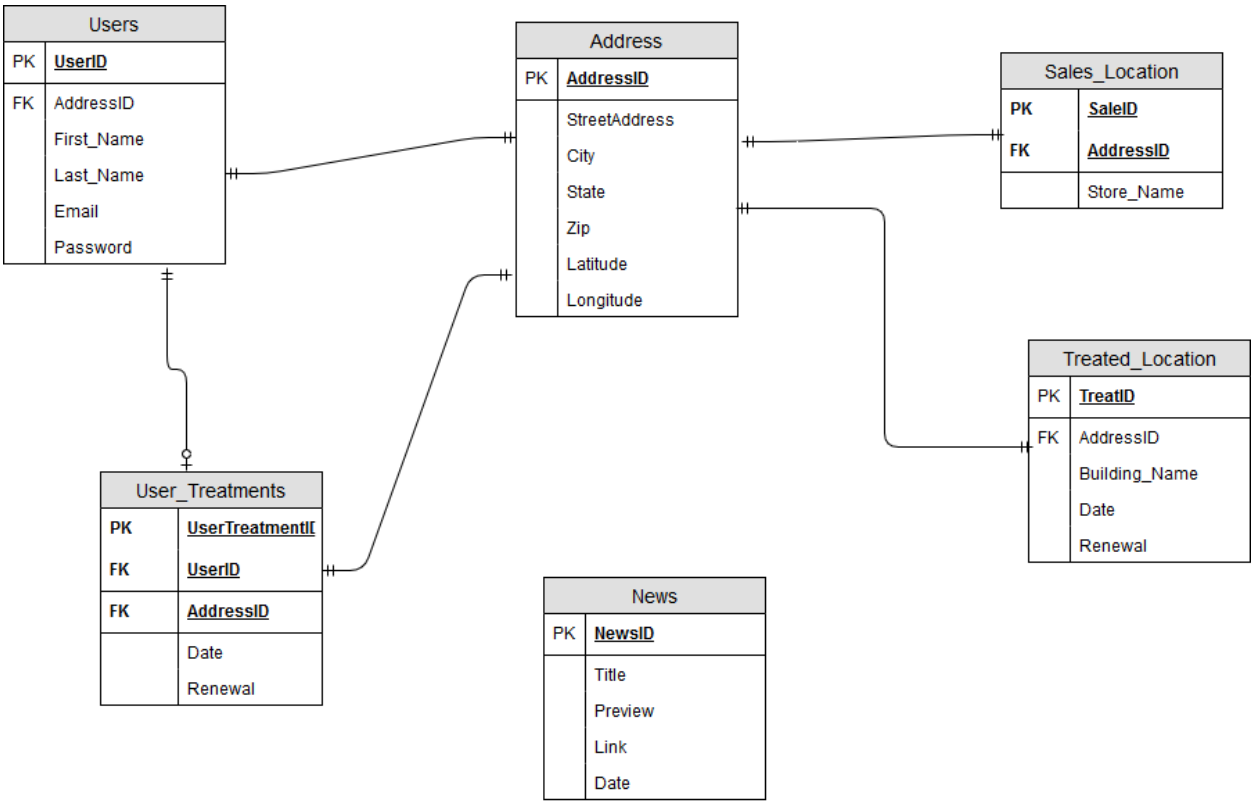
information being stored in the database. The user will also be prompted to allow/deny access to various functions, such as allowing the app the pinpoint the user's current location so the map screen can be more precise and helpful to the user.

Input and Output

In this app, there aren't too many input/output opportunities. The user must create an account, and once all the information is filled out, and will be taken back to the login screen. Once the user inputs their correct login information, they will be taken to the news screen where they can begin looking at the various sections of the app. The user may also input their address in the profile settings screen if they want to use their location to gather the weather/map data.

In the store screen, there will be a button that takes you to the room estimate calculator. Since the company we're working with provides room cleaning services, this calculator can be used to provide the user with an estimate of how much it will cost to clean a room of the inputted dimensions, as well as, possibly, an estimate of how long the process will take.

Database Schema



Endpoints

PHP code will be used to read and write information to and from the database.

When a user first starts the application, they will be required to register on the app using their gmail account username and password. Once the user finishes inputting all valid information and clicks the “register” button, the data they inputted will be added to the appropriate tables in the database (their password will be stored as a hash).

Once registration is complete, the user may then log in using their email and password. Their username and password hash will be compared with the values in the database. If the exact values do not match any sets of values in the database, prompt the user to try login in again.

If the user forgot their password, they may input their email and click the “forgot password” button, which will then send a password reset link/code to their inputted email address.

The news screen would retrieve the headlines and information from various websites that are stored in the database. This would include the article headline, summary, date, and a link to where the full article may be found.

The weather screen would retrieve the basic weather data from the user’s current location, taken by zip code. This might include temperature, weather (sunny, raining, etc.), wind, or even more detailed information such as UV index.

The maps screen would pull nearby locations from the database that sell the product or places that have been treated with the product. If the user has an address they would prefer to use, the app would pull the user’s address from their account information, otherwise, it would use the inputted zip code.

If the user wants to, they can input more information through their profile page, including their address to use in the “map” screen. Once saved, the information will be written to the database and saved alongside the user’s account information.