CS 491 Project

Acknet

1 Overview

Instead of completing a series of independent homework assignments, this semester you will be teaming up to build an *accessible* social network—called Acknet. This social network will be designed to connect people with limited language facilities.

2 Requirements

Development

1. Acknet is to be accessed by both Android and iOS clients, tablets and phones, portrait or landscape. The two clients are to support similar operations, and both must be native apps. (Abstraction frameworks like PhoneGap may not be used.)

2. Your client projects should be managed through a central code repository like Bitbucket, GitHub, or Google Code. You are to make your instructor (johnch@uwec.edu) a member of your project.

Model

1. Store the Acknet model in a central database, hosted on dario.cs.uwec.edu. Design the schema yourself.

2. Support a miniature social network, which is comprised of users, posts, and comments. You may choose to implement friendship and private messaging, but these are not required and should not be addressed until all other requirements are completed. For now, everyone may see everyone else's content. Advertisements are strictly outlawed.

3. Users may contribute images and videos captured by their mobile device or available in their device's gallery. Submitted media are to be stored using a web service of your choosing—not in your central database. (Be sure to pick a web service that has a friendly API. Many provide libraries that you can integrate into your project. Others provide a more manual URL-based REST API.) Media must be stored privately, accessible only through appropriate credentials. You may store identifiers in the database, but not the media itself.

4. An Acknet user's posts may be one of several types: text only, a single image possibly with text, a single video possibly with text, and a "geolocated" checkin. See the User Interface requirements for more detail on the last of these.

5. To access Acknet, a user must log in with a username and password. Authentication should be resolved on the server using information stored in the database. The user should remain logged in until explicitly logging out. Even if the user reboots the device, the user should not need to log in again.

User Interface

1. Support English and at least one other language. All user interface text must be available in the supported languages. (User-supplied content should not be translated.) If you have someone in your group proficient in some other language, make use of their proficiency. Otherwise, let me know and together we'll contact our friends in the Department of Foreign Languages.

2. Provide a means for text blurbs to be spoken using the device's text-to-speech capabilities.

3. Provide a means for text input to be captured using the device's speech recognition capabilities.

4. Construct a user interface that is simple, is comprised of widgets that are sized appropriately for users who may have limited dexterity, and supports back-navigation.

5. Use flexible layouts that accommodate and acknowledge the system-wide font size, which is set in device's settings, not in your app.

6. The user submits geolocated checkins with a single click. Use the GPS coordinates and resolve the coordinates to a more natural description: "Just checked in at Dairy Queen" or "Just checked in at 105 Garfield Ave." Enable this post type only if GPS is enabled.

7. All long running tasks should be performed outside the UI thread. However, all code directly affecting the UI should run on the UI thread.

8. Some facility must be provided for resetting or retrieving a user's password via email.

Networking

1. If the network is inaccessible when the app first loads, present the user with a dialog that takes them to the device's settings.

2. If a user attempts to add content while the network is inaccessible, queue up the operation until connectivity is restored, at which point, restart the pending operation.

3. Any server-side code scripts you write are to return results to your clients in JSON or XML form. Do not return HTML.