Potlatch - A Sample Capstone Project

The following is an example specification for a project that will be used in the dry run of the Capstone project for the Mobile Cloud Computing with Android (MoCCA) Specialization. Some variant of this specification may also be included in the full list of potential projects for the actual running of the Capstone project.

"Happiness is a gift and the trick is not to expect it, but to delight in it when it comes."

- Charles Dickens, Nicholas Nickleby

See a sunset that you just can't keep to yourself; a landscape that takes your breath away, an image that touches you personally? Then snap a picture, add some text or poetry to convey your thoughts, and give it to others with Potlatch[1]. Can your thoughts and images touch people around the world? Who is the greatest giver of gifts? Find out with Potlatch.

Functional Description and App Requirements:

1. A *Gift* is a unit of data containing an image, a title, and optional accompanying text.
2. A *User* can create a Gift by taking a picture (or optionally by selecting an image already stored on the device), entering a title, and optionally typing in accompanying text.
3. Once the Gift is complete the User can post the Gift to a *Gift Chain* (which is one or more related Gifts). Gift data is stored to and retrieved from a web-based service accessible in the cloud. The post operation requires an authenticated user account.
4. Users can view Gifts that have been posted.
5. Users can do text searches for Gifts. The search is performed only on the Gift's title. Gifts matching the search criterion are returned for user viewing.
6. Users can indicate that they were *touched* by a Gift and can also flag Gifts as being obscene or inappropriate. Users can set a preference that prevents the display of Gifts flagged as obscene or inappropriate.
7. Touched counts are displayed with each Gift. These counts are periodically updated in accordance with a user-specified preference (e.g., Touched counts are updated every 1, 5 or 60 minutes).
8. Potlatch can display information about the top "Gift givers," i.e., those whose Gifts have touched the most people.

Implementation Considerations

The Potlatch project specification outlined above is intentionally underspecified to maximize opportunities for student creativity. Students must therefore consider and choose between a number of design and implementation issues and solution alternatives to produce their final product solution. For example, students should consider at least the following issues for this project:

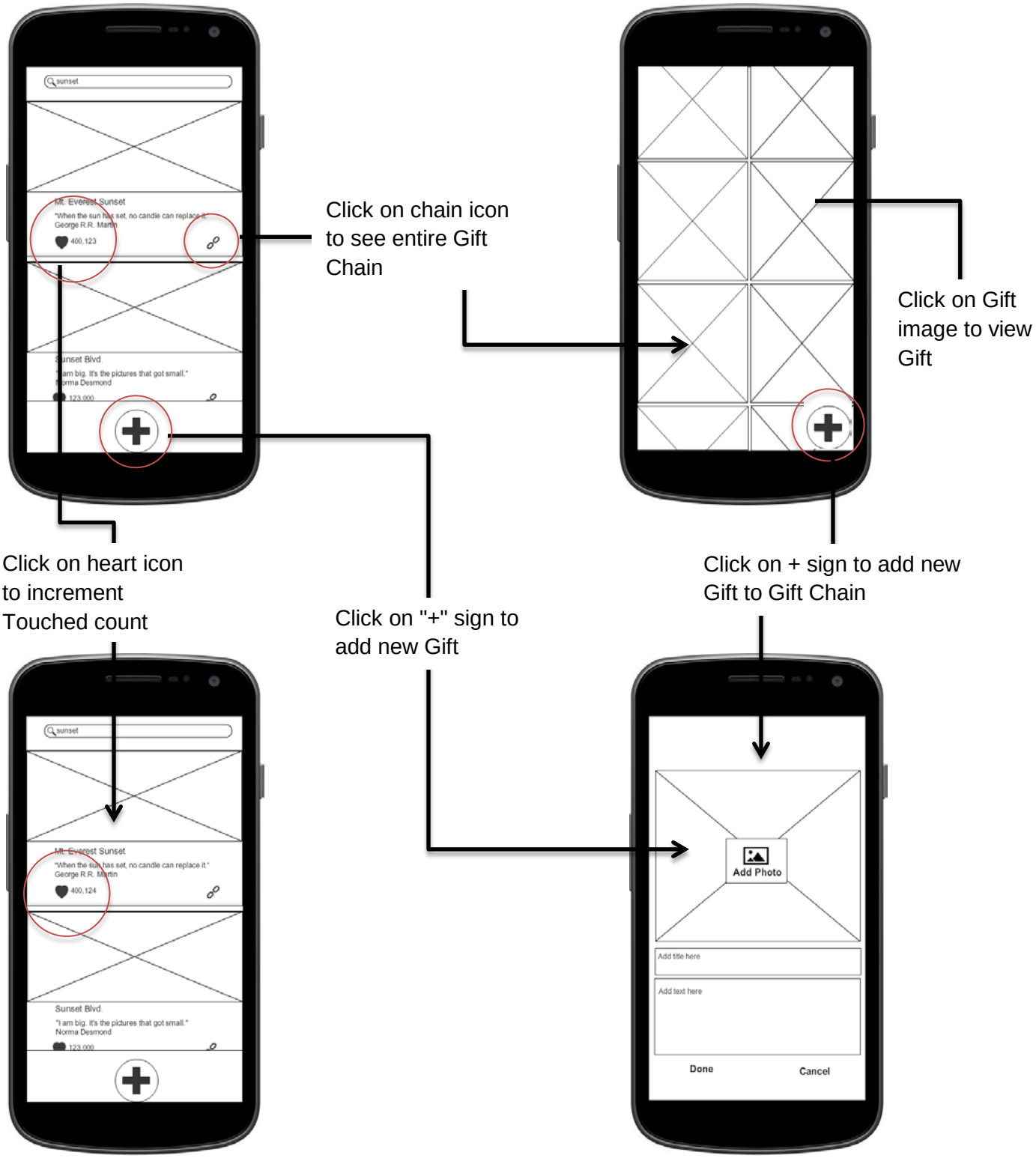
- How will Gift data be stored? In your device? In the remote service?
- What will the user interface screens look like? How will the user navigate between different screens?

[1] The word potlatch comes from the Chinook jargon, meaning "to give away" or "a gift" and is a gift-giving feast practiced by indigenous peoples of the Pacific Northwest Coast of Canada and the United States.

- How and where will users attach Gifts to a Gift Chain?
- How, when, and how often will the user enter their user account information? For example, will the user enter this information each time they run the app? Will they specify the information as part of a preference screen?
- How will Gifts and Gift Chains be accessed? Will there be some initial default display? Will users have to enter search criteria? If so, how will the user get access to the search interface?
- Will all search results be displayed at the same time? Or will only a subset of Gifts be shown at any one time? Will the Gifts be sorted in some way, such as by their Touched counts, creation time, or latest update time?
- How will users indicate that they were touched by a Gift? Can they undo their decisions? How will they flag inappropriate or obscene Gifts?
- What user preferences can the user set? How will the app be informed of changes to these user preferences?
- How will the user navigate between viewing/searching for Gifts and viewing top Gift givers?
- How will the app handle concurrency issues, such as how will periodic updates occur - via server push or app pull? How will search queries and results be efficiently processed? Will the data be pulled from the server in multiple requests or all at one time? Will the server data include full-sized images, or thumbnails plus URLs pointing to the full-sized images?
- Will the app cache information on the local device, e.g., in a Content Provider?
- Will the app provide extensions and improvements that go beyond the minimum requirements? For example, if an app collects location information for each Gift, queries by location, and then uses Maps to display Gifts at their locations, then it may be necessary to modify the various app databases and query facilities. Also, using Maps may require students to have access to an actual device. Also, how will these enhancements affect the rest of the app?
- Does your app idea really require two or more fundamental Android components? If so, which ones? For example, this app might benefit from using a ContentProvider or from using a background Service that synchronizes local and remote image data, only when the device is connected to a WiFi network.

Wire Frames

Below we've included some mocked-up screenshots suggesting a partial implementation of this app. Note that these screenshots do not cover all the required functionality and are included only to give students a sense of how they might document their intended app.


sunset
Mt. Everest Sunset
"When the sun has set, no candle can replace it."
George R.R. Martin
400,123
Sunset Blvd.
"I am big. It's the pictures that got small."
Norma Desmond
123,000
Click on chain icon to see entire Gift Chain
Click on Gift image to view Gift
Click on heart icon to increment Touched count
Click on "+" sign to add new Gift
Click on + sign to add new Gift to Gift Chain
sunset
Mt. Everest Sunset
"When the sun has set, no candle can replace it."
George R.R. Martin
400,124
Sunset Blvd.
"I am big. It's the pictures that got small."
Norma Desmond
123,000
Add Photo
Add title here
Add text here
Done
Cancel