POTLATCH PROJECT

I have really run out of time, and I'm not proud of the look of the code. A lot of improvements could be made, both to the code and comments, and to the documentation and the video. I apologize for the video being fuzzy, but you can get a good idea of how it works over a very slow internet connection.

All the code is open-source.

In the following table, all server code is relative to

  potlatch-server/src/main/java/localhost.potlatchserver

and client code is relative to

  potlatch-client/src/localhost.potlatchclient

unless otherwise specified.

	BASIC PROJECT REQUIREMENTS	Location in code
1	App supports multiple users via individual user accounts Rubric: 0 - Needs Improvement: Does not support multiple users 1 - Partially Complete: Supports multiple users via individual accounts, but cap on number of users is too small 1 - Partially Complete: Supports multiple users, but without individual accounts 2 - Complete: Supports multiple users via individual accounts (cap on number of users is reasonable)	Server: auth/OAuth2SecurityConfiguration.java line 167-174 Client: Config.java line 11
2	App contains at least one user facing function available only to authenticated users Rubric: 0 - Needs Improvement: No user facing function requires authenticated user account 1 - Partially Complete 2 - Complete: Contains one or more user facing functions requiring authenticated user account	All functions require authentication.

3	App comprises at least 1 instance of each of at least 2 of the following 4 fundamental Android components:
	    •Activity
	    •BroadcastReceiver
	    •Service
	    •ContentProvider
	Rubric:
	0 - Needs Improvement: Uses no components
	1 - Partially Complete: Uses at least one instance of only one component
	2 - Complete: Uses at least one instance each of at least 2 components
		Client: MainActivity.java and several other activities.

4	App interacts with at least one remotely-hosted Java Spring-based service Rubric: 0 - Needs Improvement: App does not interact with a Java Spring-based service 1 - Partially Complete 2 - Complete: App interacts with 1 or more remotely-hosted Java Spring-based service(s)	Client: client/MediaSvcApi.java and client/ChainSvcApi.java
5	App interacts over the network via HTTP Rubric: 0 - Needs Improvement: App does not interact over the network via HTTP 1 - Partially Compete 2 - Complete: App interacts over the network via HTTP	See above
6	App allows users to navigate between 3 or more user interface screens at runtime Rubric:	Client: MainActivity.java, CreateChainActivity.java,

POTLATCH PROJECT

	0 - Needs Improvement: Users can only access 1 (or no) interface screens at runtime 1 - Partially Complete: Users can navigate between only 2 user interface screens at runtime 2 - Complete: Users can navigate between 3 or more user interface screens at runtime	DisplayChainActivity.java, DisplayGiftActivity.java, etc.
7	App uses at least one advanced capability or API from the following list (covered in the MoCCA Specialization): multimedia capture, multimedia playback, touch gestures, sensors, animation.** **Learners are welcome to use ADDITIONAL other advanced capabilities (e.g., BlueTooth, Wifi-Direct networking, push notifications, search), but must also use at least one from the MoCCA list. Rubric: 0 - Needs Improvement: App does not use an advanced capability or API from the MoCCA list 1 - Partially Complete 2 - Complete: App uses at least one advanced capability or API from the MoCCA list above	Client: CreateGiftActivity.java line 60-64
8	App supports at least one operation that is performed off the UI Thread in one or more background Threads of Thread pool. Rubric: 0 - Needs Improvement: The app does not support any operations off the UI Thread 2 - Complete: The app does support at least one operation off the UI Thread	
	FUNCTIONAL DESCRIPTIONS AND APP REQUIREMENTS	
1	App defines a *Gift* as a unit of data containing an image, a title, and optional accompanying text. Rubric: 0 - Needs Improvement 1 - Partially Complete 2 - Complete	Client: repository/Media.java
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. Rubric: 0 - Needs Improvement 1 - Partially Complete: App supports taking a picture for a gift, but does not properly upload or asynchronously download the gift 2 - Complete: App supports taking a picture for a gift, uploading pic to server, and asynchronous download and display of image with gifts	Client: CreateGiftActivity.java and the resource /res/layout/gift_create.xml
3a	Once the Gift is complete the User can post the Gift to a *Gift Chain* (which is one or more related Gifts). Rubric: 0 - Needs Improvement 1 - Partially Complete 2 - Complete	A gift can only be created as part of a chain. See client: DisplayChainActivity.java line 54-64
3b	Gift data is stored to and retrieved from a web-based service accessible in the cloud. Rubric: 0 - Needs Improvement 1 - Partially Complete 2 - Complete	Client: CreateGiftActivity.java line 68-104
3c	The post operation used to store Gift data requires an authenticated user account Rubric: 0 - Needs Improvement: Login is not required to post a gift 1 - Partially Complete: Login & session cookie, Basic Auth, HMAC auth, or OAuth 2.0 bearer token required but w/out HTTPS to post gift 2 - Complete: Login & session cookie, Basic Auth, HMAC auth, or OAuth 2.0 bearer token required along w/HTTPS to post gift	All operations require OAuth2.
4	Users can view Gifts that have been posted. Rubric: 0 - Needs Improvement: Users cannot view Gifts 2 - Complete: Users can view Gifts	Client: DisplayGiftActivity.java

POTLATCH PROJECT

5a	Users can do text searches for Gifts performed only on the Gift's title. Rubric: 0 - Needs Improvement: The app does not provide search capabilities 1 - Partially Complete: The app supports exact searches for the gifts that are managed client-side 2 - Complete: The app supports exact searches for the gifts that are managed server-side	Server: repository/MediaRepository.java line 22 and Client: MediaSvcApi.java lines 184-185
5b	Gifts matching the search criterion are returned for user viewing. Rubric: 0 - Needs Improvement: The app does not provide search capabilities 1 - Partially Complete: The app properly searches for gifts matching the specified criteria client-side 2 - Complete: The app properly searches for gifts matching the specified criteria server-side	See above.
6a	Users can indicate that they were *touched* by a Gift, at most once per Gift (i.e., double touching is not allowed) Rubric: 0 - Needs Improvement: App does not include touched marker 1 - Partially Complete: App includes ability to be touched by a gift, but does not prevent double "touching" 2 - Complete: Includes ability to be touched by a gift & double "touching" is not permitted	Client: DisplayGiftActivity.java lines 103-117
6b	Users can flag Gifts as being obscene or inappropriate. Users can set a preference that prevents the display of Gifts flagged as obscene or inappropriate. Rubric: 0 - Needs Improvement: App does not include obscene flags 1 - Partially complete: App includes ability to flag a gift as obscene but does not properly filter out gifts flagged as obscene 2 - Complete: Includes ability to flag gift as obscene & properly filters out obscene gifts	Client: DisplayGiftActivity.java lines 125-141, MediaSvcApi.java lines 149-150,
7a	Touched counts are displayed with each Gift Rubric: 0 - Needs Improvement: App does not track touched counts 1 - Partially Complete:Touched counts are displayed w/ each gift but the counts are only updated when the associated Activity or App is launched 2 - Complete:Touched counts displayed w/each gift. e.g., auto-poll updates at specified intervals, push notifications for real-time updates, etc	Client: DisplayGiftActivity.java lines 100-101 and GiftArrayAdapter.java lines 123-124
7b	Touched counts can be periodically updated in accordance with a user-specified preference (e.g., Touched counts are updated every 1, 5 or 60 minutes) or updated via push notifications for continuous updates. Rubric: 0 - Needs Improvement: Touched counts are not updated 1 - Partially Complete: Touched counts are updated periodically, but not via a user-specified preference 2 - Complete: Touched counts are updated based on a user-specified preference	
8	App can display information about the top "Gift givers," i.e., those whose Gifts have touched the most people. Rubric: 0 - Complete: The app includes a list of the top gift givers 1 - Partially Complete: The app includes counts of the number of gifts given by each user 2 - Complete: The app includes a list of the top gift givers	