Camp Potlatch Application Project

Sometimes we suddenly fall in the middle of an unforgettable scene which we want to record for our delight or inspiration at future times. Following the ideas from the Mobile Cloud Computing with Android Specialization at Coursera, the Camp Potlatch Application intends to do more than this! It intends to give the means to not only record scenes, but also to share them with everyone. They are real gifts from the moment, from you, to be shared to the whole global society. A gift is a delightful scene, image, instant, such as an inspiring landscape, news record, which the user desires to show to others to share this special and important moment.

This document represents the Camp Potlatch project, which describes the use cases, non-functional requirements, data model, behavior and architecture for this application.

Use cases

This section describes the possible use scenarios for this application and how each actor interacts in order to perform this uses. The diagram below summarizes the use scenarios and the actors allowed to perform them.

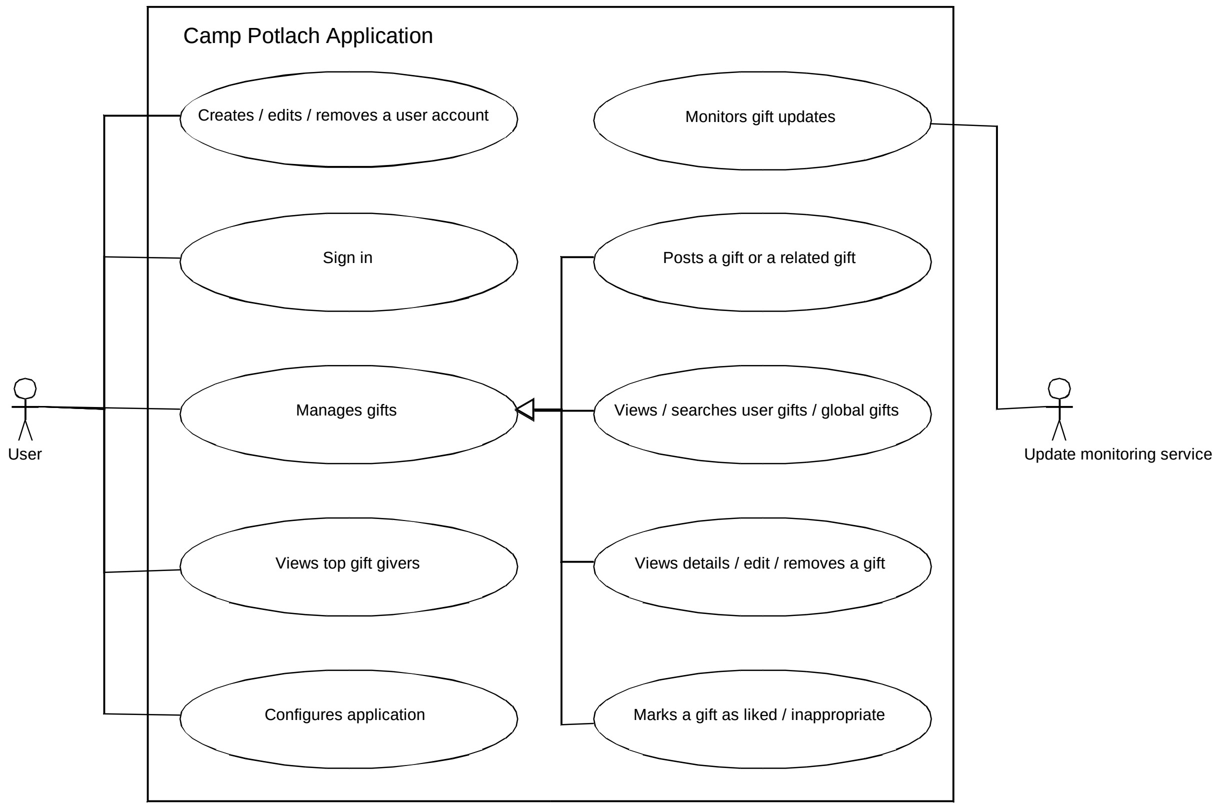
Camp Potlach Application
Creates / edits / removes a user account
Monitors gift updates
Sign in
Posts a gift or a related gift
Manages gifts
Views / searches user gifts / global gifts
Views top gift givers
Views details / edit / removes a gift
Configures application
Marks a gift as liked / inappropriate
User
Update monitoring service

Figure 1: Camp Potlatch Use Cases

User account creation

A person can create a user account in a “user account screen”. The user can access this screen touching the “sign up” button in the “login screen”.

The user account has login, password, name, e-mail, photo, city, country, and custom application settings.

The user can touch the “take photo” or “choose image” button to insert a photo in his account. If the “take photo” button is touched, the user can take a photo of himself and touch “ok” to return. If the “choose image” button is touched, he can choose an existing image from the mobile device file system and touch “ok” to return.

After filled, the user touches “create” button to create an account.

HTTP requests:

POST /user

POST /user/<login>/image

User account edition

The user can also access the “user account screen”, from the top menu, to edit his account properties, except his login. He touches “save” to update this account.

HTTP requests:

GET /user/<login>

GET /user/<login>/image

PUT /user/<login>

PUT /user/<login>/image

User list viewing

The user can access the list of users through the application menu. He is redirected to the “user list screen” that show a list of users ordered descending by creation time. Each user presented has a photo, his name, city, country, likes counting and creation time.

HTTP requests:

GET /user?offset=<offset>&limit=<limit>

GET /user/<login>/image

A certain amount of users are previously shown and other users from the list can be shown after the list end using a "next" button. If the user touches the user name, he is redirected to the “user gift list screen” of this user.

User list searching

The user can search for users using the search text field at the top of the “user list screen”. The application searches for the user using the service. All users which have a part of its name equals to the search criteria are returned. The list is also ordered by creation time.

HTTP requests:

GET /user?name_part=<name_part>&offset=<offset>&limit=<limit>

GET /user/<login>/image

A certain amount of users are previously shown and other users from the list can be shown after the list end using a "next" button. If the user touches the user name, he is redirected to the “user gift list screen” of this user.

Top gift givers list viewing

A user can access the “top gift givers list screen” through the application menu. The users which give (post) gifts which have more likes summing up are presented on the top gift givers list. This list is ordered descending by the number of likes by each user. The user can search for users using the search text field at the top of the “top gift givers list screen”. All users which have a part of its name equals to the search criteria are returned. A certain amount of users are previously shown and other users from the list can be shown after the list end using a "next" button. If the user touches the user name, he is redirected to the “user gift list screen” of this user.

HTTP requests:

GET /user/findTopGiftGivers?name_part=<name_part>&offset=<offset>&limit=<limit>

GET /user/<login>/image

Custom application configuration

A user can configure the update period of the gift list view for his account. A gift list shows new gifts and updates gifts contents, such as liked counting, and gifts related to each gift in a gift chain, from time to time.

A user can choose to hide inappropriate gifts.

HTTP requests:

GET /user/<login>

GET /user/<login>/image

PUT /user/<login>

User account statistics

The user can access the user account details from the application menu in order to view his properties and statistics.

HTTP requests:

GET /user/<login>

GET /user/<login>/image

User account removal

At anytime, the user can remove his account from the application. In order to remove his account, he enters the “user account screen” through the application menu. In this screen, the user touches the “remove” button to remove the account.

HTTP requests:

GET /user/<login>

GET /user/<login>/image

DELETE /user/<login>

Sign in

A “login screen” is used to authenticate users. The user puts his login and password and touches the “sign in” button. After authentication, the user is redirected to the “global gift list screen”.

In this process, a token is created which carries the user credentials, through the interaction with the “/oauth/user” end point. The application functionality is customized according to these credentials and related contents are presented based on them.

HTTP request:

POST /oauth/user

Sign out

A user can select the option from menu to sign out from the application. The token is removed and the user is redirected to the “login screen”.

Gift posting

A gift is an image, such as special scene, which the user desires to share with others. It is composed of an image, a title and an accompanying text. The image can be taken with the mobile device camera or chosen among the existing images in its file system.

A user can post gifts to a global list of gifts or to his particular list of gifts. The posted gifts by all the users can be viewed on the global gifts list or top gifts list; the gifts posted by a particular user can be viewed on his user gifts list.

A user posts a gift on the “global list screen” or on the “user list screen” touching “post gift” button at the top of these screens. The user is redirected to the “gift edition screen”, where he can take or select an image, fill its title and accompanying text (comments).

When he choose the “take” button, he is redirected to the “image capture screen”, where he can take the image touching the “take” button and touching “ok” button to confirm. He is then redirected in return to the “gift edition screen”.

From the “gift edition screen”, he can alternatively choose an image from the device file system. He is redirected to the “image selection screen” after touching the “choose” button. The he touches the “ok” button to confirm the selection. He is then redirected in return to the “gift edition screen”.

After, completed the gift, the user touches “post” button to post the gift and is redirected to the original gift list.

A user must to be authenticated in order to post and view gifts in the Camp Potlatch application. The integrity and confidentiality of the gifts exchange is secured by encrypted communication, using HTTPS.

HTTP requests:

POST /gift

POST /gift/<id>/image

Gifts viewing on global gifts list

The “global gift list screen” appears after the user sign in and can also be accessed using the application menu. The global list shows all the gifts posted by all users, ordered by the insertion time, limited by a predetermined number of gifts.

HTTP requests:

GET /gift?title_part=<title_part>&offset=<offset>&limit=<limit>

GET /gift/<id>/image

The “title_part” parameter from the request are used to search gifts by parts of title, as explained at the gifts searching section.

The user can touches the creator name. After, the user is redirected to the user gifts list for the selected creator. More gifts can be viewed by touching the “next” button at the end of the list.

Gifts viewing on top gifts list

The “top gift list screen” can be accessed using the application menu. The top gifts list shows the global the top gifts of all times which have more likes from users.

HTTP requests:

GET /gift/findTopGifts?title_part=<title_part>&offset=<offset>&limit=<limit>

GET /gift/<id>/image

The “title_part” parameter from the request are used to search gifts by parts of title, as explained at the gifts searching section.

The user can touches the creator name. After, the user is redirected to the user gifts list for the selected creator. More gifts can be viewed by touching the “next” button at the end of the list.

Gifts viewing on user gifts list

The “user gift list screen” of the current user can also be accessed through the application menu. The “user gift list screen” of other users can be accessed touching the user name in other gifts list or in the “users list screen”. A “user gift list” shows all the gifts posted by the associated user, ordered by the posting time, limited by a predetermined number of gifts.

More gifts can be viewed by touching the “next” button at the end of the list.

HTTP requests:

GET /gift/findByCreatorLogin?creator_login=<creator_login>&title_part=<title_part>

&offset=<offset>&limit=<limit>

GET /gift/<id>/image

The “title_part” parameter from the request are used to search gifts by parts of title, as explained at the gifts searching section.

Related gift posting (gift chain creation)

Any user can post a related gift as response to a gift presented on a gift list, known as the caption gift. Related gifts posted as response to a caption gift creates a gift chain.

A user touches the “related gifts” button, right below the caption gift, to view the related gifts in the “related gifts list screen”. The user then touches the “post” button on the top of this screen to post the related gift. He is redirected to the gift edition screen, where he can take or select an image, fill its title and accompanying text (comments). The process proceeds in similar way to the gift posting use case, but it remember which gift is its caption gift in creation.

Besides the “related gifts” button is presented the number of existing related gifts to the caption gift.

HTTP requests:

POST /gift

POST /gift/<id>/image

Related gifts viewing (gift chain viewing)

In a gift list screen, the user can touches the “related gifts” button right below a gift to view its related gifts. After he is redirected to the “related gifts list screen” which shows the related gifts ordered by the posting time with the same template from the other gifts list screens.

HTTP requests:

GET /gift/findByCaptionGiftId?caption_gift_id=<gift_id>&title_part=<title_part>

&offset=<offset>&limit=<limit>

GET /gift/<id>/image

The “title_part” parameter from the request are used to search gifts by parts of title, as explained at the gifts searching section.

Gift search

A user can search gifts by title. In order to search gifts, he must put the desired text in a query field at the top of any gift list and touch the “search” button.

The search is processed by the application service in the server tier according to the search context: global list, top list, user list or related list. Only caption gifts are searched on the global list, top list and user list. Only related gifts are searched on related list.

After the execution, the results are returned to the list that originated it, restricting this list to the gifts that are resolved according to the text searched and the context. All gifts from the context which have a part of its title equals to the search criteria are returned.

The user can clear the search field, to show all gifts again, by touching a “X” button besides the “search” button.

HTTP requests:

GET /gift?title_part=<title_part>&offset=<offset>&limit=<limit>

GET /gift/findTopGifts?title_part=<title_part>&offset=<offset>&limit=<limit>

GET /gift/findByCreatorLogin?creator_login=<creator_login>&title_part=<title_part>

&offset=<offset>&limit=<limit>

GET /gift/findByCaptionGiftId?caption_gift_id=<gift_id>&title_part=<title_part>

&offset=<offset>&limit=<limit>

GET /gift/<id>/image

The "title_part" parameter from the request are used to search gifts by parts of the gift title.

Gift details viewing

When the user touches the image of a gift, he is redirected to the "gift details screen". This screen shows the image in larger size, with title. The accompanying text is presented below in few lines, and can be truncated with the "..." characters at the end, if necessary.

HTTP requests:

GET /gift/<id>

GET /gift/<id>/image

After the user can touch "ok" to return to the original gift list.

Gift edition

A user can touch the "edit" button below a gift created by himself to edit it. In this case, he is redirected to the "gift edition screen". The user can modify the title and the accompanying text in this screen. After the edition, the user touches the "save" button to save the modification and is redirected to the original gift list, where he can see the modified gift.

HTTP requests:

GET /gift/<id>

GET /gift/<id>/image

PUT /gift/<id>

PUT /gift/<id>/image

Gift removal

A user touches the "remove" button below the gift to remove a gift created by himself. After, a dialog is presented to confirm his intention. If confirmed, this gift is removed. Moreover, if it is a caption gift, all its related gifts are removed as well. After, the gift list view is updated to reflect this removal.

HTTP requests:

DELETE /gift/<id>

Gift liked marking

A gift can be marked as liked by a user when he touches the “like” button below the gift. This count is maintained by the application in the server side. A user can only like a gift only one time. A user can unmark a liked gift touching the “unlike” button.

Gifts present besides the “liked” button the number of likes from all the users.

HTTP requests:

POST /gift/<id>/like

POST /gift/<id>/unlike

GET /gift/<id>/isLiked

Gift inappropriate marking

A user can mark a gift as inappropriate. This mark is useful to indicate obscene or inappropriate content in a gift.

Each user can mark one time the gift as inappropriate by touching the “denounce” button. This user can also reverse his judgment and unmark the gift as inappropriate by touching the “undo denounce”.

HTTP requests:

POST /gift/<id>/markInappropriate

POST /gift/<id>/unmarkInappropriate

GET /gift/<id>/isMarkedInappropriate

Gifts update monitoring

The tracking of new and modified gifts and count of likes are performed by an Android service, which queries from time to time the service in the server side in order to notify the current status to the activity responsible by the gift list view. The new and modified gifts, new and modified related gifts, and modified liked count, are then passed to the activity that handles the view redrawing.

Non-functional requirements

This section describes the non-functional requirements for the application, which are related to performance, responsiveness, security, reliability, dependability and user experience.

Support to multiple users

A database table is used to store user accounts, containing login, name, e-mail, city, country and preferences for the user. Every person can create a user account and access the application using it.

Support to user authentication

The authentication is served by the OAuth protocol, which recognizes the mobile client and the user. The authentication will be provided by matching the password against the value stored in a database.

Two or more components of the Android platform

The user interacts with the application through Android activities. Each activity presents a view and accepts user commands.

An Android service is used to monitor new gifts from other users and modification to existing gifts, such as the liked count. This gifts are stored in the server, and are monitored by the Android service in a pooling mode. It notifies the gift list activity to reload a number of gifts when there are modifications.

User interaction through three or more screens

The user can interact to some screens to enter in the application, publish gifts, view gifts from his or other profiles, add related gifts to other gifts (gift chain) and mark gifts as liked or inappropriate.

Some screens (or views) are listed below:

1. Login – used to enter in the application after proper authentication or to create a new user;
2. Global gifts list – displays the recent gifts from every user;
3. Top gifts list – displays the top gifts from every user;
4. User gifts list – displays the gifts published by the current user or by a selected user;
5. Image capture – allows to capture or select an image in the device in order to create a gift;
6. Gift creation / edition – allows to edit the gift title, accompanying text after image capture;
7. Gift details – show gift details and related gifts in the chain;
8. User account edition – user account edition and custom application settings;
9. Top gift givers list – shows a list of users ordered descending by the top gift givers and allows to query users by name.

Use of background threads for content downloading

A background thread is used to give responsiveness to the user interface, performing content downloading while the user thread is free to receive user requests.

Interaction with a server side service

The client tier, in Android mobile devices, interacts with a service hosted in the cloud in order to store and recover contents, and reach distributed functionality, such as queries, and contents stored by other users.

Interaction using the HTTPS

The communication between the client and the server tier is provided by JSON messages and multiparts bodies supported by HTTPS.

Android advanced capability

The application uses multimedia capture to take photos from the landscape or locations, the gifts given to other users. Also, it leverages the interaction using gestures to roll a list of gifts and other operations. In a future development iteration, it can also use GPS location sensor to tag gifts.

Model

The Camp Potlatch application deals with data objects in order to maintain and present gifts, users, likes and other related concepts. The types and relations among these objects are shown in the Figure below.

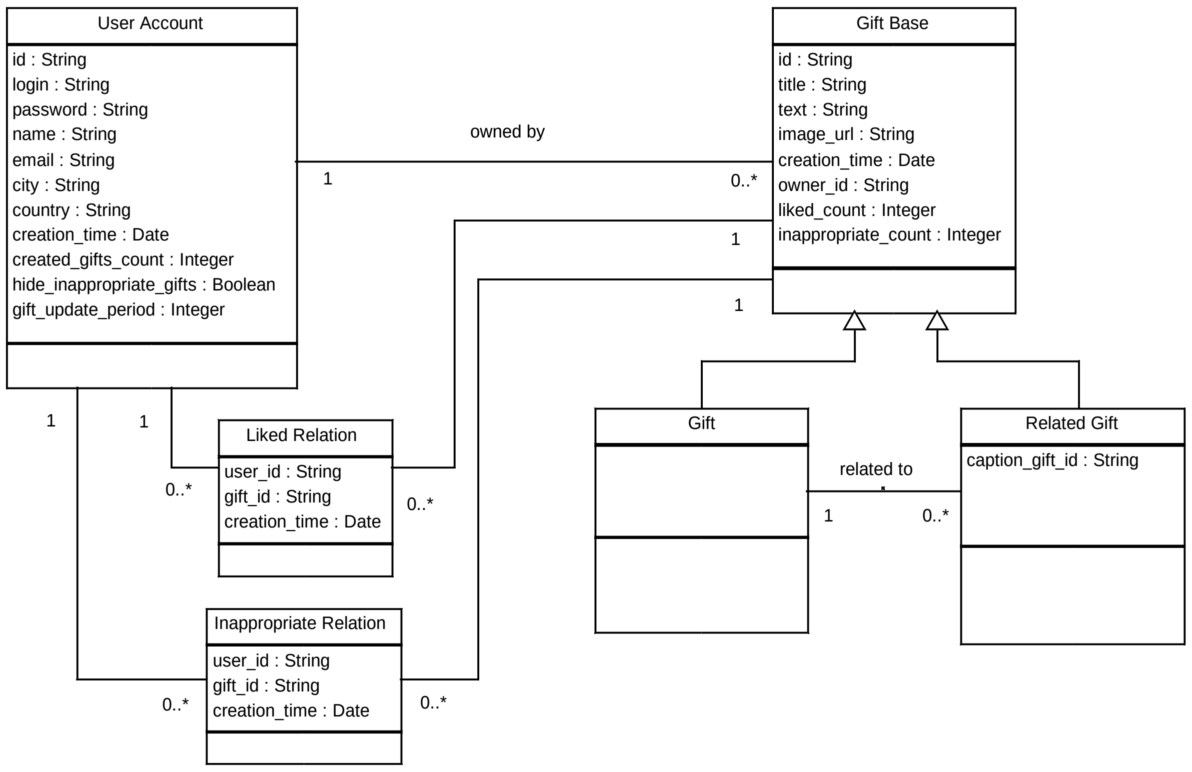

Figure 2: Camp Potlatch Model

User account

A user is represented in the application by a user account. It contains the user id, login, encrypted password, name, e-mail, city, country, creation time, created gifts count, and custom application settings.

The custom application settings specifies: if the user wants to view gifts marked as inappropriate, and the time period between verifications of new and modified gifts.

Gift

A gift is a delightful image, such as an inspiring landscape, which the user desires to show to others to share this special and important moment. It is composed of a gift id, image (captured or loaded from the mobile device), title, optional accompanying text, creation time, id of the owner user (who created it), liked count and inappropriate count.

After the gift creation, it is stored in a server from the AWS cloud. When a user wants to view gifts, on a list or its details, it is restored from a server. The contents are received and sent by a service hosted in this cloud server through HTTPS requests.

Related gift (of a gift chain)

A related gift is posted by some user which intends to complement the message passed by another posted gift, known as the caption gift. The caption gift and its related gifts form a gift chain.

On a gift list view, first is presented the caption gift in a large size. Below, other related gifts are shown in small size. The list are presented by order of insertion – first the newest, and so on, until some predetermined quantity.

A list of related gifts can be added to a posted gift by any user. The user touches the “post related gift” button below the caption gift to add a related gift in the gift chain.

A related gift has the same fields of a gift plus a caption gift id, that identifies what is the gift which it is related to.

Liked relation

If a user is particularly touched by a gift, he can express his feelings touching the “like” button below the gift. The application then stores the liked expression in a relation between the liked gift and the user who has marked it.

The liked relation has as fields: the id of the user who has liked the gift, the id of the liked gift, and creation time.

The user can also unlike the gift. In this case, the corresponding relation is removed.

Whenever a relation is created or removed, the liked count of the related gift is also updated in the server side.

Inappropriate relation

If a user is offended by a gift, because it is perceived as inappropriate or obscene, this user can mark the gift as inappropriate touching the “denounce” button. The application then stores the inappropriate relation between the denounced gift and the user who has marked it.

The inappropriate relation has as fields: the id of the user who has denounced the gift, the id of the inappropriate gift, and creation time.

The user can also unmark an inappropriate gift marked previously by himself, if he has changed his mind. In this case, the corresponding relation is removed.

Whenever a relation is created or removed, the inappropriate count of the related gift is updated in the server side. A gift is considered inappropriate if its inappropriate count is different from zero.

Application architecture

The application is divided in client and server tiers.

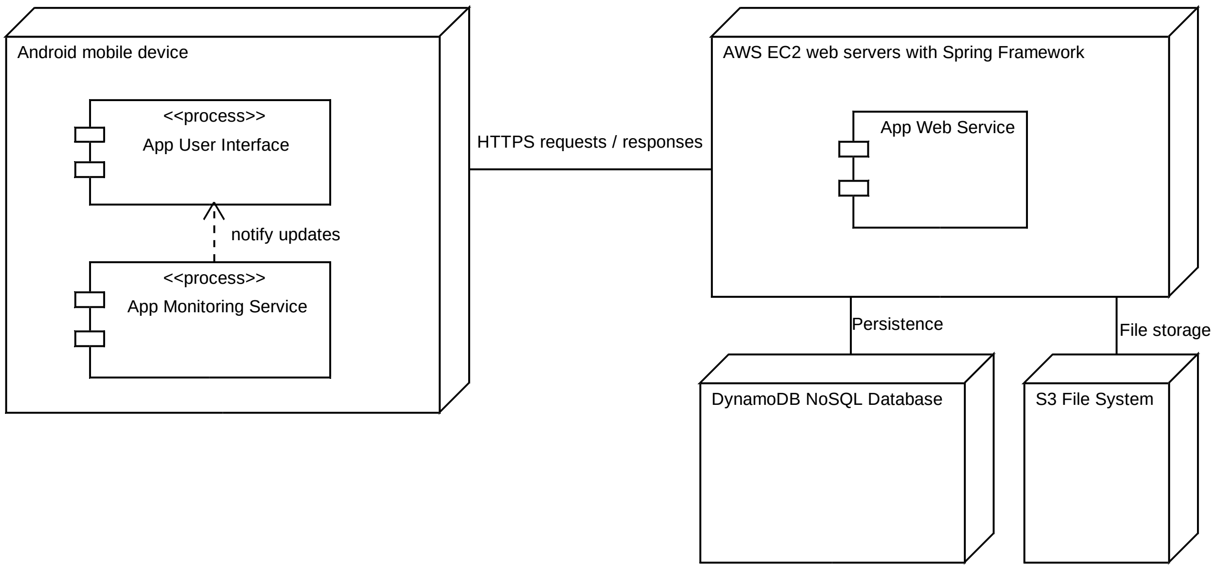

Figure 3: Camp Potlatch Architecture

The client tier is developed in Java using the Android framework and runs in Android mobile devices with Android 4.4 and newer versions. It contains a user interface, composed of several Android activities, to allow user interaction, to post gifts and related gifts, view shared gifts, and see the top gift givers, among other associated actions. Also, the client tier has an Android service to monitor new gifts and modification to existing gifts and its related properties, such as the liked count or new related gifts.

The server tier is written in Java and is implemented using the Spring framework. It is distributed, running as stateless web services allocated in Tomcat containers hosted in Amazon Web Service (AWS) EC2 Linux machines in the cloud. It is based on the MVC design pattern, and interact with the client through JSON and multipart bodies messages over HTTPS in a RESTful interface. The server is composed of: a business layer with some controllers; a model layer with data objects; and a persistence layer to interact with a noSQL database using JPA.

The Amazon DynamoDB noSQL database is used to persist data objects. The Amazon S3 file system is used to stored image files and backup.

In the client side, the interaction to the server using JSON messages is provided by Retrofit, which serializes restricted POJO objects in JSON for requests and the reverse for responses.