THE ANDROID CAPSTONE PROJECT 2015

GhostMySelfie App

DAILY SELFIE MAP OF CLASSES AND METHODS

App supports multiple users via individual user accounts.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	LoginScreenActivity - ● main Activity where user can login.	login() and signUp() methods check if the device has internet connection. login() - ● Stablish a connection with the server once the user is registered with signUp() method, if connection is not stablished throws an error message as Toast. signUp() - ● Register a user, for this, first valid a user

		entered a valid email address as username. check if user is already registered, if not, save the user credentials in the server.
com.laishidua.presenter;	GhostMySelfieOps - ● Provides all the GhostMySelfie-related operations.	dbSyncGhostMySelfieList(...) - ● Updates the local database, inserting missing selfies and updating the fields of the existing ones ● If current user loged in is other than the last one, it reload the sqlite database with the selfies that correspond to the current user.

MORE RELATED CLASSES AND METHODS FROM SERVER		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.mobilecloud.ghostmyselfie.controller;	UserController - ● It contains methods to register or login a user	addUser(...) - ● it receives a User to be registered saving its email, username (in this case the same email) and password. ● To register a user first check if email, username and password already exist, if not the server saves the new user and its attributes for example its roles.

		checkCredentialsState(...) - ● Check if a user is available to register. login(...) - ● Login a user by its credentials.
com.laishidua.mobilecloud. ghostmyselfie.client;	UserSvcApi - ● This interface defines an API for register or login a user. ● The interface is annotated with Retrofit. ● Clients can use this interface. ● The path for this interface starts with /insecure/user/...	

App contains at least one user facing function available only to authenticated users

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.presenter;	GhostMySelfieOps - ● Provides all the selfies-related operations.	uploadGhostMySelfie(...) - ● Start a service that Uploads a selfie having given Uri and a notification Id that will triggered for each selfie uploaded. getGhostMySelfieList() - ● Get the list of selfies corresponding to the logged user.
com.laishidua.presenter;	GhostMySelfieDetailOps - ● Provides all the selfie detail-related operations.	downloadGhostMySelfie(..) - ● Start a service that downloads the selfie having given Id and title to save in the MediaStore. deleteGhostMySelfie(...) - ● Start a service that deletes the selfie in client and server having given Id (title param is not used). rateGhostMySelfie(...) - ● Start a service that rates a Selfie having

		given Id and rate value. it represents in the view of detail activity as stars.

MORE RELATED CLASSES AND METHODS FROM SERVER		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.mobilecloud. ghostmyselfie.controller;	GhostMySelfieController - ● It contains methods related to each user selfie.	addVote(...) - ● Adds a vote to the collection of votes for a Selfie. updateVote(...) - ● Updates a user's vote for a Selfie. updateGhostMySelfie(...) - ● Updates the non-key properties of an existing Selfie with the new ones; used when the posted Selfie already exists according to its hashcode. getGhostMySelfieList(...) - ● Serves the Selfie list from a user, leaving the heavy lifting to Spring. rateGhostMySelfie(...) - ● method to rate a selfie, only a a selfie is rated when rate corresponds to its user owner.

App comprises at least 1 instance of each of at least 2 of the following 4 fundamental Android components:

• Activity

ACTIVITIES AND ITS MORE IMPORTANT METHODS		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	GhostMySelfieDetailActivity - ● This Activity is used to show the detail of each selected selfie	onCreate(...) - ● ● Hook method called when a new instance of Activity is created. ● It has the listener methods related to ratingBar (when user click on stars rating), thumbnail (if image is not on device, it can be downloaded from server otherwise shows the selfie in full screen) and delete a row item from the list.
com.laishidua.presenter;	GhostMySelfieDetailOps - Explained before.	showGhostMySelfie(..) - ● start an Activity to show a Selfie in fullscreen.
com.laishidua.view;	GhostMySelfieListActivity - ● This Activity is used	onCreate(...) - ● Hook method called

	upload a selected selfie to a GhostMySelfie. Service and also displays a list of selfies references available at the GhostMySelfie. Service. The user can take a selfie or get a selfie from gallery and upload it. It implements OnGhostMySelfieSelectedListener that will handle callbacks from the UploadGhostMySelfieDialog Fragment. It extends GenericActivity that provides a framework for automatically handling runtime configuration changes of an GhostMySelfieOps object, which plays the role of the "Presenter" in the MVP pattern. The GhostMySelfieOps.View interface is used to minimize dependencies between the View and Presenter layers. User can select one or more row items to make modifications to its related selfies.	when a new instance of Activity is created. ● It has the listener methods related to upload or make modifications to a selfie or selfies at sametime. onGhostMySelfieSelected(...) - ● The user selected option to get Selfie from UploadGhostMySelfieDialogFragment. Based on what the user selects either to take a Selfie or get a Selfie from the ImageGallery onOptionsItemSelected(...) - ● Hook method called whenever a settings button (config button) is selected in action bar.
com.laishidua.view;	LoginScreenActivity - ● main Activity where user can login.	Explained above.

com.laishidua.view;	SettingsActivity - ● Activity related to the settings that user choose to upload the selfies or set an alarm notification.	onCreate(...) - ● Hook method called when a new instance of Activity is created. ● Sets an alarm and the next listener methods: addListenerOnChkGhost() - ● If it is checked, a ghost will appear in the selfie that we select or take. addListenerOnChkBlur() - ● If it is checked, the selfie that we select or take will get the blur filter. addListenerOnChkGray() - ● If it is checked, the selfie that we select or take will get the gray scale filter. addListenerOnChkDark() - ● If it is checked, the selfie that we select or take will get the dark filter as took at night. addListenerOnChkActivateReminder() - ● Activate an alarm notification that when triggered it sounds a scream. addListenerOnReminderTime() - ● Related to the time, when time change it is saved in database.

• **BroadcastReceiver**

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.utils;	AlarmNotificationReceiver - ● NotificationReceiverIntent to be hold by a PendingIntent.	onReceive(...) - ● Set an Intent to be used when the user clicks on the Notification View. ● Set the PendingIntent that wraps the underlying Intent. ● Build the Notification that sounds like scary scream. ● Pass the Notification to the NotificationManager.
com.laishidua.view;	GhostMySelfieDetailActivity - Explained before. UploadResultReceiver - it is in same class of GhostMySelfieDetailActivity, the name should be DownloadAndRateResultReceiver but it is not changed for lack of time. It is the Broadcast Receiver that registers itself to receive result from GhostMySelfieService.	registerReceiver() - ● Register a BroadcastReceiver that receives a result from the GhostMySelfieService when a selfie download completes or a selfie rate completes. onReceive() - Belongs to this UploadResultReceiver Class ● Depending of “Operation” value, it

		performs an action.
com.laishidua.view;	GhostMySelfieListActivity - Explained above UploadResultReceiver - it is in same class of GhostMySelfieListActivity. It is the Broadcast Receiver that registers itself to receive result from GhostMySelfieService.	registerReceiver() - ● Register a BroadcastReceiver that receives a result from the GhostMySelfieService when a selfie upload completes. onReceive() - Belongs to this UploadResultReceiver Class ● when a selfie is uploaded get the list of selfies updating the last one.

• **Service**

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.common;	Utils - ● Helper methods shared by various Activities.	checkConn(...) - ● Check if device has internet connection via System Service.
com.laishidua.model.mediator.webdata;	GhostMySelfieServiceProxy - ● This interface defines an API for a GhostMySelfie Service web service. The interface is used to provide a	

	contract for client/server interactions. The interface is annotated with Retrofit annotations to send Requests and automatically convert the GhostMySelfie.	
com.laishidua.model.services;	GhostMySelfieService - ● Intent Service that runs in background and executes various requests to the server. After the operation, it broadcasts the Intent to send the result of the request to the calling Activity.	makeUploadIntent(...) - ● Factory method that makes the explicit intent another Activity uses to call this Service for uploading a selfie. makeDownloadIntent(...) - ● Factory method that makes the explicit intent another Activity uses to call this Service for downloading a selfie. makeDeleteIntent(...) ● Factory method that makes the explicit intent another Activity uses to call this Service for delete a selfie. makeRateIntent(...) ● Factory method that makes the explicit intent another Activity uses to call this Service for rating a selfie. onHandleIntent(...) - ● Hook method that is invoked on the

		worker thread with a request to process. Only one Intent is processed at a time, but the processing happens on a worker thread that runs independently from other application logic.
com.laishidua.presenter;	GhostMySelfieOps - ● Explained before, the methods that start a service are listed in the next column and explained before.	uploadGhostMySelfie(...) getGhostMySelfieList() -

com.laishidua.presenter;	GhostMySelfieDetailOps -
	● Explained before, the methods that start a service are listed in the next column and explained before.
		downloadGhostMySelfie(..)
		
		deleteGhostMySelfie(...)
		
		rateGhostMySelfie(...)

- **ContentProvider**

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.model;	GhostMySelfieContract - ● This contract defines the metadata for the GhostMySelfieContentProvider, including	buildUriMatcher() - ● Helper method to match each URI to the ACRONYM integers constant definedin this class.

	the provider's access URIs and its "database" constants.	● All paths added to the UriMatcher have a corresponding code to return when a match is found. The code passed into the constructor represents the code to return for the rootURI. It's common to use NO_MATCH as the code for this case. ● For each type of URI that is added, a corresponding code is created.
com.laishidua.model;	GhostMySelfieDatabaseHelper - ● The database helper used by the GhostMySelfie Content Provider to create and manage its underling database.	GhostMySelfieDatabaseHelper(...) - ● Constructor - initialize database name and version, but don't actually construct the database (which is done in the onCreate() hook method). It places the database in the application's cache directory, which will be automatically cleaned up by Android if the device runs low on storage space. onCreate(...) - ● Hook method called when the database is created. ● Create the tables. onUpgrade(...) - ● Hook method called when the database

		is upgraded. ● Delete the existing tables. ● Create the new tables.
com.laishidua.model;	GhostMySelfieProvider - ● Content Provider interface used to manage GhostMySelfies. This class plays the role of the "Abstraction" in the Bridge pattern. It and the hierarchy it abstracts play the role of the "Model" in the Model-View-Presenter pattern.	insert(...) - ● Method called to handle insert requests from client applications. bulkInsert(...) - ● Method that handles bulk insert requests. query(...) - ● Method called to handle query requests from client applications. update(...) - ● Method called to handle update requests from client applications. delete(...) - ● Method called to handle delete requests from client applications.
com.laishidua.model;	GhostMySelfieProviderImpl - ● Content Provider implementation used to manage GhostMySelfies. This class plays the role of the "Implementor" in the Bridge pattern	query(...) - ● Method called to handle query requests from client applications. This method plays the role of the "template method" in the Template Method pattern.

	and the "Abstract Class" in the Template Method pattern.	● Match the id returned by UriMatcher to query appropriate rows. update(...) - ● Method called to handle update requests from client applications. This method plays the role of the "template method" in the Template Method pattern. ● Match the id returned by UriMatcher to update appropriate rows. delete(...) - ● Method called to handle delete requests from client applications. This method plays the role of the "template method" in the Template Method pattern. ● Try to match against the path in a url. It returns the code for the matched node (added using addURI) or -1 if there is no matched node. If a match is found delete the appropriate rows.
com.laishidua.model;	GhostMySelfieProviderImpl SQLite - ● Content Provider	insertGhostMySelfies(...) - ● Method called to handle insert

	implementation that uses SQLite to manage GhostMySelfies. This class plays the role of the "Concrete Implementor" in the Bridge pattern and the "Concrete Class" in the TemplateMethod pattern.	requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to table ghostmyselfie_table. bulkInsertGhostMySelfies(...) - ● Method that handles bulk insert requests. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to table ghostmyselfie_table. queryGhostMySelfies(...) - ● Method called to handle query requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to table ghostmyselfie_table. queryGhostMySelfie(...) - ● Method called to handle query requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to table

		ghostmyselfie_table. querySetting - ● Method called to handle query requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to settings_table. updateGhostMySelfies(...) - ● Method called to handle update requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to ghostmyselfie_table. updateGhostMySelfie(...) - ● Method called to handle update requests from client applications. This plays the role of the "concrete hook method" in the Template Method pattern. ● Related to ghostmyselfie_table.
com.laishidua.presenter;	GhostMySelfieOpsContentResolver - ● Class that uses a ContentResolver to insert, query, update, and delete	insert(...) - ● Insert ContentValues into the GhostMySelfieContentProvider at the

	selfies from the ContentProvider. This class plays the role of the "Concrete Implementor" in the Bridge pattern and the "Concrete Class" in the TemplateMethod pattern. It's also an example of the "External Polymorphism" pattern.	uri. Plays the role of an "concrete hook method" in the Template Method pattern. bulkInsert(...) - ● Insert an array of ContentValues into the GhostMySelfieContentProvider at the uri. Plays the role of an "concrete hook method" in the Template Method pattern. query(...) - ● Return a Cursor from a query on the GhostMySelfieContentProvider at the uri. Plays the role of an "concrete hook method" in the Template Method pattern. update(...) - ● Delete the selection and selectionArgs with the ContentValues in the GhostMySelfieContentProvider at the @a uri. Plays the role of an "concrete hook method" in the Template Method pattern. delete(...) - ● Delete the selection and selectionArgs from the

		GhostMySelfieContentProvider at the @a uri. Plays the role of an "concrete hook method" in the Template Method pattern.
com.laishidua.presenter;	SettingsOpsContentResolver - ● Class that uses a ContentResolver only to query and update from the ContentProvider to query or update the settings in database to modify selfies we send to server and other ones. This class plays the role of the "Concrete Implementor" in the Bridge pattern and the "Concrete Class" in the TemplateMethod pattern. It's also an example of the "External Polymorphism" pattern.	insert(...) - ● Insert ContentValues into the GhostMySelfieContentProvider at the uri. Plays the role of an "concrete hook method" in the Template Method pattern. bulkInsert(...) - ● Insert an array of ContentValues into the GhostMySelfieContentProvider at the uri. Plays the role of an "concrete hook method" in the Template Method pattern. query(...) - ● Return a Cursor from a query on the GhostMySelfieContentProvider at the uri. Plays the role of an "concrete hook method" in the Template Method pattern. update(...) - ● Delete the selection and selectionArgs

		with the ContentValues in the GhostMySelfieContentProvider at the @a uri. Plays the role of an "concrete hook method" in the Template Method pattern. delete(...) - ● Delete the selection and selectionArgs from the GhostMySelfieContentProvider at the @a uri. Plays the role of an "concrete hook method" in the Template Method pattern.

App interacts with at least one remotely-hosted Java Spring-based service

Explained in the other PDF.

App interacts with at least one remotely-hosted Java Spring-based service

Explained in the other PDF.

App interacts over the network via HTTP/HTTPS.

Explained in the other PDF.

App allows users to navigate between 3 or more user interface screens at runtime

Explained in the other PDF.

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.**

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	GhostMySelfieListActivity - ● Explained before.	onGhostMySelfieSelected(...) - ● Create an intent that will start an Activity to get GhostMySelfie from Gallery. ● Create an intent that will start an Activity, this Intent action is sent to have the camera application capture an image and return it.

App supports at least one operation that is performed off the UI Thread in one or more background Threads of Thread pool.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.model.mediator;	GhostMySelfieDataMediator - ● Mediates communication between the GhostMySelfie Service and the local storage on the Android device. The methods in this class block, so some of this methods are called from a background thread via an AsyncTask and other from service. See an example in com.laishidua.presenter.GhostMySelfieOps lines from 156 to 211.	GhostMySelfieDataMediator() - ● Initialize the GhostMySelfieServiceProxy. uploadGhostMySelfie(...) - ● Uploads a Selfie having the given Uri. This Uri is the Uri of image Selfie in Android GhostMySelfie Content Provider. downloadGhostMySelfie(...) - ● Downloads a Selfie having the given Id and title. deleteGhostMySelfie(...) - ● Deletes a Selfie having a given Id from server rateGhostMySelfie(...) - ● Rates a Selfie having a given Id to the server.

Functional Description and App Requirement.

App allows user to take and save a Selfie.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	GhostMySelfieListActivity - ● Explained above	onGhostMySelfieSelected(...) - ● Create an intent that will start an Activity to get GhostMySelfie from Gallery. ● Create an intent that will start an Activity, this Intent action is sent to have the camera application capture an image and return it.

App reminds user to take a Selfie.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.utils;	AlarmNotificationReceiver - ● Extends the BroadcastReceiver to get notifications to take a selfie in certain time intervals with some scary sound.	onReceive(...) - ● Build the Notification. ● Pass the Notification to the NotificationManager.
com.laishidua.view;	SettingsActivity - ● Explained before.	addListenerOnReminderTime() - ● Set a single Alarm.

App allows user to view saved Selfies in a List View.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	GhostMySelfieListActivity - ● Explained before.	onResume(..) - ● Call the list of Selfies.
com.laishidua.presenter;	GhostMySelfieOps	getGhostMySelfieList() - ● Gets the SelfieList from Server by executing the AsyncTask without blocking the caller.

If the User closes and then reopens the app, the user has access to all saved Selfies.

MORE RELATED CLASSES AND METHODS FROM SERVER		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.mobilecloud.ghostmyselfie.controller;	GhostMySelfieController	saveSomeGhostMySelfie(...) - ● Save the selfieimage data that belongs to a user in MySQL. addGhostMySelfie(...) ● Save the data of a selfie that belongs to a user in MySQL.

The App allows the User to select one or more Selfies, select one or more graphic effects, and apply the selected effects to the selected Selfies, and then view the processed Selfies in a ListView.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	SettingsActivity - ● Activity to select the Selfies.	onCreate(...) - ● Set the listener for each Selfie. ● Settings are saved in the database device.
com.laishidua.model.mediator;	GhostMySelfieDataMediator - ● Explained before.	uploadGhostMySelfie(...) - ● lines 165 to 183.

MORE RELATED CLASSES AND METHODS FROM SERVER		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.mobilecloud.ghostmyselfie.controller;	GhostMySelfieFileManager - ● This class provides a simple implementation to store selfie binary data on the file system in a "ghostmyselfies" folder. The class provides methods for saving selfies and retrieving their binary data.	saveGhostMySelfieData(...) - ● This method reads all of the data in the provided InputStream and stores it on the file system. The data is associated with the GhostMySelfie object that is provided by the caller. ● This method saves or replace a selfie applying all the

		selected filters. overlayImage(...) - ● Overlay a ghost in the image selfie. grayFilter(...) - ● Apply the gray scale filter to the image. blurFilter(...) - ● Apply blur filter to the image darkFilter(...) - ● Apply dark filter to the selfie.

Some part of the operations corresponding to Requirement 5 (Applying Graphics Effects to Process Images) must be executed concurrently in a remote web service. Once processed, the images are returned to the device, where they are displayed in a ListView.

MORE RELATED CLASSES AND METHODS FROM CLIENT		
PACKAGE	**CLASS**	**METHODS**
com.laishidua.view;	GhostMySelfieListActivity - ● Explained before.	onCreate(...) - ● mModifierButton triggers concurrent execution to applying effects to selected selfies. Lines 138 - 155.