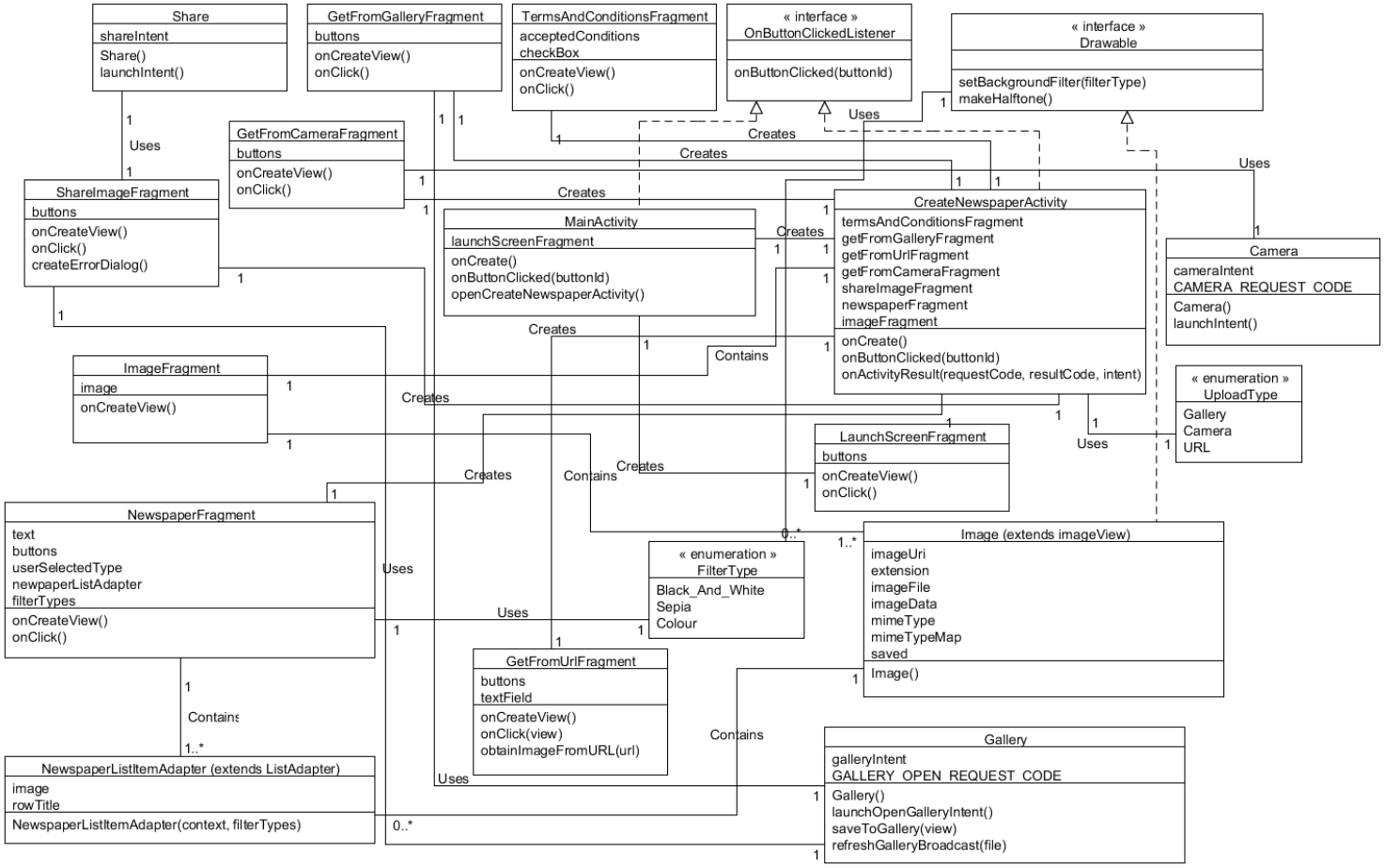
Conceptual UML Class Diagram

Provided below is the conceptual UML class diagram for the Ye Olde Times Android mobile application. The diagram shows all of the entities within the system in addition to the attributes belonging to those entities and the tasks that the entities are responsible for performing. Below the diagram is a comprehensive breakdown explaining all of the entities within the system and how they relate to one another. It is important to note when viewing the below conceptual UML class diagram that the accessors and mutators for each of the attributes in each class have been omitted for the sake of simplicity however they will be included in the final implementation of the application. Where an inheritance relationship could not be modelled in the diagram due to a class inheriting from another class belonging to the Android API, the inheriting class details the class that it extends in its class name (e.g. Image (extends imageView)). As the names of the entities within the diagram have been concatenated into a single word following class naming conventions, the names of the classes have been separated for easier readability in the explanation below (for example the "ImageFragment" class is described below as the "Image Fragment" class).



Main Activity

The Main Activity is the main entry-point of the application. It is responsible for calling the onCreate method which sets up the Launch Screen Fragment. It also contains 1 variable which corresponds to the instance of the Launch Screen Fragment which is created by the Main Activity. It also has an onButtonClicked method which is implemented from the On Button Clicked interface. This allows the clicks performed in the Launch Screen Fragment to be managed by the Main Activity in the situation that a button clicked launches the Create Newspaper Activity. Finally it contains an openCreateNewspaperActivity method which opens the Create Newspaper Activity. Thus, the Main Activity has an association relationship with the Create Newspaper Activity class, Launch Screen Fragment class and implements the OnButtonClickedListener interface.

Terms and Conditions Fragment

The Terms and Conditions Fragment contains two attributes: acceptedConditions and checkbox. AcceptedConditions is an attribute which records whether the user has accepted the terms and conditions statement. The checkbox attribute is an actual checkbox which is displayed to the screen for the user to select. The Terms and Conditions Fragment also contains an onCreateView and an onClick method. The onClick method is called on the click of the checkbox while the onCreateView method is called on creation. The fragment has one relationship to the Create Newspaper Activity entity because it is created by this activity.

Get From Gallery Fragment

The Get From Gallery Fragment has one attribute called "buttons" which contains all of the buttons on the screen. The onClick method inside of the entity handles all of the functionality to perform when these buttons are clicked. There is also an onCreateView method which is called on instantiation of the fragment. The Get From Gallery Fragment has an association relationship with the Create Newspaper Activity entity because it is created by the Create Newspaper Activity instance. It also has an association relationship with the Gallery class because it calls the openGallery intent in order to obtain an image from the gallery to pass back to the Create Newspaper Activity for display in another fragment.

Create Newspaper Activity

The Create Newspaper Activity class contains 7 fragments, 6 of which are fragments used to control and manipulate the image fragment and 1 image fragment which is consistent across all screens (except for the terms and conditions screen). The 6 fragments each display either the terms and conditions screen, controls to get an image from the gallery, from a URL, from the camera, to convert the image to a halftone newspaper style image or to share the image. Thus it has association relationships with all of these fragments' classes, and with the Main Activity as it is created by the Main Activity. Finally, it implements the OnButtonClickedListener interface in order to handle the button clicks performed in other fragments, should they launch intents that need to be handled by the Create Newspaper Activity. It also has two methods, onButtonClick which handles the clicks performed in the fragments created by the Create Newspaper Activity and an onActivityResult method which handles the results of intents called from fragments within the Create Newspaper Activity.

Share Image Fragment

The Share Image Fragment contains a single "buttons" attribute which contains all of the buttons on the screen. The onClick method inside of the entity handles all of the functionality to perform when these buttons are clicked. There is also an onCreateView method which is called when the fragment is created. The buttons on this screen allow the user to select a social networking service or email to share their image with. There is also a createErrorDialog method which allows for the creation of several error dialogs including one to deal with the situation where the user has not saved their image and another which alerts the user if they cannot share with a particular social networking service because they do not have it installed on their device. The Share Image Fragment contains 3 relationships: one with the Share entity, one with the Gallery entity and one with the Create Newspaper Activity entity. The Share entity is used by the Share Image Fragment to create a share intent for the image. The Share Image Fragment also requires access to the saveToGallery method within the Gallery class and thus has an association relationship with the Gallery class. Finally, it has an association relationship with the Create Newspaper Activity entity because it creates the Share Image Fragment.

Image Fragment

The Image Fragment contains an image attribute which is an instance of the Image class (which is essentially an Image View with some additional attributes and methods). It also contains 2 relationships to other classes in the diagram. It has an association relationship with the Image class because it contains an instance of the Image class and has an association relationship with the Create Newspaper Activity because it is created by this activity.

Newspaper Fragment

The Newspaper Fragment contains a text attribute which contains the text that the user wants to put into their 19th century style newspaper article and the buttons used on the screen in addition to an onClick method which handles what occurs when those buttons are clicked. It also has an attribute called userSelectedType which is the type of newspaper layout selected by the user. Additionally, it contains a newspaperListAdapter which is the adapter for the rows in the list view displayed in the fragment. The final attribute that it contains is a series of filterTypes which are able to be selected by the user to colour their newspaper. Additionally, it contains an onCreateView method which is called when the fragment is created. The Newspaper Fragment class contains 3 relationships. There is a relationship between the Newspaper Fragment class and the Newspaper List Item Adapter class because the Newspaper Fragment makes use of the list item adapter to display the different types of layouts selectable by the user. It also has a relationship with the Filter Type enumeration because it contains a series of Filter Types which denote the types of filters that the user should be able to select. Finally, the Newspaper Fragment has a relationship with the Create Newspaper Activity because it is created by this activity.

Get From URL Fragment

The Get From URL Fragment has a textField and buttons attribute. The buttons encompass all of the buttons in the fragment meanwhile the textField is the field used by the user to input a URL to an image online. The entity also contains an onCreateView, obtainImageFromURL and an onClick method. The onCreateView and onClick methods function in the same manner as they do in other fragments. The obtainImageFromURL method takes a URL, obtains the image from it and passes it to the Create Newspaper Activity such that it can be displayed.

Get From Camera Fragment

The Get From Camera Fragment has a buttons attribute, an onCreateView method and an onClick method. The buttons attribute, alike in other fragments, contains all of the buttons within the fragment. Meanwhile the onCreateView method is called on creation of the fragment and sets up all of the buttons to be handled in the onClick method. It has a relationship to the Create Newspaper Activity because it creates the fragment and a relationship with the Camera class because it requires an instance of the class in order to create the camera intent.

Launch Screen Fragment

The Launch Screen Fragment has some buttons, an onCreateView method and an onClick method which determines what to do when the buttons on the screen are clicked. The Launch Screen Fragment is displayed upon the user first opening the app and serves as the "home screen" of the application. It has a single relationship to the Main Activity class as the Launch Screen Fragment is created by this activity.

Newspaper List Item Adapter

The Newspaper List Item Adapter is an adapter that extends from the base adapter for list items known as the List Adapter. It has a single attribute that is an Image (image view) which displays the appropriate layout type for the newspaper. It also contains a rowTitle for the list item row that labels the image in the row. It is used by the Newspaper Fragment to display the different types of newspaper filters that can be applied to the user's image. The Newspaper List Item Adapter extends from the List Adapter class however, as it is an Android API class, it has not been included in the diagram.

Camera

The Camera class contains a cameraIntent and a CAMERA_REUQEST_CODE attribute. The cameraIntent contains the intent to launch the camera and obtain image data once the camera intent has finished. Meanwhile the CAMERA_REUQEST_CODE is used to process what is returned by the camera intent in the onActivityResult method within the activity that receives the result of the camera intent call. It also contains a constructor and a launchIntent method which launches the camera intent. It contains a single relationship to the Get From Camera Fragment class as it allows the user to obtain an image from their camera.

Gallery

The Gallery class contains the galleryIntent and GALLERY_OPEN_REQUEST_CODE. The galleryIntent contains the intent used to open the gallery while the GALLERY_OPEN_REQUEST_CODE contains the request code used to open the gallery intent. The Gallery class also has a constructor and methods to launch the gallery intent to open the gallery (launchOpenGalleryIntent), save an image to the gallery (saveToGallery), and send a broadcast to refresh the gallery (refreshGalleryBroadcast). It has a relationship to the Get From Gallery Fragment class because the fragment requires access to the gallery to load an image. It also has a relationship to the Share Fragment class as Share Fragment requires access to the saveToGallery method to save the image back into the gallery after it has been manipulated.

Image

The image class contains the imageUri (a Uri which points to the image's location), an extension for the type of image that is being displayed, an imageFile containing the original image loaded in, the imageData containing the data of the image in bytes, the mimeType (a further type which is used in the share intent to determine the type of image being shared), a mimeTypeMap used by the share intent and finally, a saved flag which determines whether the image has been saved or not. As the Image class is essentially an extension on the Android API's Image View class, it is used to display images in the application within the Newspaper List Item Adapter and the Item Fragment. Thus, the Image class has association relationships to the Newspaper List Item Adapter and Item Fragment classes.

Share

The share entity contains a shareIntent attribute which is the share intent that can be called upon to share an image from another activity or fragment. It also contains a launchIntent method which launches the intent. It has a single relationship with the Share Image Fragment class as this fragment is the sole fragment that requires the share intent.

Drawable

The Drawable interface contains a setBackgroundFilter method which sets the background filter of an image to a specific filter. It also contains a makeHalftone method which would make a given image into a halftone image. As many different types of halftone images could be included in future iterations of the application, the Drawable interface was determined to be necessary to ensure that these additional types of half-toning techniques could be easily included by implementing the Drawable interface in other image view classes. It has a relationship to the Filter Type Enumeration because it makes use of the enumeration in determining which filter was passed into its setBackgroundFilter method.

Filter Type Enumeration

The Filter Type Enumeration is an enumeration which includes the different types of filters available for the user to select from. It is used by both the Newspaper Fragment class and the Drawable interface and thus, this is why relationships are shown between the Filter Type Enumeration and these two entities.

Upload Type Enumeration

The Upload Type Enumeration is an enumeration which includes the different types of uploading methods (Gallery, URL and Camera). It is used by the Create Newspaper Activity to determine which fragment to launch based on the button clicked corresponding to opening the Gallery fragment, URL fragment or the Camera fragment.

Domain Dictionary

Described below is the Domain Dictionary that has been included to explain all of the terms specific to the design of the Ye Olde Times application. Additional terms have been included that may relate to more technical concepts referred to in the design.

Activity: A single view/ window within the application. They usually interact with the user and comprise of a user interface. The user interface may comprise of fragments to display certain components of the user interface or may contain raw widgets.

Adapter: In the context of Android mobile application development, an Adapter provides access to data items within a view.

API: Application Programming Interface. An API specifies how different components of a software application interact with one another.

Broadcast: A broadcast is a call sent out to all applications interested in listening to the broadcast. A broadcast will sometimes ask a specific application to perform a specific action upon receiving a broadcast, however it can also broadcast to several applications in the situation that several applications must perform functionality on receiving the broadcast.

Class: An entity within the system.

Entity: A term used interchangeably with the term "class" to mean a component of the system, represented as a rectangle with a name, list of attributes and list of methods as displayed in the above conceptual UML class diagram.

Enumeration: A collection of items which fully describes the collection in which it resides. For example, an enumeration could be the primary colours, which would comprise of red, green and blue.

Filter: A colour overlay to be placed on top of the newspaper to make it appear within a certain style (colour, black and white or sepia).

Fragment: A piece of the application's user interface that can be placed within an Activity. It is a component that can comprise an entire screen or just a section of a screen that is used over and over again in different activities.

Halftone: A method used to turn detailed photographs into images that would look similar to printouts in a newspaper. This is achieved through printing a dot for each group of pixels in an image that is sized according to the average level of grey across those pixels.

Image View: A view in which images can be placed. An image view allows a user to see an image on their device's screen.

Interface: An entity which comprises of several methods which must be implemented by any class which implements the given interface. The purpose of having an interface is to ensure that many different classes can have different implementations of particular methods. It also ensures that every class that implements the interface must implement the given methods in some manner.

Intent: A description of an operation to be performed. An intent can open an activity or it can launch other applications on the user's device (for example: the camera application or the gallery application).

List Item Adapter: An adapter created in order to provide a custom layout for a list item within a standard list.

Method: An activity or responsibility of a particular class. Methods provide functionality that classes are responsible for performing.

Request Code: A code sent by an intent in order to be tracked once the intent finishes. It allows for data returned from the intent to be obtained upon checking for the request code.

Sharing: Sharing refers to providing data to another application. In the context of the Ye Olde Times Android mobile application, sharing refers to sharing images with social networking applications in addition to email on the user's device.

UML Diagram: Unified Modelling Language Diagram. A UML diagram describes all of the entities in the system and how they are related. The UML diagram provided in this document is a conceptual UML diagram which means that specific details pertaining to the implementation of the application have been omitted (such as the types of attributes in each entity, the types of parameters passed in to method calls and the return types of method calls).

Uri: Uniform Resource Identifier. A Uri is an address to points to some data.

URL: Uniform Resource Locator. A URL is an address that points to a website.

Widget: A component of the user interface that allows the user to view content or interact with content. Some examples of widgets include buttons and image views.