Below is a discussion of six design features including a possible context of use, reason for choosing the particular feature and justification why the feature was designed to look and feel the way it does.

Design Feature: Filter 1

Context: The user syncs images spanning a couple of years say 2007 - 2013. The user wants to quickly view photos from a wedding or birthday party taken in 2009. He can choose to scroll through all the million pictures, which is costly in terms of time, processing power and battery life not mention the tediousness of it all.

Inclusion Reasoning: Allows the user to swap between the various views/ arrangements of images offered by the image viewer. This feature offers capabilities such as arranging images in a grid view, film strip view, user defined tags, date and time taken among others, inherently allowing the user to browse through lots of data on a small screen.

Design Justification: This feature has been implemented as a drop down menu situated on the Main Action Bar. Selecting the feature reveals a list of items for the user to choose from. Not only does this save space but also allows for a clean uncluttered interface whilst limiting the number of options offered to the user. Visual affordance was also taken into account as seen in the corresponding figures.

Design Feature: Zoom and Fast Scroll 4

Context: The user is on a battlefield with an Ariel unmanned vehicle capturing enemy location pictures and linking them to the image viewer in bulk per second. User needs to find and identify enemy targets within seconds.

Inclusion Reasoning: The Zoom and Scroll features work hand in hand and allow the user to peruse through numerous images at a high speed whilst being able to pause and zoom to further examine any identified pictures in more detail.

Design Justification: This feature has been implemented using touch gesture controls. A flick gesture in the direction you want to scroll is used for normal scrolling and quickly brushing fingertip in the intended direction activates a faster scroll. The Zoom feature is activated by a double tap and further zooming is achieved by a finger spread gesture to zoom in or pinch to zoom out. The technological advancement of mobile devices has done away with scroll bars especially with touch devices allowing for a clean user interface and a more intuitive user interaction experience.

Design Feature: Share 2

Context: The user has numerous devices and wants to sync his photos across all devices or the user wants to upload images to numerous social sites.

Inclusion Reasoning: There is a limit to the number of devices one can carry. This feature offers the option of sharing all your images across a specified number of devices through the Bluetooth mobile capability and also providing the user with feedback on completion of a given action. Additionally it

allows for backing up of images to an online drive such as drop box for safe storage and uploading to various social networks via Wi-Fi or mobile data.

Design Justification: This feature has been implemented as a drop down menu on the Main Action Bar. A drop down menu allows for a list of items to display without cluttering the small mobile screen avoiding user errors. The user makes a tap gesture to select the share feature revealing a controlled list of share options such as Bluetooth, Google+, Facebook, Google drive, drop box, including a print option to send any selected image for printout. Mobile devices no longer work in isolation, and the ability to collaborate with neighboring devices is one that cannot be ignored.

Design Feature: Delete 5

Context: The user is scrolling fast through the images and wants to delete multiple images quickly.

Inclusion Reasoning: Although storage capacity of devices has greatly improved of recent, there comes a time when you need to create space for different reasons. The advancement in mobile technology has allowed for this feature to be implemented via figure gesture, making it intuitive and easy to implement whilst also saving screen space.

Design Justification: This feature has been implemented as a touch gesture and it's activated by a hold and drag finger gesture. The user holds and drags the image of the screen in essence deleting it. This is not only intuitive but also easy to implement.

Design Feature: Edit 3

Scenario: The user wants to add fancy effects, resize, rotate and adjust image brightness. Additionally the user might want to tag or caption a given image or even modify security measures such as image hiding to prevent access by any other device or image viewer thereof.

Inclusion Reasoning: Being able to manage images on the go can come in handy especially when you are on the move and have no access to a computer. Additionally the introduction of apps like instagram has led to a trend of adding effects and uploading the adjusted image within minutes if taking it. This feature allows for keeping up with social trends as well as offering an image editing ability that rivals professional image editing software such as Photoshop.

Design Justification: This feature has been implemented as an image button. It only appears when a given image has been explicitly selected by a single tap gesture. The image on the button allows for a single interpretation of what the button does further emphasizing visual affordance. The characteristic to hide and show the image button is easy to implement and saves screen space as it only shows when needed. Selecting the image button reveals imaging editing assets such as crop among others. This feature also allows for recoverability by offering an undo option.

Design Feature: Multi-Select 6

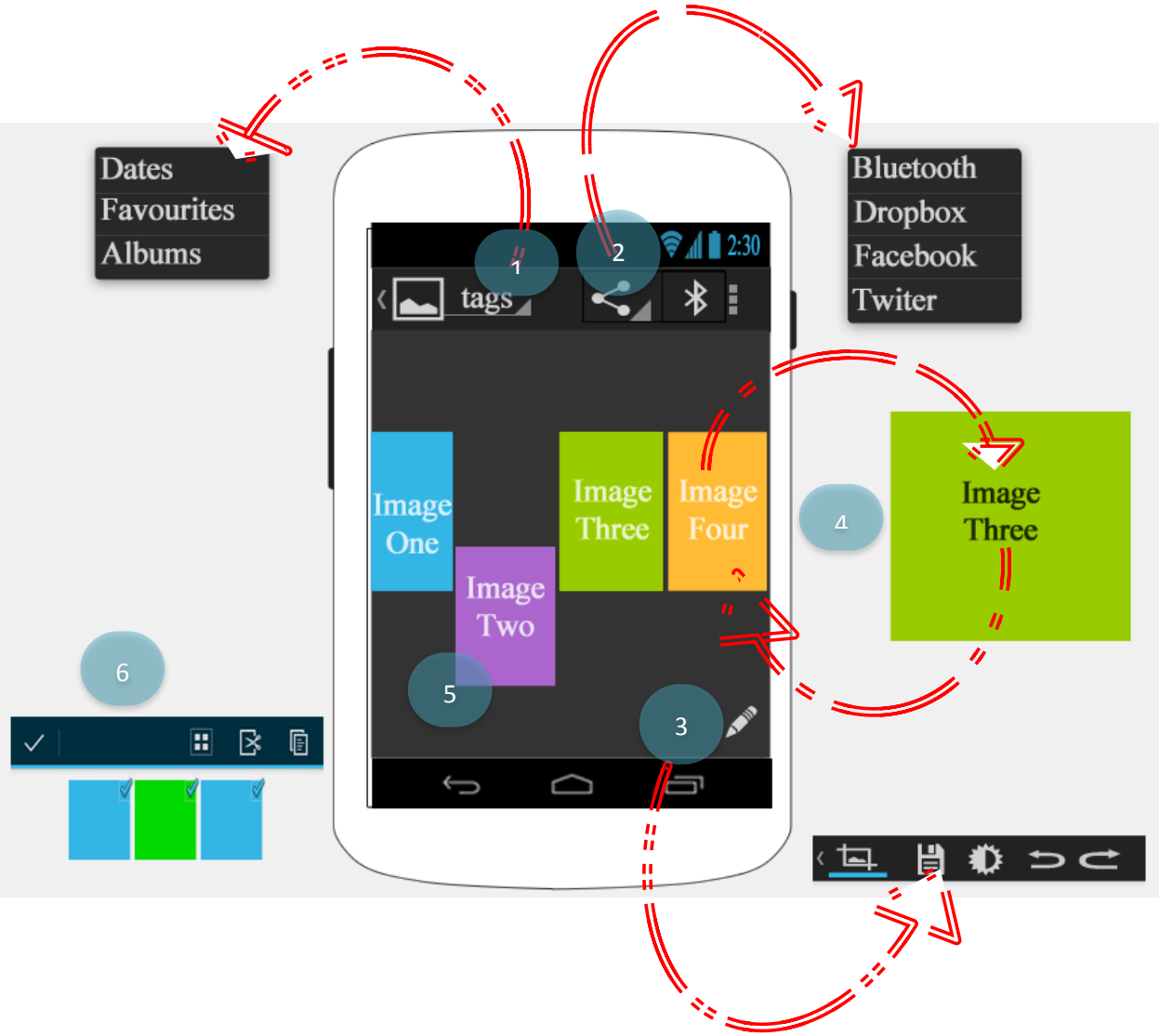
Scenario: The user wants to copy, cut, rename multiple images or even delete multiple images at the same time.

Inclusion Reasoning: The ability to select multiple images is sure to come in handy. Cases when the user needs to copy or cut multiple images to a different folder or even rename of check properties can be handled via multi select.

Design Justification: This feature has been implemented to handle both single and multi-select via a press touch gesture. The user presses a single image for a prolonged time to reveal check boxes and a list of options such as edit, cut on a split action bar as shown in the prototype. The images checked are then selected and manipulated as desired. This feature is not only elegant but also prevents control from overwhelming functionality allowing for minimal surprise.

It is important to note that since main functionality of the image viewer is to display pictures; all the design features discussed above have been designed to surround the app content area leaving a clean and an uncluttered user interface allowing for improved learnability and speed of operation.

An annotated initial design showing each design feature discussed above with a brief explanation supported by touch gesture commands. Fig1

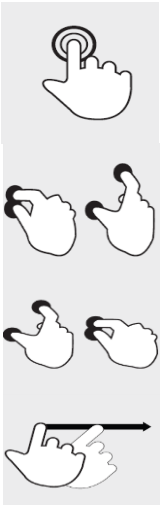


1 **Filter Feature:** The filter feature is activated by tap touch gesture (analogous to a single click). Once selected, a drop down menu with options similar to the ones pointed to by the arrow at feature 1 is displayed so the user can chose a filter criterion.

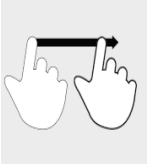
2 **Share Feature:** The share feature is also activated via tap touch gesture and once selected reveals a drop down menu with share options for the user to choose from.

3 **Edit Feature:** This feature is activated by via tap touch gesture. The user taps the image button with the pencil icon to reveal the options pointed to by the arrow. These options are displayed on the split action bar aligned at the bottom below the app content area.

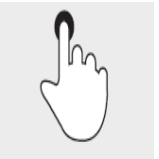
4 **Zoom and Scroll Feature:** As stated before, the zoom and scroll feature work hand in hand. To scroll to right, the user swipes in the opposite direction. This is known as a flick touch gesture to the left and the faster the swiping the faster the scrolling. To activate the zoom feature, the user "double taps" (analogous to double clicking) or uses a spread or pinch gesture to zoom in and out respectively. The arrows at feature 4 in the initial design correspond to a spread and pinch gesture with the pinch gesture returning the user to the scroll mode as indicated by the return arrow.



5 **Delete Feature:** This feature is activated via a drag touch gesture. The user places a finger on the desired image and moves the image of the screen without losing contact. Image two in the initial design is shown in the process of being dragged off the screen consequently deleting it.



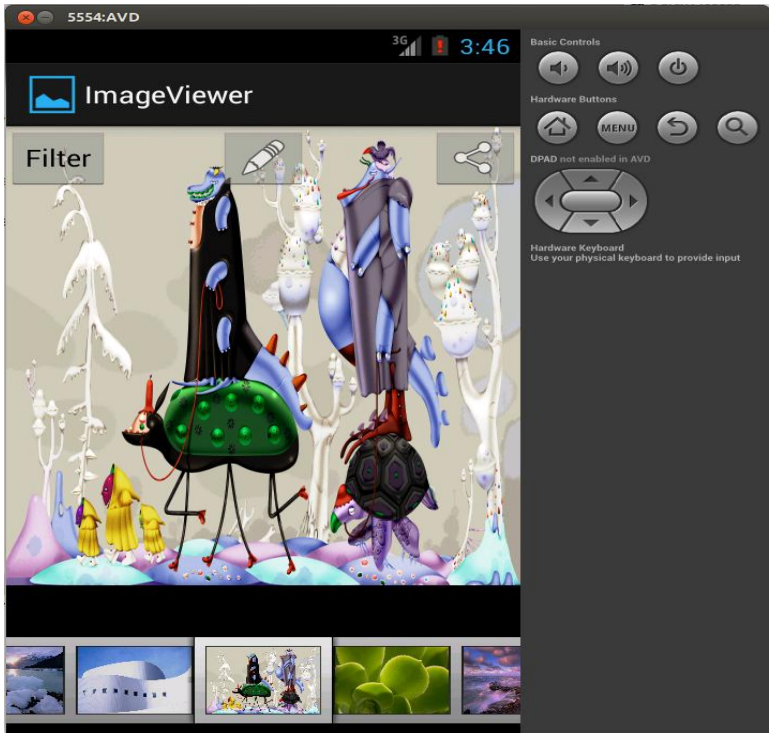
6 **Multiple-Select Feature:** This feature is activated via a touch and hold gesture to reveal the select options on offer and allow images to be selected in bulk as shown in the initial design. All images have a check box with the checked images having been selected. The select options a displayed in the main action bar above the content area.



The screenshots below show XML widget layouts portraying the functionality of the Image Viewer (minimal functionality is illustrated).

It is important to note that due the beginner nature of my skill level the eclipse xml layout looks somewhat very different from the prototype as some features were harder to implement than initially thought. The widget layout screen shots are displayed in a storyboard format.

Fig2.



- The Filter, edit, share buttons are hidden when the app loads. Every time the user taps on the picture the above mentioned buttons show.
- Selecting the **Filter or Share** image button reveals a drop down menu list of options as shown in **Fig2.1** and **Fig2.2** respectively.
- The user selects **grid view** to change the arrangement of images as shown in **Fig2.3**
- Flick (swipe)images shown in the filmstrip to the left to **scroll**

Fig2.1

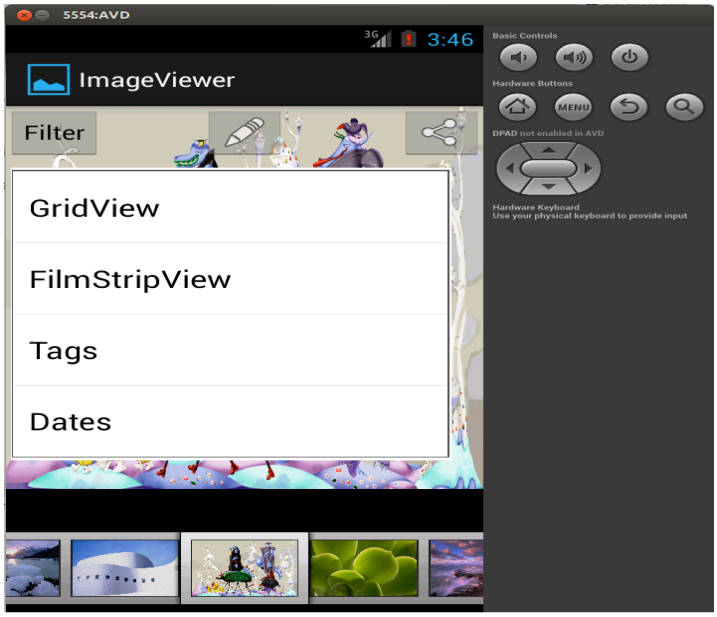


Fig2.2

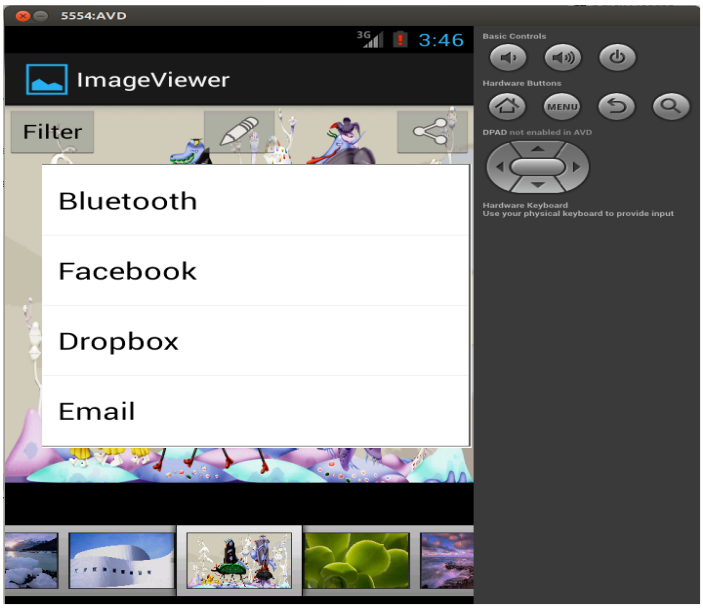
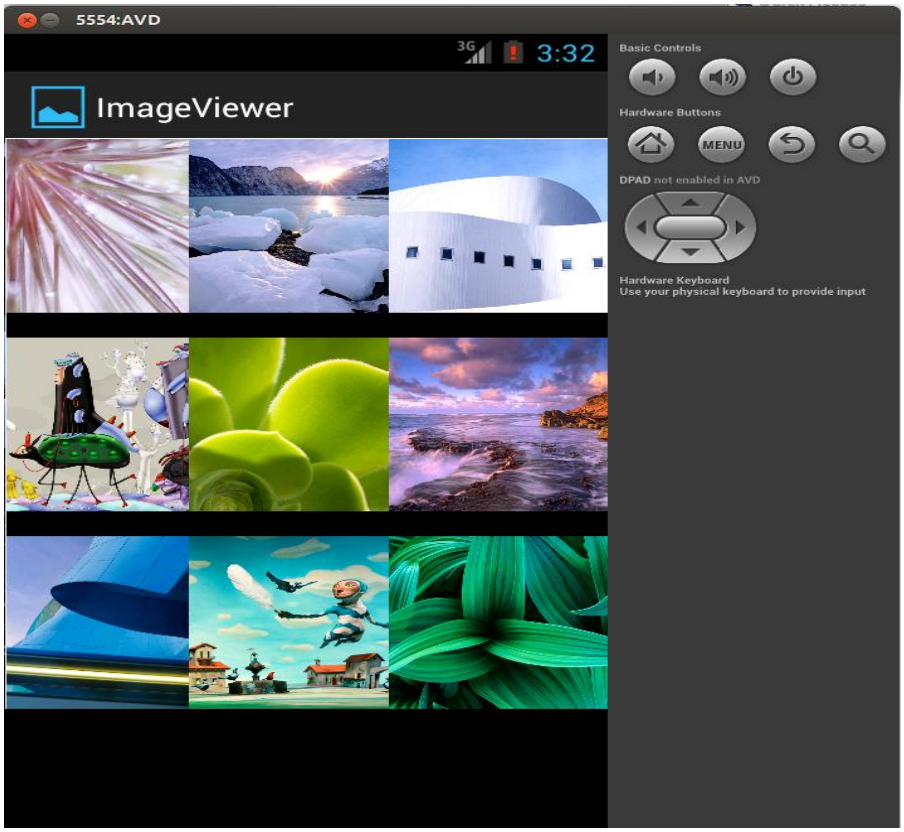


Fig2.3



- Double tap any image shown in content area to **zoom** and use pinch or spread touch gesture to achieve zoom effect as shown in **Fig2.4**
- To **edit** tap on the pencil icon in **Fig2** to reveal edit options as shown in **Fig2.5**

Fig2.4

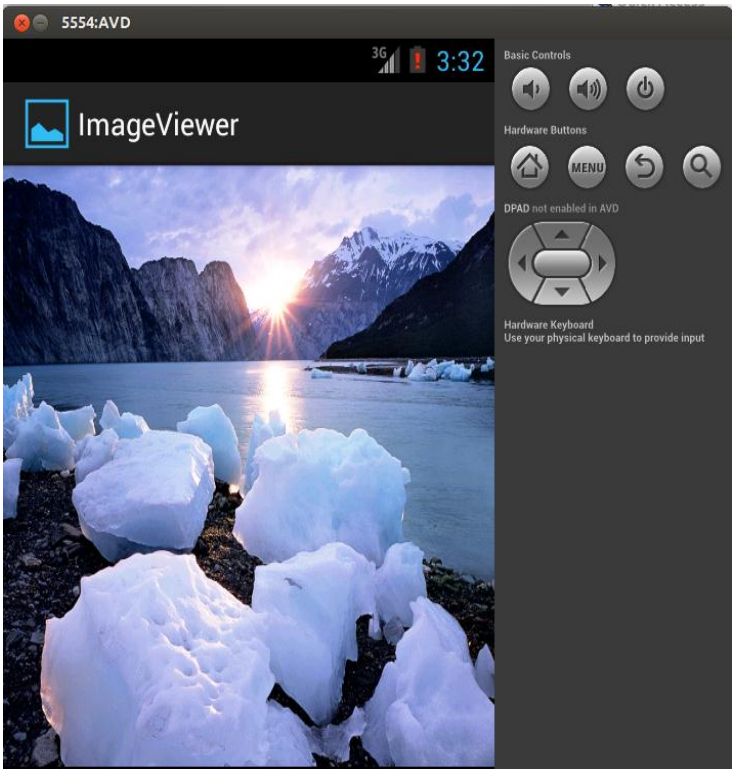


Fig2.5