Alphabet Book Application Design

Good application design keep things in the world rather than in the head. In mobile application recognition over recall criteria is said to be a good approach in designing good applications.

The alphabet book application is designed to assist young individuals master the alphabets, concurrently providing the user best utility. The application allows both standard and non-standard users of smart phone applications to be able to use it. Features on the design are designed in such a way that less time is taken by the user to complete a task. For instance, most of the features like button size and position are designed in the way that it will be easy for a user to recognize that it is a button, and according to Fitt's law; time taken to complete a task depends on the distance and size of the target. My design follows Fitt's law, Principle of least effort to name a few.

Features used in this regard include splash screen with transition animation and progress bar, image view preference with aid of raised buttons, colour change on button click, image switcher with action detection on touch and feedback messages i.e. toast messages.

(i) Feature inclusion and usability

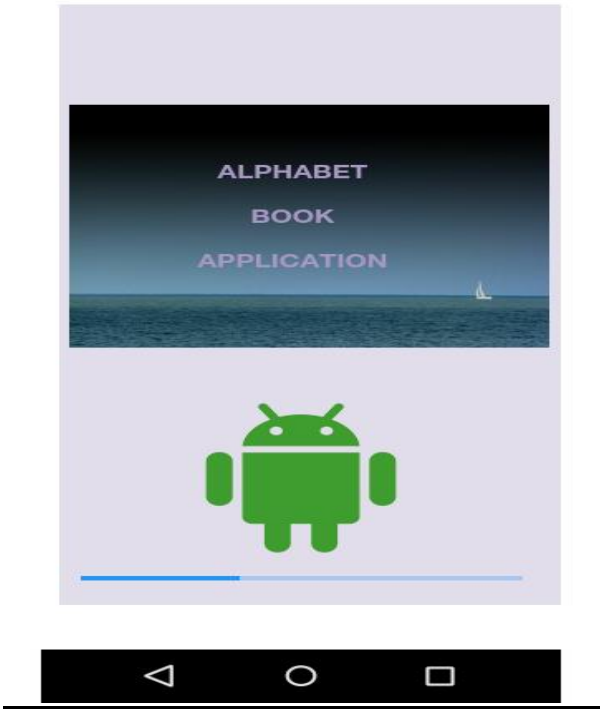

Figure 1: Splash screen with transition animation and progress

- Figure 1.1 shows the welcoming screen of the application when it is initially launched. The feature uses a progress bar which is timed by the thread in which the thread sleeps for specified time period, after the thread has timed out the welcoming screen disappears, taking the user to the main page of the application. The Splash screen is relevant

as it provide feedback that a process is underway, basically it notifies the user that the program is in the process of loading, the use of progress bar provides visibility such that the user will see the rate at which the application is progressing to launch.

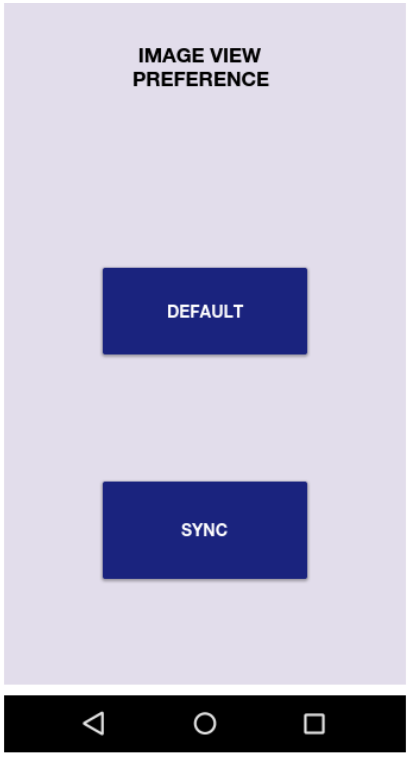

Figure 2.1: Image view preference page with aid of raised buttons

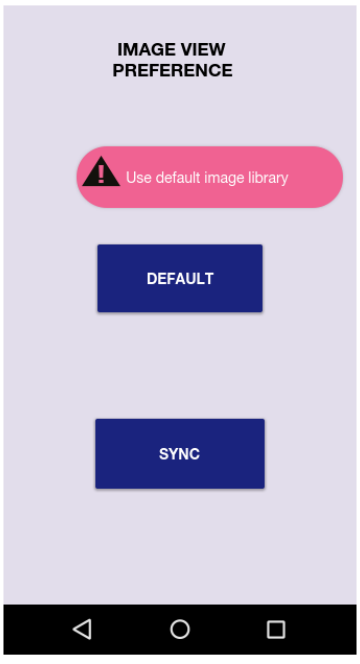

Figure 2.2: *Image view preference page with aid of raised buttons and toast message for default button.*

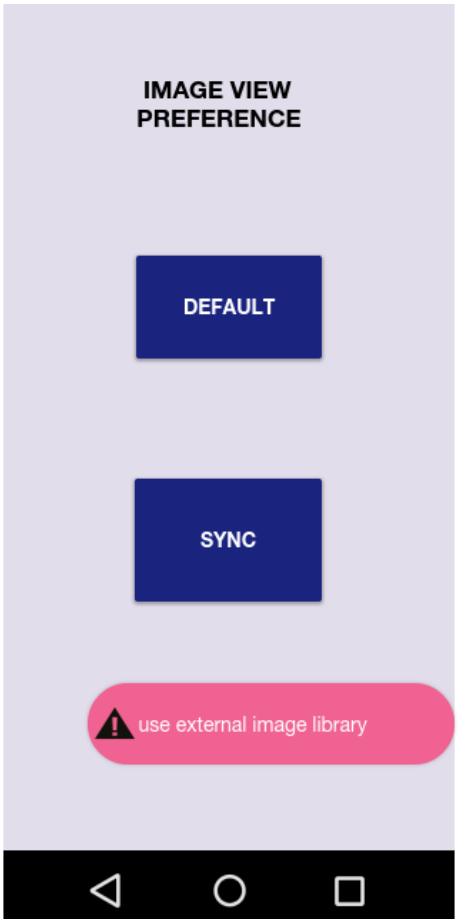

Figure 2.3: *Image view preference page with aid of raised buttons and toast message for default button.*

- Figure 2.1 illustrates image view preference page without any action. The preference page enable user control and freedom over the app. The aid of raised button is relevant as it serves as an affordance and enlightens the user that the feature must be clicked. The use of raised buttons is a good way to raise attention the button. It has a visible boarder and leaves a layer above the content (Google Developers, 2016). In general prominent buttons differentiate between information availability and action availability.
- Figure 2.2 and 2.3 both illustrates the use of pop up notification: - toast message feedback. The user is not necessary required to click the button in order for the notification to appear on screen. The way in which it is displayed is when the user places a pointer on top of the button, if using a pc, laptop, emulator etc. Similarly the user can gently place a finger on top of button, if using devices like mobile phone, tablet. Toast message feedback play a vital role in enlightening the user about the implications of the action, as notifications are

prominent in application design at large especially when a non-standard user uses the application. It fades in and out depending on the user action. Basically they are used when the application is inactive. They are mainly used for short message texts.

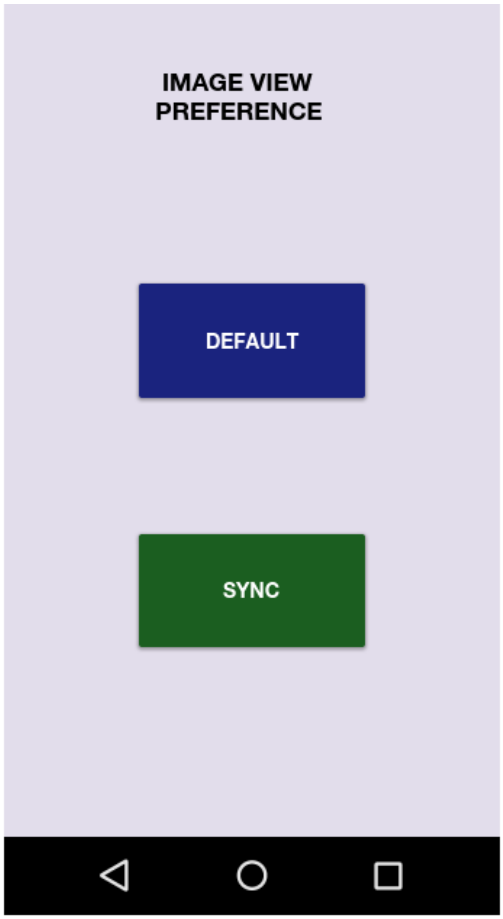

Figure 3: colour change on button click.

- Figure 3 illustrates image view preference with feedback on action performed. When the user clicks the default button, the application will switch to the overview page with default features and uses the default image library and settings.
- The user might wish to use their personal images and customize the app. The user will feel in control when they have the ability to do so. The SYNC button allows the user to import images from external storage/library, i.e. from the web, their personal phone or even take new pictures, in which where the app uses the default camera of the device, given it exists.
- When the user clicks the button it changes it current colour to green, giving feedback to the user that their action was a success.

After button click the app switches to the overview page, where everything in the user interface appears according to the user preference.

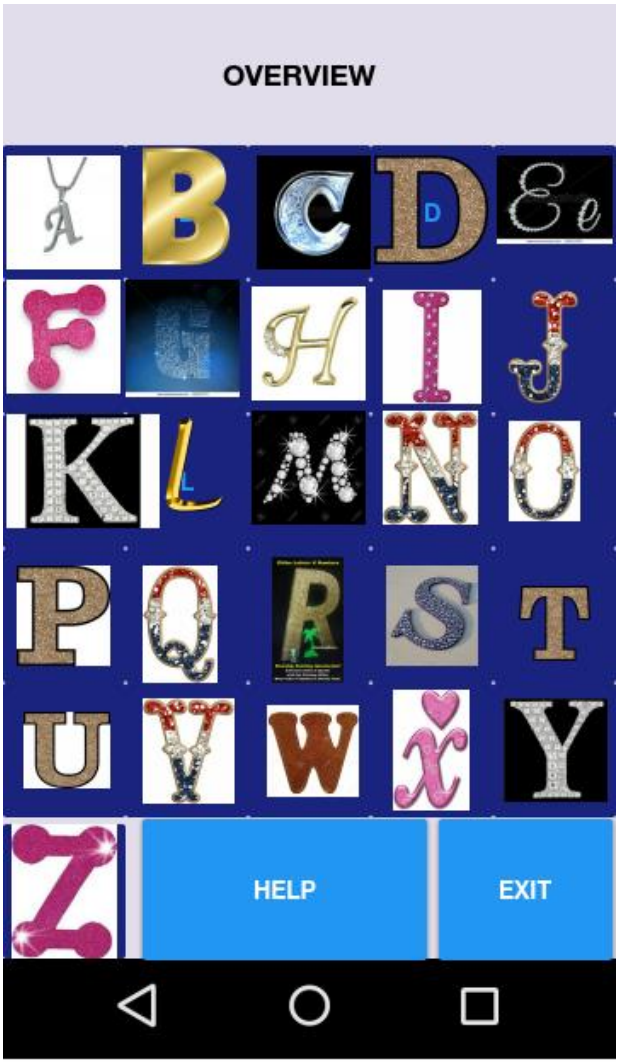

Figure 4.1: overview page, with custom image buttons, response on click.

- Figure 4.1 illustrates the overview page where buttons are customized according to the user preference. Each image in the diagram represents a button. If the user clicks once on a specific button the application switch to a new page where the image is viewed. The user may wish to change the image at any time, this page allows the user to change a certain image one at a time, unlike the previous page (image view preference), where the user is allowed to import multiple images by clicking the sync button. This is achieved by double clicking on the image, in which the application takes the user to the specific page where they are able to do this.

- If the user is not a common user of mobile apps, the help button is designed for such. It contains information on what happens for each action performed in the whole app, furthermore the user may wish to abort the app, and hence the exit button allows such. Customized buttons are flat buttons rather than raised buttons. This is to minimize distraction from the content around it. They exist in the same layer as the content they reference, thus no shadows around them exists.

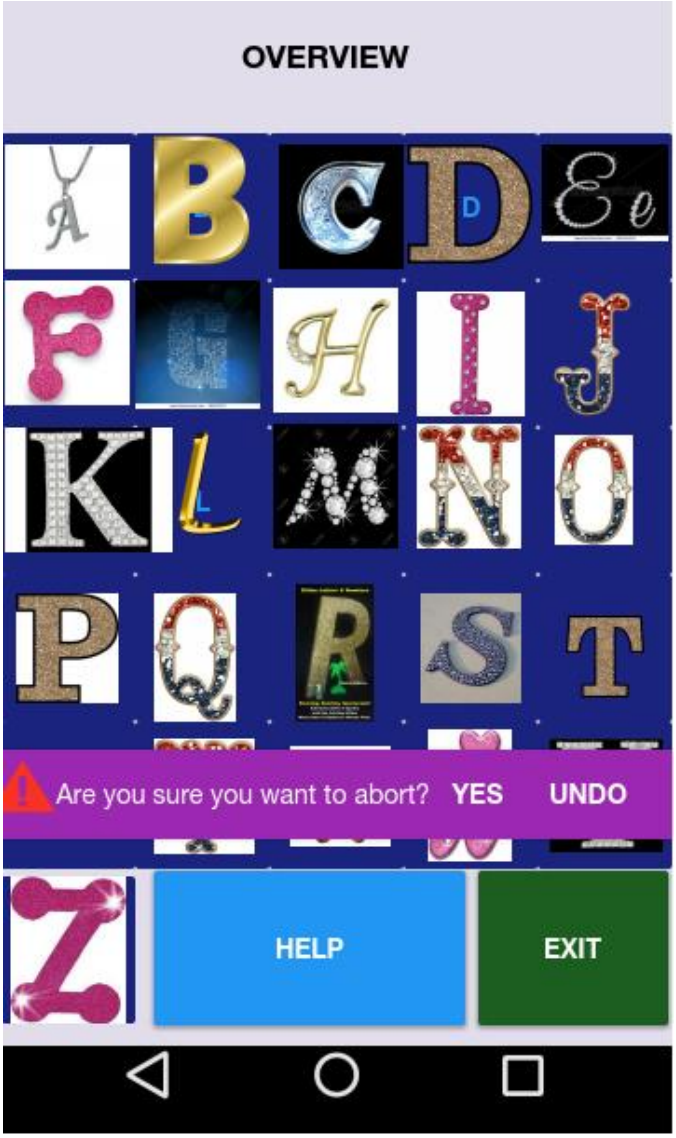

Figure 4.2: *overview page, with custom image buttons, exit button*

- On figure 4.2, the exit button was clicked. First the app ask the user if they are certain about the action performed, if the user clicks yes then the application terminates, otherwise it returns to the previous state, the overview page. The notification with the aid of the warning icon gives the user relevant and sufficient information about the

action performed. This also allow error prevention as it may be a possibility that the user may have mistakenly pressed the button.

Figure 5: letter page, image switcher with action detection on touch, response on click.

- Figure 5 illustrates the letter page where the user is able to experience the full view of the image selected. The three features at the bottom are buttons where FP button is representative of the first page, when clicked it takes the user to the first alphabet letter and displays the full view of the contained image, similarly LP button is representative of the last page, when clicked it takes the user to the last alphabet letter and displays the full view of the contained image. Overview button enables the user to return to the previous

page, the overview page. This is intended if they wish to select another image for instance, they want to select letter P and they do not want to rely on next and previous. For the user to go to the previous or next image, they can slide on the screen in the direction left or right depending on which image they want to view. Sliding on the left displays the next image, similarly right slide displays the next image. Basically the direction in which the user slides is synonymous to the image desired to be viewed. If the last letter image is on display, obtaining the next image is not possible, similarly if the first letter image is on display, obtaining the previous image is not possible. There exists action detection on touch such that it is flexible whether a pointer by pc for instance is used or touch by finger on mobile phone. There are constraints however on how the app responds when it detects touch, i.e. the app only responds when there is a right or left swipe motion, and when the user clicks on the image, there is no response.

(ii) **<u>State transition</u>**

- Figures 1-5, illustrate state transition, meaning how the current state of the page changes as action is being performed on that page.
- Unfortunately from (i) above it is not completely explicit on how the changes of states occur.
- The application is launched by the user, the welcoming screen appears on the screen and disappears once the progress bar if fully loaded. This is automatically done by the application and no user action is required.
- Once the welcoming screen has faded out, the image view preference page appears. If the user touches the button instead of clicking it, notification message appears giving the user a brief function of the button. If the user clicks on DEFAULT, application displays the overview page immediately with buttons named after the corresponding letters. And the default image library and settings are used for image display on the letter view page. If the user however clicks on SYNC, the app requires the user to import images from their desired source/library and allows customization. The user is allowed to match images according to the id's corresponding to alphabet letters, in that way the letters will appear corresponding to the alphabet they represent. If there is no image for the button then the default settings applies to that button. Once the process is complete then the overview page is displayed.
- The overview page consists of 26 buttons: - alphabet letters, help and exit. The user can do two things with regards to the alphabet letters. They can either click once or double click, in which case the app reacts in different ways. If the user double clicks, it allows them to change the current image on the button, and this takes them back to where they wish to import that image. If the user single clicks, then the letter view page appears, where the user is able to view the current image, first and last image view on button click, previous state is achieved by click on overview.

- The user may seek for help on how the app works in general if they are not non-standard users of mobile apps. A help page is displayed, giving the user sufficient information of the usability in general.
- The user may wish to exit the app without using the default exit button of the mobile device or it may be pressed by mistake. In that case pop- up notification asks the user if they are certain about the action performed. The state of the overview page is on pause and has not terminated in this regard, it only terminates once the user clicks on the yes button.

Design patterns, principles and laws

(i) Raised buttons and size

- In general raised buttons serves as an affordance to the user, meaning it gives the user the impression that the feature must be clicked as it looks like a button. On the image view preference page the buttons are placed in the order they are placed in for a specific reason. SYNC is at the bottom as it saves time for the user to click it. An application of Fitt's law is visible in this regard.
- On the overview page Help and exit are placed at the bottom, with help being big in size. This is because the target audience are children and the probability that they may be able to use the app with ease is low, and hence this button may be used more frequently which is why it must be easily accessible and be visible enough to the user while on the other hand the exit button may rarely be used, or can be clicked by mistake, which is why is placed on the last end in the app. Principle of least effort applies in this regard.
- On the letter view page all the buttons are placed at the bottom for easy accessibility, the size of an overview button follows an assumption that an overview button may be used more frequently than the rest of the button.

(ii) Flat buttons(image buttons)

- The use of flat buttons rather than raised buttons on the overview page was for simplicity as well as minimizing distractions from the content around each button. The buttons are placed in an ascending order in rows separated by small spaces in between them. This button design makes it uncommon for a user to make an error, say click two buttons at the same time, as the size is big enough.
- For non-standard users it may take a longer to make a decision as there are lot of options to choose from.

(iii) Pop- up notification messages, warning icon

- This feature gives the user feedback as an assurance to the user.
- The user is informed as to what the implications of each action has.
- The warning icon adds to the message displayed and gives user more options especially on the overview page when exit is clicked. An ultimatum is given.

The design follows most of the design principles and others may appreciate this design as it is user friendly, not ambiguous in terms of features, gives feedback, also it allows error prevention, help users recover, recognize errors on some instances. These features are well integrated. In hindsight the design yield the best utility to user.

In conclusion most of the features used were used for a specific reasons, following design patterns, also the design has constraints which ensures that the feature is used in the correct way for instance, to view the next image on the image view page, the use has to swipe in either one of the two directions, otherwise the app will not respond. In each design, the developer wishes to minimize the time taken to complete a task.