Capstone Stage 1 - Tam Love

GitHub Username: taml

Lettering Tutorial App

Description

The Lettering Tutorial App brings the world of hand lettering, brush lettering, calligraphy, type and typography to mobile users!

Whilst the Lettering Tutorial website provides a suitable experience for users, the experience could be improved for mobile and tablet devices. The Lettering Tutorial App provides the experience users need, by simplifying content that is more suited to desktop browsing and by removing unnecessary whitespace which makes scrolling on a small device tiresome. The Lettering Tutorial App also makes it easy to read articles on the go, with the ability to save favourite articles for offline viewing!

Intended User

The Lettering Tutorial App is intended for those interested in the art of hand lettering, brush lettering, calligraphy, type and typography. Those with a general interest in art may also find the App of interest.

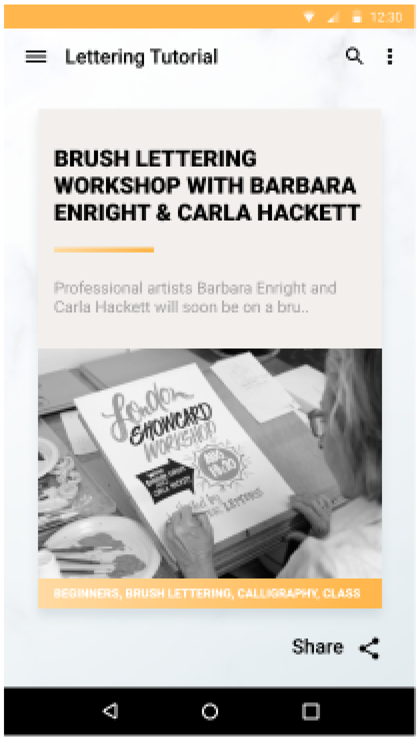
Features

The main features of the Lettering Tutorial App include:

- Fetching and displaying latest articles from the Lettering Tutorial website via an API.
- The ability to search all articles available within the Lettering Tutorial API.
- The option to share articles on social media.
- Saving favourite articles for offline reading.
- A widget which displays a list of favourite articles.

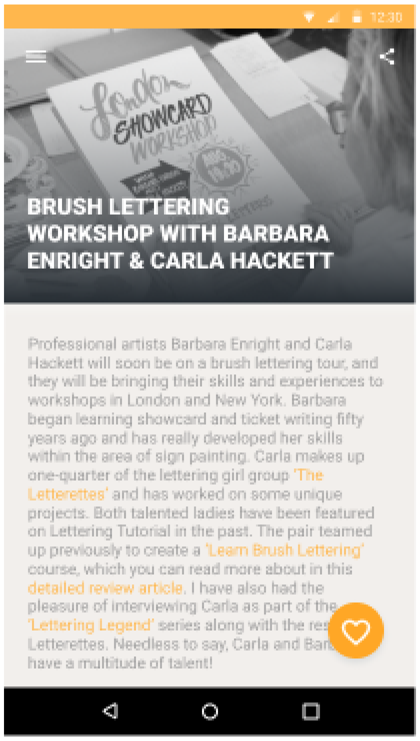
User Interface Mocks

Screen 1


12:30
Lettering Tutorial
BRUSH LETTERING WORKSHOP WITH BARBARA ENRIGHT & CARLA HACKETT
Professional artists Barbara Enright and Carla Hackett will soon be on a bru..
BEGINNERS, BRUSH LETTERING, CALLIGRAPHY, CLASS
Share

This high fidelity mockup design displays the starting Activity of the Lettering Tutorial App. It consists of a feed of articles which include an article title, hero image, a snippet of the article text, and tags that relate to the article which can be scrolled through horizontally. An option to share an article is included at this stage. From this screen a user can also search for articles, and adjust the settings of the app to return a different number of articles or change the sort order of articles.

Screen 2

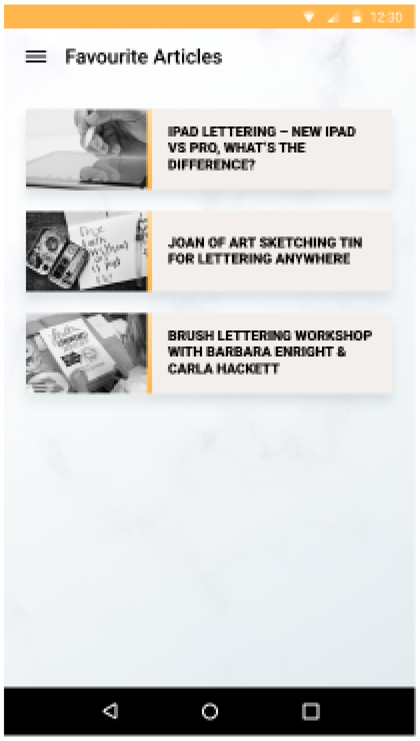


BRUSH LETTERING WORKSHOP WITH BARBARA ENRIGHT & CARLA HACKETT

Professional artists Barbara Enright and Carla Hackett will soon be on a brush lettering tour, and they will be bringing their skills and experiences to workshops in London and New York. Barbara began learning showcard and ticket writing fifty years ago and has really developed her skills within the area of sign painting. Carla makes up one-quarter of the lettering girl group 'The Letterettes' and has worked on some unique projects. Both talented ladies have been featured on Lettering Tutorial in the past. The pair teamed up previously to create a 'Learn Brush Lettering' course, which you can read more about in this detailed review article. I have also had the pleasure of interviewing Carla as part of the 'Lettering Legend' series along with the res Letterettes. Needless to say, Carla and Bar have a multitude of talent!

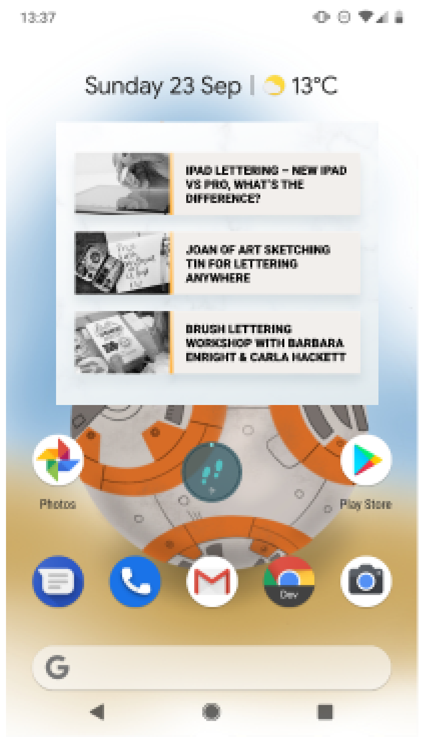
This screen displays an article Activity. A nice sized full bleed hero image is incorporated along with the article title and body text. Users have the option to share the article via the AppBar, or favourite the article for offline reading via the Floating Action Button (FAB) in the bottom right corner. If a user taps on an article from the starting Activity, or the favourites Activity this is the screen they will be taken to.

Screen 3


12:30
Favourite Articles
IPAD LETTERING – NEW IPAD VS PRO, WHAT'S THE DIFFERENCE?
JOAN OF ART SKETCHING TIN FOR LETTERING ANYWHERE
BRUSH LETTERING WORKSHOP WITH BARBARA ENRIGHT & CARLA HACKETT

When a user favourites an article they will be able to access that article and other favourites from this screen. Favourite articles are listed showing a thumbnail image and the article title. Users can remove favourites from this Activity by swiping across list items. They can also remove favourites by selecting the FAB of previously favorited articles in the articles Activity. When a user selects an article from this screen, it will take them to the article Activity screen.

Screen 4


13:37
Sunday 23 Sep | 13°C
IPAD LETTERING – NEW IPAD VS PRO, WHAT'S THE DIFFERENCE?
JOAN OF ART SKETCHING TIN FOR LETTERING ANYWHERE
BRUSH LETTERING WORKSHOP WITH BARBARA ENRIGHT & CARLA HACKETT
Photos
Play Store

Upon adding a Lettering Tutorial widget to their device, a user can view a list of the articles that they have selected as favourites. Tapping an article will launch the article Activity.

Key Considerations

How will your app handle data persistence?

Data will be fetched by an AsyncTaskLoader, and then will be persisted through the use of Room and Android Architecture Components including ViewModels and LiveData when a user adds an article to their list of favourites. An AsyncTaskLoader will be used over an AsyncTask so that if an Activity is recreated resources are not wasted on an AsyncTask which can no longer access an Activity. This will also save unnecessarily repeating networking requests to retrieve data which may have been lost when for example a user changes the orientation of their device.

Describe any edge or corner cases in the UX.

No or poor internet connectivity is an edge case which may occur within the user experience. Under these circumstances a user will be notified that there is a connectivity issue via a Toast or Snackbar message. The user will also be prompted to explore the favourites section of the App where if they have saved any, articles can be viewed without the need for internet connectivity.

Another edge case is that the Lettering Tutorial API may return with a response code that is not equal to 200, meaning that there has been an error fetching data from the API. Under these circumstances the UI will display a message notifying the user that there has been an issue fetching articles, and to try fetching articles again. A method for users to refresh data will be included within the UI. Similarly as above, the user will be prompted to explore the favourites section of the App where if they have saved any, articles can be viewed without the need to fetch data from the API.

Describe any libraries you’ll be using and share your reasoning for including them.

Picasso Library - To load and cache images retrieved from the Lettering Tutorial API.
Design Support Library - To incorporate elements such as Coordinator Layouts and Floating Action Buttons (FAB).
Retrofit2 Library - To make background networking requests to the Lettering Tutorial API so that article data can be fetched.
Espresso Library - To test that UI elements operate and display data as expected.
JSoup Library - To parse complex HTML in order to display article detail.
Room Persistence Library - To create a database in order to store, query and delete a users favourite articles.
Android Architecture Components ViewModel and LiveData Library - To enable Room database data to interact with the lifecycle of Activities and reflect changes to data without making unnecessary calls to the database.

Describe how you will implement Google Play Services or other external services.

AdMob and Firebase Performance Monitoring are Google Play / Firebase services which will be used within the Lettering Tutorial App. Banner ads will be included within the layout of some Activities, and these will be placed in a way that does not detract from the overall article reading experience. It is also important that the banner ads do not obscure important views too. Firebase Performance Monitoring will be used to detect performance of the Lettering Tutorial App. This is useful because the App relies on making networking requests and loading data. Speed and loading time impact user experience, and performance metrics can be utilised to make improvements to the App over time.

Next Steps: Required Tasks

Task 1: Project Setup

- Initialise a new git repository.
- Setup a new project supporting a minimum SDK version of 16 with an ArticleFeedActivity.class and corresponding XML layout file.
- Create packages to contain UI, data, models and networking classes.
- Add an ArticleActivity.class and corresponding XML layout file.
- Add a FavouriteArticleActivity.class and corresponding XML layout file.
- Implement the required libraries (specified above) within the app level build.gradle file.
- Add internet and network state permissions to the AndroidManifest.xml file.

Task 2: Implement UI for Each Activity and Fragment

- Build the UI for ArticleFeedActivity.
- Build the UI for ArticleFeedActivity RecyclerView list item.
- Build the UI for API search functionality.
- Build the UI for ArticleActivity.
- Build the UI for FavouriteArticleActivity.
- Build the UI for FavouriteArticleActivity RecyclerView list item.
- Configure toolbars, and create XML Menu.

Task 3: Implement Adapters and POJOs

- Create an Article POJO.
- Create a custom adapter for displaying Articles within a RecyclerView.

Task 4: Implement Networking Functionality

- Add Retrofit service Interface.
- Add Retrofit client.
- Add serializing to Article POJO.
- Call Retrofit from ArticleFeedActivity.
- Update the UI with Retrofit call results.
- Create SettingsActivity.
- Add SearchResultsActivity and create corresponding XML layout.

- Configure Retrofit to accept parameters obtained via search functionality or SettingsActivity, and update the UI accordingly.

Task 5: Implement Android Architecture Components and Room Data Persistence

- Create an Entity.
- Create a DAO interface with Query, Insert and Delete methods.
- Create a class which extends RoomDatabase.
- Setup ViewModel functionality so that data persistence is lifecycle aware.
- Observe data for changes using LiveData so that updates to the database are reflected.
- Double check that if persisting data has taken place on the main thread for testing purposes, this functionality is disabled in the getInstance() method of the class which extends RoomDatabase.

Task 6: Implement Google Play / Firebase Services

- Add Google Play Service AdMob ads to UI and related code to Activities to display test banner ads.
- Add Firebase Performance Monitoring to networking and data persistence functionality, enabling collection of performance metrics.
- Test performance metrics are being collected via the Firebase dashboard.

Task 7: Implement App Accessibility

- Add content descriptions.
- Add support for navigation using a D-pad.
- Enable RTL switching on all relevant layouts.
- Store all Strings in a Strings.xml resource file.

Task 8: Implement Widget

- Create an XML appwidget-provider.
- Create an XML layout for the Widget content and for the Widget list item layout.

- Create a custom WidgetProvider which extends AppWidgetProvider and implement the onUpdate() method.
- Create a RemoteViewsFactory class to act as an Adapter wrapper between the widget and ListView within the widget.
- Create a Service to call the RemoteViewsFactory class.

Task 9: Handle Error Cases

- Add validation to networking and data functionality.
- Provide fallback values and images.
- Add checks for internet connectivity in relevant places of the App.
- Add ability to refresh data requested by networking.
- Provide appropriate validation and connectivity feedback to users through the UI. Including hiding views and displaying TextViews with feedback or displaying Toasts / Snackbars.

Task 10: Implement UI Tests

- Add Espresso tests to check core functionality of the App.
- Run Espresso tests on mobile and tablet devices to check functionality is consistent.

Task 11: Project Building Configuration

- Use the Gradle clean task to clean the App.
- Check the App builds and deploys using the Gradle installRelease task.
- Equip the App with a signing configuration.
- Include the keystore and associated password in the App repository.
- Reference the keystore with a relative path.