**How to Use this Template**

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: “**Capstone\_Stage1**”
3. Replace the text in green

[Description](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.sm4ra97uwo11)

[Intended User](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.aws88pzfmqca)

[Features](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.zheq5430xrpq)

[User Interface Mocks](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.giquerrw6g46)

[Screen 1](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.a4jdupabry3k)

[Screen 2](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.dpcbbkx5yry)

[Key Considerations](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.gvcvmae8jn8u)

[How will your app handle data persistence?](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.v8my7nhtvz0m)

[Describe any corner cases in the UX.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.gw69vjn1ico0)

[Describe any libraries you’ll be using and share your reasoning for including them.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.6yqqubmw5bs)

[Describe how you will implement Google Play Services.](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.qrxg682nywe6)

[Next Steps: Required Tasks](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.v518bncmggeg)

[Task 1: Project Setup](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.8oe8zpk3qsmp)

[Task 2: Implement UI for Each Activity and Fragment](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.rzllsk6uqztx)

[Task 3: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.fdmohs7hes)

[Task 4: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.umfwsvmx7tpn)

[Task 5: Your Next Task](https://docs.google.com/document/d/1gKP6RxykeekNk5bYxXIKjEitKDPdxpRyIaa9t50bLSA/edit#heading=h.kjidlkq4xm3u)

**GitHub Username**: khkavinha

WANTED TICKETS

Description

“WANTED TICKETS” will allow users to buy/sell concert tickets from a third-party applicant. The biggest importance about this application is users will not feel sketch/worried about buying a ticket from the seller. Instead the user has the ability to view some information about the seller’s. The information will provide a profile picture, first name, city location and reviews about the sellers.

With the Application, you are able to

1. Post tickets you are trying to sell.
2. Buy tickets from a seller.
3. Write reviews about the seller/buyer after the transaction is complete.
4. A widget FAB chat that will allow users to chat with each other. ­
5. Report the user if there were any fraud.

I will be using java as programming language into developing this application.

Intended User

This application is intended for mainly concert goers that are willing to buy/sell tickets.

Features

List the main features of your app. For example:

* Create a thread to post tickets
* Write reviews
* Chat feature
* Login Page
* Account Creation
* Delete thread
* Widget android background

User Interface Mocks

****

Main Activity page, the X box is the image of the ticket

****

Main Activity with extensions of tabs, the X box is the image of the ticket



Main Activity with “Welcome!” Toast when the user logins, the X box is the image of the ticket



Main Activity with “Please login to create a post!!” Toast when tries to create a post when they are not login, the X box is the image of the ticket



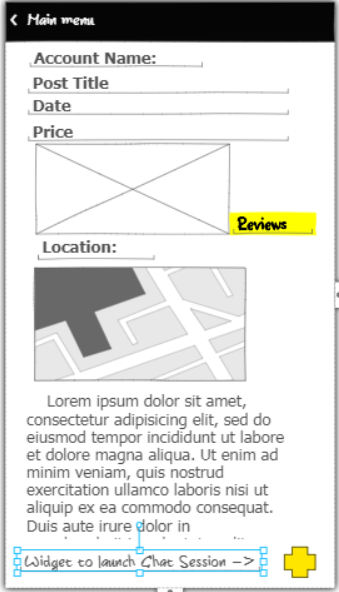
Main Activity with “Account has been deleted!!” Toast when the user deletes its account, the X box is the image of the ticket



Main Activity with “Review has been posted!!” Toast when the user finishes writing a review, the X box is the image of the ticket



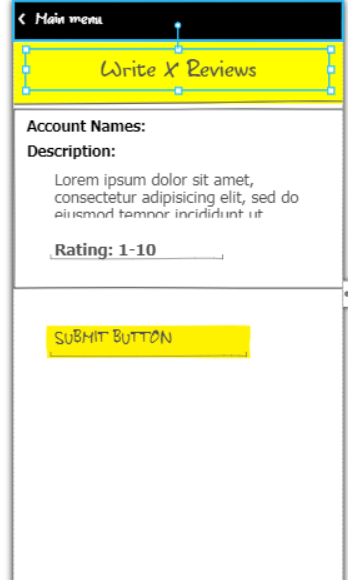
Create Posts Screen



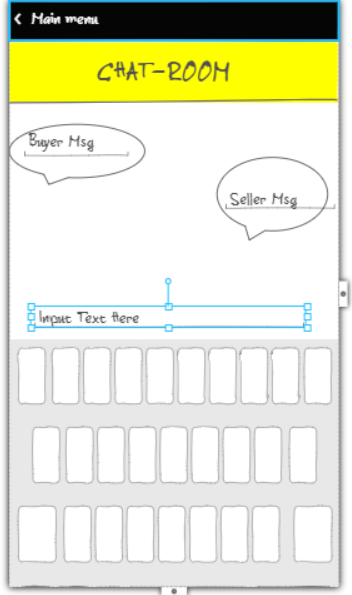
Detail activity screen when a buyer enters the screen, the X box is the image of the ticket, Reviews will show the rating. FAB widget is in the bottom right corner to launch chat session.



Detail activity screen when a seller enters the screen, the X box is the image of the ticket, Reviews will show the rating. FAB widget is in the bottom right corner to launch chat session.



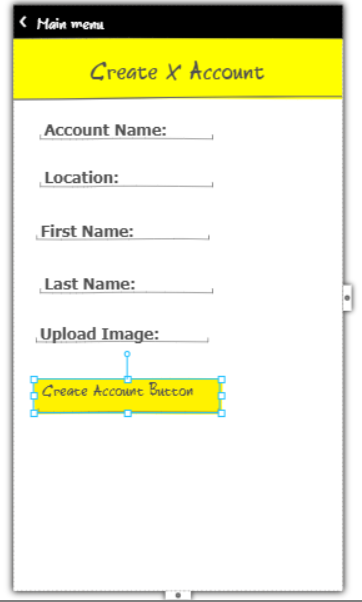
Reviews screen when the user writes a review about the seller



Chat room when the buyer and seller is communicating with one and another

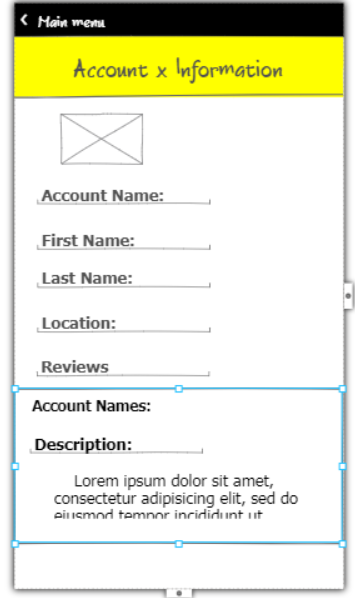


Menu tabs when the user clicks on the menu button in Main Activity

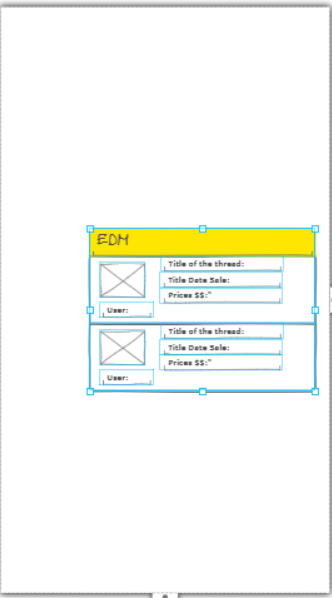
  
Create account screen



Login Page screen



Account information when the user clicks on the account name inside Detail Activity screen.



App’s widget will display on the android background base on the specific concert the user has chosen.

Key Considerations

**How will your app handle data persistence?**

Describe how your app with handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)

* Content Provider
* Firebase
* SyncAdapter to handle fetching data daily from google API.

**Describe any edge or corner cases in the UX.**

* User will be able to see the specific music genre when they switch on different tabs.
* User is able to scroll the tabs to choose a specific concert.
* User is able to delete the post in the detail activity by pressing the delete button and will go back to the main activity
* User is able to view the detail of the thread when it clicks on one of the recyclerviews in the Main Activity
* 4 Menu options will navigate the users to different screens base on the option they chose.
* Back button will allow the user to navigate back to the Main Activity.
* User does not have an ability to create a post when the user is not login.
* User has an ability to create a post when the user is Login.
* Toast message will display when the review is submitted.
* Toast message will display when the user creates a post.
* Toast message will display when the user creates an account
* Toast message will display when the user deletes an account
* Chat screen will display when the user clicks on the FAB widget button on the bottom right corner of the screen.
* Once a specific concert type tab is chosen, app’s widget will display a list of items that is on sale.

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

* Picasso version: 2.5.2
* Retrofit version: 2.3.0
* ButterKnife version: 8.8.1
* Rxjava version 2.1.5
* Fancy Toast version 0.1.3
* Firebase version 15.0.0

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

* Google Maps, it will be used to display the location of the user. Version 15.0.0
* Google Cloud Messaging, it will be used to perform messaging with one and another. Version 15.0.0
* Google Account Login & Sign In, it will be used to login an account. Version 15.0.0

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

**Task 1: Project Setup**

* Configure libraries
* Add google play services

**Task 2: Implement UI for Each Activity and Fragment**

Build the following UIs

1. Main Activity
2. Menu
3. Login
4. Account Information
5. Create Post
6. Write a Review
7. Detail Screen
8. Chat Session
9. Widget in the android background

**Task 3: Main Activity**

* Add a tool bar on top of the application
* Add a menu option on the left side of the tool bar
* Add a create thread option on the right side of the toolbar
* Add a layout to display the app Name on the bottom of the menu
* Add 5 tabs and allow user to switch within the tabs on the bottom of the app name layer
* Implement a content provider and sqlitedatabase
* Implement a cursor adapter
* Inside ListViews, add “ImageView”, “Account Name”, Title of Thread”, “Price”, “Sale Date”

**Task 4: Create Post**

* Create `Create Post` class
* Add a toolbar that has a back button on the left with `Main Menu` title next to the button.
* Add a layer to display ‘CREATExTOP’ on the bottom of the tool bar.
* In the main menu, create an intent. Test and verify it going to the correct layer.
* Add TextViews for title, price, date, upload image, description, and check boxes for EDM, Hip-Hop, Country, Musical, Musical Festivals
* Add an input data next to those following textviews.
* Add a submit button
* Test and verify the data is being display inside the Main Activity
* Implement an error case when the user didn’t add all the information to create a Post.

**Task 5: Detail Activity**

* Create ‘Detail Activity’ class.
* Create a toolbar on top
* Create a back button on the left side with Main Manu in the side
* Inside ‘Detail Activity’, Add the following RecyclerViews for “Account Name:”, “Post Title”, “Date”, “Price”, “Image”, “Location”, “Description”.
* Implement Retrofit and RxJava to fetch data from Google API
* Implement SyncAdapter to fetch data daily through Google API.
* Verify Map is display correctly inside Detail Activity

**Task 7: Chat Widget Button**

* Implement FAB button that will intent to a chatroom from Detail Activity class
* Implement firebase
* Add RivChat library to implement the chat option

**Task 8: Menu Selection in the Main Activity**

* Create a menu option inside Main Activity
* Add the 4 following tabs in the menu option “Create Account”, “Account Information”, “Log In”, “Log Out”.

**Task 9: Menu: Create Account**

* Create a toolbar on top
* Create a back button on the left side with Main Menu in the side
* Add a layer to display ‘CREATExAccount’ on the bottom of the tool bar.
* Add TextViews for “Account Name”, “Location”, First Name”, Last Name”.
* Add a button to “Create Account” on the bottom of Last name.
* Add the following library Fancy Toast to display “Welcome!” toast when the user logs in

**Task 10: Menu: Create Account Information**

* Create a toolbar on top
* Create a back button on the left side with Main Menu in the side
* Add a layer to display ‘AccountxInformation’ on the bottom of the tool bar.
* Implement a RecyclerView adapter.
* Add RecyclerViews for “Account Name”, “Location”, First Name”, Last Name”.
* Verify data is being display correct inside Account Information.
* Add a button to “Delete” on the bottom of Last name.

**Task 11: Reviews inside Detail Activity**

* Implement a review textview inside Detail Activity.
* Implement a reviews class to allow users to write a review about the seller.
* Implement reviews to be display inside the account information.

**Task 12: Menu: Log In**

* Create a toolbar on top
* Create a back button on the left side with Main Menu in the side
* Add a layer to display ‘AccountxTicket login page’ on the bottom of the tool bar.
* Add TextViews for “Login” and “Password”.
* Add a button “Login Button” on the bottom of Password.
* Add the following library Fancy Toast to display “Welcome!” toast when the user logs in.

**Task 13: Menu: Log Out**

* The user is able to log out the application when the user clicks ‘Log Out’

**Task 14: App’s Widget**

* Implement WidgetProvider and RemoteViewService classes.
* Test and verify the widget is display in the android background.
* Implement a way for the widget lists to refresh and change to the corresponding tab the user has chosen.

Add as many tasks as you need to complete your app.

**Submission Instructions**

* After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
  + Make sure the PDF is named “**Capstone\_Stage1.pdf**”
* Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

* Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
* Add this document to your repo. Make sure it’s named “**Capstone\_Stage1.pdf**”