

Technology Review

UniversityGear

November 13, 2016

Devin Foulger
Hector Trujillo
Bryan Liauw



Contents

1	Introduction	1
2	Technologies	1
2.1	Technology: Checkout	1
2.1.1	Options	1
2.1.2	Use in design	1
2.1.3	Cost, availability, speed, security	1
2.1.4	Evaluation	2
2.1.5	Best Choice	2
2.2	Technology: Checkout UI	2
2.2.1	Options	3
2.2.2	Use in design	3
2.2.3	Cost	3
2.2.4	Evaluation	3
2.2.5	Best Choice	4
2.3	Technology: Getting data for Single Item View	4
2.3.1	Options	4
2.3.2	Use in design	4
2.3.3	Cost and speed	4
2.3.4	Comparisons	5
2.3.5	Evaluation	5
2.3.6	Best Choice	5
3	Conclusion	5

1 Introduction

2 Technologies

2.1 Technology: Checkout

Checking out is a feature that is essential to any application or website where you purchase some item. This means that our application will need to allow the user to check out the items in some fashion.

2.1.1 Options

Option 1: Checkout with sign-in

Allowing the user to sign in to checkout items gives us the benefit of allowing members to purchase from our application. In order to implement this feature, we would have to use eBay's Order API. This API would allow for the user to purchase items using their existing eBay and PayPal accounts. This also allows the user to view the shipping address provided and any fulfillment information.

Option 2: Checkout as guest user

Allowing the user to checkout items through a guest check out would allow any user to purchase from our application. In order to implement guest checkout, we would need to use eBay's Order API with GUEST CHECKOUT SESSION. This is different than using the Order API normally. With guest checkout, the user is able to see that item that they have purchased and the status of the payment and order.

Option 3: Checkout is either guest user or with sign-in

Allowing the user with the option to either sign-in or use guest checkout gives us the option to offer which ever method makes the user most comfortable when checking out. This would be that we would have to use eBay's Order API to the fullest by implementing both sign-in and guest checkout. This means that the user would be able to use PayPal, if they have an account, or checkout via credit card.

2.1.2 Use in design

This is a fundamental use in our design. Without this feature, our application would simply be a searching application. The user would only be able to search for specific items that exist within eBay's stores, but not actually purchase them.

2.1.3 Cost, availability, speed, security

Option 1: Checkout with sign-in

The cost of checkout with sign-in is the development time it would take to implement. This feature is tricky and would complicate the use of eBay's APIs. However, the availability is limited to those that have an eBay account. The speed is also largely dependent on internet connectivity and the status of eBay's servers. Finally, the security is very tight. This is because the user must authenticate before making a purchase. On top of that, all information that is used is sent via HTTPS.

Option 2: Checkout as guest user

The cost of guest checkout is relatively low. It would take the least amount of development time and resources. The availability is also very high allowing us to reach anyone who would like to use our app. Again, the speed is dependent on internet connectivity and the status of eBay's servers. However, the security would be great. None of the user's information should be saved to the device. That includes things like credit card information and shipping address. Also, any information that is used is sent via HTTPS requests.

Option 3: Checkout is either guest user or with sign-in

The cost of the guest checkout coupled with sign-in is very high. It would require a significant amount of development time to complete. However, the availability would be at its greatest because it would allow for any and all users to purchase items. The speed of the checkout would, again, depend on the status of eBay's servers and internet connectivity. The security is the same as the above options.

TABLE 1
Comparison Table of Options for Checkout

	Time Cost	Availability	Speed	Security
Option 1	High	To only members of eBay	Fast	Best, because it offers an extra layer of authentication
Option 2	Low	To everyone as a guest	Best	Great
Option 3	Very high	To members of eBay and guests	This could either be best or fast	This could be best or great

2.1.4 Evaluation

Each option has their advantages and disadvantages. If we were to allow users to checkout only using sign-in, then we might only be targeting a smaller audience. The complexity for implementing such a feature is much harder than that of guest checkout as well. Using guest checkout would allow for all users to use our application with no complicated sign in. It would also mean that users would not have to create an account to user our application. If we were to implement both sign-in and guest checkout, the complexity would be too great. Given the small amount of development time it might be hard to implement both features. However, this option would give us the ability to reach every user possible.

2.1.5 Best Choice

The best choice for our application is implementing a guest checkout. This is because with a short development time, guest checkout would be the fastest to implement. This would also allow any user to purchase from our application, which means we would not be alienating someone who does not have an eBay account.

2.2 Technology: Checkout UI

Checking out needs to have a UI that is easy to use. If we only stub out a UI, it might be complicated, or worse, ugly. The UI needs to be attractive and functional.

2.2.1 Options

Option 1: Layout Editor (Native Android Studio)

The native android layout editor doesn't work very well for prototyping. However, it allows for easy UI creation while being able to map the buttons to correct functions. This tool doesn't offer much in terms of quick design, but it offers a lot of functionality. It uses XML files to create the UI. It does allow you to alter any aspect of your UI, which is very good as well. It also allows you to create your own UI objects very easily.

Option 2: Indigo Studio

Indigo Studio is a wire framing tool. With this tool I would be able to create great looking wireframes. This doesn't allow for much in terms of actual development though. It is just a tool for quick prototyping and design. However, it does offer a lot in terms of customization and they also offer a lot of images for buttons, list views, and things of that sort.

Option 3: Just In Mind

Just In Mind is a comprehensive prototyping tool for many types of development, including Android development. This tool offers a lot in terms of development. There are many UI libraries for Android. This tool would also allow me to use templates that they have already built. On top of that, any of the designs or UIs I make can be transferred into already existing projects. This tool also offers integration with other tools as well, like Photoshop. It also features interactive images and animations for use in your projects.

2.2.2 Use in design

Having a functional and beautiful looking UI keeps users attracted to your app. So, having a nice UI is a must in this project. The user will also constantly be interacting with the application.

2.2.3 Cost

These tools are simply for creating clean and responsive UIs. That being said, the native Android Studio Layout Editor is free. Indigo Studio costs 25 dollars a month and Just In Mind costs 19 dollars a month.

TABLE 2
Comparison Table of Options for Checkout UI

	Cost	Learning Curve
Option 1	Free	There is somewhat of a learning curve as it requires you to also develop at the same time
Option 2	Free with limited features, then 25 dollars a month	Low
Option 3	Free with limited features, then 19 dollars a month	Low

2.2.4 Evaluation

Each tool has clear disadvantages and advantages when compared to each other. Just In Mind and Indigo Studio have relatively similar functions. However, Just In Mind offers a bit more such as UI libraries that are compatible with Android Studio. Indigo Studio does allow for faster creation of wireframes when compared to Just In Mind. These two tools can be much better than the native layout editor in Android Studio. The editor layout also offers its own features as well. It gives developers the ability to customizing anything they like about their UI.

2.2.5 Best Choice

The best choice for this project would be to use the native Android Studio Layout Editor. Although the other tools offer a lot more in terms of prototyping, they have a cost. If you don't pay for the tools, you don't get everything that is included. However, Android Studio does offer the ultimate tool for any type of customization you would like to make. You have full control over what you are trying to do. You also don't have to worry about integration with other tools, because it is built in.

2.3 Technology: Getting data for Single Item View

The data that is received from eBay's APIs comes in the form of a JSON file. This JSON data needs to be parsed correctly so that the information is portrayed to the user in a meaningful manner. Each item needs to correctly be displayed to the user so that they can view the description or images of a particular item.

2.3.1 Options

Option 1:

GSON allows for Java objects to be converted into JSON objects, and vice versa. This is a tool that is not native to Android Studio, but offers a lot in terms of use. It has many methods that allow for array creation which holds JSON information. These arrays can then be used to fill the UI with relevant information for the user to view.

Option 2:

JsonReader is a light weight API that would allow me to easily read JSON files. It allows me to create JSON objects and arrays. The arrays would hold information that could easily be presented to the user.

Option 3:

JSON Parser is native to Android Studio and is very comprehensive. It would allow me to create objects using JSON files. I would have the ability to create JSON arrays, object, and key-value pairs using JSON Parser. It includes many methods for easy creation of JSON objects and parsing.

2.3.2 Use in design

A JSON parser is absolutely a must if we want to make sense of our data. This is because eBay's APIs return the data in JSON format. In order to display the information correctly to a user, we have to use a JSON parser.

2.3.3 Cost and speed

All three APIs do the same thing in the same fashion. They all create readable JSON arrays that hold information to be used by the developer.

2.3.4 Comparisons

All three APIs do the same thing in the same fashion. They all create readable JSON arrays that hold information to be used by the developer.

2.3.5 Evaluation

It is important to note that these APIs perform almost identically. On top of that, they all produce the same output. There is no clear advantage or disadvantages over the others. The only disadvantage to the JsonReader, is that you don't have the ability to create new JSON objects.

2.3.6 Best Choice

The best choice to use for our application would be the built in Android parser, JSON Parser. This is because all the parsers perform similarly, but with the built in parser, I get to avoid the headache of importing a new one. It also has the ability to create new JSON objects if that is needed.

3 Conclusion