

CS CAPSTONE TECHNOLOGY REVIEW

NOVEMBER 21, 2017

EBAY IOS ESPORTS APPLICATION

CS461 SENIOR SOFTWARE ENGINEERING PROJECT I

FALL 2017

PREPARED FOR

EBAY

LUTHER BOORN

Signature

Date

PREPARED BY

GROUP 17

SKILL CAPPED IRL

KATHERINE BAJNO

Signature

Date

Abstract

This document compares and contrasts various options for technology pieces in relation to the eBay iOS eSports application. User authentication is another technology piece that is analyzed. The potential choices for user authentication include Firebase, Facebook, and Google, and the three are compared and contrasted. The eCommerce options for the application are discussed including redirecting to the eBay application for purchasing, using the eBay Order API, and utilizing the Apple Pay framework. The final technology piece discussed is displaying multiple merchandise results. The options for this include a scrolling item list view, a scrolling item grid view, and a swiping single item view. The pros and cons of these are discussed.

CONTENTS

1	Introduction	2
2	User Authentication	2
2.1	Overview	2
2.2	Criteria	2
2.3	Potential Choices	2
2.3.1	Firebase	2
2.3.2	Facebook	2
2.3.3	Google	3
2.4	Discussion	3
2.5	Conclusion	3
3	eCommerce Options	3
3.1	Overview	3
3.2	Criteria	3
3.3	Potential Choices	3
3.3.1	Redirect to eBay.com	3
3.3.2	Checkout in application with eBay Order API	4
3.3.3	Checkout in application with Apple Pay	4
3.4	Discussion	4
3.5	Conclusion	4
4	Displaying Multiple Merchandise Results	4
4.1	Overview	4
4.2	Criteria	5
4.3	Potential Choices	5
4.3.1	Scrolling Item List View	5
4.3.2	Scrolling Item Grid View	5
4.3.3	Swiping Single Item View	5
4.4	Discussion	5
4.5	Conclusion	5
	References	5

1 INTRODUCTION

In order to create a successful mobile application to target users in the eSports market, it is essential that the correct technology choices are made. The user needs to be able to log-in with a low-cost and accessible account, browse merchandise relevant to them with ease, and successfully purchase the merchandise with a system that is reliable and secure. Discussed in this document are some technology options for each of these three categories, and the pros and cons of them.

2 USER AUTHENTICATION

2.1 Overview

When the application first launches, the user will be prompted to sign-in. In order to have a secure system for our user, an authentication system must be used. There are various services that could be used for this task including Firebase, Facebook, and Google that would work for our application.

2.2 Criteria

The best option for an authentication system would be a service that is accessible to the masses. It would be a low-cost option for the user. It would allow the user to sign-in through both email and password and a second mainstream account.

2.3 Potential Choices

2.3.1 *Firebase*

The Firebase Authentication system is that the service handles the UI flow for users signing in. With Firebase, we can choose to allow users to sign in through Facebook, Google, phone numbers, and by email address and password [1]. We do not have to handle the connection between our application and each service we decided to choose, and it would be easy to add on support for more sign-in options in the future. Firebase takes care of the sign-in process, and we would just have to use their API in the code. Firebase would also be free for our purposes, and would not require any review by Google to use.

2.3.2 *Facebook*

The Facebook API log-in authentication system only allows users to log-in who have a pre-existing Facebook account. If information is requested beyond people's public profiles, emails, or friends list, the application needs to be submitted for log-in review [2]. Facebook accounts are free and easy to make, and millions of users already have one. A user being logged into Facebook on an application allows them to create an account in the application without setting up a password, personalize information such as birthday and likes, and connect with friends through the social aspect. The log-in system easily integrates with Facebooks social API. Other features include being able to share their real identity in their public profile, and gradual authorization which allows us to only ask for them to log-in at the start but later on ask for more information from them in the future.

2.3.3 Google

The Google API has a secure log-in authentication system. If a user has a Google account, which is highly accessible and has zero cost associated with it for the user, they can log into the application. It allows the user to log-in with an account they use for many services already such as Gmail, Play, Google+, and many other Google services. With a Google sign-in account, the user can easily pay in the application with Android Pay, add an event to their Google Calendar, and integrate with other Google Drive services that the user is most likely already utilizing [3]. The Google log-in also uses the Google Smart Lock service, allowing for one-tap sign-in and saved passwords on Android.

2.4 Discussion

Compared to Facebook and Google, the Firebase log-in authentication system allows for multiple ways for users to log-in, rather than just through one service such as Google and Facebook. The benefit of all three options are that they are all low-cost, as making an account is free for users and they are highly accessible. A benefit of Firebase in contrast to Facebook and Google is the privacy for our users. Since it does not require having an account in an outside application, they can keep their other information safe. Compared to Facebook and Firebase, Google allows for purchasing integration through the user signing in with their system.

2.5 Conclusion

We chose choice one, Firebase, because it allows for a vast amount of options through the user signing in, while maintaining their security. The service is free for both us and the user, making it a great choice. It also allows for sign in through multiple services, unlike the other two options. While we are only allowing the user to log-in with email and password at first, this will be great for future-proofing the application.

3 ECOMMERCE OPTIONS

3.1 Overview

In-order to purchase the eSports related merchandise the user is searching for, an eCommerce platform needs to be implemented in the application. The user needs to be able to purchase their selected item. Redirecting to eBay.com, the eBay Order API, and Apple Pay are the options to consider.

3.2 Criteria

The eCommerce purchasing option needs to have a secure checkout. It needs to be easy for the user. The option should also be feasible for the developer to do, and not have to get outside reviews or certifications for it. The option needs to ensure that the entire checkout process will occur from selecting to purchase the item, putting in their card information if not already saved, inputting their shipping and billing information if not already saved, and finalizing the purchase with conformation.

3.3 Potential Choices

3.3.1 Redirect to eBay.com

A way to handle eCommerce is out of the application. When a user clicks on an item to purchase it, they will be redirected to the eBay website where they can continue the checkout process. The query results that display the item in

the application would be used to generate the web address for purchasing on the eBay website. This would ensure the checkout process is secure. The user would only be able to purchase one item at a time, and would have to go through a separate process once on the eBay website.

3.3.2 Checkout in application with eBay Order API

Using the eBay Order API in our application would allow for the user to complete the full checkout process in our application. It supports users checking out as guests or as a eBay member. The user would be required to log-in to eBay if they wanted to complete a member checkout. A member would use PayPal to make their purchase, and a guest would use a credit card [4]. The entire checkout process would have to be implemented and handled in the application including payment authorization, and notifying the user of tracking information.

3.3.3 Checkout in application with Apple Pay

Apple Pay is another option for in-application purchasing. It is for iOS applications only. Users can authorize payments using Touch ID or using their card credentials that are already stored on their Apple Wallet /citeapplePay. Since a user needs to have an Apple account to download this application, they will not need to go through a second log-in step to purchase the item. A merchant ID needs to be registered with Apple to use this feature in applications, and the application needs to be registered and validated. The items would have to be sold in an authorized merchandise website by the developer.

3.4 Discussion

In contrast to the eBay Order API and Apple Pay, redirecting the purchase to eBay.com is guaranteed to work and be secure. A pro is we would not have to worry about contacting the user after the purchase with shipping information, as eBay will do that. Compared to the eBay Order API and redirecting to eBay.com, with Apple Pay the user is guaranteed to have an Apple account with Apple Pay being easy to set up. A con of Apple Pay is the user needs to be logged into Apple, and also the application needs to be registered with Apple in-order to utilize this feature. In contrast to Apple Pay, the eBay Order API would make in-application purchases easier. Since the user is purchasing merchandise from eBay, it is likely they have an eBay account. If not, they have the option as continuing as a guest.

3.5 Conclusion

We chose choice one, redirecting to eBay.com. eBay already has a great system in place for purchasing and we will utilize that. The checkout will be secure, and the user will be able to complete the entire checkout transaction from start to finish. eBay will then take care of notifying the user in the future about shipping and tracking information, and makes for the best user experience.

4 DISPLAYING MULTIPLE MERCHANDISE RESULTS

4.1 Overview

In-order for the user to browse the merchandise they want to purchase, the items need to be displayed. How the items are displayed can change the likelihood the user will purchase an item. Our options are a scrolling item list view, a scrolling item grid view, and a swiping single item view.

4.2 Criteria

The page should display as many merchandise results as possible while still being viewable to maximize the amount shown to the user. The view should make it easy to display new items on the page.

4.3 Potential Choices

4.3.1 *Scrolling Item List View*

The items would be displayed in a list view. As the user scrolls down, more items would appear. If they scrolled back up, previous items would load. This allows for one column of items to be displayed. The list view allows for more information to be included in the item, such as a brief description along with the image, title, and price.

4.3.2 *Scrolling Item Grid View*

The items would be displayed in a grid view. Two columns of merchandise would be displayed. Since there is more merchandise on the page, only the image, title, and price would be displayed. If the user scrolled down, new merchandise would appear. If the user scrolled back up, previous merchandise would appear.

4.3.3 *Swiping Single Item View*

A single item view displays just one merchandise result per page. If the user wanted to see a new item, they would have to swipe to have a new item appear. If they swiped left, the previous item would appear. This cannot keep track of previously seen items easily, since only one item is displayed per page.

4.4 Discussion

In contrast to the scrolling item views, the swiping single item only displays one item at a time. This makes it difficult for the user to see multiple merchandise results. In contrast to the swiping single item view, the scrolling views make it easy to display new items. The user does not have to tap a button each time they want to display a new result. The scrolling allows the user to have new search results displayed without changing their motion. Compared to the scrolling item list view, the scrolling item grid view will display more search results. It displays multiple columns of merchandise rather than just the one.

4.5 Conclusion

We chose choice two, scrolling item grid view, because of the benefits of the view. It allows for the user to see many items per page. They do not have to click to view new pages, and can continue scrolling and have new merchandise appear. This meets our criteria the best.

REFERENCES

- [1] Google, "Firebase authentication." Available at <https://firebase.google.com/docs/auth/> (2017/11/13).
- [2] Facebook, "Add facebook login to your app or website." Available at <https://developers.facebook.com/docs/facebook-login/> (2017/11/13).
- [3] Google, "Google sign-in." Available at <https://developers.facebook.com/docs/facebook-login/> (2017/11/13).
- [4] eBay, "Order api overview." Available at <https://developer.ebay.com/api-docs/buy/order/overview.html> (2017/11/13).