

CS CAPSTONE REQUIREMENTS DOCUMENT

OCTOBER 27, 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

MEAGAN OLSEN

Signature

Date

WILLIAM SIMS

Signature

Date

KIARASH TEYMOURY

Signature

Date

Abstract

This document contains specific requirements that we will fulfill in order to successfully complete our project. An iOS application will be created that utilizes eBay's public APIs to help understand the eSports market and shopping opportunities. Our challenge is to target millennial gamers and connect to a variety of different APIs. The eBay public buying APIs will allow users to search for and purchase eSports merchandise within the application. There will be a social component that allows users to learn more about upcoming events by utilizing Twitter APIs to display information from relevant eSports social media accounts. The primary goal of the app is to make it easier for consumers to purchase eSports merchandise and find information about upcoming eSports related events.

CONTENTS

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, acronyms, and abbreviations	3
1.4	References	4
1.5	Overview	4
2	Overall description	4
2.1	Product perspective	4
2.1.1	System interfaces	4
2.1.2	User interfaces	4
2.1.3	Hardware interfaces	4
2.1.4	Software interfaces	5
2.1.5	Communications interfaces	5
2.1.6	Memory constraints	5
2.1.7	Operations	5
2.1.8	Site adaptation requirements	5
2.2	Product functions	5
2.3	User characteristics	5
2.4	Constraints	5
2.5	Assumptions and dependencies	6
2.6	Apportioning of requirements	6
3	Specific requirements	6
3.1	External interface requirements	6
3.1.1	User interfaces	6
3.1.2	Hardware interfaces	6
3.1.3	Software interfaces	7
3.1.4	Communications interfaces	7
3.2	Functional requirements	7
3.2.1	Functional requirement 1: Browse	7
3.2.2	Functional requirement 2: Search	7
3.2.3	Functional requirement 3: Log-in	7
3.2.4	Functional requirement 4: Log-out	7
3.2.5	Functional requirement 5: Favorites	7
3.2.6	Functional requirement 6: Un-favorite	7
3.2.7	Functional requirement 7: Checkout	8
3.3	Performance requirements	8
3.4	Design constraints	8

		2
	3.4.1 Standards compliance	8
3.5	Software system attributes	8
	3.5.1 Reliability	8
	3.5.2 Availability	8
	3.5.3 Security	8
	3.5.4 Maintainability	8
	3.5.5 Portability	8
3.6	Stretch goals	9
	3.6.1 Stretch goal 1: Cart	9
	3.6.2 Stretch goal 2: Publishing to the iTunes Store	9
	3.6.3 Stretch goal 3: Teams as a category	9
	3.6.4 Stretch goal 4: Peripherals as a category	9
	3.6.5 Stretch goal 5: Facebook API integration	9
4	Appendixes	9
4.1	Gantt Chart	9

1 INTRODUCTION

Our project involves creation an iOS application utilizing eBay public buying APIs to help understand the eSports market and shopping opportunities. Our challenge is to target millennial gamers and connect to a variety of different APIs. We will utilize event and eSports scores APIs to display information desired by gamers. The primary purpose of the app is to make it easier for customers to purchase eSports merchandise and find information about eSports events.

1.1 Purpose

The purpose of the iOS mobile application is to provide access to esports merchandise sold by eBay and display information about upcoming eSports events. The application will provide its audience a wider range of shopping opportunities by connecting them with products for sale that really interest them by providing a relatively streamlined easy to use search to purchase process with enough flexibility to provide a pleasant shopping experience. Part of this pleasant experience includes a fun social component such as Twitter where they can log into from with the application. This will perhaps make the application and eBay seem like a more attractive purchasing option as it adds a personal component that they can relate to. It will also help to consolidate esports information into one place with ease of use so that they can receive updates about their favorite games. The goal is to build an application outside of eBay that uses its various public APIs and integrates with Google Firebase for authentication. The application will also test the ability to develop on those APIs. The application will also help eBay to explore the eSports market as a corporation. The application will also demonstrate an ability to adhere to current human interface guidelines detailed in the Apple developer guidelines. It will also develop us as developers to prepare us for larger scale projects that we can expect to see in our future careers by providing the opportunity to work hands on immersed in a corporate environment with senior developers as valuable resources to assist in this transition.

1.2 Scope

Our main goal is to create an application that helps us learn more about the eSports market and new shopping opportunities for eBay. The software will allow users to purchase eSports merchandise that is being sold on eBay. We want to create an environment for millennial gamers to have easier access to eSports content such as tournament scores and events around the world.

1.3 Definitions, acronyms, and abbreviations

- **eSports:** Multiplayer gaming event thats played competitively through out the world among professional gamers.
- **Xcode:** IDE provided by Apple to test, compile and develop applications for the iOS and Mac OS X.
- **Swift:** Compile programming language provided by Apple to build application for the iOS and Mac OS X. Swift was introduced on October 22, 2014 to replace Objective-C.
- **Firebase:** Real time databases provided by Google which offers many other services such as cloud messaging, storage and notifications.
- **iOS:** An operating system used on Apple's mobile phones.
- **OS:** Operating system
- **query:** To retrieve specific information from a database.

1.4 References

- <https://go.developer.ebay.com/>
- <https://firebase.google.com/>
- <https://developer.twitter.com/en/docs>

1.5 Overview

This document contains our overall description of how the product will be structured, the specific requirements of the functions, performance, and user interaction, and details on the software system attributes.

2 OVERALL DESCRIPTION

This application will be created on iOS 11, and use eBay's public buying APIs. User's will be able to view upcoming eSports events and related merchandise.

2.1 Product perspective

Our product is independent. It is a standalone application not related to eBay that utilizes their public APIs.

2.1.1 System interfaces

- The product will be an independent iOS 11 application that makes use of eBay public buying APIs.
- It will drive traffic to eBay without users ever leaving the application, but it is still not part of eBay's larger system.
- Users will be able to search for and purchase eSports merchandise from eBay within the application. Google Firebase will be used for backend databases and authentication so that users can login to the app with their eBay accounts.
- The application will use an eSports API that has yet to be determined to display scores and information about current eSports events.

A stretch goal will be utilizing Facebook and Twitter APIs to display posts from targeted eSports accounts such as Riot Games and Blizzard so users can stay up to date with ongoing events.

2.1.2 User interfaces

The user interface will have a screen for displaying eSports scores for different games and related merchandise. There will be a login form for signing into an eBay account and search functionality for discovering relevant eSports merchandise that is actively being sold on eBay. Merchandise will be displayed as cards that contain the item name, a picture, and price. There will be a shopping cart screen that contains order information such as quantity, price, and item names. The interface will also contain fields for entering payment information and components for completing the checkout process.

2.1.3 Hardware interfaces

The application will support iPhones running iOS 11.

2.1.4 Software interfaces

The application will utilize the eBay buy APIs which consists of the marketing, browse, and order APIs. The buy APIs are currently publicly available and will require an eBay developers program account. We have not yet discussed with our client the eSports API we will be using to display score information. The Twitter APIs are publicly available.

2.1.5 Communications interfaces

The application will require an internet connection to connect to the eBay and eSports APIs that will be used.

2.1.6 Memory constraints

The application will run within the memory constraints of recent iPhone devices such as the 5,6, 7 and 8.

2.1.7 Operations

Users will be required to sign into their eBay accounts in order to purchase eSports merchandise and complete the checkout process. Viewing items that are actively being sold will not require the user to be logged in.

2.1.8 Site adaptation requirements

The user will require a recent iPhone device such as the iPhone 5,6, or 7 to run the application. The application will need to accommodate different screen sizes for both iPhone plus and normal sized models.

2.2 Product functions

The application will allow users to browse esports merchandise. It will also have filters to control which merchandise is displayed. It will allow users to perform searches for esports merchandise related to one of a select group of commonly played esports games at a time to be chosen by the user. It will advertise merchandise related to current esports events, and it will save these searches in a favorites feature. It will allow the user to sign on and maintain a personal account. It also will allow users to see twitter feeds related to esports events. It will allow the user easy navigation through different tabs.

2.3 User characteristics

The intended user of this application are those looking to purchase eSports related merchandise. The main audience will be those who are interested in eSports merchandise, such as gamers looking to support their favorite teams or other individuals looking to purchase this merchandise as a gift. This encompasses a wide age, from middle-aged students all the way to grandparents. There is also a large educational range for our intended users, as those who are into eSports and would be using the application do not need to have a high school or college degree to do so. The intended user has enough technical skills to be able to navigate a mobile application and complete a transaction. They should be able to familiarize themselves with the interface in a short period of time without needing specific instructions for how to navigate.

2.4 Constraints

The application will be dependent on eBay buy APIs in order to display relevant eSports merchandise. We will also be limited to functionalities within Google Firebase for our backend databases and authentication.

2.5 Assumptions and dependencies

The end application will load within a reasonable amount of time. The Twitter API will be used inside the application for the social component mentioned above. The application will meet basic functional/performance requirements as we have defined them. The main source of the application will be implemented in the Swift programming language and will adhere to the Apple Developer Standards. The application will incorporate the eBay public APIs. All development will take place outside the eBay domain. The application will use Google Firebase for back end authentication.

2.6 Apportioning of requirements

In a later version, the application will allow for checkout and purchase of a single product or even multiple products. It will also display individual information about a single product. It will display apple ads on the bottom of each tab. It could possible display used products for sale instead of just new products. It may include a shopping cart. It could allow searches by specific esports teams or participants. It might later have the functionality to sort the returned products. It may be able to categorize returned products.

3 SPECIFIC REQUIREMENTS

This section goes into details of the applications functions, the external interface, the software systems attributes, and the constraints.

3.1 External interface requirements

We will be using the eBay public APIs as a starting point for our application. We will also be integrating for Twitter APIs and or Facebook APIs for the social component. We will be also integrating with Google Firebase as a back end for authentication.

3.1.1 User interfaces

- Users will be able to interact with a sign in pop up module where they can Log in/ Register on the home page. Users are able to see their eSports event along with all the related merchandise. They will be navigated to eBay app to complete their purchase if interested.
- Home page will also contain favorited eSports events.
- The brows page will contain category of different events and games where users can select to choose what they are interested.
- They will be given the option to filter the search for their own personal preference
- Users will recieve and error message pop up if not connctected to the internet
- No search or events found will also be shown if no objects are found in the API.

3.1.2 Hardware interfaces

- The app will support all of iPhone devices currently running iOS 11 since the whole application will be written with latest version of Xcode and Swift.
- Application will use some of iPhone intergrated hardware to interact with the user. Local notifications and many other thing provided by apple will be intergrated.

3.1.3 *Software interfaces*

- Our Application will run all the fetching in the background to retrieve data given by the user to display all on the home page and the browse page.
- The Firebase backend will be used to let users register and log in by using an authentication system on the sign in modal.
- Purchasing an item will make a transition call to open the eBay app to complete users purchase.
- social APIs will be displayed on event page to share the event.

3.1.4 *Communications interfaces*

- The application will connect to the internet.
- An error message will be displayed when the app is not connected to the internet.

3.2 **Functional requirements**

3.2.1 *Functional requirement 1: Browse*

- The user will be able to select the game category from the initial browse screen.
- The interface will display multiple games (currently undetermined) to view merchandise.
- When a user selects a specific game, they will be able to view related eBay merchandise..
- The user will be able to select the events category from the initial browse screen.
- The interface will a number of undetermined events to view merchandise.
- Events that have already passed will not be listed.
- When a user selects a specific event, they will be able to view related eBay merchandise.

3.2.2 *Functional requirement 2: Search*

- The user will be able to select a certain category to query such as by game or event.
- The user will be displayed relevant merchandise to that category that they queried.

3.2.3 *Functional requirement 3: Log-in*

- The user will be able to log-in to a pre-existing account to access their profile.
- If an account does not already exist, they will be able to create one.

3.2.4 *Functional requirement 4: Log-out*

- The user will be able to log-out of their account.

3.2.5 *Functional requirement 5: Favorites*

- The user will be able to favorite games.
- Their favorited games will appear on their home screen.

3.2.6 *Functional requirement 6: Un-favorite*

- The user will be able to unfavorite their previously favorited games.
- Unfavorited games will no longer appear on the home screen.

3.2.7 Functional requirement 7: Checkout

- The user will be able to purchase an item from eBay.

3.3 Performance requirements

The completion of project requirements is more important than raw performance. Therefore, most of our performance metrics will be qualitative requirements. This is because gated networks can make it difficult to measure performance accurately and we will be limited by factors outside of our control. Our performance will be measured by our ability to integrate with eBay APIs and Firebase. The application should also be able to display twitter feeds using Twitters API to gather eSports event data such as dates, details, and locations. Currently, our goal to shoot for a maximum of 2 second page load. Metrics may be tweaked in the future to avoid being overly ambitious on performance.

3.4 Design constraints

We will be designing our own UI for each of the displays on the application.

3.4.1 Standards compliance

We will be obtaining proper licensing from Apple to develop this native application since our application will developed specifically for the protected iOS system.

3.5 Software system attributes

This section includes how reliable, available, secure, maintainable, and portable our application will be.

3.5.1 Reliability

This application should be able to display the latest up and coming eSport's events, and not display an event that has passed. The application should also show relevant merchandise when the user selects a specific game or event to shop.

3.5.2 Availability

The application will only be available to user's who have iPhone devices. These devices must have the latest iOS 11 OS. Those users must also be connected to the internet to use our application.

3.5.3 Security

We will be using Firebase security system that is provided by Firebase. Some of the APIs that we are using also provide https security calls.

3.5.4 Maintainability

We will be maintaining the software by making sure everything we have is fully functional and optimized. Also, we will make it our priority that our Firebase backend runs pretty smoothly when fetching to avoid delays and high memory usage.

3.5.5 Portability

This application will not be very portable. Since it will only be usable on iPhone devices with iOS 11, this means iPhones without the latest OS or other devices will not be able to use it.

3.6 Stretch goals

3.6.1 Stretch goal 1: Cart

- The user will be able to add items to their cart.
- The user will be able to view their cart.
- The user will be able to purchase multiple items from their cart at one time.

3.6.2 Stretch goal 2: Publishing to the iTunes Store

- The application will be published to the iTunes store.
- Users will be able to download the application from the iTunes store.

3.6.3 Stretch goal 3: Teams as a category

- eSports teams will be added as an additional query category.

3.6.4 Stretch goal 4: Peripherals as a category

- Peripherals including mice, headsets, and keyboards will be an additional query category.

3.6.5 Stretch goal 5: Facebook API integration

- The application will utilize Facebook's API.
- The application will allow user's to share upcoming eSports event's to their time line.

4 APPENDIXES

4.1 Gantt Chart

