CS Capstone Problem Statement

October 9, 2017

eBay iOS eSports Application

CS 461 - Fall 2017

Prepared for

eBay Inc.

Luther Boorn _____ _____
Signature *Date*

Prepared by

Group 17
Skill Capped IRL

Will Sims _____ _____
Signature *Date*

Katherine Bajno _____ _____
Signature *Date*

Meagan Olsen _____ _____
Signature *Date*

Kiarash Teymoury _____ _____
Signature *Date*

Abstract

We were unable to meet with our client before the rough draft due date, so this document was created using only the information provided on the capstone website. Many assumptions are made about the core features of the application. My team has scheduled to meet with our client for two hours on 10/12/16 and the problem statement will be updated based on what is discussed during the meeting.

Our project involves creation an iOS application utilizing eBay public buying APIs to help understand the eSports market and shopping opportunities. Our challenge is 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 app will include a social component that makes use of Twitter and Facebook APIs which will allow gamers to share content on different social media platforms. We will be developing in Swift and using the Google Firebase development platform for our backend to create an application that can be published on the iTunes store.

Contents

1 Problem Definition

eBay would like make use of the recently released public APIs in order to explore the eSports market and target new customers. Currently, eBay isnt doesnt have any products that attract millennial games. The focus of the iOS eSports project is to develop an application that contains useful information about eSports and allows customers to purchase products on eBay.

2 Proposed Solution

The core features of the application will rely on the recently released marketing, browse, feed, and order APIs through the eBay developers program. The marketing API will retrieve products based on the category and metric. The feed API will display details listed in a given eBay category and site. The browse API retrieves details of an item and the order API allows customers complete checkout. Since the focus of our application is the eSports market, we will use video games and consoles as our primary eBay shopping category. This will allow customers to purchase video game items that are relevant to the specific eSport that is displayed on the site. We will implement our solution as a native iOS app using Swift and Google Firebase.

3 Performance Metrics

There are many different eSports scores that will be displayed in the application and we want the initial eSports scores to load in under XX milliseconds. Items displayed through the eBay API should take no longer than XX milliseconds to update and an order should be processed in no longer than XX milliseconds. No interaction in the interface should take longer than 5 seconds. The app crash rate should be under XX% and API latency should be optimized to an X second response time. Since our app will eventually be published to the iOS App Store, there are a number of user, usage, and demographic performance metrics that may be useful to track after the completion of our project.