KBPartPicker

System Design Document

Team Members
Rejan Quebral r_quebral@u.pacific.edu

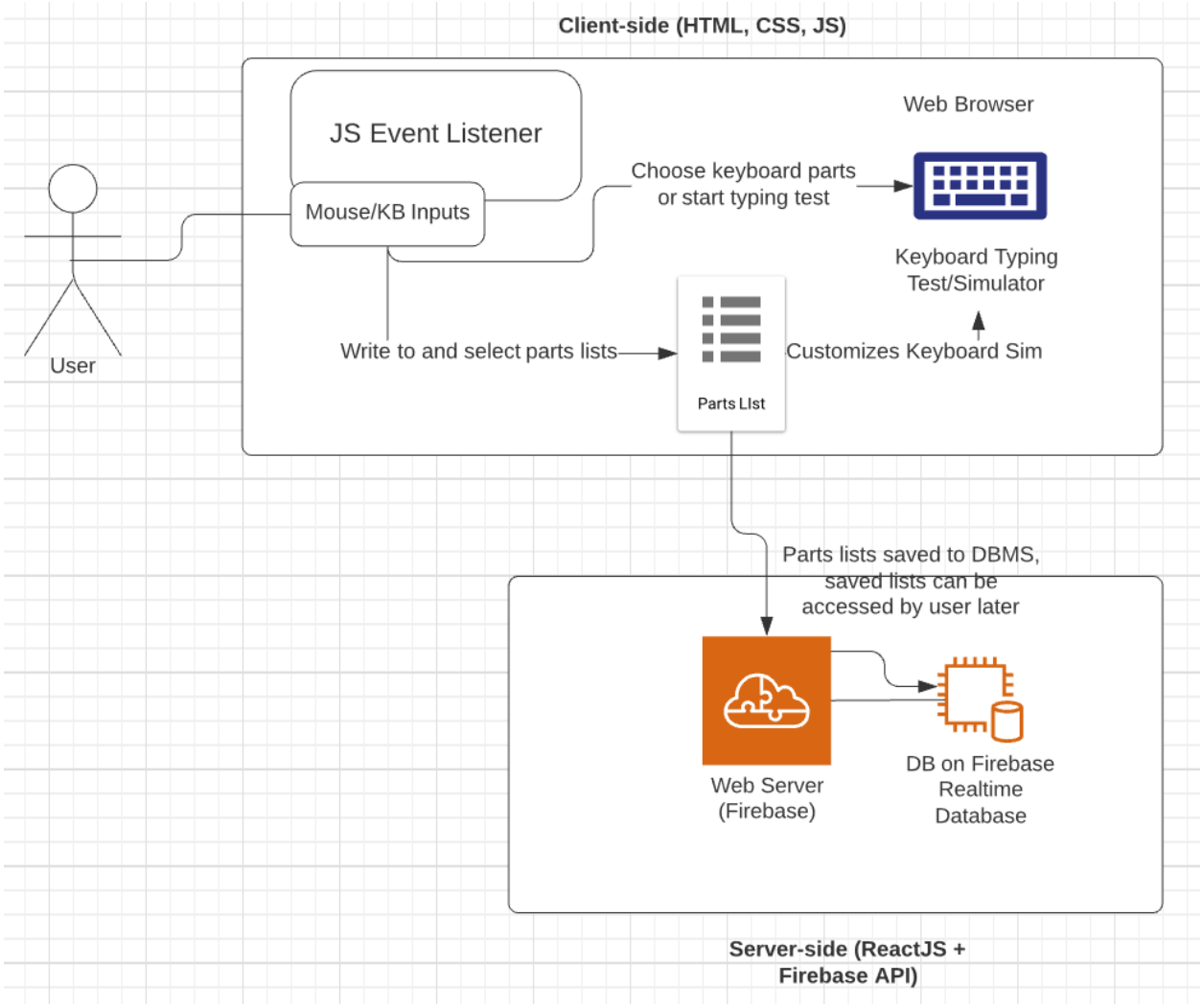
Instructor
Dr. Canniff

Github URL: https://bit.ly/3omhoks
Website URL: https://kb-part-picker.web.app

Last revised: March 18, 2022

Table of Contents

System Architecture



Software Modules

On the client side, the user will interact with the web application with the help of JavaScript EventListener() functions to get keyboard inputs to help with keyboard typing simulations. *KBPartPicker* will be displayed with HTML5 and CSS through whatever web browser the user chooses, though complete compatibility will be ensured with Mozilla Firefox first.

On the server side, PHP will be used to maintain the Apache web server and MySQL database applications.

Hardware Components

The web application will run on the user's computer through a web browser of their choice, and the web server and database will be maintained by Firebase. As *KBPartPicker* will be compatible with Mozilla Firefox first, users should make sure their computers meet the minimum requirements outlined here: https://www.mozilla.org/en-US/firefox/96.0.3/system-requirements/

User Interface

The user interface will be created with modern HTML5, CSS, and JavaScript practices as of 2022. Users will interact with the web application mainly through JavaScript EventListener() functions and will be able to select keyboard parts through dropdown menus.

Interfaces to External Systems

KBPartPicker will interact with the web server through Firebase API calls, and any requests to the database will be handled by the web server.

Hardware, Software, and System Requirements

Hardware

The minimum requirements correspond with any Windows computer that can run Mozilla Firefox:

- Pentium 4 or newer processor that supports SSE2
- 512MB of RAM / 2GB of RAM for the 64-bit version
- 200MB of hard drive space

Users will also need an internet connection. Whether they connect through wi-fi or ethernet is irrelevant as *KBPartPicker* will not require intensive loads such as software downloads.

Software

As *KBPartPicker* is a web application, it will initially be developed to display properly in the Mozilla Firefox browser but should time allow, Chromium-based browsers should also be supported. There are no plans, however, to support any mobile devices.

System

KBPartPicker will be developed with the most recent version of Mozilla Firefox available as of the writing of this document, Firefox 96.0.3, which requires a Windows 7 or higher operating system.

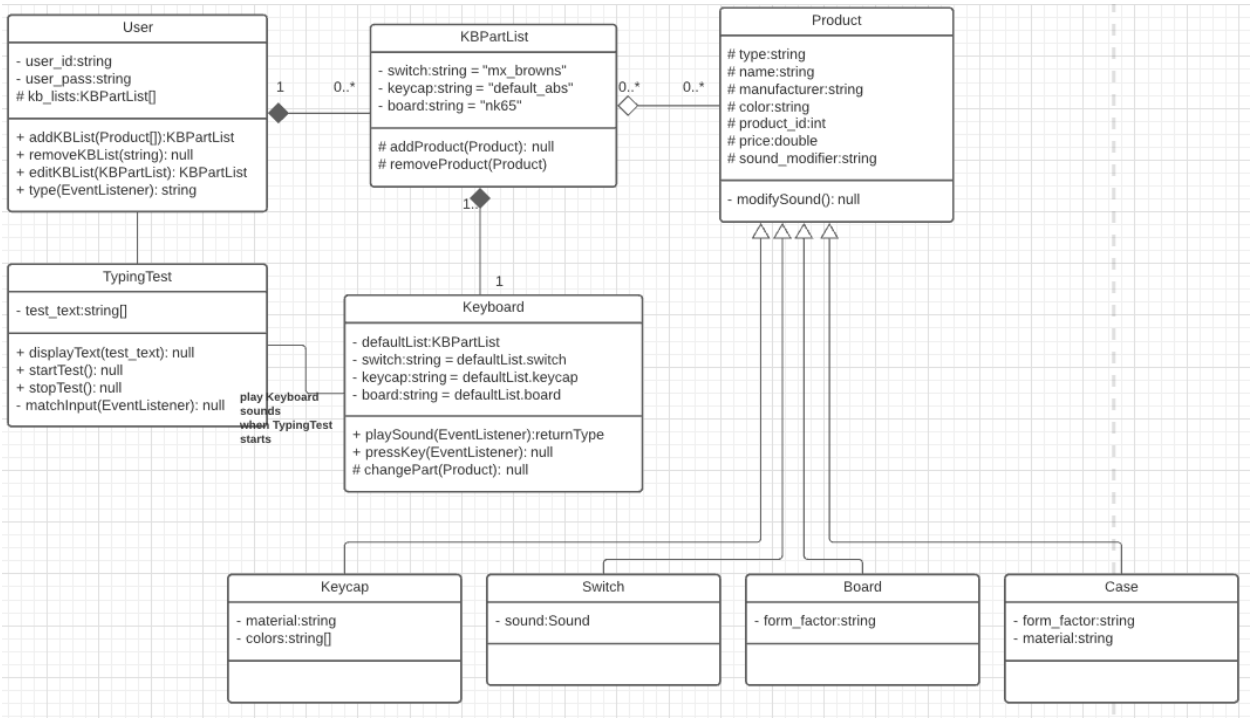
External Interfaces

Firebase

This project will be hosting a web server and database management server on Firebase. Documentation can be found here:
https://firebase.google.com/docs

Software Design

Class Diagram and Class Specifications



Design Considerations

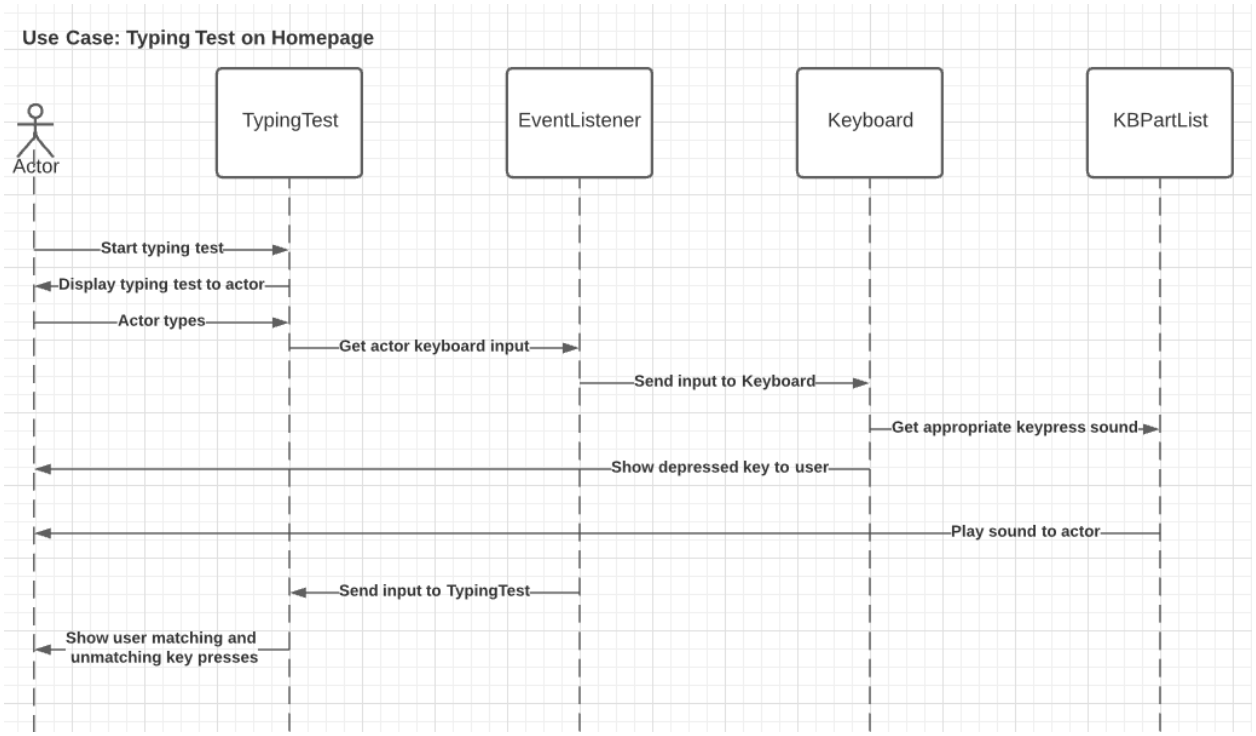
User was designed for flexibility, as there may be additional implementation in the future to allow manufacturers to upload their own sounds. For now, however, to

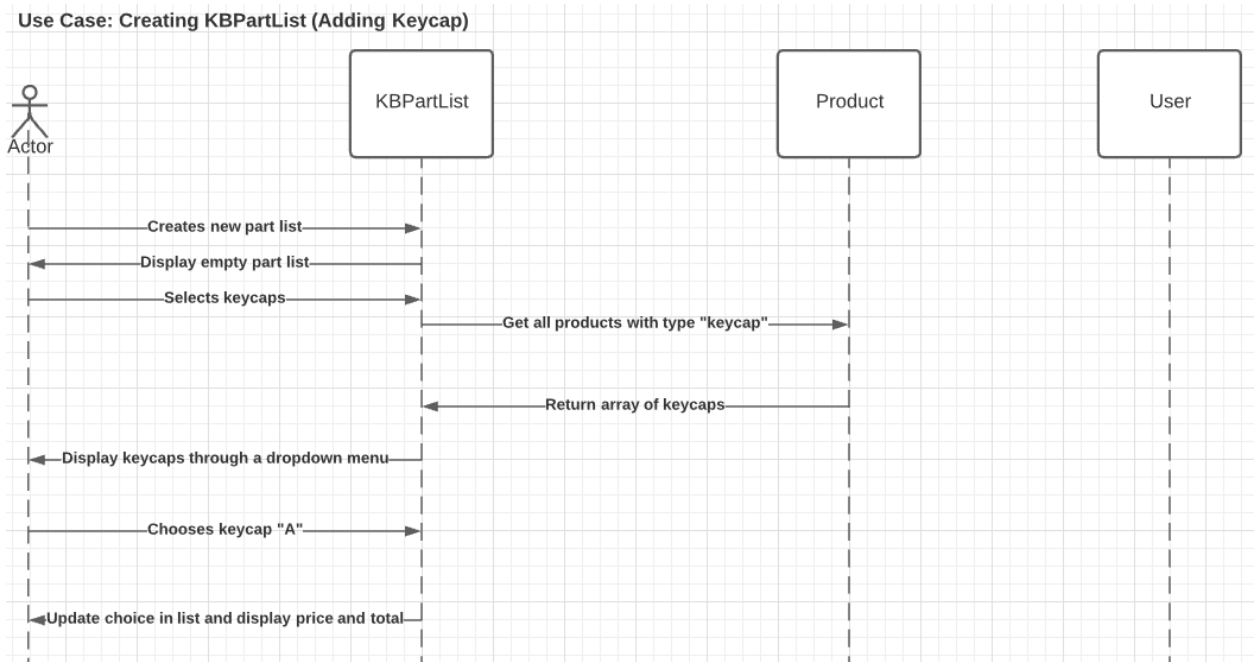
reduce complexity this will stay an additional feature and ***User****s* will only be able to create and modify lists and interact with the ***Keyboard*** through ***TypingTest***.

Product was created to increase abstraction, as ***Keycap, Switch, Board,*** and ***Case*** share multiple of the same properties.

TypingTest and ***Keyboard*** were designed with loose coupling in mind. ***TypingTest*** will gather user input which will be translated to ***Keyboard***, while ***Keyboard*** will play appropriate keypress sounds and take on the look of an actor's chosen ***KBPartList***.

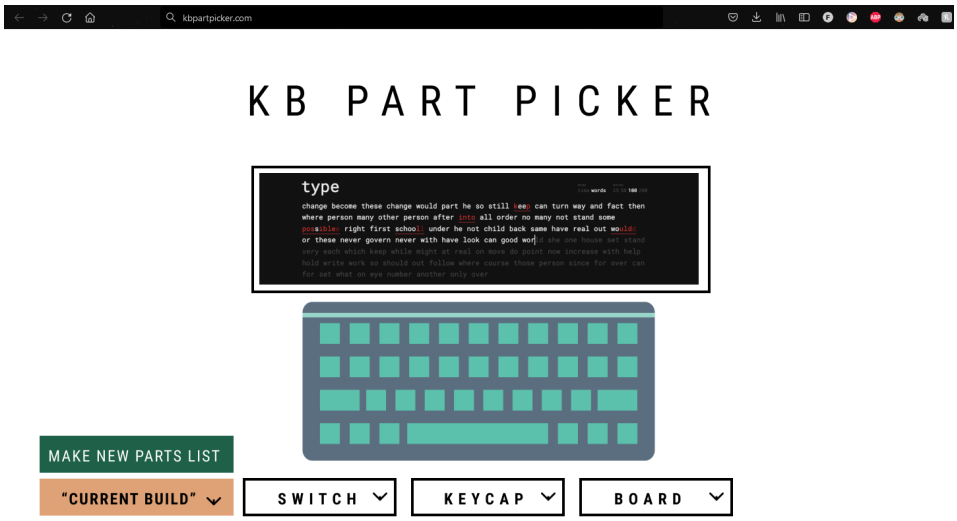
Interaction Diagrams





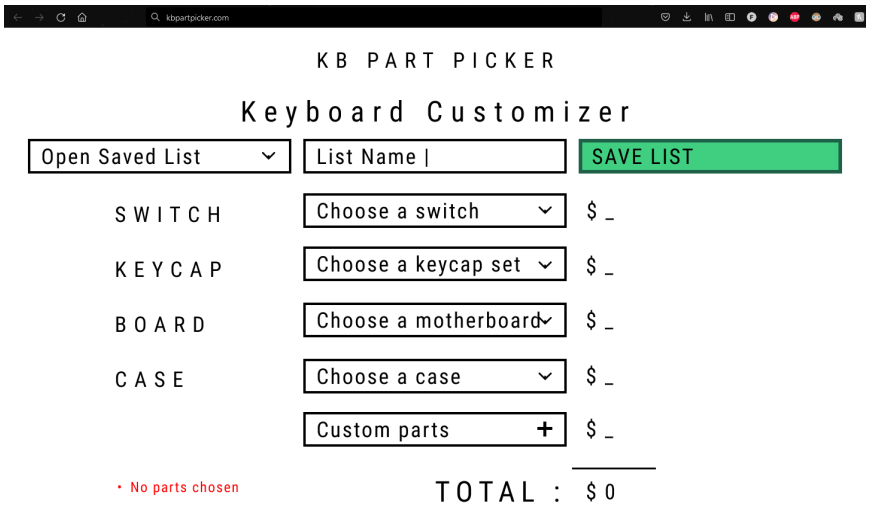
User Interface Design

Homepage



When first entering the site, users will be brought to a page with a pre-built keyboard list, the option to customize the keyboard switch, keycap, and board, and a button to bring them to a page to create a new keyboard parts list. They will also be able to conduct a typing test to hear the custom keyboard in action.

Parts List Creation Page



If the user clicks the “MAKE NEW PARTS LIST” button, they will be brought to a page similar to PCPartPicker, where they will be able to create and save a list of keyboard parts, pick an already existing parts list, and see the price of their current list’s items

and the total price. Should any compatibility issues arise, the site will alert the user with red text at the bottom left. If the user wants to return to the homepage, they can click on “KB PART PICKER” at the top of this page

Glossary of Terms

Terms	Definitions
Abstraction	Removing aspects from a class to reduce it to its most essential characteristics.
Client Side	In a client-server relationship, this refers to anything that is displayed to the client or is performed on the client’s system
Database Management Server (DBMS)	Software which stores, retrieves, and runs queries on data.
Loose Coupling	Designing dependent classes to rely on each others’ data as least as possible.
Server Side	In a client-server relationship, this refers to operations performed by a server
Web Application	An application stored on a remote server that is transmitted through the internet by a web browser.
Web Browser	Software used to access and display websites and web applications.
Web Server	A computer that stores website components and software to accept and transmit data for the website.

References

Firebase Documentation (n.d.). Retrieved February 28, 2022, from https://firebase.google.com/docs

Firefox 96.0.3 system *requirements*. Mozilla. (n.d.). Retrieved February 6, 2022, from https://www.mozilla.org/en-US/firefox/96.0.3/system-requirements/