WASHINGTON STATE UNIVERSITY
VANCOUVER

CS 320 Course Project - Software Design Document

for

The Fellowship of The Keys

Prepared by

Group Name: Fellowship of the Keys

Jonathan Meisner	**011520302**	**jonathan.meisner@wsu.edu**
Kyle Stoneberg	**011700532**	**kyle.stoneberg@wsu.edu**
Joe Boothby	**011445002**	**joseph.boothby@wsu.edu**
Elliott Long	**011686651**	**elliott.long@wsu.edu**

Date: 12/16/2020

CONTENTS II

1 Introduction

In the world of computer components, one of the most paramount and ubiquitous of them all is that of the keyboard. And within this world are numerous different varieties, from dome and optical, to that of the mechanical switch. Further still, within the world of the mechanical switch, there are innumerable variants, a somewhat daunting aspect that can leave users searching out mechanical keyboards with specific needs in mind confused and annoyed due to the lack of easy-to-parse sites of information. The project Fellowship of the Keys aims to eliminate that issue by providing a simple to use site in order to parse through differing varieties of keys, all depending on desired qualities, be that audibility of clicks, actuation strength required, or otherwise.

1.1 Project Overview

The Fellowship of the Keys is a web application designed to create a simple, unified resource for users to read on mechanical key switches. This document is an extension of the software requirements specification and serves to outline the typical use cases and class designs via diagrams. Further, it features a sequence diagram to illustrate the event flow during usage.

1.2 Definitions, Acronyms and Abbreviations

Clicky - Loudly audible
Keyswitch - See switch
Marketplace - Sites where selected keyboards can be purchased (mechanicalkeyboards.com)
Switch - Mechanical switch
URL - Uniform Resource Locator

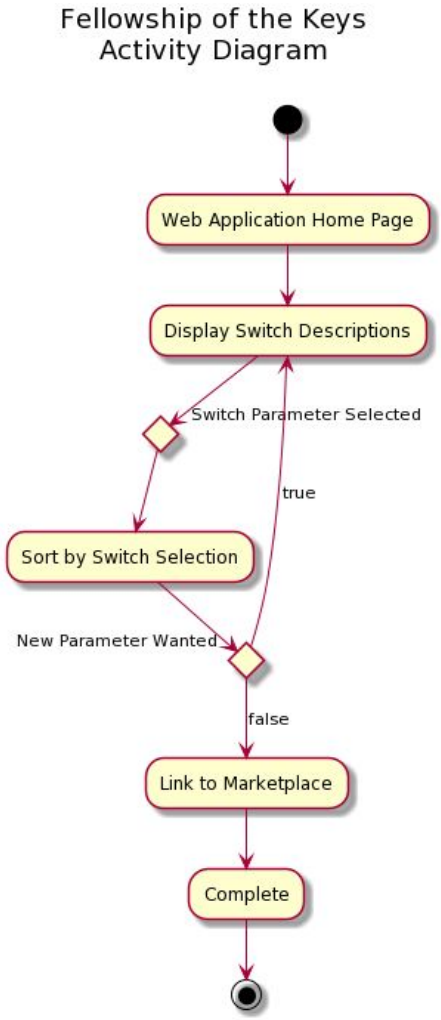
1.3 References and Acknowledgments

Unneeded - no references are made in this document.

2 Activity Diagram(s)

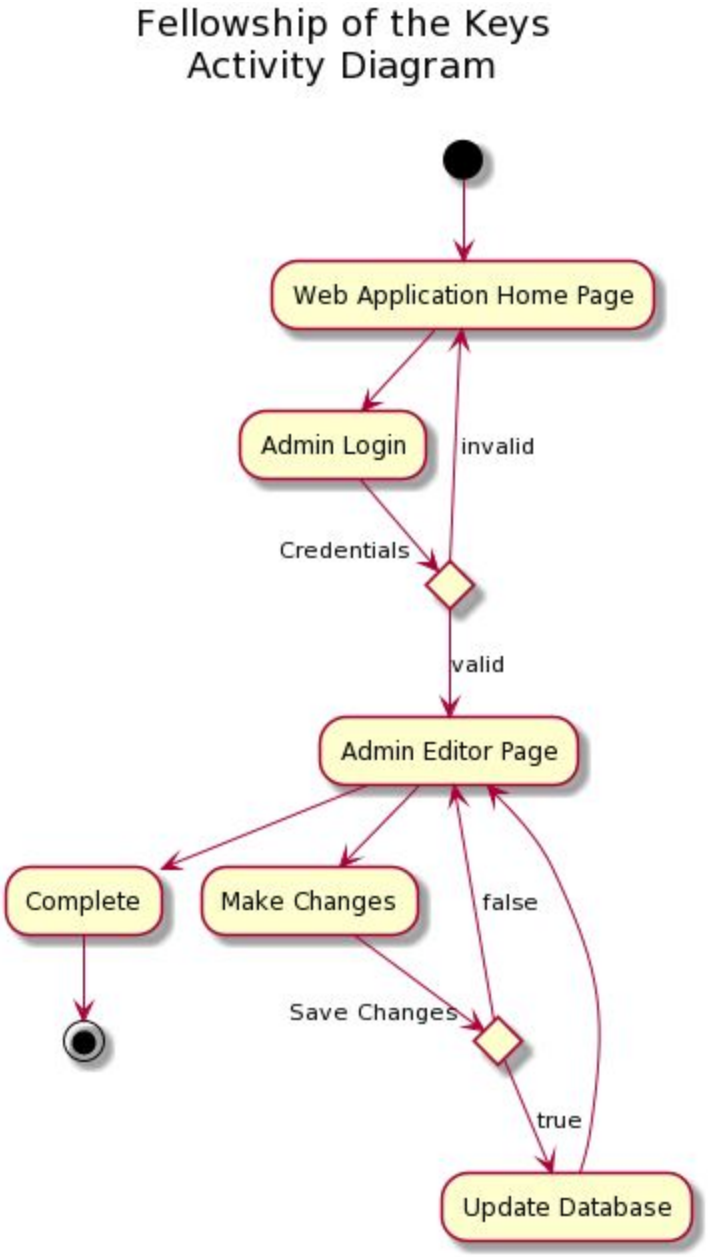
2.1 Activity Diagram for Typical User Use Case

This activity diagram covers the typical user use case of researching mechanical key switches, and then purchasing a keyboard. The user accesses the web application, then looks through different switch types, changing an attribute to order by until they decide on a specific switch. They are then provided with a link to the trusted marketplace that displays the correct keyboard and the correct pricing order. The start state is accessing our website, and the end state is being redirected to the marketplace.



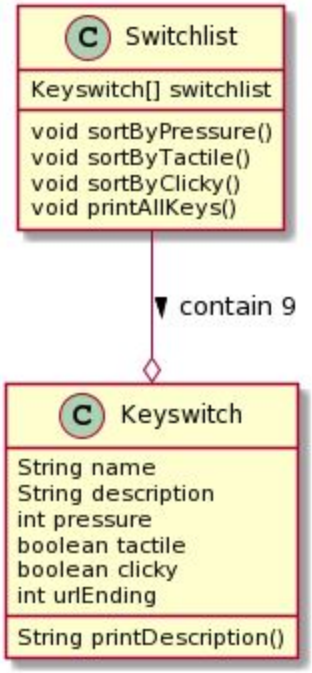
2.2 Activity Diagram for Admin Use Case

This activity diagram covers the admin use case of our webapp. The administrator accesses the website home page, then logs in to an admin account via a username/password combination. This admin function allows the administrator to manipulate the data that is stored in the Switchlist and Keyswitch classes in order to add or remove Keyswitches and change the marketplace URL for the hyperlinks. The start state is accessing the homepage of the website, and the end state is successfully modifying the web app.



3 Class Diagram(s)

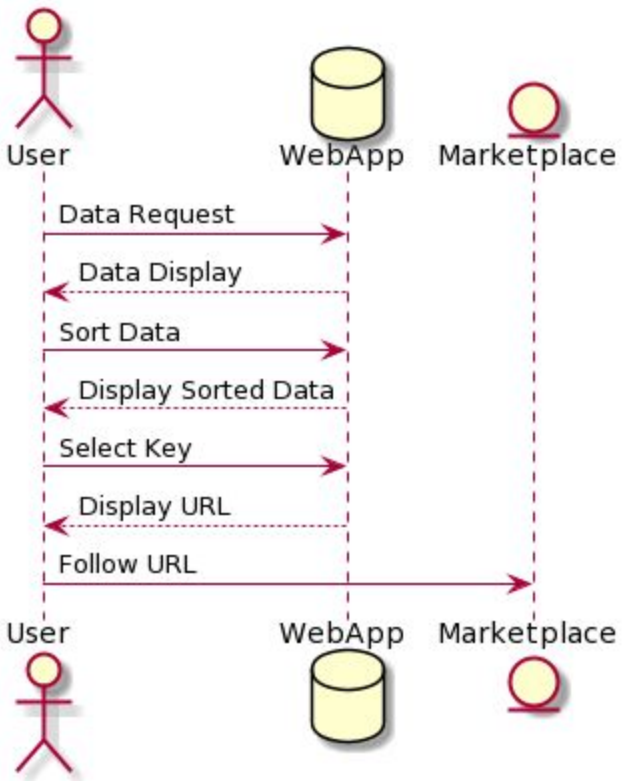
3.1 Class Diagram for Storing Switch Data



Switchlist: A class that contains all instances of Keyswitch and methods for sorting and printing. There will be one master instance of the Switchlist object, and it will be responsible for sorting the array of Keyswitch objects based on the attributes of the Keyswitches. The printAllKeys method of Switchlist will display the Keyswitches in the newly sorted order.
Keyswitch: A class that defines the attributes of a switch. These will be immutable objects hardcoded into the JavaScript file. Each Keyswitch is responsible for displaying its own attributes using the printDescription method.

4 Behavioral Diagram(s)

4.1 Typical Usecase Sequence Diagram



This sequence diagram is describing the typical use case. The first request for data by the user happens when the user accesses the web site and the key switches are displayed. Sort data is a function call to the sortByPressure(), sortByTactile(), or sortByClicky() methods in Switchlist. The Web App then displays the sorted data back to the user with a method call to printAllKeys(). Display URL happens automatically when a user clicks on a key switch type on the webpage by combining the static link the marketplace with the integer URL ending attribute to Keyswitch. This happens because the URLs for each switch in the marketplace are integer suffixes to the URL of the marketplace. Finally, the user follows the URL to the marketplace, thereby completing the use case.

Appendix A - Group Log

9/30:

Catalog build to be object oriented javascript or java with additional modules or packages to be determined

Git Repo (keyboard-webapp) Created & Invited Group Members

10/2:

First Team Meeting

Team Arrangement Documentation Filled out.

Jonathan Meisner appointed as Initial Project Lead

Submitted Document via BB ~4:45pm 10/2/2020

10/26:

Met With Team

Discussed structure of project

~1 hour 15 minute meeting time

11/4:

Met With Team

~ 1 hour

11/20:

Put together the Software Design Document with team via Zoom

~ 1.5 hour