**Touchscreen Interface**

# **Maintenance Documentation**

**Version 1.0**

**11/03/2018**

## 

## 

## 

## 

## 

## 

## 

## **Document Number**: 1.0

**Software Revision History**

## The following table records information regarding changes made to the Touchscreen Interface System over time. To provide information about the controlling and tracking of this artifact, please refer to the Record of Changes section of this document.

**Table 1 - Record of Software Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Author/Owner** | **Description of Change** |
| 0.1 | 4/28/2018 | Jonah Kubath | Baseline Version |
| 0.2 | 7/21/2018 | Matt Peter | Firefox Restrictions |
| 0.3 | 7/29/2018 | Michael Riess | * Electron Application created |
| 0.4 | 9/21/2018 | Matt Peter | * Improved Firefox Restrictions * Created carousel for displays products in the Vending application |
| 0.5 | 9/24/2018 | Michael Riess | * Testing plan developed |
| 0.6 | 10/6/2018 | Matt Peter | Custom keyboard layout for touchscreen |
| 0.7 | 10/15/2018 | Jonah Kubath | * Deployment Scripts created |
| 1.0 | 11/3/2018 | Michael Riess | Vending Application built for deployment |

**Record of Document Changes**

The table below is used to record information regarding changes made to this document over time.

**Table 2 - Record of Document Changes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Author/Owner** | **Description of Change** |
| 1.0 | 11/03/2018 | Michael Riess | Initial Instance |

## 

## **Introduction**

Screen management system and associated applications for kiosk-style vending machine with promotional capabilities, and interactive features for the solicitation of patronage. Such system is designed to run in the Linux Ubuntu 16.04 LTS ([Xenial Xerus](https://wiki.ubuntu.com/XenialXerus)). System operation restricted to use of E&T L Series Infrared Touchscreen. Computer operation restricted to any hardware capable of meeting minimum Linux Ubuntu 16.04 LTS (64-bit) requirements. This document is intended for use by the license holder of the touchscreen interface system, and the maintenance and active development of such system insofar as dictated by said license holder. While there is overlap between the document contents, this document is not intended to function as a System Security Plan (SSP), Information Security (IS), Risk Assessment (RA), System Design Document (SDD), Interface Control Document (ICD), Database Design Document, Data Conversion Plan, Release Plan, Version Description Document (VDD), Implementation Plan, Test Plan, or User Manual.

## **Referenced Documents**

Provided below is the identifying information for all documents used to arrive at and/or referenced within the Maintenance Documentation (e.g., related and/or companion documents, prerequisite documents, relevant technical documentation, etc.).

**Table 3 - Referenced Documents**

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Document Number and/or URL** | **Issuance Date** |
| L\_Series\_Infrared\_Touchscreen-ET\_Tech.pdf | * docs/touchscreen/L\_Series\_Infrared\_Touchscreen-ET\_Tech.pdf * http://eandttouch.com/product-1-1-infrared-touchscreen-en/140846/ | N/A |
| XenialXerus\_ReleaseNotes\_Ubuntu-Wiki.pdf | * docs/Operating System/XenialXerus\_ReleaseNotes\_Ubuntu-Wiki.pdf * https://wiki.ubuntu.com/XenialXerus/ReleaseNotes | 2018-09-30 |
| Ubuntu 16.04 LTS Install Guide | https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows#0 | N/A |

## **System Overview**

Describes the physical and software layout, information flow, and overall functionality of the Touchscreen Interface System.

### **Functional System Overview**

Touchscreen Interface System provides users with access to the internet through a modified web browser, limiting them to “safe” practices by removing administrative capabilities. Additionally, users are subject to displayed advertisements. Lastly, users are able to interact with a vending machine via an intuitive and visually appealing vending application at the center of the touche screen.

### **Application/System Dependency**

Provided below are dependencies of the application/system and the impact.

**Table 4 - Application/System Dependency**

|  |  |  |
| --- | --- | --- |
| **Dependent Application/System** | **Function** | **Impact (If Application is Down)** |
| Live Internet Connection | Information transfer | * Users will be unable to use the web browser * Live product and add updates will be unavailable * Patronage attraction from web access will suffer |
| Firefox Web Browser | User Web Browsing | Patronage attraction from web access will suffer |
| Vending Application | Allow users to purchase products, and display ads | Ability to generate income through patronage of the vending machine, and through advertisements will cease |

### 

### **Physical System Overview**

The Touchscreen Interface System and associated softwares were designed to run on any computer capable of running Ubuntu 16.04 LTS 64-bit desktop distributions. For full functionality, the computer should also have an internet connection. Additionally, USB and HDMI connections to an E&T-Tech L-series Infrared Touchscreen are required for proper function. Specialized OS touchscreen settings have been implemented for proper operation which are specific to said screen make and model. A lack of touchscreen or use of unsupported screen will produce an unusable state.

### **System User Overview**

### **Estimated Users**

### The number of users is not yet known as the product is not in active deployment and has not yet finished production. This section will be updated later when the data become available.

### **Hours of Operation**

The hours of operation are not yet known as the product is not in active deployment and has not yet finished production. It is likely, however, that the system will be in constant operation while not down for maintenance or restocking. This section will be updated later when the data become available.

### **User Impact for System Failure**

Upon hardware failure, the system will almost certainly become inoperable, resulting in patrons being unable to browse the web, view advertisements, or make purchases from the vending portion of the system. Due to a lack of patron interaction, income via the vending portion of the system and the ads will be negatively affected. The theoretical chance of ads not being displayed should be factored in to the sale of ad views/displays.

Upon failure of the web browser, patrons will still be able to interact with the vending portion of the system and view ads. Patrons will, however, be unable to browse the web. This will likely detract from the appeal of the system as a whole, and may result in reduced patron interaction in other areas of the system.

Upon failure of the vending system, ads will also cease to work as they are controlled by the same application. At this point, patrons will likely still have access to the web browser, but will be unable to interact with the comerse based portions of the system.

Upon Internet outage or internet-connection-specific hardware failure, neither the web browser nor the vending system and ads will be down. However, use of the web browser for its primary function -namely, browsing the web, will be impossible. Additionally, active, remote maintenance and control of product, advertisement, and purchase data will be unavailable.

### **Processing Overview**

This section provides information that is applicable to the processing of the system. It Identifies the state(s) and mode(s) of operation. It Identifies the types of inputs/access that can be made to the software and the software’s response to each type.

### **Data Sources**

The two sources of data are the internet connection of the file system. The internet connection provides users access to any desired websites. Additionally, via the internet connection, remote maintenance and control of the advertisements and product and purchase data can be achieved. The files system is used to store the advertisement and product data.

**Table 8 - Data Sources**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System Name** | **Full Name** | **Function** | **Input Source** | **Environment** |
| Internet Connection | Internet Connection | Provides access to external data and allows for remote access of system | N/A | N/A |
| File System | OS File System | Provides access to local files | SSH | Operating System |

#### 

### **Data Refresh/System Updates**

In the current state of development, all product and advertisement data is stored locally in the OS’s file system. Updates must be made locally. This is set to change with the release of version 2 wherein remote access functionality will be added. The Touchscreen Management System is currently still in active development and is functioning primarily as a proof of concept. Upon purchase of product, quantity data is updated in files on the local file system. This will likely change in the future to use a more stable database system. The system must always be capable of running without an internet connection however, thus local copies must be regularly updated and the system must be able to default to them if no connection is found.

At the current state of development, purchases are simulated, and no user currency, credit, or debit is accepted, nor is any related data accepted, stored, or processed in any way. Purchases will be handled through third party software and full purchase functionality is set to be added in release version 3.

Users have access to the internet via the web browser, and can therefore share and with and retrieve data from external sources of various forms. Some browser functionality is removed or limited to prevent users from accessing critical systems or creating security risks.

**Table 9 - Data Refresh**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System (Platform)** | **Server/Address** | **Schedule** | **Specifications** | **Update Mechanism** |
| products | N/A | N/A | 3.5.1 | Local file change |
| product-data | N/A | N/A | 3.5.2 | Local file change |
| user-data | N/A | N/A | 3.5.3 | Local file change |
| adverts | N/A | N/A | 3.5.4 | Local file change |

### 

### **Data Specifications**

### **Products**

Data for each product is stored in a folder whose name is the product’s UUID. Inside said folder are three files, two images, a large and small version of the product image, and a “data.json” file with all the product specific data.

* UUID (directory)
  + large-Image.png (file)
  + small-Image.png (file)
  + data.json (file)
    - Example data.json contents:

**{**

**"id": "0o89Dul55i",**

**"name": "Food Product",**

**"smallImg": "small-Image.png",**

**"largeImg": "large-Image.png",**

**"price": {**

**"amount": 150,**

**"currency": "USD"**

**},**

**"description": "It's good.",**

**"nutrition": "lorem ipsum..."**

**}**

Each data.json file contains the id or UUID of the product which should match the name of the product’s root directory, the product’s name, the name of the small and large image files in the product’s root directory, the price data (both amount and currency type), product description, and nutrition facts. All new products created should follow this layout.

### **Product Data**

“product-data” differs from “products” in that it contains dynamic data. The primary purpose of product-data is to hold the current quantity available in the system at any one given moment. Data for each product is stored in “product-data” in a folder whose name is the product’s UUID. Inside said folder is a single json file whose name is the products’s UUID. Each json file contains two fields, the products UUID and the quantity of said product in the system.

* 0o89Dul55i (example directory)
  + 0o89Dul55i.json (example file)

**{**

**"id": "0o89Dul55i",**

**"quantity”: 12**

**}**

There should be a one to one relationship between the directories of “products” and “product-data”.

### **User Data**

Data for each user (person with some level of administrative access) is stored in a json file whose name is that user’s ID. Each json file contains the user’s ID, an admin flag to specify whether the person has full administrative access, a and passwordB, passwordC, pasInfoPasswordSum, and pastInforPasswordCheck fields for use by the authentication system of the vending application.

* 123456.json (example file)

**{**

**"userID": "123456",**

**"admin": "false",**

**"passwordB": "63E446AF73BD43A1DD9D",**

**"passwordC": "6F4295F94790580ECF6F",**

**"pastInfoPasswordSum": "820F5FE99C9A4D0E4C74",**

**"pastInfoPasswordCheck": "2C27DD1461F5216D2320"**

**}**

### **Adverts**

All adverts are simply image files of types png, jpg, or svg in the “adverts” directory. The vending application handles the timing and rendering of the ads. All ad images present in the “adverts” directory will be loaded in to memory and displayed in a loop. In future releases this system will be updated so that ad-specific timing can be specified. At that time, this portion of the document will likewise be updated to reflect those changes.

## **Hardware**

* + 1. **Screen**

The Touchscreen Interface System was designed for use with the E&T-Tech L-series Infrared Touchscreen. The screen must be connected to the machine running the Touchscreen Interface System via both a USB and HDMI connection. The USB connection is required for receiving touch input. The HDMI is required to carry the display signal.

* + 1. **Computer**

The Touchscreen Interface System was designed for use on desktop computers which meet the requirements for Ubuntu 16.04 LTS, and which have HDMI support.

## **Operating System**

* + 1. **Overview**

The Touchscreen Interface System was built and tested exclusively on Linux (Xenial Xerus) Ubuntu 16.04 LTS. UI settings and applications specific to this distribution have been used to customize the functionality of the system as a whole. The system has not been tested on any other distribution or OS. The vending application was designed to be cross platform in the chance that the project changes and support for additional or alternative operating systems is required.

* + 1. **Installation**

A step by step walkthrough for installation can be found in Table 3 - Section 2 of this document. Additional material can be found in the *“docs/OperatingSystem”* directory in the project repository

## **Web Browser**

The touchscreen interface system comes equipped with a modified form of the Firefox Web Browser. This section of the document covers all necessary specification and directions for maintaining said application.

* + 1. **Configuration**

Firefox was configured using the *about:config* settings and the *Max Tabs* add-on. These allow for restrictions to be placed on Firefox that prevent the user from accessing things they shouldn’t be allowed to and limit the number of tabs that can be opened. With these restrictions, the user is prevented from doing things that might prove damaging to the system.

* + 1. **Installation**

Upon initial install, Firefox should be installed on the system by default. If for any reason that is not the case, it can be done manually by running the following commands in the terminal. After installation, the machine should be restarted.

*$ sudo apt-get update*

*$ sudo apt-get install firefox*

* + 1. **Update**

For regular maintenance and for continues security, updates to the web browser should be conducted. In order to update, the commands are the same as installation. After updates, the machine should be restarted.

*$ sudo apt-get update*

*$ sudo apt-get install firefox*

## **Vending Application**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

* + 1. **Test**

The vending application was designed with testing in mind. As such, it has easy to use unit testing and end-to-end testing. This section explains how to use both testing environments.

* + - 1. **Unit Testing**

Unit testing is available through the Karma and Jasmine frameworks. By running the command below while in “*spikes/Vending App/vening-app”* all unit tests will be run with the application in web mode and the status of said tests will be displayed.

*$ npm run test*

* + - 1. **End to End Testing**

End-to-end testing is available through the protractor framework. By running the command below while in “*spikes/Vending App/vening-app”* the end-to-end testing will be run with the application in web mode and the status of said tests will be displayed.

*$ npm run e2e*

* + 1. **Build**

The developmental instance of the vending software contains a large number of files that are not necessary for the functioning of a deployed application. To deploy a cleaned and minimized version of the application, it must first be built for deployment. To build for deployment run the following command in the terminal from within “*spikes/Vending App/vening-app”*

*$ make build*

* + 1. **Deploy**

The vending application software was designed to be deployed for Linux, Mac OS, and Windows systems. After building, one can deploy a compiled executable using a command for the corresponding platform. To deploy fully for linux in one simple step, run the following command while in “*builds*”. Doing so will create an executable and all necessary auxiliary files in the *“deploy”* directory.

*$ make deploy*

* + - 1. **Local**

Run the following command to deploy an executable for the current platform

*$ npm run electron:local*

* + - 1. **Linux**

Run the following command to deploy an executable and deb for linux operating systems.

*$ npm run electron:linux*

* + - 1. **Mac OS**

Run the following command to deploy an executable for Mac OS platform

*$ npm run electron:mac*

* + - 1. **Windows**

Run the following command to deploy an executable for Windows operating systems

*$ npm run electron:windows*