**Senior Design 2018-2019**



**Installation & Deployment Guide**

This guide is a customized template designed specifically for technical stakeholders. It is intended to provide installation instructions to any technical stakeholder with a need to understand, install, or deploy the project after transition.

Not all sections of this template may be applicable for your specific project or organization. You should feel free to add, move, rename, consolidate, or remove sections to best fit your needs and your project. The Installation & Deployment Guide is developed by the student team during Release 5 and 6 of the Senior Design process.

**Project Title – Sponsor Name**

|  |  |
| --- | --- |
| PREPARED BY: | Jared Nightingale, Anthony Tran, Brett Middle, Taher Ahmed, and Yang Xiao， HoaZhou Li |
| DATE: | 3/26/19 |

**Introduction**

**Purpose**

*The purpose of this guide is to define and describe the steps necessary to install and set up the Spreetail 360 capture solution. The installation of this solution requires two main steps, including setting up the hardware as well as the software for the solution. The hardware must be set up and connected before installing the necessary software. Once this software is installed and running, the front end web application must be deployed into Azure, so that it can be accessed via a web browser. Once these steps are complete, users should be able to connect to the web application and begin a capture process.*

**Intended Audience and Reading Suggestion**

*This document is intended for warehouse workers that will install and operate the hardware portion of the solution, Spreetail developers that will continue the development of the software, and Spreetail administrators that will be installing/deploying the software portion of the project into the Spreetail Azure environment.*

**References**

|  |  |  |
| --- | --- | --- |
| Reference # | Document Link | Author |
| Azure Message Service Bus Documentation | https://docs.microsoft.com/en-us/azure/service-bus-messaging/ | Microsoft |
| Gphoto2 | http://www.gphoto.org | Not Specified |
| Spreetail Design Guidelines | design.tools.spreetail.org | Spreetail |
| Deploy an Azure Web App | https://docs.microsoft.com/en-us/azure/devops/pipelines/targets/webapp?view=azure-devops&tabs=yaml | Microsoft |

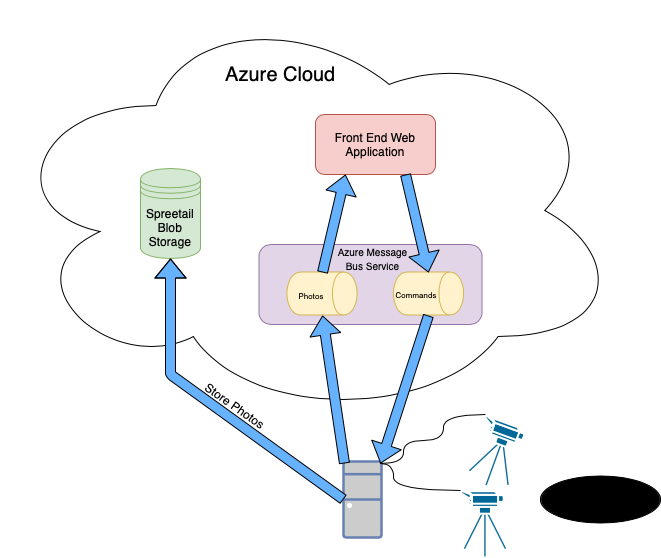
**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| SD Team | This is the Senior Design team working on this Spreetail Project. |
| Message Bus Service | This is a service that allows communication between two separate applications. In this case, it allows communication between the front end application and the hardware tower. |
| Gphoto2 | This is a third party library that allows control of DSLR cameras via a usb connection. |
| Hardware Tower | This is a computer tower being used for the project. This tower is running Linux, and is used to control the DSLR cameras. |
| React Application | This is a web application created with the javascript library React. |
| DSLR Camera | This is a Digital Single-Lens Reflex camera. |
| Electric Turntable | This is an electric rotating turntable that rotates at 97 seconds per rotation, and can hold up to 110 lbs. |

**Assumptions**

|  |  |
| --- | --- |
| Term | Definition |
| Camera Compatibility | Assumption that cameras selected by Spreetail are compatible with the gphoto2 library. |
| Number of Cameras | Software assumes that there will be two cameras used for the project. |
| Turntable Speed | Assumption that the speed of the turntable. |
| Lighting | Assumption that Spreetail Studio will provide lighting for products. |

**Architecture and System Overview**

**

*The above diagram shows the overall architecture of the Spreetail 360-degree capture solution. The front end and hardware tower communicate via an Azure Message Bus Service. This service has two queues to store messages for both directions of communication. The hardware tower also communicates with the Spreetail Azure Blob Storage to store photos taken during the capture process. Finally the hardware tower is connected to the two cameras via a usb connection. The front end application, message bus service, and blob storage are all contained within Spreetail’s Azure cloud.*

**System Requirements**

*Detail at a high-level the base requirements for deploying your solution, such as Microsoft SQL Server 2018, AWS, NLTK/Natural Language Toolkit, etc. Include specific version/build numbers of packages. Also describe any licensing requirements for these packages such as CALs, GPLv2, MIT, etc.*

**Hardware Installation Instructions**

*To set up the hardware rig, follow the directions below.*

1. *Set up the two tripods.*
2. *Open camera case and attach lenses to both DSLR cameras.*
3. *Attach cameras to the tripods.*
4. *Connect both cameras to power via wired batteries.*
5. *Connect Computer tower to power, ethernet, and monitor, mouse, and keyboard.*
6. *Connect cameras to tower via usb cable.*
7. *Connect electric turntable to power.*
8. *Turn on turntable and computer tower.*
9. *When turning on the cameras, make sure that the lower camera is turned on and recognized by the computer first.*
10. *Continue with software installation to set up tower software.*

**Software Installation Instructions**

*To set up the software part of this solution, there are 3 main steps. These steps include setting up the Azure Service Bus Service, installing and running the Tower software, and deploying the React web application into Azure.*

**Azure Service Bus Setup**

*In order to communicate between the hardware tower and the front end, it is necessary to ensure that the service bus is set up correctly to allow this communication. There must be two queues set up in the service bus service, each with a unique name. One must pertain to the one-way communication from the front end to the hardware tower, and the other pertains to the one-way communication from the hardware tower to the front end. The connection variables can be set in the tower software “.env” files to allow developers to change the service/queues that the software uses to communicate. These variables are listed in the table in the section below.*

**Tower Software**

*Installation of this product is supported on the following operating systems and versions:*

* *Ubuntu 18.04.1 LTS*
* *Mac OSX Mojave*

**Required Package Installation**

*The following packages are required to run the software. The installation of these packages is handled by the “startup.sh” script, which is also used to start the hardware server.*

* *python3-pip*
* *python3-dev*
* *gphoto2 & libgphoto2*
* *azure*
* *pillow*
* *azure-servicebus*

**Configuration**

*The variables below can be changed in the .env file found in the Controller directory.*

|  |  |
| --- | --- |
| Config Item | Default Value |
| Bus Namespace | “seniordesign” |
| Bus SAS Name |  |
| Bus SAS Value |  |
| Back End Queue Name | test\_backend |
| Front End Queue Name | test\_images |

**Installation Steps**

* *Download or clone the software from the SeniorDesign18 Spreetail Gitlab Repository.*
* *Right click on the desktop and select “Open Terminal”*
* *Navigate to the “SeniorDesign18” directory*
* *Run the script “startup.sh” with the command: “./startup.sh”*
* *This command installs all of the necessary libraries needed to run the software. It also starts up the software once all libraries are installed.*
* *This command is used to run the server even if all libraries are already installed.*
* *You may be prompted for a password for the sudo command.*
* *Once the libraries are installed, you should see “Starting Server…”. If this message appears, then the software is up and running.*

**Web Application**

*The front end web application is created with the javascript library React.js, Node.js, HTML, and CSS.*

**Roles, Features and Packages**

* *Node Package Manager*
  + - *@spreetail-components/meta*
    - *@spreetail-styles/meta*
    - *azure-sb*
    - *dotenv*
    - *eslint-config-react-app*
    - *eslint*
    - *react-router-dom*
    - *prop-types*
    - *eslint-config-airbnb*
    - *eslint-config-prettier*

**Configuration**

*The variables below can be changed in the .env file found in the root directory.*

|  |  |
| --- | --- |
| Config Item | Value |
| React App SBS Namespace | “seniordesign” |
| React App SBS Shared Access Key Name |  |
| React App SBS Shared Access Key |  |

**Installation Steps**

*To use the front end web application, it must be deployed into Azure. You must have access to Spreetail’s Azure environment in order to perform this deployment. Once the front end is deployed, it can then be accessed through the url given before deployment.*

**Testing the Installation**

1. *On the hardware tower, open a terminal window, and navigate to the Senior Design project. Navigate to “SeniorDesign18/Controller/test”. Run the command “python3 test\_camera\_integration.py”. This is a python file that contains integration tests to ensure that the cameras are connected correctly and have the ability to capture photos. If any issues arise, they must be fixed to ensure proper operation of the hardware.*
2. *Run the startup script in the Senior Design Project root folder with the command “./startup.sh”. You should be prompted for a password for sudo, and the text “Starting Server…” should be displayed. If you see this text, then the software is up and running correctly.*
3. *After deploying the front end, navigate to the url given for the app. If the front page of the web application pops up, then the web app was deployed correctly. If it does not, consult the Azure environment to debug any deployment issues.*
4. *Once on the web application, click the “Preview Product” button. This should cause the cameras to capture 2 images, and then the preview images should be displayed on the screen. If you see these images, then the hardware is working correctly as well as the communication between the front and back ends. If these images are not displayed, an error message should be displayed with the error that occurred.*

**Troubleshooting**

**Common Installation Issues**

1. *Make sure that the lower camera is plugged in and recognized by the tower first. The lower camera in the software is dictated by the first camera identified by the tower.*
2. *Web application does not show up in browser - consult the Azure environment in which the web application was deployed. There may have been an issue during deployment.*
3. *Encountered Autofocus Error - This issue can be caused by the cap on the camera not being removed as well as the camera’s inability to focus with the current view. Check to see if the caps are still on the cameras as well as ensure that the camera is able to focus by adjusting the autofocus of the camera.*

*4. Encountered Missing Camera Error - This error occurs when there are not exactly 2 cameras connected to the hardware tower. Make sure that both cameras are turned on and that both are connected to the tower via usb cable.*