

Design Document

Captain CyBeard: Neil Before Us

Ryan Breitenfeldt | Noah Farris
Trevor Surface | Kyle Thomas

February 25, 2020



Washington State University Tri-Cities
CptS 421 Software Design Project 1

Contents

1	Introduction	1
2	Architecture	1
3	Data Dictionary	1
4	User Interface	1
4.1	User Interaction	1
4.1.1	Platform Selection Screen	1
4.1.2	Authentication Screen	1
4.1.3	File Selection Screen	1
4.2	Administrator Interaction	1
4.2.1	Start Application	2
4.2.2	Stop Application	2
5	Information Repositories	2
5.1	Cloud Platform Repository	2
5.2	Local Storage or Internal Infrastructure	2

List of Figures

Revision History

Revision	Date	Author(s)	Description
0.1	02.25.2020	KT	Document Creation

1 Introduction

This document will go over the design and architecture for the *Downloader* application that was laid out in **Requirements Specification**[1]. The design document will help the client (Cypherpath) and the software development team (Captain CyBeard) understand the architecture and functionality of the application.

Subsequent sections will go over the general architecture of the application in unified modeling language (UML) followed by a data dictionary. Afterwards the user interface and examples of information repositories needed by the application will be presented.

2 Architecture

The application will consist of **NUMBER OF CLASSES** in a model, view, controller architecture. The following UML diagrams will show the classes and their relationships with each other, along with their attributes and methods.

3 Data Dictionary

The Data Dictionary sections will describe what each class does, along with a more detailed view of its associations, attributes and methods that the class has.

4 User Interface

The User interface section will over the actions that a user will be able to take, explaining all menus, buttons, pull down menus, selections, etc. This section will also describe actions that system administrators will need to take to start and stop the application.

4.1 User Interaction

The user will interact with the application through the web interface. They will have several screens (views) to navigate through. Those will include the main screen to select the platform to download from (AWS, Dropbox, Google Drive, etc) and then will be presented with a view to enter their credentials. Once their credentials are entered they will be presented with the final view which contains their directory structure and contents and they will be able to multi-select which of those contents they would like to download.

4.1.1 Platform Selection Screen

4.1.2 Authentication Screen

4.1.3 File Selection Screen

4.2 Administrator Interaction

The system administrator will interact with the application by using the builtin Django commands on the server that the application will run on. They will mainly only be starting and stopping the web application through this interface. Django does have a builtin web administration page but this application will not need to take advantage of this functionality.

4.2.1 Start Application

4.2.2 Stop Application

5 Information Repositories

The application uses several different information repositories. There will be one for each platform supported as outlined in the **Requirements Specification** [1]. For example, AWS storage will be an information repository as well as Dropbox. The user's local hard drive will also be an information repository during development and then once the client takes delivery their internal storage will become the information repository.

5.1 Cloud Platform Repository

The Cloud Repositories are the storage on the cloud for each platform that the user has files stored on. These files can be any type but will typically be an operating system ISO for the client's use case. These repositories can contain any arbitrary number of files and folders stored within this repository, the users will be able to navigate these through the application and select which ones to download.

5.2 Local Storage or Internal Infrastructure

The second type of information repository that the application will contain is the location that the application will download the user selected files from. This will be the user's local storage during the development cycle and in production the client's internal infrastructure will be the repository for these files.

References

- [1] Author *title*. [*description*] year.