Software Design Documents

For NFS Infinity App

By David Rosenberg
U00063482
Contact e-mail: dcrsnbrg@memphis.edu

Submitted
November 15, 2017

Introduction

This document is intended to provide the reader with some visual models describing NFS Infinity, a Django-based network file server app being developed as a student software engineering project exercising agile software development methods. A wire frame drawing of the user interface is included on this page, then on the following pages, the "Big 5" UML diagrams (activity, sequence, use case, state, and class) are examined as well.

Wire Frame

The following wire frame drawing is a glimpse of the client's view of the web-based graphical interface for NFS Infinity. It includes a heading which identifies the app along with login/logout/admin functions. Under that is a line identifying the remote path and user selected filters for searching within that path. Then the remote file system's directory listing is shown, including file names, file sizes, and each file's date modified. Finally are buttons representing the app's main functions: get, put, and delete, along with a button for changing the remote path.

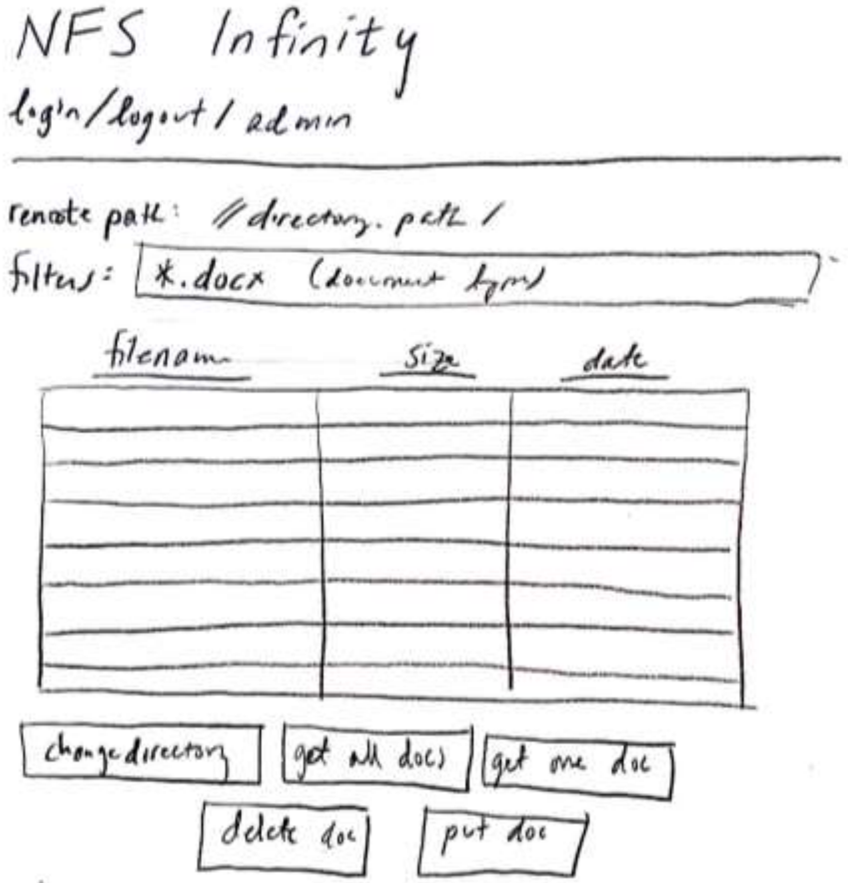

Figure 1: Wire frame for NFS Infinity client interface

Activity Diagram

A common activity carried out between a client and file server is getting a document, also known as a file download. The following activity diagram describes the interaction among the user, the Django application/database server, and remote file system when a connection is established and a download is requested by the user.

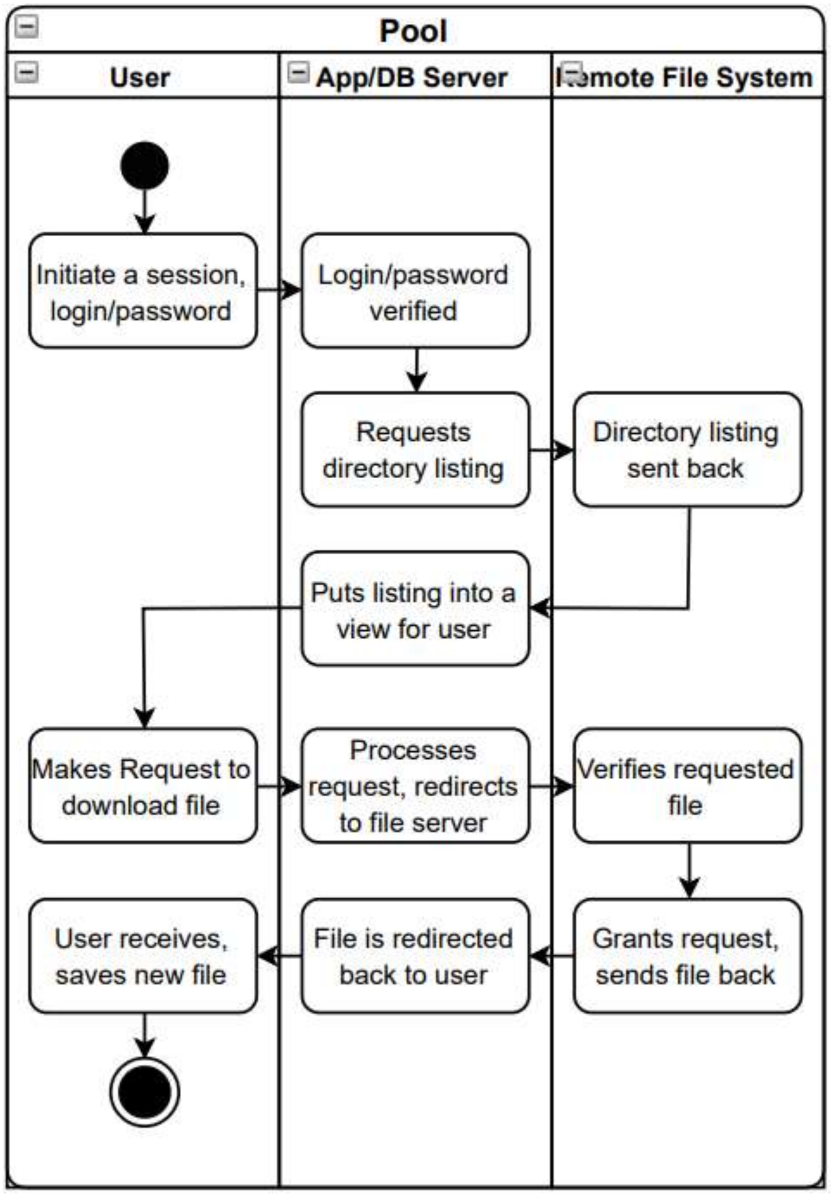
Pool
User
App/DB Server
Remote File System
Initiate a session, login/password
Login/password verified
Requests directory listing
Directory listing sent back
Puts listing into a view for user
Makes Request to download file
Processes request, redirects to file server
Verifies requested file
Grants request, sends file back
File is redirected back to user
User receives, saves new file

Figure 2: Activity diagram describing download request

Sequence Diagram

Similarly, the sequence diagram shown below illustrates the sequence of events when a user logs in to the system, downloads, uploads, or deletes a file.

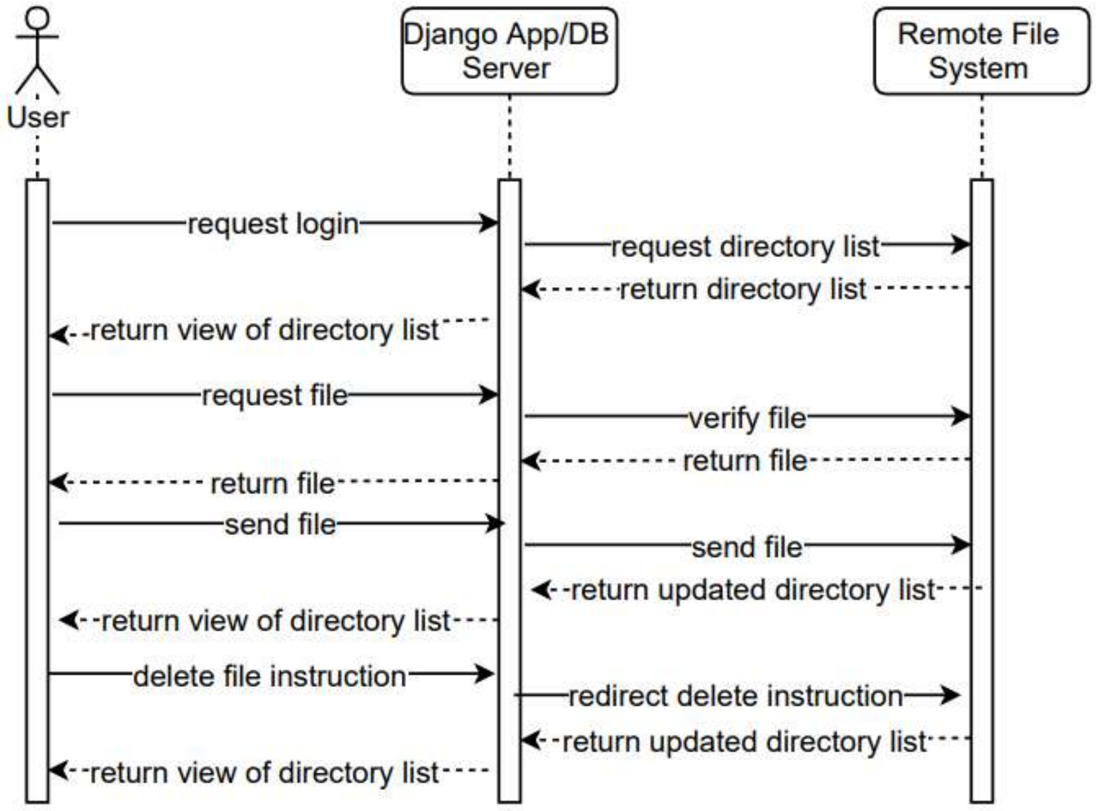

Figure 3: Activity diagram describing download request

Use Case Diagram

Each use case below represents a discrete task that involves external interactions with a system. This simple diagram illustrates the user's requests in manipulating the remote file system.

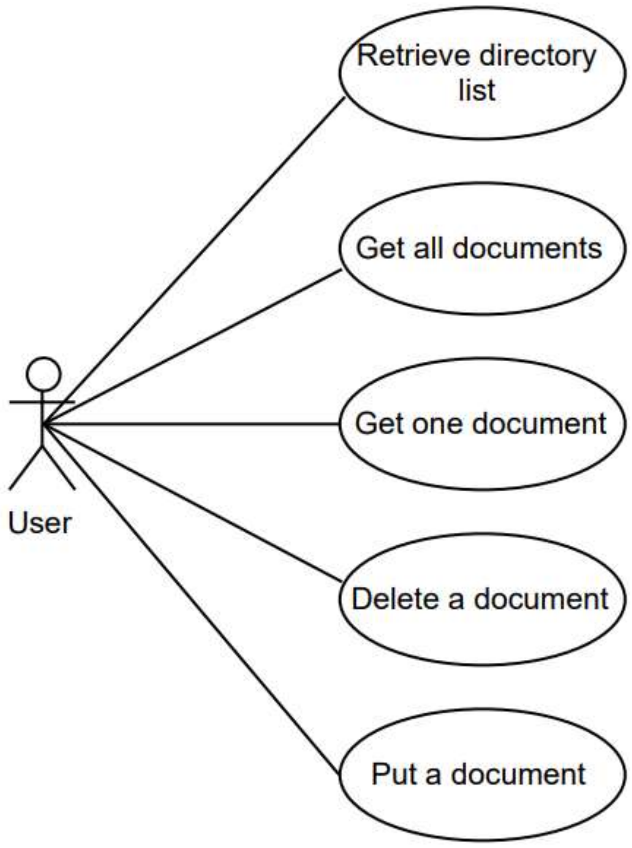

Figure 4: NFS Use Cases (from user's perspective)

State Diagram

After a user has logged in, the system has only two states. The system is prepared for an instruction from the user in its authenticated state. Once an instruction has been made, the system is then in its other state, waiting on a response from the server. That could mean the system is waiting on a file to be accepted or sent, or waiting for an updated directory listing. When it's in that state, the user must wait until it's back in its authenticated state to give a new instruction.

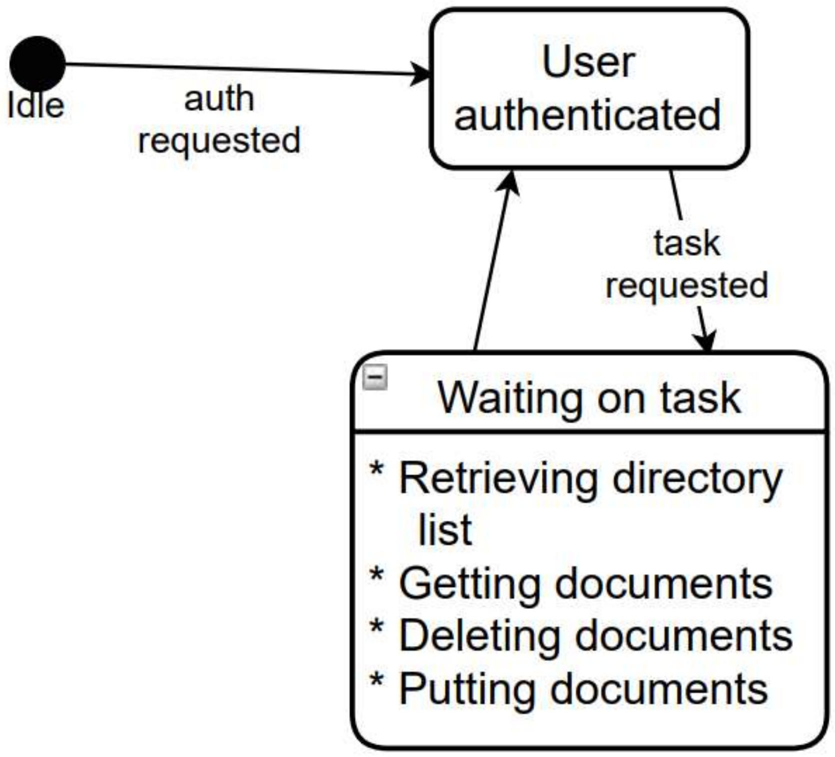

Figure 5: NFS State Diagram

Class Diagram

After gathering some basic requirements and developing some user stories, objects in the form of classes can be defined. Three essential object classes are shown below, detailing some attributes and functions of each class as well as relationships between the objects.

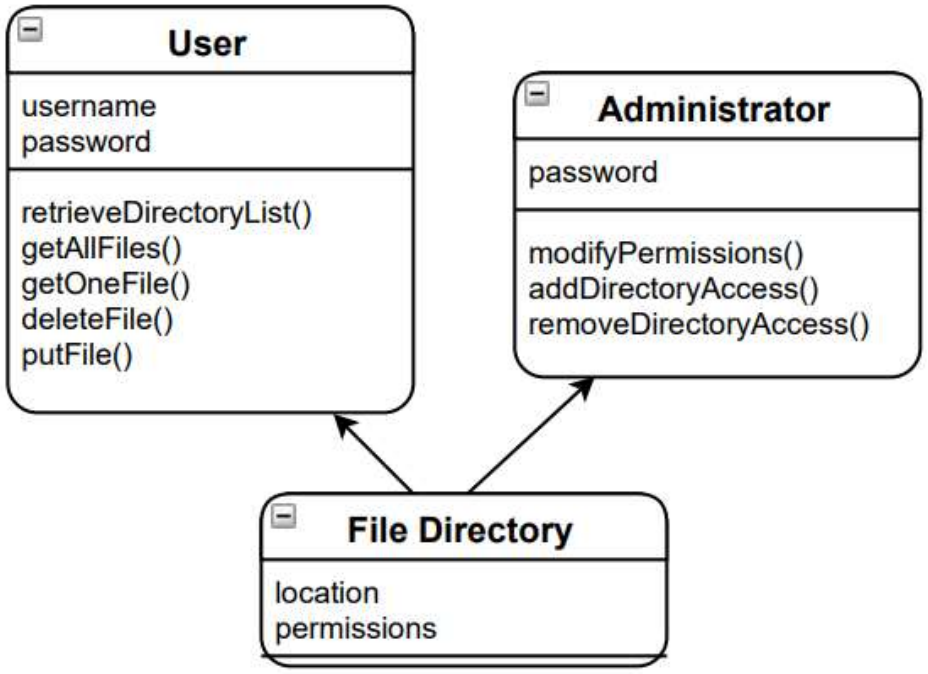

Figure 6: NFS Infinity class diagram