



FEBRUARY 22, 2020

DOCKSUSHARE
REQUIREMENTS SPECIFICATION

RYAN DOCKSTADER



Section 1: Introduction

1.1 Purpose

Currently in states that are mandated to share discovery electronically are commonly sharing it with compact disks, USB flash drives, or giving others access to their internal networks. All these methods have some extreme security risks and lack convenience. This program has the vision of making the process of discovery painless and secure. With a quick and easy way to share discovery, while being able to track, and prove that the other side of the trial received said discovery criminals will no longer get away by a simple formality, and innocent people will no longer be charged with crimes they did not commit by the same simple formality.

1.2 Product Scope

The application will consist of a backend REST based API, and a front end PWA/SPA.

The backend will serve CRUD endpoints for Cases, Case Category Types, Log Records, Media Folders, Media Items, and Users. The backend will allow for uploads/downloads of media files, media folders (zipped), and cases (zipped).

The front end will provide a simple GUI to make requests to the available CRUD endpoints, as well as uploading, and downloading files/folders/cases. It will also allow you to view certain file types (namely photos, videos, and pdf documents) directly in the browser for ease of use.

1.3 Overview

The application will be a progressive web application based on the Angular framework, using several angular modules including RxJs, NgRx, and Angular Materials. The front end should use an easy to use GUI to load cases, upload files, share those cases/files, and download individual files, folders, and entire cases. The system will also notify “authorized users” (users outside the agency that have access to certain cases/folders/files) of any changes or additions to the files they have access to. Simply put it’s a browser based document management system that is made to allow for easy, time based sharing with external users.

I will be following an agile software development lifecycle. Since a lot of the information that is needed is dependent on something else first being there, this will work very well for me. It will also allow me to consistently release to the stakeholders of the product.

1.4 Technologies Used

REST API

- ASP.NET Core 3.1
- ASP.NET MVC
- ASP.NET Authorization
- EntityFrameworkCore
- AutoMapper
- Linq
- HostedServices
- SignalR
- AWS SDK

- JSON files
- Let's Encrypt CertBot
- JWT

Angular 8 SPA/PWA

- Angular 8
- Angular Animations
- Angular Flex Layout
- Angular Materials
- Aspnet Singlar R
- Hammer JS
- JWT Decoder
- RxJs
- NgRx

1.5 User Profiles

The target audience for this application is government-based entities involved in litigation procedures. This means the primary users for will have a wide range but will generally be between 25-65 and have at minimum some college experience. They will be mildly familiar with technology but may not use more then a smart phone in their every day lives. They will be a mix of males and females. They will use a range of devices from smart phones to full desktops to view the cases and files.

1.6 Application Workflow

The primary function of this application is to store and share files with others. The application will allow the user to add a case, and then upload as many media items as needed to that case. The application will then allow the user to share that case with others that they have added to their system, in the non-agency user group.

Section 2a: Requirements

Note: rows w/ strike through have been completed

- ~~1. The application shall not give 504 error messages due to large files~~
- ~~2. The application shall inform the user the status of any active uploads/downloads/zips~~
- ~~3. The application shall display log files to the user which are searchable by case and/or user~~
- ~~4. The application shall log when an email is sent to a user~~
- ~~5. The application shall log when a user downloads a media item~~
- ~~6. The application shall track the file size of individual files~~
- ~~7. The application shall allow users to retry any failed uploads/downloads~~
- ~~8. The application shall track used drive space~~
- ~~9. The application shall allow admin users to view how much storage is being used~~
10. The application shall prevent itself using more than 80% of the available space on the drive
11. The application shall allow users to create folders and sub folders in cases
12. The application shall track usage on the primary web page via Google analytics
13. The application shall delete any stored files when the database row is deleted

Section 2b: Stretch Requirements

1. The application should keep the computer awake when uploading/downloading files
2. The application should maintain an active session while users are actively using it
3. The application should stream videos.
4. The application should allow a user to scrub through a video
5. The application should have a test for each user type, for each endpoint
6. The application should show authorized users what is new on cases since their last case view
7. The application should allow a user the option to download all new case items
8. The application should create Compressed copies of files for streaming
9. The application should use the compressed copies for the stream only
10. The application should have an installer that will setup a new instance
11. The application should have an upgrade tool to upgrade existing environment
12. The application should allow an admin to resend an invite email
13. The application should have arrows for the user to click through media files
14. The application should only have a single upload module that allows the user to modify all data points for each item.
15. The application should tell the admin when a user has setup their account
16. The application should hold deleted files/cases for 30 days before it deletes backend files
17. The application should allow admin users to be able to filter and sort users by names, emails and user types, isActive and isRegistered
18. The application should check the active checkbox by default when adding new users
19. The application should allow an anonymous user to Contact Us from the primary web page
20. The application should have an FAQ Section on the main web page
21. The application should work with gestures
22. The application should work on screens with smaller devices

Section 2c: Weekly Schedule

Week 7 – 02/17/2020 (10 hours)

- The application shall prevent itself using more than 80% of the available space on the drive (10 hours)

Week 8 – 02/24/2020 (12 hours)

- The application shall allow users to create folders and sub folders in cases (12 hours)

Week 9 – 03/02/2020 (10 hours)

- The Application shall delete any stored files when the database row is deleted (8 hours)
- The application shall track usage on the primary web page via Google Analytics (2 hours)

Week 10 – 03/09/2020 (12 hours)

- The application should keep the computer awake when uploading/downloading files pt. 1

Week 11 – 03/16/2020 (12 hours)

- The application should keep the computer awake when uploading/downloading files pt 2

Week 12 – 03/23/2020 (10 hours)

- The application should maintain an active session while users are actively using it

Week 13 – 03/30/2020 (10 hours)

- The application should stream videos (8 hours)
- The application should allow a user to scrub through a video (2 hours)

Week 14 – 04/06/2020 (15 hours)

- The application should have a test for each user type, for each endpoint

Backlog (40+ hours)

1. The application should show authorized users what is new on cases since their last case view
2. The application should allow a user the option to download all new case items
3. The application should create Compressed copies of files for streaming
4. The application should use the compressed copies for the stream only
5. The application should have an installer that will setup a new instance
6. The application should have an upgrade tool to upgrade existing environment
7. The application should allow an admin to resend an invite email
8. The application should have arrows for the user to click through media files
9. The application should only have a single upload module that allows the user to modify all data points for each item.
10. The application should tell the admin when a user has setup their account
11. The application should hold deleted files/cases for 30 days before it deletes backend files
12. The application should allow admin users to be able to filter and sort users by names, emails and user types, isActive and isRegistered
13. The application should check the active checkbox by default when adding new users
14. The application should allow an anonymous user to Contact Us from the primary web page
15. The application should have an FAQ Section on the main web page
16. The application should work with gestures
17. The application should work on screens with smaller devices

Section 3: Design Overview of the Product

Workflow

The application will start off with an email sent to a users inbox. The use will click the link in the email, and register their account. After the are registered their workflow will divert depending on their user types, but will be cascading upwards (I.E. anything a non-agency user can do, a general user can do. Anything a general user can do a super user can do. Anything a super user can do an admin user can do.)

Non-Agency Users

Updating a Password

Any user can select a link on the front page of forgot password, and enter their email. Once they have entered their email they will get an email with a unique link, that they will click on. Using that link along with their email address they will be able to reset their password

Updating Account Details

Any user can update details for their own account by clicking on their name dropdown and selecting My Account

Log In

Any user can login via the login page. They will enter their email and registered password to be able to access the application

Seeing the Case List

All users will be able to see a case list, if they are authorized. If they are not authorized on any cases they will not see any cases on the case list. Otherwise the case will display the cases they have, 10 at a time, and allow them to open any cases on the case list

Viewing a Case

Any user with access to a specific case will be able to see the media items related to that case. They can select them, view their name/description, depending on the media type see it in the browser and download it.

Downloading Entire Cases

Any user with access t a specific case will be able to download that case a zip file.

General Users

Adding a Case

Any general user can add a case. To do this they will click the + button from the case list. From the next screen they will enter the court number, first and last name of the defendant, an inactive date, and category. They also have the option to add authorized users from this screen, and upload any media necessary for that case

Updating a Case Created by the user

Any general user can update a case that they have created. They can modify the court number, first and last name of the defendant, inactive date, authorized users, and upload any additional media to the case. This is accomplished by selecting the drop down on case actions and choosing update case, or authorized media and modifying the data.

Deleting a case created by the user

Any user can delete a case they created. This is accomplished by opening the case from the case list, selecting case actions and selecting delete case.

Deleting a Media Item Created by the user

Any general user can delete a media item they created. This is accomplished by selecting the media item in the case view, selecting the actions drop down and selecting Delete Media.

Updating a Media Item Uploaded by the user

Any general user can update the metadata of a media item they have uploaded. They can modify the name, description and type.

Super User

Updating a Case not created by the user

A super user can update **any** case in the system (not just the ones they created) in the same manner as a general user.

Updating a Media Item not created by the user

A super user can update **any** media item in the system (not just the ones they created) in the same manner as a general user.

Admin User

Deleting a Case not created by the user

An admin user can delete **any** case in the system (not just the ones they created) in the same manner as a general user.

Deleting a Media Item not created by the user

An admin user can delete **any** media item in the system (not just the ones they created) in the same manner as a general user.

Reviewing Logs

An admin user can review case logs by going to the case, selecting the case actions drop down and selecting the view logs options.

An admin can also do this by following the same workflow for a media item

An admin can also view all the logs by going to the admin dropdown, and selecting logs

Case Categories

An admin user can add case categories by going to the admin menu and selecting case categories. From this screen they delete existing categories as well as add new ones

Users

An admin user can add, update or delete a user by going to the admin menu and selecting Users. They can then add a new user, update an existing one, or delete a user

Resources

<i>Item</i>	<i>Quantity</i>	<i>Billing Period</i>	<i>Total Price per Billing Period</i>	<i>Total Yearly Cost</i>
<i>AWS S3 Storage</i>	2 TB	Monthly	\$79.89	\$958.68
<i>AWS EC2 t3.large Server</i>	1	Yearly	\$567.00	\$567.00
<i>AWS EC2 t3.micro Server</i>	2	Yearly	\$71.00	\$142.00
<i>AWS Route 53 Domain</i>	1	Yearly	\$12.00	\$12.00
<i>AWS Hosted Zone</i>	1	Monthly	\$0.50	\$6.00
<i>DNS Queries</i>	?	.50/million/month	\$1.00	\$12.00
<i>AWS EBS</i>	200 GB	Monthly	\$10.00	\$240.00
			<i>Total:</i>	\$1937.68

AWS s3 Storage

This storage will be used for long term file storage. This is an encrypted share, that exists in a partitioned data center specifically for sensitive files. They will be sent to the s3 storage via the AWS s3 SDK for .net Core

AWS EC2 t3.large Server

This server will host the database for several clients at a time. The server is meant to be larger than the web servers, so it can contain multiple database and help drive costs down while keeping performance high. The database will be a Microsoft SQL 2019 Express instance, as we will not need the larger enterprise grade features. The database will be accessed using Entity Framework, using a code first design until it is deemed necessary to create a more robust database structure.

AWS EC2 t3.micro Server

This server will host the API, and the front end application. The API will be built using .net Core 3.1, and will be able to run on a Linux host. The front-end will be packaged and deployed in a location that the .net core application can server it to the client.

AWS Route 53 Domain

This will be used just like any other domain is used. We will have a single root domain with our home page, and all clients will be on a subdomain.

AWS Hosted Zone

The hosted zone will host all our DNS entries, to point our domain to the API servers

DNS Queries

These queries are the queries the client machines will be running to properly locate our application

AWS EBS

This storage is the temporary storage for the API server, for download/zipping/uploading files to/from the client.

Verification

1. The application shall not give 504 error messages due to large files
 - Verify A large file can be uploaded and then downloaded from the system, taking over 5 minutes to complete each direction.
2. The application shall inform the user the status of any active uploads/downloads/zips
 - Verify A large file can be uploaded and then download from the system. The file should be able to be monitored while going both directions via the application
3. The application shall display log files to the user which are searchable by case and/or user
 - Verify the application log files can be searched via a case and a user
4. The application shall log when an email is sent to a user
 - Verify that emails show in the logs
5. The application shall log when a user downloads a media item
 - Verify that downloads show in the logs
6. The application shall track the file size of individual files
 - Verify that the size of the file is stored in the database, and is accurate
7. The application shall allow users to retry any failed uploads/downloads
 - Verify you can retry, by terminating your connection to the internet during an upload. Then reconnect, and restart the upload
8. The application shall track used drive space
 - Verify the application in logging on the server it's drive space usage during file interactions
9. The application shall allow admin users to view how much storage is being used
 - Verify that when logged in as an admin you can see how much storage, and what percent of allowed storage you are using
10. The application shall prevent itself using more than 80% of the available space on the drive
 - Verify that when zipping a large file, you do not fill up the available space on the drive
11. The application shall allow users to create folders and sub folders in cases
 - Verify a user can create folders in cases, and sub folders in that folder
12. The application shall track usage on the primary web page via Google analytics
 - Verify that google analytics have been added to the primary web page
13. The application shall delete any stored files when the database row is deleted
 - Verify that a media item does not exist in s3 storage after it have been deleted, by deleting an item and checking the AWS bucket for that storage identifier.
14. The application should keep the computer awake when uploading/downloading files
 - Put a small sleep timer on your computer, and upload a large file. Verify the computer does not go to sleep
15. The application should maintain an active session while users are actively using it
 - Use the application for an hour, and verify you're session is never terminated
16. The application should stream videos.
 - Verify that the video streams to the client, and not downloads
17. The application should allow a user to scrub through a video

- Open a video file and verify that you can pick any point in the video to watch from
- 18. The application should have a test for each user type, for each endpoint
 - Verify that a test exists for each API endpoint
- 19. The application should show authorized users what is new on cases since their last case view
 - Login as a non-agency user, and view a case. Login as a general user and add more media items to that case. Log back in as the non-agency user and view the case again. It should be obvious what items are new
- 20. The application should allow a user the option to download all new case items
 - Login as a non-agency user, and view a case. Login as a general user and add more media items to that case. Log back in as the non-agency user and view the case again. It should be obvious what items are new. Verify that you can download all of them in a single click.
- 21. The application should create Compressed copies of files for streaming
 - Verify that two copies are stored, and one is smaller than the other for video files.
- 22. The application should use the compressed copies for the stream only
 - Verify you get the full file when downloading it.
- 23. The application should have an installer that will setup a new instance
 - Verify that you can install an application without manual intervention via AWS interfaces
- 24. The application should have an upgrade tool to upgrade existing environment
 - Verify that you can upgrade an application without manual intervention via AWS interfaces
- 25. The application should allow an admin to resend an invite email
 - Verify that you can resend an email as an admin user
- 26. The application should have arrows for the user to click through media files
 - Verify that you can arrow through media items on a case
- 27. The application should only have a single upload module that allows the user to modify all data points for each item.
 - Verify that there is not multiple upload modules
- 28. The application should tell the admin when a user has setup their account
 - Verify that you can tell who has and who hasn't setup their account.
- 29. The application should hold deleted files/cases for 30 days before it deletes backend files
 - Verify that a deleted file still exists in the database and s3 bucket after it has been deleted
- 30. The application should allow admin users to be able to filter and sort users by names, emails and user types, isActive and isRegistered
 - Verify that sorting users is possible
- 31. The application should check the active checkbox by default when adding new users
 - Verify the active checkbox is checked when adding a new user
- 32. The application should allow an anonymous user to Contact Us from the primary web page
 - Verify that you can use a contact us form on the primary page
- 33. The application should have an FAQ Section on the main web page

- Verify that there is an FAQ on the primary page
- 34. The application should work with gestures
 - Using a touch screen, verify you can swipe through media items
- 35. The application should work on screens with smaller devices
 - Using a cell phone verify the application performs well