Lab 2 Section 3 - Art Guardian Product Specification

Team Blue

Old Dominion University

CS 411W

Professor James Brunelle

October 15, 2022

Collaborative Section 3

Version 1

3 Product Requirements

3.1 Functional Requirements

3.1.1 User Interface

3.1.1.1 Account Creation *(O: Diaz, M1: Ihde, M2: Thompson)*

- Users that do not have an account shall create an account through the sign-up page.
- This function shall create and store accounts with a username, password, confirmed email address, and linked art account.
- An account shall be required for users to utilize features of Art Guardian.
- The linked art account needs to originate from either Twitter, Instagram, Tumblr, or DeviantArt.

3.1.1.2 Art Upload *(O: Ihde)*

- This function shall allow users to upload images of their artwork to be stored on the database, which will then be available to view in the art library.
- Users must upload images that are in PNG or JPG formats.

3.1.1.3 Image Library (*O: Ihde, M1: Zheng*)

- This feature shall display all of the images uploaded by the user. Images shall be displayed as cards in a rectangular grid format that adjusts its number of images per row based on the window size. Each card will display a preview of the uploaded art and the following information will be displayed under the image preview:
    - Title
    - Upload date

- Additionally, there shall be an edit button which when clicked, a pop up window will appear to the right of the art, and users can edit the title, set whitelisting status, or delete the art piece, which will also remove it from the art database.

3.1.2 Algorithms

3.1.2.1 **Whitelisting** (*O: Roberts, M1: Thompson*)

- This feature allows users to exclude specific NFTs from results based on Token ID.
- If the Token ID is found, the NFT is added to the whitelist.
- If the Token is not found, the application will state that the Token ID is not valid and to insert another ID.

3.1.2.2 **Marketplace Monitoring** (*O: Roberts, M1: Diaz*)

- This feature shall run on the backend to monitor NFT marketplaces for stolen artwork on a minted NFT.
- This feature will utilize the NFTport API to monitor popular marketplaces such as Opensea, rarible, and binance.

3.1.2.3 **Image Matching** (*O: Zheng, M1: Ihde, M1: Thompson*)

- This feature shall run as a AWS Lambda function which connects to the database.
- The feature relies on the NFTPort API for returning images that will be compared to. Each image uploaded by the user shall be searched using the NFTport API, which will return multiple NFTs that best match the uploaded image. Only the top five NFTs will be passed on to the BRISK feature-matching algorithm.
- The BRISK feature-matching algorithm will compare the image uploaded by the user to the top five NFTs returned by NFTPort API. The BRISK feature-matching algorithm will return a similarity score for each comparison, and the NFT with the highest similarity score, along with the user uploaded image, will be passed on to the Functional Requirement 3.1.2.4.

3.1.2.4 **Stolen Art Alert** (*O: Zheng, M1: Thompson*)

- Provide the capability to receive push notifications on desktop environments able to run web applications. The information within the push notifications will include:

    - The image uploaded by the user and the NFT with the highest similarity score, returned by the process described in Functional Requirement 3.1.2.3.
    - The online location, listed price, and uploader identity of the NFT.
- Users can respond to the push notification by
    - Confirming and initiating the DMCA Generation and DMCA filing processes mentioned in requirements 3.1.3.1 and 3.1.3.2.
    - Whitelisting the image, described in requirement 3.1.2.1.
    - Ignore the push notification.

3.1.3 DMCA Takedown Processes

3.1.3.1 DMCA Generation (*O: Roberts, M1: Diaz*)

- This is a page that will automatically generate a DMCA take down for a user after the user confirms the stolen artwork is theirs and that they would like to pursue a take down. The take down will be generated using the preset DMCA Takedown Template as well as data collected from the NFT that was found to be stolen art and user data. This includes:
    - User artwork title
    - Marketplace name
    - Marketplace URL
    - email address of notice sender
    - A statement that the notifier has a good faith belief that the material is not authorized by the intellectual property or copyright owner, its agent, or the law.
    - Statement that Art Guardian has permission to send this notice on behalf of the copyright holder

3.1.3.2 DMCA Filing (*O: Roberts, M1: Diaz*)

- This feature shall send the DMCA takedown request through email. This feature shall be partially implemented in the prototype, in which the DMCA takedown request will be sent to a testing email via the gmail API.

3.1.3.3 DMCA Cataloging (*O: Roberts, M1: Diaz*)

- This function tracks and catalogs information regarding generated DMCA takedown notices.
    - This includes:
    - Date and time of generation and issue
    - Parties involved
    - DMCA sent
- This catalog can later be accessed by the user via the user profile under “DMCA History”

3.2 Performance Requirements (*O: Diaz, M1: Thompson*)

Initial startup should not take more than 30 seconds and each navigation through pages/ sections of the application should not take more than 5 seconds. DMCA generation should not take longer than 30 seconds. Art Guardian should be able to store and track 100 pieces of artwork without skipping scans.

3.2.1 Scanning Frequency (*O: Zheng, M1: Ihde*)

- Art Guardian will scan NFT marketplaces for counterfeit NFTs using artwork uploaded by the user once every two weeks.

3.2.2 Web Application Performance

3.2.2.1 Web Pages (*O: Zheng, M1: Kidd*)

- All web pages will load within five seconds.

3.2.2.2 Image Upload (*O: Zheng, M1: Kidd*)

- Uploading images shall not take more than one minute.

3.2.2.3 Image Loading (*O: Zheng, M1: Kidd*)

- All entries within the Image Library will load within one minute.

3.2.2.4 Image Matcher (*O: Zheng, M1: Thompson*)

- Results will be returned from the Image Matcher within one minute.

3.2.2.5 DMCA Generator (*O: Zheng, M1: Thompson*)

- DMCA takedown notices will be generated within 30 seconds.

3.2.2.6 Network Performance (*O: Zheng, M1: Kidd*)

- Given that the user has a stable internet connection with a speed >700 mbps, all operations involving the transfer of data from a non-UI component to the UI and vice versa will not require more than two minutes in completion time.

3.2.2.7 Concurrency Performance (*O: Ihde, M1: Zheng)*

- The server shall be able to handle at least 5 requests per second.

3.3 Assumptions and Constraints (*O: Kidd, M1: Ihde)*

Condition	Type	Effect on Requirements
Users cannot occupy more than one profile.	Constraint	Bounds the problem of matching users to available profiles
Only valid data entries will be provided.	Assumption	Allows for minimal error checking for the purposes to developing and demonstrating the prototype
The desktop web application will be hosted through AWS services.	Dependency	The AWS Amplify platform must be simulated if AWS cloud services are not available

Table 2. Effects of Assumptions, Dependencies, and Constraints on Requirements.

4 Non-Functional Requirements

4.1.1 Security (*O: Kidd, M1: Roberts, M2: Ihde)*

- The website will use HTTPS to encrypt transferred data between client and server.
- Art Guardian shall implement Database access control using an MFA requirement and AWS IAM role policies.
- Art Guardian shall secure accounts using two factor authentication via google authenticator.

4.1.2 Maintainability (*O: Kidd, M1: Hite)*

- The Art Guardian uses the Amazon Amplify platform to provide a low-maintenance configuration for accomplishing NFT Uploading and Tracking. The Art Guardian Amplify project is updated on a quarterly basis to provide the very latest updates for NFT monitoring and detection. Maintenance procedures for all other components, such as the database, are conducted semiannually and can be performed by personnel appropriately skilled.

4.1.3 Reliability *(O: Roberts, M1: Diaz)*

- Art Guardian art upload will be available 24 hours a day, 7 days a week. A marketplace search will be done once when art is uploaded then once a month from then on. Art searches will happen in batches of at most 100 pieces of art at off peak hours, 12AM- 6 AM EST. The prototype must be able to complete at least 90% of its searches, DMCA generations, and dispatches without error.
- Art upload and DMCA generation must be available at all times. Tutorials and tips and tricks may go down as long as the aforementioned features are still available without affecting critical performance.
- Art Guardian must have a 90% reliability so theft does not go without notice and artists are able to act. Artists must be able to upload their art and have it tracked and DMCA's generated without interruption as that is the core function of Art Guardian. Tutorials and tips and tricks sections are non critical as they support artists in protecting their art but do not add to the core features of Art Guardian.