Requirements Draft

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The non-functional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This section should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and non-functional requirements in more detail. If necessary, further detail may also be added to the non-functional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components, the system, and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.

Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

Sections for later: System Architecture, Systems Models, and System evolution need not be filled in for this draft.

Table of Contents

Table of Contents 1

Preface 2

Introduction 2

Purpose 2

Scope of Project 2

Glossary 2

User Requirements Definition 2

System Architecture 2

System Requirements Specification 3

System Models 3

System Evolution 3

Appendices 3

Index 3

Preface

Introduction

Purpose

The purpose of this document is to provide and present a clear and detailed description of the Art Create project with the Gund Gallery. It will explain the functionalities of each individual part of the system, how they communicate, the constraints under which they should operate, and .

Scope of Project

This software system will be

Glossary

Machine learning: Machine learning is the study of computer algorithms that improve through experience and data without direct input from humans.

Genetic algorithm: A genetic algorithm is an optimization algorithm that mimics the process of evolution, They work through constant improvement with a process of iteration, repeating until it finds the optimal solution. The first basic component of a genetic algorithm is the fitness function, which is the function that the algorithm is trying to optimize. The algorithm then tries to optimize the problem with a population of chromosomes, numerical values that represent a possible solution to the problem that the genetic algorithm is trying to solve. Next, the algorithm selects which chromosomes will reproduce, and implements a crossover to produce the next generation of chromosomes. Similar to evolution, there will be a low chance of “mutation”, where a chromosome may be altered to ensure diversity and also search for hidden optimums. Through multiple iterations, called runs, the genetic algorithm will refine its chromosomes until they are the optimal solution.
Discriminator: It is trained through many thousands of images from the WikiArt database within the desired (user-inputted) category, and is used to determine if the artwork created fits that category. It will only return true or false.

Generator: The generator utilizes a genetic algorithm to create artwork, and checks with the discriminator every generation to see if the artwork meets the desired category.

User Requirements Definition

The service provided allows the user to use a simple interface to use a genetic algorithm in order to create an artwork. The main functional requirement is that the machine should create art that matches the user's inputted search term based on Gund Gallery images. The main non-functional requirement is the time constraint: the generation of the image should not take too much of the user's time and internet bandwidth.

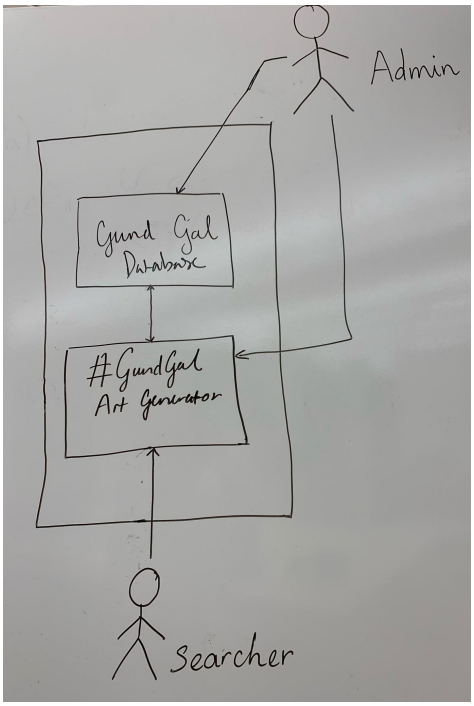

Image 1: Draft use case model

Searcher Use Case:
Use case: Using the website and program

Brief Description: The searcher accesses the website, selects a category for the art they want to create, starts the program, and views/downloads the artwork after it is created.

Step-by-step:

1. The searcher accesses the Art Create Website
2. User searches for a term for which they want their artwork to be based on
3. The system displays 20 or so terms for which the user can choose.
4. The system will then take in the term and feed it to our Discriminator.
5. The user can then click a button to run the genetic algorithm.
6. When the program is finished, the new artwork will be displayed on the screen, along with 4 images from the database in the same genre/category.
7. The user will then have an option to download the image.

Administrator Use Case:
Use case: Adding new artwork and changing topic hashtags attributed to the artworks.

A_Adding new artwork:

Brief Description: A curator or developer from the Gund Gallery or developing teams add new artwork to the Gund Gallery database and attach hashtags to the pieces.

Step-by-step:

1. The administrator logs onto a separate back-end server with login credentials on the Gund Gallery Database server.
2. The homepage shows a separate tab that allows upload of art pieces.
3. The Gund Gallery database server prompts the admin to upload new artwork in a high resolution format (.jpeg, or something similar) and the server attaches an incremental ID to the artwork.
4. The user chooses from a set list of keywords to attach to the new artwork.
5. The user saves their changes.
6. The user logs out.

B_Adding new keyword:

Brief Description: A curator or developer from the Gund Gallery or developing teams add new hashtags that can be attached to the Gund Gallery pieces.

Step-by-step:

1. The administrator logs onto a separate back-end server with login credentials on the Gund Gallery Database server.
2. The homepage shows a separate tab that allows for the addition of keywords (different from the artwork upload tab).
3. The tab allows the admin to put in the ids of the pieces that they want to be attached to this hashtag.
4. The admin puts in the new keyword and the ids of the art pieces.
5. The admin saves the changes and logs out.

C_Changing keywords attached to an existing artwork:

Brief Description: A curator or developer from the Gund Gallery or developing teams changes what keywords are attached to a certain art piece.

Step-by-step:

1. The administrator logs onto a separate back-end server with login credentials on the Gund Gallery Database server.
2. The homepage shows a separate tab that allows for changing the hashtags of an existing artwork (different from the artwork upload and keyword changing tabs).
3. The search bar in the tab allows the admin to search for an artwork in the Gund Gallery database by its unique ID, name, or author.
4. The system displays the most relevant results from the Gund Gallery database.
5. The admin clicks onto the art piece and the server fills out a display form on the current tab with all the information of the art piece (ID, name, author, hashtags attributed).

6. The hashtag line allows the admin to delete or add hashtags that already exist in the art creator or Gund Gallery database.
7. The admin makes and saves the changes.
8. The admin logs out.

System Architecture

System Requirements Specification

We will incorporate a #GundGallery that will have an art similar to the Gun Gallery in order to avoid the random choice of the completely unrelated artwork.

Searcher Use Case:
Brief Description: the system creates a composite image that matches the user's inputted "search term" based on Gund Gallery images.

1. The system goes through each artist under "#GundGallery" on the WikiArt.org alphabetically and checks to see if the artist is tagged under the "search term".
2. The system downloads each qualified image that is available in the public domain using the artist and painting name as the filename.
3. Train Discriminator:
    a. perform semantic search of "search term"
    b. use an image and text encoders
    c. compare the embedding from "search term" to the embeddings from the paintings.
4. Loop the Gund Gallery artwork database through Discriminator to qualify the "search term" criteria:
    a. Compare the embedding from "search term" to the embeddings from the paintings from Gund Gallery artwork database;
    b. Output the binary result: 1 if text embedding matches image embedding and 0 otherwise.
5. Use the Genetic Algorithm on the top four/five images that the best match the "search term"
6. Display the generated composite image to user.

Administrator Use Case:
A_Adding new art piece:
Brief Description: A curator from the Gund Gallery adds new artwork to the Gund Gallery database and attach hashtags to the pieces.

Step-by-step:
1. The entered login credentials link to a separate back-end server on the Gund Gallery Database server.

2. User's account branches to the library of art pieces on the Gund Gallery Database server via "Upload" tab.
3. The system checks the uploaded artwork's resolution format (.jpeg, or something similar):
    a. If the resolutions doesn't check the requirements, release an error message and returns to step 2.
    b. If the resolutions checks the requirements, attaches to approved uploaded artwork an incremental ID.
4. Offers a user a set list of keywords to attach to the new artwork.
5. Hashtags the artwork with user-chosen keywords.
6. Receives "Save" command and saves the new artwork on the Gund Gallery Database server.

B_Adding new keyword:
Brief Description: A curator from the Gund Gallery adds/deletes/updates hashtags attached to the Gund Gallery pieces.

Step-by-step:
1. The entered login credentials link to a separate back-end server on the Gund Gallery Database server.
2. User's account branches to the library of art pieces on the Gund Gallery Database server via "Keywords" tab.
3. The list of keywords is displayed with an "Add" Tab.
4. After receiving an "Add" Tab command, the system prompts admin to enter the new keyword.
5. Receive "Save" command and saves the new artwork on the Gund Gallery Database server.

C_Changing keywords attached to an existing artwork:
Brief Description: A curator or developer from the Gund Gallery or developing teams changes what keywords are attached to a certain art piece.

Step-by-step:
1. The entered login credentials link to a separate back-end server on the Gund Gallery Database server.
2. User's account branches to the library of art pieces on the Gund Gallery Database server via "Art Work" tab.
3. The search line asks the admin to input in the ID of the piece they are interested in
4. The input from the search line is compared to the IDs from the Gund Gallery Database to place a bijective identity.
5. Offer a user a set list of keywords to attach to the selected artwork.
6. Hashtag the artwork with user-chosen keywords.
7. Receives "Save" command and saves the new artwork on the Gund Gallery Database server.

System Models

System Evolution

Appendices

Index