**------------------------------------------------------------------------------------------------------------------------------------------  
Revision History  
------------------------------------------------------------------------------------------------------------------------------------------**

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Description* | *Author* |
| 3/3/2022 | 1.0 | First Draft | Naomi Nash & Daria Pacheco |
| 3/22/2022 | 1.1 | Cleanup of Roles | Naomi Nash |
| 4/18/2022 | 1.2 | Addition of Diagrams | Naomi Nash & Daria Pacheco |
| 4/18/2002 | 1.3 | Added Comments for Revisions | Naomi Nash & Daria Pacheco |
| 4/19/2022 | 1.4 | Revised Based on Comments | Naomi Nash & Daria Pacheco |
| 4/19/2022 | 1.5 | Added diagrams | Naomi Nash & Daria Pacheco |

**------------------------------------------------------------------------------------------------------------------------------------------1. INTRODUCTION  
------------------------------------------------------------------------------------------------------------------------------------------**

**1.1 PURPOSE**

The purpose of Drip Cards is to allow users to plan their outfits for the week without touching their closet. By uploading their clothing pieces into the website, these turn into “Drip Cards” which then can be combined into “Outfit Cards” as they plan their outfits for the day, week, or any occasion. The “Drip Cards” combine into a “Deck of Cards” which represent the user’s closet.

**1.2 CHARTER**

1) Work together as a team to achieve goals

2) Make sure to include the other in completing assignments

3) Make sure assignments do not fall on only one person

4) Meet with each other to focus on completing tasks

5) Make improvements when you see them before you forget

6) Ask questions when necessary to not fall behind/get confusedAdd

**1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS**

* Drip Cards - Individual clothing item photos that have been uploaded by the user
* Outfit Cards - Collages of drip cards/photos that make a full outfit
* Deck of Cards - A swipe-able menu that filters through all clothing items/drip cards in the user’s closet
* World Map - A page where you can look through outfit cards that other users have posted around the world after paying the World subscription
* World Subscription – A subscription that the user pays to have access to the World Map where they can view other user’s outfits
* Drip Finder – Cloud Vision API Google Lens tool used to find the clothing items in a person’s outfits from the World Map on Google
* Dripsters – The users that use the Drip Cards applications
* Drip Specialists –Verified users (those with a following over 100 users) that upload to the World Map

**1.3. PERSONAS**

1) An everyday user who does not want to sift through their closet to plan outfits

2) An influencer who wants to share their fashion styles to others

3) A hoarder who is looking to get rid of clothing they don’t wear often

**1.4 INITIAL SCENARIOS**

A college student wants to plan out their outfit

Finding what clothes someone does not wear a lot

**1.5 INTIIAL FEATURES**

1) Cloud Vision API Google lens tool used to find clothing items and prices from World Map

2) Upload page for adding clothes from a user’s closet to their Deck of Cards

3) Collage page for creating outfit cards from drip cards

4) Deck of Cards page for seeing a user’s closet of drip cards

5) Timer that says how long an item has not been used for a collage

6) World Map, locked behind subscription, to see outfit cards others have created

**1.6 USER STORIES**

* User logs into World Map and finds another user’s outfit card with clothing they like, user uses Drip Finder to discover similar clothing they can buy online from other people’s outfit cards (a complete styled look)
* User logs into application, takes pictures of their clothing items (their drip cards) then choose the drip cards they want to use out of a carousel of cards (their Deck of Cards) and combine them together into an outfit (an outfit card)

User logs into their Deck of Cards, the user checks the timer present on drip card (which increases in time as a card is not chosen) to see if they have not worn an item for an extended period of time, this helps decide if they want to get rid of clothes in their actual closet

**------------------------------------------------------------------------------------------------------------------------------------------2. REQUEST OVERVIEW  
------------------------------------------------------------------------------------------------------------------------------------------**

**2.1 BACKGROUND INFORMAITON**

Brand new website created for those who struggle with making outfits out of the clothes they already own.

**2.2 SPECIFICIATIONS**

2.2.1 Major Features

|  |  |  |
| --- | --- | --- |
| Number | Major Feature | Level |
| MF-1 | Google lens to find prices and brands of clothing | 5 |
| MF-2 | Upload and collage uploaded photos | 5 |
| MF-3 | Timer that says how long an item has not been used for a collage | 13 |

2.2.2 Functional Requirements

MF-1 – Understanding of Google Lens in required and implementation of a recognition system is required, as well as a database for matching clothing

MF-2 – An uploading portal is required as well as a simplified collage maker. This also means there needs to be a location where the individual clothing cards, outfit cards, and deck of cards is to be located.

MF-3 – A simplified timer is required to count up from the last time a clothing card was used. It may also require the ability to send a notification to the user when it hits a certain max time.

**2.3 PROJECT PLAN SCHEDULE**

Week 3 - Getting feedback from peers concerning project and making necessary changes

Week 4 – Revise project and work on coding

Week 5 – Create software diagram and get feedback from peers

Week 6 – revise code based on feedback

Week 7 – Get feedback for final draft

**2.4 DATA DICTIONARY**

* API - application programming interface, which is a set of definitions and protocols for building and integrating application software.
* Structured Database - data that has been organized into a formatted repository, typically a database, so that its elements can be made addressable for more effective processing and analysis.
* Unstructured Database - is information that either does not have a pre-defined data model or is not organized in a pre-defined manner.

**2.5 ROLES AND RESPONSIBILITIES**

|  |  |  |
| --- | --- | --- |
| Role | Name | Responsibilities |
| Team Lead | Naomi Nash | Oversees groupwork |
| Release Manager | Daria Pacheco | Keeps group on schedule |
| Documentation and Tester | Naomi Nash and Daria Pacheco | Ensures that documentation about how to use the system matches with what the system does |
| Coder and Tester | Naomi Nash and Daria Pacheco | tests the software |

**2.7 SOFTWARE DIAGRAM**Diagram

Description automatically generated

**2.8 INFRASTUCTURE DIAGRAM**Diagram

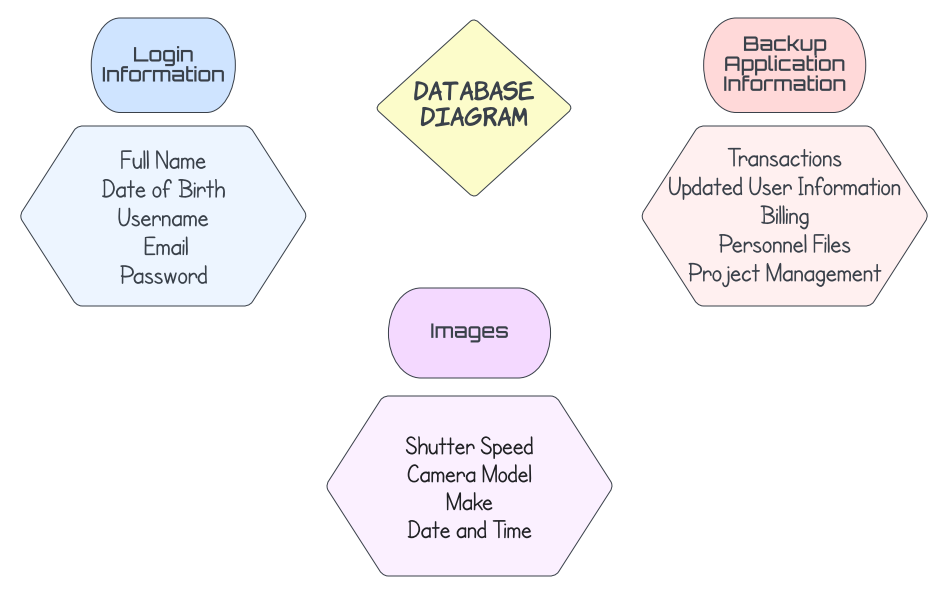
Description automatically generated

**2.9 USE CASES DIAGRAM**Diagram

Description automatically generated

**2.10 USER-BASED FUNCTIONALITY DIAGRAM**  
Diagram

Description automatically generated

**2.11 DATABASE DIAGRAM**  


**2.12 MANUAL UPLOAD TESTING DIAGRAM**Diagram

Description automatically generated

**2.13 AUTOMATED TESTING DIAGRAM**Diagram

Description automatically generated

**2.14 MEETING NOTES**

|  |  |
| --- | --- |
| 3/3 | Discussed User Stories and Requirements for project |
| 3/17 | Built upon current User Stories |
| 3/24 | Used Planning Poker to rank all current tasks (User Stories and Requirements) |
| 3/29 | Discussed what example project to use as the base for Drip Cards |
| 4/18 Morning | Met with Dr. Adams in office hours to discuss project issues |
| 4/18 Night | Created diagrams and commented and start revisions on report |
| 4/19 Morning | Finished revisions on report |
| 4/19 Night | Created diagrams and added to report |

**------------------------------------------------------------------------------------------------------------------------------------------3. RULES**  
**------------------------------------------------------------------------------------------------------------------------------------------**

**3.1 CONFIGURATION MANAGEMENT RULES**

How will GitHub be used for the Team project?

GitHub will be used for coding assistance and file organizer.

What are the rules around commits and branches?

Always issue a pull request for code changes unless working together at the time, then use commits. Make branches only when necessary, and make sure they are organized. Keep all memebers updated on any changed made

What is expected of commit messages, and how will this be enforced (if it is)?

Commit messages are only made when currently together or speaking over the phone, and it will be enforced with mutual trust.

What type of workflow will be used (feature branches? GitFlow/Git Actions? No branches?)

Workflow will be split up using branches by parts/sections of the project, i.e., development/production branches, Daria/Naomi branches, etc. which would eventually all go into a main/parent branch for the final product. Will start off working on basic web deisgn then eventually work on up.

**3.2 CODE RULES**

What is the technology stack?

Windows (Operating system), Microsoft Office (Presentation software), HTML (Programming language), GitHub, Google Lens, Azure

How can new team members get set up (onboarding)?

New team members will meet with a current member of the team. Current member will walk new team member through each part of the current code set up, ideas that have been previously tested, and rules for handling the code and introducing new code to the project.

What linters and analysis tools will be used?

HTMLHint (<https://github.com/yaniswang/HTMLHint>) and/or HTML Inspector (<https://github.com/philipwalton/html-inspector>)

Are there other ideas for tools that need to be explored?

Team may need to explored third party APIs.What technologies do some team members need to learn?

HTML needs to be reviewed, and the linters that we discovered will need to be learned. Also, GitHub will need to be practiced more as well.

How has this been factored into the project plan?

We will be reviewing technologies before we begin working on the project, during the initial stages, so that we will be prepared later. We will also learn and research more as we go.

**3.3 TESTING RULES**

At a high level, how will you test your product?

By running the code personally and testing each step of the apps use in order so that we can catch earlier problems in the program’s use.

You need to automate at least part of this this semester, so how can you automate?

One idea that we have to automate our program would be that when someone uploads a photo of their clothing, it will be automatically displayed on a “drip card” (clothing/outfit card) within the user’s “card deck” (their closet)

How does testing interact with commits (always test before commit? all tests pass? all tests pass before merge?)

We will definitely test before committing. All tests should pass before code is committed.

Security - Think about how you are going approach this.

What aspects of the stack will need more security than others?

The uploading of pictures portion, as well as user log in.

Where do you think security factors in during development?

Security should be considered at every stage of the development process.

What tools will be used?

Windows (Operating system), Microsoft Office (Presentation software), HTML (Programming language), GitHub, Google Lens, Azure,

Licensing - you must identify proper Creative Commons License and POST it to your GitHub, you must refer to the license holder.