**COM-430 (Software Engineering) Project – Group 5 – Drip Cards**

Naomi Nash and Daria Pacheco

COM-430-CA01

College of CARDS, Saint Leo University

Dr. Joshua Adams

May 3, 2022

Drip Cards GitHub: https://github.com/NaomiNash/DripCards

**Table of Contents**

[Revision History](#RevisionHistory) ………………………………………………………………………………………………………………………. 4

1. [Introduction](#Introduction) .……………………………………………………………………………………………………………………….. 5

1.1 [Purpose](#Purpose) …………………………………….……………………………………………………………………………. 5

1.2 [Charter](#Charter) …………………………………………………………………………………………………………………… 5

1.3 [Definitions, Acronyms, and Abbreviations](#DefAcroAb) ……………………………………………………………… 5

1.4 [Personas](#Personas) ………………………………………………………………………………………………………………… 6

1.5 [Initial Scenarios](#InitialScenarios) ……………………………………………………………………………………………………… 6

1.6 [Initial Features](#InitialFeatures) ……………………………………………………………………………………………………….. 6

1.7 [User Stories](#UserStories) ……………………………………………………………………………………………………………. 6

2. [Request Overview](#RequestOverview) ……………………………………………………………………………………………………………..… 7

2.1 [Background Information](#BackgroundInformation) ………………………………………………………………………………………… 7

2.2 [Specifications](#Specificiations) ……………………………………………………………………………………………………….… 7

2.3 [Project Plan Schedule](#ProjectPlanSchedule) …………………………………………………………………………………………….. 7

2.4 [Data Dictionary](#DataDictionary) ………………………………………………………………………………………………………. 7

2.5 [Roles and Responsibilities](#RolesAndResponsabilites)  ………………………………………………………………………………………. 8

2.7 [Software Diagram](#SoftwareDiagram) ………………………………………………………………………………………………….. 8

2.8 [Infrastructure Diagram](#InsfrastructureDiagram) ……………………………………………………………………………………………………….. 9

2.9 [Use Cases Diagram](#UseCasesDiagram) ……………………………………………………………………………………………………………... 9

2.10 [User-Based Functionality Diagram](#UserBasedFunctionalitiesDiagram) ………………………………………………………………………………….. 10

2.11 [Database Diagram](#DatabaseDiagram) ………………………………………………………………………………………………………….. 10

2.12 [Manual Upload Testing Diagram](#ManualUploadTestingDiagram) …………………………………………………………………………………….. 11

2.13 [Automated Testing Diagram](#AutomatedTestingDiagram) …………………………………………………………………………………………... 11

2.14 [Meeting Notes](#MeetingNotes) ………………………………………………………………………………………………………………... 11

3. [Rules](#Rules) ……………………………………………………………………………………………………………………………………………….. 12

3.1 [Configuration Management Rules](#ConfigurationManagementRules) …………………………………………………………………………………….. 12

3.2 [Code Rules](#CodeRules) ……………………………………………………………………………………………………………………….. 12

3.3 [Testing Rules](#TestingRules) …………………………………………………………………………………………………………………….. 13

[Appendix](#Appendix) …………………………………………………………………………………………………………………………………………….. 14

**------------------------------------------------------------------------------------------------------------------------------------------****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 |
| 4/22/2022 | 1.6 | Modified report to fit requirements | Naomi Nash & Daria Pacheco |
| 4/24/2022 | 1.7 | Added Customer Statements of Requirements and Stakeholder Requirements | Naomi Nash |
| 4/26/2022 | 1.8 | Added Functional and Non-Functional Requirements and added more diagrams and licensing | Naomi Nash & Daria Pacheco |
| 4/28/2022 | 1.9 | Adding required information from project checklist | Naomi Nash & Daria Pacheco |
| 4/29/2022 | 2.0 | Added Learning Plan Diagrams | Naomi Nash |
| 4/29/2022 | 2.1 | Added captions for diagrams, added step-by-step use case diagram, added tasks and edited user stories, began introductory summary | Naomi Nash & Daria Pacheco |

**------------------------------------------------------------------------------------------------------------------------------------------INTRODUCTORY SUMMARY  
------------------------------------------------------------------------------------------------------------------------------------------**

**------------------------------------------------------------------------------------------------------------------------------------------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 confused

**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

**1.4.** **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

**1.5** **INITIAL SCENARIOS**

A college student wants to plan out their outfit. The student is not very fashion conscious. However, they want to start dressing differently. Just a fresh start, turning over a new leaf, is all they are looking for. They want to discover what would look great on them. There are many people who have beautiful clothes but either rarely wear them or do not wear them at all, for whatever reason they may have. Some might feel it is the wrong color, does not fit right on their body type or simply they do not like it. With this information, which is a common factor, the college student would like to create an application where people can interchange their clothing or get advice from others with the clothing already in their own closet.

The college student attempts to initiate this application with great care. There are too many factors they need to consider. The two main objectives are security and the differences in cameras. In order to navigate the application with ease, while paying close attention to peoples’ privacy, this application would need to contain security software and a focus tool for the cameras.

**1.6** **INTIIAL FEATURES**

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

2) Collage page for creating outfit cards from drip cards

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

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

5) Outfit Card randomizer tool that takes a user’s Drip Cards and randomizes them into a brand new Outfit Card

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

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

**1.7** **USER STORIES** *(\*Note: Each User Story automatically utilizes Initial Features 1, 2, and 3. The  
 first and third User Story both also use Initial Feature 4)*

* DripQueen369, a Drip Specialist on DripCards, wants to build her influencer profile on the World Map by posting often to the World Map for other users to see and finding other user’s Outfit Card with clothing she likes to gain inspiration from to post more of her own Outfit Cards. She does this in the hopes to gain sponsorships from the clothing brand companies that use the web application to promote their clothing items and to gain followers on the application so spread her own influence and styles.
* James wants to figure out an outfit for a night out with his friends by using the Drip Cards randomizer so that he can have a new outfit out of clothes he already has in his closet and feel stylish in an outfit combination he has never worn before.
* Jessica wants to clean out her closet and maybe create some new clothing combinations in the process. By logging onto Drip Cards, Jessica checks the clothing timer on her cards to see how long it’s been since she last wore the item so that she can get rid of the items she hasn’t worn.
* Madison logs onto the World Map and sees an Outfit Card post from Taylor Swift, a famous singer/celebrity she is following, where she is wearing a beautiful and fashionable dress and she wants to find out where it is being sold. Madison views Taylor Swift’s Outfit Card and selects the Drip Finder icon so that she can find what clothing brand company is selling the dress she is wearing so that she can purchase it for herself.

**1.8 USER STORIES TASKS**

User Story 1

1. Log into DripCards account
2. Create Outfit Cards using Drip Cards she has in her Drip Deck
3. Open the World Map page of the web application
4. Post her Outfit Cards to the World Map
5. Follow other influencers on the World Map

User Story 2

1. Log into DripCards account
2. Open Drip Deck page
3. Click Random Outfit Card icon
4. Click Save button to save randomly created Outfit Card

User Story 3

1. Log into DripCards account
2. View Drip Deck page
3. View clothing timer presented on each Drip Card
4. Click Delete icon present on each Drip Card to remove the Drip Card from the Drip Deck once she donates each clothing item

User Story 4

1. Log into DripCards account
2. Open the World Map page of the web application
3. Search for specific Drip Specialist user/find Drip Specialist in followers list
4. View Drip Specialist’s Outfit Cards
5. Click Drip Finder (Cloud Vision API) icon and view search results of Drip Finder

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

**2.1** **BACKGROUND INFORMATION**

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



**2.2** **SPECIFICIATIONS**

2.2.1 Functional Requirements

MF-1 – User uses Google Lens to find new outfits Understanding of Google Lens in required and implementation of a recognition system is required, as well as a database for matching clothing

MF-2 – User takes photos of clothing and uses the upload page to create deck of cards 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 –User goes into account to view collage of outfit cards. A page of already made outfits is required to display the clothes the user had previously put together.

MF-4 – User inputs card information to create a subscription A World Map subscription is require. This may also require the ability to send a verification email to verify user card information and account.

MF-5 – User can accept or decline randomized outfit cards. A card randomizer is required to give the user different outfit cards. This may require an accept or decline button for user to keep or delete the outfit.

2.2.2 Non-Functional Requirements

MF-6 – *Timer to tell the last time a card was used*. 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.

MF-7 – *World map to display special outfits from around the world*. A page to display special event or celebrity outfits is required. An age limiter may also be required to filter out inappropriate content for minors.

MF-8 *– Third party API to protect card information*. A third party API is required to store and protect the users card information for their subscription to the World Map.

MF-9 – *Users can follow each other*. A follow and unfollow feature is required to allow user to follow friends or “style experts”. A block button may also be required to allow user to block the accounts the user doesn’t want to see.

MF-10 *– Card deck of all clothes*. A visual display of clothes is required to show user an organized list of the clothing they have uploaded from their closet.

2.2.3 Customer Statement of Requirements (CSR)

1 – As someone who has had their information stolen before, I want to ensure that my information is safe and that the security of the application is ensured so that no personal information that is not absolutely necessary will be gathered/stored by the site and used for purposes other than necessary for the application nor have the opportunity to be stolen by hackers

2 – As a concerned parent, I request that the World Map portion of the application should have either an age limit or a feature that dictates what outfit cards the user is allowed to see so that underage users do not see outfits that may be inappropriate for minors

3 – As a customer with a form of color blindness I want the coloring of the website, and overall functionalities, be catered to those who may have disabilities so that the application can be used by everyone and not be restricted so that only those who are disability free may use it

4 – As a certain customer with experience in using websites that have hefty lag times due to too many users using the application at once, I want the application have load balancing features to allow for traffic overflow so that I can avoid down times and slowed loading times when using the website

5 – As an avid of online application and annoyed with the overflow of ads that I have seen on other applications, I want, while the website can have ads to generate revenue, that there not be ads that overtake the my view of the application and are place in hindering places so that I can use the application relatively care free without having to close out ads or look for ad blockers every time I sign on

2.2.4 Stakeholder Requirements

1 – One stakeholder from a large clothing brand explained that the “Drip Finder” Cloud Vision API should direct users that are looking for a specific item from a clothing brand to that clothing brand’s site and not a smaller entity trying to resell their items for less

2 – Another clothing brand stakeholder requested that clothing ads from their business should be shown on the website over other products that do not have to do with clothing or the fashion industry

3 – One other stakeholder in the clothing brand industry requested that the World Map Challenges should feature one specific brand for a single week per month to garner attention to that brand from the users on the application

4 – A particular stakeholder requested that there be assurances put in place to ensure that the website does not go down so that their specific ads that are featured on the application do not lose exposure

5 – The last stakeholder requested that uploads to the site should be monitored to allow for no disturbing or inappropriate content that goes past application upload guidelines to be posted so that their brand or company does not receive backlash or a hit to their reputation for having a connection with the application

**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 INDIVIDUAL LEARNING PLANS**

2.4.1 Naomi Nash’s Learning Plan  
`Diagram

Description automatically generated

Diagram above shows the different elements that are needed for the project, how they learned the element and when it was learned.

2.4.1 Daria Pacheco’s Learning Plan  
Diagram

Description automatically generated

Diagram above shows the different elements that are needed for the project, how they learned the element and when it was learned.

**2.5** **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.

**2.6** **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

This diagram describes the software step process of going through the DripCards website

**2.8** **INFRASTUCTURE DIAGRAM**Diagram

Description automatically generated

This diagram is an architecture template of how the computer software and hardware components are connected.

**2.9** **USE CASE/PROCESS FLOW DIAGRAM**Diagram

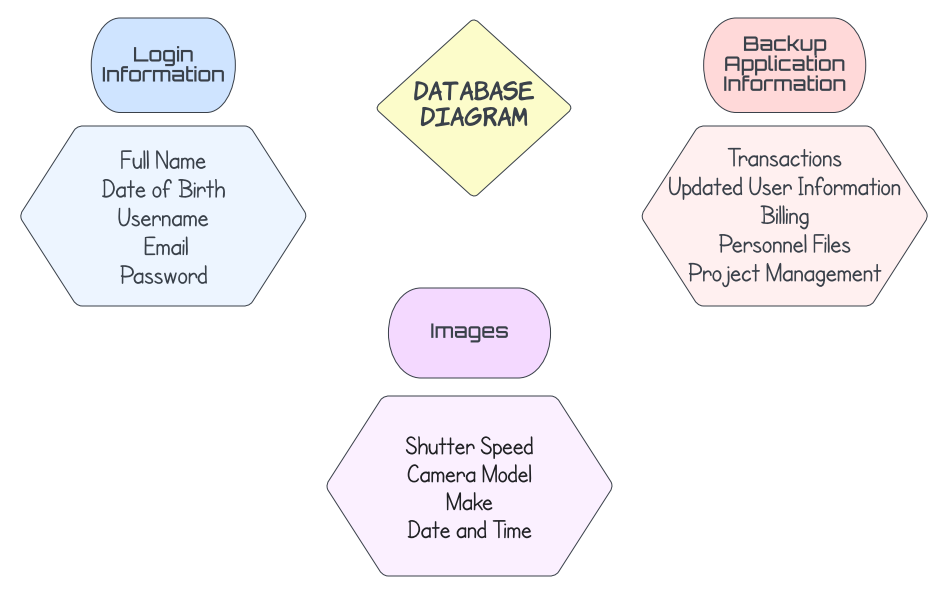
Description automatically generated

This diagram describes how the user can use the website and how the system responds to each action.

**2.10** **USER-BASED FUNCTIONALITY DIAGRAM**  
Diagram

Description automatically generated

This diagram describes the different features included for a regular user account, an admin account, and a subscribed user account.

**2.11** **DATABASE DIAGRAM**  


This diagram shows the information that is needed for each database in the system.

**2.12** **MANUAL UPLOAD TESTING DIAGRAM**Diagram

Description automatically generated

This diagram describes how a user can upload and delete photos to the Drip Card website.

**2.13** **AUTOMATED TESTING DIAGRAM**  
Diagram

Description automatically generated

This diagram shows how the admins step process of testing the Drip Card website.

**2.14 SEQUENCE DIAGRAM**  
Timeline

Description automatically generated

This diagram shows the step process of a user signing up and using the Drip Card website to upload photos and how the system interacts or verifies each step.

**2.15 PLANNING POKER RESULTS**  
Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

8

Photo above shows a list tasks that need to be complete and how long admin believes it’ll take them to complete it.

**2.16 AZURE SCRUM BOARD**Graphical user interface, application

Description automatically generated

Photo shows a scrum board which helps the admins determine which tasks should have higher priority.

**2.17 STEP-BY-STEP VISUAL DESCRIPTION OF OUFIT CARD CREATION USE CASE**Diagram

Description automatically generated

The above visual demonstrates how the user will enter information and navigate the DripCards web application when creating an Outfit Card

**2.18** **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 |
| 4/22 | Modified report to more closely reflect needed requirements |
| 4/26 | Added Functional and Non-Functional Requirements and added more diagrams and licensing; Discussed next steps for project |
| 4/27 | Met with Adams to discuss the different questions held about the project and obtained really helpful feedback |
| 4/28 | Added information to the report based off of the requirements on the checklist, added to the GitHub readme file, and edited the HTML code for the Drip Cards web application |
| 4/29 | Added Individual learning plan visuals into report as well as captions for diagrams and continued next steps of checklist including tasks, user stories edits, and introductory summary |

**------------------------------------------------------------------------------------------------------------------------------------------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

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

The license we will be using is the MIT license. It is an open-source software that allows us to modify or sell without any restrictions

**APPENDIX**

**Individual Contributions Breakdown**

Naomi Nash and Daria Pacheco completed this software engineering project as a team, with times where they worked together on portions as well as separately. The items they worked together include \_\_\_\_\_, \_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_\_\_.

As for the items they worked separately on, Naomi created the \_\_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_\_\_, while Daria created the \_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_\_\_.