COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

The Spirits Collective

Jordan Barker
Jessica Canaday
Erin Harl
Aajish Marahatta

Revised June 21, 2026

CSCI 370 Summer 2026

Dr. Iris Bahar

Table 1: Revision history

Revision	Date	Comments
New	May 20, 2026	Completed Sections: I. Introduction II. Functional Requirements III. Non-Functional Requirements IV. Risks V. Definition of Done XI. Team Profile References Appendix A - Key Terms
Rev – 2	May 31, 2026	Updated Sections: Completed Sections: VI. System Architecture
Rev – 3	June 7, 2026	Updated Sections: II. Functional Requirements Completed Sections: VII. Software Test and Quality VIII. Project Ethical Considerations
Rev – 4	June 14, 2026	Updated Sections: Completed Sections: IX. Project Completion Status X. Future Work XI. Lessons Learned
Rev – 5	June 21, 2026	Updated Sections: All Sections Completed Sections: XII. Acknowledgements

Table of Contents

I. Introduction

Imagine discovering a world-class, small-batch bourbon or a unique craft gin, only to realize it is completely unobtainable because of the archaic legal walls surrounding the alcohol industry. While the distilling industry is not the largest group in the United States, there has been significant growth in recent years, resulting in an increase in collaboration between businesses. However, the regulations to legally sell alcohol has made these collaborations quite laborious. Distilleries would have to get in contact with one another, find and file all the legal paperwork, coordinate deliveries and/or pickup, and many other tedious tasks. Additionally, in the past, distilleries have been unable to sell their own product at other locations due to manufacturing regulations or lack of distributors. This makes it difficult and almost impossible to sell their product at other locations, putting them at a disadvantage to local liquor stores. This existing system is inefficient, time consuming, and simply not worth it for small distilleries. Our project aims to address this issue by establishing a centralized platform that can handle the entire process start to finish. Our team is working with Aubrey Wigner, CEO of Method and Muse Spirits, to design and implement a website that facilitates the transfer of goods between distilleries across the US. This web-platform simplifies communication, accessibility, and visibility between distilleries while also letting them compete with liquor stores by having outsourced goods rather than only personal goods.

II. Functional Requirements

The overall function of this website depends heavily on backend databases, which store information input by us like the list of licensed distilleries in the US, and information input by the users, such as their contact information, their products, and their logins. The most critical overlap of these databases is when checking for user authenticity against the Alcohol and Tobacco Tax and Trade Bureau (the TTB) database with all the licensed distilleries. By comparing the user entered license to the official database, we can check against verified data to make sure the only users are the ones who are allowed to be here.

The following requirements matrix outlines the essential functions the DSP2DSP web platform must perform to support everyday operations for craft distilleries. These functional requirements were gathered through meetings with our client and broken down by priority to establish a clear development roadmap, highlighting critical technical dependencies such as the official TTB database and secure Stripe API integrations.

Requirement ID	Requirement	Priority	Dependencies
#001	Users must be registered with distillery by TTB	High	TTB licence database
#002	User must be able to create account if registered	High	TTB registration
#003	User must be able to log in if account exists	High	Account creation, username/password database
#004	User must be able to purchase liquors through the website	High	Sale listings, purchasing connection though Stripe, label change, licence for trade
#005	User must be able to post their products to	High	None

	sell		
#006	Users must be able to message other users	Medium	List of users, user emails
#007	User must be able to dispute orders/charges	High	List of past orders
#008	Website should show possible shipping costs for orders via different carriers	Low	Order data, limits for each carrier
#009	Website should verify user login with 2factor authentication	Medium	List of user logins, user emails, input for one time code
#010	Users should have direct access to the forms and paperwork they need to fill out to legally buy liquor	Medium	The legal documentation needed to register and verify the trade
#011	Users must be able to edit and remove their products	High	A database that links each listing to the user (either through keys or usernames)

III. Non-Functional Requirements

Requirement ID	Requirement	Priority
#001	Private account information must be encrypted for safety purposes	High
#002	All pages in UI must load within seconds	High
#003	Verification process for a distillery should also happen in a reasonable time	Medium
#004	Website should be able to handle many distilleries accessing website at the same time	High
#005	The database should be able to	High

	handle data growth in the case distilleries add more and more products over time	
#006	Stock information should update in real time with purchases so it is immediately known if a product is out of stock	Medium
#007	Users should be immediately notified when a message is sent to them to facilitate immediate communication	Medium

IV. Risks

The list of technical and skill-based risks detailed below was compiled by breaking down the platform's core functional requirements and identifying architectural single points of failure early in the lifecycle. While our client, Aubrey Wigner, did not assign the formal impact or likelihood metrics, he provided critical industry context during our early discovery sessions, emphasizing that regulatory license failures carried a "Major" impact. Ultimately, these mitigation plans directly shaped our development; identifying these threats early allowed us to successfully implement Firebase and Stripe Connect to offload security liabilities, though legal complexities forced us to adapt our shipping mitigations to manual processes. Thinking through these hurdles ahead of time was invaluable, as it prevented late-stage architectural re-writes and kept the team focused on a realistic project scope.

Technical Risks

Risk	Description	Impact	Likelihood	Mitigation Plan
Creating/altering legal documents not done correctly	Ensuring correct information so that all transactions are done legally	Major	Unlikely	Work closely with client to get correct information to provide customers with valid documents
eCommerce issues (wrong amount charged, invalid payment, etc)	Ensuring all transactions are accurate to avoid any disputes/issues	Major	Unlikely	Using a reliable outside source that is known to work
Integrating all the separate components together (backend, frontend, outsourced code, etc)	Ensuring all major components like purchasing, viewing, account creation, etc. all flow together without issues	Moderate	Likely	Work piece by piece and make sure small components work together before adding the next step
Authorization issues	Non registered distilleries being verified or legally	Major	Unlikely	Ensuring the database is correct and up to date

	registered distilleries being denied.			
Data leaks/exposures	User credit card data or product and account information being stolen	Major	Unlikely	Using secure and well protected software that has never been hacked or leaked data
Scalability/performance	Many users trying to order at the same time might cause performance issues	Moderate	Unlikely	Using a scalable software that allows more users at high speeds for a slight increased price point

Skills Risks

Risk	Impact	Likelihood	Mitigation Plan
No experience working with websites with high levels of security	Major	Unlikely	Using a third party software to handle the security instead of handling it ourselves
No legal background with alcohol markets and eCommerce	Moderate	Unlikely	Researching the legal requirements and documents, and relying on our clients knowledge.
Very small amount of experience with HTML and front-end development	Moderate	Unlikely	Watching tutorials to learn HTML

V. Definition of Done

For our project to be considered done, it will be accessible to the public. Legally registered distilleries will be allowed to create an account with all of the necessary information to verify them. Distilleries will be able to create posts of the products they have available. Those posts should contain the product's name, quantity and required information for the legal forms. They will also be able to view the list of products offered by other distilleries with accounts on the websites. They will also be able to filter the product by distillery name and product type. They will be able to click on a product to see more information, and they will be able to add that product to their cart. Users will then have the ability to checkout with the items in their cart. The seller will be provided with a printable shipping label that they can use when shipping their product. Both parties will be given the correct information and documents to fill out to make the transfer legal. The buyer should receive the product within the given lead time. If there is an issue with a product, the buyer will be able to create a dispute. All users will be able to send messages and will then get an email if they receive a message. After a user makes a purchase they will receive an email receipt. Overall, distilleries should have a safe, legal and easy system of buying other distilleries products to put on their shelves.

VI. System Architecture

System Design Diagrams

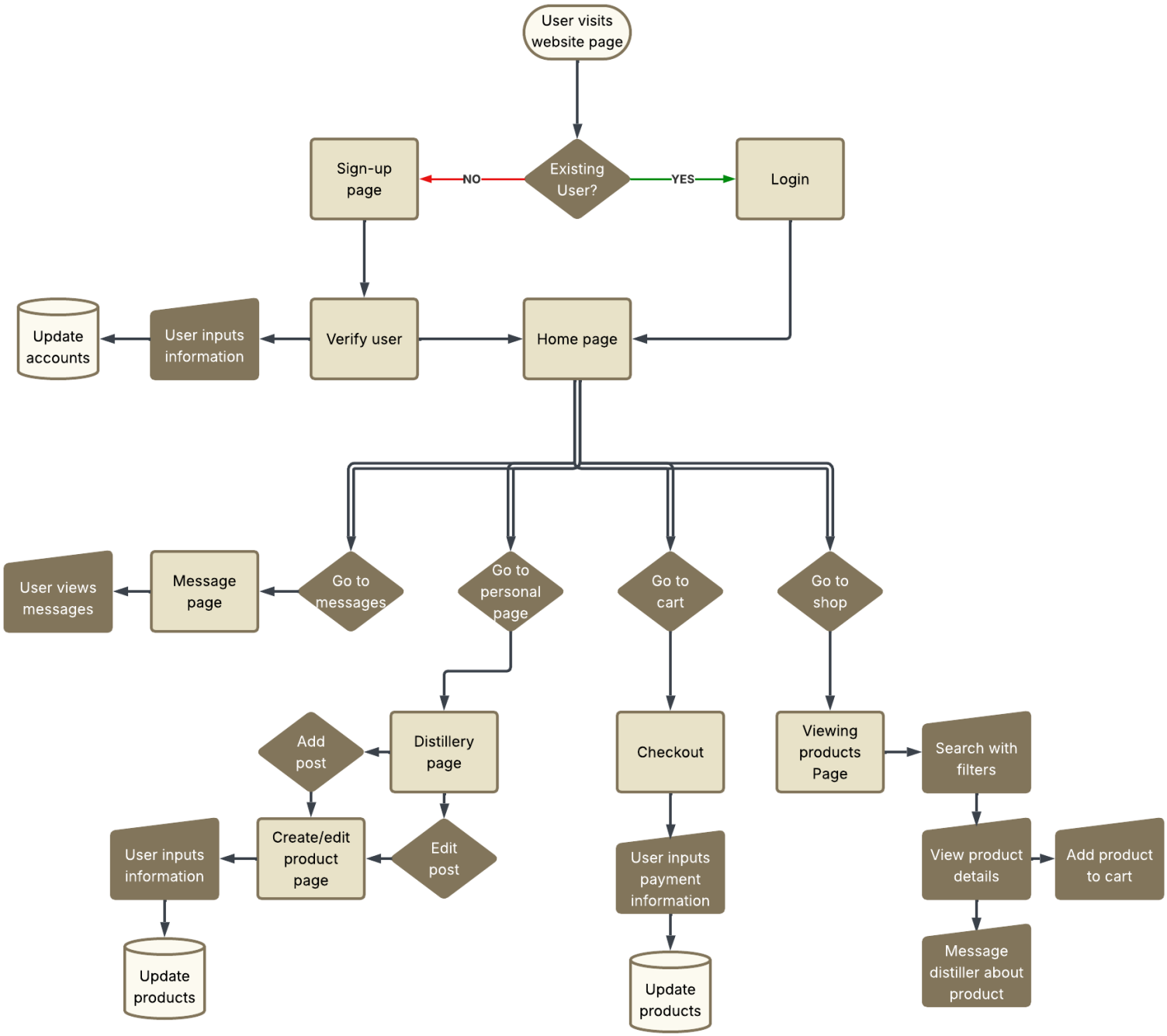

Figure 1: Typical workflow of website user

This workflow diagram illustrates the overall user interaction flow for the distillery marketplace website, beginning from the moment a user visits the site. The system first determines whether the visitor is an existing user. Existing users are directed to the login page, while new users are guided to the sign-up page to create an account.

During registration, users enter their information, which is verified and stored within the system to update account records. Once authenticated, all users are directed to the home page.

From the home page, users can navigate to several core sections of the platform: messages, their personal distillery page, the shopping section, or the cart and checkout process. The messaging feature allows users to access conversations and view messages directly through the message page.

Users visiting their personal distillery page can manage marketplace listings by creating, editing, or updating product posts. Product information entered by the user is stored in the database, ensuring that marketplace listings remain current and accurate.

Within the shopping section, users can browse available products, apply search filters, and view detailed product information. From the product details page, users have the option to contact the distillery directly regarding a product or add items to their cart.

The checkout workflow allows users to submit payment information and finalize purchases. Once completed, the system updates product inventory and records to reflect the transaction. Overall, the workflow demonstrates how the platform integrates user authentication, product management, messaging, browsing, and purchasing into a centralized marketplace experience for craft distilleries.

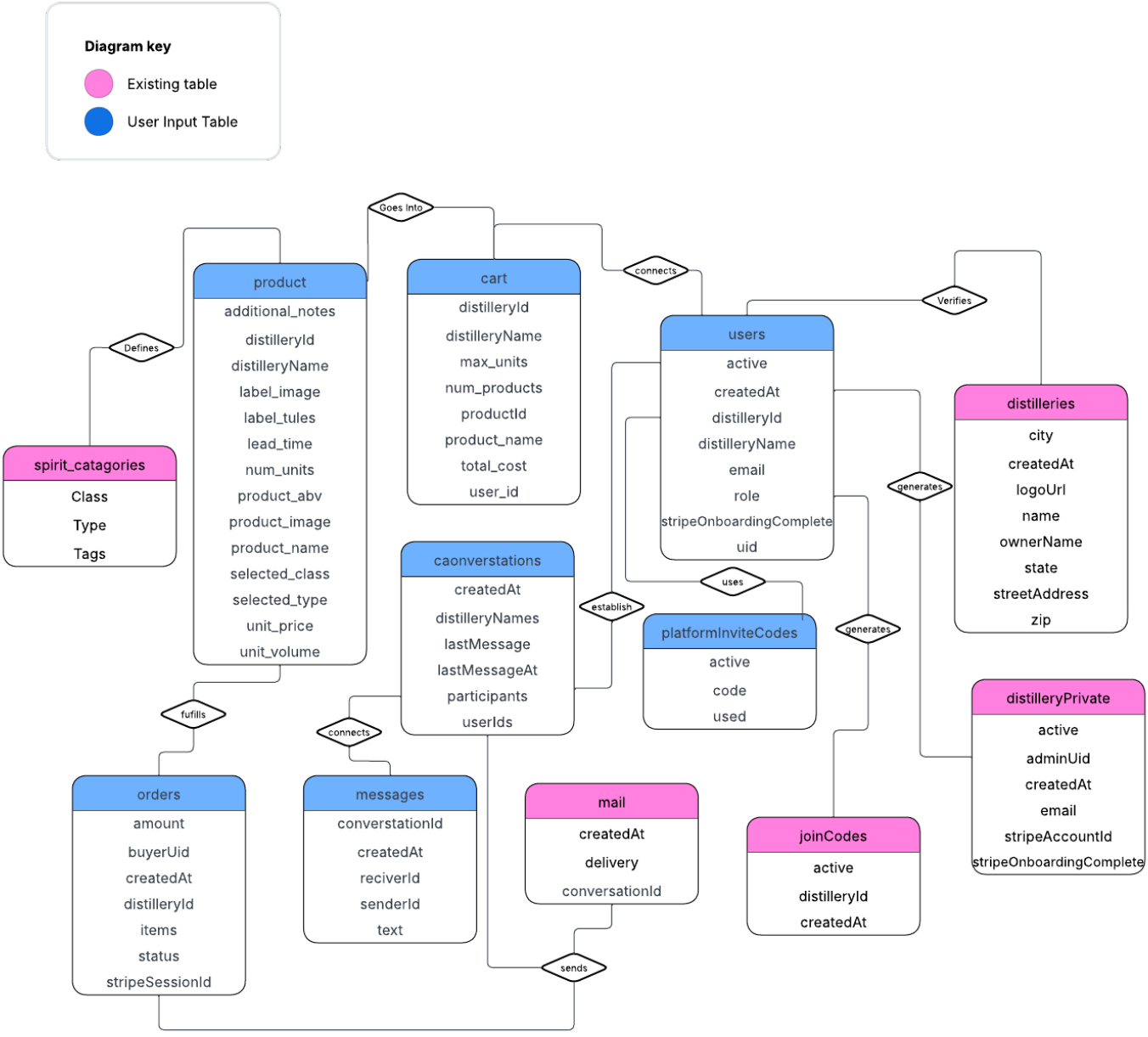

Figure 2: Base Database Schema

Figure two illustrates the database schema for the application and shows how product, user, storefront, and regulatory data are connected. The schema combines existing data sources (shown in pink)-- pulled from official databases like the TTB site or auto generated by the site– with user-generated tables (shown in blue) that are formed from data the user inputs at different parts of the site.

The Spirit Categories table defines the classifications and tags used for products, while the Labels table stores product-specific information such as ABV, bottle volume, and label images. Together, these tables provide the foundational data used to populate the Product table, which contains inventory details including spirit type, bottle size, pricing, lead time, and available quantity.

Users create accounts through the User Logins and User Account tables, which store authentication information and account details. Each user account is linked to a Store Front, allowing distilleries to manage storefront information such as addresses, logos, and messaging details. Products can be displayed through storefronts and added to a user's Cart, where selected quantities and total costs are tracked.

To ensure security for users, accounts are connected to the TTB Registered Distilleries table, which stores license numbers and business addresses. This relationship allows the system to verify that participating distilleries are properly registered before conducting transactions. Overall, the schema supports product management, storefront operations, user authentication, shopping cart functionality, and license verification within a single integrated database design.

Risk Treatment Plan

In order to address the risks associated with our project, both with our own lack of experience and with the inherent security and legal issues that may arise, we plan to spend lots of time in research and testing. We also plan to make use of third party software in places where it makes sense for security and ease of use for the customer. We want to make sure that we are developing a unique system that fits the needs of our client while also not completely reinventing the wheel. We used Firebase to host our website and handle the back end, and we used Stripe as an ecommerce platform. Using these gave us added security for our databases, account information, and payment processing. It also helps all the separate systems integrate without messing with each other or making use of the system inconvenient for the user. These design decisions help treat the current risks in our project.

Risk ID	Risk Description	Category	Likelihood (1-5)	Severity (1-5)	Risk Score	Mitigation
001	Legal documents altered wrong, so that they break the law	Technical	1	5	5	Follow closely to the legal documents we were provided, and double check all fields, have strict requirements for what information must be provided
002	Payment processed wrong, people don't receive the money they are owed	Technical	1	5	5	Use an outside reliable platform to process payments so that we are not responsible for the payments ourselves and are ensured of its reliability

003	Integration of components failure	Technical	3	4	12	Work slowly piece by piece, documenting architectural design decisions and having good communication so that we always understand where we are and where we need to go
004	Authorization issues, non registered distilleries being verified or registered ones being denied	Technical	1	5	5	Using a database with permissions restricted, not using AI for the security of the database and keeping the database updated and accurate
005	Data leaks of account information	Technical	1	5	5	Choosing good platforms that have no history of data breaches, we chose Stripe and Firebase which are both secure and reliable
006	System unable to process large amount of people at the same time	Technical	1	4	4	We are using software that is easily scalable, that's payment increases at a reasonable rate to the increase of users
007	Team has little website design experience	Skill	1	2	2	The team has done thorough research into firebase, react and all the other tools we used in developing the site
008	Team has no legal background in alcohol trading	Skill	1	3	3	The team researched the legal requirements and documents, and rely on our clients knowledge

009	Team has no experience securing websites to protect user data	Skill	1	3	3	We used a third party software instead of handling security ourselves, and managed our software security settings to ensure user data is protected

Technical Design Issues

One technical design our team ran into was finding a true secure way of accessing the database. Originally the frontend was responsible for creating, editing and accessing the database so it was impossible to write rules that would ensure the data could not be accessed by users even if it couldn't be accessed by someone without an account. To fix this, we realized we would have to transfer the responsibility of account creation to a function that way updating and accessing the sensitive parts of the database was only done on the server side and the client side never had to have permissions they shouldn't have.

Another technical issue we ran into was the loading of pages. Originally, our team had chosen to use React-Router for page navigation which posed a security risk. The problem with using just this without any safeguards is that it sends all of the code at once to the user for every page, even if those pages have security locks on them. So no matter how secure the Firstore rules we wrote were, users would have access to code they should not have. Our solution was to use lazy loading, which means that the backend holds all code for each page until the user needs it. So when a user accesses a new page, first the security checks are made, and then the page and its code are sent to the user.

VII. Software Test and Quality

Our testing strategy utilized a layered approach combining comprehensive user interface (UI) validation, end-to-end integration reviews, and rigorous role-based permission tracking within Firebase. Our client, Aubrey Wigner, was directly involved through weekly user experience testing and live demonstrations, which helped us align the software's behavior with real-world distillery operations. To identify and test critical edge cases, the team brainstormed potential system failure points, such as simultaneous inventory purchases on a single item, access attempts from disabled user accounts, and expired verification codes. We then systematically simulated these scenarios to ensure the platform handles unexpected errors gracefully without corrupting database or transactional records.

Functional Requirement Testing

ID	Purpose	Description	Tools Required	Acceptance Threshold	Edge Cases	Results
001	Ensures only verified users can access website	Test that only users who have a legally registered distillery permit are verified	Creation of Join Code: invite system	If a user is verified they actually have a legally registered permit	Disabled accounts should not be able to access the page anymore with their account	Functional, invite code ensures this is secure
002	Allows all verified distilleries to use website	Tests that all legally registered distilleries can create accounts linked to their distillery	User Interface Testing	For all registered distilleries, a distillery can create an account	Newly registered distilleries who were added to the list later	Functional, file of legally registered distilleries accessed
003	Prevent duplicate accounts and allow account data to be stored for future	Test that if an account already exists for an email another cannot be created, and a user when logged in can see their profile page and post page	User Interface Testing	Creating duplicate accounts is impossible, for all logged in users the correct pages display	Disabled accounts cannot log in, a new account cannot be created with that email even though it has been disabled	Functional, duplicates cannot be created, data is securely stored
004	Ensure legality and functionality of purchasing	Test that the website holds no funds, and that if a user purchases an item they actually receive it.	Integration Testing	Users who do not receive product, less than 1% have dispute handled correctly, the website at no times holds any funds	Users purchasing products that get lost, user orders a product that a distillery forgot to remove, two users try to purchase all that remains of a product at the same time	Functional, money is secured through Stripe disputes handled well
005	Ensure distilleries can post their	Test that if a distillery creates a	User Interface	Every product on every users	Products that are out of stock	Functional, products

	available product and other distilleries can see their post	product it appears on their posts page and on the shop page for others to see	Testing	post page is also on the shop page	should also appear on both pages	posted accurately and timely
006	Provides method of communication between users	Test to see if when a user sends a message the recipient receives it in a timely manner with all the sent information intact	User Interface and User Acceptance Testing	All sent messages are received within a time period of five minutes completely correctly	Messages sent to disabled accounts should send an error message and not be sent	Functional, messages are sent and received, email sent
007	Provides solution if issues with transaction/delivery	Test that both true and false disputes are handled correctly	Integration Testing	All disputes are dealt with within 5 days correctly based on provided evidence	A user provides false evidence that a product did not arrive, a credit card was proven to be stolen when purchase was made	Functional, handled through Stripe and FedEx
008	Users should be able to choose the cheapest shipping	Test that shipping prices are accurate and that all appear	User Interface and User Acceptance Testing	For all shipping options available for a product the prices are listed	For orders over a certain volume some shipping companies won't ship it, they should not be listed	Change: now all shipping is outsourced and not handled directly
009	Ensures user is verified to login under that distillery account	Test that a user account cannot be created unless the 2 factor authentication is fulfilled correctly	Integration Testing	A user gets sent an email and uses code in email to verify before their account is created	If there is a time window then users who try code after that window must send another code, the initial code should fail	Functional, user distillery checked when accessing protected pages
010	Transactions need to be legal	Test that a user walkthrough of the paperwork instructions would lead to a correct and legal filing of every form	User Acceptance Testing	Someone following the instructions could not file incorrectly, they are not vague or incomplete	People who are not tech savvy should still be able to understand the steps and follow without difficulty	Functional, paperwork and legal instructions provided

011	Allows users to keep product availability/information up to date	Test that if a user edits or deletes a product those changes are reflected in the shop page	Integration Testing	If the UI or database for products is changed, the website should reflect those changes in under 1 minute	Disabling or deleting a distillery account should also delete its product from the shop page, but this should not happen for a user account that is deleted otherwise	Functional, products are updated on page and in database, product page updates almost immediately

Non-Functional Requirement Testing

ID	Purpose	Description	Tools Required	Acceptance Threshold	Edge Cases	Results
001	Ensure account information can only be accessed by its connected distillery admin and itself	Test that an admin can see its employee data, every employee only has access to their own data	User Interface Testing, Code reviews	No user has access to pages or data they should not be able to view	Disabled users should not be able to see their account, admins can still see their account	Functional, firestore rules enforce protection on pages and information
002	Provides speed and ease of use to users	Test that pages load in a reasonable amount of time (< 1 minute)	Dynamic Analysis	Pages load in under 5 seconds	The initial loading takes the longest, best to test	Functional, all pages load in under 5 seconds
003	Allows setting up an account to be fast and painless	Test that a user validating their account spends little time on that part of the process	User Acceptance Testing	The validation process takes under 5 minutes to complete	Users who are not logged into their email on the device they are using to create an account	Functional, onboarding takes under 5 minutes if user has all information readily available
004	Ensure speed does not decrease with increase of user traffic	Test that the speed of the site does not decrease at high volumes of traffic	Dynamic Analysis	The speed of the site decreases by a scale of less than 0.1% slower for every 1000 users	Multiple accounts being created at the same time, many transactions being made at once	So far, functional. User traffic has not affected the overall speed

005	Allow growth of the site through a scalable database	Test that the database storage can increase by significant amounts	Static Analysis	The database is capable of holding 5000 users and their data	The amount of product entries for each distillery may hold a lot of data	Functional, the database can hold many entries and scalable prices
006	Ensure users cannot buy a product that is out of stock	Test that the add to cart button is disabled for out of stock products	User Interface Testing	Products that are out of stock cannot be added to any users cart	If a user adds something to their cart and it becomes out of stock before they check out	Functional, out of stock products cannot be added to cart
007	Provides easy and fast communication	Test that when a user gets a message on the site they are notified right away by email	Dynamic Analysis	A user receives an email about a message within 5 minutes of getting the message on the website	Either user has slow or no internet connection	Functional, messages arrive within 5 minutes on both the site and by email

VIII. Project Ethical Considerations

Relevant ACM/IEEE Principles

Several principles from the ACM/IEEE Software Engineering Code of Ethics are particularly relevant to the development of the distillery distribution and e-commerce platform. *Public Interest* is one of the most important principles because the platform facilitates the sale and distribution of alcoholic products. The system must provide accurate product information, comply with age restrictions, and ensure that users can access products responsibly and legally. Developers have a responsibility to protect users and promote safe and ethical use of the platform. *Product Quality* is also critical because users and distilleries depend on the platform to browse products, manage accounts, and complete transactions. The software must function reliably and accurately to prevent errors that could negatively affect customers or businesses. Additionally, the *Client and Employer* principle applies because both customers and distilleries rely on the platform to conduct business. The development team must act in the best interests of stakeholders while ensuring that ethical and legal standards are maintained. Finally, the *Profession* principle emphasizes maintaining the integrity of the software engineering profession through responsible development practices, strong security measures, and protection of user information.

Potential Violations

The principles most at risk of being violated are Public Interest, Product Quality, and Privacy/Confidentiality. The *Public Interest* principle could be violated if the platform fails to properly restrict access to age-restricted products or provides inaccurate information to users. Such failures could result in legal consequences, harm to users, and damage to the platform's reputation. The *Product Quality* principle is at risk if insufficient testing allows defects to reach production. Software failures could lead to incorrect order processing, inaccurate inventory data, failed transactions, or poor user experiences. The principle of *Privacy and Confidentiality* is also particularly important because the platform stores user account information and transaction data. If adequate security controls are not implemented, sensitive customer information could be exposed, potentially resulting in financial loss, identity theft, and a loss of trust among users. This risk is exemplified by the catastrophic data breach of the women's safety app Tea, where an unencrypted legacy Firebase storage bucket was left entirely unprotected and publicly accessible without authentication. Trolls on

4chan discovered the vulnerability and weaponized the leaked files, exposing 72,000 user images, which included 13,000 highly sensitive verification selfies and government-issued driver's licenses. Compounding the failure, bad actors extracted embedded EXIF metadata from the photos to build searchable maps tracking users to their approximate physical locations, while a subsequent flaw exposed over 1.1 million private direct messages containing deeply intimate personal communications. For our platform, which handles sensitive corporate data, financial transactions, and official TTB license verifications, this real-world failure underscores the absolute ethical requirement to enforce rigorous backend security definitions and completely block unauthenticated public access to any piece of system data.

Application of Michael Davis Ethical Tests

Harm Test

The Harm Test asks whether a decision causes less harm than available alternatives and whether the benefits outweigh the potential harms. For this project, collecting customer account information is necessary to support purchases, order tracking, and communication between users and distilleries. However, storing personal information creates the risk of data breaches and unauthorized access. To minimize potential harm, the platform implements secure authentication, encrypted communications, and role-based access controls. These safeguards allow the platform to provide its intended benefits while reducing risks to users.

Reversibility Test

The Reversibility Test asks whether a decision would still seem acceptable if the roles were reversed. As developers, we must consider how we would feel if our own personal information were stored on the platform. We would expect our data to be protected, transactions to be processed securely, and privacy to be respected. Applying this test supports the implementation of strong security measures and transparent data handling practices to ensure that all users receive the same level of protection that we would expect for ourselves.

Publicity Test

The Publicity Test asks how a decision would appear if it were reported publicly. If a security breach exposed customer information due to inadequate testing or weak security practices, the incident would likely damage user trust and the reputation of the platform. Knowing that development decisions could be subject to public scrutiny encourages the team to follow ethical software engineering practices, conduct thorough testing, and prioritize the protection of user data.

Ethical Considerations Related to Software Quality

If the software quality plan is not implemented properly or does not provide sufficient coverage, several ethical concerns may arise. Defects in the system could result in inaccurate product listings, failed transactions, incorrect order fulfillment, or service outages. Such issues may inconvenience users, negatively affect businesses that rely on the platform, and reduce trust in the system. Inadequate quality assurance may also leave security vulnerabilities undiscovered, increasing the risk of unauthorized access to sensitive information. To minimize these risks, the project includes comprehensive testing of functionality, security, database operations, performance, and user experience throughout the development lifecycle.

Security Considerations

Security is a critical aspect of the project due to the collection and storage of user account information and business data. The platform implements secure authentication and authorization mechanisms to ensure that users can access only the resources appropriate to their roles. Passwords are securely hashed before storage by Firebase, and all communications between users and the platform are protected using HTTPS encryption. Additional protections were implemented against common web application vulnerabilities, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Role-based access controls are used to distinguish permissions between customers, distilleries, and administrators. Regular security testing and vulnerability assessments are also conducted to identify and

address potential weaknesses before deployment. These measures help protect user privacy, maintain trust in the platform, and support compliance with ethical and professional software engineering standards.

IX. Project Completion Status

Featured Implemented and Summary of Feature Performance

The DSP2DSP platform successfully implemented all the needed functionality to facilitate communication and product exchange between distilleries. A secure authentication system was developed which utilizes private sign-up codes to make sure only authorized distilleries and users can create accounts and access the website.

The marketplace serves as the core feature of the website, allowing distilleries to browse products offered by other participating distilleries. Users may select any product to view a detailed popup with all available information associated with that product, including an image of the product, alcohol by volume (ABV), stock, spirit type, and additional details that could be relevant to buyers. To improve usability, users are able to search for products by spirit name or type and apply filters based on spirit categories and active distilleries so they can efficiently search the products that meet their specifications.

A product creation page was also added which allows distilleries to add and maintain their own product listings. Standard drop-down menus are utilized, automatically populating compliance-related information and ensuring consistency across all listings.

To help with the purchasing workflow, a shopping cart system was added where products selected from the marketplace are added directly to the cart. Here, users can review their selection, adjust the quantities of the selected products, and remove products from their cart if they ever change their mind. The order total displayed on the right dynamically updates based on user modifications. When the user proceeds to checkout, it takes them to Stripe where their purchase can officially be confirmed.

Additionally, a messaging system was developed to facilitate direct communication between distilleries. Users can initiate conversations through a searchable interface that displays all active distilleries on the platform. Once a distillery is selected, all active accounts under that distillery are displayed so users can choose a specific recipient for their message. A conversation is then created within the user's inbox, enabling real time communication regarding products, orders, or other business-related matters. When a message is sent, an email regarding the message is sent to the recipient with the message details and a link to the message page on the website. This email system is also integrated with orders, where buyers and sellers are notified about the contents of a transaction after the payment is processed.

From a performance perspective, the platform met its primary usability requirements. All pages load immediately when clicked on, creating a responsive user experience. Information displayed throughout the platform is retrieved dynamically from the database and consistently loads within a timeframe that does not inconvenience the user. This includes product listings, user account information, messaging data, and cart contents. The emails sent to recipients also are received within a minute. Testing showed that the implemented features performed reliably and supported the intended workflow that distilleries need with product discovery, communication, and the management of transactions.

Features not Implemented

All major objectives were successfully completed, however, several low-priority enhancements were not implemented due to time constraints and the complexity of integrating these additional features. One limitation that is currently in place is that buyers can only place orders from a single distillery at a time. Due to challenges with payment processing, shipping, and order management, the checkout process currently in place does not support multiple distilleries within the same order. Future development would expand this functionality to support multi-distillery purchasing, creating a more practical buying experience for distilleries. This complexity also did not allow us to include

shipping within the payment process. When users log in for the first time, they are prompted by Stripe for their payment information but the next step would be linking a FedEx account so the shipping is not a manual process. Another feature not implemented is an in-application notification system. Currently, users must manually check their inbox or their email to see whether new activity has occurred. A future enhancement would be providing real-time notifications when a user receives a new message or when an order is placed with their distillery. These notifications would increase responsiveness and a more convenient overall user experience.

Client-Initiated Updates and Platform Maintainability

A core requirement for the long-term sustainability of the DSP2DSP platform was ensuring that our client, Aubrey Wigner, could easily maintain and update administrative information without needing a developer to rewrite codebase files. To achieve this, the platform utilizes a dynamic configuration model driven entirely by Firebase Firestore.

The client can seamlessly execute updates in two primary ways:

- **Inventory and Listing Management:** Through the frontend UI, the client or any authorized distillery administrator can instantly create, edit, or remove product listings via standard dropdown menus and input forms. These updates process instantly and update the active shop database in under a minute.
- **Regulatory Data Updates (TTB Database):** To update the master list of legally permitted distilleries or modify private sign-up codes, the client can use the intuitive Firebase Console interface. Rather than altering source code or redeploying the application via the CLI, the client can simply modify entries directly inside the Firestore collections. Any changes made to the database instantly propagate to the live website, allowing the client to independently manage platform access, permissions, and compliance boundaries with minimal technical friction.

X. Future Work

While the current DSP2DSP platform provides a functional marketplace for connecting distilleries and buyers, several enhancements could improve usability, business operations, and long-term maintainability. The following recommendations represent potential future developments for the platform.

Past Orders and Purchase History

A valuable addition would be a Past Orders page that allows users to review previous transactions. Buyers could easily view their purchase history, reorder products, and track spending patterns. Distilleries could use order history data to identify their most popular products and better understand customer demand.

Resources Required

- Additional Firestore collections or database tables for storing completed order records
- Updates to the React frontend for displaying order history
- Additional cloud storage for transaction data

Knowledge/Skills Required

- Database design and data modeling
- React development
- Firebase Firestore querying and optimization

Estimated Time Required

- Approximately 1-2 weeks for design, implementation, testing, and deployment.

Distillery Management Dashboard

A dedicated dashboard for distilleries would streamline order fulfillment by displaying incoming orders that have not yet been shipped. This dashboard could provide order status tracking, shipment management tools, and sales summaries to help distilleries manage daily operations more effectively.

Resources Required

- Additional frontend components and dashboard views
- Firestore integration for order status tracking
- Potential use of charting libraries for sales visualization

Knowledge/Skills Required

- React development
- Database management
- User interface and dashboard design

Estimated Time Required

- Approximately 2-3 weeks for design, implementation and testing.

Shopping Cart Improvements

The current shopping cart experience could be enhanced through the addition of a “Clear Cart” button. This feature would allow users to quickly remove all items from their cart without deleting products individually, improving overall usability and convenience.

Resources Required

- Minor front end modifications
- Updates to cart state management

Knowledge/Skills Required

- React state management
- User experience design principles

Estimated Time Required

- Less than one day to implement and test.

Delayed Payment Options

Future versions of the platform could support delayed payment methods, allowing approved buyers to place orders and pay at a later date. This functionality would better reflect business-to-business purchasing practices and provide greater flexibility for participating distilleries and distributors.

Resources Required

- Integration with payment providers supporting invoicing or net-payment terms
- Additional database structures for tracking outstanding balances
- Potential licensing or transaction fees associated with payment processing services

Knowledge/Skills Required

- Payment gateway integration
- Financial transaction management
- Security and compliance practices

Estimated Time Required

- Approximately 3-5 weeks depending on payment provider requirements and testing needs.

Migration to Current Software Versions

The project currently utilizes software dependencies that may become outdated in October of 2026. Upgrading to the newer, actively supported versions of Firebase and related libraries would improve security, performance, and maintainability while reducing technical debt. Future development should prioritize keeping dependencies current and replacing deprecated functionality when necessary.

Resources Required

- Updated development environment
- Access to current framework documentation
- Testing infrastructure to verify compatibility

Knowledge/Skills Required

- Dependency management
- React and Firebase migration procedures
- Software testing and debugging

Estimated Time Required

- Approximately 1-2 weeks depending on the number of breaking changes introduced by newer versions.

These enhancements would expand the platform's functionality, improve the user experience for both buyers and distilleries, and ensure the long-term sustainability of the DSP2DSP marketplace.

XI. Lessons Learned

What Worked

- Using github to collaboratively work on and combine our code worked wonderfully. We were initially unsure about the best and most efficient way to work on the website code and be able to combine our work. We eventually settled on github, which allowed us to create separate branches for every page we created. This gave us space to test and develop different parts of the site without fear of messing the whole project up, and once the page was complete we could easily combine the work done on the branch into the main project. In the future, we now know this is a good option for collaborative codebases.
- Using a pre-built backend service worked very well for us. Firebase provided us with adequate shortage, and allowed us to store our data in a database that everyone could easily access to view the current stored data and add data/storage as needed. It also provided us with several optional plug-ins such as a cloud storage that allowed us to store pictures and easily access them. In the future, we know that using basic pre-built backend services help make the project more organized, and come together more efficiently.
- Creating an invitation only system allowed us to both better control the number of users we have (and allow future adjustments of the site accordingly) as well as add an extra layer of verification to the platform to ensure that only authorized distilleries are accessing and using the website. It also gave us a way to allow multiple users to attach themselves to a distillery, giving employees access to listing and purchasing powers. In the future, we know that adding in extra verification methods adds security to the site at the cost of a little extra effort.
- We had very good communication with the client and were able to maintain consistent and robust contact with the client. This allowed us to get a clear idea early on of what the client wanted, and it made it easy to get necessary information and the required go-aheads from the client, which helped us maintain a solid schedule and prevent us from having to redo a bunch of work. In the future, we know to approach other clients in a similar fashion, and work to maintain regular meetings and updates with the client.
- The visuals of the site came together very well. We were able to put together a set color scheme and style guide for the site, which led everything to be very cohesive and meld together nicely. In the future, we know that creating set style guidelines in the beginning of the project will lead to a very pretty and polished looking site.

What Did Not Work

- Using a pre-built transaction service ended up being far more complex than anticipated. While the benefits of using the pre-built service, like the extra transaction security, still remain, the fact that our platform needed to exchange money directly between users without anyone else ever holding the money makes using the service more complex. Instead of being able to use the out-of-box plug-ins, we had to edit the way the service was integrated, and it resulted in more difficulties than anticipated. In the future, we know to do more research into how transactions are handled through the different pre-built services and find one that better fits our intended implementation.
- Our original shopping platform design implemented wonderfully, but ultimately failed to function as our site needed to. Because of the unique requirements of our site, we realized that users should only be purchasing from one distillery at a time, and the site needed to limit users accordingly. This meant that we had to refactor our whole shop/cart page to incorporate these limitations. In the future, we know to analyze the project further ahead of time and make sure all user limitations are considered when planning the website design.
- Initially, we planned to create a smoother and more automated process to handle the paperwork needed to make the transactions, and to make the necessary changes to the liquor labels. Ultimately though, the legal aspects of this were beyond our expertise, and we didn't have the time or knowledge to implement this how we were anticipating it. In the future, we know to limit the scope of the project more, and try to get direct consultation from more experts in this field.
- Our product testing got repeatedly delayed, and ended up not being as through as we were anticipating. Our initial contacts for testing ended up going out of town, and we struggled to find new testers for some time. Even when we got new people to test it, the final connections for the project took longer than scheduled, so the testers were somewhat rushed in giving their feedback. In the future, we know to secure testers earlier on in the projects and ensure that the product is published in time to get proper feedback.

XII. Acknowledgments

The Method and Muse Distilleries Team would like to give special thanks to Aubrey Wigner for giving us this incredible opportunity to work on this project, and to Iris Bahar for helping guide us through this project.

XIII. Team Profile

Jessica Canaday
Senior
Computer Science
Hometown: Lone Tree, Colorado

Work Experience: Substitute Teacher
Clubs: Oaks International
BuzzFeed What Cocktail Am I?: Old Fashioned
I am excited to create a better system for small distilleries to sell each other's products, helping them to promote each other's businesses.

Jordan Barker
Senior
Computer Science
Hometown: Metamora, Illinois
Work Experience: Customer Service
Clubs: Mines Little Theater, Sigma Kappa
BuzzFeed What Cocktail Am I?: Margarita
I'm excited to learn more about secure websites and integrating legal aspects into it while helping businesses!

Erin Harl
Senior
Computer Science
Hometown: Corpus Christi, Texas
Work Experience: Internships at Onyx Engineering, CC Electric Co., and several restaurants throughout the years
Clubs: NeST and the Astronomy Club
BuzzFeed What Cocktail Am I?: Margarita
I'm excited to get to build a product from the ground up, and help create a website that will have a positive impact on the distilling landscape.

Aajish Marahatta
Senior
Computer Science
Hometown: Lakewood, Colorado
Work Experience: Internships at Janus Henderson Investors and Xcel Energy
Clubs: South Asian Student Association
BuzzFeed What Cocktail Am I?: Margarita
I am excited to be a part of this project and create a functional e-commerce website that will end up being very useful for distilleries to sell their products.

References

Appendix A – Key Terms

Term	Definition
DSP	*Distilled Spirit Plant*
ABV	*Alcohol By Volume*
TTB	*Alcohol and Tobacco Tax and Trade Bureau*

Transfer in bond	*Shipment of untaxed alcohol (like bulk spirits or wine) between two government-approved, bonded manufacturing facilities*