VCU College of Engineering

CMSC 326 and Agent Replay
Project Report

Prepared for

Derrick Murry/Lily Roark

Capital One

By

Del Huband, Brian Vo, Alex Nguyen, Parker Gonzaga

Under the supervision of

Rachita Sowle

5/2/2025

Executive Summary

Capital One's customer service system Empath has frequent bugs that hinder agents attempting to help customers. The Empath developers currently have a difficult time debugging these issues, as they have to manually read and decipher relevant clickstream data relating to the error. In order to solve this problem, our team will develop a program that can replay an agent's journey while they are on the site so that the developers can easily reproduce bugs that agents may have encountered. This replay could also be used to assist in training new agents, improve agent workflows, or generating automated tests on Empath to test the service's performance or features. However, this program cannot reveal any customer information as to not leak any sensitive information while replaying an agent's journey. It must be scalable to handle multiple agents' sessions, while still maintaining a small file size so it can be handled easier.

To produce this project, we'll be hosting a personal website on Azure that will have clickstream events attached to it. We'll read these clickstream events from the website using Google Analytics, and these files will be saved onto Google's BigQuery cloud storage so that they can be easily accessed and replayed. Now that the clickstream data is saved, it will be plugged into Playwright's browser playback service, which will show an the exact events the user took during their session in real time, allowing others to view how a user interacted with the website, and potentially find the same issues and bugs that the user encountered. Our solution also allows for API calls to be mocked with partial sanitized data from the original session being able to be played during the playback.

Table of Contents

Section A. Problem Statement

Capital One is a bank holding company. They provide both in-person and online services. When it comes to online services, customers occasionally encounter issues with their bank accounts, or want to manage their bank accounts. To do this, they contact customer support.

Capital One uses an online service called Empath to manage customer accounts during customer support. Empath is used by customer service agents to assist any customers who need help with their account, such as being locked out of it or needing to change data. The issue arises that any agent using Empath could potentially encounter bugs. This problem occurs semi-regularly, depending on how stable the current version of Empath is. When these issues arise, Empath's developers use clickstream data to track down what caused the issue. This process is time consuming, leading to slower bug fixes and update cycles being deployed for Empath. This worsens customer interactions and hinders employee productivity. Our solution should reduce the amount of time it takes to locate encountered issues, thus improving customer and agent experiences.

Empath deals with sensitive customer details as well as proprietary Capital One information, so using a publicly available solution to solve this problem directly is not the preferred approach. Additionally, Capital One agents do not have the sufficient privileges to use these solutions directly, and the solutions currently available to developers are too time-costly.

The project's client is Capital One. Stakeholders in this project include Capital One agents, developers, training teams, customers, and shareholders. The first three groups can use our project directly, and the time-saving improvements our project produces will affect the experiences of customers and shareholders.

The problem has some associated costs. When agents face a bug that cannot be replicated it can hurt both the customer and the company economically. When an agent isn't able to solve a problem it leads to a loss of productivity for the agent. And for customers if they have an urgent issue that they want to be solved, bugs can delay their business functions, such as a payment.

This project falls under the umbrella of user-interaction tracking, data analytics, and User Experience (UX) research. While leveraging Google Analytics, Heap, Amplitude, or another website analytic tool, a more comprehensive view of user behavior can be generated. This will enhance the accuracy and depth of clickstream analysis for this client, advancing the way that an agent's journey data is tracked, stored, and replayed for optimization.

In 2018, a team composed of Timur Bekmambetov, Guy Bedford, and Daniel Kehoe developed their own service which would playback a user's browser session (Bekmambetov et al., 2018). Their system allowed for users to play and rewind the journey a user took while browsing one or multiple websites using JSON data that tracked the mouse cursor and mouse events, and it could track the user's sessions through multiple websites (Bekmambetov et al., 2018). This JSON data was captured via a program that has to be run by the user to record any data (Bekmambetov et al., 2018). However, this software was only developed to be used locally, meaning that all computers owned by Capital One would have to have this external software installed (Bekmambetov et al., 2018). It also has no way of mocking client data, and it doesn't automatically record all of a user's sessions on a specific website, meaning that some potentially insightful user sessions could get lost if Capital One were to implement this solution.

One previous solution attempt implemented by Capital One was Glassbox, a software that would record agent actions in videos that could be played back (Glassbox). These screen recordings were fed through a service to censor sensitive data, and then used by developers to see what led up to encountered issues (Glassbox). Some problems with this approach included large storage size for the videos, potential mistakes releasing sensitive customer data, and videos not capturing backend data (Glassbox). Our approach should analyze clickstream data in order to avoid analysis of sensitive data and to illustrate what is happening in the backend.

Another software that is similar is Crazy Egg. It has a lot of features that compared to Glassbox that can be viewed as better. Crazy Egg is more oriented towards providing visual tools, such as heatmaps or snapshots, however it has limits when it comes to the amount of recordings it can produce, which is limited at 10,000 recordings (Crazy Egg). This makes it so that any system implementing Crazy Egg is limited in how large it can scale its operations (Crazy Egg). This software also requires that all data is stored and handled by Crazy Egg, making it difficult for Capital One to organize and manage data, as well as risks the security of the data by not being able to handle it on their own servers (Crazy Egg). This can also be very costly when exceeding the limit and losing a customer's personal information.

Lastly, Hotjar is another tool that, like the other softwares Crazy Egg and Glassbox, has heatmaps and a session recording (Contentsquare). Hotjar also allows for mouse tracking, which would allow for more data to be gleaned, which could assist in web design and workflow (Contentsquare). To save storage space, Crazy Egg and Hotjar delete recorded sessions after a period of time (Contentsquare). This may pose a significant issue because if a bug is ongoing and is eventually investigated the recording may be removed.

Our tool aims to solve the issue of space efficiency, by storing clickstream data rather than having recordings which take up a significant amount of space. Our tool also won't require third party management, allowing us to handle the data internally. This way we ensure that we can control when we delete the data or know how it's used exactly. This will also allow us to manage the data and not risk the third party compromising our data.

To also allow for a better agent playback, our tool should be able to mock API data and replicate certain data requested from the agent in the playback. However, this mocked API call should sanitize any potential identifying information such as name or social security number from the call, but return the rest of the relevant information where applicable.

In conclusion, our iteration of this technology will be designed to capture a user’s browser session while still maintaining the security of any personal or sensitive information captured during the session, ensuring that the data is stored using minimal storage space, and being able to store all user sessions on Empath. We’ll store our data on a cloud storage system to allow for it to be highly scalable, and it won’t require any external software installation to obtain a user’s session from the browser, as it will all be stored via the website’s own clickstream data. Our final project will accurately portray a user’s journey through a website, while adhering to the requirements Capital One has set out.

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

Capital One's agents that use Empath frequently encounter issues that hinder their ability to help customers. The debugging process to remove these bugs is extensive as it takes a significant amount of time and effort to determine the bug's source, as there's currently no way to easily reproduce these bugs. To help replicate issues and debug the service Capital One needs a service that will:

- Improve the process of tracking and analyzing user interactions in a digital environment, enabling more detailed insights into agents' behaviors.
- Create a streamlined replay service that allows businesses to easily visualize agents' journeys.
- Establish a more scalable and efficient system for storing/replaying the recorded data (cloud services).
- Reduce the time it takes to locate encountered issues in Empath.

B.2 Design Objectives

Our team will attempt to solve Capital One's problem by taking clickstream data from Empath and converting that into a playback for developers to debug the service with. Our team's design will:

- This design will provide a detailed replay feature, allowing users to visually track every click and interaction in the agents' journey.
- The design will replay an agent's journey after at least 1 minute.
- The design will be achievable given the resources: Heap, Amplitude, and Google Analytics, and a $1000 budget.
- The design will be realistic in terms of system scalability, allowing it to handle large amounts of data without significant performance degradation.
- The design will be completed in the timeframe of one academic school year.

B.3 Design Specifications and Constraints

Design Constraints

The design of our agent replay system must will meet its objectives by adhering to the following constraints:

- The design has to adhere to the $1000 budget for development costs and cloud services, including the potential costs for Heap, Amplitude, and Google Analytics.
- The design has to be implemented into a cloud service to allow for easier data management and scalability.
- The design has to track a user's clickstream data and store it for future access.
- The design must follow a project timeline that allows for full testing and completion within one academic year (9 months), with definable milestones.

Realistic Constraints

To ensure our product will be productive to those who use it, and to ensure its longevity, it will have the following realistic constraints:

- The design has to be seamlessly integrated with the platforms that the client is using without needing any significant infrastructure changes
- The design has to remain cost effective to match the allotted budget.
- The design must comply with data privacy regulations to ensure sensitive data is stored securely
- The design must be intuitive enough that non-technical personnel could operate it easily without any training or support.
- The design must reconstruct a journey only from collected clickstream data
- Ensure that it is scalable, with possibly many users accessing the database we want to make sure it still functions with a high load of users

B.4 Codes and Standards

To ensure our project abides by our above requirements and rules of design, we'll abide by several standards and codes while developing our project. These standards and codes will be:

Standards

- ISO/IEC 27001: Information Security Management Systems (ISMS)
    - Our design will ensure that the system handling replay data maintains confidentiality and integrity of sensitive user data.
- IEEE 829-2008: Standard for Software and System Test Documentation
    - The design will undergo testing. This will allow for documentation of the test results, allowing for future improvements and maintenance.
- ISO 9241-210: Ergonomics of Human-System Interaction
    - The design will be intuitive, user-friendly, and accessible, reducing user confusion and improving the overall user experience without need for extensive training/support to operate.
- IEEE 7000-2021
    - This design should have consideration of individual and ethical values. Such as, privacy, fairness, and accountability of user's data.

Codes

- FTC Act: Section 5 - Unfair or Deceptive Acts or Practices
    - The design must be transparent in its data collection and usage to avoid deceptive practices.
- ECPA (Electronic Communications Privacy Act)
    - Since the design involves tracking communications between the agent and customer, it must comply with ECPA to ensure privacy in electronic communications.
- GLBA(Gramm-Leach-Bliley Act)
    - Requires financial institutions to explain their information sharing practices to their customers and to safeguard their data

Section C. Scope of Work

C.1 Deliverables

Sponsor Deliverables:

- Functioning Prototype of Agent Replay
- Playwright program capturing user clickstream data to "replay" their path through a website.
- Technical and User Documentation detailing how it works/how to use
- Fall poster

Academic Deliverables:

Team Contract	(Sep. 6)
Project Proposal	(Oct. 11)
Fall Design Poster	(Nov. 15)
Preliminary Report	(Dec. 9)
Capstone Poster and Abstract	(March 28)
Final Design Report	(May 2)

Obstacles:

- Google analytic clickstream may not be usable at first we may have to convert the software we use to get clickstream data.
- Playwright may not provide the right solution maybe it does live calls, which is what we don't want for security reasons
- Open source website doesn't allow for clickstream data to be collected
- Converting clickstream data to playwright may cause issues

C.2 Milestones

Milestone	Date/Time
Presentation Completion	Oct 3, 2024
Proposal Completion	October 11, 2024
Add Clickstream Events To Website	November 1, 2024
Store And Read Clickstream Data	November 15, 2024
Implementation of Click Stream Data Into Playwright	December 6, 2024
Functioning Prototype	Jan 13, 2025
Create Tests to Verify Software Behavior	Jan 27, 2025

Creation of User Documentation	Feb 10, 2025
Internal Testing/Review	March 3, 2025
Refinement	April 2, 2025
Capstone EXPO	April 26, 2025
Final Report	May 2, 2025

C.3 Resources

Our team’s project’s resources will be composed mostly of open-source or free software that we’ll need to implement into our design, with a budget that we may use if the need arises. The resources required for this project would be:

<u>Software</u>

- IDEs for developing (VS Code)
- Playwright for Frontend Automation
- Version Control system(Github)
- Google Analytics API
- Google BigQuery
- Google Tag Manager
- Mock data generation tools (Tweak)
- API Hosting (Render)
- Website Hosting (Azure)

<u>Budget</u>

- $1000

Section D. Concept Generation

One concept our team came up with was a simple program that would record the screen of the agent using Empath, allowing for future users to replay the video as a form of playback. This method would have to have some form of real time censorship software implemented into the playback so that confidential and valuable information wouldn't be shown. The pros of this concept would be that it'd be easy to develop and would allow for a simple playback, as the users would just have to run the video file to play it. However, the cons of this concept would be the potential revealing of confidential information, such as a customer's finances or banking details, due to potential errors in the censorship program, or a change in Empath that would require changes to be manually made to the program. This would also mean that the service would have to be tailor-made to each potential version of the website. The video files would also take up a substantial amount of storage space, meaning that the amount of stored past user sessions would likely be limited, or expensive to maintain.

Another concept we decided on was taking and storing the clickstream data from a user's session on the website and replaying that clickstream data via a playback service, replaying the user's entire session in real time. We would gather the clickstream data by deploying Google Analytics on a test website, and then we'd take that data and plug the clickstream information into Playwright, which we'll configure to then playback the user's session.This would have the benefit of being light on storage size, modular in such a way that one version of the software would work for all versions of Empath, and could also appropriately censor critical user data, as this program wouldn't need to actually store that data in order to function and replay a session. Potential flaws in this program would be that it would have a more complex development process, and there could be issues that occur when trying to combine the external software we'll be implementing. Also, as this design would have to test through the actual website, it would be making actual calls through the website, which could lead to some data reappearing on the clickstream data. It could also possibly conflict with other metrics that the company takes on the website through these calls.

The last concept we created was taking the clickstream data from the website and playing it back in a mockup of the website. This option, while similar to the one mentioned previously, would instead of running the playback through the website via Playwright, it would instead use its own custom version of the website that it uses to playback the data. This would have the added benefit of being able to ensure that no sensitive data is accidentally recorded as we have more control over the website and what's being sent through. This added control would also allow us to test a wider range of features that Empath has, and overall more accurately simulate what the user did during their session. It would also not interfere with Empath's base analytics, meaning that using this design to playback user sessions would have no possibility of disrupting Empath's data. The downsides of this design would be that it would require the largest amount of work to design and maintain this synthetic version of the website. This design would also require the team to constantly make revised versions of the synthetic site if there are ever any changes to Empath, which would significantly decrease the usefulness of this tool.

Section E. Concept Evaluation and Selection

In order to evaluate our concept designs, we will work off the criteria of the design's scalability, cost, effort, reliability, and risk. Scalability and risk will be weighted the most in terms of importance, with effort being rated above cost and reliability due to the small time-frame given by the project. Each criteria will be rated out of ten, and will then be totaled to evaluate which design we will choose to develop this project.

We will determine the scoring of each criteria via separate metrics on a scale of 1-10. The metric of scalability will be how easily modifiable the playback process is to changes in the website, with a 10 being the program would need no future changes, and a 0 being it would need constant changes to match the website. The cost criteria will be measured by how much money would be required to develop and upkeep the project, with 10 being no finances needed and a 0 being a large upkeep fee being required. The effort required will be measured by how much time and research is needed to develop a design, with a 10 being little time and research and a 0 being a significant amount. Reliability will be determined by how stable the design is and how it's able to maintain itself between updates to Empath, with a 10 being it would be completely stable at all times and a 0 meaning it would require a team to constantly maintain it at all times. Finally, risk will be measured in how likely the program is to reveal confidential information, with a 10 being the design will never show confidential information, and a 0 meaning it would constantly be under threat of revealing data it's not meant to.

Criteria	**Design #1**	**Design #2**	**Design #3**
Scalability (25%)	9	10	3
Cost (15%)	4	9	6
Effort (20%)	8	7	2
Reliability (15%)	10	9	10
Risk (25%)	6	8	10
Total (100%)	7.45	8.6	6.05

Based on the results from this chart, we will be going with the second design. The first design would be too risky in terms of revealing sensitive company information, and the cost of maintaining servers capable of holding the amount of video files necessary would prove to be very expensive. The third design would be the most secure, but would take a significant amount of time to develop, and it would need constant maintenance to ensure the synthetic website is updated to the most recent version of Empath. The second design has a low cost needed, and would still be secure in holding company data and preventing leaks, meaning it has the best potential outcome for this project.

Section F. Design Methodology

Evaluation and Improvement Methods through the Iterative Design Process

This design process followed an iterative approach, with continuous feedback from internal testing and weekly sponsor meetings to guide the system's evolution. Each iteration allowed the design to be reviewed, adjusted, and improved based on technical specifications and client needs, ensuring that the system closely mimicked their existing implementation.

Internal Testing

After each iteration, the system will undergo **internal testing** to identify areas for improvement. This will include:

1. **Website Functionality:** Testing updates made to the website to ensure that the clickstream data is correctly captured and replayed, and that all features are functioning as expected.
2. **Clickstream Data Accuracy:** Ensuring the system properly gathers clickstream data and accurately replays the recorded interactions according to the provided JSON.
3. **UI and Host Improvements:** Investigating potential improvements to the website's design and its hosting environment to enhance the user interface, aiming to improve mimicry of the client's existing system.

These internal tests will ensure that the system meets the required specifications and helps identify areas for improvement before the next development phase.

Weekly Sponsor Feedback

To ensure the design is progressing in line with the client's needs, we hold **weekly meetings** with the sponsor. During these sessions, we:

1. **Present Progress**: Share updates on system functionality, including new features and improvements made based on prior feedback.
2. **Solicit Feedback**: Gather input from the sponsor on how well the system aligns with their expectations and business requirements. This feedback will be essential in refining the design and ensuring it meets the client's evolving needs.
3. **Adjust Priorities**: Based on sponsor feedback, we shift our focus or make changes to better meet their needs and project goals.
4. **Ask Clarifying Questions**: We also ask questions to better understand how the system can be improved and gather insights on specific aspects of their system. This helps guide the development of future features and ideas.

By incorporating both internal testing and sponsor feedback into the iterative process, we ensure continuous improvement and alignment with the client's goals and requirements.

F.1 Experimental Methods

Mock Data Creation and Testing

To verify that the system works as expected, we **created mock JSON data** that simulates user interactions, focusing on the essential information needed to test Playwright's functionality. This mock data does not follow the exact format of Google Analytics but was designed to capture the critical interactions for Playwright to process correctly.

- **Data Creation**: We created a simplified JSON structure containing only the essential information, such as user interactions (button clicks, entering information), without the full complexity of the Google Analytics format. This allows us to focus on the key data that Playwright needs for interaction replay.
- **Testing with Mock Data**: The mock data was then used in testing the system, ensuring that Playwright correctly captures and replays the user interactions as defined in the mock JSON.
- **Data Accuracy**: The mock data was verified against expected behavior to ensure that the interactions were recorded and replayed accurately, confirming that Playwright can handle the data correctly and that the system works as intended.

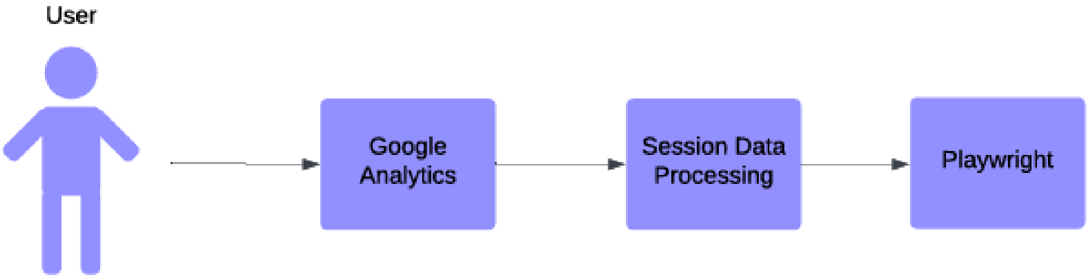
Experimental Testing with Realistic Scenarios

After validating the system with mock data, **realistic scenario testing was performed** to ensure that the current implementation of the system works as expected in practical use cases.

- **Test Environment Setup**: Team members manually interacted with the website, performing realistic actions such as button clicks, menu navigation, and form submissions to generate clickstream data.
- **Data Logging**: During these interactions, clickstream data was logged and analyzed to verify that the system correctly recorded the actions, including button clicks and navigation events, and replayed them without errors.
- **Performance Evaluation**: The system's ability to handle and replay complex user sessions was confirmed, demonstrating readiness for real-world use, including future API integrations or mobile adaptations.

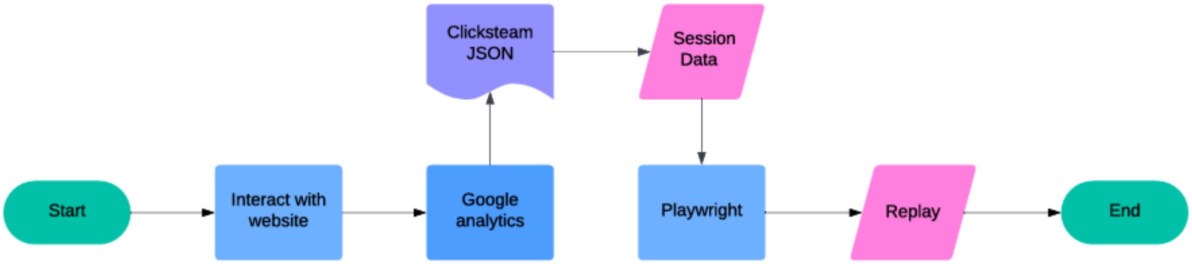
F.2 Architecture/High-level Design

1.High level architecture



a.

2.Process Data Diagram



a.

F.3 Validation Procedure

With the system completed, the design team validated the final tool during a live demo with the sponsor at the Capital One campus prior to the Capstone Expo, as well as a weekly zoom meeting with the sponsors.

- **Sponsor Demonstration:** We presented the working prototype, showcasing system functionality, session playback accuracy, and alignment with the sponsor's technical requirements.
- **Client Feedback**: During the demo, feedback was gathered from the sponsors, staff/stakeholders, highlighting strengths and identifying minor improvements, as well as suggestions for features.
- **Final Adjustments**: Based on the feedback, adjustments were made to fine-tune the system, ensuring it meets the sponsor's requirements and is ready for final delivery and presentation at the Capstone Expo.

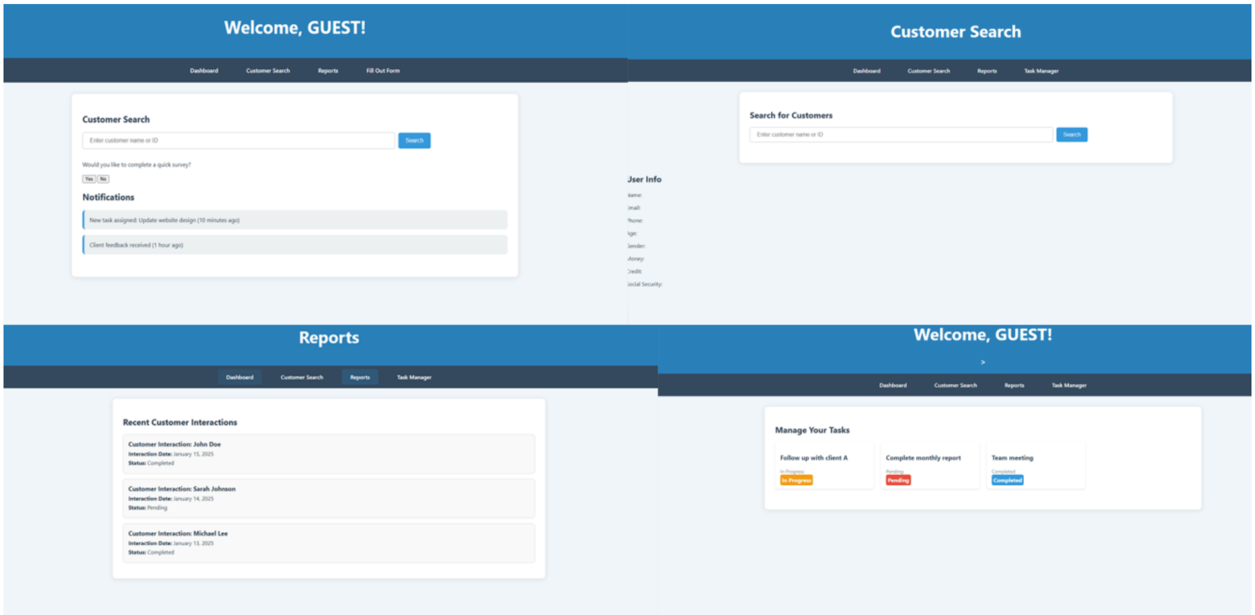
Section G. Results and Design Details

Preliminary Results

Website

Our preliminary website contains the following features, all of which act as potential sources of trackable clickstream data:

- Multiple pages
- Text entry field
- Inline Form – When users click "Yes" to a prompt, a form appears directly on the page. It includes:
  - A multiple-choice question (single selection)
  - A checkbox list (multiple selections)



Would you like to complete a quick survey?

New task assigned: Update website design (10 minutes ago)

Client feedback received (1 hour ago)

lame:
imail:
Phone:
Age:
Gender:
Money:
Credit:
Social Security:

Customer Interaction: John Doe
Interaction Date: January 15, 2025
Status: Completed

Customer Interaction: Sarah Johnson
Interaction Date: January 14, 2025
Status: Pending

Customer Interaction: Michael Lee
Interaction Date: January 13, 2025
Status: Completed

Follow up with client A
In Progress
In Progress

Complete monthly report
Pending
Pending

Team meeting
Completed
Completed

Four pages from the prototype website.

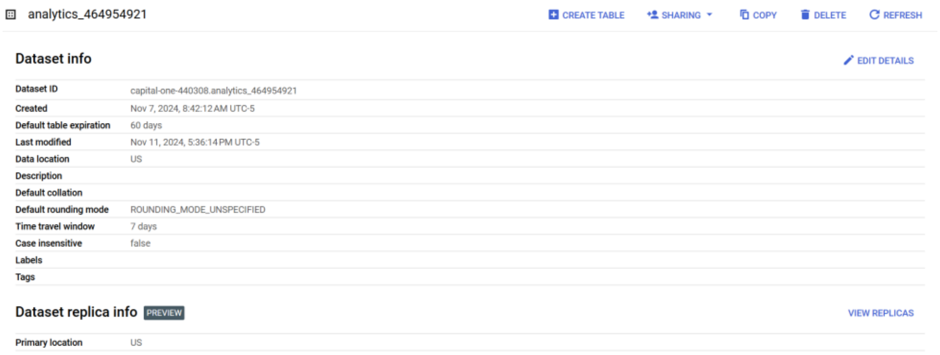
Initially, we considered using a premade open-source website as our main source of clickstream data. This approach would reduce the time needed to build a website from scratch, allowing us to gather data quickly and start working on the Playwright script sooner. Our first step was to decide where to host the site. We chose the cheapest option, GitHub Pages. However, GitHub Pages only supports a single index.html file, which limited what we could add and made the website structure messy. Due to these limitations, we pivoted to Azure, which allowed us to create dynamic pages and keep our website structure more organized.

Our current website has four pages. Three of them generate useful clickstream events like button_clicked or keypressed. These events are tracked using a Google Analytics script added to each page. The three key pages are: Dashboard, Search, and Fill Out Form. The Search page is the most complex; it includes an API call to a Render-hosted database that retrieves a user. In our Playwright script, this call is mocked as a test API call to avoid real data access and ensure security.

The website is now dynamic, meaning we can pull user information after making an API call. Playwright is fully integrated into our clickstream tracking, though there are still some caveats. For example, in manual mode, where we use ‘A’ and ‘D’ keys to move forward and backward, the backward logic isn't fully functional in some cases. Future updates could improve this feature or add support for mouse movements.

Clickstream

Our concept relies on Google Analytics to gather data from the test website. Google Analytics tracks user clicks and interactions with the website throughout the day, and then updates the dataset with new data at the end of each day. After capturing the data with Google Analytics, we then store the data into the cloud which can be accessed in our program now.



analytics_464954921

Dataset info

Dataset ID	capital-one-440308.analytics_464954921
Created	Nov 7, 2024, 8:42:12 AM UTC-5
Default table expiration	60 days
Last modified	Nov 11, 2024, 5:36:14 PM UTC-5
Data location	US
Description	
Default collation	
Default rounding mode	ROUNDING_MODE_UNSPECIFIED
Time travel window	7 days
Case insensitive	false
Labels	
Tags	

Dataset replica info

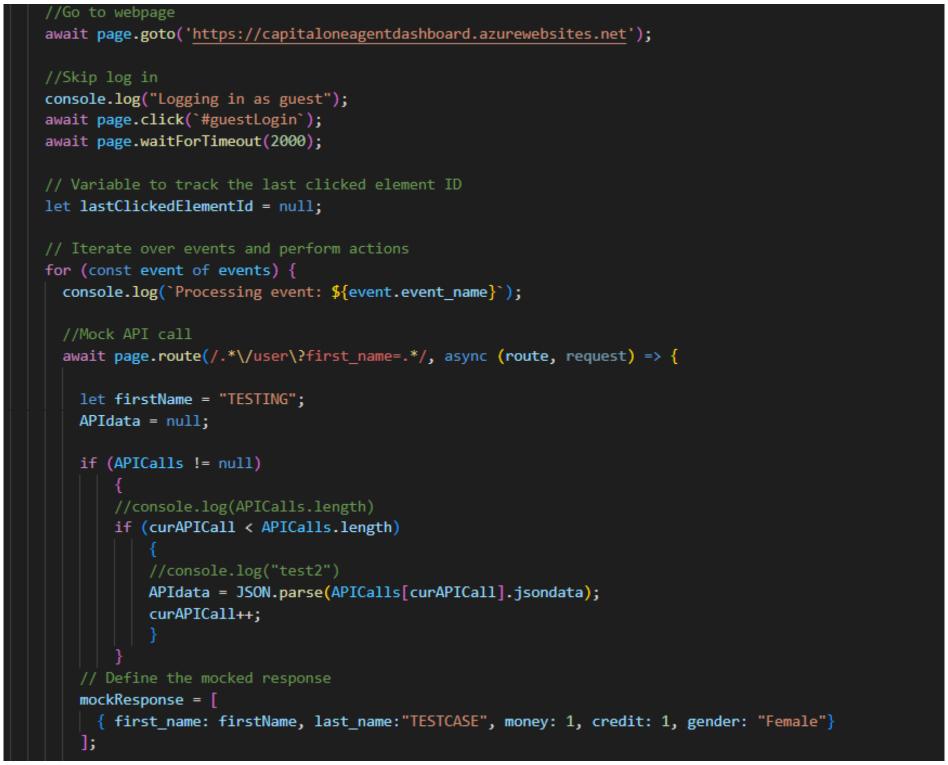
Primary location	US

Google BigQuery dashboard for our analytics dataset.

Replay Program

We’ve configured a Playwright program to parse the JSON data collected from Google Analytics and retrieved through Google BigQuery, and replay the user’s actions on a copy of the test website. The program follows the clickstream data, accurately reproducing each user interaction as outlined in the JSON.

The replay supports two modes: an automated mode, which runs through the session without any input, and a step-by-step mode, where actions are performed manually by pressing a key. Both modes process the JSON data to simulate the user’s path through the site, and the replay functions accurately in its current state.



```
//Go to webpage
await page.goto('https://capitaloneagentdashboard.azurewebsites.net');

//Skip log in
console.log("Logging in as guest");
await page.click(`#guestLogin`);
await page.waitForTimeout(2000);

// Variable to track the last clicked element ID
let lastClickedElementId = null;

// Iterate over events and perform actions
for (const event of events) {
  console.log(`Processing event: ${event.event_name}`);

  //Mock API call
  await page.route(/.*\/user\?first_name=.*/, async (route, request) => {

    let firstName = "TESTING";
    APIdata = null;

    if (APICalls != null)
        {
        //console.log(APICalls.length)
        if (curAPICall < APICalls.length)
            {
            //console.log("test2")
            APIdata = JSON.parse(APICalls[curAPICall].jsondata);
            curAPICall++;
            }
        }
    // Define the mocked response
    mockResponse = [
      { first_name: firstName, last_name:"TESTCASE", money: 1, credit: 1, gender: "Female"}
    ];
```

A code snippet from the playback program.

While we have still yet to add a user interface and the ability to manually adjust replays, our prototype acts as a proof of concept for a replay system. Our prototype:

- Automatically follows a path through the website
- Functions solely on json data straight from the database

Although there is room to add additional features and improve the user's experience, the basic replay functionality has been achieved. With some slight adjustments and a more complex cloud-based website, our final version of the program should provide users a robust and easily accessible playback tool.

Currently our user interface is the command line, we attempted to have a GUI but with the time allotted it wasn't feasible. But our prototype:

- Automatically follows a path through the website
- Manually follows a path through the website
- Masking/Mocking API calls
- Functions solely on json data pulled from the cloud

There are several features we could add in the future, such as tracking mouse movements to improve user behavior analysis. We could also enhance the website by adding more complex structures or improving our clickstream gathering tool, as there is currently a delay of about a day before the data appears in the database. Overall, our program provides users with a working clickstream experience that can be used for learning and analysis.

Section H. Societal Impacts of Design

H.1 Public Health, Safety, and Welfare

A key feature of our design is the avoidance of sensitive data. Our design tracks API calls, which could potentially contain sensitive user data. To circumvent risk of viewing sensitive user data, these API calls can be sanitized in order to remove any pertinent information. Additionally, as our design is a prototype, the API calls it tracks do not necessarily reflect the framework CapitalOne uses internally. Thus, if our design were to be used by the company, data sanitization or other means of circumventing use of sensitive customer data can be implemented.

Aside from API calls, our design's replay feature intentionally avoids caching and viewing of any sensitive user data. The clickstream data we collect is void of any personal information, and only provides information about where and what the Agent clicked throughout their journey. The names of the clicked objects are genericized and do not contain any personal information. While there might be risk of viewing potentially sensitive user data through a replay system, our project's replay system is designed in such a way to avoid displaying that information entirely. By viewing strictly clickstream data, our design circumvents any interaction with sensitive user data.

Another planned step of security we plan to take is to forward our code to our Capital One representatives. Throughout the development process we have maintained frequent contact with the company, and review any changes we make to the design with them before implementation. Upon submission of the completed project, the design should be in such a state that there are few privacy and security concerns. Additionally, if this code is to be used to debug the system ISO/IEC 27001: Information Security Management Systems (ISMS) requires that management examine information security risks, design and implement information security controls and/or other forms of risk treatment, and adopt a management process to ensure that the information security controls continue to meet the organization's information security needs. Capital One must comply with these regulations when using our program, which adds an extra layer of assurance for customers.

H.2 Societal Impacts

Agent Journey Replay should be relatively low-impact, as its primary use case is as a backend development tool to help debug a customer service web application. This code could be adapted to replay user paths on other websites, which may impact users as the information could be used to track user habits during interactions.

H.3 Political/Regulatory Impacts

The design's small scale should cause little to no political/regulatory impacts. If user data was not properly secured, then regulatory adjustments might be justified to protect user data, however compliance with any relevant codes and standards should ensure fulfillment of legal privacy requirements. Additionally, our design does not collect any information beyond that which is already being collected internally by CapitalOne, so implementation of our design should not require changes to existing regulatory legislature.

H.4. Economic Impacts

Our design intends to reduce the length of the Empath debugging process, which should reduce site downtime and allow for enhanced customer support. While Empath does not directly generate revenue for the company, its use contributes to and improves the customer experience. This could lead to greater customer satisfaction, indirectly impacting Capital One’s profitability by increasing user retention.

H.5 Environmental Impacts

Our design has little to no environmental impact. As a debugging tool, it should be fairly lightweight and non-resource intensive. Thus, concerns over energy use and potential environmental damage are minimal. It may increase the uptime of the Empath customer management system, which could increase energy usage if the operation of that service is particularly energy-intensive. It also enables usage of Capital One’s other services, such as their banking system, which could also increase energy usage due to the potential increase in user retention through use of our design. Overall, however, there should be little to no impact through direct use of the Agent Journey Replay system.

H.6 Global Impacts

Our design may improve the experience of individual customers globally, however it is unlikely to have any far-reaching impacts on a global scale. Other services already enable user tracking on the web, so our design is unlikely to introduce any novel dangers or techniques that may affect people globally.

H.7. Ethical Considerations

Capital One customers might be uncomfortable with a system that tracks actions taken through a system containing their sensitive personal data. This is a valid concern. Our design is used strictly as a debugging tool for Empath; Capital One already tracks clickstream data for the platform. We are using data that is already available to backend developers, and the design cannot access any data beyond what is directly provided to it. Our design abides by the IEEE 7000-2021 by considering ethical values such as privacy, fairness, and accountability of user’s data. The design does not store any sensitive information, and will avoid deceptive acts or practices outlined in the FTC Act: Section 5 - Unfair or Deceptive Acts or Practices. The process of accessing backend calls and tracking clickstream data is already an action Capital One developers take to debug the platform, our design just presents this information in a format that better suits debugging practices. Despite this, users should be notified of any additional privacy risks introduced in the future.

Section I. Cost Analysis

Provide a simple cost analysis of the project that includes a list of all expenditures related to the project. If an experimental test set-up or prototype was developed, provide a Bill of Materials that includes part numbers, vendor names, unit costs, quantity, total costs, delivery times, dates received, etc. Do not forget to include all manufacturing costs incurred throughout the completion of the project. If the design is expected to become a commercial product, provide a production cost estimate including fixed capital, raw materials, manufacturing (including tooling and/or casting), and labor costs to produce and package the device. Note that this type of detailed cost analysis may be listed as a project deliverable.

Note: The Preliminary Design Report should include all costs incurred to date. It is expected that this section will be expanded and updated between the preliminary and final design reports.

Costs Incurred

We did not incur any financial expenditures related to the project during its design and development. To avoid expenses, we used the free tier of the following services:

- Google Analytics
- Microsoft Azure
- Google BigQuery
- Google Tag Manager

While these services provide paid tiers with additional functionality, we were able to complete development of all deliverables for free.

Enterprise Usage

If Capital One decides to use this project on a larger scale, then their usage might incur some costs.

Conversations with our sponsors indicate that Capital One already has some method of gathering clickstream data aside from Google Analytics. This project will not increase the amount of data that must be collected, however it will encourage the continued use of their current gathering tools which will have a related cost.

Additionally, integration of our design with Capital One’s existing systems will require development time. Depending on the amount of development needed, this could potentially require the hiring of additional developers in addition to the existing cost associated with paying current employees.

Outside of the usage of this existing tool, there should be no other notable costs related to the functionality of our project.

Section J. Conclusions and Recommendations

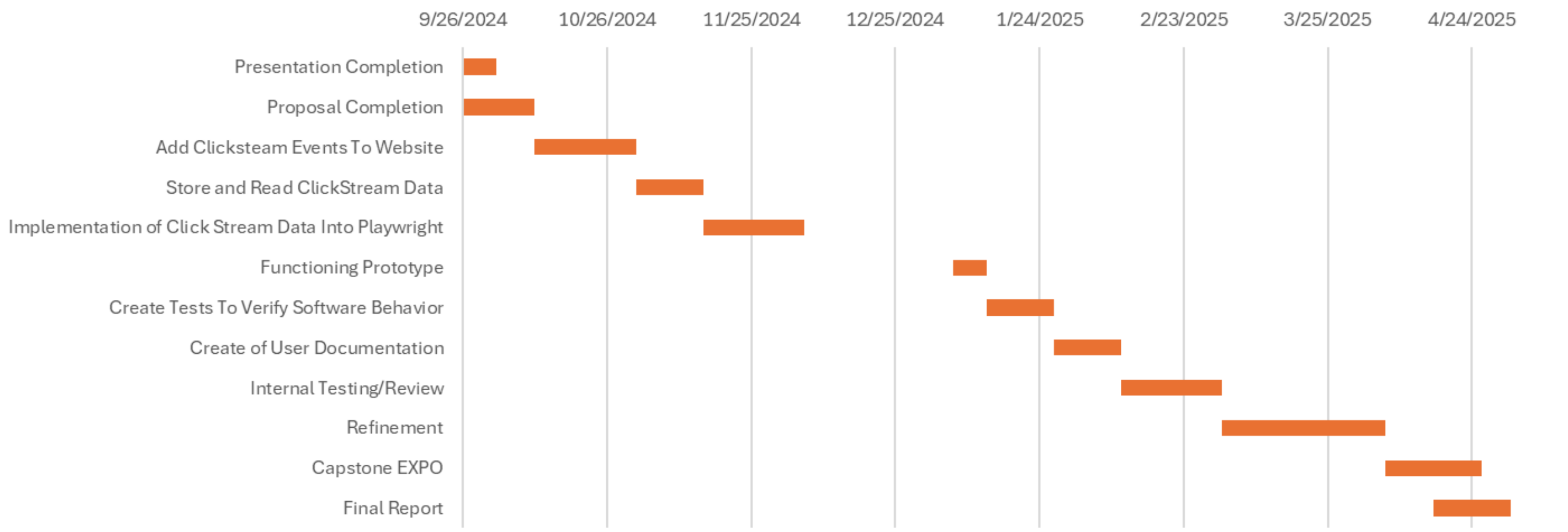
In conclusion, our team will be developing a program that can take Google Analytics clickstream data from a specific user session and plug that data into Playwright, a browser playback service that will replay that user’s session in real time. This tool will be used to test the design, usefulness, and errors of the Capital One agent service, Empath. We originally developed several other means of approach, one of which was utilizing a screen recording software that would censor critical information in real time using software, and another was running a mock version of the website that would run all the clickstream data on in real time.

Our team thinks that, while these designs could be utilized, both have several flaws that make them less than ideal. The screen recording design would require a large amount of storage space to save many video files, many of which could be hours long. The censorship program could also be faulty and not work all the time, and it could also become outdated if Empath changed its design substantially. The mock website would require a significant amount of time to develop and maintain, and would require constant maintenance to match the current version of Empath at all times, which would significantly reduce the usefulness of the tool.

We believe that this method is superior to other designs our team created because it will be compatible with Empath even if it undergoes severe changes, and will help ensure that critical information isn’t shown during the playback via wiping any potential critical information and mocking it. This design also wouldn’t need a substantial amount of storage space that other designs would, allowing it to be more versatile and cost effective.By utilizing this method, we can playback the user's session in the most efficient and safe way, while still being able to develop the project in a timely manner.

Appendix 1: Project Timeline

Provide a Gantt chart of similarly composed visual timeline showing the start and end dates of all completed tasks and how they are grouped together, overlapped, and linked together. Include all senior design requirements including design reports and Expo materials (i.e. Abstract, Poster, and Presentation). All major milestones should be included in the timeline.



Appendix 2: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

Task: This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
Del Huband	Creative, Problem Solving, Strategic	I like being able to improve my skills as a programmer.	hubanddh@vcu.edu (804) 401-9219
Parker Gonzaga	Problem Solving, organization, some database experience	I enjoy working as a team and completing tasks efficiently. Willing to learn new techniques in order to fulfill project requirements.	gonzagapv@vcu.edu (571) 524-0964
Alex Nguyen	A bit of industry experience, problem solving, communication	I'm interested in AI.	nguyena57@vcu.edu (571) 340-0035
Brian Vo	Communication, analytical thinking	I like to develop things that have practical uses.	vobq2@vcu.edu (571) - 527 - 8826

Other Stakeholders	***Notes***	***Contact Info***
Rachita Sowle (Advisor)		sowler@vcu.edu
Mahesh Nair (Sponsor)		mahesh.bahulleyannair@capitalone.com
Derrick Murry (Sponsor)		derrick.murry@capitalone.com

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Task: Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

Resources: More information and an example Team Culture can be found in the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

Culture Goals	***Actions***	***Warning Signs***
Being on time to every meeting	- Set up meetings in shared calendar - Ensure all parties have date cleared prior to setting meetings	- Student misses meetings regularly - Consistently missing meeting is brought up to advisor
Informing the group of any delays in completing assignments	- Stay up to date with each other's project responsibilities - Set reasonable deadlines and note when an extension is needed	- Student shows up for weekly meeting with no considerable work done
Accomplishing goals ahead of time	- Set deadlines - Work at a pace that would leave leeway before deadline	- Procrastinating - Missing deadlines
Delegating roles and understanding your job	- Assign each group member a task - Write out clear description for each task	- Unfinished work on any group member's end, causing project progress to slow down
Open Communication	- Don't be afraid to ask questions	- Student ceases communication with group

Step 3: Time Commitments, Meeting Structure, and Communication

Task: Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will same person take on that role for the duration of the project?

Required: How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

Meeting Participants	***Frequency*** ***Dates and Times / Locations***	***Meeting Goals*** ***Responsible Party***
Students Only	As Needed, On Discord Voice Channel	Update group on recent challenges and accomplishments (Brian will scribe important details from the meeting)
Students Only	Every Thursday 5-7 in West Eng. 101	Actively work on project with team and communicate weekly updates and set future goals (Del will write out upcoming goals and other information from this meeting)
Students + Faculty advisor	Currently planned to meet weekly or bi-weekly with our advisor.	Update faculty advisor and get answers to our questions (Del will scribe the important details from this meeting)
Project Sponsor	Currently planned every Wednesday. (Can be increased/ adjusted)	Update project sponsor and make sure we are on the right track (Brian will scribe important details from the meeting)

Step 4: Determine Individual Roles and Responsibilities

Task: As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

Communication Leaders

Suggested: Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

Suggested: Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.
6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

Team Member	***Role(s)***	***Responsibilities***
Del Huband	Test Engineer	✓Extensively test all code to ensure it operates as intended ✓Analyzes code efficiency ✓Ensures programs don't have significant overhead
Parker Gonzaga	Financial Manager	✓Keep track of expenses ✓Handle purchase requests ✓Ensure expenses stay within project budget
Brian Vo	Project Manager	✓Contact for sponsor ✓Keep group on track, track agendas, schedule meetings
Alex Nguyen	Systems Engineer	✓Analyze client's initial design specifications ✓Establishes product specifications ✓Develops system architecture

Step 5: Agree to the above team contract

Team Member: Del Huband *Signature: Del Huband*

Team Member: Alex Nguyen *Signature: Alex Nguyen*

Team Member: Parker Gonzaga *Signature: Parker Gonzaga*

Team Member: Brian Vo *Signature: Brian Vo*

References

[1] Contentsquare. (n.d.). *Hotjar*. Retrieved October 6, 2024. https://www.hotjar.com/

[2] Glassbox. (n.d.). *Glassbox*. Retrieved October 7, 2024. https://www.glassbox.com/

[3] Crazy Egg. (n.d.). *Crazy Egg*. Retrieved October 10, 2024. https://www.crazyegg.com/

[4] Bekmambetov, T. Bedford, G. Kehoe, D. 2019. Method for recording, editing and reproduction of computer session (European Patent 3457295A2). European Patent Office. https://patents.google.com/patent/EP3457295A2/en?q=(%22browser+playback%22)&oq=%22browser+playback%22