Software Requirements Specification

for

Amazon Connect Supervisor Insights

Version 2.0 approved

Prepared by Team 2

Safe Corp JIVAA

March 14th, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft	08/March/2024	Initial Development	1.0
First Revision	13/March/2024	Missing many specifications and requirements	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Amazon Connect Supervisor Insights Web Application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both stakeholders and developers of the system, so that they can better understand the application's scope and its limitations. In more detail, the document defines hardware and software requirements to execute the project and helps both designers and developers to assist in the *Software Delivery LifeCycle* (*SDLC*).

1.2 Document Conventions

For this document, the typographical convention of decreasing in font size by two points for each level of titling is used with the exception of the cover page. Bold letters are utilized for the distinction of major sections of the document shown in the Table of Contents as well as for the headers of the input fields on the Revision History section. Italic letters are used for the first use of terms found in the glossary, so that unclear terms that won't have an immediate explanation can be more easily identified and consequently understood.

Sections are enumerated, subsections use the section number followed by their respective subsection number, styled in bold text and in cursive, and listings follow this standard too. Additionally, the starting line of each paragraph is indented as per common styling guidelines such as APA or Chicago to provide a visual break in between paragraphs and for a better sense of continuity in between sections.

As for information, documentation and other sources, the convention being used is the IEEE reference style found on the IEEE Editorial Style Manual for Authors[1] since the referencing information available on the documentation sites is limited and this style was deemed to be more readable and concise than the alternatives. Moreover, an "Additional Resources Consulted" section has been added for documentation sites which outline basic information used for the ideation of realistic features, but are not an actual source of information that has been borrowed or incorporated into our work.

1.3 Project Scope

The Amazon Connect Supervisor Insights Web Application aims to elevate the performance of *Contact Centers* (*CC*) by creating a tool which can be used by supervisors for managing and leading agents in a CC. The interface will provide a dashboard for both supervisor and agents, where they will be presented with relevant information of their ongoing calls, along with their call history. The tool will be scalable, in order to provide service for bigger CC, it must also help with the stress of the agents by allowing for swift action from the supervisors whenever an issue arises. Furthermore, the data from calls must be analyzed and showcased on the dashboard in real time. Metrics with relevant insights for every call, and every group of calls pertaining to each supervisor will also be presented; transcriptions will be available in real time during ongoing calls, both of which will aid in the supervisor's decision making. Dashboard functionality will depend on the type of the user, making the usage for agents much simpler and more straight-forward. Finally, pertinent analytics of sentiments throughout calls will be presented.

1.3.1 Enhancements through Amazon Connect Integration

Scalability and Stress Management: The *Amazon Web Services* (*AWS*) cloud infrastructure supports the dynamic scaling of resources to manage fluctuating call volumes, ensuring that the CC can handle peak times without compromising on performance or agent well-being. This integration allows for a seamless adjustment to workload demands, reducing stress and burnout among agents by optimizing call distribution based on real-time analytics.

Real-Time Metrics and Insights & Lack of knowledge: Supervisors gain access to near-real-time metrics (RTM) and actionable insights on agent performance and caller-agent interactions. This includes alerts for calls requiring intervention, whether flagged by the agent or identified through *sentiment analysis*, enhancing the capacity for timely support, coaching and addressing certain knowledge gaps.

Call Transcriptions and Analysis: The application provides access to past call transcriptions and audio recordings, leveraging Amazon Connect's Contact Lens for sentiment analysis and speech analytics. This empowers supervisors with data-driven insights for training and quality assurance, fostering continuous improvement in customer service.

Agent Empowerment Dashboard: Agents access a user-friendly dashboard on their tablets, showcasing current call status, basic controls, and a feature to request supervisor intervention. Integrating with Amazon Bedrock, the application offers a stream of potential solutions for the caller's problem, tapping into Foundational Models and Vector Stores to semantically query documents for relevant information.

By embedding Amazon Connect's functionalities, the application not only addresses the immediate needs of managing call volume and improving agent performance but also strategically positions the CC to adapt to evolving demands and technological advancements. This holistic approach ensures a robust framework for exceptional customer service and operational efficiency.

1.4 References

[1] IEEE. *IEEE Editorial Style Manual for Authors.* (February 29, 2024). .https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Editorial-Style-Manual-for-Authors.pdf

[2] Amazon Web Services. *Amazon Connect Customers.* https://aws.amazon.com/connect/customers/

[3] Amazon Web Services. *Amplify Documentation*. https://docs.amplify.aws/

[4] Amazon Web Services. *Amplify UI.* https://ui.docs.amplify.aws/

[5] Amazon Web Services. *How Amazon Connect works with IAM.* https://docs.aws.amazon.com/connect/latest/adminguide/security_iam_service-with-iam.html

[6] Amazon Web Services. *Amazon Connect.* https://docs.aws.amazon.com/connect/

[7] Amazon Web Services. *Amazon Connect Contact Lens*. https://docs.aws.amazon.com/connect/latest/adminguide/contact-lens.html

[8] Acernity UI. *All Components.* https://ui.aceternity.com/components

[9] Tailwind CSS. *Rapidly build modern websites without ever leaving your HTML.* https://tailwindcss.com/

1.4.1 Additional Resources Consulted

[10] Amazon Web Services. *CurrentMetric.* https://docs.aws.amazon.com/connect/latest/APIReference/API_CurrentMetric.html

[11] Amazon Web Services. *HistoricalMetric.* https://docs.aws.amazon.com/connect/latest/APIReference/API_HistoricalMetric.html

[12] Amazon Web Services. *Launch the CCP.* https://docs.aws.amazon.com/connect/latest/adminguide/launch-ccp.html

[13] Amazon Web Services. *Metric Actions.* https://docs.aws.amazon.com/connect/latest/APIReference/metrics-api.html

[14] Amazon Web Services. *MetricV2*.
https://docs.aws.amazon.com/connect/latest/APIReference/API_MetricV2.html

[15] *Especificación de Requisitos según el estándar de IEEE 830*. (2008, October).
https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf

2. Overall Description

We are developing an application for a CC, hypothetically, a CC for a traveling company, that uses Amazon Connect for its CC management. This implies development on all three major components of software: frontend, backend, and database management.

2.1 Product Perspective

CCs are an environment for highly chaotic and variable information. This is why being able to classify the efficacy and context of calls is of the utmost importance. Currently, few services offer solutions such as this one, and thus developing this application is a great way to get ahead of the game, specifically by implementing the connection with Amazon Connect, which is used by companies such as CapitalOne, and Subway.[2]

2.2 Users and Characteristics

There will be a total of 2 classes for our users. Firstly, the supervisors, who must have a dashboard, will be the highest level of user, implying that they will be overseeing the other level of user. Thus, their dashboard will have to present much of the necessary information for that. The other level of user will only have access to their own calls, and their dashboard, if any, will not have the same level – or amount – of information.

1. Agents: Agents interact with customers, they are on the other end of the phone, receiving calls, advising and solving problems. They must be able to operate a dashboard where they will interpret KPI's to improve the quality of the call and user experience. They will have an "agent account" where they will be able to login and analyze the provided dashboard. They will have an "Ask for help" button to request their supervisor to enter the call if assistance is needed.
2. Supervisors: One supervisor is assigned at most 15 agents. They will serve as managers who evaluate the performance of the agents and have the ability to change assignments (of agents) depending on the needs of the CC and prioritizing urgent calls (urgent calls are determined by a sentiment analysis which evaluates the quality of the

call). Moreover, supervisors have administrator privileges, as they will set and manage both permissions and roles inside the Amazon Connect platform.

2.3 Operating Environment

The frontend will run using Next JS 14 and will therefore feature Server side rendering (SSR). It will be deployed in a CloudFront Global Edge Network instance in AWS using Amplify, the storage will be managed using S3 Buckets and DynamoDB (exposed through Amplify AppSync using the GraphQL *API (Application Programming Interface*)[3]. Moreover, authentication will be managed using Cognito with Amplify UI[4] components and IAM[5]. The call management will take place within an Amazon Connect instance, which will also be in turn exposed to other services like Contact Lens, Lex or Kinesis.[6][7]

2.4 Design and Implementation Constraints

Our project development is bound by several constraints that dictate our approach. Firstly, adherence to corporate directives mandates the utilization of Amazon Connect Instances as the foundational framework. JavaScript serves as our primary programming language for application development, subject to potential adaptations based on the requirements of lambda functions. A strict timeline of 15 weeks is allocated for project completion, comprising 5 weeks dedicated to documentation and 10 weeks for development. Furthermore, our development team's operational environment is constrained by the diversity of devices and operating systems, encompassing Microsoft and Apple hardware, and Windows, MacOS, and Linux operating systems. These constraints, whether corporate policies, technological dependencies, or resource allocation, shape the trajectory of our project implementation, influencing decisions at every stage. It is required for the system to incorporate AWS services and *PII (Personal Identifiable Information)* must be encrypted in transit and at rest (this is done with Amplify's DynamoDB and the AppSync API [3]).

2.5 Assumptions and Restrictions

1. The CC will be available 24 hours.
2. The CC will have a set of rules and regulations for the interactions, internal protocols and processes.

3. The Agents and Supervisors will **always** adhere to a ranking of impacts as follows: Reputational, Legal, Functional, Economical. Their actions will sacrifice a lower rank if a higher rank is at risk, avoiding at all costs a Reputational impact.
4. The dashboard users will access the app at any time from a PC or iPad.
5. Interactions with the app and client are swift and flow effortlessly between pages or sections.
6. Agents will be able to see clearly their next call.
7. The CC will follow ideals, goals, philosophy and feelings to organically merge with their daily activities, and to further these core attributes into the user experience by the Agent.
8. Supervisors can only intervene in one call at a time.
9. The Supervisor will not necessarily inform the Agent when they drop into the call; however, they will always announce privately of an intervention.
10. The majority of the reasons why callers call is because of **problems** related to their interaction with the company.
11. An Agent can only take one call at a time.
12. The system does not require users with "admin" privileges, rather, it only requires to be used by supervisors and agents, and to have different experiences for them based on their role.
13. The supervisor is **always** concentrated and on the lookout for possible problems with calls.
14. Each ongoing call queue is assigned to a single supervisor.
15. The CC has policies that allow constant performance monitoring of both agents and callers (which are valid in the country where the CC operates).

3. System Features

3.1 Authentication

3.1.1 Description

One big milestone within the application will be the authentication. This includes all aspects of a proper user experience. It will be managed using Amplify[4] which itself uses Amazon Cognito. Authentication has one of the highest priorities in the project given the fact that resources should only be provided with the appropriate roles of authenticated users.

3.1.2 Stimulus/Response Sequences

Stimulus: Supervisor or agent enters their credentials to log in to the platform.

Response: If the provided credentials are correct, the system grants access.

Response: If the credentials are incorrect, access is denied.

Reponse: Supervisor or agent clicks the log out button.

Response: The user's session is ended and they have to log in again when they come back to the system.

Response: Supervisor or agent requests a password change.

Response: Either by email or in the platform itself (to be defined), the user is able to change their password.

3.1.3 Functional Requirements

- The user should be able to login with their username and password, and if they don't get their credentials correct, they should receive feedback.
- The user should be able to logout of their account, deleting any persistent session data as well, to ensure security.
- The application should send, parse and manage JWT in order to keep the authenticated state alive, for a specified duration.
- The user should be able to change their password if they forget it.
- Both agents and supervisors should interact with the system and their views and permissions will be different, roles will be managed using Cognito User Groups.

3.2 Call Monitoring

3.2.1 Description

The supervisor should be able to have a list of ongoing calls assigned to its subordinates along with their statuses and RTM to provide accurate feedback, intervene if necessary, or mark them for tracking. The alerts should be persistent, eye-catching and informative; they should receive important information based on the caller's location to understand generalized issues. The agent should also be able to see relevant information regarding the caller or their location.

3.2.2 Stimulus/Response Sequences

<u>Supervisor's Stimulus</u>: The supervisor views the dashboard.

Response: The supervisor views the map of their area.

Response: Views an alert.

Response: Understands the information.

Response: Determines whether to intervene or not, or mark for tracking.

Response: Utilizes this knowledge in future decisions or interventions.

Agent’s Stimulus: The agent views their information dashboard with the relevant information for the call.

Response: Understands the information.

Response: Determines whether to intervene or not, or mark for tracking.

Response: Judges if the information provided is relevant and uses it to provide a solution.

Response: Utilizes this knowledge in future decisions or interventions.

3.2.3 Functional Requirements

- The ongoing calls of a supervisor’s agents should be listed in their dashboard.
- There should be recommendations for an agent in an ongoing call on how to resolve it based on the caller’s location.
- A supervisor should be able to filter and order ongoing calls based on certain criteria.
- The supervisor should receive an alert when a call is going poorly, and see the call at the top of the calls list.
- The supervisor should be able to mark calls to distinguish them when looking at the historical records and transcripts.

3.3 Sentiment Analysis

3.3.1 Description

This feature is of high priority, and it describes that a *Sentiment Analysis* should be performed for an ongoing call, in order to be able to notify the supervisor about this call’s status and how much an intervention is needed.

3.3.2 Stimulus/Response Sequences

Stimulus: The agent is on a call.

Response: Sentiment Analysis is performed as the call progresses.

Response: AI-generated suggestions are displayed for the agent.

Response: The call’s sentiment, ranging from negative to positive, is streamed.

3.3.3 Functional Requirements

- There should be a live sentiment analysis of an ongoing call to be able to alert a supervisor of potential problems in a call.

3.4 Speech to Text

3.4.1 Description

This is also a very high priority feature, since this is a dependency for many of the system's requirements, like the ability to generate live suggestions based on the call's analysis, and to provide a sentiment analysis of the current call.

3.4.2 Stimulus/Response Sequences

Stimulus: Agent is on a call.

Response: The call is analyzed and suggestions are provided to the agent based on the analysis of the call's speech.

Response: A sentiment analysis is performed given the call's analysis, and its results are streamed to the supervisor's dashboard.

Response: There is a metric that represents how confident the call transcription is.

3.4.3 Functional Requirements

- The transcription of the call should be available to the supervisor while the call is happening.
- The call's content should be analyzed in real time to provide suggestions on solutions to the agent.
- The sentiment analysis result should be streamed to the supervisor.
- The call transcription should have a confidence metric that breaks down how reliable the transcription that is currently being processed is.

3.5 Agent Performance and Recognition

3.5.1 Description

This feature is medium priority because it was specified in the project proposal, but it basically includes an internal system to track specific metrics of agents' performance and reward them (internally within the system or externally, to be defined) in some way. Also, there are periodical reports generated in order to be able to check performance and other relevant data.

3.5.2 Stimulus/Response Sequences

Stimulus: One of the agents' observable or applicable metric reaches a certain point, which is tracked by our lambda functions and now means that a reward is in order.

Response: The lambda is triggered, the person to whom it may concern is notified, and the agent is rewarded.

Response: A lambda function is triggered because the time frame is reached.

Response: The corresponding report is generated.

3.5.3 Functional Requirements

- The system should recognize the agents in some way when they meet certain metrics (to be defined).
- The system should generate reports periodically for a supervisor's agents' performance.

3.6 Historical Records and Metrics

3.6.1 Description

This feature is of high priority, it records previous calls and keeps track of important details by storing them as metrics.

3.6.2 Stimulus/Response Sequences

Stimulus: The supervisor views the dashboard

Response: The Metrics and Historical Records are displayed in the dashboard from latest to oldest.

Response: The supervisor clicks on a record or metric.

Response: The information regarding said record or metric is displayed.

3.6.3 Functional Requirements

- The system should have previous call recording transcripts stored for future consultation and analysis.
- The recordings of previous calls should be stored and available for replay.
- The system should have a set of records that can give information about the customer's past interactions with the *CC*.

3.7 Retrieval Augmented Generation

3.7.1 Description

During a call CC's participants should be able to see related articles or previous solutions to similar problems.

3.7.2 Stimulus/Response Sequences

Stimulus: The CC's Users enter a call

Response: The CC's Users view the suggestions dashboard

Response: The CC's Users determine which articles or solutions to use

Response: In case of useless suggestions, the Users change the subject to search

3.7.3 Functional Requirements

- An agent should be prompted with possible solutions to the caller’s problem, or suggested questions to ask the caller, based on the speech to text analysis performed in real time.
- An agent can ask a Copilot type of chat about actions that they can take regarding their current call, and it will look in the Documentation and Standards data in order to comply with company regulations and desired management.

3.8 Navigation

3.8.1 Description

This is a low priority feature, a “could have”, that basically implements keybindings for the system’s users.

3.8.2 Stimulus/Response Sequences

Stimulus: A user presses a supported key combination.

Response: The system reacts to the specific key binding accordingly.

3.8.3 Functional Requirements

- The system should include keybindings that require key combinations to speed up the users navigation, while avoiding accidental triggers.

4. Data Requirements

Our system will consume the data provided by AWS Connect which comes in the form of strings, arrays, numbers and audio files. From the users the inputs will be mainly arrays of strings or numbers that depict their current interactions with the end users or their managers/subordinates. As output the system will provide strings, numbers and information/article objects that may vary in content.

4.1 Logical Data Model

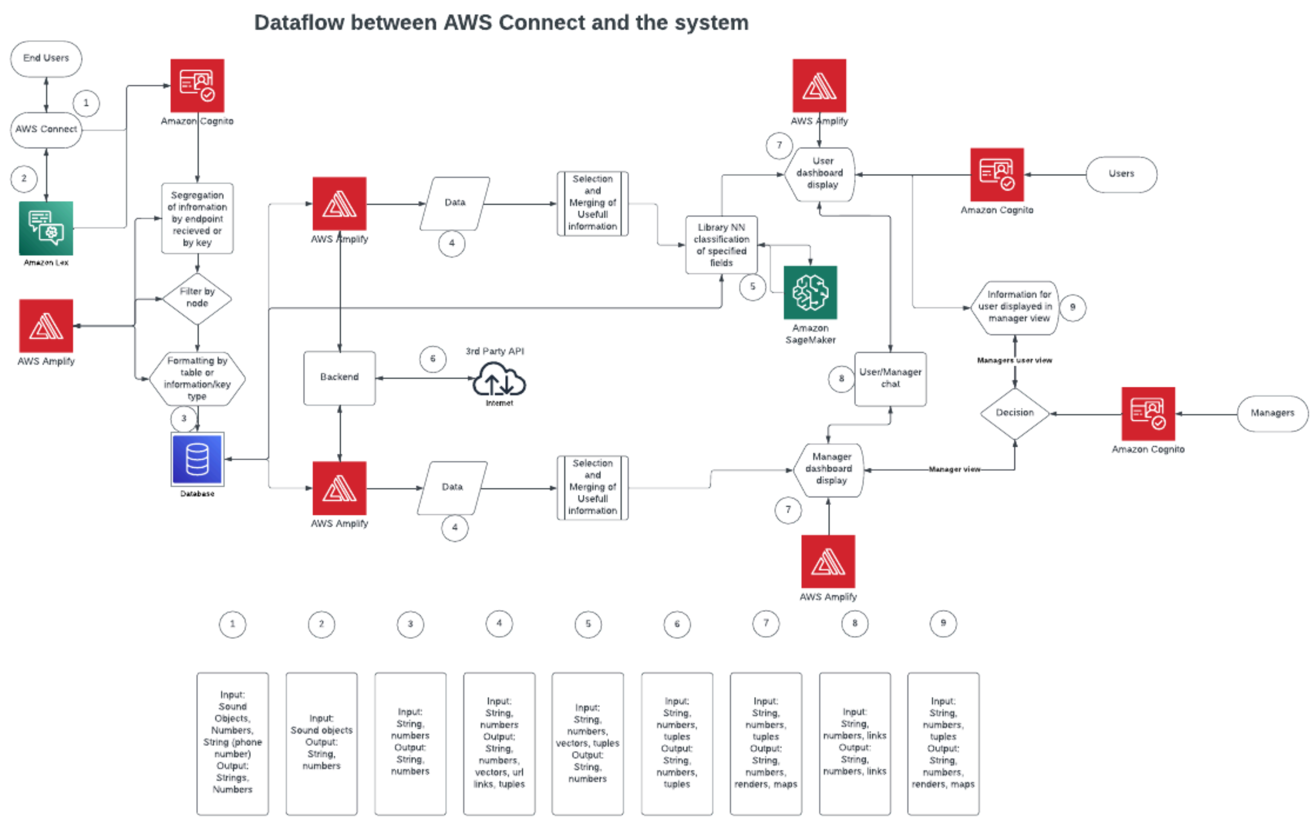

Dataflow between AWS Connect and the system
End Users
AWS Connect
Amazon Lex
Amazon Cognito
Segregation of infromation by endpoint recieved or by key
Filter by node
Formatting by table or information/key type
AWS Amplify
Database
AWS Amplify
Data
Selection and Merging of Usefull information
Backend
3rd Party API
Internet
AWS Amplify
Data
Selection and Merging of Usefull information
Library NN classification of specified fields
Amazon SageMaker
AWS Amplify
User dashboard display
Amazon Cognito
Users
Information for user displayed in manager view
Managers user view
Decision
Amazon Cognito
Managers
User/Manager chat
Manager dashboard display
Manager view
AWS Amplify
1
2
3
4
5
6
7
8
9
Input: Sound Objects, Numbers, String (phone number) Output: Strings, Numbers
Input: Sound objects Output: String, numbers
Input: String, numbers Output: String, numbers
Input: String, numbers Output: String, numbers, vectors, url links, tuples
Input: String, numbers, vectors, tuples Output: String, numbers
Input: String, numbers, tuples Output: String, numbers, tuples
Input: String, numbers, tuples Output: String, numbers, renders, maps
Input: String, numbers, links Output: String, numbers, links
Input: String, numbers, tuples Output: String, numbers, renders, maps

Figure 4.1.1. Logical Data Model

4.2 Data Dictionary

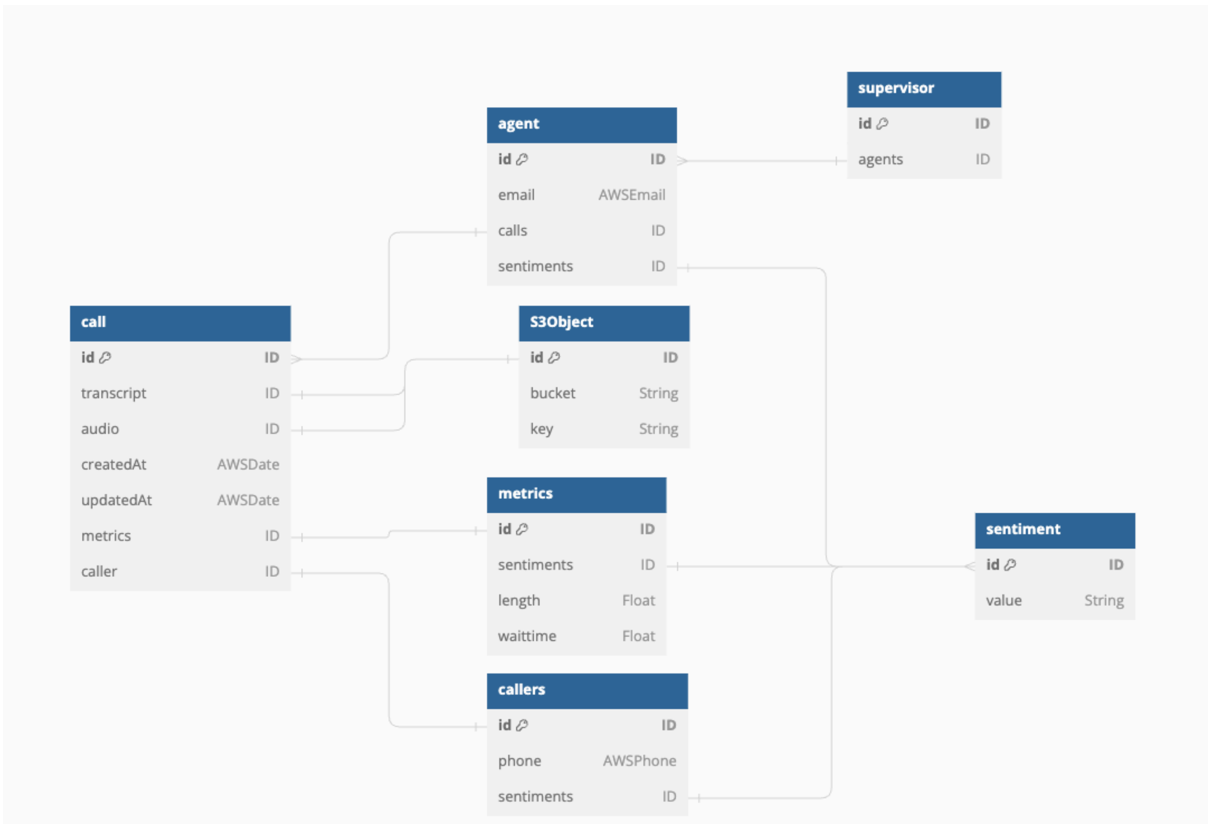


Figure 4.2.1. Entity Relationship Diagram

Caller				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	An ID to describe each instance of a new caller differently.
phone	AWSPhone	10	unique	The caller's phone, must be unique to relate the historical sentiments to themselves.
sentiments	ID	1-to-many	non null	The sentiments over the entire history of the caller's calls. This is used for analytics eventually.

Table 4.2.1 Caller Entity

Call				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	An ID to describe each instance of a new caller differently.
transcript	ID	1-to-1	unique References transcript.id	The caller’s phone, must be unique to relate the historical sentiments to themselves.
audio	ID	1-to-1	unique References audio.id	The sentiments over the entire history of the caller’s calls. This is used for analytics eventually.
agentId	ID	1-to-1	non null References agent.id	The agent’s email to link their Connect profile to the database, and therefore the calls they’re in.
createdAt	AWSDate	10	ISO 8601 (RFC 3339 - Section 5.6)	Date of creation of the call.
updatedAt	AWSDate	10	ISO 8601 (RFC 3339 - Section 5.6)	Date (time) of update of the call (in case sentiment changes).
metrics	ID	1-to-1	Refernces metrics.id	The id for the metrics of the call, serves to facilitate sentiment analysis through the call.
caller	ID	1-to-1	References callers.id	The id of the caller, linked through their phone.

Table 4.2.2 Call Entity

Metric				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	An ID to describe each instance of a new metric’s call differently.
sentiments	ID	1-to-many	non null	The sentiment analysis throughout the call. This is used to present an analysis per call.
length	Float	10	non null	The duration of the call. This is only posted once the call ends, thus triggering the final update for it.
waitTime	Float	10	non null	The wait time that the caller had to endure before they contacted an agent. This is posted on the creation of the call, and thus the metric instance.

Table 4.2.3 Metric Entity

S3Object				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	Id to link the S3 object to a transcript or audio field.
bucket	String	20	non null	The bucket in which the file is saved.
key	String	20	non null	The file key, which is located inside the bucket. This helps find the file easily.

Table 4.2.4 S3Object Entity

Supervisor				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	An ID to describe each instance of a new supervisor differently.
agents	ID	1-to-many	non null	The array of agents under the supervisor. Helps link ongoing and past calls, and their respective data to a supervisor.

Table 4.2.5 Supervisor Entity

Sentiment				
Field Name	Data type	Field Length	Constraint	Description
id	ID	28	unique	An ID to describe each instance of a new metric's call differently.
value	String		enum	The value of the sentiment, currently defined as "positive", "negative", "neutral"

Table 4.2.6 Sentiment Entity

4.3 Reports

The application will be giving real-time insights to supervisors and agents through speech-to-text analysis of the ongoing calls. For the agents, this analysis would be composed of two main parts: an analysis of the caller's profile (previous calls, behavioral characteristics, etc) and an analysis of possible actions to take regarding the caller's problem. The first would be "static data" that would not change throughout the call, and the second one would be more like a data stream that keeps on being updated according to the call's status, the caller's requests and concerns, etc. It could even be interactive, meaning that the agent could select from a range of options to improve the accuracy of the suggested "solutions" to the caller's problem.

On the other hand, the application will keep a record of the call, and store some valuable data, like its resolution status, duration, time to respond, etc (some of this data is already provided by Amazon Connect), and this can be used to further improve the live suggestions, by being added to the database.

Finally, as a feature that is not mandatory but could be included, some reports may be generated monthly or weekly for agents' performance, to gain specific insights. Metrics for this are to be defined.

4.4 Data Acquisition, Integrity, Retention, and Disposal

For the sake of development, data will be generated artificially. Some documents that are relevant for our proposal (such as *Employee Guidelines*, *Protocols and Standards* or *Wiki* documents that could be used internally in a CC) will be generated using a *Large Language Model*. Moreover, the development team will mimic CC interactions to better understand the Amazon Connect APIs and generate some data such as call transcripts and audio recordings which will be stored in S3 Buckets or relevant metadata and analytics which shall be managed in a DynamoDB database. Finally, the application users (*agents* and *supervisors*) will generate even more data which shall be managed through Amazon Connect such as performance metrics and lists of ongoing calls.

When using a NoSQL database like DynamoDB, integrity is an important consideration, as it is a common practice to *denormalize* tables (as opposed to SQL database management systems). We decided to separate tables when appropriate (see **Section 4.2** for an ER diagram) to guarantee integrity, for instance using separate entities to store S3 references, metric records and references to calls.

Data retention will occur in different services, for instance DynamoDB for structured records, Amazon S3 for file (or *Object*) storage, Amazon Cognito for user pools, client cookies or local storage to manage sessions with tokens and Pinecone to index document chunks.

Finally, data will be disposed of using an API with protected endpoints that will only allow supervisors to delete records. Moreover, Amazon Connect also manages some deletions based on the creation dates of the records.

5. External Interface Requirements

5.1 Graphical Identity

Typography

- Heading: Nunito Sans, Regular (400), 36pt.

- Title: Nunito Sans, Regular (400), 32pt.
- Subtitle: Nunito Sans, Regular (400), 20pt.
- Paragraph: Nunito Sans, Regular (400), 12pt.

Colors

● Primary:	#869AE9
● Secondary:	#27E2D3
● Tertiary:	#DAF6F4
● Text:	#0A0A0A
● Contrast:	#E4E7EB
● Gray:	#F5F7FA

5.2 Buttons

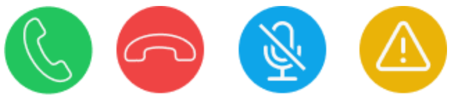


Figure 5.2.1. Action Buttons

Figure 5.2.2. Text button

5.3 UI Libraries and Frameworks

1. Aceternity UI[8]

   A headless UI library that uses *Framer Motion* and *Tailwind CSS* to style and animate components with unique transitions.

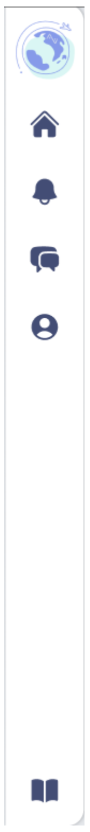
2. Amplify UI[4]

   The Amazon Amplify UI library. It makes it easy to incorporate authentication and despite using the Amplify Design System by default it can be styled and integrated with Figma. It also follows accessibility standards such as WCAG and WAI-ARIA to allow any user to interact with the system.[3]

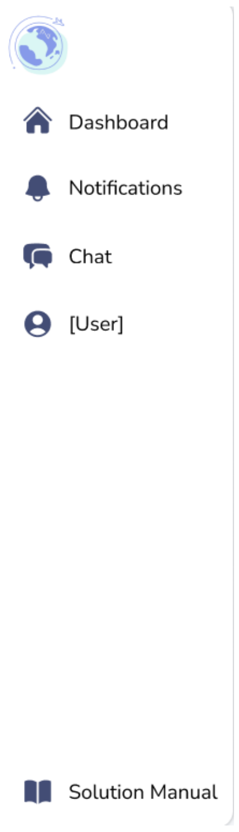
3. Tailwind CSS[9]

   Tailwind is a CSS framework that uses classes to style HTML elements (or JSX components). It follows certain conventions regarding color palettes, dimensions and typographies which despite being modifiable usually works great out of the box.

5.4 Components and Views



Subfigure A. Collapsed Navbar



Subfigure B. Expanded Navbar

Figure 5.4.1. Collapsed and expanded navigation bar.

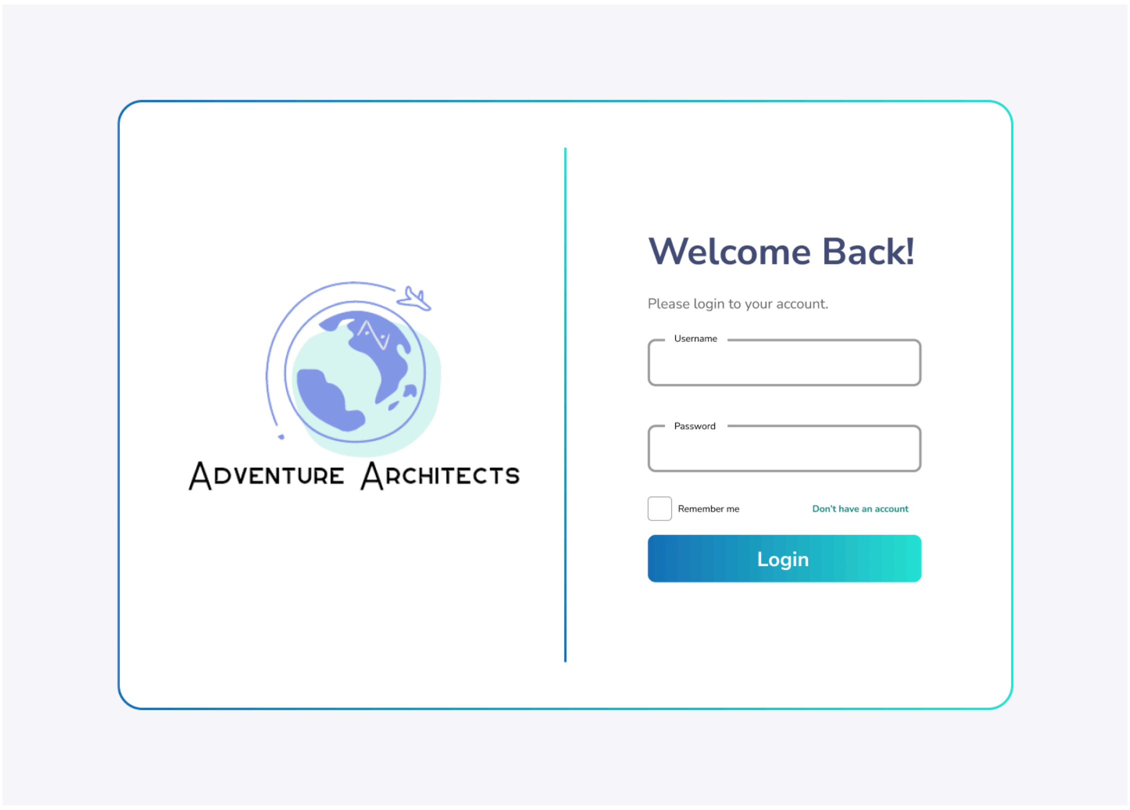
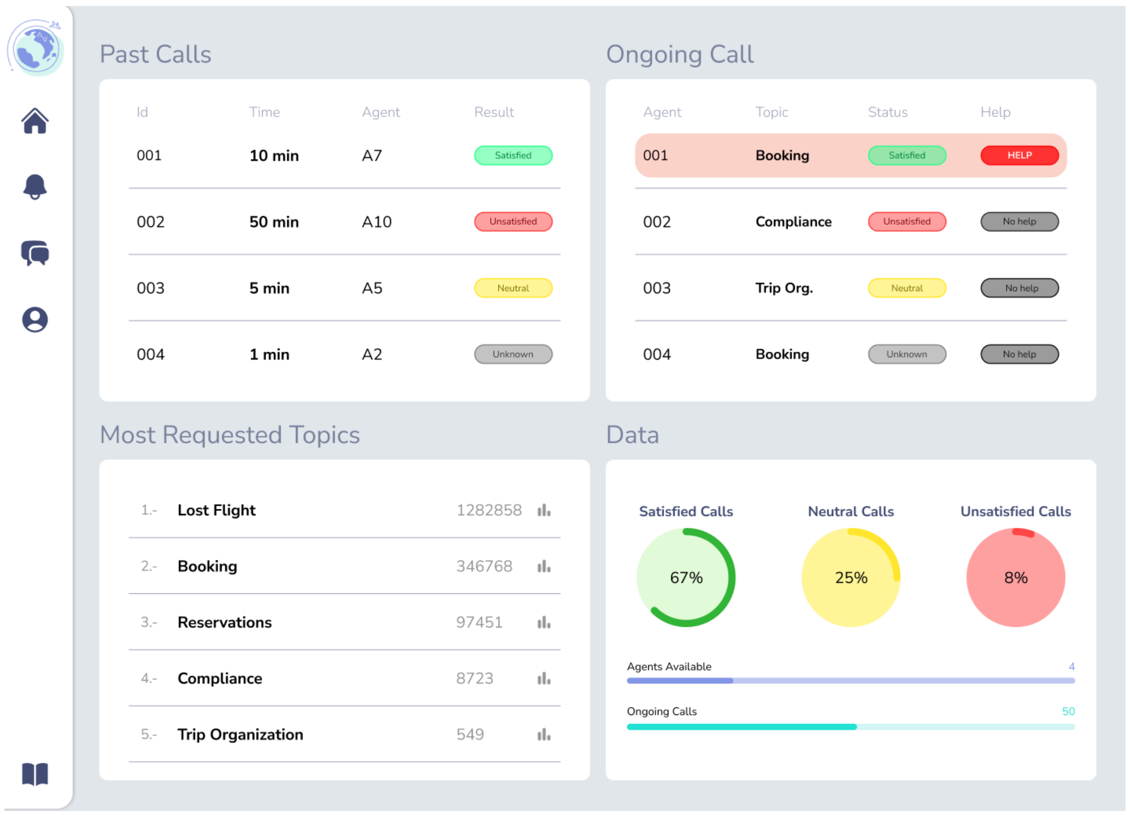


Figure 5.4.2. Login interview



Past Calls

Id	Time	Agent	Result
001	10 min	A7	Satisfied
002	50 min	A10	Unsatisfied
003	5 min	A5	Neutral
004	1 min	A2	Unknown

Ongoing Call

Agent	Topic	Status	Help
001	Booking	Satisfied	HELP
002	Compliance	Unsatisfied	No help
003	Trip Org.	Neutral	No help
004	Booking	Unknown	No help

Most Requested Topics

1.-	Lost Flight	1282858
2.-	Booking	346768
3.-	Reservations	97451
4.-	Compliance	8723
5.-	Trip Organization	549

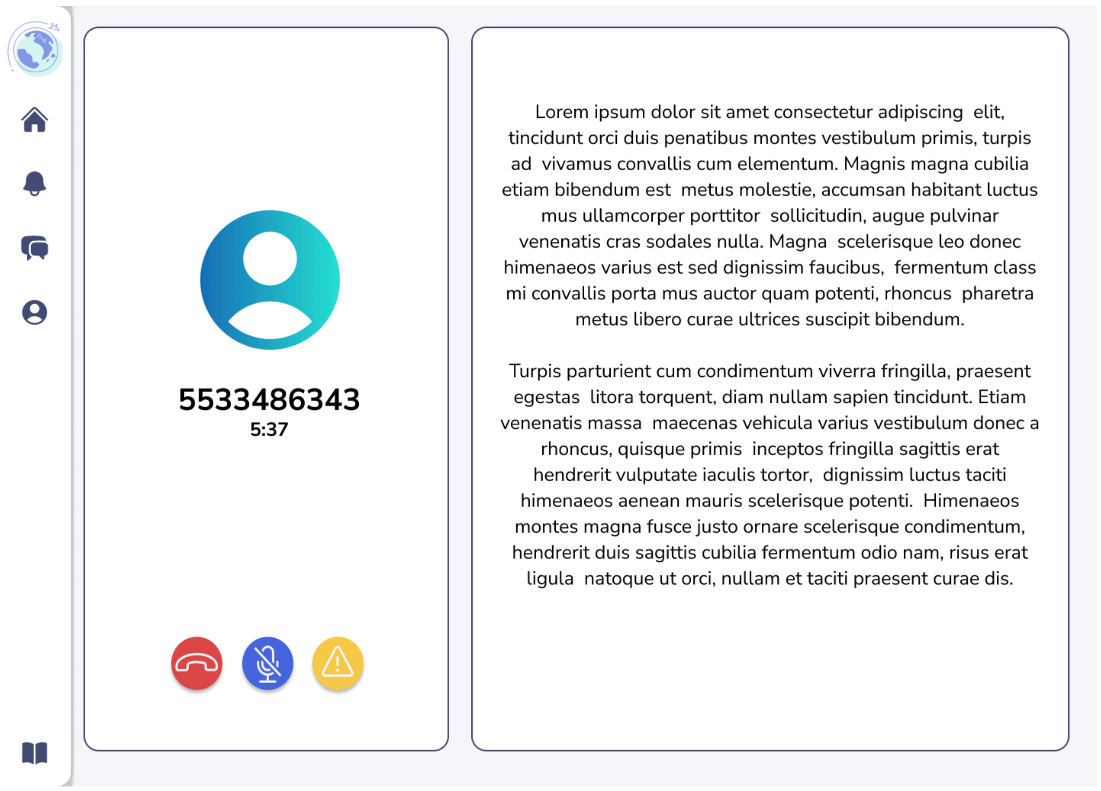
Data

Satisfied Calls	Neutral Calls	Unsatisfied Calls
67%	25%	8%

Agents Available 4

Ongoing Calls 50

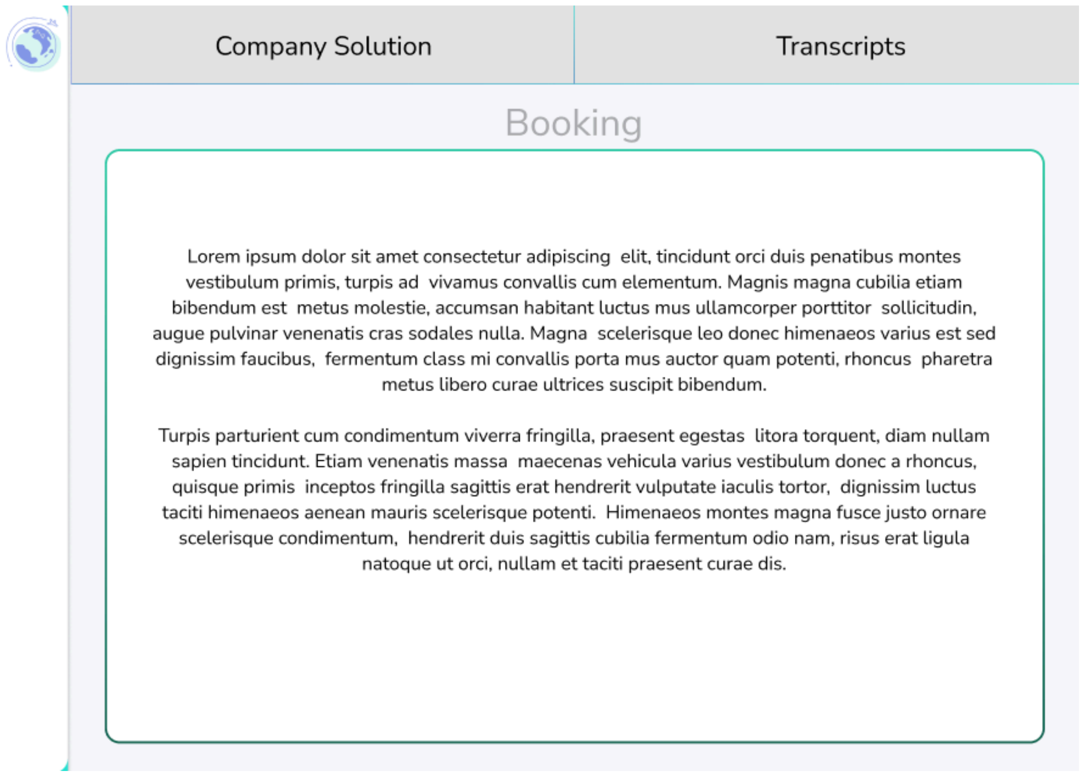
Figure 5.4.3. Supervisor Dashboard



Lorem ipsum dolor sit amet consectetur adipiscing elit, tincidunt orci duis penatibus montes vestibulum primis, turpis ad vivamus convallis cum elementum. Magnis magna cubilia etiam bibendum est metus molestie, accumsan habitant luctus mus ullamcorper porttitor sollicitudin, augue pulvinar venenatis cras sodales nulla. Magna scelerisque leo donec himenaeos varius est sed dignissim faucibus, fermentum class mi convallis porta mus auctor quam potenti, rhoncus pharetra metus libero curae ultrices suscipit bibendum.

Turpis parturient cum condimentum viverra fringilla, praesent egestas litora torquent, diam nullam sapien tincidunt. Etiam venenatis massa maecenas vehicula varius vestibulum donec a rhoncus, quisque primis inceptos fringilla sagittis erat hendrerit vulputate iaculis tortor, dignissim luctus taciti himenaeos aenean mauris scelerisque potenti. Himenaeos montes magna fusce justo ornare scelerisque condimentum, hendrerit duis sagittis cubilia fermentum odio nam, risus erat ligula natoque ut orci, nullam et taciti praesent curae dis.

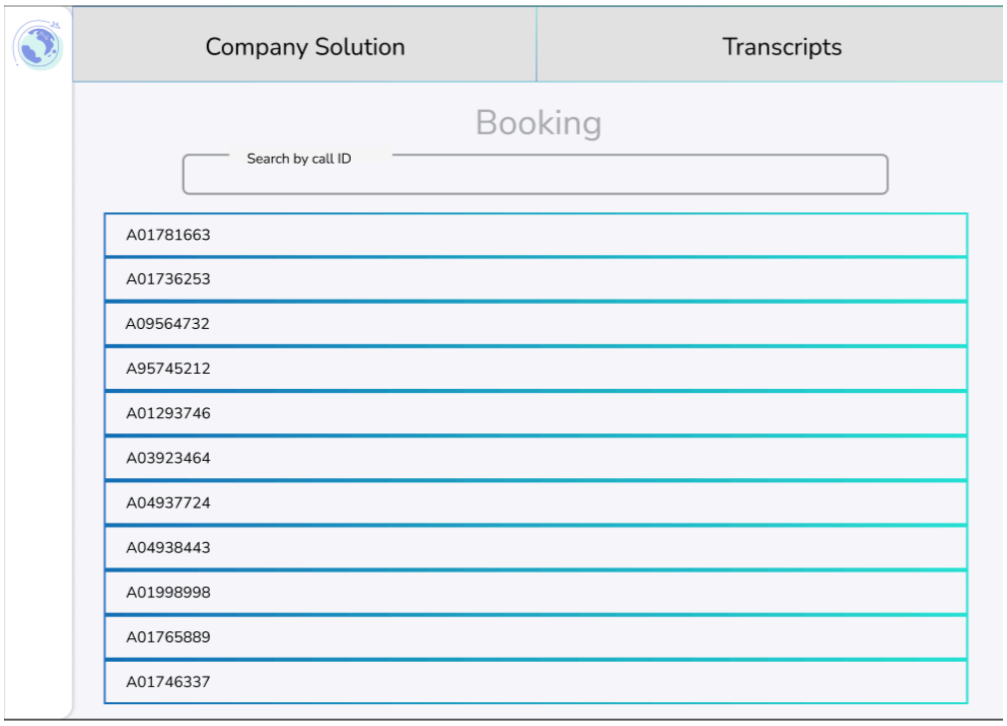
Figure 5.4.4. Agent Dashboard



Lorem ipsum dolor sit amet consectetur adipiscing elit, tincidunt orci duis penatibus montes vestibulum primis, turpis ad vivamus convallis cum elementum. Magnis magna cubilia etiam bibendum est metus molestie, accumsan habitant luctus mus ullamcorper porttitor sollicitudin, augue pulvinar venenatis cras sodales nulla. Magna scelerisque leo donec himenaeos varius est sed dignissim faucibus, fermentum class mi convallis porta mus auctor quam potenti, rhoncus pharetra metus libero curae ultrices suscipit bibendum.

Turpis parturient cum condimentum viverra fringilla, praesent egestas litora torquent, diam nullam sapien tincidunt. Etiam venenatis massa maecenas vehicula varius vestibulum donec a rhoncus, quisque primis inceptos fringilla sagittis erat hendrerit vulputate iaculis tortor, dignissim luctus taciti himenaeos aenean mauris scelerisque potenti. Himenaeos montes magna fusce justo ornare scelerisque condimentum, hendrerit duis sagittis cubilia fermentum odio nam, risus erat ligula natoque ut orci, nullam et taciti praesent curae dis.

Figure 5.4.5. Solution manual (Pop-up)



Booking

Search by call ID

A01781663
A01736253
A09564732
A95745212
A01293746
A03923464
A04937724
A04938443
A01998998
A01765889
A01746337

Figure 5.4.6. Solution manual - Transcript Search (Pop-up)

5.5 Software Interfaces

To interact with the Amplify services such as Authentication (Cognito), API (DynamoDB and AppSync) and Storage (S3) the frontend client will use Amplify's npm package (with the Next JS adapter and SSR configuration). Amazon Connect APIs will be handled in the backend including but not limited to the Next JS API routes.

For custom microservices the *Gateway Aggregator pattern* will be the preferred approach, so that the microservices themselves are hidden from the client and for this last one to use a single source of truth. However, to avoid a *single point* of failure or the introduction of bottlenecks, the API Gateway should follow a resilient design and should be load tested.

5.6 Hardware Interfaces

The web application runs over the internet, so the hardware used requires a connection to the internet (e.g. modem, WAN-LAN, Ethernet or WiFi NIC). Also, the hardware used by the agents will be a tablet, so the system needs to be compatible with these devices but also in a computer (for the supervisors).

5.7 Communications Interfaces

The system will use *HTTP* protocol for interaction with the internet and protocols TCP/IP will also be used, HTTP/2 connections are encouraged due to their ability to multiplex. The main form of communication between services and the frontend will be through JSON, although other standards like *gRPC* might be considered for communication between backend services.

To access the web application, users need an email address to be able to create a user account, it will also work for them to reset their password if they forget it. Cognito user pools store user data in Amazon Cloud Directory which encrypts data at rest and in transit by using 256-bit encryption keys. All the communication of the app will be done utilizing the tools previously mentioned (Amazon Connect, Cognito, Contact Lens etc.).

6. Quality Attributes

6.1 Usability

The User Interface for the system will be designed considering modern *UX/UI* design techniques, principles and trends for accessibility and good standards. Some of these include responsiveness, a minimalistic, yet contrasting approach, and the intention to make the UI as intuitive as possible. The team will follow guidelines like designing so that a user does not have to doubt their actions and, at most times if not at all, it is always clear what to do in the UI. Finally, as a feature that could be included, both dark and light mode designs would be considered to improve User Experience and preferences. Moreover, it is planned that a supervisor can reorder or reorganize certain widgets and functionality in their dashboard, allowing for a more customized experience.

6.2 Performance

One requirement of the system that has been repeatedly emphasized is the usage of *Real Time* data and therefore the response time is bounded, although such boundary is not necessarily standardized. Since data transmission is constrained by physical limitations, such as the location of the servers, the speed of light and the category of ethernet cables, and given the fact that computers need to process the data, there must be a delay between the events, the responses sent by the servers and the time taken for the client to process the response. We shall therefore consider the *Real Time* constraint to indicate that services must communicate using data streams with around 300 ms of delay.

6.3 Security

User authentication in this system would be mainly handled and managed by Amazon Cognito, so all security concerns regarding users and authentication are delegated to this service. Front-end code should contain no API keys, secrets, or any other type of data that could potentially give access to our sensitive data to undesired users. These keys and secrets, if any, should be stored in our API deployments, preferably as environment variables, in order to prevent them from being leaked to browsers and thus to anyone.

In the case of our data storages, user access and permissions should be limited only to people that are authorized and working on those specific services, and permissions should be meticulously set. Assuming there is sensitive customer data, it should not be exposed and all of the team members with access to it should be forced to sign an *NDA* or a document of the sort to prevent from undesired leaks.

6.4 Safety

Callers and CC employees will be notified that the call will be recorded for performance and service improvement, so that they are all aware of this. Moreover, callers will be notified where they can read the company's Privacy Statement and related documents, therefore it is the responsibility of the CC's employees to inform their end users about data retention. We must also ensure the safety of the users accounts within the application, implying a secure connection with the APIs and any extra microservices, hence some DynamoDB fields will be protected based on Cognito user groups.

7. Internationalization and Localization Requirements

The system's codebase, user interfaces, documentation, request parameters and responses will be exclusively implemented in (American) English and therefore formatting will follow the corresponding conventions.

Appendix A: Glossary

API (Application Programming Interface): It acts like a messenger, allowing applications to request and exchange data in a structured way.

Amazon Web Services (AWS): Amazon's Cloud Services and Infrastructure, which ranges from storage services to website hosting, analytics and much more. See more at: [AWS (aws.amazon.com)]

Contact Center (CC): References the hypothetical Call or Contact Center for which this software system will be developed.

gRPC: An open-source framework for efficient and reliable communication between applications, which uses *Proto Buffers* to speed and standardize message transmissions.

HTTP: One of the most standard protocols that serves as the foundation of web communication, acting as a request-response standard between browsers and servers.

NDA: Non-Disclosure Agreement, legal contract that protects confidential information shared between parties. It restricts who can access the information and what they can do with it.

PII (Personal Identifiable Information): Any representation of information that permits the identity of an individual to whom the information applies to be reasonably inferred by either direct or indirect means.

SDLC: Software Delivery LifeCycle. The process of planning, creating, testing, deploying, and maintaining software applications, ensuring efficient and high-quality development from inception to retirement.

Sentiment Analysis: the report that is made given a speech to text analysis regarding the "mood" or "outcome" of the call.

UX/UI: User Experience and User Interface. UX is about how a product works and feels for the user. UI is about what the user sees and interacts with.