Project Design Phase-I
Solution Architecture

Date	09 May 2023
Team ID	NM2023TMID02778
Project Name	Identifying Airline Passenger Satisfaction Using Machine Learning

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- **Find the best tech solution to solve existing business problems.**

  One of the key challenges faced by airlines is how to personalize the passenger experience to meet individual needs and preferences. Machine learning algorithms can be used to analyze passenger data such as booking history, flight preferences, and feedback to provide personalized recommendations for in-flight services, meals, and entertainment.

  Airlines need to capture real-time feedback from passengers to identify areas for improvement and address any issues that arise during the flight. Machine learning algorithms can be used to analyze passenger feedback in real-time and provide automated responses or alerts to airline staff.

  Airlines need to ensure that their planes are in good condition and avoid any unexpected maintenance issues that can cause delays or cancellations. Machine learning algorithms can be used to analyze data from sensors and other sources to predict when maintenance issues are likely to occur, allowing airlines to schedule maintenance proactively.

  Airlines need to optimize their flight schedules based on demand to maximize revenue and minimize costs. Machine learning algorithms can be used to analyze historical data and predict future demand, allowing airlines to optimize their schedules and pricing accordingly.

Airlines receive a large volume of customer inquiries and complaints, which can be time-consuming to handle manually. Machine learning algorithms can be used to automate customer service responses and provide personalized recommendations to customers.

- **Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.**

  The software is designed as a modular system that consists of several components, including data collection, data analysis, machine learning models, and recommendation engines. These components work together to provide personalized recommendations to passengers based on their preferences and feedback.

  The software is characterized by its ability to process large amounts of data from multiple sources, including booking history, flight preferences, and feedback. It uses machine learning algorithms to analyze this data and provide personalized recommendations to passengers in real-time.

  The software is designed to run continuously in the background, processing data and analyzing passenger feedback in real-time. It provides automated responses or alerts to airline staff when issues arise, allowing them to address issues proactively.

  The software provides a user interface that allows airline staff to view passenger feedback and recommendations in real-time. It also provides tools for monitoring the performance of the machine learning models and making adjustments as needed.

  The software is designed with security in mind and includes features such as encryption, access controls, and audit logs to ensure the confidentiality and integrity of passenger data.

  The software is designed to be scalable, allowing it to handle large amounts of data from multiple airlines and airports. It can be deployed in a cloud environment, allowing it to scale horizontally and vertically as needed.

- **Define features, development phases, and solution requirements**.

Features:

Personalization: The software should be able to analyze passenger data to provide personalized recommendations for in-flight services, meals, and entertainment.

Real-time feedback: The software should be able to capture real-time feedback from passengers and provide automated responses or alerts to airline staff.

Predictive maintenance: The software should be able to predict maintenance issues and schedule maintenance proactively.

Demand forecasting: The software should be able to analyze historical data and predict future demand to optimize flight schedules and pricing.

Customer service automation: The software should be able to automate customer service responses and provide personalized recommendations to customers.

Development Phases:

Requirement analysis: Identify the key requirements of the software, including data sources, machine learning models, user interface, and security.

Data collection and processing: Collect and process data from multiple sources, including booking history, flight preferences, and feedback.

Machine learning model development: Develop machine learning models to analyze data and provide personalized recommendations to passengers.

User interface development: Develop a user interface for airline staff to monitor passenger feedback and recommendations.

Testing and deployment: Test the software to ensure it meets requirements and deploy it in a cloud environment.

Solution Requirements:

Data sources: The software should be able to collect and process data from multiple sources, including booking history, flight preferences, and feedback.

Machine learning models: The software should be able to develop and deploy machine learning models that can analyze data and provide personalized recommendations to passengers.

User interface: The software should provide a user interface for airline staff to monitor passenger feedback and recommendations.

Security: The software should include features such as encryption, access controls, and audit logs to ensure the confidentiality and integrity of passenger data.

Scalability: The software should be able to handle large amounts of data from multiple airlines and airports and scale horizontally and vertically as needed.

Integration: The software should be able to integrate with existing airline systems, such as reservation and booking systems, to provide a seamless experience for passengers.

Performance: The software should be able to process data and provide recommendations in real-time to ensure a smooth and efficient passenger experience.

- **Provide specifications according to which the solution is defined, managed, and delivered.**

Requirements specification: Define the key requirements of the software, including data sources, machine learning models, user interface, and security.

Project management: Use an agile project management methodology to manage the development process, including iterative development, frequent testing, and regular feedback from stakeholders.

Data management: Implement a robust data management system to collect, process, and store large amounts of data from multiple sources. This should include data cleaning, data normalization, and data transformation to ensure data quality.

Machine learning models: Develop and deploy machine learning models that can analyze data and provide personalized recommendations to passengers. This should include selecting appropriate algorithms, feature engineering, model training, and hyperparameter tuning.

User interface design: Develop a user interface for airline staff to monitor passenger feedback and recommendations. This should include usability testing, user experience design, and user interface development.

Security: Implement appropriate security measures to protect passenger data, including encryption, access controls, and audit logs. This should comply with relevant data privacy and security regulations.

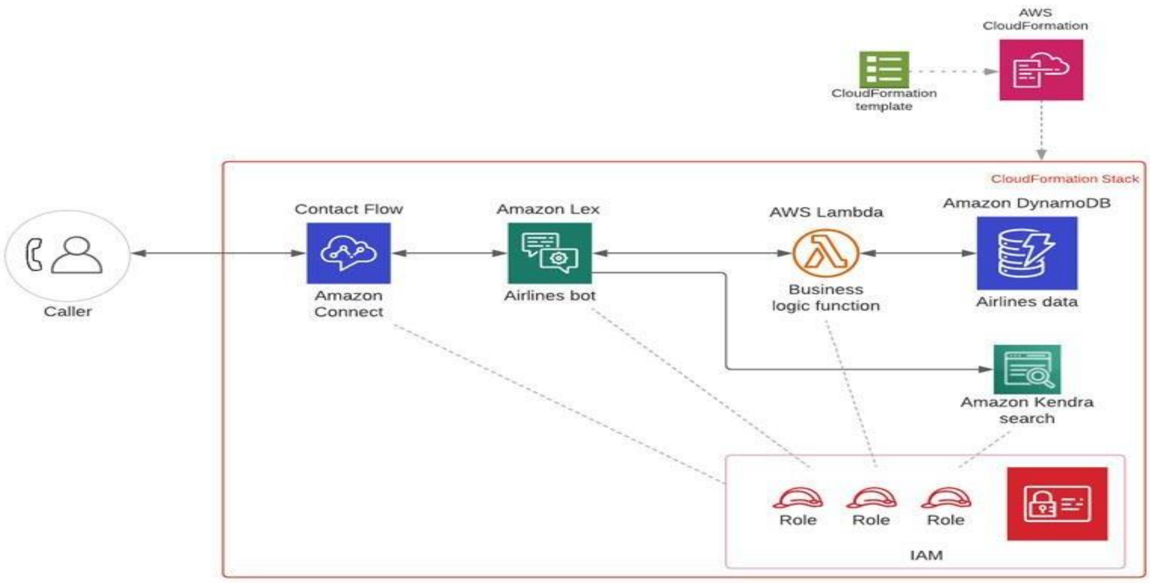
Performance testing: Conduct performance testing to ensure that the software can process data and provide recommendations in real-time, and can handle large amounts of data from multiple sources.

Integration: Ensure that the software can integrate with existing airline systems, such as reservation and booking systems, to provide a seamless experience for passengers.

Deployment: Deploy the software in a cloud environment that can scale horizontally and vertically as needed. This should include deployment testing, configuration management, and release management.

Maintenance and support: Provide ongoing maintenance and support for the software, including bug fixes, performance tuning, and feature enhancements. This should include a helpdesk, service level agreements, and regular software updates.

Solution Architecture Diagram:


AWS
CloudFormation
CloudFormation
template
CloudFormation Stack
Caller
Contact Flow
Amazon
Connect
Amazon Lex
Airlines bot
AWS Lambda
Business
logic function
Amazon DynamoDB
Airlines data
Amazon Kendra
search
Role
Role
Role
IAM