Use Cases and Logical Architecture

* X00141559
* Aoife Bergin
* Auto Loan Bot

## Section 1:

## Use Cases:

|  |  |
| --- | --- |
| Title (goal) | Ask a Question about an Auto loan |
| Primary Actor | Loan Applicant/Bot |
| Story | * The applicant will ask the bot a question, either typed by the applicant or picked from the pre-defined menu list. * The bot can handle questions such as: * What is an ARP? * What is the term? * What type of car loan do you offer? * Who is eligible to apply for an auto loan? * If I decide to pay my auto loan off early is there a pre-payment penalty? * Can I get approved for an auto loan before I pick out my car? * The bot will respond with an answer. * The applicant can continue to ask questions or choose to leave the session. |

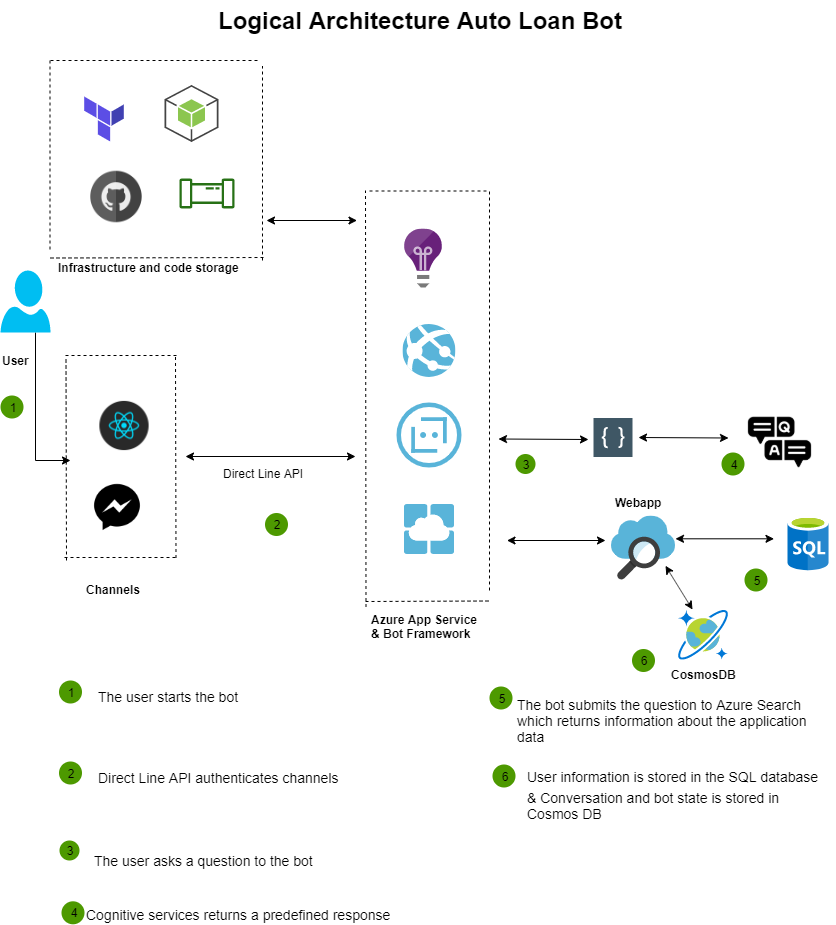
|  |  |
| --- | --- |
| Title (goal) | Get an Auto Loan Quotation |
| Primary Actor | Loan Applicant/Bot |
| Story | * The user requests an auto loan quote. * The bot responds with questions necessary to gather the information to calculate a quote. * The information the bot will request to calculate the loan quote:  1. Loan Type -There are four types of auto loans   available:   1. Dealer Purchase 2. Refinance 3. Lease Buyout 4. Private Party 5. Loan Amount – Must not be less than $7,500. 6. Loan Term Length – There are three loan terms lengths available: 7. 48 Months 8. 60 Months 9. 72 Months  * The bot will then make the calculations using the information given and the current interest rates. * The bot will make a calculation and respond to let the user know the total cost of finance and how much the monthly repayment will be. |

|  |  |
| --- | --- |
| Title (goal) | Proceed with Quotation |
| Primary Actor | Loan Applicant/Bot |
| Story | * After calculating the loan quotation and displaying the total cost of finance and the amount for monthly repayment to the applicant, the bot will ask the user if they would like to proceed with the application. * If the applicant agrees to proceed, the bot will ask them to input their email address and phone number so that a loan specialist can get in touch via phone call to proceed with the loan application. * The bot thanks the customer for their time and displays a farewell greeting   Alternative: step 2, if the customer wishes not to proceed  Bot displays ‘Do you wish to have a quote sent by email’ and asks the user to enter their email address. |

|  |  |
| --- | --- |
| Title (goal) | Initiate Chat |
| Primary Actor | Loan Applicant/Bot |
| Story | * Upon activating the bot will display a welcome message and question menu to the user. * The bot will display a menu of pre-defined questions and prompts the user to pick one or type their own. * The user chooses from the menu card list or types a question. * The bot will query the database and respond with the appropriate answer. |

|  |  |
| --- | --- |
| Title (goal) | Terminate Chat |
| Primary Actor | Loan Applicant/Bot |
| Story | * The user can type ‘stop’ at any stage. * The bot will respond with a farewell greeting and terminate the session immediately. |

## Section 2: Logical Architecture



## Logical Architecture Discussion

**Code Storage**

GitHub will be used to store the code and documents relating to this project.

**Pipelines and Testing**

Azure Pipelines will be configured for CI/CD to push the code to the GitHub repo, build and run unit tests.

**Channels**

Channels for this project will be Facebook Messenger and a React dashboard, which can be implemented via access tokens.

**Connecting the bot to channels**

Direct Line API is an Azure service that allows for connection of the bot to multiple different channels by using secret keys and tokens, this project will connect to the Facebook Messenger channel via Direct Line API.

**Cognitive Services:**

This project will leverage Azures cognitive services:

* **LUIS**

LUIS identifies user goals (intents) from entities (sentences) to build an intelligent language model. LUIS uses reinforcement learning so the model is constantly learning and improving.

* **QnA Maker Service**

This service allows you to publish a set of question and answers to a knowledge base. The client application will reach out to your QnA endpoint to retrieve the best answer to the users question. The question that is returned is handled by QnA Maker.

This allows for multiturn prompts and connection of question and answer pairs. When QnA receives questions from users it will apply active learning so that it can better suggest answers in the future.

**Azure Bot Framework**

Bot Framework is an open source SDK for bot development. Azure bot service helps to manage, deploy and connect the bot. This project will use a Yeoman template to generate the bot and deploy the code to an azure web app.

**The Bot Connector**

The Bot Connector service uses REST, JSON and HTTPS to exchange messages with channels. A token is needed from the Bot Framework to enable the bot connector service to send and receive messages.

**Bot Framework SDK for Node.js**

Bot Framework SDK for Node.js provides a good platform for Node.js developers to write bots. It can help to build UI interfaces. This service utilises restify which is a framework for building web services, in this project it will serve to create the bots web server.

**Bot Emulator**

Bot emulator allows for a bot to be run locally on your machine before deploying to Azure. This project will run the bot locally with the bot emulator.

**Azure SQL Database**

Azure SQL Database is a cloud based service that offers a high level of protection for user data. This project will use Azure SQL Database to store users personal data along with their auto loan quotations.

**Cosmos DB**

Cosmos DB is a NoSQL database that offers elastic storage, high throughput and low latency. In this project Cosmos DB will serve to store the conversations with the bot and the bot state properties. This was chosen as it is schema less model, this project will use the core API SQL for querying the database.

**Terraform**

This project will create the Cosmos DB and Azure SQL database leveraging infrastructure as code tool terraform.

**Application Insights**

This is a feature of Azure Monitor, it can detect performance anomalies in an application and give insights as to how the users interact with your application, this project will contain application insights to log user interaction.

**Azure Search**

The Bot submits the query to Azure Search, it reaches out to the SQL database and then returns information about the application data.

**Auto Loans and Formula**

There are many types of loans available in retail banking, this project will focus exclusively on auto loans. Auto loans are amortizing loans which means they are paid back monthly in equal sums. The interest rates will differ from bank to bank, the bank that this project is based on also offers a discount of up to 50% on the APR if applicants are part of the rewards club.

There are three tiers in the rewards club:

1. Gold – 25% discount on APR
2. Platinum – 35% discount on APR
3. Platinum Honours- 50% discount on APR

There will be four types of car loans on can offer:

1. Dealer Purchase (rates may differ on new or used car)
2. Refinance
3. Lease Buyout
4. Private Party

There are many terms and conditions when you are applying for an auto loan, the bot will display the terms and conditions to the user when they click proceed with loan.

Some of which being:

* Applicant must be over 18 years old.
* The car should have less than 125,000 miles on it.
* The car should not be more than ten calendar years old.
* Minimum loan amount is $7,500.

### 

### **Formula for Auto Loan Payment:**

**Loan Payment = Amount / Discount Factor** or **P = A / D**

* We need the following values:
* Number of Periodic Payments (**n**) = Payments per year times number of years
* Periodic Interest Rate (**i**) = Annual rate divided by number of payment periods
* Discount Factor (**D**) = {[(1 + i) ^n] - 1} / [i(1 + i)^n]