University of Arkansas – CSCE Department
Capstone I – Preliminary Proposal –Spring 2019

Carrier 360 Mobile Virtual Assistant

Caleb Fritz, Brandon Cox, Matthew Sij, Alicia Gillum, Anjan Poudel

Abstract

Carrier 360 Mobile Virtual Assistant

Currently, the J.B. Hunt Carrier 360 Mobile Application does not support voice interaction. Due to safety protocols, the application will lock down when the drivers are on the road. Our overall goal is to implement voice support for the existing J.B Hunt Carrier 360 Mobile Application so that it can be safely used while the drivers are on the road. We are approaching the problem with a client-server model in order to conserve client resources and to implement a solution using Microsoft's machine-learning language understanding service: LUIS. We will feed J.B Hunt endpoints with parsed information provided to us from LUIS to provide language support for the application.

1.0 Problem

The current process for booking new loads through the J.B. Hunt Carrier 360 Mobile Application requires that the driver manually enter key information throughout the booking and bidding process. Because of the inherent danger of interacting with mobile devices and applications, drivers must be stationary to search, select, or bid on a new job. This is a huge pain point for current and future users of the J.B. Hunt Carrier 360 Mobile Application, which aims to modernize freight services.

This stationary requirement causes drivers to stop and interact with the J.B. Hunt Carrier 360 Mobile Application. Drivers must maintain a delivery schedule, and winning bids is a time consuming and time sensitive process. This can cause drivers to either hurt their delivery schedule, or potentially miss out on lucrative bids because they are unable to frequently pullover.

2.0 Objective

The objective of this project is to save time for the drivers and add user friendliness to the Carrier 360 App. Because of easiness in controlling a device with a natural language rather than interacting with the screen, virtual assistance has become a crucial need of clients today to cope with quickly growing technology. Adding virtual assistance features in Carrier 360 Mobile mobile application will allow drivers to search for and book loads without even touching their device. It will enable drivers to safely command the app to find a load while they are behind the wheel.

3.0 Background

3.1 Key Concepts

In its current state, the application locks itself into a safety mode when the driver gets on the road. We will be using a frontend React Native client to serve as the replacement for this safety state. The React Native Framework allows us to design a cross-platform application with ease by using its seamless native code integration. We will be using Microsoft's Azure Web App Service platform to host our application. This platform will provide our application with access to the Azure Bot Service framework and Microsoft's Language Understanding Intelligent Service (LUIS). The Azure Bot Service framework will receive utterances sent by the client and will send them to LUIS to be processed. LUIS and the Azure Bot Service framework will interact with each other to properly assess the client's utterances. LUIS will translate the utterance into intentions and track down the entities they affect. These entities are components that we expect to be included in an utterance specific to the task at hand. They are parameters that we will set in the training process for our application. We will also incorporate Firebase Analytics to gather information on how users interact with the app. This information incorporated with the analytics gathered by the Azure Bot Service framework will be useful in further improving the application.

3.2 Related Work

J.B. Hunt hosts popular hackathons at the company headquarters in Lowell, Arkansas, where participating employees are given 24 hours to make their project ideas come to life. Several members of the Carrier 360 Mobile team engaged in the Spring 2019 hackathon with the idea of providing voice functionality for the app. The team was successful in their implementation, using Google's Dialogflow natural language understanding system to process user utterances, however, due to the time constraints, the full idea and functionality of their implementation could not be fully realized. This project hopes to take what they have learned and innovate on the idea while bringing new technology, like Microsoft LUIS and the Azure Bot Service into the project's scope.

4.0 Design

Several use cases will be fulfilled by the development of this application, allowing a user to perform the following tasks with their voice commands:

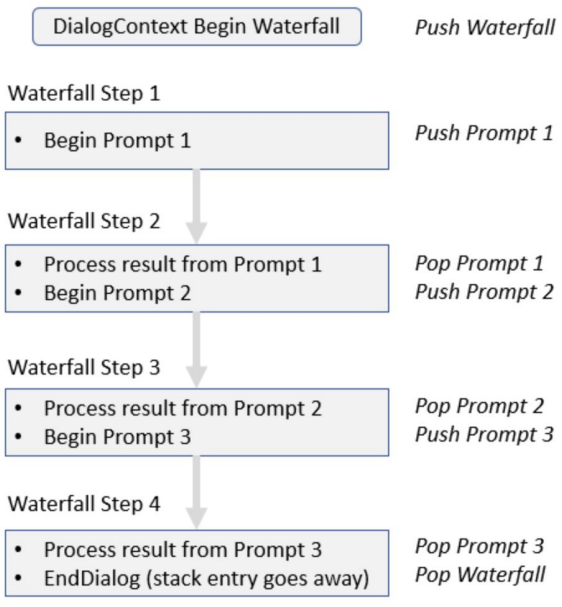
1. Users will be able to find loads from a source location to a destination location
    a. if a source location is not specified, the current location will be used
2. Users will be able to specify an equipment type when searching for a load
    a. if no equipment type is specified, all equipment types will be included in the search
    b. Equipment types include:
        i. Dry Van
        ii. Flatbed
        iii. Reefer
        iv. Power Only

3. Users will be able to specify the number of empty miles they are willing to travel for a load
    a. If unspecified, a default of 200 miles will be included implicitly
4. Users may specify a load number to search rather than specifying search parameters
5. Users can decide to place an offer on a specific load, book the load immediately (if applicable), or view more load details
6. Upon asking for more details on a load, users may specify to hear the following details
    a. Details for each stop provided by the load, including location and appointment date/time
    b. Load information like total loaded miles, payload weight, and equipment type

4.2 High Level Architecture

This application will allow carriers and their drivers within the integrated capacity solutions (ICS) segment to perform various actions using voice commands safely while driving their vehicles. The application will expose some of the current Carrier 360 Mobile app features to users through a intuitive voice interface that will grant them the ability to use the app with as little physical contact as possible, allowing drivers to focus on the road ahead. Currently, the Carrier 360 Mobile app will activate a safety feature that will shutdown usability when it detects the user is in movement, forcing them to interface with the app while stationary. Our application will allow interfacing while in movement, allowing users to perform their logistical tasks within the applications in a safe and effective way, saving them time and boosting productivity.

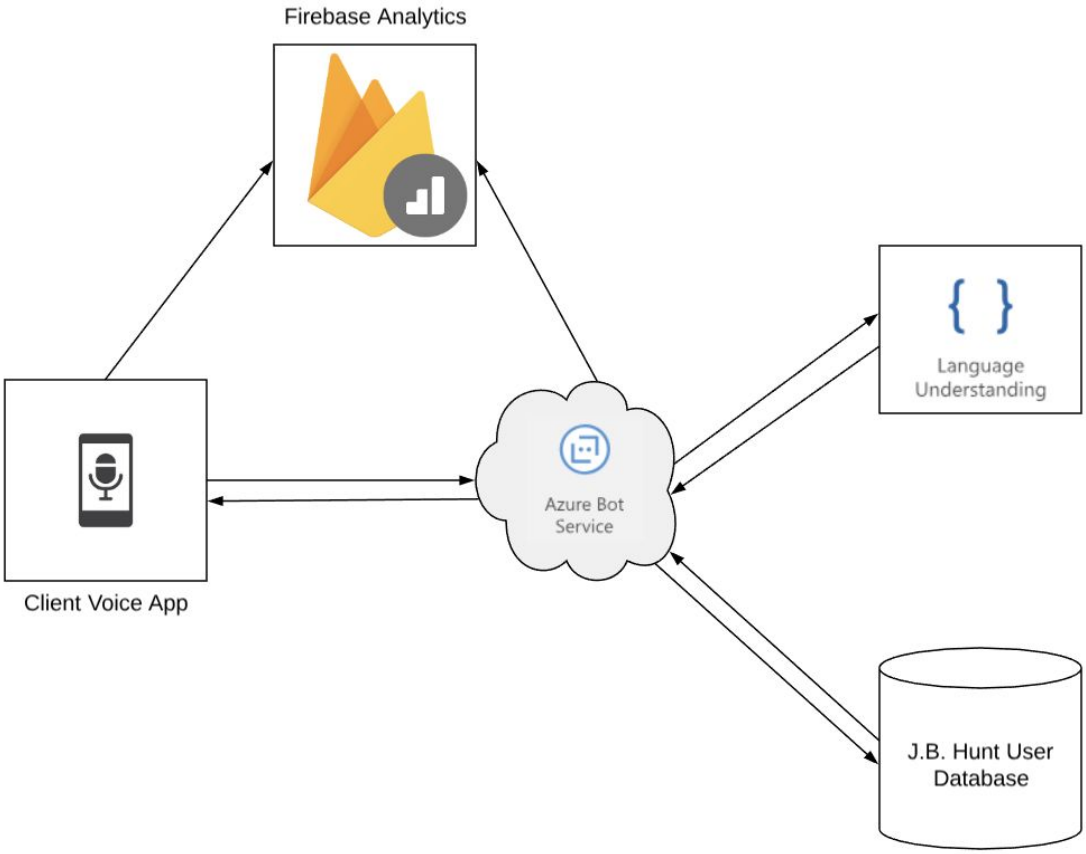
To achieve these goals, a frontend React Native application will be developed to replace the current safety screen, allowing users to interact with the Carrier 360 Mobile app while driving. This frontend piece will implement functionality that will capture user utterances and pass them to an endpoint provided by a backend Node.js server. User utterances will be captured using third-party modules either provided by the popular package manager NPM or by using voice capture cloud services. Once the backend has processed a user request and produces a response string, it will pass it back to the client, and the client will relay the response using a text to speech module.



The backend Node.js server will be hosted via Microsoft's Azure Web App Service platform and will process user utterances using the Azure Bot Service framework and Microsoft Language Understanding Intelligent Service (LUIS). The bot service will give granular control over the design of conversational flow, providing support for prompting users for input, defining question/answer sequences, modularizing conversational flow logic, and handling interruptions and fallback scenarios [1]. To understand the intentions from user utterances, the LUIS API will process the utterance using advanced natural language

understanding algorithms to extract meaningful information like keywords, which will then be passed back to the bot service, providing the next logical step in the conversational flow [2]. To fulfill user requests, J.B. Hunt endpoints will be used to fetch the requested information from users and relay that information through the client app.

To understand how this voice interface is used and how it can be improved, analytics will be implemented. Currently, the Carrier 360 Mobile app employs Firebase Analytics to understand how users interact with the app, thus for easier integration into the main application, Firebase Analytics will also be implemented into the voice app. The Azure Bot Service provides a variety of solutions for implementing Analytics as well, and may be implemented to provide even more feedback.



4.3 Risks

Risk	Risk Reduction
Restricted Access to J.B. Hunt endpoints disallows full testing of voice app for most team members	Use Tundra-Fetch endpoint capture software to generate data profile fixtures with PII attributes removed/obfuscated to mock services
Sensitive API keys and endpoints will be used within the client and server to perform operations, posing potential security risk	Employ best security practices by keeping sensitive resources within environment files and use them securely, followed by extensive testing
App usage while driving could pose legal questions	Design the application within the scope of State and Federal laws and follow requirements imposed by J.B. Hunt for optimal safety

4.4 Tasks – The following tasks are to be completed for this project:

1. Gain background knowledge of the required frameworks and APIs for the project and setup project environment
    a. Understand React-Native framework and the following libraries/middleware
        i. React-Redux for application state management
        ii. Redux-Thunk (potentially Redux-Saga) for async action handling
        iii. react-native-voice for converting user speech to text
        iv. react-native-tts for converting text to speech for relaying messages from the server to the user
        v. ESLint for code styling rules
        vi. NativeBase for basic, customizable components
        vii. node-fetch for http requests
    b. Understand LUIS to build/train a language agent to extract useful text entities from identified intents produced from user utterances
    c. Understand Azure Bot Framework SDK to design conversation flow modules
    d. Understand Firebase Analytics SDK to integrate analytics into project
    e. Create a project repository using Azure DevOps and establish code rules and PR policies
2. Design client app
    a. Initialize a React Native project and install necessary libraries, establish organizational and code styling guidelines via ESLint and a clear readme for easier collaboration
    b. Design app interface by developing mocks and use case scenarios
    c. Determine the state tree for the app
3. Implement client app design
    a. Develop screen components and style using StyleSheet attributes and components from NativeBase library

   b. Integrate react-native-voice and react-native-tts into project to capture voice data, convert to text, then back to voice data.
   c. Integrate React-Redux into app to manage state
      i. Develop synchronous and asynchronous action creators to dispatch actions
      ii. Develop reducers to capture actions and update app state
   d. Use DirectLine REST API and node-fetch to send user utterances to the Node.js bot server, and receive text responses from the server
4. Design bot server
   a. Initialize a Node.js restify server and integrate Bot Framework Service middleware
   b. Implement prompts and dialog workflows using Bot Framework SDK dialog library
   c. Connect LUIS to the bot service to extract user intents and entities for usage in the dialog workflows
   d. Implement service fulfillment by sending data payloads recovered from collected user information to J.B. Hunt servers, and receiving the response data
   e. Capture response data from fulfillment and package up into a text response to be sent back to the client
5. Design and train LUIS voice agent
   a. Create LUIS app and add app credentials to bot server
   b. Create intents to identify possible user utterances
      i. Develop fallback intents for unknown or irrelevant utterances
   c. Apply intent entities to identify key information within utterances
6. Integrate analytics
   a. Setup Firebase Analytics within Firebase Console
   b. Integrate into client app, sending events to track user interactions
   c. Integrate into bot server, potentially in tandem with bot analytics framework to track service efficacy
7. Deployment, testing, and documentation
   a. Deploy bot server to azure
   b. Test and debug bot server individually using Bot Framework Emulator
   c. Test and debug client app individually using react-native-debugger
   d. Test all possible dialogue workflows with client app and bot server integrated.
      i. Test “happy-path” scenarios, where dialog workflow criteria is successfully met
      ii. Test “sad-path” scenarios, where dialog workflow criteria is not met at various stages, and fallback scenarios successfully handle these situations
   e. Test analytics is detecting app usage and displays meaningful telemetry
   f. Generate documentation for final project

4.5 Schedule – Below is a overall schedule of the tasks to be completed. Notice some tasks will be performed in parallel with others. Completion dates are subject to change depending on progress variations during implementation.

Tasks	Assigned to	Dates
1. Gain background knowledge in frameworks and SDKs	Everyone	8/26 - 9/16
2. Design client app	Alicia, Anjan, Matthew	9/17 - 10/1
3. Implement client app	Alicia, Anjan, Matthew	10/2 - 10/30
4. Design bot server	Caleb, Brandon	9/17 - 10/30
5. Design and train LUIS agent	Caleb, Brandon	9/17 - 10/30
6. Integrate analytics	Everyone	10/31 - 11/14
7. Deploy, test, and document	Everyone	11/15 - 12/13

4.6 Deliverables – The following components will be provided at the completion of this project:

- Website code: The PHP code for the team capstone website
- Non-proprietary JavaScript code for client application
- Non-proprietary JavaScript code for bot service server
- Non-proprietary intent and entity data for LUIS agent
- Final Report

5.0 Key Personnel

Caleb Fritz (Team Lead) - Fritz is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He is engaging in undergraduate research pertaining to Computer Vision driven mobile app automation testing. He has completed relevant courses including Software Engineering, Algorithms, Information Security, and Operating Systems. He is currently an information systems intern at the J.B. Hunt headquarters in Lowell, Arkansas working as a frontend developer and automated test scripter. He will be working on the frontend client app and the backend bot service server, with a specific responsibility for service fulfillment using J.B. Hunt endpoints.

Matthew Sij - Sij is a junior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He is minoring in both Mathematics and Physics. He has completed the following relevant courses: Software Engineering, Information Security, Algorithms, and Artificial Intelligence. He currently works as a research assistant in the

department of Computer Science and Computer Engineering, where he works on Computer Vision and Deep Learning related fields. Sij will be interning at Metova, Inc. as a developer intern this summer. He will be responsible for the client side of the project.

Alicia Gillum – Gillum is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She is minoring in both Mathematics and Physics. She has completed the following relevant courses: Software Engineering, Database Management Systems, Wireless Systems Security, Information Security, and Algorithms. She currently works at the University's Security Operations Center as a security technician. She also works as an undergraduate research assistant in the department of Computer Science and Computer Engineering, where she works on projects related to Cybersecurity. Gillum will be interning with Tyson Foods on the security team during the summer. She will be working on the client side of the project.

Brandon Cox – Cox is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He is minoring in Mathematics. He has completed the following relevant courses: Software Engineering, Operating Systems, Algorithms, Information Security, and Artificial Intelligence. He has worked at Datapath, Inc. as a Developer Intern working as a full-stack developer using ASP.NET. He is currently working at Metova, Inc. as a developer intern as an API developer using ASP.NET. Cox will be responsible for the backend side of the project.

Anjan Poudel - Anjan is a Senior Computer Science student in the Computer Science and Computer Engineering Department at the University of Arkansas. He works as Application Developer at the University of Arkansas IT Services. He has completed following classes Computer Vision, Artificial Intelligence, Design and Analysis of Algorithms, Operating Systems, Database Management Systems, Software Engineering, Information Security. He will be working on the client side of the project.

6.0 Facilities and Equipment

J.B. Hunt endpoints for the Carrier 360 Mobile application will be used to provide the fulfillment pieces from user utterances.

7.0 References

[1] Taylor, J., Iqbal, K., & Berry, I. (n.d.). Dialogs within the Bot Framework SDK - Bot Service. Retrieved from https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-concept-dialog?view=azure-bot-service-4.0

[2] Diberry. (n.d.). What is Language Understanding (LUIS) - Azure Cognitive Services. Retrieved from https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis