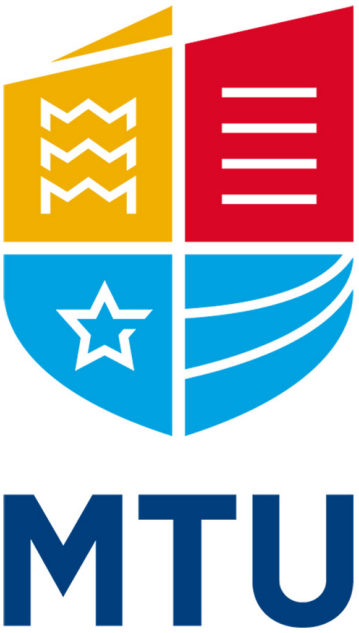

Conversational agent for transportation timetable

by

Waleed Akhtar

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in Software Development

in the
Faculty of Engineering and Science
Department of Computer Science

May 2023

Declaration of Authorship

This report, Conversational agent for transportation timetable, is submitted in partial fulfillment of the requirements of Bachelor of Science in Software Development at Munster Technological University Cork. I, Waleed Akhtar, declare that this thesis titled, Conversational agent for transportation timetable and the work represents substantially the result of my own work except where explicitly indicated in the text. This report may be freely copied and distributed provided the source is explicitly acknowledged. I confirm that:

- This work was done wholly or mainly while in candidature Bachelor of Science in Software Development at Munster Technological University Cork.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Munster Technological University Cork or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

MUNSTER TECHNOLOGICAL UNIVERSITY CORK

Abstract

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Waleed Akhtar

This project aims to create an Amazon Alexa Skills application to assist users in asking bus timetable questions. These questions can be simple such as "When is the next bus to Grand Parade from Victoria Cross?" or complex as "What time I should leave by if I want to arrive in Mallow from Parnell Place by 6 pm?". The application will be able to give users information such as when is the next available bus arriving to take them to their destination along with how long the journey could take. This service is aimed at users in the Republic of Ireland. Users will be able to access this application for free on the Amazon Alexa Skills store. This service will be available to users on their mobile phones, smartwatches, and any Alexa-supported smart home devices.

The reason why this application is being created is because of the long intent to action which is required currently to get information regarding the bus timetables currently. The user is required to search the bus number timetable on search engines such as Google to get this information. This can be a long tedious task especially if the user is not highly IT literate. With this application, the user will be able to receive bus timetable information much quicker and easily by simply asking their Amazon smart assistant.

This application will be made through the use of chatbots. This will allow easy interaction with the user and simplify the whole experience. The application will receive real-time bus information by utilising Transport for Ireland's newest API: GTFS-R. This will provide up-to-date information to the user regarding bus timetables information in Ireland.

Acknowledgements

Firstly I would like to thank my friends in no particular order Mia, Carl, Jack, Oisin, Liam, Evan and Charlotte for being there for me and helping whenever I needed them when I had questions about school work or life in general. Thank you guys for being amazing friends and I wouldn't have made it this far without you guys.

I would also like to thank my family for being there and supporting me in making the career choice to study computers and for helping me to make it this far.

Finally, I would like to thank Alex Vakaloudis who was my supervisor for this project. If it weren't for Alex this report wouldn't be what it is today, he has been a major help with the development of this project.

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

List of Figures viii

List of Tables xii

Abbreviations xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Demand for product . . . 2
- 1.3 Objectives . . . 3
- 1.4 Structure of This Document . . . 3

2 Background 5
- 2.1 Thematic Area within Computer Science . . . 5
- 2.2 Review of smart assistants products available . . . 5
  - 2.2.1 Cortana . . . 5
  - 2.2.2 Siri . . . 6
  - 2.2.3 Google Assistant . . . 6
  - 2.2.4 Alexa . . . 7
- 2.3 Smart Assistant Comparison . . . 8
- 2.4 Data Acquisition . . . 8
  - 2.4.1 PDF Parsing . . . 9
  - 2.4.2 General Transit Feed Specification API . . . 10
- 2.5 Developer Technologies . . . 11
- 2.6 Related Works . . . 12
  - 2.6.1 Bus Times . . . 12
  - 2.6.2 Next Bus . . . 12

3 Problem - Conversational agent for transportation timetable 14

3.1 Objectives . . . 14
3.2 Functional Requirements . . . 14
3.2.1 Product Backlog . . . 15
3.3 High level architecture . . . 16
3.3.1 Data Flow Diagram . . . 16
3.4 Potential Risks . . . 16
3.4.1 Become too over-loaded with assignments . . . 16
3.4.2 Restricted by GTFS API . . . 17
3.4.3 Failing Alexa Skill Validation . . . 17
3.4.4 Amazon redesigning Alexa to be less user friendly . . . 17

4 Implementation **19**
4.1 Difficulties Encountered . . . 19
4.1.1 API Limitations . . . 19
4.1.2 API Update . . . 20
4.1.3 AWS Charges . . . 21
4.1.4 Full Bus Stop Name requirement . . . 22
4.1.5 Difficulty in understanding AWS Terminology . . . 22
4.2 Resolving queries . . . 23
4.2.1 Finding Bus Number . . . 23
4.2.2 Next Bus to Location . . . 24
4.2.3 Arrive by Time . . . 25
4.2.4 Depart by Time . . . 25
4.2.5 Journey time calculation . . . 26
4.3 Dataflow Diagram . . . 27
4.3.1 RDS Database . . . 27
Comparing different architectures . . . 27
4.3.2 Lambda function . . . 28
4.3.3 CloudWatch . . . 29
4.4 Limitations & Assumptions . . . 29
4.4.1 Full Bus Stop Name . . . 29
4.4.2 Only support for Bus Éireann . . . 30
4.4.3 Journey time intent limitation . . . 30
4.5 Privacy Policy . . . 30
4.6 Project management with Jira . . . 31
4.6.1 Learn how to use Alexa SDK . . . 31
4.6.2 Set up development environment . . . 32
4.6.3 Register for TFI GTFS-Realtime API . . . 32
4.6.4 Register as Amazon Developer . . . 33
4.6.5 Get a response from GTFS API . . . 33
4.6.6 Design Alexa Skill . . . 34
4.6.7 Define user intent . . . 34
4.6.8 Define utterances . . . 35
4.6.9 Connect TFI GTFS-Realtime API with Alexa Skill . . . 35
4.6.10 Connect Alexa Skill and AWS Lambda . . . 36
4.6.11 Investigate how AWS Lambda works . . . 36
4.6.12 Define Slots . . . 37

4.6.13 Investigate new version 2 API Changes . . . 37
4.6.14 Modify the application to accommodate new API changes . . . 38
4.6.15 Learn how the code in Alexa Development Console works . . . 38
4.6.16 Investigate the difference between Python JSON query vs DynamoDB . . . 39
4.6.17 Create a proof of concept for Python JSON query . . . 40
4.6.18 Create RDS Database . . . 40
4.6.19 Create EC2 Instance . . . 40
4.6.20 Connect RDS Database with EC2 . . . 41
4.6.21 Create schema for RDS Database . . . 41
4.6.22 Connect RDS Database to MySQL Workbench . . . 42
4.6.23 Create AWS Lambda to call the API . . . 42
4.6.24 Connect RDS Database to Alexa Lambda Function . . . 43
4.6.25 Investigate VPC connectivity issues . . . 43
4.6.26 Create CloudWatch Scheduler to call the API . . . 44
4.6.27 Create IAM Roles . . . 44
4.6.28 Create SQL query to find bus number . . . 45
4.6.29 Connect Bus Number Intent in Alexa Development Console . . . 45
4.6.30 Create SQL query for next bus to area . . . 46
4.6.31 Connect SQL Query for the next bus to an area in Alexa Development Console . . . 46
4.6.32 Configure Speech to Text . . . 47
4.6.33 Configure Text to Speech . . . 47
4.6.34 Investigate Billing Charges . . . 48
4.6.35 Call AWS Billing Support . . . 48
4.6.36 Make changes to AWS to comply with free tier . . . 49
4.6.37 Rebuild RDS Database . . . 49
4.6.38 Update references to use new RDS Database . . . 50
4.6.39 Create SQL query for arriving by bus to an area by a certain time 50
4.6.40 Connect SQL Query for the arriving by to an area intent in Alexa Development Console . . . 51
4.6.41 Create SQL query for when to depart by . . . 51
4.6.42 Connect SQL Query for the departing by to an area intent in Alexa Development Console . . . 52
4.6.43 Create SQL query to calculate the time difference between two bus stops . . . 52
4.6.44 Create poster . . . 53
4.6.45 Make changes to poster . . . 53
4.6.46 Connect SQL Query to calculate the time difference between two bus stops intent in Alexa Development Console . . . 54
4.6.47 Make happy path interaction model . . . 54
4.6.48 Make exception path interaction model . . . 55
4.6.49 Transition development environment to production environment . 55
4.6.50 Beta test Alexa Skill . . . 56
4.6.51 Ensure Skill meets Amazon's Security requirements . . . 56
4.6.52 Ensure Skill meets Amazon's Policy requirements . . . 57
4.6.53 Create privacy policy . . . 57

4.6.54 Create Skill Icon . . . 58
4.6.55 Submit Skill for validation . . . 58
4.6.56 Publish Alexa Skill . . . 59
4.6.57 Make changes required by Alexa publication team . . . 59
4.6.58 Write Chapter 4 implementation phase . . . 60
4.6.59 Write Chapter 5 evaluation . . . 60
4.6.60 Write Chapter 6 conclusions . . . 61

5 Testing and Evaluation 62
5.1 Finding Bus Number Intent . . . 62
5.2 Next Bus to Location Intent . . . 64
5.3 Arrive by time Intent . . . 65
5.4 Depart by time Intent . . . 66
5.5 Journey time calculation Intent . . . 67
5.6 Metrics . . . 68
5.7 Objectives fulfilment . . . 69
5.7.1 Objective 1 . . . 69
5.7.2 Objective 2 . . . 69
5.7.3 Objective 3 . . . 69
5.7.4 Objective 4 . . . 69
5.7.5 Objective 5 . . . 69
5.7.6 Objective 6 . . . 70
5.7.7 Objective 7 . . . 70
5.7.8 Objective 8 . . . 70

6 Discussion and Conclusions 71
6.1 Solution Review . . . 71
6.2 Project Review . . . 71
6.3 Lessons Learnt . . . 72
6.3.1 AWS . . . 72
6.3.2 Alexa SDK . . . 73
6.3.3 Project Management Tool . . . 73
6.3.4 API . . . 73
6.3.5 Python . . . 73
6.3.6 Risk analysis . . . 73
6.4 Future Work . . . 74
6.5 Conclusion . . . 74

Bibliography 76

List of Figures

1.1 A screenshot from Google Trends 2

2.1 A screenshot of the 205 bus timetable in Cork 9

2.2 A screenshot of output from the GTFS API 11

3.1 A screenshot of my backlog from Jira 15

3.2 A screenshot of my Data flow diagram 16

4.1 A screenshot of HTTP Error 20

4.2 A screenshot of several Wilton Bus Stops 22

4.3 A screenshot of Amazon store page 23

4.4 A screenshot of find bus number intent 24

4.5 A screenshot of Next Bus to Location intent 24

4.6 A screenshot of Arrive by Time bus intent 25

4.7 A screenshot of Depart by Time bus intent 26

4.8 A screenshot of Journey time calculation intent 26

4.9 A screenshot of a Dataflow Diagram 27

4.10 A screenshot of GTFS Dashboard 28

4.11 A screenshot of the full bus stop name 29

4.12 A screenshot from the GTFS Dataset 30

4.13 A screenshot of Story: ABT 1 from Jira 32

4.14 A screenshot of Story: ABT 7 from Jira 32

4.15 A screenshot of Story: ABT 13 from Jira 33

4.16 A screenshot of Story: ABT 14 from Jira 33

4.17 A screenshot of Story: ABT 26 from Jira 34

4.18 A screenshot of Story: ABT 2 from Jira 34

4.19 A screenshot of Story: ABT 9 from Jira 35

4.20 A screenshot of Story: ABT 10 from Jira 35

4.21 A screenshot of Story: ABT 3 from Jira 36

4.22 A screenshot of Story: ABT 34 from Jira 36

4.23 A screenshot of Story: ABT 33 from Jira 37

4.24 A screenshot of Story: ABT 11 from Jira 37

4.25 A screenshot of Story: ABT 30 from Jira 38

4.26 A screenshot of Story: ABT 31 from Jira 38

4.27 A screenshot of Story: ABT 32 from Jira 39

4.28 A screenshot of Story: ABT 35 from Jira 39

4.29 A screenshot of Story: ABT 36 from Jira 40

4.30 A screenshot of Story: ABT 37 from Jira 40

4.31 A screenshot of Story: ABT 38 from Jira 41

4.32 A screenshot of Story: ABT 39 from Jira 41

4.33 A screenshot of Story: ABT 42 from Jira 42

4.34 A screenshot of Story: ABT 60 from Jira 42

4.35 A screenshot of Story: ABT 61 from Jira 43

4.36 A screenshot of Story: ABT 41 from Jira 43

4.37 A screenshot of Story: ABT 67 from Jira 44

4.38 A screenshot of Story: ABT 62 from Jira 44

4.39 A screenshot of Story: ABT 40 from Jira 45

4.40 A screenshot of Story: ABT 43 from Jira 45

4.41 A screenshot of Story: ABT 45 from Jira 46

4.42 A screenshot of Story: ABT 50 from Jira 46

4.43 A screenshot of Story: ABT 51 from Jira 47

4.44 A screenshot of Story: ABT 20 from Jira 47

4.45 A screenshot of Story: ABT 21 from Jira 48

4.46 A screenshot of Story: ABT 44 from Jira 48

4.47 A screenshot of Story: ABT 44 from Jira 49

4.48 A screenshot of Story: ABT 46 from Jira 49

4.49 A screenshot of Story: ABT 48 from Jira 50

4.50 A screenshot of Story: ABT 49 from Jira 50

4.51 A screenshot of Story: ABT 52 from Jira 51

4.52 A screenshot of Story: ABT 53 from Jira 51

4.53 A screenshot of Story: ABT 54 from Jira 52

4.54 A screenshot of Story: ABT 55 from Jira 52

4.55 A screenshot of Story: ABT 65 from Jira 53

4.56 A screenshot of Story: ABT 68 from Jira 53

4.57 A screenshot of Story: ABT 69 from Jira 54

4.58 A screenshot of Story: ABT 66 from Jira 54

4.59 A screenshot of Story: ABT 15 from Jira 55

4.60 A screenshot of Story: ABT 16 from Jira 55

4.61 A screenshot of Story: ABT 24 from Jira 56

4.62 A screenshot of Story: ABT 17 from Jira 56

4.63 A screenshot of Story: ABT 22 from Jira 57

4.64 A screenshot of Story: ABT 23 from Jira 57

4.65 A screenshot of Story: ABT 63 from Jira 58

4.66 A screenshot of Story: ABT 58 from Jira 58

4.67 A screenshot of Story: ABT 6 from Jira 59

4.68 A screenshot of Story: ABT 5 from Jira 59

4.69 A screenshot of Story: ABT 64 from Jira 60

4.70 A screenshot of Story: ABT 18 from Jira 60

4.71 A screenshot of Story: ABT 59 from Jira 61

4.72 A screenshot of Story: ABT 19 from Jira 61

5.1 A screenshot of finding bus number intent going to Kanturk 63

5.2 A screenshot of finding bus number intent going to Dublin Airport 63

5.3 A screenshot of Next Bus to Location intent 64

5.4 A screenshot of Next Bus to Location invalid mardyke 65

5.5 A screenshot of Next Bus to Location valid mardyke 65

5.6 A screenshot of Arrive by Time bus intent success 66

5.7 A screenshot of Depart by Time bus intent success 67

5.8 A screenshot of Journey time calculation intent 67

5.9 A screenshot of Journey time calculation failing intent 68

5.10 A screenshot Alexa Skill Metrics 68

List of Tables

2.1 Comparing major 4 smart assistants 8

Abbreviations

IoT	**I**nternet of **T**hings
GTFS	**G**eneral **T**ransit **F**eed **S**pecification
GTFS-R	**G**eneral **T**ransit **F**eed **S**pecification-**R**ealtime
API	**A**pplication **P**rogramming **I**nterface
SDK	**S**oftware **D**evelopment **K**it
TFI	**T**ransport **F**or **I**reland
AWS	**A**mazon **W**eb **S**ervices
RDS	**R**elational **D**atabase **S**ervice

For/Dedicated to/To my...

Chapter 1

Introduction

1.1 Motivation

Today in the 21st century information is extremely widespread, and almost everyone can access whatever information they seek within minutes. Humans have forever been obsessed with how fast can they get their information. In the 20th century people were obsessed with how fast they could send their mail to the same degree we are today on how fast is our internet speeds. So the United States Postal Service went as far as to explore using missiles to deliver mail[1]. We have a similar problem today, how fast can we get information quickly to aid us in our day-to-day lives? For example, if we wish to find out when the next bus arriving is arriving, we will need to take out our phones, unlock it, use a search engine and type in our query and then scroll through the bus timetable to figure out when is the next bus. This is something called a long intent to action. It takes us a long period of time to gain the information we seek.

To help reduce our intent to action there has been an introduction of smart speakers. We can simply call our smart speaker of choice with our question and get a reply back instantly. This can be done hands-free and we don't need to be holding the device physically either. Anyone can add functionality to their smart speaker by developing apps for it or downloading it from their respective company's app store. In this project, I will create an Amazon Alexa skills service that reads bus timetables and answers user's queries regarding them, such as "How do I get to Cork Airport?" or "How long is it between MTU and Kent Train Station?"

There is also the environmental side of things as well. By making this application I hope to make it more convenient for people to access public transportation. As this application will make it more accessible to users to discover public transportation near

them and how they can avail of it to get to their destination on time. Having good access to public transportation can improve the quality of life [2].

1.2 Demand for product

Amazon Alexa is a voice-controlled smart assistant made by Amazon for it's smart home products such as Echo Dot, Echo Studio, and Echo Show. But this eventually expanded to smartphones, tablets, and IoT devices such as smart doorbells and TV remotes.

Users will be able to enable this skill from the Amazon Alexa Skills store on their Alexa-supported devices. With this application, users will be able to quickly and accurately gain information about the bus services available in Ireland. It will also assist them in getting to their desired location quickly and efficiently as possible. The project will use Transport for Ireland's new GTFS-R API to get live bus information. If we look at Google Trends in figure 1.1 we can see there is high demand for people searching for bus timetables not only in just Cork but all over Ireland.

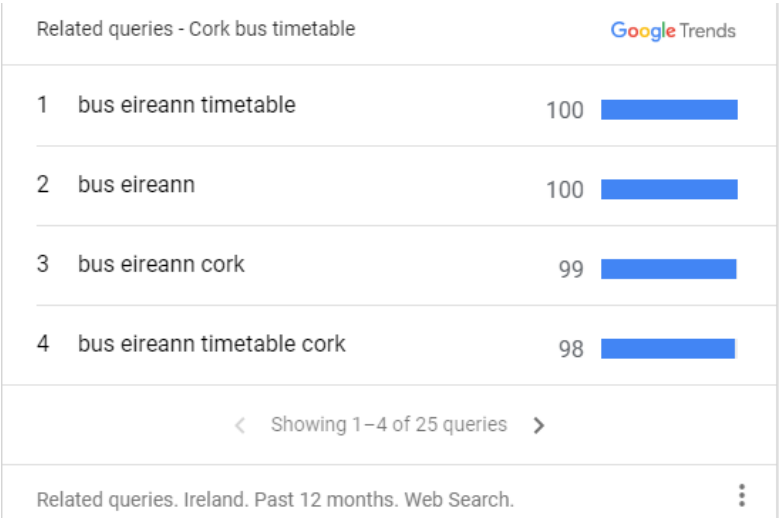

FIGURE 1.1: A screenshot from Google Trends

As shown there is a very high demand for people searching for bus timetables. These searches people are doing take time and require them to go on different websites to get the information they want. If the user clicks on a malicious link by accident this can cause harm to their devices. Using a smart home assistant reduces the risk of harmful software being potentially installed on a user's device.

There are currently similar applications on the Amazon marketplace to get bus timetables using Alexa. One application is called "Bus Times"[3]. This application is made for the UK and it requires you to enter your local bus stop identifier code. Once that is

completed it can tell you when the next bus is arriving at your local bus stop. Another application is called "Next Bus"[4] and this application is made for buses here in Dublin, Ireland. However, this application is no longer available to try out.

These examples show there is demand for this application, and users are also actively using the application. The "Bus Times" Skill for example has more than 130 reviews on the Amazon Store.

1.3 Objectives

1. Learn how to make Alexa Skills Applications so I will be able to build and later publish my own Alexa Skill
2. Plan my Alexa Skills which assist me when I am implementing my project so I will know what to do in certain areas
3. Connect and query the TFI GTFS API so I can start to receive data regarding the bus timetables.
4. Build the interaction model to handle happy path queries. Such as the user asks when is the next bus to Cork Kent Train station. The interaction model must be able to give a response back to this query.
5. Add functionality to handle error queries to the interaction model. For example, if the user asks questions about bus times in Paris an appropriate response must be given.
6. Beta test the application to make sure there aren't any flaws or bugs which I may have missed while developing the application
7. Publish the skill to demonstrate the completeness of my project
8. Prepare the report to log all the work I have done for this project and what was the result of my application

1.4 Structure of This Document

This report is divided up into several chapters in which I will discuss different aspects of the project. In Chapter 1, I will have my introduction in which I will explain my motivation for making this project, along with the demand for it in the market today. I Will

also explain my objectives behind this project which will be what I aim to accomplish at the end of the following semester.

Chapter 2, I will go over the different smart assistants available in the market today along with their benefits and drawbacks. This is to evaluate the different options available. After comparing each assistant's pros and cons I will make a decision on which smart assistant I believe will be best suited for this project. Next, I will go over the different methods of data acquisition I can use to get the bus timetable information along with which programming language I will be using for this project. I will also discuss the related work that is currently available in the market today. I will provide a brief explanation of the currently available applications along with the drawbacks each application has.

In Chapter 3, I will discuss the functional requirements of my Alexa Skill Application and the high-level architecture diagram I have created. In which I will cover the analysis and design of the project. Finally, at the end of the chapter, I will discuss potential risks which may occur during the implementation section of the project.

In Chapter 4, I will first discuss what problems occurred during the implementation phase of this project. During this discussion, I will discuss how severe the problem was, a brief description of it, and what I did to resolve the issue. Then I will discuss the queries which I made in my project. I will discuss why I made these queries and how they can be used. Then I show and explain my new updated dataflow diagram. Then I will discuss the limitation and assumptions I had to make during the implementation phase of this project. Finally, I will go over my Jira stories and give detailed explanations about them.

In Chapter 5, I will discuss the testing and my evaluation of the project. I will discuss the queries I have made and what potential faults they may have with them. I will then show the application metrics and finally, I will show how I achieved and fulfilled all of my objectives which I set out in Chapter 1.

Finally in Chapter 6, I will first discuss the original problem existing today and what my solution has done to solve that. I will do a project review on what could be done differently if I could redo this project all over again. I will then discuss the lessons I learnt when making this project and how this project could be improved in the future. Then I will give my conclusion and my final thoughts about this project.

Chapter 2

Background

2.1 Thematic Area within Computer Science

This project's core topic focus is on chatbots and using conversational agents to help us in our everyday lives.

2.2 Review of smart assistants products available

There are many smart assistants currently available in the market today but I will only focus on the 4 major ones which are: Alexa made by Amazon, Cortana made by Microsoft, Google Assistant made by Google, and finally Siri made by Apple. There are many differences and features between each assistant with some having more functionality over the others.

2.2.1 Cortana

Let's start with Cortana which was made by Microsoft and introduced to the world in April 2014. It came preinstalled on Windows 10 and was eventually expanded to Android and iOS devices in 2015. However, in 2020 Microsoft changed course and started to remove the Cortana app in certain markets and in 2021 Microsoft officially ended support for Cortana. They removed the Cortana app from the Google Play Store and Apple App Store[5]. Microsoft is no longer pushing Cortana heavily in Windows 11 either. Users are required to manually search for it and enable her services if they wish to continue to use Cortana. For this reason, I decided not to make the application for Cortana as it is no longer supported or advertised by Microsoft.

2.2.2 Siri

Next, we have Siri Made by Apple. Siri has been around since October 2011 making it the oldest smart assistant on our list. It is available on a wide range of products from everything from the Apple iPhone to the Apple Smart Watch. Apple even offers a smart speaker called the "HomePod mini" which has Siri built into it. As Apple has one of the highest market shares in the smartphone market with it's iPhones all across the world Siri is in the vast majority of the population's pocket already.

However as big and mighty as Apple is, they have heavy control over its ecosystem and limit the abilities on what a 3rd party developer can do with their app. This is the case for Siri as well. Apple has created a "SiriKit" which developers use to make Siri apps for Apple devices. However, Apple is reducing the type and number of commands for 3rd party developers to use in their latest update for all of their devices across the board[6]. They removed 22 intents and commands used by developers last year without providing a valid explanation to their users as to why they did so.

Apple also requires any software to be made for their product suite that it must be made using MacOS. Their IDE Xcode is only available on macOS with no official solution for users using a different operating system such as Windows. Another disadvantage of making the application for Apple is that you only have the choice of making the application in Swift, Apple's own programming language, or in Objective-C. This limits developers such as myself who have never worked with either of these programming languages before.

Finally, to make an application for Siri, developers are required to make either an iOS or watchOS application that contains the intent you made for Siri. This means you will be required to create not only a Siri application but also an iPhone/Apple Watch app to support the Siri app. This means you will need to pay Apple $99 USD per year to upload and host your application on the app store. This along with Apple making it more restrictive with its newest software updates makes it difficult to develop an app for Siri.

2.2.3 Google Assistant

Next, we move on to Google Assistant made by Google. Google Assistant announced in 2016 has since grown from being inside only Google's made Pixel smartphones to almost every modern Android Smartphone being sold today. It is arguably one of the smartest and most powerful virtual assistants which exist in the market today. It exists on more than a dozen different platforms today with everything from iOS to Android Auto to

use on your car. It is estimated to have more than 500 million users per month[7]. This makes it easily the biggest smart assistant currently available.

Making an application for Google Assistant can be quite a complicated task. Google has a very well-detailed developers page to support developers making apps for Google Assistant. However, it is vast and can be quite confusing at times. In my opinion, it is like a giant spiderweb that can be very difficult to navigate across.

Even for this project, it took me a solid 10 minutes with 20 different tabs open to looking through the documentation on how to make a Google Assistant application. I was given different options on how to create applications for Google Home but that is not the same as Google Assistant. If I wanted to watch a YouTube video on the topic there are thousands of different options to choose from. However, the videos and guides were all created at different points in time. This happened to me in the past when I was using one of Google's SDKs to develop an application for a project. There was a several lack of new online resources available to assist me with the new changes introduced and the depreciation of several features since the guide was first created. I also found Google's own documentation to be lacking in some areas which often led me confused.

Due to these constant changes on their API and SDKs it renders a lot of old video tutorials and articles useless and irrelevant for what I wish to accomplish. This makes the implantation part of the project more difficult and time-consuming.

2.2.4 Alexa

Finally, we have Alexa made by Amazon. Alexa was introduced to the world in 2014 with the launch of Amazon's Echo. This is a smart speaker made by Amazon as a digital personal assistant to help people with their day-to-day lives by making lists, playing music, and giving weather updates. It is available on a wide range of operating systems everything from Android phones and smartwatches to TVs. One major disadvantage Alexa has over something like Google Assistant or Siri is that users will need to download Alexa on their smartphones. Unlike its competitors, it does not come preinstalled on smartphones but is often preinstalled or available to download on many smartwatches, doorbells, and even microwaves!

To make an application for Amazon's Alexa, Amazon has created the Alexa Skills Kit. It allows developers to create an Alexa Skill for users to download from the Amazon Skills Store. Amazon even offers $100 USD AWS (Amazon Web Services) credit to publish the skill. They also allow developers to host their skills on Amazon's servers or by using an HTTPS web service.

Amazon has a very user-friendly developer page with clear instructions on how to get started with videos made by Amazon Developers on how to get started.

They have a full introduction course for free on how to get started making an Alexa Skill teaching the basics and then slowly expanding the functionality of a skill. I find Amazon's developer platform to be a lot more user-friendly. Which is why I will be using Alexa to build this application.

2.3 Smart Assistant Comparison

TABLE 2.1: Comparing major 4 smart assistants

Frequency/ Consequence	**Google Assistant**	**Alexa**	**Siri**	**Cortana**
Human Speech Recognition	Excellent	Average	Average	Excellent
Well Documented SDK	Average	Excellent	Average	N/A
Online Support	Excellent	Excellent	Average	N/A
Cost to develop	€59 + free to publish	€40 + free to publish	€100 + $100 publishing fee	Cannot publish applications
Availability	1 Billion devices in 2020	163 Million devices in 2021	500 Million devices in 2022	No longer available

From my research and comparison of the different smart assistants I reviewed today I believe Alexa is the best available assistant to develop my application. As it provides excellent support to its developers, a well-documented SDK, and has the lowest cost of development.

2.4 Data Acquisition

Next, I will discuss what data I will need from Bus Éireann 's website regarding the bus timetables and the different ways I can get this data.

One of the important data I will need from Bus Éireann is the number of different buses running around Ireland along with where and when will they be stopping and running. This is one of the most crucial bits of information I need for this project. I will need to be able to retrieve information about different buses leaving from one particular stop or be able to find out where a bus is heading using the bus number.

For example, if a user wanted to know when can he or she get the next bus to Kent Train Station they can ask "When is the next bus to Kent Station?". The application needs to understand the user's wishes to go to Kent Train Station in Cork. Then it will need to query the database with all of the bus stops near MTU that will head towards

Kent Train Station and then finally query the database which is the next bus to leave MTU. After the application has finished querying the database with all this information it can then finally read it back to the user with the timings of the next bus.

I will need a source to get this information from and from what I have researched there are two main ways I can do this. I can either use Transport for Ireland's brand new GTFS-Realtime API or I can download the timetables from Bus Éireann 's official website and then use PDF parsing to store them in a database.

2.4.1 PDF Parsing

First I will discuss using the PDF Parsing method. There are multiple ways to PDF parse the bus timetable with different methods and using different programming languages. Many popular languages such as Python or Java have libraries made for PDF parsing. This will make it relevantly easy to extract data from the timetable and store it inside a database. Since the bus times and stops are easily listed in a consistent form in the timetable as we can see in the diagram below.

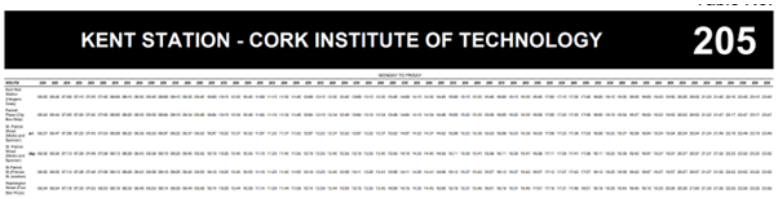

FIGURE 2.1: A screenshot of the 205 bus timetable in Cork[7]

However, one major disadvantage of this method is that I will need to download all the PDFs one by one and then store it in the database. The data will be accurate for the users until Bus Éireann changes it's timetable or the routes get moved then all of the data stored inside the database will be inaccurate and no longer relevant towards the user.

As we know the bus network in Ireland is in the middle of a change with the new BusConnects initiative by the National Transport Authority. This new initiative is changing route schedules all across Ireland with new bus times and new stops being introduced. If I went with the PDF parsing method the data stored inside the SQL servers will be out of date in a few year's time. This makes PDF parsing an undesirable option as the database will be needed to be updated every time a new bus time is released by Bus Éireann.

2.4.2 General Transit Feed Specification API

The second option for getting data is by using the Transport for Ireland API. This is a recently launched API that provides real-time updates to where the bus is currently at in its route and how long before the bus arrives at the intended bus stop. This API covers all the buses operated by Dublin Bus, Bus Éireann, and Go-Ahead Ireland. The GTFS API stands for General Transit Feed Specification. It was a side project for a Google engineer who wanted to include public transport data inside Google Maps. This eventually became successful and was then made public.

GTFS replies on transport companies creating a GTFS feed containing the routes which they serve along with times and stops. The feed is then submitted to Google where if approved it will be added to Google Maps and other Google-related services. Studies have shown that GTFS is highly versatile as a data source[8].

It is now used by countries and public transport companies around the world and is free to use. However, there are fair usage limits set by the transport provider. Which here in Ireland is set to a limit of 5,000 requests per day.

GTFS Realtime is an extension of GTFS and it allows developers to give real-time updates about the status of public transport towards users quickly and efficiently. The major advantage of using the GTFS Realtime API is that it gives real-time updates about routes in case there are any delays currently happening or buses not operating. This extra information will not be available if I were to be using PDF parsing.

An example of the output from the GTFS Realtime API can be seen below.

```
content-length: 3473796
content-type: application/json
vary: Origin

{
    "header": {
        "gtfs_realtime_version": "2.0",
        "incrementality": "FULL_DATASET",
        "timestamp": "1683055753"
    },
    "entity": [{
            "id": "T1",
            "trip_update": {
                "trip": {
                    "trip_id": "3292_32054",
                    "start_time": "13:10:00",
                    "start_date": "20230501",
                    "schedule_relationship": "SCHEDULED",
                    "route_id": "3292_46300",
                    "direction_id": 1
                },
                "stop_time_update": [{
                        "stop_sequence": 30,
                        "arrival": {
                            "delay": 1579
                        },
                        "departure": {
                            "delay": 1609
                        },
                        "stop_id": "8530B1581501",
                        "schedule_relationship": "SCHEDULED"
                    },
```


FIGURE 2.2: A screenshot of sample output from the GTFS API

With these two options in mind, I will be choosing to use GTFS Realtime API to help provide data to the Alexa application as it gives more reliable and accurate information to users. Since the data is real-time any disruptions or delays will be read out to the user whereas this flexibility is not available if I were to use PDF parsing to get the timetable data.

2.5 Developer Technologies

Finally, I need to decide which programming language I will be using to make the Alexa Skills Kit. Alexa supports various languages such as Node.js, Python, Java, and C#. However, if I want to have my skill hosted on Alexa I only have the choice of either Node.js or Python.

I used Node.js a small bit in my degree and for my group project as well. However, I would not say I am proficient at Node.js and comfortable enough to write a full application in it.

For Python on the other hand I have used the language across all three years of my degree. Making me a lot more comfortable and experienced with it. Plus there are many online resources from my research that can aid me in making applications with Alexa Skills Kit using Python. I also would like to get a job as a graduate working with Python so doing my final year project using Python will benefit me greatly. For this reasons, I will be developing my Alexa Skills Kit using Python.

2.6 Related Works

There are several different Alexa Skills currently available in the market which offer users information regarding bus timetables. However, they do have their differences which I will highlight and explain how I will differentiate from what is currently available in the market today.

2.6.1 Bus Times

Bus Times is an Alexa Skill which is free to enable for Alexa devices. The application gives real-time departures for buses in the United Kingdom only. Users need to enter their local bus stop code into the Alexa app on their smartphones. Once they have completed this step users can get real-time departures for their local bus stop.

The biggest drawback of this Alexa skill is that it is available for users in the United Kingdom only. Users in Ireland are unable to use this skill as we have a different bus stop code system compared to the UK. Another disadvantage of this Alexa skill is that users need to enter their local bus stop identifier codes when first setting up the Alexa skill. This can possibly lead to some confusion and is prone to user error.

2.6.2 Next Bus

Another Bus Timetable Skill on the Amazon App store is Next Bus. Unlike Bus Times which I discussed above, Next Bus is available to use for users in Ireland. The Skill works by asking Alexa "When is the (bus number) arriving?"

A disadvantage of this skill is that it is limited to Dublin Bus only, meaning users in Dublin, Ireland can only use the skill. One slight note to add, this Alexa skill is no longer available to download for unknown reasons.

Chapter 3

Problem - Conversational agent for transportation timetable

3.1 Objectives

For this project, I wish to create an Alexa Skills application that users will be able to enable from the Amazon Alexa Store page or they can enable it by using their chosen Alexa device via voice command.

The Alexa Skill will allow users to access real-time information regarding bus timetables all over Ireland. I will be getting this data from the GTFS-Realtime API published by Transport for Ireland.

These are all objectives that I have referred to in Chapter 1.3.

3.2 Functional Requirements

I used Jira as my application of choice to help me plan and track my application. I will be using agile methodology for this project which is why I created bi-weekly sprints. Having sprints allows me to know what exactly I need to have done by the end of each sprint and how long should easy task will take.

My sprints will begin on the 23rd of January when the semester starts and will last two weeks. By the end of the two weeks, I aim to have completed a certain level of tasks and have implemented more and more functionality into my application as the sprints progress.

My product backlog is shown below, which shows the stories I have created for this project. Inside each story, I have allocated story points that refer to the estimated hours I should spend on this particular story. Along with that I also have description and acceptance criteria for each story. A link to my backlog on Jira can be found here.

There have been 182 hours allocated for the implemented phase in the second semester. Each hour is worth one point inside the stories that I have created. In total, I have 144 points story points in my Jira backlog. I have left myself some leeway in case any unexpected stories may arise that I may not have planned for or a story takes more time to complete than I initially anticipated.

3.2.1 Product Backlog

I created my backlog using Jira as it is a free service and I have experience using it from both past projects I have done in university and during my work experience. Inside each user story I have included the following:

1. I have a story description that explains the purpose and objective of each story.
2. The acceptance criteria for the story which explains what I need to have completed to consider this story finished
3. A story point estimate on how long I think it will take me to complete this story
4. Which objective as defined in 1.3 does this user story correlate to

In figure 3.1 we can a screenshot of my backlog from Jira which has all the user stories I have made so far. When semester two starts I will start moving these stories slowly into my sprints where I aim to get them all completed before the semester ends.

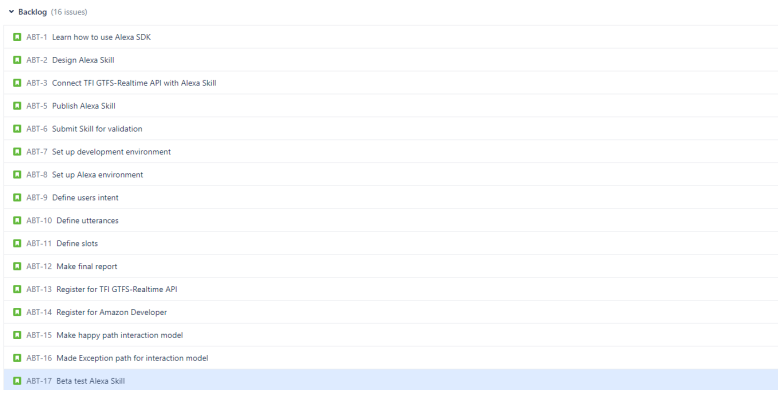

FIGURE 3.1: A screenshot of my backlog from Jira

3.3 High level architecture

Below I have created a Data Flow Diagram to provide a high-level view of my project and what I am trying to accomplish.

3.3.1 Data Flow Diagram

In figure 3.2 below we are able to see my Data Flow Diagram. This maps out the flow of information for my application. It helps us visualise the system at a quick glance and understand its intricacy.

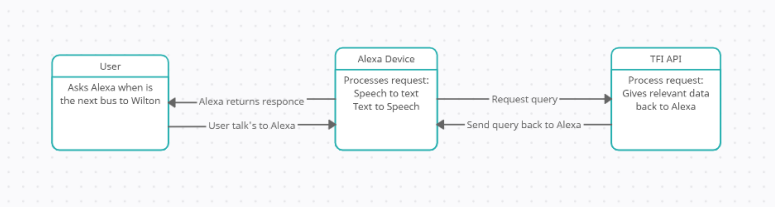

FIGURE 3.2: A screenshot of data flow diagram

3.4 Potential Risks

As with every project, nothing always goes smoothly and there is always something that can go wrong. This is why I have a backup plan in case something goes wrong. I will outline potential points of failure that could go wrong while in the development stage, the chance of that risk happening, the impact it will have and what contingency plan I will have in case it happens.

3.4.1 Become too over-loaded with assignments

- Probability: High
- Impact: High

As it will be the final semester of my degree, I imagine I will be extremely busy with assignments and exams which may prevent me from working on my final-year project. The chances of this happening will be high and will have a high impact on the project during the implementation phase.

3.4.2 Restricted by GTFS API

- Probability: Low
- Impact: Medium

The GTFS - Realtime API has been recently introduced by Transport For Ireland. There may be bugs or limitations of the API which I may not have discovered just yet. This can prevent me from using the API.

In case this happens I have my backup option of using PDF parsing to retrieve the bus timetable data. This will allow me to continue developing the Alexa Skill with some modifications in the backend.

3.4.3 Failing Alexa Skill Validation

- Probability: Low
- Impact: Medium

Before I can upload my Alexa Skill to the Amazon Store for users to download, my Alexa Skill needs to pass Amazon's validation test. If my skill passes the validation test I will be allowed to upload my Alexa Skill to the store. If my program fails the validation test I will not be allowed to publish my skill to the store. Until I fix the errors highlighted in the validation test I will not be allowed to publish my skill.

To medicate this risk I plan to publish my skill a couple of weeks in advance of the due date in order to get feedback from Amazon and make any necessary changes if required to publish the Alexa Skill.

3.4.4 Amazon redesigning Alexa to be less user friendly

- Probability: Low
- Impact: High

As Amazon has recently decided to let go of 10,000 employees from their workforce many of which worked on Alexa-related products. This massive layoff could have unforeseen side effects on the Alexa smart assistant. As there are fewer employees working on the project this may degrade the quality of the Alexa product as there may be more bugs or

software glitches when using the smart assistant. This could result in people switching away from Alexa as their smart assistant.

Another risk that may occur as a result of the layoffs is Amazon changing the way users are allowed to publish Alexa Skills which will have direct consequences on this project.

As there is no way to predict the future it is uncertain what Amazon will decide to do with Alexa in the future. If they make any drastic changes that will prevent me from publishing the Alexa Skill I will start researching alternative smart assistants I can develop my application for such as Google Assistant.

Chapter 4

Implementation

4.1 Difficulties Encountered

Below I have listed the difficulties/problems I have encountered during the implementation phase of this project. I have categorised them into three levels of difficulty. Which are:

- Easy
- Medium
- Hard

I will give a brief description of the issue rise, what risk it had on my project, and my solution on how I resolved this issue.

4.1.1 API Limitations

Difficulty: Medium

Issue Description When I first started using the GTFS API I realised very quickly there is a limitation on how many requests you are allowed to make in a short period of time. If you make too many calls to the API you are given an HTTP 429 Error as shown below in figure 4.1. There is a waiting time of 1-2 minutes between each API request.

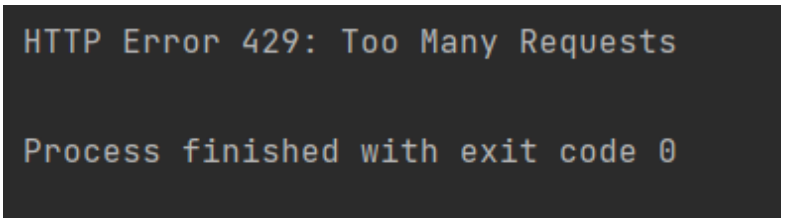

FIGURE 4.1: HTTP Error 429: Too Many Requests

Risk Imposed Due to this limitation, I was not able to make a request to the API every time a user asks a question from the Alexa skill. As if a user were to ask the Alexa skill a question regarding bus timetables the next user will have to wait 1-2 minutes before they can ask their own question. This will result in a bad user experience if customers keep having to wait their turn to ask the application a question.

Solution In order to circumvent this issue I created an Amazon EventBridge Scheduler on AWS (Amazon Web Services) which will invoke a Lambda Function I created containing the call to the GTFS every 10 minutes. This will give ample time in between requests in order to not get the HTTP 429 error again.

The Lambda function will call the GTFS API and retrieve the data from it. Once it retrieves data it will then add it to my RDS (Relational Database Service) database. When a user then requests information from my Alexa Skill the Skill will query the RDS for the data it is searching for and then return that to the user.

With this solution, multiple users can request information from the Alexa Skill at the same time without the loss of any functionality.

4.1.2 API Update

Difficulty: Easy

Issue Description As the GTFS API by NTA is fairly new they are making several changes to the specification. Whilst during the implementation stage of this project NTA upgraded their GTFS API from version 1 to version 2. This change resulted in the API the ability to provide more accurate information [9].

Risk Imposed As this API change does affect the content and structure that is provided by the API call. The changes made could affect how I am storing data inside my database. I may need to change how my tables are made inside my RDS database.

Solution Fortunately, the changes that were required on my end were quite minimal and not a lot had to be changed. NTA made a 19-slide PowerPoint explaining the

changes which were made in a clear and simple way. This allowed me to understand what changes were made to the API and helped me make the necessary modifications.

The new changes to the API in fact helped me create JOIN queries easier. With the new changes I was able to join tables more efficiently allowing me to get the tables users requested.

4.1.3 AWS Charges

Difficulty: Medium

Issue Description While I was following an online tutorial on how to set my AWS services for my Alexa Skill. I accidentally used services that were not included in the AWS Free tier suite. This resulted in me being charged for services. When I realised I was being charged for these services I stopped the services which were causing the charge. However, my AWS bill kept increasing and as a result, I had to contact billing support at AWS to resolve this issue.

Risk Imposed As the charges kept increasing at a rate of $1 USD every single day the AWS bill could become very expensive, very quickly to me, and as a result, I had to put to stop whatever was creating the charge on my account.

Solution I contacted AWS Billing support asking them for assistance on why I was getting charges for services I had previously shut down. The AWS representative explained to me some services create roles that can result in charges even if the service is shut down. He also explained to me I had accidentally created my RDS Database with 200 GiB storage allocated to it. Inside the AWS free tier, an RDS database can have a maximum of 20 GiB of storage allocated to it. Even though I was not using the full 200 GiB worth of storage I will still be charged for it.

The Amazon representative explained to me how to shut down all services which I was no longer using correctly. He informed me he was able to waive the bill on my account but in order to do that I first had to delete my RDS Database. I also needed to wait 24 hours for the bill to settle and I was advised not to use any AWS services until that had happened.

Once I was given the all-clear from the Amazon representative to continue using the services I was able to make my RDS Database once again but this time applying the knowledge he had taught me. This ensured I was able to stay inside the free tier limitations of AWS without occurring any extra charges.

4.1.4 Full Bus Stop Name requirement

Difficulty: Hard

Issue Description When I first started getting set up with the GTFS-R API I noticed that the stop names that were shown inside the API were very different to what we would normally see in real life whilst on the bus. For example, Wilton has several bus stops which are all named differently and are not places that we would associate when referring to the bus stop normally. I have shown some examples below in 4.2

2108	8370B243	243311	Wilton SC East
2109	8370B243	243341	CUH A&E
2110	8370B243	243351	Wilton Shopping Cen

FIGURE 4.2: Wilton Bus Stops

Risk Imposed This can lead to confusion for users when they are using the Alexa Skill application. As they may think they are using the correct bus stop name whereas in reality they are not and are given an error message. Which can lead to confusion and frustration among the users.

Solution I was not able to come up with a concrete solution for this problem. The datasets, the NTA provides us it does include the Longitude and Latitude of the bus stops and I attempted to utilize them to group bus stops together in a small 0.5km radius however this proved to be very difficult and time-consuming.

4.1.5 Difficulty in understanding AWS Terminology

Difficulty: Medium

Issue Description When I first started using AWS I quickly realised that AWS is more complicated than I initially thought it would be. As a result, it took me more time than I anticipated to familiarise myself with AWS and how it works.

Risk Imposed As AWS is a very crucial part of my application it is important I fully understand how it works and use it to its maximum potential.

Solution I had to take more time to fully go over how AWS works with their services such as VPC (Virtual Private Cloud) and how it can be paired with Lambda functions and RDS Databases.

4.2 Resolving queries

The Alexa Skill application has been successfully published on the Amazon Alexa Store as shown below. The skill is available for anyone to try out and download on any Alexa device. A link to the Amazon store page can be found here.

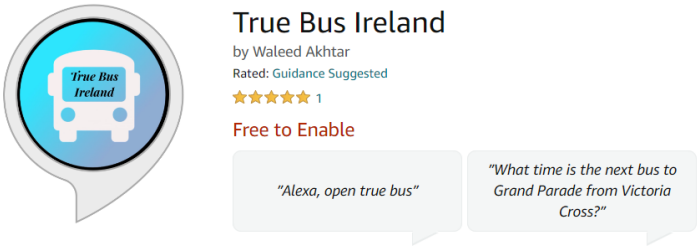

FIGURE 4.3: True Bus Ireland

The Alexa Skill allows users to ask 5 different types of questions regarding bus timetables in Ireland. These are:

- Finding Bus Number: The user can ask Alexa which bus route takes them to their desired destination
- Next Bus to Location: The user can ask Alexa what time the next bus is arriving at their local bus stop to their desired location
- Arrive by Time: The user can specify which time they would like to arrive at their destination and the Alexa Skill can provide them with buses they can take
- Depart by Time: The user can specify which time they would like to depart from a location and the Alexa skill will provide them with the next bus that is leaving after their specified time
- Journey time calculation: The user can ask how long the journey time is between two locations on the same bus route

4.2.1 Finding Bus Number

The purpose of this intent is to allow users to discover which bus number takes them to their desired destination. They can then use this information to help locate the bus stop from which the bus departs from. Users are only required to supply their desired destination to use this intent. This intent can be seen in action below in figure 4.4.

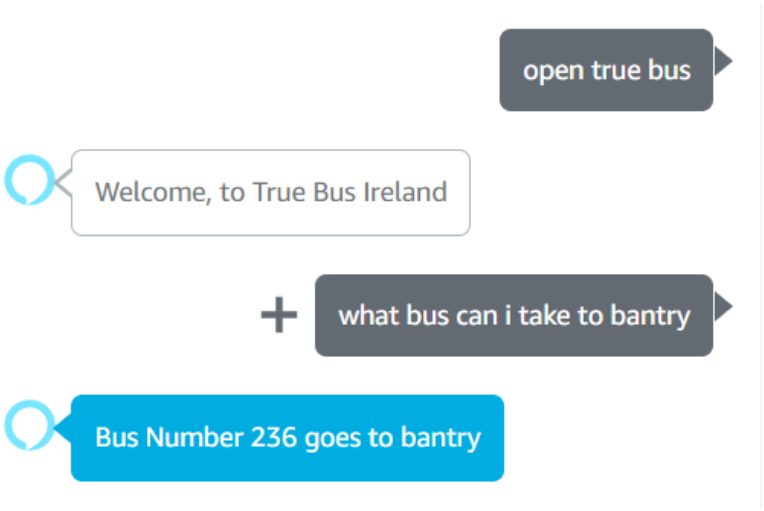

FIGURE 4.4: Find Bus intent

4.2.2 Next Bus to Location

This intent was created to help users discover the next bus that is departing from their intended bus stop going to their location. It will supply the users with 24-hour time of when the bus is departing taking into consideration daylight savings time. The user needs to provide the destination they wish to go along with the source bus stop from where they are originating from.

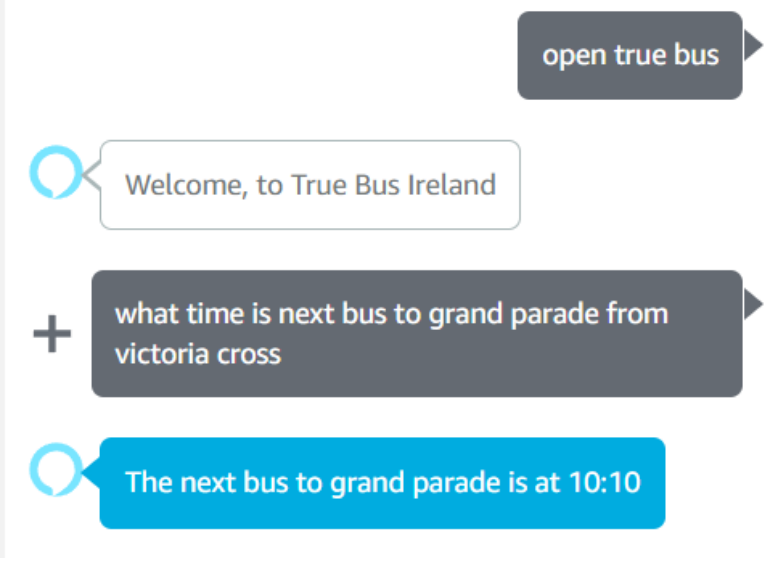

FIGURE 4.5: Next Bus to Location intent

4.2.3 Arrive by Time

The purpose of this intent is to allow users to ask Alexa what bus should they take in order to arrive at their intended destination by a given time. The user needs to specify their intended destination, source location, and time to use this query as shown below in figure 4.6.

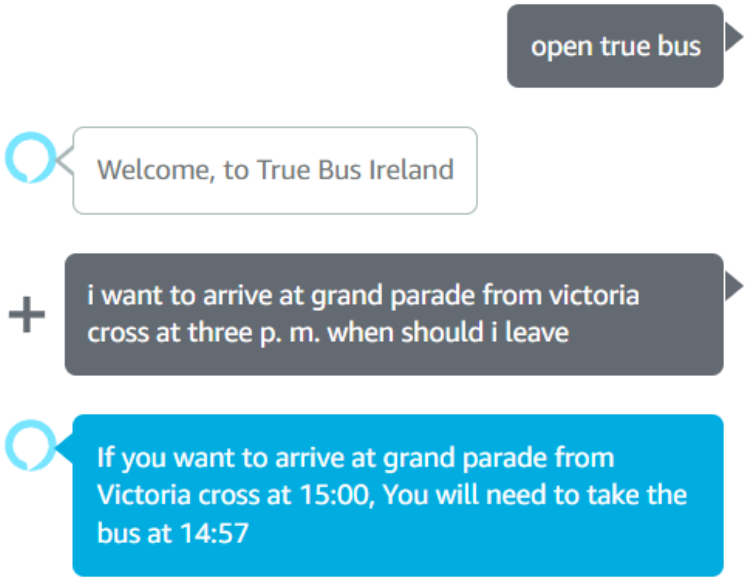

FIGURE 4.6: Arrive by Time intent

4.2.4 Depart by Time

This intent can be used by users wondering when will a bus arrive at their local bus location going to their intended destination. They will be required to specify their intended destination, source location, and time to use this query as shown below in figure 4.7.

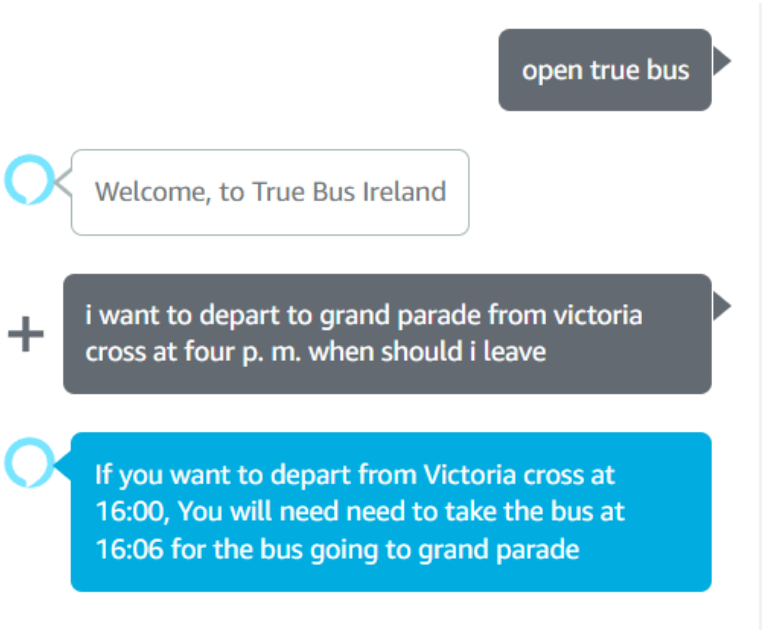

FIGURE 4.7: Depart by Time intent

4.2.5 Journey time calculation

Finally, this intent is used to inform users how long their journey can take by bus if they wish to travel between two locations. The user is required to specify their intended location and their source to use this query which is shown below in figure 4.8

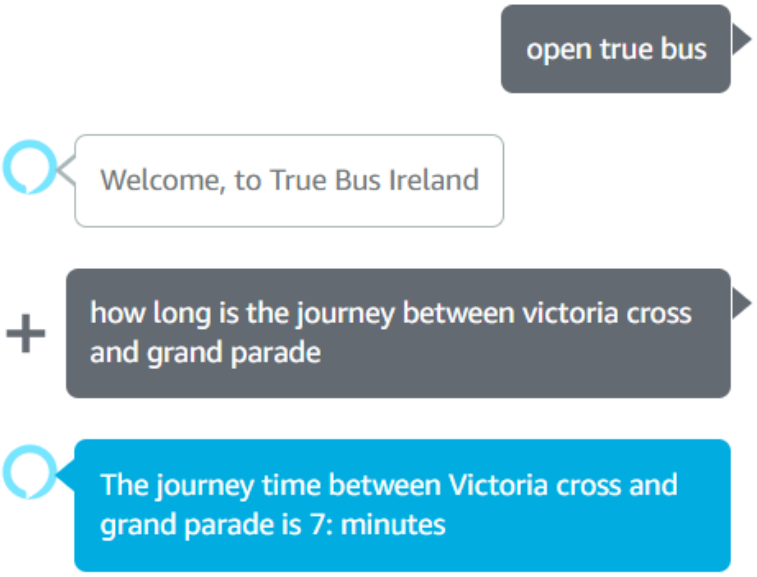

FIGURE 4.8: Journey time calculation intent

4.3 Dataflow Diagram

Below I have attached my Dataflow diagram which shows how the Alexa SKill application works and how everything is connected. I will then explain in depth why I made certain choices when developing this Alexa Skill Application.

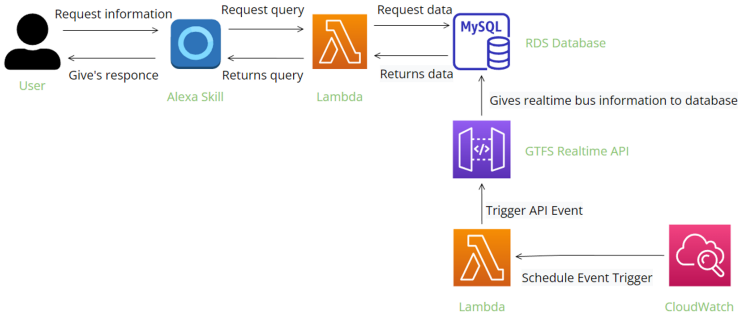

FIGURE 4.9: Dataflow Diagram

4.3.1 RDS Database

I decided to use a MySQL RDS Database in my project which is used to store all of the bus information from Bus Éireann buses running across Ireland. I discovered RDS when I was investigating the difference between using Python or DynamoDB to query the data I am getting from the GTFS API to get the desired user output. While I was online researching DynamoDB I discovered RDS and how it was a relational database and was included in the AWS free tier. I found it to be more interesting and easier to use compared to DynamoDB and began researching how to use that instead.

Comparing different architectures As I was still investigating which architecture to use for my project I decided to substitute DynamoDB with RDS. I created a proof of concept of both Python and SQL extracting data from the dataset provided by the NTA to compare which option is better and which one I would like to pursue forward in. I created the following Pros and Cons list.

Python

Pros:

- Easier to deploy

Cons:

- Loading data takes a long time
- Need to create several functions for just one query

SQL

Pros:

- Easier to make SQL statements

Cons:

- Deployment

In the end, when comparing my options I decided to go the SQL route. Why I choose a MySQL database because I am already familiar with MySQL Syntax having used it before in projects and during my work experience.

4.3.2 Lambda function

I have created a Lambda function that connects to the GTFS-R API in my diagram. The purpose of this Lambda function is that when it is executed it will call the GTFS-R API to fetch real-time data regarding Bus Éireann services running across Ireland and add this data inside the RDS Database.

Below I have attached a screenshot from the NTA Developer Portal showing the data transfer of real-time bus information into my database over a 24-hour period. We can see there is virtually 0 data being transferred in the early hours of the morning when only 24-hour buses are running and as the day progresses when more and more buses start to run more data is transferred.

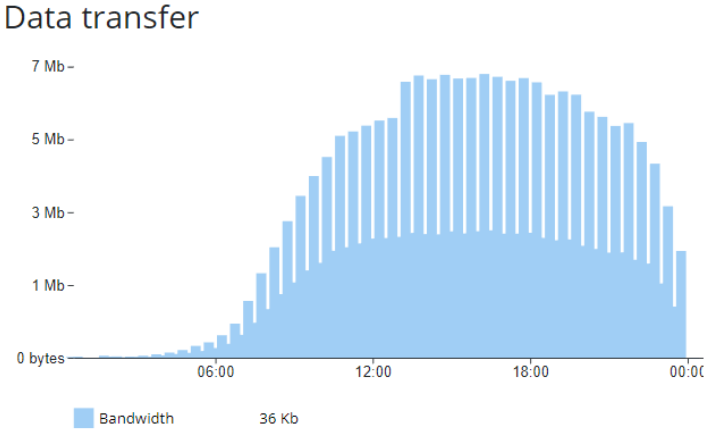

FIGURE 4.10: GTFS-R Usage reports: Data transfer in 24 hour period

4.3.3 CloudWatch

I had to create an CloudWatch scheduler to run every 10 minutes which will trigger the Lambda function to get the real-time data from the GTFS API. This is due to a limitation from the API where you cannot make too many requests in a short period of time.

4.4 Limitations & Assumptions

There are a few limitations and assumptions made when using this Alexa Skill Application. I have specified and explained these below.

4.4.1 Full Bus Stop Name

One of the biggest limitations of this Alexa Skill Application is that users need to provide the full name of the bus stop in order to use this Alexa Skill. If shortened or abbreviated names are used the Alexa Skill may not be able to find the bus stop inside the RDS Database. So in some scenarios, as shown below in figure 4.11 the user is required to say the full stop name which in this case is "Wilton rd cu".

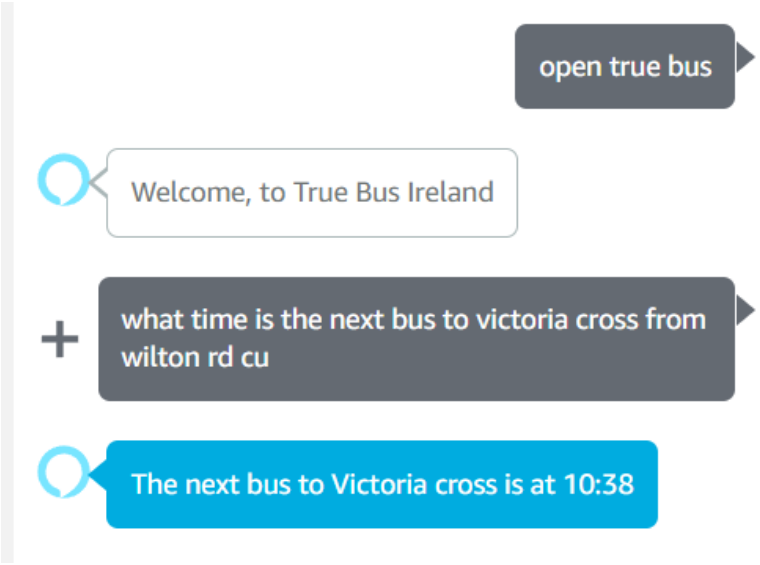

FIGURE 4.11: Full bus stop name: Wilton Rd Cu

The reason for this is that this is how the stop name is saved on the GTFS-R list of stops. There are multiple stops beginning with "Wilton" inside the dataset so unless the stop is fully specified an error message is given to the user asking them to say the full stop name.

There are even cases inside the dataset where the stop isn't the full name of the official stop in real life. For example, with Wilton Shopping Centre the stop name inside the dataset has been abbreviated to just "Wilton Shopping Cen" as shown below in 4.12.

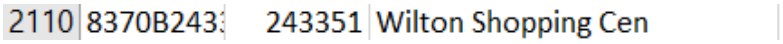

FIGURE 4.12: Wilton Shopping Cen

This is a severe problem with the application as it often cannot find bus stops users specify and often returns an error message back to the user asking them to try again.

4.4.2 Only support for Bus Éireann

The GTFS-R API made by the NTA can support numerous transport operators across Ireland not just Bus Éireann services. They also support Citylink, Dublin Bus, and GoAhead Ireland bus timetables that are running across Ireland. My Alexa Skill Application only utilises their coverage for Bus Éireann buses that are running.

This can be a later point of expansion for the application later down the line. To support numerous bus operators and even Irish Rail and LUAS times as well.

4.4.3 Journey time intent limitation

For the journey time calculation intent, there is a limitation which means you can only check the time differences between two bus stops on the same bus route. For example, you can only see the journey time between bus stops on the 205 bus route. You cannot calculate how long it would take you if you had to get a connecting bus and then continue on that bus route.

4.5 Privacy Policy

One of the requirements to publish the Alexa Skill application is to have a privacy policy made for your application. I was able to create a privacy policy using GitHub Pages to create a website that contained the privacy policy. My privacy policy for my application is shown below.

Privacy Policy for True Bus Ireland

This privacy policy describes how our ("us" or "we") protects your ("you", "the user") privacy and your data. Before using any of our Alexa Skills ("our skills"), please read

this policy and our Alexa Skills Terms of Use, as well as the Alexa Terms of Use and the Privacy Policies by the Amazon Digital Services LLC (with its affiliates, "Amazon").

Regarding Amazon, "Alexa" means their Alexa Voice Service which includes third party services (like our skills) and other related Software.

If you use one of our skills you fully agree to this privacy policy.

General

When you use our skills you have to talk to Alexa. This voice input is sent to Amazon and us where we use it to understand what our skill should do for you. This is absolutely necessary for our service to give you an appropriate answer.

Data

We never collect or share personal data with our skills.

To improve our services we analyze automatically how often utterances are spoken and other analytics. This is done automatically by Amazon in the Amazon Developer Portal

Contact Us

If you have any questions about this Privacy Policy, you can contact us at: By email: waleed.akhtar@mycit.ie

4.6 Project management with Jira

I had a total of 7 Sprints during the implementation phase of this project. Each sprint was 2 weeks long containing user stories I created. Inside each user story, there is a description, estimated story points, objective, and acceptance criteria. Inside all of my user stories above the acceptance criteria I have listed which of my objective this user story correspond to. I have defined my objectives for this project in section 1.3 of this report. Below I will list all of my user stories that I made during the implementation of this project in order of which they were completed in.

4.6.1 Learn how to use Alexa SDK

This user story is to do with learning how Alexa SDK works. Since it is an SDK I have never used before there is a lot I need to learn about the development kit can be used.

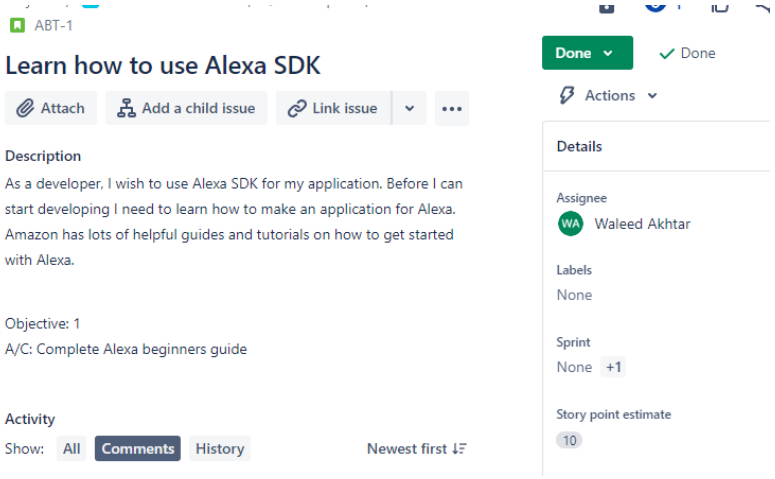

FIGURE 4.13: A screenshot of Story: Learning how to use Alexa SDK from Jira

4.6.2 Set up development environment

This user story is quite basic, as shown in figure 4.14. It is a story meant for me to set up my development environment on my laptop. I should have created a GitHub repository, PyCharm updated and Git set up on my local machine.

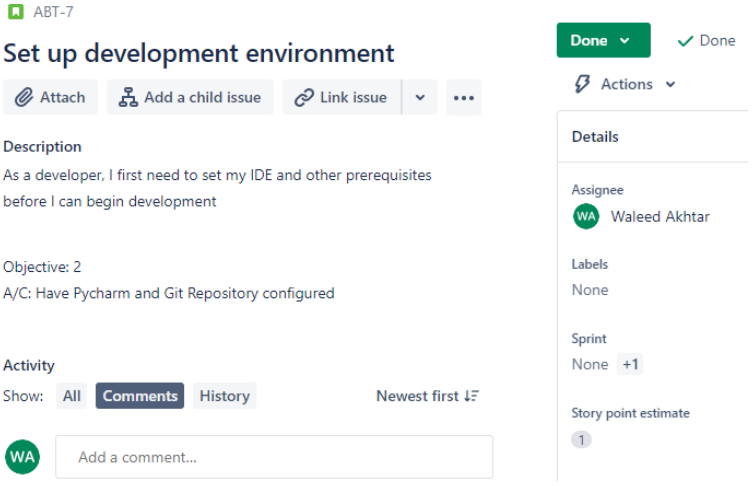

FIGURE 4.14: A screenshot of Story: Set up a development environment from Jira

4.6.3 Register for TFI GTFS-Realtime API

For this story, I need to register as a developer on Transport For Ireland's developer website to gain access to their GTFS API. When I register as a developer I will be given access to use the API along with an API Key I can use to make requests when using the API.

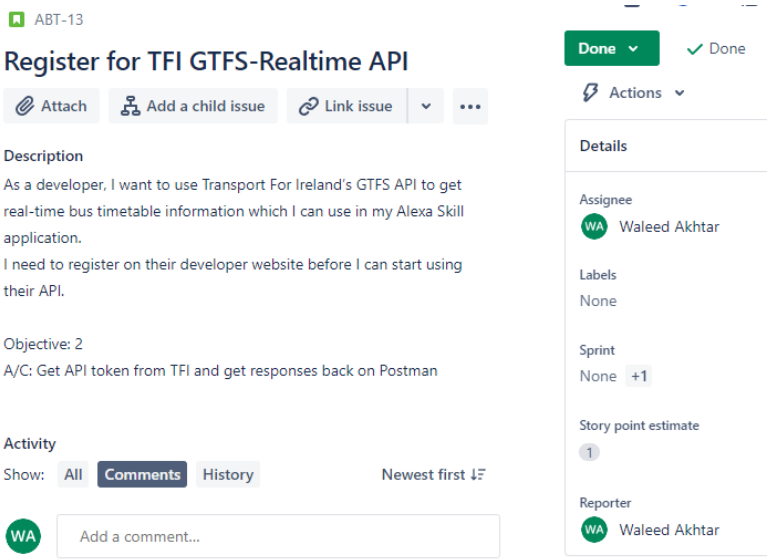

FIGURE 4.15: A screenshot of Story: Register for TFI GTFS-Realtime API from Jira

4.6.4 Register as Amazon Developer

Like with my previous story, I need to sign up for a website. This time I need to sign up as an Amazon Developer so I can gain access to Amazon's Alexa documentation, SDK support, Skill validation, and publication. Therefore it is crucial I register as a developer.

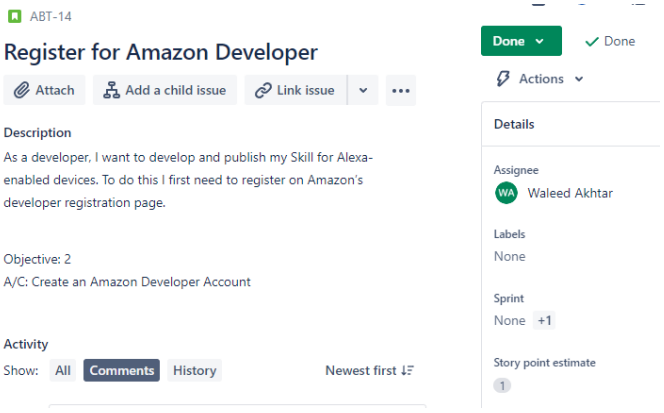

FIGURE 4.16: A screenshot of Story: Register for Amazon Developer from Jira

4.6.5 Get a response from GTFS API

For this story, I wanted to get a response from the GTFS API locally on my local environment first before I can start working in the Alexa Development Console.

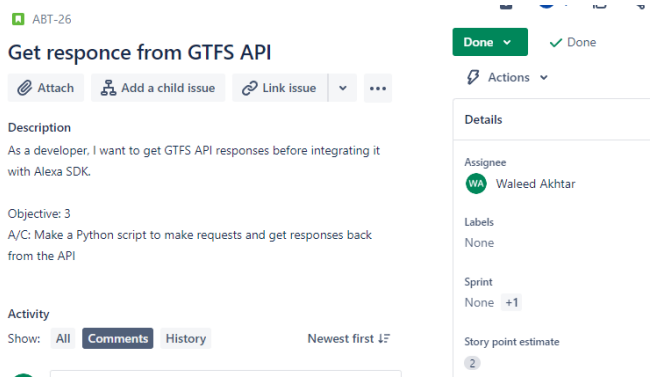

FIGURE 4.17: A screenshot of Story: Get a response from GTFS API from Jira

4.6.6 Design Alexa Skill

My next story is about designing my Alexa Skill. Amazon recommends all developers design their skills before they start any sort of development. They have guides on their website on how to design an Alexa skill to help aid the process of development. Although it is not mandatory it is highly recommended developers do this.

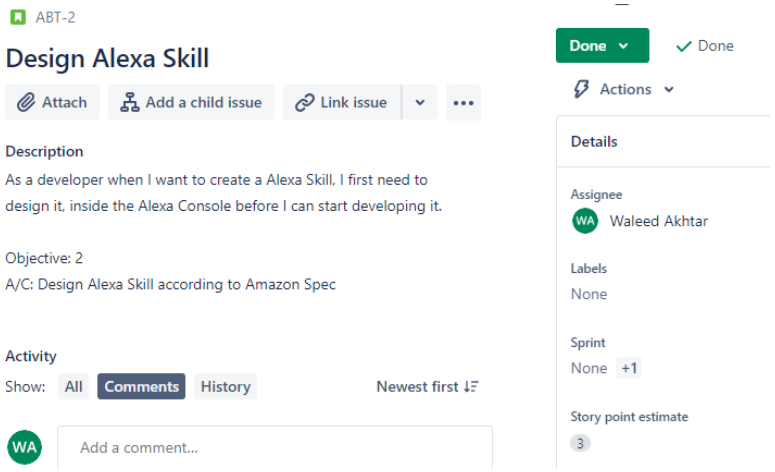

FIGURE 4.18: A screenshot of Story: Design Alexa Skill from Jira

4.6.7 Define user intent

As set out in a guide by Amazon for developers, I need to define a user's intent when they are using my Skill. In short, what is the user trying to do when they ask their Alexa assistant a question. This will be helpful later when I will be coding the application as it creates a high-level goal for me to accomplish.

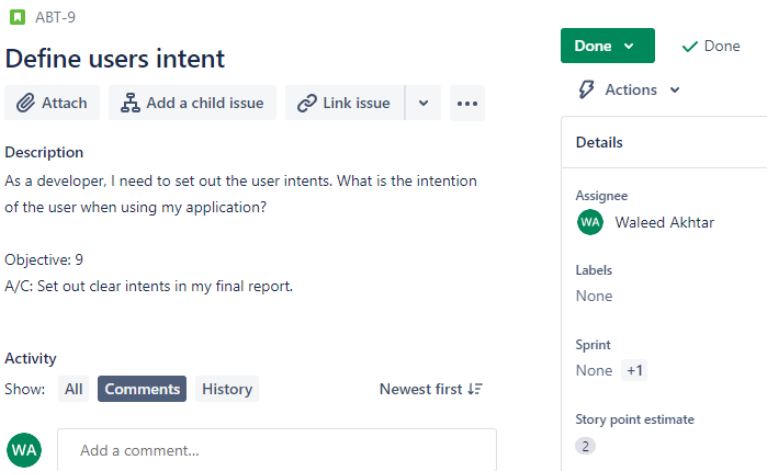

FIGURE 4.19: A screenshot of Story: Define users intent from Jira

4.6.8 Define utterances

Also set out in a guide by Amazon for developers I need to define utterances for my Alexa skill as well. An utterance is a specific phrase that the user may ask Alexa when making a request. For example, "Alexa, when is the next bus to Ballincollig from Grand Parade?" This however is one example of a way a user may ask a question to Alexa. They could ask the same question but phrase it differently such as "Alexa, when is the bus to Ballincollig arriving at Grand Parade?" The user is requesting the same information from Alexa but asking it in different ways. So for this story, I will need to think of different ways users can ask the same question.

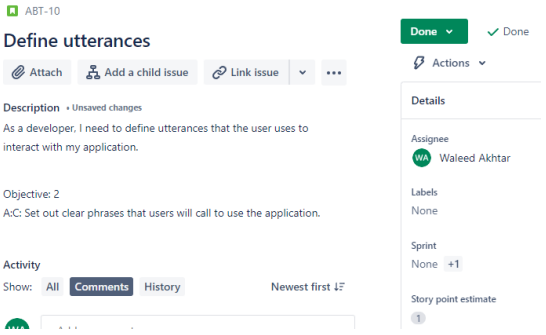

FIGURE 4.20: A screenshot of Story: Define utterances from Jira

4.6.9 Connect TFI GTFS-Realtime API with Alexa Skill

The next user story is extremely crucial. It is to do with connecting the GTFS - Realtime API with my Alexa Skill. As this is how I will get my live bus feed on how long till a bus arrives to a bus stop or the schedule for the next few hours for a particular bus.

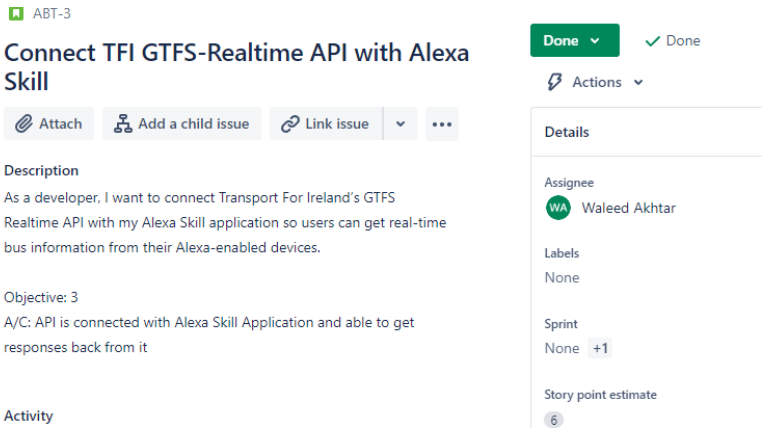

FIGURE 4.21: A screenshot of Story: Connect TFI GTFS-Realtime API with Alexa from Jira

4.6.10 Connect Alexa Skill and AWS Lambda

Inside this user story, I want to connect the AWS Lambda function I created with the Alexa Skill application I have on the Alexa Development Console.

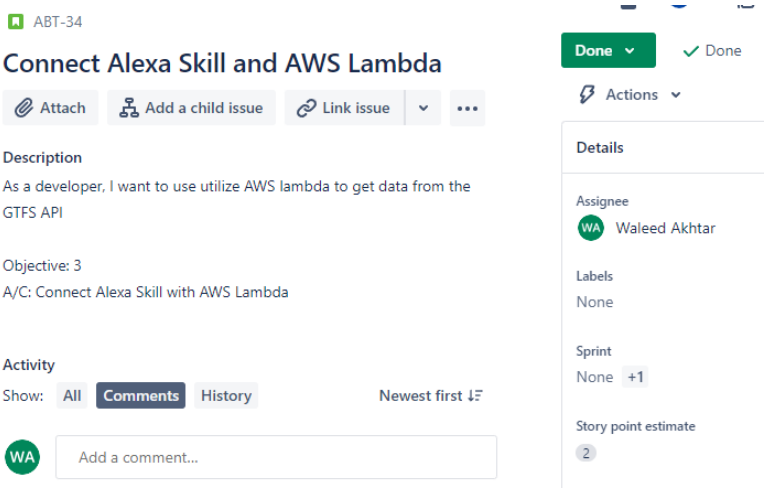

FIGURE 4.22: A screenshot of Story: Connect Alexa Skill and AWS Lambda from Jira

4.6.11 Investigate how AWS Lambda works

For this user story, I realised I was lacking knowledge in my understanding of AWS Lambda and I needed to revise it and go over some guides on how it works.

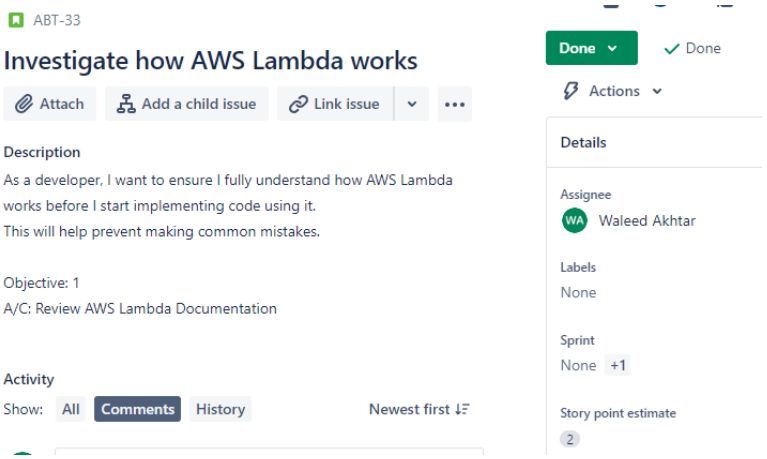

FIGURE 4.23: A screenshot of Story: Investigate how AWS Lambda works from Jira

4.6.12 Define Slots

The final definition I need for my Alexa Skill is slots. Slots are variables that are used by Alexa to understand the request the user is asking it. For example, when asking the question "Alexa when is the next 205 to the city centre?". The slot in this question is "205". This helps Alexa understand what the user is asking for which in this case is the bus timetable for 205. It can then be used to make a request to the API to retrieve this information.

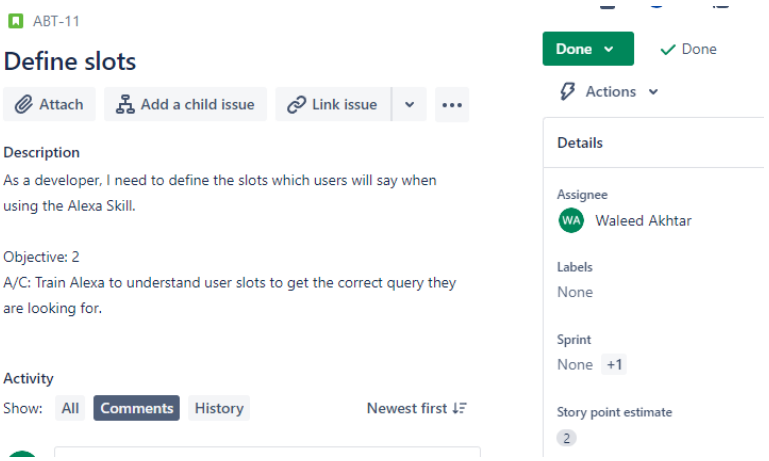

FIGURE 4.24: A screenshot of Story: Define slots from Jira

4.6.13 Investigate new version 2 API Changes

NTA has made changes to the GTFS-R API and has released version 2 of the API. This is the first major change they have made since the API's release. It is important to go over these changes to see what is new and how it will affect my project.

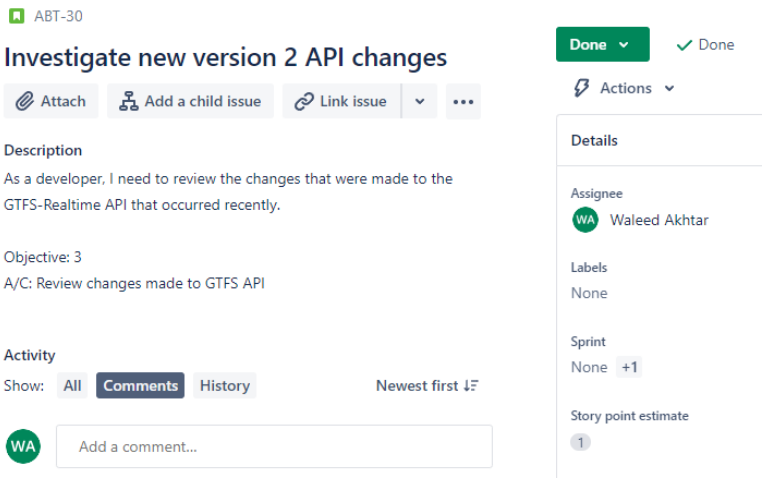

FIGURE 4.25: A screenshot of Story: Investigate new version 2 API Changes from Jira

4.6.14 Modify the application to accommodate new API changes

As there have been changes made to the GTFS API I need to make some changes on my development environment to accommodate these changes so I will be ready to support the API.

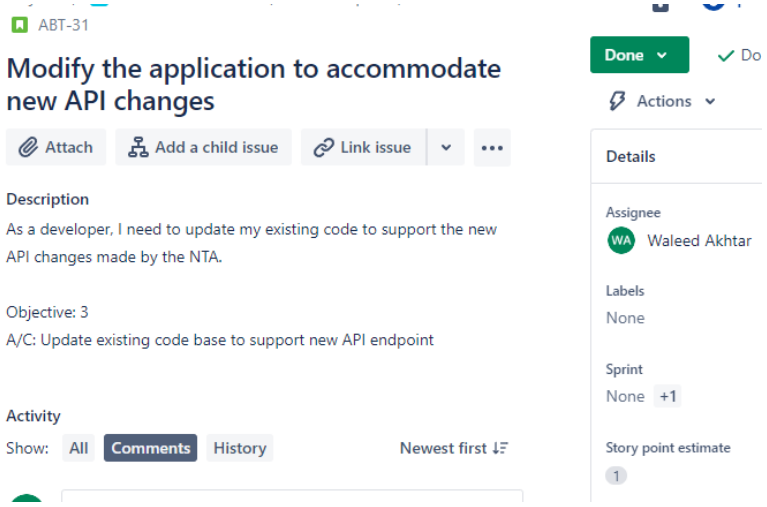

FIGURE 4.26: A screenshot of Story: Modify the application to accommodate new API changes from Jira

4.6.15 Learn how the code in Alexa Development Console works

Whilst making the API changes I realised there were parts of the Alexa Development Console I did not fully understand and needed to revise how the console works. I needed to review Alexa Documentation on how the console works and its feature.

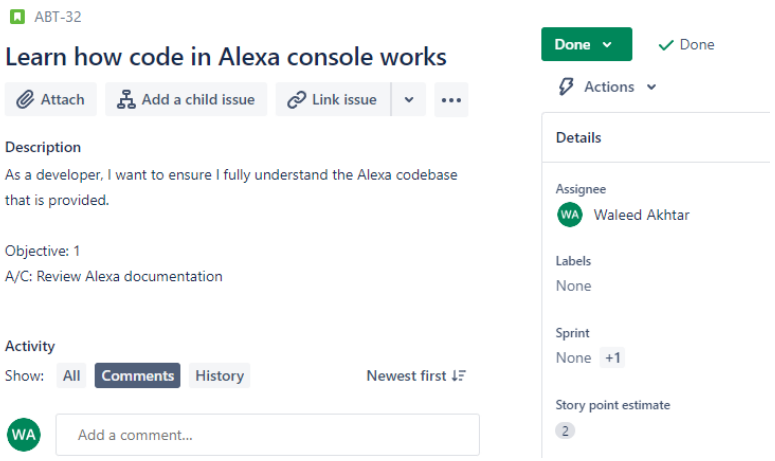

FIGURE 4.27: A screenshot of Story: Learn how the code in Alexa Development Console works from Jira

4.6.16 Investigate the difference between Python JSON query vs DynamoDB

I want to explore the different ways I can get the query output I want from the responses I get from the GTFS API. The user will provide me with key important bits of information such as their destination and their originating source location. I want to explore different ways I can get them the information they want. This can be done either by writing loops in Python or by using DynamoDB in Alexa.

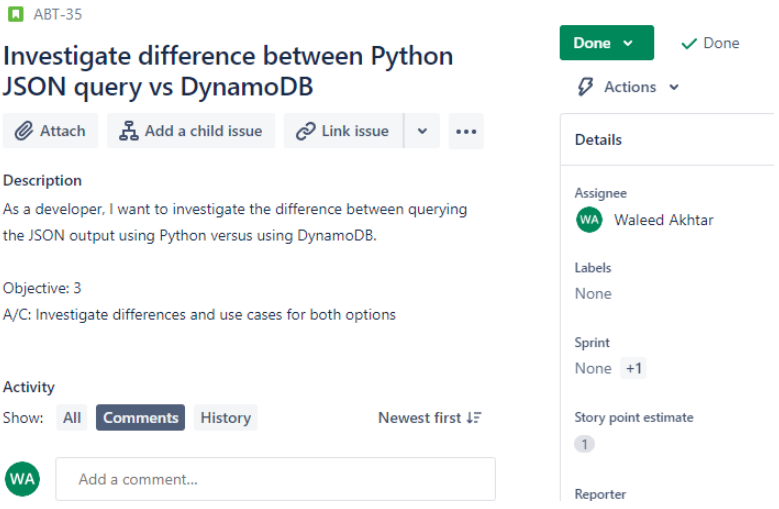

FIGURE 4.28: A screenshot of Story: Investigate the difference between Python JSON query vs DynamoDB from Jira

4.6.17 Create a proof of concept for Python JSON query

In this user story, I want to create a proof of concept for using Python loops to get the information the user wants.

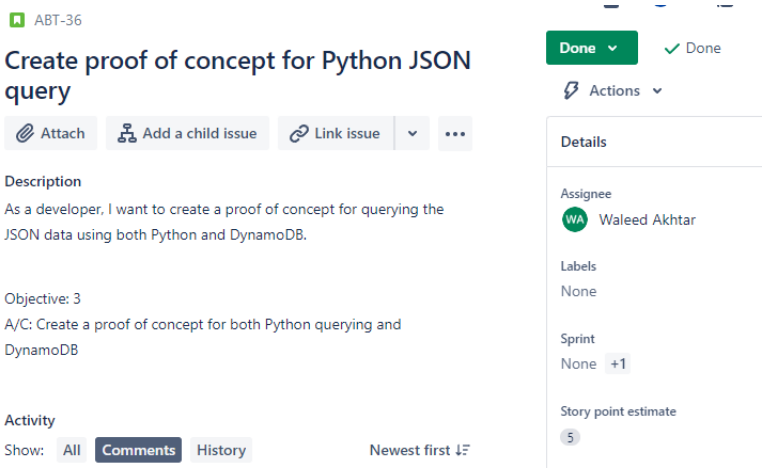

FIGURE 4.29: A screenshot of Story: Create a proof of concept for Python JSON query from Jira

4.6.18 Create RDS Database

In this user story, I want to create an RDS Database which will be used to store the stop names, trips, and the real-time bus information we get from the GTFS API.

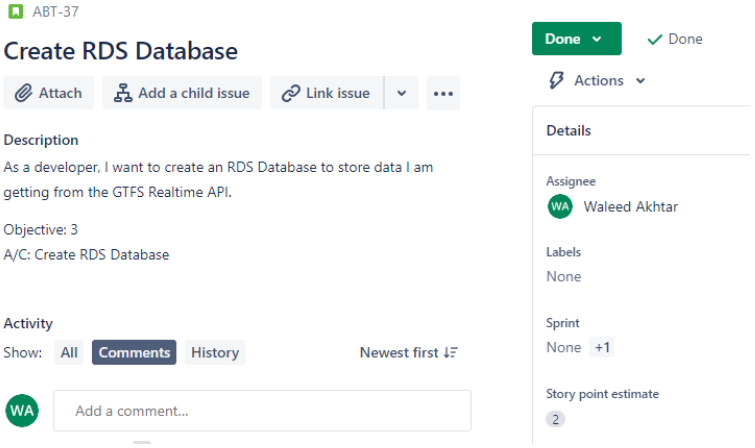

FIGURE 4.30: A screenshot of Story: Create RDS Database from Jira

4.6.19 Create EC2 Instance

In this user story, I want to create an EC2 Instance which will be used to connect to my RDS database.

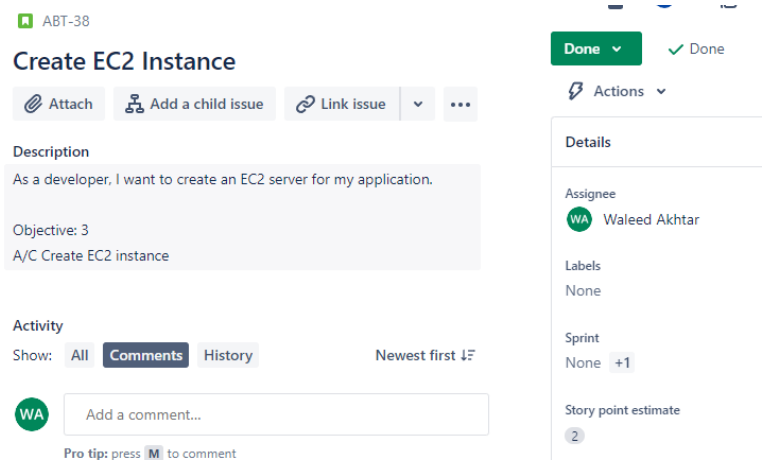

FIGURE 4.31: A screenshot of Story: Create EC2 Instance from Jira

4.6.20 Connect RDS Database with EC2

In this user story, I connected my RDS database with my EC2 instances which I created earlier.

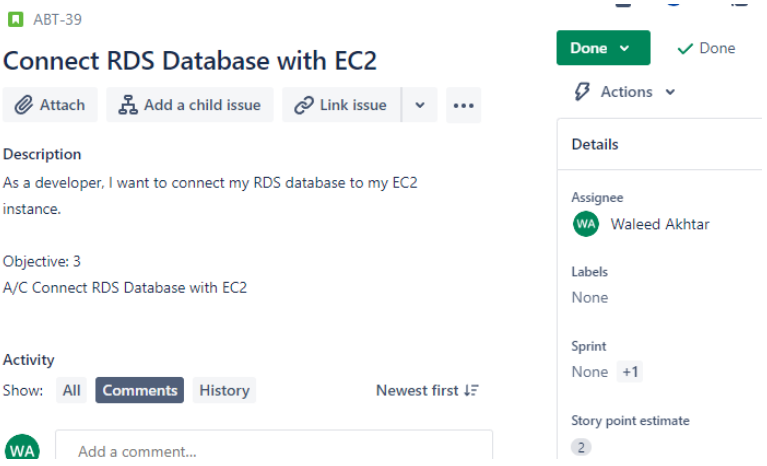

FIGURE 4.32: A screenshot of Story: Connect RDS Database with EC2 from Jira

4.6.21 Create schema for RDS Database

In this user story, I created a schema for my RDS Database.

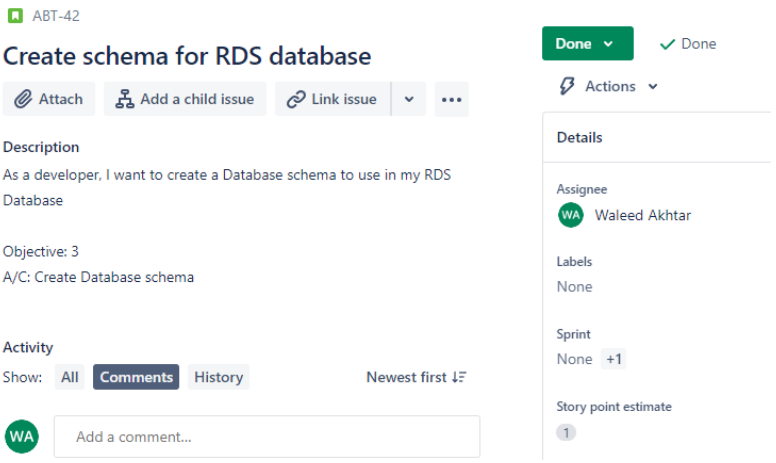

FIGURE 4.33: A screenshot of Story: Create schema for RDS Database from Jira

4.6.22 Connect RDS Database to MySQL Workbench

In this user story, I want to connect the RDS Database I made to my local MySQL workbench. This will allow me to make SQL queries and tables very easily.

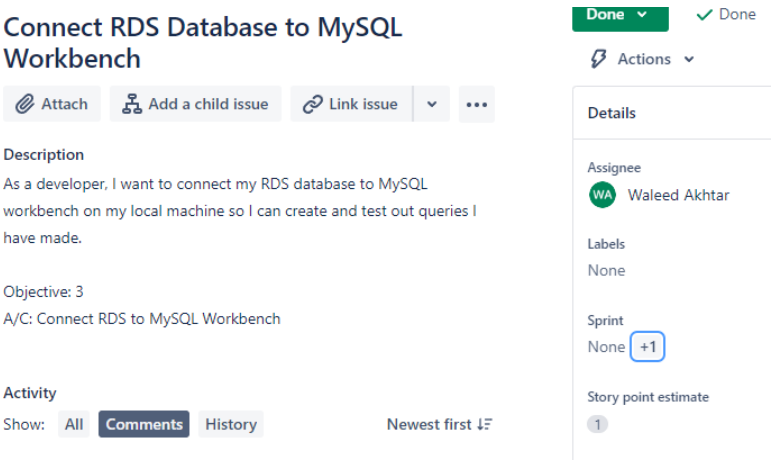

FIGURE 4.34: A screenshot of Story: Connect RDS Database to MySQL Workbench from Jira

4.6.23 Create AWS Lambda to call the API

In this user story, I created an AWS Lambda function that calls the GTFS API to get real-time bus information. Once it receives this information it will then store this information inside the RDS database which can then be used to retrieve data for the user.

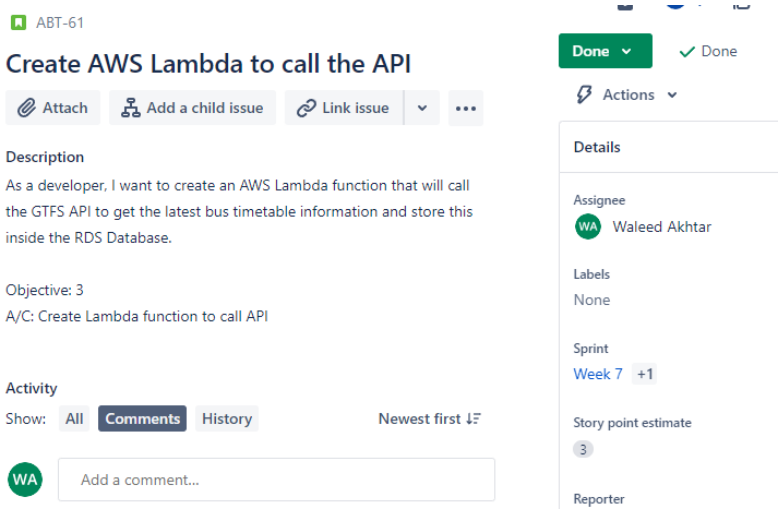

FIGURE 4.35: A screenshot of Story: Connect RDS Database to Alexa Lambda Function from Jira

4.6.24 Connect RDS Database to Alexa Lambda Function

In this user story, I want to connect the RDS Database to the Lambda function I made in the previous user story. This can then allow data to be added to the RDS database from the API.

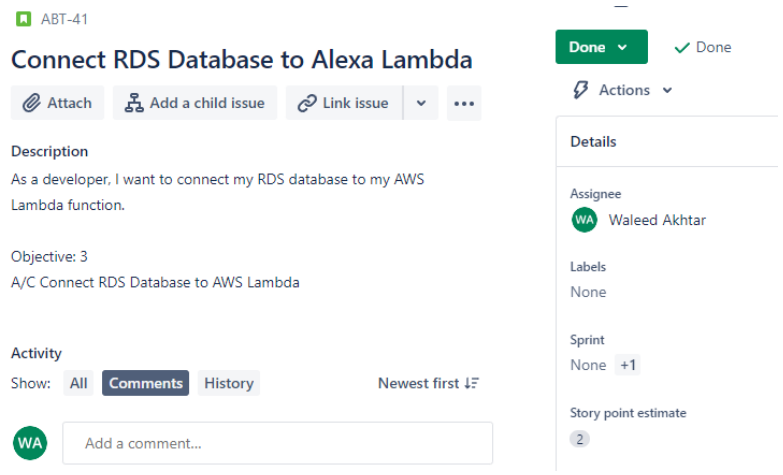

FIGURE 4.36: A screenshot of Story: Connect RDS Database to Alexa Lambda Function from Jira

4.6.25 Investigate VPC connectivity issues

In this user story, I want to investigate why I am unable to call the API. I have narrowed it down to an issue with my VPC inside AWS but I am unsure of what is causing it.

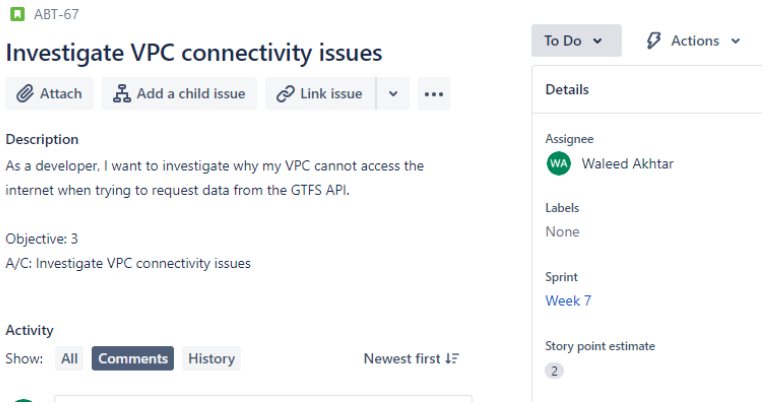

FIGURE 4.37: A screenshot of Story: Connect RDS Database to Alexa Lambda Function from Jira

4.6.26 Create CloudWatch Scheduler to call the API

In this user story, I want to create a CloudWatch scheduler that will automatically trigger my Lambda function to run once every 10 minutes. When the Lambda function is triggered it will then fetch data from the GTFS API and store it inside the database.

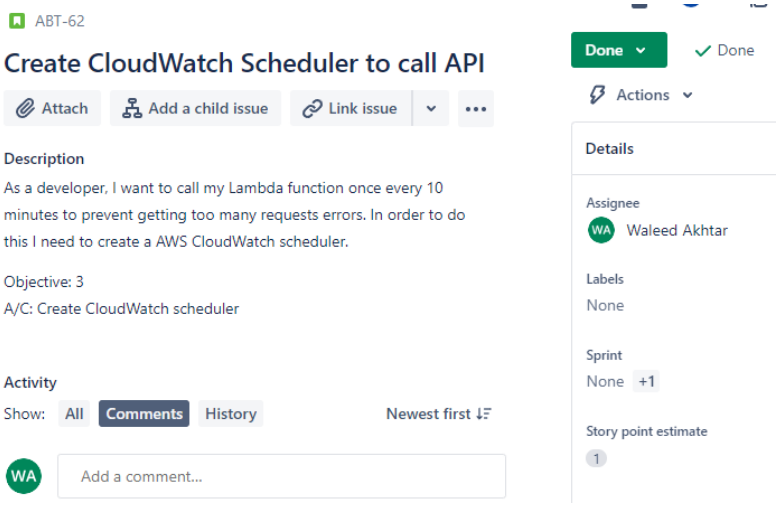

FIGURE 4.38: A screenshot of Story: Create CloudWatch Scheduler to call the API from Jira

4.6.27 Create IAM Roles

In this user story, I created IAM (Identity and Access Management Role) roles to be used by CloudWatch Scheduler to call the GTFS API.

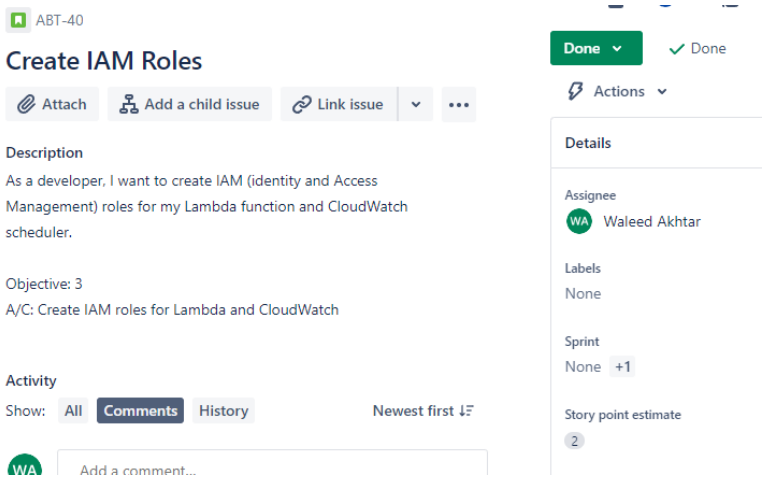

FIGURE 4.39: A screenshot of Story: Create IAM Roles from Jira

4.6.28 Create SQL query to find bus number

In this user story, I created the SQL query to find the bus number which goes to a user-specified destination.

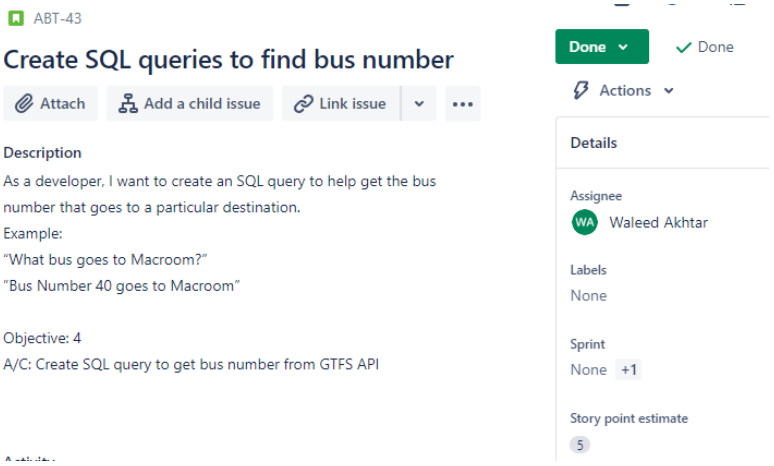

FIGURE 4.40: A screenshot of Story: Create SQL query to find bus number from Jira

4.6.29 Connect Bus Number Intent in Alexa Development Console

In this user story, I connect the SQL query I made in MySQL WorkBench to its respective intent inside teh Alexa Development Console.

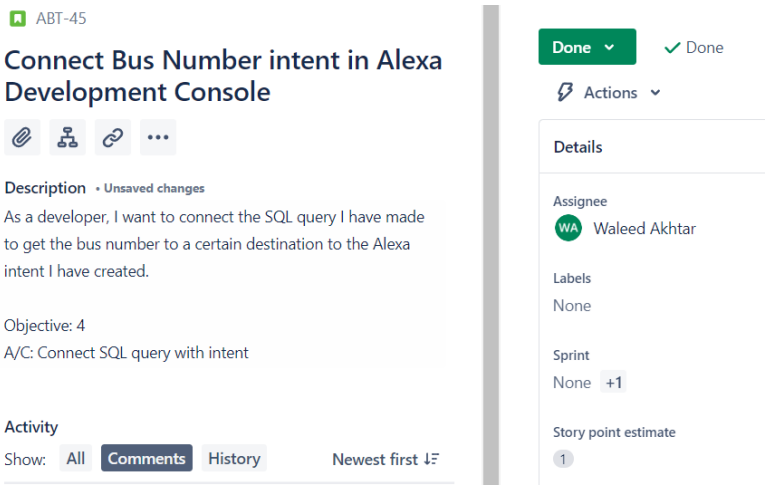

FIGURE 4.41: A screenshot of Story: Connect Bus Number Intent in Alexa Development Console from Jira

4.6.30 Create SQL query for next bus to area

In this user story, I created the SQL query to get the next bus that goes to a user-specified destination.

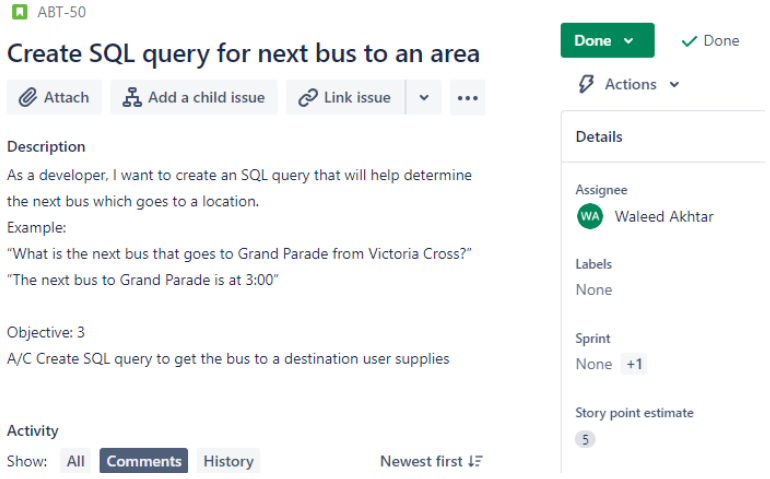

FIGURE 4.42: A screenshot of Story: Create SQL query for next bus to area from Jira

4.6.31 Connect SQL Query for the next bus to an area in Alexa Development Console

In this user story, I connect the SQL query I made in MySQL WorkBench to its respective intent inside the Alexa Development Console.

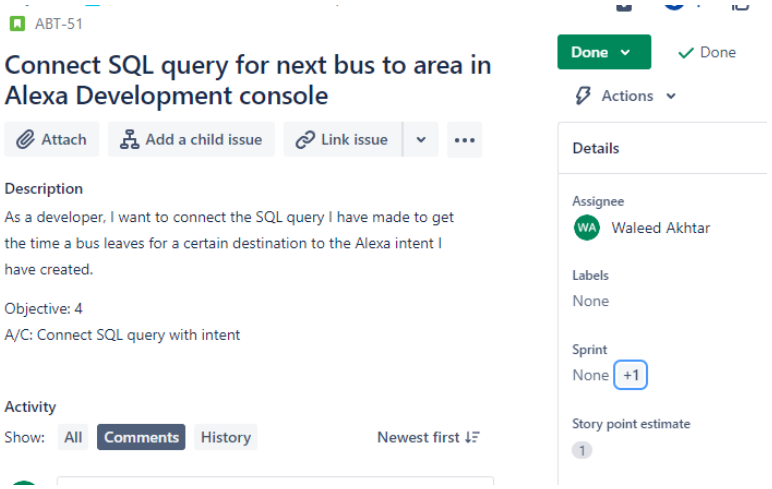

FIGURE 4.43: A screenshot of Story: Connect SQL Query for the next bus to an area in Alexa Development Console from Jira

4.6.32 Configure Speech to Text

When a user speaks to Alexa, Alexa must be able to understand the user's request through the use of utterances and slots. It must be able to understand different Irish accents to ensure everyone is able to use the application.

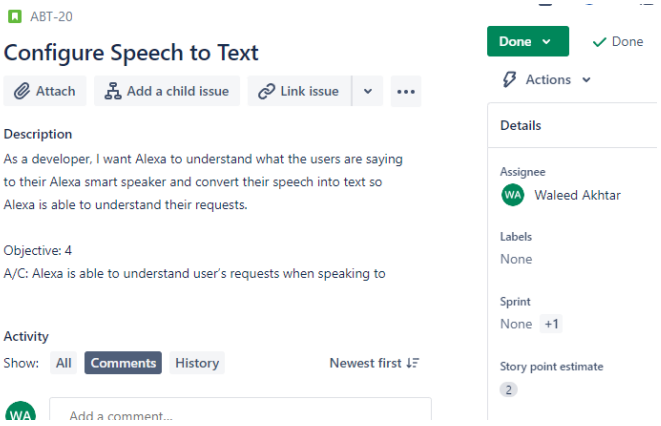

FIGURE 4.44: A screenshot of Story: Configure Speech to Text from Jira

4.6.33 Configure Text to Speech

When Alexa gets a response back from the RDS Database it must be able to parse the information into a sentence which it must read back to the user clearly.

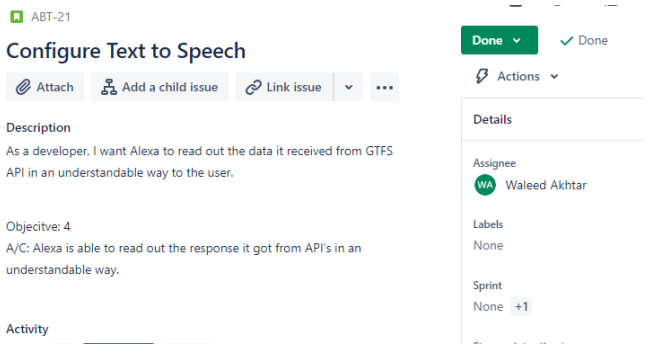

FIGURE 4.45: A screenshot of Story: Configure Text to Speech from Jira

4.6.34 Investigate Billing Charges

I started to receive emails from AWS saying I have exceeded the Free Tier usage limit and my account is getting charged. I need to investigate where are these charges coming from and how to prevent them from happening.

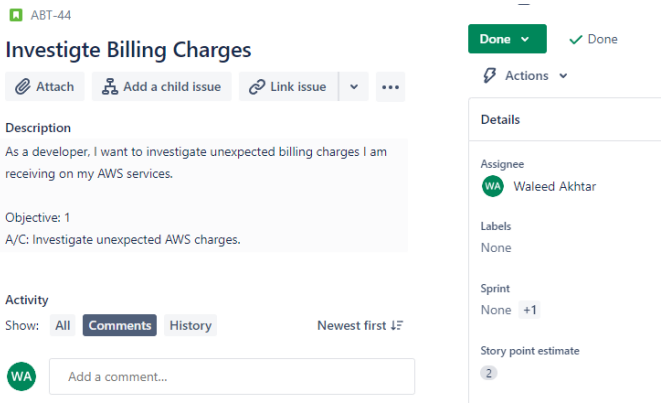

FIGURE 4.46: A screenshot of Story: Investigate Billing Charges from Jira

4.6.35 Call AWS Billing Support

As I was unable to stop the billing charges from AWS and could not find the root cause of the issue I decided to call AWS billing support to figure out the issue.

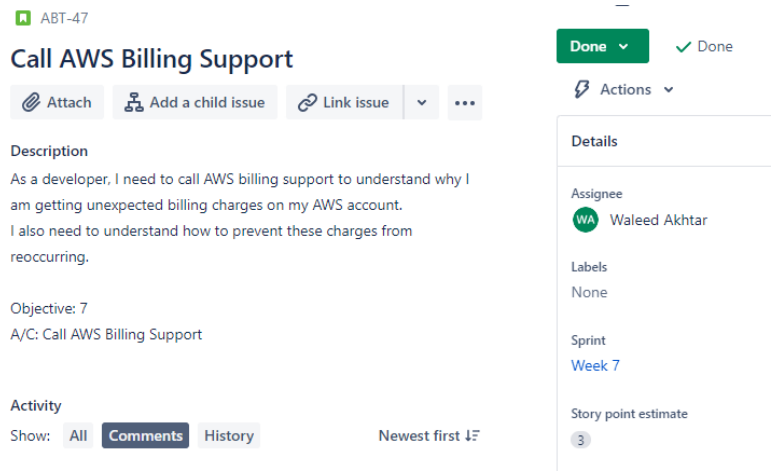

FIGURE 4.47: A screenshot of Story: Call AWS Billing Support from Jira

4.6.36 Make changes to AWS to comply with free tier

After the call with AWS billing support, I was able to understand where I had made the mistake that resulted in unexpected charges on my AWS account. The support agent advised me to make several changes to comply with the free tier restrictions which is what I will do in this user story.

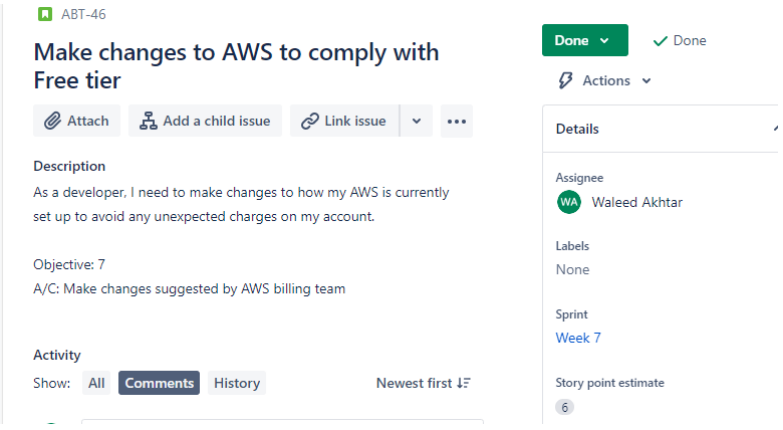

FIGURE 4.48: A screenshot of Story: Make changes to AWS to comply with free tier from Jira

4.6.37 Rebuild RDS Database

In this user story, I need to rebuild my RDS database again but this time complying with AWS free tier restrictions.

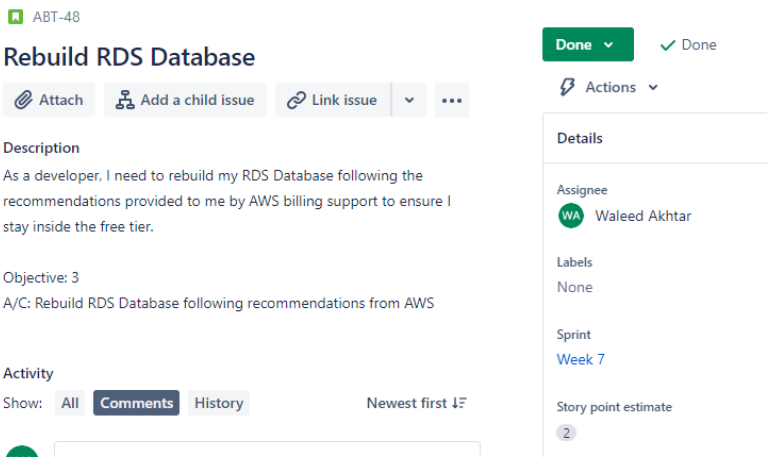

FIGURE 4.49: A screenshot of Story: Make changes to AWS to comply with free tier from Jira

4.6.38 Update references to use new RDS Database

Since I had to rebuild my RDS database the connection details to connect to my RDS database have changed. I need to manually change everywhere the old RDS database was referenced to use the new RDS Database.

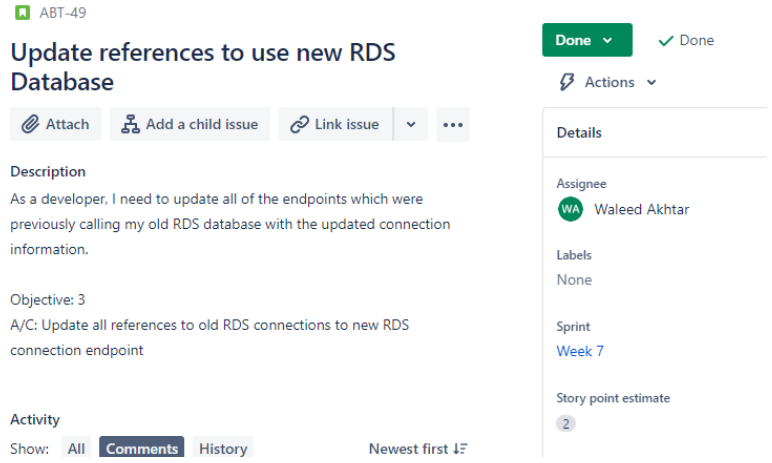

FIGURE 4.50: A screenshot of Story: Update references to use new RDS Database from Jira

4.6.39 Create SQL query for arriving by bus to an area by a certain time

In this user story, I created the SQL query to get the next bus that goes to a user-specified destination when they wish to arrive by a certain time.

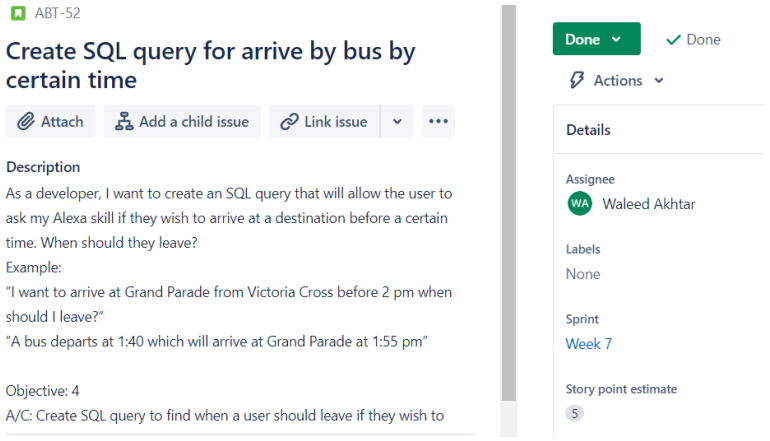

FIGURE 4.51: A screenshot of Story: Create SQL query for arriving by bus to an area by a certain time from Jira

4.6.40 Connect SQL Query for the arriving by to an area intent in Alexa Development Console

In this user story, I connect the SQL query I made in MySQL WorkBench to its respective intent inside the Alexa Development Console.

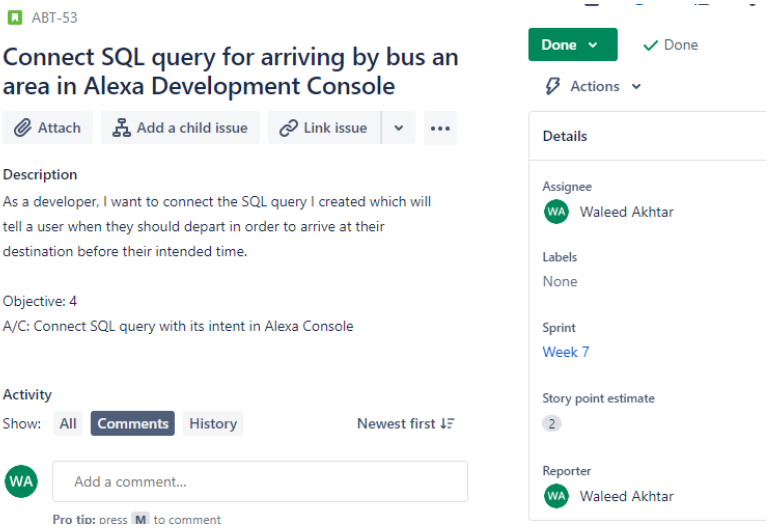

FIGURE 4.52: A screenshot of Story: Connect SQL Query for the arriving by to an area intent in Alexa Development Console from Jira

4.6.41 Create SQL query for when to depart by

In this user story, I created the SQL query to get the next bus that goes to a user-specified destination when they wish to depart by a certain time.

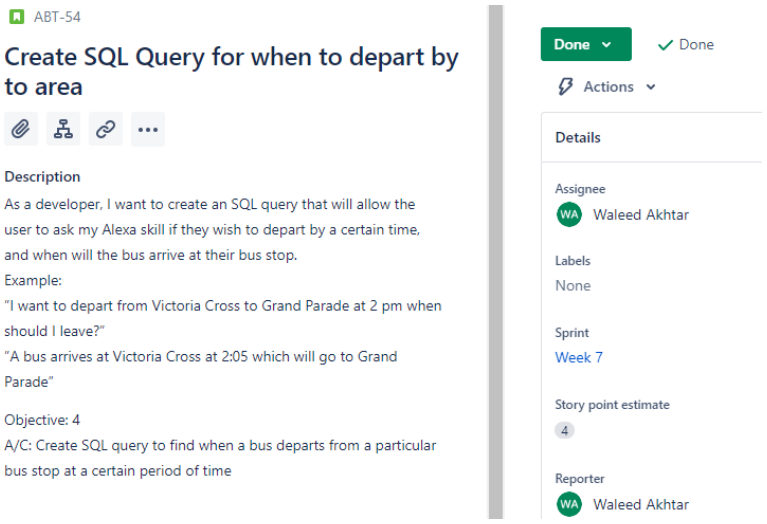

FIGURE 4.53: A screenshot of Story: Create SQL query for when to depart by from Jira

4.6.42 Connect SQL Query for the departing by to an area intent in Alexa Development Console

In this user story, I connect the SQL query I made in MySQL WorkBench to its respective intent inside the Alexa Development Console.

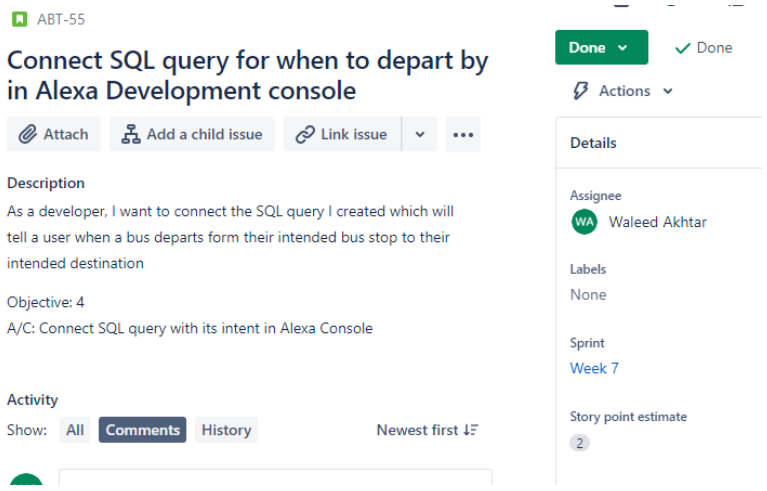

FIGURE 4.54: A screenshot of Story: Connect SQL Query for the departing by to an area intent in Alexa Development Console from Jira

4.6.43 Create SQL query to calculate the time difference between two bus stops

In this user story, I created the SQL query to get the journey time between two bus stops on the same route.

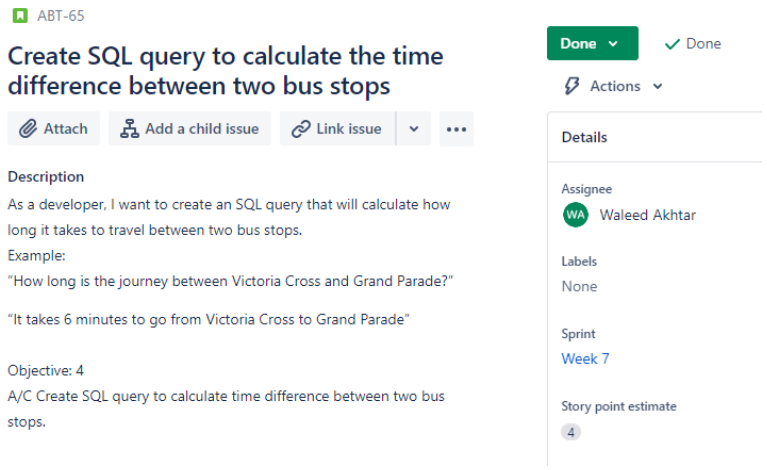

FIGURE 4.55: A screenshot of Story: Create SQL query to calculate the time difference between two bus stops from Jira

4.6.44 Create poster

In this user story, I created my poster to showcase my project.

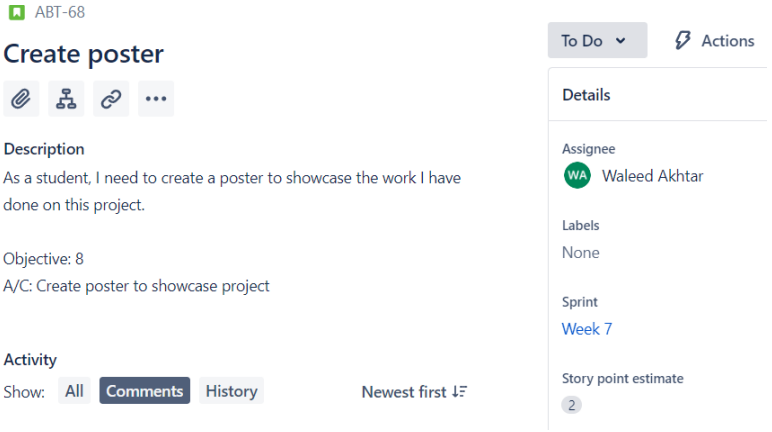

FIGURE 4.56: A screenshot of Story: Create poster from Jira

4.6.45 Make changes to poster

In this user story, I made changes to my poster as requested by my supervisor.

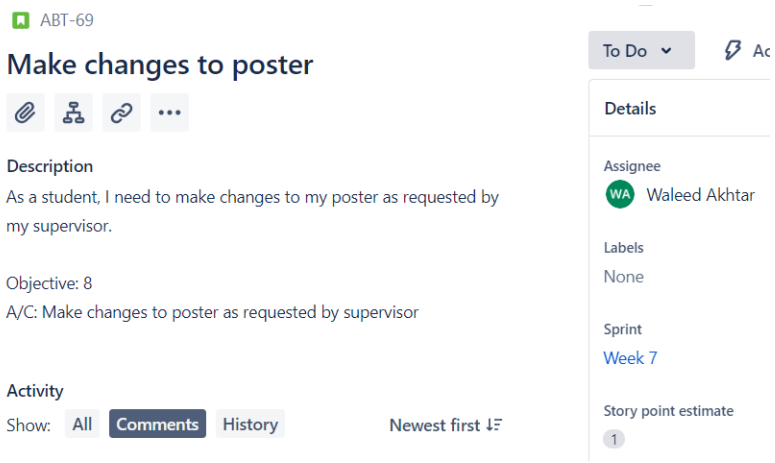

FIGURE 4.57: A screenshot of Story: Make changes to poster from Jira

4.6.46 Connect SQL Query to calculate the time difference between two bus stops intent in Alexa Development Console

In this user story, I connect the SQL query I made in MySQL WorkBench to its respective intent inside the Alexa Development Console.

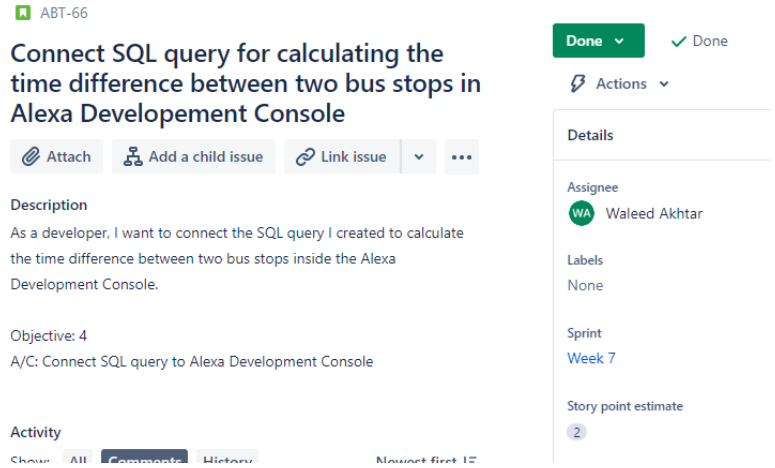

FIGURE 4.58: A screenshot of Story: Connect SQL Query to calculate the time difference between two bus stops intent in Alexa Development Console from Jira

4.6.47 Make happy path interaction model

In my interaction model with the user. I need to create a happy model path. This is where the user asks a valid question to Alexa Smart Assistant, which the assistant understands and is able to process the user's request. Then the assistant is able to give a valid response back to the user.

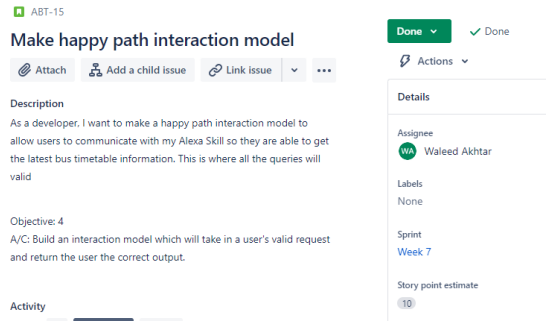

FIGURE 4.59: A screenshot of Story: Make happy path interaction model from Jira

4.6.48 Make exception path interaction model

In my interaction model with the user. I need to create an exception model path. This is where the user asks an invalid question to Alexa Smart Assistant. The Alexa skill should provide a valid response back to the user, requesting them to try again or try rephrasing the question asked.

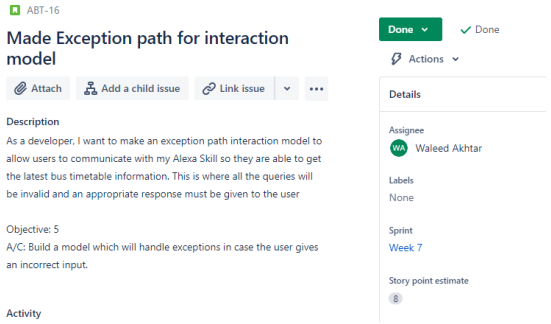

FIGURE 4.60: A screenshot of Story: Make exception path interaction model from Jira

4.6.49 Transition development environment to production environment

Before I can submit my code for certification to Amazon I need to transition my development environment so it is production ready. This involves getting rid of any comment blocks I may not need anymore, any obvious bugs have been fixed, and error handling has been implemented along with the rest of the functionality.

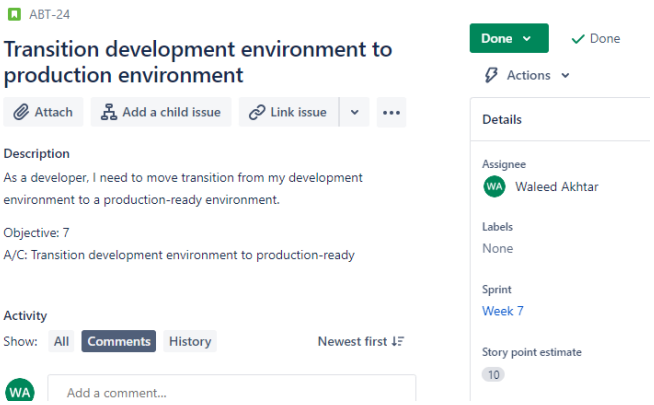

FIGURE 4.61: A screenshot of Story: Transition development environment to production environment from Jira

4.6.50 Beta test Alexa Skill

Once I have finished developing all the features for my Alexa application, I need to beta-test the application to ensure there aren't any major flaws or bugs which I may have overseen during development. This will also allow me to gain feedback from users about my Skill and make any modifications to it that could improve the user experience. Such as adding more utterances which I may think of when initially designing the application.

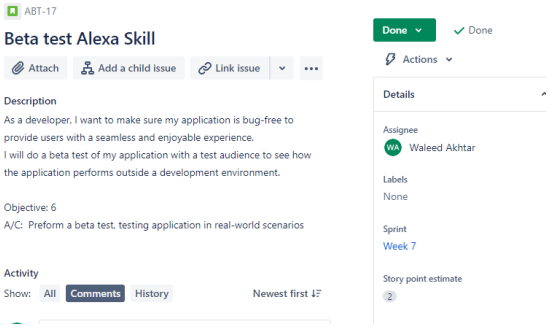

FIGURE 4.62: A screenshot of Story: Make exception path interaction model from Jira

4.6.51 Ensure Skill meets Amazon's Security requirements

I need to ensure my application passes all of Amazon's security requirements they have set in place for Alexa Skill developers. These are tests such as to prove I have valid trusted certificates when the connection is established with Amazon's servers and the requests being made are coming from Alexa devices.

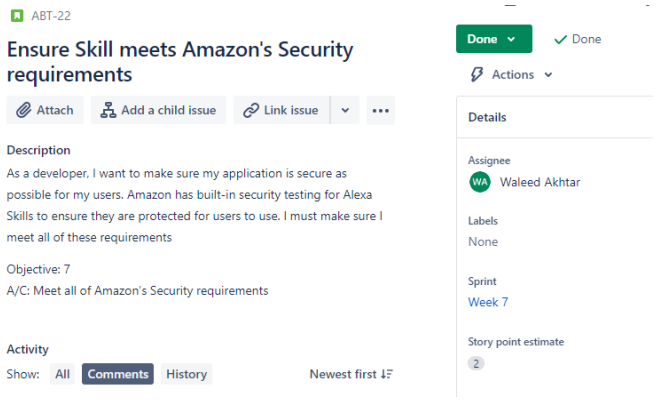

FIGURE 4.63: A screenshot of Story: Ensure Skill meets Amazon's Security requirements from Jira

4.6.52 Ensure Skill meets Amazon's Policy requirements

I need to ensure my Alexa skill adheres to all of Amazon's content guidelines. This includes that I disclose I will be using Transport For Ireland's GTFS API with my skill along with any trademarks and intellectual property I may use as well. I will need to indicate I have permission to use services when I submit my skill for certification.

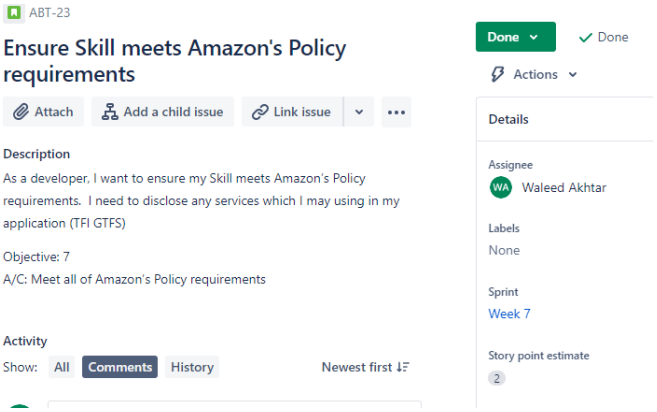

FIGURE 4.64: A screenshot of Story: Ensure Skill meets Amazon's Policy requirements from Jira

4.6.53 Create privacy policy

As part of the requirements set out by the Alexa publication team, I need to create a privacy policy for my Alexa in order for it to be published on the Alexa store.

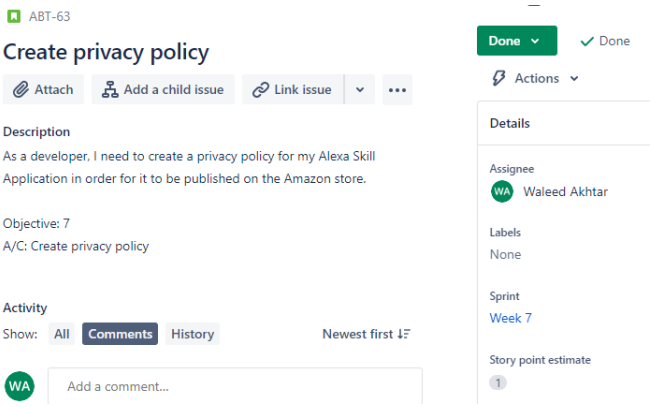

FIGURE 4.65: A screenshot of Story: Create privacy policy from Jira

4.6.54 Create Skill Icon

Another requirement set out by the Alexa publication team is that I create a skill icon. Users will be able to see the skill icon when they enable the skill on their Alexa device.

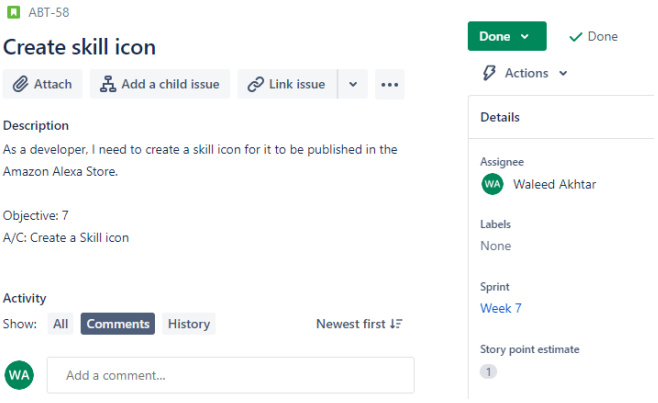

FIGURE 4.66: A screenshot of Story: Create Skill Icon from Jira

4.6.55 Submit Skill for validation

Before I am allowed to publish my Alexa Skill on Amazon's store I need to pass their validation test. This requires me to send my code to their validation tester application which scans my code for any obvious issues it may find. I will not be allowed to publish my Alexa Skill without getting my validation certificate prior so this story is crucial in terms of development for my application.

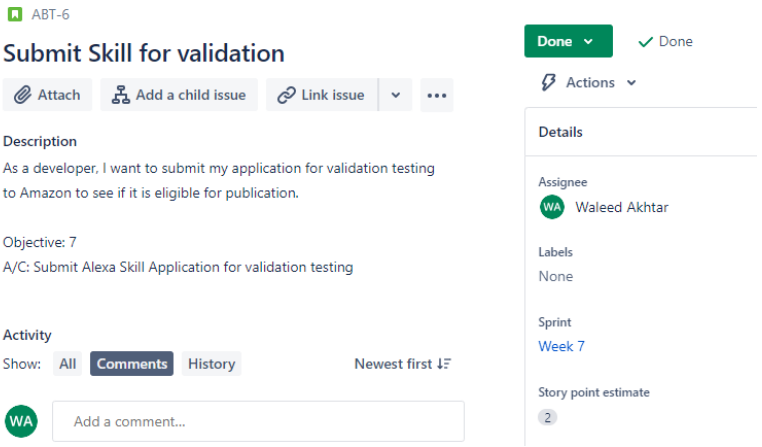

FIGURE 4.67: A screenshot of Story: Submit Skill for validation from Jira

4.6.56 Publish Alexa Skill

After I have implemented all of the features for my application and it has passed validation testing and beta testing. I will be ready to publish my Amazon Skill to the Amazon store where users can download my application for their personal use.

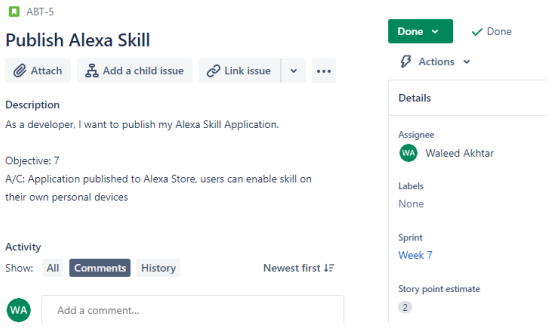

FIGURE 4.68: A screenshot of Story: Publish Alexa Skill from Jira

4.6.57 Make changes required by Alexa publication team

The Alexa publication team rejected my Alexa Skill Application and wants me to make changes to the application to improve the user experience of users when an error message is given to the user. I need to make my error messages more descriptive.

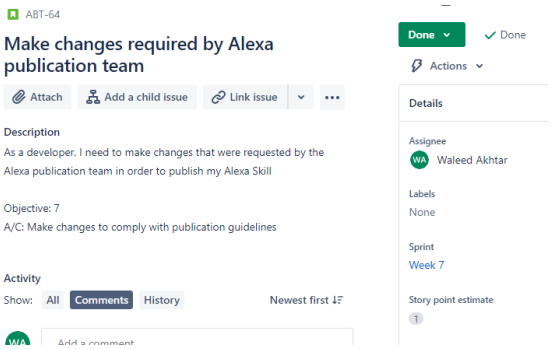

FIGURE 4.69: A screenshot of Story: Make changes required by Alexa publication team from Jira

4.6.58 Write Chapter 4 implementation phase

I will need to be writing my chapter 4, the implementation phase alongside developing the Alexa Skill. I will track how the progress of my application has been going during each sprint and any difficulties which have arisen during development.

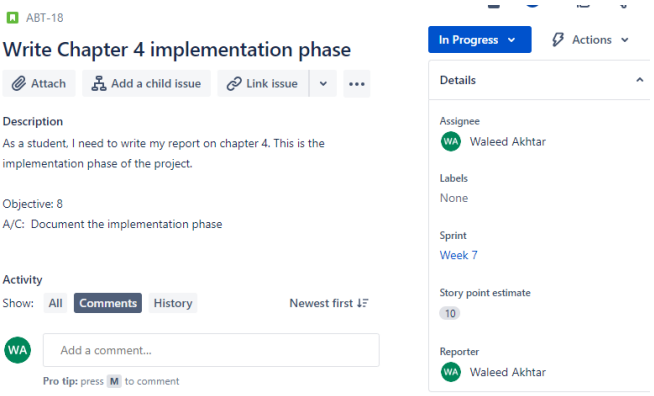

FIGURE 4.70: A screenshot of Story: Write Chapter 4 implementation phase from Jira

4.6.59 Write Chapter 5 evaluation

In this chapter, I will discuss how I think the project went and future improvements which could be added to the project later down the line.

FIGURE 4.71: A screenshot of Story: Write Chapter 5 evaluation from Jira

4.6.60 Write Chapter 6 conclusions

When my project is finally completed I will need to write down my conclusions about the project. In this chapter, I will discuss if my project was published on the Amazon Store and what were the reviews on the application. Or if I was not able to publish my application what bottlenecks prevented me from doing so.

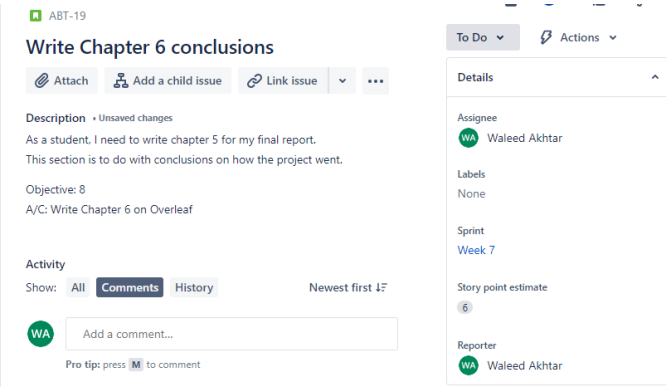

FIGURE 4.72: A screenshot of Story: Write Chapter 6 conclusions from Jira

Chapter 5

Testing and Evaluation

I believe from my testing of my Alexa Skill Application there is a lot that can be improved upon. Unless a user provides the perfect query which contains the full name of the bus stop as listed inside the dataset and it does not contain any abbreviations my Alexa Skill will return the user with an error message and ask them to try again.

This can lead to frustrations among users and may even cause them to stop using the Alexa Skill entirely. Below I will begin to go through the 5 intents I made for my Alexa Skill Application and test them accordingly with a happy path scenario and an unhappy path scenario.

Finally, I will explain how I was able to fulfil all of the objectives which I set in 1.3.

5.1 Finding Bus Number Intent

This intent is used to allow users to discover which bus they can take to get them to a certain destination that they specify. Users will need to provide a valid location and the Alexa Skill will return a bus number that goes to that location. A happy path example can be seen below going to Kanturk, Cork.

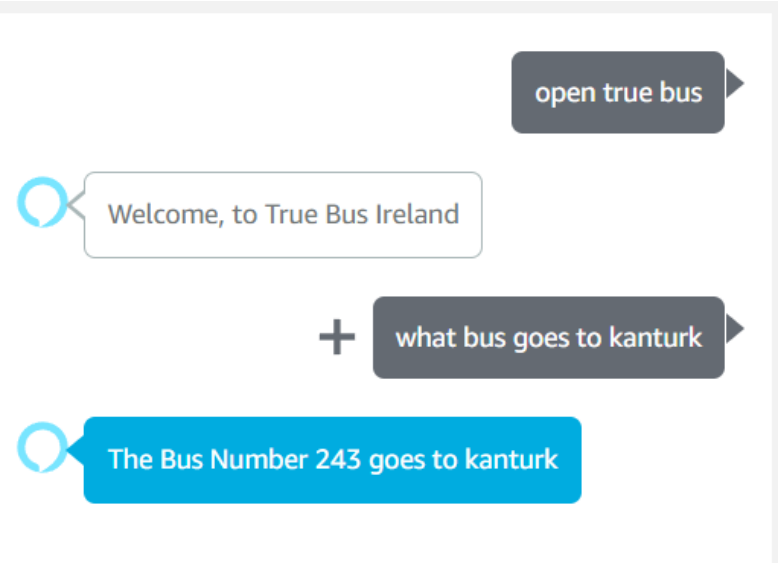

FIGURE 5.1: A screenshot of finding bus number intent going to Kanturk

However, an issue can arise when we look at destinations such as Dublin Airport where multiple buses arrive at it. When this happens the Alexa Skill returns the first bus route which appears alphabetically inside the dataset. Which in this case is the 101x buses that runs from Aston Village to Wilton Terrace as shown in figure 5.2.

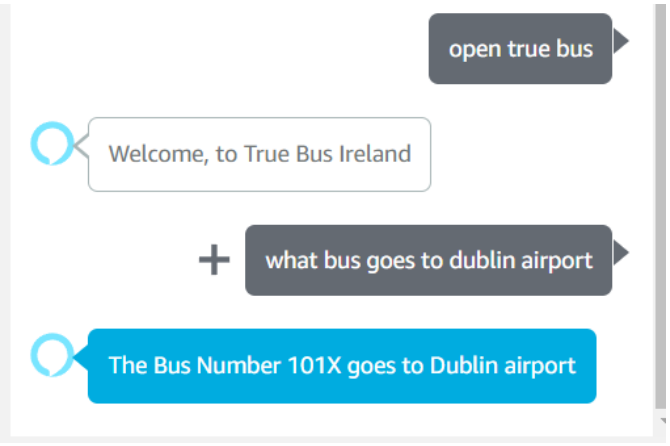

FIGURE 5.2: A screenshot of finding bus number intent going to Dublin Airport

This bus route would not be relevant to anyone who isn't living in Aston Village to Wilton Terrace. This can be fixed in future improvements by asking the user where are they departing from. Then the Alexa Skill can provide them with valid bus routes which would be relevant to them.

5.2 Next Bus to Location Intent

The purpose of this intent was to help users find out what is the next bus that goes to their desired location. Unlike the previous intent users need to specify their current destination. This makes the intent more relevant to users giving them the local information that they want. The users need to provide the full name of their intended destination and their source location. An example of a happy path scenario is seen below in 5.3.

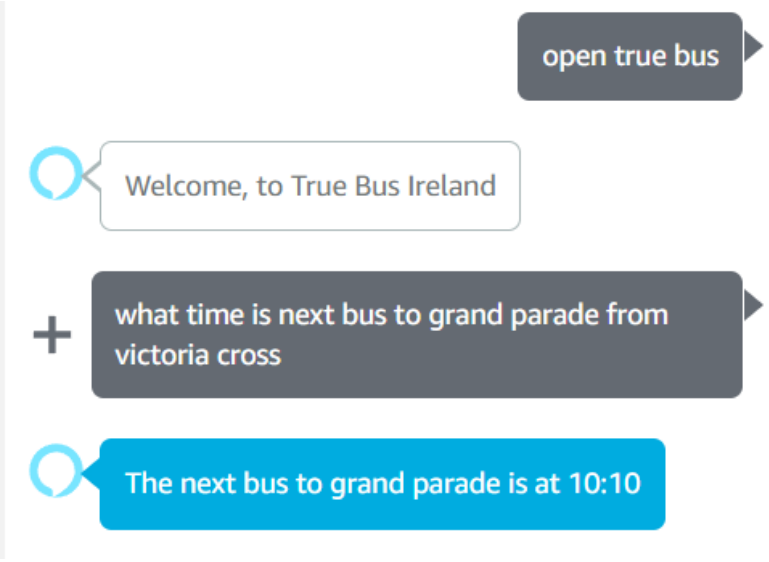

FIGURE 5.3: Next Bus to Location intent

A very big problem with this intent is that both the destination and source name need to be the same as it is written inside the dataset provided by the NTA. If there is any amount of deviation from the original name the Alexa Skill will return an error to the user. Unfortunately, this is the case more often than not.

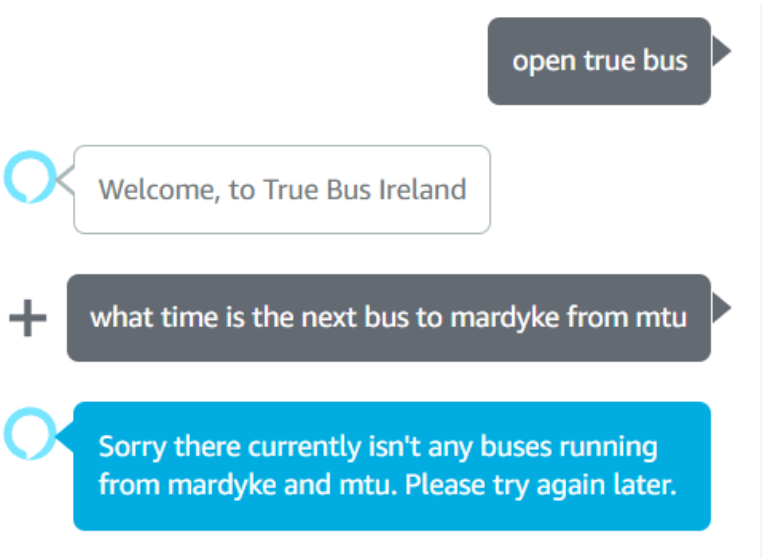

FIGURE 5.4: Invalid Next Bus to Location: Mardyke

Even though the user used the correct format of the intent, providing both the location and source of their destination the Alexa Skill still returned an error to the user because they said "Mardyke". When you check the dataset for Mardyke it comes up as "Mardyke Walk". If the user were to try again using "Mardyke Walk" as their destination the Alexa Skill will return a correct and valid output back to the user.

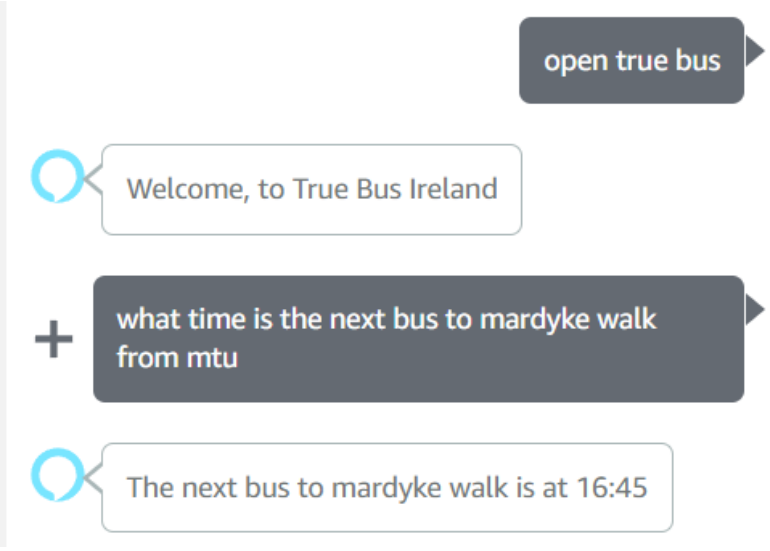

FIGURE 5.5: Valid Next Bus to Location: Mardyke

5.3 Arrive by time Intent

This intent can be used by users to discover when they should leave from their specified source to their desired destination at a specified time. This intent suffers from the same

problems as the previous one with regards to both the source and destination name needing to perfectly match the data specified inside the dataset. An example of the intent running successfully can be seen below.

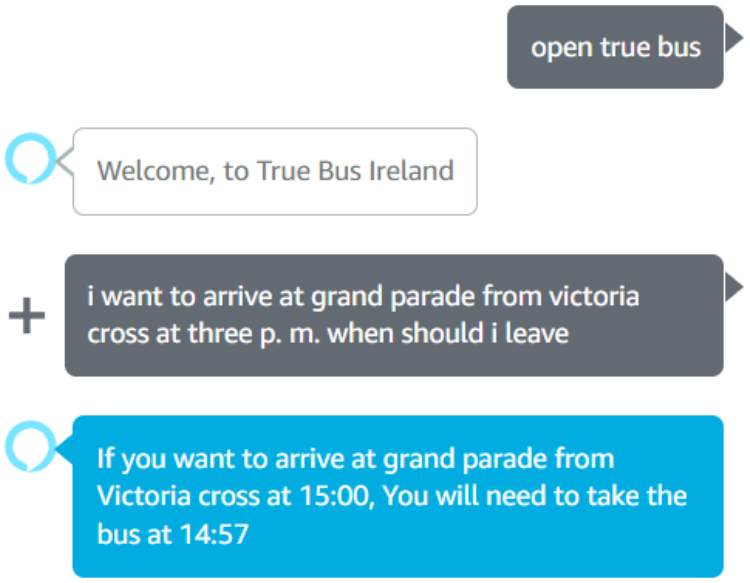

FIGURE 5.6: Arrive by Time intent

As this query is quite long there is a possibility the user can phrase the question in different ways which Alexa may not recognise. I have tried to anticipate different ways users can ask the same question inside my sample utterances for this intent however it is possible I may have missed a few. As a result, a user could try to invoke this intent and may not be able to due to different phraseology.

5.4 Depart by time Intent

This intent was created to allow users to discover which buses depart their specified location going to their desired location at a specified time. Like with the "Arrive By time" intent this intent suffers from the same problems such as the user phrasing the command differently or not using the full name of a location. A successful example of this intent can be seen below.

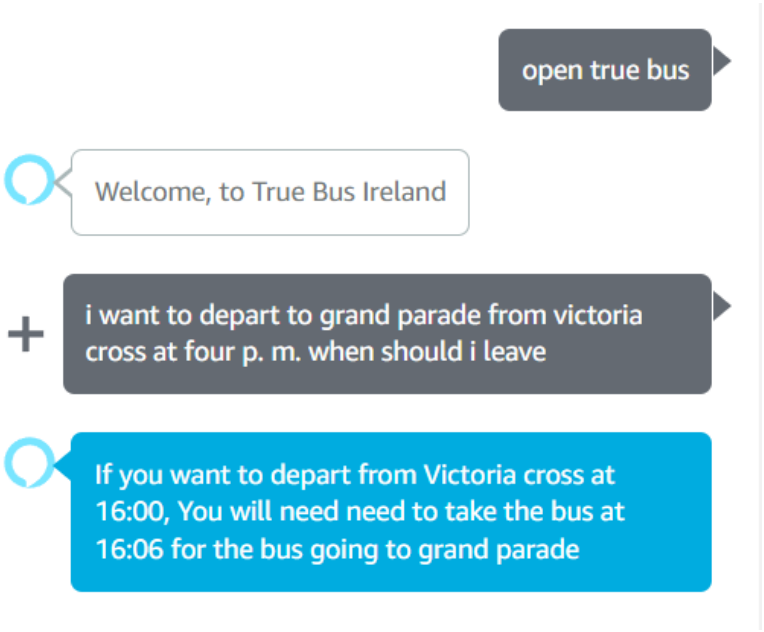

FIGURE 5.7: Depart by Time intent

5.5 Journey time calculation Intent

This intent tells users how long the journey time between two locations on the same bus route is. A common source of error using this intent could be users asking how long it takes to travel between two locations when a bus change is required. As this intent only works for trips on the same bus route. A happy path of this intent can be seen below.

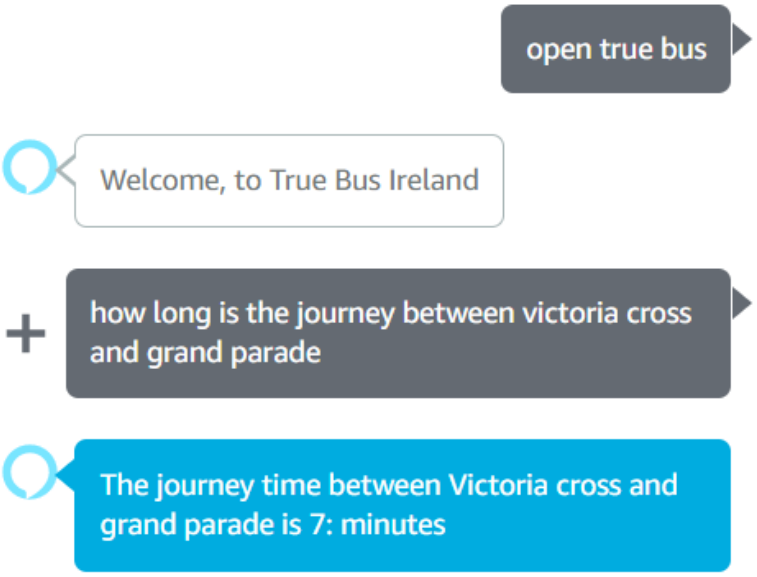

FIGURE 5.8: Journey time calculation intent

If a user were to ask for a journey that required a change in buses such as going from MTU to Douglas Shopping Centre the Alexa Skill will throw an appropriate error message.

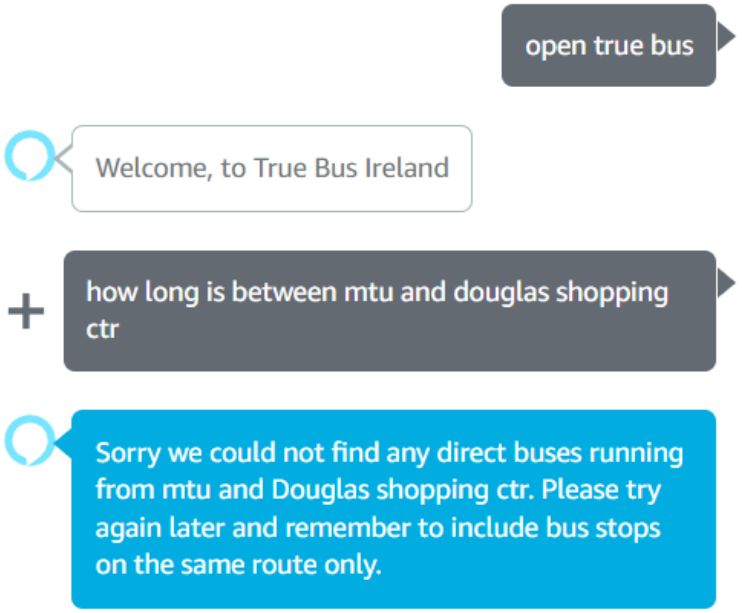

FIGURE 5.9: Journey time calculation intent exception example

5.6 Metrics

As this Alexa Skill was officially published on the Amazon store on the 28th of April not a lot of users have tried the Alexa Skill yet so it is difficult to get metrics on it. I have attached the metrics I have gotten from my Alexa Skill since its release below in figure 5.10.

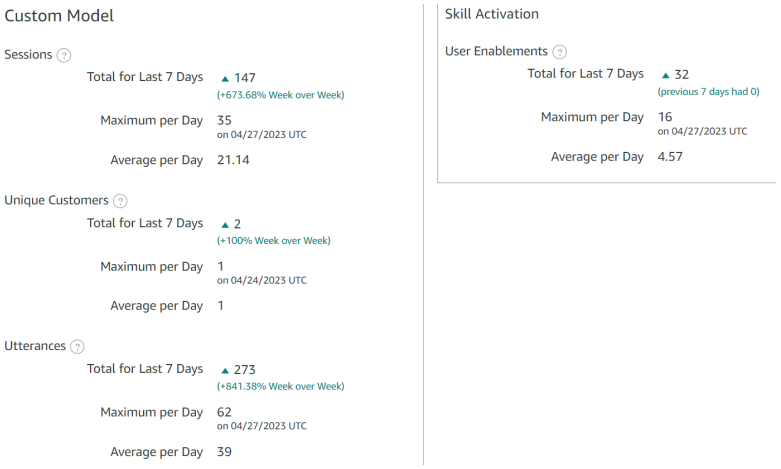

FIGURE 5.10: A screenshot of Skill Metrics

5.7 Objectives fulfilment

In this section, I will discuss the objectives I set out for this project in 1.3. I will demonstrate how these objectives were fulfilled during the implementation phase of this project.

5.7.1 Objective 1

I was able to complete objective 1 of the project as I have successfully built and published my own Alexa Skills Application.

5.7.2 Objective 2

I was able to successfully plan out my Alexa skill as shown in Chapters 2 and 3. Where I was able to identify risks, plan different data acquisition methods I could use, and developer technologies I was going to use. This can also be seen inside my Jira stories where I have marked the related objective as "2".

5.7.3 Objective 3

This objective can also be marked as fulfilled as seen in figure 4.10. I am getting data from the Realtime API over a 24-hour period which is then directly added to my RDS Database.

5.7.4 Objective 4

This objective is also fulfilled which can be seen from my conversations with the Alexa Skill application as seen above. For example in my Arrive By Time intent we can see the user asking a question and my Alexa Skill giving back a valid response.

5.7.5 Objective 5

This objective is also completed which can be seen when looking at my journey time calculation intent. When a user asks for the distance between two bus routes which are not direct. A well-detailed response is given to the user. As seen in figure 5.9.

5.7.6 Objective 6

I completed this objective and was able to find some oversights I had made with regard to error handling and was able to rectify them before publication.

5.7.7 Objective 7

This objective is fulfilled as my Alexa Skill has been published on the Alexa store which users in Ireland can enable on their Alexa devices for free. A link to the Amazon store page can be found here.

5.7.8 Objective 8

This objective can also be marked as complete. By submitting this report I will have logged all the work I have done in this project.

Chapter 6

Discussion and Conclusions

6.1 Solution Review

The aim of this project was to create an Alexa Skill application that will provide real-time bus information to users in Ireland. This will help reduce the intent to action for accessing bus timetables.

While I may have been successful in publishing in Alexa Skill on to the Amazon Alexa Skill Store I have not been successful with regards to reducing the intent to action except in a handful of scenarios.

My solution does help solve the problem of reducing the intent to action for accessing bus timetables if the user provides the full bus stop as written inside the dataset NTA provided.

However, if the user does not provide the correct full name of the bus stop then they are prompted with an error message which can lead to users becoming frustrated and stop using the Alexa Skill.

6.2 Project Review

If I could redo this project, there would be a lot I would have done differently. One of the first things I would do differently is put a lot more emphasis on research during the research phase. It would have been very helpful if I knew services from AWS such as RDS prior which would have made setting up and configuring the database much easier. Instead of discovering it by chance when I was comparing DynamoDB vs Python as two different architectures for getting data.

I would have also liked to spend more time researching the AWS Free Tier. Knowing what services are and are not included inside the service would've been a massive benefit and prevented a lot of time wasted debugging what is causing the unexpected billing charges.

I would also allocate more time to research and understanding how Lambda functions and the Alexa SDK work. I thought I had a decent enough understanding of these two services enough to start developing. However, it became very apparent I was lacking knowledge and wasn't sure how to use them in complex scenarios.

Finally, I would spend time researching and experimenting with alternate ways I could solve my dataset issue with users having to say the full exact name of a bus stop in order for the Alexa Skill to work. I believe with more time and with the knowledge and experience I have gained working I could come up with a viable solution that can resolve this issue.

6.3 Lessons Learnt

I have learned quite a lot when working on this project. Some new things which I have never used before such as cloud services with AWS and others I have already used before like Python and SQL.

6.3.1 AWS

I learned the most from working with Amazon Web Services, as it was something I had never used before. After working with AWS during the implementation stage of this project I have found a newfound interest in it. I understand why cloud services like AWS are needed in organisations across the world and the important role they serve. I found my experience working with AWS to be very educational and informative and would definitely use it again with another project if required.

One important thing I did learn from AWS is to check what services are and are not included in the Free tier. If I am using AWS again I would double check every service that I use is included in the free tier. There are even some services in the free tier which are free up to a certain value. Once you exceed that value you will start to get charged for it.

6.3.2 Alexa SDK

In this project, I had to heavily interact with the Alexa SDK. As I have never used this SDK before I had to learn quite a bit to fully master and understand it. There is quite a steep learning curve with the SDK. I learned it is quite important to look at the Python examples Amazon provides and understand what they are doing. I discovered Amazon created a full GitHub repository of Python examples using their SDK which can be very helpful in understanding how the SDK works.

6.3.3 Project Management Tool

During the course of this project, I realised the importance of tools like Jira and why they are needed in the industry. I was able to track what user stories I need to work on next, keep track of my progress and help me prioritise important tasks which would arise. Through the use of Jira, I was able to follow agile methodologies during the implementation phase of this project.

6.3.4 API

Throughout this project, I had to interact with the GTFS API. Since the beginning of this project, I have become very familiar with working with this particular API. I come to understand some restrictions this API has and how to bypass them.

6.3.5 Python

While I have been using Python since 1st year of university, I feel as if I learnt a whole lot more and became familiar with Python more than ever before since working on this project. It was definitely a new experience using Python inside a SDK but I soon realised it isn't much different from using Python normally. I eventually reached the stage where I could use both the SDK and Python comfortably during the implementation of this project.

6.3.6 Risk analysis

Another crucial learning moment for me was the importance of doing proper risk analysis. Along with planning backup plans in case something goes wrong. It is important to do planning for anything to go wrong and possible solutions to use when things eventually do fail. As no project implementation ever goes smoothly. Problems always can

always arise and often than not are not in your control. So the best you can do is plan around them in case they do inevitably rise.

6.4 Future Work

If I had more time for this project the first thing I would do is try to fix the bug where users do not have to say the full name of a bus stop. I could do this by developing an "alias" for some bus stops. For example the bus stop in Grand Parade opposite Centra where the old Argos used to be referred "Daunt Square" on the dataset provided by the NTA. I could create an alias for this location and call it "Grand Parade North".

This project can also be enhanced by allowing more support for operators other than just Bus Éireann. It can be expanded to support other bus services in Ireland such as Dublin Bus or Citylink. If more operators were supported this would allow it to have a greater reach across users all over Ireland.

This Alexa Skill Application can be expanded even further in the future to support not only just buses but also other transportation mediums running in Ireland. The NTA recently made a change to the GTFS-R API which now supports Irish Rail and LUCS. This Alexa Skill application can transform from not only a Bus timetable chatbot but even further into an all-in-one public transportation chatbot. An all-in-one chatbot can allow me to attract the largest possible audience available in Ireland who could use the Alexa Skill application.

6.5 Conclusion

In conclusion of this project, although I was not able to reduce the intent to action for users accessing bus timetables in Ireland, I was able to have a very fulfilling learning experience. I learned the importance of proper planning above all else and why project management tools such as Jira exist. I was able to create a newfound interest in AWS and would definitely pursue using it in the future.

I also learned a lot about conversational agents during this project. I came to understand how conversational agents use natural language processing to extract important parts of a sentence we humans speak and then use it to respond to user queries. This can be seen through the use of slots and utterances in Alexa. Alexa is able to extract the what are slots from a user's request and pass them to my application. Which I can then use to provide a response to the user.

All of this amazes me and we have this technology already available to us. I am excited to see where the future of conversational agents goes and how we will be interacting with them in the future.

Bibliography

[1] U. S. P. SERVICE, "Missile mail," July 2008. [Online]. Available: https://about.usps.com/who-we-are/postal-history/missile-mail.pdf

[2] Y. K. Jinjoo Bok, "Comparable measures of accessibility to public transport using the general transit feed specification," *Sustainability*, no. 3, p. 224, March 2016. [Online]. Available: https://www.mdpi.com/2071-1050/8/3/224

[3] "Bus times." [Online]. Available: https://www.amazon.co.uk/Freepoc-Bus-Times/dp/B07MN6MP6F

[4] "Next bus." [Online]. Available: https://www.amazon.co.uk/Jesus-Rodriguez-Next-Bus/dp/B01N488SN5

[5] T. Warren, "Microsoft shuts down cortana on ios and android," August 2021. [Online]. Available: https://www.theverge.com/2021/4/1/22361687/microsoft-cortana-shut-down-ios-android-mobile-app

[6] S. Fathi, "ios 15 to limit siri functionality with third-party apps," July 2021. [Online]. Available: https://www.macrumors.com/2021/07/28/ios-15-limiting-siri-for-third-party-apps/

[7] M. Bronstein, "A more helpful google assistant for your every day," January 2020. [Online]. Available: https://blog.google/products/assistant/ces-2020-google-assistant/

[8] J. Wong, "Leveraging the general transit feed specification for efficient transit analysis," *Transportation Research Record*, no. 1, p. 11–19, 2013. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3141/2338-02

[9] N. T. Authority, "Attention developers. upgrade to gtfs-realtime api," March 2023. [Online]. Available: https://www.nationaltransport.ie/news/attention-developers-upgrade-to-gtfs-realtime-api/