Alexa for Citizen Science

Donald-John Dorren
190003633
AC40001 Honours Project
BSc (Hons) Computing Science
University of Dundee, 2023
Supervisor: Dr Brian Plüss

Word Count: 16,686

Abstract
Citizen Science is a method of gathering data using the general public and has been used since the early twentieth century. Over the past couple of decades, thousands of citizen science projects with millions of participants collecting data have been completed all around the world. Citizen science projects often struggle with recruiting and maintaining data collectors. By capitalising on the massive growth of voice assistants we can massively increase the pool of possible data gatherers as well as making the submission of data easier and more efficient. The goal of this project is to allow users of Amazon's Alexa voice interface assistant to submit data using their Alexa through the use of custom Alexa skills. The purpose of this report is to outline several aspects of the Alexa for citizen science project. Covered in this report is research conducted, a description of the system developed, a critical appraisal of the project overall, reflections on past work completed and finally suggestions for future work.

1. INTRODUCTION

Although the method of collecting data through public data gathering has been happening since the early twentieth century when Wells Cooke selected amateur bird enthusiasts to collect data on bird migration [1]. Citizen science projects often struggle with recruiting and maintaining data collectors. By using Alexa we hope to make data gathering and submission easier. Alexa is a voice interface assistant this system can be used to collect Citizen Science data. Data can be added through the voice interface such as: "Alexa, tell the project it's raining".

The motivation of this project is to allow all users of voice interfaces to easily submit data allowing wider communities to collect data. This will allow any citizen science project to gain more data collectors and therefore more varied data.

To combat this problem a system will be built to allow citizen scientists to submit data on the current weather they are experiencing at their location using an Amazon Alexa. There are three main aspects of this project, the Alexa skill, the database and the website. This system will allow users to report on weather such as rain, cloud and sun as well as other aspects such as noise pollution and high pollen counts. This data will be collected using a custom Alexa skill which will also pull useful data from the physical Alexa such as the time and location. Data will be stored in an AWS database to allow a webpage to pull this information and display it graphically.

It is expected that through the use of both qualitative and quantitative evaluation methods the project will meet all requirements in building a system that allows users to submit data to a citizen science project using an Amazon Alexa.

Achieving this solution will help Citizen Science projects such as GROW to collect data from a higher number of data gatherers. Using Alexa we can improve the data gathering efficiency allowing for easy storage in a database to allow for analysis.

2. BACKGROUND

2.1 What is Citizen Science

At the time the author chose this topic as their honours project the title was "Alexa for citizen science", this is a very broad title allowing for many different understandings of the final goal. This allowed the author to fully take control of the direction they wanted the project to head in from the beginning. As this was a broad definition and title of the project the research started from the very beginning, what is citizen science?

Citizen science is a type of scientific research in which members of the public actively take part in gathering and submitting data [2]. It is most often used for biology and conservation. Citizen science can collect data through its participants by having them help collect or classify data, improving the data-capturing capacity. There are several problems with citizen science projects such as; Data collection through Citizen science can be problematic as data collection is usually region based, as some areas might have larger quantities of contributors than others. Another concern is that the data collected might not be accurate enough for full scientific testing.

2.2 Existing solutions and demand

A key aspect of the first stages of research was to determine if there was a demand for a system like the one to be developed and if a solution to this problem had already been developed. At the point of writing this report, there are a couple of Alexa skills that have been developed to submit data to citizen science projects. The first is the “Did you feel it” [3] by the United State Geological Survey, although this system isn’t fully operational yet the idea is that users can add information to the USGS database on earthquakes and also query the data using an Amazon Echo with questions such as “Alexa when was the strongest earthquake in California this year”. An interesting feature of the system is that users can see a visual representation of the data stored in the database. This is shown on the website “Did you feel it? Annual Data Viewer”, although this system isn’t very user-friendly and isn’t finished.

The next similar system available at the time of development is “Bird Spotting” by Smarthouse UG[4]. There is very little information about this skill on the Amazon page and no reviews so it is hard to understand what this project is truly about. At the time of writing this report unfortunately this Alexa skill doesn’t work. Although this skill doesn’t work it still shows that there are people interested in making voice interface systems for citizen science.

It is hard to measure demand for a product when there are very few existing products in the market, but over the last couple of years, the use of virtual assistants has risen massively. It is estimated that in 2019 there were over ninety-eight million units sold worldwide, and this growth is expected to continue to around four hundred and nine million individual units by 2025[5]. Within this growth, a notable aspect is a growth in older adults using voice assistants as they do not demand high levels of technical understanding because of their conversational interface. Although the rate of adoption in this age category is less than that of 18-29 year olds this is generally due to older adults' slow rate of technology adoption[6]. The reason this is important is that the most likely participants in citizen science projects are middle-aged and the elderly[7]. Targeting this age group in evaluation would be ideal but maybe not be possible due to time constraints and accessibility of both the application and access to testers.

An example citizen science project is GROW [8]. GROW Citizen Science, which currently uses physical devices to measure moisture content in the soil, has the goal of improving crop growing conditions and climate change. If a project like GROW could recruit users of voice interfaces to submit weather data they could easily use this data in partnership with the physical devices. The physical devices are more accurate but have higher initial build and installation costs compared to a virtual system. There are hundreds of citizen science projects like GROW that would massively benefit from data collection through the use of voice assistants not only because of the cost benefits but also the scalability. This is why there would be a demand for a voice interface for citizen science projects.

2.3 Voice Assistants

Although the title of this project depicts that it should use Amazon’s Alexa it is still important to explore other voice assistants, their development environments and user preferences. This research allowed for a greater understanding of voice assistants and standard practices used when developing voice assistants.

Smart voice assistants have a surprisingly long history, but it was only recently that their popularity grew rapidly. The first big voice assistant was Apple’s Siri, introduced to the public in February 2010 as an independent app but was acquired by Apple in April 2010 and added as a standard feature of all Apple devices. The next big addition to the voice assistant market was Microsoft's Cortana but this didn't come until 2014, later that year Amazon's Alexa was released which quickly became the most used voice assistant and captured roughly 70% of the Western market share of smart speakers. Finally, in 2016, Google launched Google Assistant which was integrated with every Android phone and was also available on competitor's devices through the use of an app. Although Google Assistant only has about a 20% market share of the smart speakers market it has a large market share in the general use of voice assistants as it is built into the most products.

2.4 Selecting Amazon Alexa

After conducting the research covered in section 2.3 it was determined that Amazon's Alexa was the best choice for both the development environment, access to user testers and ease of use for users. Another key factor in choosing Amazon's Alexa is its development console and easy integration into AWS services such as DynamoDB, IAM, and S3.

2.4.1 Alexa development console

Amazon's Alexa Development console allows developers to build in a search engine such as Chrome or Firefox or develop in an external code editor and publish the code to the development console once complete. This was a key factor in the selection of Amazon's system as it would allow for rapid prototyping, testing and development inside the web-based system while also allowing for the development of more complex systems in familiar software such as Visual Studio Code when better debugging is needed.

2.4.2 Alexa Skills Kit

The Alexa Skills Kit (ASK) is a collection of APIs, tools, and resources provided by Amazon that developers can use to create custom skills for Amazon's Alexa. With ASK, developers can build voice-based apps, or "skills," for a range of use cases such as playing music, ordering food, checking the weather, or controlling smart home devices. ASK provides a range of tools, including a software development kit (SDK), sample code, and documentation to help developers get started with building skills for Alexa. Additionally, the ASK supports a range of programming languages, including Node.js and Python which were used in the prototyping and development of this project.

2.5 Amazon Web Services

AWS is a cloud computing platform offered by Amazon that provides a wide range of services for both individuals and businesses. AWS offers the ability to store, manage and process data as well as to deploy applications and websites to the cloud. There are over 200 features offered by Amazon Web Services but for this project, the only necessary services are storage and web hosting which will be done using DynamoDB and S3 although there were other services used to link these.

AWS operates on a pay-as-you-go pricing model, which means that customers only pay for the services they use, this was another key factor in choosing Amazon Alexa. This pay model is ideal for businesses of any size as there's no need to invest in expensive hardware. AWS even offers grants to small businesses to help them scale their platforms with little worry about surprise costs.

2.5.1 DynamoDB

DynamoDB is a database service offered by AWS which is a fully managed, serverless, key-value NoSQL database. DynamoDB has the ability to run high-end applications at almost any scale from storing basic user data to storing game platforms with player data and leaderboards of millions of users. There are many large applications that rely on DynamoDB such as Disney+, Dropbox, Snapchat and Zoom. Another reason DynamoDB was selected was its easy integration with Alexa and the Alexa

Development Console. Each Alexa skill has an integrated DynamoDB table that can be used for testing (See figure 1), this table is tied to the Alexa skill and has limited capabilities due to IAM restrictions. This integrated DynamoDB table was used when developing prototypes as it allowed for quick and easy testing of adding data to tables through the use of Alexa.

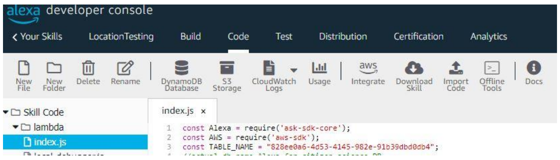


Figure 1: Photo of the integrated services (DynamoDB Database, S3 Storage and Cloudwatch Logs) offered for each individual Alexa skill within the Alexa Development Console. As shown on line three of index.js you can see the table name is a random string which corresponds to the individual skill and can only be accessed by that skill.

2.5.2 Identity and Access Management

Identity and Access Management or IAM is a service provided by AWS that allows a user to manage access to other AWS resources securely. IAM provides a centralised control area for overall resources, allowing a user to manage user accounts, permissions and access. With IAM a user can create and manage user accounts, groups and roles and assign specific permissions to each. This allows complete control over access to AWS resources based on the principle of least privileges, allowing tight security and access to only that which is necessary for an intended role.

IAM also provides other security measures to its users such as two-factor authentication and multi-factor authentication, password policies and encryption keys. This system is vital for AWS and its users, AWS users occasionally publish key information on platforms like GitHub allowing for malicious users to charge compute costs to the unknowing developer. To maximise the effectiveness of the AWS free tier IAM was used to keep keys and roles locked down to ensure no unsuspected compute costs.

2.5.3 Simple Storage Service

AWS Simple Storage Service or S3 as it is commonly known is a fully managed cloud storage service. It provides developers with an “infinitely scalable” highly adaptable and durable storage solution. AWS S3 allows a user to store and retrieve any amount of data, making it perfect for this project, as the goal of this project is to show the ability and benefits of having Citizen Science projects (which may have to store millions of data points) use voice interfaces. Another key factor in using this system is that it has an incredibly high durability of 99.999999999%[9] and availability of 99.99%[10] as quoted in the documentation.

S3 is an object-based storage service, which means that data is stored as objects in a flat address space, rather than as a hierarchical file system. Each object can be up to 5 terabytes in size and can be accessed via a unique URL. S3 also supports a range of APIs, allowing developers to integrate S3 with their applications, as well as with other AWS services easily and efficiently.

2.6 Understanding Voice Interfaces

With a better understanding of the problem and the solution needing to be developed after research conducted in steps 2.2-2.5.3 the next step is to build a deep understanding of good voice interface design and implementation. Not only will understanding how to build good voice interface features be important but also how voice interfaces themself work.

The key to understanding how voice interfaces work is their dependency on natural language processing or NLP[11]. Natural language processing is a branch of artificial intelligence that focuses on the interaction between computers and human language. NLP involves developing algorithms to understand, interpret and generate human language. The main algorithms used (with regards to NLP) by Amazon Alexa from the point of activation to response are; Automated Speech Recognition, Natural Language Understanding, Dialog Management and Text-to-Speech. Also included in this list are machine learning and neural network algorithms to improve accuracy and performance over time.

Far-field ASR is a critical technology whose goal is seamless voice communication between users and Amazon smart speakers[12]. These systems are used in every scenario that the device needs to interpret what a user is saying. At this point, the audio signal from the user is recorded and processed to remove any interference and background noise. The processed audio is then sent to a server that handles the NLU. The server then selects the skill necessary to perform the task requested by the user. At this point the server sends back this package for the device, such as an Alexa, to perform the TTS processes.

2.7 Standardised Development of Alexa Skills

To best understand the development of an Alexa skill application for this project, multiple research strategies were used. The first and most important was gaining an understanding of how to design general voice interfaces, through this step the book "Designing Voice User Interfaces: Principles of Conversational Experiences" by Cathy Pearl was referred to. This book gives good examples of guidelines. Chapter two of this book, "Basic Voice User Interface Design Principles"[13] was the most useful as it allowed for a general discussion on voice interfaces while also giving numerous examples and diagrams for many different voice interfaces including Alexa.

This book also talks about how to improve the design with regards to the conversational experience of a user, the main understanding is as much as possible include confirmations in responses to help user understanding. The book uses a good example of this on page 20 with a timer example [14], this example demonstrates the difference in effectiveness between the voice interface just confirming a timer has been set compared to confirming the timer had been set and the timer duration. These concepts would be carried forward into the build stage of this project.

This book also later talks about how to test voice user interfaces in chapter 6, "User Testing for Voice Interfaces"[15] which was referred to in the latter parts of the project once user testing was underway.

The second was rapid prototyping inside the Amazon Alexa Developer Console, which allowed learning and development of basic understanding at a fast pace while keeping engagement with the problem.

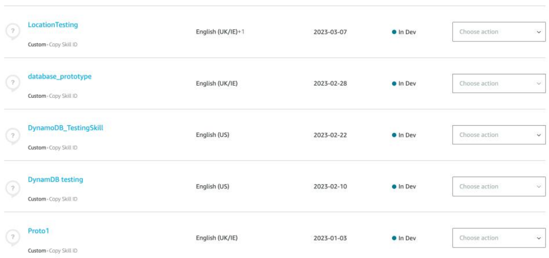


Figure 2: This image shows all prototype skills made before combining skills learned into one main skill. This image shows how the problem was broken down into 3 main aspects, getting the user's location(LocationTesting), adding data to the database(database_prototype, DynamoDB_Testingskill and DynamoDB testing) and general conversation testing (Proto1).

2.8 Accessibility in Voice Interfaces

Although not a primary goal of this project a key factor in any research and development stage is accessibility. There are many issues when working with voice interfaces and smart speakers to consider, the most common being the different accents and dialects of users. Alexa's ability to understand different accents and dialects stems from its use of phonemes to break down spoken commands. These are then compared to a database to identify the closest word matches. Additionally, the software is designed to recognise sentence structure and relevant terms for various subsystems. For example, if a user says “Alexa, kitchen lights one hundred per cent” Alexa can forward the command to a smart home API.

To support multiple languages and regional variations, Amazon has unique databases for each supported language, users sometimes need to select these language settings to get the best results from their Alexa. Machine learning also plays a key role in Alexa’s ability to understand context and history, which helps Alexa make more accurate interpretations of commands. To improve its accuracy, Amazon analyses recordings from real-world customers, since humans typically use context and history to understand the conversational meaning. Although Alexa still makes mistakes, the ability for users to report this alongside the massive amount of data being gathered means Alexa will only get better.

Another important consideration of developing a voice interface application is not all users have the ability to easily interact in this way. Alexa’s speech recognition technology is designed to handle users with speech impairments but only to a certain extent, although the level of success may vary from user to user Amazon itself is working to improve this alongside some third-party applications such as “Voiceitt”[16]. Another limitation is if a user has a limited vocabulary such as a child. Ultimately this will not affect the application being developed as it is targeting adults and will have only simple voice interactions. The research article “*Accessibility Came by Accident": Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities*”[17] provided incredibly useful information on the covered accessibility issue and was referenced through prototyping and development.

2.9 Data Visualisation

One of the extensions of this project was to allow users to see data being added to the citizen science project through the use of a public website. A key consideration in this extension was how to display the data efficiently while also making it accessible and exciting to users. To tackle this issue the book “Data Visualization a handbook for data driven design” by Andy Kirk[18]. In this book, there are hundreds of interesting ways shown to show data, but the one that influenced this project was using mapping, in particular, heat maps, proportional symbol maps and choropleth mapping. With these techniques, data would be able to be displayed in an interesting way while also being informative and possibly interactive.

Each of these map types has its own “presentation tips” alongside images to help design the best system possible.

3 SPECIFICATION

3.1 Creating Specification

With the given goal of creating a practical voice interface for citizen science projects, the first steps towards creating the specification were taken by examining the features of Alexa. This meant exploring and testing other Alexa skills and prototyping until a clear picture of the final product was formed. From this picture and user testing forms, requirements could be created

3.1.1 Use of Survey for Specification

A user survey was conducted to help produce the requirements and specifications. The goal of this survey was to understand the demand for a voice interface used for citizen science data gathering while also gauging the interest of potential users. This survey also allowed users to voice concerns and ask questions about certain aspects of the system such as its ability to pull and store a user's location.

The survey was sent to a number of participants to gain as wide a range of response data as possible. This ranged from other students, and software developers including developers of voice interfaces and researchers in the field of voice interfaces. As stated in section 2.2 the ideal participants would be 60+ as this is the most active range of users in citizen science projects. Unfortunately, the use of a user survey is the best way to gather data at this point and doesn't fit well with that demographic as many have limited technology literacy in this age demographic. This demographic would have been perfect for the use of a focus group but due to time limitations, this wasn't achievable. The full survey can be found in Appendix A.

There were a total of ten out of twelve responses to the questionnaire. At least one user from each targeted demographic responded to the questionnaire allowing for a broad range of responses and views. Feedback quality ranged from very basic to in-depth questions and long-format responses. The survey contained a total of 12 questions 10 of which were mandatory multiple choice allowing users to give quick and easy responses to minimise the time taken for a user to complete the form. The two long format questions were the final questions in the survey, both of which were deliberately kept vague so that users were more likely to fill this in as it was optional.

The first survey question gave a brief description of citizen science and then asked the participants if they had heard of citizen science before. This was asked as it would mean that responses could be split easily into categories of a user's understanding of the project and its goals. The answers to this question were 70% hadn't heard of citizen science before this point and 30% had, 0% maybe. Although this is a low percentage, the users who hadn't heard of citizen science before would still be able to give crucial feedback on important questions later in the survey.

Question three of the user questionnaire was used to gather data on what voice assistants were being used by the survey participants. Shown below in Figure 3 are the responses to this question. In this question, participants were able to select multiple answers as it is likely that someone uses more than one voice assistant application in today's information age. A surprising factor of this question is that all users who completed the survey use at least one voice assistant which shows the growth these systems have had. The most common answers to this question were Amazon's Alexa and Siri, this may be due to the fact that there is only one option for Google Assistant when this could be used over multiple devices compared to Siri and Alexa having much more restrictions on the number of devices they operate on.

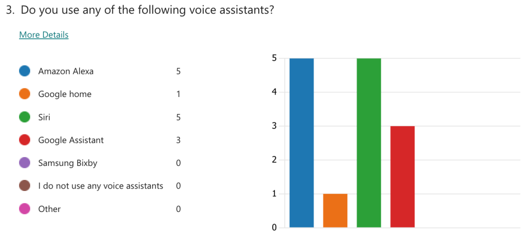


Figure 3: answers to question three of the user survey. This question was used to find what voice assistants were being used. The most common being Alexa and Siri with five votes each.

Question four of the user survey asked "Would you use your (if applicable) voice assistant to submit data to a citizen science project?". This question was designed to get an understanding of how users interact with their voice assistants. The question could have been a general question about how they use their system but this wouldn't have been much use to requirement gathering and specification, so the question asked if the user would use their device to submit citizen science data. The flaw in this question comes from question one, we know that only 30% of users taking this survey know what citizen science is. The results show this as 50% of the respondents answered "Maybe", which most likely means they don't fully understand what the question is asking. This was learnt from and taken into account later in testing.

Another important question asked in this survey was question six, this question asks "Do you believe that users are more likely to participate regularly if they can see data being gathered?". This question is important while considering the requirements of the project as it was used to show why the development of the mapping system was important. The responses to this question were 8 yes, 1 no, and 1 maybe. This result clearly shows that the average user is more likely to submit data consistently if they can connect with the data and see it change over time. This did raise further questions such as, how often should the data be refreshed to gain the perfect balance of interactivity from the public and not using high levels of AWS computing power to run, for example, a live view.

There were however some issues with this user survey, the main issue was it was vague. This was partially deliberate as it gave good answers and made the survey quick to complete. It did however limit the feedback received from the open-ended questions. For example the answer:
“I can see it's a map, but I'm not sure what is actually being visualised! What does the green area represent? Does the fact that it's green mean anything? Is it an interactive visualisation that I can use to hover over other areas of the UK?”

These questions are valid as the image used was a very early prototype of the mapping system with little contextual information, the image shown in Figure 4. To tackle these questions in later rounds of testing it was made as clear as possible what a user was looking at and why it was important to the question.

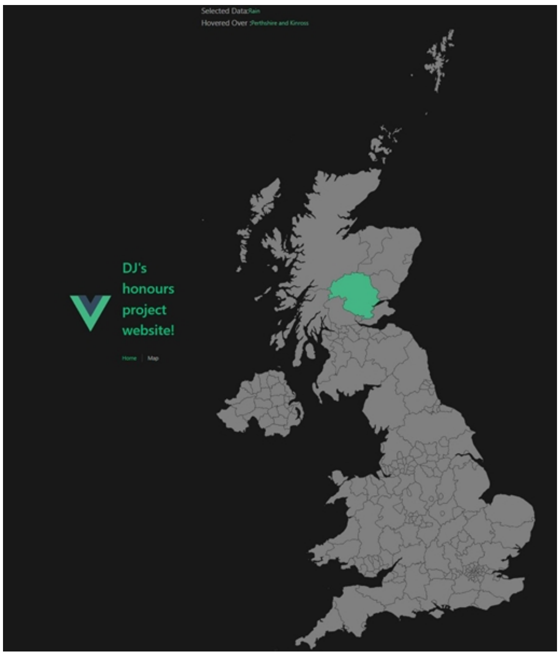


Figure 4: this is the image used in section 4 of the user survey “Data Visualization”. This image shows the UK split into counties with Perth and Kinross county selected in green.

In the final question, users were asked if there was another way they could think of displaying the data instead of a mapping system. This question was asked as the map was an extension of the project brief it would likely take up development time. Although most users didn’t answer this question there was a suggestion of using a bar chart. This would have been a good option but as the user's data is being tied to a postcode area with there being 124 different options this means that there would have to be far too many bars on the chart to show in a useful manner. The other option would be to change the granularity of the data, for example just use Scotland, England, Wales and Northern Ireland, but this would mean that the data would generally be pointless as these areas are far too big.

Overall the user survey was instrumental in forming the requirement specification. It allowed insight into areas someone who understands the topic would overlook. Connecting with possible end users through the process of the development stage was incredibly important when developing something that not only uses the general public for data gathering but makes that data publicly visible. Full responses and questions from this survey can be found in Appendix B.

3.2 Requirements

Taking the information gathered from the user survey discussed above and research done up until this point we are able to construct the requirements for the Alexa for a Citizen Science project. The produced functional requirements can be seen in Table 1 and the non-functional requirements in Table 2. These requirements will be used to produce user stories which are referenced throughout the development of the project. These requirements can also be used to understand how the goals of the project change over time as inevitable issues arise.

Table 1: This table shows the Functional Requirements for this project.

	Functional Requirements
ID	Requirement Title

1	The Alexa skill must be activated by a clear invocation name that relates to its function
2	The Alexa skill must sound natural in its questions and answers
3	The Alexa skill must have clear conversational answers confirming as many variables as possible to the user
4	The Alexa skill must follow a standard route to add data
5	The Alexa skill must be excitable at any time
6	The Alexa skill must have a help option that can be accessed at anytime
7	The Alexa skill must provide (as much as available) accurate information to the database
8	The Skill must link to an AWS database
9	The AWS database should be able to be accessed by an authorised user
10	The AWS database should be able to send the data to an S3 bucket
11	The data added to the database should only be what's needed for the skill to perform its task
12	The Alexa skill must have secure authentication and data handling mechanisms to protect user data and privacy.
13	The Alexa skill must not access irrelevant information for its task
14	The Alexa skill must be authorised to access a user's location services within the Alexa settings
15	The website must have a visualisation of the data
16	The Alexa skill must tell a user how to activate location settings if not active
17	The website should be hosted publicly
18	The website must only use the minimum user information

Table 2: This table shows the non-functional requirements for this project.

	Non-Functional Requirements
ID	Requirement Title
1	The Alexa skill must be as accessible as possible
2	The Alexa skill adding to the database must be adequately fast
3	The Website must be assessable
4	The website must be adequately fast

3.3 User Stories

After constructing the project requirements the author decided to turn these into user stories for later use on kanban boards. This allowed the author to use systems to easily visualise what development features were in development, testing, or completed. These user stories can be found in full in Appendix J.

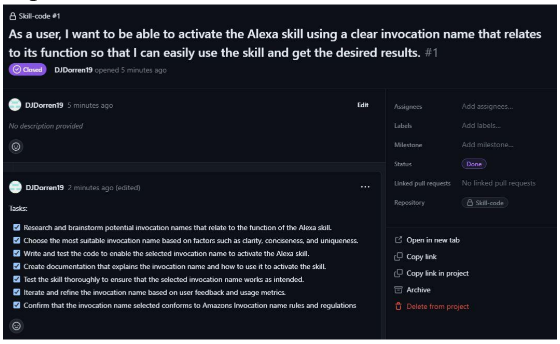
To create the user stories the author completed a number of tasks. The steps followed are outlined below. User Story 1 also shows how requirement 1 was changed to a user story with its respective tasks.

The first step was defining an acceptance criteria or a definition of done. This would allow the author to define when a development task has been fully tested and can be moved to complete.

The second step was creating user personas, although this is an important step it was mostly negated by the fact that this product would not be going to market. This step could have been undertaken but as the skill and website would be designed to be as accessible to all users as possible the author decided to define the persona as “an Alexa user”.

Step three was to create an individual task, the author broke down each requirement into what at the time they thought the task would be. This allowed the author to tackle each individual requirement in a more granular thought out manner decreasing the likelihood of errors and wasted development time.

The fourth and final step was to weigh the stories in their importance and complexity. This would allow the author to assign an expected time of development attribute to each individual task. This would be used to divide tasks into different sprints to allow for an even distribution but as the author had decided to develop in a waterfall method this wasn't as important but the task was still undertaken.



User story 1: This image shows the full completed user story for requirement 1. It shows a clear user story stating "As a user, I want to be able to activate the Alexa skill using a clear invocation name that relates to its function so that I can easily use the skill and get the desired results." as well as the breakdown of tasks that have all been completed.

4. PROJECT MANAGEMENT

During any large-scale development project, it is important to pick a development strategy. The choices for this project were between Agile and Waterfall. Regardless of the development methodology picked as this project would be undertaken by a single developer, requirements and the project as a whole must be clearly defined and understood.

4.1 Github

GitHub is an online platform that offers version control and collaboration features for developers. It has become the industry standard version control platform and is an essential tool to allow developers to manage and share their code with team members. The platform has many advantages that can be taken advantage of in this project, such as integration with both AWS platforms and visual studio code source control.

Many of Github's features are targeted at development teams although these can be used by a single developer. Examples of this include pull requests which in a team situation allow other developers to review code before committing to a branch safeguarding the code from as much human error as possible.

In this project, Github was used to safely store code in the cloud to minimise the risk of losing code through computer damage. Both the Alexa skill code and website code were saved to GitHub. Although the Alexa skill did not require GitHub integration it was decided that this would be a good fail-safe in case there was ever an issue with the Amazon system.

The website code was developed in VueJs using its system for running local servers to allow for rapid development, testing and prototyping. This meant that code from the website often was pushed to the repository in large sections and not at each addition or change.

4.2.1 Project Management Tools

Using GitHub as a code repository and version control system is not its only ability, GitHub offers a wide range of project management tools such as Kanban boards. Other tools such as Trello which may have more features and integrations were explored but it was decided as this project was being undertaken by a single developer GitHub's features would be sufficient.

The main project management tool used for this project was Kanban boards[20] and Gantt charts. Kanban boards allow for an increased view of the overall project and its tasks. This allowed for a better understanding of how the development of the project was going and more importantly, as this was a single development team, making sure nothing was missed or forgotten about.

The user stories were all put into the project backlog and as development progressed each user story would move from the backlog into

"Under-development" then "testing" and finally "complete". This allowed for a good, easy-to-read visual representation of how the project was progressing over the development term.

The only consideration to be made when using Kanban boards is they can take quite a bit of maintenance and as this was a single-member project, items were often put into complete that ended up having to be moved back as code review and testing were not done by a different person which gave bias.

4.3 Gantt Chart

A Gantt chart was produced at the early stages of the project, this was created using a tool called TeamGantt[21] which is an online project management tool. This Gantt chart was updated through the project timeline as many of the development tasks took longer than expected. The original Gantt chart is shown below in Figure 5.

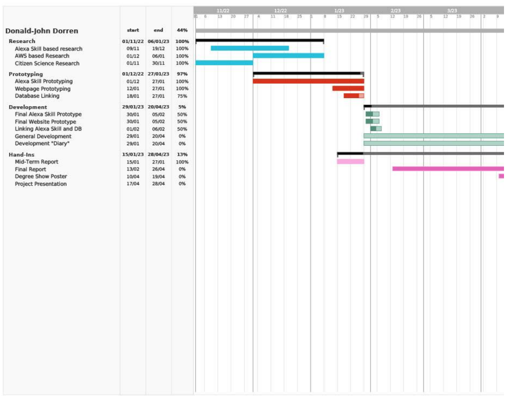


Figure 5: This image shows the initial planned work schedule in a basic form. This Gantt chart is split into 4 timelines, the blue is preliminary research, red is prototyping, green is development and pink is hand-ins.

4.4 Initial Work Schedule

The initial work schedule was very sparse and lacked detail, this improved with the development of the project as the Gantt chart was updated and tasks were understood to a greater degree.

The initial plan was split into four main sections each with their own subtasks. Although these sections were vague they were split effectively and the general structure was good. For example, it was extremely useful to include the hand-ins section in this plan as it may be an often overlooked part of the project as a whole. The hand-in section was generally overcommitted on time, for example, the Degree show poster is said to take nine days. Even though other work was planned to be completed at the same time as these nine days were still too much to allocate.

Another flaw in this Gantt chart is the vagueness of the development stage, even though requirements and user stories had been made by this time the development stage still included tasks such as "General Development".

4.5 Amendments to Work Schedule

Almost every task from the initial work schedule was changed in some way, some tasks had their length shortened due to fast development and some had their length greatly increased due to issues found in testing. Overall the amendments were applied once the fix had been applied as soon as the task was completed. As mentioned earlier, some tasks did have to move back into development and testing from completion, in this case, the Gantt chart wasn't updated as the fixes were usually quick and took a couple of hours of development time to complete. The final schedule can be found in Appendix C.

4.6 Diary

Throughout development, a diary was created to allow for easy backtracking and referencing of dates when writing the final report. This diary was also used as a reference during the weekly advisor meetings. In this diary tasks for the week ahead were written down on Monday, and then throughout the week issues were logged. At the end of the week, the tasks were split into completed and not completed and issues that were resolved were marked as fixed. This diary was just a simple text document and with hindsight, better software could have been easily implemented for this function.

4.7 Midterm Report

At around the halfway point of the project, a mid-term report was created and submitted. This report provided an insight into how the project was progressing and allowed an overview of features such as the plan. This report was extremely useful in terms of reflection on the project. The mid-term report also provided feedback on any scoping issues and allowed for practice at systems such as referencing. The full mid-term report can be found in Appendix D.

5. DESIGN

To design a functional voice interface system it was necessary to understand what the final system would look like, and how this would solve the problem stated in the project outline. To design a voice interface system it is important to understand the differences between a voice interface and a standard interface such as a phone or computer. Apart from the obvious input method, how does a user navigate the application easily with no visual aid? How does a user get easy-to-understand and accurate feedback on incorrect inputs?

5.1 Design Considerations

As covered in the previous research section there were considerations to make before starting the development of the voice interface system. Using the knowledge gained from the research into voice interfaces using resources such as "Designing Voice User Interfaces: Principles of Conversational Experiences" by C. Pearl [13], tasks such as user flows were completed which can be seen in Figure 6, and with notes in Appendix E. This user flow gave the developer a better understanding of how a user might interact with the system. The flows were first sketched on paper and then designed in Figma[22].

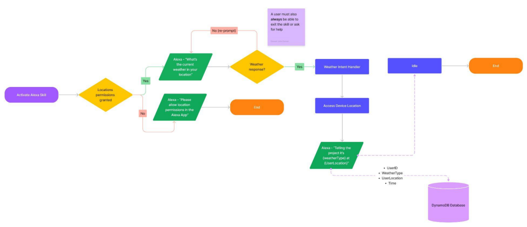


Figure 6: This image shows a basic interaction between a user and the Alexa for Citizen Science Alexa skill. This flow takes into account all user interactions and possible choices. Not displayed is the option to exit or ask for help at any point in the flow, this is as it confused the diagram so was substituted for a note at the top of the diagram. This image can also be found in Appendix E with its full annotations and notes.

Creating this visualisation allowed for a greater understanding of user interaction with the Alexa system. This diagram was often referenced throughout development, but as this was made early in the design and development process it adapted over time. While building the user flow system all learning from section 2, Background was implemented, for example making sure the voice interface system confirmed as many of the details as possible to the user, and making sure the help and exit functions were always able to be activated.

5.2 Selecting a Development Platform

When building a custom Alexa skill there are multiple choices available to a developer. From choosing the invocation name to the coding language to be used. Amazon's Alexa Skill Developer Console[23] and its built-in web IDE helps guide a developer through these choices. During the process of creating a new Alexa skill, the developer is presented with several choices, each with their benefits and skeletonised starting code that is easy to build upon. These options range from selecting an experience such as a movie, music or smart home. This approach to development is very user-friendly and allows for a great understanding of the development environment in the early stages of development.

Once the developer has a better understanding of the development environment and platform they are able to start the development of their custom skill completely from scratch. A developer can also choose from a number of different programming languages that are supported by Alexa. The main languages are Node.js and Python, and both of these languages have their own documentation and examples. Both languages were used in the development of this project, most of the prototyping was done in Python but later in the development process development changed to Node.js. This decision was taken as the author had more experience with this language.

5.3 Rapid Prototyping

In the early stages of development, a rapid prototyping approach was taken allowing the development of multiple small projects that could be learnt from and eventually combined into the main project. This was an effective way to get hands-on experience testing, coding and debugging inside the Alexa Development Console.

After research was conducted the first prototype was produced. This basic skill allowed a user to simply tell Alexa what the weather was, at which point this would be stored in a variable and then confirmed to the user by Alexa. This was a simple but essential project to undertake as it allowed an understanding of simple aspects inside the development environment such as the interaction model. Intents are an aspect of the interaction model much like a function. Intents are activated by a user saying a certain phrase called utterances[32]. These utterances can be anything for example in this project to tell the skill it's raining a user would have to use either a word or phrase associated with rain. At that point, the intent and its code will be activated.

As soon as enough was learned from the first prototype development started on the second. This fast-paced development allowed quick learning through failures that didn't have a large impact on the project as a whole. The second prototype made use of slots[24]. A slot refers to a placeholder for a specific type of data that a user can provide as an input to an Alexa skill. Slots are used to extract information from the user's spoken utterances which are then used by the skill as explained above. In prototype two the skill didn't have an individual intent for each weather type unlike prototype one. In this skill, there was just one intent "WeatherIntentt" which tried to extract the weather type from the user's speech. This approach massively reduced code size and made the overall skill more efficient.

The most important prototype developed was the location prototype. This was a key turning point in development as everything up until this point had been simple, this was more complex. The goal of this prototype was to pull the location from either the physical device. This prototype was harder than initially planned for two reasons. Accessing the device's location in a skill is something that at a base level skill can not do, it needs special privileges set both inside the development console for testing and inside the Alexa App[25] for testing on a physical device. The second issue within the development of this project was it wasn't made clear when the custom skill was being activated as there is already a built-in skill to the Alexa that when asked what the user's location is will return the device's physical location. This meant that for weeks it was perceived as if this prototype was working when in fact the custom skill wasn't even being activated and an inbuilt skill was running.

Finally, this issue was fixed allowing for development to start on a skill that sent user data to a DynamoDB database. This skill was created quickly and inefficiently because of deviations from the schedule with the location prototype. Although this skill was made quickly it gave great insight into the ability of databases with Alexa. For this project, the integrated services were used which allowed for some customisation but not to the level needed for the final project.

5.4 Technical Implementation Design Decisions

Throughout this project there were many changes to the scope and original designs made, some improved the project some ended up making more issues. Most changes came when integrating databases into the skills. Originally the author had planned on just using an integrated DynamoDB table such as that used in the database prototyping. This would have not been the correct way to do this but was simple and easy. The problem was that the Alexa skills integrated systems are heavily restricted in their abilities, for example, they can not export data to an S3 bucket which was needed to integrate the skill and the website. This meant that for the final product, a much larger number of AWS systems would be involved, meaning a better understanding of AWS systems would be needed.

Another important technical decision would be that of data capturing. Ideally, the skill would be able to capture the geographical location of the user using latitude and longitude, but this is not possible (on the date 09/04/2023) on a physical device. As getting the location of a user was a requirement for this project the next best option would be pulling the address linked to an Amazon Alexa account. There are issues with this that will be discussed later in this report. To solve this problem it was decided that the device would capture the full address so that this can be confirmed to the user but only the postcode area[26] section would be saved.

Deciding to only store the postcode of the customer location had a knock-on effect. This meant that for the website mapping system, a third-party tool such as MapBox[27] wouldn't be able to be used as these systems generally require latitude and longitude to be effective and translating an address to latitude and longitude would be possible it would just take integration of another system such as Google's Geocoding API[28], which would increase the cost of development. At this point, it was decided that the mapping system would use an SVG map created from a GeoJson.

5.5 Figma

Figma[29] was used to create graphical designs used not only for development but also for the mid-term and final report. Figma is a cloud-based design tool used for creating user interfaces, prototypes, and design systems. It was first introduced in 2015, and has since become an industry standard choice among designers and design teams due to its collaborative features and ease of use. Although one of Figma's main strengths is collaboration it was still picked for this individual project as the author had previous experience using this software

6. IMPLEMENTATION AND TESTING

To best cover all aspects of implementation this section will be split into, The Alexa Skill, DynamoDB and AWS and finally data visualisation and web development. The implementation stage will cover all aspects of the developed project starting with activating the Alexa device to seeing the displayed data.

6.1 The Alexa Skill

As described in section 5.3, this project made use of rapid prototyping. This approach allowed the author to build knowledge fast and effectively through trial and error. The implementation stage will mainly cover the final product but will reference prototypes.

6.1.2 Invocation

One of the most important choices a voice interface developer can make is deciding how a user will activate the system. Alexa has the standard wake word of "Alexa" which is then followed by the invocation name. Amazon supplies developers with a list of Invocation Name Requirements[30] and rules for invocation names which help guide a developer through picking an appropriate invocation name.

For this project, the Invocation name "Citizen Science" was used as this is descriptive of the skill while also not infringing on any copyright. There are three main ways to invoke a custom Alexa skill all of which can be seen in the flows below in Figure 7. Invoking the skill with a request i.e. "Alexa, add rain to Citizen Science" This way of activating the skill in one step activates the Alexa, activates the custom citizen science skill and supplies it with the slot "Rain". This is a quick and effective way of using the Alexa voice interface and would most likely be used by an experienced voice interface user.

The second way to activate the custom Alexa skill is by invoking the skill without a particular request. This way of activating the skill will usually use a phrase such as open or start. An example of this could be "Alexa, start citizen science", in this instance, the Alexa is woken and then immediately supplied with the skill name. In this flow the skill still requires the weather type so must ask for that before anything else. The final invocation flow is when a user just activates the skill using the invocation name such as "Alexa, citizen science" This case is much like the previous for this project so the flow was the same.

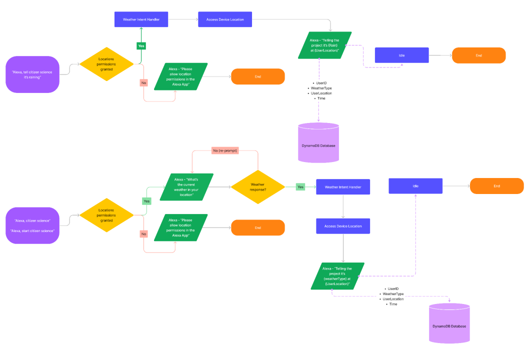


Figure 7: This image shows the different flows depending on how the skill is invoked. In the

top flow, the user wakes the Alexa device, invokes the skill and adds the rain weather type all in one command. In the second flow, the user only wakes the Alexa device and invokes the skill meaning that the custom skill still needs to capture the weather type.

The invocation name citizen science was chosen as it abides by all the requirements of an invocation laid out by the developer documentation[30]. Although this skill will not be deployed to the general public, efforts were still made to abide by Amazon's ethical and legal rules. These rules state that the invocation name must not infringe on the intellectual property rights of an individual or entity. One-word invocation names are not allowed. Invocation names can not include names of people or places. Invocation names must not include any of the Alexa launch phrases such as start or run. The last rule is that invocation names can not be two words but this only applies when one of the words is an Article[31] such as, "the" or "a".

The chosen invocation, named "citizen science", applies all these rules and regulations at the time of writing 10/04/2023.

6.1.3 Intents

Intents are another fundamental building block of the interaction model. An intent represents an action that fulfils a user's spoken request. An intent works like a function of a standard development environment, but it is called when a user says what is called an Utterance[32]. Utterances are defined by the developer when creating a custom intent, they define when this intent should be activated and can be any phrase. For example, the RainIntent inside this project could be called by saying any of the phrases seen in Figure 8. A developer could write hundreds of utterances for this intent as they're hundreds of ways of saying it's raining. When picking which phrases to use and when to stop the documentation by Amazon, "Best Practices for Sample Utterances and Custom Slot Type Values"[33] was referenced frequently.

```
{
    "name": "rainIntent",
    "slots": [],
    "samples": [
        "it is drizzling",
        "drizzle",
        "it's drizzling",
        "rain",
        "it is raining",
        "it's raining"
    ]
},
```

Figure 8: This image shows the utterances used in the early stage of the final product. As shown there are lots of different ways a user can tell the skill it's raining and hence there are multiple options for how this intent is triggered.

6.1.4 Slots

Slots while being one of the most important features in the voice interface system were the last to be understood and implemented into this project. Slots work like variables storing information that would otherwise be repeated. Using slots can greatly improve the readability and scalability of the custom skill. Combining Intents, Utterances and slots creates the interaction model of the skill.

6.1.5 Permissions

A vital part of this project is knowing where the citizen scientist is submitting data from, without this the data would serve almost no purpose. To get the data gatherer's location the custom-built Alexa skill must access the device address linked to the physical Alexa device. All Amazon Echo products have this capability. Within the skill editor, there is a section called "Permissions" that allow features that would otherwise be unavailable without customer consent. Most Alexa skills will use at least one of these features such as gathering the customer's name, email address and phone number, but all need to be approved within the Alexa app.

Like many systems that access user data, there are regulations on how this information can be accessed and used. To be certain this project abided by these rules the documentation "Configure Permissions for Customer Information in Your Skill"[34] was followed. This documentation states that only when the user grants permission can the skill be executed, this is shown in the user flows in Figure 7.

6.1.6 Location

When gathering the location data from the citizen scientist there were two choices, using the customer's full address or just country and postal code. When supplying the database with the location data only the postcode is used, so the less intrusive option would have been preferred. This would have been ideal but unfortunately, this didn't work on any device while testing. This meant that the full address option was used, but only to respond to the user with their whole address, while only the postcode was stored meaning that all guidelines laid down by the legal documentation were abided by.

Another issue with the location system was its reliability. When setting up an Alexa device for the first time the user is prompted to sign into the device application using the same email address associated with their Amazon.com account. This gives the ability to link the physical device with a single user and location. The user is also asked to input the address where the device will be located in case this is different to the Amazon.com account. From this using the location system discussed above we are able to get the address linked to the physical device. Unfortunately, as this location is manually inputted by a user it can not be accurately relied upon. When testing it was even discovered that the author's location had been incorrectly entered when setting up an Alexa device meaning that the location system would return null instead of the street address.

6.1.7 Database

Once the weather type and location have been gathered it's added to the database along with some extra information to make each addition unique and identifiable. The Alexa skill sends four items, the id which uses the Date.now().toString() functions to make a unique key, the weather data, postcode, and the sliced postcode. The sliced postcode is the postcode area. This is calculated by taking the first two characters from the postcode, in most cases this will work to calculate the postcode area, but there are some exceptions. In some areas of the UK where the postcode area is just one character, for example, north London is just defined by N. To handle these exceptions the second character of the sliced postcode is checked using isNan() function. If this returns true then we know that the sliced postcode contains a number meaning that the area code for this postcode is a single letter. If this is the case then the sliced postcode is sliced again down to a single character. Shown below in code snippet 1 is an example code snippet of how this operation was tested before being implemented in the skill code.

Code Snippet 1:

```
//let slicedPostCode = "N1"; // Example slicedPostCode string

// Check if the second character is a number
if (slicedPostCode.length === 2 && !isNaN(slicedPostCode[1])) {
	// Slice to just the first character
	slicedPostCode = slicedPostCode[0];
}
```

At this point, Alexa has all the user information necessary. This information is sent to the DynamoDB database.

6.1.8 Publishing

The final stage of developing an Alexa skill would be publishing, allowing any Alexa user to use and download the skill onto their device. The developed skill for this project meets all requirements to be published. The author decided that there was no need to publish this skill to the general public as the testing could be done by using the author's Alexa. Publishing skills can also take up to a week to be verified of meeting all requirements. During this verification process, no changes can be made to the skill which would have been a large risk to take late into the implementation stage.

6.2 Amazon Web Services

Section two of this project is implementing the Amazon web services. There are five different services used in this project. These services will each have their own sections and describe their use cases in general and their use in this project.

6.2.2 IAM

Identity and Access Management is a web service provided by AWS that allows users to create, manage, and control access to AWS resources. It offers features such as fine-grained access control, integration with AWS services, and auditing and logging capabilities. This makes it essential for securing AWS environments. IAM is important to all AWS integrated systems but it is essential for beginner AWS users as they may be more likely to have unsecured base systems.

IAM was integrated into this project to allow the Alexa skill to send data to a DynamoDB database that wasn't the integrated and restricted system inside the Alexa Developer Console. This was necessary as it would allow the data to be sent to an S3 bucket in turn providing the data to the website.

To set up the project's Alexa skill with IAM, the author created an IAM role that had all the necessary permissions to access the DynamoDB table. To follow the standard practice of the principle of least privilege[35] the IAM role was given minimal access to the database.

6.2.3 DynamoDB

The implementation of DynamoDB into the alexa skill system was thought to be a simple task. As the project developed and became more complex it required more and more permissions and storage. The author discovered that the prototype DynamoDB skill that had been used up until this point was causing a number of issues. As discussed earlier the integrated DynamoDb database is heavily restricted. To fix this issue and allow the skill to send information to a full database system the author had to change how the skill was interacting with the database. To do this the author implemented dbhelper.js as well as creating new IAM permissions to allow the skill to interact and add data to this new database called “Honours Project". The code for dbhelper.js can be seen below in code snippet 2.

Code Snippet 2

```
var AWS = require("aws-sdk");
AWS.config.update({region: "eu-west-1"});
const tableName = "Honours Project";
var dbHelper = function () { };
var docClient = new AWS.DynamoDB.DocumentClient();

dbHelper.prototype.addWeather = (weather, userID) => {
  return new Promise((resolve, reject) => {
    const params = {
      TableName: tableName,
      Item: {
        'weatherType' : weather,
        'userId': userID
      }
    };
    docClient.put(params, (err, data) => {
      if (err) {
          console.log("Unable to insert =>", JSON.stringify(err))
          return reject("Unable to insert");
      }
          console.log("Saved Data, ", JSON.stringify(data));
      resolve(data);
    });
  });
}
module.exports = new dbHelper();
```

There were a number of issues while implementing this new system, the issue that caused the most loss of time was that the author hadn’t set up the permissions correctly. This meant that when the skill was trying to add the weather report it would just return an error response. Unfortunately the error codes and responses associated with developing Alexa systems are incredibly vague and hard to fix and so take extra time to locate and fix.

After an extended period of being unable to fix this issue the author decided to give the skill full access to the table. Although this fixed the issue the author was having, it is not the correct way this issue should be resolved. The author had planned on coming back to implement a proper fix for this but no matter what was changed this was the only way the author could fix the issue.

6.2.4 S3

S3 implementation was simple. There were very few errors associated with the implementation of this AWS system. The author managed to integrate this system with dynamoDB very easily with no code needed. All that needed to be done was linking the export destination of the database information to the S3 bucket name.

6.2.5 Cloudwatch

Cloudwatch was used a number of times throughout the implementation and testing of this project to allow for monitoring of all systems being used throughout development. This system allowed the developer to monitor the system as a whole using alarms and logs. A

6.3 Website Development

The website was the last feature added to the project. This was because the author has the most experience in web development and data visualisation. It was decided very early on in the implementation stage that the website should use Vue.js, and the data should be visualised in the form of a choropleth map.

6.3.2 Vue.js

Vue.js[36], or Vue, is a popular open-source JavaScript framework used for building web applications. Developed by Evan You and first released in 2015. Vue.js has gained significant popularity among web developers for its flexibility, simplicity, and performance.

Vue follows a Model-View-ViewModel[37] architectural pattern and provides a set of tools and features for building reactive and dynamic user interfaces. It allows developers to create reusable components, which can be composed to build complex user interfaces. Vue also allows developers to use computed properties that are automatically updated based on dependent data, and watchers that trigger when specific data changes.

When creating the Vue application for this project the Vue.js quickstart documentation[38] was used allowing for an easy and simple base application that could run a locally hosted testing environment. This locally hosted environment allowed for rapid development and testing of all aspects of the website.

6.3.3 Packages and Dependencies

Many Packages were installed to streamline the website development process. Most of these systems were installed using NPM or Node Package Manager[39] which is a standard package manager used for JavaScript. The most important will be talked about below.

6.3.4 D3

Data-Driven Documents or D3[40] is a popular JavaScript library for creating dynamic data visualisations in web browsers. Some key features of D3 are; Data binding, allowing developers to bind data to DOM elements. Charting, although D3 doesn't include pre-built charts it has all the necessary building blocks for a developer to build custom chart data visualisations. Most importantly D3 has SVG-based graphics allowing for scalable high-quality graphics to be built into the browser.

The SVG system D3 uses allows a developer to build maps by using a topojson or geojson file as input. The author chose this system as they were already somewhat familiar with it. This allowed the author to quickly create the system used in the final product with little to no issues.

In early stages of development the author had used one map to display all reported weather instances on one map. This was an inefficient way to display this data so the author decided to split the reports into their own individual weathers allowing for a much clearer visualisation to be made. As this was done after the original map had been created the author simply duplicated the map page and added the Vue.js map component to each new page. This allowed the author to individually bind the report data to their respective map and also change the visual appearance of the maps depending on what weather type it was by changing the colour.

6.3.6 Three.js

As a data visualisation tool the author explored three.js as well as D3. The author developed a three dimensional spinning globe that allowed users to see information displayed globally. While the goal was not to display the data globally the author believed this would be a good way to explore the package and it's abilities. The author, with hindsight, understands that this wasn't an efficient use of time, but it did allow for more options to be explored when developing a solution. The globe system that was developed can be seen on the world tab of the web page. The author left this on the page as when the second round of testing was being conducted they asked participants what they thought could be good uses for this system so that the time spent wasn't completely wasted. The testers gave a number of useful suggestions but in the end the system was never used.

6.3.7 Vue Router

6.4 Testing During Development

As part of the online Alexa Skills Development Console, a developer can deploy and test their application as often as necessary. The test area uses a text interface that mimics interacting with the Alexa skill through voice commands which can also take voice input through the use of a text-to-speech button.

At the side of this Alexa simulator, the developer is able to see a JSON input and output after each request. This is a great tool for debugging as it allows the user to see more of what is happening in the background of the running skill. Some Alexa devices such as the Alexa Hub have inbuilt screens that can display images and text relevant to the running skill. This is shown in a visual window in the testing suit. This was used rarely in development as the author decided that it would be an extension to add visual responses to the skill. An example of the test environment is shown below in Figure 9. This testing environment was the main tool used for debugging and testing the Alexa application throughout development.

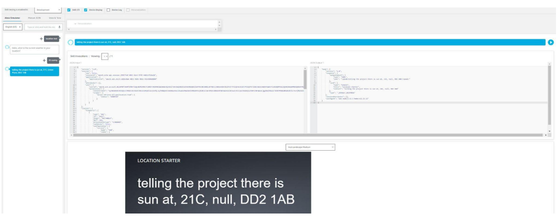


Figure 9: This image shows the testing environment built into the Alexa Developer Console. On the left, there is the Alexa simulator that allows a developer to interact with the skill as if they were a user. On the top right, both JSON input and output are shown allowing the developer to see into the backend of the request. Finally at the bottom of the screen is the visual window that mimics what would be seen on an Alexa device with a screen.

There were however some issues with the testing in this way. Every time the application's code or interaction model was changed it would need to be saved and re-deployed which took 3-4 minutes. This along with the substandard integrated code environment meant that a huge amount of time was wasted due to simple errors that would normally be highlighted within an IDE such as Visual Studio. These simple errors were also hard to locate in the code as most errors would just result in a standard error response within the Alexa simulator. These problems are part of the reason for the slow development of the custom skill.

Testing was also conducted on a physical Alexa device throughout development. Testing on a physical Alexa allowed for better testing of the conversation model and general interaction with the skill. This style of testing was most effective at highlighting when responses were too complicated or long, as when using the Alexa simulator the developer would often read the responses meaning that being overloaded with information wasn't a problem.

7. EVALUATION

7.1 Determining the Evaluation Method

The goal of this project was "To develop a voice-enabled interface using Amazon's Alexa for citizen science projects allowing users to add data to an AWS database for later analysis." An important part of this project is to evaluate the developed system against the original aims of the project brief. To do this a number of evaluation systems were used in both qualitative and quantitative methods. The author decided to get the best feedback from users, both these methods would be implemented.

7.2 Quantitative Evaluation

The first was quantitative, allowing for a large number of participants to give broad feedback which would later be displayed and analysed in graphical systems. This method was conducted early into the evaluation process using Microsoft Forms, which not only allowed for the evaluation of the requirements and aims but also allowed improvements based on user feedback to be implemented before the deadline. This system allowed the author to build and adapt the specification of the project as shown in section 3.1.1.

7.2.1 Recruiting Participants for User Survey

As quantitative evaluation methods depend on gathering a larger number of participants it was decided it would be best to recruit a wide range of demographics. As stated in section 3.1.1 of this report, the most common demographic for citizen science participants are those 60+ years of age. This would be a very hard demographic to recruit into the survey because many people of this age have a more limited understanding of voice interfaces and technology in general.

The most common demographic of participants used in this survey were other students of a varied number of degrees. This was because they were the easiest to recruit and the most available to the author. However, there were a number of other participants recruited who provided a different level of feedback. This group included several developers who the author had previously worked with and a member of staff at the university who has research experience in this field. This group who we will define as “professionals” gave far better feedback than others but were unfortunately unavailable for the second round of evaluation.

In total 11 responses to the user testing form were collected. This was below the goal of 15 but above the minimum number of 10 so rather than waiting for users, the testing was closed after a period of a week.

7.2.2 Quantitative Method

The method of user testing and evaluation using online forms is quick and easy but does require consideration of several areas. First, the issue of user anonymity was considered. It would have been ideal to be able to identify each user's response so that the best testers could be picked for the second round of testing but this wasn’t possible as it would breach the ethics and consent forms. To combat this issue all responses were anonymous. The University also states that the use of Microsoft Forms meets all ethical requirements for storing user data.
Participants of the survey were required to complete the Informed Consent Form which can be found in Appendix F and read the Participant Information Sheet Appendix G before completing the survey. This was insured by only sending the link to the survey once these were completed and verified by the author. As a precautionary measure, the first question of the survey was “I have read the participant information sheet or it has been read to me…”. This question was mandatory to answer yes to and would not let a participant see or complete any other question until this had been completed.

The user survey which can be found with its questions and responses in Appendix A was designed to be quick and easy for a user to complete while also providing quality feedback. The survey consisted of 12 total questions with 10 being multiple choice and the final two being open-ended questions allowing for optional feedback. The survey took an average of six minutes and eleven seconds to complete.

One key consideration that was taken into consideration was that users might not understand what citizen science was, to combat this a brief description was provided for them so that they could carry that information through the survey to inform their answers to later questions.

7.2.3 Results

The users were given several images from early development to help inform their feedback on areas such as the data visualisation. Question 10 asked, “Please rate the following attributes of the map system.”. This question allowed users to rate many aspects of the data visualisation from very poor to very good. This question used a five-point scale that gave a variety of responses. It was decided that all aspects of the map should score a minimum of 70% good or very good to pass. If an attribute fell below the 70% mark it would be considered a top priority to improve for future testing. all of the results can be seen below in Figure 10.

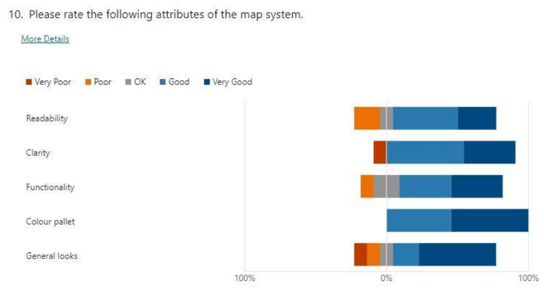


Figure 10: This image shows the responses gathered from question 10 of the user survey. The users were asked to rate Figure 4 on a scale of very poor to very good.

This question focuses on several different important aspects of the map system and gains varied responses. The first section is readability, this section was rated good on average but has a high range with 18.2% saying this is poor. Even though this is a very important section of data visualisation there are only a couple of ways to improve this, such as creating a zoom function when the postcode areas are clicked which was a possible extension that was explored. The next section is similar to that of readability. Clarity of the data scored higher on the rating system with 90.9% of testers rating this good or very good. Although over 90% of users rate this above good, one user did rate this very poorly which raised some concerns.

The issue of readability and clarity would always be present when displaying data of postcode areas in the UK. This is due to the fact that postcode areas vary massively in size usually due to population density. For example, the largest postcode area, Inverness (IV) covers a total of 6,243 square miles compared to Western Central (WC), which is the smallest area, covering just 1 square mile[41]. To visualise data with this range using a map system would almost be impossible.

The functionality section of this question had a broad range of answers. 9.1% of users rated this poor and 18.2% rated the functionality as ok. Although this still leaves 72.8% of users rating the functionality as good and very good. With this answer having the most amount of "ok" suggests that the testers may be unsure what the functionality of this map system actually is. Even with these very mixed ratings the minimum target was still met.

The final section to take note of was the General looks of the map system. This scored 72.7% good and very good meaning it passed the predefined minimum satisfaction level. Although this scored well overall it has the highest range in answers with at least one tester picking each option. This could be for several reasons as it is a completely preference-based question. First, it was checked if the user who selected very poor for this question also picked very poor for clarity as then there would be a correlation. Unfortunately, participant 6 who picked very poor for the general looks rated every other section very good. As there is no correlation in this selection it could be argued that there may have been a miss input. There are, however, correlations between most user responses, these can be seen below in Figure 11.

10. Please rate the following attributes of the map system.

11 Responses

ID ↑	Name	Readability	Clarity	Functionality	Colour pallet	General looks
1	anonymous	Good	Good	Good	Very Good	Good
2	anonymous	Very Good	Good	OK	Good	Very Good
3	anonymous	Good	Good	Very Good	Very Good	Very Good
4	anonymous	Very Good	Very Good	Very Good	Very Good	Very Good
5	anonymous	Good	Very Good	Very Good	Very Good	Very Good
6	anonymous	Very Good	Very Good	Very Good	Very Good	Very Poor
7	anonymous	Good	Very Good	Good	Very Good	Very Good
8	anonymous	OK	Good	OK	Good	Very Good
9	anonymous	Poor	Good	Good	Good	Poor
10	anonymous	Poor	Very Poor	Poor	Good	OK
11	anonymous	Good	Good	Good	Good	Good

Figure 11: This image shows each user tester's responses to all attributes in question 10. This image was used to show and determine the correlation between users' answers.

The next two questions in the user survey were open-ended optional questions. The first allowed users to ask questions themselves and give feedback on any missed areas. This was used as user preferences vary and an evaluation method such as a survey often doesn't capture the granularity of feedback. All feedback gathered from this question was useful in improving the final product and there were some common themes.

Theme 1 - Lack of clarity hindering understanding
Across a number of these answers, users seemed to not completely understand what was being displayed to them. Over four of the six total responses to this question, users raised some kind of query about what they were looking at in terms of data or the physical map. Shown below are the user responses relating to this theme.

User 1: "Could have some more writing/ info on it"
User 3: "I don't know the regions of Scotland well enough to know where each location is just by looking at the map…"
User 4: "Functionality looks good but it isn't very clear from the brief image the full extent of this."
User 6: "I can see it's a map, but I'm not sure what is actually being visualised! What does the green area represent? Does the fact that it's green mean anything?…"

As you can see there is general confusion between these user testers. These responses were some of the most important data gathered from this round of testing. They allowed the Author to not only improve the product but also improve the method of evaluation used in the second round of testing.

User 6 gave extensive feedback as they were one of the professionals mentioned in section 7.2.1. This feedback was one of the most important aspects taken into account when designing the second stage of evaluation and further development. This was because not only did it give the author valuable feedback but also insight from someone who has worked with Alexa skills themself.

When these user studies were completed the prototype map system only had one map, this meant that all weather reports were grouped together and the map only showed the number of reports in each area. User 6 asked if the green means anything on the map, this gave the author the idea of separating the data and creating a different map for each weather type. This would not only make the data more impactful in tracking weather but also clearer to the user what they were looking at tackling this theme of data clarity and user understanding.

7.3 Qualitative Evaluation
The author decided that another round of user testing was necessary for the final stages of development. This second testing round would use qualitative evaluation and build on what was learned in the quantitative testing. This method was chosen as it would allow the users to interact with the product developed in a natural way having any questions answered at the time of testing allowing for a deeper understanding of what the goals of the project were and how efficiently they believed it met their goals.

It was decided that several one on one interviews would be conducted with selected participants from the original testing round. The author believed this would gain the best data to perform better thematic analysis than the original round of user testing.

7.3.1 Recruiting Participants for Interviews
Ideally, the interviews would be conducted with users of the professional group mentioned in section 7.2.1. Unfortunately, all members of this group were unavailable for testing. This meant that three other participants were selected, all being students. Although this wasn't the perfect outcome it did have benefits, such as allowing for the interviews to be conducted in person meaning that users could have hands-on experience with both the Alexa skill and the website.

7.3.2 Qualitative Method
Consideration was given to how to best perform these interviews, the author researched how to design and conduct interviews and focus groups successfully. The author decided that to get the most out of these interviews they would follow a semi-structured system. This would allow the interviews to all follow the same structure but would allow users to explore the system themselves and ask their own original questions in their own time.

The users were given access to the Alexa system through the use of the Alexa simulator inside the Alexa Developer Console and were also able to interact with the skill directly through the use of the author's physical Alexa device. The reason that both systems were used was that the author's Alexa Dot does not have a screen so some features were unavailable.

The participants were first given access to the physical Alexa system. As all the participants had experience using Alexa systems there was no need to explain how to interact with them. The author then gave a brief description of how to interact with the skill itself and what its purpose was.

The participants were then prompted to interact with the system, and while doing so they were encouraged to think out loud. This allowed the author to capture as much information as possible. After the first interaction, the participants were then given a list of tasks to perform while the author made notes.

After performing the set tasks, participants were then asked semi-structured interview questions about the tasks they had just completed. The author also used some questions about how users interacted with voice interfaces in day-to-day life. This was done to give a better understanding of the participants and also make the interviews feel less formal and more conversational. The author believed that by making the interactions more conversational a participant would give longer answers allowing for more feedback and data to be collected. As Well as setting an informal environment the author asked prompting follow-up questions when given short answers. The author had planned to conduct three individual one on one interviews originally but due to participant schedules this was changed to a 1-1 and a 2-1 interview.

These interviews were recorded and then transcribed using Microsoft Streams which is accessible through the dictate feature when uploading voice recordings. Then thematic analysis was performed on the transcribed interviews. The transcribed user interviews can be found in Appendix H

7.3.3 Interview Results

The interview results will be shown using thematic analysis. While results of testing were overwhelmingly positive the author has mostly highlighted negative themes as to show what needed to be improved.

Theme 1 - Users not understanding AWS systems in demo

During both interviews participants didn't seem to grasp how AWS systems were interacting. While this wasn't essential the author believed an understanding of background processes would help the participants understand the project as a whole. This was an error as it ended up confusing participants which in turn impacted the interview process as a whole. This can be seen below in the quotes from participants.

Theme 2 - Not being able to zoom on map

During both interview sessions while interacting with the maps participants asked if or assumed they could zoom or enlarge the map to get a better view. This was an intended feature but unfortunately the author didn't manage to get it implemented before the deadline. This would definitely be a recommendation for future work. This would have improved a number of issues associated with the readability of the map system.

Interview 1
00:08:46 Speaker 2
Why won't it let me increase the sizes?
Interview 2
00:05:33 Speaker 2
Can you zoom in?

Theme 3 - Confusion about world map

During implementation testing was done with three.js as a mapping system. This was eventually scrapped but the author had spent some time creating a 3D world map. This was still on the website at the time of testing. Users who were looking through the website themselves came across this but were unsure of its point. The author noticed this and to keep interaction during the interviews asked the participants what they thought could be displayed on this system. During interview two the tester suggested an animation of a day-night cycle. The author at the time liked this suggestion but once thought was put into if this should become a feature the author decided to not implement it as it made no improvement to the project goals.

Interview 1
00:11:08 Speaker 2
Yes, about the globe. What's the point?
00:11:15 Speaker 1
I'm not entirely sure. I spent a while doing it during a testing stage but couldn't bind data.
00:11:22 Speaker 1
Do you have any suggestions on what it could be used for?
00:11:30 Speaker 2
No.
00:11:33 Speaker 2
It looks cool but I'm not sure what it could be for.

Interview 2
00:11:15 Speaker 1
Yeah. So the globe, what, what what would be a good feature for the globe?
00:11:21 Speaker 2
You could put.
00:11:22 Speaker 2
Wind currents on it you could.
00:11:26 Speaker 2
Put rain clouds on it.
00:11:29 Speaker 2
The direction, obviously that's why and.
00:11:32 Speaker 1
So do any of these.
00:11:32 Speaker 2
Ohh, show what parts in daylight.
That's actually quite a good.
00:11:39 Speaker 1
Idea.

Theme 4 - Not liking grey as cloud map colour
During the demonstrations all participants voiced concerns about the colour of the cloud weather report map. At this time the map for this weather report was grey. As the alpha was being changed in conjunction with the number of reports this created a strange effect of the map becoming more and more like the background colour of the website. This made users think that the dark grey was for low numbers of reports and the light grey to be high numbers. This was the opposite of what was actually being reported so the Author decided to gain user input on what colour to change this to. After the interviews it was decided that the cloud map should be changed to purple.

Interview 1
00:04:00 Speaker 2
UM. I'm not sure, grey makes sense but doesn't look good.
00:04:05 Speaker 2
Rain makes sense being blue though, and wind would be purple or something that might look nice. Like which makes sense.
00:04:19 Speaker 2
It's different if you're not doing temperature. I mean this makes sense for that.
00:04:25 Speaker 1
I might do like.
00:04:25 Speaker 1
Purple for wind or cloud?

Interview 2
00:08:06 Speaker 1
The problem I'm having with cloud is the grey colour scheme. It's not probably the best colour scheme. What colour would be good for cloud?
00:08:10 Speaker 2
Yeah, I was going to say it doesn't look good. Green?
00:08:18 Speaker 1
Greens being used for wind.
00:08:20 Speaker 2
Ohh sorry obviously for so you've got blue for rain, yellow for sun, red for cloud.
00:08:30 Speaker 1
That could work.
00:08:38 Speaker 2
Maybe purples, quite a cloud colour.

Overall the interviews were successful and provided the author with a number of different solutions to issues they had faced. It also highlighted a couple key areas that could be improved for the final demonstration of the product to markers. One key issue that the author noticed when analysing the interviews is that they often brought up issues that they knew already existed and would often suggest a fix instead of allowing the user to explore and offer solutions themselves.

7.4 Limitations of the Evaluation Process
Even by using both quantitative and qualitative evaluation methods, there were some limitations of the evaluation process that could be improved.

The largest limitation was both the number and demographic of the participants. Ideally, the participants would have been 60+ as these are the most likely end users. If this was not possible it would have been preferred to have more participants in the professional group, especially in the interviews. By having these users they would likely be more interested in improving the product and engage more throughout testing and interviews.

It would have also been preferable for the users to be able to install the skill on their own device as ease of installation has not been tested. Unfortunately, this is not possible without publishing the skill to the general public.

7.5 Improvements Made Based on Evaluation

Several improvements were made based on the evaluation. The first was adding a different map for each weather type, which allowed for clearer and more accurate visualisation. Another change made to the map system was adding a key below the map that showed the number of reports and the name of the postcode area. This change was once again made to make the map system more user-friendly.

Some of the weather colours were also changed after user feedback was gathered. During the interviews the author asked the testers what they thought of the colours associated with each weather type. As discussed in the interview themes section 7.3.3 many users didn't like the grey of the cloud system as it looked mundane and boring compared to the others. Although grey was an obvious choice for the cloud weather reports map it wasn't a very good colour for data visualisation in practice and was considerably harder to read than the other maps. Taking tester feedback into account the cloud weather type was changed to purple instead of grey.

8. DESCRIPTION OF THE FINAL PRODUCT

The final product is a custom Alexa skill that allows citizen scientists to add data about the weather in their current location using an Alexa device. When activated this skill captures the device location and time, which it then binds to the weather data and adds to an AWS database.

The system uses a DynamoDB database with custom IAM roles that limit and provide access to different AWS components. The database is exported to an S3 bucket which also contains a Vue.js website.

This S3 bucket allows the web application that has access to the database to collect and display this data using choropleth maps. This webpage has a map for each possible weather which is accessible through a navigation bar.

This website allows anyone to view the data added to the citizen science project visually. This was done to boost engagement and activity in the area of citizen science by the general public and users of voice interfaces.

9. ETHICAL AND LEGAL CONSIDERATIONS

During this project, it was important to consider the ethical and legal implications of research, testing and evaluation methods as well as that of the final produced product.

When gathering information through the use of surveys to both create the specification and complete the evaluation, the utmost care was taken to follow the ethical requirements laid out by both the University and the Ethical Application Form.

The final product is not available to the public, which makes the legal considerations more simple. While this is the case the product does abide by all of Amazon's legal and ethical requirements for an Alexa skill and only uses the necessary data for its functions. This application could be published if the author wished but they have decided not to do this as it would incur risks and expose them to unwanted AWS charges.

During testing, demonstration and evaluation only the author's Amazon account was accessed by the skill to mitigate the chance of leaking sensitive information. To test the functionality of the system with a broad range of data the author randomly generated weather reports and locations using a Python script which can be found in Appendix I. This allowed for full testing of the mapping system on the webpage while not exposing users' location data during implementation. Finally all data was stored in consideration to both the Data Protection Act (2018) and GDPR.

The author also took into account the social impact of the project. While the project itself has had little impact on society as a whole the method of citizen science has. If this project was to be implemented into a real life citizen science project there could be great gain due to increased participation by Alexa users.

The final consideration made during the completion of this project was professional. The author referenced the BCS Code of Conduct[42] to make sure that interactions with users, and testers was done to a professional high standard.

10. SUMMARY AND CONCLUSION

10.1 Critical Appraisal

The final product meets all aims and requirements outlined in the project description and designed at the start of the project. This is shown by the results of the evaluation and user testing of the product.

The developed Alexa skill meets all standards, rules and regulations outlined by Amazon and could be released to the public. User data is stored securely by Amazon and collected data does not infringe on any legal laws or legislation. Despite this, there could still be improvements made to all aspects of the project to polish user experience to the perfect level.

If the author could improve any part of the project as a whole, they may decide to focus on learning about the AWS systems sooner as this was thought of as more of a link between the two major development projects, being the webpage and the skill. The AWS systems ended up slowing down development due to poor understanding and original research performed into these areas.

Another aspect of the project the author would have liked to improve would be to opt for a more structured approach to development. The author followed a waterfall approach, but with a project of this magnitude, it may have been better to use an Agile development approach even though the project's development was only being undertaken by the author. This would have also been useful when writing this final report as with a weekly sprint format, writing may have been easier and likely better logged.

Many lessons were learned through the research, development and evaluation of this project. A large amount of knowledge was gathered in systems such as AWS and Alexa. The author also gained knowledge of voice interfaces in general through the reading of academic papers and documentation.

The key factors learned in a project of this magnitude were the importance of design and the effectiveness of rapid prototyping. These methods sometimes do not work effectively together as a developer can easily neglect design for the excitement of prototyping, which in the case of this project did happen. Although this happened the author was able to adapt the poorly designed prototypes into a solid final product that followed standard practices.

10.2 Recommendations for Future Work

Developing voice interface systems is a complex and large area that is rapidly evolving in both its technical ability and use cases. Despite this, the author has outlined what they would like to improve if more time had been given to the project.

The main area that the author would like to improve upon would be the custom skill. While the skill carried out all its needs effectively and efficiently it could be improved to capture wider amounts of data. For example, the skill only allows the user to report four types of weather, while this is enough it would be good if the language model was able to understand more types of weather and categorise these into one of the original weather types. The author started making improvements to this system such as adding the ability for a user to say "Drizzle" instead of rain amd "overcast" instead of cloudy.

While this project didn't target a specific citizen science project, the skill could easily be adapted to manage data gathering for many different projects. This would massively increase its use cases in the real world and would be as easy as adding a new layer to the interaction model allowing a user to select which project they would like to contribute data.

The AWS systems used throughout this project all work together effectively, but there are a number of issues that would need to be addressed if this project were to go live to the general public. The final product exports the database to the S3 bucket once a day at 8:00 AM. This means that the data on the webpage is only updated once a day, ideally, this would be more frequently such as every couple minutes to allow a user to add data and see the map update as close to live as possible. While this would be impractical and expensive on resources it was the original vision of the project.

The webpage is very basic and while it doesn't need to be complex it could have more information and be more user-friendly. In a perfect world and development pipeline would have been implemented for the final project but as the webpage and mapping system was viewed as an extension to the project and so this task wasn't completed in time for submission.

ACKNOWLEDGMENTS

The author would like to thank their supervisor, Dr Brian Plüss for all his guidance and support throughout the project. The author would also like to thank all those who participated in both the user survey and interviews.

REFERENCES

1. 'National Geographic' (2022) *Citizen Science* https://education.nationalgeographic.org/resource/citizen-science-article (Accessed: 28/10/22)
2. 'Citizen Science' (2022) *Citizen Science* https://en.wikipedia.org/wiki/Citizen_science (Accessed: 26/10/22)
3. 'Voice ModevNetwork LLC ' (2023) *Did You Feel It* https://www.voicesummit.ai/alexa-blog/blog-old/did-you-feel-an-earthquake-report-it-to-alexa (Accessed: 04/04/2023)
4. 'SmartHouse UG' *Bird spotting Alexa skill* https://www.amazon.com/SmartHouse-UG-Bird-Spotting/dp/B07L8CKBLJ (Accessed: 04/04/2023)
5. ' Institute of Electrical and Electronics Engineers, C Jiménez, E Saavedra, G del Campo, & A Santamaría' (2022) *Alexa-based voice assistant for smart home applications.*

   https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9475608 (Accessed: 04/04/2023)
6. 'Science Direct, S. Kim, A. Choudhury' (2021) *Exploring older adults' perception and use of smart speaker-based voice assistants: A longitudinal study* https://www.sciencedirect.com/science/article/abs/pii/S0747563221002375#preview-section-cited-by (Accessed:04/04/2023)
7. 'White Rose Research Online, R. Pateman, A. Dyke, S. West' (2021) *The Diversity of Participants in Environmental Citizen Science* https://eprints.whiterose.ac.uk/173813/ (Accessed: 04/04/2023)
8. 'GROW' (Online) *Build better soil through citizen science* http://growobservatory.futureeverything.org/ (Accessed: 27/12/22)
9. 'Amazon' (Online) *Amazon S3 FAQs* https://aws.amazon.com/s3/faqs/#durability (Accessed: 05/04/2023)
10. 'Amazon' (Online) *Amazon S3 FAQs* https://aws.amazon.com/s3/faqs/#availability (Accessed: 05/04/2023)
11. 'IBM' (Online) *What is natural language processing?* https://www.ibm.com/topics/natural-language-processing (Accessed: 22/02/2023)
12. 'Institute of Electrical and Electronics Engineers, A. Chhetri, et al' (2018) *Multichannel Audio Front-End for Far-Field Automatic Speech Recognition* (Accessed: 05/01/2023)
13. C. Pearl, Designing Voice User Interfaces: Principles of

Conversational Experiences, O'Reilly, 2016.
(Accessed: 21/12/2022)

14. C. Pearl, Designing Voice User Interfaces: Principles of Conversational Experiences, O'Reilly, 2016.
Chapter 2, Basic Voice User Interface Design Principles, Page 20
(Accessed: 21/12/2022)
15. C. Pearl, Designing Voice User Interfaces: Principles of Conversational Experiences, O'Reilly, 2016.
Chapter 6, User Testing for Voice Interfaces, Page 179
(Accessed: 15/3/2023)
16. 'CISION PR Newswire' (2020) *Voiceitt makes Alexa accessible for people with disabilities*
https://www.prnewswire.com/il/news-releases/voiceitt-makes-alexa-accessible-for-people-with-disabilities-301185816.html
(Accessed:04/04/2023)

17. 'ACM Digital Library, A. Pradhan, K. Mehta and L. Findlater' (2018) *"Accessibility Came by Accident": Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities*
(Accessed: 25/01/2023)
18. Andy Kirk, Data Visualisation a Handbook for Data Driven Design, SAGE Publications 2016
(Accessed: 05/01/2023)
19. Andy Kirk, Data Visualisation a Handbook for Data Driven Design, SAGE Publications 2016
Page 200
(Accessed: 05/01/2023)
20. 'Kanbanize', (Online) *What is a Kanban Board and How to Use It? Basics Explained*
https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board
(Accessed:15/03/2023)
21. TeamGantt, "TeamGantt,"(Online).
https://www.teamgantt.com/
(Accessed: 08/04/2023)
22. Adobe, "Figma", (Online)
https://www.figma.com/
(Accessed: 27/01/2023)
23. Amazon, "Alexa Development Console", (Online)
https://developer.amazon.com/alexa/console/ask
(Accessed: 15/11/2023)
24. Amazon, "Slot Type Reference"
https://developer.amazon.com/en-US/docs/alexa/custom-skills/slot-type-reference.html
(Accessed: 13/12/2022)
25. Amazon, "Amazon Alexa"
(Mobile Application)
(Accessed: 01/02/2023)
26. Royal Mail, "What is a postcode area?"(Online)
https://personal.help.royalmail.com/app/answers/detail/a_id/155/~/what-is-a-postcode-area%3F
(Accessed: 9/04/2023)
27. MapBox, "Mapbox" (Online)
https://www.mapbox.com/
(Accessed: 01/03/2023)
28. Google, "Google Maps Platform - GeoCoding API" (Online)
https://developers.google.com/maps/documentation/geocoding/overview
(Accessed: 03/03/2023)
29. Adobe, "Figma" (Online)
https://www.figma.com
(Accessed: 15/02/2023)
30. Amazon, "Choose the Invocation Name for a Custom Skill" (Online)
https://developer.amazon.com/en-US/docs/alexa/custom-skills/choose-the-invocation-name-for-a-custom-skill.html
(Accessed: 12/02/2023)
31. Butte College, "Definite and Indefinite Articles (a, an, the)" (Online)
https://www.butte.edu/departments/cas/tipsheets/grammar/articles.html#:~:text=Articles%20are%20used%20before%20nouns,its%20identity%20is%20not%20known.
(Accessed: 10/04/2023)
32. Amazon, "Create Intents, Utterances, and Slots" (Online)
https://developer.amazon.com/de-DE/docs/alexa/custom-skills/create-intents-utterances-and-slots.html
(Accessed: 09/11/2022)
33. Amazon, "Best Practices for Sample Utterances and Custom Slot Type Values" (Online)

https://developer.amazon.com/de-DE/docs/alexa/custom-skills/best-practices-for-sample-utterances-and-custom-slot-type-values.html
(Accessed: 11/11/2022)

34. Amazon, “Configure Permissions for Customer Information in Your Skill” (Online)
https://developer.amazon.com/en-US/docs/alexa/custom-skills/configure-permissions-for-customer-information-in-your-skill.html
(Accessed: 15/02/2023)

35. Paloalto Networks, “What Is the Principle of Least Privilege” (Online)
https://www.paloaltonetworks.com/cyberpedia/what-is-the-principle-of-least-privilege
(Accessed: 15/03/2023)

36. Vue.js, “The Progressive JavaScript Framework”(Online)
https://vuejs.org/
(Accessed:11/04/2023)

37. Microsoft, “The Model-View-ViewModel Pattern”(Online)
https://learn.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm
(Accessed: 11/04/2023)

38. Vue.js, “Quick Start” (Online)
https://vuejs.org/guide/quick-start.html
(Accessed: 25/02/2023)

39. Npmjs, “npmjs” (Online)
https://www.npmjs.com/
(Accessed:04/03/2023)

40. Data-Driven-Documents, “D3.js”(Online)
https://d3js.org/
(Accessed: 10/03/2023)

41. The Postal Museum, “12 Interesting Facts about Postcodes” (Online)
https://www.postalmuseum.org/blog/12-interesting-facts-about-postcodes/#:~:text=WC%20in%20London%20is%20the,which%20covers%206%2C243%20square%20miles.
(Accessed: 14/04/2023)

42. BCS The Chartered Institute of IT, “BCS Code of Conduct” (Online)
https://www.bcs.org/
(Accessed:24/04/2023)

APPENDIX

A. User Study Questions
B. User Study Responses
C. Final Ghantt Chart
D. Mid Term Report
E. User Flow With Notes
F. Informed Consent Form
G. Participant Information Sheet
H. User Interview Transcript
I. Mock_Data_Generator
J. Main Code
    - Skill Code
    - Website Code