MSc Computer Science Thesis: Leonard Goldschmidt

*Smart Assistant for Data Collection in Epidemiology*

1. Main Body

# Aims and Objectives

Meeting the aims and objectives listed below should result in successful completion of this project. For clarity purposes, the aims are split into high level and low level. High level aims focus on the overall outcome of the study, whereas low level aims focus more on application development. The minimum work required is laid out in the key deliverables section.

* + 1. High Level Aims

The study is centered around epidemiology as well as utilizing modern technology for the relevant research. Therefore, the high-level aims are:

1. To determine if it is possible and how to implement and application that is able to run on an Amazon Alexa and/or Google Home device, as well as understanding the back-end differences for both platforms.
2. To build an application, runnable on a smart assistant, that allows a user to food and/or drink data by saying phrases like *“For lunch I had an apple and a diet coke.”*
3. To determine if an application, running on a smart assistant, can provide an alternative food logging technique for epidemiological research.
   * 1. Low Level Aims

These low-level aims are introduced to give clear guidelines for necessary development milestones for a successful project implementation. The low-level aims are:

1. To create a custom skill for a smart assistant that can translate verbal input and interpret food information.
2. To create a method of storing the food information in a centralized database.
3. To be able to distribute the skill to multiple users.
4. To assess this alternative food dairy method in comparison with traditional methods used in epidemiology.
   * 1. Objectives

The objectives of this study are:

1. To evaluate and compare *Amazon Alexa* and *Google Home* in terms of criteria including capabilities, performance, limitations, and back-end implementation options.
2. To determine how interchangeable back-end systems developed for these two devices are. For example, whether the same back-end system be used for both devices, or some degree of adaptation between them, and how to minimize any needed adaptation through appropriate software design.
3. To prototype an implementation of a voice-driven food diary system for collecting food and drink information from participants in research studies. Ideally, the system should be designed such that it can be integrated with both *Amazon’s Alexa* and the *Google Assistant*. If the project demonstrates that this is not feasible then *Alexa* or *Google Assistant* will be chosen, depending on which was deemed more appropriate from evaluation of objective 1.

# Key Deliverables

For successful completion of this study, a set of minimum key deliverables are set as follows:

1. An investigation of similarities and differences when programming applications on Amazon or Google developer platforms.
2. An application running on a smart assistant that enables translating and interpreting speech and storing key food data.
3. A study report with detailed information about the process throughout development of the application.
4. Background and Context

Health research is interested in understanding factors that influence people’s food intake and how different diets affect health. Collecting reliable and accurate dietary information is a difficult process in this area and can result in inconsistent and poorly evidenced data for epidemiological research.

*“By definition, epidemiology is the study (scientific, systematic, and data-driven) of the distribution (frequency, pattern) and determinants (causes, risk factors) of health-related states and events (not just diseases) in specified populations (neighborhood, school, city, state, country, global).”[[1]](#endnote-1)*

History provides multiple studies that show diseases that appeared as result of a certain diet. Most commonly, the affected had certain deficiencies in their diet. Traditional methods of identifying diseases caused by specific diets involve asking participants to complete food frequency questionnaires either on pen and paper, mobile or web applications. Methods like these require significant time and effort, possibly resulting in either inconsistent or inaccurate data. Reason being, that participants may enter wrong information, miss to log meals, or lack effort due to the tedious and time-consuming logging process. Therefore, an alternative method of collecting food data should be assessed. It may be possible to utilize arising technologies such as voice assistants to build an application that enables tracking food intake by simply speaking to a smart speaker. This process may come more natural to participants, thus, potentially being quicker than traditional methods. This section will describe cases of traditional epidemiological research and how effective the given nutrition logging techniques were.

A study carried out by James Lind in 1753[[2]](#endnote-2) showed that lemons and oranges had positive effects on the illness. Ultimately, scurvy was found to be a vitamin C deficiency. In the late nineteenth century the illness *Beriberi*, most commonly found with infants, occurred with a group of sailors, whose diet largely consisted of polished rice. Scientist *Takaki* found that the disease was eliminated by adding milk and vegetables to their diet. The cause was later tracked back to a lack of thiamine, which is a vitamin found in milk and vegetables. In 1987, a study showed that the common occurrence of *Kesahn* disease in central China, was caused by a selenium deficiency. It becomes apparent that nutritional epidemiology relates commonly to deficiency type diseases. These diseases usually take months of starting a deficient diet to become apparent and may be cured quickly by increasing the deficient vitamin amount. It may therefore be ideal if the nutritional reason for a deficiency within a large group can easily be identify. Thus, an easy and centralized way of tracking nutrition for a large group is needed. In many epidemiological studies, existing data is used. However, these studies often incorporate a small group of subjects. For accurate results, the field data needs to be clean, valid, reliable and should not take long to gather. Traditionally, collection of this type of data involves questionnaires that are handed to the subjects and filled in manually over a period of time.

*“The second generation of the Avon Longitudinal Study of Parent and Children”[[3]](#endnote-3)* involves logging and tracking behavioral data, including nutrition, of over 14,000 participants since their birth in 1992. Having such a large group of participants and collecting a wide-range of social, lifestyle, clinical, anthropometric and biological data on all family members repeatedly, requires reliable collection methods. As this study involves so many different areas of data, separate methods of collection can be investigated. Epidemiological studies have shown that nutrition can affect health majorly and thus provides an important section of the Avon Longitudinal study. The research was set up to provide a unique multi-generational cohort and represents the potential scale that is sometimes needed for epidemiological studies. The study has been ongoing for over 26 years and is now a study ranging over three generations.

1. Development

# Introduction

In order to successfully create an application that is runnable on a smart assistant device, that can be distributable to many users, as well as collecting and story data in a centralized database, several components were necessary. The diagram below (Figure 1: Full Stack Application) provides an overview of the stack used to implement the system. The application stack consists of several different parts and to understand it fully, one first has to define the individual components (from right to left):

1. **User**
   1. The users only responsibility is to speak to the smart assistant and tell it to log certain food and/or drinks items.
2. **Smart Assistant**
   1. The smart assistant (Echo Dot for this study), running a custom Alexa Skill, waits for the correct invocation phrase and then interprets the users’ phrase.
   2. The custom Alexa Skill populates custom dietary slots. The skill was trained using sample phrases, so that it populates the slots with the right information and prepares a JSON object.
3. **Web Connectivity**
   1. The smart assistant sends the JSON Object to a configured endpoint. Jojo.epi.bris.ac.uk:9449
4. **Server**
   1. The endpoint is configured to speak to the server jojo.epi.birs.ac.uk, which is located at the University of Bristol. This server contains the backend logic, including the database connectivity.
5. **Docker**
   1. *“Docker is a tool designed to make it easier to create, deploy, and run application s by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”[[4]](#endnote-4)*
   2. The server contains a docker cluster, running different applications. A docker image was created containing the back-end logic of this application.
6. **Docker Image**
   1. The docker image has its own stack containing the following components:
      1. *Apache Tomcat Server*
         1. *“The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies”[[5]](#endnote-5)*
         2. This server runs and deploys the java application.
         3. The server is configured to support SSL (https), which increases security and is necessary to work with Amazon skills.
      2. *Java Application*
         1. This application contains the bulk logic of the web application.
         2. Intent-handlers handle the triggered intent that was sent to the server from the smart assistant via the JSON object.
         3. The handlers implement logic such as storing the data or returning useful responses back to the smart assistant.
         4. To connect to the Alexa skill, a java servlet is used, containing the skill ID.
7. **Database**
   1. To close the stack loop, the nutrition data needs to be stored in a save location.
   2. A *mySQL* database, running on the server, is used to store the nutrition data for each user.

By putting all these components together, it is possible to provide a full stack application that allows a user to use a skill in a food diary like fashion. However, due to the integrity of the full stack, each component required basic proof of concepts, which later increased in complexity to implement full desired functionality. The following section provides a detailed explanation of each development step.



Figure 1: Full Stack Application

# Building a basic Alexa Skill

To get familiar with developing skills for an Alexa supported device, a simple custom skill was set up, with no custom functionality. There mains steps needed to create a custom skill are:

1. *Create an invocation name*

The name given to a custom skill, also acts as the invocation name. This means that the custom skill can be lunched by saying “*Alexa, open {invocation name}”*.

1. *Create Intents*

An intent represents a functionality of the skill. For example, if the skills purpose is to book a flight, one intent might be called “*bookFlightIntent*”, which processes the speech input and populates slots with the necessary flight information. A skill can have any number of intents. A custom skill comes with five pre made built in intents, provided by Amazon. These are basic intents that handle simple operations such as closing the app or providing help. To ensure the skill understand which and when a certain intent should be selected, the developer should provide a list of sample utterances. A minimum of thirty different utterances should be given. To ensure the correct slot types are populated, the slots can be included in a sample utterance. See Figure 2: Sample Utterance for an example of such utterance. Slot types can be included by wrapping them in curly brackets “{}”.A close up of a logo

Description automatically generated

Figure 2: Sample Utterance

1. *Create Slot Types*

A slot type represents a value that can be populated by the user. Taking the flight booking intent example, one may require a slot “*departureCity*” and slot “*destinationCity*”. To ensure that the skill understands what kind of values may go into these slots, the developer should provide each slot with a large set of sample values. Alexa uses these sample values to train the model once the user has finished configuring the skill.

1. *Set Endpoint*

To publish the skill and make it available for testing, an endpoint needs to be set. This should generally be the URL that connects to the server, which handles the back-end logic. There are two options for the endpoint. The recommended one is *“AWS Lambda ARN”.* Essentially, this means hosting the skill on the Amazon cloud. To use a custom endpoint, the second option “HTTPS” should be chosen. For basic skills and to test basic functionality, Lambda was chosen initially throughout developing this study.

1. (*What is Epidemiology? | Teacher Roadmap | Career Paths to Public Health | CDC*, no date) [↑](#endnote-ref-1)
2. (Willett, 2013) [↑](#endnote-ref-2)
3. (Northstone *et al.*, 2019) [↑](#endnote-ref-3)
4. (*What is Docker? | Opensource.com*, no date) [↑](#endnote-ref-4)
5. (*Apache Tomcat® - Welcome!*, no date) [↑](#endnote-ref-5)