Health Concierge

Daniel Hwang, Ride Bu, Nairong Zhang

HCI Project4

Introduction

The on-demand gig economy, voice-controlled mobile device assistants, and digital health applications were the primary sources of motivation for our voice initiated digital health application. Using a Google Assistant application as a remote control for accessing and interacting with a digital health web application, we explored design flows for creating a novel system for users within the contexts of accessibility, HCI + AI, and usability.

Fluidity and efficiency have played major roles in the growing establishment of the gig economy. Many types of concierge-type apps from food delivery, cosmetics, and laundering services show a glimpse of the future of both work in the gig-economy to the demands that drive it. The digital healthcare ecosystem is a space that can also benefit as well. From the inconvenience of medication procurement to serious medical emergencies, fluidity and efficiency are sorely lacking. The American regulatory environment is a byzantine ecosystem to navigate and has represented high barriers to entry for these types of concierge services.

We present a digital health application system that is accessible via voice commands to interface with a health based app that can provide concierge type services under the principles of human computer interaction.

Problem Space

The problem space that our group tackled was under the contexts of Accessibility, HCI + AI, and Usability for the digital health ecosystem. These concepts spanned the ideation of our design and motivated the key components involved with building our application. We mainly wanted to focus on how users would be able to interact with voice assistants to control access to applications related to digital health data and monitoring. In essence, the application system that we designed can be likened to that of a voice-based remote controller for digital health applications. We constructed a Google Assistant remote control to interface with a digital health application we designed. Not only do we see this system as applicable to the digital health application we created, we can also see how this system can be integrated into other digital health applications as well.

Access and Accessibility

To emphasize, the first problem we addressed was one of ease of access and accessibility. We felt that the user flow in which digital health applications could be accessed was clunky and not familiar. There is no doubt that native mobile applications are accessible at the touch of a button on a mobile device. It is also apparent that website are also just a query away. However, we sought to explore an area of familiar voice control to access applications. We wanted the user to feel in control during the process of accomplishing particular tasks. This alternative user flow to access applications ultimately attempted to give the user faster access beyond the traditional search and click methods as well as give the full opportunity to on-demand medical application services.

Web Application

The next problem we addressed was on the web application side. Here we attempted to explore the space in which what types of digital health data could be accessed and used. The problem with digital health applications is representative of the fragmented nature of the US healthcare system. Thus it is difficult, if not impossible, to have a universal aggregator of the data you own with clear flow and connection to different services. Beyond existing health monitoring applications and devices that exist, we sought to extend these services to include the on-demand nature of today's apps into ours. We showcase the possibility of how digital health applications could provide pharmaceutical delivery and telemedicine services from an application. Perhaps, difficult in the real world, this component of the project is important in understanding the integration of modern tools fitting to modern solutions in the digital health space.

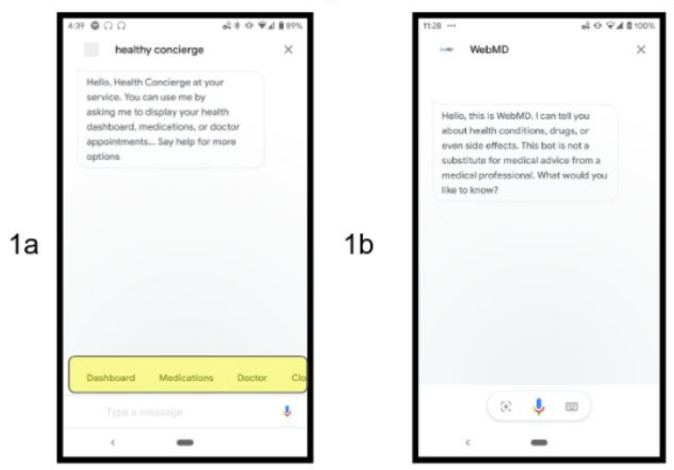
Methods and Implementation

The best way we believed we could understand the use and interaction of particular digital health and assistant applications was through a limited user-centered design of contextual inquiry. As there were not any voice activated remote control digital health applications, we conducted each contextual inquiry separately. We wanted to understand, foremost, the way users interacted with the Google Assistant and how they interacted with digital health applications.

Contextual Inquiry

The contextual inquiry of the Google Assistant involved a user, a non-student working male, who has had some experience using the Google Assistant and was familiar with the general basics of the appropriate invocations--such as starting it up with 'Ok, Google'. We mention that this particular approach of contextual inquiry was limited, specifically for the Google Assistant part because many if not most people (even the user who had some experience with the Google Assistant) were not aware that there could be Google Assistant apps within the Assistant Ecosystem. For example, for the purposes of the reader, Figure 1 shows two Google Assistant apps that have specific functionality and logic when invoked within the Google Assistant Ecosystem. Once the application is invoked with Google Assistant, the invoked application behaves programmatically within that context. We therefore made it clear that initial invocation into Google Assistant applications would be to call upon the application after an "Ok, Google. Talk to health concierge". Beyond us explaining how to interact with the Google Assistant application, we took a backseat to the process and allowed the user to navigate through the application and options. Another component that we needed to address was that the 'think-aloud' process was controlled by angling away from the device so it would not pick up the audio as commands or if it did pick it up, we would just cancel the voice command. In addition to interaction with a generic Google Assistant application, we also had the user poke around rough prototype flows of our in-progress application's design ideas.

The primary comments that were taken from the Google Assistant part was that the user's comments of "the helpful hints" were useful in understanding how to navigate through the application. This was understandable in that there are no traditional menus in a voice-based application. Further, we extended the voice activated 'help hint' with suggestion boxes in our application as shown in the highlighted portion of Figure 1a. These suggestion boxes act as clickable buttons that appear within the Google Assistant application to aid where no menu exists. Especially in the voice activated portion of the application, it was necessary that there were troubleshooting methods available--such as the help methods. Further, the user was very responsive to the detailed interactivity that displayed the correct information and the buttons that acted as portals to other parts of the application and the web application. Though there were comments that made it seem as if the voice responses from the Google Assistant were "too lengthy and took too much time when returning to parts i've been before" we found that it was a system that was unavoidable. We thought some parts we could work on in future work could be ways to speed up Google Assistant audio or have bookmarks in the Google Assistant system to cache information if users have already seen certain navigation points.



Hello, Health Concierge at your service. You can use me by asking me to display your health dashboard, medications, or doctor appointments... Say help for more options

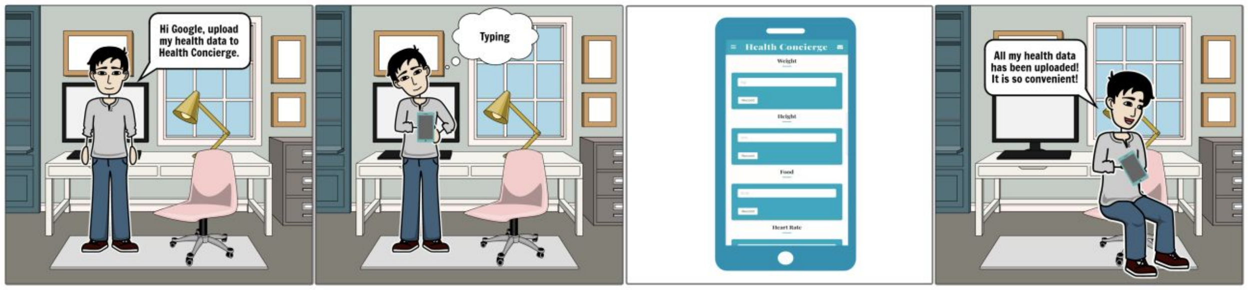
Hello, this is WebMD. I can tell you about health conditions, drugs, or even side effects. This bot is not a substitute for medical advice from a medical professional. What would you like to know?

Figure 1. Invocation of the Google Assistant application 'healthy concierge' and 'webMD'

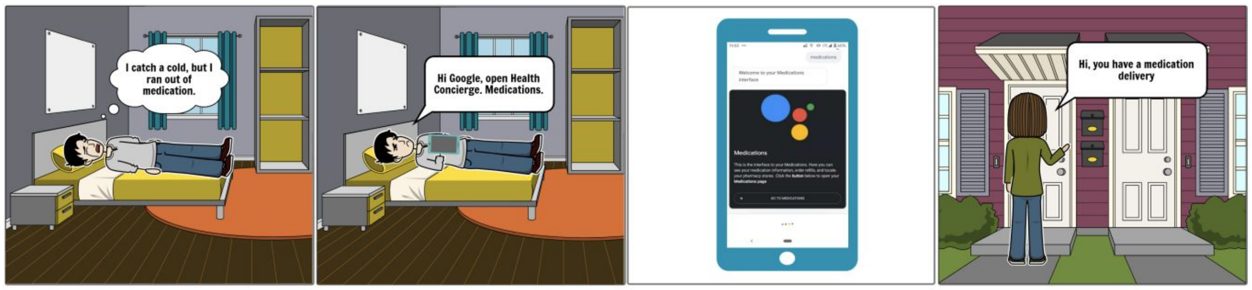
The contextual inquiry of the web application was straightforward as there were not too many items that may have seemed novel in comparison to the Google Assistant application. When working with the webMD interfaces the user, a current student, was able to successfully navigate the website as it was "intuitive and familiar". Further, when introducing initial designs of our web application, it was "easy to navigate and understand what to do". From a digital health application and extension of an application interface that can be entered and used for usability, we found that clear direction and easy to understand interfaces were important. We found that the ability to enter into different part of the application and not have to enter into nested webpages, as we were familiar with other healthcare webpages were prone to do, was an important point of feedback and design choice. Our understanding of the space and bad design choices from different attempts at digital health web applications combined with the feedback from users using the different types of applications and our initial design motivations led us to iterate on additional extensions of navigation and content aggregation. An additional point we found important was that, although difficult and perhaps a bit of a regulatory hurdle, we wanted to showcase the possibility of having different healthcare products available in one place. These included the ability to have pharmaceutical delivery and telemedicine video calls. In the real world, regulatory hurdles would be frighteningly high, but the interaction of a successful aggregation point for these services, we found to be valuable and useful from users.

Storyboards

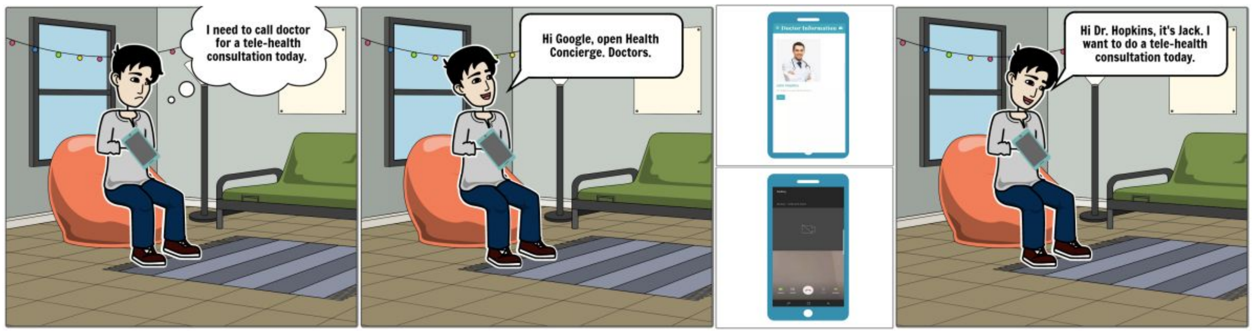
During our designing phase and subsequent implementation, we found that storyboarding was a method that we found most important. We focused the storyboarding on different tasks and goals of how users would reach for them. Starting from points of Google Assistant enabled devices in hand, users would navigate from voice to application.


Hi Google, upload my health data to Health Concierge.
Typing
Health Concierge
Weight
Height
Food
Heart Rate
All my health data has been uploaded! It is so convenient!

1) User wants to upload data about her health onto the dashboard and so calls up the app and uploads the data


I catch a cold, but I ran out of medication.
Hi Google, open Health Concierge. Medications.
Medications
Hi, you have a medication delivery

2) User ran out of medication and wants to order more medication. User calls up the app and orders more medication


I need to call doctor for a tele-health consultation today.
Hi Google, open Health Concierge. Doctors.
Doctor Information
Hi Dr. Hopkins, it's Jack. I want to do a tele-health consultation today.

3) User needs to jump on a call with doctor for a telehealth consultation and so calls up the app and clicks on a link to be taken directly to a video call with the doctor

Work Model

Sequence Model

"ok google" revoke the google assistant

"open the health concierge app", opens the app

First time user?

Register login in (load from cache)

Create credentials

Upload personal health related information
(Link activewear and other health accounts)

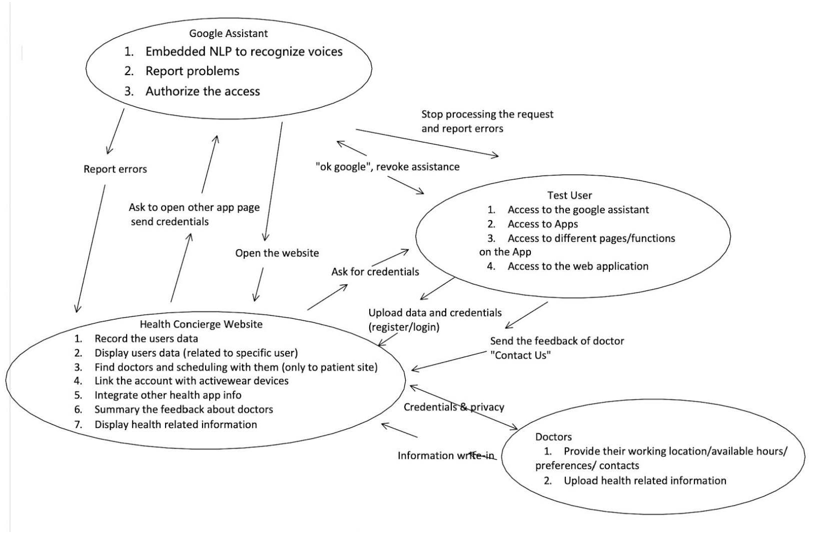
Credentials from outside resources

Display the current data

Jump to dashboard

"ok google, check my medicine"	"ok google, call my doctor"	"ok google, medicine delivery "	ok google, contact the health concierge
Open up with "My Medicine Page"	Direct to calling page (Associate with favorite doctor)	open up medicine shop page	open up contact page
Search other mediations to update		search and purchase OTC medicine	text in message and send

Flow Model

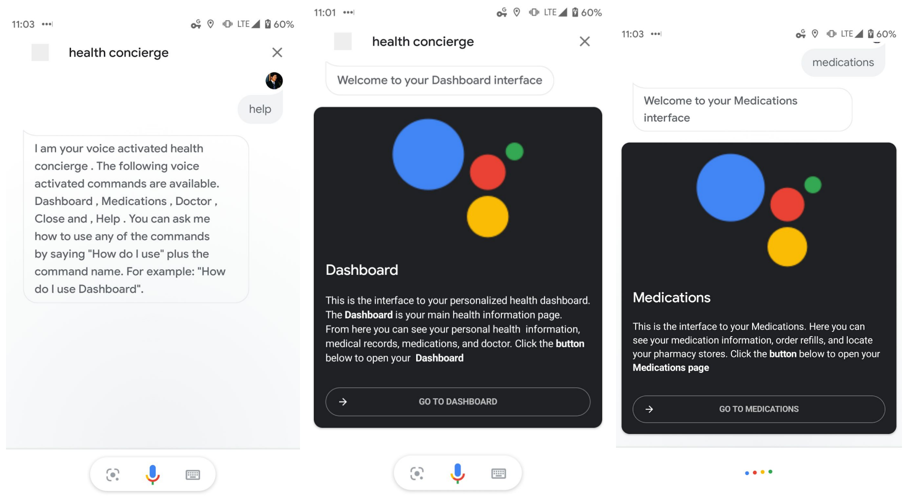

Google Assistant
1. Embedded NLP to recognize voices
2. Report problems
3. Authorize the access
Stop processing the request and report errors
Report errors
"ok google", revoke assistance
Test User
1. Access to the google assistant
2. Access to Apps
3. Access to different pages/functions on the App
4. Access to the web application
Ask to open other app page send credentials
Open the website
Ask for credentials
Upload data and credentials (register/login)
Health Concierge Website
1. Record the users data
2. Display users data (related to specific user)
3. Find doctors and scheduling with them (only to patient site)
4. Link the account with activewear devices
5. Integrate other health app info
6. Summary the feedback about doctors
7. Display health related information
Send the feedback of doctor "Contact Us"
Credentials & privacy
Information write-in
Doctors
1. Provide their working location/available hours/ preferences/ contacts
2. Upload health related information

Implementation

The implementation of our application system can be broken up into two different systems: The Google Assistant (which we can refer to as the 'remote') and the Web Application.

Google Assistant

The Google Assistant application was developed using the Dialogflow components in the Google Assistant Developers console. The Dialogflow components represented custom logic that ingest voice commands, interpret them, and programmatically render or redirect to different application interfaces within the Google Assistant application or to the health concierge web application we developed. The Dialogflow components we used are referred to as Intents within the Google Developers console and interact primarily with the application we created with it called health-concierge. Invocation of the Google Assistant app is done by accessing Google Assistant by saying, "Ok, Google" and then accessing the health-concierge application by saying, "let me talk to health-concierge". This interaction process is shown in Fig.3. Currently, our assistant app is used as a test version as usage on public devices must pass a publishing step for the public Google Store. As indicated, the Google assistant interface acts as the main entrypoint into the web application we created for the purposes of quick and easy accessibility. The Dialogflow component uses natural language processing and tokenizes voice and text input to activate particular commands called the Intents. Appendix A describes the descriptions of the intent parameters and the intents we created to handle the logic. Appendix B describes the webhooks involved to redirect to the web application component that we describe next. Appendix C describes the Google Assistant system design for our application flow.



I am your voice activated health concierge . The following voice activated commands are available. Dashboard , Medications , Doctor , Close and , Help . You can ask me how to use any of the commands by saying "How do I use" plus the command name. For example: "How do I use Dashboard".

Welcome to your Dashboard interface

Dashboard

This is the interface to your personalized health dashboard. The **Dashboard** is your main health information page. From here you can see your personal health information, medical records, medications, and doctor. Click the **button** below to open your **Dashboard**

Welcome to your Medications interface

Medications

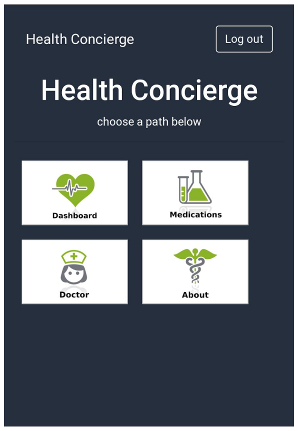
This is the interface to your Medications. Here you can see your medication information, order refills, and locate your pharmacy stores. Click the **button** below to open your **Medications page**

(a) (b) (c)

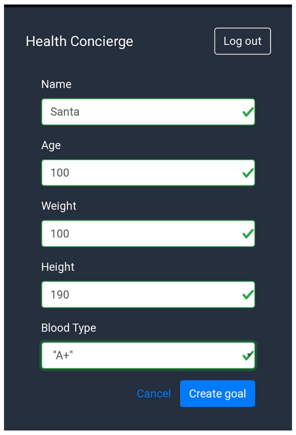
Fig.3 (a). Call health concierge by saying: "Ok, Google. Let me talk to health concierge". (b). Google Assistant helps to open the dashboard of health concierge. (c). When the user says "Medications", the Google Assistant helps to open Medications Interface which acts as a portal into the web application

Web Application

The web application side of our overall application represents the database and primary health information pages. This was primarily built using a suite of AWS tools to handle user authentication, the database, and the hosting. The tools in the AWS stack we used were Cognito for authentication, S3 for the web asset storage, Lambda and API Gateway to handle arbitrary server functions, and DynamoDB for specific user data storage. Further implementation details can also be seen in README of our github repository provided in the Appendix. The importance of having the two separate components to our total application was to take advantage of and illustrate the capabilities of a voice interface from the Google Assistant and the obviously feature-rich platform we could have from a traditional web application interface. The web application display fundamental dynamic components of register, sign in, login, information update, medication ordering, and video calling. The main pages we created for the application included the following pages: about, dashboard, doctor, and medications.

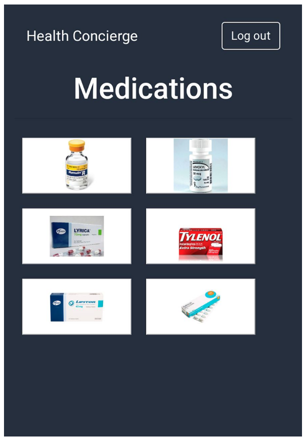


(a)

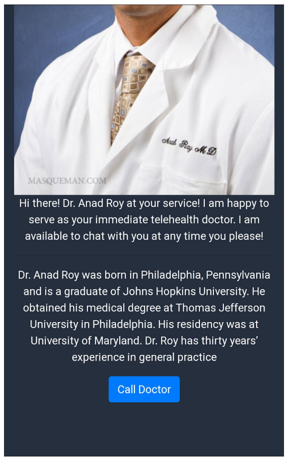


Name: Santa
Age: 100
Weight: 100
Height: 190
Blood Type: "A+"

(b)



(c)



Hi there! Dr. Anad Roy at your service! I am happy to serve as your immediate telehealth doctor. I am available to chat with you at any time you please!

Dr. Anad Roy was born in Philadelphia, Pennsylvania and is a graduate of Johns Hopkins University. He obtained his medical degree at Thomas Jefferson University in Philadelphia. His residency was at University of Maryland. Dr. Roy has thirty years' experience in general practice

(d)

Fig.4 Web application pages: (a). The dashboard of Health Concierge (b). Health Monitor (c). Medications Shop (d). Call Doctor

Evaluation Findings and Outcome

During test drive and showcase of our application system with our test user we found that there were both positives and negatives to the system we created. As it was a novel approach to interacting with directed digital health applications, there was of course a learning curve and some necessary guidance that spurred the necessity of immediate help and tutorial messages from the Assistant's starting of the app. There were also issues of repetitive voice commands that were brought up during the user's testing of the application. We found this as a drawback in our view because one of the goals we wanted was fluidity and efficiency for the system navigating from wanting to accomplish a task (thinking of ordering medication) to getting to accomplish the task in the application (web app medications page). However, because the Google Assistant side of the application did not have any way to know that the user either already knew the commands or did not want to hear the command tutorial again, it wound up repeating many of the same commands. Another drawback we understood was the real world feasibility of the web application. There would be of course many regulatory hurdles involved with being able to aggregate healthcare services onto one platform which we were able to simulate, but not recreate for a real world scenario. The positive feedback we received included the novelty aspect of the system. As mentioned, there are taught ways of accessing systems. We are taught to enter a web address to access a web application. We are also taught to search up a native mobile application to access a native app and click to enter. However, as there are stark differences between web browsers being retro-fit to mobile screens, we believed that the interaction between mobile and web application can be improved upon and more frictionless. We saw that this system could be a more natural bridge that could connect initiated goal (thoughts of accessing certain digital healthcare services) through an immediate mobile (or on hand) device to the more powerful and feature rich web applications and portals. Further, the natural language interactivity that the Google Assistant served as for our "remote's" interface was received with positivity. There were comments on how more friendly and easily the user could want to interact with an 'AI' voice-based interface to access products and how it felt more natural and easier to go about doing the task in contrast to formally making time to sit down at a computer to go about ordering medications. The language aspect we found to be a more natural a welcoming bridge to accomplishing such goals. Overall, we had an interesting look into this alternative form of human computer interaction and bridging technologies for better usability and accessibility. We believe that the blending of technologies and how they interact with each other will continue in ways that we may not perceive, but our attempt shows just one approach to the many we believe will come.

Limitations and Future Work

Some future work we would explore is to have direct signup and login access be rendered seamlessly from the assistant. We think that is a reasonable and achievable goal because the website, although it could be accessed from a regular web browser, when accessed from a phone, it would be immediately personal to the user. A person accessing the site from his/her Google Assistant would already have a google account email associated. As such, the signup and login process can be eliminated--thus removing a further point of friction.

Also, as we saw from the testing from our users, further voice integration into the web application itself was another path to removing barriers under the context of accessibility. Voice controlled data input and control of a web application is something that can be done in controlled parts of the application. This includes data input, voice authentication, medication ordering, and telehealth conferencing. The continued blending of usage between voice controlled assistants and interaction with applications is an interesting path between HCI and AI assistants. As web based and mobile applications reside in much more feature rich domains, voice control with these assistants (the Google Assistant in our case) can provide further functionality in ease of use. These accessibility features can even encompass usage for those without the ability to type or as tools to simulate access to a keyboard. This being provided gives more potential access to using applications that may have been difficult or impossible to use. Another point of improvement and future work is also to streamline the voice assistant aspect of the application. During contextual inquiry, there were points in which repeated dialogue from the Google Assistant may have seemed repetitive or too lengthy--especially if the user had visited those particular navigation tooltips before. Perhaps working on speeding up the voice dialogue or including suggestion buttons at the bottom of the Google Assistant interface to either skip or speed up could be options to extend as well.

Appendix

Github Repository:
https://github.com/JudyZnr/HCI-Project4-master

- includes initial to final writeups (including this one) as well
- includes code from:
    - Google Assistant in folder ‘GoogleAssistantLogic’
    - Web Application in folder ‘WebApp’
        - contains source code and AWS lambda functions

Appendix A. Dialogflow Intent parameters and Intents.

Dialogflow Intent Parameters

Name	Name of the intent
Context	used to remember the parameter value and can be passed between related intents
Events	An optional way to trigger a particular intent without the need for a matched text or spoken input. Uses Google’s predefined platform specific events or can use custom-defined ones
Training Phrases	Phrases that should be expected from users trying to trigger this particular intent
Actions and Parameters	Saves tokenized user input or voice input for use in current or future action
Responses	Response from Google Assistant application after accomplishing the intent action or failure fallback
Fulfillment	The code deployed through a web service to provide additional data to the user

Dialogflow Intents

Intent Name	Dashboard

Context	N/A
Events	N/A
Training Phrases	“Dashboard” “Open Dashboard” “Go to my Dashboard”
Action and Parameters	N/A
Responses	“Webhook failed for intent: Dashboard”
Fulfilment	(via webhook from index.js) [START df_js_dashboard_card]

Intent Name	Doctor
Context	N/A
Events	N/A
Training Phrases	“Doctor” “Open Doctor” “Go to my Doctor”
Action and Parameters	N/A
Responses	“Webhook failed for intent: Doctor”
Fulfilment	(via webhook from index.js) [START df_js_doctor_card]

Intent Name	help
Context	N/A
Events	N/A
Training Phrases	“help” “I need help” “Help me”
Action and Parameters	N/A
Responses	“<speak> I am your voice activated health concierge <break time= "800ms" />. The following voice activated commands are available. <break time= "700ms" /> Dashboard <break time= "700ms" />, Medications <break time= "700ms" />, Doctor <break time= "700ms" />, Close and <break time= "1s" />, Help <break time= 700ms" />. You can ask me how to use any of the commands by saying "How do I use" <break time= "700ms" /> plus the command name. <break time= "700ms" /> For example: "How do I use Dashboard". </speak>”
Fulfilment	N/A

Intent Name	How do I use close?
Context	N/A
Events	N/A
Training Phrases	“How do I use close?”
Action and Parameters	N/A
Responses	The close command will terminate your session with Health Concierge. Just say "Close" and I will be terminated.
Fulfilment	N/A

Intent Name	How do I use Dashboard?
Context	N/A
Events	N/A
Training Phrases	“How do I use Dashboard?”
Action and Parameters	N/A
Responses	The Dashboard is your main health information page. From the Dashboard you can see your health information, active-wear information, medical records, medications, and doctor. Access your dashboard by saying: "Dashboard" or "Open Dashboard"
Fulfilment	N/A

Intent Name	How do I use Doctor?
Context	N/A
Events	N/A
Training Phrases	“How do I use Doctor?”
Action and Parameters	N/A
Responses	The Doctor section displays your doctor information. From the Doctor section you can see your doctor's information, schedule appointments, and start a telehealth videochat with your doctor. Access the Doctor section by saying: "Doctor", "Open Doctor", or "Let me talk to my Doctor"
Fulfilment	N/A

Intent Name	How do I use help?
Context	N/A

Events	N/A
Training Phrases	"How do I use help?"
Action and Parameters	N/A
Responses	The help command will display your options for using me.
Fulfilment	N/A

Intent Name	How do I use Medications?
Context	N/A
Events	N/A
Training Phrases	"How do I use Medications?"
Action and Parameters	N/A
Responses	The Medications section displays your medications. From the medications section you can see your medication information and even order refills. Access medications by saying: "Medications" or "Open Medications"
Fulfilment	N/A

Intent Name	Medications
Context	N/A
Events	N/A
Training Phrases	"Medications" "Go to my Medications" "Open Medications"
Action and Parameters	N/A
Responses	Webhook failed for intent: Medications
Fulfilment	(via webhook from index.js) [START df_js_medications_card]

Intent Name	SignIn
Context	Signin-followup
Events	N/A
Training Phrases	"Log in" "Let me log in" "Sign me in"

	“Start signin” “Sign in”
Action and Parameters	N/A
Responses	Webhook failed for intent: SignIn
Fulfilment	(via webhook from index.js)

Intent Name	SignIn Helper
Context	Signin-followup
Events	Google Assistant Sign In
Training Phrases	N/A
Action and Parameters	N/A
Responses	Webhook failed for intent: SignIn Handler
Fulfilment	(via webhook from index.js)

Appendix B. Javascript webhooks for web application redirection can be found in our github repository.

Appendix C. Google Assistant System Design Documentation.

This documentation is for informational purposes of understanding the system design of how Google Assistant Dialogflow and the Google Assistant NLP toolkit is used in conjunction with our Health Concierge web application.

Google Dialogflow is a platform which allows components called “Actions” that can be integrated into an interactive session with Google Assistant supported devices inclusive of smartphones and smart speakers (Google Home).

Within the Dialogflow platform there are separate parts of the system design that allow for an interactive conversational experience that can result in dynamic web application interaction and user interaction and input.

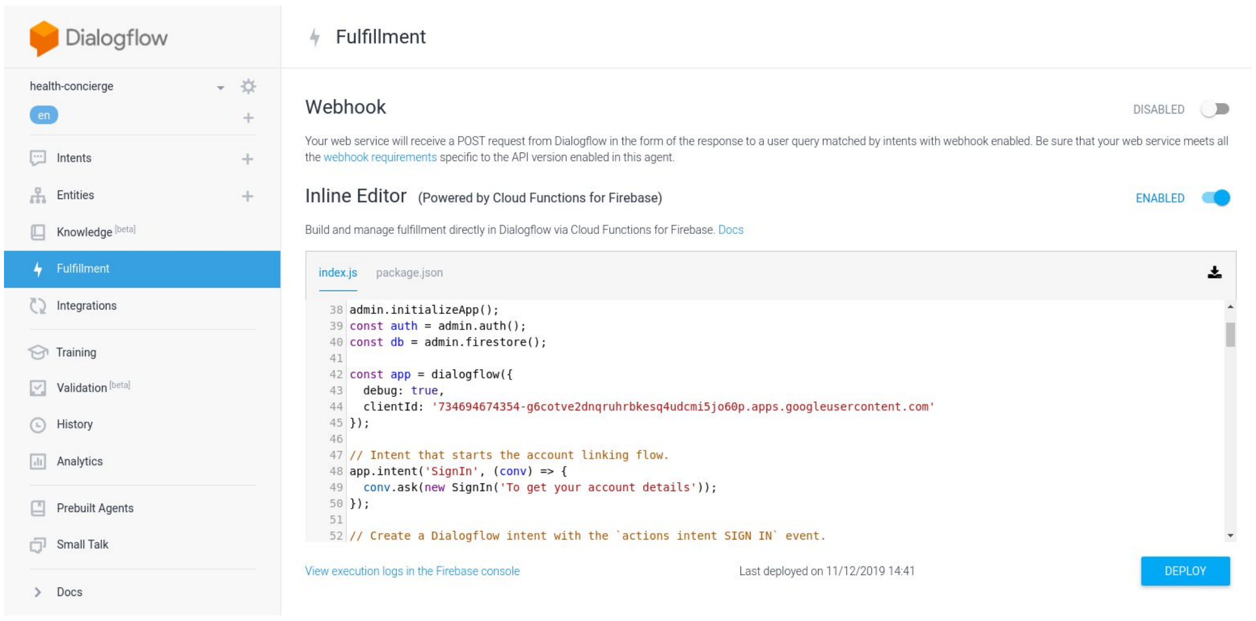
Intents within Dialogflow represent a unique identifier that we have specified to be hooks to open specific parts of our health concierge web application. Intents can be also defined as “Conversational Actions”. In this context, the Intents direct the generalized action process to only open up parts of the web application that we will be hosting the functions of our health concierge web app.

The health concierge web app will be providing the services as defined in our initial and since updated writeup. These services include a dashboard page, input page, telemedicine services, and pharmaceutical delivery page.

The interaction shall be defined between three parties: The USER, the ASSISTANT, and the CONCIERGE.

The USER will initiate a Google Assistant session and request the opening of the CONCIERGE application built within the ASSISTANT. The ASSISTANT will request input from the user via voice command or text command. The USER, on first interaction can request a list of services he/she can access. The ASSISTANT will list options of: dashboard, input, telemedicine, and pharmaceutical delivery. The USER, on subsequent interaction can immediately request CONCIERGE interaction without going through the list prompts. An example interactive action the USER can take is to request access to the dashboard of the CONCIERGE. The USER will speak a variant of “please open my dashboard”. The NLP system of Dialogflow that we trained will recognize a range of variants from just the word “dashboard” to “open my dashboard please”. Upon recognition of request from the USER, the CONCIERGE will open up the dashboard page of the health concierge web application. This is done by a webhook and opening of a browser page from the user’s phone.

We thought it necessary to detail this system design and workflow because the logic of the system is built into the Dialogflow platform. Screenshots have been added below as an example of the Dialogflow platform we used to build out these processes.



Webhook DISABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

Inline Editor (Powered by Cloud Functions for Firebase) ENABLED

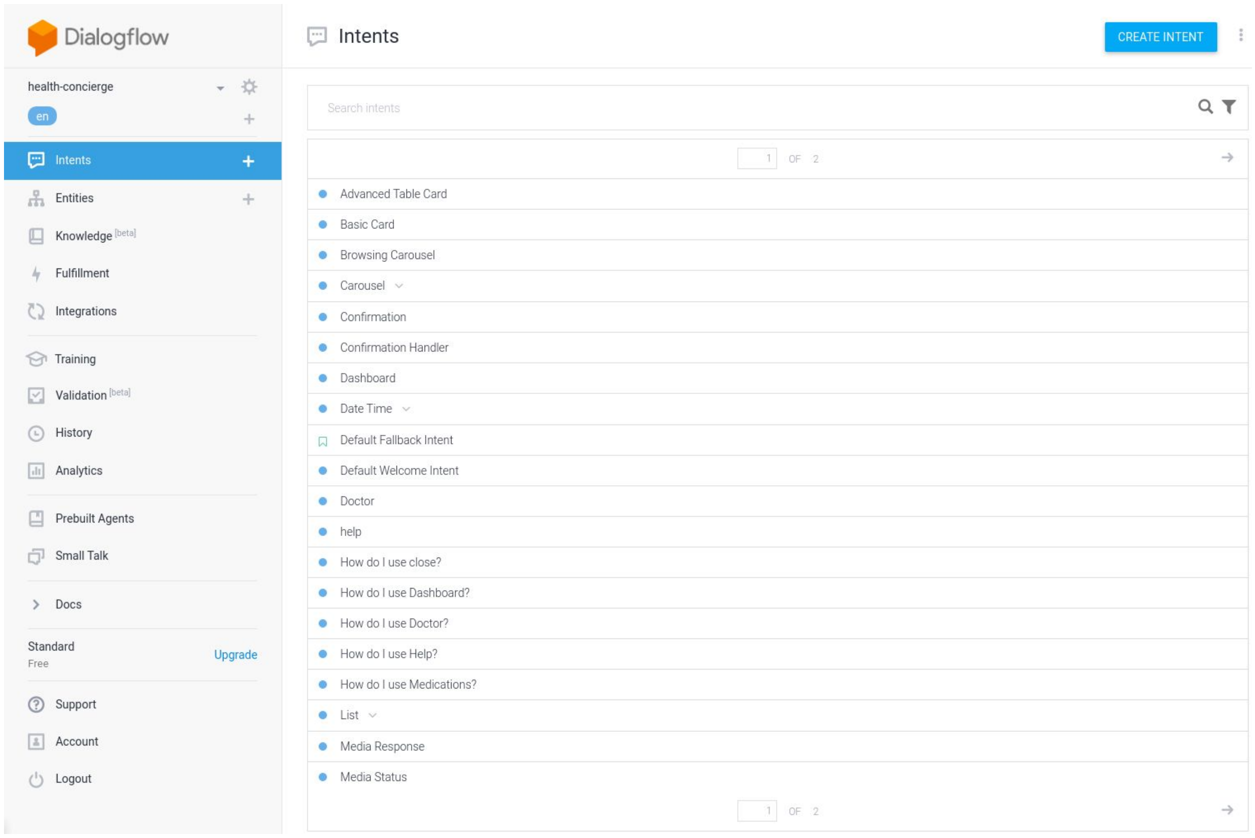
Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs

```
admin.initializeApp();
const auth = admin.auth();
const db = admin.firestore();

const app = dialogflow({
  debug: true,
  clientId: '734694674354-g6cotve2dnqruhrbkesq4udcmi5jo60p.apps.googleusercontent.com'
});

// Intent that starts the account linking flow.
app.intent('SignIn', (conv) => {
  conv.ask(new SignIn('To get your account details'));
});

// Create a Dialogflow intent with the `actions intent SIGN IN` event.
```

View execution logs in the Firebase console

Last deployed on 11/12/2019 14:41

Webhook and custom fulfillment console. Which we have customized to fulfill our health concierge web application redirections. The full webhook code can be found on our github repository.



- Advanced Table Card
- Basic Card
- Browsing Carousel
- Carousel
- Confirmation
- Confirmation Handler
- Dashboard
- Date Time
- Default Fallback Intent
- Default Welcome Intent
- Doctor
- help
- How do I use close?
- How do I use Dashboard?
- How do I use Doctor?
- How do I use Help?
- How do I use Medications?
- List
- Media Response
- Media Status

Intents. The three main ‘hooks’ to catch user language from the Dialogflow NLP engine are defined and customized here, along with the welcome message.