Health Concierge Milestone 2 Writeup

Daniel Hwang, Ride Bu, Nancy Zhang

Working beta - Demonstration

Walkthrough prototype with Gopika: done

Working beta - Google Assistant

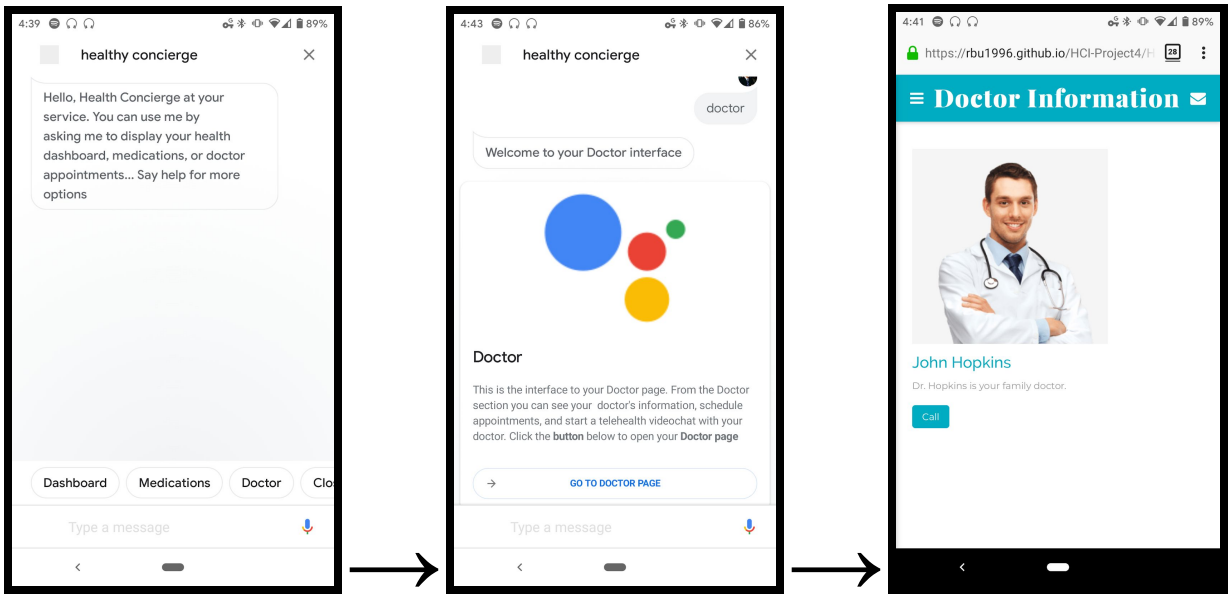
Done

Working beta - Web Application Flow

Done

System Design Components

Google Assistant, Web Application



Google assistant → Google Assistant → Web app

Web Application

The web application side of our overall application represents the database and primary health information pages. 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.

Pages

The pages we created for the application included: about-us, appointments, blog, call-doctor, cart, checkout, contact, doctor-information, doctor, health-monitor, home, login, register.

Google Assistant Developer API

The main application logic for the Google Assistant aspect was controlled via the Dialogflow components of the Google Assistant Developers console. In addition to additional javascript code that were called via webhooks, we used specialized components called Intents that handled the voice and text interaction from a particular user interacting with the Google Assistant app we created, **health-concierge**. 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 Intents. Below is a description of the intent parameters and the intents we created to handle the logic.

Design process

We iterated primarily on the human accessibility aspect of the Google Assistant platform. Although the Google Assistant offers text interaction, we specifically focused on voice commands and the conversational flow of the conversation. This included the help of the natural language processing modules that the Google Assistant Dialogflow offers.

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

```
// Copyright 2019, Google, Inc.
// Licensed under the Apache License, Version 2.0 (the 'License');
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//    http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an 'AS IS' BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

'use strict';

const {
  dialogflow,
  Permission,
  DateTime,
  Place,
  SimpleResponse,
  BasicCard,
  Button,
  Image,
  BrowseCarousel,
  BrowseCarouselItem,
  Suggestions,
  LinkOutSuggestion,
  MediaObject,
  Table,
```

```
  List,
  Carousel,
  SignIn,
  Confirmation,
} = require('actions-on-google');
const admin = require('firebase-admin');
const functions = require('firebase-functions');
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.
app.intent('SignIn Helper', (conv, params, signin) => {
  if (signin.status === 'OK') {
    const payload = conv.user.profile.payload;
    conv.ask(`I got your account details, ${payload.name}. What do you want
to do next?`);
  } else {
    conv.ask(`I won't be able to save your data, but what do you want to do
next?`);
  }
});


// [START df_js_permission_reason]
app.intent('Permission', (conv) => {
  const permissions = ['NAME'];
  let context = 'To address you by name';
  // Location permissions only work for verified users
  // https://developers.google.com/actions/assistant/guest-users
  if (conv.user.verification === 'VERIFIED') {
```

```
    // Could use DEVICE_COARSE_LOCATION instead for city, zip code
    permissions.push('DEVICE_PRECISE_LOCATION');
    context += ' and know your location';
  }
  const options = {
    context,
    permissions,
  };
  conv.ask(new Permission(options));
});
// [END df_js_permission_reason]

// [START df_js_permission_accepted]
app.intent('Permission Handler', (conv, params, confirmationGranted) => {
  // Also, can access latitude and longitude
  // const { latitude, longitude } = location.coordinates;
  const {location} = conv.device;
  const {name} = conv.user;
  if (confirmationGranted && name && location) {
    conv.ask(`Okay ${name.display}, I see you're at ` +
      `${location.formattedAddress}`);
  } else {
    conv.ask(`Looks like I can't get your information.`);
  }
  conv.ask(`Would you like to try another helper?`);
  conv.ask(new Suggestions([
    'Confirmation',
    'DateTime',
    'Place',
  ]));
});
// [END df_js_permission_accepted]

// [START df_js_place_reason]
app.intent('Place', (conv) => {
  const options = {
    context: 'To find a location',
    prompt: 'Where would you like to go?',
  };
  conv.ask(new Place(options));
});
// [END df_js_place_reason]
```

```
// [START df_js_place_accepted]
app.intent('Place Handler', (conv, params, place) => {
  if (!place) {
    conv.ask(`Sorry, I couldn't get a location from you.`);
  } else {
    // Place also contains formattedAddress and coordinates
    const {name} = place;
    conv.ask(`${name} sounds like a great place to go!`);
  }
  conv.ask('Would you like to try another helper?');
  conv.ask(new Suggestions([
    'Confirmation',
    'DateTime',
    'Permission',
  ]));
});
// [END df_js_place_accepted]

// [START df_js_confirmation_reason]
app.intent('Confirmation', (conv) => {
  conv.ask(new Confirmation('Can you confirm?'));
});
// [END df_js_confirmation_reason]

// [START df_js_confirmation_accepted]
app.intent('Confirmation Handler', (conv, params, confirmationGranted) => {
  conv.ask(confirmationGranted
    ? 'Thank you for confirming'
    : 'No problem, you have not confirmed');
  conv.ask('Would you like to try another helper?');
  conv.ask(new Suggestions([
    'DateTime',
    'Permission',
    'Place',
  ]));
});
// [END df_js_confirmation_accepted]

// [START df_js_simple_response]
app.intent('Simple Response', (conv) => {
  conv.ask(new SimpleResponse({
```

```
    speech: `Here's an example of a simple response. ` +
      `Which type of response would you like to see next?`,
    text: `Here's a simple response. ` +
      `Which response would you like to see next?`,
  }));
});
// [END df_js_simple_response]

// [START df_js_ssml_demo]
app.intent('SSML', (conv) => {
  conv.ask(`<speak>` +
    `Here are <say-as interpet-as="characters">SSML</say-as> examples.` +
    `Here is a buzzing fly ` +
    `<audio
src="https://actions.google.com/sounds/v1/animals/buzzing_fly.ogg"></audio>
` +
    `and here's a short pause <break time="800ms"/>` +
    `</speak>`);
  conv.ask('Which response would you like to see next?');
});
// [END df_js_ssml_demo]

// [START df_js_basic_card]
app.intent('Basic Card', (conv) => {
  if (!conv.screen) {
    conv.ask('Sorry, try this on a screen device or select the ' +
      'phone surface in the simulator.');
    conv.ask('Which response would you like to see next?');
    return;
  }

  conv.ask(`Here's an example of a basic card.`);
  conv.ask(new BasicCard({
    text: `This is a basic card.  Text in a basic card can include "quotes"
and
    most other unicode characters including emojis.  Basic cards also
support
    some markdown formatting like *emphasis* or _italics_, **strong** or
    __bold__, and ***bold itallic*** or ___strong emphasis___ as well as
other
    things like line  \nbreaks`, // Note the two spaces before '\n'
required for
```

```
                                // a line break to be rendered in the
card.
    subtitle: 'This is a subtitle',
    title: 'Title: this is a title',
    buttons: new Button({
      title: 'This is a button',
      url: 'https://assistant.google.com/',
    }),
    image: new Image({
      url:
'https://storage.googleapis.com/actionsresources/logo_assistant_2x_64dp.png
',
      alt: 'Image alternate text',
    }),
    display: 'CROPPED',
  }));
  conv.ask('Which response would you like to see next?');
});
// [END df_js_basic_card]


// [START df_js_showuserprofile_card]
app.intent("Show User Profile", conv => {
  const payload = conv.user.profile.payload;
  if (payload) {

    const userId = payload.aud;
    const name = payload.name;
    const givenName = payload.given_name;
    const familyName = payload.family_name;
    const email = payload.email;
    const emailVerified = payload.email_verified;
    const picture = payload.picture;

    conv.ask("This is your profile information.");
    conv.ask(new BasicCard({
      text: `ID:${userId}
      Name:${name}
      Given name:${givenName}
      Family name:${familyName}
      Email:${email}
      Email verified:${emailVerified}`,
```

```
      image: new Image({
        url: picture,
        alt: "Profile Image"
      })
    }));
  } else {
    conv.ask("Not signed in yet.");
    //conv.ask(new Suggestion("want to sign in"));
  }
});
// [END df_js_showuserprofile_card]

// [START df_js_personalinformation_card]
app.intent('Personal Information', (conv) => {
  if (!conv.screen) {
    conv.ask('Sorry, try this on a screen device or select the ' +
      'phone surface in the simulator.');
    conv.ask('Which response would you like to see next?');
    return;
  }

  conv.ask(`Here is your Personal Information`);
  conv.ask(new BasicCard({
    //text: `Your userId is: ${app.getUser().userId}`,
    title: 'Personal Information',
    image: new Image({
      url:
'https://storage.googleapis.com/actionsresources/logo_assistant_2x_64dp.png
',
      alt: 'Image alternate text',
    }),
    display: 'CROPPED',
  }));
});
// [END df_js_personalinformation_card]

// [START df_js_dashboard_card]
app.intent('Dashboard', (conv) => {
  if (!conv.screen) {
    conv.ask('Sorry, try this on a screen device or select the ' +
      'phone surface in the simulator.');
    conv.ask('Which response would you like to see next?');
```

```
    return;
  }
  conv.ask(`Welcome to your Dashboard interface`);
  conv.ask(new BasicCard({
    text: `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__ `,
    title: 'Dashboard',
    buttons: new Button({
      title: 'Go to Dashboard',
      url:
'https://rbu1996.github.io/HCI-Project4/HCI-P4-website/health-monitor.html'
,
    }),
    image: new Image({
      url:
'https://storage.googleapis.com/actionsresources/logo_assistant_2x_64dp.png
',
      alt: 'Image alternate text',
    }),
    display: 'CROPPED',
  }));
});
// [END df_js_dashboard_card]


// [START df_js_medications_card]
app.intent('Medications', (conv) => {
  if (!conv.screen) {
    conv.ask('Sorry, try this on a screen device or select the ' +
      'phone surface in the simulator.');
    conv.ask('Which response would you like to see next?');
    return;
  }
  conv.ask(`Welcome to your Medications interface`);
  conv.ask(new BasicCard({
    text: `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__ `,
    title: 'Medications',
    buttons: new Button({
      title: 'Go to Medications',
      url:
'https://rbu1996.github.io/HCI-Project4/HCI-P4-website/home-shop.html',
    }),
    image: new Image({
      url:
'https://storage.googleapis.com/actionsresources/logo_assistant_2x_64dp.png
',
      alt: 'Image alternate text',
    }),
    display: 'CROPPED',
  }));
});
// [END df_js_medications_card]


// [START df_js_doctor_card]
app.intent('Doctor', (conv) => {
  if (!conv.screen) {
    conv.ask('Sorry, try this on a screen device or select the ' +
      'phone surface in the simulator.');
    conv.ask('Which response would you like to see next?');
    return;
  }
  conv.ask(`Welcome to your Doctor interface`);
  conv.ask(new BasicCard({
    text: `This is the interface to your Doctor page. From the Doctor
section you can see your
    doctor's information, schedule appointments, and start a telehealth
videochat with your doctor.
    Click the __button__ below to open your __Doctor page__ `,
    title: 'Doctor',
    buttons: new Button({
      title: 'Go to Doctor Page',
      url:
'https://rbu1996.github.io/HCI-Project4/HCI-P4-website/doctor.html',
    }),
```

```
    image: new Image({
      url:
'https://storage.googleapis.com/actionsresources/logo_assistant_2x_64dp.png
',
      alt: 'Image alternate text',
    }),
    display: 'CROPPED',
  }));
});
// [END df_js_doctor_card]

// [START df_js_suggestion_chips]
app.intent('Suggestion Chips', (conv) => {
  if (!conv.screen) {
    conv.ask('Chips can be demonstrated on screen devices.');
    conv.ask('Which response would you like to see next?');
    return;
  }

  conv.ask('These are suggestion chips.');
  conv.ask(new Suggestions('Suggestion 1'));
  conv.ask(new Suggestions(['Suggestion 2', 'Suggestion 3']));
  conv.ask(new LinkOutSuggestion({
    name: 'Suggestion Link',
    url: 'https://assistant.google.com/',
  }));
  conv.ask('Which type of response would you like to see next?'); ;
});
// [END df_js_suggestion_chips]

exports.dialogflowFirebaseFulfillment = functions.https.onRequest(app);
```