

MICHIGAN STATE

U N I V E R S I T Y

Project Plan

Self-Service Support Chatbot for Google Cloud

The Capstone Experience

Team Google

Brandon Hu

Emma Hettinger

Keerthana Byreddy

Kyle Kraski

Tyler Weisner

Department of Computer Science and Engineering

Michigan State University

Fall 2020



*From Students...
...to Professionals*

Functional Specifications

- Chatbot to suggest solutions to GCP users' problems
- Finds and links relevant documentation
- Helps customers troubleshoot for common problems
- Collects relevant error information
- Refers customer to human support if the chatbot is unable to help
- Has a dashboard of analytics for chatbot performance

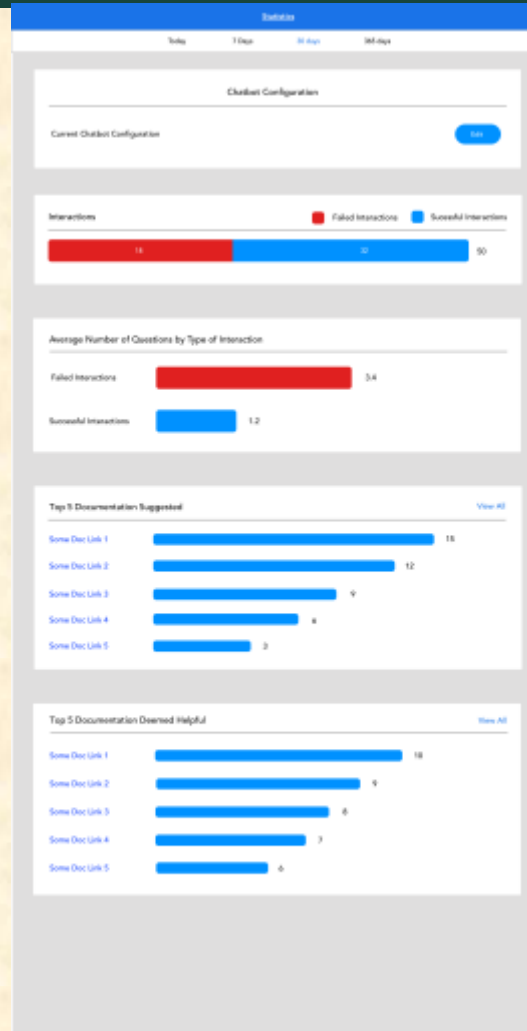


Design Specifications

- The chatbot is using Dialogflow's standard UI
- The dashboard shows analytics about bot performance:
 - Bot's success rate
 - Number of questions before a solution
- Statistics about documentation being accessed
- Ability to configure
 - Add, delete, or edit documentation being scraped



Screen Mockup: Chatbot Statistics



Screen Mockup: Chatbot Configuration

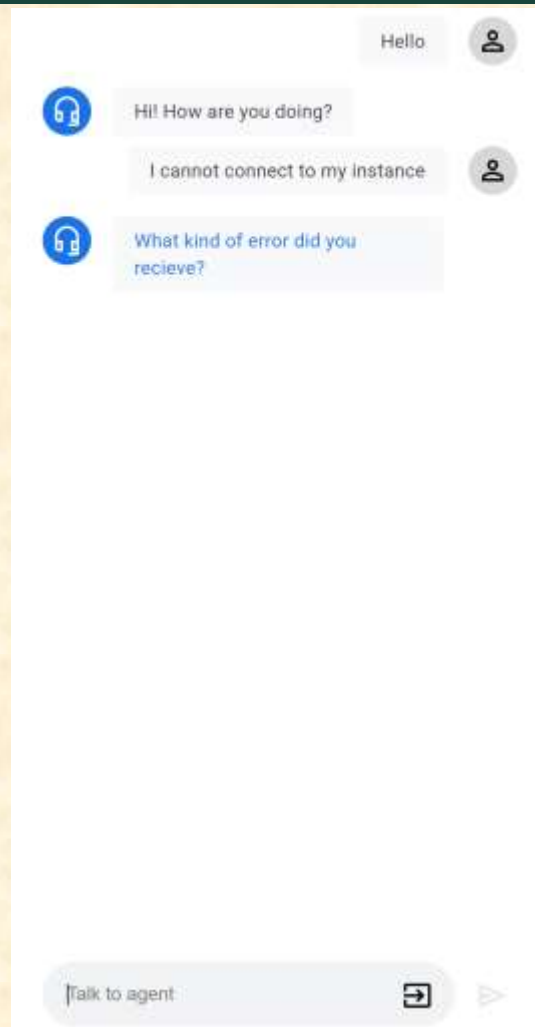
Documentation

Some Document Doc url	Enabled <input checked="" type="checkbox"/> Delete
Some Document Doc url	Enabled <input checked="" type="checkbox"/> Delete
Some Document Doc url	Enabled <input checked="" type="checkbox"/> Delete
Some Document Doc url	Enabled <input type="checkbox"/> Delete

[Save](#) [Add](#) [Cancel](#)



Screen Mockup: Chatbot Interaction

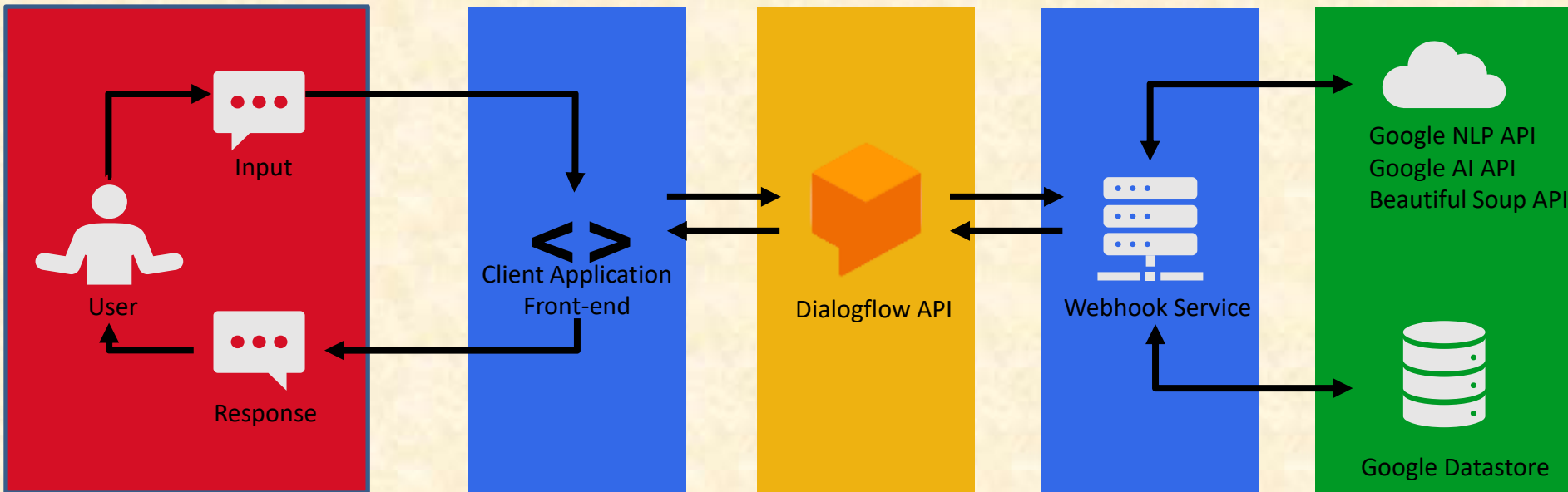


Technical Specifications

- The user will interact with Dialogflow's chatbot interface and ask it a question.
- Dialogflow passes text inputs to webhook which will pass it onto NLP
- NLP is used to find relevant documentation in Datastore
- Dialogflow provides an answer to the user's question
- At the end of the session, interaction data with the user is stored in Datastore



System Architecture



System Components

- Software Platforms / Technologies
 - Google Cloud Platform
 - Google Dialogflow CX
 - Google Datastore
 - GCP AI platform
 - GCP Natural Language Processing API
 - React Framework



Risks

- Working on GCP simultaneously
 - Could break things if multiple people try to use something at once
 - Mitigation: discuss when doing any GCP deployments, assign one person to handle DevOps
- Might miss stuff in support documents
 - GCP has extensive support documents, might not access important information
 - Mitigation: find a comprehensive way to scrape vast data, extensive testing with a variety of samples



Risks

- Doing decision trees by hand
 - We need to make a lot of decision trees
 - Mitigation: either find a way to automate or limit the number that we create
- Running code within client's GCP environment for security
 - Have no idea how to do this
 - Mitigation: Create a script that users can run within their GCP platform



Questions?

?

?

?

?

?

?

?

?

?

