HEALTHCARE EQUIPMENT ADVISOR CHATBOT

Harsh Sharma and Prithviraj Khelkar

May 5, 2023

Boston University

Graduate School of Arts and Sciences

BOSTON UNIVERSITY

HEALTHCARE EQUIPMENT ADVISOR CHATBOT

Harsh Sharma and Prithviraj Khelkar

Boston University
Graduate School of Arts and Sciences
Boston, MA 02215
`www.bu.edu/grs`

May 5, 2023

Contents

1 Introduction

The healthcare industry is continually advancing, and there is a growing need for innovative solutions to enhance the quality of patient care. One area where technology can significantly impact healthcare is through chatbots. Chatbots are becoming increasingly popular in the healthcare industry because they provide quick and personalized responses to patient inquiries, offer medical advice, and even assist with managing chronic diseases.

As consultants in XYZ Corporation, our team has been tasked with developing a chatbot for ACME Corporation, a leading medical equipment provider. ACME has subsidiaries in different countries and wants to expand into the conversational assistant space to provide better access to younger demographics and expand its services and market. The chatbot we are developing will serve as a Medical Equipment Advisor, providing information and advice on medical equipment, taking orders, and offering technical support to patients and healthcare providers. We aim to provide a detailed Statement of Work (SOW) outlining the design choices, data preparation, and evaluation criteria required to build a first Proof of Concept for testing in a few select markets. By leveraging the power of conversational assistants, ACME can enhance its customer experience and improve the quality of patient care in the healthcare industry.

2 Purpose and Objective

The primary objective of this project is to develop a single language chatbot specifically designed to cater to the younger generation, aiming to streamline and simplify the process of ordering items on ACME Corporation's website. The chatbot will provide a user-friendly interface and an engaging experience while minimizing errors throughout the ordering process.

The purpose of this document is to outline the strategy and steps necessary for creating such a chatbot, focusing on its ability to enhance the user experience and ensure accuracy during the entire interaction. By detailing the development process, this document serves as a comprehensive guide for successfully implementing an efficient and reliable chatbot that meets the project's objectives.

3 Tasks

This section provides a detailed overview of the tasks involved in developing the chatbot. It covers the stories supported, data preparation, required APIs, handling digressions, validation and testing, continuous upgrading strategy, and client-server architecture.

3.1 Data Preparation

1. Preprocessing the user reviews: Before using the user reviews to train the chatbot, we need to clean the data by removing any irrelevant or redundant information, correcting spelling or grammatical errors, and standardizing the format of the reviews.

2. Labeling the data: After preprocessing the user reviews, we need to sieve intents and entities for the chatbot to be able to recognize. The chatbot should be able to extract from the reviews data such as the type of equipment and the features the user was looking for.

3. Defining the Rasa pipeline:

   (a) SpacyNLP

   (b) SpacyTokenizer

   (c) SpacyFeaturizer

   (d) RegexFeaturizer

   (e) CRFEntityExtractor

   (f) EntitySynonymMapper

   (g) DIETClassifier

   (h) EntityExtractorFallback

4. Fine-tuning the pipeline: After defining the pipeline, it would be fine-tuned by experimenting with different hyperparameters and configurations to improve the chatbot's performance on predefined testing set testing set.

3.2 Stories Supported

This sub-section outlines the different stories or scenarios that the chatbot is designed to handle. It includes the various intents and entities that the chatbot can recognize and respond to.

3.2.1 Buy or Rent Items

This story is triggered when the user wants to buy or rent items sold by ACME corporation. The chatbot asks the user for the name of the item they are interested in buying or renting. The chatbot extracts the entity "User Input Item" based on the user's response.

Next, if the user has not provided whether they want to buy or rent the item, the chatbot asks the user. The chatbot extracts the entity "Buying or Ordering" from the user's response. If the user chooses to rent the item, the chatbot asks for the rental period and calculates the rental cost based on the item's price and rental period.

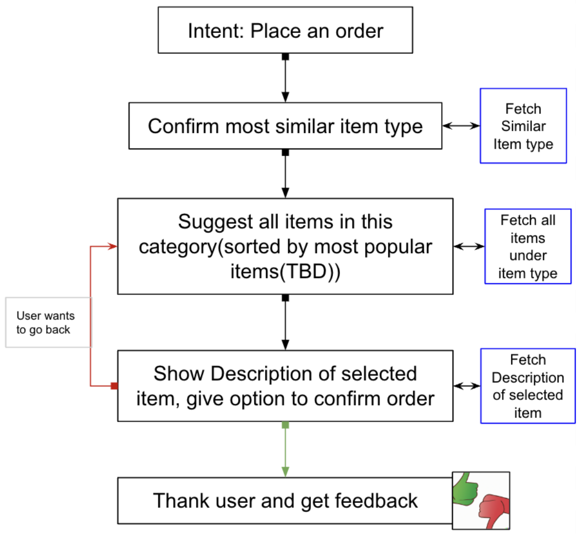

Intent: Place an order
Confirm most similar item type
Fetch Similar Item type
Suggest all items in this category(sorted by most popular items(TBD))
Fetch all items under item type
User wants to go back
Show Description of selected item, give option to confirm order
Fetch Description of selected item
Thank user and get feedback

Figure 1: Story to buy rent item

If the user chooses to buy the item, the chatbot asks for the product category and name to confirm the product details. The chatbot also extracts the "Buying or Ordering" entity to confirm the user's intent. Finally, the user is asked to complete the payment before placing the order; the chatbot sends the payment link to the user to confirm the payment before placing the order and sends the confirmation number to the user.

The conversation Figure 2 is an example of how the chatbot interacts with the user based on the story in Figure 1. In this example, the user wants to buy a thermometer. The chatbot confirms the category of the product and lists the options available. The user clicks more descriptions of the item before placing the order. The chatbot provides the user with a confirmation number.

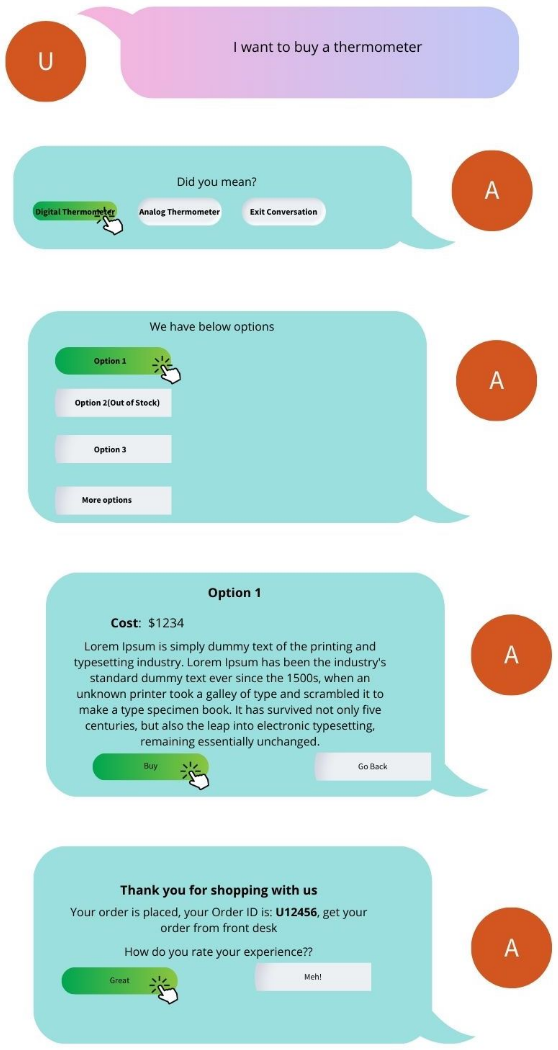
Intent: Buy or Rent Equipment

Data Requirements:

1. Training
   - Sample Intent: Basic Intents created using Natural Language generation tools for this intent
2. Live chatbot
   - Item availability(API)

Entities:

- User Input Item: Entity extracted from user input



I want to buy a thermometer

Did you mean?

We have below options

Option 1

Cost: $1234

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

Thank you for shopping with us

Your order is placed, your Order ID is: **U12456**, get your order from front desk

How do you rate your experience??

Figure 2: Sample conversation for the story in Figure 1

- Buying or Ordering: Whether the user wants to buy or rent the item

Slots:

- User Input Item: Entity extracted from user input
- Product Category: The product category selected by the user based on the closest matches of available categories
- Product Name: Product selected by the user to buy or rent
- Buying or Ordering: Whether the user wants to buy or rent the item

3.2.2 Recommend equipment

This story would suggest items based on the user's input. The chatbot first asks the user the type of equipment they are interested in. Based on the user's response, the chatbot extracts keywords.

After generating the search results, the chatbot suggests the best-rated items that fulfill the user's requirements. The chatbot furnishes further details regarding the recommended products, such as their name, image, and price. The user can review the items individually, and once they have made their choice, they can purchase them. The chatbot then shares a payment link with the user to finalize the transaction, and after the payment is confirmed, the order is placed. The chatbot then sends a confirmation number to the user.

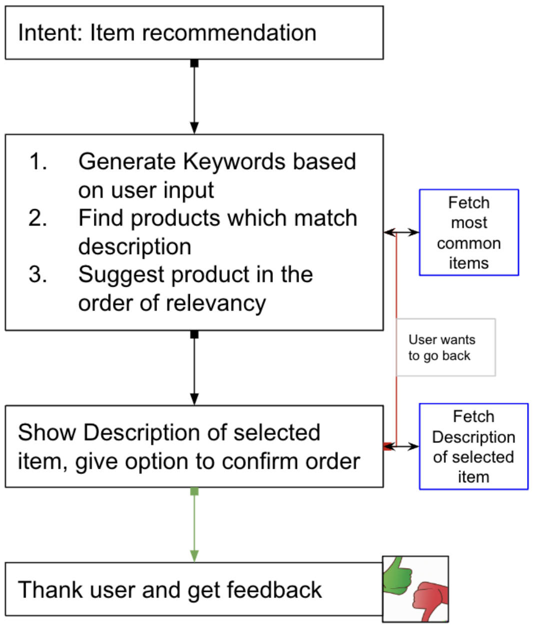

Intent: Item recommendation
1. Generate Keywords based on user input
2. Find products which match description
3. Suggest product in the order of relevancy
Fetch most common items
User wants to go back
Show Description of selected item, give option to confirm order
Fetch Description of selected item
Thank user and get feedback

Figure 3: Story to Recommend Items based on input

The conversation provided in Figure 4 is an example of how the chatbot interacts with the user based on the story in Figure 3. In this example, the user wants to buy a good walking stick for her grandfather. The chatbot recommends the top-rated cameras based on their rating and user feedback and provides additional information about each recommended item. The user selects one of the recommended items and proceeds to purchase it.

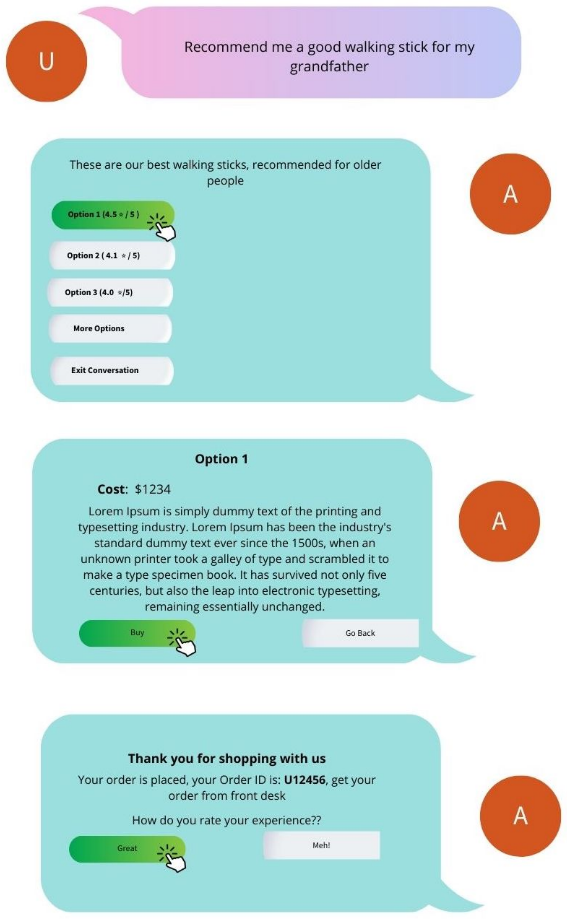
Intent: Item Recommendation

Data Requirements:

1. Training
   - Sample Intent: Basic Intents created using Natural Language generation tools for this intent
   - Data on Search terms used in ACME's website
   - Description of all products sold at ACME
   - All user product reviews for items sold at ACME
2. Live chatbot
   - Item Availability(API)
   - Top recommended items based on keywords(API)

Entities:

- Buying or Ordering: Whether the user wants to buy or rent the item



Recommend me a good walking stick for my grandfather

These are our best walking sticks, recommended for older people

Option 1

Cost: $1234

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

Thank you for shopping with us

Your order is placed, your Order ID is: **U12456**, get your order from front desk

How do you rate your experience??

Figure 4: Sample Conversation for the story in Figure 3

Slots:

- Product Category: The product category selected by the user based on the closest matches of available categories
- Product Name: Product selected by the user to buy or rent
- Buying or Ordering: Whether the user wants to buy or rent the item

3.2.3 Technical Support

This story is triggered when a user requests technical support for an item they purchased from the company. The chatbot first asks the user to describe their issue with the item. The chatbot first confirms the recent order name which has the issue. The chatbot extracts the "Product name" based on the user's response.

Next, the chatbot asks the user to provide the issue in detail; here, the user can describe the error codes or messages they received and, in detail, whatever they think

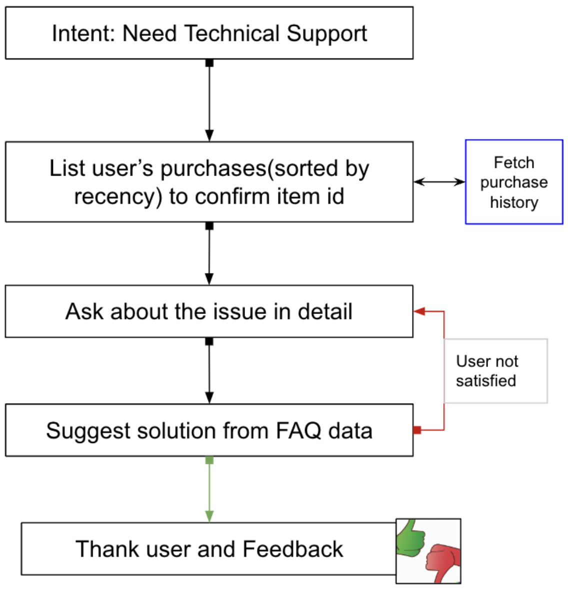

Intent: Need Technical Support
List user's purchases(sorted by recency) to confirm item id
Fetch purchase history
Ask about the issue in detail
User not satisfied
Suggest solution from FAQ data
Thank user and Feedback

Figure 5: Story to provide technical support for items

is the issue. The chatbot extracts this information and uses it to diagnose the issue.

Once the issue has been identified, the chatbot provides the user with a list of potential solutions extracted from the products' FAQ data. The chatbot also asks the user if they need further assistance or if the issue has been resolved.

The conversation in Figure 6 is an example of how the chatbot interacts with the user based on the story in Figure 5. In this example, the user faces an issue with their CPAP machine. The chatbot asks for the issue in detail and uses it to diagnose it. The chatbot lists potential solutions, and the user tries one. The issue is resolved, and the chatbot asks the user if they need further assistance. The user ends the conversation since the issue has been resolved.

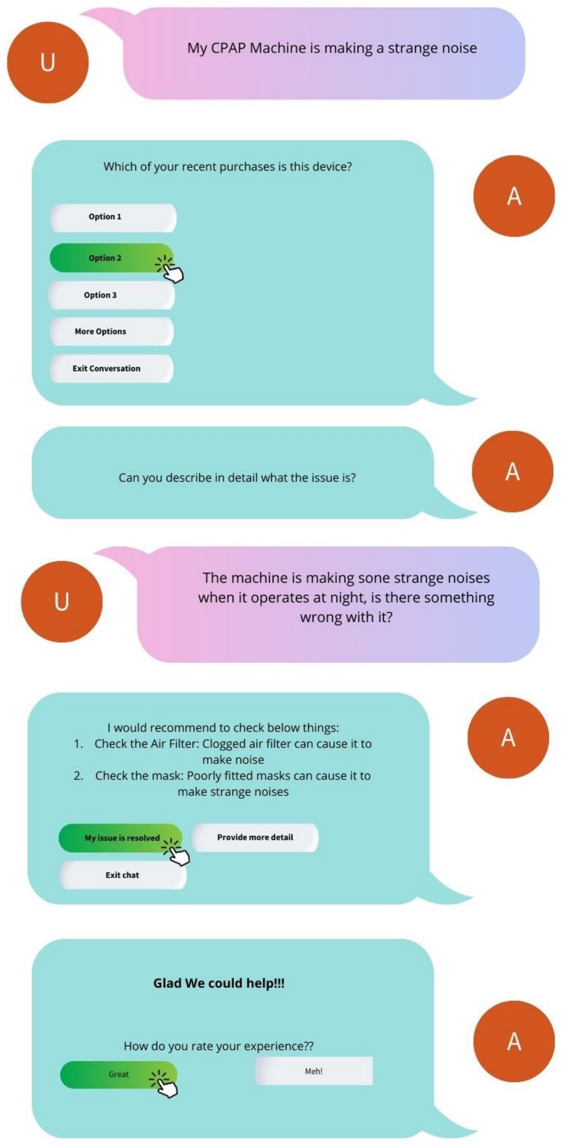
Intent: Get Technical Support

Data Requirements:

1. Training
   - Sample Intent: Basic Intents created using Natural Language generation tools for this intent
   - FAQ data for each product
2. Live chatbot
   - Availability of items(API)

Entities:

- User Input Item: Extracted from user input


U: My CPAP Machine is making a strange noise

A: Which of your recent purchases is this device?

A: Can you describe in detail what the issue is?

U: The machine is making sone strange noises when it operates at night, is there something wrong with it?

A: I would recommend to check below things:
1. Check the Air Filter: Clogged air filter can cause it to make noise
2. Check the mask: Poorly fitted masks can cause it to make strange noises

A: **Glad We could help!!!**
How do you rate your experience??

Figure 6: Sample Conversation for the story in Figure 5

Slots:

- Order number: Order number selected based on user input
- Resolved: If the issue is resolved; it will be used to exit the form; if not, it will loop until the user asks to exit

3.2.4 Currently issued items

This story is triggered when a user wants to view their currently issued items from the company. The chatbot displays the list of items to the user and additional information

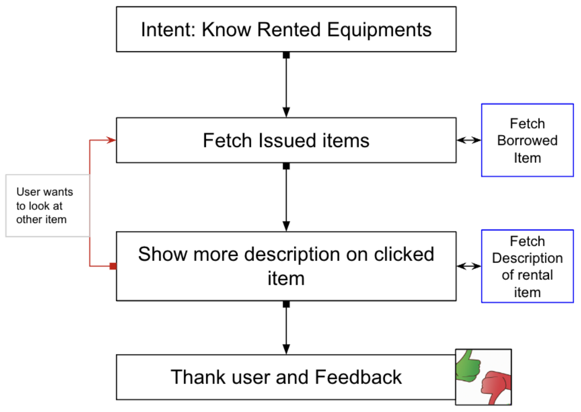

Intent: Know Rented Equipments
Fetch Issued items
Fetch Borrowed Item
User wants to look at other item
Show more description on clicked item
Fetch Description of rental item
Thank user and Feedback

Figure 7: Story to show currently issued items

such as the rental period and cost.

If the user wants to return any of the items, the chatbot provides instructions on how to do so. The chatbot also asks the user if they need further assistance or have any additional questions.

The conversation in Figure 8 is an example of how the chatbot interacts with the user based on the story in Figure 7. In this example, the user wants to view their currently rented items. The chatbot displays the list of items to the user and additional information such as the rental period and cost. The user confirms that they do not need further assistance, and the conversation ends.

Intent: View currently issued items

Data Requirements:

1. Training
   - Sample Intent: Basic Intents created using Natural Language generation tools for this intent
2. Live chatbot
   - Currently Issued items for the user(API)

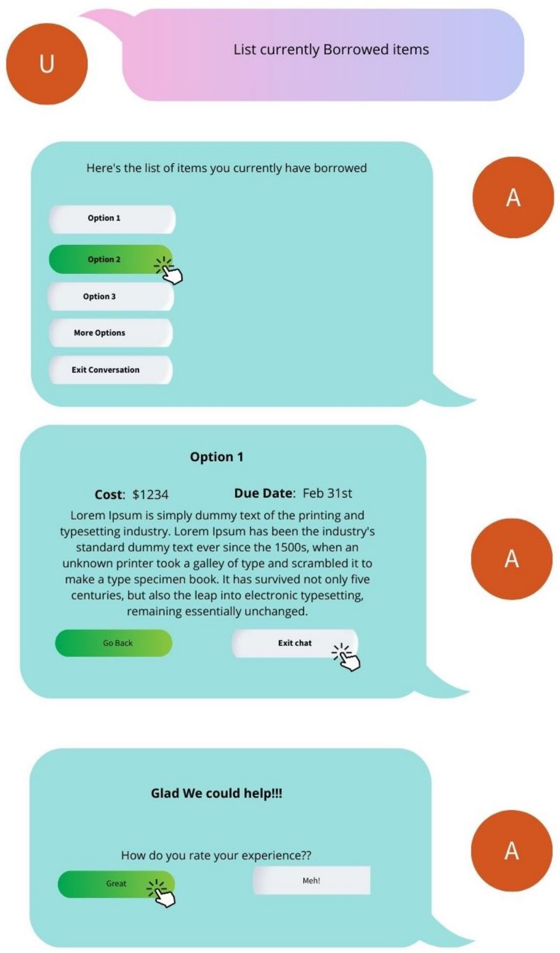
Entities: None required

Slots:

- Resolved: If the issue is resolved; it will be used to exit the form; if not, it will loop until the user asks to exit

3.2.5 File Complaint or Review

This story is triggered when a user wants to file a complaint or put a review about a product they purchased from the company. The chatbot first asks the user whether


List currently Borrowed items

Here's the list of items you currently have borrowed

Option 1

Cost: $1234 Due Date: Feb 31st

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

Glad We could help!!!

How do you rate your experience??

Figure 8: Sample Conversation for the story in Figure 7

they want to file a complaint or put a review. The chatbot extracts the "Complaint or Review" entity based on the user's response.

If the user wants to file a complaint, the chatbot asks for the details of the issue. Based on the user's response, the chatbot creates a ticket management system ticket and provides the user with a ticket number.

If the user wants to put in a review, the chatbot asks for the product name and the user's review. The chatbot posts the review on the company's website based on the user's response.

The conversation in Figure 10 is an example of how the chatbot interacts with the user based on the story in Figure 9. In this example, the user wants to file a complaint about a product they purchased. The chatbot asks for the details of the issue and creates a ticket in the ticket management system. The chatbot provides the user with a ticket number and asks if they need further assistance. The user confirms that they do not need further assistance, and the conversation ends.

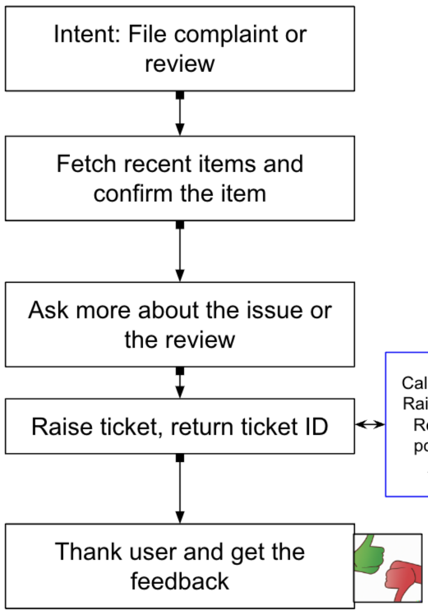

Intent: File complaint or review
Fetch recent items and confirm the item
Ask more about the issue or the review
Raise ticket, return ticket ID
Call Ticket Raising or Review posting API
Thank user and get the feedback

Figure 9: Story to file a complaint or put a review

Intent: File a complaint or put a review
Data Requirements:

1. Training:
   - Sample Intent: Basic Intents created using Natural Language generation tools for this intent
2. Live chatbot:
   - Complaint API, based on the ticket management tool
   - Review API; currently being already used by the company

Entities:

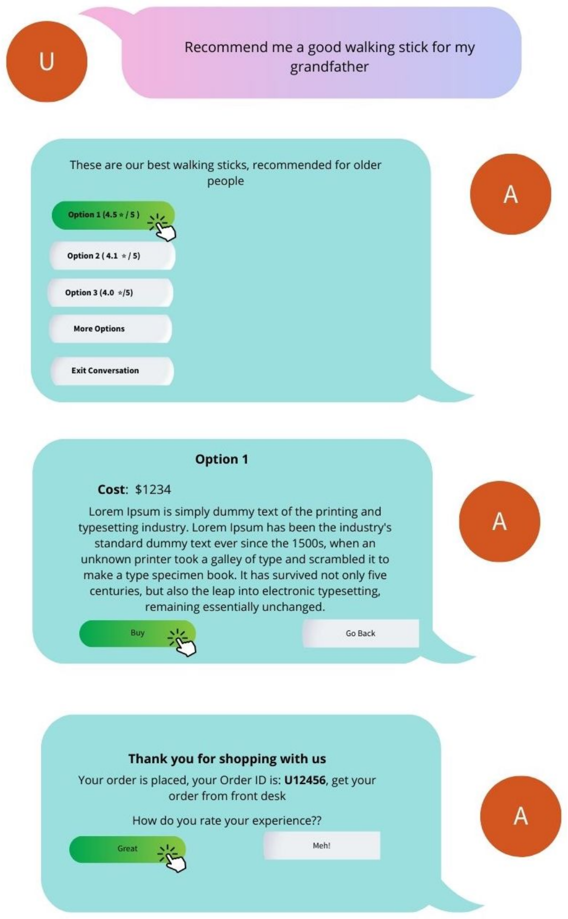
- Complaint or Review: User mentions complaint or review of the product

Slots:

- Complaint or Review: User mentions complaint or review
- Selected Item: User input for the item they want to review or complaint about

3.3 Generic Conversational Intents

In developing a chatbot, it is essential to define a set of generic conversational intents that can help guide the interaction between the chatbot and the user. These intents provide a foundation for the chatbot to handle common user queries and requests.



Recommend me a good walking stick for my grandfather

These are our best walking sticks, recommended for older people

Option 1

Cost: $1234

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

Thank you for shopping with us

Your order is placed, your Order ID is: **U12456**, get your order from front desk

How do you rate your experience??

Figure 10: Sample Conversation for the story in Figure 9

3.3.1 Are you a bot?

The "Are you a bot?" intent is designed to help users understand that they are interacting with a chatbot rather than a human. This intent is triggered when the user asks a question or makes a statement suggesting confusion about the conversation's nature. The chatbot can respond by acknowledging it is a bot and explaining how it can assist the user.

3.3.2 Greetings (Hello/Hi)

The "Greetings" intent acknowledges the user's greeting and initiates the conversation. This intent is triggered when the user says "Hello" or "Hi" to the chatbot. The chatbot can respond with a friendly greeting and ask how it can assist the user.

3.3.3 Farewell (Goodbye)

The "Farewell" intent ends the conversation between the user and the chatbot. This intent is triggered when the user says "Goodbye" or a similar phrase. The chatbot can respond with a polite farewell and offer to assist the user again in the future.

3.3.4 Thank you

The "Thank you" intent acknowledges the chatbot's assistance and expresses gratitude. This intent is triggered when the user says "Thank you" or a similar phrase. The chatbot can respond with a friendly message, such as "You're welcome" or "Glad I could help".

3.3.5 Help

The "Help" intent is used to assist the user in finding the information they need. This intent is triggered when the user asks for help or struggles to find the necessary information. The chatbot can respond by offering suggestions and guiding the user to the appropriate resources.

3.4 Handling Digression

The proposed system is designed to limit user input to a set of predetermined options in situations where the conversation may go off-topic or deviate from the intended purpose. This is particularly useful when the user must choose from a list of options, such as selecting a piece of equipment from a list of available options.

By limiting the user's input to a predefined list of options, the system can prevent the conversation from straying into irrelevant or unwanted areas. This approach helps ensure that the user's intentions are clearly understood and that the conversation remains focused on the task.

Furthermore, by restricting user input to a set of options, the system can more easily identify the user's intent and provide a suitable response. This can help streamline the conversation, reducing the risk of miscommunication or confusion.

Overall, the proposed system helps prevent the spawning of unwanted conversation trails by restricting the conversation to the user's intent. By limiting user input in situations where there is a possibility of digression, the system can improve the efficiency and effectiveness of the conversation, leading to a better overall user experience.

3.5 Required APIs

This sub-section lists the different APIs that are required for the development and functioning of the chatbot. These APIs include natural language processing APIs, customer relationship management APIs, and ticket management APIs.

The below list talks about the required API's needed to support the chatbot, some of these API's would already be present with ACME, since they've a functioning website and are already taking in orders. Others would be created by our backend engineers as part of the solution.

1. GET
   (a) **search item by name/type**
   returns a list of equipments using query parameters name or type. Ideally this should already exist before the chatbot implementation.
   (b) **get item by id**
   Similar to the one above, retrieve an equipment using it's id.
   (c) **get user purchase history**
   List all the items the user has bought or rented (but not returned) in the past.
2. POST
   (a) **chat**
   The primary API used to talk to the rasa instance. It's defined in the backend and is re-routed to hit the rasa endpoint.
   (b) **chat buttons**
   Similar to the one above but used on button clicks. Would required additional validation to ensure it has been produced from a button click.
   (c) **buy item**
   Handles when user wants to go ahead and buy an item. Would use the existing payment wall before it can be completed.
   (d) **rent item**
   Similar to the one above but for renting items.
   (e) **submit complaint**
   Submits a complaint (text) for an item in question.

3.6 Client-Server Architecture

This sub-section covers the client-server architecture of the chatbot, which includes the frontend interface for users and the backend infrastructure for processing user requests and generating responses.

1. **Frontend:** The frontend is the part of the chatbot architecture that provides a user interface for the clients to interact with. It typically includes a web or mobile application that allows users to send messages and receive responses from the chatbot. The frontend sends user requests to the load balancer, which then forwards them to the appropriate backend instance.

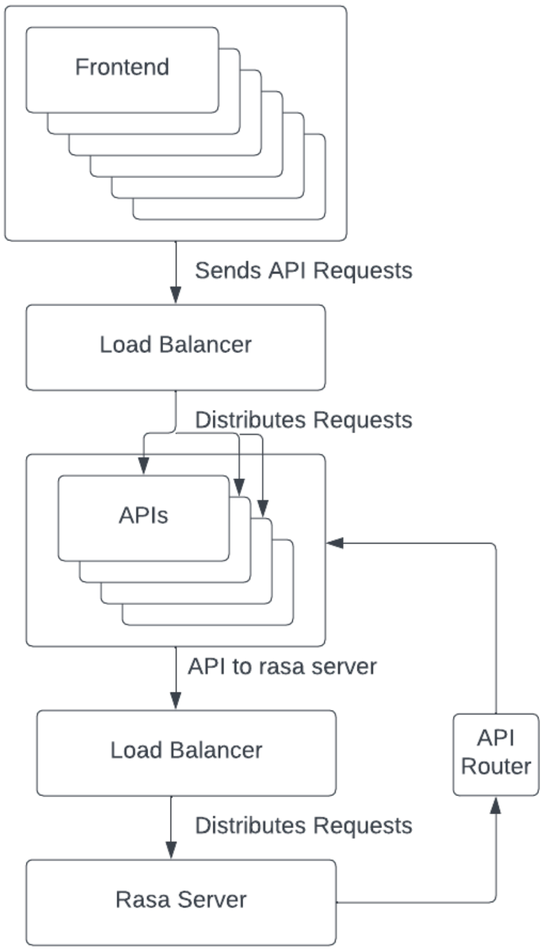


Figure 11: Client-Server Architecture

2. **Load Balancer 1:** The first load balancer in the chatbot architecture is responsible for distributing incoming user requests across multiple instances of the backend application. Its primary function is to ensure that no single server gets overloaded with requests and that the requests are handled in a timely and efficient manner.

3. **Backend:** The backend is the server-side component of the chatbot architecture that processes the requests sent by the frontend and returns the appropriate responses. It typically includes various functionalities, such as authentication, database management, and business logic. The backend communicates with the API router to send user requests to the Rasa server for natural language processing.

4. **API Router:** The API router acts as an intermediary between the backend and the Rasa server. It receives requests from the backend and forwards them

to the appropriate Rasa server instance, and vice versa. The API router also handles the necessary tokens required for authentication.

5. **Rasa Server:** The Rasa server is the component responsible for handling natural language processing (NLP) tasks, such as intent classification and entity extraction. It receives user requests from the API router and returns responses containing NLP outputs. The Rasa server uses machine learning algorithms and models to process the user requests and generate appropriate responses.

6. **Load Balancer 2:** The second load balancer in the chatbot architecture is responsible for distributing incoming user requests across multiple instances of the Rasa server. Its primary function is to ensure that no single server gets overloaded with requests and that the requests are handled in a timely and efficient manner.

3.7 Testing in Production

To ensure that the chatbot continues to perform optimally and effectively in production, it is important to closely monitor the system and retrain the model periodically. This involves setting up a robust monitoring process that tests for concept drift using failure rates and feedback, as well as collecting and tagging new data to retrain the model every three months.

In addition to testing for concept drift, it is important to continuously monitor the chatbot's performance metrics, such as response time, accuracy, completion rate, user satisfaction, and engagement. Any significant changes in these metrics may indicate issues or errors in the chatbot's responses, which need to be addressed promptly.

Furthermore, the model should be retrained every three months based on collected and tagged data to ensure that it remains up-to-date and accurate. The new data should include a diverse set of queries and intents to ensure that the model can handle a wide range of use cases. During the retraining process, the model would be evaluated using various performance metrics, such as accuracy and precision on already created validation data, to ensure that it is performing optimally.

To streamline the testing and deployment process, a continuous integration and continuous deployment (CICD) process can be implemented. This involves automating the build, test, and deployment process, allowing for more efficient and reliable updates to the chatbot.

3.8 Continuous upgrading strategy

Our continuous upgrading strategy for the advising chatbot involves implementing a mechanism to capture and collect all conversations between the user and the chatbot. By doing so, we will be able to analyze and identify sentences that fall under certain intents but were not captured by the model during the conversation.

This mechanism will enable us to improve the chatbot's performance by identifying gaps in its current understanding and addressing them. By capturing more well-rounded conversations and corner cases for sentences that the model has not encountered before, we can ensure that the chatbot is better equipped to handle a wide range of queries and provide accurate recommendations to users.

Furthermore, since the chatbot provides healthcare equipment advice, it is essential to ensure that it remains updated and relevant. Therefore, we plan to incorporate a regular review process to analyze the data collected and update the model accordingly. This review process will involve examining the conversations and identifying patterns and trends to improve the chatbot's performance.

4 Deliverables

This section outlines the deliverables for the project, including the development timeline, milestones, and completion deadlines.

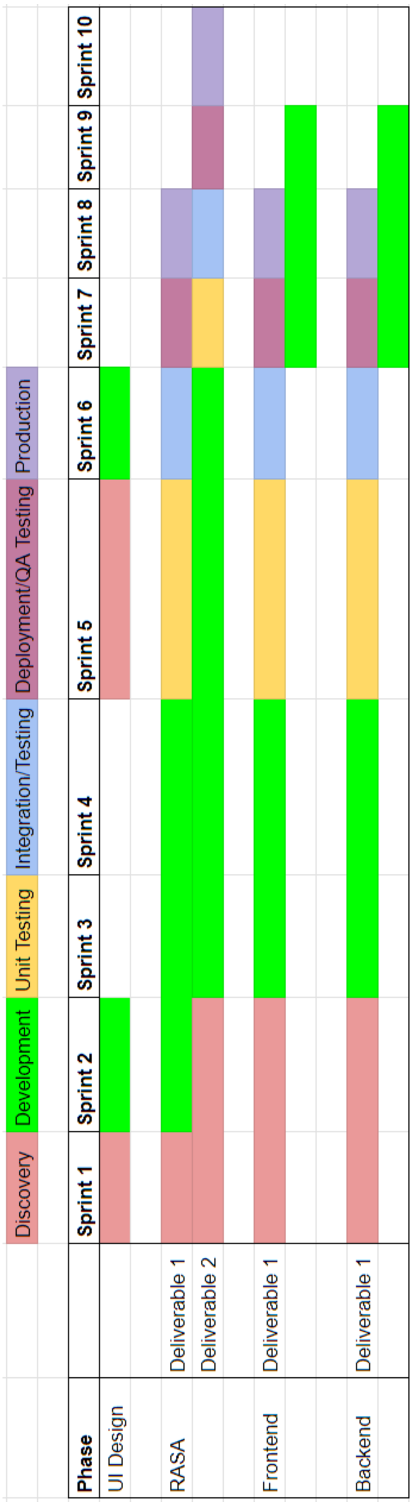
4.1 Development Timeline

The image in Figure 12 describes the development timelines requested to complete the project. We plan to complete the first deliverable by sprint eight and the second by sprint 10. The four teams: UI design, RASA, frontend and backend, would start parallelly. Provided below is a description of each section in the development timelines section.

Red Section: Discovery Phase
The red section on the development timeline represents the project's discovery phase. During this initial phase, the UI design team collaborates with stakeholders to gather requirements for the chatbot interface, ensuring that it aligns with their needs and expectations. The outcome of this phase is a comprehensive design document, which will serve as a blueprint for the subsequent development stages.

Green Section: Development Phase
The green section on the development timeline indicates the commencement of the development phase. The RASA, frontend, and backend teams will begin working on their respective components. The RASA team will focus on natural language processing (NLP) and the underlying chatbot logic, the frontend team will develop the user interface, and the backend team will be responsible for the server-side processing of user requests. All teams will work concurrently, adhering to the design document from the discovery phase and this document.

Yellow Section: Unit Testing Phase
The yellow section on the development timeline signifies the conclusion of the development phase and the beginning of unit testing. During this phase, each team will rigorously test their respective components to ensure they function correctly and meet the project requirements. Unit testing is essential in identifying and rectifying errors or bugs before integrating different components.



🟥 Discovery | 🟩 Development | 🟨 Unit Testing | 🟦 Integration/Testing | 🟫 Deployment/QA Testing | 🟪 Production

Phase		Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10
UI Design		🟥	🟩			🟥	🟩				
RASA	Deliverable 1	🟥	🟩	🟩	🟩	🟨	🟦	🟫	🟪		
	Deliverable 2	🟥	🟥	🟩	🟩	🟩	🟩	🟨	🟦	🟫	🟪
Frontend	Deliverable 1	🟥	🟥	🟩	🟩	🟨	🟦	🟫	🟪		
								🟩	🟩	🟩	
Backend	Deliverable 1	🟥	🟥	🟩	🟩	🟨	🟦	🟫	🟪		
								🟩	🟩	🟩	

Figure 12: Delivery timeline

Light Blue Section: Integration and Testing Phase

The light blue section on the development timeline denotes the integration and testing phase. In this phase, the RASA, frontend, and backend teams will integrate their respective components and test their compatibility and overall functionality. This crucial step ensures that the chatbot functions cohesively and effectively, meeting the expectations during the discovery phase.

Magenta Section: Deployment and QA Testing Phase
The magenta section on the development timeline corresponds to the deployment phase. The chatbot will be deployed on development systems and subjected to quality assurance (QA) testing. The QA team will verify that the chatbot satisfies the requirements and specifications established during the discovery phase. Any issues or bugs discovered during this phase will be resolved before the chatbot's deployment in the production environment.

Purple Section: Production and Maintenance Phase
The purple section on the development timeline represents the production phase, marking the final stage of the development process. The chatbot will be deployed to the production environment and available to users. The development team will be responsible for ongoing maintenance, support, and necessary updates to ensure a reliable, functional, and user-friendly chatbot experience. The successful completion of the project signifies the end of the development cycle and the beginning of a new journey toward creating an effective and engaging chatbot.

4.2 Out of scope/future improvements

Currently, the proposed chatbot's functionality is limited to the English language, which narrows its potential user base to only those who speak English. However, we acknowledge the importance of language diversity, and as the following deliverables, we could plan to incorporate support for multiple languages in the future. This feature will enable individuals who do not speak English as their primary language to use the chatbot and reap its benefits.

Moreover, we recognize that some patients may have specific disabilities preventing them from typing their queries. To ensure that the chatbot remains accessible to all, the chatbot can be further developed to include voice support in the future. This feature will enable patients to interact with the chatbot using voice commands, providing a more inclusive and user-friendly experience.

5 Resources

This section covers the resources required for the development and deployment of the chatbot. This includes the costs associated with human resources, hosting, and development equipment.

5.1 Human Resource Costs

Phase	Task	Engineers	Timeline	Hours	Count	Hourly Wage($)	Cost Range
Discovery	Business analysis	Business Analyst	2-3 weeks	80-120	1	$70 - $120	$5.6k - $14.4k
	User research	User Researcher	2-3 weeks	80-120	1	$70 - $120	$5.6k - $14.4k
	Requirements analysis	Technical Analyst	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
	Project management	Project Manager	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
Design	UI/UX design	UX Designer	3-4 weeks	120-160	1	$70 - $120	$8.4k - $19.2k
	Visual design	Visual Designer	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
	Content creation	Content Creator	1-2 weeks	40-80	1	$50 - $100	$2k - $8k
Development	Front-end development	Front-end Developer	6-8 weeks	240-320	2	$70 - $120	$33.6k - $76.8k
	Back-end development	Back-end Developer	6-8 weeks	240-320	2	$70 - $120	$33.6k - $76.8k
	Integration and testing	Quality Assurance	1-2 weeks	40-80	2	$50 - $100	$4k - $16k
Infrastructure	Server setup	DevOps Engineer	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
	Load balancer setup	DevOps Engineer	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
	Monitoring setup	DevOps Engineer	1-2 weeks	40-80	1	$70 - $120	$2.8k - $9.6k
Data Science	Data pre-processing	Data Scientist	4-6 weeks	160-240	2	$120 - $180	$38.4k - $86.4k
	Building and training the model	Data Scientist	8-12 weeks	320-480	2	$120 - $180	$76.8k - $172.8k
Total							$224.8k - $542.4k

5.2 Role Descriptions

1. **Business Analyst**
   Responsible for gathering requirements and identifying the chatbot's features and functionalities, translating business needs into technical specifications, and designing a solution that meets the needs of stakeholders. Additionally, they map out the processes that the chatbot will support, such as equipment selection, inventory management, and order processing, to ensure that it is designed efficiently and effectively. The business analyst works with the development team to define test cases, validate the chatbot's performance, and provide training and support to end-users and support staff. Their collaborative efforts ensure that the chatbot delivers value to the organization by meeting user needs and improving overall efficiency.

2. **User Researcher**
   The User Researcher is responsible for understanding the needs and preferences of the target audience. They conduct research to identify user behaviors and pain points and use the insights gathered to inform the design of the chatbot.

3. **Technical Analyst**
   The Technical Analyst is responsible for designing the technical infrastructure and systems that support the chatbot. They evaluate the feasibility of integrating different technologies and identify the most suitable development platform to create the chatbot.

4. **Project Manager**
   The Project Manager is responsible for managing the overall project and ensuring that the chatbot is delivered on time and within budget. They coordinate with stakeholders, manage project resources, and oversee the development team to ensure that the chatbot is built according to the project plan.

5. **UX Designer**
   The UX Designer is responsible for designing the user experience of the chatbot. They create wireframes, prototypes, and user flows to ensure that the chatbot is easy to use and meets the needs of its intended audience.

6. **Visual Designer**
   The Visual Designer is responsible for creating the visual design elements of the chatbot, such as color schemes, typography, and graphic elements. They ensure that the chatbot's design is visually appealing and consistent with the organization's branding guidelines.

7. **Content Creator**
   The Content Creator is responsible for creating the chatbot's content, such as product descriptions, FAQs, and responses to user queries. They ensure that

the chatbot's content is accurate, helpful, and aligned with the chatbot's overall messaging and tone.

8. **Frontend Developer**
   The Frontend Developer is responsible for developing the user interface of the chatbot. They create a responsive and interactive interface that allows users to interact with the chatbot.

9. **Backend Developer**
   The Backend Developer is responsible for developing the backend of the chatbot. They build the server-side application logic that powers the chatbot, handles user requests, and interacts with the chatbot's database.

10. **Quality Analyst**
    The Quality Analyst is responsible for ensuring that the chatbot meets the desired quality standards. They create test cases, perform manual and automated testing, and identify defects and issues that need to be addressed.

11. **DevOps Engineer**
    The DevOps Engineer is responsible for the deployment, testing, and monitoring of the chatbot. They ensure that the chatbot is deployed and running smoothly and securely, and that any issues that arise are addressed quickly.

12. **Data Scientist**
    The Data Scientist is responsible for analyzing the data generated by the chatbot. They use machine learning and data analytics techniques to identify trends, insights, and patterns that can help improve the chatbot's performance and user experience. They also ensure that the chatbot is collecting and storing data in a secure and compliant manner.

5.3 Hosting Costs

Resource	Type	Hourly Cost	Monthly Cost
EC2 instance	p3.2xlarge p2.xlarge	$3.06 $0.90	$2,203.00 $648.00
S3 bucket			$2.3 (10 GB storage, 10 GB data transfer)
Elastic Load Balancer	Application Load Balancer Network Load Balancer	$0.02 $0.05	$16.20 $34.56
AWS Lambda			$0.15 (1 million requests, 500ms average duration, 512 MB memory)
AWS API Gateway			$0.65 (1 million API calls, 100 GB data transfer)
Total			$1000 - $ 2500

5.4 Development Equipment Costs

Resource	Type	Rental cost per month	license required	Cost per month
Rasa	Free Tier	-	-	-
EC2 instance	p2.xlarge-p3.2xlarge	$650-$2200	2	$1300 - $4400
Laptop Rental Cost	MacBook rental cost	$100	12	$1,200.00
IntelliJ Licensing		$60	6	$360.00
Postman		$12	6	$72.00
Gitlab		$24	12	$288.00
Total				$3800 - $6800

NOTE:

It is worth noting that many of the costs presented in this proposal are based on the assumption that the client does not currently possess the necessary infrastructure to support the development of the proposed solution. However, if the client has already acquired licenses for these infrastructural components within their organization, then the costs associated with this project can be reduced accordingly. In other words, if the client already possesses the necessary hardware or software, the project costs can be lowered since the required infrastructure is already in place. Therefore, it is essential to assess the client's existing infrastructure before finalizing the budget to ensure that the costs are accurate and reflect the true requirements of the project.

References

[1] AWS Services Prices. https://aws.amazon.com/pricing.

[2] IntelliJ Jetbrains Prices. https://www.jetbrains.com/idea/buy/?section=commercial&billing=yearly.

[3] Gitlab Prices. https://about.gitlab.com/company/pricing/.

[4] Postman Prices. https://www.postman.com/pricing/.

[5] Wage Estimates. https://www.payscale.com/research/US/Industry=Information_Technology_(IT)_Services/Salary.