Aalto University

Shipping/Cargo CLI ChatBot

Rizzi Matteo

808914

A report submitted in partial fulfilment of the requirements
for the User Interface Construction course

November 20, 2019

Abstract

This report describes the design and the implementation of an Interactive Command Line Interface. It has been developed in the form of a Web Application so everyone can use it through the browser with their smartphone, laptop or any device. The goal of this simple application is to provide a registered user with a comfortable and very intuitive command-line interface for adding a new delivery with all the required data. The tool is implemented in such a way that it is easy enough to use by users with different skills and knowledge. This report assumes that users have access to any kind of browser with an Internet connection. Throughout the report there will be also a login page but, since it is a "chatbot prototype", the signup/login functionalities are just designed neither implemented nor stored in a database.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Nowadays, a lot of people started to create their small company with their own products. A common issue is the delivery process because usually there are a lot of steps to plan an efficient express delivery. For this reason, the introduction of an easy-accessible web application should be a crucial step to facilitate the growth of these new entrepreneurs.

1.2 Solution

The web application presented in this report introduces an intuitive way to plan one or more deliveries. In fact, all different types of users could easily insert a new delivery to the system through the use of an innovative chat bot. It works as a classic chat bot but in the end, the user should appreciate the mixing between "typing-commands" and "button-choices". To better understand this point, please watch the live-demo on this link:
https://youtu.be/S5S2xPZemDw

Chapter 2

Users

2.1 Skills and knowledge

As stated above in the previous chapter, the application was designed as simple as possible. Despite this, a small previous knowledge about IT terms such as log-in, sign-in, and submit, is required.

2.2 Tasks

A typical use case for this web application is the following one: a rural micro-entrepreneur has an emerging economy and he has made a deal with a larger company to deliver them gems/jewellery. Once the business owner has produced a batch of products and the delivery is ready, he needs to plan the delivery to the "destination company". Therefore, to reach his final goal, the user has to carry out different tasks:

- access to the browser;
- sign-up/log-in to the application;
- interacts with the chat bot by providing details about the delivery (sender, customer, pick-up address, destination address, amount of products, price);
- (optional) find an already-inserted delivery to check its details and its status;
- (optional) contact an operator or check the f.a.q to solve problems and doubts.

Chapter 3

Application

3.1 Dialogue

As mentioned in the Introduction, the user interacts with an innovative chat bot which provides a mix of input commands and button choices. Diagram 3.1 shows the flow of the dialogue between the user and the application itself [1].

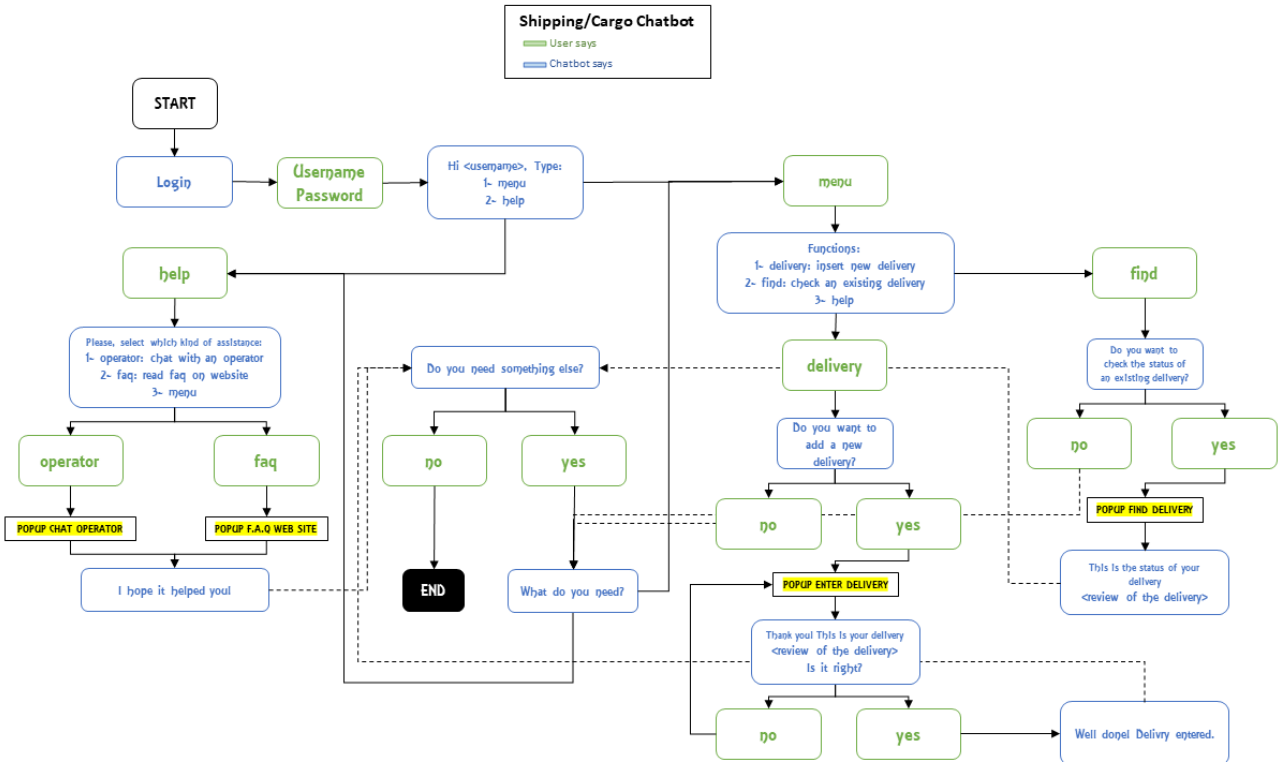

Figure 3.1: *Chat bot flow of dialogue.*

3.2 UI

Since the application is targeted at any device with a working browser, the UI was conceived to work well both on laptops and smartphones. It has been developed using front-end coding language such as HTML5, CSS and JavaScript [6]. Talking about the server-side, it is a very basic Flask Python Web Application [2] actually hosted by Heroku service. It is accessible by the following link:
https://chatbot-uic2019.herokuapp.com.

As suggested by guideline number 1.3/25 (refer to Data entry guideline section), all the commands submitted by users are shown in the chat text area. It goes to create a sort of history of the chat that makes it clearer and readable. Multiple-choice questions always adopt buttons with the possible alternatives, while the details of the delivery can be inserted through different input-prompt popups [4]. Furthermore, once a user has become an expert user, he has at his disposal some "escape-words" to make faster different operations:

- *delivery*,*add delivery*, *add*: allow to directly add a new delivery without following the "guided mode";
- *find*, *find delivery*: allow to look for an existing delivery without following the "guided mode";
- *help*, *help me*, *?*: allow to look for help directly.
- *esc*, *exit*, *finish*, *stop*: allow to close the conversation.

Here there are some screenshots to get a better look to the above description.

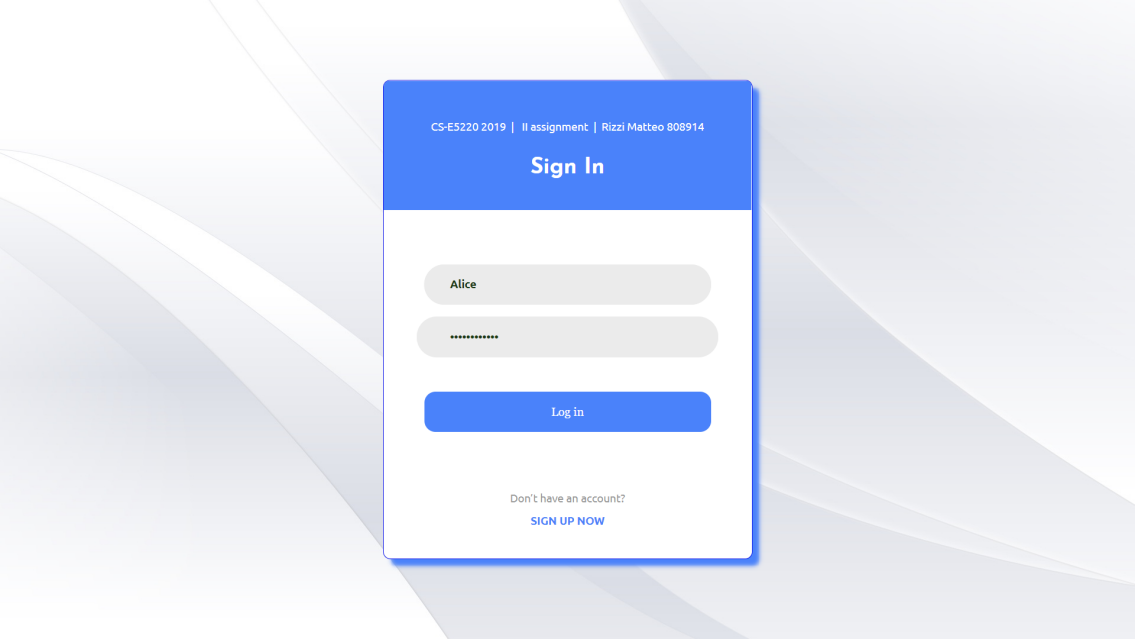

Figure 3.2: *Login page.*

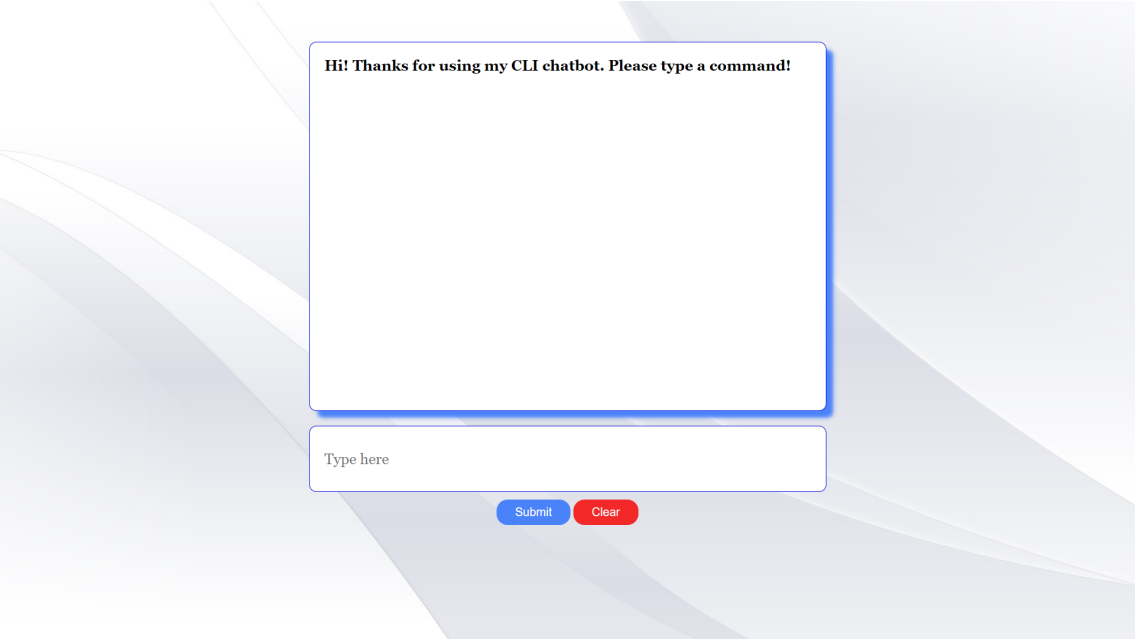

Figure 3.3: *Chat bot empty window.*

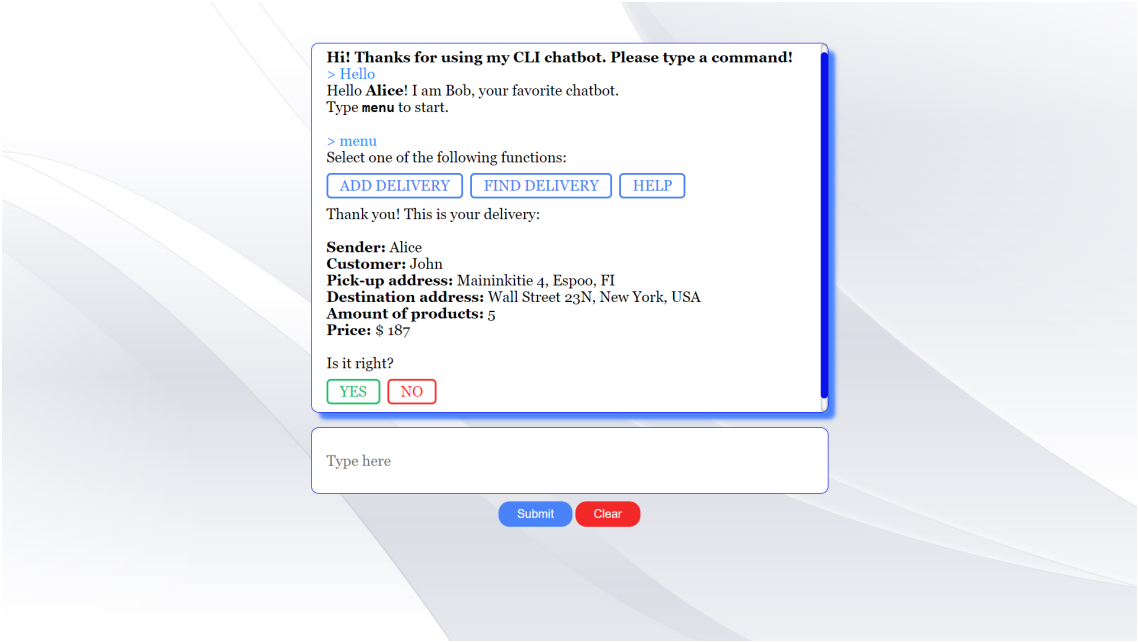

Figure 3.4: *Chat bot example of dialogue.*

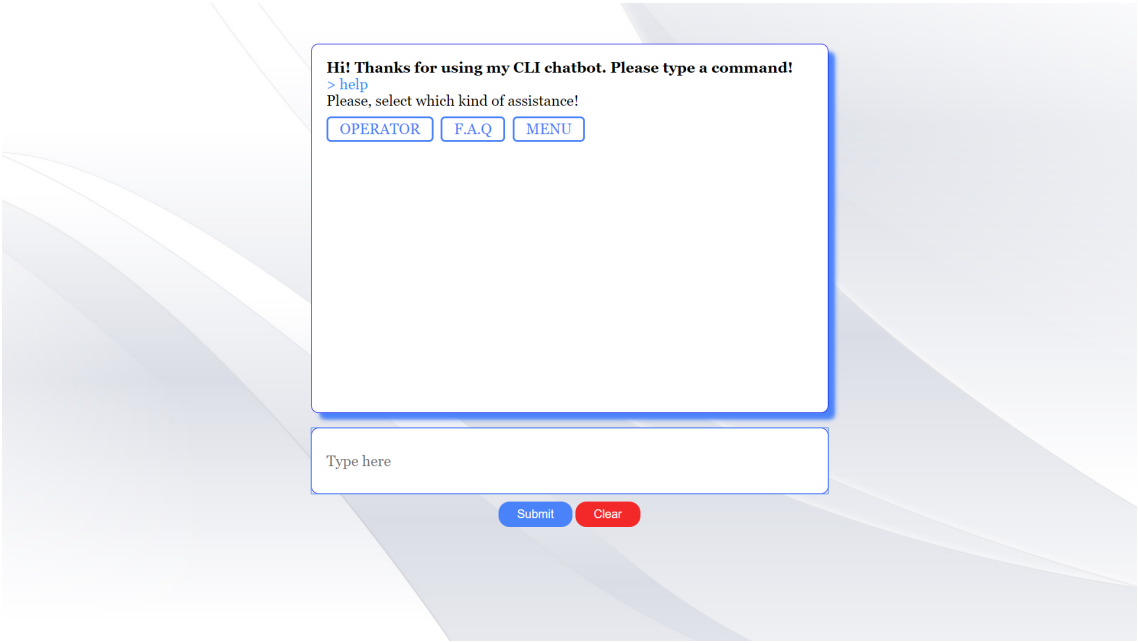

Figure 3.5: *Chat bot example of help function.*

Chapter 4

Guidelines

This chapter of the document describes the guidelines that are implemented. They are taken from Smith and Mosier guidelines [5] and grouped according to their specific functionalities.

4.1 Data entry

- 1.0/1 Data Entered Only Once
- 1.0/4 + Fast Response
- 1.0/6 Defined Display Areas for Data Entry
- 1.0/9 Explicit ENTER Action (e.g. Is it right? Yes)
- 1.0/11 Explicit CANCEL Action (e.g. Is it right) No)
- 1.3/1 Adequate Display Capacity
- 1.3/10 + Upper and Lower Case Equivalent
- 1.3/25 Necessary Data Displayed
- 1.4/1 Combined Entry of Related Data
- 1.4/3 Minimal Use of Delimiters (e.g. Dollars symbol is not asked, it is already written)
- 1.6/7 + Highlighting Selected Elements (e.g. Buttons change shape when hovering on)

4.2 Data Display

- 2.0/12 Familiar Wording
- 2.0/13 + Consistent Wording
- 2.0/15 + Consistent Grammatical Structure
- 2.0/16 Minimal Use of Abbreviation
- 2.1/4 + Adequate Display Capacity
- 2.1/7 + Separation of Paragraphs (e.g in the review of the inserted delivery)
- 2.1/8 + Consistent Word Spacing
- 2.4.7/9 Single Decision at Each Step
- 2.4.7/10 Logical Ordering of Options
- 2.6/27 + Easily Discriminable Colors
- 2.6/28 + Conservative Use of Color (e.g. green for Yes, red for No, just one light blue colour used for others functions)

4.3 Sequence control

- 3.0/2 Minimal User Actions
- 3.0/3 Control Matched to User Skill
- 3.0/4 User Initiative in Sequence Control (e.g. The users have the control in their commands)
- 3.0/5 Control by Explicit User Action
- 3.0/18 Appropriate Computer Response Time
- 3.1.1/1 Question-and-Answer Dialogue
- 3.1.1/2 Questions Displayed Singly
- 3.1.1/3 Recapitulating Prior Answers

- 3.1.3/10 Explanatory Title for Menu (e.g. "What do you need?" and then Help, Menu buttons)
- 3.1.3/17 + Complete Display of Menu Options
- 3.1.3/21 Logical Ordering of Menu Options (e.g. Add Delivery, Find Delivery, Help)
- 3.1.4/2 + Function Keys for Frequent Control Entries (e.g hello, esc)
- 3.2/10 Only Available Options Offered
- 3.3/5 REVIEW Option (e.g. review of the delivery)
- 3.3/7 END Option (e.g. end with escape characters)

4.4 User guidance

- 4.0/2 Explicit User Actions
- 4.4/8 + Standard Display Location for Prompting (e.g. all data entry prompts are at the top-middle of the page)
- 4.4/9 + Consistent Format for Prompts
- 5.3/6 Sender Identification (e.g. "Delivery inserted!" message)

Chapter 5

Result

The aim of this chapter is to evaluate in a "tangible way" the user interface showed in the previous chapters. To do this, it is useful follow the Nielsen's Ten (reformed 1999) Heuristics [3]. The following table helps to evaluate the UI according to different parameters:

Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?
1. Visibility of system status	Yes, since there is almost no waiting time. And again: when the delivery is inserted a popup confirming it is shown	No
2. Match between system and the real world	Yes, since all the chat bot is a digital imitation of a real world conversation	No
3. User control and freedom	Yes, since user has always the control on the next step. He is also free because of the "esc" function to quit the chat	No
4. Consistency and standards	Yes, since all functions and commands are clear and easy to be executed	No
5. Error prevention	No	Yes, since the user is not helped in preventing errors but he is just notified when the error occurs
6. Recognition rather than recall	Yes, because all the information are always shown to the user. Infact he does not need to remember any information	No
7. Flexibility and efficiency of use	Yes, because an expert user can speed up the interaction using "shotcuts" like the command "add" instead of following the guided mode	No
8. Aesthetic and minimalist design	Yes, the visual layout is very basic and it puts in evidence just the strictly needed informations	No
Help users recognize, diagnose and recover from errors	No	Yes, even if there are some error messages, they are not enough to allow user to diagnose and recover from its errors
Help and documentation	No	Yes, because of the simplicity of the CLI, documentation is not provided to the users

Bibliography

[1] *Chat bot flow*. https://www.userlike.com/en/blog/chatbot-conversation-flow. Accessed: 2019-11-19.

[2] *Flask*. https://www.fullstackpython.com/flask.html. Accessed: 2019-11-19.

[3] *Nielsen Heuristics*. http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5. Accessed: 2019-11-19.

[4] Ben Shneiderman et al. *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 2016.

[5] Sidney L Smith and Jane N Mosier. *Guidelines for designing user interface software*. Tech. rep. Citeseer, 1986.

[6] *W3Schools Online Web Tutorials*. https://www.w3schools.com. Accessed: 2019-11-19.