

UNIVERSITY OF TARTU
FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE
INFORMATION TECHNOLOGY SYSTEMS DEVELOPMENT

Ilja Andrejev

CREATING AN AUTO-REPLY BOT FOR CAFÉ MUNA FACEBOOK PAGE

Diploma thesis

Supervisor: Assistant Andre Säask

NARVA 2019

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

...../töö autori allkiri/

CONTENTS

CONTENTS	3
TERMS AND ABBREVIATIONS	6
INTRODUCTION	7
The problem.....	7
The solution	8
The aim.....	8
Tasks	8
Outline.....	9
1 PREPARING FOR THE PROJECT	10
1.1 Market research	10
1.2 Similar solutions	11
1.2.1 ManyChat	11
1.2.2 Chatfuel	11
1.2.3 Conclusion.....	12
1.3 Tools and technologies	12
1.3.1 Node.JS	12
1.3.2 JavaScript	13
1.3.3 PHP	13
1.3.4 jQuery.....	13
1.3.5 HTML and CSS	14
1.3.6 Skeleton.....	14
1.3.7 Microsoft Visual Studio Code	14
1.3.8 XAMPP.....	14
1.3.9 ngrok	15
2 OVERVIEW OF THE BOT	16
2.1 Requirements.....	16
2.1.1 Functional.....	16
2.1.2 Non-functional.....	16
2.2 The proposed system	17
2.3 Modular description.....	17
2.3.1 Command identification module	18
2.3.2 Making function module	18
2.3.3 JSON knowledge base.....	18
3 OVERVIEW OF THE BOT'S ADMINISTRATION CONSOLE.....	19

3.1	Requirements.....	19
3.1.1	Functional.....	19
3.1.2	Non-functional.....	19
3.2	Administrator rights.....	20
3.3	Security.....	20
3.3.1	Log in.....	20
3.3.2	Password standard.....	20
3.3.3	Automatic Log out.....	20
3.4	Back-end.....	21
3.5	Front-end.....	21
4	THE RESULT.....	22
4.1	The final system diagram of the developed system.....	22
4.2	Bot views.....	23
4.2.1	Get started and welcome message views.....	23
4.2.2	Lunch of the day view.....	24
4.2.3	A la carte menu view.....	25
4.2.4	Top-level commands view.....	26
4.2.5	Dishes view.....	27
4.3	Administration console views.....	27
4.3.1	Log in.....	27
4.3.2	Lunch view.....	29
4.3.3	Navigation tabs bar view.....	29
4.3.4	Course view.....	29
4.4	Testing.....	31
4.5	The bot testing.....	31
4.5.1	Mobile.....	31
4.5.2	Desktop.....	31
4.6	The administration console testing.....	32
	CONCLUSION.....	33
	SUMMARY IN ESTONIAN.....	34
	REFERENCES.....	35
	APPENDICES.....	37
	APPENDIX 1 Desktop & Mobile Social Media Stats Europe.....	37
	APPENDIX 2 Desktop & Mobile Social Media Stats Estonia.....	38
	APPENDIX 3 Facebook requirements.....	39
	APPENDIX 4 Transcript of the Interview.....	40

APPENDIX 5 Source code.....	41
NON-EXCLUSIVE LICENSE TO REPRODUCE THESIS	42

TERMS AND ABBREVIATIONS

API – Application Programming Interface

IDE – Integrated Development Environment

HTTP – Hyper Text Transfer Protocol

HTML – Hyper Text Markup Language

CSS – Cascading Style Sheet

Bot – a computer program that works automatically. (Definition of “Bot” ... 2019). Bots are sets of algorithms that take on the duties of repetitive sets of instructions in order to establish a service or connection among social networking users. Various designs of networking bots vary from chat bots, algorithms designed to converse with a human user, to social bots, algorithms designed to mimic human behaviors to converse with behavioral patterns similar to that of a human user. (Internet Bot, Wikipedia 2019)

Social network – a website or computer program that allows people to communicate and share information on the internet using a computer or mobile phone. (Definition of “Social network” ... 2019)

Facebook Messenger – Facebook Messenger (commonly known as Messenger) is a messaging app and platform. Originally developed as Facebook Chat in 2008, the company revamped its messaging service in 2010, and subsequently released standalone iOS and Android apps in August 2011. Over the years, Facebook has released new apps on a variety of different operating systems, launched a dedicated website interface, and separated the messaging functionality from the main Facebook app, requiring users to use the web interface or download one of the standalone apps. (Facebook, Wikipedia 2019).

WebHook – A WebHook is an HTTP callback: an HTTP POST that occurs when something happens; a simple event-notification via HTTP POST. (Davis 2019)

HTTP POST – HTTP method is used to send data to a server to create/update a resource. (HTTP Request Methods ... 2019)

JSON – JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. (JSON ... 2019)

INTRODUCTION

The way people communicate with businesses has been evolving at a rapid pace. For years, meetings and phone calls were the main way of communication. Then, with the rise of the Internet, a few new important options became available, from email, to filling out a form on a website, to social media, to mobile apps.

With the prevalence of virtual assistants like Siri and Alexa, the number of apps people use has declined. Now, for convenience, they simply ask their virtual assistant to pull up a detail over the Internet, set reminders or even make a call. Asking Siri or Alexa to do something is a lot easier than doing it yourself, especially if you are otherwise occupied like eating or driving a car. Asking a virtual assistant to help decidedly saves the hassle and potentially a little time. Therefore, with the growth of chatbots, apps have started to become the thing of the past. While the change may take up a lot of time to garner traction and not everyone will adhere to it right away, the time for it is coming. In other places, though, the change has already taken place and the apps have already been phased out and have been replaced with chatbots as the latter are far more reliable and beneficial to businesses.

Social networks are integral part of our modern life. Each of us has a page in one or even several popular social networks, such as Facebook¹, Twitter², VK³ and LinkedIn⁴. Social networks have long been a platform for businesses. Many modern companies have their own pages on social network platforms. Café “Muna”⁵, which is located in the building of University of Tartu Narva College, is not an exception.

According to TripAdvisor⁶ café “Muna” is the best café in Narva. Daily, it serves a wide variety of dishes from a la carte menu. It also has daily specials for a quick lunch every working day. Additionally, there’s a possibility to order catering services as needed.

The problem

With the increasing popularity of social networks and instant messaging applications, there is a growing number of clients who are trying to get information using them. According to the

¹ <https://facebook.com>

² <https://twitter.com>

³ <https://vk.com>

⁴ <https://linkedin.com>

⁵ <https://www.facebook.com/kohvikmuna/>

⁶ https://www.tripadvisor.com/Restaurants-g277793-Narva_Ida_Viru_County.html

manager of the café “Muna”, there are dozens of clients contacting her daily with the same questions regarding the menus and dishes. Answering all those questions promptly is a challenging task, which is often impossible to accomplish due to the heavy load of main work at the café. At the same, waiting time is a crucial factor influencing relationships with customers, so improving it is vital for the café.

The solution

One of the possible solutions for the problem would be developing a bot for the café “Muna” that would automate the process of communicating with the client through instant messaging application. Bots enable businesses to respond to customer messages instantly and to radically increase the number of customers you can communicate with at once. Also, unlike human personnel, bots are available 24 hours a day, 7 days a week and 365 days a year.

The aim

The aim of this diploma thesis is to develop an automatic answering application (a bot) and an administration console for it.

The bot should be able:

- to communicate with a client in three languages (English, Estonian and Russian)
- to send a whole menu to a client
- to send a course of dishes to a client
- to send the lunch of the day to a client
- to send catering contacts to a client
- to send café contacts to a client
- to redirect a client to the café web page

The administration console should be able:

- to add a new dish to the bot data
- to remove a dish from the bot data
- to add a new lunch of the day to the bot data
- to remove lunch of the day from the bot data

Tasks

To reach the aim of this diploma thesis, the author has to complete the following tasks:

- Choose appropriate tools and technologies for developing a bot.

- Choose appropriate tools and technologies for developing an administration console.
- Develop bot functions for receiving and sending messages using Facebook API.
- Develop bot functions for communication with the administration console.
- Develop administration console functions for communication with the bot.
- Create a web-based user interface for the administration console.

Outline

This thesis consists from introduction, four chapters, conclusion and four appendices. The first chapter speaks about tools and technologies chosen for the thesis project. The second chapter speaks about developing the bot, the third chapter is dedicated to the administration console of the bot and the fourth chapter presents the overall results of the development process. The first two appendices contain statistics about Facebook usage (Estonia and Europe). The third appendix contains the list of functional and nonfunctional requirements. The fourth appendix contains a transcript of the interview with the manager of café “Muna”.

1 PREPARING FOR THE PROJECT

1.1 Market research

The choice of a social network for the bot development did not arise since café “Muna” already had its own official Facebook page with 1,342 followers, therefore it was an obvious choice to develop the bot for this specific platform.

Facebook has more than 2 billion monthly active users (Facebook, Wikipedia 2019).

According to statistics from StatsCounter⁷, over the last year in Europe Facebook have been used by 74.11% (Figure 1) and in Estonia by 60.86% (Figure 2) of active Internet users.

“Word of mouth” has always been the best marketing tool. Nothing convinces people more than the opinion of a close friend or even just a familiar person. But, until recent times, before massive use of social networks began, this method of advertising had still very limited use due to the restricted nature of existing channels of communications.

With the advent of social networks like Facebook, the situation has changed dramatically. Now users can share their favorite brands or discuss them publicly. It became possible to ask your regular customers or social network friends to put “Like” or click “Share” under your publications. Thus, promoting products or services became much simpler. Also, if you are developing a new product or service, it became much easier to carry out surveys to help make the final product more appealing to customers, since you would know their needs better.

According to Facebook IQ 48% of 6,250 local business owners across US, DE, FR, UK, IT, ES, PL and CA surveyed have started optimizing their marketing for mobile devices (Facebook IQ, 1, 2019). 64% of those local business owners say having an online presence is important for their long-term success (Facebook IQ, 2, 2019).

In recent years bots for Facebook are gaining popularity among users and local businesses. There are currently more than 300,000 active chatbots on Facebook Messenger (Nealon, 2018). So, we can conclude that Facebook is a promising platform for the growth of local businesses.

Bots allow customers to get in contact with companies whenever they want to, without paying attention to time zones, opening times and waiting loops of call and service centers. Bots are very promising for international and digital companies like online retailers or web shops.

⁷ <http://gs.statcounter.com>

Customers sometimes buy products in different countries and they do not want to be dependent on local time zone or foreign languages. A further advantage is one-to-one communication. So far, users often must search and browse a website for a long time to find the right information like product, price, service or contact information. In the case of complaints or other customer service inquiries, bots are helpful, straight-forward and efficient. (Zumstein, Hundertmark 2017: 102)

1.2 Similar solutions

At the moment there are various platforms on which you can create your own bot for Facebook messenger. The most popular are ManyChat or Chatfuel.

1.2.1 ManyChat

ManyChat⁸ was founded in 2015. This is the visual bot builder for Facebook Messenger. ManyChat has a free plan that comes with limited features and a link to ManyChat in the bot menu and promotional tools. Paid plans open access to all functions, unlimited tags, custom fields, and remove the ManyChat link in the menu. (ManyChat logo will still appear in some marketing functions).

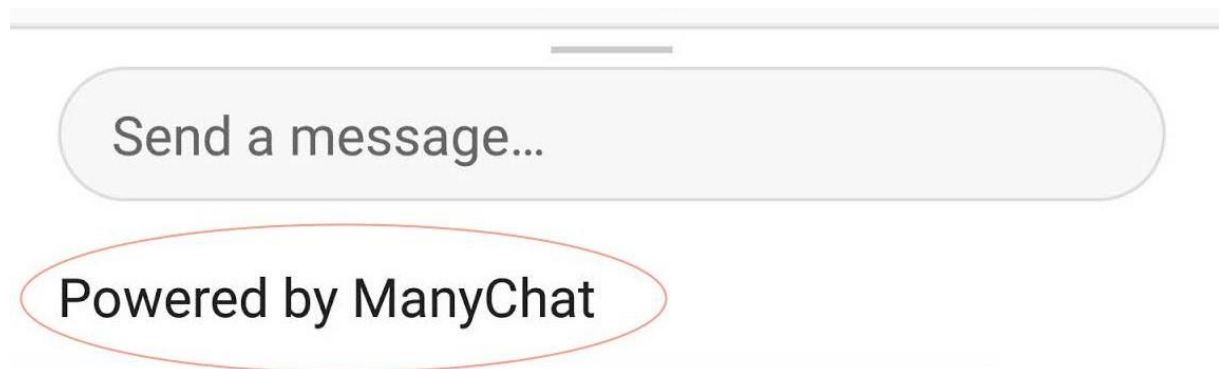


Figure 1 Powered by ManyChat (Source : Author)

1.2.2 Chatfuel

Chatfuel⁹ was founded in 2015 as a tool to quickly create bots. Initially, the service worked only with Telegram, and later Facebook Messenger was connected. Chatfuel is completely free. You get access to all functions, but the main menu of the bot will always display the link

⁸ <https://manychat.com>

⁹ <https://chatfuel.com>

"Powered by Chatfuel". To remove this branding, you need to go to one of the paid plans. Price depends on the maximum number of users.

1.2.3 Conclusion

Platform	Multilanguage	Easy to admin	Standalone	Price
Chatfuel	Yes	Poor	No	Depends on subscribers
ManyChat	Additional module	Good	No	Start from 10\$

Table 1 Compare ManyChat and Chatfuel (Source: Author)

For the thesis tasks, these platforms are not suitable for several reasons. They are not entirely free, free planes have restrictions. If you use a free plan, there will always be a message about what platform the chat is working on (Figure 1), which is not very good, it turns out we will advertise a third-party company. Language support is based on plug-ins only and requires additional costs and knowledge. A large number of settings and extensive functionality of these platforms can cause inconvenience of use for a regular user.

Our solution is more optimal. It includes only the necessary functions for our work. It does not advertise other brands. It can be used on modern hosting, which does not require additional costs. Since this is our personal bot, in the future we will be able to connect the additional functions that we need. We completely control the work of the bot and do not depend on third-party developers.

1.3 Tools and technologies

In the following subsection the author describes tools and technologies, which were chosen for the development.

1.3.1 Node.JS

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write

command line tools and for server-side scripting (Node.JS 2019). According to statistics from Stackoverflow¹⁰ Node.JS framework won the first place in the “other frameworks” category.

As Facebook API manual has examples for Node.JS, the author has chosen this technology for the bot development.

1.3.2 JavaScript

JavaScript is lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. (Mozilla 2018)

Since we have chosen Node.JS technology, we are already using JavaScript language. Also, JavaScript is most popular language in the world for the front-end development.

1.3.3 PHP

For the back-end of the application the author has chosen PHP. PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world (PHP 2019). For example, Yahoo, Wikipedia and Wordpress web applications are built on PHP.

For the administration console development, the author has chosen PHP, because PHP projects are very easy to scale, and PHP has built-in functions for working with files. Also, projects for PHP can be moved with little effort from one hosting to another, which allows you to quickly deploy them. Additionally, PHP is sufficiently secure and stable, and the author already has previous experience with it.

1.3.4 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of

¹⁰ <https://insights.stackoverflow.com/survey/2019#technology--other-frameworks-libraries-and-tools>

versatility and extensibility, jQuery has changed the way that millions of people write JavaScript (jQuery 2019)

jQuery is an additional tool which helps developers to create modern, useful and fast user interface for their web projects.

1.3.5 HTML and CSS

HTML and CSS are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications. (HTML and CSS, 2019)

1.3.6 Skeleton

Skeleton is lightweight CSS framework. Skeleton only styles a handful of standard HTML elements and includes a grid. (Skeleton 2019)

Skeleton was chosen mostly because it is very light, only about 400 lines of code, and it has a sufficient CSS grid system.

1.3.7 Microsoft Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with a built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). (Visual studio code 2019)

The author has chosen this editor, because he thinks that it is the best free code editor with many extensions for comfortable code writing.

1.3.8 XAMPP

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use. (XAMPP 2019)

XAMPP is a way to install a web server, database and PHP from the box. It includes a most popular web server Apache, which is installed by default in many hosting servers.

1.3.9 ngrok

ngrok is a multiplatform tunneling, reverse proxy software that establishes secure tunnels from a public endpoint, such as Internet, to a locally running network service, while capturing all traffic for detailed inspection and replay (ngrok 2019).

This service was chosen to create a secure access to the author's local web server from the Internet. It was required for the configuration of the Facebook webhook.

2 OVERVIEW OF THE BOT

In the following section the author describes the proposed bot system, including the data structure and requirements.

2.1 Requirements

2.1.1 Functional

The bot should be able to respond to client's request from Facebook page and Facebook Messenger application.

The bot should be able to respond in client's language such as Estonian, English or Russian.

The bot should provide a commands selection panel.

The bot should not allow entering text and media data.

The bot should provide a graphical a la carte menu by courses.

The bot should provide a graphical list of dishes for each course of the a la carte menu.

The bot should provide a text block about each dish, which includes the following data:

- Name of the dish.
- Ingredients of the dish.
- Price of the dish.
- Optional comment, which notifies the client about whether the dish is Vegan, Vegetarian, Gluten-free or Lactose-free.

2.1.2 Non-functional

The bot response time to client's request should be less than 10 seconds.

The bot should recognize requests from Facebook API.

The bot should generate right requests to Facebook API.

The bot should determine client's language based on client's Facebook page language settings or client's Facebook Messenger application language settings.

The bot should be able to read remote JSON files.

2.2 The proposed system

The proposed system is a bot that will get and process commands received from Facebook messenger platform. It will also generate responses to received commands. Responses will be generated based on data which is stored in JSON format in files on a remote server.

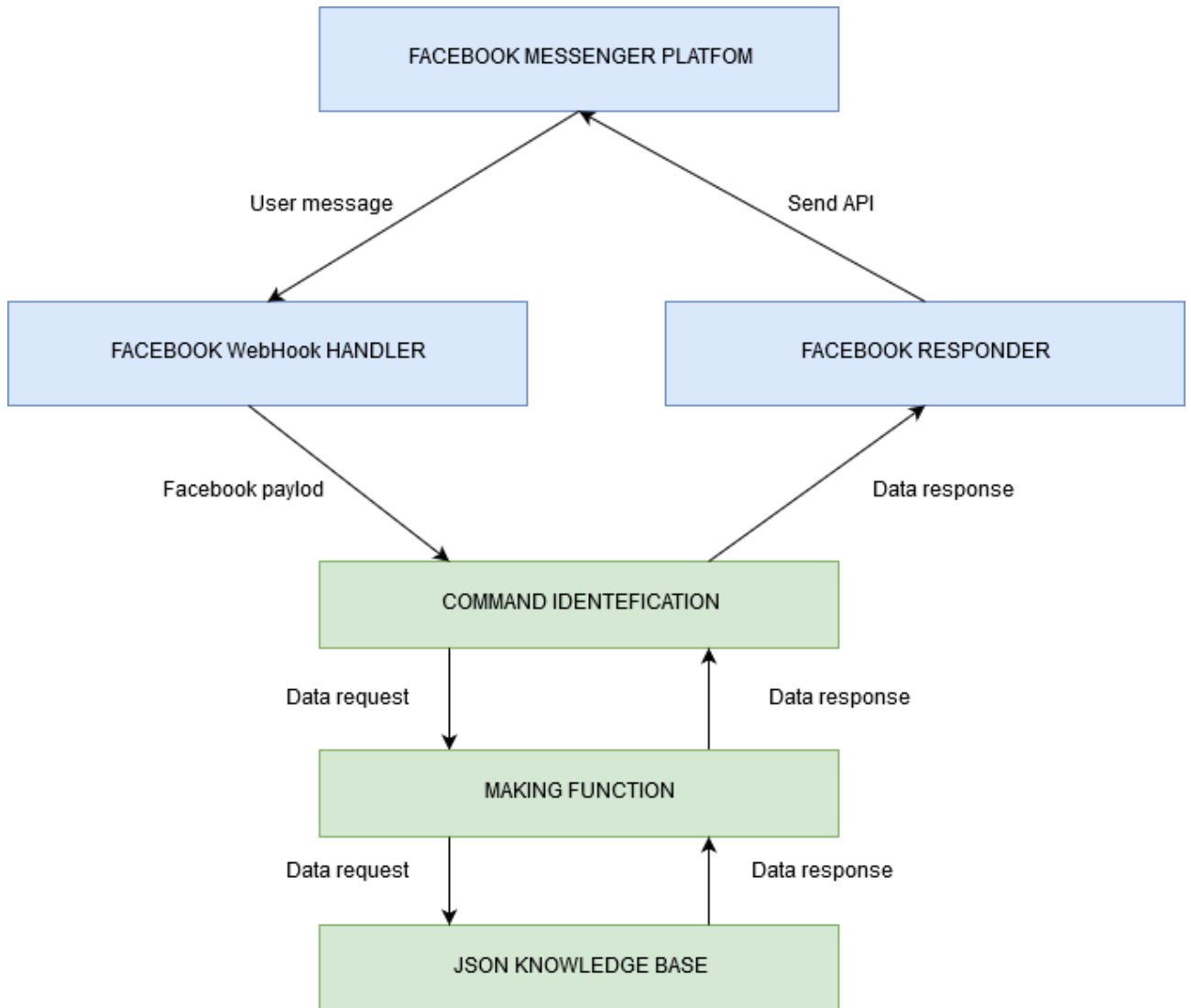


Figure 2 Architecture diagram (Source: author)

Upon receiving Facebook payload, a command identifier will launch a right function to generate the response. After generating the response, a command identifier will send this data to the Facebook API, after that the end user will receive a message.

2.3 Modular description

With the proposed architecture, the author has divided the whole implementation into three modules.

2.3.1 Command identification module

Since the bot will only work with preset commands, it will not give a possibility to enter any text by a user. The task of this module is to determine which function needs to be run to generate a response. Since we will not have many commands to solve this problem, the author uses the switch-case statement.

Also, this module sends a generated response to Facebook API.

2.3.2 Making function module

This executable module is responsible for loading and processing data in JSON format.

This is the central module of our bot that links the Command identification module and the JSON Knowledge base. This module is also responsible for the language in which the response will be generated.

Functions are generating response for:

- A Group of dishes like starters, soups, main courses and desserts.
- The whole menu.
- The lunch of the day.

Each of the responses has three language presets.

2.3.3 JSON knowledge base

A set of JSON files that are used to store information about dishes. This storage strategy was chosen because of the speed of reading. The amount of information is very small, and it does not make sense to use SQL databases. Files are stored on a remote server where the bot's administration console is located. It is a good and secure way to manipulate with data from administration console.

Knowledge base structure:

- starters.json – stored all data about each dish of this category in three languages.
- soup.json – stored all data about each dish of this category in three languages.
- maincourse.json – stored all data about each dish of this category in three languages.
- desserts.json – stored all data about each dish of this category in three languages.
- lunch.json – stored all data about the lunch of the day in three languages.

3 OVERVIEW OF THE BOT'S ADMINISTRATION CONSOLE

In the following section the author writes down the bot's administration console requirements and describes the administration console, including a data management system that comes from the user to the data storage JSON files.

3.1 Requirements

3.1.1 Functional

Administration console should be accessible to the authorized user only.

Administration console should allow to add the lunch of the day.

Administration console should automatically replace the lunch of the day if a new is added.

Administration console should allow to add a new dish to each course.

Administration console should allow to delete a dish from each course.

3.1.2 Non-functional

Administration console should be working in following web browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera.

Administration console load time should be less than 5 seconds.

Administration console user interface should only contain minimal possible interface elements to simplify its use.

Administration console user interface should be intuitively understandable for its user.

Administration console user password should not be stored in an unencrypted form.

Administration console should only have one user.

Administration console session lifetime should be 30 minutes.

Administration console should be storing data in JSON format.

Administration console should be able to create JSON files.

Administration console should be able to read JSON files.

Administration console should be able to write JSON files.

3.2 Administrator rights

Administrator should be able to add and remove data from JSON data storage files.

Administrator's activities:

- Replace the lunch of the day in the JSON file.
- Add a new dish to the JSON file.
- Remove the selected dish from the JSON file.

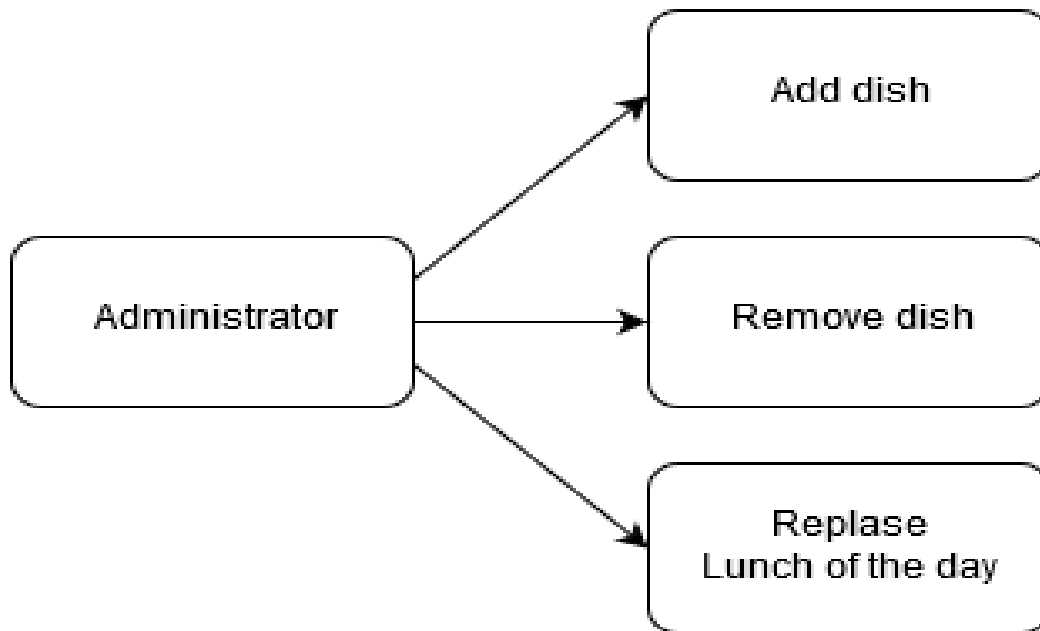


Figure 3 Administrator access level diagram (Source: author)

3.3 Security

Because this is a web application, security is very important. To ensure the security of the application the author uses a variety of security methods.

3.3.1 Log in

Everyone who wants to access the system should know the right password.

3.3.2 Password standard

The length of the user password is at least 8 characters, which includes at least one lowercase latter, one uppercase letter and at least one number.

3.3.3 Automatic Log out

The application automatically logs out after a certain time.

3.4 Back-end

This is the part of the system that will process the entered data by the user and convert it to JSON. It is also responsible for recording and deleting data from JSON files which are used by the bot as a database. The security part is also handled by the back-end.

This part is written in PHP language and contains following files:

- parser.php – includes functions to handle operations with JSON files such as read, write and delete
- config.php – includes security layer and global variables
- index.php – includes authentication algorithm and security control algorithm

3.5 Front-end

This is a part of the system that is responsible for the user interface. Provides fields and data entry forms. It checks the correctness of the entered data and ensures its transfer to the back-end.

This part is written in HTML, CSS and JavaScript languages. Part of the code is integrated into the back-end files.

4 THE RESULT

In the following section the author presents the result of this thesis: the bot and the administration console.

4.1 The final system diagram of the developed system

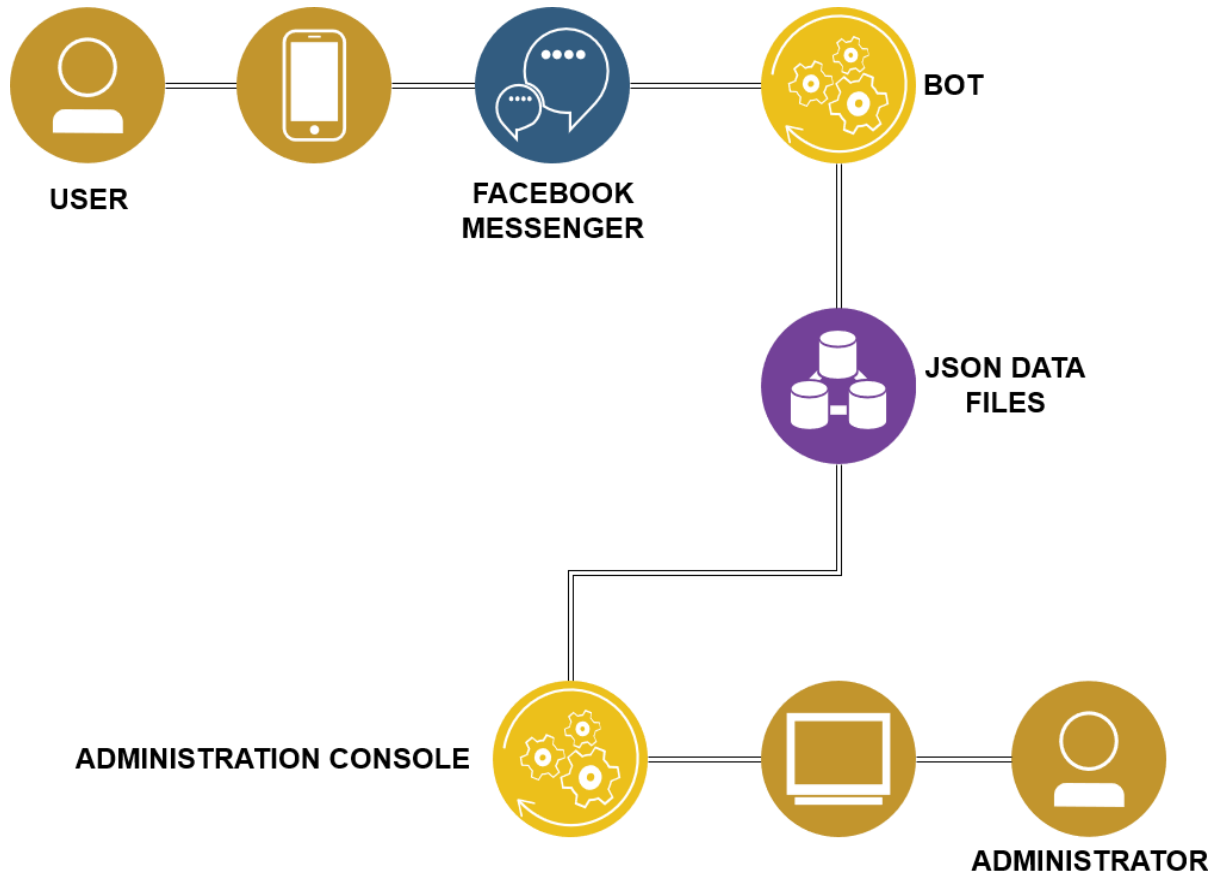


Figure 4 The final system diagram of the developed system (Source: Author)

This diagram visually shows how the developed system works. For example, a client uses a mobile device to open the Facebook Messenger app. When the app opens, the client proceeds to the café “Muna” chat channel and selects a command in the bottom part of the chat window or from the previously received bot replies still available in the chat window. After that, the command is processed by the bot. The bot selects proper data from the JSON files and sends it back to the Facebook Messenger. On the other side of the system, the administrator uses a web browser to get access to the bot’s administration console and enters or edits data, which is stored to the JSON files by the console.

4.2 Bot views

The following is how the bot looks like in the Android version of the Facebook Messenger. All screens are made on the testing Facebook page and not the real business page.

4.2.1 Get started and welcome message views

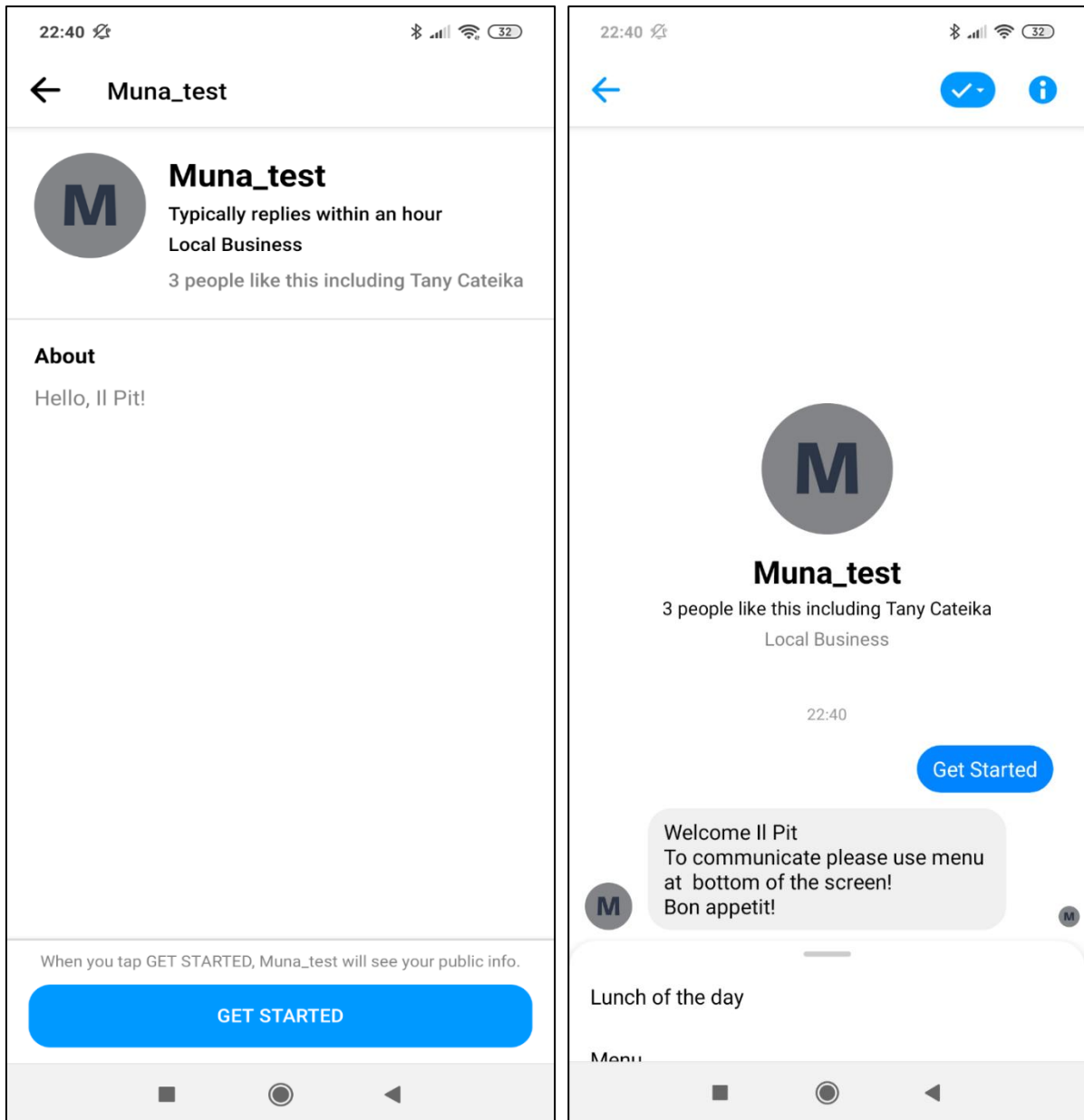


Figure 5 Get started and welcome message views (Source: author)

On the left side of Figure 5 a reader can see a get started view of the bot. On the right side of Figure 5 a reader can see the welcome message, which language is defined by the language of the user profile. Users only see both views once, when they first time start communicating with the bot.

4.2.2 Lunch of the day view

On the Figure 6 a reader can see a lunch of the day text message. It is shown when a user presses the Lunch of the day option.

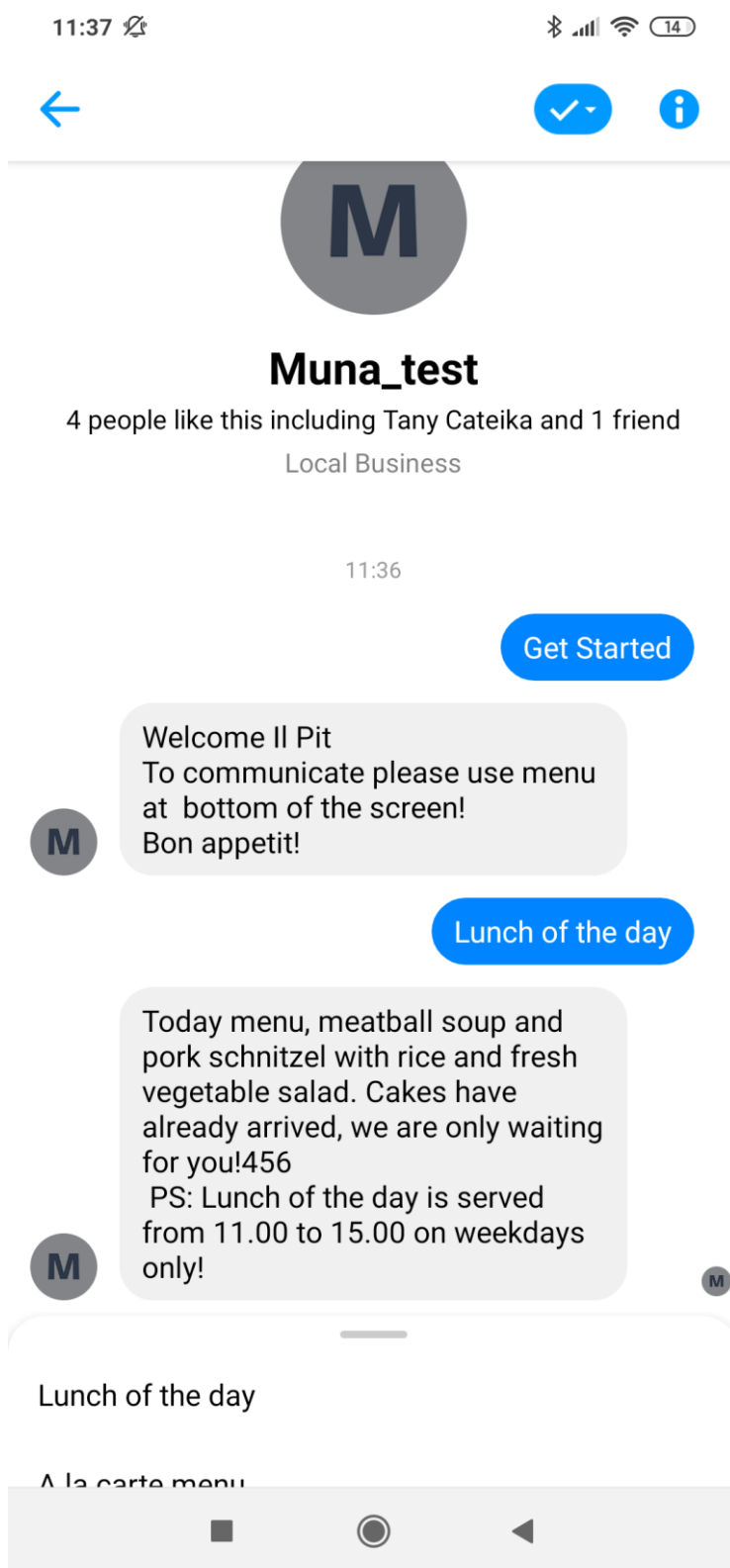


Figure 6 Lunch of the day view (Source: author)

4.2.3 A la carte menu view

On the Figure 7 a reader can see the list of courses: starters, soups, main courses and desserts. Each course has a view button to show the list of dishes of the course.

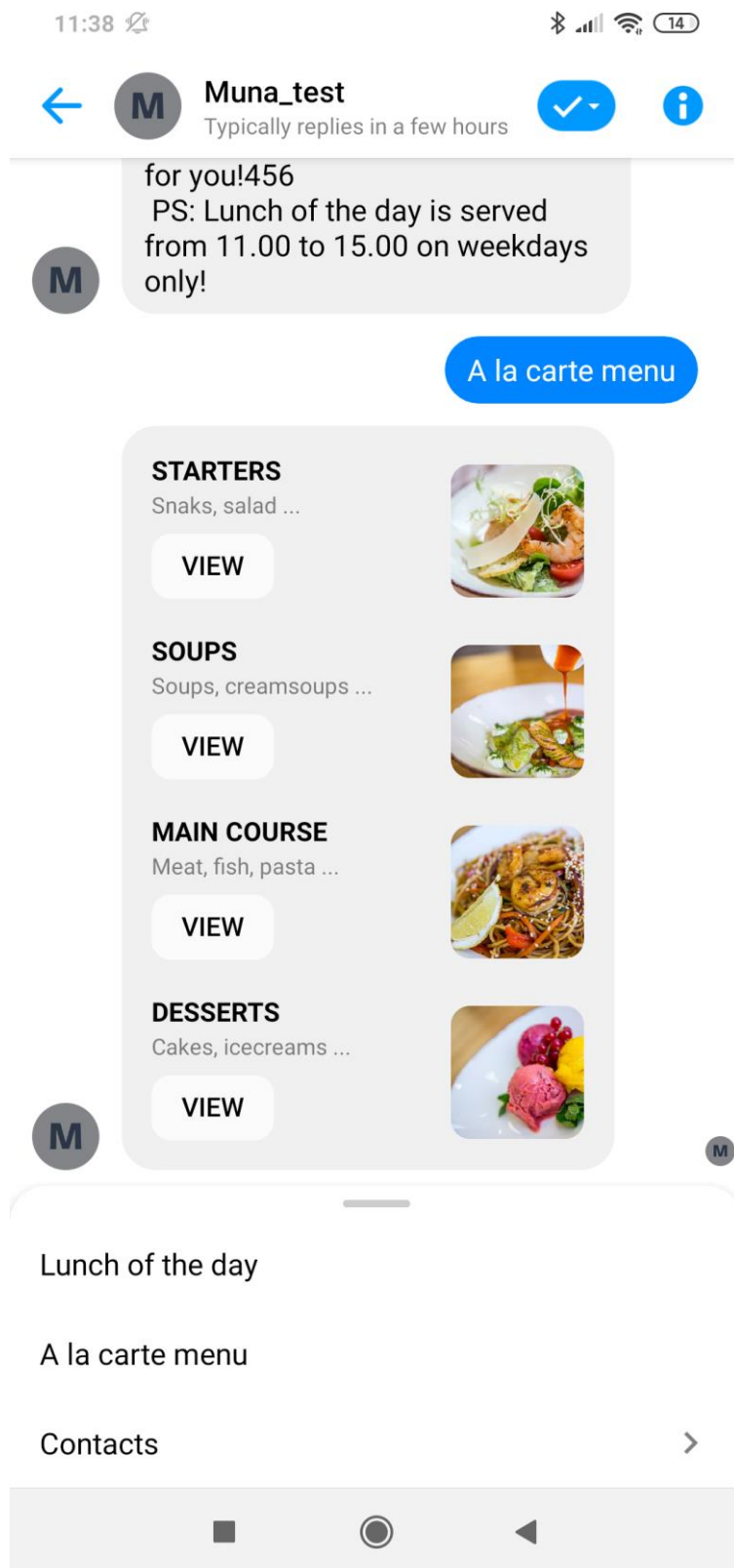


Figure 7 A la carte menu view (Source: author)

4.2.4 Top-level commands view

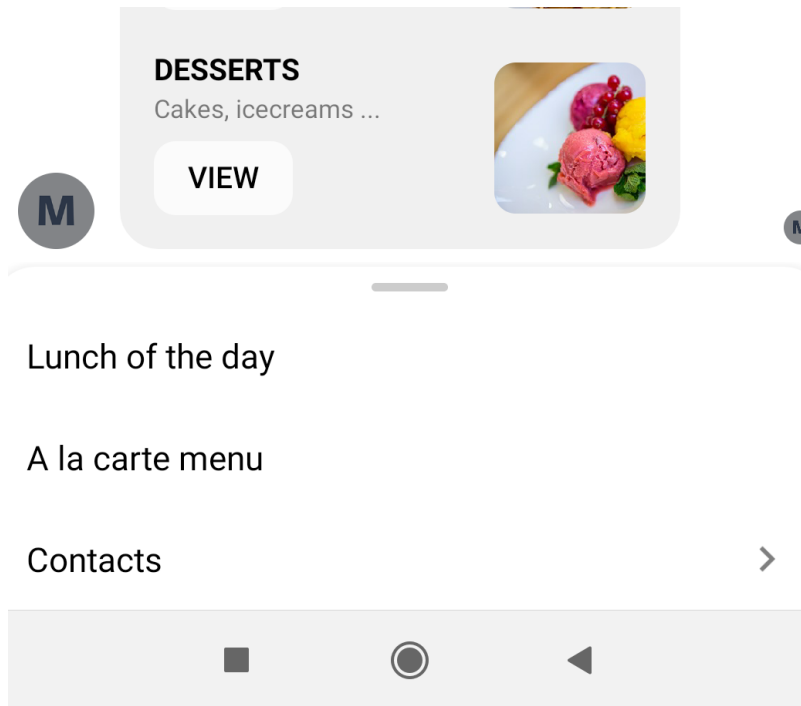


Figure 8 Top-level commands view (Source: author)

On the Figure 8 a reader can see a static menu of top-level commands of the bot, which has three commands, because only three elements are allowed by the Facebook API (see appendix 3). As the bot has more than three commands, the author had to use a second-level command view for the Contacts command, which is shown on the Figure 9.

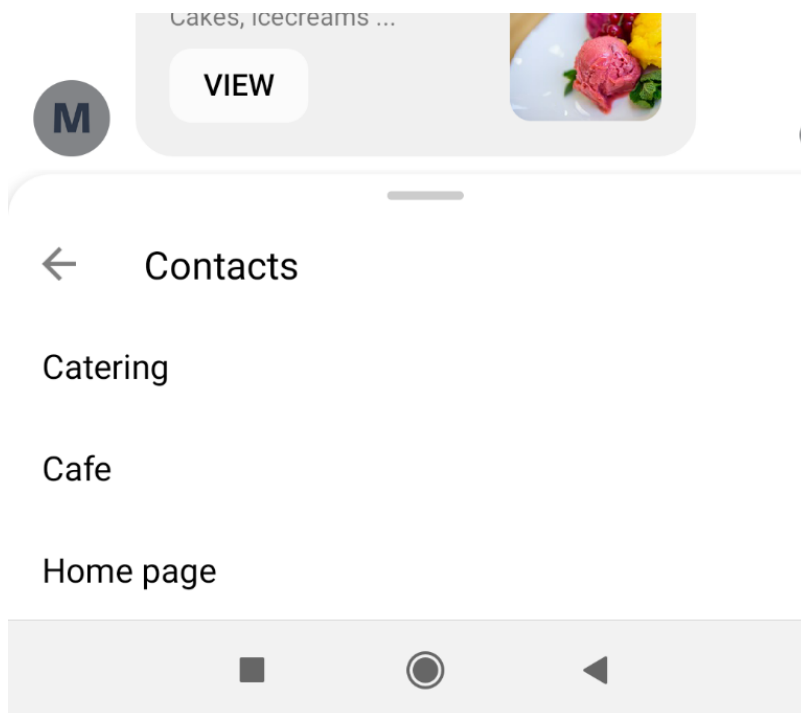


Figure 9 Second-level command view for the Contact command (Source: author)

This view contains catering contacts, café contacts and café web page link.

4.2.5 Dishes view

If users press one of the “View” buttons in the A la carte menu view, they get a list of corresponding dishes. For example, if the button “View” under the “Soups” section is pressed, the user will get a list of soups, as shown on the Figure 10. Each dish has a name, a list of components and a price. Some of the dishes may also have a comment, which contain notes like “Vegan” or “Gluten free” etc.

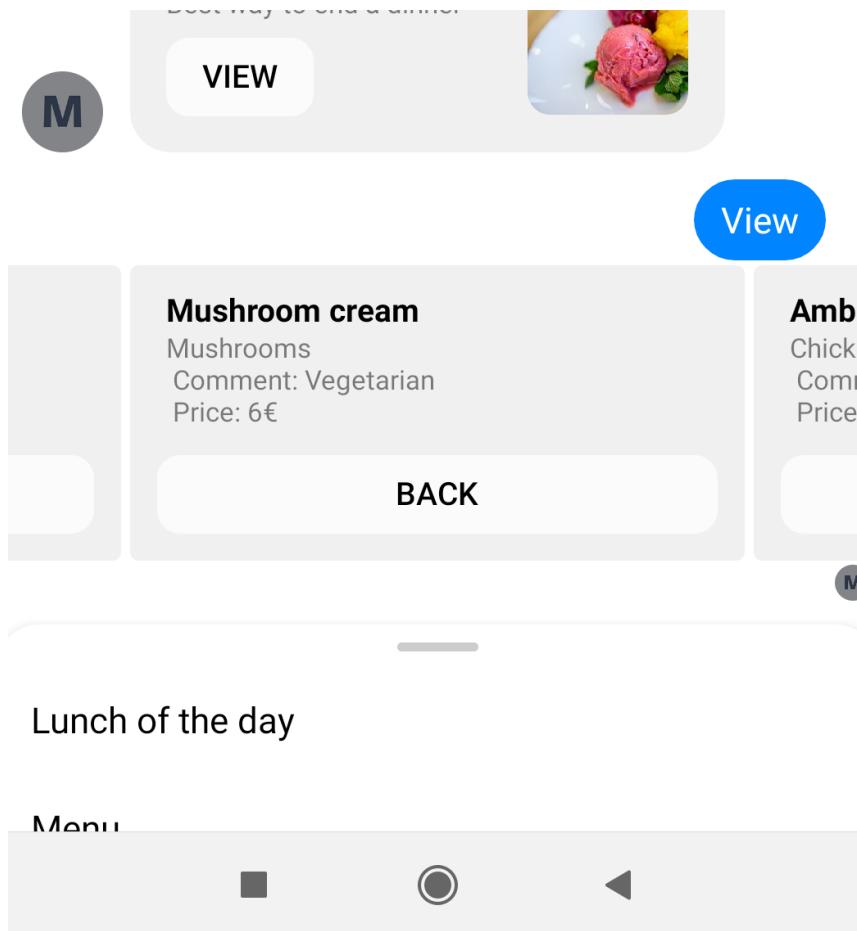


Figure 10 Dishes view (Source: author)

Other “View” buttons under other sections behave similarly.

4.3 Administration console views

Next the author shows how the bot’s administration console looks like.

4.3.1 Log in

On the Figure 11 a reader can see a login page.

The login page is needed for the security reasons to control access to the administration console. Only one user, the café manager, has access to the console. For this reason, a username is not required.

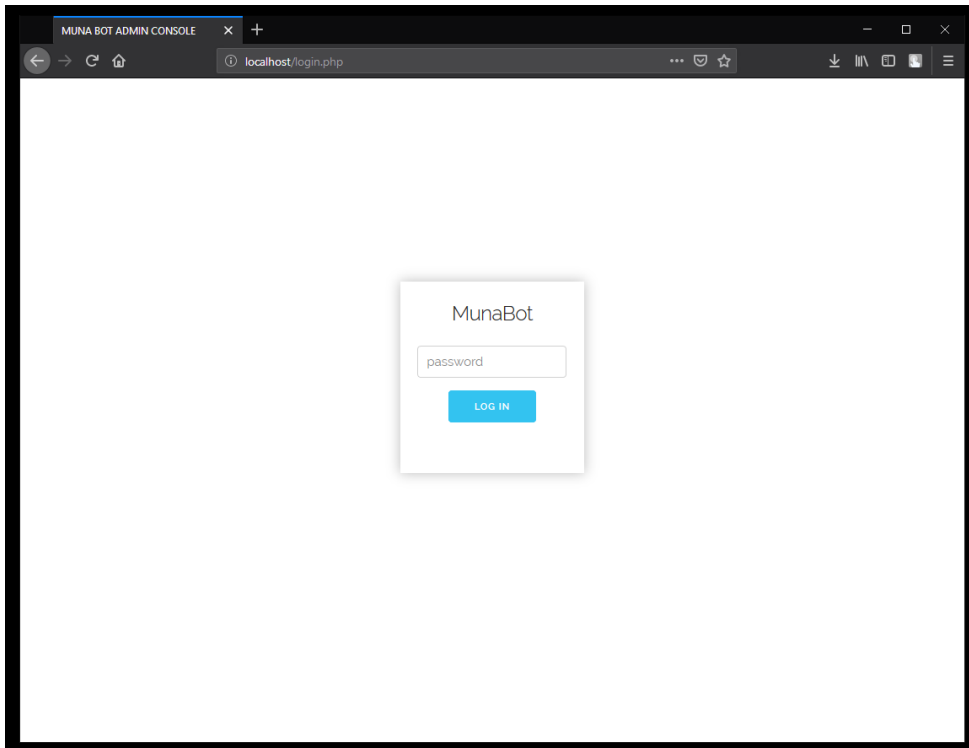


Figure 11 Log-in view (Source: author)

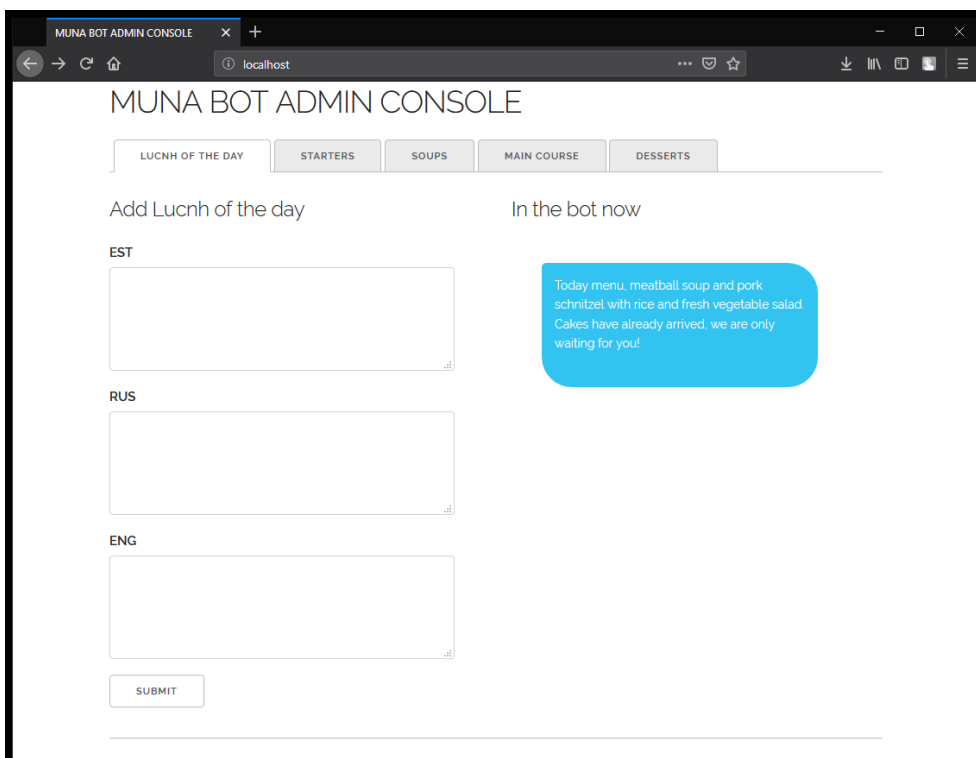


Figure 12 Lunch view (Source: author)

4.3.2 Lunch view

On the Figure 12 a reader can see the lunch of the day page. This is the first page which is displayed after a successful authorization from the login page.

Lunch of the day page has three text area fields on the left side. Each field has a language label, according to ISO 639-2 Code¹¹ standard. All three fields are required. Empty fields checking is implemented through HTML “required” tag attributes. On the right side, in the blue field, the administrator can see the current lunch of the day, which loaded from the lunch.json file.

4.3.3 Navigation tabs bar view

On the Figure 13 a reader can see the navigation bar tabs.



Figure 13 Navigation bar tabs view (Source: author)

Beside lunch of the day page the navigation bar also contains tabs for a la carte menu courses and allows the administrator to switch between courses and lunch of the day tabs. The author of this thesis believes that this solution satisfies the requirements for simple and convenient user interface for the administration console. Since the number of tabs is not big and pages do not contain large data amount, this is a fast and reasonable solution.

4.3.4 Course view

On the Figure 14 a reader can see the course view. For example, the soups course tab was selected, the same way as it was selected as an example in the bot description above.

¹¹ http://www.loc.gov/standards/iso639-2/php/code_list.php

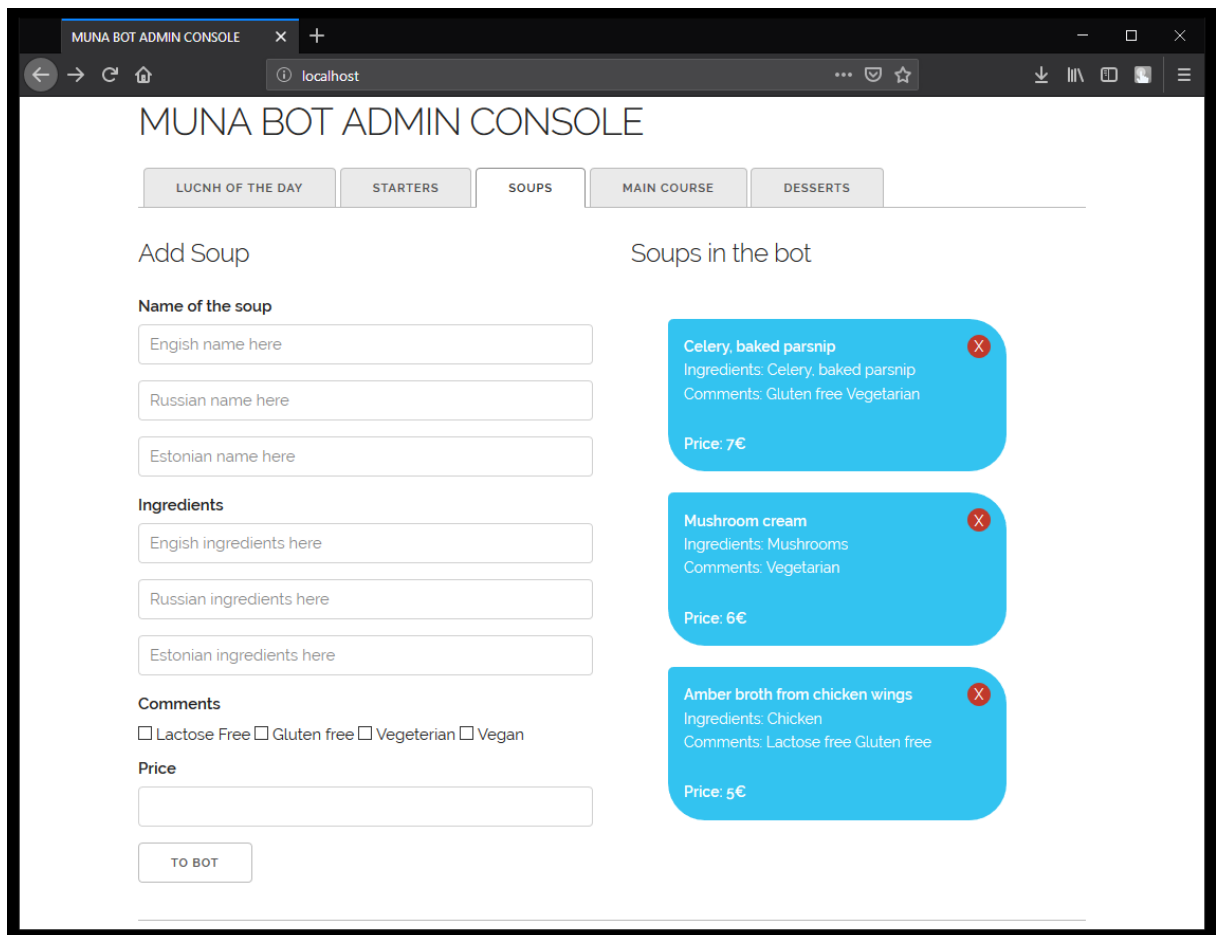


Figure 14 Course view (Source: author)

The soup course tab view, like any other course tab view, is visually divided into 2 parts, left and right. On the left side there is a form for adding a dish to the course. The right side shows a list of dishes of this course, which the bot displays on the corresponding request.

The form for adding a dish to the course have four field groups:

- The name of the dish group with text fields.
- The ingredients group with text fields.
- The comments group with checkboxes.
- The price text field.

The first and the second groups include three fields each to duplicate data in Estonian, English and Russian languages. To hint which field is for which language, the author used HTML placeholder attributes.

The third group is optional and contains checkboxes to add additional information about the dish to inform clients of the café about “Gluten free”, “Lactose Free”, “Vegetarian” or

“Vegan” dishes. The administrator does not need to enter this information in three languages, because translation is made automatically in the background by the administration console.

The last field is the price, which only accepts numerical input. That field is required and does not need to be translated.

In the right side of the page one can see all dishes of the selected course, which are currently available in the bot. Each dish is presented as a separate blue box for easy visual navigation between dishes. Also, each box has a remove button, implemented as small red circle with a cross, in the top-right corner of the box. This button is needed for easy removal of the dish from the course.

4.4 Testing

In the following subsection the author describes the testing results. For this project the black box testing method was chosen.

Black box testing plays a significant role in the software testing, as it aids in overall functionality validation of the system. Black box testing is done based on customer requirements — so any incomplete or unpredictable requirements can be easily identified, and it can be addressed later. Black box testing is done based on end user perspective. The main importance of black box testing is that it handles both valid and invalid inputs from customer’s perspective. (Murnane, Reed 2001)

Black box testing is done from the end-user point of view and can help detect inaccuracies and contradictions in the specification. The tester does not need to know programming languages and be aware of the program implementation.

4.5 The bot testing

The test results were transmitted verbally in real-time to the author. The test time took from 3 to 5 minutes on each device or browser.

4.5.1 Mobile

The bot functionality was tested on Android and iOS mobile operation systems.

4.5.2 Desktop

The bot functionality was tested in Mozilla Firefox, Google Chrome, Microsoft Edge and Safari browsers on Windows 10 and Mac OSX operation systems.

4.6 The administration console testing

The test results were transmitted verbally in real-time to the author. The test time took from 5 to 10 minutes on each device and browser.

The administration console functionality was tested in Mozilla Firefox, Google Chrome, Microsoft Edge and Safari browsers in Windows 10 and Mac OSX operation systems.

CONCLUSION

It is a widely known fact that social networks, especially Facebook, are very popular in our time. What is perhaps a somewhat lesser known fact that social networks are not only a place for information exchange, but also a platform for business as described in the section 1.1.

This platform allows everyone to promote their business and automate the process of communicating with customers, giving them the opportunity to do it quickly and at any time. One of the possible solutions for this is an auto-replay application. This solution was proposed for the café “Muna” by the author of this thesis.

The search among available solutions, as presented in the section 1.2, lead the author to the conclusion that developing a separate application is the best way to address the business needs of the café “Muna”.

To resolve the problem of this thesis, the author used modern and open source technologies such as JavaScript, Node.JS and PHP as described in the section 1.3, and his programming skills obtained during studying. All this eventually helped the author to achieve the aim of this thesis.

The result of this thesis is an automatic reply system, which includes the bot application and the bot’s administration console, implemented as a web application. The system meets all functional and non-functional requirements described in section 2.1 for the bot and in section 3.1 for the bot’s administration console. The requirements were based on the interview with the café “Muna” manager. Transcript of the interview is presented in the appendix 4.

Bots are quick and easy to handle, which have a potential to increase customer satisfaction. It is now becoming a new business communication channel to attract new customers. In addition, a proper automation of business communication may increase efficiency and reduce business costs.

In the future, the author intends to add additional features to the bot such as automatic post publisher for café Facebook page and table reservation system.

SUMMARY IN ESTONIAN

Antud töö pealkiri on „Automaatse vastamise vestlusroboti loomine kohviku „Muna“ Facebook’i lehekülje jaoks“.

Laialdaselt tuntud faktiks on sotsiaalvõrkude, eriti Facebook’i, populaarsus praegusel ajal. Ehk mõnevõrra vähem tuntud faktiks on see, et sotsiaalvõrgud ei ole vaid koht informatsiooni vahetamiseks, vaid ka äriplatvorm, nagu kirjeldatud alapeatükis 1.1.

See platvorm võimaldab igapäev reklaamida oma äri ja automatiseerida suhtlusprotsessi klientidega, andes neile võimaluse teha seda kiiresti ja suvalisel ajal. Üheks võimalikuks lahenduseks selle jaoks on automaatse vastamise rakendus. Just selle lahenduse pakkus selle töö autor kohviku „Muna“ jaoks.

Otsing olemasolevate lahenduste seas, nagu näidatud alapeatükis 1.2, viis autori järelduseni, et parimaks lahenduseks kohviku „Muna“ ärivajaduste rahuldamiseks oleks ikkagi eraldi rakenduse välja töötamine.

Selle töö probleemi lahendamiseks autor kasutas kaasaegseid ja vaba lähtekoodiga tehnoloogiaid, nagu JavaScript, Node.JS ja PHP, mis oli kirjeldatud alapeatükis 1.3, ning oma programmeerimise oskusi, mis olid omandatud õppimise ajal. Kõik see aitas autorit edukalt saavutada antud töö eesmärgi.

Selle töö tulemuseks on automaatse vastamise süsteem, mis sisaldab vestlusroboti rakendust ja veebirakendusena arendatud administratiivset konsooli vestlusroboti jaoks. Süsteem täidab kõiki funktsionaalseid ja mitte-funktsionaalseid nõudeid vestlusroboti jaoks, mis olid kirja pandud alapeatükis 2.1, ja administratiivse konsooli jaoks, mis olid kirja pandud alapeatükis 3.1. Kõik need nõuded olid rajatud intervjuule kohviku „Muna“ tegevjuhiga. Selle intervjuu tekst on toodud lisa 4.

Vestlusrobotid on kiired ja kergesti hallatavad, mis loob eeldusi klientide rahulolu suurendamiseks. Vestlusrobotid on muutumas uueks äriliseks suhtluskanaliks uute klientide meelitamiseks. Lisaks sellele, hästi automatiseeritud ärisuhtlemine võib suurendada efektiivsust ja vähendada äriulusid.

Tulevikus autor plaanib lisada oma rakendusele selliseid võimalusi nagu automaatne uudiste postitamine kohviku „Muna“ Facebooki leheküljele ja laudade reserveerimine.

REFERENCES

Cambridge Dictionary 2019. Meaning of “Bot” in the English Dictionary. Cambridge University Press. Available at <https://dictionary.cambridge.org/us/dictionary/english/bot>, accessed April 25, 2019.

Cambridge Dictionary 2019. Meaning of “Social network” in the English Dictionary. Cambridge University Press. Available at <https://dictionary.cambridge.org/us/dictionary/english/social-network>, accessed April 25, 2019.

Davis, Troy 2019 “What is a WebHook?”. Available at <https://webhooks.pbworks.com/w/page/13385124/FrontPage> , April 25, 2019

Facebook IQ, 1. 2019 Available at <https://www.facebook.com/iq/insights-to-go/48-48-of-local-businessesa-local-business-is-defined-as-one-that-is-not-part-of-a-chain-or-franchise-and-isnt-purely-online-but-has-fewer-than-five-physical-stores-they-may-also-travel-to-their-customers-at-home-or-at-work-across-all-countries-surveyed-have-started-optimizing-their-marketing-for-mobile-devices/>, accessed April 25, 2019

Facebook IQ,2. 2019 Available at <https://www.facebook.com/iq/insights-to-go/6-in-10-6-in-1064-local-businessesa-local-business-is-defined-as-one-that-is-not-part-of-a-chain-or-franchise-and-isnt-purely-online-but-has-fewer-than-five-physical-stores-they-may-also-travel-to-their-customers-at-home-or-at-work-across-all-countries-studied-say-having-an-online-presence-is-important-for-their-long-term-success/> , accessed April 25, 2019

HTML and CSS, 2019. Available at <https://www.w3.org/standards/webdesign/htmlcss>, accessed April 25, 2019

jQuery 2019, Available at <https://jquery.com/>, accessed April 28, 2019.

JSON 2019, Available at <https://www.json.org/> , accessed April 25, 2019.

Mozilla 2019, Available at https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript, accessed April 28, 2019

Murnane, Taflin and Reed, Karl 2001 “On the effectiveness of mutation analysis as a black box testing technique”. Available at http://homepage.cs.latrobe.edu.au/kreed/Publications_and_Keynotes_files/MurnaneReedASWEC2001.pdf <https://www.w3.org/standards/webdesign/htmlcss>, accessed May 15, 2019

Nealon, Gary. 2018 “Using Facebook Messenger And Chatbots To Grow Your Audience”. Available at <https://www.forbes.com/sites/forbesagencycouncil/2018/06/04/using-facebook-messenger-and-chatbots-to-grow-your-audience/> , accessed April 29, 2019

ngrok, 2019. Available at <https://ngrok.com/docs>, accessed April 25, 2019

PHP 2019, Available at <https://secure.php.net/>, accessed April 28, 2019.

Skeleton, 2019. Available at <http://getskeleton.com>, accessed April 25, 2019

Visual studio code 2019, Available at <https://code.visualstudio.com/>, accessed April 28, 2019.

W3Schools 2019. HTTP Request Methods. Available at https://www.w3schools.com/js/js_json.asp, accessed April 25, 2019

Wikipedia 2019. Facebook. Available at <https://en.wikipedia.org/wiki/Facebook> , accessed April 25, 2019

Wikipedia 2019. Internet Bot Available at https://en.wikipedia.org/wiki/Internet_bot , accessed April 25, 2019.

XAMPP 2019, Available at <https://www.apachefriends.org/ru/index.html>, accessed April 28, 2019.

Zumstein, Darius and Hundertmark, Sophie 2017 “Chatbots –An Interactive Technology for Personalized Communication, Transactions And Services”. Available at https://www.forbes.com/sites/forbesagencycouncil/2018/06/04/using-facebook-messenger-and-chatbots-to-grow-your-audience/https://www.researchgate.net/profile/Darius_Zumstein2/publication/322855718_Chatsbots_-_An_Interactive_Technology_for_Personalized_Communication_Transactions_and_Services/links/5a72ecde458515512076b406/Chatbots-An-Interactive-Technology-for-Personalized-Communication-Transactions-and-Services.pdf , accessed April 29, 2019

APPENDICES

APPENDIX 1 Desktop & Mobile Social Media Stats Europe

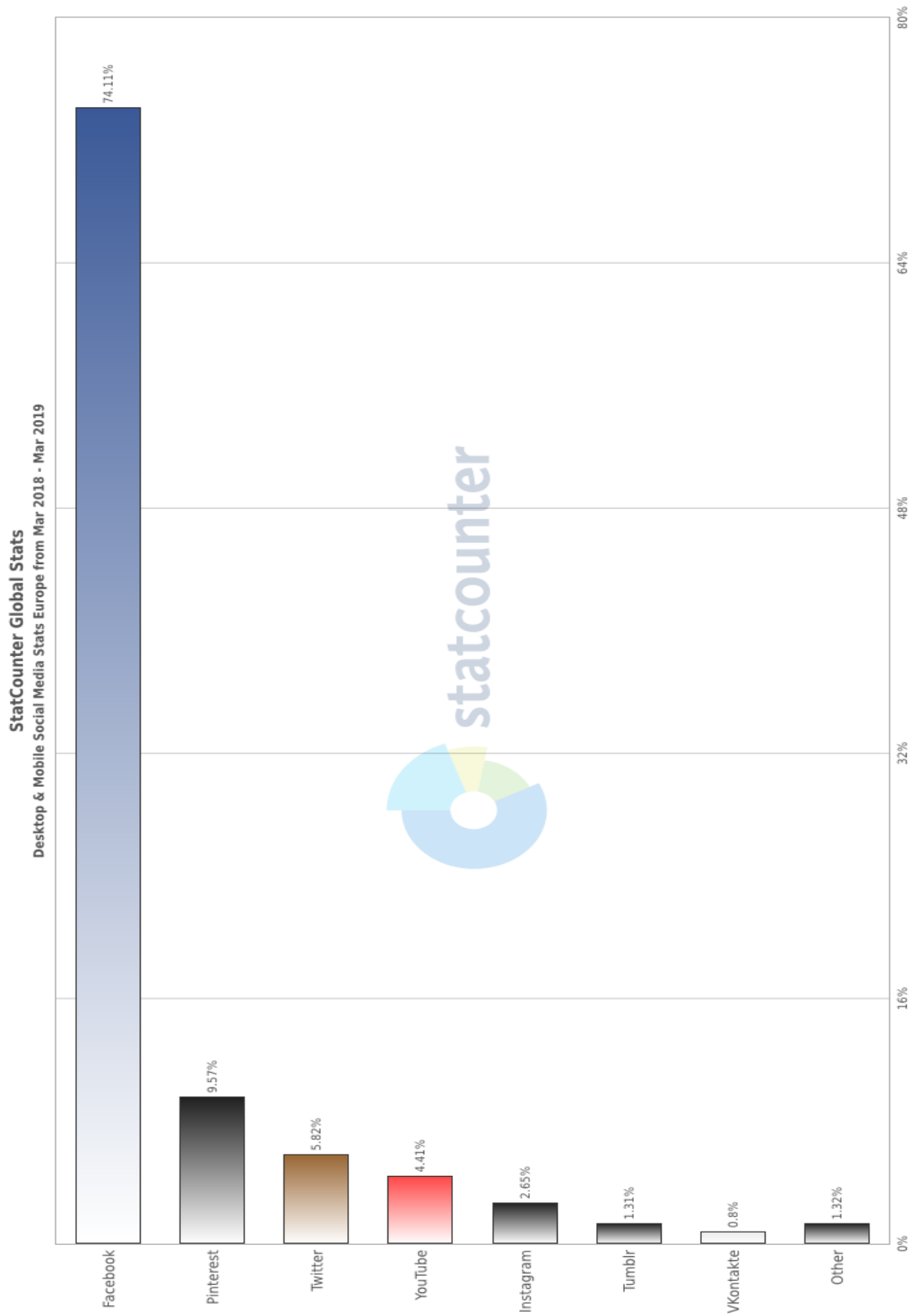


Figure 15 Desktop & Mobile Social Media Stats Europe (Source: gs.statscounter.com)

APPENDIX 2 Desktop & Mobile Social Media Stats Estonia

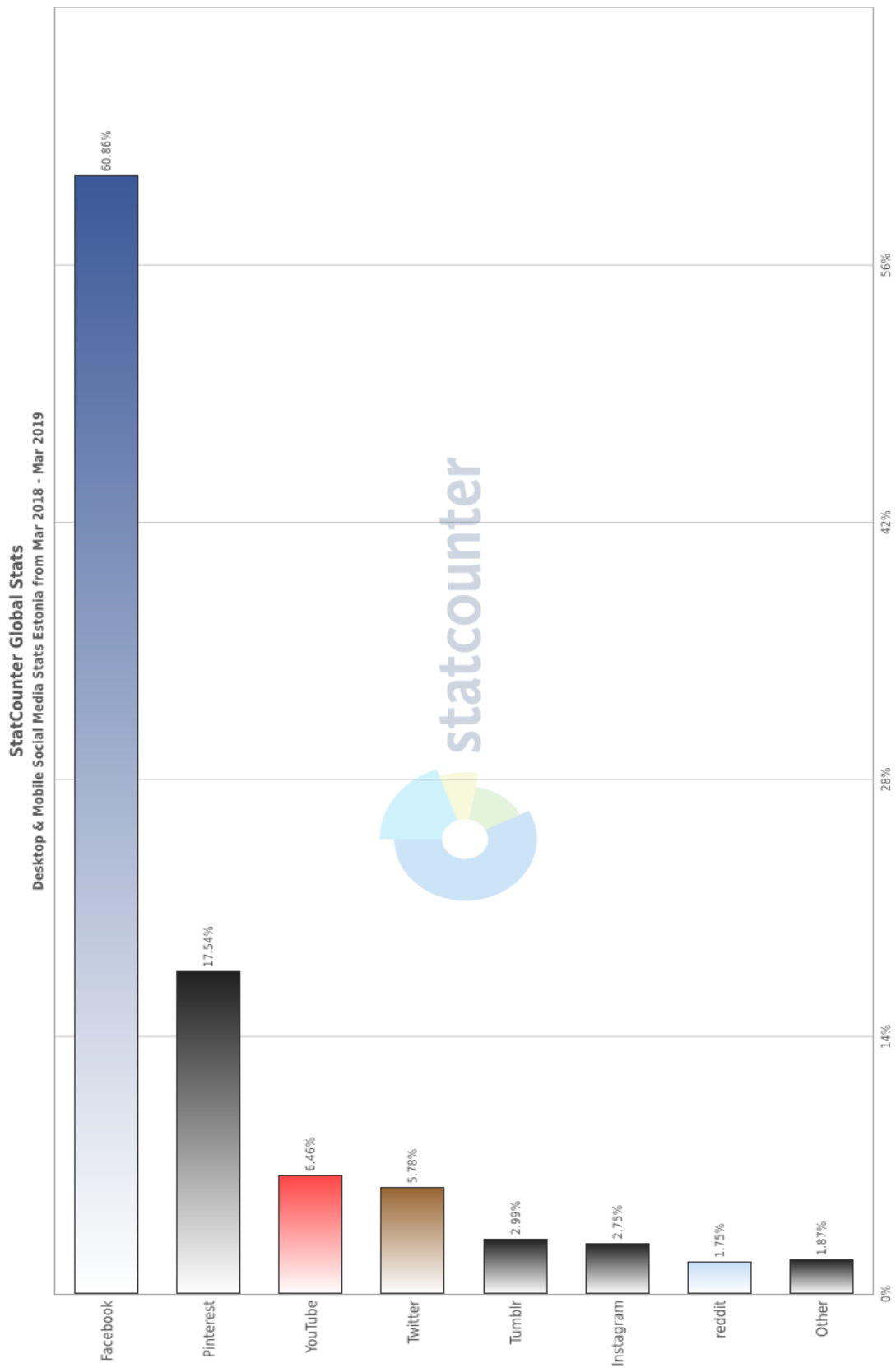


Figure 16 Desktop & Mobile Social Media Stats Estonia (Source: gs.statscounter.com)

APPENDIX 3 Facebook requirements

Webhook

To deploy a live webhook that can receive webhook events from the Messenger Platform, your code must be hosted on a public HTTPS server, which must have the following¹²:

- HTTPS support, Self-signed certificates are not supported.
- A valid SSL certificate.
- An open port that accepts GET and POST requests.

The Persistent Menu

An array of top-level menu items for the persistent menu. A maximum of 3 items is allowed. A maximum of two nested menus are supported¹³.

List template

The list template is a list of 2-4 structured items with an optional global button rendered at the bottom. Each item may contain a thumbnail image, title, subtitle, and one button¹⁴.

General template

The generic template is a simple structured message that includes a title, subtitle, image, and up to three buttons. You may also specify a `default_action` object that sets a URL that will be opened in the Messenger web-view when the template is tapped¹⁵.

Messenger

The person must be running Messenger v106 or above on iOS or Android¹⁶.

¹² <https://developers.facebook.com/docs/messenger-platform/getting-started/webhook-setup>

¹³ <https://developers.facebook.com/docs/messenger-platform/reference/messenger-profile-api/persistent-menu>

¹⁴ <https://developers.facebook.com/docs/messenger-platform/send-messages/template/list>

¹⁵ <https://developers.facebook.com/docs/messenger-platform/send-messages/template/generic>

¹⁶ <https://developers.facebook.com/docs/messenger-platform/send-messages/persistent-menu>

APPENDIX 4 Transcript of the Interview

Q: Do you use social networks to attract new customers to your café?

A: Yes, of course.

Q: What social networks do you use?

A: Facebook and Instagram.

Q: How many clients are trying to communicate with you via café Facebook page daily?

A: 3-10

Q: How quickly do you answer the Facebook page requests?

A: In an hour.

Q: What are the most popular languages you receive messages in?

A: Russian and Estonian, sometimes English.

Q: Which questions are the most popular in those messages?

A: Table reservation, catering, looking for job, menu.

Q: Do you want to automatize that channel?

A: Yes, of course.

Q: Which functionality do you want to see in the bot first?

A: Dishes menu, lunch of the day, contacts. I would prefer to receive official requests not through messenger, but to the café's e-mail. It would be very nice if the bot would be able to implement this strategy and sort those things out.

Q: What languages should the bot communicate with customers in?

A: Estonian, Russian and English.

Q: What functions of the bot do you consider in the future?

A: Table reservation.

APPENDIX 5 Source code

The source code can be found by following this link:

<https://github.com/IljaAndrejev/munabot>

NON-EXCLUSIVE LICENSE TO REPRODUCE THESIS

I, Ilja Andrejev (date of birth: 19.01.1982),

1. herewith grant the University of Tartu a free permit (non-exclusive license) to reproduce, for the purpose of preservation, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, “Creating an auto-reply Bot for Café Muna Facebook page” supervised by assistant Andre Säask.

2. I am aware of the fact that the author retains the right referred to in point 1.

3. This is to certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Narva, **22.05.2019**