

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж

“ДОПУСТИТИ ДО ЗАХИСТУ”
Завідувач кафедри

_____ Жуков І.А.

“ _____ ” _____ 2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня “МАГІСТР”

спеціальності 123 «Комп'ютерна інженерія»

освітньо-професійної програми «Комп'ютерні системи та мережі»

на тему: **“Організація зв'язку користувачів Salesforce з клієнтами”**

Виконавець: _____ Бурба М.А.

Керівник: _____ Іскренко Ю.Ю.

Нормоконтролер: _____ Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань
Бурба М.А.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

“PERMISSION TO DEFEND GRANTED”

The Head of the Department

_____ Zhukov I.A.

“ _____ ” _____ 2020

MASTER’S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: **“Organization of communication between Salesforce customers and clients”**

Completed by: _____ Burba M.A.

Supervisor: _____ Iskrenko Yu.Yu.

Standard’s Inspector: _____ Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних систем та мереж

Освітній ступінь: «Магістр»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерні системи та мережі»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри

Жуков І.А.

“ ” 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи

Бурби Максима Андрійовича

(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: “Організація зв'язку користувачів Salesforce з клієнтами”
затверджена наказом ректора від 25.09.2020 р. № 1793/ст
2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.
3. Вихідні дані до роботи (проекту): Salesforce CRM, Developer Edition
4. Зміст пояснювальної записки: Вступ, огляд теми, огляд сучасного стану CRM платформ з точки зору налагодження комунікаційних процесів їхніх користувачів з клієнтами, розробка та тестування пакету для Salesforce платформи, що дозволяє користувачам налагодити комунікаційний канал з використанням СМС, висновки по роботі.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: Матеріали представлені у вигляді презентації в Power Point.

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering

Department: Computer Systems and Networks

Educational Degree: “Master”

Specialty: 123 “Computer Engineering”

Educational-Professional Program: “Computer Systems and Networks”

“APPROVED BY”

The Head of the Department

_____ Zhukov I.A.
“ _____ ” _____ 2020 p.

Graduate Student’s Degree Thesis Assignment

_____ Burba Maksym Andriiovych

1. Thesis topic: “Organization of communication between Salesforce customers and clients” approved by the Rector’s order of 25.09.2020 p. № 1793/CT
2. Thesis to be completed between 01.10.2020 and 25.12.2020
3. Initial data for the project (thesis): *Salesforce CRM, Developer Edition.*
4. The content of the explanatory note (the list of problems to be considered): *Introduction, review of the topic, review of the current state of CRM platforms in terms of establishing communication processes of their users with customers, development and testing of a package for Salesforce platform that allows users to establish a communication channel using SMS, conclusions.*
5. The list of mandatory graphic materials: *Graphic materials are given in MS Power Point presentation.*

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи	1.10.20- 8.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи	9.10.20- 15.10.20	
3	Опрацювати теоретичний матеріал щодо налагодження зв'язку в CRM платформах та обрати стек технологій	16.10.20- 18.10.20	
4	Розробити програму та провести тестування з метою переконання в тому, що вона працює так, як очікувалось	19.10.20- 03.11.20	
5	Виконати аналіз результатів програми, розробити рекомендації щодо її використання та оформити пояснювальну записку	04.11.20- 12.12.20	
6	Оформити графічну частину записки та подати матеріали роботи на антиплагіатну перевірку матеріалів	13.12.20- 14.12.20	
7	Отримати рецензію та відгук керівника. Надати матеріали роботи на кафедру	15.12.20- 18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи _____ Іскренко Ю.Ю.
(підпис керівника)

Завдання прийняв до виконання _____ Бурба М.А.
(підпис випускника)

6. TIMETABLE

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Technical task coordination with the supervisor	1.10.20- 8.10.20	
2	Selection and study scientific literature on the topic	9.10.20- 15.10.20	
3	Process theoretical material on communication in CRM platforms and choose a stack of technologies	16.10.20- 18.10.20	
4	Develop a program and conduct testing to ensure that it works as expected	19.10.20- 03.11.20	
5	Analyze the results of the program, develop recommendations for its usage and make the explanatory notes	04.11.20- 12.12.20	
6	Preparation of the graphical materials and filing materials of work to antiplagiarism checking of materials	13.12.20- 14.12.20	
7	Receiving the reviews from reviewer and supervisor and providing the materials to the department	15.12.20- 18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor _____ Iskrenko Yu.Yu.
(Signature)

Assignment accepted for completion _____ Burba M.A.
(Student's Signature)

ABSTRACT

Explanatory note to the thesis "Organization of communication between Salesforce customers and clients ": XX pp., XX figures, 14 references.

Object of research: organization of communication between Salesforce customers and their clients.

Purpose: to develop a program for organization of communication between Salesforce customers and their clients.

Research methods: comparative analysis, processing of literature sources.

Obtained results: The Messaging Application was developed a result of the project and it can currently be configured to be used with Twilio messaging service. The main idea behind this system is that it can be expanded in the future and be used with other main communication channels. For example, it can be still SMS, but with the usage of any other messaging service which providing a decent API, or it can be any more or less popular messenger like Telegram, Viber, WhatsApp or any other messenger, providing an API encouraging integration. The application is integrated with the Salesforce platform.

CONTENTS

LIST OF SYMBOLS, ABBREVEATIONS, TERMS	10
INTRODUCTION	11
PART 1 NOTION OF CRM AND THE CONCEPT OF SALESFORCE.....	13
1.1. Introduction to Notion of CRM.....	13
1.2. The Reasons Why CRM Matters to Modern Business.....	14
1.3. Use Cases of a CRM System.....	15
1.4. Salesforce as the First and the Best CRM Platform	22
1.5. Salesforce Alternatives	24
Conclusions on the First Part.....	25
PART 2 COMMUNICATION CHANNELS AND MESSAGING SERVICES	26
2.1. Communication Channels in Business	26
2.2. Notion of Messaging Services and SMS API	32
2.3. Reasons to Use Twilio as a Messaging Service	35
2.4. Salesforce OMNI Channel as a Way to Configure Communication	37
Conclusions on the Second Part	41
PART 3 NOTION OF API, REST AND CLIENT-SERVER ARCHITECTURES	43
3.1. Notion of API	43
3.2. Client-Server Model.....	45
3.2.1. The Concept of Client-Server Interaction	45
3.2.2. Client-Server Architecture	47
3.2.3. Advantages and Disadvantages of Client-Server Architecture	49
3.3. HTTP Protocol.....	50
3.4. Representational State Transfer Architecture	53
Conclusions on the Third Part	56
PART 4 PROGRAMMING LANGUAGES AND FRAMEWORK FOR THE MESSAGING APP DEVELOPMENT	58
4.1. Apex as a Salesforce programming language	58

4.2. SLDS as a Markup and Styling tool	60
4.3. Notion of the Lightning Component Framework.....	62
4.4. Aura vs Lightning Web Components	66
Conclusions on the Fourth Part	68
PART 5 IMPLEMENTATION OF THE MESSAGING APPLICATION.....	70
5.1. Preparation of the Data Model	70
5.2. Implementation of the Messaging Administration Page	74
5.3. Concept of the Inbound Messages Tab.....	79
5.4. The Usage of the Chat with the Record.....	83
5.5. Setup of the Twilio.....	86
Conclusions on the Fifth Part	90
CONCLUSIONS.....	92
REFERENCE LIST	94

LIST OF SYMBOLS, ABBREVEATIONS, TERMS

SF	– Salesforce
CRM	– Customer Relationship Management
SFDC	– Salesforce Dot Com
ERP	– Enterprise resource planning
SMS	– Short message service
API	– Application programming interface
HTML	– HyperText Markup Language
HTTP	– HyperText Transfer Protocol
HTTPS	– HyperText Transfer Protocol Secure
XML	– eXtensible Markup Language
JSON	– JavaScript Object Notation
BEM	– Block Element Modifier
FTP	– File Transfer Protocol
OSI	– Open systems interconnection basic reference model
DBMS	– Database Management System
LWC	– Lightning Web Component
UI	– User Interface
SLDS	– Salesforce Lightning Design System

INTRODUCTION

In this graduation project there was covered a significant topic – organization of communication between the Salesforce users and their clients. In the first part the concept of CRM and its applications in the real world was researched in order to understand the actuality of this field, so there it is possible to find information about the everything mentioned above in addition to modern tasks of CRM in scope of Salesforce, examples of its usage, the reasons concerning the question why people need it and why it is important to expand the area of communication channels in CRM.

In the second part there is consideration of communication channel, which is a part of any business in terms of CRM, as this concept was used during the development of Messaging Application on the Salesforce platform for organization of communication. The part particularly includes the concept of communication channel itself, some information about its importance with its advantages, the benefits of setup of additional communication channel for CRM and even its disadvantages and problems, that prevent the modern CRM platforms from fast evolving – there is still some problems with it. In the end of the part there is a decision about what is the best messaging service for integration with the Salesforce for the application implementation.

In the third part it was considered the specification of client-server model and the notion of API as both were tightly used in the app. It is necessary to understand these concepts for anyone who decides to make an application similar to one that was implemented in this graduation project. The concepts was described enough to get the idea behind them and realize their importance and beneficial impact on the development. It is also possible to find here which exactly API was used to build the application.

In the fourth part there is information about Salesforce programming language and its frameworks, as it is a platform in which the application was developed, some its features and important facts about it, such as Apex, SLDS, LWC and Aura. After all, in the fifth part there is an information about the developed app itself, its principles of work, examples

of real use and some explanations about the work with the chosen APIs, several examples of the information that of of them provide and some advices on configuration of the system.

As the result of graduation project, there was gained understanding of current state of the concept of communication channels in CRM, its tasks and problems. The Messaging Application was also developed a result of the project and it can currently be configured to be used with Twilio messaging service only, but the main idea behind this system is that it can be expanded in the future and be used with other main communication channels. For example, it can be still SMS, but with the usage of any other messaging service which providing a decent API, or it can be any more or less popular messenger like Telegram, Viber, WhatsApp or any other messenger, providing an API encouraging integration. There was also gained information about concepts, necessary for implementation of the Salesforce app, which was also successfully developed. The project covers a lot of spesific themes and can be a good starting point for further and more deep investigation of the field of CRM, importance of its communication channels and areas related to it.

PART 1

NOTION OF CRM AND THE CONCEPT OF SALESFORCE

1.1. Introduction to Notion of CRM

Customer relationship management (CRM) is a technology for managing all the company's relationships and interactions with customers and potential customers. The goal is simple: Improve business relationships to grow your business. A CRM system helps companies stay connected to customers, streamline processes, and improve profitability.

When people talk about CRM, they are usually referring to a CRM system, a tool that helps with contact management, sales management, agent productivity, and more. CRM tools can now be used to manage customer relationships across the entire customer lifecycle, spanning marketing, sales, digital commerce, and customer service interactions [1].

A CRM solution helps you focus on the organization's relationships with individual people — including customers, service users, colleagues, or suppliers — throughout your lifecycle with them, including finding new customers, winning their business, and providing support and additional services throughout the relationship.

A CRM system gives everyone — from sales, customer service, business development, recruiting, marketing, or any other line of business — a better way to manage the external interactions and relationships that drive success. A CRM tool lets you store customer and prospect contact information, identify sales opportunities, record service issues, and manage marketing campaigns, all in one central location — and make information about every customer interaction available to anyone at your company who might need it.

With visibility and easy access to data, it's easier to collaborate and increase productivity. Everyone in the company can see how customers have been communicated with, what they've bought, when they last purchased, what they paid, and so much more. CRM can help companies of all sizes drive business growth, and it can be especially beneficial to a small business, where teams often need to find ways to do more with less [2].

1.2. The Reasons Why CRM Matters to Modern Business

In modern business, the need to automate various processes has become commonplace. It is already becoming difficult to imagine warehouse or accounting without the use of specialized software, sales representatives use special applications to place and send an order to the office directly from a tablet or mobile phone, a fairly large part of orders comes from the site already in the form of documents ready for processing. But at the same time, relationships with customers, at least in medium and small businesses, for some reason very often are conducted without the introduction of automation and sufficient attention to accounting [3].

People observe all this quite often, since as a business consultant. And every time they have to tell the clients how to automate customer relationships, how CRM systems work, what they are, and why in a particular case it is worth choosing one or another system.

What happens if the sales department operates without an accounting system. Each sales manager works in the way that is more convenient for him, records calls, other types of interaction with clients at his own discretion: someone on paper, someone in Excel spreadsheets, and someone does not consider it necessary to record their process. work.

Incoming calls or requests from the site from new customers are also not recorded, it is often even difficult to understand which of the managers is in charge of the incoming request. As a result, real accounting is kept only at the level of paid orders and shipment of goods. And it turns out to be impossible to determine how efficiently the sales department works, whether all incoming LEADs are being worked out, whether any work is being done with existing contacts.

In addition, in the event of an employee's dismissal or illness, the company may lose all his unfinished negotiations and unprocessed contacts, which is also highly undesirable for the effective work of the sales department.

The way out of this situation is the automation and standardization of customer relationship management, i.e. implementation of a CRM system.

This solution will help:

- 1) Get a standardized database of contacts (clients, counterparties) common for the company;
- 2) Effectively monitor the quality of the work of the sales department at any time;
- 3) Get statistics and analytics on the effectiveness of work with leads (incoming calls, requests);
- 4) Plan to improve the quality of work and develop a business development strategy.

In fact, any control and accounting option that can help improve customer interaction can be considered a CRM system. Even if you keep a history of calls and contacts on paper or in Excel, this can be considered a CRM system if the developed accounting and control scheme works and allows you to control all options for interacting with customers. Of course, such accounting methods are a thing of the past, because in the modern world it is difficult to imagine the work of any business without effective automation. Therefore, when they talk about a CRM system, they usually mean special software.

In some cases, the developers introduce additional confusion into the terminology by promoting the term "CRM". Software developers also often try to impose their vision of what might be called a CRM system. Usually, their list of the necessary qualities of a real CRM completely coincides with the option that they implemented in their software product. In some cases, such marketing works quite explicitly and obtrusively, according to the principle CRM should be this and only this.

1.3. Use Cases of a CRM System

Before choosing a CRM system, it is needed to understand whether the business needs it in principle. It often happens that someone told a business owner about the existence of such systems, or software vendors try to impose their product.

CRM systems are essential for any business that works directly with customers and seeks to expand the number of buyers.

So, if incoming calls or requests (leads) from new customers are important in the work of a business, if the business is making any efforts to get and retain new customers, then a CRM system is necessary. For example, an online store, wholesale company, or beauty salon

will not be able to work effectively without careful attention to leads (incoming requests and calls). Indeed, in each of these types of business, it is very important that all orders are fulfilled, that buyers of goods and services are satisfied, and that customer loyalty increases. On the other hand, if your business is not interested in increasing the number of customers at this stage of development, if the loyalty of regular customers is based on long-term contracts, and all contracts with new customers are based on personal meetings, even the best CRM system will do nothing.

For example, a retail store will not use a CRM system in its work, since customer contacts cannot be recorded. Here, customer loyalty is based on product quality and service quality, i.e. on personal contact with the seller. Also, the CRM system will not help in any way in cases where the company has entered into long-term contracts for the supply of goods to the chain of stores, the supplier's capacity is barely enough to fulfill the orders of this distribution network, and everyone is satisfied with this state of affairs.

But if to develop the business, if to invest in advertising, make other efforts in order to attract and retain customers, then the CRM system will be an excellent assistant in the work.

The question of explaining complex and new things is an integral. And people often have to explain why a client needs a CRM system. What it is, a businessman can know himself. But at the same time, very often representatives of small and medium-sized businesses do not understand why they need it. After all, the number of clients is relatively small, the sales department also in companies of this level consists of only a few people. And it seems that even without a CRM system, it is as easy as shelling pears to control work with customers. In fact, this is not the case. Very quickly after the introduction of an automated system, a huge number of shortcomings are revealed, and the quality of the sales department's work increases significantly. CRM is needed in order to:

- 1) Don't lose a potential client, don't miss a single incoming call and request. In small and medium-sized businesses in our country, competition is very high. Companies go to great lengths to attract customers to get noticed. Compared to other customer acquisition costs, a significant budget is allocated. And it is very important that all these funds and efforts are not wasted. Automated systems allow you to get confidence that this is how the

sales department will work. You will receive a record of every incoming call, every request, every lead.

2) Control of employees' work and standardization of work with clients. Without a common standardized CRM system, every employee works the way they are used to. Someone keeps records in spreadsheets, someone - in a notebook or diary, someone does not keep records at all, focuses exclusively on reports from 1C or on their own memory. Contacts are also quite chaotic. Letters to clients can be sent from both corporate and personal mailboxes, calls can be made from any convenient phone, quality control of work is impossible. A CRM system almost completely solves this problem. Information about all incoming and outgoing contacts will be stored in one repository, from where it can be retrieved at any time.

3) The statistical base is accumulating, which is also very important for the successful development of any business. Thanks to the use of a CRM system, all working information is collected in one common database in a standardized form. As a result, the manager can analyze the statistics of work, draw up various reports (many of which are already in ready-made form in CRM systems), i.e. analyze work and plan subsequent work more consciously.

4) Ready-made solutions from which you can build on in building your own system of work. Each CRM system is the embodiment of the developers' vision of how to work with a client. It contains many ready-made tools that allow you to take your work to a qualitatively new level. For example, the integration of a CRM system with telephony allows you to record all calls, remember all new contacts and analyze the quality of the sales department's work with leads. In small and medium-sized businesses, work with clients is most often directed directly by the head (owner) of the business. He does not have experts, and often does not have the best practices for organizing work with clients. The manager has nothing to rely on in this matter, and therefore the sales department often does not work well. The implementation of a CRM system allows you to get not only a tool, but also help, a view of the developers on how the sales department should work. In turn, when developing CRM systems, they usually rely on best practices, on experts in working with clients. Therefore, if you actively use the tools provided by the CRM system, then the work

of your sales department will also be optimized. Various system tools themselves suggest which steps should be taken in the process of optimizing work with clients.

In addition, the user of the CRM system will receive many other useful things, most of which depend on the chosen system. But I always explain these four basic things, because they are very important for small and medium-sized businesses, and also because any of the existing CRM systems provides them.

When choosing a CRM system, the most important thing is to make sure that you have all the functions that you would like to see in the process. So, if incoming calls are very important to you, you need to make sure that the selected CRM system supports integration with telephony. And if you get most of the leads through the site, then one of the main criteria will be the ability to integrate the CRM system with your CMS. Otherwise, much depends on your tastes, as well as on the recommendations that your specialist gives you. In principle, if a specialist who will implement a CRM system offers you a certain software product, then, provided that the functions you need are implemented in this system, and you are satisfied with the cost of the product, it makes sense to agree with his opinion. Usually, experts advise the product that they know well, which will undoubtedly be a plus at the stage of implementation. It is quite difficult to study a CRM system based on videos and test access, in any system there are many nuances that you will learn while working with the system. But there are some fundamental points that will help you make the right choice. So, the main thing is the very fundamental decision to implement a CRM system. Further, if you have any preferences, you saw a system that you liked for some reason, implement it. In all other cases, it is best to rely on the opinion of a specialist.

There are two types of CRM systems based on different technologies:

1) SaaS or system as a service. With this option, all software and data is located on the server of the service provider. You get online access to the system through a browser, client program or mobile application. All processes take place on the side of the service provider.

2) Standalone is a license to install and use a software product. One gets a solution that to install on someone's own server, if they wish, you modify it to fit the needs, depending on the capabilities provided by the supplier of the CRM system.

There are some limitations when choosing a SaaS solution. One will not be able to change anything in the product code, since the software solutions are located on the vendor side of the CRM system. Typically, such CRM systems allow someone to set up employee access rights, integrate some external systems (receive data from the site, record incoming calls, etc.), change the design using the designer, customize reports, etc. But all this will be stored on the servers of the supplier of the CRM system. It is also important to understand that when using SaaS solutions, one must always have access to the Internet. Of course, in our time, a reliable Internet has long become an important part of any business; in the absence of access to the network, many business processes stop. Therefore, the optimal solution is to have, in addition to the reliable main, also a backup Internet access channel. Another important point to understand when choosing SaaS solutions: most likely, you will need to pay separately for each database backup and other similar operations. For example, on a system that I actively use, a backup costs \$ 10 per backup. Pros of SaaS solutions:

- 1) No need in own server to host the software;

- 2) No need to do the updates yourself, all this lies with the service provider, just use the solution.

- 3) A lot can be said about CRM as a SaaS solution. The features described above are sufficient to choose the type of solution.

Stand-Alone solutions, as was said above, are the purchase of a “boxed” solution that one install on his own server and can change the program code (within the access provided by the developer). In some cases, for example, when it becomes necessary to implement atypical solutions, this level of access is very important. But more often than not, Stand-Alone solutions are not required for small and medium businesses. The need for deep changes is extremely rare, and therefore the best way usually is to use SaaS.

Any CRM system should integrate with telephony. If users cannot record incoming calls and initiate outgoing calls, then this is a big disadvantage. Therefore, when choosing a software product for the clients, they should always pay special attention to the availability of this feature, as well as to how it is implemented. It would seem that it is possible to enter data on calls into the system manually. But practice shows that this method does not

work. People begin to resist, they are annoyed with the need to do extra work. In addition, anyone can simply forget to enter this or that important call into the system. Therefore, this method usually does not work in practice. So, it is necessary to record calls in the system. There are 2 options for implementation:

1) The call is made from the browser itself, it goes completely through the system, all interaction take place through the browser. It is important to understand that the entire call goes through the system, and therefore the sound quality, signal processing speed, etc. depend on the browser and CRM code.

2) Telephony is integrated with third-party services - asterisk, avaya, etc. In this case, user install a virtual telephony system based on these services and connect his numbers to this telephony. At the same time, user will be able to make all outgoing calls and receive incoming calls through sip tubes, and not through the browser. How does this happen. The VPS provider accepts a call from a client, redirects it to the virtual PBX system, and it already transmits information about the call to the CRM program. At the same time, the CRM database records the phone number, time, call duration, etc. The user only has to add his own notes to the call record (short topic of conversation, result, comments).

Of course, there is another option for recording information about calls manually, but it was already described above why this will not work in practice. The main routine operations must be automated, otherwise the system will not work.

API integration: availability of ready-made solutions - any business uses various services to receive applications, keep records, draw up documents, etc. When choosing a CRM system, they should pay attention to whether there are API solutions for integration with your website, data exchange with 1C, IT telephony, other programs they need, and services. Having a ready-made integration API is a big plus. Interaction with a contact (client) usually consists of several things:

- 1) Phone calls;
- 2) Email correspondence;
- 3) Mailings (SMS or email);
- 4) Meetings.

The first 3 things need to be automated. They give an idea about the history of relationships with the client, help to understand what is happening in the work with him at the moment, what were the last actions.

Planning and working with tasks - when choosing a CRM system, it is also worth paying attention to whether the system has the ability to plan, formulate and work with tasks. Employees should be able to create tasks for themselves and colleagues, set reminders, etc. In addition to working with clients, the CRM system must have the necessary tools to record the interaction between employees. It is not enough that they communicate with each other in the office or on the phone, discuss projects and solve some problems. Users also need to be able to control the workflow and employee interactions. Users should be able to set themselves and others tasks related to both working with the client and reporting, with other necessary actions. Also, a very convenient function is the ability to customize the automatic task setting service, for example, once a quarter, a task for preparing quarterly reports will be created, etc. Or, when the system receives the contact information of a potential client (a new lead appears), the system will automatically create a task for a specific employee, for example, call back at the specified phone number.

Integration with SMS service - as such, there is no integration with an SMS service, for example, a ready-made solution for integration with certain SMS messaging services. This is done as part of business processes or as part of working with an API. Accordingly, the system should be able to send bulk SMS for all customers, for a specific group of contacts or for a personally specific contact (for example, notification of certain actions).

Data import - take a close look at what capabilities the CRM system provides for importing data. In what format can the information be downloaded. Is there a ready-made module for migration from other systems, and if so, from which ones. Or do you need to prepare information for uploading in some specific format. Users must understand how the initial filling of data will be carried out at system startup. This process is somewhat similar to the input of leftovers into the system. It is very important that the data import is fast, simple and transparent. Without a convenient automatic transfer of all contacts and other important information for work, the launch of the system will most likely end in failure. Of

course, users can enter all the data manually, but this is very long and inconvenient. And if they enter this data in parts, then the risk of duplication of customer cards increases, as a result, confusion and overlays await you. Obviously, the best option of transferring data from an Excel spreadsheet, this option is universal, quite visual and convenient. In Excel, it is possible to unload from almost any system, including 1C. And uploading data in this format to the system is also quite fast and convenient.

1.4. Salesforce as the First and the Best CRM Platform

Salesforce is a popular cloud-based CRM owned by a US company of the same name. Gartner named Salesforce the best PaaS solution for business in 2018, and in 2019 Salesforce was recognized as the leader in the CRM market for the last 6 years. Salesforce is already used by more than 150 thousand companies, including Sony, Financial Times, Toyota and many other well-known global brands.

Salesforce is dynamically developing not only CRM, but also an ecosystem for development for this platform. According to a study by the international consulting company IDC, by 2022 the Salesforce platform will create about 3.3 million jobs worldwide, including positions for developers.

Belarusian IT companies working with foreign clients already have vacancies for Salesforce developers. But in our market, employers still find it difficult to find suitable candidates, which increases the chances of novice Salesforce developers for employment and intensive career growth.

Salesforce is a cloud-based business process management platform for sales, customer service, digital marketing. And the key products for these purposes are Sales Cloud, Service Cloud, Marketing Cloud. Also among the demanded solutions for business are Commerce Cloud (for e-commerce), Financial Services Cloud (for companies from the financial sector), Health Cloud (for medical organizations). Salesforce also offers additional components that you can incorporate into their products. For example, CPQ (Configure, Price, Quote) is a tool for automatically generating price offers, Chatter is a corporate social network.

In 2007, Salesforce launched Force.com, the first PaaS platform for developers, which allowed the creation and deployment of applications on the Salesforce infrastructure. In 2018, Force.com was transformed into the Lightning Platform, but the first name is often heard among developers [4].

Salesforce regularly updates its services and adds new modern solutions. One such product is Einstein AI, which contains elements of artificial intelligence. With the help of Einstein AI, businesses can integrate predictive analytics into their CRM, for example, to predict which potential customers (leads) are most promising for further cooperation.

Another promising product from Salesforce is IoT Cloud. This platform allows you to collect and process data in real time from devices integrated into the IoT system, and CRM users can manage this data. For example, in retail, IoT Cloud can aggregate data from sensors on goods so that you can track the conditions of their transportation and storage.

Salesforce can be tailored to the needs of the business in two ways - through configuration and customization. Developers are primarily involved in customizing Salesforce, but can also partially do configuration.

When configured, CRM functionality is configured through the Salesforce admin user interface without using any code. So it is possible to create tables, objects, fields, form relationships between them, set rules for automating workflows. A large business usually hires a separate specialist for the CRM configuration - the Salesforce administrator.

Customization is needed if the configuration options are not enough to automate complex business logic.

Examples of Salesforce customization:

- 1) Development of new functionality for standard CRM modules;
- 2) Automation of complex business processes for the needs of a particular company;
- 3) Development of custom templates for email messages;
- 4) Implementation of Einstein AI capabilities in CRM;
- 5) Creation of custom reports and dashboards;

6) Integration of Salesforce with other systems (ERP, company website, social media, etc.);

7) Development of custom applications for Salesforce (IT grocery companies can place such applications in a special market - AppExchange).

1.5. Salesforce Alternatives

While Salesforce is the market leader for CRM, there are plenty of alternative platforms. Salesforce's two primary competitors are SAP and Oracle. SAP offers SAP C/4HANA, a full suite of customer experience solutions that include customer data management, marketing, commerce, service, and sales. The company also acquired Qualtrics, which has strong analytics capabilities. Oracle's CRM product is Oracle CX Solutions.

There are other lesser-known (but still viable) Salesforce alternatives that are all cloud based as well:

1) Adobe is probably best known for its creative suite, but it also offers a viable CRM product, Adobe Marketing Cloud, which is geared toward the enterprise market.

2) Microsoft Dynamics 365 provides CRM functions as well as field service, human resources, and artificial intelligence features.

3) HubSpot CRM is free and provides automation for a lot of tasks like sending follow-ups. It integrates with Gmail, social media, and call platforms, and has tools for live chat.

4) Freshsales provides email tracking and event tracking. Their pipeline is a drag-and-drop interface so you can see where prospects are in their customer journeys.

5) Zoho CRM automates sales operations and provides real-time data access. They also integrate with Google Apps.

6) Bpm'online CRM lets to use both business process management (BPM) and CRM tools in one program, and is relatively easy to customize.

If user is hesitant about choosing a CRM system, he can consider Salesforce and its competitors [5].

Conclusions on the First Part

CRM is an interaction model based on the theory that the center of the entire business philosophy is the client, and the main areas of the company's activity are measures to ensure effective marketing, sales and customer service. Supporting these business goals includes collecting, storing and analyzing information about customers, suppliers, partners, as well as about the internal processes of the company. Functions to support these business goals include sales, marketing, customer support.

Working with a CRM system will have a positive effect on the control of the sales department and will be able to minimize unnecessary burden on managers. Most of the routine processes (execution of contracts, commercial offers, invoices, specifications) can be automated and not waste valuable employee time on this. The program will store all information on your clients and transactions, as well as analytics, statistics and reporting, which will be built dynamically based on the incoming data.

Salesforce (salesforce.com) is an American company that develops the eponymous CRM system provided to customers exclusively on the SaaS model. The company provides a PaaS system for self-development applications under the Force.com name, and a cloud-based database management system under the Database.com brand name.

PART 2

COMMUNICATION CHANNELS AND MESSAGING SERVICES

2.1. Communication Channels in Business

Today, in the era of modern digital technologies, the number of communication channels is growing rapidly and has already exceeded a couple of dozen. When communicating with their consumers, companies use direct and indirect channels, depending on the goals and objectives.

For example, in order to inform about discounts, promotions, as well as increase sales, it is best to use instant messengers, SMS, email, contextual advertising and remarketing (direct channels).

When it comes to creating an image, forming a certain message for a specific audience, it is more rational to use indirect channels: social networks, YouTube, media, company websites, etc. Social networks are one of the key shareware communication channels that allow you to broadcast company news, announce marketing activities, maintain direct contact with customers (including interactive). By correctly combining entertainment and selling content, you can effectively build an engagement-sales chain. Using Facebook as an example, I propose to consider this channel in more detail [6].

The indisputable advantage of Facebook is the huge number of communication formats (from posts to dynamic product retargeting). In addition, the functionality of business pages allows you to effectively work with reviews. Along with information from customers business owner to obtain an opportunity to improve your product or service.

To ensure a sustainable online presence, be sure to create separate pages for the main office of the company and each of its branches. And also keep in mind that regional users may differ from “metropolitan” users (interests, behavioral factor) due to different levels of income, opportunities, list of visited resources. Determining the interests of the target audience and detailed targeting will allow you to find potential customers.

When using only the "main" business pages measured data can be contained errors. It is difficult for marketers to determine which cities have the highest engagement. As a result, there is a lack of data to improve the relevance of content at the local level. Affiliate pages don't just drive analytics, they provide the ability to get the data you need for proper accounting.

Companies can add all their affiliates to Facebook, managing information about each of them centrally (Ads Manager Resources Company Points). You can control and receive all information from one profile. This is especially true for chain establishments, as well as those working on a franchise.

Besides people live in a multi-channel world. It has become a norm that has become so deeply rooted in their daily experience that we no longer notice how attention is constantly jumping from one information channel to another.

Multichannel marketing occurs where sales are made through multiple channels of communication with customers. This strategy, also known as omnichannel marketing, provides many opportunities for brand engagement.

Simply put, multichannel marketing is "selling everywhere". For example, a fashion company uses multichannel marketing to promote its online store on social media, advertise in fashion magazines, e-mail company news, and broadcast videos on a local mall display. Each of these channels is a possible point of contact for potential customers with this company.

Multichannel marketing puts customers at the forefront of communication, allowing them to decide how, where and when to learn about the company and its products. Multichannel marketing ensures that all points of sale remain relevant in the consumer journey of customers, whether they want to interact with your brand online, via computer or mobile phone, via SMS or in person.

The key word is "choice". Here are just a few of the main channels involved:

- 1) Retail store;
- 2) Web site;
- 3) Email;
- 4) Mobile app;

- 5) Search Engine Marketing (SEM);
- 6) Social Media Marketing (SMM);
- 7) Mailing list;
- 8) Mobile.

As Malcolm Gladwell pointed out in his famous talk, Choice, Happiness, and Spaghetti Sauce, people love choice. Thus, the more communication channels there are, the greater the choice and the higher the probability of making a sale. Making the most of these opportunities is what omnichannel marketing is. The channel use percentage can be found on the figure 2.1.

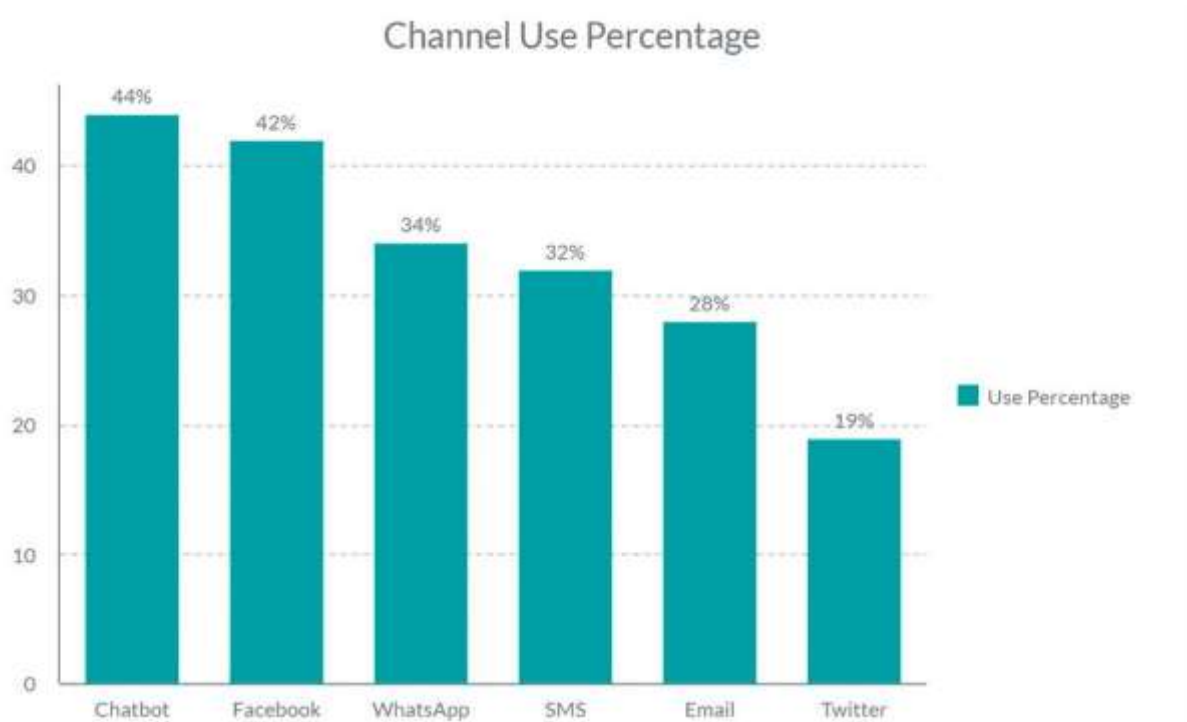


Fig. 2.1. Channel use percentage

Although a lot has changed since then. This journey is now an immersive, interconnected experience. It includes various locations available through a variety of channels, both digital and face-to-face. Today, consumer interaction with brands occurs on average through six communication channels simultaneously, almost 50% of which use more than four of them [7].

“Multichannel” clients spend 3-4 times more than those connected to only one communication channel. What other arguments are needed here. It is difficult to argue with such facts.

Moreover, marketing teams that manage to integrate multiple channels are 2x more likely to see improved marketing effectiveness than teams with less integration. More consumers using more brand communication channels means more earning opportunities. Simply put, the more the better.

A minimum of 5-7 contacts with a brand is required before the consumer remembers it. The more communication channels there are, the more likely it is to attract the attention of the target audience and brand awareness. Simple math. An experienced marketer will be able to achieve this high level of representativeness using a multi-channel strategy. Especially when almost everyone has the best brand marketing tool in their pocket - a mobile phone. Whatever one may say, 51% of consumers use it to find new brands and products, and in general, to interact with brands, people are 2 times more likely to prefer mobile to other communication channels.

Multichannel marketing provides a better understanding of demand. New data comes from each channel of communication. The resulting analytics provide a clear understanding of the core of the customer audience, which in turn will help increase sales through more personalized offerings.

The customer is the driving force behind the marketing strategy. And not vice versa. This may seem obvious, but it was not always so. Increasingly, customers are dictating how and where they want to interact with the brand. And wherever that happens, marketers need to be prepared.

Multichannel marketing can be intimidating in scale, especially for smaller companies with limited resources. But the beauty of an open modern digital world lies in the fact that now everyone has access to what only a few could previously afford. All thanks to mobility.

Mobile communications have made communication (including marketing communication) easier than ever. Given that the number of mobile users in the world is approaching 3.9 billion, and in some regions mobile devices account for more than 50% of Internet traffic, there is no doubt that mobile messaging remains the best way to communicate with customers.

Fast, efficient and diverse mobile messaging services have become the backbone of multichannel marketing. The absence of a shortage in the choice of mobile communication

channels and the unique advantages of each of them play a key role in the full disclosure of communication potential.

Proper integration of mobile marketing channels is paramount to taking your business to the next level. Below, we'll take a look at the best channels to build an effective multi-channel mobile marketing strategy.

Push notifications are automatic messages that a mobile app sends to users, even if it is not currently in use. It can be any notification, reminder, promo or call-to-action that ensures constant contact with customers in real time. The quintessential multichannel marketing concept, push notifications also easily connect to other forms of messaging such as SMS, email and voice calls. Also reach of levels of communication can be found on the figure 2.2.

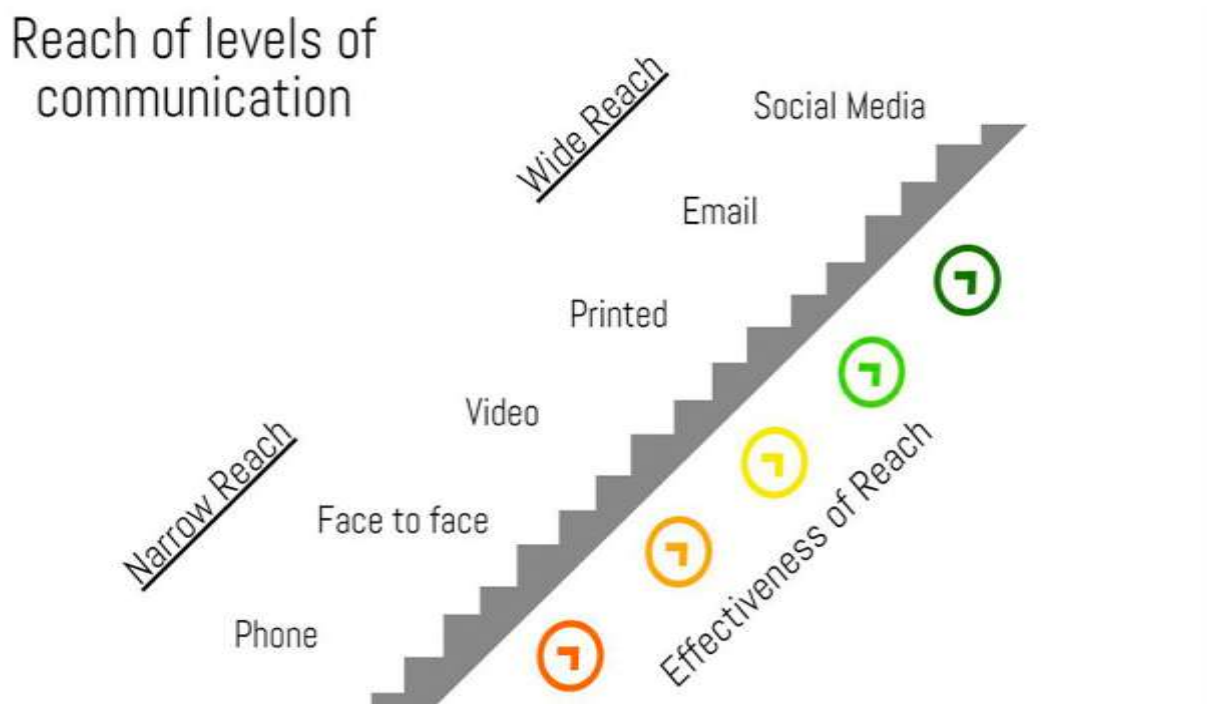


Fig. 2.2. Reach of levels of communication

With the rise in popularity of smartphones, push notifications have become one of the best ways to communicate with customers. Better than others - users willingly agree to receive them. This means that push notifications lead to better engagement than most other forms of communication.

1) E- mail. No marketing strategy is complete without email. This communication channel may have suffered from oversaturation and thoughtless use in recent years, but e-

mail messaging remains the most versatile medium of online marketing. Especially if you are following a 5-step approach to creating effective email campaigns.

This is the best channel for mass mailings, as well as long-term communication, with which few channels can match in terms of adaptability to brand goals. The challenge for digital marketers is to make their brands' emails stand out from the crowd, while avoiding the dreaded “spam” label. This is where multichannel marketing begins, which increases the efficiency of e- mail communication by integrating it with other mobile services, creating an engaging, brand- absorbing consumer experience.

2) SMS has long been one of the foundations of mobile communication. Since then, other services have appeared, but SMS still retains its position as one of the most effective means of marketing communication. It is worth taking into account the high percentage of SMS open - 98% and response rate - up to 48%. Compare this with the efficiency of e- mail campaigns, which averages 20-30% and 8%, respectively.

And let the limitation of SMS messages to 160 characters do not mislead users. With A2P messaging, brands can quickly and efficiently reach their audiences at a variety of levels, from personalized support to offering up-to-date discounts and affordable products.

3) OTT messages. Originally the term OTT (“over-the-top”) described digital media services such as Netflix , which offers the ability to watch content without the need for a TV network or cable provider. Today, online messaging services are taking OTT services to the next level. These services allow you to communicate with others from virtually anywhere using any mobile device without relying on the device's built-in text messaging capabilities.

For example, compared to standard SMS, a messenger like Viber allows to send messages significantly longer than 160 characters.

The Viber Business Messaging service opens up much more opportunities for communication with customers. In addition to text, the message may contain pictures, links, the possibility of two-way communication with the client. The service also provides end-to-end encryption from subscriber to subscriber, ensuring that message texts are not available to third parties.

It is predicted that by 2025, almost a third of the world's population will use OTT services to send messages, resulting in more than 60 trillion messages sent via instant

messengers, compared to 4.9 trillion via SMS. This means that communication with customers via messengers is ideal for progressive digital marketers who want to reach the largest possible audience.

4) Mobile Marketing Channels: One For All, All For One. Individually, each of the above mobile marketing channels is a powerful marketing communications tool that will help the development of any business. But what makes them really useful is that, when used together, they maximize business / marketer challenges in building or maintaining communication with customers.

With seamless cross-channel engagement, it's no surprise that choosing a multichannel strategy focused on mobile is not a challenge for any brand. To learn more about how Push , E- mail , SMS and OTT messages can help your business grow, go to the detailed description of mobile messaging services .

There is a reason more and more companies are opting for multichannel marketing. The results speak for themselves. Multichannel marketing gives customers a personal touch. This empowers customers by building trust and brand loyalty, in addition to increasing engagement and conversions.

Thanks to the development of messaging services, it has never been easier to reach an audience, regardless of location. First of all, this approach has proven to be effective in retail.

2.2. Notion of Messaging Services and SMS API

Online services for sending sms-mailings, e-mail-mailings, as well as mailings on popular messengers significantly simplify the work of informing clients and customers about important company events. Internet platforms for sending SMS messages have become an indispensable tool for employees responsible for marketing and communication with customers.

For the most part, SMS services offer many different filters and settings, thanks to which you can take into account time zones, regional affiliation of the subscriber, send mailings to certain operators, and much more. Tariffication for services depends on the

number of SMS messages sent, the amount of top-up, as well as on the choice of the channel for sending. Depending on your needs, you can choose a suitable SMS aggregator and a comfortable tariff plan.

Today, a large number of services are presented on the Internet that provide a service for sending SMS mailings. In order to make the right choice, we suggest that you familiarize yourself with the main advantages and features of their work.

According to several IT resources, there are 8 APIs to be the very best and worth mentioning, which are also depicted on the figure 2.3 with their best advantages in comparison with other services:

- 1) Twilio SMS API;
- 2) Telesign SMS Verify API;
- 3) Nexmo SMS Messaging API;
- 4) D7SMS API;
- 5) Telnyx API;
- 6) MessageBird API;
- 7) ClickSend (Inteltech) API;
- 8) Nexmo Verify API.

Twilio SMS	Best for SMS Messaging, Verification, and Number Lookup
Telesign SMS Verify	Best for Verification & 2-factor Authentication
Nexmo SMS Messaging	Best for Sending and Receiving Text Messages Globally
D7SMS	Best for Worldwide SMS Transmisson
Telnyx	Best for Voice, Data & Messaging
MessageBird	Best for Voice, SMS and WhatsApp
ClickSend (Inteltech)	Best for Sending & Receiving SMS
Nexmo Verify	Best for SMS Verification

Fig. 2.3. The list of the most famous messaging services

Twilio is a platform for creating sms messages, calls and other communications. With its help, you can set up automated mailings and receive response messages, make voice calls and video calls, receive verification codes for authorization on various portals, create call centers and chat bots.

Building a unique communication app used to take a long time. Twilio made this process easier, as there is no need for it to develop a special infrastructure. This allows programmers to immediately create the desired product in the programming language that is convenient for them.

Twilio invites its clients to focus on their current goals (communicating with partners, customers or subordinates) instead of spending a huge amount of time negotiating with mobile operators and solving communication problems. This is especially important for those firms that operate in several regions. The correct distribution of system support will make it possible to reach any person, no matter where in the world he is.

The Telesign SMS Verify API offers verification and two-factor authentication (over SMS) capabilities through their REST API. This is the perfect API for adding authentication into your mobile app with a verification code (via text message).

The Telesign API has one endpoint: Send Verification Code

The Nexmo SMS API is one of the top SMS APIs out there. It allows developers to send and receive SMS text messages to users around the globe through their REST API. Nexmo is one of the preferred APIs if you're looking to integrate SMS messages into your Android or iOS mobile app.

The API has 4 endpoints:

- 1) Send an SMS;
- 2) Pricing by Country Code;
- 3) Pricing by Phone Number;
- 4) Search Numbers.

For the most part, SMS services offer many different filters and settings, thanks to which you can take into account time zones, regional affiliation of the subscriber, send mailings to certain operators, and much more.

2.3. Reasons to Use Twilio as a Messaging Service

The main advantages when using Twilio are:

1) Adequate price. A flexible tariff system is presented. This means that the customer does not need to conclude a contract for a month or a year. You only pay for what you actually use. This allows you to constantly monitor the budget and redirect the surplus to other important goals.

2) Work without delay. Users add communication and identification tools to their applications, regardless of which programming language they are using. The simple yet powerful API supports Python, Java, PHP, Ruby, Node, and .Net (C #). Therefore, developers save time and can immediately start designing communication channels and work at the highest possible speed.

3) Reliable connections. It has become much easier to connect with colleagues, clients and subordinates located on the other side of the world. Moreover, the reliability of the software is estimated at a maximum of 99.95% according to the SLA (Service Level Agreement). This means that interested parties will be able to communicate with each other with an extremely low risk of the conversation being interrupted.

4) Regional distribution. If the technology needs to be tailored to local needs or needs to be compatible across different territories, developers can easily change part of the code without having to rewrite the entire program. Thus, subsidiary websites of an enterprise operating in several cities or countries can easily interact with each other.

The platform gives the API the following properties:

- 1) Communicativeness;
- 2) Task routing;
- 3) Multichannel notifications;
- 4) Authorization;
- 5) Account verification;
- 6) Proxy;
- 7) Natural speech processing.

Since every business organization has specific needs, there is no point in trying to find the perfect one-stop system solution that will suit everyone. It cannot be found among popular and widely used programs. Therefore, the smartest decision is to make a list of priority tasks based on such indicators as technical requirements, financial capabilities, employee skills, company size, etc.

Bust users must be careful about your preparatory work. Carefully read the reviews for each of the systems in question (including Twilio). Such research will allow you to get rid of "weeds" - those options that for one reason or another you will not like. As a result, you will choose the application that will satisfy your needs.

After preparing to obtain a Twilio account, sign up for the Twilio Trial page. You can start with a free account and upgrade later.

After registering for a Twilio account, you will receive a free phone number for your application. You will also receive an account SID and an authentication token. These elements are required for Twilio API calls. To prevent unauthorized access to your account, keep the authentication token secure. The account SID and Authentication Token appear on the Twilio Account page in the ACCOUNT SID and AUTH TOKEN fields, respectively.

You can think of a Messaging Service as a container to hold all of your sender IDs and to manage the configuration affecting the delivery of your messages. For example, you could put your short code and local numbers in one Messaging Service sender pool and configure them to point to the same webhook URL to respond to incoming messages. You could also define the same set of Stop words per country for all of the senders in your sender pool using the Messaging Service Advanced Opt-Out feature.

When sending messages from a Messaging Service, you can set the From parameter to your Messaging Service SID, instead of a specific phone number. This not only provides intelligent routing for the senders in your Sender Pool, but also gives you access to the various Messaging Service Features, like Advanced Opt-Out, Sticky Sender, Shortcode Reroute, Smart Encoding and more.

A Messaging Service's features can be managed and configured directly through the Console as well as using the REST API.

With each Messaging Service that you add through the console, you'll start by configuring both the Inbound Request and Status Callback URLs. You'll also need to associate one or more phone numbers or short codes to the Service to send messages. You can add additional messaging features (described below) to your application through your newly configured messaging service.

2.4. Salesforce OMNI Channel as a Way to Configure Communication

New technologies will make it possible to earn more through the intensive and comprehensive use of information. Proponents of the omni-channel approach believe they can get the most out of their data business. Is this really so and is there a practice of applying such an approach in the modern world.

Omni-channel has been talked about for several years. It is believed to be the logical product of the evolution of electronic retail. The main message of this approach comes down to placing the client at the forefront and building an infrastructure for interaction with him. This is very similar to how people build relationships with other people in life. For example, on the way to creating an alliance with the person of the dreams, people successively go through three stages:

- 1) Looking for a suitable partner (search stage);
- 2) Collecting information about him (recognition stage);
- 3) Making an offer that is difficult to refuse (offer stage).

When meeting a potential partner, users try to present themselves from the best side, demonstrating their positive qualities. Likewise, a business is engaged in the placement of advertisements and other information that can attract a client.

The stage of recognition, i.e. obtaining information means collecting information from the very first contact. In Internet marketing, from the very first contact with advertising, if they are allowed to put their pixel in an advertising medium (banner or article), they immediately begin to collect information about a potential client, and this process continues throughout the subsequent interaction with him. Even in case of refusal to work with them, they ask him to inform him why this happened. As in real life, users

make an offer that is difficult to refuse to their potential partner more than once, but time after time they go back one step and present it with a new version. By going iteratively, they make a large number of small but carefully thought out proposals. In ordinary life, this allows users to extend the relationship with a potential partner, and in business – to sell more goods to the client [8].

For all the simplicity of such a scheme, a poorly thought out proposal will certainly return users to the first stage - finding a suitable partner. At the same time, they will have to spend significant resources on building new relationships.

Thus, in the chain of establishing productive relationships, the process of collecting information becomes the most important link, on which, in fact, the desired result depends.

With the development of technology, the process of obtaining data evolves.

At the single-channel stage, one communication channel is used for interaction between two subjects, for example, e-mail, within which the correspondence takes place. In this case, if necessary, you can always raise the history of relationships by returning to earlier letters, i.e. information is contained within one branch, which greatly simplifies the work with it.

With the development of communication methods, the communication model moves to the multi-channel stage, which is characterized by the use of a large number of communication channels, but within the framework of this model, communication is also conducted between two subjects. For example, when communicating with your wife on the phone, through instant messengers and e-mail, you can easily keep in your head what you agreed with her. However, as the number of transmitting and receiving entities increases, problems inevitably begin to arise in the multi-channel model. The main one - the impossibility of spreading information between different subjects - creates the prerequisites for the transition to the next stage with a more effective communication model.

At the omni-channel stage, the received information is used by different subjects. In this case, the transfer of data from one to another should be carried out quite simply. If, within the framework of a multi-channel, information is processed by one person without any particular difficulties and its constant updating is available for analysis, then at the omni-channel stage you need some kind of external data storage, access to which must be provided

to a large number of subjects. In the future, they will become the consumers of this information.

So, let's define what an omnichannel is. This is an individual approach to the client, regardless of the channel of calls and the employee with whom he interacts in the company. The trend towards building relationships in the omnichannel format is quite young. In 2015, Econsultancy conducted an online survey of 2,000 UK e-commerce firms. As a result, it turned out that only 5% of respondents have fully integrated channel management, and this is what allows them to systematically influence the behavior of their customers.

The first difficulty lies in the processing of incoming requests. As part of the research conducted for Google3, the work of 500 online stores from the existing customer base was studied. At the same time, companies were considered that sell goods and services using their website as a virtual showcase. On such a "showcase", a customer of an online store can select the goods or services he needs, move them to an electronic basket and pay.

Another difficulty arises with communication channels. In the process of communication, the client goes through different stages:

- 1) the formation of a need - the client has a need for a certain product or service;
- 2) search for a solution - the client is looking for a solution that would satisfy his need in the best way;
- 3) collection of information - the client collects information about a product or service;
- 4) initial acquaintance - the client gets acquainted with the product / service;
- 5) purchases - the client purchases a product / service;
- 6) use - the client uses the purchased product / service;
- 7) forming a positive impression - the client successfully satisfies his need with the purchased product / service;
- 8) recommendations to friends.

At the stages of initial acquaintance, acquisition and use, the client communicates with the supplier company in different ways: through a phone call, chat on the site, an application from the site, a callback form. All of the above communication methods make it possible to substantially approach the omni-channel format in terms of customer

identification. At the same time, it is important to understand that identification is necessary not only for direct contact. It is also important if the client suddenly visits the "Tariffs" section on the supplier's website. Such a visit may indicate that he is making a price comparison in the market, i.e. considers an alternative supplier of goods / services.

The main problem lies in the identification of the client who physically interacts with company representatives when visiting a retail outlet or during communication with a courier when delivering goods. Many people think that this kind of communication is beyond the edge of internet marketing, and they are wrong. At each point of contact, it is necessary to collect additional information that allows you to more accurately customize the Internet tools used, i.e. more carefully define your audience segment.

Service Cloud helps companies overcome barriers to customer satisfaction. The platform provides analytics and productivity tools to help agents, managers and mobile workers deliver personalized, proactive customer service across all channels and devices. As a flexible and scalable platform, Service Cloud also easily adapts to the growing and changing needs of enterprises and customers.

To launch an omnichannel contact center, a user needs to provide the right information to the right people at the right time. Salesforce only has a product. Omni-Channel, a comprehensive Service Cloud customer service solution, switches agents in real time, right from the Salesforce console. In the omni channel, everything happens to objects, which are just a buzzword for anything that can be directed to your agents. Cases, chats, leads, and social media posts are all examples of objects.

Delivery of objects depends on the scope of work and the capabilities of the agent. As a manager, you can use Omni-Channel Routing to set the size and priority of various objects. Salesforce stores objects in queues until the agent is ready to accept them. When you associate a queue with a routing configuration, Salesforce directs work on that queue to agents based on the routing model, size, and priority that you set. Omni-Channel Presence allows your agents to indicate when they are available to work and for which service channels. Managers can control all of these features using Omni-Channel Supervisor, so they always know where their team is in relation to their service level goals.

Omni-Channel Supervisor allows users to track agent activity, queue and operation with tabs that give them an overview of how their agents are performing. Users also can use filtering and sorting to find what they need and then drill down to get detailed information about specific agents, queues, or work items - all from one console, as shown on the figure 2.4.

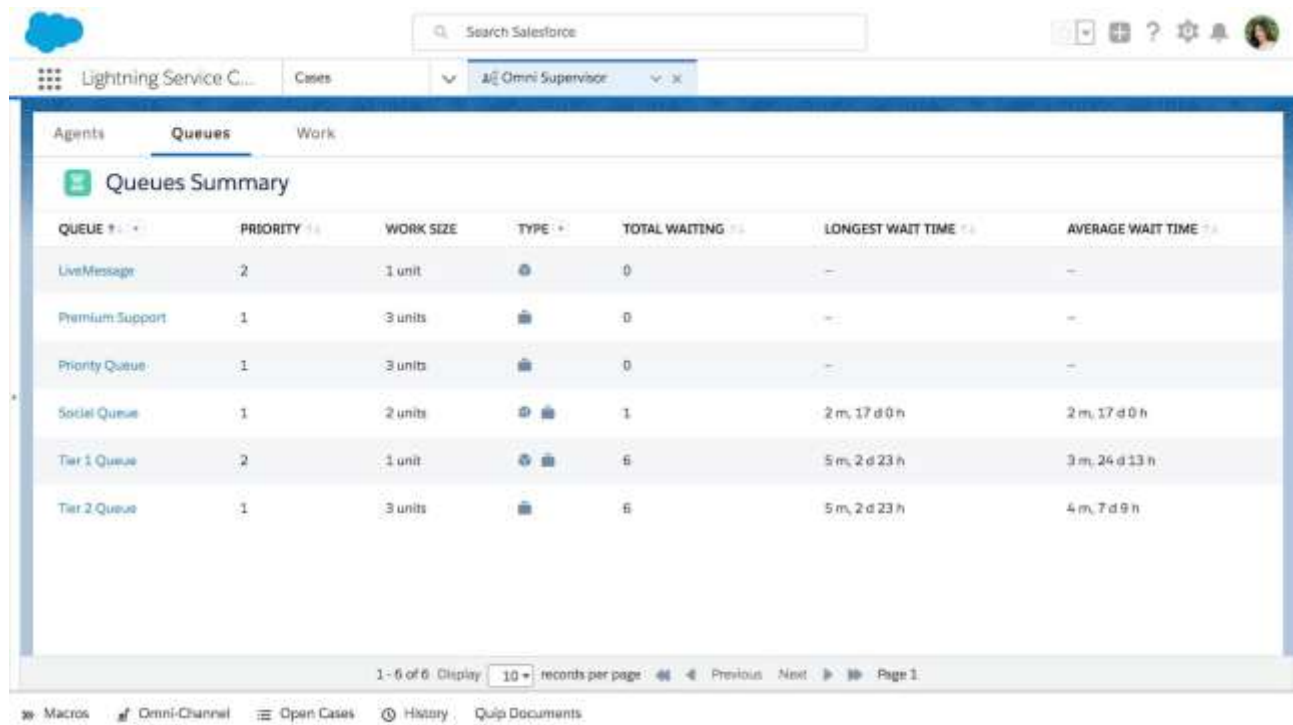


Fig. 2.4. Monitoring activities in Salesforce Omni-Channel

Conclusions on the Second Part

Online services for sending sms-mailings, e-mail-mailings, as well as mailings on popular messengers significantly simplify the work of informing clients and customers about important company events. Internet platforms for sending SMS messages have become an indispensable tool for employees responsible for marketing and communication with customers, especially in terms of CRM platforms.

For the most part, SMS services offer many different filters and settings, thanks to which users can take into account time zones, regional affiliation of the subscriber, send mailings to certain operators, and much more. Tariffication for services depends on the

number of SMS messages sent, the amount of top-up, as well as on the choice of the channel for sending.

Twilio is a platform for creating sms messages, calls and other communications. With its help, it is possible to set up automated mailings and receive response messages, make voice calls and video calls, receive verification codes for authorization on various portals, create call centers and chat bots.

Omni-channel in Salesforce is a powerful tool, which potentially could be used in terms of communication with users, but it doesn't provide an integration with any external to Salesforce communication channels. Thou, Omni-channel could be used for the distribution of the clients between users inside of the platform.

PART 3

NOTION OF API, REST AND CLIENT-SERVER ARCHITECTURES

3.1. Notion of API

Let's start with the basics: what is an API. The abbreviation stands for Application Programming Interface, or an interface for programming applications. The name, it seems, speaks for itself, but it is better to consider a more detailed explanation.

As already was mentioned, the API is, first of all, an interface. An interface that allows developers to use some ready blocks to create applications. In this case, the API can be opened in the library for working with a smart home.

In the case of web applications, the API can provide data in a format other than the standard HTML, which makes it convenient for developers to use when writing their own applications. Third-party public APIs most often provide data in one of two formats: XML or JSON. In case the developer decides to make an API for his application, he would probably use JSON which is much more concise and easy to use than XML.

For example, Github has its own API, which can be used by other developers. The way of its usage by developers depends on the capabilities that the API provides and on how well the imagination of the developers works.

The API allows, for example, to obtain information about the user, his avatars, readers, repositories, and many other useful and interesting data.

If to consider, for example, API of Twitter – it is the interface of this service that can give some information about user, his readers and those who is read by him, and so on. This can be only a small part of the capabilities that can be implemented using the API of a third-party service or creating own implementation.

There are several situations in which developer may want to create an API for his own lovingly written and refactored application.

The first of them is mobile app. A lot of mobile applications for various services work with the usage of the API of these same services. So developer describes the API, makes a

simple mobile application and the client with the smartphone will receive information into his device through the API. It is convenient, it is reasonable, it makes sense.

Another reason is open source. Everything gets better if to use open source. Actually, if the application has a certain audience that makes use of the application, it is quite reasonable to make some advantage of it. And to the benefit of the audience, of course, too. Create an API, with the help of which the users will be able to create new clients for the application, new services based on it and, perhaps, discover its new features.

And even one more reason is maximum separation of frontend and backend. For example, when using front-end frameworks.

Creating a complete API for the application is only half the work. And there is still a question about how users intend to access the API and how they would contact with it.

The first thing that comes to mind is the usual series of HTTP requests to get the right information, and this is the wrong answer. The most obvious way in this case is not the most convenient and simple. It would be much more reasonable to create a special library for working with the interface, which will describe all the necessary ways to receive and send information using the API.

Once again, the Github is used for bringing an example: to work with the API of this excellent service (and its interface provides the most extensive possibilities) several libraries have been created in various languages. In the documentation for such libraries any interested developer can find all the necessary ways to get information from GitHub and send it back through the service API [9].

Thus, if developer creates his own API, perhaps he should take care to create the same libraries for working with it in the most common languages. And to be prepared that with a certain level of demand for his application, someone else can create their own library to work with the API. This is normal.

So, as API makes life easier for both developers and users, an API is used in this project - Twilio API.

3.2. Client-Server Model

First of all, it is needed to deal with the concept of client-server interaction, then to consider why the webmaster needs to understand the client-server model. Next, there is a description of the architecture of applications that operate on the client-server principle and finally consideration of the advantages and disadvantages of this model.

3.2.1. The Concept of Client-Server Interaction

Millions of people surf the Internet every day to read news, chat with friends, get useful information, make a purchase or pay a bill. But most of ordinary users do not even know how and with the help of what they do all this, and in fact they don't even need it [10].

As the name implies, the client-server concept involves two parties: the client and the server. Here, everything is as in life: the client is the customer of a service, and the server is the service provider. The client and server are physically programs, for example, a typical client is a browser.

The client and server interact with each other on the Internet or on any other computer network using various network protocols, such as IP protocol, HTTP protocol, FTP, and others. There are actually a lot of protocols, and each protocol allows a user to provide this or that service. For example, using the HTTP protocol, the browser sends a special HTTP message, which indicates what information and in what form it wants to receive from the server, the server, having received such a message, sends the browser a response message similar in structure (or several messages), which contains the desired information, usually in a form of HTML page.

Messages that clients send are called HTTP requests. Requests have special methods that tell the server how to handle the message. And the messages that the server sends are called HTTP responses; they contain, in addition to useful information, special status codes that allow the browser to find out how the server understood its request.

Now there was schematically described how the client and the server interact at the seventh level of the OSI model, but in fact this interaction occurs at all seven levels. When

a client sends a request, the message is packaged, it is possible to imagine that the message is wrapped in seven wrappers (although there may be much more or less), and when the message is received by the server, it starts to unwrap these wrappers.

It is also worth noting that the client-server interaction is based on the principle that the client starts this interaction, the server only responds to the client and informs whether it can provide the service to the client and if so, under what conditions. Client software and server software are usually installed on different machines, but they can also work on the same computer.

This interaction concept was developed primarily to share the burden between the participants in the information exchange process, as well as to separate the program code of the supplier and the customer. It is possible to see a simplified client-server interaction scheme on the figure 3.1.

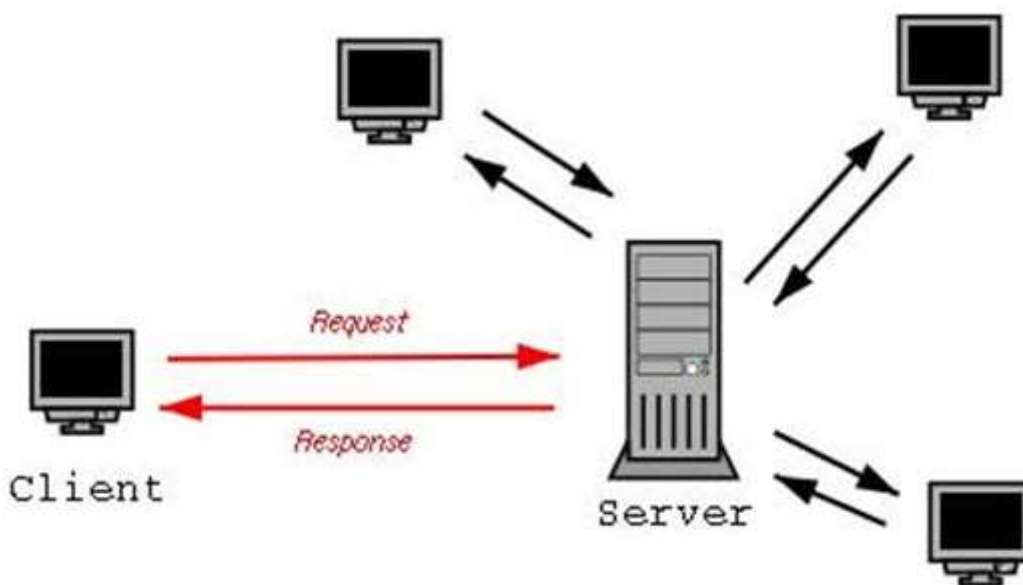


Fig. 3.1. Client-server interaction scheme

It is evident that several clients can access the same server at once (indeed, there can be several visitors on one site). It is also worth mentioning that the number of clients that can simultaneously interact with the server depends on the capacity of the server and on what the client wants from the server.

Many network protocols are built on a client-server architecture, so they are usually based on the same or similar principles of interaction, and we see the difference only in the

details, which are due to the characteristics and specificity of the area for which a particular network protocol was developed.

Let's now answer the question: “why does a webmaster or a web developer should understand the concept of client-server interaction.”. The answer, of course, is obvious. To do something with own hands it is needed to understand how it works. In order to make a website and make it work correctly on the Internet, it is needed to understand how the Internet works.

Most network protocols have a client-server architecture. For example, the webmaster or web developer will be interested in the protocols of the seventh and sixth level of the OSI model. It is important for network administrators to understand how they interact at levels five through two. For communication engineers, protocols from the fourth to the first level of the OSI model are of the greatest interest.

Therefore, if someone really wants to be a professional in the field of web, then first he needs to understand how interaction takes place in the network (at the seventh level), and only then begin to learn the tools that will allow him to create websites.

3.2.2. Client-Server Architecture

The client-server architecture defines only the general principles of interaction between computers, the details of the interaction determine the various protocols. This concept tells people that they need to divide the machines on the network into client machines, which always need something and servers, which give what is needed. In this case, the client always starts the interaction, and the rules by which the interaction occurs describes the protocol. Usual structure of the client-server architecture can be seen on the figure 3.2.

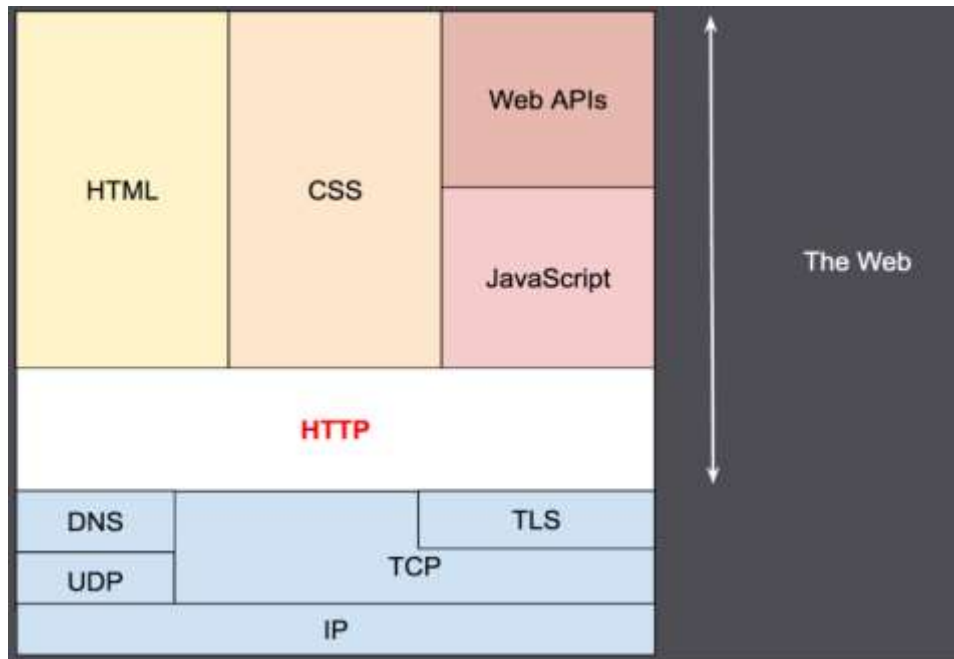


Fig. 3.2. Structure of the client-server architecture

There are two types of client-server interaction architecture: the first is called the two-tier client-server interaction architecture, the second is the multi-tier client-server architecture.

The principle of the two-tier client-server interaction architecture is that the request is processed on a single machine without the use of third-party resources. The two-tier architecture imposes strict server performance requirements, but at the same time is very reliable. It's possible to see the two-level client-server interaction model in the figure 3.3.

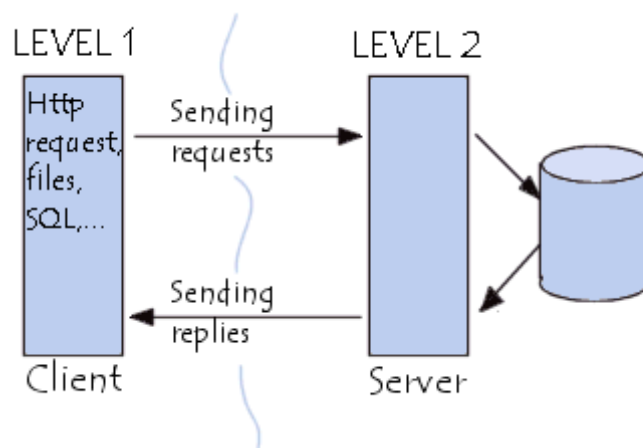


Fig. 3.3. Two-tier client-server interaction model

Here it is can be clearly seen that there is a client (1st level), which allows a person to make a request, and there is a server that processes the client's request.

If we talk about the multi-level client-server interaction architecture, then we can take as an example any modern DBMS. The essence of the layered architecture is that the client's request is processed by several servers at once. This approach can significantly reduce server load due to the fact that there is a distribution of operations, but at the same time, this approach is not as reliable as the two-tier architecture. It's possible to see an example of a multi-tier client-server architecture in the figure 3.4.

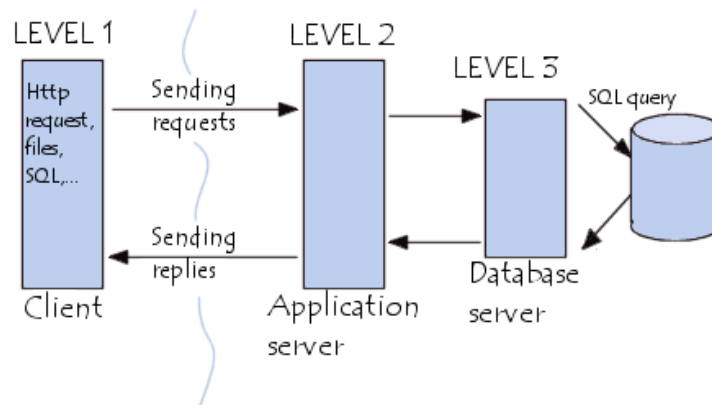


Fig. 3.4. Multi-tier client-server interaction architecture

A typical example of a three-tier client-server model: speaking in the context of database management systems, the first level is the client, which allows to write various SQL queries to the database. The second level is the database engine that interprets requests and implements the interaction between the client and the file system, and the third level is the data storage.

If to look at this architecture from the perspective of the site, then first level can be considered a browser with which a visitor comes to the site, the second level is a bunch of Apache + PHP, and the third level is a database. Speaking quite simply, PHP doesn't do anything else, except it drives the strings and databases to the screen and back to the database.

3.2.3. Advantages and Disadvantages of Client-Server Architecture

The advantage of the client-server interaction model is that the program code of the client application and the server code are separated. If we are talking about local computer

networks, the advantages of client-server architecture include reduced requirements for client machines, since most of the computational operations will be performed on the server, and the client-server architecture is quite flexible and allows the administrator to make the local area network more secure.

The disadvantages of the client-server interaction model include the fact that the cost of server hardware is much higher than the client. The server must be serviced by a specially trained person. If a server is in a local network, then clients will not be able to work (for example, server capacity is not always enough to satisfy client requests, if you have ever worked with billing systems, then you know that waiting time response from the server can be very large).

As a conclusion, it should be clearly emphasized that the client-server architecture does not divide the machines into only the client or only the server, but rather allows to distribute the load and divide the functionality between the client part and the server part.

3.3. HTTP Protocol

The protocol of the site, also known as the data transfer protocol, is a peculiar set of rules that describe the order, the peculiarities of interaction between two or more devices connected to the same network and communicating. Without a data transfer protocol, devices connected to the Internet simply could not agree among themselves who sent the requests and in what order, who handled them and how, what this or that error means. The site protocol helps to settle all this and allows all users (clients) to interact with servers (hosts) without any problems [11].

There are many different data transfer protocols, many of which are outdated or simply unpopular. Others are used for very specific, highly specialized tasks and will be of little use to the average user.

The most basic protocols are HTTP and HTTPS. These 2 protocols are used for everything that interests the average user. Web browsing, interaction with web applications, interactive sites. In general, everything that arises in the window of a browser somehow

works precisely because of these two data transfer protocols. The GET request structure can be found on the figure 3.5.

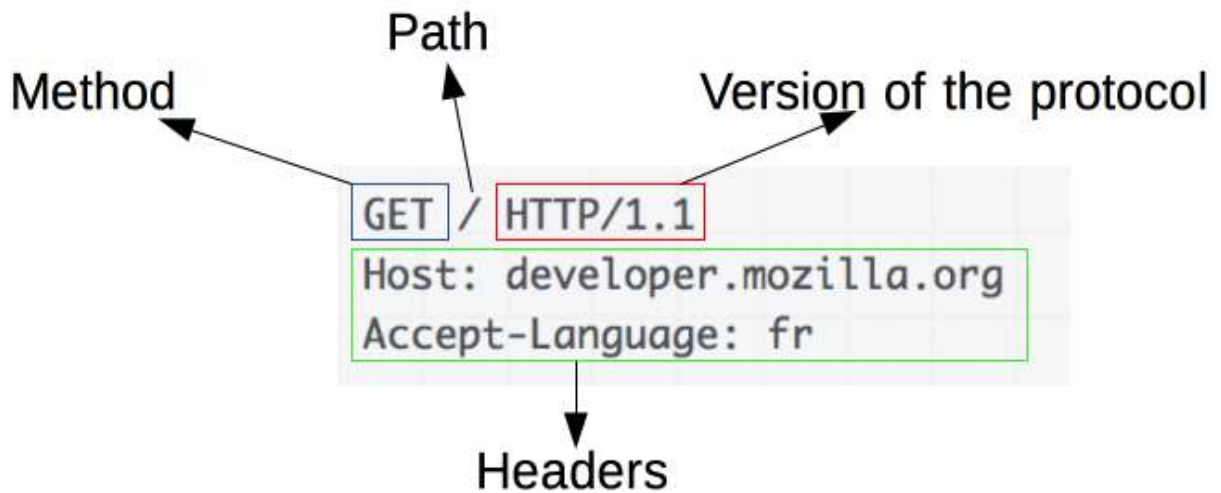


Fig. 3.5. An example of GET request structure

HTTP is a data transfer application protocol. The principle of its work is complicated, if to understand it thoroughly, and it is extremely simple, if to delve into its essence. Work through this protocol is carried out according to the client-server scheme. There is a server that, in passive mode, constantly waits for a connection to it. This connection with him, sooner or later, will be established by the client, that is, the machine user interface by the Internet. The client wants to get something from the server: get a page, open a picture, download a song. To report exactly what the client wants, the user sends requests that the server expertly handles. The server is able to handle user requests thanks to the instructions with which the HTTP protocol supplied it. If the request cannot be processed, the server knows what error it should give.

HTTPS is an enhanced version of HTTP. The main difference is that now requests from the client are sent not in a bare form, but encrypted using the cryptographic mechanisms of SSL and TLS. Using this protocol allows to achieve such a result that the request from the client can be really read only on the server side, and can in no way be intercepted by a third party somewhere in the middle. This third party can be hackers, Trojans, unscrupulous providers, special services of any countries, and so on. By intercepting a unprotected request sent via the HTTP protocol, the thief can modify it, can simply find out valuable information and use it for personal gain. Currently, the HTTPS

protocol is a completely uncompromised method of interaction of devices on the Internet, and can withstand any hacker attack, thereby ensuring the most secure interaction of devices on the network. An example of successful HTTP response can be found on the figure 3.6.

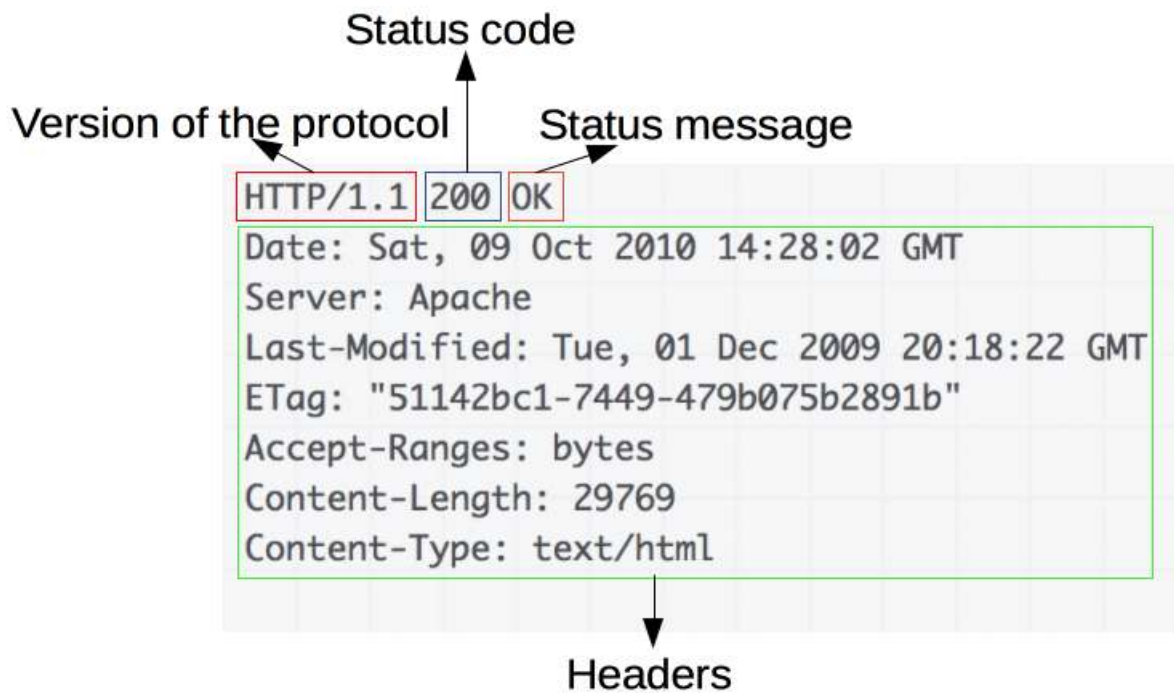


Fig. 3.6. An example of successful HTTP response

Answer to the question about which protocol is better pretty obvious. HTTP protocol is outdated. When it was created, it was akin to a technical revolution, and its use raised the convenience of user communication with servers to an unusually high level. But times are changing and now it is vulnerable. The HTTPS protocol is the safest way to communicate between devices today. It cannot be hacked, circumvented, compromised, today this data transfer protocol is invulnerable. It is impossible to determine whether this will always be the case, but at the moment most search engines mark sites that still work via the HTTP protocol as unreliable and inform users that they may be in danger on this site. And this is not just the case, it is necessary to ensure the safety of users. If developer have not translated his website to the HTTPS data transfer protocol, then he should do this as soon as possible. So, he will increase the confidence of search engines when promoting a site with natural links, he can use a large number of services and make using the site safe and convenient.

3.4. Representational State Transfer Architecture

REpresentational State Transfer is architecture, i.e. principles of building distributed hypermedia systems, which in other words is called the World Wide Web, including universal methods of processing and transferring resource states over HTTP. REST has practically supplanted all other approaches today, including design based on SOAP and WSDL.

Benefits of REST:

1) Lack of additional internal layers, which means data transfer in the same form as the data itself. Those. data is not wrapped in XML like SOAP and XML-RPC do, AMF is not used like Flash does, etc. The data itself is simply given.

2) Each piece of information (resource) is uniquely identified by a URL - this means that the URL is essentially the primary key for the item. Moreover, it does not matter at all in what format the data is located at the address - it can be HTML, jpeg, or a Microsoft Word document.

3) How resource information is managed is entirely based on the data transfer protocol. The most common protocol is of course HTTP. For HTTP, the action on data is set using the following methods: GET (receive), PUT (add, replace), POST (add, change, delete), DELETE (delete). Thus, CRUD (Create-Read-Update-Delete) actions can be performed with all 4 methods, or only with GET and POST.

For a distributed system to be considered RESTful architecture (Restful), it must meet the following criteria [12]:

1) Client-Server. The system should be divided into clients and servers. The separation of interfaces means that, for example, clients are not concerned with the storage of data that remains within each server, so the portability of the client code is improved. Servers are not associated with user interface or state, so servers can be simpler and scalable. Servers and clients can be replaced and developed independently as long as the interface is not changed.

2) Stateless. The server should not store any information about clients. The request must contain all the necessary information to process the request and, if necessary, identify the client.

3) Cache. Each response should be flagged as cacheable or not, to prevent clients from reusing stale or invalid data in response to further requests.

4) Uniform Interface. A single interface defines the interface between clients and servers. This simplifies and decouples the architecture that allows each piece to evolve on its own.

The four principles of a unified interface are identification of resources (resource based). In REST, a resource is anything that can be given a name. For example, a user, an image, an item (T-shirt, hungry dog, current weather), etc. Every resource in REST must be identified by a stable identifier that does not change when the state of the resource changes. An identifier in REST is a URI.

Manipulation of resources through representations. (Manipulation of resources through views). A REST view is used to perform actions on resources. A resource view represents the current or desired state of a resource. For example, if the resource is a user, the view can be an XML or HTML description of that user.

Self-descriptive messages (self-documenting messages). Self-descriptiveness means that a request and a response must contain all the necessary information to process them. There should not be additional messages or caches to handle a single request. In other words, there is no state that persists between requests for resources. This is very important for scaling the system.

HATEOAS (hypermedia as the engine of application state). The status of a resource is passed through the body content, query string parameters, request headers, and the requested URI (resource name). This is called hypermedia (or hyperlinks with hypertext). HATEOAS also means that, if necessary, links can be contained in the response body (or headers) to support the URI, retrieve the object itself, or the requested objects.

5) Layered System. In REST, it is allowed to divide the system into a hierarchy of layers, but with the condition that each component can only see the components of the

immediately next layer. For example, if you call the PayPal service and it, in turn, calls the Visa service, you do not need to know anything about calling the Visa service.

6) Code-On-Demand (optional). REST allows client-side code or program to be loaded and executed.

Servers can temporarily expand or customize the client's functionality by passing it logic that he can execute. For example, it can be compiled Java applets or client scripts in Javascript

The REST architecture itself is not tied to specific technologies and protocols, but in the realities of the modern web, building a RESTful API almost always implies the use of HTTP and some common resource representation formats, such as JSON, or, less popular today, XML.

From the point of view of a RESTful service, an operation (or a service call) is idempotent when clients can make the same call repeatedly with the same result on the server. In other words, creating a large number of identical queries has the same effect as one query. Note that while idempotent operations produce the same result on the server, the response itself may not be the same (for example, the state of the resource may change between requests).

The PUT and DELETE methods are idempotent by definition. However, there is one nuance with the DELETE method. The problem is that a successful DELETE request returns a status of 200 (OK) or 204 (No Content), but for subsequent requests it will return 404 (Not Found) all the time. The status on the server after each DELETE call is the same, but the responses are different.

The GET, HEAD, OPTIONS and TRACE methods are defined to be safe. This means they are for information purposes only and should not change the state of the server. They should have no side effects, except for harmless effects such as logging, caching, displaying banner ads, or increasing the web counter.

By definition, safe operations are idempotent since they lead to the same result on the server. Safe methods are implemented as read-only operations. However, security does not mean that the server has to return the same result every time. From the point of view of a RESTful service, an operation (or a service call) is idempotent when clients can make the

same call repeatedly with the same result on the server. ... In other words, creating a large number of identical queries has the same effect as one query. Note that while idempotent operations produce the same result on the server, the response itself may not be the same (for example, the state of the resource may change between requests).

The PUT and DELETE methods are idempotent by definition. However, there is one nuance with the DELETE method. The problem is that a successful DELETE request returns status 200 (OK) or 204 (No Content), but for subsequent requests it will return 404 (Not Found) all the time. The state on the server after each DELETE call is the same, but the responses are different ...

The GET, HEAD, OPTIONS and TRACE methods are defined to be safe. This means they are for information purposes only and should not change the state of the server. They should have no side effects, except for harmless effects such as logging, caching, displaying banner ads, or increasing the web counter.

By definition, safe operations are idempotent since they lead to the same result on the server. Safe methods are implemented as read-only operations. However, security does not mean that the server has to return the same result every time.

Conclusions on the Third Part

APIs are an essential component of software design and they exist at every level of the software stack. They provide a way to define and manage abstractions by telling people what to can do with software components. Well-designed APIs support efficient, fluid, and effortless adoption and use, while poorly designed APIs tend to cause headaches every time they're used.

REST is an acronym for Representational State Transfer. It is an architectural style of interaction of components of a distributed system in a computer network. Simply put, REST defines the style of communication (data exchange) between different system components, each of which can be physically located in different places.

This architectural style is a consistent set of constraints to consider when designing a distributed system. These constraints are sometimes referred to as REST principles.

There are many different data transfer protocols, many of which are outdated or simply unpopular. Others are used for very specific, highly specialized tasks and will be of little use to the average user.

The most basic protocols are HTTP and HTTPS. These 2 protocols are used for everything that interests the average user. Web browsing, interaction with web applications, interactive sites. In general, everything that arises in the window of a browser somehow works precisely because of these two data transfer protocols.

Most network protocols have a client-server architecture. For example, the webmaster or web developer will be interested in the protocols of the seventh and sixth level of the OSI model. It is important for network administrators to understand how they interact at levels five through two. For communication engineers, protocols from the fourth to the first level of the OSI model are of the greatest interest.

In the Messaging App project the Twilio API is actively used for creation of the messages, sending them to users and following their current statuses. REST from the side of Twilio to create a webhook which gives an ability to update the statuses of any outgoing messages.

PART 4

PROGRAMMING LANGUAGES AND FRAMEWORK FOR THE MESSAGING APP DEVELOPMENT

4.1. Apex as a Salesforce programming language

Apex is a proprietary language developed by Salesforce.com. Officially defined, Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on a Force.com platform server in conjunction with Force.com API calls.

It has Java-like syntax and acts like database stored procedures. It allows developers to add business logic to most system events, including button clicks, linked record updates, and Visualforce pages. Apex code can be triggered by web service requests and triggers on objects. Apex is available in Performance Edition, Unlimited Edition, Enterprise Edition, and Developer Edition.

Features of Apex as a language:

1) Apex has built-in support for DML operations like INSERT, UPDATE, DELETE as well as DML exception handling. It supports inline SOQL and SOSL query processing that returns a sObject recordset. We will explore sObject, SOQL, SOSL in detail in the next chapters.

2) Apex is easy to use as it uses syntax like Java. For example, variable declaration, loop syntax, and conditional statements.

3) Apex is data oriented and designed to execute multiple queries and DML statements at the same time. It issues multiple transaction statements on the database.

4) Apex is a strongly typed language. It uses a direct reference to schema objects such as sObject, and any invalid reference will quickly fail if deleted or is of the wrong data type.

5) Apex works in a multi-user environment. Consequently, the Apex runtime is designed to carefully guard against exploitable code, preventing it from monopolizing

shared resources. Any code that violates the constraints will fail with easy-to-understand error messages.

6) Apex is being updated as part of the Salesforce releases. We don't need to update it manually.

7) Apex provides built-in support for building and running unit tests, including test results that show how much code is covered and which parts of your code might be more efficient.

Apex working structure is as shown in the figure 4.1 - Apex is fully on-demand Force.com Platform.

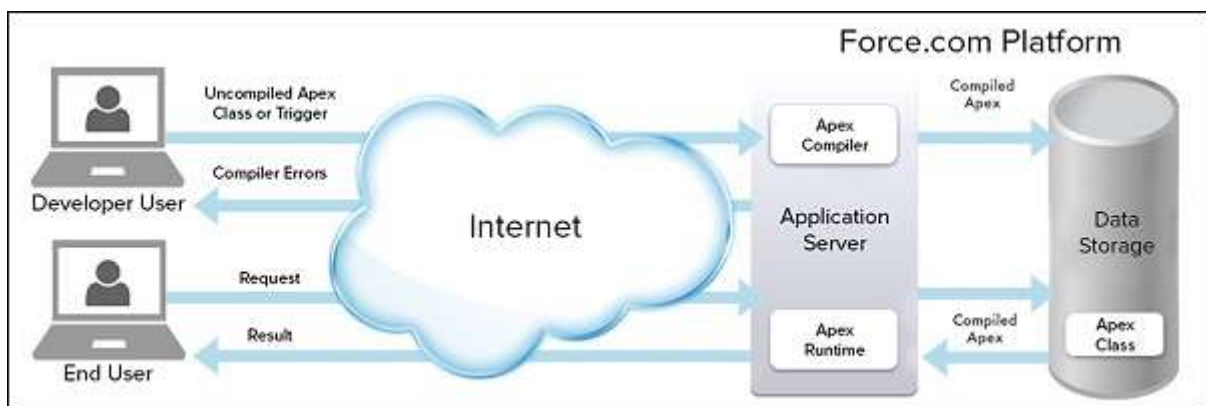


Fig. 4.1. Apex in terms of Force.com platform

There are two sequences of actions when the developer saves the code and when the end user takes some action that invokes the Apex code, as shown below:

When a developer writes and stores Apex code on the platform, the platform application server first compiles the code into a set of instructions that can be understood by the Apex runtime interpreter, and then stores those instructions as metadata.

When an end user launches Apex execution by clicking a button or opening a Visualforce page, the platform app server extracts the compiled instructions from the metadata and sends them through the runtime interpreter before returning the result. The end user sees no difference in runtime compared to a standard application framework request.

Since Apex is a proprietary Salesforce.com language, it does not support some of the features that a regular programming language does. Below are some of the features that Apex does not support.

User cannot change the standard functionality provided by SFDC, and it is also impossible to prevent the standard functionality from being executed, cannot change the standard functionality provided by SFDC, and it is also impossible to prevent the standard functionality from being executed. Creating multiple streams is also not possible as we can do it in other languages.

Apex developers also cannot change the standard functionality provided by SFDC, and it is also impossible to prevent the standard functionality from being executed, cannot change the standard functionality provided by SFDC, and it is also impossible to prevent the standard functionality from being executed, it is also not possible to create multiple threads as we can do it in other languages.

4.2. SLDS as a Markup and Styling tool

The Salesforce Lightning Design System (SLDS) Content Library is under active development to enable Salesforce developers to create a consistent look and feel across all Salesforce-related applications, while adhering to CSS best practices and conventions.

To achieve this goal, it was decided to use very specific naming methods. This allows developers to keep their codebase flat, with low specificity, and keeps them from a specificity war that starts with frustration and ends! Important. Although they base their naming on the BEM method, they have some additions of their own base, described below.

BEM is a well-known way of naming components - block, element, modifier. For those unfamiliar or in need of a quick update, let's take a quick look at how BEM works. As an example, we will build a house component.

The block represents the name of the main component. If you were building a component at home, the class name would be `.house`. Any properties you want to include for all houses will be included in the base class `.house`. An element represents a part of a component and is separated by two underscores. The door of the house would be represented by the class `.house__door`. A window would be `.house__window`.

The components built with the custom-scoped SLDS will need your custom scoped class at the highest level of the DOM where the components are included. In some cases this

will be a wrapper at the component level, in other cases you may wrap several components. Do not place your custom class on the body element of the application if you are including non-SLDS components within that area as well. This would cause the non-SLDS components to be scoped and could override expected styles with unwanted effects.

When a component has a variety of states, we add a class to indicate the state the component, or certain portions of the component, are in. Some examples are: `.slds-is-selected`, `.slds-is-active`, `.slds-is-expanded`, `.slds-is-nested`, `.slds-is-open`, `.slds-has-focus`, `.slds-has-error`, etc.

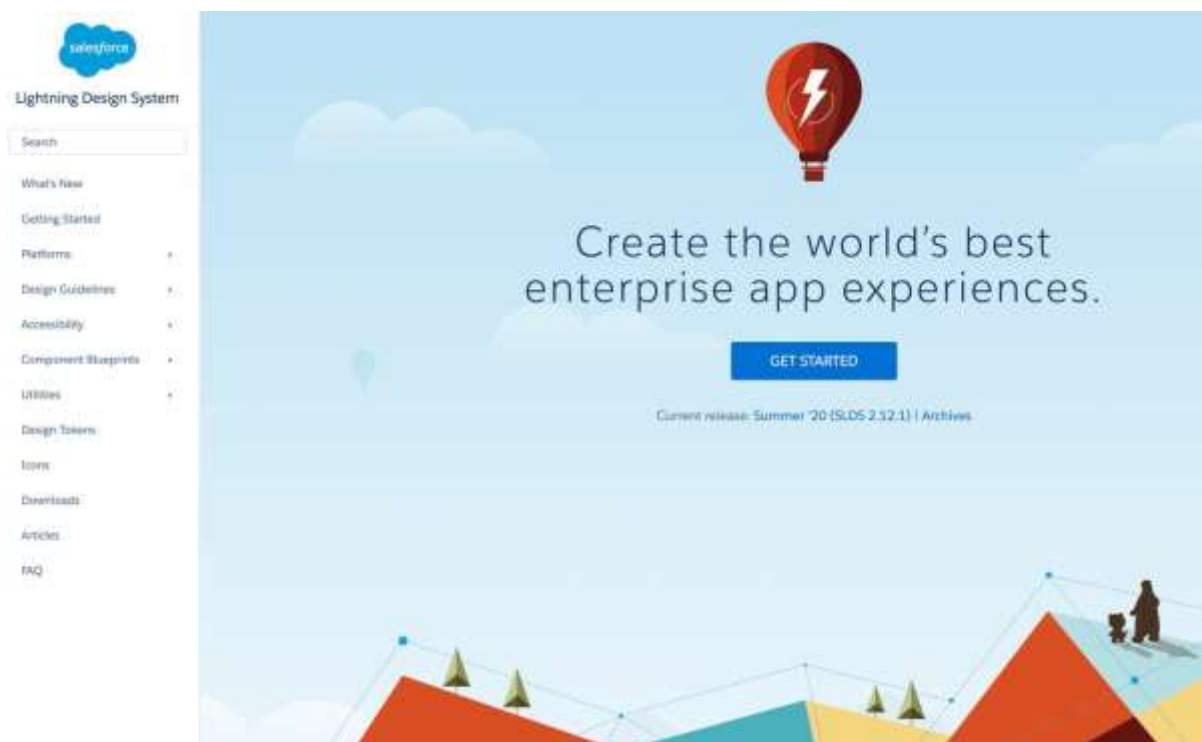


Fig. 4.2. Lightning Design System as guide of usage

The Salesforce User Experience team has been working hard to bring Enterprise UX into the 21st century, and is proud to present the Lightning Design System. The Design System makes it easy for you to build applications that comply with the new Salesforce Lightning look and feel without reverse engineering the UI as custom CSS. In fact using the new Design System markup results in pages which have the Lightning look and feel without writing any CSS.

Lightning Design System is quite a mouthful, so we're referring to it here as the "Design System".

Developers might have used other similar design systems, such as Twitter Bootstrap or Foundation for building websites. The key benefits of this Design System are:

1) It's tailored for building Salesforce apps. Using the Design System markup and CSS framework results in UIs that reflect the Salesforce Lightning look and feel. The focus on building applications is an important one to keep in mind. The Design System does not over-enforce defaults such as padding and margins, in contrast to some other frameworks which are focused on building marketing pages. The Design System lets you easily specify the exact layouts you require, whilst conforming to the new Lightning UI.

2) It's continuously updated. As long as you're using the latest version of the Design System, your pages are always up to date with Salesforce UI changes.

3) Accessibility is baked in to the CSS framework behind the components.

4) The CSS is fully namespaced with the slds- prefix and scoped with the slds-scope class to avoid CSS conflicts.

4.3. Notion of the Lightning Component Framework

The Lightning Component framework is a UI framework for developing web apps for mobile and desktop devices. It's a modern framework for building single-page applications with dynamic, responsive user interfaces for Lightning Platform apps. It uses JavaScript on the client side and Apex on the server side.

Lightning is a component-based framework that replaces Visualforce. Now many who use Salesforce are gradually switching to Lightning, and Salesforce itself is pushing for this in every possible way.

Salesforce Lightning makes it easy to build responsive apps for any device, includes the Lightning Component Framework, and helpful developer tools.

Below are some of the new features that developers will find useful:

1) Experience. The modern user interface is optimized for fast and responsive design for all devices (the same programming code is optimized for computers, tablets, phones and watch screens).

2) Lightning App Builder. Drag-and-drop development tool for easy application creation and customization. Developers can work with both off-the-shelf components and custom components.

3) Lightning Component Framework. Tools that enable developers to create reusable components, customize the Salesforce1 Mobile app, and build standalone apps. Since the framework is built right into the platform, the user interface is free of the responsive design constraints of Visualforce interfaces.

4) Design System. A responsive CSS framework, style guides, and a modern approach to building user interfaces using design best practices.

5) Lightning Connect. An integration tool that makes it easy for Force.com 1 applications to consume data from external sources that conform to the OData specification.

Lightning Experience is an engaging, modern, customizable user interface that is compatible with many products and devices. The Lightning Platform allows you to quickly build and customize apps for desktop and mobile using powerful drag & drop tools. The Lightning interface is also much faster than the Salesforce Classic interface.

The Lightning Component Framework uses JavaScript on the client side and Apex on the server side. Benefits include a pre-built set of components and interfaces, an event-driven architecture, and a performance-optimized framework.

Advantages of using Lightning Web Framework:

1) Rich component ecosystem and faster development. The Lightning component can be used both in the Salesforce Lightning app and in the Lightning Experience. We don't need to use different metadata for this. Components are encapsulated and their internal characteristics remain private, while their external form is visible to users. This strong separation gives developers the freedom to change internal implementation details and isolate component consumers from those changes.

2) Performance. It is driven by the use of a stateful client and stateless server architecture that relies on client-side JavaScript to manage component metadata and application data. The client contacts the server only when needed; for example, to get more metadata or data. The server only sends the data that the user needs, which ensures maximum efficiency. The framework uses JSON to exchange data between server and

client. It intelligently uses your server, browser, devices and network so that you can focus on the logic and interactions of your applications.

3) Event-oriented programming. In some cases, within the Salesforce ecosystem, SLDS CSS is not yet available. In those cases, for example, when building within Visualforce, Lightning Applications (my.app), or deploying components via Lightning Out, you should scope components built with SLDS. In this case, we have provided a tool for you to create your custom-scoped CSS.

4) Developers who have used frameworks written in languages such as .NET, Java, and JavaScript are most likely familiar with the concepts of event-driven programming. The event-driven app model is required in apps that rely heavily on a graphical user interface, such as those found in the browser, mobile sites, or mobile apps. These applications focus heavily on performing various actions in response to data input and user interaction with the interface. Actions include clicks, button clicks, and scrolls, all of which determine how you interact with your application.

The Lightning Component Framework supports these actions and also provides custom events that let developers pass messages through loosely linked, independent component instances in their application.

The eventing model, in general, relies on an always-present loop that listens for various events and runs associated programming logic to handle each event it encounters.

The Lightning Component Framework uses a JavaScript event loop to listen for events related to user actions in the browser, as well as other events fired from JavaScript code. While tracking UI events is automatically handled by the browser, developers are responsible for registering the events that will listen in each component and for implementing associated JavaScript functions to handle the events.

5) Compatible with devices and browsers. Apps use responsive design and user-friendly interface. The Lightning Component Framework supports the latest browser technologies such as HTML5, CSS3, and touch-actions.

Lightning components are no different from Angular, React, VueJs, or other modern Javascript frameworks. Therefore, if you are familiar with another Javascript MVC

framework, the Lightning Component concepts will seem easy enough to you to understand. On the figure 4.3 it is possible to see the structure of the Lightning Component.

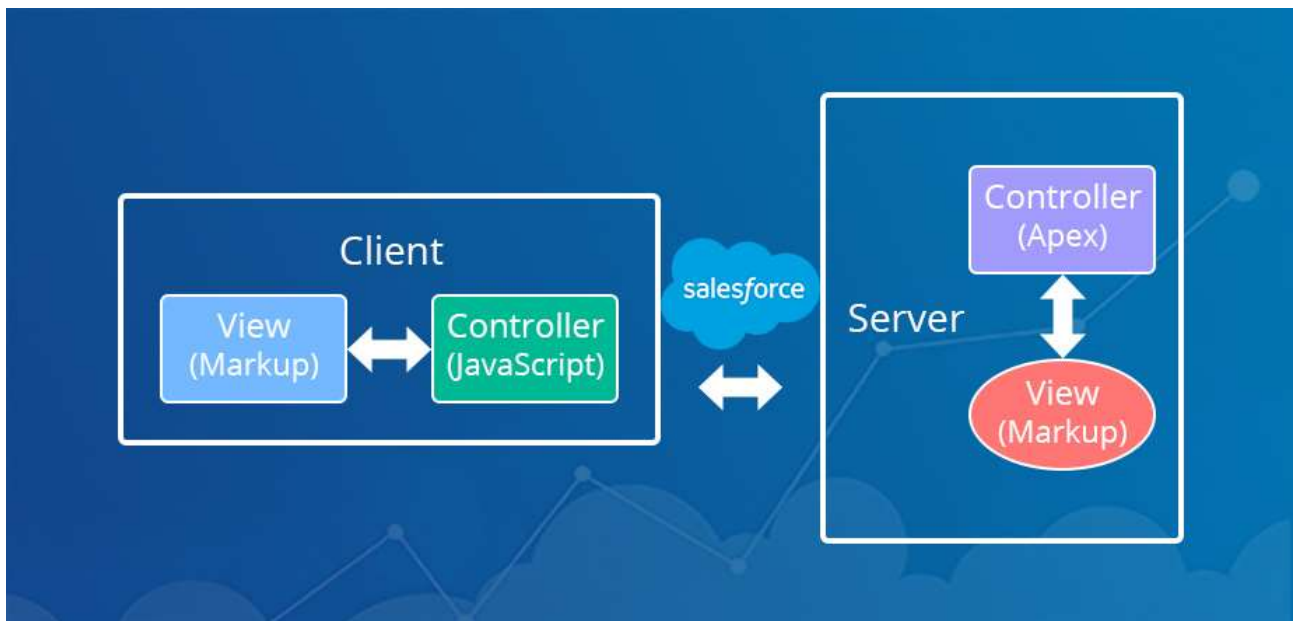


Fig. 4.3. Structure of a Lightning Component

The Lightning Component framework is a framework for developing web apps. That seems understandable. An app framework is a collection of code and services that make it easier for you to create your own custom apps, without having to write all the code yourself. There are lots of different web app frameworks out there, like Ruby on Rails, Grails, AngularJS, Django, CakePHP, and on and on. Developers even got one of their own, Visualforce, that customers know and love. They think Lightning components are pretty special.

Web apps for mobile and desktop devices. Again, that seems pretty easy to grasp. Lightning components were born out of and used to build the Salesforce platform for mobile apps. Mobile is baked into the core of the Lightning Component framework, and it makes developing apps that work on both mobile and desktop devices far simpler than many other frameworks.

The Lightning Component Framework supports these actions and also provides custom events that let developers pass messages through loosely linked, independent component instances in their application.

4.4. Aura vs Lightning Web Components

Salesforce has been on a fantastic journey with Lightning components. Years ago, Salesforce developers were using a unique HTML tag-based Mark-up language known as Visual force pages and Apex as their controller for logic.

In 2014, Salesforce launched the Lightning Component framework supported by the Aura programming framework. Since the web standards offered limited feasibility to build large-scale web applications at the time, Aura came with its component-driven model that allowed developers to build large-scale client applications on the web.

Fast-forward to 5 years from the time of Aura; the web stack has seen an unprecedented level of innovation and standardization that transformed it from being a rudimentary page-rendering platform to a web development platform.

Lightning Web Components (LWC) is a stack of modern lightweight frameworks built on the latest web standards. It is a DOM (Document Object Model), element created through reusable code and is used to generate a dynamic interface without using JavaScript or building a Library. This feasibility makes it quick and seamless, saving the developers a ton of time and effort on the Web Stack. Let's look at some of its remarkable features [13]:

1) Improved performance of the component as most of the code is recognized by the native web browser engine and web stack;

2) Ability to compose applications using smaller chunks of code since the crucial elements that are required to create a component is part of the native web browser engine and web stack;

3) Increase in the robustness of the applications built using LWCs as they are inclusive of the said modern web standards;

4) Parallel interoperability and feasibility to use both Lightning Web Components and Aura components together in the applications with no visible differentiation to the end-users.

Languages like React, Angular and the Lightning Components Framework which use JavaScript are now part of the modern Web Stack. Consequently, the Web Stack no longer requires an additional mid-layer which used to be a challenge for the developers working with Aura as it impacted the speed and performance of the applications [14].

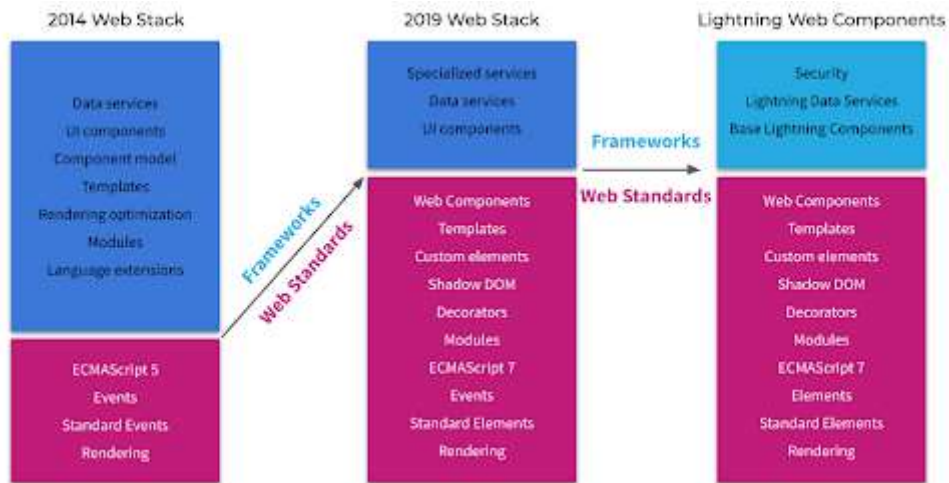


Fig. 4.4. Web Stack Transformation

The addition of above features give more power to our web Stack to create a lightning UI component, It doesn't require a mid-Layer to the browser which impacts our speed and performance. This one is the main reason developers are struggling within Aura and now LWC will work for them.

Aura-based Lightning components are built using both HTML and JavaScript, but LWC is built directly on the Web stack.

Web components can easily interact with the Aura component and can handle the events of one another

A developer who works on LWC is not only using coding skills on a particular framework but in other frameworks like React or Angular, which are based on the Web Stack.

In addition to the latest features, LWC still embraces the features like the Security, Lightning Data Service, and Base lightning components from Aura.

Creating an LWC is fast as it no longer requires the user to download the JavaScript and wait for the engine to compile it before rendering the component.

It is possible to include LWC in Aura, but not the other way around as the generic elements cannot be added to the custom components.

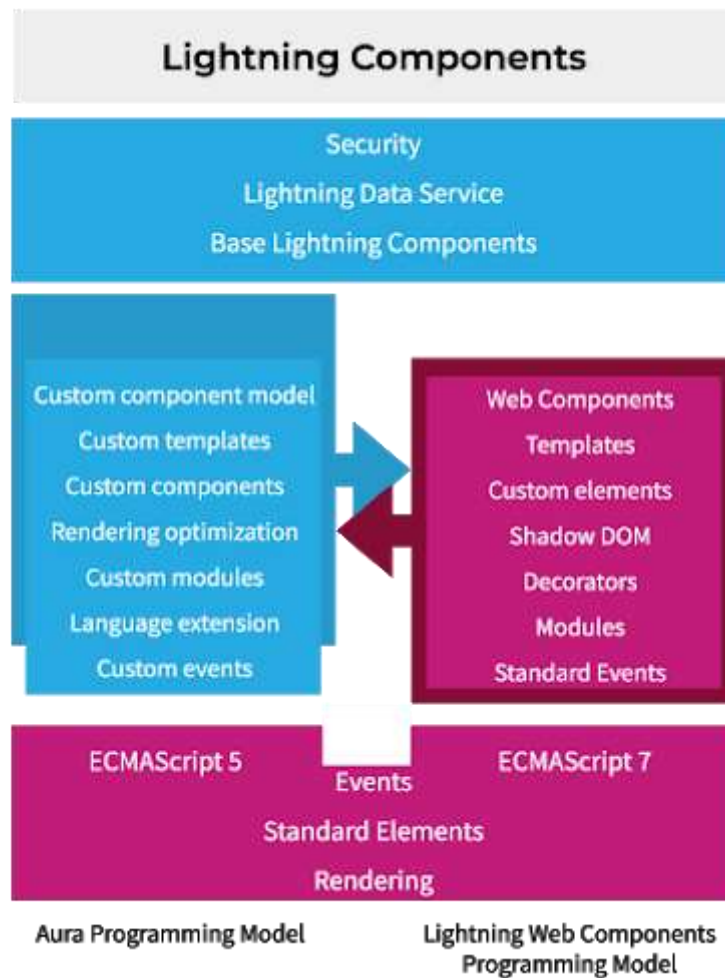


Fig. 4.5. Aura in comparison with LWC

If the organization is already using Aura-based lightning components and the development team is highly skilled with the Aura framework, here's the good news — the Aura Framework is not going anywhere. However, LWCs are the future of lightning development [14].

For the most part, if someone is building new functionality consider using LWCs to benefit from the increased native operation in the browser and performance.

Conclusions on the Fourth Part

Apex is a proprietary language developed by Salesforce.com. Officially defined, Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on a Force.com platform server in conjunction with Force.com API calls.

It has Java-like syntax and acts like database stored procedures. It allows developers to add business logic to most system events, including button clicks, linked record updates, and Visualforce pages. Apex code can be triggered by web service requests and triggers on objects.

The Lightning Component framework is a framework for developing web apps. That seems understandable. An app framework is a collection of code and services that make it easier for users to create their own custom apps, without having to write all the code from scratch. There are lots of different web app frameworks out there, like Ruby on Rails, Grails, AngularJS, Django, CakePHP, and so on. Developers even got one of their own, Visualforce, that customers know and love. They think Lightning components are pretty special.

The Lightning Component Framework uses a JavaScript event loop to listen for events related to user actions in the browser, as well as other events fired from JavaScript code. While tracking UI events is automatically handled by the browser, developers are responsible for registering the events that will listen in each component and for implementing associated JavaScript functions to handle the events.

Lightning Web Components are chosen as a main front-end framework instead of Visualforce and Aura components as it is more convenient, flexible and easy to use due to the lower amount of code needed to be written in comparison with other Salesforce frameworks. The main disadvantage is that LWC is a relatively new framework, but still it seems to be stable enough to complete the tasks in terms of the project.

PART 5

IMPLEMENTATION OF THE MESSAGING APPLICATION

5.1. Preparation of the Data Model

First of all, it is needed to setup the model of the objects, which will be used to hold all the data regarding the messaging app.

Salesforce supports several different types of objects. There are standard objects, custom objects, external objects, platform events, and BigObjects. For our purposes, it is enough to use the two most common types of objects: standard and custom.

Standard objects are objects that are included with Salesforce. Common business objects like Account, Contact, Lead, and Opportunity are all standard objects.

Custom objects are objects that users create to store information that's specific to his company or industry. For Messaging App, it is needed to build several custom objects that store the data about the messages and their relationships with the customers.

Objects in Salesforce are containers for the information, but they also give some special functionality, which makes a usage of objects easier and helps to save some time. For example, when a custom object is created, the platform automatically builds things like the page layout for the user interface [16].

From the list of standard objects it was decided to use only one object – Task due to its ability to be related to literary any other object, which is convenient for generating any kind of reports if needed. It is used for holding the text body of the message, which is also important since these messages will always be displayed in the Activity history, as on the figure 5.1.

If the activity history is needed, it should be turned on by navigating to the settings of the object and by turning it on, otherwise it is not displayed by default and the messages won't be displayed on the record page of the object.

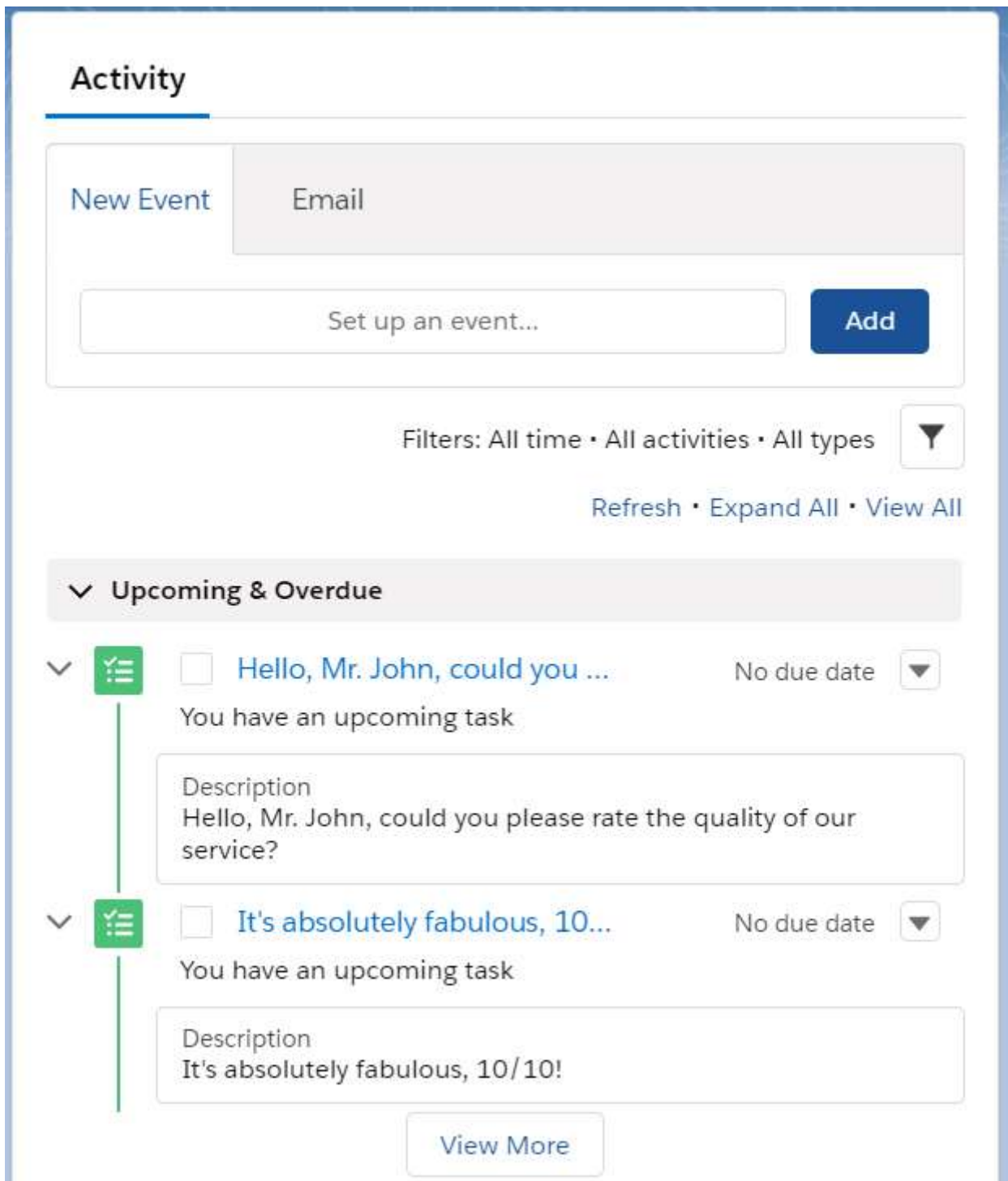


Fig. 5.1. Tasks can be related to objects for displaying a message body

But event this standard needs some modifying as it is needed to add some custom fields to hold all the needed data. On the figure 5.2. the full list of the custom fields is depicted, while the use of these fields is the following:

- 1) Conversation – relationship with the Conversation custom objects that helps to identify the recipient or a sender of the message;

2) Error Description is used for holding any info about the error in delivering a message if the message couldn't be delivered;

3) Message send date holds the actual date of when the message was sent to the customer;

4) Message Sid holds a Twilio ID of the message so that when Twilio sends a callback about the message status, Salesforce understands which exactly message is considered to update the status for;

5) Message Status contains the status of the message, for example Error, Sent, Delivered, Received;

6) Message Status Check Date is invisible for the user fields, but it is used by the behind the scene automated process which checks if Salesforce received any feedback about the status of the message;

7) Message Type can be Outbound or Inbound to understand if the message was sent to customer or if it was received from a customer.

FIELD LABEL	FIELD NAME	DATA TYPE
Conversation	Conversation__c	Lookup(Conversation)
Error Description	Error_Description__c	Text(255)
Message sent date	Message_sent_date__c	Date/Time
Message Sid	Message_Sid__c	Text(34)
Message Status	Message_Status__c	Picklist
Message Status Check Date	Message_Status_Check_Date__c	Formula (Date/Time)
Message Type	Message_Type__c	Picklist

Fig. 5.2. The list of custom fields on the Task standard object

As Task is ready to be in use, the next step is to create some custom objects.

Channel Type – an object that represents a channel type within which a communication between a client and his customer is made. In terms of this project it used only in terms of Twilio, but could be potentially used to represent any other communication

channel like Telegram, Viber or even any other messaging service. Contains the following fields:

- 1) Channel Type Name – a field for identification of the channel by users;
- 2) Account Sid – a login from Twilio;
- 3) Auth Token – authorization token with the help of which the authorization to Twilio is held;
- 4) Callback Url – url where Twilio sends all the info regarding the change of the message status;
- 5) Max Message Length – Twilio doesn't support message length more than 1600 symbols per message, so it is needed to prevent users entering messages longer than allowed.

Channel – an object that represents one Twilio number of the user. Contains the following fields:

- 1) Name – a field to identify a channel by the user;
- 2) Phone Number – a phone number for a channel;
- 3) Queue Name – a field to setup a Queue on which all the messages will be assigned.

External Channel – an object to represent the field on the record where the user will save a phone number of the customer to reference to it further. Contains the following fields:

- 1) Name – a field to identify an external channel by the user;
- 2) Object API Name – an object name on which the phone number is saved. Can be any object, which contains at least one String field;
- 3) Field API Name – a field of the object where the phone number is saved;
- 4) Channel Type – a channel type for which current external channel is set up.

Message Template – an object for creating any message templates for quick sending of the messages. Contains the following fields:

- 1) Name – a field to identify a message template by the user;
- 2) sObject API Name – an object for which the message template is created;
- 3) Message Text contains the message itself which will be sent by the users of the Messaging App.

Message Template Translation is needed to setup different translations for any particular message, created in the message template. Contains only two fields:

- 1) Language – a field to set up a language in which the message is;
- 2) Text Body – an actual translation according to the language.

Conversation – an object to connect the customers with the record where the number is saved and associate it with some channel number. Contains the following fields:

- 1) Name – name of the conversation, which is generated automatically;
- 2) Status can be open or closed to understand if the conversation with the current customer is active or not;
- 3) Channel here is used to associate all the messages with the specified channel and its phone number;
- 4) Message Recipient – phone number of the customer;
- 5) Record Id – Id of the record, where the customer is saved as an entity;
- 6) External Channel is needed here to understand in which of the fields of the record the phone number is saved.

5.2. Implementation of the Messaging Administration Page

The whole Messaging App first of all requires a fundamental configuration and setup of the major setting such as instantiation of the connection with the Twilio and config inside the Salesforce, which includes a majority of the different kinds of objects, such as Channel, External channel and all other main objects. As it is hard to understand the whole end-to-end process of the configuration, it was decided to develop one page where all the settings could be found in one place. The page was placed in the separate tab of the Messaging App and can be found on the figure 5.3.

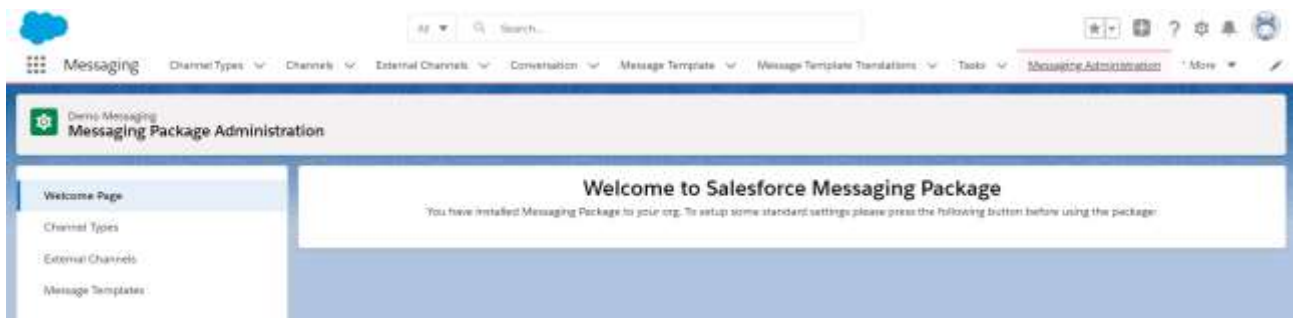


Fig. 5.3. Messaging Package Administration

Once opened, the page shows a greetings message and propose to install all the missing components, such as PushTopics and some default External Channels. The page itself contains four main tabs, which can be seen on the figure 5.4.

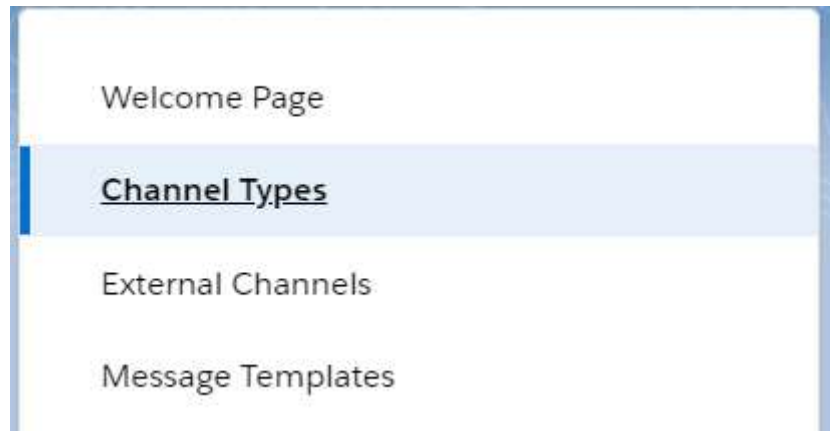


Fig. 5.4. Navigation panel of the administration page

The second option in the navigation bar is Channel Types, where users are able to setup the Channel Types and channels for them. It also gives an option to activate or deactivate a channel for the current Channel Type. The whole page is shown on the figure 5.5. Besides, it gives an ability to create or edit a new channel, as shown on the figure 5.6.

The third option on the administration tab gives ability of administration of the External Channels, that is here it is possible to create, delete or clone some existing External Channels. There are also two default external channels, which cannot be deleted as they are used in a separate part of the Messaging App functionality. The External channels setup abilities can be seen on the figures 5.7 and 5.8.

The last tab contains the configuration for the Message Templates and their translation. It is possible here to create the first template, which is always in English language, and only after that it is possible to add some language translations for it. The creation of the template can be found on the figure 5.9, while the creation of the translation can be seen on the figure 5.10.

Message Templates are used for sending messages to the users by template, so that once the message is created, it can be send multiple times to anyone in the system and the message can be in any available language.

Channel Type: Twilio Back

Channel Type Name Twilio	Account Sid AC3c7fdd88b5a00b592fce33b96302c0fe
Auth Token 8802566db5c3f31e77068f7e524fe4d0	Callback Url https://mustard-whale-4247.twilio.com/salesforce
Max Message Length 1,600	

Active Channels New Channel

Name	Phone number	Queue Name	
Twilio	+17754178798	Inbound_Task	<input type="button" value="Edit"/> <input type="button" value="Deactivate"/>

Deactivated Channels

Name	Phone number	Queue Name	
Twilio_Test	+15005550006	Inbound_Task	<input type="button" value="Edit"/> <input type="button" value="Activate"/>

Fig. 5.5. Channel Types setup in the administration tab

Edit Channel

* Name

* Phone number

* Queue Name

Fig. 5.6. Editing channel on the administration tab

External Channels				New External Channel	
Name	Object API Name	Field API Name	Channel Type	Clone	Delete
AccountFax	Account	Fax	Twilio	Clone	Delete
AccountPhone	Account	Phone	Twilio	Clone	Delete
Standard_Contact_Phone	Contact	Phone	Twilio	Clone	Delete
Standard_Lead_Phone	Lead	Phone	Twilio	Clone	Delete

Fig. 5.7. List of created via administration tab external channels.

New External Channel

* External Channel Name

Channel Type

* Object API Name

Get objects from packages

* Field API Name

- Assistant's Name (AssistantName)
- Asst. Phone (AssistantPhone)
- Business Fax (Fax)

Fig. 5.8. Creation of new external channel

Message Template: FeedbackAccount

* Message Template Name
FeedbackAccount

* sObject Api Name
Account (Account)

Get objects from packages

Cancel Save

Fig. 5.9. Form for creation of the message template

New Message Template Translation

* Language
English

* Text Body
Hello, could you please give us feedback about the quality of our service?

Cancel Save

Fig. 5.10. Form for creation of the translation of the message template

The Administration tab also can process different kind of mistakes during the creation of the needed objects. For example, which can be seen on the figure 5.11, the external channel can be created with the unique name, which begins with a letter and cannot contain spaces. The form can handle these mistakes and display them in a handy toast.

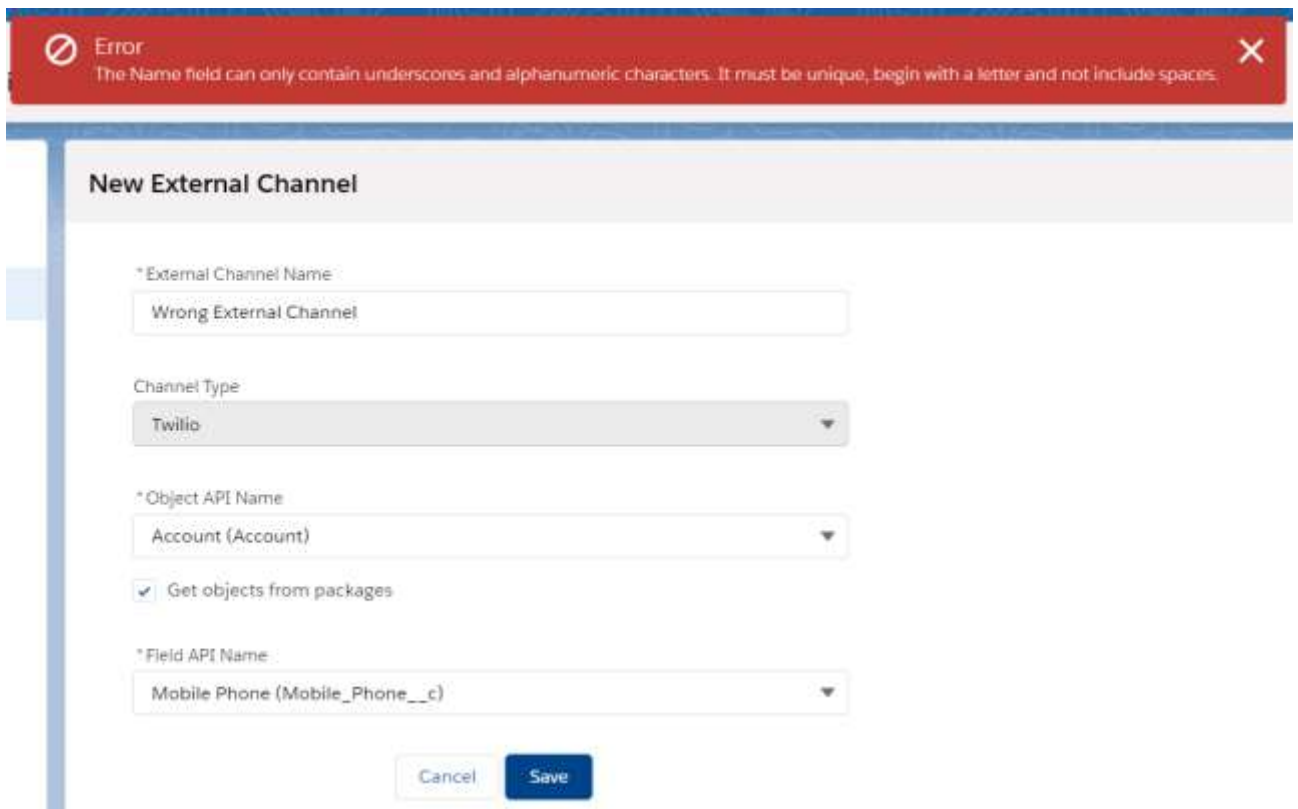


Fig. 5.11. An example of the toast message for displaying error messages

5.3. Concept of the Inbound Messages Tab

In the case, when a customer sends a message to the user, it is needed to save this message and display it somewhere. For this purpose, the Inbound Messages Tab was developed. It gives an ability to read the message from a customer and even to communicate with the customer via the chat in the tab. The tab can be seen on the figure 5.12.

The component for the tab was developed from scratch according to the Salesforce styling guidelines with the use of default styling patterns, and the styling itself is similar to the general concept around the Salesforce so the component looks and feels like it is a standard one.

The styles are configured similar to the Tasks Kanban to keep the look of the Inbounds tab understandable and consistent for the ordinary users.

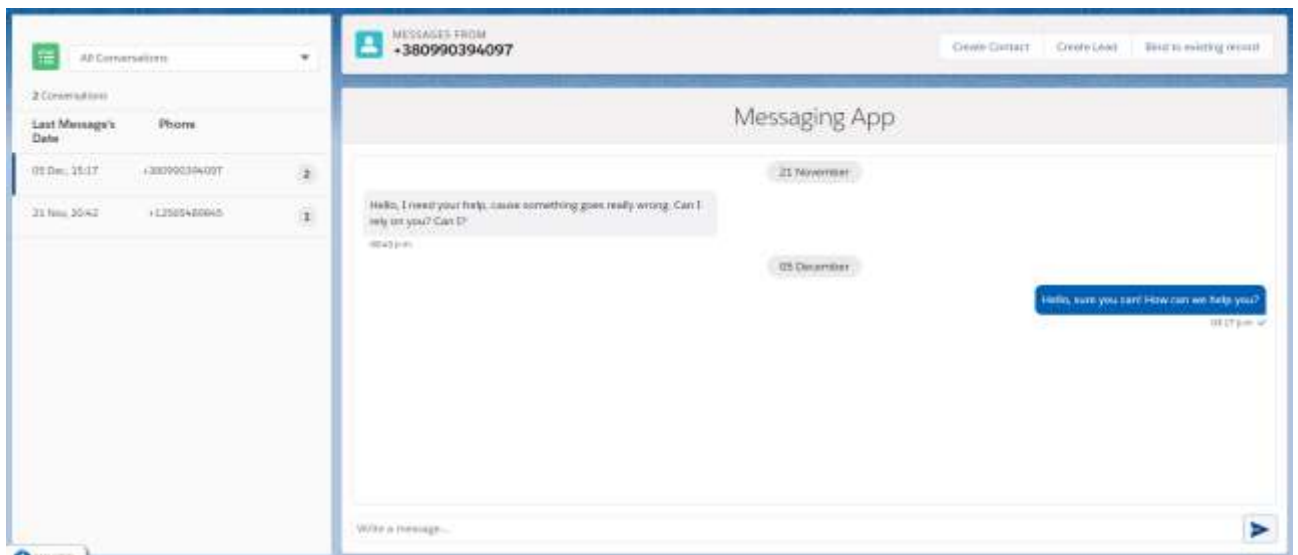


Fig. 5.12. Inbound Messages Tab

On the left part it is possible to see all the conversations from the customers and their numbers, besides the number of messages in the conversation is also shown. The conversations can be filtered by a particular channel, which can be chosen with the help of the picklist on the upper part, or they can be shown without any filtering, that is to show all the conversations, which can be seen on the figures 5.13 and 5.14.

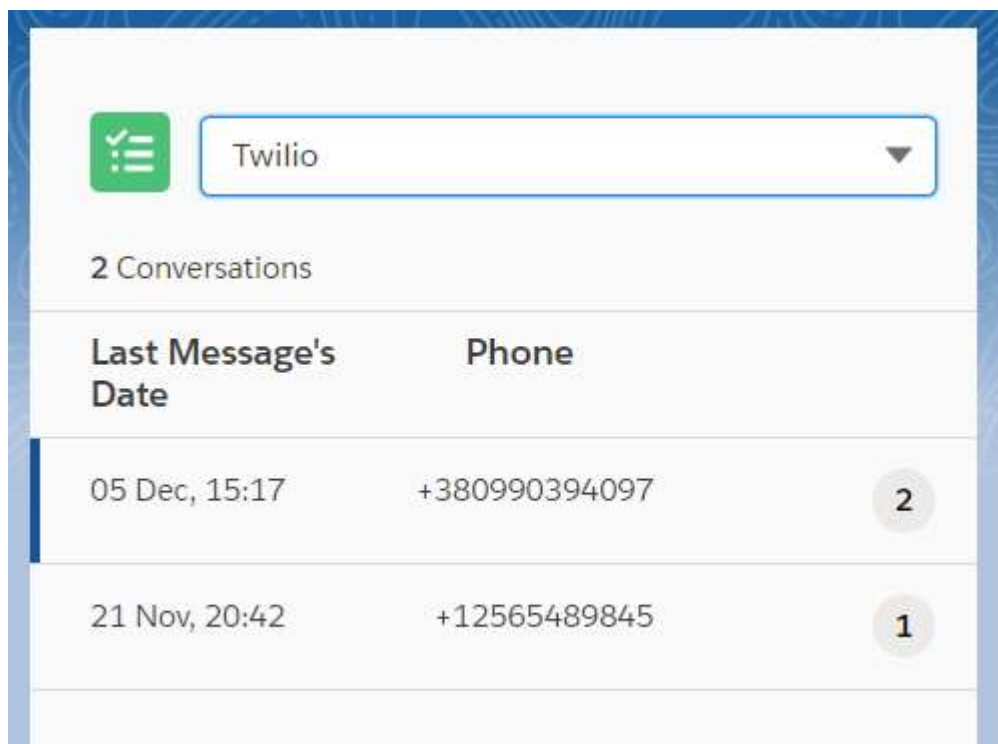


Fig. 5.13. Conversations which are related to the Twilio channel

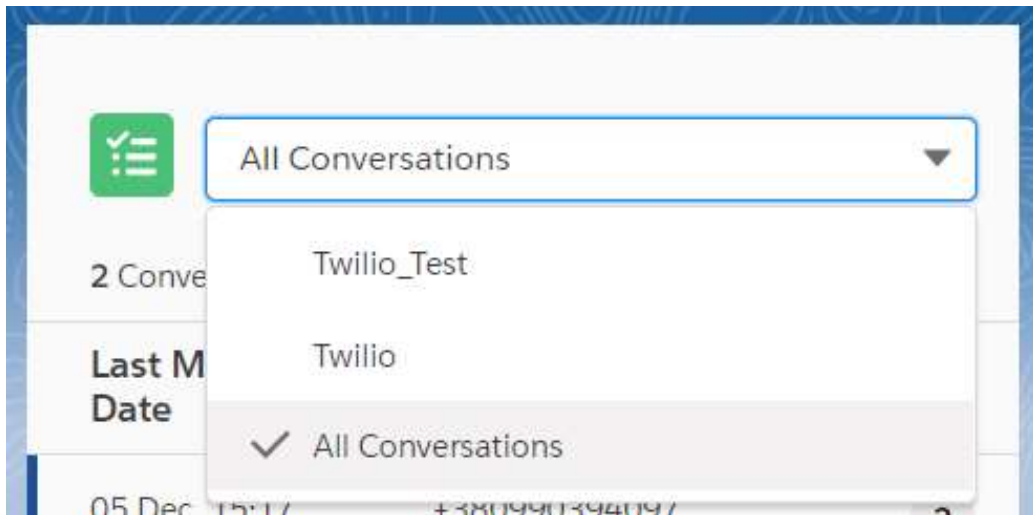


Fig. 5.14. Picklist for choosing a channel for conversations filtering

As soon as the user choose the conversation, he is shown a chat where he can see the number, from which the message has come and a window with the chat, as on the figure 5.15. The window contains all the messages from the number and gives an ability to respond by entering a message the box at the bottom and hitting a Send button. The chat also helps to understand the date and the time of sending of every message, which was sent to the customer or received from him.

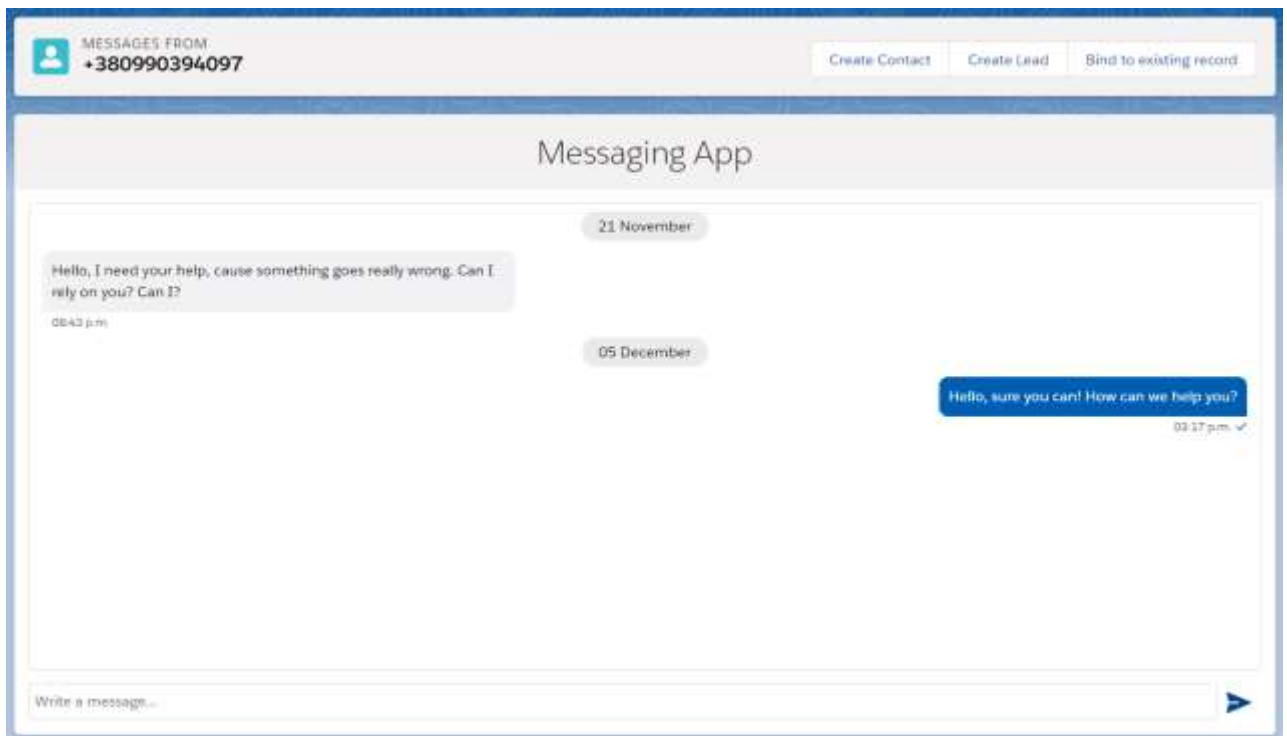


Fig. 5.15. Window with the chat for the chosen conversation

The next option for the window with selected conversation is to save the conversation on some record. Here it is possible to create a new Contact or Lead, if the record doesn't exist yet, as depicted on the figures 5.16 and 5.17. When creating a new record, the phone number is recorded into the standard phone field, but any other field can be chosen.

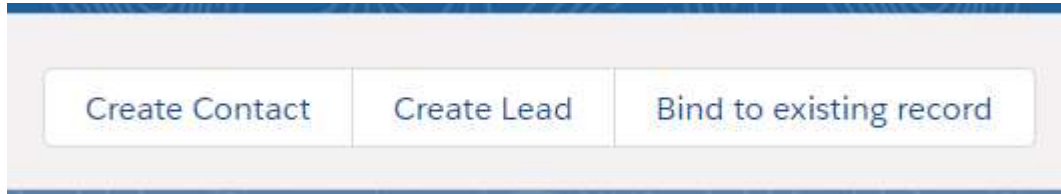


Fig. 5.16. Buttons for moving conversation to a record

A screenshot of a 'New Contact' form. The form is titled 'New Contact' and has a section header 'Contact Information'. It contains several input fields: 'Contact Owner' (Max Burba), 'Phone' (+380990394097), '* Name' (Salutation: Mr., First Name: empty, Last Name: Someone who wants to rely on me), 'Account Name' (Search Accounts...), 'Title' (empty), 'Department' (empty), 'Home Phone' (empty), 'Mobile' (empty), 'Other Phone' (empty), and 'Fax' (empty). At the bottom right, there are three buttons: 'Cancel', 'Save & New', and 'Save'. The 'Save' button is highlighted in blue.

Fig. 5.17. Creation of a contact to bind the conversation

The next option is to bind the conversation to some existing record by clicking the last button from the group. It opens a new modal window, where the system suggests to search for the record of a particular type, which can be Account, Contact or Lead, search for

the existing record by its name, select a field where the phone number will be saved and bind to record, as on the figures 5.18 and 5.19.



Fig. 5.18. Selection of the object to wire an existing record

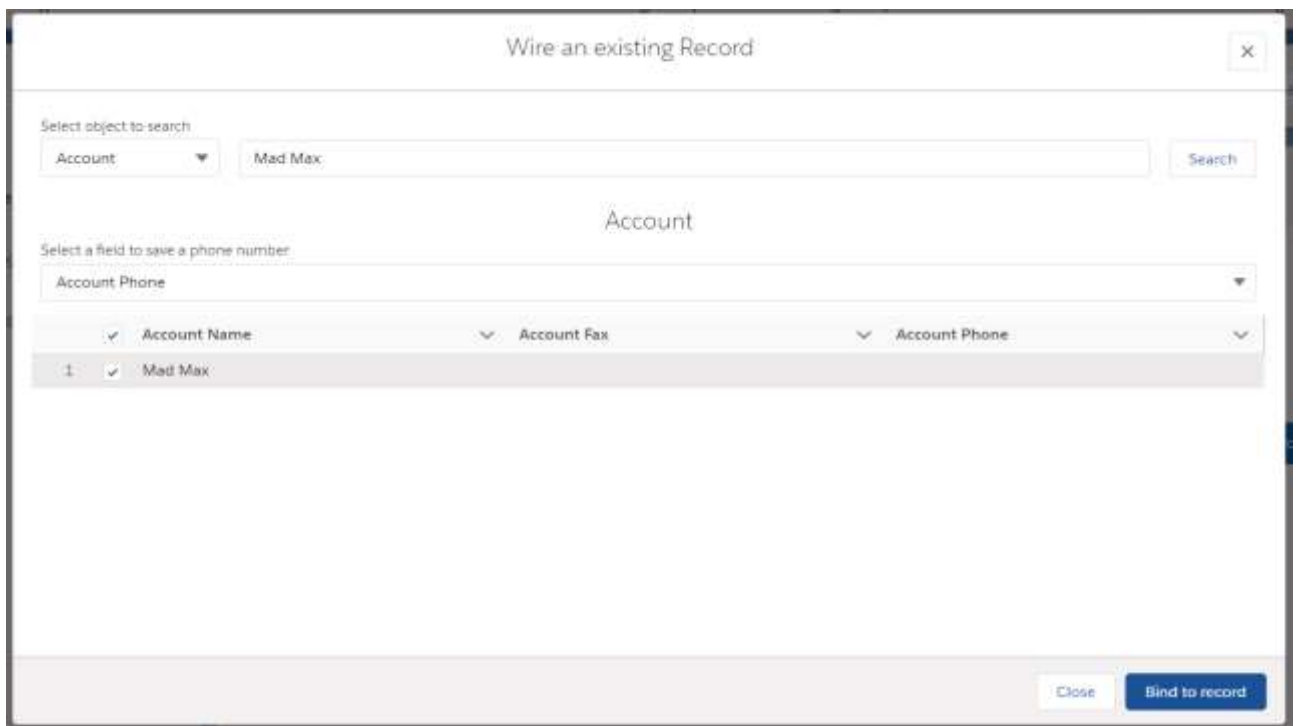


Fig. 5.19. Binding a conversation to found existing record

5.4. The Usage of the Chat with the Record

After the user create a record for the conversation or wires it with some existing record, the conversation is displayed as a chat on the record, but first of all it is needed to add the component with the chat to the page layout. To do this, it is needed to enter a special

mode of the page for the current record, where the component can be edited and updated with the needed component, as on the figure 5.20.

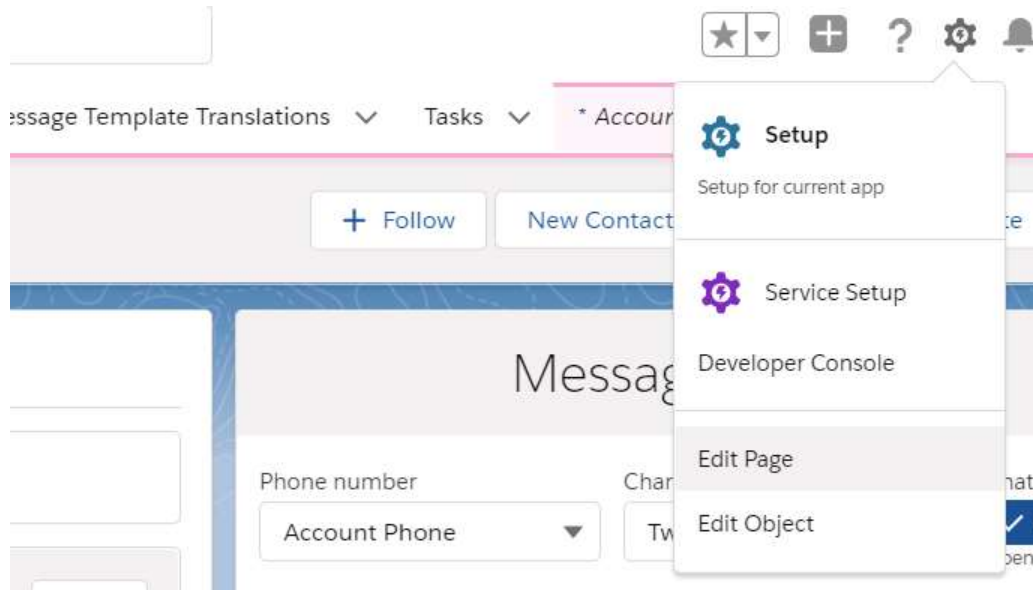


Fig. 5.20. Dropdown list with the option to edit page

Once the view for editing page is opened, it is needed to pick the chat component from the Components section under the Custom heading and place it at any convenient place. The component has an adaptive design and will fill as much space as needed, which makes it look good under any circumstances and can be of any width. After choosing a place for the component, it is needed to click the Save button at the upper right corner of the page. It might also be needed to click the Activation button before saving to make this page default for particular profile or any other option if the user had never changed the page before.

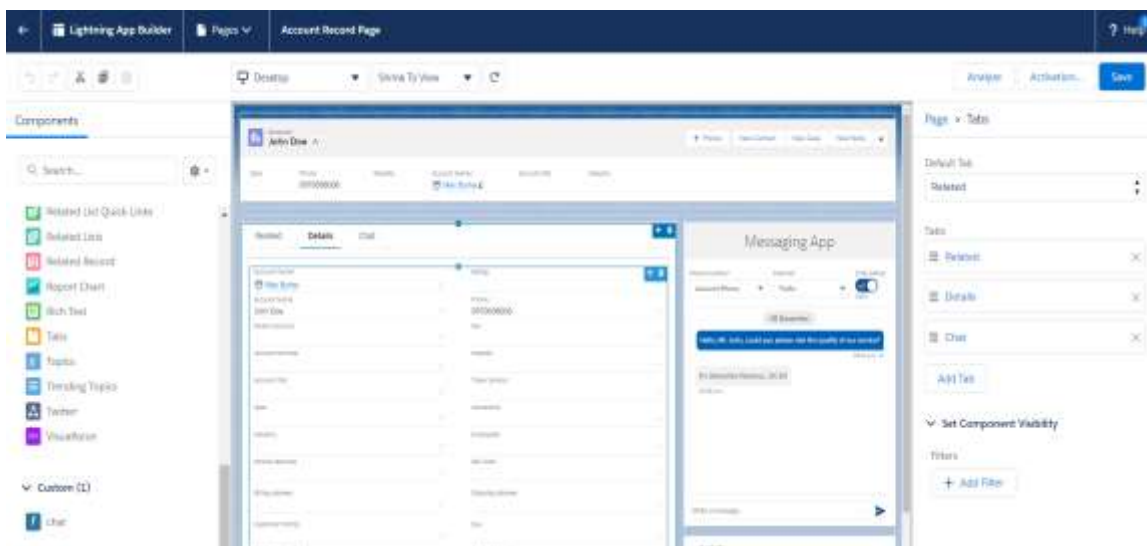


Fig. 5.21. Edit mode of the page

Once the chat component is placed on the page, it can be used to continue the conversation with the customer. The conversation history is migrated from the Inbound Messages tab to the record and can be continued here. All the outbound messages to the customer and all the inbound messages from the customer will be further displayed in this chat, which is displayed on the figure 5.22.

The chat is capable to display conversation on the record for different customer's phone numbers, the Phone number picklist exists exactly for this purpose and allows to choose the phone number to be displayed for the customer. The picklist is the straight representation of an External Channel object, which should be created for this record from the administration tab. The chat also gives the ability to close the conversation. In this case, if the customer previously had the conversation with the user and sends a new message, it will be displayed on the Inbound Messages tab and the new conversation will be created.

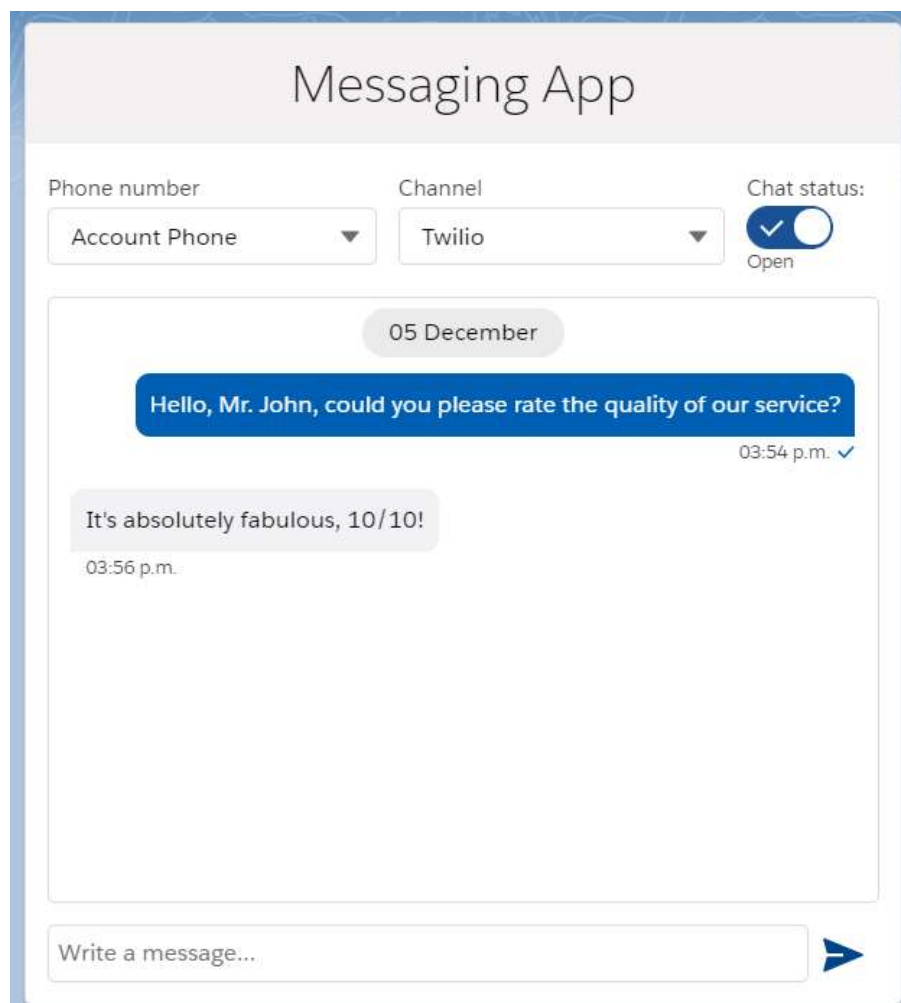


Fig. 5.22. The chat with the conversation

5.5. Setup of the Twilio

After creating a Twilio account it is needed to navigate to All Products & Services, Phone Numbers, Buy a Number and buy any number with SMS and MMS capabilities. The next step will be creating of the Connected App withing Salesforce to setup a connection between a Salesforce and Twilio:

1) In Lightning Experience, use the App Manager to create Connected Apps. From Setup, enter “App” in the Quick Find box, then select App Manager. Click New Connected App;

2) Enter the name of your application;

3) Enter the contact email information, as well as any other information appropriate for the application;

4) Select Enable OAuth Settings;

5) Enter “https://www.twilio.com” as the callback url;

6) Select Access and manage your data (API) to Selected OAuth Scopes.

An example of the filling the fields during the creation of the connected app can be found on the figure 5.23.

The screenshot shows the 'Basic Information' and 'API (Enable OAuth Settings)' sections of the Salesforce Connected App setup page. The 'Basic Information' section includes fields for Connected App Name (Twilio), API Name (Twilio), Contact Email (user@email.com), Contact Phone, Logo Image URL, Icon URL, Info URL, and Description. The 'API (Enable OAuth Settings)' section includes checkboxes for 'Enable OAuth Settings' (checked), 'Enable for Device Flow', and 'Use digital signatures'. The 'Callback URL' field is filled with 'https://www.twilio.com'. The 'Selected OAuth Scopes' section shows a list of 'Available OAuth Scopes' and a 'Selected OAuth Scopes' box containing 'Access and manage your data (api)'. The 'Require Secret for Web Server Flow' checkbox is also checked.

Fig. 5.23. Creation of the Connected App for the Twilio

Needed Consumer key and Consumer key will be available after Connected App creation and can be seen on the figure 5.24.

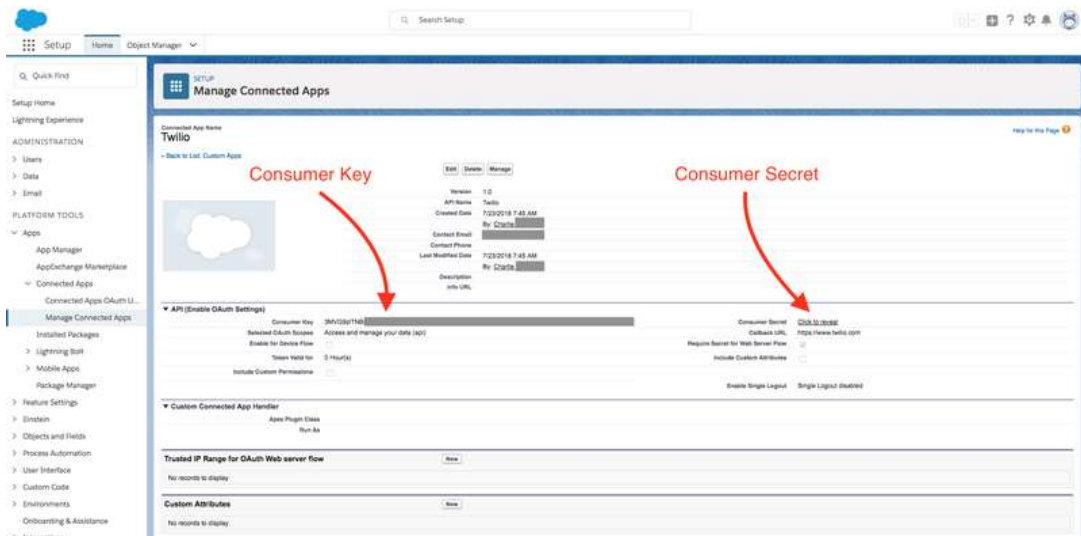


Fig. 5.24. Keys of the created connected app

The next step would be creation of the Twilio function. In a new tab, visit the functions page in the Twilio Console, click on the red "+" button to add a new function. In the modal that opens, click on "Connect to Salesforce".

Next, it's possible to see a list of configuration variables for the function. Fill them out with the values specified in the following way:

- 1) SF_IS_SANDBOX – false;
- 2) SF_CONSUMER_KEY - Consumer key from Connected App;
- 3) SF_CONSUMER_SECRET - Consumer Secret from Connected App;
- 4) SF_USERNAME - The login from Salesforce;
- 5) SF_PASSWORD - the password from Salesforce;
- 6) SF_TOKEN - A Salesforce Security Token.

Once the user has added his configuration variables, click the red "Create" button. He will be directed to a Function page that looks like the one on the figure 5.25.

When the function is created, it is needed to connect it with the Messaging Service. First of all, it is necessary to copy the function path containing the unique URL, which is presented on the figure 5.26.

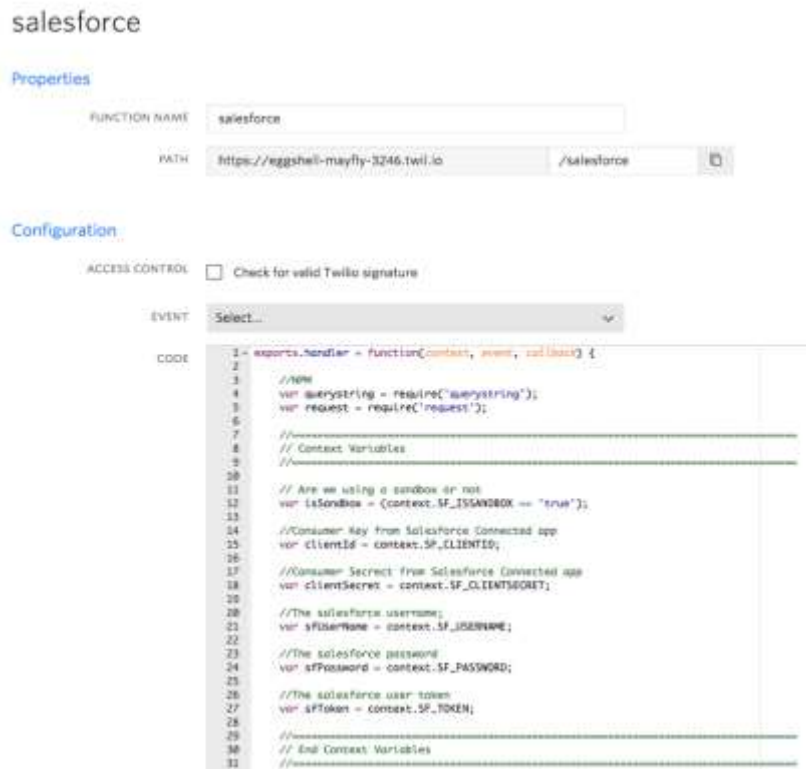


Fig. 5.25. The created Twilio function

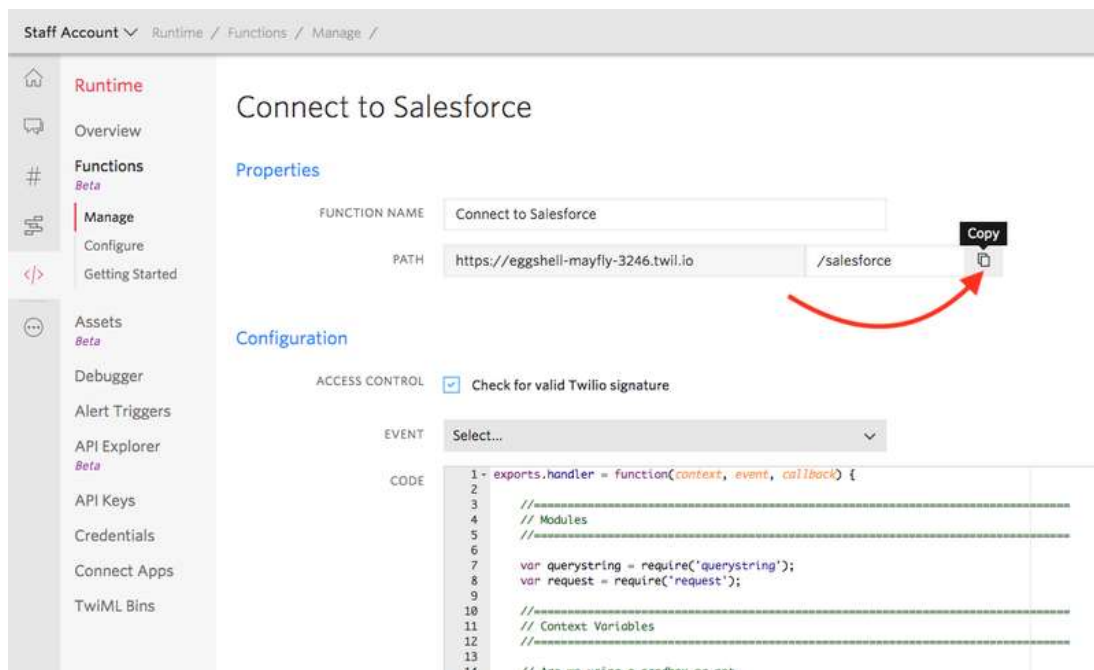


Fig. 5.26. Making a copy of the function's unique path

After making a copy of the link it is needed to navigate to the Twilio Services and click Create new Messaging Service button. Name can be any that the user likes, Use Case must be Notifications, 2-Way.

In Inbound Settings, it is needed to choose SEND AN INCOMING_MESSAGE WEBHOOK and paste a copied link from the created function to REQUEST URL and STATUS CALLBACK URL. The result should be similar to the content of the figure 5.27.

The screenshot shows the Twilio Inbound Settings configuration page. Under the 'Properties' section, the 'FRIENDLY NAME' is 'New Messaging Service', the 'SERVICE SID' is 'MCa6492e8daa4E380861966007811852c54', and the 'USE CASE' is 'Notifications, 2-Way'. In the 'Inbound Settings' section, the option 'SEND AN INCOMING_MESSAGE WEBHOOK' is selected. Below this, the 'REQUEST URL' is 'https://boysenberry-millipede-2792.twilio.com/salesforce' and the 'FALLBACK URL' is empty. In the 'Outbound Settings' section, the 'STATUS CALLBACK URL' is 'https://boysenberry-millipede-2792.twilio.com/salesforce'.

Fig. 5.27. Process of creation of the messaging service

The last step of the configuration would be the navigation to Remote Site Settings and clicking on New Remote Site button, which is shown on the figure 5.28.

The screenshot shows the Salesforce 'Remote Site Settings' page. The left sidebar shows 'Setup' and 'Security' with 'Remote Site Settings' selected. The main content area has a search bar with 'Remote S' and a 'New Remote Site' button highlighted in yellow. Below the search bar, there is a section titled 'All Remote Sites' with a description: 'Below is the list of Web addresses that your organization can invoke from salesforce.com. To add another Web address, click New Remote Site.' There is a 'View: All Remote Sites' dropdown and a 'Create New View' button.

Fig. 5.28. Adding Twilio URL to the Remote Sites in Salesforce

Set up Remote Site URL to https://api.twilio.com/ and Remote Site Name to whatever the user likes. The end result should be similar to what is shown on the figure 5.29.

Fig.5.29. Creating of a new Remote Site for Twilio

Conclusions on the Fifth Part

The whole Messaging App first of all requires a fundamental configuration and setup of the major setting such as instantiation of the connection with the Twilio and config inside the Salesforce, which includes a majority of the different kinds of objects, such as Channel, External channel and all other main objects. As it is hard to understand the whole end-to-end process of the configuration, it was decided to develop one page where all the settings could be found in one place.

Generally there was developed 2 custom tabs. The first one is the Messaging Administration Tab, which is needed to show all the needed settings in one place to give users the ability to setup everything as quick as possible instead of providing them with the confusing guide as it is hard to explain how to configure the system if its part are situated in different parts of the platform. The second tab is needed to show all the inbound messages from unknown numbers and to give ability to communicate with customers even when the number is not wired to the specific record.

Twilio needs to be properly configured before it is used in Salesforce in terms of Messaging Application. It is needed to create a new account and buy a number inside of the Twilio which will be used as a number to which customers are going to send all their messages.

The Messaging Application can currently be configured to be used with Twilio messaging service only, but the main idea behind this system is that it can be expanded in

the future and be used with other main communication channels. For example, it can be still SMS, but with the usage of any other messaging service which providing a decent API, or it can be any more or less popular messenger like Telegram, Viber, WhatsApp or any other messenger, providing an API encouraging integration.

In the case, when a customer sends a message to the user, it is needed to save this message and display it somewhere. For this purpose, the Inbound Messages Tab was developed. It gives an ability to read the message from a customer and even to communicate with the customer via the chat in the tab.

The whole Messaging App first of all requires a fundamental configuration and setup of the major setting such as instantiation of the connection with the Twilio and config inside the Salesforce, which includes a majority of the different kinds of objects, such as Channel, External channel and all other main objects. As it is hard to understand the whole end-to-end process of the configuration, it was decided to develop one page where all the settings could be found in one place. The page was placed in the separate tab of the Messaging App

CONCLUSIONS

CRM is an interaction model based on the theory that the center of the entire business philosophy is the client, and the main areas of the company's activity are measures to ensure effective marketing, sales and customer service. Supporting these business goals includes collecting, storing and analyzing information about customers, suppliers, partners, as well as about the internal processes of the company. Functions to support these business goals include sales, marketing, customer support.

Online services for sending sms-mailings, e-mail-mailings, as well as mailings on popular messengers significantly simplify the work of informing clients and customers about important company events. Internet platforms for sending SMS messages have become an indispensable tool for employees responsible for marketing and communication with customers, especially in terms of CRM platforms.

Twilio is a platform for creating sms messages, calls and other communications. With its help, it is possible to set up automated mailings and receive response messages, make voice calls and video calls, receive verification codes for authorization on various portals, create call centers and chat bots.

Twilio needs to be properly configured before it is used in Salesforce in terms of Messaging Application. It is needed to create a new account and buy a number inside of the Twilio which will be used as a number to which customers are going to send all their messages.

In the Messaging App project the Twilio API is actively used for creation of the messages, sending them to users and following their current statuses. REST from the side of Twilio to create a webhook which gives an ability to update the statuses of any outgoing messages.

Lightning Web Components are chosen as a main front-end framework instead of Visualforce and Aura components as it is more convenient, flexible and easy to use due to the lower amount of code needed to be written in comparison with other Salesforce

frameworks. The main disadvantage is that LWC is a relatively new framework, but still it seems to be stable enough to complete the tasks in terms of the project.

The Messaging Application can currently be configured to be used with Twilio messaging service only, but the main idea behind this system is that it can be expanded in the future and be used with other main communication channels. For example, it can be still SMS, but with the usage of any other messaging service which providing a decent API, or it can be any more or less popular messenger like Telegram, Viber, WhatsApp or any other messenger, providing an API encouraging integration.

As the result of graduation project, there was gained understanding of current state of the concept of communication channels in CRM, its tasks and problems; there was also gained information about concepts, necessary for implementation of the Salesforce app, which was also successfully developed. The project covers a lot of specific themes and can be a good starting point for further and more deep investigation of the field of CRM, importance of its communication channels and areas related to it.

REFERENCE LIST

1. Understanding the notion of CRM [Internet Resources] / Website: Salesforce; Access mode: <https://www.salesforce.com/crm/what-is-crm/>, free.
2. Intro to Point & Click App Development [Internet Resources] / Website: Udacity; Access mode: <https://www.udacity.com/course/intro-to-point-click-app-development--ud162>, free.
3. Sending Messages with Twilio API [Internet Resources] / Website: Twilio; Access mode: <https://www.twilio.com/docs/sms/send-messages>, free.
4. Definition of Paas [Internet Resources] / Website: Salesforce; Access mode: <https://www.salesforce.com/paas/overview/>, free.
5. The 10 Best Salesforce CRM Alternatives [Internet Resources] / Website: The Blueprint; Access mode: <https://www.fool.com/the-blueprint/salesforce-alternatives/>, free.
6. Types of chat channels [Internet Resources] / Website: Tolkative; Access mode: <https://gettalkative.com/info/12-types-of-customer-contact-channels-and-their-key-benefits>, free.
7. What Communication Channels Do Customers Prefer [Internet Resources] / Website: CustomerThink; Access mode: <https://customerthink.com/what-communication-channels-do-customers-prefer-it-depends/>.
8. What is Omni-Channel for Salesforce [Internet Resources] / Website: Arkus; Access mode: <https://www.arkusinc.com/archive/2016/what-is-omni-channel-for-salesforce>, free.
9. Notion of API [Internet Resources] / Website: Mulesoft; Access mode: <https://www.mulesoft.com/resources/api/what-is-an-api>, free.
10. Client-server architecture [Internet Resources] / Website: Britannica; Access mode: <https://www.britannica.com/technology/client-server-architecture>, free.
11. Understanding notion of HTTP [Internet Resources] / Website: Cloudflare; Access mode: <https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/>, free.

12. Understanding REST [Internet Resources] / Website: Codecademy; Access mode: <https://www.codecademy.com/articles/what-is-rest>, free.

13. Lightning Web Components [Internet Resources] / Website: Tolkative; Access mode: <https://springml.com/blog/aura-vs-lightning-web-components/>, free.

14. Aura vs Lightning Web Components [Internet Resources] / Website: SpringML; Access mode: <https://springml.com/blog/aura-vs-lightning-web-components/>, free.