TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Julian Dinovski 193627IAAB

Optimizing Support Experience Using Zendesk Platform

Bachelor's thesis

Supervisor:	Tauseef Ahmed
	PhD in Electronics and Telecommunication

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Julian Dinovski 193627IAAB

Kliendikogemuse optimeerimine kasutades Zendesk platvormi

Bakalaureusetöö

Juhendaja: Tauseef Ahmed

Doktorikraad
elektroonikas ja
telekommunikatsioonis

Tallinn 2022

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Julian Dinovski

15.05.2022

Abstract

When an incident or problem with customer's order delivery takes place, it is mandatory to notify client as fast as possible and give possibilities to deliver goods later. If done manually, this work takes a lot of time and resources.

The aim of this bachelor thesis is to analyze, develop and implement the best possible order changing solution, according to the company's requirements and needs.

The proposed solution presented in this thesis is focused for the company that requires a better automated order changing system for customers in case of logistic incidents. Web forms for new business client's registration and customer's feedback are implemented in order to improve service quality. It is required by the company to develop solution utilizing Zendesk platform, which is currently used for customer requests and issues resolving.

This thesis is written in English and is 63 pages long, including 7 chapters, 27 figures and 7 tables.

Annotatsioon
Kliendikogemuse optimeerimine kasutades Zendesk platvormi

Kliendi tellimuse kohaletoimetamisega seotud probleemide ilmnemisel on kohustus klienti probleemist kiiresti teavitada ja anda võimalus kaup kohale toimetada. Kui teavitamise ja tellimuse muutmise tööd teha käsitsi, töö võtab palju aega ja ressursse.

Käesoleva bakalaureusetöö eesmärgiks on analüüsida ja välja töötada tellimuste muutmise lahendust, mis võimaldab teavitada klienti tarneprobleemidest ning anda võimalust valida nendele sobiv tellimuse tarneaeg. Lahendus on välja töötatud vastavalt ettevõtte nõudmistele ja vajadustele.

Käesolevas lõputöös arenenud lahendus on suunatud ettevõttele, mis eeldab logistiliste probleemide korral pakkuda klientidele paremini automatiseeritud tellimuste muutmise süsteemi. Teenuse kvaliteedi tõstmiseks rakendatakse veebivorme uute äriklientide registreerimiseks ja klientide tagasiside saamiseks. Ettevõte soov on lahendust välja töötada kasutades Zendesk platvormi. Zendesk on praegu kasutusel klientide probleemide ja pöördumiste lahendamiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 63 leheküljel, 7 peatükki, 27 joonist, 7 tabelit.

List of abbreviations and terms

Agent	Support staff who interacts with customers to resolve issues.
Automation	Automation executes when a time event occurs after a ticket property is set or updated.
CSS	Cascading Style Sheets. Language for styling web documents (XML/HTML).
CX	Customer experience.
HTML	HyperText Markup Language. Markup language designed for web pages.
Placeholder	The reference to ticket and user data that can be included in the subject and text fields of email notifications.
Requester	Person who is asking for support through a ticket.
Status	Each ticket is assigned a status. There are five status values: New, Open, Pending, On-hold, Solved, and Closed.
Trigger	Trigger runs immediately after a ticket is created or updated and automatically performs actions if specified conditions are met.
View	Defines a collection of tickets based on a set of criteria.

Table of contents

List of figures

List of tables

Introduction

Nowadays good customer experience is one of the most important aims and metrics for companies who are working closely with their customers and want to provide the highest level of services. Great customer experience ensures that the company and the customer will have a long and healthy relationship and customer will spread positive feedback. It is important that the customer is satisfied with the product itself and also with the support and help that he receives.

Companies sometimes may spend unnecessarily significant amount of time working with customers and doing repetitive tasks - trying to solve their problems, notifying of logistic problems or of any changes in their order. Support specialists must do it fast with high level of professionalism to make sure that the customer is satisfied.

Many actions in problem resolving and customer notification chain can be simplified and automated in order to save time for both customers and customer support specialists.

Company whose problem this work solves requires a better solution for customers order changing system in case of logistic incidents and in situations where orders are not delivered in time. In its current state, notifying clients of order changes and offering possible solutions consumes extra time and contains a lot of repetitive manual work. In addition, web forms for customer's feedback and new business client's registration must be implemented for the reason that the feedback and new client's information gathering is not optimized and makes grouping and analyzing extra hard.

The company uses Zendesk for gathering and resolving customer requests and problems. It is required by the company to develop the solution using Zendesk functionality and services.

1 Background

The company serves to a big number of customers and uses Zendesk platform in order to gather and resolve customers' requests and issues. This thesis proposes a solution to the order changing system, the feedback and new business client's registration problems using Zendesk environment capabilities.

It is important to mention that widespread business disruptions due to pandemic and the growth of disruptive business models have shifted boardroom and organizational priorities. Organizations expect to spend more on agility and business continuity, and they no longer seek to invest more in legacy technologies and systems that do not support those two areas [1]. Company does not want to invest into legacy solution, hire more agents and provide additional investment in facilities and employee training [2]. The aim for the company is more agile systems that are controlled, automated and require fewer human resources.

1.1 Problem definition

This part describes problems that this work solves. The problems are:

- Orders delivery time changing and cancelling in case of incidents, logistic problems. Support specialists must do calls to the customers and ask for suitable time or cancellation. This process takes a lot of time for specialists.

- Customer's feedback and new business client's registration receiving and processing. No specific registration and feedback forms with pre-defined structure exist, which leads to hard data aggregation and analysis.

It is required by the company to simplify and automate the processes to resolve problems and save both agents' and customers' time.

Most of the times order changes are accompanied by many different documents and spreadsheets in order to track customers' order numbers, decisions, chosen and suitable delivery time. Customer support specialists must often wait for call-backs or call again

12

themselves. This entire order changing process requires additional time to resolve problems in case of logistic incidents mainly due to the amount of phone calls. It is required by the company to reduce the workload on the support specialists. Moreover, enhancing the customer experience during the service delivery process requires additional investment in facilities and employee training [2]. Specialists need to do specific trainings associated with the customer support activities and behavior. This leads to additional expenses that can be saved, if customers can simply pick suitable delivery time themselves.

Problem with feedbacks and new business client's registration is that there is no pre-defined form with strict and proper information that company wants to gather. Customers are simply leaving their feedbacks via emails in a free form, which makes it hard to group and analyze. There can be appraisal, complaint, suggestion feedback in the same mailbox. This leads to additional time spending on data processing. The problem with the new business client registration is similar and a sufficient number of calls is required before company clients understand which information must be provided.

The additional challenge for firm is to ensure a reasonable profit while providing service delivery enhancements that justify higher prices so that their customers still perceive value [2]. Compared to the previous manual approach, the proposed solution must provide a better method of sending notifications and changing order for both customers and agents in terms of time usage and convenience.

For Company additional reasons for the continued growth in self-service include [2]:

- the customer's perception that it is faster.

- the customer's desire to be more in control.

With self-services functionality customers must be able to resolve order problems themselves after receiving notification. There is no need for them of taking additional contact with the company and the support agents do not have to contact the customer. Customers feel more freedom and can choose suitable delivery time according to their requirements.

1.2 Alternatives analysis

This part describes the analysis of alternative services, solutions and chosen approach.

1.2.1 Order change notification system analysis

The orders changing and customer's notification problem can be solved using different solutions and different approaches.

The initial idea is to send mass notification, preferably both email and SMS to the customers that got affected by the logistic incident or by any other delivery problem related to their order. The SMS notification system is outside of this works scope. The affected customer must receive message containing information related to their order and explanations/reasons of goods not being delivered today. The customer must be able to pick new preferred delivery time, for example tomorrows delivery time, or to cancel order without the need to do phone calls. Finally, changed orders must be accessible in one specific environment, where logistic specialists and support agents can find them and further work with those affected orders.

Different Zendesk services and ideas are analyzed with the company and CX specialists to determine the best possible solution. Analyzed Zendesk services are Zendesk Support, Zendesk Gather and Zendesk Client lists add-on.

Zendesk Support can be used to solve the problem. This solution requires additional development of web environment from where mass notifications will be sent and from where order changing forms are defined and sent. Using received in email changing form customer can pick suitable delivery time. All order changes remain within the initial Zendesk ticket that represent the changed order. Tickets can be easily analyzed and grouped. The entire solution remains within the current Zendesk licensing plan – Zendesk Suite enterprise. The only additional expense that takes place for order changing system using Zendesk Support is the solution development and implementation, which is estimated at ca 3000 euros.

Zendesk Gather solution and functionality is analyzed and discussed. Gather is a community forum solution [3]. It is possible to create a mass message/post to a specific group of people notifying them about the delivery issues and providing/asking for a possible way on resolving the problem.

Zendesk Gather service functionality costs 20 euro every month per agent. However, it is mandatory to have at least 10 Zendesk Gather agents, even though in company only 5 agents will use it. For 10 agents this leads to 2400 annual payment. The solution implementation and configuration are estimated at ca 2000 euros. In total Zendesk Gather one-year service licenses and implementation is estimated at ca 4400 euros.

Zendesk Customer lists add-on is another decent solution to the proposed problem. Customer lists define a segment of users based on a set of criteria. It is possible to create a customer list by applying filters to determine which customers are included [4]. After clients list is defined, it is possible to send mass notification message about delivery issues.

Customer lists add-on requires current license update, in order to support Zendesk add-ons. The company has more than seven months of pre-paid Zendesk Suit Enterprise license plan and switching to another Zendesk licensing plan without losing investments is possible only after August 2022. Currently, the implementation of new license leads to seven months of paid license being cancelled. Seven pre-paid months of license for five support agents is ca 3000 euros. New Add-ons license is more expensive as well, but for the company it offers possibilities not only for described in thesis order changing project. There are many other different possibilities that will open with add-on's license, and because of that, annual increase in Zendesk add-on license cost is not counted. Zendesk Customers list system installation and configuration is estimated at ca 2000 euro. In total the Zendesk Customers list solution implementation with the license cancellation losses is estimated at ca 5000 euros.

In the company five agents are resolving delivery issues and working with the Zendesk solution. Per week, it is possible to have one to three delivery accidents/cases where orders cannot be delivered in time. This affects roughly 50 customers per week. For every affected customer, support specialist spends approximately ten minutes on calls, waiting for call-backs, filling spreadsheets, and gathering information. It can be more than 500 minutes or eight hours per week and more than 25-30 hours per month, which is ca 20% of one agent monthly working time. This time can be saved and described Zendesk solutions are analyzed based on the return of investments. Calculations are not disclosed, approximate ROI percentages are presented.

Using Zendesk Support solution yearly return of investments equals to approximately 60%. With Zendesk Gather solution yearly return of investments is approximately 10%. Zendesk Customer lists add-on yearly return of investments equals to approximately -5%. It will take more than a year for Zendesk Customer lists add-on solution to cover the expenses and to start making any profit for the company. Mainly due to the Zendesk license cancellation.

Zendesk Customer lists add-on license expenses are not justified. Compared to the Zendesk Gather, Custom developed Zendesk Support solution shows a better yearly return of investments for the company. Custom developed Zendesk Support solution is chosen. The solution remains within the currently used license plan and no additional expenses besides solution development and implementation take place.

1.2.2 Solution for feedback and new customer forms

Problem that this work solves is the absence of direct channel/point for submitting feedbacks and registering new business clients. Using correct forms, correct data can be gathered. Information can be grouped and analyzed in a much faster and better way. New business clients receive a better experience using pre-provided forms and filling all the necessary fields.

There are many solutions that offer different forms creation and redirection of those filled forms either to email boxes or their aggregation in a specific environment. The initial idea is that emails with filled form information must be sent to the Zendesk environment via specific email channel. All emails written to a specific configured email address appear within Zendesk as new tickets/requests using forwarding [5]. In Zendesk agents can work with gathered data, view statistics and feedbacks, provide answers to the customers. There is a Zendesk Guide product that offers such functionality with no additional layers of emails redirection, as it immediately creates Zendesk tickets from the web forms. Another way is to a use specific feedback/questionnaire software.

Different services are analyzed, and the best solution is chosen. Each of three solutions uses a different approach to solve the problem.

Examined solutions are:

16

- Zendesk Guide

- Simple web forms developed using HTML and PHP

- LimeSurvey

Zendesk Guide is a Zendesk solution with own additional licensing plan and comes as a part of specific Zendesk bundles [6]. Guide is used to create a Zendesk Help Center that contains the support portal provided to customer. Zendesk Guide offers possibilities to gather requests using help center web forms and transforms them straight into Zendesk Support tickets. Company's current Zendesk plan "Zendesk Suite Enterprise" supports many of the Zendesk products and capabilities, including Zendesk Guide. Created request registration web form can use pre-created ticket fields and there is no need in any additional text recognition automations to help with feedbacks grouping and routing. Customer fills and submits the form, the ticket in Zendesk Support is created. Form fields represent ticket fields. Full Zendesk tickets functionality can be used to work with these feedbacks, registrations, and requests.

Simple web page using HTML and PHP can be developed. This page contains correct fields and data that customer needs to provide. One page for customer's feedback and another page for new business client's registration. The customer submits form, and the backend script runs and sends email with the filled data as simple message to our Zendesk channel. In script, receiver is specified as Zendesk email channel and the sender is the customer's email defined in the form, email content is the filled form data. In Zendesk environment, such email is received as a new ticket with correct requester - customer.

The problem with developed web page solution is that the filled form is just a content of the email, Zendesk ticket fields are empty and there is only requester and receiver. A lot of additional recognition and acknowledging triggers must be created to group feedbacks and different new business clients. Statistics gathering is very problematic with such solution and requires further aggregation and separation logics development.

LimeSurvey is an open-source survey software [7]. This solution provides environment and functionality to create and distribute surveys/questionaries. Software can be used for free, however the number of possible participants per month with free version equals 25 and is not suitable. 'Basic' or 'Expert' plan with a sufficient number of participants can

be used for about 25 to 35 euros of monthly payment. With this software a completely different approach compared to Zendesk Guide solution is used. LimeSurvey is used to create different questionnaires, surveys and then send them to a bulk of emails/customers. The solution is to create feedback surveys and also registration forms, then it is possible to send them to a number of customers. For new business clients specific survey (registration form) with correct fields is created that can be later served using specific URL and redirections. All filled forms statistics can be gathered and analyzed within LimeSurvey.

This entire LimeSurvey process requires an additional specialist who is going to work with the LimeSurvey solution, create forms and send them to the customers. The problem with this solution is that it is inconvenient for specialists to navigate between Zendesk and LimeSurvey environments to work with customer feedbacks and requests. However, this process can be automated and further developed to automatically send survey to customer's whose order is fulfilled and then redirect survey results into Zendesk's correct ticket fields using API endpoints. Such Zendesk integration leads to additional expenses.

Zendesk Guide is chosen to solve the feedbacks and new business client's registration problem. Zendesk Guide solution remains within Zendesk environment, no additional expenses, licenses and integrations are needed.

2 Methodology

This chapter describes Zendesk Support tools, proposed solution design, and the methodology that are used to solve the problem.

2.1 Zendesk

Zendesk is a customer service solution that is designed to be simple and is used by many of the world's largest organizations to provide support to their customers [8]. Zendesk provides a big variety of different customer services with different functionality and purposes. Zendesk is used to create the solution described in this work. Two main Zendesk services that are utilized to solve the proposed problems are Zendesk Support and Zendesk Guide.

Zendesk Support is "The core of customer support". Zendesk Support is a ticketing system, a central location, where all customer requests and questions are gathered. Such solution provides abilities to efficiently solve problems and a flawless communication with the customers. It can be configured to provide a support across any possible channel – emails, help center, social medias, phone calls and more. Zendesk Support solution is highly configurable with a lot of automation and tickets aggregation possibilities that help to organize customer requests in a better and faster way.

Zendesk Guide provides end users with a complete self-service support options and empowers agents to better help customers [9]. It is possible to create a help center using Zendesk Guide. Customers can submit tickets/requests using help center. Ticket form and ticket fields are fully customizable. This functionality is used to solve forms problem and provide correct request forms for customer feedbacks and new business clients registrations.

Utilizing Zendesk Support and Zendesk Guide services - order changing and web forms registration problems are solved. Compared to the other possible ways of solving the problems, Zendesk based solution has a big advantage and is chosen by the company due to all parts being within one environment, one Zendesk system, one license plan.

2.2 Main Zendesk parts

The main Zendesk Support and Guide parts that are used for this work are:

- Tickets, ticket fields, ticket forms

- Triggers

- Automations

- Views

- Placeholders

- Tags

- API endpoints

Tickets are the main part of customer support workflow and Zendesk solution. It is a customer's request. Tickets are the means through which the end users (customers) communicate with agents in Zendesk Support. Tickets can originate from a number of channels, including email, help center, chat, phone call, social medias, or the API.

There are five standard statuses that can be applied to a ticket [10]:

1. New – indicates that no action has been taken on the ticket. Once a New ticket's status has been changed, it can never be set back to New.

2. Open – indicates that a ticket has been assigned to an agent and is in progress. It is waiting for action by the agent.

3. Pending – indicates that the agent is waiting for more information from the requester. When the requester responds and a new comment is added, the ticket status is automatically reset to Open.

4. On-hold – indicates that the agent is waiting for information or action from someone other than the requester.

5. Solved – indicates that the agent has submitted a solution.

Knowing how to manage ticket status changes, including when to use which status label, it is possible to make aggregation, navigation, and the overall customer-agent relationship smoother [10].

Ticket fields and ticket forms are used to provide necessary information related to the ticket. The predefined set of ticket fields are a ticket form. Ticket forms are visible to end users in the help center (request form) and to agents in a ticket within Zendesk [11].

There are two types of ticket fields [11]:

1. System ticket fields are the standard, default fields that agents see in a ticket.

2. Custom ticket fields can be created in addition to system ticket fields to gather additional information from the person/customer who is requesting support.

Typically, when end users submit support requests, they provide the subject and description of their question or support issue in system ticket fields. They may also be prompted to provide additional data using custom ticket fields [11]. In this work ticket fields are used to connect additional information such as order number with the changed order ticket. Ticket fields and ticket forms are also used to create feedback and client registration web forms.

Triggers run immediately after a ticket is created or updated and automatically perform actions if specified conditions are met. Triggers are created from conditions and actions. Conditions set the qualifications needed for the trigger to fire and actions represent what will be performed when those qualifications are met [12]. Examples of triggers usage within this work:

- Notifying customers

- Routing priority customer request into a specialized support group, view

- Adding and removing tags

Automations are similar to triggers because both define conditions and actions that modify ticket properties and optionally send email notifications to the customers and agents. The check for automation criteria runs every hour, the automation criteria can be

set on a time basis. Automations run on all non-closed tickets [13]. Using triggers and automations unique solution and workflow is developed according to the business needs.

Views are a way to organize tickets by grouping them into lists based on certain criteria [14]. These can simply be described as "Saved searches". View is defined using the conditions for the search [8]. In this work views are used to group orders based on their chosen delivery time and based on the order state/ticket status - new delivery time picked, or order change message ignored.

Placeholders are the elements that are contained within double curly brackets. They are references to ticket and user data that are inserted into ticket or email notifications when they are sent to customers [15]. Liquid template language is used for placeholders in Zendesk [16]. Liquid is an open-source template language created by Shopify and written in Ruby [17]. In designed solution placeholders are used to retrieve necessary information related to the order within notification messages.

Tags are words, or combinations of words, that are used to add more context to tickets. Tags provide an unlimited amount of flexibility to manage and customize support workflow [18]. In proposed solution tags are used to define different order changing messages and forms. Tags play a key role in activating different triggers.

API endpoints are used to further automate process and provide the best possible solution. Zendesk Support API endpoints have predictable resource-oriented URLs, accepts form-encoded request bodies, returns JSON-encoded responses, and uses standard HTTP response codes, authentication, and verbs [19]. API endpoints can be used for full Zendesk Support functionality – to create triggers, automations, tickets, fields and more. In this work API is used for tickets creation and ticket properties changing. All configurations are done in the Zendesk web environment. Web environment provides better overview, easier navigation and it is more convenient compared to the solution functionality creation using API endpoints.

2.3 Solution design

2.3.1 Forms

Zendesk help center environment is created within Zendesk Guide in order to provide web forms for client's feedback and new client registration. Idea is that new request form is added in the help center using pre-created ticket forms. Later request submission form page/URL from help center is used to navigate customers to the correct form and gather needed data.

2.3.2 Order changing

The main three parts of the order changing solution as described in problem definition are Notification, Change and Aggregation as presented in Figure 1.
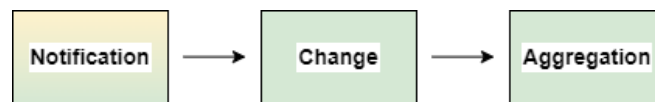
Figure 1. Simple solution diagram

Firstly, customers must receive information about their order and why it cannot be delivered in time. Then customer must be able to pick new suitable time for their order delivery. Finally, all changes must be gathered and grouped in a specific place.

A more expanded Zendesk based proposed solution is designed as presented in Figure 2. This diagram contains main logic and steps that must be done within the Zendesk.
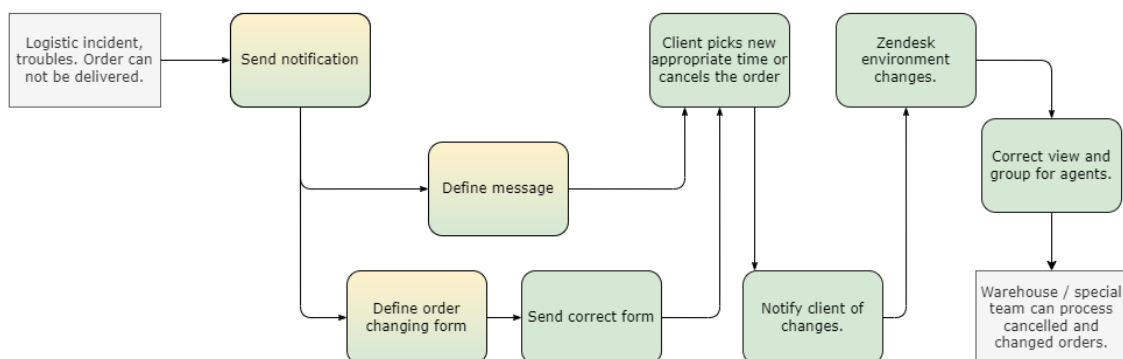
Figure 2. Expanded Zendesk solution diagram

The first step is to send a notification message containing reasons for goods not being delivered in time. For this part, a special web environment is created, from where

23

notifications are being sent, both email and SMS can be sent. SMS notification task is outside of this works scope. Next, customer must receive correct order changing message with appropriate delivery time frames based on their order. Order changing functionality within the form must be provided. Customer is able to pick new suitable time and this information reaches order ticket in the Zendesk environment. From Zendesk environment specialists can further work with changed orders with all needed information. Changed orders can be easily checked and retrieved to further transfer this information to the warehouse.

The design shown in Figure 2 uses only Zendesk Support functionality. Customers provide all the information needed to the company via choosing time frame within the message. It is a very fast and convenient task for customers. For agents, the new changed orders appear in Zendesk environment whenever customers have chosen suitable time based on the provided choices.

2.4 Zendesk solution concept

As a first step of technical solution implementation – Zendesk solution concept for main functionality is designed based on the solution diagram described in Figure 2. Using such table with triggers and necessary ticket fields a better overall image of the solution is created, and it is easier to understand what parts are missing and what needs to be further developed. Table 1 describes mandatory ticket fields and tags that must be created for the solution to operate properly.

Table 1. Ticket fields for orders changing system

Ticket object	Description	Example
Tag for time changing email form.	Specific tag is associated with a specific time changing form.	Admin writes "ny_late" tag in web environment. API adds this tag to a ticket. Appropriate trigger activates and sends form for New York.
Tag for sent SMS.	Specific tag that helps to determine whether SMS is sent. Later this tag is used to distinguish those tickets.	Admin picks 'Send SMS' in web environment. API adds tag 'outbound_sms'.

Ticket object	Description	Example
Tag for sent EMAIL.	Specific tag that helps to determine whether EMAIL is sent. Later this tag is used to distinguish those tickets. This tag is also used to send EMAIL notification to customer.	Admin picks 'Send E-MAIL' in web environment. API adds tag 'outbound_email'.
Ticket field with Order number.	Every order changing ticket in Zendesk system is associated with a specific customer's order for tracking, order changing and integrity purposes.	Web app reads Order numbers from spreadsheet file and API adds Order number to a ticket field. Later Order numbers are gathered from this ticket field.
Ticket field for chosen delivery time.	After customer picks time within order changing email, chosen time is added into this field. This field is used to later group orders for the agents.	Customer clicks "10:00 – 12:00". This ticket field becomes 10-12 and is used for agent Views creation.
Ticket requester.	Every order changing ticket is associated with a specific customer for tracking, order changing and integrity purposes.	Web app reads customer info from spreadsheet and API adds appropriate requester name and requester email details to the ticket.

Table 2 describes the necessary triggers for the correct automated workflow.

Table 2. Triggers and their role

Trigger	Action	Example
Send time changing form via email.	This trigger will send order time changing form based on the tag. Every form contains different time frames, message.	System will run this trigger and send time changing form once ticket with tag 'ny_late' appears.
Send EMAIL notification to the customer.	Sends EMAIL to the customer with the notification defined by an admin in the web environment.	Admin picks EMAIL notifications in web. System will run this trigger once ticket with tag 'outbound_email' appears.

Trigger	Action	Example
Send changed time notification to the customer.	Sends message to the customer with their chosen time.	Customer choses 12:00 – 14:00 time frame in the email form - receives email with the chosen time.
Add changed time field in View for agents.	Whenever customer changes time, trigger changes time ticket field to suit the chosen time. Needed for flawless agents' environment and work.	Customer chooses 12:00 – 14:00 time frame in the email form. For agents this order will appear under '12 – 14' time frame in the changed orders View.
Remove time from agents View when order was cancelled by customer.	If customer cancels the order, all tags and ticket fields associated with the time frames must be erased.	Customer cancels order in the email form. For agents this order will appear in the cancelled orders View.

2.5 Triggers and automations description in work

Trigger actions and configurations are designed before implementing the solution. In the solution part of this work same description approach is used to describe triggers, views, and automations.

To describe actions that are taken and what exactly activates the trigger/automation run - a simple text format is used. Figure 3 shows an example of trigger design description.

```
MEET ALL REQUIRMENTS
Ticket: Status | is | New


ACTIONS
Ticket: Add tags | new_status
```

Figure 3. Trigger description example

First section 'MEET ALL REQUIRMENTS' describes criteria for trigger run. 'MEET ALL REQUIRMENTS' is a logical AND, which means that all of the condition must be met [20]. It is possible to use 'MEET ANY REQUIRMENTS' which is a logical OR, in this case only one of the conditions must meet. In this work Logical OR is not used. In the example shown in Figure 3, when ticket status is 'New' (means customer did not receive any reply yet) – trigger action runs.

Next section "ACTIONS" describes operations that this trigger/automation will perform. In provided example (Figure 3), for all tickets that met the criteria and are in 'New' status additional tag 'new_status' is added. Zendesk Views configurations have only requirements/criterias field because views do not take any actions, they are simply a configuration for tickets listing.

Actions and requirements are self-explanatory and contain all the needed information to understand how the trigger is supposed to work. Additionally, in this work below every trigger design/configuration an additional explanation is always present.

In work process, the initial step is to design and document trigger, automation actions using descriptions. This helps to better visualize their relationship and order. The triggers in the Zendesk web environment are created only after the trigger descriptions are done.

Compared to the work method with no trigger designs/descriptions documentation, this simple approach helps to better understand the logic behind the created triggers and get a better overall picture of automation steps created in this work. This information and automation designs documentation can be later granted and presented to the company for whom this Zendesk solution is developed. This gives the company a good understanding of how the process is automated and will save time searching for specific automations.

3 Solution

This chapter contains the main solution and configuration. Described solution consists of six parts:

1. Creation of ticket forms for both problems.

2. Web environment for order changing system.

3. Initial email notification.

4. Order changing process.

5. Web forms.

6. Agents' side environment enhancements.

3.1 Tickets

The important part of both solutions are tickets. They contain necessary information related to the order, customer, feedback, or registration. All later automations, triggers and grouping views are created and work in cooperation with these Zendesk tickets.

In total three additional custom ticket forms are created:

- Order changing ticket

- Feedback ticket

- Business client registration ticket

For additional ticket forms, new empty ticket form with only default fields is used. Ticket fields such as subject, requester, tags are default, and are always present. Every custom field has a unique ID or key [21]. Unique for every ticket "Field id" is used to retrieve data from custom ticket fields into email message or within the ticket. For default ticket fields placeholders are used to retrieve the data.

Additional created ticket fields for order changing ticket are presented in Table 3. Order changing ticket form must contain two additional fields – Order number and chosen Delivery time. This information is mandatory to connect the customer's order and the ticket related to their order delivery change.

Table 3. Order changing ticket form additional fields

Ticket field name	Type	Field id	Usage
Delivery time	Drop-down (Time choices, delivery time frames)	36001394764	Used for tickets grouping in Views via delivery time.
Order nr.	Text	36001394755	Used to connect ticket with customer and their order.

Next are the ticket forms for business client registration and the feedback ticket form. Mandatory fields for both new business client and feedback forms are provided by the company. Later in the work, these two ticket forms are used within Zendesk Guide help center to create request web forms. Ticket form for new business client registration is presented in Table 4.

Table 4. New business client registration ticket form additional fields

Ticket field name	Type	Field id
Company name	Text	360022848341
Address	Text	360022848357
Register code	Text	360022848222
Contact person	Text	360022848124
Phone number	Numeric	19000064821
E-mail address	Text	360022848962
Company representative	Text	360022848641
Comment	Multi-line	360013947259

The feedback ticket form fields are shown in Table 5.

Table 5. Feedback ticket form additional fields

Ticket field name	Type	Field id
Feedback reason	Drop-down (Complaint, Praise, Proposal, Other)	360016524521
Online shop	Checkbox	360023587124
Name	Text	360021485213
Phone number	Numeric	19000064821
Topic	Text	360021463321
Comment	Multi-line	360013947259

3.2 Web environment

The first step in the order changing system is the notification. A special web page is created for support specilists who are going to work with the proposed solution. From this web page it is possible to send mass email or SMS or even both. Web page development - fields, file upload/reading and SMS notification system are outside of this thesis scope. Communication with the Zendesk system and API requests that the web page sends, are a part of the works scope. API requests and logic are described and provided. Web page contains these fields for submission:

- Subject

- Message

- Tags

- Spreadsheet submitting form

Fields are filled according to the needs and spreadsheet file is submitted with affected customers. "Send" button activates API request presented in Appendix 2. API request creates ticket in the Zendesk environment based on the provided data. Ticket creation activates the chain of automations.

On the web page, specialists can define specific tags. This tag goes into the created Zendesk ticket via API request (Appendix 2). This is an important field that is used to

send correct and different suitable order changing forms to the customers based on the time frames that the form contains.

Next is the spreadsheet upload functionality. Web page reads data from file. File structure is simple and is described in Table 6. This data is used inside Zendesk API request (Appendix 2) to create ticket for every affected customer's order.

Table 6. Spreadsheet file for affected orders/customers

Row	Explanation
Name	Used to add affected customer's name into the created ticket via API.
Email	Used to add affected customer's email into the created ticket via API. Used as the ticket requester.
Order. nr	Used to add affected order number into the created ticket via API.
Phone number	Used to send SMS notification message.

Correctly structured spreadsheet with affected by logistic incident customer orders is shown in Figure 4. Phone number is not mandatory and is used for the SMS notification sending.

Figure 4. Affected customer's file structure used for API

In the web page Subject field, ticket subject that is seen in Zendesk environment is defined, usually the subject is similar to "Company X - Logistic incident 06.03.2022". This way different notification messages can be grouped and distinguished in Zendesk environment based on their subject.

After web fields are filled, spreadsheet file is uploaded and the data is submitted - web solution creates tickets in Zendesk according to the provided data, given customers list.

Next is the notification message that customers will receive, it can be both SMS and email at the same time. With SMS it is simply the SMS text. With the email notification it is the initial notification message the the customer receives.

3.3 Initial email notification

For email notification API request described in Appendix 2 creates ticket in the Zendesk environment based on the information defined in web page fields. Message field from the web page is the first message within the created Zendesk ticket based on the API request. API adds specific tag when email notification checkbox is chosen in the web page – tag 'outbound_email'. The initial email notification trigger is presented in Figure 5.

```
MEET ALL REQUIRMENTS
Ticket | is | Created
Tags | Contains at least one of the following | outbound_email

ACTIONS
Email user | Requester
Email subject | Notification message
Email body |

Hello, {{ticket.requester.first_name}}

Notification from X:
{{ticket.description}}

Best wishes
Company X
```

Figure 5. Initial email notification trigger

This is a notification trigger that sends an email to the customer. When ticket is created and contains 'outbound_email' tag, which means that the send email notification checkbox is clicked, criteria for this notification message trigger is met and it will run.

Trigger action is to send email to the ticket requester. Ticket requester is the affected customer. Email contains message defined in the web environment (it is the first message

within the ticket). First comment/message within the ticket is defined via specific {{ticket.description}} placeholder [22].

Problem that appeared with the default placeholder usage is that the message is shown as it is written by the customer as shown in Figure 6.
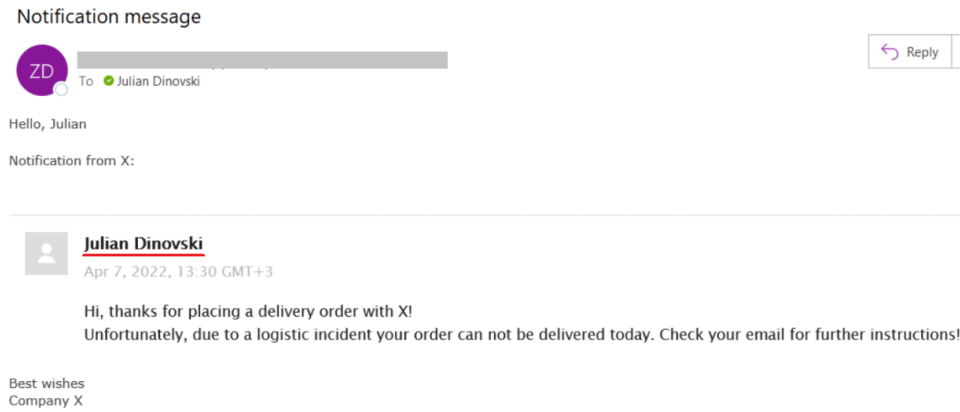
Figure 6. Notification message using placeholder

Message behaves this way, because the customer is the ticket requester and the first message in ticket is from the ticket requester (it is the notification message defined in the web). Such notification message cannot be used, because the message is defined by an admin in the web environment, not by the customer.

In Zendesk it is possible to develop own placeholders that can be used in automations and triggers using Liquid markup. Liquid allows to create simple programming logic such as case statements, if statements, for loops, and more [16]. Using Liquid markup, a better approach is developed instead of default placeholder. Updated email notification is presented in Figure 7.

Figure 7. Notification message using Liquid markup

33

Updated email notification trigger looks as described in Figure 8.
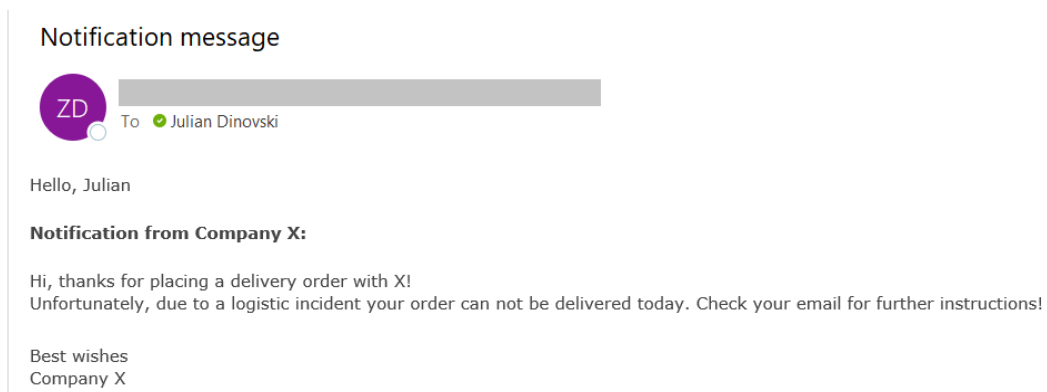
```
MEET ALL REQUIRMENTS
Ticket | is | Created
Tags | Contains at least one of the following | outbound_email

ACTIONS
Email user | Requester
Email subject | Notification message
Email body |

Hello, {{ticket.requester.first_name}}

{% for comment in ticket.comments %}
   <b>Notification from Company X:</b>
   {{comment}}
{% endfor %}

Best wishes
Company X
```

Figure 8. Updated email notification trigger

Solution goes through comments within the ticket. Ticket always has only one message within it when email notification trigger (Figure 8) is activated, because this trigger is activated only once when ticket is created. This way the first comment message is added into the email.

This approach presented in Figure 7 is much better compared to the default placeholder shown in Figure 6. It is simplified and contains only the main information that the customer must receive.

Using the web environment and Zendesk functionality, customer receives both SMS and email notifications and the order changing tickets are created in Zendesk environment. Next is the order delivery time changing.

3.4 Order changing process

The next step after sending the notification email about logistic incident is to send the order changing form. Trigger that is responsible for the order changing process is presented in Figure 9.

```
MEET ALL REQUIRMENTS
Ticket | is | Created
Tags | Contains none of the following | change_form_sent
Tags | Contains at least one of the following |
time_change_form_1


ACTIONS
Status | Pending
Email user | Requester
Email subject | Company X – Order delivery change
Email body | (Appendix 3)
Add tags | change_form_sent
```

Figure 9. Order change email message trigger

Important trigger action described in Figure 9 is to set the ticket's status to Pending. This means that the order change form is sent, and the order is waiting for the customer's choice/pick. Whenever customer picks suitable time, ticket will automatically go into the Open state, meaning that it has received the response from the customer.

Trigger (Figure 9) looks for a prohibiting tag to check whether the order changing form is not already sent. If it is already sent, trigger actions are ignored. Customers do not receive duplicates of order change form. Trigger looks for a specific tag, defined in the web environment by specialists (section 3.2). Different tags are associated with different order change email body, with its own unique time frames/choices. In trigger described in Figure 9 the message is connected with tag 'time_change_form_1' and the message body is defined in Appendix 3. Appendix 3 email contains six delivery choices. The email message that the customer receives is shown in Figure 10.
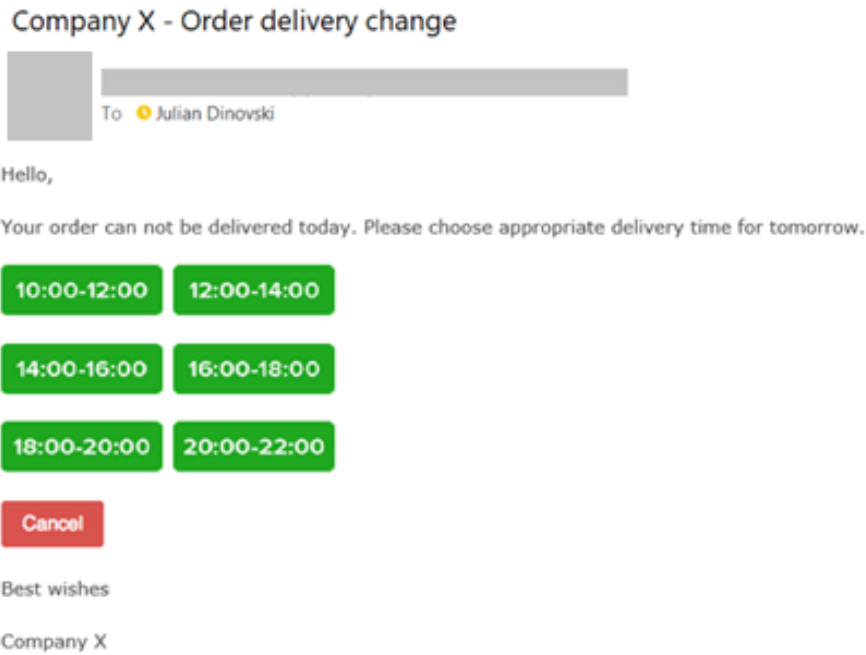
Figure 10. Order change message

Unique order change messages are created and connected with different triggers using different tags. Each message contains delivery time frames that suit different cities, counties. For example, presented in Appendix 3 order change message is connected with tag 'time_change_form_1' and contains all time frames. Message connected with tag 'time_change_form_2' can contain only early time frames, for example, 10:00-12:00, 12:00-14:00. Forms can always be changed, and new ones can be added.

In trigger shown in Figure 9 action is to send an email with a specific time frames, based on the tag. Every time frame within the message is an image with a link. Image URL example is shown in Figure 11.

```
https://dev-env.com/{{ticket.id}}/{{url_number}}
```

Figure 11. Order change URL

This URL is a specific order changing API hosted on company's server. The API hosting and the server part is outside of this work scope. The API related to the Zendesk logic is created and provided. URL contains ticket ID using placeholders [22]. The next URL part is a number that defines actions described in Table 7.

Table 7. URL actions

URL number	Action
1	Cancel order. Adds 'order_cancelled' tag to the ticket.
2	Pick 10:00 – 12:00 delivery time frame. Adds '10-12' tag to the ticket.
3	Pick 12:00 – 14:00 delivery time frame. Adds '12-14' tag to the ticket.
4	Pick 14:00 – 16:00 delivery time frame. Adds '14-16' tag to the ticket.
5	Pick 16:00 – 18:00 delivery time frame. Adds '16-18' tag to the ticket.
6	Pick 18:00 – 20:00 delivery time frame. Adds '18-20' tag to the ticket.
7	Pick 20:00 – 22:00 delivery time frame. Adds '20-22' tag to the ticket.

In API request ticket ID for adding or removing tags from the ticket is taken from the URL part. The tag adding API request is described in Appendix 4. It is important to delete old tags if customer picks new delivery time after the initial choice. Appendix 5 represents the API request that runs before tag adding API request to make sure that only the last delivery choice remains. Whenever client clicks the button, new tag is added and old tag if presented is deleted. If client does not click the button, the order ticket remains in the Pending status.

After client's decision is finalized and client chooses delivery time, it is mandatory to send confirmation message to the customer containing changed order delivery information. This notification trigger is shown in Figure 12.

```
MEET ALL REQUIREMENTS
Ticket | is | Updated
Ticket: Status | Not changed to | Solved
Ticket: Status | Not changed to | Closed
Ticket: Tags | Contains at least one of the following |
order_cancelled order_ch_10-12 order_ch_12-14 order_ch_14-16
order_ch_16-18 order_ch_18-20 order_ch_20-22


ACTIONS
Ticket: Status | Open
Email user | Requester
Email subject | Company X – Order delivery changed
Email body | (Appendix 6)
```

Figure 12. Order change confirmation trigger

Trigger (Figure 12) checks for ticket status. It is a change from Pending into Open ticket status in cases where client picks new delivery time for the first time. Ticket status remains in Open status when customer rethinks their chosen delivery time for additional time. Status check is important. When agents are working with changed order ticket in Zendesk environment and for example, close or solve the ticket, customer must not receive additional email message because of the trigger activation.

Next is the check for a tag. The ticket tag must contain one of the possible time choices presented in the email order changing form described in Figure 10.

The action of the confirmation trigger shown in Figure 12 is to send simple email message. Message contains information about the customer's order and the change that customer chose. The email message that customer receives after picking suitable delivery time or cancelling the order is shown in Figure 13.
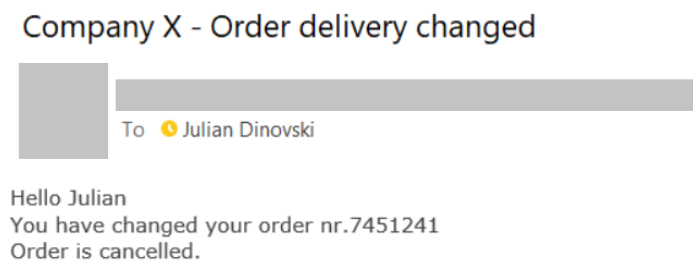
Figure 13. Confirmation message

3.5 Web forms

In Zendesk two help centers are created in order to have two different web forms. It is possible to have both request forms on the same help center page, using dropdown field to select correct one, however, this is not the best approach based on the company requirements. Two different help centers are created, each with its own one request form:

- Help center for feedback – "feedback.x.com"

- Help center for new business client registration – "newclient.x.com"

In ticket forms settings, for feedback help center all other ticket forms are disabled for requests submission, and only the feedback ticket form is enabled (section 3.1, Table 5).

For new business client help center only the new business client registration ticket form is enabled (section 3.1, Table 4).

After the help center creation and request forms configuration, feedback form can be found under www.feedback.x.com/requests/new and new business client registration form under www.newclient.x.com/requests/new. Feedback form is shown in Appendix 9, new business client registration form is shown in Appendix 10.

Company uses and sends the help center/request pages/forms URL to the customers if they want to gather feedback and if new business client needs to fill the registration form.

3.6 Agent side enhancements

Eventually, after the order changing and forms solutions are implemented and configured, it is mandatory to gather all requests, changed orders and feedbacks in suitable for customer support specialists place. For that Zendesk Support views are used and created.

3.6.1 Order changes

The changed orders must be grouped in one specific view. Changed orders have two states – client picked new time and client did not pick new time yet. In implemented solution those two states are distinguished by the ticket status. Ticket remains in 'Pending' status until client picks suitable delivery time. View for sent change order email and not picked new delivery time is described in Figure 14.

```
MEET ALL REQUIREMENTS
Form | Is | Order
Status | Is | Pending
Tags | Contains at least one of the following | change_form_sent
```

Figure 14. Sent order change message view

Firstly, ticket form is checked. It must be order changing ticket (section 3.1, Table 3). Next is the status criteria, it must be 'Pending' status which means that the customer did not pick time yet. Lastly, check for order changing email form being sent, tag must present.

After support specialists sends mass notification email, defines tag for order changing form, order change ticket appears under this view described in Figure 14. Agents' visual view is presented in Figure 15. Enabled columns (ticket fields) for sent order change email view are – subject, requester, requested time and order number.

Figure 15. Orders. Agents' view

Next are changed orders with picked new delivery time. It is required by the company to group orders by delivery time. Whenever customer picks suitable time frame a new tag is added. The problem is that it is not possible to group tickets by the ticket tag in view. Additional ticket field 'Delivery time' is created (section 3.1, Table 3) in order to group changed orders. However, tag is added to the ticket via clicking the time frame in email (sec 3.4, Figure 10), the 'Delivery time' ticket field remains unchanged and additional trigger for that purpose is created. Trigger that defines customer picked time is presented in Figure 16.

```
MEET ALL REQUIREMENTS
Ticket | is | Updated
Tags | Contains at least one of the following | order_ch_10-12

ACTIONS
Ticket field: Delivery time | 10-12
```

Figure 16. Delivery time trigger

Trigger looks for time frame tag and defines Delivery time ticket field based on it. Six different triggers with different time frames are created identical to the trigger presented in Figure 16. Whenever client picks new delivery time, ticket field always changes accordingly to the present time tag and the order change ticket can be grouped using delivery time ticket field.

40

Additional unique trigger is created for order cancellation ticket field. Trigger is described in Figure 17.

```
MEET ALL REQUIREMENTS
Ticket | is | Updated
Tags | Contains at least one of the following | order_cancelled


ACTIONS
Ticket field: Delivery time | -
```

Figure 17. Order cancellation trigger

This trigger looks for order cancellation tag and if it presents, trigger erases the delivery time ticket field, which means that the customer cancelled the order delivery.

Finally, after the entire delivery time ticket field logic is created, it is possible to group changed orders by their delivery time. Changed orders view configuration is presented in Figure 18.

```
MEET ALL REQUIREMENTS
Form | Is | Order
Status | Is | open
Tags | Contains at least one of the following | order_cancelled
order_ch_10-12 order_ch_12-14 order_ch_14-16 order_ch_16-18
order_ch_18-20 order_ch_20-22
```

Figure 18. Changed order's view

The ticket form must be the Order changing ticket form. Status of the ticket must be 'Open', this means that the customer picked new time or cancelled the order. Next is the check for one of the order changing tags, one of them must present. In view settings, view (Figure 18) is grouped by the Delivery time ticket field in ascending order. Agents' visual view for changed orders is presented in Figure 19.

Figure 19. Changed orders. Agents' view

3.6.2 Forms

For feedback form a simple view is created. Feedback view configuration is presented in Figure 20.

```
MEET ALL REQUIREMENTS
Form | Is | Feedback
```

Figure 20. Feedbacks' view

Whenever customer submits feedback form, ticket will appear within this view, because requests are made using feedback ticket form.

For new business client's registrations, no additional view is needed. Instead, a unique trigger that notifies specialist working with new business clients is created. It is requested by the company support specialist to receive all new business client messages in the mailbox. The specialist does not work with Zendesk that much and this way there is no need to login into Zendesk and check for new registrations that often. The trigger is shown in Figure 21.

```
MEET ALL REQUIREMENTS
Ticket | is | Created
Form | Is | New business client registration form


ACTIONS
Email user | Company business specialist
Email subject | New business client
Email body | (Appendix 7)
```

Figure 21. New business client trigger

Trigger (Figure 21) activates whenever new business client registration form is filled. Trigger checks for ticket form, it must be new business client registration form. The ticket is just created, it is an initial trigger run on new ticket. Trigger sends an email notification to the specific specialist that works with company business partners, new business clients. Email body is described in Appendix 7, it contains all the information from the filled web form (section 3.1, Table 4). Email that the specialist receives is shown in Figure 22.

New business client

To ✅ specialist's email

New business client (filled web form)

E-mail: abc@abc.com
Company name: ABC OÜ
Company registers code: 1234567
Address: Tee 41-1
Contact person: Julian Dinovski
Company representative: John Doe
Phone number: +372 5432 2345
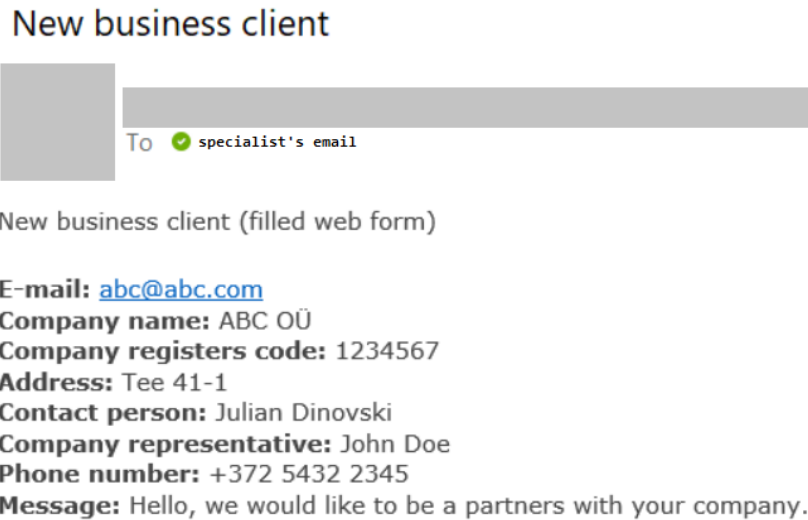**Message:** Hello, we would like to be a partners with your company.

Figure 22. New business client filled form notification

4 Experiment

After the solution implementation and configuration, the support specialists group that consists of 5 agents get to use and practice with the implemented system in an experimental way for few weeks.

This is a very important part of the work, as it helps to see the solution from a different angle and detect problems, defects, flaws that are unnoticed during the design and implementation processes.

Support specialists are submitting feedback in a free form after the testing period is over. Based on the gathered feedback the proposed solution is updated and problems are fixed.

4.1 Improvements based on feedback

This chapter describes enhancements to the proposed solution based on the support specialists and customers feedback.

4.1.1 Order change emails via HTML

Problem with the email message presented in Appendix 3 appeared. During the solution implementation described in the thesis two mailboxes with enabled content auto-download are used and the problem is not discovered. During the experiment by the customer support specialists, two customers contacted the company with a problem that they cannot pick suitable time. They only see red crosses – which means that they need to manually download the images, however, this is not intended and can lead to misunderstandings and interruptions in work.

The solution proposed in thesis uses images as buttons, then buttons redirect to the correct web page which adds tags to the ticket. The problem is that certain mailboxes have disabled content download by default. Including images. Situation is presented in Figure 23.

Figure 23. Time choice buttons with disabled content load

To solve this problem buttons are defined using HTML instead of images. The updated order changing email body is presented in Appendix 8. Implemented solution buttons are shown in Figure 24.

Figure 24. Time frames using HTML buttons: (A) Gmail, (B) Outlook

As compared to the situation shown in Figure 23, approach in Figure 24 ensures that the customer can always see the offered delivery time frames and no accidents and disruptions take place.

4.1.2 Time restriction for order changing system

Major issue detected during the testing period is that customers can change their orders delivery without any limitations, and it is hard to gather final state of information. For example, customers can change their orders five minutes after the information to the warehouse is already delivered. Perhaps, the customer might rethink and change previous/initial choice, and this poses a challenge.

The decision by the company is that the customers must have a possibility to change their order within a one-hour frame after the first choice. This means that if customer clicks

time frame 12:00 – 14:00 in the received mail form, within one hour after this initial choice, customer can still change the preferable time or cancel the order. After one hour passes, none of the changes are applied to the order in Zendesk environment.

To achieve this goal an automation rule is created. This automation runs every hour and adds specific tag to the ticket. Additionally, all triggers related to the order changing within the Zendesk system are reconfigured and have additional criteria check for this specific tag. Figure 25 represents flow how other triggers are going to work after the forbidding tag solution implementation.
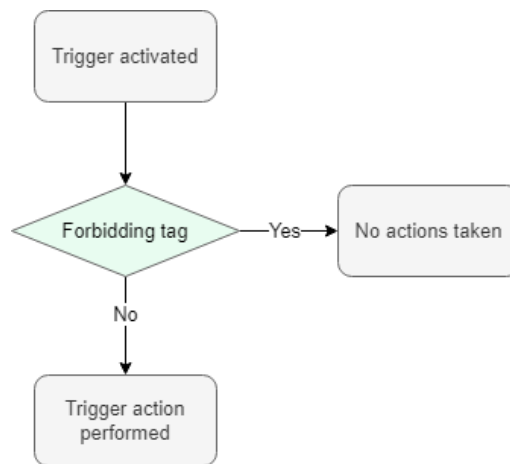
Figure 25. Order change prohibiting using tag

Triggers check for this one-hour passed automation tag and if this tag is present then trigger actions are cancelled, no matter what other criterias are met and no changes to the order ticket are done. Time changing restriction automation rule configuration is described in Figure 26.

```
MEET ALL REQUIREMENTS
Ticket: Status | is | Open
Ticket: Tags | Contains at least one of the following |
order_cancelled order_ch_10-12 order_ch_12-14 order_ch_14-16
order_ch_16-18 order_ch_18-20 order_ch_20-22
Ticket: Hours since open | (calendar) Greater than | 1
Ticket: Tags | Contains none of the following |
hour_after_choice


ACTIONS
Ticket: Add tags | hour_after_choice
```

Figure 26. Time restriction automation

Firstly, ticket status is checked, if status is 'Open' this mean that the customer has chosen suitable time and clicked the button, because in their initial state order change tickets have 'Pending' status and are waiting for customer's decision. Then there is a check for one of the order changing tags, it must be order cancelling tag or chosen time tag. After that a time check is added, whether one-hour have passed since the first decision. Final check is whether the ticket already does not contain this order change prohibiting tag. An automation must contain a condition that is true only once or an action that nullifies at least one of the conditions [13].

Then for every trigger related to the orders changing system in Zendesk environment an additional filter is added to check for this prohibiting tag as shown in Figure 27.

```
MEET ALL REQUIREMENTS
Ticket: Tags | Contains none of the following |
hour_after_choice
```

Figure 27. Check for prohibiting tag

After this solution implementation if one hour passes since the initial choice, none of the changes are applied to the order.

5 Implemented solution results analysis

The result of the implemented solution is a working order changing system that utilizes Zendesk platform. The feedbacks and new business client's registration forms are implemented.

Based on the company requirements, the developed solution is used to:

- Notify customers about the delivery issues

- Provide order delivery changing forms to resolve delivery issues

- Gather all changed orders in one specific place

- Provide feedback and registration forms to the customers

- Analyze and group customer feedback

- Optimize the registration process with the new business clients

The main aim of the implemented solution is to save agents time on resolving logistic issues with the orders. Compared to the previous monthly time cost of about 25-30 hours on resolving delivery issues, with the developed solution, it takes only few minutes for support specialists to submit spreadsheet of affected customers and fill the needed data. As compared to the time consumption of the previous manual approach - the proposed solution has minimized the time to less than an hour per month.

Based on the analysis, the company, and its workflow in terms of the time consumption will benefit from the investment into the proposed custom developed Zendesk solution within less than a year, with a yearly return of investments being equal to approximately 60%. This leads to the evaluation that the investment is justified and contributes to the company service quality and performance.

6 Future goals

The implemented solution is developed based on the company requirements and needs. The solution solves described problems and performs required tasks. Despite that, the solution can be further developed, and additional functionality and automation can be implemented. The further development ideas that are discussed with the company include:

- Automated warehouse notification system when the new order delivery choice is finalized

- Automated feedback form sending after the customer problem is solved

- Automated customer reminder after new order time is not picked for prolonged period of time

- Coupon code system to automatically send coupon codes to the customers affected by the logistic incident

System will be further developed and enhanced based on the company business needs, requirements and their opportunities. The proposed solution in its current state described in the thesis, fulfilled the main objectives, and resolved the time consumption issues.

7 Conclusion

The aim of this bachelor's thesis is to analyze and develop the best possible notification and order changing solution, which allows to notify client of any delivery issues and provide them with an opportunity to pick another suitable order delivery time.

The company for whom the proposed solution is designed, uses Zendesk platform to resolve customer issues and the developed order changing solution utilizes Zendesk platform.

In the analysis part, three different Zendesk services are analyzed and the custom Zendesk Support solution development for order changing problem is chosen as it leads to the best yearly return of investments (60%) for the company.

Zendesk functionality is described and analyzed, and the best possible solution is designed and implemented based on the company requirements and needs in the practical part of this work. Solution for customer's feedback gathering and new business client's registration is implemented.

After the solution implementation, the system is used by the support specialists for testing, in order to gather feedback and improve the solution. Based on the feedback, problems are eliminated, and the workflow is improved.

The main aim of the implemented solution is to save agents time on resolving delivery issues with the orders. The results show a significant decrease in time consumption using the proposed solution. The thesis result greatly contributes to the company service quality and performance.

References

[1]   *Automation Is the Future of CX.* R 'Ray' Wang, 2020. [Online]. Available:
        https://web.p.ebscohost.com/ehost/detail/detail?vid=6&sid=2174ebd5-c511-45b1-a9ca-
        a8c26f5e9d0d%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZSZz
        Y29wZT1zaXRl#AN=145044109&db=bth. [Accessed: 23.02.2022]

[2]   *Guest editorial: How technology is changing the design and delivery of services.* Mark
        M. Davis, James C. Spohrer and Paul P. Maglio, 2011. [Online],

        Available: https://link.springer.com/article/10.1007/s12063-011-0046-6.

        [Accessed: 25.02.2022]

[3]   Zendesk Gather. [Online]. Available: https://support.zendesk.com/hc/en-
        us/articles/4408882689306-Getting-started-with-Gather-for-your-community.
        [Accessed: 11.02.2022]

[4]   Zendesk Customer lists add-on. [Online]. Avalable: https://support.zendesk.com/hc/en-
        us/articles/4408845617306-Creating-and-using-customer-lists. [Accessed: 13.02.2022]

[5]   Email forwarding to Zendesk Support. [Online]. Available:
        https://support.zendesk.com/hc/en-us/articles/4408886828698-Forwarding-incoming-
        email-to-Zendesk-Support. [Accessed: 20.03.2022]

[6]   Zendesk licenses and plans. [Online]. Available: https://support.zendesk.com/hc/en-
        us/articles/4408881937306-Getting-Started-with-Zendesk-Suite.
        [Accessed: 11.02.2022]

[7]   LimeSurvey software. [Online]. Available:
        https://github.com/LimeSurvey/LimeSurvey. [Accessed: 21.02.2022]

[8]   Stafford Vaughan and Anton de Young, *Zendesk Administration: A World-Class
        Customer Service Platform, 2nd edition.* O'Reilly Media, 2014.

[9]   Zendesk Guide resources. [Online]. Available: https://support.zendesk.com/hc/en-
        us/articles/4408823842842-Guide-resources#topic_o1g_qqx_sqb.

        [Accessed: 21.02.2022]

[10]   Tickets and their states. [Online]. Available: https://support.zendesk.com/hc/en-
        us/articles/4408832151834-Updating-and-solving-tickets.

        [Accessed: 25.03.2022]

[11]   Zendesk Support ticket fields, ticket forms. [Online]. Available:
        https://support.zendesk.com/hc/en-us/articles/4408886739098-About-ticket-fields.
        [Accessed: 03.04.2022]

[12]   Zendesk Support triggers. [Online]. Available: https://support.zendesk.com/hc/en-
        us/articles/4408822236058-About-triggers-and-how-they-work. [Accessed: 03.04.2022]

[13]   Zendesk Support automations. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408832701850-About-automations-and-how-they-work.

[Accessed: 03.04.2022]

[14]   Zendesk Support views. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408888828570-Creating-views-to-manage-ticket-workflow.
[Accessed: 03.04.2022]

[15]   Support Email channels. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408888639258-Getting-started-with-email-Part-1-How-the-email-channel-works. [Accessed: 11.04.2022]

[16]   Liquid markup Zendesk. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408883291290-Understanding-Liquid-markup-and-Zendesk-Support.
[Accessed: 21.03.2022]

[17]   Shopify Liqud. [Online]. Available: https://shopify.dev/api/liquid.
[Accessed: 21.03.2022]

[18]   Zendesk tags. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408888664474-About-tags. [Accessed: 03.04.2022]

[19]   Zendesk Support API documentation. [Online] Available: https://developer.zendesk.com/api-reference/ticketing/introduction/.
[Accessed: 11.02.2022]

[20]   Conditions reference. [Online]. Available: https://developer.zendesk.com/documentation/ticketing/reference-guides/conditions-reference/. [Accessed: 11.04.2022]

[21]   Custom ticket field references. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408887218330-Using-placeholders#topic_nfp_nja_vb.
[Accessed: 11.04.2022]

[22]   Zendesk Ticket placeholders. [Online]. Available: https://support.zendesk.com/hc/en-us/articles/4408886858138-Zendesk-Support-placeholders-reference.
[Accessed: 03.04.2022]

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Julian Dinovski

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Optimizing Support Experience Using Zendesk Platform", supervised by Tauseef Ahmed.

   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

24.04.2022

---

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – API request to create order changing ticket

```
curl https://x.zendesk.com/api/v2/tickets.json -d \
{
  "ticket": {
    "requester": {"name": NAME_FROM_SPREADSHEET, "email":
EMAIL_FROM_SPREADSHEET
    },
    "assignee_id": 17562,          # Id of a specific account
    "comment": {
      "body": MESSAGE_FROM_FORM
    },
    "priority": "urgent",
    "subject": SUBJECT_FROM_FORM,
    "tags": [TAG_FROM_FORM, "outbound_sms", "outbound_email"],
    "custom_fields": {
      "id": 36001394755,           # "Order nr." ticket field id
      "value": ORDER_NUMBER_FROM_SPREADSHEET
    }
  }
} \
-H "Content-Type: application/json" -v -u \
julian.dinovski@x.x/token:API_TOKEN -X POST
```

Appendix 3 – Notification email template with images

```
Hello,

Your order can not be delivered today. Please choose appropriate
delivery time for tomorrow. <br>

<a href="https://dev-env.com/{{ticket.id}}/2"><img
src="https://x.com/x.png" alt="10:00-12:00"> </a><a
href="https://dev-env.com/{{ticket.id}}/3" img
src="https://x.com/x.png" alt="12:00-14:00"></a>

<a href="https://dev-env.com/{{ticket.id}}/4" img
src="https://x.com/x.png" alt="14:00-16:00"> </a><a
href="https://dev-env.com/{{ticket.id}}/5" img
src="https://x.com/x.png" alt="16:00-18:00"></a>

<a href="https://dev-env.com/{{ticket.id}}/6" img
src="https://x.com/x.png" alt="18:00-20:00"> </a><a
href="https://dev-env.com/{{ticket.id}}/7" img
src="https://x.com/x.png" alt="20:00-22:00"></a>

<a href="https://dev-env.com/{{ticket.id}}/1" img
src="https://x.com/x.png" alt="Cancel order"></a><br>

Best wishes
Company X
```

Appendix 4 – Tag adding API request

```
curl https://x.zendesk.com/api/v2/tickets/update_many.json?ids=
{{ticket.id}} \
  -d '{"ticket": {"additional_tags":["order_ch_XX-XX"]}}'\
  -H "Content-Type: application/json" \
  -v -u julian.dinovski@x.x/token:API_TOKEN -X PUT
```

Appendix 5 – Tag deletion API request

```
curl https://x.zendesk.com/api/v2/tickets/update_many.json?ids=
{{ticket.id}} \
  -d '{"ticket": {"remove_tags":["order_cancelled", \
  "order_ch_10-12", "order_ch_12-14", "order_ch_14-16", \
  "order_ch_16-18", "order_ch_18-20", "order_ch_20-22"]}}'\
  -H "Content-Type: application/json" \
  -v -u julian.dinovski@x.x/token:API_TOKEN -X  PUT
```

Appendix 6 – Order change customer notification email

```
Hello {{ticket.requester.first_name}}
You have changed your order
nr.{{ticket.ticket_field_36001394755}}
{% if {ticket.tags} contains 'order_ch_10-12' %}
Delivery time: 10:00 - 12:00
{% endif %}
{% if {ticket.tags} contains 'order_ch_12-14' %}
Delivery time: 12:00 - 14:00
{% endif %}
{% if {ticket.tags} contains 'order_ch_14-16' %}
Delivery time: 14:00 - 16:00
{% endif %}
{% if {ticket.tags} contains 'order_ch_16-18' %}
Delivery time: 16:00 - 18:00
{% endif %}
{% if {ticket.tags} contains 'order_ch_18-20' %}
Delivery time: 18:00 - 20:00
{% endif %}
{% if {ticket.tags} contains 'order_ch_20-22' %}
Delivery time: 20:00 - 22:00
{% endif %}
{% if {ticket.tags} contains 'order_cancelled' %}
Order is cancelled.
{% endif %}
```

Appendix 7 – New business client notification email template

```
New business client (filled web form)

<b>E-mail:</b> {{ticket.requester.email}}
<b>Company name:</b> {{ticket.ticket_field_360022848341}}
<b>Company registers code:</b>
{{ticket.ticket_field_360022848222}}
<b>Address:</b> {{ticket.ticket_field_360022848357}}
<b>Contact person:</b> {{ticket.ticket_field_360022848124}}
<b>Company representative:</b>
{{ticket.ticket_field_360022848641}}
<b>Phone number:</b> {{ticket.ticket_field_19000064821}}
<b>Message:</b> {{ ticket.ticket_field_360013947259}}
```

Appendix 8 – Notification email HTML template

```html
<style>
 .email_x_button {
      background-color: #4CAF50;
      border: none;
      color: white;
      padding: 10px 20px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 18px;
      margin: 6px 4px;
}
.email_x_button2 {
      background-color: #DC143C;
      border: none;
      color: white;
      padding: 8px 30px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 18px;
      margin: 12px 4px;
}
</style>

Hello,

Your order can not be delivered today. Please choose appropriate
delivery time for tomorrow. <br>

<a href="https://dev-env.com/{{ticket.id}}/2" style="text-
decoration: none" class="email_x_button">10:00 -
12:00</a>   <a href="https://dev-
env.com/{{ticket.id}}/3" class="email_x_button">12:00 -
14:00</a><br>

<a href="https://dev-env.com/{{ticket.id}}/4"
class="email_x_button">14:00 - 16:00</a>   <a
href="https://dev-env.com/{{ticket.id}}/5"
class="email_x_button">16:00 - 18:00</a><br>
```

```
<a href="https://dev-env.com/{{ticket.id}}/6"
class="email_x_button">18:00 - 20:00</a>   <a
href="https://dev-env.com/{{ticket.id}}/7"
class="email_x_button">20:00 - 22:00</a>
<br>

<a href="https://dev-env.com/{{ticket.id}}/1"
class="email_x_button2">Cancel order</a><br>

Best wishes
Company X
```

Appendix 9 – Feedback form

Submit a request

Your email address

Feedback reason

-   ▾

Online shop (optional)

☐

Name

Phone number

Subject

Comment

T B *I*	☰ ☰	☒ 🔗 ❞

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Attachments (optional)

Add file or drop files here

Submit

Appendix 10 – New business client registration form

Submit a request

Your email address

Subject

Company name

Address

Register code

Contact person

Phone number

Company representative

Comment

| T | B | *I* | | ⋮☰ | ☰ | | ▣ | ∞ | ❞ |

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Attachments (optional)

Add file or drop files here

Submit