

Tukuma Raiņa ģimnāzija

Īsziņu sūtīšanu automatizācija

Zinātniski pētnieciskais darbs dabaszinātņu nozarē, datorzinātņu un informātikas
apakšnozarē

Darba autors:

Ilja Polozovs, 11.klase

Darba vadītāja:

Anna Beāte Būmane
Tukuma Raiņa ģimnāzijas
informātikas skolotāja

Tukums, 2021

Anotācija

Darba tēma: Īsziņu sūtīšanu automatizācija.

Darba autors: Tukuma Raiņa ģimnāzijas 11.b klases skolnieks Ilja Polozovs.

Darba vadītājs: Tukuma Raiņa ģimnāzijas informātikas skolotāja Anna Beāte Būmane.

Darba saturs: darbā ir aprakstīta automatizācijas vēsture, kas ir automatizācija un kā to pielietot īsziņu sūtīšanai. Praktiskajā daļā aprakstīts programmas kods, kas spēj automatizēt īsziņu sūtīšanu no lietotāja lietotājam, kā arī tika veikta programmas uzlabošana, lai notiktu automātisko atbilžu sūtīšana.

Darba mērķis: noskaidrot, vai ir efektīvi automatizēt īsziņu sūtīšanu starp lietotājiem, kā arī pielietot to biznesā.

Pētnieciskā darba uzdevumi:

- izpētīt automatizācijas attīstības vēsturi;
- izpētīt pieejamo teoriju par telefoniju;
- apskatīt, kur to pielieto;
- pārbaudīt, vai var automatizēt īsziņu sūtīšanu, uzrakstot programmu *Python* programmēšanas valodā;
- aprakstīt programmas kodu;
- noskaidrot, kur vēl var pielietot automātisko īsziņu sūtīšanu;
- apkopot iegūtos datus.

Hipotēze: izmantojot īsziņu automātisko sūtīšanu ir iespējams mazināt rutīnas darbu ciklu, atvieglot dažus dzīves aspektus, pielietot izveidoto programmu nelielā biznesā.

Secinājumi: izmantojot uzrakstīto programmu ir iespējams komunicēt ar lietotājiem, atvieglot dažu uzņēmumu darbu, kas būs efektīvi ne tikai no darbinieku slodzes samazināšanas viedokļa, bet arī finanšu ziņā.

Atslēgas vārdi: automatizācija, īsziņu sūtīšanas automatizācija, automatizētā atbilde, SMS ziņojumi, telefonija.

Darba struktūra: darbs satur 16 lapaspuses un 14 attēlus.

Abstract

The title of the research paper: Texting automation.

Author: Tukums Rainis Gymnasium form 11b student Ilja Polozovs.

Advisor: Tukuma Raina Gymnasium informatics teacher Anna Beāte Būmane.

The content of the paper: : in the theoretical part, research paper provides information about history of automation, what automation is and how to use it to send text messages. The practical part describes the program code, which is able to automate the sending of text messages from the user to the user, as well as the program was improved to send automatic replies.

The aim of the paper: find out if it is effective to automate texting between users as well as use in business.

The objectives:

- study the history of automation development;
- study the available theory of telephony;
- see where it is used;
- test whether texting can be automated by writing programm in Python programming language;
- describe the program code;
- find out where automatic text messaging can still be used;
- collect the obtained data.

Hypothesis: using automatic text messaging, it is possible to reduce the work cycle, facilitate some aspects of life, apply the created program to a small business.

Conclusions: using the written program it is possible to communicate with users, to facilitate the work of some companies, which will be effective not only in terms of reducing the load, but also financially.

Keywords: automation, text message automation, automated reply, SMS messages, telephony.

The structure of the paper: the work contains 17 pages and 14 images.

Saturs

Anotācija	2
Abstract	3
Skaidrojumi	5
Ievads	6
1. Teorētiskā daļa	7
1.1. Datoru automatizāciju ideju sākumi.....	7
1.2. Automatizācija IT sfērā.....	7
1.3. Elektroniskā komunikācija un SMS ziņojumi.....	8
1.4. Telefonijas pakalpojumu sniedzēji.....	8
1.5. Twilio	8
1.6. <i>Python</i> valoda.....	9
1.7. Integrētās izstrādes vide - <i>PyCharm IDE</i>	10
2. Praktiskā daļa	11
2.1. Telefonijas pakalpojumu sniedzēja izvēlēšanās.....	11
2.2. Vides sagatavošana SMS sūtīšanai	12
2.3. SMS sūtīšana.....	12
2.4. SMS ziņojumu automatizētā atbilde	13
2.5. SMS ziņojumu automātisko atbilžu paplašinājums un pielietojums	15
Secinājumi.....	17
Izmantotās literatūras un interneta resursu avoti.....	18

Skaidrojumi

Python	Augsta līmeņa vispārējas nozīmes programmēšanas valoda.
AI	Mākslīgais intelekts.
Cloud Drive	Krātuves vieta attāļajā serverī.
SOE	Standarta darbības vide.
PBX	Korporatīvās telefonijas sistēma.
PSTN	Publiskais telefonu tīkls.
Cloud telephony	Tālruņa sistēma, kas darbojas caur interneta savienojumu.
Twilio	Mākoņu sakaru platforma.
SETL	Augsta līmeņa programmēšanas valoda.
Google	Programma, kas meklē un identificē vienumus datu bāzē.
YouTube	Video mitināšanas serviss.
Dropbox	Failu mitināšanas serviss.
Netflix	Filmu un seriālu straumēšanas multivide.
Hulu	Vietne, kas piedāvā piekļuvi TV pārraižu straumēšanai.
IDE	Izstrādes vide jeb programmatūras rīku komplekts.
PyCharm	Integrēta Python programmēšanas valodas izstrādes vide.
JetBrains	Uzņēmums - izstrādā programmēšanas valodu izstrādes rīkus.
IntelliJ IDEA	Integrēta programmatūras izstrādes vide.
Twilio API	(Lietojumprogrammu saskarne) nodrošina platformu saziņai.
ID	Rakstzīmju virkne, ko izmanto mainīgā identificēšanai.
Flask	Ietvars tīmekļa programmu izveidei Python valodā.
URL	Vienotais resursu vietrādis.
Ngrok	Programma, kas atklāj vietējos serverus publiskam internetam.
Firewall	Ugunsbūris, kas filtrē interneta tīkla trafiku.
Ports	Nodrošina īpašu fizisku saskarni starp ierīcēm.
NAT	"Tīkla adresu tulkošana". NAT tulko lokālā tīkla datoru IP adreses uz vienu IP adresi.

Ievads

Mūsdienās tehnoloģiju attīstības ātrums ir pārsteidzošs. Zinātniskā un tehnoloģiskā progresa lēcieni pēdējos gados var salīdzināt tikai ar kosmosa nozares attīstības tempu laika posmā no 50. gadu beigām līdz divdesmitā gadsimta 70. gadu vidum. Tā kā tad cilvēka klātbūtne kosmosā kļuva par realitāti, tā tagad cilvēku plašā aizstāšana ar mašīnām, šķiet, nav kaut kas pārspasaulīgs. Procesu automatizācija ir kļuvusi par pilnvērtīgu mūsu laika “tendenci” un turpina paplašināt savu ietekmi gandrīz visās darbības jomās: sākot ar lauksaimniecību un beidzot ar “gudrām mājām” vai mākslīgo intelektu.

Vai esat kādreiz domājuši, ka automatizācija ir visur? Viņa ir katrā mūsu kustībā. Katrā darbībā. Un katrā vēlmē. Ļoti daudzus ikdienas darbus atvieglo automatizācija – automātiska rēķinu apmaksa, veļas mašīna jūsu mājās automatizē veļas mazgāšanas procesu, televizora tālvadības pults automatizē televizora vadības procesu. Automatizāciju pielieto arī loģistikā un marketingā, bet šie nav vienīgie automatizācijas procesi, kas var atvieglot cilvēku ikdienu. Kurš gan nemīlētu automatizāciju, ja tā arī palīdz sazināties ar tuvajiem cilvēkiem un risināt dažas biznesa problēmas, tikai un vienīgi caur telefona servisiem-?

Lielākā daļa cilvēku izvēlas tālruni pie rokas nekā datoru, tāpēc īsziņas var būt tiešāks un uzticamāks paziņojumu sūtīšanas veids nekā e-pasts. Turklāt nelielais īsziņu lielums ļauj tos ātri nolasīt tiem, kuriem tie ir piešķirti. Taču veidojas jautājums, vai ir iespējams šo procesu automatizēt un padarīt daudz vieglāku? Šis jautājums ir aktuāls un tādēļ esmu izvēlēties veikt pētījumu par īsziņu sūtīšanas automatizācijas iespējām.

Darba mērķis: izziņāt automatizācijas vēsturi, kas tas ir un izveidot automatizētu īsziņu sūtīšanas procesu, izmantojot *Python* programmēšanas valodu un pielietot to biznesā resursu taupīšanai.

Hipotēze: izmantojot *Python* programmēšanas valodu ir iespējams uzrakstīt programmu, kas spēs automātiski sūtīt īsziņas vienai vai vairākām personām, atvieglot uzņēmuma rutīnas darbu, automatizēt atbildes uz lietotāju pieprasījumiem.

Pētnieciskā darba uzdevumi:

- izpētīt literatūru un interneta resursus par automatizācijas procesu skaidrojumu un pielietošanu;
- uzrakstīt reālu programmu, kas spēs automatizēt īsziņu sūtīšanu vienai vai vairākām personām;
- aprakstīt, kur programmu varēs izmantot;
- secināt, vai izmantot izveidoto programmu ir efektīvi sūtīt īsziņas.

Pētījuma bāze: pētījums tika veikts Tukuma novadā, Tukumā.

Pētījuma zinātniskais un praktiskais devums, novitāte: Zinātniskās pētniecības darbs pierāda, ka ir iespējams atvieglot darba slodzi un rutīnas darbu, ieviešot automatizācijas idejas biznesā.

1. Teorētiskā daļa

1.1. Datoru automatizāciju ideju sākumi.

Kad mēs ikdienā dzirdam vārdu “automatizācija”, mūsu prāts automātiski pāriet uz progresīvām tehnoloģijām, piemēram, mākslīgo intelektu (AI), mašīnmācīšanos un robotiku. Tomēr automatizācijas tehnoloģiju vēsture ir daudz dziļāka nekā tikai šie automatizācijas paplašinājumi, kas mūsdienās tiek izmantoti darba vietā. Automatizācija tiek definēta kā tehnoloģiju radīšana un pielietošana, lai kontrolētu un uzraudzītu dažādu preču un pakalpojumu ražošanu un piegādi. Automatizācijas ideja ne vienmēr ir mūsdienīga, jo automatizācijas tehnoloģiju izmantošanas teorija pastāv jau gadsimtiem ilgi, lai gan pēdējo 100 gadu laikā tā ir kļuvusi specifiskāka, kā arī uzlabota, lai pielāgotos noteiktām nozarēm.[6]

Vārda “automatizācija” senākās saknes meklējamas seno grieķu laikā, īpaši ap 762. gadu p.m.ē. Agrākais automatizācijas tehnoloģiju pieminējums bija Homēra eposā “Illiāda”, kurā Homērs apspriež uguns un amatniecības dievu Hefaistu. Stāsta gaitā Homērs apraksta Hefaistes darbnīcu un to, kā Hefaistam palīdzēja “automāti”, kas būtībā bija pašdarbīgi roboti, kas palīdzēja viņam izstrādāt spēcīgus ieročus un citus priekšmetus grieķu dieviem. Lai gan ir maz pierādījumu tam, ka Hefaistes darbnīca patiešām pastāvēja, šo stāstu sarakstīja īsts grieķu dzejnieks Homērs. Tas parāda, ka grieķi vismaz bija domājuši par ideju izmantot automatizācijas tehnoloģijas problēmu risināšanai, kas viņiem ļautu uzlabot ieroču un instrumentu radīšanas efektivitāti.[6]

Kopumā, kā liecina informācija, automatizācijai ir dziļa un bagāta vēsture, kas aptver gadsimtus. Galvenie automatizācijas izmantošanas veidi pirms 20. un 21. gadsimta ir bijuši rūpniecības jomā, un tikai nesen tie ir iekļauti IT pasaulē. Izmantojot rūpniecisko automatizāciju, mērķis vienmēr bija skaidrs: uzlabot dažādu priekšmetu izgatavošanas efektivitāti. Izmantojot IT automatizāciju, mērķis ir uzlabot efektivitāti, izveidojot procesu, kas ir pašpietiekams un aizstāj IT darbinieka roku darbu datu centros. Paralēles ir skaidras, un tās parāda, kāpēc automatizācija vienmēr būs izplatīta sabiedrībā. Tehnoloģiju attīstīšana, lai mazinātu cilvēku strādnieku slogu, palielinātu uzņēmējdarbības efektivitāti un atvieglotu mūsu dzīvi, samazinot roku darbu, ir tas, kas mums ir bijis svarīgs gadsimtiem ilgi un turpinās savu ietekmi arī turpmāk.

1.2. Automatizācija IT sfērā

IT automatizācija, ko dažkārt dēvē par infrastruktūras automatizāciju, ir programmatūras izmantošana, lai izveidotu atkārtojamas instrukcijas un procesus, lai aizstātu vai samazinātu cilvēku mijiedarbību ar IT sistēmām. Automatizācijas programmatūra darbojas šo instrukciju, rīku un ietvaru robežās, lai veiktu uzdevumus ar mazu vai bez cilvēka iejaukšanos. [5]

Teorētiski, ja tas ir IT uzdevums, šim uzdevumam var piemērot zināmu automatizācijas līmeni. Tātad automatizāciju var integrēt un pielietot visur, sākot no tīkla automatizācijas līdz infrastruktūrai, mākoņa nodrošināšanai (*Cloud Drive*) un standarta darbības videi (*SOE*) un tālāk, līdz lietojumprogrammu izvietošanai un konfigurācijas pārvaldībai. [3]

IT automatizācija uzņēmumos arvien vairāk kļūst par nepieciešamu procesu, kas ļauj pārvaldīt sarežģītus mūsdienu tehnoloģiskos procesus. Neatkarīgi no tā, vai automatizācija tiek izmantota biznesa īstīnās vai e-pastā, tās ietaupa laiku un samazina kļūdas, izmantojot šīs darbplūsmas. Šī iemesla dēļ automatizācija ir kļuvusi par galveno nozari vairākās nozarēs. Mārketinga speciālisti apgalvo, ka lielākie automatizācijas ieguvumi ir laika ietaupīšana (74%), palielināta klientu piesaiste (68%) un savlaicīgāka saziņa (58%). [2]

Katru dienu IT operāciju vadītāji cenšas panākt vairāk darba ar mazāk cilvēkiem. IT automatizācija piedāvā vairākas priekšrocības, kas palīdz racionalizēt IT darbības, tostarp:

- **Samazinātas izmaksas.** Atkārtojamu darbības uzdevumu automatizēšana, piemēram, lietojumprogrammu izvietošana un pakalpojumu izpilde, var palīdzēt IT operācijām ietaupīt naudu, darbojoties efektīvāk, pieļaujot mazāk kļūdu un samazinot darbinieku skaitu. [9]

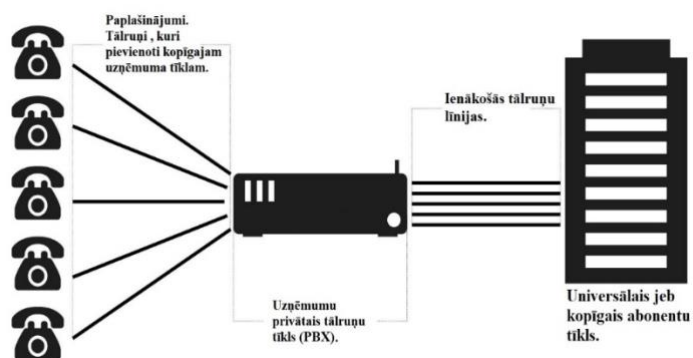
- **Paaugstināta produktivitāte.** Darbplūsmu automatizēšana novērš manuālo darbu, tostarp manuālo testēšanu, jaudas palielināšanu un darbinieku atbrīvošanu koncentrēties uz svarīgākiem projektiem. Darbinieki katru dienu var paveikt vairāk darba. [9]
- **Lielāka uzticamība.** Nogurdinošu, atkārtotu uzdevumu automatizēšana samazina dārgas kļūdas, novēršot cilvēcisko faktoru. Tas ir īpaši izdevīgi lielākos tīklos ar daudzām operētājsistēmām. [9]

Neatkarīgi no tā, vai tā ir vienkārša vai sarežģīta, automatizācija ir būtiska jebkurai komandai, kas cenšas palielināt produktivitāti. Automatizētie īsziņu sūtīšanas pakalpojumi ir īpaši prasmīgi, lai palīdzētu klientu apkalpošanas un mārketinga komandām racionalizēt savu darbību, izmantojot biznesa SMS.

1.3. Elektroniskā komunikācija un SMS ziņojumi

Līdz ar interneta parādīšanos uzņēmumi ir pārcēlušies savas sakaru sistēmas arī uz mākoņu servisiem (*Cloud Drive*). Internetā izvietotā telefonija ir aizstājusi parasto telefona aprīkojumu, piemēram, *PBX*. (skat. 1.attēlu) [1]

PBX nozīmē privāto filiāļu biržu, kas ir privāts tālrunu tīkls, ko izmanto uzņēmumā vai organizācijā. *PBX* tālrunu sistēmas lietotāji var sazināties gan iekšēji (sava uzņēmuma ietvaros), gan ārēji (ar ārpusauli), izmantojot dažādus saziņas kanālus. *PBX* ļauj arī izmantot vairāk tālrunu nekā fiziskās tālruna līnijas (*PSTN*), kā arī ļauj bez maksas zvanīt starp lietotājiem. [4]



1. attēls. Privāto tālrunu tīkls (*PBX*)

Tas ir pavēris mākoņtelefonijas (*Cloud*

telephony) izmantošanas iespēju pasauli biznesa vajadzību risināšanai, un arī *Python* programmēšanas valoda ir tajā iesaistīta. Mākoņtelefonijas izmantošana uzņēmējdarbībai ļauj vienlaikus veikt un saņemt vairākus zvanus un īsziņas. Pakalpojumi, piemēram, zvanu pārsūtīšana, ierakstīšana, lielapjoma īsziņu sūtīšana, ir dažas no lieliskajām funkcijām, kuras var izmantot mākoņtelefoniju servisā. Mākoņtelefonija nodrošina biznesa vajadzību pārvaldību, neapdraudot kvalitāti, izmaksas un bez nepieciešamības ieguldīt papildu infrastruktūrā. [1]

Elektroniskā komunikācija ir galvenais tehnoloģiju iespējotājs. Pārvadājumos visizplatītākā izmantotā forma ir īsziņu pakalpojums (SMS), ko parasti izmanto, lai paziņotu darba statusu, piemēram, kad vadītājs ir piegādājis pasūtījumu. Tehnoloģija jau ir gatava, lai automātiski apstrādātu īsziņas un ievadītu datus maršruta plānotājā. [1]

Lai strādātu ar mākoņtelefoniju pakalpojumiem ir jāreģistrējas pie telefonijas pakalpojumu sniedzējiem.

1.4. Telefonijas pakalpojumu sniedzēji.

Lai strādātu ar mākoņtelefonijas pakalpojumiem ir jāreģistrē savs konts pie telefonijas pakalpojumu sniedzēja. Ir pāris populāri šo servisu nodrošinātāji, kurus var atrast, veicot meklēšanu interneta vietnēs. Praktiskajā daļā autora izvēle apstājās uz *Twilio* pakalpojumiem.

1.5. Twilio

Pastāv vecs tīkls, kurš ir vecāks pat par internetu. Tas, kas realizē balss un SMS saziņu komunikāciju tieši tā, kā mēs to zinām. Tas ir telefonijas tīkls. Lai gan tas ir kritiski svarīgs

savienojamībai visā pasaulē, šis tīkls darbojas samērā noslēgtā, lēni kustīgā un dārgā ekosistēmā. [7]

Telefonijas tīklu savieno mobilo sakaru operatori. Pārvadātāji ir atbildīgi par balss, teksta un datu plūsmu visā pasaulē. Sakaru operatori pārdod savienojumus savos tīklos.

Jums un darba autoram šis savienojums ir vienkāršs: mobilais tālrunis vai mājas interneta savienojums. Patērētāji ir paslēpti no sarežģītās valodu, protokolu un aparatūras pasaules, kas nodrošina globālā telefonu tīkla pastāvēšanu. [4]

Twilio koncentrējās uz to, ka internetā nav gandrīz tādu pašu sarežģītību kā telefonijas tīklā. Bet bija milzīgs šķērslis: internets nebija savienots ar telefonijas tīklu. [7]

Twilio redz savu vietu starp šiem diviem tīkliem. Tas veic smago darbu, nodrošinot tiltu starp mobilajiem operatoriem un internetu. Izstrādātāji var rakstīt programmatūru, kuru var izvietot tur, kur tā ir lēta un viegli iemācāma. *Twilio* var dot iespēju šai programmatūrai sadarboties ar augsti specializēto sakaru operatoru ekosistēmu. Bet *Twilio* faktiski nedzīvo plaisā starp internetu un telekomunikācijām, jo visu savu infrastruktūru izvieto internetā. Izmantojot īpašus savienojumus ar pārvadātājiem, tas spēj savienot interneta un telefonijas pasauli. (skat. 2.attēlu) [7]



2. attēls. *Twilio* darba princips (twilio.com)

Twilio ir mākoņu sakaru platforma, kura prezentējas kā pakalpojums. Tas ļauj programmatūras izstrādātājiem programmatiski veikt un saņemt tālruņa zvanus, sūtīt un saņemt īsziņas un veikt citas komunikācijas funkcijas, izmantojot sava tīmekļa pakalpojumu iespējas. Produkts, ko sniedz *Twilio* ļauj programmētājiem vieglāk integrēt dažādas saziņas metodes savā programmatūrā un programmās. Īpaši labvēlīgi šis serviss attiecas pret *Python* programmēšanas valodu.

1.6. *Python* valoda

Python tika izgudrots 80. gadu beigās. Tā ieviešanu 1989. gada beigās sāka Gvido van Rosums Nīderlandes *Centrum Wiskunde & Informatica* (Valsts matemātikas un datorzinātņu pētniecības institūts). *Python* ir ABC programmēšanas valodas pēctecis, kuru pašu iedvesmoja *SETL*. 1991. gada februārī Van Rossums publicēja *Python* kodu uz *alt.sources* ziņu grupas. *Python* nosaukumu iedvesmoja televīzijas šovs "*Monty Python's Flying Circus*", jo Van Rossum ir šī šova liels fans. Van Rossums vēl joprojām skaitās kā galvenais *Python* autors. Viņam joprojām ir galvenā loma, virzot *Python* attīstības virzienu un turpmāko attīstību. [11]

Cilvēkiem tiek maksāta nauda, lai rakstītu *Python* programmas, kuras mēs izmantojam katru dienu - *Google*, *YouTube*, *Dropbox*, *Netflix* un *Hulu*. Autors izmantoja *Python*, lai izveidotu automātisko SMS ziņojumu sūtīšanas programmu. *Python* ir slavena kā augstas veiktspējas programmēšanas valoda, kas ir populāra dinamiski attīstītās organizācijās. [10]

Programmas, kas rakstītas *Python* programmēšanas valodā, var būt vai nu vienreizēji skripti, vai sarežģītas sistēmas, kas satur miljoniem kodu rindas. Lai vieglāk paveiktu kodu rindiņu rakstīšanas uzdevumu tiek piedāvāti dažādi integrētās izstrādes vides – *IDE*.

2. Praktiskā daļa


Autora zinātniskās pētniecības darba praktiskās daļas mērķis ir noskaidrot, vai ir iespējams automatizēt īsziņu sūtīšanas procesu. Lai būtu iespējams sasniegt mērķi un atbildēt uz jautājumu – vai ir iespējams *Python* programmēšanas valodā uzrakstīt programmu, kas spēs automātiski sūtīt īsziņas, tika izvirzīti uzdevumi:

1. izvēlēties telefonijas pakalpojumu sniedzēju;
2. izmantojot servisa piedāvātās iespējas, uzrakstīt programmas kodu;
3. izpētīt programmas darbošanās principu;
4. pielietot to kādā uzdevumā.

2.1. Telefonijas pakalpojumu sniedzēja izvēlēšanās

Lai strādātu ar mākoņdatības telefonijas pakalpojumiem, autors reģistrēja kontu pie telefonijas pakalpojumu sniedzēja – *Twilio*. Pēc konta izveides, bija jāverificē tālruņa numurs uz kura tika sūtītas īsziņas. (skat. 4.attēlu)

Verified Caller IDs

	<input type="text" value="Number"/>	<input type="text" value="Friendly Name"/>	<input type="button" value="Filter"/>
	NUMBER	FRIENDLY NAME	
	+371 27 722 338	Ilja Polozvs. ZPD	




4. attēls. Verificēts telefona numurs (Darba autors)

Šī verificēšanās ir nepieciešama, lai neļautu cilvēkiem izmantot pakalpojumu, lai nejauši vai speciāli nosūtītu nejausus tālruņu numurus ar liela apjoma īsziņām.

Kā minēts iepriekš, lai strādātu ar *Twilio API* ir jāreģistrē konts. *Twilio* izveido kontu un nodrošina unikālu *AccountSID* un *AuthToken*, kas apstiprinās autora pieprasījumus un nodrošinās, lai autora kontā tiktu iekasēti rēķini par veiktajiem *API* zvaniem.

Twilio tālruņa numuri tiek izmantoti kā zvanītāja *ID*, lai nosūtītu īsziņas vai balss zvanus. Zvanītāja *ID* (pazīstams arī kā izsaukēja līnijas identifikācija) ir zvanītāja numurs, kas mirgo uz izsauktās personas aprīkojuma (fiksētā vai mobilā). Šajā gadījumā numurs, kuru autors noīrēja no *Twilio*, tiks izmantots kā zvanītāja *ID*. (skat. 5.attēlu)

Active Numbers

Inventory Filters		Configuration Filters			
Number	<input type="text"/>	Voice URL	<input type="text"/>	<input type="button" value="Filter"/>	<input type="button" value="Reset filters"/>
Number	Friendly Name	Capabilities	Configuration		
+1 833 332 3648 TOLL-FREE	Zvanītāja ID	  	Voice Webhook (POST): https://demo.twilio.com/welcome/voice/ Messaging Webhook (POST): https://demo.twilio.com/welcome/sms/reply		

5. attēls. Noīrētais telefona numurs jeb zvanītāja ID (Darba autors)

2.2. Vides sagatavošana SMS sūtīšanai

Īsziņu sūtīšanai autors izvēlējās *Twilio API*, bet pirms tam tika izveidota virtuāla *Python* vide – *PyCharm*. Tika instalēti *flask* un *twilio Python* moduļi, lai izveidotu vienkāršu tīmekļa pakalpojumu, kuru izsauks telefonijas pakalpojumu sniedzējs *Twilio*. Pēc tam *flask* modulis veiks nepieciešamo uzņēmējdarbību uz *Twilio* atzvanīšanu. (skat. 6.attēlu)

```
C:\Windows\system32>pip install flask
Collecting flask
  Using cached Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Requirement already satisfied: itsdangerous>=0.24 in c:\python37\lib\site-packages (from flask) (1.1.0)
Requirement already satisfied: Werkzeug>=0.15 in c:\python37\lib\site-packages (from flask) (1.0.1)
Requirement already satisfied: Jinja2>=2.10.1 in c:\python37\lib\site-packages (from flask) (2.11.2)
Requirement already satisfied: click>=5.1 in c:\python37\lib\site-packages (from flask) (7.1.2)
Requirement already satisfied: MarkupSafe>=0.23 in c:\python37\lib\site-packages (from Jinja2>=2.10.1->flask) (1.1.1)
Installing collected packages: flask
Successfully installed flask-1.1.2

C:\Windows\system32>pip install twilio
Collecting twilio
  Downloading twilio-6.51.0.tar.gz (457 kB)
    |#####| 457 kB 6.8 MB/s
Requirement already satisfied: six in c:\python37\lib\site-packages (from twilio) (1.15.0)
Requirement already satisfied: pytz in c:\python37\lib\site-packages (from twilio) (2020.4)
Requirement already satisfied: PyJWT==1.7.1 in c:\python37\lib\site-packages (from twilio) (1.7.1)
Requirement already satisfied: requests>=2.0.0 in c:\python37\lib\site-packages (from twilio) (2.24.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26.0,>=1.21.1 in c:\python37\lib\site-packages (from requests>=2.0.0->twilio) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in c:\python37\lib\site-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\python37\lib\site-packages (from requests>=2.0.0->twilio) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\python37\lib\site-packages (from requests>=2.0.0->twilio) (3.0.4)
Building wheels for collected packages: twilio
  Building wheel for twilio (setup.py) ... done
  Created wheel for twilio: filename=twilio-6.51.0-py2.py3-none-any.whl size=1209292 sha256=1d8fb0314181f524442b4c40b99d4c43be711c94a683ad29c38db951f88fa2
  Stored in directory: c:\users\lilja\appdata\local\pip\cache\wheels\56\83\94\4f45f04b6db724ad6e1d6c917fa46cb33cbc5b5a6688a05946
Successfully built twilio
Installing collected packages: twilio
Successfully installed twilio-6.51.0

C:\Windows\system32>
```

6. attēls. Flask un Twilio moduļu instalēšanas process (Darba autors)

2.3.SMS sūtīšana

Autors sāka konfigurācijas izveidošanu, kurā tika ierakstīti *Twilio AccountSID* un *AuthToken*. Tika arī pievienots autora nomātais telefona numurs kā zvanītāja *ID* un pielāgotais numurs, kuram ziņa tika nosūtīta.

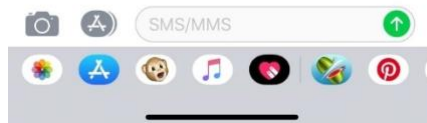
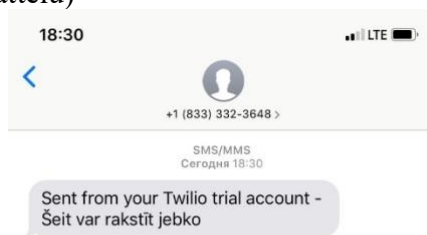
```
1 from twilio.rest import Client
2
3 accountsID = 'AC2e8cc25b852f94e71387637ee59fa4d7' #Šeit lietotājs ievada savu accountSID no Twilio.
4 authToken = '5d5e8bff62ed319c8ffe87ae94e9644d' #Šeit lietotājs ievada savu authToken no Twilio.
5 mans_numurs = '+37127722338' #lietotājs ievada savu telefona numuru, kurš ir verificēts.
6 nomatais_numurs = '+18333323648' #lietotājs ievada nomāto numuru, kuru piedāvā Twilio.
7
8 client = Client(accountsID, authToken) #pieprasa un pārbauda accountSID un authToken.
9
10 client.messages.create(_išziņas_veidošanas_forma,
11     to=mans_numurs, #sūta uz ievadīto numuru,
12     from_=nomatais_numurs, #sūta no nomātā numura,
13     body='Šeit var rakstīt jebko') #išziņas teksts.
14
```

7. attēls. Kods, kurš ļauj sūtīt īsziņas uz verificētiem numuriem (Darba autors)

Pirmkārt, autors izveidoja konfigurācijas failu un aizpildīja to ar visiem nepieciešamajiem laukiem. Failā *text.py* tika importēti arī nepieciešamie moduļi, tas ir, *Python Twilio* moduli. Pēc tam autors izveidoja objektu jeb *client*, izmantojot klasi *TwilioRestClient*, kas tika importēta no *twilio.rest* moduļa. Objekts tiek izveidots ar *AccountSID* un *AuthToken*, kurus ievada lietotājs balstoties, kādas vērtības *Twilio* izsniegs kontam. Pēc tam tika izmantota *create()* metode no *client.messages* ziņojumu klases. Koda rindās ir numurs, uz kuru tiek nosūtīts ziņojums (*to*), *from_* ir zvanītāja *ID*, un *body* ir teksts, kas jānosūta personai. (skat. 7.attēlu)

Autors nosūtīja īsziņu ar tekstu: “Šeit var rakstīt jebko” uz telefona numuru, kuru verificēja iepriekš. SMS uz telefona numuru tika nosūtīts gandrīz uzreiz pēc koda izpildīšanas,

tas ir, 5 sekundes. Atkarībā no mobilo sakaru operatora un tīkla pārslodzes ziņojuma saņemšana var ilgt 2–3 minūtes. (skat. 8.attēlu)



8. attēls. Telefona SMS lietotne, kurā tika saņemta īsziņa (Darba autors)

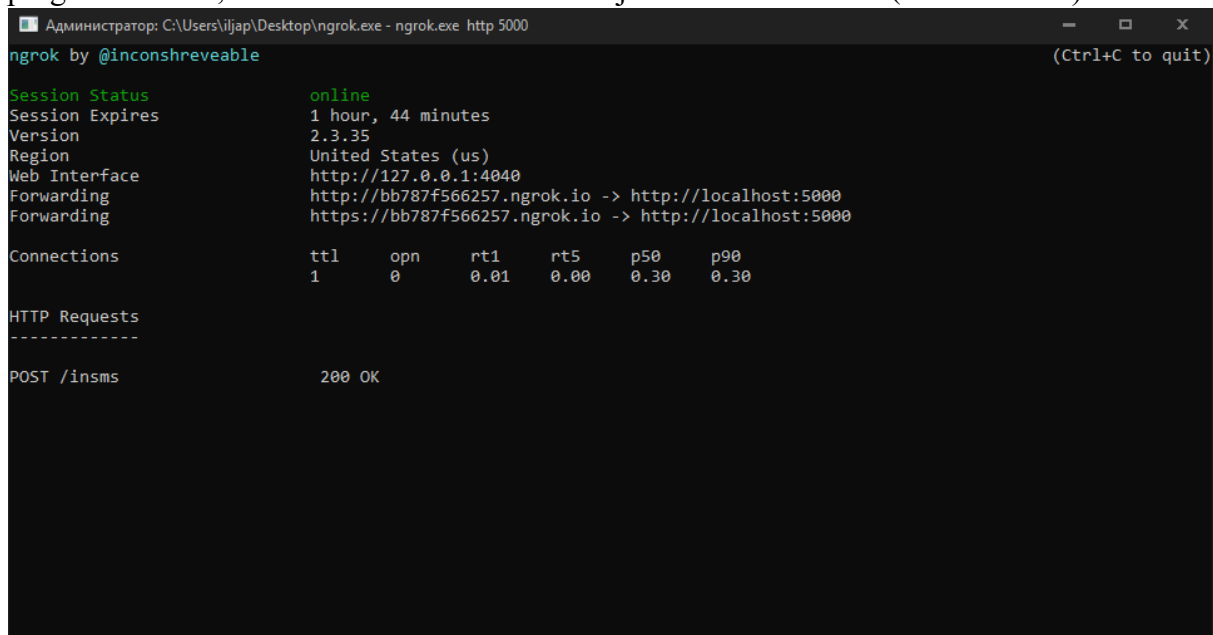
2.4. SMS ziņojumu automatizētā atbilde

Pēc pamatprogrammas uzrakstīšanas tika izvirzīts mērķis paplašināt tās funkcionalitāti ar ne tikai vieglu SMS ziņojumu sūtīšanu, bet arī ar automātisko atbildi uz lietotāja pieprasījumiem, caur SMS ziņojumiem.

Katru reizi, kad *Twilio* saņem īsziņu uz reģistrēto (nomāto) tālruņa numuru, tas izpilda pieprasījumu uz publiski pieejamu *URL* (Vienotais resursu vietrādis). Tas ir konfigurēts pakalpojumā *Twilio*, tas nozīmē, ka kodam, kas jāizpilda, piekļūstot *URL*, jābūt autora kontrolē. Tas radīja problēmu, ka šis *URL* ir pieejams internetā. Tas nozīmēja, ka tikai autora vietējais dators nedarbosies, jo bija maz ticams, ka tas ir adresējams ārēji no vietējā tīkla. Lai sniegtu pieejamu pakalpojumu, autors izvēlējās alternatīvu, izmantojot “*ngrok*”. Tas ļāva izveidot tā saucamo tuneli starp publisko adresi un vietējo izstrādes vidi.

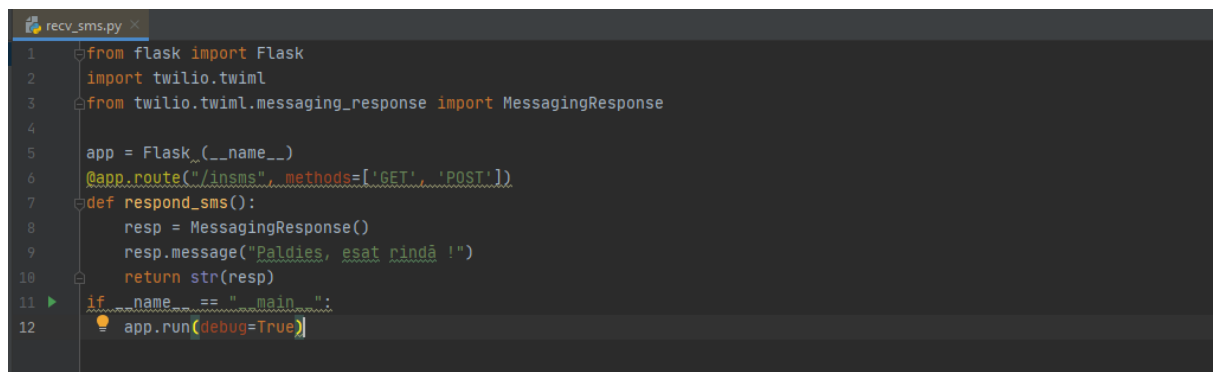
Pēc *ngrok* programmatūras instalēšanas, tika izveidots tunelis starp autora lokālo virtuālo mašīnu un internetu. Tas sniedza iespēju atvērt lokālo serveri aiz *NAT* (Tīkla adresu translēšana)

un *firewall* (ugunsmūris). Serveris tika atvērts uz 5000 porta, kuru visbiežāk izmanto programmatūrās, kuras kaut kādā veidā sadarbojas ar interneta tīklu. (skat. 9. attēlu)



9. attēls. ngrok veidotais serveris (Darba autors)

Tā kā serveris tika bez problēmām atvērts, autors izveidoja konfigurācijas failu, kas ļautu lokālajam serverim saņemt un uztvert lietotāja sūtītās īsziņas. (skat. 10.attēlu)

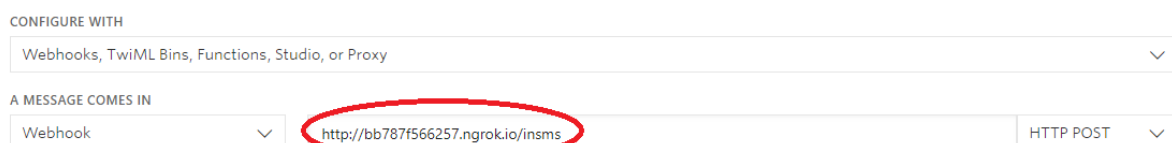


10. attēls. Programmas kods, kas ļauj serverim uztvert īsziņas un atbildēt uz tām ar noteikto tekstu (Darba autors)

Serveris, kas pieņem ienākošo ziņojumu, tiek ierakstīts *flask* modulī un darbojas portā 5000. Tas liek serverim darboties vietējā mašīnā. Lai padarītu to pieejamu internetā, autors aktivizēja *ngrok* serveri un lieka tam darboties tajā pašā portā kā *flask* moduļa serveris, tas ir, 5000 ports.

Autors konfigurēja *Twilio* tālruņa numuru (nomātais telefona numurs), lai ienākošos ziņojumus novirzītu uz *flask* serveri. Lai to panāktu, *Twilio* tika pievienots pieprasījuma URL. (skat.11.attēlu)

Messaging



11. attēls. ngrok lokālā servera URL (Darba autors)

Tātad, ikreiz, kad uz īrēto numuru tiek nosūtīts ziņojums, tas caur *ngrok* tiek novirzīts uz autora *flask* serveri. *Flask* serverī jau ir izveidots maršruts, kas konfigurēts ar *URL/insms*. Šis maršruts saņem *POST* zvanu no *Twilio* servera (skat. 6.rindu, 10.attēlu), kas pēc tam atbild ar ziņojumu - Paldies, esat rindā !



12. attēls. SMS atbilde uz lietotāja aizsūtīto īsziņu (Darba autors)

2.5.SMS ziņojumu automātisko atbilžu paplašinājums un pielietojums

Meklējot interneta vietnēs uzrakstītās programmas kodu pielietošanas iespējas, autors pamanīja, ka daudzi uzņēmēji uzdod vienu un to pašu jautājumu – kā paspēt atbildēt visiem lietotājiem ?

Autors, lai eksperiments liktos reālāks, nāca klajā ar reālistisku uzdevuma izveidi, kuram būtu iespējams piemērot iepriekš uzrakstītu pamatprogrammu. Tika izvirzīts šāds uzdevums: Jānis, “Gudrā Pica” tirdzniecības vietas īpašnieks Latvijā, meklē veidus, kā uzlabot picu pārdošanas apjomus. Viens no veidiem, kā uzlabot pārdošanas apjomus, būtu koncentrēties uz to, lai atvieglotu picu pasūtīšanas procesu. Viņš arī vēlas automatizēt darba plūsmas, lai klientus varētu informēt par pasūtījumu un piegādes statusu. Jānis arī uzskata, ka, lai arī internets ir lielisks, klienti varētu vēlēt pasūtīt picas arī atrodoties vietās, kur tīkla uztveršana ir vāja.

Tika izveidoti trīs mērķi kā uzlabot programmas kodu, lai veiksmīgi paveiktu uzdevuma izpildi:

1. spēja saņemt ienākošos ziņojumus;
2. uzturēt un pieprasīt pasūtījuma statusu;
3. nosūtīt izejošā statusa ziņojumu.

Lai paveiktu šo uzdevumu, autors paplašināja automātisko atbilžu pamatprogrammas kodu. Uzrakstītā koda fragments ir sadalīts trīs galvenajos aspektos: spēja saņemt ienākošos ziņojumus ar *flask* maršrutu, pasūtījuma statusa saglabāšana klientiem, kuri vaicā pēc viņu pasūtījumiem, un iespēja nosūtīt izejošos ziņojumus no lietotnes.

```

1 from flask import Flask, request
2 import twilio.twiml
3 from twilio.twiml.messaging_response import MessagingResponse
4 import requests
5
6 class Pizza:
7     def __init__(self):
8         self.status = None
9
10    def setStatus(self, status):
11        self.status = status
12
13    def getStatus(self):
14        return self.status
15
16 app = Flask(__name__)
17 @app.route("/sms", methods=['GET', 'POST'])
18 def respond_sms():
19     content = requests.post('http://98589d5cf87d.ngrok.io', data={'key': 'value'})
20     resp = MessagingResponse()
21     if content == "Pica":
22         resp.message("Paldies ! Jau gatavojam pasūtījumu!")
23         pizza = Pizza()
24         pizza.setStatus('complete')
25         return str(resp)
26     if content == "Statuss":
27         pizza = Pizza()
28         status = pizza.getStatus()
29         if status == 'complete':
30             resp.message("Jūsu pasūtījums ir gatavs")
31             return str(resp)
32         else:
33             resp.message("Atvainojiet ! Nevaram atrast jūsu pasūtījumu!")
34             return str(resp)
35     else:
36         resp.message("Nepareizā funkcija ! Lai pasūtītu picu, atsūdiet - Pica")
37         return str(resp)
38
39 if __name__ == '__main__':
40     app.run(debug=True)

```

Autoram, līdz šim momentam, jau bija uzrakstīta lietotne priekš ienākošo ziņojumu saņemšanai. Tātad, šī lietotne tika paplašināta un izmantota savām vajadzībām. Autora nomātais numurs, šajā uzdevuma ietvaros, kļūst par “Gudrās Picas” numuru.

- Lietošanas gadījumā tika nolemts izmantot divus atslēgvārdus “Pica” un “Statuss”, kurus “Gudrās Picas” klienti var izmantot.
- Kad klienti nosūta ziņojumu “Pica”, tie var veikt pasūtījumu pēc izvēlētas picas. Picu veikals pozitīvi reaģē uz pasūtījumu, atbildot, ka viņi jau strādā pie pasūtījuma.
- Kad klients vēlas uzzināt sava pasūtījuma statusu, viņi to var pārbaudīt ar tekstu – “Statuss”. Autora gadījumā, kad klients pieprasa savu pasūtījumu, viņš saņem picu veikala atbildi, sakot, ka viņa pasūtījums ir gatavs.



14. attēls. Automātiskās atbildes no nomātā numura (Darba autors)

Secinājumi

Darba tēma ir aktuāla un svarīga, jo tehnoloģijas attīstās ļoti ātri, taču mazie uzņēmumi joprojām izmanto cilvēku darbu, kā rezultātā notiek cilvēku spēku pārslodze un palēninās darba process. Aprakstītā automatizācija autora darbā atrisina šo problēmu.

Darbā izvirzītais mērķis tika sasniegts. Pētījuma hipotēze apstiprinājās. Tika uzrakstīts programmas kods, kas var automatizēt ikdienas atbildes uz lietotāju pieprasījumiem caur SMS ziņojumiem. Tas, samazināja slogu cilvēkiem, kuri iepriekš veica šo darbu pilnīgi manuāli. Tomēr, lai sasniegtu šo rezultātu, būs jāpieņem lēmums par profesionālā programmētāja pakalpojumu apmaksāšanu, kas nodrošinātu iespēju modificēt programmu konkrētajam biznesam.

Pāreja uz automatizētu darbu ir daudz rentablāka nekā mēneša budžeta tērēšana mobilo sakaru operatoru pakalpojumu apmaksai, kas ir 11,99 EUR. Ikmēneša maksa par nomāto telefona numuru izmaksās 3 reizes lētāk.

Autors, veicot zinātniskās pētniecības darbu, sastapās ar dažādām problēmām, kuras bija grūti risināmas, piemēram, saistībā ar autora interneta pakalpojumu sniedzēja noteikumiem, tika bloķēts 5000. ports. Problēmas risināšanai tika veltīts pārāk daudz laika. Darbā šīs problēmas netiek minētas, jo tās neko neietekmēja un nekāds lēmums, balstoties uz tām, netika pieņemts.

Pētnieciskā uzdevuma laikā autors paplašināja savas zināšanas telekomunikācijas sfērā un *Python* programmēšanas valodas darba aspektos, lai veiksmīgi uzrakstītu programmas kodu. Veiktais darbs spēs automatizēt dažu biznesu galveno problēmu – tikt galā ar lietotāju pieplūdumu, izmantojot automātiskās atbildes uz lietotāju pieprasījumiem. Autors pēc šī darba plāno attīstīt izveidoto lietotni, palielinot tās efektivitāti, kuras pamatā ir atvieglot cilvēku rutīnas darbu un palielināt komunikācijas iespējas starp lietotāju un pakalpojuma sniedzēju.

Izmantotās literatūras un interneta resursu avoti

1. Brown K. *IP Telephony Unveiled*. USA: Cisco Press, 2004.
2. Howie L., Debra C., *IT Automation: The Quest for Lights Out*. USA: Prentice Hall PTR, 2000.
3. Love J. *Process Automation Handbook: A Guide to Theory and Practice*. London: Springer Ltd., 2007
4. Martelli A. *Python in a Nutshell: A Desktop Quick Reference*. USA: O'Reilly Media, 2006.
5. Sahoo R., Sahoo G. *Computer Science with Python*. India: New Saraswati House Ltd., 2016.
6. Taulli T. *The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems*. Monrovia: Apress, 2020.
7. Twilio. *What is Twilio ?*
<https://www.twilio.com/blog/what-does-twilio-do>
8. Wielenga G. *Beginning NetBeans IDE: For Java Developers*. USA: Apress, 2015.
9. Любанович Б. *Простой Python. Современный стиль программирования*. СПб: Питер, 2017.
10. Прохоренок Н. *Python*. СПб: БХВ-Петербург, 2011.
11. Рейтц К., Шлюссер Т. *Автостоном по Python*. СПб: Питер, 2017.