

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 5.015	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2018 Issue: 12 Volume: 68

Published: 27.12.2018 <http://T-Science.org>

QR – Issue



QR – Article



Vadim Andreevich Kozhevnikov

Senior Lecturer

Peter the Great St. Petersburg Polytechnic University

vadim.kozhevnikov@gmail.com

Evgeniya Sergeevna Pankratova

student

Peter the Great St. Petersburg Polytechnic University

jane_koks@mail.ru

RESEARCH OF THE CUSTOMER SUPPORT SERVICE DEVELOPMENT FOR USER APPLICATIONS

Abstract: The article includes information about chat-bots, support mechanism - how to make the interaction of users and support department useful for both sides. It also includes the description of subject area, service's components and their purpose and steps of bot working process.

Key words: Telegram, chat-bot, helpdesk, GLPI.

Language: English

Citation: Kozhevnikov, V. A., & Pankratova, E. S. (2018). Research of the customer support service development for user applications. *ISJ Theoretical & Applied Science*, 12 (68), 271-276.

Soi: <http://s-o-i.org/1.1/TAS-12-68-39> **Doi:**  <https://dx.doi.org/10.15863/TAS.2018.12.68.39>

Introduction

It is difficult to imagine the modern world without a smartphone. We use for completely different purposes: listening to music, reading electronic books, looking for a route from one place to another, making calls, writing messages to friends and relatives.

According to statistics, every day people use from one to four mobile applications, and one of these applications is an instant messenger [1, 2].

Messengers now are not only a way to transfer messages between users, but they also perform the information part, there are a large number of different channels where admins (the main person of channel who has access to this channel) post daily about news, sports, music, so they are like blogs, but they are not on special platforms for blogging, they are in messengers.

And messengers also have bots - some kind of program which is fully automatic and can do a lot of things, it is an assistant like Siri or Alexa. For example, bot can send you everyday jokes or fun pictures, music or news, or may be if you want to receive weather announcements it can do it too [3]. And also they can be used for getting information from users for example support bots. It means that

there is a special bot and if you have a problem with some site or application and its development company has such mechanism like this in messenger, you can send a message to bot and it will answer you sooner or later [4].

Since July 2018, one of us have been an intern at a software development company whose knowledge is applied in the field of medicine, where the company has extensive experience in successful projects.

At the beginning of work one of us was offered to develop its own project independently. Every good user application should have a support mechanism to help developers (or company itself) get information about possible errors while users exploring this app. The problem is how to do it right, how to organize this interaction between support and users?

We think, that letters to the mail for support are not very effective [5]. Because a lot of people write about their problems to support every day and they write not just one letter, but they can really spam when trying to know why there is still no answer. So because of this, we guess it is possible to lost some letters, also they can be removed, or for example user write two letters and one is removed and support will never understand what was the problem, so they will never answer to this user.

Impact Factor:

ISRA (India)	= 3.117	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	PIHHI (Russia)	= 0.156	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 5.015	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667		

So the project that was offered for the internship is precisely engaged in providing user support - this is the built-in chat bot [6, 7] in the instant messenger platform that collects problems (or tickets) from users and create them in a special helpdesk system where employees of support department can see each tickets and not lose it (because helpdesk is a more convenient interface than regular mail) and user will finally receive feedback even if his problem is not solved.

The aim of the article

Since one of us already work in a company and has its own project (and this project is a new product for my company), my tasks include not only development of chat bot, but also tasks of analytics, planning and other commercial development processes. Therefore, the aim of this work is to get acquainted with the processes of analytics and planning.

To achieve this goal the following tasks were set: plan the components of the application under development, create a component scheme on which the main application services will be displayed, create sequence diagrams which will help other people (not developers) to understand the processes of app's work, although it will be useful other developer, which will support this app after our development. And the main tasks of this article is to know what helpdesk system is, outline the benefits of these systems and make the review of the main interaction with helpdesk methods.

Formulation of the problem

Need to develop the chat-bot that will communicate with the users, help the create a request (or speaking of helpdesk system terms - tickets) for the existing problem. After the user confirms the creation of the ticket, all the data (messages, creation time) will be transferred to the helpdesk system for further processing by the support service staff. After the request is resolved and a response (or solution) is left for it, a corresponding message will be sent to the user.

The subject area description

The whole project consists of several parts: the platform (messenger platform, web site, plugin and so on) on which the program will run, the bot program, the database, which is the data repository and the server on which the code is laid out, and most importantly the helpdesk system. We give a more detailed definition of each of the components.

For current development the Telegram messenger was chosen. Telegram is a cross-platform instant messenger that allows you to exchange

messages and media files of many formats. Since the company still has two priority messengers for implementation - Facebook Messenger and Telegram, and for Telegram it was proposed to develop the first version of the product, he was chosen. But we would like to note that you can embed the program in any other environment.

A bot is a program that tries to create the impression that it is not a program, but a person; a program that automatically or on a predetermined schedule performs any actions intended for people. The functions of the bot include communication with the user, working with the database, sending messages to the application service, which we will also develop, sending messages to users about the decision of their applications.

A database (DB) is an organized structure for storing, modifying, and processing related information. In the described work there will be several databases, so the project architecture is assumed to be microservice, and thus each service should have its own database. One of the databases will be designed to store information about users, channels that can work with the bot, support services. The other is for storing information about messages compiled by users and other related information.

The server on which the bot and other services will be laid out that will be needed to ensure its operation is provided by the company.

And helpdesk-system itself - a system for handling applications and incidents. For this work, the choice of tool was behind the company and it was proposed to use the GLPI system.

The description of the app architecture

The project being developed is a microservice architecture. Its plus is that if in traditional versions of a service-oriented architecture modules can be quite complex software systems themselves, and the interaction between them often relies on standardized heavyweight protocols, in the microservice architecture of the system they are built from components that perform relatively elementary functions, and interoperable using cost-effective network communication protocols.

Thus, the project will consist of individual microservices. Consider them in more detail. First, it is necessary to identify the main components for the project. According to the collected requirements and own reasoning, these will be the following parts:

- Support Bot Service is the most important microservice that is responsible for working with the bot, includes all the basic logic for communicating with the user;
- Account Service - a component that is responsible for working with user accounts, it provides an interface for working with the

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 5.015	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	

database. In fact, it will not be a microservice in this context, as it will be built into the Support Bot Service, but it could well become one. To do this, it would be necessary to develop it as a separate service that would provide its own API for interaction with it;

- Ticket Service is a component that will work with a database that stores all information about tickets and works with helpdesk-system. Unlike the Account Service, it will already be a microservice and will provide an API that the Support Bot Service can use.

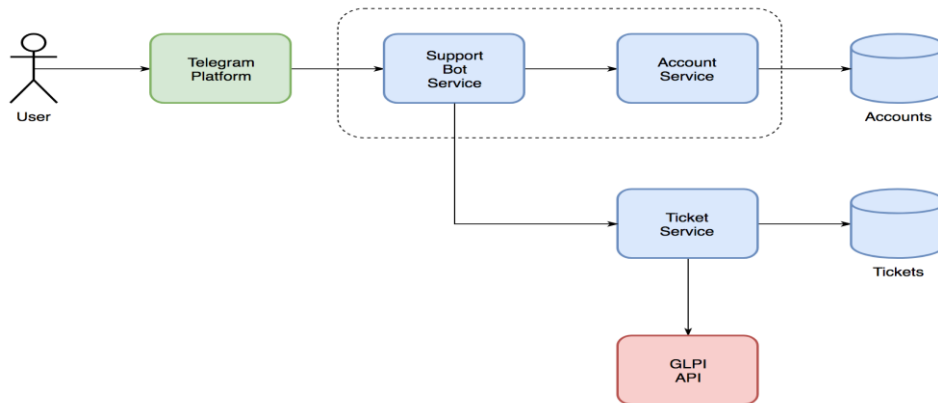


Fig.1 - Component scheme

In Fig. 1, you can see the component scheme, which consists of the components described above - Support Bot Service, Ticket Service, Account Service. In addition, the diagram shows the Accounts and Tickets databases, each base for its component. You can also see a branch from the Ticket Service - GLPI API - an API provided by the tickets processing service. Dashed line combined services, which are one application (one service).

After we have created the component diagram, it is necessary to consider the processes that the entire project will implement - creating user accounts, creating tickets locally in our database (collecting user messages), creating tickets in the helpdesk system and receiving answers from the support service. For such purposes, you can create a sequence diagram (see Fig. 2), which will visually describe the processes occurring in the service.

Few words about helpdesks

The bot, which was proposed for development, is committed to providing users with an interface to communicate with customer support. He communicates with it indirectly, as well as through a special tool called HelpDesk-system designed to automate the processing of customer requests.

When a request comes from a client, a "ticket" is created in the system. This ticket, depending on its content and importance, is placed in a queue to one of the support staff and he is already working with the client to solve the problem. During this process,

the status of the ticket is updated and the head of the support service can monitor how the support staff cope with the load.

Implementing HelpDesk is beneficial for everyone - business owners, company managers, IT departments and end users. Such services eliminate the problems associated with the slow response of IT services to requests. The quality of services of organizations that have several offices and divisions in different regions also increases. Accordingly, the security of the IT infrastructure of companies is improved, costs are reduced, and it becomes possible to predict them.

For users, the implementation of HelpDesk is characterized by an increase in the level of service provided and a decrease in the periods of stops due to problems in the IT infrastructure. And for the IT services themselves, the introduction of such an organization of work becomes a solution to the problem when a business cannot understand the requests of IT people. It is possible to simply justify investments in IT and plan the budget according to the needs in the development of business processes. In addition, the IT department can provide up-to-date and reliable data about its work and increase the availability of its services to users.

GLPI description

In company, as probably in any other, there is a support department or technical support department. Employees of the service and was invited to use GLPI.

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 5.015	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	

GLPI (Gestionnaire libre de parc informatique - free IT infrastructure manager) - is a system for handling requests and incidents, as well as for inventory of computer equipment (computers, software, printers, etc.). It has advanced features,

such as notification mail tracking and methods, to create a database with basic information about network topology [8].

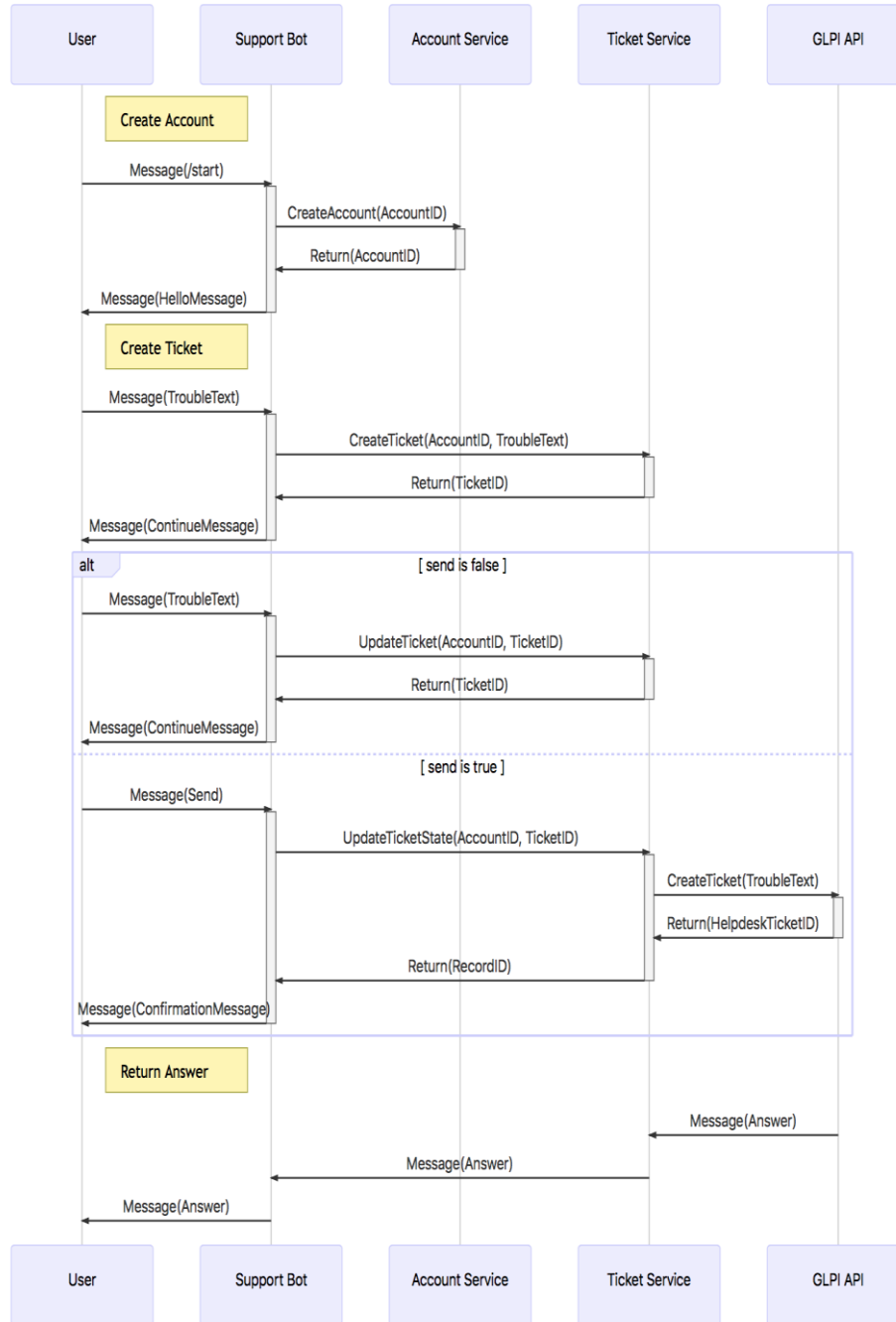


Fig.2 - Sequence diagram

GLPI is software tool for managing IT services for free and open use that makes it easy to plan, manage, and easily solve problems when they occur.

In addition, this system allows you to easily control the company's budget and its costs, if necessary.

Impact Factor:

ISRA (India)	= 3.117	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИИ (Russia)	= 0.156	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 5.015	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667		

Description of the mechanism of interaction with GLPI

Before starting development, it is necessary to study the capabilities of the chosen system for working with applications, see its API, select methods and consider their use for our task.

Since the work on this project has been started as part of internship, one of us was faced with not only development tasks, but also analytics and planning. Each service we develop has a large number of diagrams, sequence diagrams so that the manager/customer has an idea of what kind of product it is, what functions it performs, what processes occur in it.

Since the one of the tasks of this research paper is devoted to acquaintance with the ticket processing system, the relevant components will be considered in this section. Previously the basic architecture of the project was described. Based on this information, we can build a sequence diagram, which in general terms describes the user's communication with the bot and the bot with the helpdesk.

As can be seen from the diagram that was presented above, we need to create a request (or ticket) in GLPI, add a comment to it, delete (close) the ticket. Consider a separate part of the diagram associated with the interaction with GLPI - the part where we choose if send is false or true (Fig.2).

This sequence diagram shows the following: the first time a message is sent by a user, we locally create a ticket record in our database. After the user continues to work with the ticker, we must recognize whether the user has clicked the "Send" button or not.

If he hasn't clicked it yet ("send is false"), then we continue to save his messages in our database within the same ticket with which he works and invite him to continue typing.

If the user has already clicked the "Send" button ("send is true"), then we update the information that the request has been sent, send it to GLPI, and also update the task identifier that came from the helpdesk system (since they also have their own database with completely different identifiers), and we send the user a response stating that the application has been successfully sent and he can only wait for a response (Fig.3).

Overview of the main methods of interaction with HelpDesk

In order to understand how to organize the entire declared process, it is necessary to study the API (Application Program Interface) provided by GLPI.

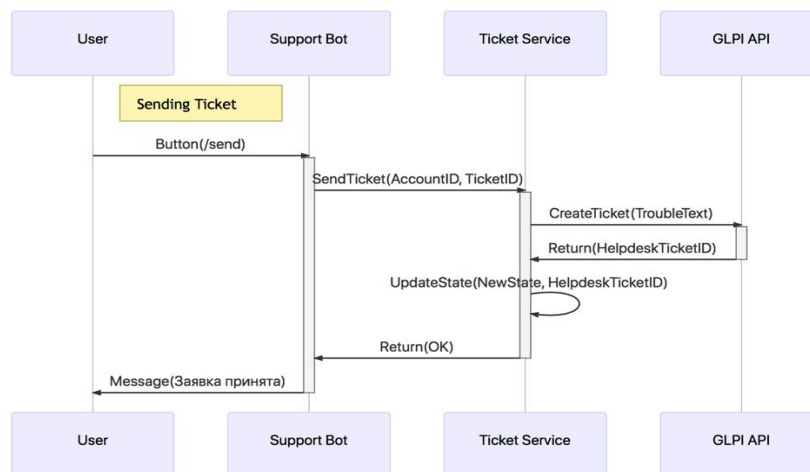


Fig.3 - Sequence diagram of interaction with GLPI

First you need to define the methods of the API Ticket Service, which it will provide to customers:

- create tickets;
- send ticket - there will be a little more complicated logic, since we can send all messages within the ticket as one, and then when you click the "Send" button, we will form a ticket and in GLPI it will be created as the only

entity, but we can supplement the existing one GLPI ticket, and then all new messages (we have the opportunity to add) will be sent as a comment to the application - so here we will need two methods from the GLPI side, but more on that later;

- append ticket (if the user didn't write all the information in one message and wants to supplement the ticket);

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PИИИИ (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 5.015	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	

- close ticket;

Now that we know the Ticket Service API methods, we can combine them with the GLPI API methods. First, the method for creating an application will not be combined with GLPI, because the creation of an application implies the local storage of ticket information in the database when the user first calls (that is, on each first click on "Create request"). And secondly, we do not need the add-on method, since it also works only with the local database and stores all incoming add-on messages. Thus, we still have methods for sending the ticket itself, creating a comment for the ticket and closing the request from the user. In addition to the methods of working with tickets, we need a method of session initialization, which accept the application token and the user's token as the request headers, and as a result sends the token of the session. We will search for detailed information about the declared methods in the GLPI documentation.

In order to use the methods declared in the API description, it is necessary to study some concepts. Firstly, it is endpoint (access point), since the

resource is available through a certain api, then endpoint is just the address (url) by which api can be accessed by client applications, that is, it is the address of GLPI itself specifically deployed under our needs, plus "apirest.php/". Further, the user token is used in the login (authorization) process instead of the login and password, represents the user as a string. This token can be found in the GLPI settings. Application token is an additional way to access API of GLPI. Finally, a session token is a string describing a valid session in GLPI; all methods that access GLPI are required by this token.

Conclusion

In this article we analyzed the principle work of future development support bot which will be integrated with special system for processing tickets - helpdesk.

In future articles we plan to continue the study of the mechanism of support, and specifically develop a bot and service that allows the bot to communicate with the helpdesk.

References:

1. (2018). *Chat-bot — kak napisat udobnogo i poleznogo?*. Retrieved December 1, 2018, from <https://habr.com/post/311008/>
2. (2018). *Boti: chto eto takoe, kak oni rabotayt i pochemu prishla pora v nih razobratsya*. Retrieved December 1, 2018, from <https://test.ru/2016/06/08/chatbots/>
3. (2018). *An introduction to chatbots: the future of customer engagement*. Retrieved December 1, 2018, from <https://oursocialtimes.com/chatbots-huge-opportunity/>
4. (2018). *The future of chatbots for social media*. Retrieved December 1, 2018, from <https://www.falcon.io/insights-hub/topics/social-media-strategy/the-future-of-chatbots-for-social-media/>
5. (2018). *8 Powerful Ways Chatbots Can Transform Your Customer Experience (Insights From David Cancel, CEO of Drift)*. Retrieved December 1, 2018, from <https://www.coredna.com/blogs/chatbot-improve-customer-experience>
6. (2018). *Top 10 Ways a Help Desk Can Help You*. Retrieved December 1, 2018, from <https://www.spiceworks.com/it-articles/it-help-desk/>
7. (2018). *The 3 Essentials of AI Bots for IT Help Desk*. Retrieved December 1, 2018, from <https://chatbotsmagazine.com/the-3-essentials-of-ai-bots-for-it-help-desk-9bce2ffa4446>
8. (2018). *GLPI REST API: Documentation*. Retrieved December 1, 2018, from <https://github.com/glpi-project/glpi/blob/9.1/bugfixes/apirest.md>