# Intelligent Methods and Means of Supporting User-System Interaction Based on its Pattern Analysis

Myroslav Komar[1,†], Vladyslav Poidych[1,*,†], Viktor Fedorovych[1,†], Tetiana Nadvynychna[1,†] and Vadym Vitenko[1,†]

[1] *West Ukrainian National University, Lvivska str., 11, Ternopil, 46009, Ukraine*

**Abstract**

This paper examines the role of AI in modern applications and analyzes existing implementations. The paper is focused on the usage of AI for the purpose of creating a user guide or user engagement. The paper analyzes existing solutions that provide the service from manual creation of user guide and user engagement, or with the help of AI. After analysis, the study provides an experimental architectural solution to address the issue of improvement of user experience when interacting with digital applications that provide the service. The goal of the study is to analyze and compare the existing solutions, highlight key problems and suggest the workflow to remove as much human factor from the implementation as possible but allow the administrator to adjust certain parts of the system to their liking to facilitate better guidance.

**Keywords**

Server, Artificial Intelligence, Tech, Solution, User Experience, User Interface, Web application, Client

## 1. Introduction

In the digital age more and more application users encounter different issues when interacting with applications on the internet. Not everyone delves deep into the guides that the applications provide, as most of the time the core idea of the general guide is to cover every possible aspect in the most abstract manner possible to cater to a wide audience of members. It would have been much better if the service would analyze in real time the user interaction with the system and react upon the specifics of that interaction to help guide users towards a correct resolution of the issue.

Over the course of a couple of years the AI got a lot of traction and has been incorporated in a lot of daily things [1-5]. This increase is sudden, but welcome, and the more the AI is used in different parts of our services, the faster it will learn and become better. So, allowing the users to interact with a service that provides real-time action analysis is very beneficial.

The main subject and object of this research is artificial intelligence and its application in systems to support user guidance. There has been very little application of proactive AI in user guidance, so it's important to research the topic and propose a potential solution. This is a new domain to discover, so the algorithms that work for a standard request-response type of an AI interaction will require adjustment. There are a couple of existing tools (AI and non-AI) that provide the said functionality, so it's important to analyze them and draw up conclusions on how they help solve the issue for proactive user-system interaction. Overall, it would improve the performance of the assistance as

*The Second International Conference of Young Scientists on Artificial Intelligence for Sustainable Development (YAISD), May 8-9, 2025, Ternopil, Ukraine*

* Corresponding author.

† These authors contributed equally.

mko@wunu.edu.ua (M. Komar); poydich.vladislav@gmail.com (V. Poidych); zizix781227@gmail.com (V. Fedorovych); rescen@wunu.edu.ua (T. Nadvynychna); vadvit009@gmail.com (V. Vitenko);

0000-0001-6541-0359 (M. Komar); 0009-0007-5290-051X (V. Poidych); 0009-0009-7806-7751 (V. Fedorovych); 0000-0001-7286-2150 (T. Nadvynychna); 0009-0002-2824-1044 (V. Vitenko)

humans can only process that many users whereas AI can improve that tremendously. The assistant will solve the underlying issue by utilizing a dynamic AI analysis for providing points of reference, and support. The main responsibilities of it will be the analysis of user-system interaction, providing interaction patterns, the number of clicks, mouse movements, link navigation etc. The metrics will be configurable in the main core of the analyzer. The AI will then provide meaningful tips for the users based on user experience. For example, if the user tries to perform an action, but receives an error over that action, but continues to try and perform said action, the AI will react upon this and provide the user with a specific solution to the encountered issue. Based on user interaction with the system, the AI will also be creating static user guide pages based on the most visited and required pages and build an engagement plan for users to tackle the ever-growing user base. This will help engage users faster, which will in the end increase the net amount of value the service provides, as well as increase service throughput and help users better understand how to use the underlying functionality.

The key concepts on how AI could improve user engagement are presented in [6-8]. This information provides a boost in understanding of the user guides as well as possibilities for further implementation.

## 2. Analysis of existing solutions

AI is now embedded into a lot of different applications throughout the internet. This analysis will help us understand how they function. The analysis is aimed at comparing the different implementations of the service which uses AI, or doesn't use it, as well as other services that use AI for a different task, to help us better understand how the AI work is performed. The end goal is to understand how different solutions solve the problem of dynamic proactive user guide, and define findings and key insights of each with a breakdown. This will then help us understand what exactly needs to be solved and how we can apply the studied materials.

### 2.1. Non-AI Facebook user guide

One of the examples is the Facebook virtual center that allows a user to receive help in different sets of inquiries. The implementation of this page is a static web page serving various results based on user inquiries. It is important to take into account existing user guide systems that are not AI focused. It will help us to understand the difference between using an AI system to help a user solve the problem versus using a manually created system supported by people. It is also a great indicator of data, as the Facebook company is vast and Facebook as an application has undergone significant changes over the years, which makes it a perfect study candidate [9].

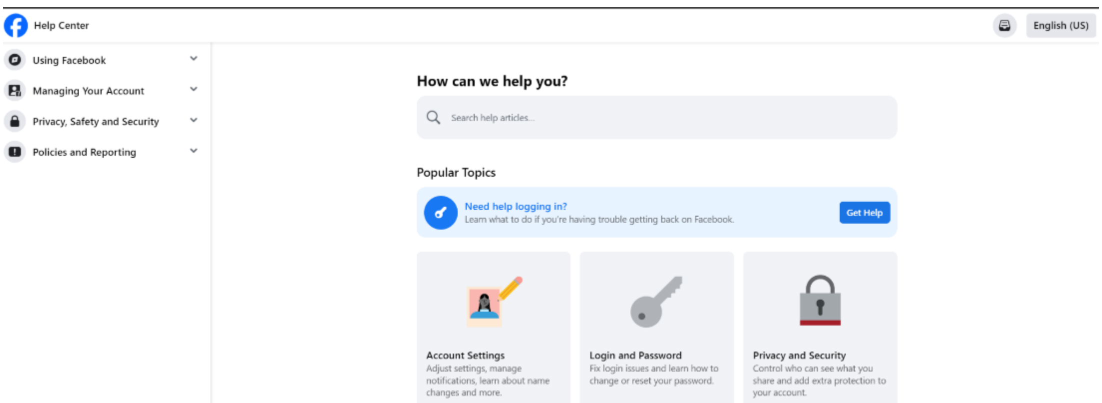


**Figure 1:** Facebook Help Center

Facebook help center is a basis of all Facebook user guide help. It provides a simple UI with a search bar and most searched topics. The user is required to perform the inquiry themselves at first. When you click on the search tab, the commonly searched topics are opened, and the search process can be initiated. When entering a specific inquiry, the data gets displayed to the user in a form of list. The list is very big for the example of the search we performed which at first is a little confusing as to what our next step is. However, defeating the ambiguous results we can arrive at the "Change Email" functionality that Facebook provides.

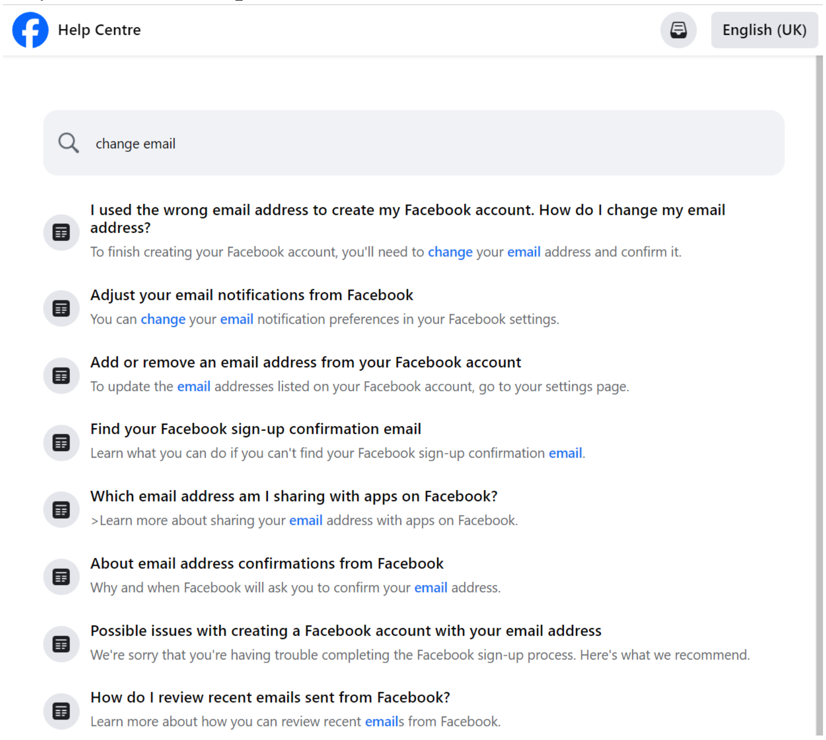


**Figure 2:** Inquiry search

As we can see, the Facebook user guide is very big and tries to encompass a lot of things at once. There are key issues with this.

The search bar is non-interactive outside of user inquiry. What I search for gets translated by Facebook into results.

The results information is overwhelming. The amount of data presented supposedly must cover everything and everyone, but to a user that searches for a specific thing this result only takes away the time and desire to fix the issue.

The deprecated functionality is not updated within Facebook user guide. This means that users will not only deal with tremendous amounts of data, but also be redirected to incorrect links in case those are not updated by the related service. This created a decoupled state between the system and its functionality guide.

In conclusion, the user is responsible for navigating the vast amount of information that is present in the virtual center guide, which in the end not only requires a significant time effort but also requires specific knowledge of the field and how to correctly address the search function. Also, if the logic change has happened within Facebook and, for example, the page responsible for handling user permissions is now moved - the virtual center information must be manually updated to correspond to correct system links and system text. This creates an enormous burden on the application support, as managing and updating the static user guide in a timely manner requires significant effort in communication with the development team, release team, quality assurance team, and with customers, to correctly present the answer to the required inquiry.

## 2.2. Junia AI

Junia AI is an AI assistant aimed at generating content under specific keywords, by utilizing an already existing database of models for each specific field. This AI solution was selected as a candidate because of its capabilities to focus on specific keywords and configurational steps that allow us to better guide the process of the results generation [10].

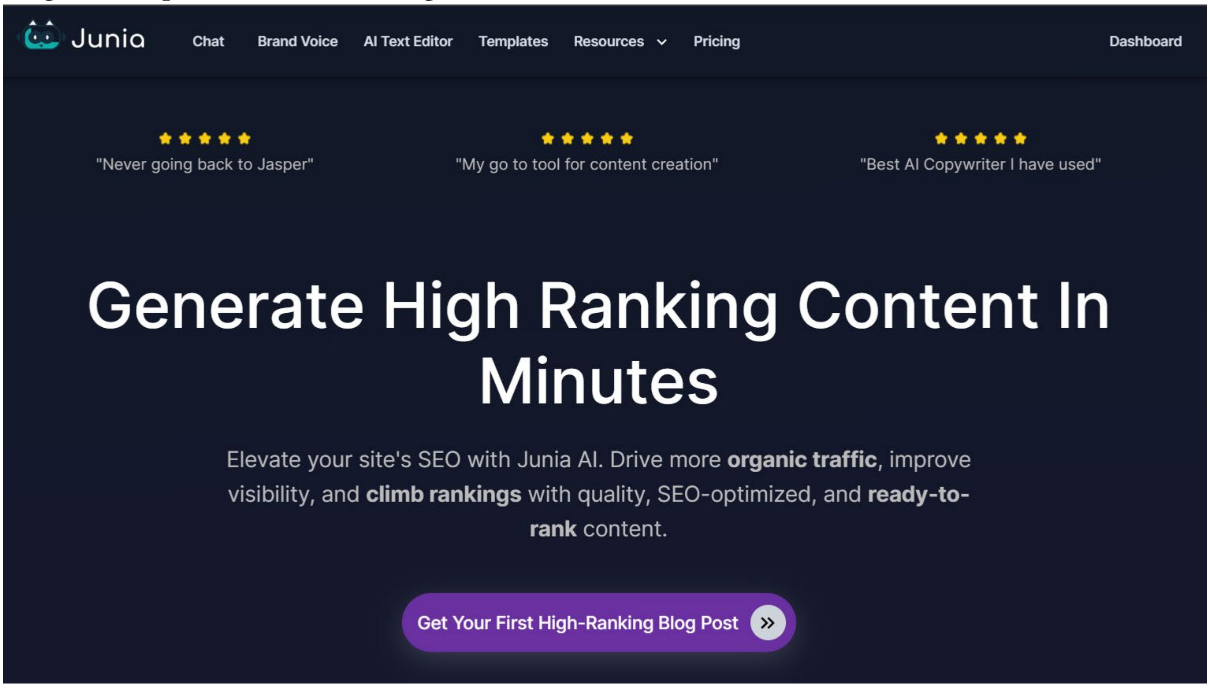


**Figure 3:** Junia AI main interface

It also provides a specific configurational aspect of user-guide called "User Manual", which can be used for implementation of a user-guide based on specific keywords.

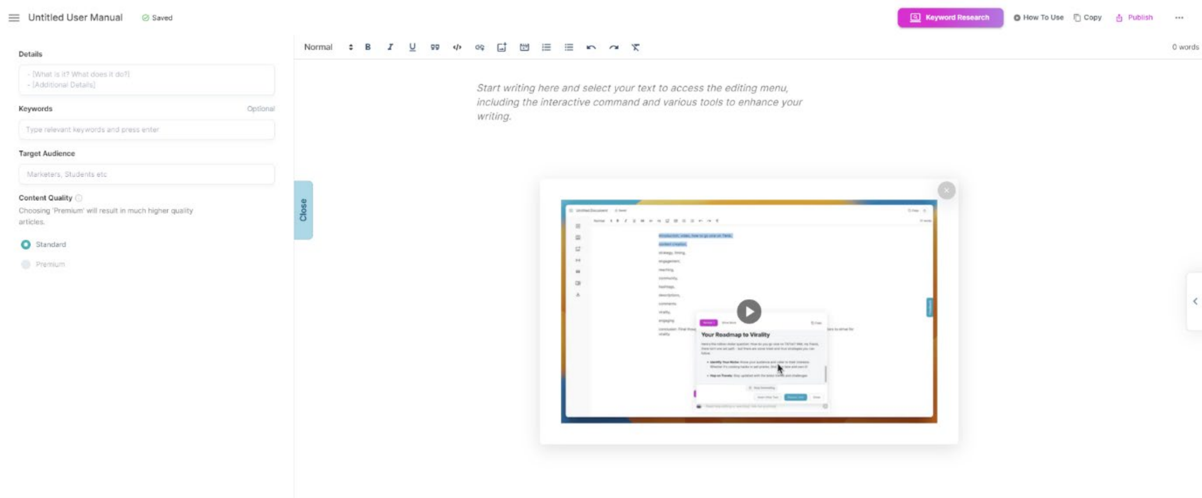


**Figure 4:** Junia AI building blocks

The application provides users the possibility to generate a static page user guide based on the key fields.

1. Field - Details. Provide general information of what the application you are describing does, and additional details.

2. Field - Keywords. Provide keywords to highlight the informational structure of the generated output and focus on a very specific subset of words that best describe the product.
3. Field - Target Audience. Provide information about the target audience for your user manual. The target audience will encompass the technical words usage for the user manual guide, so that no unknown words are used, and the target audience can easily understand what is going on.

The resulting output is a static user guide manual based on general preferences and domain specified in the keywords.

In conclusion, the output provides a very general amount of information that is most likely applicable to the described keywords. The configuration is limited for the free version and doesn't allow the user to highlight more information over the provided services. The output is still required to be processed by something or someone else to correctly gauge the effectiveness of the provided data. The output cannot be regenerated for a specific step, so if something changes the generation process must follow the same pattern from the very start, which becomes tedious if the application changes fast.

## 2.3. GitHub Copilot

GitHub copilot is an AI service that provides users with the possibility to enhance their coding experience by outsourcing certain code implementation to an ai. This facilitates faster software development, by utilizing a knowledgeable companion that helps implement mundane functionality faster. Github copilot is a good solution that highlights user-system interaction. The code insights that copilot provides to the developer in real time is a significant building block, and it can be leveraged to apply the similar approach when working with proactive user-guidance enabled systems, which is why copilot is selected for the study.

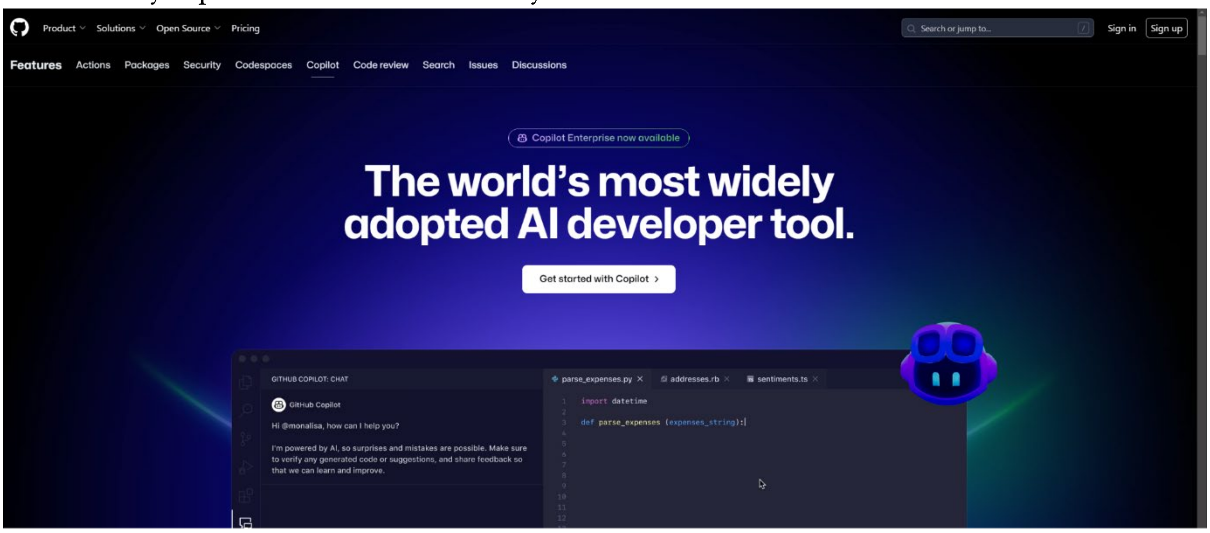


**Figure 5:** GitHub Copilot main interface

Copilot can be installed on a local machine as integrated with an existing variety of IDE solutions. After installation the copilot can be configured to write code in the desired programming language. The communication of the developer with copilot is facilitated via comments that explain what must be done. The copilot will continue to analyze the code that is being worked on and proactively propose solutions for the user. The user can then decide if they want to approve the generated snippet, ask the copilot to make adjustments to it, or completely reject.

```
// write a binary search algorithm
const binarySearch = (arr, target) => {
    let left = 0;
    let right = arr.length - 1;
    let middle = Math.floor((left + right) / 2);
    while (arr[middle] !== target && left <= right) {
        if (target < arr[middle]) {
        right = middle - 1;
        } else {
        left = middle + 1;
        }
        middle = Math.floor((left + right) / 2);
    }
    return arr[middle] === target ? middle : -1;
    }
```

**Figure 6:** GitHub Copilot code suggestions

In the following use case, the copilot will write a binary search algorithm over an array. Once the developer confirms this implementation, the code is now embedded into the file the developer was writing code in and can be further used by the application.

The AI for copilot was implemented by analyzing a vast amount of data from different sources [11, 12]. One of those sources is the public GitHub repositories that GitHub provides to the users. This allows for a very big jumpstart over the AIs that don't have access to that vast amount of code data. Copilot was also analyzed to have improved the performance of the developers significantly by covering tasks that require mundane implementation time automatically. One of the drawbacks is that copilot generates code according to the developer inquiry, and developer is solely responsible for the generated code. This creates an issue when inexperienced developers use copilot, which results in vast amounts of extra code generated that can be omitted, as well as inefficient solutions. However, copilot is a big step in improving development work and greatly increases productivity when used correctly.

## 3. Proposed solution for implementation of the user experience helper

With the help of analysis of existing solutions, the user experience helper will be implemented as an abstract solution for a specific domain.

The core of the AI implementation into the system will consist of multiple phases. The model training will only be done in English and all the data must be added in English. The understanding of the process and general information on the approach that helped envision the architecture is presented in [13, 14, 15]. The aim of this proposed solution is to address the drawbacks that were found when analyzing the counterparts, adding the proactive component that is required for a user-system interaction and possibility to be embedded into multiple domains.

### 3.1. Phase 1. Model base training

This phase will be the building block over the required incorporation of the user experience facilitator into the application. By default, the solution will provide default basics for a subset of specific fields: finance, agriculture, social networking, insurance, banking, food and beverages, gym. During phase one, the model will be trained based on the specific information of the field and highlight field structure and keywords that can be used to identify the field. The model will also learn about the structure of operation within the field, the intricacies and the main driving factors that keep the field alive. The base model training will be done in accordance with existing datasets for required fields.

The base training will then be shared between common models that are presented to the end consumer. The idea is that while in general, the field knowledge is the same, each consumer can then modify the required model to their needs. So, we can assume that model base training will be a sort of “Template”.

### 3.2. Phase 2. Model analytical solution implementation

This phase is the next step of using AI to help improve user experience. In this step the model will be implemented into the end system in the form of an analytical solution. In phase 1 the model will learn about the field of application in general, however in phase 2 the model will be learning about the application in specific.

The phase will require the following:

- Implementation of the model analytical solution into the codebase. The model will be hooked up to the application and capture user interaction with the client. The main interaction keys are: Mouse movement, Mouse clicks, Keyboard presses
- All this information will be sent to the analysis server to group it together into specific keywords along with the target of the user interaction (i.e. the user has pressed a link to navigate to a new page). The model analytical solution will also provide the possibility to use the model in an explicit way by calling each data point manually instead of relying on the model to capture everything that the user does. The developers are free to define their own set of specific model attributes and implement them via the framework of their choice by utilizing the Artificial Intelligence Software Development Kit.
- Scraper server that will be used to analyze the web application content. The key logic of the scraper service is to constantly check over the changes in the application layout by utilizing the already existing file caching. The scraping server will navigate to the application and scrape every possible page within the application as pure text data. The scraping server will then send the file data onto the analysis server.
- The analysis server will analyze the pages’ content and segregate them according to the keywords. The data that is being scraped will also be used during implementation of the model analytical solution to provide developers with meaningful mappers to their specific keywords that they are implementing.
- The processing server will be processing the data analyzed by the analysis server from the scraping solution, as well as processing data from user requests into the system.
- The analysis solution data server. Will contain gathered data.
- The administration portal server will provide an overview of everything that is going on inside the application. Provides possibility for the developers and system admin to see that gathered data, the separated and aggregated information, tweak the information in case the AI was wrong in its assumption.

### 3.3. Phase 3. Model assistant solution implementation

In this phase the developers will need to connect the assistant solution with the application. In phase 2 the analytical solution is responsible for gathering information over the application usage and its codebase and main features. Now the assistant solution is responsible for providing assistance to users based on the collected data before, and now.

The phase 3 will require the following:

- Implementation of the assistant solution into the application. The assistant solution similar to the analytical solution will be presented to developers in the form of UI components for the specific framework of their choice.

- The assistant solution data server. The assistant solution will directly connect with analytical solutions in the form of analyzing the user interaction data. The user interaction data will be sent to the analytical server for processing, however it will also be sent to the assistant server. While in an analytical server the key responsibility is to highlight keywords to then build upon, in the assistant server the key logic is to create patterns of user interaction. These patterns will be used to identify user workflows. By user workflows it is meant that the assistant solution will create a specific path of interaction for a user instead of solely relying on one concrete interaction. As an example, user logic, which consists of entering login information, entering password information and confirming that you are not a robot. Analytical solutions will have keywords identified as “login”, “password”, “captcha”, while assistant solution will create a path of interaction called “login flow” which encompasses these keywords.
- The repetition analysis server. This service is responsible to match the “interaction” the user takes with the “allowed” amount that the “interaction” can take. The “interaction” example would be “login flow”. The administrator will be able to configure all the max parameters for an “interaction” in the administration portal. When the user reaches maximum amount of “interaction”, the system will prompt the user with a resolution to its potential problem.
- The administration portal server. Here the administrators will be able to see “interactions” created by the AI, adjust the names and path of these interactions if necessary, and specify additional metadata information over how the “interaction” should function.

After phase 3 the model is ready to roll out and work on providing the user experience help to affected users.

### 3.4. Phase 4. Static user guide

Phase 4 will include creating a static user guide that the AI can reference in the assistant solution.

The phase will include the following:

- The static user guide handler service. This service will be responsible for generating user guides over the “interactions” present in the system. The static user guide will have specific “interaction” information, how to proceed with the interaction to consider it marked as completed, how to avoid certain pitfalls over the interaction to make sure the experience is the best it can be. After the user guide generation, the service will be responsible to keep it up to date based on the data gathered by analytical solution and assistant solution. The assistant solution will also be able to reference the user guide in the real time help it provides to the users.
- Administration portal for user guide service. This portal will allow administrators and developers to manage the user guide by adjusting it however necessary, according to their best practices. It will allow us to prioritize certain keywords to drive the interaction. The developers will also be able to style the user guide correctly according to the application design requirements.

The first iteration of AI training will be done by utilizing the supervised learning algorithm type. Since the solution is required to be tweakable by the end user, supervised learning will help with this the best. For the algorithm type the following basis for implementations will be used [16]:

- Decision Tree. One of the most common supervised learning algorithms, decision trees get their name because of their tree-like structure (even though the tree is inverted). The “roots” of the tree are the training datasets, and they lead to specific nodes which denote a test attribute. Nodes often lead to other nodes, and a node that doesn't lead onward is called a "leaf".

- Linear regression. Linear regression is a supervised learning AI algorithm used for regression modeling. It's mostly used for discovering the relationship between data points, predictions, and forecasting. Much like SVM, it works by plotting pieces of data on a chart with the X-axis as the independent variable and the Y-axis as the dependent variable. The data points are then plotted out in a linear fashion to determine their relationship and forecast possible future data.
- Logistic regression A logistic regression algorithm usually uses a binary value (0/1) to estimate values from a set of independent variables. The output of logistic regression is either 1 or 0, yes or no. An example of this would be a spam filter in email. The filter uses logistic regression to mark whether an incoming email is spam (0) or not (1). Logistic regression is only useful when the dependent variable is categorical, either yes or no.

The flexibility and robustness of the applied algorithms during the training period presents a challenge of supporting user guidance. The supervised learning should be perfect for the requirements presented, as it will be focusing on working with inputs of a user and deciding on how to react upon it, with another internal system managing the supervision of the specific assigned labels, or "Paths of interaction". As the throughput grows, it is important to assess the performance risks, as the difficulty to collect supervision labels will fall onto another system, and this increases complexity of interaction, and the labels will have to be correctly identified based on a specific user pattern managed outside of the learning spectrum. The amount of data covered in the system is also enormous, as the interaction with users includes not only clicks on specific system parts, but also mouse movements to identify "Paths of interaction", or user confusion with the application[17]. examples of data processing are discussed in [18-20]. The main focus is to understand and label how the interaction is done with the system by using those metrics and identify what combination of specific metrics will define the "Path of interaction".

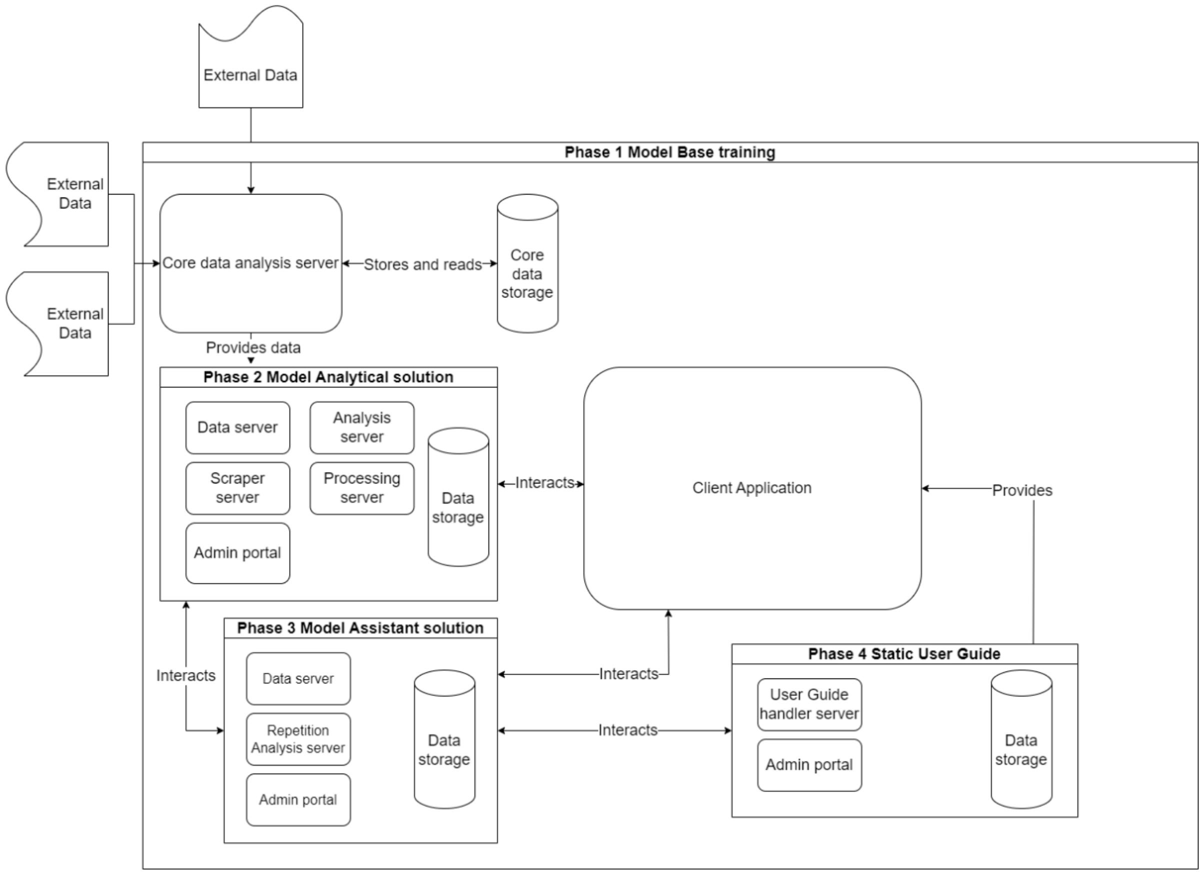

**Figure 7:** High level solution diagram

## 4. Discussions

Selecting the three solutions was carefully decided upon to highlight main problems of user-system proactive interaction study. The Junia AI and Copilot are both great AI tools to provide more meaningful information over the user guides, while Facebook's static user guide is a great way to compare the possibilities of existing AI systems vs non-AI systems, and challenges encountered. The following table highlights the key results of the study based on specific metrics.

**Table 1**
Comparative table of studied solutions

| Criteria | Facebook user guide | Junia AI | Github copilot | Proposed solution |
|---|---|---|---|---|
| Aimed at | The user of the system. | The admin of the system. | The admin of the system. | The user of the system. |
| Usability | Difficult to navigate and find required information. Requires high knowledge of the product and its intricacies to be able to build a result. | Easy to use. | Medium difficulty to use. | Easy to use. |
| Features | Offers a static user guide. User is solely responsible for everything related to their issue with no reactivity. | Offers a dynamic user guide based on the inquiry, which allows for dynamic adjustments in case something changes. However, there is no proactive communication, so the admin is solely responsible for the inquiry. | Offers a dynamic user guide based on the inquiry, allows for dynamic adjustments in case something changes. There is some degree of proactive communication, providing the admin with insights over the current interaction, and suggestions that the user can accept or discard. | Offers a dynamic user guide based on the inquiry. Offers a completely reactive and proactive type of communication providing the user with insights over the complete history of interactions, and suggestions that the user can accept or discard. |
| Possibility to incorporate into existing system | Non-existent. Is designed for solely one system. | Some degree of incorporation is possible, however requires a specific setup to make this work and is limited to their website, | Non-existent. Is designed for solely one domain of software development. | Can be incorporated into any domain or field. And can support any application type. |

| | | | | |
|---|---|---|---|---|
| | | which requires a specific approach of setup. | | |
| Complexity | Convoluted. Searching for an answer can take a lot of time with each page being intervened with one another, old references not being updated, etc. | Easy to use, allows for quick setup and user guide generation. | Medium complexity, requires prior installation process. | High complexity, system design requires prior setup phase. If a new domain or field must be introduced, a complex setup of AI training time is required. Infrastructure setup also requires significant effort. |
| Maintenance | Extremely hard. Maintenance requires knowledge of area changes, team communication, process planning and is strictly tied to what is happening within the team. | Extremely hard. The need of a proxy person to write and update queries requires a significant support of documentation that must be fed into AI at first to produce meaningful continuous results. | Extremely hard. The need of a proxy person to write and update queries requires a significant support of documentation that must be fed into AI at first to produce meaningful continuous results. | Easy. After the initial setup and training, the AI will be responsible for providing user guidance in a form of reactive and proactive approaches. As well as building a static user guide based on the collected information. |

From the comparative table we can derive that the existing solutions have the following key issues:

- They are lacking proactive user communication.
- They are extremely hard in support of dynamic changes introduced to applications.
- The design is very specific to one product/use-case with two of the systems being designed that way.

The Facebook static user guide offers no proactive communication. In the example of interaction, Facebook offered a mix of things per user inquiry, which resulted in convoluted complexity of the specific user guide. The further usage of such a user guide requires not only a lot of user knowledge over the Facebook system, but also tech knowledge to help understand the process. The results of the inquiry are also very vast and try to cover not only the original user inquiry, but also potential commonly asked questions, which makes it more confusing for the user to use the system. Another issue with this is the support requirement. Facebook is ever growing and there is a need to provide a high support for the user guide, which in turn creates a need for a high-cost high maintenance solution and processes. And it shows as the Facebook user guide is very complex and hard to navigate, signifying that the team meets hardships in their support.

The Junia AI, however, provides a better use-case for a user guide with an AI approach. It allows for a specific person that is in charge of writing the user guide to perform a keyword setup, for which the AI will provide the commonly written user guide information. In the example, the user guide was set up for a financial institution, to create a commonly used steps for setup of that domain application. The inquiry of the user gets adjusted and transformed the more information is provided. However, there must be a middle person, between the AI providing the user guide, that is responsible for analyzing the provided result and making adjustments. This specific use case, and the fact that the AI is paid per word, instead of a specific monthly fee, makes it an undesirable choice for a system where users will be performing communication. It also makes it impossible to keep track of interaction per user.

The copilot is a good example of a proactive solution, however the design is only for a specific field of software development. The copilot interaction is seamless, and it continuously analyzes the user inquiries and work, while providing the proactive suggestions to the user. The user can also provide an inquiry and the copilot will provide the answer in a reactive manner. The combination of these approaches helps elevate the user experience and address the issues user encounters. However the current application of copilot is limited to user interaction of software development, and it cannot be used as a supportive measure towards a proactive user guide in other systems.

## 5. Conclusions

From the analysis we can highlight the key issue that is absent in all three of the analyzed solutions - a dynamic proactive system that will interactively support user interaction and provide meaningful response if a user has encountered a problem. Each of the analyzed systems provides a part of the required interaction. The Facebook static guide provides a vast informational base, but unfortunately lacks interactive mechanics and proactive response. The Junia AI provides interactive mechanics and is able to generate information for a subset of data but requires a middleman to work through it and finalize the product. The copilot provides interactive mechanics and proactive suggestions but is limited to a specific field of work and cannot be applied to a lot of other applications. With all this information in mind it is clear that there is a severe lack of a system that would be an AI assistant, with the possibility to be embedded into a specific application for a specific field.

The proposed high-level architecture highlights significant points of the implementation and how to proceed with the onboarding of the suggested solution that should fix the underlying issues. The separation into phases means that the onboarding process will not be blocked by intermittent approaches, as phases are done sequentially one by one. The specifics of the architecture are designed to cover high workloads of data, where the tiered system within the tiered system highlights the effectiveness of each service working independently. This allows us to scale the workload significantly by performing either horizontal or vertical scaling. This also allows us to offset part of the workload for analysis onto the user, which will allow us to offload the CPU and RAM requirement onto clients (web browsers/desktop applications/mobile applications). In the end, the solution will be able to provide meaningful support for new users throughout their service experience, as well as analyze static data gathered from the app to further increase the reliability of provided service, and automatically adjust to any changes done in the app to reduce the time and complexity of maintenance of support information.

Further research will be studying the algorithm application in a set of tasks. The main purpose is to understand what types of algorithms from the proposed set are best used for a particular task, for example - data gathering, data analysis, data aggregation, user interaction analysis and repetition analysis in a use case of user-guidance. Furthermore, the architecture behind the solution will be implemented to facilitate the testing of the algorithms and their adjustments. We can sum up further research steps as follows:

- Study the application of the algorithms used to drive the AI systems and how they can be applied to the designed solution of interactive user-guide system

- Simulation of the algorithms over a custom set of data for finance domain
- Implementation of solution and its architectural phases
- Application of proposed metrics into the real-time web application
- Analysis of the data results and adjustments to the proposed algorithms based on the real data usage
- SDK development for Angular web framework
- Data model training for a finance domain

## Declaration on Generative AI

During the preparation of this work, the author(s) used Junia AI and Github Copilot in order to: Abstract drafting, Paraphrase and reword. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References


[1] Komar, M., Dorosh, V., Hladiy, G., Sachenko, A. Deep neural network for detection of cyber attacks. 2018 IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC 2018 - Proceedings, 2018, 8516753.

[2] Turchenko, I., Osolinsky, O., Kochan, V., Sachenko A., Tkachenko R., Svyatnyy, V., Komar, M. Approach to neural-based identification of multisensor conversion characteristic. Proceedings of the 5th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2009, 2009, pp. 27-31, 5343030.

[3] Golovko, V., Kroshchanka, A., Mikhno, E., Komar, M., Sachenko, A. Deep convolutional neural network for detection of solar panels. Lecture Notes on Data Engineering and Communications Technologies. 2021, 48, pp. 371-389.

[4] Lipianina-Honcharenko, K., Wolff, C., Sachenko, A., Kit, I., & Zahorodnia, D. (2023). Intelligent Method for Classifying the Level of Anthropogenic Disasters. Big Data and Cognitive Computing, 7(3), 157.

[5] Lipianina-Honcharenko, K., Wolff, C., Chyzhovska, Z., Sachenko, A., Lendiuk, T., & Grodskyi, S. (2022). Intelligent method for forming the consumer basket. In International Conference on Information and Software Technologies (pp. 221-231). Cham: Springer International Publishing.

[6] Turing team. 8 Ways AI Can SuperCharge Your Application Development. URL: https://www.turing.com/resources/ai-application-development#user-experience.

[7] LinkedIn community. How can you use AI to create more effective user manuals?, 2024 URL: https://www.linkedin.com/advice/3/how-can-you-use-ai-create-more-effective-user-gwanc

[8] Yuliang Yun, Dexin Ma, Meihong Yang. Human–computer interaction-based Decision Support System with Applications in Data Mining. Elsevier. pp. 285-289.

[9] Facebook. Facebook Help Center. 2024 Meta. https://www.facebook.com/help

[10] Himanshi Singh. Junia AI: One of the Best AI Writing Tool. 2024 URL: https://www.analyticsvidhya.com/blog/2024/01/junia-ai-one-of-the-best-ai-writing-tool/

[11] I. Nilsson, 'An empirical analysis of GitHub Copilot', Dissertation, 2023.

[12] Rizel Scarlett. 8 things you didn't know you could do with GitHub Copilot. Github Blog. URL: https://github.blog/2022-09-14-8-things-you-didnt-know-you-could-do-with-github-copilot/

[13] Alessandro Suglia, Ioannis Konstas, Oliver Lemon. 2024. Visually Grounded Language Learning: A Review of Language Games, Datasets, Tasks, and Models. Journal of Artificial Intelligence Research, Vol. 79 (2024). pp.173-239. URL: https://www.jair.org/index.php/jair/issue/view/1168

[14] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, Ming Zhou3, 2020. CodeBERT: A Pre-Trained Model for

Programming and Natural Languages. Association for Computational Linguistics. pp. 1536–1547. URL: https://aclanthology.org/2020.findings-emnlp.139/
[15] Antonio Mastropaolo, Luca Pascarella, Emanuela Guglielmi, Matteo Ciniselli, Simone Scalabrino, Rocco Oliveto, Gabriele Bavota, 2023. On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot. URL: https://arxiv.org/pdf/2302.00438.pdf
[16] Tableau team. Artificial intelligence (AI) algorithms: a complete overview. URL: https://www.tableau.com/data-insights/ai/algorithms#how-algorithms-work
[17] Liu, Qiong & Wu, Ying. (2012). Supervised Learning. 10.1007/978-1-4419-1428-6_451. URL:https://www.researchgate.net/publication/229031588_Supervised_Learning
[18] S. Rippa, S. Sachenko and Y. Krupka, "Pre-conditions of ontological approaches application for knowledge management in accounting," 2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Rende, Italy, 2009, pp. 605-608, doi: 10.1109/IDAACS.2009.5342906.
[19] M. Komar, A. Sachenko, V. Golovko and V. Dorosh, "Compression of network traffic parameters for detecting cyber attacks based on deep learning," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, UKraine, 2018, pp. 43-47, doi: 10.1109/DESSERT.2018.8409096.
[20] Lytvyn, V., Vysotska, V., Shatskykh, V., Kohut, I., Petruchenko, O., Dzyubyk, L., Bobrivetc, V., Panasyuk, V., Sachenko, S., & Komar, M. (2019). Design of a recommendation system based on collaborative filtering and machine learning considering personal needs of the user. Eastern-European Journal of Enterprise Technologies, 4(2 (100), 6–28. https://doi.org/10.15587/1729-4061.2019.175507