Documentation AI
Find answers to any questions from your documentation

Master’s Thesis submitted to the Constructor Institute
in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Software Engineering

presented by
Maxim Sokolov

under the supervision of
Prof. Manuel Oriol

co-supervised by
Julia Kotovich

June 2023
I certify that except where due acknowledgment has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Maxim Sokolov
Schaffhausen, 15 June 2023
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>3</td>
</tr>
<tr>
<td><strong>Introduction</strong></td>
<td>4</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Suggested solution - DocumentationAI</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Summary</td>
<td>5</td>
</tr>
<tr>
<td><strong>Competitor Analysis and State-Of-The-Art Solutions</strong></td>
<td>7</td>
</tr>
<tr>
<td>2.1 Documentation trends</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Competitors analysis</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Market analysis</td>
<td>10</td>
</tr>
<tr>
<td>2.3.1. SAM</td>
<td>11</td>
</tr>
<tr>
<td>2.3.2. SOM</td>
<td>12</td>
</tr>
<tr>
<td>2.3.3. TAM</td>
<td>12</td>
</tr>
<tr>
<td>2.3.4. PAM</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Customer development</td>
<td>13</td>
</tr>
<tr>
<td>2.4.1 Personal interviews</td>
<td>13</td>
</tr>
<tr>
<td>2.4.2 Public questionnaire</td>
<td>14</td>
</tr>
<tr>
<td><strong>Business Model</strong></td>
<td>23</td>
</tr>
<tr>
<td>3.1. Business model canvas</td>
<td>23</td>
</tr>
<tr>
<td>3.2. Key partners</td>
<td>23</td>
</tr>
<tr>
<td>3.3. Key activities</td>
<td>24</td>
</tr>
<tr>
<td>3.4. Key resources</td>
<td>25</td>
</tr>
<tr>
<td>3.5. Key propositions</td>
<td>26</td>
</tr>
<tr>
<td>3.6. Customer relationship</td>
<td>27</td>
</tr>
<tr>
<td>3.7. Customer segments</td>
<td>27</td>
</tr>
<tr>
<td>3.8. Channels</td>
<td>28</td>
</tr>
<tr>
<td>3.9. Cost structure</td>
<td>29</td>
</tr>
<tr>
<td>3.10. Revenue streams</td>
<td>30</td>
</tr>
<tr>
<td><strong>MVP Technical Implementation</strong></td>
<td>31</td>
</tr>
<tr>
<td>4.1. Overview of DocumentationAI platform</td>
<td>31</td>
</tr>
<tr>
<td>4.2. Requirements</td>
<td>31</td>
</tr>
<tr>
<td>4.2.1 User stories list</td>
<td>31</td>
</tr>
<tr>
<td>4.2.2 Non-functional requirements</td>
<td>33</td>
</tr>
<tr>
<td>4.3. Architecture</td>
<td>33</td>
</tr>
<tr>
<td><strong>Description of The Demo</strong></td>
<td>38</td>
</tr>
<tr>
<td>5.1 Interfaces</td>
<td>38</td>
</tr>
<tr>
<td>5.2 Demo description</td>
<td>38</td>
</tr>
<tr>
<td><strong>Conclusion</strong></td>
<td>41</td>
</tr>
<tr>
<td>6.1. Key Takeaway points</td>
<td>41</td>
</tr>
<tr>
<td>6.2. Future work</td>
<td>41</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>43</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter contains a problem statement and the solution provided by the DocumentationAI platform. The background of the problem and current global trends in IT and Natural Language Processing (NLP) are addressed in this work. In addition, a detailed market review is presented as part of this work. This thesis work primarily focused on the business part of DocumentationAI with a brief explanation of the architecture and implementation and Minimal Viable Product (MVP) description.

1.1 Background

Firstly, I want to refer to my experience in IT. At this point, I have six years of work experience in different companies: product-oriented, outsourcing, and my startup. Based on this, I would argue that most IT employees spend significant time reading documentation and exploring code and new technologies.

And the biggest challenge is finding the proper document and analyzing it. It especially matters because the number of IT specialists is 55.3 million (2020) [1] and, in particular, 8.5 million in the EU (2020) [2]. Keeping this problem with me I started to develop ideas. And for it, I used Design Thinking [3] approach.

For the Ideation phase, I have generated about 20 ideas, and after the selection process, I have the best one covered in the next section.

Another essential point to keep in mind is implementation and technologies. To build the best product the idea should be well-correlated with technologies. The current trend in Large Language Models (LLMs) [4] is helpful here. LLMs have made a huge step forward during the last couple of years. Plus, the hype around ChatGPT [5] has popularized this theme and has made it known to broad layers of people. The important to note: many LLMs have been developed by various companies from many countries for different tasks.
So, I present my idea as a combination of a real problem and modern technologies.

1.2 Suggested solution - DocumentationAI

As a suggested solution for the above-highlighted problem, I am introducing DocumentationAI. It is a web-based chatbot for user questions answering based on internal and external documentation. It allows adding links of Github repositories to the index system. After the setup, users can ask any question regarding mentioned repositories and get answers in chat mode. Moreover, it can get additional plugins and integrations. The platform will allow every IT specialist to cut time off from finding the best information source and from getting answers.

1.3 Summary

The thesis has seven chapters with sections and subsections. This chapter contains a problem statement and the solution provided by the DocumentationAI platform. The second chapter presented the competitor analysis and customer development research. Chapter three discusses the business model canvas and all the details. The fourth chapter provides the implementation requirements, architecture, and
design decisions. The fifth chapter provides a brief description of the DocumentationAI demo. Finally, chapter six discusses the project conclusion and future work description.
Chapter 2

Competitor Analysis and State-Of-The-Art Solutions

2.1 Documentation trends

The current trends in such fields as documentation tools, search engines, and chatbots are covered here.

At the start, for documentation field analysis, I want to categorize documentation. In IT companies, you can find the following types of documentation:

- Internal software documentation. This documentation is written by IT specialists for themselves.
- External software documentation. There is documentation of open-source or enterprise solutions like Oracle, Django, and Linux for IT specialists who use this documentation.
- End-used documentation. It describes how to use the product. Also, it could cover technical details and processes such as installation and configuration.
- Operational documentation. There is a work result of an operational team. Usually, it aimed to organize non-principal activities, such as salary payments and vacations.

In DocumentationAI, I decided to concentrate on the first three types as on most-commonly used.

Some documentation management statistics include:

- The market size has a Compound Annual Growth Rate (CAGR) of 16.8% during the forecasting period till 2029 [13]
- The documentation size will grow three times over the next five years
- The knowledge worker spends about 2.5 hours per day, or roughly 30% of the workday, searching for information

Making a long journey from paper cards through basic systems on early computers to modern online web-based tools, documentation management systems have reached their current state. Now there are modern cloud-based systems. Documentation management trends are outlined below:

- Integration and Interoperability [6]

  The state-of-the-art DMS platforms must have plenty of integrations with a wide range of software and third-party applications like messengers, code management systems, and task management systems, such as Slack, and Jira. It helps create cohesive ecosystems that facilitate cross-functional collaborations and process automation.

- Scalable Architectures [6]
Emerging DMS tools should have a scalable architecture. It grants the ability to scale businesses, products, and teams, to change business needs.

- **Cloud-Based DMS** [6]
  This feature can bring companies flexibility, scalability, and accessibility. Moreover, leveraging cloud infrastructure decreases infrastructure costs vastly.

- **Reliable security** [6]
  Almost all companies could have sensitive data in documentation. Therefore, high security becomes a must-have feature.

- **Artificial Intelligence And Machine Learning** [6]
  It holds the core potential to automate document classification, extraction, and analysis, which can significantly reduce manual effort and minimize human error. Moreover, these advancements can allow organizations to uncover patterns, trends, and different insights from their data. Consequently, they can make informed decisions more proficiently.

- **Smart Search functionality** [7]
  It mentioned earlier, document search takes a long time. Therefore there is a feature with huge potential to cut costs and accelerate work.

- **Smarter Automatization** [8]
  Routine operations exist in thousands of companies. And automatization is a good helper for these tasks. And modern technologies like Robotic Process Automatization (RPA) [9] are the basis for a revolution in these ready-to-use processes and applications.

- **Team Collaboration** [10]
  In a time when the number of teams and team sizes grow exponentially, DMS tools must provide collaboration property for fast and easy cross-team work.

### 2.2 Competitors analysis

In my opinion, the analysis stage should start with a competitor definition. Direct competitor is an application answering questions based on internal documentation. At the same time, we shouldn't forget about indirect competitors. Especially now, in a time of AI hype and insane growth of these startups. It includes products that develop B2B AI chatbots, search engines, and other similar products to DocumentationAI.
For making a comparison, the feature list presence is mandatory. And definitely, it would be hard to compare. Hence I have selected the most important features. The final feature list is the following and their abbreviations:

- **Question answering [QnA]**
  There is a basic feature. A comparable product should answer questions from customers’ users.

- **Working with internal documentation [Internal documentation]**
  Moreover, a product should have the functionality to work with internal documents. Including uploading, indexing, and storing. In our case, with IT technical documentation, a comparable product should be able to work with such systems as GitHub. The main reason for this selection criterion is security of sensitive data.

- **Working with external documentation [External documentation]**
  The similar to the previous point.

- **Working in chatting mode [Chatting mode]**
  Today, most AI models can’t reach 100% accuracy (or other metrics). For example, some Google models have reached only 90%+ [11]. Therefore, such undetermined tasks as the current question-answering problem, won’t be perfectly solved. And tricks like additional questions in chatbot mode allow tuning quality of conversation via follow-up questions.

- **Handling IT docs [Handling IT docs]**
  This criterion has been selected based on poor answers quality from some services. Any new corpus domain like IT is a big challenge for AI systems. And for accurate solution getting, some companies like Microsoft train AI models from scratch [12]. That's why all product accuracies should be checked on IT questions.

- **Providing source links within answers [Source links]**
  The same as chatting mode providing source links is a trick to upgrade product quality metrics. The functionality covers cases where an answer is not accurate enough or when an answer is partly correct. Going to links provides answer sources. Additionally, this feature adds more trust from users to the product, because DocumentationAI's answers are always based on the user’s documentation.

- **Ability to have a prebuilt index or source documentation for work [Prebuild index]**
It is a criterion aimed to provide convenience to clients. IT specialists should have a ready-to-work environment to use in everyday work and not setup it up every working session.

- Presence of standalone application [Standalone]

As I mentioned in the previous point (‘Internal documentation’) the security of sensitive data is very important for customers. This point is based on interviews with IT specialists (see ‘Customer development’ section). And in my opinion, it will be the key to disrupting the market. The ideal product should have a standalone application that could be launched on customers’ servers, and probably, work in an isolated network.

### Thesis comparison (in production) ...

<table>
<thead>
<tr>
<th>Name</th>
<th>QnA</th>
<th>Internal docs</th>
<th>External docs</th>
<th>Chat mode</th>
<th>Handle IT docs</th>
<th>Source links</th>
<th>Prebuilt index</th>
<th>Standalone</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentationAI</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>In Future</td>
</tr>
<tr>
<td>Nuila</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fini</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>9 integrations</td>
</tr>
<tr>
<td>HeyBot</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CustomGPT.ai</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FileChat.io</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Mendable</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Glean</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Moveworks</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Alsera</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>mem.ai</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 2. Competitors comparison table

The table above shows that the ideal tool does not exist and DocumentationAI could be them.

### 2.3 Market analysis

There are many ways to estimate the market size. As we have only open-source data available and need a quick but as good as possible approach to estimate our metrics for the market size, possible options include:

- PAM - Potential Available Market
- TAM - Total Available Market
- SAM - Serviceable Available Market
- SOM - Serviceable Obtainable Market
An additional challenge here is determining of market. The reason for it is a non-standard platform that could be attached to markets: documentation management systems, chatbots, and search engines. For a more fair evaluation, I get a minimum of these markets or use the most correlated market. So let me provide some statistics:

- **Documentation Management Systems**
  
  The global market size for document management systems is projected to grow from $6.23 billion in 2023 to $18.97 billion by 2030, at a CAGR of 17.2%.[13]

- **Chatbots**
  
  The global chatbot market size reached $0.84 billion in 2022. And experts estimate the market size as around $4.9 billion by 2032 with a noteworthy CAGR of 19.29% from 2023 to 2032.[14]

- **Search Engines**
  
  The global market size of enterprise search systems valued at $3.75 billion in 2021, and experts expect to a value of $7.84 billion by 2028, at a CAGR of 11.75% over the forecast period (2022–2028).[15] One of the representative examples, Elastic Search[16], obtained $1.02 billiard revenue in 2023.

Finally, we have different markets with sizes from $1 billion to $6.23 billion. It has a six times difference. It will be on my mind during the calculations. And next part is SAM as the most challenging problem.

2.3.1. **SAM**

In my opinion, the determination of SAM for DocumentationAI is the market of English-speaking IT companies in North America and Europe. These regions have similar business and IT cultures and are one of the biggest economies.

6.2 million [17] ICT specialists work in North America and 9.3 million - in the EU [18]. In sum, 15.5 million work in the ICT sector in these regions. Let's assume they need a question answering as part of their work.

The next step is pricing. It could be solved using different ways. Anyway, most market evaluations want to understand the level of numbers. So assume $10 as our price per user per month. The similar tool TabNine [19] that improves coding speed has $12 as a subscription price. The well-known tool GitHub Copilot [20] has a price from $10 to $19. That's why I use $10 for the price.
Calculations:

\[
15.5 \times 10^{9} \text{ people} \times 10 \text{ USD per month per user} \times 12 \text{ months} = \$1,86 \text{ billiard} - \text{Service Available Market}
\]

Additionally, we can look at SAM using another way - using the main competitor or similar platform. For that, I selected StackOverflow. It is one of the largest question-answering platforms, but it collects and shows human answers. Their annual revenue is $172,1 million [21]. Remembering that it is not an entire market, the conclusion is the result of the $1,86 billiard is correct.

2.3.2. SOM

Certainly, no one can say the exact evaluation of Obtained Market without sales. And I can only assume that DocumentationAI will get 10% and SOM size will be a $186 million market.

The 10% is a simple assumption to get the correct number level.

2.3.3. TAM

For Total Available Market I can apply the following definition. It is the market of English-speaking IT companies worldwide. It could be easily calculated based on SAM.

The worldwide full-time employment size in the ICT sector is projected to reach 62 million in 2023. So TAM is:

\[
\frac{\$1,86 \text{ billiard}}{15.5 \times 62} = \$7.44 \text{ billiard} - \text{Total Available Market}
\]

2.3.4. PAM

Firstly, Potential Available Market should have a definition of potential. Thinking about new features, I would write that DocumentationAI can handle all documentation types potentially. Based on statistics [22], I argue that IT documentation is 7.4% of all documentation. Hence, potential DocumentationAI could handle all of that and have the following market:

\[
\frac{\$7.44 \text{ billiard}}{0.074} = \$100.54 \text{ billiard} - \text{Potential Available Market}
\]

In conclusion, the market analysis shows a huge potential market and, at the same time, big enough serviceable available market size. All of that means that DocumentationAI could be successful in reality.
2.4 Customer development

This section covers customer development. This method allows validation of a problem existing via interviews with real people. Customer development is a formal methodology for building startups and new corporate ventures. It is one of the three parts that make up a lean startup (business model design, customer development, and agile engineering) [23]

I used two forms of interviews: personal face-to-face deep interviews and the public form with questions.

2.4.1 Personal interviews

The personal interviews gave the following insights:

- Time consumption
  
  Searching for required documents is a time-consuming everyday activity.

- Messengers are sources of information
  
  In most cases, engineers ask questions in chats, and colleagues answer them. Partly because of the problem from the previous point.

- Documentation is outdated
  
  Usually, IT products grow so fast that documentation can’t follow them. That’s why documentation is going to be outdated.

- Onboarding spends a significant time
  
  Every company needs new employees. So sometimes teams are growing too fast and the spending time for the onboarding process became significant. Mostly onboarding includes activities aimed to get acquainted with the company and reading all required internal documents. DocumentationAI can solve this problem as well.

- No all-documents search
  
  The modern IT company has plenty of tools for work, e.g. Jira for task management, Confluence for documentation, and Slack for messaging. All of that has everyday read information. That’s why an all-documents search is very important for possible customers.
After getting some insights above, I have started preparing a public questionnaire. It gives answers from a bigger target group. All the necessary data, including the questionnaire, insights, and comments, are described below.

2.4.2 Public questionnaire

The questionnaire was built using Google Forms and shared between experienced colleagues, friends, and social media connections. I have received 24 responses, and the results are shown below.

The first block is dedicated to the ‘know your customer’ purpose.

![Bar chart showing official job titles](image1)

**Figure 3.** It shows that most interviewees are engineers from IT companies.

![Bar chart showing approximate company size](image2)

**Figure 4**

Figure 4 represents the company sizes where the respondents work.
Age distribution in Figure 5 also is essential data, because questionnaire answers could be biased. But the average tech employee’s age is less than other industries workers [24].

The next section is dedicated to understanding everyday activities and analyzing time consumption.

Developing, onboarding, operations, project management, and research are leading activities where DocumentationAI can help potential customers.
It is also essential for me for the same reasons as the reasons from the previous point.

The third block is dedicated to solution finding. It covers the ordinary question-answering process.

This figure shows that mostly respondents use additional instruments like documentation or search engines like Google for this activity. The interesting point is that asking colleagues in chat is used by 62.5% of respondents. Erasing it could improve productivity significantly.
In continuing to the previous point, this result highlights the same problem with more details.

Not only asking in chat could be problematic. Answering too, but not from time consumption time. Context switching kills productivity seriously [25].
The fourth block covers onboarding process questions.

How long is the active phase of onboarding on average (active phase requires big attention from employee's manager)? [in weeks]
24 responses

![Bar chart showing time spent onboarding](chart11)

Figure 11

This is a statistical question. Here I have tried to predict time consumption saved theoretically.

How much time do you spend on question answering during the active onboarding phase on average? [in % of working time]
24 responses

![Bar chart showing time spent onboarding](chart12)

Figure 12

More than half of the respondents spend more than 20% of their time on onboarding. It is a considerable time that could be cut off. And onboarding optimization could get rid of routine.
The next block is dedicated to understanding how people work with documentation.

How many questions do you usually solve using searching inside documentation (internal and external) [in % of questions]?

24 responses

![Bar chart showing the distribution of time spent solving questions using documentation.]

Figure 13

The result is more than 40% of respondents' requests solved the documentation on average.

What is average time for finding answers using documentation? [in minutes]

24 responses

![Bar chart showing the distribution of time spent finding answers using documentation.]

Figure 14

Half of the respondents spend more than 10 minutes finding needed information on average.
Mostly, external and internal tech documentation is the key to getting all information.

The first five documentation sources are popular. And DocumentationAI should have integration with them.
These documentation sizes could be handled by such modern IT systems as DocumentationAI.

Figure 17

Figure 18
The statistics highlight the documentation outdating problem. The statistics below show the solution form problem.

All of that shows how important to have a standalone version for security purposes.
Chapter 3

Business Model

3.1. Business model canvas


The Business Model Canvas for the DocumentatioAI platform is presented below and contains a detailed description of the key partners and customers, key project activities, and other important blocks.

![Business Model Canvas](image)

Figure 22. The business model canvas

3.2. Key partners

The DocumentatioAI partners are described below.

- AI model producers

At the current state, the plan supposes to use third-party AI models. Now I use GPT4 from OpenAI. And other solutions could be applied, such as Cohere, AI21, and Jurassic.
• Cloud infrastructure providers

For web-based platforms like DocumentationAI, cloud infrastructure providers are the primary working partners. Some vendors allow deploying code as SaaS, training our AI models, and using infrastructure for development. One from Google Cloud Platform, Amazon Web Services, and Microsoft Azure could be a good choice. Because they are market leaders and have plenty of ready-to-use services.

• Payment platforms

The main goal of a business is gaining money. For online platforms, electronic payment platforms are a single way to receive payments. Therefore such platforms as PayPal and Stripe are necessary for DocumentationAI.

3.3. Key activities

Here are the principal activities for the DocumentationAI startup:

• Marketing, Sales, BizDev

There are the main activities of any business. Sales gain money. Marketing gives new customers and reduces churn of them. Business development (BizDev) is a wide range of activities, including building strong relationships with customers and partners, growing business globally, and caring about the team.

• Platform development

There is a core technical activity. In our case, DocumentationAI is a tech product. Therefore the team should spend significant efforts to develop and upgrade technologies under the product.

• Continuous AI model training

DocumentationAI is an AI-first product. And own model is one of the greatest competitive advantages. Especially nowadays when we can see new models every day. That’s why the team should train its own AI model and retrain it endlessly.
• Logging feedback

To be competitive, DocumentationAI should upgrade itself regularly. For it, anyone can use feedback from customers. And DocumentationAI has a unique ability. The platform communicates with users every day via chatting with question answering. Also, it could ask for feedback after every answer. And buttons like 'Like' and 'Don't like' could be a simple key for a strong feedback channel.

• Metric calculations

To be competitive, the team should evaluate itself. DocumentationAI's market is very young, and it doesn't have any public benchmark. Hence the team should have its metric evaluation system. The team should evaluate all product parts in real-time, including AI model accuracy, the platform behavior in production, customer churn, etc.

• Maintenance

Definitely, DocumentationAI should follow its behavior and states. Based on real-time metrics from the previous point, the team could build a system for maintenance purposes.

3.4. Key resources

The principal resources for this startup project are the following:

• Team

The team is a core key for a successful startup. In case of any product failure, the team can pivot and invite another product. Therefore, DocumentationAI desires to have skilled, motivated and involved people on the team. And obviously, people with business and technical backgrounds on the team are great to have.

• Involved customers

Customers could be our partners also. They can not only churn. They can acquire new customers via the word-of-the-mouth channel. Also, they can help
to improve products. There is a reason why it matters to have customers involved in product life.

- **Servers**

  Infrastructure is a principal component of any development. Such web-based products as DocumentaionAI use cloud infrastructure every day.

- **GPU cluster**

  This component could seem strange and unnecessary, but GPU machines are a mandatory part of modern training loops of AI models.

- **AI models**

  Following the ‘fail fast’ strategy from Lean Canvas methodology [26] means going as fast as possible. And for that, DocumentaionAI should utilize ready-to-use AI third-party models at the start.

- **Data markup system**

  To be a competitive product, as I mentioned before, DocumentaionAI has to have its own trained AI model. For that, the team should have labeled data for training. That’s why I am going to log all marked user feedback, and additionally, I am going to have a third-party data markup system. It provides an extra labeled dataset for training too.

### 3.5. Key propositions

The principal propositions for the customers of DocumentaionAI are:

- **Fast Q&A**

  There is a killer feature of DocumentaionAI. Now customers’ users can obtain answers to any questions according to their internal or external documentation.

- **Decreasing the time to find any information**
There is another killer feature of DocumentationAI. Now customers/users can obtain questions/answers, and explore source documentation links. Moreover, DocumentationAI could be used as a search engine too.

- Smaller context switching

  Based on the previous points, our much-loved users obtain the ability to switch their context critically rarely.

- All data sources search

  DocumentationAI platform is a flexible product as puzzles. The set of integration transforms to all data source search. That means that users can find all requested information in one interface, in DocumentationAI from such sources as Jira, Slack, Confluence, etc.

- More accurate Q&A via chatting mode

  As I mentioned before, now no one can reach 100% accuracy, especially on a too undefined task like question answering. That’s why all have to provide additional instruments for improving accuracy. Chatting mode is a solution for this task.

3.6. Customer relationship

As I mentioned before, DocumentationAI wants to have involved customers. It requires special customer relationship activities. There are outlined below:

- Our platform
- Online contact

There are simple ways to reach this goal. And the team is flexible and ready to have custom channels like direct Zoom and face-to-face meetings.

3.7. Customer segments

It is critically important to understand the specific customer segment, as, in the future, this metric will help the project streamline its marketing initiatives and not waste money. Based on the customer development initiative, I have split potential customers into three groups:

- Small-size IT companies
These companies are fast in making decisions and deals. It is a big plus for DocumentationAI. But they have got a non-standard characteristic such as tiny documentation. They grow so fast and have not got enough time to write documentation. Therefore, the main proposal is a question-answering system that works with external documentation like Github and Gitlab Wikies.

- Medium-size IT companies

  Vice versa, that companies have stable documentation with regular updates. That's why they could consume all features of DocumentationAI.

- Non-IT Companies with IT teams

  There is an interesting case. Today, most big companies become IT-first, because the competitive atmosphere pressures them. And they hire some IT teams for internal purposes. Summarizing, I claim that they have as diverse documentation as team purposes. Hence, they could be early adopters for developing additional products for others documents types.

Important to point out why I didn't include big-size IT companies. The reason for that is the long deal lifecycle. For early-stage startups such as DocumentationAI, it could waste all money before a deal.

3.8. Channels

First of all, for the young startup, it is essential to have a clear vision of how to promote the new app on the market and what channel types are suited for that purpose. A good way for the project to start is to have a list of channels sorted in a hierarchical order based on relevance. Firstly, I show channels for acquisition and after for maintenance. The channels list for this project is outlined below:

- Digital marketing

  It is a good idea to use digital instruments to acquire new users who are IT companies. There is no question that digital marketing for business-to-business (B2B) works not such perfectly as for business-to-customers (B2C). But this channel is a practical way for acquiring small-size IT companies. Most importantly, DocumentationAI might handle thousands of them.
● Conferences

This is a classical channel for B2B marketing. Within a few days, a startup can make dozens of connections that could become long-term relationships.

● Word-of-the-mouth

Hope, customers will be happy to use DocumentationAI and share their experience with colleagues, partners, and friends.

● Our platform

For maintaining customer relationships, any startup has to have a direct channel. Fortunately, DocumentationAI could be it. For example, we can send weekly statistics of using and statistics of successful helps.

● Emails

This classical way could be used too for the same purposes as the previous point.

3.9. Cost structure

The following is a structure of costs:

- Platform development
- Train AI models
- Data markup
- Platform maintenance
- Servers
- Team
- Marketing, Sales, BizDev
- Legal and taxes

All of that should be clear for understanding based on previous points. And additionally, I want to comment on the ‘Platform maintenance’ point. Besides regular activities such as development and business development, DocumentationAI should maintain the system quality, set up an environment for customers, and help with any questions along their work with DocumentationAI.
3.10. Revenue streams

For the startup DocumentationAI there is a consideration that the app will have a model of the revenue streams:

- **Subscription per user**

  There is a primary revenue stream. Subscription is a convenient payment way for customers. For DocumentationAI, there is a simple way of growing exponentially. By default, subscriptions are per user per month. And definitely, this scheme could be modified along with DocumentationAI's progress.

- **Payment for installation**

  The regular work with indexed documentation is not such resource-consumed as the first installation and indexing of all documentation for a customer. That's why installation could be paid once, separately from the subscription.

- **Retraining payments**

  After the previous points, DocumentationAI could provide an additional proposition. It is continuous or sometimes retraining their own AI model. These activities could be paid separately, or it could be another subscription line.

The above points describe only forms of payments. Another important thing is pricing. It is a hard-to-determine property of DocumentationAI. And it will be changed by market reality. But I have some ideas for forming it.

First of all, the value for customers determines the price. Let’s see some statistics regarding working processes.

According to McKinsey report [27], The knowledge worker spends about 2.5 hours per day, or roughly 30% of the workday, searching for information. According to Worlddata [28], the average annual wage for employees in the United States in 2021 is $70,430. So each employee costs the organization over $21,000 per year merely to search for information, and that’s $1740 a month.

If DocumentationAI costs $1740 per month per user, then it could cost for customers as 10% of that. It means that the $50-100 subscription price could be fair.
Chapter 4

MVP Technical Implementation

This chapter contains an architectural description with a high-level system overview and justification of chosen user scenarios and design decisions.

4.1. Overview of DocumentationAI platform

The DocumentationAI software offers an interactive platform to question answering based on internal or external technical documentation. With cutting-edge technology, the platform aims to equip users with an easy-to-use interface. DocumentationAI application aims to provide thousands of users with a unifying instrument for getting any information and answers regarding it.

For starting requirements engineering, features should be listed. There is a combination of all the previous points. Recap of them:

- Question answering
- Working with internal documentation
- Working with external documentation
- Working in chatting mode
- Handling IT docs
- Providing source links within answers
- Ability to have a prebuilt index or source documentation for work
- Presence of a standalone application

The following section contains transformed product requirements into technical ones.

4.2. Requirements

4.2.1 User stories list

The user stories approach is a classical and industrial way to describe functional requirements. Additionally, this way is a part of the Software Requirements Specification standard by IEEE [29].
At the start, the users' types should be described, as well as their activities. The users' types:

- End users
- Customer managers
- Developer
- DevOps Engineer

The user stories list is:

- As an end user, I want to have a web interface for requests to DocumentationAI.
- As an end user, I want to have the ability to get answers on my questions for purposes of decreasing search time and decreasing context switching.
- As an end user, I want to get source links for purposes of checking answers and getting more detailed information.
- As an end user, I want to have answers from internal as well as external documentation.
- As an end user, I want to be able to change documentation sources for the purpose of customizing the platform.
- As an end user, I want to have chat mode for the purpose of requesting additional detailed questions.
- As an end user, I want to have the ability to set up an environment and don’t do it again every time of using the platform for the purpose of convenience.
- As an end user, I want to have a standalone version of the platform for the purpose of sensitive data safety.
- As a customer manager, I want to receive reports with analytics of successful uses of the platform for the purpose of providing reasons to continue uses of the platform.
- As a DevOps, I want to have a real-time analysis system for the purpose of handling any issues as soon as possible.
- As a DevOps, I want to have the ability to get the platform in a ready-to-use box for the purpose of providing DocumentationAI for end users.
- As a developer, I want to have the feedback button shown for any answers for the purpose of collecting labeled data for further training AI models and for providing reports for customer management.
4.2.2 Non-functional requirements

The list of Quality Attributes used for the non-functional requirements description is below:

- The waiting time for an answer should be less than 30 seconds.
- Every answer should contain at least two sources.
- The accuracy of answers should be more than 80% by F1, and SAS metrics.
- The platform should handle 1000 users simultaneously.
- Any issues with the platform should appear in the maintenance system within 5 minutes.
- The chatbot should support up to 20 messages during one conversation line.
- The platform should store up to 1 million pairs of questions and answers got from user feedback.

4.3. Architecture

Software Architecture is a broad discipline of Software Engineering. And definitely, different approaches are applied to the Architecture Design task. A widely used approach within the IT industry is the C4 model [30]. It implies a top-down approach and splits architecture into four levels: System context diagram, Container diagram, Component diagram, and Code diagram. By the C4 approach, a Container diagram inherits a System context diagram, and so on. All diagrams are utilized for certain software parts.

Before the architecture presentation, I want to outline the technology stack used in the architecture diagram:

- Programming languages: Python and JavaScript
- Maintenance system: Zabbix and Grafana
- Containers: Docker and Kubernetes
- Frontend: ReactJS
- Backend: Django, FastAPI
- Authentication: AWS Cognito,
- AI Stack: NVIDIA Triton, AWS SageMaker, LangChain, LLamaIndex
- Storages: AWS S3, AWS DynamoDB

In this section, high-level architecture according to C4 model is covered below.
Figure 23. The system context diagram
Figure 24. The container diagram of DocumentationAI Subsystem
Figure 25. The container diagram of AI Subsystem
Figure 26. The container diagram of the Feedback and Logging Subsystem
Chapter 5

Description of The Demo

This chapter includes a description of the supported platforms and a short description of the demo based on some implemented scenarios.

5.1 Interfaces

The DocumentationAI is designed and built using Streamlit [31], which helps to develop and deploy web-based systems. It provides a well-designed web interface for the platform.

5.2 Demo description

The workflow of DocumentationAI consists of 3 steps:

- Document source initialization

  There is the simplest step for a user and one of the hardest for the platform. Now the platform supports data ingestion only from GitHub. So a user should insert a Github link to the platform and run indexing. During this time, DocumentationAI downloads the repository, finds all documentation files, and indexes documentation.

- Asking the first question

  There is a text area fulfillment. A user can ask any question related to the data source. Here the user can use words like ‘the repository’.

- Continue a conversation in chatting mode

  It is similar to the previous point, but the user can ask clarifying questions to the platform.

The screenshots of described points are outlined below.
Documentation AI

Enter your Github repository link

https://github.com/UKPLab/sentence-transformers

Figure 27. Initialization screenshot of Documentation AI

Documentation AI

Enter your Github repository link

https://github.com/UKPLab/sentence-transformers

Figure 28. Documentation AI with inserted GitHub link in the processing state

Documentation AI

Enter your Github repository link

https://github.com/UKPLab/sentence-transformers

Your question about repository:

What this repository does do?

Figure 29. Documentation AI is processing the question
Documentation AI

Enter your Github repository link

https://github.com/UKPLab/sentence-transformers

Your question about repository:

What this repository does do?

This repository provides a library for sentence embeddings, allowing users to represent sentences as vectors and compare them using cosine similarity. It is maintained by Nils Reimers of the Ubiquitous Knowledge Processing (UKP) Lab at Technische Universität Darmstadt. The webserver/web hosting company may collect certain log files to prevent abuse of services, but no tracking services or cookies are used to track or re-identify visitors. Answer correctness: 77.3% Source link: https://github.com/UKPLab/sentence-transformers/tree/master/docs/contact.md

Figure 30. DocumentationAI provides the answer to the question

Documentation AI

Enter your Github repository link

https://github.com/UKPLab/sentence-transformers

Your question about repository:

Please provide emails of authors

The authors of SentenceTransformers are Nils Reimers and the Ubiquitous Knowledge Processing (UKP) Lab. Their email addresses are nreimers@uni-darmstadt.de and ukplab.informatik.tu-darmstadt.de. Answer correctness: 76.9% Source link: https://github.com/UKPLab/sentence-transformers/tree/master/docs/contact.md

What this repository does do?

This repository provides a library for sentence embeddings, allowing users to represent sentences as vectors and compare them using cosine similarity. It is maintained by Nils Reimers of the Ubiquitous Knowledge Processing (UKP) Lab at Technische Universität Darmstadt. The webserver/web hosting company may collect certain log files to prevent abuse of services, but no tracking services or cookies are used to track or re-identify visitors. Answer correctness: 77.3% Source link: https://github.com/UKPLab/sentence-transformers/tree/master/docs/contact.md

Figure 31. Chatting mode of DocumentationAI
Chapter 6

Conclusion

This thesis has presented the work on the business aspects of the DocumentationAI startup project. The complete competitor analysis and business model were described, along with the principal points of the architecture and implementation of the project. Additionally, a description of the demo is provided to the reader.

6.1. Key Takeaway points

As was discovered during the market research and customer development work, question-answering problems are actual. And experts expect steady growth in the future. Taking into account the fact that AI and LLMs particularly continue to grow throughout the world and this trend does not show any indications of stopping yet. In addition to covering the basic requirements, the project has to offer new functionality and features to prospective users. Previously, it was impossible to imagine that people will search for answers to their questions via the Internet. Nowadays, it is a new normal. So why not find answers via chatbots as well?

6.2. Future work

The future work plan is the following:

- Transform MVP for the public service
  
  DocumentationAI is still not deployed. Adding user management and logging subsystem, DocumentationAI could be published.

- Share it with the IT industry
  
  It is a good idea to share across the IT industry via social media and industry communities.

- Collect user base and feedback from their
  
  The logging subsystem allows doing so. The statistics are an important factor for promotion start.

- Calculate metrics on real requests
According to the previous point, I can sell the product with real numbers of benefits.

- Plugin donat payments

In a time when DocumentationAI will be published, users can pay donations and it will be a good metric of the demand.

- Funding

Funding is a mandatory point of the growth path. Based on traction, real users, and statistics, investing money could be received.

- Develop a standalone version

There is time-consuming activity. It requires a big team and funding. But it provides a production-ready solution that could be sold very well.

- Sale and scale it

There is the main activity for making our customers happy and gaining money for further growth.

- Profit!

This outcome is expected by our investors and the team.
Bibliography

5. https://openai.com/blog/chatgpt
14. https://www.precedenceresearch.com/chatbot-market#:~:text=The%20global%20chatbot%20market%20size%20was%20accounted%20at%20USD%2084,19.29%25%20from%202023%20to%202032
20. https://github.com/features/copilot
24. https://www.diversityintech.co.uk/the-truth-about-ageism-in-tech#:~:text=The%20report%20also%20stated%20that,47%20for%20non%20tech%20managers
30. https://c4model.com/
31. https://streamlit.io/