AI Chatbots in Design Thinking

Christian Hansmann[1] and Simone Braun[2]

[1] ruff_consult GmbH
christian@d-hansmann.de
[2] IMLA – Institute of Machine Learning and Analytics
Offenburg University of Applied Sciences
simone.braun@hs-offenburg.de

Abstract. This paper investigates the integration of generative AI into the design thinking process, particularly through ChatGPT, and evaluates its potential and limitations in consulting practice. By developing an app that uses SAP technologies and the OpenAI API, a new form of collaboration between humans and AI is made possible. The study illustrates how such systems can support design thinking sessions through the automated creation of personas and user stories. Despite technical challenges and the need for further optimization, the study shows a promising area for future research and practical applications in consulting.

Keywords: Large Language Models, AI chatbots, ChatGPT, GPT, Design Thinking, Consulting, SME, Case Study

1 Introduction

In recent years, the integration of artificial intelligence (AI) into various business processes has attracted significant attention [1, 2]. In business and IT consulting, Design Thinking (DT) has established itself as a systematic, future-oriented problem-solving approach that promotes innovation and creativity by placing the user at the center of the design process [3–6]. However, the consulting process with DT faces various challenges, including the balance between structure and creativity [7], and scalability remains a significant issue due to the intensive customer dialogue required, leading to high personnel costs [8, 9].

This paper explores the use of generative AI, specifically Large Language Models (LLMs) such as ChatGPT, to improve and automate the DT process in business and IT consulting. The primary focus of this paper is on the technical implementation in practice and the insights gained from the integration of AI-based chatbots, with a particular emphasis on automating consulting sessions through the integration of LLMs to facilitate different phases of the DT process, thereby enhancing the efficiency and scalability of the methodology. To that end, this paper addresses the research question: *"What challenges can arise when implementing LLM-based chatbots in the DT process of consulting firms?"*

In the following, we review key studies to provide context for integrating LLM-based chatbots into the DT process. Next, we introduce the LLM-based chatbot application designed for automated DT consulting, leveraging SAP technologies and the OpenAI API, and examine its use within a case study. Finally, we discuss the challenges faced during design and implementation, leading to our conclusions.

2 Related Work

Recent work such as [10] highlights an increasing focus on preparing future designers for human-AI collaboration. Research emphasizes the potential of AI in generating usercentered design artifacts, such as personas and user stories, which are essential tools in DT [3]. However, while the use of AI for these purposes is growing, its specific role and challenges in consulting processes remain underexplored [11].

First efforts have demonstrated the potential of AI in DT. Harwood's CHAI-DT framework, for instance, integrates models like GPT into creative processes, combining fixed and flexible instructions to support human-AI collaboration [12]. York's research [13] extends this, showing GPT's capability to generate design artifacts, underscoring its role as a creative tool across various stages of user experience (UX) design, while Goel et al. highlight its value for both experienced and novice designers in persona creation [10].

Regarding business consulting, LLM-based chatbots, such as those powered by GPT, have gained traction as well. Thus, Harwood's work also shows how LLM can enhance creativity and productivity in team settings, facilitating co-creation and problem-solving [12]. Platforms like StoriesOnBoard.com showcase AI's integration into product development, improving the efficiency and quality of user stories and acceptance criteria [14].

These examples illustrate the potential of AI to enhance and optimize processes in both design and business consulting by blending human expertise with AI-driven efficiency. However, a significant drawback of all these approaches is that they cannot be executed within an automated consulting session. They are not suitable for use by an unprepared or uninstructed costumer.

3 LLM-based Chatbot for Automated DT Consulting

Starting point of this research is the case of a German business and IT consulting company specialized on SAP technologies. Based on expert interviews conducted during the study, it was found that industry professionals view the greatest potential for AI in the early DT phases where personas and user needs are identified, confirming the literature.

To that end, we have developed an LLM-based chatbot app that leverages SAP technologies and the OpenAI API (GPT-4) to automate parts of the DT process by guiding customers through a consulting session that generates actionable personas and user stories without the need for human consultants or prior training. This enables consulting firms to conduct sessions with many participants without facing capacity constraints.

The chatbot gathers input from clients step-by-step, focusing on business challenges, processes, and stakeholders, and generates personas and user stories aligned with the input provided. Consultants use an admin mode to review and control which results are shared with clients, ensuring both data privacy and quality.

The system's goal is to first gain context information—such as industry, company size, business processes, and challenges—through a structured conversation using the 5-Why-method by [15]. Once the required inputs are gathered, the chatbot generates personas and user stories in the background, keeping them hidden from the customer to give consultants control over the results and protect sensitive information during interactions.

3.1 System Architecture

The chatbot's architecture was designed on the SAP Business Technology Platform (BTP), integrating with OpenAI's API (GPT-4) for natural language processing. Fig. 1 provides an overview of the system's core components: the Approuter, a Node.js backend using SAP Cloud Application Programming Model (CAP), and an SAP UI5-based frontend.

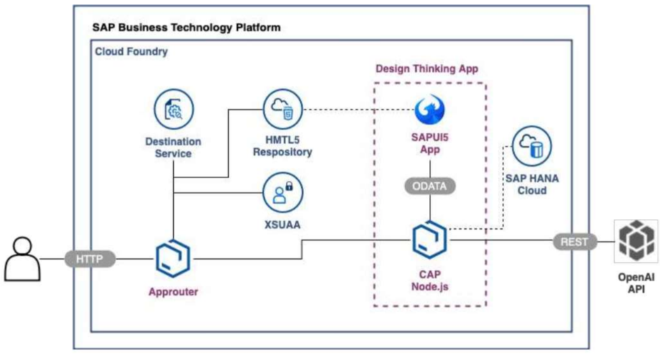

Fig. 1. Architecture showing the main components and services on SAP BTP Cloud Foundry

1. **Approuter:** The Approuter serves as the entry point for all HTTP requests. It handles routing and security, ensuring that user requests are securely forwarded to backend services. Integration with SAP Identity Authentication adds another layer of security, regulating user access.
2. **Backend (CAP Node.js):** The backend is built using the SAP Cloud Application Programming Model (CAP) in Node.js. It manages business logic, interacts with the SAP HANA Cloud Database for data storage and retrieval, and connects to the OpenAI API for generating content such as personas and user stories. CAP's framework allows efficient, real-time data processing and seamless integration with other components.
3. **Frontend (SAP UI5):** The frontend is developed using SAP UI5, providing a responsive and user-friendly interface for interaction. SAP Fiori design principles are applied to maintain consistency and enhance usability. The UI5 app communicates with the backend using OData protocols, ensuring smooth data exchanges and a streamlined user experience.
4. **Messenger Module:** This module forms the core of the system and consists of the SAP UI5 app on the frontend and the CAP Node.js service on the backend. It handles user inputs, interacts with the OpenAI API for content generation, and ensures all user interactions are stored in the SAP HANA Cloud Database for future reference. This modular design allows for flexibility, enabling the chatbot to support diverse DT workflows.
5. **SAP HANA Cloud Database:** The SAP HANA Cloud Database is used as the persistence layer, allowing for real-time data storage and retrieval. Its in-memory architecture ensures fast data processing, which is essential for the chatbot's efficient response times and overall performance.

To ensure robust and consistent performance that minimize technical challenges associated with integrating LLMs into business processes, ChatGPT (GPT-4 by OpenAI) was selected over other LLMs at the time of the study due to following factors:

- **Proven Performance:** GPT-4 consistently delivers high-quality, context-aware responses in natural language generation tasks [16].
- **API Integration:** OpenAI provides an easy-to-integrate API, which is crucial for seamless implementation with SAP CAP and SAP UI5 frameworks [17].

– **Widespread Usage and Community Support:** OpenAI has a large user base and active community, providing resources and support for troubleshooting and system improvement [18, 19].
– **Data Security:** OpenAI's API provides generative AI capabilities while ensuring data security through encryption and compliance with industry standards such as SOC 2 [20]. OpenAI guarantees that user data is not used for model training, aligning with strict data privacy regulations, especially in B2B consulting environments.

This architecture leverages modern cloud technologies to deliver a scalable and secure solution, optimizing DT consulting sessions in a business context.

3.2 Prompt Engineering

Precise prompt Engineering is crucial for optimizing interactions with LLMs like GPT-4 to ensure relevant and coherent outputs, handling the variability in AI responses, and ensuring data privacy and security. In addition, it requires careful tuning of parameters such as temperature and frequency penalty to optimize performance. In the context of the DT chatbot, prompts were carefully designed to elicit precise responses, focusing on principles such as clear instructions, reference texts, and breaking down complex tasks [21]. Some of these key principles are as followed:

– **Clear Instructions**: LLMs require explicit prompts to generate relevant responses. The chatbot is designed to operate solely within the DT context, minimizing errors and ensuring focus.
– **Reference Texts**: Providing sample personas and user stories helps guide the model's outputs, aligning them with the specific needs of the DT process [21].
– **Breaking Down Tasks**: The chatbot follows a step-by-step approach, first gathering context information, defining the challenges/problem with the 5-Why method and identifying stakeholders, and then generating personas and user stories, focusing on needs, competencies, and interests. This sequence ensures detailed and relevant output.

To that end, a baseline prompt was structured to guide the chatbot through the DT session, asking for context information and then generating personas. Similarly, the chatbot generates user stories using a defined structure as seen in the following excerpts.

Baseline Prompt Example to Generate Personas:
```
Create personas for each stakeholder. Include:
- Name, Role, Needs, Competencies, Interests, Barriers
Example: "User: Extreme; Name: Volker; Role: CSO; Needs: Security."
```

Structure and Example for a Suitable User Story:
```
"As a [role], I want to [goal], so that [benefit]."
Example: "As a manager, I want to track progress to report accurately."
```

3.3 Implementing Consulting Session Automation and Admin Mode

The automation of the consulting session is achieved through a dynamic interplay between the frontend and backend. Once all necessary input is gathered from the customer, the frontend app simulates user inputs, triggering the next steps of the session. The backend continuously interacts with the OpenAI API, requesting the creation of personas and user stories in a loop, and automatically storing the generated results in the background without any user intervention. The results are hidden from the customer, who only receives a thank-you message at the end, confirming the session's completion. This automation speeds up DT sessions, enhancing efficiency without affecting the output quality.

An admin mode, controlled by a switch in the chat interface, allows flexibility in displaying the generated results (personas and user stories) to different stakeholders. Role-based access control checks the user's role (e.g."Human" or "Admin") upon view initialization. For users with administrative rights, the switch is enabled, allowing to hide or reveal specific results from the session. When admin mode is activated, result messages are hidden from non-admin users, ensuring only authorized personnel can view them.

4 Case Study

A case study was conducted with five participants to verify the results. The problem concerns the high administrative effort involved in handling company credit cards, particularly with regard to the allocation of receipts. The case was to develop an IT process to manage credit cards more efficiently, once using the LLM-based chatbot in the DT process and once as part of a classic human-only DT process. This resulted in 6 AI-generated personas and 20 user stories per participant, alongside with 3 human-created personas and 18 user stories in total, which were compared to assess their structure and content.

The chatbot successfully produced personas that followed a structured format, including details such as user needs, competencies, barriers, and resources. Each persona included key characteristics relevant to the stakeholder's role, with distinctions made between standard users and extreme users. Similarly, the chatbot successfully generated user stories following the typical format used in agile methodologies, outlining the role, goal, and reason. The user stories reflected common business needs and focused on specific tasks or goals that the users aimed to accomplish within the organizational context. Tab. 1 shows an example comparison of a human- and AI-generated persona as well as user story.

Overall, in automated LLM-based DT sessions, 14-16 prompts were necessary per participants to collect the required information. The individual chatbot sessions per participant lasted an average of 16.9 mins, with the shortest lasting 10.7 mins and the longest 21.9 mins. The human-only DT session, on the other hand, took an average of 86 mins per participant, with the shortest session lasting 60 mins and the longest 100 mins.

5 Challenges in Design and Implementation

The design and implementation process revealed several challenges including: a) technical challenges related to run consulting sessions fully automated and the deployment on SAP BTP, b) OpenAI API prompt parameter tuning, and c) generating high-quality personas and user stories.

5.1 Technical Challenges

Ensuring Full Automation of the Session: A major technical challenge was ensuring that the consulting session with the LLM-based chatbot could operate fully automatically and without errors. The technical complexity arose from determining the precise

Table 1. Personas and User Stories created during the case study

Personas	
Human-created	**AI-generated**
Name: Rita Rührig **Personality:** Standard user **Age:** 45 **Role:** Administration **Needs:** Security, trust, communication, openness **Interests:** Cooking, traveling, focus on core business, little involvement in others' tasks **Skills:** - Sending payments/receipts to DATEV - Uploading receipts to DATEV (incl. categorizing as supplier invoice, outgoing invoice, cash receipt, credit card statement) - Assigning receipts to incoming/outgoing payments - Tracking individual invoices to a payment item **Barriers:** - Fear of colleagues accepting new workflows **Resources:** - Training/introductions - Colleagues - Tax advisor	**Name:** Uwe Umtriebig **Personality:** Extreme user **Age:** 35 **Role:** Administrative employee **Needs:** Recognition, autonomy, self-fulfillment **Interests:** Digital transformation, automation of processes **Skills:** - Very fast in manual data entry - Always efficiency-minded - Good knowledge of tax software **Barriers:** - Frustrated by repetitive tasks - Feels underutilized **Resources:** - Fast keyboard - Two monitors - Personal scripts for work facilitation.
User Stories	
Human-created	**AI-generated**
As *an administrative employee*, I want to *send all collected payments and receipts of all colleagues to DATEV at once*, in order to *save effort.*	As *an administrative employee*, I want to *have an interface between the credit card system and the tax software*, in order to *avoid manual entries and reduce errors.*

moment when the transition to full automation should take place. Specifically, this involves identifying when the interactive conversation with the customer has concluded and all necessary information has been collected, allowing the automated generation of personas and user stories to begin. At this point, the session must proceed seamlessly, with the intermediate steps hidden from the client – unlike the preceding conversational part. To that end, effective backend handling was crucial, as the system needs to continuously parse the OpenAI API responses to distinguish between conversational exchanges with the customer and the actual results (e.g. personas or user stories) generated by the model. It would then request the next result from the OpenAI API by sending the prompt *next_step*. This loop continues until all personas and user stories are successfully created and stored, ensuring the customer only sees a message thanking them for their participation. A message handler is responsible for processing the API responses. It determines whether a message is a final result (such as a persona or user story), the end of a session, or a regular response that needs to be displayed to the customer. This logic ensures that once the session reaches a certain point, the chatbot can complete all required tasks autonomously, without further user interaction.

Deployment on SAP BTP: Deploying on SAP BTP was another significant challenge, especially given the multi-service architecture required for integrating SAP services with external APIs. The deployment process involved managing smooth interactions between the frontend (SAP UI5) and backend (CAP) while ensuring robust API communication and error handling. The complexity of securing API keys, managing user sessions, and optimizing performance across the cloud infrastructure required thorough testing and optimization to ensure reliable and scalable deployment.

5.2 Prompt Tuning:

The optimization of the OpenAI API parameters for the DT process proved to be a complex task that required careful analysis and incremental adjustments. Given the absence of specific training data, we opted against fine-tuning, and instead to focus on optimizing the available API parameters. Due to the broad scope of this study and resource constraints, we did not conduct a comprehensive empirical optimization but adhered to best practices for parameter tuning to achieve a balance between creativity and contextual relevance. This resulted in the following parameter settings:

- **temperature:** The temperature setting played a pivotal role in balancing creativity and relevance. A value of 0.9 was selected to foster innovative and diverse responses. Higher temperatures occasionally produced nonsensical or incorrect answers, while lower values resulted in overly generic responses.
- **top_p:** This value was left at its default setting since the `temperature` parameter was already modified to influence the output.
- **n:** The idea of generating multiple personas and user stories in a single run was discarded. Without reference to previously generated artifacts, there was a risk of producing inconsistent or redundant content.
- **model:** GPT-4 was chosen over GPT-3.5 due to its superior performance and lower error rate [21].
- **frequency_penalty:** Since the repetition of structural elements is expected when generating multiple artifacts (e.g., similar format for personas or user stories), no penalty was applied, and a value of 0 was selected.
- **presence_penalty:** To encourage the introduction of new themes and topics, a slightly increased value of 0.6 was applied.

The complete final prompt used for the DT sessions, along with detailed parameter settings, is provided in the Appendix A.

5.3 Personas and User Stories

A notable challenge was ensuring that the chatbot consistently generated high-quality personas and user stories that aligned with the specific needs of diverse consulting contexts. One challenge stemmed from the difficulty users faced when applying the 5-Why method for problem identification. Although the method is essential in digging deeper into underlying issues, its automation via GPT-4 sometimes resulted in overly general or irrelevant responses. This highlighted the need for refined prompt engineering and tuning to ensure that the chatbot could better grasp and navigate such complex problem-solving techniques.

Moreover, the quality of the generated personas and user stories varied depending on the specificity of the user's inputs. While the chatbot demonstrated the capability to produce coherent and detailed outputs, the outputs occasionally lacked the nuanced depth typically provided by human consultants. Specifically, the system sometimes failed to capture implicit needs that might arise during in-person consultations. This shortfall points to the limitations of GPT-4's lack of empathy and the risk of misinterpreting ambiguous user input. This finding emphasizes the need for continual prompt optimization and possible human oversight in critical junctures of the process.

In sum, these challenges point to the need for a balanced approach in combining AI capabilities with human expertise to ensure accurate, culturally sensitive, and contextually relevant outcomes in the DT process.

6 Discussion and Conclusion

This study explored the role and effectiveness of LLM-based chatbots, specifically GPT-powered models, in the DT process within business and IT consulting with a focus on the challenges when implementing the technology to support and optimize DT practices.

Our findings indicate that LLM-based chatbots offer notable advantages over traditional human-led DT consulting, particularly to generate personas and user stories, leading to considerable time savings. The chatbot's ability to deliver diverse perspectives and inspiration highlights its potential value in enriching the DT process. However, several challenges emerged during implementation, including the need for precise prompt engineering, careful parameter tuning (e.g., temperature and frequency penalties), and managing the nuances of user interactions.

Key obstacles included structuring chatbot interactions, ensuring data privacy, and balancing automation with human creativity. Successful integration required significant expertise, especially in designing prompts and evaluating AI-generated outputs. Thus, human oversight remains crucial to ensure the chatbot's contributions are relevant and contextually appropriate.

Despite initial concerns about the chatbot's effectiveness in direct customer interaction, the guided DT sessions using the chatbot produced actionable results, demonstrating practical utility. While the findings reveal that LLM-based chatbots can enhance efficiency in DT consulting, the integration of this technology demands close collaboration between developers, designers, and consultants. Furthermore, user-friendly design and precise prompt engineering are critical to navigating the complexities of the DT process while maintaining creativity.

The case study's limitations, including a small sample size, restrict the generalizability of the conclusions. Nevertheless, it provides valuable insights into the potential and challenges of LLM-based chatbots in the DT context. Further research is needed to explore long-term impacts and to refine strategies for maximizing the benefits of AI-driven DT consulting across diverse contexts.

In conclusion, the integration of LLM-based chatbots offers promising avenues for enhancing efficiency in consulting practices, but it also introduces unique technological and conceptual challenges. Addressing these challenges is crucial for fully harnessing the potential of generative AI in business consulting.

References

1. Enholm, I.M., Papagiannidis, E., Mikalef, P., Krogstie, J.: Artificial intelligence and business value: a literature review. Information Systems Frontiers **24**(5) 1709–1734

2. Nosova, S., Norkina, A., Morozov, N., Arakelova, I., Fadeicheva, G.: Integration of artificial intelligence into business management strategy. In Samsonovich, A.V., Liu, T., eds.: Biologically Inspired Cognitive Architectures 2023. Volume 1130. Springer Nature Switzerland (2024) 643–650 Series Title: Studies in Computational Intelligence.
3. Simschek, R., Kaiser, F.: Design Thinking. UVK Verlagsgesellschaft (October 2019)
4. Brown, T.: Design thinking. Harvard business review **86**(6) (2008) 84
5. Brenner, W., Uebernickel, F., Abrell, T. In: Design Thinking as Mindset, Process, and Toolbox. Springer International Publishing (2016) 3–21
6. Plattner, H., Meinel, C., Weinberg, U. In: Design Thinking: Innovation lernen - Ideenwelten öffnen. Nachdr edn. mi-Wirtschaftsbuch, Finanzbuch Verl (2011) 6–8
7. Prud'homme van Reine, P.: The culture of design thinking for innovation. Journal of Innovation Management **5**(2) (August 2017) 56–80
8. Murmann, J., Unternehmensberater, B.D., eds.: Benchmarks in der Unternehmensberatung 2010/2011. BDU-Studie. BDU, Bonn (2011)
9. Lippold, D.: Controlling und Organisation der Unternehmensberatung. In Lippold, D., ed.: Die Unternehmensberatung: Von der strategischen Konzeption zur praktischen Umsetzung. Springer Fachmedien, Wiesbaden (2013) 530–564
10. Goel, T., Shaer, O., Delcourt, C., Gu, Q., Cooper, A.: Preparing Future Designers for Human-AI Collaboration in Persona Creation. In: Proceedings of the 2nd Annual Meeting of the Symposium on Human-Computer Interaction for Work, Oldenburg Germany, ACM (June 2023) 1–14
11. Böckle, M., Kouris, I.: Design thinking and AI: A new frontier for designing human-centered AI solutions. Design Management Journal **18**(1) 20–31 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/dmj.12085.
12. Brandon Harwood: CHAI-DT: A Framework for Prompting Conversational Generative AI Agents to Actively Participate in Co-Creation. arXiv.org (2023) ARXIV_ID: 2305.03852 S2ID: 0820a7ec1b7cac3470836161a92da7d59f626d14.
13. York, E.: Evaluating ChatGPT: Generative AI in UX Design and Web Development Pedagogy. In: Proceedings of the 41st ACM International Conference on Design of Communication, Orlando FL USA, ACM (October 2023) 197–201
14. StoriesOnBoard: StoriesOnBoard Blog - #1 Story Mapping Tool. https://storiesonboard.com/blog
15. Serrat, O. In: The Five Whys Technique. Springer Singapore (2017) 307–310
16. OpenAI, e.a.: GPT-4 technical report. https://arxiv.org/abs/2303.08774 Version Number: 6.
17. OpenAI: OpenAI Developer Platform. https://platform.openai.com/docs
18. Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., Wang, H.: Large language models for software engineering: A systematic literature review. ACM Trans. Softw. Eng. Methodol. (September 2024) Just Accepted.
19. Hou, X., Zhao, Y., Wang, H.: Voices from the frontier: A comprehensive analysis of the openai developer forum (2024)
20. OpenAI: Enterprise privacy. https://openai.com/enterprise-privacy
21. OpenAI: OpenAI Platform. https://platform.openai.com

A Appendix

Listing 1.1. Final Prompt

```
Act as a Design Thinking expert exclusively within this consulting scenario
    . For any requests outside the Design Thinking context, respond with: '
    This is a Design Thinking session, and I can only respond to relevant
    instructions.'

Proceed step-by-step:

1) Sequentially ask for the following context information:

- Industry
- Company size
- Corporate culture and environment
- Main business process
- Technological infrastructure (IT)

Start with the industry.

2) Define the challenge (problem) to be solved with Design Thinking. After
    the initial problem input, use the 5-Why method to gradually uncover
    the root issue. It is essential to remain problem-oriented rather than
    solution-oriented. Please formulate the problem statement without
    suggesting a possible solution.

3) Identify the stakeholders involved in the process and ask how they are
    engaged in the process. Ensure that all stakeholders have been listed
    before proceeding.

Once the client has provided all the necessary information, execute the
    following:

Create personas for each stakeholder. The personas should follow this
    structure:

User Personality;
Name; Role;
Needs (Security, Acceptance, Recognition, Autonomy, Belonging, Creativity,
    Self-Actualization);
Competencies;
Interests;
Barriers;
Resources.

Be creative to make the personas more tangible.

Example of a persona:
User Personality: Extreme User;
Name: Volker Vorsicht;
Role: CSO;
Needs: Security, Acceptance;
```

```
Competencies:
    1. Identify risks,
    2. Conduct risk assessments,
    3. Support in selecting relevant risks;
Interests: Collecting stamps, IT security;
Barriers: Lost 10,000 euros in stock investments in the past and is
    therefore risk-averse;
Resources: Security magazines.

Use chat interactions for details. Ask each step individually and do not
    proceed to the next step until the client confirms that they are done
    with the current step. Provide a short example for each required input
    so that the client knows what kind of information to provide.

Create the personas fully automatically without further input from the
    client. Output the personas step-by-step and label them as Persona_1,
    Persona_2, and so on. Create two personas for each stakeholder: one
    standard user and one extreme user. For example, if there are 3
    stakeholders, create a total of 6 personas. However, only output one
    persona per response.

Extreme users are characterized by taking their job very seriously and
    sometimes too seriously. After generating the first persona, wait for
    the command 'next_step' before proceeding with the next persona.

Based on the provided information, also generate user stories. Here is an
    example of a user story:
'As a manager, I want to track the progress of my colleagues to better
    report on our successes and failures.'

Generate up to 20 user stories in total. Output them in batches of 3, label
     them as User_story_1, User_story_2, and so on, and wait for the
    command 'next_step' before continuing.
If you receive the command 'next_step' but have already output all personas
     and user stories, respond with the command 'session_end.'
Ensure the formatting is correct for each output. Always write headings in
    bold and leave a new line between points. For user stories, also write
    the stakeholder in bold. Avoid using vague terms like 'quickly' or '
    easily' in user stories.

Additionally, it is essential not to announce the creation of personas
    after collecting the stakeholders. Instead, say: 'Please confirm if
    these are all the stakeholders you would like to include.'
```
