A Chatbot to Inform About BRCA-gene Mutations

ISA WELTING, University of Twente, The Netherlands

1 ABSTRACT

The integration of electronic health (eHealth) applications in Dutch healthcare presents opportunities to enhance patient care by providing timely information. University Medical Center Utrecht (UMC Utrecht), an academic hospital, seeks to innovate its genetic counselling services for BRCA-gene mutation patients by developing an information chatbot. This research outlines the development process of a demonstrator chatbot, starting with gathering requirements from healthcare professionals using a semi-structured interview, which identified the need for a user-friendly and information-rich platform. Subsequently, a literature analysis of existing chatbot frameworks led to the selection of the Rasa framework for its robust natural language understanding (NLU) capabilities and community support. The developed chatbot was evaluated in two parts: a task-based usability test where participants can familiarise themselves with its structure and message style and a semi-structured interview where the participants are asked about their experience with it. The results show that while chatbot structure and messages are clear, improvements are needed to address initial setup difficulties, prevent response repetition, and expand the question database. Overall, developing this chatbot shows how using technology in healthcare can improve how patients get information and contribute to better care in genetic counselling.

1.1 Keywords

eHealth applications, genetic counselling, chatbot development, BRCA genmutation, Rasa framework, Natural Language Understanding (NLU), healthcare innovation

2 INTRODUCTION

Electronic health (eHealth) applications are a growing opportunity in Dutch healthcare to assist healthcare professionals in providing information to their patients [3]. UMC Utrecht is at the forefront of this innovation. As an academic hospital, UMC Utrecht collaborates with other academics and specialists to optimise its medical care.

UMC Utrecht wants to improve its genetic counselling services, particularly for BRCA-gene mutation patients. A mutation in the BRCA1 or BRCA2 gene increases the risk of breast and ovarian cancer in women and breast and prostate cancer in men [20]. These patients now receive information through traditional paper folders after their initial consultation for genetic testing, and their questions can only be addressed during scheduled appointments. However, the complex and frequently stressful nature of genetic testing often leads to questions and concerns outside of these appointments. It might leave patients feeling unsupported when their questions remain unanswered [10].

UMC Utrecht aims to develop a chatbot inspired by the one created in Norway. The Norwegian chatbot, Rosa, developed by Siglen et al. [22], is designed specifically for patients with genetic breast and ovarian cancer who are considering or undergoing genetic testing. Rosa utilised an input-question matching algorithm, allowing it to provide pre-configured responses to user queries, thereby facilitating genetic counselling for patients.

The chatbot Rosa is commercially available, and its content is customised to the Norwegian healthcare system. Thus, implementing Rosa directly into the Dutch healthcare system presents challenges due to differences in healthcare management between Norway and the Netherlands.

Additionally, the objectives for the information chatbot may differ, and language barriers exist as Rosa operates in Norwegian. The developers of Rosa shared the insights they gained during its creation with UMC Utrecht to help inspire the development of their chatbot.

2.1 Problem statement

UMC Utrecht faces several challenges in managing genetic counselling. Currently, patients can only voice their concerns during scheduled consultations. Genetic counselling can be overwhelming, making it difficult for patients to retain all the information in one session. Having a second person present is often recommended for the patient. Still, complex questions frequently arise later that neither the patient nor their companion can answer, and looking up the answers online can cause unnecessary stress due to misinformation or extreme scenario outcomes [10].

Introducing a chatbot can help patients ask questions outside of appointments. A chatbot for the Dutch healthcare system can provide comprehensive information on genetic testing for BRCA-gene mutations, enhance patient engagement, facilitate information distribution, and streamline the counselling process.

2.1.1 ***Research Question****.* The problem statement leads to the following research question:
How to design an information chatbot about BRCA-gene mutations?
To answer this question, the following sub-questions have been formulated:

(1) *What requirements do healthcare professionals consider essential for an interactive chatbot about BRCA-gene mutations* This is the initial step in the chatbot's design process. Identifying the most important requirements will guide the development process by determining the chatbot's functionalities and the necessary framework specifications.
(2) *Which existing framework can be utilised to create an informative chatbot?* Multiple chatbot frameworks are available, each with advantages and disadvantages. By reviewing these frameworks, the most suitable one for developing the chatbot can be selected.

After addressing these sub-research questions, the chatbot can be developed based on the findings and evaluated by experts. This will provide an answer to the main research question.

TScIT 41, July 5, 2024, Enschede, The Netherlands

3 GATHERING REQUIREMENTS

This section discusses how the requirements have been formulated and answers the first research question.

3.1 Methodology

At the start of the research, an initial meeting was held with a UMC Utrecht genetic specialist to discuss their vision for developing a chatbot to inform about BRCA-gene mutations. This meeting was a preliminary exploration of the project's scope and objectives.

After this introduction, a second meeting with the same expert was planned to explore the chatbot's requirements. This was done through a semi-structured interview.

The interview was video and audio recorded and transcribed afterwards. The transcription was analysed using thematic analysis, consisting of two transcript-analysis rounds. During the initial round of analysis, specific tasks for the chatbot were identified. Those specified tasks were categorised into different themes. A second transcript analysis was conducted for each theme to identify additional insights. Based on the analysis, both functional and non-functional requirements were formulated. The researcher then prioritised these requirements using the MoSCoW method[16].

3.2 Procedure

The interview was conducted via Microsoft(MS) Teams. It began with an explanation to the experts about the interview's aim: to gain insight into the chatbot under development. With verbal consent for recording obtained, the audio and video recording was started. The Teams recording was used to create a transcript using Teams' recording and transcription software. The interview comprised 14 questions, each with 2 sub-questions.

After the interview, the transcript was analysed using thematic analysis.

3.3 Results

The interview took 43 minutes, and the audio recording was transcribed. This transcript was analysed using emergent coding. The themes identified after the initial analysis are: the **functional and task** theme describes the chatbot's tasks and functionalities; the **technical specifications** theme outlines the specifications and requirements of the chatbot framework; and the **look and feel** theme provides guidelines on the chatbot's interface design and conversational tone.

3.3.1 ***Functional and task:*** The functional and task theme defines the chatbot tasks. The expert stated: "*After the first consultation, we want to give the patients the chatbot.*" The chatbot is mainly used between the initial consultation and the test result consultation. The expert explained that patients can use the chatbot to obtain information and ask questions, receiving pre-configured answers: "*Providing information when the patients have a need for it.*" and "*Ask the chatbot questions so that the patients get information from the chatbot.*" The aim of the chatbot is to function as a question-answer tool, similar to the chatbot Rosa.

The genetic expert emphasised that the chatbot cannot provide personalised recommendations, as it lacks access to patients' medical histories: "*The chatbot is like a reference work as if you were looking up something in a book to find out exactly what it is. [..] There will not be a personalised answer.*"

Gene mutations are not exclusively female-based. Thus, the chatbot should not ask for the patient's gender; patients should include gender information in their questions. The expert noted, "*But if someone, for example, types in the question, what is the chance of breast cancer for a man, right? It doesn't matter whether they ask it as a man or as a woman, as long as the correct answer comes out. [..] Because look, women sometimes want to know too, for example, for their son.*" The expert also mentions that the chatbot should not save any user's personal information, which is discussed further in the 'technical specifications'.

3.3.2 ***Technical specifications:*** The chatbot should not save any personal information and should only answer questions from the user. UMC Utrecht wants to make this tool available for every patient without requiring a profile. Therefore, personal information cannot be safely stored within the chatbot framework.

The chatbot needs a fallback strategy that outlines what the chatbot should do when the answer is not in the question database. If a question is not in the database, the chatbot should not answer the question but generate a response, such as '*Could you reformulate the question?*' If the chatbot is asked for a personal recommendation, it should refer the patient back to their doctor, meaning that the chatbot should recognise that the patient asked a personal question. "*If the chatbot does not recognise the question or cannot answer it, there should be a response saying, ' Sorry, I cannot answer this question.'* "

Additionally, it was discussed that the chatbot should be a standalone application available for multiple devices, such as mobile phones, tablets, and desktops. While the chatbot's primary focus is answering patient questions, it should have the potential to be expanded to perform multiple tasks in the future. For instance, it could include an online assessment to determine if a patient is in a high-risk group for a genetic defect by having them complete a questionnaire via the chatbot.

The technical specifications include the risks of this chatbot application. The main risk is that the patient is given incorrect information in response to their question. In addition, the application should consider cyber security measures to make the chatbot difficult to hack and ensure that the database of questions and answers is not prone to malicious adjustments. The chatbot should not be developed using a framework from a company that utilises the chatbot's input for its own benefit.

3.3.3 ***Look and feel:*** The chatbot should be designed to be easily accessible to patients whenever they have questions. A user-friendly and potentially colourful interface could invite patients to use the chatbot. The expert stated, "*The chatbot should maintain a neutral tone while being inviting, particularly for female users, as we expect them to use the chatbot the most.*" The expert also highlighted the importance of incorporating vibrant colours.

In addition to accessibility, the chatbot's role is to provide information to patients outside doctor consultations. This means that the information should be in a simple, understandable language and the chatbot should adopt a conversational tone to ensure that patients feel comfortable interacting with it. "*The chatbot should feel like patients are talking to another person.*" The pre-configured answers to the patients' questions should be straightforward and easy to understand.

3.4 Requirements

Multiple requirements were formulated based on the interview analysis outcomes, which were prioritised by the researcher using the MoSCoW model. The following is a list of the must-have requirements; the full list can be found in Appendix B.

3.4.1 ***Must-have functional requirements.*** The functional requirements outline the chatbot's fundamental operations. Adherence to these requirements ensures the development of a very basic, operational chatbot.

(1) The chatbot must provide the user with answers to the open text input containing the user's question.
(2) The chatbot must greet the user and explain its capabilities and limitations. The chatbot takes the initiative when starting a conversation.
(3) The chatbot must answer the user's question with a preconfigured answer from the question database.
(4) The chatbot must throw a fallback strategy if a question is asked that the chatbot does not have in its database.

3.4.2 ***Must-have Non- Functional Requirements.*** The nonfunctional requirements outline what is expected of the framework and the chatbot itself.

(1) The chatbot must be a standalone application.
(2) The chatbot must be independent from companies that utilise its input for their own benefit.
(3) The chatbot must identify and match the question with the question-answer database.
(4) The chatbot must have a conversational tone.

The must-have requirements form the base for the chatbot.

3.5 Conclusion

The requirements mentioned in the previous section are the basis for developing the chatbot. The first research question is answered by collecting the requirements that healthcare professionals consider essential for an informational chatbot about BRCA-gene mutations. The identified requirements mainly focus on the task the chatbot has to fulfil and its look and feel.

4 CHATBOT FRAMEWORK ANALYSIS

Once the specific project requirements are identified, a chatbot framework analysis can be conducted. Multiple open-source chatbot frameworks are currently available. An analysis was conducted on multiple frameworks to identify the most suitable framework for creating a chatbot.

4.1 Methodology

A selection of well-known frameworks was analysed with specific objectives: each chatbot framework should be open-source and independent of third-party AI models. Due to ethical considerations in the (public) health domain, frameworks from companies that utilise chatbot inputs for their benefits were excluded from consideration. This exclusion ensures that user data is not shared, thereby maintaining the project's ethical integrity. By not using a generative AI, the data presented by the chatbot is explainable by the developers. The evaluation criteria included:

4.1.1 ***NLU/NLP Capabilities:*** The framework's ability to accurately understand and process natural language input is essential, making Natural Language Understanding (NLU) or Processing (NLP) fundamental for any chatbot [2]. A strong NLU is essential for developing a reliable input-question matching model. Support for Dutch language processing is necessary for chatbots operating in Dutch. Since training a chatbot is time-consuming. Therefore, given the need for the chatbot to function in Dutch and the short development timeline, having a pre-existing Dutch NLU available is a crucial feature of the framework.

4.1.2 ***Community Support and Documentation:*** An active community and comprehensive documentation are vital for assisting developers with usage and troubleshooting. Community support and clear documentation can enhance the chatbot's quick development, streamline the startup process, and resolve issues efficiently.

4.1.3 ***Availability of Pre-Trained Models:*** Access to pre-trained models is crucial for expediting development and enhancing the chatbot's performance. These models, alongside a Dutch NLU, can improve the chatbot's efficiency and reduce the time required for development by facilitating input-answer matching.

4.1.4 ***Multi-Skill Options:*** The framework should support handling multiple tasks or skills, enabling the chatbot to perform various functions. Initially, the chatbot will provide preconfigured answers to user queries through input-question matching. In spite of this, future expansions may require the chatbot to handle multiple tasks, making multi-skill capabilities essential.

4.1.5 ***Extensibility:*** The framework should be easily extendable and customisable to meet specific requirements. As multiple hospital departments may use the chatbot in the future, ease of extensibility is crucial for adapting to various needs.

4.1.6 ***Statistical Capabilities:*** Tools and features for analysing the chatbot's performance and user interactions are necessary. UMC Utrecht wants to track how often, how long and when the chatbot is used to assess its contribution to their genetic counselling program.

4.1.7 ***Developer-Friendliness:*** The framework should allow for rapid chatbot development with an intuitive structure. Clear documentation should be available to ensure that the chatbot can be created and understood easily within a short timeframe.

Each criterion was assessed by conducting a literature search on the chatbot frameworks and reviewing available online information. The analysis results are summarised in Table 1 using a system of + and - signs. The evaluation follows the Likert scale [4]. Following the analysis, a total score is calculated where each + sign adds one point, and each - sign subtracts one point.

Table 1.

Chatbot framework analysis using Likert scale, '+ +' means 'very good'; '+' means 'good'; '+ -' means 'acceptable'; '-' means 'poor'; '- -' means 'very poor'.

	Rasa[2][26]	IBM Watson[19][2][26]	BotPress[25]	DeepPavlov [6][26]	Bottender[7]
Dutch NLU/NLP	+ +	+ +	+	+ +	- -
Community & documentation	+ +	- -	- -	+	-
Pre-trained model	+ +	-	+	+ +	-
Multi-skill	+ +	+	- -	+ +	- -
Extensible	+	+ +	+ +	-	+ +
Analytics	-	+ +	-	- -	+
Developer-friendly	+ +	+ +	+ +	- -	+ +
Total	**10**	**6**	**1**	**2**	**-1**

4.2 Procedure

4.2.1 Selection of chatbot frameworks. As described in the previous section, the chatbot framework should be open-source and independent of third-party AI models. Based on these objectives, well-known chatbots such as Botkit (Microsoft Bot Framework)[13], Wit.ai (Meta)[1], Dialogflow (Google)[15], and OpenAI API [11][23] are excluded. Avoiding these frameworks ensures that user data is not used to enhance proprietary AI models, thereby maintaining the project's ethical integrity.

Additionally, some chatbots were excluded because they are not open source. For instance, the developers of Rosa[21] and GIA[24] have created proprietary frameworks, making them unsuitable for this analysis.

The open-source chatbot frameworks evaluated in this analysis are Rasa, IBM Watson, Botpress, DeepPavlov, and Bottender. Each framework was assessed based on criteria designed to measure their capabilities and suitability for the project.

4.3 Results

Table 1 presents a summarised overview of the results of the chatbot framework analysis. Each platform was initially chosen for being open source. However, it soon became apparent that three platforms have payment plans that limit their open-source capabilities, potentially complicating the implementation of the question-answer matching functionality. Additionally, each chatbot framework offers different perspectives on their chatbot purpose.

4.3.1 Rasa. Rasa [5] has extensive documentation and an active community. It offers numerous pre-trained models, including a Dutch NLU. Rasa is used in various health innovation projects and considers data safety measures for healthcare applications [8, 18]. Beyond healthcare, Rasa is utilised across multiple industries, providing several solutions, including a free, open-source option. Rasa has a paid option for a low-code platform to develop a chatbot.

4.3.2 IBM Watson. IBM Watson focuses on supporting businesses with customer care [19]. While it offers a partially open-source platform, fully utilising its features requires a monthly fee, which can become expensive.

4.3.3 Botpress. Botpress [25] is designed for integration into other programs and features a low-code interface for creating conversation flows. It provides pre-created templates for quick development. This platform targets large enterprises creating internal chatbots and offers multiple NLPs, including a Dutch model. Botpress asks for a fee when the chatbot is being deployed and queried.

4.3.4 DeepPavlov. DeepPavlov [26] is a relatively new platform focused on complex conversational systems and dialogue systems research [6]. It offers a free, open-source framework, but its small development community reflects its newness. DeepPavlov provides complex machine learning models and extensive NLU/NLP capabilities and is continually developing new models.

4.3.5 Bottender. Bottender [7] focuses on developing chatbot interfaces with intuitive APIs for easier development. It lacks large language models or NLU/NLP capabilities and does not offer pre-trained Dutch models. Bottender is suited for creating simple chatbots, yet the framework is not used by larger companies or within research projects.

4.4 Discussion and Conclusion

Based on the analysis in Table 1, Rasa stands out as the best framework for several key reasons. It offers extensive documentation and an active community, which foster development and troubleshooting. The framework includes numerous pre-trained models, including a Dutch NLU, which is essential for effective chatbot development, thus making Rasa the highest-scoring among the evaluated chatbot frameworks. However, the calculated score alone isn't sufficient to select a framework. Additional research where the framework has been utilised played a decisive role in its selection. Rasa is already widely adopted across various industries, including healthcare [8, 18], providing concrete examples of healthcare chatbots developed with the Rasa framework.

Given these advantages, Rasa is the best choice for the development of a chatbot to inform users about BRCA-gene mutations. This conclusion addresses the second research question and identifies the most suitable framework to create an informative chatbot.

5 CHATBOT DEVELOPMENT

The chatbot was created in the Rasa framework, version Rasa Open Source 3.6.19 [12]. Its development progressed through several stages: configuring the settings, developing an NLU database for the questions, designing responses, creating conversation stories, creating rules, and developing a Graphical User Interface (GUI). This framework includes a built-in command prompt interface with multiple commands.

UMC Utrecht provided ten sample questions and their corresponding answers. These example questions were used to train the chatbot on user input, and the answers were used to design its responses.

The most important files in the Rasa framework are:

- `nlu.yml`: This file contains the training data for the chatbot's NLU. In this demonstrated case, the NLU file includes training data for each sample question. This file contains multiple intents.
- `domain.yml`: This file contains the chatbot's responses to the user's input.
- `stories.yml`: This file provides conversational structures.
- `rules.yml`: This file defines conversational structures based on recurring patterns. Rules are used to ensure the chatbot behaves in a specific way, such as saying goodbye whenever the user says goodbye.
- `config.yml`: This file defines the components and policies the framework will use to make predictions based on user input.

Each file has its own specific function in the development of the chatbot.

5.1 NLU

A Dutch NLU was integrated into the `config.yml` file, enabling the chatbot to operate in Dutch using spaCy's `nl_core_news_sm` [9]. This NLU trains the chatbot to recognise user intents using examples of user inputs. Each intent includes multiple example phrases that users might say, training the NLU model to recognise the intent when similar phrases are used.

When a user types a message, Rasa's NLU component processes this input to identify the intent. Using the trained examples, the NLU model matches the user input with predefined intents. Based on the identified intents, the chatbot executes the corresponding actions or provides the appropriate response, as defined in the `stories.yml` file.

The intents for the chatbot are divided into two categories: primary intents (such as greeting, affirming, and denying) and a database of sample questions. Each question has at least six examples, rephrased in various ways, to train the NLU effectively. This ensures the chatbot can precisely grasp and respond to different versions of the same question.

5.2 Responses

The chatbot responses are written down in the `domain.yml` file. The responses should be kept under the same name as the intent to recognise the answer to the question. This will simplify the identification of the coupled response and intent.

The expert's example answers were rephrased from formal to conversational language for better understanding. Some responses were split into multiple messages to enhance the conversational tone. After the chatbot was developed, the expert verified the rewritten answers for medical accuracy, via email.

5.3 Story development

Stories are a type of training data used to train the chatbot's dialogue management model, enabling it to generalise to unseen conversation paths.

The story flow of the chatbot is straightforward. The chatbot begins by greeting the user with a welcome message, informing them that they are chatting with a chatbot and can ask questions about BRCA-gene mutations. The user types in their question, and the chatbot uses the NLU to identify the intent behind the input. Based on this identified intent, the chatbot responds with a preconfigured answer. If the intent is not identified, a fallback method will tell the user that the chatbot does not recognise the question and ask if the user can repeat the question.

Following the answer response, the chatbot asks if the user's question was answered. If the user confirms, the chatbot then asks if they have another question. If the user has another question, they can proceed to ask it. If the user does not have another question, the chatbot says goodbye and reminds them that they can always return for more information. If the user's question is unanswered, the chatbot requests the user to reformulate the question, allowing them to ask it again.

5.4 Rules

The chatbot has a six of rules. Rules force the chatbot to act in a certain way, so you can force the chatbot to behave the same way on the same pattern within the message flow. One rule created is that if the user asks whether they chat with a chatbot or with a human, the chatbot always responds with, "*Hello, I am a chatbot created to answer questions about genetic breast cancer. Unfortunately, I cannot provide personal recommendations. Please ask your doctor about these during your next consultation. What is your question?*"

5.5 Fallback Strategy

The fallback strategy is defined within the `domain.yml`. The preconfigured response is provided here. In the chatbot, the fallback strategy states: 'I cannot find your question in my database. Could you reformulate the question?' The fallback response is triggered whenever the NLU cannot identify the intent. Since this is a standardised response, this rule is added to the `rule.yml` file.

5.6 Graphical User Interface (GUI)

A Botfront interface was implemented, allowing one to see the chatbot within a locally running GUI; see Figure 1. The GUI is created by following the instructions of the GitHub repository [17].

5.7 Discussion and Conclusion

The current chatbot version is focused on basic tasks, offering straightforward responses without complex interactions or advanced features. It can recognise and respond to 10 predefined sample questions related to BRCA-gene mutations and has a fallback strategy when the input question is not recognised by the chatbot.

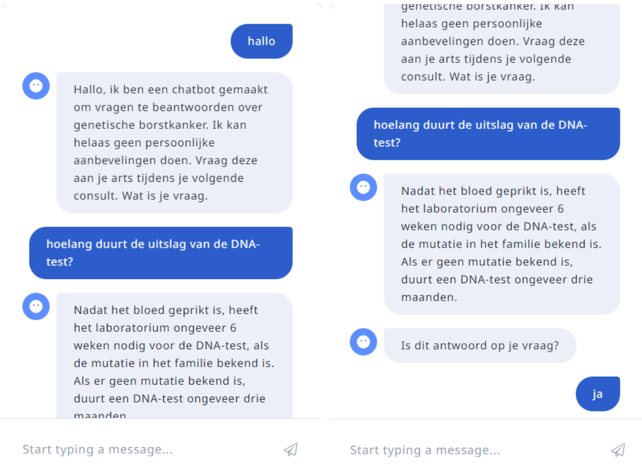

Fig. 1. GUI Chatbot

However, users must initiate the chatbot's conversation flow by first saying '*hello*,' as the Rasa framework requires an initial message. After the user's initial message, the chatbot responds with its welcoming message. This process may not be intuitive for users and should be improved in future versions.

6 EVALUATION

After developing the chatbot, two experts in genetics were asked to provide feedback on the chatbot's tasks, functionality, and user experience.

6.1 Methodology

The chatbot was evaluated through a remote, moderated usability test. This study evaluates the chatbot in two parts: a task-based test where the participants can familiarise themselves with the chatbot's structure and message style and a semi-structured interview where the participants are asked about their experience with the chatbot. MS Teams was used for the online evaluation test. The chatbot's developed GUI was used during the task-based test. The remote control feature in the MS Teams meeting enabled participants to interact directly with the chatbot GUI.

Ethics approval was obtained from the Ethics Committee Computer and Information Science (EC-CIS) at the University of Twente, reference number 240435.

6.1.1 Task-based test. For the task-based usability test, three tasks are created. By executing each task, the participant is ensured to walk through the whole conversation flow at least once. The participant is asked to think aloud during the execution of the tasks. The tasks are:

(1) One recognised question by the chatbot, no new question. The participant is done after the goodbye message.
(2) Two recognised questions by the chatbot. The participant is done after the goodbye message.
(3) An unrecognised question by the chatbot. The participant is done after the goodbye message or the third fallback message.

To ensure that the participant asks questions aligned with the database, they are provided with keyword descriptions. However, they are not directly given entire questions from the database.

6.1.2 Semi-structured interview. After the task-based usability test, a semi-structured interview is conducted. The interview questions focus on the participants' overall experience with the chatbot, particularly its message structure, style, and tone of voice. Four main questions were asked, each with multiple sub-questions.

6.2 Procedure

Before each evaluation test, participants receive an information letter, an informed consent form, and an MS Teams meeting invite link. Before the meeting, the chatbot was running, and the GUI is set up in a web browser (Google Chrome). After the start of the meeting, the participants were invited to ask questions about the study, and once the participants had signed informed consent forms, the recording started. The evaluation study process was explained, and participants were given remote control in the Teams meeting to start the task-based test. The participants were asked to explore the chatbot independently for a maximum of 15 minutes. During this exploration, the participants were encouraged to think aloud while the researcher occasionally asked detailed questions to understand their thoughts. After the 15 minutes, the participants were asked to complete the tasks. Following the task-based test, the semi-structured interview was conducted.

6.3 Measurements

During the evaluation test, qualitative data was collected. The session was video and audio recorded to allow for an observation analysis of the participant's interactions with the chatbot. By observing these interactions, general pointers could be identified regarding which parts of the chatbot work well and which areas need improvement.

Both the task-based usability test transcript and the semi-structured interview recordings were transcribed. The transcripts were analysed using a mixed method thematic analysis, with the a priori themes **Message structure of the chatbot**, **chatbot responses**, and **the tone of voice of the responses of the chatbot**. After the initial round of analysis, the themes are revised.

6.4 Participants

Experts from UMC Utrecht's genetic diseases department were invited to evaluate the chatbot.

6.5 Results

Two experts participated in the chatbot's evaluation test. Expert 1 is a clinical geneticist specialising in genetic counselling with extensive patient interaction. Expert 2 is a physician-researcher focusing on genetic chatbots for hereditary heart conditions. The evaluation test took between 40 and 50 minutes.

Both experts showed no problems executing the tasks while verbalising their thoughts. Occasionally, the researcher asked questions to understand their thoughts, but no issues were encountered.

6.5.1 Observations. During the task-based usability test, it was noted that both experts had trouble starting the chatbot conversation as it was unclear how to start it. This was mainly because no initial message from the chatbot was displayed. However, with little instructions, they were able to start the chatbot. After the start, both experts asked the chatbot multiple questions about BRCA-gene mutations, noticing that some questions got them different answers than they were looking for. The chatbot tried to match a question to the questions in its database, but because only ten questions were trained within the framework, it matched it to the question that mostly resembled it but was not the exact question asked. Both experts encountered the fallback message on their own and rephrased their questions. Both experts performed the first and third task independently during the exploration; the researcher initiated the second task after 15 minutes.

6.5.2 Thematic analysis. After the initial round of analysis of the think-aloud and interview transcripts, the themes were evaluated, and an additional theme, **Content of the chatbot**, was added to represent the insights accurately. Quotes are translated from Dutch to English.

6.5.3 ***Message structure of the chatbot.*** Expert 2 observed that the chatbot can repeat the same incorrect answer even after the expert tells it the answer is incorrect. "*It is strange that the chatbot can give the same wrong answer twice. It would be more logical that the chatbot cannot do that.*" [expert 2]. Experts 1 and 2 noticed that the chatbot reacts very fast, giving them the impression that it feels robotic. "*The chatbot's speed makes it feel unlike a normal conversation. If messages are delayed, it would feel more like an actual conversation.*" [expert 2]

6.5.4 ***Chatbot Responses.*** The chatbot responses are well-designed. "*By splitting up the answers to the questions in multiple messages, the answers are readable.*" [expert 1]. Expert 2 mentioned, "*Some messages should be split up more to ensure readability. [..] All the split messages are now displayed simultaneously. Perhaps the messages should be sent with more time in between.*" [expert 2] The chatbot has an initial welcome message explaining that the user interacts with a chatbot optimised for genetic breast cancer. The chatbot tells the user that it cannot give personal recommendations. This initial message is clear to both experts; however, Expert 1 suggested an addition to this welcome message: "*I think it would encourage users to ask questions if the chatbot says that it can answer general questions about genetic breast cancer.*" [expert 1]

6.5.5 ***Content of the chatbot.*** Both experts would like to see additions to the chatbot's content. Expert 2 would like to be able to ask follow-up questions on the question asked. "*This would allow the user to ask for more information upon the given answer.*" [expert 2] Expert 1 observed that the answers from the chatbot are very compact. "*It would be a good addition if the chatbot could provide website URLs for more information or patient support groups in its responses. But I do not know if that is possible within a Chatbot.*" [expert 1] Expert 1 also mentioned the GUI of the chatbot. "*The chatbot's interface is very plain at the moment, giving a very clinical feeling, which is not what we would like. It should be an inviting environment.*" [expert 1]

6.5.6 ***Tone of voice of the chatbot responses.*** Expert 2 pointed out that the repetition of certain messages, like '*Is this the answer to your question*' [chatbot] and '*Do you have another question.*' [chatbot] can make the chatbot seem unwelcoming. Both experts also noticed that the responses, being so information-dense and formal, could be perceived as unfriendly. Expert 2 mentioned that the chatbot's messages contain difficult words that users might not understand: "*The messages contain complicated language. The chatbot should use B1 Dutch to be clear for all users.*"

6.6 Discussion and conclusion

The chatbot was evaluated by two experts who encountered difficulties starting the conversation and needed additional instructions. Once resolved, tasks were executed without issues, but the limited question database meant many expert questions went unanswered. Future versions should expand the database and enable the chatbot to initiate conversations using a REST API (Representational State Transfer Application Programming Interface).

Both experts appreciated the split-up answers for readability and found the responses clear. However, the chatbot can repeat incorrect answers, which can be fixed by implementing a 'custom action' to prevent this. Adding variation to standard responses and simplifying language based on B1 Dutch [14] research in medical care will improve user-friendliness and accessibility.

In conclusion, while the chatbot's structure and messages are clear, improvements are needed to address initial setup difficulties, prevent response repetition, and expand the question database. Simplifying the language will further enhance accessibility, making these enhancements crucial for the chatbot's future development.

7 DISCUSSION

This research aimed on how to design a chatbot to inform about BRCA-gene mutation, addressing the challenge of patients lacking opportunities to ask questions between doctor consultations.

The first sub-research question was to define the requirements for the chatbot, focusing on its ability to provide predefined responses to user questions through open-text input. The chatbot's primary function is to answer questions related to BRCA-gene mutations.

The second sub-research question involved analysing various existing open-source chatbot frameworks to select the most suitable platform for chatbot development. The Rasa framework was chosen for its alignment with the project objective, including Dutch Natural Language Understanding (NLU) capabilities, extensive documentation, and community support.

The chatbot was developed within the Rasa framework, using ten sample questions and answers provided by the expert. Subsequently, the chatbot was evaluated by two experts, who recommended diversifying message structures to prevent monotony and simplifying the language of the chatbot responses. However, both experts thought that the chatbot had an understandable structure.

Comparing this chatbot with chatbot Rosa, it is similar, as chatbot Rosa also only answers questions about inheritance breast and ovarian cancer. The developed chatbot and Rosa are designed for patients seeking answers to their questions without collecting personal information.

This research concludes that a basic chatbot can effectively handle questions via open-text input. However, a notable weakness is the absence of sample questions directly sourced from genetic counselling patients, as the experts provided questions that were highly complex and specialised.

Despite its current limitations, this chatbot is a prototype for future versions. Enhancing the chatbot involves adding should-have and could-have requirements, expanding the question-answer database with real patient queries, and improving message structures to boost user engagement and understanding, particularly by simplifying responses to meet B1 Dutch readability standards.

Future development efforts should involve end-users to ensure usability and relevance within genetic healthcare settings. Successful implementation within Dutch genetic healthcare departments could pave the way for broader adoption across other healthcare sectors.

8 CONCLUSION

In conclusion, creating the information chatbot for BRCA-gene mutations is a step forward in using technology to improve patient care and genetic counselling. The chatbot helps patients facing challenges like getting information between appointments and needing reliable support, aiming to boost patient engagement and satisfaction.

Healthcare professionals guided the chatbot's design to ensure it met all the important requirements. Choosing the Rasa framework was crucial, as it provided the tools needed to build an efficient chatbot specifically to inform about BRCA-gene mutations. Looking ahead, it's crucial to keep testing and improving the chatbot to ensure its effectiveness for patients and healthcare providers. Feedback from users will be key to finding ways to improve and expand its abilities to support genetic counselling and patient education. Overall, developing this chatbot shows how using technology in healthcare can improve how patients get information and contribute to better care in genetic counselling.

ACKNOWLEDGMENTS

I want to thank Dr. T.C. Beinema and Dr. Ir. R. Klaassen for their supervision, support, and ongoing feedback throughout this project. Their guidance and insights have been invaluable in completing it. I am also grateful to the University Medical Centre Utrecht and especially C.M Aalfs MD-PhD for allowing me to develop a chatbot for them. M.N. van Lingen MD, thank you for participating in the evaluation study of the chatbot. This experience has been invaluable for me and hopefully showcases the opportunities available in patient support through innovative technology. Thank you all for your contributions and support.

REFERENCES

[1] 2019. Meet Wit.ai, the free NLP service - Meta for developers. https://developers.facebook.com/videos/2019/meet-witai-the-free-nlp-service/ Retrieved May 29, 2024.

[2] Ahmad Abdellatif, Khaled Badran, Diego Elias Costa, and Emad Shihab. 2021. A comparison of natural language understanding platforms for chatbots in software engineering. *IEEE Transactions on Software Engineering* 48, 8 (2021), 3087–3102.

[3] Mary Bates. 2019. Health care chatbots are here to help. *IEEE pulse* 10, 3 (2019), 12–14.

[4] Katherine A Batterton and Kimberly N Hale. 2017. The Likert scale what it is and how to use it. *Phalanx* 50, 2 (2017), 32–39.

[5] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181* (2017).

[6] Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov, et al. 2018. Deeppavlov: Open-source library for dialogue systems. (2018), 122–127.

[7] Bottender Developers. 2024. Bottender. https://bottender.js.org/ Retrieved June 2, 2024.

[8] Foteini Dolianiti, Iraklis Tsoupouroglou, Panagiotis Antoniou, Stathis Konstantinidis, Savvas Anastasiades, and Panagiotis Bamidis. 2020. Chatbots in healthcare curricula: the case of a conversational virtual patient. (2020), 137–147.

[9] Alessandro Fantechi, Stefania Gnesi, Samuele Livi, and Laura Semini. 2021. A spaCy-based tool for extracting variability from NL requirements. In *Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume B*. 32–35. Retrieved June 7, 2024.

[10] Thomas A Fergus and Sara L Dolan. 2014. Problematic internet use and internet searches for medical information: the role of health anxiety. *Cyberpsychology, Behavior, and Social Networking* 17, 12 (2014), 761–765.

[11] Tira Nur Fitria. 2023. Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay. 12, 1 (2023), 44–58.

[12] Rasa Technologies GmbH. 2024. Introduction to Rasa Open Source Rasa Pro. https://rasa.com/docs/rasa/ version 3.6.19, Retrieved June 4, 2024.

[13] Howdyai. 2023. howdyai/botkit - Cisco Code Exchange. https://developer.cisco.com/codeexchange/github/repo/howdyai/botkit/ Retrieved May 30, 2024.

[14] Carel Jansen. 2013. Taalniveau B1: de nieuwste kleren van de keizer. *Onze Taal* 82, 2 (2013), 56–57.

[15] Nikita Kanodia, Khandakar Ahmed, and Yuan Miao. 2021. Question answering model based conversational chatbot using bert model and google dialogflow. (2021), 19–22.

[16] Tatiana Kravchenko, Tatiana Bogdanova, and Timofey Shevgunov. 2022. *Ranking requirements using MoSCoW methodology in practice*. 188–199 pages.

[17] Arthur Loiselle. 2024. botfront / rasa-webchat. https://github.com/botfront/rasa-webchat, Retreived June 24, 2024.

[18] Nikolaos Malamas, Konstantinos Papangelou, and Andreas L. Symeonidis. 2022. Upon improving the performance of localized healthcare virtual assistants. *Healthcare* 10, 1 (11 2022), 99. https://doi.org/10.3390/healthcare10010099

[19] Anina Ot. 2023. IBM Artificial Intelligence (AI) Portfolio review. https://www.datamation.com/applications/ibm-artificial-intelligence-ai-portfolio/ Retrieved May 31, 2024.

[20] Masayuki Sekine, Koji Nishino, and Takayuki Enomoto. 2021. Differences in ovarian and other cancers risks by population and BRCA mutation location. *Genes* 12, 7 (2021), 1050.

[21] Elen Siglen, Hildegunn Høberg Vetti, Mirjam Augestad, Vidar M Steen, Åshild Lunde, and Cathrine Bjorvatn. 2023. Evaluation of the Rosa chatbot providing genetic information to patients at risk of hereditary breast and ovarian cancer: qualitative interview study. *Journal of Medical Internet Research* 25 (2023), e46571.

[22] Elen Siglen, Hildegunn Høberg Vetti, Aslaug Beathe Forberg Lunde, Thomas Akselberg Hatlebrekke, Nina Strømsvik, Anniken Hamang, Sigrid Tronsli Hovland, Jill Walker Rettberg, Vidar M. Steen, and Cathrine Bjorvatn. 2022. Ask Rosa – The making of a digital genetic conversation tool, a chatbot, about hereditary breast and ovarian cancer. *Patient Education and Counseling* 105, 6 (2022), 1488–1494. https://doi.org/10.1016/j.pec.2021.09.027

[23] Bernd Carsten Stahl and Damian Eke. 2024. The ethics of ChatGPT – Exploring the ethical issues of an emerging technology. *International Journal of Information Management* 74 (2024), 102700. https://doi.org/10.1016/j.ijinfomgt.2023.102700

[24] Kaylee Underkofler, Tanya Wanchek, Martha Thomas, Julia Green, Brie Worden, Sarah Nielsen, Sarah Savage, Robert Nussbaum, Chad Moretz, and Kari Ring. 2023. Modeling and comparison of costs and high-risk variant detection between a genetic counseling chatbot and a traditional genetic counseling pathway (169). *Gynecologic Oncology* 176 (2023), S64. Gia.

[25] Joren Wouters. 2024. Botpress Review 2024: Features, Pros and Cons, Pricing. https://chatimize.com/reviews/botpress/ botpress.

[26] Diliara Zharikova, Daniel Kornev, Fedor Ignatov, Maxim Talimanchuk, Dmitry Evseev, Ksenya Petukhova, Veronika Smilga, Dmitry Karpov, Yana Shishkina, Dmitry Kosenko, et al. 2023. DeepPavlov dream: platform for building generative AI assistants. (2023), 599–607.

A AI STATEMENT

During the preparation of this work, the author used Grammarly Premium and Writefull Premium in order to correct grammar and spelling mistakes and enhance readability. After using this tool/service, the author(s) reviewed and edited the content as needed and takes full responsibility for the content of the work."

B REQUIREMENTS

Functional and non-functional requirements for the chatbot to inform about BRCA-gene mutations.

B.1 Functional Requirements

Must have

(1) The chatbot must provide the user with answers to the open text input containing the user's question.
(2) The chatbot must greet the user and explain its capabilities and limitations. The chatbot takes the initiative when starting a conversation.
(3) The chatbot must answer the user's question with a pre-configured answer from the question database.
(4) The chatbot must throw a fallback strategy if a question is asked that the chatbot does not have in its database.

Should have

(1) The chatbot cannot make personal recommendations.
(2) The chatbot should include information about all genders
(3) The chatbot should collect interaction data on the times of day the chatbot is being used.
(4) The chatbot should collect interaction data on session length.

Could have

(1) The chatbot could collect interaction data on session length.
(2) The chatbot could gather feedback on how satisfied the user is with the answer to their question.

B.2 Non-Functional Requirements

Must have

(1) The chatbot must be a standalone application.
(2) The chatbot must be independent from companies that utilise its input for their own benefit.
(3) The chatbot must identify and match the question with the question-answer database.
(4) The chatbot must have a conversational tone.

Should have

(1) The Chatbot should take cyber-security measures
(2) The chatbot should use an open-source chatbot framework.
(3) The chatbot should be web-available on mobile and desktop devices.
(4) The chatbot should be able to expand tasks in the future.
(5) The chatbot should not collect personally identifiable information of users.

Could have

(1) The chatbot could have a user-friendly interface
(2) The chatbot's interface could be colourful
(3) The chatbot's interface could be informal
(4) The chatbot's interface could be inviting for all genders.
(5) The chatbot's interface could be accessible