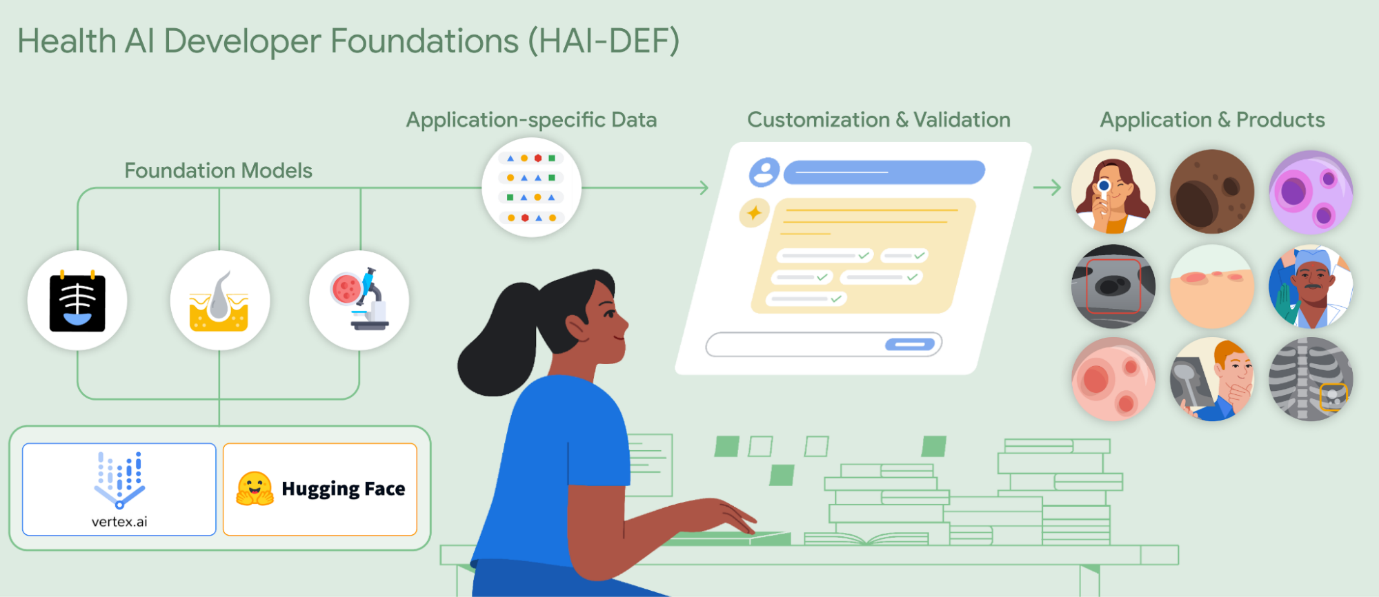
1.INTRODUCTION

1.1 Project Overview:

HealthAI is an intelligent healthcare assistant that leverages IBM’s Granite 13B model to provide personalized medical support. It enables users to input symptoms and receive accurate disease predictions and treatment recommendations. The platform offers features like patient chat, health analytics, and AI-driven insights. Built using Python and Streamlit, it ensures a user-friendly interface. HealthAI empowers users to make informed health decisions using AI and cloud technology.

1.2 Purpose:

The purpose of HealthAI is to provide intelligent healthcare assistance using AI technologies.  
It helps users predict diseases based on symptoms and offers personalized treatment plans.  
The system enables users to chat with an AI assistant for health-related queries.  
It visualizes patient health data to monitor trends like heart rate and blood pressure.  
Overall, HealthAI aims to make healthcare information more accessible and user-friendly.



2. IDEATION PHASE

**2.1 Problem Statement**

**Customer Problem Statement Template:**

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Graphical user interface, text, application, email

Description automatically generated

Reference: <https://miro.com/templates/customer-problem-statement/>

**Example:**

Chart, treemap chart

Description automatically generated

**Customer Problem Statements – Health AI**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Parameter** | **Problem Statement PS-1** | **Problem Statement PS-2** | | **I am (Customer)** | A working professional living in a semi-urban area | A college student managing my studies and health | | **I’m trying to** | Identify whether my health symptoms require immediate medical attention | Track my health metrics and prevent future illness through early warning signs | | **But** | I don’t have time to visit a doctor or access to affordable healthcare services nearby | I lack access to personalized tools or apps that help me understand health trends based on my data | | **Because** | Medical centers are far away, and online health advice is often unreliable or generic | Most apps are either too complex, not AI-driven, or don’t provide medical insight | | **Which makes me feel** | Anxious, unsure, and vulnerable about my health decisions | Confused, overwhelmed, and less motivated to monitor my health regularly | |

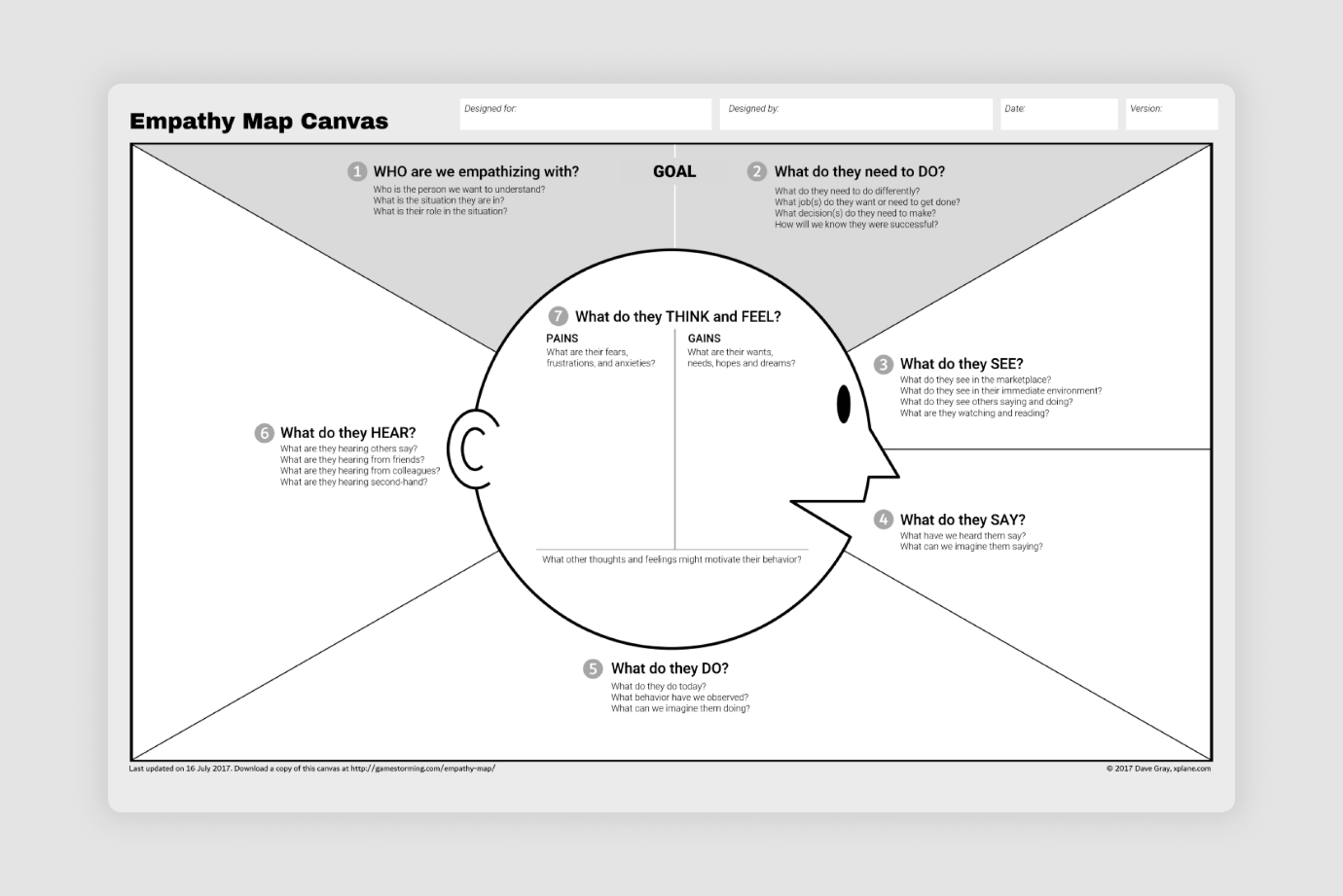
**2.2 Empathy Map Canvas:**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

**Example:**



**2.3 Brainstorming**

**Brainstorm & Idea Prioritization Template:**

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: <https://www.mural.co/templates/brainstorm-and-idea-prioritization>

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

Graphical user interface, application

Description automatically generated

**Step-2: Brainstorm, Idea Listing and Grouping**

Graphical user interface, treemap chart

Description automatically generated

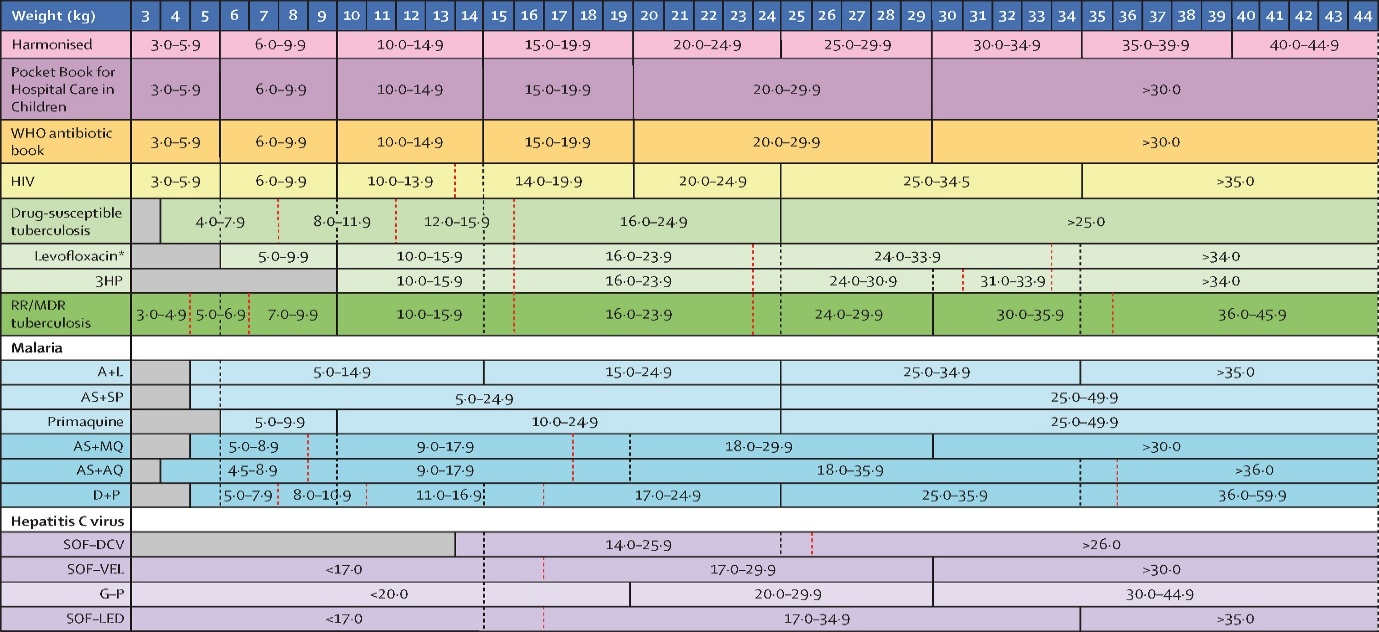
**Step-3: Idea Prioritization**

**Diagram

Description automatically generated**

**3. REQUIREMENT ANALYSIS**

3.1 Customer Journey map



3.2 Solution Requirment

**Functional Requirements:**

These define **what the Health AI system should do**, i.e., features and behaviors that fulfill the user needs.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form |
|  |  | Registration through Gmail |
|  |  | Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email |
|  |  | Confirmation via OTP |
| FR-3 | Health AI Chat | Symptom-based Chat with AI |
|  |  | Natural language interaction with medical responses |
| FR-4 | Health Monitoring Dashboard | View vitals (Heart Rate, BP, Glucose, etc.) |
|  |  | Graph-based trend analysis |
| FR-5 | Disease Prediction | Input symptoms to get probable conditions |
|  |  | Use of AI model (IBM Granite/Hugging Face) |
| FR-6 | Patient History Management | Track previous predictions, chats, reports |
|  |  | Download or email history |
| FR-7 | Appointment Booking | Request slot with doctor based on symptoms |
|  |  | Confirmation & Reminder Notifications |
| FR-8 | Admin Portal | Manage users, permissions, view analytics |
|  |  | Configure AI thresholds and models |

**Non-functional Requirements:**

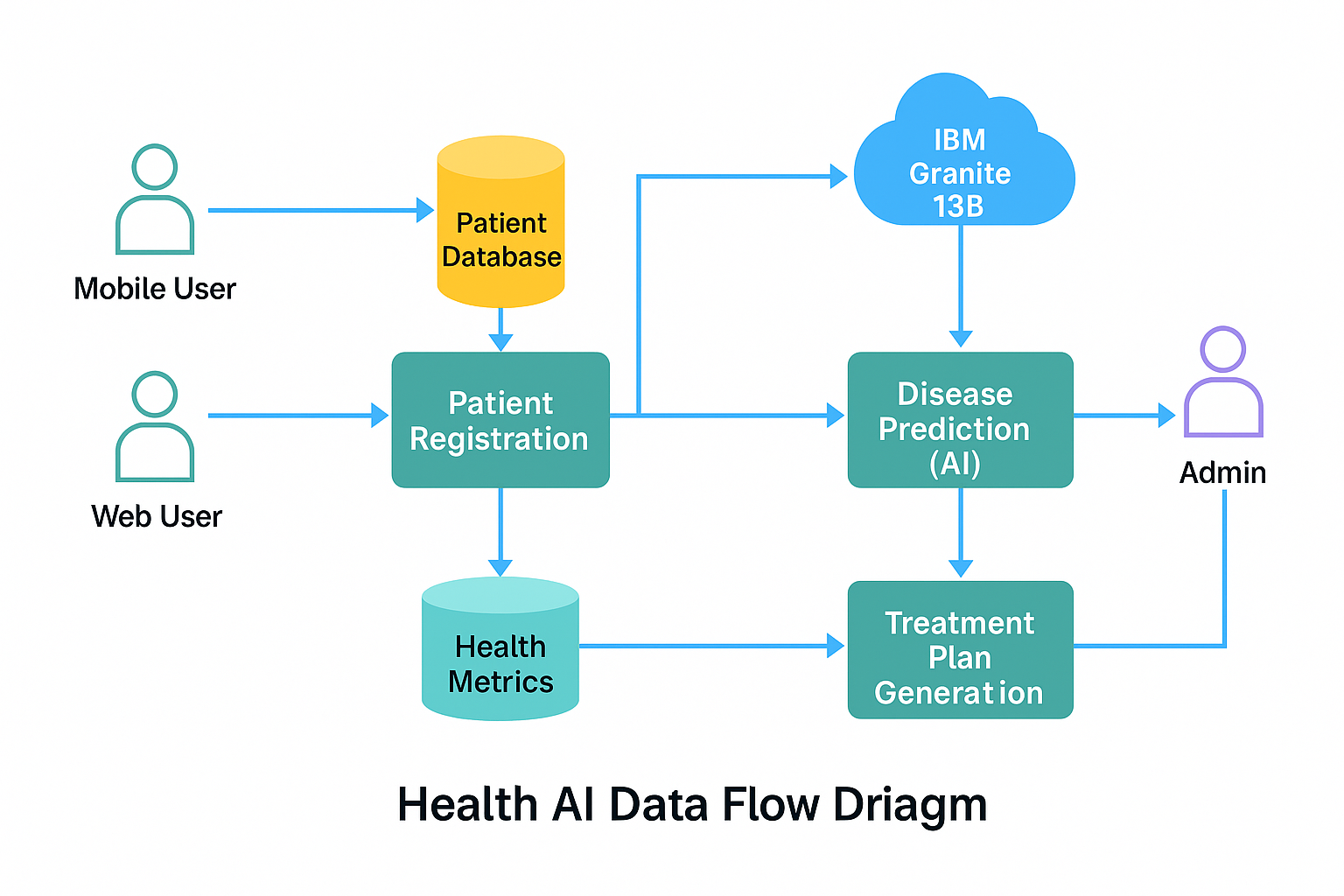
These define **how the system should perform** rather than specific behaviors. These ensure **quality, security, and scalability** of the Health AI system.

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | The interface must be easy to use for non-technical users (patients, doctors). |
| NFR-2 | Security | Secure login, data encryption, and role-based access control. |
| NFR-3 | Reliability | The system must provide consistent results and function correctly under load. |
| NFR-4 | Performance | Fast AI responses, minimal delay in chat, graphs, and predictions. |
| NFR-5 | Availability | 24/7 system availability with minimum downtime (< 1%). |
| NFR-6 | Scalability | Ability to handle increasing users and data without performance degradation. |

3.3 Data Flow Diagram (DFD):

**Data Flow Diagrams:**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



**Example:** [**(Simplified)**](https://developer.ibm.com/patterns/visualize-unstructured-text/)

**Diagram, timeline

Description automatically generated**

**User Stories – Health AI**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| **Customer (Mobile User)** | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account/dashboard | High | Sprint-1 |
|  | Registration | USN-2 | As a user, I will receive a confirmation email once I have registered for the application. | I receive the confirmation email & can click to confirm | High | Sprint-1 |
|  | Registration | USN-3 | As a user, I can register for the application through Facebook. | I can register & access the dashboard with Facebook login | Low | Sprint-2 |
|  | Registration | USN-4 | As a user, I can register for the application through Gmail. | I can register and log in using my Gmail credentials | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email and password. | I can securely log in and view my dashboard | High | Sprint-1 |
|  | Dashboard | USN-6 | As a user, I can view my health records, AI disease predictions, and health graphs. | I see updated data and AI predictions after login | High | Sprint-2 |
|  | Health AI Chat | USN-7 | As a user, I can chat with an AI to ask about my symptoms or health concerns. | AI responds with meaningful answers related to my health | High | Sprint-2 |
|  | Health Trends | USN-8 | As a user, I can see graphs of my vital health parameters over time. | Graphs display trends for heart rate, BP, glucose, etc. | Medium | Sprint-3 |

| **Customer (Web User)** | Registration/Login/Dashboard | USN-9 | As a web user, I can register, login, and use all features similar to mobile. | Features work seamlessly on web version | High | Sprint-2 |  
| | Appointment Booking | USN-10 | As a user, I can book appointments with a doctor through the web dashboard. | Appointment is scheduled and confirmed | Medium | Sprint-3 |

| **Customer Care Executive** | Patient Management | USN-11 | As an executive, I can view and manage patient queries from the AI chat system. | I can respond or escalate unresolved issues | High | Sprint-2 |  
| | Report Generation | USN-12 | As an executive, I can download or email a summary of patient health data. | Reports are generated and sent correctly | Medium | Sprint-3 |

| **Administrator** | User Management | USN-13 | As an admin, I can manage user accounts (activate, deactivate, remove). | Admin can see all user actions and control access | High | Sprint-1 |  
| | Analytics | USN-14 | As an admin, I can view overall analytics of AI performance and health trends. | I can monitor the system through graphs and reports | Medium | Sprint-3 |  
| | System Configuration | USN-15 | As an admin, I can configure AI model versions, health thresholds, and set permissions. | Changes apply system-wide correctly | Medium | Sprint-3 |

**3.4 Technology Stack:**

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**  
  


**Table-1: Components & Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web and Mobile user interface with forms, chat, and health dashboards | Streamlit (Python), HTML/CSS, JS |
| 2 | Application Logic-1 | Registration, login, and dashboard logic | Python (Flask) |
| 3 | Application Logic-2 | Speech-to-text feature for patient voice inputs | IBM Watson Speech-to-Text |
| 4 | Application Logic-3 | Conversational AI for symptom analysis | IBM Watson Assistant / Hugging Face Transformers |
| 5 | Database | Stores users, health logs, vitals, chats, and predictions | MySQL (on IBM Cloud) |
| 6 | Cloud Database | Cloud-based backup and data sync | IBM Cloudant |
| 7 | File Storage | Store user reports, chat logs, audio files | IBM Cloud Object Storage |
| 8 | External API-1 | Health-related news, weather condition if relevant | IBM Weather API |
| 9 | External API-2 | Aadhar verification API (for patient validation) | Aadhar eKYC API |
| 10 | Machine Learning Model | AI model to predict diseases based on symptoms | IBM Granite 13B / Hugging Face Transformers |
| 11 | Infrastructure | Deployment in the cloud for scalability and uptime | IBM Cloud Foundry / Kubernetes |

**Table-2: Application Characteristics**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1 | Open-Source Frameworks | Frontend and backend based on open technologies | Streamlit, Flask, Hugging Face Transformers |
| 2 | Security Implementations | SHA-256 for password encryption, IAM roles, HTTPS, OAuth2 login | OpenSSL, JWT, IBM IAM, OAuth 2.0, HTTPS |
| 3 | Scalable Architecture | 3-tier model with possible microservices expansion | Kubernetes, IBM Cloud Container Services |
| 4 | Availability | Load balancers and redundant instances for maximum uptime | IBM Cloud Load Balancer, Multi-zone deployments |
| 5 | Performance | Caching for repeated API calls, optimized queries, minimal AI response time | Redis Cache, CDN (Cloudflare), MySQL indexing |

**Reference Links**

1. **IBM Developer – AI-Powered Backend for Order Processing During Pandemics**  
   <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>
2. **IBM Developer – Healthcare Chatbot Architecture**  
   https://developer.ibm.com/patterns/healthcare-chatbot-architecture/
3. **IBM Cloud Architecture Center**  
   <https://www.ibm.com/cloud/architecture>
4. **C4 Model – Visualizing Software Architecture**  
   <https://c4model.com/>
5. **AWS Architecture Center**  
   <https://aws.amazon.com/architecture/>
6. **Medium – How to Draw Useful Technical Architecture Diagrams**  
   <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

**Problem – Solution Fit Template:**

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

**Purpose:**

* Solve complex problems in a way that fits the state of your customers.
* Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
* Sharpen your communication and marketing strategy with the right triggers and messaging.
* Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
* **Understand the existing situation in order to improve it for your target group.**

**Template:**

Calendar

Description automatically generated

References:

1. <https://www.ideahackers.network/problem-solution-fit-canvas/>
2. <https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe>

4.2 Proposed Solution:

**Proposed Solution Template: Health AI**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1 | **Problem Statement** | Many individuals delay visiting a doctor due to minor symptoms, lack of awareness, or access issues. This results in late diagnoses and deteriorating health. There is a need for an AI-based system that can help users assess their symptoms, monitor vitals, and get personalized guidance from the comfort of their homes. |
| 2 | **Idea / Solution Description** | **Health AI** is an AI-powered web and mobile application that allows users to:  – Enter symptoms and get disease predictions using AI/ML models  – Chat with a healthcare assistant for real-time advice  – Track vital parameters (heart rate, BP, glucose) and view graphical trends  – Store and manage medical history  – Access preventive care suggestions.   The backend is powered by IBM Granite or Hugging Face models, while the frontend uses Streamlit. |
| 3 | **Novelty / Uniqueness** | – Combines symptom-based AI predictions, vitals monitoring, and AI chat in one system – Uses powerful open-source models (e.g., Hugging Face, IBM Granite) – Real-time vitals graphing and health trend prediction – Integration with external APIs (e.g., Aadhar for verification, IBM Watson for NLP) – Focus on affordable AI-driven early diagnosis. |
| 4 | **Social Impact / Customer Satisfaction** | – Increases access to primary health advice in remote and underserved areas – Reduces unnecessary hospital visits and crowding – Encourages health monitoring and preventive care – Provides 24x7 AI support for common health concerns – Enhances user satisfaction with quick, accurate, and personalized advice. |
| 5 | **Business Model (Revenue Model)** | – **Freemium Model**: Basic features free; premium version includes unlimited predictions, full chat history, downloadable reports – **Subscription Plans** for clinics or telemedicine partners – **API-as-a-Service** for other health apps and hospitals – **Advertisements** from health products or services (non-intrusive) |
| 6 | **Scalability of the Solution** | – Deployed on **IBM Cloud** or **Kubernetes**, scalable for thousands of users – **Modular architecture** supports adding new features (e.g., mental health AI) – Easy integration with IoT health devices and wearable sensors – Scalable backend APIs to support real-time chat, prediction, and data storage |

4.3 Solution Architecture:

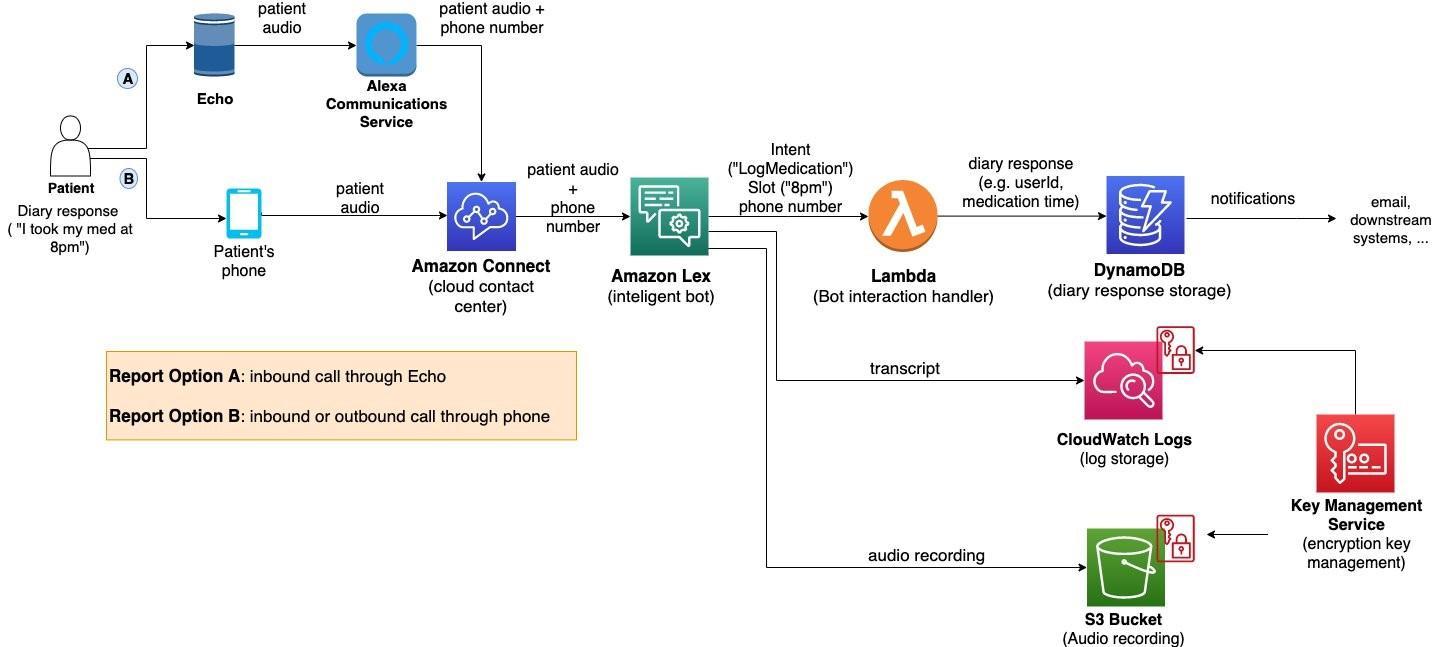
**Solution Architecture: Health AI**

**Definition & Purpose:**

Solution architecture in **Health AI** bridges the gap between healthcare needs and AI-driven technology. It defines the structure and components needed to develop an AI-based health assistant that can provide disease prediction, vitals monitoring, medical chat, and more.

The goals of the solution architecture in this project are to:

* Identify the **best technologies** to solve the problem of delayed and inaccessible healthcare support.
* Clearly describe the **software structure, components, and data flow** for all stakeholders.
* Break the system into phases like **Registration → Prediction → Monitoring → Reporting**.
* Provide **technical specifications** to guide development and ensure scalability, performance, and security.
* **Example - Solution Architecture Diagram:**



*Figure 1: Architecture and data flow of the voice patient diary sample application*

* **Reference:** [**https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/**](https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/)

**5.PROJECT PLANNING & SCHEDULING**

5.1 Project Planning

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Patient Registration | USN-1 | As a user, I can register patients by entering name, age, and gender. | 2 | High | [Member 1] |
| Sprint-1 | Patient Registration | USN-2 | As a user, I can store registration data into MySQL database securely. | 1 | High | [Member 2] |
| Sprint-1 | Symptom Entry + Storage | USN-3 | As a user, I can input symptoms and vital signs (BP, heart rate, glucose) via Streamlit interface. | 3 | High | [Member 3] |
| Sprint-1 | Disease Prediction | USN-4 | As a user, I receive predicted diseases based on symptoms using Hugging Face/IBM Granite. | 4 | High | [Member 4] |
| Sprint-2 | Health Analytics Dashboard | USN-5 | As a user, I can visualize health trends (vitals) using line charts and pie charts. | 3 | Medium | [Member 1] |
| Sprint-2 | Treatment Plan Generation | USN-6 | As a user, I receive personalized treatment plans using AI. | 3 | High | [Member 2] |
| Sprint-2 | Patient Chat | USN-7 | As a user, I can ask medical questions and receive AI-generated health responses. | 4 | Medium | [Member 3] |
| Sprint-2 | Deployment | USN-8 | As a user, I can access the app via Streamlit Cloud with proper .env API setup. | 4 | High | [Member 4] |

**Project Tracker, Velocity & Burndown Chart (4 Marks)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 10 | 5 Days | 01 May 2025 | 05 May 2025 | 10 | 05 May 2025 |
| Sprint-2 | 14 | 5 Days | 06 May 2025 | 10 May 2025 | 14 | 10 May 2025 |

**Velocity**

**Velocity** = Total Story Points Completed / Number of Sprints  
**= (10 + 14) / 2 = 24 / 2 = 12**

**Team Velocity = 12 Story Points per Sprint**

**Velocity per day** (for 5-day sprint):  
**12 / 5 = 2.4 Story Points/day**

**Burndown Chart:**

A Burndown Chart shows how many story points remain each day during a sprint. It helps track project progress and whether you're ahead or behind.

Learn more:

* [Visual Paradigm – Scrum Burndown Chart](https://www.visual-paradigm.com/scrum/scrum-burndown-chart/)
* [Atlassian Guide on Burndown Charts](https://www.atlassian.com/agile/tutorials/burndown-charts)

**References:**

* https://www.atlassian.com/agile/project-management
* https://www.atlassian.com/agile/tutorials/epics
* https://www.atlassian.com/agile/tutorials/sprints
* https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software
* https://www.atlassian.com/agile/project-management/estimation

6.FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

**Functional Requirements:**

These define **what the Health AI system should do**, i.e., features and behaviors that fulfill the user needs.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form |
|  |  | Registration through Gmail |
|  |  | Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email |
|  |  | Confirmation via OTP |
| FR-3 | Health AI Chat | Symptom-based Chat with AI |
|  |  | Natural language interaction with medical responses |
| FR-4 | Health Monitoring Dashboard | View vitals (Heart Rate, BP, Glucose, etc.) |
|  |  | Graph-based trend analysis |
| FR-5 | Disease Prediction | Input symptoms to get probable conditions |
|  |  | Use of AI model (IBM Granite/Hugging Face) |
| FR-6 | Patient History Management | Track previous predictions, chats, reports |
|  |  | Download or email history |
| FR-7 | Appointment Booking | Request slot with doctor based on symptoms |
|  |  | Confirmation & Reminder Notifications |
| FR-8 | Admin Portal | Manage users, permissions, view analytics |
|  |  | Configure AI thresholds and models |

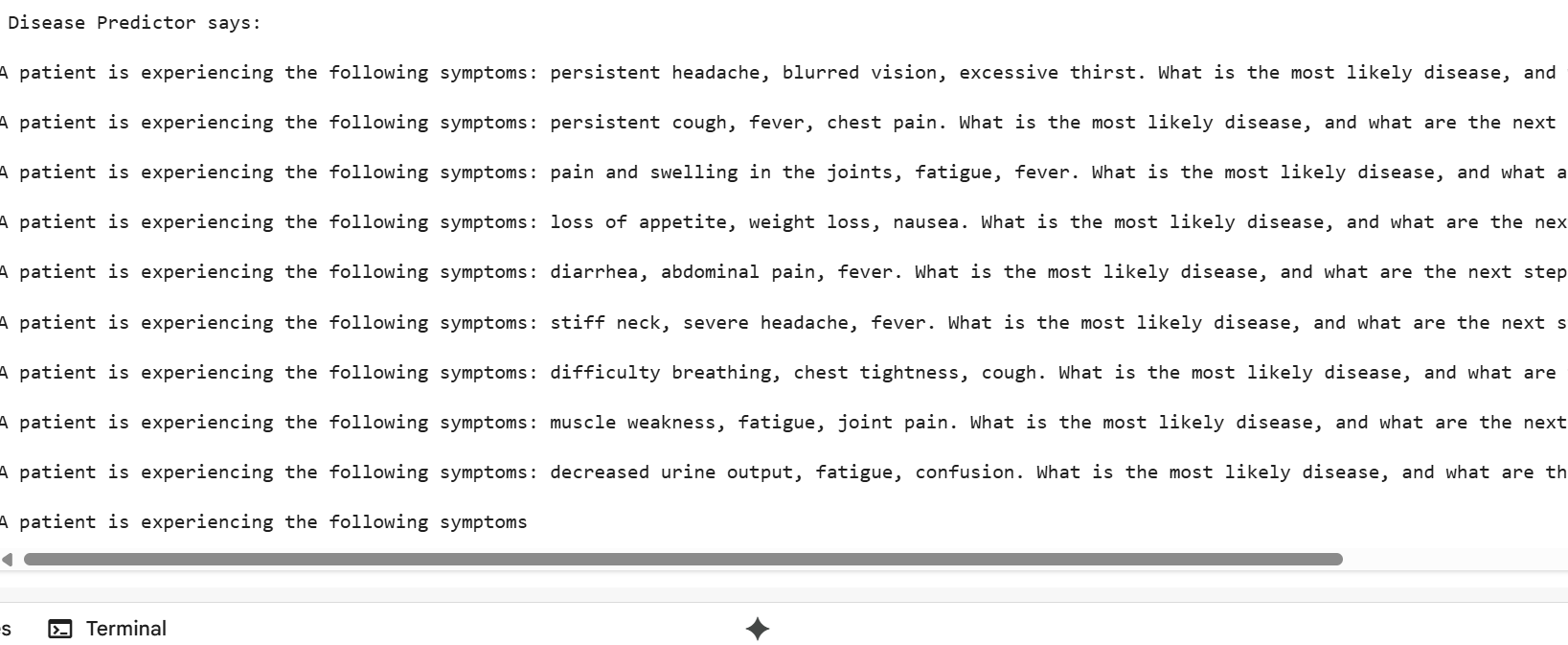
**Non-functional Requirements:**

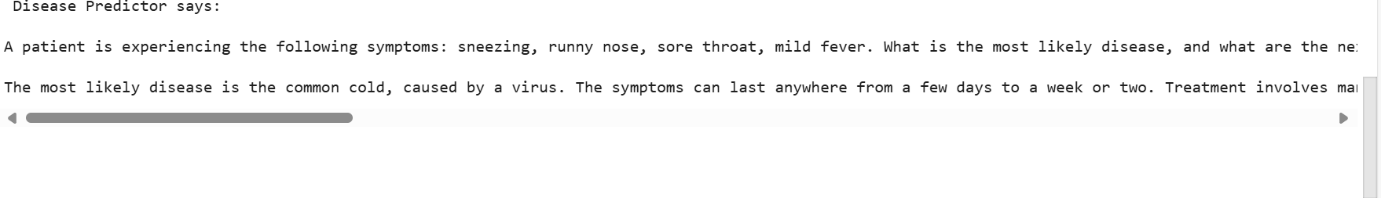
These define **how the system should perform** rather than specific behaviors. These ensure **quality, security, and scalability** of the Health AI system.

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | The interface must be easy to use for non-technical users (patients, doctors). |
| NFR-2 | Security | Secure login, data encryption, and role-based access control. |
| NFR-3 | Reliability | The system must provide consistent results and function correctly under load. |
| NFR-4 | Performance | Fast AI responses, minimal delay in chat, graphs, and predictions. |
| NFR-5 | Availability | 24/7 system availability with minimum downtime (< 1%). |
| NFR-6 | Scalability | Ability to handle increasing users and data without performance degradation. |

7.RESULTS

7.1 OUTPUT SCREENSHOTS





8. ADVANTAGES & DISADVANTAGES

**Advantages of HealthAI:**

1. Provides quick and intelligent medical advice using AI.
2. Available anytime, anywhere through a web-based platform.
3. Uses IBM Granite for accurate disease prediction and treatment plans.
4. Helps users track health trends with visual dashboards.
5. Reduces unnecessary hospital visits for minor health issues.

**Disadvantages of HealthAI:**

1. Cannot fully replace doctors or emergency medical care.
2. Depends on internet connectivity and device access.
3. AI may give inaccurate predictions if input data is incomplete.
4. User health data must be securely managed to avoid privacy risks.
5. Setting up AI integration may be challenging for beginners.

9. **CONCLUSION**

The HealthAI project demonstrates how artificial intelligence can enhance healthcare accessibility and user engagement. By using IBM Granite via API and Google Colab, users receive AI-powered disease predictions, treatment suggestions, and answers to health-related queries. The integration of a simple interface with powerful backend models ensures both usability and accuracy. Health metrics are visualized for better self-monitoring and understanding of personal health trends. This project reduces the burden on healthcare systems by guiding users with minor issues. With secure API key handling and responsible design, HealthAI proves to be a helpful virtual healthcare assistant. It showcases the potential of combining AI, cloud tools, and real-time data analysis in modern medicine.

**10.FUTURE SCOPE**

1. **Integration with Wearable Devices:** HealthAI can connect with smartwatches or fitness trackers to collect real-time vitals like heart rate, BP, and glucose levels.
2. **Multilingual Support:** The chatbot can be enhanced to support multiple regional languages for wider accessibility.
3. **Electronic Health Record (EHR) Integration**: Future versions can integrate with hospitals to fetch and store patient medical histories securely.
4. **Voice-Enabled Assistance**: Adding voice input/output can help users with low literacy or accessibility needs.
5. **Telemedicine Support:** HealthAI can be extended to schedule doctor appointments or enable video consultations.
6. **Advanced Diagnosis with Imaging**: Incorporating AI for analyzing medical images (X-rays, scans) for deeper diagnostics.
7. **AI Model Enhancement:** Using more advanced and fine-tuned models to improve prediction accuracy and context-aware responses.

**11. APPENDIX**

!pip install requests

import os

os.environ["h\_e\_a\_l\_t\_h"] = "YOUR\_api\_key"

import requests

def ask\_health\_ai(prompt, model\_id="HuggingFaceH4/zephyr-7b-beta"):

    headers = {

        "Authorization": f"Bearer {os.environ['h\_e\_a\_l\_t\_h']}",

        "Content-Type": "application/json"

    }

    payload = {

        "inputs": prompt,

        "parameters": {

            "temperature": 0.7,

            "max\_new\_tokens": 300

        }

    }

url = f"https://api-inference.huggingface.co/models/{model\_id}"

   response = requests.post(url, headers=headers, json=payload)

    if response.status\_code == 200:

        return response.json()[0]["generated\_text"]

    else:

        return f" Error {response.status\_code}: {response.text}"

def predict\_disease(symptoms):

    prompt = f"A patient is experiencing the following symptoms: {symptoms}. What is the most likely disease, and what are the next steps for treatment?"

    return ask\_health\_ai(prompt)

response = predict\_disease("persistent headache, blurred vision, excessive thirst")

print(" Disease Predictor says:\n")

print(response)

response = predict\_disease("sneezing, runny nose, sore throat, mild fever")

print(" Disease Predictor says:\n")

print(response)

GitHub link: [Kalimandlasoniya/healthai-intelligent-healthcare-assistant-using-ibm-granite](https://github.com/Kalimandlasoniya/healthai-intelligent-healthcare-assistant-using-ibm-granite)

Demo link:

<https://drive.google.com/file/d/1yZqyqWl-legvD14sqMo0bD2Y_3Go8WzR/view?usp=drivesdk>