WEB APPLICATION FOR CORRELATING DISEASES AND EMOTIONS ACCORDING TO GERMAN NEW MEDICINE

M.D. Stroia, *Babes-Bolyai University, Faculty of Engineering, Resita, ROMANIA*
C. Muscai*(corresponding author)**,** *Babes-Bolyai University, Faculty of Engineering, Resita, ROMANIA*
C. Hațiegan, *Babes-Bolyai University, Faculty of Engineering, Resita, ROMANIA*
C. Popescu, Constantin Brâncuși University of Târgu-Jiu, ROMÂNIA

ABSTRACT: This paper presents a web application designed to explore the relationship between diseases and emotions based on the principles of German New Medicine. The purpose of the system is to provide users with an interactive and accessible tool that associates emotional states with specific physical conditions, offering insight into possible psychological origins of illness. The application was developed using Visual Studio Code as the development environment, with React and Vite technologies ensuring high interactivity and optimized performance. Data is managed locally through JSON files, eliminating the need for a dedicated server or complex backend structure. This lightweight architecture enables fast response times, easy scalability, and simplified deployment. The project demonstrates how modern web technologies can effectively support exploratory tools in alternative medical frameworks, bridging emotional awareness and health understanding through digital means.

KEY WORDS: Web application, React, Vite, German New Medicine, emotion, disease, correlation.

1. BACKGROUND AND CONTEXT: EMOTIONAL INFLUENCES ON PHYSICAL HEALTH

The connection between emotional states and physical health has long been explored in psychology, physiology, and holistic medicine. From the classic "fight-or-flight" response, described as the autonomic reaction to perceived threat that prepares the body for either confrontation or escape, to modern notions of somatization and body-mind interplay, researchers and clinicians recognize that emotional and psychological factors can influence bodily functioning [6].

In stress physiology, for instance, acute activation of the sympathetic nervous system leads to heart rate increase, breathing acceleration, muscle tension and hormonal cascade — mechanisms that in short term are adaptive, yet when chronically engaged may contribute to somatic health issues such as cardiovascular, gastrointestinal or immune dysfunction [3].

Meanwhile, in psychology and psychosomatic medicine, somatization (or functional somatic symptoms) is characterized by physical symptoms without fully explained medical pathology, and emotional regulation deficits, such as alexithymia, poor emotional awareness, or maladaptive coping, have been implicated as contributing factors [9].

Holistic and alternative medical frameworks such as German New Medicine extend this thinking by proposing systematic

correspondences between emotional-conflicts, psychic shock, trauma, persistent emotional themes, and specific physical organ or tissue responses [13]. While the empirical validation of some of these correspondences remains limited, the conceptual premise aligns with the broader notion that emotion and physiology are entwined.

Against this backdrop, this paper presents a web application developed to allow users to explore correlations between diseases and emotions according to the German New Medicine model. The aim is to provide a digital, interactive tool supporting awareness of the potential emotional dimension of physical conditions, rather than to claim diagnostic authority. The application's development using React, Vite and JSON-based local data storage emphasizes accessibility, speed and simplicity in implementation.

This work addresses several needs: making the mind-body relationship more accessible via software, applying modern web engineering practices to a psychosomatic mapping tool; and offering a platform for exploratory engagement with alternative medicine theories within an IT context.

2. HEALTH IN A CLICK – DESIGN AND IMPLEMENTATION

"Health in a Click" is a web application designed to help users explore the correlations between diseases and emotions according to the principles of German New Medicine. GNM model proposes that specific emotional conflicts or shocks correspond to physical responses in the body, suggesting a systematic link between certain types of emotions, the corresponding organ or tissue reactions, and the associated brain relay that mediates these responses [10]. While GNM remains a complementary and alternative medical approach, it provides a structured model for understanding emotional dimensions of physical conditions.

2.1. Architecture of "Health in a Click"

The "Health in a Click" application is structured to provide a simple and intuitive user experience while demonstrating the correlations between diseases, emotions, and their corresponding organs based on the GNM model.

As shown in figure 1, the application begins with a landing web page, where users are introduced to GNM principles and given a brief explanation of how to use the system. A prominently displayed button allows users to enter the main application interface. This page also serves as an educational section, providing context about the emotional-physical links central to the GNM framework.

Within the application, users are presented with a form in which they must enter the required information about their condition. Once the form is completed, a submit button triggers the application logic, which processes the data, correlates the entered disease with the associated organ and emotional state, and generates a summary output. This output provides the user with a concise synthesis of their input, highlighting the relevant emotional conflicts, affected organs, and corresponding brain relays, according to the GNM model.

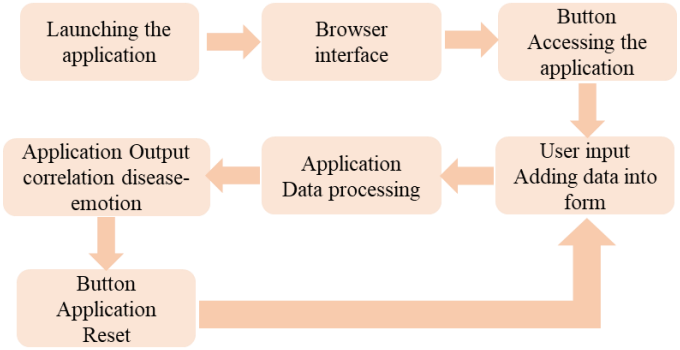

Figure 1. Application flow diagram

The evaluation form in "Health in a Click" is designed to capture user input systematically and provide meaningful insights according to the principles of GNM. The logical flow of the form, as depicted in figure 2, is as follows:

First step. Organ Selection

- The user first selects the affected organ from a dropdown menu.
- In GNM, each organ is associated with a specific type of emotional conflict. Selecting the organ allows the system to link the reported condition to its corresponding emotional dimension [12].

Second step. Affected Part / Lateralization

- If the organ has distinct regions, the user selects the specific part affected.
- If no specific part is required, the user indicates lateralization, left or right side of the body.
- GNM significance of lateralization: the left side typically corresponds to conflicts related to others, e.g., relational or social conflicts. Correct identification of lateralization helps refine the emotional correlation of the condition [8,12].

Third step. Symptom Selection

- The user selects predefined symptoms associated with the organ or affected area.
- This ensures that the system can validate and refine the GNM-based correlation between the organ, the emotional conflict, and the user's experience.

Fourth step. Date of Symptom Onset

- The user selects the date when symptoms began using a calendar component.
- The application calculates the duration of the conflict period, as GNM assumes that the duration of the healing phase mirrors the length of the initial conflict-active phase. This allows the system to contextualize the timing and potential resolution process for the emotional-physical correlation [8].

Fifth step. Processing and Output

- Once the user submits the form, the application processes the inputs and generates a summary section, including:
  - ✓ User-provided data.
  - ✓ The emotional state correlated to the conflict, according to GNM.
  - ✓ Example situational contexts from the user's life to help them reflect on possible triggers or unresolved conflicts.

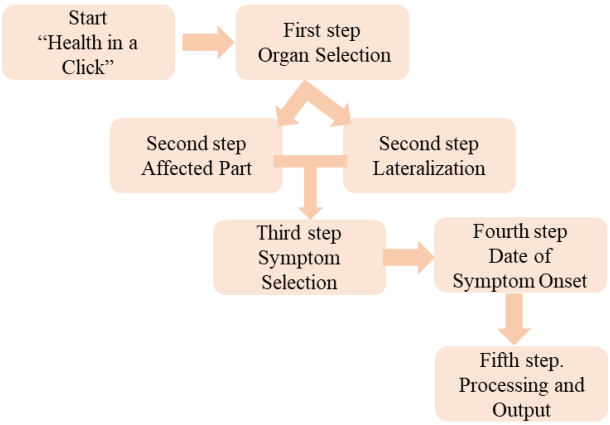

Figure 2. Form logical diagram

From GNM perspective the emotional shocks or conflicts manifest in specific organs thus selecting the organ is critical for accurate mapping. Lateralization helps distinguish whether the conflict involves personal issues or interpersonal/relational issues. The duration of symptom manifestation is linked to the duration of the conflict. Tracking onset dates allows the system to provide a temporal understanding of the healing phase.

2.2. Coding the "Health in a Click"

From a technical perspective, the architecture relies on a single-page application (SPA) model built with React and Vite, ensuring fast

navigation and interactivity [1, 2]. The system uses JSON files for local data storage, allowing the application to operate without a backend server, simplifying deployment and maintenance [4]. This design ensures both responsiveness and accessibility, making it easy for users to explore the emotional-somatic relationships in a guided digital environment.

Visual Studio Code (VS Code) was chosen as the development environment for its lightweight nature, extensive extensions, and built-in support for JavaScript and React development. Features like IntelliSense, debugging tools, and Git integration made it ideal for efficient development and testing [5].

React was selected for building the user interface as a component-based library that simplifies the creation of interactive UIs. React's state management and event handling are essential for implementing the dynamic forms and submit logic, allowing real-time updates of the interface based on user input [1, 11].

Vite provides a fast, modern build tool and development server. It ensures instant hot module replacement, quick start-up times, and efficient bundling, making the development process faster and smoother compared to older tools [2].

JSON is used for local data storage of diseases, organs, emotions, and predefined situations.

This choice eliminates the need for a backend server, simplifies deployment, and allows easy updates of the dataset without changing application logic [4, 7].

The front-end of the application is developed in React and is responsible for managing user interaction. The main component collects the data entered in the form, processes it through the *handleSubmit* function, and generates a personalized response that is displayed directly in the interface. The entire logic is implemented in the browser, without relying on any backend.

The *handleSubmit* function (figure 3) represents the central element for data processing within the application. It is triggered upon form submission and retrieves the information entered by the user — the affected organ, the diagnosis provided by the specialist, the specific part of the body in the case of paired organs, the list of symptoms, and the onset period.

Subsequently, the function checks whether the selected diagnosis involves lateralization and, based on the result, extracts from the *treatmentInfo* object the associated biological conflict and the corresponding examples. The information is then formatted into a clear and structured text, which is displayed to the user.

The entire process is performed locally, in the browser, without any dependency on a backend server, contributing to the application's speed, efficiency, and autonomy.

```
function App() {
  const [organ, setOrgan] = useState("");
  const [diagnostic, setDiagnostic] = useState("");
  const [partea, setPartea] = useState("");
  const [simptome, setSimptome] = useState([]);
  const [result, setResult] = useState("");
  const [treatment, setTreatment] = useState("");
  const [calendarResult, setCalendarResult] = useState("");
  const [showResult, setShowResult] = useState(false);
  const handleCalendarResult = (text) => {
    setCalendarResult(text);  };
  const handleSubmit = (e) => {
    e.preventDefault();
    const simptomeText = simptome.join(", ");
    let tratament = "";
    let exemple = "";
    if (lateralDiagnoses.includes(diagnostic)) {
      const info = treatmentInfo[diagnostic]?.[partea];
      if (info) {
        tratament = info.conflict.join("<br /><br />");
        exemple = info.exemple.join("<br /><br />");
      } else {
        tratament = "Tratament nespecificat.";
        exemple = ""; }
    } else {
      const info = treatmentInfo[diagnostic];}
      if (info) {
        tratament = Array.isArray(info)
          ? info.join("<br /><br />")
          : info.conflict?.join("<br /><br />");
        exemple = info.exemple?.join("<br /><br />") || "";
      } else {
        tratament = "Tratament nespecificat.";
        exemple = ""; };
      }
    }
```

Figure 3. *handleSubmit* function sample coding

The constant *treatmentInfo* represents an object-based structure used as the internal database of the application. It contains information about the biological conflicts corresponding to each diagnosis. A coding sample is given in figure 4. The keys of this object are the names of the affected organs or structures, while the associated values are objects that include the fields conflict and examples, each represented as lists of text entries. For structures involving lateralization, the information is further organized according to the side of the body selected by the user. Thus, *treatmentInfo* enables fast access to relevant content and generates a personalized response based on the data entered, without requiring a backend server or any external connection.

```
const treatmentInfo = {
"Glande lacrimale": {
    dreapta: {
      conflict: ["Conflict legat de o bucățică vizuală",
        "Partea dreaptă: neputința de a captura o bucățică vizuală - nu pot percepe sau înțelege o informație vizuală importantă
        În acest caz, persoana simte că nu poate înțelege sau nu poate reține o imagine,
        o scenă sau un detaliu important pe care l-a văzut.",
      ],
      exemple: [
        "O mamă își pierde copilul în mulțime și, deși se uită în toate direcțiile, nu reușește să-l mai vadă -
        conflict de neputință de a capta imaginea copilului.",
        "O persoană primește un act important (contract, factură, poză), dar nu apucă să-l vadă clar sau îl pierde
        „am ratat ceva esențial".",
        "O persoana observă o expresie ciudată pe fața partenerului (posibilă infidelitate), dar nu poate înțelege
        exact ce a văzut - conflict de neînțelegere a semnalului vizual.",
        "Un om asistă la un accident dar nu reușește să vadă clar detaliile (numere de  mașină, fața vinovatului)
      „nu pot capta imaginea care conta".",
        "Un copil isi doresti foarte mult sa "vada" jucaria promisa de parinti dar acea jucarie nu mai "vine". ", ], },
    stanga: {
      conflict: [ "Conflict legat de o bucățică vizuală",
        "Partea stângă: neputința de a elimina o bucățică vizuală, nu pot scăpa de o imagine, un detaliu vizual sau
        o imagine care mă obsedează - Aici, conflictul apare atunci când o persoană este obsedată sau traumatizată de
        o imagine sau un detaliu pe care nu îl poate sterge din mintea sa.",  ],
      exemple: [
        "O mamă își surprinde copilul într-o situație periculoasă (căzând, plângând etc.) și imaginea îi rămâne întipărită în minte.",
        "O persoană vede o poză tulburătoare pe internet (război, cruzime față de animale) și nu o poate uita
        conflict cu imagine vizuală traumatică.",
        "O persoană asistă la un accident grav sau la o scenă violentă, iar imaginea îi revine obsesiv în minte
        „nu pot să uit ce am văzut"."
      ], }, }, };
```

Figure 4. Coding sample for *treatmentInfo* object construction

The *CalendarCalculator* component from figure 5 is responsible for calculating the time interval elapsed from the date selected by the user to the present. It uses an internal function, *calculateDifference*, which determines the difference expressed in years, months, and days, based on the chosen date. Through the pluralize function, the numerical values are

accompanied by the grammatically correct forms of the time units. The result is then formatted into an informative message, such as "1 month has passed" or "2 years and 3 months have passed", depending on the case. The message is automatically sent to the main component (App) via the *onResult* function whenever the user selects a new date, thanks to the use of the *useEffect* function. This approach allows the automatic display of the calculated period in a clear and user-friendly format.

```
const CalendarCalculator = ({ onResult }) => {
  const [selectedDate, setSelectedDate] = useState(null);
  const pluralize = (value, singular, plural) => {
    return `${value} ${value === 1 ? singular : plural}`;  };
    const calculateDifference = () => {
    if (!selectedDate) return;
    const today = new Date();
    const totalMonths = differenceInMonths(today, selectedDate);
    const years = Math.floor(totalMonths / 12);
    const months = totalMonths % 12;
   const datePlusYearsAndMonths = addMonths(selectedDate, years * 12 + months);
    const days = differenceInDays(today, datePlusYearsAndMonths);
    let parts = [];
    if (years > 0) parts.push(pluralize(years, "an", "ani"));
    if (months > 0) parts.push(pluralize(months, "lună", "luni"));
    if (days > 0) parts.push(pluralize(days, "zi", "zile"));
    const message =
      parts.length === 1
        ? `${parts[0].startsWith("1 ") ? "A trecut" : "Au trecut"} ${parts[0]}.`
        : parts.length > 1
        ? `Au trecut ${parts.join(" și ")}.`
        : "Data aleasă este chiar astăzi.";
    onResult(message);};
  React.useEffect(() => {
    calculateDifference();
  }, [selectedDate]);}
```

Figure 5. *CalendarCalculator* component implementation

2.3. "Health in a Click" features

The main page of the platform (Figure 6) was developed using modern web technologies, with an HTML structure written in JSX syntax, specific to the React library, and styled using CSS.

At the top of the page, there is a navigation bar containing the centered platform logo and links to the "Home" and "Form" sections. Below the navbar, a motivational message — "Start your journey toward a healthier life" — is displayed, accompanied by an action button, "Start Evaluation", which initiates the form completion process.

The following section highlights the platform's three main advantages: simplicity, through quick form completion; confidentiality, by ensuring the protection of personal data; and accessibility, through continuous availability of the platform from any location.

At the bottom of the page, the "How It Works" section provides a brief explanation of the application's process. The page ends with a footer containing the platform name, copyright notice, additional navigation links ("About", "Contact"), and social media icons.

The overall design aims to facilitate quick access to the application's main functionality while maintaining a clean, user-friendly interface.

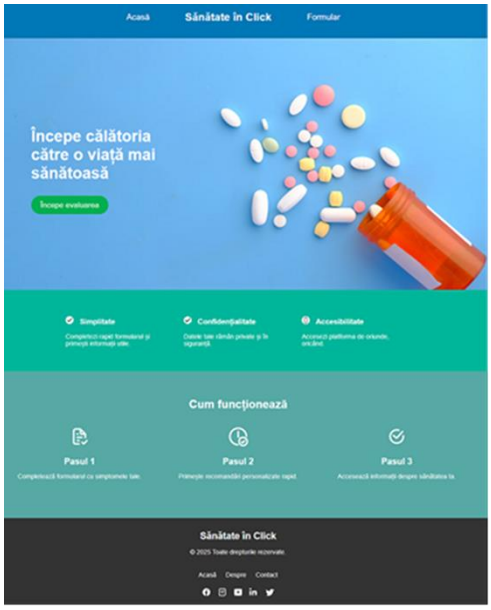

Figure 6. Main page of "Health in a Click"

The application structure is modular and has been organized based on React components. The main components include the evaluation form, the data input elements (questions and answers), navigation buttons, and the results display section. The application state was managed using the *useState* function, allowing for the real-time collection and updating of user responses.

For navigation between pages or sections, React Router was employed, while the visual layout was styled using CSS to ensure a pleasant and responsive interface.

The form, as illustrated in Figure 7, was built using reusable components, which facilitated the application's scalability and maintenance. The user's responses are processed locally, and based on the selected options, corresponding suggestions are generated. These suggestions are retrieved from predefined JSON files, through a matching mechanism between the reported symptoms and the associated informational messages.

After completing the form, the user is presented with the results in a dedicated section. These results include descriptive information regarding the possible emotions and experiences that may have contributed to potential biological conditions, as well as examples of life situations the user can reflect upon in their process of self-awareness concerning the mind – body – emotion connection.

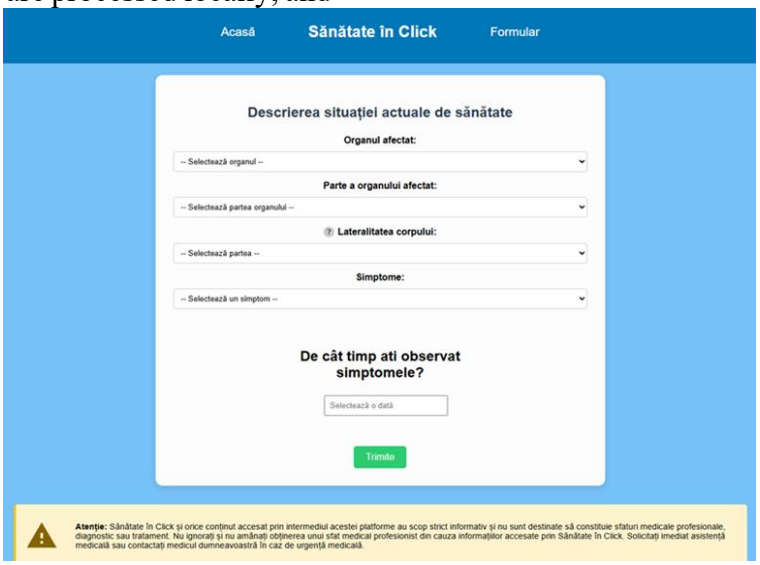

Figure 7. "Health in a Click" application form

Additionally, the user has the option to restart the form, allowing either a new evaluation for another user or the modification of previously entered responses. The display of results was implemented using the same technologies as the evaluation form — JSX for content structuring and CSS for interface styling.
Figure 8 illustrates how the final result is generated and displayed after form submission. The displayed information includes the affected organ, the specific part of the organ, the selected symptoms, the time period indicated by the user, as well as details about the associated biological conflict and relevant examples. At the bottom of the page, a visual disclaimer message informs the user that the platform serves a strictly informational purpose and does not provide professional medical advice or recommendations.

Figure 8. "Health in a Click" response to user's data

The platform is designed with a responsive layout, ensuring full functionality and accessibility across various devices such as personal computers, laptops, tablets, and mobile devices.

3. CONCLUSION

The development of the Health in a Click application demonstrates how modern web technologies can be effectively combined with conceptual models from alternative medicine to create interactive and educational digital tools. By implementing a fully client-side architecture using React, Vite, and JSON, the application achieves both high performance and accessibility, functioning seamlessly across all device types — desktop, laptop, tablet, and mobile.
From a functional perspective, the system provides users with an intuitive way to explore the possible connections between physical conditions and emotional experiences, based on the principles of German New Medicine (GNM). The interface design and local data processing allow for fast response times, data privacy, and ease of use without the need for a complex backend infrastructure.
On a broader level, the project highlights the potential of information technology to support personal reflection and self-awareness regarding the mind–body–emotion relationship. While the platform is not intended as a diagnostic tool, it offers a structured and accessible framework that can inspire further interdisciplinary research, bringing together computer science, psychology, and complementary health sciences.
Future work may include extending the dataset, integrating user feedback, or

implementing multilingual support to enhance the platform's usability.

REFERENCES

[1]Banks, A., Porcello, E., Learning React: Modern Patterns for Developing React Apps 2nd Edition, O'Reilly Media Publishing, 2020.

[2]Deniz, E., Vite: The Build Tool for Modern Web Development, 2024, https://medium.com/@emre.deniz/vite-the-build-tool-for-modern-web-development-0e99906f2f0c

[3]Dhabhar, F. S., The short-term stress response - Mother nature's mechanism for enhancing protection and performance under conditions of threat, challenge, and opportunity, Frontiers in Neuroendocrinology, Vol. 49, Elsevier Publishing, 2018.

[4]D'mello, B. J., Sriparasa, S.S., JavaScript and JSON Essentials Build light weight, scalable, and faster web applications with the power of JSON, 2nd Edition, Packt Publishing, 2018.

[5]Johnson, B., Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers, Wiley Publishing, 2025.

[6]LeWine, H. E., Understanding the stress response. Chronic activation of this survival mechanism impairs health, Harvard Health Publishing, 2024.

[7]Marrs, T., JSON at Work, O'Reilly Media Publishing, 2017.

[8]Renaud, G., Piramida sănătăţii: conflictele biologice şi emoţionale din spatele afecţiunilor şi comportamentelor, Oradea Publishing, 2021.

[9]Schnabel, K., Petzke T. M., Witthöft M., The emotion regulation process in somatic symptom disorders and related conditions - A systematic narrative review, Clinical Psychology Review, Vol. 97, Elsevier Publishing, 2022.

[10] Thron, R., German New Medicine: Unraveling the Revolutionary Approach to Healing and Understanding the Mind-Body Connection for Optimal Health, Independently Published, 2024.

[11] Wieruch, R., The Road to React: The React.js 19 with Hooks in JavaScript Book (2025 Edition), Independently Published, 2025.

[12] Willow, K., German New Medicine Experiences in Practice: An introduction to the medical discoveries of Dr. Ryke Geerd Hamer Dr. Katherine, Independently Published, 2019.

[13] Winston, O., German New Medicine: An Introduction to Understanding the Mind-Body Connection and Healing Naturally, Independently published, 2024