Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 832–839

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

OUTB: application for decision-support in the outcomes of Tuberculosis

Mariana MOZINI[a]*, Raul ROTHSCHILD[a], Ana Clara MIOTO[b], Filipe BERNARDI[b], Vinícius LIMA[a], Giovane SOARES[a], Renan SEGAMARCHI[a], Domingos ALVES[a]

[a]*Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, Brazil*
[b]*Bioengineering Postgraduate Program, University of São Paulo, São Carlos, Brazil*

Abstract

Brazil is among the 30 countries with the highest tuberculosis prevalence, and many studies are trying to understand and fight the harmful outcomes of this disease. In that sense, the following project proposes the development of a mobile application prototype that centralizes diagnosis aid tools for tuberculosis, making the process more transparent for the patient, using the Single Patient Application (SPA) concept. The application was built based on JavaScript and the frameworks chosen were Vue Framework along with Framework7 due to its focus on interface components. Communication with the back-end was done through AJAX calls. As a result, a prototype was built with the five main screens that the user will interact with: authentication/login screen, home screen, list of available algorithms, outcomes algorithm home screen and patient history. The interface design is expected to facilitate the planning of other application areas and contribute to defining business rules. Once implemented on the server side, these rules can support an organized, patient-centered tuberculosis database.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: mtmozini@usp.br

1877-0509
10.1016/j.procs.2024.06.241

Keywords: Tuberculosis; Mobile application; Decision support; Health guidelines.

1. Introduction

Tuberculosis (TB) remains a fatal infectious disease worldwide. It is a curable disease that kills 4500 people daily and sickens about 30,000 people [1]. Along with other comorbidities such as HIV, TB becomes even more lethal. In 2016, TB contributed to over 374,000 HIV deaths [2].

An essential point in the fight against TB is socioeconomic and cultural issues. Countries with greater social inequality and lower per capita income have a higher prevalence of TB, evidencing the relationship between poverty and TB globally [3]. Brazil is among the 30 countries with the highest prevalence of TB, and several studies have been carried out to analyze how socioeconomic conditions interventions made toward TB outcomes issues are related [4].

One strategy for supporting TB treatment is the use of mobile applications. The increasing adoption of technology in healthcare and education has paved the way for mobile learning, where mobile devices are used alone or in combination with other Information and Communication Technologies in Health (ICTs). These technologies facilitate learning by providing portable, accessible, and interactive opportunities, enabling users to interact with essential treatment information, irrespective of their educational background [5,6].

A predictive model of bad outcomes in treating TB patients is being developed based on machine learning techniques. These analyses are supported by retrospective data from several sources, including TBWeb, a notification and follow-up system for tuberculosis cases in the State of São Paulo, Brazil.

An important aspect is the definition of outcomes in a standardized way and that they are mutually exclusive. This work followed the World Health Organization (WHO) recommendations for each type of outcome [7] and addressed the wide range of treatment regimens and durations currently used worldwide. The harmful outcomes can be interpreted as: 1. Death, 2. Hospitalization, 3. Abandonment of treatment, 4. Development of resistance (MDR and XDR), 5. Adherence to low treatment.

2. Background

2.1. Mobile learning

In recent years, Brazil has experienced a significant increase in the use of mobile devices, such as smartphones and tablets. This trend is driven by several factors, including increased accessibility, reduced device prices, and advancements in communication technologies. According to data from the Brazilian Institute of Geography and Statistics (IBGE), in 2021, approximately 99% of households had a cellphone as their primary device for internet access [8].

The healthcare field is certainly one of the most influenced by the growing use of communication and information technologies. Over the years, the utilization of mobile devices has increased and gained more space in this field, giving rise to the concept of mobile health (mHealth), defined as medical and public health practices aided by portable devices such as mobile phones, patient monitoring devices, digital personal assistants, and other wireless devices [9].

2.2. TB monitoring

Several initiatives in information systems have been developed to support the management of tuberculosis treatment. For instance, e-TB Manager enables case monitoring, medicine management, report generation and data analysis [10]. Both PIH-EMR and OpenMRS MDR-TB aid in management and support the care of multidrug-resistant TB patients [11,12]. Video Directly Observed Therapy (VDOT) enables patients to record and transmit videos of medication ingestion, which are remotely monitored by healthcare professionals [13]. Furthermore, the SISTB Ecosystem comprises various tools aimed at facilitating the monitoring of TB treatment. These modules and layers have been designed to enhance the daily workflow of healthcare professionals, whether used in conjunction or independently [14].

Taking into consideration the results of the previously listed research, the OUTB is a software that can easily integrate with the SISTB ecosystem, allowing for greater interoperability and serving as a suitable alternative for the Brazilian healthcare system.

3. Methods

3.1. Study design

The methodological basis of scientific research for this work is Action Research. This strategy is appropriate since the project has a practical component in addition to the theoretical development of the study. This way, the sociotechnical approach was applied to construct the evaluation process. Potential users actively collaborated with technical developers through various strategies, such as regular group meetings, where potential users were motivated to use and improve the evaluated system, integrating workflows and making suggestions based on their experiences.

3.2. Development process

The initial concept of developing OuTB originated within LIS (Laboratório de Inteligência em Saúde), where a team of scientists and undergraduate students had previously conducted research on tuberculosis and healthcare. Their expertise in the field played a crucial role in shaping the project's conception. They contributed their knowledge and insights, assuming the perspective of potential end users, to ensure the development of a solution that aligned with the needs of healthcare professionals.

The software engineering stages, requirements gathering, elaboration and specification of functionalities were built following an iterative software development model called The Prototype Paradigm [15]. Few key principles from that paradigm are the continuous delivery, early user involvement, and the use of prototypes for gathering feedback and refining the software that is being developed.

Therefore, within biweekly meetings collaboratively with LIS members, prototypes were created and refined. Subsequently, the next step was the development application until the next meeting, where the whole process would restart.

The development process began with designing the screens, focusing on the visual elements and general flow of the application. Figma [16], a popular collaborative platform for creating user interfaces, was utilized to develop the visual concept of the mobile application. Leveraging its features, interactive components, and design versioning capabilities, Figma facilitated the iterative adaptation of the prototypes.

In addition, the continuous delivery factor led to an early development of the front-end application using the concept of Single Page Application (SPA), where the users can interact with through a single page, dynamically changing its contente [17]. Moreover, as the front-end and design matured, it was realized that the server-side application needed a deeper understanding of how the entire architecture should be embraced to seamlessly incorporate the prediction model for adverse effects of TB treatment.

In the server-side development process, a Representational State Transfer Application Programming Interface (REST API) was designed with the intention of following an adapted architecture from the Open API Initiative (OAI) [18]. This initiative provided a specification standard that guided the thinking behind information flow at each stage of the API lifecycle. Furthermore, an authentication system was devised, along with modules for integrating third-party systems. The architecture was conceived to accommodate the adaptation and integration of different machine learning algorithms, and to format the obtained results in a simpler way for visualization on the client-side.

3.3. Technologies

The chosen technologies and development tools were considered based on their ease of learning and the comprehensive ecosystem they provide.

3.3.1. Client-side application

In web development, the choice of frameworks for developing SPAs has evolved over time. React, Vue, and Angular have emerged as popular solutions in the past five years, as indicated by the current download statistics on NpmTrends [19, 20].

When considering React, it is primarily a user interface library that is relatively easy to learn. However, certain features such as routing and state management are not included in the library itself but are provided by community-developed solutions like ReactRouter [21] and Redux [22]. On the other hand, Angular is a robust framework with a comprehensive set of pre-implemented features. However, it has a steeper learning curve due to its specific design patterns and architecture [20]. Ultimately, Vue was chosen primarily for its simplicity and user-friendly nature. The Vue ecosystem, which is officially maintained by the Vue team, includes essential features like VueRouter [23] and Vuex [24].

Furthermore, the OuTB user interface was built using Framework7 and Bootstrap as the selected frameworks. The decision to utilize these frameworks was driven by the goal of maximizing productivity throughout the application development process. By offering a wide range of pre-built components and extensive customization options, Framework7 [25] and Bootstrap [26] played a crucial role in supporting continuous delivery and enabling the creation of constantly evolving prototypes.

Lastly, for packaging the entire application, the combination of Vite.js [27] and Cordova [28] was employed. Vite served as the tool for bundling and building HTML, CSS, and JavaScript files. On the other hand, Cordova was utilized as a platform targeting solution, enabling the application to be deployed across multiple platforms using a single code base [29]. This approach simplified the deployment process and ensured seamless compatibility with various operating systems and devices.

3.3.2. Server-side application

A JavaScript framework built on Node.js [30] was chosen for backend development, allowing for the utilization of the same programming language for both frontend and backend, thereby reducing the learning curve. To enhance communication between the backend and the database, a JavaScript-compatible tool was selected. For the server-side application, Express [31] was chosen as the web application framework to streamline API development and manage HTTP requests. Prisma, an Object-Relational Mapping (ORM) tool [32], simplifies database interaction by handling the complexities of SQL queries. PostgreSQL, a reliable and scalable relational database [33], was chosen to ensure data integrity and support complex relationships. This combination of Express, Prisma, and PostgreSQL created a robust stack, enabling the development of a scalable and high-performing server-side application.

4. Results

The first part of the application is the authentication module, which comprises two main processes: authentication and user access authorization. Authentication is verifying the user's identity, typically using credentials like email and password, authorization pertains to the user's permission to perform CRUD (Create, Read, Update, Delete) operations according to their access profile. It's crucial to authenticate the user correctly so that the system only returns data suitable for their access profile.

When the application is accessed for the first time, the user is redirected to the login screen (Figure 1), where they must provide their credentials for authentication. If the authentication is successful, the application grants access to the initial screen (Figure 2). If already authenticated, the user’s credentials are constantly checked for possible timeouts.

a b

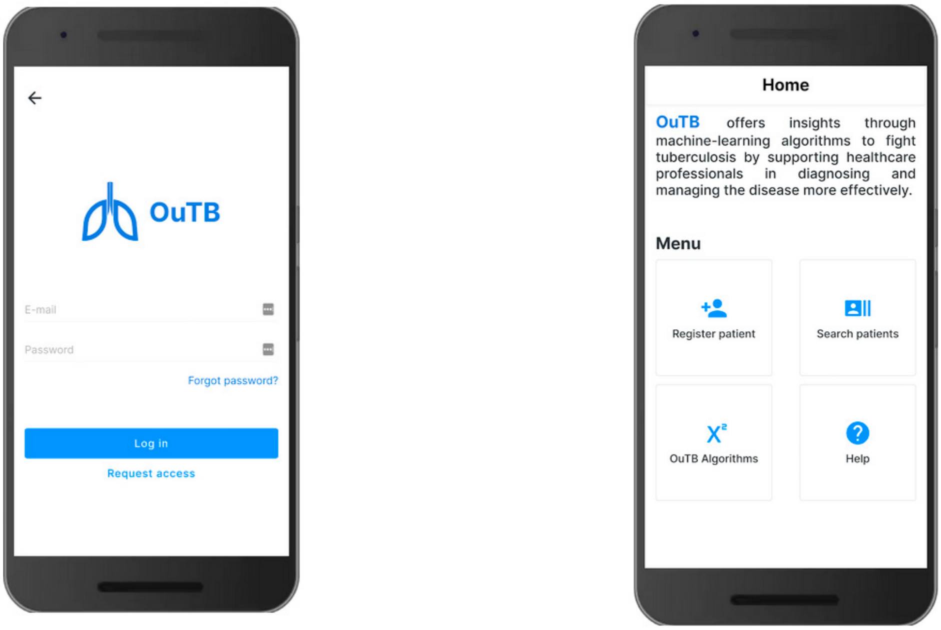

Fig. 1. (a) OuTB's login screen; (b) Home screen (TBWEB/SISTB module)

The SISTB module is a healthcare provider system that connects to various health information systems and retrieves specific user data, Where users can register patients and search for registered patients. Finally, the user can navigate to the algorithms module where OuTB functionalities are available.

a b

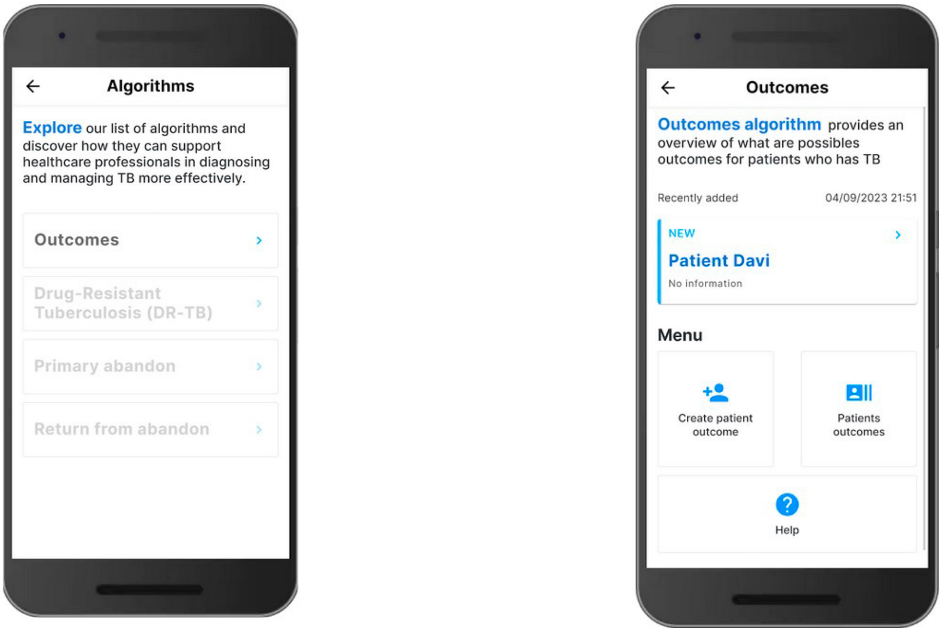

Fig. 2. (a) List of available algorithms; (b) Outcomes algorithm home screen

The Algorithms module lists all the current machine learning-based algorithms (Figure 3). When selecting an algorithm, it is necessary to collect some patient general, social-economic, and health information through a form to produce an output related to that specific treatment scenario. Then the role of OuTB is to display that technical response on a screen where it is possible to visualize the algorithm's response to the potential insights in a simple way.

Each algorithm targets a specific scenario that can occur during the treatment of a patient with tuberculosis. As an example, there may be an algorithm of "Outcomes" that could indicate what are the possible outcomes for a patient treatment between "Stable", "Deterioration", "Critical", "Death" or "Healed". The Drug-Resistant Tuberculosis (DR-TB) would indicate if the patient's bacillus becomes resistant to the drug(s) administered in the treatment.

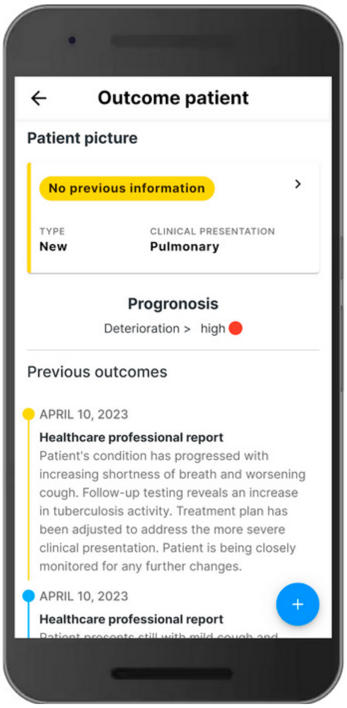

FIg. 3. Algorithm module (Outcomes) showing the patient's treatment history.

In the Outcomes algorithm (Figure 4), the user can register a patient with the required data or select an existing patient retrieved from the connected healthcare provider and then complete the possible required data that may not exist on the source database. Once the necessary information is filled out, a response is generated.

The Outcome Patient screen (Figure 5) displays an overview of a single patient and what was the last output made by the machine-learn algorithm. The user can update the information previously inserted to generate new and perhaps more relevant results according to the evolution of the patient's treatment. Each result is displayed in a timeline sorted by time with some information about the current step of the patient's treatment.

5. Discussion

The developed application prototype benefits patients, health professionals, and TB researchers. It gives patients a more transparent way of how parts of the treatment work. Although they are not end users, the application can be a bridge between the healthcare professional and the patient. The healthcare professional can visually and simply show why specific treatment measures were taken [34].

For professionals, OuTB can facilitate understanding the patient's general state of health, as the application provides algorithms to aid in the diagnosis. For TB researchers, the application becomes a tool capable of implementing and enhancing tuberculosis-related studies and allowing more excellent data collection that can be used in future research [35].

OuTB, unlike other developed applications, brings current front-end development technologies and has the potential to bring together several solutions capable of assisting in treating tuberculosis all on a single platform. Some challenges exist in the context of tuberculosis applications [36] for their implementation and adherence.

One of the technical problems is the need for interoperability of health systems [37] since the application depends on the connection between systems such as TBWeb. There is also a need to validate the prototype, where the minimum viable product (MVP) improvement is necessary, and carry out field tests so health professionals can completely understand its usability.

Therefore, the ideal scenario for OuTB would be to evaluate what has already been developed regarding interface, functionalities, and technology, define which systems and protocols will be used, propose an MVP, and test the application to receive feedback from health professionals. In addition, carrying out training for these professionals to facilitate the handling of the application would be necessary for adapting and understanding this new tool.

6. Conclusion

This study achieved the objective of developing a mobile application prototype capable of visually making the results of tuberculosis outcomes available. As OuTB is a prototype, several technical requirements need improvement.

Building an API that allows communication between the front and backend and constructing the backend itself are necessary activities in future work. Therefore, from the concept of SPA, the product of this work is the availability of a flexible development structure. Through the interface design, it is expected that the planning of other application areas will be favored, such as the database modeling and the backend. Fostering such tasks can also contribute to the definition of business rules that, once implemented on the server side, can subsidize an organized patient-centered tuberculosis database.

Acknowledgements

This work was supported by the São Paulo Research Foundation (FAPESP) - grant number 2020/01975-9, coordinated by author DA.

References

[1] Danilova I, Granich R, Williams P, et al. World TB Day-March 24, 2001. MMWR Morbidity and Mortality Weekly Report. 2001;50(11):201-204.

[2] Glaziou P, Floyd K, Raviglione MC. Global epidemiology of tuberculosis. Seminars in Respiratory and Critical Care Medicine. 2018;39(3):271-285.

[3]World TB Day — March 24, 2019, MMWR. Morbidity and Mortality Weekly Report. 68 (2019) 257. doi:10.15585/mmwr.mm6811a1

[4] Nery JS, Rodrigues LCS, Rasella D, et al. Effect of Brazil's conditional cash transfer programme on tuberculosis incidence. The International Journal of Tuberculosis and Lung Disease. 2017;21(7):790-796.

[5] Kraunt R. Unesco policy guidelines for mobile learning. Paris (FR): United Nations Educational, Scientific and Cultural Organization (UNESCO); 2013 [cited 2022 Dec 11].

[6] Feijó VC, Gonçalves BS, Gomez LSR. Heurística para avaliação de usabilidade em interfaces de aplicativos smartphones: utilidade, produtividade e imersão. Design & Tecnologia. 2013;6(Suppl. 1):33-42.

[7] Laserson KF, Thorpe LE, Leimane V, et al. Speaking the same language: treatment outcome definitions for multidrug-resistant tuberculosis. Int J Tuberc Lung Dis. 2005;9(6):640-645.

[8] Nery C, Britto V. Internet já é acessível em 90,0% dos domicílios do país em 2021 | Agência de Notícias –IBGE. 2022.

[9] Silva RH, Gatti MAN, Marta SN, Marafon RGC, Gatti Neto GG, Andrade EB de O de, et al. Aplicativos de saúde para dispositivos móveis: Uma revisão integrativa / Health applications for mobile devices: An integrative review. Brazilian Journal of Health Review. 2020;3(5):11754–65.

[10] Konduri N, Sawyer K, Nizova N. User experience analysis of e-TB Manager, a nationwide electronic tuberculosis recording and reporting system in Ukraine. ERJ Open Res 2017;3:00002–2017. doi:10.1183/23120541.00002-2017.

[11] Fraser HSF, Jazayeri D, Mitnick CD, Mukherjee JS, Bayona J. Informatics tools to monitor progress and outcomes of patients with drug resistant tuberculosis in Peru. Proc AMIA Symp 2002:270–4

[12] OpenMRS. MDR-TB Module. n.d.

[13] Costa WDL, Santos LRA dos, Rodrigues L, Monedeiro BS, Prado G, Crepaldi NY, et al. A free and open-source computer vision system applied to tuberculosis treatment, 2018.

[14] Crepaldi NY, Lima VC, Bernardi FA, Albano Santos LR, Yamaguti VH, Pellison FC, et al. SISTB: an ecosystem for monitoring TB. Procedia Computer Science. 2019;164:587–94.

[15] Pressman RS. Chapter 2, Topic 2.3.3. In: Pressman RS, Software engineering: a practitioner's approach. New York: McGraw-Hill; 7 ed. 2001. p. 42-45.

[16] Figma. Connects everyone in the design process so teams can deliver better products, faster.

[17] Gavrila V, Bajenaru L, Dobre C. Modern single page application architecture: a case study. Studies in Informatics and Control. 2019;28(2):231-238.

[18] OpenAPI Initiative. OpenAPI Initiative website.

[19] NpmTrends. Angular vs Ember CLI vs React vs Solid.js vs Svelte vs Vue

[20] Saks E. JavaScript Frameworks: Angular vs React vs Vue

[21] React Router: Declarative Routing for React.js.

[22] Redux: A Predictable State Container for JS Apps.

[23] Vue Router: The official Router for Vue.js - Expressive, configurable and convenient routing for Vue.js.

[24] Vuex. Vuex documentation. Available from: https://vuex.vuejs.org/. Accessed June 12, 2023.

[25] Framework7. Build full featured iOS, Android & Desktop apps. 2014.

[26] Bootstrap. Build fast, responsive sites with Bootstrap. 2014 [cited 2023 Feb 27].

[27] Vite.js: Next Generation Frontend Tooling - Get ready for a development environment that can finally catch up with you.

[28] Cordova: Mobile apps with HTML, CSS & JS - Target multiple platforms with one code base - Free and open source.

[29] Bosnic S, Papp I, Novak S. The development of hybrid mobile applications with Apache Cordova. In2016 24th Telecommunications Forum (TELFOR) 2016 Nov 22 (pp. 1-4). IEEE.

[30] Node.js. An open-source, cross-platform JavaScript runtime environment.

[31] Express: Fast, unopinionated, minimalist web framework for Node.js.

[32] Prisma: Next-generation Node.js and TypeScript ORM.

[33] PostgreSQL: The World's Most Advanced Open Source Relational Database.

[34] Moore J. The benefits of mobile apps for patients and providers. British Journal of Healthcare Management. 2012 Sep;18(9):465-469. Available from: https://doi.org/10.12968/bjhc.2012.18.9.465.

[35] Siegler A, Knox J, Bauermeister J, Golinkoff J, Hightow-Weidman L, Scott H. Mobile app development in health research: pitfalls and solutions. mHealth. 2020 Jul;7.

[36] Iribarren SJ, Schnall R, Stone PW, Carballo-Diéguez A. Smartphone Applications to Support Tuberculosis Prevention and Treatment: Review and Evaluation. JMIR Mhealth Uhealth. 2016;4(2):e25 doi: 10.2196/mhealth.5022.

[37] Pellison FC, Rijo RPCL, Lima VC, Crepaldi NY, Bernardi FA, Galliez RM, Kritski A, Abhishek K, Alves D. Data Integration in the Brazilian Public Health System for Tuberculosis: Use of the Semantic Web to Establish Interoperability. JMIR Med Inform. 2020 Jul 6;8(7):e17176. doi: 10.2196/17176.