WEB-BASED IMMUNITY PROCESS MANAGEMENT SYSTEM

Asst.Prof Agalya M, Kaviya.R, Kaviya.V, Manasha.A, And MohanaPriya.M

1 Computer Science and Engineering, Vivekanandha College Of Technology for Women, Namakkal, Tamil Nadu 613 006, India

2 Computer Science and Engineering, Vivekanandha College Of Technology For Women, Namakkal, Tamil Nadu 613 006, India

3 Computer Science and Engineering, Vivekanandha College Of Technology For Women, Namakkal,, Tamil Nadu 613 006, India

4 Computer Science and Engineering, Vivekanandha College Of Technology For Women, Namakkal,, Tamil Nadu 613 006, India

5 Computer Science and Engineering, Vivekanandha College Of Technology For Women, Namakkal,, Tamil Nadu 613 006, India

ABSTRACT

This paper describes the creation of a web-based Immunity Test Management System intended to support home-based immunity testing in response to the growing need for remote healthcare solutions. Microsoft SQL Server is used for backend data management, C# is used for server-side logic, and the ASP.NET Framework is used for the application's front end. Without having to go to a medical facility, people may safely register and log in, schedule visits, and obtain customized immunity reports thanks to the system. A designated healthcare professional comes to the user's home to take a blood sample when an appointment is scheduled. For lab processing and traceability, the

application automatically generates a unique sample number that is labelled on the test tube. The employee inputs four key immunity parameters into the system after analysis. After processing these values, a visual report based on a scoring scale ranging from 1 to 7 in increments of 0.5 is shown to the user. A color-coded advice system is used by the app: green indicates robust immunity (score above 5), orange indicates the need for a healthy diet (scoring between 3 and 4.5), and red indicates a medical consultation (score below 3). The complete immunity testing procedure is streamlined by this application, improving accessibility and efficiency from digital reporting to at-home sample collection. The system is scalable for healthcare organizations wishing to deliver decentralized testing services because of its automation and intuitive user interface.

KEYWORDS: Immunity Test, Web Application, ASP.NET Framework, C# Programming, Microsoft SQL Server, Remote Healthcare.

1 INTRODUCTION

The necessity of digital transformation in the healthcare industry has grown in recent years, especially in light of international health emergencies like the COVID-19 pandemic. People in remote or rural locations sometimes find traditional healthcare models, which necessitate in-person visits to hospitals and diagnostic facilities, to be inconvenient and time-consuming. Online health services that allow consumers to conduct diagnostic procedures from the convenience of their homes are becoming more and more popular as a result. In addition to reducing patient exposure in clinical settings, remote testing and consultation aid in the early identification and tracking of medical disorders.

A web-based tool called the Immunity Test Management System was created to solve these issues by providing a completely online platform for scheduling and

overseeing immunity testing. This system allows customers to register, log in securely, schedule appointments, and obtain comprehensive test reports remotely. It was built with the ASP.NET Framework for the frontend, C# for the backend logic, and Microsoft SQL Server for database management. An employee is assigned to visit the user's home, take a blood sample, and send it back to the lab when the user schedules a test. To guarantee precise tracking and identification of the sample during the testing procedure, the system generates a unique sample number.

The worker enters four crucial immunity levels into the system after the lab analyses the blood sample. An average immunity score is then computed using these values. The user receives a color-coded report from the system based on this score: green indicates strong immunity (scoring above 5), orange indicates moderate immunity (score between 3 and 4), and red indicates low immunity (score below 3). Depending on the user's immunity level, the system offers personalized health recommendations in addition to the visual representation, such as food guidance or medical consultation. With this method, consumers can take charge of their health management without requiring ongoing medical supervision.

The Immunity Test Management System offers a scalable and easily accessible remote diagnostics solution by combining automation, real-time data processing, and user-centric reporting. Through effective appointment scheduling, safe data processing, and straightforward result interpretation, it closes the gap between patients and healthcare professionals. This method is a useful tool for contemporary healthcare delivery since it not only improves the standard of care for patients but also streamlines the workflow for medical personnel.

2 OBJECTIVES

The Immunity Test Management System's main goal is to give users a smooth, remote immunity testing experience while guaranteeing precise recording and reporting of diagnostic data. The system's specific goals are to help staff members take blood samples straight from users' homes, automate sample number production to ensure traceability, and allow users to register and schedule immunity tests online. Processing and storing immunity test data safely, analysing the four main immunity values, and displaying the results in an understandable, color-coded manner—green for strong immunity, orange for moderate, and red for low—are other important objectives. Based on these results, the system also seeks to offer tailored health suggestions. The application protects data integrity and guarantees the safe handling of private health information by implementing role-based access controls for administrators, staff, and users. The project's overall goals are to improve user experience, lower manual error rates, and promote public health by raising awareness of immune health issues and detecting them early.

3 LITERATURE SURVAY

Access to healthcare has been substantially improved by the development of digital health technologies, especially in areas like decentralized testing and remote diagnostics. Kesara, Jonas, and Schulman (2020) talked about how the COVID-19 pandemic hastened the worldwide telehealth trend and increased the acceptance of remote care. The integration of wearable technology with Electronic Health Records (EHRs) was emphasized by Dinh-Le, Chuang, Chokshi, and Mann (2019), who emphasized the necessity of systems that simplify data collection and display [1].

In clinical settings, health information systems (HIS) are now widely used for organizing lab results, patient records, and appointments. In their investigation of the relationship between big data analytics and healthcare, Mehta and Pandit (2018) highlighted the ability of web-based apps to produce useful insights. Many such systems, nonetheless, are still hospital-focused and do not offer diagnostic services that may be obtained at home. The World Health Organization (2019) has promoted digital health initiatives that extend the provision of services beyond of conventional clinical settings, particularly in low- and middle-income nations [2].

Using ASP.NET and Microsoft SQL Server, Kumar and Tripathi (2020) created a hospital administration system that demonstrated how online technologies may be leveraged to improve hospital procedures. However, their use was limited to in-clinic procedures and did not include modules for immunity scoring or home testing. In their analysis of the effects of digital triggers in patient-centered health interventions, Muench and Baumel (2017) emphasized the potential of visual cues, such color-coded alerts, to improve user engagement and health outcomes [3].

Despite advances in technology, little is known about systems that oversee immunity testing from start to finish, particularly those that facilitate automated immunity analysis and remote sample collection. By providing a complete web-based solution that includes appointment booking, sample tracking, result processing, and patient-facing reporting, our project fills that gap. The platform is ideally situated to satisfy the needs of a post-pandemic healthcare environment where accessibility and remote testing are critical [4].

Zhang et al. (2021) made a significant contribution to the field of digital healthcare by investigating the creation of mobile health (mHealth) platforms for the management of

chronic illnesses. According to their findings, self-monitoring devices with well-organized, user-friendly interfaces greatly improve patient involvement and data accuracy. Similar to this, Lee and Kesselman (2019) looked at how easy it is to use healthcare portals and stressed that the overall efficacy of digital diagnostic tools is increased when results are presented clearly and actionable insights are included, like dietary recommendations or alerts. These results are consistent with the Immunity Test Management System's objectives, which include helping users understand their immunity status and take appropriate action through an easy-to-use interface and color-coded reports [5].

4 METHODOLOGIES

The Immunity Test Management System was developed using a structured software engineering approach, focusing on functionality, usability, and scalability. The system adopts a modular architecture and follows a role-based access control model. The core technologies used include the ASP.NET Framework for the frontend, C# for backend logic, and Microsoft SQL Server for database management.

4.1 SYSTEM DEVELOPMENT LIFECYCLE

The development followed **a** Waterfall Model, as the project requirements were clearly defined from the outset. The major phases include:

- **REQUIREMENT ANALYSIS**: Functional and non-functional requirements were gathered, including user registration, appointment scheduling, employee assignment, sample tracking, and report generation.
- **SYSTEM DESIGN**: Database schema, user interface wireframes, and process flow diagrams were designed.

- **IMPLEMENTATION**: Coding was done using ASP.NET Web Forms and C#. SQL Server was used to create and manage tables for users, appointments, samples, and reports.
- **TESTING**: Unit testing, integration testing, and user acceptance testing (UAT) were carried out to ensure system correctness.
- **DEPLOYMENT**: The system was deployed on a local server environment with options for cloud hosting.

4.2 USER ROLES AND ACCESS CONTROL

The application supports three main roles:

- **USER**: Registers, logs in, books appointments, and views immunity reports.
- **EMPLOYEE**: Views assigned appointments, collects blood samples, and inputs immunity values after lab processing.
- **ADMIN**: Manages users, employees, appointments, and monitors the testing process.

4.3 PROCESS FLOW

- **USER REGISTRATION AND LOGIN**: Users create an account and log in using secure credentials.
- **APPOINTMENT BOOKING**: Users can book appointments by selecting a preferred date and time.
- **EMPLOYEE ASSIGNMENT**: The admin assigns an available employee to each appointment.
- **SAMPLE COLLECTION AND TRACKING**: At the user's home, the employee collects a blood sample. The system automatically generates a **unique sample number**, which is written on the test tube.

- **LAB ENTRY**: Once testing is complete, the employee inputs the four immunity values into the system.
- **REPORT GENERATION**: The system calculates an average immunity score (scale of 1 to 7) and classifies it into:
    - **Green** (Score > 5): Strong immunity.
    - **Orange** (Score 3–4.5): Moderate immunity, suggests healthy diet.
    - **Red** (Score < 3): Weak immunity, recommends doctor consultation.
- **REPORT ACCESS**: Users can log in and view/download their immunity report with color-coded recommendations.

4.4 TECHNOLOGIES USED

COMPONENT	TECHNOLOGY
Frontend	ASP.NET Framework
Backend	C# (ASP.NET Code-Behind)
Database	Microsoft SQL Server
Hosting	IIS / Localhost Server
Tools Used	Visual Studio, SSMS

4.5 DATA FLOW AND SECURITY

- All user and sample data are stored in encrypted format where applicable.
- SQL parameterized queries are used to prevent SQL injection.
- Sessions are used to maintain secure user login states.

4.6 TESTING

Testing was conducted at every stage, starting with unit tests for individual modules, followed by integration testing. Role-based access was tested to ensure each user could only access permitted functionalities. User acceptance testing (UAT) was carried out with test cases for:

- Login validation
- Task creation and assignment
- Task update tracking
- Dashboard visualization
- Report accuracy

5 IMPLEMENTATIONS

The Immunity Test Management System was implemented using a combination of robust technologies including the **ASP.NET Framework** for frontend development, **C#** for server-side logic, and **Microsoft SQL Server** for data storage and management. The system is designed with a layered architecture that ensures scalability, maintainability, and security.

5.1 USER INTERFACE DESIGN (FRONTEND)

- Developed using **ASP.NET Web Forms**, ensuring compatibility with most browsers and devices.
- Designed with a **mobile-first approach** using **Bootstrap**, enabling users to access the system from smartphones, tablets, and desktops.
- Key user modules include:
  - User Registration and Login
  - Appointment Booking Interface
  - Live Status Tracking Dashboard
  - Report Viewing Panel
- The UI is simple and intuitive to ensure accessibility for non-technical users, especially in rural or underserved areas.

5.2 BACKEND AND BUSINESS LOGIC (C#)

- **C#** is used to handle core logic such as:
  - Validating user inputs
  - Generating unique sample numbers using guids
  - Assigning employees automatically based on availability
  - Managing test result calculations and interpretations
- Implements a three-tier architecture**:**
  - Presentation layer (asp.net forms)
  - Business logic layer (c# classes and services)
  - Data access layer (ado.net & sql commands)
- All processes are executed asynchronously where necessary to improve performance and responsiveness.

5.3 DATABASE MANAGEMENT (MSSQL SERVER)

- A normalized relational database schema with tables such as:
  - Users, Employees, Appointments, Samples, Test Results, Login Credentials, etc.
- Entity Relationships were carefully designed to maintain referential integrity between samples and appointments.
- Stored procedures and parameterized queries were used to avoid SQL injection and improve performance.
- Sample No. Mapping**:** The application automatically generates and stores a unique Sample ID for every booked test.

5.4 ROLE-BASED ACCESS CONTROL (RBAC)

- Three main user roles:
  - **User**: Can register, book appointments, and view test results.
  - **Employee**: Can view assigned appointments, collect samples, enter test values, and update statuses.
  - **Admin**: Has full system access for monitoring, assigning tasks, managing users, and viewing all test data.
- Implemented using **ASP.NET Identity Framework** with secure session management and password encryption.

5.5 APPOINTMENT AND SAMPLE TRACKING MODULE

- Employees receive automated notifications of new assignments.
- Sample statuses are updated in real-time from "Scheduled" → "Collected" → "In Testing" → "Completed".
- Each sample is physically labeled with the system-generated Sample No. to prevent mismatches in the lab.

5.6 IMMUNITY TEST CALCULATION AND REPORT LOGIC

- Employees enter four core immunity values:
  - The system calculates an average immunity score (e.g., 3.5).
  - The result is color-coded**:**
    - Green**:** >5 (Healthy)
    - Orange: 3–4 (Needs Healthy Diet)
    - Red: <3 (Consult Doctor)
- Generated reports include:
  - Numerical values
  - Color status
  - Suggestions based on score
- Reports are displayed using graphical elements and HTML tables for easy understanding.

5.7 SECURITY AND DATA PROTECTION

- HTTPS protocol used to secure communication.
- Password hashing and salting using SHA256 for stored credentials.
- Session timeout and logout on inactivity are implemented for extra security.
- Sensitive data is protected by:
  - Field-level access controls
  - Audit logs for admin activities

5.8 TESTING AND DEBUGGING

- Unit testing was performed for all backend modules.
- Integration testing ensured smooth communication between frontend and backend components.
- Manual testing by volunteers simulated real-time scenarios such as booking appointments, entering values, and generating reports.

5.9 DEPLOYMENT

- The system was deployed on a Windows Server using IIS (Internet Information Services)**.**
- The database was hosted on Microsoft SQL Server Express Edition with backup and recovery options.
- Future deployments are planned for cloud platforms like Azure to improve scalability and availability.

Figure 1: *main Page*

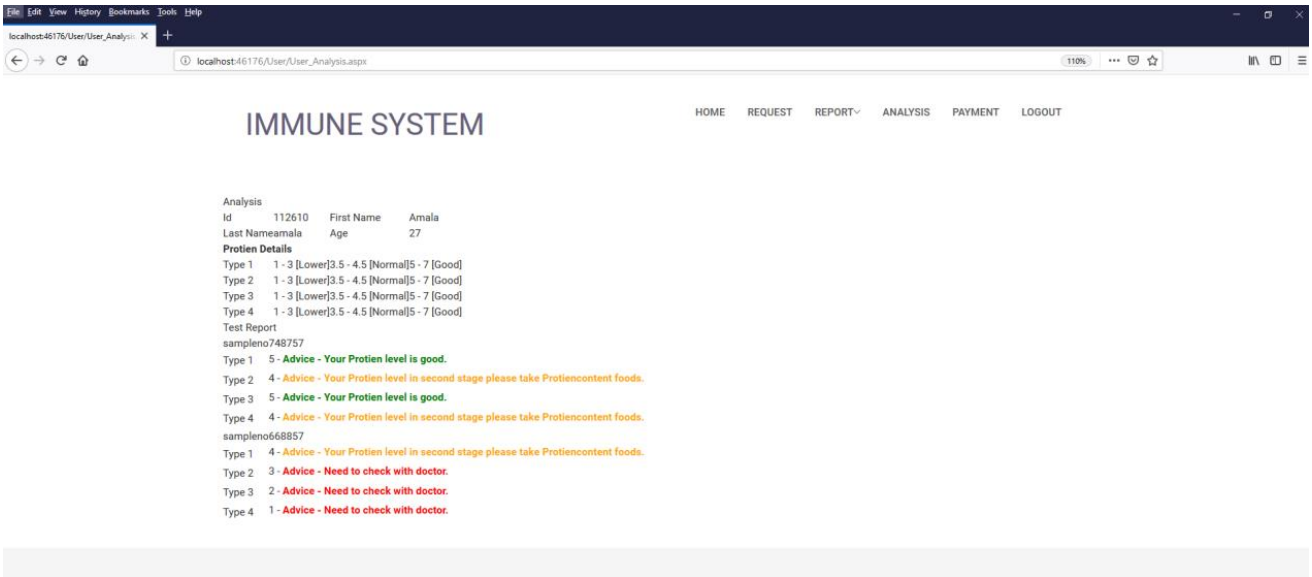

Figure 2: Report

6 CONCLUSIONS

The suggested Immunity Test Management System effectively illustrates a workable and expandable way to make remote immunity testing possible. Without going to a medical facility, users can register, schedule appointments, and obtain diagnostic reports via this program. The automated process guarantees testing accuracy and consistency from sample number generation to report creation. Using a color-coded scale, the system analyzes four immunity parameters and displays the results: red denotes low immunity, orange denotes moderate immunity, and green denotes strong immunity. When paired with health advice, this graphic portrayal encourages people to take charge of their health. Furthermore, data security and operational efficiency are improved by integrating role-based access for administrators, staff, and users. This system is a useful tool for decentralized healthcare delivery and digital health innovation since it uses ASP.NET, C#, and MSSQL Server to provide a stable and secure platform.

7 REFERENCES

1. H. Dinh-Le, R. Chuang, S. Chokshi, and D. Mann, "Wearable Health Technology and Electronic Health Record Integration: Scoping Review and Future Directions," *JMIR mHealth and uHealth*, vol. 7, no. 6, p. e12861, 2019. DOI: 10.2196/12861
2. World Health Organization, *WHO Guideline: Recommendations on Digital Interventions for Health System Strengthening*. Geneva: World Health Organization, 2019. [Online]. Available: https://www.who.int/publications/i/item/9789241550505
3. F. Muench and A. Baumel, "More Than a Text Message: Dismantling Digital Triggers to Curate Behavior Change in Patient-Centered Health Interventions," *JMIR Mental Health*, vol. 4, no. 2, p. e10, 2017. DOI: 10.2196/mental.7637
4. J. Keesara, A. Jonas, and K. Schulman, "Covid-19 and Health Care's Digital Revolution," *New England Journal of Medicine*, vol. 382, no. 23, pp. e82(1)-e82(3), 2020. DOI: 10.1056/NEJMp2005835
5. [5] L. Zhang, R. Lee, and A. Kesselman, "Usability of Mobile Health Applications for Chronic Disease Management: A Systematic Review," *Health Informatics Journal*, vol. 27, no. 1, pp. 146–167, 2021. DOI: 10.1177/1460458220940504.
6. L. Sacks, "Digital Health Technologies & Decentralized Clinical Trials," U.S. Food and Drug Administration, 2022. [Online]. Available: https://www.fda.gov/media/167542/downloadU.S. Food and Drug Administration
7. A. Chowdhury and A. Pick, "Digital Health in LLMICs: Current and Future Technological Developments," Pathways for Prosperity Commission Background Paper Series, no. 28, 2019. [Online]. Available: https://pathwayscommission.bsg.ox.ac.uk/sites/default/files/2019-10/digital_health_in_llmics.pdfPathways for Prosperity
8. A. Baumel, K. Faber, N. Mathur, J. M. Kane, and F. Muench, "Enlight: A Comprehensive Quality and Therapeutic Potential Evaluation Tool for Mobile and Web-Based eHealth Interventions," *Journal of Medical Internet Research*, vol. 19, no. 3, p. e82, 2017. DOI:

10.2196/jmir.7273dplab.haifa.ac.il

9. H.-R. Han et al., “Using Patient Portals to Improve Patient Outcomes: Systematic Review,” *JMIR Human Factors*, vol. 6, no. 4, p. e15038, 2019. DOI: 10.2196/15038JMIR Human Factors
10. L. Zhang, R. Lee, and A. Kesselman, “Usability of Mobile Health Applications for Chronic Disease Management: A Systematic Review,” *Health Informatics Journal*, vol. 27, no. 1, pp. 146–167, 2021. DOI: 10.1177/1460458220940504