

CSCI 409 Senior Project II Final Report:

Medlink: Unified Medical Platform for Patients and Doctors

Alikhan Nashtay, Bolat Labakbay, Nurbolat Satybaldy, Zhibek Yessilzhankyzy, Ilyas Telman

Department of Computer Science

Nazarbayev University

Astana, Kazakhstan

alikhhan.nashtay@nu.edu.kz, bolat.labakbay@nu.edu.kz, nurbolat.satybaldy@nu.edu.kz,

zhibek.yessilzhankyzy@nu.edu.kz, ilyas.telman@nu.edu.kz

1. EXECUTIVE SUMMARY

Medlink is an application created within the domain of telemedicine that use the power of nowadays computing powers of programming to revolutionize way patients can connect with their physicians. This project attempts to resolve the issues that arise from existing healthcare applications which do not properly balance user accessibility, physician productivity, communications security, and medical confidentiality. The primary aim of this project is to create an integrated solution which will allow patients to browse doctors, schedule appointments, communicate and access medical information, while simultaneously allowing doctors to manage appointments, profiles, and information, and administrative staff to control users' access. The approach taken to developing the system involves combining elements of system design, iterative development, and prototyping. The project used an Agile development methodology to iteratively refine the system across two consecutive semesters, creating a full-stack application which included Go-based backend, PostgreSQL database, React/Next.js web interface, and planning support for Swift/SwiftUI mobile app. The process involved creating system architecture that included RESTful APIs, WebSocket-based chat, JWT authentication, roles-based permissions, and modularization. The entire development process was managed via GitHub version control and Notion work management tool. The key results obtained during the project include creating a functional prototype out of the designed application. At this stage of development, Medlink includes registration and authentication processes, doctor verification, organization and department management, scheduling appointments with the doctor, and even a basic real-time chat functionality. Furthermore, this project laid foundation for many more advanced features such as WebRTC-powered video consultation, structured electronic health record system, notifications, and other. Thus, the outcomes of the project clearly show that the objectives were met and the computing-based solution

was indeed implemented and evaluated.

2. INTRODUCTION

A. Overview of the Project

Medlink is integrated telemedicine solution, which aimed at enhancing interaction of patients and physicians remotely by providing a secure, accessible, and well-organized digital space. One of the key gaps that the project fills in most of the existing telemedicine solutions is that remote consultancy tools have been on the rise, but some lack both patient-friendliness and efficiency in workflow and a high degree of privacy and trust in a single consistent framework [1], [3], [6]. Telemedicine cannot be perceived in healthcare facilities as a video-calling option only. It rather needs to be deployed as a holistic computing-based method that facilitates the management of the appointment, managing the medical-record, providing secure communications, administrative control and regulatory awareness [3], [6], [7]. The idea of Medlink is preconditioned by the increasing significance of digital access to healthcare and the necessity to provide platforms which could help patients and medical workers in a stable and systematized way. Throughout the previous project documents, the project was always described as a system where patients can find doctors, book appointments, keep longitudinal medical records, and remotely communicate, and where physicians can manage professional profiles, schedules, consultations, prescriptions and patient data using the same infrastructure. Furthermore, administrative users perform duties of doctor verification, organizational management, and audit logging which enhances trust, accountability and compliance in the platform. These combined designs are based on the key purpose of the project to provide a safe and practical telemedicine setting that does not jeopardize professional workflow and data confidentiality to enhance access to healthcare. The importance of Medlink is that it is consistent with the design, implementation and assessment of a computing-based solution. The platform is not restricted to just user-to-user communication,

but rather, incorporates the use of backend services, user interfaces, authentication, scheduling, messaging, video support, medical records, and administration tools into one software platform. This renders the project a feasible case study of how system requirements, system architecture, iterative system implementation and system performance can be used to address a real-world healthcare issue using the principles of software engineering [1], [3], [6]. In line with this, this report introduces Medlink as a complete telemedicine system that was developed in two semesters, and it talks about the background, constraints, methodology, project execution, evaluation and conclusions as per the final report rubric.

B. Objectives and Goals

Medlink has a background which could be summarized by the following key points:

- Expansion of telemedicine: Telemedicine has taken a significant role in the healthcare delivery process particularly following the COVID-19 era since it enhances care accessibility, lessens commuting time, and enables care continuity [1].
- Integration through platforms: A large number of current solutions are based on a single functional area, e.g. communication or booking appointments, yet successful telemedicine involves an integrated environment that integrates consultation, appointment making, data management and administrative control [3], [6].
- Value of privacy and trust: The medical information is very sensitive, and effective telemedicine systems should provide protection of authentication, access control and privacy preserving communication to win the trust of patients and physicians [6], [8].
- Patient portal and digital records role: The study demonstrates that patient engagement, satisfaction, and communication with healthcare providers could be enhanced with the help of digital patient portals and electronic health-record access, and record management becomes a crucial component of the telemedicine system today [3], [7].
- Requirement of real-time communication: An actual telemedicine system must provide more than just on-demand information access but allow real-time communication including chat and video-conferencing. WebSockets and WebRTC are some of the technologies that can be applied in healthcare systems that need low latency communications [4].
- Administrative and organizational needs: The technology is not the only way to establish trust in medical systems but also governance. Accountability and safer clinical interactions are facilitated by such features as doctor verification, audit logs, and management of healthcare organizations.
- Computerish nature of the solution: Medlink was developed as an all-software solution with a Go backend, PostgreSQL database, React/Next.js web frontend, and mobile support (Swift based). The architecture of this project is a representation of its objective to address a healthcare-access issue with software design, implementation, and evaluation, as opposed to a single, isolated capability.

In general, the project background indicates that Medlink was created due to the practical necessity of healthcare and technical opportunity. Current studies focus on the advantages of telemedicine, however, they also bring up the significance of usability, privacy, secure communication, and reliable clinical processes [1], [3], [6], [7]. Medlink addresses these issues by offering an all-in-one solution in the form of an integrated platform that would integrate remote consultation, schedule management, record keeping and administrative controls into a single computing-based platform that would meet the needs of contemporary practice of telemedicine.

3. IDENTIFICATION OF CONSTRAINTS

Medlink design and implementation was influenced by a combination of set constraints which influenced not only technical choices, but the scope of the project, implementation plan and project boasting of readiness. Economically and resource wise the project was made within the constraints of a senior-project in an academic environment. The kick off materials explicitly acknowledge the potential of limited hardware and server resources and uses cloud or university-hosted infrastructure as backup resources. This meant that the system architecture would need to be viable to a prototype that would run, showcase and be serviced without an enterprise scale infrastructure. This has been a limiting factor, leading to the use of a modular monolithic backend, containers, and pragmatics-limited scope of features instead of a costlier or operationally constraining distributed design. Social and usability constraints were also significant to influence the project. Telemedicine has the potential of greatly enhancing access to care as it is able to lessen the travel load, provide continuity and permits patients to meet with the doctors remotely [1], [3]. Simultaneously, the project documents acknowledge the fact that digital healthcare systems could be inadvertently unfriendly to all people, who have poor technical skills, lack of internet connection, or unpredictable internet access. This limitation manifests itself in Medlink as a social problem (that the interface should be comprehensible to patients and clinicians) and a practical design (because the interface should not be excessively complicated, the system should support workflows in a basic form). The new requirements also presuppose the stable internet service and initially limit

the scope of platforms to web and iOS, which is a highly reasonable scope control choice in terms of development, but, on the other hand, creates limitations when it comes to more comprehensive access and application to all groups of users. Another group of constraints is related to the environment and sustainability. Although the project materials lack a quantitative analysis of the environmental analysis, it suggests that remote consultation may decrease the travel of the patients and, consequently, the environmental cost of transportation. In this respect, Medlink could be viewed as benefiting a more economical model of acquiring health care, particularly with regular consultations and the one that does not need face-to-face contact. This possible sustainability advantage, however, needs to be displayed cautiously in the final report, as it is not a direct advantage that was directly measured during the project, but an inferred advantage. The shorter-term sustainability aspect of the project itself has to be software sustainability the system should be maintainable, expandable, and useful after the semester to continue to be a valuable result of the engineering activity. Political, regulatory, ethical and legal are the greatest limitation, though. The revised requirements directly mention the requirements of GDPR/HIPAA-like requirements, audit history, patient consent, telemedicine practice based on privacy consideration, and control by the administrator. These are not optional, but domain requirements that have influenced the overall design of the platform. According to the U.S. Department of Health and Human Services, the HIPAA Security Rule requires that electronic protected health information ought to be safeguarded by employing appropriate administrative, physical and technical controls whereas the GDPR outright stipulates that data security must be designed and defaulted in systems which handle personal data. These guidelines directly reinforce the focus of Medlink on access control and secure authentication, audit logging and controlled processing of medical records. Health, and safety, and ethical restraints are also crucial. The initial-semester report is focused on patient privacy, doctor checking, accountability, non-recording of consultations without the patient consent and necessity to make the users informed about the boundaries of telemedicine. In healthcare in particular these are critical as design errors can not only decrease usability but can also impact the levels of trust, confidentiality and clinical safety. To do so, the administrative doctor verification, audit trails, role-based access, and privacy-sensitive communication practices will be part of the basic features of Medlink instead of additional improvements. Also recognized in the project is the fact that telemedicine cannot completely substitute face-to-face medical treatment in all situations; thus, one of the ethical needs of the system is to ensure that one

communicates those shortcomings to users. Lastly, the manufacturability aspect of the rubric can be understood in relation to this software project as the implementability, deployability and maintainability. The system was to be fully constructed and testable by a student team in two semesters and also show engineering discipline. This had an impact on Docker Compose, health checks, one-line structured logs, database constraints and backups, and a modular monolithic codebase. These design decisions have enhanced recoverability and maintainability, and are indicative of a trade-off: the project is not intended to be full production scale of complexity, but rather to be realistic in implementation and academically feasible. This is the reason why the constraints have to be placed in the final report not as by-products, but as design motivators which influenced the authentication policy, time management, choices in infrastructure and security controls and general system readiness understanding.

4. BACKGROUND AND RELATED WORK

The Medlink project falls within larger scope of telemedicine, digital patient portals, and electronic medical records, and real-time healthcare communication that is secure. Existing literature provides consistent evidence that telemedicine has the potential to enhance the access to healthcare services by lowering the geographic obstacles, wasting less time on travelling, and ensuring continuity of care, especially when it comes to follow-up appointments and regular contacts [1]. This extended background is what gives Medlink its background: telemedicine is not a new concept that is being put forth in the project, but it is instead attempting to resolve the practical issue of how to make it a reliable and useful computing system that would benefit both patients and physicians. Concurrently, the literature also reveals that adoption of telemedicine is limited due to lingering issues over privacy, security, usability, and trust. The initial-semester report rightly outlines these as the constant obstacles in the field and adopts them to support the design of the integrated project. This is a critical consideration to the final report, most telemedicine systems are successful in delivering remote communication, but fewer have succeeded in integrating secure identity management, organized patient records, appointment scheduling and administrative controls into a unified workflow. The contribution of Medlink can be thus not merely as an app in telemedicine, but a system-level effort to balance between convenience and professional reliability and privacy protection [1], [6], [7]. One of the key areas of similar research is patient portals and access to electronic health-record. Research mentioned in the first-semester materials suggests that patient engagement, patient satisfaction and enhanced communication between patients and healthcare

providers are possible through the use of digital patient portal [3], [7]. This literature is particularly applicable to Medlink since the project does not end upon messaging or video calls; the project also involves longitudinal patient records, document upload, information related to prescription and access to appointment history. That is to say, Medlink is more in line with the literature on integrated, digital healthcare platforms than it is with the usage of narrow, teleconferencing tools. This can be used to explain why it is better to be in one architecture than to have communication, scheduling and records management as separate functionalities. Another significant field of related research is the technology of real-time consultation, audio/video communication, as well as low-latency interaction. WebRTC is discussed as a suitable basis of the real-time, privacy-sensitive communication between patients and doctors in the project documentation and is methodologically justified. The current W3C standards still support WebRTC as a web standard of real-time communication in the browser, which can be used by Medlink to use WebRTC in video sessions and WebSockets in real-time signaling and chat. This mix is suitable in a telemedicine context since it allows interacting in real-time and still can be integrated with the contemporary web-based architectures. The literature of security and privacy is also key to the justification of the project. Healthcare systems offer personal data that is very sensitive and any breach in confidentiality, integrity or access control can all lead to dire ethical, legal, and practical implications. This is addressed in the first-semester report, which talks about password hashing, JWT-based authentication, access control based on roles, audit trails, and consultations being privacy-conscious. These design decisions are consistent with the overall trend of health IT security practice where the systems need to ensure safety of data both on its way and in points of access, limit the visibility of records, and maintain accountability, using logs and governance controls. The significance of the safeguards is supported by regulatory frameworks like HIPAA and GDPR that underline the safe processing of health-related and personal information via organizational and technical means. In terms of systems, Medlink can be compared to narrower options that address just one aspect of the healthcare-access issue. There are those platforms that specialize in video consultation, some in booking of appointments and some in record management or secure messaging. Although these specialized tools could be effective in their small scope, it can fail to support the larger workflow needs of telemedicine, including verified identity of the physician, manage conflicts of appointments, auditing and interaction between roles. The integrated architecture of Medlink is thus one of its primary methodological advantages: differentiating

between scheduling, communication, records and administration into a single platform, it tries to offer a more comprehensive computing-based solution than the single-purpose ones. This comparison needs to be highlighted in the final report as it clarifies reasons why the selected architecture is not only justified technically, but in terms of restrictions of current categories of systems as well. Technical decisions made in the project also have a solid foundation on the terms of related-work. The chosen stack, a Go backend, PostgreSQL database, React/Next.js frontend, and Swift-based mobile support, represents a sensible trade-off between the performance, maintainability, responsiveness, and real-time support. The previous report uses this architecture as a layered full-stack, as opposed to an experiment prototype put together using unrelated components. It is that framing which ought to be maintained in the final report. Specifically, the proposed scale and resource limitations of the project justify the use of a modular monolithic backend, whereas WebSockets, JWT authentication, and structured persistence will help with integrated system behavior needed in telemedicine. This renders the methodology to be defensible as both an engineering option, and as a reaction to the workflow and trust requirements particular to the literature. In general, the background and related work demonstrates that Medlink lies at the crossroad of a few established lines of research and practice: access to telemedicine, the use of patient-portals, digital medical records, safe communications, and healthcare data governance. The literature justifies the necessity to have systems that are not only functional, but also reliable, usable, and privacy conscious [1], [3], [6], [7]. Medlink addresses this requirement by suggesting a built-in architecture where communication, scheduling, records and administrative control are considered components of a single computing-based solution as opposed to separate modules. This is the main rationale of the scope of the project and of the design choices that are detailed in the rest of the report.

5. METHODOLOGY AND PROJECT APPROACH

Medlink was developed using a combination of software engineering practice driven by iterative development, and a well-defined full-stack system architecture. In all the Fall 2025 design documentation, kickoff materials, and first semester final report, the project is continually defined as an application of a layered telemedicine platform with a Go-based backend, a PostgreSQL database, a React/Next.js webapp, and an intended Swift/SwiftUI mobile client. The need to meet core technical specifications of telemedicine such as secure data manipulation, real-time interaction, documented medical records, multi-based user access, and long-term sustainability did not contribute to the

arbitrary choice of this architecture. As shown in Figure 1, system architecture unites these components in single layered platform, with the backend, database, client applications, and with security mechanisms being components of a single computing-based solution.

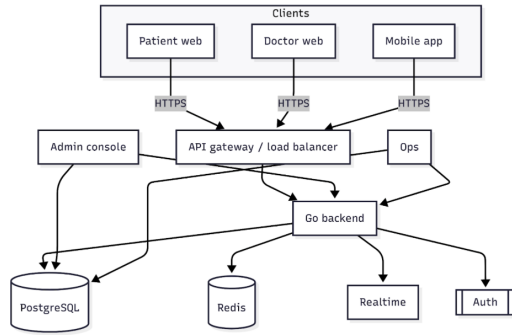


Fig. 1. Medlink System Architecture.

Medlink was developed as a layered technical architecture, yet as a feature-complete product, featuring unique patient, doctor and administration interfaces. The platform has dedicated route surfaces on the frontend side, which support role-conscious layouts and guarded navigation, and are dedicated to authentication, patient care, doctor practice, and administration. The patient side consists of organization discovery, the department and doctor search, booking of appointments, consultations, chat, records and profile management. The physician side comprises a home dashboard, practice management, department allocation, schedule definition, appointment lifecycle management, encounter documentation, prescription order and patient review. The administrator section has platform-level and organization-level governance capabilities like user management, doctor authentication, organization and department management, platform-admin management, and audit-log review. This separation of features aligns with the project objective of establishing a cohesive yet role-conscious telemedicine space where all stakeholders engage with the same platform via a workflow that is suitable to their permissions and roles. The role-based framework can be described with the help of Figure 2 where the patient and doctor and the admin interface should be displayed together and the primary route groups and access control paths throughout the system demonstrated. The session and access-control model have the same role-aware design. It is authenticated using JWT tokens, provides route protection using middleware and guarded layouts and not restricted to single role string but may be extended to capability-based access decisions and doctor-specific or administrator-specific permissions. On the backend level, it has both the standard middleware and dynamic

middleware, like authenticated-user checks, role-based access control, and doctor-context-injection of the authorised doctor operations. It is significant to Medlink since most of its workflows are role-dependent: such as only eligible users can create or administer an appointment, only doctors can use doctor-facing scheduling features, and only authorized administrators may administer verification, organizations, departments, or audit controls. These design decisions render the platform more secure and in line with the needs of governance of healthcare. Figure 2 should once again be used to highlight the relationship between guarded access and role specific surfaces since such a figure would assist the reader to associate access control with the real product structure that each type of user would see.

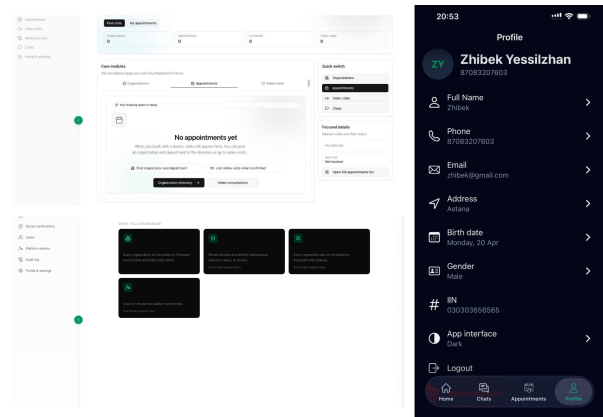


Fig. 2. Surfaces of Medlink.

The Medlink at the backend level was to be a monolithic application that is modular in Go and provided with both RESTful APIs and WebSocket-based interfaces. This design enabled the system to implement the normal request response that could be used in administering an account, scheduling, as well as updating records and to have real time features like chat and signaling to support a video consultation. The primary persistent store was PostgreSQL where we stored structured information such as user accounts, profile information on doctors and patients, appointment information, data on chat, and audit history. In the realm of this project, this pairing of a monolithic back end and relational database was suitable since it gave a balance that was easy to manage, transaction consistency and future scalability. This architecture was not developed as a mere prototype interface, but as a competent solution, a represented computing base, that would be able to carry numerous workflows in healthcare within a single platform. This backend structure was complemented with the front end and client-side approach. The web interface was also developed using React/Next.js and TypeScript that facilitated

the interactive responses, component-oriented approach and allowed the safer frontend development with straw type discipline. Simultaneously, the project designed and partially deployed an iOS client with SwiftUI, through which the system would have accessed beyond the browser into mobile access as well. Collectively, these decisions are indicative of the intention of the project to ensure that Medlink can be applicable in various settings and remain connected to the same back end services. The final report must be that, this front-end and mobile approach was one of a comprehensive approach of accessibility and multi platform support as opposed to a set of non-linked implementation decisions. Security also was integrated into the project approach. The documents attached always explain how password hashing will be used, short lived JWT-based authentication, role based access control, audit trails and TLS/HTTPS as components of platform design. The importance of these mechanisms in particular is due to the fact that in a healthcare environment, such as, Medlink, privacy, accountability, and limited access to sensitive information are more of needs rather than features of design. The approach to the project in that respect was, in these ways, influenced by the expectations of the domain: authentication, authorization, encrypted communication and auditability were designed to be part of the core system architecture since the start to implement an environment that is sensitive to privacy and telemedicine. The real-life application of these requirements may be related to the user-facing workflow depicted further in Figure 3, with authentication and registration being the point of entry in the platform. Another key methodological consideration was a real-time communication. Project materials give API WebSockets as the foundation upon which the live messaging works, with WebRTC intended to be the foundation upon which low-latency peer-to-peer video consultation works. The backend was supposed to serve as a signaling layer to orchestrate communications between doctors and patients and enable Medlink to ascend above merely accessing records or merely booking an appointment. It is a key fact to the report, as it shows that the methodology used in the project took into consideration both the synchronous and asynchronous requirements of telemedicine: All methodologies, such as messaging, coordination of consultation, and future video interaction were given as one with the rest of the architecture. The project method was process and organizational based too. These reports demonstrate the constant application of GitHub to version control and branching, integration, and subsequent building, as well as Notice to plan work and record it, and Docker and/or Docker Compose to prepare development and deployment tests. This tooling was part of an Agile-like process of development based on iterative development,

frequent feedback on the current state and blockers, and the delivery of working functionality in small steps. This kind of process played a role in a five-person endeavor, as it aided in giving the team some way of coordinating individual deliveries, yet ensuring a technical direction overall. It also presented evidence, as the final-report rubric requires, that the project was actually done by an organized software-engineering process as opposed to ad hoc implementation. The Spring 2026 specifications provide further insight on how the project approach was related to tangible delivery objectives. Particularly, the implementation state of significant groups of requirements can be outlined in a requirements status matrix, that is differentiating between completed, partially completed and impending functionality. It is helpful to refer to this material at the beginning of the section since it indicates that the methodology itself turned out not only to be architectural, but also to be tracked down to the needs of the project and the anticipated results.

TABLE I
REQUIREMENTS STATUS MATRIX

Requirement Group	Main Requirement / Capability	Status	Evidence in System
Authentication and access control	User signup, login, logout, JWT session handling, guarded routes, role-aware access	Implemented	Secure registration/login, JWT-based authentication, protected route behavior, role-aware platform surfaces
Doctor verification and administration	Doctor verification workflow, admin review, organization/departement control, audit-oriented governance	Implemented	Admin verification flows, organization and department management, user/admin oversight present
Doctor profiles and scheduling	Doctor profile management, department assignments, availability slots, schedule management	Implemented	Doctor profile setup, department assignment, schedule view and slot management
Patient appointment management	Doctor discovery, booking, cancellation, status tracking, patient appointment history	Implemented	Doctor search, appointment booking, cancellation, and appointment list views
Real-time communication	Chat and video consultation support	Implemented	WebSocket-based real-time chat prototype implemented; video consultation infrastructure and signaling prepared
Electronic medical records	Longitudinal patient records, health profile, prescriptions, encounter documentation	Implemented, can be enhanced	Patient health data editing, prescription history, encounter/prescription views available; richer EMR still expanding
Notifications and reminders	In-app notifications, appointment reminders, status-change alerts	Implemented	Notification center and unread/read flows present; broader notification maturity still developing
Security and privacy	Password hashing, JWT, RBAC, encrypted transport, privacy-aware consultation design	Implemented	Authentication, role-based access, protected routes, auditability, and secure backend design
Performance and consistency	Low-latency messaging, concurrent request consistency, booking conflict prevention	Implemented	Real-time chat present; appointment uniqueness and backend constraints support consistency
Deployment and maintainability	Dockerized setup, modular monolith, backups/health checks, scalable structure	Partially implemented	Docker/Docker Compose, modular backend architecture, deployment-oriented foundation established

The distribution of team roles was based on a realistic approach to software-development process. Based on the project documentation, they allocated responsibility to the frontend development, backend development, real-time communication and scheduling logic, iOS development, and the project management/testing coordination. These roles may not be emitted in the form of nominal assignments in the final report; rather, they are to be related to the real system outputs that came out of them. To illustrate, the frontend/backend collaboration could be seen in the authentication process depicted in Figure

3, doctor onboarding and verification in Figure 4, and administrative control of healthcare organizations and departments in Figure 4. The way in which the figures are presented enhances the argument that the team was working very well not only by sharing the work, Figure 3 displays the user authentication and registration workflow providing an entry point where various users can enter the system. This number is crucial since it is the basis of the role-aware interaction within Medlink: only having gone through a secure authentication procedure, patients, doctors, and administrators will be able to use their respective functions.

Fig. 3. Authentication Interface.

The registration of doctors is given in Figure 4 where the doctor specific information is taken by the system and facilitates the workflow that ultimately results in administrative verification. This number contributes to illustrating that Medlink was implemented so that it can be not only accessed by unprofessional users, but also a regulated process of onboarding medical professionals.

Doctor	License	Status	Actions
Alperin Yakovlev	A2-40-08626	Pending	Approve, Reject
Alfharan Hossain	2206204	Pending	Approve, Reject

Fig. 4. Doctor Registration Workflow.

Figure 5 depicts the administrative assistance to organization/department. This number is of particular interest since it demonstrates that this means that Medlink adds governance and institutional-management features to the interaction between the patients and doctors. That is, the platform was modeled more realistically to represent actual healthcare structures and not as a telemedicine separate communication tool.

Organization	License	Status	Actions
Alperin Central Hospital	A2-40-08626	Pending	Approve, Reject
Alfharan Hossain	2206204	Pending	Approve, Reject

Fig. 5. Management of administrative organizations and departments.

6. PROJECT EXECUTION

As can be seen in the implementation of Medlink, the project was far past the conceptual planning stage and into a real implementation stage that involved working with a multi-role system behavior. The best indication is the first-semester final report, which reveals that at this point the team had already developed secure user registration and login, password hashing, JWT issue, administrator support to organizations and departments, doctor profile and schedule, patient-side doctor search, appointment booking and a prototype real-time chat. This indicates that the project was already an operational prototype with database-supported workflows with various user roles. The introduction of the core workflow between the patients and doctors that was based on the same system was one of the most significant accomplishments. Patients would be able to find physicians, see the open slots and request an appointment, whereas doctors would be able to maintain schedules and check appointments via their interface. Figure 6 demonstrates this workflow through the doctor side with the doctor dashboard where the doctor can see the appointment information, profile controls and schedule-management tools. The figure shows that the doctor interface already accommodated realistic workflow actions as opposed to being a mockup.

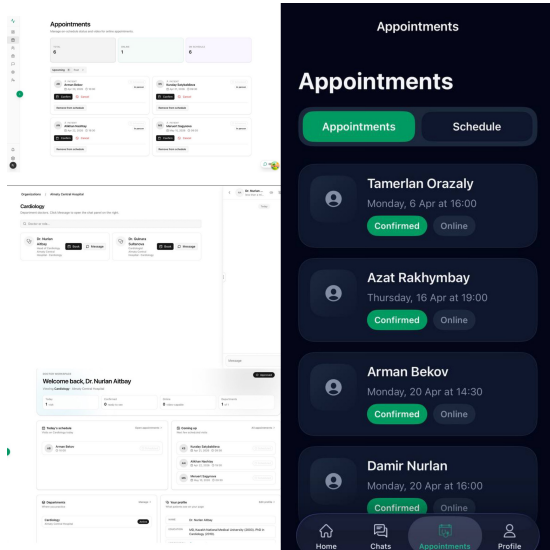


Fig. 6. Doctor dashboard showing appointments, profile controls, and schedule-management tools.

The implementation of Medlink demonstrates that the project has gone far beyond conceptual planning and has a significant implementation phase that includes working multi-role behaviour throughout the platform. At the prototype level, the system was already secured in terms of creating a registration and a log-in system, hashing of passwords, issuing of JWTs, administration of organizations and departments, doctor profile and schedule management, patient-side doctor search, booking appointments, and prototype real-time chat. Practically this implies that Medlink had ceased to be merely an architectural proposal; and was already a working telemedicine platform with integrated workflows, continuous data, and different role based product surfaces between patients, doctors and administrators. Figure 7 can be used to support this more inclusive implementation perspective and illustrate the end-to-end process of the system being implemented by discovery to consultation and help the reader see the interrelation between the two. Another significant strength of execution was the fact that the features implemented were not in isolation. At the patient level, the user would no longer have to browse organizations and departments but be able to find a doctor, choose an appointment, track the status of appointments, and get access to consultations. The prototype on the doctor side assisted in setup of profiles that were department-aware, schedule definition, appointments review and lifecycle actions based on actual patient requests. At the administrative level, the system already assisted in the management of organizations and the verification of doctors, this provided the system with an institutional and administration layer besides patient-doctor interaction. This linked presentation is noteworthy

in that it demonstrates that Medlink was already operating as a single product as opposed to a series of disjointed screens. This is precisely the continuity of workflow that needs to be reflected in Figure 7 where the process of patient discovery to doctor-side confirmation, chat and consultation entry could be depicted as a single process. The other significant outcome of the implementation was the introduction of structured rules in the back-end which imposed realistic platform behavior. The uniqueness constraints at the database level were used to prevent a case of double-booking of an appointment, video-call token generation was limited to the appointment status and online eligibility, and the chat relationships were limited in such a way that a pair of patients, doctors and departments was allowed to use one chat thread. Moreover, the change of statuses in the upper levels of administration might trickle down to departments, schedules, and subsequent appointments. This information reinforces the engineering worth of the project since they demonstrate that the backend was not simply storing data, but also imposing business rules and integrity of the platform. These controlled flows, and the visible patient/doctor process, should also be related in a practical manner (as supported by Figure 7), as that figure can bridge the gap between workflow displayed to the user and the logic implemented beneath it.

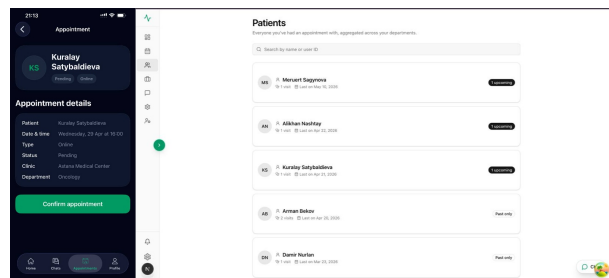


Fig. 7. Discovery to Consultation End-to-End Workflow.

The workflow on the patient side is depicted in Figure 8, where the interface helps to search a doctor and make an appointment. This number aids in illustrating that Medlink assisted an end-to-end interaction channel: patients could locate a doctor, select one of the slots available, and generate an appointment request which would subsequently show up on the doctor side of the system. This is significant since it demonstrates that the platform had already had regular interrelated workflows as opposed to individual features.

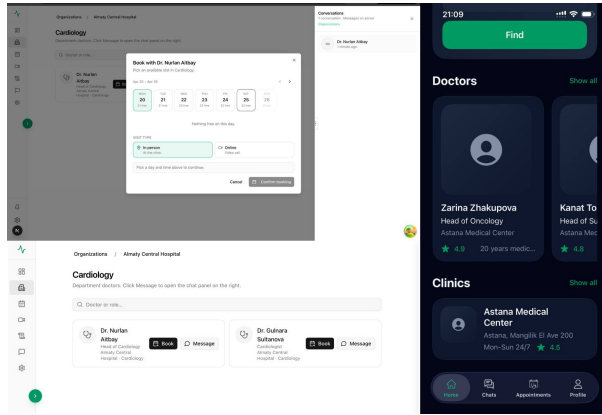


Fig. 8. Patient doctor-search and appointment-booking interface.

The administrative functionality was also a significant aspect of the execution of the project. The support of managing organizations and departments implemented by the team enabled the platform to be more representative of the actual healthcare organizations and departments. This level of administration also enhanced the trust and governance by establishing the framework within which the profiles of doctors as well as their affiliations could be managed and checked. Another advancement that was made in the project was with regard to real-time communication although this was only partially implemented. The documents reveal that a prototype chat component was already created based on WebSockets and it was possible to send and store messages. This is reflected in Figure 9 which depicts the prototype doctor patient text-chat interface. The number is helpful since it demonstrates that Medlink had already proved that it was possible to conduct live low-latency communication within the platform.

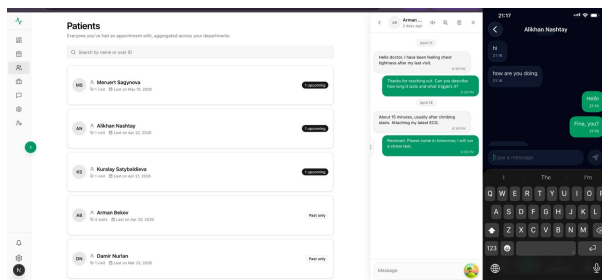


Fig. 9. Prototype doctor-patient real-time text chat implemented via WebSocket.

Besides the features that are already presented in the screenshots, the implemented system architecture allows a wider range of operation flows to be highlighted which could be further highlighted in the final report. These are patient profile CRUD, patient appointment CRUD, doctor department membership management, doctor schedule-slot CRUD, doctor appointment status updates,

platform-admin, organization-admin, department-admin, chat event delivery based on WebSockets and video-room token access based on appointment identity. Emphasizing such flows would enhance the execution section since it demonstrates that this project also involved more than just visible UI screens and involved more in-depth backend-supported processes needed to make up a real telemedicine platform. These larger operational relationships too can be connected back to Figure 7, as it can be used as the overview of a user workflow, as well as as a visual summary of how various actions supported by backends relate to each other into a single coherent care process.

Simultaneously, video consultation still was a more developed objective. The architecture, and the signaling strategy had already been determined, however, the full video experience based on WebRTC would be postponed to subsequent phases. This is a good thing to state in the report since it will differentiate between the features that have been implemented and the partially implemented and planned ones. A good project-execution part must also address the issues that were experienced when developing the project. The first-semester report notes various key challenges such as WebRTC/STUN/TURN integration complexity, the necessity to avoid appointment double-booking, the trade-off between the convenience of development and security-configured, scope creep, and merge conflicts due to the five-team members working concurrently. These problems are not to be concealed, as they demonstrate actual engineering work and making practical decisions. The way that the team dealt with these challenges reflects good software-engineering judgment. Video calling was not implemented until the supporting architecture had been stabilized, and scheduling logic was refined using database constraints, and backend validation, scope was managed by going back to core requirements, and collaboration was enhanced by making the branching more explicit and communication more effective. This renders the project execution section an important part to document what has been constructed, as well as providing testimony to the process of optimisation and problem solving. The new Spring 2026 plan reveals that following the first semester, the team shifted its focus to the most significant unfinished elements, such as video consultation, enhanced electronic medical records, notifications, testing, and deployment readiness. These would be better recapped in an independent table that would be shown right after this discussion.

TABLE II
SPRING 2026 COMPLETION PRIORITIES

Priority Area	Planned Work	Reason for Priority	Expected Outcome
Video consultation	Complete WebRTC-based online consultation flow and integrate final doctor-patient video room experience	Video consultation is a core telemedicine requirement and was only partially completed earlier	Fully usable online consultation workflow tied to eligible appointments
Electronic medical records expansion	Extend structured EMR support, improve encounter documentation, record organization, and prescription handling	Medical-record continuity is central to the platform's healthcare value	More complete patient history and clinically useful doctor-facing record tools
Notifications and reminders	Improve appointment reminders, confirmations, schedule-change alerts, and general in-app notification workflows	Users need timely awareness of booking and consultation events	Stronger continuity, reduced missed appointments, and better communication
Testing and validation	Conduct functional, security, performance, and usability evaluation with clearer evidence collection	Final report requires validation, not only implementation description	More credible evidence that the system solves the identified problem
Deployment readiness	Improve operational preparedness through monitoring, recovery planning, and deployment refinement	Academic prototype should move closer to stable deployable form	More maintainable and reliable platform foundation
Usability refinement	Improve interface clarity, workflow simplicity, and accessibility across patient, doctor, and admin roles	Multi-role platforms require low-friction use for all stakeholders	Better task completion, user confidence, and adoption potential

In general, the development of the Medlink demonstrates a project that started with an idea and continued to the operational prototype with operational user flows, persistent data, role-based interaction, and partially tested real-time capabilities. It further demonstrates that the team had a lesson in both technical and organizational challenges and applied the same to prioritize better in second semester.

A. Platform F-O-D

In terms of functionality, Medlink can be perceived as a complex of interrelated product spaces, which are based on a single platform foundation. The former is authentication, session and access control, it allows the signing up, signing in, signing out, persistent session management, token expiry, authenticated behavior and guest behavior of routes and landing logic based on roles after logging in. This functionality is the entry point to the rest of the system and makes sure that the patients, doctors and administrators are redirected to the appropriate workflows upon authentication. These role-specific pathways should be separate as a whole in Figure 2 because this figure can be used to explain how the concepts of login and guarded navigation can help connect the users to the area they are supposed to be in. The second one is the patient workflow surface. Patients have an opportunity to search the organizations, narrow down to departments, search doctors, make appointments, see the history of appointments, cancel appointments, join online appointments, see medical records, update their personal health information, and see a history of prescriptions. This renders the patient aspect of the platform to be more than a booking portal; a continuity-of-care surface, with discovery, consultation, follow-up, and record access services related under the same system. The best way to present these features that are face to the patient is by having them in a

figure that would be better illustrated in Figure 10 that would display the patient feature map along with organization browsing, doctor discovery, booking, records, prescriptions, consultations and chats. The workflow surface of the doctor is the third area. Physicians are able to establish their professional profile, create and manage departments, schedule the availability, review and update appointments, engage in consultations, view patient-related perspectives, record encounter notes and prescribe medications. Doctor-department assignments and parameters on appointment duration are also supported by the backend model, meaning that the workflow of doctors is not modeled as the type of generic account, but as a structured clinical role that is linked to the departments, schedules, and appointments. These functions on the doctor side can be depicted alongside the administrative side in Figure 11 which should emphasise on the doctor profile management, schedule management, appointments and related actions to do with prescription. The fourth space is the administrative console which aids in governing the platform and in organizational control. Administrators are able to manage organizations and departments, allocate or delete administrative roles, look into a doctor verification request, search users and get access to audit-log capabilities. Status-cascade is also a part of the backend, wherein, any change made at an organization or departmental level is cascaded to other entities that are dependent on it, such as schedules and future appointments. This renders the operation of the admin surface operationally significant, as it does not only influence the data management, but also the availability of the services and the compliance-focused control. These functions of governance ought to be reflected in Figure 11, as well, since the latter can show the features of the admin-side verification, organization, department, and audit-management in a single picture. The fifth is communication and consultation support that comprises of patient-doctor chat, real-time message delivery via WebSocket, access to online consultation, notification management and video sessions, provided by issuing LiveKit tokens. These characteristics are the key aspects of the telemedicine nature of the platform since the features enable Medlink to cater to both types of interaction asynchronous (messages and reminders) and synchronous (live consultation). These communication-related capabilities can be traced back to Figure 10 to chat and consult the patient on the patient-side and to communicate and handle consultations with the doctor on the doctor-side as shown in Figures 10 and 11 respectively.

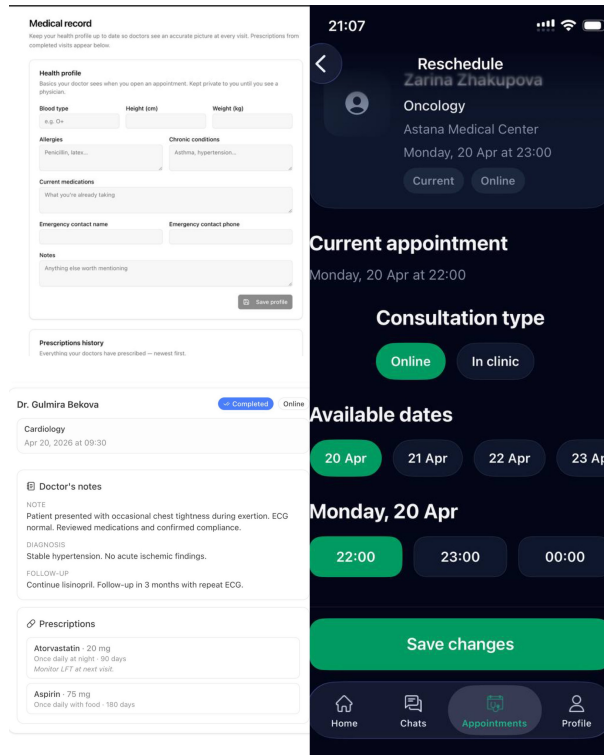


Fig. 10. Patient Feature Map.

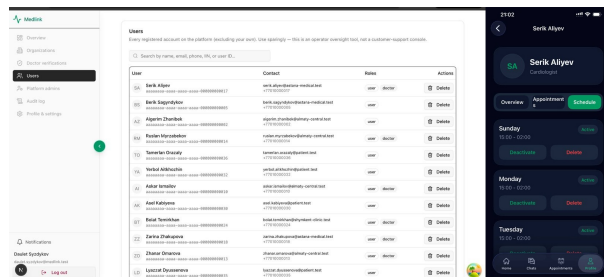


Fig. 11. Doctor/Admin Feature Map.

7. RESULTS AND EVALUATION

Medlink should be evaluated according to the official final-report rubric that will demand demonstrating that the identified problem was really solved by the system, and not just enumeration of usual tests of development. In the case of Medlink, that issue would be the necessity of a telemedicine solution that would integrate the access to patients, their effective workflow with doctors and provide safe management of medical information in a single system. Thus, the assessment should be based on the fact whether the implemented platform helps to realize these goals in practice and whether it can be evaluated as a valid computing-based solution. The project documents already give a good foundation to the evaluation. The kickoff document suggests functional

testing and performance testing and security testing, usability testing and pilot-user feedback, whereas the revised Spring 2026 requirements introduce more tangible targets like low-latency messaging, preventing against any conflicts between appointments, secure access control, and actual, useful usability among both patients and doctors. Combined, these resources could indicate the ultimate report will assess Medlink on four key dimensions, namely workflow correctness, communication performance, privacy and security and perceived usability by the user. A particularly valuable means of reinforcing the evaluation would be to map the four evaluation dimensions to feature groups of concrete features already in the system. The workflow correctness can be evaluated using authentication, doctor verification, organization browse, appointment booking, schedule management and appointment lifecycle transitions. The performance of communication can be evaluated in terms of chat delivery, WebSocket update behavior, and online appointment token access of video calls. JWT based authentication, protected-route behavior, role-aware access restrictions and verification workflows administered by an administrator can be used to measure privacy and security. The ease of the process of finding doctors and handling appointments, maintaining schedules and appointments, and reviewing verification and management tasks can be used to gauge usability. The mapping would help the evaluation section become more product-specific and more related to the implemented system as opposed to abstract criteria alone. This mapping scheme should be presented in Table 3 that will be able to group the features, illustrative workflows, evaluation dimensions, and types of evidence in a manner that will reveal the evaluation logic.

The first dimension is the workflow correctness which deals with the issue of whether the Medlink promotes the central interactions as anticipated by the users of the product. Evidence of the most important workflows, including registration and login, registering and verifying doctors, administration in an organization, managing doctor schedules, searching doctors on a patient side, booking a patient appointment and synchronizing patient and doctor interfaces, is already provided in the attached reports. Provided the final system will also have the extended medical-record support, one will be able to demonstrate it with the help of Figure 12 that, hopefully, will demonstrate the longitudinal EMR with uploaded documents, diagnosis notes, and prescriptions. This number would be used to show how, instead of only assisting in communication and booking, Medlink, as well, assists in a structured management of clinical information.

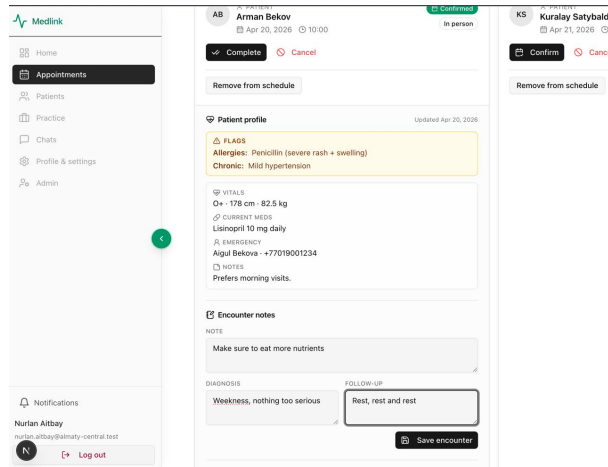


Fig. 12. EMR Record View.

The second dimension is the performance of communication that is critical in particular due to the fact that Medlink is developed as the interaction platform in real time along with the information system. The new requirements are to have chat delivering sub-second content at normal load and that concurrent appointment and messaging requests must be response with no inconsistency. This element of the assessment ought to be backed by benchmark summaries or uncomplicated performance measurements ideally. In case video consultation has been done, it is also possible to depict the final WebRTC consultation interface with connection status, consent notice and call controls with the help of Figure 13.

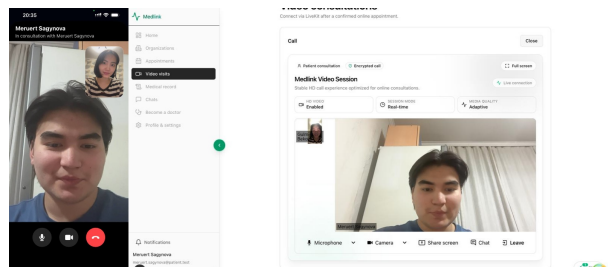


Fig. 13. Video Consultation Interface.

TABLE III
FEATURE-TO-EVALUATION MAPPING

Feature Group	Example Workflow	Evaluation Dimension	Evidence Type
Authentication and RBAC	User logs in and is redirected to the correct protected route based on role/capability	Security / Usability	Route-guard checks, successful login flow, role-based landing behavior
Doctor verification	Admin reviews and approves/rejects a doctor verification request	Governance / Security	Permission checks, workflow completion, admin action validation
Organization and doctor discovery	Patient browses organization → department → doctor	Workflow correctness / Usability	Task completion time, successful navigation, user feedback
Appointment booking	Patient selects a doctor, slot, and consultation mode, then submits booking	Workflow correctness	Successful booking cases, failed conflict cases, booking-state updates
Appointment lifecycle	Doctor confirms, starts, completes, cancels, or marks no-show	Workflow correctness	Lifecycle state transitions, status validation, UI/backend consistency
Real-time chat	Patient or doctor sends message, message appears in conversation, failed message can retry	Communication performance / Usability	Message delivery latency, retry behavior, chat success rate
Video consultation	Eligible user opens consultation and receives valid video-call access	Workflow correctness / Performance	Token issuance, call-entry success, time-window validation
EMR and prescriptions	Doctor records encounter and prescription; patient views updated record history	Workflow correctness / Usability	Record creation success, correct visibility by role, task completion
Notifications	User receives alerts and marks one or all as read	Usability / Workflow correctness	Notification visibility, unread/read state behavior, user interaction logs
Admin management and audit	Admin manages organizations/departments/users and reviews logs	Governance / Security	Admin task completion, capability enforcement, audit-log visibility

Privacy and security evaluation is assessed on the third dimension. Since Medlink requires handling of healthcare data, the aspect of security measures ought to be regarded as the main aspects of evaluation, as opposed to the peripheral details of regulatory implementation. The materials attached are a consistent description of password hashing, authentication using JWT, role-based access control, encrypted communication, audit logging, and privacy-aware approach to consultations. This is a dimension that must be represented in the final report either in terms of unauthorized access attempts being blocked, the functionality of doctor verification and administrative controls, and user-restrictive access to records. Although the full penetration test is not conducted, the report still can be helpful to provide the evidence based on the protected-route checks, role-based scenarios, and achieving the evidence of access-control requirements. The fourth dimension is usability as perceived by the users where it is imperative to have a telemedicine system that is aimed at patients as well as doctors. The conclusion report must reveal whether the users could handle essential tasks like registration, appointment booking, schedule viewing, messaging and accessing records without significant confusion and challenge. In case of implementations of notifications, it is possible to demonstrate how appointment reminders, confirmations and changes of schedule were reported in the system using Figure 14.

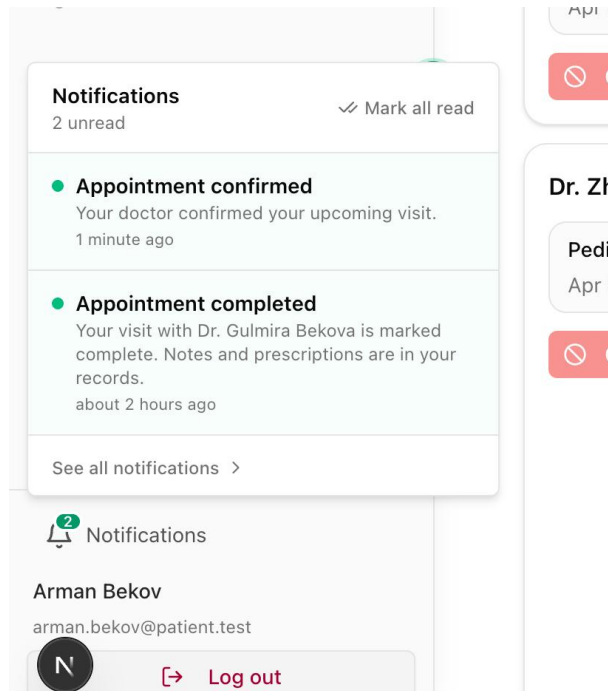


Fig. 14. Notification Center.

Meanwhile, the existing attachments for the current attachments are not yet in a position to furnish all the empirics required to give a complete documented assessment. The uploaded files lack final questionnaire tables and benchmark charts, sample sizes, detailed pilot-user descriptions, as well as a final requirement-coverage matrix indicating what Spring 2026 targets were met. That is why, final report must not overestimate the results, which are not directly supported by data. It is proper to introduce the evaluation scheme and explain what evidence must be contained but better academically not to trace that all targets were fulfilled at the end other supporting results must be inserted. Based on this, as many concrete artifacts as possible would be added to the final version of this section. These can be in the form of a requirement-traceability table, an overview of the appointment-conflict tests, chat-latency or backend-response metrics, usability questionnaire data, and some sample user comments. In case pilot users have been used then the report should document who they were and what they did, and how their observations prove that Medlink is addressing the initial healthcare-access issue in a significantly meaningful way.

8. DISCUSSION

Medlink is an effort at approaching telemedicine as an issue of systems-engineering as opposed to a feature by itself. This is its greatest strength; it is not only dedicated to video calls or messaging, but rather a unification of

authentication, physician verification, scheduling, communication, medical-record management, and administrative supervision is all implemented in a single system. This is crucial since successful telemedicine is contingent on more than remote contact, but trust, continuity of care, accountability, and secure management of clinical data. One of the project results is that it developed out of the conceptual design to a working prototype with multi-role workflows. Registration, doctor management, appointment booking, scheduling and real-time chat are the main features demonstrating that the system passed the requirements and architecture boundaries to the implemented software. This is important as it shows that the team could work out the problem analysis into a linked technical elements with a common back-end and persistent database. Meanwhile, the project illuminates the significant difference between implemented and planned as well as verified functionality. There were some functions that were obviously functional, and others that were half baked or needed more extensive testing and assessment. The acknowledgment of this difference makes the report scholarly, as in software engineering just the fact that something is a feature does not necessarily imply that the feature is complete and robust and that it is well tested in the context of use. There are also realistic trade-offs in engineering in the project. The development of a telemedicine platform meant achieving a balance between the scope of features and the scope of the possible, particularly regarding video consultation, consistency of operation, secure setting, and multi-platform compatibility. These issues are the engineering reality of healthcare systems and their open discussion makes them more credible since they admitted that the difference between a solid prototype and a platform ready to be used in production existed. The last aspect is that it is not the value of one single feature that makes Medlink important, but the interactions between its features. A telemedicine platform can only be practical when there is a coherent and stable communication, scheduling, record management, and access control work. That is why in case of this project, it must be considered not only with regard to whether separate components were introduced or not, but also with regard to whether they contribute to reliable and comprehensible patient, doctor, and administration workflows. It is one of the strengths of the project in terms of systems-level point of view. In general, Medlink can be considered as a powerful prototype level telemedicine system that has an apparent engineering value. Its greatest contribution is the fact that several workflows related to healthcare are unified into a single computing-powered solution. The project also illustrates significant advances on the way to a safe, role-conscious, and feasible telemedicine environment and, at the same time, displays the fact that only the subsequent

confirmation of performance, usability, and security will allow entirely trusting the final solution.

9. CONCLUSION AND FUTURE WORK

A. Conclusion

Medlink is a single telemedicine platform that has been created to enhance remote healthcare communication using a safe, structured, and role-aware digital environment. The project stood out as having an essential gap in telemedicine, where both access of the patient and access to the physician were integrated with appointment scheduling, medical-record management, administrative control, and secure communication in a single computing-based system. Medlink was not developed as a single, isolated component of the broader workflow, like messaging or video consultation but rather as a unified platform that can facilitate the requirements of remote care delivery. One of the key results of the project is that Medlink has moved out of a basic idea to a prototype that has a real multi-role functionality. The system implemented has secure registration and authentication, role-conscious access control, doctor verification workflows, organization and department management, doctor profile and schedule management, patient-side organization and doctor discovery, appointment booking and lifecycle management, patient records and prescription-related views, real-time chat, notification support, and video-consultation infrastructure. These features, combined, demonstrate that Medlink became a multi-purpose telemedicine platform where patient care processes, physician practice processes, and administrative governance can be all combined into a single computing solution. These elements indicate that the project has indeed transitioned out of planning and architecture into real implementation with inter-related workflows based on a common backend and persistent database. The project also showed the real world implementation of software engineering concepts to a healthcare issue. The layered architecture, role based access model, and ability to support both asynchronous and synchronous interaction, Medlink demonstrates how a computing-based solution can be developed to not only meet technical requirements, but also privacy, accountability, and usability issues in a healthcare environment. In this regard, the primary project contribution is not the development of the specific features as such, but the creation of a consistent platform where the latter features interact and contribute to the reliable telemedicine processes. Simultaneously, the project also demonstrated the distinction between the functionality implemented, its extensions, and the outcomes that were well-tested. Although the prototype is a significant step towards improvement and a solid system base, the ultimate strength of the solution will rely on the additional testing of the performance, usability, and

safety under the realistic conditions. Despite this, it can be concluded that Medlink is a powerful and valuable telemedicine project that supports further improvement and further development of the project in the form of deployment.

B. Future Work

A number of critical directions to future work which would improve Medlink and make it a more comprehensive telemedicine platform are available. The further development of the electronic medical record component is one of the main areas. More sophisticated support of structured EMRs, enhanced management of uploaded medical records, enhanced diagnostic and prescription processes, and more refined doctor-facing interfaces to manage patient records may be improved in the future. The other significant field is enhancement of communication and notification functionalities. Additions to future releases of Medlink may involve additional video consultation functionality, enhanced chat, application notifications, email alerts, and more trustworthy mechanisms of appointment confirmations, schedule adjustments, and follow-up. These would enhance user interaction and make the site more useful to the daily medical practice. The future work is also a direction that is important in terms of deployment maturity. Though the project has already created a solid architectural and implementation base, one way in which subsequent development might be directed is by work on production preparedness by enhancing monitoring, scaling test, recovery process, and backup measures. These would be critical in case Medlink shifted off of an academic prototype to actual deployment. The security can be further extended in the future. Possible enhancements are multi-factor authentication, better session management, better consent controls, and superior audit and monitoring systems. As Medlink is sensitive healthcare data, these additions would enhance trust, increase privacy protection, and augment ability to support the alignment to real-world healthcare data needs. A more long-term strategy of future work is the interoperability with external healthcare systems. The possibility of integrating with hospital systems, standardized healthcare APIs, or other medical information services could make Medlink much more useful practically and enable it to operate as a component of a larger digital-health ecosystem, instead of a single platform. Lastly, the usability will always be required to be refined. As telemedicine systems have to provide service to users with varying degrees of digital literacy, future work should continue enhancing interface clarity, navigation, accessibility, and overall user confidence. Further usability testing with patients and doctors would aid in finding out the remaining usability problems and further reinforce the real-world

applicability of the platform. In general, Medlink offers a good platform of a safe and combined telemedicine system. As it continues to grow, be tested, and improved, it can become a viable academic prototype or become more mature and useful in actual practice as a healthcare platform.

REFERENCES

- [1] A. Haleem, M. Javaid, R. P. Singh, and R. Suman, "Telemedicine for healthcare: Capabilities, features, barriers, and applications," *Sensors International*, vol. 2, art. no. 100117, 2021.
- [2] A. L. Sønderlund et al., "Development of a video consultation patient-satisfaction measure for general practice: Instrument development study," *JMIR Formative Research*, vol. 8, e58928, 2024.
- [3] E. Carini, F. Villani, A. Pezzullo, G. Fiorillo, and F. Graffigna, "The impact of digital patient portals on health outcomes, system efficiency, and patient attitudes: Updated systematic literature review," *Journal of Medical Internet Research*, vol. 23, no. 9, e26189, 2021.
- [4] European Union[1], "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, 2016.
- [5] H. Mahmoud and R. Abozariba, "A systematic review on WebRTC for potential applications and challenges beyond audio-video streaming," *Multimedia Tools and Applications*, vol. 84, pp. 2909–2946, 2025.
- [6] L. L. Beal et al., "Quantifying patient portal use: Systematic review of electronically generated data," *Journal of Medical Internet Research*, vol. 23, no. 2, e23493, 2021.
- [7] M. Ansarian and Z. Baharlouei, "Applications and challenges of telemedicine: Privacy-preservation as a case study," *Archives of Iranian Medicine*, vol. 26, no. 11, pp. 654–661, 2023.
- [8] S. Simola et al., "Patients' experiences of a national patient portal and its usability: Cross-sectional survey study," *Journal of Medical Internet Research*, vol. 25, e45974, 2023.
- [9] T. Suleski, M. Ahmed, W. Yang, and E. Wang, "A review of multi-factor authentication in the Internet of Healthcare Things," *Digital Health*, vol. 9, 2023.
- [10] U.S. Department of Health and Human Services, "Summary of the HIPAA Security Rule," 2024.