

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Амангелді Ернар Нұржанұлы
Жансерік Мирас Қанатұлы

ТҮСІНДІРМЕ ЖАЗБА
дипломдық жобаға

6B06102 – Computer Science білім беру бағдарламасы

Алматы 2025

**ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ**

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедрасының меңгерушісі

техн. ғыл. канд., қауымдастырылған
профессор

_____ Ф.Н. Абдолдина
« ____ » _____ 2025 ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: Медициналық деректерді бақылау және инвазивті емес құралды
басқару үшін Web қосымша жасау (backend / front-end)

6B06102 – Computer Science білім беру бағдарламасы

Орындады

Амангелді Е.Н

Жансерік М.Қ

Рецензент

Ғылыми жетекшісі

техн. ғыл. канд.

техн. ғыл. канд., қауымдастырылған
профессор

_____ Сейлова Н.А.

_____ Бейсембекова Р.Н.

« ____ » _____ 2025 ж.

« ____ » _____ 2025 ж.

Алматы 2025

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН

ПИ кафедрасының меңгерушісі
техн. ғыл. канд., қауымдастырылған
профессор

_____ Ф.Н. Абдолдина
« ____ » _____ 2025 ж.

Дипломдық жобаны орындауға
ТАПСЫРМА

Білім алушыларға Жансерік Мирас Қанатұлы

Амангелді Ернар Нұржанұлы

Тақырыбы: *Медициналық деректерді бақылау және инвазивті емес құралды басқару үшін Web қосымша жасау (backend / front-end).*

Академиялық мәселелер жөніндегі проректор бұйрығының № _____ « ____ »
_____ 202_ ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі: « ____ » _____ 2025 ж.

Дипломдық жобаның бастапқы мәліметтері:

- а) пәндік саланы талдау, ұқсас жобаларды талдау;
- б) техникалық тапсырманы әзірлеу;
- в) бағдарламаның архитектурасын әзірлеу;
- г) түсінікті және интерактивті дизайнды жобалау және әзірлеу;
- д) бағдарламалық жасақтаманы әзірлеу және тестілеу;

Дипломдық жобада әзірленуге жататын сұрақтардың тізбесі: (міндетті
сызбаларды дәл көрсете отырып): презентацияның _____ слайдтары ұсынылды.

Ұсынылатын негізгі әдебиеттер: _____ пайдаланылған әдебиеттер тізімінен.

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жоба тақырыбына байланысты әдебиеттерді іздеу және талдау	01.12.2024	Орындалды
2. Зерттеліп отырған тақырыптың жалпы сипаттамасы	10.12.2024	Орындалды
3. Әзірленген бағдарламалау ортасын таңдау	02.02.2025	Орындалды
4. Дипломдық жобаның веб-сайт қосымшасын дайындау және іске асыру	29.03.2025	Орындалды
5. Қосымша материалдар мен презентацияны дайындау	29.03.2025	Орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы	Баға
Бағдарламалық бөлім	Еспаев Ғ.Б., тех.ғыл.магистрі, оқытушы			
Нормалық бақылаушы	Нұралықызы С., тех.ғыл.магистрі, аға оқытушы			

Ғылыми жетекші _____ Бейсембекова Р.Н.

Тапсырманы орындауға қабылдап алған студент _____ Амангелді Е.Н

Тапсырманы орындауға қабылдап алған студент _____ Жансерік М.Қ

Күні _____ «__» _____ 2025 ж.

АҢДАТПА

Бұл дипломдық жобамыз қант диабетімен ауыратын науқастарға арналған медициналық ақпараттық жүйені әзірлеуге бағытталған. Жобаның негізгі мақсаты – науқастардың денсаулық көрсеткіштерін тиімді бақылауға, дәрігерлермен өзара байланысты оңтайландыруға және жалпы емдеу сапасын арттыруға арналған цифрлық шешім ұсыну.

Жүйе Java бағдарламалау тілінде Spring Framework негізінде әзірленді, бұл шешімнің тұрақтылығын, модульдік құрылымын және кеңейтілу мүмкіндігін қамтамасыз етеді. Backend бөлігі RESTful API архитектурасы негізінде жасалған және мәліметтер PostgreSQL дерекқорында сақталады. Ал клиенттік (frontend) бөлігі React.js кітапханасының көмегімен әзірленіп, заманауи әрі мобильді құрылғыларға бейімделген интуитивті интерфейс ұсынады.

Жүйеде келесі негізгі функцияларды қарастырсақ:

- Науқастың қандағы қант деңгейін күнделікті тіркеу;
- Инсулин дозасын енгізу;
- Науқастың денсаулық көрсеткіштерін бақылап отыру;
- Дәрігер мен науқас арасындағы кері байланысты қамтамасыз ету.

Қолданушылар жүйе арқылы күнделікті глюкоза деңгейін, инсулин дозасын тіркеп отыра алады. Дәрігер мен науқас арасындағы байланысты жақсартып, уақтылы емдеу шараларын қабылдауға және аурудың асқыну қаупін төмендетуге мүмкіндік береді.

Аталған жүйе – медициналық технология мен бағдарламалық жасақтаманың тоғысында әзірленген интеллектуалды шешім болып табылады. Ол науқастың жағдайын үздіксіз бақылауға және жедел әрекет етуге жағдай жасайды. Сонымен қатар, бұл жоба Қазақстандағы денсаулық сақтау саласын цифрландыруға өзіндік үлес қосатын маңызды платформа бола алады.

АННОТАЦИЯ

Целью данного дипломного проекта является разработка медицинской информационной системы для пациентов с сахарным диабетом. Основная цель проекта - предоставить цифровое решение для эффективного мониторинга показателей здоровья пациентов, оптимизации взаимодействия с врачами и повышения общего качества лечения.

Система разработана на языке программирования Java на базе платформы Spring Framework, что обеспечивает стабильность, модульность и расширяемость решения. Серверная часть основана на архитектуре RESTful API, а данные хранятся в базе данных PostgreSQL. Клиентская (фронтенд) часть разработана с использованием библиотеки React. Она использует JS и предлагает интуитивно понятный интерфейс, современный и адаптированный для мобильных устройств.

Рассмотрим основные функции системы:

- Ежедневный мониторинг уровня сахара в крови пациента;
- Введение дозы инсулина;
- Мониторинг показателей здоровья пациента;
- Обеспечение обратной связи между врачом и пациентом.

С помощью системы пользователи могут отслеживать ежедневный уровень глюкозы и дозы инсулина. Это улучшает коммуникацию между врачом и пациентом, позволяя своевременно назначать лечение и снижая риск осложнений при заболеваниях.

Эта система представляет собой интеллектуальное решение, разработанное на стыке медицинских технологий и программного обеспечения. Она позволяет постоянно контролировать состояние пациента и оперативно принимать меры. Кроме того, этот проект может стать важной платформой, которая будет способствовать цифровизации сферы здравоохранения в Казахстане.

ANNOTATION

The goal of this diploma project is to develop a medical information system for patients with diabetes. The main goal of the project is to provide a digital solution for effective monitoring of patients' health indicators, optimization of communication with doctors and improvement of the overall quality of treatment.

The system is developed in the Java programming language based on the Spring Framework, which ensures stability, modularity and scalability of the solution. The backend is built on the RESTful API architecture, and the data is stored in the PostgreSQL database. The client (frontend) part is developed using the React.js library and offers an intuitive interface, modern and adapted for mobile devices.

Let's consider the following main functions in the system:

- Daily recording of the patient's blood sugar level;
- Administration of insulin dose;
- Monitoring the patient's health indicators;
- Provide feedback between the doctor and the patient.

With the help of the system, users can track daily glucose levels and insulin doses. This improves communication between the doctor and the patient, allowing for timely treatment and reducing the risk of complications of diseases. This system is an intelligent solution developed at the intersection of medical technologies and software. It allows for continuous monitoring of the patient's condition and prompt action. In addition, this project can become an important platform that will contribute to the digitalization of Kazakhstan's healthcare sector.

МАЗМҰНЫ

	Кіріспе	9
1	Негізгі бөлім	10
1.1	Жобаның өзектілігі	10
1.2	Жобаның мақсаты	11
1.3	Терминдер мен қысқартулар	12
1.4	Жүйені модельдеу	14
1.5.1	ER диаграммасы	15
1.5.2	BPMN диаграммасы (дәрігер)	17
1.5.3	BPMN диаграммасы (науқас)	19
1.6	UML диаграммасы	21
1.6.1	MVVM	21
1.7	Технологиялар бөлімі	23
1.7.1	Java бағдарламау тілі	24
1.7.2	Spring фреймворк	24
1.7.3	React JS кітапханасы	25
1.7.4	Docker контейнеризация	25
1.7.5	Git	26
1.7.6	GitLab	27
2	Жобаның құрылымы	28
2.1	Регистрация және авторизация бөлімі	29
2.2	Қолданушының басты бет көрінісі	31
2.3	Деректер беті	32
2.4	Дәрі қабылдау беті	34
2.5	Глюкоза өлшемі	35
3	Қолданушы интерфейсінің жобасы	38
3.1	Интерфейс элементтерінің жобасы	42
3.2	Қосымша беттер	43
3.3	Чат	45
3.4	Администратор панелі	46
4	Бағдарламаны жасау және тестілеу	48
4.1	Бағдарламалық жасақтама компоненттері	49
4.2	Тестілеу	51
	Қорытынды	52
	Пайдаланылған әдебиеттер тізімі	53
	Қосымша А: Техникалық тапсырма	55
	Қосымша Ә: Пайдаланушы нұсқаулығы	58
	Қосымша Б: Бағдарлама коды	60

КІРІСПЕ

Қазақстан Республикасының денсаулық сақтау саласы қазіргі кезеңде цифрлық технологияларды кеңінен енгізу арқылы халыққа көрсетілетін медициналық қызметтердің сапасын арттыруды мақсат етіп отыр. Әсіресе, созылмалы аурулармен, соның ішінде қант диабетімен күрес – ұлттық деңгейде өзекті мәселелердің бірі болып табылады. Қант диабеті науқастың өмір сүру сапасына, еңбекке қабілеттілігіне және жалпы денсаулығына тікелей әсер ететін аурулардың қатарында. Бұл дертті тиімді басқару үшін медициналық қызметтердің уақтылы көрсетілуі, науқастың жағдайын үздіксіз бақылау және жеке емдеу тәсілдерін қолдану қажет.

Осыған байланысты заманауи ақпараттық технологиялар негізінде әзірленген медициналық жүйелердің рөлі ерекше. Цифрлық шешімдер науқас пен дәрігер арасындағы өзара әрекеттестікті жетілдіруге, деректерді тиімді басқаруға және емдеу процесін оңтайландыруға мүмкіндік береді. Дипломдық жоба – қант диабетімен ауыратын науқастарға арналған медициналық ақпараттық жүйені әзірлеуге бағытталған. Жүйенің мақсаты – науқастың денсаулық көрсеткіштерін автоматтандырылған түрде бақылау, жинақталған мәліметтер негізінде жеке ұсыныстар беру және медициналық көмектің сапасын арттыру.

Жүйе Java бағдарламалау тілі және Spring Framework негізінде әзірленген. Backend бөлігі RESTful API архитектурасын қолдану арқылы жүзеге асырылып, барлық деректер PostgreSQL мәліметтер қорында сақталады. Ал клиенттік бөлігі React.js кітапханасы арқылы құрылып, қолданушыға ыңғайлы және заманауи интерфейс ұсынады. Жүйеде қандағы глюкоза деңгейін тіркеу, инсулин дозасын жазу, сондай-ақ дәрігер мен науқас арасындағы кері байланысты қамтамасыз ету мүмкіндіктері қарастырылған.

Медициналық ақпараттық жүйелерді пайдалану арқылы қант диабетімен күресті тиімді ету, емдеу сапасын арттыру және науқастың өмір сапасын жақсарту. Сонымен қатар, жүйе Қазақстанның денсаулық сақтау саласын цифрландыруға нақты үлес қоса отырып, медициналық қызмет көрсетуді жаңа деңгейге көтеруге мүмкіндік береді.

1 Негізгі бөлім

1.1 Жобаның өзектілігі

Қазіргі цифрлық дәуірде медициналық ақпараттық жүйелерді пайдалану денсаулық сақтау саласының ажырамас бөлігіне айналуда. Сандық технологиялар науқастың жағдайын үздіксіз бақылауға, медициналық деректерді тиімді басқаруға және емдеу процесін жекелендіруге жаңа мүмкіндіктер ашады. Атап айтқанда, созылмалы аурулардың бірі – қант диабетін тиімді басқару мәселесі қазіргі қоғамда аса өзекті болып отыр. Бұл дерт әлем бойынша миллиондаған адамның өміріне әсер етіп, әлеуметтік және экономикалық тұрғыда да күрделі салдарлар туғызуда.

Қант диабеті – қандағы глюкоза деңгейінің бұзылуымен сипатталатын созылмалы ауру. Оны дұрыс бақылап отырмаса, жүрек-қан тамырлары аурулары, бүйрек жеткіліксіздігі, көру қабілетінің нашарлауы сияқты түрлі асқынуларға әкелуі мүмкін. Мұндай асқынулардың алдын алу үшін науқастың күнделікті денсаулығын үнемі бақылап, дер кезінде қажетті шараларды қабылдау маңызды. Бұл процесті тиімді ету үшін медициналық ақпараттық жүйелерді қолданудың маңызы зор.

«Медициналық деректерді бақылау» ақпараттық жүйесін әзірлеу осы қажеттіліктерді қанағаттандыру мақсатында қолға алынды. Бұл жүйе қант диабетімен ауыратын науқастардың қан құрамындағы қант деңгейін және инсулин қабылдауын тіркеп, осы мәліметтерді талдау арқылы жеке ұсыныстар ұсынуға мүмкіндік береді. Сонымен қатар, жүйе медициналық қызметкерлерге әрбір науқас туралы нақты әрі құрылымдалған мәлімет ұсына отырып, диагноз қою және ем жүргізу процесін оңтайландырады.

Жобаның өзектілігі, ең алдымен, Қазақстанда да қант диабетімен ауыратындар санының артуымен, сондай-ақ ауруды ерте кезеңде анықтап, оны басқарудың тиімді құралдарының жетіспеушілігімен байланысты. Медициналық мекемелерде қолданылатын көптеген жүйелер науқаспен тікелей кері байланыс орнату және нақты уақыттағы деректерге негізделген ұсыныстар беру мүмкіндігін толық қамтамасыз ете алмайды. Ал бұл жоба науқас пен дәрігер арасындағы байланысты күшейтіп, медициналық көмекті жедел әрі дәл көрсетуге жол ашады.

Сонымен қатар, жобада заманауи технологиялардың (Java, Spring Boot, React, PostgreSQL) қолданылуы – жүйенің қауіпсіздігі мен тұрақтылығын қамтамасыз етіп қана қоймай, оны болашақта кеңейтуге және басқа медициналық қызмет түрлерімен интеграциялауға жол ашады. Бұл өз кезегінде Қазақстанның денсаулық сақтау саласын цифрландыру бағытындағы стратегиялық мақсаттарымен үндес келеді.

Осы жобаны жүзеге асыру арқылы біз қант диабетімен ауыратын адамдардың өмір сапасын арттыруға, медициналық көмектің қолжетімділігі мен тиімділігін жақсартуға, сондай-ақ денсаулық сақтау жүйесіндегі ресурстарды оңтайлы пайдалануға мүмкіндік аламыз. Сондықтан,

«Медициналық деректерді бақылау» жобасының өзектілігі бүгінгі күні ерекше маңызға ие.

1.2 Жобаның мақсаты

Жоба қант диабетімен ауыратын науқастардың медициналық деректерін бақылауға арналған жүйені әзірлеуге бағытталған. Бұл жүйенің басты мақсаты – науқастардың медициналық жағдайын тиімді бақылау, оларды дер кезінде емдеу және диагностиканың дәлдігін арттыру. Медициналық деректерді басқаруға арналған инвазивті емес құралдармен жұмыс істейтін веб қосымшаны жасау. Қосымша, дәрігерлерге науқастардың жағдайын қадағалауға, ал науқастарға өз дәрі-дәрмектерін уақытында қабылдау туралы ескертулер жіберуге мүмкіндік береді.

Қосымшаның міндеті – қант диабетімен ауыратын науқастардың жағдайын бақылау процесін жеңілдету, медициналық ақпаратты автоматты түрде жинақтау және оны тиімді түрде өңдеу. Бұл жүйе дәрігерлерге науқастың денсаулық жағдайын жылдам әрі дәл бағалауға мүмкіндік береді, ал науқастарға дәрі-дәрмек қабылдау уақытын ұмытпауға көмектеседі.

Жобаның негізгі мақсаттары мен міндеттері:

1. Қант диабетімен ауыратын науқастардың медициналық көрсеткіш деректерін бақылауға арналған жүйе жасау: Қант диабетімен ауыратын науқастардың денсаулық жағдайын бақылап, медициналық деректерді жинақтау және талдау арқылы емдеуді тиімді ету.

2. Науқастың электрокардиосигналын тіркеп, оны өңдеп көрсете алатын инвазивті емес құралды ұсыну: Қосымша науқастардың қант деңгейін, жүрек-қан тамырлары жүйесін және басқа да денсаулық көрсеткіштерін бақылауға арналған инвазивті емес құралдарды басқару.

3. Дәрігерлерге науқастардың денсаулығын қадағалауға мүмкіндік беретін интерфейс жасау: Дәрігерлерге науқастардың медициналық деректерін қадағалап, қажетті шешімдер қабылдауға мүмкіндік беретін визуализация және талдау құралдарын ұсыну.

4. Науқастарға дәрі-дәрмектерді қабылдау туралы уақытылы ескертулер жолдау жүйесін енгізу: Науқастарға дәрі-дәрмек қабылдау уақытын еске салатын автоматты хабарламалар жіберу жүйесін енгізу арқылы емдеу процесін оңтайландыру.

5. Диагностика мен емдеудің тиімділігін арттыру: Қосымша арқылы науқастардың медициналық деректерін уақытылы және дәл жинақтау арқылы диагностиканың дәлдігін арттыру, емдеу процестерінің тиімділігін қамтамасыз ету.

Жобаның ерекшеліктері:

- **Интерактивті интерфейс:** Қосымша пайдаланушыларға ыңғайлы интерфейс ұсынады, ол арқылы медициналық деректерді оңай енгізіп, бақылап отыруға болады.
- **Қашықтан мониторинг:** Қосымша дәрігерлерге науқастарды қашықтан бақылауға және уақытылы әрекет етуге мүмкіндік береді.
- **Жоғары қауіпсіздік:** Қосымша барлық медициналық деректердің қауіпсіздігін қамтамасыз ету үшін заманауи шифрлау және қауіпсіздік жүйелерін қолданады.
- **Пайдаланушыға ыңғайлы:** Қосымша жеке қажеттіліктерді ескере отырып, пайдаланушыларға деректерді енгізу, бақылау және талдау процесін жеңілдетеді.

Жобаның болашағы:

Жоба тек медициналық деректерді бақылауға арналған қосымша ғана емес, сонымен қатар дәрі-дәрмектердің уақытында қабылдануын қадағалау арқылы емдеуді тиімді ететін құрал ретінде де қолданылатын болады. Қосымша түрлі медициналық мекемелерде енгізілсе, қант диабетімен ауыратын науқастардың жағдайы қашықтан бақыланып, емдеу процессінің тиімділігі артады.

1.3 Терминдер мен қысқартулар

1 кесте – Анықтамалар, терминдер және қысқартулар

Терминдер және қысқартулар	Анықтамалары
API	Application Programming Interface – бағдарламалар арасындағы өзара әрекеттесуін реттейтін интерфейс. API қосымшаның әртүрлі компоненттері арасындағы байланыс пен деректер алмасуды қамтамасыз етеді[9].
ДБ	Деректер базасы – мәліметтерді сақтау, өңдеу және басқару жүйесі. Қант диабетімен ауыратын науқастардың деректерін сақтау үшін қолданылатын құрылым[11].
MVVM	Model-View-ViewModel — қолданба логикасын пайдаланушы интерфейсінің кодынан бөлу арқылы тиімді әзірлеуге болатын архитектуралық үлгі.[9]

1 кестенің жалғасы

UI	User Interface – қолданбаларға пайдаланушылармен өзара әрекеттесуге мүмкіндік беретін графикалық және интерактивті интерфейс элементтері[9].
UX	User Experience – қосымшамен жұмыс істеудің жалпы тәжірибесі. Пайдаланушыларға қолданбаны тиімді және ыңғайлы пайдалануды қамтамасыз етеді [8].
SDK	Software Development Kit – бағдарламалық қамтамасыз етуді әзірлеу үшін қажетті құралдар мен кітапханалар жиынтығы[8].
HTTP	Hypertext Transfer Protocol – веб-браузерлер мен серверлер арасында деректер алмасу хаттамасы[12].
JSON	JavaScript Object Notation – деректерді алмастыру және сақтау үшін қолданылатын жеңіл, адамға түсінікті формат[9].
IDE	Integrated Development Environment – бағдарламалау тілдерін жазу, түзету және тестілеу үшін пайдаланылатын орта[11].
CRUD	Create, Read, Update, Delete – мәліметтер базасымен жұмыс істегенде қолданылатын төрт негізгі операция: деректерді жасау, оқу, жаңарту және жою[8].
Java	Java – кросс-платформалық бағдарламалау тілі. Қосымшаның серверлік бөлігін әзірлеуде кеңінен қолданылады. Java бағдарламалары жоғары өнімділік пен қауіпсіздікке ие[8].
Spring	Spring Framework – java платформасында веб-қосымшаларды жасауға арналған ашық кодты бағдарламалық қамтамасыз ету фреймворкі. Бұл фреймворк серверлік қосымшалар үшін қолданылатын көптеген құралдар мен компоненттерді ұсынады[8].
React	React – қолданушы интерфейстерін құруға арналған JavaScript кітапханасы. React қосымшаның фронтендін құруда кеңінен қолданылады және компоненттік негізде жұмыс істейді[9].
Docker	Docker – қосымшаларды контейнерлеу және виртуализациялау үшін қолданылатын құрал. Docker бағдарламалар мен олардың тәуелділіктерін бірдей ортада іске қосуға мүмкіндік береді, бұл әзірлеу мен тестілеуді жеңілдетеді[10].

1.4 Жүйені модельдеу

Бұл бөлімде сәулет үлгілері мен құрылымдарына арналған дипломдық жобаның бөлімі сипатталған. Бұл қосымша Java Spring және reaction технологияларына негізделген және Docker контейнерлеу жүйесін қолдана отырып, оңай басқарылатын және масштабталатын жүйе ретінде жасалған.

Негізгі құрылымдық бөлімдер:

1. **Data Layer:** Бұл бөлімде медициналық деректер базасы және деректерді өңдеуге қажетті компоненттер орналастырылады. Java Spring фреймворкімен құрылған Spring Data JPA арқылы деректер базасына қатынас жасалады. Сонымен қатар, JPA репозиторийлері деректерді сақтау және алу үшін қолданылады. Мысалы, PatientRepository науқастардың деректерін сақтау үшін қызмет етеді. Деректер базасымен байланыс, REST API қызметтері арқылы жүзеге асады, олар React қолданушы интерфейсінен сұраулар алады.

2. **Business Logic Layer (Service Layer):** Бұл бөлімде бизнес логика жүзеге асырылады. Spring Boot қосымшасының бизнес логикасы осы жерде басқарылып, PatientService, MedicationReminderService сияқты қызметтер арқылы жұмыс істейді. Бұл қызметтер науқастың жағдайын бақылау, дәрі-дәрмектер туралы ескертулерді жіберу және басқа да медициналық операцияларды жүзеге асырады.

3. **Presentation Layer:** Қосымшаның пайдаланушы интерфейсі React кітапханасы арқылы жасалған. React компоненттері пайдаланушының деректерін көрсету, дәрі-дәрмек қабылдауды ескертулермен қамтамасыз ету, сондай-ақ науқастың медициналық ақпараттарын визуализациялау үшін қолданылады. React Router арқылы түрлі экрандар (мысалы, науқастың профилі, дәрі-дәрмек туралы ескертулер және т.б.) арасында навигация ұйымдастырылады. Пайдаланушы интерфейсінің логикасы Redux арқылы басқарылып, деректерді сақтау және синхрондау жүзеге асырылады.

4. **API Layer: Spring Boot** арқылы әзірленген RESTful API қосымша мен клиент арасындағы деректер алмасуды қамтамасыз етеді. API, науқастардың медициналық деректерін алу, жаңарту, дәрі-дәрмек қабылдау туралы ескертулер жіберу және басқа да операцияларды орындау үшін қолданылады. Бұл API React фронтендімен интеграцияланып, пайдаланушы әрекеттеріне сәйкес сұраулар мен жауаптарды өңдейді.

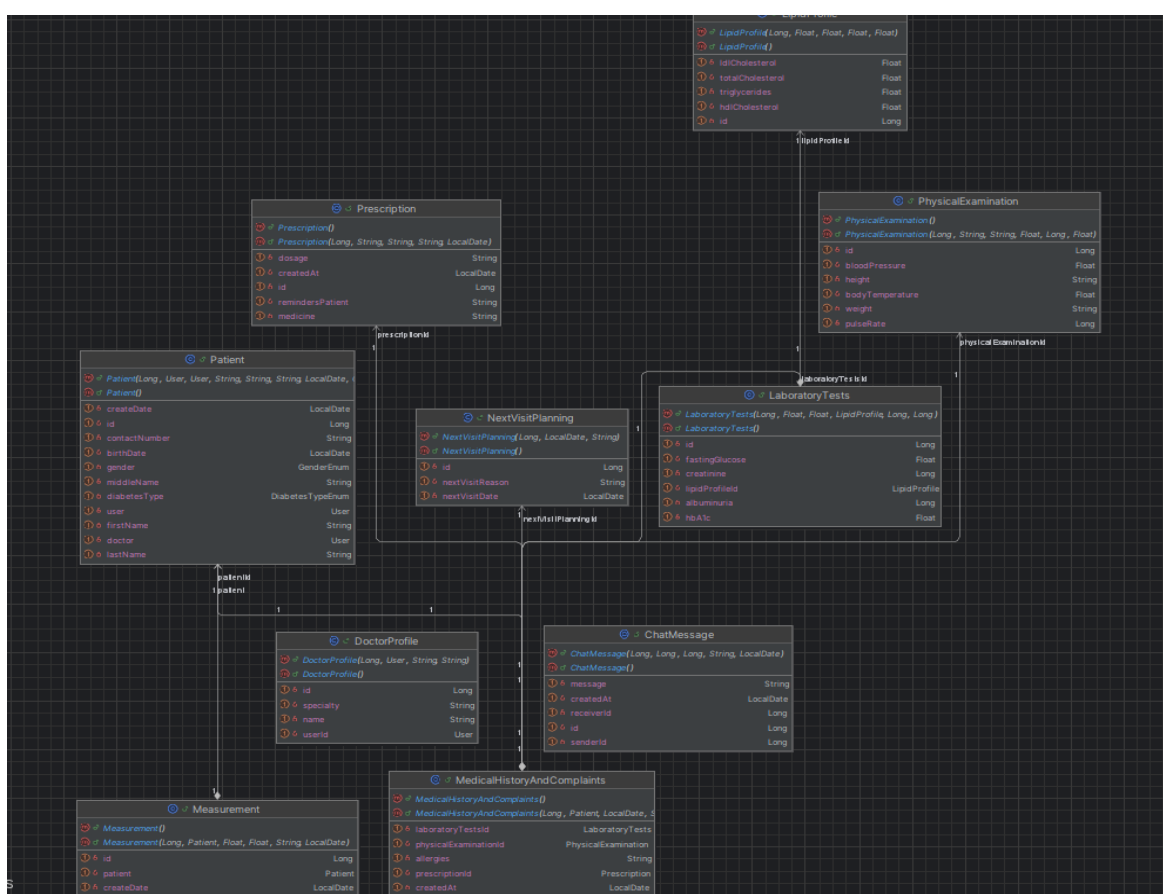
5. **Dependency Injection (DI) Layer: Spring Framework** қосымшасының барлық тәуелділіктерін инъекциялау үшін қолданылады. Spring инъекция механизмін пайдалану арқылы компоненттер арасындағы байланыс анықталады және автоматты түрде инъекция жасалады. Бұл бөлікте @Autowired аннотациясы арқылы сервис, репозиторий, контроллер және басқа да компоненттер байланысады.

6. **Docker Layer:** Қосымшаның барлық компоненттері Docker контейнерлері арқылы орнатылып, басқарылатын болады. Docker контейнерлеу технологиясы қосымшаның дамуын, тестілеуін және өндірістік ортада іске

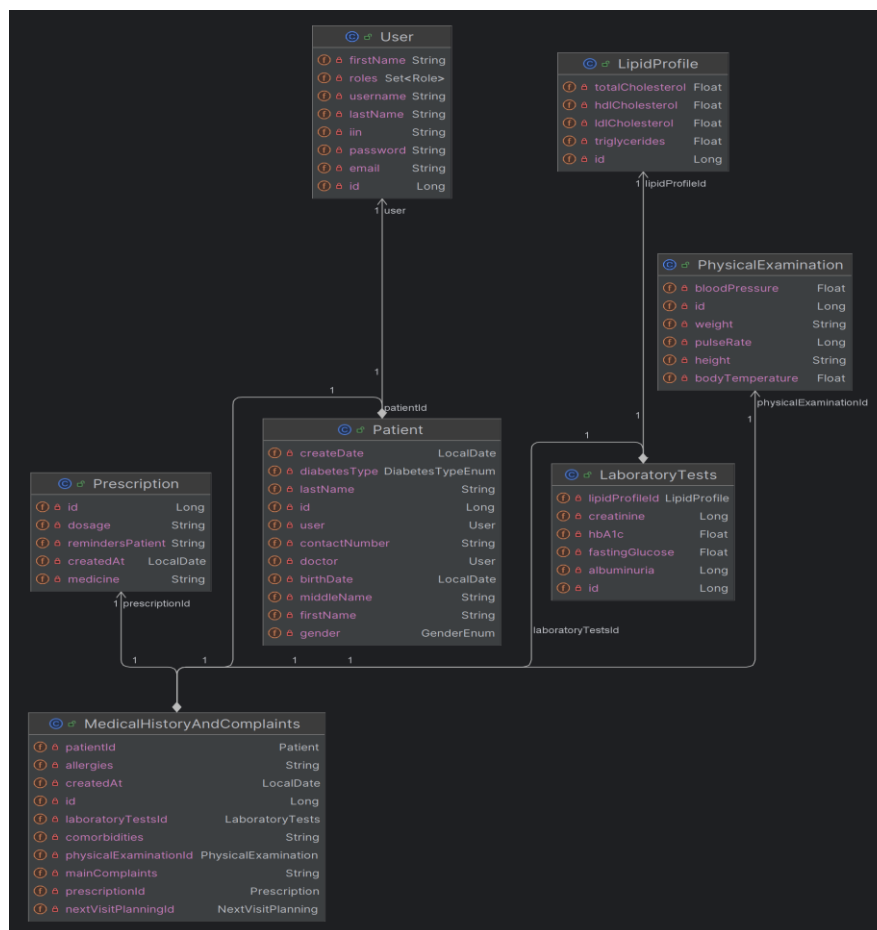
қосылуын оңайлатады. Әр компонент (Spring Boot, React, деректер базасы) бөлек контейнерлерде іске қосылады, бұл оларды бір ортада толық басқаруды және масштабтауды қамтамасыз етеді. Docker Compose арқылы әртүрлі контейнерлерді бірге іске қосуға болады, бұл қосымшаның толықтай оңай басқарылуын қамтамасыз етеді.

7. **Административті панель: React** технологиясын пайдалану арқылы жобаның әкімшілік басқару панелі жасалды. Бұл панель қосымшаның мазмұнын басқаруға, пайдаланушы деректерін және медициналық ақпараттарды тиімді түрде басқаруға мүмкіндік береді. React негізіндегі панель платформалар арасында бір код базасын қолдануға мүмкіндік береді.

1.5.1 ER диаграммасы



1 сурет – Науқастың күнделікті деректерін енгізу диаграммасы



2 сурет – Науқастың деректерін енгізу диаграммасы

ER (Entity-Relationship) диаграммасы – дерекқорды жобалау үшін қолданылатын графикалық құрал. Ол жүйедегі объектілер мен олардың арасындағы қатынастарды бейнелейді. ER диаграммасы мәліметтерді ұйымдастыру мен дерекқорды басқару үшін маңызды болып табылады. Ол негізінен келесі элементтерді қамтиды:

1. **Entity (Объект)**: ER диаграммасында "Entity" – нақты немесе абстракт объекті немесе тұтас жүйе. Мысалы, аурухана жүйесінде "Науқас" немесе "Дәрігер" сияқты объектілер болуы мүмкін. Әрбір объектінің атрибуттары болуы мүмкін (мысалы, Науқас үшін аты, жасы, диагнозы).

2. **Attribute (Атрибут)**: Атрибут – объектінің сипаттамасы немесе қасиеті. Мысалы, "Науқас" объектісінің атрибуттары оның аты, жасы, диагнозы болуы мүмкін.

3. **Relationship (Қатынастар)**: Қатынас – екі немесе одан да көп объекті арасында болатын байланыс. Мысалы, "Науқас" және "Дәрігер" арасындағы қатынас "Қаралу" немесе "Емдеу" деп атала алады.

4. **Primary Key (Негізгі кілт)**: Әрбір объектінің бірегей идентификаторы. Бұл атрибут деректердің бірегейлігін қамтамасыз етеді, мысалы, науқастардың ID коды.

ER диаграммасының түрлері:

- **One-to-One (1:1):** Бір объекті екінші объектімен бір байланысқа ие. Мысалы, әр науқастың бір дәрігері болуы мүмкін.
- **One-to-Many (1:N):** Бір объекті бірнеше объектімен байланысқан. Мысалы, бір дәрігердің көптеген науқастары болуы мүмкін.
- **Many-to-Many (N:M):** Көптеген объектілер арасында көптеген байланыстар бар. Мысалы, бір дәрігер бірнеше науқаспен, ал әр науқас бірнеше дәрігермен байланыса алады.

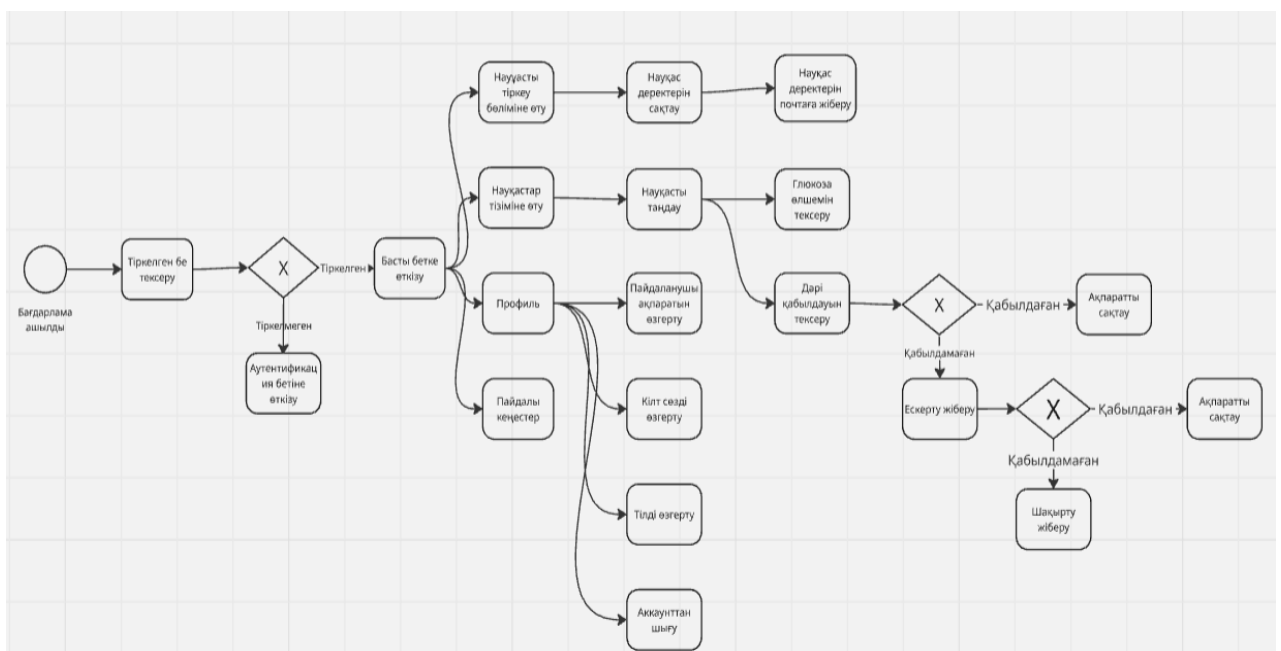
ER диаграммасының элементтері:

- **Қораптар** (Entity немесе атрибуттарды көрсетеді).
- **Сызықтар** (Қатынастарды көрсетеді).
- **Ромбтар** (Қатынастарды қосатын элементтер).

ER диаграммасын құру кезінде, алдымен жүйедегі негізгі объектілерді (entity) анықтап, олардың арасындағы байланыстарды (relationship) анықтаймыз. Содан кейін, әрбір объектінің атрибуттарын қосамыз және байланыстарды айқындаймыз.

Бұл диаграмма дерекқордың құрылымын тиімді жобалау үшін өте пайдалы болып табылады.

1.5.2 BPMN диаграммасы (дәрігер)



3 сурет – BPMN диаграммасы (дәрігер)

BPMN (Business Process Model and Notation) диаграммасы – бизнес процестерін моделдеуге арналған стандартты графикалық тіл. Бұл диаграмма бизнес процестерін түсінікті түрде көрсету, талдау және оңтайландыру үшін

пайдаланылады. BPMN диаграммасы ұйымдардың жұмысын жүйелі түрде бейнелеуге мүмкіндік береді және процесстер арасындағы байланыстарды, әрекеттер мен шешімдерді айқын түрде көрсетеді.

BPMN диаграммасының негізгі элементтері:

1. **Процесс:** BPMN диаграммасының негізі – бизнес процесі, яғни ұйымдағы белгілі бір мақсатты жүзеге асыру үшін орындалатын әрекеттер тізбегі.

2. **Қызметтер (Activities):** Процесс ішінде орындалатын нақты әрекеттер немесе тапсырмалар. Бұл қызметтер бірнеше түрге бөлінеді:

- **Task (Тапсырма):** Бір әрекет немесе жұмыс бір уақытта орындалатын элемент.

- **Sub-Process (Қосымша процесс):** Әрекеттердің топтамасы, оны басқа диаграммамен көрсетіп, ішіне қосымша процесстерді қосуға болады.

3. **Іс-шаралар (Events):** Процестің басталуы, аяқталуы немесе аралық жағдайлар. Іс-шаралар әртүрлі түрлерге бөлінеді:

- **Start Event (Бастау іс-шарасы):** Процестің басталуын көрсетеді.
- **End Event (Аяқталу іс-шарасы):** Процесс немесе әрекеттің аяқталуын көрсетеді.

- **Intermediate Event (Аралық іс-шара):** Процесс кезінде туындайтын жағдайларды (мысалы, уақыт шегі немесе сыртқы оқиғалар) білдіреді.

4. **Қатынастар (Gateways):** Процесс ішіндегі бағыттарды немесе шартты бөліністерді көрсетеді. Бұл элементтер бірнеше жағдайларға немесе әрекеттерге жол ашады:

- **Exclusive Gateway (XOR):** Егер бір ғана шарт орындалса, процесстің келесі қадамы таңдалады.

- **Parallel Gateway (AND):** Барлық жағдайлар орындалса, бірнеше әрекет қатарынан жүзеге асады.

- **Inclusive Gateway (OR):** Бір немесе бірнеше шарттар орындалған жағдайда бірнеше әрекет орындалуы мүмкін.

5. **Құралдар (Artifacts):** Диаграмманың қосымша ақпараттарын қосуға мүмкіндік береді. Бұл элементтер, мысалы:

- **Data Objects (Дерек объектілері):** Процесс барысында қолданылатын немесе жасалатын деректер.

- **Groups (Топтар):** Әр түрлі элементтерді топтау үшін қолданылады.

- **Annotations (Ескертпелер):** Диаграммаға қосымша түсініктемелер немесе ақпараттарды қосуға мүмкіндік береді.

6. **Ағындар (Flows):** Әрекеттер мен оқиғалар арасындағы байланыстарды көрсететін сызықтар:

- **Sequence Flow (Тізбекті ағын):** Әрекеттердің немесе іс-шаралардың орындалу тәртібін көрсетеді.

- **Message Flow (Хабарламалар ағыны):** Процесс ішіндегі әртүрлі элементтер арасында хабарламалармен алмасуды білдіреді.

- **Association (Қатысу):** Деректер мен процестер арасындағы байланыстарды көрсетеді.

BPMN диаграммасының артықшылықтары:

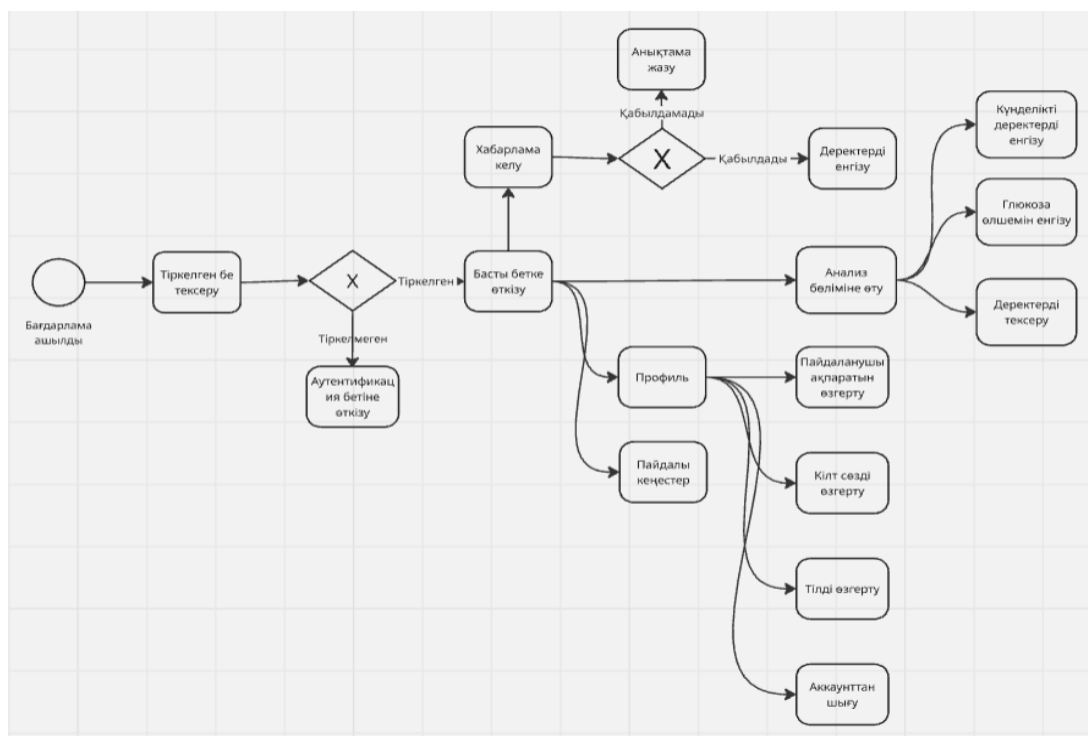
- **Түсінікті:** Барлық бизнес пайдаланушыларға (техникалық емес) бизнес процестерін түсінуге көмектеседі.
- **Стандартты:** BPMN диаграммалары әлем бойынша кеңінен қабылданған стандарт болып табылады, сондықтан олар бір ұйымнан екіншісіне оңай аударылады.
- **Жүйелі:** Біз бір уақытта бірнеше процесс пен әрекетті жүйелі түрде көрсету мүмкіндігіне ие боламыз.
- **Құжаттау:** Процестерді нақты түрде құжаттап, оларды талдауға және оңтайландыруға мүмкіндік береді.

BPMN диаграммаларын пайдалану салалары:

- **Бизнес процестерін талдау:** Қазіргі бизнес процестерін зерттеп, тиімділігін арттыру үшін пайдалануға болады.
- **Автоматтандыру:** Процестерді автоматтандыру үшін нақты сұлбаларды жасауға көмектеседі.
- **Оңтайландыру:** Процестердің қай жерлерде өзгерістер енгізу қажеттігін көруге мүмкіндік береді.

BPMN диаграммалары, әсіресе бизнес процестерін нақты бейнелеу мен оларды бақылау мақсатында, маңызды құрал болып табылады.

1.5.3 BPMN диаграммасы (науқас)



4 сурет – BPMN диаграммасы (науқас)

1. Бағдарлама ашылады
 - Қолданушы қосымшаны іске қосады.
2. Тіркелген бе тексеру
 - Жүйе қолданушы тіркелген бе, жоқ па екенін тексереді.
3. Егер тіркелмесе → "Аутентификация" не "Тіркелу" бетіне өтеді
 - Қолданушы тіркелмеген болса, ол тіркелу немесе кіру экранына бағытталады.
4. Егер тіркелген болса → "Басты бетке өту"
 - Тіркелген қолданушы тікелей басты бетке өтеді.

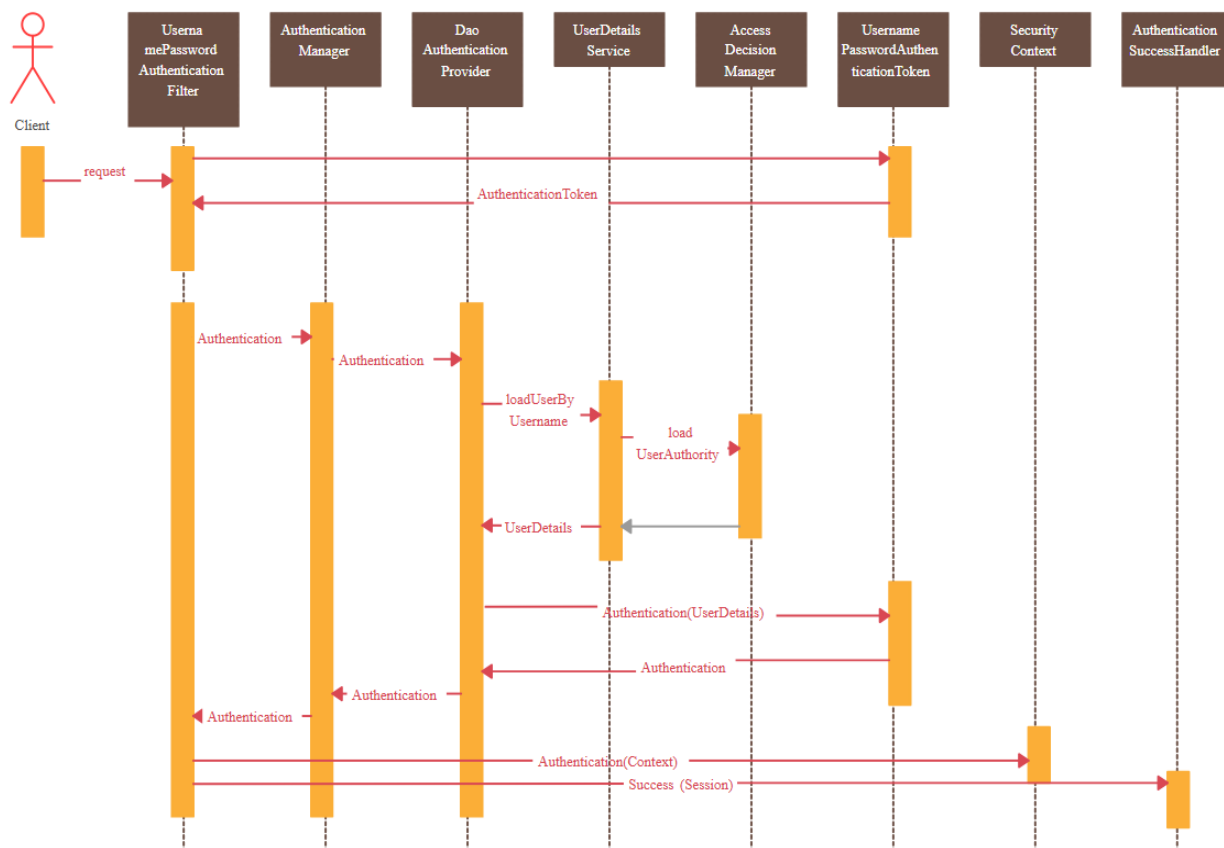
Басты беттен мүмкін болатын әрекеттер:

 - Профиль → Жеке мәліметтерді қарау.
 - Пайдаланушылар → Басқа қолданушылар тізімін қарау.
 - Пайдаланушының ақпаратын өзгерту → Профильді өңдеу.
 - Кіші сөзді өзгерту → Құпия сөзді жаңарту.
 - Тілді өзгерту → Интерфейс тілін ауыстыру.
 - Аккаунттан шығу → Жүйеден шығу.

Медициналық деректермен жұмыс:

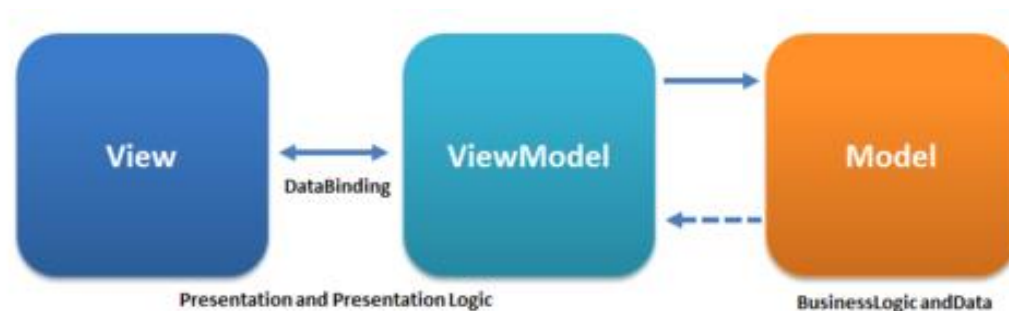
 - Анализ бөліміне өту → Медициналық көрсеткіштермен жұмыс.
 - Деректерді енгізу
 - Глюкоза өлшемдерін енгізу
 - Күнделікті деректерді енгізу
 - Деректерді тексеру
 - Анықтама жазу → Қолданушы дәрігерге хабарлама немесе өтініш жазады.
 - Қабылданды/Қабылданбады → Нәтижеге байланысты хабарлама шығады.

1.6 UML диаграммасы



5 сурет – UML диаграмма (Класс диаграммасы)

1.6.1 MVVM (Модель-View-ViewModel)



6 сурет – MVVM сызбасы

MVVM ол медициналық деректерді бақылау және инвазивті емес құрылғыларды басқару үшін веб-қосымшаларда қолданылады. Бұл үлгі кодтық

базаны басқаруды жеңілдетеді және әзірлеу мен тестілеу процесінің тиімділігін арттырады.

Негізгі компоненттер:

Үлгі: қолданба деректері мен бизнес логикасын ұсынады. Бизнес ережелерін, деректерге қол жетімділікті, модель сыныптарын және деректердің сенімділігін басқарады.

Көрініс: пайдаланушы интерфейсінің барлық компоненттерінен тұрады. Модельден деректерді көрсетеді және пайдаланушы командаларын (оқиғаларын) көрініс моделіне жібереді. Ұсыну белсенді емес, яғни бизнес логикасы жоқ декларативті көрініс.

Өкілдік моделі: модель мен өкілдік арасындағы делдал ретінде әрекет етеді. Көріністе қол жетімді нысанда деректерді ұсынады және көріністегі пайдаланушылардың әрекеттерін басқарады.

Жобаның артықшылықтары:

Бөлу: логиканы бөлу-үлкен артықшылық. Бизнес логикасы мен деректерді көрсету логикасын бөлу қолданбаларды басқаруды және масштабтауды жеңілдетеді. Әзірлеушілер мен дизайнерлер тиімді және тәуелсіз жұмыс істей алады.

Қарапайым тест: бизнес логикасын презентация модельдеріне бөлу арқылы сіз бұл компоненттерді пайдаланушы интерфейсінен (view) бөлек оңай тексере аласыз.

Деректер қосылымы: әдетте, көрініс моделі көрініске деректер қосылымын қолдайды. Бұл байланыс кодтың стандартты бөлігін қысқартады және интерфейс пен пайдаланушы әрекеттері арасында синхрондауды жеңілдетеді.

Кодты қайта пайдалануды және техникалық қызмет көрсетуді жеңілдетеді: компоненттерді қолданбаның басқа бөліктерінде немесе басқа қолданбаларда оңай қайта пайдалануға болады. Бизнес логикасын немесе деректерді көрсету тәсілін өзгерту техникалық қызмет көрсетуді жеңілдетеді, өйткені көріністі өзгертудің қажеті жоқ.

Медициналық деректерді бақылау және инвазивті емес құрылғыларды басқару үшін веб-қосымшалардағы қозғалыстарды пайдаланыңыз.

Медициналық деректерді бақылау және инвазивті емес құралдарды басқару үшін веб-қосымшаны әзірлеу біз жаңа мүмкіндіктерді тиімді енгізе алатынымызды және кодты оңай сақтай алатынымызды білдіреді. Мысалы, дәрі - дәрмектерді қабылдау уақытын картаға түсірудегі өзгерістер тек презентация моделі арқылы жүзеге асырылуы мүмкін, Бұл бизнес логикасын төменгі деңгейде өзгерту қажеттілігін жояды. Сонымен қатар, дәрілік деректер үлгісіндегі өзгерістер (мысалы, қант деңгейін Жаңа бақылау, дәрі-дәрмек түрлерін қосу) даму ортасын тұрақты және икемді етеді, пайдаланушы интерфейсінде ешқандай өзгеріс жоқ.

1.7 Технологиялар бөлімі

Қант диабетімен ауыратын науқастарға арналған ақпараттық жүйені әзірлеу барысында Java, Spring, React JS, Docker, Git және GitLab сияқты заманауи технологиялар қолданылды. Бұл құралдар өздерінің тұрақтылығы, қауіпсіздігі және икемділігі арқасында жобаның барлық компоненттерінің үздіксіз жұмыс істеуін қамтамасыз етті. Әр технология белгілі бір мақсатқа сай таңдалып, жүйенің тиімділігіне оң әсер етті.

- **Java:** Жобаның серверлік (backend) бөлігі Java бағдарламалау тілінде әзірленді. Бұл тіл платформаға тәуелсіз, өнімділігі жоғары және кең қолданылатын тілдердің бірі. Java арқылы бизнес-логика құрылып, дерекқормен байланыс орнатылды. Сонымен қатар, оның қауіпсіздік пен модульдік артықшылықтары жүйенің тұрақты жұмыс істеуіне ықпал етті[8].

- **Spring Framework:** Java экожүйесіне арналған Spring фреймворкі серверлік қосымшаның негізін құрады. Spring Boot арқылы REST API интерфейстері жасалып, пайдаланушылар мен жүйе арасындағы деректер алмасуы қамтамасыз етілді. Spring Security модулі арқылы авторизация және аутентификация механизмдері енгізілді, ал Spring Data JPA дерекқормен жұмыс жасау процесін жеңілдетті[8].

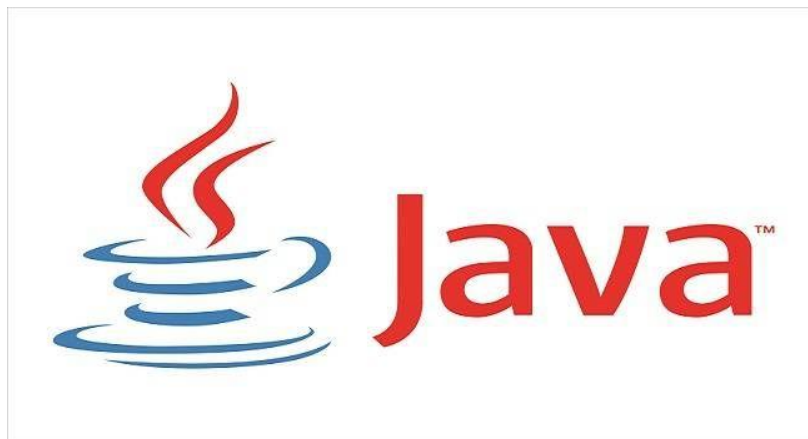
- **React JS:** Қолданушы интерфейсін құру үшін React JS кітапханасы қолданылды. Бұл кітапхана компоненттік құрылым арқылы интерфейсін модульдік және қайта қолдануға ыңғайлы етіп құруға мүмкіндік береді. React виртуалды DOM қолдану арқылы өнімділікті арттырып, интерфейсін жылдам жүктелуін қамтамасыз етті[9].

- **Docker:** Жобаның барлық компоненттері (frontend, backend, database) Docker контейнерлерінде іске қосылды. Бұл жүйенің конфигурацияларын оңай басқаруға және әртүрлі ортада (локальды, серверлік) бірдей жұмыс жасауына мүмкіндік берді. Контейнеризация жүйені тестілеу және деплой процесін айтарлықтай жеңілдетті[11].

- **Git:** Бағдарламалық кодтың нұсқаларын басқару үшін Git қолданылды. Команда мүшелері өз тармақтарында (branch) жұмыс істеп, кейін өзгерістерді біріктіріп отырды. Git көмегімен код тарихын бақылау, өзгерістерді қайтару және командалық жұмыс жүргізу оңтайландырылды[12].

- **GitLab:** Жобаны ортақ репозиторийде сақтау және CI/CD процестерін автоматтандыру үшін GitLab платформасы пайдаланылды. GitLab арқылы кодты тексеру, сынақтан өткізу және өндіріске (production) шығару процестері автоматты түрде жүзеге асырылып, даму процесі жылдам әрі жүйелі болды[12].

1.7.1 Java бағдарламау тілі



7 сурет – Java логотипі

Java – объектіге бағытталған, қауіпсіз әрі платформаға тәуелсіз бағдарламалау тілі. Бұл тілдің тұрақтылығы мен кең қауымдастықпен қолдау табуы оны өндірістік жүйелерге, соның ішінде денсаулық сақтау саласына бейімдеуге ең қолайлы құрал етеді. Жоба барысында Java тілі backend жүйесінің негізгі компоненттерін, соның ішінде бизнес-логика, пайдаланушы мәліметтерін өңдеу және серверлік интеграция қызметтерін жасау үшін қолданылды.

Java тілі Spring фреймворкімен табиғи интеграцияға ие, бұл код құрылымын модульдік етіп ұйымдастыруға және масштабталатын жүйе жасауға мүмкіндік берді. Сонымен қатар, Java тілі арқылы RESTful сервистер жасалып, медициналық деректердің қауіпсіз берілуі қамтамасыз етілді[8].

1.7.2 Spring фреймворк



8 сурет – Spring Framework логотипі

Spring – Java негізіндегі веб-қосымшаларды құруға арналған ең танымал фреймворктердің бірі. Жобада Spring Boot модулі қолданылып, REST API арқылы қолданушы интерфейсімен өзара әрекеттесу қамтамасыз етілді. Сонымен қатар, Spring Security көмегімен аутентификация және рұқсат ету

процестері орындалды, бұл деректер қауіпсіздігін қамтамасыз етуде маңызды рөл атқарды.

Spring Data JPA модулі мәліметтер қорымен жұмыс істеуді оңтайландырды, ал Spring-тің модульдік құрылымы арқасында жобаны бірнеше бөлімге бөліп, әрқайсысын жеке дамытуға мүмкіндік туды. Бұл тәсіл медицина саласындағы күрделі және сенімді жүйелерді жасауға аса тиімді болды[8].

1.7.3 React JS кітапханасы



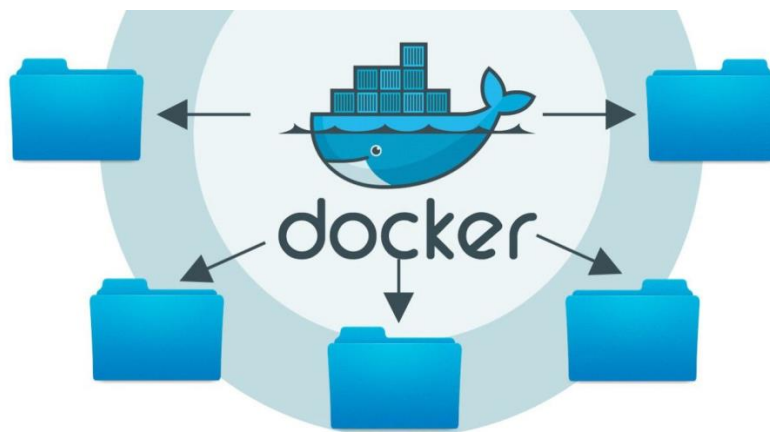
9 сурет – React JS логотипі

React – JavaScript негізінде жасалған заманауи кітапхана, қолданушы интерфейстерін құруда жиі қолданылады. Жобада React кітапханасы арқылы әкімшілік панель мен науқас интерфейстері жасалды. Компоненттік құрылым деректердің көрнекі әрі интерактивті түрде ұсынылуына мүмкіндік берді.

React қолдану нәтижесінде дәрігерлер мен науқастарға арналған интерфейстер жеңіл, жауапты (responsive) және модульдік құрылымда жасалды. Виртуалды DOM технологиясы кескіндер мен деректерді жылдам жүктеп, өнімділікті едәуір арттырды[9].

1.7.4 Docker контейнеризация

Docker – қосымшаларды контейнер түрінде оқшаулап іске қосуға арналған технология. Жобада backend (Java + Spring), frontend (React), және мәліметтер базасы компоненттері жеке контейнерлерге бөлініп, толық интеграцияланған микросервис архитектурасы жасалды.



10 сурет – Docker логотипі

Docker қолдану арқылы әзірлеу және деплой процесі автоматтандырылды. Әрбір компонент оқшауланған күйде жұмыс істеп, жүйенің икемділігі мен тасымалдануын қамтамасыз етті. Бұл – медициналық деректермен жұмыс істейтін жүйелер үшін сенімді әрі қауіпсіз шешім[11].

1.7.6 Git



11 сурет – Git логотипі

Git – бағдарламашыларға әртүрлі нұсқаларды оңай басқаруға және олармен жұмыс істеуге мүмкіндік беретін таратылған кодты басқару жүйесі. Git сіздің жобаңыздағы өзгерістерді тиімді басқара алады, қауіпсіз сақтай алады және кез-келген уақытта жүйенің кез-келген нұсқасына ауыса алады. Ашық бастапқы жүйені Linux ядросының әзірлеушісі Линус Торвальдс жасаған. GIT жүйелерінің ең маңызды ерекшелігі-олар үлестірілген архитектураны қолданады. Бұл дегеніміз, барлық әзірлеушілер Интернетке қосылмаған болса да, бүкіл жоба репозиторийінің көшірмесін компьютерде сақтай және іске қоса алады. Әзірлеушілер жергілікті жадпен жұмыс істей алады, содан кейін оларды басқа мүшелермен бөлісу үшін серверге өзгертулер жібере алады. Git платформасының тағы бір маңызды бөлігі-тармақталу және біріктіру мүмкіндігі. Бұл мүмкіндіктер әзірлеушілерге жаңа мүмкіндіктерді өздігінен қосуға және дамытуға мүмкіндік береді. Жүйе ақаулықтарды жою және жаңа мүмкіндіктерді енгізу кезінде үлкен ыңғайлылықты қамтамасыз етеді. Git

эртүрлі нұсқалар арасындағы айырмашылықтарды түсіну және жүйелі түрде бақылау үшін жоба кодының тарихын бақылауға мүмкіндік береді [12].

1.7.7 GitLab



12 сурет – GitLab логотипі

GitLab – бұл жобаның барлық кезеңдерін қамтитын және даму процесін толығымен автоматтандыратын қуатты жүйе. Бұл әзірлеушілерге кодтың қауіпсіздігін жақсартуға, мүмкіндігінше қысқа мерзімде өзгертулер енгізуге және бүкіл команданың бірлесіп жұмыс істеуіне мүмкіндік береді. Даму процесін басқаруды жеңілдету үшін оған код жазу, тестілеу, орналастыру және техникалық қолдау сияқты қадамдар енгізілген.

Сонымен қатар, Gitrap-та қауіпсіздікті жақсартуға арналған арнайы құралдар бар. Мысалы, кодты тексерудің автоматтандырылған жүйесі қолданбадағы осалдықтарды анықтайды және бағдарламашыларға қауіпсіздік бойынша ұсыныстар береді. Сонымен қатар, бағдарламашылар кодты бір жүйеде жасай алады, тексере алады және іске қоса алады. Gitrap қосымшасында модульдік архитектура бар, ол пайдаланушыларға тек қажетті модульдерді таңдауға және өз жобалары үшін тиісті мүмкіндіктерді пайдалануға мүмкіндік береді. Осы техникалық шешімдерді оңтайландырудың арқасында одан әрі даму процестері тез және тиімді жүзеге асырылады. Сонымен қатар, ашық ақпарат көзі жобалардың тиімділігі мен сапасын арттыруға көмектесетін қоғамдастық пен жобаларға үлкен үлес қосады [12].

2 Жобаның құрылымы

Қант диабетімен ауыратын науқастардың жағдайын бақылауға және инвазивті емес құрылғыларды басқаруға арналған веб-қосымша бірнеше негізгі компоненттерден тұрады. Бұл компоненттер әртүрлі функционалдық мүмкіндіктерді қамтиды және жүйенің үздіксіз, қауіпсіз әрі тиімді жұмыс істеуін қамтамасыз етеді.

Жобаның құрылымын құру барысында төмендегі аспектілерге ерекше көңіл бөлінді:

Медициналық деректерді сақтау және өңдеу

Қосымшада пайдаланушы (науқас) және дәрігер арасындағы өзара әрекет деректері, глюкоза деңгейі, зертханалық талдаулар, тағайындалған емдеу және өмір салты бойынша ақпараттар сақталады. Бұл деректер Firebase немесе арнайы шифрланған SQL дерекқорларында сақталып, авторизацияланған қолданушылар ғана қол жеткізе алады. Мәліметтердің сақталу қауіпсіздігі двухфакторлық аутентификация және деректерді шифрлау арқылы қамтамасыз етіледі.

Пайдаланушы интерфейсі (UI)

Пайдаланушы интерфейсі – жобаның маңызды бөлігі. Қосымша интерфейсі React JS кітапханасы арқылы әзірленіп, заманауи, интуитивті және жауапты дизайнмен ерекшеленеді. Интерфейс арқылы науқастар өз глюкоза деңгейлерін енгізіп, тағам күнделігін жүргізе алады, ал дәрігерлер науқастың динамикасын көріп, емдеу жоспарын түзете алады.

UI компоненттеріне мыналар кіреді:

- Интерактивті графиктер (HbA1c, глюкоза трендтері)
- Телемедицина және чат модулі
- Медициналық жазбалар және зертханалық мәліметтер панелі
- Дәрі қабылдауды еске салу терезелері

Интеграциялық модульдер

Веб-қосымша инвазивті емес құрылғылармен (мысалы, глюкометрлер немесе смарт-сағаттар) интеграциялануға бейімделген. Құрылғылардан алынатын деректер автоматты түрде жүйеге жүктеліп, пайдаланушы мен дәрігер үшін шынайы уақыт режимінде көрсетіледі. Бұл деректер машиналық оқыту алгоритмдері арқылы талданып, жүйе автоматты түрде ұсыныстар береді (инсулин дозасын түзету, диета және т.б.).

Тестілеу модульдері

Интеграциялық тестілеу

Барлық модульдердің өзара әрекетін қамтамасыз ету үшін интеграциялық тестілеу жүргізіледі. Бұл жүйенің тұрақты жұмыс істеуіне және пайдаланушы деректерінің дұрыс өңделуіне кепілдік береді.

Қауіпсіздік тестілеуі

Қолданба қауіпсіздігінің жоғары деңгейде болуын қамтамасыз ету мақсатында осал тұстарды анықтау үшін қауіпсіздік аудиттері мен автоматты сканерлер қолданылады. Бұл процесс науқас деректерінің құпиялығын және жүйенің сенімділігін қамтамасыз етеді.

2.1 Регистрация және авторизация бөлімі

13 сурет – Авторизация

Қант диабетімен ауыратын науқастардың деректерін сенімді қорғау және әрбір пайдаланушыға жеке қолжетімділік ұсыну мақсатында қосымшада заманауи тіркеу және аутентификация жүйесі енгізілді. Бұл жүйе пайдаланушылардың қауіпсіздігін қамтамасыз етумен қатар, олардың қосымшаны жекелендіріп пайдалануын да қолдайды.

14 сурет – Регистрация

Firestore аутентификация қызметінің қолданылуы

Жоба барысында Google ұсынған бұлтты платформа – Firebase Authentication қолданылды. Бұл қызмет электрондық пошта және құпия сөз арқылы тіркелу, Google немесе басқа әлеуметтік желілермен кіру, сонымен қатар SMS арқылы екі факторлы аутентификация (2FA) мүмкіндігін қамтамасыз етеді.

Firestore-тің артықшылықтары:

- Деректерді автоматты түрде шифрлау және қауіпсіз сақтау;
- Кіру сессияларын басқару;
- Пайдаланушы статусы (дәрігер, науқас) бойынша рөлдік жүйені бөлу;
- Әр түрлі құрылғылардан кіруді синхрондау.

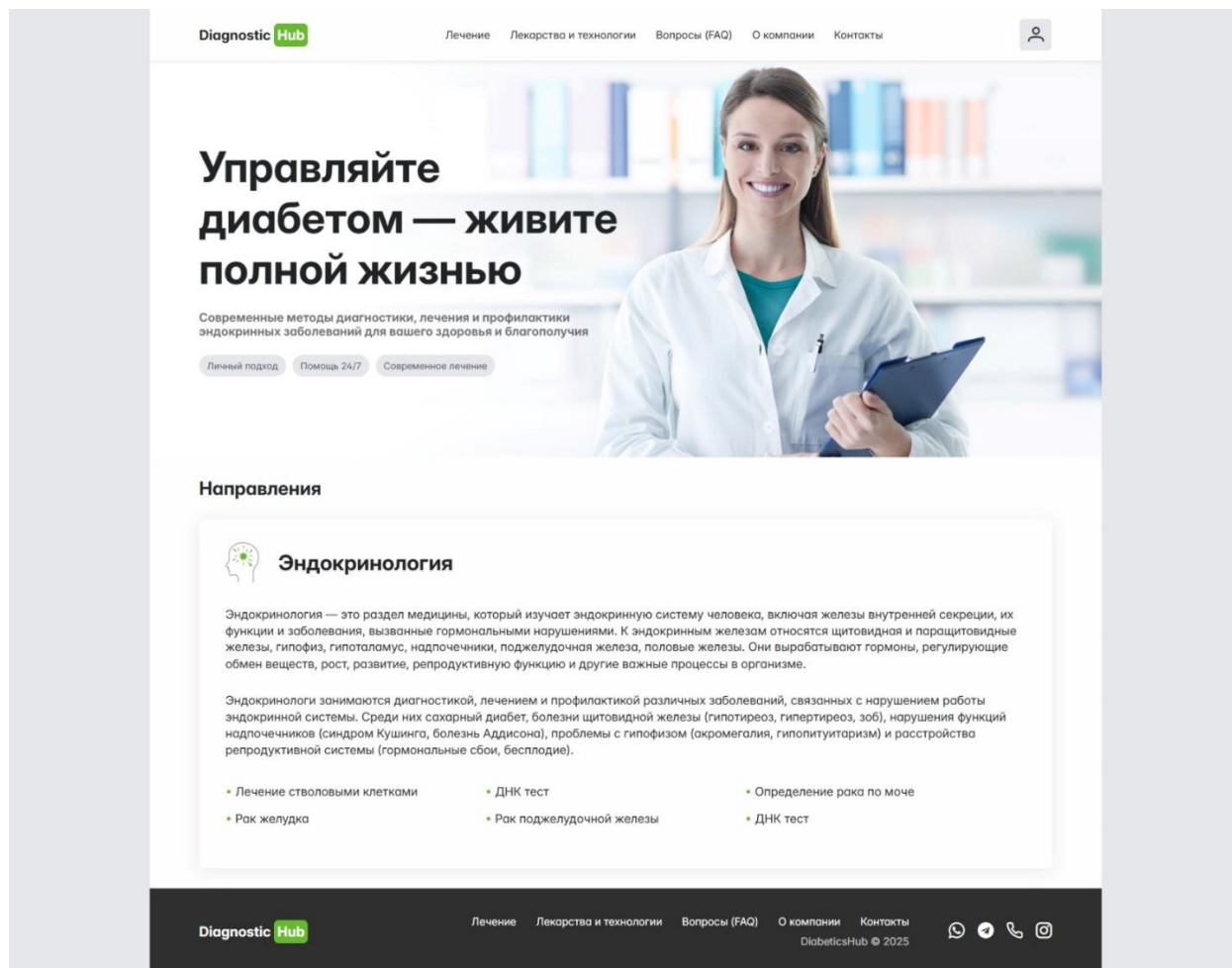
Қауіпсіздік пен сенімділік

Аутентификация жүйесі науқастардың медициналық деректерінің құпиялығын сақтау үшін маңызды рөл атқарады. Firestore көмегімен деректерге рұқсатсыз қолжетімділікке жол берілмейді. Әрбір тіркелген пайдаланушыға тек өзінің ақпараттарын көруге және өңдеуге рұқсат беріледі, ал дәрігерлер тек бекітілген науқастардың деректерін көре алады.

Жекелендірілген тәжірибе

Аутентификациядан өткеннен кейін жүйе әр пайдаланушының профиліне сай ақпаратты көрсетеді. Науқасқа – глюкоза деңгейі, емдеу жоспары және талдау нәтижелері, ал дәрігерге – науқастың динамикасы мен медициналық жазбалары ұсынылады. Бұл тәсіл пайдаланушы тәжірибесін жекелендіріп, жүйені интуитивті әрі тиімді етеді.

2.2 Қолданушының басты бет көрінісі



15 сурет – Веб-сайттың басты беті

Қосымшаның басты беті пайдаланушыларға қант диабетін басқаруға және эндокриндік денсаулығын жақсартуға арналған заманауи әдістермен танысуға мүмкіндік береді. Бұл интерфейс React JS технологиясының көмегімен жасалған, ол қолданушыларға ыңғайлы және түсінікті UI (қолданушы интерфейсі) ұсынады.

Басты бетте қолданушыларға бағытталған маңызды ақпараттар, соның ішінде диагностика, емдеу және профилактика әдістері ұсынылады. Сонымен қатар, жедел көмек, заманауи емдеу тәсілдері және жеке медициналық тәсілдер туралы бөлімдер бар.

React JS дизайны

Осы дипломдық жобада React JS технологиясын қолдана отырып, қант диабетімен ауыратын науқастарға арналған веб-қосымшаның қолданушы интерфейсі (UI) әзірленді. Бұл технология қолданбаның визуалды элементтерін жылдам және икемді түрде жасап, функционалды және көрнекі интерфейс

құруға мүмкіндік береді. Осы арқылы дизайн мен мазмұн арасындағы байланыс жақсарады.

Интерфейстің негізгі ерекшеліктері:

- Үлкен, айқын тақырып: «Қант диабетін басқарыңыз – толыққанды өмір сүріңіз»
- Пайдаланушыларды сенімділікпен қарсы алатын дәрігер бейнесі
- Мазмұндық бөлім: «Эндокринология» бағыты туралы толық сипаттама
- Қосымша қызметтер: ДНҚ тесті, ракқа анализ, жасушалық емдеу туралы ақпарат
- Жоғарғы мәзір арқылы емдеу, дәрі-дәрмек, жиі қойылатын сұрақтар, компания туралы және байланыс бөлімдеріне жылдам өту.

Қолданушыға бағытталған тәжірибе:

Басты бет қолданушыларға қажетті ақпаратты тез табуға және медициналық қызметтерге оңай қол жеткізуге арналған. Интерфейс құрылымы логикалық және визуалды түрде түсінікті болғандықтан, әртүрлі жастағы пайдаланушылар бұл қосымшаны қиындықсыз пайдалана алады.

Жалпы алғанда, бұл интерфейс диабетпен өмір сүрушілерге немесе эндокринологиялық мәселелері бар адамдарға медициналық шешімдерді ұсынуға бағытталған. React JS негізінде жасалған бұл шешім – сенімділік пен заманауилықтың үйлесімі.

2.3 Деректер беті

Бағдарламаның маңызды құрамдас бөліктерінің бірі ретінде **деректер беті** қолданушылар туралы маңызды медициналық және жеке ақпараттарды жинауға және басқаруға арналған. Бұл бет дәл сіз жіберген интерфейс секілді қарапайым, қолданушыға ыңғайлы түрде жасақталған және нақты құрылымымен ерекшеленеді.

Деректер бетінде пайдаланушының аты-жөні, жынысы, туған күні, ИИН/ЖСН, диабет түрі, байланыс апараты сияқты маңызды мәліметтерді енгізу үшін арнайы өрістер ұсынылған. Мұндай құрылым деректерді жүйелі түрде жинауға мүмкіндік беріп, медициналық қызметкерлерге науқас туралы толық әрі нақты ақпарат алуға жағдай жасайды.

Пайдаланушының ыңғайлылығы үшін әрбір өріс логикалық ретпен орналасқан және кейбір өрістер (мысалы, туған күн) күнтізбе түрінде ұсынылады, бұл деректерді қателіксіз енгізуге көмектеседі. Сонымен қатар, "Тип диабета" сияқты бірнеше нұсқалы таңдаулар визуалды түрде көрсетілген, бұл дұрыс таңдау жасау үдерісін жеңілдетеді.

Diagnostic **Hub**

Создания нового пациента

Имя пациента

Фамилия пациента

Отчество пациента

Пол пациента

Мужчина

Женщина

Дата рождения

ИИН/ЖСН пациента

Тип диабета

Тип 1

Тип 2

Гестационный

Другой

Никнейм для пациента

Почта пациента

Пароль

Телефон номер пациента

Diagnostic **Hub**

[Лечение](#)
[Лекарства и технологии](#)
[Вопросы \(FAQ\)](#)
[О компании](#)
[Контакты](#)

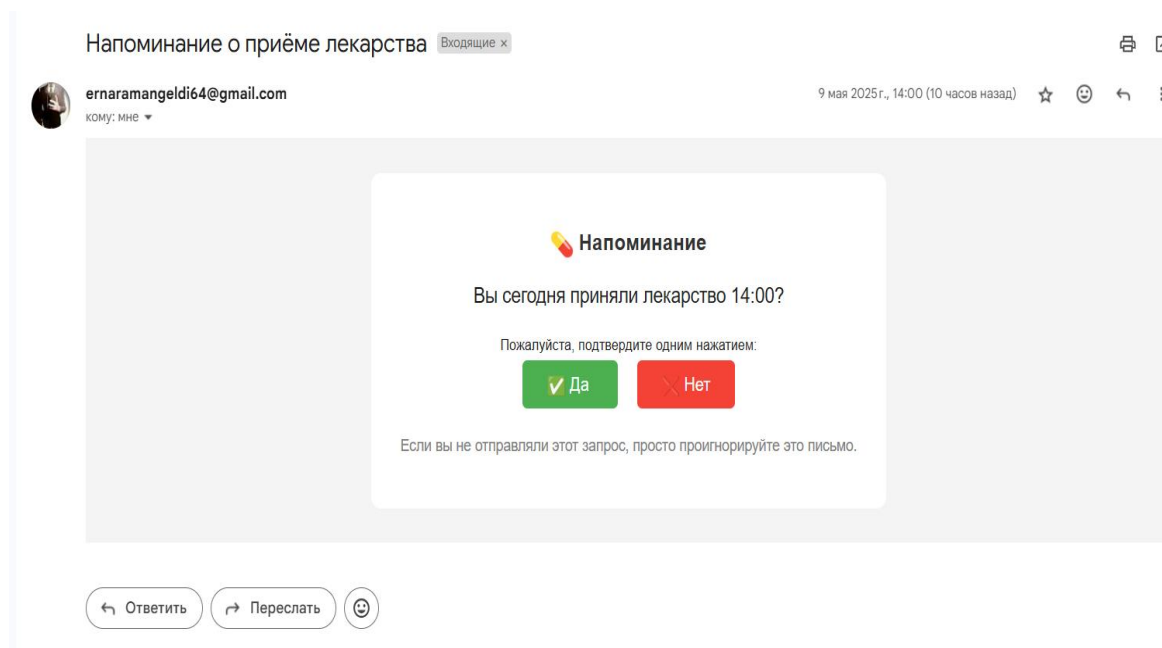
DiabeticsHub © 2025

16 сурет – Науқастың деректерін енгізу

Бұл бет сонымен қатар жүйеге жаңа науқасты тіркеу үшін маңызды қадамдардың бірі болып табылады. Деректерді толтырғаннан кейін "Создать пациента" батырмасы арқылы жазба жүйеге енгізіледі. Осындай тәсіл деректердің қауіпсіз әрі нақты сақталуын қамтамасыз етеді.

Деректер беті медицина қызметкерлеріне науқастарды басқару мен бақылау барысында қажетті ақпаратты тиімді пайдалану үшін үлкен көмек көрсетеді. Ол заманауи визуалды шешімдермен үйлесіп, электронды денсаулық сақтау платформасының ажырамас бөлігіне айналған.

2.4 Дәрі қабылдау беті



17 сурет – Дәрі қабылдауды еске түсіру

Дәрі қабылдау беті – медициналық қосымшаның науқасқа бағытталған ең маңызды функционалдық элементтерінің бірі. Бұл бет науқастардың дәрі-дәрмекті уақытылы және дұрыс қабылдауын бақылау арқылы емдеу процесінің тиімділігін арттыруға бағытталған. Қосымша қолданушылардың емделу тәртібіне деген жауапкершілігін арттырып, дәрігермен тұрақты байланыс орнатуға мүмкіндік береді.

Интерфейс сипаттамасы:

Бет құрылымы өте қарапайым әрі қолданушыға түсінікті. Орталықта дәрі қабылдау уақыты туралы нақты сұрақ орналасады:

«Сіз бүгін дәріні 14:00-де қабылдадыңыз ба?»

Төменде екі батырма ұсынылады:

Иә – пайдаланушы дәріні қабылдағанын растайды;

Жоқ – пайдаланушы дәріні қабылдамағанын хабарлайды.

Бұл қарапайым өзара әрекет формасы интерфейсін жеңілдетіп, жылдам жауап беруге жағдай жасайды.

Функционалды мүмкіндіктері

Автоматты еске салу жүйесі:

Дәріні қабылдау уақыты келгенде, қосымша push-хабарлама немесе email арқылы еске салады.

Жеке дәрілік жоспар:

Әр науқасқа арналған дәрі-дәрмек кестесі сақталады. Бұл кестеде препараттың атауы, қабылдау уақыты және қабылдау ұзақтығы (күндер саны) көрсетіледі.

Күнтізбе арқылы бақылау:

Қолданушы өткен күндердегі дәрі қабылдау статистикасын күнтізбе арқылы көре алады. Мысалы, жасыл күндер – қабылданған, қызыл күндер – қабылданбаған күндер.

Дәрігерге есеп жіберу:

Жауап берілгеннен кейін ақпарат автоматты түрде базаға жазылып, дәрігердің панелінде көрінеді. Бұл дәрігерге науқастың емді сақтап жүргенін қашықтан бақылауға мүмкіндік береді.

Хат арқылы растау (Email integration):

Кейбір жағдайларда науқастарға электрондық пошта арқылы да еске салғыш жіберіледі, онда тікелей сілтеме арқылы "Иә" немесе "Жоқ" батырмасын басу арқылы жауап беруге болады.

Қате немесе өткізіп алған жағдайда екінші еске салу:

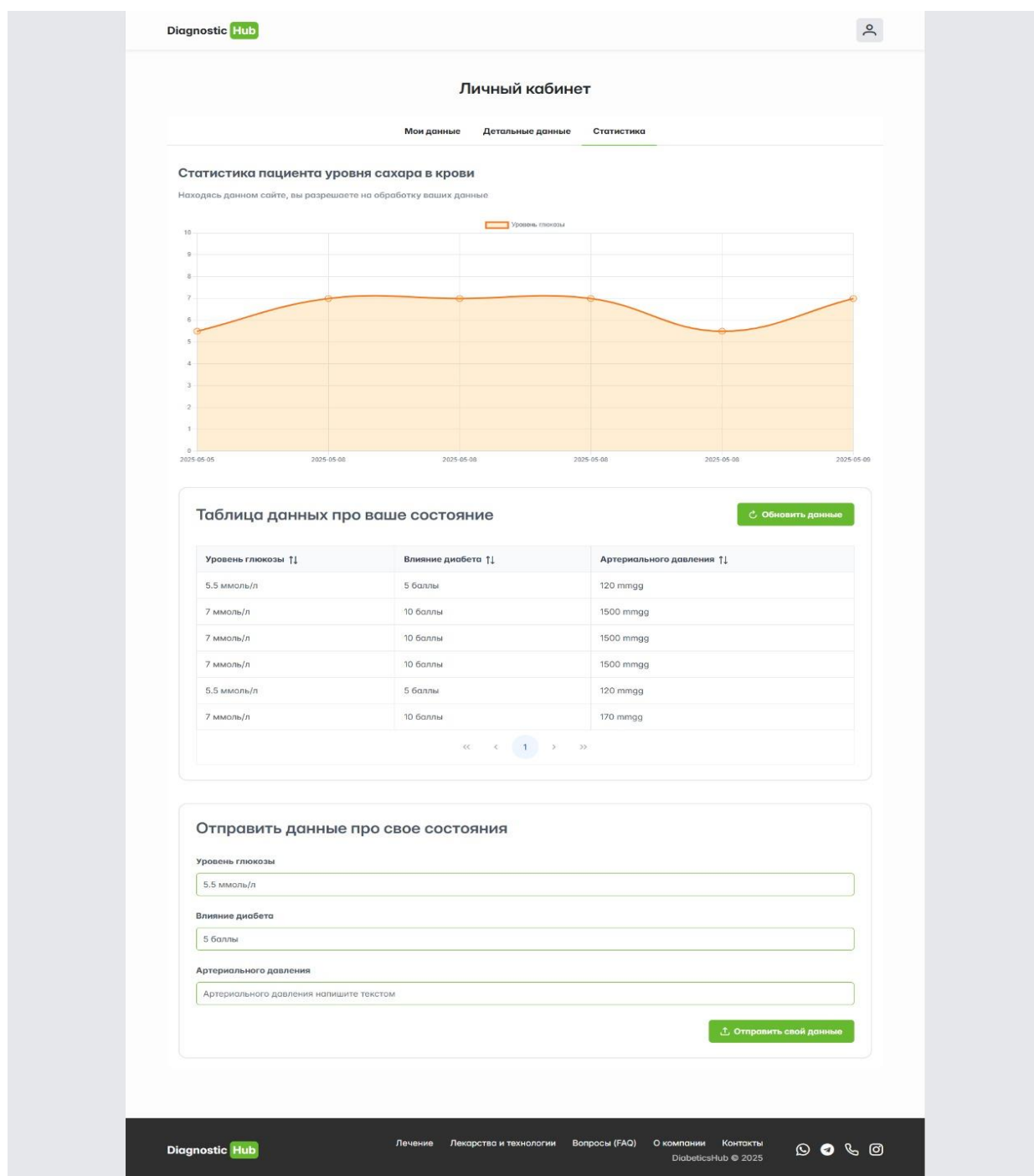
Егер белгілі бір уақыт аралығында жауап болмаса, жүйе екінші рет еске салу хабарламасын жібереді.

Қауіпсіздік және деректерді қорғау:

Барлық дәрі қабылдау деректері Firebase Firestore немесе сервер базасында қауіпсіз түрде сақталады. Қосымшада пайдаланушылардың жеке мәліметтері (аты-жөні, дәрі атауы, кесте) тек авторизацияланған дәрігерлер мен өз иесіне ғана көрінетін етіп шектелген.

2.5 Глюкоза өлшемі

Науқастың денсаулық жағдайын күнделікті бақылау – емделудің тиімділігін арттыратын және аурудың даму барысын уақытылы анықтауға мүмкіндік беретін маңызды процесс. Қосымшаның бұл беті – науқастың қан құрамындағы қант деңгейін, диабет ықпалын және қан қысымын бақылауға арналған арнайы функционалды модуль.



18 сурет – Қан құрамы деңгейіне арналған мониторинг

Негізгі мүмкіндіктер:

1. Графикалық статистика (диаграмма):

Экранның жоғарғы жағында науқастың күн сайын енгізген қант деңгейінің статистикасы көрсетілген. Диаграммада әрбір күннің деректері визуалды түрде бейнеленіп, уақыт өте келе өзгерістерді бақылауға мүмкіндік береді. Бұл науқасқа да, дәрігерге де емнің әсерін нақты көруге көмектеседі.

2. Мәліметтер кестесі:

Диаграмманың астында күнделікті енгізілген деректер егжей-тегжейлі кесте түрінде көрсетіледі:

- Қант деңгейі (ммоль/л).
- Диабет ықпалы (ұпаймен).
- Артериялық қан қысымы (мм сынап бағанасы бойынша). Бұл кесте жүйеде сақталған деректердің толық көрінісін береді және қажет болған жағдайда оларды жаңартуға болады.

3. Деректерді енгізу формасы:

Беттің төменгі жағында арнайы форма бар, ол арқылы науқасқа күнделікті өз жағдайын жаза алады.

- Қант деңгейі.
- Диабет әсері.
- Қан қысымы мәтін түрінде енгізіледі.

Бұл мәліметтер "Өз деректерімді жіберу" батырмасы арқылы базаға жіберіледі және бірден статистикаға қосылады.

Қосымша ерекшеліктер:

- Деректерді жаңарту:

Науқас "Обновить данные" батырмасын басу арқылы кестені нақты уақыт режимінде жаңарта алады.

- Автоматты диаграмма генерациясы:

Әрбір жаңа дерек енгізілген сәтте жүйе автоматты түрде диаграмманы жаңартып, өзгерістерді визуалды түрде көрсетеді. Бұл трендтерді оңай байқауға және болжам жасауға мүмкіндік береді.

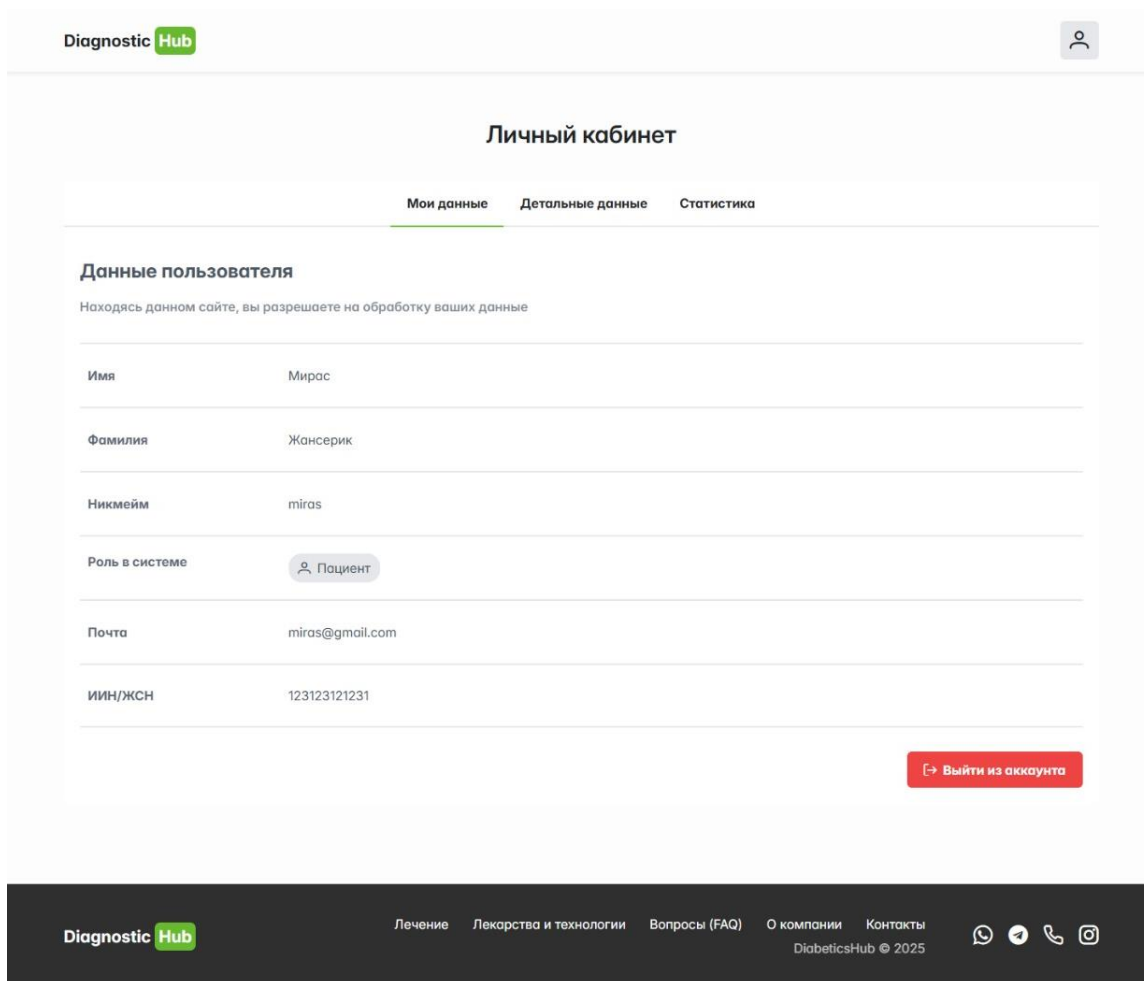
- Деректердің қауіпсіз сақталуы:

Барлық енгізілген ақпараттар қауіпсіз түрде дерекқорда сақталады және тек авторизацияланған пайдаланушыларға (науқас пен дәрігер) ғана көрінеді.

- Қолданушының бақылау құралы:

Бұл бет науқасқа өз денсаулығы үшін жауапкершілікті арттыруға көмектеседі және медициналық қызметкерлерге нақты шешім қабылдауда үлкен қолдау көрсетеді.

3 Қолданушы интерфейсінің жобасы



19 сурет – Науқастың жеке бөлімі

(Қан диабетімен ауыратын науқастарға арналған)

Қосымшаның қолданушы интерфейсі - диабетпен ауыратын науқастардың күнделікті медициналық көрсеткіштерін бақылауға және оларды дәрігермен бөлісуге бағытталған. Жобаланған интерфейс науқасқа интуитивті, қарапайым әрі ыңғайлы қызмет көрсетуге негізделген. Қан диабеті секілді созылмалы ауруды бақылау үшін деректерді уақытылы енгізіп отыру өте маңызды, сол себепті интерфейс функционалды және қолжетімді етіп әзірленді.

Интерфейс құрылымының негізгі элементтері:

1. Басты бет (Личный кабинет):

Науқас жүйеге кірген соң бірінші болып жеке парағын көреді. Мұнда үш негізгі бөлім орналасқан:

- Менің деректерім.
- Толық мәліметтер.
- Статистика.

Әр бөлім пайдаланушының медициналық көрсеткіштерін қарап, жаңартуға мүмкіндік береді.

2. Статистикалық диаграмма:

Бұл график науқастың қан құрамындағы қант деңгейінің күнделікті өзгерісін бейнелейді. Уақыт шкаласы бойынша трендті бақылауға болады. График визуалды түрде жеңіл қабылданатын, түрлі түсті сызықтармен бейнеленген.

3. Медициналық көрсеткіштер кестесі:

Қант деңгейі, диабет деңгейінің ықпалы және артериялық қан қысымы секілді маңызды мәліметтер арнайы кесте түрінде беріледі. Пайдаланушы кестені жаңартып, өз денсаулығының өзгерістерін бақылап отыра алады.

4. Деректер енгізу формасы:

Пайдаланушы арнайы форманы толтыру арқылы өз ағзасындағы өзгерістерді енгізе алады:

- Қандағы глюкоза деңгейі (ммоль/л),
- Диабет ықпалының ұпайы,
- Артериялық қысым (мм сын.бағ.).

Бұл форма құрылымы қарапайым және кез келген жастағы қолданушыға қолайлы.

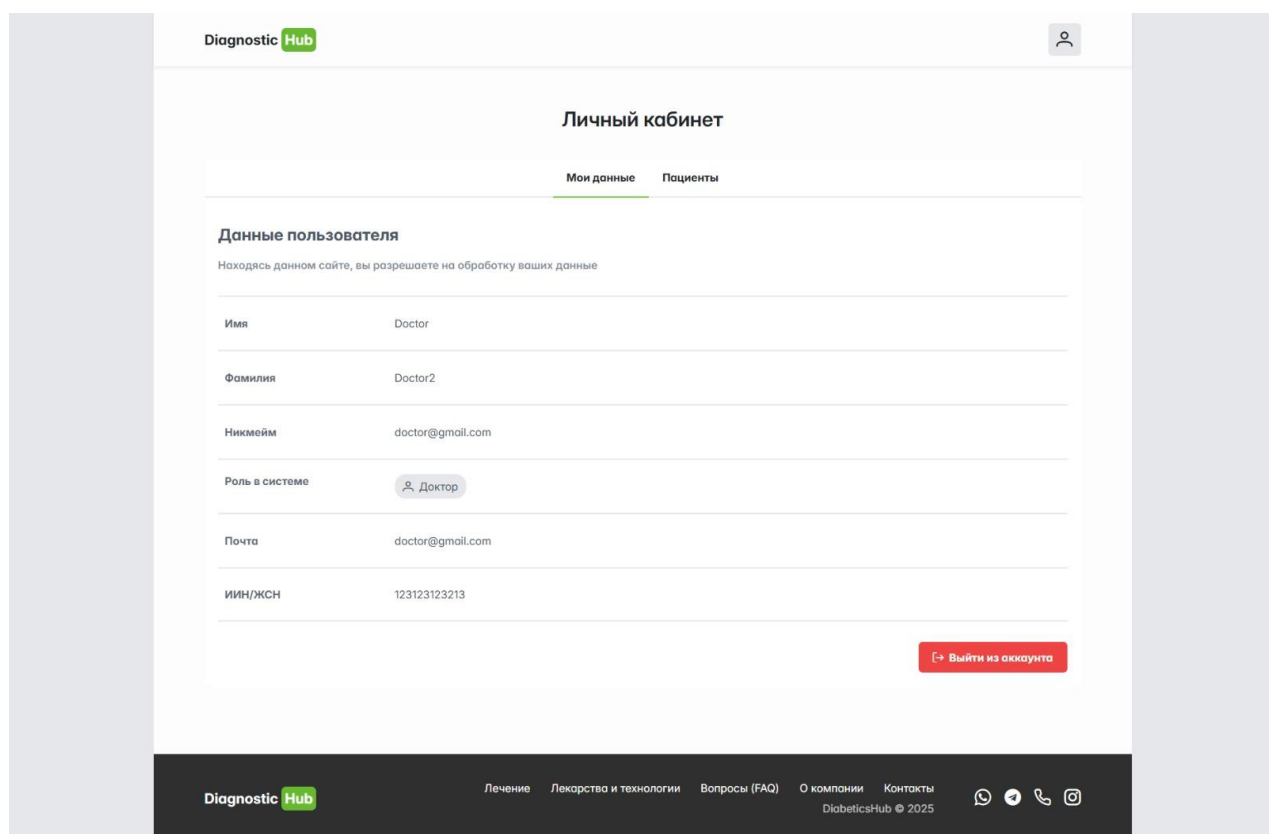
5. Жауап батырмалары және интерактив:

Кейбір беттерде (мысалы, дәрі қабылдау парағында) "Иә / Жоқ" батырмалары арқылы тез жауап беру мүмкіндігі бар. Бұл функция науқастың өзара әрекет етуін жеңілдетеді және уақыт үнемдейді.

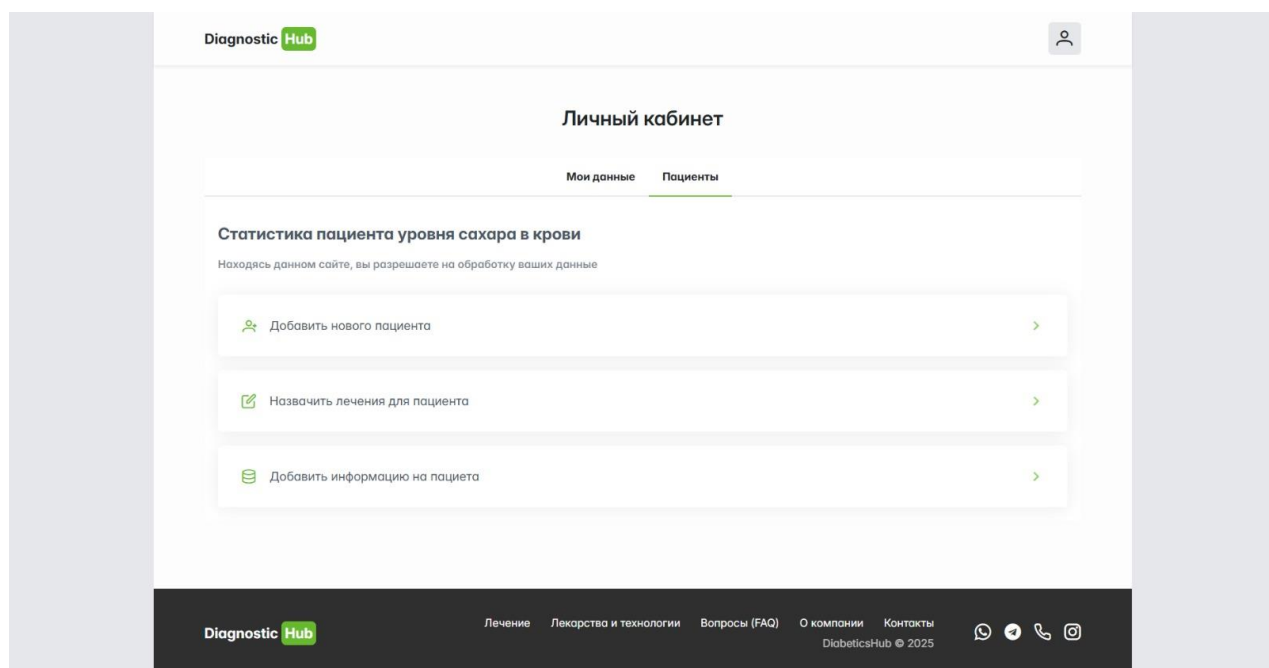
Интерфейстің дизайнының артықшылықтары:

- Түсінікті және таза навигация: барлық әрекеттер бірнеше қадам арқылы орындалады.
- Мобильді құрылғыларға бейімделген: интерфейс смартфон мен планшетте де бірдей ыңғайлы.
- Қаріп өлшемі мен түстер: егде жастағы адамдар үшін қарапайым және анық көрінетін элементтер.
- Қауіпсіздік: деректерді көру және өңдеу тек авторизацияланған қолданушыларға рұқсат етілген.

Дәрігердің жеке кабинеті және науқастың басқару интерфейсі



20 сурет – Дәрігердің жеке бөлімі



21 сурет – Дәрігердің науқасты тіркеуі

Дәрігердің жеке кабинеті және науқастарды басқару интерфейсі

Медициналық ақпараттық жүйеде дәрігер рөлінде тіркелген қолданушыға арналған арнайы жеке кабинет қарастырылған. Бұл кабинет арқылы дәрігер өз деректерін көре алады және науқастармен байланысты әрекеттерді орындай алады. Интерфейс қарапайым, интуитивті түрде ұйымдастырылған және қауіпсіздік талаптарына сәйкес әзірленген.

1) Жеке кабинет – "Менің деректерім" бөлімі (20 сурет):

Бірінші скриншотта көрсетілгендей, "Менің деректерім" беті арқылы дәрігер өз жеке ақпараттарын көреді:

- Аты-жөні
- Никнейм (электрондық пошта)
- Рөлі (жүйеде "Дәрігер" ретінде тіркелген)
- ИИН/ЖСН (жеке сәйкестендіру нөмірі)

Бет төменгі жағында "Аккаунттан шығу" батырмасы орналасқан, бұл ақпараттық қауіпсіздік тұрғысынан маңызды.

Интерфейс қолданушыға өзінің мәліметтерін тексеруге және қажет болған жағдайда жүйеден шығуға мүмкіндік береді. Қарапайым кестелік құрылым деректерді жеңіл қабылдауға жағдай жасайды.

2) Науқастармен жұмыс – "Науқастар" бөлімі (21 сурет):

Екінші скриншотта көрсетілген "Науқастар" беті дәрігердің негізгі жұмыс алаңы болып табылады. Мұнда дәрігер үш негізгі әрекетті орындай алады:

- Жаңа науқасты қосу:

Дәрігер жүйеге жаңа науқас тіркей алады. Бұл әрекет науқастың негізгі деректерін енгізуден басталады (аты, ИИН, туған күні және т.б.).

- Науқасқа ем тағайындау:

Белгілі бір науқасқа арналған емдеу жоспарын енгізу (мысалы, дәрілер, жағдайын білу).

- Науқасқа қатысты ақпарат енгізу:

Дәрігер науқастың денсаулық жағдайына қатысты қосымша мәліметтерді (қан қысымы, глюкоза деңгейі т.б.) жүйеге енгізе алады.

Беттің жоғарғы жағында "Статистика пациента уровня сахара в крови" деген жазба көрсетілген, бұл науқастың негізгі көрсеткіштерін бақылау мүмкіндігін білдіреді.

Интерфейстің ерекшеліктері:

- Түсінікті навигация: "Менің деректерім" және "Науқастар" деген екі негізгі бөлім бар.

- Иконкалар мен батырмалар визуалды тұрғыда түсінікті: қосу, өңдеу, шығу әрекеттері.

- Қауіпсіздік: әрбір әрекет тек авторизациядан өткен қолданушыларға ғана рұқсат етілген.

- Медициналық процестерді сандық форматта жүргізу: қағазсыз тіркеу, қашықтан бақылау және онлайн жоспарлау.

3.1 Интерфейс элементтерінің жобасы

(Қан диабетімен ауыратындар үшін медициналық ақпараттық жүйе мысалында)

Қолданушы интерфейсі – медициналық жүйенің пайдаланушы мен жүйе арасындағы негізгі өзара әрекет алаңы. Қан диабетімен ауыратын науқастар үшін интерфейс элементтері нақты, ыңғайлы және күнделікті қолдануға бейімделіп жобалануы тиіс. Интерфейс құрылымы науқастың денсаулығын үздіксіз бақылап отыруға, дәрігермен байланыста болуға және күнделікті медициналық ақпаратты енгізуге мүмкіндік береді.

Интерфейстің негізгі элементтері:

1. Жоғарғы навигациялық панель:

Барлық беттердің жоғарғы жағында орналасқан бұл панельде жүйенің логотипі, қолданушының профилі (иконка), сондай-ақ "Менің деректерім", "Науқастар", "Статистика" секілді беттерге сілтемелер бар. Бұл пайдаланушының жүйеде еркін жүруін қамтамасыз етеді.

2. Деректер енгізу формалары:

Науқас немесе дәрігер әртүрлі медициналық көрсеткіштерді енгізу үшін арнайы өрістермен жабдықталған формаларды пайдаланады. Мысалы:

- Глюкоза деңгейі (ммоль/л).
- Қан қысымы.
- Диабет әсері (ұпаймен бағалау).
- Дәрі қабылдау уақыттары.

Әр форма логикалық құрылыммен орналасқан және визуалды элементтермен (иконка, түс, шрифт) толықтырылған.

3. Интерактивті батырмалар:

Әрекет жасау үшін түрлі батырмалар қолданылады:

«Деректерді жіберу».

«Жаңа науқас қосу».

«Науқасқа ем тағайындау».

«Аккаунттан шығу».

Батырмалар түсімен және белгішесімен нақты әрекетті бейнелейді, бұл пайдаланушыға шатаспай әрекет етуге көмектеседі.

4. Диаграммалар және кестелер:

Науқастың қант деңгейінің өзгерісін бейнелейтін диаграмма визуалды қабылдауға өте ыңғайлы. Сонымен қатар, енгізілген деректер арнайы кесте арқылы төменгі жағында көрсетіледі. Кестелерде бағандар анық бөлінген:

- Күні.
- Қант деңгейі.
- Диабет ұпайы.
- Қан қысымы.

5. Рөлге бейімделген интерфейс: Әрбір қолданушының интерфейсі оның рөліне қарай бейімделеді:

- Науқас: өзінің денсаулық деректерін енгізіп, статистиканы қарай алады.
- Дәрігер: жаңа науқас қосып, ем тағайындайды, науқастың прогресін бақылайды.
- Ғалым/талдаушы: анонимді деректер негізінде жалпы диаграмма мен есептер шығара алады.

Пайдаланушы ыңғайлылығы үшін:

- Интерфейс қарапайым түстермен (ақ фонда жасыл, көк, қызыл акценттер) жасалған;
- Түсінікті иконкалар әрекетті тез түсінуге көмектеседі;
- Ашық қаріп пен арақашықтық – көзге жайлы;
- Батырмалардың үлкендігі мен түстері егде жастағы адамдар үшін де ыңғайлы.

3.2 Қосымша беттер

Негізгі беттерден бөлек, жүйеде пайдаланушы тәжірибесін жақсартатын және функционалдық мүмкіндіктерді кеңейтетін бірнеше қосымша беттер қарастырылған. Бұл беттер әртүрлі қолданушы топтары (науқас, дәрігер, ғалым) үшін түрлі мақсатта қызмет атқарады.

1) Кері байланыс беті

Бұл бет арқылы қолданушылар жүйеге ұсыныс, пікір немесе шағым қалдыра алады. Қарапайым форма (аты, поштасы, хабарлама) және "Жіберу" батырмасы орналасқан. Кері байланыс жүйе әкімшілігіне жетіп, жүйені жақсарту бойынша жұмыстар жүргізіледі.

2) Құпия сөзді қалпына келтіру беті

Қолданушы логин немесе құпия сөзін ұмытып қалған жағдайда, жүйеге кіру беті арқылы "Құпия сөзді ұмыттыңыз ба?" сілтемесін басып, жаңа құпия сөз орната алады. Бұл бет SMS/email арқылы код жіберу функционалын қамтиды.

3) Хабарламалар және ескертулер беті

Қолданушыға дәрі қабылдау уақыты, дәрігер тағайындаған емдеу жоспары немесе жаңа хабарламалар туралы ескертулер жіберіледі. Бұл бетте өткен ескертулердің тізімі сақталып отырады (мысалы, “Сіз бүгін 14:00-де дәрі қабылдадыңыз ба?”).

4) Пайдаланушы профилін өңдеу беті

Қолданушы өз аты-жөнін, поштасын, туған күнін, рөлін өзгерте алады. Интерфейс қарапайым және деректерді өзгерткен соң "Сақтау" батырмасын басу арқылы жаңартуға мүмкіндік береді.

5) Көмек / Жиі қойылатын сұрақтар (FAQ)

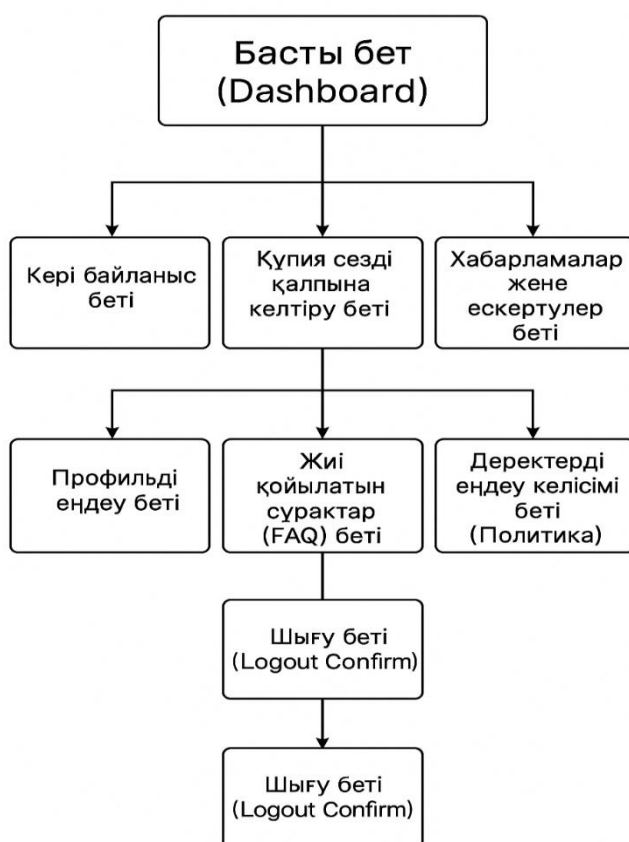
Қолданушылар жүйе жұмысын түсінбей қалған жағдайда, арнайы сұрақ-жауап бетіне кіріп, "Деректер қалай енгізіледі?", "Дәрігерді қалай табуға болады?" сияқты жиі қойылатын сұрақтарға жауап таба алады.

6) Құқықтық ақпарат және деректерді өңдеу келісімі

Бұл бетте пайдаланушының деректері қалай қорғалатыны, қайда және қалай қолданылатыны жөнінде толық ақпарат беріледі. Бұл GDPR немесе ұлттық заңнамаларға сәйкес құрастырылған.

7) Жүйеден шығу беті (logout confirmation)

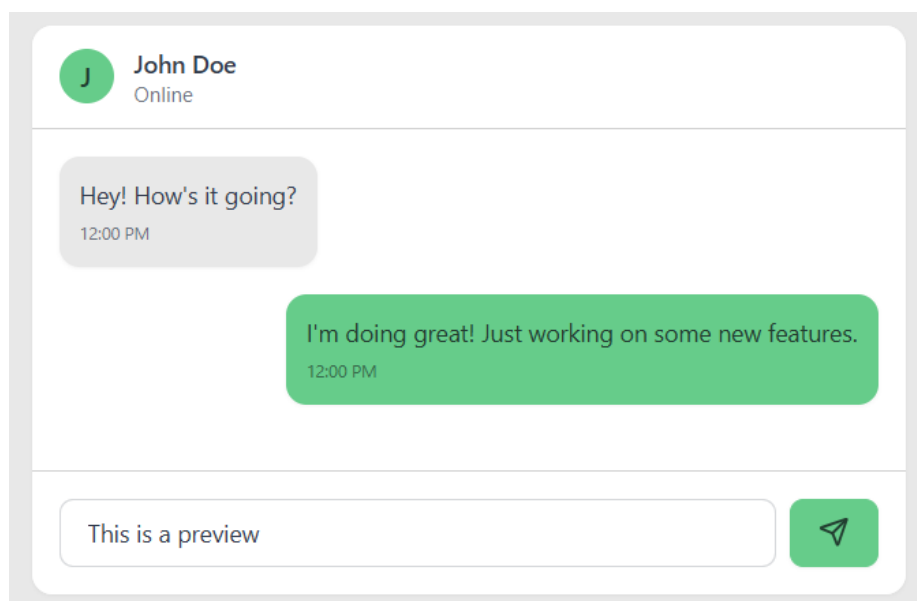
Пайдаланушы жүйеден шығу батырмасын басқанда, "Шынымен де шығасыз ба?" деген растау беті көрсетіледі. Бұл кездейсоқ басуды болдырмауға арналған.



22 сурет – Қосымша беттер арасындағы өзара байланыс диаграммасы

Бұл диаграмма қосымша беттердің басты бетпен байланысын және өзара әрекет ету ретін көрсетеді. Интерфейс логикасы қолданушы әрекетіне қарай бағытталып, әр бөлім нақты функцияны атқарады (мысалы, деректерді өңдеу, ескертулерді көру, көмек алу немесе жүйеден шығу).

3.3 Чат



3.3 сурет – Дәрігер мен науқас арасындағы чат интерфейсі

Бұл интерфейс науқас пен дәрігер арасында жедел және қауіпсіз байланыс орнатуға арналған. Хабарламалар уақытпен белгіленіп, пайдаланушы рөліне сәйкес бағытталады.

Дәрігер мен науқас арасындағы чат модулі

Қан диабеті секілді созылмалы ауруларды емдеу кезінде дәрігер мен науқастың тұрақты байланыста болуы өте маңызды. Осы мақсатта жүйеде арнайы жеке чат модулі енгізілген. Бұл модуль арқылы дәрігер мен науқас арасындағы тікелей және жедел байланыс қамтамасыз етіледі.

Интерфейс сипаттамасы (суретпен байланыс)

Сіз жіберген суретте көрсетілгендей, чат екі жақты сөйлесу терезесі түрінде ұйымдастырылған:

- Сұр фондағы хабарлама – басқа қатысушының (мысалы, науқастың) хабары.
- Жасыл фондағы хабарлама – пайдаланушының (мысалы, дәрігердің) жауабы.
- Әр хабарламаның төменгі жағында уақыты көрсетілген.
- Төменгі бөлікте жаңа хабарлама теріп жіберуге арналған мәтін жолағы және жіберу батырмасы (қағаз ұшақ белгісі) орналасқан.

Мұндай дизайн чат интерфейсін жеңіл, заманауи әрі пайдаланушыға ыңғайлы етеді.

Рөлге негізделген байланыс

Чат жүйесі рөлдерге байланысты шектелген:

- Дәрігер:
 - Тек өзі бақылайтын науқастармен ғана байланыса алады.
 - Әр науқасқа жеке-жеке хабарлама жібере алады.
 - Хабарламалар науқастың жеке парағына тіркеліп, қажет жағдайда емдеу тарихымен бірге қарастырылады.

- Науқас:
 - Тек өзін бақылайтын дәрігерді ғана көре алады.
 - Басқа науқастармен немесе дәрігерлермен байланысқа шыға алмайды.
 - Дәрігермен жеке сөйлесу мүмкіндігі арқылы кез келген уақытта сұрақ қоя алады (дәрі, тамақтану, глюкоза деңгейі т.б.).

Қауіпсіздік және құпиялық:

- Барлық чат хабарламалары шифрланып сақталады.
- Деректер үшінші тарапқа берілмейді.
- Әр хабарлама Firestore немесе басқа базада уақыт таңбасымен бірге сақталады.

Қосымша мүмкіндіктер:

- Дәрігерлер чат арқылы науқасқа видео, тапсырма, дәрі қабылдау еске салғышын жібере алады.
- Науқас сұраққа жауап алмай қалса, жүйе автоматты түрде дәрігерге ескерту жібереді.
- Алдағы уақытта бейнеқоңырау мүмкіндігі қосу жоспарлануда.

3.4 Администратор панелі

Медициналық ақпараттық жүйеде администратор рөлі – ең жоғары басқару құқығына ие, барлық жүйелік процестерді бақылайтын және қадағалайтын орталық фигура болып табылады. Админ панелі жүйенің қауіпсіздігі, тұрақтылығы мен деректердің тұтастығы үшін жауап береді.

Жалпы сипаттама:

Администратор панелі – тек бір тұлғаға (жүйелік әкімшіге) арналған интерфейс. Бұл панель арқылы әкімші веб-сайттың барлық модульдеріне шолу жасап, барлық қолданушылар мен олардың әрекеттерін басқара алады. Бұл бөлім жүйенің жоғары деңгейдегі басқару орталығы ретінде қызмет етеді.

Администратордың негізгі мүмкіндіктері:

- i. Қолданушыларды басқару:
 1. Барлық рөлдердегі қолданушыларды (науқас, дәрігер, ғалым, т.б.) көру
 2. Қолданушы тіркеуін растау немесе бұғаттау
 3. Қолданушы рөлін өзгерту (мысалы, дәрігерден ғалымға ауыстыру)
 4. Жаңа қолданушыны жүйеге қолмен қосу немесе жою
- ii. Деректерге толық қолжетімділік:
 1. Барлық науқастардың медициналық мәліметтерін қарау

2. Дәрігерлердің тағайындаған емдеу жоспарын, чатты, видеосабақтарды көру
 3. Қан құрамының, глюкоза деңгейінің, қысым көрсеткіштерінің статистикасын талдау
 4. Жүйе базасына енгізілген барлық тарихи деректерді көру және өңдеу
- iii. Контентті және модульдерді басқару:
 1. Веб-сайттағы барлық беттерді, модульдерді, компоненттерді өңдеу (мысалы, FAQ беті, басты бет контенті)
 - iv. Жүйе қауіпсіздігін бақылау:
 1. Кіру әрекеттерінің журналына (login history) қол жеткізу
 2. Қолданушылардың IP-мекенжайларын және құрылғыларын талдау
 3. Шифрлау алгоритмдерін бақылау және аутентификация параметрлерін баптау
 - v. Статистика және есептер:
 1. Жүйеге кірушілер саны, белсенді қолданушылар, емделу прогресі және басқа метрикаларды визуалды түрде бақылау
 2. Қолданушылардың белсенділік картасын көру
 3. Қателер мен жүйелік журналдарды (logs) қарап отыру

Интерфейс сипаттамасы:

Админ панелі визуалды түрде заманауи, құрылымы жағынан логикалық түрде бөлінген:

- Басқару тақтасы (Dashboard): жалпы статистика мен күй.
- Қолданушылар бөлімі: барлық тіркелген қолданушылар тізімі.
- Модульдер: емдеу жоспарлары, видеосабақтар, хабарламалар.
- Қауіпсіздік: жүйеге кіру журналдары мен құқықтарды бақылау.

Маңызды: Жүйеде тек бір ғана администратор болады. Қауіпсіздік мақсатында бұл рөл ешкімге тағайындалмайды және басқа рөлге ауыстыруға болмайды. Админ өзі ғана өзгеріс енгізе алады.

Қосымша мүмкіндіктер:

- Жүйені резервтік көшіру және қалпына келтіру (backup & restore)
- Хабарландыру жүйесін басқару (кімге және қашан не жіберілетінін)
- Жаңа модульдерді жүктеу, орнату немесе өшіру

4 Бағдарламаны жасау және тестілеу

Бұл бөлімде медициналық ақпараттық жүйені әзірлеу барысы мен оның сенімділігін тексеру үшін жүргізілген тестілеу әрекеттері сипатталады. Жоба диабетпен ауыратын науқастар мен дәрігерлер арасындағы өзара байланысты жеңілдетуге, науқастың күнделікті көрсеткіштерін тіркеуге және бақылауға арналған. Жүйеде сондай-ақ чат, видеосабактар, дәрі қабылдауды еске салғыш және рөлдік басқару функционалы қарастырылған.

Жүйені жасау кезеңдері

Жүйе үш негізгі бағытта құрылды:

- науқастың деректерін күнделікті тіркеп отыру мүмкіндігі;
- дәрігердің науқастарды бақылауы және ем тағайындауы;
- администратордың жалпы жүйені басқару мүмкіндігі.

Жоба келесі кезеңдерден өтті:

Талдау және жобалау:

Пайдаланушы рөлдері (науқас, дәрігер, админ, ғалым) анықталып, функционалдық талаптар негізінде жүйе құрылымы жобаланды. Қолданушы интерфейстері сызылып (mockup), UX қағидалары ескерілді.

Модульдерге бөлу:

Әр рөл үшін жеке функционал әзірленді. Науқас - дерек енгізеді, дәрігер - оқиды, админ - басқарады. Чат, статистика, дәрі еске салғыш және видеосабак модульдері бөлектеліп жасалды.

Архитектура:

Жүйе клиент-серверлік принциппен (REST) құрастырылды. Фронтенд пен бэкенд модульдері бір-бірінен бөлінген, мәліметтер базасы — бұлттық негізде.

Қолданылған технологиялар мен құралдар:

2 кесте – Қолданылған технологиялар мен құралдар

Құрал/Технология	Мақсаты
Java + Spring Boot	Бэкенд логика мен API-ларды құру
Firebase (Firestore, Auth)	Деректерді сақтау, авторизация
Android (Java + View Binding)	Мобильді интерфейс
Firebase Cloud Messaging	Хабарлама және еске салғыш жіберу
JUnit, Postman	API мен логиканы тестілеу
Firebase Emulator Suite	Жергілікті тестілеу үшін
Git + GitHub	Кодты басқару, командалық жұмыс

Тестілеу стратегиясы:

Жүйенің сенімділігі мен дұрыстығын тексеру үшін бірнеше деңгейдегі тестілеулер жүргізілді:

- Модульдік тестілеу (Unit Test):

Әр жеке функция (мысалы, дәрі қабылдау уақыты тексерісі, еске салғышты белсендіру) JUnit арқылы тестіленді.

- Интеграциялық тестілеу:

Мысалы, науқас дәріні қабылдағанын белгілейді → дерек базаға түседі → дәрігер өз панелінен бұл ақпаратты көреді → Success.

- UI тестілеу (қолмен және автоматты):

React интерфейсі бірнеше құрылғыда тексерілді (элементтер дұрыс көрінуі, енгізу өрістерінің жұмыс істеуі, батырмалар).

- Рөлдермен тестілеу:

Науқас басқа науқасқа жаза алмайды, дәрігер тек өз науқасына көреді, админ бәрін көреді — осы құқық шектеулері толық тексерілді.

- Қауіпсіздік тестілеуі:

Авторизациясыз кіру әрекеттері, токендермен қорғау және деректерге бөгде қолжетімділікті шектеу тексерілді.

4.1 Бағдарламалық жасақтама компоненттері

Медициналық ақпараттық жүйе бірнеше өзара байланысты бағдарламалық компоненттерден тұрады. Әр компонент жүйенің белгілі бір бөлігін қамтамасыз етіп, нақты функционал үшін жауап береді. Бұл модульдік құрылым жүйенің икемділігін, кеңейтілу мүмкіндігін және техникалық қызмет көрсетуін жеңілдетеді.

Жалпы архитектура

Жүйе клиент-сервер архитектурасына негізделген. Қолданушылар (науқастармен мен дәрігерлер) веб немесе мобильді қосымша арқылы әрекет етеді, ал серверлік бөлік барлық логиканы, деректермен жұмыс істеуді және қауіпсіздікті қамтамасыз етеді.

Компоненттер үш негізгі қабатқа бөлінеді:

- Клиенттік қабат (UI)
- Қолданбалы логика қабаты (Backend/API)
- Деректер қоры қабаты (Database)

Компоненттердің сипаттамасы

1) Клиенттік қабат (Front-end компоненті):

Технологиялар: React js, js, axios, Firebase UI,
Міндеті:

- Қолданушыға визуалды интерфейс ұсыну

- Мәліметтерді енгізу, оқу, дәрі қабылдау белгілеу
- Чат, хабарландыру, статистика диаграммаларын көрсету
- Firebase API-мен байланыс орнату

2) Серверлік қабат (Back-end компоненті):

Технологиялар: Java, Spring Boot, Firebase Admin SDK

Міндеті:

- REST API арқылы клиентпен байланыс
- Рөлдік қолжетімділік басқару (админ, дәрігер, науқас, ғалым)
- Жазбаларды тексеру және сақтау
- Емдеу жоспарларын өңдеу
- Статистика және хабарландыру логикасы

3) Деректер қоры қабаты:

Технологиялар: Firebase Firestore (NoSQL), Firebase Storage.

Міндеті:

• Қолданушы деректері, медициналық жазбалар, чат хабарламаларын жасау.

• Құрылым:

- Users
- Patients
- Doctors
- Medicines
- Chats

4) Хабарлама және еске салғыш компоненті:

Технологиялар: Firebase Cloud Messaging (FCM), Email API

Міндеті:

- Науқасқа дәрі қабылдау туралы еске салғыш жіберу
- Email арқылы растау немесе нотификация жеткізу
- Админге маңызды жүйелік оқиғалар жөнінде хабарлау

5) Администратор панелі компоненті:

Технологиялар: Web-based (Spring Boot + Thymeleaf немесе Firebase Console).

Міндеті:

- Барлық қолданушылар мен жазбаларға толық қолжетімділік
- Контентті басқару, қолданушыны бұғаттау
- Деректерді өңдеу және жүйе логтарын қарау

6) Чат компоненті:

Технологиялар: Firestore Realtime updates, Custom UI.

Міндеті:

- Науқас пен дәрігер арасындағы тікелей байланыс
- Хабарламаларды нақты уақытта көрсету
- Әр хабарлама уақыты мен рөлге қарай бөлініп сақталады

Компоненттердің өзара байланысы

Барлық компоненттер REST API немесе Firebase арқылы біріктірілген. Пайдаланушы әрекеті (мысалы, дәрі қабылдауды белгілеу) Firestore-ға жазылады, содан кейін басқа компонент (дәрігер панелі, статистика диаграммасы) автоматты түрде жаңарып отырады.

4.2 Тестілеу

Қан диабетіне арналған медициналық ақпараттық жүйе — қолданушы деректері мен денсаулық көрсеткіштерімен тікелей жұмыс істейтін жүйе болғандықтан, оның әрбір функционалды бөлігі мұқият және жан-жақты тестілеуден өтті. Тестілеу кезеңі жүйенің сенімділігін, дұрыстығын және қауіпсіздігін қамтамасыз ету мақсатында жүргізілді.

Тестілеу мақсаты:

- Жүйе функционалының дұрыс жұмыс істейтінін тексеру
- Қолданушы әрекеттеріне жүйенің сәйкес жауап беруін бақылау
- Қауіпсіздік талаптарының орындалуын қадағалау
- Жүйе өнімділігі мен деректер тұтастығын тексеру

Тестілеу әдістері:

Жобада келесі тестілеу түрлері қолданылды:

1) Модульдік тестілеу (Unit Testing):

- Әр функция немесе класс бөлек тексерілді
- Мысалы:
 - Науқастың дерек енгізу логикасы (submitVitalsData())
 - Уақыт тексерісі: дәрі қабылдау уақыты өткен бе?
 - Рөлдік шектеу: қолданушы тек өз рөліне тән функцияға ғана кіре ала

ма?

Құралдар: JUnit, Mockito

2) Интеграциялық тестілеу (Integration Testing):

- Компоненттердің бірлескен жұмысын тексеру:
 - Науқас дәрі қабылдағанын белгілейді → базаға түседі → дәрігер панелінде көрсетіледі.

Құралдар: Postman, Firebase Emulator, API Tester

3) UI тестілеу (Interface Testing):

- Android құрылғысында интерфейстің толық тексерісі
- Батырмалар жұмыс істей ме, формалар дұрыс ашыла ма, хабарлама

шыға ма

- Экрандардың әртүрлі өлшемінде көрінісі

Құралдар: React js, js (қажет болса)

4) Қауіпсіздік тестілеуі (Security Testing):

- Авторизациясыз кіру әрекеттері: жүйе кіргізбей ме?
- Науқас басқа біреудің дерегін көре алмай ма?
- Токенмен қорғау жұмыс істей ме?

Қорытынды

Осы дипломдық жобаның аясында қант диабетімен ауыратын Науқастарға арналған медициналық ақпараттық жүйе әзірленді. Жоба заманауи бағдарламалық технологиялар мен архитектуралық үлгілерді қолдана отырып жүзеге асырылды және денсаулық сақтау саласының өзекті мәселелерін шешуге бағытталды.

Жасалған жүйенің басты мақсаты - науқастардың медициналық жағдайын үздіксіз бақылауға мүмкіндік беру, дәрігермен тиімді байланыс орнату және емдеу сапасын арттыру. Бұл мақсатқа жету үшін жоба аясында бірнеше негізгі модульдер мен компоненттер жасалды: науқастың деректерін тіркеу беті, дәрі қабылдауды бақылау жүйесі, қандағы глюкоза деңгейін күнделікті енгізу модулі, дәрігермен чат арқылы байланыс және видеосабақтармен жұмыс істеу мүмкіндігі.

Жүйе Spring Boot фреймворкі және Java тілі арқылы әзірленіп, деректер Firebase Firestore және PostgreSQL базаларында сақталады. Фронтенд бөлігі Android платформасында View Binding және Firebase UI көмегімен құрылды. Бұл құрылым жүйенің икемділігі мен қауіпсіздігін қамтамасыз етті. Сонымен қатар, жүйеде нақты рөлдер бөлінісі енгізілді: науқас, дәрігер, ғалым және жалғыз администратор. Әр рөл өзіне тиесілі интерфейс пен функционалға ие.

Жоба барысында жүргізілген тестілеу нәтижелері жүйенің тұрақты, сенімді және қауіпсіз екенін дәлелдеді. Барлық функционалдар модульдік және интеграциялық деңгейде тексерілді. Қолданушы интерфейсі қарапайым әрі кез келген жастағы қолданушыға түсінікті етіп жасалған. Жүйе тек техникалық тұрғыдан ғана емес, сонымен қатар әлеуметтік жағынан да тиімді - ол дәрігер мен науқас арасындағы байланысты нығайтып, науқастың денсаулығын қашықтан бақылауға мүмкіндік береді.

Жобаның ерекше артықшылығы – оның бейімделгіштігі, геймификация элементтері және науқасқа жекелендірілген тәжірибе ұсыну қабілеті. Алдағы уақытта бұл жүйеге қосымша мүмкіндіктер қосу және оны басқа созылмалы ауруларды бақылауға бейімдеу арқылы кеңейту жоспарлануда.

Қорыта келе, бұл дипломдық жоба – Қазақстандағы денсаулық сақтау саласын цифрландыруға үлес қосатын, ғылыми негізделген, әлеуметтік маңызы жоғары, әрі толыққанды жұмыс істейтін заманауи медициналық платформа болып табылады.

Пайдаланылған әдебиеттер тізімі

1. Жақсыбаев Қ.Қ., Омашев Б.М. Программалау негіздері. – Алматы: Дәуір, 2020.
2. Төлегенов Е.Т. Ақпараттық жүйелерге кіріспе. – Алматы: Эверо, 2019.
3. Гусев В.Е., Корольков Д.В. Информационные системы в медицине. – Москва: Инфра-М, 2018.
4. Жақыпбаев Ә., Мырзабеков Қ. Медициналық ақпараттық жүйелердің негіздері. – Алматы: Қазақ университеті, 2021.

2. Ресми стандарттар мен нормативтік құжаттар:

5. ҚР Денсаулық сақтау министрлігі. Денсаулық сақтауды цифрландыру тұжырымдамасы 2020–2025 ж.
6. ISO/IEC 25010:2011. System and software quality models.
7. GDPR. General Data Protection Regulation. <https://gdpr.eu/>

3. Веб-ресурстар және технологиялық құжаттама:

8. Spring Boot Reference Documentation — <https://spring.io/projects/spring-boot>
9. React JS Documentation — <https://reactjs.org/docs>
10. Docker Docs — <https://docs.docker.com>
11. PostgreSQL Official Docs — <https://www.postgresql.org/docs/>
12. GitLab CI/CD Docs — <https://docs.gitlab.com/ee/ci/>

4. Зерттеулер мен ғылыми мақалалар:

14. Alotaibi Y., Federico F. The impact of health information technology on patient safety. – Saudi Medical Journal, 2017.
15. Kudyba S. Healthcare Informatics: Improving Efficiency and Productivity. – CRC Press, 2020.
16. Мусаева А.Б. Цифровизация в здравоохранении: вызовы и перспективы. – Информационные технологии в медицине, 2021.

Қосымша А

Техникалық тапсырма

Медициналық деректерді бақылау және инвазивті емес құралды басқару үшін Web қосымша жасау (backend / front-end).

Қант диабеті бар науқастарды бақылауға арналған

1. Негізгі талаптар:

1.1 Жалпы талаптар:

- Кросс-платформалық жүйе (веб нұсқалары)
- Жоғары сенімділік және деректердің қауіпсіздігі (шифрлау, екі факторлы аутентификация)
- Қауіп-қатерлер мен асқынуларды болжау мүмкіндігі
- Зертханалық деректерді дәрігер қолмен енгізеді

1.2 Функционалдық талаптар

1.2.1 Науқастар модулі:

- Тіркелу, жүйеге кіру
- Жеке кабинет: профиль, өлшемдер тарихы
- Қандағы глюкоза деңгейін бақылау (қолмен енгізу)
- Тамақтану мен дене белсенділігі күнделігі
- Дәрі-дәрмек қабылдау және процедуралар туралы еске салғыштар
- Глюкоза деңгейінің графиктері мен талдауы
- Дәрігермен чат, телемедицина
- Дәрігерге арналған есеп генерациясы

1.2.2 Дәрігер модулі:

- Науқастың барлық деректеріне қолжетімділік
- Ауру тарихын қарау
- Трендтерді талдау және ұсыныстар беру
- Инсулин дозасын түзету бойынша автоматты ұсыныстар
- Ем тағайындау және бақылау
- Зертханалық деректерді қолмен енгізу және қарау
- Науқастармен чат
- Келесі қабылдауды жоспарлау

1.2.3 Ғалымдар модулі:

- Анонимді науқастар базасына қолжетімділік
- Деректерді талдау құралдары (статистика, болжам жасау)

1.2.4 Ғалымдар үшін функцияларды кеңейту:

- Жас, жыныс, диабет түрі бойынша іріктеу параметрлерін икемді орнату
- Глюкоза динамикасын визуализациялау құралы

А қосымшасының жалғасы

4. Пайдаланушы тәжірибесін жақсарту:

- Офлайн режимі бар мобильді қосымша
- Дәрі-дәрмек қабылдау туралы дауыстық немесе мәтіндік хабарламалар
- Науқастарды ынталандыру үшін геймификация (тұрақты өлшемдер

үшін марапаттар)

Дәрігерге арналған анкета: Науқас туралы мәліметтерді енгізу

1. Науқас туралы жалпы ақпарат

- Науқастың ID: (автоматты түрде беріледі)
- Науқастың толық аты-жөні: (анонимді жағдайда тек ID)
- Туған күні: □□/□□/□□□□
- Жынысы: ☐ Ер ☐ Әйел
- Байланыс нөмірі: _____
- Диабет диагнозы қойылған күн: □□/□□/□□□□
- Диабет түрі:

☐ 1-тип ☐ 2-тип ☐ Гестациялық ☐ Басқа: _____

2. Ауру тарихы және шағымдары

- Соңғы қабылдау күні: □□/□□/□□□□
- Науқастың негізгі шағымдары:

- ☐ Гипогликемия
- ☐ Гипергликемия
- ☐ Шаршағыштық
- ☐ Аяқ-қолдардың ұйып қалуы
 - ☐ Салмақтың өзгеруі
 - ☐ Басқа: _____
- Қосымша аурулар:
- ☐ Қан қысымы жоғары
- ☐ Семіздік
- ☐ Бүйрек аурулары
- ☐ Жүрек-қан тамырлары аурулары
- ☐ Басқа: _____
 - Аллергиялар:
- ☐ Жоқ
- ☐ Иә (көрсету): _____

3. Физикалық тексеріс

А қосымшасының жалғасы

- Бойы (см):
- Салмағы (кг):
- Дене массасының индексі (ИМТ): (автоматты есептеледі)
- Қан қысымы (мм сын.бағ.): /
- Жүрек соғу жиілігі (соғ./мин):
- Дене қызуы (°C): ,

4. Зертханалық талдаулар (қолмен енгізіледі)

- Ашқарындағы глюкоза (ммоль/л): ,
- HbA1c (%): ,
- Липидтік профиль:
 - Жалпы холестерин (ммоль/л): ,
 - Төмен тығыздықтағы липопротеиндер (ЛПНП): ,
 - Жоғары тығыздықтағы липопротеиндер (ЛПВП): ,
 - Триглицеридтер (ммоль/л): ,
- Креатинин (мкмоль/л):
- Альбуминурия (мг/л):

5. Ем және ұсыныстар

- Тағайындалған терапия:
 - ☐ Инсулин (түрі, мөлшері): _____
 - ☐ Ауызша дәрілер (метформин, т.б.): _____
 - ☐ Диетотерапия
 - ☐ Физикалық белсенділік (сипаттама): _____
- Машинамен оқыту ұсыныстары негізінде дозаны түзету:
 - ☐ Иә (жүйе автоматты түрде ұсынады)
 - ☐ Жоқ (дәрігер өзі тағайындайды)
- Науқасқа еске салғыштар:
 - ☐ Глюкозаны үнемі өлшеу
 - ☐ Эндокринологқа бару
 - ☐ Салмақты бақылау
 - ☐ Басқа: _____

6. Келесі қабылдауды жоспарлау

А қосымшасының жалғасы

- Келесі қабылдау күні: □□/□□/□□□□
- Себебі:
- ☐ Жай-күйді бақылау
- ☐ Емдеу жоспарын түзету
- ☐ Талдау тапсыру
- ☐ Басқа: _____

Қосымша Ә

Пайдаланушы нұсқаулығы

Бұл пайдаланушы нұсқаулығы – медициналық ақпараттық жүйені (МАЖ) тиімді және қауіпсіз пайдалануға арналған толыққанды басшылық. Жүйе қант диабетімен ауыратын Науқастар мен олардың дәрігерлеріне арналған. Пайдаланушыға арналған бұл бөлім қосымшаның барлық негізгі мүмкіндіктері мен интерфейс элементтерін қалай қолдану керектігін егжей-тегжейлі түсіндіреді.

1. Қолданбаны орнату және жүйеге кіру

- Қосымшаны Android құрылғыңызға орнату үшін Google Play Store дүкеніне кіріп, “RehabCare” немесе “Diabetes MIS” деп іздеңіз.

- Орнатқан соң, қолданбаны ашып, екі мүмкіндік ұсынылады:

☐ Тіркелу (жаңа пайдаланушы)

☐ Кіру (бар қолданушы)

- Тіркелу кезінде келесі деректер қажет:

- Аты-жөні

- Туған күні

- Байланыс нөмірі

- Диабет түрі

- Құпия сөз

- Рөлі (Науқаснемесе дәрігер)

2. Науқас үшін нұсқаулық

2.1 Жеке кабинет:

- Пайдаланушы жүйеге кірген соң, басты бетте өзінің жеке көрсеткіштерін көреді.

- "Менің деректерім" бөлімінде: аты-жөні, диагноз қойылған күні, диабет түрі, салмағы, қан қысымы және глюкоза деңгейі сақталады.

2.2 Деректер енгізу:

- Науқас күнделікті қан құрамындағы глюкоза деңгейін, қан қысымын және дене салмағын енгізе алады.

- "Өз деректерімді жіберу" батырмасы арқылы ақпарат серверге тіркеледі.

2.3 Дәрі қабылдау еске салғышы

- Белгіленген уақытта қосымша еске салады:

"Сіз бүгін сағат 14:00-де дәрі қабылдадыңыз ба?"

- Пайдаланушы **Иә** / **Жоқ** деп жауап береді.

Ә қосымшасының жалғасы

2.5 Чат

- Науқас тек өзінің дәрігерімен сөйлесе алады.
- Жазылған хабарламалар нақты уақыт режимінде көрсетіледі.

3. Дәрігер үшін нұсқаулық

- Дәрігер жүйеге кірген соң, барлық тіркелген Науқастарды көре алады.
- Әр Науқастың:

- ☐ Медициналық тарихы
- ☐ Соңғы көрсеткіштері
- ☐ Зертханалық нәтижелері
- ☐ Видеосабактарға қатысуы көрсетіледі.

3.1 Жаңа Науқас қосу

- " Науқас қосу" батырмасы арқылы дәрігер жаңа Науқас тіркей алады.

3.2 Емдеу тағайындау

- Инсулин дозасы, диета және физикалық белсенділік бойынша тағайындаулар енгізіледі.

- Жүйе ML алгоритм арқылы автоматты ұсыныстар береді.

3.3 Чат және жоспарлау

- Науқаспен тікелей байланыс орната алады.
- Келесі қабылдау күнін белгілеп, жүйеге енгізеді.

5. Қауіпсіздік және қолдау

- Барлық деректер Firebase арқылы қорғалады.
- Әр қолданушы тек өз рөліне сай мүмкіндіктерге ғана қол жеткізе алады.
- Кіру және деректерді сақтау шифрланған түрде жүзеге асырылады.
- Қолдау қызметіне хабарласу үшін «Кері байланыс» беті бар.

Бұл пайдаланушы нұсқаулығы қосымшаның негізгі мүмкіндіктерін толық пайдаланып, оны тиімді әрі қауіпсіз қолдануға көмектеседі.

Қосымша Б

Бағдарлама коды

Басты бет:

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import { BrowserRouter, Route, Routes } from "react-router";
import { Provider as ReduxProvider } from "react-redux";
import { QueryClient, QueryClientProvider } from "@tanstack/react-query";

import "@master/css";
import "@master/keyframes.css";

import "primeicons/primeicons.css";

import "./assets/scss/index.scss";

import { store } from "./store";

import { ROUTING } from "./constants/routing.ts";

import { AppLayout } from "./layouts/AppLayout.tsx";
import { AuthLayout } from "./layouts/AuthLayout.tsx";
import { DashboardLayout } from "./layouts/DashboardLayout.tsx";

import { IndexPage } from "./pages/IndexPage.tsx";
import { AuthLoginPage } from "./pages/auth/AuthLoginPage.tsx";
import { DashboardPage } from "./pages/dashboard/DashboardPage.tsx";
import { AuthRegisterPage } from "./pages/auth/AuthRegisterPage.tsx";
import { CreatePatientAccountPage } from "./pages/dashboard/doctor/CreatePatientAccountPage.tsx";
import { CreatePatientInfoPage } from "./pages/dashboard/doctor/CreatePatientInfoPage.tsx";
import { CreateMedicalHistoryAndComplaints } from "./pages/dashboard/doctor/CreateMedicalHistoryAndComplaints.tsx";

const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      refetchOnWindowFocus: false,
    },
  },
});
```

Б қосымшасының жалғасы

```
    },
  });

createRoot(document.getElementById("root")!).render(
  <StrictMode>
    <ReduxProvider store={store}>
      <QueryClientProvider client={queryClient}>
        <BrowserRouter>
          <Routes>
            <Route element={<AppLayout />}>
              <Route
                path={ROUTING.INDEX}
                index={true}
                element={<IndexPage />}
              />

              <Route element={<AuthLayout />}>
                <Route
                  path={ROUTING.AUTH.LOGIN}
                  element={<AuthLoginPage />}
                />

                <Route
                  path={ROUTING.AUTH.REGISTER}
                  element={<AuthRegisterPage />}
                />
              </Route>

              <Route element={<DashboardLayout />}>
                <Route
                  path={ROUTING.DASHBOARD.INDEX}
                  element={<DashboardPage />}
                />

                <Route>
                  <Route
                    path={ROUTING.DASHBOARD.DOCTOR.CREATE_PATIENT}
                    element={<CreatePatientAccountPage />}
                  />
                  <Route
```

Б қосымшасының жалғасы

```
path={ROUTING.DASHBOARD.DOCTOR.CREATE_PATIENT_INFO}
  element={<CreatePatientInfoPage />}
/>

<Route
  path={
    ROUTING.DASHBOARD.DOCTOR
    .CREATE_MEDICAL_HISTORY_AND_COMPLAINTS
  }
  element={<CreateMedicalHistoryAndComplaints />}
/>
</Route>
</Route>
</Route>
</Routes>
</BrowserRouter>
</QueryClientProvider>
</ReduxProvider>
</StrictMode>,
);
```