Bilkent University
Department of Computer Engineering

Senior Design Project

T2312

MediXAI

Detailed Design Report

Project Team Member Names:

Ahmet Arda Ceylan	(22003148)
Mahmut Furkan Gön	(21902582)
Mustafa Yetgin	(21903191)
Omar Hamdache	(22001398)
Ömer Kağan Danacı	(21902584)

Advisor: Selim Aksoy

Course Instructors: Atakan Erdem, Mert Bıçakçı

03/14/2024

Contents

1 Introduction

1.1 Purpose of the System

We have identified challenges in the current healthcare system that lead to difficulties for individuals and institutions involved, including high costs, time-consuming processes, and significant error rates. The growing population and diversity of diseases further complicate the ability of the current healthcare systems to meet the demand. Additionally, inadequate working conditions for doctors have led us to develop a project to make improvements in this field.

Technological advancements, such as robotic applications and machine learning techniques, have inevitably started to automate the healthcare system due to their advantages. We are focusing on this transition process and aim to propose a transition model developed with novel technologies.

In our project, we have chosen to focus on breast cancer. Breast cancer accounts for 12.5% of all new annual cancer cases worldwide, making it the most common cancer in the world [1]. It is more common among women. One out of every eight women has the possibility of having breast cancer [2]. Breast cancer can become fatal in late stages, comprising 15.5% of cancer-related deaths [2]. However, if it is detected in early stages and treatment begins, about 70-80% recovery rate has been observed [3]. Therefore, periodic controls are necessary to enable early diagnosis, particularly mammography, which is recommended once a year for women between the ages of 40 and 69 for breast cancer screening [2].

We propose an end-to-end system that is a doctor and patient assistant for early breast cancer screening. Empowered by XAI (explainable AI), it helps doctors, especially doctors analyze breast cancer at early stages with higher accuracy. The flexibility and accessibility that our system provides make it easier for patients to follow their screening results and contact their doctors. The availability of many doctors on one platform allows patients to access many opinions, satisfying the desire of the second opinion market.

In our system's workflow, patients upload their mammography images, which are analyzed by a doctor, and our XAI model is used separately. Then, if the results match, the diagnosis will be finalized; otherwise, another doctor will review both results by taking advantage of the explainability of the model and give a terminal decision.

The key objectives of our project include improving healthcare accessibility, reducing diagnosis errors through doctor and XAI collaboration, and providing a more flexible and comfortable working environment for doctors. Ultimately, our pioneering work seeks to revolutionize breast cancer screening and contribute to broader improvements in the healthcare system by integrating medical expertise with cutting-edge technology for the benefit of both patients and healthcare providers.

1.2 Design Goals

1.2.1 Usability

The mobile application and web-based platform will feature an intuitive and user-friendly interface with a learning curve not exceeding 30 minutes for new users. Being a user-friendly application means users can achieve their planned operations as quickly as possible. For this reason, medical image uploading can be done in three clicks, and XAI explanations to the doctors will be accessible in two clicks. The system currently supports English but will also support Turkish and Arabic languages.

1.2.2 Performance

Medical image uploads, processing, and assessment will have response times of 2 seconds, 5 seconds, and 3 seconds, respectively. The system will support a 200% increase in concurrent users without a decrease in response time. The system will ensure a minimum data transfer speed of 10 MBps for medical image uploads. System resource utilization will be at most 70% of available CPU and memory to ensure optimal performance.

1.2.3 Scalability

The system will automatically scale resources to accommodate a 100% increase in concurrent users within 5 minutes. Load balancing mechanisms will evenly distribute incoming requests across available servers to prevent overload and optimize performance. We aim to make our ML model usable across many platforms, including other applications, such as integration to e-Nabız [4]. To do that, we are deploying our ML model to AWS SageMaker [5].

1.2.4 Reliability

The system is aimed to maintain a minimum uptime of 99% to ensure 24/7 availability. We will implement our system in a way that is fault-tolerant. The system will implement robust encryption and access control mechanisms to ensure data integrity and prevent unauthorized access. We will integrate KPSPublic Web Service[6] into our system to enable users to register and sign in to our system with TR ID No, enhancing security. The system will recover from failures within 5 minutes and maintain core functionality even with faults. The system will follow GDPR[7] and KVKK [8] regulations in private data management.

1.2.5 Marketability

Marketability is a key consideration in our application's design. We prioritize user needs, industry trends, and feedback to create a product that offers clear value and competitive appeal.

By focusing on usability and aesthetics, we aim to position our application as a leading solution in the healthcare technology market, driving adoption and recognition for its innovation and effectiveness.

1.2.6 Extendibility

This application will initially be developed as a cross-platform mobile application on the patient side and a web application on the doctor side. For the upcoming releases, the application scope can be extended as a web application on the patient side, and new features can be added. System updates and maintenance will be scheduled during off-peak hours to minimize disruption, and the process will not exceed 1 hour. The system will provide a full suite of documentation, including FAQs, troubleshooting guides, and system architecture documentation, accessible from the user interface for patients and doctors.

1.2.7 Maintabilty

MediXAI is a patient-service tracking system, so it should be open to change in the future if extra needs or requests come from the patient or doctor side. An object-oriented approach will be used to make the application more maintainable and open to future changes. We plan to put minimal effort into changing the structure of the database in case of changes to the database. This is one of the reasons why we chose Cloud Firestore as the database service in our application [9].

1.2.8 Security

To satisfy the security measurements, our application stores the user credentials encrypted. Since we have a payment feature, cash transfers should be executed in a secure manner. Thus, we will use Stripe API to manage payments [10]. Email verification will be used during account creation. As mentioned, we will use the TC ID verification system [6] to ensure the user's existence and correct information.

1.2.9 Flexibility

We want to deploy our mobile application on the patient side and the web application on the doctor side. However, this may not happen for several reasons, such as the regularity conditions of the Health Ministry. Hence, we have other options, such as providing our application and ML model as a service to the Health Ministry.

1.2.10 Modularity

Modularity is a key design principle guiding the development of our application. We ensure flexibility and scalability by breaking down the system into smaller, self-contained modules. This modular approach facilitates easier integration of new features, enhances maintainability, and allows for independent updates to different parts of the system. Ultimately, modularity enables efficient collaboration among developers and ensures that our application remains adaptable to changing requirements and advancements in technology.

1.2.11 Aesthetics

Aesthetics play a crucial role in our application's design, ensuring an engaging and visually pleasing user experience. By incorporating clean and intuitive design elements, we aim to enhance usability and appeal to our target audience. Thoughtful use of color, typography, and layout not only improves the overall user interface but also reflects the professionalism and quality of our offering. Furthermore, attention to detail in graphical elements and visual hierarchy enhances clarity and guides users seamlessly through the application. Ultimately, prioritizing aesthetics elevates the user experience, making our application more attractive and enjoyable to use.

1.3 Definitions, Acronyms, and Abbreviations

NoSQL Database: NoSQL databases, short for "not only SQL databases," eschew traditional table structures for data storage, employing alternative methods such as JSON document storage. They excel in performance, adeptly managing large volumes of data and heavy user traffic.

JSON: JSON stands for JavaScript Object Notation, serving as a user-friendly method for object representation. It offers ease of parsing and serves as an excellent format for data interchange.

Flutter: Flutter is a Dart-based open-source framework created by Google. It enables the development of native mobile applications for both iOS and Android platforms using a unified codebase.

Dart: A programming language based on object-oriented principles, widely favored for constructing applications that run seamlessly across multiple platforms.

UI: UI stands for user interface, denoting the visual and interactive elements of software or hardware systems enabling user interaction. The primary aim of a user interface is to streamline communication between users and the system, ensuring efficient achievement of user objectives.

Firebase: Firebase is a comprehensive platform developed by Google that provides various cloud-based services for building and managing web and mobile applications. It offers services such as real-time database, authentication, hosting, storage, and more, all integrated into a single platform. Firebase enables developers to quickly develop and deploy applications without managing infrastructure, allowing them to focus more on creating engaging user experiences. Currently, we are utilizing authentication, storage, and real-time database.

AWS: AWS stands for Amazon Web Services, a cloud computing platform offered by Amazon. It encompasses various cloud-based services like computing power, storage, database management, and content delivery. Currently, we're exclusively utilizing hosting and storage services.

FlutterFlow: FlutterFlow is a visual development platform that simplifies the creation of Flutter applications. It offers an intuitive interface for designing and connecting user interfaces to backend services without writing code. With FlutterFlow, developers can easily prototype, iterate, and deploy cross-platform Flutter apps.

KVKK: Personal Data Protection Law No. 6698 (6698 Sayılı Kişisel Verilerin Korunması Kanunu)

GDPR: General Data Protection Regulation

BI-RADS: Breast Imaging-Reporting and Data System

API: Application Programming Interface

HIPAA: Health Insurance Portability and Accountability Act

KETEM: Cancer Early Diagnosis, Screening and Education Center (Kanser Erken Teşhis, Tarama ve Eğitim Merkezi)

XAI: Explainable Artificial Intelligence

SageMaker: Amazon SageMaker is a managed service by AWS that builds, trains, and deploys machine learning models at scale. It simplifies ML development with built-in algorithms, scalable training, hyperparameter tuning, and easy deployment.

Lex chat: Amazon Lex is a fully managed artificial intelligence (AI) service with advanced natural language models to design, build, test, and deploy conversational interfaces in applications [11].

Route 53: Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises [12].

Simple Email Service (SES): Amazon Simple Email Service (Amazon SES) lets you reach customers confidently without an on-premises Simple Mail Transfer Protocol (SMTP) email server using the Amazon SES API or SMTP interface [13].

Simple Storage Service (S3): Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps [14].

Elastic Compute Cloud (EC2): Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud [15].

Elastic Load Balancer (ELB): Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones (AZs) [16].

Load Balancer: A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the availability and fault tolerance of your applications [17].

1.4 Overview

MediXAI is a cross-platform application that is built using Flutter and Firebase. It is designed to be used as a mobile app by the patients and a web application by the doctors. There are two types of users: patients and doctors. Upon the account creation and verification, patients can start breast cancer checkups. They can upload their mammograms and additional materials like blood tests. Following that, they are waited for the payment. After payment completion, their checkup is assigned to a doctor, who assesses the mammogram. Our XAI model evaluates the mammogram at the same time. If both assessments yield the same result, a report will be provided to the patient with the BI-RAD classification and required steps. Otherwise, a second doctor's opinion will be consulted, and the results will be provided in the same format. If patients have questions regarding their checkup, they will be able to ask questions in the chat with the doctor part. Patients can also chat with the chatbot that we developed specifically for breast cancer detection and treatment. Doctors can answer the messages through the chat. They can ask for additional materials or follow-up mammograms to track the patient's progress.

2 Current Software Architecture

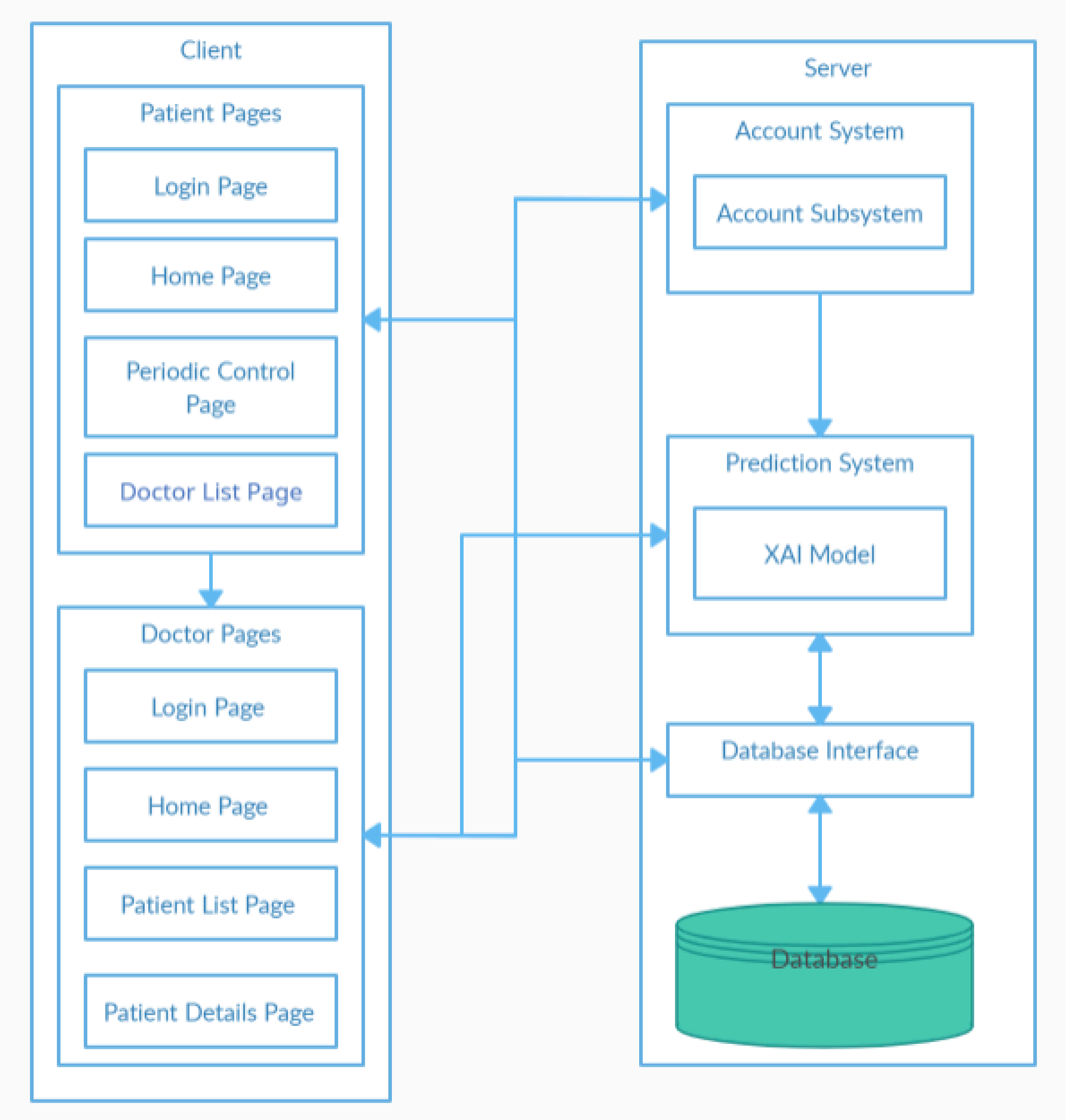


Fig.1: High-Level System Architecture

The diagram illustrates the structure of the application, where on the client side, patient-facing pages include login, home, and a periodic control section for health monitoring, whereas doctor-facing pages offer login, home, patient listing, and detailed patient information access. These interfaces interact with a server-side architecture that houses an accounting system to manage authentication and permissions, a prediction system that has an explainable AI (XAI) model to analyze health data and make forecasts, and a database interface acting as a mediator between the server processes and the underlying database that stores all user and data. Arrows show the flow of information: both patient and doctor pages connect to the accounting system for secure access, patient data can be routed towards the AI-powered prediction system for health insights, while doctors have the additional capability to interact with the prediction outcomes and directly with the database interface, suggesting more comprehensive access to data for treatment planning and monitoring purposes, all within a coherent framework designed to streamline the management of healthcare information.

Currently, as it is common in all healthcare domains, periodic breast cancer checks are also heavily dependent on the doctors in the decision-making process. In the current system, women between 40 and 69 are highly recommended to do their checks at a health center every 18 months. The process is like this: firstly, the patient describes her clinical information, diseases in the past, and whether she has a child or not. After that, they took a mammography image, and according to mammography classification, which is in the BI-RAD classification standard, they are guided to take further actions or come for a check again after 18 months[4].

This process described above is conducted in public and private health centers in Turkey. Additionally, there are public health centers called KETEM that are only designated for the periodic checks of mammography, colon cancer, and HPV. KETEM provides imaging and analysis of the result for free [5]. Furthermore, KETEM buys mammography analysis services from private image analysis companies in the current system. However, in the current system, based on the observations of the doctors and nurses in the field, it is known that there are significant delays(2-3 months), which is creating disapprobation of the clients and creating health risks. To sum up, in the current system, mammography analysis requires effort, takes time, and takes money.

There are a couple of private competitors that are also aiming to take part in the atomization of mammography analysis. The application has no direct competitors but indirect competitors. Those indirect competitors can be classified into two main application areas: the second medical opinion market and the doctor assistance market. Our application is a combination of these services. and providing an end-to-end product.

Companies that are providing second opinions in mammography analysis and operating in Turkey:

Hastalar Soruyor: Hastalar Soruyor is a web-based application that provides patients with several services, which can be listed as consulting a doctor, analyzing medical test results, getting a second doctor's opinion, and making an appointment with a doctor. Patients are asked to pay for these services, which is determined before each service [6].

Medicopin.com: Medicopin is a healthcare service that connects users with top medical experts worldwide, allowing them to share health concerns and receive expert evaluations via personalized video recordings or face-to-face consultations with doctors through Medicopin.com. It aims to provide users with reliable medical advice and the peace of mind needed during severe health issues while offering ongoing support for a stress-free healthcare experience [7].

Companies that are providing doctor assistance in mammography analysis and operating in Turkey:

Screen Point Transpara: Screen Point Transpara is a product that aims to make early detection using machine learning methods for breast cancer. Additionally, it is FDA-cleared for use on all major 2D and 3D mammography systems. Their research shows they have a similar recall, cancer detection rate, and false positive rate compared to the standard double-checking. They currently provide services to the local private oncology clinics in İstanbul [8].

ICTerra: ICTerra’s effort in AI is framed in oncology and radiology. They have a project to classify mammograms using XAI. Their main target market is governmental agencies and hospitals. The details of the project and benchmarks of the model are not shared publicly [9].

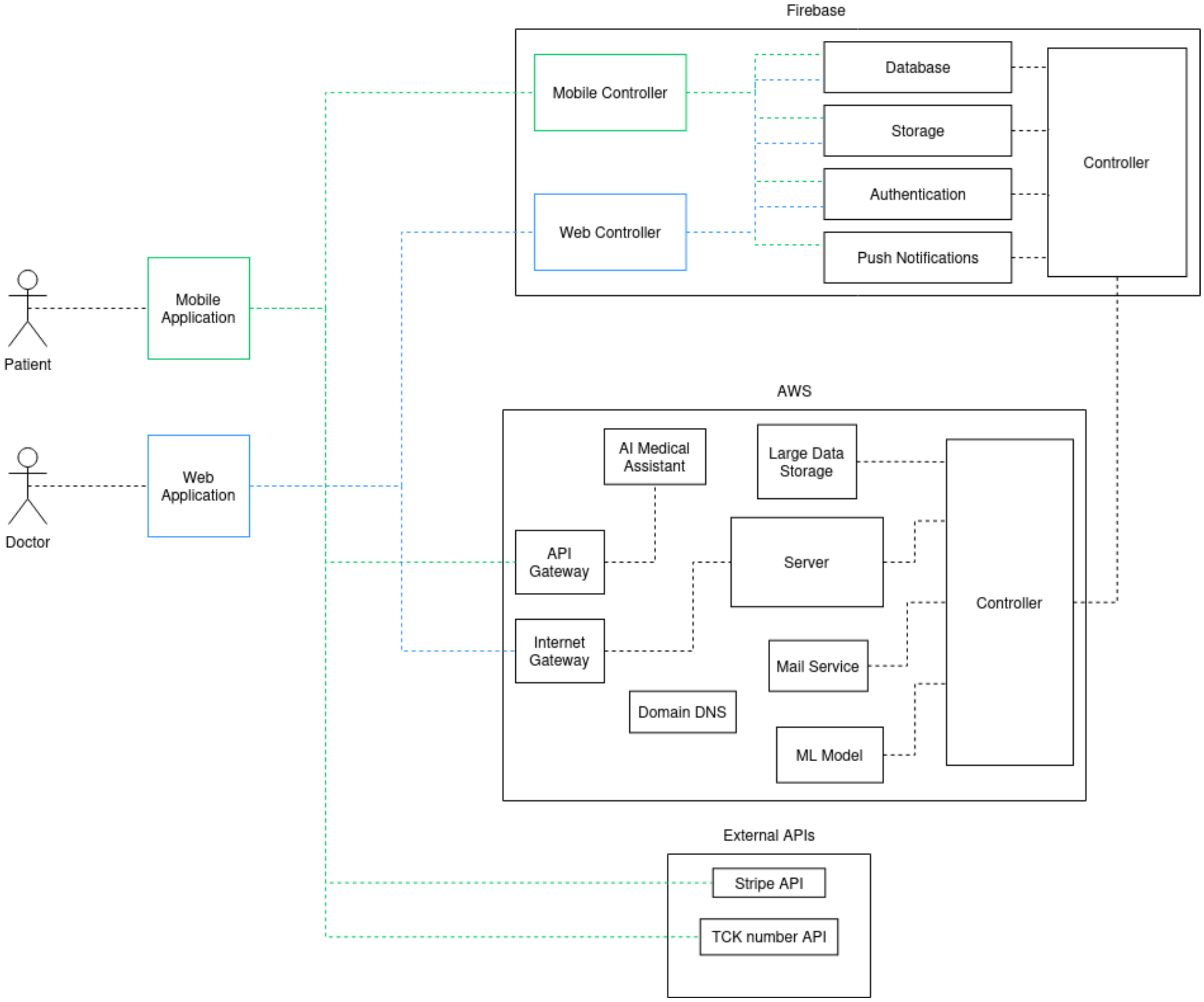
Siemens Syngo Breast Care: Siemens is one of the leading companies in the imaging machine manufacturing industry. They also developed software integrated into the imaging mammography machine [10].

3 Proposed Software Architecture

3.1 Overview

The architecture or system of the project comprises 5 main subsystems which are a mobile application, a web application, and integrates with AWS, Firebase, and two external APIs. In the following sections, we'll go over subsystem decomposition, breaking down our system into the mentioned 5 parts and how they talk to each other. We'll also discuss hardware/software mapping, explaining how our system works with hardware and how different parts interact. Plus, we'll cover persistent data management, explaining how we save information for later use, making sure it stays available. Lastly, we'll talk about access control and security, detailing how we keep everything safe and only accessible to the right people. These sections will give you a clearer picture of how our system works behind the scenes.

3.2 Subsystem Decomposition



We have decided to build our application on five main components: Mobile Application, Web Application, AWS, Firebase, and External APIs.

Mobile Application: The mobile application is designed for patients to upload their mammography data, view the BI-RAD result conveyed according to the uploaded mammography data, and communicate with the radiologist who reviewed the mammography.

Web Application: The web application is intended for radiologists to evaluate incoming mammography data, communicate with patients, and review AI model results for the same data.

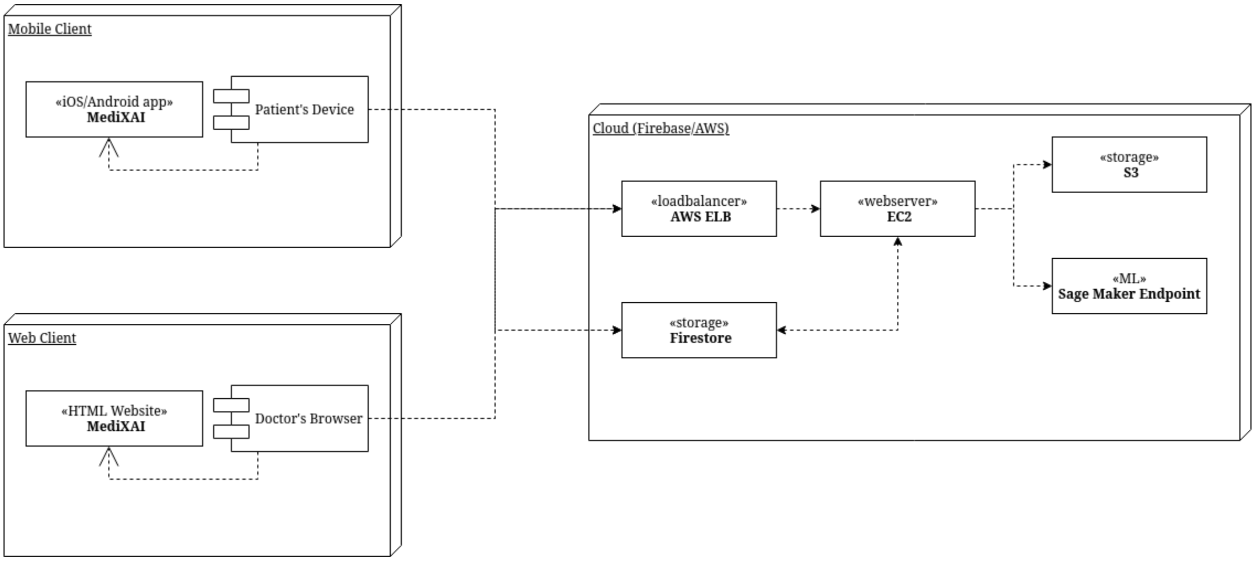
AWS: Amazon Web Services (AWS) is a crucial component of our architecture. It hosts the ML model that is responsible for evaluating incoming mammography data. Additionally, AWS provides services for email communication, large data storage, and Domain DNS management and hosts an AI medical assistant, which functions as a chatbot for users to discuss their medical concerns.

Firebase: Firebase is utilized for various purposes, including database management for user registration information, authentication, notifications, and storage of user medical data. It ensures secure and efficient data handling for our application's backend operations.

External APIs:

- **TCK Number API:** This API is integrated into our system for specific functionality related to TCK number validation or other relevant operations.
- **Stripe API:** The Stripe API is utilized for payment management within the application, ensuring secure and seamless transactions for users.

3.3 Hardware/Software Mapping



In our system architecture, we have two primary endpoints: the Patient Device, where users access our MediXAI application, and the Doctor's Browser, through which healthcare professionals interact with our system by accessing the MediXAI HTML website.

Patient Device:

- Users utilize our MediXAI application on their devices.
- Transactions initiated on the Patient Device are directed either to AWS or Firestore based on the request made.

Doctor's Browser:

- Healthcare professionals access our system via their web browsers by directing to the MediXAI HTML website.
- Similar to the Patient Device, transactions from the Doctor's Browser are routed to either AWS or Firestore, depending on the request.

AWS Path:

- Requests from both the Patient Device and the Doctor's Browser are directed through AWS Elastic Load Balancer (ELB) for load distribution.

- The requests are then forwarded to server instances hosted on Amazon Elastic Compute Cloud (EC2).
- Depending on the nature of the request, transactions proceed to either Amazon S3 for storage or to a SageMaker Endpoint for machine learning model inference.

Firestore Side:

- Firestore handles requests made to it.
- Firestore is connected to EC2 instances, enabling bidirectional communication between Firestore and AWS.
- This connectivity allows for seamless integration between Firestore and AWS services, facilitating the exchange of data and requests between the two platforms.

3.4 Persistent Data Management

Firestore Usage:

- Initially, all user data will be stored in Firestore.
- Firestore provides real-time synchronization and seamless integration with our application's frontend and backend components.
- This choice aligns with our current architecture and development preferences, allowing for rapid iteration and deployment.

Integration with Amazon S3:

- As our data volume increases and storage requirements evolve, we anticipate leveraging Amazon S3 for its scalability and cost-effectiveness, particularly for storing large datasets.
- We will implement a strategy to transition data from Firestore to Amazon S3 as needed, based on access patterns and data usage.
- Data that has not been accessed recently or does not require real-time synchronization can be migrated to Amazon S3 to free up space and optimize costs on Firestore, especially considering the pay-as-you-go model of Firestore compared to the credits available on AWS.

3.5 Access Control and Security

3.5.1 Firebase

Since we are using Flutterflow to develop our frontend application, we chose to integrate FIrebase Authentication as it is automatically integrated into the application. All authentication operations from signup, login, change password, session management and more are automatically handled by the combination of automatically generated flutter code and Firebase Authentication that connects to the mobile or web application to provide the necessary authentication services. As for Firestore and Firebase Storage, they are directly integrated with

Firebase Authentication, allowing data to be only accessed from authenticated users according to the permissions that we defined in the configuration file.

3.5.2 AWS

Our services on AWS also follow security and access control mechanisms. The main security and access control service used is AWS Security Groups (SG) [18]. This allows for custom control over all services on AWS as to who has access to what and what protocols or ports they can use. SGs are mandatory for all AWS services and have a deny policy by default for all transactions. Therefore unless allowed, no service or unauthorized access can occur since it will be rejected by default.

3.5.3 User Data Protection (Our encryption scheme)

As to the users' data, and especially data like mammograms and medical reports, we follow strict security policies. We will be adopting a system that is currently under development by our security and privacy team. We will follow an encryption scheme for the data in storage that will not allow anyone to access it even if our cloud servers got compromised. We as service providers will not be able to access the data as well. All medical data will only be accessible by the patients uploading them and by the doctors that are assigned to those patients. Our encryption scheme utilizes three main key generation mechanisms to allow for more flexibility while maintaining full security. The data encryption process consists of four steps (in this section two subscripts are used for the keys, P indicates a key generated by the Patient and D indicates a key generated by the Doctor):

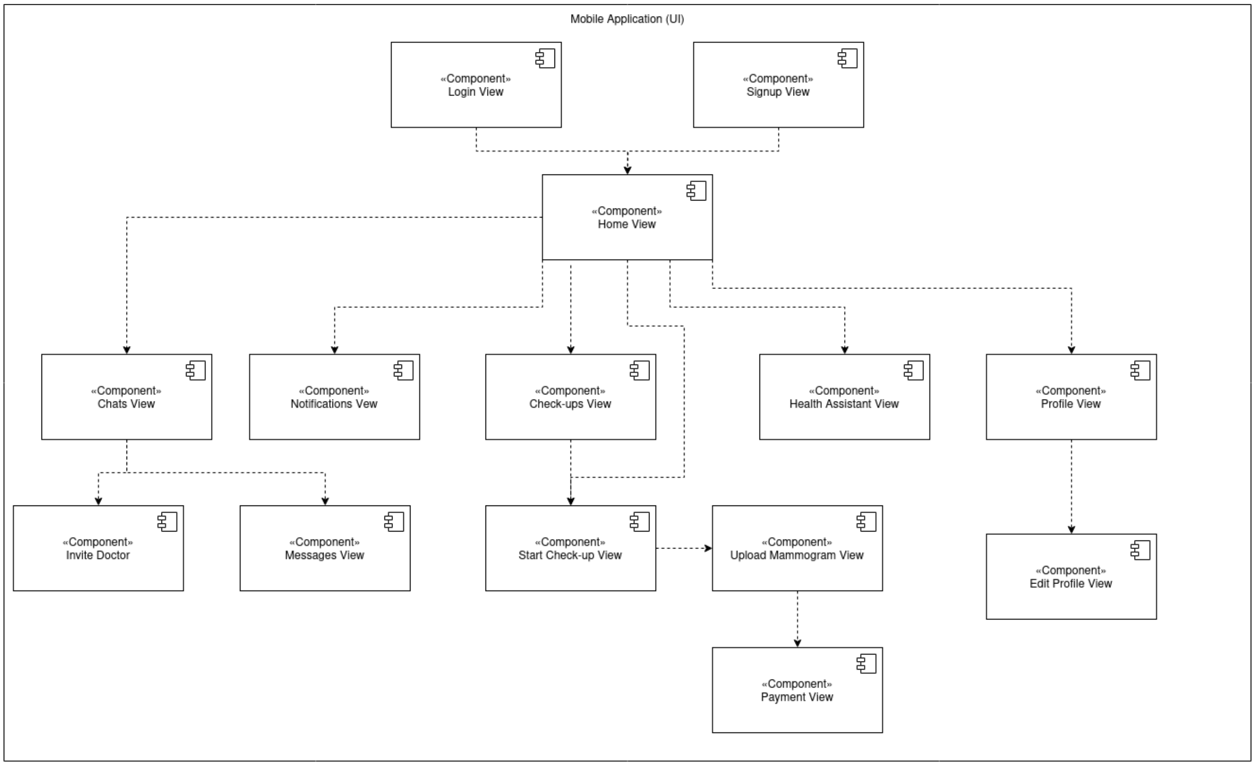
1) Both the patient and the associated doctor will generate a public/private key pairs ($\mathbf{PBK_D}$/$\mathbf{PVK_D}$ and $\mathbf{PBK_P}$/$\mathbf{PVK_P}$) using the RSA key generation algorithm. Each of the doctor and patient will store their public keys ($\mathbf{PBK_D}$ and $\mathbf{PBK_P}$) in the database for later use.
2) While uploading their medical data, patients will generate a new key ($\mathbf{K_i}$) to encrypt the data within appointment i. Each appointment will have its own key $\mathbf{K_i}$. This key will be generated using AES-256, allowing for fast symmetrical encryption, while having extremely high security.
3) Then, both the patients and the doctors will use Argon2, which is a Password Based Key Derivation Function (PBKDF) to generate keys ($\mathbf{PassK_P}$ and $\mathbf{PassK_D}$) derived from their accounts' passwords. This key will be used to encrypt many other keys since the doctors and patients always have access to their passwords, enabling them to switch devices seamlessly and eliminating the need to store encryption keys locally, since now the keys can be encrypted using this key obtained by the PBKDF and stored on the cloud privately. $\mathbf{PassK_P}$ will be used to encrypt $\mathbf{PVK_P}$ and $\mathbf{K_i}$, while $\mathbf{PassK_D}$ will be used to encrypt $\mathbf{PVK_D}$.
4) After all keys are generated and the data is encrypted the following data will be uploaded to the cloud:
    a) The medical data encrypted with key $\mathbf{K_i}$

b) **PVK_P** encrypted with key **$PassK_P$** to allow patients to access their private key from other devices
c) **K_i** encrypted with key **PBK_D** to allow the doctor to access it
d) **K_i** encrypted with **$PassK_P$** to allow the patient to access it
e) **PVK_D** encrypted with **$PassK_D$** to allow doctors to access their private key from other devices
f) **PBK_D**
g) **PBK_P**

The above encryption scheme allows for maximum security, making it computationally infeasible to decrypt user data even if the data was obtained through some malicious exploit. One consideration to note is that if any of the users forgets their password, their previously uploaded data will not be accessible anymore. We cannot recover their data since the data is encrypted with their password derived key. In that case, users can create a new account and notify us that they lost access to their previous account. In that case we will request their corresponding doctors to access the patients' data since they are the only ones that can do so, and after that the data can be re-encrypted with the new patient's password derived key. The old inaccessible data will be deleted from our servers.

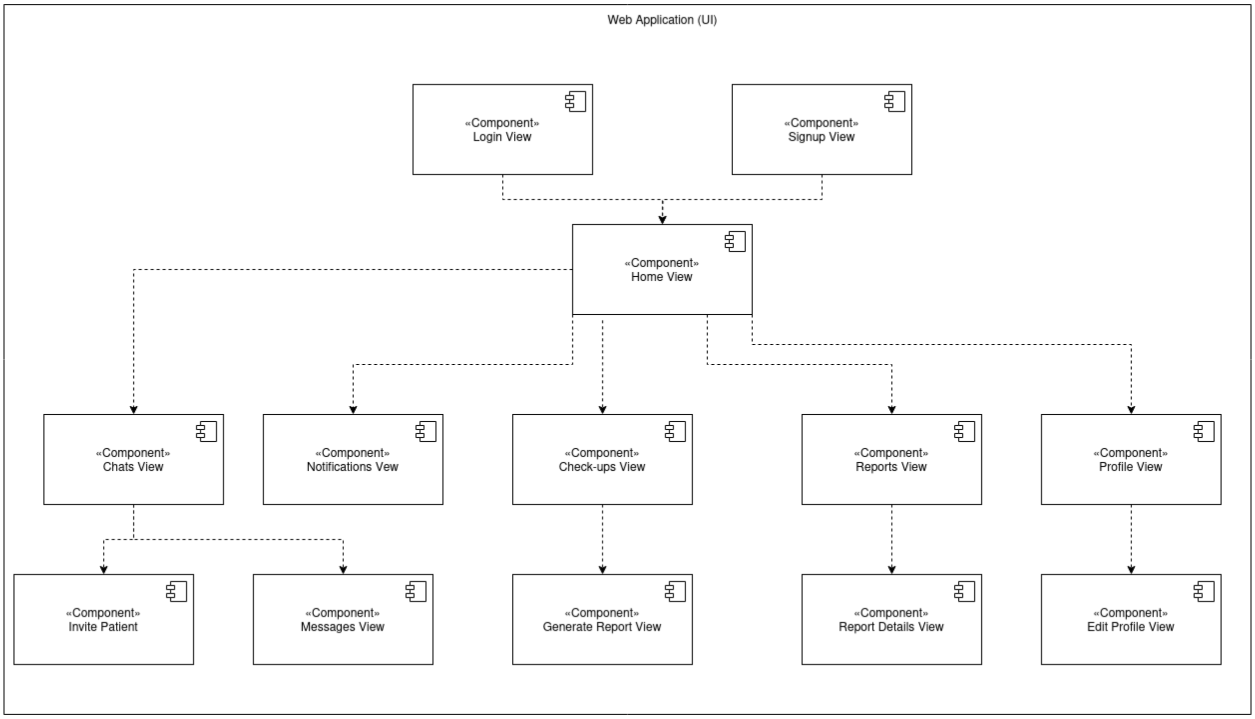
4 Subsystem Services

4.1 Mobile Application


Mobile Application (UI)
«Component» Login View
«Component» Signup View
«Component» Home View
«Component» Chats View
«Component» Notifications Vew
«Component» Check-ups View
«Component» Health Assistant View
«Component» Profile View
«Component» Invite Doctor
«Component» Messages View
«Component» Start Check-up View
«Component» Upload Mammogram View
«Component» Edit Profile View
«Component» Payment View

Login View: This view contains the Login screen.
Signup View: This view contains the Signup screen.
Home View: This view contains a screen where there is a dashboard with which patients can visit other Views.
Chats View: This view contains a screen where patients can see their chat history with others and also link to both Invite Doctor View and Messages View.
Invite Doctor View: In this view, patients can send chat invitations to their assigned doctor via a button.
Messages View: In this view, patients can review their messaging with a doctor.
Notifications View: This view contains notifications that come to patients.
Check-up View: This view contains a screen where patients can see all their check-ups made so far.
Start Check-up View: Via a button in this view, patients can start the process of starting a checkup and they will be directed to Upload Mammogram View.
Upload Mammogram View: In this view, patients can upload their mammogram data to the application. After uploading, they will be directed to Payment View if they click the "Continue" button.
Payment View: From this view, patients can pay their fee for forming a check-up.
Profile View: This view contains a screen where patients can see their information.
Edit Profile View: This view allows patients to edit their profile information.

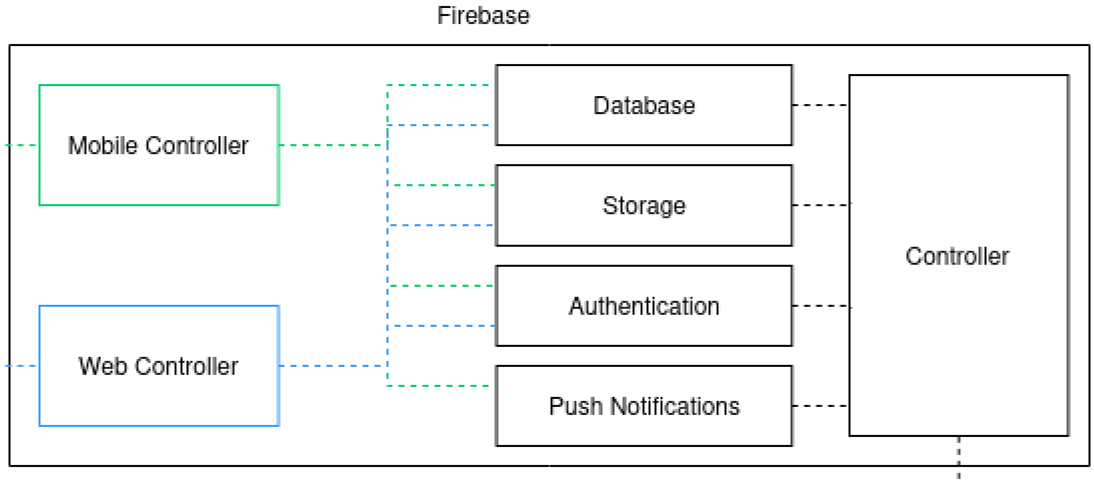
4.2 Web Application


Web Application (UI)
«Component» Login View
«Component» Signup View
«Component» Home View
«Component» Chats View
«Component» Notifications Vew
«Component» Check-ups View
«Component» Reports View
«Component» Profile View
«Component» Invite Patient
«Component» Messages View
«Component» Generate Report View
«Component» Report Details View
«Component» Edit Profile View

Login View: This view contains the Login screen.

Signup View: This view contains the Signup screen.
Home View: This view contains a screen where there is a dashboard with which doctors can visit other views.
Chats View: This view contains a screen where doctors can see their chat history with others and also link to both Invite Patient View and Messages View.
Invite Patient View: In this view, doctors can send chat invitations to their patients via a button.
Messages View: In this view, doctors can review their messaging with a patient.
Notifications View: This view contains notifications that come to doctors.
Check-up View: This view contains a screen where doctors can see all the check-ups assigned to them.
Generate Report View: This view allows doctors to generate a report about their diagnosis on a check-up assigned to them.
Reports View: This view contains a screen where doctors can see all the reports they generated so far.
Report Details View: This view contains a screen where doctors can review the details of a chosen report.
Profile View: This view contains a screen where doctors can see their information.
Edit Profile View: This view allows doctors to edit their profile information.

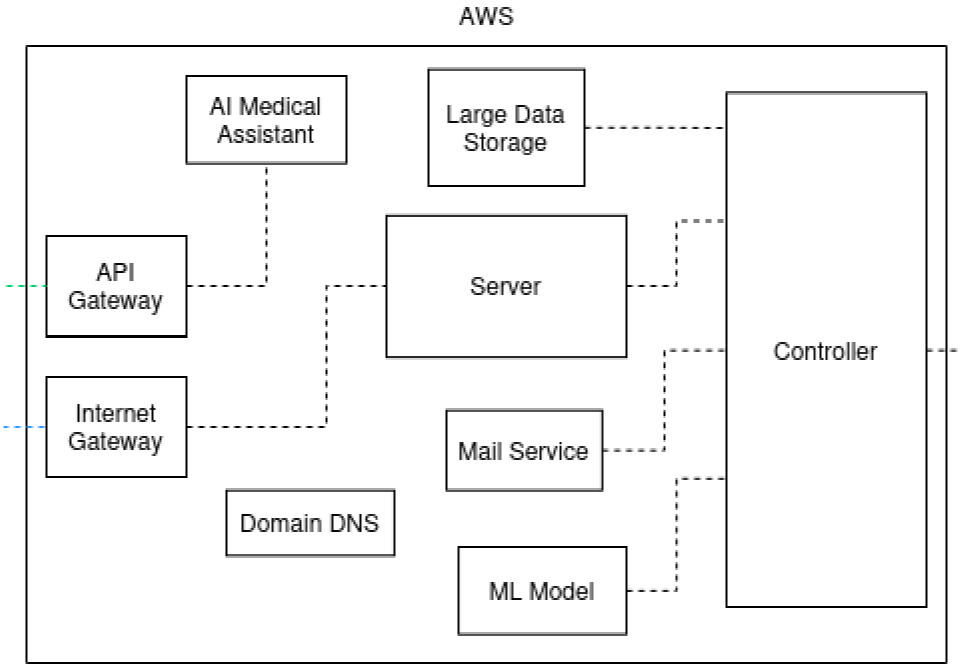
4.3 Firebase



Web Controller: The Web Controller handles requests initiated from the doctor's side and interacts with Firebase to perform various operations such as retrieving patient data, updating medical records, and communicating with the backend system.
Mobile Controller: The Mobile Controller manages requests made from the patient's side through the mobile application and communicates with Firebase to handle tasks like uploading mammogram data, accessing medical reports, and facilitating communication with healthcare providers.

Database: The Database component in Firebase stores structured data related to the breast cancer mammogram project, including patient profiles, medical records, mammogram results, and communication logs between patients and doctors.
Storage: The Storage feature in Firebase provides scalable cloud storage for storing large files such as mammogram images, patient documents, and other medical data. It ensures secure and reliable storage for user-generated content associated with the project.
Authentication: Firebase Authentication handles user authentication and authorization, ensuring secure access to the application. It supports various authentication methods such as email/password, phone number authentication, and third-party providers, allowing users to log in and access their medical data securely.
Push Notifications: Push Notifications enable real-time communication and updates for users of the application. Firebase Cloud Messaging (FCM) is used to send push notifications to patients and doctors, notifying them of new medical reports, appointment reminders, and important updates related to the project.

4.4 AWS



AI Medical Assistant (Lex chat): The AI Medical Assistant, powered by Amazon Lex, provides a conversational interface for users to interact with the application. It understands natural language queries and provides responses, helping users navigate the application, access medical information, and receive assistance with their healthcare concerns.
Domain DNS (Route 53): Route 53 is used to manage the domain name system (DNS) for our project. It translates human-readable domain names into IP addresses, ensuring users can access the application using a custom domain name. Route 53 also provides features like

domain registration, DNS health checks, and routing policies for improved reliability and performance.

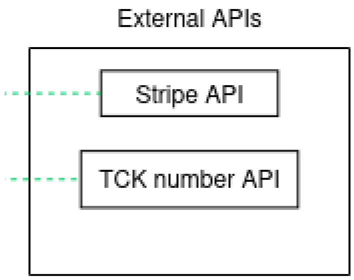
ML Model (SageMaker Endpoint): SageMaker Endpoint hosts the machine learning model used in the application for analyzing mammogram images and providing diagnostic insights. It serves as an endpoint for real-time inference, allowing the application to make predictions based on incoming data.

Mail Service (Simple Email Service (SES)): The Mail Service, powered by Amazon Simple Email Service (SES), facilitates email verification and report distribution within the app. It ensures reliable email delivery, enables email verification for user registration, and facilitates the automated sending of medical reports to patients and healthcare providers.

Large Data Storage (Simple Storage Service (S3)): S3 is utilized for large-scale storage of data associated with MediXAI project, including mammogram images, medical records, and other healthcare-related files.

Server (Elastic Compute Cloud (EC2)): EC2 instances serve as the underlying compute infrastructure for hosting backend services, web applications, and other components of the MediXAI project. EC2 offers scalable virtual servers in the cloud, providing flexibility and control over computing resources to support the application's requirements for processing data, running algorithms, and serving user requests.

4.5 External APIs



TCK Number API: This API is integrated into our system for TCK number validation.

Stripe API: The Stripe API is utilized for payment management within the application, ensuring secure and seamless transactions for users.

5 Test Cases

Test ID Format (from left to right):
First two digit: 00 (Functional), 01 (Non-functional)
Third digit: 0 (mobile app), 1 (web app), 2 (XAI Model)
Last two digit: incremental

Patient Signup

Test ID	00001	**Category**	Functional	**Severity**	Critical
Objective	1-To verify that patients can successfully sign up for the mobile app. 2-To verify that patients cannot sign up and get an error message when they enter invalid information for the mobile app.				
Steps	1. Open the mobile app. 2. Click on the signup button. 3. Enter valid/invalid patient details (name, email, password). 4. Click on the signup button.				
Expected Result	1-The patient should be registered successfully and redirected to the login page. 2-The patient should not be registered and should get an error message.				
Date Tested & Test Result					

Patient Login

Test ID	00002	**Category**	Functional	**Severity**	Critical
Objective	1-To verify that patients can successfully login to the mobile app. 2-To verify that patients cannot login and get an error message when they enter invalid information for the mobile app.				
Steps	1. Open the mobile app. 2. Tap on the login button. 3. Enter valid/invalid patient credentials (email and password). 4. Tap on the login button.				

Expected Result	1-Patient should be successfully logged in and redirected to the home page. 2-The patient should not be logged in and should get an error message.
Date Tested & Test Result	

Doctor Signup

Test ID	00101	**Category**	Functional	**Severity**	Critical
Objective	1-To verify that doctors can successfully sign up for the web app. 2-To verify that doctors cannot sign up and get an error message when they enter invalid information for the web app.				
Steps	1. Open the web app. 2. Click on the signup button. 3. Fill in the required doctor information (name, email, password, doctor verification info). 4. Click on the signup button.				
Expected Result	1-Doctor should be successfully registered and redirected to the login page. 2-The doctor should not be registered and should get an error message.				
Date Tested & Test Result					

Doctor Login

Test ID	00102	**Category**	Functional	**Severity**	Critical

Objective	1-To verify that doctors can successfully login to the web app. 2-To verify that doctors cannot login and get an error message when they enter invalid information for the web app.
Steps	1. Open the web app. 2. Enter valid doctor credentials (email and password). 3. Click on the login button.
Expected Result	1-Doctor should be successfully logged in and redirected to the dashboard. 2-The doctor should not be logged in and should get an error message.
Date Tested & Test Result	

Start a Check-Up

Test ID	00003	**Category**	Functional	**Severity**	Critical

Objective	To confirm patients can start a check-up.
Steps	1. Log in to the patient app. 2. Navigate to the Check-Up section. 3. Click on "Start Check-Up" button.
Expected Result	Check-up request recorded, waiting for user medical data.
Date Tested & Test Result	

Upload Medical Data

Test ID	00004	**Category**	Functional	**Severity**	Critical

Objective	To verify that patients can upload their medical data.
Steps	1. Login to the mobile app. 2. Navigate to the Check-up section. 3. Click on the Start Check-up button. 4. Click on the Upload Data button. 5. Select a medical data file (mammogram images). 6. Click on the upload button.
Expected Result	Files should be successfully uploaded to the cloud storage and their path to the database.
Date Tested & Test Result	

View Patient's Medical Data

Test ID	00103	**Category**	Functional	**Severity**	Critical
Objective	To verify doctors can view patient's medical data.				
Steps	1. Login to the web app as a doctor. 2. Navigate to the patient list. 3. Click on a patient's check-up. 4. View the uploaded medical data.				
Expected Result	Doctor should be able to view patient's uploaded medical data.				
Date Tested & Test Result					

Generate Report

Test ID	00104	**Category**	Functional	**Severity**	Critical
Objective	To verify the report generation process.				
Steps	1. Make a doctor analyze patient's medical data. 2. Make XAI model analyze patient's medical data. 3. Trigger the report generation process.				
Expected Result	A comprehensive report should be generated based on the medical data analysis by doctor and XAI model.				
Date Tested & Test Result					

View Report

Test ID	00005	**Category**	Functional	**Severity**	Critical
Objective	To verify that patients can view their check-up reports.				
Steps	1. Login to the mobile app as a patient. 2. Navigate to the My Check-ups section. 3. Click on the View Report button.				
Expected Result	Patient should be able to view the generated check-up report.				
Date Tested & Test Result					

XAI Model Training & Test

Test ID	00201	**Category**	Functional	**Severity**	Critical
Objective	To verify the training and testing of the XAI model.				

Steps	1. Prepare medical image dataset. 2. Train model on that dataset with tuning hyperparameters. 3. Get the test result. 4. Evaluate the result.
Expected Result	XAI model should achieve good scores based on determined metrics (e.g. accuracy, f1-score)
Date Tested & Test Result	

View Waiting Assessments

Test ID	00105	**Category**	Functional	**Severity**	Critical
Objective	To verify doctors can see pending assessments.				
Steps	1. Log in to the doctor app. 2. Navigate to the "My Patients" section.				
Expected Result	Doctors can see pending assessments for their patients.				
Date Tested & Test Result					

Assess Medical Data

Test ID	00106	**Category**	Functional	**Severity**	Critical
Objective	To confirm doctors can assess medical data.				

Steps	1. Log in to the doctor app. 2. Select a pending assessment. 3. Review uploaded image and other medical data. 4. Give a BI-RADS score for mammograms. 5. Add notes and suggestions. 6. Click on "Submit Assessment" button.
Expected Result	Analysis provided with optional notes.
Date Tested & Test Result	

View First Doctor’s Diagnosis

Test ID	00107	**Category**	Functional	**Severity**	Critical
Objective	To ensure the second doctors can access the first doctor’s diagnosis.				
Steps	1. Log in to the doctor app. 2. Select a pending assessment. 3. Click on “First Doctor Analysis” button.				
Expected Result	The first doctor’s diagnosis displayed to the second doctor.				
Date Tested & Test Result					

View XAI Diagnosis

Test ID	00108	**Category**	Functional	**Severity**	Critical

Objective	To ensure the second doctors can access XAI diagnosis.
Steps	4. Log in to the doctor app. 5. Select a pending assessment. 6. Click on “XAI Analysis” button.
Expected Result	XAI diagnosis displayed to the second doctor.
Date Tested & Test Result	

Edit Profile for Patients

Test ID	00006	**Category**	Functional	**Severity**	Minor
Objective	To verify patients can edit their profiles.				
Steps	1. Log in to the mobile app. 2. Navigate to the “My Profile” section. 3. Click on the "Edit Profile" button. 4. Update necessary information. 5. Click on the "Save Changes" button.				
Expected Result	Patient profile updated successfully.				
Date Tested & Test Result					

Edit Profile for Doctors

Test ID	00109	**Category**	Functional	**Severity**	Minor

Objective	To verify doctors can edit their profiles.
Steps	1. Log in to the web app. 2. Navigate to the “My Profile” section. 3. Click on the "Edit Profile" button. 4. Update necessary information. 5. Click on the "Save Changes" button.
Expected Result	Doctor profile updated successfully.
Date Tested & Test Result	

Notification for Pending Assessments

Test ID	00110	**Category**	Functional	**Severity**	Major
Objective	To ensure doctors receive notifications for pending assessments.				
Steps	1. Log in as a doctor. 2. Wait for a new check-up request.				
Expected Result	Doctor receives a notification indicating a new assessment is pending.				
Date Tested & Test Result					

Notifications for Patients

Test ID	00007	**Category**	Functional	**Severity**	Major
Objective	To ensure patients can get notifications.				

Steps	1. Click on “Notifications” button. 2. Get for different events (e.g., new diagnosis, chat messages).
Expected Result	Patients receive notifications successfully.
Date Tested & Test Result	

Message Sending for Patients

Test ID	00008	**Category**	Functional	**Severity**	Major
Objective	To verify that patients can send messages to doctors.				
Steps	1. Log in as a patient. 2. Navigate to the chat section. 3. Select a doctor and compose a message. 4. Click send.				
Expected Result	Message is sent successfully to the selected doctor.				
Date Tested & Test Result					

Message Receiving for Patients

Test ID	00009	**Category**	Functional	**Severity**	Major
Objective	To confirm that patients receive messages from doctors.				
Steps	1. Log in as a patient. 2. Check the chat for new messages from doctors.				

Expected Result	Patient receives the message sent by doctors.
Date Tested & Test Result	

Message Sending for Doctors

Test ID	00111	**Category**	Functional	**Severity**	Major
Objective	To verify that doctors can send messages to patients.				
Steps	5. Log in as a doctor. 6. Navigate to the chat section. 7. Select a patient and compose a message. 8. Click send.				
Expected Result	Message is sent successfully to the selected patient.				
Date Tested & Test Result					

Message Receiving for Doctors

Test ID	00112	**Category**	Functional	**Severity**	Major
Objective	To confirm that doctors receive messages from patients.				
Steps	3. Log in as a doctor. 4. Check the chat for new messages from patients.				
Expected Result	Doctor receives the message sent by patients.				

Date Tested & Test Result	

Successful Payment

Test ID	00010	**Category**	Functional	**Severity**	Critical
Objective	To verify that patients can successfully make payments for check-up.				
Steps	1. Log in as a patient. 2. Navigate to the payment section. 3. Select an assessment to pay for. 4. Enter payment details and confirm payment.				
Expected Result	Payment is processed successfully, and the check-up is marked as paid.				
Date Tested & Test Result					

XAI Model Integration

Test ID	01201	**Category**	Non-Functional	**Severity**	Critical
Objective	To verify the integration of the XAI model.				
Steps	1. Upload a sample medical data. 2. Verify the XAI model's prediction results on the database.				
Expected Result	XAI model should accurately analyze the medical data, provide relevant insights and write the prediction to the database.				
Date Tested & Test Result					

Device Compatibility

Test ID	01001	**Category**	Non-Functional	**Severity**	Major
Objective	To verify that the mobile application work across different devices.				
Steps	1. Use the mobile app on various devices (Android, iOS).				
Expected Result	App should function properly across different devices without any issues.				
Date Tested & Test Result					

Browser Compatibility

Test ID	01101	**Category**	Non-Functional	**Severity**	Major
Objective	To verify that the web application work across different browsers.				
Steps	1. Access the web app from different browsers (Chrome, Firefox).				
Expected Result	App should function properly across different browsers without any issues.				
Date Tested & Test Result					

Data Encryption

Test ID	01002	**Category**	Non-Functional	**Severity**	Critical
Objective	To verify data security through encryption.				

Steps	1. Review the data transmission process. 2. Verify data encryption protocols are implemented (for password or other sensitive info).
Expected Result	Patient's data should be encrypted during transmission and storage.
Date Tested & Test Result	

Response Time

Test ID	01003	**Category**	Non-Functional	**Severity**	Major
Objective	To test the system's response time.				
Steps	1. Upload medical data. 2. Analyze the data and generate reports.				
Expected Result	System should respond within acceptable time limits for all operations.				
Date Tested & Test Result					

Scalability

Test ID	01004	**Category**	Non-Functional	**Severity**	Major
Objective	To test the system's ability to handle increased load.				
Steps	1. Simulate increased user activity. 2. Monitor system performance.				
Expected Result	System should maintain performance even under high loads without crashing or slowing down significantly.				

Date Tested & Test Result	

Reliability

Test ID	01005	**Category**	Non-Functional	**Severity**	Major
Objective	To verify the reliability of the applications.				
Steps	1. Perform tasks repeatedly. 2. Monitor for crashes or unexpected behavior.				
Expected Result	Applications should function reliably without frequent crashes or errors.				
Date Tested & Test Result					

Multi-device Synchronization for Patient App

Test ID	01006	**Category**	Non-Functional	**Severity**	Major
Objective	To confirm synchronization of data across multiple devices for patients.				
Steps	1. Log in as a patient on multiple devices simultaneously. 2. Upload an image from one device and check if it reflects on the other.				
Expected Result	Data is synchronized across all logged-in devices in real-time.				
Date Tested & Test Result					

Multi-device Synchronization for Doctor App

Test ID	01102	**Category**	Non-Functional	**Severity**	Major
Objective	To confirm synchronization of data across multiple devices for doctors.				
Steps	1. Log in as a doctor on multiple devices simultaneously. 2. Analyze an image and check if data synchronized on the other device.				
Expected Result	Data is synchronized across all logged-in devices in real-time.				
Date Tested & Test Result					

Patient App Installation

Test ID	01007	**Category**	Non-Functional	**Severity**	Critical
Objective	To verify app installation process without any problem.				
Steps	1. Install app on different devices.				
Expected Result	App installs without errors.				
Date Tested & Test Result					

Doctor App Deployment

Test ID	01103	**Category**	Non-Functional	**Severity**	Critical
Objective	To ensure deployment of the web app without any problem.				
Steps	1. Deploy web app on a server.				

	2. Try to access it by providing the domain name.
Expected Result	Web app accessible without issues.
Date Tested & Test Result	

Usability for Patient App

Test ID	01008	**Category**	Non-Functional	**Severity**	Major
Objective	To evaluate the user-friendliness of the patient app.				
Steps	1. Provide a sample user with the patient app. 2. Ask them to perform common tasks such as check-up request and image upload.				
Expected Result	User finds the app intuitive and easy to use.				
Date Tested & Test Result					

Usability for Doctor App

Test ID	01104	**Category**	Non-Functional	**Severity**	Major
Objective	To evaluate the user-friendliness of the doctor app.				
Steps	1. Provide a sample doctor with the doctor app. 2. Ask them to perform common tasks such as reviewing all pending analyses and providing analysis.				
Expected Result	Doctor finds the app efficient and easy to navigate.				

Date Tested & Test Result	

6 Consideration of Various Factors in Engineering Design

In this section the possible effects of various factors on the engineering design process of the project are discussed in detail.

6.1 Public Health Considerations

MediXAI is a mammography analysis app because of that it prioritizes accuracy in diagnostics as its main consideration. At the same time it ensures the collaboration between machine learning (ML) models and doctors is efficient and free from bias. Considering the profound impact that a misdiagnosis can have on someone's life, we designed our system to maintain the highest levels of accuracy, ensuring our models achieve a success rate of over 90%. To increase confidence in our results, we implement a dual-check system where both the ML model and a doctor independently review the findings. In cases of a conflict in the analysis, a third doctor's opinion is taken. This system is replicated by the conflict resolution system by the Ministry of Health in Turkey. Furthermore, our ML model is designed to be explainable, allowing doctors to see the logic behind its decisions. This transparency ensures that doctors can use the model's insights as a supplementary tool rather than a decisive authority. MediXAI also extends its impact by offering guidelines for regular mammography screenings and spreading knowledge to increase societal awareness about breast cancer. By following these comprehensive measures, MediXAI aims to contribute significantly to public health.

6.2 Public Safety Considerations

It's crucial to address how MediXAI handles data privacy, especially in compliance with Turkey's Personal Data Protection Law (KVKK). We make sure that all personal information and medical images are stored securely using advanced encryption techniques, which helps protect them from being accessed without permission. Additionally, consent forms of the users are expected in the registration stage. If health data were to get exploited , it could cause serious issues for individuals' privacy. That's why we're always working to make our security even stronger, to keep everyone's information safe and meet all legal and ethical standards.

6.3 Public Welfare Considerations

For public welfare, it's important to consider ethical and legal laws about keeping people's information private. Additionally, our app helps increase the public welfare by various factors: People do not waste time or effort to get healthcare. It brings benefits like letting doctors work from anywhere, which makes public hospitals less busy. Thanks to the convenience and increased awareness from our app, more people are likely to get checked each year. We also expect fewer mistakes in diagnosing illnesses, which is great for everyone's health.

6.4 Global Considerations

Data security laws of the countries and the awareness level should be considered while operating globally. However at the beginning we are planning to only operate in Turkey.

6.5 Cultural Considerations

Only cultural consideration is about the awareness about the mammography checks people have to know the breast cancer risks so that they use our application. We are presenting informative pages to increase the awareness in the application.

6.6 Social Considerations

Social factors do not have a significant effect on our application.

6.7 Environmental Considerations

MediXI doesn't harm or be constrained by the environment. In fact, it helps by allowing doctors to work from anywhere and patients to get checked without having to travel. This means less travel and less energy used which is good for our planet.

6.8 Economic Considerations

For economic considerations, it's important to make MediXAI affordable for everyone while also making sure doctors are paid well. We want the app to cost less than what people usually pay for these health checks, so more people can use it. At the same time, we need to pay the doctors in our system a fair amount for their work. This balance helps us keep the app helpful and widely used, and it also respects the hard work of the doctors. This approach is really important for MediXAI to be successful and helpful in the long run.

	Effect Level	Effect
Public health	10	MediXAI has to be accurate, make sure doctors and Ml systems are collaborating smoothly.
Public safety	8	MediXAI has to follow KVKK and ethical rules.
Public welfare	5	MediXAI has to be affordable
Global Considerations	2	MediXAI has to follow the privacy laws(currently not planed to operate globally)
Cultural Considerations	3	MediXAI has to be affordable
Social Considerations	1	No evident effect.
Environmental Considerations	1	No evident effect.
Economic Considerations	6	MediXAI has to be affordable

Table 1: Factors that can affect analysis and design.

7 Teamwork Details

7.1 Contributing and Functioning Effectively on the Team

All significant decisions are made collectively following thorough discussion. Every project member participated in report-writing sessions. Each team member was readily available and approachable for communication. The work distribution is the following:

Ahmet Arda: He is responsible for the machine learning part with Mustafa. He was involved in dataset finding, model implementation, explainability parts. Additionally involved in meetings with people in the healthcare field.
Mahmut Furkan: He is responsible for the web application for the doctors. He created some functionalities specific to the doctors like report generation. He integrated the already prepared

frontend with the Firebase and made some modifications as needed. Contributes highly to the software part.
Mustafa: He is responsible for the machine learning part with Ahmet. He was involved in dataset finding, model implementation, explainability parts etc.
Omar: He is responsible for the patients' mobile application. He worked on improving the UI and integrating it with Firebase. He is also working on the AWS side of the project by setting up and integrating services like Simple Email Service (SES), Lex chat, Cloud privacy and security, S3, and EC2. He developed the encryption scheme used to secure medical data and is currently working on machine learning security.
Ömer Kağan: He was responsible for the frontend part of both mobile and web apps. He also worked on XAI model deployment on cloud (AWS SageMaker) and send inference to the database. He also helps machine learning team as needed.

7.2 Helping Creating a Collaborative and Inclusive Environment

In the project, we emphasized a collaborative and inclusive work environment by strategically dividing tasks based on each team member's interests, skills, and past experiences. This ensured everyone was engaged and contributed to areas where they were most passionate and knowledgeable. Our project was separated into software development and machine learning teams, fostering specialization while encouraging cross-team collaboration for shared insights and problem-solving.Weekly discussion meetings were crucial, especially in the early stages, providing a platform for everyone to freely express opinions and ideas, thereby fostering a culture of open communication and mutual respect. Thus, we shared works fairly and everyone's ideas were considered. In our project, we made sure everyone could talk openly and support one another. This helped us create a positive space where every team member felt important.

7.3 Taking Lead Role and Sharing Leadership on the Team

In our team, we spread out leadership roles and we did not have a particular leader in the team. This means anyone can lead when it's their moment to express himself, it gives us all a chance to show our leadership skills. We all took charge of necessary situations, acted responsibly, and helped each other. With this mindset, every team member really owns their part of the project and we all give effort. This way of sharing leadership doesn't just bring us closer as a team, it also makes us more effective in achieving our goals as a team.

8 Glossary

Term	Definition
XAI	Explainable Artificial Intelligence
HIPAA	Health Insurance Portability and Accountability Act
KETEM	Cancer Early Diagnosis, Screening and Education Center
BI-RADS	Breast Imaging-Reporting and Data System
API	Application Programming Interface
UI	User Interface
PBKDF	Password-Based Key Derivation Function

9 References

[1] Breast cancer facts and statistics 2023. Available at: https://www.breastcancer.org/facts-statistics (Accessed: 28 October 2023).

[2] Prof.Dr.M. (2018) Meme Kanseri belirtileri, Teşhisi Ve Tedavi yöntemleri: Anadolu Sağlık Merkezi, Anadolu Sağlık. Available at: https://www.anadolusaglik.org/blog/meme-kanseri (Accessed: 28 October 2023).

[3] Harbeck, N. *et al.* (2019) 'Breast cancer,' *Nature Reviews Disease Primers*, 5(1). doi:10.1038/s41572-019-0111-2.

[4] e-Nabız| T.C. Sağlık Bakanlığı. Available at: https://enabiz.gov.tr/ (Accessed: 27 October 2023).

[5] Machine Learning Service - Amazon SageMaker| AWS. Available at: https://aws.amazon.com/sagemaker/ (Accessed: 14 March 2024).

[6] KPSPublic Web Service. Available at: https://tckimlik.nvi.gov.tr/Service/KPSPublic.asmx?op=TCKimlikNoDogrula (Accessed: 27 October 2023).

[7] General Data Protection Regulation. Available at: https://gdpr-info.eu/ (Accessed: 27 October 2023).

[8] Kişisel Verileri Koruma Kanunu. Available at: https://www.kvkk.gov.tr/ (Accessed: 27 October 2023).

[9] Firestore | Firebase. Available at: https://firebase.google.com/docs/firestore (Accessed: 14 March 2024).

[10] Stripe | Financial Infrastructure for the Internet. Available at: https://stripe.com/ (Accessed: 14 March 2024).

[11] AI Chatbot - Amazon Lex - AWS. Available at: https://aws.amazon.com/lex/ (Accessed: 15 March 2024).

[12] DNS Service - Amazon Route 53 - AWS. Available at: https://aws.amazon.com/route53/ (Accessed: 15 March 2024).

[13] Bulk Cloud Email Service - Amazon Simple Email Service - AWS. Available at: https://aws.amazon.com/ses/ (Accessed: 15 March 2024).

[14] Cloud Object Storage - Amazon S3 - AWS. Available at: https://aws.amazon.com/s3/ (Accessed: 15 March 2024).

[15] What is Amazon EC2? - Amazon Elastic Compute Cloud. Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html (Accessed: 15 March 2024).

[16] Load Balancer - Elastic Load Balancing (ELB) - AWS. Available at: https://aws.amazon.com/elasticloadbalancing/ (Accessed: 15 March 2024).

[17] What is Elastic Load Balancing? - Elastic Load Balancing. https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html (Accessed: 15 March 2024).

[18] Control traffic to your AWS resources using security groups - Amazon Virtual Private Cloud. Available at: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html (Accessed: 15 March 2024).