BABEŞ–BOLYAI UNIVERSITY OF CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

Diploma Thesis

E-me, the secure document-handling application

Abstract

EZ AZ OLDAL NEM RÉSZE A DOLGOZATNAK!

Ezt az angol kivonatot külön lapra kell nyomtatni és alá kell írni!

A DOLGOZATTAL EGYÜTT KELL BEADNI!

Kötelező befejezés:

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

2021 BALÁZS MÁRK

ADVISOR:
ASSIST PROF. DR. KOLUMBÁN SÁNDOR

BABEŞ–BOLYAI UNIVERSITY OF CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

Diploma Thesis

E-me, the secure document-handling application

ADVISOR:
ASSIST PROF. DR. KOLUMBÁN SÁNDOR

STUDENT:
BALÁZS MÁRK

2021

UNIVERSITATEA BABEŞ–BOLYAI, CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

Lucrare de licenţă

E-me, aplicația de gestionare securizată a documentelor

CONDUCĂTOR ŞTIINŢIFIC:
LECTOR DR. KOLUMBÁN SÁNDOR

ABSOLVENT:
BALÁZS MÁRK

2021

BABEŞ–BOLYAI TUDOMÁNYEGYETEM KOLOZSVÁR
MATEMATIKA ÉS INFORMATIKA KAR
INFORMATIKA SZAK

Szakdolgozat

E-me, a biztonságos dokumentum-kezelő alkalmazás

TÉMAVEZETŐ:

DR. KOLUMBÁN SÁNDOR,
EGYETEMI ADJUNKTUS

SZERZŐ:

BALÁZS MÁRK

2021

Contents

1. Chapter

Introduction

1.1 About E-me

A brief introduction of the application, 1-2 pages.

- general introduction
- why somebody would use this app
- the main features/selling points of the app

1.2 Similar applications in the field

A list of similar applications, their advantages and disadvantages, comparisons, 2-3 pages.

1. Google Docs
   - create and edit documents
   - sync between multiple devices
   - view PDF docs/presentations
   - upload and manage files
2. Documents to Go
   - edit/view/create word, excel, PowerPoint docs
   - supports password protection
   - Google Docs support
   - bi-directional sync
3. SecureSafe

   - secure file and data storage
   - double encryption
   - secure AES-256 and RSA-2048 encryption
   - https
   - MFA with SMS
   - send files up to 2GB to recipients

4. Quick Office Pro
   - create/edit/share Microsoft Office files
   - offline file access

1.3 Comparison

1.3.1 Similarities

- Similarly to the **SecureSafe** app, E-me uses AES-256 symmetric encryption standard to securely store and transfer documents.
- E-me allows users to upload their PDF documents.
- Users have quick and secure access to their data and files.

1.3.2 Differences

- E-me only supports PDF documents.
- E-me uses End-to-End Encryption over HTTPS to communicate with the clients.
- Users are able to **generate** their PDF docs using predefined templates filled out with their personal data.
- All PDF documents **(generated or uploaded)** will be verified for authenticity by the system administrators (later government) and will receive a digital signature to mark their authenticity.
- Authorities can request access to users' documents in order to verify their identity or other personal information (this access is temporary).

1.4 Summary

Describes the structure of the following document, 1 page.

2. Chapter

User documentation

Summary: *In this chapter E-me is described from a user point of view.*

The building blocks of E-me are pages. The primary role of these pages is allowing the user to manage their PDF documents that were created through the application. This management includes the ability to request a new document, view already owned documents, make modifications to them, delete the ones that are obsolete or contain incorrect data, or even share them with other devices using a QR code.

The secondary role is collecting the data from the user that is necessary for generating their documents. This data is then encrypted, processed and later on added to the documents that the user requests.

2.1 Introductory Pages

For an unauthenticated user there are 3 pages available in the application: Greeting, Login and Registration pages. Upon opening the application, the user is confronted with the Greeting page which allows them to navigate to the Login and Registration pages using the dedicated buttons.

The Registration page consists of five text fields. Each of these fields are required in order to create a new user, however only two of them are needed for authentication: Login name and Password. The restrictions for these fields are the following:

- The Email and Login Name fields should be unique.
- The Password and Confirm password fields should be identical.
- The Email field should be a valid email address.

Upon successful registration, the application navigates the user to the Login Page in order for them to authenticate themselves using the Login Name and Password they entered. If the authentication was successful, the main shell appears where three tabs can be seen: My Documents, Request Document and Personal Info.

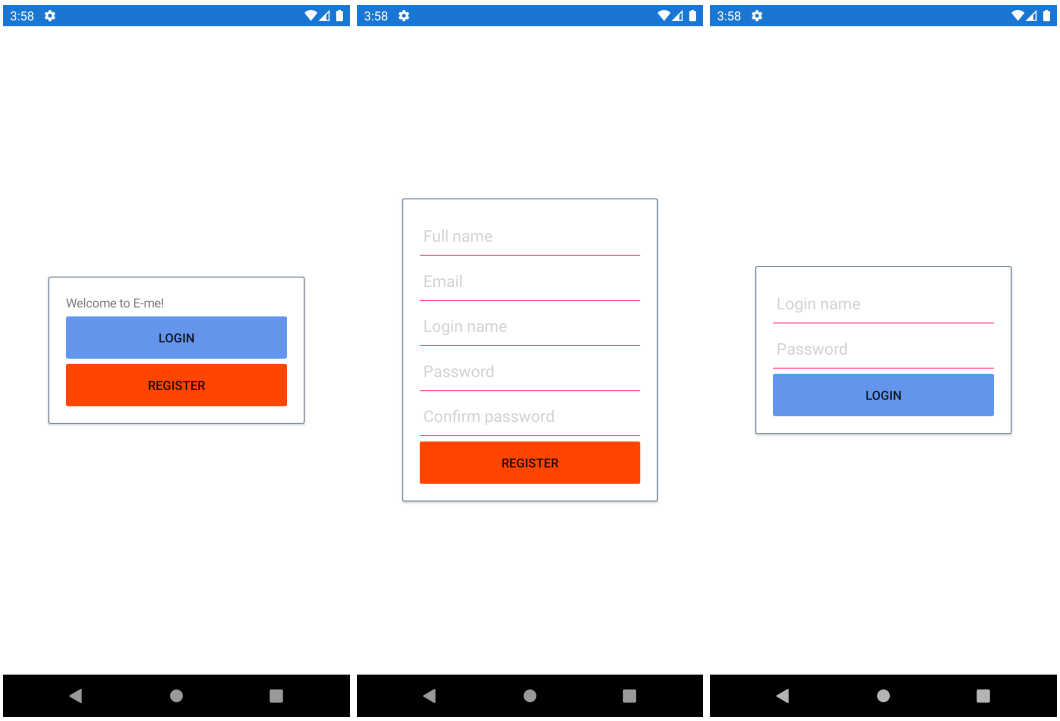

Figure 2.1: Introductory pages (Greeting, Registration, Login)

2.2 My Documents

The My Documents page consists of two main parts: the list of documents and the "Scan QR code" button. The list of documents allows the user to visualize what types of documents they own. On each document two actions can be performed: Share and Remove.

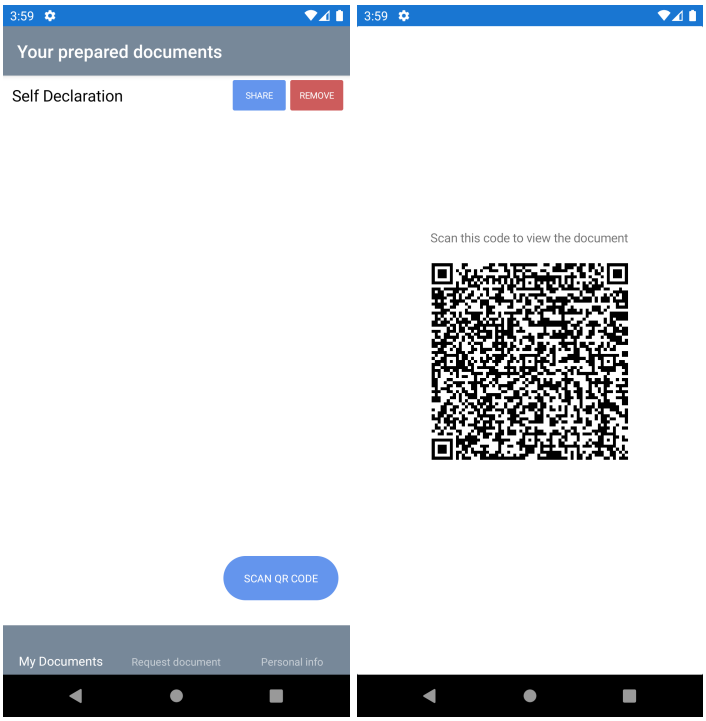

Figure 2.2: Documents and Share Document pages

Tapping the Remove button irreversibly deletes the document from the list. After a successful removal the user is able to request a new document of the same type. If the document template was modified since the previous request or the user modified their personal information, the resulting document may be different than the previously deleted one.

The Share button allows the user to safely transfer their selected document to a different device. Upon tapping the button, the application generates a unique code for the document which is then displayed on the screen via a QR code.

The code can be read using the "Scan QR code" button. Tapping this button will attempt to open the device's main camera in order to scan the code. If this feature is accessed for the first time, a prompt appears asking for the user's permission to use the camera. If the permission is granted, the application will open the camera app. Upon successfully reading a QR code generated by E-me, the selected document will appear on the screen (see Document Viewer).

2.3 Requesting a Document

The Request Document page consists of a list of document types that can be acquired by the user. The list only contains types that are not yet owned by the corresponding user. If the user acquires one of these document types, it will disappear from the list (it will be listed on the My Documents page instead).

Combined with the My Documents page, these two lists contain every document type that can be managed through the application.

Figure 2.3: Request Document and Personal Information pages

The user can request a document by simply tapping on an item from the list. The requested document will be generated and automatically displayed on the screen (see Documents Page).

2.4 Personal Information

This page contains several fields containing the personal details of the user. Neither of these fields are required, however if no information is provided about the user, the application will not be able to automatically fill the requested documents. The information provided by the user on this page will be stored in an encrypted form.

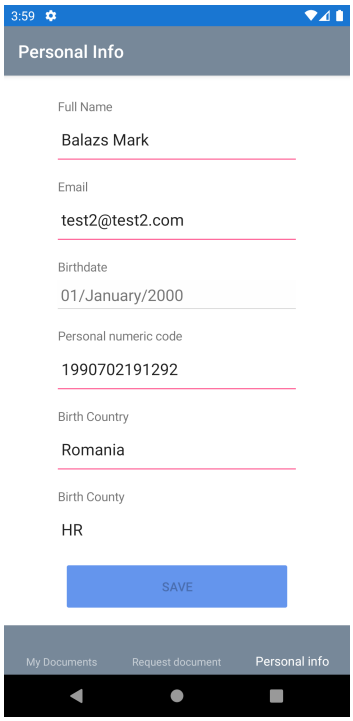

Figure 2.4: Request Document and Personal Information pages

Upon requesting a document, E-me attempts to match the fields of the document with the information of the user and fill them out respectively. Fields that require information that is not provided by the user will be left blank and can be filled manually on the Document Viewer.

2.5 Document Viewer

On this page a PDF Viewer can be seen which allows the user to inspect or even edit their documents. This viewer supports text searching, bookmarking, zooming, printing and/or saving a PDF to the local storage of the mobile device. The contents of the PDF are generated by the application, however if the document contains fields that could not be matched with any user data, the user is able to fill them out manually. This page can be closed by pressing the Back button.

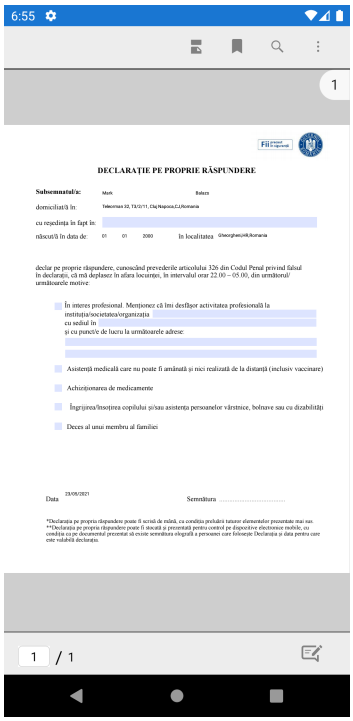

Figure 2.5: Document Viewer

3. Chapter

Technical details

3.1 Architecture

E-me follows a commonly used N-tier architecture with three main parts: data, application (backend) and presentation (frontend) layers. Each of these tiers can be broken down into layers that are defined by their responsibilities within the application. This tier-based architectural approach adds modularity to the application which results in a low cost of change when compared to a single-tier structure.

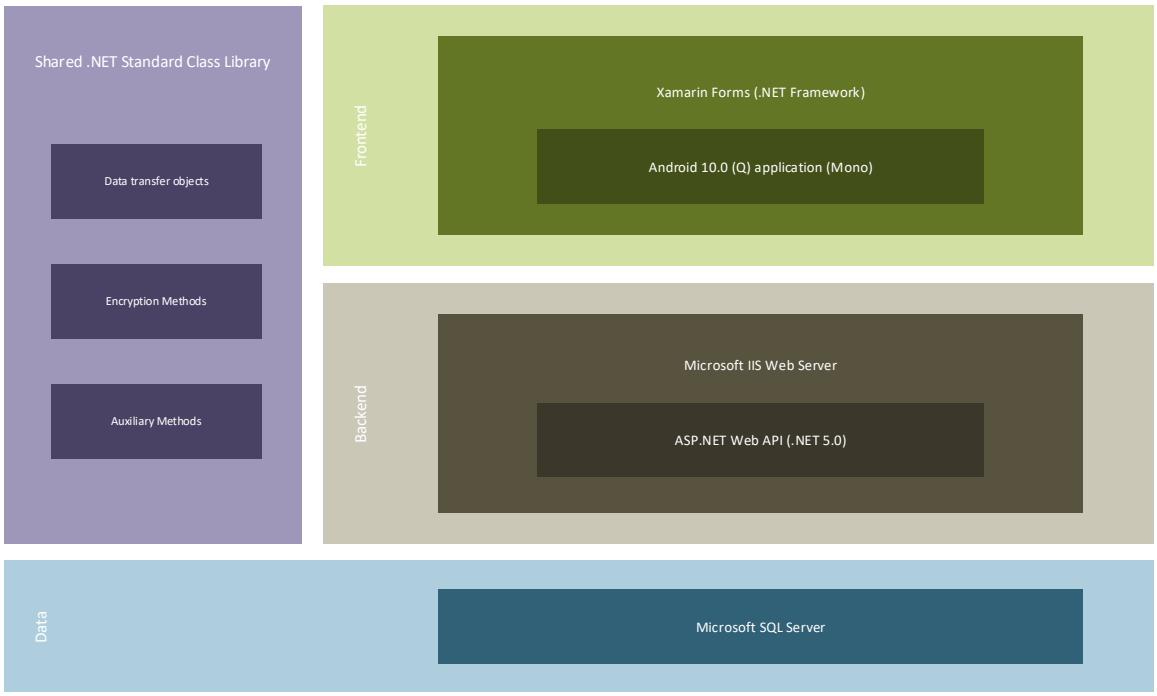

Figure 3.1: General architecture

Along with the low cost of change, the independence of the layers allows for efficient future expansion of the application by means of multiple frontend platforms (ex. web applications, desktop applications), cloud storage/services and additional Web API's that can easily be integrated into the existing application. This, combined with the high compatibility of the .NET 5.0, provides a high level of scalability and maintainability for the application.

The application and presentation layers share a common class library that contains communication-related models and auxiliary methods. This library allows both layers to benefit from the cross-platform nature of .NET 5.0, speeding up the process of development and ensuring there are no discrepancies betweeen the layers in terms of encryption and communication.

3.1.1 Application layer / Backend

The architecture of the application layer has a similar design approach to the general architecture. Consisting of 4 layers, the backend follows the single-responsibility principle in it's core. Because of it's vertical structure, each layer is dependent on one single layer that is directly

below it, providing a high level of maintainability.

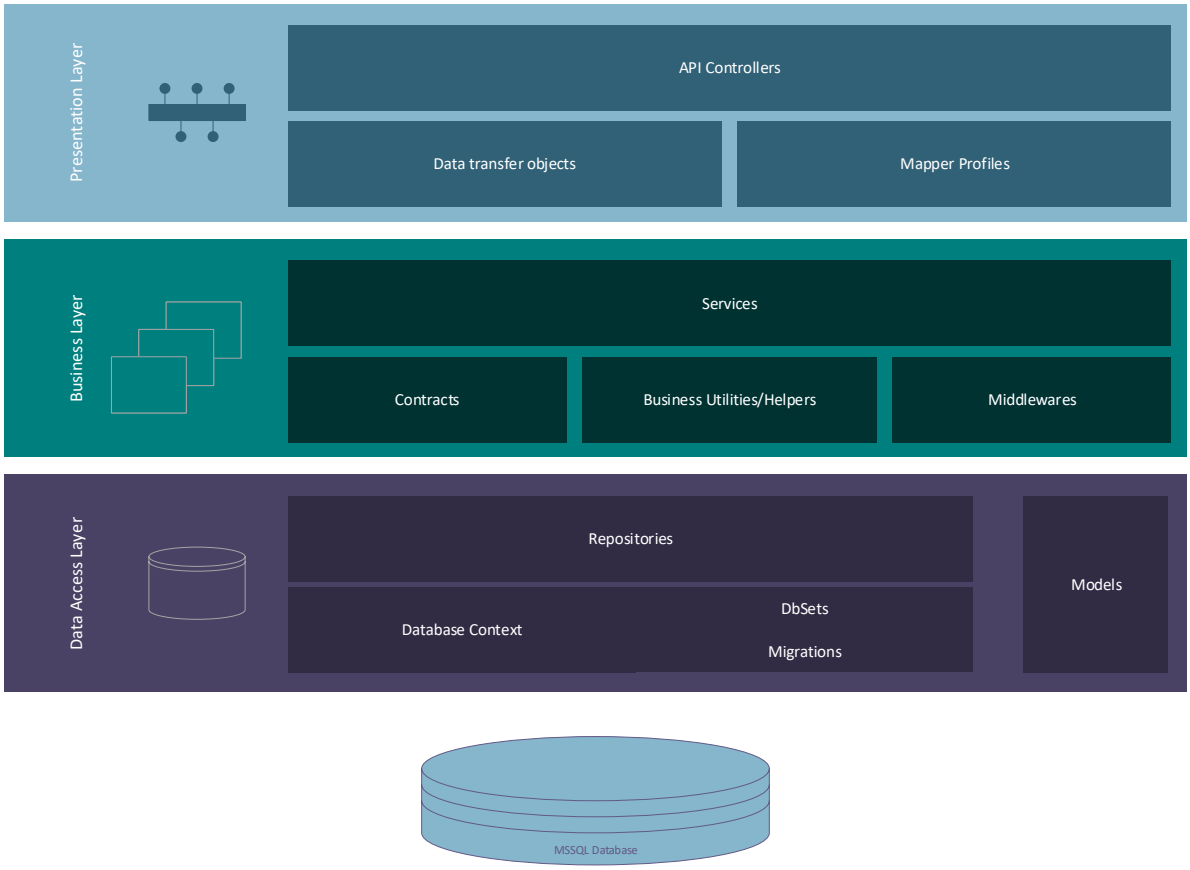

Figure 3.2: Backend architecture

Due to this abstract layout, each layer can independently be replaced or updated. This aspect enables the utilization of external and/or third-party services and components without damaging the integrity of the application.

The API structure of the presentation layer allows for a wide variety of applications or even services in which the backend can be used: desktop applications, WPF, web servers and more. This characteristic has major role in preserving the flexibility of the application layer.

Another component being responsible for the independece and reusability of the backend is the Data Access Layer. The role of this layer is to provide the data requested by the Business Layer. By reason of abstraction, this layer is independent of the technology of the data source. This enables the utilization of a wide range of data sources, including relational (SQL) and non-relational (NoSQL) databases, cloud services and/or APIs.

Entity relations in the application layer

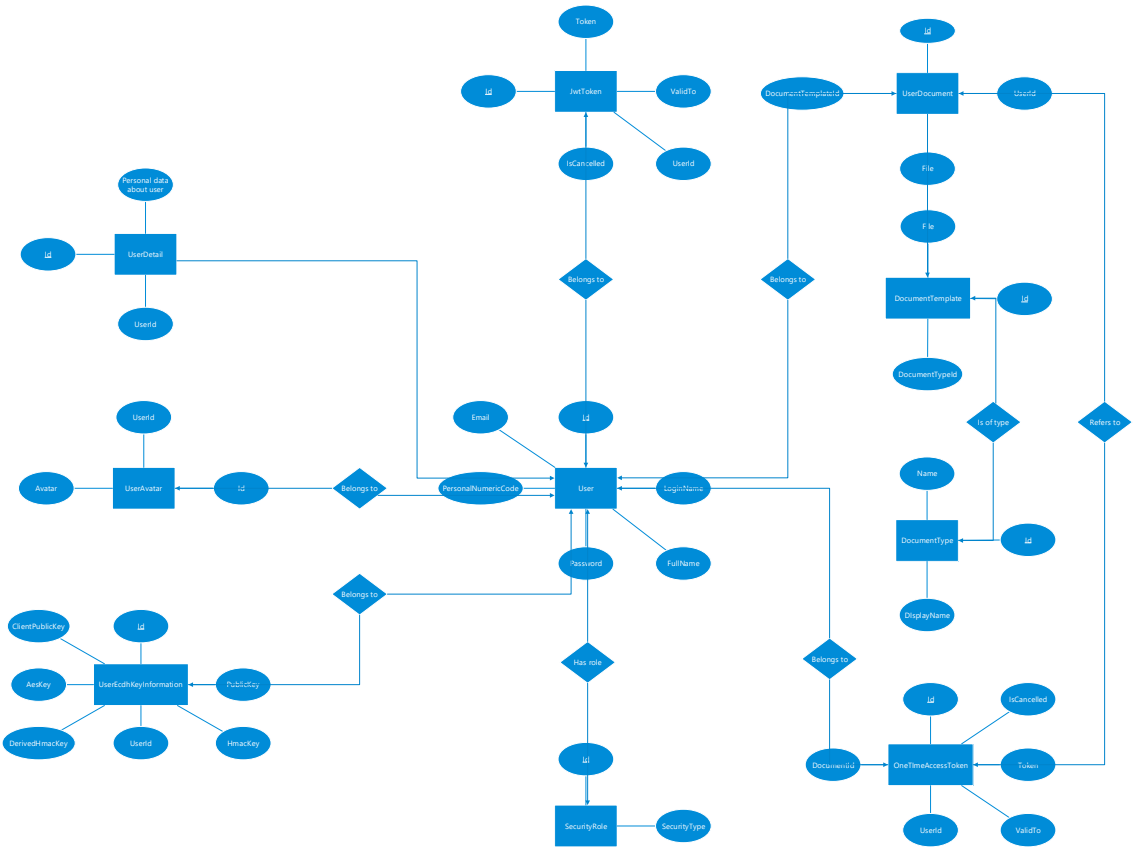

Figure 3.3: ER diagram

3.1.2 Presentation layer / Frontend

The frontend of the application consists of three major layers that separate the user interface from the business logic and the data sources. This separation allows for easy horizontal expansion and quick feature development.

The Presentation or UI Layer contains the visuals of the application which are separated into independent pages, however it is also responsible for receiving user events and connecting them with the underlying services. Each page contains it's own event listeners and separate View Model that is responsible for making use of the Business Layer, which has a similar structure and role to the backend's Business Layer: data processing and calculations.

The main components of the Data Layer are Data Stores. These stores are functionally similar to the repositories found in the Backend of the application, although here the data is retrieved using the endpoints of the backend.

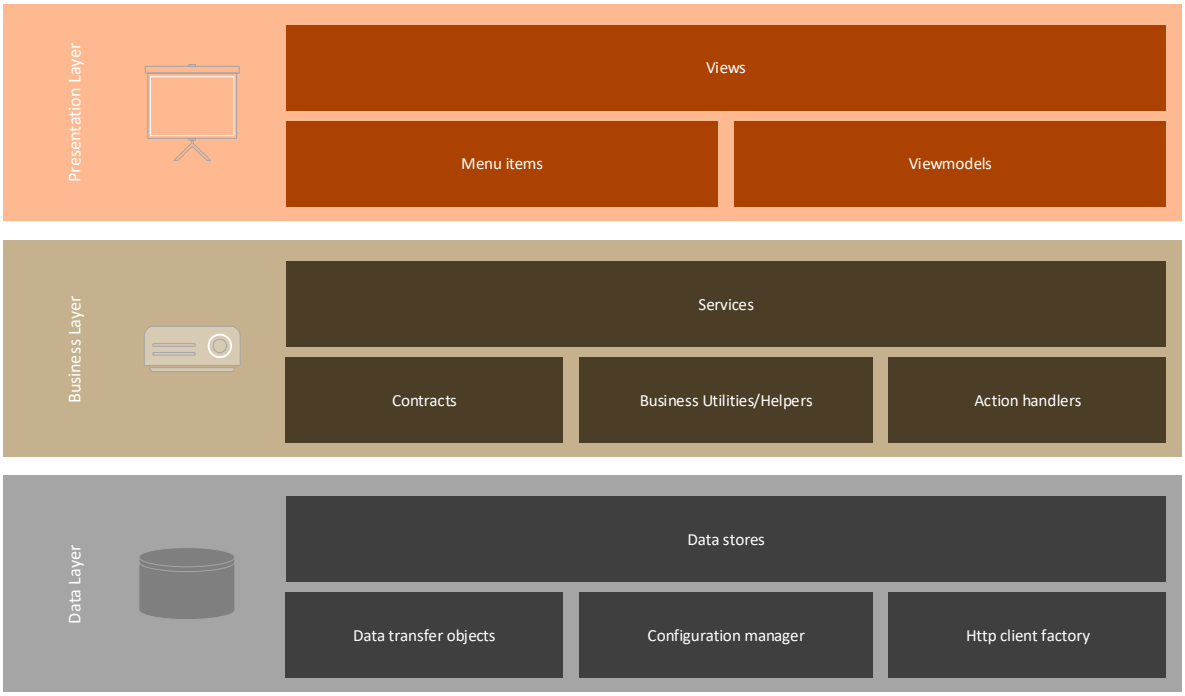

Figure 3.4: Frontend architecture

3.2 Technologies

E-me makes great use of the cross-platform nature of .NET, meaning both the frontend and the backend of the application are implemented using the same programming language: C# 9. The flexibility that is provided by the language and the wide variety of frameworks that make use of it allows for a cleaner codebase and a simpler project structure when compared to a multilingual application design.

3.2.1 Backend

In terms of frameworks, the backend relies on .NET 5.0. Being the next major release of .NET Core following 3.1, this open-source cross-platform software framework unifies a wide variety of frameworks .NET has to offer, including Windows Forms, ASP.NET Core, Azure and more. This unification made the framework a perfect candidate for the backend of an application like E-me, since it can be used on a wide variety of platforms and ensures long-term support.

E-me uses Internet Information Services (Microsoft IIS), which is an extensible web server software, as it's backend server. It allows the application to communicate through HTTPS and provides it's separate development-time SSL certificate in order to secure the transport layer. With the *Development-time IIS support* feature, the web server allows *hot reloading*, which speeds up the development and debugging processes.

The main constituent of the backend itself is the ASP.NET Core web framework. Using it's built-in IoC, this framework allows for a high level of abstraction in order to build well-structured enterprise level applications using dependency injection. This framework is responsible for registering and configuring any other dependencies that may appear in the application backend.

Alongside with ASP.NET Core, one of the main components of the backend is the Entity Framework Core, which is the base of the *Data Access Layer*. In terms of building and maintaining the data access layer, E-me uses the code-first approach through *EF Migrations*. This approach provides flexibility in case of a potential database change and/or creation, but more importantly it allows the entire structure of the entities to be managed based on the models and datasets present in the code.

E-me heavily relies on encryption and security, which demands the utiliation of advanced cryptographic libraries. For this purpose, the application uses the *Windows CNG (Cryptographic Next Generation) API*, which enables exchanging documents and other data in a secure environment. In terms of cryptography, this API provides a full implementation of the *Elliptic-curve Diffie-Hellman* key agreement protocol, alongside with many other cryptographic protocols and algorithms such as RSA, AES (Advanced Encryption Standard) or DSA (Digital Signature Algorithm) and hash functions like SHA-2 or MD5.

The documents handled by the application are created and managed by the *Telerik Document Processing Core* framework, which provides flexibility and efficiency in terms of document management. It enables processing the most commonly used text, PDF and spreadsheet file formats, allowing the application to create, import and export documents without relying on other external dependencies.

3.2.2 Frontend

Being a .NET-based application, E-me uses Microsoft's open-source mobile UI framework, *Xamarin Forms* as a base for it's frontend. This library allows developing native mobile applications on multiple platforms, such as iOS, Android or UWP using *.NET Standard*. The multi-platform capability allows for a shared codebase and logic across all applications, which speed up the development process immensely. Xamarin uses an XML-based interface designer in order to create the visuals for the application. Alongside with the commonly used Xamarin

UI components, E-me uses the *Telerik UI for Xamarin, GoogleVision API and Syncfusion* UI libraries for additional components and controls in it's user interface.

Although the frameworks used by the different layers of the application vary, these can have a shared codebase in the form of a *.NET Standard class library*, which can be used by both the backend and the frontend, allowing for closely integrated layers.

3.3 Implementation

3.4 Security

Here I describe the Diffie-Hellman key exchange and the used encryption techniques in more detail, 3-4 pages.

- data-layer security
  - using built-in EF Data Encryption with AES256
- transport-layer security (TLS)
  - https
  - JWT auth and auth verification
  - protected and unprotected endpoints
- End-To-End encryption
  - Elliptic Curve Diffie-Hellman key derivation - open-source implementation
  - encryption of documents
  - hash-based message authentication (HMAC)

4. Chapter

Results and evaluation

Summary: *In this chapter I describe decisions I made, difficulties I faced during development and the quality of my code.*

4.1 Metrics

In this section I will evaluate some of the algorithms used in the application, test coverage, code analysis. 4 pages

- chart about the duration of the encryption (time versus file size)
- service-level test coverage
- code metrics
    - maintainability
    - cyclomatic complexity
    - average depth of inheritance
    - average class coupling
- system requirements
    - server
    - mobile

4.2 Decisions

Here I describe decisions I made about what technologies to use, what I considered using and how they can be replaced with other ones. 3-4 pages

- backend

    - Java
    - Python
  - frontend
    - Kotlin
    - Java
    - React Native
  - why I chose C# and .NET instead of native languages
  - Data storage
    - MySQL
    - MongoDB
    - Oracle
    - Firebase
  - Auth technologies
    - Cookie-based auth
    - Multi-factor auth
    - Biometric auth

4.3 Obstacles and difficulties

In this section I contemplate about different parts of the applications that were problematic to develop. 3 pages

- HTTPS on Android
  - difficulties connecting to the server on HTTPS
  - certificate issues
- Windows CNG not being implemented in Mono
  - switching to the open-source ECDH implementation
- accessing resources within the Android secure storage (icons, config files etc.)

4.4 Possibilities

Here I describe possible features, future upgrades for the app. 1-2 pages

- Adding biometric authentication
  - fingerprint
  - face recognition
- ML based form categorization
  - categorize unknown fields based on user inputs
- Requirement-tree for documents
- digital signatures
- notifications (expired/soon-to-be expired documents)
- Administration application
  - validating data
  - granting permission for document release
  - preparing templates
- IOS release

4.5 Retrospective

In this section I review the development process and describe what would I do differently and why. 1-2 pages