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

Diploma Thesis

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

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

Aplicație 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

Biztonságos dokumentum-kezelő alkalmazás

Témavezető:

dr. Kolumbán Sándor,
egyetemi adjunktus

Szerző:

Balázs Márk

2021

Contents

1. Chapter

Introduction

1.1 About the application

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 Similarities 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 Contrast

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

Basics

Summary: *In this chapter I describe the application from a user point of view.*

2.1 General outlook

Here I describe the visuals of the application with images, 2-3 pages.

- screenshots about the outlook/pages of the app with description
  - Login
  - Registration
  - My Documents
  - Request Document
  - Personal Details
  - Share document (QR code)
- basic information about the pages
  - static content
  - data-related content

2.2 Feature-showcase

Here I talk about the features of the application, 2-3 pages

- a more detailed description about the features of the app
  - what actions can a user make
    - buttons
    - selecting list items

- ∗ describing use-cases
    - · requesting a document
    - · sharing a document
    - · scanning a QR code to obtain a document

3. Chapter

Implementation

Summary: *This is the summary of the chapter where I describe the general form-factors of the application from a technological standpoint.*

3.1 Technologies

Here I list the technologies used for building the application with logos, descriptions for each, 6-7 pages.

- Backend
    - .NET 5
    - Entity Framework Core 5
        - Code-first
        - Microsoft SQL Server
        - additional Data Encryption layer
    - NSwag
    - Serilog
    - AutoMapper
    - Newtonsoft Json
    - Windows CNG (Cryptographic Next Generation) API
- Frontend
    - Xamarin Forms
    - Telerik UI for Xamarin
    - Telerik Document Processing Core
    - Syncfusion Xamarin PDF viewer
    - GoogleVision API - BarcodeScanner XF implementation

3.2 Architecture

In this section I describe the architecture with multiple diagrams, 4-5 pages.

- General 3-tier architecture
    - diagram
    - general description
- Backend multi-tier architecture
    - diagram
    - general description
- Model UML diagram
    - diagram
    - general description
- Frontend multi-tier architecture
    - diagram
    - general description

3.3 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