CryptoCamGuard-Elevating Image Security App

Project Proposal

Supervisor

Sir Saifullah Adnan

Co-Supervisor

Submitted by

SARDAR NAZEER
{2630-2021}

ALI SHER SIAL
{2201-2021}

Department of Computing,
Hamdard University, Karachi.
27 – June - 2024

Table of Contents

1. Introduction

CryptoCamGuard is a mobile application developed to provide users with a secure environment to capture, encrypt, store, and manage personal images. In an age where digital privacy is constantly threatened, this app ensures that sensitive photos are protected using AES-256 encryption, stored securely, and accessible only through authentication mechanisms.

2. Objective

The primary objective of CryptoCamGuard is to provide users with a secure platform for capturing, encrypting, and storing personal images, thereby ensuring their privacy and protection from unauthorized access. The application aims to streamline the photo management process while empowering users to maintain full control over their digital memories.

KEY FEATURES

1. Advanced Encryption: Secures images upon capture with cutting-edge encryption algorithms.
2. User-Friendly Interface: Intuitive design for easy navigation by all users.
3. Secure Image Storage: Safely stores images within the app, away from traditional galleries.
4. Authentication Mechanisms: Ensures only authorized users can access their images.
5. Optimized Performance: Fast and reliable handling of photos for efficient uploads and downloads.

3. Problem Description

In the digital era, mobile devices have become an integral part of everyday life, serving as repositories for personal memories and sensitive information. Among the most frequently stored data are images, which often include private moments, family photos, and

confidential documents. Unfortunately, the security measures in place for managing these images on mobile devices are inadequate.

MANUAL PROCUREMENT

Manual procurement refers to the traditional process of sourcing goods and services through manual paperwork, phone calls, or in-person meetings. This method is often inefficient, time consuming, and prone to errors due to a lack of digital automation.

LIMITED TRANSPARENCY

In conventional image storage solutions, users often face limited transparency regarding who can access their photos and how their data is managed. This lack of visibility increases the risk of unauthorized access and data misuse, as users are unaware of potential privacy breaches.

INEFFICIENCY

Traditional image management methods often involve manual processes that are time consuming and prone to errors, leading to delays in accessing or organizing photos. This inefficiency can frustrate users, making it difficult to manage and retrieve their personal images quickly and securely.

4. Methodology

- Agile development with iterative sprints.
- React Native for cross-platform frontend.
- .NET Core backend with JWT authentication.
- AES-256 for image encryption.
- PostgreSQL database to manage metadata.
- Manual and automated testing using Postman, Jest, and Mocha.

Feasibility Study:

- **Technical:** Uses well-supported tools and frameworks.
- **Economic:** Open-source stack with no licensing cost.
- **Operational:** User-friendly interface ensures wide usability.

Risks Involved:

- Key management for encrypted data.
- App permissions and camera access limitations.
- Device-level storage limitations.
- User forgets credentials or key access.

5. Flow Chart:

Big Diagram:

6. Project Scope

- Develop a **mobile app** that securely captures and stores personal images.
- Implement **advanced encryption** (e.g., AES) to protect images from unauthorized access.
- Create an **easy-to-use interface** so users can navigate the app without technical knowledge.
- Ensure **performance optimization** so the app works quickly and smoothly.
- Allow access to encrypted images **only through the app** using proper authentication.

Not in scope:

- Encryption of non-image files.
- No image sharing or file transfer features

It's important to note that the project scope may be refined and adjusted during the development process based on the project's time frame, available resources, and client requirements.

7. Resource Requirement

- Devices: Android phones for testing
- Development tools: Android Studio, Visual Studio, GitHub
- Libraries: React Native Camera, AES packages

---

8. Solution Application Areas

- Personal photo vaults
- Legal and medical image confidentiality
- Journalists and researchers in sensitive environments

9. Tools/Technology

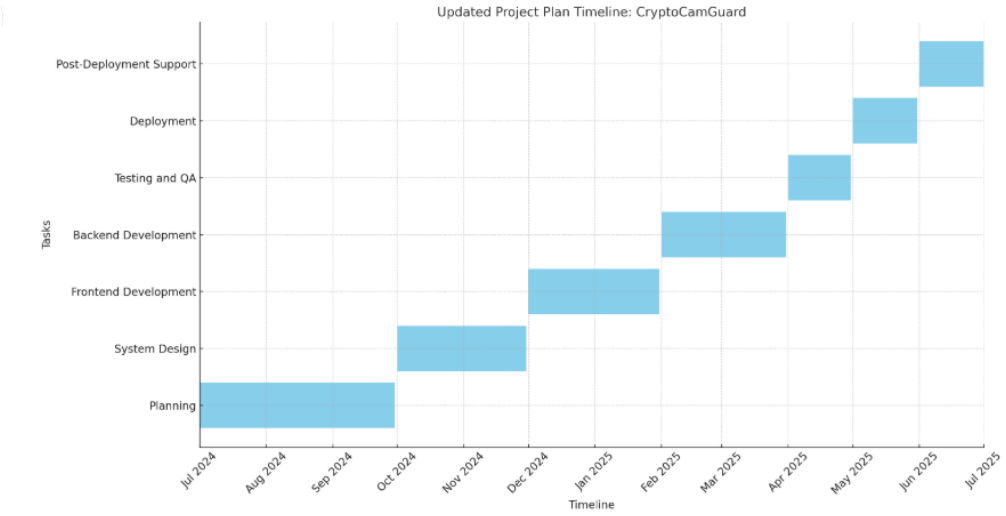
React Native, .NET Core, PostgreSQL, JavaScript, Postman, GitHub

10. Software Requirement

- OS: Windows/macOS/Linux
- IDEs: VS Code, Visual Studio
- Platforms: GitHub, Firebase (optional)

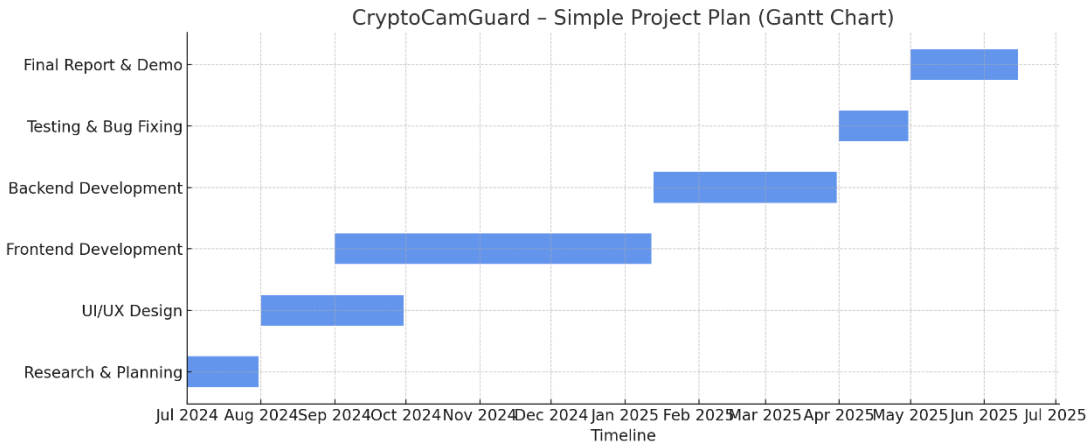
Mention RACI Matrix for the project

RACI MATRIX



10. Milestones

GANTT CHART



11. References:

• **Tresorit** uses encryption technology to protect your files end-to-end, meaning no third-party or service provider can access them from upload to download.Overall, Tresorit is a safe and secure way to store and share your sensitive data. For more information, visit their website directly. https://tresorit.com/

• **PixelKnot** is an Android app that allows you to hide messages within images and share them securely through any messaging platform. Its main purpose is to keep your images secure and aid in sharing them confidentially. PixelKnot utilizes steganography, meaning it hides your messages within the pixels of images, making them appear normal but containing hidden information. Pixel Knot