Functional Document – Question Secure

Matas Aleksiejus – C00239630

Contents

Introduction

This document outlines the scope of the entire project, it's deliverables as outlined in the original spec, the technologies that will be used, functionality, use case diagrams with use cases.

Project Scope

Main Goal

There are two main parts in this project. The first part will focus on developing an android application that will allow a user to securely store their used security questions. It will also allow the user to generate a secure security answers to use in many different websites and for many different accounts. The generated security questions will take user input as well as input relevant to the website they are making the account for i.e. banking, utilities, gaming websites to generate a secure and unique answer that cannot be recreated. The second part of this project will focus on a browser plugin that will take data from a website at the time of account creation and generate a QR code that can be scanned by the android application to assist with generating a secure and unique answer to the security question on the account creation page.

Target Market

This application will be aimed at an average end user that create a lot of different accounts. This application targets the two biggest downfalls of security questions, memorability and security. By generating secure and unique answers I hope to solve the guessability issue of common questions while being able to store in the application aims to negate the memorability issue as outlined in my research document. The application will be marketed with the end users security as it's forefront and ensure that creating accounts on websites that still use security questions can be safe as long as this application is used.

Project Objectives

Main objectives of this application are as follows:

- To provide the end user with an easy to use interface for generating and storing security questions and answers.

- To ensure end user safety with the ability to generate secure and unique answers that cannot be recreated.

- To securely store previously used security questions and answers as well as store newly generated ones.

- To provide the end user with a browser plugin that makes it as easy as possible to create secure answers.

Technologies

Below are outlined the technologies that I plan to use to implement this project. I have selected the following due to the research I've carried out into secure android application development and my previous knowledge of cryptography.

Main Application – Question Secure

Below are outline the technologies I will utilise in the implementation of my main Android application. I have carried out research into possible solutions and have decided that my vision for this application will be best carried out with the following technologies:

Java

Java is a high-level, object orientated programming language. Java syntax is very similar to C++ and C however Java is build with the run anywhere mindset to allow it to run on any device regardless of underlying hardware as long as it has the required runtime environment. This is achieved by Java running on a virtual machine that translates code to java bytecode instead of machine code which is then translated into machine code corresponding to the device by the interpreter. Java is a major language when it comes to android application development and it has a lot of built in support for security. I plan to build the application and process the information provided by the user in Java.

- ### Java Cryptography Architecture (JCA)
  The java Cryptography Architecture is a framework used for working with cryptography when using the java programming language. The JCA has references to a lot of different security provides and when implemented it chooses what provider to use depending on what security function i.e. message digest and what algorithm i.e. SHA-256 you choose. This is achieved by having all provider implementations conform to well-defined interfaces (Oracle, n.d.). The JCA is supported by android studio and includes important classes that can be used in this project such as SecretKeyFactory, SecureRandom, KeySpec and Cipher.

- Zxing library
  Zxing library ("zebra crossing") is an open-source, multi-fromat,1D/2D barcode image processing library implemented in Java, while it contains ports to other languages I will be using it for my android application which will be developed in java and will be using it for scanning QR codes that are generated by the additional browser extension (Pankaj, n.d.).

- BiometricPrompt
  This is a class in android that manages a system-provided biometric prompt. This will show a system-provided authentication prompt, using one of the devices supported biometric authentication modules(fingerprint, face, iris). On a device where all types of available authentication are enrolled the device will allow the user to choose which they prefer. This class is crucial to my application as it allows the user to authenticate to the application without requiring the use of any other credentials thus not requiring the storage of any other private information such as email, username, password, etc (Android, 2021).

Cryptography

For encrypting and decrypting user data i.e. answers to security questions I plan to use the Advanced Encryption Standard (AES) with Cipher Block Chaining (CBC) mode. There are a couple of things necessary for this first is input which will be a security question either generated by the application or an existing answer the user has input. A secret key which will be derived from a function and an initialization vector (IV).

- SecureRandom

I plan to use the secure random class from JCA to generate a unique, secure and unrepeatable passphrases to use as security answers to security questions for the user. The secure answers will be an amalgamation of user data relating to the question, input data relevant to the account or website and a random value to greatly reduce the guessability of the password. The JCA SecureRandom class produces cryptographically strong random values using a cryptographically strong pseudo-random number generator. By default the class uses the SHA1PRNG algorithm which is also supported by android studio.

- Android Key Store

The Android keystore is a specific key provider for Android. It allows the application to generate and store a key in the Android keystore and keep it there securely. Keys store in the android keystore cannot be accessed programmatically and can only be referenced via alias. Therefore you cannot get the actual key bytes and view them. The Android key store is hardware backed and any cryptographical functions that use a key referenced in the android key store take place in the trusted execution environment rather than in the application runtime (Android, 2022).

Android Studio

Android studio is the official IDE for Google's android operating system. It replaced Eclipse Android Development Tools and was designed specifically for the design of android applications. The latest stable version of Android studio includes a lot of features such as an android virtual environment for testing and compiling and drag and drop elements for common applications modules. It supports Java, C++ and Kotlin with even more available languages can be added with extensions. I'll be using Android studio to develop the application in Java.

Browser Extension – Question Secure

As above I have outline my technologies to be used for the main application below I will describe the technologies I will be using to implement the accompanying browser extension.

HTML

HTML or Hypertext Markup Langauage is a language used in front end web development. It allows for documents and images to be displayed on webpages and it is often accompanied by CSS and JavaScript. HTML documents are what browsers interpret and display to webpages and so browser extensions and apps are also created in HTML. HTML uses elements or tags as building blocks to display and format text, images and links for webpages. However HTML lacks and processing or computing capabilities.

JavaScript

For computing capabilities I will use JavaScript to add functionality to my browser extension. JavaScript is a programming language that is one of the core technologies that make up web development. Over 97% of all websites use some form of JavaScript. (W3Techs, 2022) I need to use JavaScript in this project to allow for my web extension to generate QR codes to allow the application which is on an android device to generate secure answers to secure questions easier.

- ### QRCode.js

  For QR code generation I plan to use QRCode.js. This is a JavaScript library for making QR codes, it is cross browser compatible and it also has no dependencies which makes it easy to generate QR codes from input quickly (davidshimjs, n.d.).

Database Design

The database for the program will be local due to time constraints and ease of configuration. I plan to use SQLite as the database structure as it's easily supported with android development as I discovered in my research. SQLite creates a .db file in the local storage of the application under databases. This allows me to access the files easier and ensure that the data base structure is correct.

Since the application will only store a combination of the website where the security question is utilised, the question that the user has chosen and the generated answer the database will only contain one table. This is possible because the user will authenticate using the devices biometric system therefore no credential storage is required by the application. The database structure is as follows:

Database name: 'questions.db'

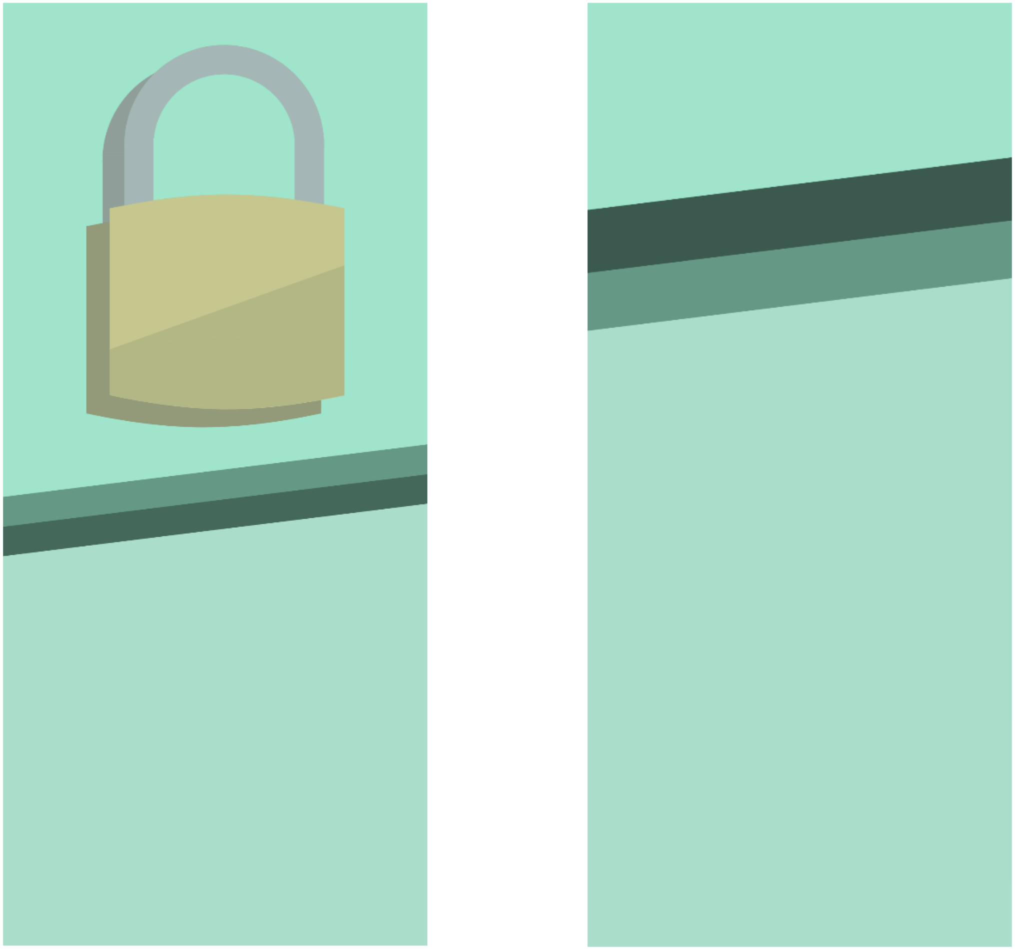
Table name: 'questions'
field 1: 'id', integer, primary key, autoincrement not null
field 2: 'website', text
field 3: 'question', text, encrypted
field 4: 'answer', text, encrypted

I will use DB Browser(SQLite) to browse the database while working on my application as you can use this application to see table structure and contents of a .db file. Since encrypted data is stored in byte format and the database table accepts text all encrypted fields will be Base64 encoded after encryption. This will also assist with the viewability of entries in the database during development.

Visual Design

Since the main application for this project is an android application the UI has a part to play in the flow and usability of the project. The main colour I sued for the project was green due to it being a colour I really liked at the time mostly drawing inspiration from the colour scheme of the protagonist Kamado Tanjiro of a Japanese animation Demon Slayer: Kimetsu no Yaiba. The aim of the User Interface for this application is to make the application simple to use and to be very self-intuitive with each button clearly labelled and explaining what to do.

Any visual parts of this application will be designed in Canva. Canva is an Australian graphic design platform that can be used to create all sorts of visual content (Perez, 2013). The platform is web based and offers basic configurations to a registered user on the free account version. The functionality increases greatly with a subscription but for my level knowledge of visual design and skill the free version is enough. Below are some things that I have already made for the design of the application:

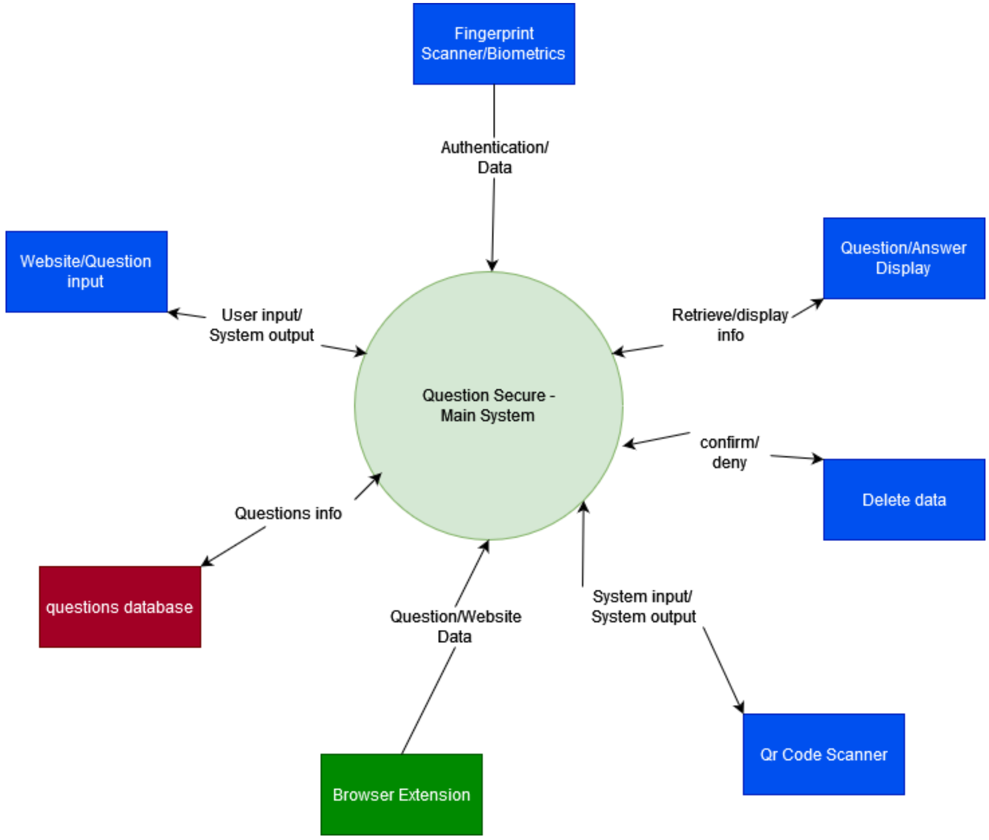


These are backgrounds for the title page or launcher page of the app and the background for the rest of the app, following a green colour scheme.

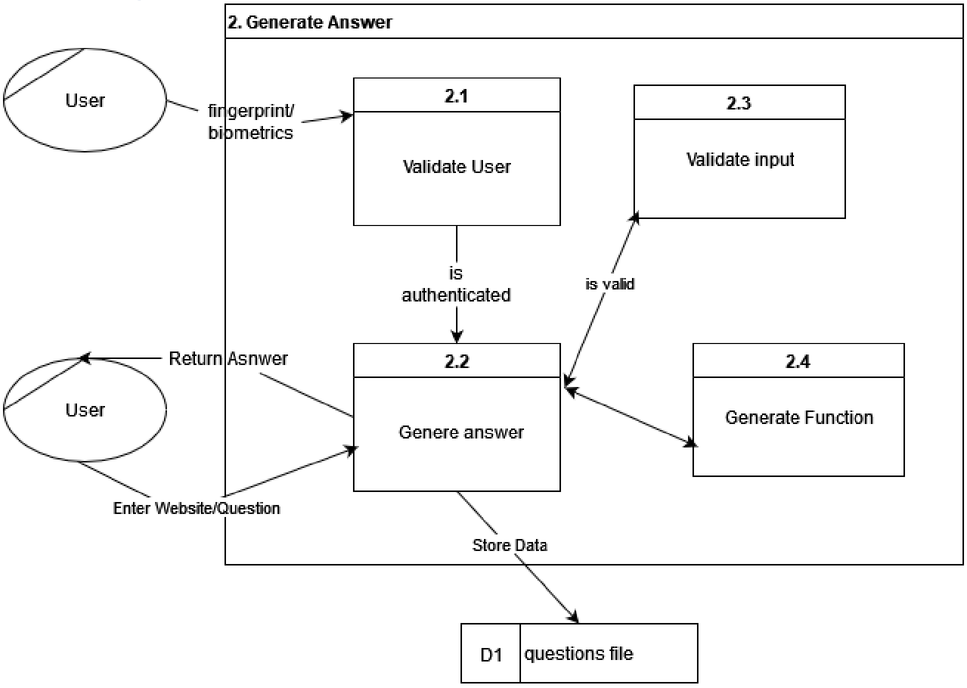
These are logos I designed in Canva. The name of the app Questions Secure is a transparent .png that can be added as an element to the XML files to ensure that it's displayed correctly on every page and will have no overlap with other elements.

Context Diagrams

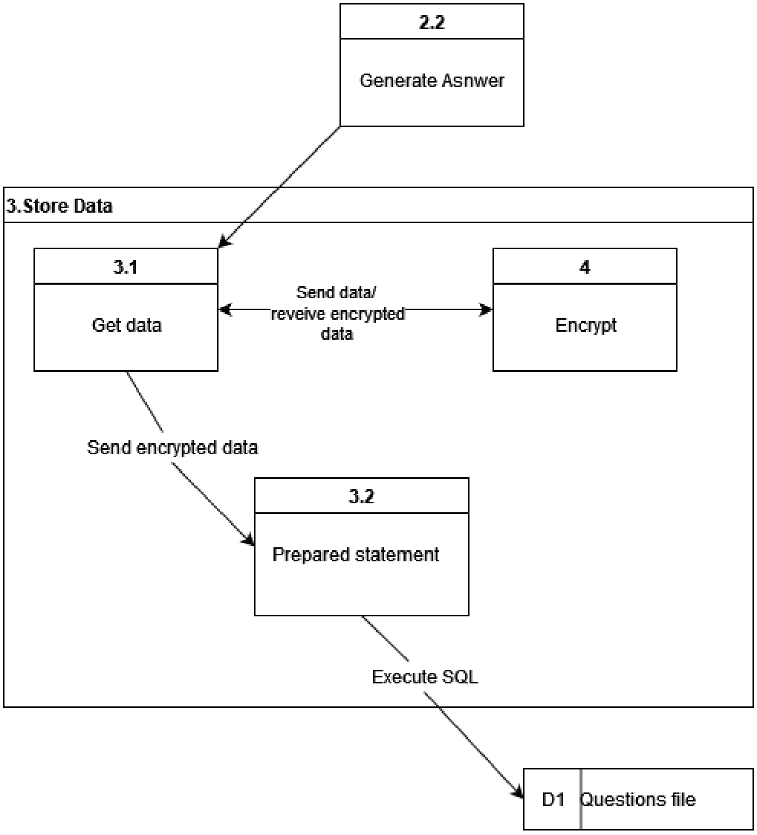
Context Diagram – Question Secure main system



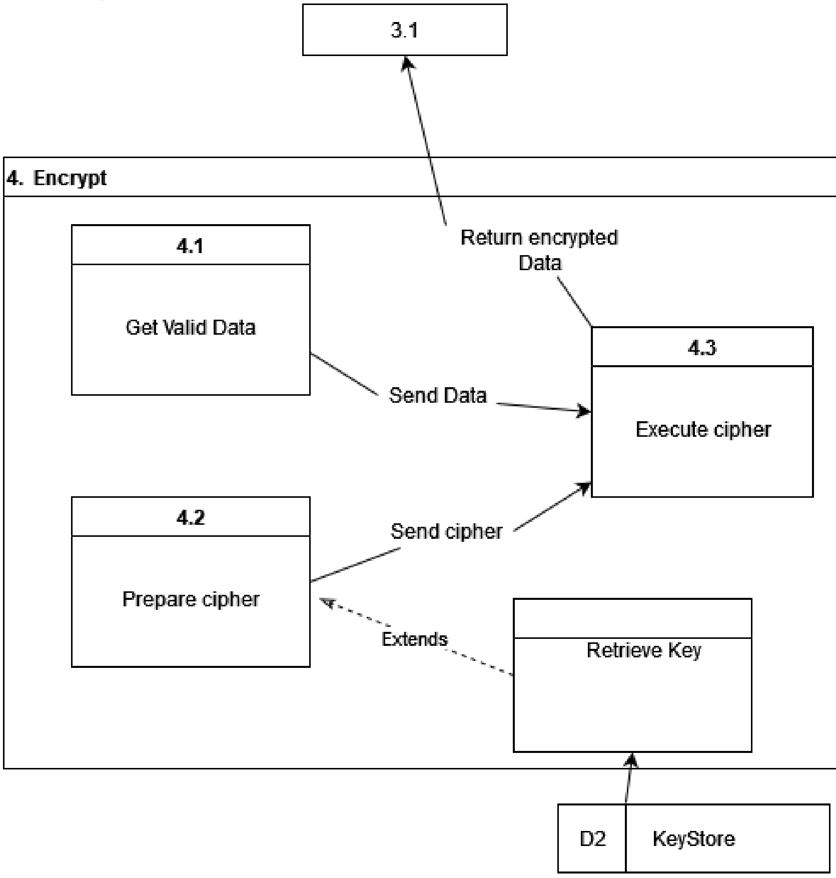
Data flow diagram – Generate Answer



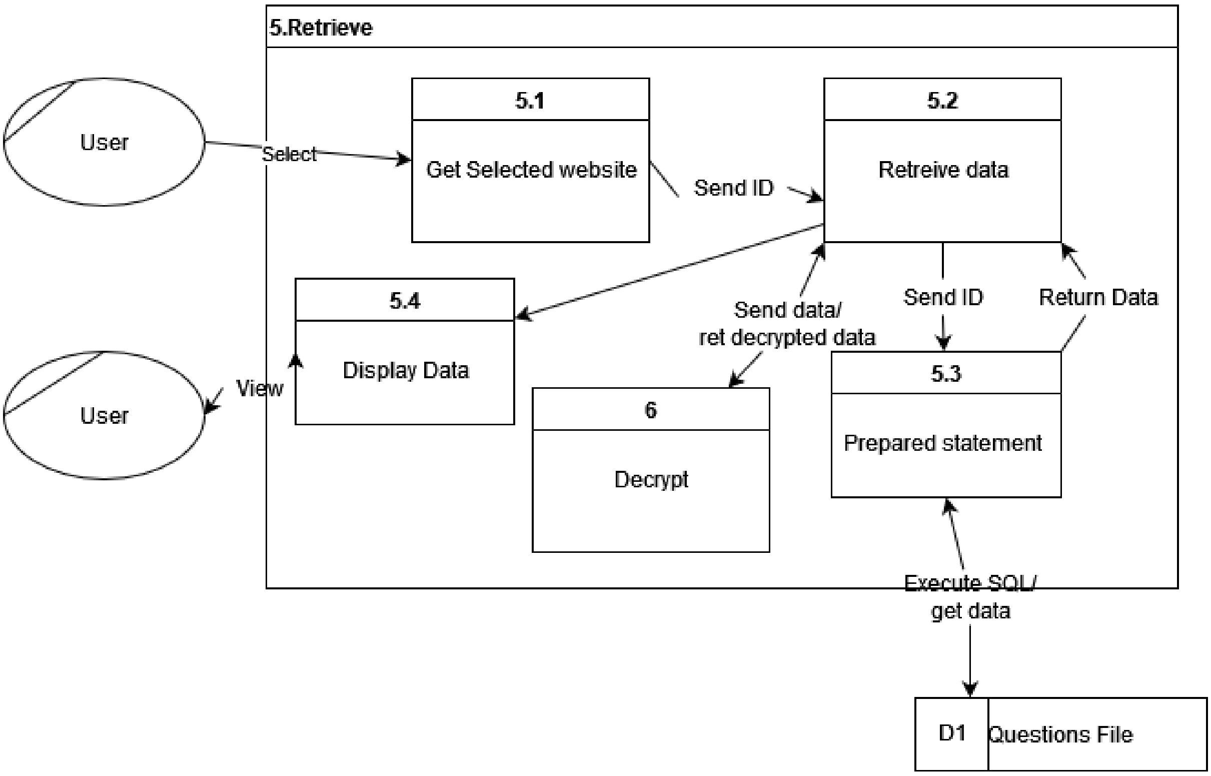
Data flow diagram – Store Answer



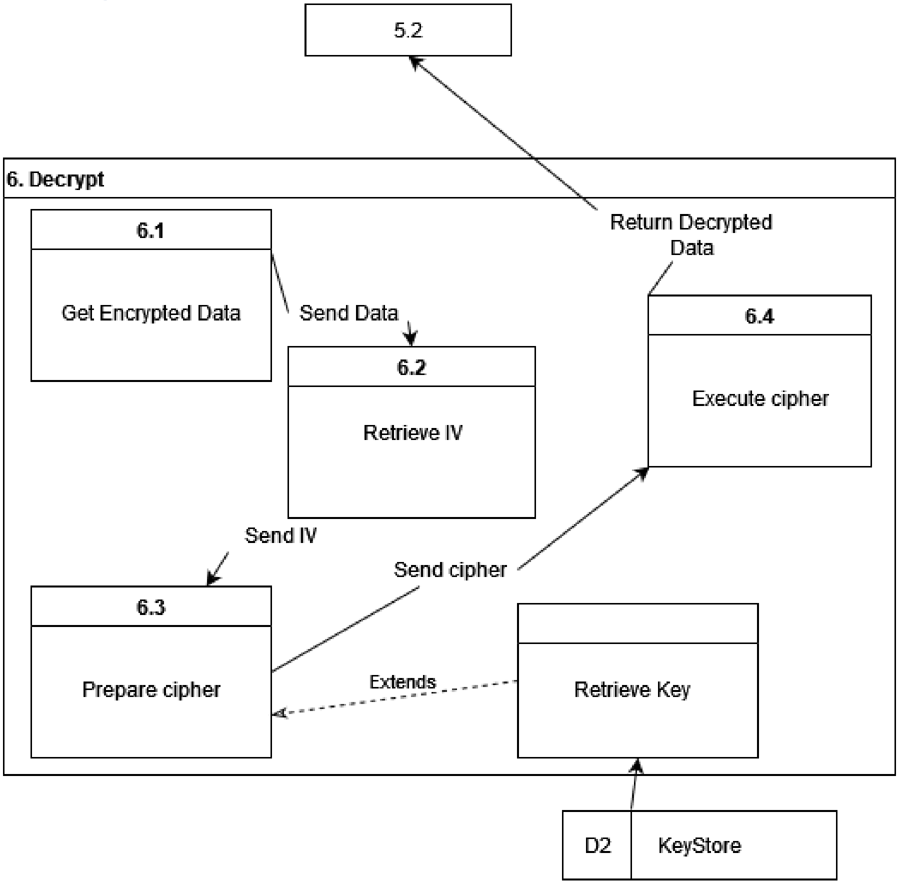
Data flow diagram – Encrypt



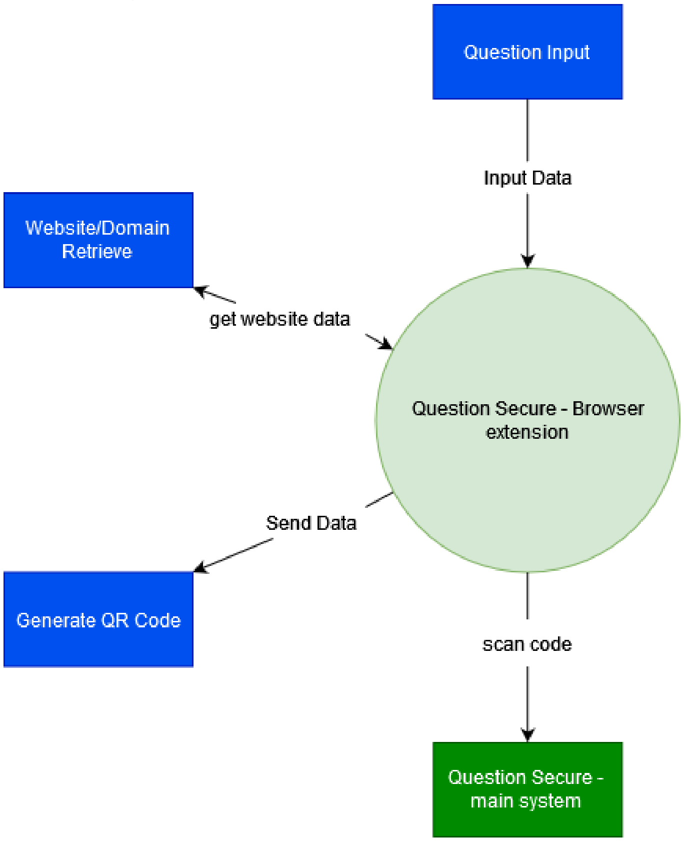
Data flow diagram – Retrieve



Data flow diagram – Decrypt



Context Diagram – Question Secure browser extension



Bibliography

Android. (2021, Febuary 24). *BiometricPrompt*. Retrieved from Android Developers: https://developer.android.google.cn/reference/androidx/biometric/BiometricPrompt

Android. (2022, April 22). *Android keystore system*. Retrieved from Android Developers: https://developer.android.com/training/articles/keystore

davidshimjs. (n.d.). *qrcode.js*. Retrieved from github.io: http://davidshimjs.github.io/qrcodejs/

Oracle. (n.d.). *Java Cryptography Architecture (JCA) Reference Guide*. Retrieved from docs.oracle.com: https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html#Introduction

Pankaj. (n.d.). *Java QR Code Generator – zxing example*. Retrieved from Journal Dev: https://www.journaldev.com/470/java-qr-code-generator-zxing-example

Perez, S. (2013, August 26). *Backed By $3 Million In Funding, Canva Launches A Graphic Design Platform Anyone Can Use*. Retrieved from Internet Archive: https://web.archive.org/web/20190902223751/https://techcrunch.com/2013/08/26/backed-by-3-million-in-funding-canva-launches-a-graphic-design-platform-anyone-can-use/

W3Techs. (2022, April 12). *Usage statistics of JavaScript as client-side programming language on websites*. Retrieved from w3techs.com: Usage statistics of JavaScript as client-side programming language on websites