iBlock

Person Verification Platform
Software Requirements Specification
PID 6

Version 1.0

iBlock

Mentor, Group members and contributions

Mentor: Dr. Thanuja Ambegoda

Index Number	**Name**	**Contribution** (Topics covered by each student)
190239A	K. G. Akila Induranga	● Overall description ● Functionality ● Reliability ● On-line User Documentation and Help System Requirements ● Wireframes for demo application ● Hardware Interfaces ● Database Requirements ● Licensing, Legal, Copyright, and Other Notices
190241X	M. I. M. Ishad	● Functionality ● Performance and Security ● Design Constraints ● Wireframes for mobile application ● Applicable standards
190242C	Kasun Isuranga	● Introduction ● Usability ● Supportability ● Hardware and software interfaces.

Revision History

Date	Version	Description	Author
18/09/2022	1.0	First Revision	K. G. Akila Induranga M. I. M. Ishad Kasun Isuranga

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This Software Requirements Specification document intends to provide a detailed description of the proposed system and might be useful for end users, designers, and third-party service providers.

This document covers the following sections.

- Scope of the project
- Overall Description of the project
- Functional and non-functional requirements
- Design Constraints
- Interfaces

1.2 Scope

The proposed system is a digital person verification platform using blockchain. The system aims to solve current issues associated with digital person verification. With blockchain, decentralized storage is provided, giving users full control over their credentials. A mobile/desktop application is used to store and manage credentials and an API is provided to integrate the system with third-party service providers.

1.3 Definitions, Acronyms, and Abbreviations

API – Application Programming Interface

HTTP – HyperText Transfer Protocol

HTML – HyperText Markup Language

JS – Javascript

1.4 Overview

This SRS document provides a high-level description of the iBlock digital person verification system and its subcomponents. The second section of the documents provides an overall description of the system. The third section provides detailed information about the requirements of the system. The last section provides supportive information to use the document.

2. Overall Description

iBlock being a software solution for person verification, is proposed to be used in digital platforms of any service provider such as banking systems, government offices, educational institutions, etc. to get verified by their service users in a single QR code scan. Implementing the mechanism as a smart contract on the **Ethereum Blockchain** facilitates a trustless environment for everyone to work with and for no one to tamper with. As the system requirements, identified as functional, and non-functional requirements, are described in detail in the next section, an overall description of a use case scenario of iBlock can be stated as follows.

Any person who wishes to use iBlock as his/her digital person verification platform is required to **download and install** the iBlock mobile application on his/her smart device and **set up an account** with the personal details, which are only **stored in his/her own digital wallet** due to privacy reasons. One time setting up an account is all you need to use iBlock, and you are ready to use it to **verify yourself** with any service provider who accepts iBlock as their personal verification platform. **Scan the QR code** that appears on the service provider's screen, **approve the request** on your iBlock app, and you are verified. For keeping the user informed, all the past **request approvals can be monitored** on the iBlock app as well.

For the service providers, we are developing a dedicated **API and documentation** that can be easily implemented in their current system, without affecting the system. A **Demo application** is provided for service providers, as well as the users to get familiarized with how to use iBlock in-person verification, as this is a leading-edge technological software solution.

Because blockchain technology is still evolving, the decentralized applications deployed on blockchains still have considerable limitations. iBlock tries to address as many of the issues arising through those limitations, but it still is limited in recovering the lost accounts and revoking the approved requests. The development team will be working to address them as well, in the maintenance stage.

General knowledge about installing a mobile application, signing up, and scanning QR codes are expected from the potential users and such skills are assumed to have in the users.

3. Specific Requirements

3.1 Functionality

3.1.1. Functional requirements of the Mobile application

3.1.1.1. Register to the Platform.

Lets a new user with the mobile application installed on his/her smart device sign up to the platform using the email.

3.1.1.2. Recover the account.

Allows the user to recover his previously created account using the provided secret recovery phrase when a circumstance of losing his/her device.

3.1.1.3. Get the identity verified.

Users registered to the iBlock platform can verify their identity by scanning the QR code and approving the request of the service provider to access their personal information.

3.1.1.4. Track the history of identity information shared services.

Users can monitor the history of sharing their personal information with different service providers, including both approvals and rejections.

3.1.1.5. Add more identity information fields added.

It lets the user add his own custom personal information fields to his/her wallet linked to the iBlock platform other than the required fields the user is supposed to fill at the signup phase.

3.1.2. Functional requirements of the integration library integrated platform

3.1.2.1. Generate a QR code to get the identity verified by the users.

Service providers who are supposed to use the iBlock platform in their web-based solutions can use the integration API provided by the iBlock project to generate a QR code for the user to scan and verify his identity.

3.1.2.2. Verify the user's identity.

Once a user scans and approves sharing his personal information with the service provider, the service provider gets to autofill the information on the forms that previously had to be filled by the user manually.

3.2 Usability

3.2.1 User Interfaces

As the system will be used by inexperienced users of electronic devices, the user interfaces need to be as simple and understandable as possible. Providing tooltips might improve the user experience.

3.2.2 Time consumption for various tasks

To set up the iBlock Application Program, the estimated time requirement is 15 – 30 minutes. For initiating and completing a verification, the estimated time requirement is 10 – 25 minutes.

3.2.3 Training times for normal and power users

Setting up iBlock Application Program

For a normal user to become familiar with setting up the iBlock Application Program, about two hours of training time is required. For a power user for the same procedure, about one hour of training is required.

Initiating verification process

The training time required to initiate a verification process also depends on the third-party service provider's web interface. On average about half an hour of training time will be required by a normal user while about 15 minutes will be required by a power user.

3.3 Reliability

iBlock is being proposed to work with verifying people, required to be reliable at all possible times. However, trying to guarantee the decentralized nature comes with a cost in the reliability of the system, which has been identified as follows.

3.3.1 Availability

Since iBlock requires no centralized server, the maximum availability can be guaranteed. The Ethereum testnets such as Rinkeby and Gorli are robust enough to ensure consistent availability by running on enough nodes set up on the internet. However, because of the 'block time', the time period it would take to verify a transaction on the blockchain, being 15 seconds makes a maximum transaction process time of 15 seconds, where the details of that specific transaction are not available to other users.

3.3.2 Mean Time Between Failures (MTBF)

iBlock not having a single server makes it immune to server-side failures. If a transaction in the blockchain fails, it also can be informed to the transaction starter (wallet address) within the block time (15 seconds), and it does not cause any damage or failure to the system.

3.3.1 Mean Time to Repair (MTTR)

iBlock the mobile application itself needs no repair time other than the time to receive updates to the application in the maintenance stage. However, the API provided by the iBlock project to the service providers may sometimes result in the necessity of repairing/ debugging their already existing systems. We ensure customer support for them as well, and the meantime for improving service providers' systems is currently estimated as less than one working day. During that time, the service providers can follow their traditional procedure of verifying users.

3.3.2 Accuracy

Accuracy takes a major requirement in iBlock as it is identified as a person verification platform. To guarantee the accuracy in verifying people, iBlock is proposed to use 2 Factor Authentication in creating user accounts, and prompting the recovery phrase only at the account creation. This makes it difficult to restore a lost account, but such measures are identified as a priority to maintain the accuracy of the project.

3.3.3 Maximum Bugs or Defect Rate

iBlock mobile application, written in Dart using the Flutter framework, tends to be less error-prone and raises lesser bugs in both developments as well as in deployment.

However, the smart contract written in Solidity and deployed on an Ethereum testnet is expected to be more error-prone since it makes it difficult to debug decentralized application executions. iBlock estimates the maximum defect rate for their smart contract as 1 defect per function point.

3.4 Performance and Security

3.4.1. Performance

Following are the main performance requirements of the proposed system

3.4.1.1. Responsiveness

Refers to the ability to get an assigned task done without getting the user frustrated. 1.0s delay is the generally accepted threshold for any interactive application.

3.4.1.2. Capacity

The proposed platform should be able to accommodate a large number of users. Since the proposed platform is decentralized it can facilitate any number of users. But, the concurrent performance has a limitation on transacting identity data as the Ethereum blockchain has a performance limitation of - transaction per second. So, the number of verification requests served per second is limited.

3.4.1.3. Response Time

Refers to the amount of time the application will take to process a request and respond to the user. The transaction time is the bottleneck operation as the processing time for a transaction in the blockchain is high. However, the processing time is to be decreased by the recent Ethereum merge (transition of transaction validating mechanism from **proof of work** to **proof of stake**).

3.4.2. Security

Following are the main security requirements of the system.

- Prevent unauthorized access to the application.
- Every information sharing request needs to be approved by the user.
- The information should only be shared with the requested party.
- Data transfers should use HTTPS protocol

3.5 Supportability

3.5.1 Coding standards and naming conventions

When naming variables, underscores will be used to separate parts of the name.

Function, method or procedure names will start with a verb.

For class names, singular nouns will be used.

3.5.2 Maintenance

User feedback should be collected when possible to improve the system. As the system is highly sensitive due to security reasons, any security related issues should be prioritized. Scaling the provided services of the system should be done when possible depending on the changing nature of the technologies.

3.6 Design Constraints

3.6.1. Limitation in blockchain transactions

The system is proposed to be designed using the following open source programming language and frameworks.

- Flutter for the application
- Solidity for smart contracts
- Javascript for integration library

These programming languages and frameworks can be subject to limitations such as no access to device native features, and memory allocation constraints. Though according to the feasibility study research, these languages provide resources and interfaces which are required to implement all the functionalities proposed in the system. But, this may be a constraint on the supportability of future devices.

3.6.2. Limitations in blockchain transactions

In the Ethereum blockchain, transaction speed is limited to 10 - 15 transactions per second because it takes more time to verify a transaction. If a large number of users use the application concurrently, the verification process can take up to minutes. So, users feel the app is unresponsive. This is the main design limitation of the DApp.

3.6.3. Limited options to recover accounts

The proposed design of the platform is to ensure the user's information resides only with the user. So, no user information is stored in any centralized database. Though, the application doesn't provide a sign-in option. The only way to recover the account is using the provided recovery phrase which is provided only at the beginning.

3.7 On-line User Documentation and Help System Requirements

iBlock mobile application is designed to be easy to use even with very little/ limited knowledge of using mobile applications. Because of that, the online user documentation for the mobile app will be none, but starting guidance tips will be provided within the app when the user starts it for the first time.

iBlock web API is expected to provide a fully detailed online user documentation for the service providers and their system developers to get an understanding of how to embed iBlock person verification in their services and how the flow goes. Also, a Helpdesk will be available to open tickets for service providers in any need of developer assistance from the iBlock team.

3.8 Purchased Components

No component/ service will be purchased in this cycle of development. However, if any service provider needs to have a more robust performance with more distributed nodes, the smart contract can be deployed on the official Ethereum blockchain making every transaction required to be paid in ether.

3.9 Interfaces

3.9.1. User Interfaces

3.9.1.1. Wireframes for Mobile Application

Prototype of the mobile application: https://bit.ly/3xdMi33

Below are some featured screens in the prototype.

- Loading Screen - This screen appears while the app is initializing.

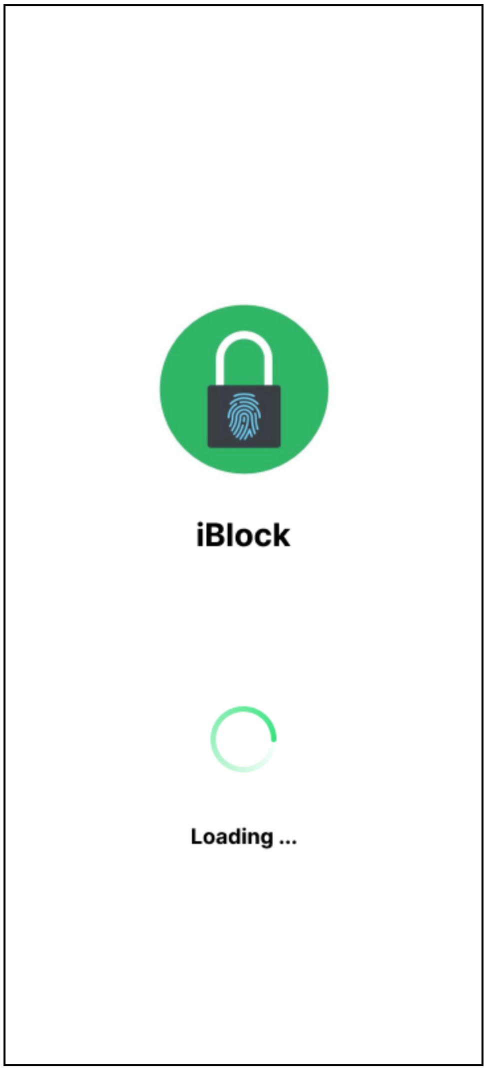
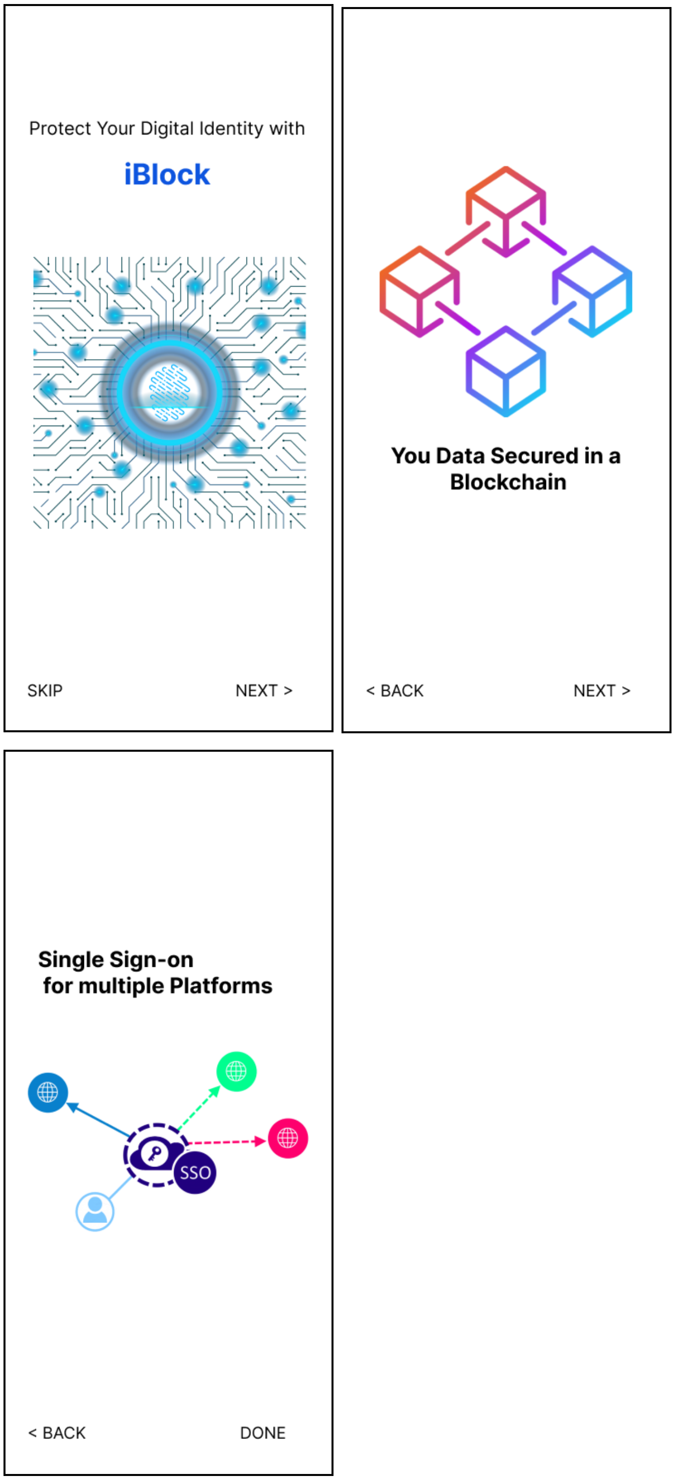


Figure 1. Loading Screen of iBlock Mobile Application

- Welcome Tour Pages - These pages appear when a user opens the app for the first time after installation or an update.



Protect Your Digital Identity with

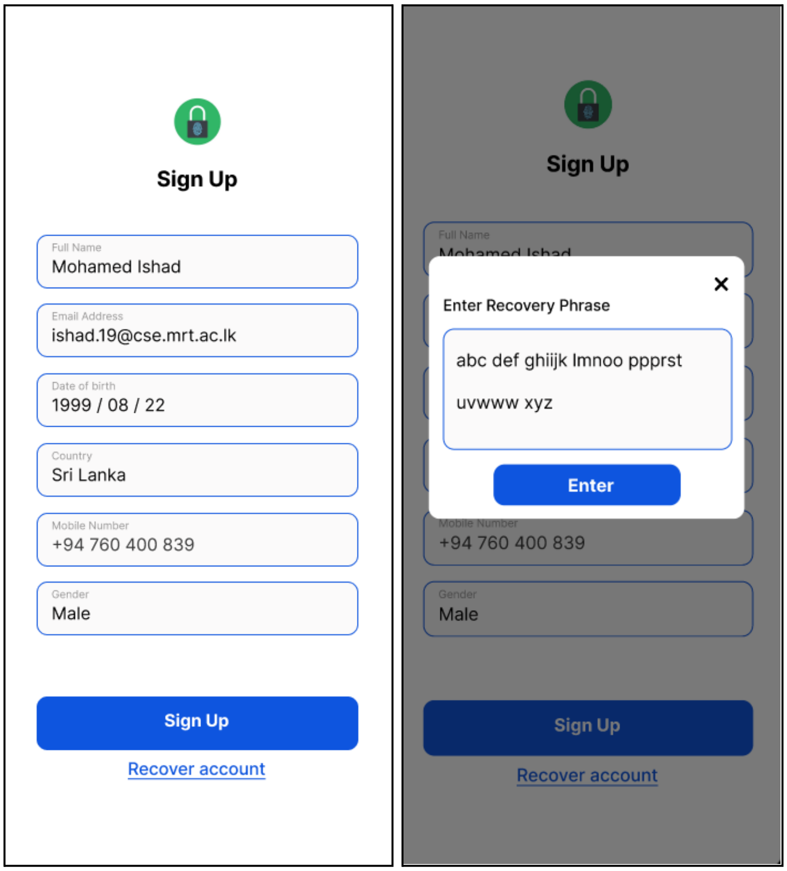
iBlock

You Data Secured in a Blockchain

Single Sign-on for multiple Platforms

Figure 2. Welcome tour screens of iBlock Mobile Application

- Signup Page - Appears when the user is not signup to the system



Full Name
Mohamed Ishad

Email Address
ishad.19@cse.mrt.ac.lk

Date of birth
1999 / 08 / 22

Country
Sri Lanka

Mobile Number
+94 760 400 839

Gender
Male

Full Name
Mohamed Ishad

Enter Recovery Phrase

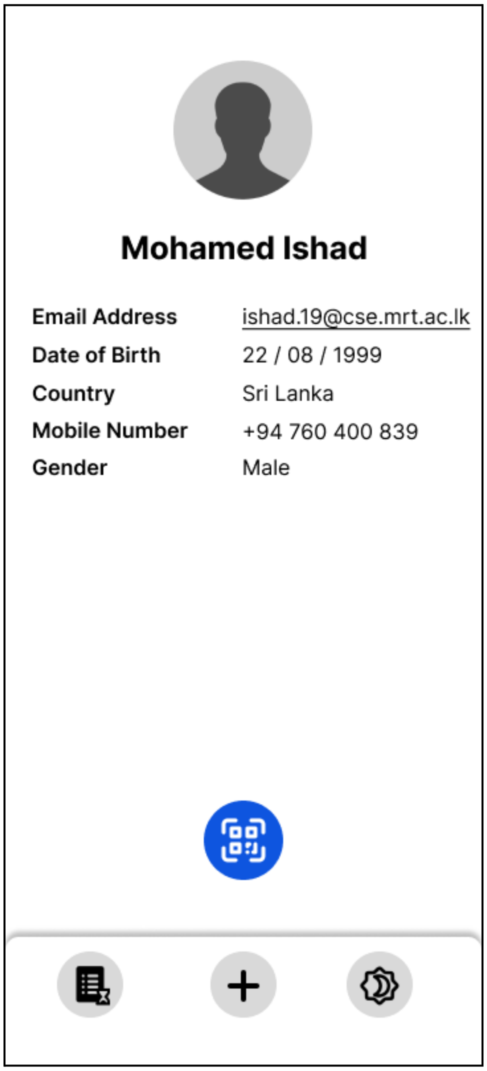
abc def ghiijk lmnoo ppprst uvwww xyz

Mobile Number
+94 760 400 839

Gender
Male

Figure 3. Signup, Recovery Screens of iBlock Mobile Application

- Home Page - Default page appears if the user is already logged in to the system.



Mohamed Ishad

Email Address	ishad.19@cse.mrt.ac.lk
Date of Birth	22 / 08 / 1999
Country	Sri Lanka
Mobile Number	+94 760 400 839
Gender	Male

Figure 4. Home screen of iBlock Mobile Application

- QR Reader

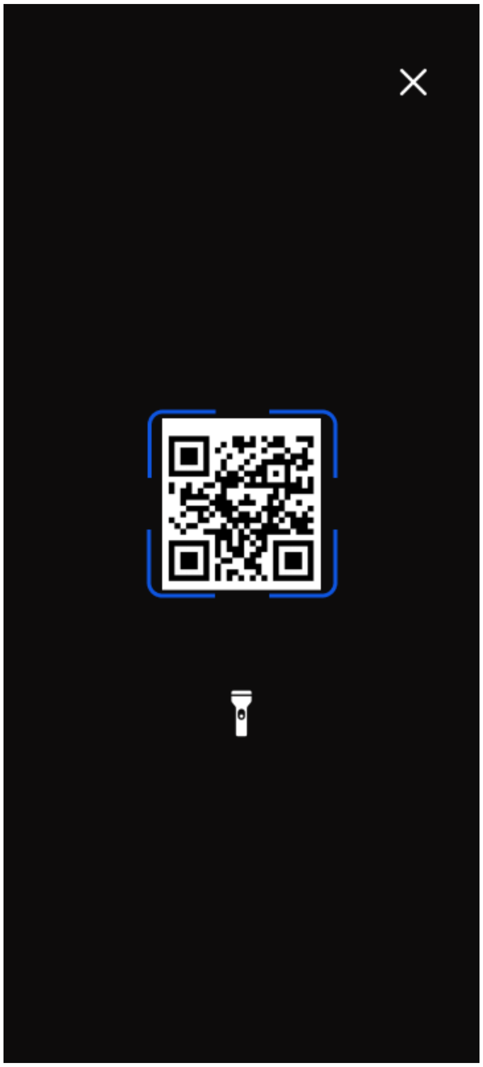
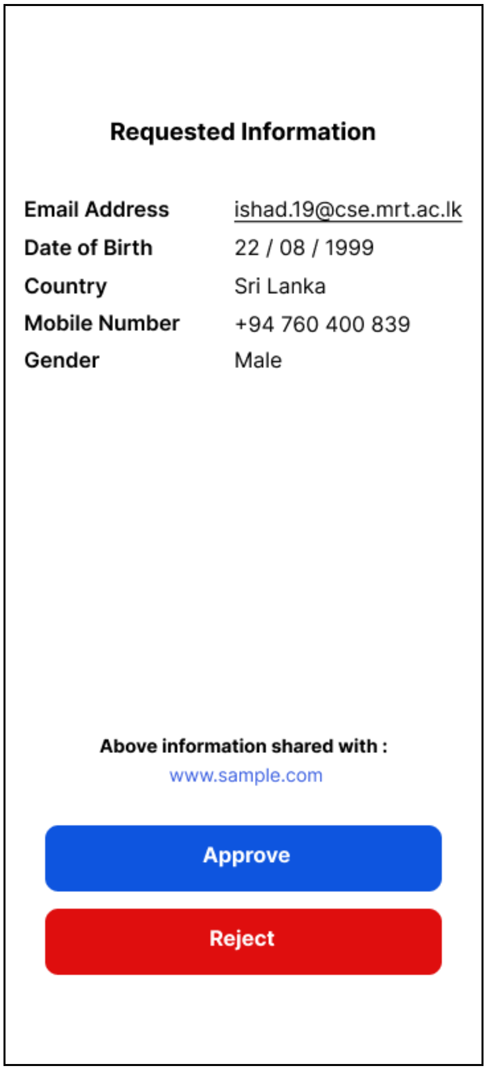


Figure 5. QR code scanning screen of iBlock Mobile Application

- Information Sharing Approval / Denial Page



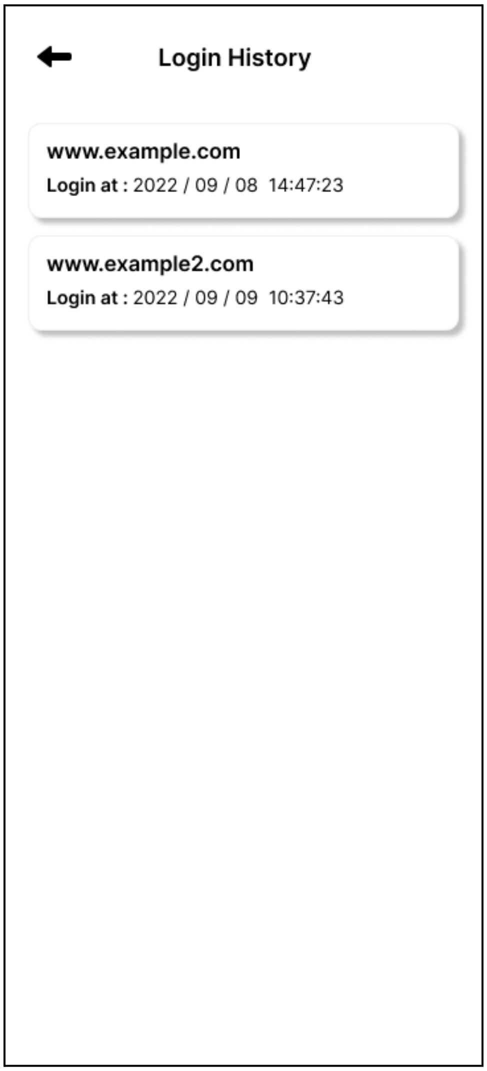
Requested Information

Email Address	ishad.19@cse.mrt.ac.lk
Date of Birth	22 / 08 / 1999
Country	Sri Lanka
Mobile Number	+94 760 400 839
Gender	Male

Above information shared with :
www.sample.com

Figure 6. Information Sharing screen of iBlock Mobile Application

- Information Shared History View Page



Login History

www.example.com
Login at : 2022 / 09 / 08 14:47:23

www.example2.com
Login at : 2022 / 09 / 09 10:37:43

Figure 7. Information sharing history screen of iBlock Mobile Application

3.9.1.1. Wireframes for Demo Application of a Service Provider (eg: a bank)

To select a service a user must be provided with a page with QR code which the iBlock user has to scan a page that prompts after the user verified himself and approved to use his personal information. The fields that the service provider gets access to are auto-filled in the form.

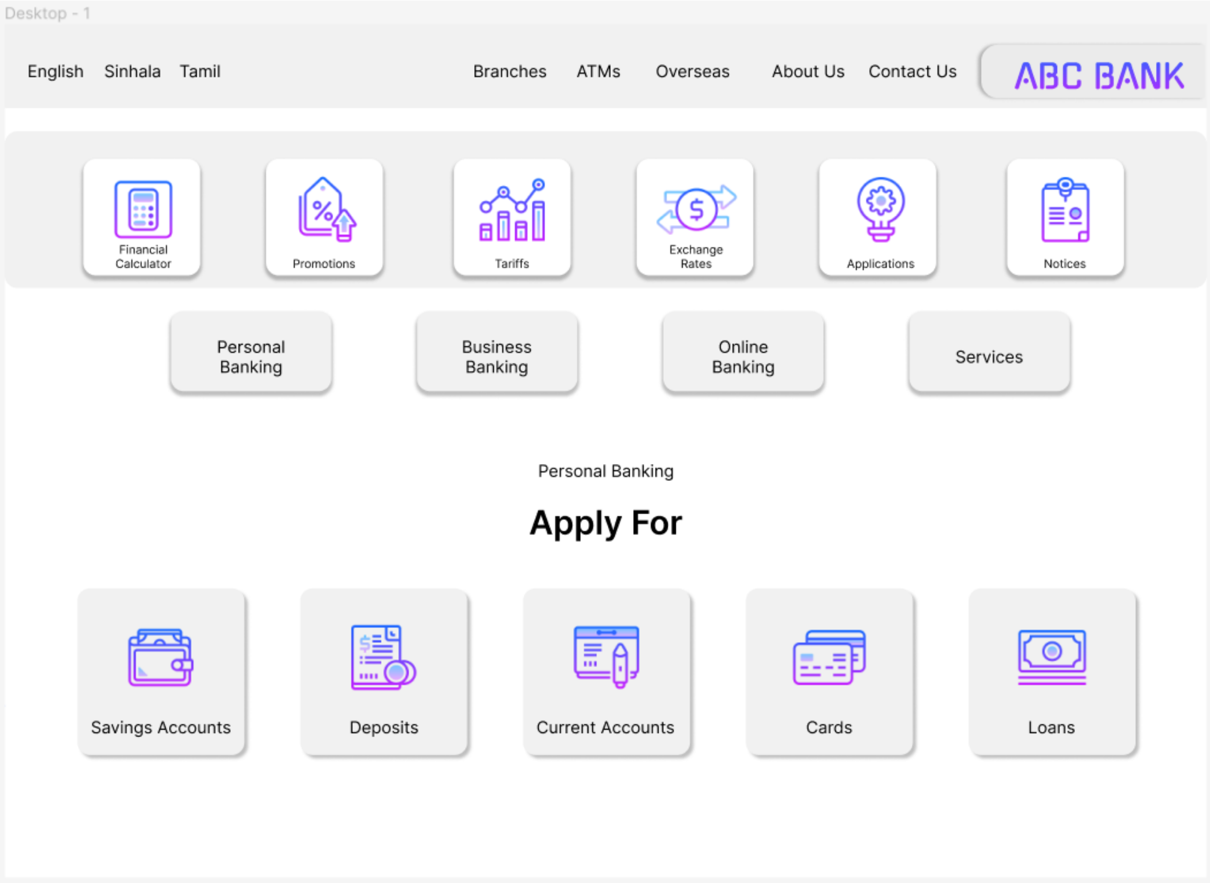
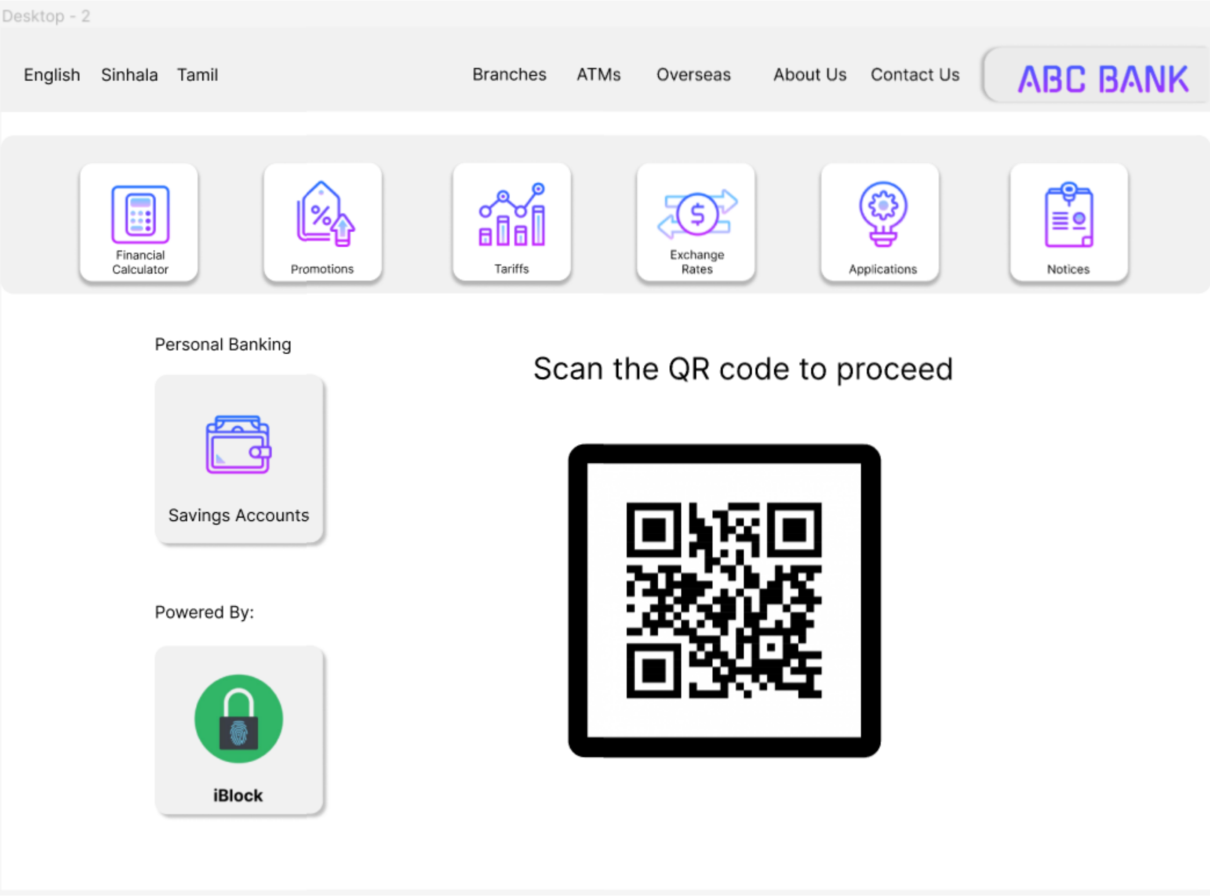
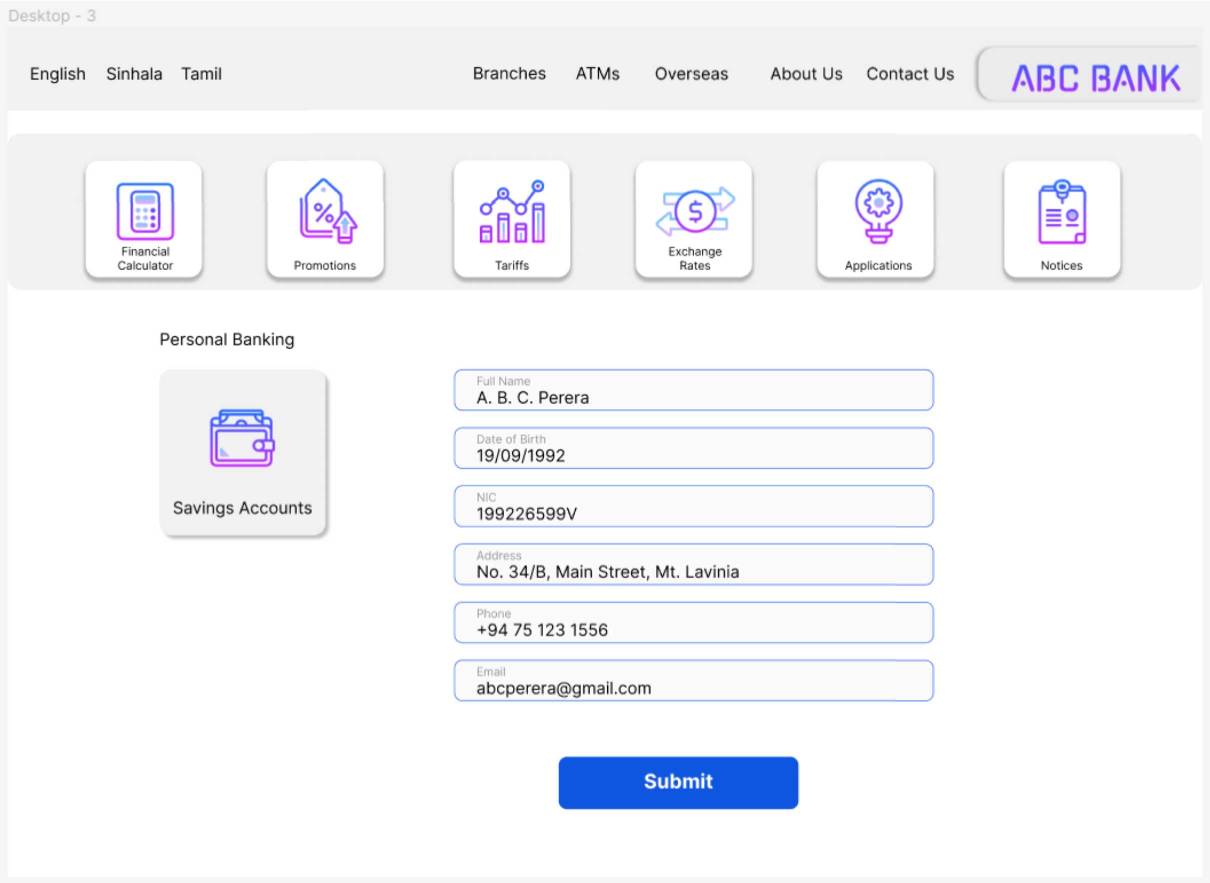


Figure 8. Landing page of a demo Service Provider’s Web Application



Scan the QR code to proceed

Figure 9. QR code generated page of a demo Service Provider's Web Application



Full Name
A. B. C. Perera

Date of Birth
19/09/1992

NIC
199226599V

Address
No. 34/B, Main Street, Mt. Lavinia

Phone
+94 75 123 1556

Email
abcperera@gmail.com

Figure 10. Information Received page of a demo Service Provider’s Web Application

3.9.2. Hardware Interfaces

The iBlock mobile app will be required to have an interface with the device’s camera to scan QR codes inside the application. This is the only hardware interface identified to be used in the project.

3.9.3. Software Interfaces

For API

Platform – Javascript

Third-party libraries – QR code generator

For Smart Contract

Platform – Ethereum
Language - Solidity

Application

Platform – Flutter

3.9.4. Communications Interfaces

A device capable of connecting to the internet and scanning QR codes is required to use the Application program.

A device capable of connecting to the internet is required to use any third-party applications that use the provided functionalities of the system.

For both cases, the HTTPS communication protocol will be used.

3.10 Database Requirements

The Ethereum testnet itself is proposed to be used as the distributed ledger to store all the request approval/rejection details and retrieve them through the user's wallet address whenever necessary. Avoiding the usage of a centralized database server ensures a trustless environment and makes the system tamper-proof.

3.11 Licensing, Legal, Copyright, and Other Notices

For the design phase, the proprietary, but free-to-use tools will be used hence the user interfaces will be bound to non-distributable licenses. The mobile application written using the open source Flutter framework allows the code of the mobile application to be publicly available on the GitHub repository. All the interface designs are copyrighted to the developer team. The Smart Contracts deployed on the Ethereum blockchain will not be allowed for the general public to access, only the source code will be available on the Github repository. The API supposed to be used by the service providers will be under Free and Open Source licenses.

3.12 Applicable Standards

- All reports and documentation will follow the IEEE standard format for each relevant document.
- The mobile application will follow the guidelines specified by the play store and app store.
- Will follow the widely accepted coding conventions (eg. variable, function, class naming conventions) for the respective languages used.
- Integration library will follow the standards specified by the npm registry.

4. Supporting Information

4.1. References

[1] Figma (https://www.figma.com/)

[2] Ethereum (https://ethereum.org/en/)

[3] Flutter (https://flutter.dev/)