Kyle Russell - 13831056

Highly Secure Systems

Documentation

TABLE OF CONTENTS

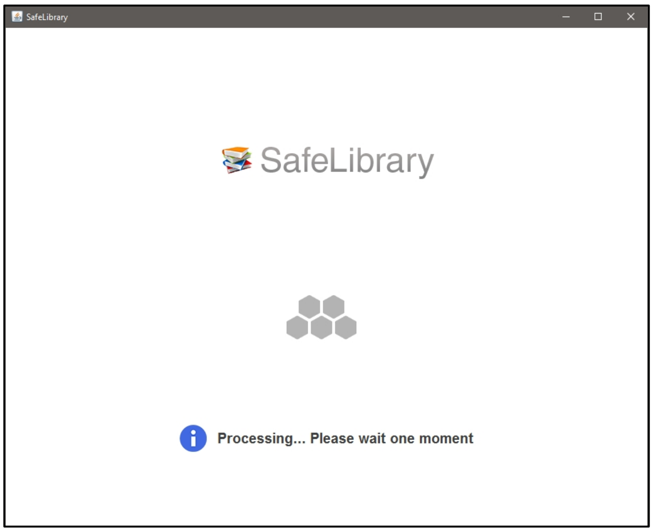
1.0 About

SafeLibrary is a secure smart card application that allows users to browse, borrow and return library books. All communication between the host application and the 'smart library card' is encrypted and verified to ensure confidentiality and authenticity for users. Additionally, an extra layer of security was added to help users protect themselves in the event that their card is lost/stolen where users can secure their account with a PIN code. The PIN code once set by the user, will be required each time they wish to use the library.

2.0 User Guide

2.1 Connecting

When a user first opens the SafeLibrary application, they will be faced with the connect view as shown in figure 1. If this is your first time connecting to SafeLibrary and you have no profile set up, you will not be prompted to enter a PIN code. Instead, once you have successfully connected, you will be asked to create a new profile and set up a PIN code to secure your account.



Processing... Please wait one moment

Figure 1: Connect view

2.2 Profile creation & settings

When a user first connects to SafeLibrary they will be required to set up a profile and PIN code. After connecting, you will be brought to the view shown in figure 2. Here you can enter your name, email, address and phone details. Additionally, you will need to enter a PIN code if you haven't done so already which will be required each time you login to prevent intruders accessing your account in the event that your card is lost/stolen. Once you have entered your details, click the 'Submit' button to confirm. You can come back anytime and change these settings and reset your PIN code by simply going to the settings tab.

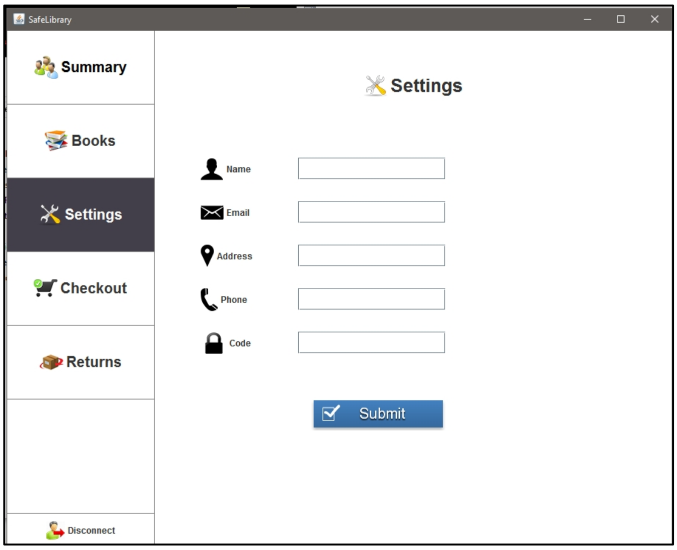
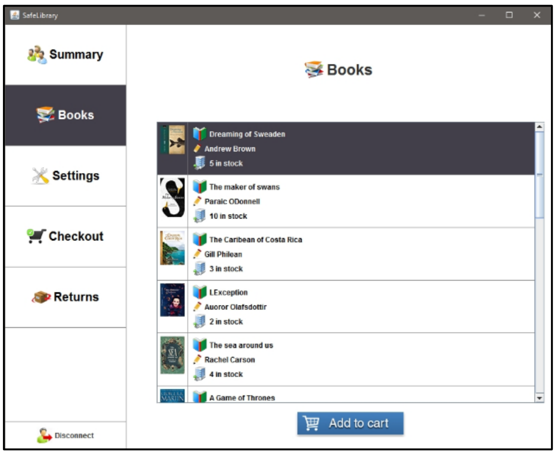


Figure 2: Settings view

2.3 Borrowing books

SafeLibrary allows users to borrow from a selection of books as shown in figure 3. When a user wishes to borrow a book, head to the 'Books' tab and browse for a desired book. When you have chosen a book, select it by clicking it in the list and click the 'Add to cart' button. When a book is added to a cart, it will appear in the checkout window as shown in figure 4. You can add multiple books to the cart however you can have a maximum of 8 books loaned at any given time, if the number of items in your shopping cart exceed this, then you may need to remove some. When you are ready to complete the transaction and borrow the books in your shopping cart, click the 'Submit' button in the checkout window. If you need to remove an item from your shopping cart, select the item in the checkout and click the 'Remove' button to remove it from the checkout.



Dreaming of Sweaden
Andrew Brown
5 in stock

The maker of swans
Paraic ODonnell
10 in stock

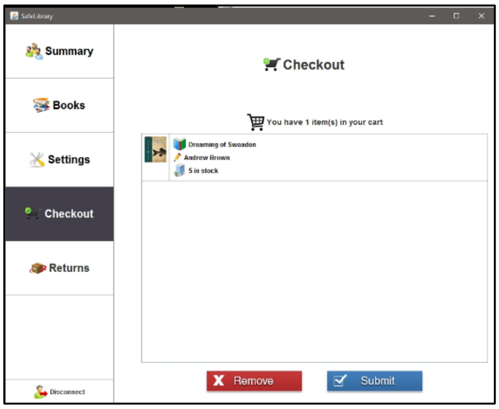
The Caribean of Costa Rica
Gill Philean
3 in stock

LException
Auoror Olafsdottir
2 in stock

The sea around us
Rachel Carson
4 in stock

A Game of Thrones

Figure 3:Book selection view



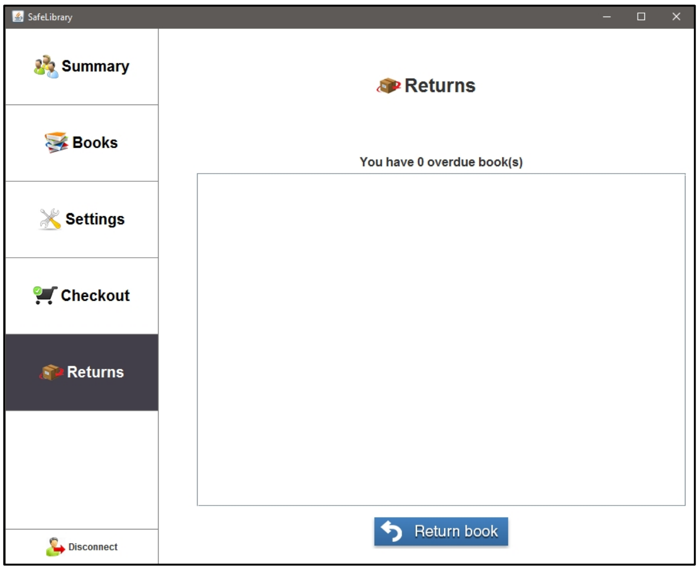
You have 1 item(s) in your cart

Dreaming of Sweaden
Andrew Brown
5 in stock

Figure 4: Checkout view

2.4 Returning books

A user who has previously borrowed books from the library and wishes to return some books so that they can loan new books, can do so by going to the ‘Returns’ tab in the SafeLibrary application as shown in figure 5. In this window, a list of your currently loaned books will be displayed and a book that needs to be returned can be, by selecting it from the list of books and clicking the ‘Return book’ button. You will be able to re-loan the book from the ‘Books’ tab again if you would like.



You have 0 overdue book(s)

Figure 5: Book return view

3.0 Technical notes

3.1 Installation

In order to install the host and card application, the respective sources need to be built, the applet needs to be deployed and installed then finally the cad simulator needs to be started and the host application run. The user will also need to have installed apache ant, JDK and the java card libraries. Once these packages have been downloaded, each respective bin directory will need to be added to the windows PATH environment variable. Next, you will need to edit the ‘build.xml’ file and change the ‘jdk.home’ property value to your JDK home directory path and the ‘javacardkit.home’ property value to your java card directory path.

Next build the applet by entering: ‘ant build-applet’
Then deploy the applet: ‘ant deploy-applet’
Then install the applet: ‘ant run-script’

Now build the host application sources by entering: ‘ant build-host’
Then start the cad simulator: ‘ant start-cad’
Open a new command line in the same directory and run the host by entering: ‘ant run-host’
The application will now start and connect to the deployed smart card.

3.2 Security

The project contains two remote objects: ‘User’ and ‘Books’ on the applet where the User object is used to maintain basic details about the card holder including its PIN and has various methods for manipulating these details. The Books remote object holds the users currently loaned books and has methods for adding, removing etc. books. The User object is locked by the PIN code if it is set by the user and restricts access to its methods if the PIN has not been and/or unsuccessfully verified. Additionally, the remote object maintains a reference to the applets security service instance and on each method its various properties are verified otherwise UserException’s are thrown. Also note, that each many of the methods that perform atomic operations on persistent objects are encapsulated in transactions in case of an outage or unexpected shutdown. Communication between the card and the host is encrypted and verified to ensure confidentiality and authenticity. Input APDU command and response blocks are encrypted with AES using a 128bit fixed key in CBC mode. Additionally, 128bit digital signatures were generated where input & ouput blocks were signed and verified using RSA_SHA_PKCS1 such that a 20 byte SHA digest block was generated then padded using the PKCS1 padding scheme and finally encrypted with 1024bit key RSA.

3.3 Database

The project comes with a small sqlite database ‘safelibrary.db’ which contains a single table ‘Books’ for maintaining all book records in the system. The decision to use a sqlite database over a data structure was that the books were persistent in the host and also there was scalability consideration such that the library may contain thousands of books that need to be maintained and frequently searched so it would be unreasonable to keep these in unorganized files or a simple data structure. By using sqlite, we can make use of indexing and powerful queries to make searches and updates very efficient. The table attributes consist of an integer primary key ID used to identify each book record, the books name, author and image are also stored.