**CS683 Project Assignment   
Certificate Inspector  
Will Aftring**

**Instructions**

* This is the template of your final project report. As this document will be constantly updated during the semester, please enable the “track changes” in your doc. Or if you prefer to use the md file, we can also see the change in the commit history.
* Please name your report as CS683\_<Last Name><First Name>\_<ProjectTitle>. It can be either a PDF or Word document.
* Make sure to push all your code into your github repository, create a release/tag and submit the link on blackboard.
* Please provide your feedback in the “Add comments” section when submitting your report. Thanks!

[**Overview**](#_g6igqliy7rm) **2**

[**Related Work**](#_bf21eadgjj29) **2**

[**Requirement Analysis and Testing**](#_9dheewbiht5g) **2**

[**Design and Implementation**](#_312k3b3li0xh) **3**

[**Project Structure**](#_hkcglxnjhrt2) **4**

[**Timeline**](#_tp0jpote18vj) **4**

[**Future Work (Optional)**](#_wx5fnmke6x6g) **4**

[**Project Demo Links**](#_nl6zntsisnrv) **4**

# Overview

The purpose of this project is to allow for an easier way to interface with x509 certificates in the Android stores. The default android interface for this is limited and does not provide adequate details for system administrators or Power Users to investigate and troubleshoot certificate related issues.

# Related Work

The most similar application would be the Root Certificate Manager(ROOT) application.

It offers much of the same functionality however is more intended to change the state of the certificate store. My application is intended as a tool to investigate the state of the store as it exists. It has no intention of changing the state. This limits the required privileges of the application.

# Requirement Analysis and Testing

(*This section should clearly describe all features/requirements that you plan to implement or have implemented for your application. You should separate them into three categories: essential, desirable and optional.*

*For each requirement/feature, you should provide the following details:*

|  |  |
| --- | --- |
| *Title* | *Enumerate Android certificate stores (Essential) E1* |
| *Description* | *Enumerate certificates stored in the various Android certificate stores.* |
| *Mockups* | *A graph paper with a diagram  Description automatically generated* |
| *Acceptance tests* | *The user selects a certificate store from the spinner.*  *The certificates within the selected store are enumerated in a list.*  *Identifying information for each certificate is displayed in the list.* |
| *Test Results* | *Success. We can select different certificate stores.* |
| *Status* | * *07/19/23: We are now able to enumerate all certificates in the stores. However we are still pending general details about those certificates. An initial framework is in place so I should be able to iterate much more quickly now.* * *07/26/23: Overhaul on this component. We have now better associations between fragments and the display objects* * *08/09/23: Basic implementation done* |

|  |  |
| --- | --- |
| *Title* | *Display Certificate Details (Essential) E2* |
| *Description* | *Display details of selected certificate* |
| *Mockups* | *A graph paper with a drawing of a test result  Description automatically generated* |
| *Acceptance tests* | *User can scroll through certificate details in the list.* |
| *Test Results* |  |
| *Status* | * *07/19/23: Pending. Dependent on E1* * *07/26/23: Initial view for this has been created* * *08/09/23: Backing data is handled and basic view is put together* |

|  |  |
| --- | --- |
| *Title* | *Validate certificate revocation list (Desirable) D1* |
| *Description* | *Reach out over the network via CRL defined protocol to query certificate validity.* |
| *Mockups* | *A graph paper with a drawing of a test result  Description automatically generated* |
| *Acceptance tests* | *User selects one of the CRL endpoints defined on the certificate.*  *User selects “Test” for the endpoint.*  *A pop-up is generated to determine if the certificate is valid or any failure conditions.* |
| *Test Results* |  |
| *Status* | * *07/19/23: Pending. Dependent on E2* * *07/26/23: No updates* * *08/09/23: No updates* |

|  |  |
| --- | --- |
| *Title* | *Validate Certificate Chain (Optional) D2* |
| *Description* | *Walk through the certificate chain to ensure that it is valid.* |
| *Mockups* |  |
| *Acceptance tests* | *User selects certificate from certificate list*  *User selects walk-chain*  *Application walks certificate chain validating certificate validity*  *Application reports success or failure with condition of failure.* |
| *Test Results* |  |
| *Status* | * *07/19/23: Pending. Dependent on E2* * *07/26/23: No updates* * *08/09/23: No updates* |

|  |  |
| --- | --- |
| *Title* | *Certificate Validity Caching (Optional)* |
| *Description* | *Include a visual to indicate certificate validity in the list* |
| *Mockups* |  |
| *Acceptance tests* | *After a test of a certificate the validity of that certificate will be cached dynamically based on:*   * *The test that was performed* * *When that test would need to be performed again* |
| *Test Results* |  |
| *Status* | * *07/19/23: Pending. Dependent on D1 OR D2* * *07/26/23: We now indicate previous status* * *08/09/23: No updates* |

# Design and Implementation

(*This section should describe the basic architecture (e.g. MVC, or MVVM) and your detailed design and implementation. This section may contain the following aspects:*

* *Basic architecture*
* *UI design and implementation*
  + *Activities,*
  + *Fragments*
  + *Lists*
  + *Spinners*
  + *Buttons*
  + *RecyclerViews*
* *Other android features* 
  + *Android certificate store*
  + *Android network connectivity*
  + *Data Binding*
* *Third party APIs: N/A*
* *Data Design and implementation* 
  + *Given data is already stored Android certificate limited data storage*
  + *If caching is implemented likely leverage a lightweight database for storing certificate validity state.*
* *Algorithms*
* *…*

# Project Structure

* Activities:
  + MainActivity.kt – Entry point
  + CertificateInspectorApplication.kt – Application class
  + MDebug.kt – Debug helper class
* DataLayer
  + CertificateDetails.kt – Backing data structure for the X.509 certificates
  + CertifiacteStore.kt – Backing for the certificate stores / interfacing with the android certificate stores
* Adapter
  + CertificateRecyclerViewAdapter.kt – The backing of the adapter for the certificate store recycler view
* Fragments
  + CertificateStoreFragment.kt – Backing view for the certificate store fragment
  + CertifiacteDetailFragment.kt – Backing view for the certificate details fragment
* ViewModel
  + CertificateStoreViewModel.kt – Class to interface with the different certificate stores
  + CurCertificateStoreViewModel.kt – Class to query the current certificate store
  + CurCertificateViewModel.kt – Class to query the current certificate
* Layouts:
  + fragment\_certificate\_card.xml – This is the fragment used to display data in the certificate store recycler
  + fragment\_certificate\_details.xml – Details about the certificate once the item is selected from the certificate store view
  + certificate\_store.xml – Spinner & Recycler for selecting certificates from different stores
* Navigation:
  + Nav\_graph.xml – Logic for navigation between certificate\_store fragment and the fragment\_certificate\_details

# Timeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Application Requirements  (Essential/Desirable/Optional) | Android Components and Features to be used | Member 1 contribution/ planned tasks | Member 2 contribution/ planned tasks |
| 1 | Begin E1, and create framework to be built upon for certificate interaction | KeyStore  Spinner  ScrollView  Navigation  Fragments |  |  |
| 2 | Complete E1 & E2 |  |  |  |
| 3 | Complete details E2 | X.509 certificate details. |  |  |

# Future Work (Optional)

(*This section can describe possible future works. Particularly the requirements you planned but didn’t get time to implement, and possible Android components or features to implement them.*

*This section is optional, and you can include this section in the final iteration if you want.*)

# Project Demo Links

(*For on campus students, we will have project presentations in class. For online students, you are required to submit a video of your project presentation which includes a demo of your app and explanation of your implementation. You can use Kaltura or zoom or any video tool to make the video and then submit it on blackboard. Please check the following link for the details of using Kaltura to make and submit videos on blackboard. You can also use other video tools and upload your video to youtube if you like:* [*https://onlinecampus.bu.edu/bbcswebdav/courses/00cwr\_odeelements/metcs/cs\_Kaltura.htm*](https://onlinecampus.bu.edu/bbcswebdav/pid-523716-dt-announcement-rid-19162119_1/xid-19162119_1) )

1. References   
   (*any references you used for the project*)