Simple Software Requirements Specification ("SSRS")
for
CSE 3310-002

Fivetran

December 6, 2019

Prepared by:

Kyra Belgica
Hamilton Nguyen
Marvin Wellington

Project Description

Secret Calculator: Original need and purpose for the system

Secret calculator is a fully functional basic calculator with a hidden app that stores and masks files and/or pictures behind it. The calculator can perform basic operations such as addition, subtraction, division and multiplication on all real numbers. The calculator may also clear, perform percentage arithmetic operations, and input negative numbers. Upon the first time use of the app, the user will be prompted to create a four-digit numerical code within a GUI window. To access the "masked" files, the user will enter a four-digit numerical password followed by a "%" (percentage symbol). Once the user has access to the spectre app by their set password, it will allow the user to store photos (e.g. .jpeg) and text files. Additionally, the app will allow a user to add domains, usernames, and passwords in a list on the app. After 10 seconds of idle activity in the Spectre Screen, the app will automatically timeout and return back to the calculator GUI. If the user wishes to perform a factory reset of their app, they may do so by entering the string "0000+".

Files uploaded to the app will be stored on the device's storage. They will be accessed through the app and will be encrypted when not in use. When in use, they will be decrypted before viewing. The files, which the user chooses to view, will be opened through the phone's default apps. For example, image files will be opened through the Gallery app. In this project, the app will support the minimum of Android 4.0 or higher.

Overview and scope of the system

A. Calculator - Add, subtract, multiply, and divide. The role of this function is to disguise the app from the secret folder
B. Passcode - Enter a four-digit password to gain access to the spectre screen
C. Storage - Upload pictures, videos, and text files. These may be viewed by type or by all

Resources

A. Computer Hardware - Laptops and desktops (at least a minimum of 5 computer units)
B. Android Studio IDE (free distribution, Windows based software)
C. Literature and Online Android development resources (e.g. StackOverFlow, Youtube, Head First Android Development Textbook, and Android Programming: Pushing the Limits textbook)
D. A physical Android cell phone

Our Team

Kyra Belgica

I have never done Android development! The only thing tangent to that is making very basic apps with Scratch in high school. I'm comfortable with a few languages such as Java, which will be very helpful in this project. Even though I'm not the strongest programmer, I have a lot of project managing experience through Air Force ROTC, which can help us tremendously in our sprints.

Hamilton Nguyen

My educational background is in Software Engineering and Chemistry. I have two years of programming and development experience in various languages (e.g. C, C++, Java, HTML5, PHP, CSS, MySQL, Python). I have some practical experiences with IDE platforms, OS platforms and other computing operations (e.g. Shell Scripting, Eclipse, Netbeans, Jetbeans, UNIX, Linux Distributions, GNU debugger, SQL). I also have four years of practical industry experience as a professional chemist, I specialized in Drug Discovery Proteomics, large data analytics, and High-Throughput Screening. This project will be my first exposure to android based application development through a driven Agile Process.

Marvin Wellington

Creating an android app is new to me. I've learned to program in Java, C, and C++ within my undergraduate program. Prior to starting my pre-professional courses, I had no experience in programming. I look forward to learning how to build an android app.

Requirements

#	Source	Functional Requirements
R1	3	The app shall allow the user to perform the four basic functions of a calculator (addition, subtraction, multiplication, and division).
R2	5	The app shall allow the user to clear, perform percentage arithmetic operations, and input negative numbers.
R3	6	The app shall allow the user to enter their personalized passcode upon opening the "Spectre" screen. First time access only.
R4	13	The app shall allow the user to reset the calculator factory setting with the input "0000%"
R5	8	The app shall bring the user to a storage section in "spectre" screen upon entering the passcode followed by "%" sign.
R6	10	The app shall allow the user to store images files via camera or upload. jpeg file only.
R7	11	The app shall allow the user to store credential information such as domain, username and password within the app.
R8	10	The app shall allow the user to store text files .txt format only.
R9	11	The app shall allow the user to view the stored domain names, usernames,and passwords upon request.
R10	11	The spectre screen shall timeout and return to the calculator screen after 10 seconds of inactivity.
R11	16	The files uploaded to the app shall be stored on the phone's external secure digital card.
R12	17	The files shall be encrypted when not viewed. (AES, SHA-256)
R13	17	The files shall be decrypted before viewing. (AES, SHA-256)
R14	18	The files uploaded to the app shall be opened by the phone's default apps (e.g. .jpegs will be opened with the phone's Gallery app).
Non-Functional Requirements		
R15	19	The app shall support the minimum of Android 4.x or higher.
R16	30	The app shall be developed using Android Studio 3.0.
R17	31	The app shall be developed using a reference textbook name Android Studio 3.0 Essentials.
Interface Requirements		
R18	3	The app's calculator interface shall be the same GUI as a regular android calculator.
R19	26	The app's storage interface shall allow the user to view files by all or by type.

High Level Use Cases

UC 1: Open the Secret Calculator app.

-TUCBW the user clicked on the app icon “Calculator” through an android phone.

-TUCEW the user will enter a four numerical lock code into the prompt. Upon completion, the app will automatically store the four numerical lock code and display to the calculator screen.

UC 2: Secret Calculator basic operations.

-TUCBW the user clicked on the app icon “Calculator” through an android phone.

-TUCEW the user completes basic operations by inputting it exactly in this syntax e.g. ((integer)(arithmetic operator)(integer)) and received an output. The user can also input AC, negative numbers, and percentage operation.

UC 2.1: Secret Calculator addition operation.

-TUCBW the user performed basic addition operation (e.g. 2+2) by inputting in the calculator.

-TUCEW the user received a calculated output in the output screen.

UC 2.2: Secret Calculator subtraction operation.

-TUCBW the user performed basic subtraction operation (e.g. 2-2) by inputting in the calculator.

-TUCEW the user received a calculated output in the output screen.

UC 2.3: Secret Calculator multiplication operation.

-TUCBW the user performed basic multiplication operation (e.g. 2*2) by inputting in the calculator without unexpected fail execution.

-TUCEW the user received a calculated output in the output screen.

UC 2.4: Secret Calculator division operation.

-TUCBW the user performed basic division operation (e.g. 2/2) by inputting in the calculator without unexpected fail execution.

-TUCEW the user received a calculated output in the output screen.

UC 3: Opening the hidden interface GUI “Spectre Screen”.

-TUCBW the user will enter the four numerical lock code through the calculator GUI keypad followed by the “%” sign. (e.g. 1234%)

-TUCEW the app will accept the code, then it will display to the hidden interface GUI “Spectre Screen”. After 10 seconds of inactivity, the spectre screen will return to the calculator screen.

UC 4: Store Pics file in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user clicked on “Store Pics” button, the app will prompt a dialog box, and the user will click on the media files to upload (e.g. .jpeg files).

-TUCEW the app displaying a dialog box "upload complete” and will store the file on the device.

UC 4.1: Store Documents file in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user clicked on “Store Documents” button, the app will prompt a dialog box, and the user will click on the media files to upload (e.g. .txt files).

-TUCEW the app displaying a dialog box "upload complete” and will store the files on the device.

UC 4.2: Delete media file in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user clicked on “delete” button, the app will prompt a dialog box, and the user will click on the media file to delete (e.g. .txt and .jpeg files).

-TUCEW the app displaying a dialog box "delete complete”.

UC 4.3: Open media file in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user clicked on “open” button, the app will prompt a dialog box, and the user will click on the media file to open and execute it with a default application to open file. (e.g. .jpeg files open by a phone’s photo gallery app).

-TUCEW the user hitting the back button once on the android phone and will navigate back to “Spectre” screen.

UC 4.4: Storing domain, username and password in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user entering the domain, username and password to store in the GUI “spectre Screen”. After entering, the user will click save button on the app.

-TUCEW the app accepting the input from the user and saving it in the app.

UC 4.5: View credentials in the hidden Interface GUI “Spectre Screen”.

-TUCBW the user click on the button View Credentials in the spectre Screen.

-TUCEW the app accepts the input from the user and view the domain, username, and password to the user.

UC 4.6: Reset “Spectre” screen

-TUCBW the user clicked on “reset” button, the app will prompt a dialog box, the user will click accept and all media files will delete from the hidden storage.

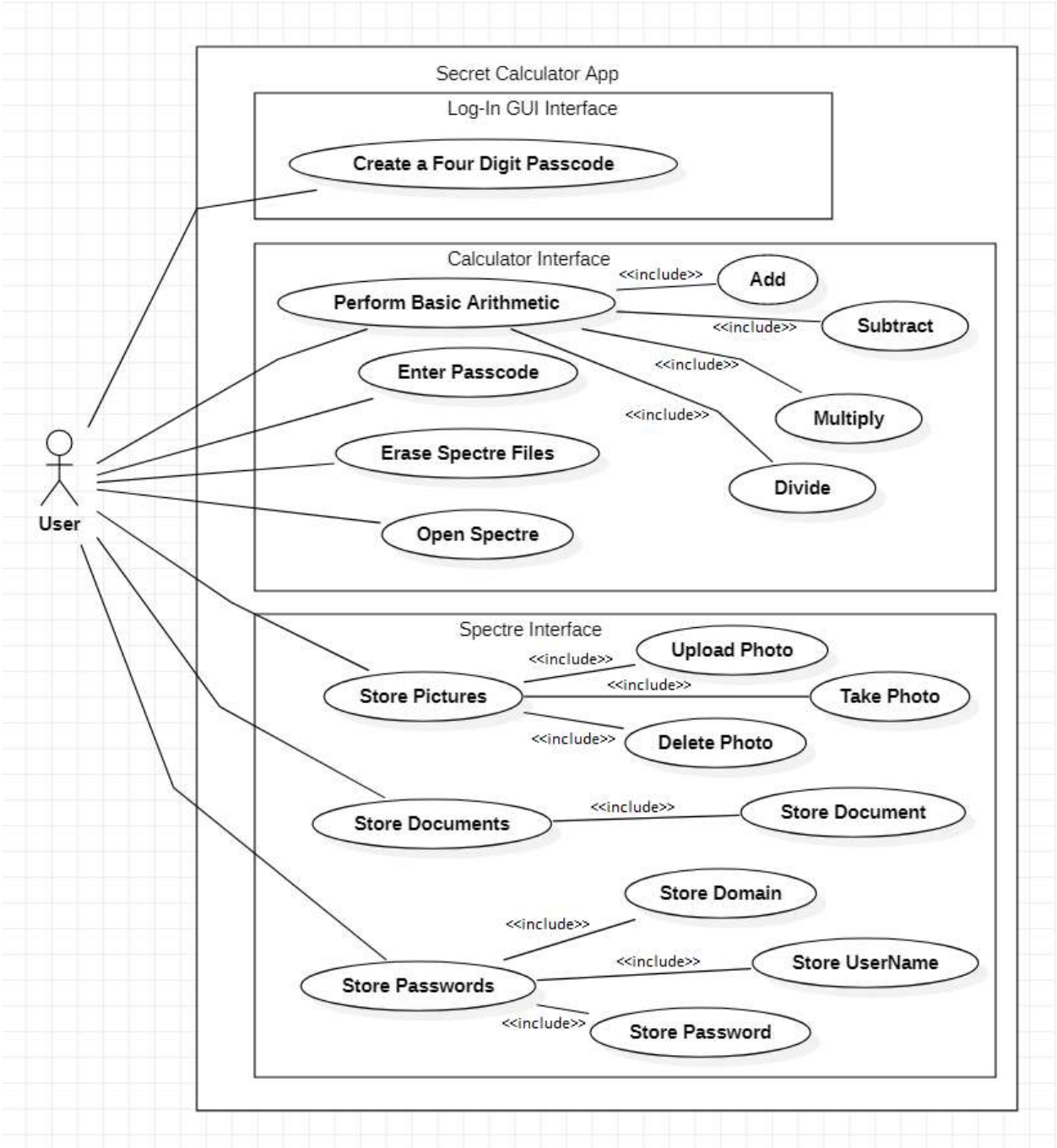
- TUCEW the app displaying a dialog box "reset complete”.

UC 5: Resetting to original calculator app in the calculator screen.

-TUCBW the user will enter “0000%” through a calculator numerical keypad. The app will prompt a display to the user, the user will either click agree or disagree and then the app execute the user input.

-TUCEW the app will delete all media files stored in the calculator, and reset to the original distributed version of the android app.

Use Case Diagram



Requirements-Use Case Traceability Matrix

	Priority Weight	All UC	UC1	UC 2, 2.1-2.4	UC 3	UC 4, 4.1-4.5	UC 5
R1	4	x		x			
R2	4	x		x			
R3	3	x	x				
R4	3	x					x
R5	2	x				x	
R6	2	x				x	
R7	2	x				x	
R8	2	x				x	
R9	3	x				x	
R10	3	x			x		
R11	4	x				x	
R12	4	x				x	
R13	4	x	x				
R14	4	x				x	
R15	3	x	x				
R16	2	x	x				
R17	2	x	x				
R18	2	x	x				
R19	2	x	x				x
SCORE		53	18	8	7	22	3

LEGEND SCORE:	**PRIORITY WEIGHT:**
HIGH NUMBER = HIGH PRIORITY TASKS	1 = lowest priority
LOW NUMBER = LOW PRIORITY TASKS	4 = highest priority

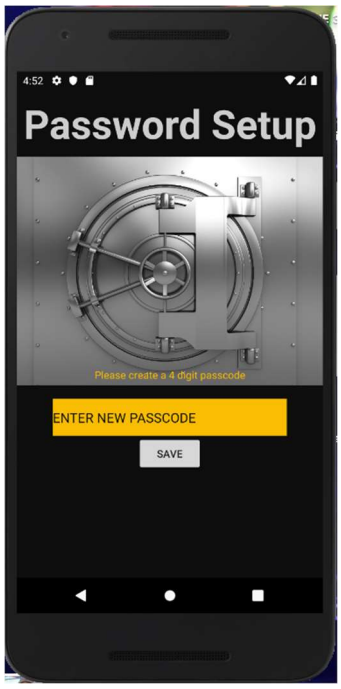
Increment Matrix

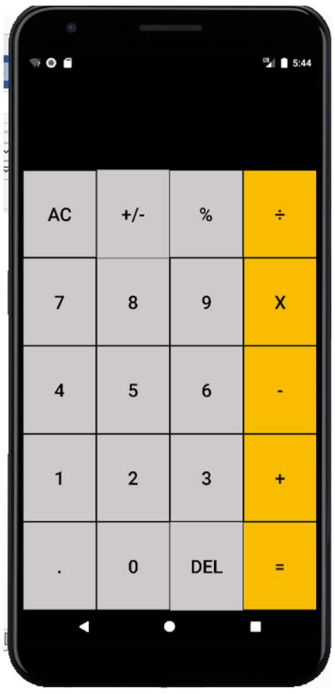
Use Case	Priority	Effort (people)	Depends on	Increment 1	Increment 2	Increment 3
UC 1	18	2	None	2	-	-
UC 2, 2.1-2.4	8	2	UC 1	2	-	-
UC 3	3	3	UC 1	-	2	1
UC 4, 4.1-4.5	22	5	UC 3	-	3	3
UC 5	3	3	UC 3, 4, 4.1-4.5	-	1	2
Total Effort		15		4	6	6

LEGEND SCORE:
HIGH NUMBER = HIGH PRIORITY TASKS
LOW NUMBER = LOW PRIORITY TASKS

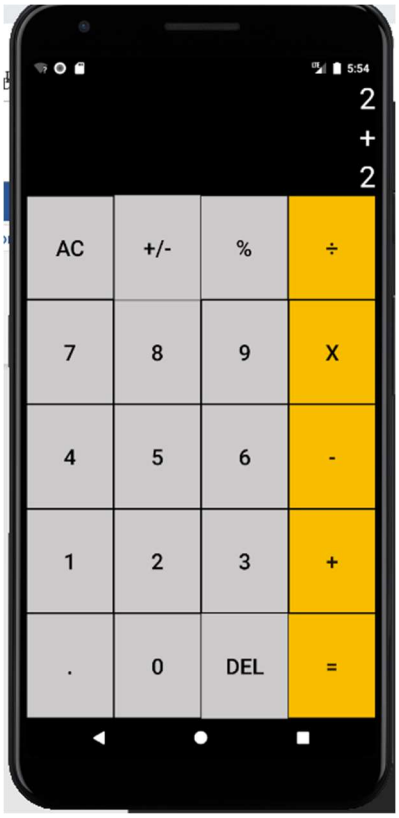
Expanded Use Case

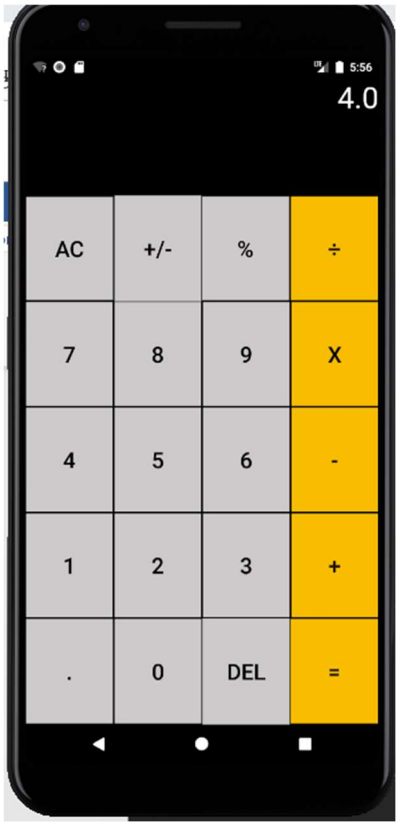
Use Case 1: Open the Secret Calculator app.	
Precondition: This use case assumes that the user has turned on the android phone unit and is on the home screen of the android phone.	
Actor: User	System: Secret Calculator
	0. System displays the Secret calculator Icon.
1. **TUCBW** the user clicked on the app icon "Calculator" through an android phone.	2. The system will open the Password Setup screen.
2. The user will enter a four numerical lock code into the prompt and click save.	3. System will process and store the four numerical lock code.
	4. the system display to the calculator screen.
5. **TUCEW** the user seeing the display of the calculator screen.	
Postcondition: The app is immediately available for arithmetic calculations and access to the hidden screen, "Spectre" screen.	



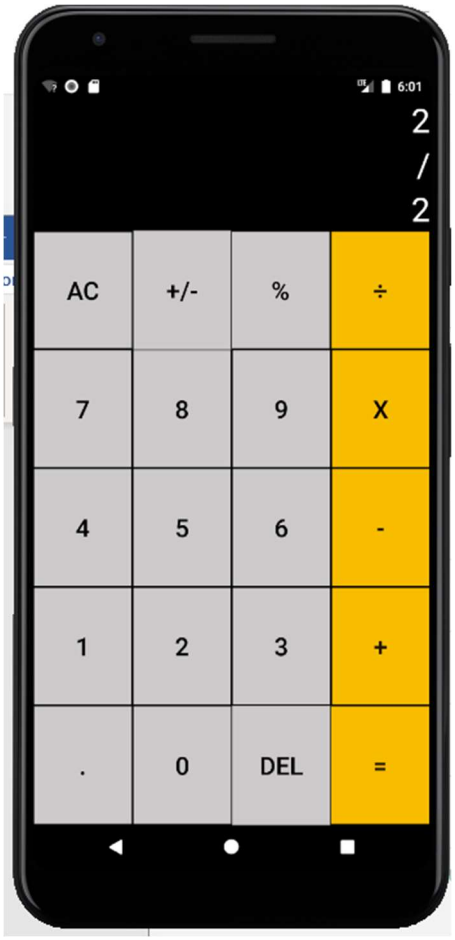


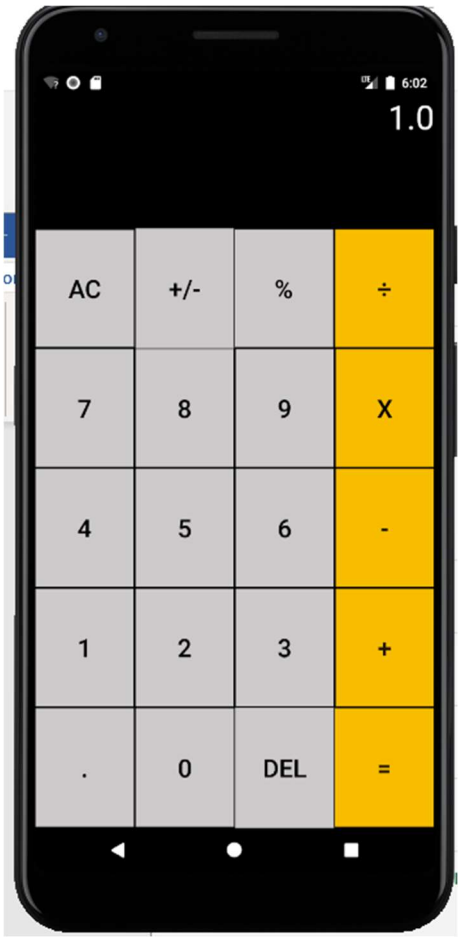
Use Case 2.1: Secret Calculator addition operation.	
Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the calculator screen of the app.	
Actor: User	System: Secret Calculator
	0. The System displays the calculator screen.
1. **TUCBW** the user performed basic addition operation (e.g. 2+2) by inputting it in the calculator.	*2. The System accepts the input and perform an arithmetic operation.
	3. The system will display the output to the calculator screen.
4. **TUCEW** the user will see the output of the operation.	
Postcondition: The app is immediately available for another arithmetic calculations and have access to the hidden screen, “Spectre” screen.	



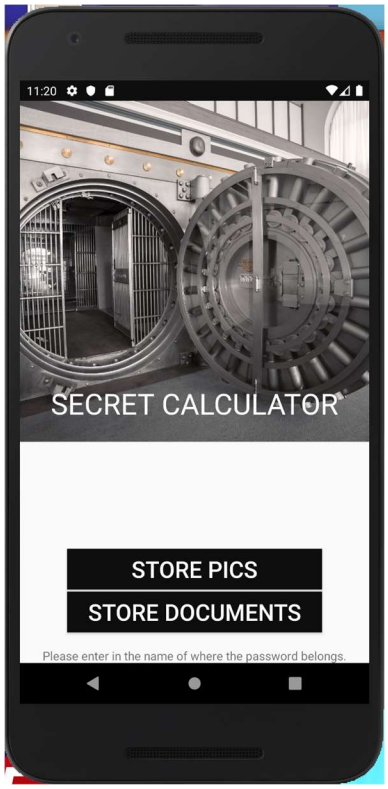


Use Case 2.4: Secret Calculator division operation.	
Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the calculator screen of the app.	
Actor: User	System: Secret Calculator
	0. The System displays the calculator screen.
1. **TUCBW** the user performed basic division operation (e.g. 2/2) by inputting in the calculator.	*2. System accepts the input and perform basic arithmetic operations.
	3. The system will display the output to the calculator screen.
4. **TUCEW** the user will see the output of the operation.	
Postcondition: The app is immediately available for another arithmetic calculations and have access to the hidden screen, “Spectre” screen.	



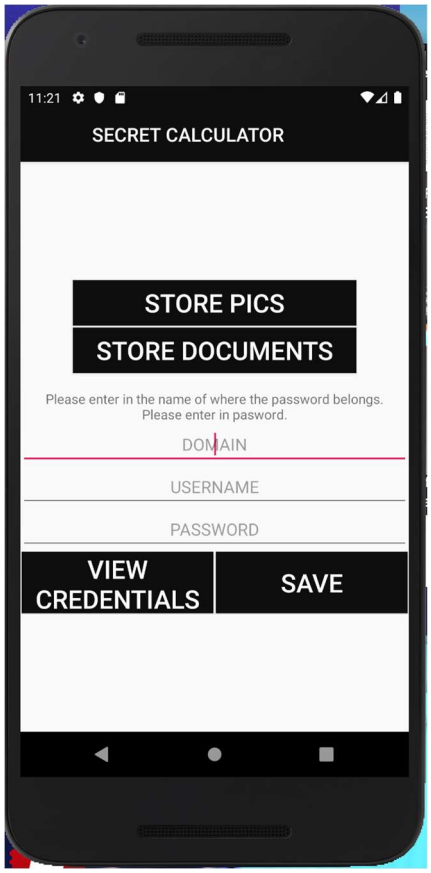


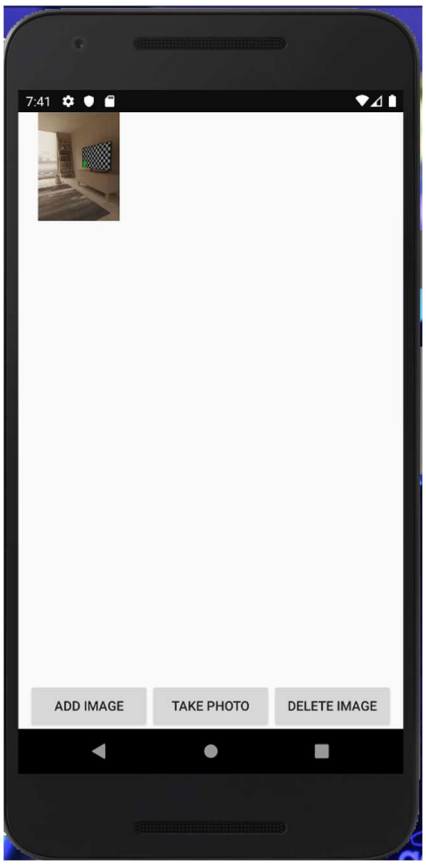
Use Case 3: Opening the hidden interface GUI “Spectre Screen”	
Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the calculator screen of the app.	
Actor: User	System: Secret Calculator
	0. The System display a calculator screen.
1. **TUCBW** the user will enter the four numerical lock code through calculator GUI keypad followed by the “%’ sign. (e.g. 1234%)	*2. System accepts the input and performed background processing.
	3. The system will display and access to the hidden interface GUI “Spectre Screen”.
4. **TUCEW** the user having access to the “spectre Screen”.	
Postcondition: The app is immediately available for other operations inside the hidden screen, “Spectre” screen.	



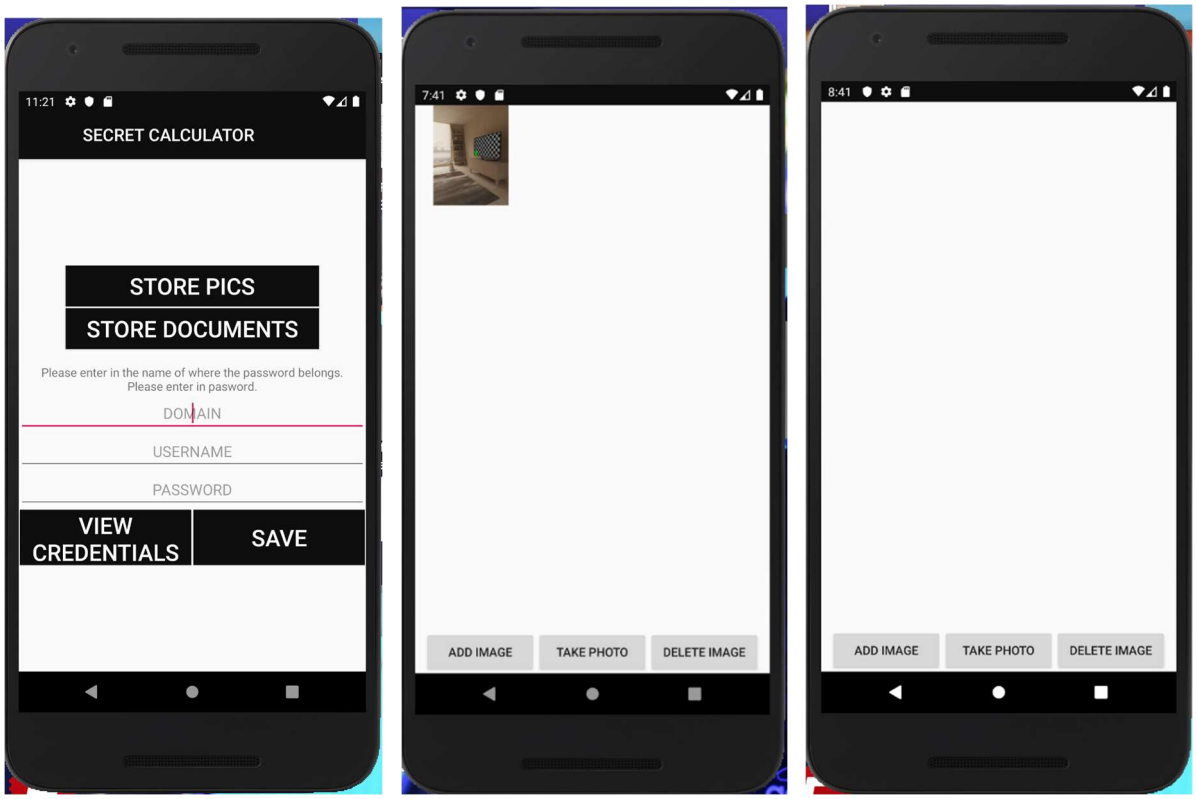
Use Case 4: Store Pic file in the hidden Interface GUI “Spectre Screen”

Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the “Spectre Screen” of the app.	
Actor: User	System: Secret Calculator
	0. System displays a “Spectre Screen”.
1. **TUCBW** the user clicked on “Store Pics” button.	2. System will prompt a dialog box.
3. the user will click on the media files to upload (e.g.jpeg files).	4. the system will accept and display the uploaded media file in a dialog box.
5. **TUCEW** the user seeing the uploaded media file.	
Postcondition: The app is immediately available for other operations inside the dialog box such as e.g. add image, take photo, delete image.	

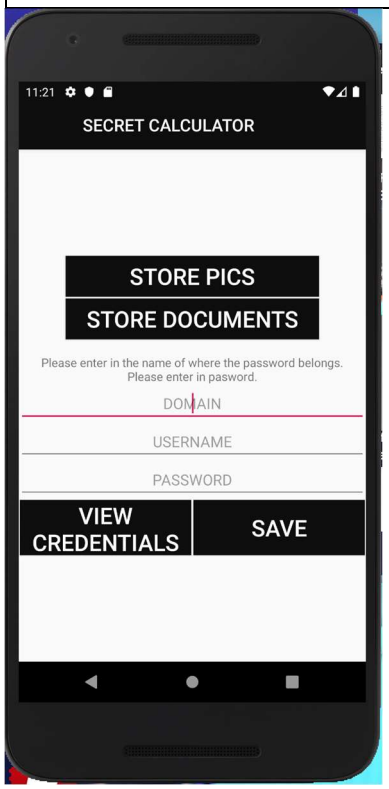




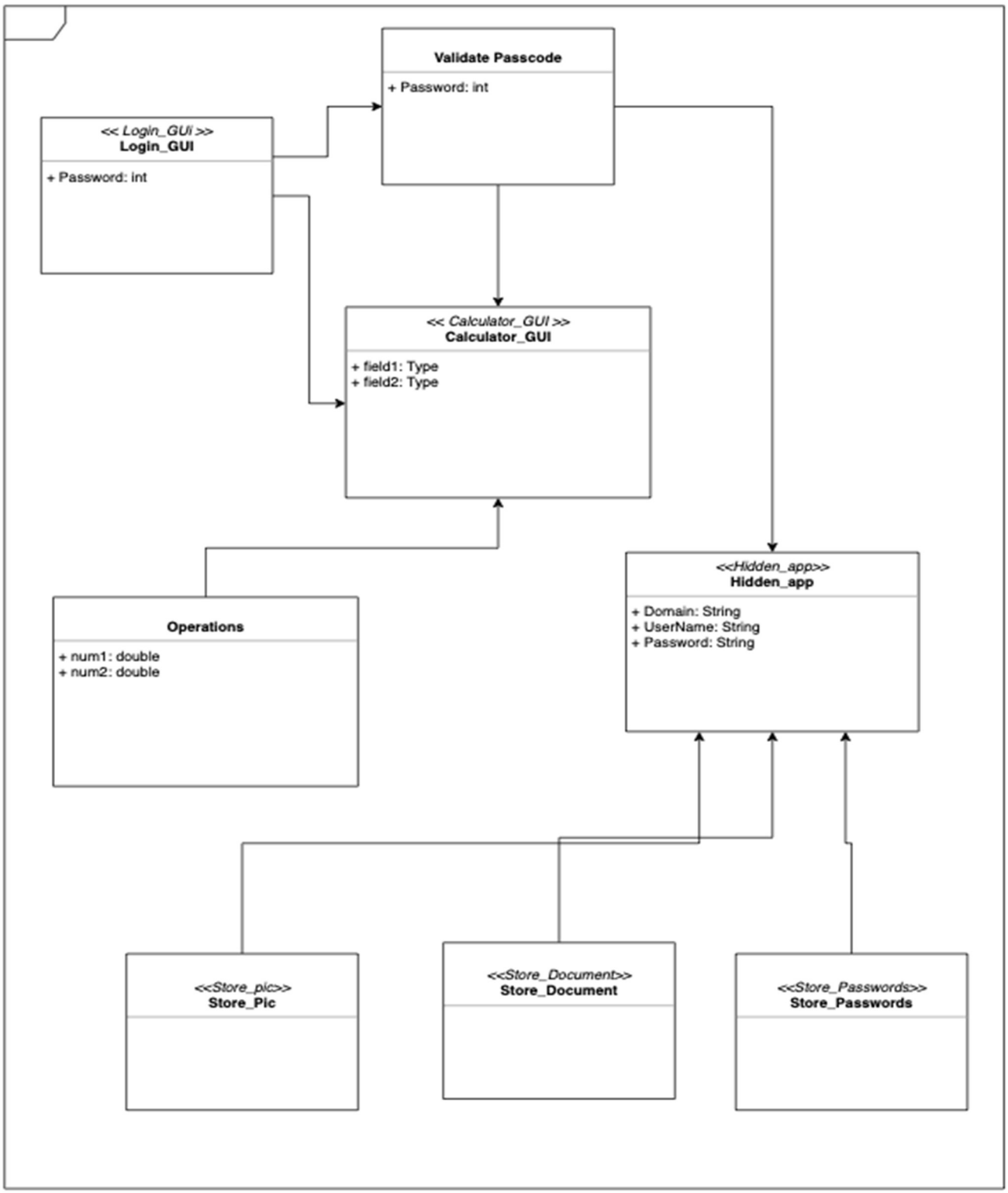
Use Case 4.2: Delete media file in the hidden Interface GUI “Spectre Screen”	
Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the “Spectre Screen” of the app.	
Actor: User	System: Secret Calculator
	0. System displays a “Spectre Screen”.
1. **TUCBW** the user clicked on “store Pics” button.	2. System will prompt a dialog box.
3. The user will click on the “delete image” button.	
4. The user will select the file in the dialog box for deletion.	
	5. System accepts the input and delete the selected file.
6. **TUCEW** the user seeing the media file that is deleted.	
Postcondition: The app is immediately available for other operations inside the dialog screen such as e.g. add image, take photo, delete image.	



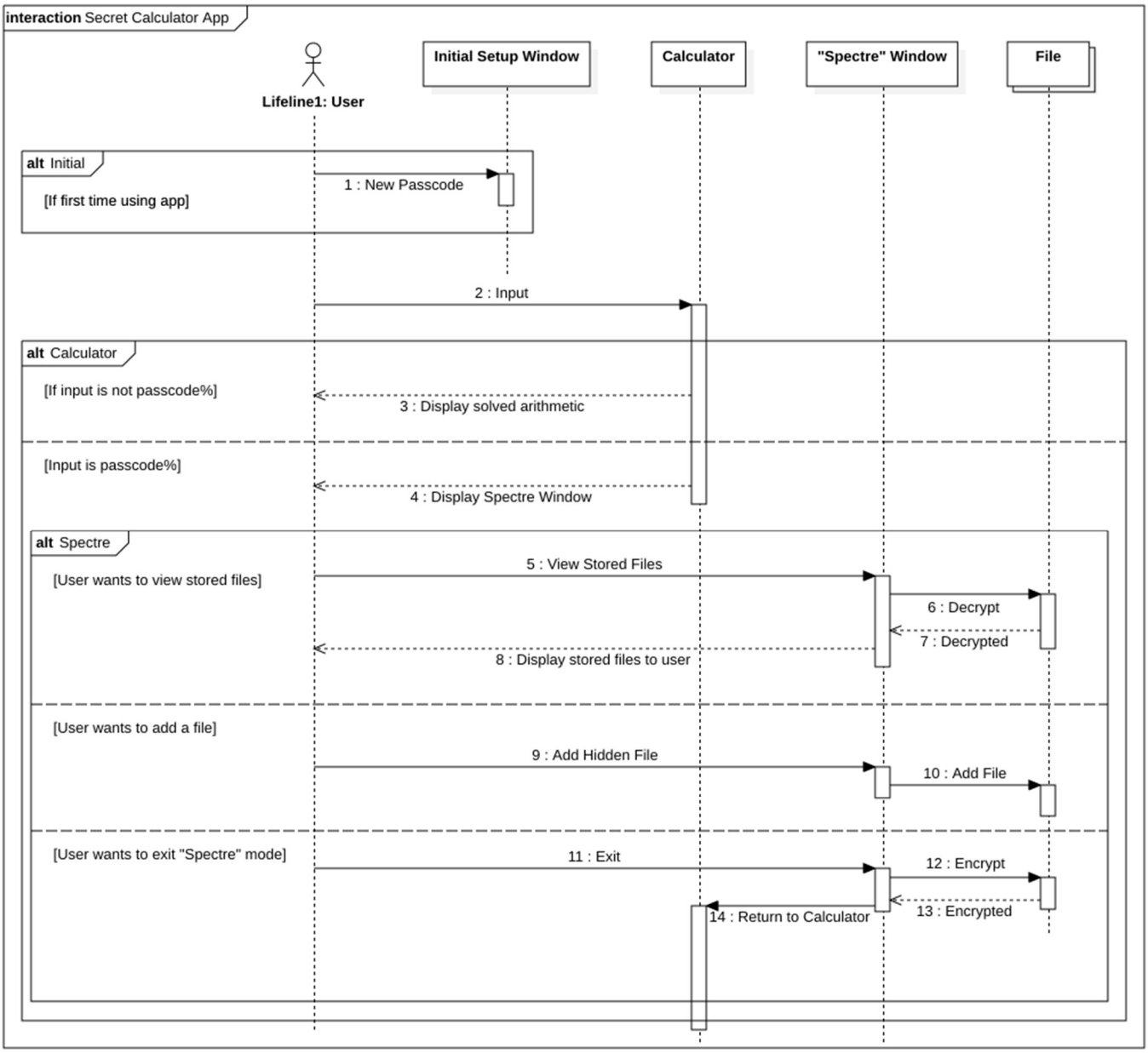
Use Case 4.4: Storing domain, username and password in the hidden Interface GUI “Spectre Screen”.	
Precondition: This use case assumes that the user has turned on the android phone unit, clicked on the app icon, and is on the “Spectre Screen” of the app.	
Actor: User	System: Secret Calculator
	0. System displays a “Spectre Screen”
1. **TUCBW** the user enter information for domain, username, and password in the edit text box.	
2. The user will click on the “save” button.	
	3. System will accept the user inputted information, and will encrypt using AES and generate SHA-256 key.
4. **TUCEW** the user will be able to view the stored information by clicking the button “View Credentials”.	
Postcondition: The app is immediately available for other operations inside the spectre screen.	



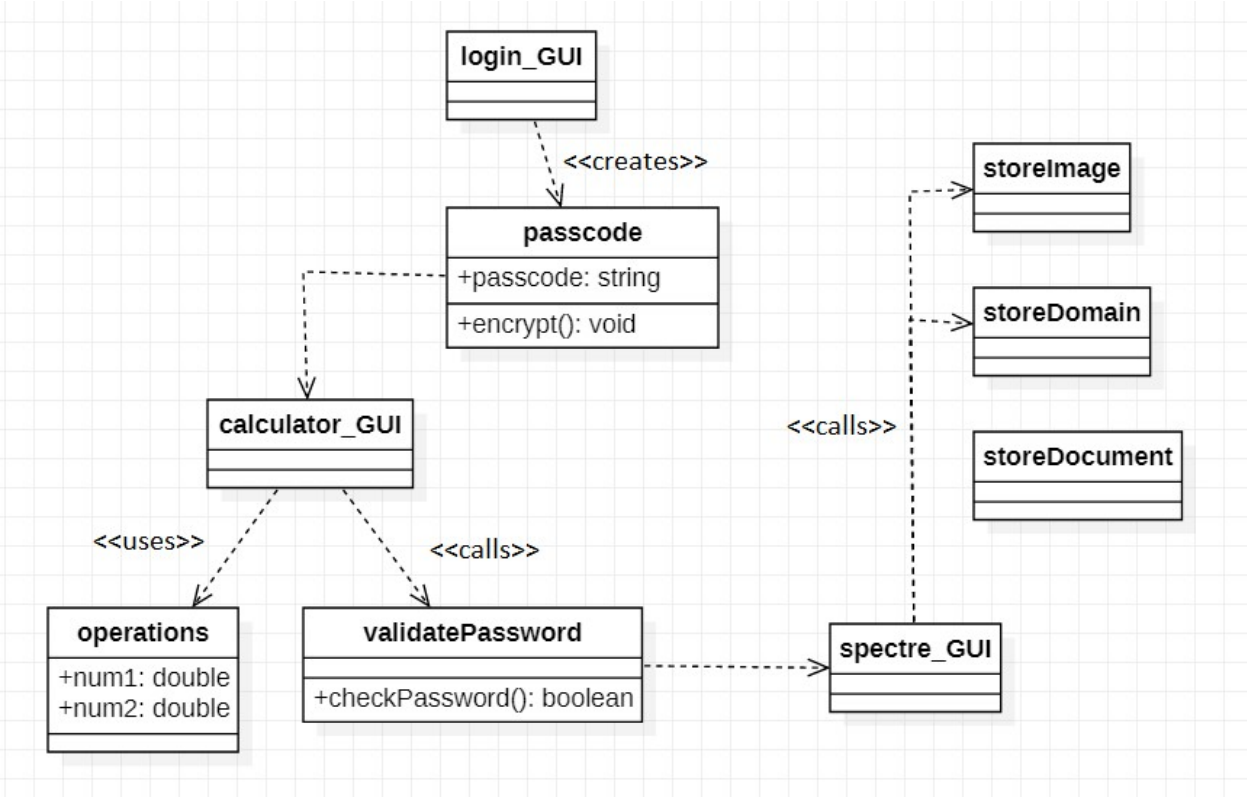
Domain Model



Sequence Diagram



Design Class Diagram



YouTube Link to Demo

https://www.youtube.com/watch?v=E_E79pDgfNY