# System Requirements - Mobile Device Authentication

## General Overview

From a high level perspective, this is a mobile system that will support the following list of functional requirements. These functional requirements are derived from the overall functionality that was previously described, but with modifications to meet the actual needs of the known application domain.

## Use Case Descriptions

Given below are the descriptions of the use case functions.

### Set up\Enter Acceptable Tap Sequence

The first time that the application runs, it will capture the tapping pressure (up and down) and the exact timestamp of each tapping. This data will then be passed along to the ‘Record Acceptable Tapped Data’ function.

### Record Acceptable Tapped Data

This function will take in data from the ‘Set up\Enter Acceptable Tap Sequence’ function. It will then record this data locally in memory. This is going to be the master pattern that is used for comparing actual entry attempts from the phone user.

### Enter Tap Sequence

When the user is ready to use the application after initial set-up, they will use this function on the UI to enter their sequence.

### Record Tapped Data

This function will take in data from the ‘Enter Tap Sequence’ function. It will then record this data locally in a text file.

### Compare Tap Sequence to Accepted Pattern

This function will be triggered by the ‘Set Tap Sequence’ function. It will then get and compare the acceptable sequence up against the current input data for the current tap sequence. It will then run some algorithm to see if the pattern is legitimate or not. Results will be sent to the ‘Display Results’ function.

### Check # of Sequences in local storage

This function will take input data from the ‘Record Tap Data’ function and it will run some logic to determine if it is necessary to offload data to a desktop PC. If it should, then it will call the ‘Offload Data’ function.

### Display Results

This functions sole primary responsibility is to show the results back to the user specifying that the sequence is an acceptable sequence or not. More information needs to be determined if other output or actions are required at this point.

### Offload Data

The primary purpose of this function is to take necessary data that is passed to it from the ‘Check # of Sequences in local storage’ function and send it off to a Desktop PC. The output format will be a simple CSV file that can be loaded into Microsoft Excel.

## Use Case Diagram

Shown below is the full use-case diagram (as it is currently planned to date, may be subject to change).



# User Stories

Described in the section are the user stories and the use cases that will be applied to them.

## Story # 1

*“As a phone user, I want the ability to enter my own personal tap sequence, so that my phone will be able to successfully recognize me by my acceptable pattern for logging in.”*

This story will be supported through the following use cases:

**Set up\Enter Acceptable Tap Sequence**

**Pre-Conditions:**

**N\A**

**Post-Conditions:**

**Story # 2**

## Story # 2

*“As a developer, I need the ability for the phone application to record the acceptable tapped sequence, so that I can ensure there is a base pattern for comparison at future login attempts.”*

**Record Acceptable Tapped Data**

**Pre-Conditions:**

**Story # 1**

**Post-Conditions:**

**N\A**

## Story # 3

*“As a phone user, I want the ability to enter a tap sequence for authentication.”*

This story will be supported through the following use cases:

**Enter Tap Sequence**

**Pre-Conditions:**

**Story # 2**

**Post-Conditions:**

**N\A**

## Story # 4

*“As a phone user, I expect the phone to successfully compare my taping pattern to my acceptable pattern, so that I can ensure that my device can know that it is me.”*

This story will be supported through the following use cases:

**Compare Tap Sequence to Accepted Pattern**

**Pre-Conditions:**

**Story # 3**

**Post-Conditions:**

**N\A**

## Story # 5

*“As a developer, I need the phone to record tapped in data that the user inputs, so that I can ensure that data is valid and working.”*

This story will be supported through the following use case:

**Record Tapped Data**

**Pre-Conditions:**

**Story # 3**

**Post-Conditions:**

**Story # 6**

## Story # 6

*“As a developer, I need to ensure that the phone doesn’t record tapped data indefinitely, so that I can ensure memory is not wasted. I need to check against a threshold.”*

This story will be supported through the following use cases:

**Check # of Sequences in local storage**

**Pre-Conditions:**

**Story # 5**

**Post-Conditions:**

**Story # 7**

## Story # 7

*“As a developer, I need the phone application to offload data to be stored onto a PC, so that I can ensure that data is retained for any future purposes.”*

This story will be supported through the following use cases:

**Offload Data**

**Pre-Conditions:**

**Story # 6**

**Post-Conditions:**

**N\A**

## Story # 8

*“As a desktop PC user, I need the ability to be able to take in tapped recordings from the phone, to help ensure data validity and integrity for any researching purposes.”*

This story will be supported through the following use cases:

**Offload Data**

**Pre-Conditions:**

**N\A**

**Post-Conditions:**

**N\A**

## Story # 9

*“As a phone application, I need to be able to show the accepted or rejection message back to the user for their login attempt, so as to ensure a response was properly acknowledged.”*

This story will be supported through the following use cases:

**Display Results**

**Pre-Conditions:**

**Story # 4**

**Post-Conditions:**

**N\A**

# User Story Review

When going over the user stories, they don’t appear to be too big or complex. In fact, they support their intended purpose of what the actor requires and why. However, that is not to say that they cannot be broken down into further steps, which are commonly known as sub-tasks. As a tool that aids developers, sub-tasks take each story and effectively break them down into easier and simpler components to be worked on, thus eliminating complexity.

# Non-functional Requirements

This section describes the non-functional requirements that are needed to support this system.

* A phone user who understands the concepts of tapping phone screens.
* Any Smart Phone or table device with the ability to capture tapping demographics.
* Any basic Desktop PC or Mac with the ability to accept a CSV file via phone to PC interface.

# Glossary

Tap Sequence – The pattern captured by the phone for pressure points on up and down contacts. It also contains the time taken between consecutive taps.

Tapped Data – Data for tap sequences that is recorded and stored in either memory, a text file, or a computer.

Accepted Tap Sequence \ Pattern – The pattern of a tap sequence that is considered the acceptable pattern for authentication.

Offload Data – The functionality of the system where it transfers stored data from the phone device out to the receiving hardware (generally a PC in this case).