Functions/Classes

### Class: scannedIDEntry

Member variables:

* primaryKey: int that can be used to search within the student ID database for an entry
* IDHash: string. It is the hashed student ID entry. The hash takes in the student ID and the current time in minutes.

### Class: studentInfo

Member variables:

* name: string
* Photo: jpg
* Id: string

# Android App

Always communicate with JSON object

## Scanner

### Function: scanIDByNFC

This function scans another phone’s student ID using NFC.

Input:

* NFC Object

Output:

* An instance of scannedIDEntry. This can be checked against the server’s database.

### Function: verifyID

This function will verify whether or not the scanned scannedIDEntry instance is authentic by consulting the server.

Input:

* Instance of scannedIDEntry ofwjeiorp

Output

* True/False/Error
* studentInfo (obj)
  + Name
  + Photo
  + ID

### Function: showResult

This function scans another phone’s student ID using NFC.

Input:

* studentInfo

Output:

* Void (print info to the screen)

### Function: createUser

This function creates a user in the database.

Input:

* ID, name

Output:

* passcode

## Scanee

### Function: initialize

This function retrieve primary key and secret salt from the server, and store them in the local storage

Input:

* Passcode Obtain from the staff

Output:

* Void (print info to the screen)

### Function: createHashed

This function hashes the inputs and creates a scannedIDEntry,

Input:

* ID, SecretSalt, primaryKey

Output:

* scannedIDEntry

### Function: sendNFC

This function encapsulates scannedIDEntry into an NFC object and send it to the scanner.

Input:

* scannedIDEntry

Output:

* Void (print success/error)

# Server http://protected-u.appspot.com/

### Function: /verifyid called by scanner

This function verified the hashed studentID with the database and return the result.

Input:

* {“primaryKey”:”thekey”,  
   “IDHash”:”thehash”}

Output:

* result: <true/false>  
  name: <studentname>  
  Email: <studentemail>

### Function: /createuser called by scanner

This function sends the primary key and secret salt to the student when they first verified.

Input:

* {“name”:”student”,  
   “email”:”email”}

Output:

* result: <true/false>  
  verification code: <6 digit niumber>  
  Successfully create user!

Function: /verifyuser called by scannee

This function verify the student after scanner has sent the student’s info to the server.

Input:

* {“name”:”student”,   
   “vcode”:”6digits code”}

Output

* result: <true/false>  
  primaryKey: <the key>  
  studentID: <ID>  
  salt: <salt>

Function: /retrieveInfo

This function obtains the student's info with the primary key.

Input:

* {“primaryKey”:”thekey”}

Output:

* result: <true/false>  
  primaryKey: <theKey>  
  name: <studentname>  
  email: <studentemail>  
  studentID: <theID>  
  salt: <theSalt>

Function: /find

This function obtains the student’s info with name and email

Input:

* {“name”:”student”,  
   “email”:”email”}

Output:

* result: <true/false>  
  primaryKey: <theKey>  
  studentID: <id>  
  salt: <salt>

Student creation flow:

/createuser?data={“name”:”name”,”email”:”email”} //scanner

Receive vcode from server //scanner

/verifyuser?data={“name”:”name”,”vcode”,”6digits”} //scannee

Receive primaryKey, studentID, salt //scannee

Within 5 mins

Student verification flow:

/verifyid?data={“primaryKey”:”key”,”IDHash”:”hash”} //scanner

Receive true/false, name, email //scanner

Hash function:

Student ID, time, salt

# Database

Collection1: security

* Primary key
* StudentID
* SecretSalt

Collection2: info

* Primary key
* Name
* Photo
* Email (maybe