|  |
| --- |
|  |
| Face-Away-PC-Lock |
| Modulo 306 |

|  |
| --- |
| Bruno, Luca, Matteo, Jonas  5/8/2020 |

Contents

[Introduzione 3](#_Toc39851992)

[Informazioni sul progetto 3](#_Toc39851993)

[Abstract 4](#_Toc39851994)

[Scopo 4](#_Toc39851995)

[Analisi 5](#_Toc39851996)

[Analisi del dominio 5](#_Toc39851997)

[Analisi e specifica dei requisiti 6](#_Toc39851998)

[Use case 7](#_Toc39851999)

[Pianificazione 7](#_Toc39852000)

[Pianificazione iniziale 7](#_Toc39852001)

[Pianificazione finale 8](#_Toc39852003)

[Analisi dei mezzi 9](#_Toc39852004)

[Hardware 9](#_Toc39852005)

[Software 9](#_Toc39852006)

[Progettazione 9](#_Toc39852007)

[Design dell’architettura del sistema 9](#_Toc39852008)

[Design dei dati e database 10](#_Toc39852009)

[Design delle interfacce 10](#_Toc39852010)

[Interfaccia grafica (con le impostazioni) 10](#_Toc39852011)

[Countdown 11](#_Toc39852012)

[Design procedurale 11](#_Toc39852013)

[Implementazione 11](#_Toc39852014)

[Test 15](#_Toc39852015)

[Protocollo di test 15](#_Toc39852016)

[macOS: 15](#_Toc39852017)

[Risultati test 20](#_Toc39852018)

[Mancanze/limitazioni conosciute 20](#_Toc39852019)

[Conclusioni 20](#_Toc39852020)

[Sviluppi futuri 20](#_Toc39852021)

[Considerazioni personali 20](#_Toc39852022)

[Indice delle Figure 21](#_Toc39852023)

[Indice delle tabelle 21](#_Toc39852024)

[Sitografia 21](#_Toc39852025)

# Introduzione

## Informazioni sul progetto

* **Allievi coinvolti (Progettisti e Sviluppatori):** 
  + Matteo Arena
  + Bruno Gomes
  + Luca Mazza
  + Jonas Bertossa
* **Docenti coinvolti (Consulenti e Clienti):** Geo Petrini e Luca Muggiasca
* **Scuola:** Scuola Arti e Mestieri (SAM) Trevano, sezione informatica
* **Classe:** I3AA.
* **Materia:** Modulo 306.
* **Durata progetto:** 17.01.2020 - 08.05.2020

## Abstract

Le persone si dimenticano frequentemente delle cose. Spesso uno si ricorda di fare qualcosa solo quando è gia troppo tardi e questo gli può costare. In termini informatici, la sicurezza al rispetto dei dati in un PC è una delle cose più importanti da tenere in conto. Se uno se ne va e si dimentica di bloccare il PC, qualunque persona potrebbe provare ad accedere a tutti i documenti presenti. Con l’applicazione che abbiamo sviluppato, Face-Away-PC-Lock, l’utente non dovrà mai più preoccuparsi di problemi di sicurezza come questi. Il programma riconosce le facce di tutti gli utenti con i permessi nelle sue impostazioni e blocca il PC quando questi non sono più d’avanti allo schermo. Semplice, intuitivo, veloce, affidabile e flessibile. Disponibile in tutte le 3 principali piattaforme (Windows, MacOS e Linux) completamente gratuito.

## Scopo

Ci sono diversi obiettivi da raggiungere in un progetto di scala così grande. Questo lavoro ci offre l'opportunità di provare per la prima volta a gestire un grande progetto in team. La suddivisione del lavoro, la comunicazione, lo sviluppo cooperativo, la sinergia e i punti forti e deboli di ognuno verranno messi a dura prova.

Nel progetto vero e proprio ci saranno diverse nuove cose che noi dovremmo imparare, tra queste compare la più importante per permettere il completamento del progetto, imparare il linguaggio di programmazione python. I punti importanti, oltre al programma effettivo, sono la documentazione e i diari che dovranno essere completi ed esaustivi.

Con questo progetto desideriamo realizzare un’applicazione che permetta di bloccare il PC quando l'utente non si trova davanti alla webcam.

# Analisi

## Analisi del dominio

Per ottenere un’idea generale di come si potrebbe svillupare la nostra applicazione, abbiamo messo diverse ore all’inizio del progetto a cercare online dei progetti in cui la faccia veniva riconosciuta.  
Ci siamo però accorti che i progetti erano non esistenti in termini di similitudine al nostro scopo, e in più, quei che utilizzavano dei componenti per tracciare la faccia dell’utente erano molto spesso lenti, complessi, non flessibili e non pratici. In più tutti i programmi richiedevano una conoscenza tecnica abbastanza elevata.

Abbiamo quindi deciso di creare qualcosa che sia comfortevole da utilizzare, che non richieda troppo impegno dal utente per impostare ed eseguire (utile per qualsiasi utente, dal più esperto a quello che usa il computer solo per navigare online), che sia veloce e funzionante.

L’idea è creare un’interfaccia GUI User-Friendly (svilupata in Java perché è il linguaggio in cui noi abbiamo più esperienza per creare qualcosa di qualità) nella quale l’utente possa velocemente aggiungere e rimuovere utenti, aggiungere delle foto da riconoscere e impostare comportamenti specifici del codice.

## Analisi e specifica dei requisiti

Ci è stato imposto da fare un programma che riconosce la faccia di un utente e che crea un countdown visibile al’utente se non lo riesce a vedere più. Questo intervallo di tempo può essere cambiato dall’utente. Il programma deve anche potere supportare/riconoscere diverse faccie. Per ultimo l’applicazione deve anche essere Multipiattaforma, ovvero, disponibile per tutti i sistemi operativi.

Abbiamo deciso anche di aggiungere dei requisiti personali che pensiamo siano oppurtuni da mettere insieme nel nostro progetto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Nome | Descrizione | Importanza | Versione |
| 0 | Interfaccia grafica | Tutte le impostazioni devono essere modificate da un'interfaccia grafica (electron o java) desktop | Alta | 1.0 |
| 1 | Modifica impostazioni in modo sicuro | La modifica delle impostazioni richiede una password | Media | 1.0 |
| 2 | Registrare utente | L'utente deve poter registrare il proprio volto | Alta | 1.0 |
| 3 | Riconoscere Volto | Capire quando qualcuno si trova davanti alla webcam | Alta | 1.0 |
| 4 | Associare volto all'utente | Fare in modo che in base al volto visto dalla webcam il software capisca di che utente si tratta | Alta | 1.0 |
| 5 | Countdown | Quando non viene riconosciuto un volto parte il countdown | Alta | 1.0 |
| 6 | Intervallo di blocco | Dopo x tempo che non è presente il volto il PC si deve bloccare | Alta | 1.0 |
| 7 | Bloccare PC | Il PC si deve bloccare quando non viene riconosciuto un volto | Alta | 1.0 |
| 8 | Aggiornamento progressivo volto | I volti vengono sovrascritti ogni giorno per fare in modo che sia il più duraturo possibile | Bassa | 1.0 |
| 9 | Scelta intervallo di blocco | L'utente può scegliere dopo quanto tempo il PC si deve bloccare | Alta | 1.0 |
| 10 | Registrazione di più volti | Più utenti (sulla macchina) possono aggiungere il proprio volto | Media | 1.0 |
| 11 | Utente corrente | L'utente da riconoscere deve essere quello dell'account | Bassa | 1.0 |
| 12 | Cifratura dati volto | I dati del volto sono cifrati | Bassa | 1.0 |
| 13 | Shortcut | Tramite una shortcut deve essere possibile attivare e disabilitare le funzionalità del software | Bassa | 1.0 |
| 14 | Mail/notifica di blocco | Quando il PC viene bloccato ci arriverà una mail/notifica | Bassa | 1.0 |
| 15 | Mail/Notifica utilizzo | Quando qualcuno usa il PC e non sei tu arriva una notifica/mail | Bassa | 1.0 |
| 16 | Multipiattaformità | Il software deve poter essere eseguito sui sistemi operativi più usati (Windows, MacOS, Linux) | Alta | 1.0 |

## Use case

Figura 1 -- Use Case

## Pianificazione

### Pianificazione iniziale

Figura -- Gantt Iniziale

### 

### Pianificazione finale

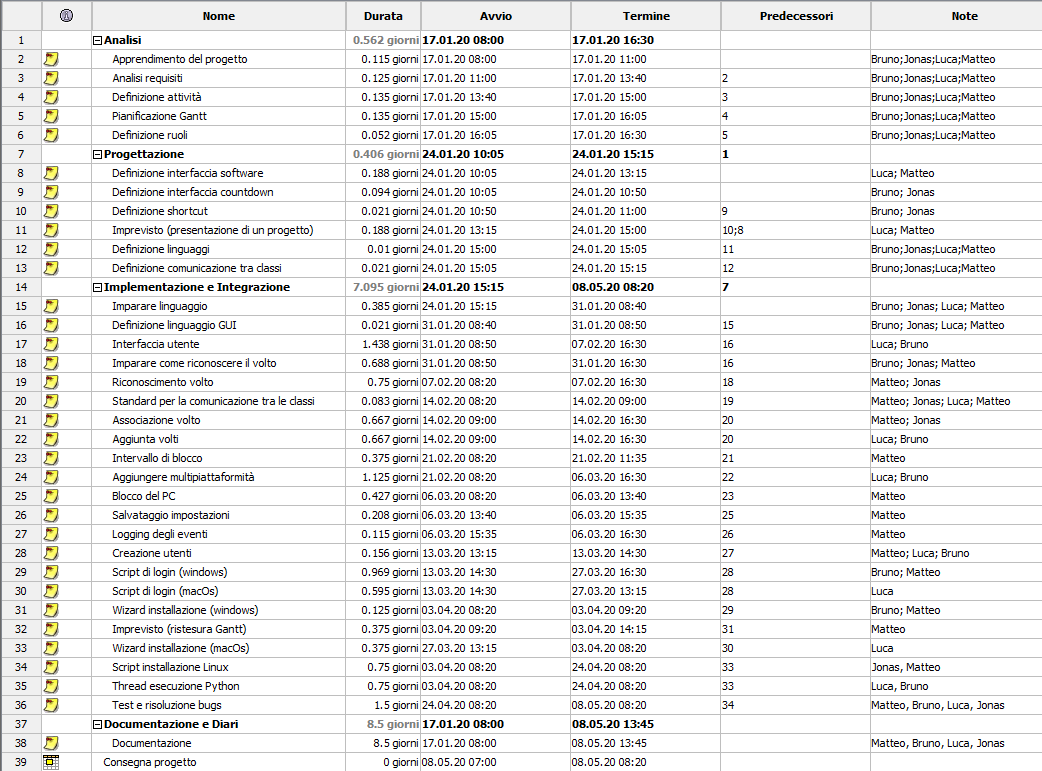
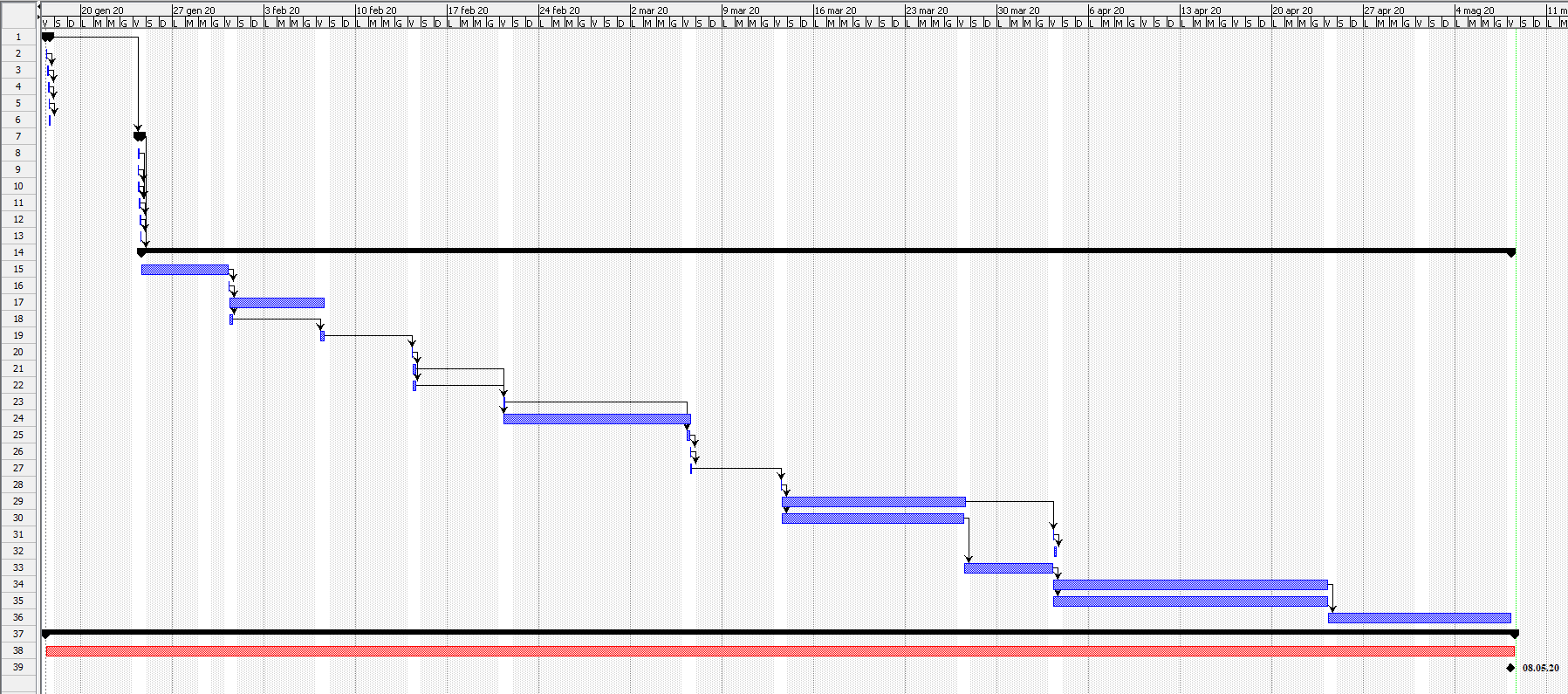
Figura -- Gantt Finale 1

Figura 4 -- Gantt Finale 2



Come si può notare facilmente la durata del progetto è stata estesa, questo allungamento non è stato a causa della difficoltà sottovalutata ma perché abbiamo deciso di aggiungere degli elementi aggiuntivi. Il 5 febbraio infatti avevamo già finito la grande maggioranza dei requisiti ma per rendere il programma più completo abbiamo aggiunto ancora altri requisiti.

## Analisi dei mezzi

### Hardware

Tabella 1 -- Hardware

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nome | Proprietario | Sistema operativo | RAM | Storage | Scheda grafica | CPU | Risoluzione |
| Acer Aspire 5 | Matteo | Windows 10 1909 | 16GB | 128GB SSD 1TB HDD | NVIDIA 940mx | i7 7500U | 1920x1080 |
| MacBook Pro (15-inch, 2017) | Luca | MacOS Catalina 10.15.4 | 16GB | 256GB SSD | Radeon Pro 555 | i7 7700HQ | 2880x1800 |
| Acer Predator Orion 3000 | Matteo | Windows 10 1903 | 16GB | 500GB SSD 1TB HDD | NVIDIA Geforce RTX 2060 super | i7 9700 | 1920x1080 (monitor esterno) |
| Asus | Bruno | Windows 10 1909 | 4GB | 128GB | - | i3 7020U | 1366x768 |
| MacBook Pro 2016 | Jonas | MacOS Catalina 10.15.4 | 16GB | 500GB SSD | Radeon Pro 455 | i7 6820HQ | 2880x1800 |
| VM | Jonas | Ubuntu 19.10 | 4GB | 64GB | - | i7 6820HQ | 2880x1800 |

### Software

Tabella 2 -- Software

|  |  |
| --- | --- |
| Nome | Versione |
| Python | 3.6.6 e 3.7.3 |
| cmake | 3.16.3 |
| dlib | 19.7.0 |
| face-recognition | 1.3.0 |
| face-recognition-models | 0.3.0 |
| numpy | 1.18.2 |
| opencv-python | 4.1.0.25 |
| pip | 20.0.2 |
| psutil | 5.7.0 |
| netbeans | 11.1 |
| JRE | 12.0.2 |
| JDK | 12.0.2 |
| NSIS | 3.05 |

# 

# Progettazione

## Design dell’architettura del sistema

Lo script principale (quello che è sempre eseguito) si chiama faceCheck.py e non ha una directory precisa ma perché abbiamo preferito fare un programma portatile. Una volta avviato lo script viene eseguito in background sulla macchina ospite e ogni secondo accende la webcam per controllare se il proprietario è davanti al computer. Dentro allo script sono presenti molte funzioni per fare in modo che il main possa richiamarle più volte avendo in questa maniera un codice estremamente pulito e molto facile da leggere.

## Design dei dati e database

I dati dei volti sono codificati tramite degli encodings di face-recognition (libreria di Python), per leggere i dati del volto o crearne uno nuovo bisogna passare tramite la funzione dedicata della libreria.

## Design delle interfacce

Figura 5 -- GUI

### Interfaccia grafica (con le impostazioni)

L'interfaccia grafica (in questa schermata ideata per MacOS ma praticamente invariata per Windows e Linux) è pensata per essere molto minimale ma che comprende tutte le informazioni indispensabili. Come si può notare infatti le prime informazioni che si notano sono i nomi delle facce da riconoscere (John Zillo, Pier Telo e George Bo) con a fianco una x per eliminarli. Subito sotto troviamo la categoria General contenente le preferenze dell'utente, più specificatamente:

* Notifica quando il computer viene spento (sul telefono)
* Scelta dei secondi dopo il quale bloccare il computer
* Shortcut per abilitare o disattivare lo script che controlla se qualcuno è davanti al computer
* Dark mode del programma

Infine è presente un checkbox per scegliere se notificare (sempre sul telefono) se qualcuno che non è il proprietario sta utilizzando il computer.

### Countdown



In questa immagine è rappresentata come il countdown una volta che non vedrà uno degli utenti definiti in precedenza, ovviamente dovrà essere completamente funzionante e dovrà essere visualizzato unicamente quando lo script è sicuro che davanti alla webcam non ci sia nessuno.

Figura 6 -- Counter

## Design procedurale

All'interno del progetto dovranno essere presenti due file con gli script (quello in Python e quello in Java), la differenza tra i due sarà che quello in Python si occuperà di riconoscere i volti, bloccare lo schermo e gestire il countdown mentre il programma in java si occuperà di gestire le impostazioni. Le impostazioni verranno poi salvate in un file dove il file Java scriverà e quello Python leggerà.

Come primissima cosa nel file Python bisognerà leggere le immagini contenenti i volti degli utenti, poi lo script potrà partire e, se non riconoscerà alcun volto per un tot di tempo definito nelle impostazioni, esso bloccherà lo schermo.

# Implementazione

La struttura del nostro progetto è divisa in 3 file principali, più precisamente: **“Preferences.jar”, “image.py” e “faceCheck.py”**

Image.py

Questa classe (scritta in **Python**) si occupa unicamente di scattare una foto all'utente (con un'interfaccia grafica dedicata). Questo codice è duplicato anche in **faceCheck.py**, abbiamo optato per questa opzione per non fare troppe richieste alla classe principale e poter gestire al meglio lo scatto della fotocamera.

user = ""

files = []

isempty = True

if len(sys.argv) == 2:

user = sys.argv[1]

print(user)

else:

user = "DEFAULT"

if (not os.path.isdir("Dataset/" + user + "/")):

os.mkdir("Dataset/" + user + "/")

for file in os.listdir('Dataset/' + user + "/"):

files.insert(0, file.strip(".jpg"))

isempty = False

if isempty:

files.insert(0, "0")

else:

files[0] = int(files[0]) + 1

camera = cv2.VideoCapture(0)

while True:

return\_value,image = camera.read()

cv2.imshow('image',image)

if cv2.waitKey(1)& 0xFF == ord(' '):

cv2.imwrite('' + str(files[0]) + '.jpg',image)

break

camera.release()

cv2.destroyAllWindows()

if platform == 'win32':

shutil.move('.\\' + (str(files[0]) + '.jpg'),

'Dataset\\' + user + '\\' + str(files[0]) + '.jpg')

else:

shutil.move('./' + (str(files[0]) + '.jpg'),

'Dataset/' + user + '/' + str(files[0]) + '.jpg')

FaceCheck.py

Questa classe è sicuramente la più complicata e intrinseca del progetto, tutte le operazioni complicate all'interno di essa sono state gestite in funzione, in modo che il codice eseguito potesse essere il più pulito possibile e quindi anche comprensibile e facile.

#### Riconoscimento della piattaforma

A causa di qualche problema (siccome del codice funzionava unicamente su una certa piattaforma e non su un'altra) abbiamo implementato all'inizio del codice un metodo per riconoscere facilmente la piattaforma utilizzata. Abbiamo deciso di cambiare la nomenclatura normale (Darwin per MacOS, win32 per windows e altrimenti Linux) con una più semplice per non doverci ricordare ogni volta come si chiama.

system = ""

if platform == "darwin":

system = "macOS"

elif platform == "win32":

system = "Windows"

else:

system = "Linux"

#### Riconoscimento degli utenti dalle cartelle

Con questo metodo riusciamo a capire quali utenti sono registrati (e quindi compariranno nell'interfaccia grafica), queste cartelle infatti sono all'interno della cartella Dataset (in Source/FaceLock) e le cartelle a loro volta contengono tutte le immagini che gli utenti hanno scattato.

def getUsers(self):

path = './Dataset/'

files = os.listdir(path)

for name in files:

full\_path = os.path.join(path, name + "/")

if os.path.exists(full\_path):

self.users.append(name)

#### Blocco dello schermo

Grazie all'utilizzo della funzione “**lockScreen**” il programma riesce con un solo comando a bloccare lo schermo, qua dentro il codice riconosce automaticamente il sistema operativo utilizzato basandosi sull'array creato all'inizio del codice. In caso che il sistema non venga riconosciuto esso non verrà bloccato e su linea di comando verrà stampato l'errore "Non posso ancora bloccare questo dispositivo".

def lockScreen(self, system):

if(self.system == "Windows"):

ctypes.windll.user32.LockWorkStation()

elif (system == "macOS"):

loginPF = CDLL('/System/Library/PrivateFrameworks/

login.framework/Versions/Current/login')

result = loginPF.SACLockScreenImmediate()

elif (system == "Linux"):

cmd = '/bin/bash /etc/facelock/ubuntulock.sh'

os.system(cmd)

Verificare l'utilizzo simultaneo della fotocamera

Su windows abbiamo avuto un problema riguardante l'utilizzo della fotocamera su più processi, per evitare questo problema abbiamo scritto questo codice che si occupa di controllare se un processo sta venendo eseguito. Ovviamente il processo non dirà che è già eseguito dalla classe FaceCheck perché la fotocamera viene attivata e disattivata e non lasciata continuamente attiva.

def checkIfProcessRunning(self, processName):

for proc in psutil.process\_iter():

try:

if processName.lower() in proc.name().lower():

return True

except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):

pass

return False

Preferences.jar

Preferences.jar fa parte di un progetto NetBeans (ovviamente in **Java**) che permette di cambiare le impostazioni legate allo script FaceCheck. Tutte queste impostazioni vengono successivamente scritte nel file settings.csv (all'interno della cartella Settings) e verrà poi interpretato da FaceCheck per capire come dovrà comportarsi.

# Test

## Protocollo di test

### macOS:

Tabella 3 -- Test

| **Test Case** | **TC-000** |
| --- | --- |
| **Nome** | Interfaccia grafica |
| **Riferimento** | 0 |
| **Descrizione** | Ci deve essere un'interfaccia grafica desktop che permette di modificare le impostazioni. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock |
| **Procedura** | Apro la cartella del software (/Applications/FaceLock/) ed eseguo l'applicazione Settings.app |
| **Risultati attesi** | Apparizione del prompt di input della password. |

| **Test Case** | **TC-001** |
| --- | --- |
| **Nome** | Modifica impostazioni in modo sicuro |
| **Riferimento** | 1 |
| **Descrizione** | Le impostazioni devono essere protette dalla password di sistema. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock |
| **Procedura** | Apro la cartella del software (/Applications/FaceLock/) ed eseguo l'applicazione Settings.app |
| **Risultati attesi** | Apparizione del prompt di input della password. |

| **Test Case** | **TC-002** |
| --- | --- |
| **Nome** | Registrare utente |
| **Riferimento** | 2 |
| **Descrizione** | Deve essere possibile registrare un utente con annessa la registrazione del proprio volto. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock |
| **Procedura** | Immetto la password e clicco su un box utente. Appare un prompt che richiede il nome utente e la registrazione della faccia. Immetto il nome e clicco su *Register Face*. Appare la schermata di fotografia. Clicco [spazio]. La schermata si chiude. Clicco su ok. |
| **Risultati attesi** | Appare un nuovo utente con il nome assegnatogli. |

| **Test Case** | **TC-003** |
| --- | --- |
| **Nome** | Riconoscere volto + associare volto ad utente |
| **Riferimento** | 3;4 |
| **Descrizione** | Capire quando qualcuno si trova davanti alla webcam e associarlo ad un'utente registrato. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. Essersi aggiunti come utente. Aver avviato lo script faceCheck.py |
| **Procedura** | Apro la cartella del software (/Applications/FaceLock/Resources), eseguo lo script faceCheck.py. Appare a terminale la scritta "Loading images...". Mettendomi di fronte alla webcam dovrebbe apparire il nome del'utente al quale mi sono registrato. |
| **Risultati attesi** | Apparizione del mio nome nella CLI. |

| **Test Case** | **TC-004** |
| --- | --- |
| **Nome** | Countdown + Intervallo di blocco |
| **Riferimento** | 5;6 |
| **Descrizione** | Quando non viene riconosciuto un volto parte il countdown e poi si spegne. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. Essersi aggiunti come utente. Aver avviato lo script faceCheck.py |
| **Procedura** | Coprendo la webcam dovrebbe iniziare il countdown. |
| **Risultati attesi** | Apparizione del countdown nella CLI. |

| **Test Case** | **TC-005** |
| --- | --- |
| **Nome** | Bloccare PC |
| **Riferimento** | 7 |
| **Descrizione** | Il PC si deve bloccare quando non viene riconosciuto un volto. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. Essersi aggiunti come utente. Aver avviato lo script faceCheck.py |
| **Procedura** | Coprendo la webcam dovrebbe iniziare il countdown. Alla fine del countdown il pc dovrebbe bloccarsi. |
| **Risultati attesi** | Blocco del PC alla fine del countdown. |

| **Test Case** | **TC-006** |
| --- | --- |
| **Nome** | Aggiornamento progressivo del volto. |
| **Riferimento** | 8 |
| **Descrizione** | I volti vengono sovrascritti ogni giorno per fare in modo che sia il più duraturo possibile |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. Essersi aggiunti come utente. Aver avviato lo script faceCheck.py |
| **Procedura** | Dopo avermi riconosciuto il software dovrebbe aggiungere una mia foto. Rifacendo lo stesso il giorno dopo, la foto del giorno prima viene sovrascritta. |
| **Risultati attesi** | Blocco del PC alla fine del countdown. |

| **Test Case** | **TC-007** |
| --- | --- |
| **Nome** | Scelta intervallo di blocco. |
| **Riferimento** | 9 |
| **Descrizione** | L'utente può scegliere dopo quanto tempo il PC si deve bloccare |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Aumento e diminusco lo spinner dei secondi di countdown. |
| **Risultati attesi** | Avviando lo script faceCheck.py e non facendosi riconoscere viene stampato un countdown più lungo/corto. |

| **Test Case** | **TC-008** |
| --- | --- |
| **Nome** | Registrazione di più volti. |
| **Riferimento** | 10 |
| **Descrizione** | Più utenti (sulla macchina) possono aggiungere il proprio volto. |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Clicco su tutte e quattro i box utente e aggiungo un utente ciascuna, anche con facce diverse. |
| **Risultati attesi** | Avviando lo script faceCheck.py e facendo riconoscere le molteplici facce vengono stampati più nomi. |

| **Test Case** | **TC-009** |
| --- | --- |
| **Nome** | Utente corrente |
| **Riferimento** | 11 |
| **Descrizione** | L'utente da riconoscere deve essere quello dell'account |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Clicco su tutte e quattro i box utente e aggiungo un utente ciascuna, anche con facce diverse. |
| **Risultati attesi** | Mi viene chiesto di aggiungere un utente come quello di sistema. |

| **Test Case** | **TC-010** |
| --- | --- |
| **Nome** | Cifratura dati volto |
| **Riferimento** | 12 |
| **Descrizione** | I dati del volto sono cifrati |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Clicco su tutte e quattro i box utente e aggiungo un utente ciascuna, anche con facce diverse. |
| **Risultati attesi** | Guardando le foto salvate non riesco a riconoscerne l'originale. **Fallito** |

| **Test Case** | **TC-011** |
| --- | --- |
| **Nome** | Shortcut |
| **Riferimento** | 13 |
| **Descrizione** | Tramite una shortcut deve essere possibile attivare e disabilitare le funzionalità del software |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Clicco sull'impostazione della shortcut. |
| **Risultati attesi** | Posso utilizzare la shortcut per abilitare/disabilitare le funzionalità **Fallito** |

| **Test Case** | **TC-012** |
| --- | --- |
| **Nome** | Mail/Notifica di blocco e utilizzo |
| **Riferimento** | 14;15 |
| **Descrizione** | Quando il PC viene bloccato o quando qualcuno lo usa ci arriverà una mail/notifica |
| **Prerequisiti** | Aver installato java, aver scaricato il software FaceLock. Aver avviato le impostazioni. |
| **Procedura** | Clicco sull'impostazione della della notifica. Apro il software, faccio bloccare il pc e lo faccio usare ad un volto sconosciuto. |
| **Risultati attesi** | Mi arriva una mail con all'interno la notifica. **Fallito** |

| **Test Case** | **TC-013** |
| --- | --- |
| **Nome** | Multipiattaformità |
| **Riferimento** | 16 |
| **Descrizione** | L’applicazione deve funzionare su Windows, MacOS e Linux |
| **Prerequisiti** | Aver scaricato il software FaceLock. |
| **Procedura** | Eseguire il programma su 3 piattaforme diverse. |
| **Risultati attesi** | Funzionamento corretto del programma in tutte 3. |

## Risultati test

Tabella 4 -- Risultati test

|  |  |
| --- | --- |
| **Test** | **Risultato** |
| **TC-000** | Passato |
| **TC-001** | Passato |
| **TC-002** | Passato |
| **TC-003** | Passato |
| **TC-004** | Passato |
| **TC-005** | Passato |
| **TC-006** | Passato |
| **TC-007** | Passato |
| **TC-008** | Passato |
| **TC-009** | Passato |
| **TC-010** | Fallito |
| **TC-011** | Fallito |
| **TC-012** | Fallito |
| **TC-013** | Passato |

## Mancanze/limitazioni conosciute

- Password sul file .jar su Windows.  
 - Anche se opzionale, non abbiamo fatto l’interfaccia del countdown.

# Conclusioni

Per concludere il progetto ci abbiamo messo un totale di 8.5 giorni lavorativi, l’equivalente di 68 ore e quindi 11.3 lezioni. Nonostante pensassimo di essere avanti alla fine abbiamo concluso il progetto esattamente entro la data prevista (8.5.2020), questo non credo che sia stato causato dalla complessità del programma o da degli inghippi in cui ci scontravamo ma la nostra voglia di voler aggiungere sempre qualcosa in più.

Questo programma non cambierà il mondo ma sicuramente ci ha aiutato moltissimo a gestire un progetto di gruppo, in cui ci sono aspetti che non avevamo assolutamente considerato o che avevamo sottovalutato. Nonostante qualche complicazione siamo riusciti a completarlo in maniera concisa e ordinata.

## Sviluppi futuri

Potrebbe essere fatta una versione non portatile per windows (Con PATH definita) e l’anteprima del countdown.

## Considerazioni personali

**Matteo:** Personalmente sono molto fiero di questo progetto, sono riuscito a completarlo contando sulla fiducia e la collaborazione dei miei compagni, creando insieme questo programma suddividendoci in modo abbastanza equo il lavoro.

**Luca:** Sento di aver contribuito in modo eccellente allo sviluppo dell’applicazione, siamo riusciti a completarla, con lavoro di squadra e collaborazione, dividendoci i compiti in base a ciò ciascuno di noi era più capace a fare. Avrei preferito la partecipazione più attiva di Jonas, soprattutto nella parte mediana dello sviluppo.

**Bruno:** Mi è piaciuto molto lavorare su questo progetto, mi sono trovato bene a comunicare con i miei compagni e mi è piaciuto l’ambiente lavorativo, in cui se qualcuno aveva dei problemi, un’altro poteva aiutarlo a risolverli. Quest’ambiente è molto vicino a quello delle aziende e ho trovato che sia stata un’ottima sperienza per imparare cose nuove che non conoscevo al’inizio di questo progetto come il linguaggio di programmazione Python.

**Jonas:**

# Indice delle Figure

[Figura 1 -- Use Case 6](#_Toc39851734)

[Figura 2 -- Gantt Iniziale 6](#_Toc39851735)

[Figura 3 -- Gantt Finale 1 7](#_Toc39851736)

[Figura 4 -- Gantt Finale 2 7](#_Toc39851737)

[Figura 5 -- GUI 9](file:///C:\Users\matte\Documents\SAM\3°%20anno\306\Face-Away-PC-Lock\Documenti\Documentazione.docx#_Toc39851738)

[Figura 6 -- Counter 10](file:///C:\Users\matte\Documents\SAM\3°%20anno\306\Face-Away-PC-Lock\Documenti\Documentazione.docx#_Toc39851739)

# Indice delle tabelle

[Tabella 1 -- Hardware 8](#_Toc39851740)

[Tabella 2 -- Software 8](#_Toc39851741)

[Tabella 3 -- Test 14](#_Toc39851742)

[Tabella 4 -- Risultati test 19](#_Toc39851743)

# Sitografia

<https://www.w3schools.com/python/>, *Python Tutorial*, 29-01-2020

<https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html>, *OpenCV-Python Tutorials*, 31-01-2020

<https://pysource.com/2019/03/20/how-to-install-dlib-for-python-3-on-windows/>, *How to install Dlib for Python*, 31-01-2020

<https://pypi.org/project/opencv-python/>, *opencv-python*, 31-01-2020

<https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html>, *OpenCV: Cascade Classifier*, 31-01-2020

<https://stackoverflow.com/questions/54273050/opencv-4-0-0-systemerror-class-cv2-cascadeclassifier-returned-a-result-with>, *python - OpenCV 4.0.0*, 07-02-2020

<https://gist.github.com/radames/effc25355c194bd23bbd1d0cbf87d449>, *Example of Python with Opencv*, 07-02-2020

<https://realpython.com/face-recognition-with-python/>, *Face Recognition with Python*, 07-02-2020

<https://github.com/opencv/opencv/find/master>, *File Finder GitHub*, 07-02-2020

<https://stackoverflow.com/questions/48905775/difference-betweeen-haarcascade-eye-and-haarcascade-eye-tree-eyeglasses-xml>, *node.js*, 07-02-2020

<https://stackoverflow.com/questions/604749/how-do-i-access-my-webcam-in-python>, *How do I access my webcam in Python?*, 07-02-2020

<https://stackoverflow.com/questions/2601194/displaying-a-webcam-feed-using-opencv-and-python/11449901#11449901>, *Displaying a webcam feed*, 07-02-2020

<https://stackoverflow.com/questions/11094481/capturing-a-single-image-from-my-webcam-in-java-or-python>, *Capturing a single image from my webcam*, 07-02-2020

<https://www.quora.com/How-do-I-call-Python-script-from-Java>, *How to call Python from Java - Quora*, 07-02-2020

<https://www.biostars.org/p/61404/>, *Connecting Python to Java*, 07-02-2020

<https://www.geeksforgeeks.org/java-program-open-command-prompt-insert-commands/>, *Java Program to open the command prompt*, 14-02-2020

<https://www.baeldung.com/run-shell-command-in-java>, *How to Run a Shell Command in Java*, 07-02-2020

<https://ss64.com/nt/cmd.html>, *CMD.exe (Command Shell)*, 07-02-2020

<https://docs.python.org/3/library/os.html#os.listdir>, *os - Miscellaneous*, 14-02-2020

<https://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html#command(java.lang.String...)>, *ProcessBuilder (Java Platform SE 7)*, 14-02-2020

<https://docs.python.org/3.6/library/datetime.html>, *8.1. datetime – Basic date*, 14-02-2020

<https://stackoverflow.com/questions/8858008/how-to-move-a-file-in-python>, *How to move a file in Python*, 21-02-2020

<https://stackoverflow.com/questions/11525056/how-to-create-a-batch-file-to-run-cmd-as-administrator>, *How to create a batch file run cmd as administrator*, 21-02-2020

<https://mkyong.com/java/how-to-detect-os-in-java-systemgetpropertyosname/>, H*ow to detect OS in Java*, 06-03-2020