Project Plan

for

Swift Unlock

Version 2.0

Prepared by

Group Name: Team 5 – Safe Tech

Alexis Castellanos	26052984	acastel@umich.edu
Basel Ismail Fawaz	30500096	basself@umich.edu
Dalia Nasr	06702342	nasrd@umich.edu
Leila Ismail Fawaz	84106203	ismaill@umich.edu

Instructor:	Dr. Foyzul Hassan
Course:	Software Engineering 2
Lab Section:	*001/002*
Teaching Assistant:	Dhia Elhaq Rzig
Date:	04/20/2021

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Project Plan V1.0	Alexis Castellanos Basel Ismail Fawaz Dalia Nasr Leila Ismail Fawaz	This is the initial version of the Project Plan. It includes the Gantt chart, along with sequence and class diagrams.	03/10/2021

1 Introduction

The introduction of the Project Plan Document provides an outline of the entire document with purpose, scope, audience, management constraints, and overview of the whole document as it relates to Swift Unlock.

1.1 Document Purpose

This document describes Project Plan for version 1.0 of the Swift Unlock software system. It serves as a detailed guideline for the developers and stakeholders and specifies when requirements need to be implemented through deadlines and sprints.

In this document, a description of the overall software system will be provided, along with the functional requirements, class and sequence UML diagrams, and project schedule/ Gantt chart. The UML diagrams help developers build a solid understanding of the software being built along with the expected features and environment which allows for time estimation and schedule building.

Given the time constraint, the project requirements will be developed and implemented incrementally, and that is what our schedule reflects.

1.2 Product Scope

Facial recognition and visual processing come easy to humans. We see an image, analyze it, and make sense of what we see within milliseconds. Computer Vision aims to imitate our human vision capabilities which is quite challenging since we have not yet completely deciphered human cognition.

Swift Unlock is a facial recognition biometric software that deals with static images captured from a video to allow access to a desktop. We assume that each desktop has a single user whose facial patterns will be used to unlock it. The software will detect a face, “read” the facial features, and use an algorithm to verify the face by encoding it into a facial signature and comparing it with the user’s pre-analyzed facial patterns. If a match is found, only then will the desktop unlock and allow access to the user.

There are numerous applications and uses for facial recognition, and Swift Unlock mainly serves as a way for users to unlock their desktop using a camera, thus saving them the trouble of typing their password every time they need to access their desktop.

1.3 Intended Audience and Document Overview

This document is intended mainly for us, the developers, to meet the specified deadlines in the schedule as well as attending the sprint meetings scheduled. Our audience also extends to Dr. Foyzul who will oversee the development of our software through every stage.

Section 1.0 of the SRS provides an introduction to the general purpose of the document and the software Swift Unlock. For our product overview and high-level functions/features and constraints, please refer to section 2.0. For class and sequence UML models of the functional requirements of the software, please see section 3.0. Finally, please see section 4.0 for the project schedule/Gantt chart.

1.4 Document Conventions

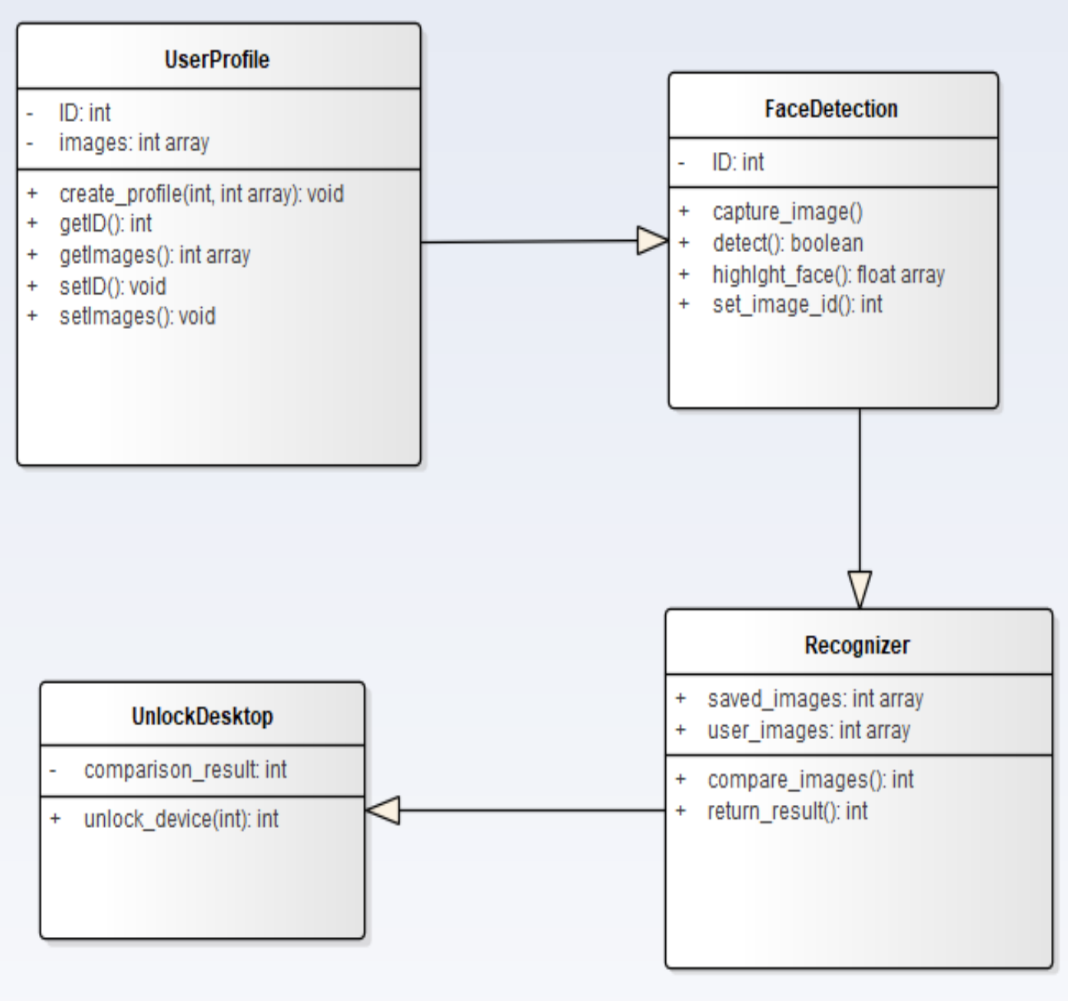
In general, this document follows the IEEE formatting requirements. Use Arial font size 11 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template.

2 UML Diagrams

2.1 UML diagrams

2.1.1 Class diagram

In the class diagram the UserProfile class has the information about the uploaded images by the user. Those images help create a profile for the user to be distinguished from others when trying to unlock his desktop. Once the desktop restarts, the camera opens up and starts giving the computer feedback on what it is seeing. Once the face gets detected, the FaceDetection class highlights it and captures an image of it. The Recognizer class inherits information from both the FaceDetection class and UserProfile class so that it can compare captured images with images present in profile. Once comparison ends the Recognizer class returns a result that is accessed by its child class UnlockDesktop to check whether it should unlock the desktop or keep it locked.

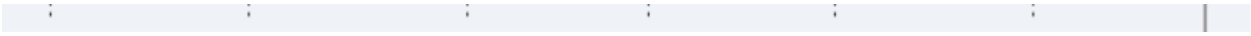


2.1.2 Sequence Diagrams

This sequence diagram illustrates a scenario in which the user runs the program, uploads a set of images, and creates their profile. (Use case 1 and 2)

This sequence diagram illustrates the scenarios that contain the rest of the use cases where the user starts their desktop and their camera turns on to detect their face. After detection, a set of images are taken and saved temporarily to compare against the images in the user profile.

If not recognized (images don’t match), then the desktop locks. It unlocks once the user is back in the frame.



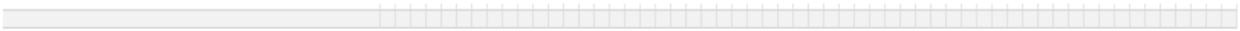
3 Project Schedule

3.1 Schedule Timetable

Task/Functional req	*Team Member*	*Deadline*	*Deliverable*	*Notes*
Access Camera	Alexis	03/01/2021	N/A	
Gather user images	Alexis	03/08/2021	User Profile - F2	100 images/5sec (can be increased)
Design face detection algorithm	Leila	03/15/2021	Design Document - V1.0	
Face Detection	Leila	03/21/2021	Face Detection - F3	Includes live feed image capture and face detection algorithm - F3+F4
Sprint 1	**Alexis, Leila, Basel, Dalia**	**03/24/2021**	**Test detection accuracy**	**Record accuracy scores**

Evaluate face detection	Leila, Basel	03/27/2021	N/A	Take in different parameters like lighting, angles, etc.. to evaluate the algorithm
Enhance face detection algorithm	Alexis, Leila, Basel	03/30/2021	Enhanced Face Detection feature	Remove faults/defects for better performance
Sprint 2	**Alexis, Leila, Basel, Dalia**	**03/31/2021**	**Re-evaluate detection algorithm**	
Update Design	Leila	04/01/2021	Design Document V2.0	
Image Analysis and Comparison	Alexis, Leila, Basel, Dalia	04/07/2021	facial recognition demo	Compare new images captured during face detection to user profile images - F5+F6
Update Design Document	Leila	04/08//2021	Design Document V3.0	
Sprint 3	**Alexis, Leila, Basel, Dalia**	**04/08/2021**	**Demo Results**	
Last minute changes and testing	**Alexis,Leila, Basel, Dalia**	TBD	Swift Unlock Software	
Update any documents necessary	**Alexis, Leila, Basel, Dalia**	TBD	Documents	

3.2 Gantt Chart



4 Results

4.1 Functionalities Implemented

The functionalities of Swift Unlock include:

1. Launching the application and interacting with the graphical user interface (GUI) - The user shall download the project scripts and run them from a python IDE, like PyCharm on Linux. The downloadable file includes a virtual environment folder, venv, that includes the libraries and site packages used, most importantly opencv-python (or cv2) and face_detection library.

2. Create user profile - The user can run the script and select “Set up a profile” from the main menu. They are prompted for a username to enter. The camera used, Logitech C270, will turn on and the user can take pictures by pressing the spacebar, and then press the ESC key when they’re done.This will create a folder in the user’s name inside the project directory with the images saved inside it.

3. Face Detection - Swift Unlock detects faces in real time through the Logitech C270 camera. If a face is not detected, an error message is thrown to the console “User not found”.

4. Image Capture - The user can take pictures of their face for their profile, and the software itself will capture images for comparison.

5. Image Analysis - The software analyzes the facial patterns from the camera live feed.

6. Assessment - After analysis, the software decides if the patterns match those in the user’s profile.

7. Desktop Lock/Unlock - If the face is not recognized, the desktop locks. Once the user is present in the frame again, the desktop will unlock.

4.2 Execution of Test Cases

Test Case	Input	Expected Result	Actual Result	Pass/Fail
Capture Images	Press space bar when creating profile	Images captured	Images captured	Pass
Create user profile	Enter username and capture images	Folder created with same name as username and images added inside it	Folder created with same name as username and images added inside it	Pass
Face Detection	User is not in the frame	User not found	User not found	Pass
Facial Recognition	User with profile created is in the frame	Device remains unlocked, “user found true”	Device remains unlocked, “user found true”	Pass

Lock Desktop	User not in the frame or wrong user in the frame	Device is locked	Device is locked	Pass
Unlock Desktop	User goes back to the frame after leaving	Device unlocks	Device unlocks	Pass

5 Threats to Validity

5.1 Project Limitations

Overall, the project has met our expectations. However, there exists some limitations and challenges.

- A lot of time was spent on locking and unlocking the desktop. We ended up resorting to developing the project on Linux because it was easier to implement the lock/unlock feature. Moreover, we needed to use a lot of packages and libraries which were not all easy to install on Windows due to different file path dependencies. Since we were not all familiar with Linux, our team lead had to handle most of the load to set up the project in a Linux environment. We did not develop the project on Windows or MacOS, and thus could not test it there.
- We tried to resort to multiprocessing (multithreading) to get our project to work faster, but it caused a major bug in our program.
- The integrated webcams on our laptops were not highly efficient. It was a lot easier to use a Logitech C270 for development and testing.

6 Conclusion

6.1 Summary and Significant Results

All of our features passed their tests. The user can create a profile successfully. This will add a folder inside the project directory named after the username, and the images captured are saved there.
The software also allows for facial recognition. If the user matches their profile, the device will remain unlocked, otherwise it locks until the user is back in the frame.

Some of the significant results we found was that it sometimes allows access to someone other than the user, if that person is similar in looks to the user. We found a 98% accuracy match between 2 siblings, for example.

6.2 Plans for Future Work

We plan on enhancing our project during summer time. We would like it to be more efficient (differentiate better between siblings) and faster (take less time to compare and unlock/lock). We also plan on adding more features, like allowing facial recognition with a mask on. In addition, we would like to develop this project again on Windows, and eventually, MacOS.

Appendix A – Data Dictionary

Data	**Type**	**Description**
UserProfile	**Class**	**Class that creates and handles user profile**
ID	Integer	Integer associated with user profile
Images	Integer array	Images of user in uploaded to profile
create_profile()	Function	Creates user profile with ID and images associated with it
FaceDetection	**Class**	**Class that handles F3 and F4**
ID	Integer	ID associated with image captured
capture_image()	Function	Function to capture image once face is detected
Detect()	Function	Function to return a Boolean true if face is detected, false otherwise
Highlight_face()	Function	Function to draw a box around user's face
Recognizer	**Class**	**Class that handles F5**
Compare_images()	Function	Function to compare set of captured images and set of images in user profile
Return_result()	Function	Function to return Boolean value true if compared sets of images are compatible, false otherwise
UnlockDesktop	**Class**	**Class that handles F6**
Unlock_device()	Function	Function to unlock device if return_result() return true
Exception in running script	Exception	If python 3 is not installed, exception error message is printed by OS

Appendix B - Group Log

Team Members	Meeting(s) attended	Time spent	Project Plan section(s)
Alexis Castellanos	All (1 hr/week)	1.5 hrs	1,3
Basel Ismail Fawaz	All (1 hr/week)	1.5 hrs	2
Dalia Nasr	All (1 hr/week)	1.5 hrs	2
Leila Ismail Fawaz	All (1 hr/week)	1.5 hrs	1,3