**Project Postmortem Report**

Class: CS3307A Object-Oriented Design and Analysis

Date: 2019/December/3rd

Group #43

Malachi Kitsa

Mohamad Karami

Junyoung Kim

Harish Nagallapati

Deok-Hwan Kim

### 

### **Project Summary**

In this project, we used a Raspberry Pi and a camera(Logitech C270) to make a smart security system. This security system was going to use facial recognition to allow access into the building and disallow detected intruders. We originally planned to improve the project further by implementing machine learning. Machine learning was going to be used to differentiate the intruder and authorized persons coming into a building. Unfortunately we weren’t able to implement this feature on our project (this will be further explained below).

Our project was mainly developed by using two C++ libraries/toolkits. One of the toolkits we used was Qt. Qt is a free and open-source widget toolkit for creating graphical user interfaces. We decided to use Qt to implement our graphical user interface because Qt was able to run on various software and hardware platforms such as Linux and Raspberry Pi. With Qt, we were able to create an easy to use graphical user interface and implemented features that allowed users to register themselves as a non-intruder. Registered users were able to login to the system and receive an email when an intruder was detected.

The second C++ library/toolkit we used was OpenCV. OpenCV is a library of programming functions mainly aimed at real-time computer vision. We used OpenCV to capture real-time video and was able to detect faces with one of the functions in OpenCV.

Overall, we were able to meet most of the features on our proposal except machine learning.

### **Key Accomplishments**

What Went Right? What Worked Well?

*Group Dynamic*

The group dynamic was a success from the beginning, as all group members were able to decide on a concept for the project very quickly. Meetings were scheduled on an approximately bi-weekly basis, depending on the upcoming stage submission deadlines, and the number of people that attended each meeting was always sufficient. All group members voiced their opinions and concerns for the project at every meeting and there were no conflicts. A Facebook Messenger chat was created so that there could be an ongoing discussion of the project. If a group member could not attend a meeting they could always know what was discussed in the meeting by taking a look at this chat. This chat was constantly being updated by all group members saying the tasks they had accomplished so far, assigning meeting dates and what our future plans were.

For each submission, the amount of work was divided into roughly equal parts for each group member. This also depended on the particular expertise each group member had to offer. For example, one group member had prior experience working with OpenCV, so they were assigned this task for the Stage Four submission.

Overall, all group members were happy with how the project was completed. We were able to agree upon project tasks easily, achieve all task deadlines and did not experience any conflict.

What Was Particularly Useful?

*Jira*

Jira was a very useful service to use as it allowed us to explicitly assign tasks to all group members. Also, the ability to assign priority levels to these tasks let the entire group know how important their task may be. Higher priority tasks indicated to the group that this task should be finished as soon as possible.

*StackOverflow*

Stackoverflow.com is a commonly used website where people ask questions related to programming and have their questions answered by people all over the world. Many of the questions and concerns we came across in our project have already been dealt with by people in the past. Stackoverflow was our go-to resource when a question needed to be answered, or a concept needed clarification. It was mostly used for clarification with the Qt C++ modules since this was the first time any of our group members had used Qt.

What Design Decisions Contributed to the Success of the Project?

The decision to use the Qt and OpenCV libraries were a major factor in the success of our project. Qt is the leading software for C++ desktop applications. It is very robust and possessed all the properties we needed for our particular project. We used its networking and GUI modules mostly. OpenCV also leads its category, real-time computer vision. It is usually used in the Python programming language but also has a C++ version which met our needs.

Overall, using the leading software in their respective categories led us to achieve a great final product for our project, because of how robust but easy to use the softwares were.

### **Key Problem Areas**

The initial configuration of the Pi was one of the biggest challenges in this project. Because it is very different from the environment we are used to, there were issues configuring and working with the Pi. Installing the proper dependencies and program requirements took more time than we anticipated. OpenCV could not be installed using the conventional method; instead, we had to use alternative methods to get it on to our system.

Jira helped the project to be ordered and organized. Working with Jira was very helpful since we could easily see the progress and tasks that needed to be done. However, it was our first time using Jira and the initial setup of it was very challenging. The user stories we made could not be deleted by us, and they needed to be deleted by the TAs which led to some confusion. We could not figure out how to use Jira properly, resulting us to ask multiple questions to the TAs, while the project progression came to a halt each time we waited for the response. Overall, when we finally figured out how to use Jira properly, it became a valuable asset that helped us be significantly more organized.

The biggest challenge we encountered was the machine learning part of the project. Our project plan included the machine learning part which can feed the face objects recorded from the webcam to the convolutional neural network that can detect faces. The entire machine learning part was simply too time consuming for the processor. The processor ended up being occupied by the machine learning algorithm, which then halted every other process on the Pi. Our project ended up abandoning the idea of machine learning entirely to salvage the functional and practical application we could build out of the resources we had.

The problems we encountered while we progressed in our project were surmountable. Many minor bugs and difficulties that were not reported above were diagnosed and fixed in a relatively short amount of time.

### **Lessons Learned**

One important lesson learnt from the project as a whole is integrating different modular parts in forming one robust part that is more powerful and more useful in solving problems. The modular parts comprised of some of the Qt classes used for Graphical user interface, the OpenCV library for real time computing and the raspberry Pi environment.

Additionally, working as a team made this project possible within a limited amount of time. Proper communication and collaborative stood out through out the project.

Each team member was able to improve their technical skills to better understand the materials taught in class by actually implementing the project. The lesson learnt involved proper planning, designing and implementing a project using the different tools available which includes UML diagrams among others. Furthermore, what stood out is the mechanism of seeing a project to completion stage by stage by having deadlines to meet.

All in all, a lot was learned from this project but the points above stood out the most.