Video Security and Alert System

CS2720 - Spring 2013 Project Presentation

Produced by Team 4:
Abu Audu, Levi Bostian, Taylor Brown, Kyle Mueller, Kristen Nielsen

Index

A
What have we accomplished?

Reviewing the Software Requirement Specifications, here is a compiled list of what our group is able to deliver with our motion detection software. Our team was able to create software that delivers what is believed to be the core functionality for the customer. VSAS is packaged with the ability to detect motion and alert a preferred user. Specifically:

SRS - completed requirements

3.1.1 - System shall use a USB webcam feed to detect motion (and subsequently decide when a trespasser is present), capture images of the motion, and record the motion.

3.1.2 - System shall be able to detect motion with the images processed by the system, retrieved from the USB webcam feed.

3.1.3a - System shall be able to detect motion

3.1.3b - System shall record video footage of detected motion until a limit is reached.

3.1.4 - System shall save an image of initial motion capture.

3.1.5 - System shall save video recordings to the web hosting service Dropbox.

3.2.1 - User will access all available options concerning the software through a Graphical user interface.

3.2.2. - System shall present an image of the webcam feed.

3.2.4 - Anytime user does not feel able to complete a task, a “Help” option is always provided

3.2.5 - User will be given direct access to full software documentation.

3.3.1 - Emails are sent to user defined list of email addresses

3.3.2 - User will receive details concerning the captured motion.

3.3.3 - System will be able to record a text description of an event whenever motion is detected.

3.4.1 - System shall allow user to add, edit, delete email addresses from system that will receive motion detection alerts.

B
What are we missing?

SRS - incomplete requirements

**** - All features labeled all have been discussed multiple times with plans on implementation and would be part of software with more time.*

3.2.6 - GUI main screen will display the details of the last recorded event. ***

Plan to complete: To implement this feature, we would simply need to allocate space within the main screen to display information that is already being sent to the user via email. Given time, this text could easily be copied and displayed.

3.2.3 - The system shall provide a way to pause or start/stop recording. ***

Plan to complete: Within our software, motion detection is centered on a ‘while True’ loop that houses if statements. These if statements represent the different states of the software (recording, stop recording, not recording, over threshold), so we believe implementing this feature would involve adding just another if statement taking in this start/stop user input.

3.4.2 - User may adjust a tolerance that determines when to record based on how much motion is detected ***

Plan to complete: To implement this feature, as well as the next, would only require time. The current software has preset parameters concerning tolerance within the code, but they are not open to editing. Our motion class would need to take in parameters concerning the threshold value instead of it always being predefined, and then we would need an additional screen to take in the user inputted values.

3.4.3 - User may adjust a tolerance that determines when to record based on the length of disruption seen ***

Plan to complete: Similar to previous.

3.1.7 - System shall take in parameters concerning extended periods of motion and their alert notifications. ***

Plan to complete: Currently, our team hasn't approached this requirement

3.1.6 - System shall be able to shutdown when a motion continues after a specified amount of time. ***

Plan to complete: Currently, our team hasn't approached this requirement

3.1.6.a - If the system has to halt for reaching the excessive time limit, the system shall inform the user to manually restart the system. ***

Plan to complete: This could be implemented immediately after the ability for user to register an "administrator" user into the system.

Future plans for VSAS outside of our SRS documentation
(Features have been discussed very little within the team with no current plans of implementation)

1. No way to delete video files from Dropbox account except manual login and deletion. This results in Dropbox account filling up with videos that may be not wanted anymore as well as the 2gb limit of a free account being met. Plans for gettings this accomplished would be to create some sort of way that user may select a row in the previous recordings log screen and delete it which would call appropriate methods to delete associated files on Dropbox.
2. Android version of VSAS software. If more time was permitted, this could become an option. We did some research on the subject and found many potential libraries to overcome the risks of the application. More risks would be approached as we will have

to find all libraries that work with Java and Android instead of the Python libraries we are currently using to continue with the same functionality as VSAS desktop application. Plans for getting this accomplished would be to first create some documentation including risk assessment, SRS, data flow diagrams, as well as design sketch ups for the UI. Then after the completed documentation, construction of application may begin.

3. Code cleanup for maintainability, for ability to upgrade. For our team to be able to add more features to the VSAS software, refactoring would have to be completed by the team for code maintainability. Code at current moment is not at a satisfactory level to continue adding features. Plans for this task is to meet as a team and discuss design of all parts of the software. Potential documentation may be created (example: UML diagrams) to help design software features. After discussion is completed, all members of team go into code and refactor code according to what we previously discussed.
4. Create a more robust user interface. With time, we would like to strengthen the interface to filter user input, as to avoid any unforeseen errors. Currently, our e-mail input has checks in place (@, . , string length), and ideally we would extend similar checks to all user inputs such as threshold and time limits when they are implemented.
5. Rework previous recordings list. Currently, the url to access previous video recordings is listed as text within the interface. Ideally, we would edit the software to display them as hyperlinks able to redirect the user to the webpage within their browser.

C

Effects of documentation, documentation, documentation

The Software Requirements Specification document has been extremely helpful in creating VSAS. The large amount of time spent on creating it has no doubt saved time in the inception as well as development of the software. Multiple drafts of the SRS sheet solidified what exactly we wanted the software to accomplish, and logged it for future reference. With this in hand, we could avoid bugging the customer throughout the construction phase of the project. The document also helped track our group's progress as we tackled each requirement statement and planned what to do next. The SRS, once finalized, served as a permanent resource from which we could plan out the construction phase. We could identify dependencies within the requirement statements, separate what's important from what's less so, and identify to the customer exactly what functionality we were able to deliver, and also what's left for further development.

As far as design documentation, we certainly created a few data flow diagrams but given the time constraints on the project, they were mostly created as the software was implemented. The diagrams did though help convey how various parts of the software functioned, such as our dropbox implementation and GUI interface, and this helped keep everyone on the same page within development.

Additionally, a noteworthy item that we believe fell under the documentation category and effectively contributed to our progress was our Trello board. To similar effect as the SRS document, the online board helped us track what our “completes” and “to-dos”, but in a more applied format. Ideas for how to accomplish our requirement statements could be posted and explored, with resources being noted and linked within the same card. Ideas for the software, help with code management, as well as documentation standards could all be documented on Trello, making it a valuable tool for the project.

D
Timeline to completion

Given another week of development time, free from exams, we could finish this project. Meaning, we could finish the functionality laid out in the SRS document. This last sprint has been a struggle for time, and we were not able to fully complete the project. If time were not so much of a constraint, we could finish the project in another week. Time, at this stage, is the only thing stopping us from completion of the project.

E
Future roadblocks

1. Section: B.1: This feature was actually tried to become accomplished. Because Dropbox is abstracted away from the user, there must be an alternative method to delete the video files without logging into the account and manually deleting. A PHP Dropbox API method was attempted and failed because of hosting server limitations. Alternatives would be to use Python by user deleting the videos from the previous recordings log screen, but that means user must be present in front of the system to complete this task. That is reason why PHP implementation was adopted and later failed.
2. Section: B. 2: With our current VSAS desktop application, we are dependent on many Python libraries not written by the software team. Because of limits on time, knowledge and experience, the software team is not able to write these libraries ourselves. With this in mind, equivalent libraries written in the Java programming language will need to be discovered to use with the Android platform. Some research from team has been done for this already, but not all dependencies have been met.
3. Expanding the software to accommodate multiple users may prove an issue, for the software is binded to a single Dropbox account that is again, abstracted away from the user. WIth this setup, if we were to open up access to the account through VSAS, it may prove a security risk to the customers storing all of their videos within the same location. The software may need to be re-implemented with user-specific accounts if this

functionality was ever desired.

F
Team composition comments

The composition of the team enhanced our ability to complete this project. Team 4 was very good with communication. We shared our ideas openly, and decided on the ideas we found best fit our group. Group meetings ran smoothly, will all members contributing well to the project, no matter how long our sessions lasted. Our group did not have very much diversity, but we all knew Python. This turned out to work well, as Python was the language we found best suited to the job. Our group was good not only with the programming aspect of this project, but also the documentation aspect. Each member participated in the creation of almost all of our documents and diagrams. As the project progressed, each member took on tasks and kept each phase of the process moving along well to best meet our goals, both set by the professor as well as by our group.