IDENTIFICATION OF ABNORMALITIES IN LIVE FEEDS IN THE BANKING INDUSTRY

Project ID **18-116**

Software Requirements Specification (SRS)

Uzma Alimudeen IT 15086426

Bachelor of Science in Information Technology (Hons) Specializing in Information Systems Engineering

April 2018

IDENTFICATION OF ABNORMALITIES IN LIVE FEEDS IN THE BANKING INDUSTRY

Project ID **18-116**

Software Requirements Specification (SRS)

Uzma Alimudeen IT 15086426

Supervisor: Ms. Shashika Lokuliyana
Co Supervisor: Mr. Anuradha Jayakody

Bachelor of Science in Information Technology (Hons) Specializing in Information Systems Engineering

Department of Information Systems Engineering/Software Engineering/Information Technology

Sri Lanka Institute of Information Technology

April 2018

DECLARATION

I declare that this is my own work and this SRS does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Uzma Alimudeen	IT 15086426	

The above candidate is carrying out research for their undergraduate Dissertation under my supervision.

Signature of the supervisor: Date

Table of Contents

Table of Figures

2. OVERALL DESCRIPTIONS

This section will cover the overall aspects of the parallel processing module that is to be implemented in the Identification of Abnormalities through live feeds in the banking industry more specifically focused on the ATM environment.

The parallel processing module will look at streamlining the processes that are created by the solution to make it efficient and easy to run on a stand-alone node that is available at a security point. The idea is to make the processes run parallelly to save time and use the available resources as efficiently as possible.

This chapter will cover the entire module, functions and capabilities on a high level.

2.1. Product Perspective

Parallel processing is the sharing of tasks between two or more processors. The concept was created by computer scientists to partly solve the issue of not being able to build more powerful microprocessors than there already is [1]. Using a different approach like parallel processing helps in gaining the advantage of using multiple processors, which would help solve the complex task as effectively as a supercomputer would.

Parallel processing will be used to run process parallelly and not sequentially as it is currently being run. Currently in the video surveillance domain, many studies and systems do not focus on running the algorithms manually for the sole reason they have access to many computers or super computers. The idea behind adapting the concept of parallel processing is for an organization to be able to merge this solution with their existing surveillance system to enhance the results and make the process of video surveillance more effective.

In the context of a banking industry, security control is critical and needs to be available in real time. Parallel processing would help achieve this with the use of the computers at the security station, currently in the use of parallel processing, several personal computers are clustered for the running of the processes. But here we deal with one or two PCs and the concept of clustering can be considered.

There is a very limited research pool on using personal computers to aid in parallel processing, however, most available literature discusses the use of clustering. Clustering is simply connecting the computers together over a network, these computers could have their own memory which they use or could be all using one memory over the network.

However, with the innovation in automation, virtualization, and technology parallel processing can be implemented in the computer’s CPU and GPU, [2].

Sharanjit Singh, Parneet Kaur & Kamaldeep Kaur's paper [3] talks about how to perform better using parallel processing for digital image processing they talk about different forms that parallel processing can take and they apply it for medical image processing using a tool called Taverna and work with it to provide the ideal performance. However, it does not mention speaking of applying it to other fields and many tools like Taverna exist only for critical industries like medicine, however, banks also need to have a way of identifying anomalies from the images they capture, and it is also very important as they deal with the finances of the public.

The solution hopes to push the concept of parallel processing by implementing it with the use of threading and parallel programming in code, for it to be suitable for the banking to industry to implement, for faster anomaly detection and smoother running of processes with as little human contact as possible by using existing resources to its full potential.

The parallel processing modules will streamline the processes of the live feed, human and object modules. By breaking down the individual components into their specific smaller processes for processing, allowing for more processes to be done in less time. Other than waiting for one module to finish its processes for the next to be run, this not only takes time but eliminates the entire concept of real time abnormality identification – which the team is hoping to achieve with the implementation of the solution.

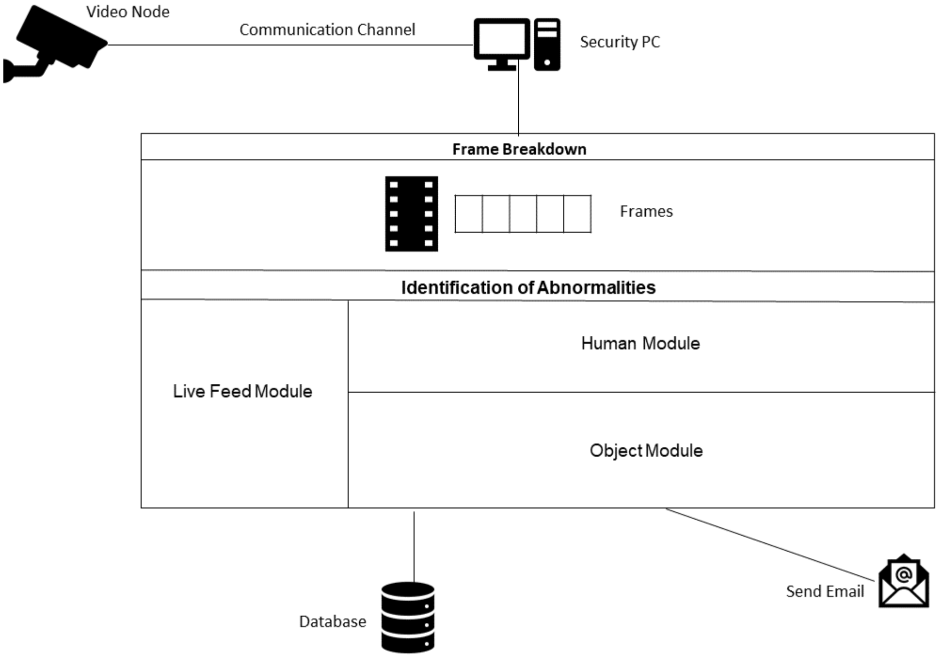

Figure 1 High Level Diagram

2.1.1. Software Interfaces

Visual Studio – for development of the service being created.

Administrative tools – For creating documents and for to draw diagrams during the development life cycle.

- Draw.io
- Microsoft Office 365 applications

2.1.2. Memory Constraints

The module proposes to work with the existing computer(s) available. Since the application will run mostly ad a service, it would be made to work like a background service thus using memory as low as possible. Parallel processing would further reduce the consumption by breaking down the process into manageable threads an i3 processor or higher has been identified as optimal for implementation of this module.

2.2. Product Functions

Parallel processing will open doors to more effective and efficient running of the entire service being created. The different modules incorporated into the service in the identification and notification of abnormalities will be broken down into their individual smaller processes and aligned for smoother processing, with the use of multiple methods, techniques and tools to achiever this.

To ensure real time abnormality identification the processes developed will be run in parallel to the breaking down of the live video feed coming into the system. This maximizes integrity. The processes identifying the abnormalities will also run in parallel to one another.

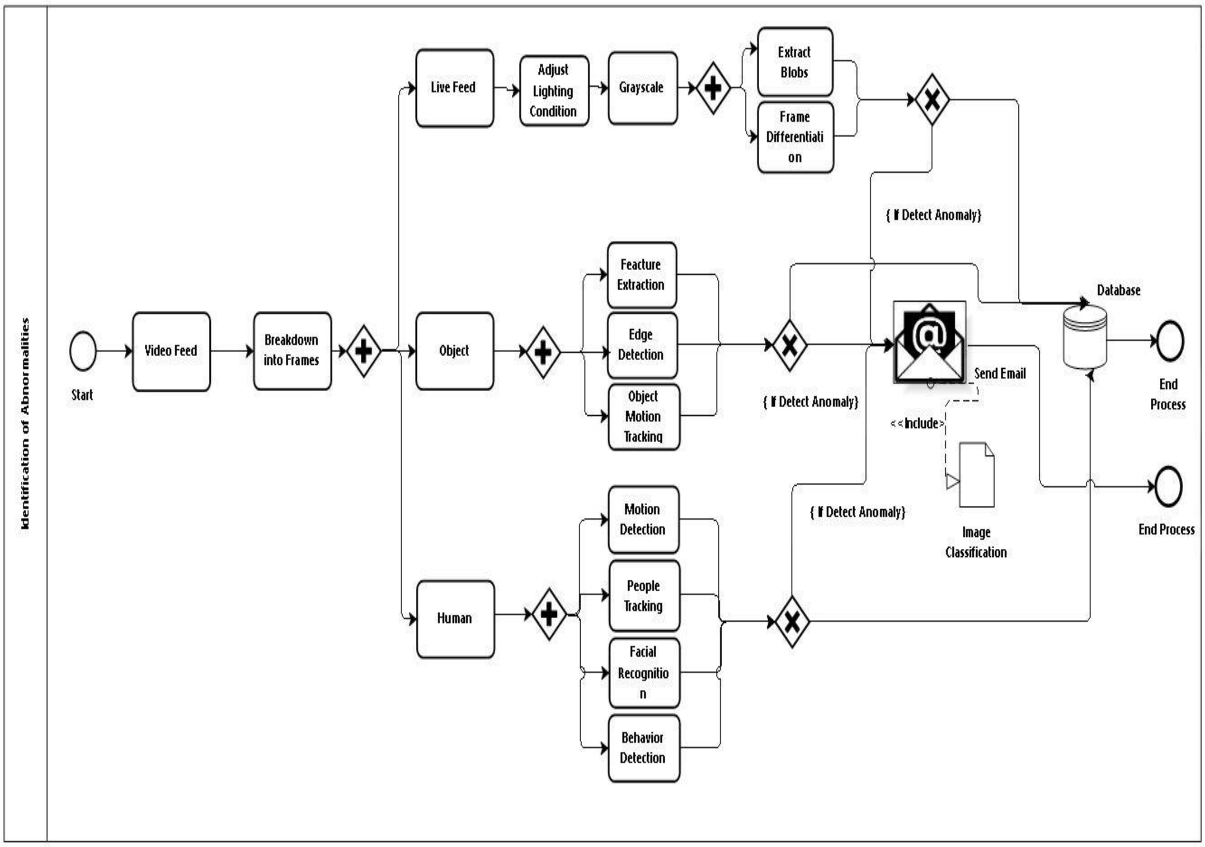

Figure 2 Process Diagram

2.3. Constraints

In developing the module, the constraints come through the available resources, the processing capabilities of the existing system will tell the developers how much they can allocate for processing of the service.

The module will be developed on C#, installed on a Windows PC.

2.4. Assumptions and dependencies

The developers assume the security point hosts a standard PC with and i3 or greater processor, running on a windows machine.

2.5. Apportioning of Requirements

The requirements that were stated are to be implemented fully, the entire solution depends on the back-end service done by the parallel processing module to run smoothly. There are no broken-down requirements of the module that can be apportioned. The entire requirement of breaking down the processes and aligning it for efficient running of the solution.

Additionally, the requirement of processing feed of many other cameras in parallel to the ATM cameras, may be looked at as future enhancements when the solution wants to expand its capabilities.

3. Specific Requirements

3.1. External Interface

3.1.1. Software Interfaces

Visual Studio - The module will use in built classes and components like threading and parallel programming to be more effective. The IDE boasts a user-friendly programming environment to unexperienced developers.

3.2. Classes

A main class will be created to incorporate the parallel processing methods and attributes. The existing C# libraries available for parallel programming can be used for the development of the module.

3.3. Performance Requirements

The system requires real time processing of the live feed to detect abnormalities, and that is key in making the solution successful. The performance of all related modules need to be optimized for the entire service to work well. The Parallel processing module will ensure that the processes of the other modules are created with utmost care in terms of precision of code for it to work as efficiently as possible.

3.4. Design constraints

The design constraints are only limited to ensuring the functionality of the module is met and does not interrupt the processes of the other modules.

3.5. Software System Attributes

3.5.1. Reliability

The module needs to be running always for the service to be running efficiently to avoid discrepancy in results. The module needs to be developed with great care to reduce the down time as much as possible. This module is the back bone of the entire system making it the module needing the highest reliability.

3.5.2. Availability

In terms of restart and recovery there will not be much to keep track of, the module should be set to start the process fresh each time the service is restarted or is booted on recovery mode.

3.5.3. Maintainability

The class containing the parallel processing code could be altered over time to increase efficiency of the service by rearranging the processes or adding/removing processes belonging to the other modules of the service. The back end of the entire service being parallel processing will need to be checked and updated over time for better continuous results.

References

[1] J. Strickland, “How Parallel Processing Works,” *HowStuffWorks.com*, 2008. [Online]. Available: https://computer.howstuffworks.com/parallel-processing.htm. [Accessed: 02-Apr-2018].

[2] A. Klimentov *et al.*, “Digital image processing using parallel computing based on CUDA technology Next Generation Workload Management System For Big Data on Heterogeneous Distributed Computing TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction Digital image processing us,” *J. Phys. Conf. Ser*, vol. 803.

[3] S. Singh, P. Kaur, and K. Kaur, “Parallel computing in digital image processing,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 1, pp. 183–186, 2015.

[4] J. Melonakos, “Parallel Computing on a Personal Computer,” *Biomedical Computation Review*, p. 29, 2008.

[5] H. Tan and L. Chen, “An approach for fast and parallel video processing on Apache Hadoop clusters,” *2014 IEEE Int. Conf. Multimed. Expo*, pp. 1–6, 2014.

[6] D. Black-Schaffer, “Block Parallel Programming For Real-Time Applications On Multi-Core Processors,” Stanford University, 2008.

[7] A. Klimentov *et al.*, “Digital image processing using parallel computing based on CUDA technology Next Generation Workload Management System For Big Data on Heterogeneous Distributed Computing TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction Digital image processing us,” *J. Phys. Conf. Ser*, vol. 803.

[8] A. R. Bhatia, N. M. Patel, and N. C. Chauhan, “Parallel implementation of face detection algorithm on GPU,” *2016 2nd Int. Conf. Next Gener. Comput. Technol.*, no. October, pp. 674–677, 2016.

[9] L. Deligiannidis and H. R. Arabnia, “Parallel Video Processing Techniques for Surveillance Applications,” *2014 Int. Conf. Comput. Sci. Comput. Intell.*, pp. 183–189, 2014.