41492
Engenharia de Software/Software Engineering

universidade de aveiro
departamento de eletrónica, telecomunicações e informática

2

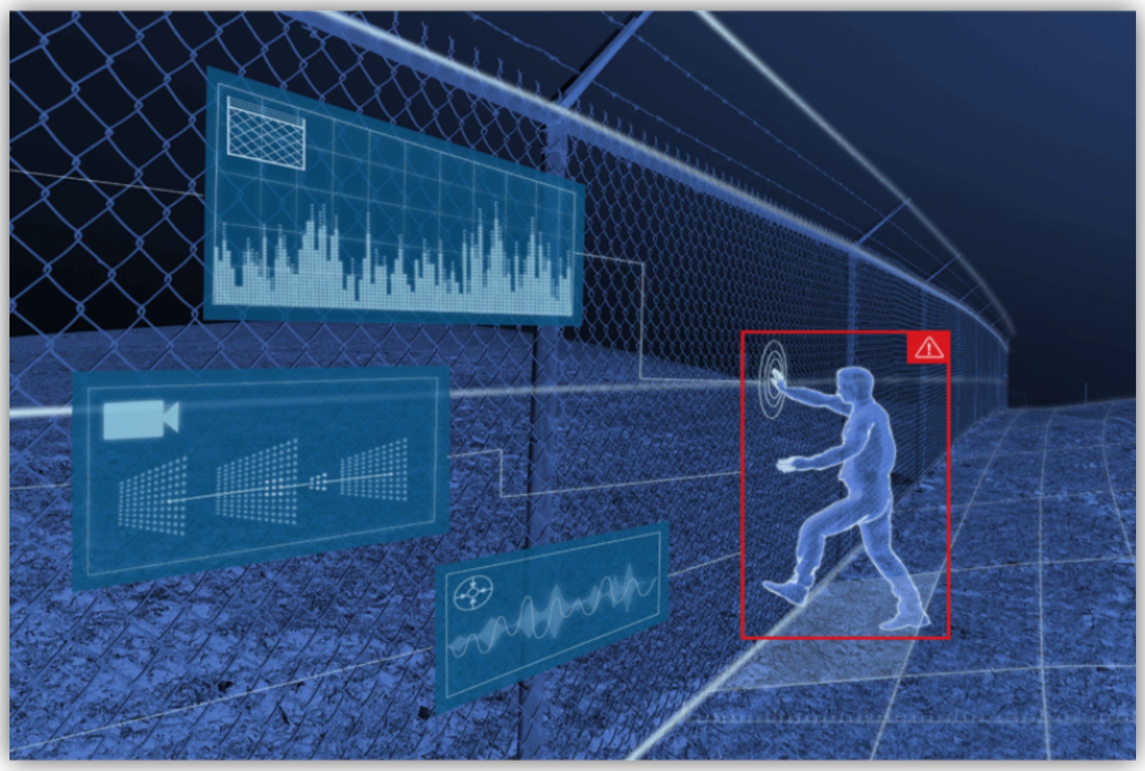
Software Engineering

1st Semester

2022/2023

Group Assignment Project

Human Detection in CCTV Systems



Under the supervision of:
Nuno Sá Couto
Rafael Direito

Table of Contents

1 Document History

Revision	Date	Updated By	Update Description
v0	27-09-2022	Rafael Direito	Initial Version
v0.1	01-10-2022	Rafael Direito	Added evaluation criteria and information regarding the teams' size

2 Motivation

This project will tackle the development of a software solution for a security company named SecCom. SecCom is a company that ensures critical buildings are not broken into, through the installation and operation of CCTV cameras on-premises.

Although, SecCom is still not foresting the technological advancements in the security monitoring field, having several people monitoring the cameras of the most critical buildings.

The goal of your team is to help SecCom with their transition to the digital world, creating an automatic system that can identify intruders without human-intervention and act accordingly.

With their digital transition, SecCom will also install several light and sound alarms on the spaces they are protecting and wishes for them to be automatically activated every time an intruder is detected. Besides this, SecCom also expects that your team develops a top-to-bottom solution they can use manage the cameras and alarms installed on-premises and to manage all their clients and buildings monitored.

3 Base Use Cases and Considerations

3.1 Mandatory

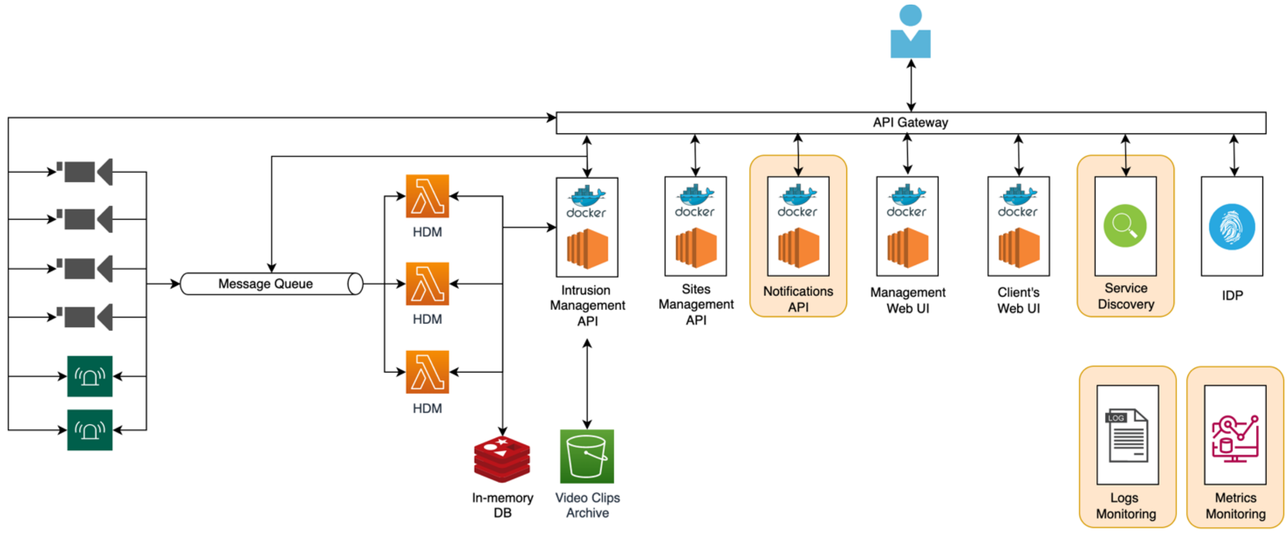
- The CCTV cameras will be continuously recording video and sending frames to the “Human Detection Module” in order to evaluate if there is a human on-premises or not;
- If the “Human Detection Module” detects a human in 3 consecutive frames, you should assume there is an intruder on-site;
- In case an intruder is detected, your software solution should persist the camera recording. To do so, your solution will have to query the camera and request a portion of the recording, starting 3 minutes before the intrusion and ending 3 minutes after the intrusion is detected. This video clip should be stored in a file archival solution;
- If an intrusion is detected, the light and sound alarms should be automatically triggered;
- Your solution should provide a portal to manage all buildings being monitored, register new cameras and alarms and manage them;
- Your solution should also provide a client-targeted portal to obtain data from the cameras and alarms, manage how the client wishes to receive intrusion notifications, and check all events triggered in the client’s building;
- The access to these portals should be mediated through an Identity Provider mechanism, for user authentication and authorization. Please adopt RBAC and a widely known protocol for authentication and authorization;
- The previous web-portals and the IDP should be accessible through an API Gateway;

3.2 Extra

- In case of an intrusion, the system you will develop should notify the building administrators and the police;

- Your solution should also monitor the health of the cameras and alarms installed on-premises and, in case any of them is not working properly, it should raise an alarm addressing this situation;
- All site's cameras and alarms should be registered through a Service Discovery mechanism;
- Your system should store all notifications sent and alarms raised;
- You should also monitor all system's components (metrics + logs).

4 Architecture



4.1 Components and Micro-Services

Component	Description
Human Detection Module (HDM)	This module analyzes the frames sent by the security cameras, to detect if there is a human on-site or not
In-memory Database	This database should be used by the HDM to evaluate if there are N consecutive frames where a human was detected
Video Clips Archive	The Video Clips Archive should store the video recordings of the intrusions. The Intrusion Management API should request the video clips from the cameras and they should be stored in AWS S3
Intrusion Management API	This API will be used to act whenever an intrusion is detected. It will get the intrusion video clips from the cameras, activate the alarms, and trigger a new notification in the Notifications API
Sites Management API	This API should be used to track all properties being monitored, along with all the logic regarding the "owners" of each property. It should make available endpoints for the creation/update/deletion of new properties, creation/update/deletion of new property owners, etc…
Notifications API	This API is responsible for informing the property owners and the police whenever an intrusion is detected
Management Web UI	Via this graphical interface, the platform's admins should be able to see all properties being monitored, the intrusions that took place, a list of each

	property cameras and sensors (also their health), and all data regarding the platform's clients. Basically, this UI is used to manage the entire platform
Client's UI	This UI is solely offered to the owners of the properties. Through it, the property owners should be able to see a listing of all cameras, sensors, intrusion events, etc. Besides this, the property owners should be able to update their information through this UI
Service Discovery	Every time a new camera or sensor is added to a property, the camera/sensor should register itself in the Service Registry, listing how it can be accessed
Identity Provider (IDP)	The IDP provides authentication and authorization mechanisms for all the aforementioned APIs and UIs
Logs Monitor	All system logs should be centralized in this entity
Metrics Monitor	All system metrics should be centralized in this entity

4.2 Suggested Technologies (but NOT MANDATORY)

Component	**Suggested Technologies**
Human Detection Module (HDM)	**This module will have to be scaled and descaled on demand**. We suggest using serverless computing to implement it, for instance AWS Lambda. Regarding the Human Detection algorithms, students can use several technologies: OpenCV, Pytorch, Tensorflow, Keras, etc.
In-memory Database	Suggested Technologies: REDIS, Amazon ElastiCache, Memcached, etc
Video Clips Archive	Suggested Technologies: AWS S3 (**highly recommended)**, FTP Server, etc
Intrusion Management API	Suggested Technologies: FastAPI, Django, SpringBoot, Ruby on Rails, Express.js, etc
Sites Management API	Suggested Technologies: FastAPI, Django, SpringBoot, Ruby on Rails, Express.js, etc
Notifications API	Suggested Technologies: FastAPI, Django, SpringBoot, Ruby on Rails, Express.js, etc
Management Web UI	Suggested Technologies: React, Angular, Vue.js, Plain HTML + JS + CSS, etc
Client's UI	Suggested Technologies: React, Angular, Vue.js, Plain HTML + JS + CSS, etc
Service Discovery	Suggested Technologies: AWS Cloud Map, Consul, Etcd, Zookeeper, Eureka, DNS Service Discovery, etc
Identity Provider (IDP)	Suggested Technologies: Identity Server, Keycloak, Ory, AWS IAM
Logs Monitor	Suggested Technologies: ELK Stack, Fluentd, Graylog
Metrics Monitor	Suggested Technologies: TICK Stack, Prometheus + Grafana, Zabbix, Nagios, etc

5 General Project Guidelines

In this project, you should:

- Deliver working software applying an **Agile/Scrum** software development methodology
- Apply a **microservice architecture** and best practices
- Use as much as possible the free tier **AWS services**
- Document all APIs using the **OpenAPI** specification
- Having a **CI/CD pipeline** with Automated Unit Testing, and **integrated with the SCM**, is **considered as plus**

You must use:

- **Docker**: To package most of the system's components
- **Git**: Source Code Management
- **JIRA**: To manage and communicate project sprints, user stores and other relevant issues
- **Docusauros**: To create a documentation page / wiki. In this page you should list all the deliverables (see next section)

6 Deliverables

1. Application **detailed architecture** diagram (APIs, databases, etc) + **workflows**
2. **User Stories detail** (for the ones implemented) by sprint – **to be presented in weekly sprint Reviews**
3. **Application code** (the course lecturers should have access to your Git repository)
4. **Demo** of the system (video that should be included in the final presentation) **-** to be discussed later
5. Final presentation

7 Teams

Each team must be composed of **6 elements.** Taking into consideration the number of students enrolled in this class, it is predictable the existence of groups with only **5 elements. This situation will have to be addressed with the teachers** during the creation of the teams. If a student is not enrolled in any team, **he will be assigned a random one**.
The composition of each team must be communicated to the course teachers by the end of the second week of the semester.

8 Evaluation

The project scope (backlog) will be divided into several iterations (sprints) in order to achieve the Minimum Viable Prototype (MVP) envisioned and committed by the team. Increments in project/system ́s functionalities between sprints are mandatory.

The final project's grade will be based on the project́ s final demonstration, interim presentations and in the fulfilment of planning objectives defined for each sprint, observed through agile ceremonies and code review.

8.1 Project key evaluation factors

- Follow a microservice architecture whenever possible;
- Group applied and Agile / Scrum development methodology for project implementation, planning and team alignment. Evidence of agile ceremonies and backlog grooming activities performed must be shared;
- Correct use of Source Code Management Tool (ex: Github / Gitlab / *) and branching strategy;
- Use of cloud services (including serverless architectures) to support project delivery;
- Overall MVP delivery quality and value delivered in user stories implemented.

8.2 Individual grade

Individual grade will be based on individual performance:

- In Sprint Review presentations (and other agile ceremonies);
- Contribution in Project implementation and Delivery (Git commits will be part of one's individual evaluation);
- Peer feedback.

The evaluation guidelines will be further defined by the course teachers and the students will be informed of the final guidelines via the university's E-Learning platform.

9 FAQ

[Question]
Which are the camera/alarms-level metrics that the Client UI should make available?

[Answer]
You have total control of the metrics you should provide via the Client UI. Although, there are a few questions that must be answered:

1. Is the camera alive?
2. Is the camera streaming video?
3. How much bandwidth is the camera is using?
4. Are the alarms alive?

[Question]
…

[Answer]
…