bustedAI

Software Engineering Course Project

Hirica Ioan Alexandru
Grosu Ilinca
Tabacaru Andrei
Buzescu Alexandru

CONTENTS

CHAPTER

1

PROBLEM STATEMENT

These days we might find it though to be sure that our belongings are secured from external threats. Even the most modern technologies we use fail us some times. With the ever-increasing number of strategies employed by people with bad intentions, almost anyone feels the need to install CCTV systems in order to monitor their home 24/7. But, even if you have the recordings it is a tedious task to watch the entire video in order to catch the people lurking around your household.

Since the early days of the "AI Winter" (1960-1980) artificial intelligence has come a long way and evolved to a point that people even fear being replaced by it. It doesn't have to be that way, we can use it in order to make our life easier by using it's power to replace unnecessary manual labour such as - watching a long video in order to catch the burglar! That's where the vision of our application was born. While it might not be revolutionary, bustedAI can be really useful with all the people's profile recorded by your camera being a click away from you.

Our application uses the YOLO (You only look once) algorithm[1] to fit the people from your recording into "boxes" that are then passed to the SORT (Simple online and realtime tracking) algorithm[2] to check, based on all previous information, if there is a new person on the screen or one that has already been seen. For more information check the articles listed in the bibliography.

CHAPTER

2

FUNCTIONAL DECOMPOSITION

Perspective	Module	Description	Functionality	Implementation priority
User	User account	First focus of the project is for the user to create accounts on the platform. By Creating an account the user can choose a subscription option or the free trial. Attribute categories of the user account: A. Personal information (e.g. Name, Email, Picture) B. Role (Admin, User)	Register	Sprint 1
			Login	Sprint 1
			Logout	Sprint 3
			Choose Subscription	Sprint 3
	Video Import	The main focus of the project is for the user to be able to send a video file and receive an archive containing image files with with each human. The User should be able to choose, depending if he wants the archive to contain only humans or humans and cars.	Choose File	Sprint 2
			Choose option: • humans • cars • humans and cars	Sprint 3
Application	Computer Vision System	The Core part of our application is for the application to be able to do what it promies. It receives the video file from the Video Import Controller and then applies the algorithm described above to create and send back the archive containing the images.	Receive File	Sprint 1
			Send File	Sprint 1

CHAPTER

3

NON-FUNCTIONAL REQUIREMENTS LIST

3.1 Performance

- Realtime analyzing of video samples.
- Any page should successfully load in less than 3 seconds.
- The application should not have a maximum number of accounts registered.

3.2 Data Integrity

- Users can see only their own data.

3.3 Security

- 2FA Login.
- Passwords are not stored in database, only hash strings.

3.4 Usability

- Users should be able to logout from any page.
- Users should be able to use the application only if they are logged in a registered account.
- Users should be able to get their archive in maximum 4 clicks.

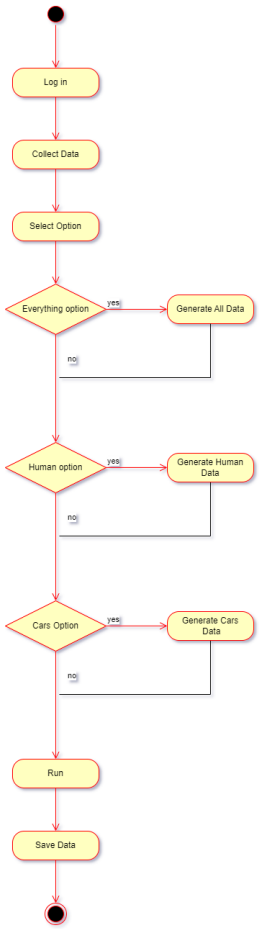
3.5 Compatibility

- All the features should be available on any device, browser and operating system.

CHAPTER

4

ACTIVITY/STATE DIAGRAM



CHAPTER

5

PRIORITIEZD PRODUCT BACKLOG

5.1 Product Backlog

5.1.1 User Account

- Task 1: Login Component with server database (Sprint 3)
- Task 2: Register Component with server database (Sprint 3)
- Task 3: Add routing to pages (Sprint 3)
- Task 4: Home Page with different animation (Sprint 3)
- Task 5: Demo Page with how to use box and video (Sprint 3)
- Task 6: About Page with explaining video and info text boxes (Sprint 3)
- Task 7: Added animation to About page (Sprint 3)
- Task 8: Added footer component with icon styling (Sprint 3)
- Task 9: Added animation to Login Register pages (Sprint 3)
- Task 10: Home Page with Video Import add spin animation when request is loading (Sprint 3)
- Task 11: Authentication: Login (Sprint 1)
- Task 12: Authentication: Register API (Sprint 2)
- Task 13: Document about using Browser Module Animation (Sprint 1)
- Task 14: Document about using Lottie Files (Sprint 1)
- Task 15: Document about how to use hash codes with passwords (Sprint 1)

5.1.2 Video Import

- Task 1: Home Page with Video Import componet calling the video import API from backend (Sprint 3)
- Task 2: Video Import: Get request from api a .zip file from AI algorithm and download it (Sprint 2)
- Task 3: Video Import: Link you file to run the python script (Sprint 2)
- Task 4: Video Import: Choose your file from your pc (Sprint 2)
- Task 5: Document about CLI.Wrap (Sprint 1)

5.1.3 Computer Vision System

- Task 1: Integrate sort algorithm into detection (Sprint 2)
- Task 2: Add gpu option to detection (Sprint 3)
- Task 3: Zip files from the Output folder (Sprint 2)
- Task 4: Cleanup of unused code in detection (Sprint 3)
- Task 5: Implement unit tests for detection (Sprint 3)
- Task 6: Adapt algorithm to work with .tmp files (Sprint 2)
- Task 7: Save cropping of images for detection (Sprint 1)
- Task 8: Implement detection for detection (Sprint 1)
- Task 9: Implement cropping for Detection (Sprint 1)

5.2 User Stories

5.2.1 Sprint 1

- As a user I want to have the best security and my password can not be obtained by anyone
- As a user I want to see animations in the application
- As a user I want to have .NET libraries to easy out the possibile operations

5.2.2 Sprint 2

- As a user, I want to have an api that loads a video from my computer and give me a zip of images with people
- As a user, I want to be able to have register component
- As a user, I want to be able to have a log in component
- As a user, I want to be able to have a download link to the folder with the images

5.2.3 Sprint 3

- As a user, I want to have a log in and register pages
- As a user, I'd like to see more infos about the algorithm that is behind all the process
- As a user, I'd like to simply understand the algorithm with explaining video
- As a user, I'd like to have the folder with the wanted images as simple as it can be
- As a user, I'd like to see a How to use/Demo page
- As a user, I want to have animations and smooth transitionns between pages
- As a user, I want to be able to log out of the application

5.3 Sprints

5.3.1 Sprint 1

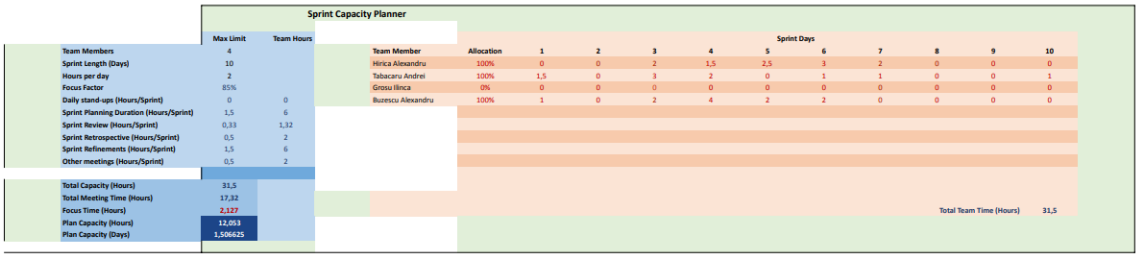


Figure 5.1: Sprint 1 Capacity Planner

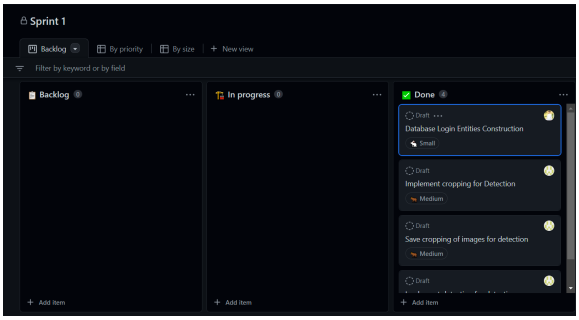


Figure 5.2: Sprint 1 Backlog

Dates of the sprint

November 30, 2022 - December 14, 2022

Team capacity

The team capacity was 75%

Sprint achivements

- Fully documented about the best libraries to use for .NET and Angular

What went well

In sprint 1, we had proper information about the YOLO and SORT algorithm as well as the frameworks we used and the best libraries for doing what we dreamed about

What could be improved

Communication and better understanding of the flow of jointing every step of the application – > From BE – > Detection Algorithm – > FE

What we commit to improve

We commit to improve in the next sprint the start of the application and have a solid BE env to link with the algorithm

5.3.2 Sprint 2

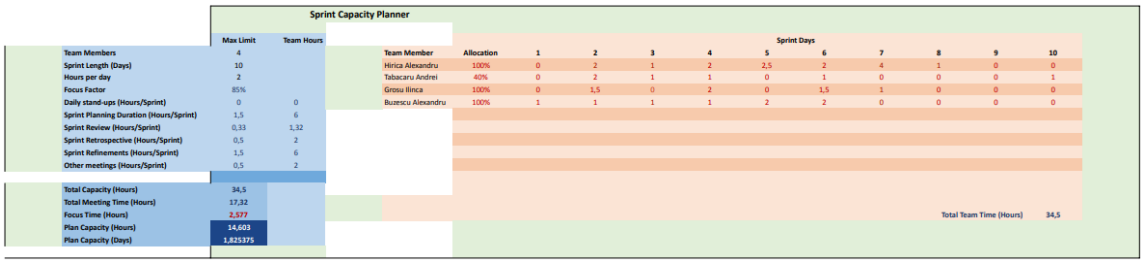


Figure 5.3: Sprint 2 Capacity Planner

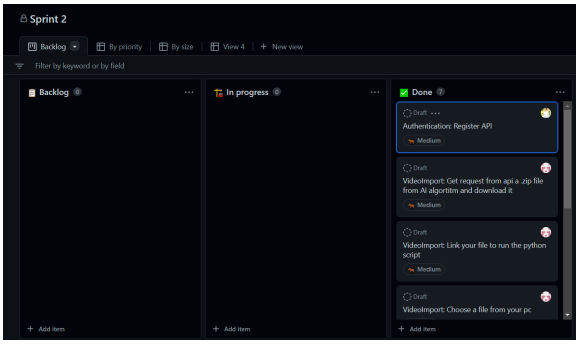


Figure 5.4: Sprint 2 Backlogr

Dates of the sprint

January 5, 2023 - January 19, 2023

Team capacity

The team capacity was 85%

Sprint achivements

- Designed the api for taking a video from computer and load the folder of images
- Designed the api for authentification
- Designed the api for login

What went well

In sprint 2, we fully established a solid BE with link to the algorithm. The algorithm in sprint 2 can now detect humans and when calling the script the output folder containing the images appear

What could be improved

Having a clarity of the steps to be taken to improve and shortage the time spent while implementing the code

What we commit to improve

In the next sprint, we commit to have a fully working fe flow and to have a proper shape of our application

5.3.3 Sprint 3

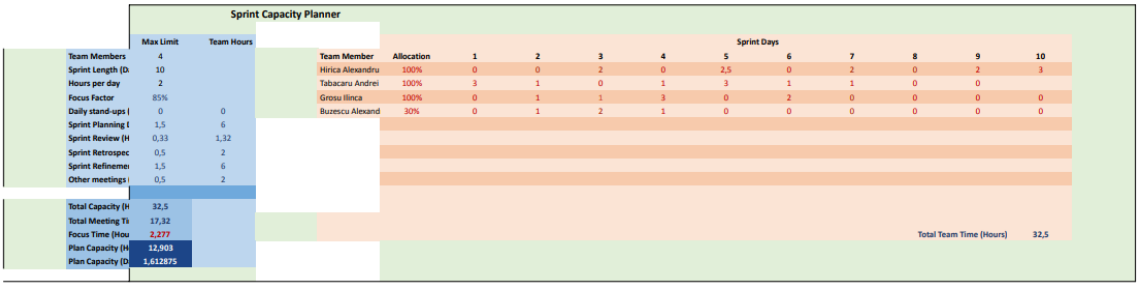


Figure 5.5: Sprint 3 Capacity Planner

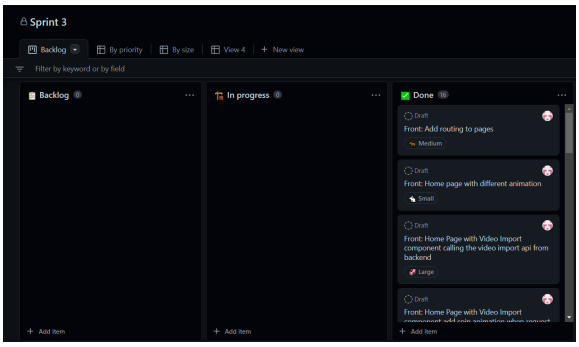


Figure 5.6: Sprint 3 Backlog

Dates of the sprint

January 20, 2023 - February 1, 2023

Team capacity

The team capacity was 82,5%

Sprint achivements

- Designed Login Page
- Designed Register Page
- Designed Home Page with responsive buttons calling Video Import api
- Designed About Page with info about algorithm
- Designed Demo Page with steps and video
- Designed Spinner Component calling when the request is loading
- Implemented Animations with Lottie Files
- Implemented transitiong between pages with Browser Module Animation

What went well

In sprint 3, we fully encounterd the FE and be flow as well as the important algorithm part of human detenction and had a final shape of our application. Moreover, the jointing between the Artificial Intelligence and Web Development has beautifully resulted in bustedAI, a project that we think can be expended in so many areas and purposes. For the volume of information, we can say that we shipped needed flows in the perfect amount of time.

The collaboration was great as we knew from the start the areas that each of us needs to complete at it's good at.

What could be improved

For the "what could be improved" part, we think that another great idea to give the user the folder that he needs, was to create a different page showing the images that he wants and also add animation there

CHAPTER

6

PROJECT CHARTER DOCUMENT

6.1 Project Overview

Project Name	bustedAI		
Project Charter Author	Tabacaru Andrei		
Project Manager	Hirica Ioan Alexandru		
Project Charter Status (Pending/Approve/Reject)	Approve	**Date of project approval**	November 29, 2022
Proposed Project Start Date	November 29, 2022	**Proposed Project End Date**	February 1, 2023

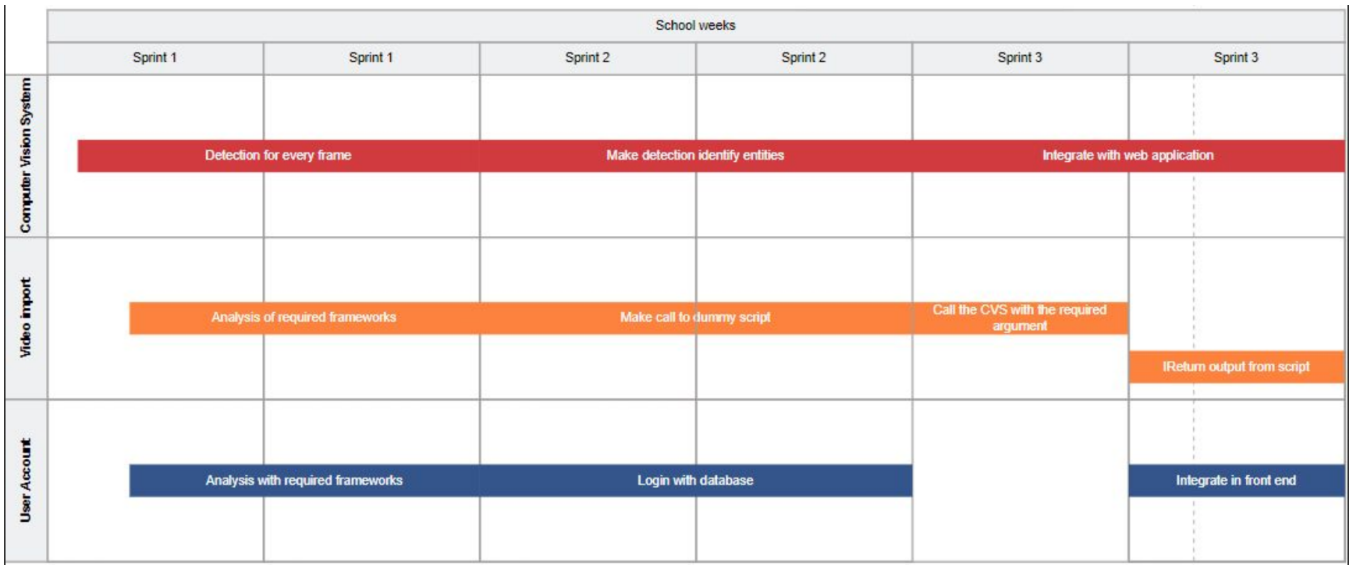
6.2 Project Details

<table>
<tr><td>Product Vision</td><td colspan="2">To bringing the best user experience to its customers through its useful software.</td></tr>
<tr><td>Mission Statement</td><td colspan="2">We’re in business to save your home while it’s alone.</td></tr>
<tr><td>Customer Expectations</td><td colspan="2">• Data protection and privacy
• Quick and easy to use solution
• Friendly interface</td></tr>
<tr><td>Succes Criteria</td><td colspan="2">• Meet the objectives
• Deliver the best possible quality
• Prioritize functionality over buggy quantity
• Achieve 100 user registrations within 2 months</td></tr>
<tr><td>Objectives</td><td colspan="2">Create an easy to use interface that allow users to collect images from their recordings</td></tr>
<tr><td rowspan="5">Team Members</td><td>Name</td><td>Role</td></tr>
<tr><td>Hirica Ioan Alexandru</td><td>Project Manager, Machine Learning Engineer</td></tr>
<tr><td>Grosu Ilinca</td><td>Full Stack Developer</td></tr>
<tr><td>Tabacaru Andrei</td><td>QA</td></tr>
<tr><td>Buzescu Alexandru</td><td>Backend Developer</td></tr>
<tr><td>Risks and Issues</td><td colspan="2">The team members are not experienced</td></tr>
<tr><td>Assumtions and Constraints</td><td colspan="2">The user might be tehnologically illiterate, so the interface has to be intuitive</td></tr>
</table>

CHAPTER

7

ROADMAP



CHAPTER

8

DEFINITION OF DONE

Definition of Done	
User Story Defined	✓
User Story Acceptance Criteria defined	✓
User Story dependencies identified	✓
User Story sized by Delivery Team	✓
Scrum Team accepts UE artefacts	✓
Performance criteria identified, where appropriate	✓
Person who will accept the User Story is identified	✓
Team has a good idea what will mean to Demo the User Story	✓

CHAPTER

9

DEFINITION OF READY

Definition of Ready	
Coding Done	✓
Testing Done	✓
No defects	✓
Acceptance testing	✓
Uploaded and live on production repository	✓
Backlog updated	✓
Task closed	✓

CHAPTER

10

SOFTWARE ARHITECTURE REPORT

This document is the first approach to present the information of this project in a structured fashion and discuss its architecture.

The structure of this document

1. A summarized description of the software architecture, including major components
2. Architectural constraints and decisions
3. A detailed description of each component
4. Skateholders

10.1 Architecture Summary

Here we will give a brief description about the application.

10.1.1 Overview

The goal of **bustedAI** is to give a simple, intuitive solution for users that value their time and want an easier process of finding out who is around their home while they are not there.

10.1.2 Basic Usage

- The User creates or log in their account
- The main page of the application has a button called "Choose File". The user presses the button then he has to upload a video. After the page loads successfully the application will automatically starts downloading the archive containing the desired images.
- The about page contains easy to swallow info about the used algorithms.
- The How to Use page explains how to use the application.

10.1.3 Sanity Checks

- Checks if the password has at least one capital letter, one lower case letter, one number, one symbol (e.g. !, @) and if it has at least 8 letters.
- Checks if the email address is a valid one.
- Checks if the email has already been used.

10.2 Constraints and decisions

10.2.1 About the architecture summary

Considering the fact that the majority of our users might have limited browsering knowledge we decided to have a very simple UI that is straight to the point. We also considered that a demo page might help the user, it also has a video showing exactly how they are expected to use our services.

Every component of the application has been constructed in such a way that adding new features will not require the developer to change what has been already done. Adding new features does not break the already existing functionality.

10.2.2 Concerning the used technologies

Here we will list the technologies used to build our application and the reason we chose them.

ASP.NET Core 6.0

ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps. With ASP.NET Core, you can: Build web apps and services, Internet of Things (IoT) apps, and mobile backends. Use your favorite development tools on Windows, macOS, and Linux.

Reasons for choosing ASP.NET Core are:

- Mature
- Cross-platform compatibility
- High performance
- Integration with modern client-side frameworks, such as Angular
- All team members took the optional course about this framework last year, so all of them have minimal understanding about it

Angular

Angular is a comprehensive JavaScript framework for building dynamic, client-side web applications. It uses a declarative approach to building user interfaces and leverages a modular architecture to make it easier to develop and maintain complex applications.

Reasons for choosing Angular are:

- Two-way data binding
- Modularity
- Reusable components
- Our only Frontend developer has some experience with this framework

You Only Look Once (YOLO)

YOLO (You Only Look Once)[1] is an object detection algorithm that uses a single neural network to detect objects in an image. Unlike previous object detection algorithms that apply a sliding window to detect objects, YOLO divides an image into a grid of cells and predicts the presence of an object and its bounding box coordinates within each cell. This allows YOLO to process an image as a whole and make predictions much faster. The algorithm also provides more accurate object detection than sliding window algorithms, especially for small objects.

Reasons for choosing YOLO are:

- Real-time detection:
- High accuracy
- End-to-end training

Simple online and realtime tracking (SORT)

An online and real-time tracking algorithm typically works by continuously updating the position and status of an object (e.g. a person, vehicle, etc.) in real-time based on data received from sensors such as GPS, cameras, or other devices. The algorithm processes this data in real-time, making decisions about the object's location and behavior, and updating its status in the system. The algorithm may also use predictive modeling and other techniques to anticipate future movement of the object and make predictions about its behavior. The goal of the algorithm is to provide accurate and up-to-date information about the object's position and status to be used for various purposes such as navigation, monitoring, and control.

Reasons for choosing SORT are:

- Enhanced visibility
- Increased efficiency
- Improved decision making

10.3 Components

10.3.1 Computer Vision System

Responsibilities

This component must recieve a path to a video saved with the .tmp format and it will serve to the user a zip file named "Output.zip" which contains all the different cars and persons present in the video.

Features

It goes through the video frame by frame and feeds the bounding boxes provided by the YOLO algorithm it feeds them to SORT algorithm which determines if it assigns the same id to the person or car, and then it crops the bounding box.

Setup

To setup the script, you need to run "pip install -r /Detection/requirements.txt" when being in the git base folder.

Usage

Run the script and it will create an Output folder containing the images, which will be transformed into a zip file.

10.3.2 Database

Responsibilities

Storing user information: The database stores information about the users of the system, such as their username, email and any other relevant information such as their name and authorization role.

Hashing passwords: The database should stores the passwords of the users in a secure manner, by hashing the passwords using a secure one-way hash function.

Features

Data storage: The primary feature of the database is its ability to store data in an organized and structured manner, allowing for efficient retrieval and manipulation of the data.The database provides efficient data indexing and searching capabilities, allowing users to quickly locate specific data records. It is designed to be scaled for meeting increasing demands for storage and processing power as the size of the data stored in the database grows.

Setup

Microsoft SQL Server could run on Windows and Linux. For easier database handling, we use Windows because Microsoft do not provide a Linux version of SSMS(SQL Server Management Studio).

Usage

The database is used to manage user authentication and authorization for the web application. The database stores the credentials of registered users, and is used to verify their identity when they log in. User authentication: When the user tries to log in, the system retrieves the entered username and password from the database and verifies it against the stored credentials. If the credentials match, the user is granted access. If the credentials match, the user is granted access.

User authorization: Based on the user's privileges and permissions, the system allows or denies access to certain parts of the site or application. The user's authorization level is stored in the database and is used to determine their access rights.

10.3.3 Web Server

Responsibilities

.NET Server is a server-side platform that is used to develop and run web applications, services, and APIs.

Request handling: The .NET Server is responsible for receiving HTTP requests from clients and processing them to generate the appropriate response.

Data processing: The server is also be responsible for querying the database for login/register API, but also for sending the video file for being analyzed. The server is managing its resources efficiently, such as memory, CPU (by writing asynchronous functions), and network resources, to ensure high performance and scalability. Dividing the code in 3 layers(the presentation layer, the bussiness logic layer and the data access layer) we assure the scalability of the project.

Features

.NET Server supports a wide range of operating systems, including Windows, Linux, and macOS, allowing for flexible deployment options. .NET Server provides features for scalability and performance optimization, such as multithreading and caching, to ensure the ability to handle high traffic and resource-intensive applications.

.NET Server provides a large number of libraries and tools for common tasks, such as database access, network communication, and data processing, making it easier to build and deploy applications.

.NET Server has a large and active community of developers who contribute to its development and provide support to users. This allows for a wealth of resources, such as tutorials, documentation, and code samples, to be available.

Setup

Installing the .NET 6 framework. Configuring the development environment, including setting up the database connections.

Usage

Data processing: The .NET server is used for querying the database for login/register APIs, but also for sending the video file for being analyzed.

10.3.4 Web Application

Responsibilities

Using Angular framework, the main responsabilities is linking the .NET API with a friendly user interface and ensuring the application is responsive and optimized for performance.

Features

With Angular, developers can create reusable components and modules, manage the application state with its two-way data binding, and leverage its powerful templating system to create user-friendly, highly interactive web pages.

Component-based architecture: Angular follows a component-based architecture, allowing for the modular and reusable building of web pages and applications. This helps to keep code organized and maintainable, making it easier to scale and update the application over time.

Improved performance: Angular uses lazy loading and ahead-of-time (AOT) compilation, which speeds up the initial load time of the application and improves its overall performance.

Setup

Installing Node.js, npm, Angular CLI and also the project's dependencies is needed.

Usage

By using Angular as the frontend framework and .NET as the backend, we developed a full-stack application that leverages the power of computer vision algorithm to provide a rich and interactive user experience.

10.4 Skateholders

Each stakeholder is concerned with different characteristics of the system. Here is a list of the stakeholder roles considered in the development of the architecture.

Name
Software developers
Description
They are the coders of the application
Responsibilities
Write code compliant with the requirements specified by the product owner and customers
Concerns
Security, performance, UI, programming language, database, workplace and workstations

Name
Software testers
Description
Software developers specialized in testing the application
Responsibilities
Find bugs, security holes and checking the functionality against the requirements
Concerns
Security, platforms, architecture, database

Name
Customers
Description
People interested in buying something or just browsing
Responsibilities
Behave appropriately and be nice. Spend money.

CHAPTER

11

TESTING

Unit Testing

- The Database, the Web Server and the Web Application have been tested manually.
- For the Computer Vision System we used automated testing and the results can be found at Figure 11.1

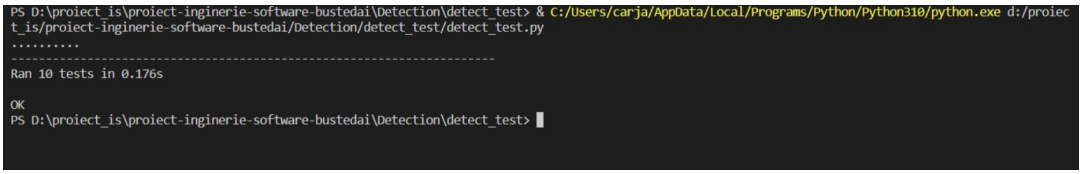


Figure 11.1: Computer Vision System Test Result

Scalability Testing
The application does not have a maximum number of accounts registered.

Response Time
The authentication successfully loads in less than 3 seconds.
Each page loads in less than 2 seconds.

User Acceptance Testing (UAT)	
Has the project team been made aware of its role in advising on changes to business processes and procedures?	✓
Has the project team been made aware of its role in providing support for all testing issues?	✓
Has the project team been made aware of its role in tracking and managing website bugs?	✓

UAT Team Preparations	
Has the UAT team been defined?	✓
Does the UAT team understand its responsibility in executing the test cases and ensuring that the final outcomes of the tests are satisfactory?	✓
Has the UAT team been told about its role and responsibility in ensuring that all test case input sources and output results are documented?	✓
Has the UAT team agreed that the test cases provide comprehensive and effective coverage of all aspects of functionality of the application?	✓
Has the UAT team been told about its role in documenting bugs/problems and working with the project team to resolve problems identified during testing?	✓
Does the UAT team understand the responsibilities and required actions for each category of problem identified during testing?	✓
Has the UAT team been made aware of its role in accepting the results on behalf of the relevant user population?	✓
Does the UAT team understand that it must recognize any changes necessary to existing processes and take a lead role in ensuring that the changes are made and communicated to other users?	✓
Does the UAT team understand its role in verifying performance on business critical functions?	✓
Does the UAT team understand its role in confirming the integrity of data?	✓
Does the UAT team understand its role in assessing system final production readiness?	✓

Test Preparation	
Has the plan for acceptance testing been created?	✓
Have all possible system functions been described?	✓
Is all input data available that is required for testing?	✓
Has acceptance criteria been defined on which the completion of the acceptance test will be judged?	✓
Have all user specific constraints been considered?	✓
Has the testing procedure been defined?	✓
Have test cases been created to discover contradictions between the software product and the requirements?	✓
Have test cases been created to review whether timing constraints are met by the system?	✓

Test Execution and Evaluation	
Were all steps of the test run documented?	✓
Was the acceptance test performed according to the test plan?	✓
Did the users review the test results?	✓
Are the services provided by the system in compliance with user requirements?	✓
Were all identified defects and issues resolved?	✓
Did the users judge acceptability in accordance with the predetermined criteria?	✓
Did each user sign off on output?	✓

BIBLIOGRAPHY

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.