

# **RUP Artifacts**

**for**

***Trajectory Detection Web Services Project***

([Repository](#))

**Prepared by Denis Kalachev, Ehsan Shaghaei**

**25.09.2021**

## Version History

Editor's Name	Date (DD/MM/YYYY)	Reason for Changes/Sections Updated	Version
Denis Kalachev	25.08.2021	Creation, adding user stories and dev plan	0.1
Denis Kalachev	01.09.2021	Adding title, version history, roles	0.2
Ehsan Shaghaei	05.09.2021	Adding business goals, rewriting features, iteration planning	1
Denis Kalachev, Ehsan Shaghaei	23.09.2021	Updating features, development plan	1.1



# Table of Contents

<b>Business Goals and Objectives</b>	<b>4</b>
<b>Roles and responsibilities</b>	<b>4</b>
<b>Requirement Analysis and Specifications</b>	<b>5</b>
Features	5
User Stories	5
<b>Software Development plan</b>	<b>6</b>

## 1. Business Goals and Objectives

No	Goals
1	Providing lightweight web-services for autonomous vehicles clients
2	Providing lightweight web-services for IoT clients
3	Creating an evolving environment for computer vision developers with trajectory estimation perspective
4	Data collection for improvement of models
5	Data collection for data banks

## 2. Roles and responsibilities

Stakeholder's Name	Roles	Responsibilities
Ehsan Shaghaei	Project Manager, Computer Vision Engineer, Developer , QA	Designing the backlog and creating iteration tasks, Verifying and developing computer vision models, Developing Backend
Denis Kalachev	Documentation manager, Full-Stack Developer, QA, DevOps	Managing documentation, Development and maintaining API, Deployments
Unmanned Technology Lab , Innopolis University	User	Evaluated the software development

### 3. Requirement Analysis and Specifications

#### 3.1. Features

ID #	Features	Priority
1	Registration / Login / Logout	Medium
2	Reports Sessions	Medium
3	Trajectory Estimation Session	Must
4	REST API	Must
5	Server backup	Low
6	Server recovery	Low

### 3.2. User Stories

User Type	Features	User stories
Web User	Registration / Login / Logout	User <u>registers</u> through the API providing his <i>email</i> and <i>password</i> and receives a VERIFICATION EMAIL.
		User <u>logs in</u> to the portal through the API using his verified <i>credentials(email, and password)</i> and receives a session TOKEN.
		User <u>logs out</u> through the provided API and provides his/her session TOKEN.
	Reports Sessions	A logged in user <u>gets</u> the reports of previous sessions using the provided API; providing his/her session TOKEN.
	Trajectory Estimation Session	Logged in user <u>uploads</u> an .mp4 video to the server through the provided API; providing his/her session TOKEN.
	REST API	The web service <u>has</u> a well-defined REST API
Server	Database recovery	In case of server <u>failure/relocation</u> , respectively failed/new server <u>pulls</u> the newest version of the software from the repository and <u>deploys</u> it. Server <u>runs</u> server initialization script and <u>downloads</u> and <u>extracts</u> the database back up.
	Database backup	After every 10 requests, server <u>creates</u> a new backup from database and <u>uploads</u> it to <i>Google Drive</i> and <i>files.fm</i>

## 4. Non-functional requirements

NFR	Sub-Characteristics	How will you achieve it
Usability	Simplicity	Service <u>provides</u> a well defined REST API for the well defined user use case scenarios.
	Accessibility	We want this service to have 99% uptime and be accessible from any device. We will achieve that using Amazon Cloud Web Service.
	Stability	After every 10 requests, server creates a backup and uploads it to <i>Google drive</i> and <i>file.fm</i> . In case of server <u>failure/relocation</u> , respectively failed/new server <u>pulls</u> the newest version of the software from the repository and <u>deploys</u> it. Server runs server initialization script and <u>downloads</u> and <u>extracts</u> the database back up.
Security	Authentication	Service <u>provides</u> the sign up feature with email verification requirement for user, <u>provides</u> a session TOKEN for user login, provides logout
	Session safety	Service will log off the user automatically after 10 minutes after last activity. Service supports HTTPS protocol for safer communication.
	Login Tracking	Service tracks the user login activities



## 5. Technical stack

Technical stack	Tools
Task management	Github task project management tools
Web service backend	Python Flask
Database	PostgreSQL, SQL Alchemy
Detection models	OpenCV, Yolo
Continuous integration	Travis CI
Continuous deployment	Heroku

## 6. Software Development plan

Inception Phase				
#Iteration	Timeline	Stakeholders	Activities	Artifacts
#1	24/08/2021 - 26/08/2021	Denis Kalachev	Determine project objectives with valid justification Identify the stakeholders Establish roles and responsibilities	Deliver the documentation of achieved milestones
#2	27/08/2021 -29/08/2021	Ehsan Shaghaei, Denis Kalachev	Determine proposal for software development plan.	Start Implementation
#3	30/08/2021 - 01/09/2021	Ehsan Shaghaei, Denis Kalachev	Requirement engineering(60% user stories) Identify plan	Update the documentation of achieved milestones with User stories and very basic work plan

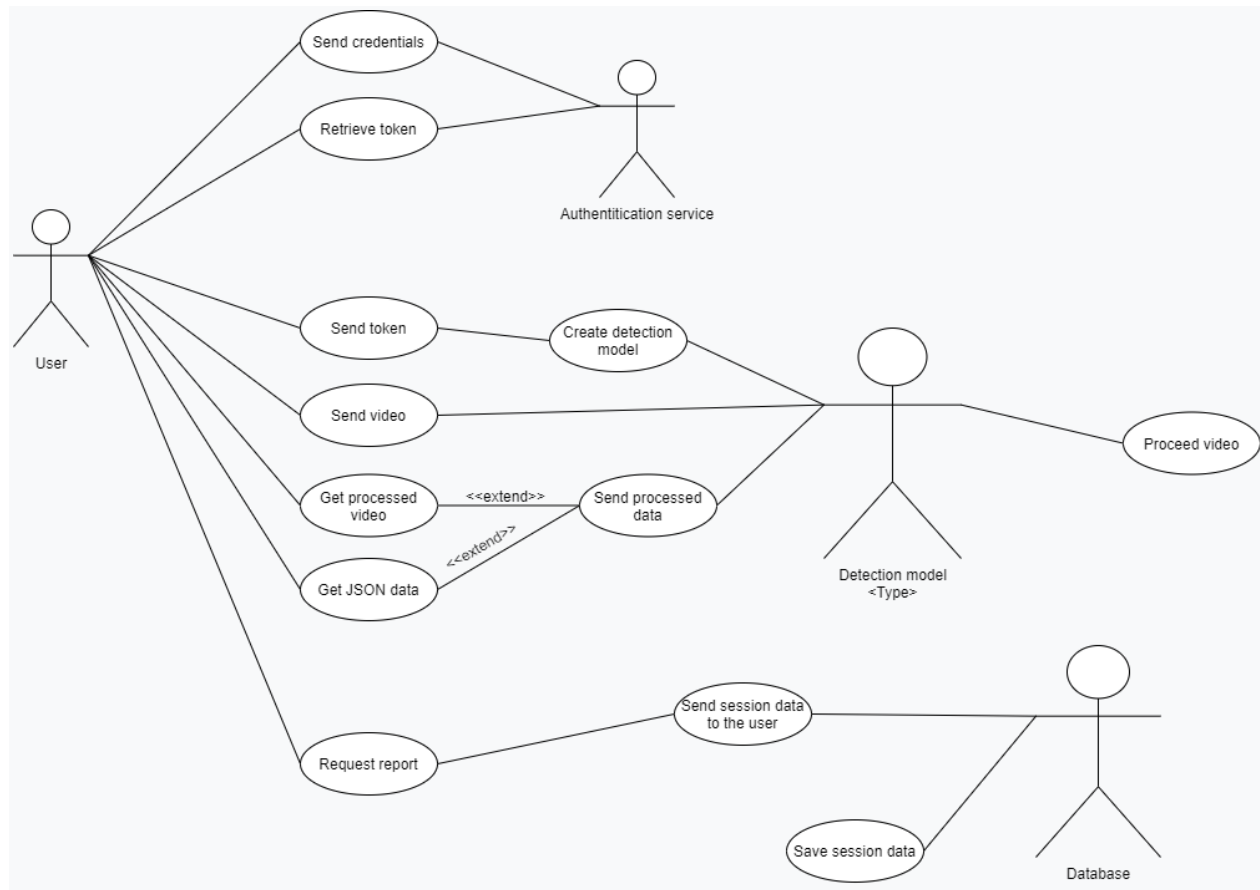
Elaboration Phase				
#Iteration	Timeline	Stakeholders	Activities	Artifacts
#1	29/08/2021-30/08/2021	Denis Kalachev, Ehsan Shaghaei	Revise User Stories (100%)	Document 100% user stories
#2	29/08/2021-30/08/2021	Denis Kalachev, Ehsan Shaghaei	Software development planning	Iteration Plan
#3	30/08/2021-02/08/2021	Ehsan Shaghaei	Create project structure	Project structure
#4	30/08/2021-31/08/2021	Denis Kalachev, Ehsan Shaghaei	Software Architecture Test Plan	Software architecture document Test Plan Document
#5	20/09/2021-22/09/2021	Denis Kalachev, Ehsan Shaghaei	Updating requirements	Glossary and technical stack

Construction Phase				
#Iteration	Timeline	Stakeholders	Activities	Artifacts
#1	15/09/2021-30/09/2021	Denis Kalachev	Implement Features 1, 4 - REST API	Working REST API/ Register/Login/Logout
#2	20/09/2021-01/10/2021	Denis Kalachev, Ehsan Shaghaei	Implement Features 6, 7 - Server database, backup and recovery scripts, continuous integration	Stable database, sustainable server
#3	26/08/2021-31/08/2021	Ehsan Shaghaei	Implement Feature 3 - Basic Object Detection model	Working for tennis ball detection
#4	31/08/2021-05/09/2021	Ehsan Shaghaei	Implement Feature 3 - Basic Trajectory Detection model	Working for tennis ball trajectory estimation
#5	31/08/2021-24/09/2021	Denis Kalachev, Ehsan Shaghaei	Implement Feature 2 - Reports Sessions	Retrievable report sessions

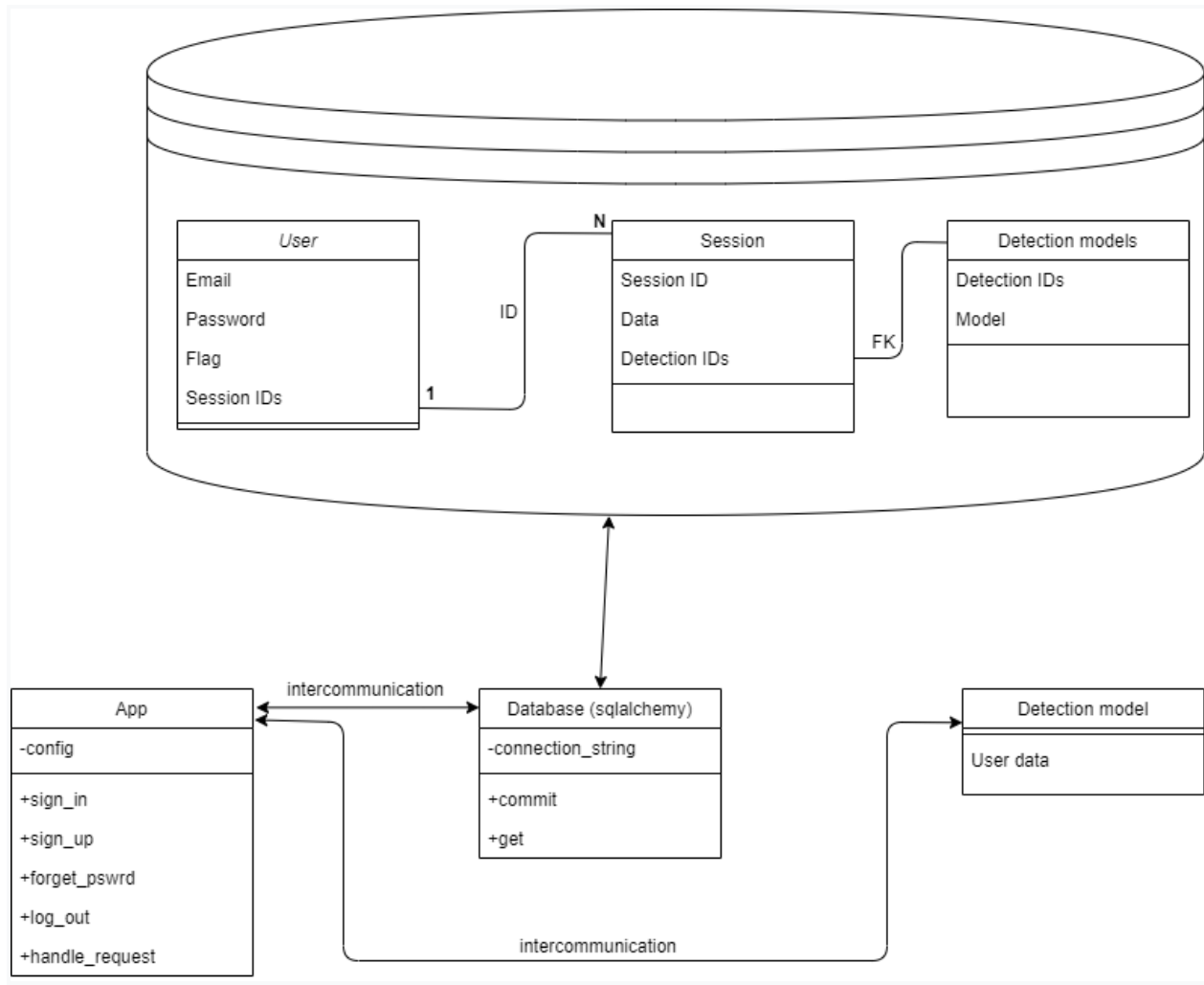
Transition Phase				
#Iteration	Timeline	Stakeholders	Activities	Artifacts
#1	01/10/2021-12/10/2021	Denis Kalachev, Ehsan Shaghaei	Integration, Unit test,QA,User Validation	Github repository Merged branches Integration and ended to end test results Final README for developers and Users
#2	13/10/2021	Denis Kalachev, Ehsan Shaghaei	Final web service release	Working Product

## 7. Design documentation

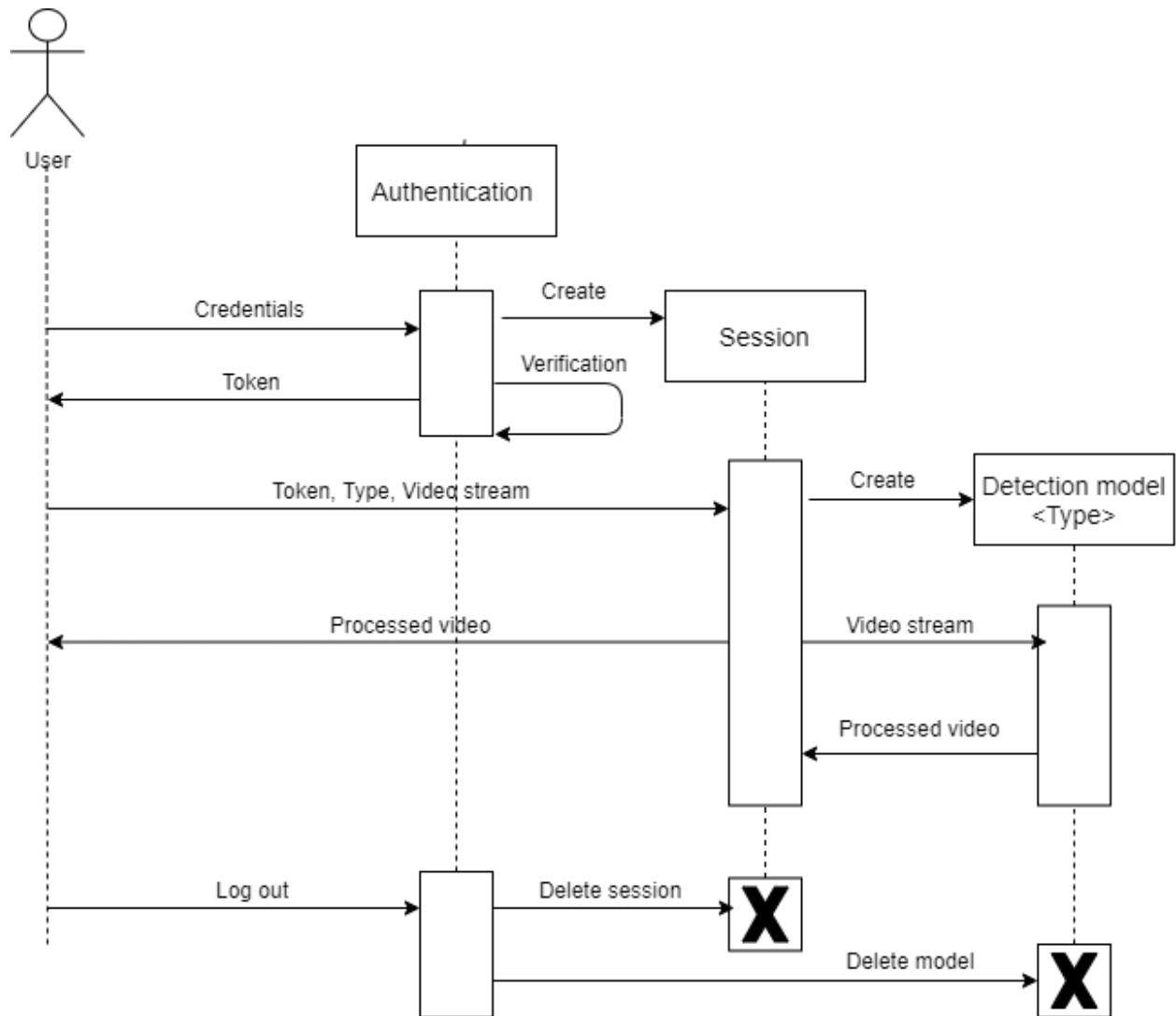
### Use cases diagram



## Class/DB diagram



## Sequence diagram



## **8. Glossary**

Token - 256 bytes unique string for session id, which is responsible for authentication and identification of a user session, which can also be used for retrieving reports or responding to user requests

Detection model - class of machine learning models with a well defined interface for detecting user desired class of entities

Trajectory detection - a sort of detection models which estimates trajectory of the user desired class of entities