Software Requirements Specification

for

<Fake Attendance Signature Detection Application>

Prepared by <- Alperen Kayhan, 2388532 - Ali Efe Sarıgöz, 2385664

- Bora Bedirhan Uyar, 2526788 - Halil Burak Özdemir, 2316107>

<Middle East Technical University NCC>

<Computer Engineeringg>

Supervised by <Meryem Erbilek>

<28.10.2025>

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to define the requirements and specifications for the Fake Attendance Signature Detection and Analysis (FASDA) system. The main purpose of FASDA is to identify forged signatures in student attendance sheets by scanning, processing, and verifying handwritten signatures using machine learning and computer vision techniques. Our Motivation, by automating verification, FASDA reduces administrative workload and attendance validation. SRS sheet includes detailed requirements for software design, dataset, ethical points, and user interaction.

1.2 Scope

The Fake Attendance Signature Detection and Analysis (FASDA) system is a standalone web application built to automate the detection of fake and forgery signatures on student attendance sheets. It combines computer-vision preprocessing, deep-learning verification, and a graphical user interface to deliver an end-to-end solution to academic staff.

FASDA is designed to scan attendance sheets, identify and extract individual regions of signatures, and compare each extracted signatures to system stored reference signatures. A Siamese convolutional-neural-network (CNN) was used to classify each signature as legitimate or a forgery. It was trained on public offline signature datasets which include GPDS and CEDAR Dataset and was augmented with a locally curated METU NCC dataset collected under BAYEK ethical approval.

Goals:

- *System/application that uploads scanned attendance forms which later; automatically generates a report showing whether they are forgety signuters or not .*
- *Achieve robust detection of forged signatures across varying scan qualities and writing conditions.*
- *Offer an intuitive interface usable by academic staff.*
- *According to the ethical approvel of the BAYEK, our project will establish a small but representative signature dataset*

Objectives:

- *Collect genuine and simulated forged signatures from participants under BAYEK approval to build a training/validation dataset.*
- *Design and implement ML/DL pipelines for offline signature verification (pre-processing, feature extraction, classification).*
- *Build a web-based interface for upload, batch processing, and report generation.*
- *Set up a secure system to store signatures, models, and results, making sure only authorized people can access them and that all actions are tracked for accountability.*

1.3 Related Work

This section examines the key studies and applications concerning offline handwritten signature verification. These studies have directed the FASDA (Fake Attendance Signature Detection and Analysis) design particularly in the choice of Siamese CNNs, OpenCV for preprocessing, and the incorporation of ethical frameworks for the use of biometric data within a research study. Yatbaz and Erbilek (2020) demonstrated that by utilizing deep learning tools, particularly convolutional neural networks (CNN), variations in offline signatures associated with stress could be classified. The distinguishing factors caused by stress were geometric distortion and stroke pressure variations, which can facilitate distinguishing between a series of vulnerabilities that provide minor behavioral cues associated with irregular or forged writing. In FASDA, this finding shapes behavior accuracy feature as part of the preprocessing and feature-engineering component, as it is expected that in instances of behavioral inconsistency that geometric and texture signature similarities will be captured by the CNN in terms of forgery detection of attendance signatures. Bay Ayzeren, Erbilek, and Çelebi (2019) progressed this interpretation by articulating emotional states to the differences in handwriting dynamics often not discussed in online biometric systems. Even if FASDA were studying offline scanned signatures, we still can connect their comment on temporal and spatial variability similar to our verbal concern with intra-writer variability. FASDA uses the same tolerances and "irregular signature" features to identify Bay Ayzeren, Erbilek, and Çelebi's (2019) finding that we would not fit to penalize changes related to emotion with the same degree of rigor when assessing forgery. Hafemann et al. (2017) presented the architecture of a Siamese CNN structure to frame signature assessments based upon pairwise comparison. The permissible comparison of a dual-branch CNN-based signature verification model produces an embedding that reduces the similarity of distance based upon repeated interpretations of intra-writer samples, and increases inter-writer distance, which is a fundamental design for current verification systems. FASDA utilizes this dual-branch architecture to observe the similarity of new attendance signature produced by the operation against the adopted enrolled reference signature, with consistent comparisons with little visual differentiation being observable. Dey, et al. (2020) proposed a hybrid CNN-SVM pipeline that merges deep feature extraction from CNNs with SVM classifiers that refine decision boundaries. Building on that hybridization, FASDA could follow-up with lightweight SVM modules for edge-deployed validation in cases where deep inference is too expensive, such as in low-power institutional scanner. Hafemann's later work with the SigNet model that provided an open source benchmark strengthened its value by being trained on the GPDS dataset to produce consistent embeddings to represent signatures patterns. This will be FASDA's preliminary pre-training model for our own new Siamese network to fine-tune on our attendance dataset instead of training from scratch. Kumar, et al. (2020) advanced this direction by optimizing a Siamese CNN with contrastive loss for threshold based decision making on scanned signatures. FASDA can utilize a similar threshold tuning to distinguish levels of verification confidence and map signature similarity scores into "genuine," "forged," or "irregular" categories on report results. At last, DocuSign Inc. has developed a commercial AI verification platform that determines the authenticity of documents by examining the ink distribution and the continuity of hand strokes made to sign the document. While the system is proprietary, it illustrates the ways that ethical, secure, and scalable AI verification systems can function in collaborative and professional spaces. The FASDA translates these categories into an educational context by embracing ethical deployment (BAYEK approval), user transparency, and the ability to store reports in the cloud. Together, these works demonstrate both theoretical and technical foundations for the architecture of FASDA. The data use of Siamese CNNs for verification, hybrid models to maximize training efficiency, and ethical AI deployment within educational institutions. Make the our sources to be great working steps on our FASDA project.

1.4 Product Overview

1.4.1 Product perspective

As shown in Figure 1, the system will depend on four external systems. The first is the **External Bank System**, responsible for handling the payment process. The second is the **SheerID App**, which provides tokens verifying that users are genuine university instructors. Lastly, the **Cloud** serves as a platform where user, student, and dataset information is stored and retrieved.

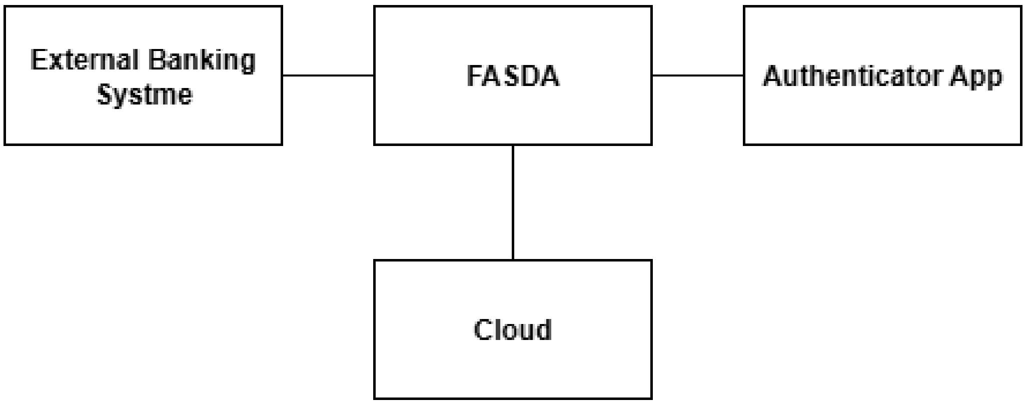


Figure 1. FASDA Product Perspective

1.4.1.1 System Interfaces

List each system interface to interact with external systems and explain their functionalities.

Cloud – the system that we pulled and store the user attributes and the signutre data set.
Bank – The banking system allowed us to perform to pay monthly subscription. With that transaction user may use the FASDA app. Thus, Our system will send the details and the bank will perform the payment.
Authenticator App – it checks out the instructor information and verifies the existence of the instructor.

1.4.1.2 User interfaces

The FASDA Web Application is an organized, responsive, and role-based interface for technical and non-technical users alike. All pages share a similar layout with a central content panel, an upper navigation bar, and a role-related side navigation menu. The system promotes readability, speed of upload, and a minimal number of interaction steps.

Main Interface Components:

1. ***Login and Registration Page***

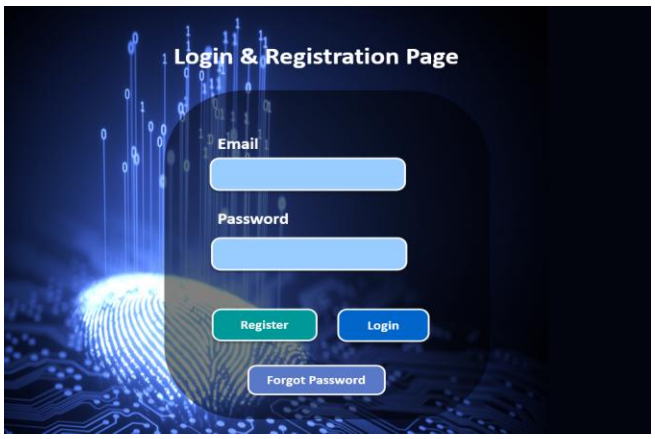


Figure 2. 1.Login and Registration Page Representation

Purpose: Allows instructors and admins a way to access the site safely.

- This page contains an email / password field and a "Login" button; for those not registered, they will be offered a "Register" option.
- A successful registration will involve verifying the entering university email with SheerID in order to continue onto the payment page.
- When necessary, the page will provide discreet feedback if the user entered invalid credentials, the user has not yet completed a verification step, or the user's subscription has expired.

(Example layout - two-column card: left side illustration - right side login.)

2. *Dashboard (Home)*

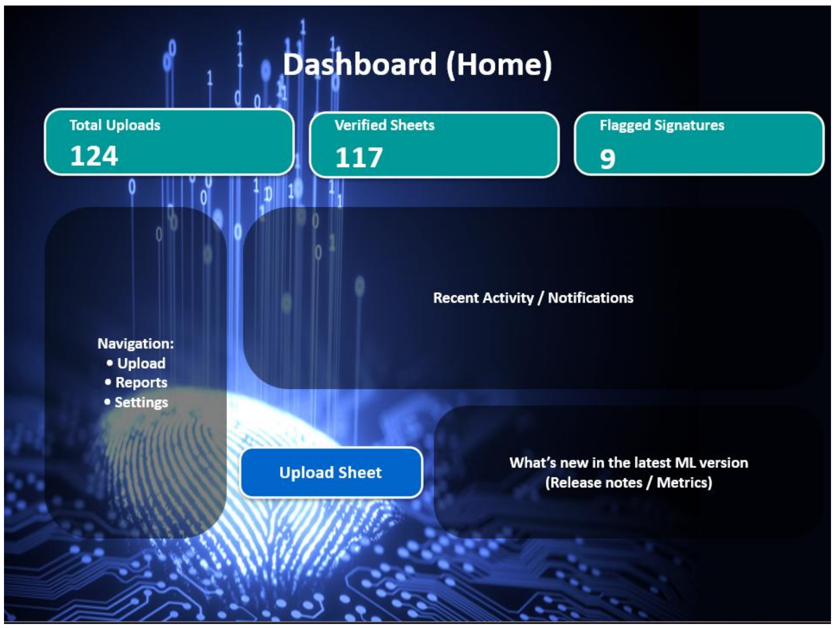


Figure 3. Dashboard Page Representation

Purpose: The default overview the user sees when logging into the site; it shows the status of the system and provides links to each dashboard without have to navigate back to the home page.

- Top summary card: Shows number of uploads, number of verified sheets, and how many triggered flagged signatures.
- There is a navigation panel located on the side: the instructor should be able to jump to Upload, Reports, and Settings from the dashboard.
- Dynamic Notification Area: Review a subscription status, as well as what is new in the latest version of ML.

(Example layout: top row stats + center quick action buttons.)

3. *Upload and Pre-processing Screen*

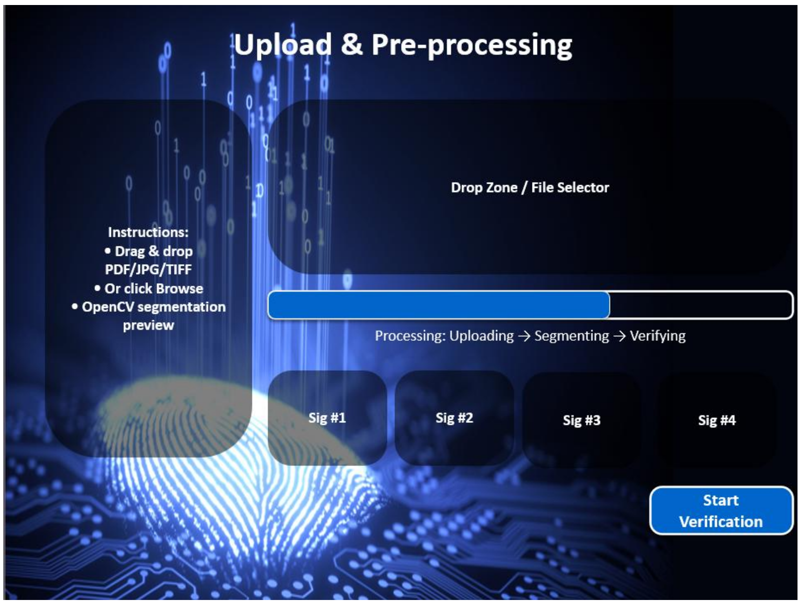

Upload & Pre-processing
Instructions:
• Drag & drop PDF/JPG/TIFF
• Or click Browse
• OpenCV segmentation preview
Drop Zone / File Selector
Processing: Uploading → Segmenting → Verifying
Sig #1
Sig #2
Sig #3
Sig #4
Start Verification

Figure 4. Upload and Pre-processing Page

Purpose: To allow the instructor to upload the attestation sheet for analysis, and demonstrate how it may be scanned.

- Provides a drag and drop area, or "Browse" button, for uploading a PDF/JPEG/TIFF.
- Provides a progress bar, and indicates progress for scanning, segmenting, and confirming on the screen as it is being processed in real time.
- Provides thumbnails of extracted signatures, using OpenCV segmentation to show us the signatures as we preview them.

(Example layout: file input panel + progress meter + preview area.)

4. *Verification Results / Report View*

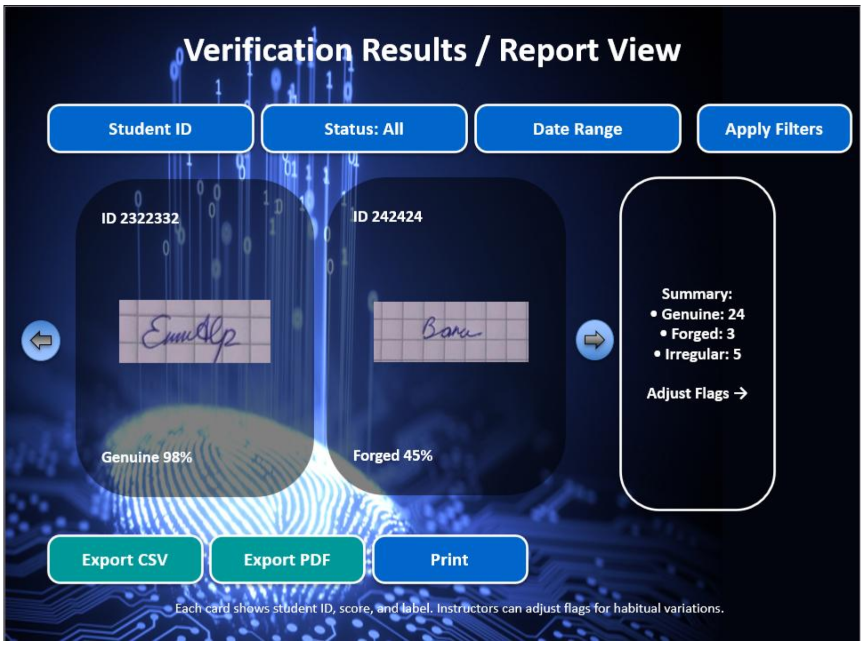


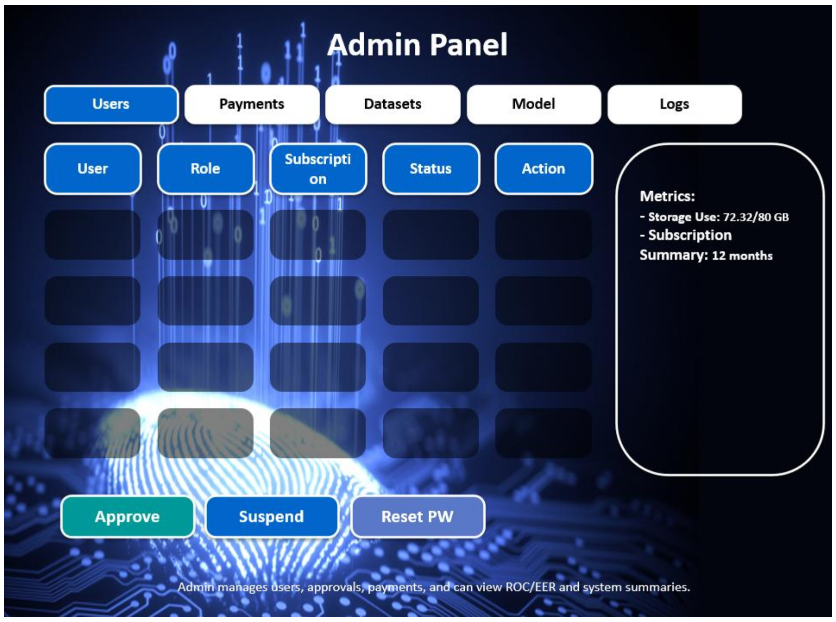
Figure 5. Verification Results / Report Page

Purpose: This page shows the results of signature authenticity verification and allows instructors to comment.

- Each signature card shows the student's ID, scoring, and a designation as either Genuine, Forged, or Irregular.
- The instructor can interact with buttons to mark a flagged signature for routine writers (students who continuously change signatures).
- Can inform the instructor and give options to save a report in a CSV or PDF formats or just use the print-output button.

(Example layout: area with signature cards in grid + summary panel on the right + export toolbar.)

5. *Admin Panel*



Metrics:
- Storage Use: 72.32/80 GB
- Subscription
Summary: 12 months

Admin manages users, approvals, payments, and can view ROC/EER and system summaries.

Figure 6. Admin Page

Purpose: Admin panel is a workspace for managing user attributes, data sets, and adjusting the system functions.

- Admin view can manage users, approve registrations, verify payments, and suspend/reactivate accounts.
- Admin view manages a user's roles and access, retraining the ML model, and log information.
- Display visual graphs for ROC/EER metrics, storage utilization, and subscription summary.

(Example layout: Tabbed UI with different views. User Mgmt / Datasets / Model / Logs.)

1.4.2 Product functions

- *Attendance Sheet Upload: User upload scanned attendance sheets for verification*
- *Image Processing: After the uploading of the image, backend perfomrs some actions like; binarization, noise reduction and signature segmentation.*
- *Feature Extraction: System converts the processed signatures into vector embeddings which using CNN.*
- *Signature Verification: Siamese CNN comprares the vectorified processed signatures with stored references from Data set and calculates similarity scores.*
- *Result Visualization: The web interface displays verification outcomes with color-coded labels and confidence percentages.*
- *Dataset Management: Admins can update, retrain, or manage dataset entries.*
- *Audit Logging: All actions (uploads, verifications, exports) are logged for traceability and ethical compliance.*
- *User Authentication: Secure login via email/password with session tokens.*

1.4.3 User characteristics

The FASDA system is focused primarily on three groups of users: the instructors, administrators or researchers, and the developers or maintainers.

- ***Instructors** are the primary end users of the system. They will use the system's web interface to upload scanned attendance sheets, start the verification process, and examine the signature verification results. They will be expected to have some basic computer skills that will allow them to use web applications and interpret the reports generated by the system.*
- ***The administrators** and researchers have higher privileges since they can manage and edit the system's datasets, oversee verification logs, adjust system thresholds, and trigger retraining of the deep learning model. They will need to possess some intermediate level of technical expertise in organizing datasets and verification metrics such as accuracy and graphs based on ROC, EER, and other verification metrics.*
- ***The developers** and maintainers of the system need to extend the system's functionalities, update machine learning models, and maintain the system's stable operation, both frontend and backend. They will need to possess high level skills in Python and Java, other web technologies, databases, and machine learning using tools like TensorFlow and OpenCV.*

1.4.4 Limitations

***Estimating the Size of the Database and Ethical Concerns:** The database of genuine and forged signatures collected from the students is small and will be used for both training and testing the model. This limited size and lower level of diversity will affect the model's ability to test generalization for the population of signatures and different styles of handwriting.*

An ethics application to METU NCC's ethics board (BAYEK) is mandatory for the proposed data collection. This may place some limitations on the method of data collection and sample size for data collection and their acceptable use in the study.

***Conditions for Offline Dependence and Scanned Image Quality:** The attendance sheets which are scanned and issued to the instructors and academic staff. The scanned image quality will very much determine the detection accuracy which could be poor owing to resolution, lighting, shadows, and even wrinkles, smudging, and paper folds.*

FASDA is an offline signature detection application. It uses image processing techniques and cannot analyze dynamic biometric data such as pen pressure, writing speed, and stroke order which are crucial for advanced forgery detection.

Detection Accuracy (False Anticipation and Negation): *The machine is designed to operate using elements of machine learning. So, it is logical that 100% accuracy is not guaranteed. The system will have to deal with the possibility of False Anticipation (a genuine signature flagged as forged) and False Negation (a forged signature undetected as genuine). This risk is always present.*

1.4.5 Assumptions and dependencies

- *It is assumed that the necessary ethics board (BAYEK) approval will be obtained quickly, and a sufficient number of genuine and forged signatures will be successfully collected from students to build the signature database for model testing.*
- *It is assumed that the project team members will have continuous access to mandatory hardware, such as a scanner and a computer/laptop, throughout the design and implementation phase.*
- *It is assumed that the project team will be able to access Database platforms and ML platforms anytime with stable connection.*
- *It is assumed that for accuracy and reliability, instructors are expected to use a scanner capable of providing consistently high-quality and uniform images.*
- *It is assumed that the scanned attendance forms uploaded to the system will conform to a predefined, acceptable resolution and file format (PDF) to ensure the effective operation of the image processing algorithms.*
- *It is assumed that the students' biometric attributes derived from the authentic signatures should remain consistent throughout the project without losing their core identification features. We also presume that no students will perform drastic alterations to their signatures over the course of the project.*
- *It is assumed that the quality of the forged signatures collected will be at a reasonable skill level, depicting plausible attendance fraud attempts to accurately prepare the machine learning model for real-life situations.*

1.5 Definitions

Term	Definition	Source
Siamese CNN	This refers to a deep learning model that consists of two identical neural networks that learn how to compare two signature images to measure similarity. It is used to differentiate genuine signature from forgeries by observing specific pixel features of each signature.	Project Definition / Hafemann et al. (2017)
OpenCV	An open-source computer vision library that is implemented to preprocess scanned attendance sheets which includes binarization, isolated the contour, and signature region segmentation.	OpenCV Official Documentation
TensorFlow	A deep learning framework that was selected to develop, train and deploy the FASDA signature verification model.	TensorFlow Documentation
PostgreSQL	A relational database that is used to securely store user information, signatures and verification findings.	PostgreSQL Documentation
BAYEK	METU NCC Ethics Committee, which assigns ethical approval codes to biometric studies related to the researcher's data protection policy to provide confidence.	METU NCC Ethical Research Guidelines
Signature Embedding	A numerical feature vector representation of the signature provided by the Siamese CNN for the purposes of comparing two signatures.	Project Definition

2 Specific requirements

2.1 External interfaces

Use a tabular format to define the inputs and outputs of the system interfaces, which should be consistent with Section 1.4.1. See an example below:

System Interface	Functionality	Input	Output
1.Bank API	Handles the payment for registration through secure transaction requests.	Payment details sent from the application	Confirmation and payment receipt returned to system
2.SheerID	Validates the user's academic identity during registration	Institutional credentials and documents	Verification confirmation sent to the system for approval

2.2 Functions

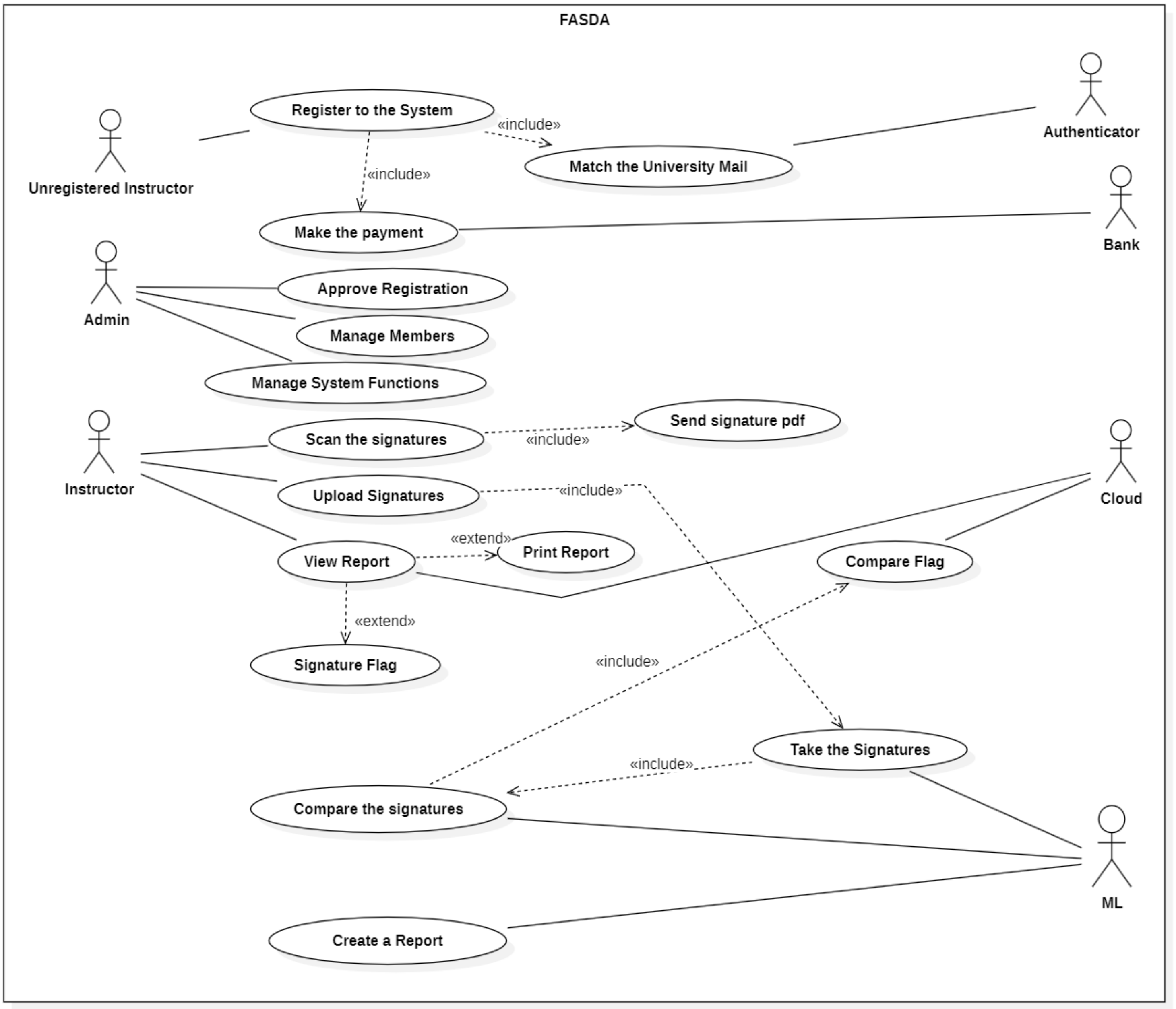
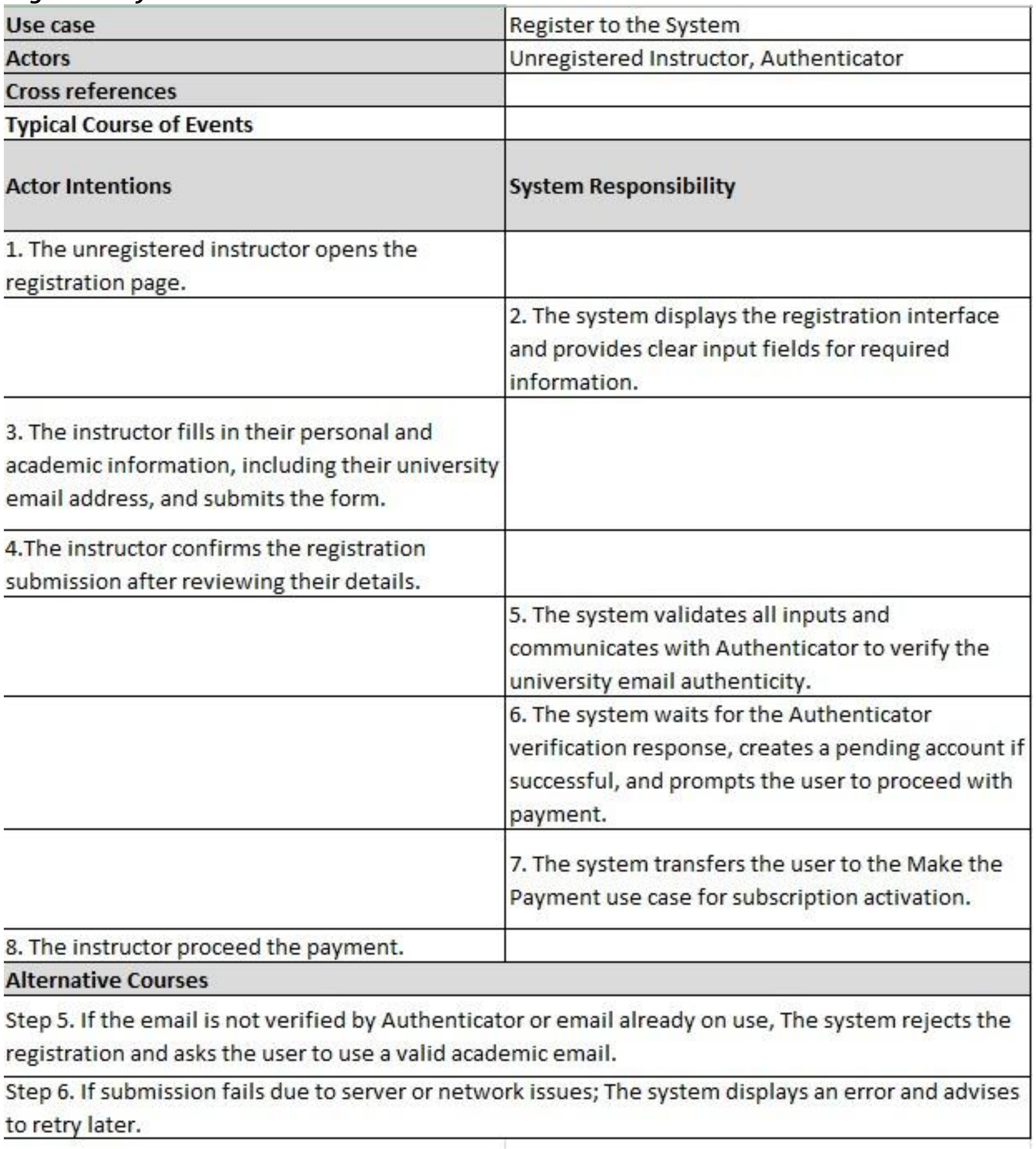


Figure 7. FASDA Use Case Diagram

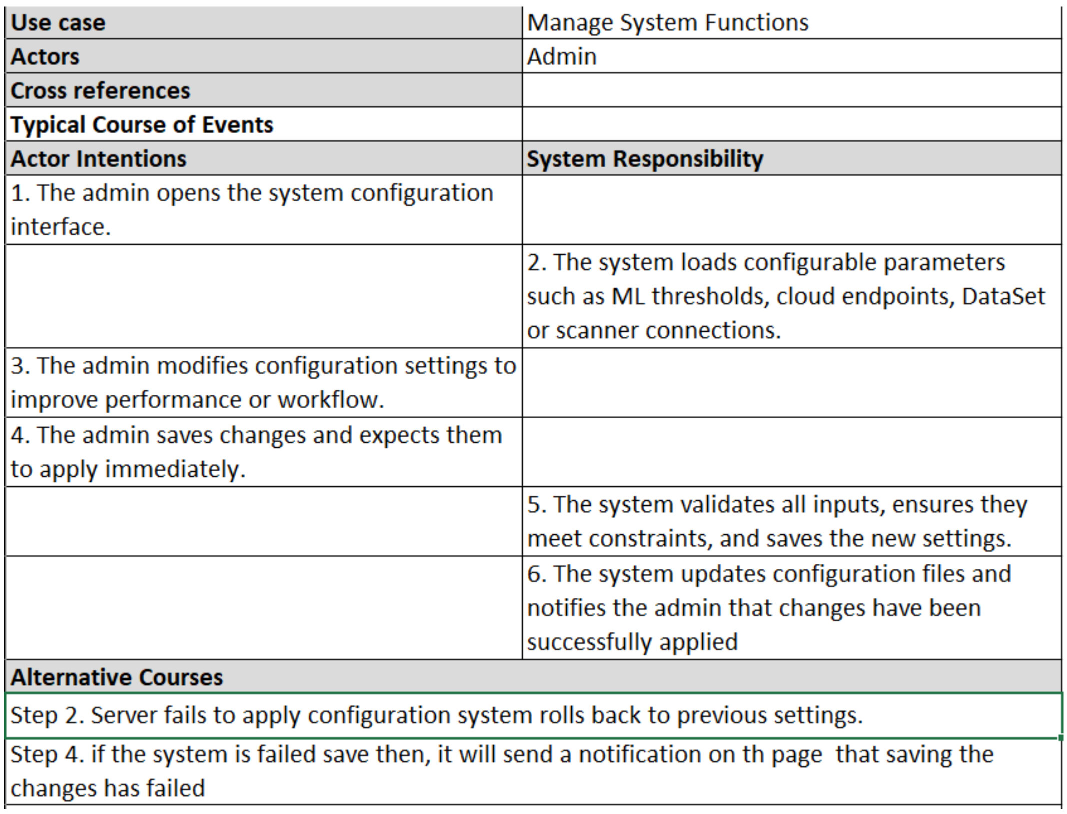
Register to System:



Use case	Register to the System
Actors	Unregistered Instructor, Authenticator
Cross references	
Typical Course of Events	
Actor Intentions	**System Responsibility**
1. The unregistered instructor opens the registration page.	
	2. The system displays the registration interface and provides clear input fields for required information.
3. The instructor fills in their personal and academic information, including their university email address, and submits the form.	
4.The instructor confirms the registration submission after reviewing their details.	
	5. The system validates all inputs and communicates with Authenticator to verify the university email authenticity.
	6. The system waits for the Authenticator verification response, creates a pending account if successful, and prompts the user to proceed with payment.
	7. The system transfers the user to the Make the Payment use case for subscription activation.
8. The instructor proceed the payment.	
Alternative Courses	
Step 5. If the email is not verified by Authenticator or email already on use, The system rejects the registration and asks the user to use a valid academic email.	
Step 6. If submission fails due to server or network issues; The system displays an error and advises to retry later.	

Figure 8. Register to the System UseCase Function Description

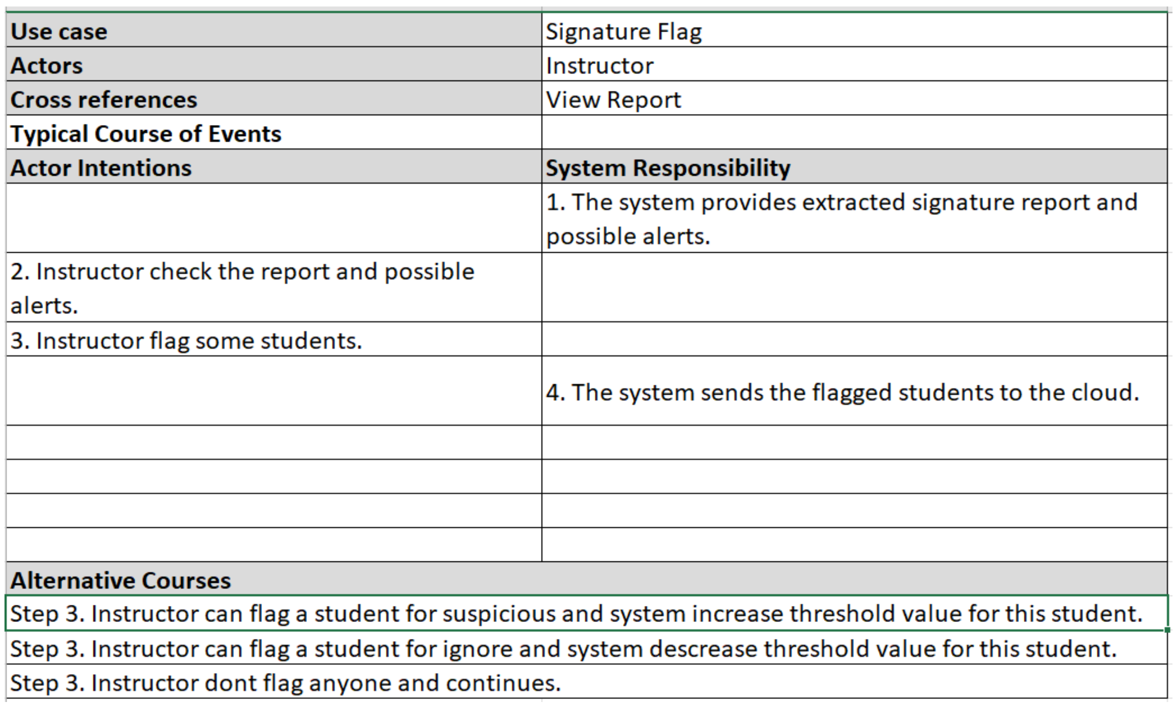
Manage system Functions:



Use case	Manage System Functions
Actors	Admin
Cross references	
Typical Course of Events	
Actor Intentions	**System Responsibility**
1. The admin opens the system configuration interface.	
	2. The system loads configurable parameters such as ML thresholds, cloud endpoints, DataSet or scanner connections.
3. The admin modifies configuration settings to improve performance or workflow.	
4. The admin saves changes and expects them to apply immediately.	
	5. The system validates all inputs, ensures they meet constraints, and saves the new settings.
	6. The system updates configuration files and notifies the admin that changes have been successfully applied
Alternative Courses	
Step 2. Server fails to apply configuration system rolls back to previous settings.	
Step 4. if the system is failed save then, it will send a notification on th page that saving the changes has failed	

Figure 9. Manage System Functions UseCase Function Description

Signature Flag:



Use case	Signature Flag
Actors	Instructor
Cross references	View Report
Typical Course of Events	
Actor Intentions	**System Responsibility**
	1. The system provides extracted signature report and possible alerts.
2. Instructor check the report and possible alerts.	
3. Instructor flag some students.	
	4. The system sends the flagged students to the cloud.
Alternative Courses	
Step 3. Instructor can flag a student for suspicious and system increase threshold value for this student.	
Step 3. Instructor can flag a student for ignore and system descrease threshold value for this student.	
Step 3. Instructor dont flag anyone and continues.	

Figure 10. Signature Flag UseCase Function Description

2.3 Usability Requirements

The FASDA system must deliver a high level of usability, as instructors need to confirm attendance reliably and efficiently during the course of an academic class. The following measurable attributes of effectiveness, efficiency, and satisfaction will address usability needs.

2.3.1 Task Success Rate (effectiveness):

The goal is for at least 75% of users to successfully and clearly complete the primary tasks of scanning, uploading and identifying attendance reports.

***Justification:** This will show that the system's workflow is intuitive and reliable enough for instructors to successfully carry out their responsibilities with minimal assistance or training.*

2.3.2 Average Task Completion Rate (efficiency):

Each task related to a normal operation (e.g. scanning and verifying an attendance sheet) should be completed in less than 2 minutes in a normal situation.

***Justification:** This will support good practice in the classroom by ensuring that instructors are not overly consumed by administrative tasks during or after class.*

2.3.3 User Satisfaction Score (satisfaction):

A user satisfaction score of at least 80%, as determined by a System Usability Scale (SUS) survey or the like.

***Justification:** This means that their interface is perceived by the user as easy to use, visually clear and aesthetically pleasing to use in day-to-day academic practice.*

2.4 Performance requirements

Response Time:

*The system shall process a scanned attendance sheet and generate verification results within **30 seconds**.*

Detection Accuracy:

*The signature verification model should achieve at least **70% accuracy** in distinguishing between genuine and forged signatures during testing and deployment.*

Throughput:

The system shall be able to analyze 5 attendance sheets per minute in batch mode.

2.5 Logical database requirements

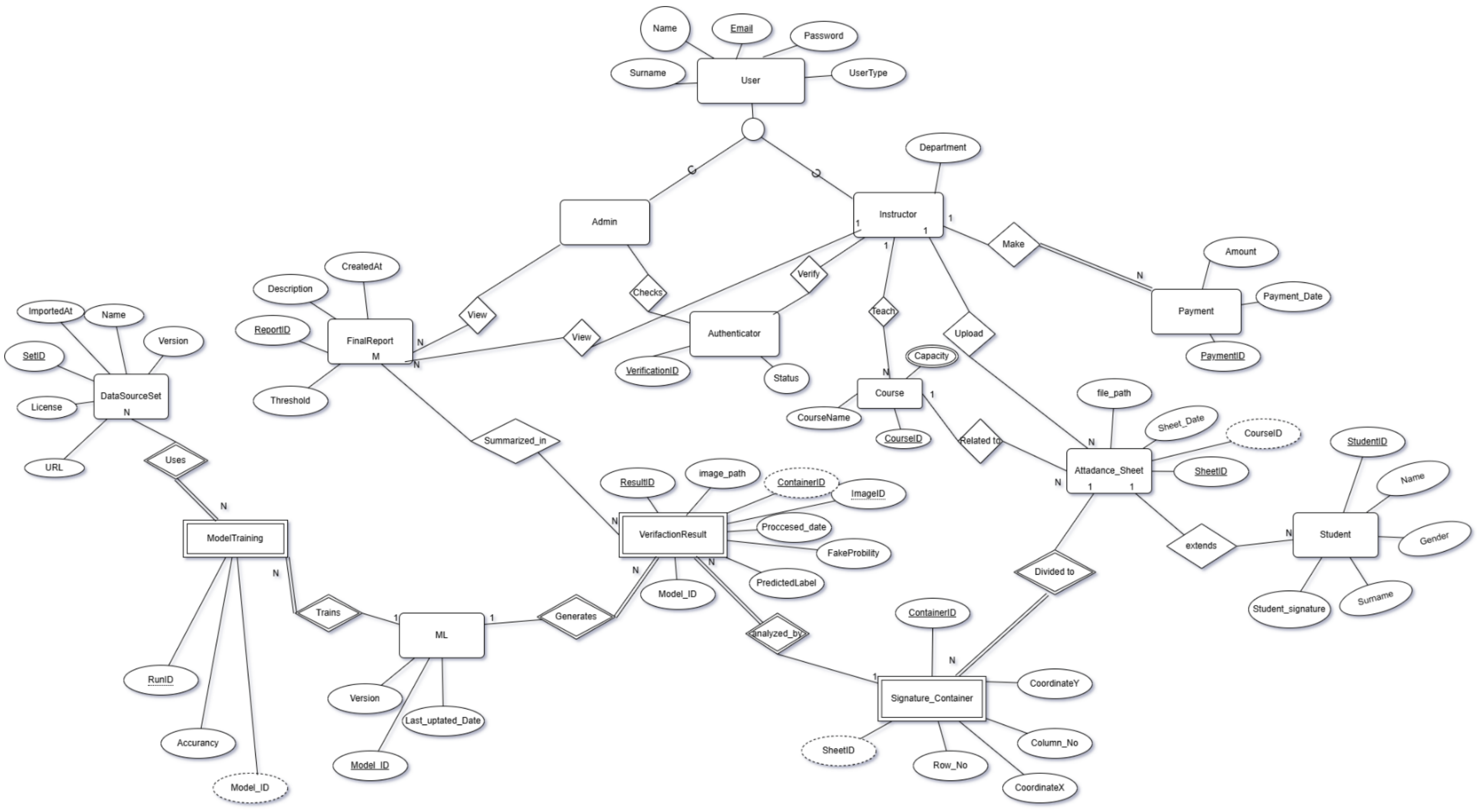


Figure 11. EER Diagram

Entity Descriptions:

Entity	Attributes	Relationships
User	Name, Surname, Email(PK), Password, UserType	ISA: User ⊥ Admin, Instructor (disjoint/partial). Reviews VerificationResult (1–N).
Admin	(inherits from User)	Checks SheerVerification (1–N). Views FinalReport (1–N).
Instructor	Department	Makes Payment (1–N). Teaches Course (1–N). Uploads Attendance_Sheet (1–N). Views FinalReport(1-N).
Student	StudentID (PK), Name, Surname, Gender, Student_Signature	Extends Attendance_Sheet (N–1). Each student may appear in expected signature boxes.
Authenticator	VerificationID (PK), Status	Verified by Instructor (1–N). Checked by Admin (1–N).
Payment	PaymentID (PK), Amount, Payment_Date	Made by Instructor (1–N).
Course	CourseID (PK), CourseName, Capacity	Teachs by Instructor (1–N). Related to Attendance_Sheet (1–N).
Attendance_Sheet	SheetID (PK), Sheet_Date, File_Path, CourseID (FK),	Course (1–N) → Attendance_Sheet. Instructor (1–N) → Attendance_Sheet. Divided into Signature_Container (1–N, identifying).
Signature_Container(Weak Entity)	ContainerID (PK), SheetID (FK), Row_No, Column_No, CoordinateX, CoordinateY	Attendance_Sheet (1–N). Analyzed by VerificationResult (1–N, identifying).
ML	Model_ID (PK), Version, Last_Updated_Date	Trains ModelTraining (1–N). Generates VerificationResult (1–N, identifying).
ModelTraining	RunID (PK), Model_ID (FK), Accuracy	Belongs to ML (N–1). Uses DataSourceSet (N–M).
DataSourceSet	SetID (PK), Name, Version, License, URL, ImportedAt	Used in ModelTraining (N–M).
VerificationResult (Weak Entity)	ResultID (PK), ContainerID (FK), Model_ID (FK), ImageID,Image_Path, Processed_Date, FakeProbability, PredictedLabel,	Weakly dependent on Signature_Container (1–N) and ML (1–N). Summarized in FinalReport (N–M).
FinalReport	ReportID (PK), Description, CreatedAt, Threshold,	Summarized in VerificationResult (N–M).

2.6 Supporting information

N/A

3 Software Estimation+

For this section we estimated project size and development effort for FASDA using function point analysis(FPA), COCOMO, and the Putnam/Jones model. These estimates are required under the software Estimation section of the SRS.

3.1 Function Point Analysis (FPA)

We categorised the crucial functions of FASDA. We labeled them with their complexity, and then mapped them to standard function point weights.

Inputs:

- *Instructor registration with SheerID + payment (High)*
- *Login (Low)*
- *Upload attendance sheet (High)*
- *Manual override / confirm suspicious signature (Medium)*
- *Admin dataset update / retraining trigger (High)*
- *Admin threshold & model settings (Medium)*

Outputs:

- *Forgery detection report for each signature (High)*
- *Instructor dashboard summary (Medium)*
- *Admin analytics / model health dashboard (High)*

Inquiries (search / filtered views):

- *Search previous attendance uploads and reports (Medium)*
- *List flagged / suspicious signatures (Low)*

Logical Data Files (internal DB structures):

- *Users / Instructors / Subscription info (Medium)*
- *AttendanceSheet / SignatureExtract (High)*
- *VerificationResult / FinalReport (Medium)*
- *ModelMetadata / TrainingHistory (Medium)*

External Interface Files:

- *SheerID verification result (Low)*
- *Payment confirmation from bank API (Low)*

With standard IFPUG weights (Inputs 3/4/6 FP,Outputs 4/5/7 FP, Inquires 3/4/6FP,Logical Files 7/10/15 FP, External Files 5/7/10 FP), the total is:

- ***External Inputs:*** *29FP*
- ***External Outputs:*** *19FP*
- ***External Inquires:*** *7FP*
- ***Logical Files:*** *45FP*
- ***Interface Files:*** *10FP*
- ***Unadjusted Function Points(UFP):*** *110FP*

Then if we apply GSC to compute VAF with each characteristic it is scored 0-5 for its Degree of Influence:

General System Characteristic (GSC)	***Degree of Influence (0-5)***	***Explanation***
Data communications	*2*	*Remote access/ web-based upload*
Distributed processing	*2*	*Process is split between web app and ML services*
Performance	*3*	*Report needs to be produced quickly for instructor*
Heavily used configuration	*2*	*Runs on standard server environment*
Transaction Rate	*2*	*Uploads are not constantly high-freq.*
On-Line data entry	*4*	*Instructor enters data and uploads sheets*
End-User efficiency	*3*	*Dashboards need to be easy to read*
On-Line updates	*3*	*Instructor may relabel suspicious signatures*
Complex processing	*4*	*Image analysis uses*
Reusability	*2*	*Core modules reusable across courses/instructors*
Installation ease	*2*	*Web/SaaS style deployment*
Operational ease	*3*	*Instructors should run reports without admin help*
Multiple sites	*1*	*Primarily single-tenant per institution*
Facilitate change	*2*	*Thresholds/retraining can be updated by admin*

Total Degree of Influence: 35

- ***VAF:** 0.65+0.01*35 = **1.00***
- ***ATFP:** UFP *VAF = 110*1.00 ≈ **110***

If we convert function points to Delivered Source ınstructions, we can etmimate language mix and average LOC/FP:

Language/Tech	***Percantage in project***	***LOC per FP***	***Weighted***
Python(ML/analysis)	*%60*	*40*	*24.0*
PHP/Laravel(backend)	*%30*	*50*	*15.0*
SQL/DB logic	*%10*	*30*	*3.0*

*Avg LOC/FP≈ 42.0->LOC≈4620->KDSI = **4.62 KLOC***

3.2 COCOMO (Basic,Semi-Detached)

FASDA project is moderate size/complexity project, so we model it as a semi-detached project.

We use COCOMO semi-detached coefficients as a=3.0,b=1.12,c=2.25,d=0.35.

- *Effort in man-months = a*(KDSI^b) = 3.0*(4.62^1.12) ≈ **16.65 person-months.***
- *Estimated Development Time in Months = 2.5*(Effort)^0.35 ≈ 2.5*(16.65^0.35) ≈ **6.69 months***
- *Estimated Team size = Effort / Time ≈ **2.49 developers***

Development Type	***Semi-Detached***
Estimated Effort	*16.65*
Estimated development Time	*6.69*
Estimated Team size	*2.49*
Actual size	*4*

3.3 Putnam / SLIM Model

For second method, we agreed to use Putnam/Jones model. This method uses AFP and a class exponent which depends on product type and team capability.

We assume the product as being similar to a shrink-wrap / SaaS tool for instructors with average team capability; for that class, the exponent is ~0.42

*Effort->(ATFP^(3*ClassExponent))/27=(110^(3*0.42))/27 ≈ **13.83 person-months***

Development Tİme = 3.0(Effort)^(1/3) ≈3.0*(13.83)^(1/3) ≈ **7.20 months***

Because our team consist of 4 member, we keep the estimated team size as 4.

Adjusted Total FP	*110*
Kind of software	*shrink-wrap/SaaS*
Skill level	*Average*
Estimated Effort	*13.83*
Estimated Development Time(in months)	*7.20*
Team size	*4*

In conclusion, the three different method shows us with team of 4 people can deliver FASDA within ~6-8 months including integration,testing and reporting.

4 References

Bay Ayzeren, Y., Erbilek, M., & Çelebi, E. (2019). Emotional state prediction from online handwriting and signature biometrics. IEEE Access, 7, 164255–164267. https://doi.org/10.1109/ACCESS.2019.2952313

C.P. Jetlin, G. Mahalakshmi, D. Divya, and P. Babu, "Handwritten Recognition Using Hybrid CNN-SVM Model for Cerebral Palsy," Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024). https://ssrn.com/abstract=5086728

Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Offline handwritten signature verification—Deep transfer learning. https://arxiv.org/abs/1507.07909

Kumar, R., Sharma, M., & Tiwari, P. (2020). Offline signature verification using Siamese network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3525–3533). https://openaccess.thecvf.com/content_WACV_2020/html/Kumar_Offline_Signature_Verification_Using_Siamese_Network_WACV_2020_paper.html

Reni, R. (2021). Signature verification dataset. Kaggle. https://www.kaggle.com/datasets/robinreni/signature-verification-dataset

*Srihari, S. N., Cha, S. H., Arora, H., & Lee, S. (2002). Individuality of handwriting. Journal of Forensic Sciences, 47(4), 856–872.*https://www.ojp.gov/pdffiles1/nij/grants/190133.pdf

Yatbaz, H. Y., & Erbilek, M. (2020). Deep learning-based stress prediction from offline signatures. In 2020 International Workshop on Biometrics and Forensics (IWBF) (pp. 1–6). IEEE. https://doi.org/10.1109/IWBF49977.2020.9107942

DocuSign Inc. (n.d.). AI verification technologies for digital document security. Retrieved October 23, 2025, from https://www.docusign.com

Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Offline handwritten signature verification—Deep transfer learning. arXiv preprint arXiv:1706.02204. https://arxiv.org/abs/1706.02204

OpenCV Team. (2024). OpenCV documentation (version 4.x). OpenCV Foundation. https://docs.opencv.org

TensorFlow Team. (2024). TensorFlow documentation (version 2.x). Google LLC. https://www.tensorflow.org

PostgreSQL Global Development Group. (2024). PostgreSQL documentation (version 16). https://www.postgresql.org/docs/

METU Northern Cyprus Campus (BAYEK). (2024). Ethics committee guidelines for biometric research. Middle East Technical University. https://ncc.metu.edu.tr/bayek

Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine). https://ics.uci.edu/~fielding/pubs/dissertation/top.htm

Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT): RFC 7519. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc7519

Sammut, C., & Webb, G. I. (Eds.). (2011). Encyclopedia of machine learning and data mining. Springer. https://pzs.dstu.dp.ua/DataMining/bibl/Encyclopedia%20Machine%20Learning%202011.pdf

Sophos. (2020). "What is a Signature." Sophos News. [Online]. Available: https://home.sophos.com/en-us/security-news/2020/what-is-a-signature.

wikiHow Staff. (2025). "How to Spot a Fake Autograph." wikiHow. [Online]. Available: https://www.wikihow.com/Spot-a-Fake-Autograph.

5 Appendices

5.1 Acronyms and abbreviations

- ***FASDA:*** *Fake Attendance Signature Detection Application*
- ***SRS:*** *Software Requirements Specification*
- ***ML:*** *Machine Learning*
- ***DL:*** *Deep Learning*
- ***IP:*** *Image Processing*
- ***CNN:*** *Convolutional Neural Network*
- ***BAYEK:*** *Bilimsel Araştırma ve Yayın Etiği Kurulu (The Scientific Research and Publication Ethics Committee of METU NCC)*
- ***PDF:*** *Portable Document Format*
- ***JPEG:*** *Joint Photographic Experts Group (File format for scanned image uploads)*
- ***TIFF:*** *Tagged Image File Format (File format for scanned image uploads)*
- ***CSV:*** *Comma Separated Values (Report export format)*
- ***ROC:*** *Receiver Operating Characteristic (A metric for ML model evaluation)*
- ***EER (ML):*** *Extended entity relationship*
- ***ERD:*** *Entity Relationship Diagram (Database design model)*
- ***API:*** *Application Programming Interface (Used for third-party service interaction)*
- ***SUS:*** *System Usability Scale (A survey for measuring user satisfaction)*
- ***PK:*** *Primary Key (Database attribute)*
- ***FK:*** *Foreign Key (Database attribute)*
- ***FPA:*** *Function Point Analysis (A software effort estimation method)*
- ***IFPUG:*** *International Function Point Users Group (An organization that sets FPA standards)*
- ***FP:*** *Function Point (Unit of measurement in FPA)*
- ***GSC:*** *General System Characteristic (A factor used in FPA calculation)*
- ***VAF:*** *Value Adjustment Factor (A component used in FPA calculation)*
- ***ATFP:*** *Adjusted Total Function Points (The resulting metric from FPA)*
- ***LOC:*** *Lines Of Code (Software size metric)*
- ***PHP:*** *PHP: Hypertext Preprocessor (Programming language/backend technology)*
- ***KDSI:*** *Kilo Delivered Source Instructions (Software size in thousands of lines)*
- ***KLOC:*** *Kilo Lines Of Code (Software size in thousands of lines)*

5.2 Sprint 1 Overview

5.2.1 Sprint Backlog

ID	*Sprint Backlog Item*	*Expected Number of Hours*	*Week 1*	*Week 2*	*Week 3*	*Week 4*	*Week 5*	*Week 6*
1	***Dataset Search and Selection (GPDS, CEDAR)***	*3*	*1.5*	*1.5*	*0*	*0*	*0*	*0*
2	***Dataset Licensing and Agreement Preparation***	*2*	*1*	*1*	*0*	*0*	*0*	*0*
3	***ML Model Research for Forgery Detection (CNN, SVM, Siamese)***	*4*	*2*	*1*	*1*	*0*	*0*	*0*
4	***Collaborative SRS Document Setup (Google Docs)***	*3*	*1.5*	*1.5*	*0*	*0*	*0*	*0*
5	***Instructor Discussions & Feedback Sessions***	*3*	*1.5*	*1*	*0.5*	*0*	*0*	*0*
6	***SheerID and Bank API Research for Verification Workflow***	*2*	*1*	*1*	*0*	*0*	*0*	*0*
7	***Use Case Diagram and Workflow Update***	*1.5*	*0.5*	*1*	*0*	*0*	*0*	*0*
8	***Software Estimation (COCOMO + Jones Model)***	*1.5*	*0.5*	*0.5*	*0.5*	*0*	*0*	*0*
—	***Total Hours per Week***	***9.5***	***8.5***	***2.5***	***0***	***0***	***0***	
Total Sprint Hours	***20.5***							

5.2.2 Sprint Burndown Chart

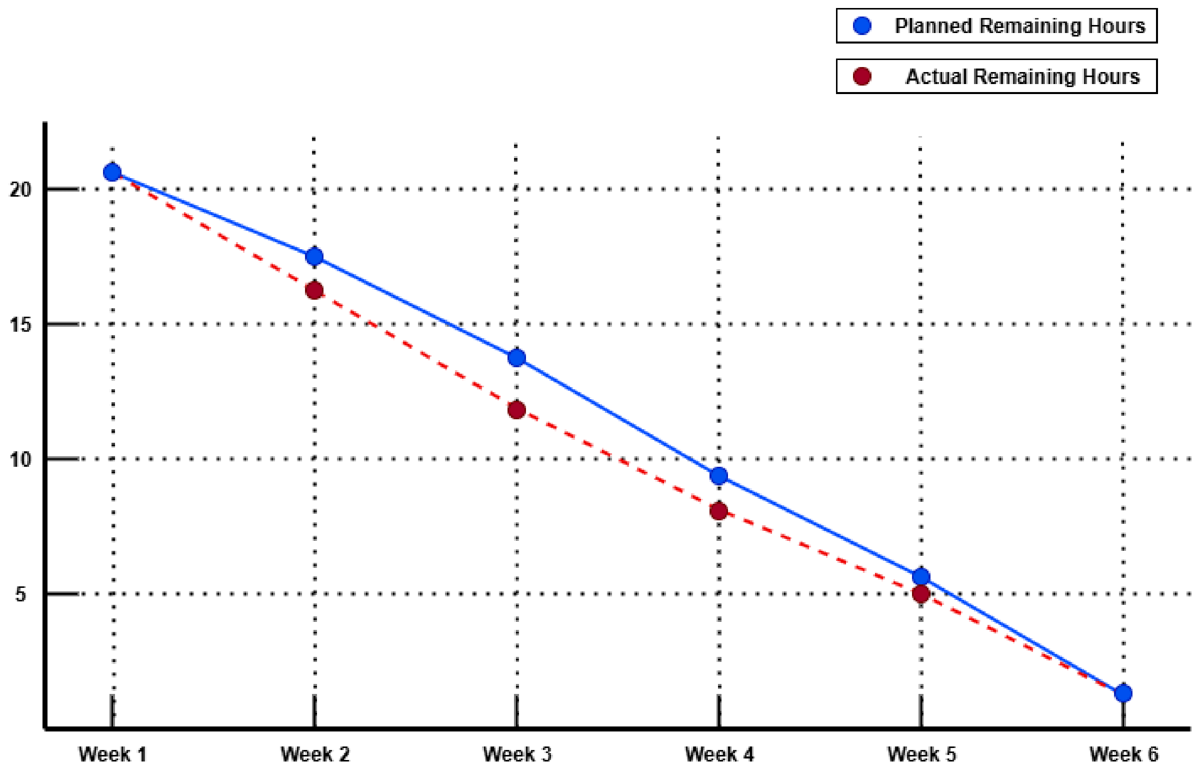


Figure 12. Burndown Chart

5.2.3 Sprint Review

In case of creating the foundations the spring 1 has met its goal in case of task distribution and working methodically. Important datasets have been found and decided which to use. Key machine learning models have been reviewed. The documentation for the licensing of the datasets has been gathered. Some of the licensing problems occurred during the sprint because of the old documentation.

5.2.4 Sprint Retrospective

Every single task was equally distributed between the group members. While one member was working on the datasets, other member found a suitable ml model and others dealt with ethics. We successfully found necessary datasets which we will use and suitable ML for our project. Every team member collaborated with each other and shared common documents based on Google doc. Every team member researched online sources about fake and genuine signatures.

For the next sprint, we will plan daily meetings for improvement. Also, we will take necessary ethics permissions from Metu NCC (BAYEK). Our team will apply for licenses for datasets. We will set a format for the attendance sheet for better results and accuracy improvement. We will start preparations for SDD report.