



**UNIVERSITÀ DI PISA**

***Department of Information Engineering***

**MSc in Computer Engineering**

**Software System Engineering Course Project**

**SMART APPLIANCE MANAGER**

Students:

MAZZANTI CARLO

MOHAMMAD TESFAYE YIMAM

SIMONCINI ALESSIO

VEZZANI LORENZO

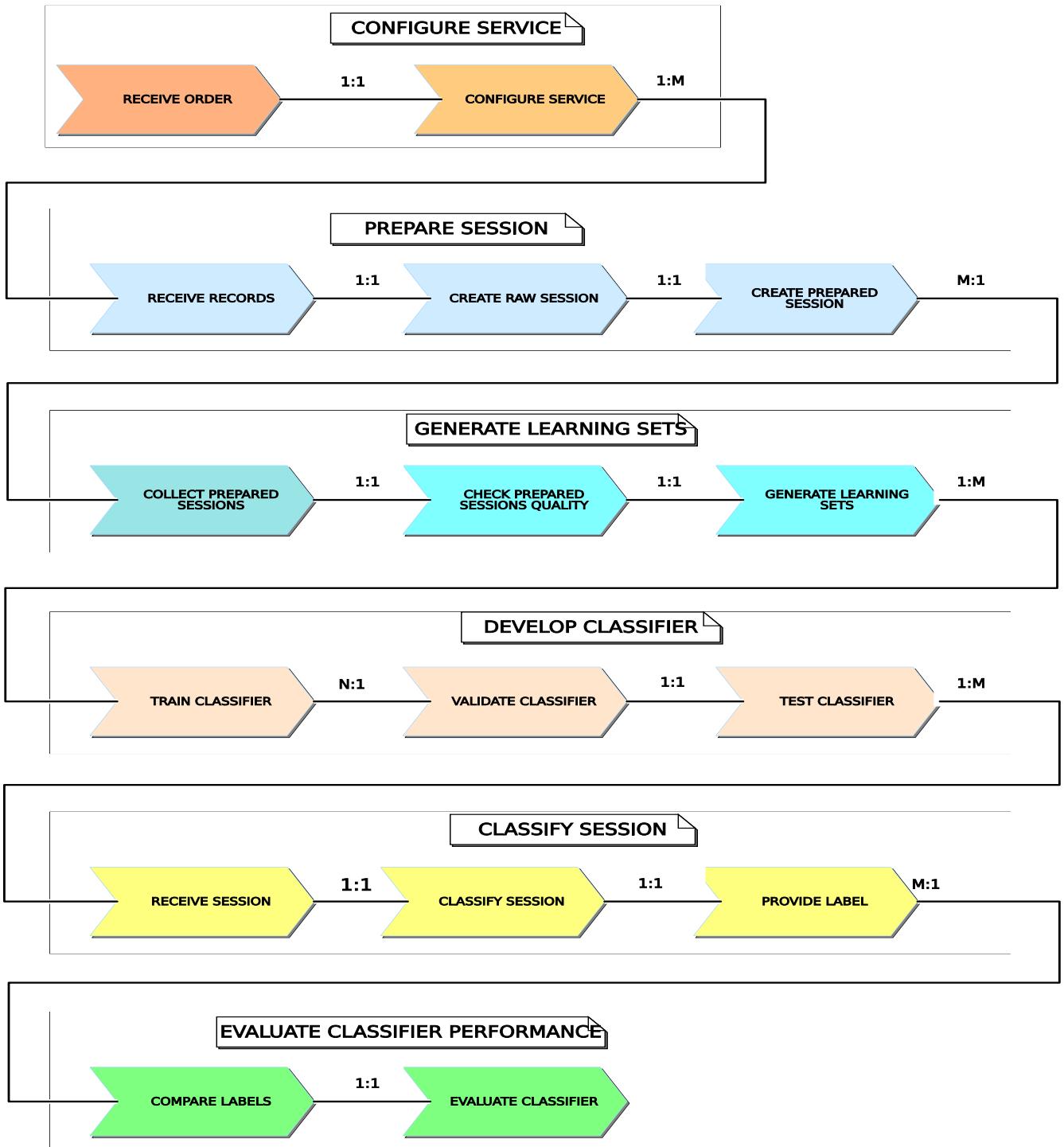
## Table of Contents

Workflow Requirements: Process Landscape Diagram (Everyone) .....	4
Workflow Requirements: BPMN Modelling .....	5
1 - CONFIGURATION PROCESS(Everyone) .....	5
2 - PREPARE SESSION PROCESS (Carlo) .....	6
3 - GENERATE LEARNING-SET PROCESS (Lorenzo) .....	7
4 - DEVELOP CLASSIFIER PROCESS(Tesfaye) .....	7
5 - CLASSIFY SESSION PROCESS (Tesfaye).....	8
6 - EVALUATE CLASSIFIER PERFORMANCE PROCESS (Alessio) .....	9
Workflow ANALYSIS: Human Tasks and Use-Case Diagrams.....	10
1. CLIENT-SIDE SYSTEM(Tesfaye).....	10
Use-case Diagram.....	10
Human Task: Configure Client-Side System .....	10
2. INGESTION SYSTEM (Carlo) .....	12
Use-case Diagram.....	12
Human task: Configure Ingestion System.....	12
3. PREPARATION SYSTEM(Alessio) .....	13
Use-case Diagram.....	13
Human Task: Configure Preparation System.....	14
4. SEGREGATION SYSTEM (Lorenzo) .....	16
Use-case Diagram.....	16
Human Task: Check Data Balancing .....	17
Human Task: Check Data Coverage.....	18
Human Task: Configure the Segregation System .....	19
5. DEVELOPMENT SYSTEM (Tesfaye, Carlo) .....	21
Use-case Diagram.....	21
Human Task: Set #Iterations .....	21
Human Task: Check Calibration Plot .....	22
Human Task: Check Validation Results.....	24
Human Task: Check Test Results .....	25
Human Task: Configure Development System .....	26
6. CLASSIFICATION SYSTEM(Alessio) .....	28
Use-case Diagram.....	28
Human Task: Configure Classification System.....	28
7. EVALUATION SYSTEM (Alessio) .....	29

Use-case Diagram.....	29
Human Tasks: Configure Evaluation System.....	30
Human Task: Evaluate Classifier.....	31
Workflow ANALYSIS: Data and Application Logic Models.....	33
1. INGESTION SYSTEM (Carlo) .....	33
Data model .....	33
Application logic model.....	34
2. PREPARATION SYSTEM(Tesfaye) .....	36
Data Model.....	36
Application Logic Model.....	37
3. SEGREGATION SYSTEM (Lorenzo V) .....	39
Data Model.....	39
Application Logic Model.....	39
4. DEVELOPMENT SYSTEM(Tesfaye) .....	41
Data Model.....	41
Application Logic Model.....	42
5. CLASSIFICATION SYSTEM (Alessio) .....	44
Data model .....	44
Application logic model.....	44
6. EVALUATION SYSTEM (Alessio) .....	46
Data model .....	46
Application logic model.....	46
Workflow Design: UML class and Sequence diagrams.....	48
1. PREPARE SESSION PROCESS (Carlo) .....	48
2. PREPARATION SYSTEM.....	50
3. SEGREGATION SYSTEM.....	50
4. DEVELOPMENT SYSTEM (Tesfaye).....	51
5. CLASSIFICATION SYSTEM (Alessio) .....	54
6. EVALUATION SYSTEM (Alessio) .....	54
Workflow Implementation:.....	56
Deployment Diagram .....	56
Workflow Testing: .....	57
1. RESPONSIVENESS-ELASTICITY of the production phase .....	57
2. AUTOMATION RESPONSIVENESS-ELASTICITY of the development phase .....	58
3. RESILIENCY .....	59

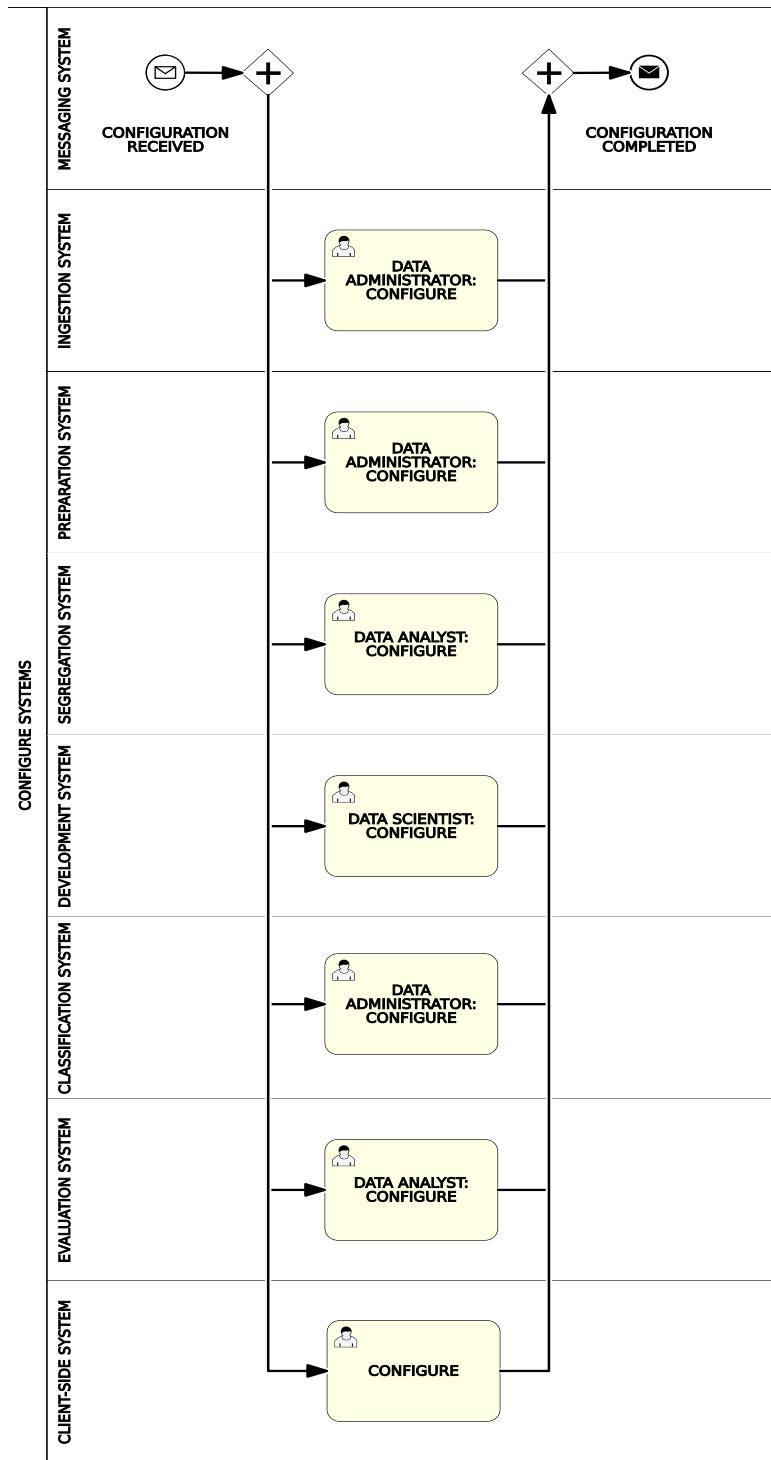
Ingestion System .....	59
Preparation System .....	59
Segregation System .....	59
Development System .....	60
Classification System .....	60
Evaluation System .....	60

# Workflow Requirements: Process Landscape Diagram (Everyone)

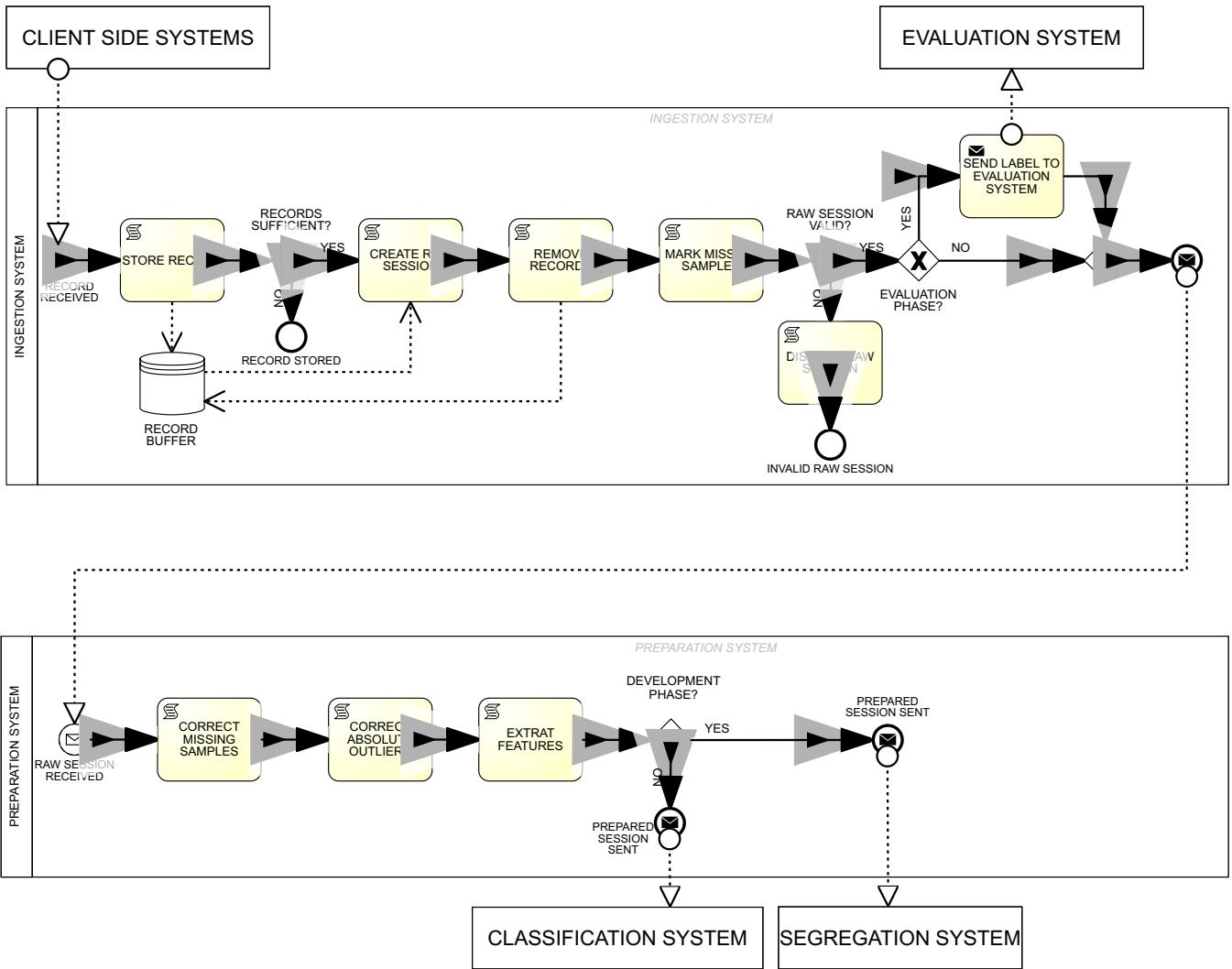


# Workflow Requirements: BPMN Modelling

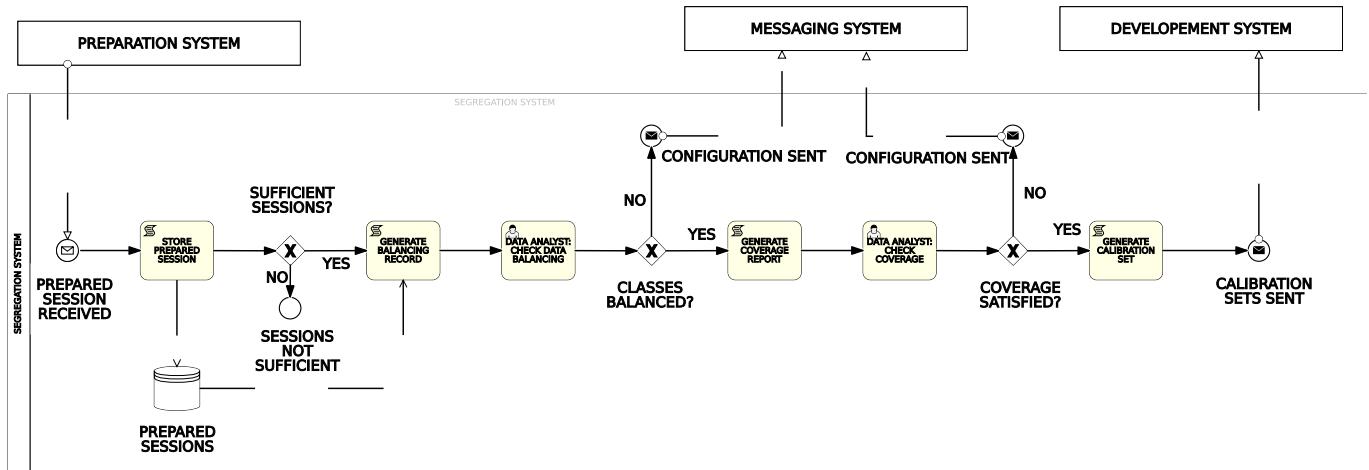
## 1 - CONFIGURATION PROCESS(Everyone)



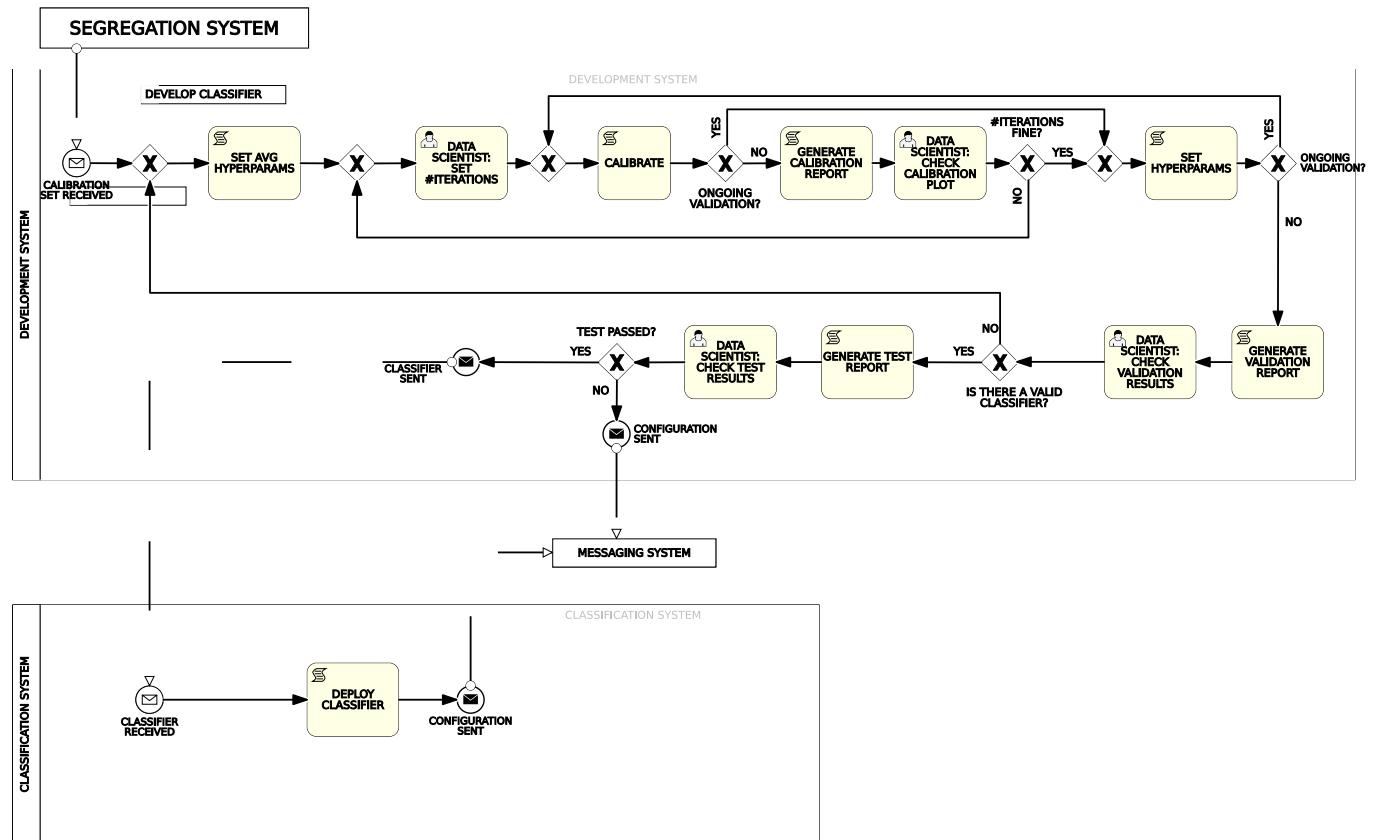
## 2 - PREPARE SESSION PROCESS (Carlo)



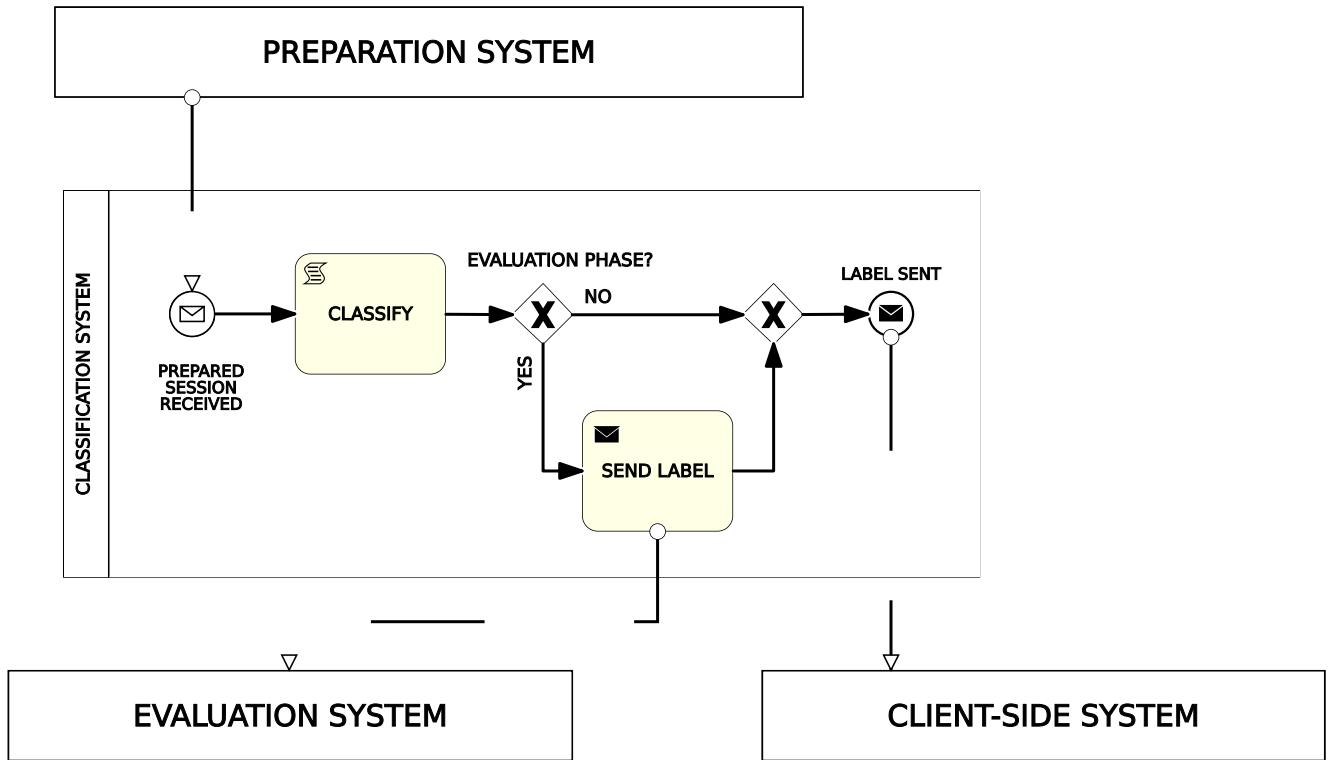
### 3 - GENERATE LEARNING-SET PROCESS (Lorenzo)



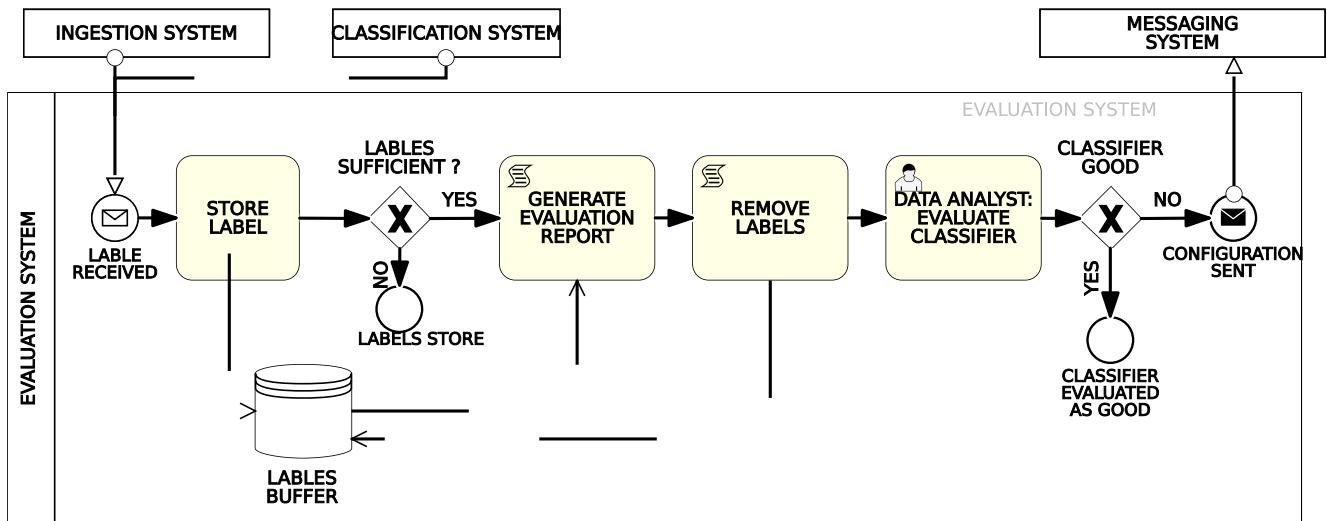
### 4 - DEVELOP CLASSIFIER PROCESS(Tesfaye)



## 5 - CLASSIFY SESSION PROCESS (Tsfaye)



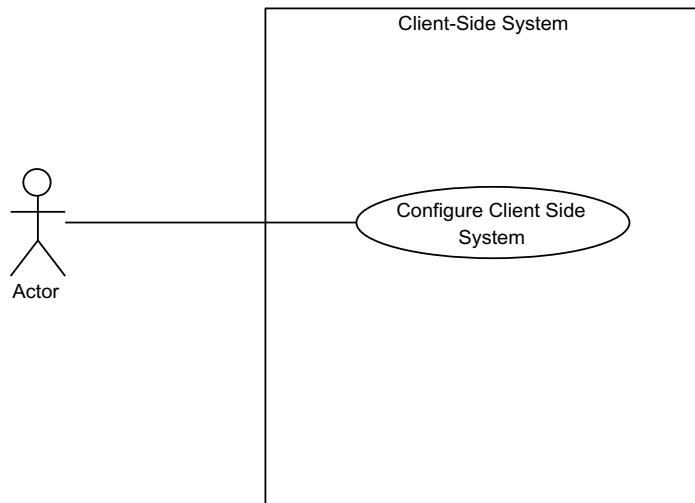
## 6 - EVALUATE CLASSIFIER PERFORMANCE PROCESS (Alessio)



# Workflow ANALYSIS: Human Tasks and Use-Case Diagrams

## 1. CLIENT-SIDE SYSTEM(Tesfaye)

Use-case Diagram



Human Task: Configure Client-Side System

*Mock-up*

**Configure Client Systems**

Src IP address:	<input type="text" value="10.5.0.1"/>
Dst IP address:	<input type="text" value="10.5.0.2"/>
Data Sending Interval:	<input type="text" value="10"/>

**Apply**

### *Scenario*

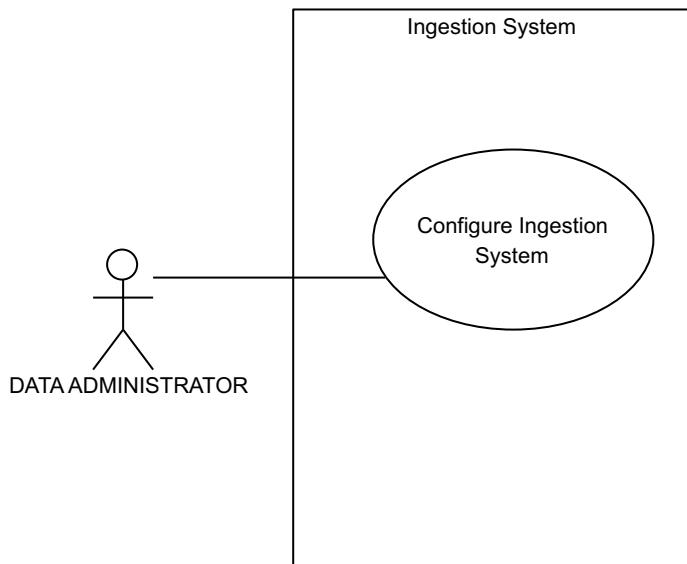
- |  |
|--|
| 1.The use case starts when the <b>System</b> shows the <b>Client-Side Configuration Panel</b>  |
| 2. The  <b>Actor</b> digits the Source IP address in the <Src IP address> input field             |
| 3. The  <b>Actor</b> digits the Destination IP address in the <Dst IP address> input field.       |
| 4. The  <b>Actor</b> digits the Data feeding Interval in the <Data Sending Interval> input field. |
| 5. The  <b>Actor</b> presses on the <Apply> button.   |

### *Details*

Name	Value
Post-conditions	The client system is successfully configured.
	The configuration pop-up is closed.
	The system is now operating with the updated configuration.

## 2. INGESTION SYSTEM (Carlo)

Use-case Diagram



Human task: Configure Ingestion System

*Mock-up*

The mock-up interface consists of several input fields and a dropdown menu:

- IP address of Evaluation System: 192.168.37.14
- IP address of Preparation System: 192.168.37.89
- Minimum records per session: 40
- Missing samples threshold: 5
- Current phase: Development phase (selected)
- Other options in the dropdown: Production phase, Evaluation phase

A blue 'Submit' button is located at the bottom left of the form.

### Scenario

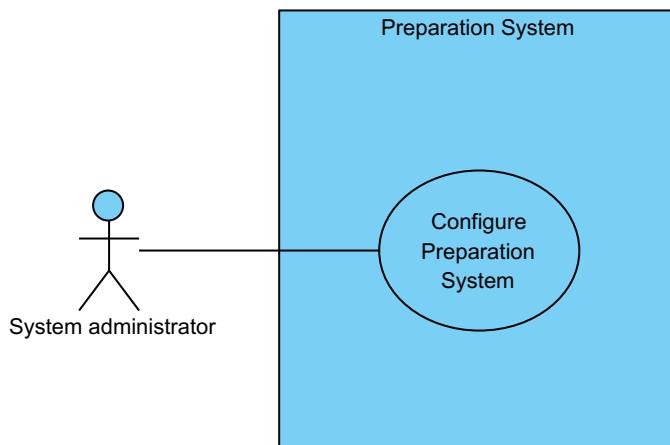
1. The use case starts when the  DATA ADMINISTRATOR opens the configuration interface
  2. **SYSTEM** shows inputs for IP address of Evaluation system, IP address of Ingestion system, configuring minimum records per session, missing samples threshold, current phase
  3.  DATA ADMINISTRATOR enters IP addresses of the Evaluation system and Ingestion system
  4.  DATA ADMINISTRATOR enters the minimum records per session
  5.  DATA ADMINISTRATOR enters the missing samples thresholds
  6. **if** the current phase is Development phase
    - 6.1.  DATA ADMINISTRATOR selects Development phase
  7. **else if** the current phase is Production phase
    - 7.1.  DATA ADMINISTRATOR selects Production phase
  8. **else if** the current phase is Evaluation phase
    - 8.1.  DATA ADMINISTRATOR selects Evaluation phase
- end if**

### Details

Post-conditions	The Ingestion System is configured
-----------------	------------------------------------

## 3. PREPARATION SYSTEM(Alessio)

Use-case Diagram



Human Task: Configure Preparation System

Mock-up

# Configure Preparation System

Appliance current:

Lower bound:  Am<sub>I</sub>

Upper bound:  Am<sub>I</sub>

Environmental temperature:

Lower bound:  °C

Upper bound:  °C

Appliance voltage:

Lower bound:  Volt

Upper bound:  Volt

Environmental humidity

Lower bound:  %

Upper bound:  %

Appliance temperature:

Lower bound:  °C

Upper bound:  °C

House occupancy

Lower bound:  Pec

Upper bound:  Pec

IP address Ingestion system:

IP address Segregation system:

IP address Classification system:

Development phase?

### Scenario

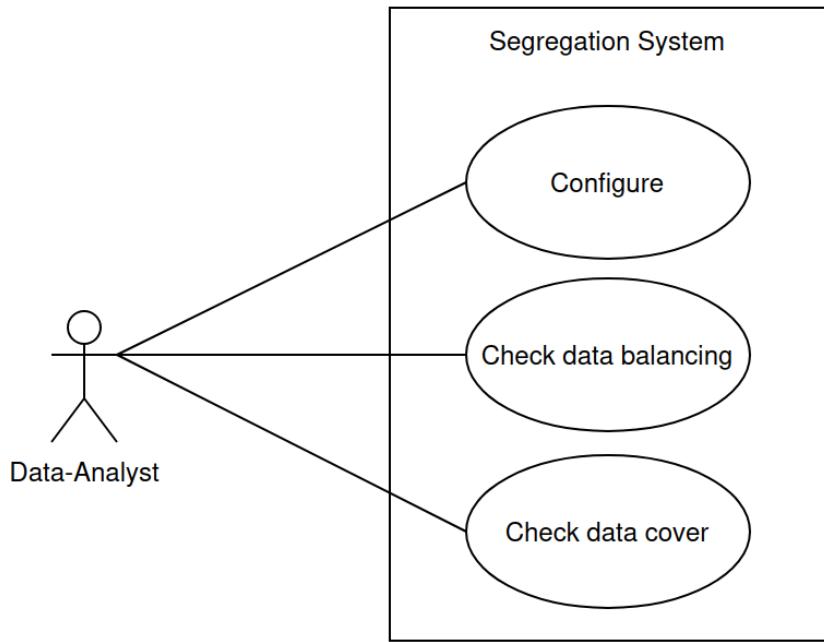
1. Use case starts when the  <b>System administrator</b> opens the <b>Configure Preparation System</b> interface
2. <b>SYSTEM</b> Displays all the parameters and fields to be set
3. The  <b>System administrator</b> sets the <i>Appliance current lower bound</i>
4. The  <b>System administrator</b> sets the <i>Appliance current upper bound</i>
5. The  <b>System administrator</b> sets the <i>Appliance voltage lower bound</i>
6. The  <b>System administrator</b> sets the <i>Appliance voltage upper bound</i>
7. The  <b>System administrator</b> sets the <i>Appliance temperature lower bound</i>
8. The  <b>System administrator</b> sets the <i>Appliance temperature upper bound</i>
9. The  <b>System administrator</b> sets the <i>Environmental temperature lower bound</i>
10. The  <b>System administrator</b> sets the <i>Environmental temperature upper bound</i>
11. The  <b>System administrator</b> sets the <i>Environmental humidity lower bound</i>
12. The  <b>System administrator</b> sets the <i>Environmental humidity upper bound</i>
13. The  <b>System administrator</b> sets the <i>House occupancy lower bound</i>
14. The  <b>System administrator</b> sets the <i>House occupancy upper bound</i>
15. The  <b>System administrator</b> sets the ' <i>IP address Ingestion system</i> ' field
16. The  <b>System administrator</b> sets the ' <i>IP address Segregation system</i> ' field
17. The  <b>System administrator</b> sets the ' <i>IP address Classification system</i> ' field
18. The  <b>System administrator</b> sets the configuration by pressing the <b>YES</b> button or the <b>NO</b> button
19. The  <b>System administrator</b> saves the configuration by pressing the <b>Save configuration</b> button

### Details

Name	Value
Post-conditions	The phase of the preparation system is set The outlier parameters are set The connected systems' IP addresses are set

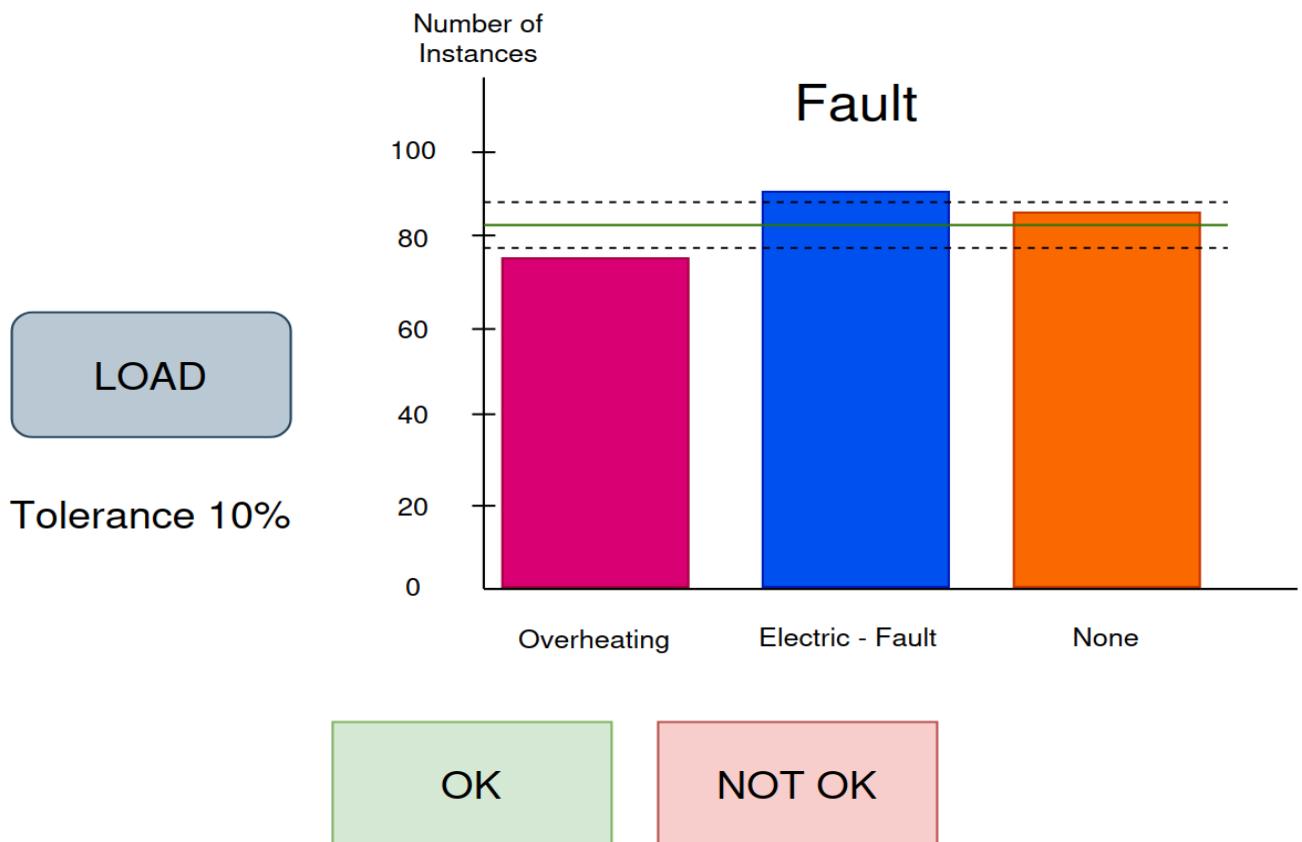
#### 4. SEGREGATION SYSTEM (Lorenzo)

Use-case Diagram



## Human Task: Check Data Balancing

Mock-up

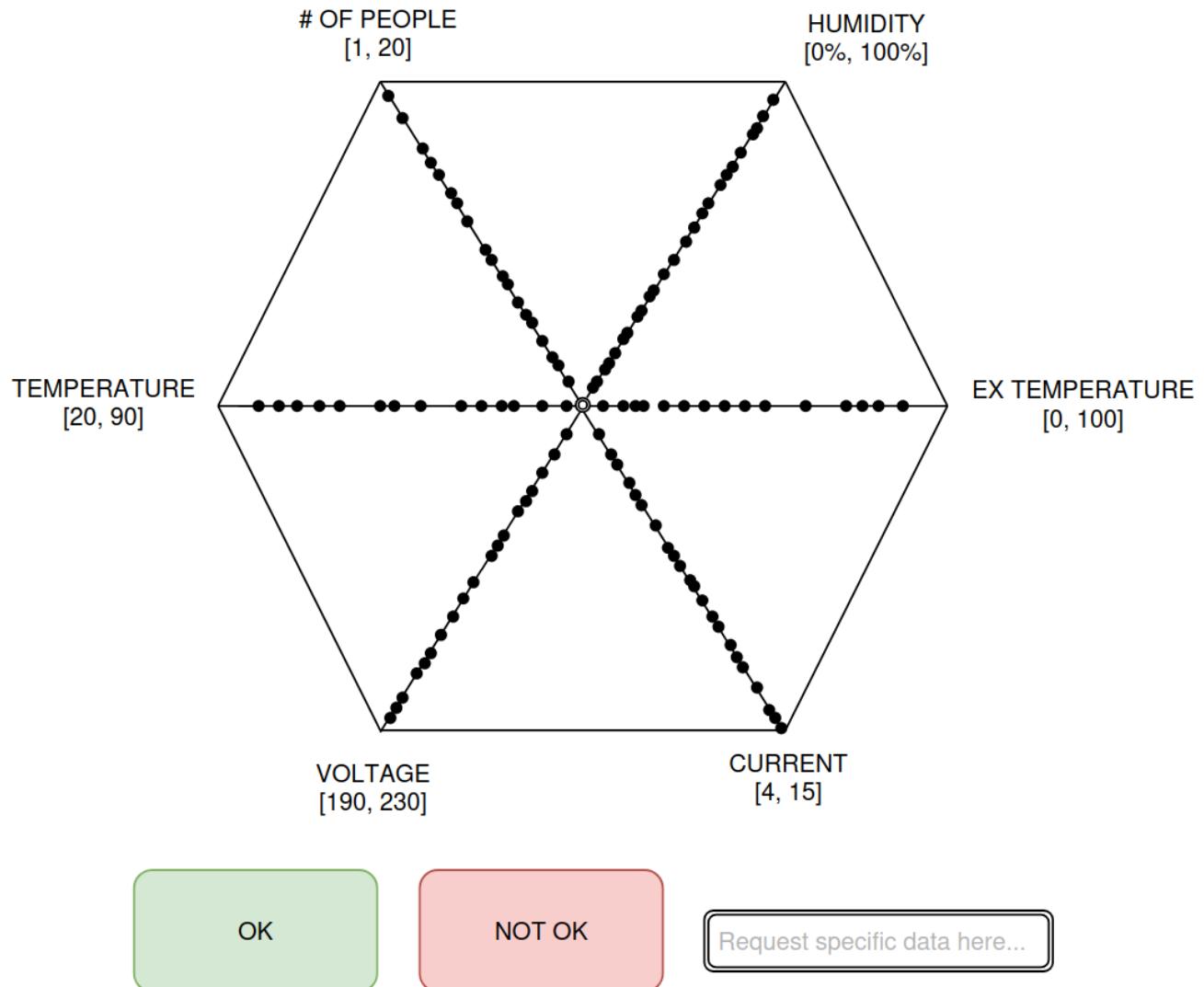


### Scenario

1. The  <b>Data Analyst</b> launches the software
2. <b>SYSTEM</b> Displays GUI for the  <b>Data Analyst</b>
3. The  <b>Data Analyst</b> presses the LOAD button
4. <b>for each</b> class
4.1. <b>if</b> the bar is not inside the interval
4.1.2. The  <b>Data Analyst</b> selects NOT OK
4.1.3. The use case terminates.
<b>end if</b>
<b>end for</b>
5. The  <b>Data Analyst</b> selects OK.

## Human Task: Check Data Coverage

Mock-up



## Scenario

1.The  Data Analyst launches the software
2. SYSTEM Displays GUI for the  Data Analyst
3. The  Data Analyst presses the LOAD button
4. if the radar scatterplot is not sufficiently well distributed according to  Data Analyst
4.1 The  Data Analyst requires for specific data using the textbox
4.2. The  Data Analyst selects "NOT OK"
4.3. The use case terminates.
end if
5. The  Data Analyst selects "OK".

Human Task: Configure the Segregation System

Mock-up

### Configure Segregation System

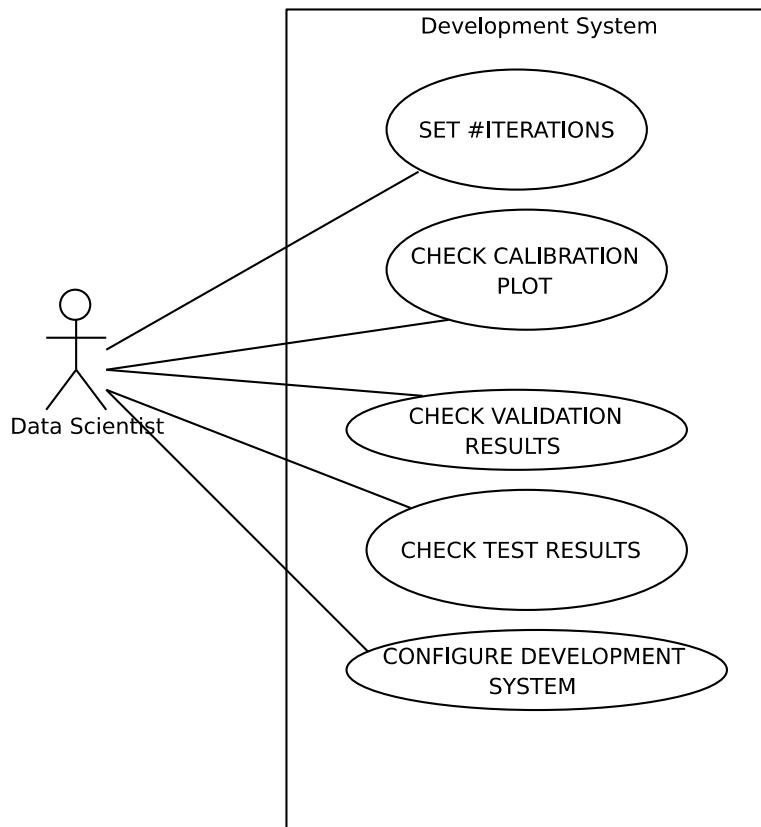
Select Number of Sessions	500 <input type="button" value="↔"/>
Select Tolerance Interval	10% <input type="button" value="↔"/>
Insert Preparation System IP	192.168.0.1
Insert Messaging System IP	192.168.0.2
Insert Development System IP	192.168.0.3

*Scenario*

1. The  Data Analyst launches the software.
2. The  Data Analyst selects the number of sessions.
3. The  Data Analyst selects a tolerance percentage.
4. The  Data Analyst sets Preparation System IP address.
5. The  Data Analyst sets Messaging System IP address.
6. The  Data Analyst sets Development System IP address.
7. The  Data Analyst selects OK.

## 5. DEVELOPMENT SYSTEM (Tesfaye, Carlo)

Use-case Diagram



Human Task: Set #Iterations

Mock-up

The mock-up interface is a rectangular form with a light gray background. At the top, a green header bar contains the text "#Iteration Settings" in white. Below the header, the text "Enter # of iterations:" is displayed in bold black font. To the right of this text is a white input field containing the number "30". At the bottom of the form is a large green button with the word "Apply" in white.

### Scenario

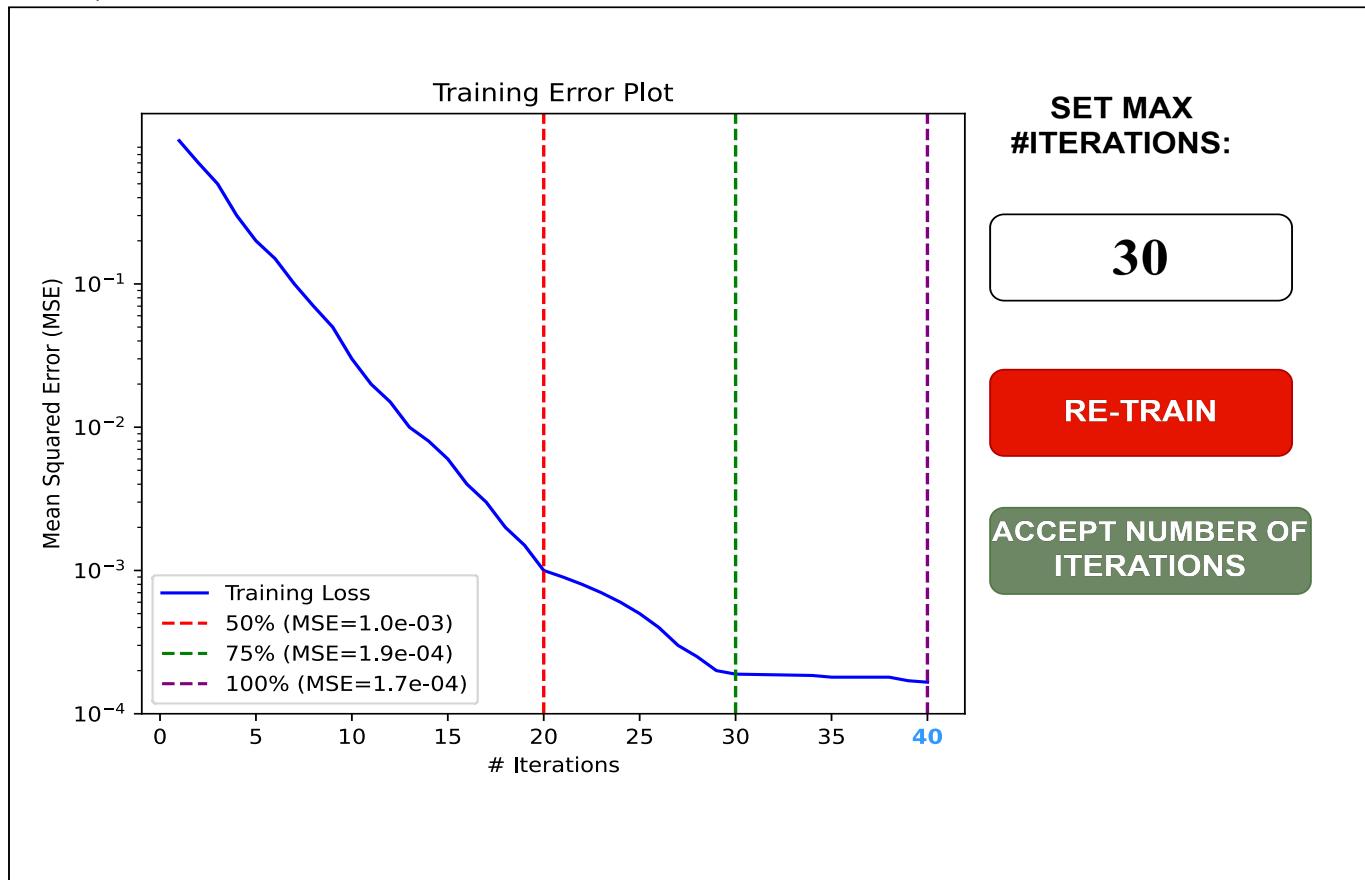
- 1.The use case starts when the  Data Scientist opens the UI to set the # of iterations.
- 2.The  Data Scientist inserts the #iterations from the keyboard in the <Enter # of iterations> field.
- 3.The  Data Scientist presses the <Apply> button.

### Details

Name	Value
Post-conditions	The development system updates the number of calibration/learning iterations.

### Human Task: Check Calibration Plot

#### Mock-up



## Scenario

1. The use case starts when the  Data Scientist opens on <Check Calibration Plot>
2.SYSTEM shows a pop-up window with the Mean Squared Error (MSE) plot against the number of iterations.
3. <b>if</b> the  Data Scientist observes that the MSE curve becomes almost flat well before the last iterations <b>then</b>
3.1. The  Data Scientist reduces the number of iterations via the provided field to manage overfitting.
3.2. The  Data Scientist press the <RE-TRAIN> button
3.3. SYSTEM updates the plot with the new number of iterations
4. <b>else if</b> the  Data Scientist sees that the MSE curve is still clearly decreasing near the final iterations
4.1. The  Data Scientist increases the number of iterations based on the provided fields and the button on the pop-up window.
4.2. The  Data Scientist press the <RE-TRAIN> button
4.3. SYSTEM updates the plot with the new number of iterations
5. <b>else if</b> the  Data Scientist considers the current number of iterations are adequate
5.1. The  Data Scientist presses the <ACCEPT NUMBER OF ITERATIONS> button.
6. <b>end if</b>

## Details

Name	Value
Post-conditions	<p><b>Alternative Flow 1:</b></p> <p>1.The Training Error Report has been analysed.</p> <p>2.The system continued with its execution by setting the hyperparameters according to the grid search.</p> <p><b>Alternative Flow 2:</b></p> <p>1.The number of <b>iterations</b> has been tuned.</p> <p>2.A new neural network training has started.</p>

## Human Task: Check Validation Results

Mock-up

Validation Results							
	ID	Validation Error	Training Error	Difference	Number of Layers	Number of Neurons	Complexity
<input checked="" type="checkbox"/>	1	0.12	0.10	0.02	7	20	140
<input type="checkbox"/>	2	0.15	0.12	0.03	3	40	120
<input type="checkbox"/>	3	0.20	0.15	0.05	7	30	210
<input type="checkbox"/>	4	0.30	0.20	0.10	8	50	400
<input type="checkbox"/>	5	0.40	0.21	0.19	8	100	800

OVERFITTING TOLERANCE: 0.05
CONFIRM
REFUSE ALL NETWORKS

### Scenario

1. the use case starts when **SYSTEM** visualizes the test results for the top 5 classifiers
2. **for each** classifier in the report
  - 2.1. **SYSTEM** shows network ID, Validation Error, Training Error, Difference, Number of Layers, Number of Neurons, Complexity
3. **end for each**
4. **SYSTEM** shows Overfitting Tolerance
5. **for each** classifier in the report
  - 4.1.  **DATA SCIENTIST** checks the difference
6. **end for each**
7. **if**  **DATA SCIENTIST** sees no classifiers with difference lower than the overfitting tolerance
  - 5.1.  **DATA SCIENTIST** clicks on "REFUSE ALL NETWORKS" button
8. **else**
  - 6.1. **if** there are multiple networks with difference lower than the overfitting tolerance
    - 6.1.1.  **DATA SCIENTIST** selects the network with lowest difference
  - 6.2. **if** there are multiple networks with similar difference w.r.t. the lowest difference
    - 6.2.1.  **DATA SCIENTIST** selects the network with the lowest complexity
9. **end if**
10. **if**  **DATA SCIENTIST** clicks on "CONFIRM" button

end if

#### Details

<b>Preconditions</b>	The validation report has been generated
<b>Post-conditions</b>	the best classifier is selected or a new training is started

#### Human Task: Check Test Results

##### Mock-up

The mock-up shows a user interface for reviewing test results. At the top, a table displays the 'Test results of the winner network' with three rows: Validation Error (0.12), Test Error (0.20), and Difference (0.08). Below this, a message box shows 'GENERALIZATION TOLERANCE: 0.10'. To the right is a blue 'APPROVE' button. Further down, there is a section for suggesting a new layers interval with 'min: 10' and 'max: 100'. To the right of this is a red 'DENY' button.

Test results of the winner network	
Validation Error	0.12
Test Error	0.20
Difference	0.08

GENERALIZATION TOLERANCE: 0.10

APPROVE

Suggest new layers interval (optional)

min: 10

max: 100

DENY

#### Scenario

1. the use case starts when the [DATA SCIENTIST](#) opens the test results
  2. **SYSTEM** shows validation error, test error and difference of the winner network
  3. **SYSTEM** shows the generalization tolerance
  4. [DATA SCIENTIST](#) checks the results
  5. **if** [DATA SCIENTIST](#) sees difference lower than the generalization tolerance
    - 5.1. [DATA SCIENTIST](#) clicks on the "APPROVE" button
  6. **else**
    - 6.1. [DATA SCIENTIST](#) optionally enters a suggestion on the new interval of number of layers
    - 6.2. [DATA SCIENTIST](#) clicks on the "DENY" button
- end if

*Details*

<b>Preconditions</b>	The test report has been generated
<b>Post-conditions</b>	the test result is approved or not

Human Task: Configure Development System

*Mock-up*

## Configuration Parameters

Insert Overfitting Tolerance:	0.2
Insert Min # of Layers:	12
Insert Max # of Layers:	32
Insert Variation step Layers:	4
Insert Min # of Neurons:	4
Insert Max # of Neurons:	20
Insert Variation step Neurons:	4
Insert Generalization Tolerance:	0.2

**Apply**

### Scenario

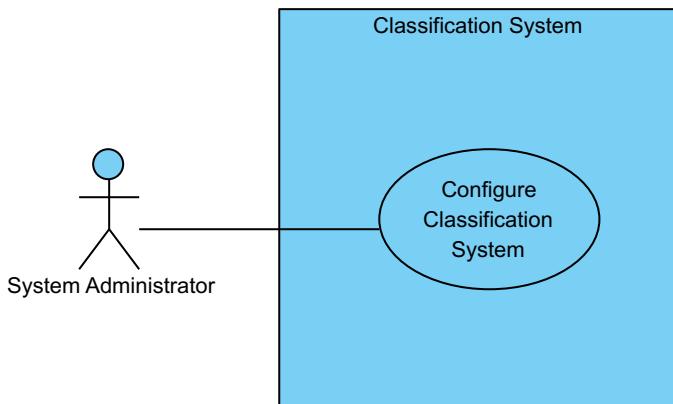
1.The use case starts when the  Data Scientist opens the interface to configure the development system.
2.SYSTEM shows Development System configuration parameter interface.
3.The  Data Scientist inserts from the keyboard the Overfitting tolerance in the <Insert Overfitting tolerance> field.
4.The  Data Scientist digits the Min # of Layers in the <Insert Min # of Layers> field.
5.The  Data Scientist digits the Max # of Layers in the <Insert Max # of Layers> field.
6.The  Data Scientist digits the Variation step Layers in the “Insert Variation step Layers” field.
7.The  Data Scientist digits the Min # of Neurons in the “Insert Min # of Neurons” field.
8.The  Data Scientist digits the Max # of Neurons in the “Insert Max # of Neurons” field.
9.The  Data Scientist digits the Variation step Neurons in the “Insert Variation step Neurons” field.
10.The  Data Scientist digits the Generalization tolerance in the “Select Generalization tolerance” field.
11.The  Data Scientist presses the "Apply" button.

### Details

Name	Value
Post-conditions	The configuration of the development system has been updated.

## 6. CLASSIFICATION SYSTEM(Alessio)

Use-case Diagram



Human Task: Configure Classification System

Mock-up

# Configure Classification System

IP address Preparation system: 192.168.0.14

IP address Evaluation system: 192.168.0.3

IP address End-client system: 131.114.3.6

**Save configuration**

Scenario

1. Use case starts when the  <b>System Administrator</b> opens the <b>Configure Classification System</b> interface
2. <b>SYSTEM</b> Displays the interface with the IP addresses to be set
3. The  <b>System Administrator</b> sets the 'IP address Preparation system' field
4. The  <b>System Administrator</b> sets the 'IP address Evaluation system' field
5. The  <b>System Administrator</b> sets the 'IP address End-Client system' field

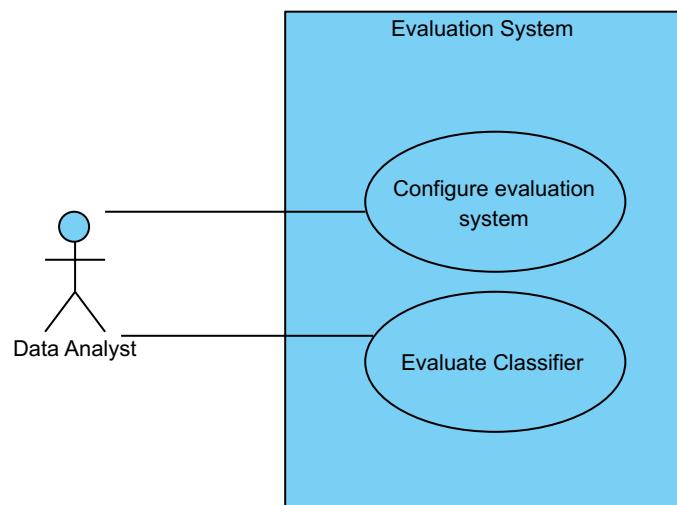
6. The  **System Administrator** saves the configuration by pressing the **Save configuration** button

*Details*

Name	Value
Post-conditions	The phase of the classification system is set The connected systems' IP addresses are set

## 7. EVALUATION SYSTEM (Alessio)

Use-case Diagram



## Human Tasks: Configure Evaluation System

Mock-up

# Configure Evaluation System

Number of labels:  IP address Messaging system

Maximum number of errors:  IP address Ingestion system

Maximum number of consecutive errors:  IP address Classification system

**Save configuration**

## Scenario

1. Use case starts when the  Data Analyst opens the **Configure Evaluation System** interface
2. **SYSTEM** Displays all the parameters to be set and a writable textbox for each one
3. The  Data Analyst sets the 'Number of labels' parameter
4. The  Data Analyst sets the 'Maximum number of errors' parameter
5. The  Data Analyst sets the 'Maximum number of consecutive errors' parameter
6. The  Data Analyst sets the 'IP address Messaging system' field
7. The  Data Analyst sets the 'IP address Ingestion system' field
8. The  Data Analyst sets the 'IP address Classification system' field
9. The  Data Analyst saves the configuration by pressing the **Save configuration** button

## Details

Name	Value
Post-conditions	The Evaluation System is configured

	The connected systems' IP addresses are set
--	---

Human Task: Evaluate Classifier

Mock-up

# Evaluation report

**Report**

Classifier	Expert	Control
none	none	✓
overheating	overheating	✓
none	none	✓
overheating	none	✗
none	electricalfault	✗
electricalfault	electricalfault	✓
...	...	(Scroll) ▾

**Thresholds**

Maximum number of errors:	20
Maximum number of consecutive errors:	1

**Results**

Number of errors:	12
Consecutive errors:	2

**Evaluation**

GOOD
NOT GOOD

**Check**

Threshold 1:	
Threshold 2:	

Scenario

1. Use case starts when the  Data Analyst opens the <b>Evaluate Classifier</b> interface
2. <b>SYSTEM</b> Displays the interface with all the relative information
3. <b>if</b> The  Data Analyst sees that the threshold 1 ( <i>Number of errors</i> ) is respected:
3.1. <b>if</b> The  Data Analyst sees that the threshold 2 ( <i>Consecutive errors</i> ) is respected:
3.1.1. The  Data Analyst selects the <b>GOOD</b> button
3.2. <b>else</b>
3.2.1. The  Data Analyst selects the <b>NOT GOOD</b> button

end if
4. else
4.1. The  Data Analyst selects the NOT GOOD button
end if

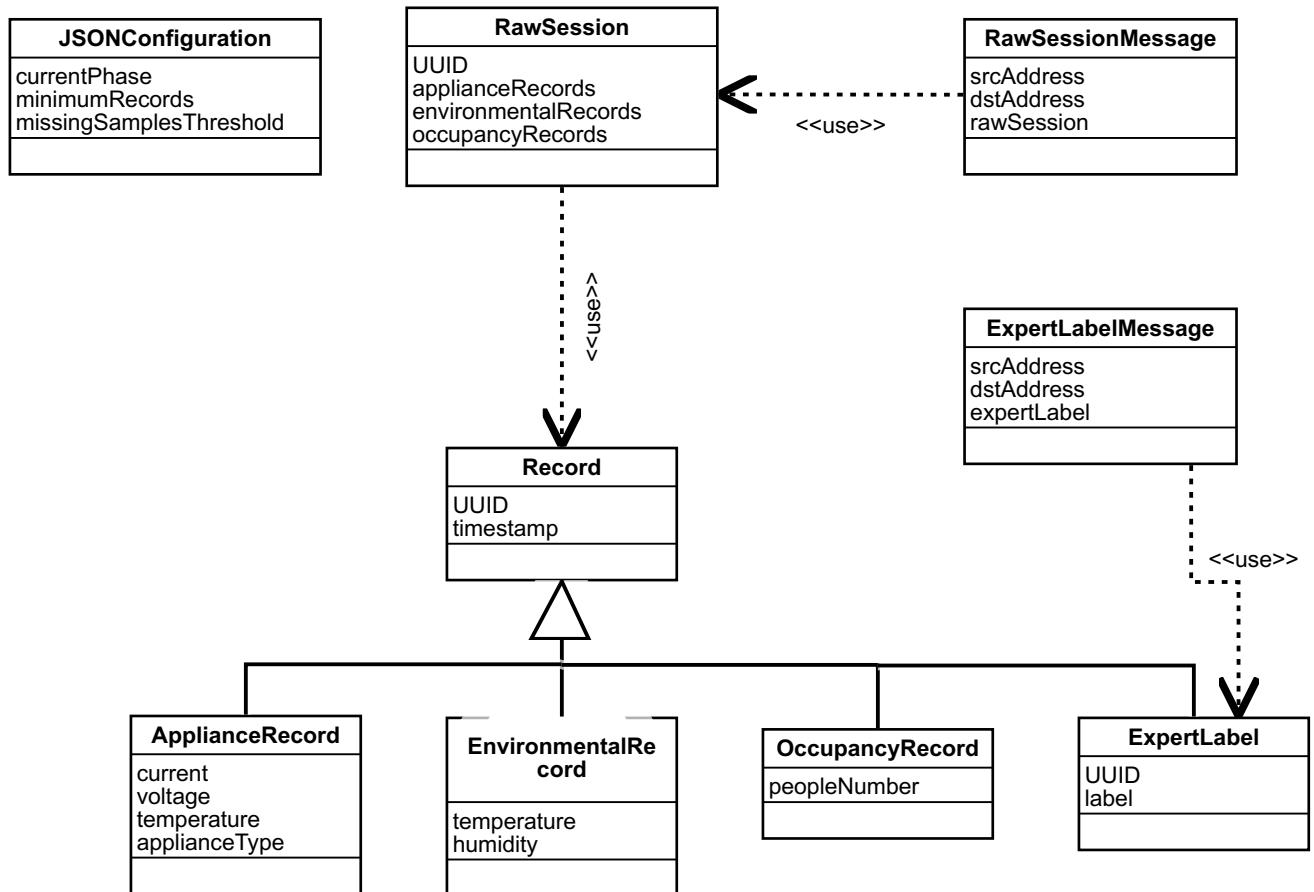
*Details*

Name	Value
Preconditions	A sufficient number of labels (specified in the configuration) has been collected
Post-conditions	The classifier is evaluated

# Workflow ANALYSIS: Data and Application Logic Models

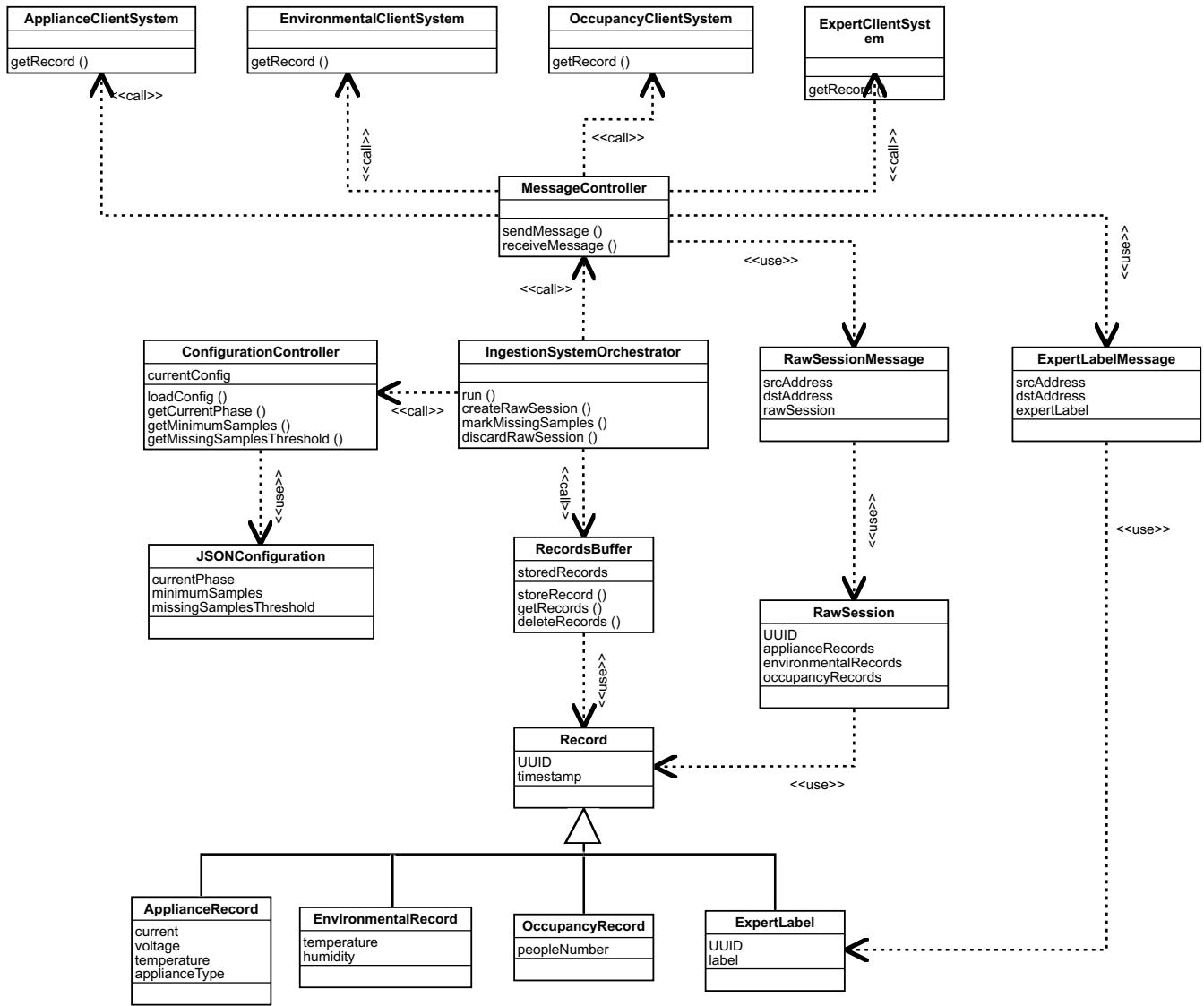
## 1. INGESTION SYSTEM (Carlo)

Data model

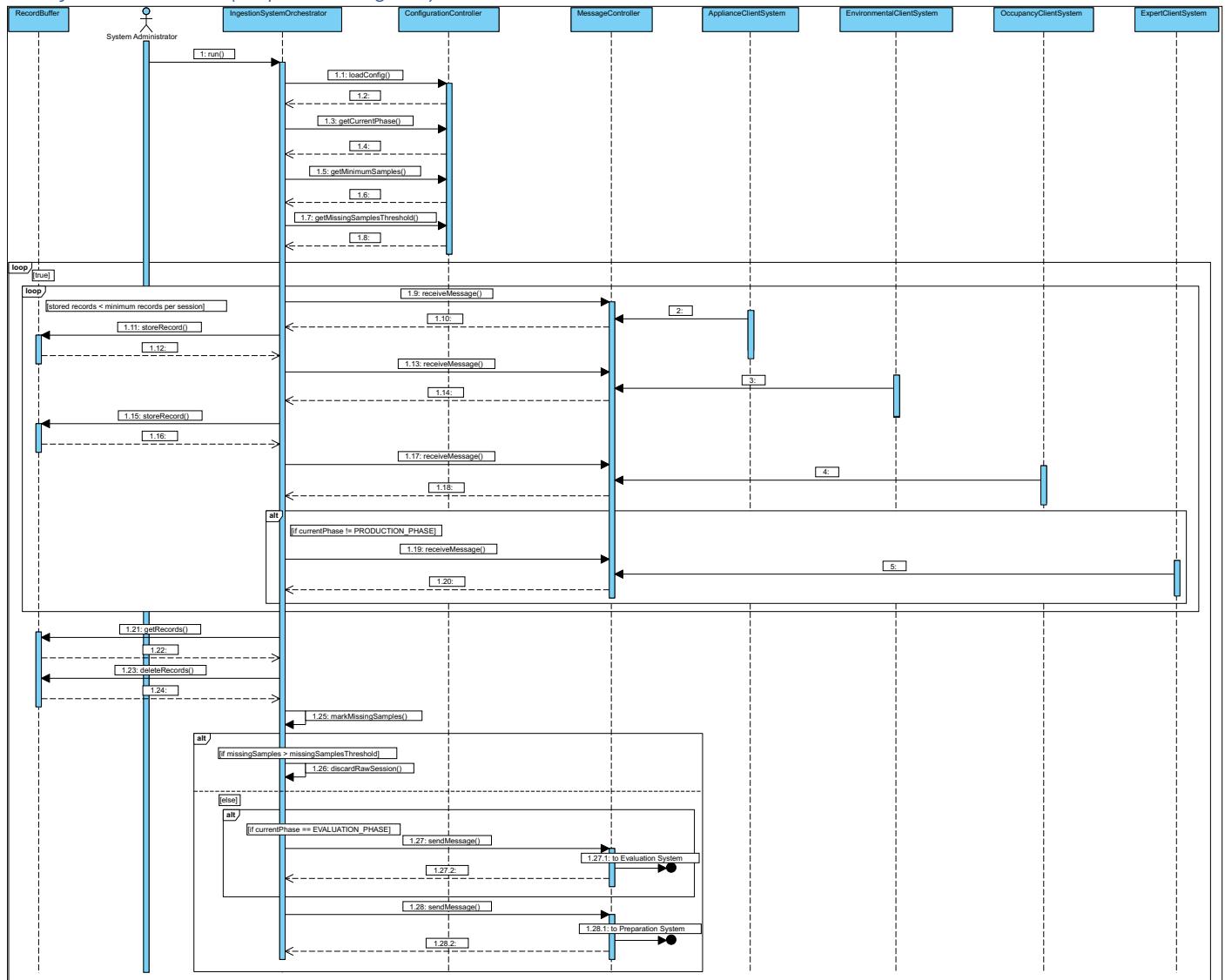


## Application logic model

### UML Class Diagram

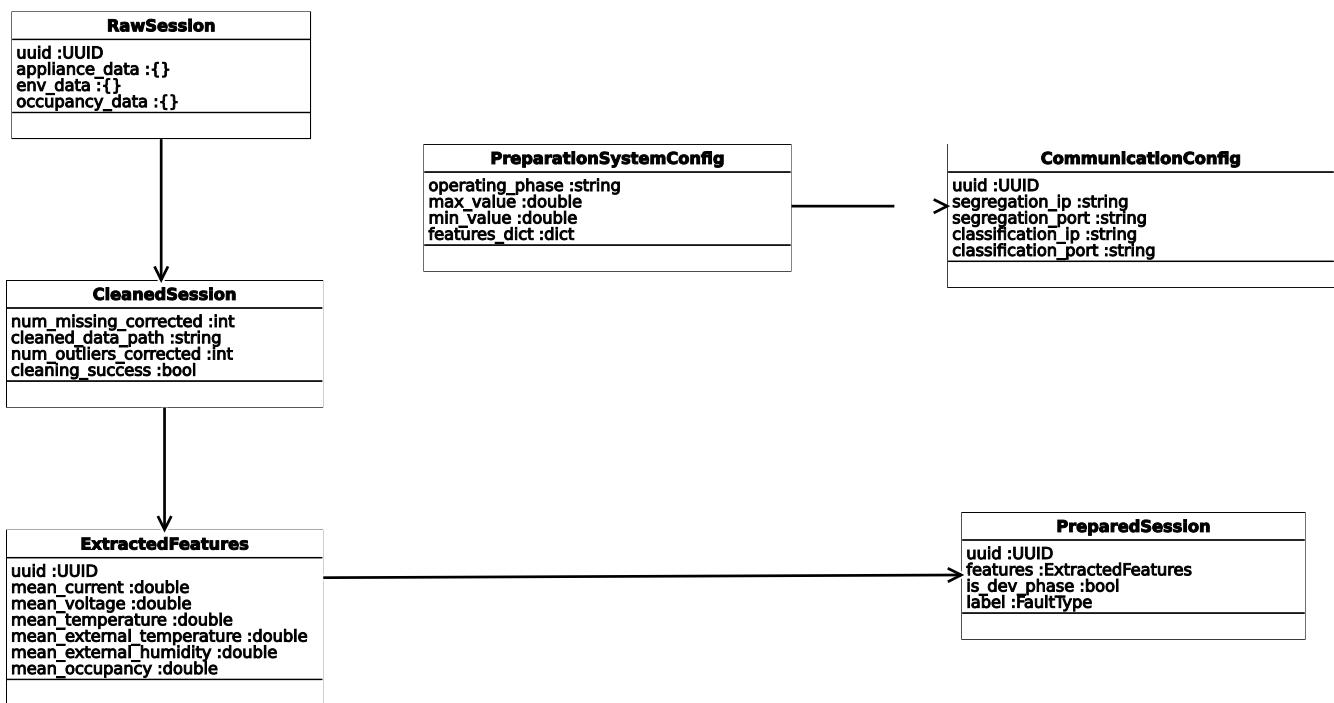


### Workflow Realization (Sequence Diagram)



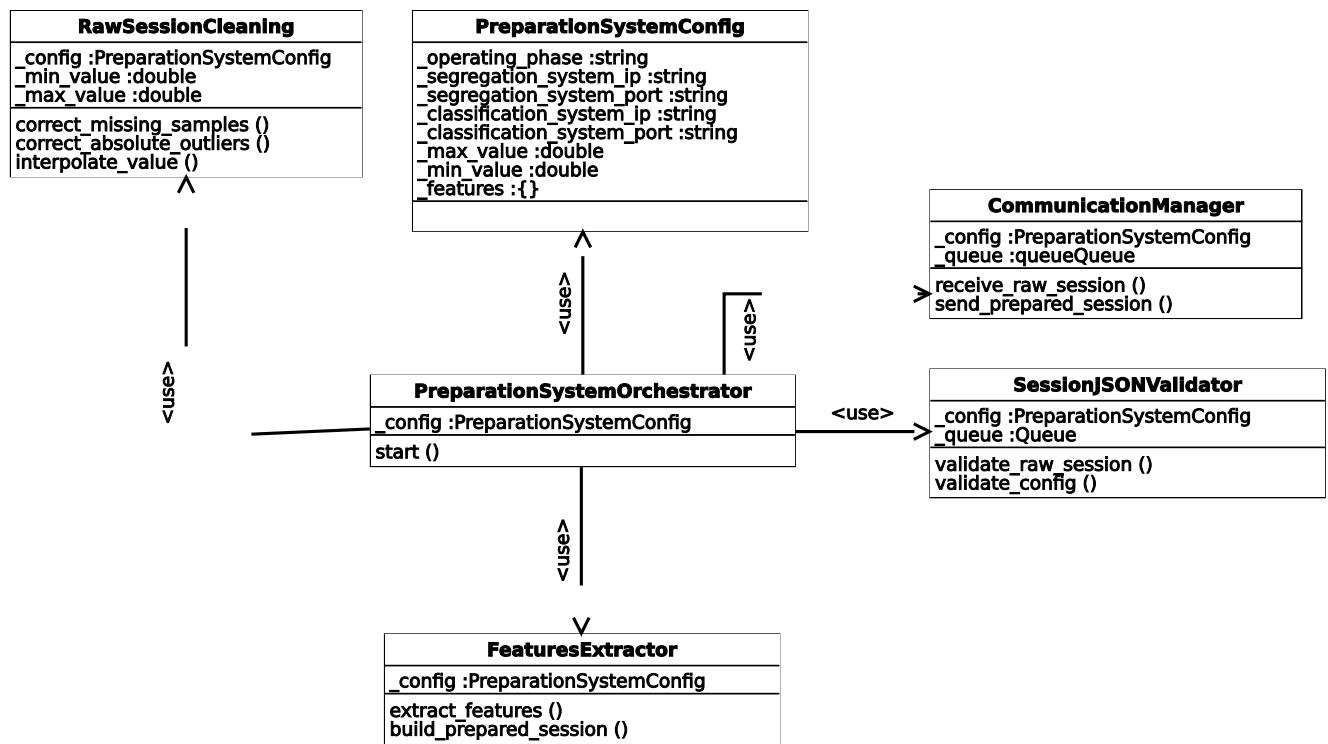
## 2. PREPARATION SYSTEM(Tesfaye)

Data Model

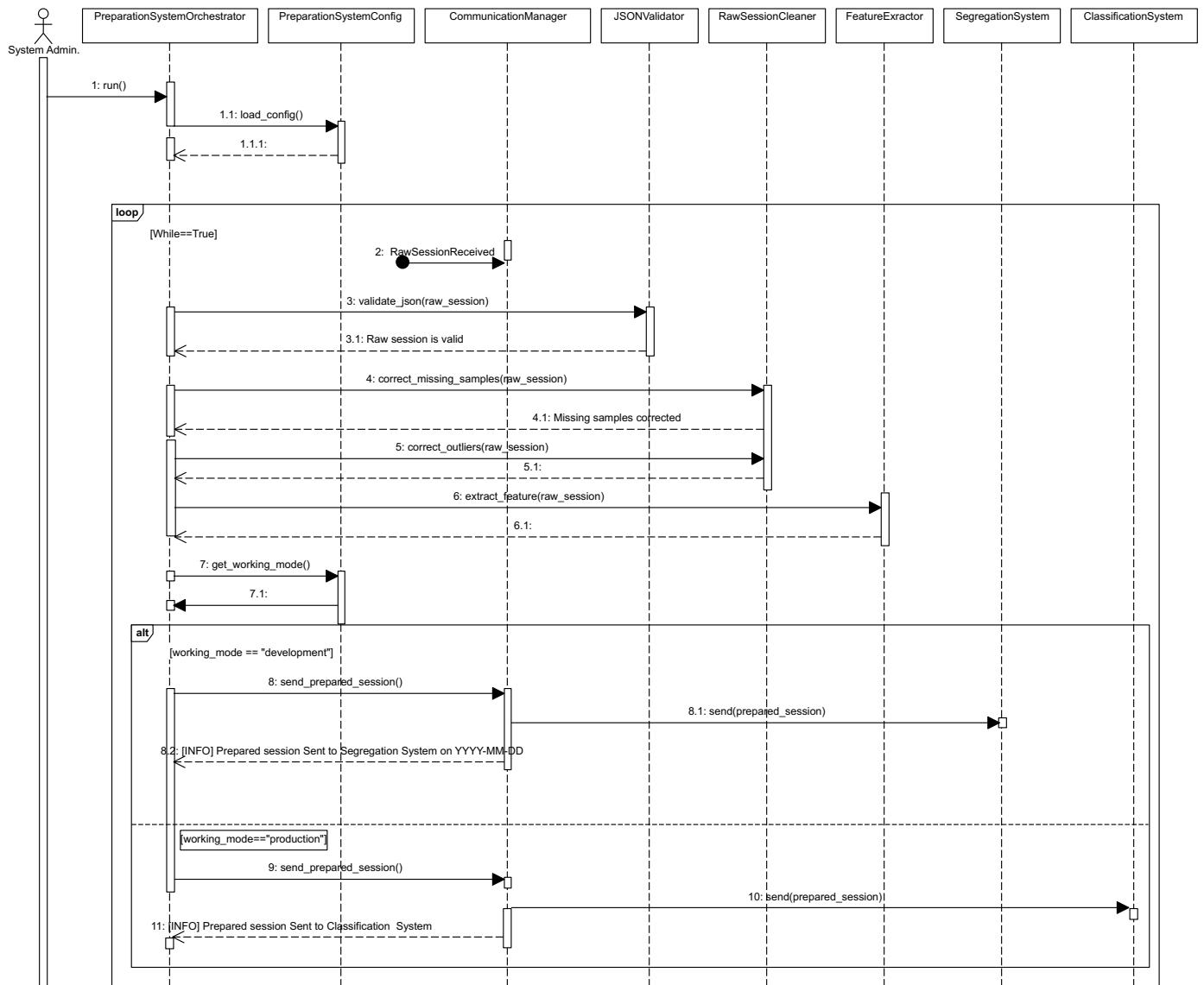


## Application Logic Model

### UML Class Diagram

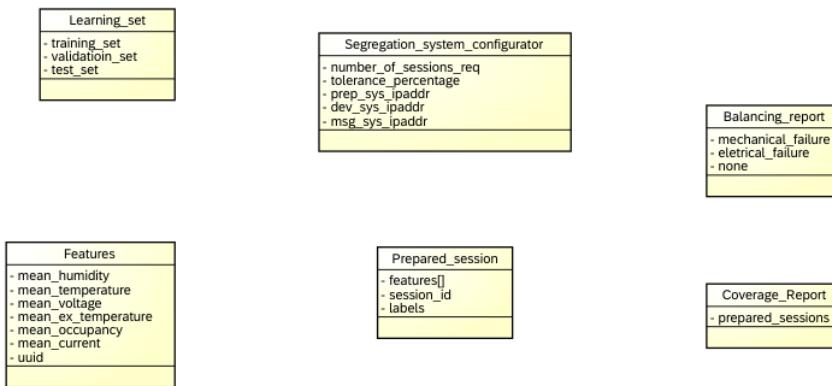


## Workflow Realization (Sequence Diagram)



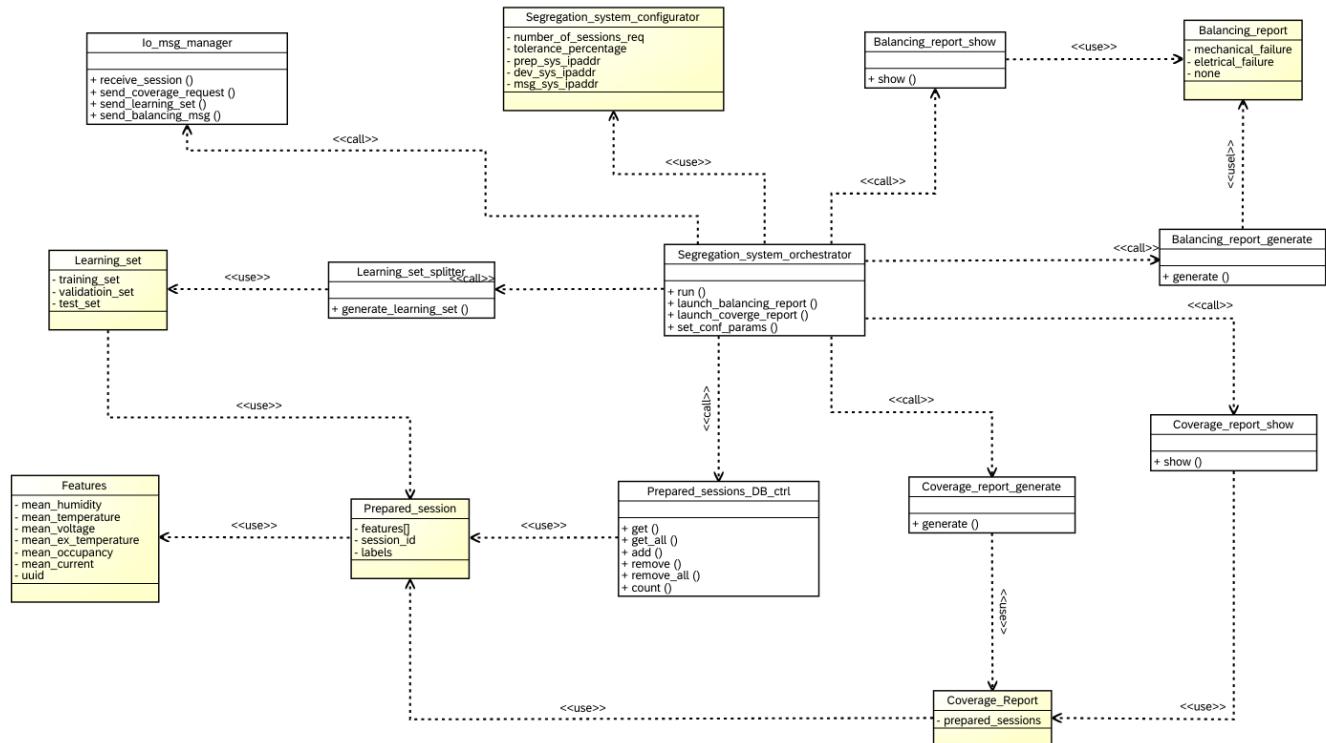
### 3. SEGREGATION SYSTEM (Lorenzo V)

#### Data Model

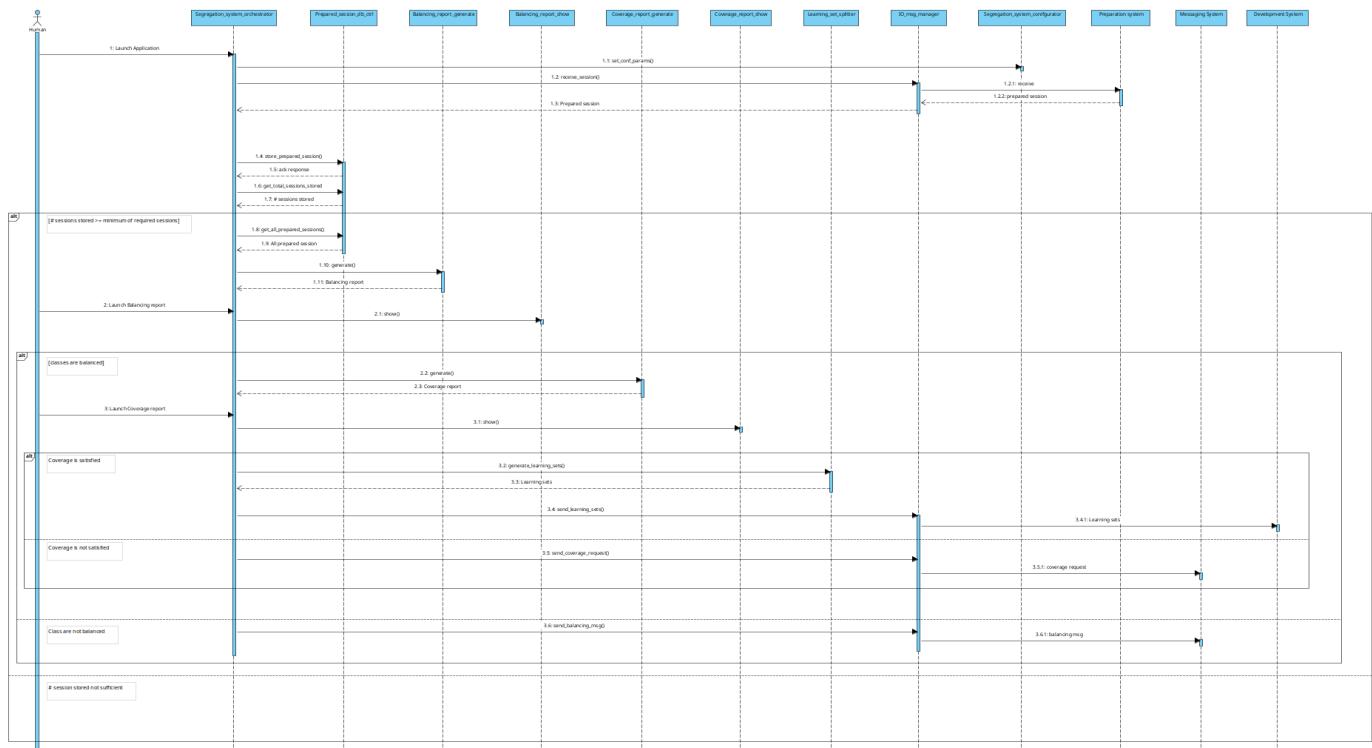


#### Application Logic Model

*UML class diagram*

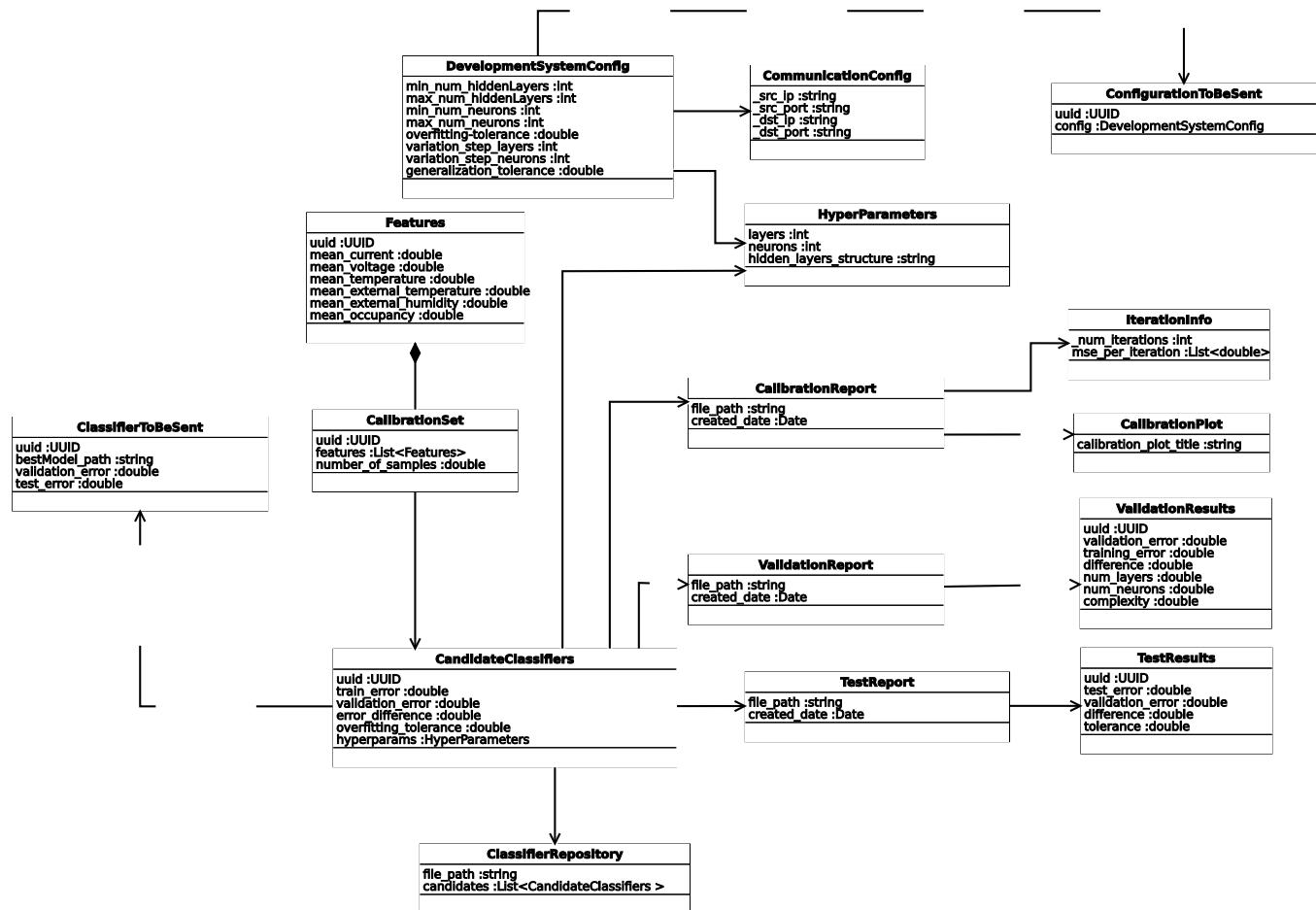


## Workflow Realization (Sequence Diagram)



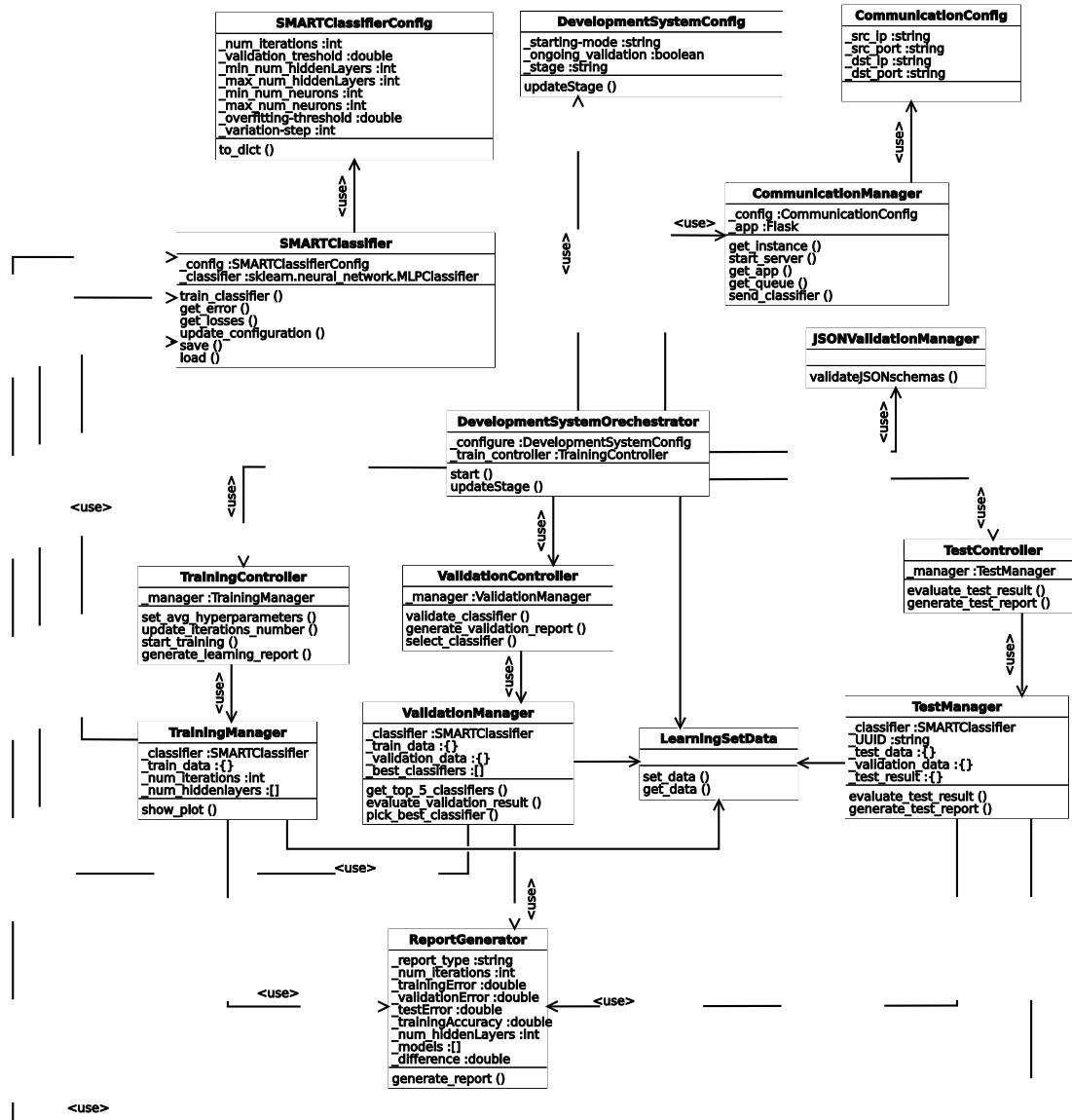
## 4. DEVELOPMENT SYSTEM(Tesfaye)

### Data Model

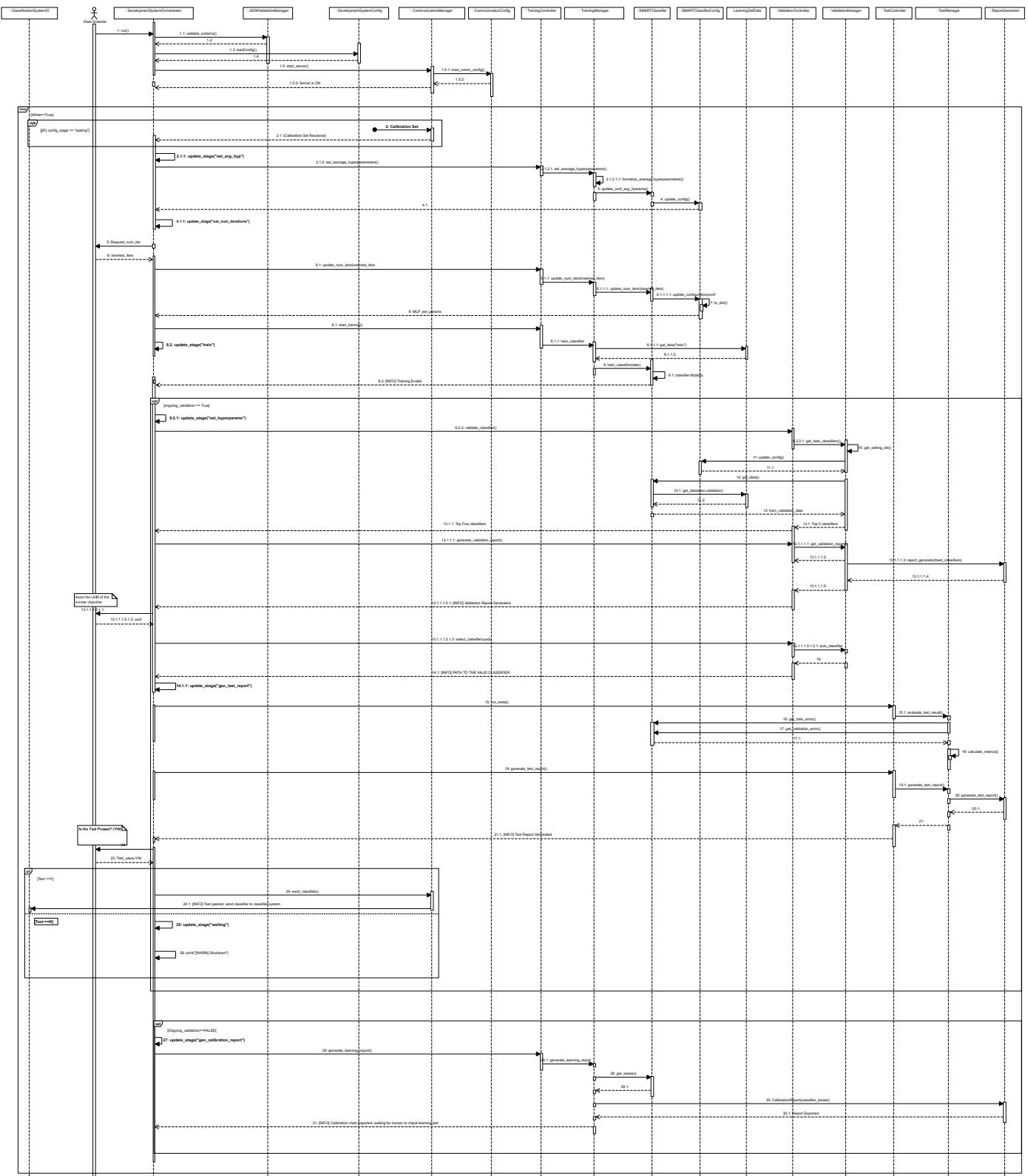


# Application Logic Model

## UML Class Diagram

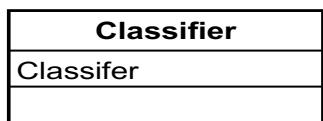
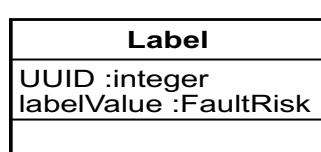
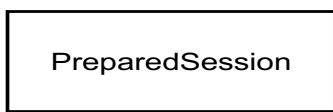
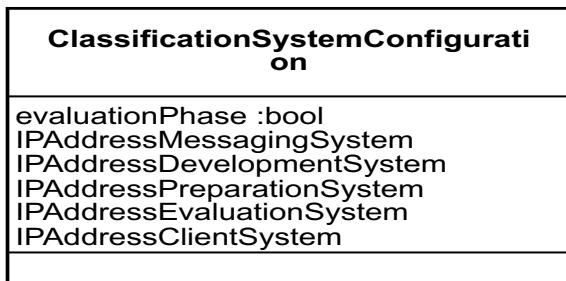


## *Workflow Realization (Sequence Diagram)*



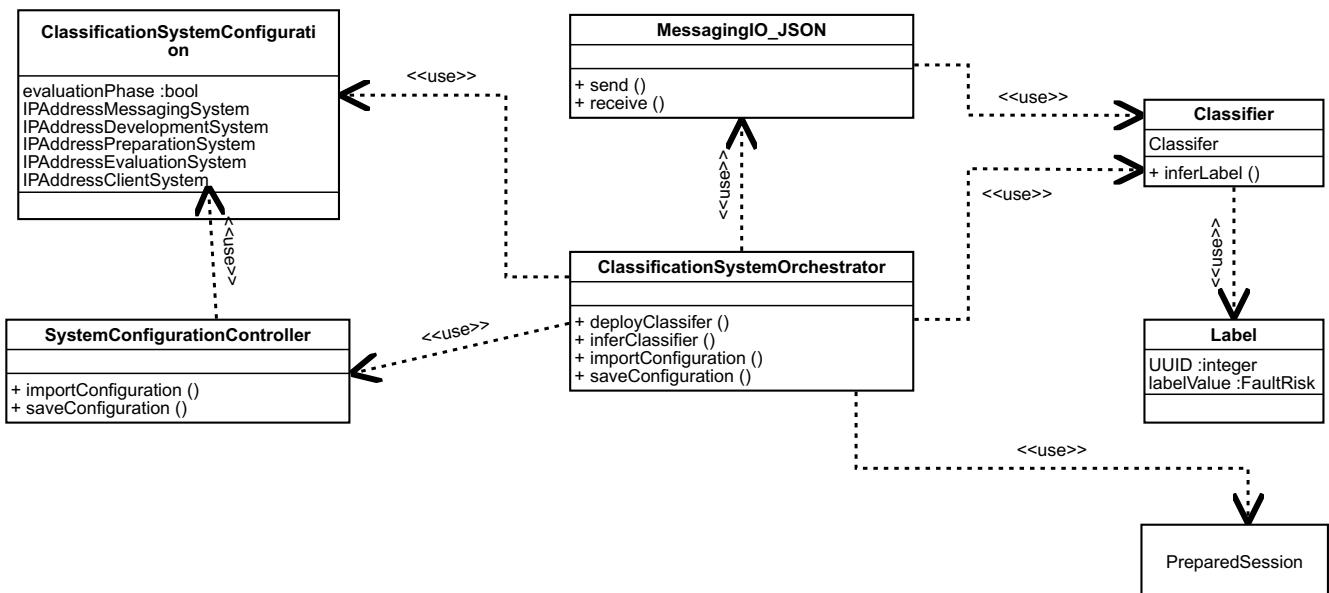
## 5. CLASSIFICATION SYSTEM (Alessio)

Data model

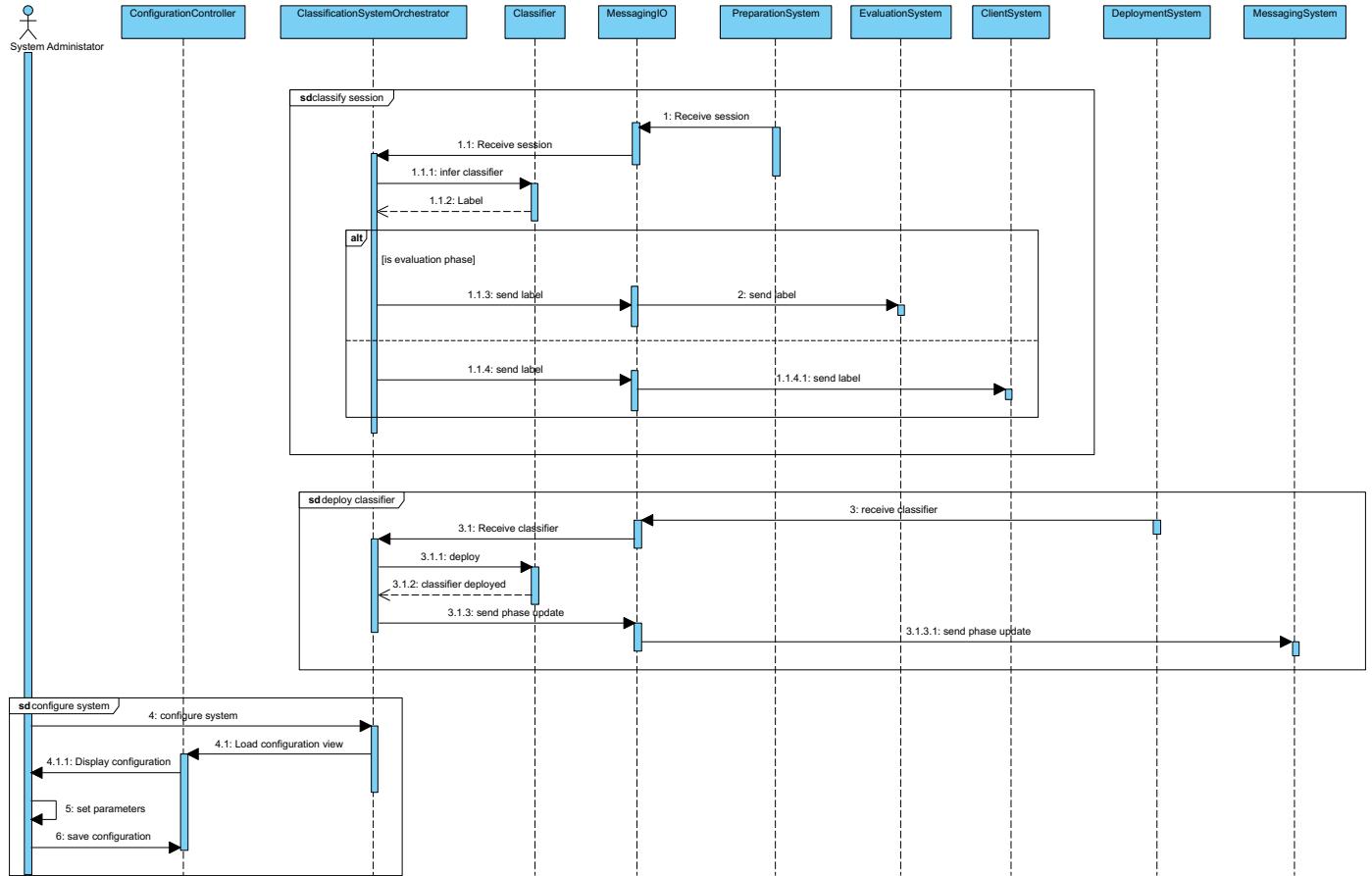


Application logic model

*UML Class Diagram*

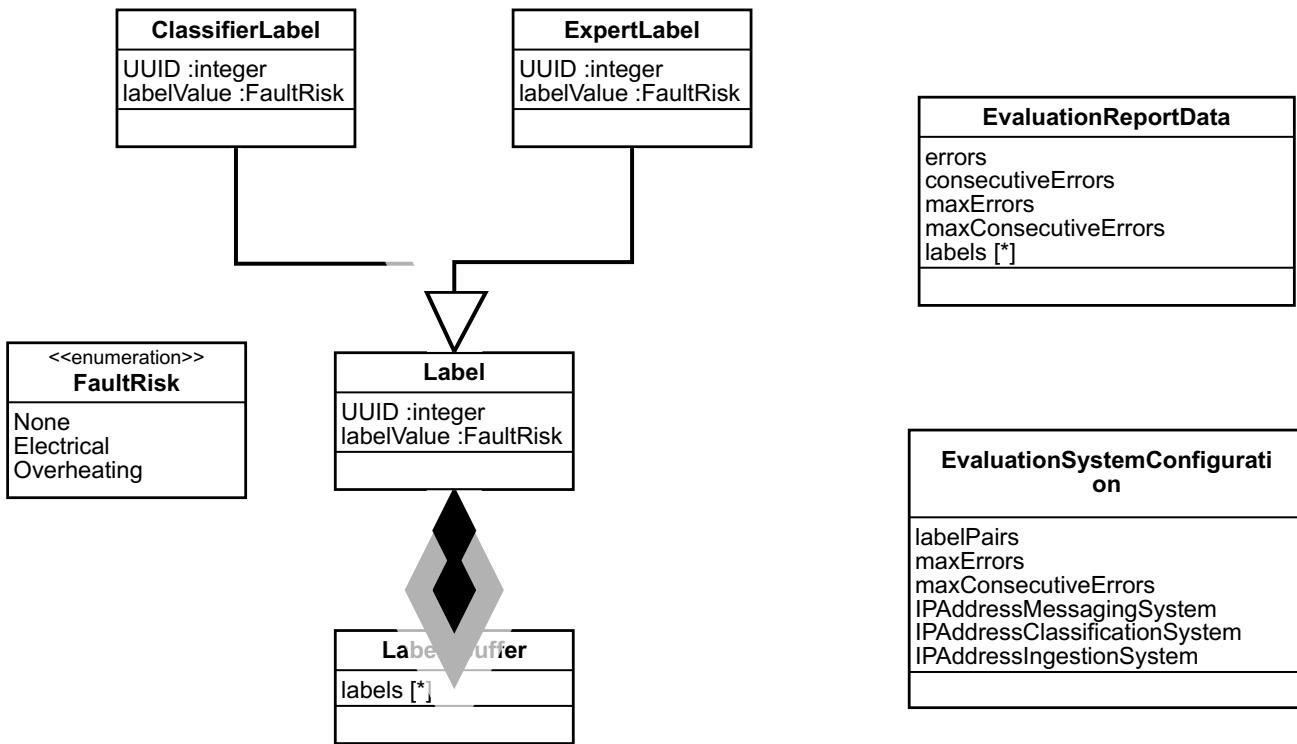


### Workflow Realization (Sequence Diagram)



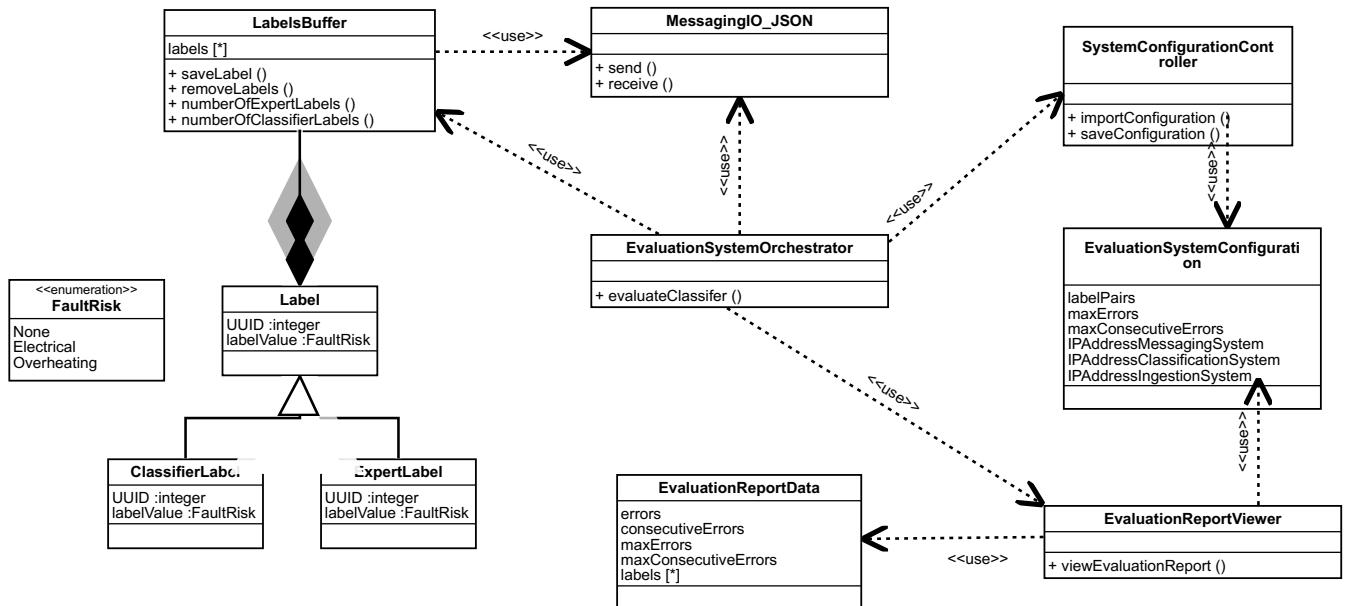
## 6. EVALUATION SYSTEM (Alessio)

Data model

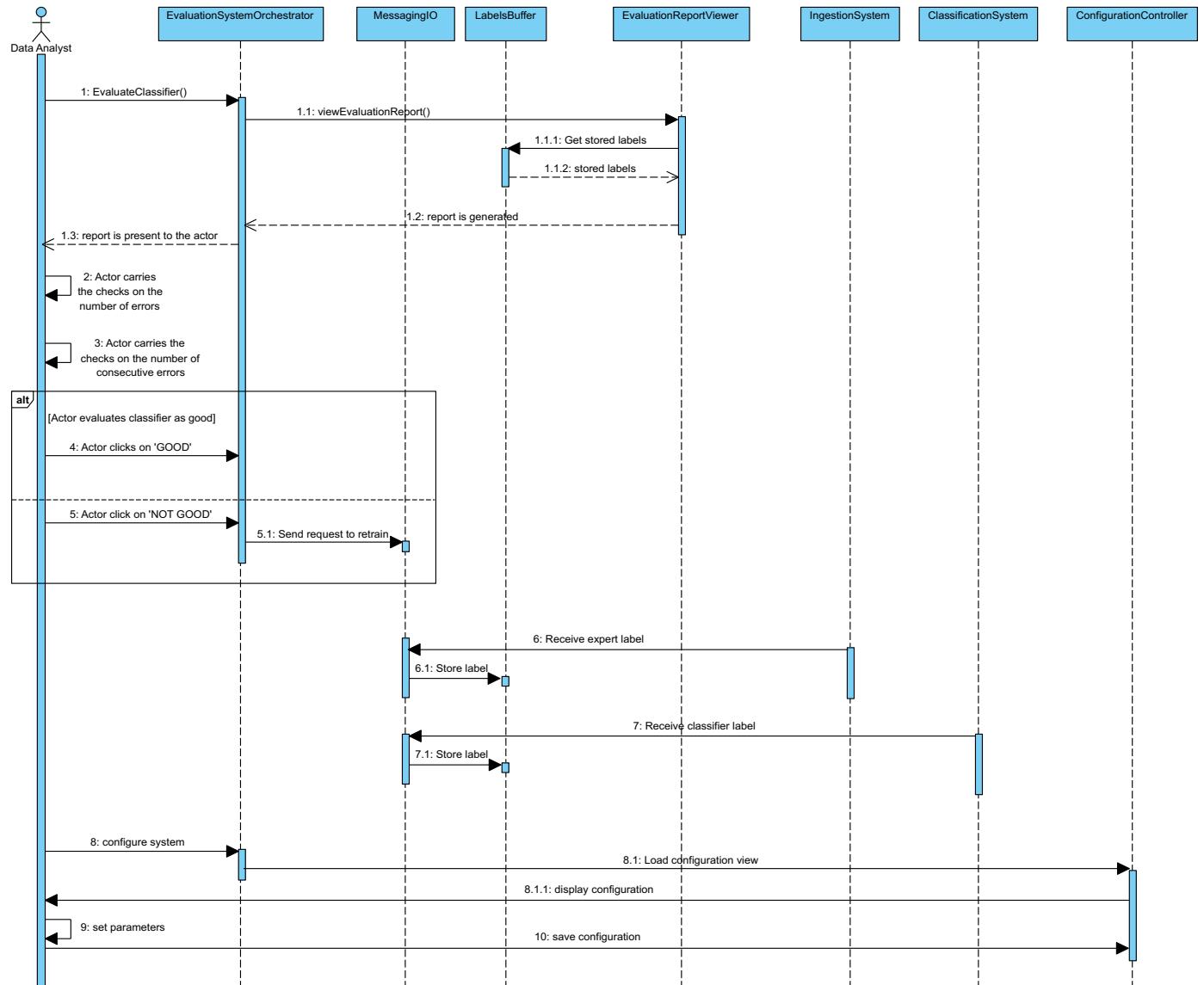


Application logic model

*UML Class Diagram*



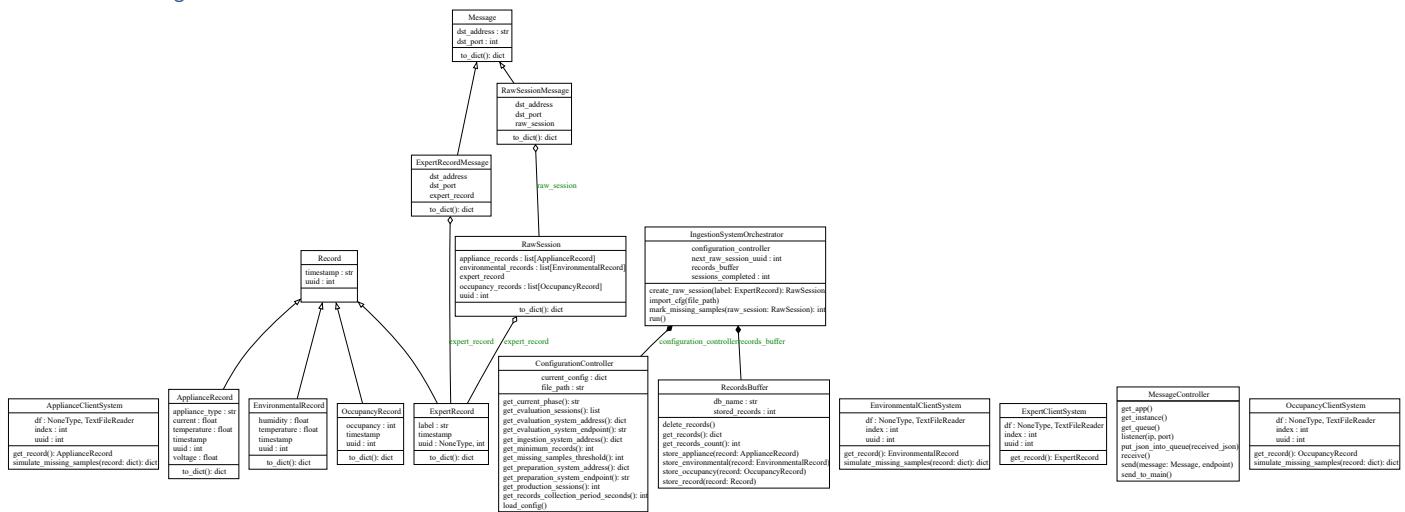
## Workflow Realization (Sequence Diagram)



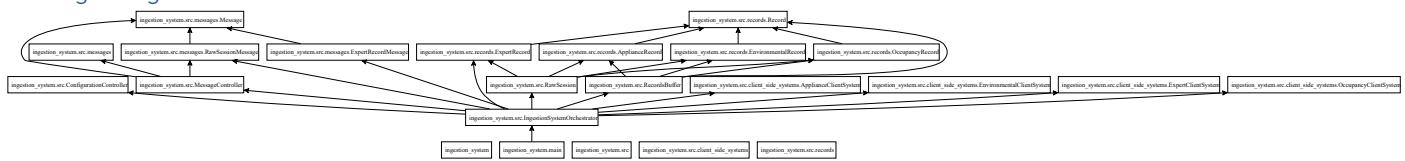
# Workflow Design: UML class and Sequence diagrams

## 1. PREPARE SESSION PROCESS (Carlo)

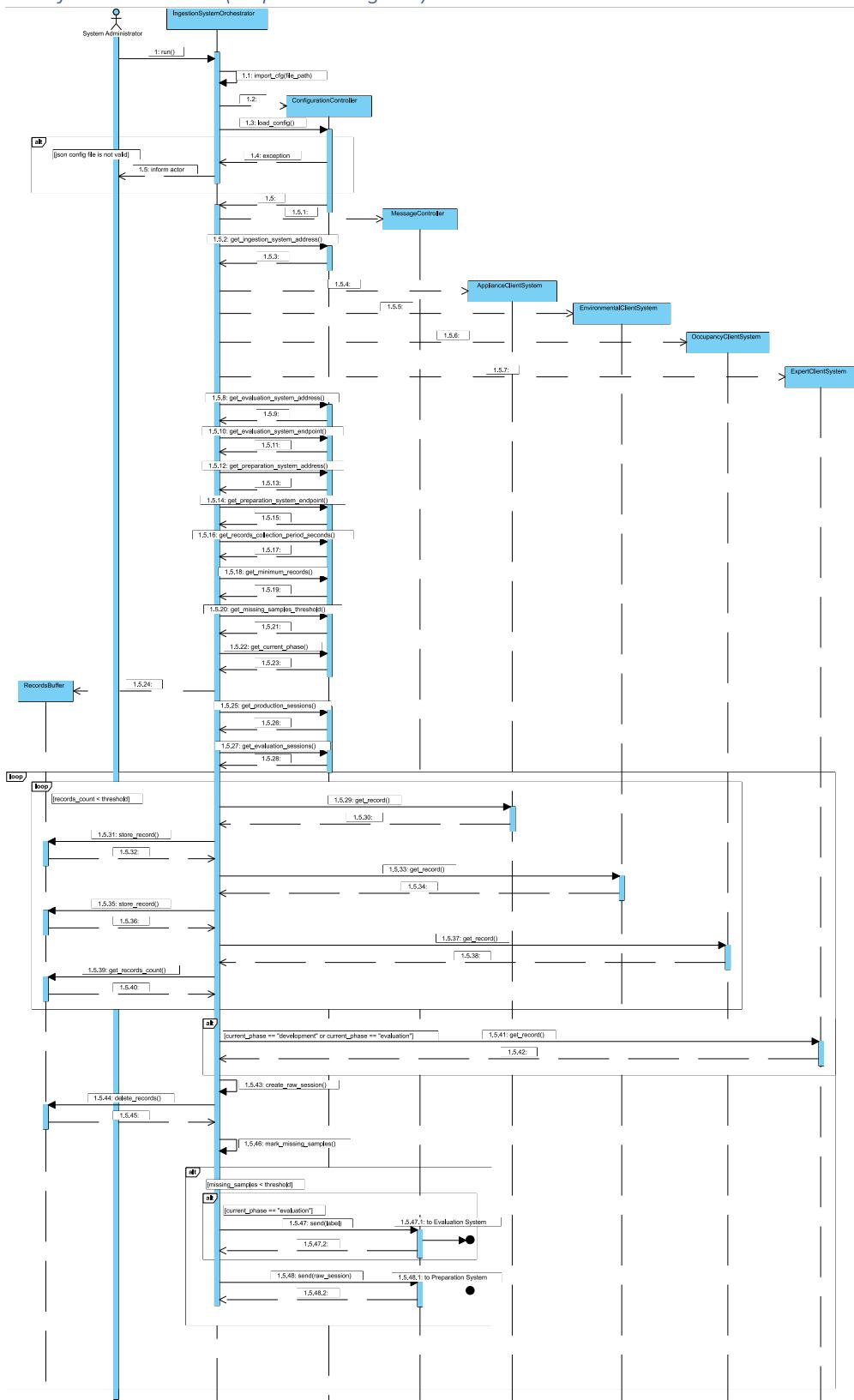
## *UML Class Diagram*



## *Package diagram*

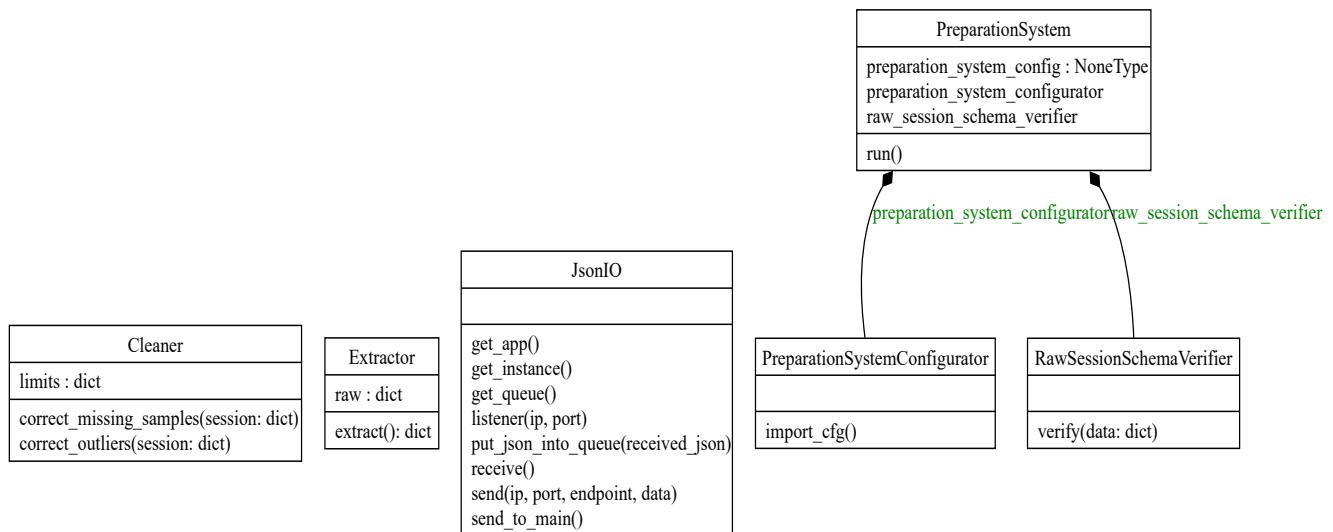


## *Workflow Realization (Sequence Diagram)*

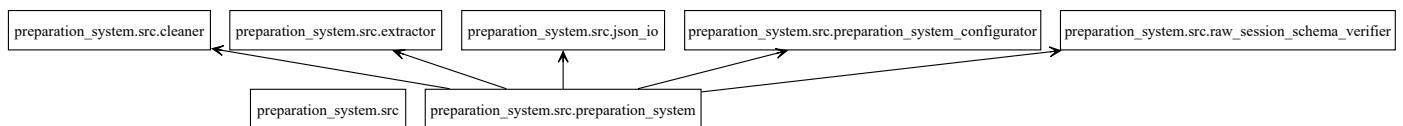


## 2. PREPARATION SYSTEM

*UML Class Diagram*

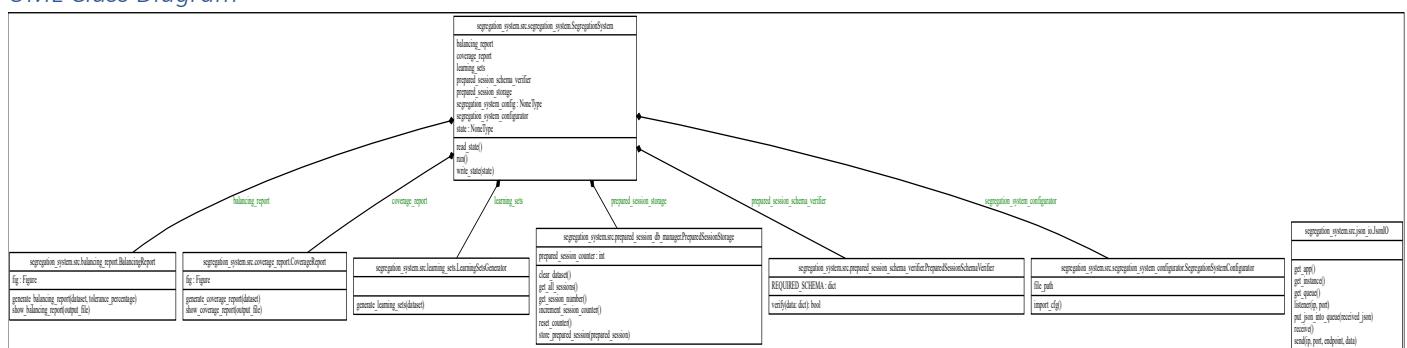


*Package Diagram*

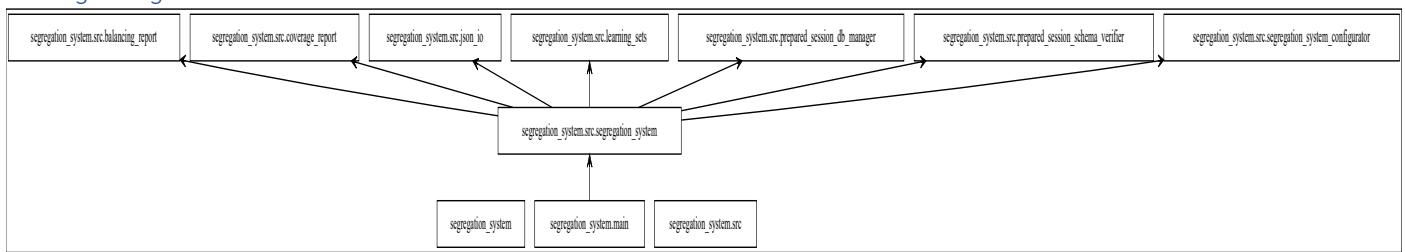


## 3. SEGREGATION SYSTEM

*UML Class Diagram*



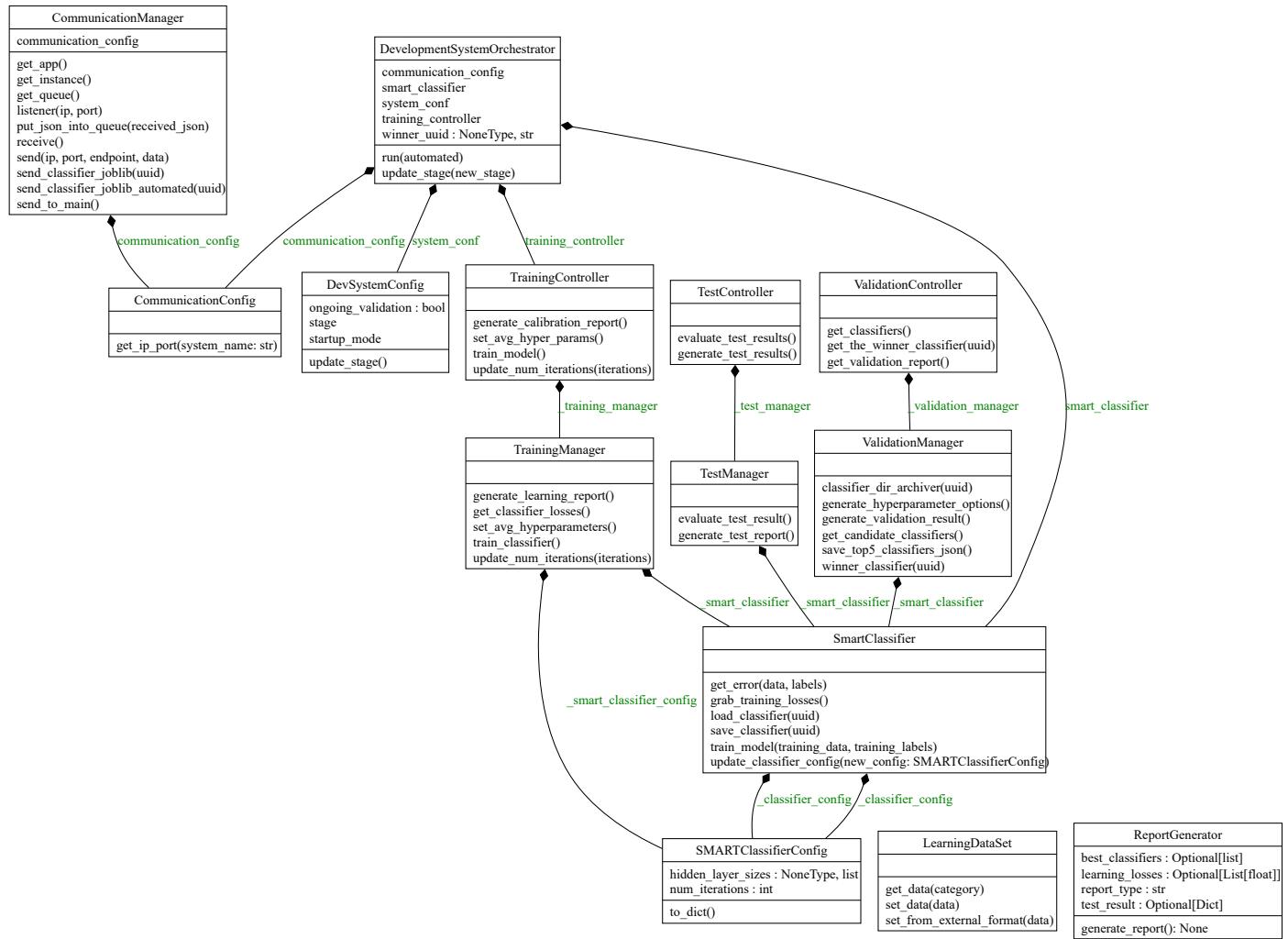
## Package Diagram



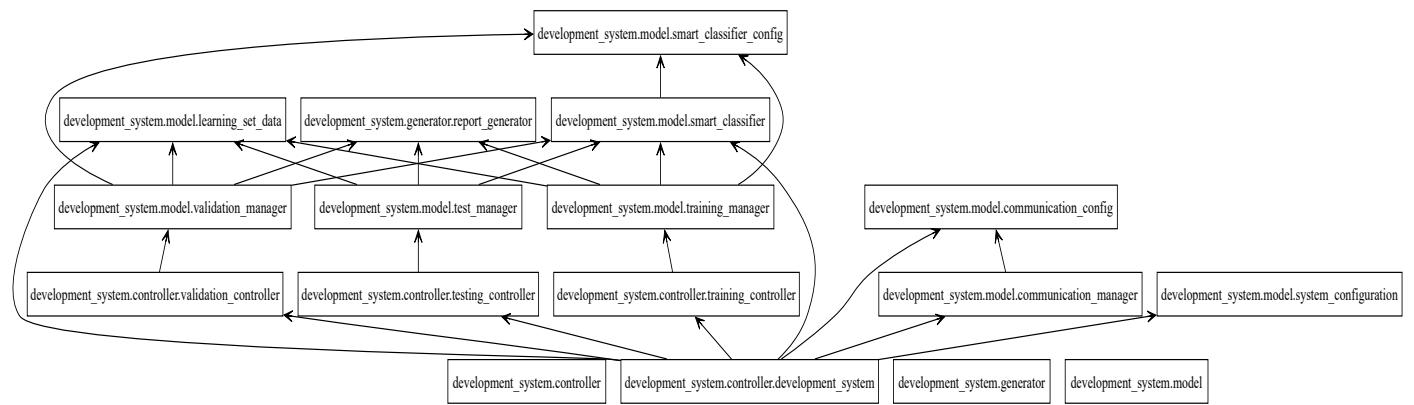
## 4. DEVELOPMENT SYSTEM (Tesfaye)

The UML class diagrams for the Development System were automatically derived using the Pyreverse module from the Pylint suite in Python.

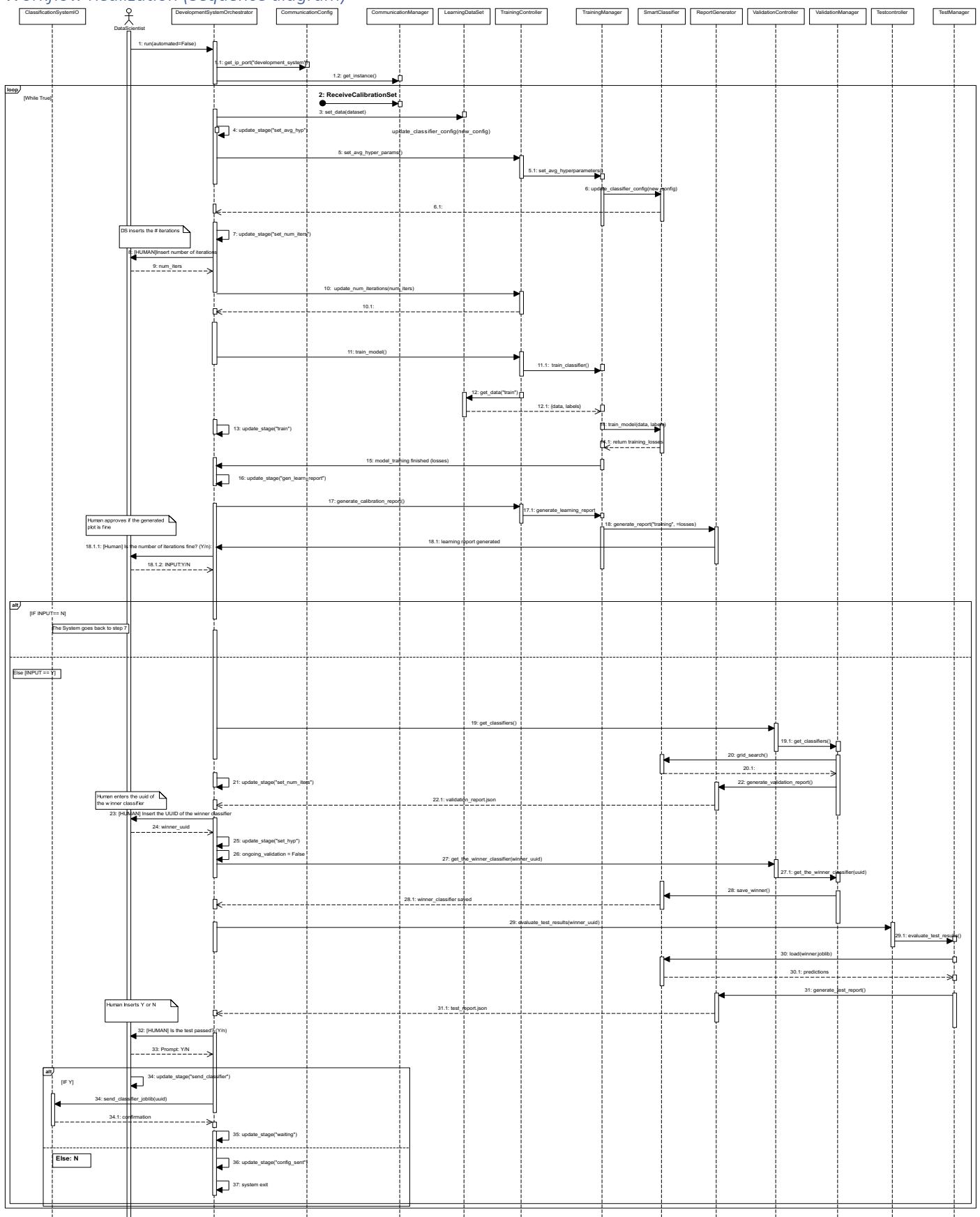
### UML Class Diagram



## Package Diagram

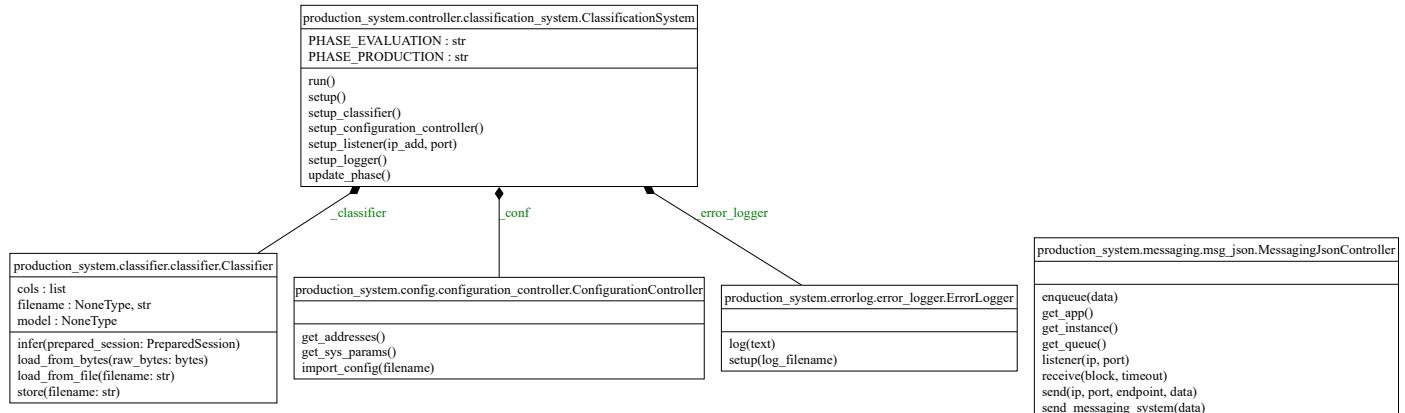


## Workflow Realization (Sequence diagram)

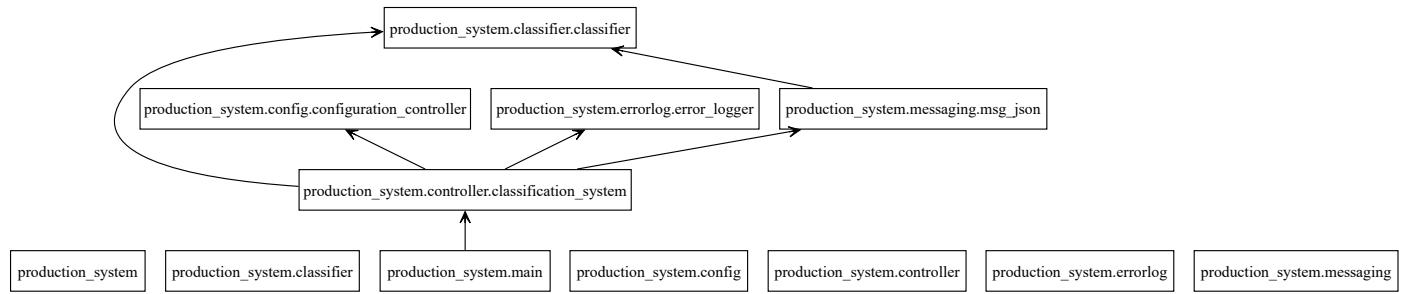


## 5. CLASSIFICATION SYSTEM (Alessio)

## *UML Class Diagram*

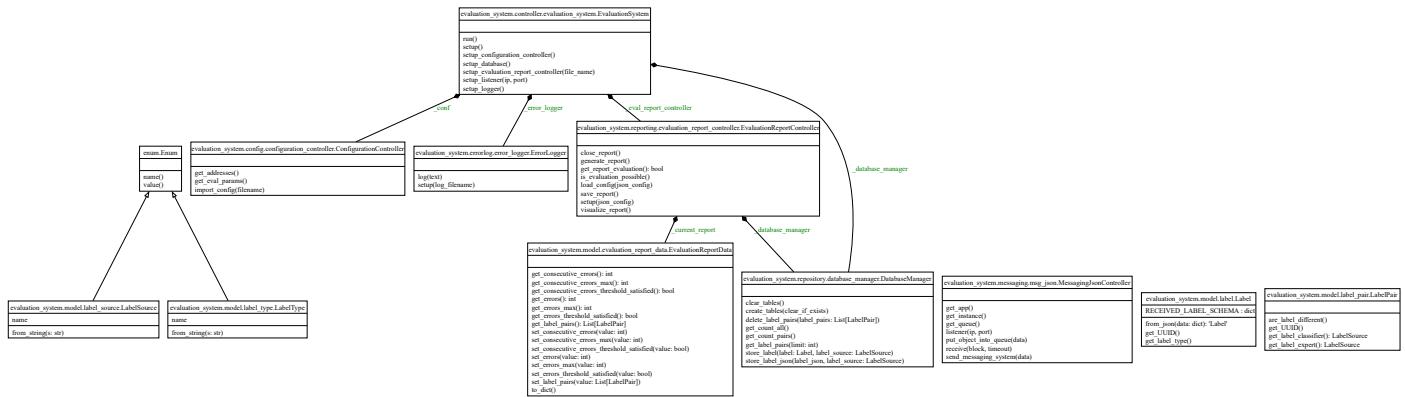


## Packages Diagram

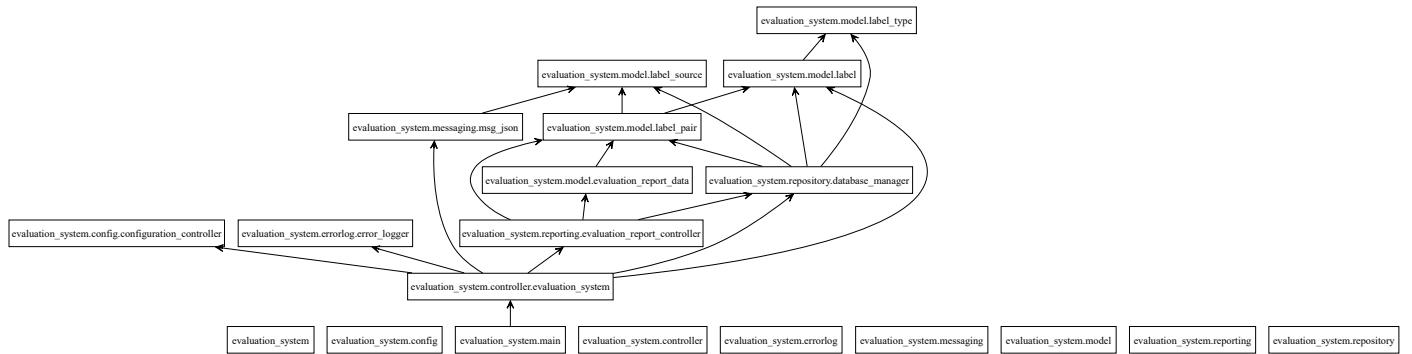


## 6. EVALUATION SYSTEM (Alessio)

## *UML Class Diagram*

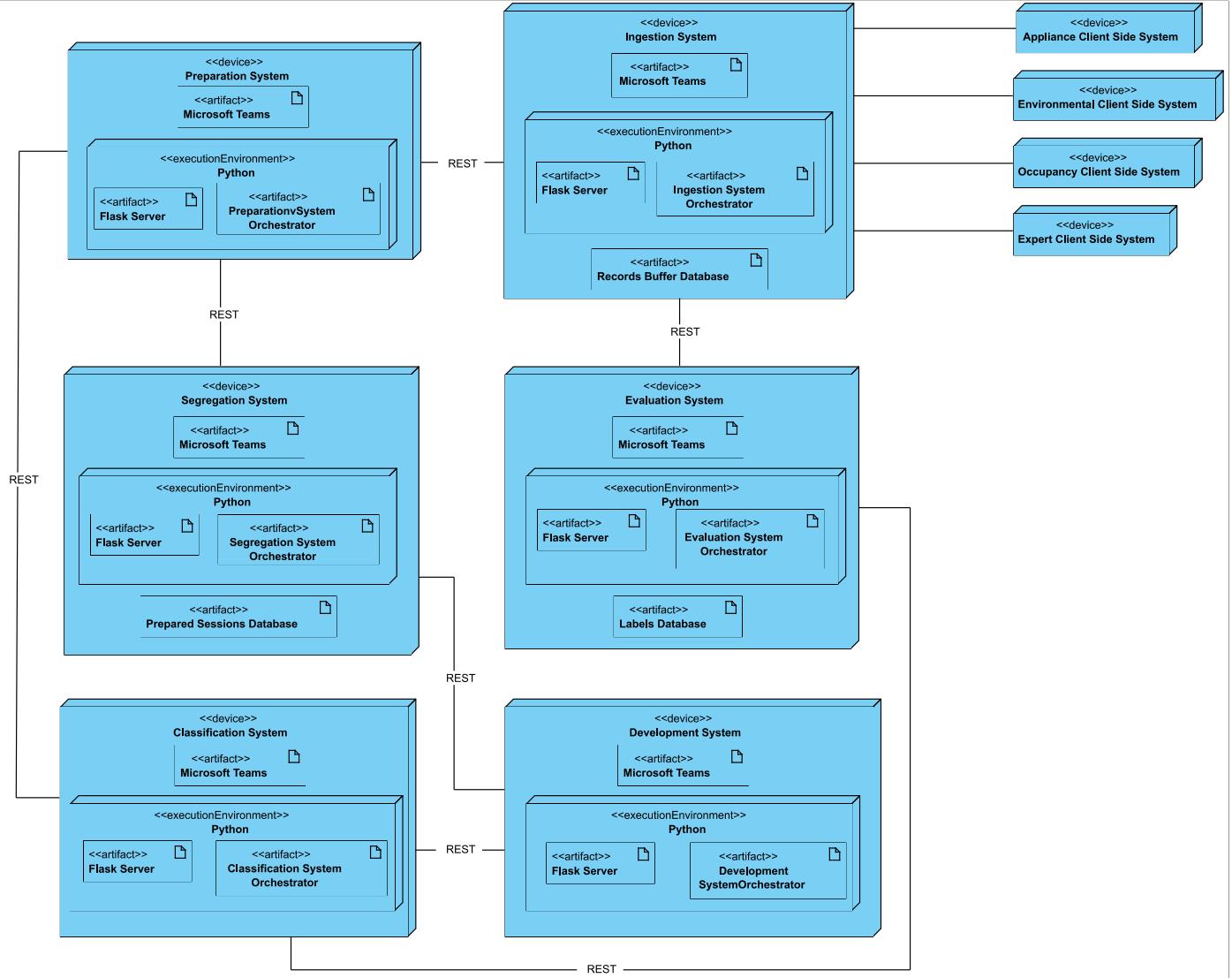


## Package Diagram



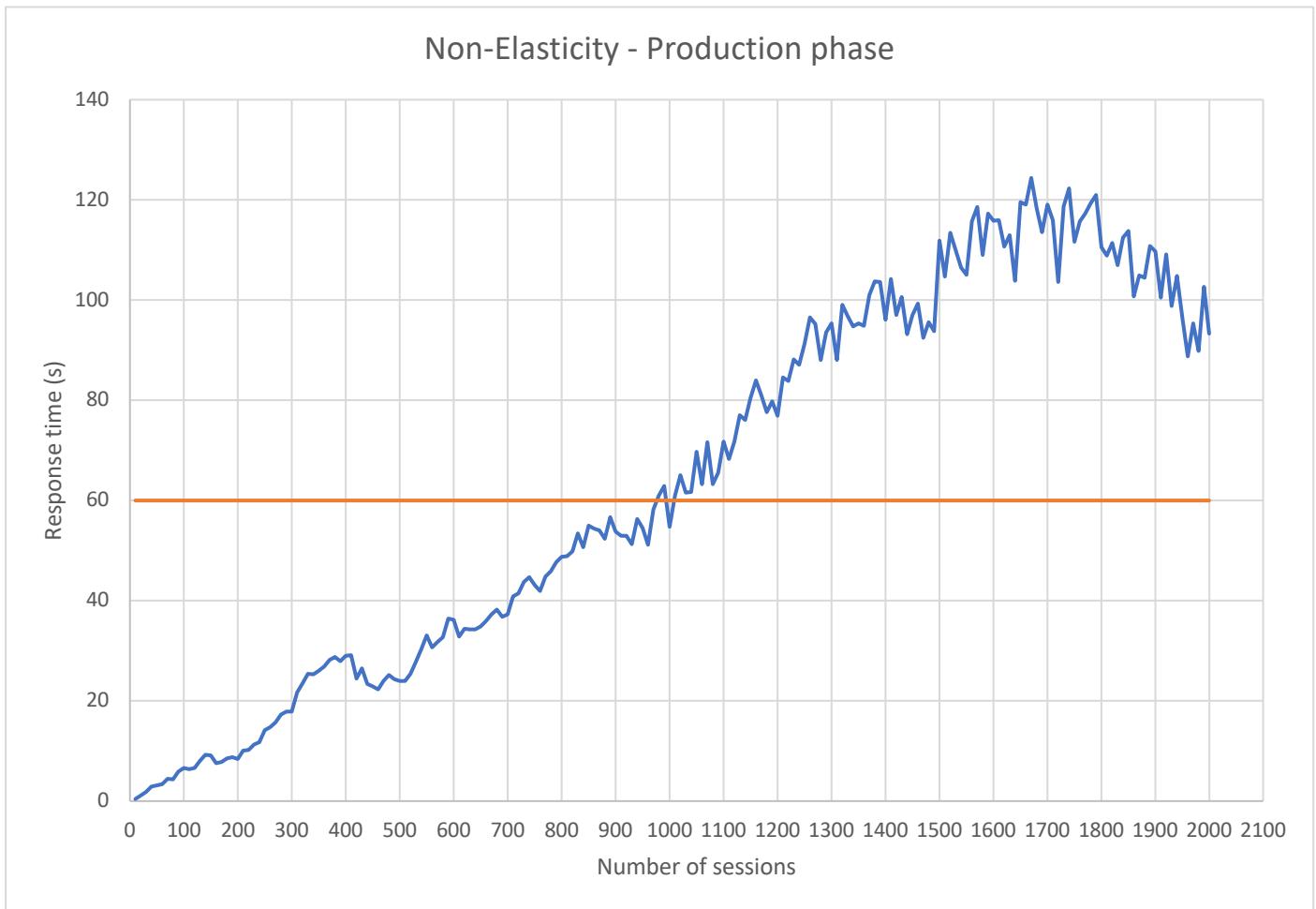
# Workflow Implementation:

## Deployment Diagram



# Workflow Testing:

## 1. RESPONSIVENESS-ELASTICITY of the production phase



After all systems startup, the response time for each session begins to gradually increase as the number of classifications requests arrive. This linear-like behaviour is because the queues of the preparation system and the classification system begin to fill and cache memories are empty, so many misses are generated.

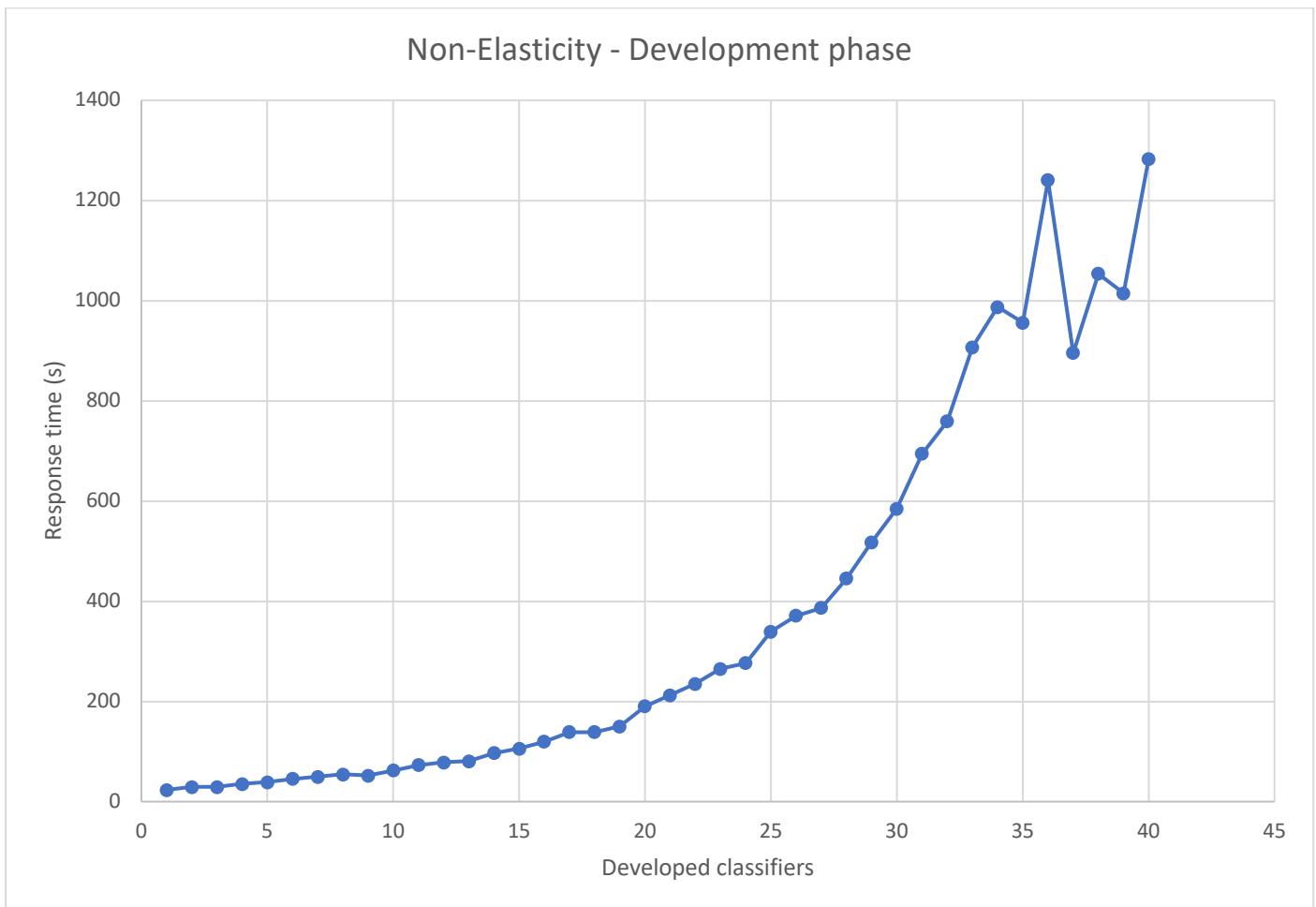
This queues buildup is caused by the important rate of arrival of records from the client-side systems and the performance bottleneck of classification system during inferences.

After a certain threshold, the caches of the systems will contain most of the useful data necessary for the computations, so the processing will be faster, hence the final stabilization of the response time.

In this phase the Ingestion system will be the bottleneck because it won't be able to increase the sending rate of the raw sessions.

For our purposes, we would like to provide a classification for each 60 seconds, so our responsiveness threshold lays at around 1000 sessions.

## 2. AUTOMATION RESPONSIVENESS-ELASTICITY of the development phase



Increasing the number of classifiers to develop, the response time of the development phase will increase exponentially due to the heavy computational load sustained by the development system during the training process.

For developing a new classifier, we would have the possibility to create it during the night, where clients won't be bothered from the temporary service discontinuation, so we would have about 5 hours available. Tests demonstrate that we would be able to withstand a considerable number of developed classifiers and still be below the time constraint of 5 hours.

### 3. RESILIENCY

#### Ingestion System

ID	INPUT	CONSEQUENCE	SCORE
I1	Missing records	If below the threshold, it is accepted (correction is assigned to the preparation system). If over the threshold, the raw session is discarded	1
I2	Missing labels	Error detected, raw session is discarded	4
I3	Duplicated records	Error raised by the unique identifier constraint	4
I4	Record containing an outlier	Error undetected	5
I5	Malformed configuration file	Error detected, the system will stop providing the details	2
I6	Late records	Error undetected	5
I7	Wrong record data type	Error undetected	5
<b>TOTAL</b>			<b>26</b>

#### Preparation System

ID	INPUT	CONSEQUENCE	SCORE
P1	Wrong structured Raw Session	Error detected and Raw Session discarded	4
P2	Missing samples	Error detected and corrected exploiting interpolation	1
P3	Late Samples	Error non detected	5
P4	Absolute Outliers	Error detected and corrected using predefined min / max values for each physical quantity	1
<b>TOTAL</b>			<b>11</b>

#### Segregation System

ID	INPUT	CONSEQUENCE	SCORE
S1	Wrong structured Prepared Session	Error detected and Prepared Session discarded	4
S2	Unbalanced Class	Error not detected. A user must analyse the graph and decided if classes are well balanced.	5
S3	Not uniform Data Distribution	Error not detected. A user must analyse the graph and decided if data are sufficiently well plotted.	5
TOTAL			14

#### Development System

ID	Input	Consequence	Score
D1	Late learning sets	System remains in the waiting stage.	4
D2	Hyperparameters file corrupted.	Issue detected. Trace back displayed and System stops.	3
D3	Wrong input of the Human task of inserting the UUID of the winner classifier.	The development system goes back to where the <b>average hyper parameters</b> are set.	1
D4	Wrong calibration set <b>values</b>	The model is trained on the wrong data.	5
D5	Wrong request body from the <b>Segregation</b> system.	Returns 404	1
TOTAL			14

#### Classification System

ID	INPUT	CONSEQUENCE	SCORE
CS1	Wrong classifier structure received	No detection, the classifier is deployed	5
CS2	Wrong prepared session structure	Validation error: the error is detected and noted in a log file for future analysis	3
CS3	Wrong prepared session values	No detection	5
CS4	Receiving a prepared session while in development phase	Validation error: the error is detected and noted in a log file for future analysis	3
CS5	Receiving a classifier while in production phase	Validation error: the error is detected and noted in a log file for future analysis	3
TOTAL			19

#### Evaluation System

ID	SCORE	CONSEQUENCE	SCORE
ES1	Invalid Label received	Validation error: the error is detected and noted in a log file for future analysis	3

<b>ES2</b>	Duplicated Label received	Duplicated label is automatically detected and is not added to the database	1
<b>ES3</b>	Missing labels from either expert or classifier	Presence of unpaired labels are detected and noted. Unpaired labels are exclude from the reporting process	3
<b>TOTAL</b>			7