

UNIVERSITÀ DI PISA

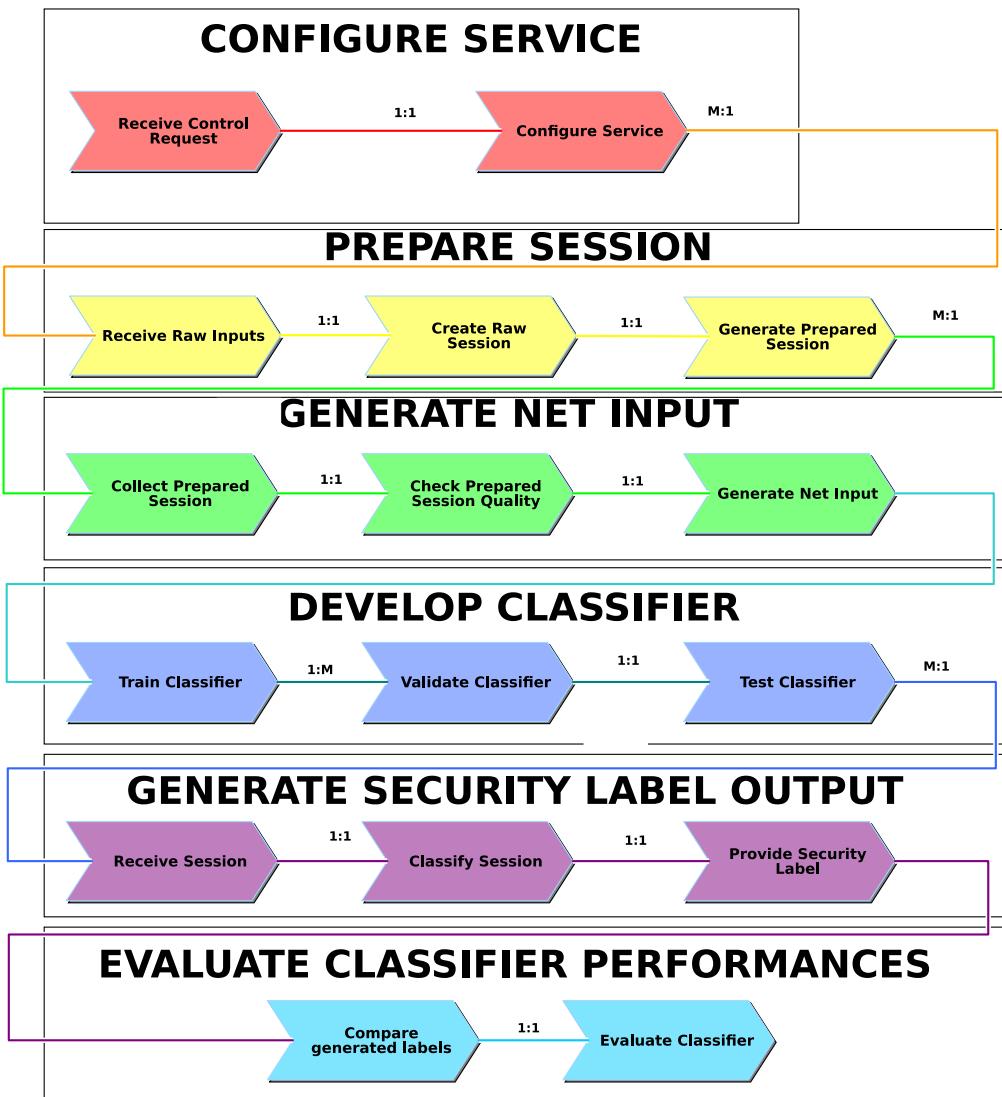
Software System engineering project

\$ecure Pos

a.a. 2023/2024

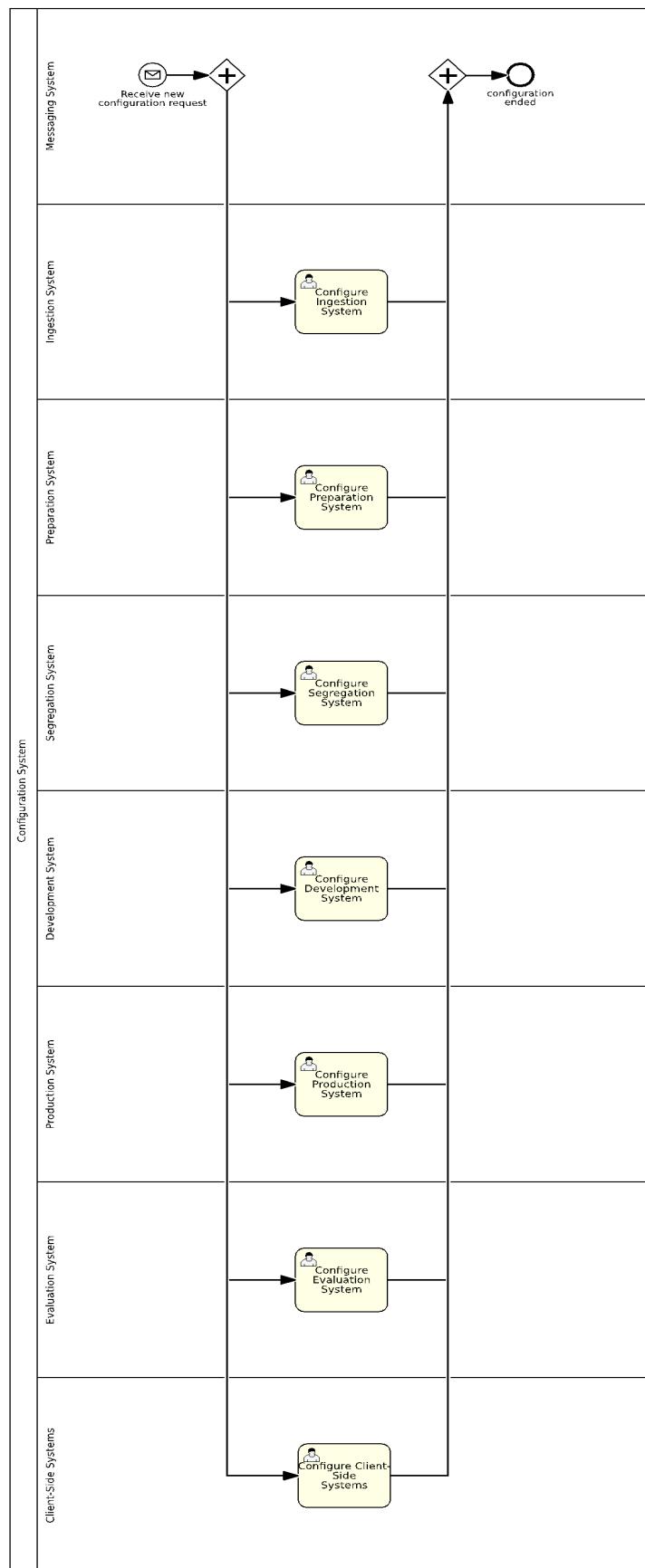
Antonino Nigro
Fabio Piras
Giacomo Volpi
Guillaume Quint
Mirco Concu

Process landscape

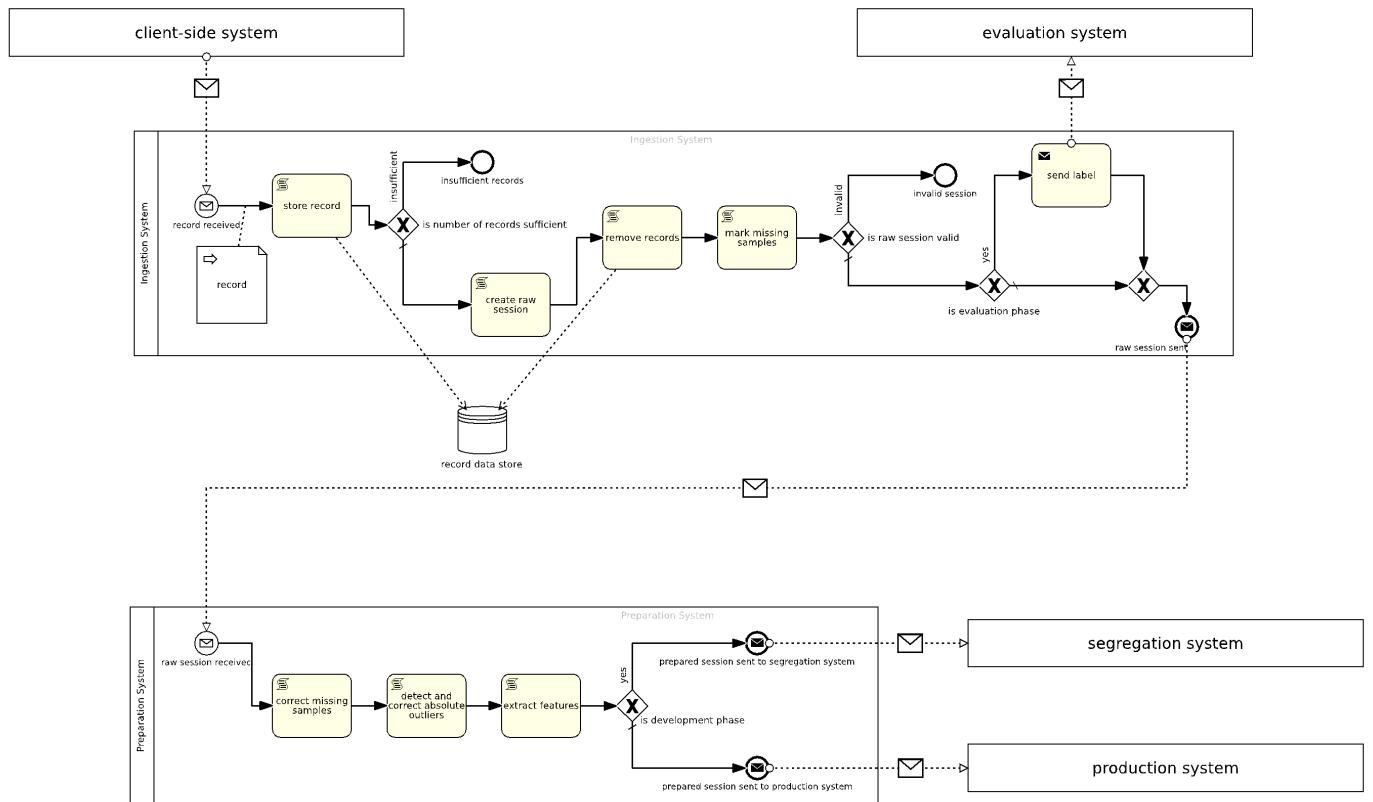


Design

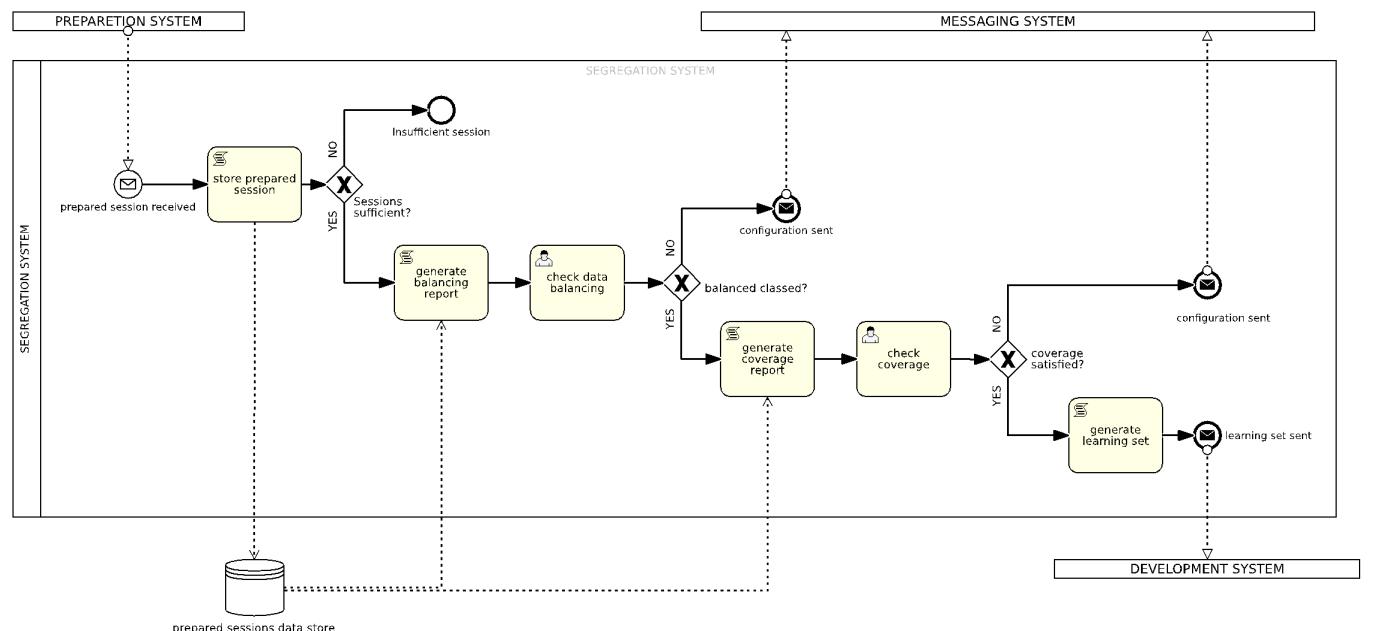
0) Configuration System



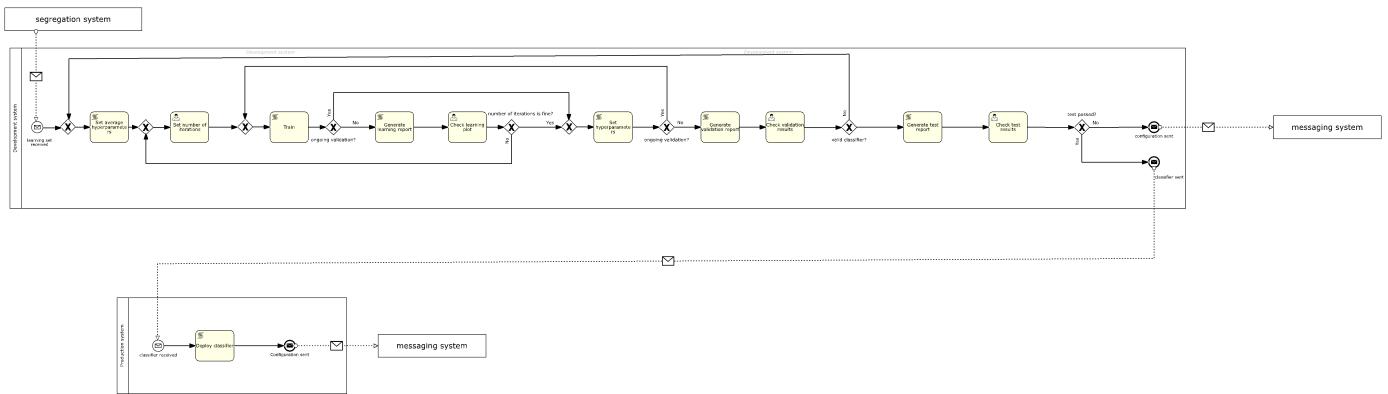
1) Prepare Session



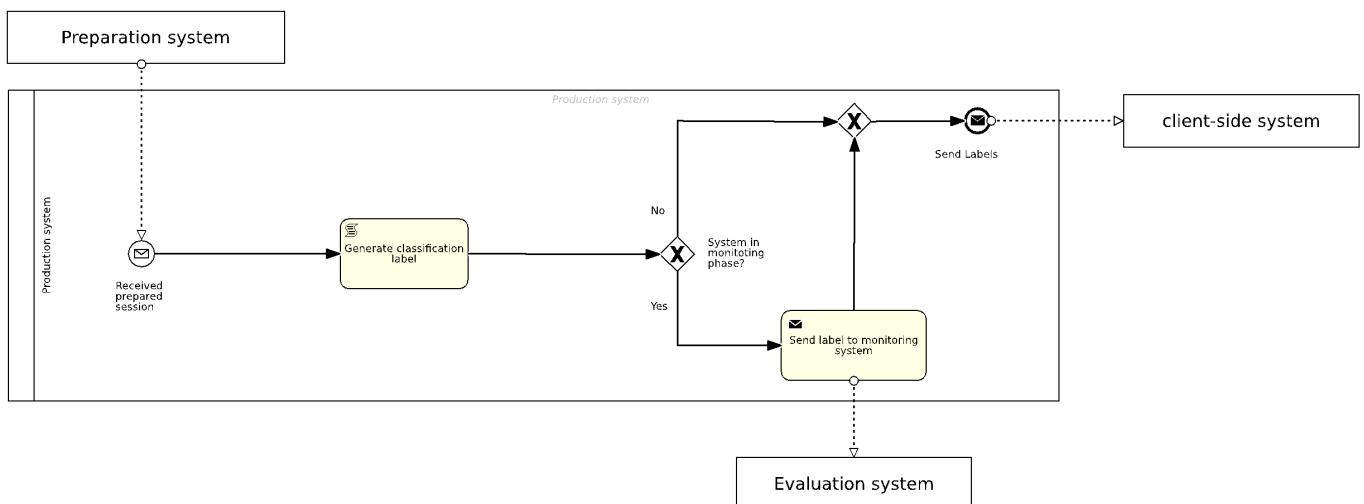
2) Generate learning set



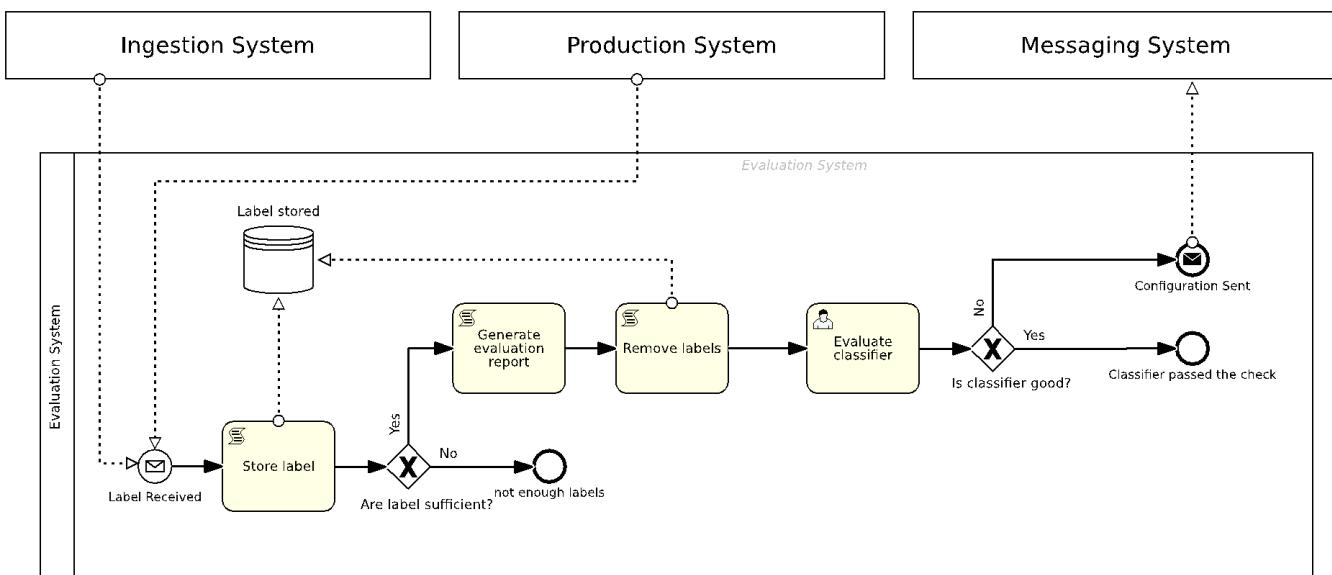
3)Develop Classifier



4)Classify session



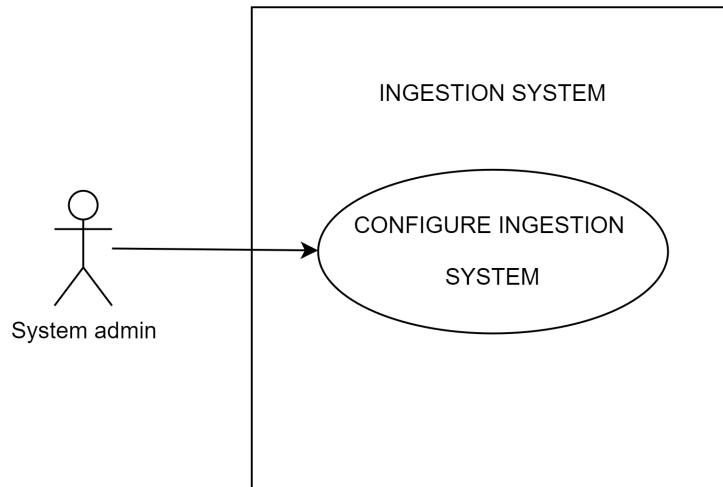
5)Evaluation System



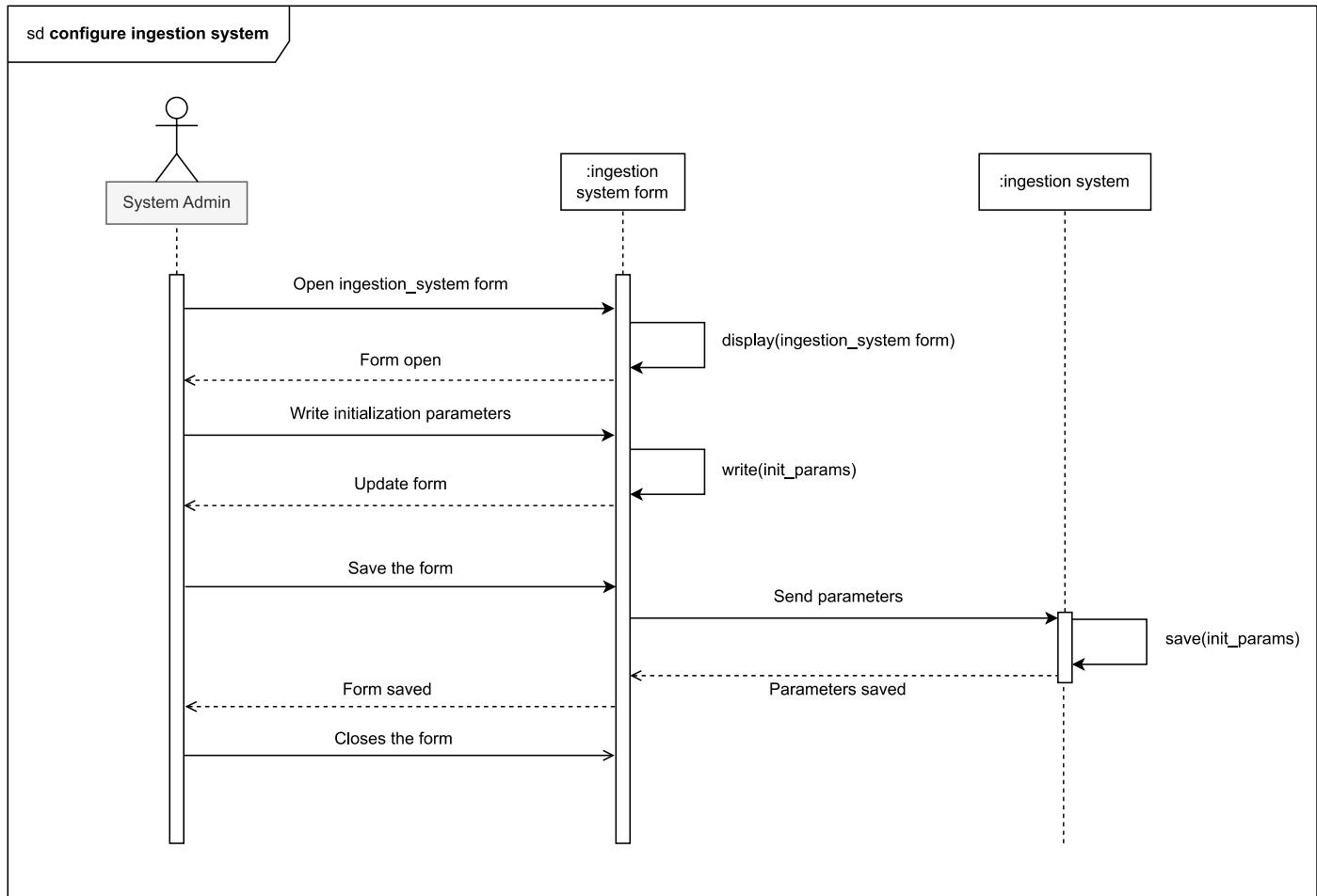
Analysis

Task modeling

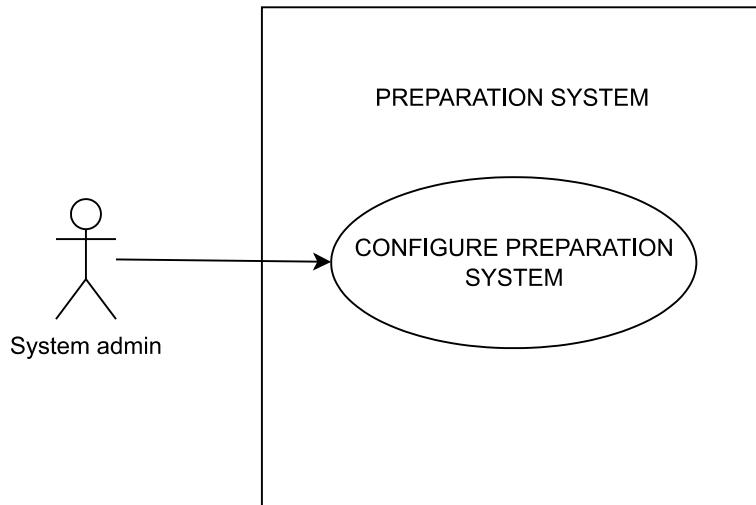
Ingestion System

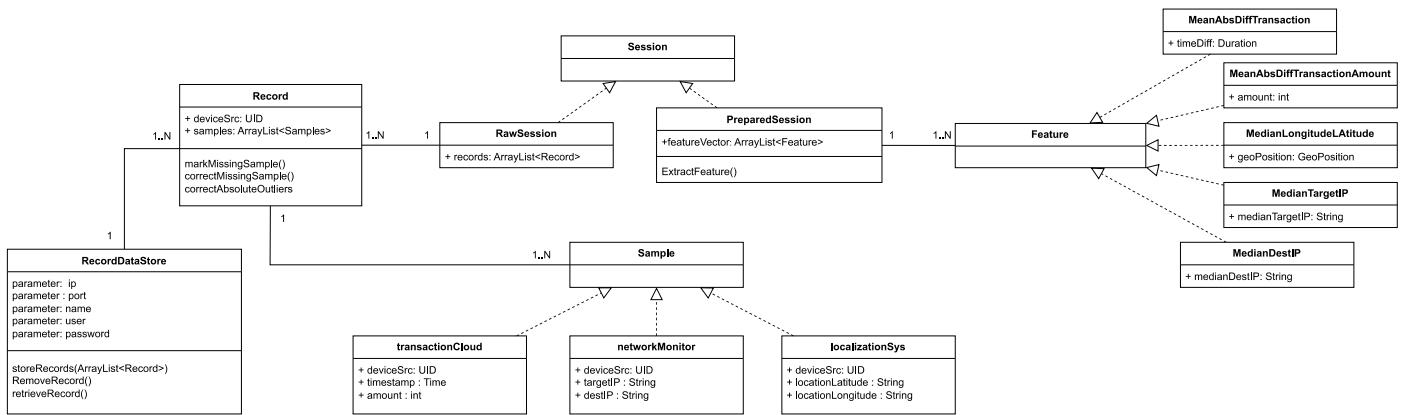
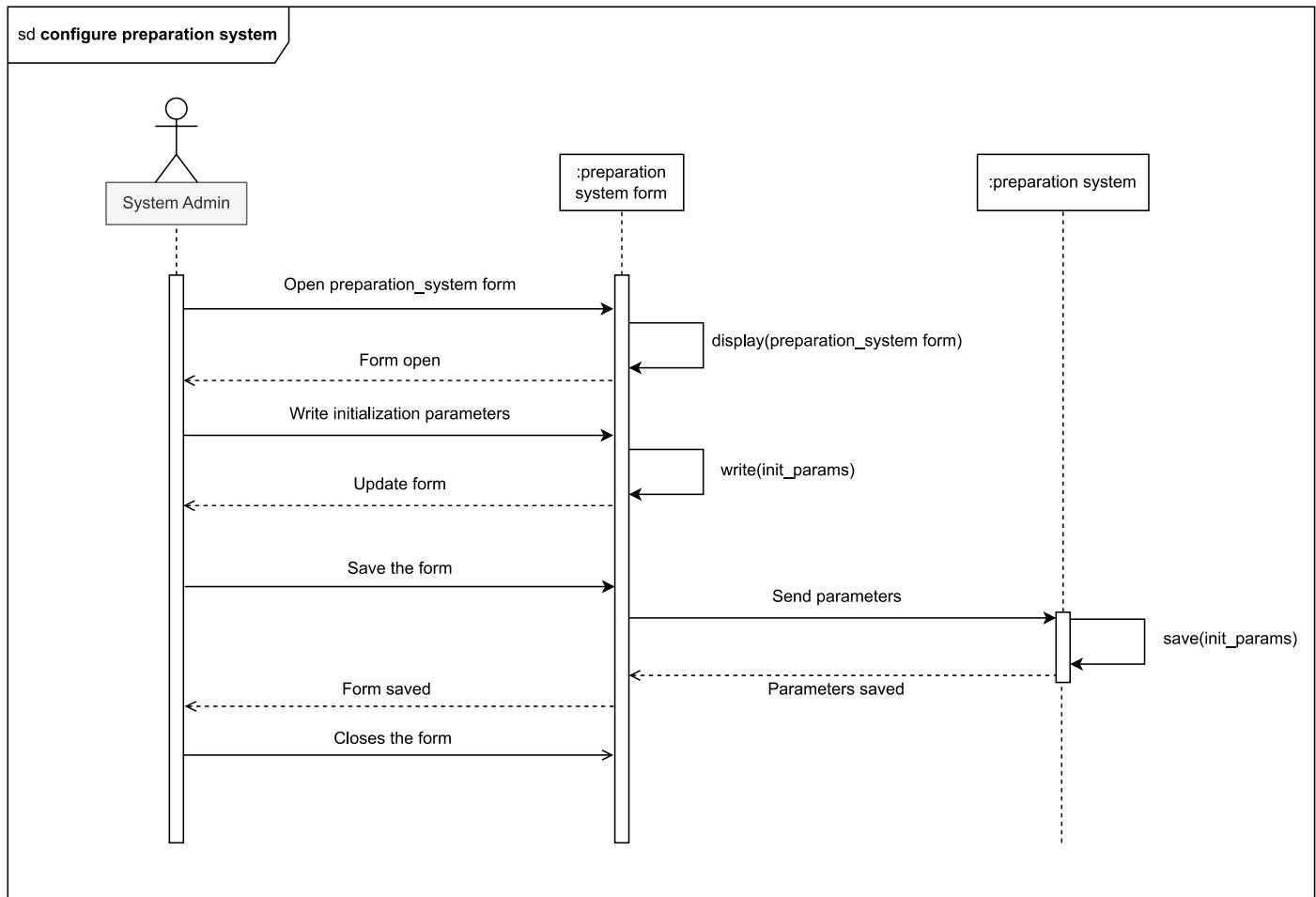
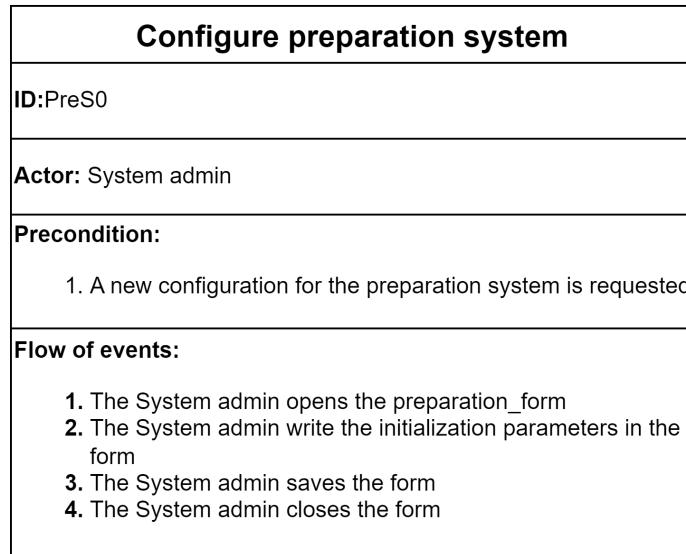


Configure ingestion system	
ID:	IS0
Actor:	System admin
Precondition:	<ol style="list-style-type: none">1. A new configuration for the ingestion system is requested
Flow of events:	<ol style="list-style-type: none">1. The System admin opens the ingestion_form2. The System admin write the initialization parameters in the form3. The System admin saves the form4. The System admin closes the form

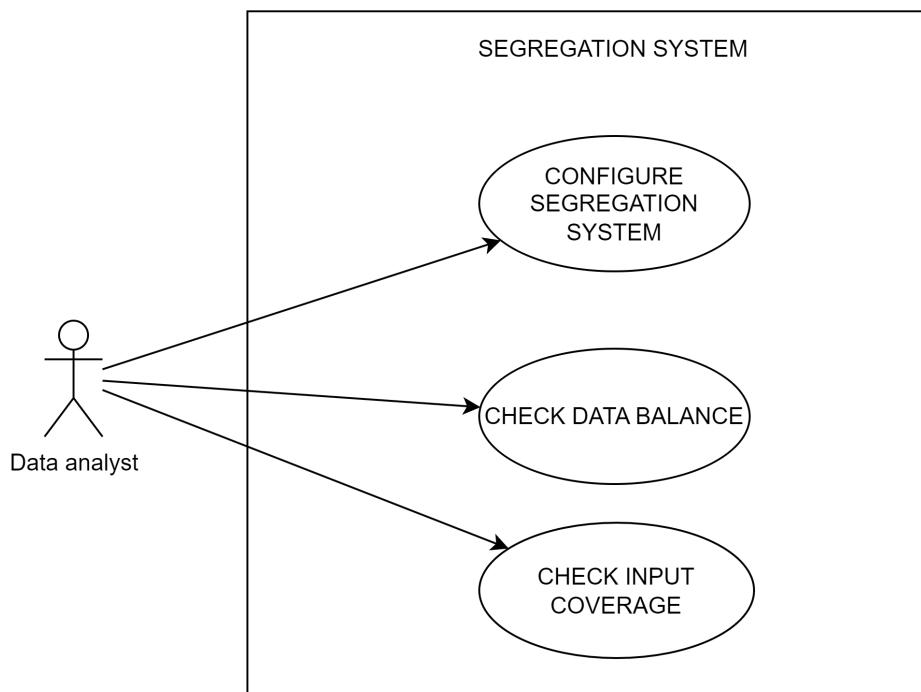


Preparation System

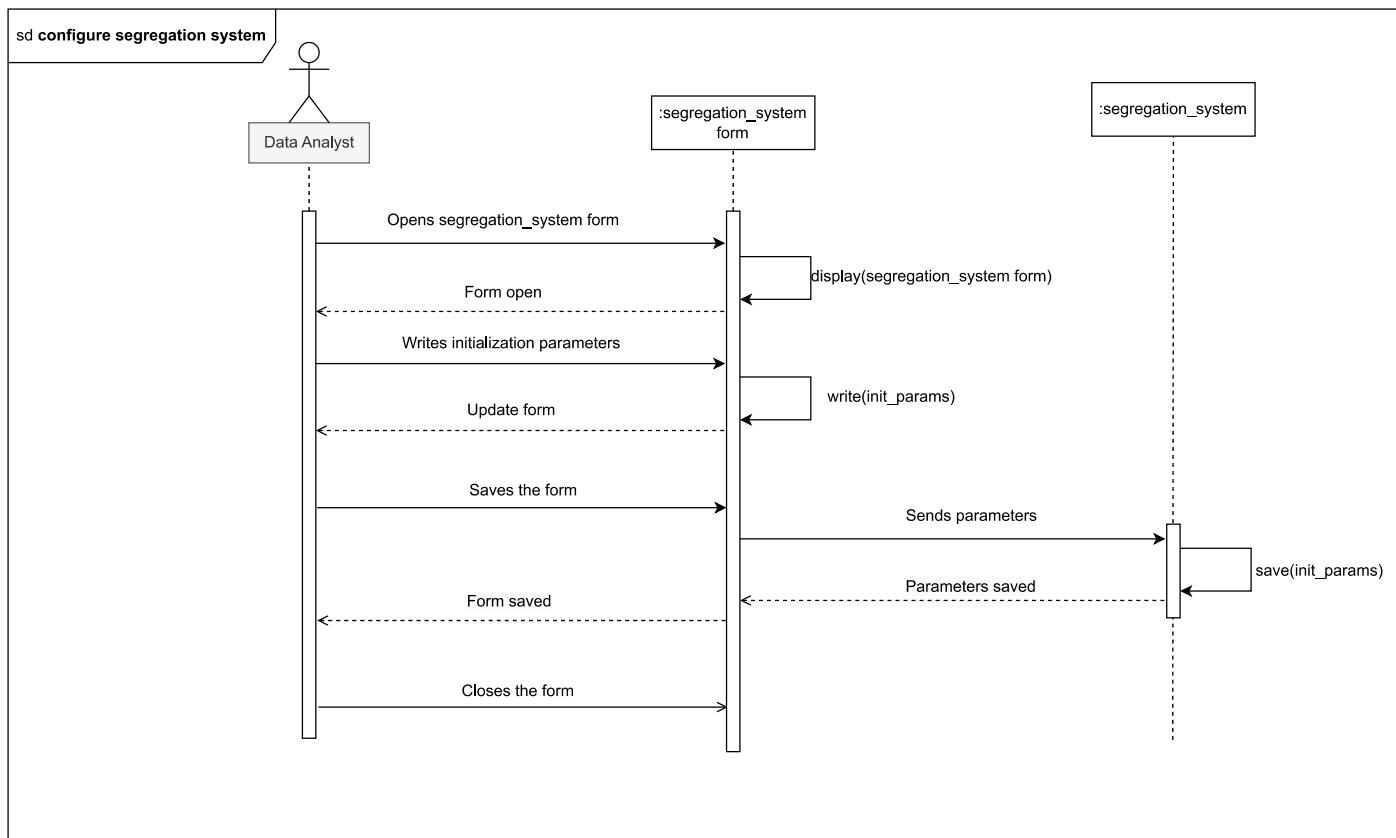




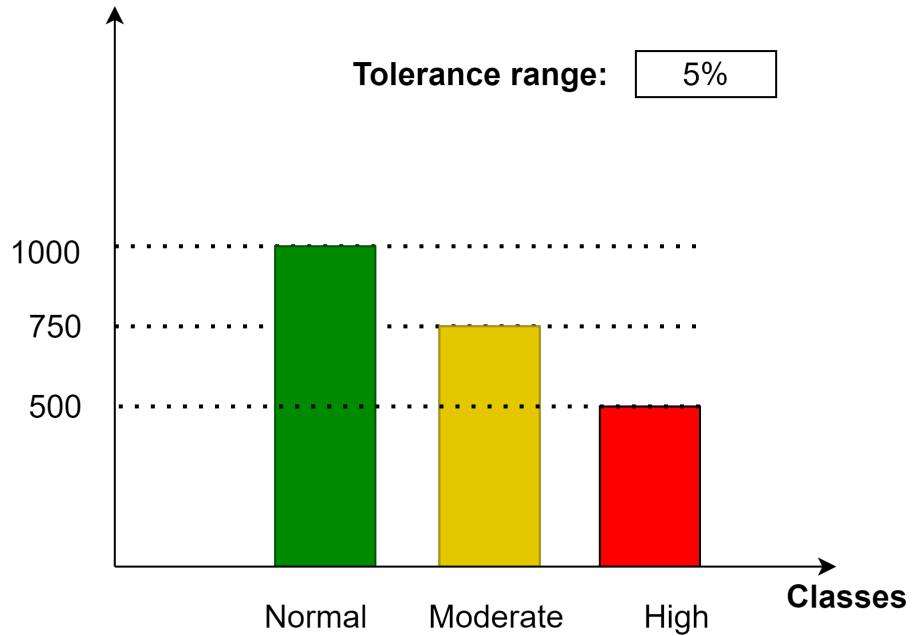
Segregation System



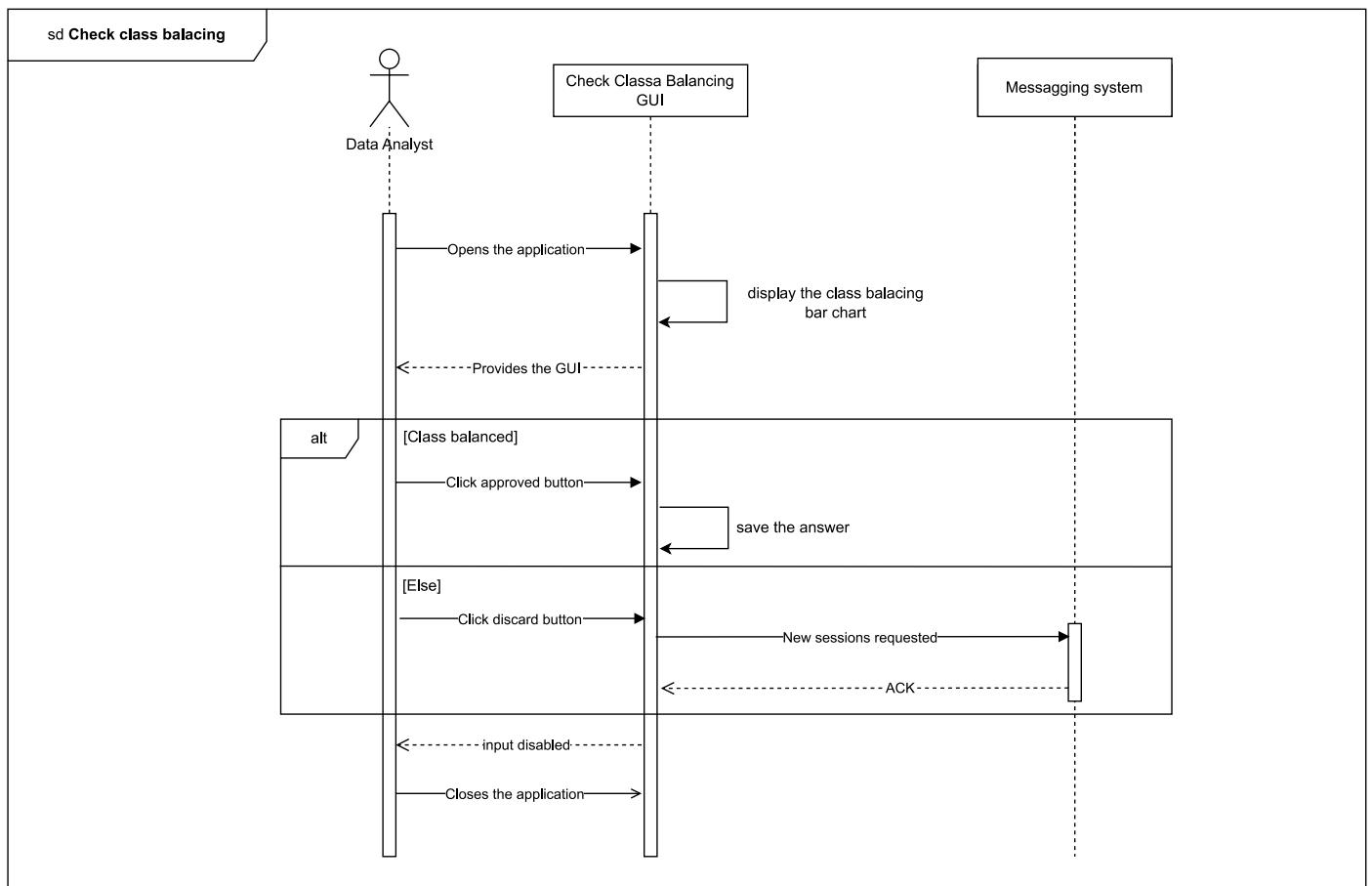
Configure segregation system	
ID:	SS0
Actor:	Data Analyst
Precondition:	<ol style="list-style-type: none">1. A new configuration for the segregation system is requested
Flow of events:	<ol style="list-style-type: none">1. The Data Analyst opens the segregation_config.json file2. The Data Analyst writes the initialization parameters in the form3. The Data Analyst saves the form4. The Data Analyst closes the form



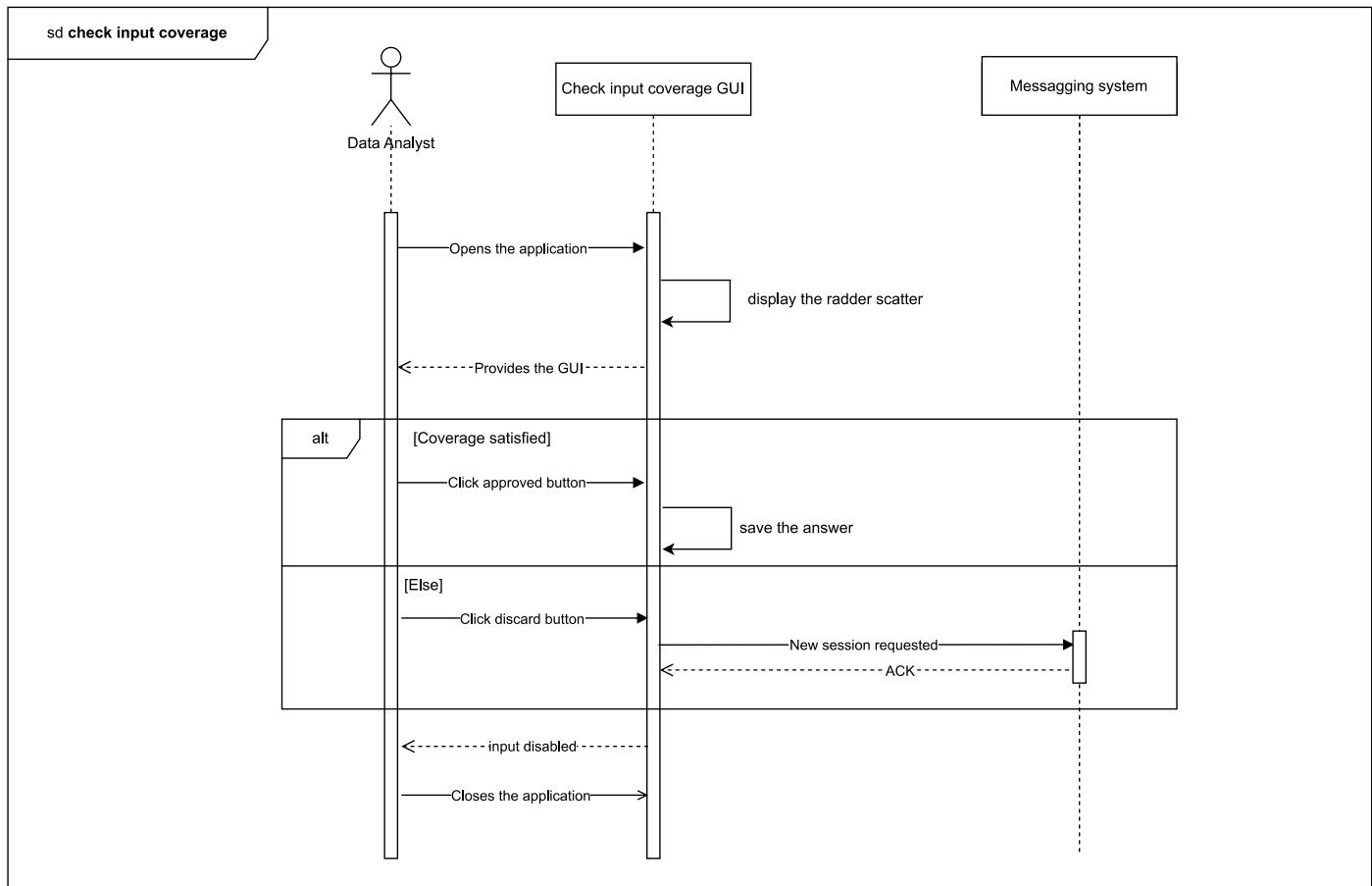
Check data balance	
ID:SS1	
Actor: Data Analyst	
Precondition:	<ol style="list-style-type: none"> 1. The application has generated the balancing report
Flow of events:	<ol style="list-style-type: none"> 1. The Data Analyst opens the data balancing report 2. The Data Analyst checks data balance 3. IF the labels are within the tolerance range <ol style="list-style-type: none"> 3.1 The Data Analyst presses the "Accept" button 4. ELSE the Data Analyst inserts the required amount of samples <ol style="list-style-type: none"> 4.1 The Data Analyst presses the "Discard" button
Postcondition:	
Alternative flow 1:	<ol style="list-style-type: none"> 1. The system continues normally
Alternative flow 2:	<ol style="list-style-type: none"> 1. The system requests a new configuration

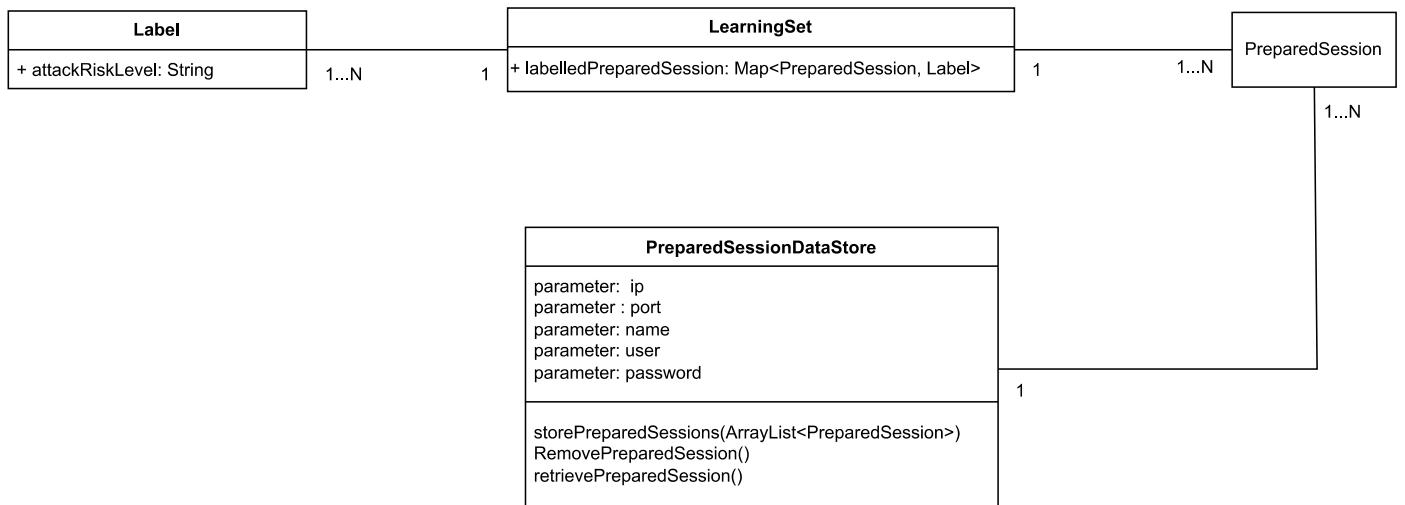
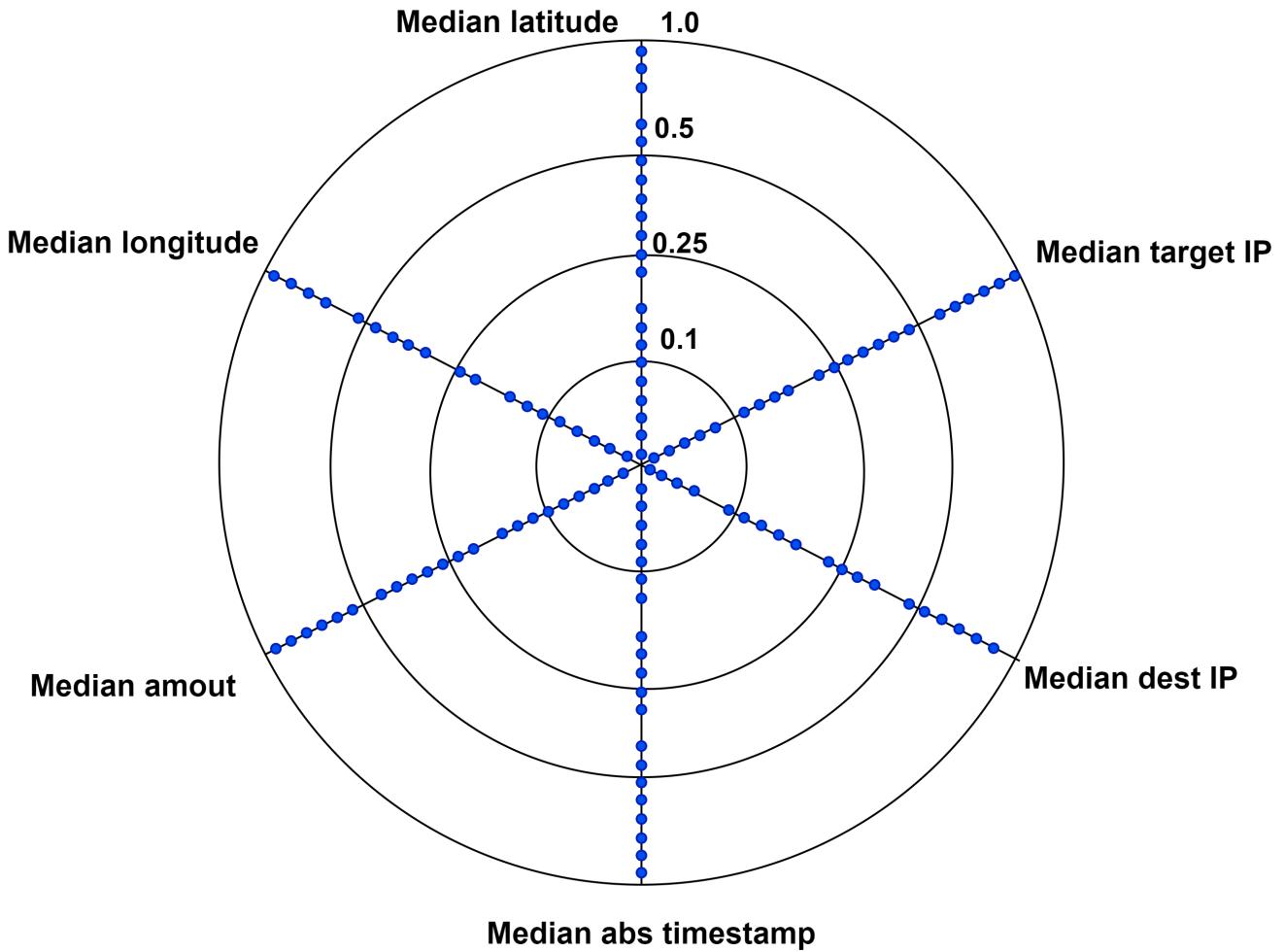


Approve **Discard**

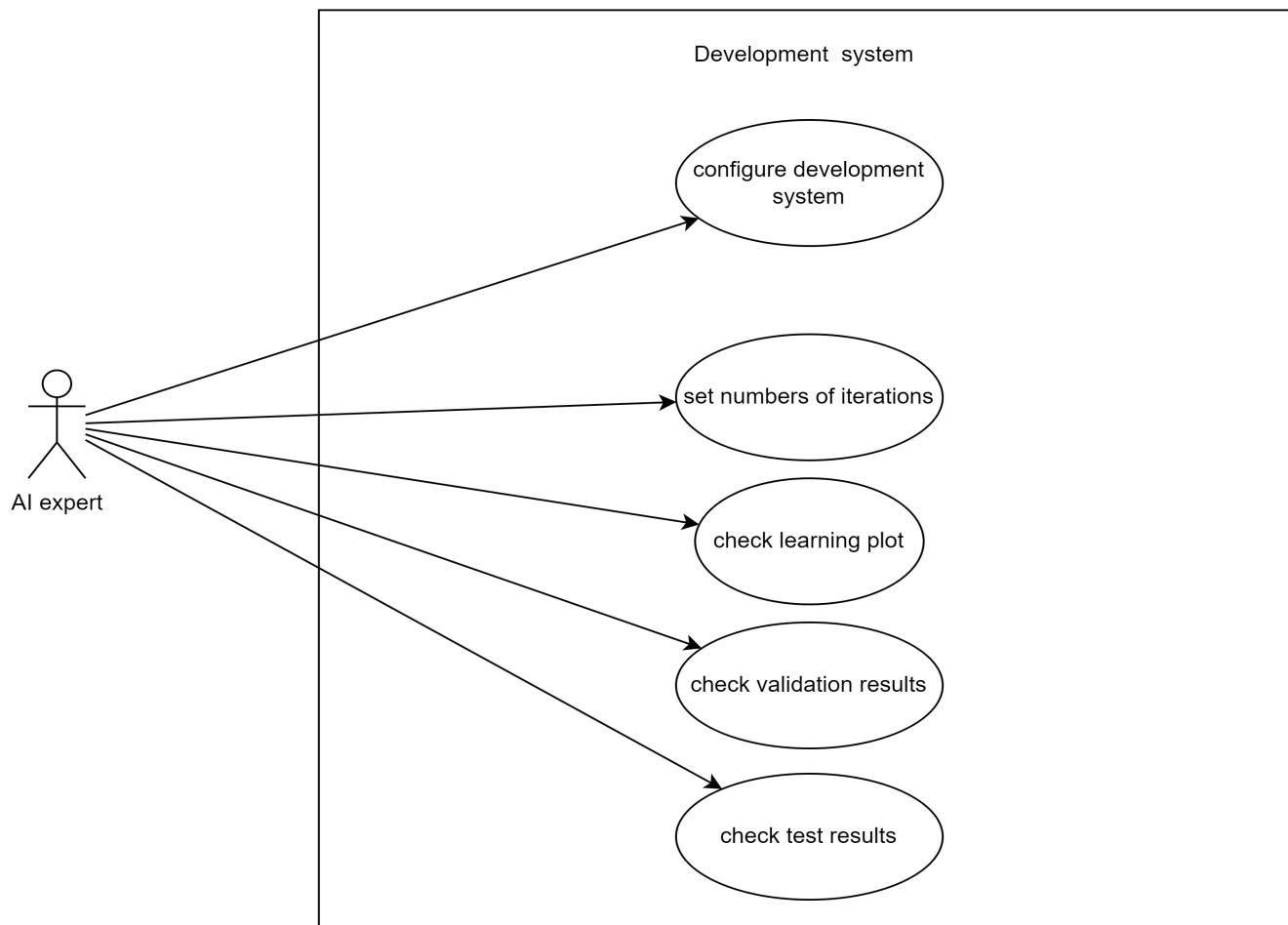


Check input coverage
ID: SS2
Actor: Data Analyst
Precondition:
1. The application has generated the coverage report
Flow of events:
1. The Data Analyst opens the data coverage report 2. The Data Analyst checks the coverage radar chart 3. IF the classes coverage are satisfied 3.1 The Data Analyst presses the "Accept" button 4. ELSE the Data Analyst presses the "Reject" button 5. The Data Analyst closes the report
Postcondition:
Alternative flow 1:
1. The system continues normally
Alternative flow 2:
1. The system requests a new configuration

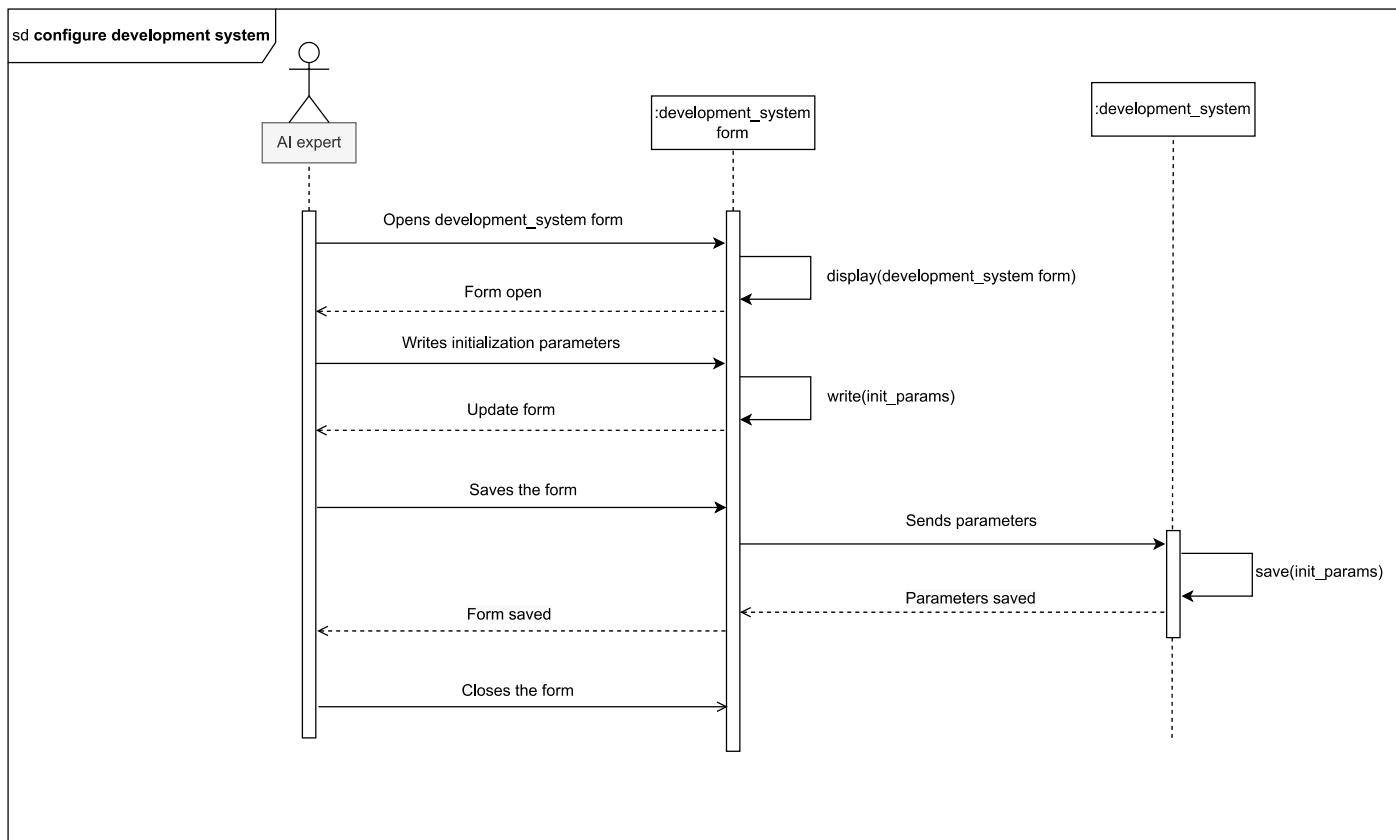




Development System



Configure development system	
ID:	DS0
Actor:	AI expert
Precondition:	<ul style="list-style-type: none">1. A new configuration for the Development system is requested
Flow of events:	<ul style="list-style-type: none">1. The AI expert opens the development_form file2. The AI expert write the initialization parameters in the form3. The AI expert saves the form4. The AI expert closes the form

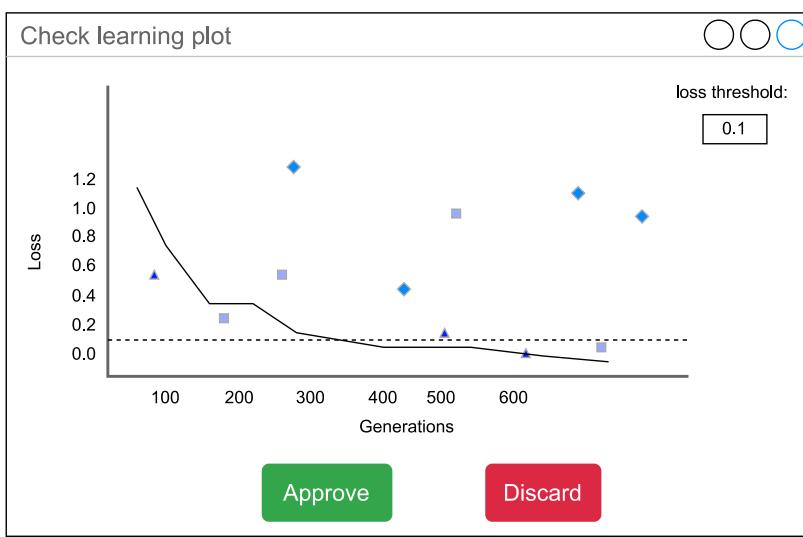
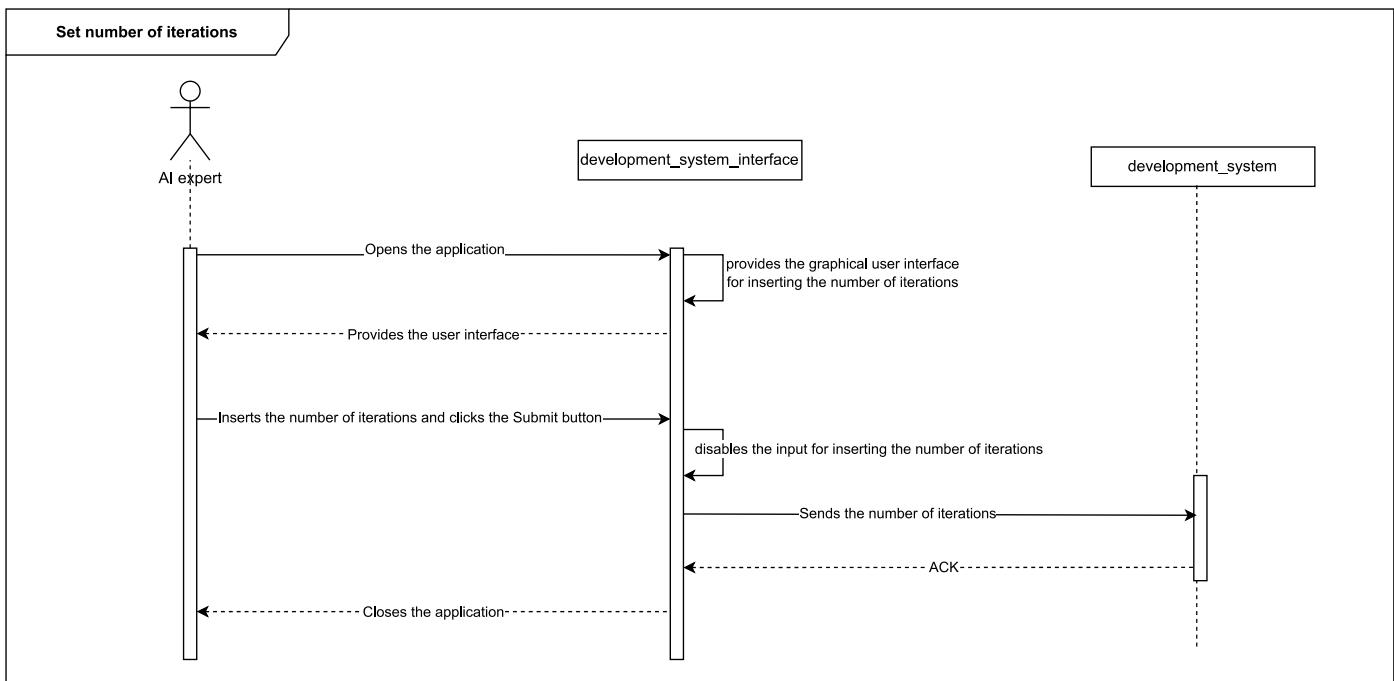


Set number of iterations

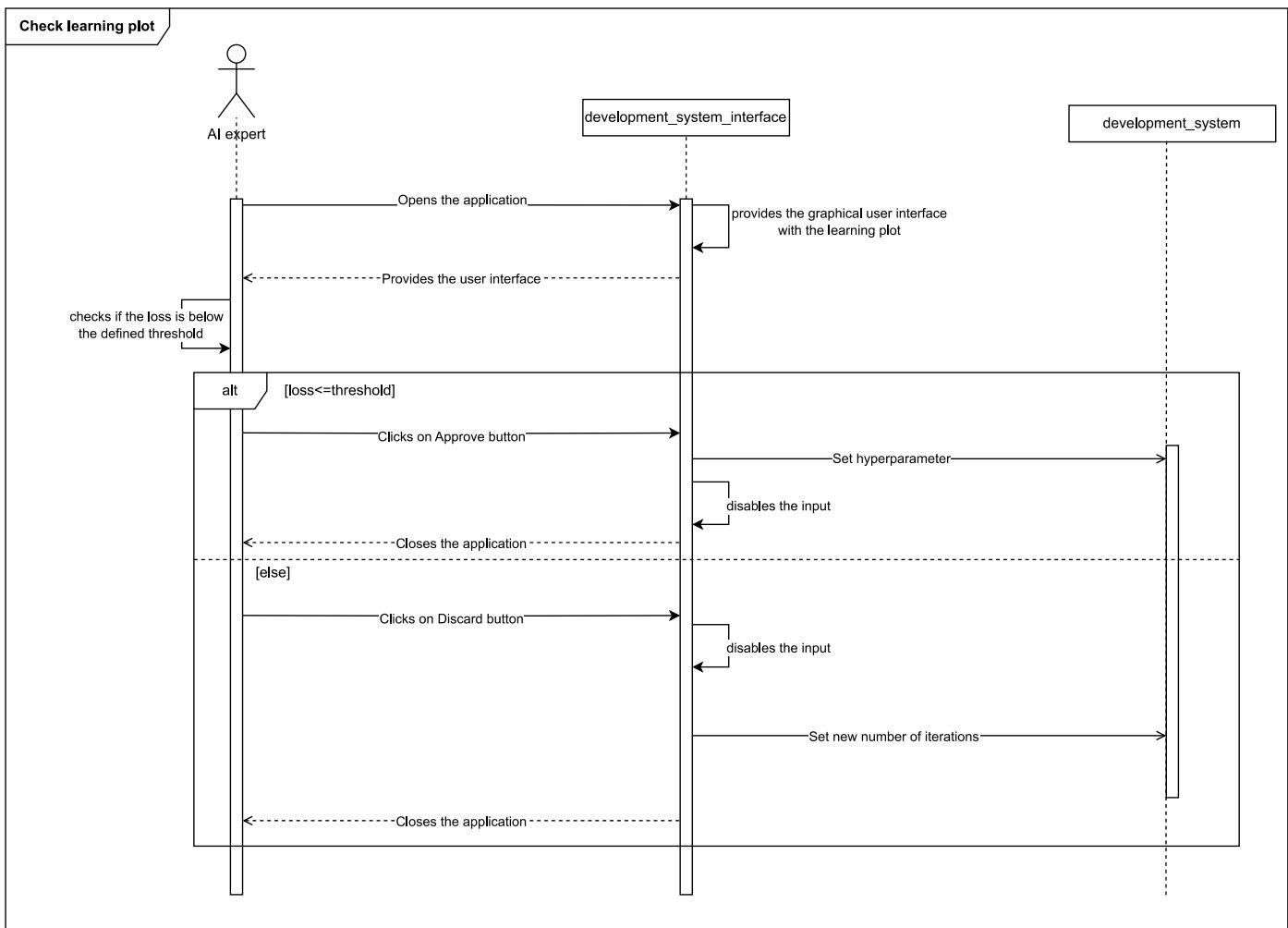
Please insert a number

set number of iterations Submit

Set number of iterations		
ID: DS1		
Actor: AI expert		
Precondition:		
1. The application has received the learning set and has set the numbers of hyperparameters		
Flow of events:		
1. IA expert opens the interface 2. IA expert inserts the number of iterations 3. IA expert clicks the submit button to confirm his/her choice		
Postcondition:		
1. The development system starts the training phase		



Check learning plot	
ID: DS2	
Actor: AI expert	
Precondition:	1. The application has generated the learning plot
Flow of events:	<ol style="list-style-type: none"> 1. AI expert opens the interface 2. AI expert checks the learning plot 3. IF the loss is below the defined threshold <ol style="list-style-type: none"> 3.1 AI expert press the "Approve" button 3.2 ELSE AI expert press the "Discard" button
Postcondition:	1. The development systems set the hyperparameter
Alternative flow 1:	1. The development system requests a new number of iterations
Alternative flow 2:	

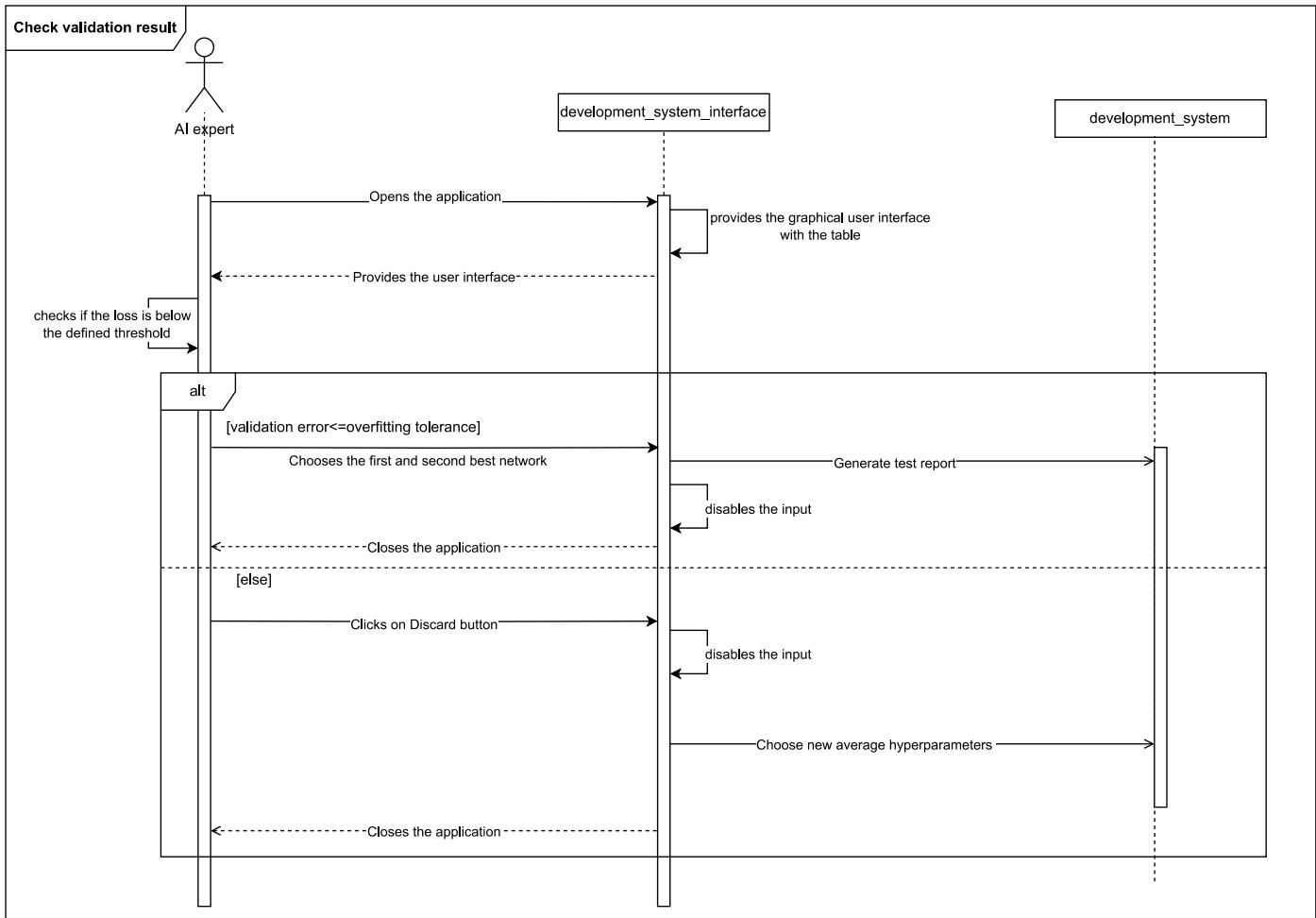


Check validation result

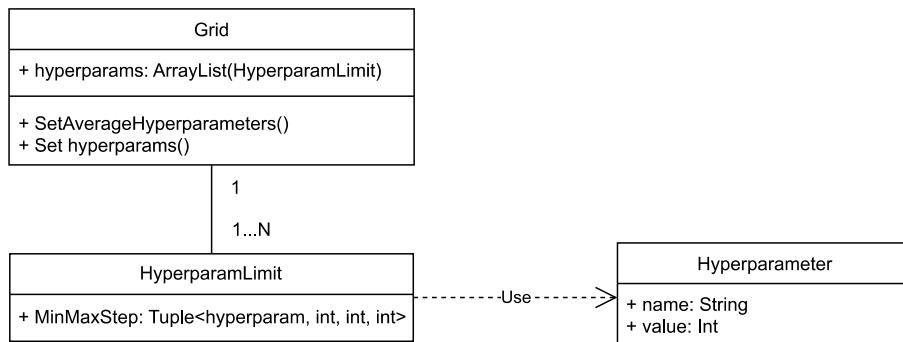
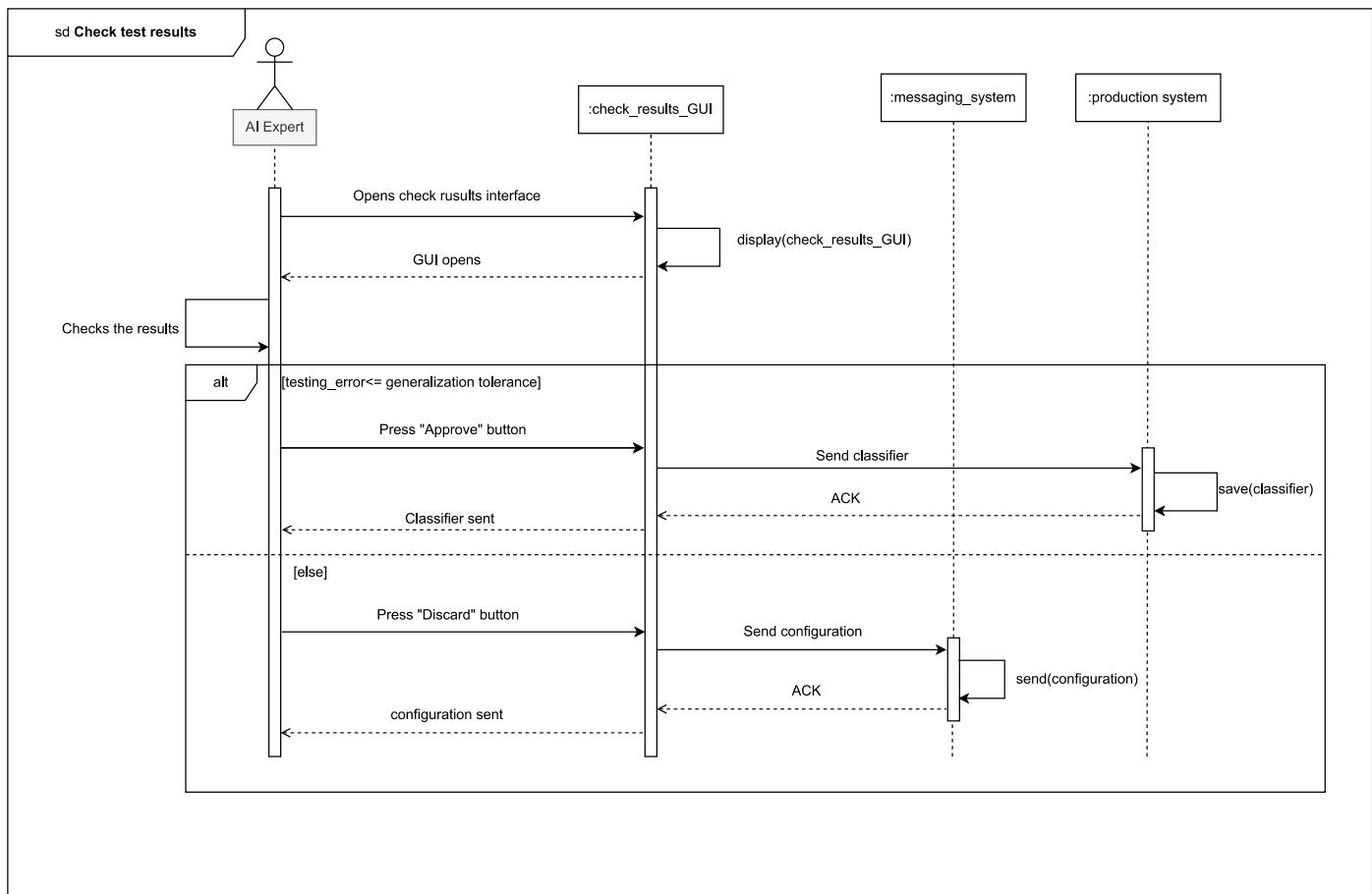
Result						
Name	MSE ▼	Validation error	Training error	# layers	# neurons	Validation-Training error
Classifier 1	0,05	3	2	7	289	1
Classifier 2	0,1	6	4	5	456	2
Classifier 3	0,15	7	3	7	1736	4
Classifier 4	0,20	8	5	6	539	3
Classifier 5	0,25	9	2	5	473	5

 Overfitting tolerance:
1.5

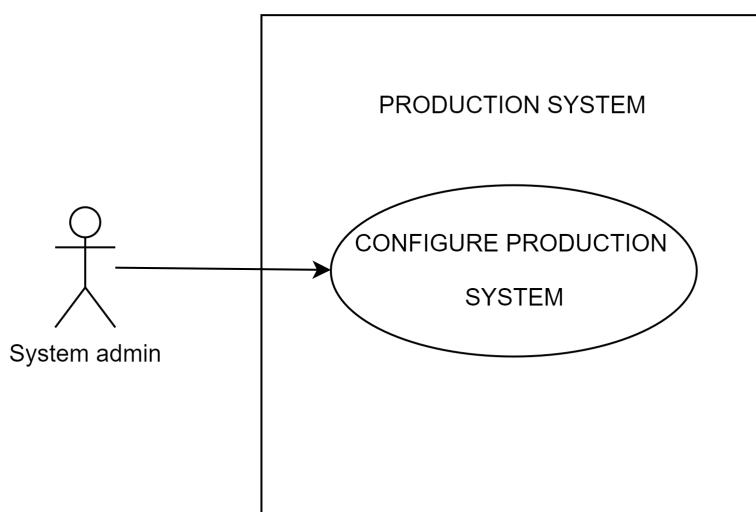
Check validation results	
ID:DS3	
Actor:	AI expert
Precondition:	
1. The application has generated the validation report	
Flow of events:	
1. AI expert opens the interface 2. AI expert checks the validation results 3. AI expert chooses the first best network, which has the lowest validation error and is under the overfitting tolerance 4. AI expert chooses the second best network 5. IF the validation error between the first and best network is very similar 5.1 AI expert choose the network with the lowest complexity	
Postcondition:	
Alternative flow 1:	
1. The development systems generate the test report	
Alternative flow 2:	
1. The development system chooses new average hyperparameter	

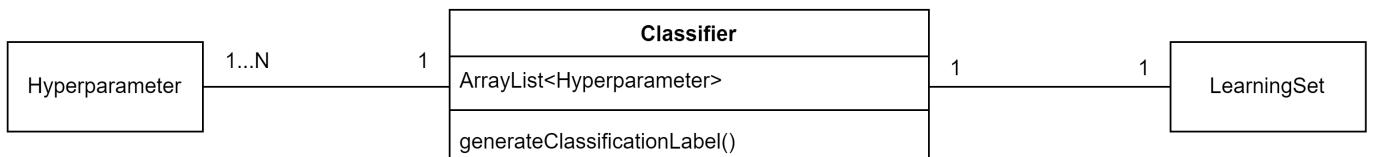
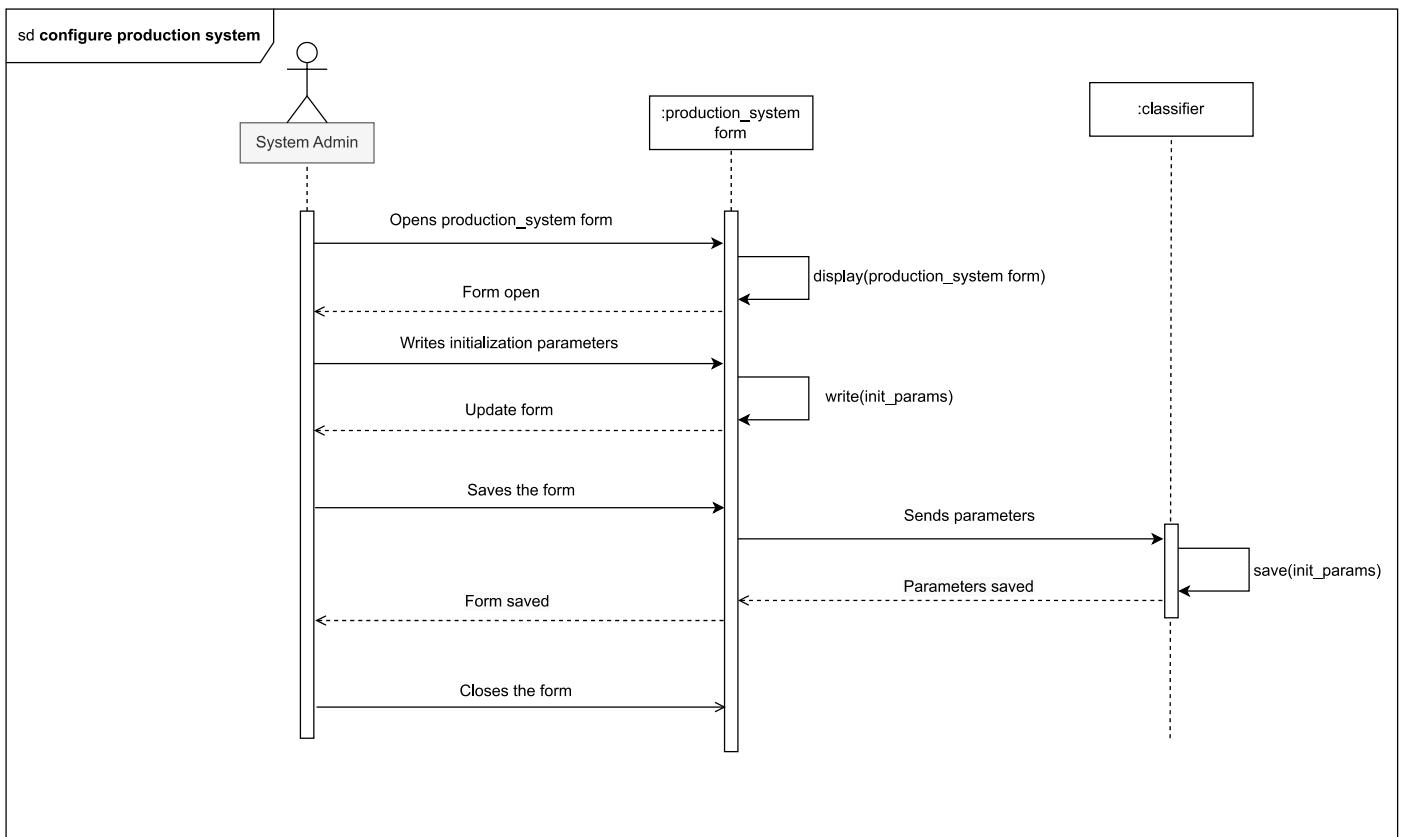
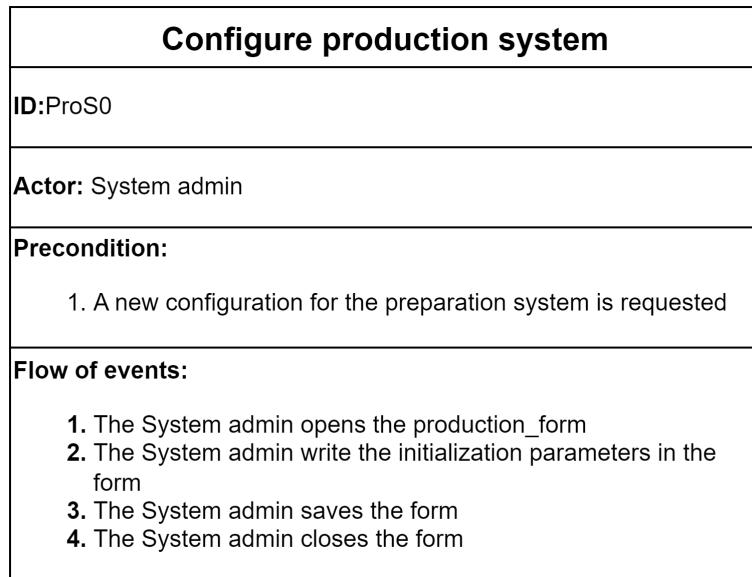


Check test results
ID: DS4
Actor: AI expert
Precondition:
1. The application has generated the test report
Flow of events:
1. AI expert opens the interface 2. AI expert checks the test report 3. If the difference between validation and test error is below the generalization tolerance: 3.1 AI expert press the "Approve" button 4. ELSE AI expert press the "Discard" button
Postcondition: Alternative flow 1: 1. The development systems sends the classifier to the production system
Alternative flow 2: 1. The development system sends the configuration

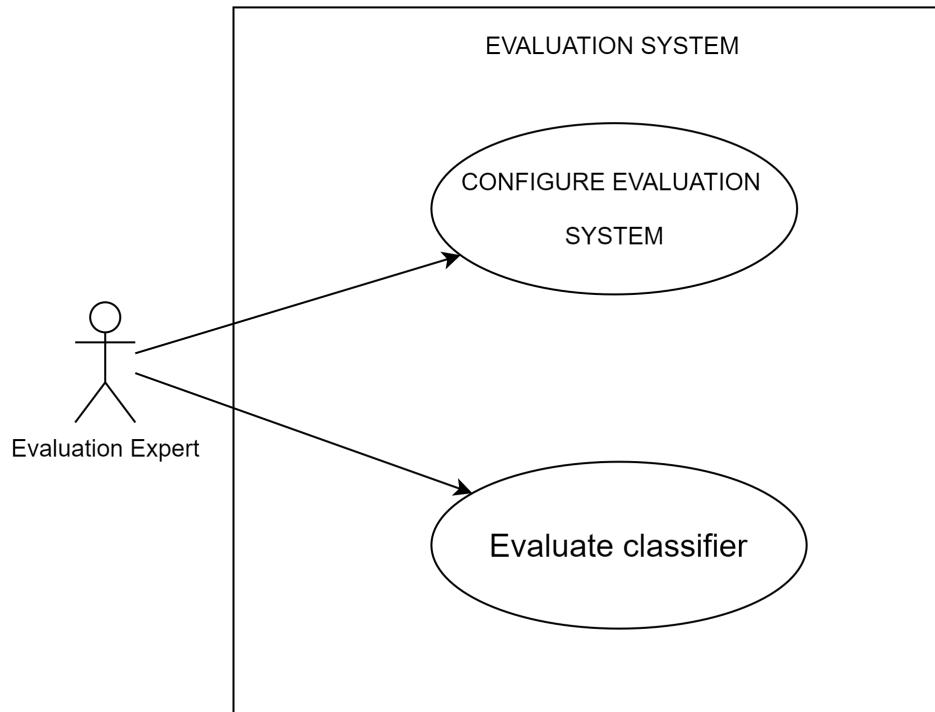


Production System

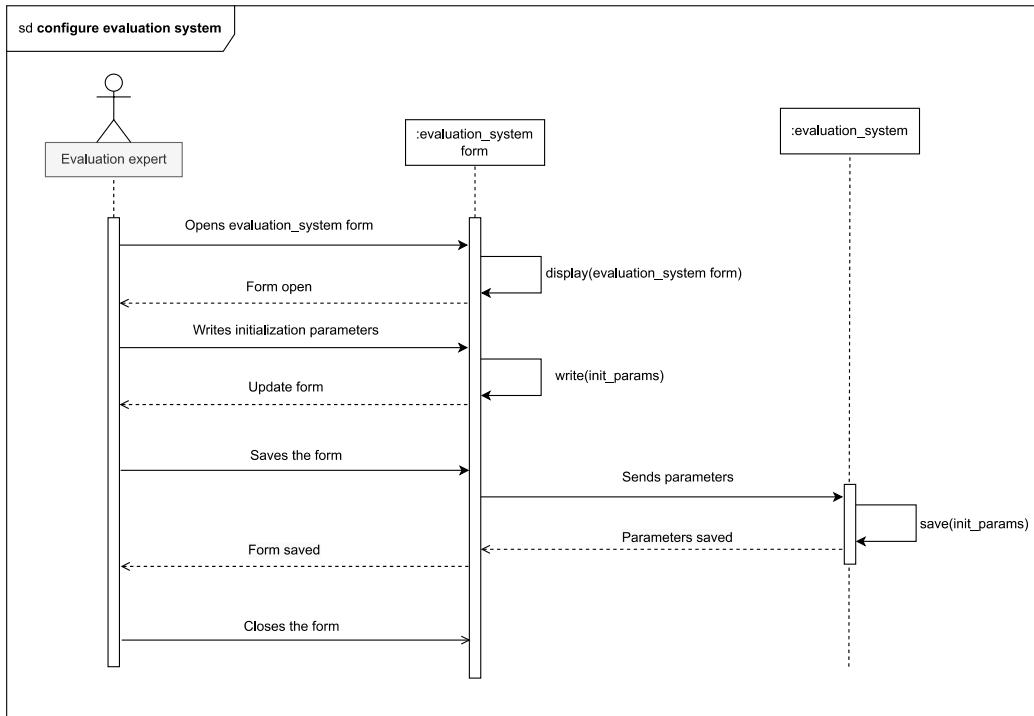




Evaluation System



Configure evaluation system	
ID: ES0	
Actor: Evaluation Expert	
Precondition:	<ol style="list-style-type: none">1. A new configuration for the evaluation system is requested
Flow of events:	<ol style="list-style-type: none">1. The Evaluation Expert admin opens the evaluation_form2. The Evaluation Expert write the initialization parameters in the form3. The Evaluation Expert saves the form4. The Evaluation Expert closes the form



Evaluate classifier	
ID:ES1	
Actor:	Evaluation Expert
Precondition:	<ol style="list-style-type: none"> Labels are enough and report has been generated.
Flow of events:	<ol style="list-style-type: none"> The actor opens the Evaluation Report The actor checks the labels, number of error and consecutives errors IF the total error are less than the tolerated amount and the max consecutive errors are less than the tolerated <ol style="list-style-type: none"> The actor presses the "Accept" button ELSE the actor presses the "Reject" button
Postcondition: Alternative flow 1:	<ol style="list-style-type: none"> The system continues normally.
Alternative flow 2:	<ol style="list-style-type: none"> The system waits for new labels.

**QUALITY CHECK ERROR REPORT**

Expert label class	Anomaly detector label class	Result
1	2	✗
2	2	✓
3	3	✓
3	1	✗
4	5	✗
3	4	✗
2	2	✓

Max number of errors tolerated (th1): **5**

Max number of consecutive errors tolerated (th2): **2**

Discard

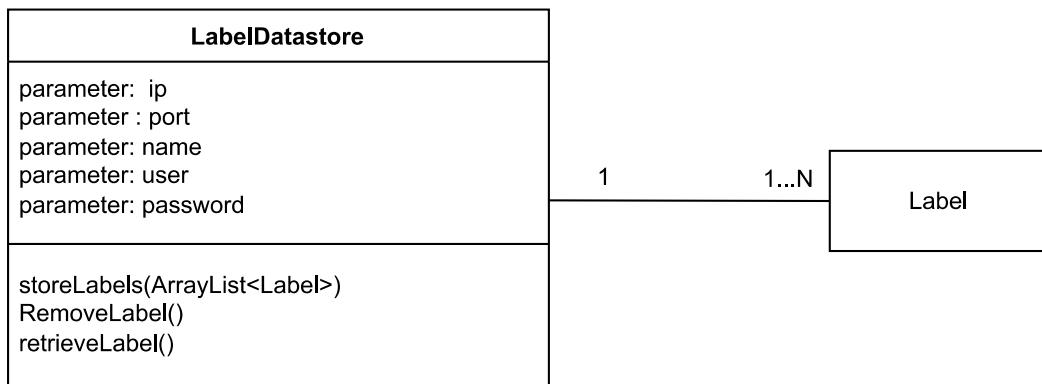
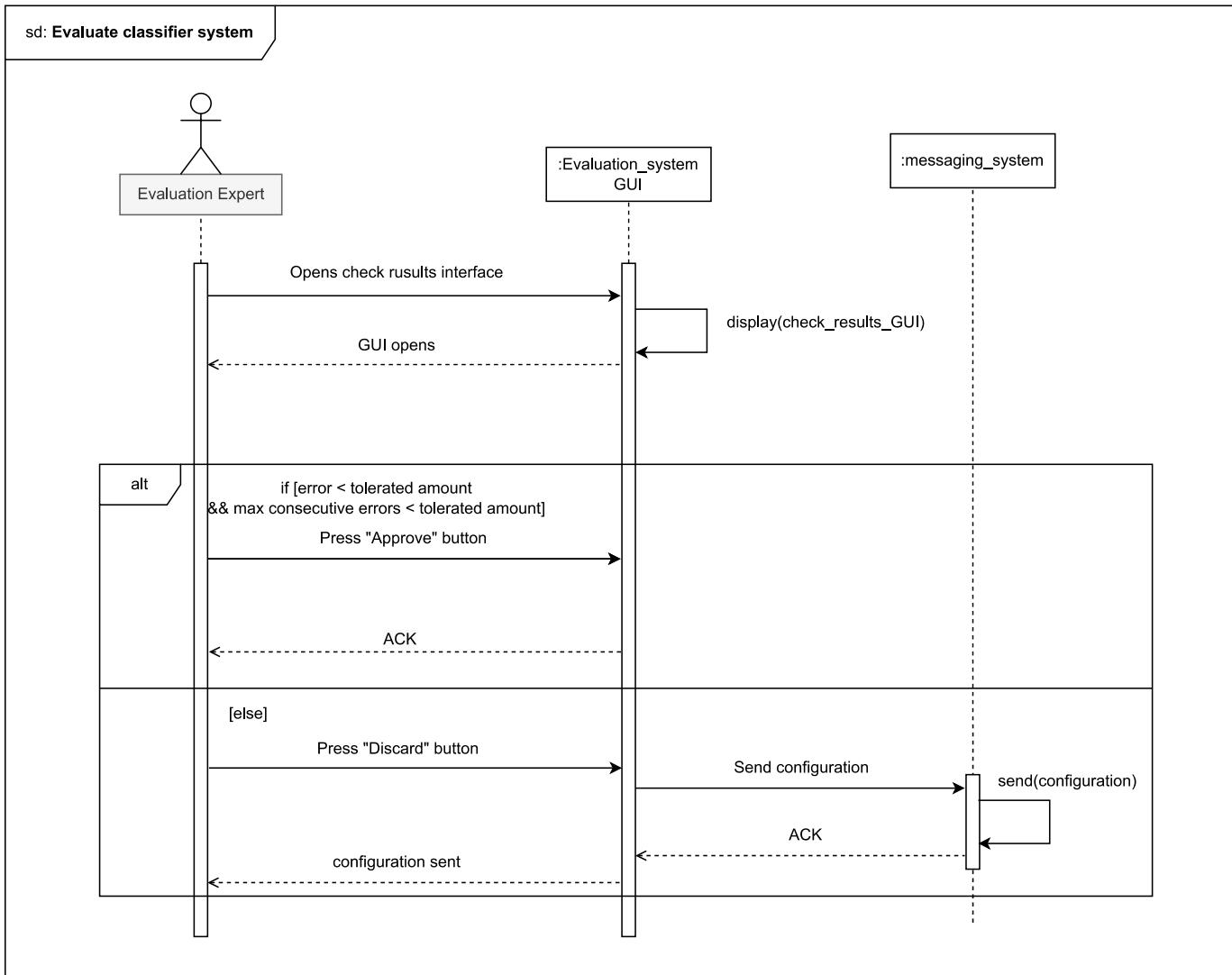
Total errors: 4

Max consecutive errors: 3

Approve

th1 satisfied ($4 < 5$)

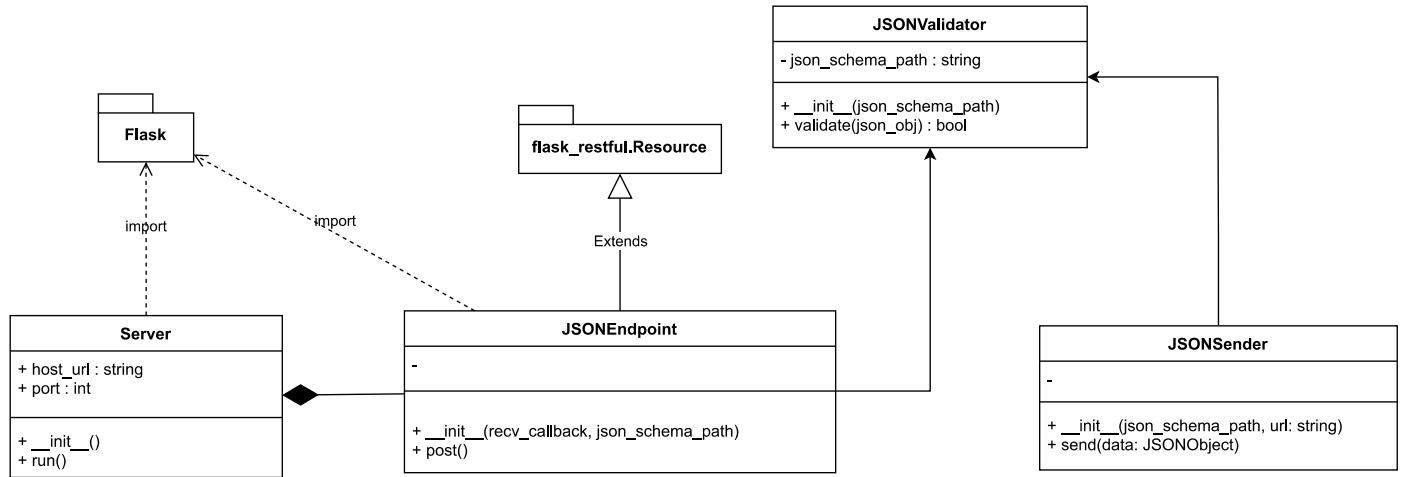
th2 exceeded ($3 > 2$)



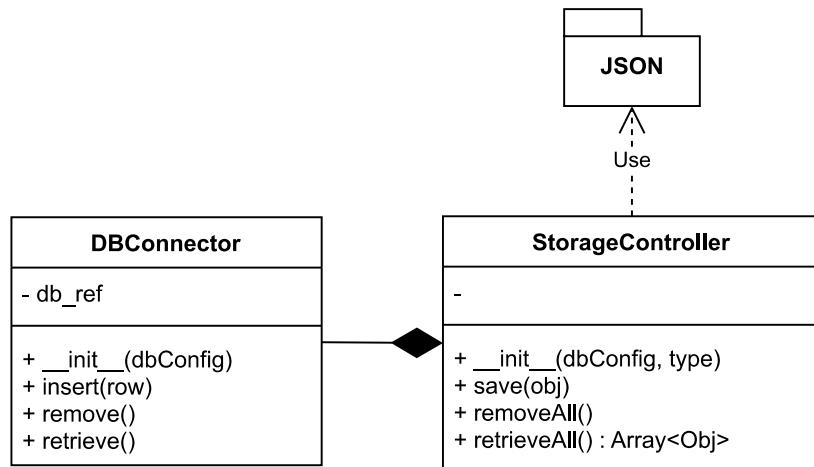
APPLICATION LOGIC

Common Systems

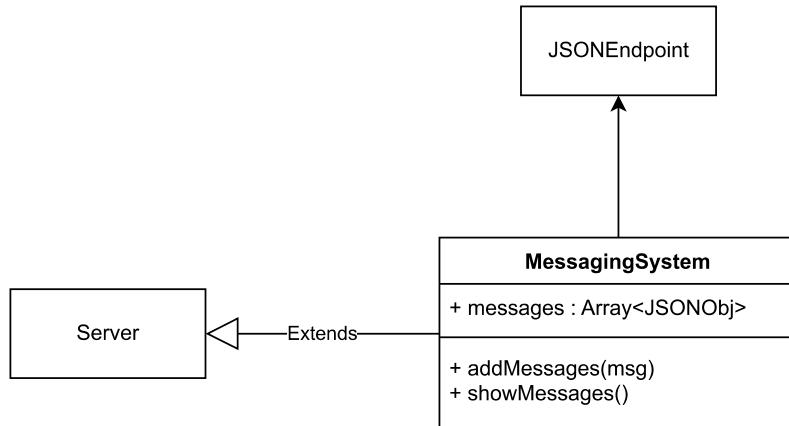
JSONIO



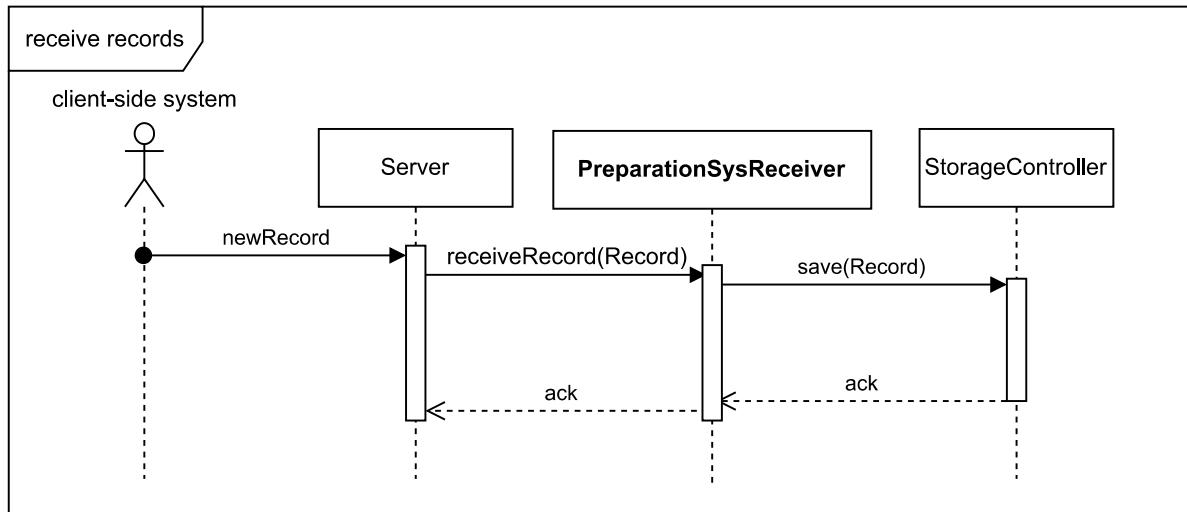
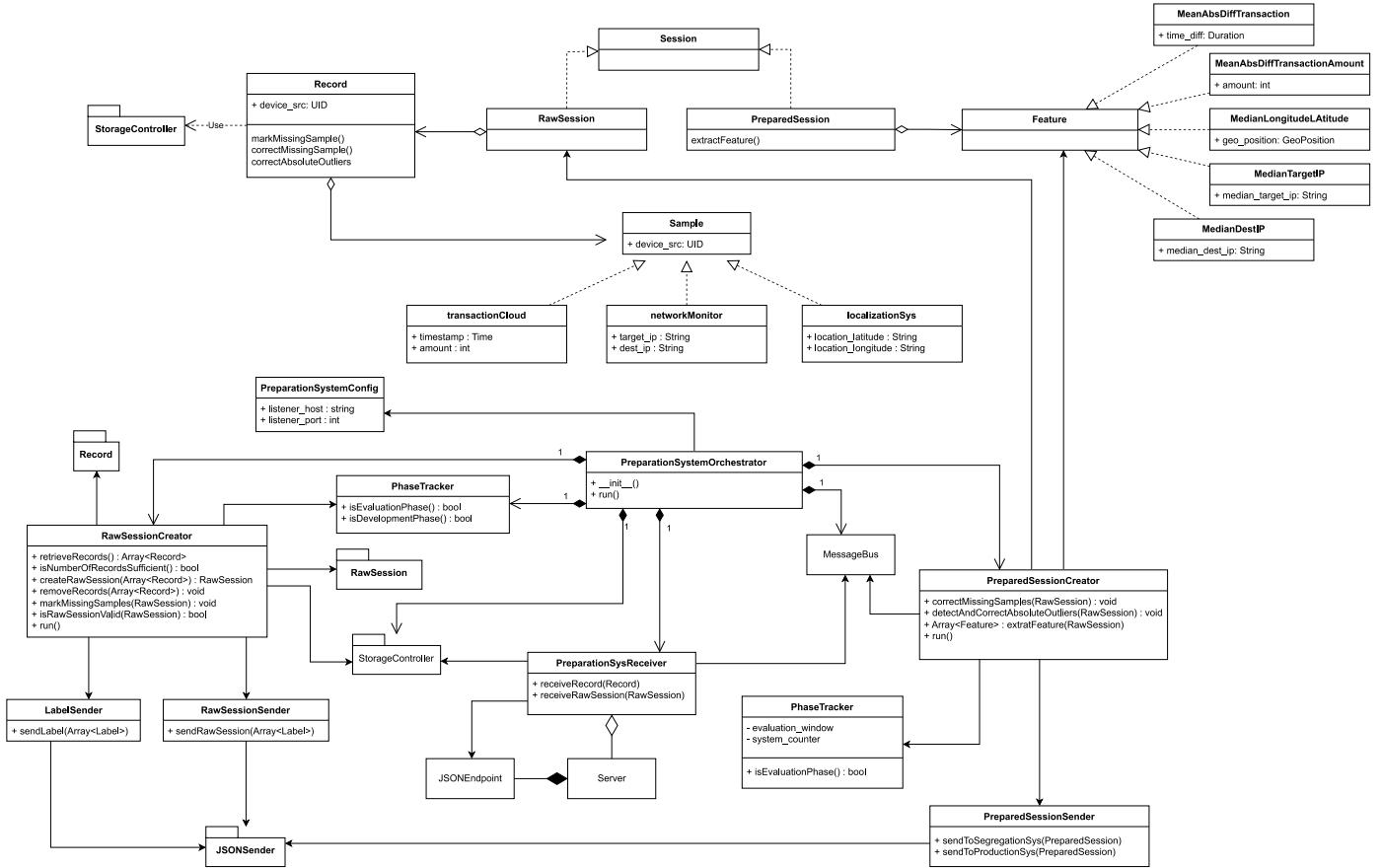
Storage Controller

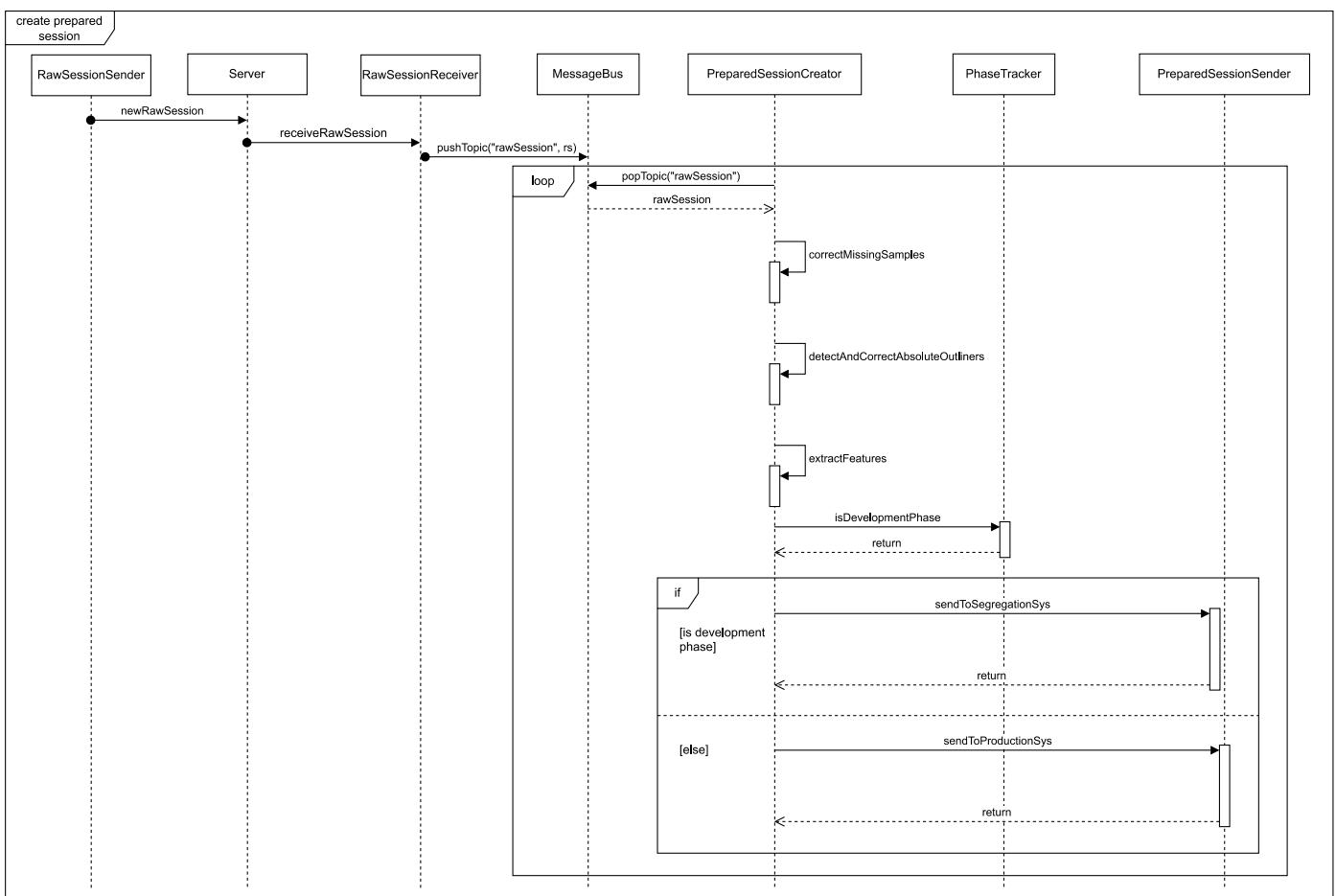
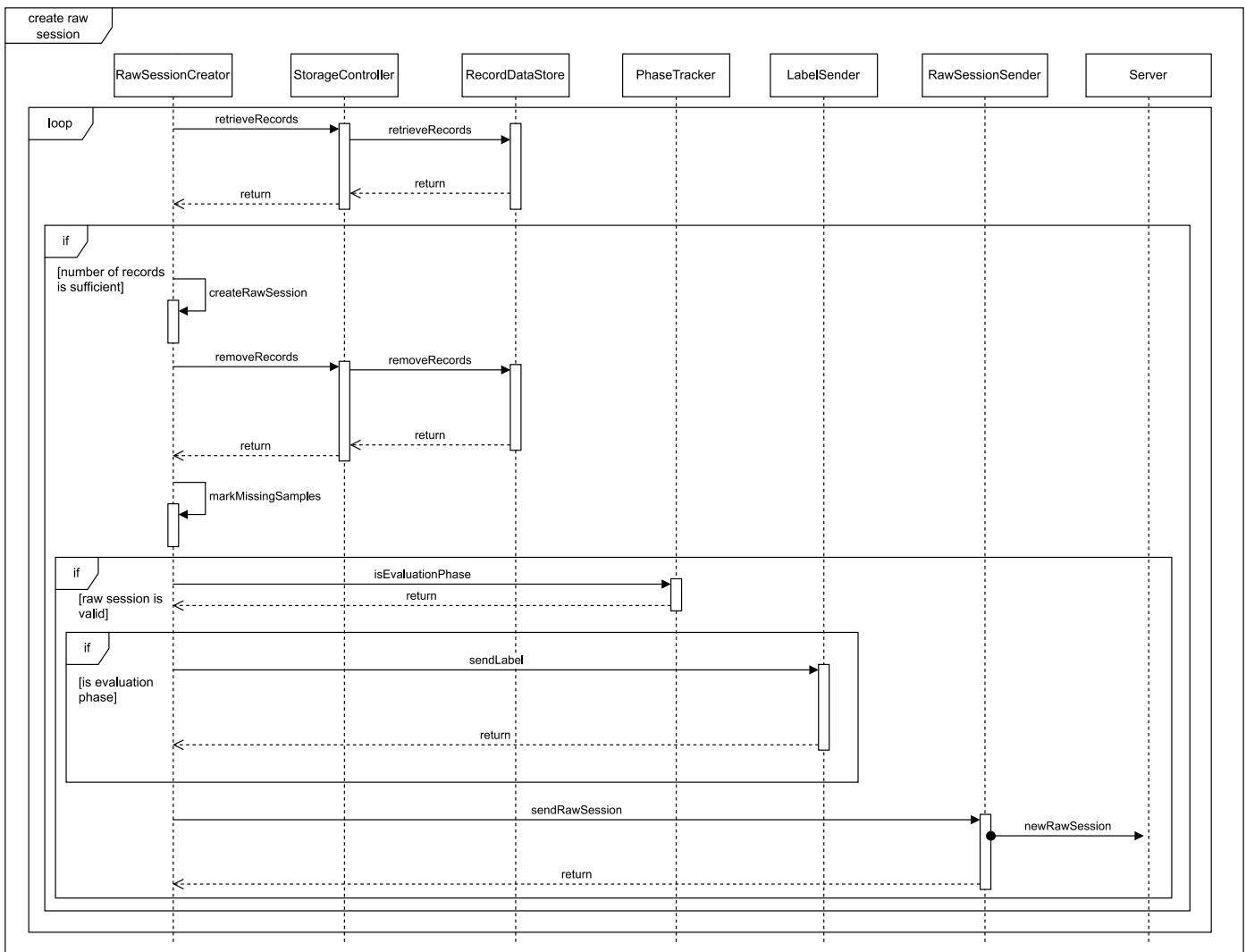


Messaging System

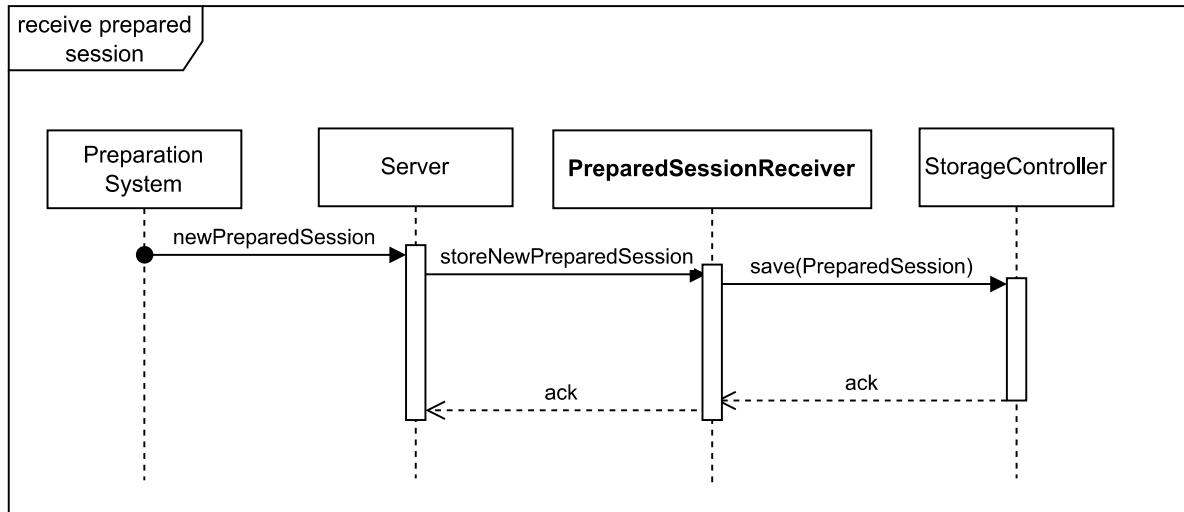
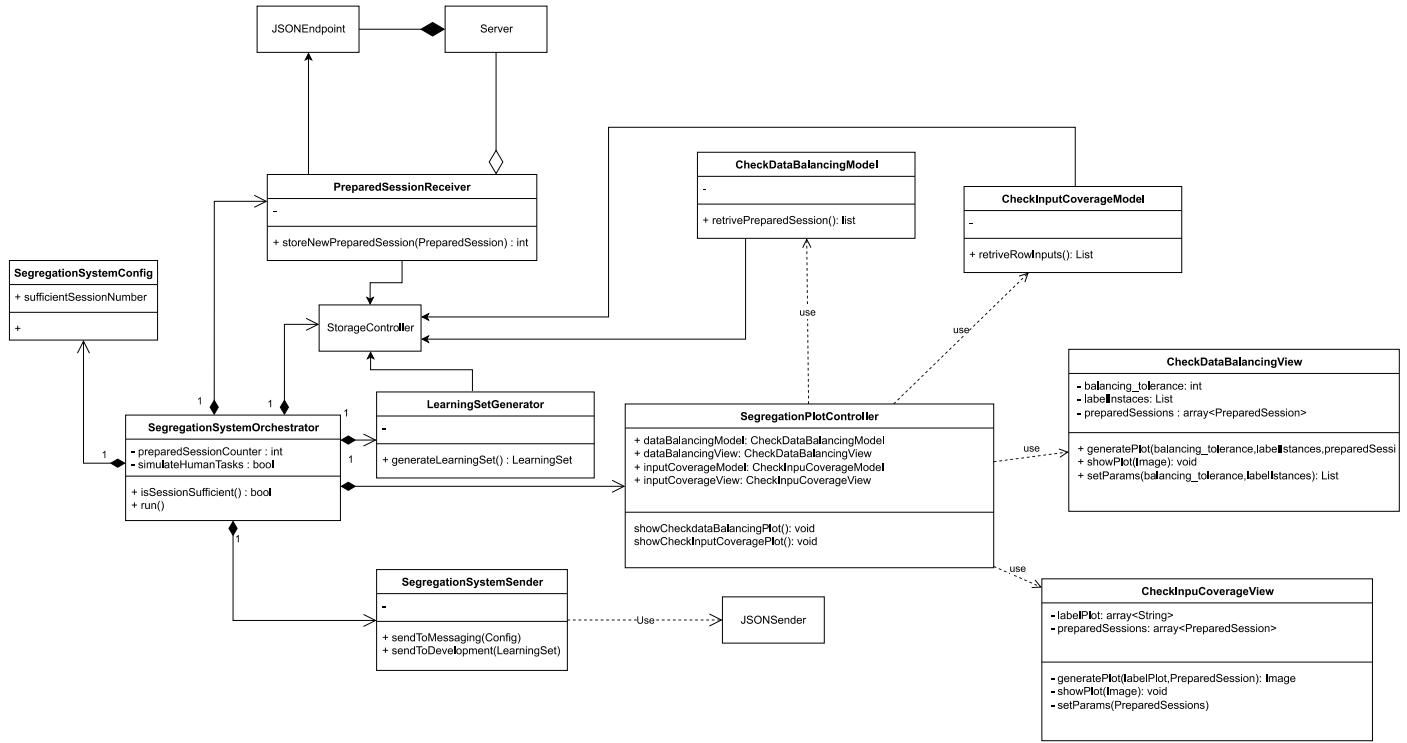


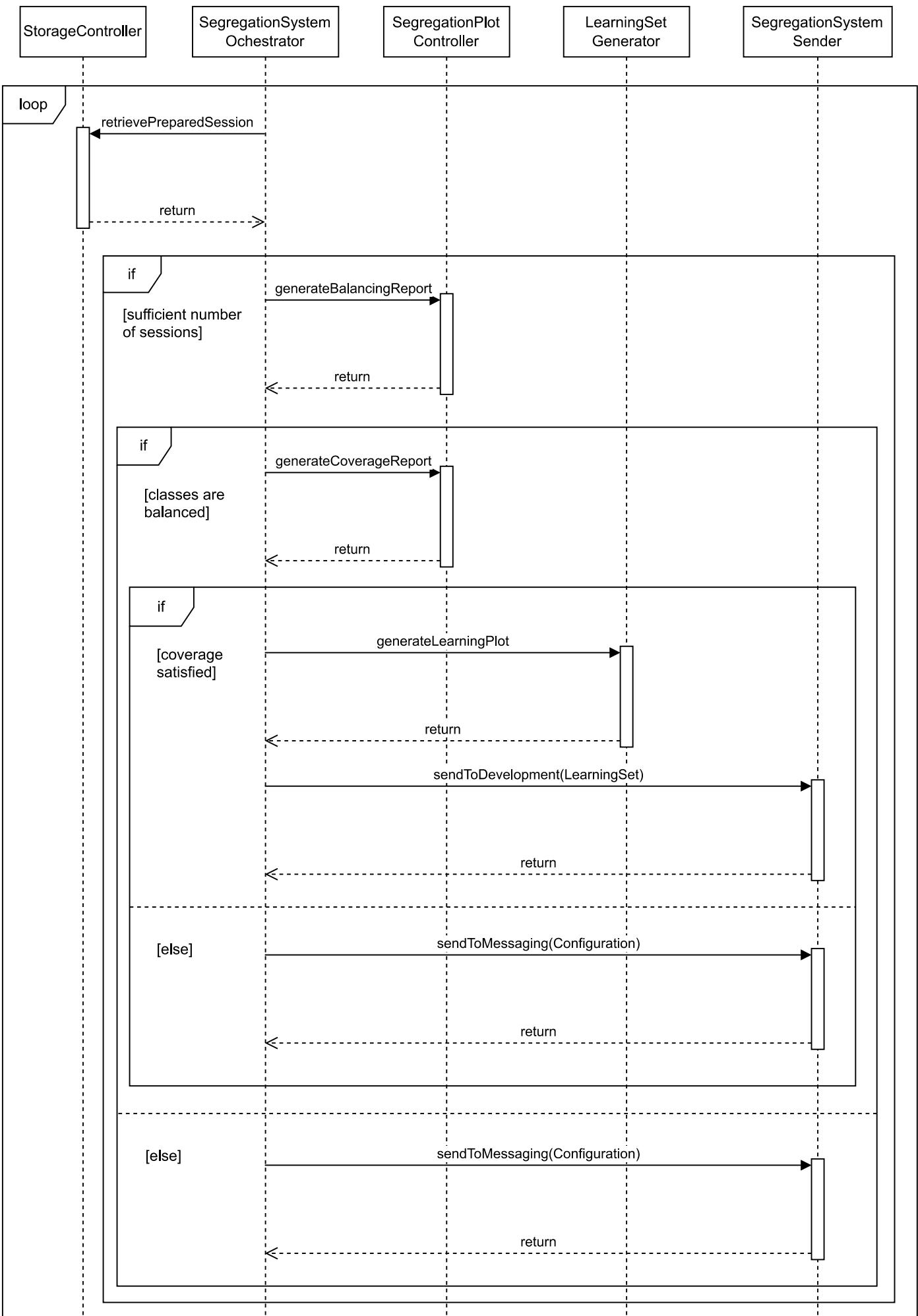
Ingestion & Preparation system



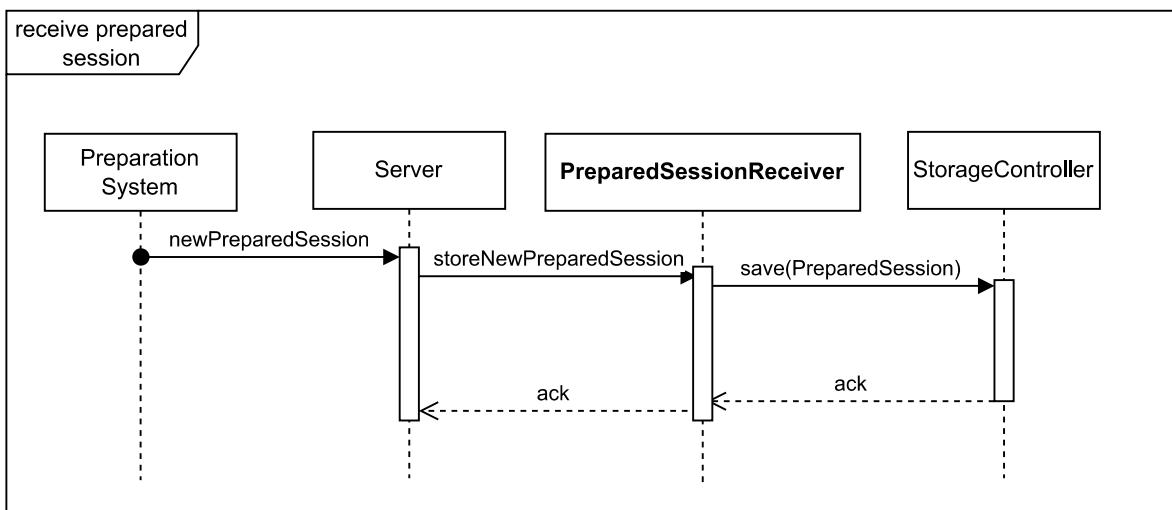
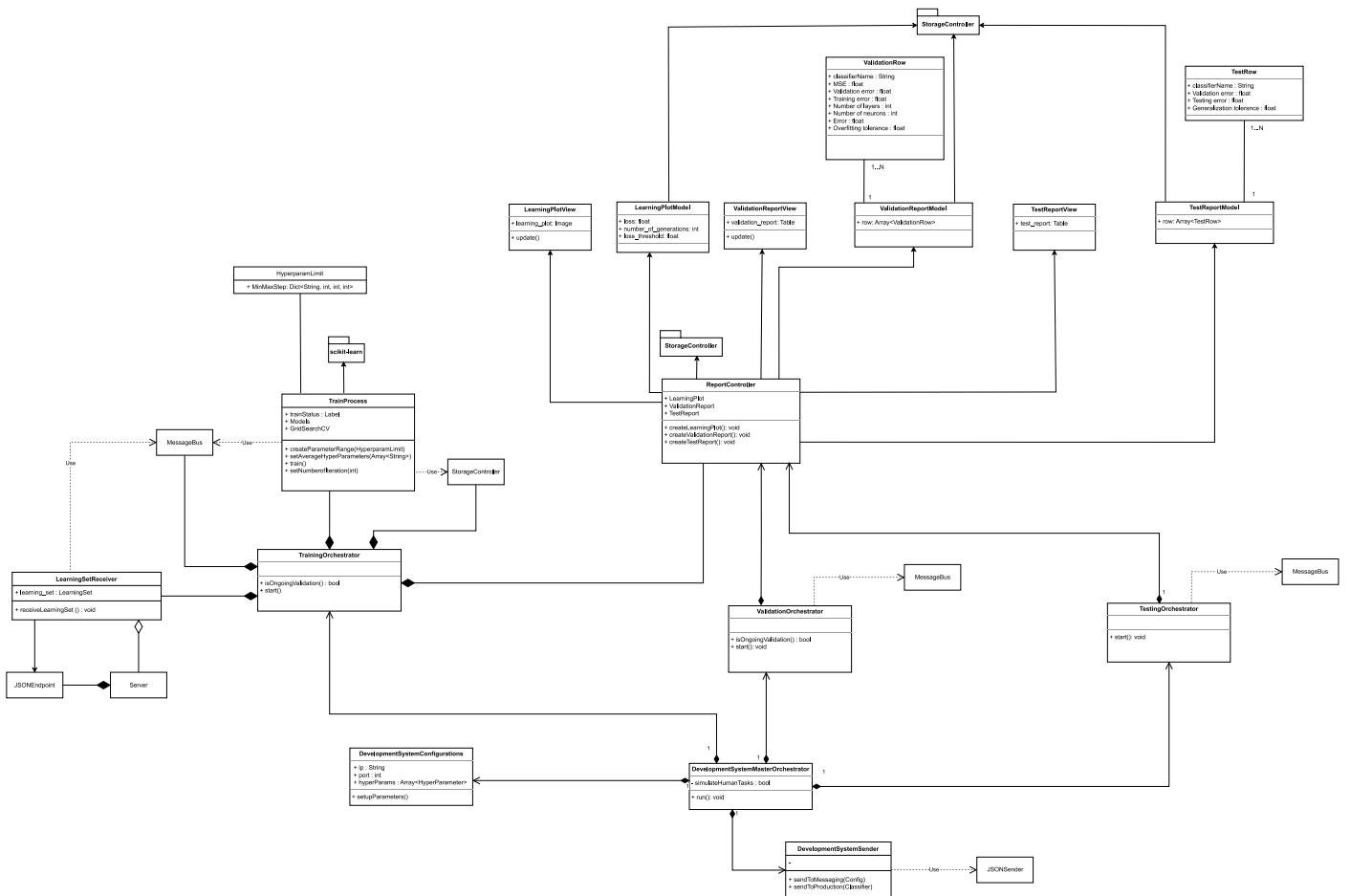


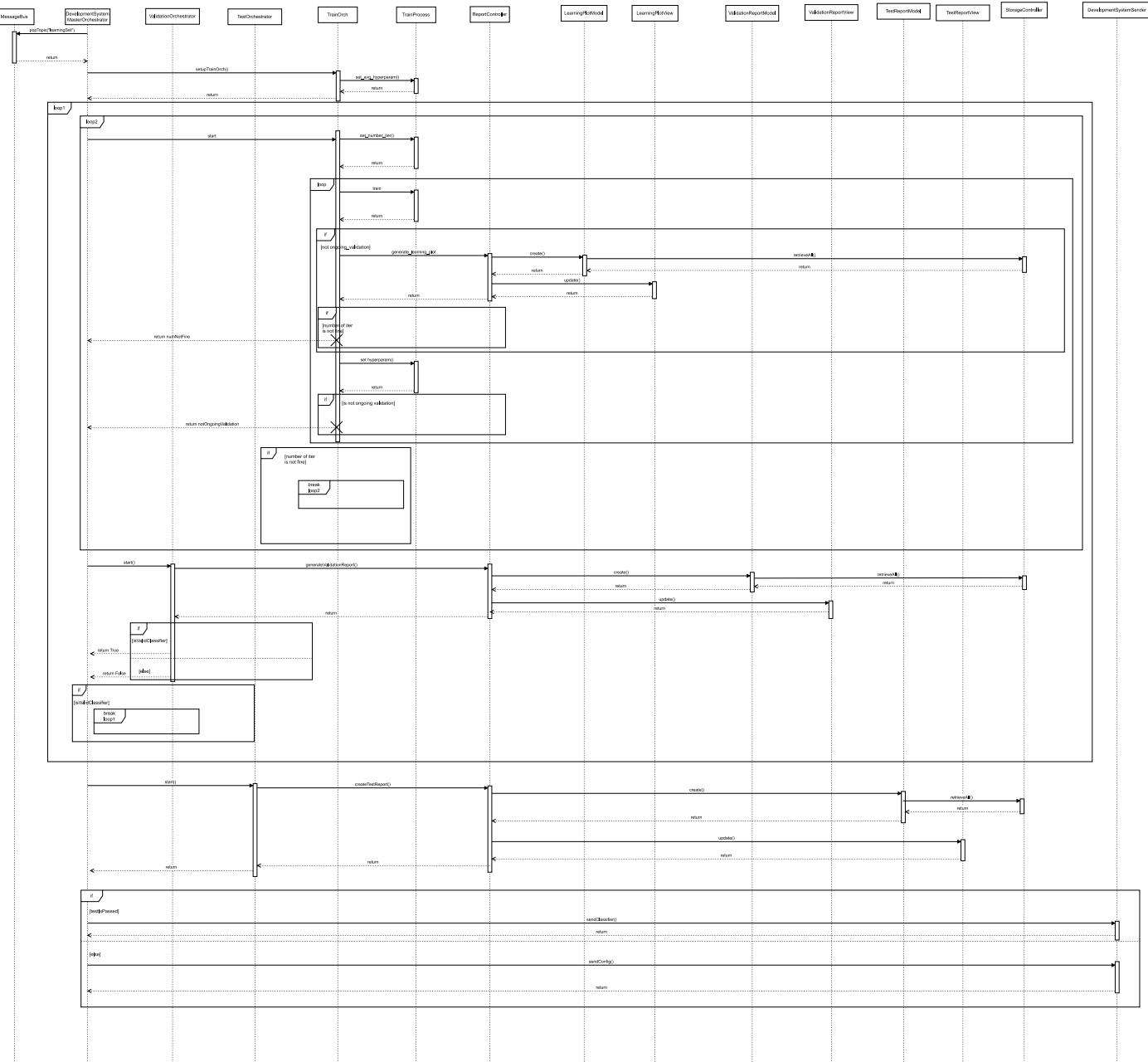
Segregation system



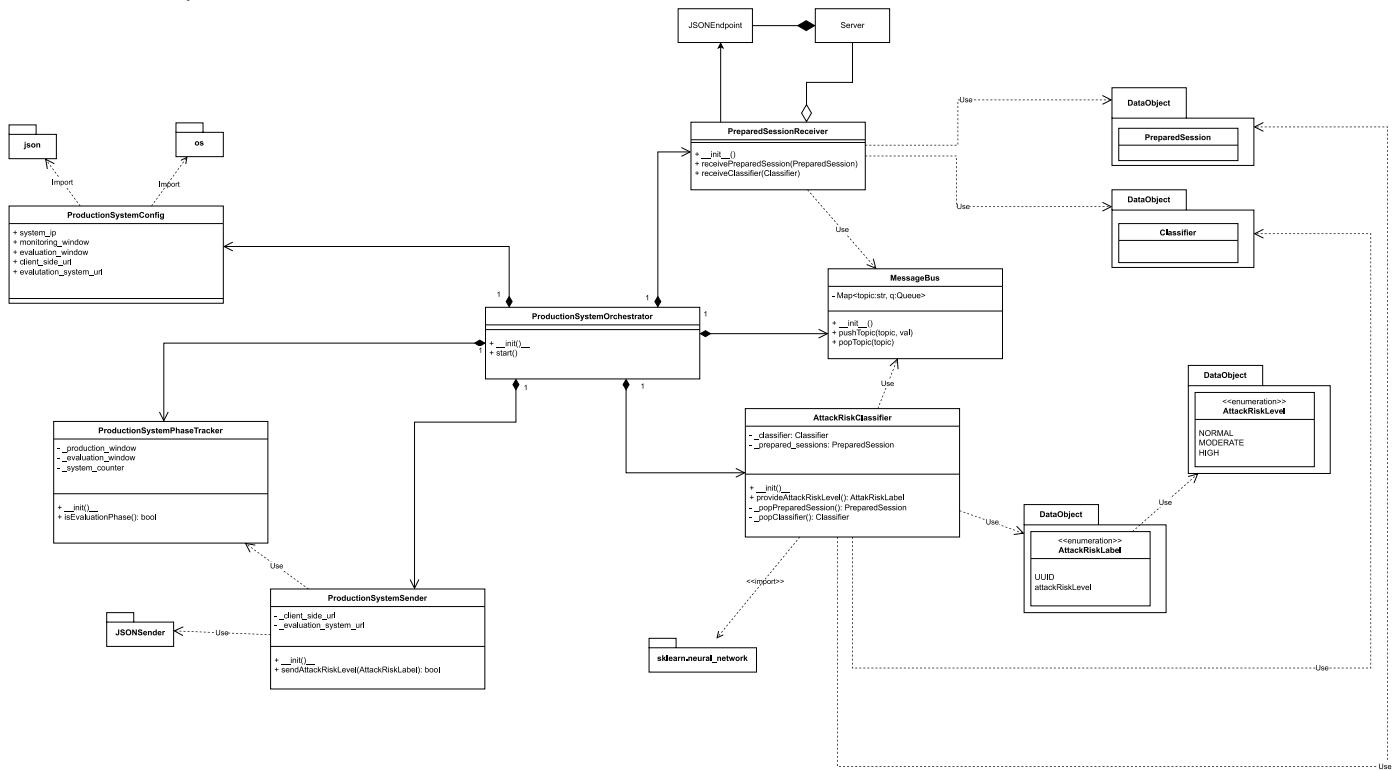


Development system

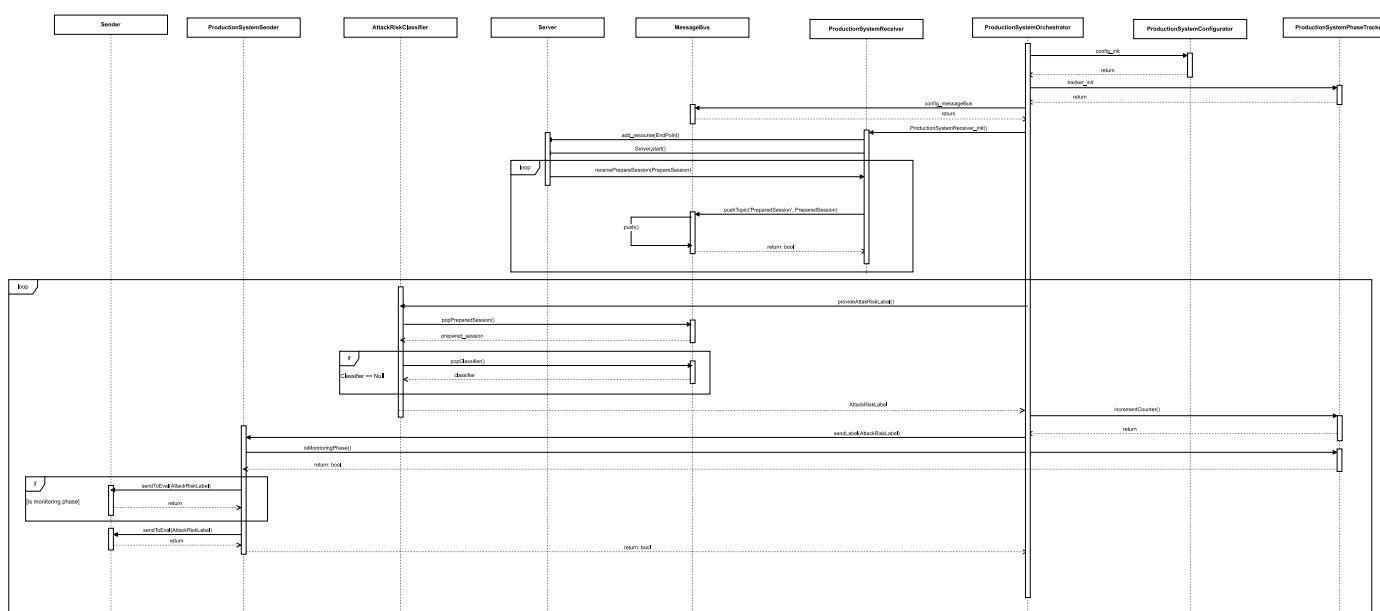




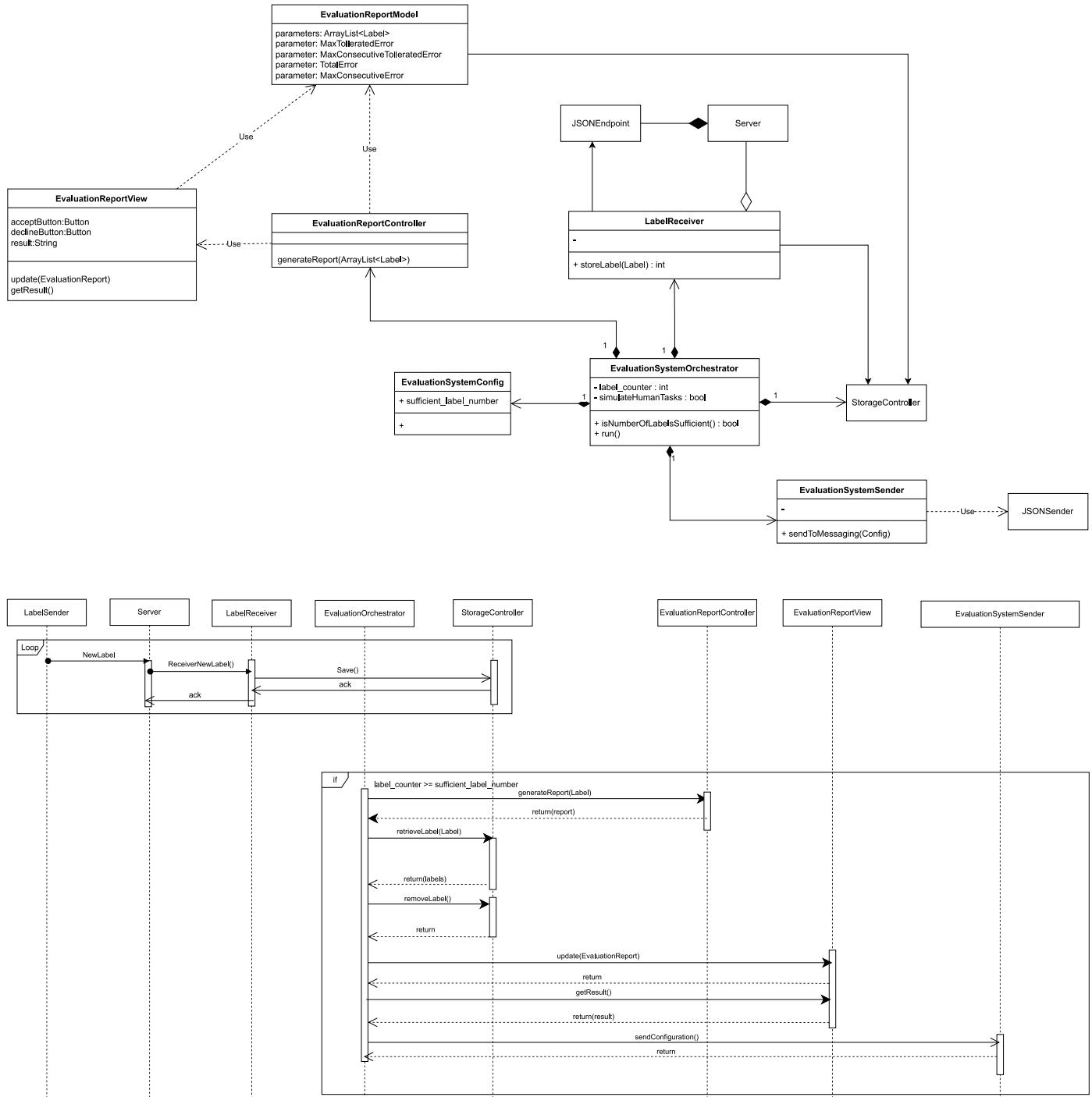
Production system



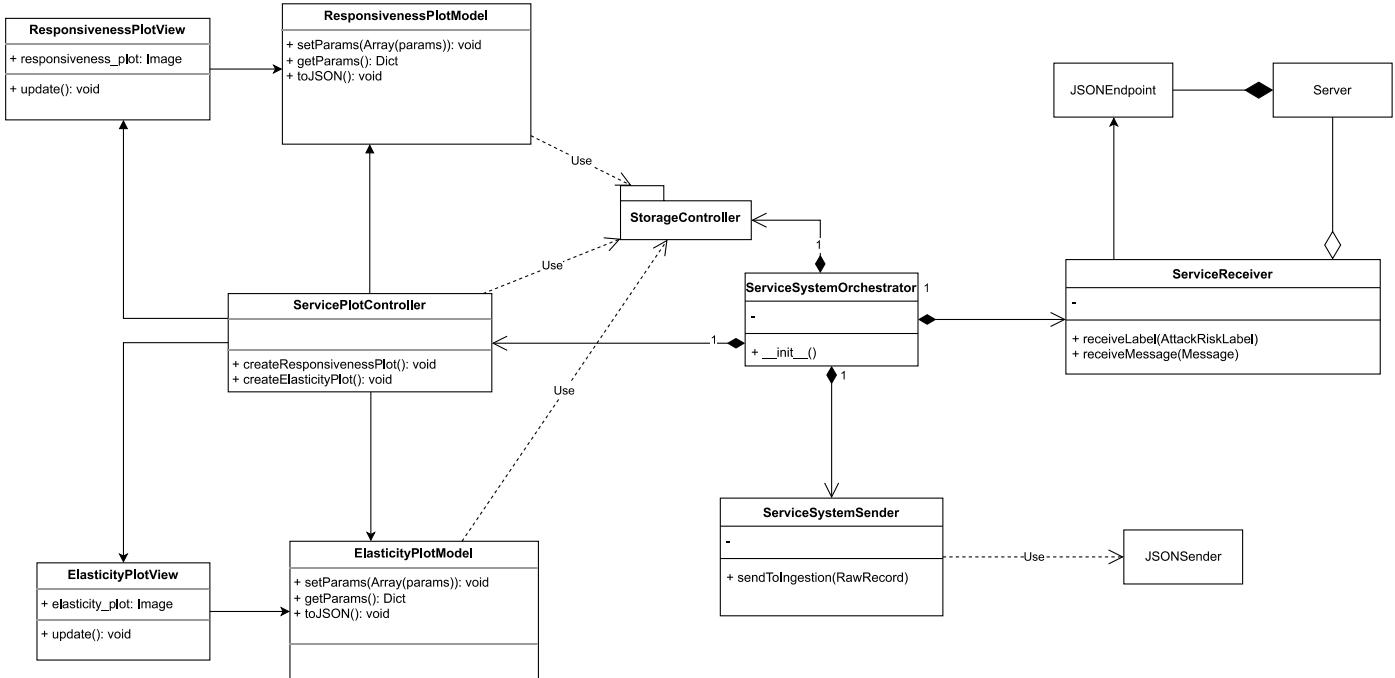
NOTE: This graph only accounts for the PreparedSession reception, the same procedure is to be made for the classifier but without the loop.



Evaluation system



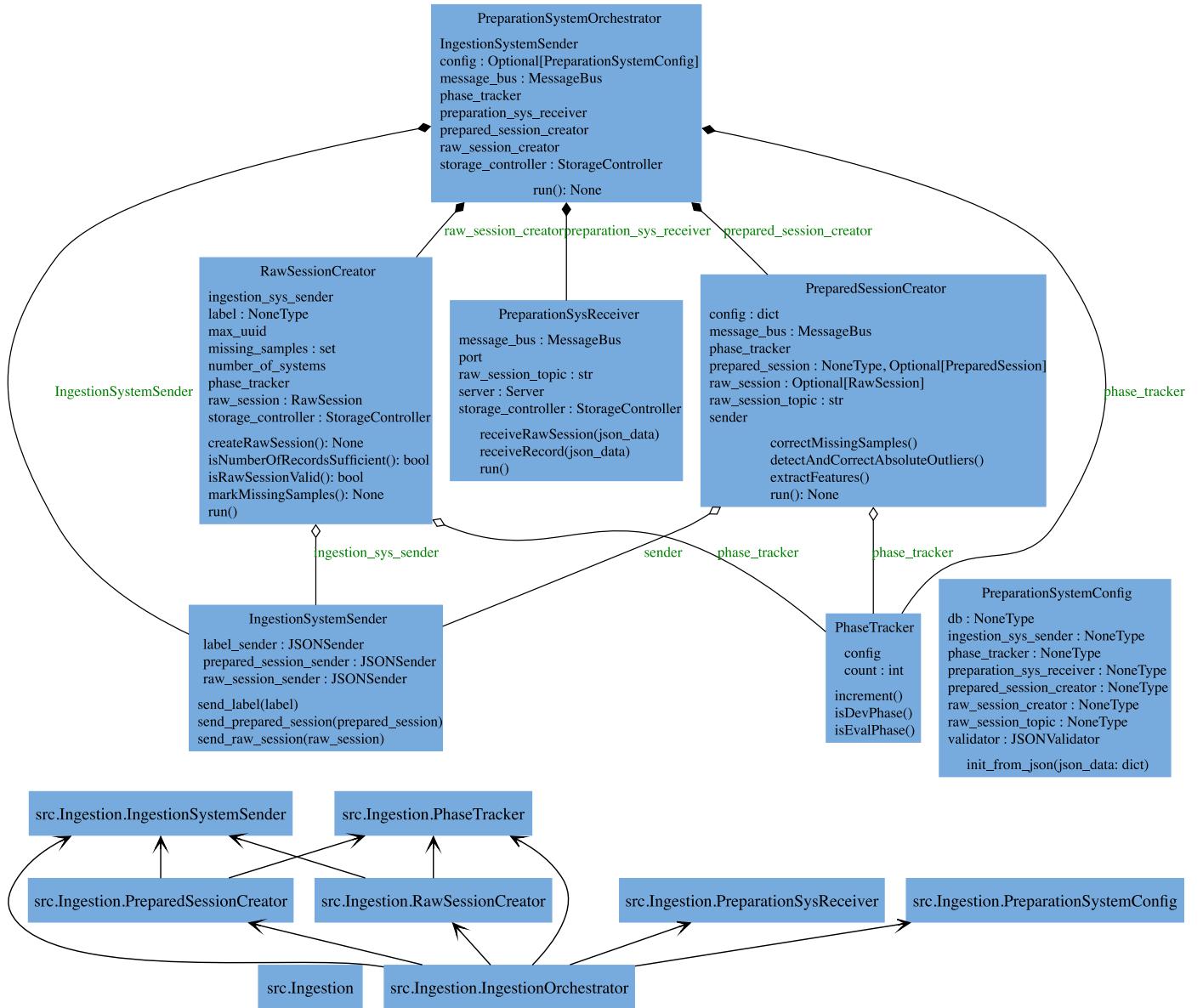
Service System



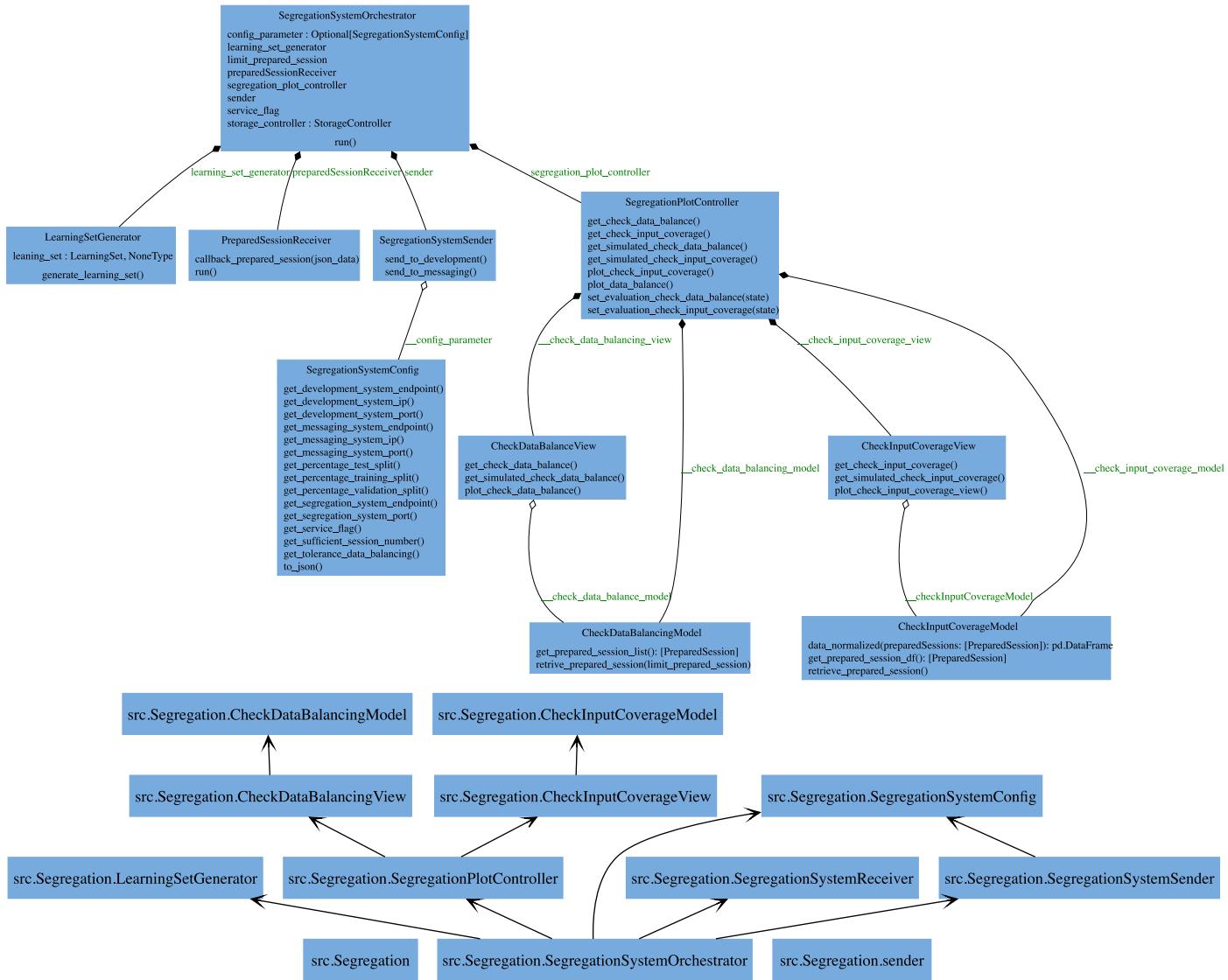
Design

UML Class Diagrams

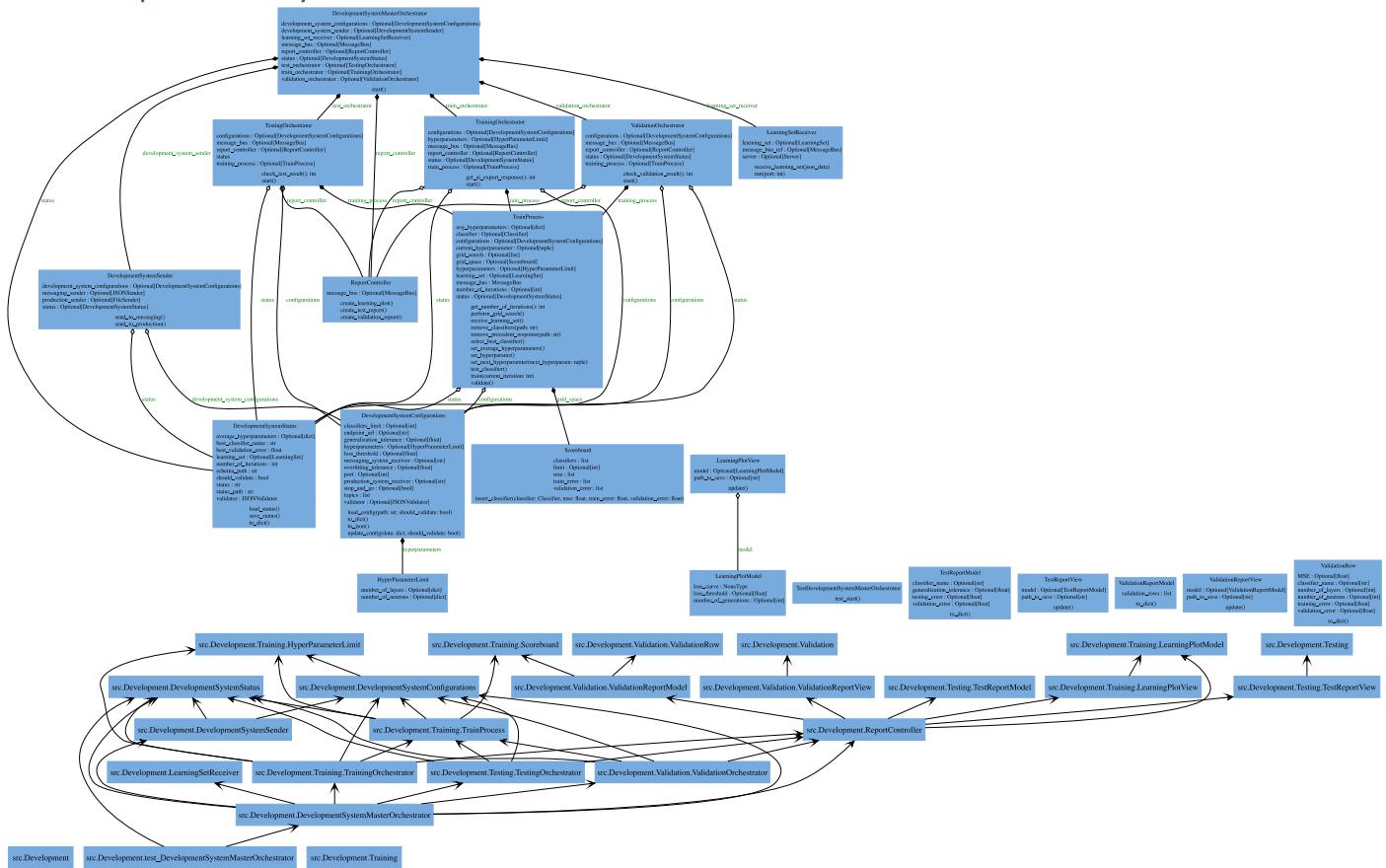
Ingestion System



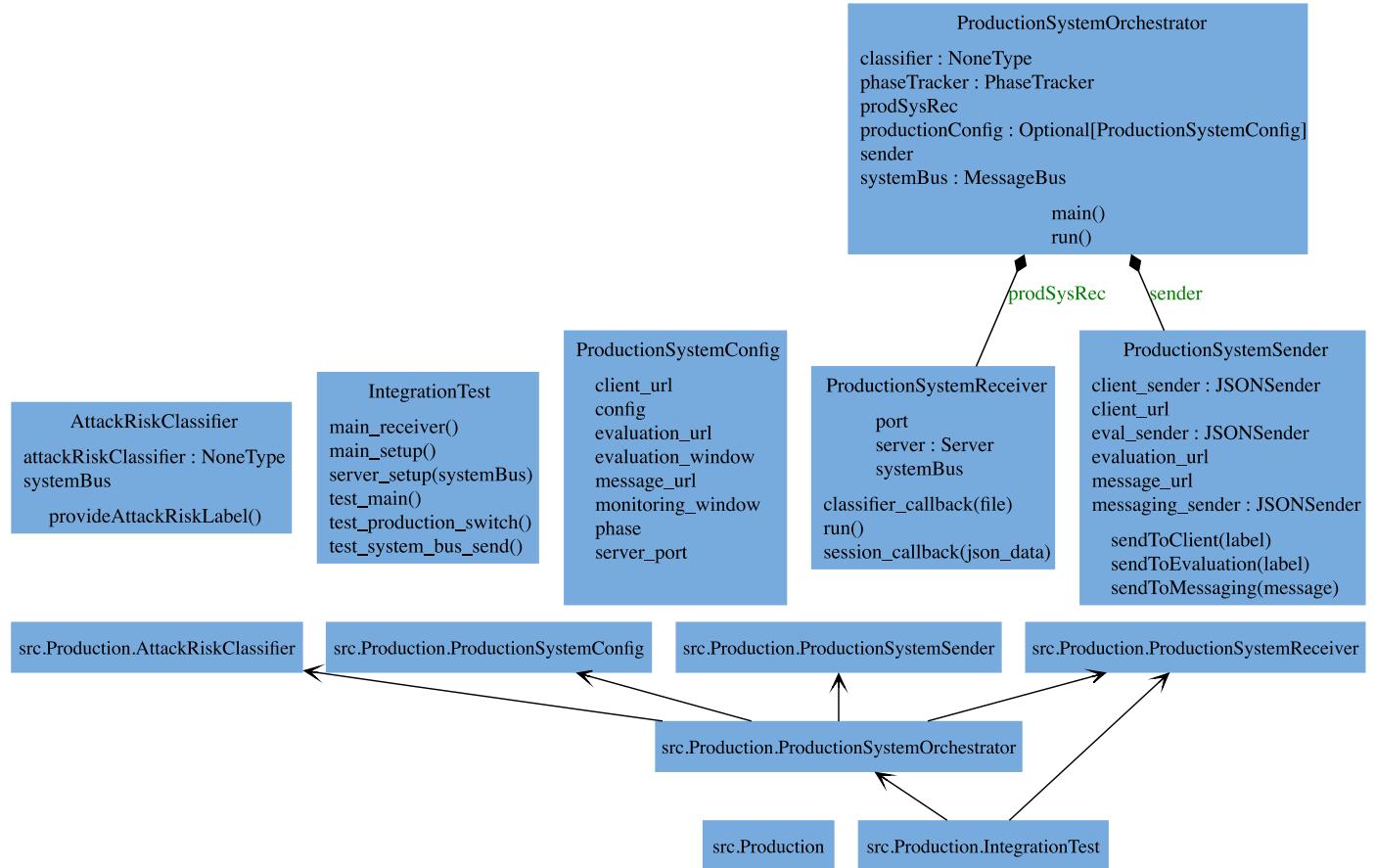
Segregation System



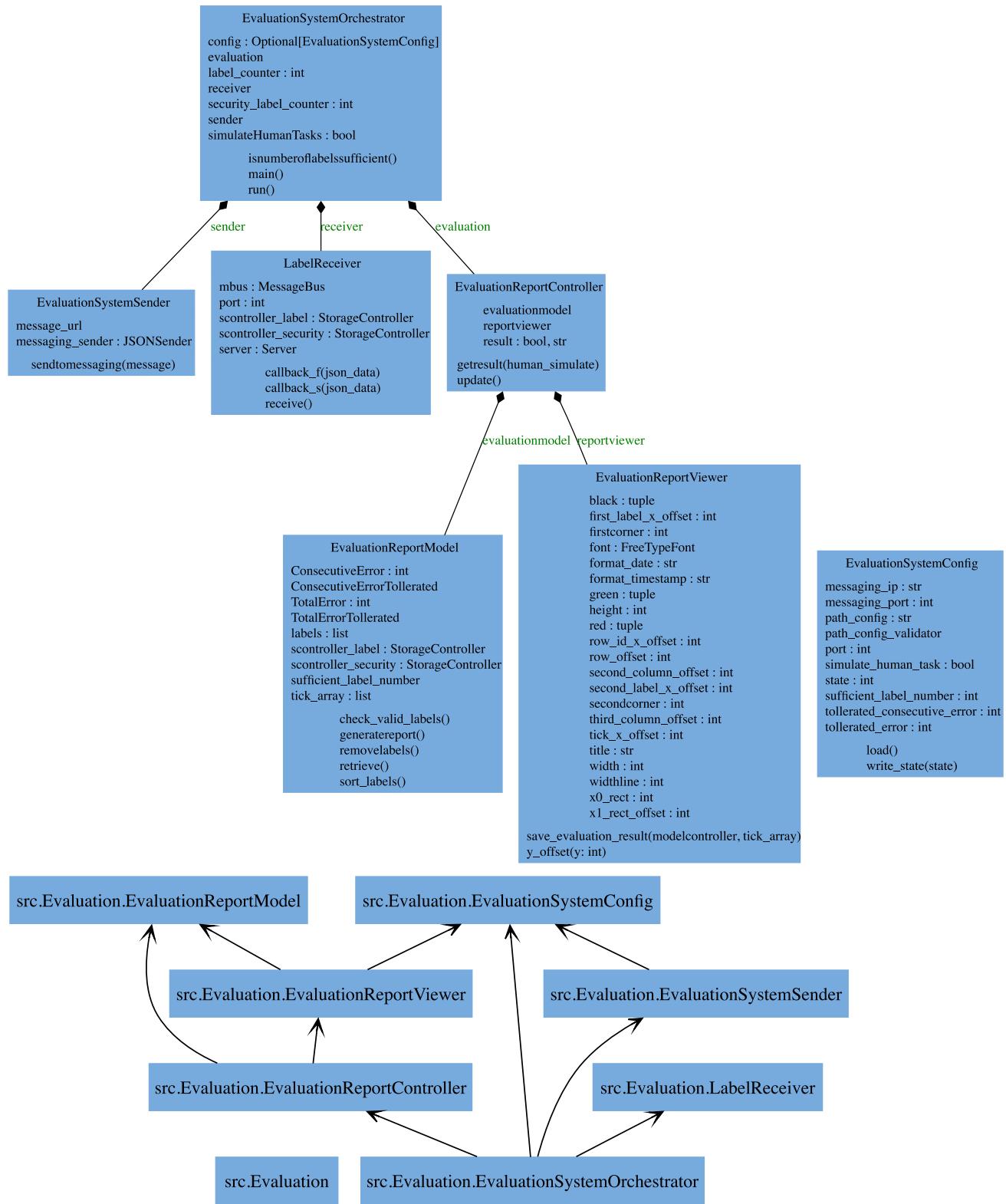
Development System



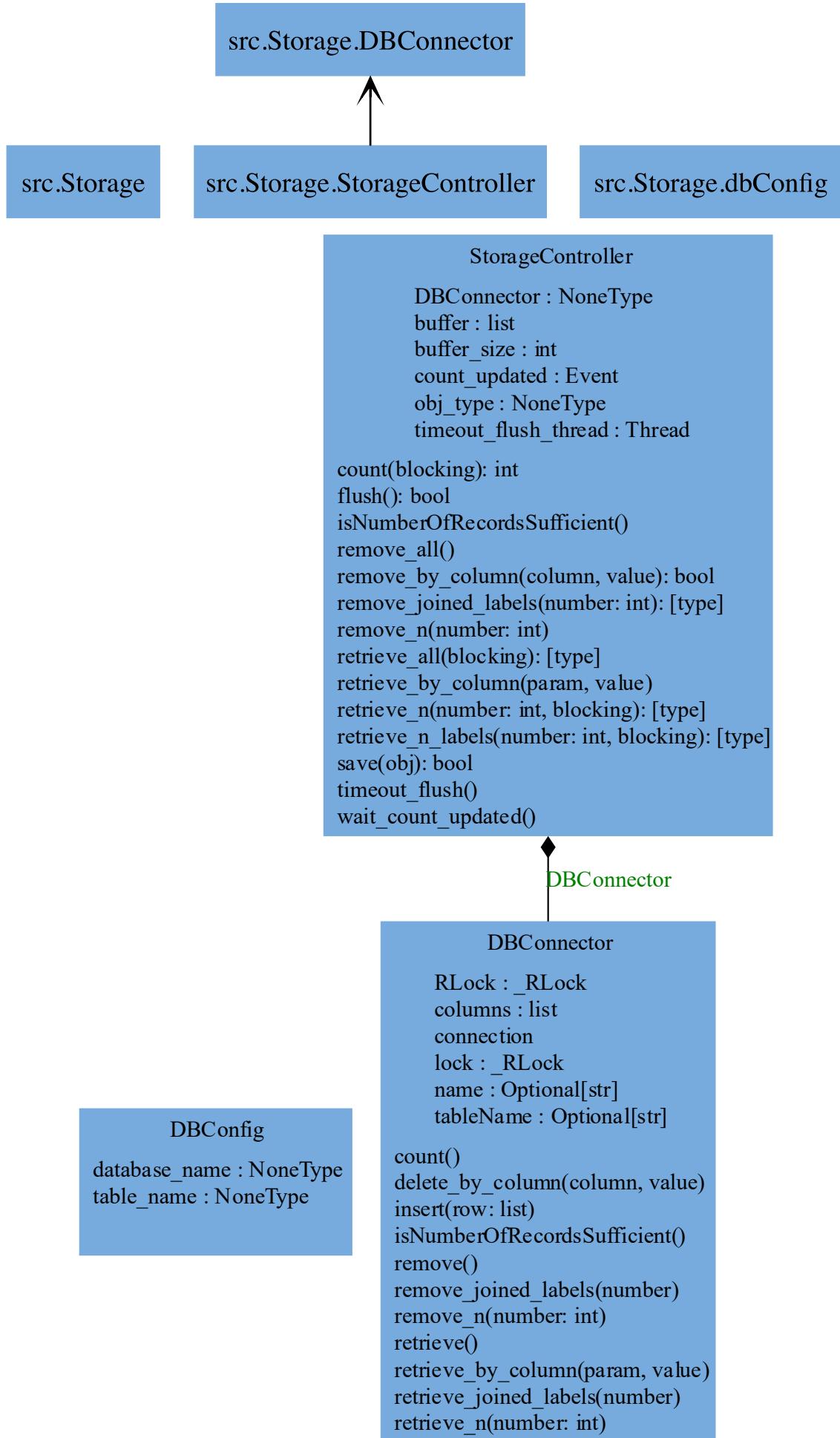
Production System



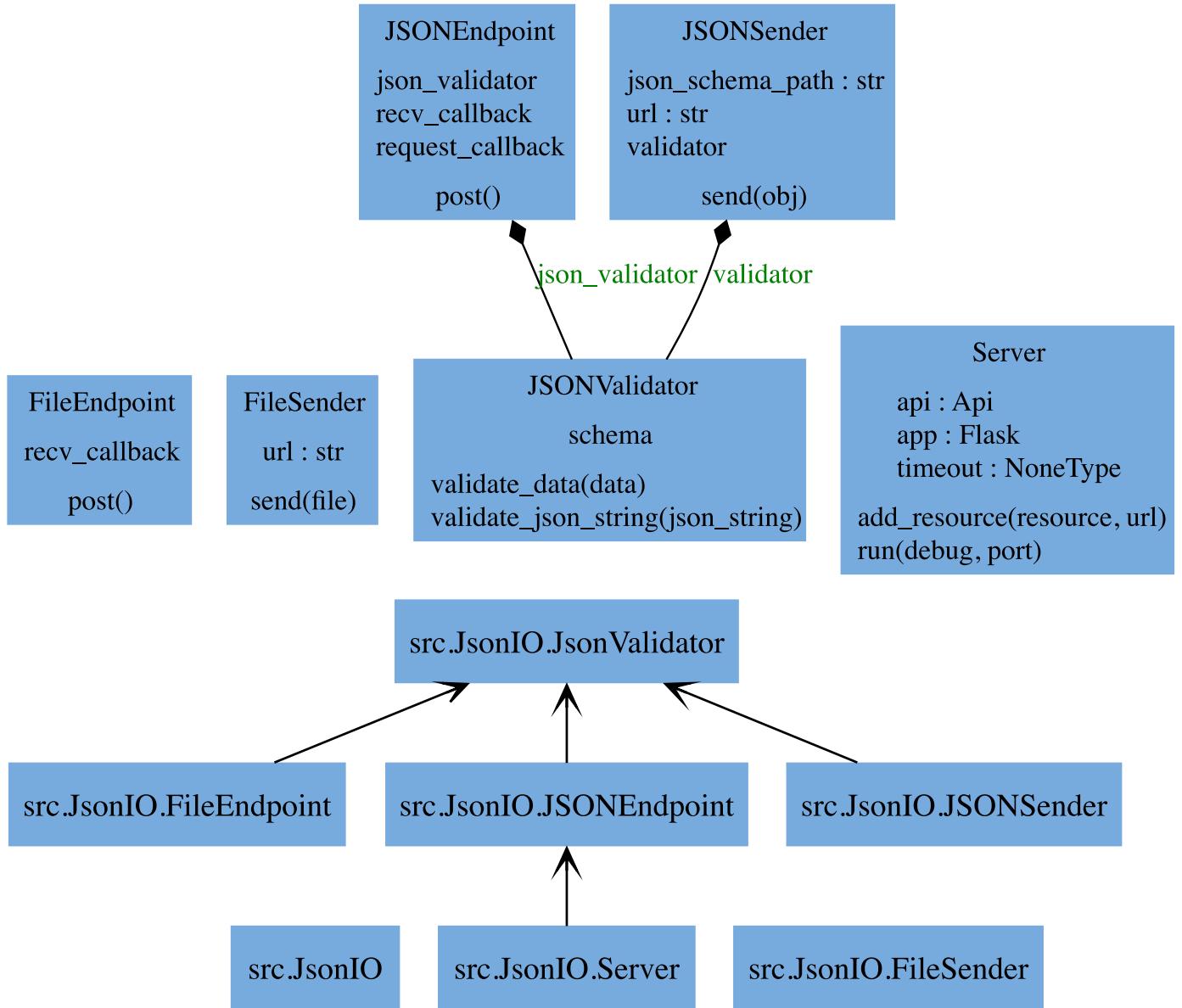
Evaluation System



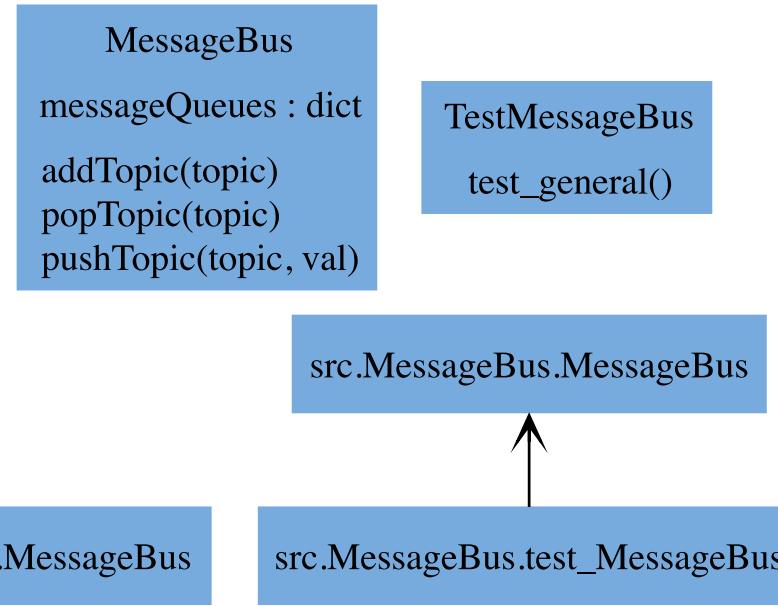
Storage



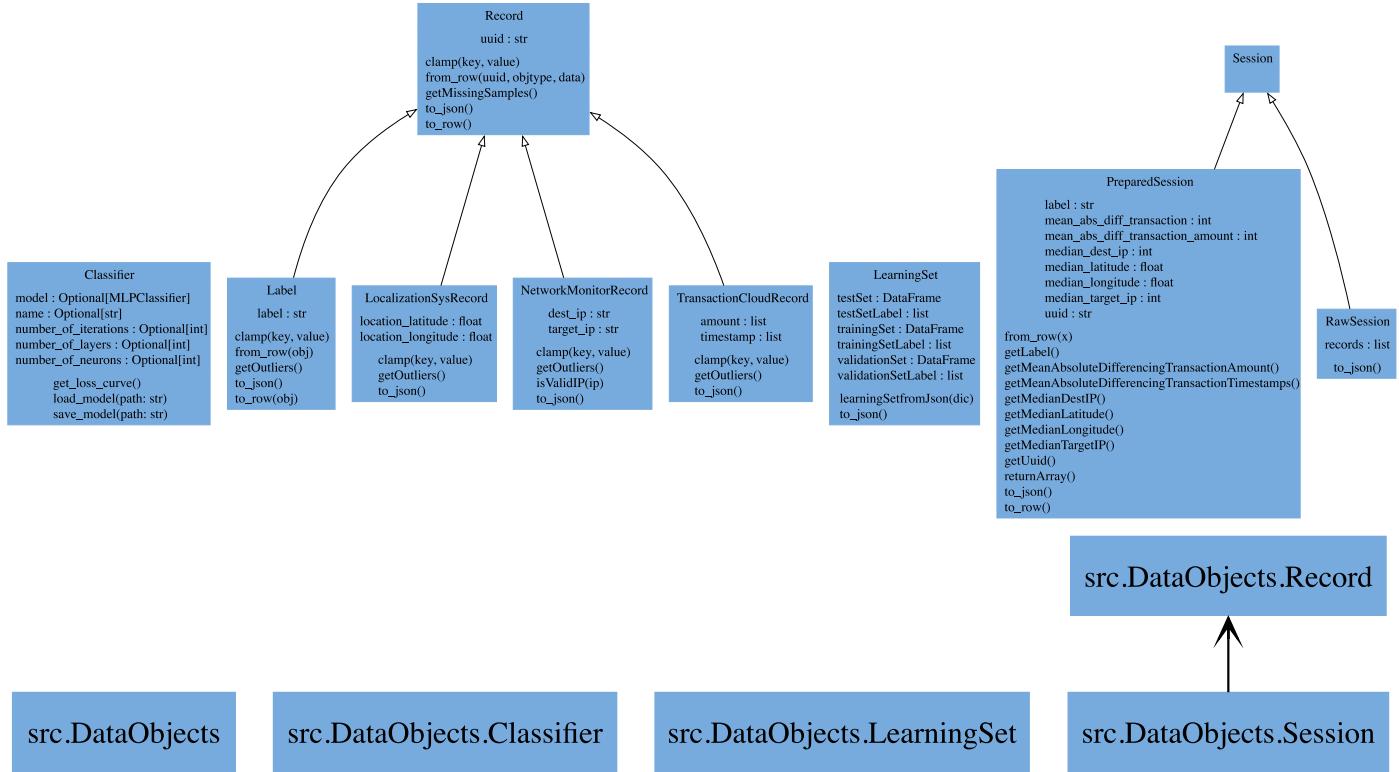
JSONIO



Message Bus



Data Objects



Service

```
Service

config
development_system : DevelopmentSystemMasterOrchestrator
df : NoneType
evaluation_system : EvaluationSystemOrchestrator
ingestion_system : PreparationSystemOrchestrator
message_bus : MessageBus
messaging_system : Server
production_system : ProductionSystemOrchestrator
segregation_system : SegregationSystemOrchestrator
server : Server

client_callback(json_data)
load_data()
messaging_system_callback(json_data)
performance_request_callback(request)
run()
send_data()
setup_client_listener()
setup.messaging_listener()
start_clientside_server()
start_development_system()
start_evaluation_system()
start_factory()
start_ingestion_system()
start.messaging_server()
start_production_system()
start_segregation_system()
```

src.Service

src.Service.Service

Message
msg : str
to_json()

PerformanceSample
class_name
function_name
timestamp
to_json()

Test Automation

In the table below is represented the score for each cognitive effort: remember, understand, apply, analyze, evaluate. The score is evaluated by multiplying a score for each effort from 1 to 6 by the number of steps of the use case, and then multiplying it again for the percentage of occurrence if there are any, otherwise it is multiplied by 100.

$$\text{Cognitive effort score} = \text{score} \cdot \# \text{ of steps of use case} \cdot \% \text{ of occurrence}$$

The following table contains the values of each factor showed in the precedent formula:

	Number of steps of use case	Percentage of positive occurrence	TOTAL
Check data balance	4	90	360
Check input coverage	5	90	450
Set number of iterations	3	100	300
Check learning plot	4	90	360
Check validation result	5	70	350
Check test result	4	90	360
Evaluate classifier	4	90	360

The following table show the total cognitive effort score, in bold there is the score from 1 (low cognitive effort) to maximum 6 (high cognitive effort):

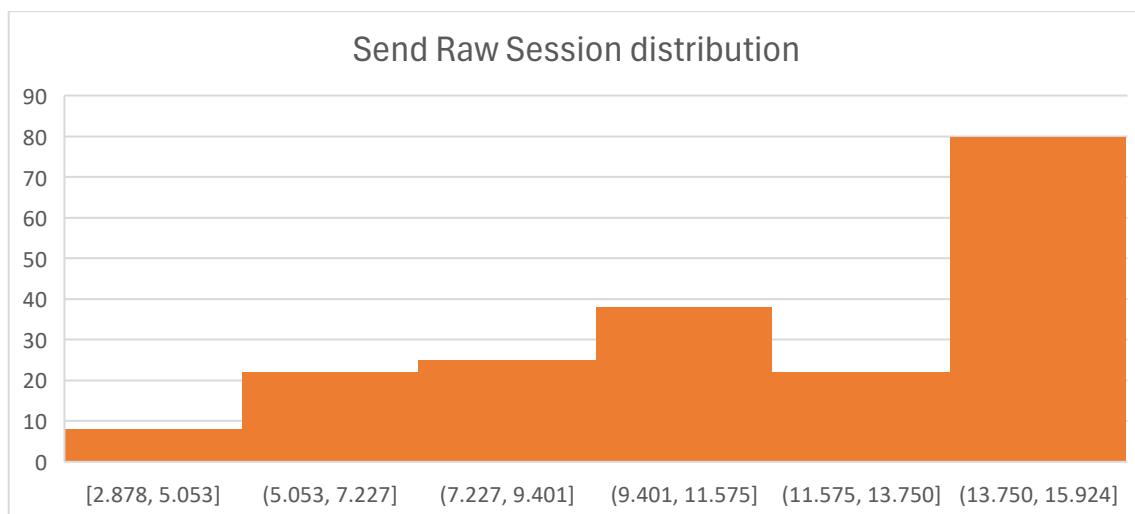
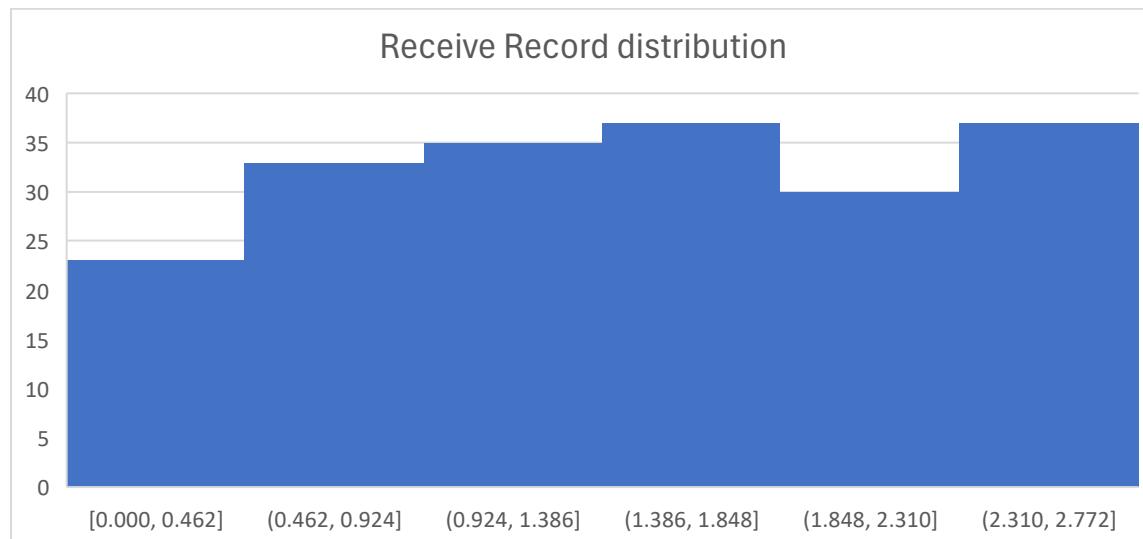
	Check data Balance (360)	Check input coverage (450)	Set number of iterations (300)	Check learning plot (360)	Check validation result (350)	Check test result (360)	Evaluate classifier (360)
Remember	1*360=360	1*450=450	5*300=1500	1*360=360	1*350=350	1*360=360	1*360=360
Understand	5*360=1800	4*450=1800	1*300=300	2*360=720	4*350=1400	4*360=1440	6*360=2160
Apply	1*360=360	1*450=450	1*300=300	1*360=360	5*350=1750	2*360=720	2*360=720
Analyze	1*360=360	6*450=2700	1*300=300	5*360=1800	1*350=350	3*360=1080	1*360=360

Responsiveness development

Important note: with the * are marked the BPMN path that requires a human intervention, meaning that the calculated time is a lower bound since we do not consider the human intervention. In addition, the net delay is not considered. If the occurrence rate is 100% the time * rate is not calculated.

	Is number or record sufficient ~75%	Failed is raw session valid ~15%	Send raw session 100%	Send Prepared Session 100%
Ingestion	0.0163 (0.0122)	0.651 (0.097)	3.542	--
Preparation	--	--	--	3.676

The following graph shows the distribution of the arrival of records at random, the times in the previous tables are calculated with the record sent in order (sorted by UUID) to exclude a source of randomness. The second graph shows a more realistic behavior where the raw sessions are very unlikely to be sent at the beginning of the action given the low probability of having enough records.



	Failed is number or session valid	Failed check data balance*	Failed check input coverage*	Send Learning set 100%	Send classifier 100%
	~90%	~75%	~75%		
Segregation	0.0127 (0.0114)	0.349 (0.262)	2.754 (2.066)	9.613	--
Development	--	--	--	--	9.721

Responsiveness production

Since the BPMN have no possible other path for the production system, no percentage are reported, in addition since the time for generating an output may vary on the “structure” of the input we decided to calculate the frequency, meaning the average time to produce an output given a generic input.

	Receive prepared session	Output labels
Time	43.840	43.782
Frequency	0.229	0.229

For simplicity the two input of the Evaluation system are into merged into a single one

	Failed: is number of label sufficient	Classifier passed the check* 50%	Send configuration 50%
	~78%		
Evaluation	0.148 (0.115)	0.269 (0.135)	0.269 (0.135)

Elasticity development

In the following table there are the values for the Development Phase run. There is a column for a stressed run that means that there were other tasks in background to simulate load work.

	Non-stressed run	Stressed run	Difference	Ratio
Ingestion	51,43 s	62,61 s	11,18 s	1,21
Preparation	51,52 s	62,8	11,28 s	1,21
Segregation	45,16 s	58,7	13,54 s	1,3
Development	9,72 s	11,39	1,67 s	1,17
Production	0,27 s	0,35	0,08 s	1,3
TOTAL	158,1 s	195,85 s	37,75 s	1,24 (mean)

Elasticity production

In the following table there are the values for the Production Phase run. There is a column for a stressed run that means that there were other tasks in background to simulate load work.

	Non-stressed run	Stressed run	Difference	Ratio
Ingestion	0,13 s	0.13 s	0 s	0.94
Preparation	4,29 s	3,56 s	0.73 s	1,20
Prep.+Ing.	4,43s	3,69 s	0.73 s	1.20
Production	0,97 s	0,35 s	0.62 s	2.76
Evaluation	9,07 s	7,52 s	1.55 s	1,20
TOTAL	93,25 s	38.05 s	55,20 s	2,45 (mean)

Resiliency

In the following table there are the score for each input situation

	Score level
Missing record	5
Late record	1
Unbalanced classes	3
Malformed record	4
Unbalanced coverage	3
Inconsistent record	1
Malformed configurations	4
TOTAL	21

Inter-Operability

Actor \ S/R	Ingestion	Preparation	Segregation	Development	Production	Evaluation	Client
Ingestion	--	0	--	--	--	1	0
Preparation	0	--	0	--	0	--	--
Segregation	--	0	--	0	--	--	--
Development	--	--	0	--	0	--	--
Production	--	1	--	0	--	0	0
Evaluation	0	--	--	--	0	--	--
						Total	2

The inter-operability between systems is considered for each message exchanged from an Actor system to a (S)ender (coming before) or a (R)eceiver (coming after). Intersections marked with '-' refer to the absence of communication between systems. The only notable cases where the value is not zero are for Ingestion system sending the ground-truth label to Evaluation and for Production receiving the prepared session from Preparation. In the first case, Ingestion does not consider the rest of the raw session for the label to be sent, while in the latter case Production does not consider the ground truth label, if present, because it will substitute it with the label predicted from the classifier.