Interreg

EUROPEAN UNION

Balkan-Mediterranean

SMEINNOBOOST

# „Software development consultancy for Internet-based decision-making support system for the creation of innovative clusters“, project “SME Innovation Capacity Boost”, SMEInnoBoost, co-financed by the European Program „Interreg Balkan-Mediterranean 2014-2020”

## Contents

## Main objective of the development

Establishment of an Internet-based system to support Small and Medium Enterprises from the Republic of Bulgaria, the Former Yugoslav Republic of Macedonia, the Republic of Albania, the Republic of Greece and the Republic of Cyprus with regard to:

1. Creating a database of SMEs with intentions for innovations;
2. Supporting decision-making to build international clusters of SMEs providing co-operation in innovation / clusters for innovativeness /;
3. Offering solutions for improvement of the inovativness of SMEs on the basis of evaluatin of their innovativness and multicriteria analysis.

## Operational characteristics of the Internet-based system.

1. Building a WEB-based programming system;
2. The system should consist of three parts:

- A brochure with informational purpose accessible to all visitors;
- A portal containing the features of item 1 and accessible only to registered users.
- An administrative part containing the functions of item 3 accessible to the Main Administrator and partly by the specialized administrators.

3. The organization of an Internet-based decision-support system for the creation of innovative clusters by SMEs should allow the system to be modified and expanded by adding new subsystems without changing the existing ones;
4. The system should be focused on providing services;
5. Multilingual User Interface Support;
6. Support for different formats for entering statistics data.
7. Using Open source environments and software for building and running the system;
8. The system must support both a relational database and a nonrelated database designed to support the Big data;
9. The system must maintain at least 100,000 registered enterprises;
10. Portability of the system on different platforms;
11. Scalability;
12. Maintenance and changes to the system for 6 months after termination of the contract.

## Functional requirements to the Internet-based programming system

1. User Functions / SME /: Company Registration; Sign in and Sign out; Entering of information related to the assessment of the innovativeness of SMEs; Assessment of the possibility of joining a cluster (s) of a registered SME; Offering solutions to improve SME innovation based on multi-criteria or inter-criteria analysis; A simulation tool for playing alternative solutions to improve the innovativeness of SMEs by setting up variable data on innovation; SME communication functions supporting cluster creation.
2. Main Administrator functions: Setting Enterprise Access Rights to Work with the System; Removing Enterprise Access Rights; Set up groups of specialized administrators; Content portal management; Reporting on appropriate clustering of enterprises according to their level of innovation, clustering of SMEs on the basis of innovation criteria and on the activity of each registered SME.
3. Functions of specialized administrators:

- Ensuring different approaches to acquiring and inputting statistical information from the National Statistical Institutes of the countries of the region as well as other sources of statistical information;
- Starting and managing a subsystem to improve the innovation process for SMEs by using methods and approaches for multi-criteria decision-making and choice in defining innovative clusters by SMEs or on the basis of Inter-Criteria Analysis;
- Starting and managing the subsystem for analysis of alternatives (what if analyses).

4. Information functions: Promoting new innovative practices in the Balkan region; Promoting innovative companies and their production; Establishing links between cooperative enterprises (including those outside the Balkan-Mediterranean ).

# Use cases

## Actors

The system will operate with two types of actors - external and internal. .

The external ones operate with the system outside. We will designate subsystems with internal actors.

### External actors

The external actors are represented in Fig. 1.
All external actors inherit a basic external actor. Below, we will define the instances of use of the basic actor.

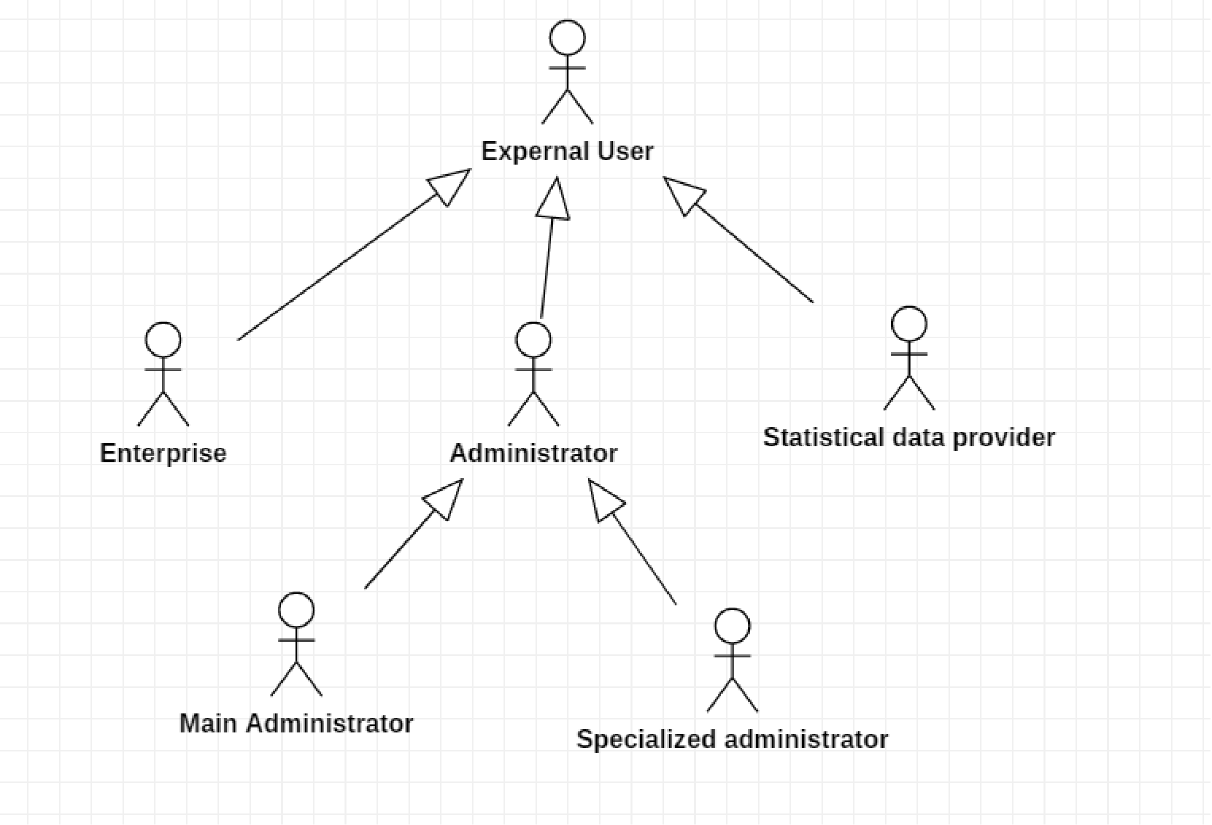


**Figure 1. External actors**

The external actors directly inheriting the base External User are:

- Enterprise - a user who uses the platform for innovation assessment as well as clustering with similar businesses and also obtaining advices for improvement of innovation process;
- Administrator. The Administrator, in turn, is the basis for the Main Administrator and the Specialized Administrator. The role of the ChiefМаин Administrator is to administer the overall activity of the platform and the access and rights of all other external users. The roles of the Specialized Administrator are related to individual aspects of the platform functionality described below in the document;
- Statistical data provider. Provides in electronic form data from the national statistics of the participating countries.

**Specialized Administrator**

The role of the Specialist Administrator is to administer the activities related to the management of incoming data streams and their processing and conducting analytics processing.

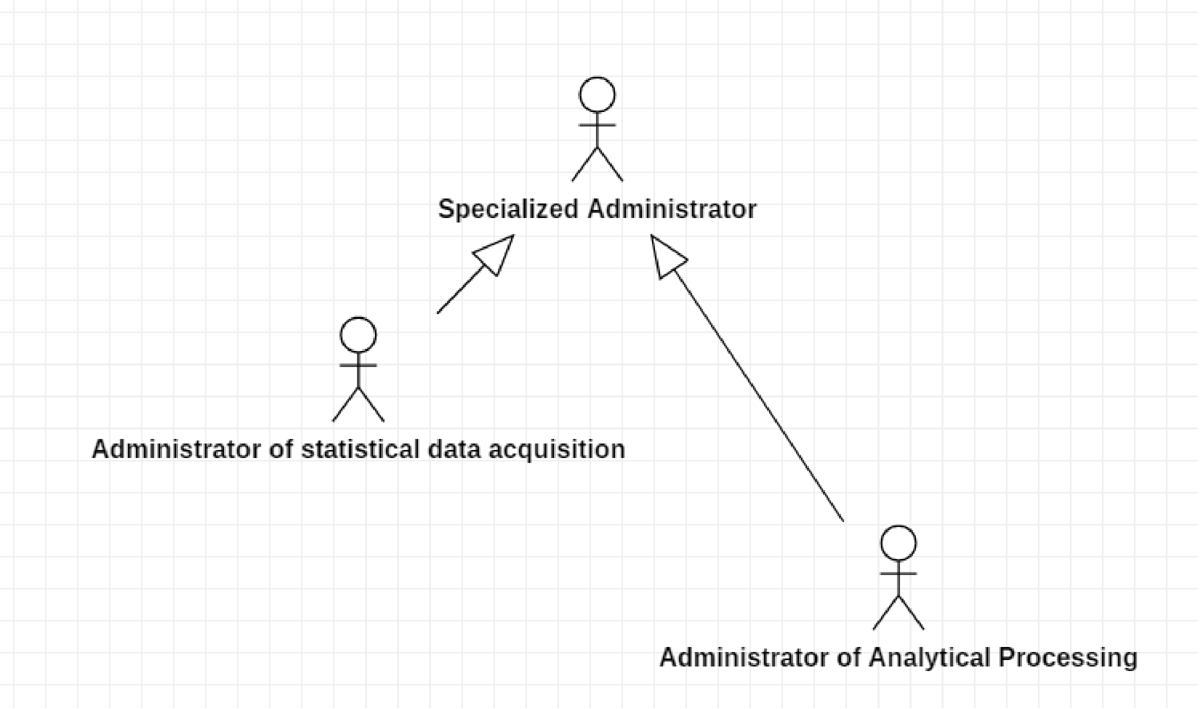

Specialized Administrator
Administrator of statistical data acquisition
Administrator of Analytical Processing

**Figure 2. Specialized Administrator**

*Statistical data provider*

The statistical information provider is a basic actor of five statistical information providers - one for each of the participating countries. The information will focus on enterprise innovation indicators. It may be at national or regional level.

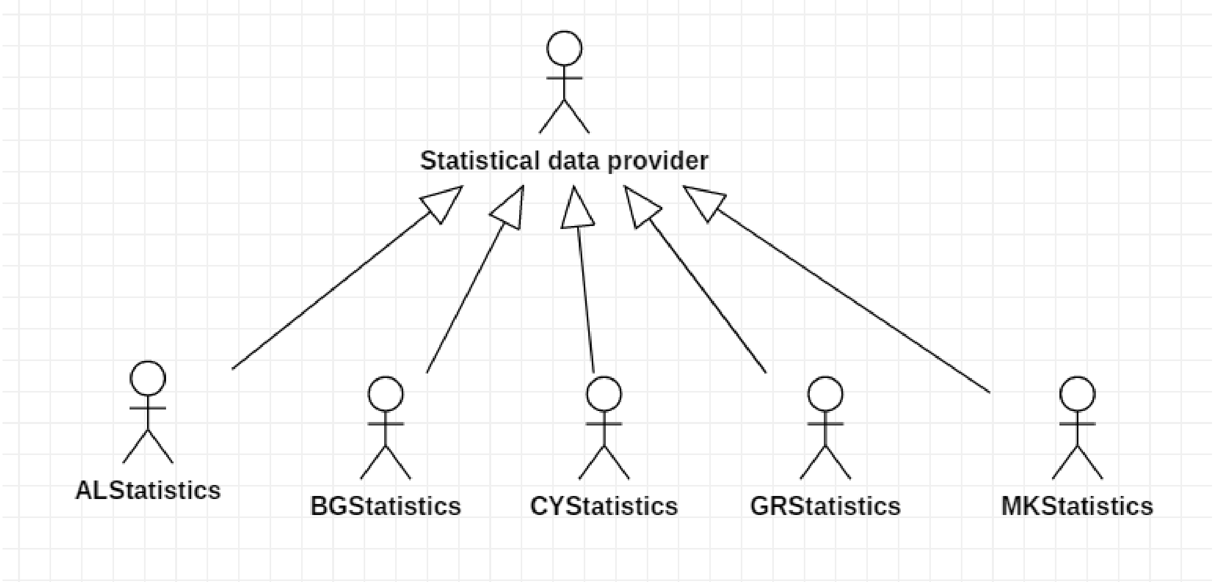


**Figure 3. Statistical data provider**

## Internal actors

Innernal actors inherit a basic Internal actor as shown in Figure 4. As can be seen from Figure 4, the internal actors are the following:

- Rights Manager - manages access rights for all actors;
- Access Manager - checks for system logging rights as well as sign-up;
- Questionnaire Manager - manages access storage and iterations with enterprise questionnaires;
- Data Manager - Provides an interface to data stored in databases. Statistical Adapter - designed to adapt data coming from external providers into a single internal form;
- Statistical Data Adapter - designed to adapt data coming from external providers into a single internal form;
- Simulator - serves to play what-if situations where businesses ask test data to play different variants to achieve better innovation A statistical receiver provides input of statistical information into the system;
- Statistical analyst - performs innovation analysis;
- Content Manager - Creates, edits and deletes the following components of the system: Modules of the system:
    - The UI (User Interface);
    - Menus;
    - The Brochure;
- Report Generator Generates Reports for:
    - The registered users;
    - Reports on appropriate grouping of enterprises according to their level of innovation, clustering of SMEs on the basis of innovation criteria and on the activity of each registered SME.
- Communications Manager - Connects the registered Enterprise;

- Advisor on Improving Innovation - Tips for Improving the Innovation Process for SMEs by Using Methods and Approaches for Multi-Criteria Decision Making and Choice in Identifying Innovative Clusters by SMEs;
- Statistical Data Acquirerer – conducts acquisition of statistical data provided bi national ststistics institution.

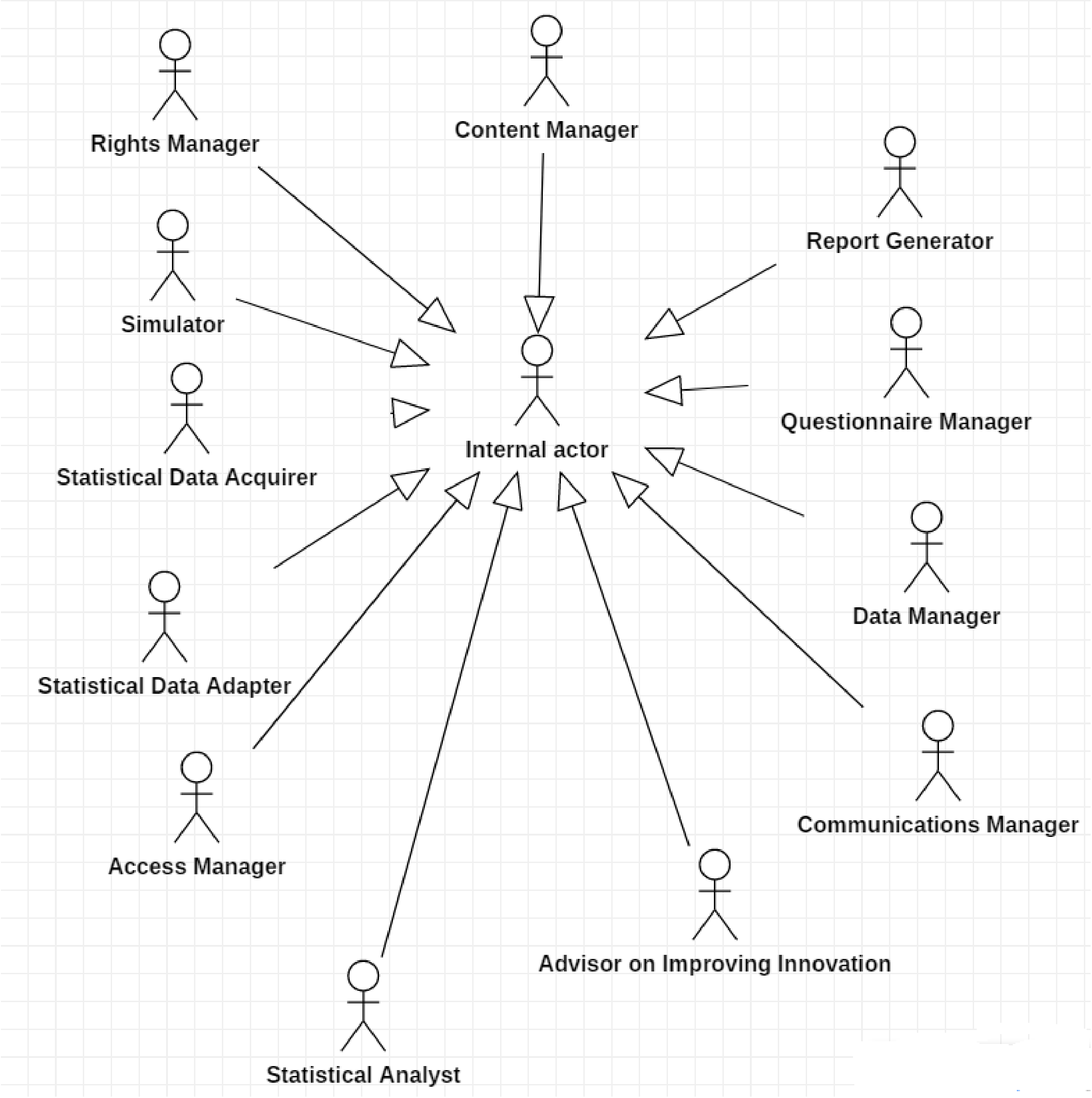


**Figure 4. Internal actors.**

### *Statistical Analizer*

In turn, the data analyst is the basis for specialized analysts. In this way, it is possible to add polymorphically new analyzers. Figure 5 presents the following data analysts:

- Statistical Analizer- Clustering of SMEs based on innovation criteria and on the activity of each registered SME by using methods and approaches for multi-criterion decision-making;
- Statistical Analyser according to other methods.

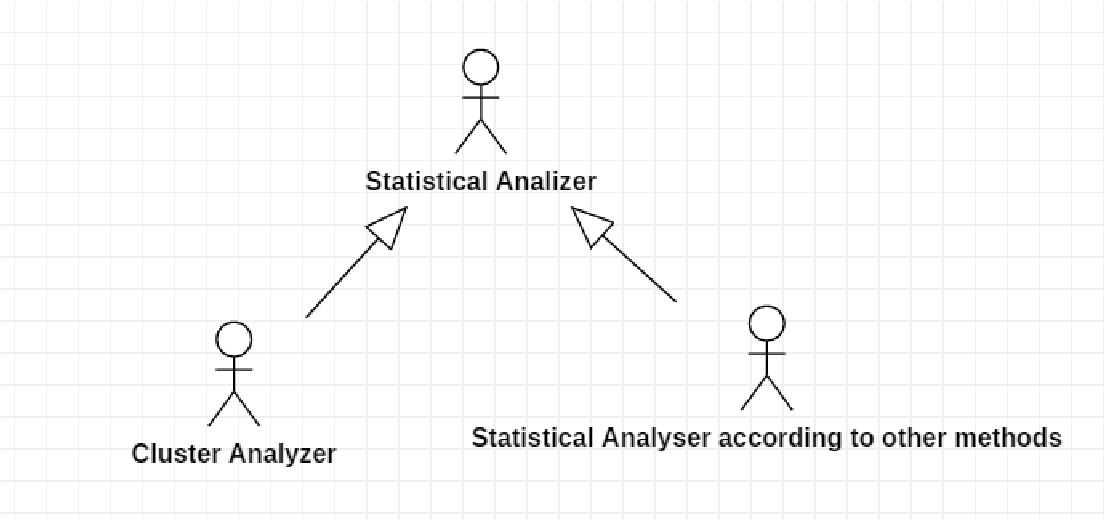


**Figure 5. Statistical Analyzer.**

## Use cases

### Use Cases of an External Actor

#### *Registration of an енмерпрuce*

From external actors only, the enterprise registers itself. The Main Administrator is registered during the installation of the system, and the specialized Administrators are registered by the Chief Administrator. In Fig. 6 shows the registration of an enterprise. The registration of the specialized administrators is different is only in the graphical interface - in the first case via the enterprise interface and in the second via the administrative panel.

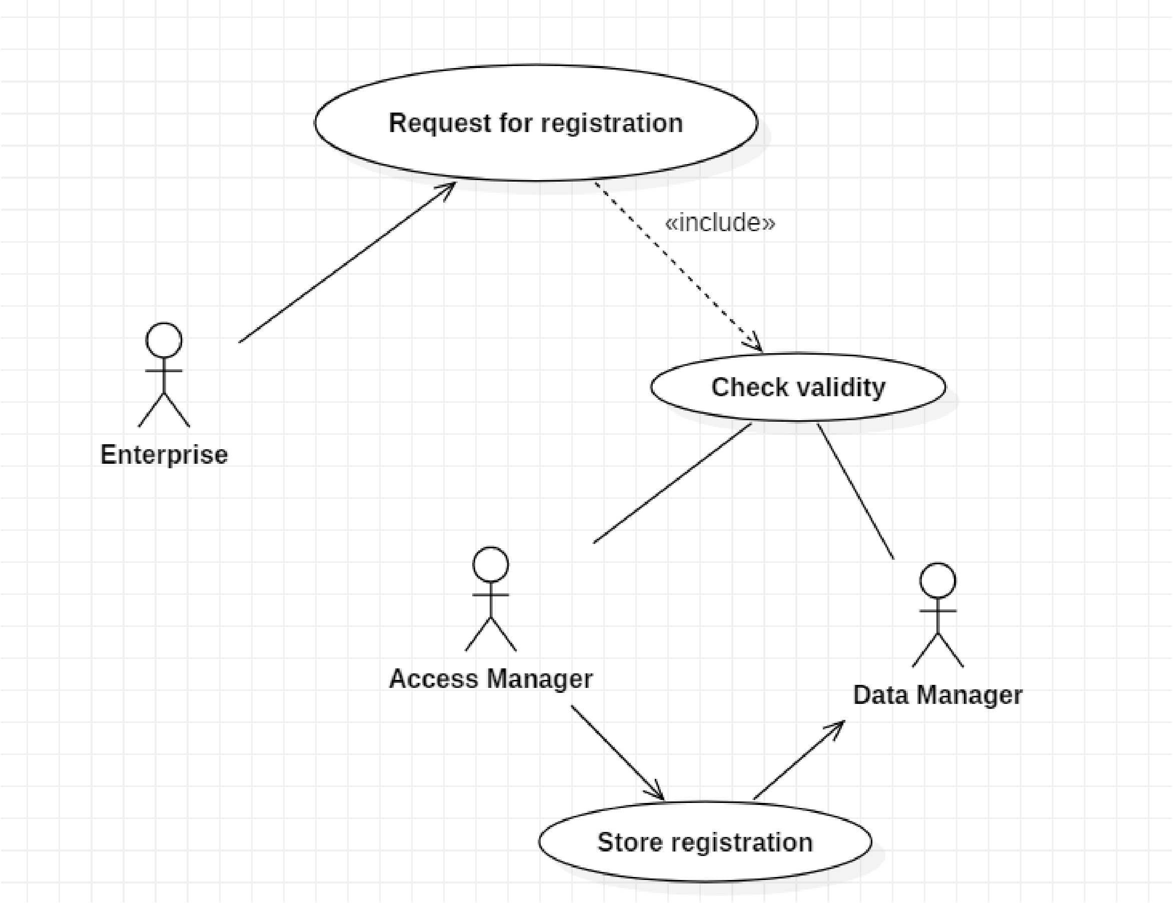


**Figure 6. Registration of an enterprise.**

**Implementation scenario**

**Preconditions**

- The enterprise is not registered.

**Steps to Implement**

1. Select a registration button from the browser;
2. The registration form is filled in;

3. The registration data is sent to the system;
4. The following checks shall be carried out:
   - Checking whether there is a registered business under this name;
   - Are the fields filled in correctly?;
5. In case or error error message is displayed;
6. In case of  properly filled-in form, the registration is retained and the enterprise or the specialized administrator receives the appropriate access rights to the resources and functions of the system.

### *SignIn*

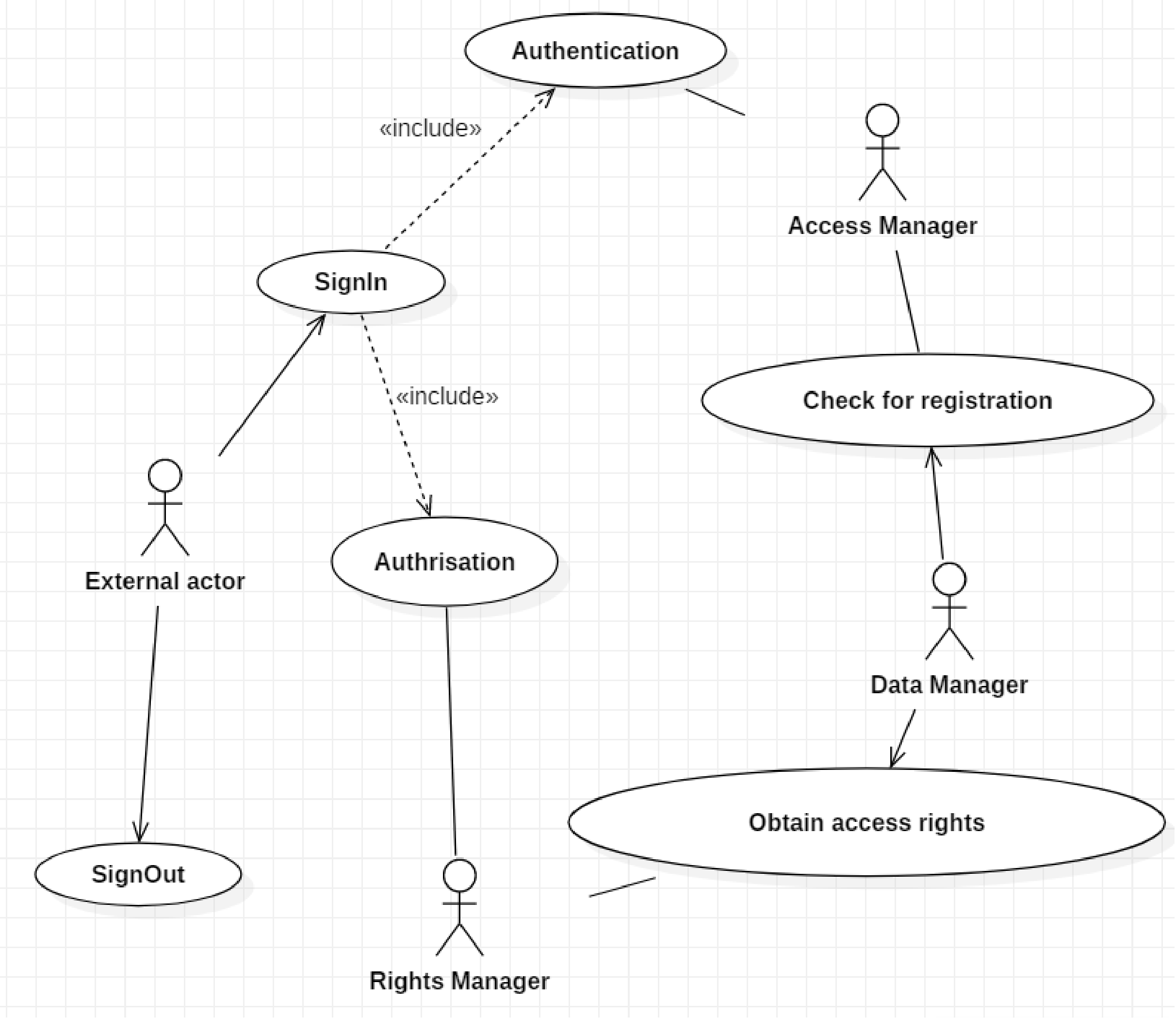


**Figure 7. SignIn/SignOut of an external user**

**Implementation scenario**

**Preconditions**

- A registered actor with access granted.

**Steps to Implement**

1. The external actor introduces a username and password;
2. The username and password are checked for correctness;
3. If the authentication fails, an access denied message appears;
4. If the authentication passes, the permissions given to the user are restored and accessed to the system.

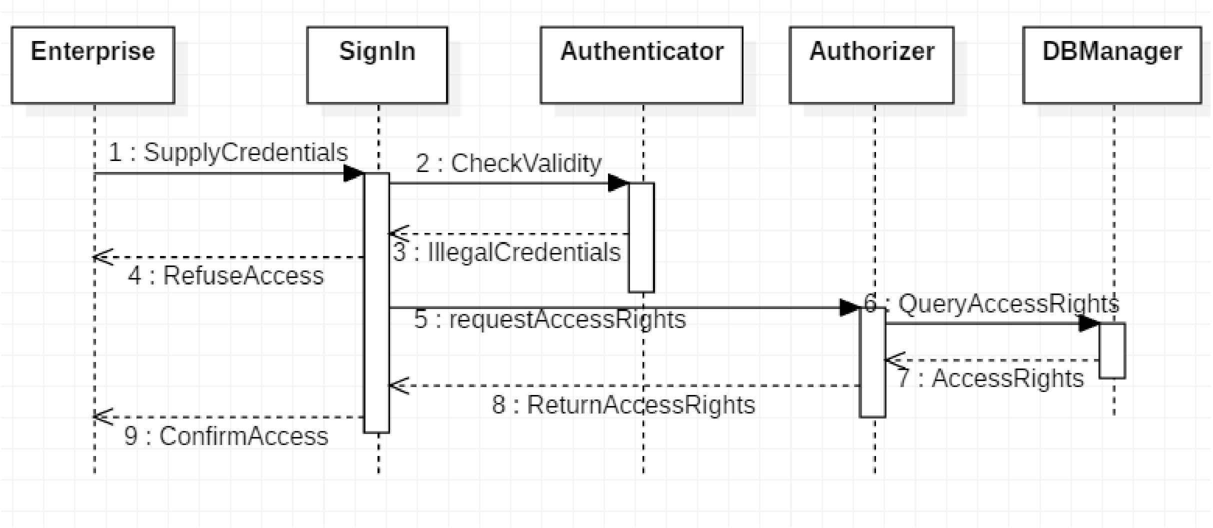


**Figure 8. Input sequence diagram.**

**Post conditions**

- One of two possible states - the user can work with the system or be allowed access to the system.

### *SignOut*

**Implementation scenario**
**Precondition**

- The external actor must be signed in.

**Steps to Implement**

1. The external actor presses the SignOut button;
2. His authorization for this session is no longer valid.

**Post conditions**

- The external actor can not operate with the system.

## Use cases of Main Administrator

### Grant access to enterprise

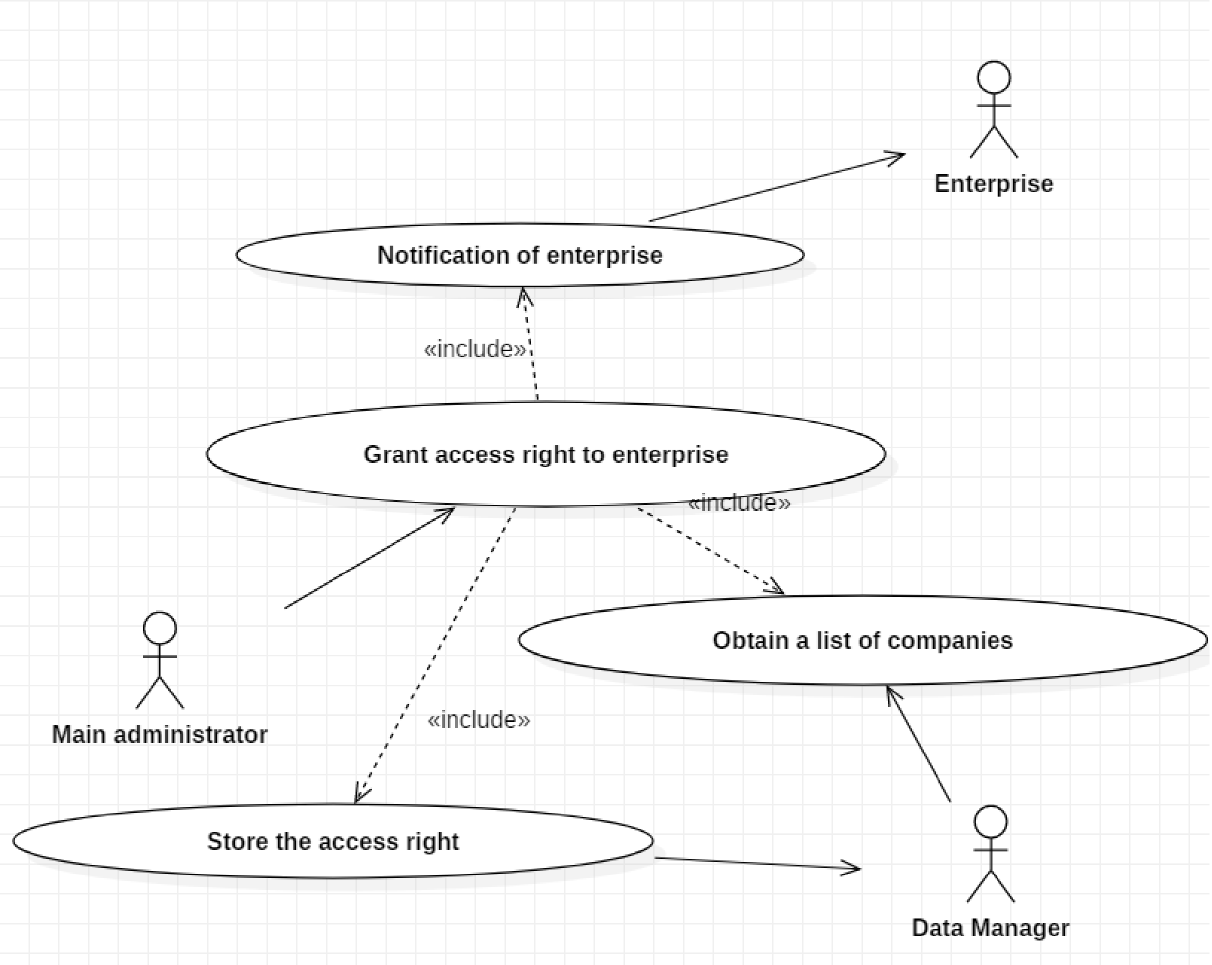


**Figure 9. Use case grant access to Enterprise.**

**Implementation scenario.**
**Preconditions**
- The Main Administrator must be signed in;
- The enterprise must be on the list of enterprises that have requested access to the system.

**Steps to Implement**.
1. The Main ddministrator selects Enterprise from the Enterprise List;
2. The Main administrator adds right to a selected enterprise;
3. The Main administrator stores the rights of the enterprise;
4. The Enterpise if notified about granted accsses right.

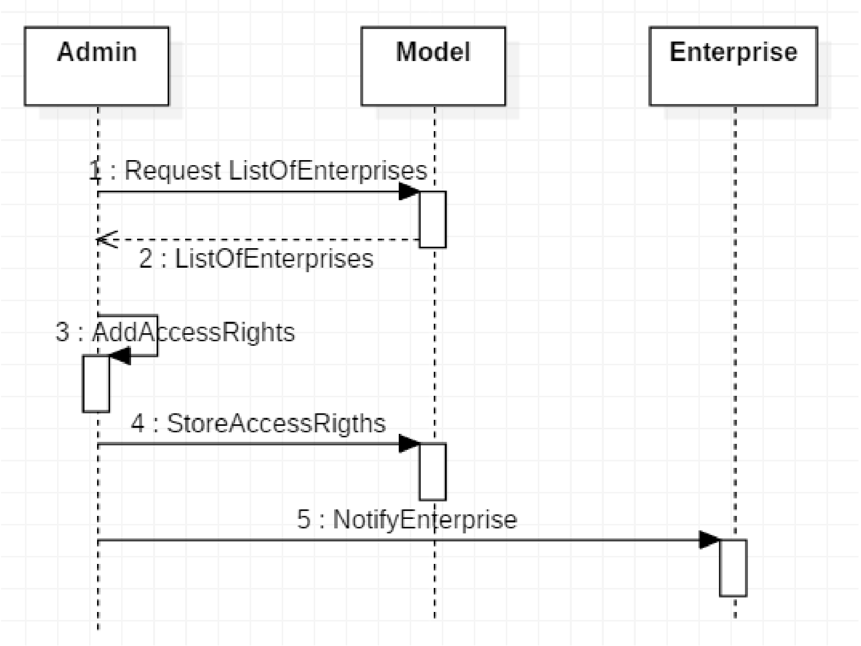


**Figure 10. Sequence diagram of granting access rights.**

**Post conditions**

– The enterprise has access rights to specified system resource or function.

## Revoke of access rights from Enterprise

**Implementation scenario.**

**Preconditions**

1. The Main Administrator must be signed in;
2. An enterprise should not be on the list of enterprises that have requested access to the system.

**Steps to Implement**.

1. The Main Administrator selects Enterprise from the Enterprise List;
2. The Main Administrator revokes right from a selected enterprise;
3. The Main Administrator stores stores the information about revoked access rights from the enterprise;
4. The Enteprise if notified about revoked accsses right.

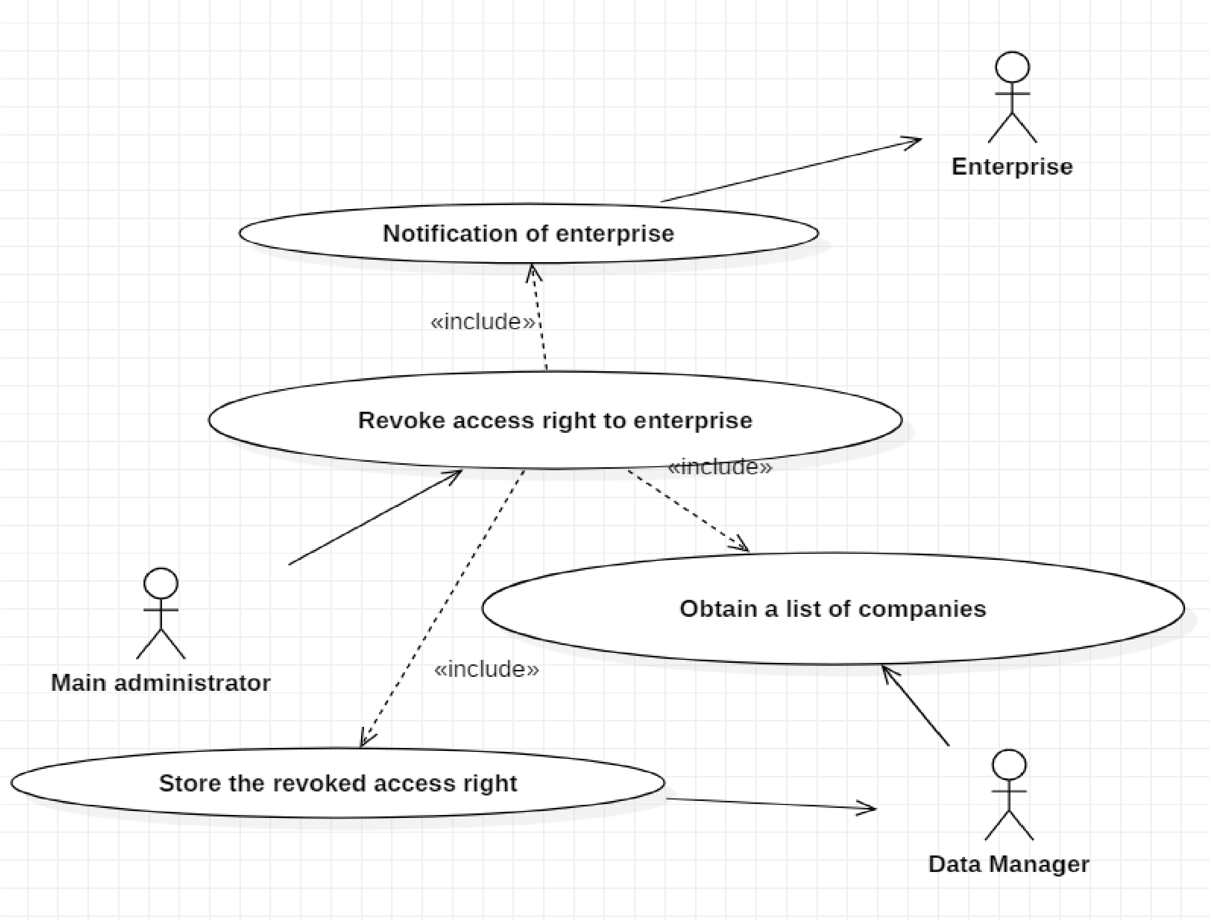

Enterprise
Notification of enterprise
«include»
Revoke access right to enterprise
«include»
Obtain a list of companies
Main administrator
«include»
Store the revoked access right
Data Manager

Figure 11. Use case for revoke access right from Enterprise

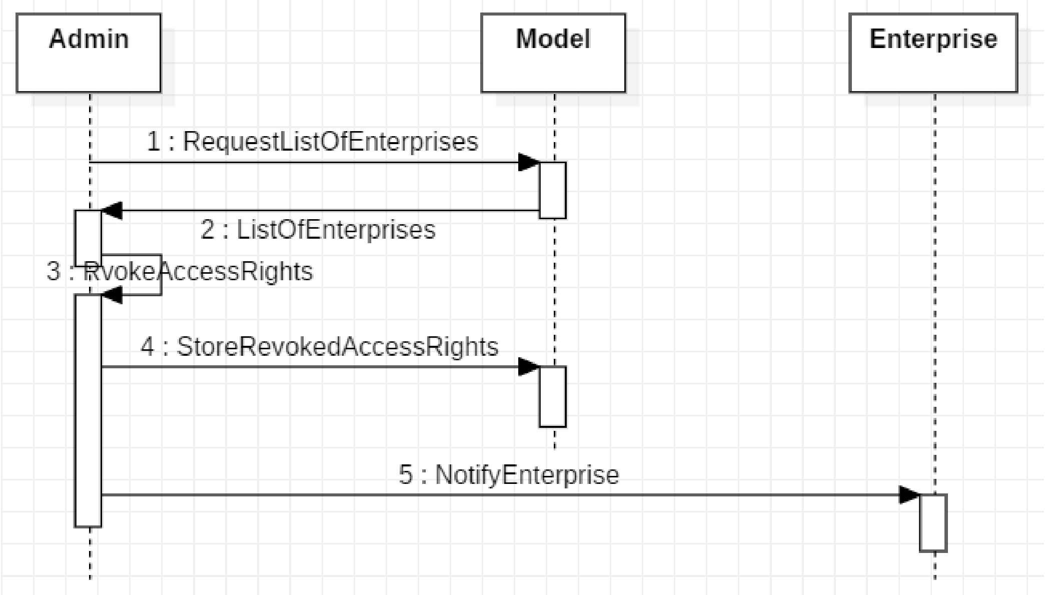


Figure 12. Sequence Diagram for revoke access right from Enterprise.

**Post conditions**

– The enterprise does not have access rights to system resources and functions.

The cases of use of the granting and revoking of rights of the specialized administrators are completely analogous to the use cases of enterprises and therefore are not described in this document.

## Create groups of specialized administrators

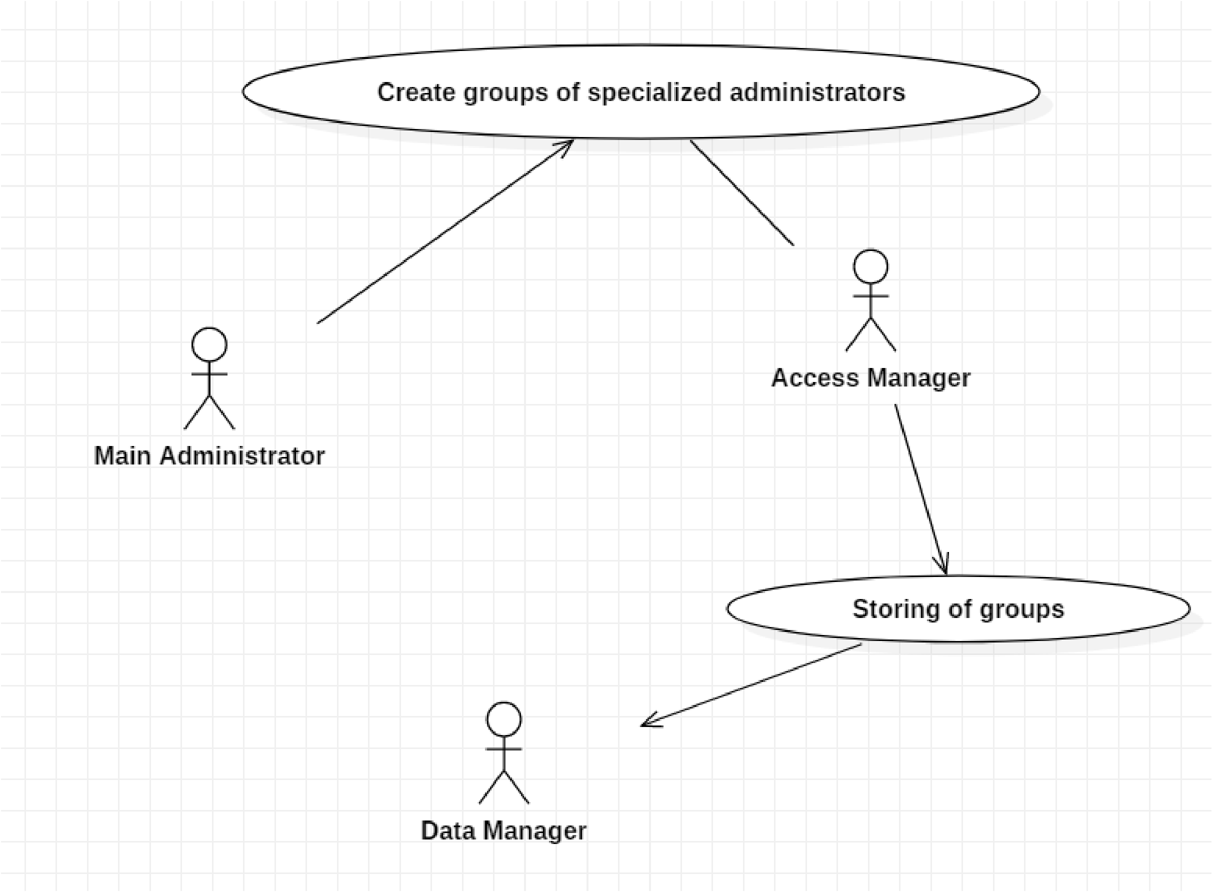


**Figure 13. Use case of creating groups of specialized administrators.**

**Implementation scenario.**

**Preconditions**

1. The Main Administrator must be signed in.

**Steps to Implement**.

1. The Main Administrator creates a new group of dedicated administrators;
2. TheAccess Manager is notified for the new group of specialized administrators;
3. The new group and associated access rights are stored in the database.

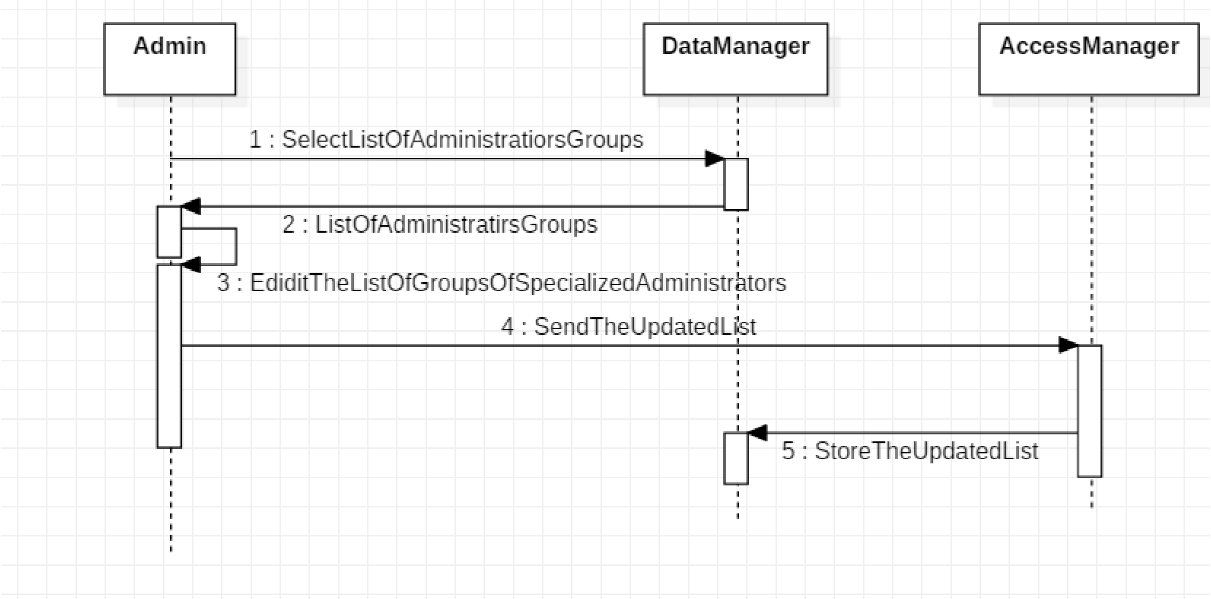

Admin
DataManager
AccessManager
1 : SelectListOfAdministratiorsGroups
2 : ListOfAdministratirsGroups
3 : EdiditTheListOfGroupsOfSpecializedAdministrators
4 : SendTheUpdatedList
5 : StoreTheUpdatedList

**Figure 14. Sequence diagram of creating a group of specialized administrators.**

**Post conditions**

- A new group of specialized administrators has been created.

## System content management

**Implementation scenario.**

**Preconditions**

1. The Main Administrator must be signed in.

**Steps to Implement**.

1. The Chief Administrator starts Content Management;
2. Depending on the selected type of system components, the Content Manager performs one of the following activities:

- Managing System Module;
- Managing Menus;
- Manage the User Interface;
- Editing a Brochure.

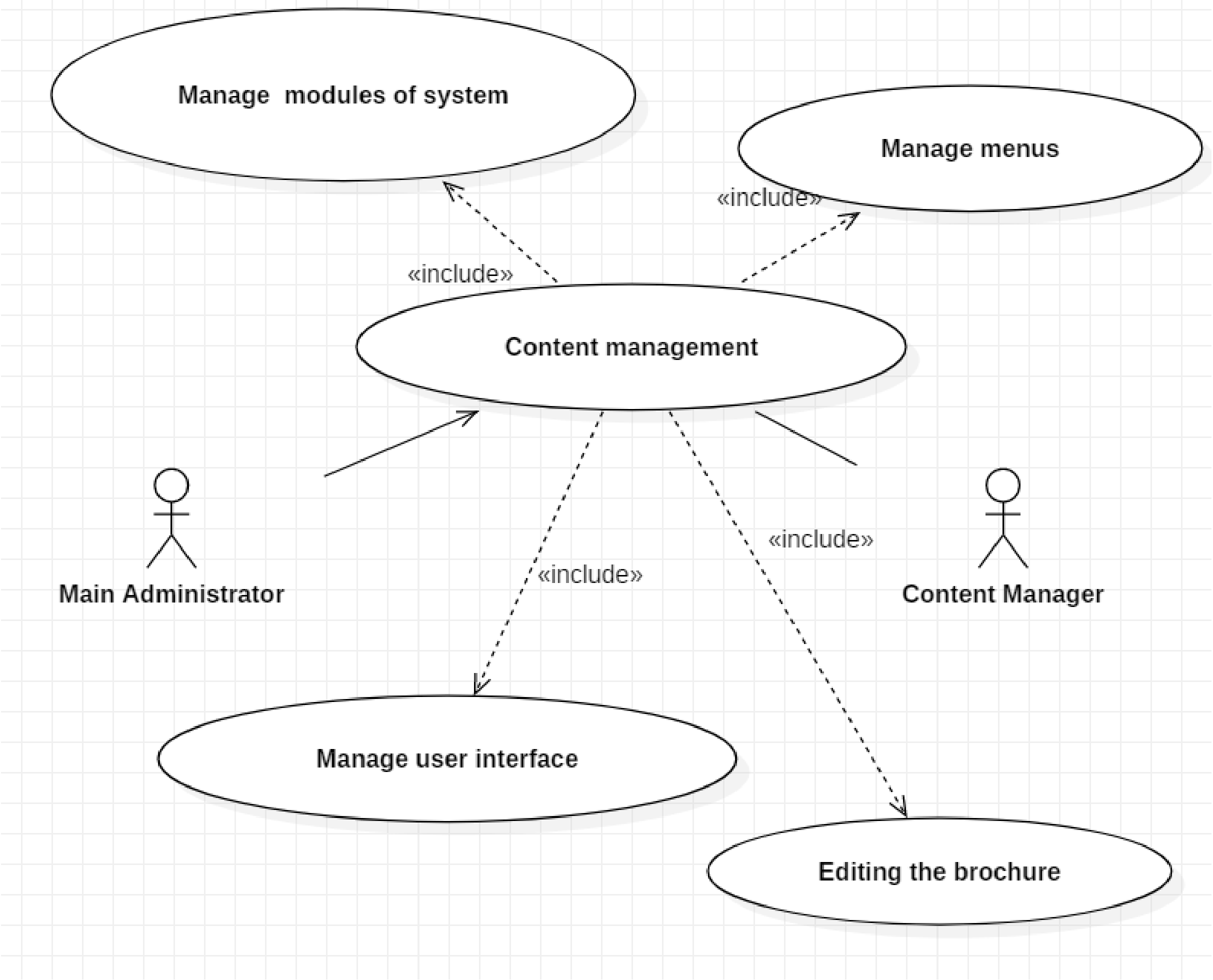


**Figure 15. Use case content management**

Below we will discuss the management of system modules as it involves the seamless addition of new system features. Each one new functionality is implemented through a module located in a module repository, which in turn can be a folder accessible only to the Main administrator.

**Implementation scenario.**
**Preconditions**
1. The Main Administrator must be signed in;
2. When adding a new module, the new module must be located in the module repository.
**Steps to Implement**.
1. The Main administrator chooses one of the two functions:
- Add a module;
- Delete a module;
2. In case of adding a module, the new module is imported into the system;
3. In case of deletion the selected module is removed from the system.

**Post conditions**

- Added or deleted module from the system.

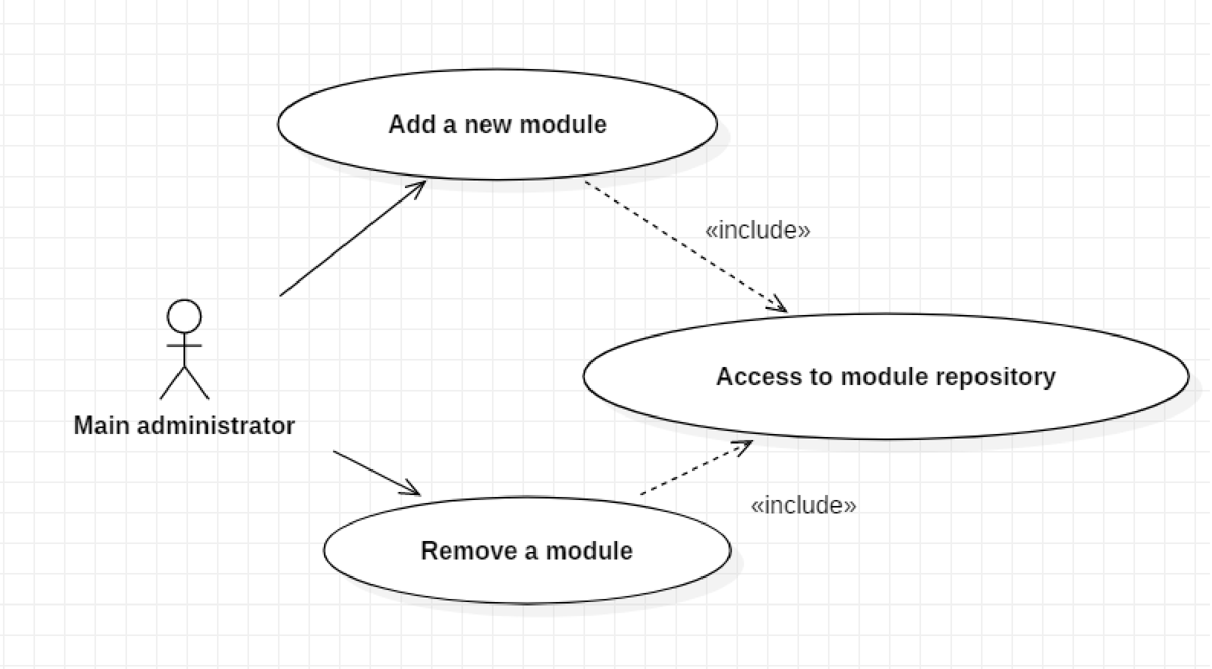


**Figure 16. Use case of add or remove a module**

## Reports generation

**Implementation scenario**
**Preconditions**

1. The Main Administrator must be signed in.

**Steps to Implement**

1. The Main administrator chooses the report type:
- Users signed in at that moment;
- Reports on appropriate grouping of enterprises according to their level of innovation, clustering of SMEs on the basis of innovation criteria and on the activity of each registered SME;
- Other type of report;

2. The Report Generator displays the content of the report in the form from the administrative panel.

**Post conditions**

- Generated relevant report.

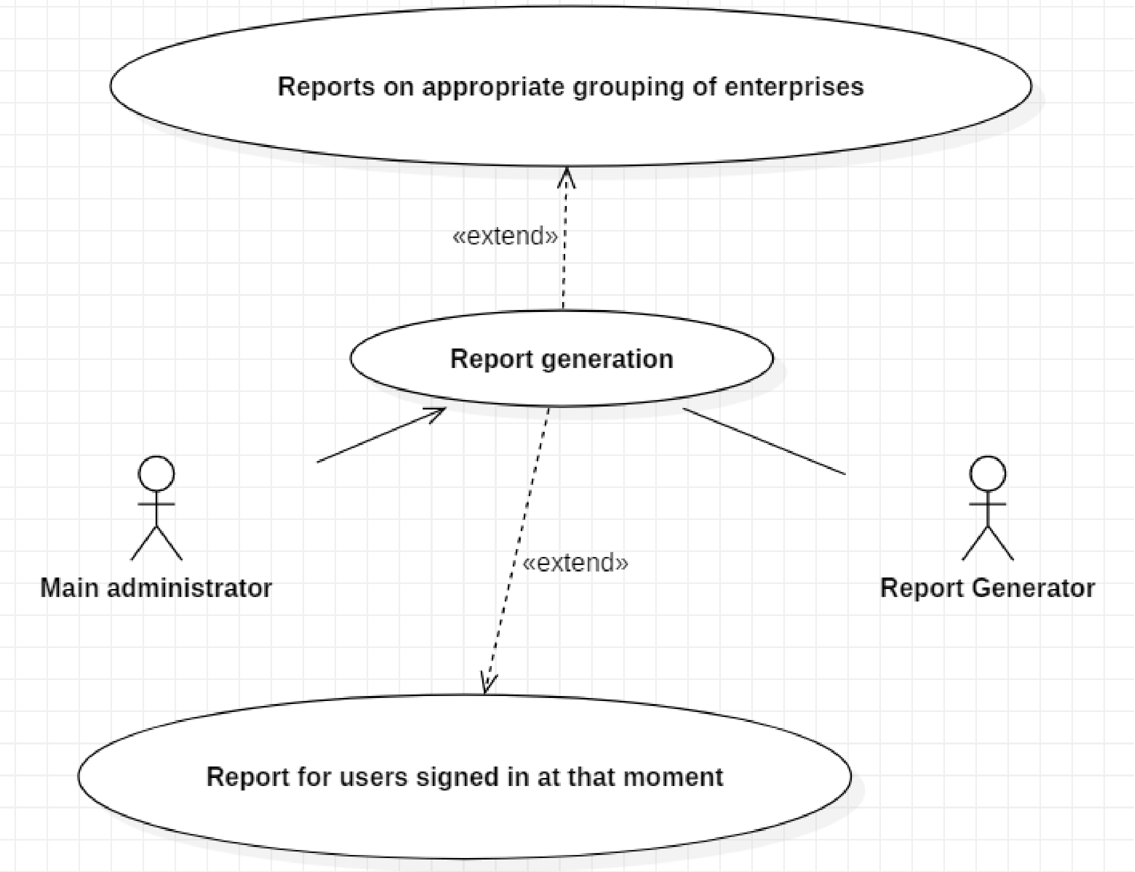

Reports on appropriate grouping of enterprises
«extend»
Report generation
Main administrator
«extend»
Report Generator
Report for users signed in at that moment

**Figure 17. Use case of generation of a report**

## Use cases of specialized administrators

### Obtaining of statistical data

**Implementation scenario**

**Preconditions**

1. Administrator of statistical data acquisition must be signed in;
2. New statistical data from national statistics or another source are available.

**Steps to Implement**.

1. The Administrator of statistical data acquisition sends a request to the Statistical Data Provider;
2. The Statistical data provider sends unformatted statistical data to the Statistical Data Acquirerer;
3. The Statistical Data Acquirerer checks them for errors and format them and then submits them to the Statistical Data Adapter;
4. The Statistical Data Adapter converts them into an internal presentation that is stored by the Data Manager in the database;

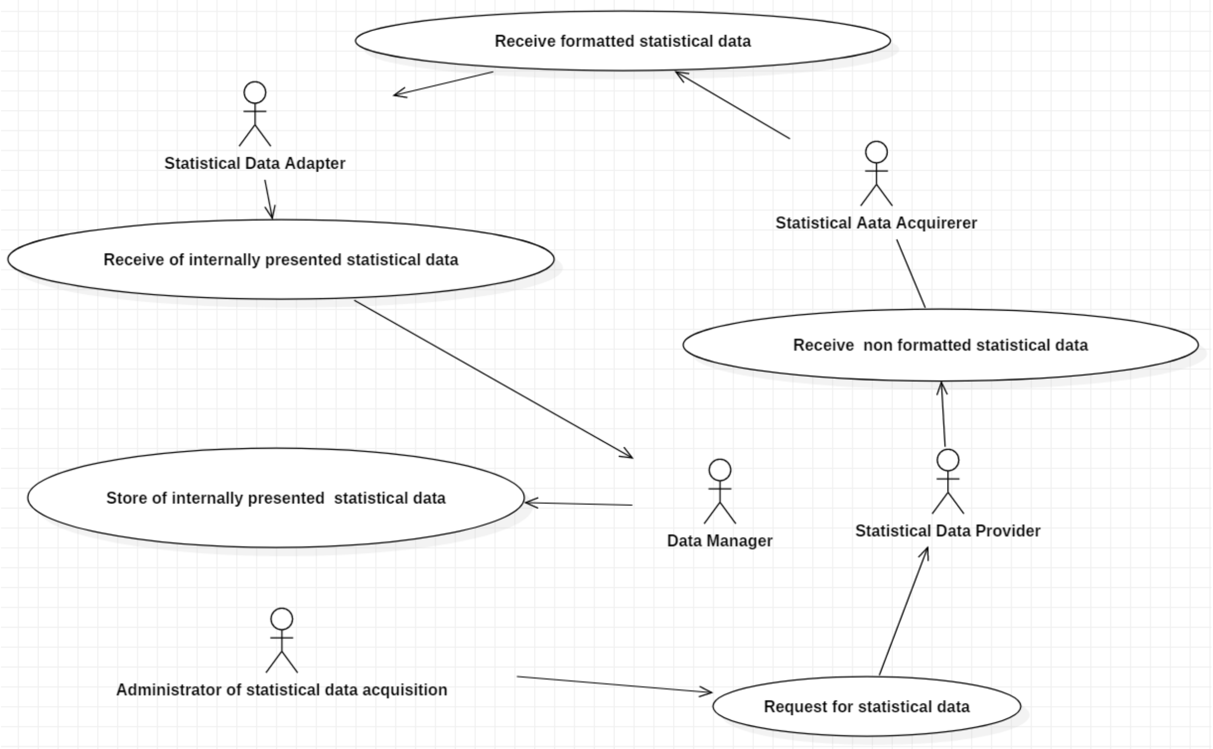
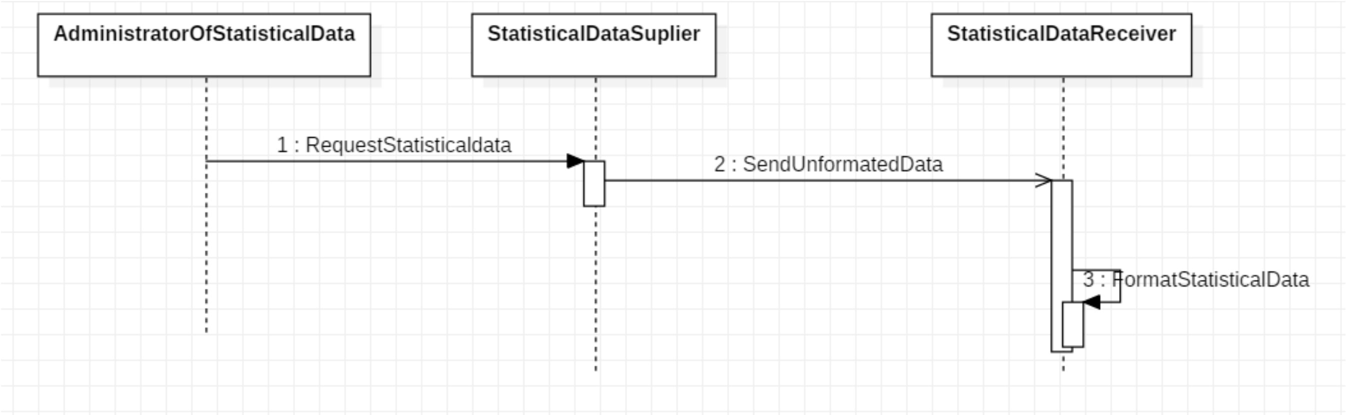

Receive formatted statistical data
Statistical Data Adapter
Statistical Aata Acquirerer
Receive of internally presented statistical data
Receive non formatted statistical data
Store of internally presented statistical data
Data Manager
Statistical Data Provider
Administrator of statistical data acquisition
Request for statistical data

**Figure 18. Use case of obtaining of statistical data.**


AdministratorOfStatisticalData
StatisticalDataSuplier
StatisticalDataReceiver
1 : RequestStatisticaldata
2 : SendUnformatedData
3 : FormatStatisticalData

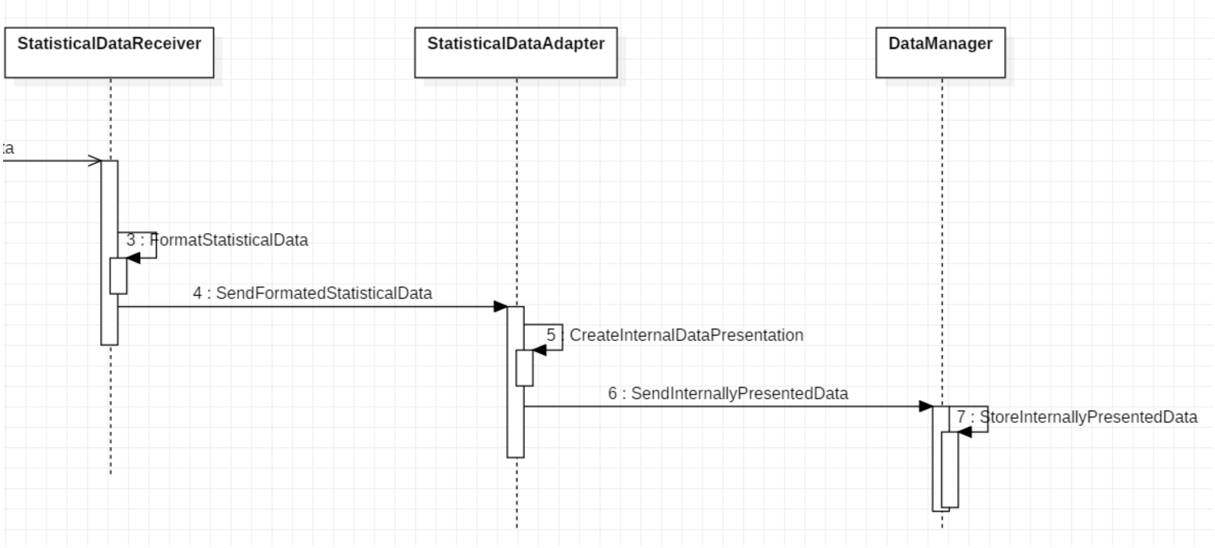

StatisticalDataReceiver
StatisticalDataAdapter
DataManager
a
3 : FormatStatisticalData
4 : SendFormatedStatisticalData
5 : CreateInternalDataPresentation
6 : SendInternallyPresentedData
7 : StoreInternallyPresentedData

**Figure 19. Sequence diagram of obtaining of statistical data.**

**Post conditions**

- The new statistical data is stored in the database.

## Perform analytical calculations

**Implementation scenario**

**Preconditions**

1. Administrator of Analytical Processing must be signed in;
2. Existence of new statistical dat for analysis statistics.

**Steps to Implement**.

1. Administrator of Analytical Processing starts the analytical processing process with a request for a certain type of analytical calculations;
2. The data required for analytical calculations from the relevant database are retrieved;
3. Performing of analytical calculations;
4. The result of the analytical calculations is saved in the database.

**Post conditions**

- The result of the analytical calculations is saved in the database.

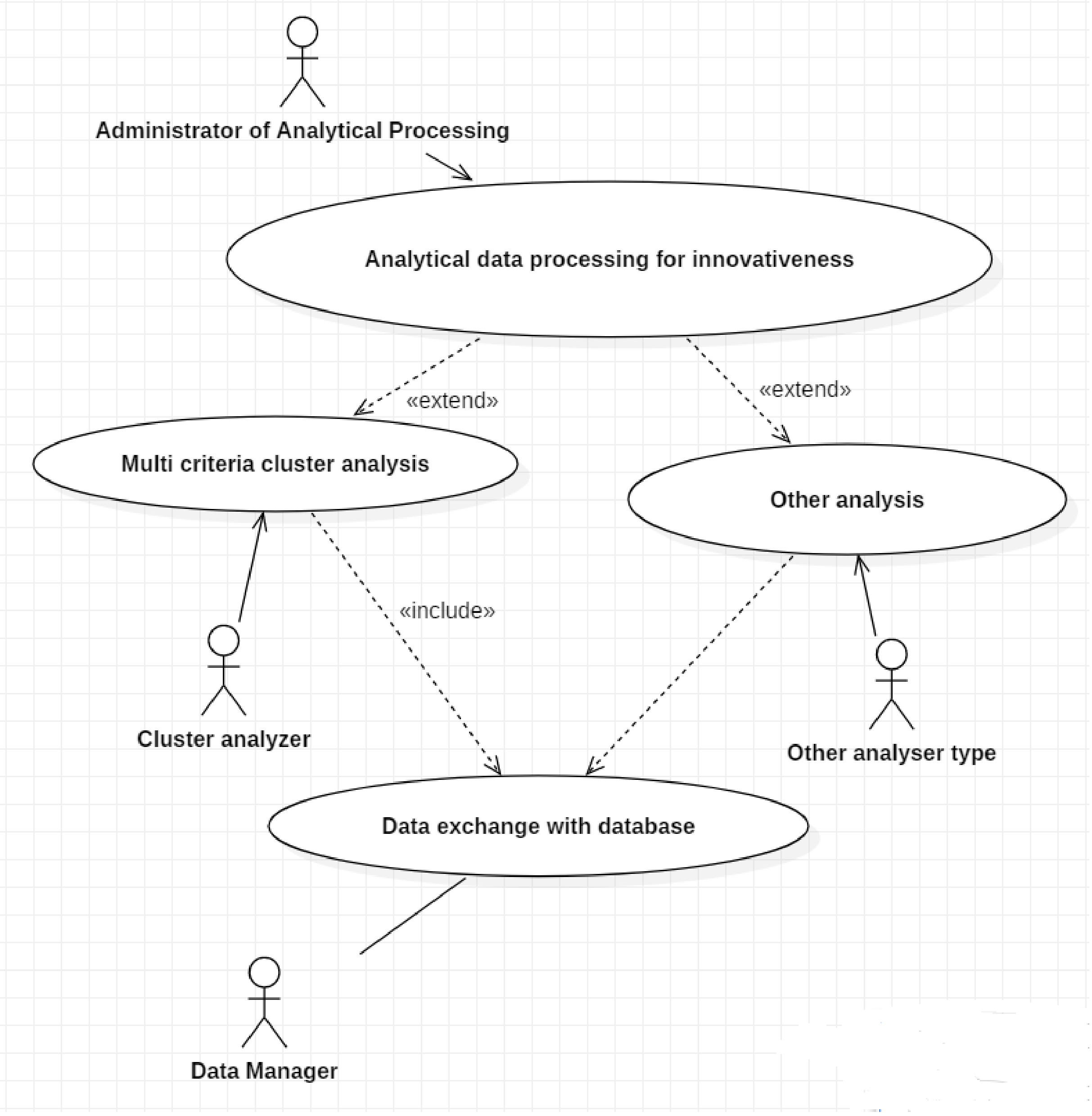
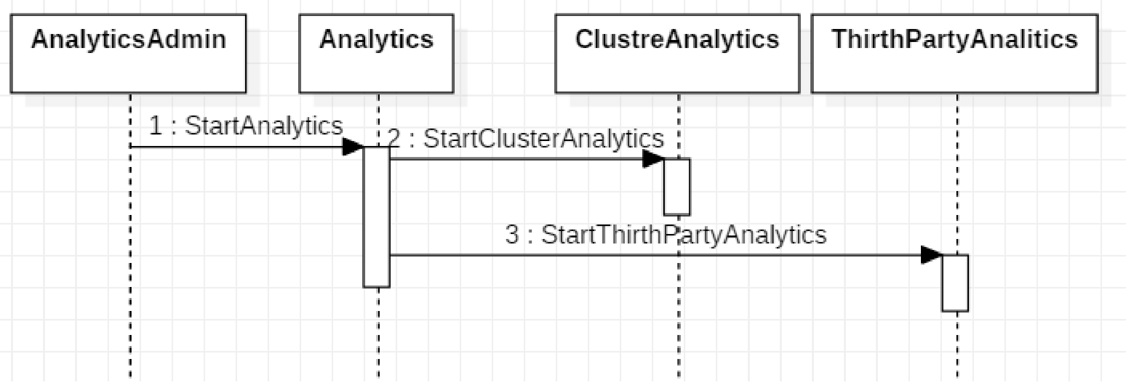

Administrator of Analytical Processing
Analytical data processing for innovativeness
«extend»
«extend»
Multi criteria cluster analysis
Other analysis
«include»
Cluster analyzer
Other analyser type
Data exchange with database
Data Manager

**Figure 20.  Use case of analytics computations**



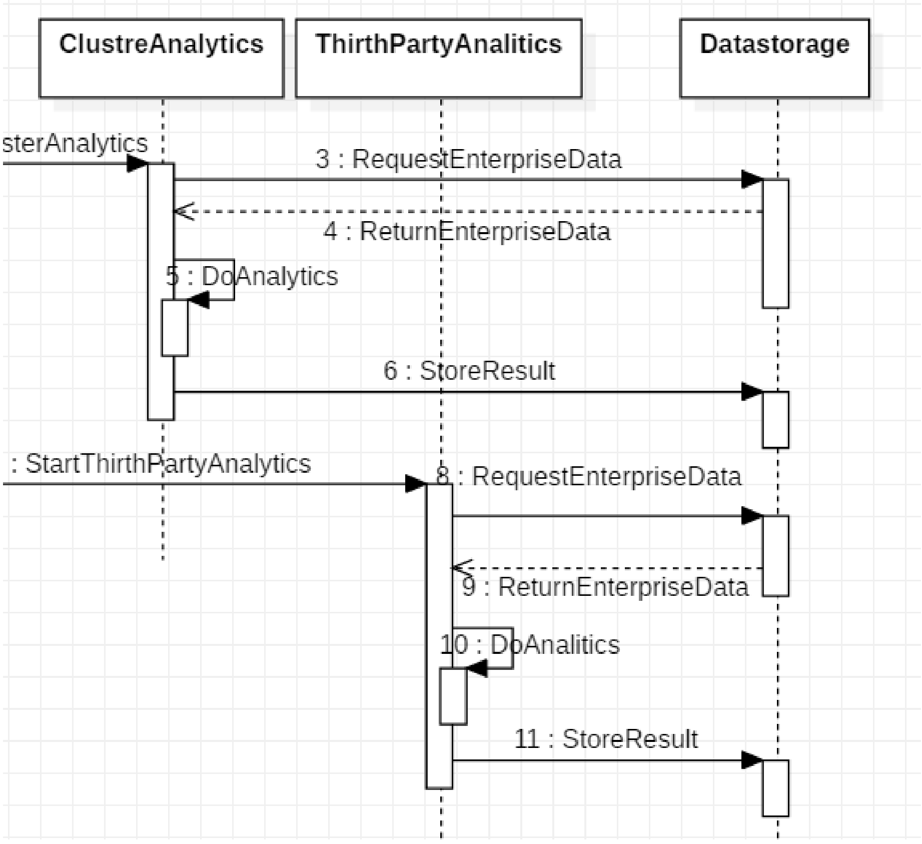


**Figure 21. Sequence diagram of analytical computations**

## Use cases of Enterprise

### Filling in a questionnaire

**Implementation scenario**
**Preconditions**
1. The Enterprise must be signed in;
2. The questionnaire was not completed for the current year by the Enterprise.
**Steps to Implement**.
1. The Enterprise receives questionnaires related to its innovativeness;
2. The enterprise fills in the questionnaires;
3. Newly received enterprise statistical data related with innovation proces are stored in the database.
**Post conditions**

- The entered statistical data related with innovation proces may be used for analytical computations.

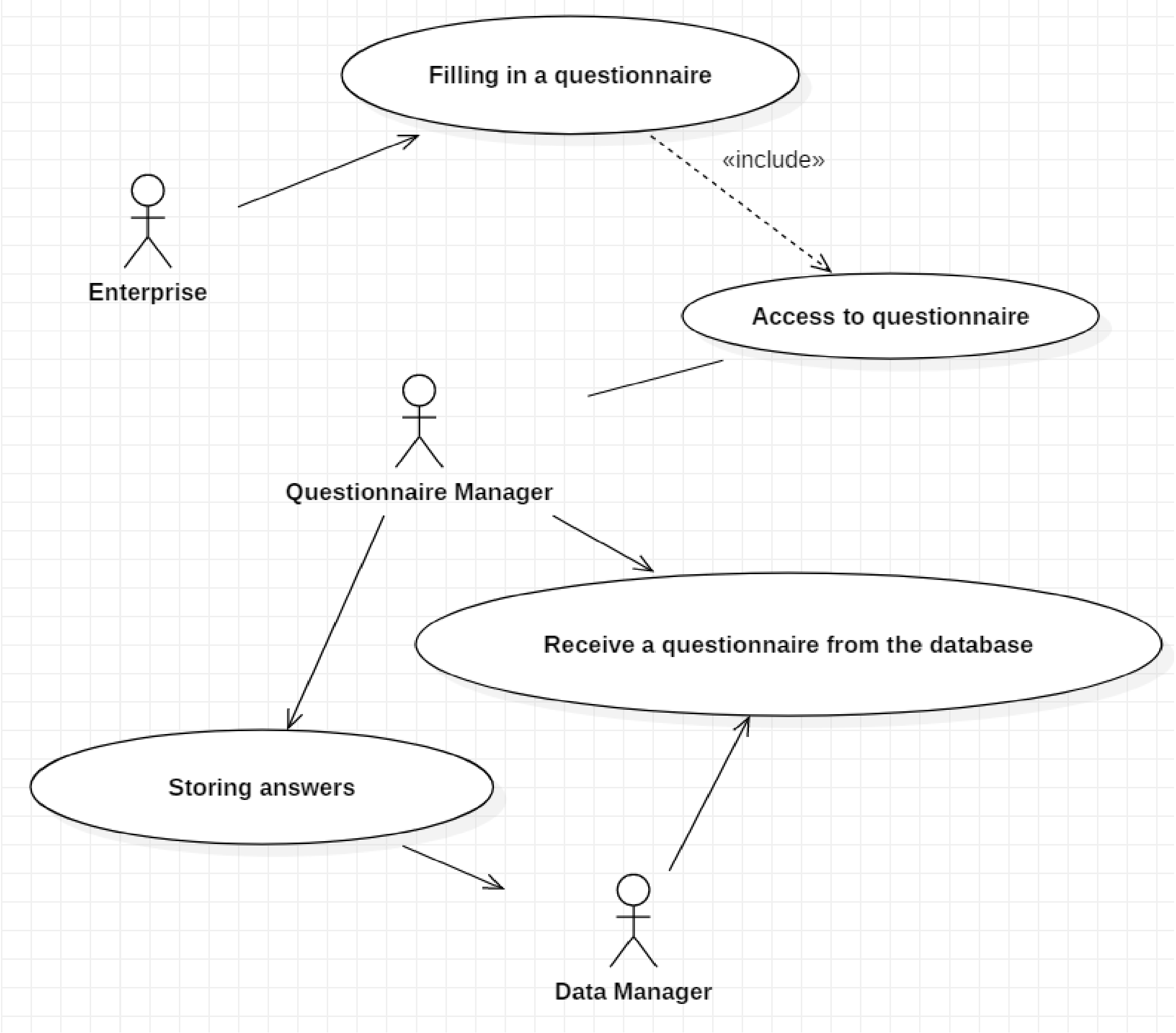


Figure 22. Use case of filling in quetionnaire.

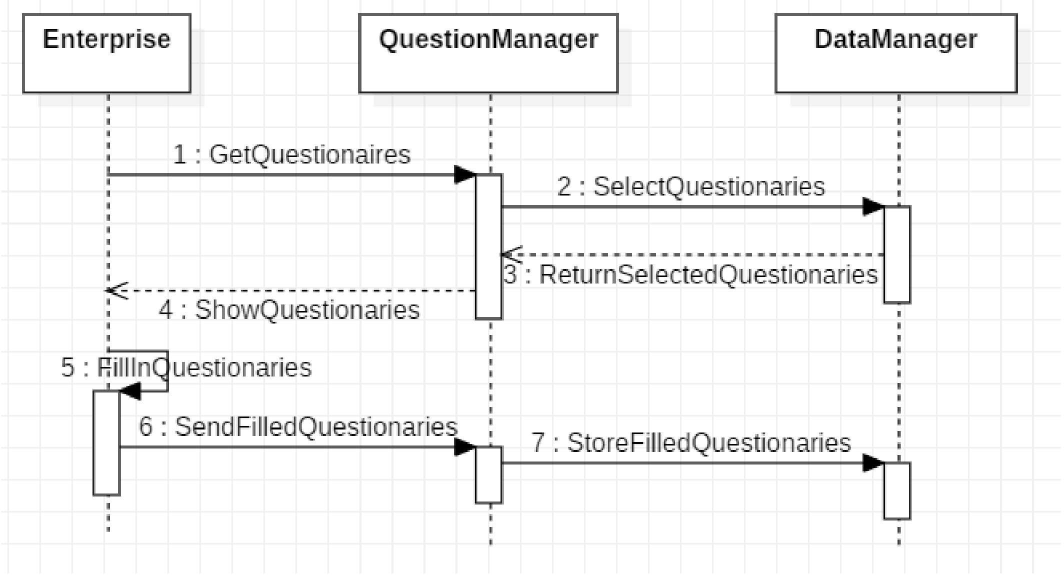


Figure 22. Sequence diagram of filling in quetionnaire

## Browsing of analytical calculation results

**Implementation scenario**

**Preconditions**
1. The Enterprise must be signed in;
2. An analytical processing for innovaiveness of data from questionnaires filled in by enterprises was carried out.

**Steps to Implement**.
1. The Enterprise considers the results of the analysis of its innovation;
2. The Enterprise discovers its innovativeness in comparison with other enterprises or countries;
3. The Enterprise fints the enterprises close to its innovative results.

**Post conditions**
- The Enterprise is informed about the level of its innovation;
- The Enterprise is informed about neighboring enterprises with a similar level of innovation.

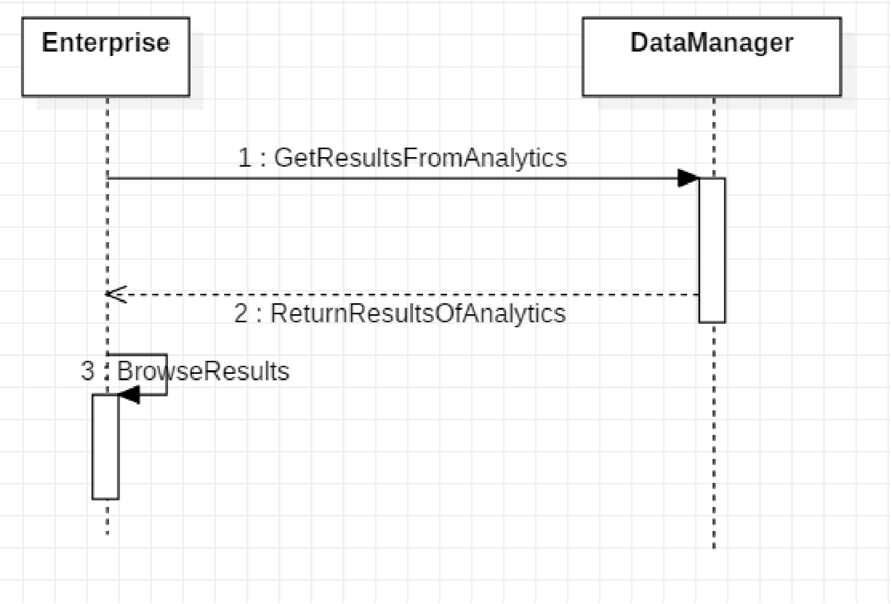


**Figure 23. Sequence diagram of browsing of analytical calculation results**

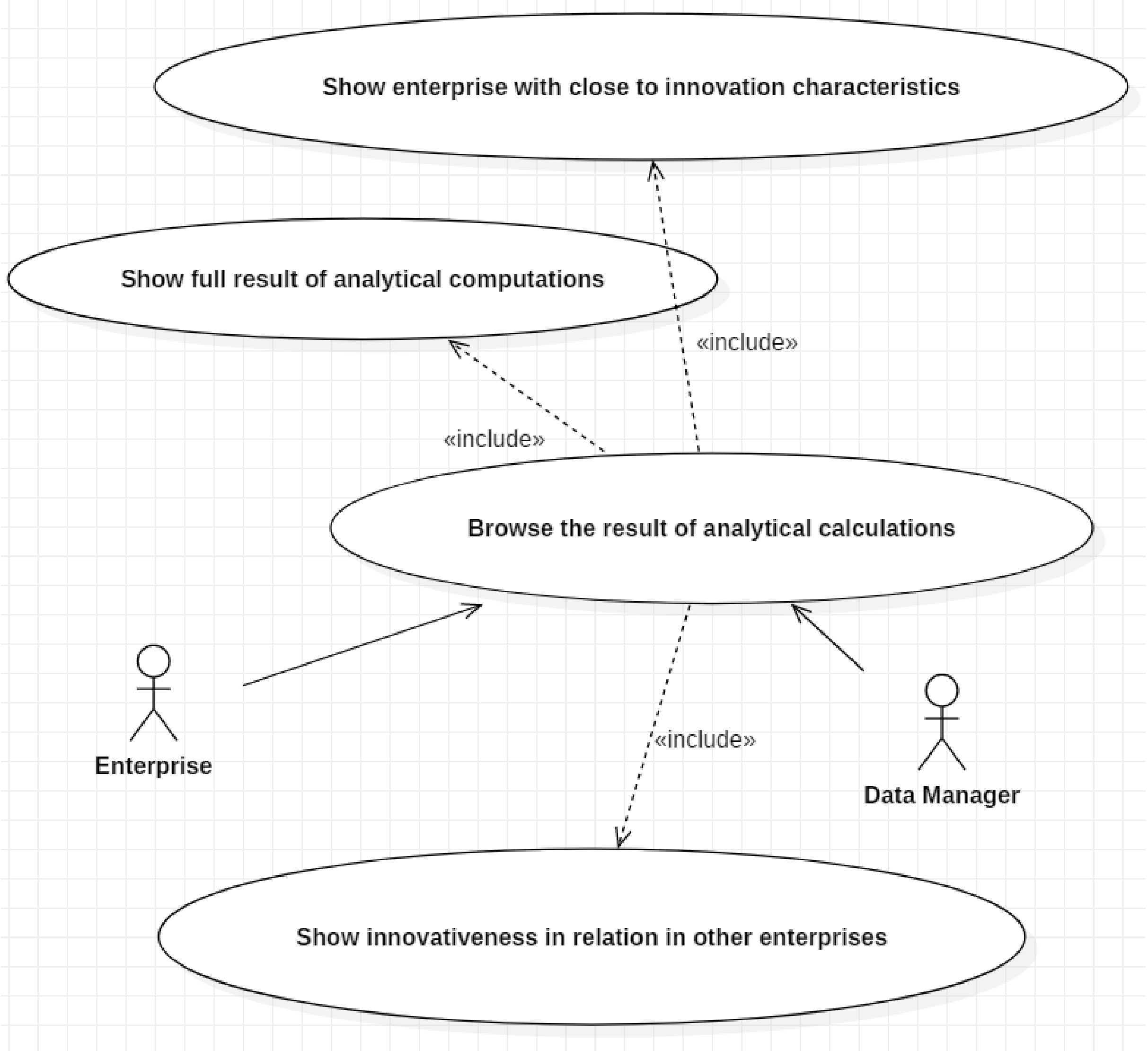

Show enterprise with close to innovation characteristics
Show full result of analytical computations
«include»
«include»
Browse the result of analytical calculations
Enterprise
«include»
Data Manager
Show innovativeness in relation in other enterprises

**Figure 24. Use case of browsing results of analytical computations**

## Using a simulator to test possible innovativeness of Enterprise

The preparation for a simulation analytical computation is similar to the introduction of data for assessing the innovation of enterprises, except that the introduction of simulation analysis data can be carried out repeatedly, as well as the possibility of filling in the questionnaires with different data variants for the enterprise. The analysis is also started immediately after entering the simulation data. The reason for this is that cluster evaluation will only be performed on the particular enterprise.

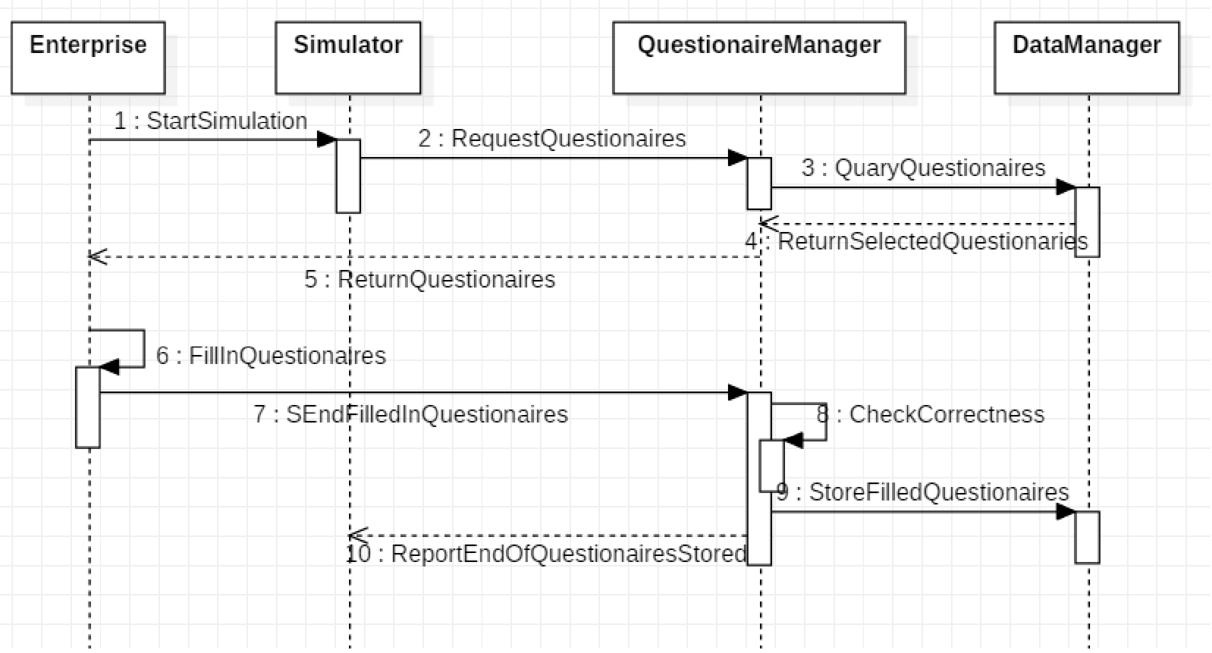
**Implementation scenario**
**Preconditions**

1. The Enterprise must be signed in.

**Steps to Implement**.
1. The enterprise turns to the Simulator forsimulation initialization simulator;
2. The Simulator questionnaires for completion;
3. Questionnaires are received from the Questionnaire Manager;
4. Questionnaires for completion are filled in by the Enterprise;
5. Questionnaire completion is controlled by the Questionnaire Manager;
6. Correctly filled out questionnaires are sent to the Data Manager for storing in a database;
7. The Simulator launches the appropriate statistical data analyzer;
8. The results of the performance of the data analyzer are stored for subsequent consideration.

**Post conditions**

1. The results of the simulation analysis are stored only for use by the Enterprise.



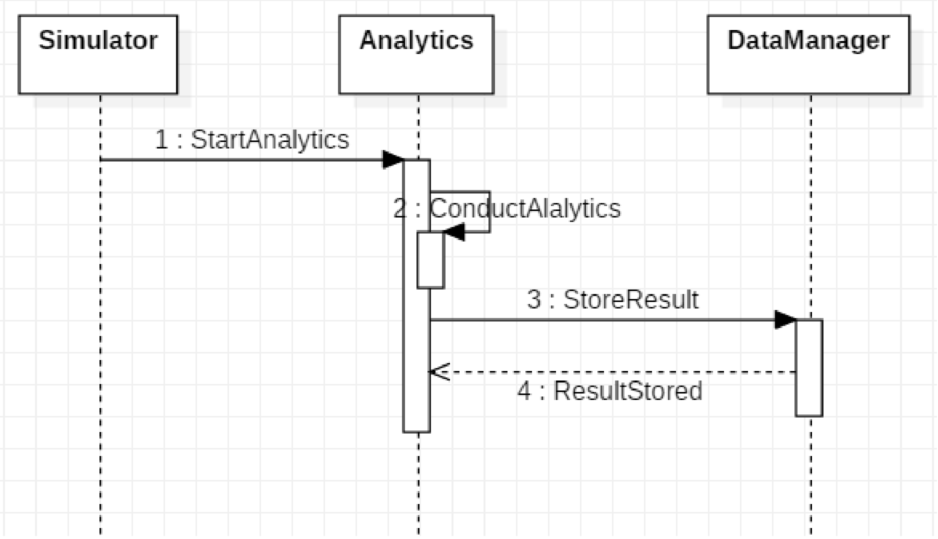


**Figure 23. Sequence diagram of using a simulator to test possible innovativeness of Enterprise**

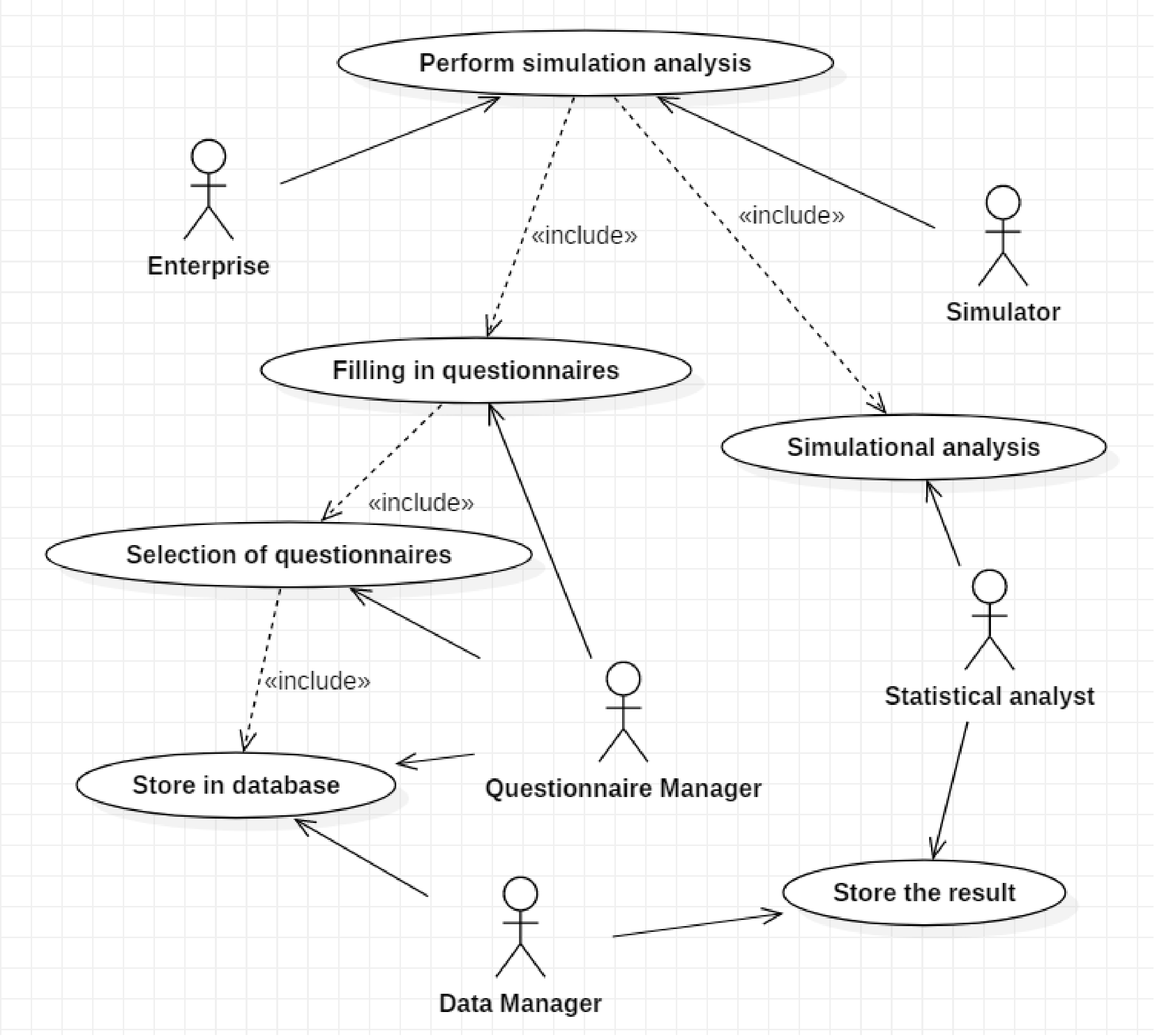


**Figure 25. Use case of using a simulator to test possible innovativeness of Enterprise**

## Browsing the results of a Simulator use

**Implementation scenario**
**Preconditions**
1. The Enterprise must be signed in;
2. Results of preliminary assessment of innovation through a simulator are available.

**Steps to Implement**.
1. The enterprise calls the Simulator for query results;
2. The Simulator selects the results of the Data Manager;
3. The results provided by the Data Manager are displayed to the enterprise.

**Post conditions**
1.  The results of the simulation analysis are examined.

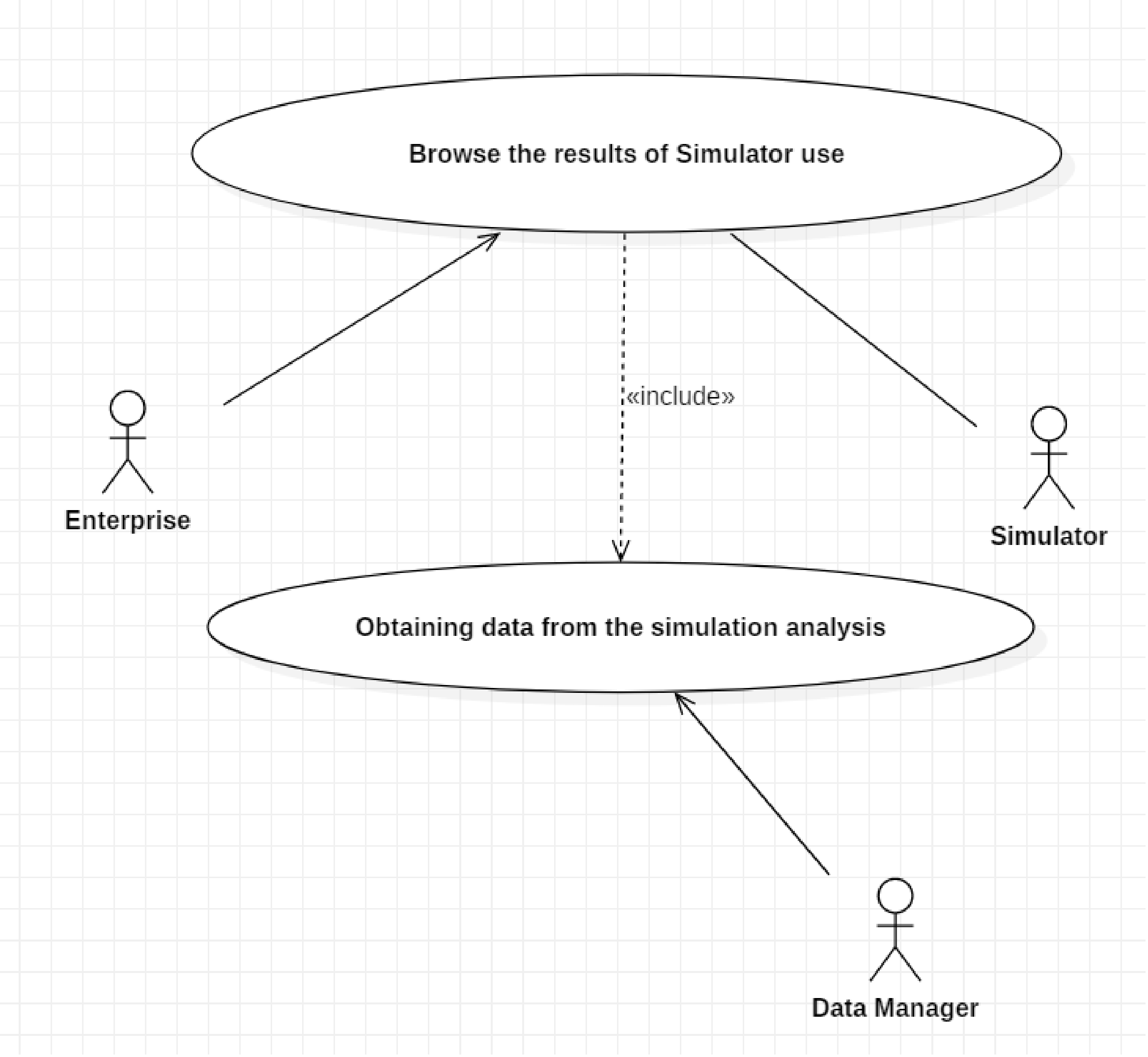

Browse the results of Simulator use
«include»
Enterprise
Simulator
Obtaining data from the simulation analysis
Data Manager

**Figure 26. Use case of Browsing the results of a Simulator use.**

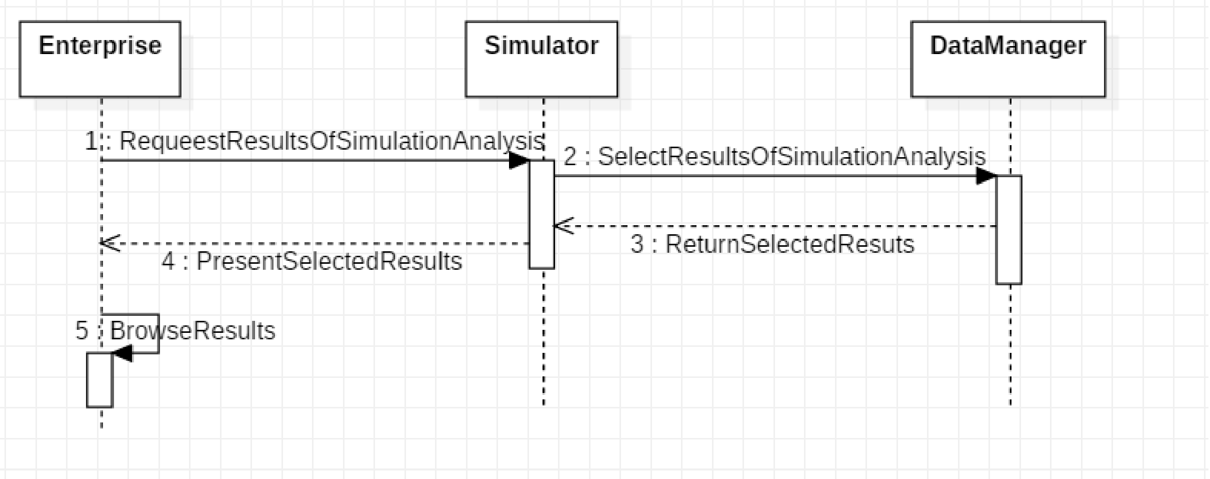

Enterprise
Simulator
DataManager
1 : RequeestResultsOfSimulationAnalysis
2 : SelectResultsOfSimulationAnalysis
3 : ReturnSelectedResuts
4 : PresentSelectedResults
5 : BrowseResults

**Figure 27. Sequence diagram of Browsing the results of a Simulator use**

## Get advises for improving the innovative process

**Implementation scenario**
**Preconditions**
1. The Enterprise must be signed in.
**Steps to Implement**.
1. The Enterprise calls the Advisor on Improving Innovation to get a advise to improve innovation;
2. The Advisor on Improving Innovation compares the results of the enterprise with the results of similar enterprises that have better innovations results;
3. The enterprise receives advice.
**Post conditions**
1. The Enterprise has received advises for improving innovation process.

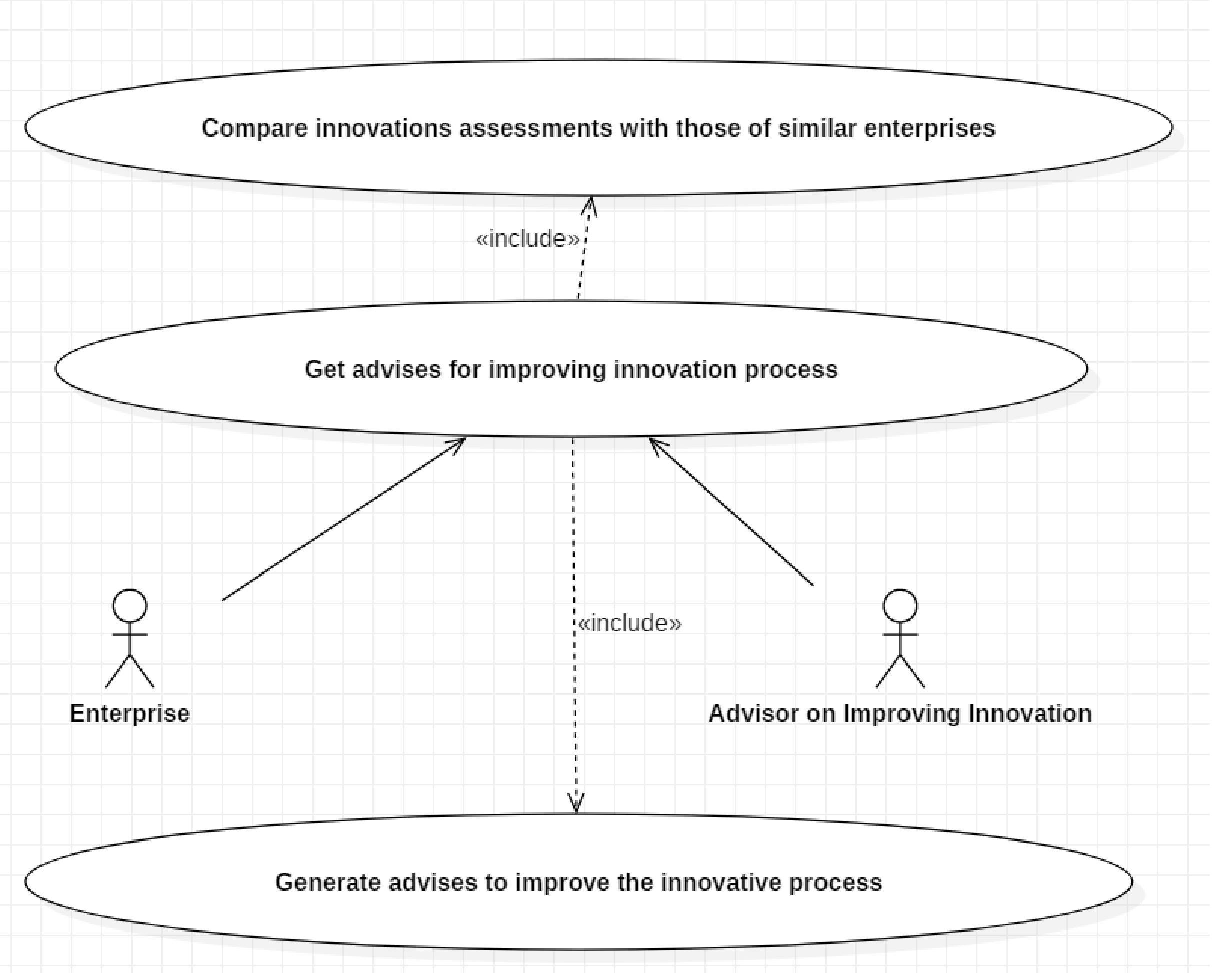

Compare innovations assessments with those of similar enterprises
«include»
Get advises for improving innovation process
«include»
Enterprise
Advisor on Improving Innovation
Generate advises to improve the innovative process

**Figure 28. Use case of get advises for improving the innovative process.**

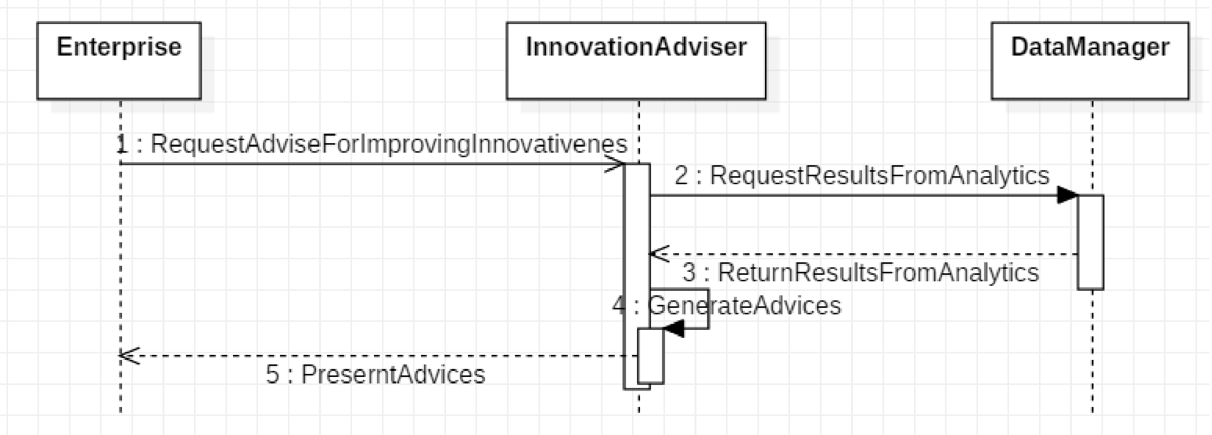

Enterprise
InnovationAdviser
DataManager
1 : RequestAdviseForImprovingInnovativenes
2 : RequestResultsFromAnalytics
3 : ReturnResultsFromAnalytics
4 : GenerateAdvices
5 : PreserntAdvices

**Figure 29. Sequence diagram of get advises for improving the innovative process.**

## Communication with other enterprises

**Implementation scenario**
**Preconditions:**

1. The Enterprise must be signed in..
2. The Enterprise addressee wants to make a connection.

**Steps to Implement**.

1. An enterprise sends a request to Communicator for communication with a particular enterprise;
2. The Communicator provides a list of enterprises wishing to establish a link;
3. The Enterpris chooses a communication partner;
4. The Communicator sends an invitation to communicate with a selected enterprise;

**Post conditions**

1. Communication between enterprise has taken place.

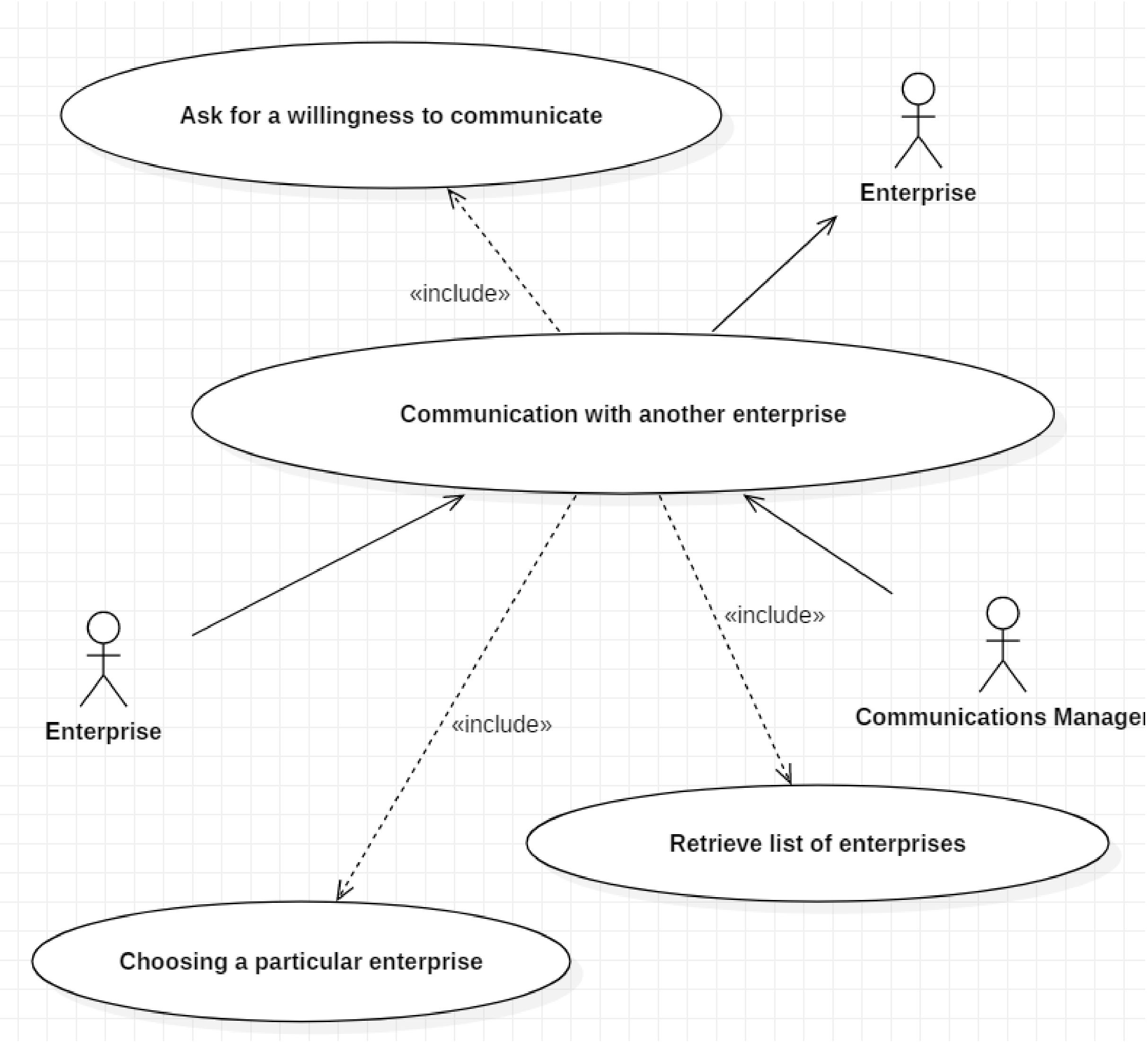

Ask for a willingness to communicate
Enterprise
«include»
Communication with another enterprise
«include»
Enterprise
«include»
Communications Managei
Retrieve list of enterprises
Choosing a particular enterprise

**Figure 30. Use case of communication with other enterprises**

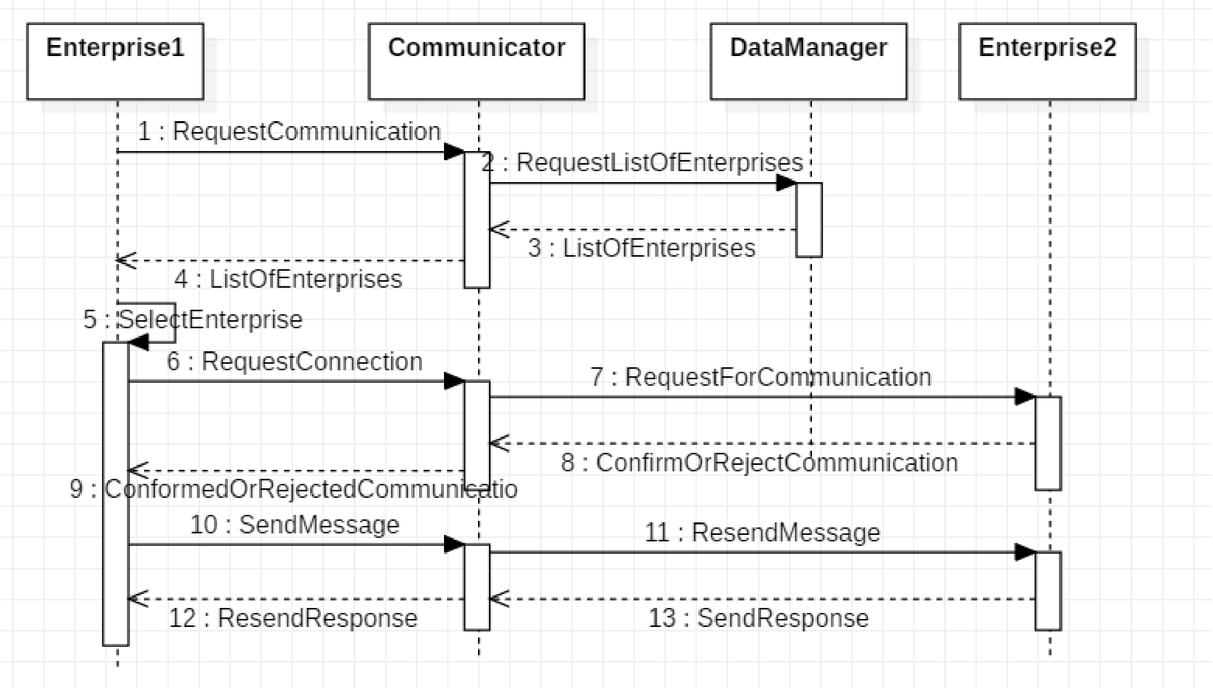


**Figure 31. Sequence diagram of communication with other enterprises**

## Object model  of the system

### Paradigm of object design of the system

The main paradigm of the platform's software design is the so called Model View Controller (MVC)) (Figure 32).
The model contains all application data and has the ability to manipulate this data. It does not interact with the View but communicates with the Controller.
The Wiew is everything you see in the browser. Typically, HTML, CSS, and Javascript. There are no references to the Model but it receives data for visualization on request from the Controller.
The C  ontroller contains business logic and communicates in two ways with the View. It also sends data to the Model as well as data visualization requests.

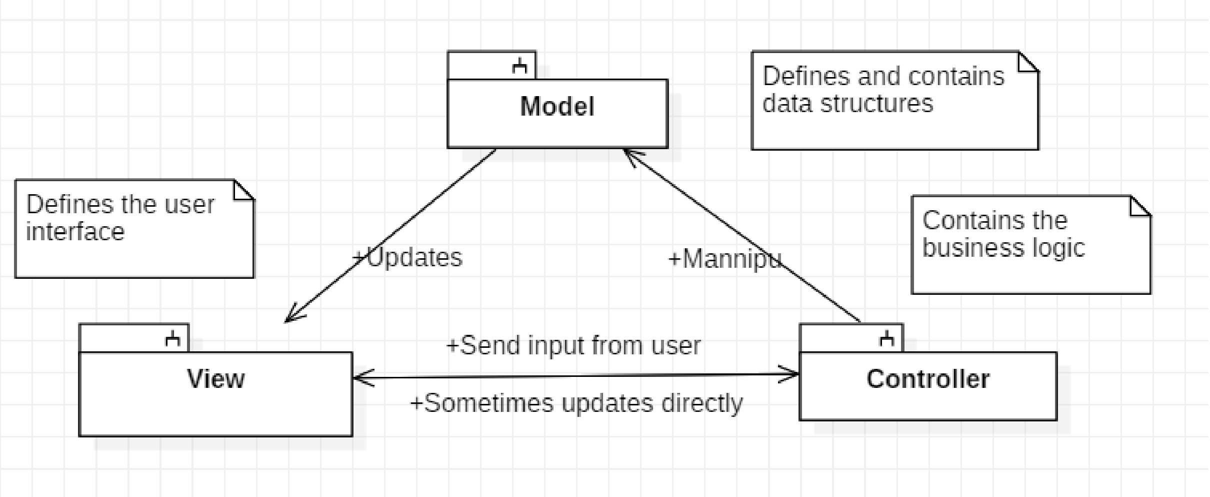

Model
Defines and contains data structures
Defines the user interface
+Updates
+Mannipu
Contains the business logic
View
+Send input from user
+Sometimes updates directly
Controller

**Figure 32. Object diagram of design pattern Model View Controller.**

### Object diagram  of the Model

The model is an implementation of the actor Data Manager.
The model consists of the following classes and subsystems:
- The Model Manager class manages bass communication. Returns to the DBAccessFactory class factory to instantiate a DBAccess object by which it communicates with the selected database;
- DBAccessFactory - creates a MySQLAccess or MongoDBAccess object depending on the input parameter;
- MySQL Relational Database, which contains enterprise data, questionnaires, administrative information;
- Non-relational database MongoDB, which contains the answers to the questions that are used for analytical calculations;

- Abstract class DBAccess, which is base class for the specific MySQLAccess and MongoDBAccess classes, which communicate with the corresponding databases;
- An abstract factory that creates objects from the inheritance classes of DBAccess. In this way, independence from the specific database is achieved through a polymorphic approach to their methods.

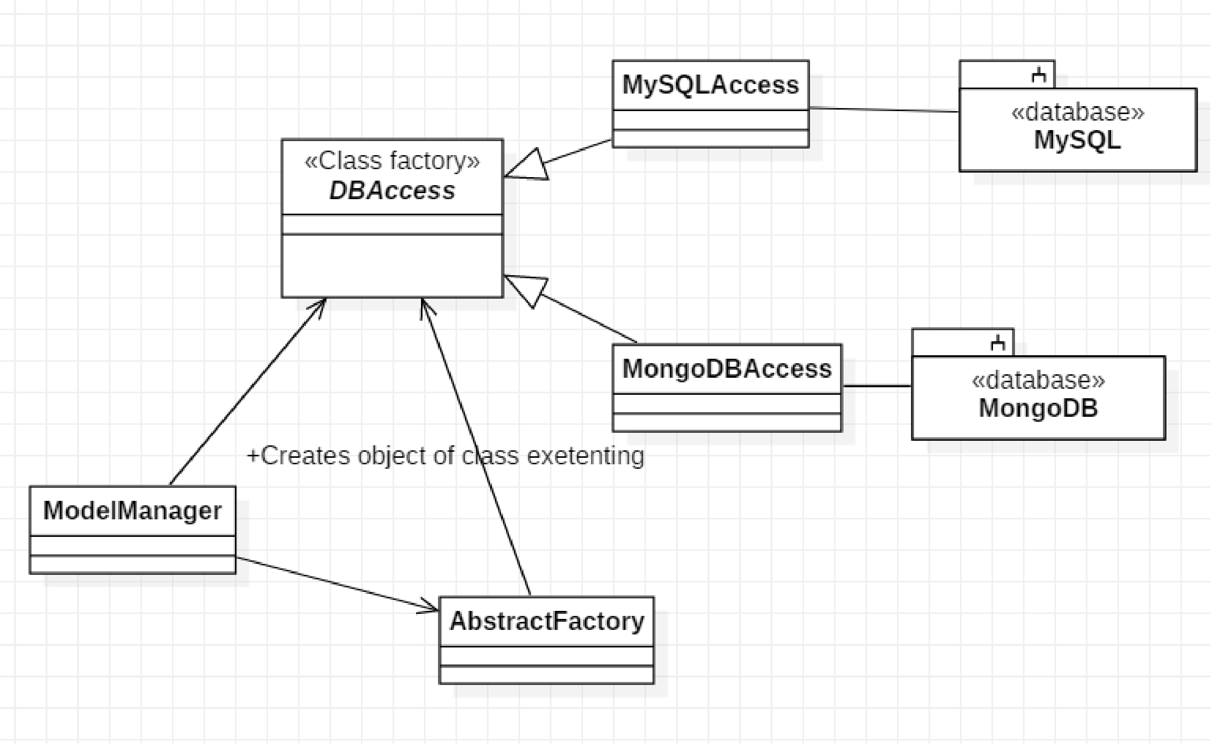


**Figure 33. Object diagram of Model.**

On Fig. 34 is presented a sequence diagram of operation with the Model.

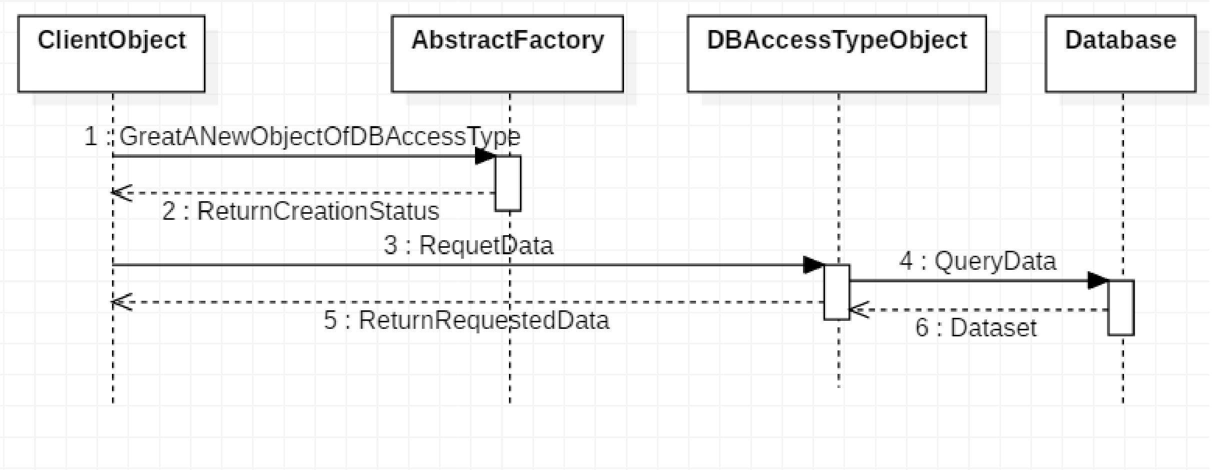


**Figure 34. Sequence diagram of operation with the Model.**

## Object diagram of the View

The View subsystem consists of four packages (Fig.35):
1. EnterpriseUI Interface Packet;
2. Interface Pack with AdministratorUI Admins;
3. Packaging for Internationalization Internationalization;
4. A open source package that is used by the above three OpenCardWebInterface.

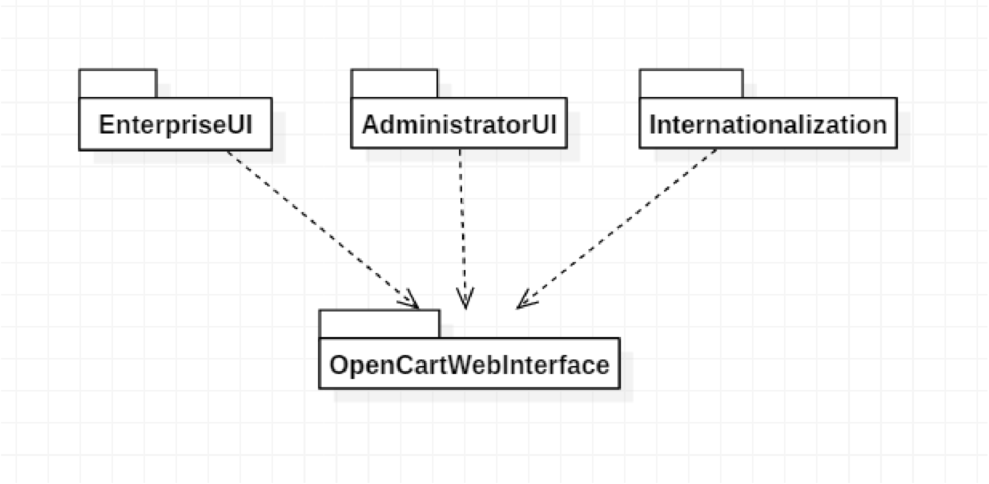


**Figure 35. Object diagram of View.**

## Object diagram of the Controller

The Controller Object Diagram is shown in Fig. 36.
The packets that are included in the controller are:
- RightsManager - Manages access rights for all actors;
- AccessManager - checks permissions for logging into the system, as well as registering it;
- QuestionaireManager - manages access storage and iterations with enterprise questionnaires;
- StatisticalDataAdapter - designed to adapt data from external vendors in a single internal form;
- Simulator - serves to play what-if situations where enterprises ask test data to play different variants to achieve better innovation;
- StatisticalDataAcquirer provides input of statistical information into the system;
- StatisticalAnalyst - conducts enterprise innovation analysis;
- InnovationAdviser - generates advice to SMEs to improve their innovation process;
- ContentManager - serves to manage the content of the web-based system;
- ReportGenerator - generates reports for the Chief Administrator;
- Communicator - communicates between registered enterprises.

The work of these packages is managed and coordinated by the Coordinator package.

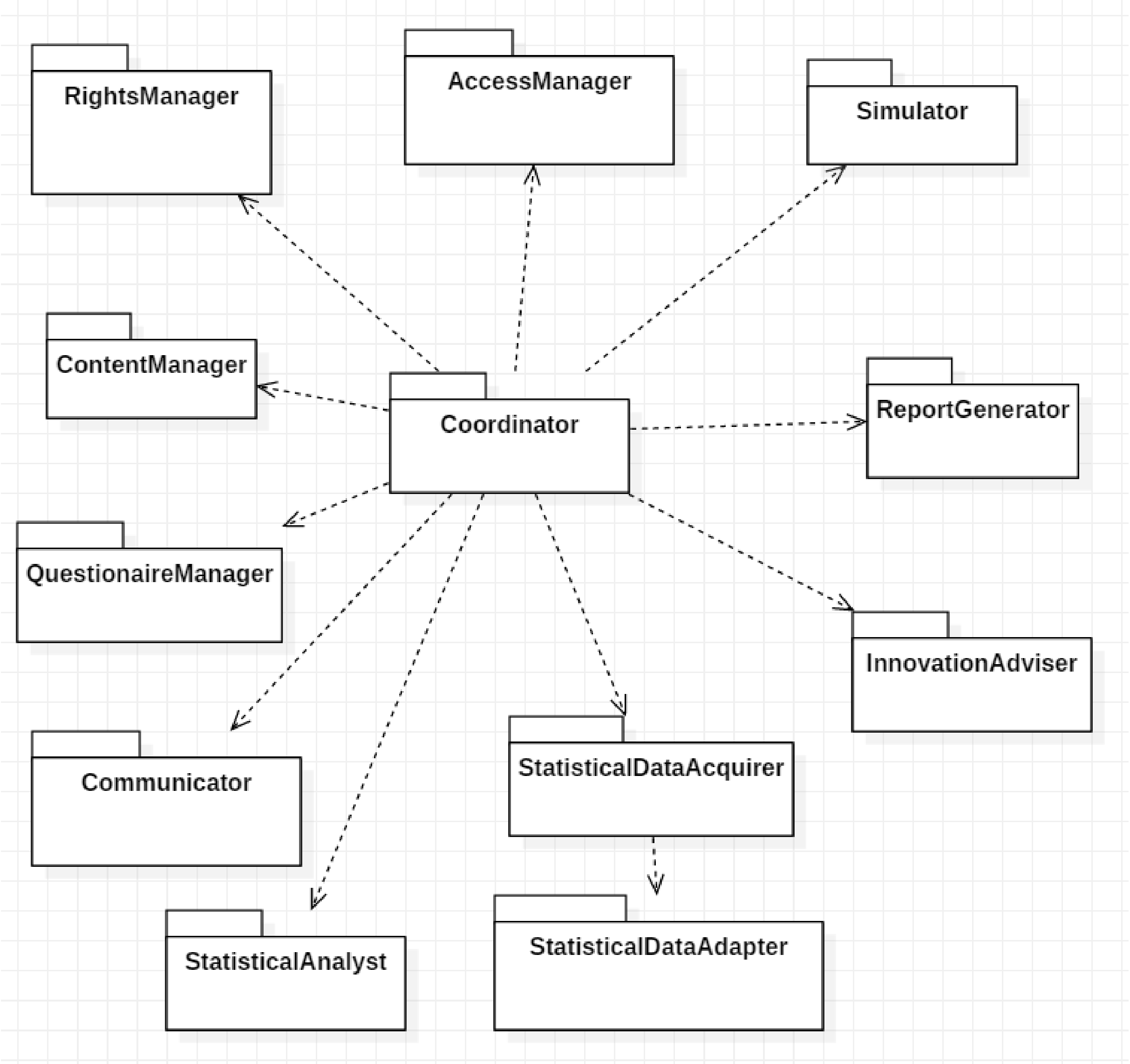


**Figure 36. Object diagram of Controller**

## Object diagram of Coordinator

The Coordinator's responsibilities are:

- Receive asynchronous queries to execute commands from external actors;
- Syntactic and semantic analysis of received commands;
- Distribution of the execution of the received commands.

The Coordinator package consists of the following main classes:

2. OperationRequest, which is a stereo signal. Objects in this class contain a request for an operation. Requests for operations are generated by external actors and are asynchronous;

3. InputRequestsReceiver is a class that contains a basic public AcceptCommand () method. This method consists of one main thread that expects an asynchronous command, ac a signal - object of type OperationRequest that contains the code of the desired operation and parameters. Upon receiving a signal in a new thread, a RequestProcessor object is created;
4. RequestProcessor - it's responsibility is to process the request. This is done by first creating a RequestInterpreter object and then RequestDispatcher;
5. RequestInterpreter class verifies syntactically the correctness of the query, as well as doing semantic analysis. In the case of an error, the error message is displayed through the View;
6. RequestDispatcher - serves to transmit the command to the appropriate package that is responsible for its execution.

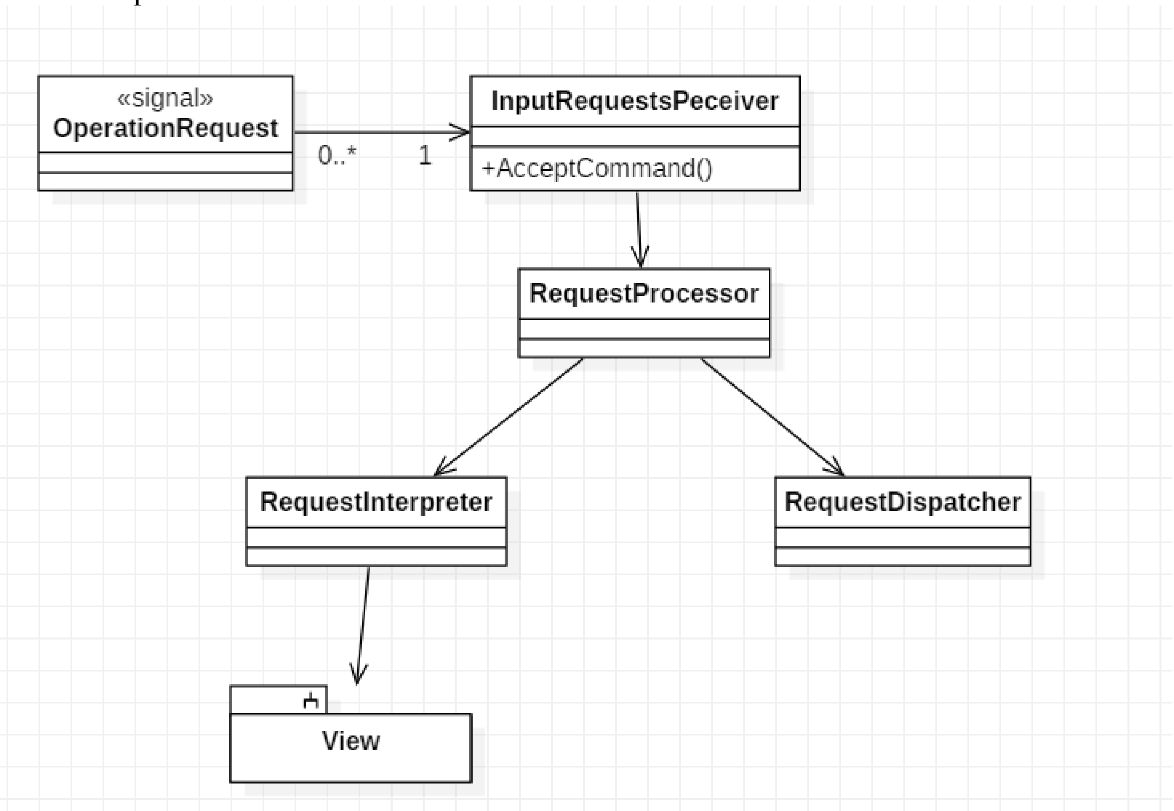
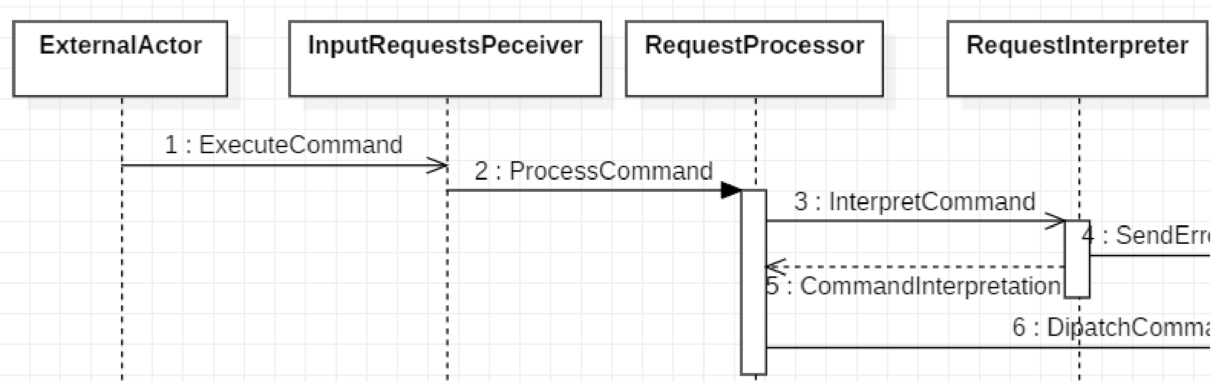


**Figure 37. Object diagram of the Coordinator.**



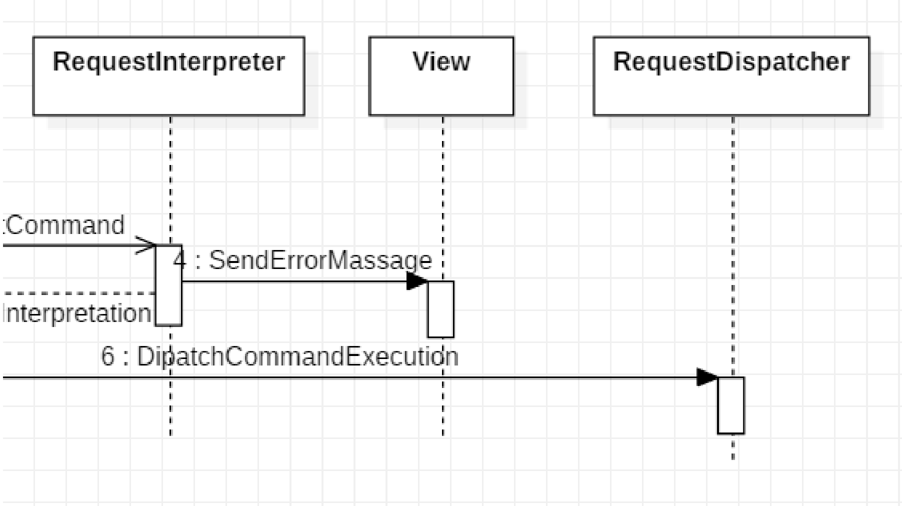


**Figure 38. Sequence diagram of when a command request arises**

For the purpose of simplifying sequence diagrams, we will skip the Coordinator calls  the Coordinator, and we will show operations running directly with objects in the appropriate classes.

## Object diagram of RightsManager

The RightsManager package is a software implementation of the Actor Rights Manager.
The RightsManager's responsibilities are:
- Maintain an up-to-date list of all rights of access to resources of system actors;
- Maintain an up-to-date list of all access rights to system actors' functions;
- Return specific rights related to specific actors upon request.

Actor's rights are contained in a container. Upon editing administrator access rights, these changes are reflected in the container and the database. The Access Rights container serves as a cache for quick access to registered actors' rights. The rights themselves as such are stored in the database. Each object in the container contains the actor's identification and the right to access a resource or function. The actors can be both external and internal.
The RightsManager consists of the following classes:

- RightsProcessor – accepts requests for verification of external actors' access rights. It has a basic CheckRight () method through which requests are received. This method performs tracking activities:
  - Checking the correctness of the request;
  - Checking the availability of the container Rights;
  - If it does not exist, it creates a container of type Rights and fills it with objects of the type ResourceAccessRight и FunctionAccessRight;
- Iterates the contents of Rights to verify the actor's rights;
- Rights – container of objects of class Right;
- ResourceAccessRight a class that contains an external actor ID and a right to access a particular resource;

- FunctionAccessRight a class that contains an external actor ID and a right to access a particular feature.

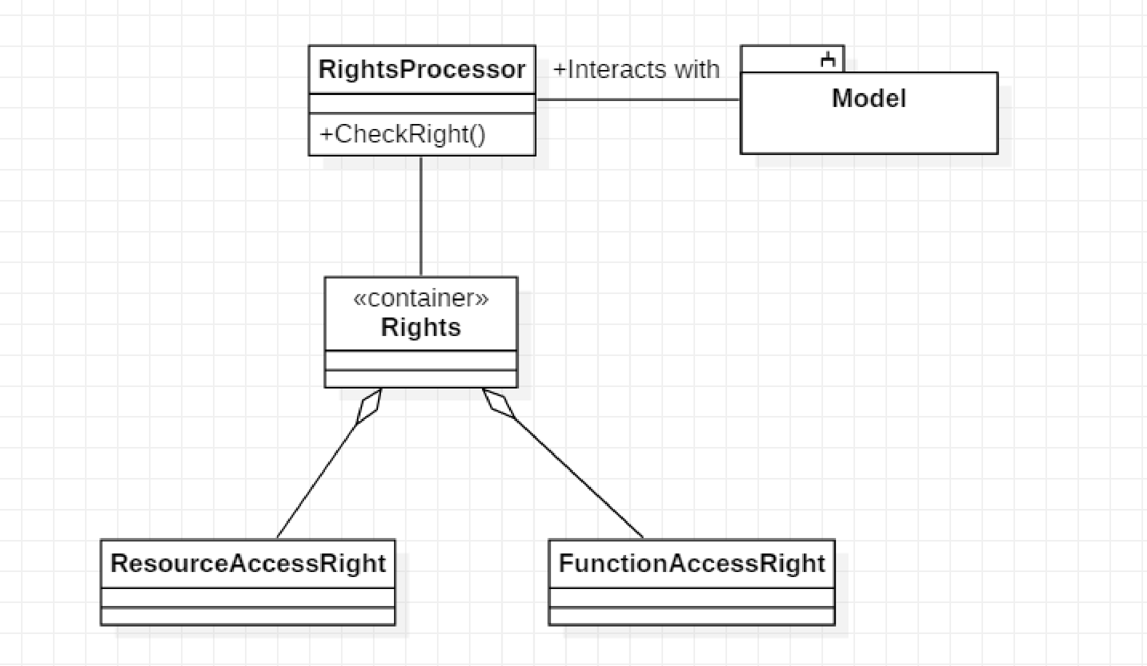


**Figure 39. Object diagram of RightsManager.**

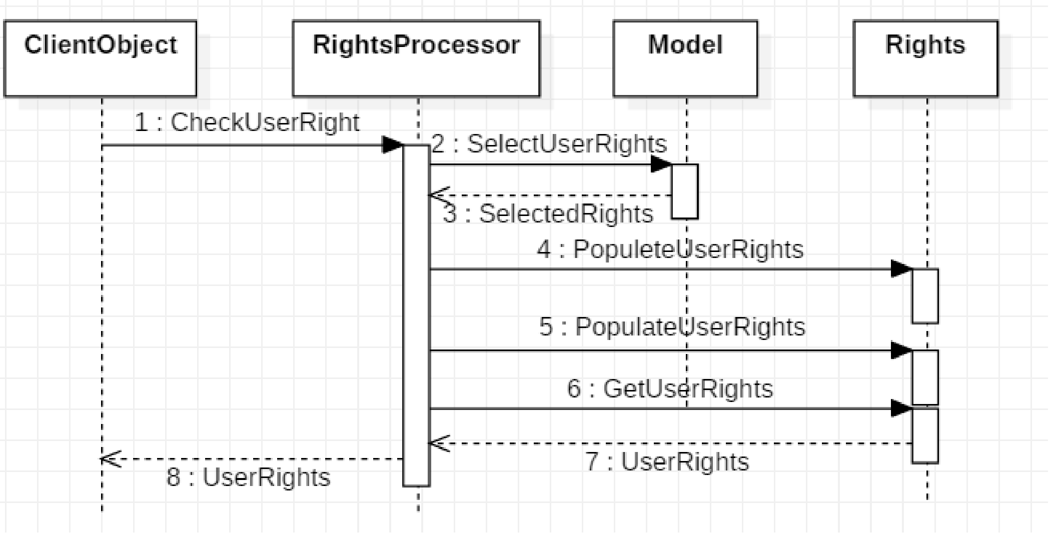


**Figure 40. Sequence diagram of obtaining the rights of an external actor**

## Object diagram of QuestionnaireManager

The QuestionnaireManager package is a software implementation of the actor Questionnaire Manager.
The responsibility QuestionnaireManager is:

- Providing questionnaires for completion;
- Storing data from completed questionnaires in a database;

- Creating new questionnaires;
- Editing questionnaires.

The QuestionnaireManager consists of the following classes:

- QuestionnaireManager - allocates tasks for creating and modifying questionnaires, maintaining, storing data from completed questionnaires, maintaining the container of steady state questionnaires;
- *QuestionnaireIterator* is a class for iteration of container-grade Questionnaires, it serves as a facade for the specific implementation of this container class;
- Questionnaires is a container class containing objects of type Questionnaire: Communicates with the Data Manager to store and download questionnaires;
- Questionnaire is a class that contains closed questions, possible answers, and their assessments;
- QuestionnaireCreator - a class responsible for creating new questionnaires;
- QuestionnaireEditor - a class responsible for editing questionnaires;
- QuestionnaireFiller - a class responsible for completing questionnaires.

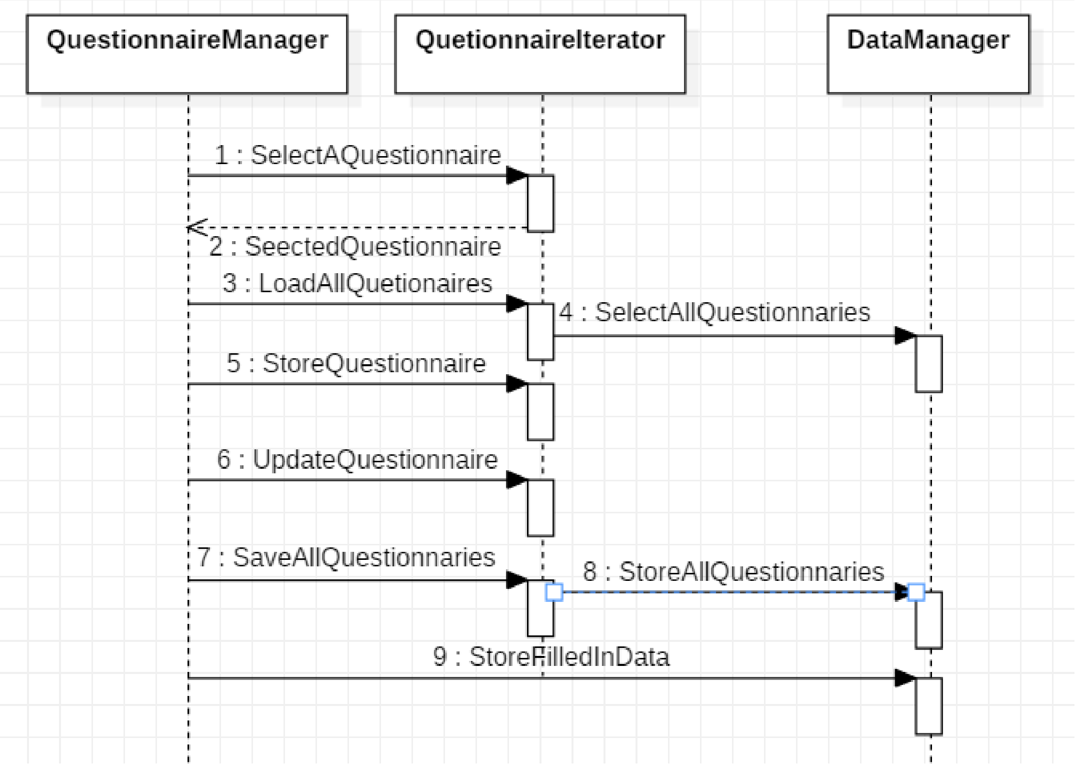


**Figure 41. Sequence diagrams when loading questionnaires and storing questionnaires and data.**

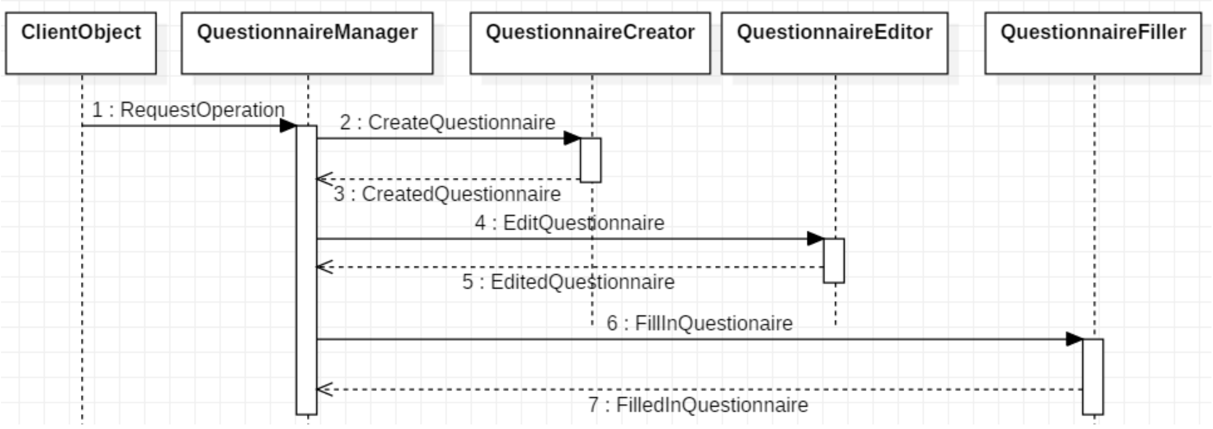


**Figure 42. Sequence diagrams for creating, editing, and filling in questionnaires**

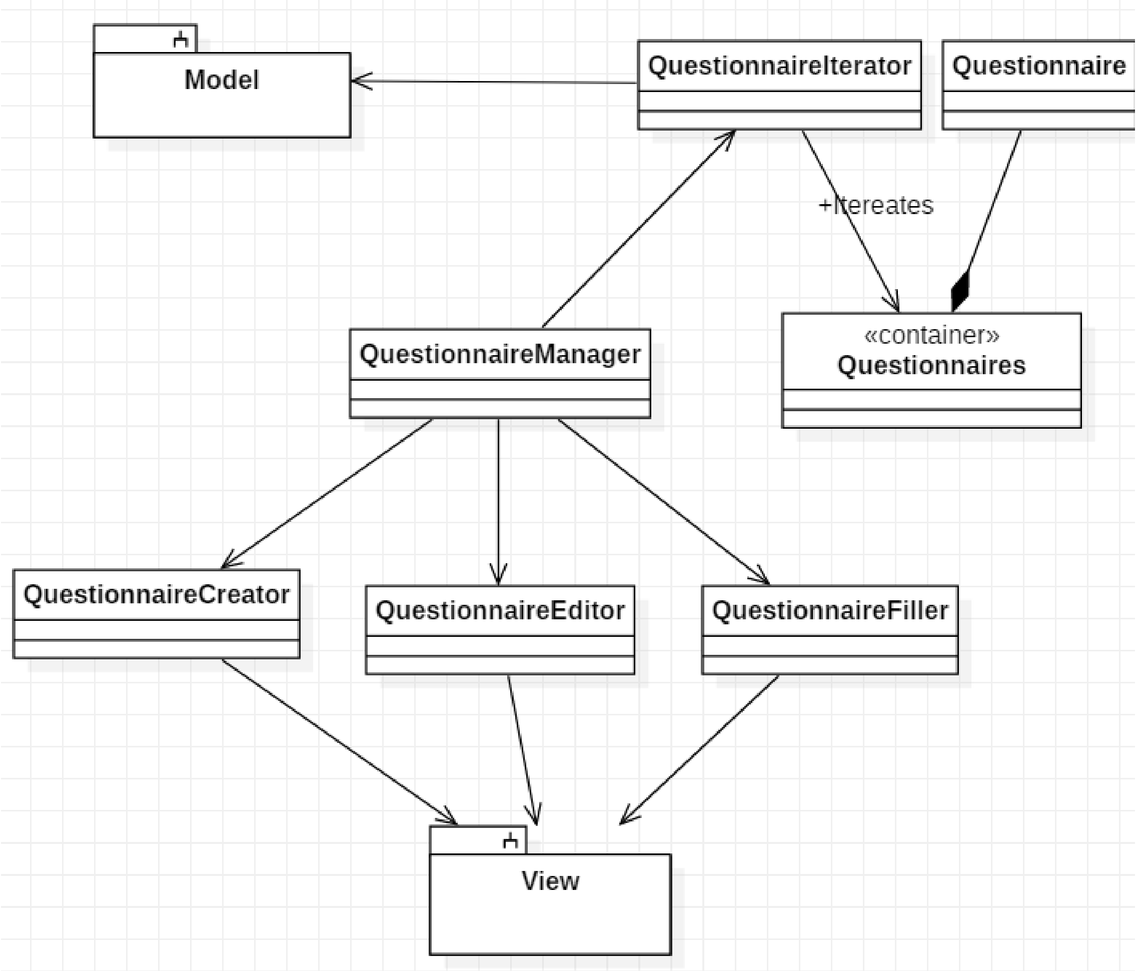


**Figure 43. Object diagram of QuestionnaireManager**

## Object diagram of AccessManager

AccessManager package is a software implementation of actor Access Manager.

The responsibility of AccessManager is to check whether an outside actor has access to the system.

The AccessManager consists of the following classes:

- AccessProcessor - handles queries for verifying, adding, and editing access rights. The editing also includes disabling access. In order to maintain information completeness, access rights are not erased, but only deactivated;
- Credentials is a container containing Credential objects - Its responsibility is to maintain a list of credentials; The class is also responsible for storing rights to access the system in a database;
- Credential is a class that contains actor's details as a username and password, and whether it is active;
- AddCredential is a class that creates a new Credential object;
- ChangeCredential is a class that edits a Credential object.

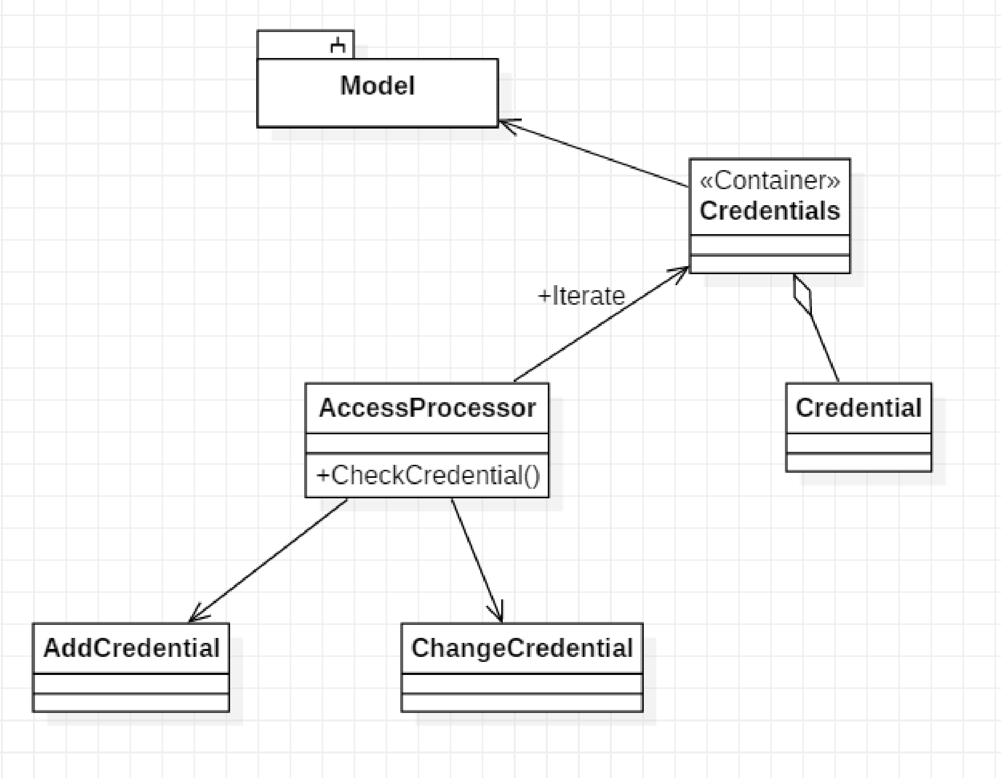
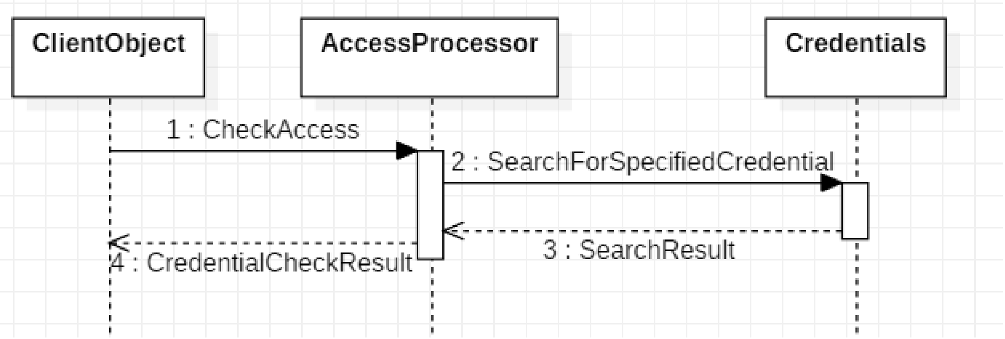


**Figure 44. Object diagram AccesssManager.**



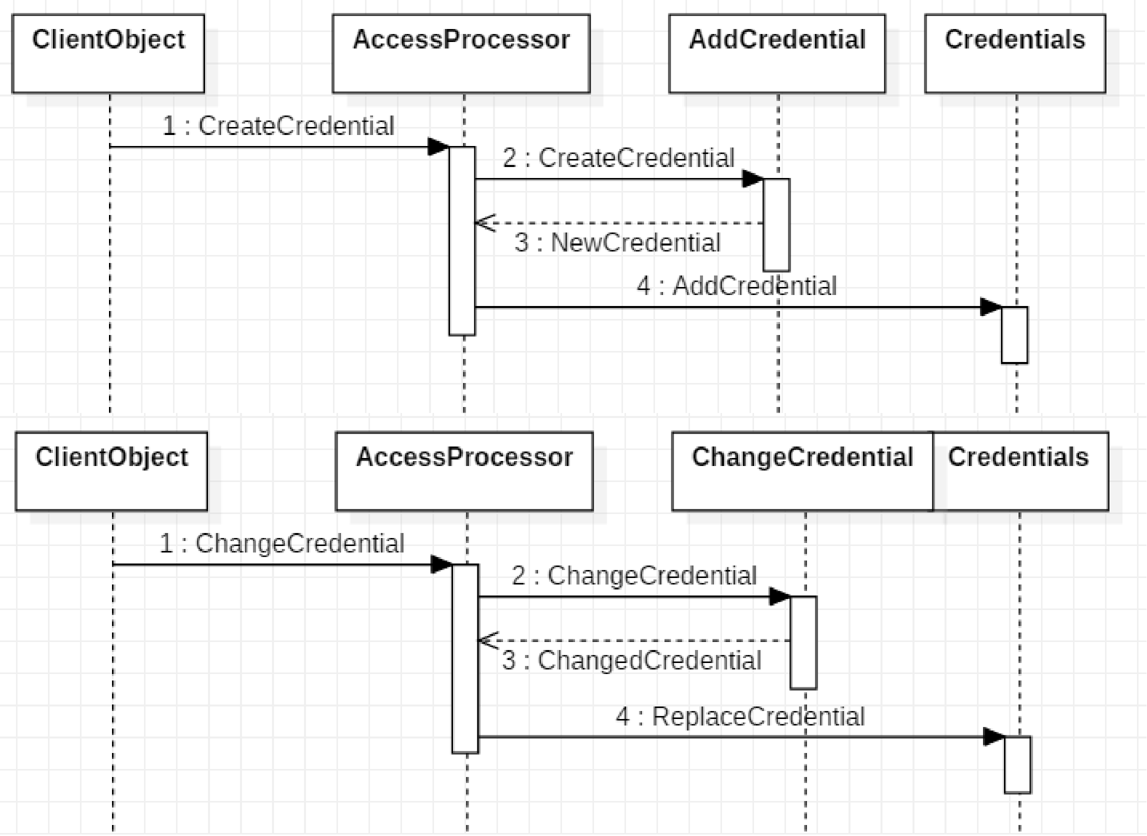


**Figure 45. Sequence diagrams for check, add and modify system actor's access to the system.**

## Object diagram of Communicator

Communicator is a software implementation of actor Communications Manager. It is the responsibility of the Communications Manager to establish a link between registered enterprises.

The Communicator consists of the following classes:

- CommunicationProcessor - processes requests for communication between registered enterprises;
- Communications is a container containing communication objects;
- Communication is a class that contains data about established communications, such as enterprise-related data, an archive identifier containing the exchanged messages, and messages from the current session when it is started;
- CommunicationCreator - creates a relationship between two enterprises;
- CommunictionConductor - communicates messages after establishing a connection between two enterprises.

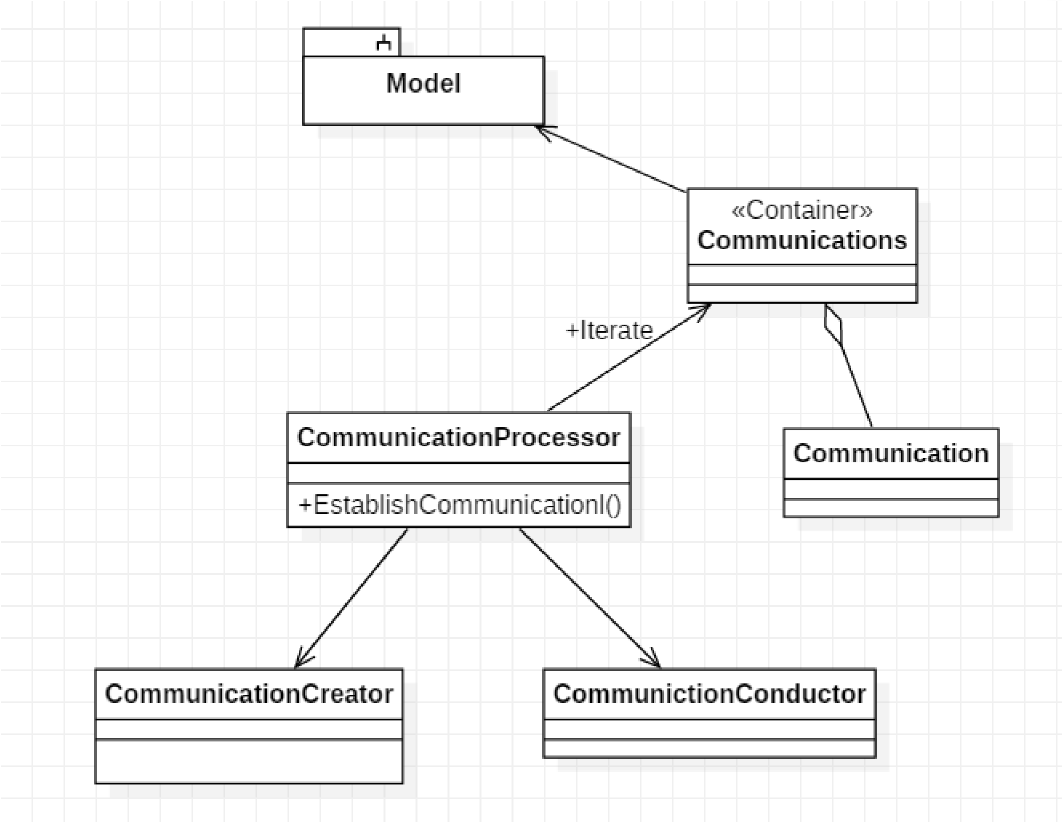


**Figure 45. Object diagram of Communicator**

.

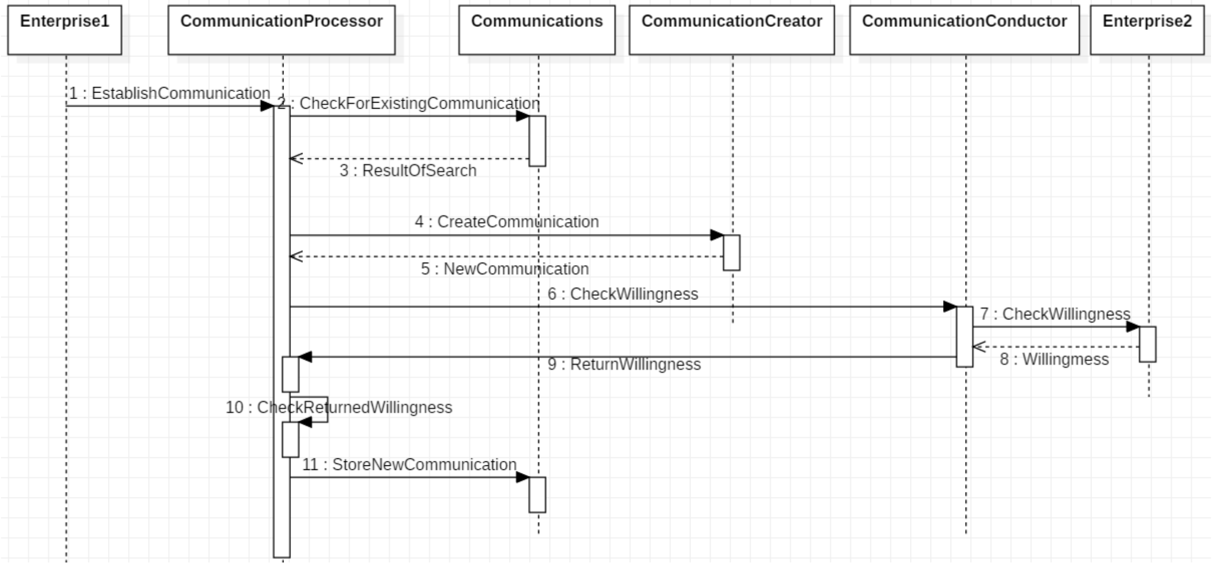


**Figure 46. Sequensce diagram for creation a communications connection**

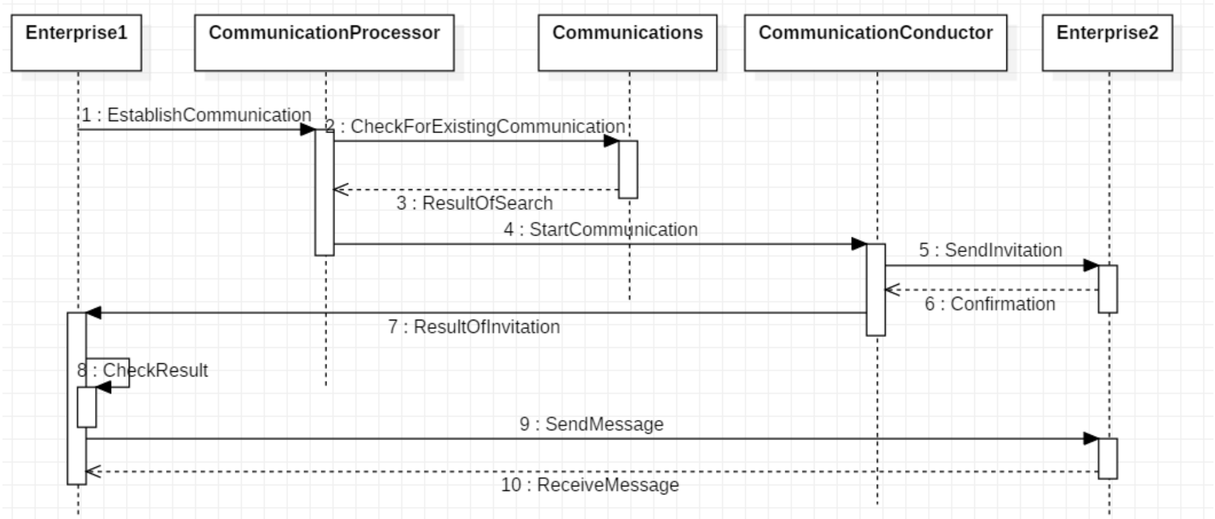


**Figure 47. Sequensce diagram for communicating after creation of communications connection**

## Object diagram of StatisticalDataAcquirer

The StatisticalDataAcquirer is a software implementation of actor Statistical Data Acquirer. The responsibility of the StatisticalDataAcquirer is to perform incoming syntactic control of the statistics produced by the National Statistical Institutes of the Republic of Albania, the Republic of Bulgaria, the Republic of Greece, Cyprus and the Republic of Macedonia.
The StatisticalDataAcquirer consists of following classes:
- StatisticalDataAcquirer - The responsibility of this class is to read input data generated by the partner countries' statistical institutes. Use a class of SyntaxAnalyzerFactory object of the appropriate type to perform a syntactic analysis of the incoming statistical information;
- SyntaxAnalyzerFactory - returns a type object specified with a type parameter that inherits the SyntaxAnalyzer base class;
- SyntaxAnalyzer - a base class of specific syntactic analysis classes from partner countries: It is used for polymorphous calling of objects from specific classes created by SyntaxAnalyzerFactory that will ensure the future expansion of the functionality of the system, adding the ability to obtain statistical information from other institutions or countries;
- ALSyntaxAnayzer, BG SyntaxAnalyzer, CYSyntaxAnalizer, GRSyntaxAnalizer, MKSyntaxAnalyzer - inheritance classes SyntaxAnalizer.

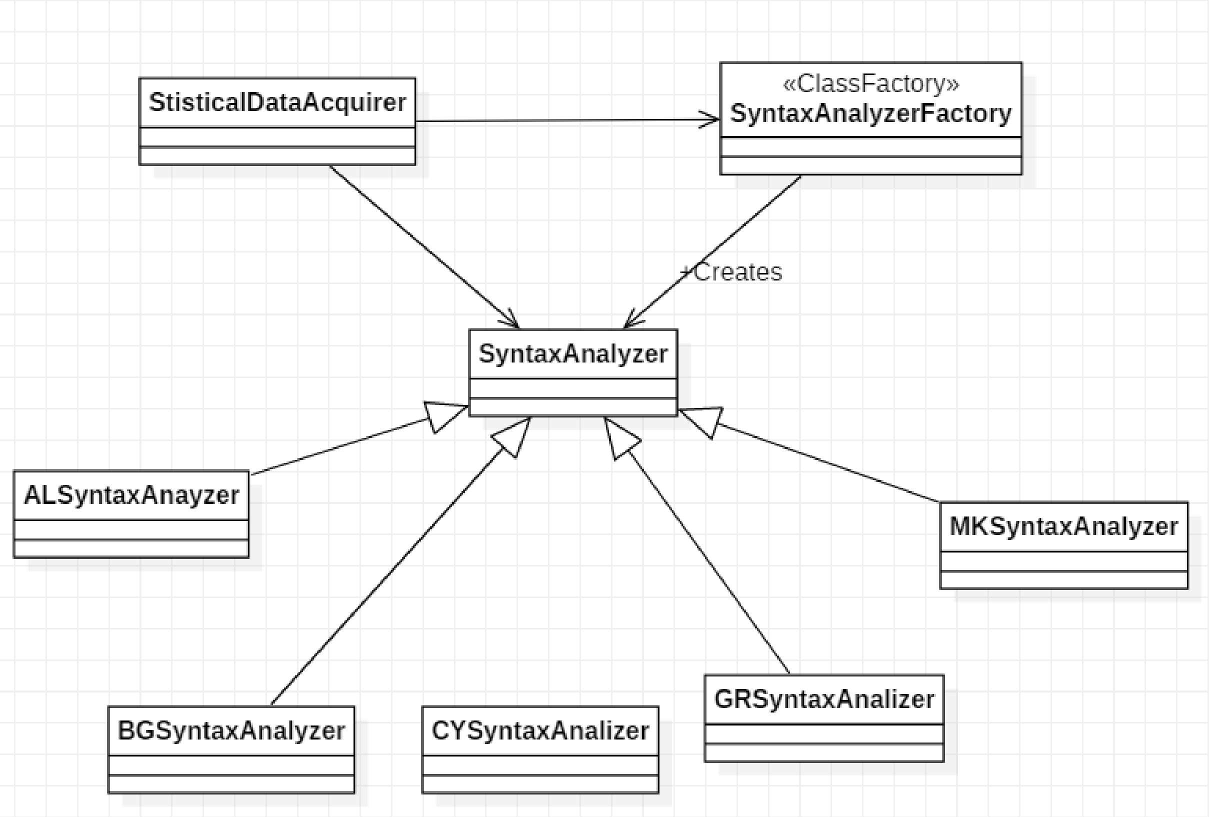


**Figure 48. Object diagram of StatisticalDataAcquirer.**

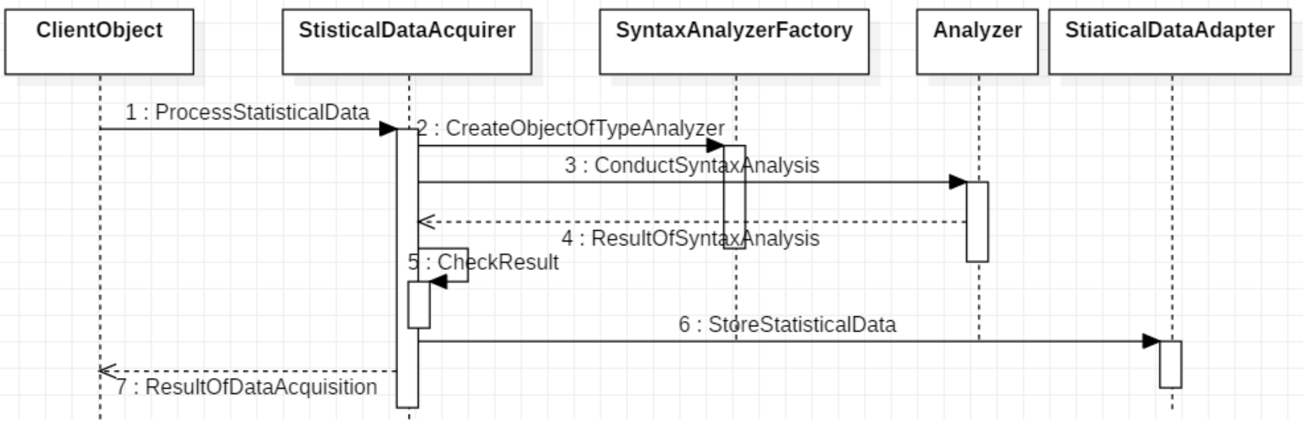

ClientObject
StisticalDataAcquirer
SyntaxAnalyzerFactory
Analyzer
StiaticalDataAdapter
1 : ProcessStatisticalData
2 : CreateObjectOfTypeAnalyzer
3 : ConductSyntaxAnalysis
4 : ResultOfSyntaxAnalysis
5 : CheckResult
6 : StoreStatisticalData
7 : ResultOfDataAcquisition

**Figure 49. Sequence diagram of statistical data acquisition**

## Object diagram of StatisticalDataAdapter

StatisticalDataAdapter is a software implementation of actor Statistical Data Adapter. StatisticalDataAdapter is responsible for transforming descriptions of statistics received from different institutions into a single internal format in which they are recorded in the database. StatisticalDataAdapter consists of following classes:

- DataReceiver - The responsibility of this class is to obtain syntactically correct statistical data generated by the partner countries' statistical institutes and to use an AdapterFactory object of the appropriate type to transform the incoming statistical information into a single internal presentation;
- AdapterFactory - returns a type object specified with a type parameter that inherits the base class Adapter;
- Adapter - a base class of specific classes for transforming statistics from partner countries into a single internal. Used for polymorphous call of objects from specific classes created by AdapterFactory. This will ensure the future expansion of the functionality of the system, adding the ability to obtain statistical information from other institutions or countries. Once the data is transformed into a unified presentation, it is stored in the database;
- ALAdapter, BGAdapter, GRAdapter, MKAdapter - inheritance classes of Adapter.

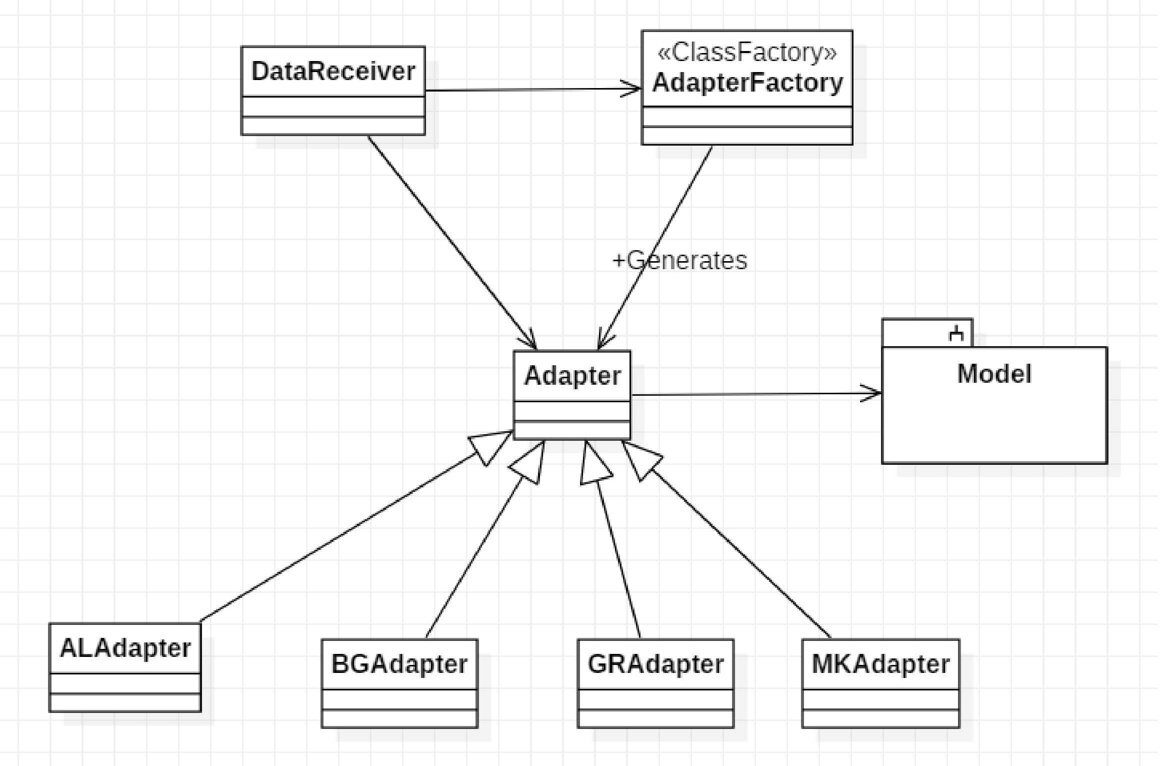


**Figure 50. Object diagram of Statistical Data Adapter.**

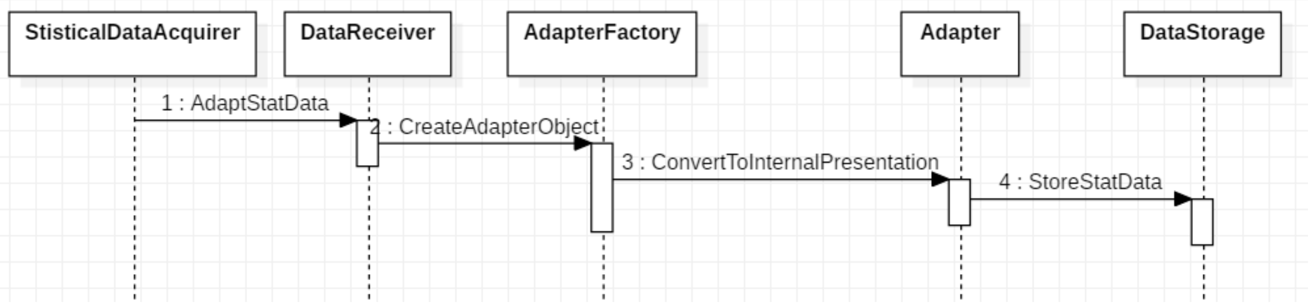


**Figure 51. Sequence diagram of transformation and storage of statistical data.**

### Object scheme of StatisticalAnalyst

The StatisticalAnalyst is a software implementation of actor Statistical Analyst.
The StatisticalAnalyst is responsible for multi-criteria decision-making and choice in the definition of innovative clusters by SMEs or on the basis of Inter-Criteria Analysis.
The StatisticalAnalyst consistes of following classes:
- AnalyticsProcessor - the responsibility of this class is to select data for analytical calculations: Use an object of the appropriate type to create multi-choice decision-making and choice in identifying innovative enterprise clusters;
- AnalyticsFactory - returns a type object specified with a type parameter that inherits the base class Adapter;
- Analytics - a base class of specific classes responsible for multi-choice decision-making and choice in identifying innovative clusters of enterprises. Used for polymorphic invocation of objects from specific classes created by AnalyticsFactory. This will ensure the future expansion of system functionality by adding decision-making and selection choices to defining innovative clusters of enterprises;
- MultyCriterialAnalysis, InterCriterialAnalysis - Multidisciplinary decision-making and selection classes for innovation clusters in enterprises and Inter-Criteria for Multi-Criteria Decision Making and Choice in Identifying Innovative Business Clusters.

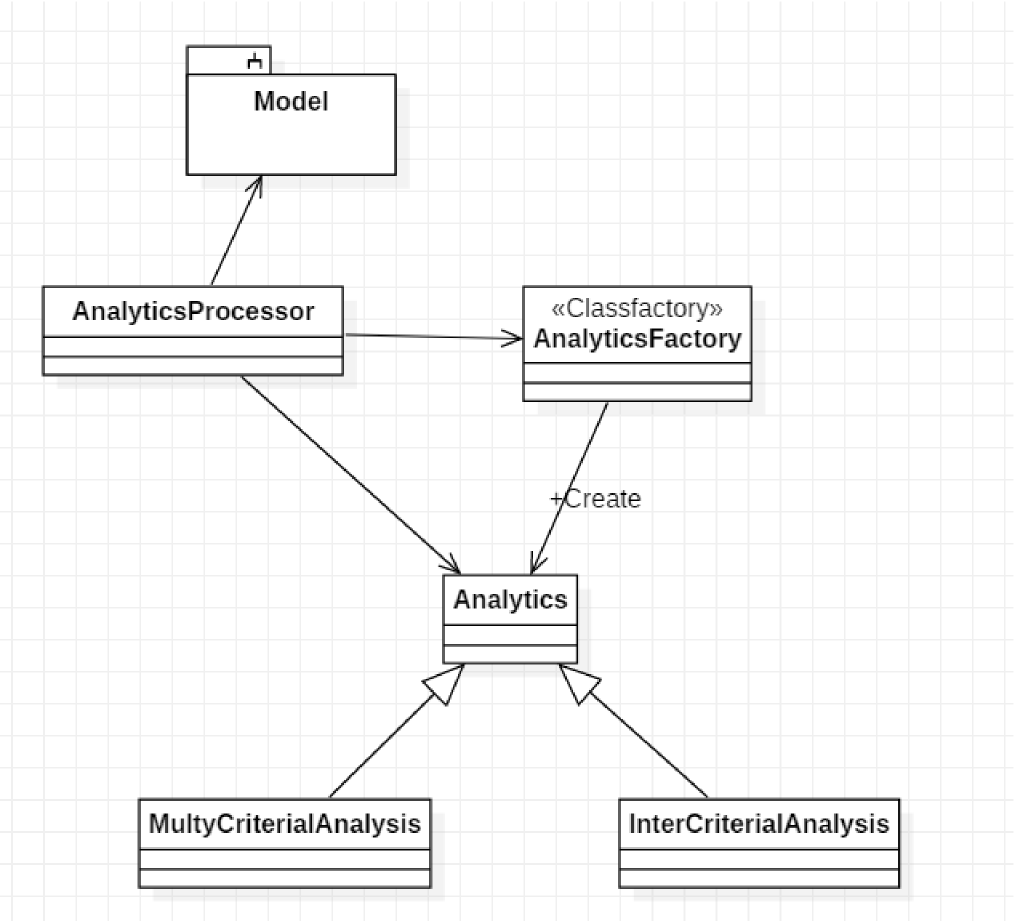


**Figure 52. Object diagram of StatisticalAnalyst.**

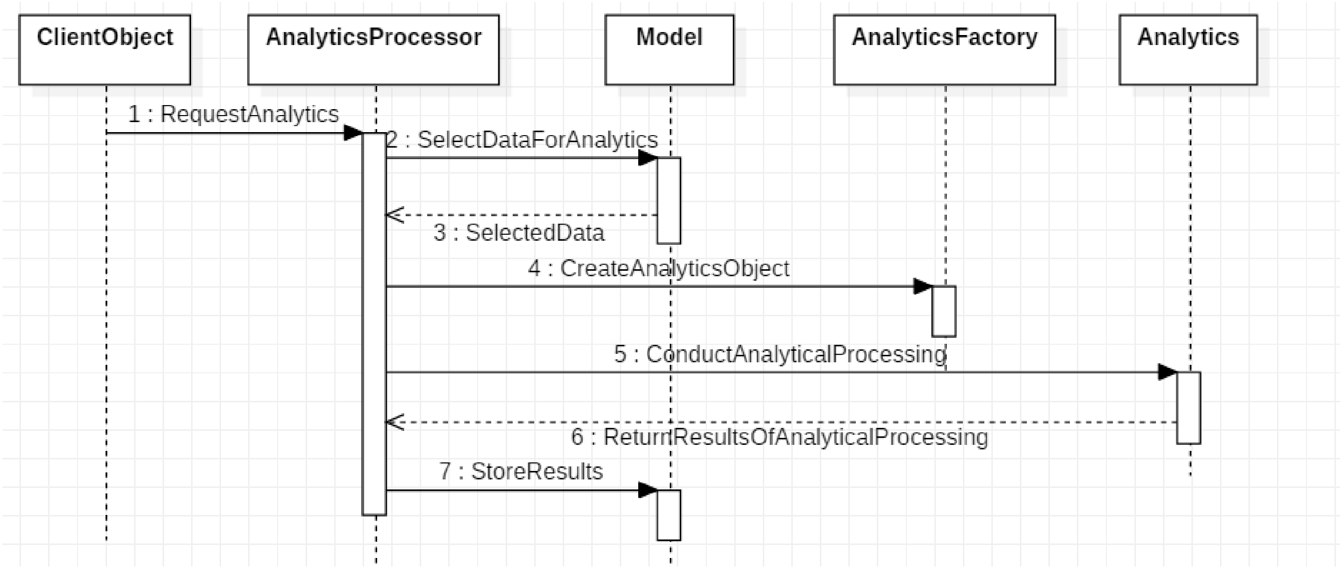

ClientObject
AnalyticsProcessor
Model
AnalyticsFactory
Analytics
1 : RequestAnalytics
2 : SelectDataForAnalytics
3 : SelectedData
4 : CreateAnalyticsObject
5 : ConductAnalyticalProcessing
6 : ReturnResultsOfAnalyticalProcessing
7 : StoreResults

**Figure 53. Sequence diagram of performing analytical calculations.**

## Object diagram of InnovationAdviser

InnovationAdviser is a software implementation of actor Advisor on Improving Innovation.
InnovationAdviser is responsible for generating advices for enterprises to improve their innovative process.
InnovationAdviser consists of the following classes:
- InnovationAdviser - this class is responsible for generating advice to businesses to improve their innovative process. Sessions are generated based on processing of the InnovationComparator class;
- InnovationComparator - is responsible for finding similar businesses of the same type, but with a better assessment of the innovative process.

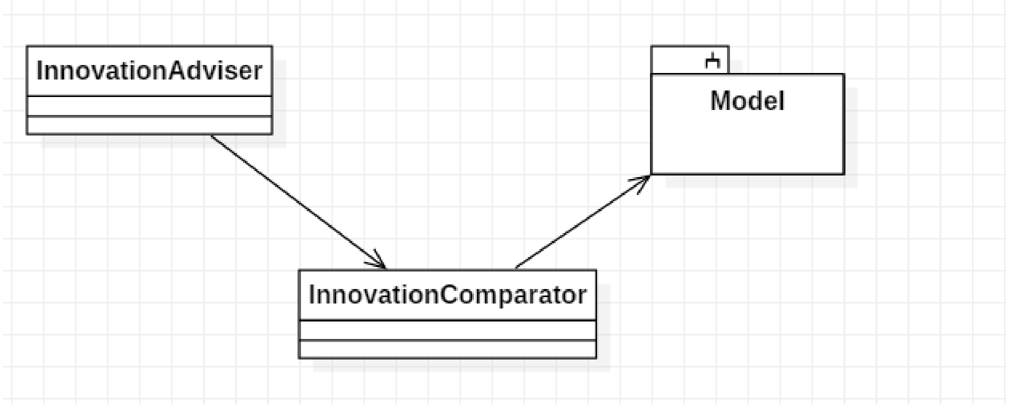


**Figure 54. Object scheme of InnovationAdviser.**

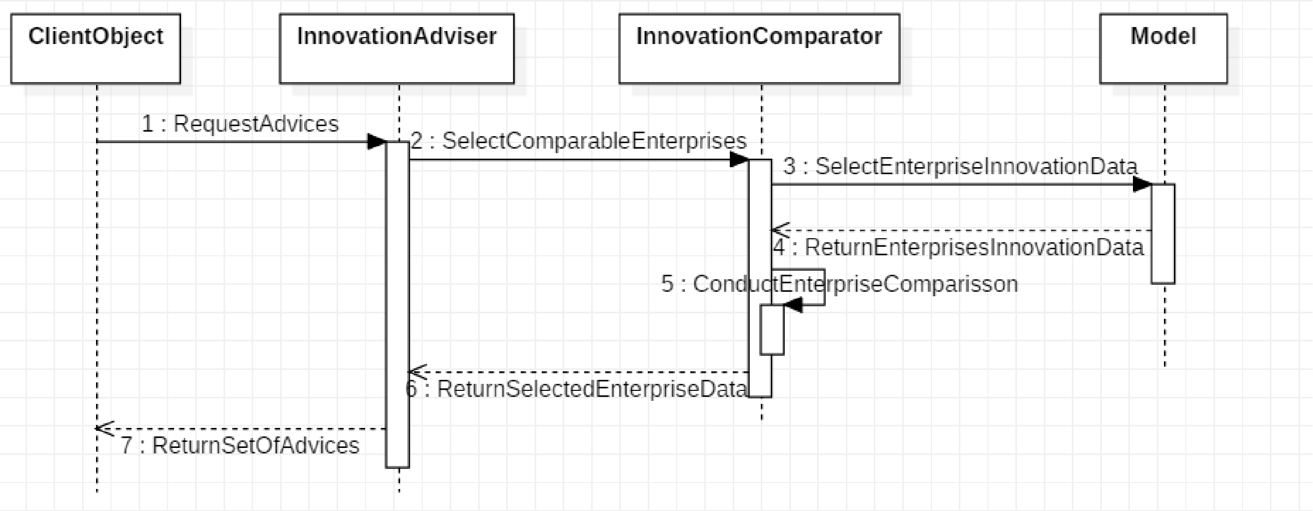

ClientObject
InnovationAdviser
InnovationComparator
Model
1 : RequestAdvices
2 : SelectComparableEnterprises
3 : SelectEnterpriseInnovationData
4 : ReturnEnterprisesInnovationData
5 : ConductEnterpriseComparisson
6 : ReturnSelectedEnterpriseData
7 : ReturnSetOfAdvices

**Figure 55. Sequence diagram of generating advices to improve the innovative process.**

## Object diagram of Simulator

Simulator is a software implementation of actor Simulator.
Simulator is responsible for playing different variants for assessing an enterprise's innovation - what if analysis.
Simulator consists of the following classes:
- Simulator - This class is responsible for conducting what if analysis. The class uses classes from other packages to perform this analysis.

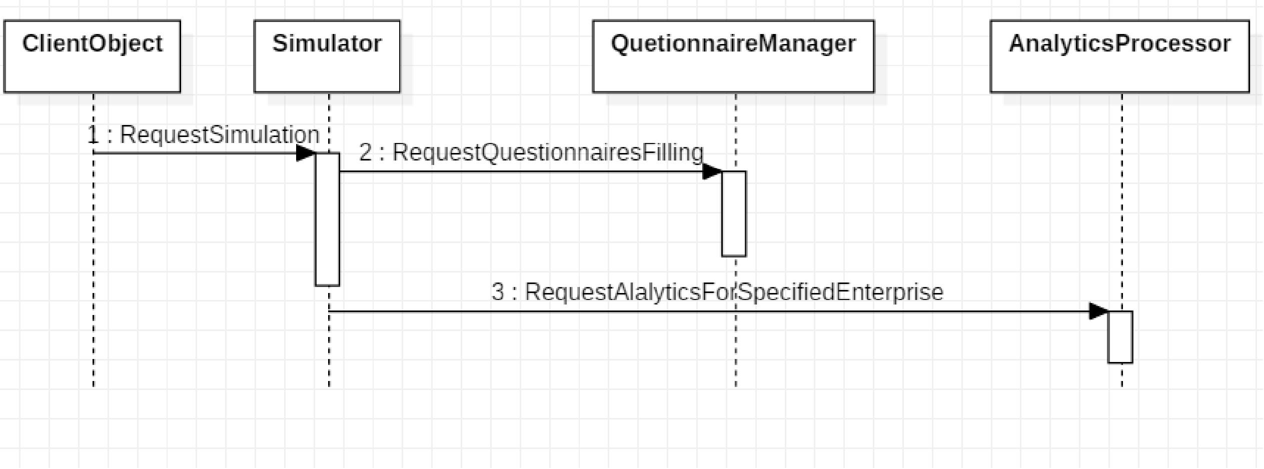


**Figure 56. Sequence diagram of the simulation analysis of Enterprise innovation.**

## Object diagram of ReportGenerator

ReportGenerator is a software implementation of actor Report Generator.
ReportGenerator is responsible for generating reports from the Main Administrator.

ReportGenerator consists of the following classes:

- ReportGenerator - the responsibility of this class is to use a ReportFactory-created object of the appropriate type to generate an appropriate report;
- ReportFactory - returns a type of object specified with a type parameter that inherits the Report base class;
- Report - base class of specific classes for report generation: It is used for polymorphous call of objects from specific classes created by ReportFactory. In this way, a further expansion of system functionality is provided, with the option of creating other types of reports.
- ReportType1, ReportType1- Inheritance Classes Report.

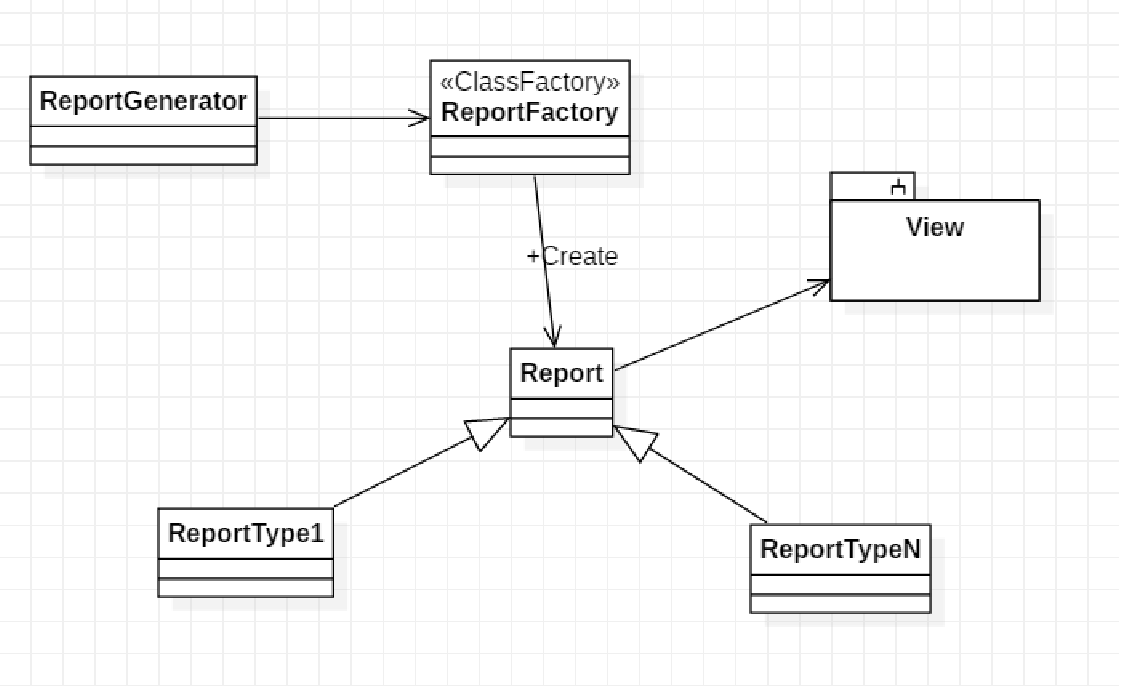


**Figure 57. Object diagram ReportGenerator.**

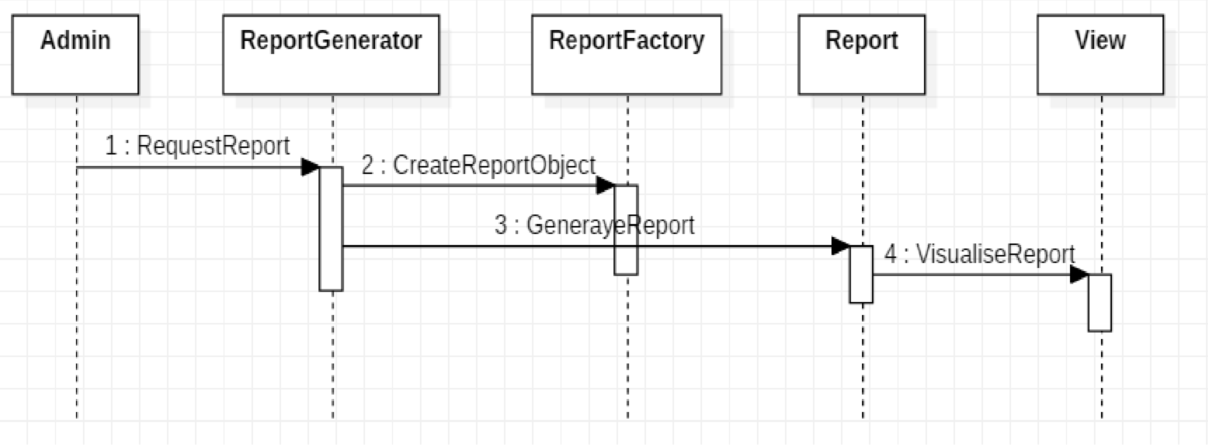


**Figure 58. Sequence diagem of report generation.**

## Object diagram of ContentManager

ContentManager is a software implementation of actor Content Manager. It is responsible for management the content of the system by the Main Administrator;

ContentManager consists of the following classes:

- ContentManager - the responsibility of this class is to manage the maintenance of the visual and content part of the system - the brochure and the portal. To use a ContentProcessor class-created object of the appropriate type to create and edit Content objects that are components of the brochure and portal;
- ContentProcessorFactory - returns a type object specified with a type parameter that inherits the BaseProcessor base class;
- ContentProcessor - a base class of specific classes for creating and editing system components: It is used for polymorphic call to objects from specific classes created by ContentProcessorFactory. In this way, future functionality of the system is enhanced by adding new system components;
- VisualContentProcessor, FileContentProcessor - classes for creating and editing system components;
- Contents is a container containing Content objects. Its responsibility is to maintain a list of components. The class is also responsible for storing the components on external media;
- Content is a class that contains data about a particular component of the system - visual, textual, and so;
- VisualContent, FileContent, TextContent - system component classes.

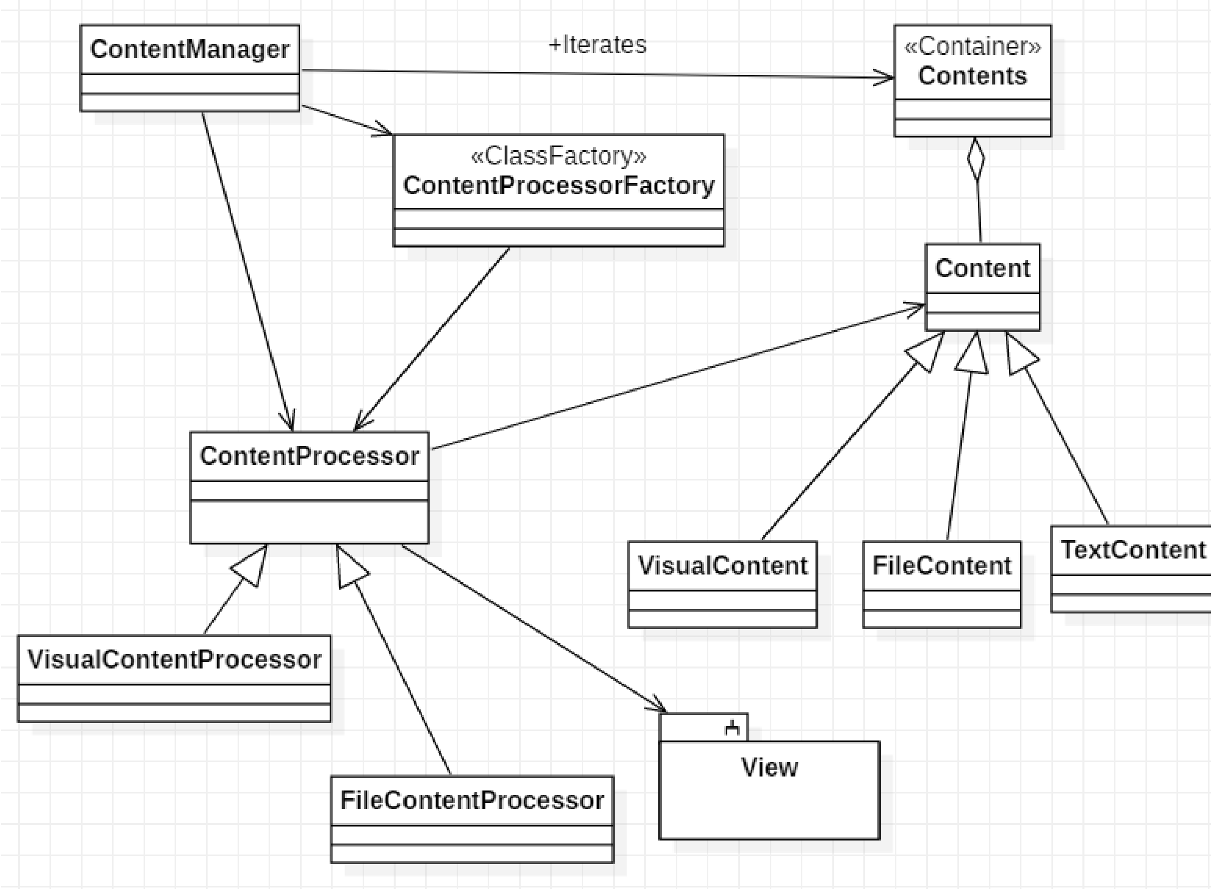


Figure 59. Object diagram of ContentManager.

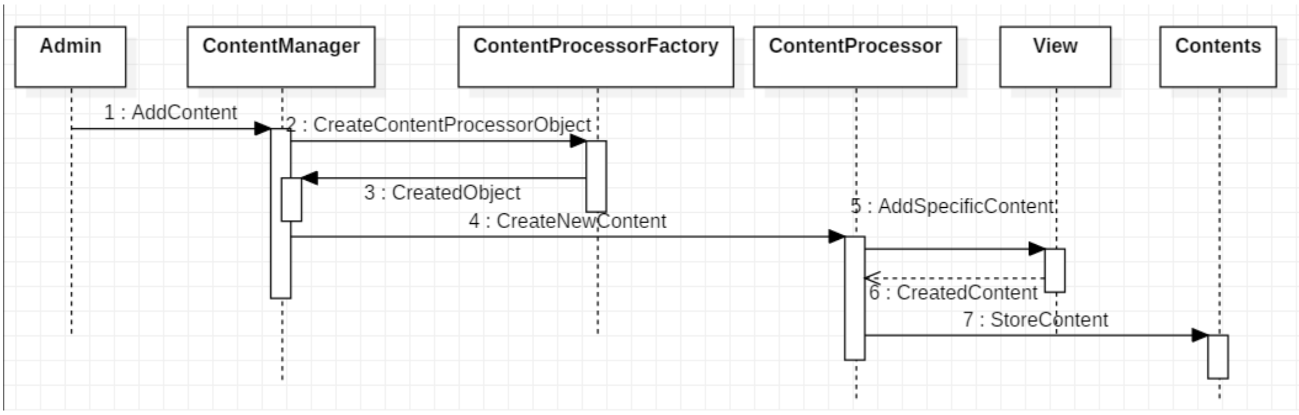


Figure 60. Sequence diagram of creation of a new content.

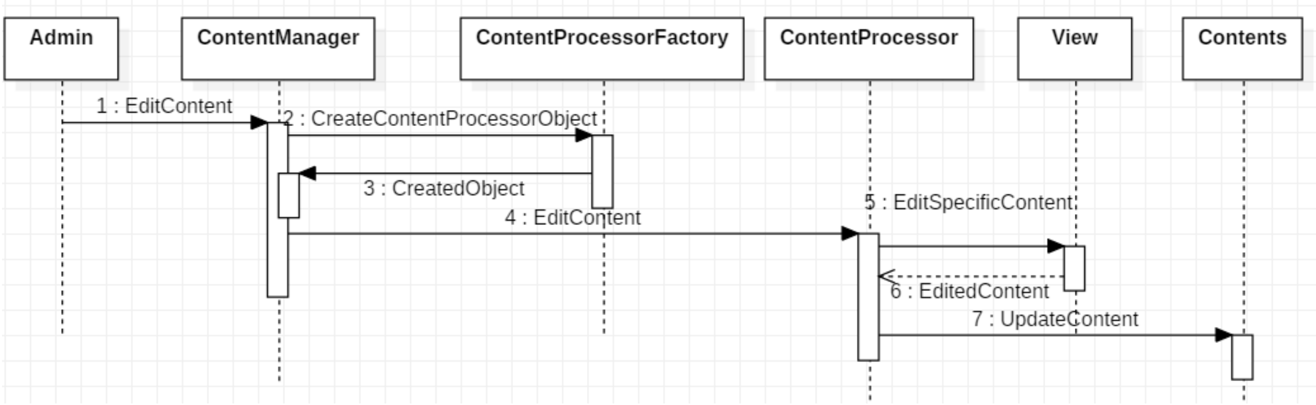


**Figure 61. Sequence diagram of content editing.**