

# Contents

<b>1</b>	<b>General Project Presentation</b>	<b>6</b>
I	Host Company Presentation . . . . .	6
1	Presentation of MASS Analytics . . . . .	6
2	Services . . . . .	7
3	Customers . . . . .	8
II	Project Presentation . . . . .	8
1	Context . . . . .	8
2	Objectives . . . . .	9
3	Problematic . . . . .	9
III	Methodology . . . . .	9
IV	Software Development Life Cycle . . . . .	10
V	Software Development Process . . . . .	11
VI	Existing Presentation . . . . .	11
VII	Conclusion . . . . .	12
<b>2</b>	<b>Analysis and specification of requirements</b>	<b>13</b>
I	Introduction . . . . .	13
II	Actors Identification . . . . .	13
III	Requirement Analysis . . . . .	14
1	Functional Requirements . . . . .	14
2	Non-Functional Requirements . . . . .	15
IV	Use Case Diagrams . . . . .	16
1	Global Use Case Diagram . . . . .	16
2	Detailed Use Case Diagrams . . . . .	17
V	Conclusion . . . . .	27
<b>3</b>	<b>Conception</b>	<b>28</b>
I	Introduction . . . . .	28

II	Modeling Language . . . . .	28
III	Global Conception . . . . .	28
	1 Physical Architecture . . . . .	29
	2 Logical Architecture . . . . .	30
	3 Design Pattern . . . . .	31
IV	Detailed Conception . . . . .	32
	1 Package Diagram . . . . .	33
	2 Class Diagram . . . . .	34
	3 Sequence Diagram . . . . .	35
V	Conclusion . . . . .	57
<b>4</b>	<b>Implementation</b>	<b>58</b>
I	Introduction . . . . .	58
II	Java Platform, Enterprise Edition (JEE) . . . . .	59
	1 Introduction . . . . .	59
	2 Features . . . . .	59
	3 Motivation . . . . .	60
III	Angular JS . . . . .	62
	1 Introduction . . . . .	62
	2 Features . . . . .	63
	3 Motivation . . . . .	64
IV	Bootstrap . . . . .	65
	1 Introduction . . . . .	65
	2 Features . . . . .	65
	3 Motivation . . . . .	66
V	Spring mvc framework . . . . .	68
	1 Introduction . . . . .	68
	2 Features . . . . .	69
	3 Motivation . . . . .	71
VI	Pivotal tc Server . . . . .	72
VII	Spring Tool Suite . . . . .	73
VIII	Apache Maven . . . . .	74
IX	HTML5 . . . . .	75
X	Web Service . . . . .	75
<b>5</b>	<b>Deployment</b>	<b>76</b>
I	Introduction . . . . .	76

# List of Figures

1.1	Mass-Analytics logo . . . . .	6
1.2	MassTer Software logo . . . . .	7
1.3	MASS Analytics' customers . . . . .	8
1.4	Software Development Life Cycle iterative . . . . .	10
1.5	TDD Global Lifecycle . . . . .	11
2.1	Global Use Case Diagram . . . . .	16
2.2	Connect to MassTer Server Use Case Diagram . . . . .	17
2.3	Load Project Use Case Diagram . . . . .	19
2.4	Load Project Use Case Diagram . . . . .	22
2.5	run scenario Use Case Diagram . . . . .	25
3.1	Deployment Diagram Physical Architecture . . . . .	29
3.2	Logical Architecture . . . . .	30
3.3	package diagram . . . . .	33
3.4	Class diagram . . . . .	34
3.5	Sequence Diagram <b>Class diagram</b> . . . . .	36
3.6	Sequence Diagram <b>Display MassTerInsight</b> . . . . .	38
3.7	Sequence Diagram <b>Load MassTerInsight Project Use Case</b> . . . . .	40
3.8	Sequence Diagram <b>Load Report Use Case</b> . . . . .	42
3.9	Sequence Diagram <b>Save Report Use Case</b> . . . . .	44
3.10	Sequence Diagram <b>Remove Report Use Case</b> . . . . .	46
3.11	Sequence Diagram <b>update settings Use Case</b> . . . . .	48
3.12	Sequence Diagram <b>Run Scenario Use Case</b> . . . . .	50
3.13	Sequence Diagram <b>Save Project Use Case</b> . . . . .	52
3.14	Sequence Diagram <b>Save As Project Use Case</b> . . . . .	55
4.1	Java Enterprise Edition . . . . .	59
4.2	java statics . . . . .	61
4.3	java rank . . . . .	61

4.4	AngularJS . . . . .	62
4.5	AngularJS on google trends . . . . .	64
4.6	Bootstrap . . . . .	65
4.7	Bootstrap second starred on github . . . . .	66
4.8	Bootstrap on google trends . . . . .	67
4.9	Spring MVC . . . . .	68
4.10	The request processing workflow in Spring Web MVC . . . . .	69
4.11	Spring framework on google trends . . . . .	71
4.12	Pivotal tc Server . . . . .	72
4.13	spring tool suite . . . . .	73
4.14	Apache Maven . . . . .	74
4.15	HTML5 . . . . .	75

# List of Tables

2.1	Description of the scenario <b>Connect to MassTer Server Use Case.</b>	18
2.2	Description of the scenario <b>Load Project Use Case.</b> . . . . .	20
2.3	Description of the scenario <b>Save Project Use Case.</b> . . . . .	20
2.4	Description of the scenario <b>Save As Project Use Case.</b> . . . . .	21
2.5	Description of the scenario <b>load report Use Case.</b> . . . . .	23
2.6	Description of the scenario <b>Save Report Use Case .</b> . . . .	23
2.7	Description of the scenario <b>Remove Report Use Case.</b> . . . . .	24
2.8	Description of the scenario <b>update settings Use Case.</b> . . . . .	24
2.9	Description of the scenario <b>Run Scenario Use Case.</b> . . . . .	26

# Chapter 1

## General Project Presentation

### I Host Company Presentation

#### 1 Presentation of MASS Analytics

MASS Analytics, a Tunisian start-up founded in 2012, is the first and only independent Marketing Mix Modeling (MMM) agency in the MENA region. MASS Analytics' core competency is the deep analysis and understanding of what impacts the consumer's path to purchase to make companies more effective with their marketing budget.

It was founded by **Dr. Ramla Jarrar** ( Chief Executive Officer ), **Dr. Firas Jabloun** (Chief Technology Officer), **Nadia Bouzguenda** (Business Development Director) & **Rafal Kozlowski** (Director). They brought the essence of more than 20 cumulative years of experience in marketing effectiveness & technology services at the international level to the creation of MASS-Analytics. [1]



Figure 1.1: Mass-Analytics logo

## 2 Services

- **MassTer Software** : MASS Analytics has been developing internally its own Marketing Mix Modeling Software “MassTer”. It is one of the most powerful Marketing Mix Modeling software products/solutions in the world and comes in three packages: standard, professional , and premium. It provides the user with a powerful Modeling platform coupled with a comprehensive data visualization capability to help understand the relationship between different variables and measure their impact on business performance.



Figure 1.2: MassTer Software logo

- **Training and Consultancy** : MASS Analytics runs specific courses and training sessions on advanced predictive modeling (log linear, nested modeling, fixed effect modeling...), budget optimization and return on Investment calculation. It also offers its clients coaching sessions to help improve their marketing analytics process and project delivery.



Figure 1.3: MASS Analytics' customers

source: <https://www.mass-analytics.com/values/>

### 3 Customers

## II Project Presentation

### 1 Context

As part of its software development and consulting activities, oriented towards the business of Marketing Mix Modelling (MMM), Mass-Analytics is always looking to offer its customers products that are the most efficient and smart. Thus, It seeking to provide innovative services that satisfied the different needs of customers based on new technologies.

In this context, this project entitled **MassTer Insight SaaS** was offered to me by Mass-Analytics as part of a graduation project to obtain the national diploma of computer engineer.



## 2 Objectives

The main objective of this project is to develop MassTer Insight from the exiting Desktop solution to a SaaS.

We are in charge to keep the same business logic of the desktop solution, but offered to the customers in cloud SaaS version with Security guaranteed, high perform in term of Speed, Visualization, Graphics, Design .

## 3 Problematic

Today all the companies move to cloud, they are tend to by their softwares through cloud SaaS, for more reasons : one amongst the reason is security, when you offer an executable software, this one is threaten by the crack, also an executable requires sometimes to take care of your resources needs, an important RAM, CPU, and more. Almost the executable save data locally which is a very bad way to stock data, it is possible to loose these data once the hard disk is defect by an external effect or even internal. we don't need to inform the customer for each updates to keep installing our software, once he/she connect on the web application he/she got the last version without any process of installation.

That's what we care about, our current solution Masster Insight desktop is threaten by the crack, may will be heavy on machine with smaller resources, works only on pc. it requires the process of installation wasted time.

## III Methodology

The choice of the methodology is an important step in software development since it grants formalizing the preliminary steps when establishing a system in pursuance of the client's requirements. The scrum , an approach that is part of the Agile movement, was used when carrying out this project.

## IV Software Development Life Cycle

We chosen the **Iterative Model** as Software Development Life cycle (SDLC).

This model turns the process of development in cyclic manner repeating very step after every cycle of SDLC process.

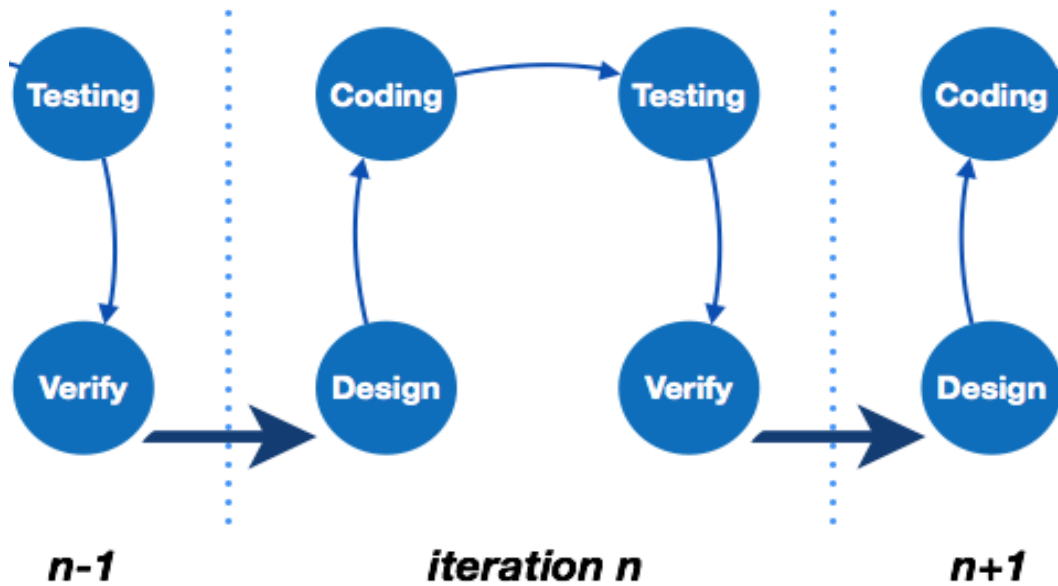


Figure 1.4: Software Development Life Cycle iterative  
**source:** <https://www.tutorialspoint.com/>

## V Software Development Process

Test-driven development (TDD) is a techniques of implementation code, starts with developing test for each features.

The developer writes a little test code and then implements the business, in the traditional method, the test comes from the code, but in TDD the business code is submissive to tests then it is corrected until the tests are validated.

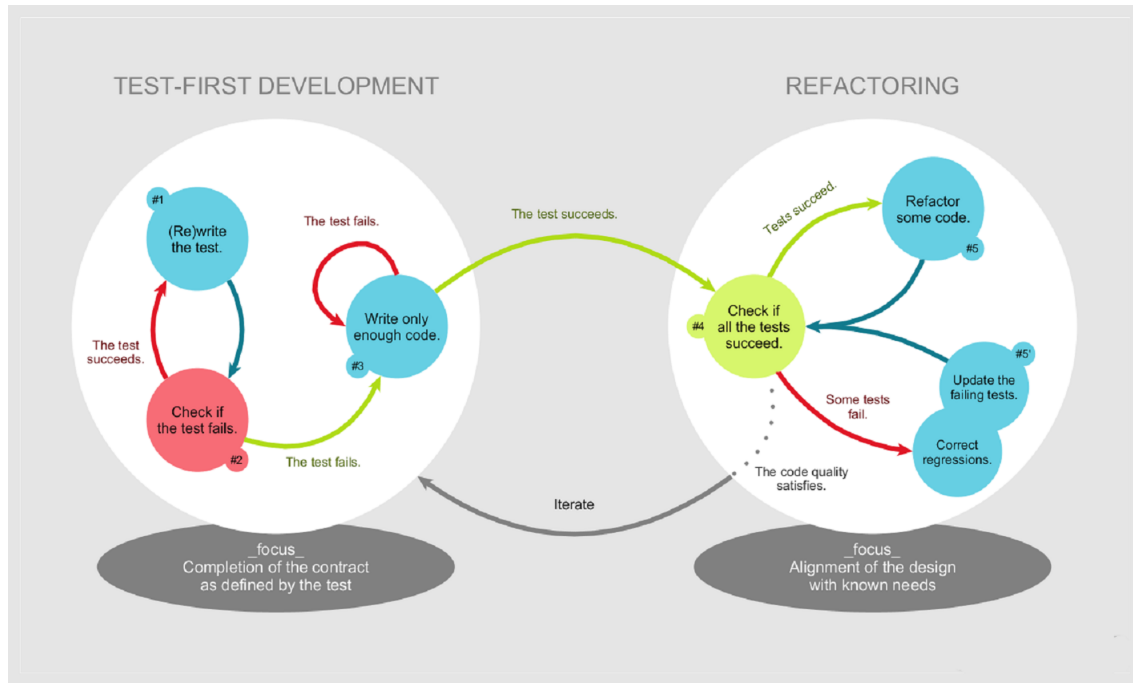


Figure 1.5: TDD Global Lifecycle

source: <https://commons.wikimedia.org/wiki/>

## VI Existing Presentation

**MassTer Insight Desktop Application** is an easy to use software that allows you to run simulation scenarios and allocate your budget optimally across Regions, Products, Channels and Periods. It tells you how much budget to spend on every

single media channel and in which period, given a complex modeling structure. It will help you to benefit from your Marketing Mix Modeling projects.

## **VII Conclusion**

This chapter was a presentation of the hosting company, its services, and clients. The problematic of the project was also highlighted, along with the proposed solution and the methodology followed while carrying out the project.

# Chapter 2

## Analysis and specification of requirements

### I Introduction

Being the first in the development cycle of the project, this phase is the most important. Indeed, it is during this period that the needs of the user are identified and specified. These requirements also represent the functionalities that should be present in the application, which also makes it possible to validate the application as the development progresses.

### II Actors Identification

**MassTer Insight web Application** was mainly designed to be used by **Data Analysts** in MMM agencies, which is the case of **MASS Analytics**, **Media Agencies** that have a MMM division, and **Advertisers** who have an in-house MMM team.

### III Requirement Analysis

#### 1 Functional Requirements

These functional requirements express the expectations of different users for the product to be produced.

In this part, we present the different functionalities and services that the application must ensure.

CONNECT TO SERVER

- **Connect To MassTer Server :**

LOAD PROJECT

- **Load MassTer Insight Project :**

SAVE PROJECT

- **Save MassTer Insight Project :**

SAVE AS PROJECT

- **Save as MassTer Insight Project :**

MANAGE REPORT

- **Load a Report :**
- **Save a Report :**
- **Remove a Report :**

UPDATE SETTINGS

- **Update Settings Report :**

RUN

- **Run new scenario :**

## 2 Non-Functional Requirements

- **Ergonomics** : The application offers a user-friendly and easy-to-use interface without refer to particular knowledge.
- **Security** : ????
- **Scalability** : ????
- **Modularity** : a code that is easy to maintain and simple to understand in order to ensure the scalability of application.

## IV Use Case Diagrams

### 1 Global Use Case Diagram

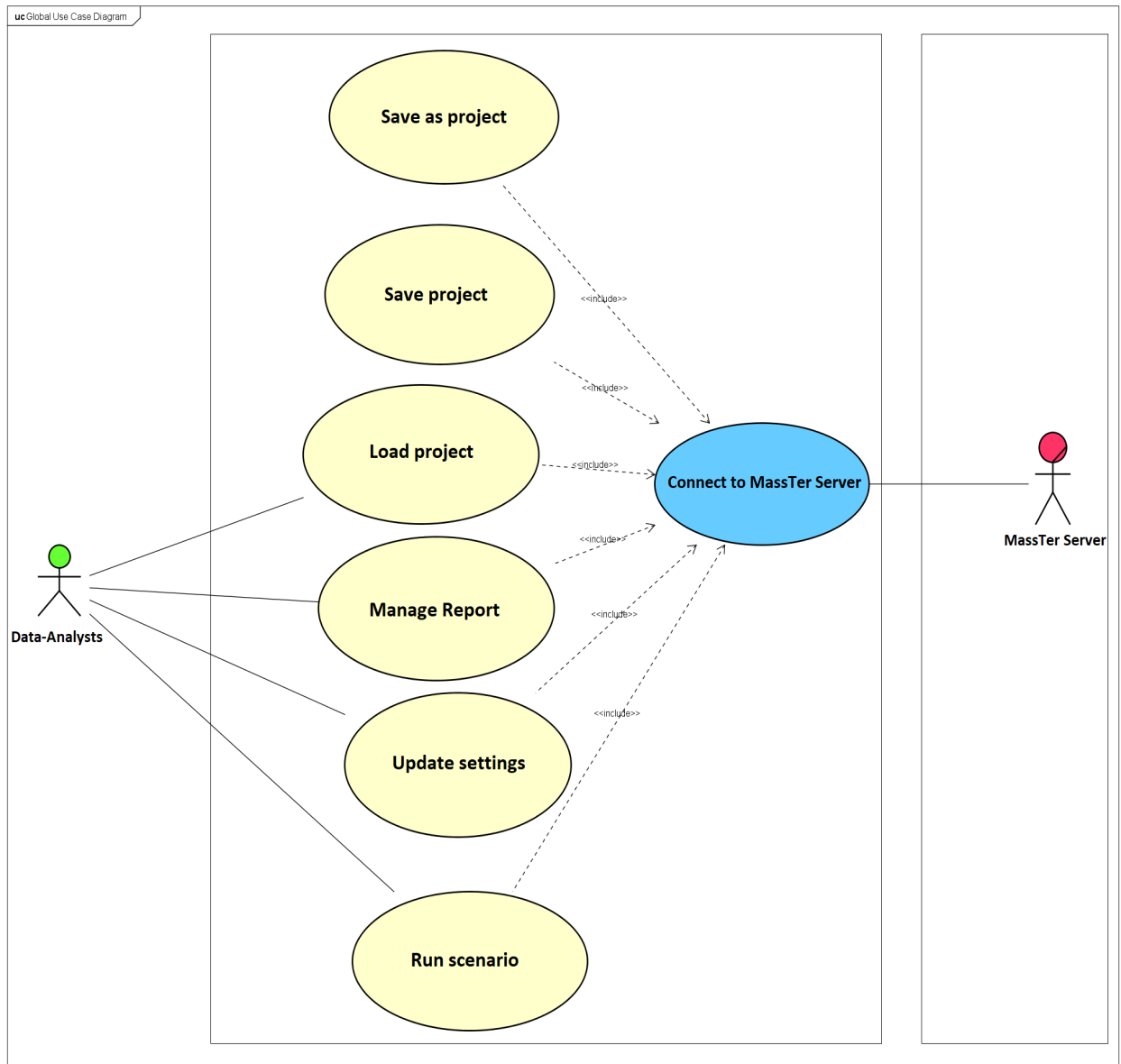


Figure 2.1: Global Use Case Diagram



## 2 Detailed Use Case Diagrams

### Connect to MassTer Server Use Case

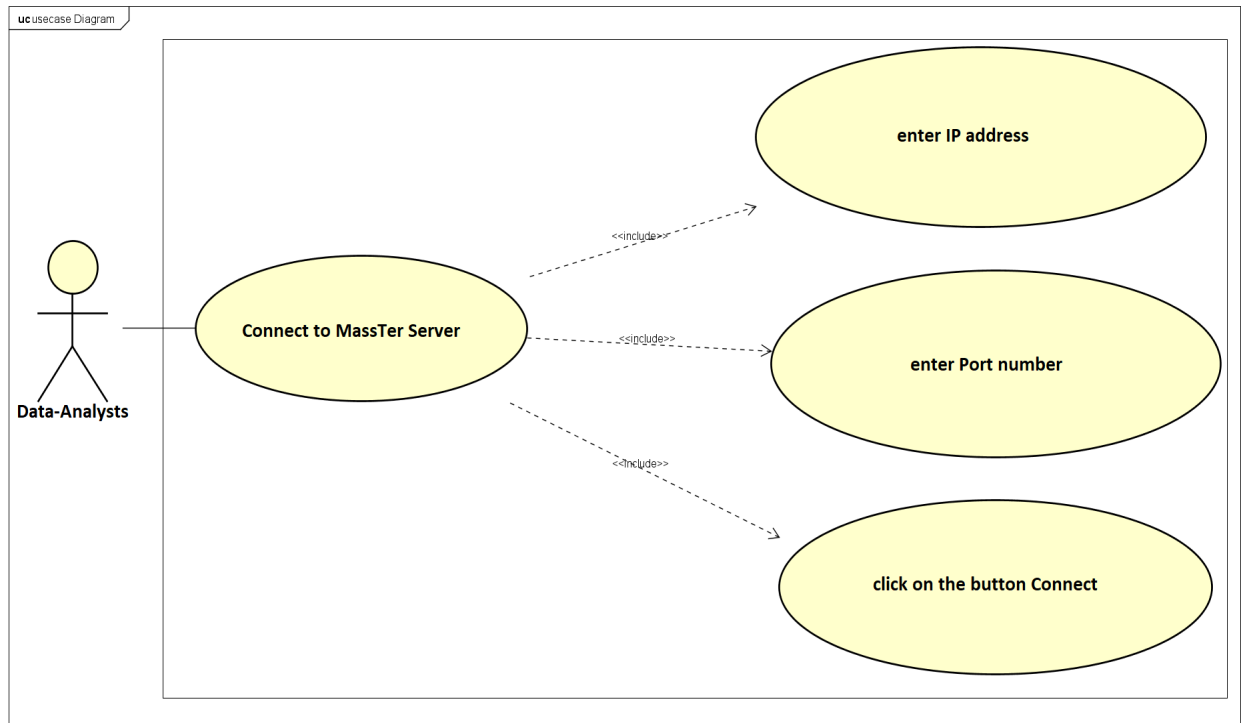


Figure 2.2: Connect to MassTer Server Use Case Diagram

Table 2.1: Description of the scenario **Connect to MassTer Server Use Case**.

<b>Pre-Conditions</b>	MassTer Server is running
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user click on the item MassTer Server in Main Menu.</li> <li>2. The Application display a drop down.</li> <li>3. The user type MassTer Server IP address.</li> <li>4. The user type MassTer Server port number.</li> <li>5. The user click on button connect.</li> </ol>
<b>Post-Conditions</b>	Connect to MassTer Server is successfully.

## Load Project Use Case

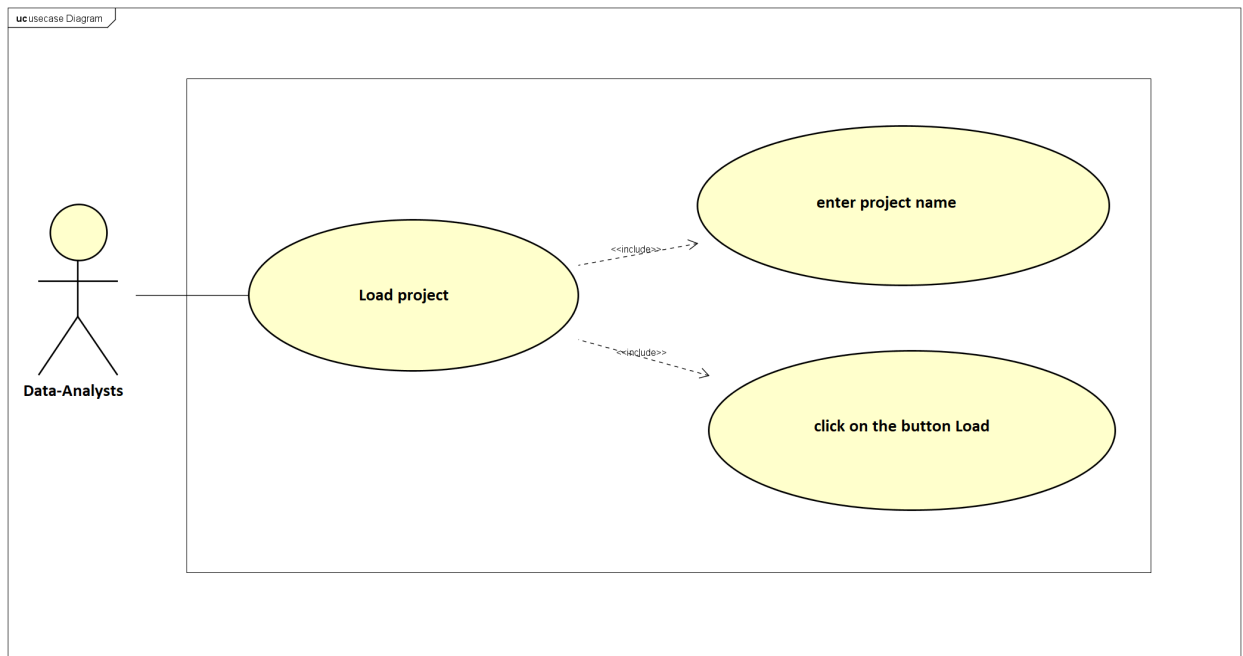


Figure 2.3: Load Project Use Case Diagram

Table 2.2: Description of the scenario **Load Project Use Case**.

<b>Pre-conditions</b>	MassTer Server is running
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user click on the item Load project in Main Menu.</li> <li>2. The user type MassTer Insight project name.</li> <li>3. The user click on the button load.</li> </ol>
<b>Post-conditions</b>	The Application display optimization page.

Table 2.3: Description of the scenario **Save Project Use Case**.

<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. MassTer Server is running.</li> <li>2. Load Project is done successfully.</li> </ol>
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user click on the item save project in Main Menu.</li> <li>2. The user select save item in the drop down.</li> </ol>
<b>Post-conditions</b>	the project is saved successfully.

Table 2.4: Description of the scenario **Save As Project Use Case**.

<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>1. MassTer Server is running.</li><li>2. Load project is done successfully.</li></ol>
<b>Nominal Scenario</b>	<ol style="list-style-type: none"><li>1. The user click on the item save as in Main Menu.</li><li>2. The user select save as item in the drop down.</li></ol>
<b>Post-conditions</b>	The project is saved with a new name successfully.

## Manage Report Use Case

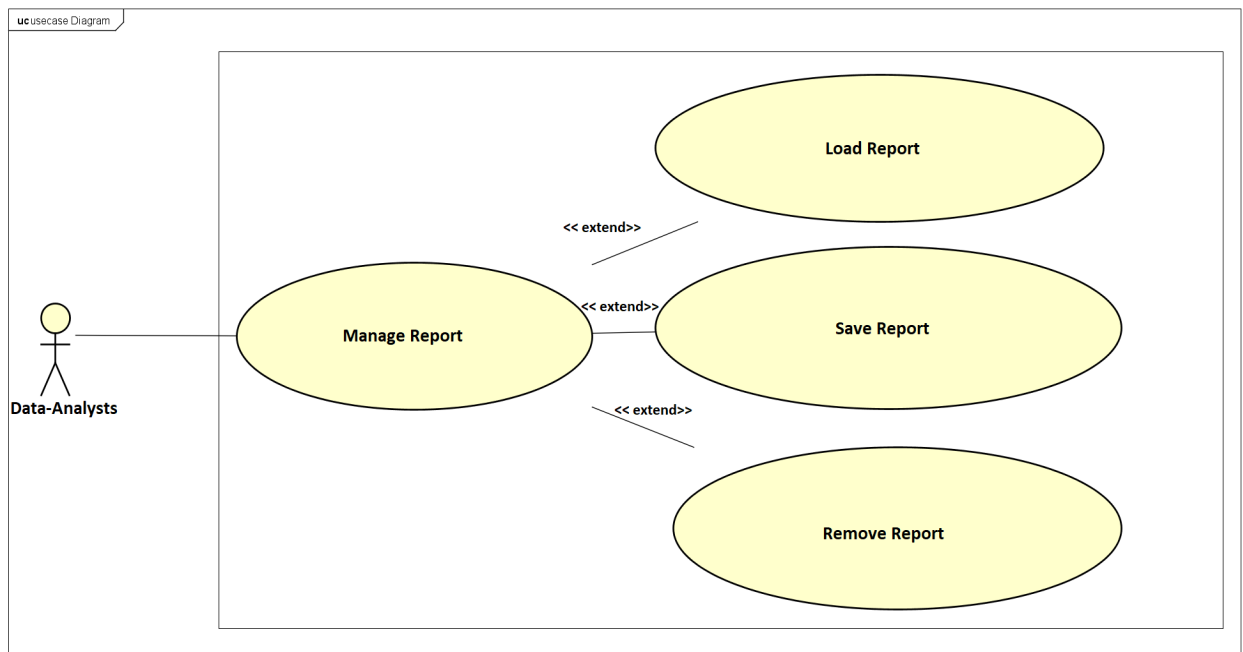


Figure 2.4: Load Project Use Case Diagram

Table 2.5: Description of the scenario **load report Use Case**.

<b>Pre-Conditions</b>	MassTer Server is running
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user select a report from the drop down.</li> <li>2. The user click on the button load.</li> </ol>
<b>Post-Conditions</b>	The application display a new optimisation report related to the selected report

Table 2.6: Description of the scenario **Save Report Use Case** .

<b>Pre-Conditions</b>	MassTer Server is running
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user click on the button save.</li> <li>2. The user type a name for the new report in the pop up.</li> <li>3. The user click on the button save in the pop up.</li> </ol>
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>• The drop down refresh the list of reports.</li> <li>• Add the new report to the list of reports.</li> </ul>

Table 2.7: Description of the scenario **Remove Report Use Case**.

<b>Pre-Conditions</b>	MassTer Server is running
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user select a report from the drop down.</li> <li>2. The user click on the button remove.</li> </ol>
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>• The selected report was deleted.</li> <li>• The drop down refresh the list of reports.</li> <li>• Removes the deleted report from the list.</li> </ul>

Table 2.8: Description of the scenario **update settings Use Case**.

<b>Pre-conditions</b>	The loaded MassTer Insight project contains at least one report saved
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user select new channels .</li> <li>2. The user click on the button update.</li> </ol>
<b>Post-conditions</b>	The Application display a pop up notification to inform the user that the update was done successfully.



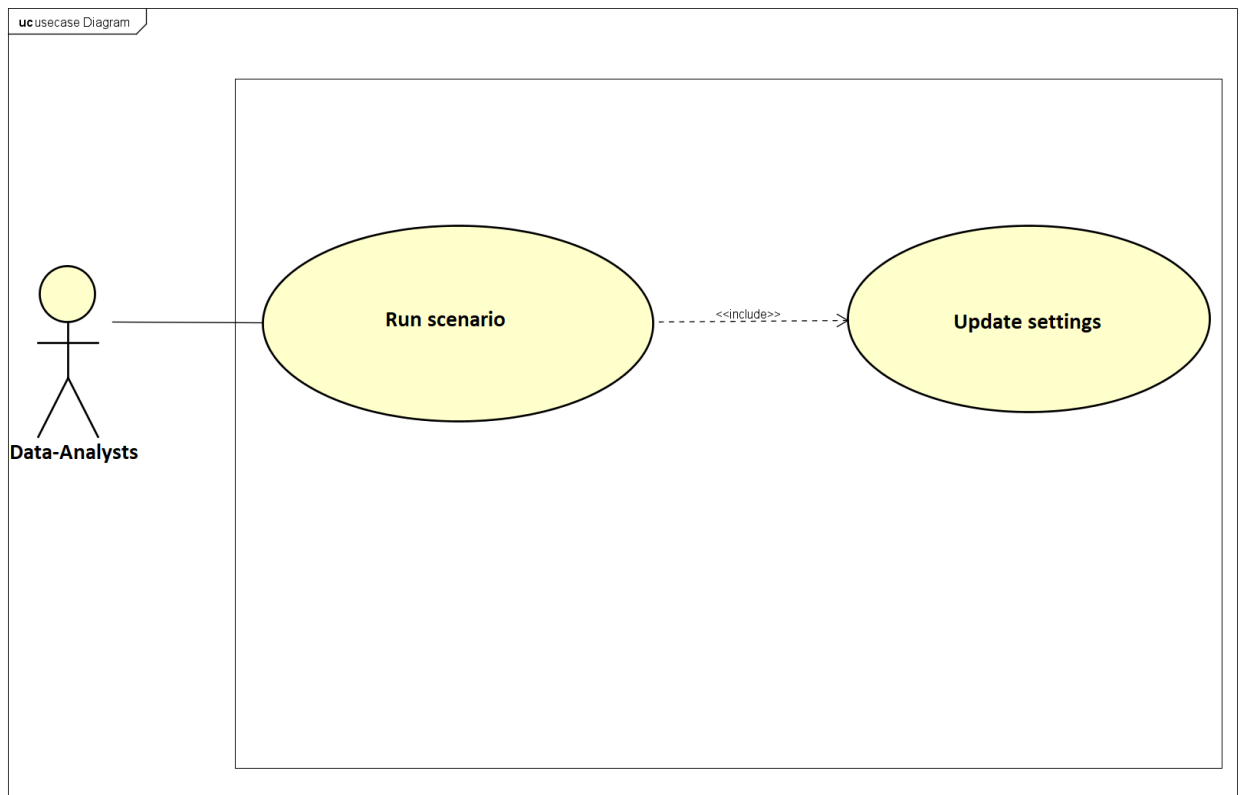


Figure 2.5: run scenario Use Case Diagram

Table 2.9: Description of the scenario **Run Scenario Use Case**.

<b>Pre-conditions</b>	The update was done with success
<b>Nominal Scenario</b>	<ol style="list-style-type: none"> <li>1. The user check max Budget or min Target.</li> <li>2. The user type new value for radio checked.</li> <li>3. The user type the budget Range.</li> <li>4. The user type the resolution.</li> <li>5. the user modifier the constraints table.</li> <li>6. The user click on the button run.</li> </ol>
<b>Post-conditions</b>	The Application display a new optimisation report with new Optimisation Results.

## V Conclusion

This chapter has allowed us to identify the actors that may interact with the developed system, to define the functional and non-functional requirements of the project and modeling the use case diagrams.

In what follows, we present the general and detailed conception phase of the System.

# Chapter 3

## Conception

### I Introduction

### II Modeling Language

A modeling language is used to describe a system, a standard or methodology, general or domain-specific and / or context based on its components and relationships. There are several modeling languages, the best known are UML and Merise. In our project we chose UML as Modeling language.

UML is a Unified Modeling language that can model a problem in a standard way.

#### Why UML ?

We chose UML for these reasons :

- To obtain a very high level modeling independent of the language and environments
- Document a project.

### III Global Conception

In this section, we highlight the architecture of our application, we starting with physical architecture and the logical architecture.

# 1 Physical Architecture

It is primordial to designing any computer system to choose the model architecture that will be adequate to ensure proper functioning, performance, the reuse and reliable interconnection of this system with others. We opt for this purpose for the physical architecture described in the figure below.

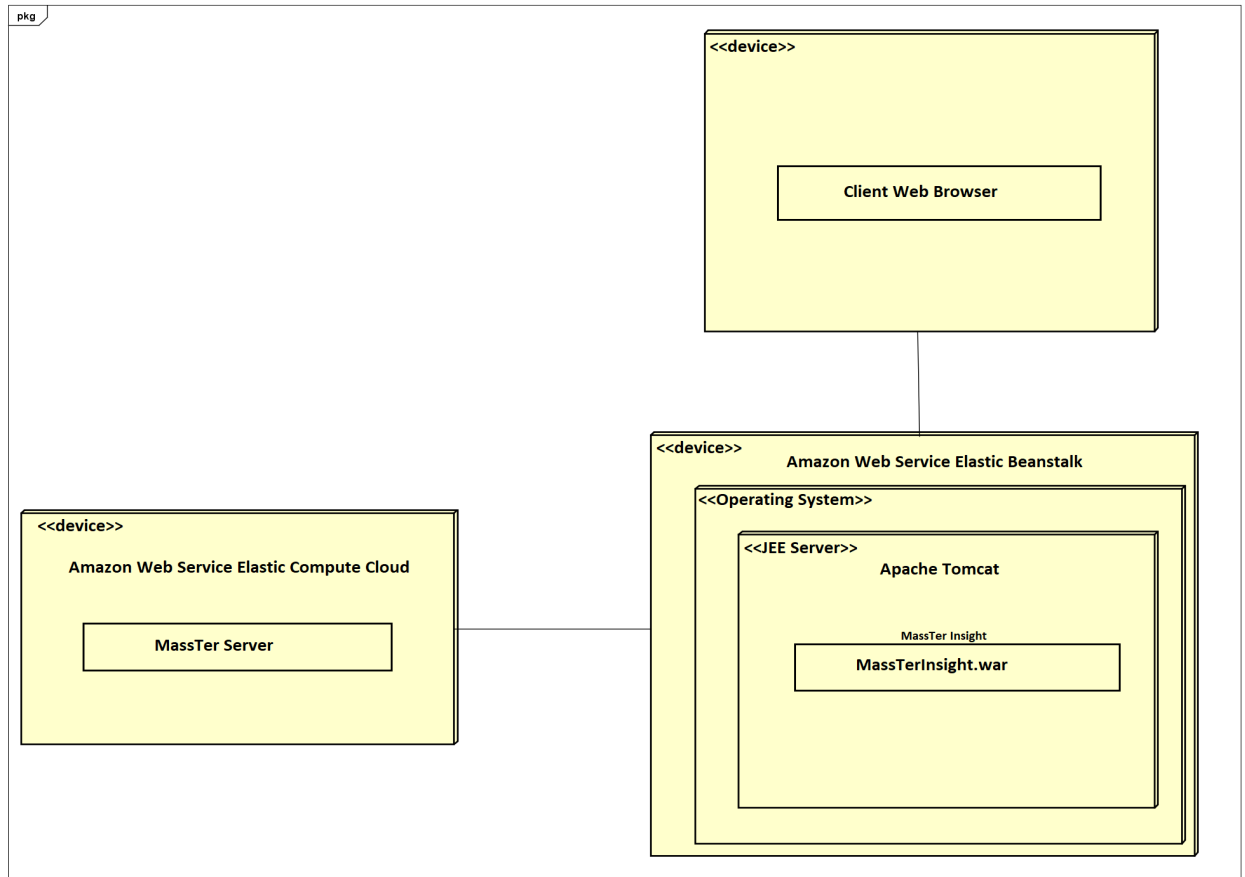


Figure 3.1: Deployment Diagram Physical Architecture

## 2 Logical Architecture

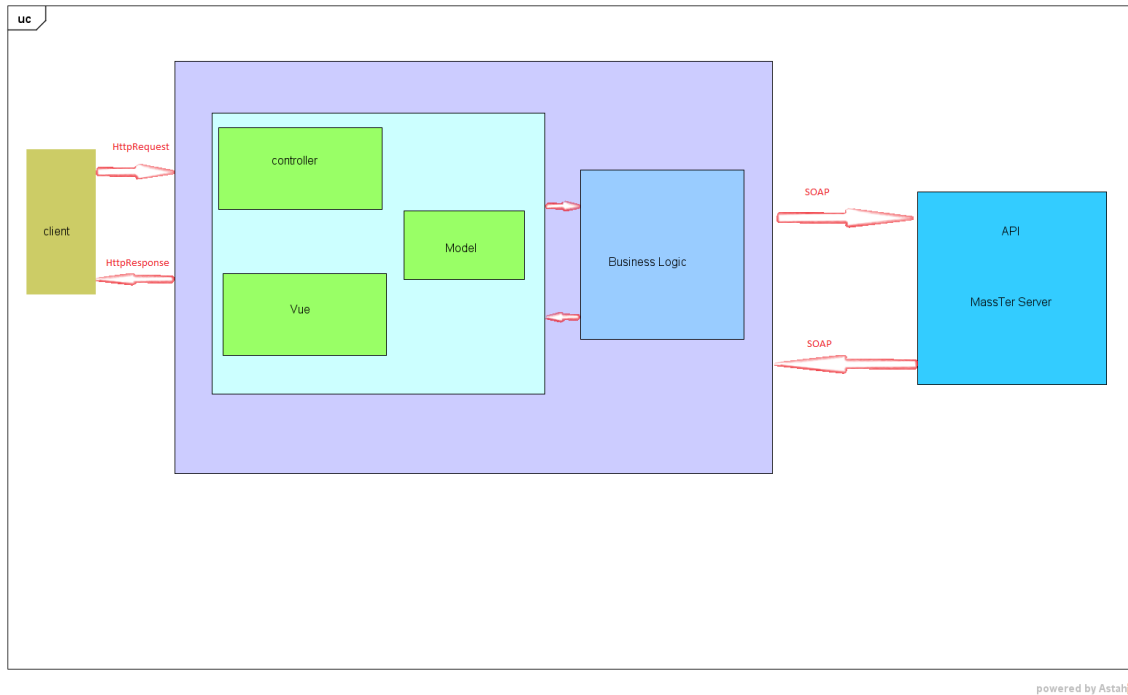


Figure 3.2: Logical Architecture

### 3 Design Pattern

We opt to use the MVC design pattern for the benefits it offers :

- **Reliability** : The presentation and business layers are completely separate, so that the business can change without necessarily affecting the presentation, or vice versa.
- **Adaptability** : Any visual representation can be easily integrated.
- **Productivity** : The duration of development is significantly reduced, in allowing parallel work teams.
- **Extensible** : With MVC the code is extensible.

## IV Detailed Conception



## 1 Package Diagram

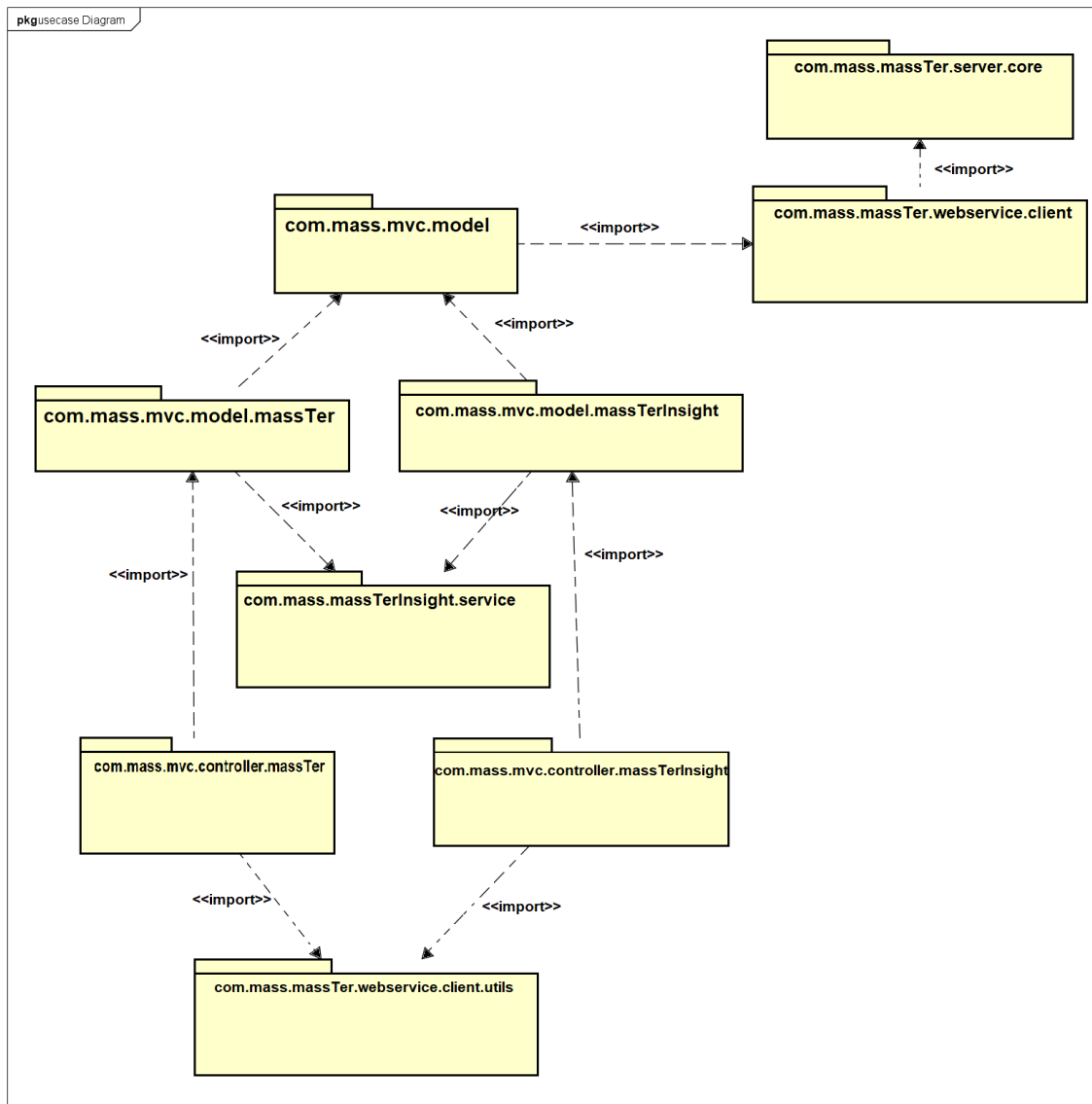


Figure 3.3: package diagram

## 2 Class Diagram

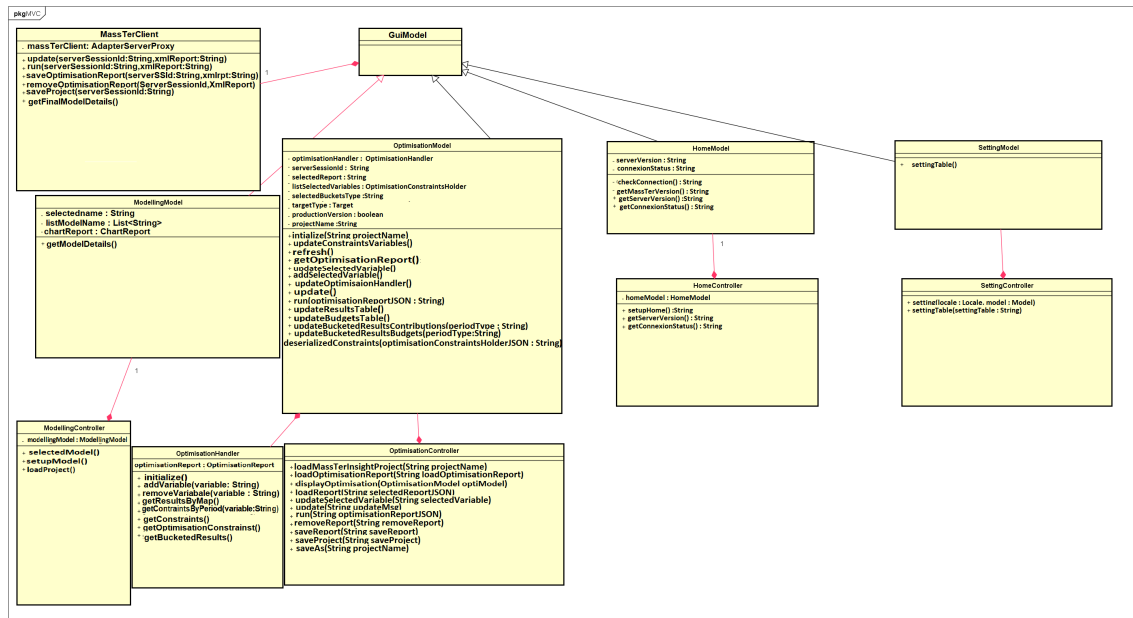
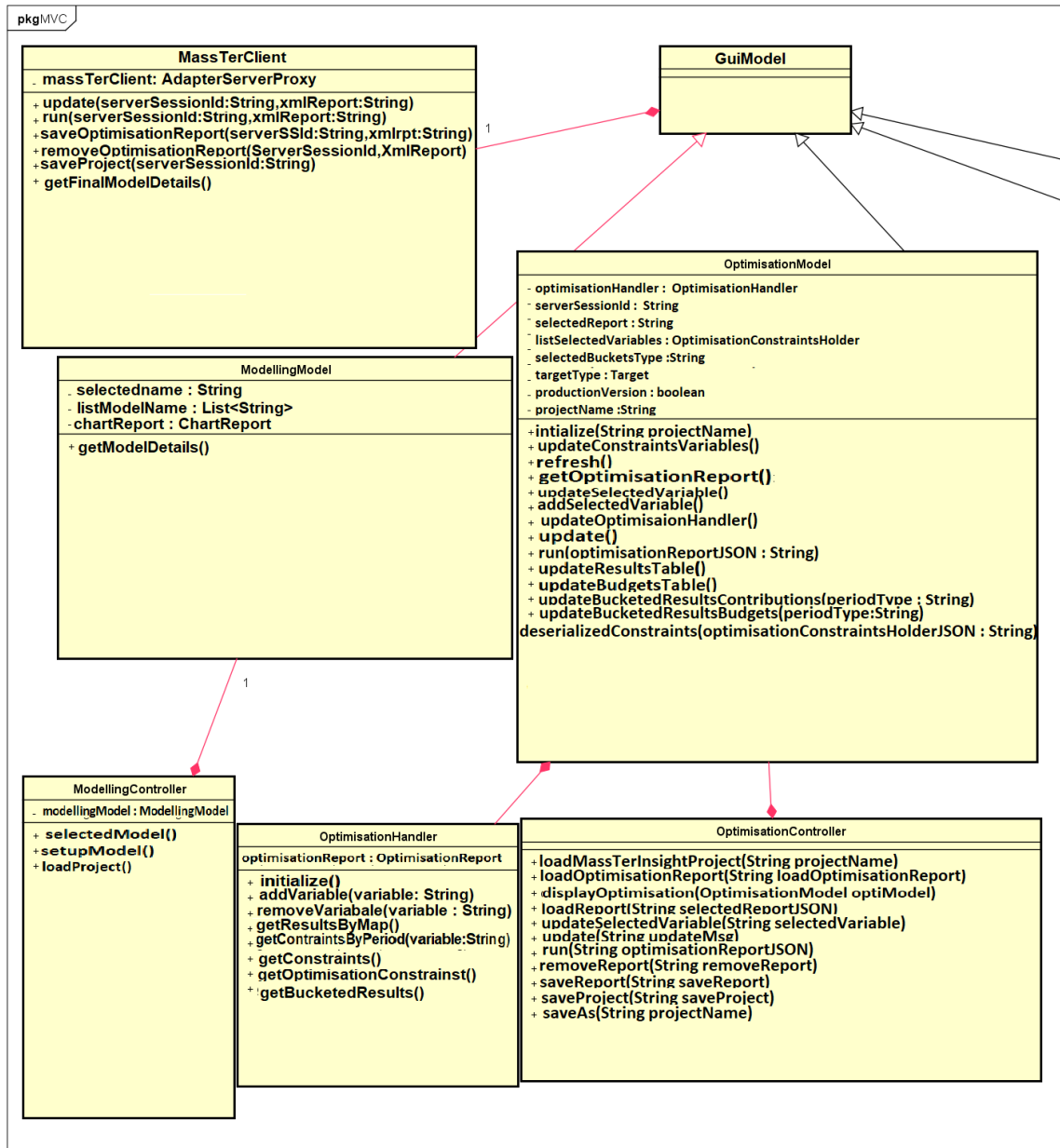


Figure 3.4: Class diagram

### 3 Sequence Diagram

Figure 3.5: Sequence Diagram Class diagram



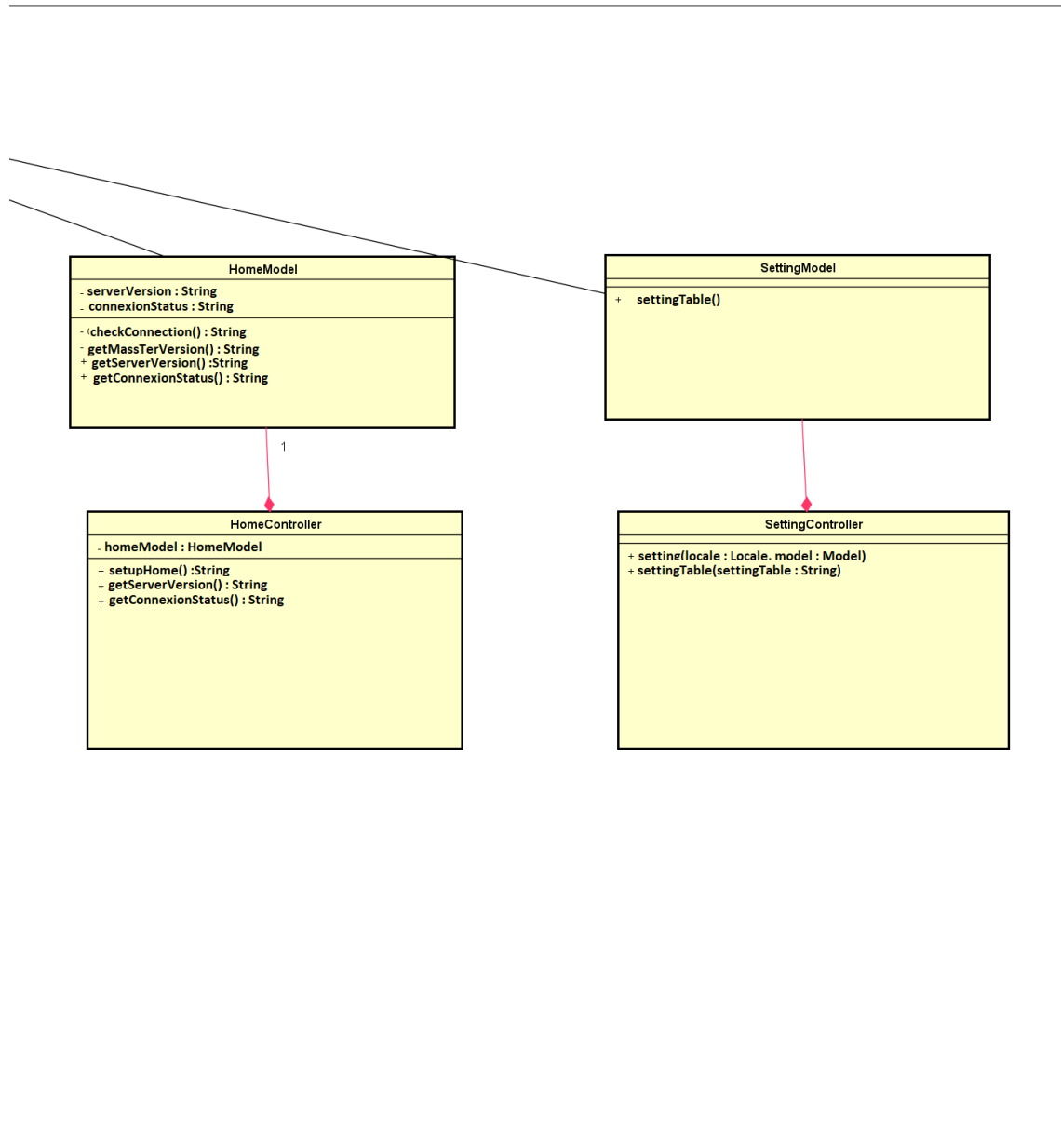
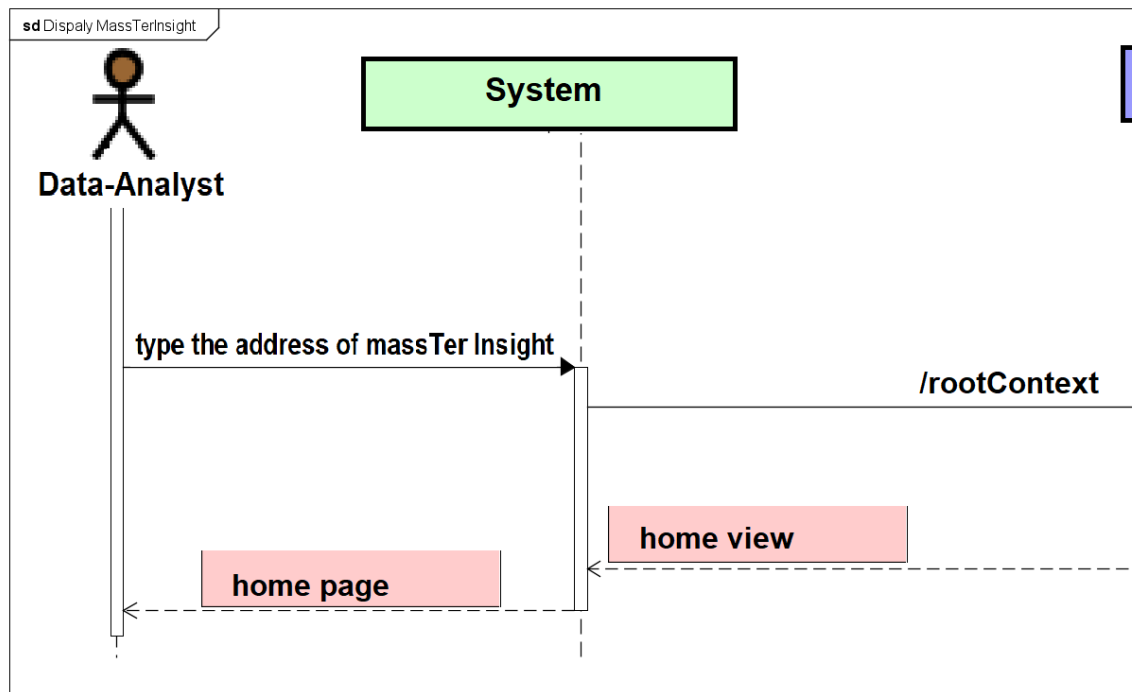


Figure 3.6: Sequence Diagram **Display MassTerInsight**



Sequence Diagram use case “Load report”

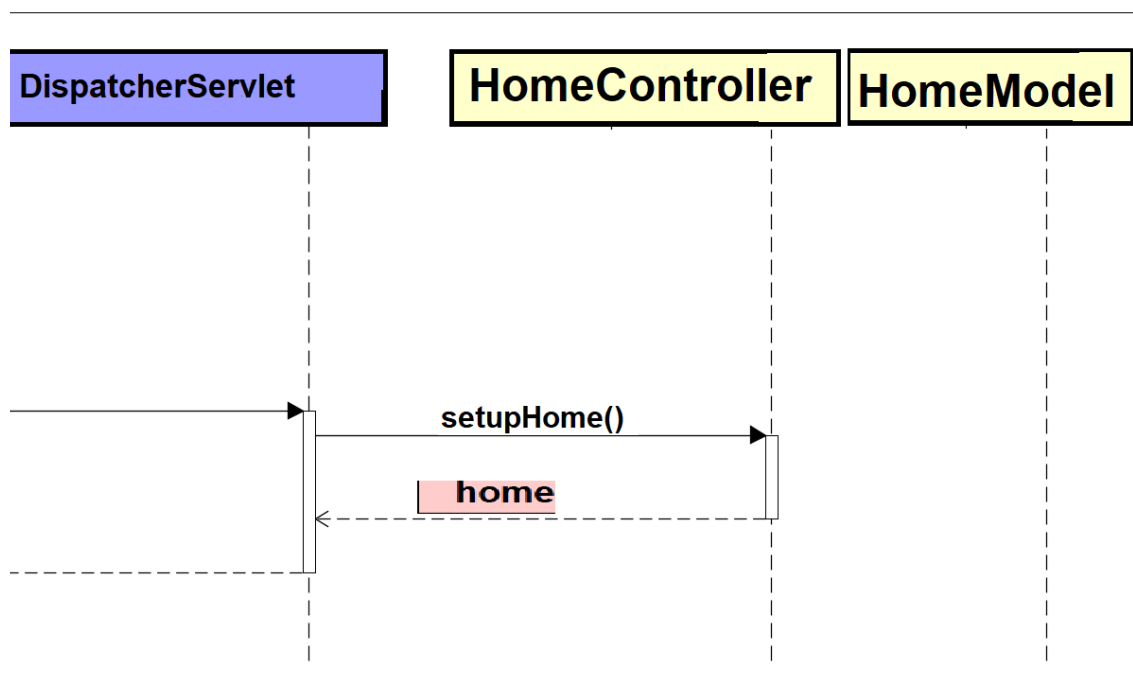


Figure 3.7: Sequence Diagram Load MassTerInsight Project Use Case

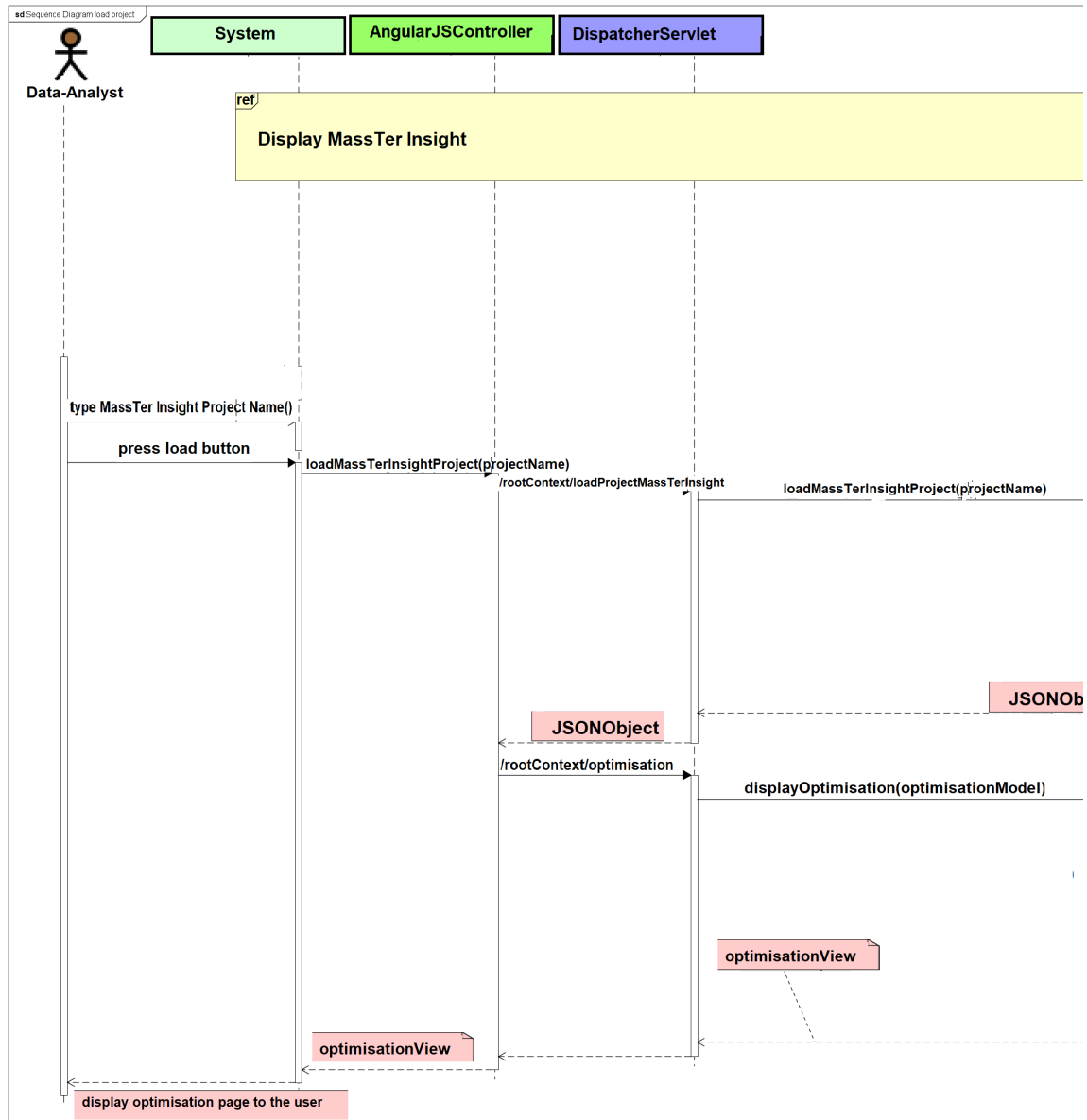
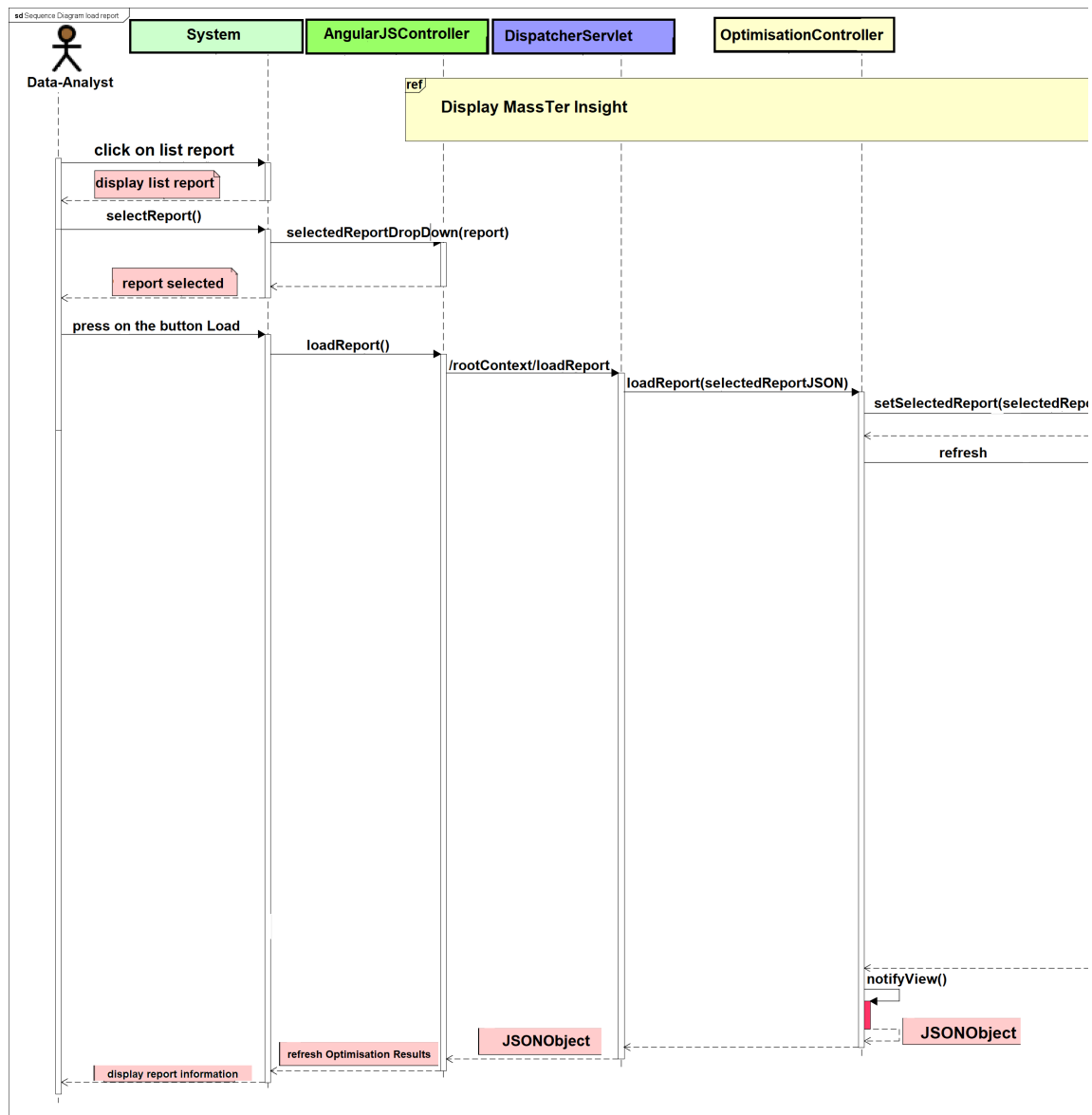






Figure 3.8: Sequence Diagram Load Report Use Case



Sequence Diagram use case “save report”

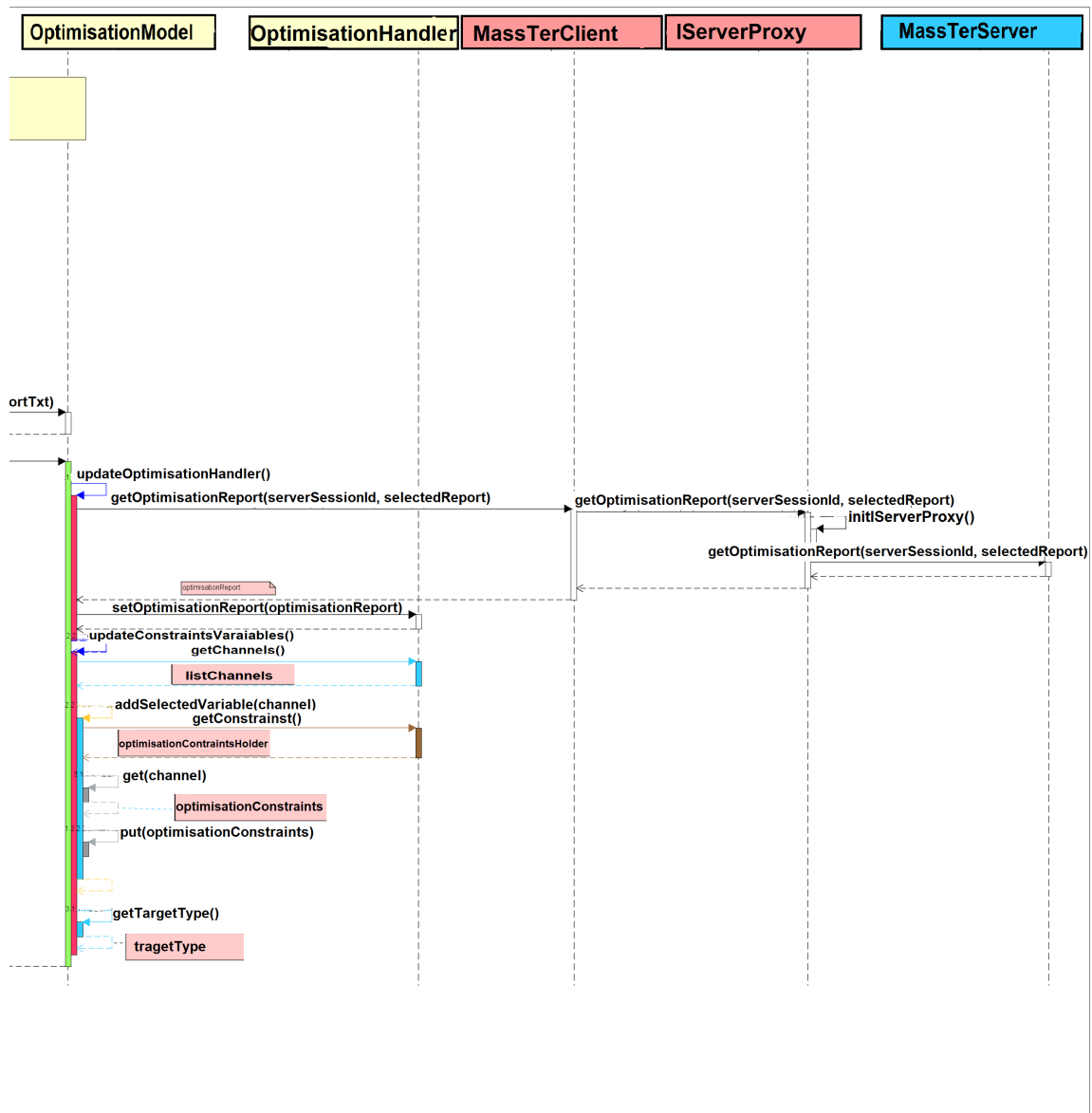
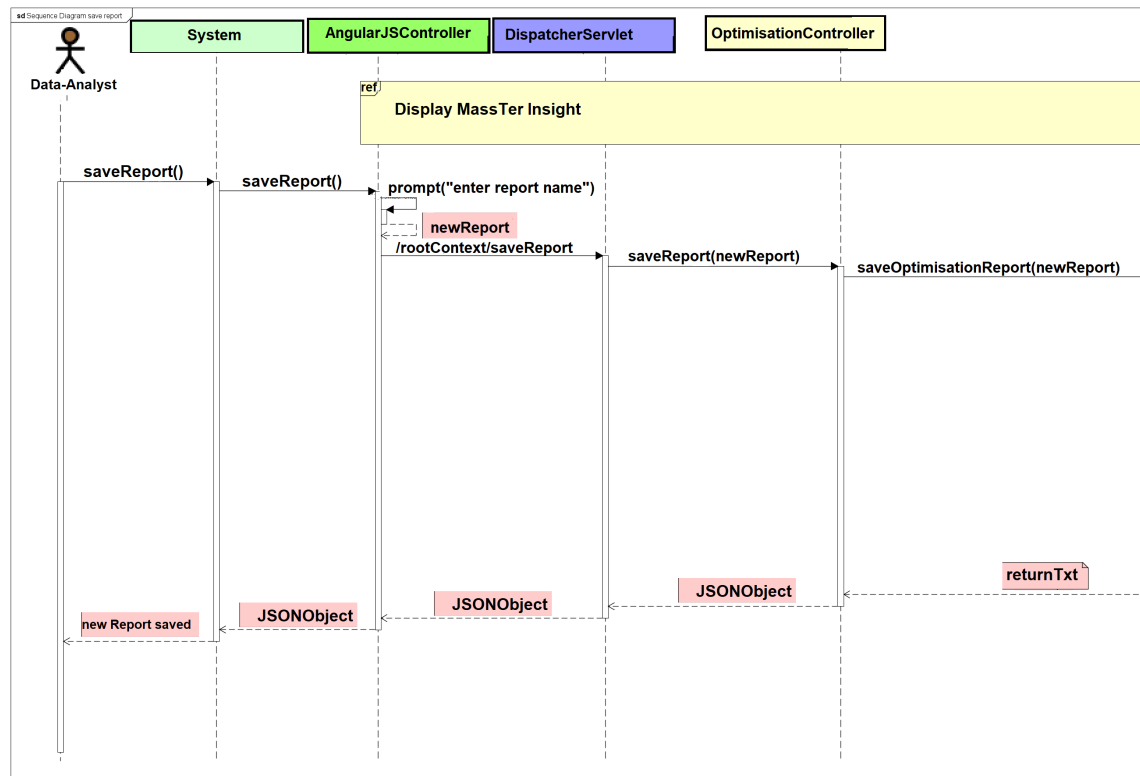
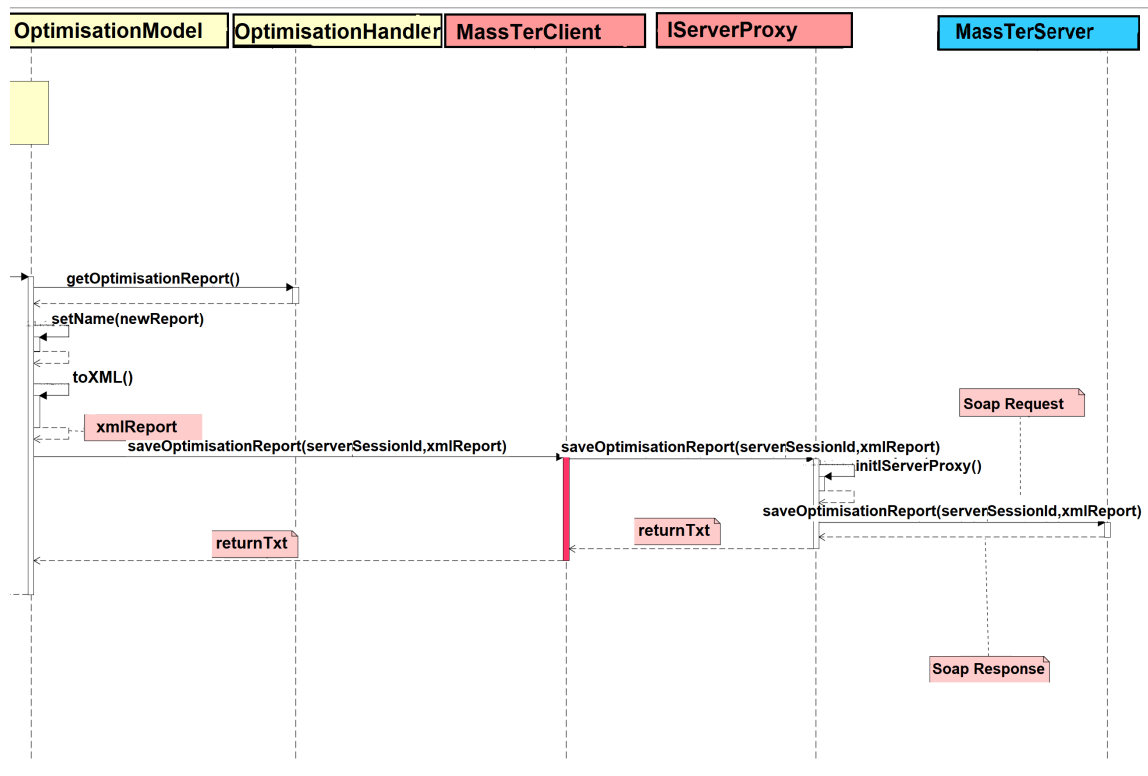


Figure 3.9: Sequence Diagram **Save Report Use Case**

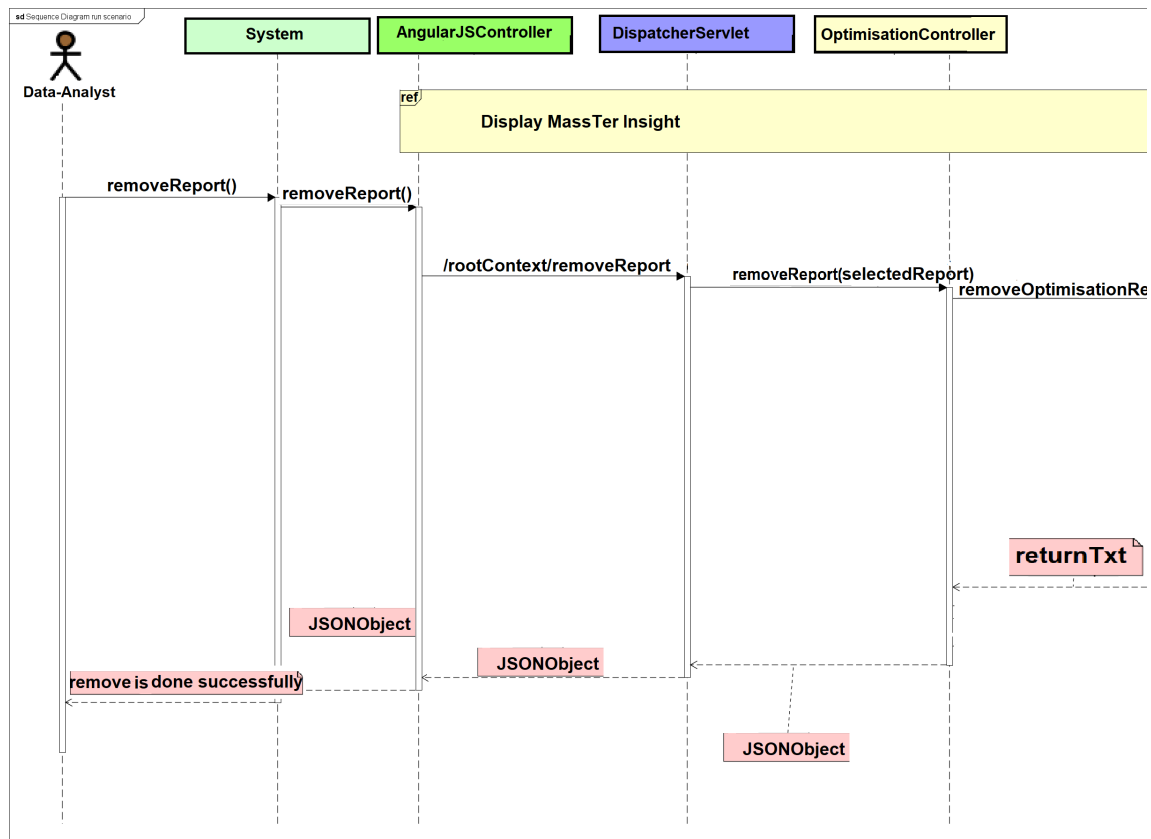


Sequence Diagram use case “remove report”



powered by Acta

Figure 3.10: Sequence Diagram **Remove Report Use Case**



Sequence Diagram use case “update settings”

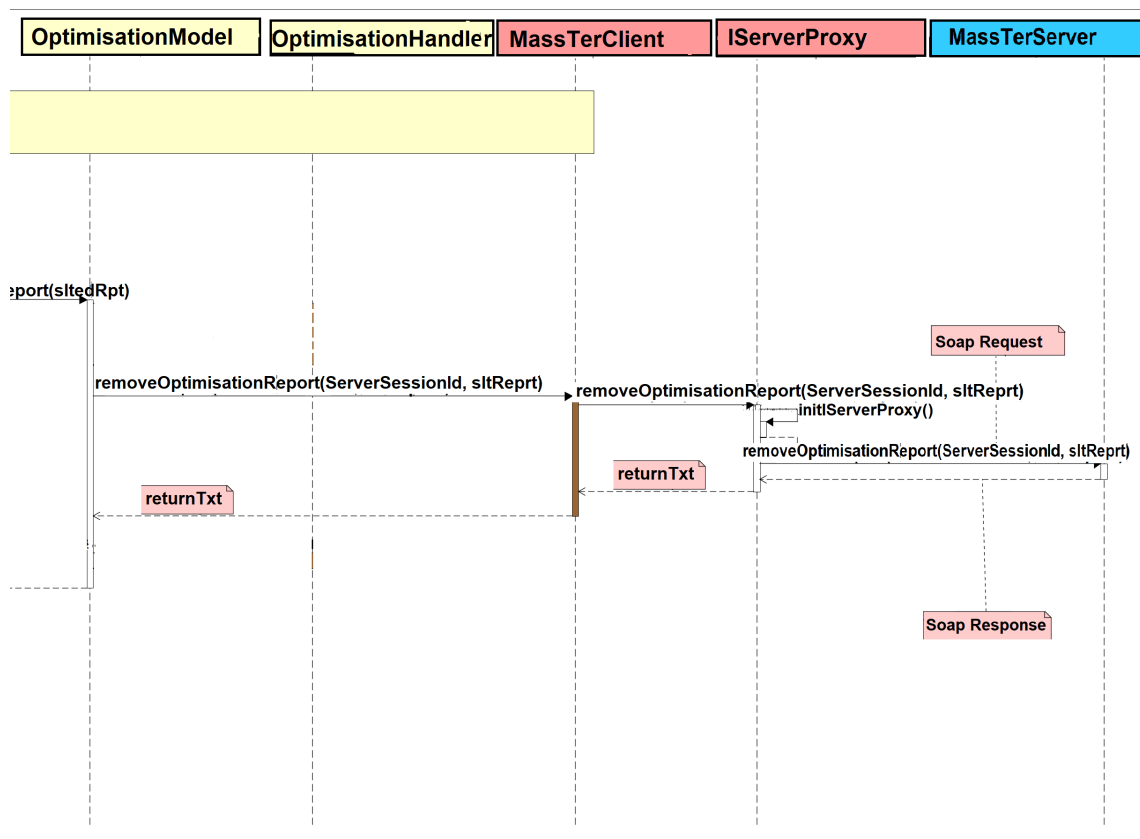
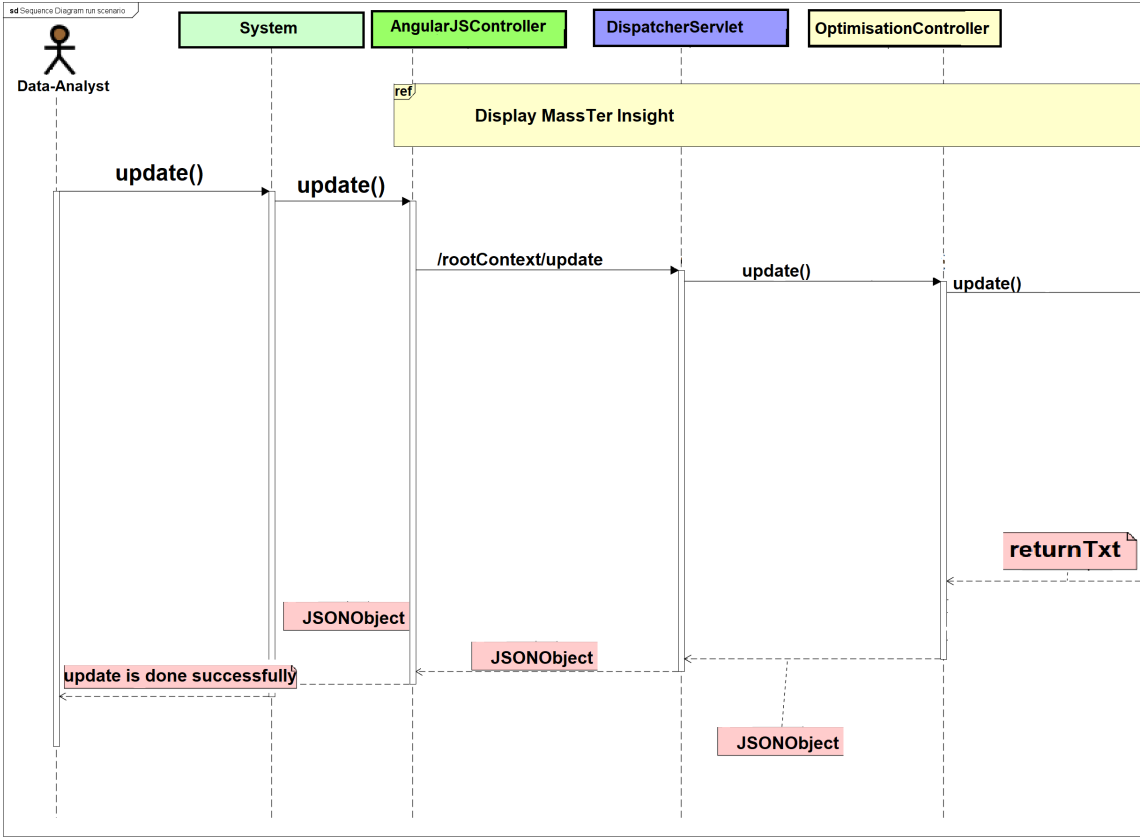


Figure 3.11: Sequence Diagram **update settings** Use Case





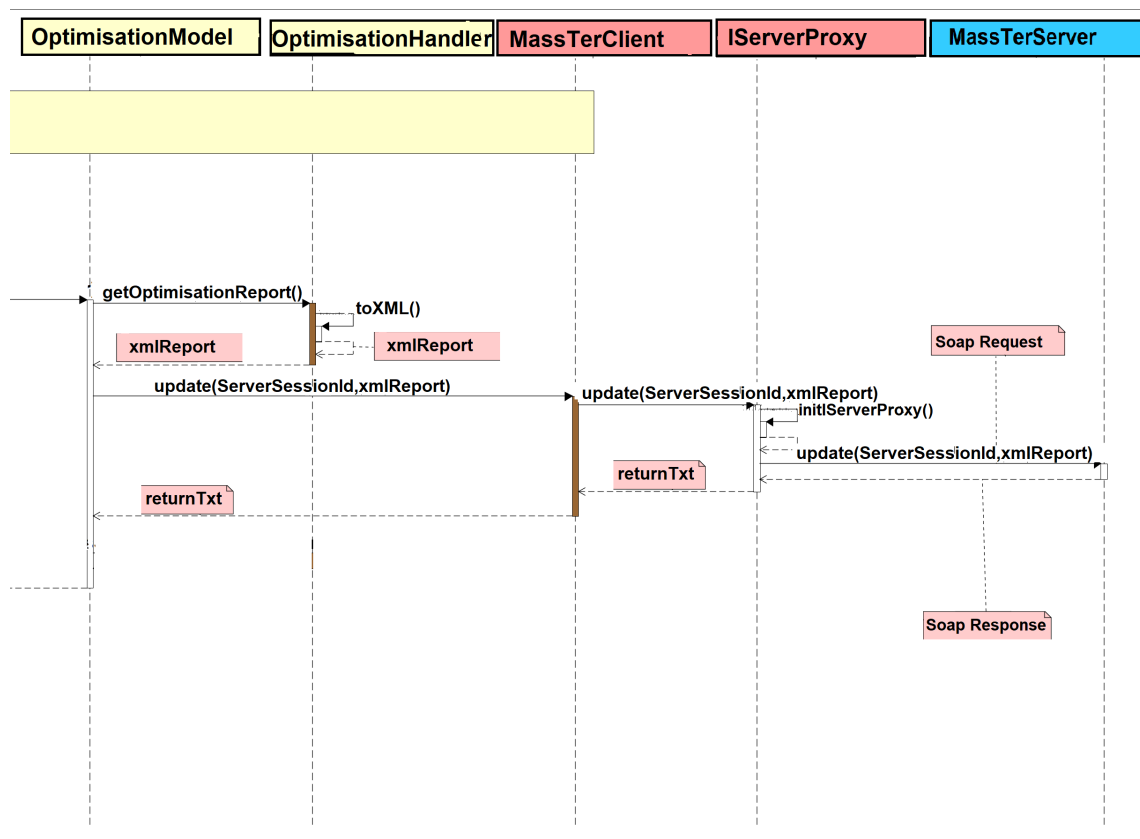
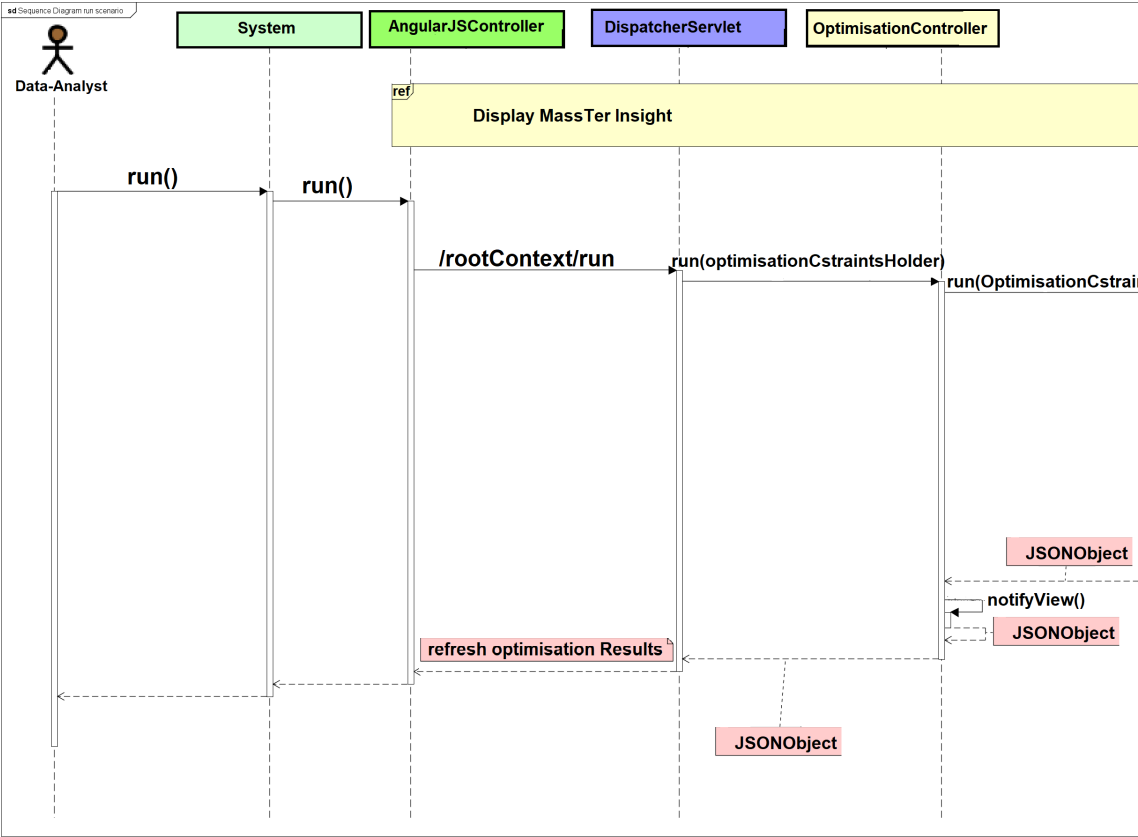


Figure 3.12: Sequence Diagram **Run Scenario Use Case**



Sequence Diagram use case “save project”

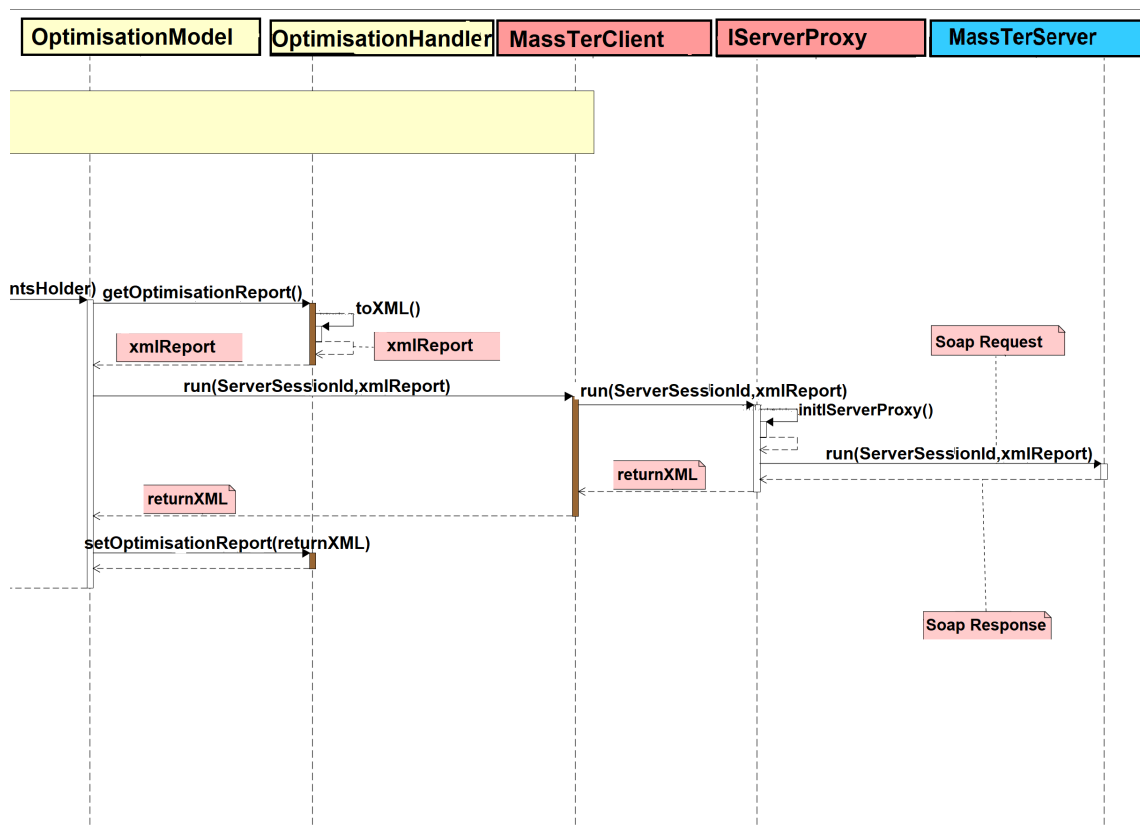
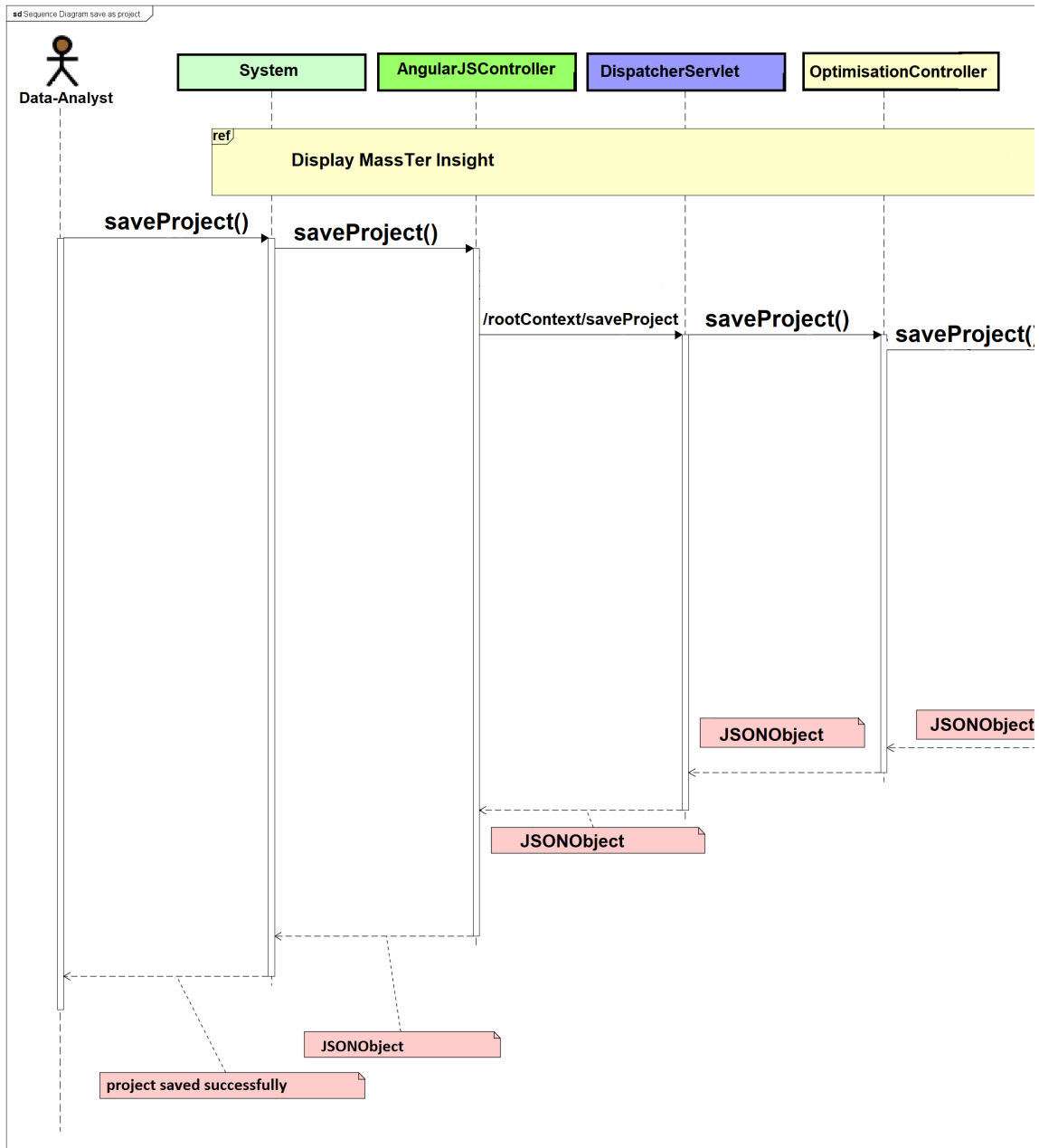
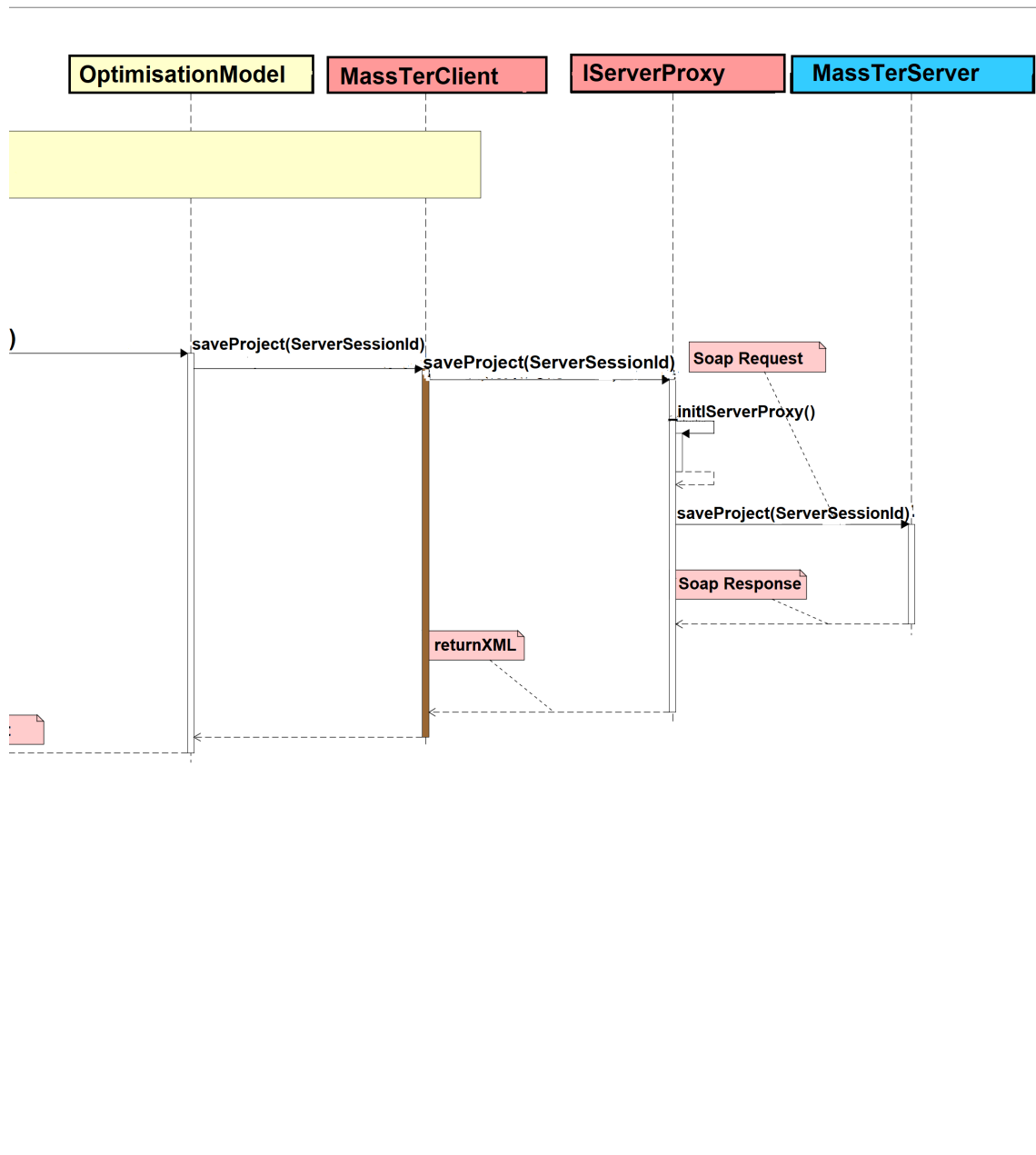


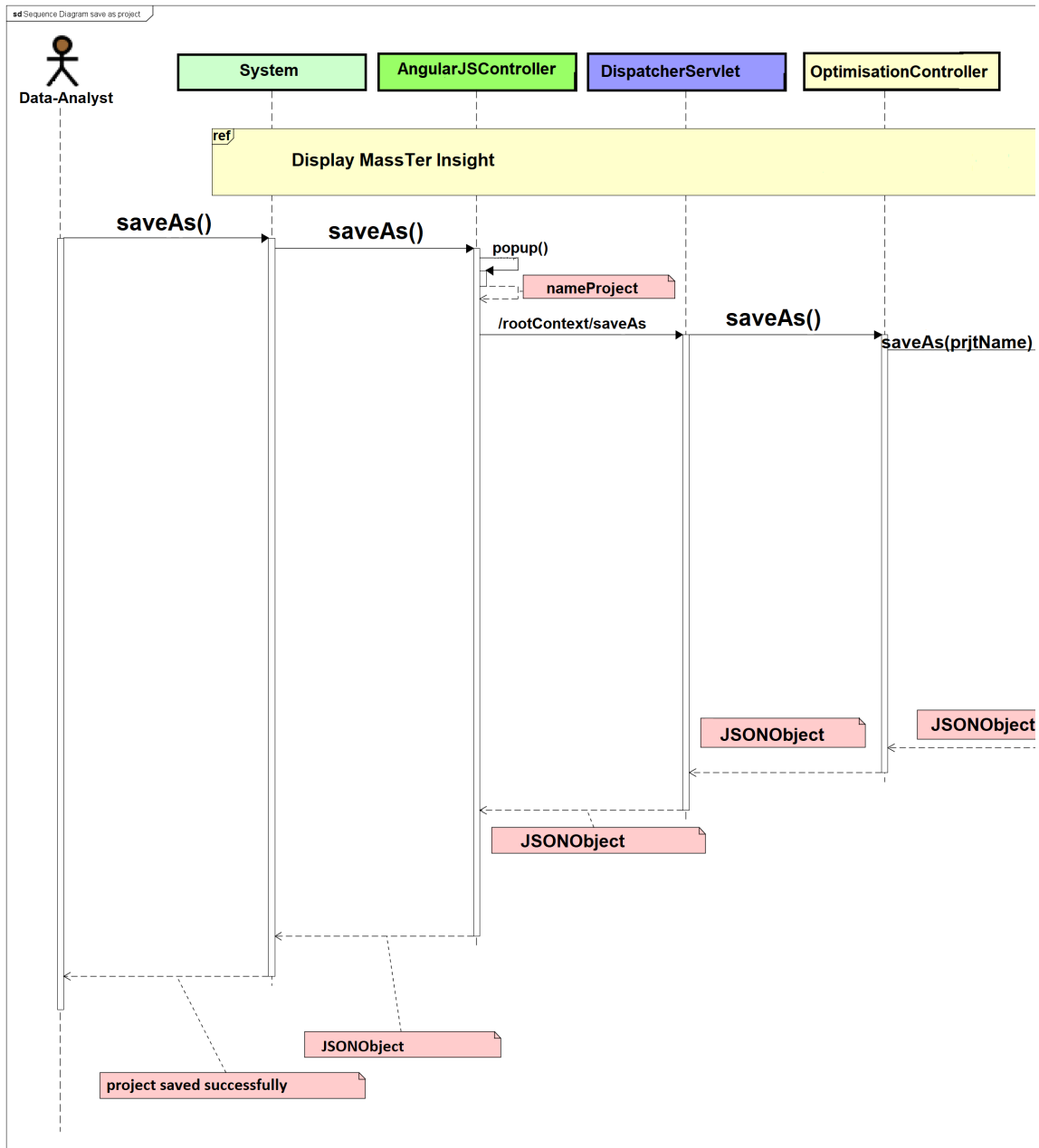
Figure 3.13: Sequence Diagram Save Project Use Case

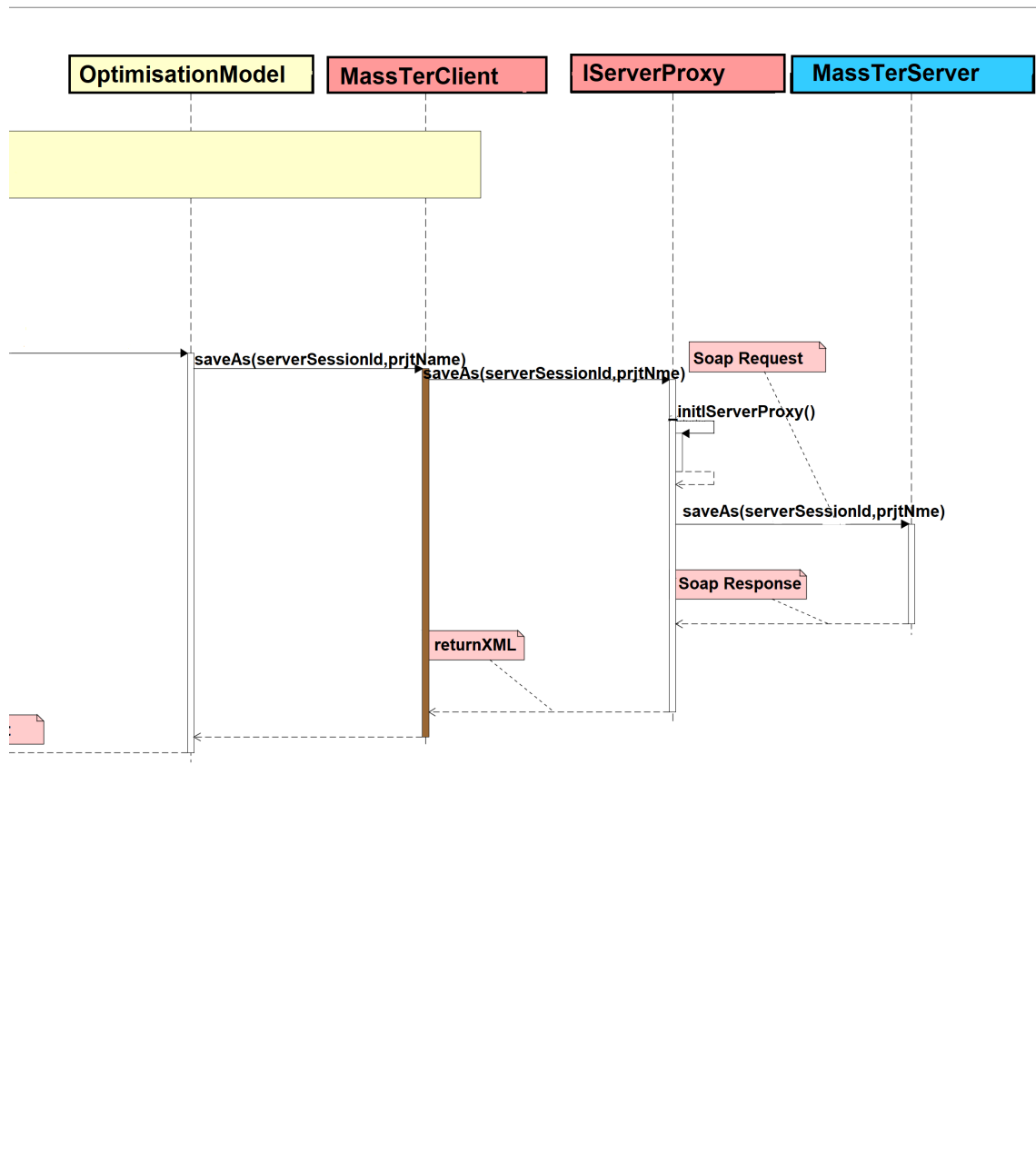




## Sequence Diagram use case Save As Project Use Case

Figure 3.14: Sequence Diagram Save As Project Use Case







## V Conclusion

Throughout this chapter, we have dissected the application to achieve gradually. We started by presenting the general architecture of the application, then we explained the global conception, unveiled through the packages diagrams, to go then to the detailed conception using class diagram, the sequences diagrams.

# Chapter 4

## Implementation

### I Introduction

In this chapter we will focus on the technologies.

## II Java Platform, Enterprise Edition (JEE)



Figure 4.1: Java Enterprise Edition

### 1 Introduction

**Java EE** is the Java platform edition for Enterprise Software, extending **Java SE** with APIs for enterprise features such as distributed computing and web services. Java EE applications are run on an application server, which handle transactions, security, scalability, concurrency and management of the components it is deploying.

### 2 Features

The main advantages of using Java EE are :

- **Portability**
- **Independence**
- **Security**
- **The multitude of libraries it offers**

The Java EE platform is based on specifications, which means projects are portable on any compliant application server (GlassFish, JBoss...) to these specifications. This implementation is free and allows you to benefit from the entire API without any investment. The Java EE platform is the richest of Java platforms and provides a standard environment for multi-tenant business application development and execution.

The JEE platform provides the following :

- Complete Web services support. The JEE platform provides a framework for developing and deploying web services on the Java platform.
- The Java API for XML-based RPC (JAX-RPC) enables Java technology developers to develop SOAP based interoperable and portable web services.
- Developers use the standard JAX-RPC programming model to develop SOAP based web service clients and endpoints.
- A web service endpoint is described using a Web Services Description Language (WSDL) document.
- JAX-RPC enables JAX-RPC clients to invoke web services developed across heterogeneous platforms. In a similar manner, JAX-RPC web service endpoints can be invoked by heterogeneous clients

### 3 Motivation

According to a trusted source "TIOBE index", Java is the most popular language ever. The TIOBE Programming Community index [1] is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. As the above statics show, JAVA is the most used language in the industry. Today java it is not an option, it is a requirement for almost all the IT Job Apply.

Other Major key that push us to use java as language of programming in our project (MassTerInsight) [2] is that the core business of MassTer software [3] is developed using java. As a consequence, it is easy to integrate in our project without any midelware or such web services.

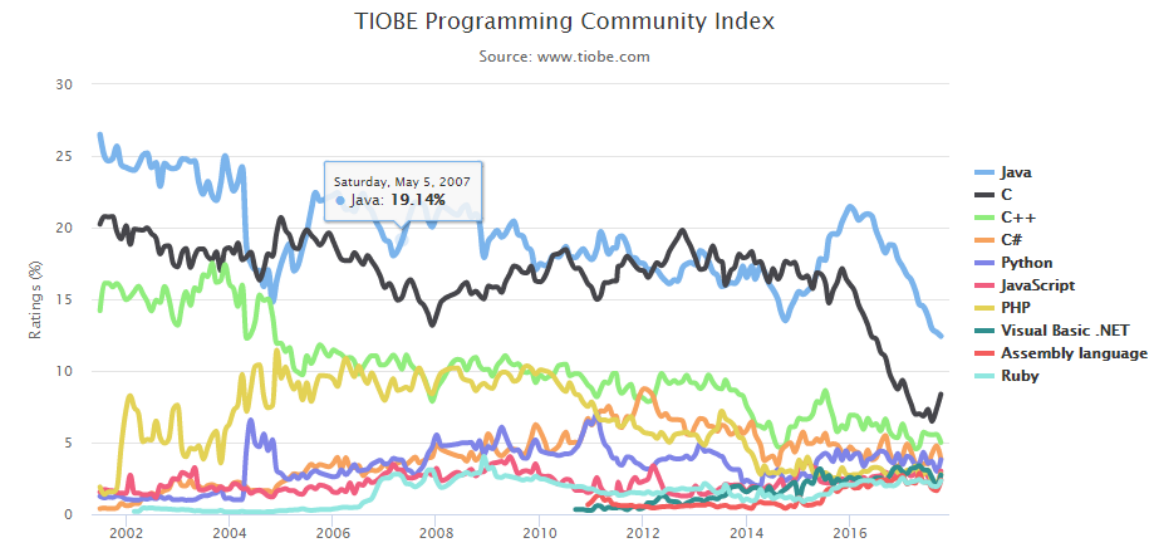


Figure 4.2: java statics

Oct 2017	Oct 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.431%	-6.37%
2	2		C	8.374%	-1.46%
3	3		C++	5.007%	-0.79%
4	4		C#	3.858%	-0.51%
5	5		Python	3.803%	+0.03%
6	6		JavaScript	3.010%	+0.26%
7	7		PHP	2.790%	+0.05%
8	8		Visual Basic .NET	2.735%	+0.08%
9	11	▲	Assembly language	2.374%	+0.14%
10	13	▲	Ruby	2.324%	+0.32%

Figure 4.3: java rank

### III Angular JS



Figure 4.4: AngularJS

#### 1 Introduction

AngularJS is a structural framework for dynamic web applications.

It gives to the developers the way to interact with their html components in ease way instead of using a long javascript code.

The impedance mismatch between dynamic applications and static documents is often solved with:

- **a library** - a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., `jQuery`.
- **frameworks** - a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something application specific. E.g., `durandal`, `ember`, etc.

AngularJS takes another approach. It attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs. AngularJS teaches the browser new syntax through **directives**. Examples include:

- Data binding, as in `{{}}`

Directive	Description
ng-app	tells AngularJS that this is the root element of AngularJS application, we can only have one ng-app directive in the application.
ng-model ng-switch, ng-switch-when	binds an HTML form element to a variable in the scope the directive ng-switch lets you hide or show HTML elements depending on an expression. Child elements with the directive ng-switch-when will be showed up if it gets match, otherwise the element, and its children will be removed.
ng-switch-default	define a default section if none of the other sections get match

- DOM control structures for repeating, showing and hiding DOM fragments.
- Support for forms and form validation.
- Attaching new behavior to DOM elements, such as DOM event handling.
- Grouping of HTML into reusable components.

## 2 Features

- AngularJS is a powerful JavaScript based development framework to create Rich Internet Application(RIA).
- AngularJS is a free open source framework.
- Large of developers around the world used AngularJS

Overall, AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain.

We present in this table the directives used in our project so far.

### 3 Motivation

Before starting to use any javascript framework, we have made a research about the most recommended javascript framework.

Almost all the articles that I read shows that the most popular frameworks are AngularJS, Ember.js, ReactJS and Backbone.js, AngularJS is the most used framework amongst these frameworks and this is my first motivation.

Based on recent search on Google Trends, the requirements skills of the industry in all over the world tend to use more and more the framework AngularJS

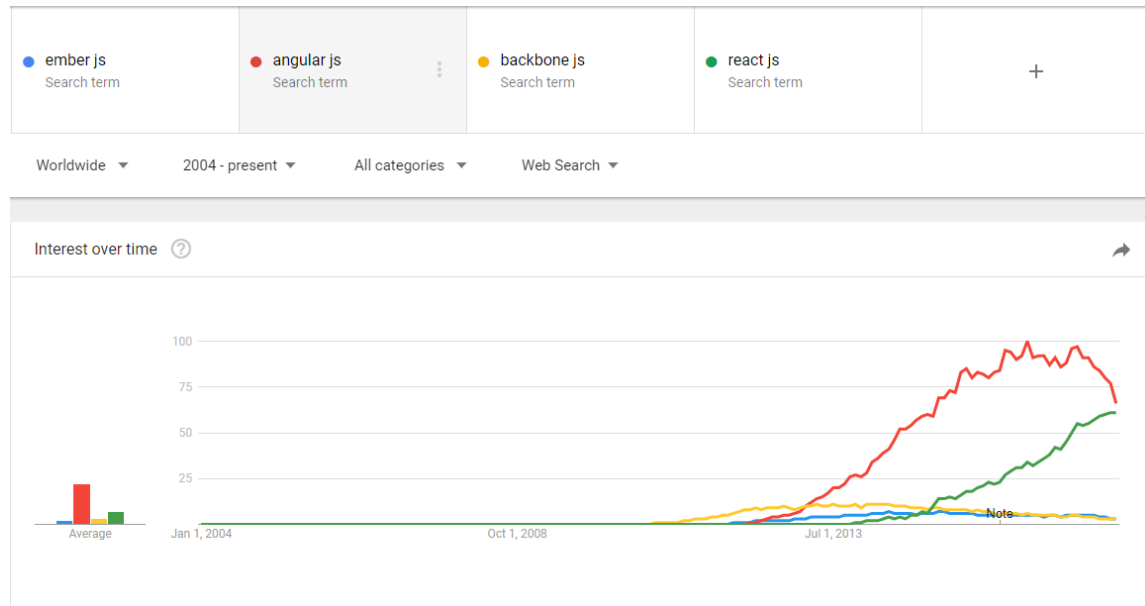


Figure 4.5: AngularJS on google trends



## IV Bootstrap



Figure 4.6: Bootstrap

### 1 Introduction

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript plugins. Unlike many web frameworks, it considers front-end development.

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub!

### 2 Features

**Bootstrap 3** supports the latest versions of the **Google Chrome**, **Firefox**, **Internet Explorer**, **Opera**, and **Safari** (except on Windows). It additionally supports IE8 and the latest Firefox Extended Support Release (ESR).

Since **2.0**, **Bootstrap** supports **responsive web design**. This means the layout of web pages adjusts dynamically, taking into account the characteristics of used device (desktop, tablet, mobile phone).

Starting with **version 3.0**, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default. The **version 4.0** alpha release added **Sass** and **flexbox** support.

### 3 Motivation

All the research shows that Bootstrap is The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.

According to github, Bootstrap is the second starred repository, with **115k stars**.

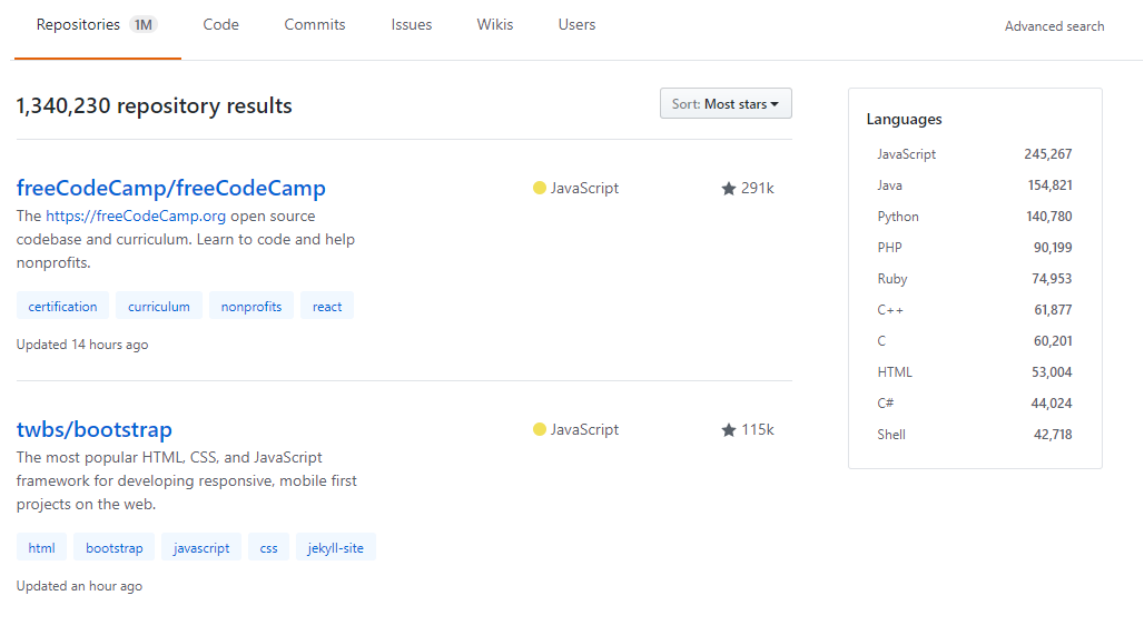


Figure 4.7: Bootstrap second starred on github

According to a research in the net, all the high-tech blogs encourage developers

to use bootstrap.

I used a tool offered by google named **google trends** to make a comparison between subjects in term of most researched.

After this comparison in google trends, Bootstrap also is the most googled css framework on the web , compared to the others css frameworks Eg. **Material UI** which was considered as the most popular css framework after Bootstrap.

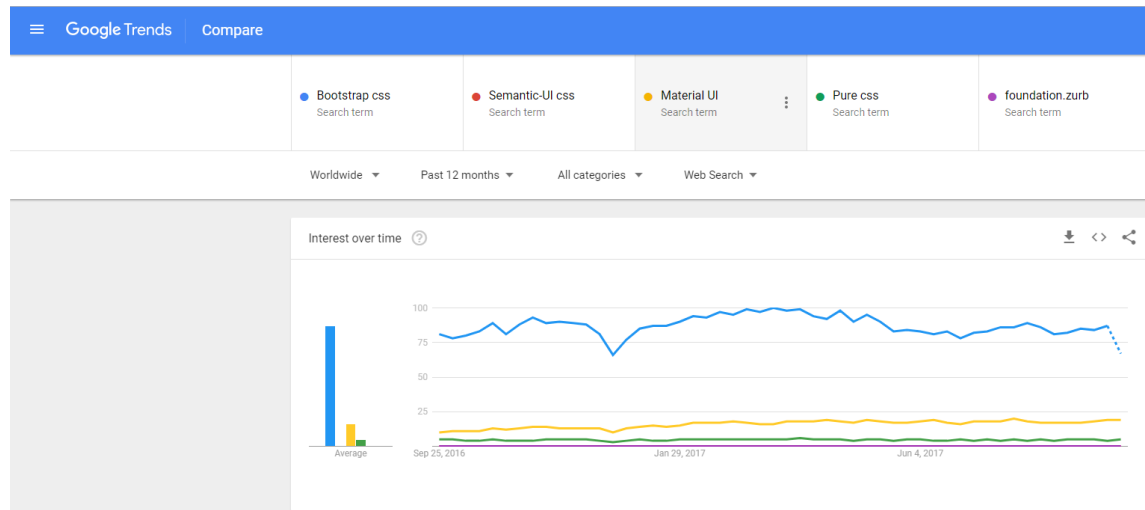


Figure 4.8: Bootstrap on google trends

## V Spring mvc framework



Figure 4.9: Spring MVC

### 1 Introduction

The Spring Web model-view-controller (MVC) framework is designed around a `DispatcherServlet` that dispatches requests to handlers, with configurable handler mappings, view resolution, locale, time zone and theme resolution as well as support for uploading files. The default handler is based on the `@Controller` and `@RequestMapping` annotations, offering a wide range of flexible handling methods. With the introduction of Spring 3.0, the `@Controller` mechanism also allows you to create RESTful Web sites and applications, through the `@PathVariable` annotation and other features.

Spring's web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central Servlet that dispatches requests to controllers and offers other functionality that facilitates the development of web applications. Spring's `DispatcherServlet` however, does more than just that. It is completely integrated with the Spring IoC container and allows you to use every other feature that Spring has.

The request processing workflow of the Spring Web MVC DispatcherServlet is illustrated in the following diagram. The pattern-savvy reader will recognize that the DispatcherServlet is an expression of the “Front Controller” design pattern (this is a pattern that Spring Web MVC shares with many other leading web frameworks).

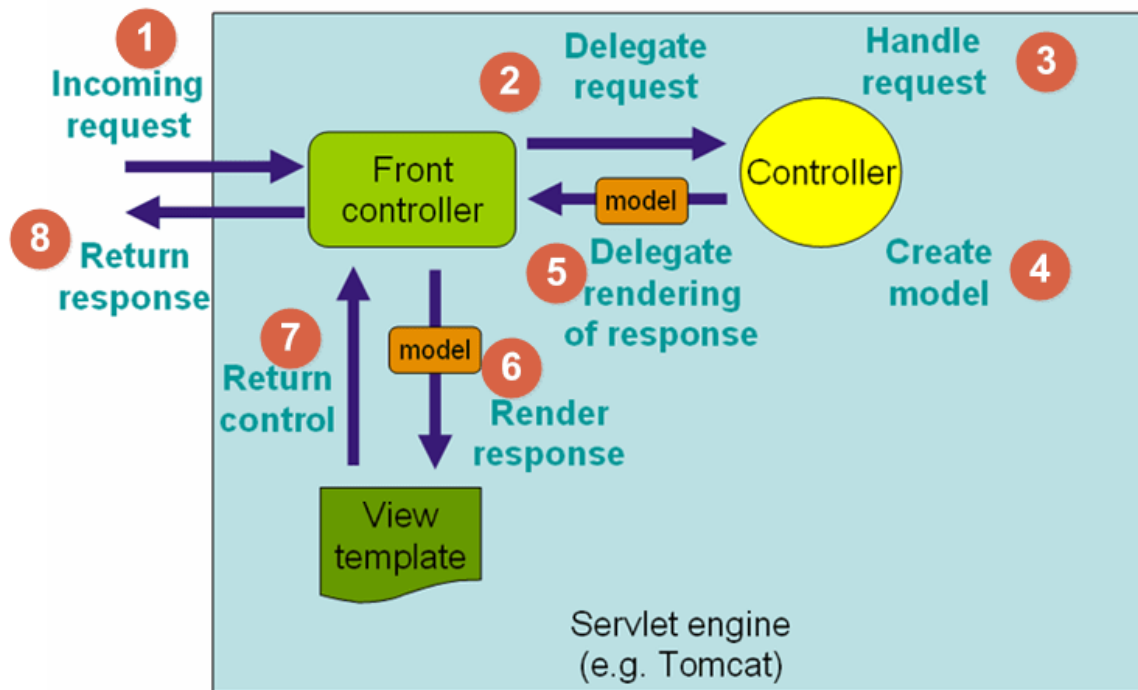


Figure 4.10: The request processing workflow in Spring Web MVC

## 2 Features

- Since Spring MVC framework is designed like any other Spring module, So, it is not necessary to spend any extra time to learn it.
- Since all the layers are independent of each others, unit testing can be easier.
- Spring framework doesn't force you to follow any pattern, or implementations to write your business logic. So it gives developer flexibility to implement or integrate any other design pattern to suffice his needs.

Annotation	Description
@Controller	it is responsible for preparing a model Map with data and selecting a view name but it can also write directly to the response stream and complete the request.
@RequestMapping	it used with method to provide the URI pattern for which handler method will be used.
@RequestBody	the @RequestBody method parameter annotation indicates that a method parameter should be bound to the value of the http request body.
@ResponseBody	it used to send String response for the web request.
@RequestParam	it used to retrieve the URL parameter and map it to the method argument.

- Spring provides good separation flexibility between Controller, Service and Data access layers.
- Spring provides you with a tag library that is simple and yet very powerful.
- View side you can integrate with any UI framework like JSF, Velocity, Freemarker etc.
- Supports Annotation based programming along with XML, which makes development faster and cleaner.

We present in this table the annotations used in our project :

### 3 Motivation

According to trusted sources like google engine, yahoo, Linkedin and satchoverflow: spring mvc, struts And JSF are the most JEE frameworks searched and posted by the developers.

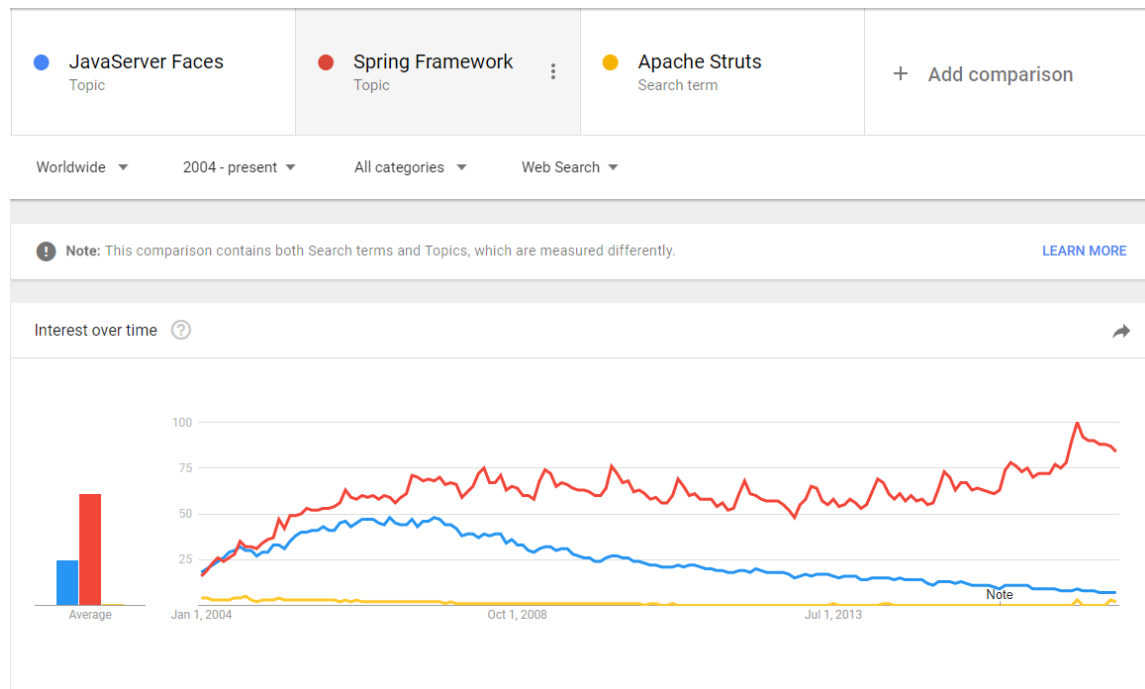


Figure 4.11: Spring framework on google trends

The curve of spring mvc in the figure 4.11 is very high, which means that spring is the most used among other frameworks, another key is that spring is the most framework recommended on job application.

The community behind spring is very active.

## VI Pivotal tc Server



Figure 4.12: Pivotal tc Server

Pivotal tc Server is a open-source Web application server based on Apache Tomcat.

Pivotal tc Server provides :

- The best of Apache Tomcat.
- Harnesses the power of traditional JEE architectures.
- Eliminates the JEE's complexity and performance drawbacks, making it easier, faster.
- Pivotal tc Server requires significantly fewer resources than conventional servers.
- It is completely compatible with Apache Tomcat.



## VII Spring Tool Suite



Figure 4.13: spring tool suite

- The Spring Tool suite is an eclipse-based developing environnement that customiwed for developement Spring applications.
- Provides a enviroment to implement, debug, run, and deploy.
- It integrates Git,Maven, Pivotal tc Server but also you can use other Web Application Server.

## VIII Apache Maven



Figure 4.14: Apache Maven

Maven is an open-source tool developed by the foundation Apache, which main role is to build projects and the management of libraries used.

Maven provides :

- compilation
- packaging
- dependancy management
- deployment

## IX HTML5



Figure 4.15: HTML5

HTML5 is the last version of HTML, it comes with new semantics tags (header, aside, ...) to divide your page in semantic way, those tags it helps the engine (google, yahoo...) to perform the results. It is easy to learn, compatible with all browsers, now HTML5 replace the flash, it include tag video, audio, geolocation, and more.

## X Web Service

# Chapter 5

## Deployment

### I Introduction

# Bibliography

- [1] Mass. Mass-analytics, November 2017. [www.mass-analytics.com](http://www.mass-analytics.com).