



---

# Visual Portfolio Requirements Definition Document

Version 1.0

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

## Revision History

Date	Version	Description	Author
2017-10-26	0.1	Initial Draft	Valentina, Luca
2017-11-04	1.0	First complete document	Valentina, Luca

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

## 1. Introduction

### 1.1. Purpose of this document

The purpose of this document is to describe and clarify the requirements of the project “Visual Portfolio” included in the DSD course 2017/2018.

This document is useful to provide a common basis for the future development; it will also be used to guide the development of the system later.

### 1.2. Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes contents of this guide, used documentation during developing process, intended audience and scope of the project.
- Section 2, *Background and Objectives*, describes the background and the goals of the project.
- Section 3, *Web Application*, describes the web application requirements, user stories and use cases.

### 1.3. Intended Audience

The intended audience is:

- Team members
- Supervisors
- Customer

### 1.4. Scope

The document provides general background information and objectives about the project. It also contains the requirements and user stories of the web application. The document also provides the high level requirements description of the project. Both the functional and nonfunctional requirements of the project are presented using some UML diagrams such as use case diagrams, sequence diagrams and activity diagrams.

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

## 1.5. Definitions and Acronyms

### 1.5.1. Definitions

Keyword	Definitions
<i>User</i>	A person who uses Visual Portfolio application.
<i>Kommuninvest Employee</i>	A user from Kommuninvest.
<i>Kommuninvest Customer</i>	A user from one of the Kommuninvest's Clients
<i>Backend</i>	The system developed to manage data.
<i>Frontend</i>	Web application that the user is going to use to analyze and visualize portfolios.

### 1.5.2. Acronyms and abbreviation

Acronym or abbreviation	Definitions
<b>NTR</b>	Nothing to Report. There is no information to a specific topic available or necessary.
<b>UML</b>	Unified Modeling Language
<b>GUI</b>	Graphical User Interface

## 1.6. References

### 1.6.1. Project plan document

[https://www.fer.unizg.hr/download/repository/Project\\_plan\\_-\\_1.0-2.pdf](https://www.fer.unizg.hr/download/repository/Project_plan_-_1.0-2.pdf)

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

## 2. Background and Objectives

The customer for this project is Kommuninvest, a Swedish local government funding agency based in Örebro, Sweden. Kommuninvest currently has a debt management system named KI Finans. This system can be used by their customers in order to track transactions, perform analysis and simulations, and extract exports.

The purpose of this project is to create a prototype of a visual analysis tool that can be used to show the performance of individual debt portfolios or compare portfolios from different customers with each other and/or financial markets. More precisely, the objective of the Visual Portfolio Project is to develop a web-based application where users can compare different types of bond portfolios using either time-series graphs or scatter plots. According to the customer, the idea is to "give easy access to economic and financial key indicators". It should be easy to choose the peer-group benchmarks, which indicators to analyse and how the indicators should be displayed.

The project team will not develop the final product that Kommuninvest will use, just a prototype. The goal is that this project will produce well-defined requirements and design that can be used by a company to implement the final product.



Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 3. System Requirements

#### 3.1. Actors

##### 3.1.1. Customer

Customers login into the application and will see their own portfolio. Customers are able to compare their own portfolio with the average of a peer group. Customers choose between different entities (municipality, municipality group and companies) filtering displayed data. Customers create different type of charts, choosing to populate axis with the whole pool of data available in database, creating an high customizable method to display data. Customers could also apply various types of filter, and download charts in different formats. The Customer actor is represented by Kommuninvest's Client

##### 3.1.2. Account managers

Account managers can do everything that customers can do, but in addition to that, they can see and compare any portfolios without constraints about peer groups. Account managers can navigate through all the data saved in database thanks to GUI. The account manager actor represents a *Kommuninvest Employee*.

##### 3.1.3. Admin

Admin is a special profile created in order to add, edit or delete Customers and Account Managers from the system. Through an API that manages users' accounts (that will be added due to customer's request) the system will allow to manage its users' accounts from an external system. Moreover, Admin be used also for managing the application from a high level point of view during, for example, update or maintenance.

#### 3.2. Functional requirements

In this section we describe functional requirements of the system using a UML use case diagram. The list of functional requirements is listed below. For simplicity purpose, in **Motivation/Source** column **A.** stands for "gotten from customer" and **B.** stands for "identified by the project team"

Code	Name	Description	Motivation/Source	Priority
FR1	<i>Login Customer</i>	Customer logs into the system using credentials.	B	High
FR2	<i>Login Manager</i>	Account Manager logs into the system using credentials.	B	High
FR3	<i>Login Admin</i>	Admin logs into the system using credentials.	B	High
FR3	<i>Choose entity</i>	The user selects between different entities: municipality, municipality group, company.	A	High
FR4	<i>Visualization of personal portfolio</i>	The user visualizes his/her own portfolio.	A	High

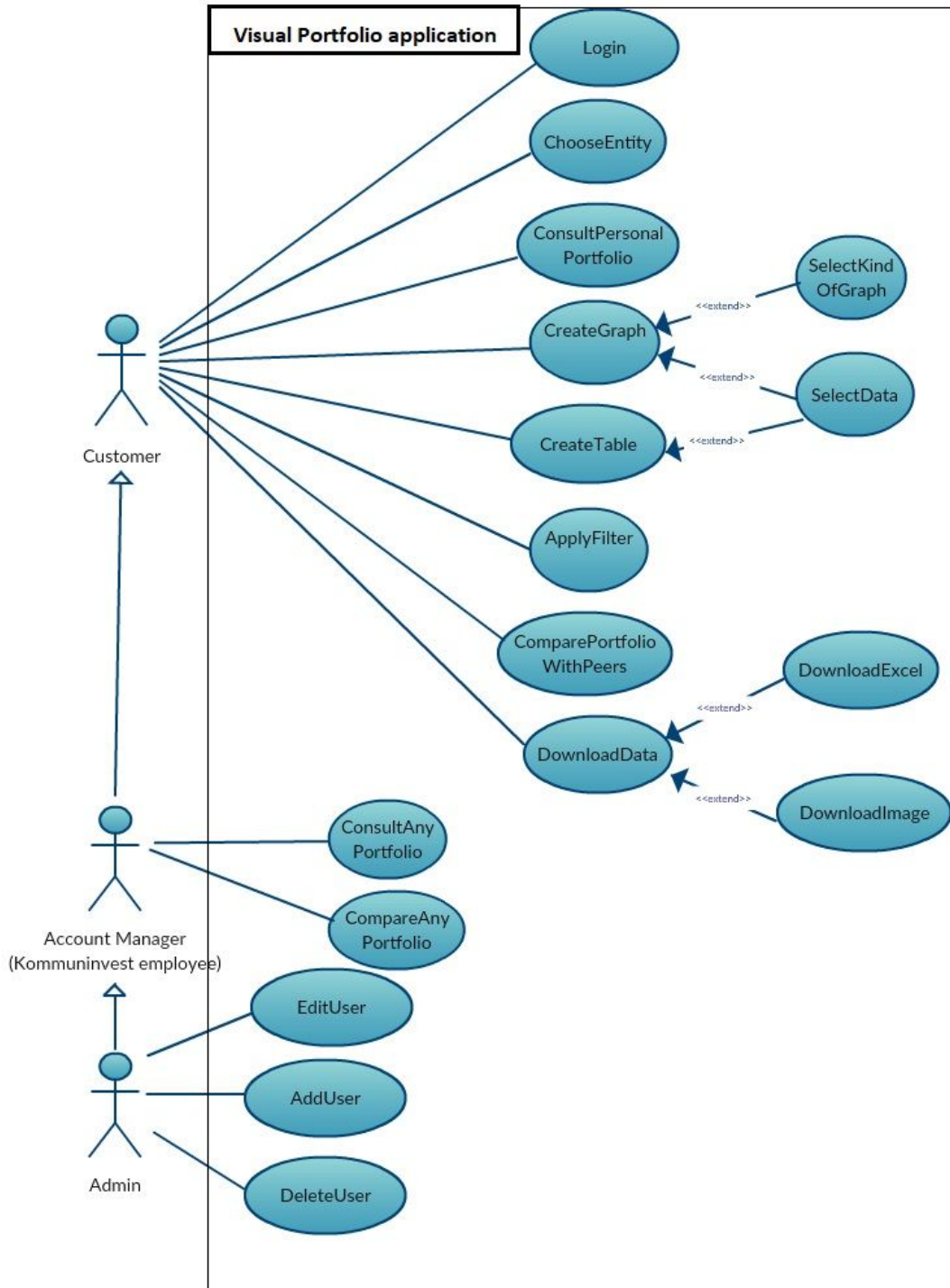
Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

FR5	<i>Visualization of any portfolio</i>	The account manager visualizes any portfolio he/she wants.	A	High
FR6	<i>Comparison of portfolio with peers</i>	The Customer compares their company's portfolio with their peers.	A	High
FR7	<i>Comparison of any portfolios</i>	The Account Manager compares any kind of portfolios.	A	High
FR8	<i>Creation of a graph</i>	The user generates a graph.	A	High
FR8.1	<i>Selection kind of graph</i>	The user decides which kind of graph to plot.	B	Low
FR8.2	<i>Selection of data</i>	The user decides which kind of data the graph has to show.	B	High
FR9	<i>Creation of table</i>	The user generates a table.	B	High
FR9.1	<i>Selection of data</i>	The user decides which data to insert in the table.	A	High
FR10	<i>Filtration of data</i>	The user can apply filters on graphs and tables he/she generated	A	High
FR11	<i>Download of displayed data</i>	The user downloads data as an image (PNG or JPEG format) or as row data (SVG with images).	A	High
FR12	<i>Additional filter</i>	The user can apply additional custom filters, defined by the user himself, to displayed chart/table.	B	Low
FR13	<i>Add user</i>	An admin creates a user	B	Mid
FR14	<i>Edit user</i>	An admin edits an existing user	B	Mid
FR15	<i>Delete user</i>	An admin removes an existing user	B	Mid
FR16	<i>Add data from external input source</i>	Account manager can insert new data via uploading files such as Excel or CSV	B	Mid
FR17	<i>Accounts management through an API</i>	To manage (create, modify and delete) system's accounts by the Kommuninvest Admin. Not visible to the end-user.	A and B	Low

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 3.3. Use Case diagram

Below you can view initial Use Case Diagram of our application. Functional Requirements were started to derive from here. Please note the functional inheritance between the actors in the system.





Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 3.4. Non functional requirements

In addition to functional requirements the system must satisfy this nonfunctional requirements. Listed below, they are described in details afterwards, in the context of our application.

Code	Name	Description	Priority
NFR1	<i>Security and Data Sensitivity</i>	Nobody should be able to access the application or the database without authentication.	Mid
NFR2	<i>Performance</i>	The system has not to waste time on page refreshes, so it must load the data asynchronously via AJAX calls.	High
NFR3	<i>User-friendly interface</i>	The application's interface has to be: clean, simple, intuitive and reliable.	High
NFR4	<i>API Documentation</i>	APIs will be documented through Swagger.io.	Low
NFR6	<i>Availability</i>	The service must run in a good way in positive conditions (internet connection stable).	Low

#### 3.4.1. Security and Data Sensitivity

- Any data visualized by the application will be accessed only after a session is started. A session can be started with the login procedure.
- Nobody will be able to use the application without a valid and active account. Active or inactive state will be a characteristic of each account in the system.
- No sensible information will be accessible to a third party (e.g. journalists) to avoid any data leakage.
- User roles will have different data access permissions. For example, Kommuninvest Client can view only data of his and the peers, when Kommuninvest Employee can view any kind of data

#### 3.4.2. Performance

- This will be a Single-page application. In other words, templates of different pages are loaded in a single page. Data and page sections will be loaded asynchronously via AJAX. This means no wasted time on page refreshes. Requests that will be run in background, dependently on the connection speed, will be executed within 500 ms.
- Being a private application for only Kommuninvest Clients we do not expect to have high volume or requests with a limited number of users.
- Future improvements and deployments will be easy to apply when needed thanks to the modularity of the architecture that we have decided to implement (i.e. REST architecture - to be explained later).

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 3.4.3. *User-friendly Interface*

- **User-friendly**
  - The application GUI should be appealing to the user when used. Graphical interface guidelines and best practices will be followed in order to achieve the expected behaviours for users. “Material design” (<https://material.io/>) is one of the most famous graphical style so we decided to use it.
  - Reactivity: the user is happy when task are done in the fastest way possible.
  - Easily understandable by a user with an average experience in using applications or surfing online.
  - Ready template, that is used and tested a lot. This brings more reliability during rendering of GUI.
  - Adaptiveness: layout is adaptive to tablet and smartphone sized screens. The graphical user interface will remain comfortable and well usable even on small size screens.
- **Ease of use**
  - Filtering data should be done with the minimum number of steps.
  - GUI must be intuitive. Even without further details on each button, the user must be able to understand what a particularly shaped button does. For example, by using Icons (e.g. Trash Can for Delete) or titles on button hover.
  - No eye-disturbing visualisations such as animated banners or Flash elements.
  - Data to be visualized must be displayed in an understandable way. A person with no prior knowledge in Statistics or Charts/Graphs should easily manage to navigate through them, and derive information that he came here for. :
- **Understandability**
  - Interface elements (e.g. menus) should be easy to understand.
  - The screen layout and colour should be appealing and design to be similar to the Kommuninvest website style.
- **Learnability**
  - The user documentation and help should be complete.
  - The interface should be simple and allow user to achieve complex tasks in easy few steps
  - The system should be easy to learn.
- **Operability**
  - The interface actions and elements should be consistent.
  - Error messages should explain how to recover from the error
  - Undo should be available for most actions
  - Actions which cannot be undone should ask for confirmation
  - The system must be customisable to meet specific user needs
  - A style guide should be used

### 3.4.4. *Documentation*

- If successful, our prototype is to be used by the Kommuninvest for further development. Which means all design and architecture decisions must be reasoned and documented, code and scripts to be well commented. For example, a tool named Swagger (<https://swagger.io/>) will be used to auto-generate documentation for our REST API calls/architecture. This is very handy for our total separation and independence of the Back-end and Front-end

### 3.4.5. *Availability*

The Visual Portfolio application should guarantee high availability and data redundancy. However, since the application will be created in the context of the DSD course, our team will not build as wasn't required any dedicated infrastructure for it. So estimating and providing exact value for data redundancy and uptime is not possible. However, in the case there's the chance to build and test a dedicated infrastructure, an uptime of at least 99.99% is desirable along with at least one database replication.

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

#### 4. User stories

This section describes requirements as user stories to give an overview of the main functionalities required. User stories are short and simple sentences that contain the features customers expect to find in the system. The customer's requirements are not equally important; for this reason high, average or low priority is attributed to each of them.

Code	User story	Description	Priority	Related Use Cases
S1	As an administrator I want to add a user	Through a dedicated page, an Administrator is able to create new profile for users.	Mid	FR13
S2	As an administrator I want to edit a user	Through a dedicated page, an Administrator is able to edit existing users' profile.	Mid	FR14
S3	As an administrator I want to remove a user	Through a dedicated page, an Administrator is able to delete existing users' profile.	Mid	FR15
S4	As a user I want to be able to login (or logout) into the system with my account at any time.	To use the service the user must provide credentials.	High	FR1/FR2
S5	As a user I want to be able to consult my own portfolio.	The application must show the user's portfolio when user select this function.	High	FR4
S6	As a user I want to be able to compare my portfolio with the average of my peers.	The application must compare the user's portfolio with the average of his/her peers when user select this function.	High	FR6
S7	As a user I want to filter displayed data.	The application must filter the data of graphs or tables.	High	FR11

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

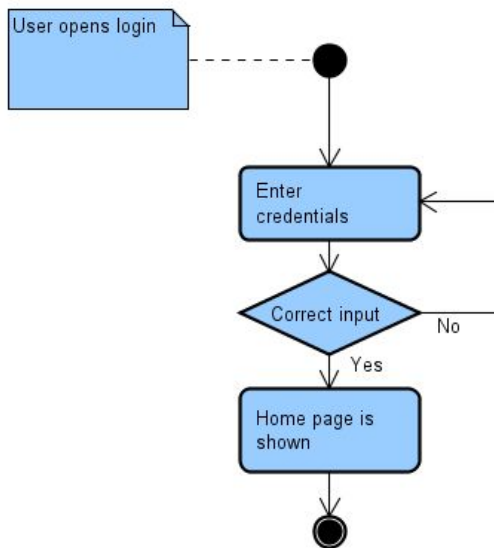
S8	As a user I want to download data displayed on the application in different format.	The application must download data displayed in different format if the user selects this option.	High	FR11/FR12
S9	As a user I want to understand my data better through meaningful charts and tables.	The application must provide the creation of charts and graphs in order to allow users to have a better comprehension of data.	High	FR8/FR9
S10	As an account manager I want to be able to see any portfolio.	The application must show any portfolio only if the user is an account manager.	High	FR5
S11	As an account manager I want to be able to compare any portfolio and/or peer group.	The application must compare any portfolio and/or any peer group only if the user is an account manager.	High	FR6
S12	As a user I want to choose a specific type of chart for my data.	The application must provide an option to select between different kinds of chars.	Low	FR8.1
S13	As an account manager I want to insert new data into the system using Excel or CSV.	The application must provide the possibility to insert data using Excel or CSV.	Low	FR14

Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

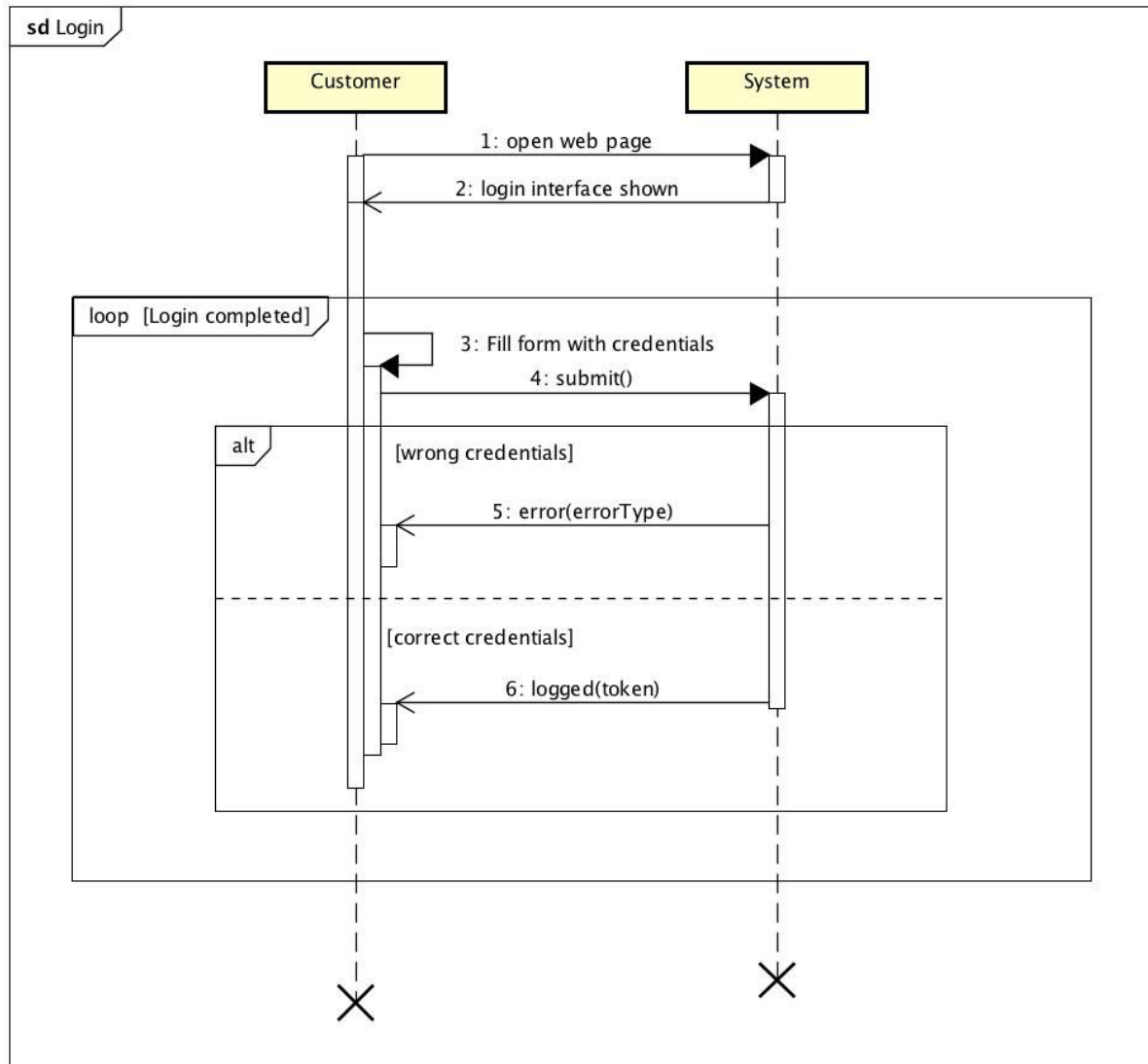
## 5. Use Case description

### 5.1. User login

<b>Name</b>	User login
<b>Actor</b>	User (Customer, Account manager, Admin)
<b>Entry conditions</b>	<ul style="list-style-type: none"> <li>• Internet connection working</li> <li>• Existing account for the user</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Login web page opened</li> <li>2. User enter his/her credentials</li> <li>3. "Login" button pressed</li> </ol>
<b>Exit conditions</b>	<ul style="list-style-type: none"> <li>• New user session is created in the system</li> <li>• Homepage is shown</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Credentials are wrong</li> <li>• Internet connectivity is lost by the client</li> <li>• User tries to login without having an account present in the system</li> </ul>



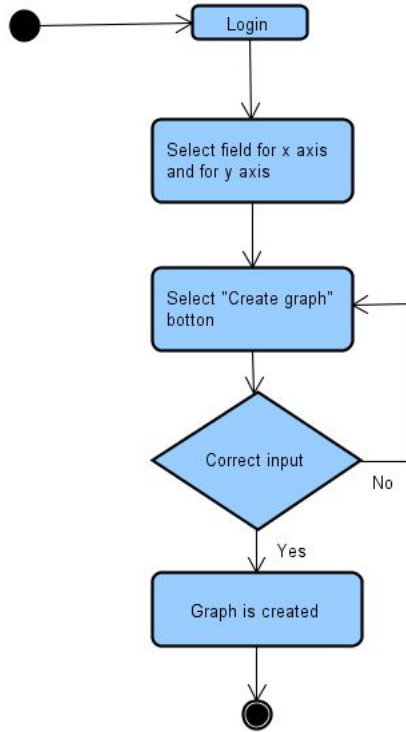
Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04



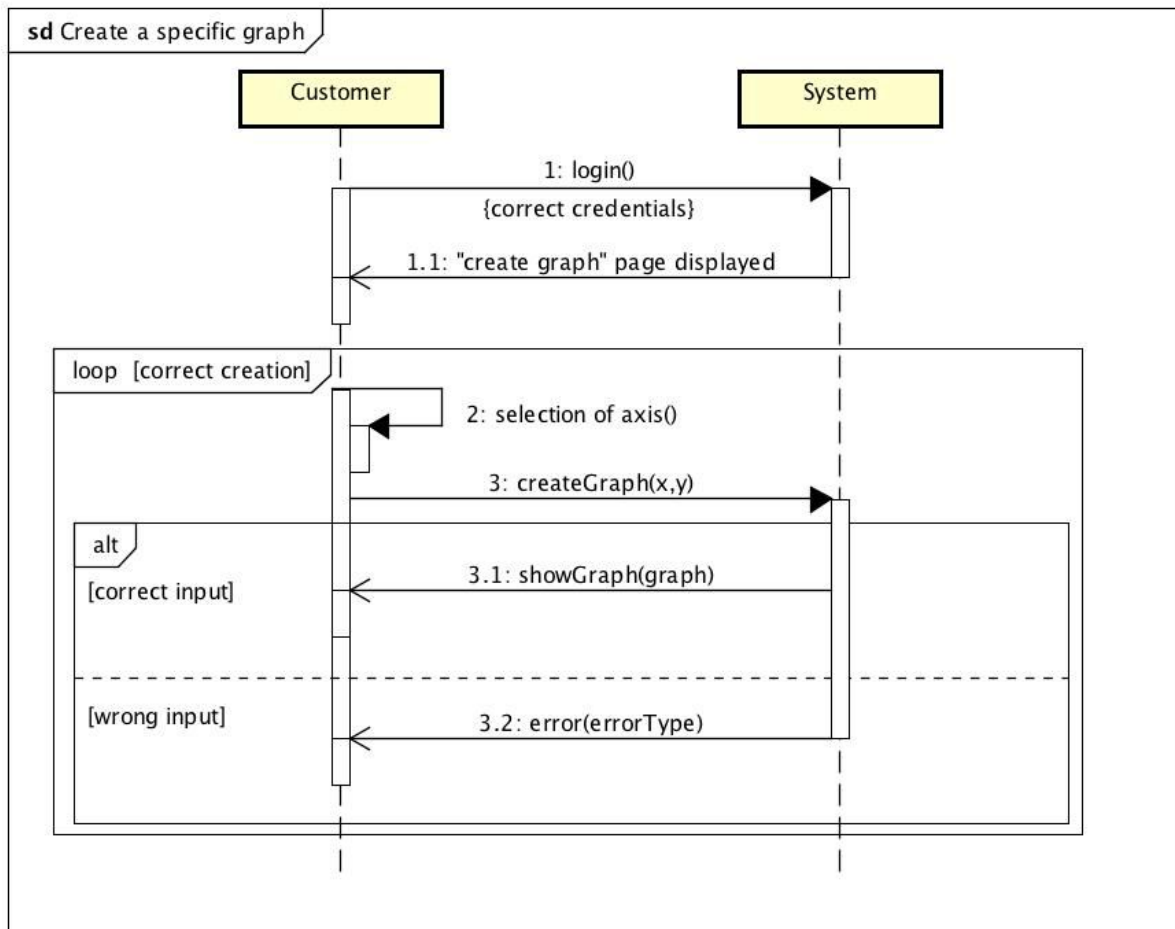
Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 5.2. Create specific graph

<b>Name</b>	Creation of specific graph
<b>Actor</b>	User (Customer, Account manager, Admin)
<b>Entry conditions</b>	<ul style="list-style-type: none"> <li>• Logged in user</li> <li>• Login session not expired</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Login successful</li> <li>2. "Create graph" page is shown</li> <li>3. Field for x-axis and y-axis are selected</li> <li>4. "Create" button is pressed</li> </ol>
<b>Exit conditions</b>	A graph is created and it appears in the page
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• A graph is created with wrong or empty field for the axis and it doesn't appear in the page</li> <li>• User is disconnected due to session inactivity timeout, so he/she is logged out automatically</li> </ul>



Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

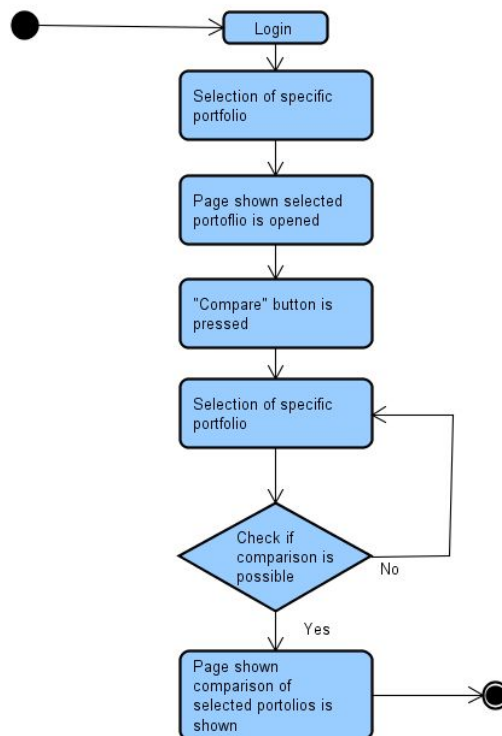




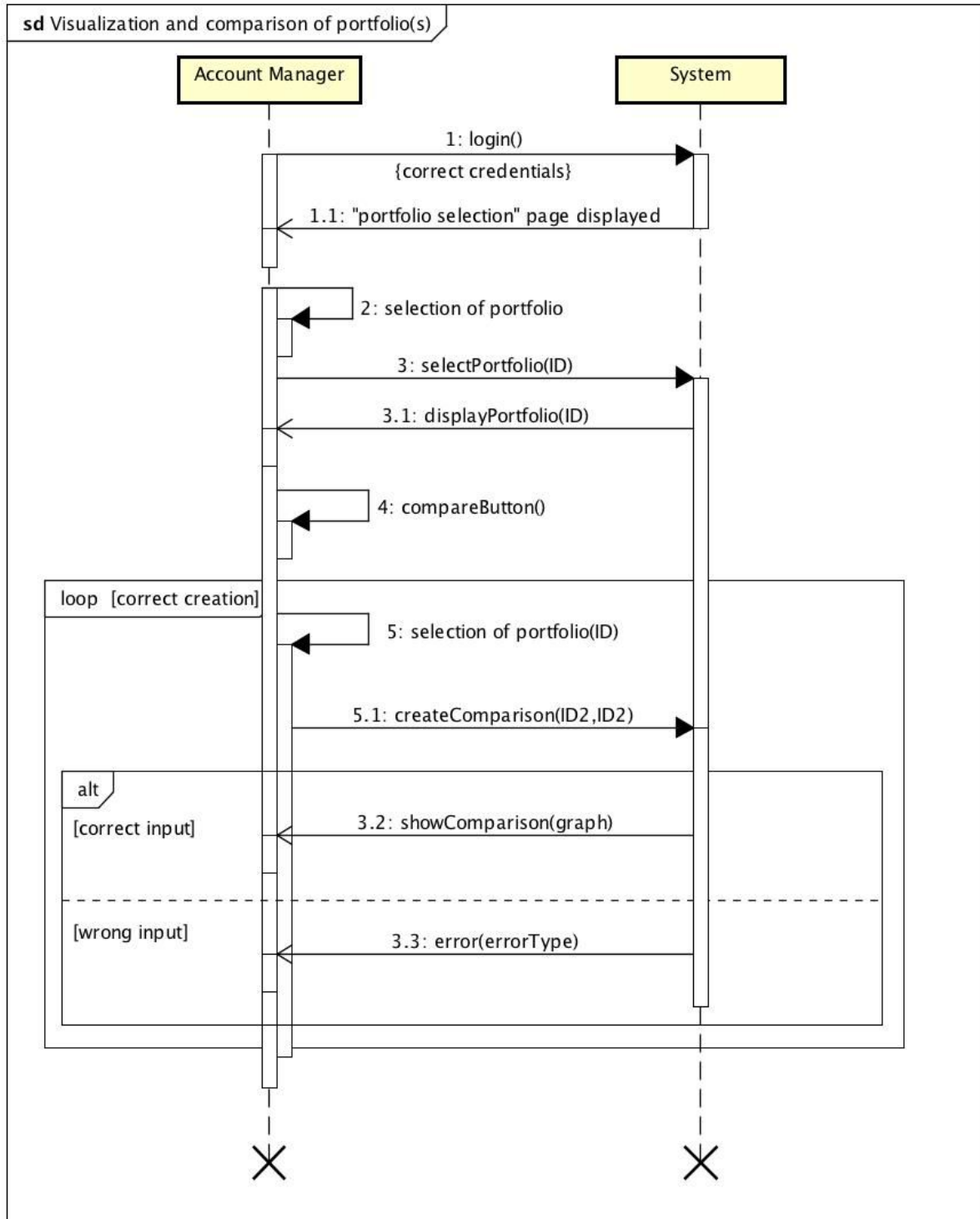
Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

### 5.3. Account manager visualization and comparison of portfolio(s)

<b>Name</b>	Account manager visualization and comparison of portfolio(s)
<b>Actor</b>	Account Manager, Admin
<b>Entry conditions</b>	<ul style="list-style-type: none"> <li>The user logged has Account Manager valid credentials</li> <li>Login session not expired</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>Selection of specific portfolio</li> <li>Page with selected portfolio is created</li> <li>“Compare” button is pressed</li> <li>Selection of specific portfolio (the one to compare)</li> </ol>
<b>Exit conditions</b>	Page with comparison between two selected portfolios is shown
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>The comparison between the two portfolios is not possible and the page with comparison between two selected portfolios is not shown</li> <li>The user is not allowed to visualize a comparison between two portfolios because he/she has not enough rights (i.e. a customer account)</li> <li>User is disconnected due to session inactivity timeout, so he/she is logged out automatically</li> </ul>



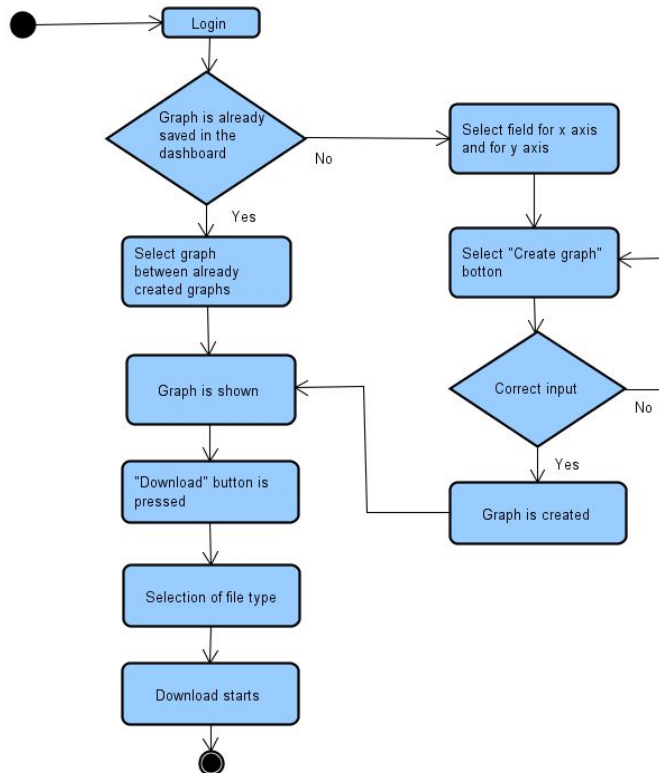
Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04



Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

#### 5.4. Download data

<b>Name</b>	Download data
<b>Actor</b>	User (Customer, Account manager, Admin)
<b>Entry conditions</b>	<ul style="list-style-type: none"> <li>• Logged in user</li> <li>• Login session not expired</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Login successful</li> <li>2. If graph is already created select it, if not create it</li> <li>3. Graph is shown</li> <li>4. "Download" button is pressed</li> <li>5. Selection of file type</li> <li>6. Download starts</li> </ol>
<b>Exit conditions</b>	Download complete
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The comparison between the two portfolios is not possible and the page with comparison between two selected portfolios is not shown</li> <li>• User is disconnected due to session inactivity timeout, so he/she is logged out automatically</li> </ul>



Visual Portfolio	Version: 1.0
Requirements Definition Document	Date: 2017-11-04

