

---

---

# <SALES\_COMMISSION MANAGEMENT APP>

## OVERALL REPORT

VERSION <2.0>

<Konstantinos Papadopoulos AM : 4761

Ilias Papathanasiou AM : 4765>



# TABLE OF CONTENTS

Introduction	4
Refactored Design	5
Use Cases	5
Architecture	17
Detailed Design	17
Addressed Issues	20
Classes Responsibilities and Collaborations (CRC CARDS)	21

## INTRODUCTION

In this project, our objective was to enhance the quality and maintainability of a sales and commission management application by addressing various coding issues. We applied several key refactorings to handle problems such as code duplication and the presence of underused or "lazy" classes. By implementing these refactorings, we improved the codebase's structure, making it more organized, efficient, and easier to understand. The result is a more reliable and maintainable application that provides a solid foundation for future enhancements and reduces the risk of potential bugs or errors related to code duplication and unutilized classes.

**\*\* Sometimes when we load a "test-case" file it seems that the program cannot load the information correctly, but after restarting the application it works as expected. \*\***

## REFACTORED DESIGN

### USE CASES

UC : Load sales representative info from a TXT File

<b>Use case ID</b>	UC1
<b>Actors</b>	System, user
<b>Pre conditions</b>	The application is launched.  The SalesRepInfoInputWindow is opened.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the «Εισαγωγή από TXT» button.</li><li>2. The system displays a File Chooser window.  2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without loading the sales representative's file.</li><li>3. The user navigates to the TXT file of his choosing and double-clicks it.</li></ol>
<b>Alternative flow 1</b>	If the file is not in TXT format the system displays an error dialogue box with the message : "The TXT file, you're trying to load has a problem with the Receipt format! Please check your file and try again."
<b>Alternative flow 2</b>	If the user has already loaded the specific TXT file, the system displays an error dialogue box with the message : "The selected Sales Representative's file is already loaded."
<b>Alternative flow 3</b>	If the user clicks on the "Cancel" button in the main window, the system closes the application.
<b>Post conditions</b>	The system has loaded the sales representative's file.

---

**UC : Load sales representative info from an XML File**

<b>Use case ID</b>	UC2
<b>Actors</b>	System, user
<b>Pre conditions</b>	The application is launched.  The SalesRepInfoInputWindow is opened.
<b>Main flow of events</b>	1. The use case begins when the user clicks on the «Εισαγωγή από XML» button. 2. The system displays a File Chooser window. 2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without loading the sales representative's file. 3. The user navigates to the XML file of his choosing and double-clicks it.
<b>Alternative flow 1</b>	If the file is not in XML format the system displays an error dialogue box with the message : "The XML file you're trying to load has an illegal XML argument. Please check your file and try again."
<b>Alternative flow 2</b>	If the user has already loaded a specific XML file, the system displays an error dialogue box with the message : "The selected Sales Representative's file is already loaded."
<b>Alternative flow 3</b>	If the user clicks on the "Cancel" button in the main window, the system closes the application.
<b>Post conditions</b>	The system has loaded the sales representative's file.

---

UC : Load sales representative info from an HTML File

<b>Use case ID</b>	UC3
<b>Actors</b>	System, user
<b>Pre conditions</b>	The application is launched.  The SalesRepInfoInputWindow is opened.
<b>Main flow of events</b>	1. The use case begins when the user clicks on the «Εισαγωγή από HTML» button. 2. The system displays a File Chooser window. 2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without loading the sales representative's file. 3. The user navigates to the HTML file of his choosing and double-clicks it.
<b>Alternative flow 1</b>	If the file is not in HTML format the system displays an error dialogue box with the message : "The HTML file you're trying to load has an illegal HTML argument. Please check your file and try again."
<b>Alternative flow 2</b>	If the user has already loaded a specific HTML file, the system displays an error dialogue box with the message : "The selected Sales Representative's file is already loaded."
<b>Alternative flow 3</b>	If the user clicks on the "Cancel" button in the main window, the system closes the application.
<b>Post conditions</b>	The system has loaded the sales representative's file.

#### UC : Select Sales Representative's Profile and View Information

<b>Use case ID</b>	UC4
<b>Actors</b>	System, user
<b>Pre conditions</b>	The sales representative's profile is loaded.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on a sales representative's name from the dropdown list to select it.</li><li>2. The user clicks the OK button.</li><li>3. The system opens a window with various options for the selected profile.</li></ol>
<b>Alternative flow 1</b>	If the user hasn't selected a profile and presses the OK button the system displays an error dialogue box with the message : "You haven't selected any Sales Representative's profile."
<b>Alternative flow 2</b>	If the user clicks on the "Cancel" button in the main window, the system closes the application.
<b>Post conditions</b>	The system has displayed the window for the sales representative's options.



UC : Select "Συνολική Αξία Πωλήσεων" to be Displayed

<b>Use case ID</b>	UC5
<b>Actors</b>	System, user
<b>Pre conditions</b>	<p>The application is running and the main window is displayed.</p> <p>The user has selected a specific sales representative profile.</p>
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the "Συνολική Αξία Πωλήσεων" check box.</li><li>2. The user clicks the "OK" button.</li><li>3. The system calculates the total sales amount for the loaded sales representative and creates a new window which displays the calculated value.</li></ol>
<b>Alternative flow 1</b>	If the user clicks on the "Cancel" button the system closes the window and returns the user to the previous one.
<b>Post conditions</b>	The system has calculated and displayed the amount of total sales for the selected sales representative.

UC : Select "Συνολικός Αριθμός Πωλήσεων" to be Displayed

<b>Use case ID</b>	UC6
<b>Actors</b>	System, user
<b>Pre conditions</b>	<p>The application is running and the main window is displayed.</p> <p>The user has selected a specific sales representative profile.</p>
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the "Συνολικός Αριθμός Πωλήσεων" check box.</li><li>2. The user clicks the "OK" button.</li><li>3. The system calculates the total items sold for the loaded sales representative and creates a new window which displays the calculated value.</li></ol>
<b>Alternative flow 1</b>	If the user clicks on the "Cancel" button the system closes the window and returns the user to the previous one.
<b>Post conditions</b>	The system has calculated and displayed the number of total items sold for the selected sales representative.

UC : Select "Πωλήσεις Συγκεκριμένης Κατηγορίας" to be Displayed

<b>Use case ID</b>	UC7
<b>Actors</b>	System, user
<b>Pre conditions</b>	<p>The application is running and the main window is displayed.</p> <p>The user has selected a specific sales representative profile.</p>
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the "Πωλήσεις Συγκεκριμένης Κατηγορίας" check box.</li><li>2. This action enables radio buttons for clothing categories.<ol style="list-style-type: none"><li>2.1 The categories are Shirts, Trousers, Coats, Skirts.</li></ol></li><li>3. The user selects one or more categories.</li><li>4. The user clicks the "OK" button.</li><li>5. The system calculates the total sales for each selected category of clothes, for the loaded sales representative and creates a new window which displays the calculated values.</li></ol>
<b>Alternative flow 1</b>	<p>If the user clicks on the "Cancel" button the system closes the window and returns the user to the previous one.</p>
<b>Post conditions</b>	<p>The system has calculated and displayed the amount of sales for each selected category of clothes, for the selected sales representative.</p>

UC : Select "Εμφάνιση Προμήθειας" to be Displayed

<b>Use case ID</b>	UC8
<b>Actors</b>	System, user
<b>Pre conditions</b>	The application is running and the main window is displayed.  The user has selected a specific sales representative profile.
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the " Εμφάνιση Προμήθειας" check box.</li><li>2. The user clicks the "OK" button.</li><li>3. The system calculates the total commission for all sales made by the loaded sales representative and creates a new window which displays the calculated value.</li></ol>
<b>Alternative flow 1</b>	If the user clicks on the "Cancel" button the system closes the window and returns the user to the previous one.
<b>Post conditions</b>	The system has calculated and displayed the amount of commission made by the selected sales representative.

## UC : Select "Προσθήκη Νέας Απόδειξης"

<b>Use case ID</b>	UC9
<b>Actors</b>	System, user
<b>Pre conditions</b>	<p>The application is running and the main window is displayed.</p> <p>The user has selected a specific sales representative profile.</p>
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case begins when the user clicks on the " Προσθήκη Νέας Απόδειξης" button.</li> <li>2. The system shows the hidden fields corresponding to receipt information. <ol style="list-style-type: none"> <li>2.1 The Receipt Information are ReceiptID, Date, Kind, Sales, Items, Company, Country, City, Street, Street Number.</li> </ol> </li> <li>3. The user fills out all necessary Receipt information.</li> <li>4. The user clicks the "OK" button.</li> <li>5. The system creates a new Receipt item, appends it to the sales representative's file and increments the "Αποδείξεις που προστέθηκαν" counter.</li> </ol>
<b>Alternative flow 1</b>	If the user clicks on the "Cancel" button the system closes the window and returns the user to the previous one.
<b>Alternative flow 2</b>	<ol style="list-style-type: none"> <li>1. The system checks the entered Receipt info. <ol style="list-style-type: none"> <li>1.1 If all Receipt fields are empty, the system displays an error dialogue box with the message : "All new Receipt fields are empty!"</li> <li>1.2 If one or more of the fields ,but not all of them are not filled the system displays an error dialogue box with the message : "Error: The receipt you're trying to add contains invalid or missing values. Please check the</li> </ol> </li> </ol>

	entered information."
<b>Post conditions</b>	The system has created and added the new Receipt object in the selected sales representative's file.

#### UC : Export Selected Data to TXT File

<b>Use case ID</b>	UC10
<b>Actors</b>	System, user
<b>Pre conditions</b>	-
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case begins when the user clicks on the "Εξαγωγή σε TXT" button.</li> <li>2. The system displays a File Chooser window. <ol style="list-style-type: none"> <li>2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without creating the Report.</li> </ol> </li> <li>3. The user navigates to the directory he wants to save the report and creates a name for the report.</li> </ol>
<b>Post conditions</b>	The system has created a new Report TXT file in the selected directory with the chosen name.

---

**UC : Export Selected Data to XML File**

<b>Use case ID</b>	UC11
<b>Actors</b>	System, user
<b>Pre conditions</b>	-
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>3. The use case begins when the user clicks on the "Εξαγωγή σε XML" button.</li><li>4. The system displays a File Chooser window.<ol style="list-style-type: none"><li>2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without creating the Report.</li></ol></li><li>3. The user navigates to the directory he wants to save the report and creates a name for the report.</li></ol>
<b>Post conditions</b>	The system has created a new Report XML file in the selected directory with the chosen name.

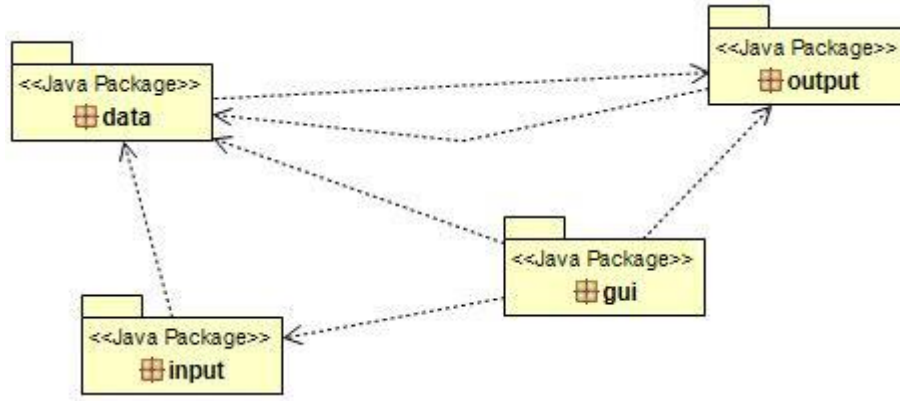
---

#### UC: Export Selected Data to HTML File

<b>Use case ID</b>	UC12
<b>Actors</b>	System, user
<b>Pre conditions</b>	-
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The use case begins when the user clicks on the "Εξαγωγή σε HTML" button.</li><li>2. The system displays a File Chooser window.<ol style="list-style-type: none"><li>2.1 If the user clicks on the "Cancel" button in the File Chooser window, the system closes the File Chooser window without creating the Report.</li></ol></li><li>3. The user navigates to the directory he wants to save the report and creates a name for the report.</li></ol>
<b>Post conditions</b>	The system has created a new Report HTML file in the selected directory with the chosen name.

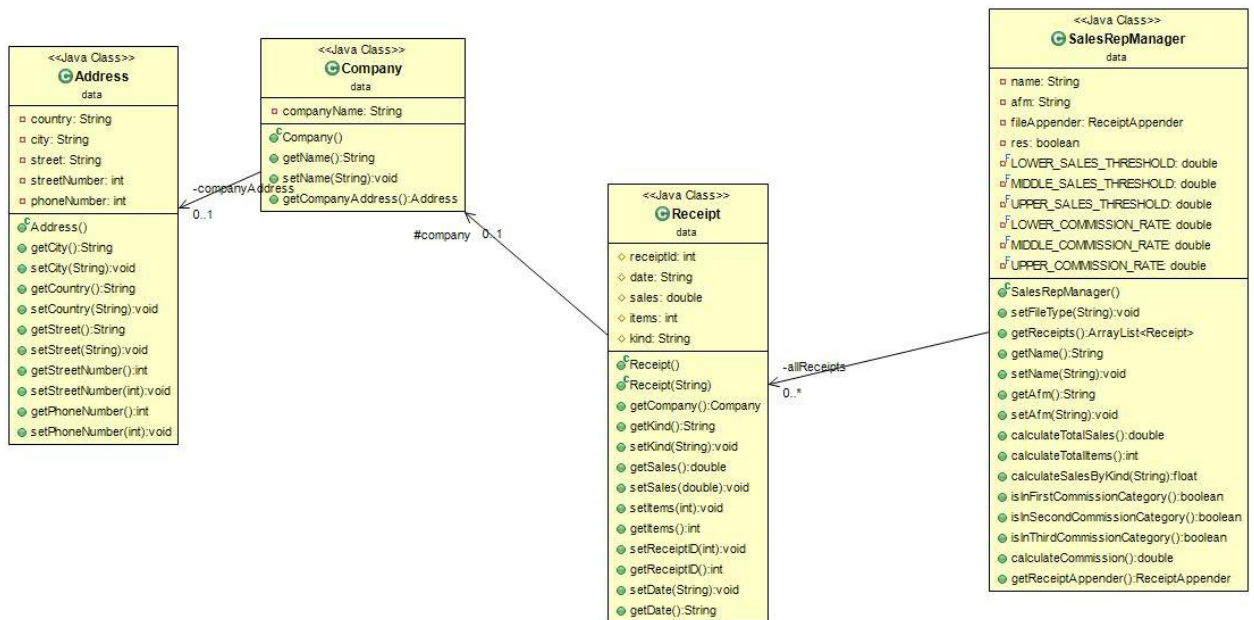


## ARCHITECTURE

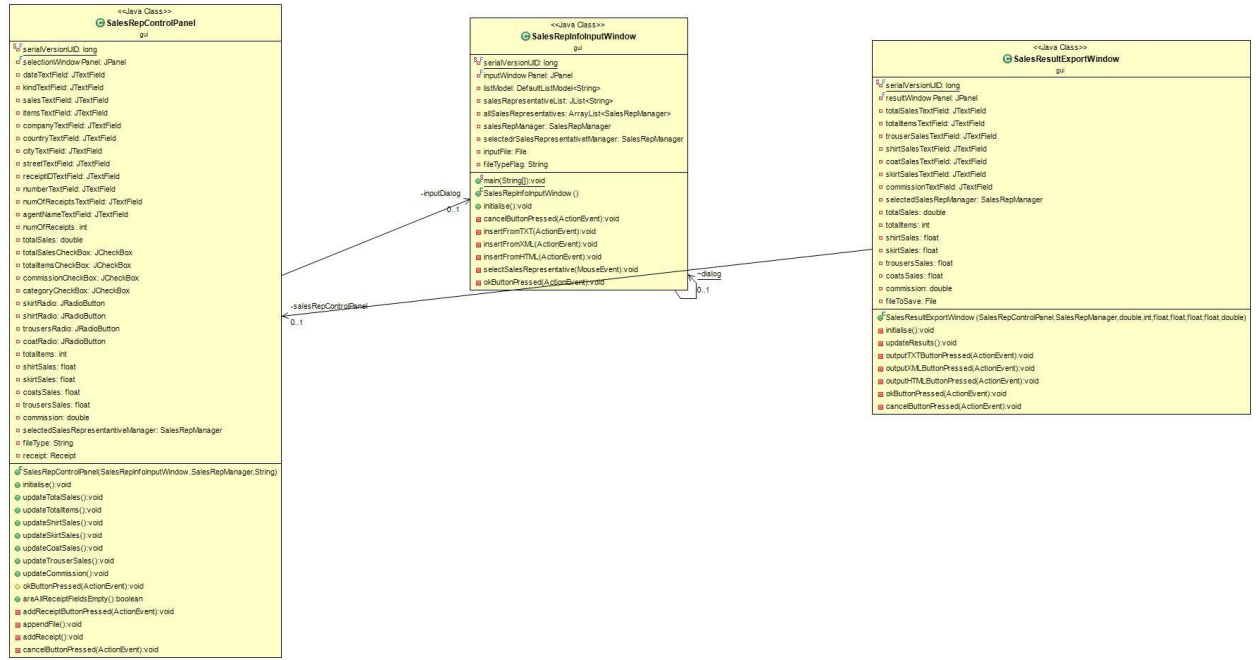


## DETAILED DESIGN

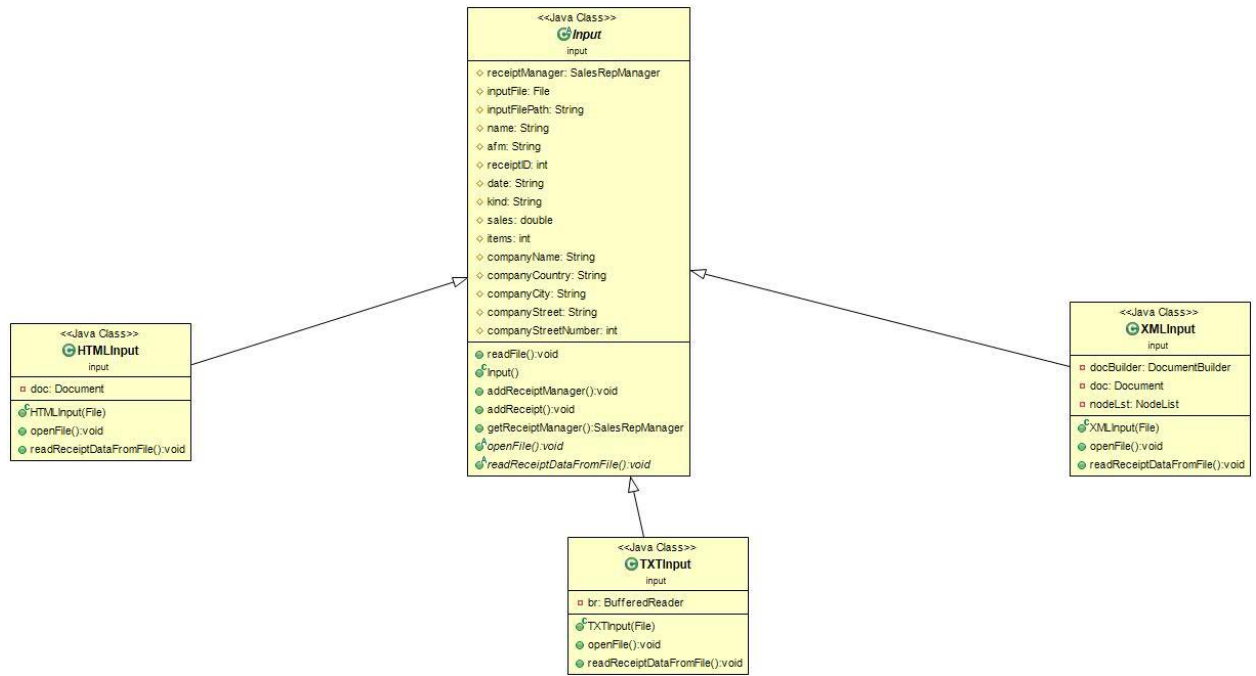
### Package Data :



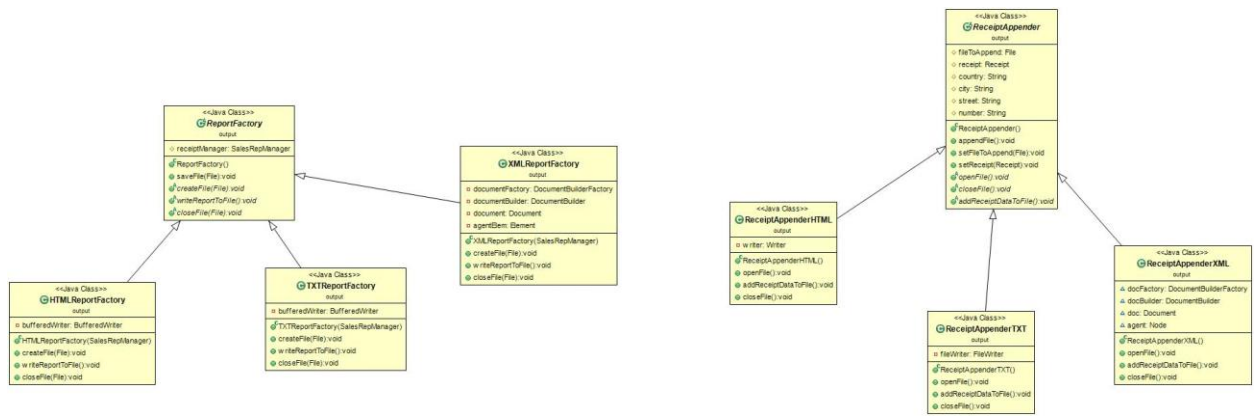
■ Package GUI :



■ Package Input :



## Package Output :



## OLD DESIGN ADDRESSED ISSUES

- **Lazy Classes** : The unnecessary complexity was handled by removing the classes and passing their responsibilities to the “Kind” field of the Receipt class.
- **Names** : Some classes names are falsely depicting the role of each class. Changing the names results in a better understanding of their responsibilities.
- **Duplicate Code** : To handle the Duplicate Code problem, we refactored the code using Parameterize Method, Form Template Method and Replace magic number with symbolic constant.
- **Primitive Obsession** : To handle the Primitive Obsession problem, we refactored the code using Replace Data Value with Object.
- **Wrong Package** : Based on ReceiptAppender and its implementation’s responsibilities we concluded that they belong in the Output Package.
- **Unnecessary Complexity in TXTInput** : Since the receipt’s file field order is predetermined, we simplified the algorithm by removing the redundant checks.
- **GUI Improvements** : We modified the GUI to permit users to select a custom file for saving a report, instead of relying on a predetermined file name. We addressed the complex conditional logic inside “SalesRepControlPanel” by using Consolidate Conditional. We addressed the chained method calls in appendFile( ) of “SalesRepControlPanel” by replacing them with a single setter method that passes the newly created Receipt object (which includes all those fields) to the FileAppender class. Lastly, we handled the complex conditional logic inside okButtonPressed( ) method by using Extract Method.

## CLASSES RESPONSIBILITIES AND COLLABORATIONS (CRC CARDS)

<b>Class Name: Address</b>	
<b>Responsibilities</b>	<b>Collaborations</b>
<ul style="list-style-type: none"> <li>○ <u>Store information about a physical address (country, city, street, street number, phone number).</u></li> <li>○ <u>Provide methods to access and modify individual address components.</u></li> </ul>	

Class Name: Company	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Hold information about a company including its name and address.</u></li> <li>○ <u>Provide methods to access and modify the company's name.</u></li> <li>○ <u>Manage the company's address through the Address class instance.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Address: The Company class collaborates with the Address class to store and manage the company's address details.</u></li> </ul>

Class Name: Receipt	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Store information about a sales representative's receipt, including its ID, date, sales amount, items count, associated company, and kind of receipt.</u></li> <li>○ <u>Provide methods to access and modify receipt attributes.</u></li> <li>○ <u>Create and manage the company associated with the receipt through the Company class instance.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Company: The Receipt class collaborates with the Company class to manage and associate a company with the receipt.</u></li> </ul>

Class Name: SalesRepManager	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Manage information about a sales representative, including their name (name) and tax identification number (afm).</u></li> <li>○ <u>Maintain a list of receipts (allReceipts) associated with the sales representative.</u></li> <li>○ <u>Calculate various statistics related to receipts, such as total sales, total items sold, sales by a specific kind, and commission based on sales thresholds.</u></li> <li>○ <u>Control the type of file for appending receipt details (fileAppender) using different ReceiptAppender implementations.</u></li> <li>○ <u>Provide methods to access and modify sales representative information and receipt data.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Receipt: The SalesRepManager class collaborates with the Receipt class to manage and analyze receipt data.</u></li> <li>▪ <u>ReceiptAppender, ReceiptAppenderTXT, ReceiptAppenderXML, ReceiptAppenderHTML: Collaborates with the SalesRepManager to append receipt details to different file types (TXT, XML or HTML).</u></li> </ul>



Class Name: SalesRepControlPanel	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Handles user inputs through various components like checkboxes, radio buttons, and text fields.</u></li> <li>○ <u>Verifies and validates the entered receipt details before processing.</u></li> <li>○ <u>Adds a new receipt with provided details to the sales representative's records.</u></li> <li>○ <u>Utilizes ReceiptAppender to append receipt data to files for record-keeping.</u></li> <li>○ <u>Calculates and manages various sales statistics, including total sales, total items, and sales per category.</u></li> <li>○ <u>Manages the display of the dialog window and user interface elements.</u></li> <li>○ <u>Proceed to the Sales</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>SalesRepManager: Interacts with the SalesRepManager object to perform operations like calculating sales statistics and managing receipts.</u></li> <li>▪ <u>ReceiptAppender: Collaborates with ReceiptAppender to append receipt data to file storage.</u></li> <li>▪ <u>Swing Components: Utilizes Swing components such as checkboxes, text fields, buttons, etc., for user interaction and data display within the dialog window.</u></li> <li>▪ <u>Collaborates with the SalesRepExportWindow to switch to the control panel for the selected sales representative.</u></li> <li>▪ <u>Collaborates with the SalesRepInfoInputWindow in order to load SalesRepManager's info.</u></li> </ul>

<u>Representative Export Window</u> <u>upon pressing the OK button.</u>	
--	--

Class Name: SalesRepInfoInputWindow	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Initialize the sales representative input window interface.</u></li> <li>○ <u>Manage input from TXT, XML, and HTML files for sales representatives.</u></li> <li>○ <u>Display a list of sales representatives and allow selection.</u></li> <li>○ <u>Handle the OK and Cancel button functionalities.</u></li> <li>○ <u>Identify and handle duplicated sales representative profiles.</u></li> <li>○ <u>Manage selection of a sales representative upon clicking the list.</u></li> <li>○ <u>Proceed to the Sales Representative Control Panel upon pressing the OK button.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with SalesRepManager to handle sales representative data.</u></li> <li>▪ <u>Interacts with TXTInput, XMLInput, and HTMLInput for reading files.</u></li> <li>▪ <u>Utilizes Swing components like buttons, labels, and lists for the graphical interface.</u></li> <li>▪ <u>Collaborates with SalesRepControlPanel to switch to the control panel for the selected sales representative.</u></li> </ul>

Class Name: SalesResultExportWindow	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Display a window to export sales result reports in TXT, XML, or HTML format.</u></li> <li>○ <u>Provide fields to show various sales-related data (total sales, total items, sales of different categories of items, commission).</u></li> <li>○ <u>Enable the user to save sales reports in TXT, XML, or HTML formats.</u></li> <li>○ <u>Handle OK and Cancel button functionalities.</u></li> <li>○ <u>Update displayed results based on received sales data.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Interacts with SalesRepControlPanel for control flow management.</u></li> <li>▪ <u>Collaborates with SalesRepManager to access and manage sales-related data.</u></li> <li>▪ <u>Utilizes Swing components for the graphical user interface, including buttons, labels, and text fields.</u></li> <li>▪ <u>Collaborates with TXTReportFactory, XMLReportFactory, and HTMLReportFactory to create and save sales reports in respective formats.</u></li> </ul>

Class Name: Input	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Abstract class serving as a blueprint for file input handling.</u></li> <li>○ <u>Contains attributes to manage receipt data including sales, items, company information, etc.</u></li> <li>○ <u>Provides methods to read a file and extract receipt data.</u></li> <li>○ <u>Abstract methods openFile() and readReceiptDataFromFile() to be implemented by subclasses.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with SalesRepManager and Receipt classes to manage and store receipt-related data.</u></li> <li>▪ <u>Interacts with File class to manage file-related operations.</u></li> </ul>

Class Name: TXTInput	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Handle File Input (Inherited):</u> <u>Extends the abstract class</u> <u>Input and specializes in</u> <u>handling TXT files.</u></li> <li>○ <u>Implement the openFile</u> <u>method to initialize a</u> <u>BufferedReader for reading</u> <u>from the TXT file.</u></li> <li>○ <u>Implement the</u> <u>readReceiptDataFromFile</u> <u>method to read and parse</u> <u>receipt data from the TXT file,</u> <u>populating the attributes of</u> <u>the Input class.</u></li> <li>○ <u>Invokes methods like</u> <u>addReceiptManager and</u> <u>addReceipt to manage and</u> <u>store receipt-related data.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with the</u> <u>SalesRepManager class to manage and</u> <u>store receipt-related data.</u></li> <li>▪ <u>Collaborates with the Receipt class to</u> <u>manage and store receipt-related</u> <u>data.</u></li> <li>▪ <u>Interacts with the File class to manage</u> <u>file-related operations.</u></li> </ul>

Class Name: XMLInput	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Extends the abstract class Input and specializes in handling XML files.</u></li> <li>○ <u>Implement the openFile method to initialize a DocumentBuilder for parsing files using the provided File.</u></li> <li>○ <u>Implement the readReceiptDataFromFile method to read and parse receipt data from the XML file, populating the attributes of the Input class.</u></li> <li>○ <u>Invokes methods like addReceiptManager and addReceipt to manage and store receipt-related data.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with the SalesRepManager class to manage and store receipt-related data.</u></li> <li>▪ <u>Collaborates with the Receipt class to manage and store receipt-related data.</u></li> <li>▪ <u>Utilizes the File class for file-related operations.</u></li> </ul>

Class Name: HTMLInput	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Extends the abstract class Input and specializes in handling HTML files.</u></li> <li>○ <u>Implement the openFile method to read the HTML file using Jsoup and parse it.</u></li> <li>○ <u>Implement the readReceiptDataFromFile method to extract and parse receipt data from the HTML file, populating the attributes of the Input class.</u></li> <li>○ <u>Invokes methods like addReceiptManager and addReceipt to manage and store receipt-related data.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with the SalesRepManager class to manage and store receipt-related data.</u></li> <li>▪ <u>Collaborates with the Receipt class to manage and store receipt-related data.</u></li> <li>▪ <u>Utilizes the File class for file-related operations.</u></li> <li>▪ <u>Uses the Jsoup library for HTML parsing.</u></li> </ul>



Class Name: ReportFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Contains an instance to a SalesRepManager to interact with and retrieve data for report generation.</u></li> <li>○ <u>Provides a high-level method saveFile that handles the file saving process.</u></li> <li>○ <u>Defines abstract method createFile to create the file.</u></li> <li>○ <u>Defines writeReportToFile to write the report content to the file.</u></li> <li>○ <u>Defines abstract methods (createFile, writeReportToFile, closeFile) for handling creating and closing the file along with writing the report to be implemented by its subclasses</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with the SalesRepManager class to retrieve data for generating reports.</u></li> <li>▪ <u>Utilizes the File class for file-related operations.</u></li> </ul>

Class Name: HTMLReportFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Create HTML sales reports based on sales data.</u></li> <li>○ <u>Handle exceptions related to file operations and report writing.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with SalesRepManager to obtain necessary sales-related data.</u></li> <li>▪ <u>Utilizes JOptionPane for displaying error messages in case of exceptions.</u></li> <li>▪ <u>Works with the File class to manage file operations.</u></li> </ul>

Class Name: TXTReportFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Implement methods to create a TXT file and write the report content.</u></li> <li>○ <u>Handles exceptions related to file operations and report writing.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with SalesRepManager to obtain necessary sales-related data.</u></li> <li>▪ <u>Utilizes JOptionPane for displaying error messages in case of exceptions.</u></li> <li>▪ <u>Works with the File class to manage file operations.</u></li> </ul>

Class Name: XMLReportFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Implements methods to create an XML file and write the report content.</u></li> <li>○ <u>Handles exceptions related to XML document creation and report writing.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Collaborates with SalesRepManager to obtain necessary sales-related data.</u></li> <li>▪ <u>Utilizes various XML-related classes (DocumentBuilderFactory, DocumentBuilder, Document, Element) for XML document creation.</u></li> <li>▪ <u>Works with the TransformerFactory and Transformer classes for transforming the XML document into a file.</u></li> </ul>

Class Name: ReceiptAppender	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Abstract class serving as a blueprint for appending data to a file.</u></li> <li>○ <u>Provides the method appendFile() to execute the file appending process.</u></li> <li>○ <u>Defines abstract methods to open, close, and add receipt data to the file.</u></li> <li>○ <u>Handles exceptions related to file operations.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Interacts with the File class to represent the file to which data will be appended.</u></li> <li>▪ <u>Collaborates with the Receipt class to manage and retrieve receipt data.</u></li> </ul>

Class Name: ReceiptAppenderTXT	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Concrete implementation of ReceiptAppender for appending data to a text (TXT) file.</u></li> <li>○ <u>Manages the FileWriter to write data to the text file.</u></li> <li>○ <u>Overrides abstract methods to open, close, and add receipt data from/to the text file.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Interacts with the FileWriter class to write data to the text file.</u></li> </ul>

Class Name: ReceiptAppenderXML	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Concrete implementation of ReceiptAppender for appending data to an XML file.</u></li> <li>○ <u>Manages the XML document creation, parsing, and writing of receipt data to the XML file.</u></li> <li>○ <u>Overrides abstract methods to open, close, and add receipt data from/to the XML file.</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Interacts with XML-related classes (DocumentBuilderFactory, DocumentBuilder, Document, Element, Node) to manage XML document creation and manipulation.</u></li> <li>▪ <u>Utilizes the TransformerFactory and Transformer classes to transform the XML document into a file.</u></li> </ul>

Class Name: ReceiptAppenderHTML	
Responsibilities	Collaborations
<ul style="list-style-type: none"> <li>○ <u>Concrete implementation of ReceiptAppender for appending data to an HTML file.</u></li> <li>○ <u>Manages the Writer to write data to the HTML file.</u></li> <li>○ <u>Overrides abstract methods to open, close, and add receipt data from/to the HTML file</u></li> </ul>	<ul style="list-style-type: none"> <li>▪ <u>Interacts with the Writer class to write data to the HTML file.</u></li> <li>▪ <u>Interacts with the I/O classes (BufferedWriter, FileWriter, Writer) to populate the report with the required data.</u></li> </ul>