

Διαχείριση Πωλήσεων και Προμηθειών

VERSION <1.0>

Χριστίνα-Ολυμπία Σολδάτου, 4001

Τσαπικούνη Γεωργία , 4304

TABLE OF CONTENTS

Introduction	3
Refactored Design	3
Use Cases	3
Architecture	8
Detailed Design	8
Classes Responsibilities and Collaborations (CRC CARDS)	11

INTRODUCTION

Η εργασία αποσκοπεί στην ανάπτυξη μιας εφαρμογής διαχείρισης πωλήσεων και προμηθειών για αντιπροσώπους μιας βιοτεχνίας ενδυμάτων. Η γραφική διεπαφή προσφέρει ευκολία στη χρήση, ενώ η επεκτασιμότητα εξασφαλίζει την ευέλικτη αποθήκευση και φόρτωση πληροφοριών σε διάφορες μορφές, πέραν του TXT και του XML. Συνολικά, ο στόχος είναι η δημιουργία μιας αποδοτικής και ευέλικτης εφαρμογής για τη διαχείριση των εργασιών πωλήσεων και προμηθειών.

REFACTORED DESIGN

USE CASES

ΕισήγαγεΑρχείοΑντιπροσώπου

Use case ID	UC1
Actors	Ο χρήστης
Pre conditions	Ο χρήστης βρίσκεται στο homepage της εφαρμογής.
Main flow of events	<ol style="list-style-type: none">1. Το use case ξεκινά όταν ο χρήστης επιλέξει είδος αρχείου για φόρτωση.2. Ο χρήστης έχει τις ακόλουθες επιλογές:<ol style="list-style-type: none">2.1 Είτε επιλέγει εισαγωγή από txt .2.2 Είτε επιλέγει εισαγωγή από xml.2.3 Είτε επιλέγει εισαγωγή από html.3. Ο χρήστης πατάει το κουμπί εισαγωγή από txt ή xml ή html.4. Το σύστημα εμφανίζει το παράθυρο περιήγησης για την εύρεση του αρχείου μέσα στο test_input_files .5. Ο χρήστης επιλέγει να φορτώσει το txt ή xml ή html αρχείο.

	6. Το σύστημα εμφανίζει τον επιλεγθέντα πωλητή στο homepage. 7. Ο χρήστης επιλέγει τον πωλητή. 8. Ο χρήστης πατάει το κουμπί ok.
Alternative flow 1	1. Εάν ο χρήστης επιθυμεί να κλείσει την εφαρμογή μπορεί να πατήσει το κουμπί cancel . 2. Εάν ο χρήστης δεν επιλέξει πωλητή το σύστημα εμφανίζει μήνυμα « Please choose a salesman.» 3. Εάν ο χρήστης επιλέξει να ανοίξει ένα αρχείο το οποίο έχει ήδη επιλεγθεί το σύστημα εμφανίζει μήνυμα «The salesman is already on the list.» 4. Εάν ο χρήστης δεν επιλέξει κανένα αρχείο το σύστημα εμφανίζει μήνυμα «no file selected. Please choose a file».
Post conditions	Το σύστημα έχει καταχωρήσει το αντιπρόσωπο που επιλέχθηκε .

Επέλεξε Πληροφορίες Αντιπροσώπου

Use case ID	UC2
Actors	Ο χρήστης
Pre conditions	Ο χρήστης έχει επιλέξει πωλητή.
Main flow of events	1. Το σύστημα εμφανίζει μία φόρμα με πληροφορίες του εκάστοτε πωλητή προς εμφάνιση. 2. Ο χρήστης επιλέγει τις πληροφορίες που επιθυμεί να εμφανιστούν στην αναφορά του συγκεκριμένου πωλητή . 3. Ο χρήστης πατάει το κουμπί ok.

Alternative flow 1	1. Εάν ο χρήστης επιθυμεί επιτρέψει στο homepage μπορεί να πατήσει το κουμπί cancel.
Post conditions	Το σύστημα έχει καταχωρήσει τις επιλεγμένες πληροφορίες ενός συγκεκριμένου πωλητή.

Εξήγαγε Πληροφορίες Αντιπροσώπου

Use case ID	UC3
Actors	Ο χρήστης
Pre conditions	Ο χρήστης έχει επιλέξει τις πληροφορίες προς εμφάνιση ενός πωλητή.
Main flow of events	<ol style="list-style-type: none"> 1. Το σύστημα εμφανίζει τα δεδομένα των πληροφοριών που έχει επιλέξει ο χρήστης. 2. Ο χρήστης έχει δύο επιλογές : <ol style="list-style-type: none"> 2.1 Είτε επιλέγει να κάνει εξαγωγή σε txt. 2.2 Είτε επιλέγει να κάνει εξαγωγή σε xml. 2.3 Είτε επιλέγει να κάνει εξαγωγή σε html. 3. Ο χρήστης πατάει το κουμπί εξαγωγή σε txt ή xml ή html. 4. Ο χρήστης επιλέγει που θα καταχωρήσει το αρχείο. 5. Ο χρήστης πατάει το κουμπί save.

	<p>6. Το σύστημα εμφανίζει μήνυμα «Report saved succesfully ».</p> <p>7. Ο χρήστης πατάει το κουμπί ok.</p>
Alternative flow 1	<p>1. Εάν ο χρήστης πατάει θέλει να επιστρέφει στη φόρμα επιλογής των πληροφοριών για τον επιλεγμένο πωλητή πατάει το κουμπί cancel.</p>
Post conditions	<p>Το σύστημα έχει καταχωρήσει και εκτυπώσει τις επιλεγμένες πληροφορίες ενός συγκεκριμένου πωλητή.</p>

ΠρόσθεσεΑπόδειξη

Use case ID	UC4
Actors	Ο χρήστης
Pre conditions	Ο χρήστης έχει επιλέξει πληροφορίες προς εμφάνιση ενός συγκεκριμένου πωλητή.
Main flow of events	<ol style="list-style-type: none"> 1. Το use case ξεκινά όταν ο χρήστης πατάει το κουμπί Προσθήκη νέας απόδειξης . 2. Ο χρήστης συμπληρώνει τα πεδία που είναι διαθέσιμα. 3. Ο χρήστης πατάει το κουμπί προσθήκη. 4. Το σύστημα εμφανίζει το μήνυμα «Receipt added succesfully» 5. Ο χρήστης πατάει το κουμπί ok.
Alternative flow 1	<ol style="list-style-type: none"> 1. Εάν ο χρήστης προσθέσει λάθος ReceiptID το τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid ReceiptID. Please enter a valid integer » 2. Εάν ο χρήστης προσθέσει λάθος το πεδίο Date τότε το σύστημα εμφανίζει μήνυμα λάθους «Invalid date format.Please use DD/MM/YYYY».

	<ol style="list-style-type: none"> 3. Εάν ο χρήστης δεν προσθέσει σωστά το πεδίο kind , τότε το σύστημα εμφανίζει μήνυμα λάθους « Invalid Kind. Please use Coats, Shirts, Skirts, Trousers». 4. Εάν ο χρήστης προσθέσει λάθος το πεδίο Sales τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid sales amount. Please enter a valid number ». 5. Εάν ο χρήστης προσθέσει λάθος το πεδίο Items τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid number of items. Please enter a valid integer ». 6. Εάν ο χρήστης προσθέσει λάθος το πεδίο Company τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid name. Please enter a valid string ». 7. Εάν ο χρήστης προσθέσει λάθος το πεδίο City ή Country ή Street τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid City or Country or Street. Please enter valid strings » 8. Εάν ο χρήστης προσθέσει λάθος το πεδίο Number τότε το σύστημα εμφανίζει μήνυμα λάθους «invalid street number. Please enter a valid integer». 9. Εάν ο χρήστης εν συμπληρώσει κάποιο πεδίο τότε το σύστημα του εμφανίζει μήνυμα «You must fill in all fields».
Post conditions	Το σύστημα έχει καταχωρήσει επιτυχώς την απόδειξη.

ARCHITECTURE

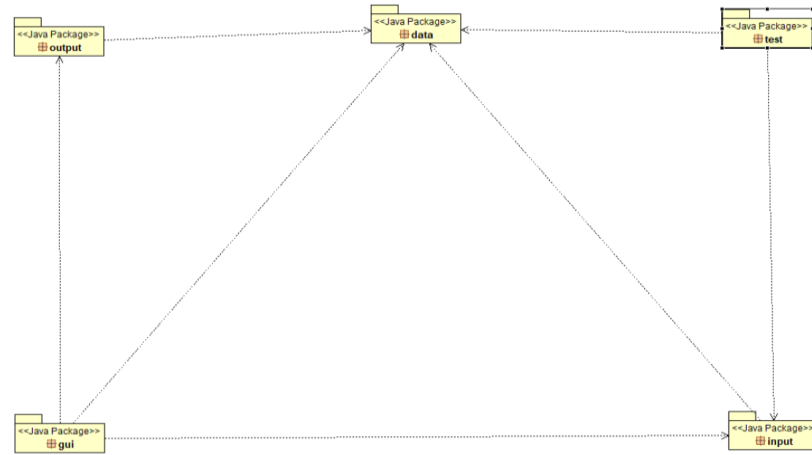


Figure 1

DETAILED DESIGN

✓ Package data

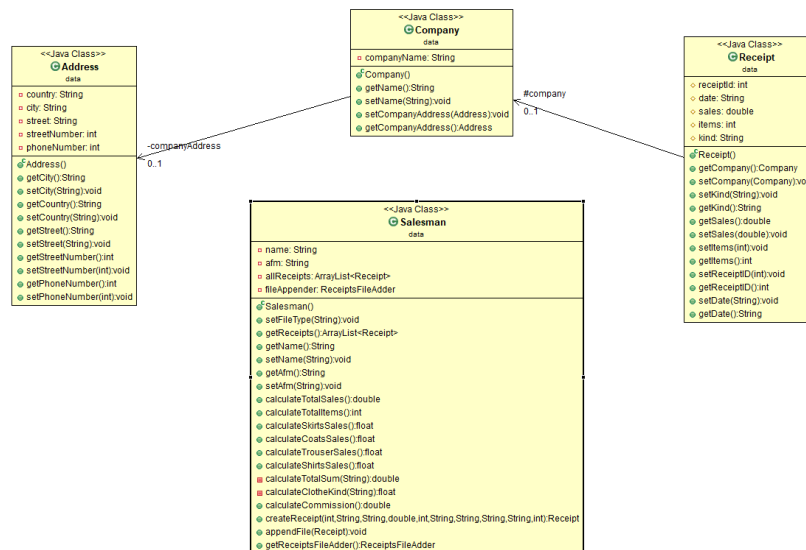


Figure 2

✓ **Package gui:**



Figure 3

✓ Package input:

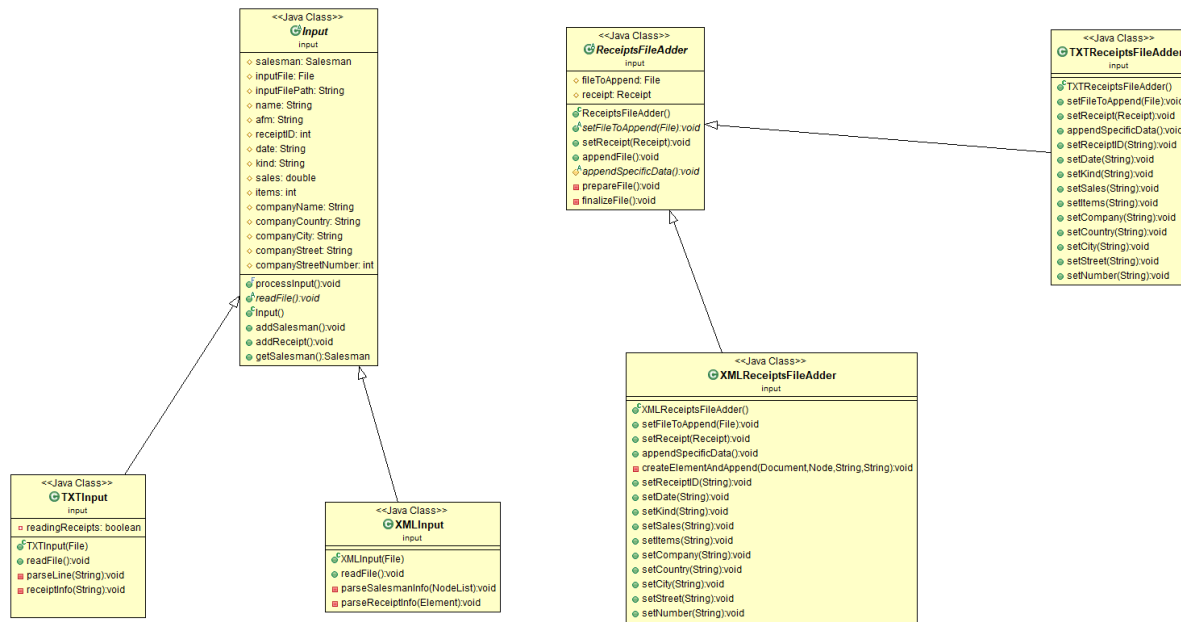


Figure 4

✓ **Package output:**

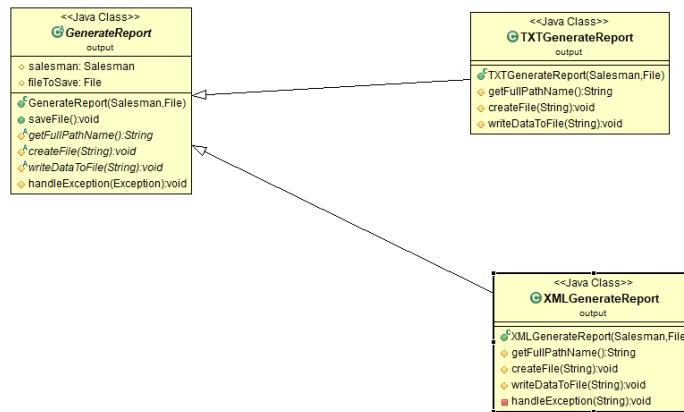


Figure 5

CLASSES RESPONSIBILITIES AND COLLABORATIONS (CRC CARDS)

Class Name: Salesman	
Responsibilities	Collaborations
<ul style="list-style-type: none"> Η κλάση <code>salesman</code> είναι υπεύθυνη για τη διαχείριση των πληροφοριών σχετικά με τις πωλήσεις ενός πωλητή, τον χειρισμό αρχείων καθώς και τον υπολογισμό προμηθειών. Αναλυτικότερα φαίνονται παρακάτω: 	<ul style="list-style-type: none"> <code>Receipt</code>: Αντιπροσωπεύει μεμονωμένες sales receipts. <code>ReceiptFileAdder</code>: Interface για προσθήκη αποδείξεων σε ένα αρχείο. <code>TXTRceiptsFileAdder</code>: Υλοποίηση του

<ul style="list-style-type: none"> • Προσάρτηση μίας απόδειξης σε ένα αρχείο(txt ή XML). • Αποθήκευση μίας λίστας από αποδείξεις που σχετίζονται με τον εκάστοτε πωλητή. • Δημιουργία μίας απόδειξης με καθορισμένα στοιχεία(ID, Date,kind,Sales, Items,Company details). • Υπολογισμός των πωλήσεων που αφορούν τα διάφορα είδη (coats, trousers, skirts, shirts). • Υπολογισμός των συνολικών πωλήσεων που πραγματοποίησε ο κάθε πωλητής. • Υπολογισμός του συνολικού αριθμού των ειδών που πωλήθηκαν από τον συγκεκριμένο πωλητή. 	<p>ReceiptsFileAdder για αρχεία TXT.</p> <ul style="list-style-type: none"> • XMLReceiptsFileAdder: Υλοποίηση του ReceiptsFileAdder για αρχεία XML. • Company: Έχει πληροφορίες για την εταιρεία που σχετίζεται με μία απόδειξη. • Address: Έχει πληροφορίες σχετικά με την διεύθυνση της εταιρείας .
--	--

Class Name: Address	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • Η κλάση είναι υπεύθυνη για να κρατάει πληροφορίες σχετικά με τη country, city, street και το street number 	<ul style="list-style-type: none"> • Αυτή η κλάση συνεργάζεται κυρίως με άλλες κλάσεις που μπορεί να

	χρειάζονται πληροφορίες διεύθυνσης, όπως η κλάση Company.
--	---

Class Name: Company	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • Η κλάση είναι υπεύθυνη για την εταιρεία με ότι στοιχεία αυτή περιλαμβάνει καθώς και για την διαχείριση στοιχείων της τοποθεσίας μέσω της κλάσης Address . Πιο αναλυτικά η κλάση αυτή είναι υπεύθυνη για τα εξής : • Διαχείριση πληροφοριών σχετικά με την εταιρία • Παροχή μεθόδων για την απόκτηση και τον καθορισμό της επωνυμίας της εταιρείας. • Παροχή μεθόδων για τη λήψη και τον 	<ul style="list-style-type: none"> • Address: Αντιπροσωπεύει τα στοιχεία διεύθυνσης που σχετίζονται με την εταιρεία.

καθορισμό της διεύθυνσης της εταιρείας.	
---	--

Class Name: Receipt	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ➤ Η κλάση είναι υπεύθυνη για την αναπαράσταση των αποδείξεων καθώς και την ενσωμάτωση πληροφοριών σχετικά με την κάθε συναλλαγή. Αναλυτικότερα οι αρμοδιότητες της είναι οι εξής: • Αντιπροσωπεύει μια μεμονωμένη απόδειξη πώλησης με πληροφορίες όπως ID, date, sales, items, company, και kind. • Παρέχει getters και setters για τις πληροφορίες σχετικά με την απόδειξη συμπεριλαμβάνοντας το ID, date, sales, items, company, and kind. 	<ul style="list-style-type: none"> • Company : Περιέχει πληροφορίες σχετικά με την εταιρεία που σχετίζεται με τη συγκεκριμένη απόδειξη.

Class Name: ImportSalesmanFileWindow	
Responsibilities	Collaborations
<p>Αυτή η κλάση είναι υπεύθυνη για τα εξής:</p> <ul style="list-style-type: none"> • Παροχή ενός παραθύρου γραφικής διεπαφής χρήστη (GUI) για την εισαγωγή πωλητών από αρχεία TXT ή XML. • Εμφάνιση λίστας των εισαγόμενων πωλητών. • Επιτρέπει στο χρήστη να επιλέξει έναν πωλητή από τη λίστα. • Να επικοινωνεί με κλάσεις εισόδου (TXTInput και XMLInput) για την ανάγνωση δεδομένων πωλητών από αρχεία. • Να χειρίζεται αλληλεπιδράσεις του χρήστη, όπως κλικ κουμπιών για την ενεργοποίηση συγκεκριμένων ενεργειών. • Άνοιγμα ενός νέου παραθύρου (SelesmanReportManagerWindow) για τη διαχείριση reports για τον επιλεγμένο πωλητή. 	<ul style="list-style-type: none"> • TXTInput: Διαβάζει δεδομένα πωλητή από αρχεία TXT. • XMLInput: Διαβάζει δεδομένα πωλητή από αρχεία XML. • Salesman: Αντιπροσωπεύει μεμονωμένους πωλητές και τα δεδομένα τους. • SelesmanReportManagerWindow: Διαχειρίζεται τη δημιουργία reports για επιλεγμένους πωλητές.

Class Name: ExtractSalesmanFileWindow	
Responsibilities	Collaborations
<p>Η συγκεκριμένη κλάση είναι υπεύθυνη για τα ακόλουθα:</p> <ul style="list-style-type: none"> • Παροχή ενός παραθύρου γραφικής διεπαφής χρήστη (GUI) για την εξαγωγή και αποθήκευση αναφορών για έναν επιλεγμένο πωλητή. • Εμφάνιση διαφόρων στατιστικών στοιχείων, όπως συνολικές πωλήσεις, σύνολο ειδών, πωλήσεις για διαφορετικούς τύπους ενδυμάτων και προμήθεια. • Επιτρέπει στο χρήστη να επιλέξει μεταξύ της δημιουργίας και της αποθήκευσης της αναφοράς σε μορφή TXT ή XML. • Επικοινωνεί με τις κλάσεις(TXTGenerateReport και XMLGenerateReport) για τη δημιουργία και αποθήκευση των αναφορών. • Ενημερώνει το GUI με τα στατιστικά αποτελέσματα του εκάστοτε πωλητή. 	<ul style="list-style-type: none"> • TXTGenerateReport: Δημιουργεί και αποθηκεύει αναφορές σε μορφή TXT. • XMLGenerateReport: Δημιουργεί και αποθηκεύει αναφορές σε μορφή XML. • SelesmanReportManagerWindow: Διαχειρίζεται το παράθυρο δημιουργίας και εμφάνισης αναφορών. • Salesman: Αντιπροσωπεύει τον εκάστοτε πωλητή για τον οποίο παράγεται η αναφορά.

Class Name: SelesmanReportManagerWindow	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ➤ Αυτή η κλάση είναι υπεύθυνη για τη διαχείριση γραφικής διεπαφής(GUI) των αλληλεπιδράσεων του χρήστη που σχετίζονται με τη δημιουργία και τη διαχείριση αναφορών για έναν πωλητή. Ακολουθεί μια ανάλυση των αρμοδιοτήτων της: • ρύθμιση των στοιχείων της γραφικής διεπαφής χρήστη, συμπεριλαμβανομένων των κουμπιών, των πλαισίων ελέγχου και των πεδίων κειμένου. • Εμφάνιση πεδίων που σχετίζονται με μεμονωμένες αποδείξεις, όπως ID,date , kind , sales etc. • Παροχή μιας επιλογής για την προσθήκη νέων αποδείξεων. • Επεξεργασία των εισερχομένων από τον χρήστη, υπολογισμός διαφόρων στατιστικών στοιχείων (total sales, total items, commission etc). 	<p>ImportSalesmanFileWindow: Συνεργάζεται με το παράθυρο για την εισαγωγή δεδομένων πωλητή.</p> <p>Salesman: Συνεργάζεται με την κλάση Salesman για την ανάκτηση και ενημέρωση πληροφοριών που σχετίζονται με τον πωλητή.</p> <p>Receipt : Συνεργάζεται με την κλάση Receipt για τη δημιουργία και διαχείριση αποδείξεων.</p> <p>JOptionPane: Χρησιμοποιεί την κλάση JOptionPane για την εμφάνιση μηνυμάτων σφάλματος</p> <p>ExtractSalesmanFileWindow : Συνεργάζεται με το παράθυρο για την εξαγωγή δεδομένων πωλητή.</p>

Class Name: Input	
Responsibilities	Collaborations
<p>Η συγκεκριμένη κλάση είναι υπεύθυνη για τις εξής αρμοδιότητες:</p> <ul style="list-style-type: none"> • Επεξεργασία δεδομένων εισόδου. • Ανάγνωση αρχείου (υλοποιείται ξεχωριστά για το txt ή xml). • Προσθήκη ενός απόδειξης για τον εκάστοτε πωλητή. 	<p>Salesman: Συνεργάζεται για την αποθήκευση πληροφοριών σχετικά με τον πωλητή που σχετίζεται με το Input.</p> <p>Receipt :Συνεργάζεται για την αποθήκευση πληροφοριών σχετικά με τις αποδείξεις που σχετίζονται με το Input.</p> <p>File: Συνεργάζεται για την ανάγνωση εισόδου από αρχείο.</p>

Class Name: TXTInput	
Responsibilities	Collaborations
<p>Η συγκεκριμένη κλάση είναι υπεύθυνη για τα ακόλουθα:</p> <ul style="list-style-type: none"> • Ανάγνωση εισόδου από ένα αρχείο TXT. • Ανάλυση γραμμών από το αρχείο TXT και εξαγωγή σχετικών πληροφοριών. • Ορισμός χαρακτηριστικών στην υπερκλάση (Input) με βάση τις αναλυμένες πληροφορίες. 	<p>Input : Επεκτείνει την κλάση Input και χρησιμοποιεί τις μεθόδους της.</p> <p>File : Αντιπροσωπεύει το αρχείο TXT από το οποίο διαβάζεται η είσοδος.</p> <p>BufferedReader : Διαβάζει γραμμές από το αρχείο TXT.</p> <p>FileNotFoundException : Χειρίζεται εξαιρέσεις που σχετίζονται με τη μη εύρεση του αρχείου.</p>

Class Name: XMLInput	
Responsibilities	Collaborations
<p>Η κλάση αυτή είναι υπεύθυνη για τα εξής:</p> <ul style="list-style-type: none"> • Ανάγνωση εισόδου από ένα αρχείο XML. • Ανάλυση στοιχείων XML και εξαγωγή σχετικών πληροφοριών. • Χειρισμός εξαιρέσεων που σχετίζονται με την ανάγνωση του αρχείου XML. • Ορισμός χαρακτηριστικών στην υπερκλάση (Input) με βάση τις αναλυμένες πληροφορίες. • Προσθήκη ενός πωλητή και μιας απόδειξης στο Input. 	<ul style="list-style-type: none"> • Input : Επεκτείνει την κλάση Input και χρησιμοποιεί τις μεθόδους της. • File : Αντιπροσωπεύει το αρχείο XML από το οποίο διαβάζεται η είσοδος. • JOptionPane : Εμφανίζει ένα παράθυρο διαλόγου για το χειρισμό εξαιρέσεων.

Class Name: ReceiptsFileAdder	
Responsibilities	Collaborations
<p>Η κλάση αυτή είναι υπεύθυνη για τα εξής:</p> <ul style="list-style-type: none"> • Προσθήκη συγκεκριμένων δεδομένων σε ένα αρχείο που περιέχει πληροφορίες παραλαβής. • Ορισμός του αρχείου στο οποίο θα προστεθούν τα κατάλληλα δεδομένα . • Ορισμός της απόδειξης της οποίας οι πληροφορίες θα προσαρτηθούν στο αρχείο. • Προετοιμασία του αρχείου πριν από την προσθήκη δεδομένων. • Οριστικοποίηση του αρχείο μετά την προσθήκη δεδομένων. 	<p>File: Αντιπροσωπεύει το αρχείο στο οποίο θα προσαρτηθούν τα δεδομένα.</p> <p>Receipt: Αντιπροσωπεύει την απόδειξη της οποίας οι πληροφορίες θα προσαρτηθούν στο αρχείο.</p>

Class Name: TXTReceiptsFileAdder	
Responsibilities	Collaborations
<p>Η κλάση αυτή είναι υπεύθυνη για τα ακόλουθα:</p> <ul style="list-style-type: none"> Αποθήκευση μιας αναφοράς στο αρχείο στο οποίο θα προσαρτηθούν τα δεδομένα της απόδειξης. Ορισμός διαφόρων χαρακτηριστικών του αντικειμένου Receipt, όπως ID, date, kind, sales, items, company, city , number Άνοιγμα του αρχείου και εγγραφή των δεδομένων με μορφοποιημένο τρόπο. 	<ul style="list-style-type: none"> FILE: Συνεργάζεται με την κλάση File για τη διαχείριση λειτουργιών που σχετίζονται με αρχεία. FileWriter: Χρησιμοποιεί την κλάση FileWriter για την εγγραφή δεδομένων στο καθορισμένο αρχείο. Receipt: Συνθέτει και χειρίζεται ένα αντικείμενο Receipt για τη συλλογή και αποθήκευση δεδομένων παραλαβής. IOException: Χειρίζεται IOException που μπορεί να προκύψουν κατά τη διάρκεια λειτουργιών αρχείου.

Class Name: XMLReceiptsFileAdder	
Responsibilities	Collaborations
<p>Η κλάση αυτή είναι υπεύθυνη για τα παρακάτω:</p> <ul style="list-style-type: none"> Αποθήκευση μιας αναφοράς στο αρχείο στο οποίο θα προσαρτηθούν τα δεδομένα της απόδειξης. Ορισμός διαφόρων χαρακτηριστικών του αντικειμένου Receipt, όπως ID, date, kind, sales, items, company, city , number 	<p>FILE: Συνεργάζεται με την κλάση File για τη διαχείριση λειτουργιών που σχετίζονται με αρχεία.</p> <p>FileWriter: Χρησιμοποιεί την κλάση FileWriter για την εγγραφή δεδομένων στο καθορισμένο αρχείο XML.</p>

Άνοιγμα του αρχείου και εγγραφή των δεδομένων με μορφοποιημένο τρόπο.	Receipt: Συνθέτει και επεξεργάζεται ένα αντικείμενο Receipt για τη συλλογή και αποθήκευση δεδομένων παραλαβής.
---	--

Class Name: GenerateReport	
Responsibilities	Collaborations
<p>Οι αρμοδιότητες για τις οποίες είναι υπεύθυνη η κλάση είναι οι εξής:</p> <ul style="list-style-type: none"> Αποθήκευση μίας αναφοράς στο αντικείμενο File όπου θα αποθηκευτεί το report. Αποθήκευση του αρχείου. Παροχή του πλήρους ονόματος διαδρομής όπου θα αποθηκευτεί το αρχείο του . Εγγραφή δεδομένων στο αρχείο αναφοράς που δημιουργήθηκε. Οι λεπτομέρειες υλοποίησης αφήνονται στις υποκλάσεις. 	<ul style="list-style-type: none"> Salesman: Παρέχει δεδομένα που σχετίζονται με έναν πωλητή τα οποία χρησιμοποιούνται στην έκθεση. FILE: Αντιπροσωπεύει το αρχείο προορισμού στο οποίο θα αποθηκευτεί η έκθεση.

Class Name: TXTGenerateReport	
Responsibilities	Collaborations
<ul style="list-style-type: none"> Υπεύθυνη για τη δημιουργία μιας έκθεσης κειμένου (TXT) με βάση τα δεδομένα ενός πωλητή. Δημιουργία αρχείου txt για την αποθήκευση του Report. Είναι υπεύθυνη για να γράφει τις πληροφορίες του πωλητή και τα υπολογισμένα δεδομένα πωλήσεων στο αρχείο TXT που δημιουργήθηκε. 	<ul style="list-style-type: none"> Salesman: Αναφέρεται στον πωλητή για τον οποίο δημιουργείται το report. FILE: Αναφέρεται στο αρχείο το οποίο θα αποθηκευτεί το Report.

Class Name: XMLGenerateReport	
Responsibilities	Collaborations
<ul style="list-style-type: none"> Υπεύθυνος για τη δημιουργία μιας έκθεσης XML με βάση τα δεδομένα ενός πωλητή. Δημιουργεί ένα νέο αρχείο XML για την έκθεση. Γράφει τις πληροφορίες του Salesman και τα υπολογισμένα δεδομένα πωλήσεων στο αρχείο XML που δημιουργήθηκε. 	<ul style="list-style-type: none"> Salesman: Αναφέρεται στο πωλητή για τον οποίο δημιουργείται το report. FILE: Αναφέρεται στο αρχείο στο οποίο θα αποθηκευτεί η αναφορά.

Additional Changes

Σε ορισμένα σημεία του κώδικα έχουμε κρίνει ότι θα πρέπει να γίνουν περαιτέρω αλλαγές οι οποίες δεν ζητήθηκαν στο πλαίσιο του Project μας. Παρακάτω φαίνονται αναλυτικά.

- Αλλαγές στην κλάση **SalesmanReportManagerWindow** του πακέτου **gui**. Ο κεντρικός μας στόχος ήταν να διορθώσουμε τα προβλήματα του application με την εισαγωγή των στοιχείων των αποδείξεων μας.
- a) Στην αρχική μέθοδο **addReceiptButtonPressed()** παρατηρήσαμε πως με την χρήση των **&&** θα πρέπει όλα τα πεδία να είναι κενά ώστε να εμφανίσει το μήνυμα λάθους. Αυτό έχει ως αποτέλεσμα οι αποδείξεις ακόμα και ένα κενό να έχουν να αποθηκεύονται.

```
private void addReceiptButtonPressed(ActionEvent evt) {
    if(receiptIDTextField.getText().isEmpty() && dateTextField.getText().isEmpty()
        && kindTextField.getText().isEmpty() && salesTextField.getText().isEmpty()
        && itemsTextField.getText().isEmpty() && companyTextField.getText().isEmpty()
        && countryTextField.getText().isEmpty() && cityTextField.getText().isEmpty()
        && streetTextField.getText().isEmpty() && numberTextField.getText().isEmpty()){
        JOptionPane.showMessageDialog(null,"You must fill in all fields");

    }

    else{

        addReceipt();
        appendFile();

    }
    receiptIDTextField.setText("");
    dateTextField.setText("");
    kindTextField.setText("");
    salesTextField.setText("");
    itemsTextField.setText("");
    companyTextField.setText("");
    countryTextField.setText("");
    cityTextField.setText("");
    streetTextField.setText("");
    numberTextField.setText("");
}
```

Figure 6

Παρακάτω φαίνεται η νέα προσέγγιση της μεθόδου:

```
private void addReceiptButtonPressed(ActionEvent evt) {
    if (areFieldsEmpty()) {
        JOptionPane.showMessageDialog(null, "You must fill in all fields.");
    } else {
        addReceipt();
        appendFile();
    }

    receiptIDTextField.setText("");
    dateTextField.setText("");
    kindTextField.setText("");
    salesTextField.setText("");
    itemsTextField.setText("");
    companyTextField.setText("");
    countryTextField.setText("");
    cityTextField.setText("");
    streetTextField.setText("");
    numberTextField.setText("");
}

private boolean areFieldsEmpty() {
    return isEmptyField(receiptIDTextField) || isEmptyField(dateTextField) || isEmptyField(kindTextField)
        || isEmptyField(salesTextField) || isEmptyField(itemsTextField) || isEmptyField(companyTextField)
        || isEmptyField(countryTextField) || isEmptyField(cityTextField) || isEmptyField(streetTextField)
        || isEmptyField(numberTextField);
}
```

Figure 7

Αλλάξαμε τα && με || και φτιάξαμε μία νέα συνάρτηση την οποία καλούμε μέσα στη δομή ελέγχου.

- Στην αρχική μεθοδο **addReceipt()** παρατηρήσαμε διάφορες ελλείψεις όσον αφορά τον έλεγχο για την ορθή εισαγωγή πεδίων στην νέα απόδειξη. Είναι απαραίτητο εάν ο χρήστης εισάγει κάποιο πεδίο λανθασμένα να γνωρίζει ποιο ακριβώς πεδίο ήταν αυτό και για ποιο λόγο το έκανε λάθος. Συνεπώς, προσθέσαμε μηνύματα λάθους για invalid fields ελέγχοντας κάθε πιθανή λάθος εισαγωγή του χρήστη.

```

private void addReceipt(){
    Receipt receipt = new Receipt();

    if(kindTextField.equals("Shirts"))
        receipt = new Shirt();
    else if (kindTextField.equals("Skirts"))
        receipt = new Skirt();
    else if (kindTextField.equals("Trousers"))
        receipt = new Trouser();
    else if (kindTextField.equals("Coats"))
        receipt = new Coat();
    try{
        receipt.setReceiptID(Integer.parseInt(receiptIDTextField.getText()));
        receipt.setDate(dateTextField.getText());
        receipt.setSales(Double.parseDouble(salesTextField.getText()));
        receipt.setItems(Integer.parseInt(itemsTextField.getText()));
        receipt.getCompany().setName(companyTextField.getText());
        receipt.getCompany().getCompanyAddress().setCountry(countryTextField.getText());
        receipt.getCompany().getCompanyAddress().setCity(cityTextField.getText());
        receipt.getCompany().getCompanyAddress().setStreet(streetTextField.getText());
        receipt.getCompany().getCompanyAddress().setStreetNumber(Integer.parseInt(numberTextField.getText()));
        selectedAgent.getReceipts().add(receipt);
        numOfReceipts++;
        numOfReceiptsTextField.setText(Integer.toString(numOfReceipts));
        JOptionPane.showMessageDialog(null,"Receipt added successfully");
    }catch (NumberFormatException e){
        JOptionPane.showMessageDialog(null,"Invalid input. Please check the entered values");
    }
}

```

Figure 8

Πιο συγκεκριμένα:

- a) Για το field **ReceiptID** θα πρέπει ο χρήστης να εισάγει έναν αριθμό. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```

String receiptIDInput = receiptIDTextField.getText().trim();
if (isValidInteger(receiptIDInput)) {
    receipt.setReceiptID(Integer.parseInt(receiptIDInput));
} else {
    JOptionPane.showMessageDialog(null, "Invalid Receipt ID. Please enter a valid integer.");
    return;
}

```

Figure 9

```

private boolean isValidInteger(String input) {
    try {
        Integer.parseInt(input);
        return true;
    } catch (NumberFormatException e) {
        return false;
    }
}

```

- b) Για το field **Date** θα πρέπει ο χρήστης να εισάγει μία ημερομηνία με την μορφή DD/MM/YYYY. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
String dateInput = dateTextField.getText();
if (isValidDateFormat(dateInput)) {
    receipt.setDate(dateInput);
} else {
    JOptionPane.showMessageDialog(null, "Invalid date format. Please use DD/MM/YYYY");
    return;
}
```

Figure 10

Εδώ βλέπουμε και το αναλυτικό έλεγχο της date:

```
private boolean isValidDateFormat(String date) {
    String dateFormatRegex = "^([0-9]|[12][0-9]|3[01])/([0-9]|1[0-2])/([\\d]{4})$";

    if (date.matches(dateFormatRegex)) {
        String[] dateParts = date.split("/");
        int day = Integer.parseInt(dateParts[0]);
        int month = Integer.parseInt(dateParts[1]);
        int year = Integer.parseInt(dateParts[2]);

        if (month >= 1 && month <= 12) {
            if ((month == 4 || month == 6 || month == 9 || month == 11) && (day >= 1 && day <= 30)) {
                return true;
            } else if ((month == 1 || month == 3 || month == 5 || month == 7 ||
                month == 8 || month == 10 || month == 12) && (day >= 1 && day <= 31)) {
                return true;
            } else if (month == 2) {
                if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
                    return day >= 1 && day <= 29;
                } else {
                    return day >= 1 && day <= 28;
                }
            }
        }
    }

    return false;
}
```

Figure 11

- c) Για το field **Kind** θα πρέπει ο χρήστης να εισάγει ένα **Coats, Shirts, Skirts, Trousers**. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
String kindInput = kindTextField.getText();
if (isValidKind(kindInput)) {
    receipt.setKind(kindInput);
} else {
    JOptionPane.showMessageDialog(null, "Invalid kind. Please use Coats, Shirts, Skirts, Trousers.");
    return;
}
```

Figure 12

```
private boolean isValidKind(String kind) {  
    if (kind != null && !kind.isEmpty()) {  
        if (kind.equals("Coats") || kind.equals("Shirts") || kind.equals("Skirts") || kind.equals("Trousers")) {  
            return true;  
        }  
    }  
    return false;  
}
```

Figure 13

- d) Για το field **Sales** θα πρέπει ο χρήστης να εισάγει ένα double. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
String salesInput = salesTextField.getText().trim();  
if (isValidDouble(salesInput)) {  
    receipt.setSales(Double.parseDouble(salesInput));  
} else {  
    JOptionPane.showMessageDialog(null, "Invalid sales amount. Please enter a valid number.");  
    return;  
}
```

Figure 14

```
private boolean isValidDouble(String input) {  
    try {  
        Double.parseDouble(input);  
        return true;  
    } catch (NumberFormatException e) {  
        return false;  
    }  
}
```

Figure15

- e) Για το field **Items** θα πρέπει ο χρήστης να εισάγει έναν αριθμό. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
String itemsInput = itemsTextField.getText().trim();
if (isValidInteger(itemsInput)) {
    receipt.setItems(Integer.parseInt(itemsInput));
} else {
    JOptionPane.showMessageDialog(null, "Invalid number of items. Please enter a valid integer.");
    return;
}
```

Figure 16

- f) Για το field **Company** θα πρέπει ο χρήστης να εισάγει έναν αλφαριθμητικό. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
String companyNameInput = companyTextField.getText().trim();
if (isValidStringInput(companyNameInput)) {
    receipt.getCompany().setName(companyNameInput);
} else {
    JOptionPane.showMessageDialog(null, "Invalid company name. Please enter a valid string.");
    return;
}
```

Figure 17

- g) Για το fields **City, Country, Street** θα πρέπει ο χρήστης να εισάγει έναν αλφαριθμητικό. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```
Address companyAddress = receipt.getCompany().getCompanyAddress();
String countryInput = countryTextField.getText().trim();
String cityInput = cityTextField.getText().trim();
String streetInput = streetTextField.getText().trim();

if (isValidString(countryInput) && isValidString(cityInput) && isValidString(streetInput)) {
    companyAddress.setCountry(countryInput);
    companyAddress.setCity(cityInput);
    companyAddress.setStreet(streetInput);
} else {
    JOptionPane.showMessageDialog(null, "Invalid City or Country or Street name. Please enter valid strings.");
    return;
}
```

Figure 18

```

private boolean isValidInteger(String input) {
    try {
        Integer.parseInt(input);
        return true;
    } catch (NumberFormatException e) {
        return false;
    }
}

```

Figure 19

- h) Για το field **Number** θα πρέπει ο χρήστης να εισάγει έναν αριθμό. Εάν είναι οτιδήποτε άλλο το μήνυμα λάθους είναι το ακόλουθο:

```

String numberInput = numberTextField.getText().trim();
if (isValidInteger(numberInput)) {
    receipt.getCompany().getCompanyAddress().setStreetNumber(Integer.parseInt(numberInput));
} else {
    JOptionPane.showMessageDialog(null, "Invalid street number. Please enter a valid integer.");
    return;
}

```

Figure 20

Αλλαγές έγιναν και στα μηνύματα των errors. Αρχικά είχαμε πρόβλημα στην αναγνώριση των χαρακτήρων λόγω της κωδικοποίησης των ελληνικών γραμμάτων. Παρόλα αυτά για να εξασφαλίσουμε την απουσία του θέματος αυτού όλα τα μηνύματα πλέον είναι στα αγγλικά.

- Τέλος , πραγματοποιήσαμε κάποιες αλλαγές στο interface της εφαρμογής. Σας παραθέτουμε το ανανεωμένο παρακάτω:

Επιλέξτε είδος αρχείου για φόρτωση αποδείξεων:

Εισαγωγή από TXT

Εισαγωγή από XML

Εισαγωγή από HTML

Λίστα Αντιπροσώπων

OK

Cancel

Figure 21

Αντιπρόσωπος: Apostolos Zarras

Προσθήκη νέας απόδειξης

Επιλέξτε πληροφορίες προς εμφάνιση

☒ Συνολική αξία πωλήσεων
☒ Συνολικός αριθμός πωλήσεων
☒ Πωλήσεις συγκεκριμένης κατηγορίας

☒ Μπλούζες
☒ Παντελόνια
☒ Παλτά
☒ Φούστες

☒ Εμφάνιση Προμήθειας

OK

Cancel

ReceiptID: 1

Date: 12/12/2024

Kind: Coats

Sales: 250

4

y: Hand Made cloths

Greece

City: Ioannina

Street: Kalari

Number: 23

Προσθήκη

Αποδείξεις που προστέθηκαν: 1

Message

Receipt added successfully

OK

Figure 22

Συνολική αξία

Σύνολο πωλήσεων

Πωλήσεις παντελονιών

Πωλήσεις Μπλουζών

Πωλήσεις παλτών

Πωλήσεις φουστών

Προμήθεια

Εξαγωγή σε TXT

Εξαγωγή σε XML

Εξαγωγή σε HTML

OK Cancel

Figure 23

Running the code

Για να τρέξουμε το application όταν ανοίγει πρώτη φορά η εφαρμογή προτείνουμε τα παρακάτω:

- Κάνοντας δεξί κλικ στο αρχείο src, πηγαίνουμε στα Properties και στη συνέχεια στο Java Build Path. Κατόπιν, στο Libraries πατάμε διπλό κλικ στο JRE System Library και επιλέγουμε το Alternate JRE βάζοντας το jdk που έχουμε διαθέσιμο. Εμείς χρησιμοποιούμε jdk-21.

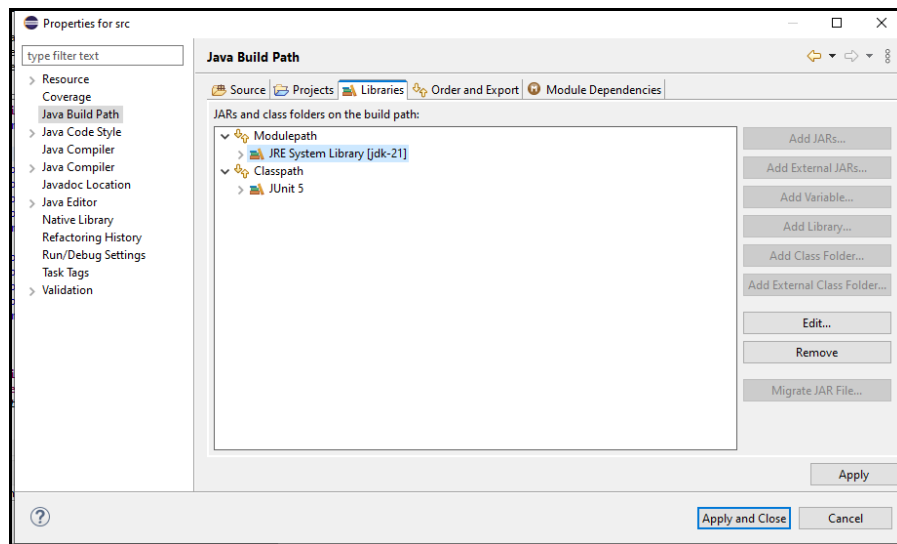


Figure 24

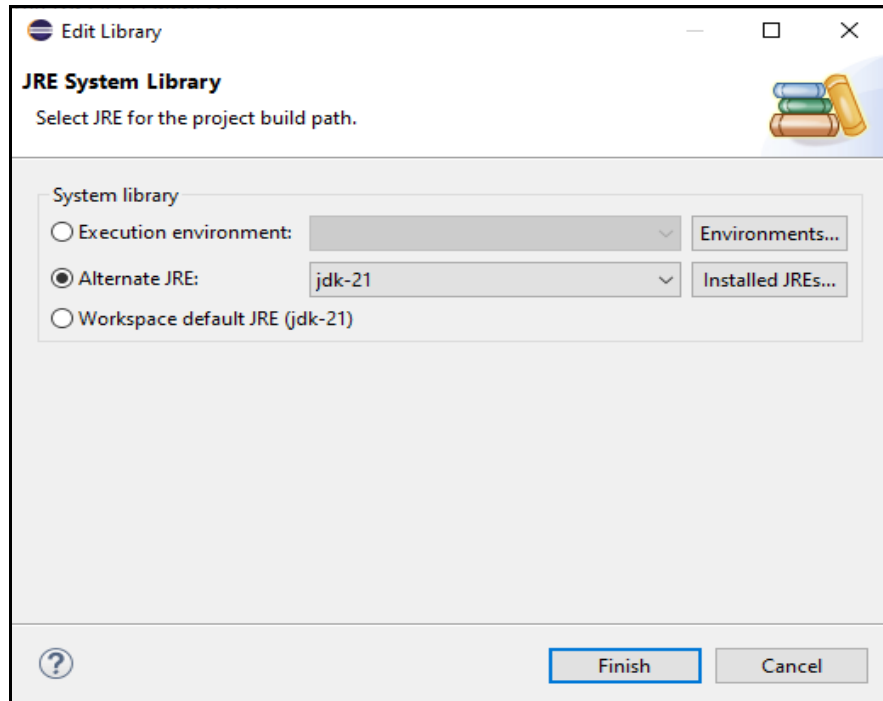


Figure 25