ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

INFORMATION RETRIEVAL

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2014-2015

ΟΜΑΔΑ007

ΓΙΝΑΡΓΥΡΟΣ ΝΙΚΟΣ ,2038

ΜΠΟΥΡΗΣ ΔΗΜΗΤΡΗΣ, 1894

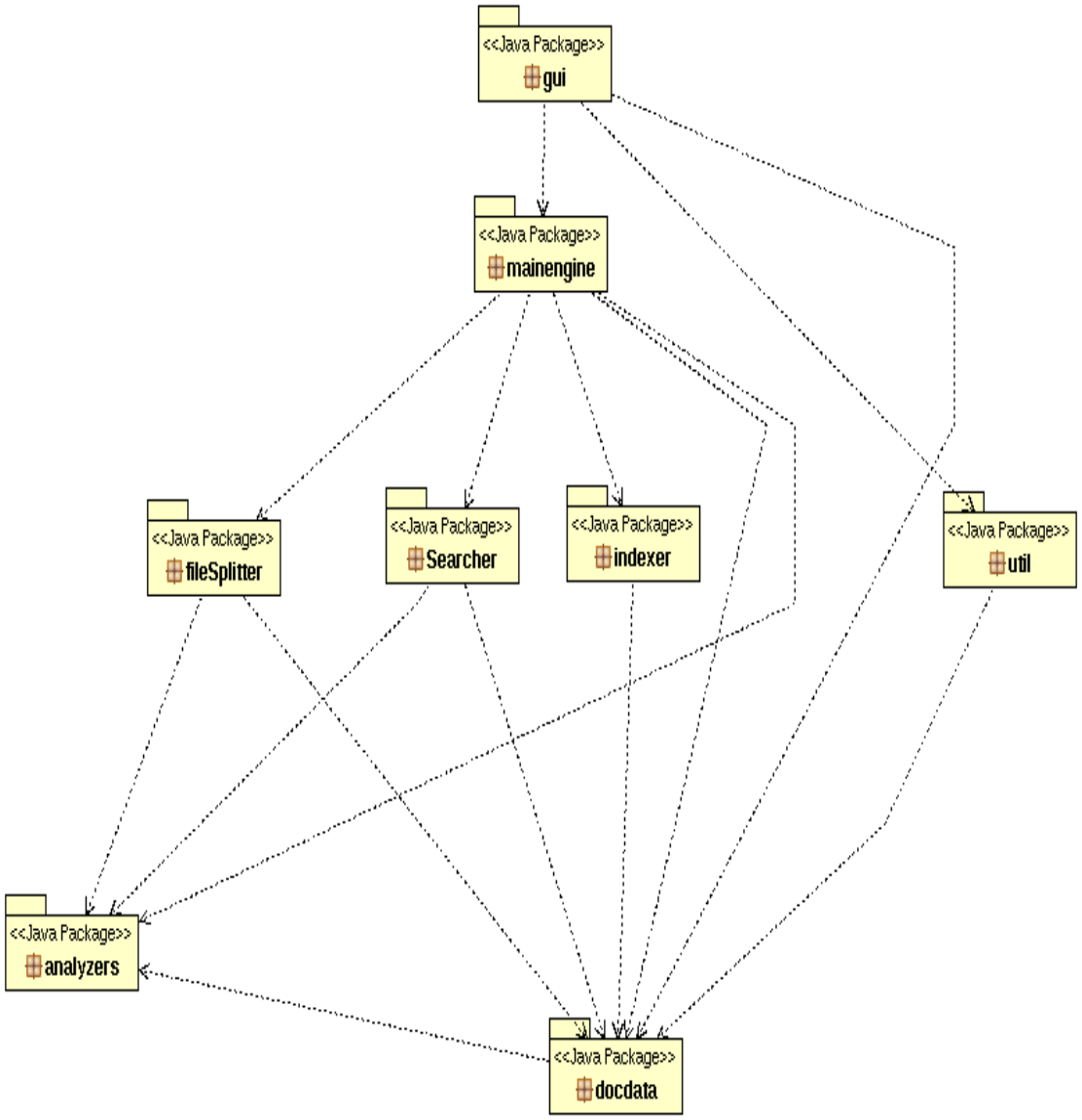
ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΔΕΚΕΜΒΡΙΟΣ 2014

1 CONTENTS

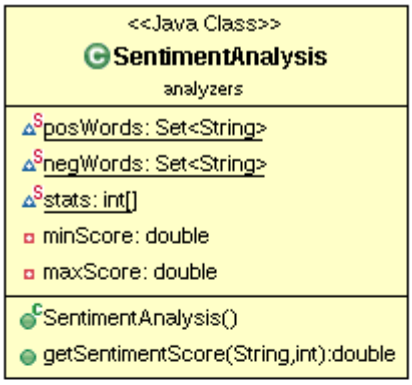
2 ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

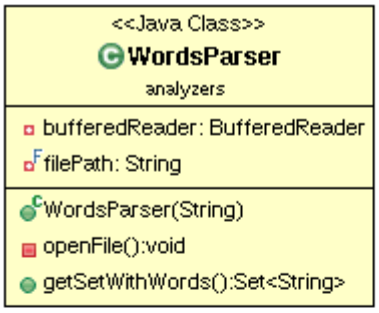
2.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ

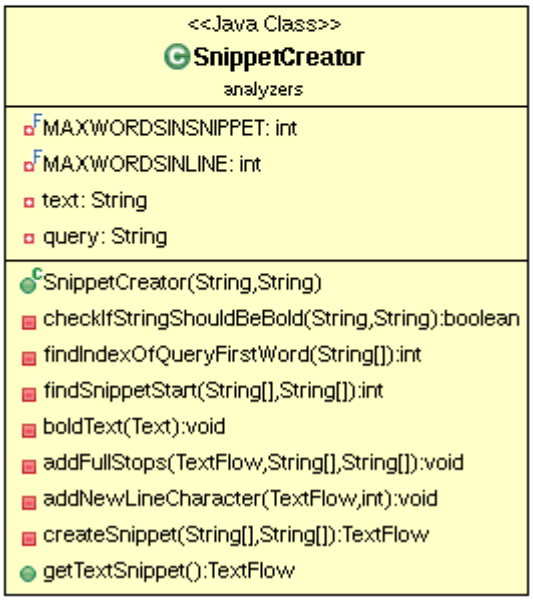


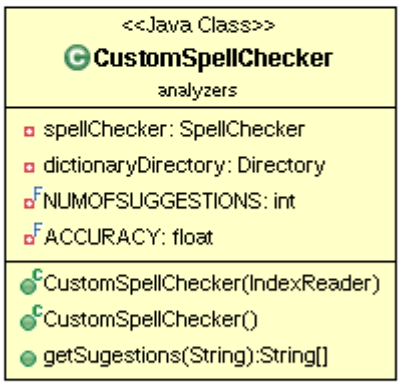
2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

package analyzers;

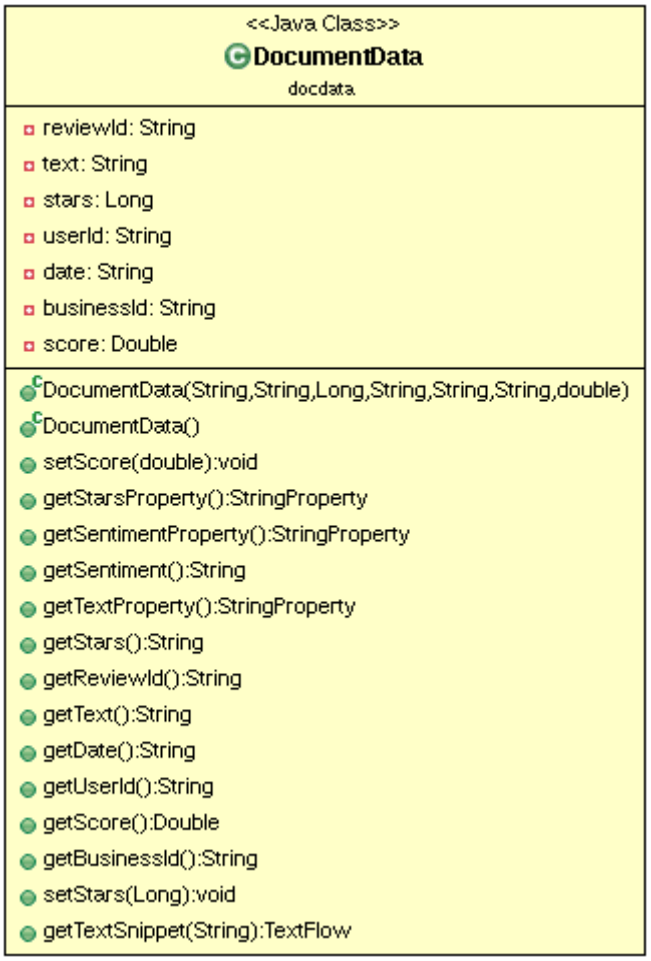




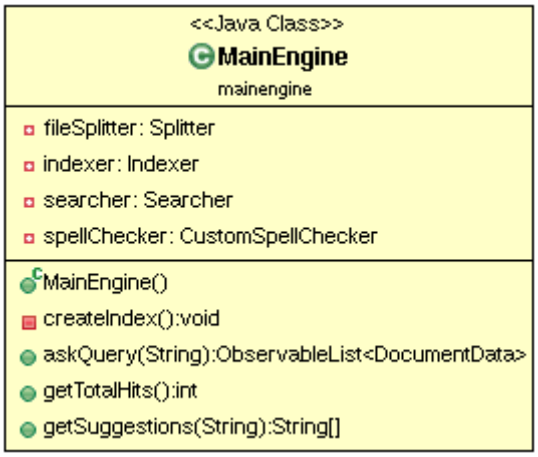




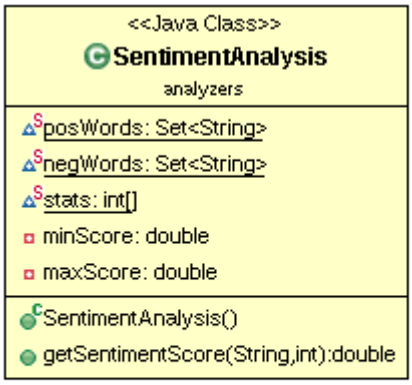
package docdata;

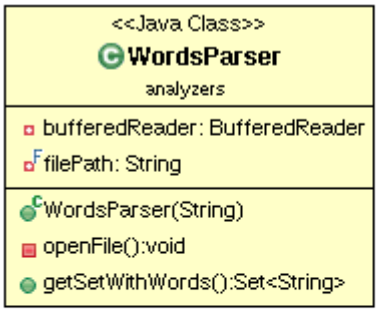


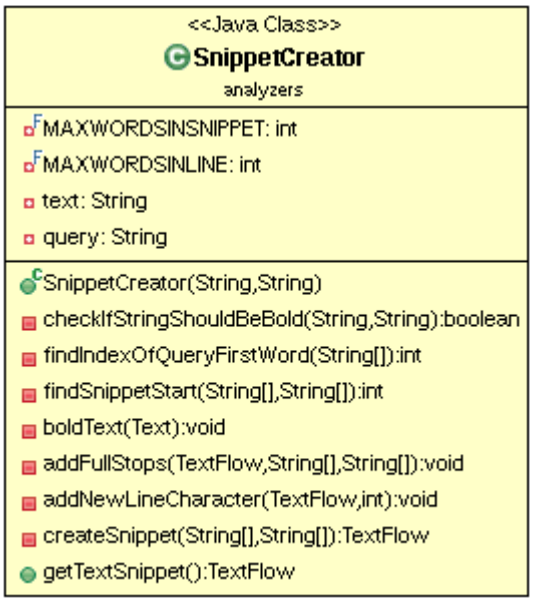
package mainengine;

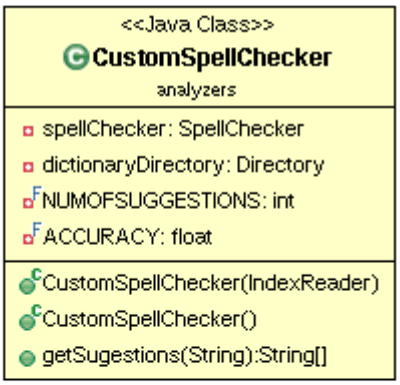


package analyzers;

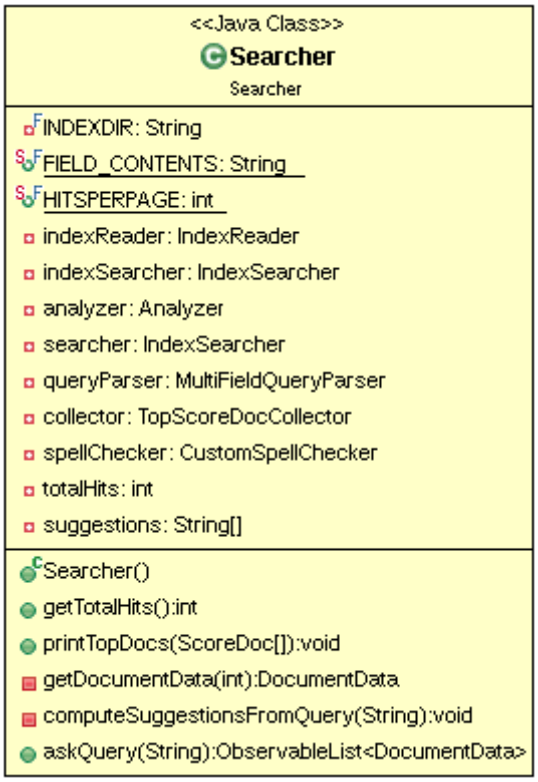




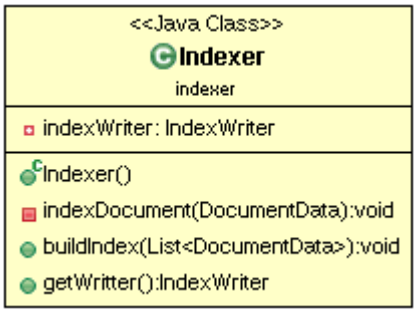




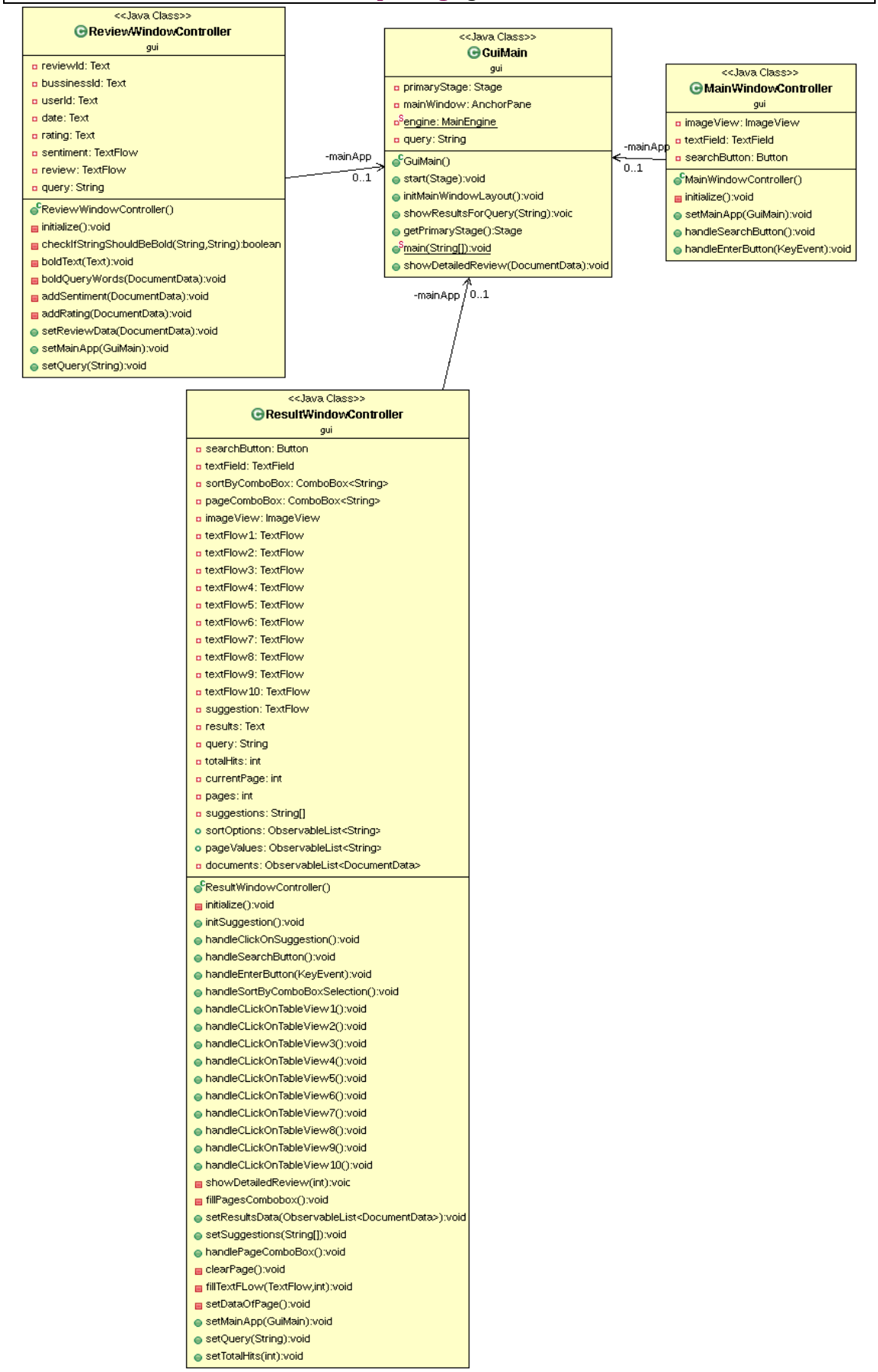
package searcher;



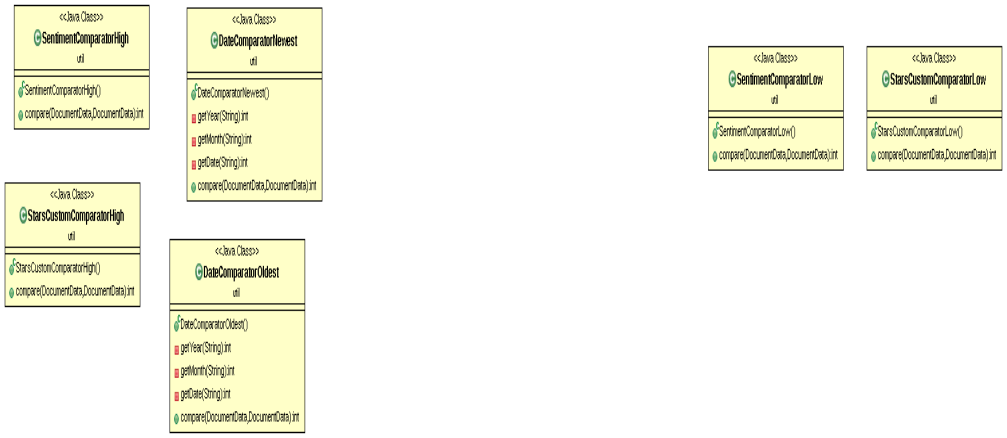
package indexer;



package gui;


<<Java Class>>
ReviewWindowController
gui
reviewId: Text
bussinessId: Text
userId: Text
date: Text
rating: Text
sentiment: TextFlow
review: TextFlow
query: String
ReviewWindowController()
initialize():void
checkIfStringShouldBeBold(String,String):boolean
boldText(Text):void
boldQueryWords(DocumentData):void
addSentiment(DocumentData):void
addRating(DocumentData):void
setReviewData(DocumentData):void
setMainApp(GuiMain):void
setQuery(String):void
-mainApp
0..1
<<Java Class>>
GuiMain
gui
primaryStage: Stage
mainWindow: AnchorPane
engine: MainEngine
query: String
GuiMain()
start(Stage):void
initMainWindowLayout():void
showResultsForQuery(String):voic
getPrimaryStage():Stage
main(String[]):void
showDetailedReview(DocumentData):void
-mainApp
0..1
<<Java Class>>
MainWindowController
gui
imageView: ImageView
textField: TextField
searchButton: Button
MainWindowController()
initialize():void
setMainApp(GuiMain):void
handleSearchButton():void
handleEnterButton(KeyEvent):void
-mainApp
0..1
<<Java Class>>
ResultWindowController
gui
searchButton: Button
textField: TextField
sortByComboBox: ComboBox<String>
pageComboBox: ComboBox<String>
imageView: ImageView
textFlow1: TextFlow
textFlow2: TextFlow
textFlow3: TextFlow
textFlow4: TextFlow
textFlow5: TextFlow
textFlow6: TextFlow
textFlow7: TextFlow
textFlow8: TextFlow
textFlow9: TextFlow
textFlow10: TextFlow
suggestion: TextFlow
results: Text
query: String
totalHits: int
currentPage: int
pages: int
suggestions: String[]
sortOptions: ObservableList<String>
pageValues: ObservableList<String>
documents: ObservableList<DocumentData>
ResultWindowController()
initialize():void
initSuggestion():void
handleClickOnSuggestion():void
handleSearchButton():void
handleEnterButton(KeyEvent):void
handleSortByComboBoxSelection():void
handleCLickOnTableView1():void
handleCLickOnTableView2():void
handleCLickOnTableView3():void
handleCLickOnTableView4():void
handleCLickOnTableView5():void
handleCLickOnTableView6():void
handleCLickOnTableView7():void
handleCLickOnTableView8():void
handleCLickOnTableView9():void
handleCLickOnTableView10():void
showDetailedReview(int):voic
fillPagesCombobox():void
setResultsData(ObservableList<DocumentData>):void
setSuggestions(String[]):void
handlePageComboBox():void
clearPage():void
fillTextFLow(TextFlow,int):void
setDataOfPage():void
setMainApp(GuiMain):void
setQuery(String):void
setTotalHits(int):void

```
package analyzers;
```



3 ΤΕΚΜΗΡΙΩΣΗ ΚΑΙ ΛΟΙΠΑ ΣΧΟΛΙΑ

3.1 ΣΥΛΛΟΓΗ

Η εφαρμογή έχει υλοποιηθεί πάνω στα δεδομένα της Yelp dataset challenge τα οποία αφορούν σε κριτικές(reviews) από εστιατόρια.Τα αρχικά δεδομένα είναι σε μορφή json.Για την δημιουργία του ευρετηρίου σπάσαμε το αρχικό αρχείο σε μικρότερα, και χρησιμοποιήσαμε το simple_json_api για το διαβασμά τους. Τα δεδομένα είνα περίπου 1.4 gb και περιέχουν περιπου 900.000 κριτικές.

3.2 ΔΙΟΡΘΩΣΗ ΛΑΘΩΝ

Η εφαρμογή υποστηρίζει διόρθωση λαθών.Για την υλοποίηση δημιουργήσαμε μια κλάση ονόματι CustomSpellChecker που κάνει wrap το SpellChecker της javafx.

Ο spellChecker είναι φτιαγμένος με λεξικό αγγλικών .Το λεξικό που χρησιμοποιήσαμε είναι των linux και βρισκεται στον φάκελο `/usr/share/dict/words.Είναι το αρχείο English-american.txt.Δεν υποστηρίζεται διόρθωση λαθών για ερωτήσεις με περισσότερες απο μία λέξεις.`

3.3 ΔΗΜΙΟΥΡΓΙΑ ΠΕΡΙΛΗΨΗΣ

Όταν ο χρήστης κάνει μια ερώτηση εμφανίζεται ένα παράθυρο με τις σελίδες των αποτελεσμάτων.Για κάθε κριτική(αποτέλεσμα) εμφανίζεται μια περίληψη(snippet) του κειμένου της στην οποία οι λέξεις της ερώτησης είναι έντονα σκιασμένες(bold).Για την κατασκευή της περίληψης δημιουργήσαμε την κλάση SnippetCreator η οποία επεξεργάζεται όλο το αρχικό κείμενο της κριτικής και εμφάνίζει κομμάτια που περιέχουν λέξεις της ερώτησης.

3.4 ΥΛΟΠΟΙΗΣΗ ΑΝΑΛΥΣΗΣ ΚΕΙΜΕΝΟΥ

Για την ανάλυση του κειμένου χρησιμοποιείται η κλάση ***splitter.fileSplitter.java.***Το αρχικό αρχείο με τις κριτικές (yelp_challenge_academic.json) χωρίστηκε σε 80 επιμέρους αρχεία τα οπόια φορτώνονται ανα 25.Για κάθε αρχείο που φορτώνεται διαβάζονται οι επιμέρους κριτικές του (reviewId, userId, text, rating, bussinesid) και για κάθε κρητική δημιουργείται ένα αντικείμενο της κλάσης documentdata.docData (κλάση που περιέχει τα δεδομένα μιας κριτικής).Το αντικείμενο αυτό εισάγεται σε μία λίστα που στο τέλος της διαδικασίας έχει όλες τις κριτικές από όλα τα αρχεία.Με αυτή την διαδικασία μετατρέπεται το αρχικό σύνολο δεδομένων σε αντικείμενα docData τα οπόια σε επόμενη φάση θα δημιουργήσουν το ευρετήριο.

3.5 ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΕΥΡΕΤΗΡΙΟΥ

Για την κατασκευή του ευρετηρίου χρησιμοποιούμε την λίστα που δημιουργείται όπως περιγράψαμε στο 3.4. Την διατρέχουμε και προσθέτουμε για κάθε κριτική που έχουμε διαβάσει μια εγγραφή στο ευρετήριο. Κρατάμε τα βασικά πεδία μιας κριτικής user_id, business_id, review_id, rating, text που τα χρησιμοποιούμε και στην παρουσίαση που κάνουμε για το review στον χρήστη

3.6 ΕΠΕΞΕΡΓΑΣΙΑ ΤΗΣ ΕΡΩΤΗΣΗΣ

Η ερώτηση που κάνει ο χρήστης υπόκειται στην ίδια επεξεργασία με αυτήν που κάναμε στα κείμενα των κριτικών. Γι αυτό τον σκοπό χρησιμοποιήθηκε ο EnglishAnalyzer της Lucene που υποστηρίζει και stemming. Η ερώτηση εκτελείται απευθείας από τον core της lucene και αναλόγως με τον αριθμό των αποτελεσμάτων χρησιμοποιούμε τον δικό μας Custom Spell Checker ώστε να κάνουμε πρόταση στον χρήστη για να ψάξει κάποια άλλη λέξη που θα του φέρει καλύτερα αποτελέσματα.

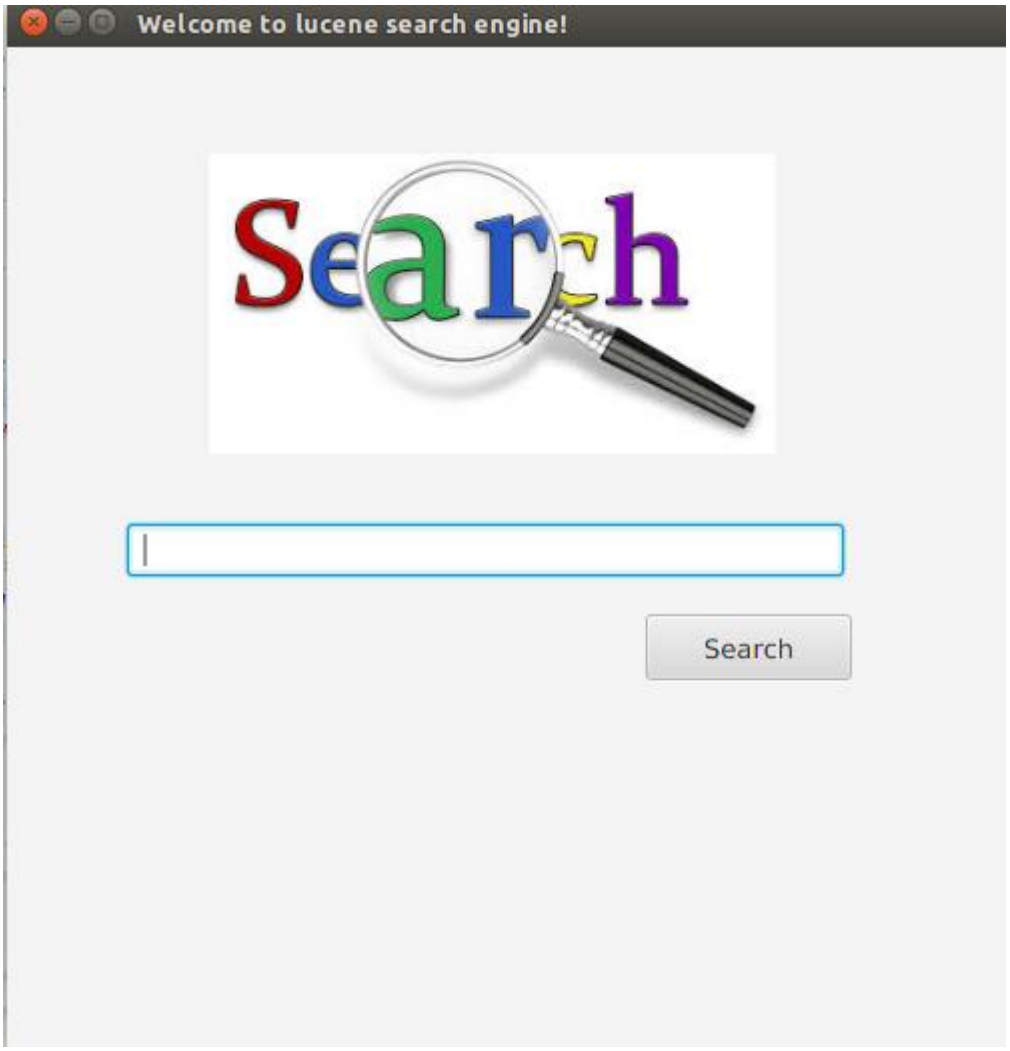
3.7 ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΡΩΤΗΣΗΣ

Για την εκτέλεση της ερώτησης χρησιμοποιούμε την κλάση `MultiFieldQueryParser` που ψάχνει για διάφορα πεδία στο ευρετήριο και τα συγκρίνει με την ερώτηση του χρήστη.Ο `IndexSearcher` ψάχνει στο ευρετήριο και αποθηκεύει τα αποτελέσματα σε ένα `TopScoreDocCollector`. Εμείς εμφανίζουμε στον χρήστη τα καλύτερα 100 αποτελέσματα που πήραμε για την ερώτηση που έκανε.

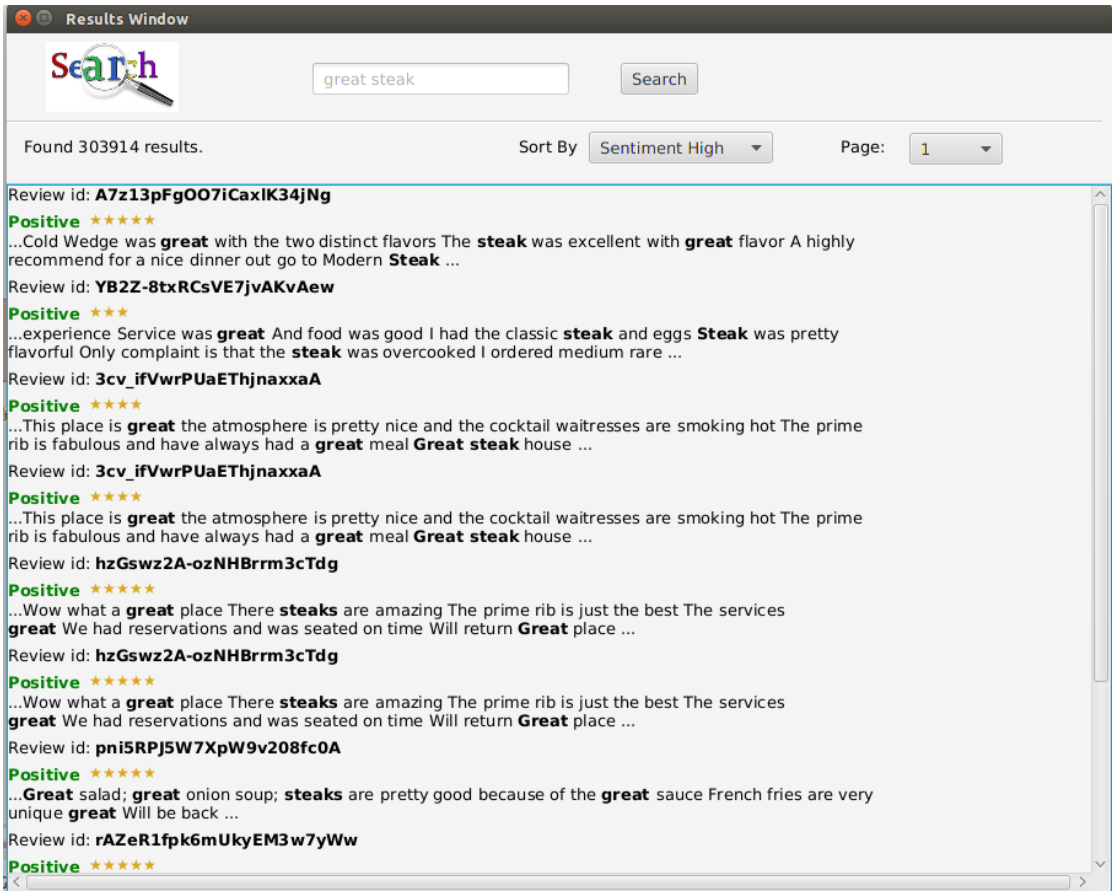
3.8 ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΟΣ

Για την παρουσίαση των αποτελεσμάτων εμφανίζουμε τα πρώτα 100 καλύτερα αποτελέσματα που πήραμε από την lucene ταξινομημένα ως προς το sentiment score που έχουμε υπολογίσει. Επίσης δίνεται η δυνατότητα στον χρήστη να ταξινομήσει τα αποτελέσματα με βάση το Sentiment, date, rating.

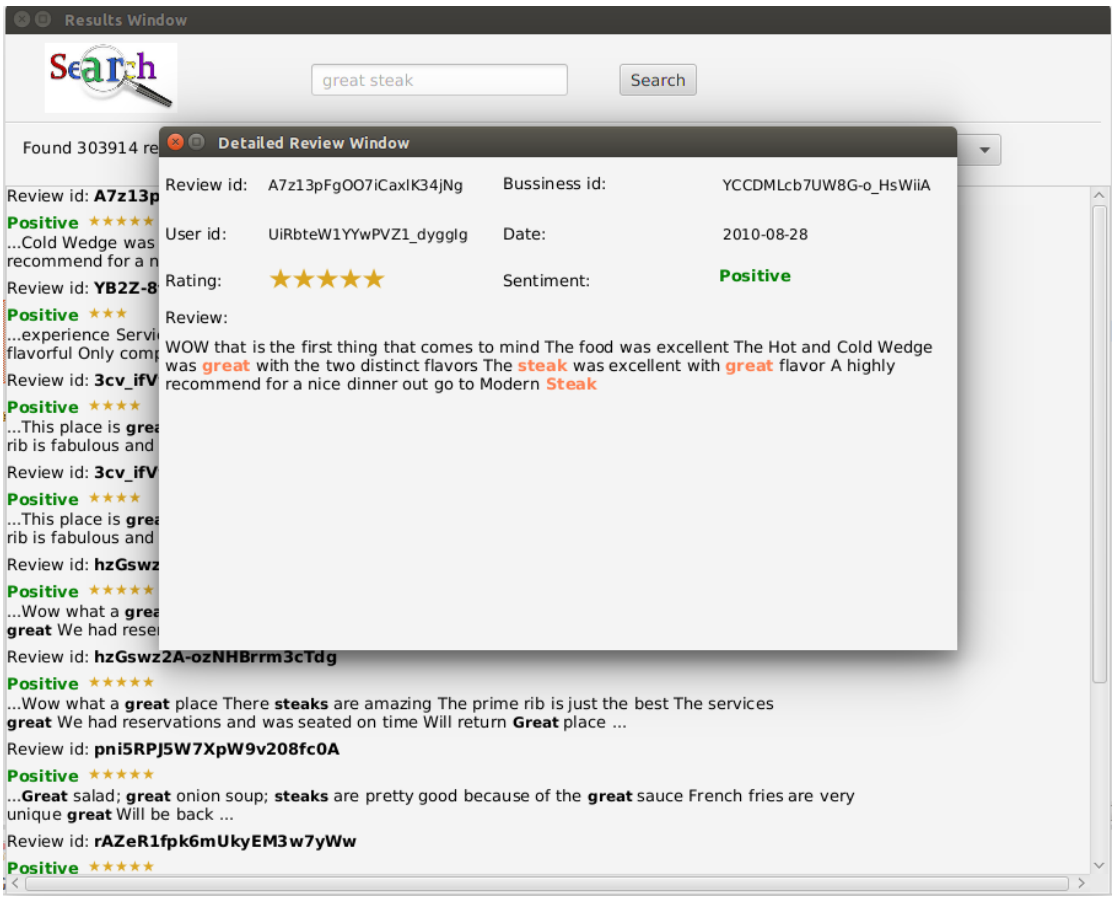
3.8.1 ΑΡΧΙΚΗ ΣΕΛΙΔΑ



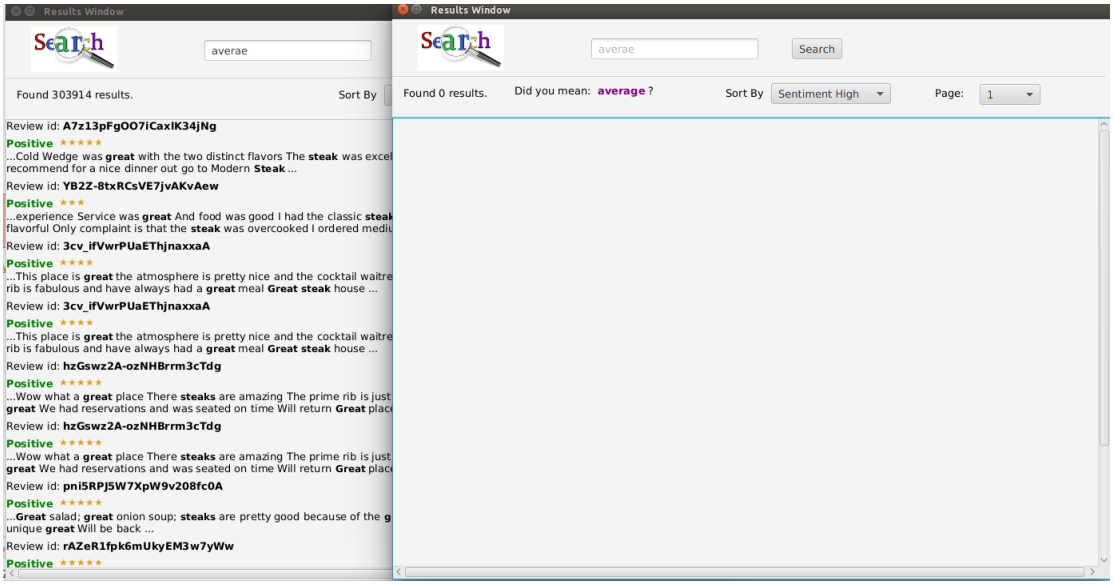
3.8.2 ΠΑΡΑΘΥΡΟ ΕΜΦΑΝΙΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ



3.8.3 ΠΑΡΑΘΥΡΟ ΕΜΦΑΝΙΣΗΣ ΛΕΠΤΟΜΕΡΕΙΩΝ ΚΡΙΤΙΚΗΣ



3.8.4 SPELL CHECKING



4 SENTIMENT ANALYSIS

Για την ανάλυση του αισθήματος των κριτικών αναλύουμε το κείμενο λέξη προς λέξη. Έχουμε ένα σύνολο λέξεων περίπου 15.000 που έχουν χαρακτηριστεί από πριν ως θετικές η αρνητικές. Οι λέξεις αυτές είναι γραμμένες σε δυο διαφορετικά αρχεία το positive-words και το negative-words. Κάθε λέξη της κριτικής συγκρίνεται με αυτά τα δυο σύνολα. Για λόγους απόδοσης τα δυο αυτά αρχεία έχουν αποθηκευθεί προσωρινά στην μνήμη σε ένα hashSet ώστε να πετύχουμε καλύτερο χρόνο για την κατασκευή του ευρετηρίου. Έτσι αναλόγως ποιας κατηγορίας οι λέξεις είναι περισσότερες έχουμε μια αρχική εικόνα για το sentiment score. Επίσης σαν συμπλήρωμα χρησιμοποιούμε και την βαθμολογία που έχει δώσει ο χρήστης αν για παράδειγμα έχει δώσει 5 αστέρια το sentiment score του προσαυξάνεται κατά ένα 20%. Αν έχει δώσει 1 αστέρι μειώνεται κατά 20%. Αυτό μας βοηθάει καλύτερα στην διάταξη των αποτελεσμάτων. Στο πρόγραμμα μας έχουμε προσθέσει και μια από τις πολλές ειδικές περιπτώσεις που μπορούν να εμφανιστούν αν κάνουμε και περαιτέρω ανάλυση συμφραζομένων. Η περίπτωση αυτή είναι όταν πριν από μια θετική η αρνητική λέξη βρίσκεται η λέξη not.

5 ΠΑΡΑΠΟΜΠΕΣ

- Oracle Java 8
- Eclipse Luna
- Lucene 4.10.4
- Json-Simple Parser
- JavaFx (GUI)
- Scene Builder
- Object Aid UML Explorer
- Yelp dataset challenge