

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ»

| | |
|------------------------------|------------------------|
| Τελική | ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ |
| Όνομα φοιτητών – Αρ. Μητρώου | ΜΑΡΚΟ ΠΛΑΚΟΥ Π17107 |
| | ΣΤΑΥΡΟΣ ΜΠΑΚΕΑΣ ?????? |
| Ημερομηνία παράδοσης | ?/?/2020 |

Εκφώνηση της άσκησης :

ΘΕΜΑ ΕΡΓΑΣΙΑΣ ΟΜΑΔΑΣ 9

ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ: Με βάση το ειδικό θέμα της κάθε ομάδας καλείστε να αναπτύξετε λογισμικό, καθώς και να το συνοδεύσετε με την αντίστοιχη ανάπτυξή του σε μοντέλο κύκλου ζωής λογισμικού. Σύμφωνα με το ειδικό θέμα της εργασίας που έχει δοθεί στην ομάδα σας να γίνουν τα εξής:

1. Καταγραφή και παρουσίαση της ανάλυσης απαιτήσεων της εργασίας σας. Παραγωγή παρουσίασης απαιτήσεων και αρχιτεκτονικής δομής με τη χρήση των βασικών διαγραμμάτων της UML.
2. Ανάπτυξη των τεσσάρων φάσεων του αντικειμενοστραφούς μοντέλου ανάπτυξης λογισμικού (Rational Unified Process) χρησιμοποιώντας τα 9 διαγράμματα της UML με χρήση ενός εργαλείου CASE της επιλογής σας. Προτεινόμενα CASE TOOLS: Rational Rose, Visual Studio 2010 Ultimate, ArgoUML.
3. Ανάπτυξη της εργασίας. Η εργασία πρέπει να είναι πλήρως λειτουργική και σωστά δομημένη. Χρήση της γλώσσας ή του εργαλείου που έχει δηλώσει η ομάδα σας. Εφόσον υπάρχει, χρήση της γλώσσας/τεχνολογίας που υποδηλώνει το θέμα της εργασίας σας. Σε κάθε περίπτωση, προτείνεται η χρήση κάποιας αντικειμενοστραφούς γλώσσας προγραμματισμού.

Διευκρινήσεις:

➤ Τα διαγράμματα που θα υλοποιήσετε κατά την ανάπτυξη των μοντέλων θα πρέπει υποχρεωτικά να σχετίζονται με την υλοποιημένη εργασία και το αντίστροφο.

ΕΙΔΙΚΟ ΘΕΜΑ ΟΜΑΔΑΣ:

Η ομάδα σας θα υλοποιήσει μια ιστοσελίδα αξιολόγησης εστιατορίων.

Μπορείτε να πάρετε ιδέες από

πολλές ιστοσελίδες που υπάρχουν στο διαδίκτυο, όπως η:

<http://www.askforfood.gr/>

Δεν πρέπει να χρησιμοποιήσετε κάποια open source πλατφόρμα ως βάση της ιστοσελίδας σας.

Τελικός/οι χρήστης/ες της εφαρμογής: οι ενδιαφερόμενοι για καταχώρηση ενός εστιατορίου, οι

ενδιαφερόμενοι για αξιολόγηση των διαθέσιμων εστιατορίων, ο διαχειριστής και οι επισκέπτες.

Hint: Μας ενδιαφέρει και ο αλγόριθμος με τον οποίο θα γίνει η κατάταξη των διαθέσιμων εστιατορίων.

Πρέπει να είναι όσο το δυνατόν πιο «δίκαιος» γίνεται.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή

1.1 Στόχοι της εργασίας.....

1.2 Ορισμός του προβλήματος προς επίλυση.....

2. Σύνοψη παρουσίαση της RUP

3. Φάση: Έναρξη (Inception)

3.1 Σύλληψη απαιτήσεων.....

3.2 Ανάλυση-Σχεδιασμός.....

3.2.1 Διαγράμματα Περιπτώσεων Χρήσης.....

3.2.2 Διαγράμματα Τάξεων.....

4. Φάση: Εκπόνηση Μελέτης (Elaboration)

4.1 Ανάλυση-Σχεδιασμός.....

4.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2 η έκδοση).....

4.1.2 Διαγράμματα Τάξεων (2 η έκδοση).....

4.1.3 Διαγράμματα Αντικειμένων (1 η έκδοση).....

4.1.4 Διαγράμματα Συνεργασίας (1 η έκδοση).....

4.1.5 Διαγράμματα Σειράς (1 η έκδοση).....

4.1.6 Διαγράμματα Δραστηριοτήτων (1 η έκδοση).....

4.1.7 Διαγράμματα Καταστάσεων (1 η έκδοση).....

4.1.8 Διαγράμματα Εξαρτημάτων (1 η έκδοση).....

4.1.9 Διαγράμματα Διανομής (1 η έκδοση).....

4.2 Υλοποίηση-Έλεγχος

4.2.1 Υλοποίηση: 1 η εκτελέσιμη έκδοση.....

4.2.2 Αναφορά ελέγχου για την 1 η εκτελέσιμη έκδοση.....

5. Φάση: Κατασκευή (Construction)

5.1 Ανάλυση-Σχεδιασμός.....

5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3 η έκδοση).....

5.1.2 Διαγράμματα Τάξεων (3 η έκδοση).....

5.1.3 Διαγράμματα Αντικειμένων (2 η έκδοση).....

5.1.4 Διαγράμματα Συνεργασίας (2 η έκδοση).....

5.1.5 Διαγράμματα Σειράς (2 η έκδοση).....

5.1.6 Διαγράμματα Δραστηριοτήτων (2 η έκδοση).....

5.1.7 Διαγράμματα Καταστάσεων (2 η έκδοση).....

5.1.8 Διαγράμματα Εξαρτημάτων (2 η έκδοση).....

5.1.9 Διαγράμματα Διανομής (2 η έκδοση).....

5.2 Υλοποίηση-Έλεγχος

5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση.....

5.2.2 Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση.....

6. Εγχειρίδιο Χρήστη

6.1 Σύνοψη παρουσίαση του προγράμματος.....

6.2 Παρουσίαση σεναρίων λειτουργίας.....

1. Εισαγωγή

(1) Στόχοι της εργασίας

Στόχος της εργασίας στο μάθημα της τεχνολογίας λογισμικού όσον αφορά το εκπαιδευτικό κομμάτι είναι η κατανόηση της διαδικασίας RUP με την οποία θα γίνει η ανάλυση και ο σχεδιασμός του λογισμικού. Η παρουσίαση των φάσεων του μοντέλου θα γίνει διαγραμματικά σύμφωνα με την γλώσσα μοντελοποίησης λογισμικού UML στα πρότυπα-Φάσεις της RUP διαδικασίας.

Όσον αφορά το κομμάτι του λογισμικού, η εργασία της ομάδας 9 που μας έχει ανατεθεί είναι η υλοποίηση μιας ιστοσελίδας αξιολόγησης εστιατορίων.

(2) Ορισμός του προβλήματος προς επίλυση.

Το πρόβλημα που καλούμαστε να επιλύσουμε όπως προαναφέρθηκε είναι η δημιουργία μιας ιστοσελίδας αξιολόγησης εστιατορίων. Αναλυτικότερα δεν θα πρέπει να χρησιμοποιήσουμε κάποια έτοιμη βάση δεδομένων όποτε καλούμαστε να δημιουργήσουμε την βάση και τα δεδομένα. Ως τελικοί χρήστες θα είναι οι ενδιαφερόμενοι για καταχώρηση εστιατορίου, οι ενδιαφερόμενοι για αξιολόγηση εστιατορίων, ο διαχειριστής και οι επισκέπτες. Εκτός αυτού ιδιαίτερη σημασία έχει και ο αλγόριθμος κατάταξης των εστιατορίων σύμφωνα με τις κριτικές τους ώστε να είναι αντικειμενικός και δίκαιος. Τέλος, ως εργαλείο υλοποίησης (IDE) είναι το eclipse. Επιπλέον η υλοποίηση θα γίνει στην γλώσσα προγραμματισμού java με την χρήση jsp σελιδών και των java servlets ακολουθώντας την αρχιτεκτονική mvc. Η βάση δεδομένων θα υλοποιηθεί με την χρήση της PostgreSQL 10 και ως τοπικός server χρησιμοποιείται ο apache tomcat v8.5.

2.Σύντομη Παρουσίαση της RUP

Η Rational Unified Process (RUP) είναι μία διαδικασία υλοποίησης λογισμικού. Κάθε τέτοια διαδικασία, ουσιαστικά χωρίζει την υλοποίηση σε συγκεκριμένα μέρη. Αυτός ο διαχωρισμός διευκολύνει την ομάδα υπεύθυνη για την υλοποίηση, να κατανοήσει σε βάθος τις απαιτήσεις που έχει το προϊόν. Πιο απλά, βοηθά την ομάδα να έχει ένα καθαρό πλάνο για το πως θα “φτιάξει” το προϊόν αυτό.

Οι φάσεις στις οποίες χωρίζεται σε σχέση με το χρόνο είναι τέσσερις και είναι οι εξής:

1)Φάση έναρξης. Σε αυτή την φάση, καθορίζεται η κατεύθυνση προς την οποία θα κινηθεί το λογισμικό.

2) Εκπόνηση μελέτης. Αφού είναι γνωστή η προοπτική του έργου, γίνεται εκτενής ερεύνα για τους πόρους που θα χρειαστεί το έργο και σχεδιάζεται μια πρώτη μορφή της αρχιτεκτονικής του, η οποία πρέπει να είναι σταθερή για να υπάρξει μετάβαση στην επόμενη φάση.

3)Υλοποίηση. Σε αυτή την φάση, αναπτύσσεται το λογισμικό δηλαδή γράφεται ο κώδικας. Για να γίνει όλη αυτή η διαδικασία ομαλά, πρέπει το προϊόν να αναπτυχθεί με βηματικές επαναλήψεις, δηλαδή να διαμοιραστεί η δουλεία σε τομείς, οι οποίοι είναι εύκολα διαχειρίσιμοι.

4) Μετάβαση. Η υλοποίηση ολοκληρώνεται και η παραγωγή ξεκινά. Οι χρήστες εκπαιδεύονται πάνω στο λογισμικό και γίνεται έλεγχος, εάν το προϊόν ανταποκρίνεται στις απαιτήσεις των πελατών.

3.Φάση Έναρξης

3.1) Σύλληψη Απαιτήσεων

Μπορούμε να χωρίσουμε το είδος των απαιτήσεων σε δύο μέρη. Τις λειτουργικές απαιτήσεις, οι οποίες περιγράφουν μία δραστηριότητα ή διαδικασία που πρέπει να φέρνει εις πέρας το σύστημα, και τις μη-λειτουργικές απαιτήσεις, οι οποίες περιγράφουν πώς το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις. Οι μη-λειτουργικές απαιτήσεις αφορούν τον προγραμματιστή.

Στην εφαρμογή μας, οι βασικές λειτουργικές απαιτήσεις είναι:

- Εγγραφή νέων χρηστών στην ιστοσελίδα.Αφορά τόσο τους πελάτες-κριτικούς όσο και τους εστιάτορες.
 - Είσοδος νέου χρήστη/Είσοδος υπάρχων χρήστη
 - Εμφάνιση βοηθητικών μυνημάτων στον νέο χρήστη, με σκοπό την ομαλότερη χρήση της ιστοσελίδας.
 - Δυνατότητα προβολής εστιατορίων προς κριτική.
 - Κατάταξη των εστιατορίων.Τα εστιατόρια με τον καλύτερο μέσο όρο προβάλλονται πρώτα
 - Δυνατότητα αξιολόγησης εστιατορίου από χρήστη, εφόσον έχει κάνει εγγραφή στην πλατφόρμα και έχει συνδεθεί.
 - Δυνατότητα εισαγωγής νέου εστιατορίου από ιδιοκτήτη.
 - Διαγραφή εστιατορίου από ιδιοκτήτη/διαχειριστή απο τον διαχειρηστή
 - Αποσύνδεση των χρηστών όλων των κατηγοριών από τον λογαριασμό τους.
 - Αποδοχή ή απορριψη εγγραφής νέου εστιατορίου απο τον διαχειρηστή
- Κατα την υλοποίηση εμφανίστηκαν περισσότερες λειτυργικές απαιτήσεις όπως:
- Εμφάνιση εστιατορίων ανα περιοχή και ανα κατηγορία κουζίνας.

- Διαγραφή άρμοστων σχολίων απο τον διαχειρηστή.
- Προβολή σχολίων εστιατορίου στους ιδιοκτήτες.

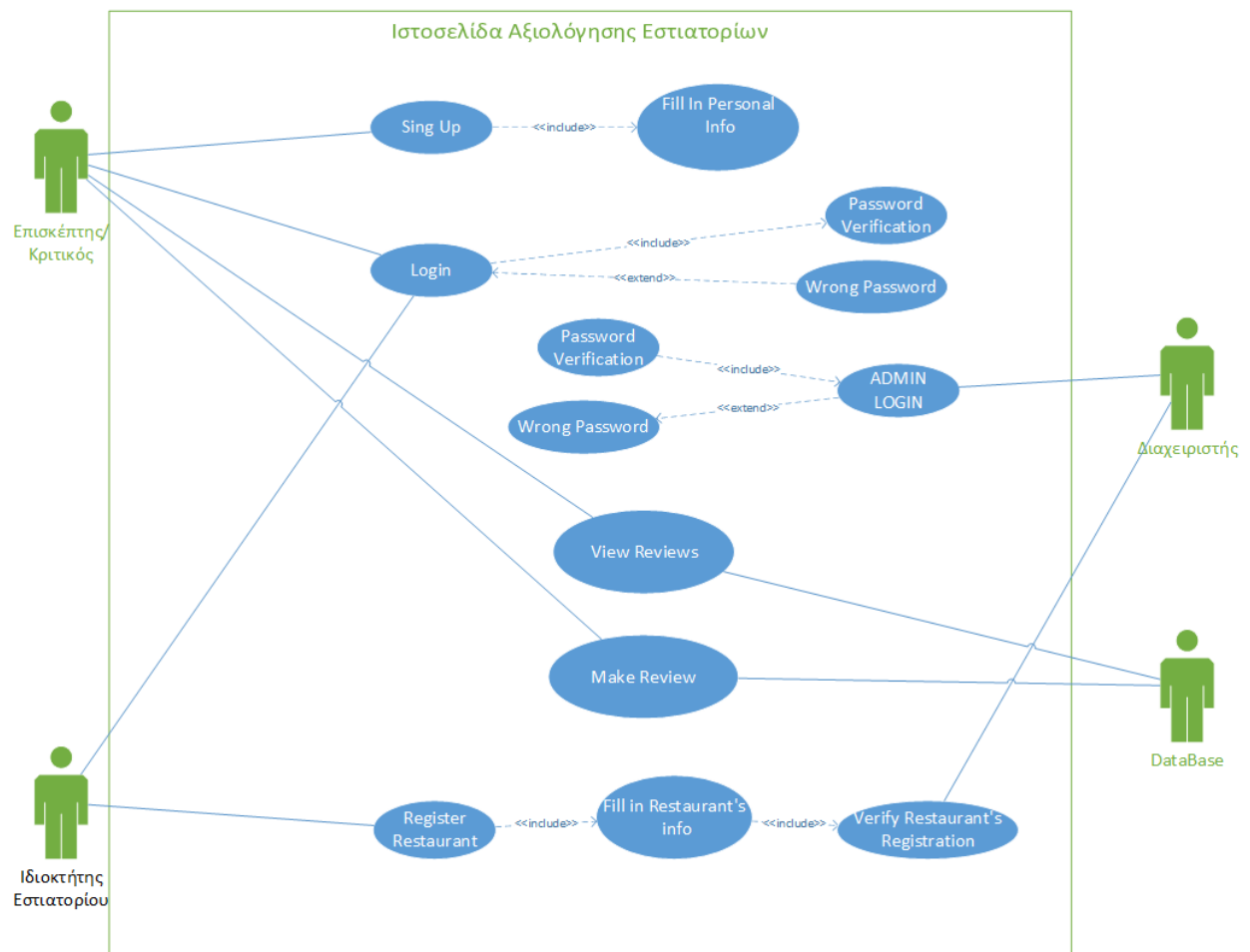
Κάποιες μη-λειτουργικές απαιτήσεις είναι:

- Δημιουργία και σύνδεση της εφαρμογής μας με βάση δεδομένων
- Περιορισμός περιήγησης του χρήστη από το address bar
- Προστασία προσωπικών δεδομένων του χρήστη.

3.2)Ανάλυση και Σχεδιασμός

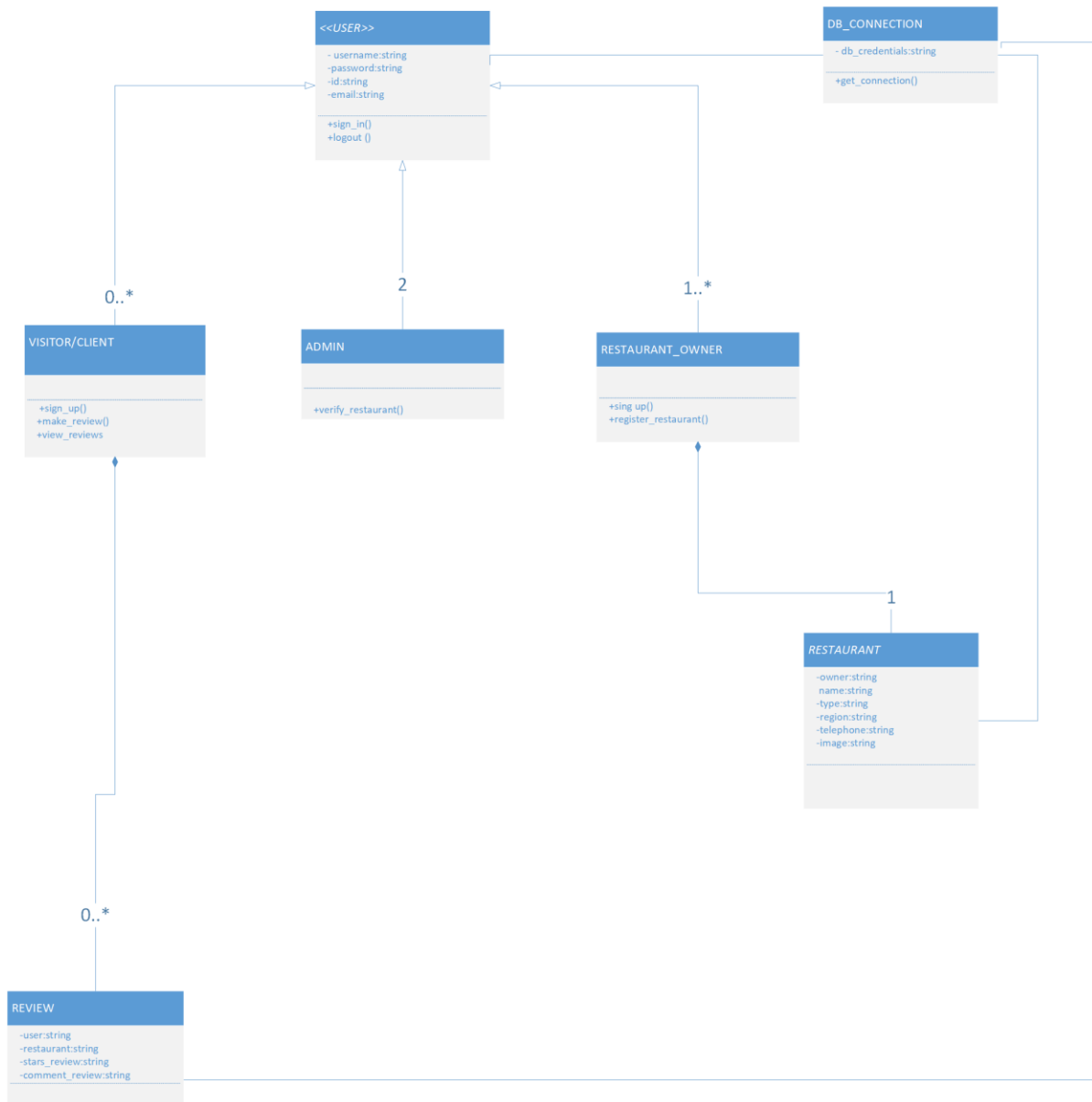
3.2.1)Διαγράμματα Περιπτώσεων Χρήσης

UML USE CASE DIAGRAMM PHASE 1 3.1.1



3.2.2) Διαγράμματα Κλάσεων

UML CLASS DIAGRAMM VERSION 1 PHASE 1 3.1.1



Εξήγηση

3.2.1)

Οι actors (τα ανθρωπάκια) χωρίζονται σε δύο κατηγορίες: Οι πρωταρχικοί actors, οι οποίοι βρίσκονται στην αριστερή πλευρά του διαγράμματος και χρησιμοποιούν το σύστημα έτσι ώστε να

πετύχουν ένα στόχο, και οι δευτερεύοντες actors, που βρίσκονται στην δεξιά πλευρά του διαγράμματος και το σύστημα χρειάζεται την βοήθειά τους για να πετύχουν οι πρωταρχικοί actors τον στόχο τους.

Στην εφαρμογή μας, έχουμε τους Χρήστες ως πρωταρχικούς actors και τους διαχειριστές ως δευτερεύοντες actors.

Οι χρήστες έχουν χωριστεί σε Επισκέπτες/Κριτικούς και Ιδιοκτήτες Εστιατορίου, γιατί θέλουμε να προσδιορίσουμε κάποια μοναδικά χαρακτηριστικά τους. Στο σύστημά μας, θέλουμε μόνο οι ιδιοκτήτες να μπορούν να κάνουν αίτηση για καταχώρηση ή διαγραφή εστιατορίου.

Οι περιπτώσεις χρήσης, χαρακτηρίζονται από δύο βασικές σχέσεις. Την include και την extend. Στην include, κάθε φορά που εκτελείται η βασική περίπτωση(η περίπτωση που δείχνει), εκτελείται και η included περίπτωση(η περίπτωση που την δείχνει το Include). Στην extend, κάθε φορά που εκτελείται η βασική περίπτωση, η extended περίπτωση εκτελείται όταν πληροί κάποια κριτήρια. Δηλαδή, εκτελείται κάποιες φορές, όχι πάντα.

Στην LOGIN, κάθε φορά που ο χρήστης θέλει να συνδεθεί, επαληθεύονται τα στοιχεία του. Άρα, η LOGIN είναι η βασική περίπτωση και η Password Verification είναι η included. Όμως, ο χρήστης μπορεί να κάνει λάθος στα στοιχεία του. Άρα η Wrong Credentials είναι η extended περίπτωση της LOGIN.

Στην View Restaurants, εάν δίνουμε την δυνατότητα στον χρήστη να δει όλα τα εστιατόρια, θα ήταν πρώτον μη πρακτικό, αφού θα έπρεπε να ψάξει πολλή ώρα για να βρει το εστιατόριο που θέλει, και δεύτερον χρονοβόρο, αφού θα έπρεπε να φορτώσουμε όλη την βάση. Για να λυθούν τα παραπάνω προβλήματα, αποφασίσαμε ο χρήστης να μπορεί να δει τα εστιατόρια μιας περιοχής που έχει επιλέξει. Άρα η περίπτωση View Restaurants

είναι η βασική και η `Select Area` η `included`. Επιπλέον, αν ο χρήστης το επιθυμεί, μπορεί να δει τα εστιατόρια ενός συγκεκριμένου είδους. Άρα η `ChooseToViewByKind` είναι η `extended` περίπτωση της `View Restaurants`.

Στην `Rate Restaurant`, κρίθηκε απαραίτητο ο χρήστης να βάζει αστέρια στην κριτική του, έτσι ώστε να μπορεί να δημιουργηθεί αλγόριθμος δίκαιης κατάταξης. Εάν θέλει, μπορεί να προσθέσει και μια περιγραφή της εμπειρίας του στο εστιατόριο. Άρα η περίπτωση `RateRestaurants` είναι η βασική, η περίπτωση `ChooseAmountOfStars` είναι η `included` και η περίπτωση `WriteDescription` είναι η `extended`. Υπάρχει η περίπτωση κάποιος χρήστης να είναι κακόβουλος και να χρησιμοποιεί απρεπές λεξιλόγιο, οπότε ο διαχειριστής πρέπει να έχει την δυνατότητα να σβήσει κάποια περιγραφή. Η βασική περίπτωση είναι η `WriteDescription` και η `extended` περίπτωση είναι η `DeleteForInappropriateContent`.

Στην `Register Restaurant`, μόνο οι ιδιοκτήτες εστιατορίου μπορούν να κάνουν αίτηση να καταχωρηθεί το εστιατόριό τους στην ιστοσελίδα. Για αυτό τον λόγο, είναι απαραίτητο να καταχωρούν πιο λεπτομερή στοιχεία για τους ίδιους αλλά και για το εστιατόριο. (Βασική περίπτωση: `Register Restaurant`, `Included` Περίπτωση: `Write Address and Contact Details`). Σε κάθε περίπτωση, πρέπει η εγκυρότητα αυτών των στοιχείων να ελέγχεται και μόνο τότε να καταχωρείται το εστιατόριο από τον διαχειριστή. (Βασική περίπτωση: `Register Included` περίπτωση: `Verify Restaurant`).

3.2.2)

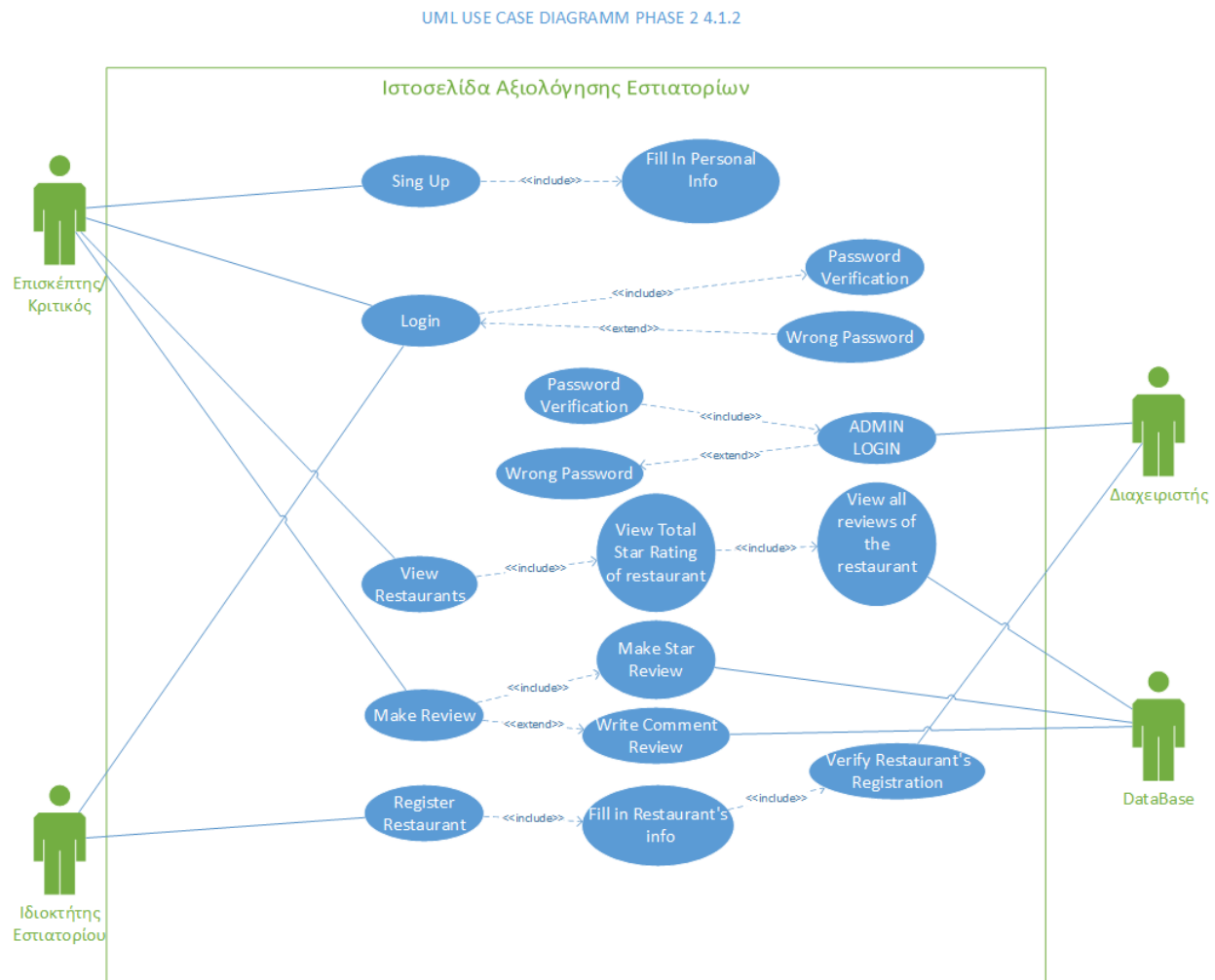
Οι κλάσεις `Visitor`, `Admin` και `Restaurant_Owner` είναι κλάσεις-παιδιά της `abstract` κλάσης `User`. Κληρονομούν όλες τις μεταβλητές και τις μεθόδους της, άρα δεν επαναλαμβανόμαστε, και κάθε κλάση έχει δικές της επιπλέον μεθόδους. Για

παράδειγμα, ο visitor δεν μπορεί να διαγράψει έναν χρήστη, ενώ ο διαχειριστής μπορεί. Η εφαρμογή μας δεν θα έχει περιορισμό για το πλήθος των χρηστών, αλλά θα πρέπει να έχει ακριβώς 2 διαχειριστές και τουλάχιστον 1 ιδιοκτήτη.

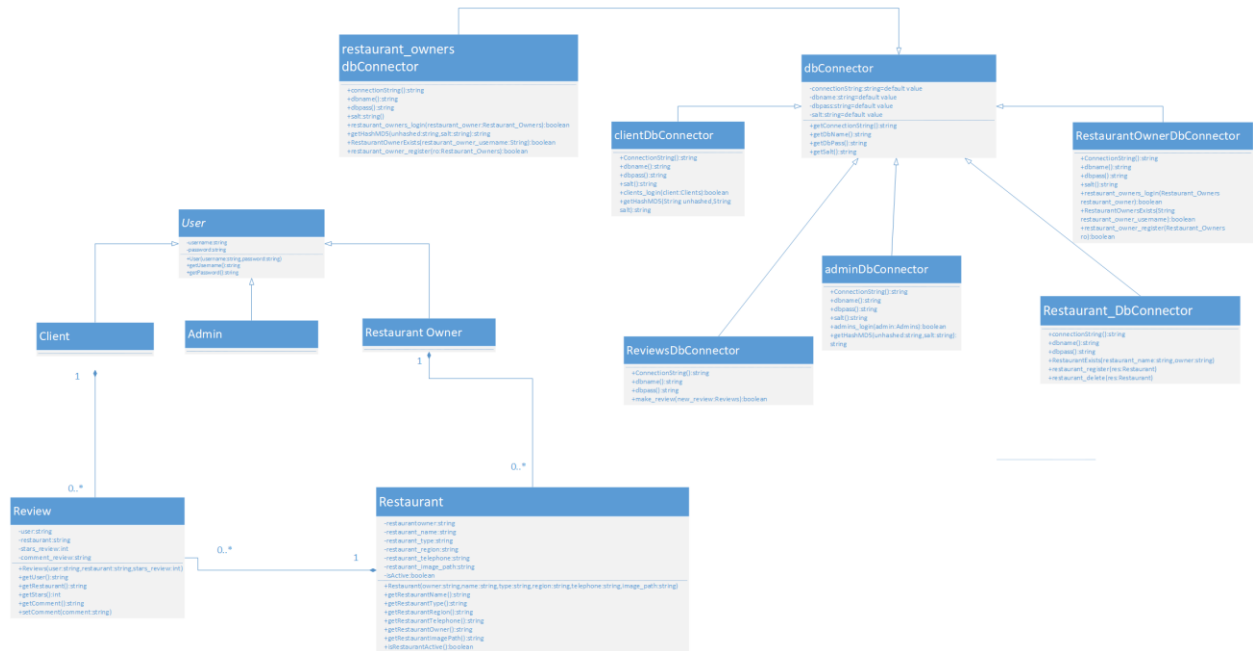
4.Φάση Εκπόνησης Μελέτης(Elaboration)

4.1)Ανάλυση-Σχεδιασμός

4.1.1

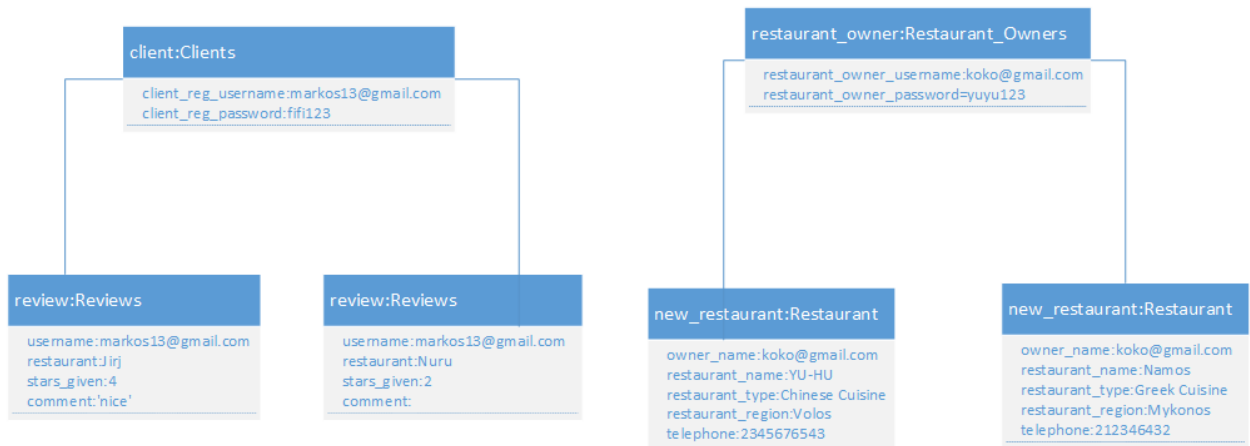


4.1.2

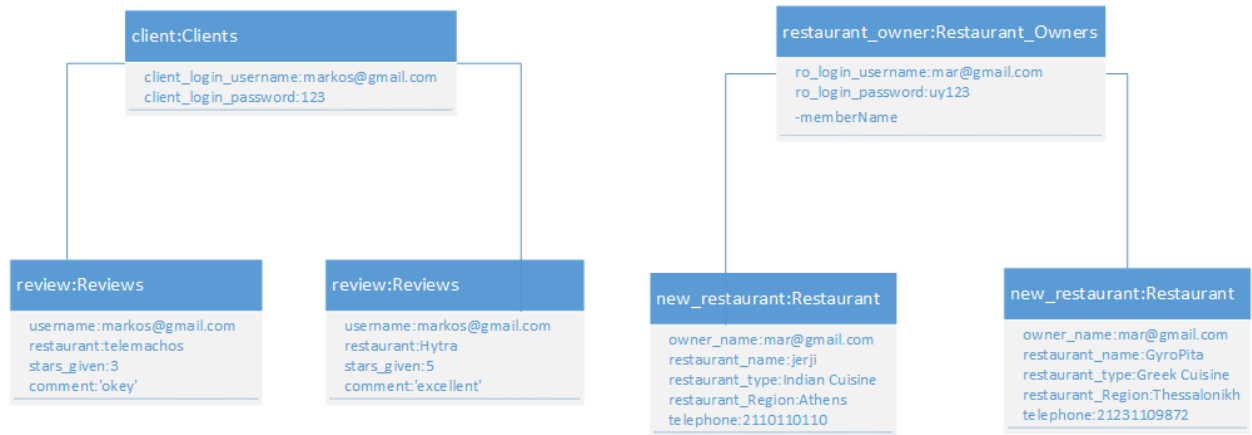


4.1.3

UML OBJECT DIAGRAM PHASE 2 FOR CLIENTS/RESTAURANT OWNERS AFTER REGISTRATION
4.1.3

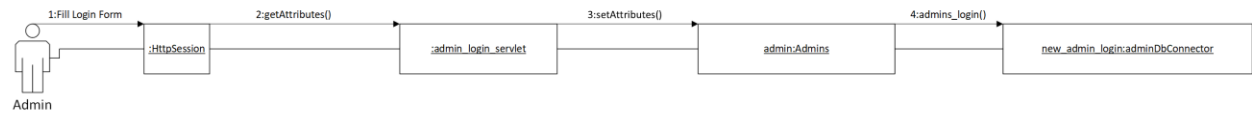


UML OBJECT DIAGRAMM PHASE 2 FOR EXISTING CLIENTS/RESTAURANT OWNERS 4.1.3

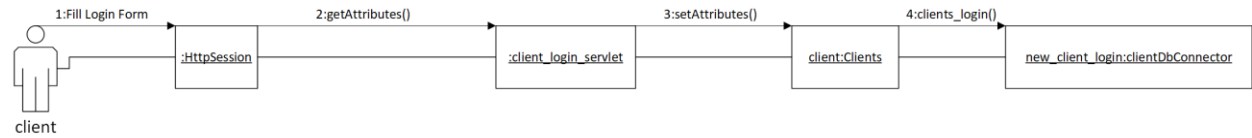


4.1.4

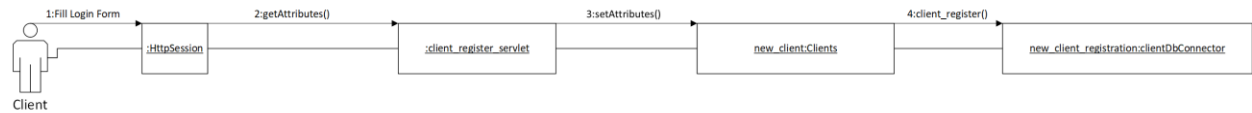
UML COLLABORATION DIAGRAMM FOR
ADMINS LOGIN PHASE 2 4.1.4



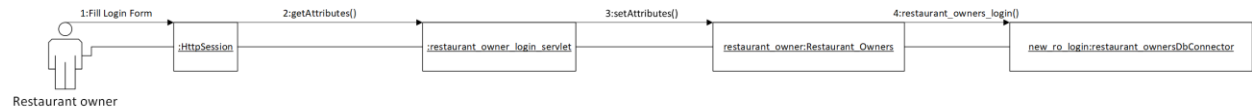
UML COLLABORATION DIAGRAMM FOR
CLIENT'S LOGIN PHASE 2 4.1.4



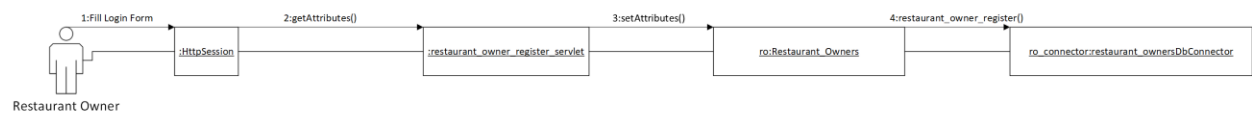
UML COLLABORATION DIAGRAMM FOR
CLIENTS REGISTER PHASE 2 4.1.4



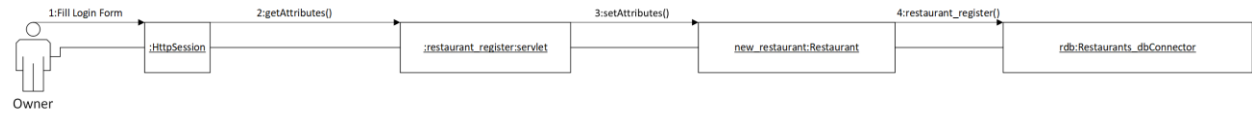
UML COLLABORATION DIAGRAMM FOR
RESTAURANT OWNER LOGIN PHASE 2
4.1.4



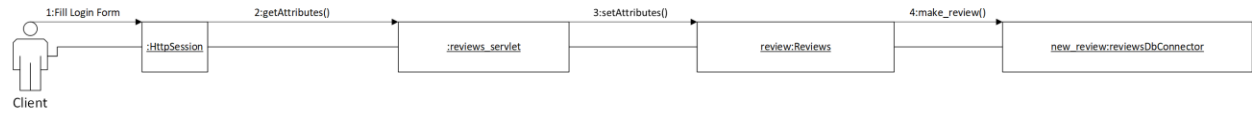
UML COLLABORATION DIAGRAMM FOR
OWNERS REGISTER PHASE 2 4.1.4



UML COLLABORATION DIAGRAMM FOR
RESTAURANT PHASE 2 4.1.4

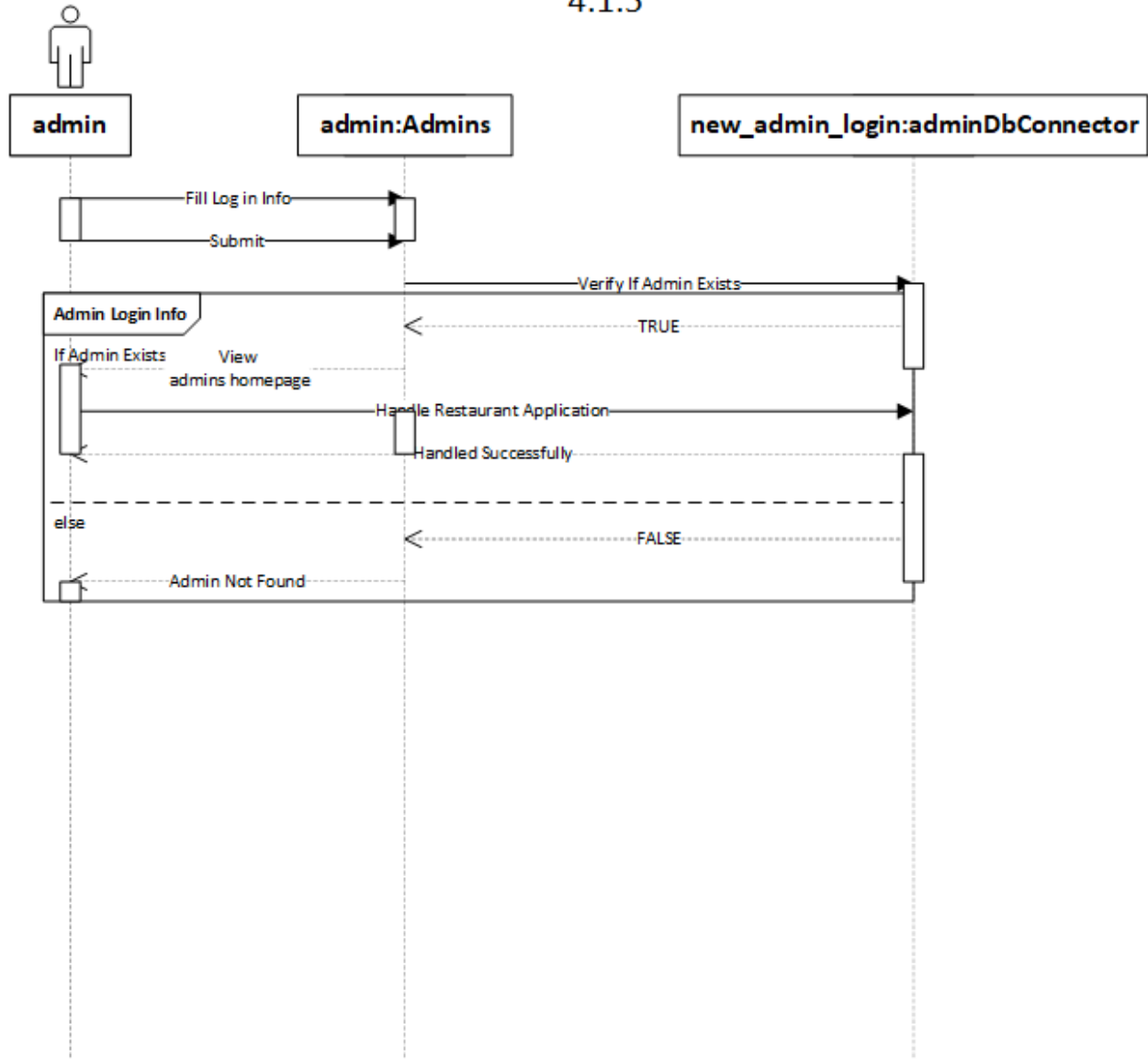


UML COLLABORATION DIAGRAMM FOR
REVIEWS PHASE 2 4.1.4

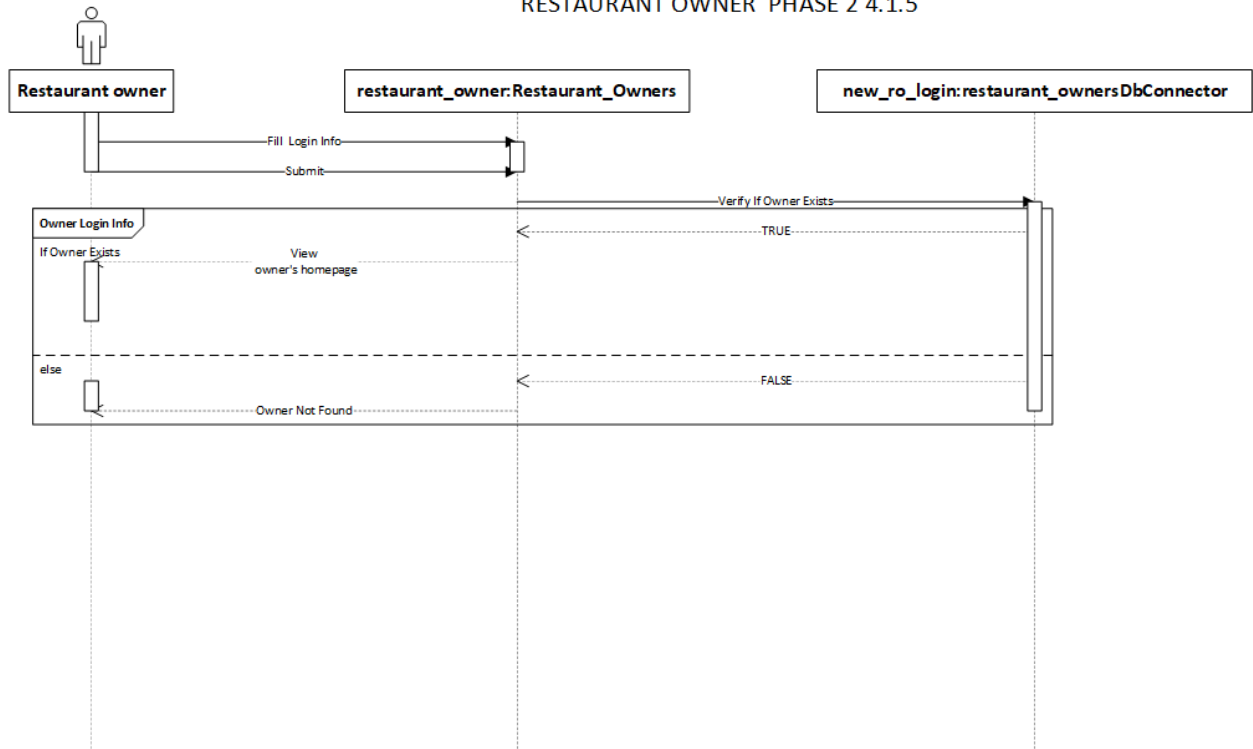


4.1.5

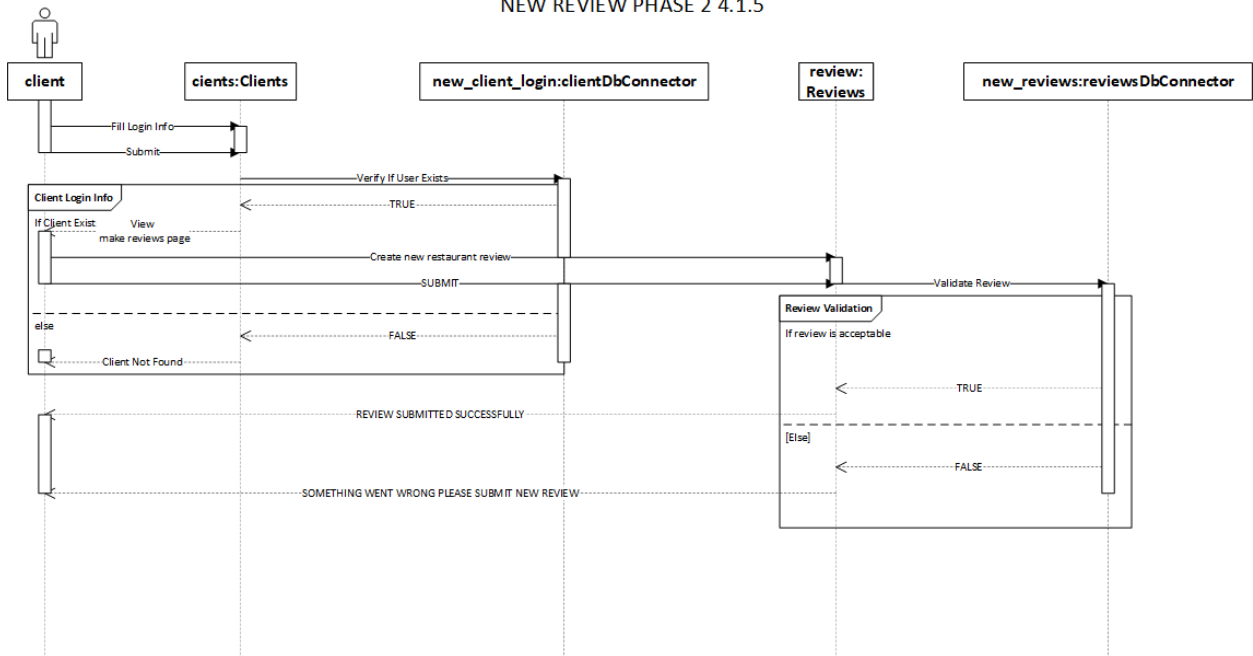
UML SEQUENCE DIAGRAMM FOR ADMIN PHASE 2 4.1.5



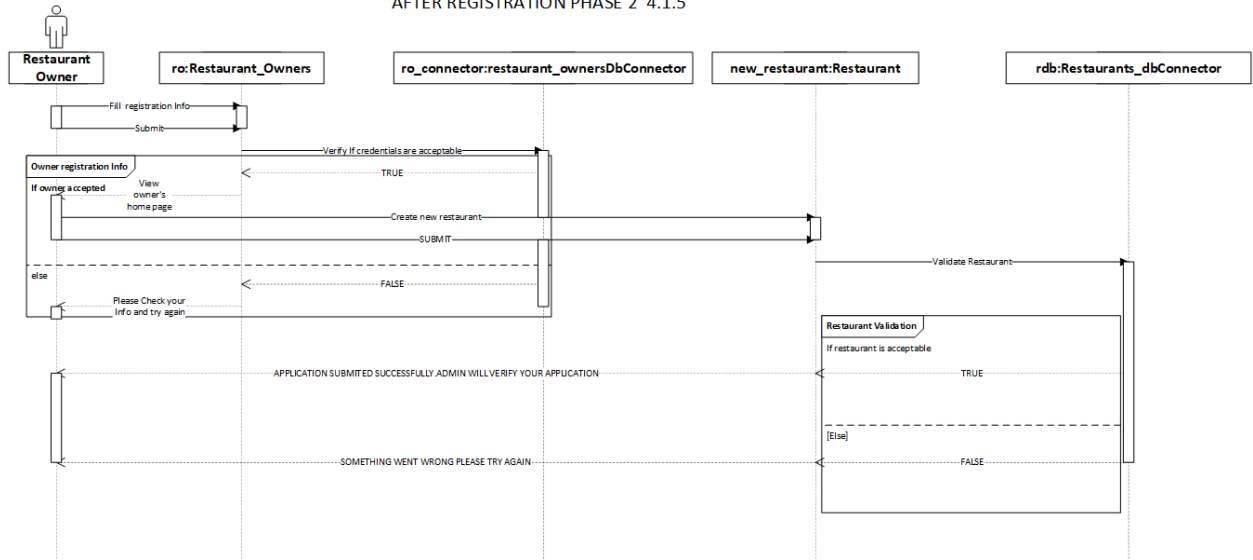
UML SEQUENCE DIAGRAMM FOR EXISTING RESTAURANT OWNER PHASE 2 4.1.5



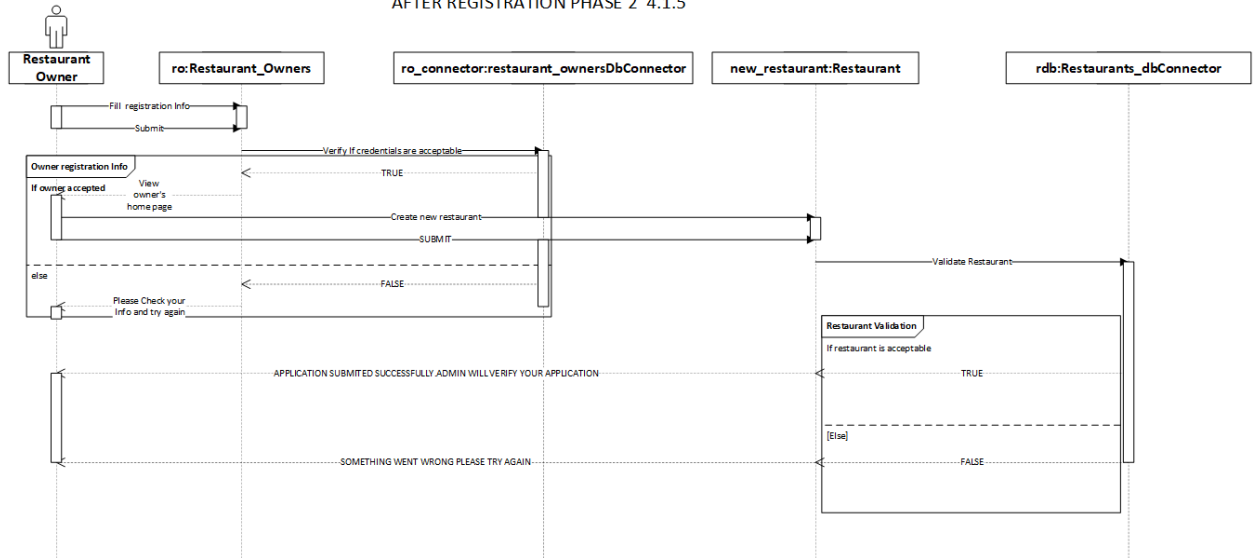
UML SEQUENCE DIAGRAMM FOR EXISTING CLIENT NEW REVIEW PHASE 2 4.1.5



UML SEQUENCE DIAGRAMM FOR NEW REVIEW
AFTER REGISTRATION PHASE 2 4.1.5

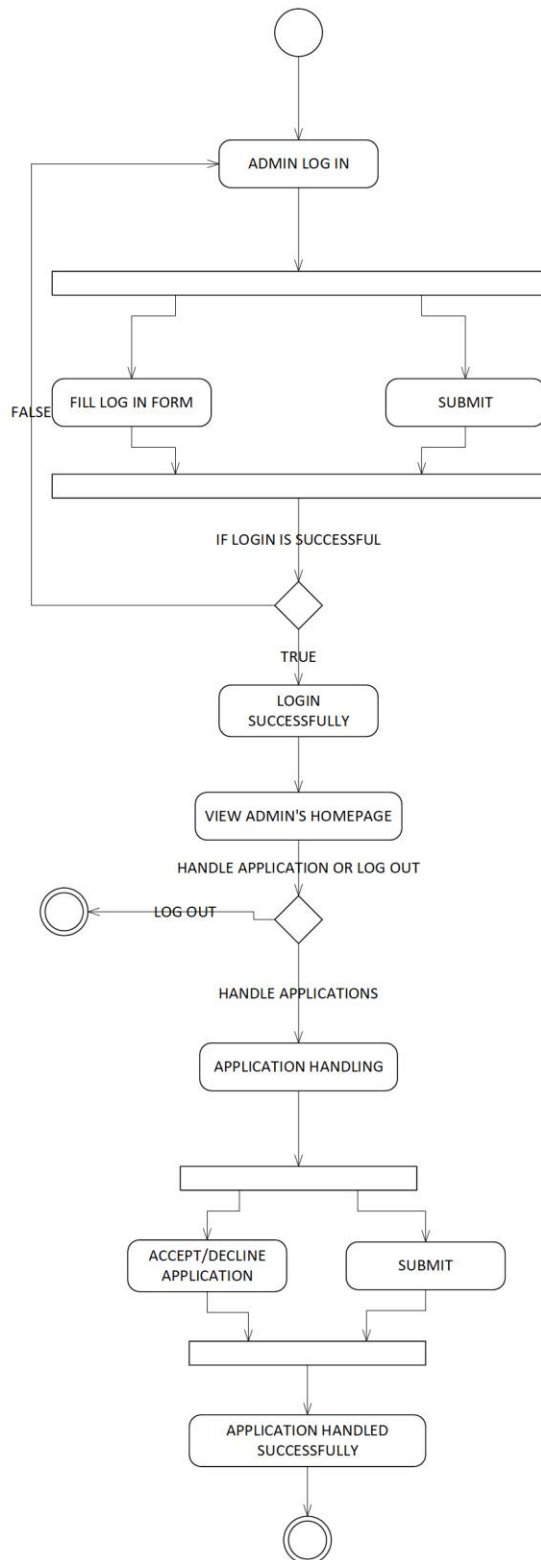


UML SEQUENCE DIAGRAMM FOR NEW REVIEW
AFTER REGISTRATION PHASE 2 4.1.5

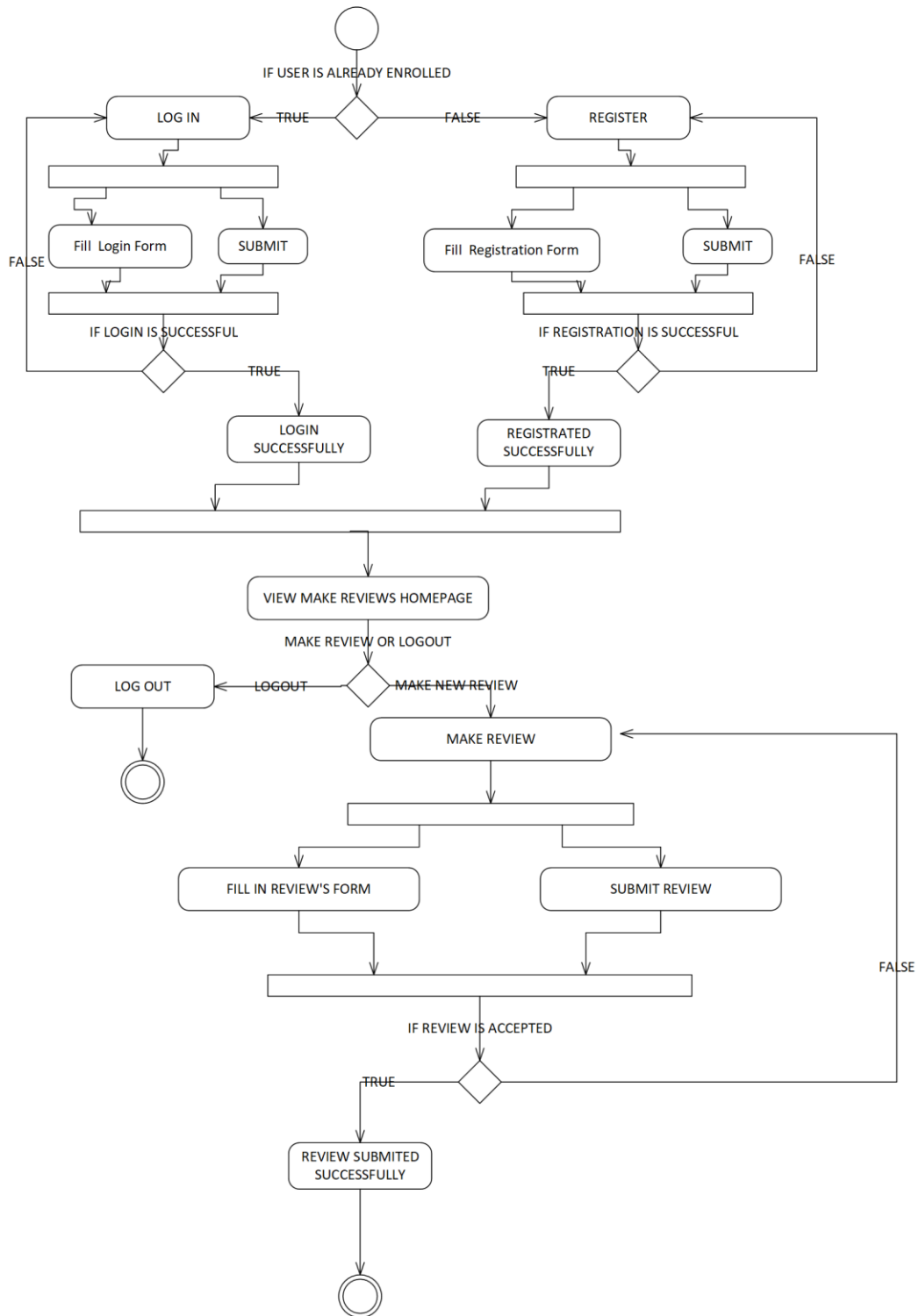


4.1.6

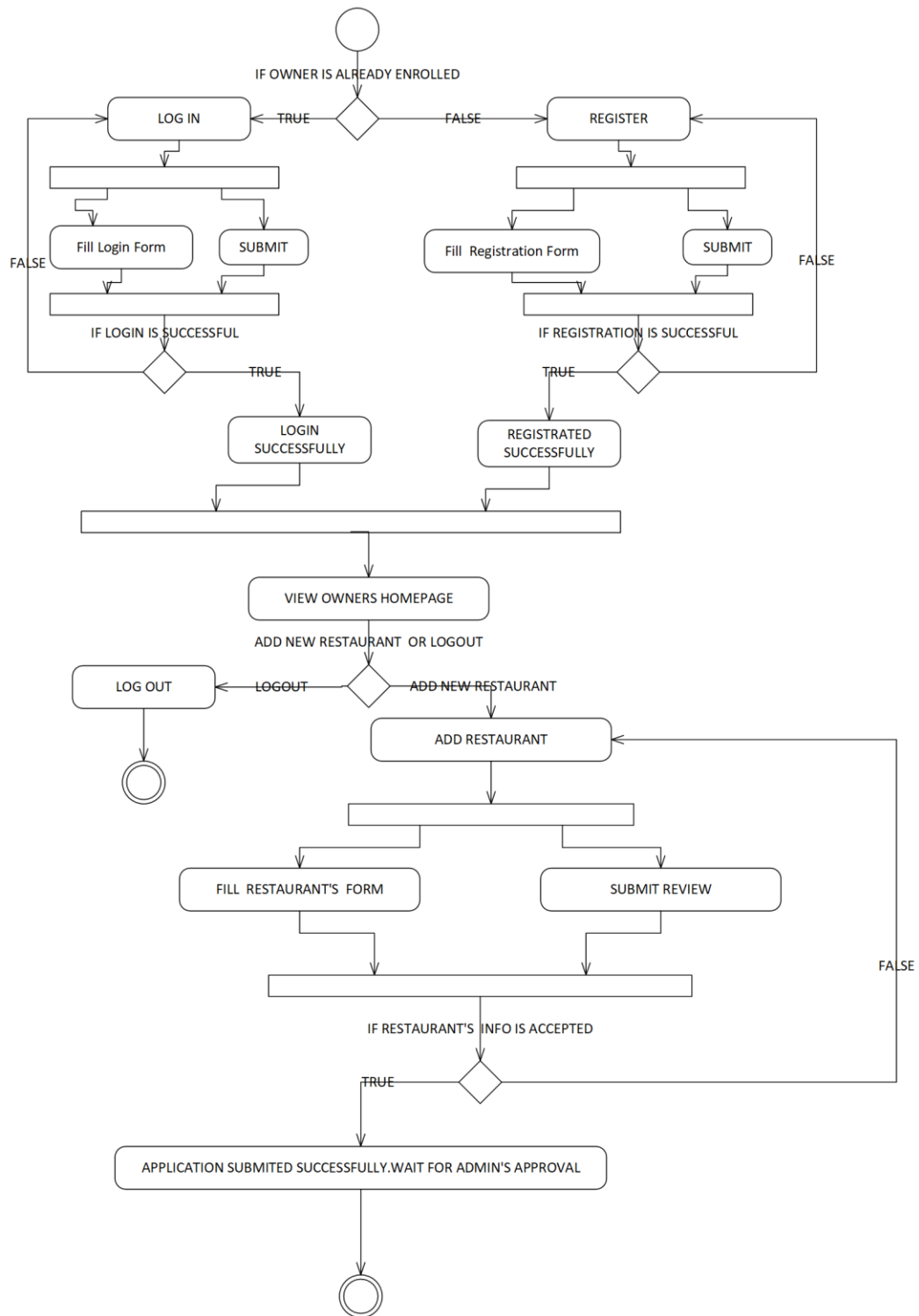
UML ADMIN ACTIVITY DIAGRAM PHASE 2 4.1.6



UML ACTIVITY DIAGRAMM CLIENT PHASE 2 4.1.6

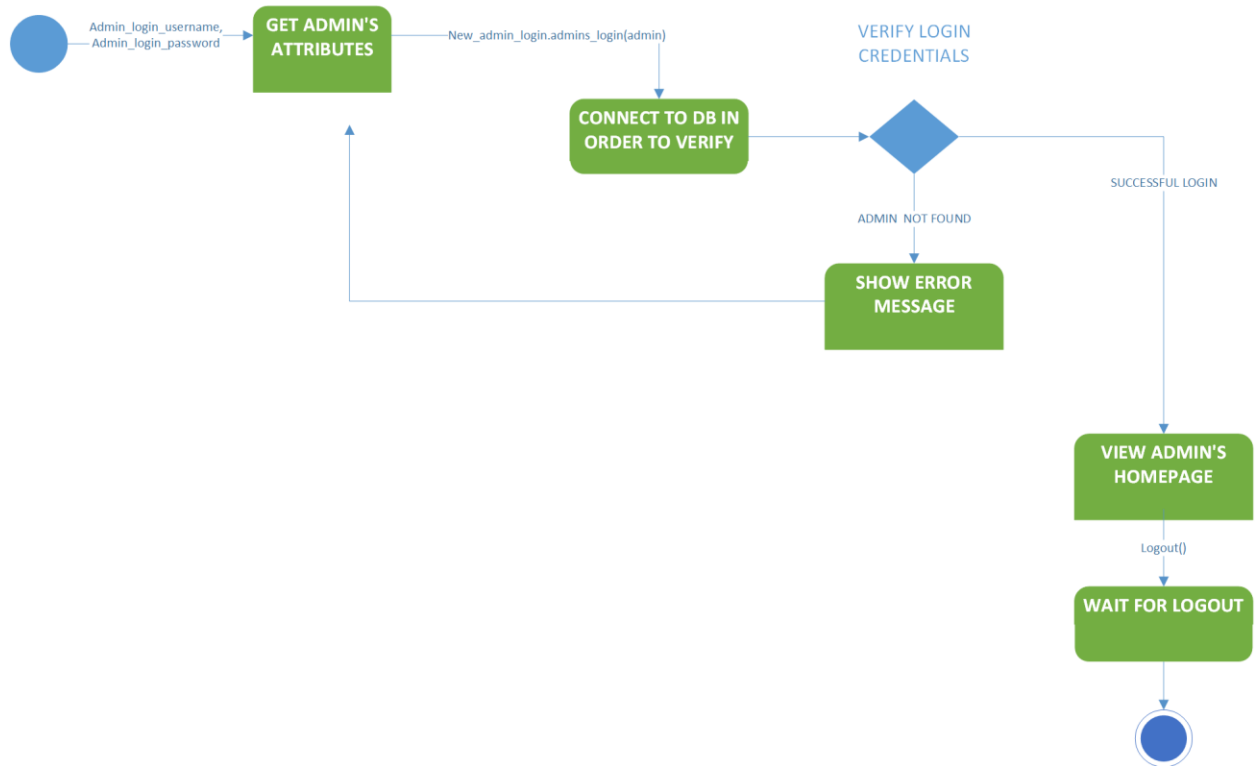


UML ACTIVITY DIAGRAMM RESTAURANT OWNER PHASE 2
4.1.6

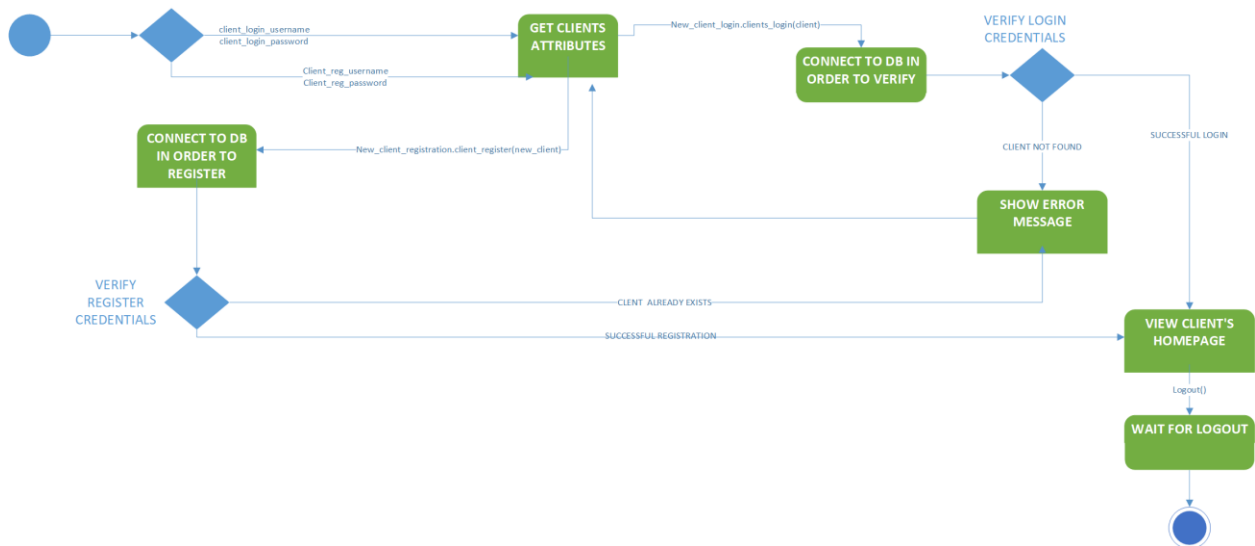


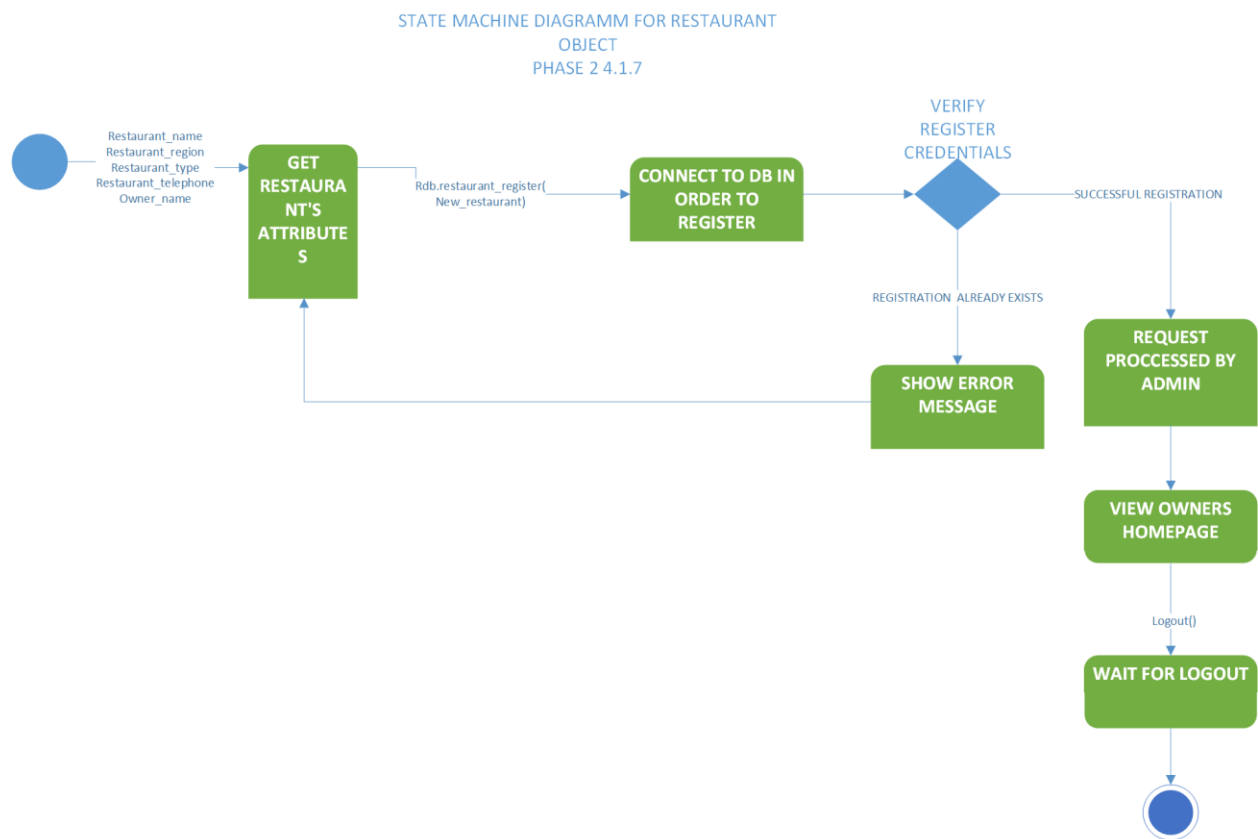
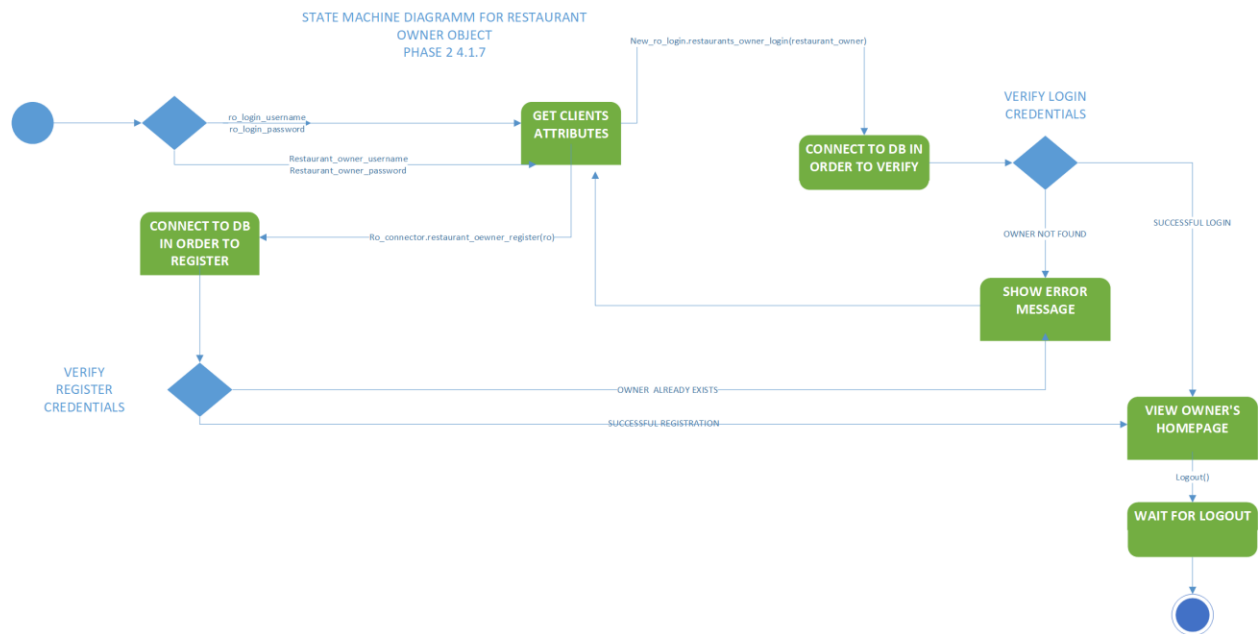
4.1.7

STATE MACHINE DIAGRAMM FOR ADMIN OBJECT
PHASE 2 4.1.7

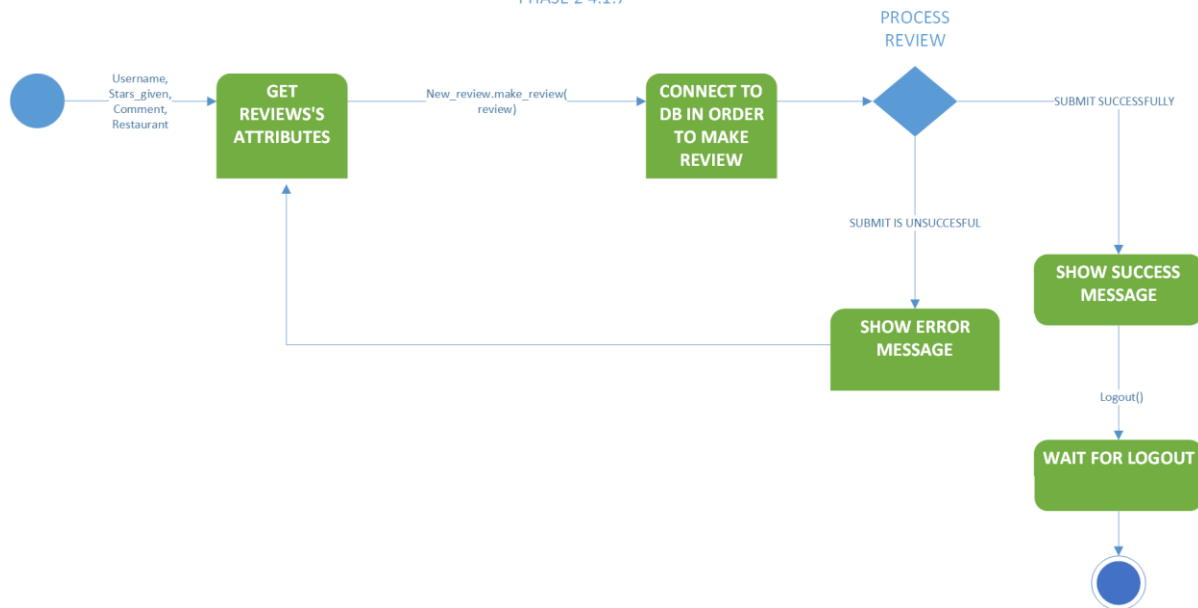


STATE MACHINE DIAGRAMM FOR CLIENT OBJECT
PHASE 2 4.1.7



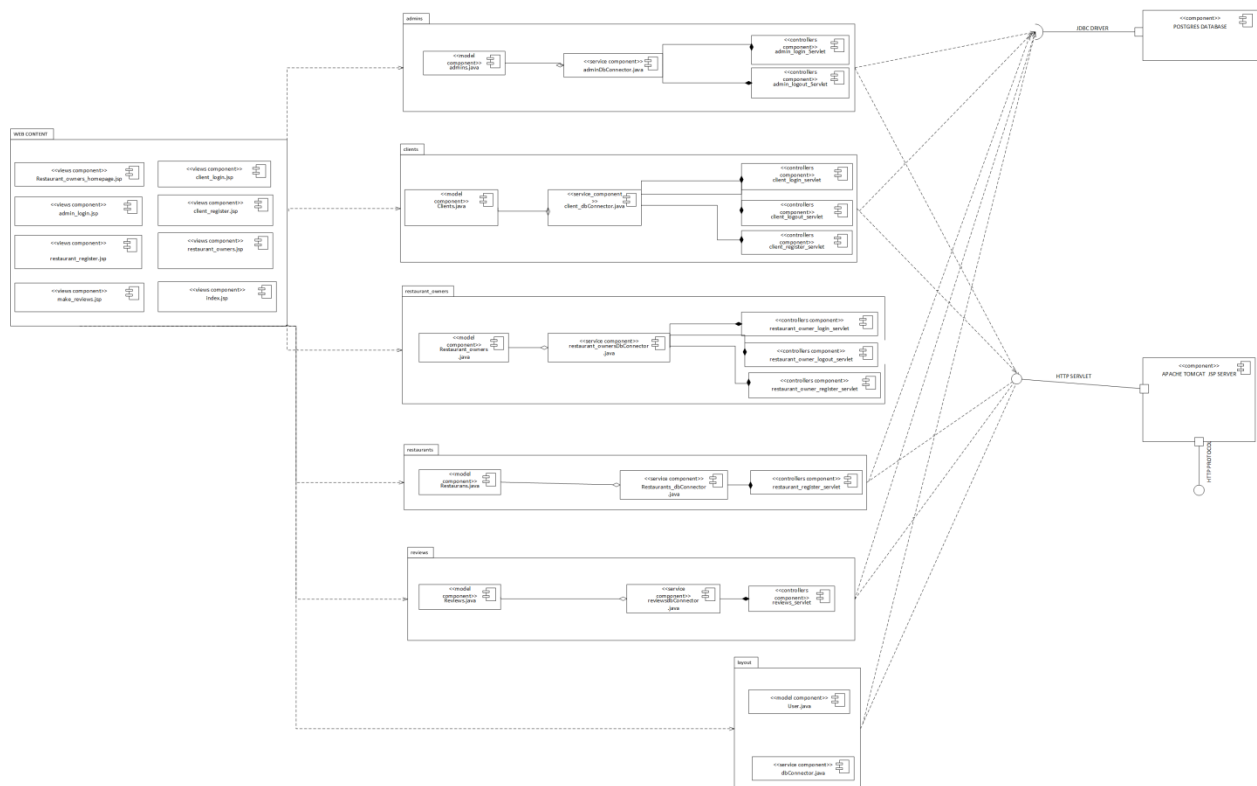


STATE MACHINE DIAGRAMM FOR REVIEWS OBJECT
PHASE 2 4.1.7

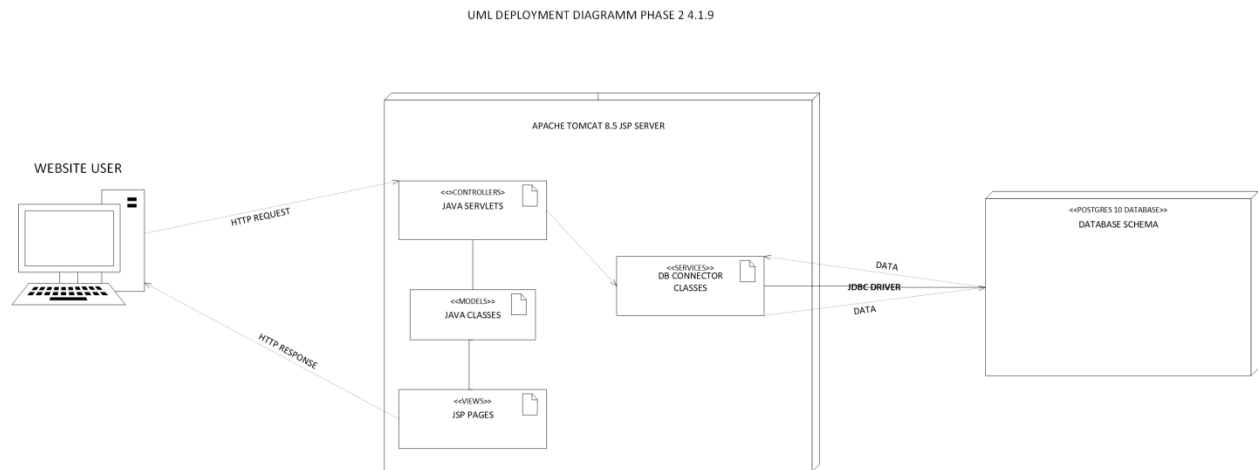


4.1.8

UML Component Diagramm Phase 2
4.1.8



4.1.9



4.2 Υλοποίηση και έλεγχος

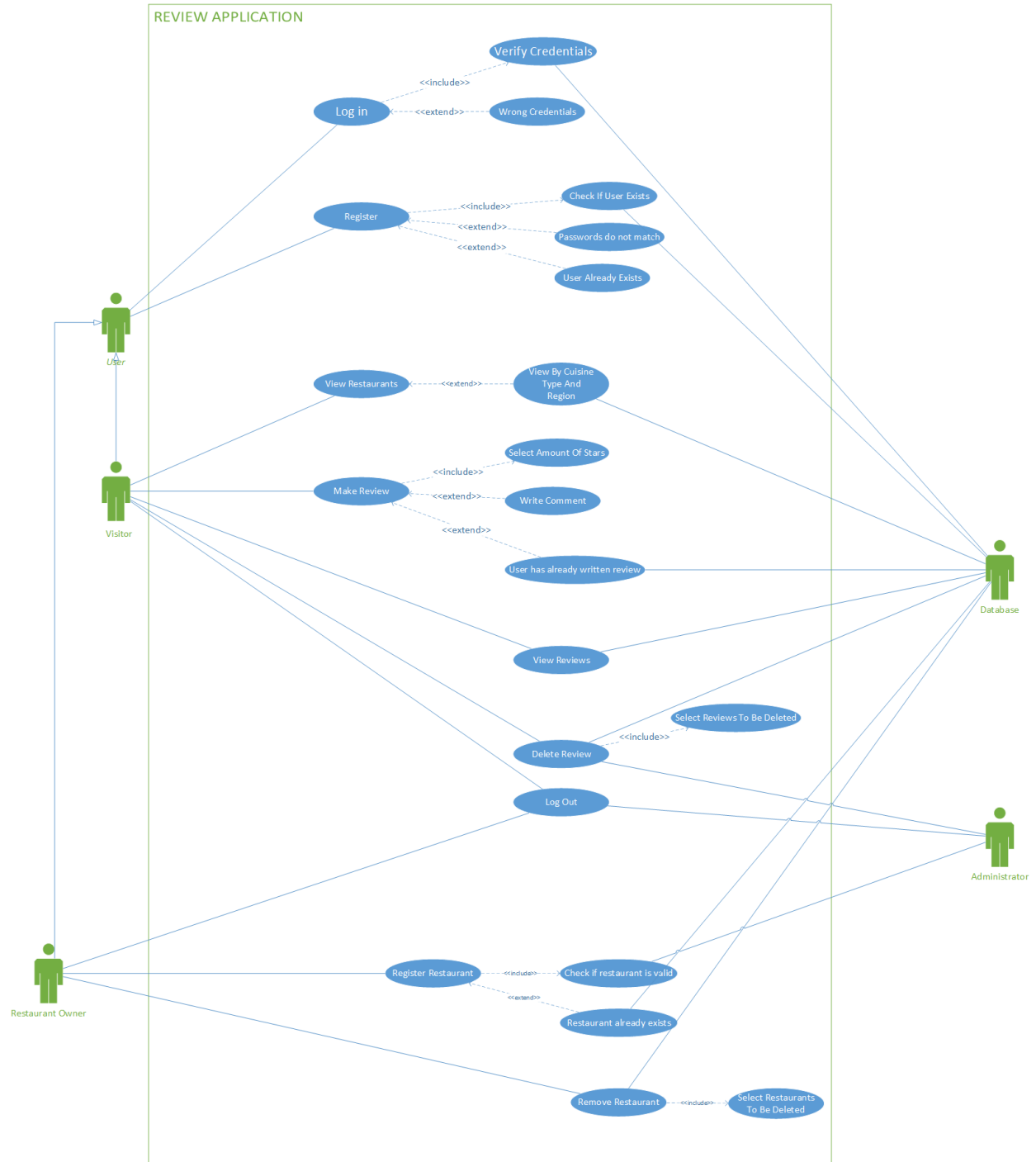
4.2.2

Στην πρώτη εκτελέσιμη έκδοση περιλαμβάνονται οι αρχικές απαιτήσεις δηλαδή αυτές της εμφάνισης εστιατορίου βάση του καλύτερο μέσο όρο βαθμολογίας, την εγγραφή νέου χρήστη, νέου ιδιοκτήτη και την εισαγωγή νέου εστιατορίου αλλά και την αποδοχή ή απορριψη αιτημάτων από τον διαχειριστή.

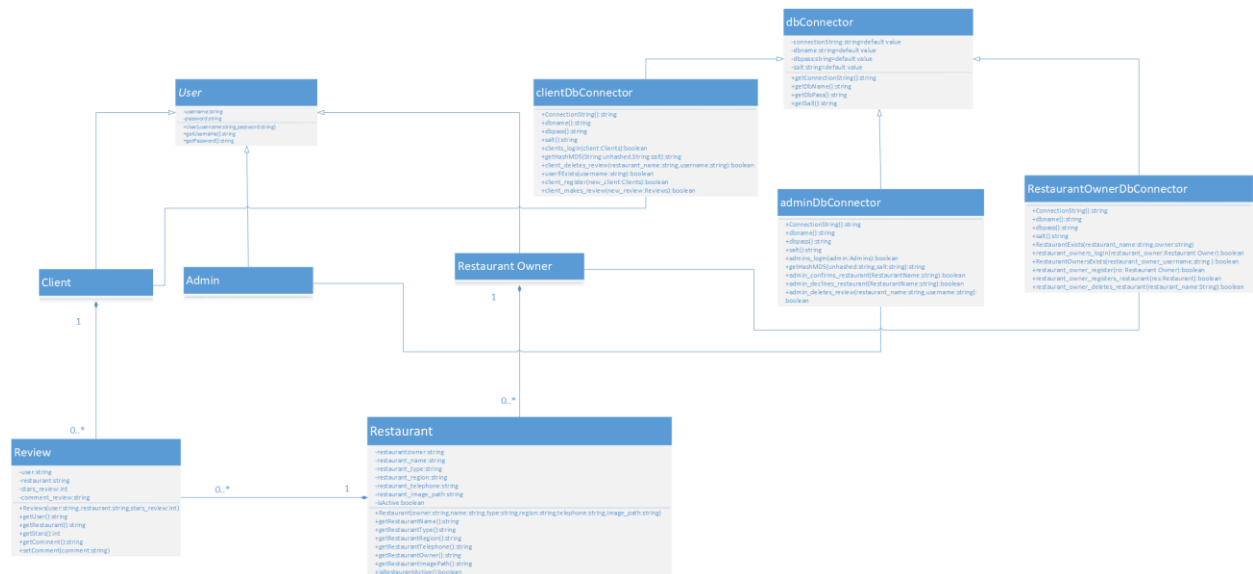
5. Φάση: Κατασκευή: (Construction)

5.1

5.1.1

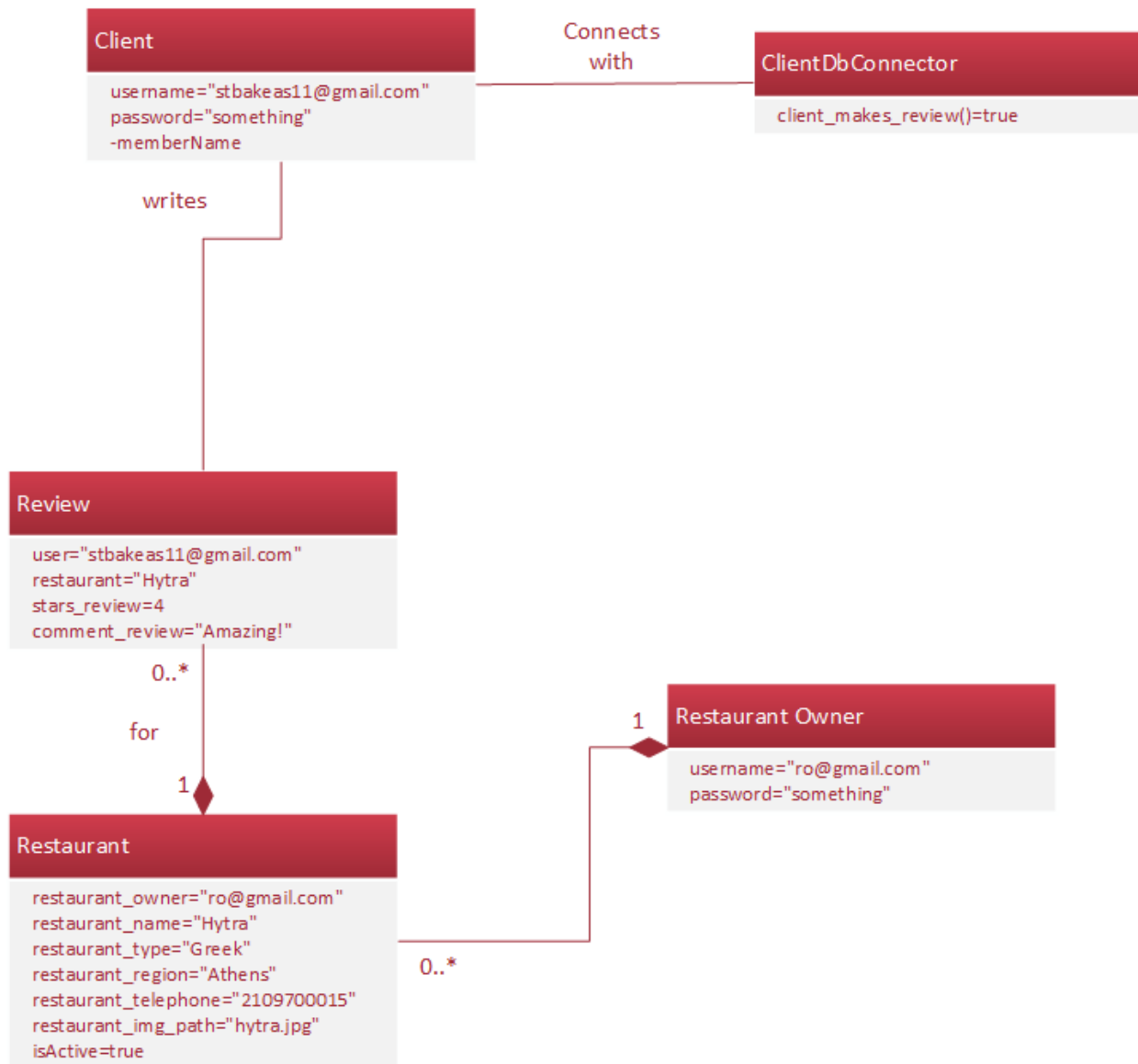


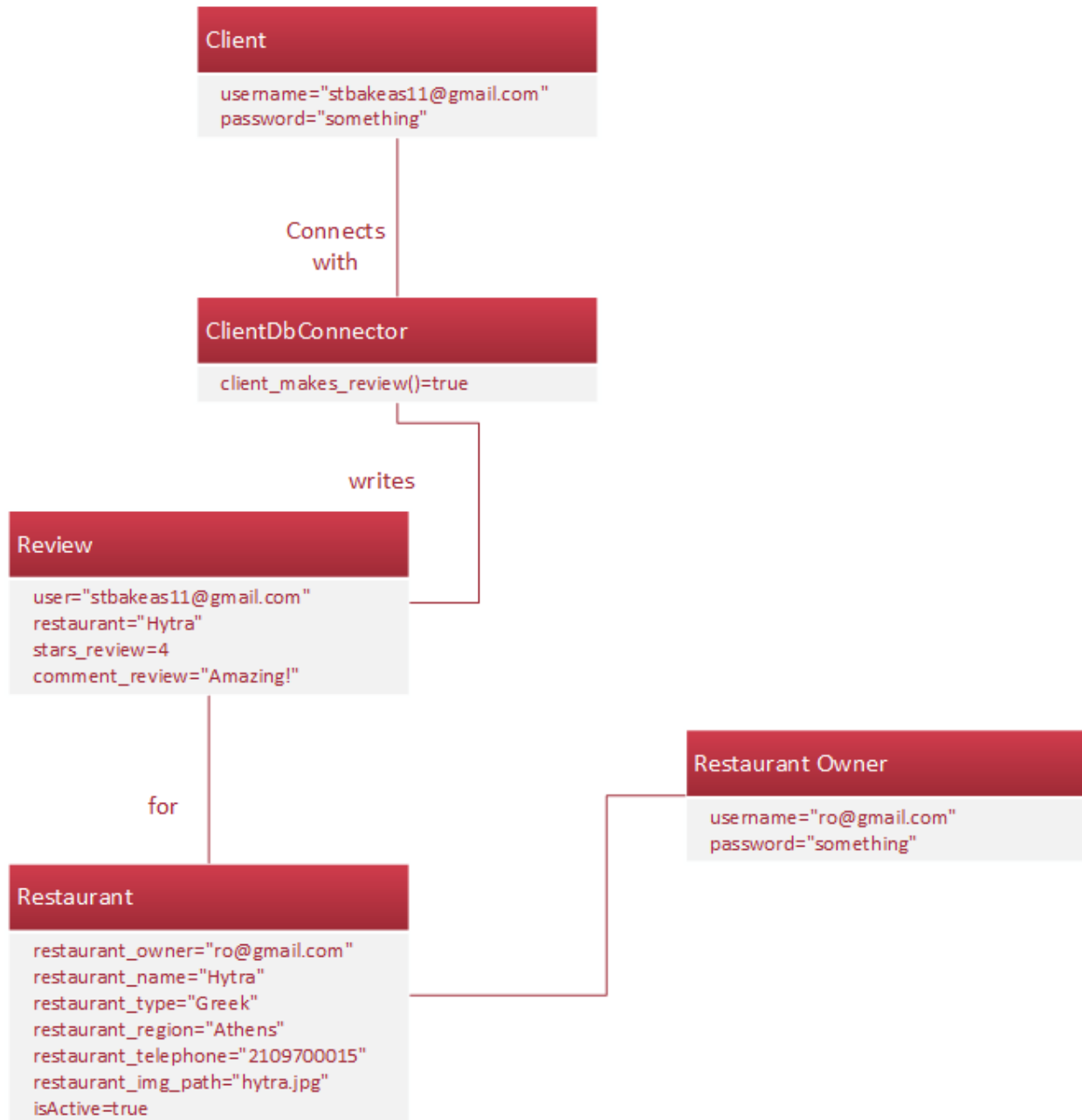
5.1.2



5.1.3

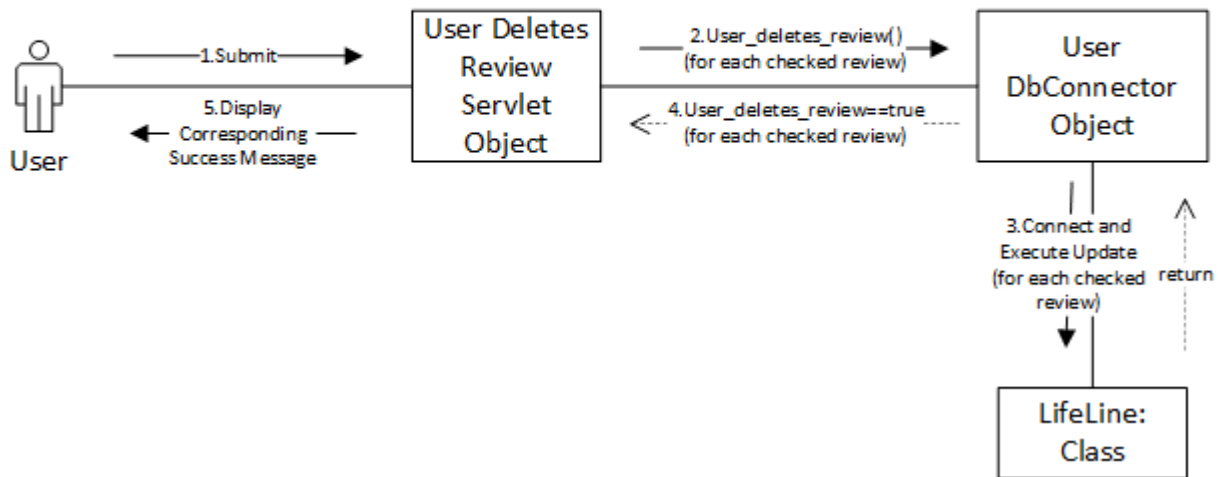




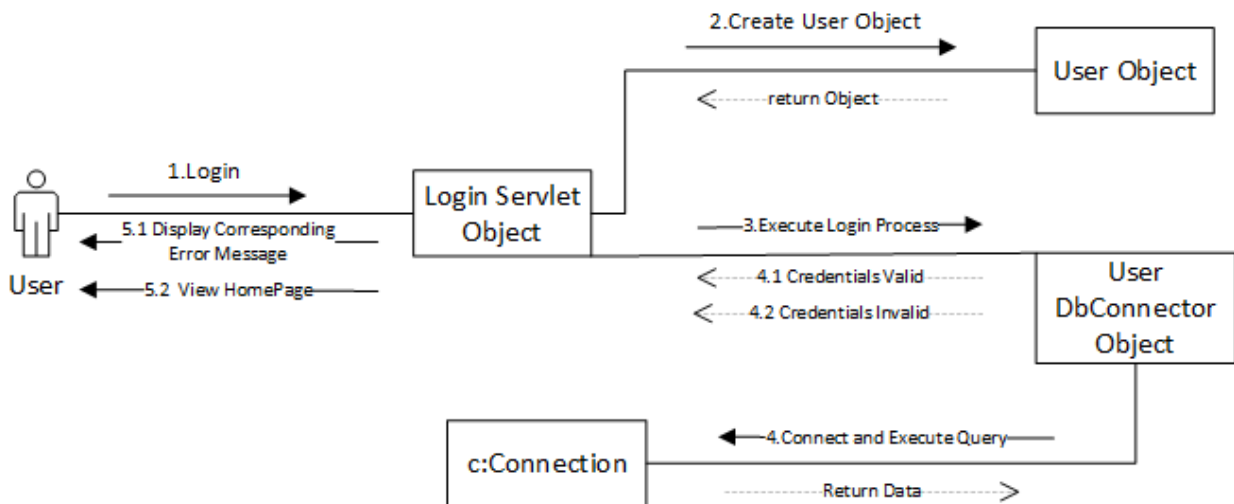


5.1.4

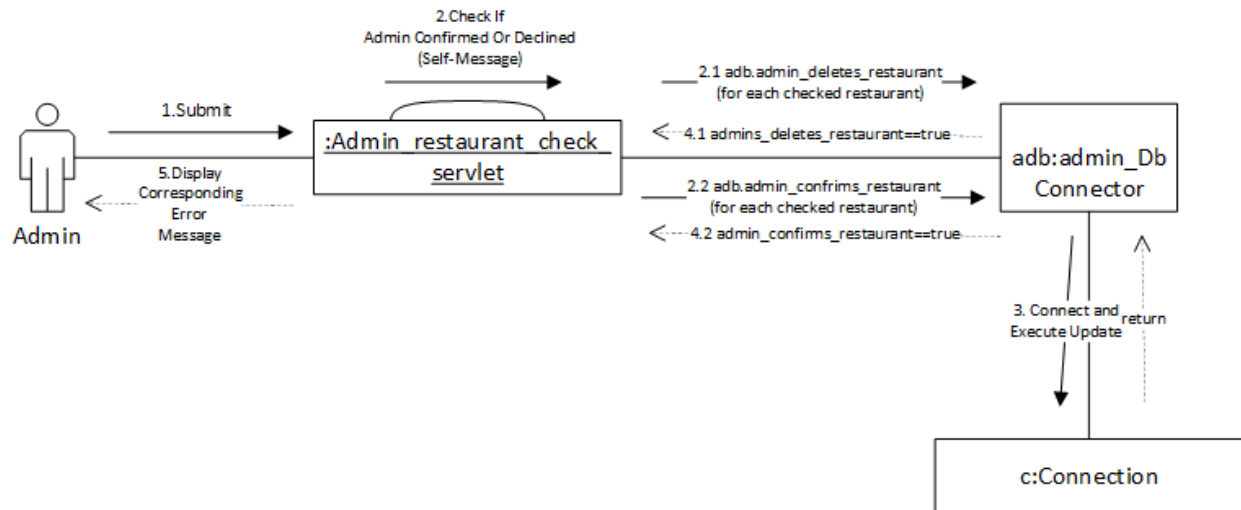
Collaboration Diagram for Deletion of Review.User represents Admin and Client



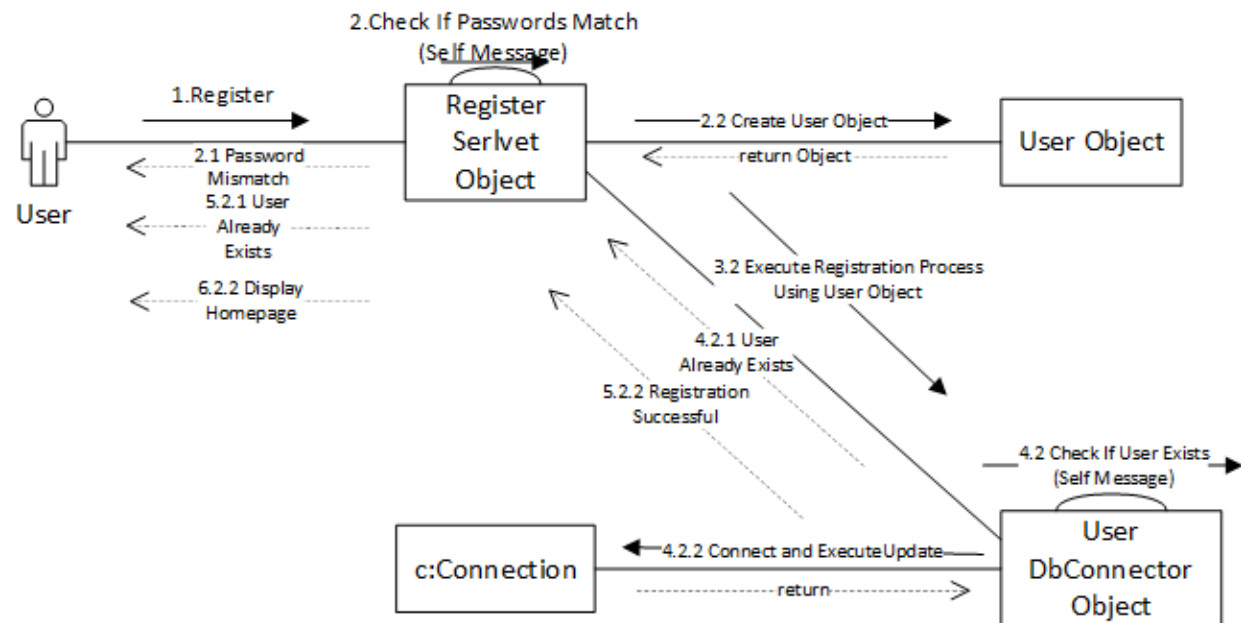
Collaboration Diagram of Login Process for Every User(Admin,Restaurant Owner,User)



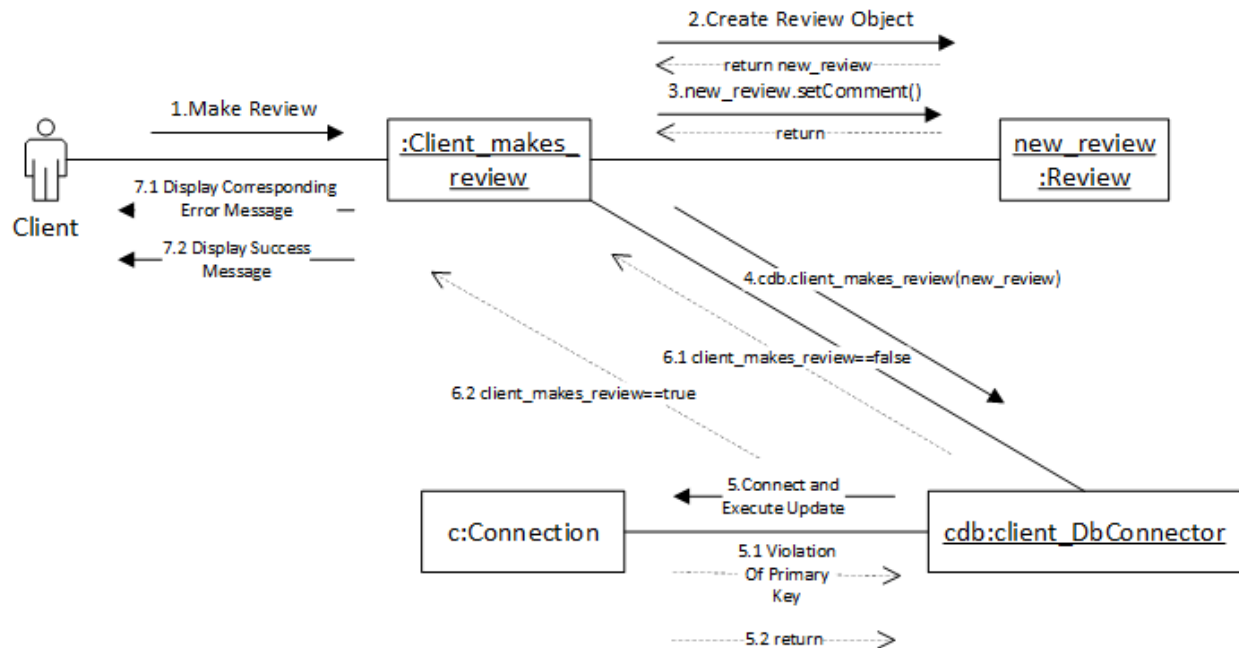
COLLABORATION DIAGRAM OF REGISTRATION HANDLING FOR ADMIN



Collaboration Diagram of Registration Process For Client And Restaurant Owner.

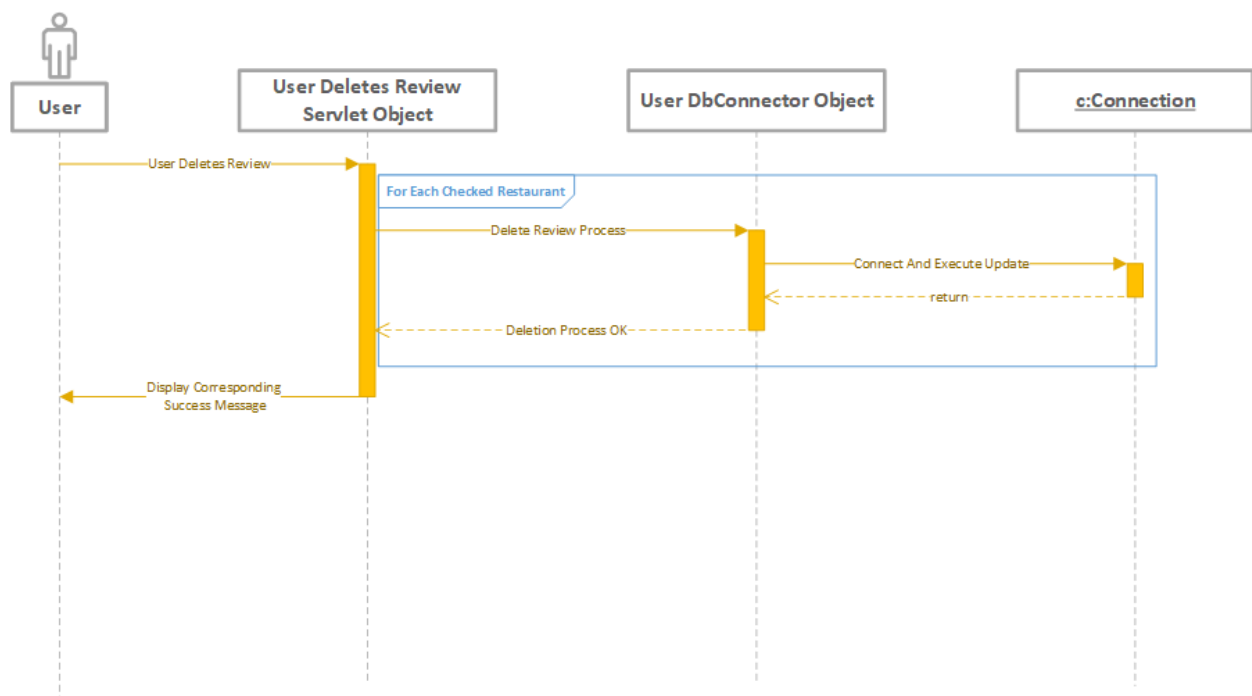


Collaboration Diagram for Review of User

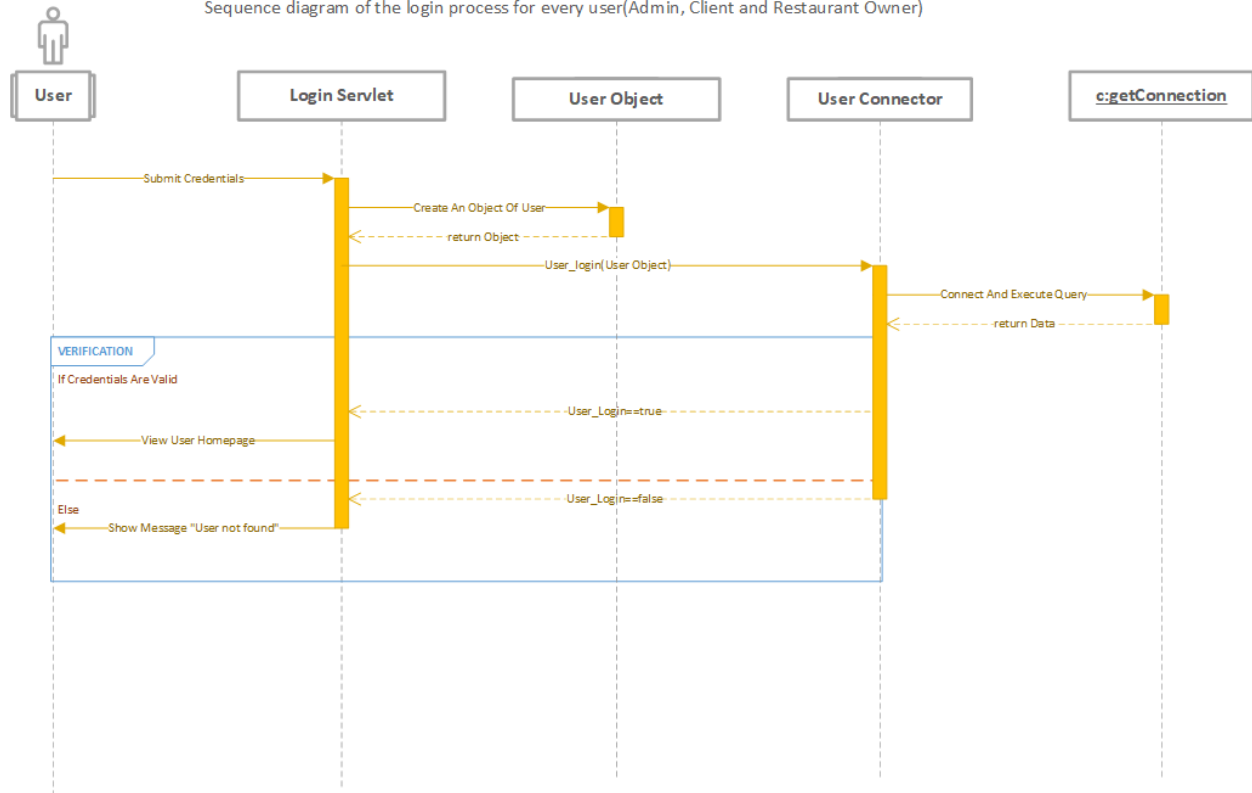


5.1.5

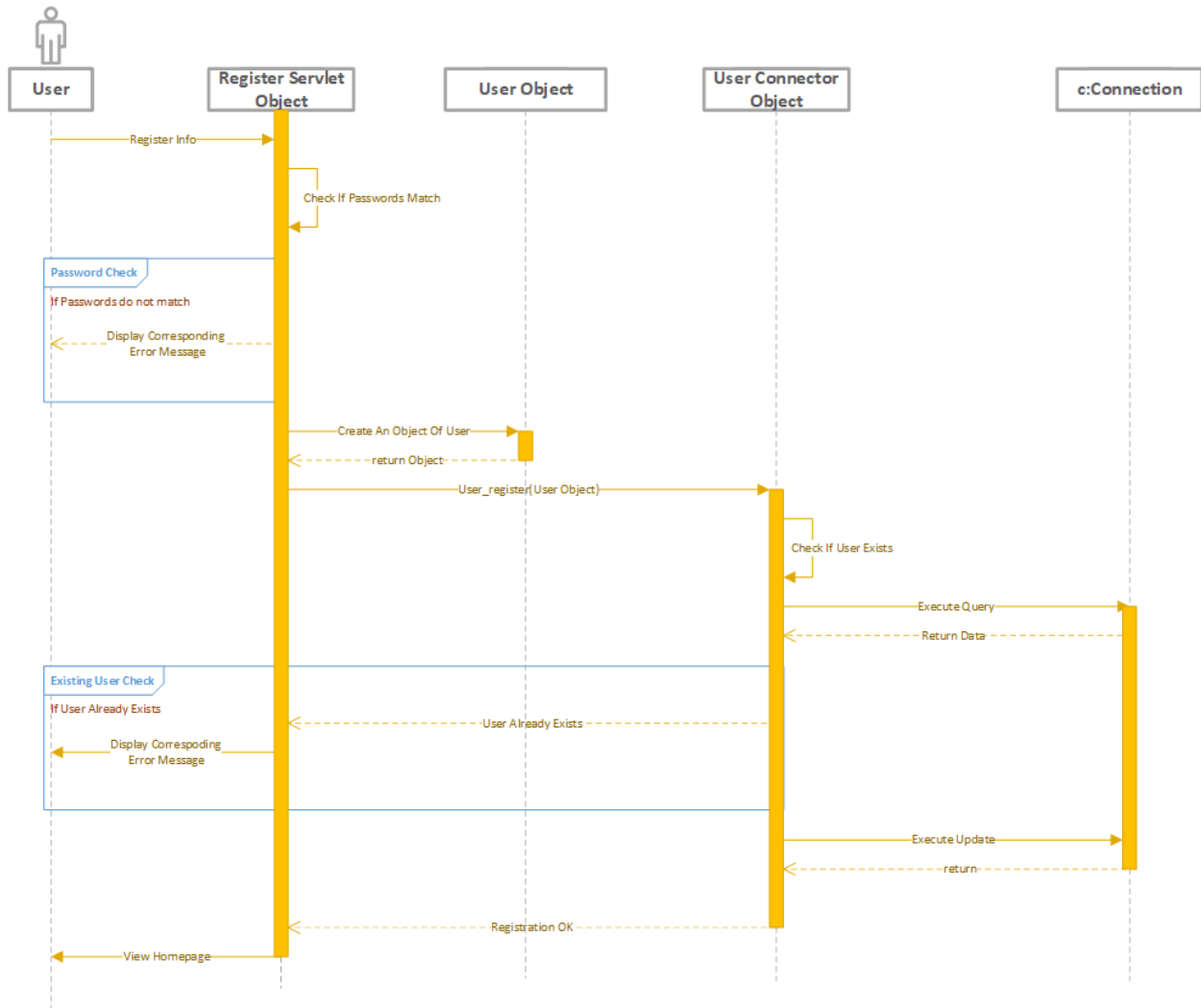
Sequence Diagram of Deleting A Review. For Clients And Admins



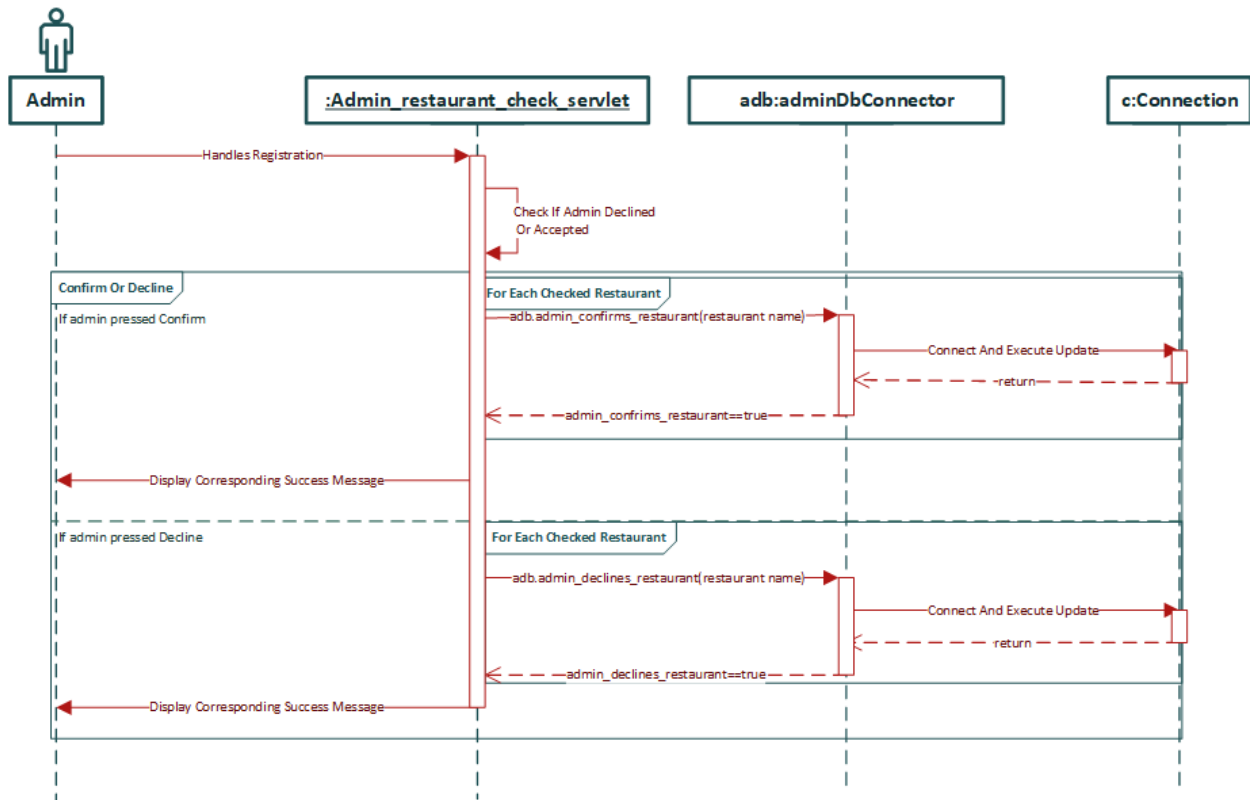
Sequence diagram of the login process for every user(Admin, Client and Restaurant Owner)



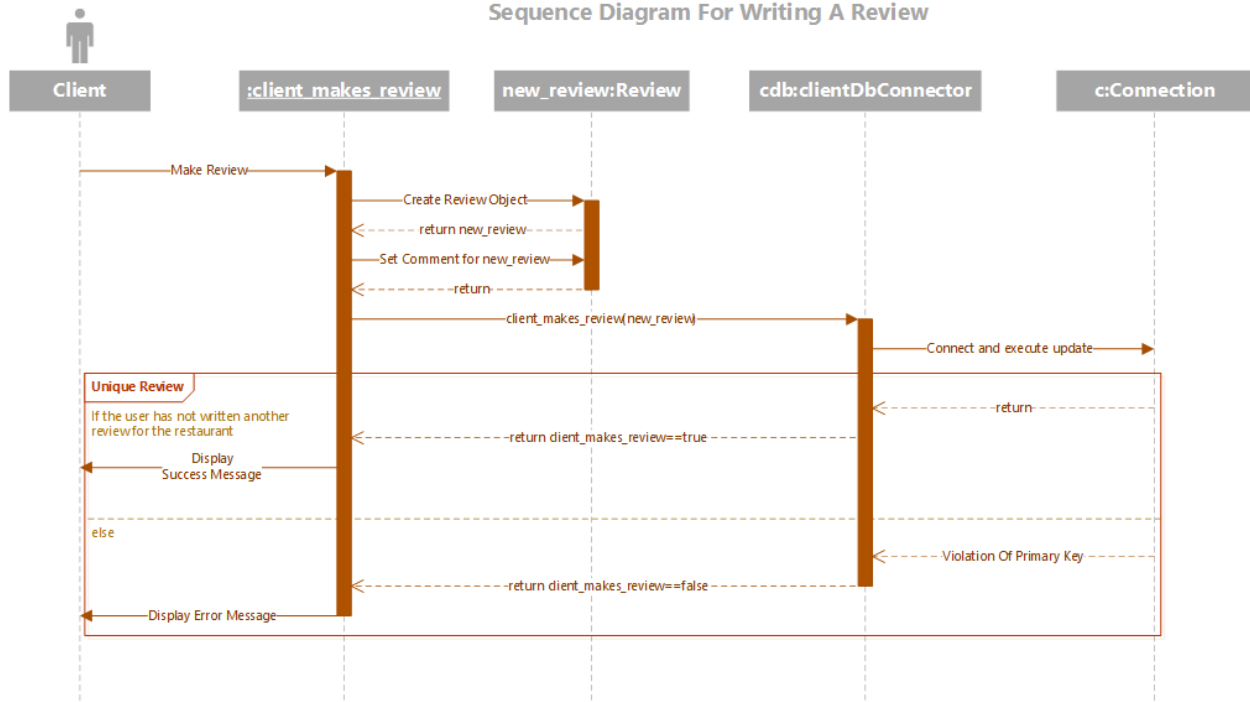
Sequence Diagram For Registration of Client And Restaurant Owner



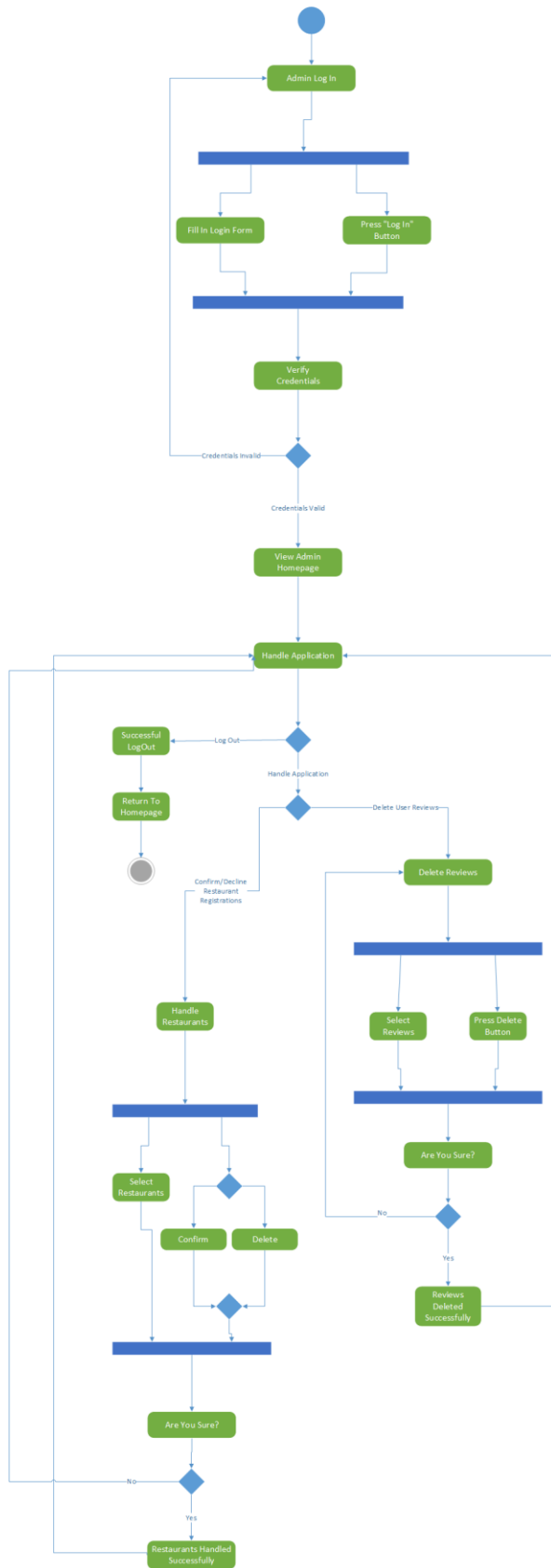
SEQUENCE DIAGRAM FOR CONFIRMING OR REJECTING A RESTAURANT REGISTRATION

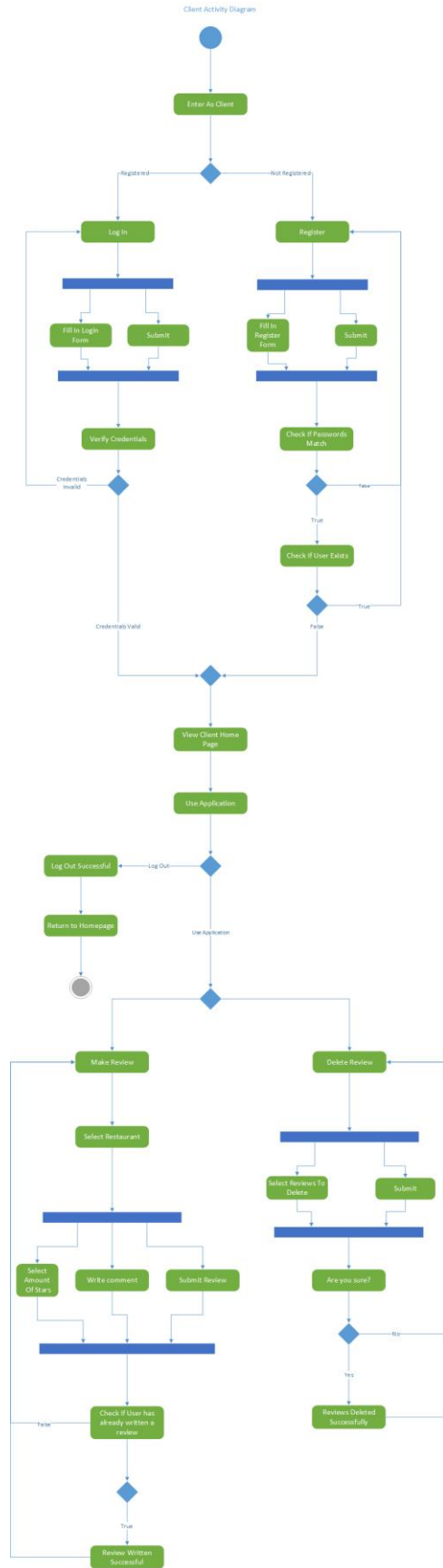


Sequence Diagram For Writing A Review

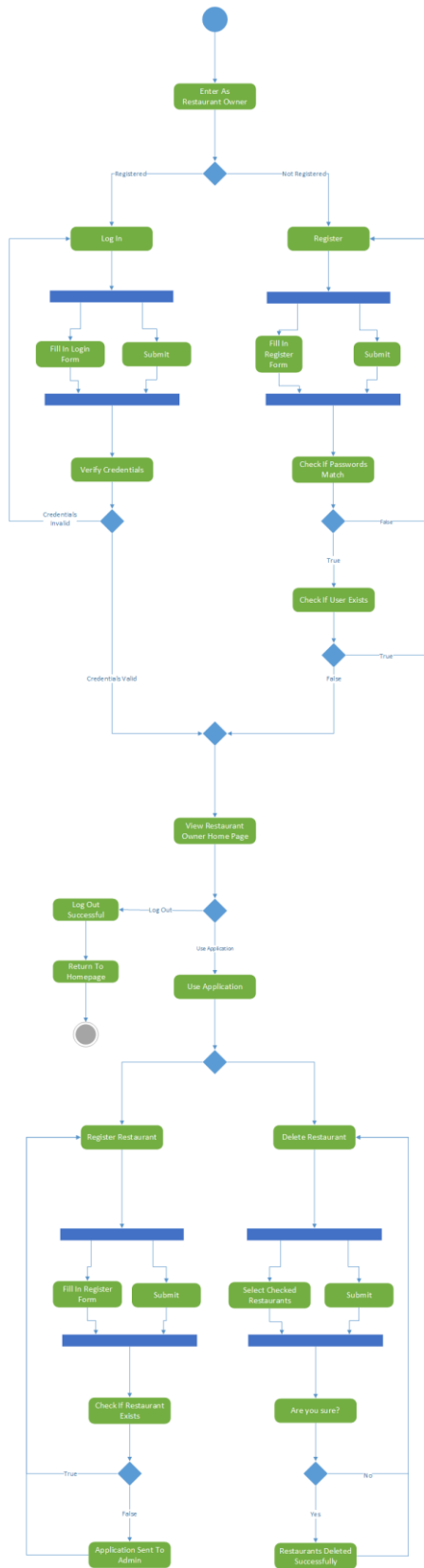


ADMIN ACTIVITY DIAGRAM



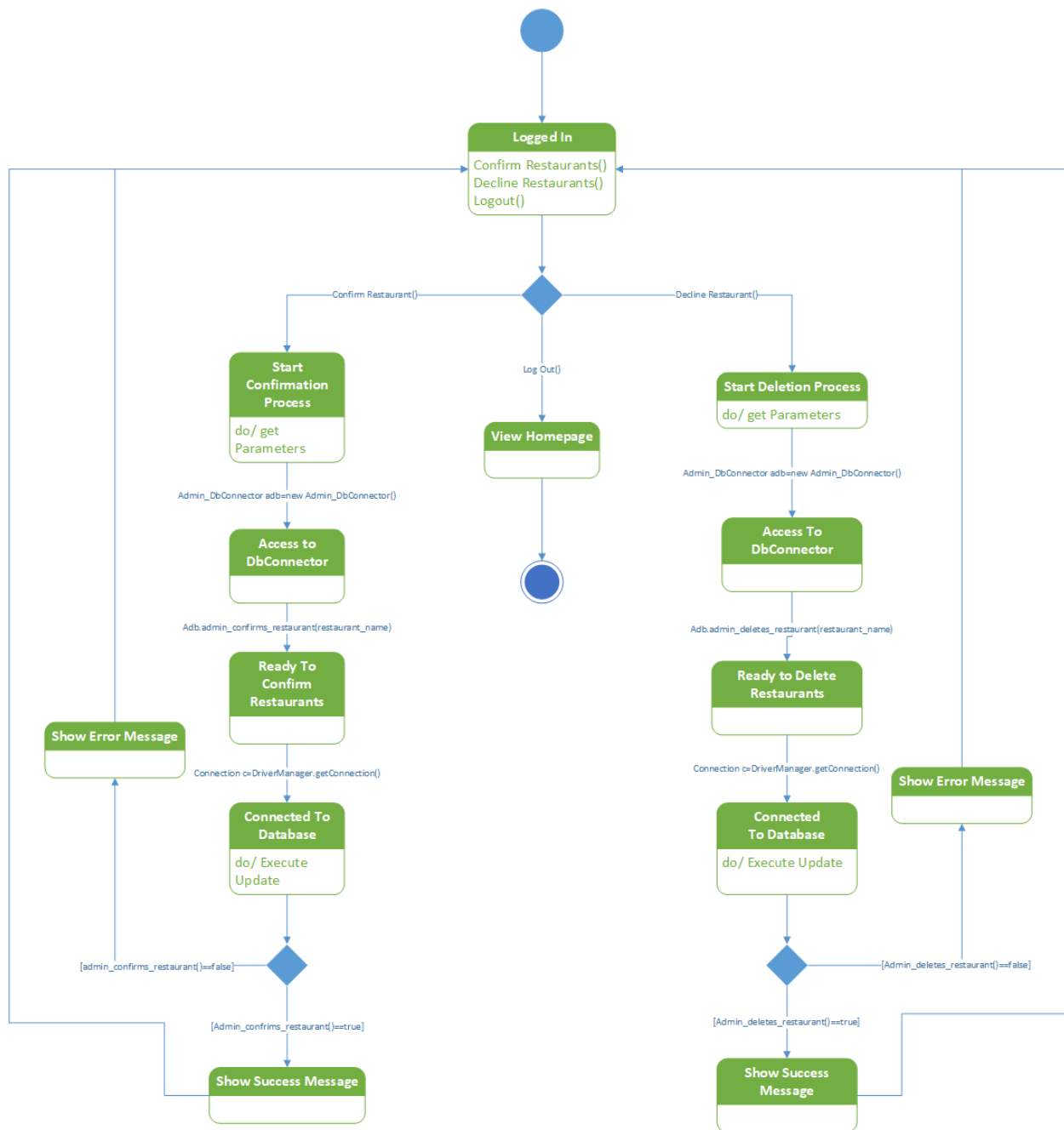


Restaurant Owner Activity Diagram

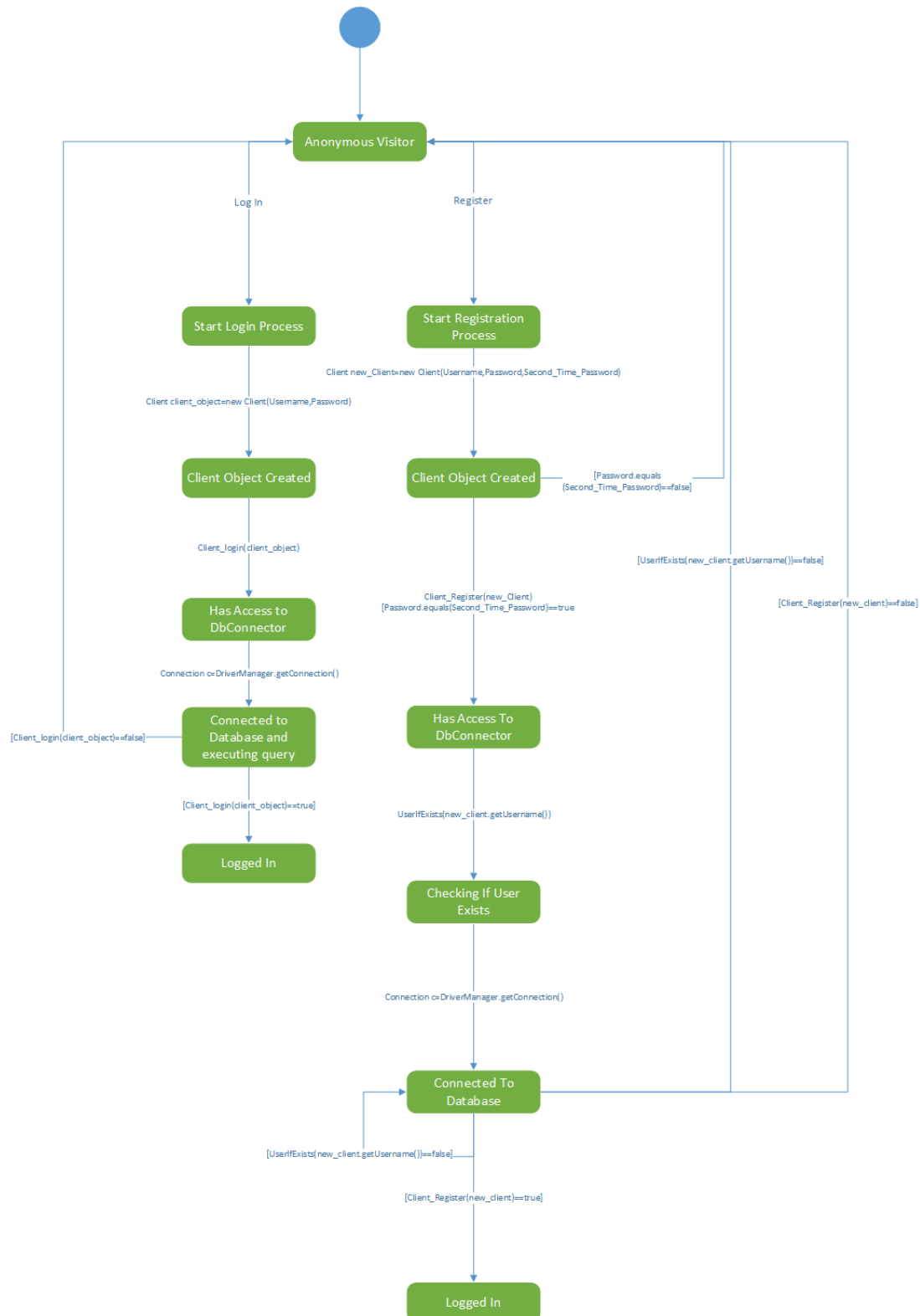


5.1.7

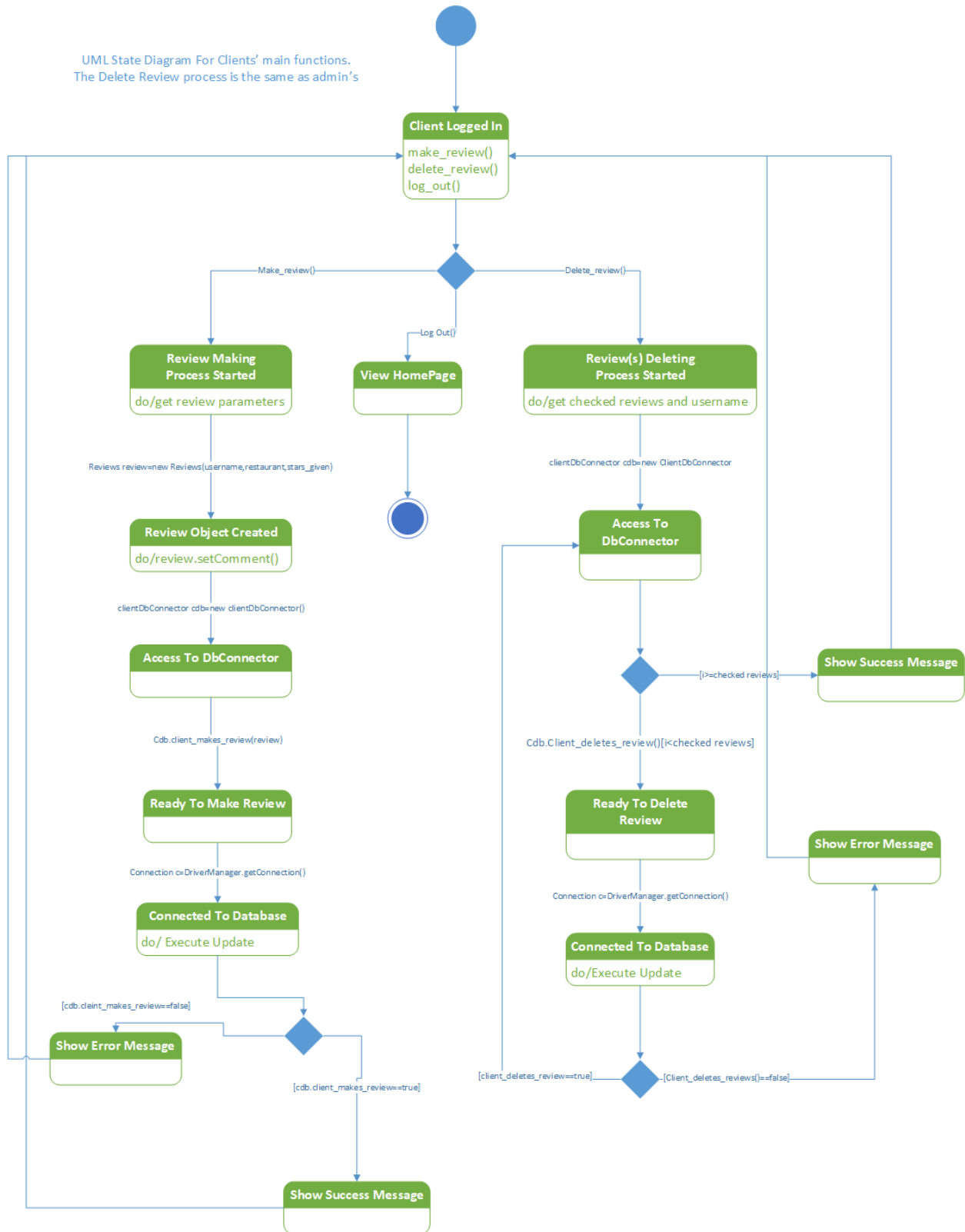
Admin Handles Restaurant Registrations



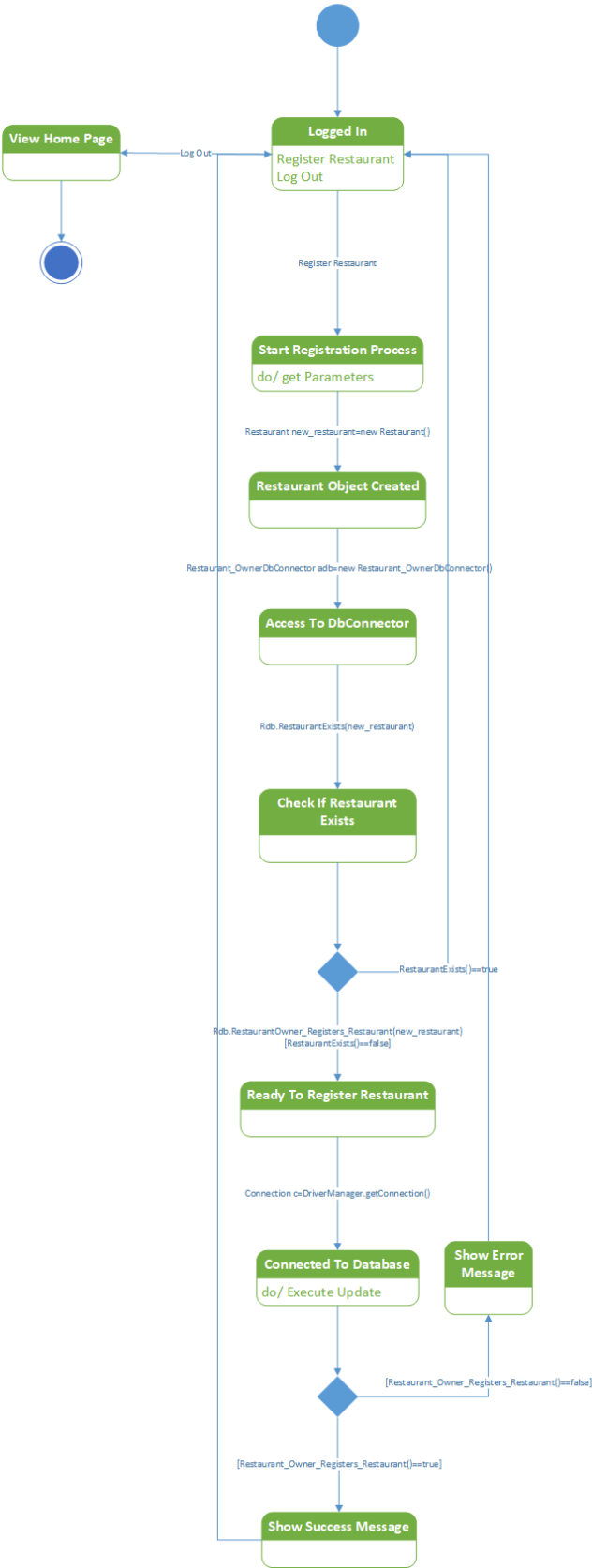
State Diagram Of Login And Register Process(Client Example).
 The Login Process is the same for every type of User and The Registration Process is the same for Restaurant Owner.



UML State Diagram For Clients' main functions.
The Delete Review process is the same as admin's

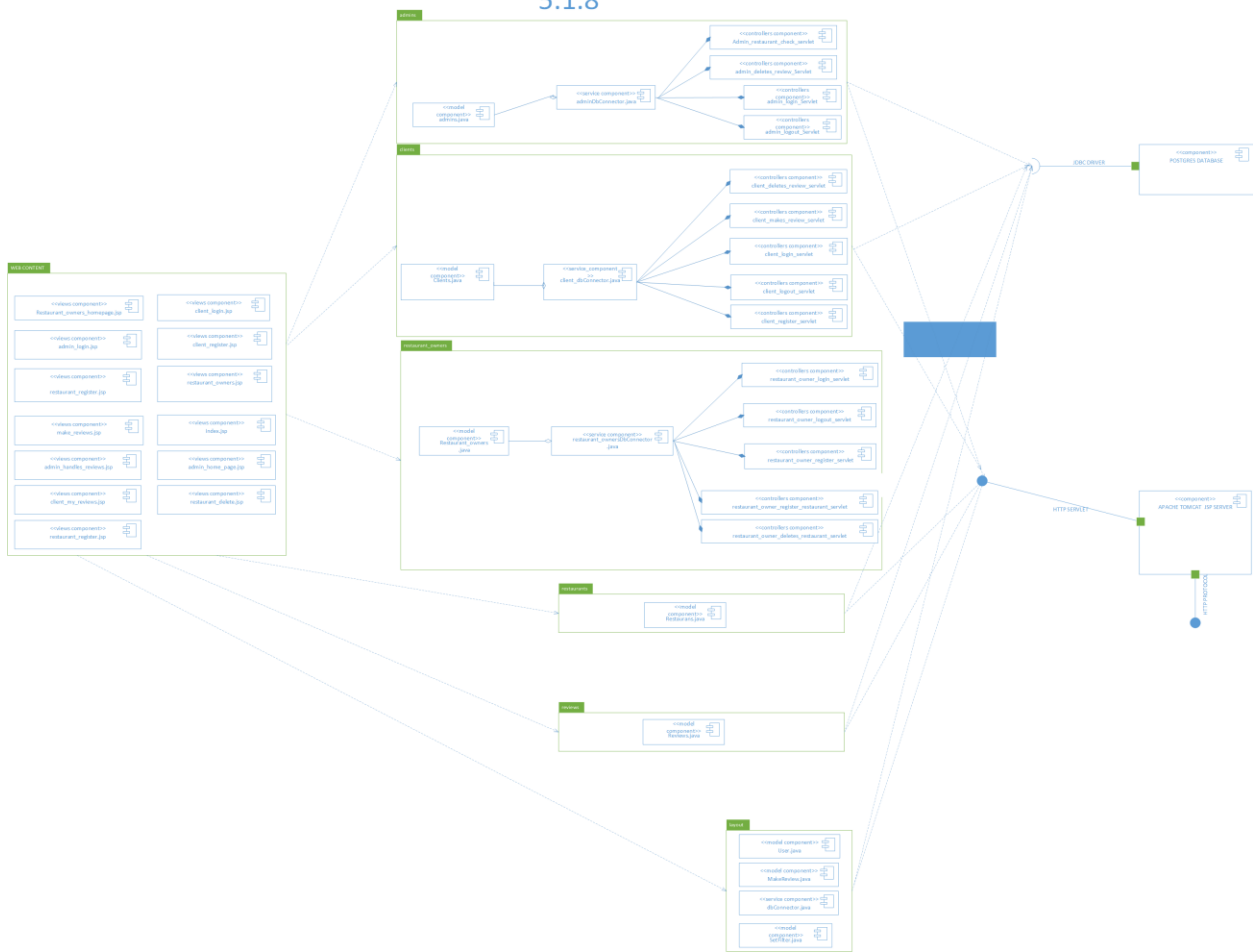


State Diagram Of Restaurant Owner Register



5.1.8

UML Component Diagram Phase 5 5.1.8



5.1.9

UML DEPLOYMENT DIAGRAM PHASE 5 5.1.9

