

Информационна система на агенция за недвижими имоти
Доклад за индивидуална работа по проекта

Александър Велин 71512
(*еким* $\pi \approx 3.1$)

5 юни 2016 г.

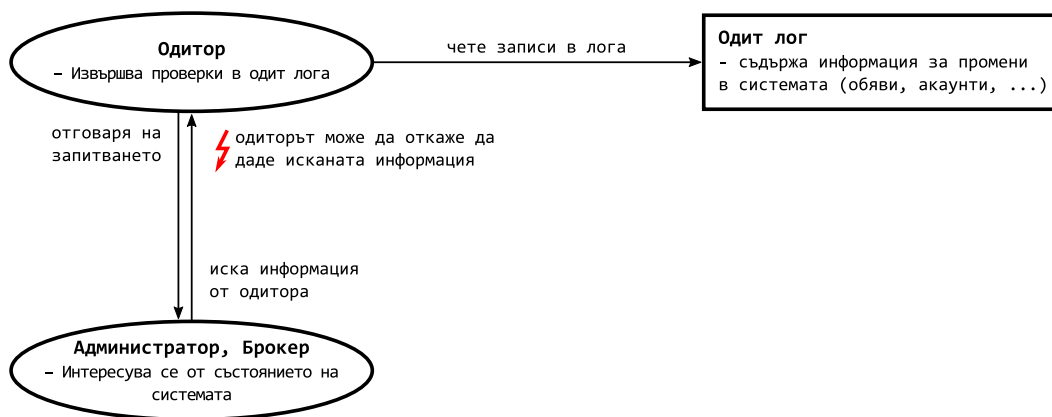
Интервюта

Участвах и в двете проведени две интервюта с Борислав Арnaudов (собственик на стартираща агенция за недвижими имоти) за изясняване изискванията към системата:

- на 2016-03-18 от 16:00 до 17:30 в зала 01 на ФМИ
- на 2016-04-01 от 09:00 до 10:30 в офиса на САП България

Contextual Design модел

От идентифицираните на базата на събраната информация роли, разработих модел на дейностите на Одитор.



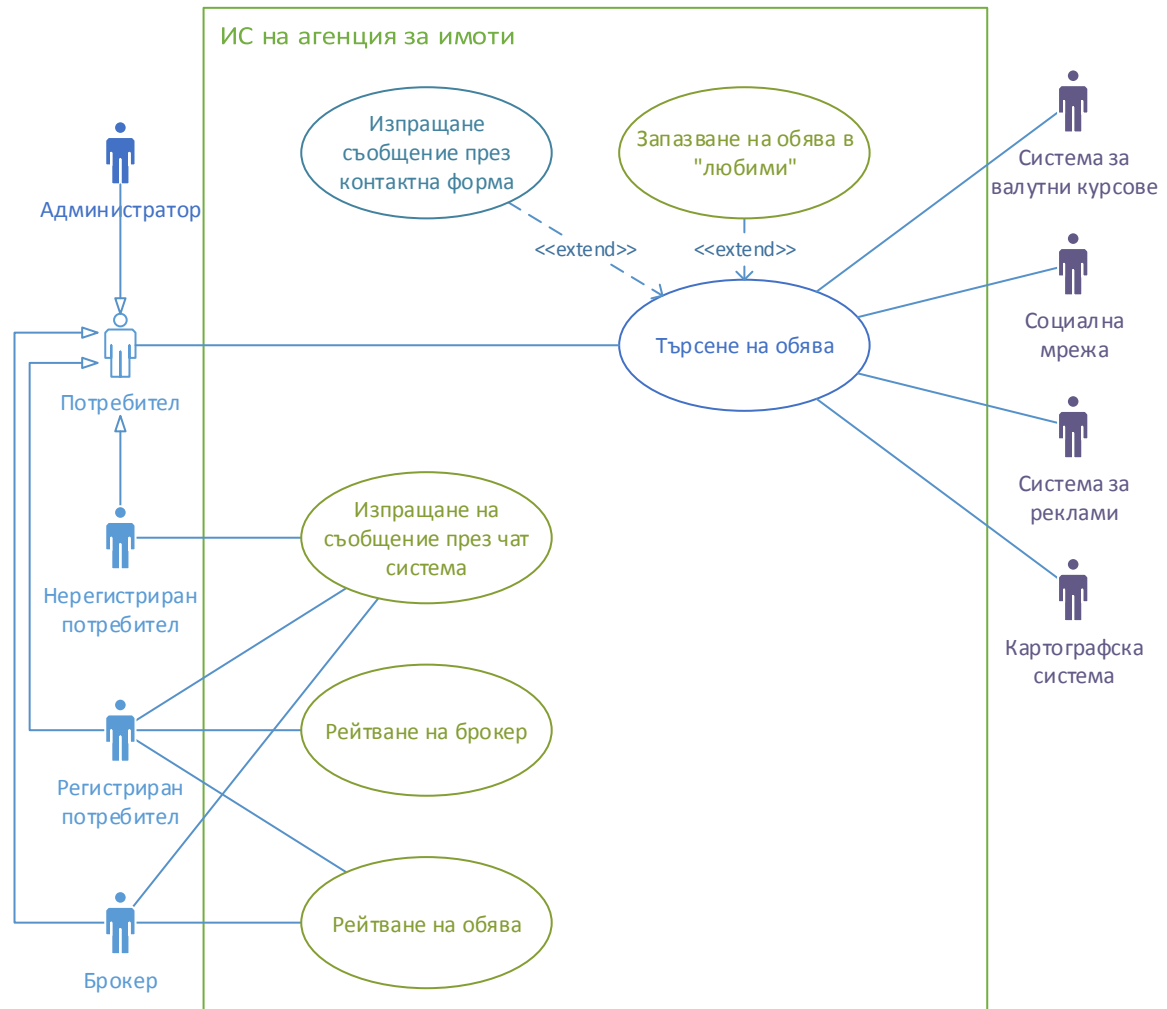
Фигура 1: Flow модел за Одитор

В системата съществува вграден акаунт за одитор, който има права само да чете одит лого. При нужда от информация за настъпила промяна в системата (за даден акаунт, обява и т.н.), брокерите, администраторът, собственикът на фирмата могат да поискат от одитора информация защо е настъпила промяната.

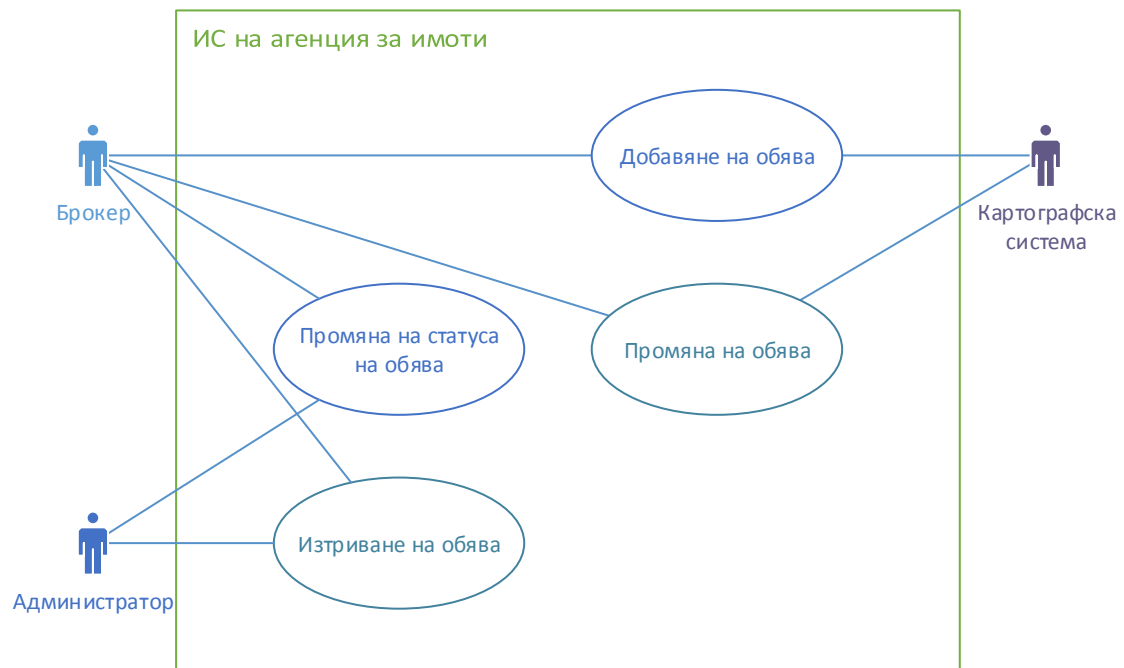
Заб. Одиторът може да откаже достъп до поисканата информация. Правилата за допуск са извън обхвата на системата.

Use case модел

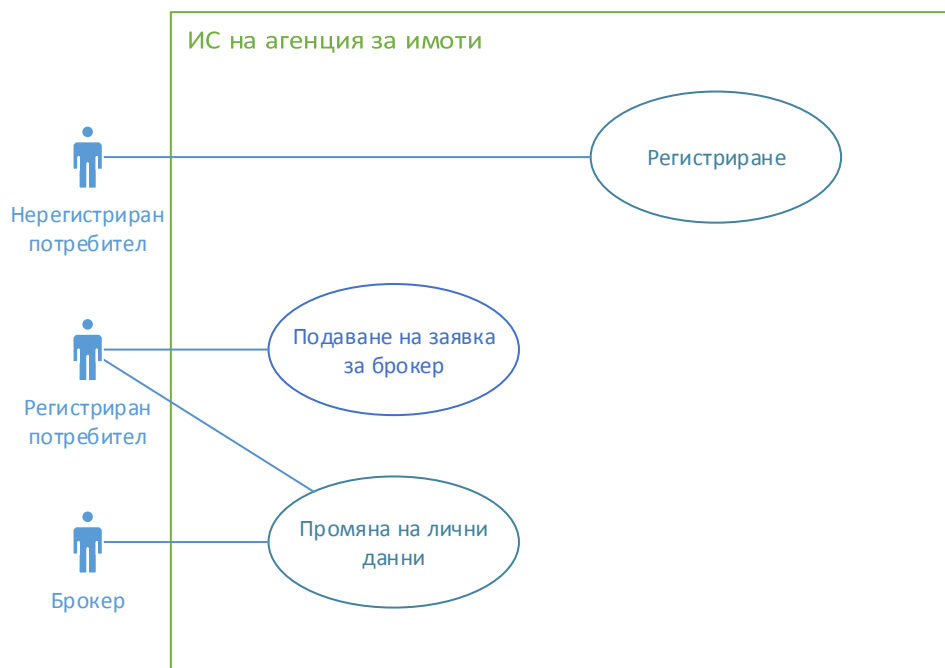
На базата на идентифицираните потребителски случаи, участвах в разработката на Модела на потребителските случаи:



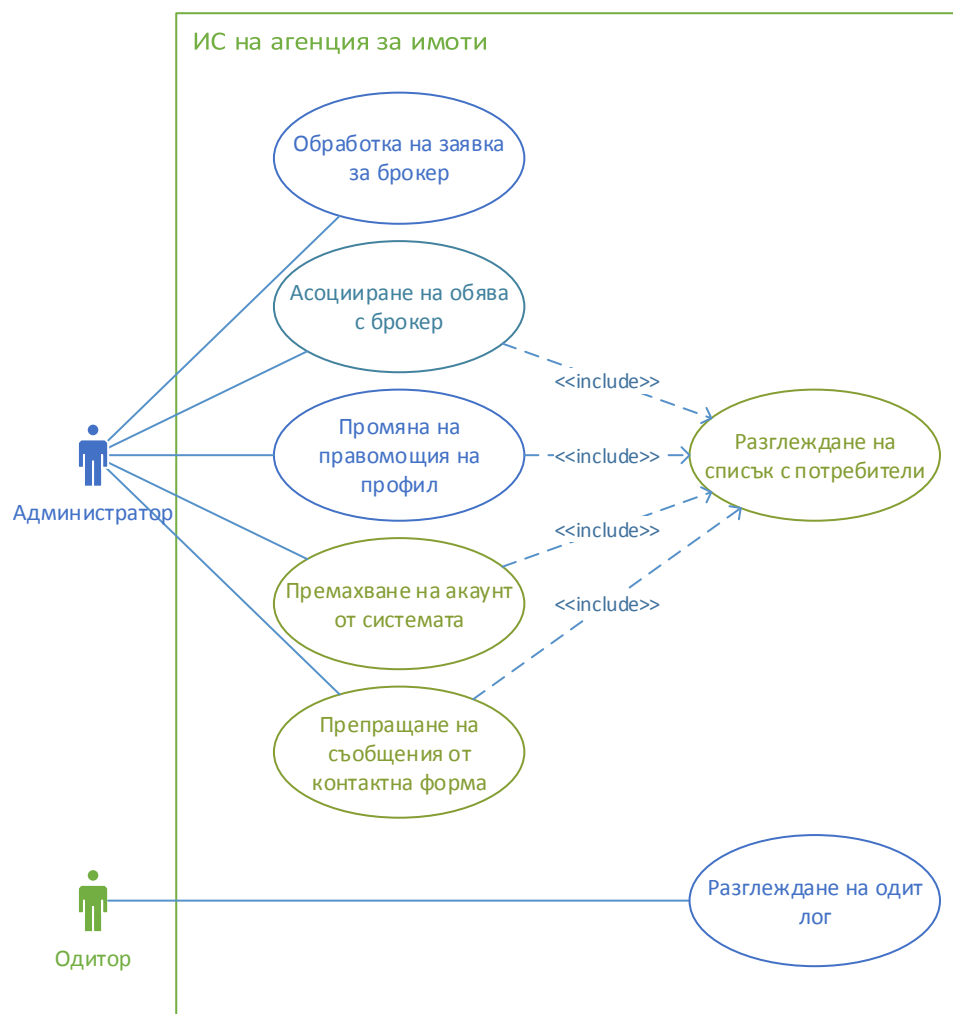
Фигура 2: Модел на потребителските случаи, свързани с основните функционалности на системата (от гледна точка на потребителите)



Фигура 3: Модел на потребителските случаи, свързани с обяви за имоти (от гледна точка на брокерите)



Фигура 4: Модел на потребителските случаи, свързани с акаунти на потребители



Фигура 5: Модел на потребителските случаи, свързани с административни дейности

Шаблон за пълно описание на потребителските случаи

Съгласно изискванията за част II на проекта, разработих шаблон за пълно описание на потребителските случаи.

Потребителски случай <номер>	<име на потребителския случай>
Ниво:	<user-goal или subfunction>
Основен актьор:	<основен актьор>
Заинтересовани лица и техните интереси:	<ul style="list-style-type: none"><заинтересовано лице и интереси>
Предусловия:	<ul style="list-style-type: none"><предусловие>
Следусловия:	<ul style="list-style-type: none"><следусловие>
Тригер:	<стартване>
Основен успешен сценарий:	<ol style="list-style-type: none"><стъпка><стъпка>
Алтернативни сценарии:	<ol style="list-style-type: none">2.a <Алтернативна стъпка на такава от основния сценарий><ol style="list-style-type: none"><стъпка от алтернативния сценарий><стъпка от алтернативния сценарий>5.a <Алтернативна стъпка на такава от основния сценарий><ol style="list-style-type: none"><стъпка от алтернативния сценарий><стъпка от алтернативния сценарий>
Специални изисквания:	<ul style="list-style-type: none"><Незадължително поле за специални изисквания>
Честота на настъпване:	<колко често се случва потребителския случай>
Коментари:	<Незадължително поле за коментари и въпроси>

Пълно описание на потребителските случаи

От идентифицираните потребителски случаи, описах следните два в пълен (fully dressed) формат:

Разглеждане на одит лог

Потребителски случай А-5	Разглеждане на одит лог
<i>Ниво:</i>	user-goal
<i>Основен актьор:</i>	Одитор
<i>Заинтересовани лица и техните интереси:</i>	<ul style="list-style-type: none">Администратор, Брокери, Бизнес: Интересуват се защо данните в системата са в определено състояниеОдитор: иска да види какви събития са отразените в одит лога събития, настъпили в системата, които биха могли да обяснят текущото ѝ състояние
<i>Предусловия:</i>	<ul style="list-style-type: none">Само брокерите, администраторът и собственикът на фирмата могат да поискат справка от одитораИскането към одитора трябва да отговаря на установените правила за допуск до информация. (Извън обхвата на системата.)Одиторът е логнат в системата
<i>Следусловия:</i>	Ако информация, отговаряща на критериите за търсене, дефинирани от одитора съществува, то системата трябва да я предостави на одитора.
<i>Тригер:</i>	Одиторът избира достъп до функционалност “Разглеждане на одит лог”
<i>Основен успешен сценарий:</i>	<ol style="list-style-type: none">Системата предоставя възможност за дефиниране на ограничения по времеви интервал, username, IP адрес, тип на действието, субект на действиетоОдиторът променя критериите за търсене в одит логаСистемата предоставя всички записи, отговарящи на текущите критерии за търсене в одит лога, чрез подмножества (страници)Одиторът разглежда предоставената информация
<i>Алтернативни сценарии:</i>	<p>4.а Одиторът променя критериите за търсене в одит лога</p> <ol style="list-style-type: none">Системата препраща потребителя към стъпка 3 от основния сценарий

<i>Специални изисквания:</i>	<ul style="list-style-type: none"> • Информацията в одит лога не може да се променя от никого • Системата премахва от одит лога информация, по-стара от 6 месеца.
------------------------------	---

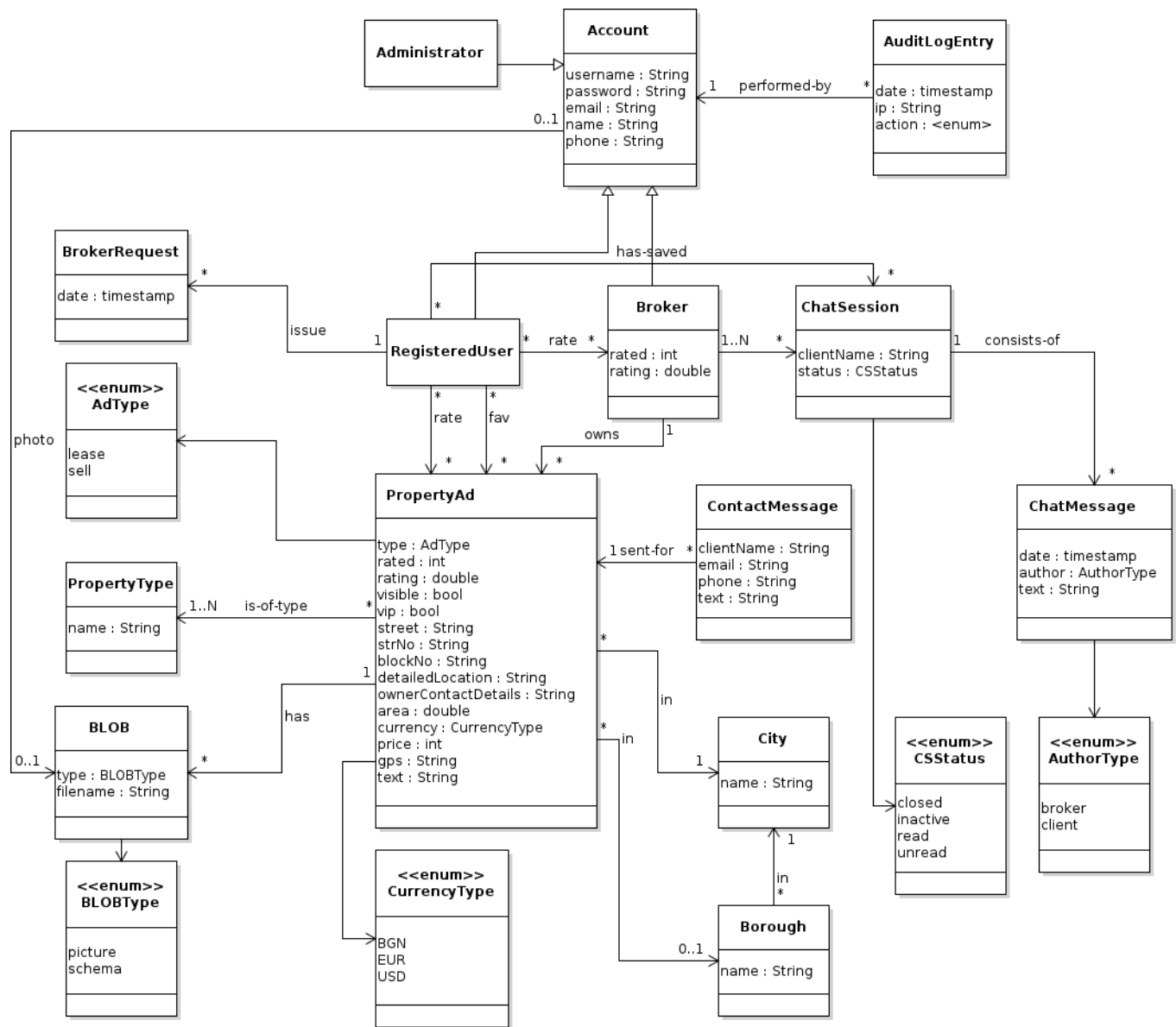
<i>Честота на настъпване:</i>	Рядко, не се очаква по-често от 10-тина пъти седмично.
-------------------------------	--

Разглеждане на списък с потребители

Потребителски случай С-7 Разглеждане на списък с потребители	
<i>Ниво:</i>	subfunction
<i>Основен актьор:</i>	Администратор
<i>Заинтересовани лица и техните интереси:</i>	<ul style="list-style-type: none"> • Администратор: Искане да избере конкретен потребител на системата
<i>Предусловия:</i>	<ul style="list-style-type: none"> • Администраторът да е логнат в системата • Да съществуват други акаунти в системата освен вградените Администратор и Одитор
<i>Следусловия:</i>	Системата “знае” дали и кой потребител е избрал Администратора
<i>Тригер:</i>	Администраторът избира достъп до функционалност за разглеждане на списък с потребители
<i>Основен успешен сценарий:</i> <ol style="list-style-type: none"> 1. Системата предоставя списък с всички регистрирани потребители, отговарящи на текущите критерии за търсене на потребители (при нужда - разделени на страници) 2. Администраторът избира потребител 	
<i>Алтернативни сценарии:</i> <ol style="list-style-type: none"> 2.a Администраторът променя критериите за търсене на потребители <ol style="list-style-type: none"> 1. Системата препраща Администратора към стъпка 1 от оригиналния сценарий 2.b Администраторът не избира потребител 	
<i>Специални изисквания:</i>	Системата позволява избиране само на акаунти тип Регистриран потребител и Брокер, без сервизните Администратор и Одитор
<i>Честота на настъпване:</i>	Очаквано – няколко пъти седмично.

Домейн модел

На базата на участието ми в определянето на домейн модела, създадох следната диаграма:



Класът Account съдържа атрибути за потребителско име, парола, email адрес, име и телефонен номер. Той се наследява от класовете Administrator, RegisteredUser и Broker. Всеки акаунт би могъл да е асоцииран с не повече от един обект от клас BLOB, като асоциацията представлява снимка на съответният потребител.

Класът BLOB представя съществуващи двоични файлове в системата. Той има атрибути за тип на файла (type - изброен тип, picture или schema) и име на файла (filename).

Когато регистриран потребител подаде заявка, че иска да стане брокер, се създава обект от клас BrokerRequest (асоцииран с RegisteredUser-a), който обект съществува до момента на одобряване или отхвърляне на заявката.

Класът Broker съдържа допълнителни атрибути:

- rating – натрупан до момента рейтинг на брокера
- rated – брой на гласовете за рейтинг

При започване на чат сесия, в системата се създава обект от клас `ChatSession` (асоцииран с участващите в сесията `Broker`-и), който има атрибути:

- `clientName` – име, с което потребителят участва в чат сесията
- `status` – избран тип с възможните състояния на сесията (`closed`, `inactive`, `read`, `unread`)

Сесията е асоциирана с обекти `ChatMessage`, които имат атрибути `date`, `author` (избран тип – `broker` или `client`), и `text` на самото съобщение.

Обектите от клас `City` (с атрибут `name`) дефинират на населените места, които поддържа системата. Подобен е и класът `Borough` (с атрибут `name`), който дефинира квартали. Всеки обект от клас `Borough` е асоцииран с град, в който се намира.

Класът `ContactMessage` описва дадено съобщение, изпратено през формата за контакт. Съдържа атрибути за име клиента (`clientName`), `email` адрес на клиента (`email`), телефон за обратна връзка (`phone`) и самото текстово съобщение (`text`).

Основан е класът `PropertyAd`, който съдържа информация за обява в системата. Основни негови атрибути са:

- `type` – тип на обявата, избран тип (`lease/sell`)
- `area` – застроена площ на имота
- `price` – цена на имота
- `currency` – валута на цената, избран тип (`BGN/EUR/USD`)
- `rating` – натрупан до момента рейтинг на обявата
- `rated` – брой на гласовете за рейтинг
- `visible` – `bool`, който дефинира дали обявата е публично видима
- `vip` – `bool`, който дефинира дали обявата е нормална или `vip`

Класът участва в следните асоциации:

- с `BLOB` – дефинира снимки и скици на имота
- с `City` – дефинира в кой населен пункт се намира имота
- с `Borough` – дефинира в кой квартал се намира имота
- с `ContactMessage` – всяко изпратено през контакт формата съобщение се отнася за конкретна обява в системата

Действията, променящи системата създават обекти от клас `AuditLogEntry`, които са асоциирани с акаунта, извършил действието и съдържат атрибути за дата (`date`), IP адрес (`ip`) и извършено действие (`action`).

Разпределение на времето

В таблицата е описано времето (в минути) за работата по различните части и задачи на проекта.

	71512
Интервю 1	90
Интервю 2	90
Транскрибиране 1	90
Транскрибиране 2	120
Записки 1	0
Записки 2	0
Резюме 1	360
Резюме 2	120
Уточняване на модели	60
Работа по CD модели	120
Подготовка на Доклад I	540
Коригиране на I-ва част	0
Визия	15
Use Case модел	90
Списък Актьори	15
Списък UC (brief)	30
Шаблон за UC	30
Fully dressed UC	30
Нефункционални изисквания	10
Речник	20
Review	180
Подготовка на Доклад II	450
Коригиране на II-ра част	90
Use Case модел	180
Пълен списък UC	0
Fully dressed UC	120
Домейн модел	450
UML диаграми	0
Примерен план на проекта	60
Подготовка на Доклад III	390
Общо	3750