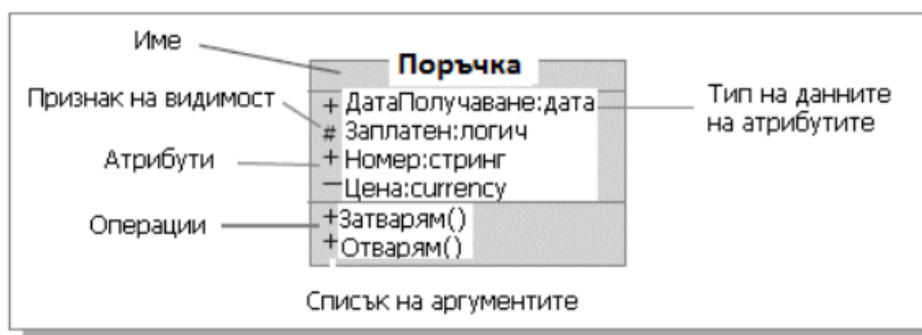


Упражнение 4: Детайлно проектиране на системата. Използване на UML нотация.

Диаграма на класовете на дизайна

Клас диаграмите се явяват основни при анализ на изискванията и проектиране на ОО системи, тъй като позволяват нагледно да се изобрази структурата на класовете в приложенията и статичните връзки между тях. Това са структурни UML диаграми, чрез които се описва статичната структура на системата или на части от нея.



Между класовете са възможни различни отношения, представени на клас диаграмата по-долу:

- зависимости (dependency relationship), които описват съществуващо между класовете отношение на използване;
- обобщения (generalization relationship), свързващи обобщените класове със специализираните;
- асоциации (association relationship), отразяващи структурни отношения между обектите на класовете.

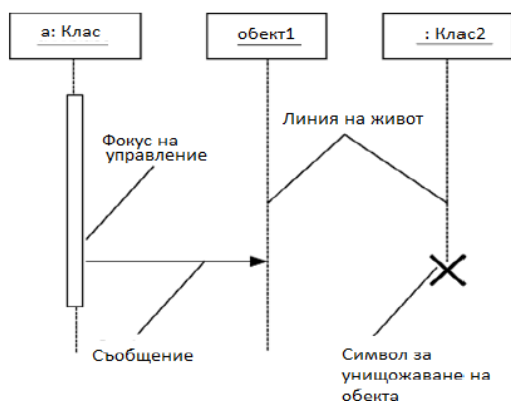
Диаграма на последователностите

Обикновено най-ляво е обекта, инициатор на моделирания процес на взаимодействие (обект „а“ на фигура 1). Следва – обекта, който непосредствено взаимодейства с инициатора, тоест подредбата на обектите е също важна, от съображение за удобство при визуализацията на взаимодействията между обектите.

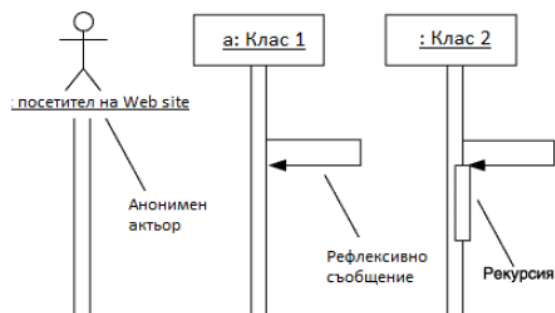
На началния момент време съответства най-горната част на диаграмата. Процесът на взаимодействие се реализира посредством съобщения, които се изпращат от едни обекти към други. Съобщенията се изобразяват във вид на хоризонтални стрелки, като тези, които са по-горе в диаграмата се предават по-рано от тези, които са по-ниско.

Линия на живот на обекта (object lifeline): това е вертикалната пунктирна линия на диаграмата на последователността, която представя съществуването на обекта в течение на определен период от време (обект 1 и обект а тип „Клас“ на фигура 1 съществуват през цялото време). За обозначаване на момента на

унищожаване на обект се използва символ "X". На фигура 1, този символ се използва за унищожаване на анонимния обект, от тип Клас 2.



Фигура 1. Графични елементи на диаграмите на последователностите



Фигура 2. Актьор, рефлексивно съобщение и рекурсия в диаграмата на последователностите

Правоъгълникът върху линията на живот на обекта обозначава „активно” състояние на обекта. Този правоъгълник се нарича фокус на управление (focus of control) – указва периода време, в течение на който обектът изпълнява някакво действие, тоест периода време, в течение на който обектът се намира в активно състояние. В отделни случаи, инициатор на взаимодействието може да е актьор или външен потребител, тогава той се изобразява като първи обект в диаграмата, със свой фокус на управление (фигура 2). В отделни случаи, обектът може да изпраща съобщения на себе си (рефлексивни съобщения) – вижте съобщение на обект а:Клас1 във фигура 2.

Пример - Система за управление на наем на недвижим имот

Диаграма на класовете на дизайна

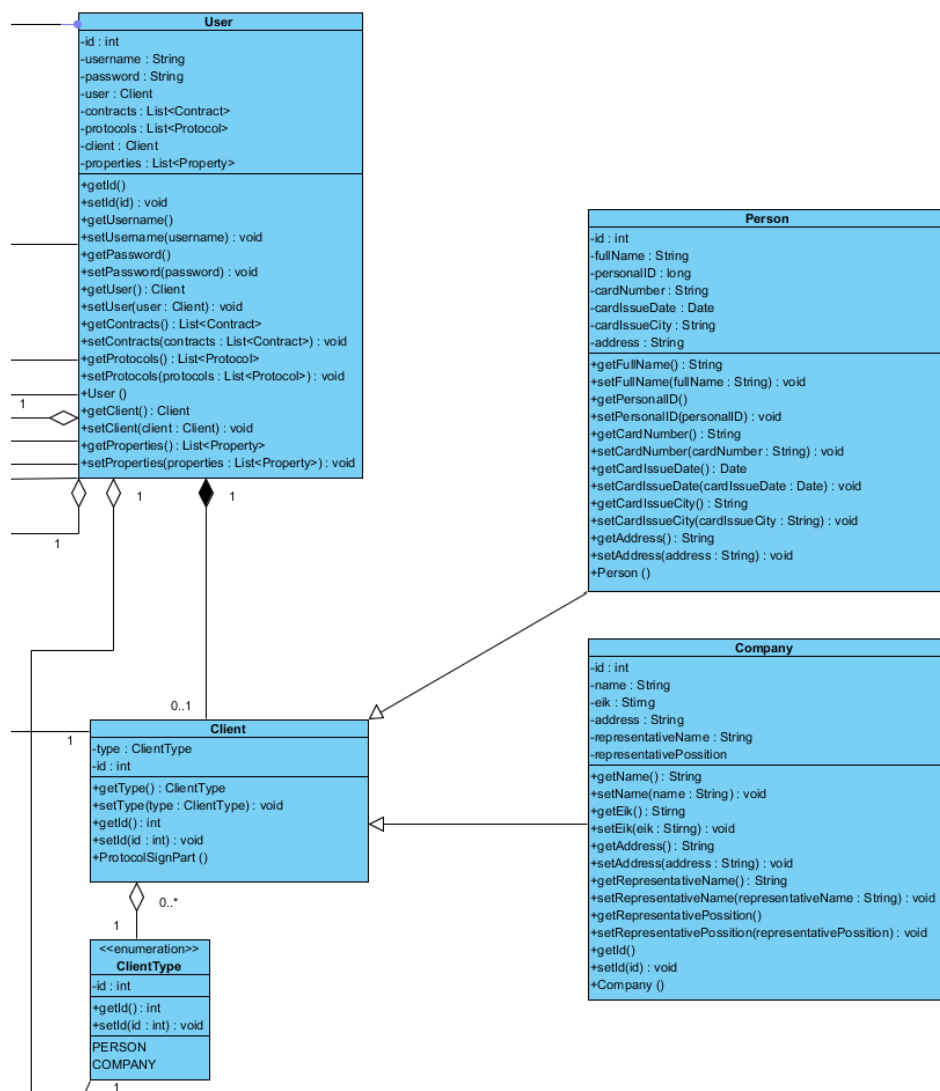
Диаграмата на класовете на дизайна описва всички класове, които ще бъдат създадени по време на имплементацията на системата. За всеки клас е предоставена информация за неговото име, състояние, поведение и взаимодействието му с останалите класове.

User	
Description	Съхранява информацията за регистриран потребител
Relationships	Композиция: <ul style="list-style-type: none"> • съдържа 1 обект от тип Client Агрегация <ul style="list-style-type: none"> • съдържа 0..* обекта от тип Property • съдържа 0..* обекта от тип Contract • съдържа 0..* обекта от тип Protocol

Person	
Description	Съхранява информация за лице, което е физическо лице
Relationships	Специализация на типа Client

Company	
Description	Съхранява информация за лице, който е юридическо лице
Relationships	Специализация на типа Client

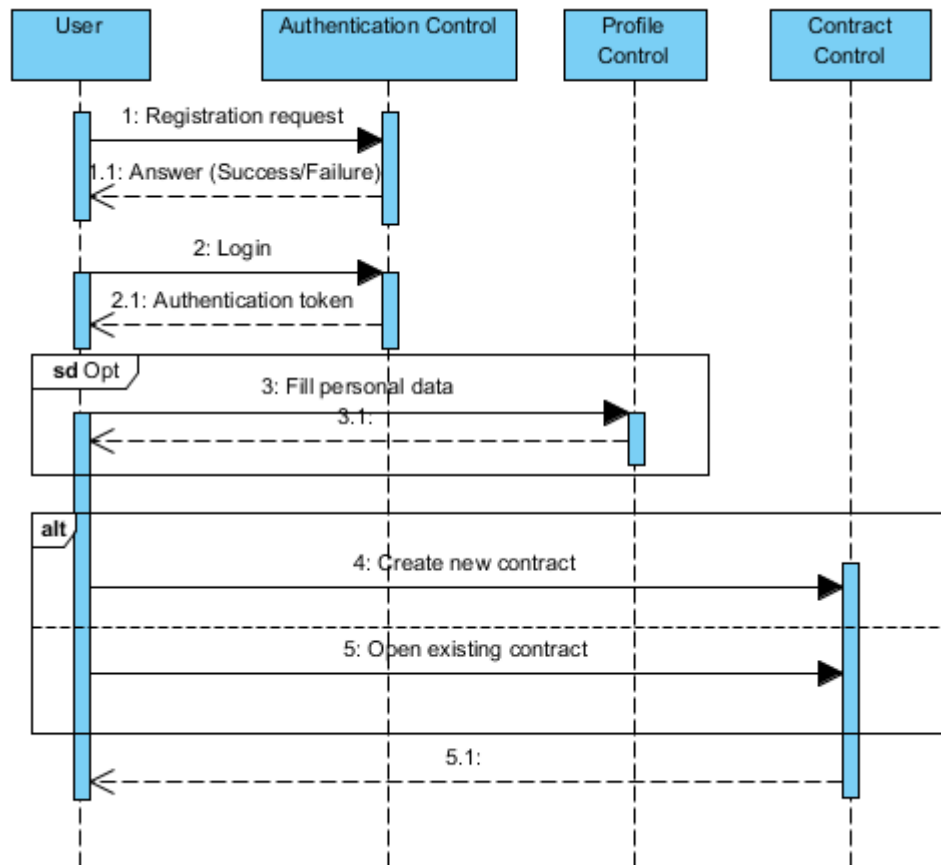
Client	
Description	Съхранява информация за клиент
Relationships	<p>Генерализация на типовете Person и Company</p> <p>Агрегация:</p> <ul style="list-style-type: none"> сдържа 1 обект от тип ClientType



Диаграма на взаимодействието на обектите

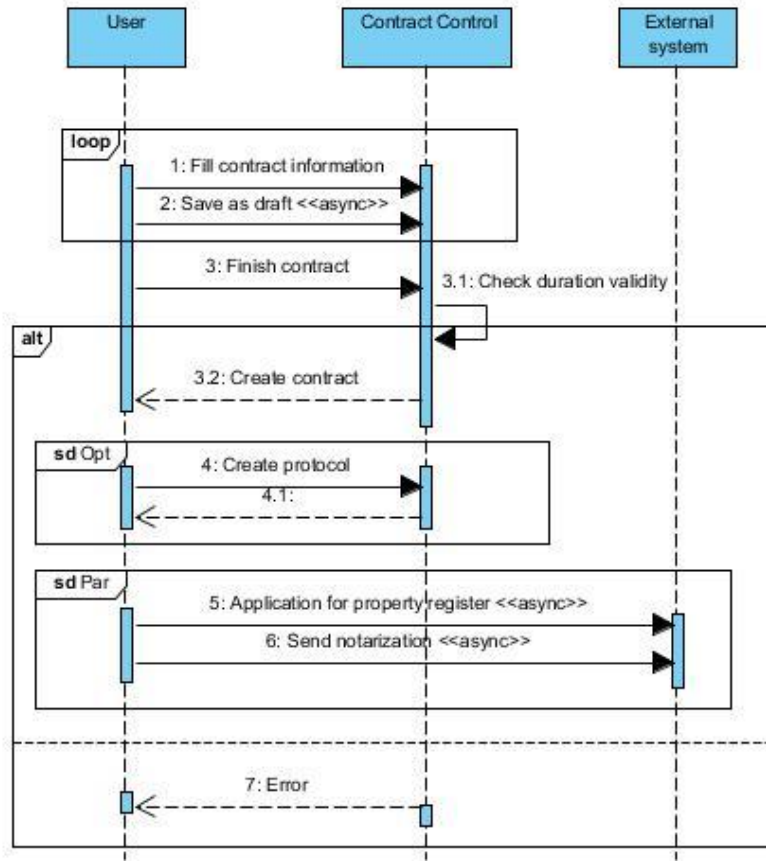
Диаграмата илюстрира процеса на отваряне на договор за редакция след регистрация:

1. Потребителят се регистрира в системата -> системата връща отговор за успешна/неуспешна регистрация.
2. Потребителят влиза чрез направената регистрация -> системата връща токен за аутентикация.
3. Потребителят попълва лични данни (Това действие не е задължително!).
4. Потребителят създава нов договор ИЛИ Потребителят отваря вече създаден договор -> системата предоставя екран за редакция на договор



Следващата диаграма илюстрира процеса на редакция и приключване на договор. Началната точка е отворен договор за редакция:

1. Потребителят попълва информация в договора
2. Потребителят запазва въведените данни като чернова (Асинхронно съобщение!) * Забележка: Действия 1 и 2 могат да се повтарят повече от веднъж
3. Потребителят приключва договор -> системата проверява за валидност на данните. Ако те не са валидни, връща грешка. В противен случай -> към точка 4
4. Изготвя се приемо-предавателен протокол. (Това действие не е задължително!)
5. Изпраща се заявка за вписване в имотния регистър. (Асинхронно съобщение!)



Задача:

1. Определяне на класовете на отделните модули. Обектно-Ориентиран Дизайн методология - Class Responsibility Collaboration cards:
 - определяне на участниците – обектите;
 - определяне на взаимодействията и отговорностите.
2. Изграждане на диаграма на класовете (напр. с <https://www.draw.io/>).
3. Проектиране на работния процес на потребителски случай на употреба (user story)- от действието на крайния клиент, през бизнес логиката, до системата за съхранение на данни и обратно
Резултат: диаграма на последователностите (напр. с <https://www.draw.io/>, <https://www.websequencediagrams.com/>) описваща конкретен бизнес случай на употреба.