# Глава 2. Архитектура на облачна система за управление на поръчките от клиенти

## 2.1. Същност, цел и обхват на софтуерната архитектура

Думата „**архитектура**“ често се използва в контекста на нещо от високо ниво, което е отделено от детайлите на по-ниско ниво (Martin et al., 2017). Софтуерният продукт, разглеждан в настоящия труд, ще се състои от 2 клиентски приложения, които ще се свързват към клъстер от услуги.

### 2.1.1. **Ключови бизнес процеси и дейности при системата за** управление на поръчките

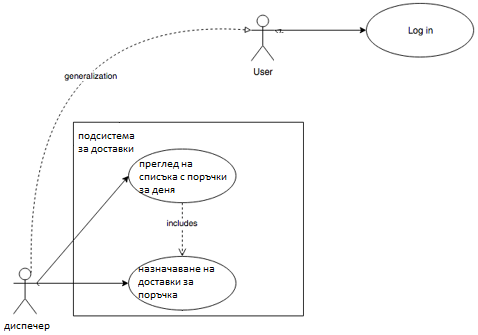
Тази подточка ще представи важни случаи на употреба, които са критични за бизнеса, част от основния домейн. Ще бъдат използвани диаграми на бизнес сценариите. Те ще идентифицират действия, които очакваме потребителите да направят.

Най-подходящ за взаимодействие с крайните потребители са мобилните приложения (). Също така поддържат функции като местоположение и камера и работят с уеб API. Обхватът на мобилното приложение включва екран за вход както за клиент, така и за шофьор, интерфейс за текущите поръчки и доставки към тях. Също така панел за създаване или промяна на поръчка. Фигура 2.1 представя процесите под формата на диаграма:



Фиг. 2.1. Диаграма на главен бизнес сценарий (разработка на автора)

- Обхват на уеб портала



Фиг. 2.2. Диаграма на главен бизнес сценарий (разработка на автора)

## 2.2. Концептуален модел на системата

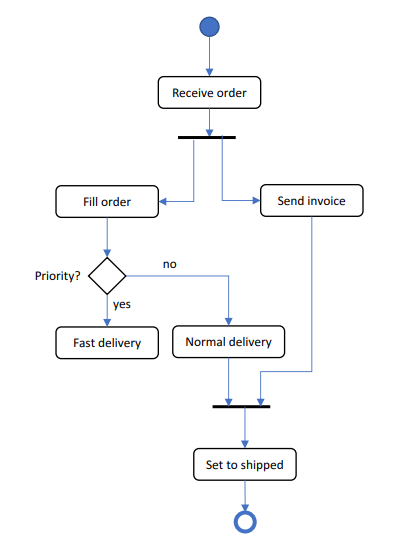
Концептуалните модели са абстрактни представяния за това как трябва да протича изпълнението на задачите. Представят визуално концепция или операция.

### 2.2.1. Поведенчески диаграми

Поведенческите диаграми идентифицират как различните елементи взаимодействат помежду си, за разлика от структурните диаграми, които описват блоковете, които изграждат самата системата.

#### 2.2.1.1 Диаграми на дейностите UML

Диаграмите на дейността изглеждат много подобни на блок-схемите. Наличието на тези прилики улеснява комуникацията между технически и не-технически лица. Следната диаграма представя работни потоци и общи операции в мобилното приложение. ТОДО+



Фиг. 2.6. Диаграма на дейностите за мобилно приложение. (разработка на автора)

#### 2.2.1.2. Диаграма на последователностите UML

Диаграмите на последователностите също са често използвани поведенчески диаграми в UML. Те идентифицират как обектите в система взаимодействат помежду си, за да реализират определена функционалност, като визуализират времевата линия и редът, в който се извършват операциите.

Пример може да разгледаме в контекста на подсистемата за поръчки:



Фиг. 2.7. Диаграма на последователностите. (разработка на автора)

https://blog.openreplay.com/jwt-authentication-best-practices

### 2.2.2. Структурни диаграми

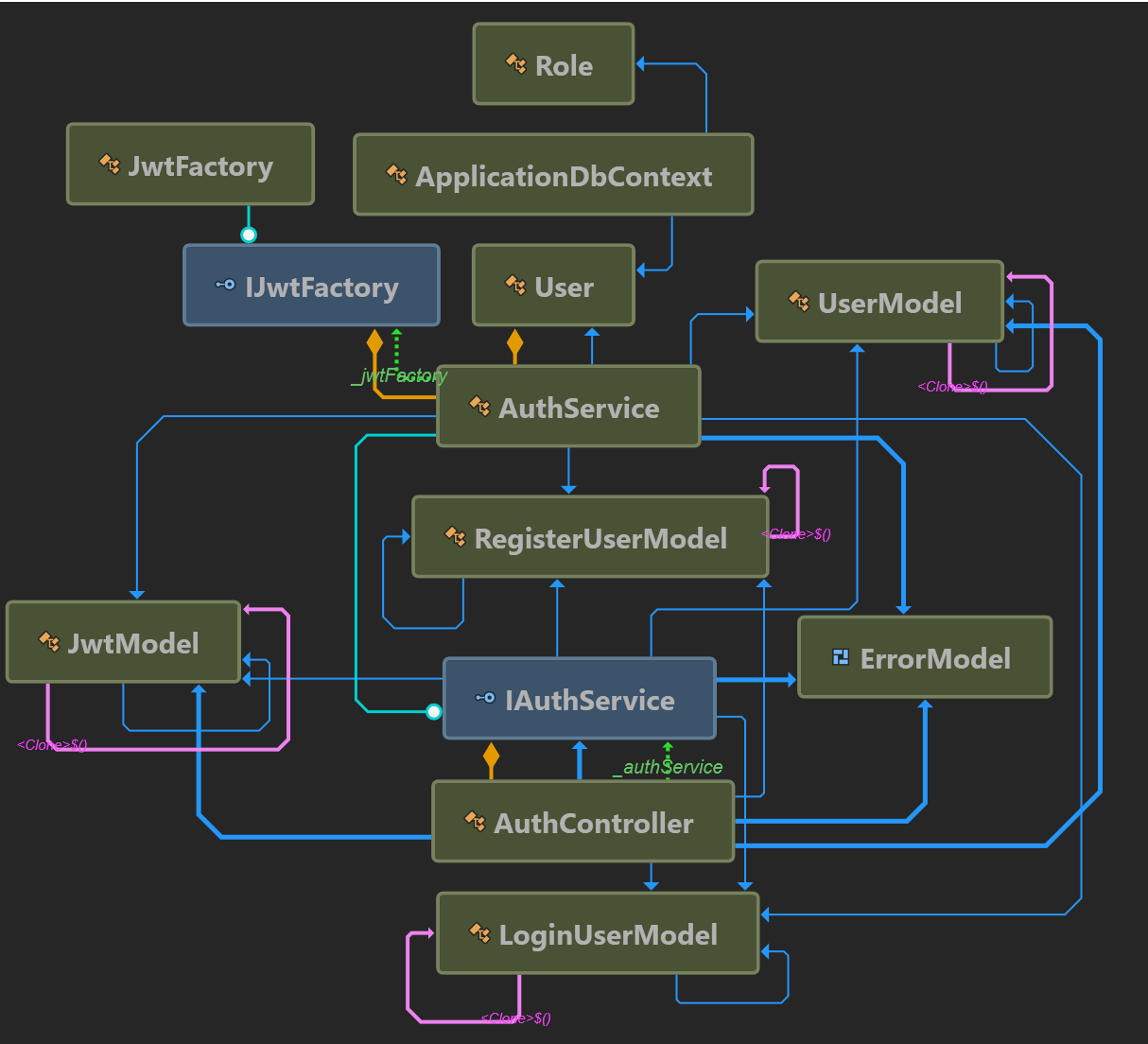
Структурните диаграми помагат за дефиниране цялостната структура системата, подобно на плана, който определя как изглежда една къща. Структурните диаграми моделират как ще изглежда системата в архитектурно отношение. Те ни помагат да дефинираме „речника“ на системата, гарантират съгласуваност от заинтересовани страни в проекта. Идентифицират различни връзки между различните частти.

Структурните UML диаграми изобразяват елементите на система, които са независими от времето и които предават концепциите и как те се свързват помежду си. Елементите в тези диаграми приличат на съществителните в естествения език.

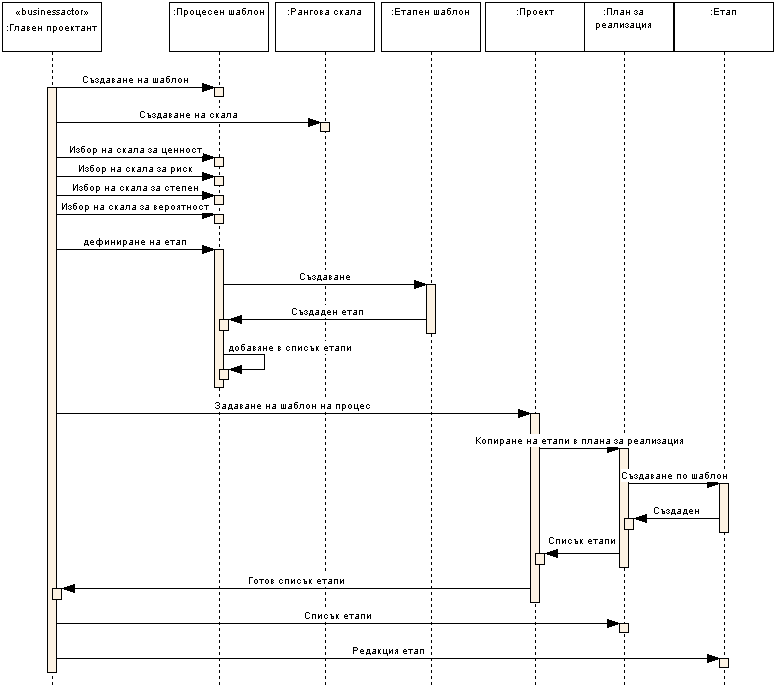
#### 2.2.2.1. Диаграма на класовете UML

Диаграмите на класове са едни от най-често срещаните, когато става на въпрос за разработката на софтуер. Едно от основните неща, които тези диаграми правят е да идентифицира речника на системата. Например, те определят връзките между обектите, които съответстват на основните съществителни.

Следващата част представя диаграма на класовете, свързани с удостоверяване. Това е процесът на определяне кой има достъп до системата. Елементите от приложението и зависимости, които ще обслужват тази част са визуализирани на фиг. 2.2. **DbContext** и **ApplicationUser** представляват комбинация от класове, които оперират с базата от данни.  **AccountController** използва тези свойствa чрез **UsersService**, който капсулира логиката по безопасен за използване начин.



Фиг. 2.8. Интерактивна диаграма за БСВС „Разработка на продукт / Планиране на проект”. (разработка на автора)



Фиг. 2.9. Диаграма за реализация на шаблони за процес. (разработка на автора)

На тази база може да се определят операциите на отделните класове, което спомага както да се завърши дефиницията за интерфейса на бизнес логиката, така и функционалността на системата като необходим интерфейс. Допълването с диаграми на състоянията за статуса на класовете, които бяха представени в информационния модел на концепцията (фиг. 2.4), също е от съществено значение. Те посочват жизнения цикъл на обектите и необходимата логика при управлението им съгласно техния статус и събития.

# Използвана литература

1. Галоуей, С. (2021) *Посткорона. От криза към възможност*. Изток-Запад, София.
2. Ескенази, А., & Манева, Н. (2006). *Софтуерни технологии.* София: КЛМН.
3. Закон за защита на конкуренцията. Държавен вестник, София, бр.17 от 26.02.2021 г.
4. Закон за корпоративното подоходно облагане. Държавен вестник, София, бр. 95 от 8.12.2015 г.
5. Закон за малки и средни предприятия. Държавен вестник, София, бр. 17 от 1.03.2016 г.
6. Закон за насърчаване на инвестициите. Държавен вестник, София, бр. 61 от 11.08.2015 г.
7. Закон за счетоводството. Държавен вестник, София, бр.19 от 05.03.2021 г.
8. Илиев, П., Сълов, В., & Петров, П. (2010). *Виртуални системи.* Монографична библиотека „Цани Калянджиев”, Варна: Наука и икономика.
9. Сълов, В. (2014). *Производителност и ефективност на компютърните системи*. Варна : Унив. изд. Наука и икономика.

**Интернет източници**

1. Даскал, Л. (2018). 7 стратегически умения за по-ефективно лидерство. Мениджър. <https://manager.bg/liderstvo/trayna-7-nachina-da-bdete-po-efektivni-lideri> [23.08.2022]
2. Икономи, П. (2018). Тайните на ефективното лидерство. Мениджър. <https://manager.bg/liderstvo/trayna-taynite-na-efektivnoto-liderstvo> [23.08.2022]