

Список выполненных проектов:

1. Распределенная система контроля incoming/outgoing транзакций к БД была разработана в процессе летней производственной практики. Приложение было разделено между тремя удаленными серверами Amazon AWS EC2 instances, один выступал в качестве MySQL-сервера, второй содержал backend-часть приложения, третий – frontend. Бэкэнд был написан на Spring Boot. Обращение к веб-части (REST endpoints) – Spring Web. Обращение к БД – Spring Data JPA. Была простая конфигурация ролей/доступов на Spring Security. Некоторые эндпоинты были скрыты и предназначались для использования самим приложением, другие – были частью публичного веб-интерфейса. Фронтенд был написан на Angular, представлял из себя веб станицу с возможностью отправления http-запросов транзакций бэкэнду о начислении/списании средств той или иной фирме. Суть приложения – в синхронизации поступающих транзакций. Для этого проверки данных были написаны на всех уровнях приложения, а уровень бекэнда генерировал синхронизированные транзакции к БД в ответ на асинхронные запросы фронтенда. Несколько примеров исходного кода:

```
@Transactional(propagation = Propagation.REQUIRES_NEW, isolation = Isolation.SERIALIZABLE)
@PostMapping(path = "/incoming", consumes = "application/json", produces = "application/json")
@ResponseBody
public TransactionBean incomingTransaction(@RequestBody TransactionBean transaction) {
    return transactionsRepository.save(transaction);
}

@Transactional(propagation = Propagation.REQUIRES_NEW, isolation = Isolation.SERIALIZABLE)
@PostMapping(path = "/outgoing", consumes = "application/json", produces = "application/json")
@ResponseBody
public TransactionBean outgoingTransaction(@RequestBody TransactionBean transaction) {
    CompanyBean company = companiesRepository.findById(transaction.getCompany().getId()).get();
    Double difference = company.getBalance() - transaction.getTransactionValue();
    if (difference < 0) {
        throw new TransactionValueBiggerThanCompanyBalanceException(company.getBalance(),
            transaction.getTransactionValue());
    }
    transaction.setTransactionValue(transaction.getTransactionValue() * (-1));
    TransactionBean result = transactionsRepository.save(transaction);
    return result;
}
```

1 Эндпоинты приема транзакций

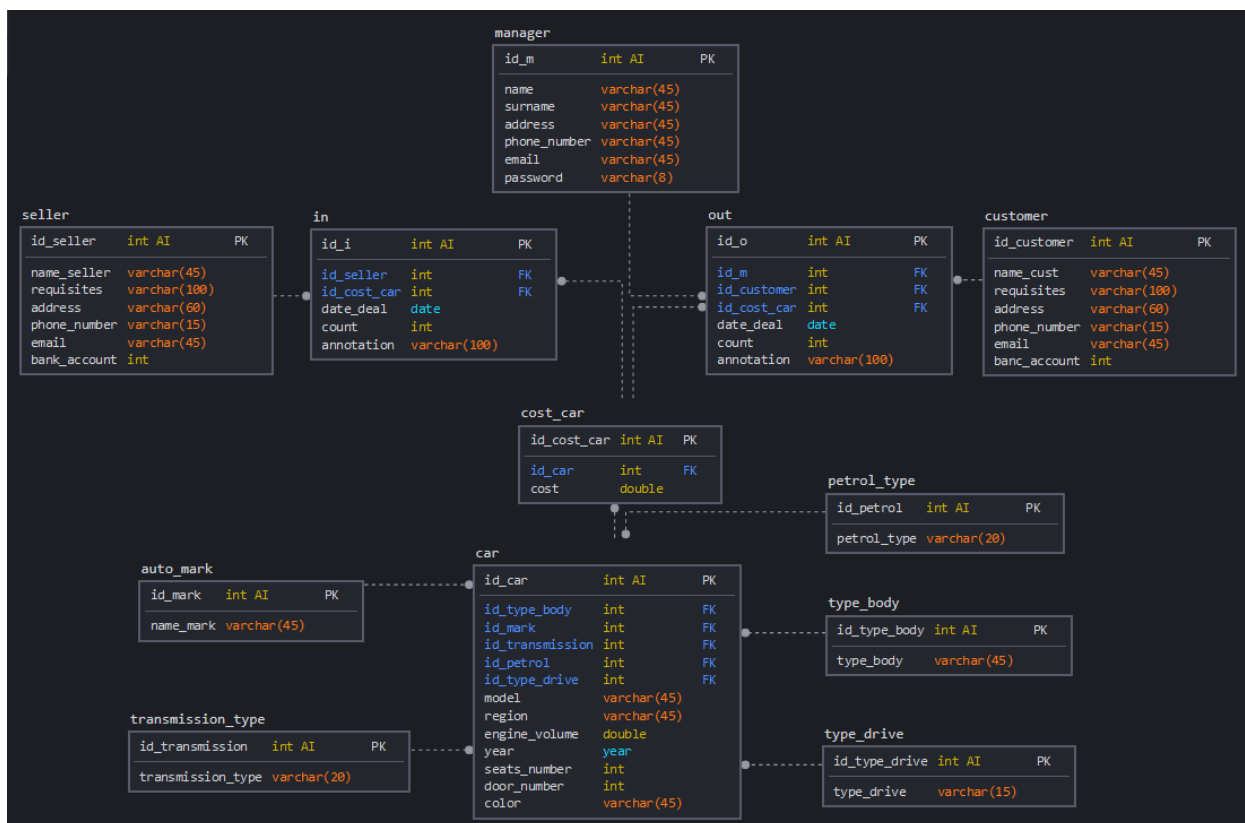
```

private post(index: number, endpoint: string, attemptCount: number = 0) {
  this.setErrorVisible(false);
  let company = this.companies[index];
  let transactionValue: number = Number(this.transactionValues[index]);
  if (isNaN(transactionValue)) {
    this.errorText = Error.PARSE_ERROR;
    this.setErrorVisible(true);
    return;
  }
  this.transactionsService.post(endpoint, transaction: {company, transactionValue})
    .subscribe(
      next: (data: any) => {
        this.companies[index].balance += data.transactionValue;
        console.log(data);
      },
      error: (error: HttpErrorResponse) => {
        console.log(error);
        //422 == UnprocessableEntity, throws when transaction value > company's balance
        if (error.status == 422) {
          this.errorText = Error.VALUE_ERROR;
          this.setErrorVisible(true);
        }
        //500 == InternalServerError, throws when another transactions blocks or lost con
        if (error.status == 500) {
          console.log('attempt: '+attemptCount);
          if(attemptCount >= 50){
            this.errorText = Error.SERVER_ERROR;
            this.setErrorVisible(true);
            return;
          }
          this.post(index, endpoint, attemptCount: attemptCount + 1);
        }
      }
    )
}
}

```

2 Посылка запросов к бекэнду

2. Десктоп-приложение с подключением к БД «система регистрации транспортных средств автосалона». Разрабатывалось в качестве курсовой работы к курсу Организации Баз Данных, для показа практических знаний в области проектирования БД и написания различных запросов. GUI был написан на Swing, подключение к БД – JDBC. Приложение отличалось большим количеством разнообразных корректировок вводимых/получаемых данных, которые регулировали выходные запросы к БД. Схема БД и несколько скриншотов программы:



AutoSalon

AUTO SALON

CATALOG DELIVERIES SALES SELLERS CUSTOMERS

MARK: --Any--

MANUFACTURE REGION: --Any--

COLOR: --Any--

BODYTYPE: --Any--

PETROL TYPE: --Any--

TRANSMISSION: --Any--

TYPE OF DRIVE: --Any--

SEATS COUNT: --Any--

YEAR FROM: 2000

YEAR TO: 2021

SEATS COUNT: --Any--

DOORS COUNT: --Any--

COST FROM: 50 000

COST TO: 1000 000

TOYOTA ZXC

Year: 2020

Color: Red

Body type: Sedan

Petrol type: Diesel

Transmission: Automatic

Cost: 50000.0

MORE >

KIA RIO

Year: 2021

Color: Black

Body type: Sedan

Petrol type: Benzine

Transmission: Semiautomatic

Cost: 80000.0

MORE >

TOYOTA ZXC

Year: 2020

Color: Red

Body type: Sedan

Petrol type: Diesel

Transmission: Automatic

Cost: 60000.0

MORE >

BMW MS

Year: 2021

Color: Black

Body type: Crossover

Petrol type: Gas

Transmission: Automatic

Cost: 50000.0

MORE >

TOYOTA COROLA

Year: 2015

Color: Purple

Body type: Sedan

Petrol type: Diesel

Transmission: Semiautomatic

Cost: 120000.0

MORE >

MERCEDES-BENZ C200

Year: 2016

Color: Blue

Body type: Hatchback

Petrol type: Benzine

Transmission: Sequential

Cost: 70000.0

MORE >

BEZRUK YURIJ

LOG OUT

AutoSalon

AUTO SALON

CATALOG DELIVERIES SALES SELLERS CUSTOMERS

CAR:

TOYOTA ZXC

Year: 2020

Color: Red

Bodytype: Sedan

Petroltype: Diesel

Transmission: Automatic

Cost: 50000.0

MORE >

< BACK

CLEAR

SELL

COUNT:

1

CUSTOMER DATA:

NAME:

ADDRESS:

PHONE:

EMAIL:

BANK ACCOUNT:

ANNOTATION:

PROPOSED CUSTOMERS:

NAME	ADDRESS	PHONE	EMAIL	BANK ACCOUNT	REQUISITES	FAST CHOOSING
KrepaturaA.A.	st. Hurts, 20	88005553555	krep@pain.com	GB98MDLD 700931234 678	No credit history, have high-paying job and home high-paying job and home	CHOOSE *
Sobolev M.I.	Russia, Saint Petersburg	+79307066318	sobolevwork@gmail.com	RU8937040 044063301	Blager, work with govern	CHOOSE *

BEZRUK YURIJ

LOG OUT

AutoSalon

AUTO SALON

CATALOG DELIVERIES SALES SELLERS CUSTOMERS

BACK

CLEAR

SELL

COUNT: 1

CUSTOMER DATA:

NAME:

ADDRESS:

PHONE:

EMAIL:

BANK ACCOUNT:

ANNOTATION:

PROPOSED CUSTOMERS:

NAME	ADDRESS	PHONE	EMAIL	BANK ACCOUNT	REQUISITES	FAST CHOOSING
Krepatura A.A.	st. Hurts, 20	88005553555	krep@pain.com	GB98MDLO 700931234 678	No credit history, have high-paying job and home high-paying job and home	CHOOSE *
Sobolev N.I.	Russia, Saint Petersburg , Nekrasovst.	+8307955318	sobolevwork@gmail.com	RUR8937040 044053201 000	Bluger, work with govern ment	CHOOSE *
Сатори С.В.	Russua, Moscow, avenue L enina	+88016677564	lkakprosto.ru@gmail.com	RU4550000 000058398 57466	Smooth gay, respect him	CHOOSE *
Dibtseva A.N.	Kharkovskikh Divizii, 18	45564564564564	little.lemur.dl@gmail.com	545645645 34135465		CHOOSE *
Alina	5555, Russia	+9885221	stalin.ua@gmail.com	RU951651		CHOOSE *

BEZRUK YURIJ

LOG OUT

AutoSalon

AUTO SALON

CATALOG DELIVERIES SALES SELLERS CUSTOMERS

DATE FROM: 01.01.70 0:00 DATE TO: 18.05.210:00 SELLER: --Any-- [ADD NEW DELIVERY >](#)

SELLER	CAR	COST	COUNT	DATE	ANNOTATION
000 Kaloshi MORE >	KIA Rio MORE >	80000.0	5	2021-05-10	Тестовая покупка
000 Kaloshi MORE >	Toyota ZXC MORE >	60000.0	4	2021-05-16	шiff
000 Kaloshi MORE >	Toyota ZXC MORE >	50000.0	3	2021-05-17	wertyulo
000 IdeaSoft MORE >	BMW M5 MORE >	50000.0	3	2021-05-17	ann
000 IdeaSoft MORE >	BMW M5 MORE >	50000.0	3	2021-05-17	ann
PE IKEA MORE >	Toyota Corola MORE >	120000.0	5	2021-05-17	For delivery
PE Karazin MORE >	Mercedes-Benz C200 MORE >	70000.0	2	2021-05-17	For student

BEZRUK YURIY [LOG OUT](#)

Authorization ✕

Enter your account data:

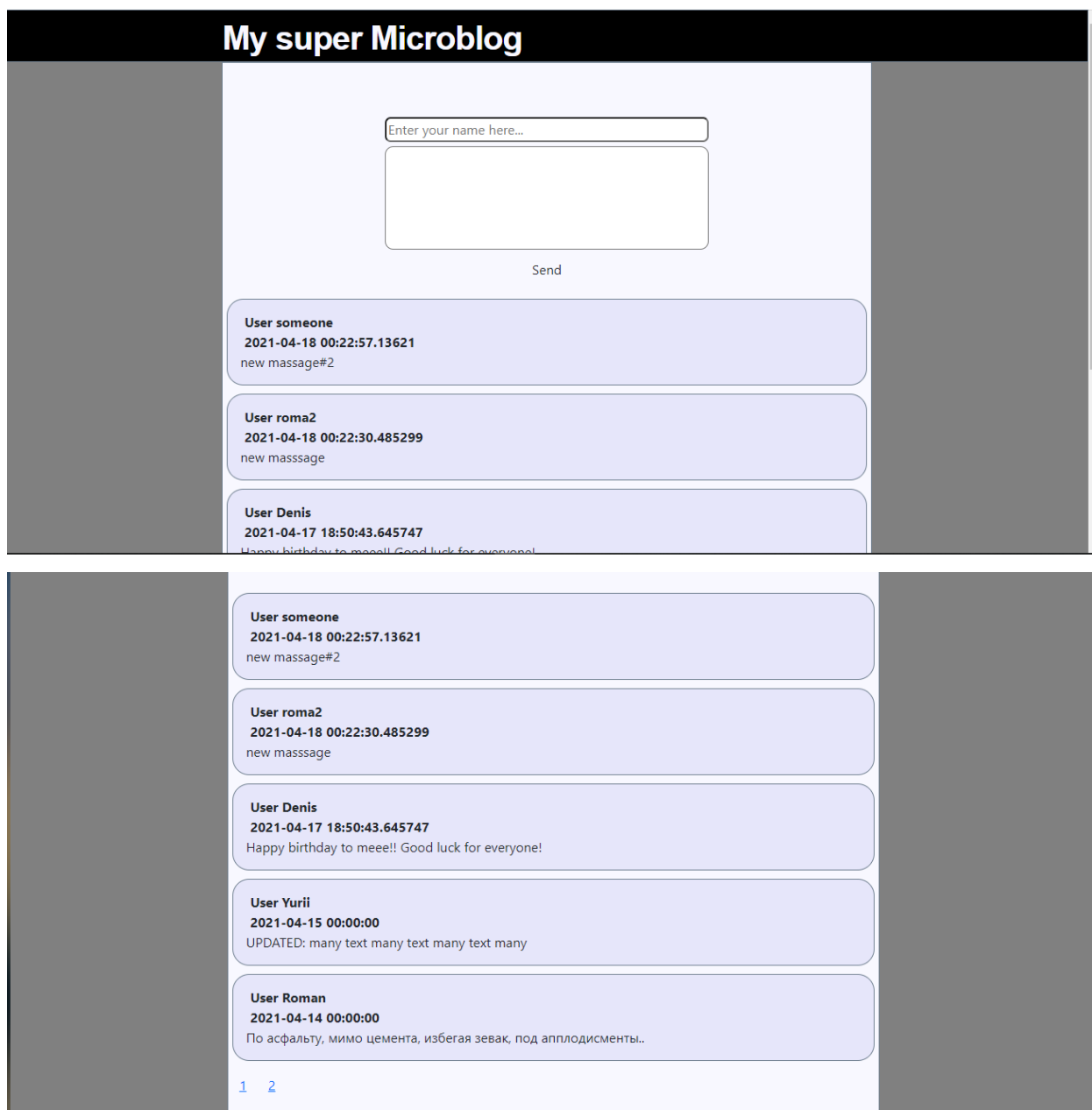
Your name:

Password:

[SIGN IN](#)
[EXIT](#)

If you don't have an account, contact the administrator

3. Веб-приложение «Мой блог» разрабатывалось в качестве лабораторной работы по курсу Баз Данных. Приложение имело возможность различным пользователям регистрировать сообщения в блоге. В качестве БД использовался PostgreSQL, было реализовано партиционирование таблицы постов по датам, что облегчало поиск. Приложение является сервлетом, страница написана при помощи JSP. На странице была предусмотрена пагинация с ограничением по количеству постов на страницу. Скриншоты:



4. Групповой проект по анализу и моделированию требований к приложению. В процессе были проведены интервью с заказчиком, разработана проектная документация в виде функциональных и бизнес-требований, бизнес-правил, ограничений, спецификаций классов, сценариев прецедентов, диаграмм UML Use case, Class, Sequence. Примеры документов и диаграмм:

БІЗНЕС-ПРАВИЛА**Версія 3.1 від 30/06/2021****БП1**

- [БП1] Гра "Дурень" є картковою грою.
- [БП2] У грі використовується колода з 36 карт, по 8 кожної масті:
(1 - піка, 2 - трефа, 3 - бубна, 4 - ~~чирва~~).
- [БП3] Достойнство карт за старшинством в порядку від старших до молодших: (туз, король, дама, валет, 10, 9, 8, 7, 6).
- [БП4] У грі можуть брати участь від 2 до 4 гравців.
- [БП5] Кожному гравцю роздається по 6 карт (по 3 карти 2 рази).
- [БП6] Гравці не бачать карти інших гравців.
- [БП7] Після видачі карт гравцем наступною з верху колоди витягується карта, масть якої встановлює козир для даної гри.
- [БП7.1] Козирна карта стає останньою картою колоди і розташовується так, щоб всі гравці її бачили.
- [БП8] Колоду розташовують закритою стороною догори на козирній карті так, щоб вона не читалася гравцями.
- [БП9] У першій здачі першим ходить гравець з молодшим козирем.
- [БП9.1] У разі коли ні в одного гравця на руках немає козирної карти, першим ходить гравець з молодшою картою наступної за порядком масті (про порядок див. БП2).
- [БП10] Гравець ходить однією картою номіналом, рівним будь-якій серед тих, що лежать на столі і утримує право ходу доки не спасує.
- [БП10.1] Першою картою ходу можна обирати будь-яку.
- [БП10.2] Якщо ведучий гравець (гравець, що ходив) пасує, після нього підіймати карту може наступний після покриваючого гравця за годинниковою стрілкою гравець (напрямок обходу договірний, див. §6 Функціональних вимог)
- [БП10.3] Підняті карти покриваючий гравець теж повинен покрити.
- [БП11] Щоб покрити карту, необхідно з наявних на руках карт покласти на неї старшу карту тієї ж масті, або козир.

СЦЕНАРІЙ ПРЕЦЕДЕНТУ «PLAYING»

Версія 4 від 29/04/2021

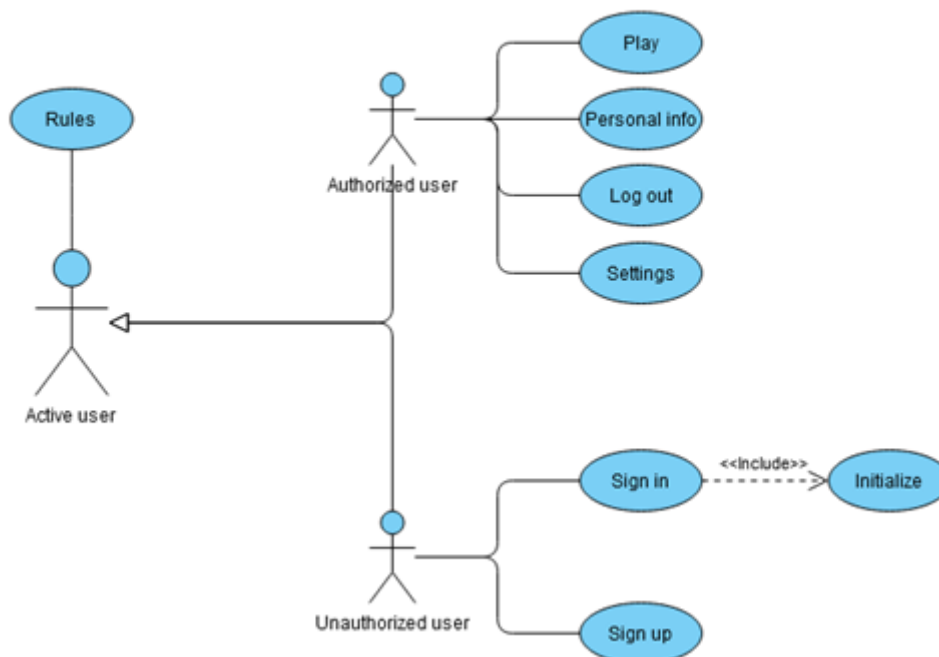
СЦ10

1. Програма виконує сценарій «DRAW».
2. Якщо у колоді ще є карти – програма виконує сценарій «TAKING CARDS».
3. Якщо хабар пішов у відбій – право ходу переходить до наступного гравця, інакше – до гравця після позивавшого.
4. Якщо користувач хоче завершити партію – програма виконує сценарій «END GAME».
5. Пункти 1-5 повторюються, доки у гравців не закінчаться карти.
6. Програма визначає дурнем гравця, який останній залишився з картами, або нічию якщо після останнього розіграшу не залишилося гравців з картами на руках.

ДІАГРАМА ПРЕЦЕДЕНТІВ ТОЧКИ ВХОДУ В ПРОГРАМУ

Версія 03/03/2021

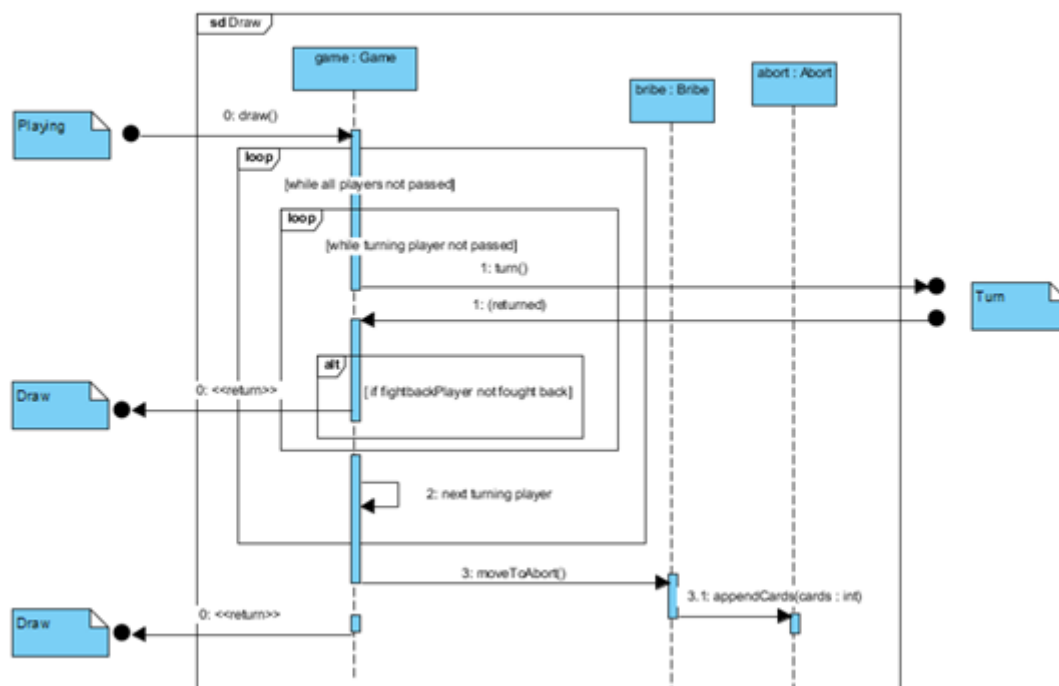
ПД



ДІАГРАМА ПОСЛІДОВНОСТІ ПРЕЦЕДЕНТУ «DRAW»

Версія 2 від 22.05.2021

ПД5



KC-32

проект «Підкидний дурень»

ДІАГРАМА ПОСЛІДОВНОСТІ ПРЕЦЕДЕНТУ «TURN»

Версія 2 від 22.05.2021

ПД6

