



**SPEAR**

Bank system

# Цель создания проекта

- Создать ПО для банкомата, которое может работать с базами данных, а также имеет графический интерфейс
- Научиться работать в команде
- Ознакомиться с тонкостями работы каждой из представленных профессий
- Освоить методы определения требований в ПИ при работе с проектом

# Жизненный цикл

Эволюционная модель

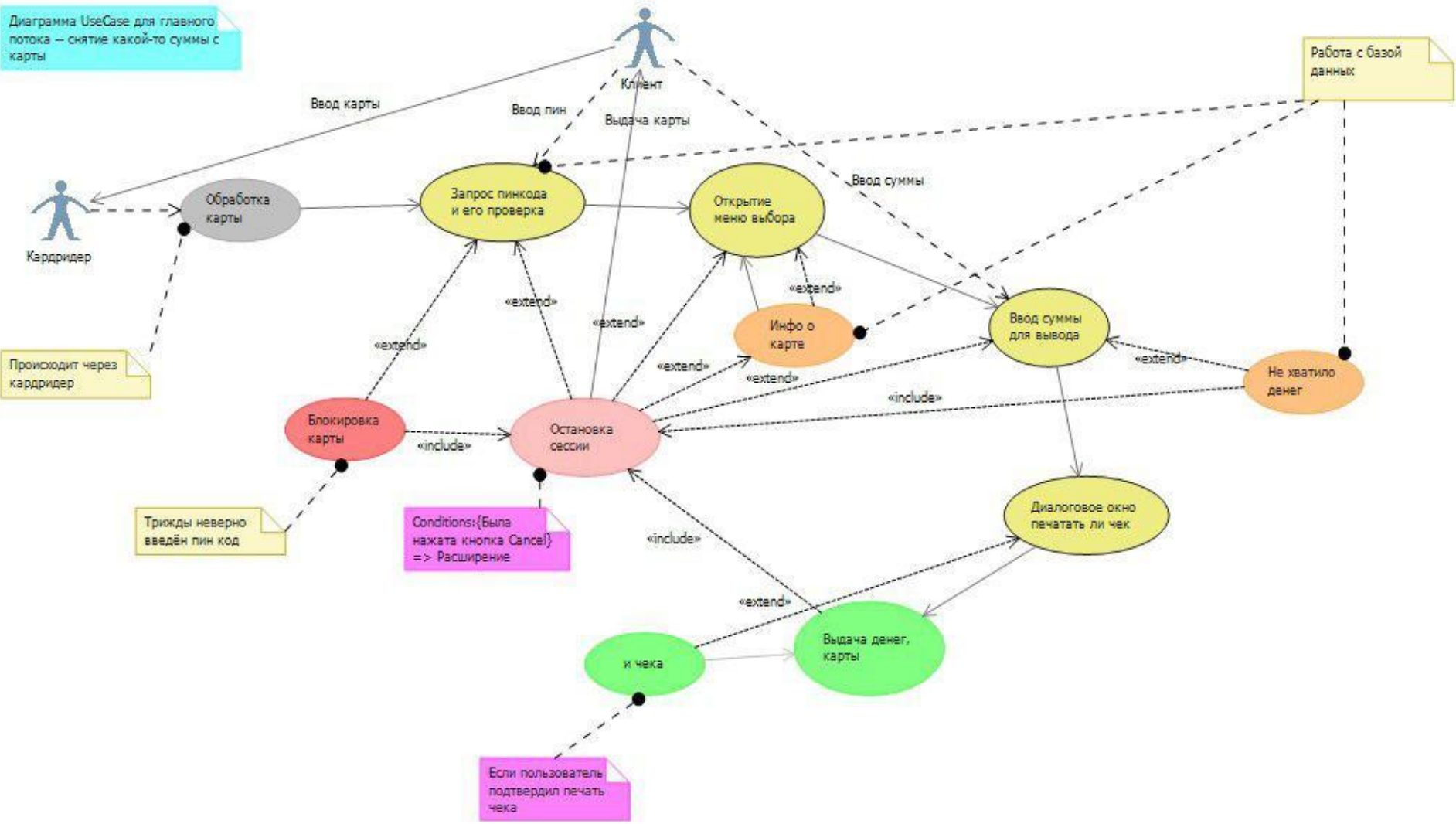


# Спецификации

- **Создание звена банковской системы для банкомата , задачей которого будет обеспечивать GUI и логику работы с базой данных**
- **Работа с банковскими счетами: pin-кодом, кредитными лимитами и балансом этого счета**
- **Обеспечить возможность блокировки карты в случае неверного ввода pin-кода**
- **Системное звено должно достаточно просто интегрируемым в общую цепь**

# USE CASE

Диаграмма UseCase для главного потока – снятие какой-то суммы с карты



Происходит через кардридер

Трижды неверно введён пин код

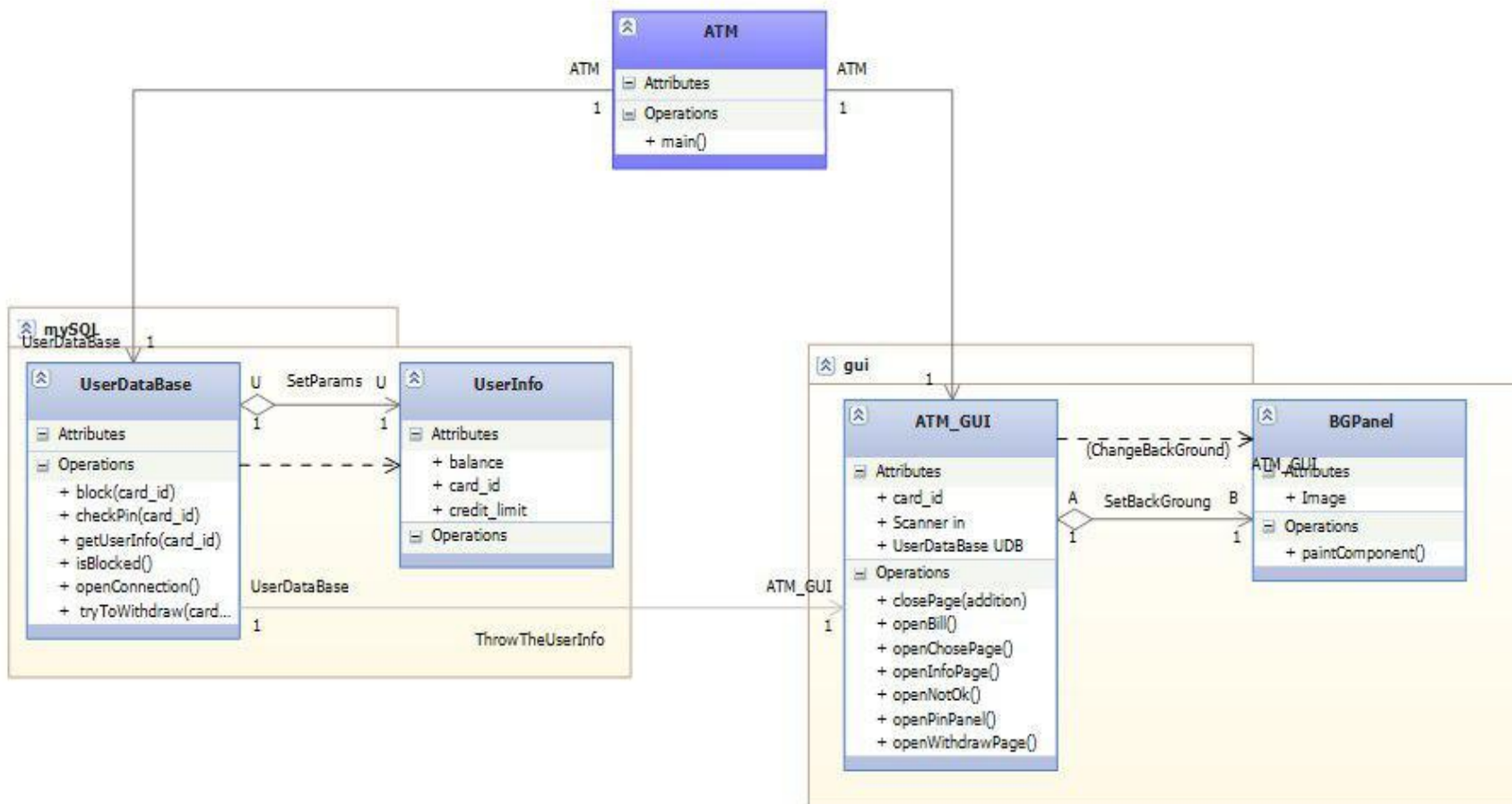
Conditions: {Была нажата кнопка Cancel} => Расширение

Если пользователь подтвердил печать чека

Работа с базой данных

# Диаграмма классов

cd SpearBankSystem



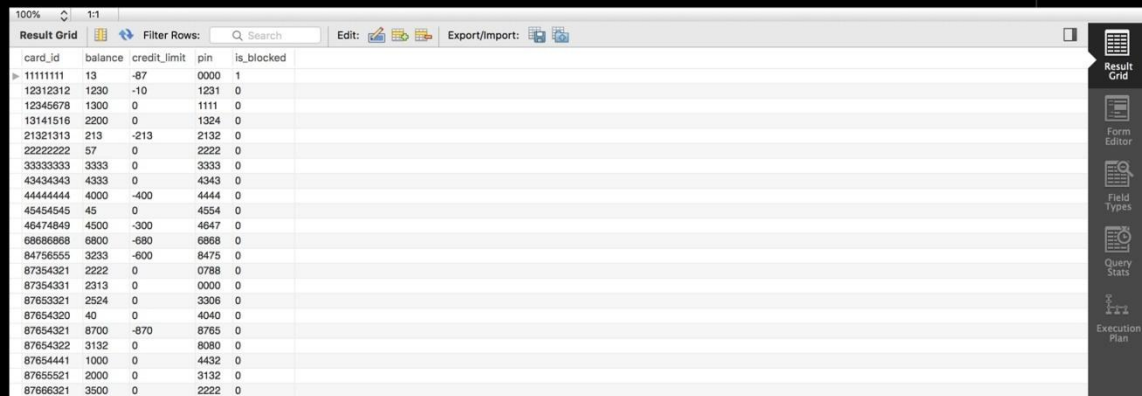
# Скрин кода

```
public boolean tryToWithdraw(int card_id, int sum){ //Снимает баланс если хватает, возвращает 1. Иначе 0.

    try {
        Statement statement = conn.createStatement();

        ResultSet rs=statement.executeQuery("SELECT balance, credit_limit from users WHERE card_id="+card_id+"");
        rs.next();
        double balance=rs.getDouble(1);
        double credit_limit=rs.getDouble(2);
        statement.close();
        double final_balance=balance+Math.abs(credit_limit);

        if(final_balance-sum>=0){
            Statement statement2 = conn.createStatement();
            double new_balance=balance-sum;
            statement2.executeUpdate("UPDATE users SET balance ="+new_balance+" WHERE card_id="+card_id);
            statement2.close();
            return true;
        }
        else{
            return false;
        }
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
    return false;
}
```



card_id	balance	credit_limit	pin	is_blocked
11111111	13	-87	0000	1
12312312	1230	-10	1231	0
12345678	1300	0	1111	0
13141516	2200	0	1204	0
21321313	213	-213	2132	0
22222222	57	0	2222	0
33333333	3333	0	3333	0
43434343	4333	0	4343	0
44444444	4000	-400	4444	0
45454545	45	0	4554	0
46474849	4500	-300	4647	0
68686868	6800	-680	6868	0
84756555	3233	-600	8475	0
87354321	2222	0	0788	0
87354331	2313	0	0000	0
87653321	2524	0	3306	0
87654320	40	0	4040	0
87654321	8700	-870	8765	0
87654322	3132	0	8930	0
87654441	1000	0	4432	0
87655521	2000	0	3132	0
87666321	3500	0	2222	0

# Внешний вид проекта

Enter your pin-code

7	8	9
4	5	6
3	2	1
0	<-	Cancel

Cancel

Информация про счет

Снятие денег

Карта №12345678 Баланс на карте: 1300.0

Кредитный лимит карты: 0.0

Доступная сумма для снятия: 1300.0

Вернуться назад

Cancel

0	100
500	200
1000	Снять деньги
Вернуться назад	Cancel

Печатать ли чек.

Yes

Nope

Заберите свою карту!

Начать новую сессию



# Выводы

- Нашей команде удалось создать ПО для банкомата, которое может работать с базами данных, а также имеет графический интерфейс и тестирование методом Белого ящика.
- Убедились, что последняя версия программы полностью соответствует требованиям заказчика
- Мы провели тестирование методом Черного ящика и в ходе тестирования не обнаружено ошибок при работе интерфейса , соблюдалась логика выполнения программой команд. Все функции взаимодействия с пользователем и отображения информации

# Наша команда

- **Тимченко Марина** - менеджер
- **Владимиров Иван** - кодер
- **Васильев Игорь** - тестировщик
- **Крайнюк Николай** - дизайнер