

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Факультет информационных технологий и программирования

Анализ и проектирование на UML

3 этап. Приемо-сдаточные испытания

Выполнил студент группы М33051
Ефимов Вячеслав Иосифович

САНКТ-ПЕТЕРБУРГ
2020

Описание проекта

Представленный прототип реализует систему продажи билетов на футбольный стадион. В системе существует три вида пользователей: покупатель (Customer), кассир (Cashier), организатор (Organizer). Каждый пользователь в зависимости от своей роли имеет определенный набор операций (см. диаграмму использования). Для выполнения той или иной операции необходимо авторизоваться в систему по заданному никнейму (username) и паролю (password). Далее система по команде /start предоставит список доступных операций, при нажатии на любую из которых будет происходить интерактивное взаимодействие с ботом.

В системе есть контроллер, который перехватывает запросы пользователя и в зависимости от той или иной операции использует остальные классы, определенные на диаграмме классов (см. ниже). Реализованные классы можно условно поделить на 2 типа: классы сущностей и классы DAO. Классы DAO используются для обращения к базе данных PostgreSQL через SQL-запросы, в то время как классы сущностей описывают хранимые сущности в терминах ООП. Помимо этого, для повышения безопасности база данных хранит пароли пользователей в зашифрованном формате MD5.

Диаграмма использования

Данная диаграмма описывает, какие варианты использования существуют для разных ролей в системе.

Покупатель (**Customer**) может:

- Купить билет (buy ticket)
- Вернуть билет (return ticket)
- Пополнить баланс (Add balance)
- Посмотреть список приобретенных билетов (Show tickets)

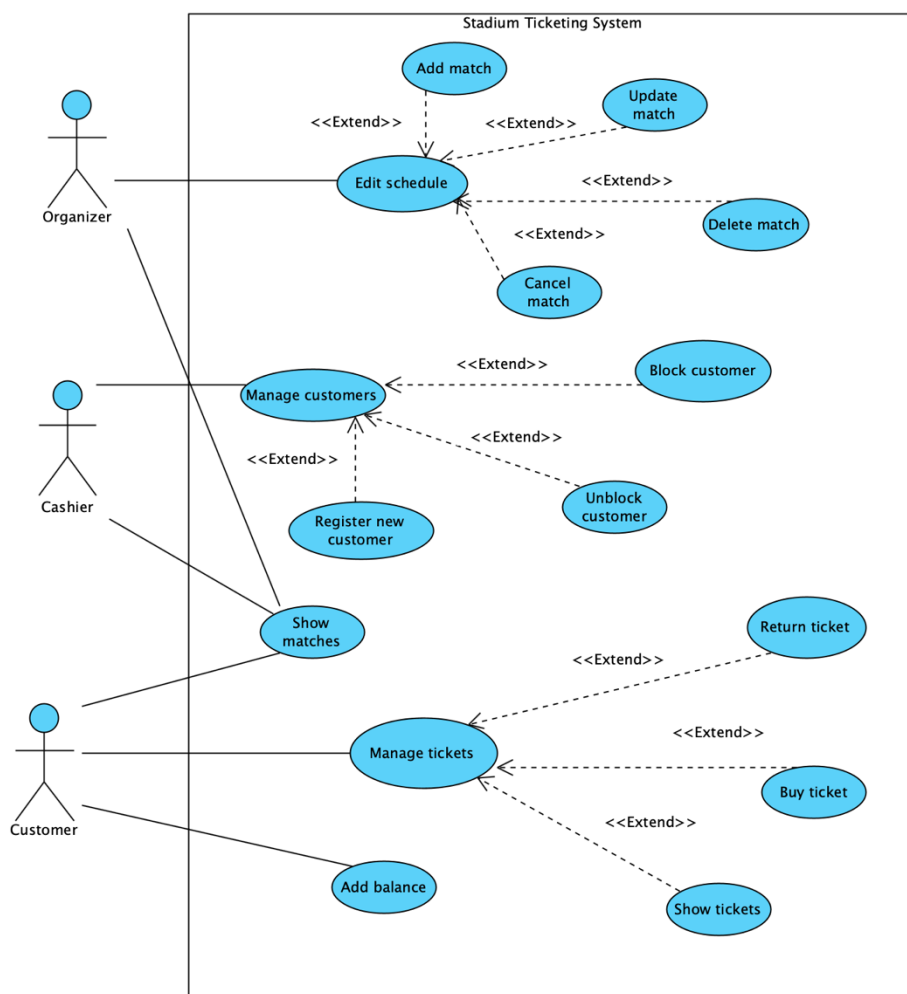
Кассир (**Cashier**) может:

- Зарегистрировать нового покупателя (Register new customer)
- Заблокировать покупателя (Block customer)
- Разблокировать покупателя (Unblock customer)

Организатор (**Organizer**) может:

- Добавить матч (Add match)
- Обновить матч (Update match)
- Удалить матч (Delete match)
- Отменить матч (Cancel match)

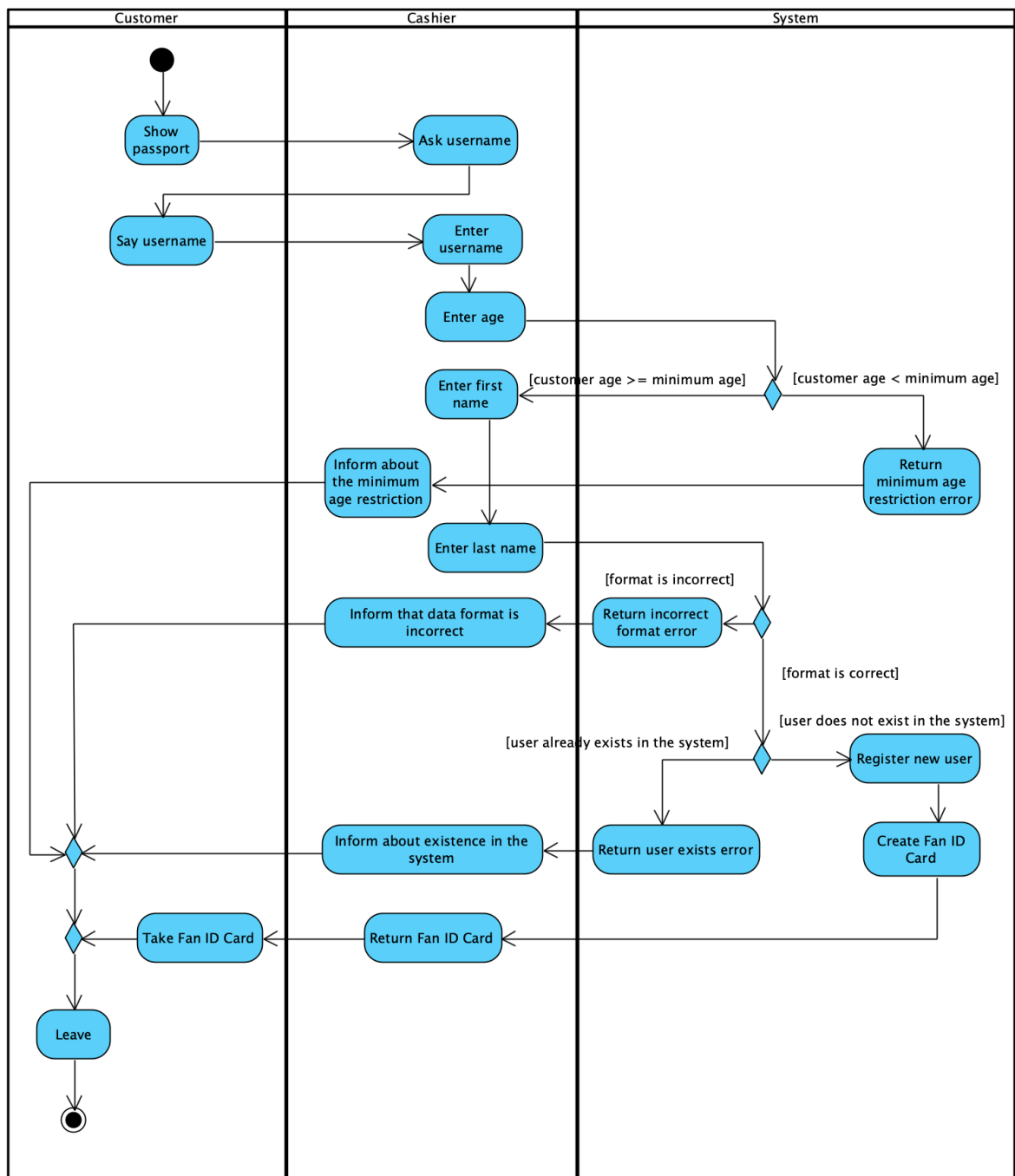
Помимо этого, все пользователи системы также могут просматривать расписание матчей (Show matches).



Диаграммы деятельности

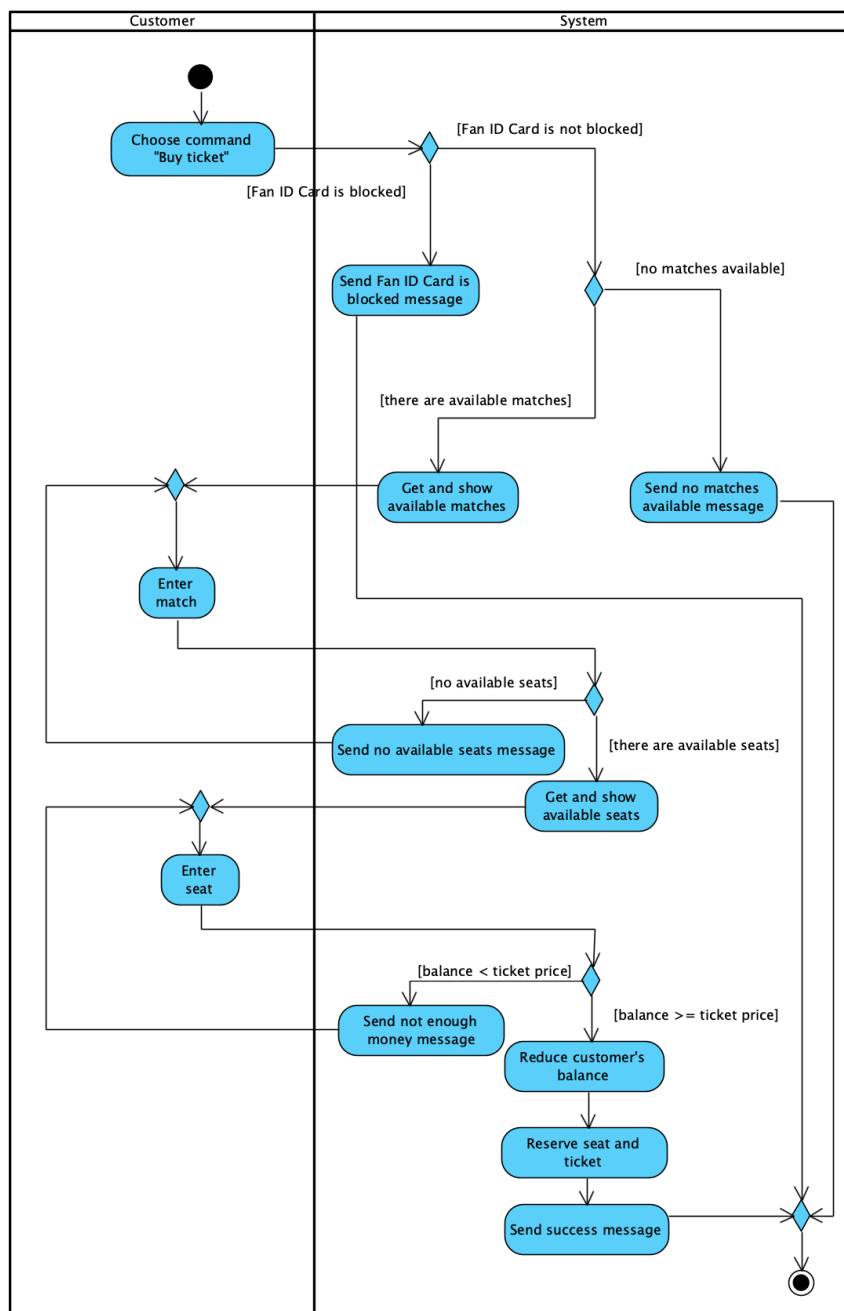
Зарегистрировать нового покупателя (Register new customer)

Данная диаграмма описывает процесс регистрации кассиром нового покупателя, который подошел к кассе. В качестве данных кассир вводит новый никнейм пользователя, возраст, имя и фамилию. Система верифицирует формат вводимых данных и в случае неудачи возвращает сообщение об ошибке. Также системой происходит проверка на то, что покупатель достиг минимального порога возраста. Если пользователя до этого не существовало, то система генерирует Fan ID Card (карточку фаната), сохраняет данные пользователя в базе данных, а кассир выдает покупателю искомый Fan ID Card.



Купить билет (Buy ticket)

Для покупки билета система проверяет не заблокирован ли Fan ID Card покупателя. Если все в порядке, то она обращается к базе данных, чтобы получить список матчей. В случае отсутствия каких-либо матчей покупатель получает соответствующее сообщение. В противном случае среди предложенного списка матчей ему предлагается ввести идентификатор матча, который он хотел бы посетить. После ввода матча система выдает список доступных мест на стадионе и пользователь выбирает одно из них, далее, учитывая выбранное место, система вычитает из баланса Fan ID Card нужную сумму и резервирует для покупателя искомое место на стадионе. Если все места на данный матч заняты, то покупателю будет предложено выбрать другой матч.



Вернуть билет (Return ticket)

Для возврата пользователем билета система убеждается, что Fan ID Card покупателя не заблокирован. Если это так, то покупатель далее вводит идентификатор одного из своих приобретенных билетов и система отменяет бронь на место, указанное в билете, подсчитывает сумму возврата и увеличивает баланс Fan ID Card покупателя.

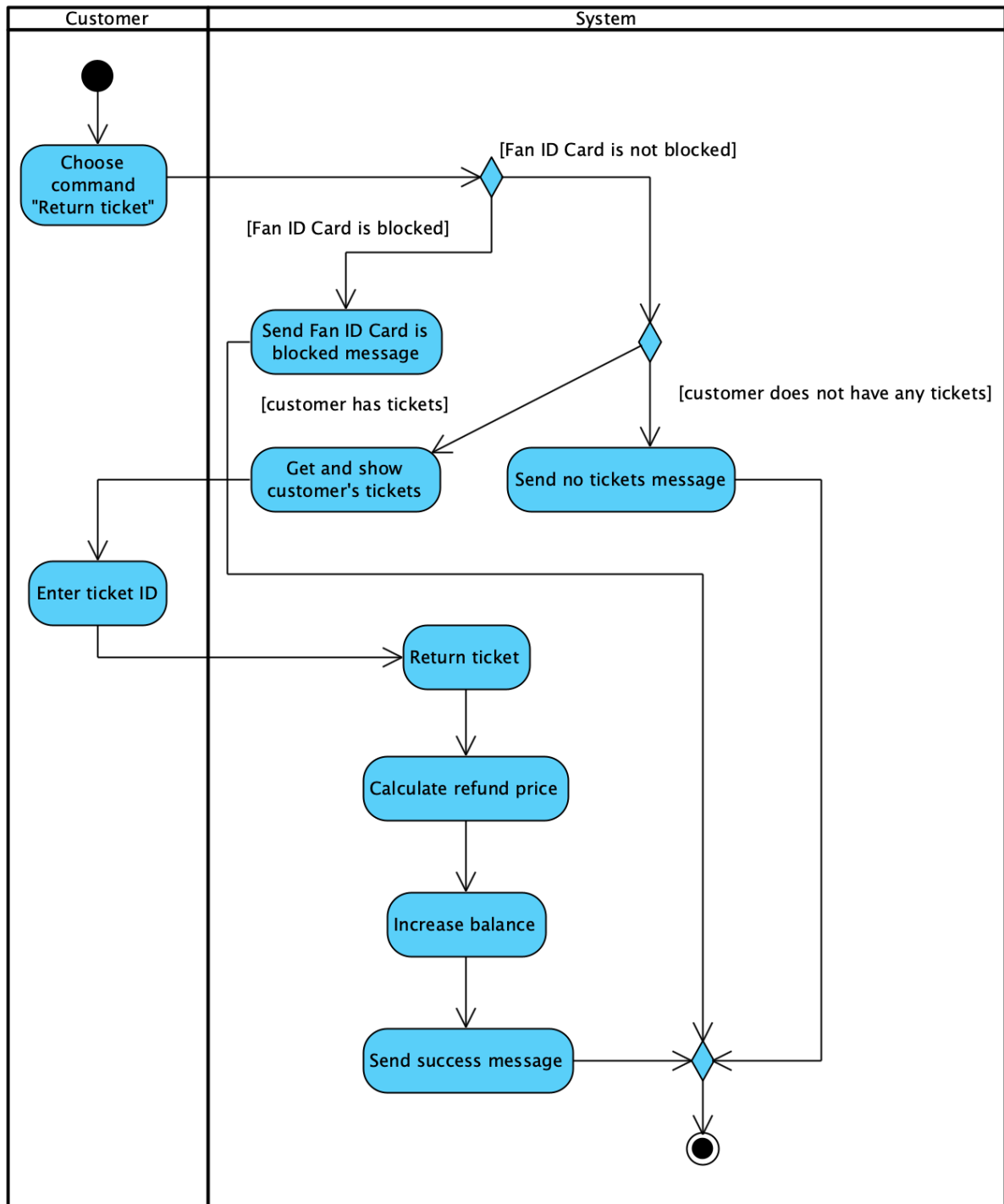


Диаграмма классов

Данная диаграмма описывает классы, которые присутствуют в системе, а также связи между ними.

- **DAO** – абстрактный класс, использующийся для обращения к базе данных
- **PersonDAO** – наследуется от DAO, отвечает за взаимодействие с таблицей пользователей системы (person)
- **FanIDCardDAO** – наследуется от DAO, отвечает за взаимодействие с таблицей карточек фанатов Fan ID Card (cards)
- **MatchDAO** – наследуется от DAO, отвечает за взаимодействие с таблицей матчей (matches)
- **TicketDAO** – наследуется от DAO, отвечает за взаимодействие с таблицей билетов системы (tickets)
- **Person** – абстрактный класс, хранящий основную информацию о пользователе в системе
- **Customer** – наследуется от Person, представляет покупателя в системе
- **Cashier** – наследуется от Person, представляет кассира в системе
- **Organizer** – наследуется от Person, представляет организатора в системе
- **Role** – перечисление, используется для отображения роли пользователя в системе
- **Match** – класс, хранящий информацию о матче
- **MatchType** – перечисление, представляет из себя игровую стадию матча (игра в группе, четвертьфинал, полуфинал, финал)
- **Ticket** – абстрактный класс, хранящий основную информацию о билете
- **SingleTicket** – наследуется от Ticket, используется для представления обычного билета на конкретный матч
- **Seat** – класс, отвечающий за конкретное посадочное место на стадионе

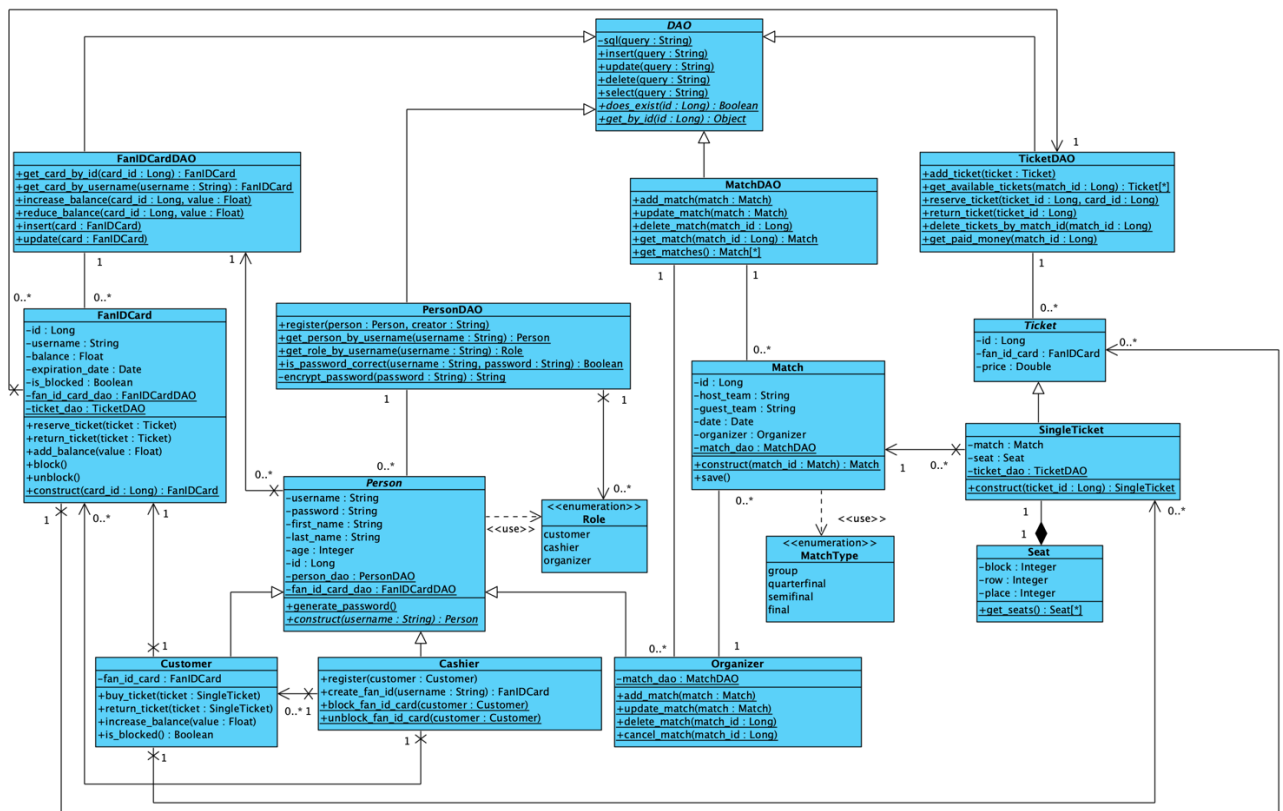
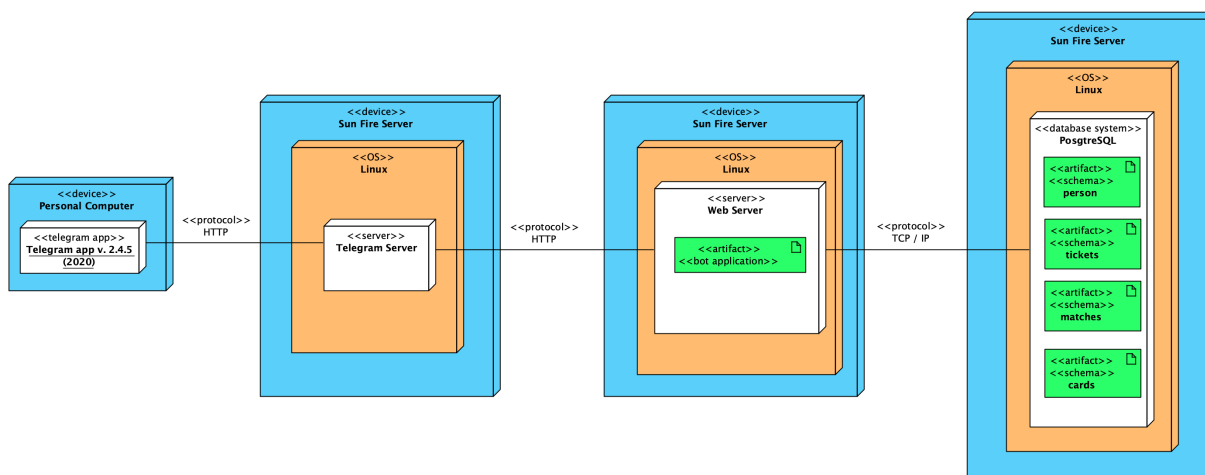


Диаграмма развертывания

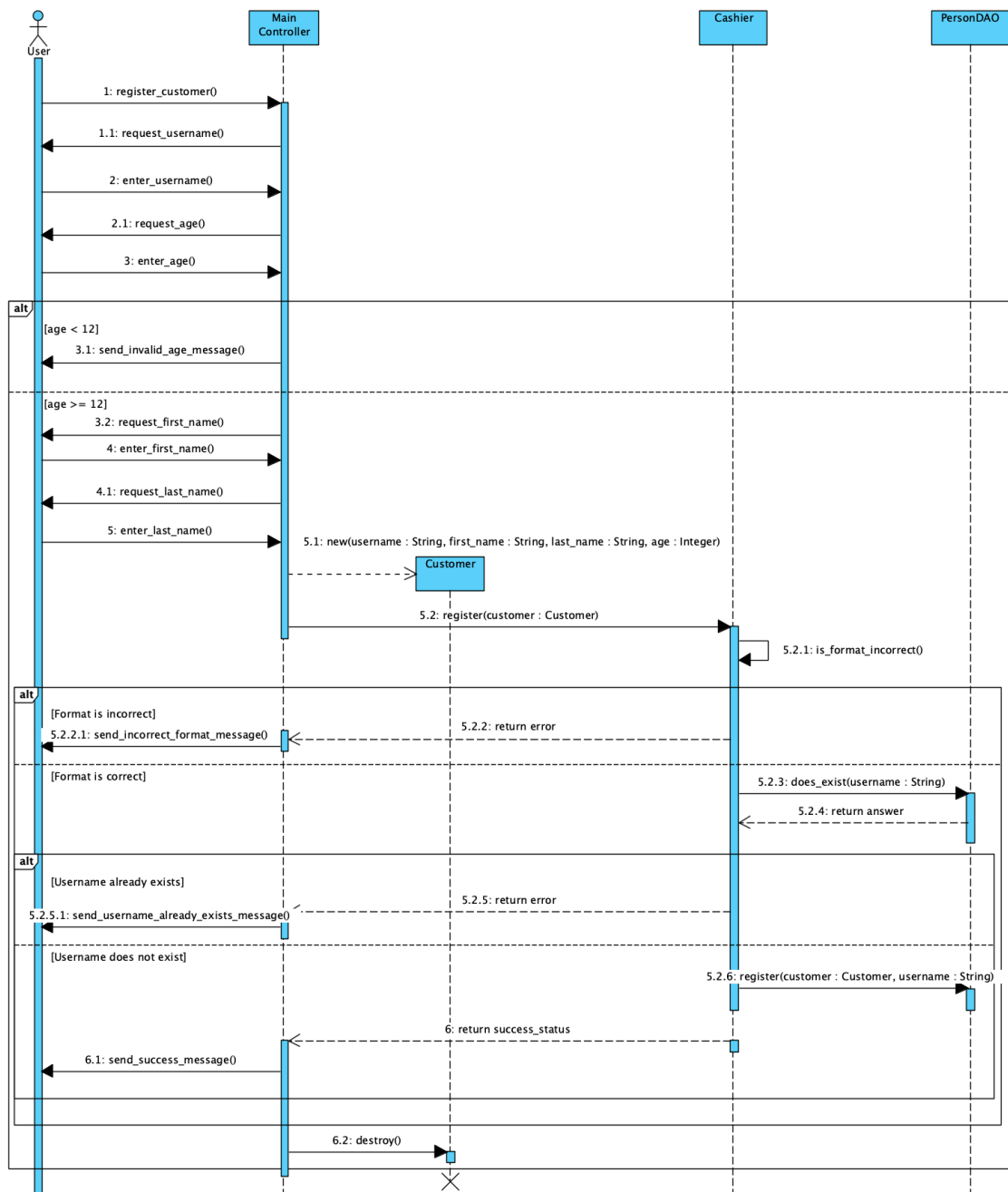
Диаграмма развертывания описывает, как физически происходит взаимодействие между конечным пользователем и приложением. Заходя в мессенджер “Telegram”, пользователь, вводя те или иные команды, и далее нужные запросы поступают по HTTP-протоколу на Telegram Server, который перенаправляет их на веб-сервер, где развернуто само приложение. Для взаимодействия с базой данных приложение обращается по протоколу TCP / IP на сервер, куда поступают и возвращаются обратно искомые данные.

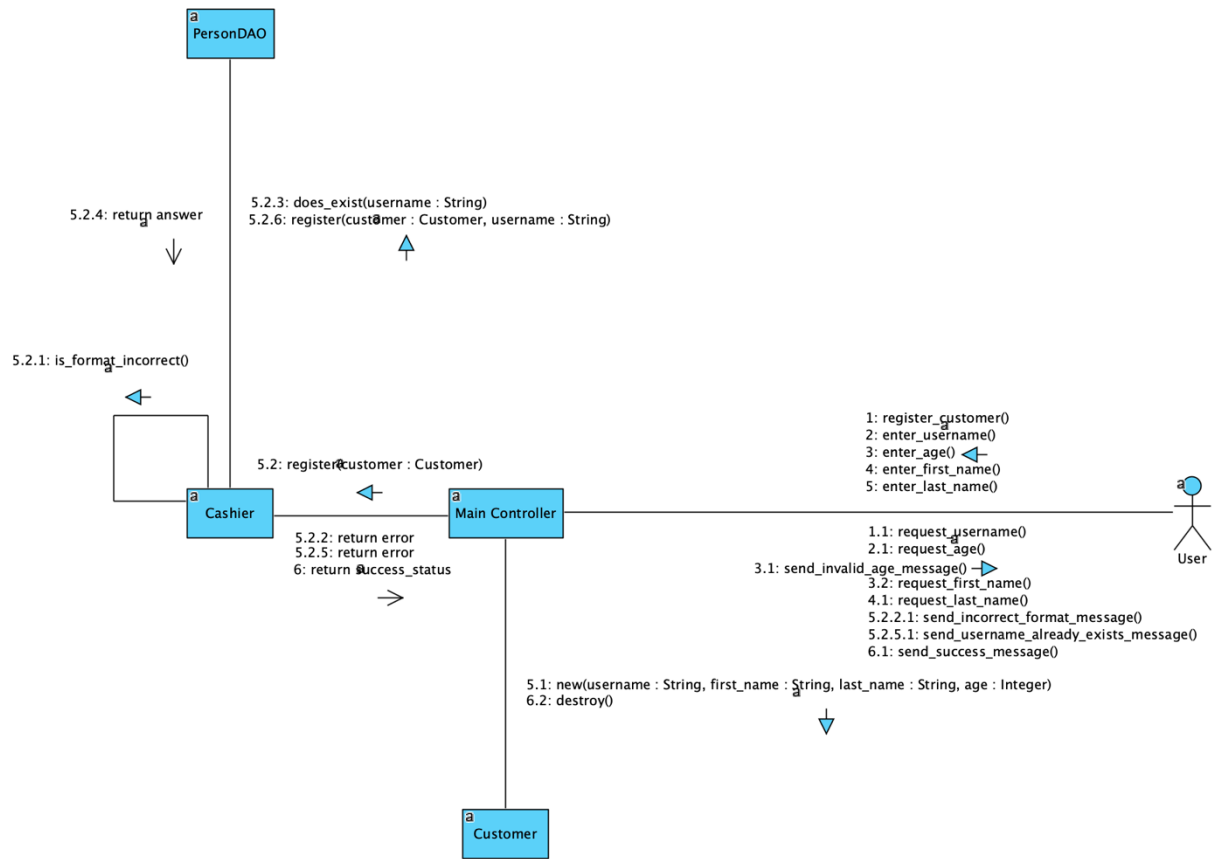


Диаграммы последовательности и коммуникации

Зарегистрировать нового покупателя (Register new customer)

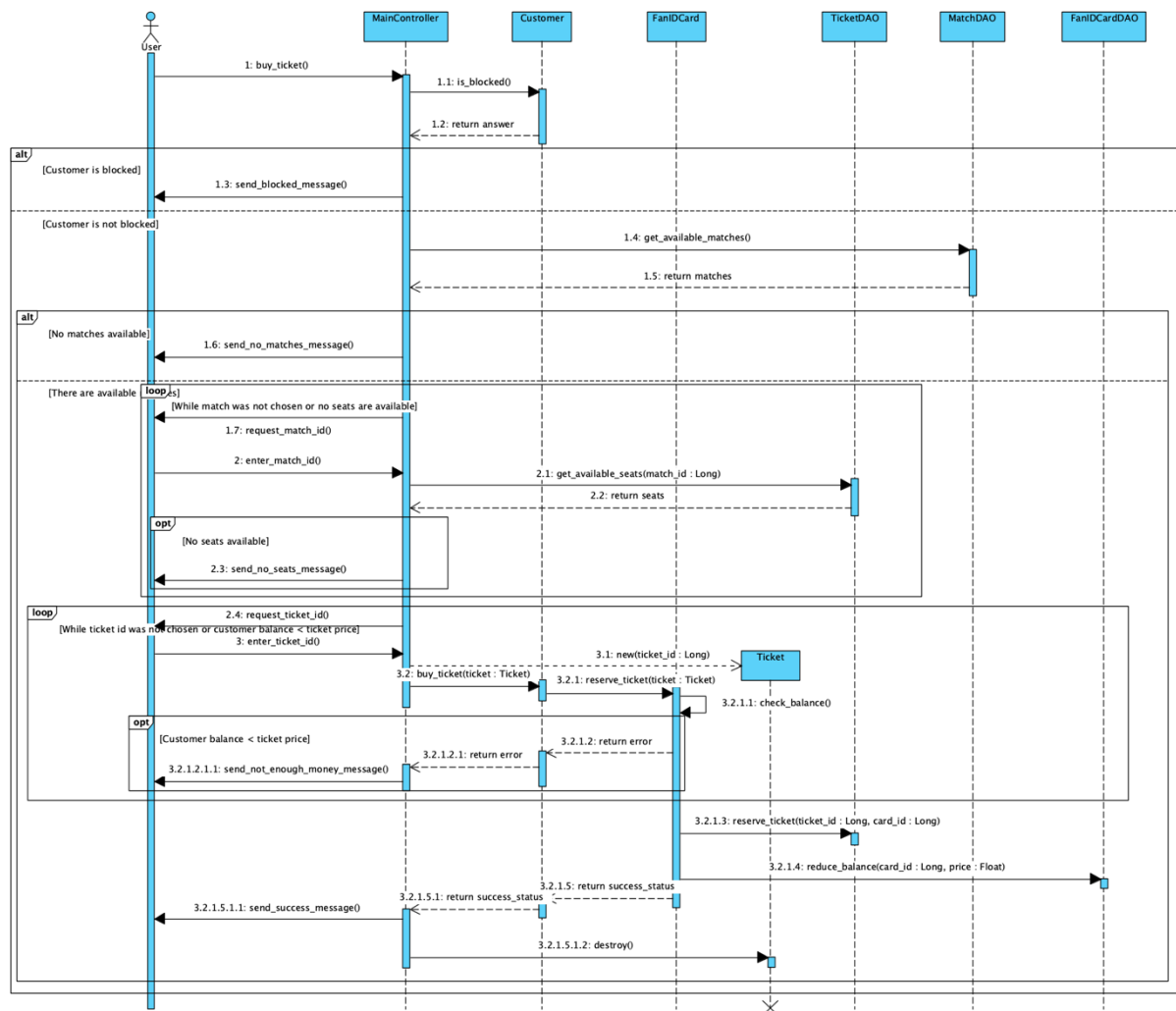
В процессе регистрации нового покупателя контроллер (Main Controller) перехватывает соответствующие вызовы методов от пользователя и далее создает новый объект покупателя, передавая его затем классу кассира (Cashier). Кассир определяет, удовлетворяет ли новый покупатель требованиям в системе и в успешном случае обращается к классу PersonDAO, чтобы сохранить созданного покупателя в базу данных. В результате выполнения любого сценария Main Controller получает статус операции и возвращает его в виде сообщения пользователю.

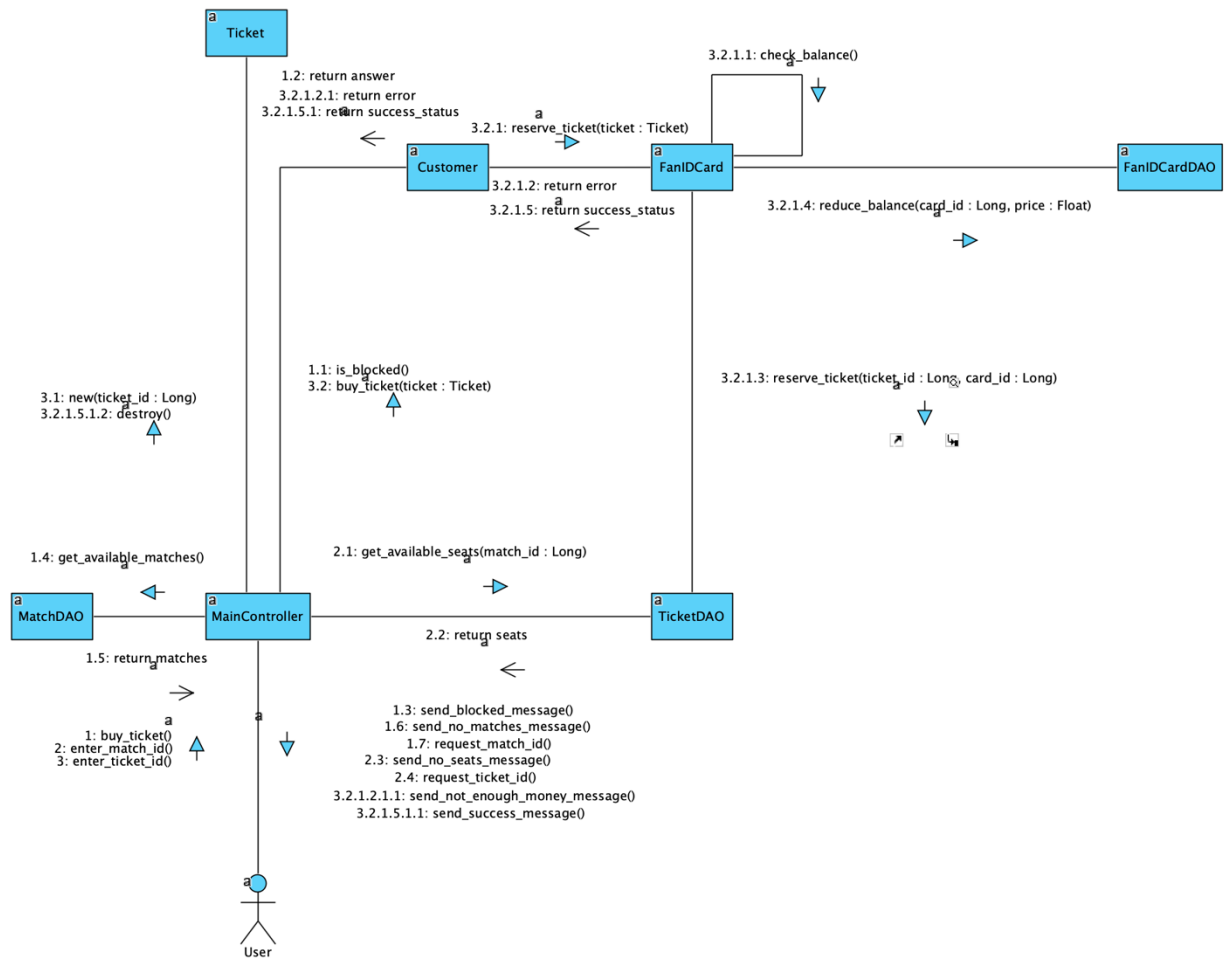




Купить билет (Buy ticket)

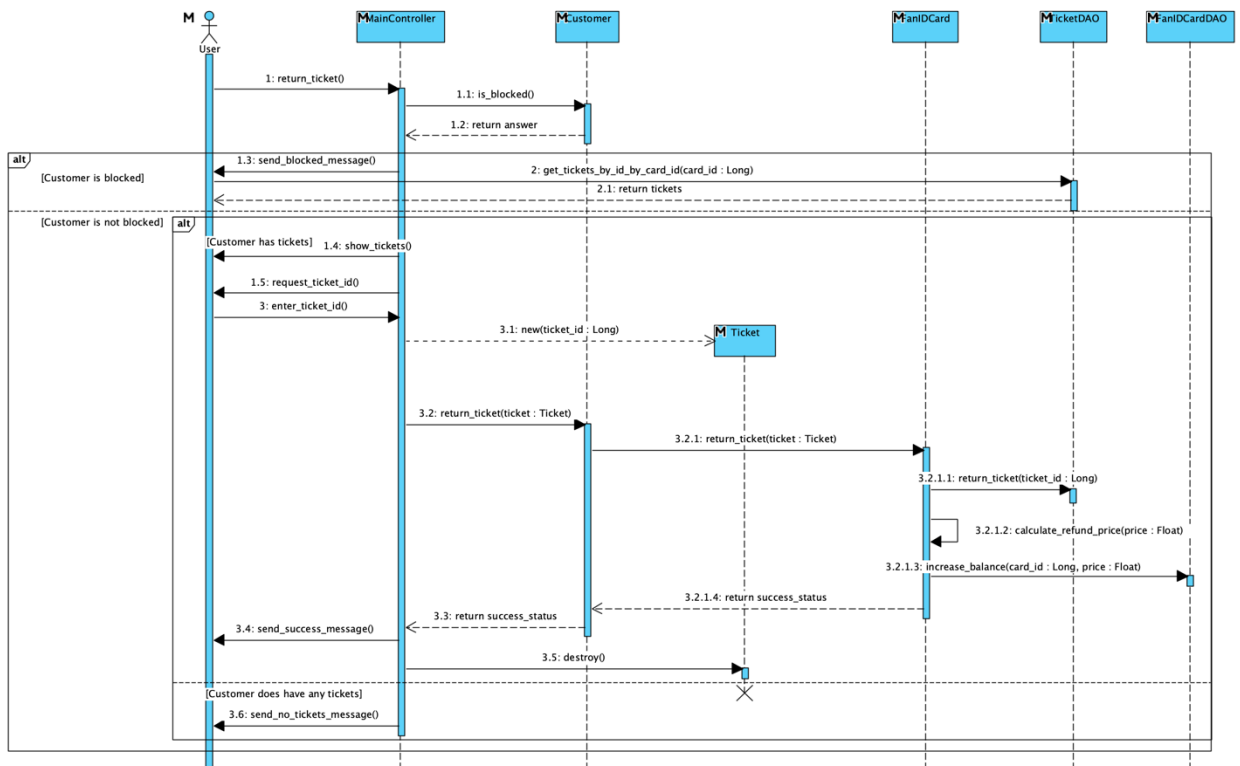
Для покупки билета контроллер обращается к MatchDAO, чтобы получить список доступных матчей, далее происходит обращение к TicketDAO для получения доступных билетов с местами. В случае успешном случае в классе FanIDCard создается объект Ticket, id которого передается в классы MatchDAO и FanIDCardDAO для сохранения информации в базу данных. Завершение метода происходит возвратом пользователю сообщения от контроллера о (не-)успешном проведении операции.

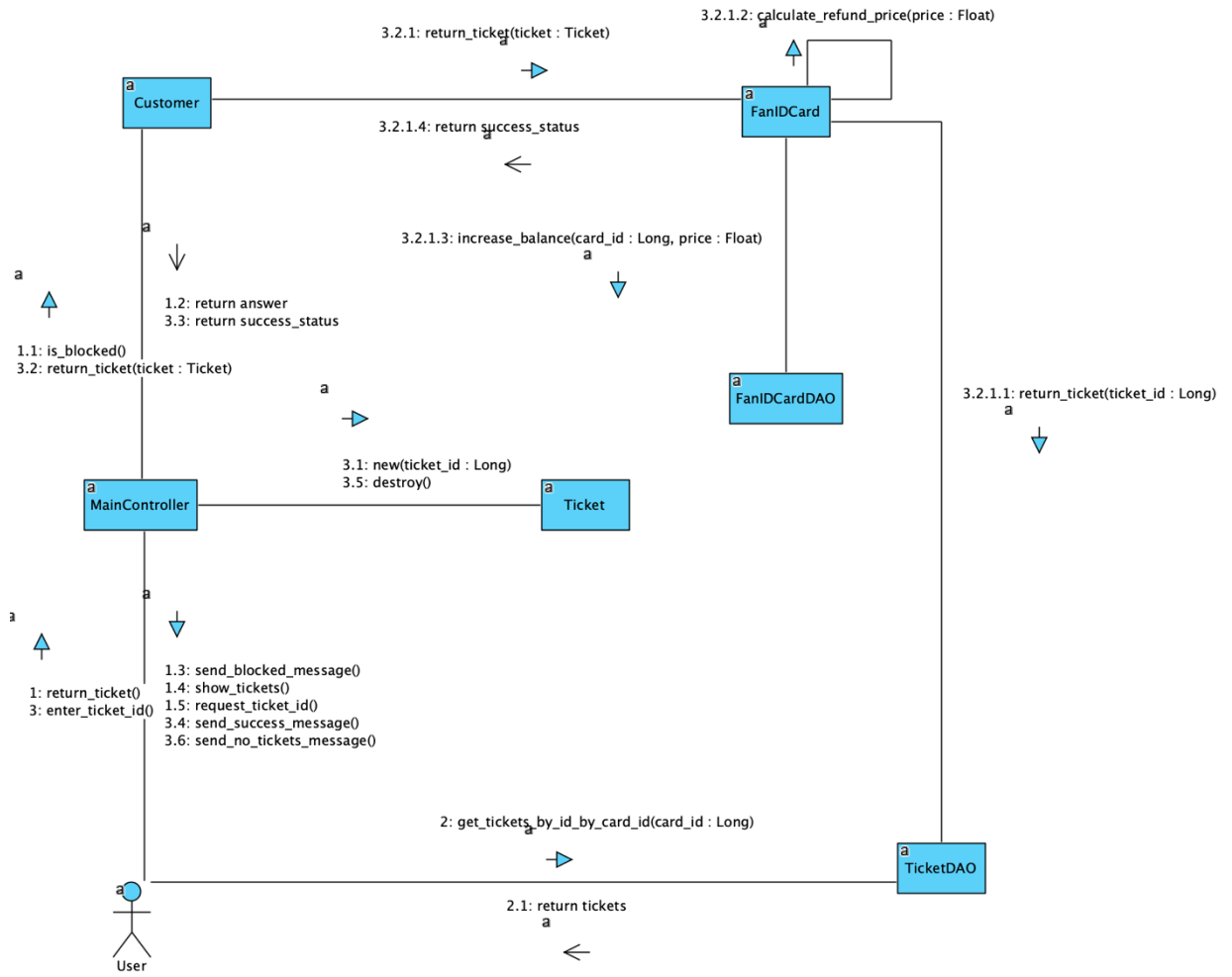




Вернуть билет (Return ticket)

Для возврата билета контроллер создает объект Ticket и передает его в класс Customer путем вызова метода `return_ticket()`. Далее из класса Customer вызывается метод `return_ticket()` у поля типа `FanIDCard`. Чтобы сохранить изменения в базу данных о билете, класс `FanIDCard` обращается к классам `TicketDAO` и `FanIDCardDAO` для обновления информации. Завершение метода происходит возвратом пользователю сообщения от контроллера о (не-)успешном проведении операции.

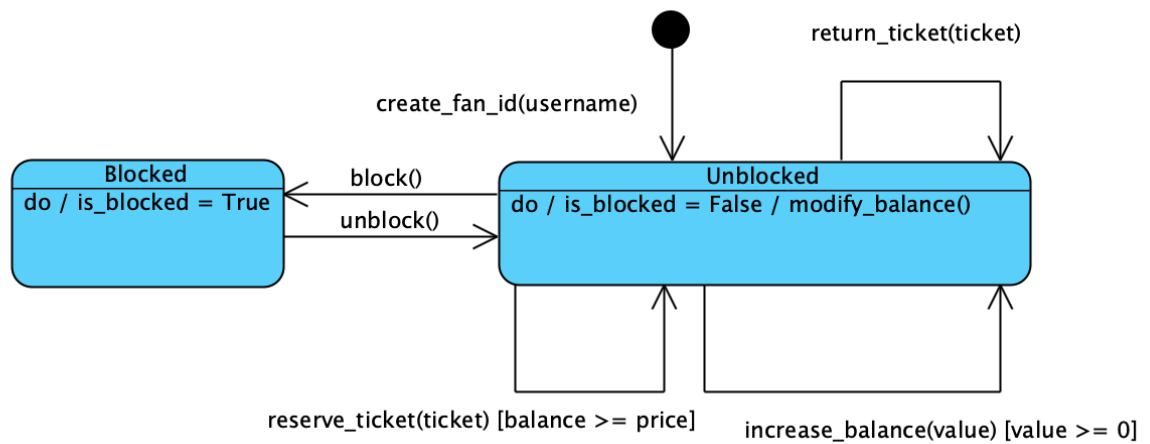




Диаграммы состояния

Fan ID Card (карточка фаната)

Fan ID Card (карточка фаната) имеет два основных состояния: Blocked (заблокирована) и Unblocked (разблокирована). Переход между этими состояниями осуществляется вызовами соответствующих методов block() и unblock(). Помимо этого, если Fan ID Card разблокирована, то у нее могут быть вызваны методы покупки, сдачи билета или пополнения баланса, не влияющие на изменение ее состояния.



Ticket (билет)

В процессе добавления матча генерируется билет, соответствующий определенному месту на стадионе. Изначально он никому не принадлежит и может быть приобретен одним из покупателей. После его покупки билет может быть возвращен обратно в продажу. Также при удалении матча автоматически удаляется и любой билет на этот матч.

