**Software Architecture Document (v 01.11.23)**

**1.** **Introduction (Введение)**

Представленная в данном документе информация определяет слои, на которые делится приложение, а также интерфейсы их взаимодействия. Благодаря этим данным возможно оценить сложность системы, возможности ее улучшения и принять меры по дальнейшей поддержке и обслуживанию/масштабированию в долгосрочной перспективе.

**1.1** **Purpose**

Данный документ содержит описание архитектуры программного обеспечения с целью формирования четкого представления структуры, компонентов и взаимодействий внутри разрабатываемой системы. Документ выступает в качестве образца, позволяющего визуализировать дизайн и организацию системы в целом. В качестве эффективного представления информации выступают UML-диаграммы.

**1.2 Scope (Область применения)**

Документ разрабатывается в рамках проекта “Финист” - системе знакомств для серьезных отношений в старославянском сеттинге на основе RUP методологии. Предназначен для заинтересованных лиц, лиц участвующих в разработке и других членов команды.

**1.3 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)**

Для дальнейшего понимания документа важно изучить роли пользователей системы, которые они принимают во время исполнения бизнес-процесса. Значение всех терминов дано в глоссарии проекта:

[Glossary](https://docs.google.com/document/d/1reBBvZCW8-QsYr5F5Ixum-bbKSkMxmGXP-yf2m_T3RU/edit)

**1.4 References (Ссылки)**

[SRS](https://docs.google.com/document/d/1KZPhcczRg4XHN3is2uTxyx07Gj8I-aY-/edit?usp=drive_link&ouid=108798662073330685189&rtpof=true&sd=true)

[UCS](https://docs.google.com/document/d/1ydDMM3psIyYvH4MX7Mlkt2e8n_XKm07K/edit?usp=drive_link&ouid=108798662073330685189&rtpof=true&sd=true)

**1.5 Overview (Обзор документа)**

**Вторая глава** описывает архитектуру разрабатываемой системы в целом.

**Третья глава** описывает цели и ограничения архитектуры системы.

**Четвертая глава** содержит описание основных сценариев использования системы разными типами пользователей.

**Пятая глава** содержит описание слоев, на которые делится приложение, а также интерфейсов их взаимодействия.

**Шестая глава** описывает систему как совокупность взаимодействующих процессов, привязанных к определенным моментам времени.

**Седьмая глава** содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом.

**Восьмая глава** содержит подробное описание системы в уже реализованном виде.

**Девятая глава** описывает основные характеристики измерения производительности системы и их границы, которые могут оказать влияние на архитектуру.

**Десятая глава** описывает, каким образом архитектура системы удовлетворяет ее показателям качества - масштабируемости, надежности, мобильности, безопасности и т.д.

**2.** **Architectural Representation (Представление архитектуры)**

| ***Diagram\View*** | ***Use Case View*** | ***Logical View*** | ***Implementation view*** | ***Process view\*\**** | ***Deployment View*** |
| --- | --- | --- | --- | --- | --- |
| ***Use Case Diagram*** | + |  |  |  |  |
| ***Class Diagram*** | + (Взаимодействие сущностей) | +  (Описание основных классов и интерфейсов их взаимодействия) | +  (Полное описание классов с указанием их методов/полей, указать типы связей между классами) |  |  |
| ***Activity Diagram*** | +  (Абстрактное описание) | +  (Более подробное описание, уровни взаимодействия) | +  (Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  |  |
| ***State Machine Diagram*** | +  (Абстрактное описание) | +  (Более подробное описание, уровни взаимодействия) | +  (Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  |  |
| ***Sequence Diagram*** | +  (Абстрактное описание) | +  (Более подробное описание, уровни взаимодействия) | +  (Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  |  |
| ***Cooperative Diagram*** | +  (Абстрактное описание) | +  (Более подробное описание, уровни взаимодействия) | +  (Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов). |  |  |
| ***Package Diagram*** |  | + | - |  |  |
| ***Data Base Diagram*** |  |  | +  (Полная ER модель базы данных  + её даталогическая модель) |  |  |
| ***Deployment Diagram*** |  |  |  |  | +  (Подробная диаграмма развертывания с указанием характеристик машин и интерфейсов взаимодействия) |
| Timeline diagram |  |  |  | + |  |

*\*Activity, Sequence, Cooperative и State Machine диаграммы составляются на основе одного прецедента (каждый тип диаграмм - на основе своего).*

*\*\*Всё представление описывается только в случае, если в системе есть процессы, жестко привязанные к определенным моментам времени (пример - наступление нового месяца, времени суток и т.д.)*

**3.** **Architectural Goals and Constraints (Цели и ограничения архитектуры)**

Технические ограничения архитектуры прямо следуют из оборудования заказчика. ПО будет развёрнуто на сервере Helios:

* FreeBSD 13
* 128GB ram / 560GB space / Intel Xeon E5-2643 3.3GHz / 16 cores
* PSQL 14
* Java 17
* Фронтенд будет упакован в jar (war), сервер бекенда будет использоваться как сервер для фронтенда
* Firefox 117, Chrome 117, Safari 17
* Стабильное интернет-соединение (LTE минимум)

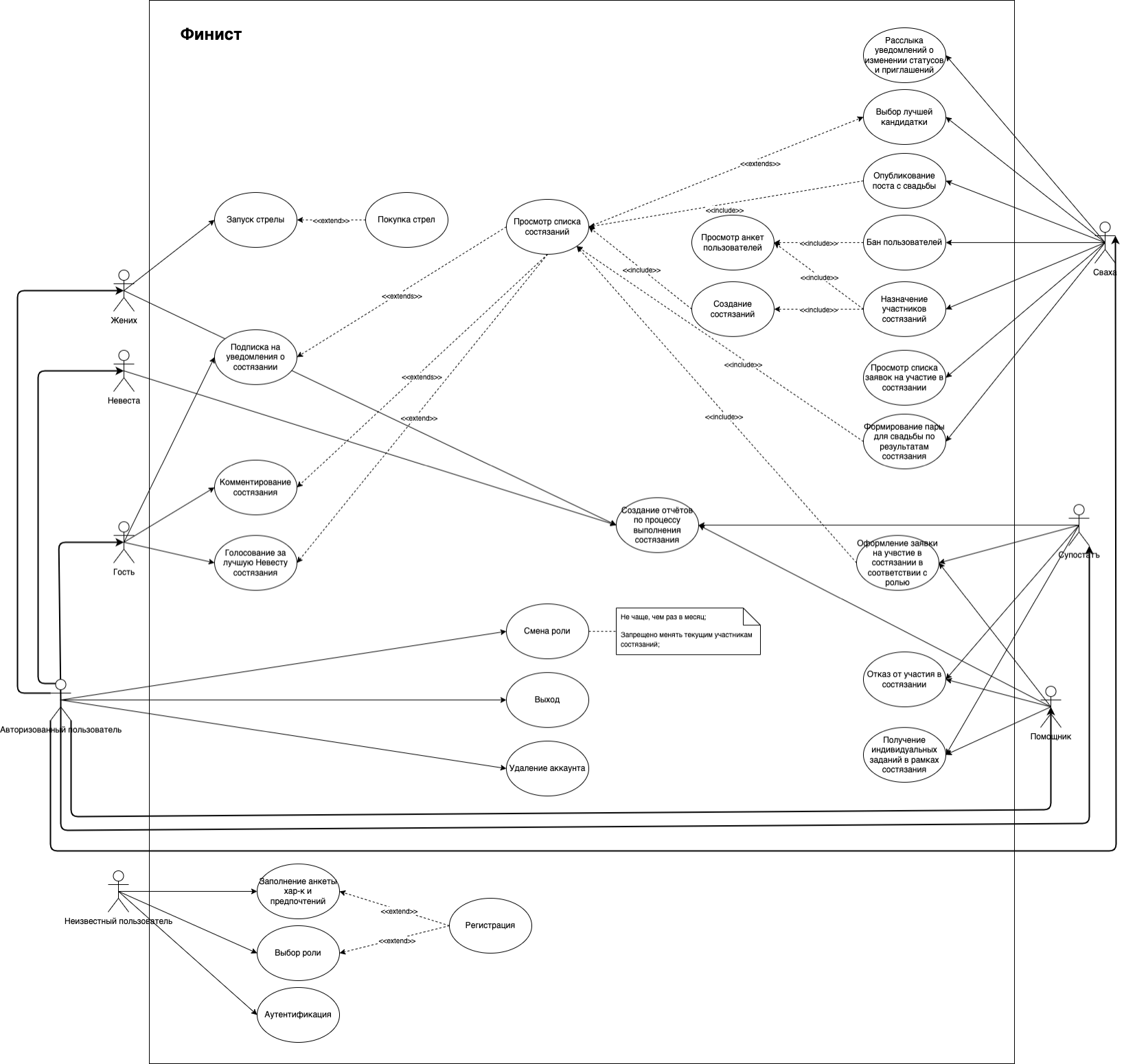
Ни один из useCase-ов не создаёт особых архитектурных ограничений.

Система поддерживает как горизонтальное, так и вертикальное масштабирование средствами фреймворка Sprint, встроенного сервера и PSQL, путём развёртывания дополнительных инстансов программы и подключения к общей базе данных.

**4.** **Use-Case View**

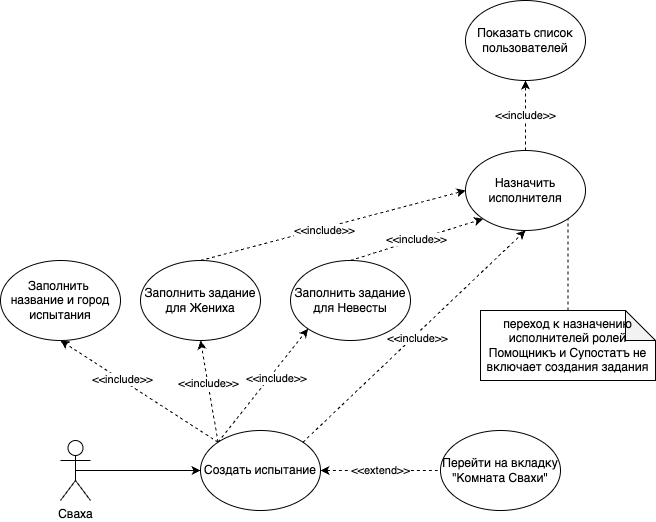
*[Данный раздел содержит описание основных сценариев использования системы разными типами пользователей. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]*

Глобальная диаграмма UseCase:

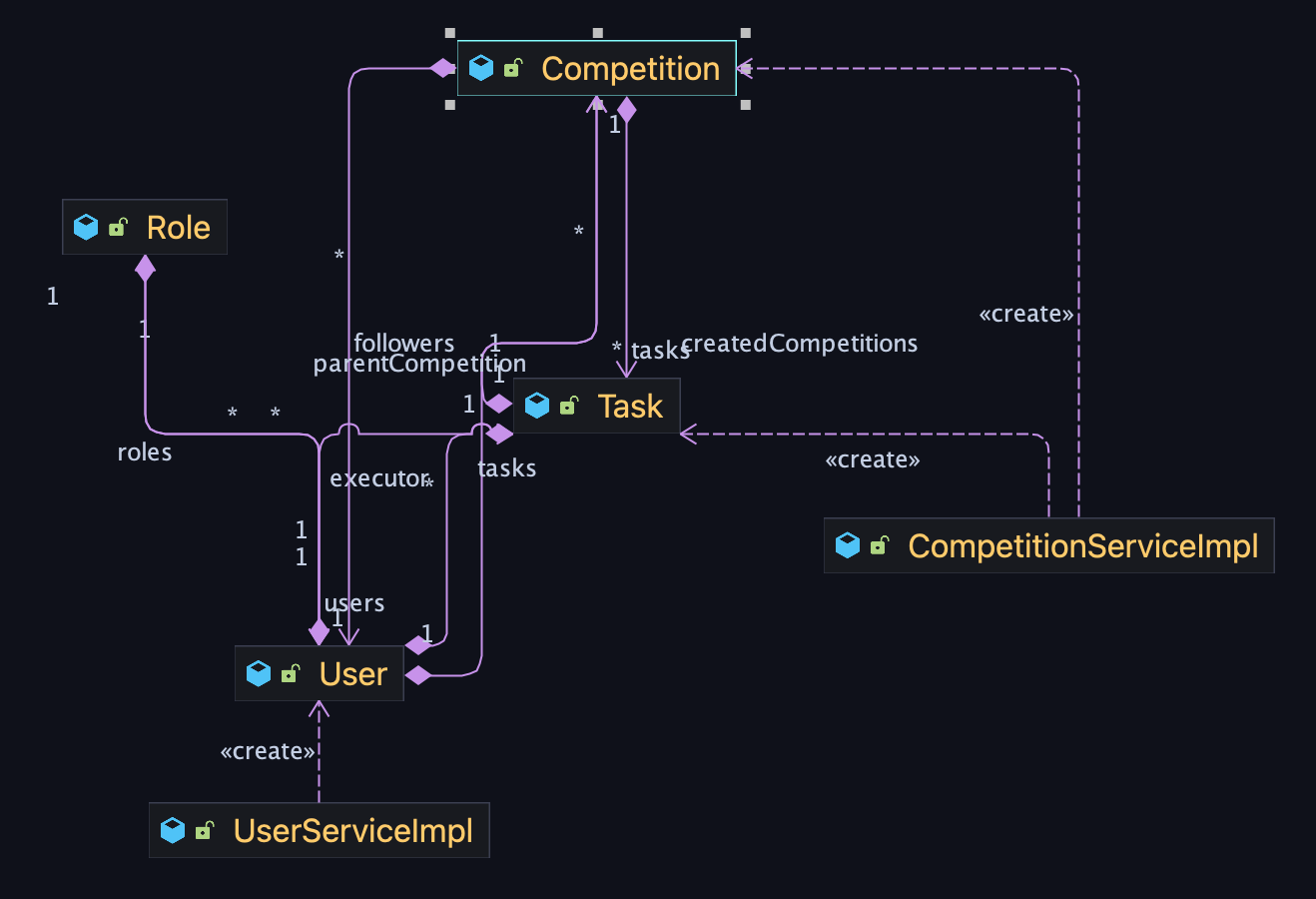


**18. Создание испытания**

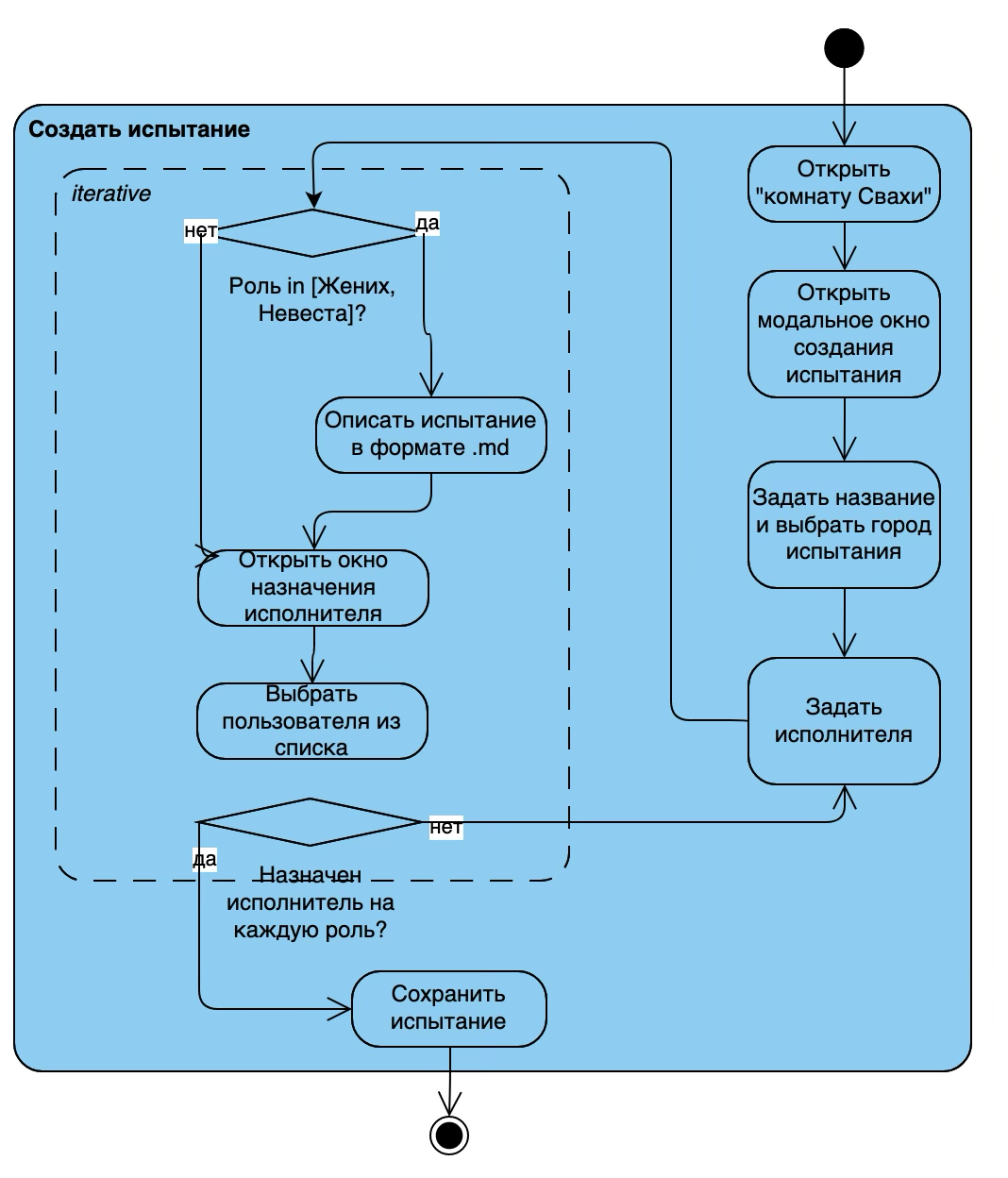
*UseCase:*

****

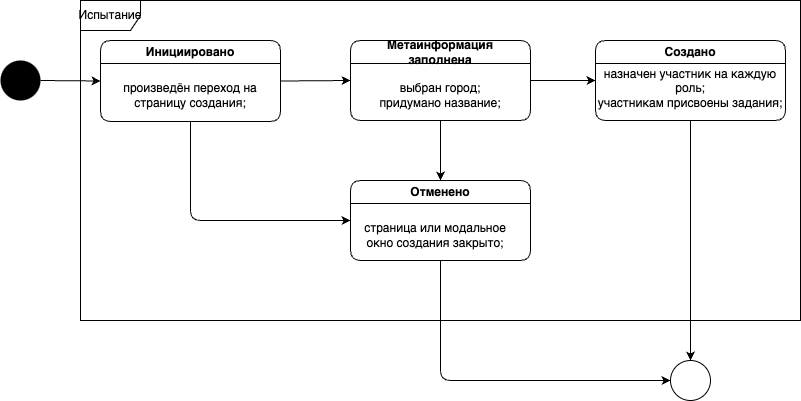
*Class:*

****

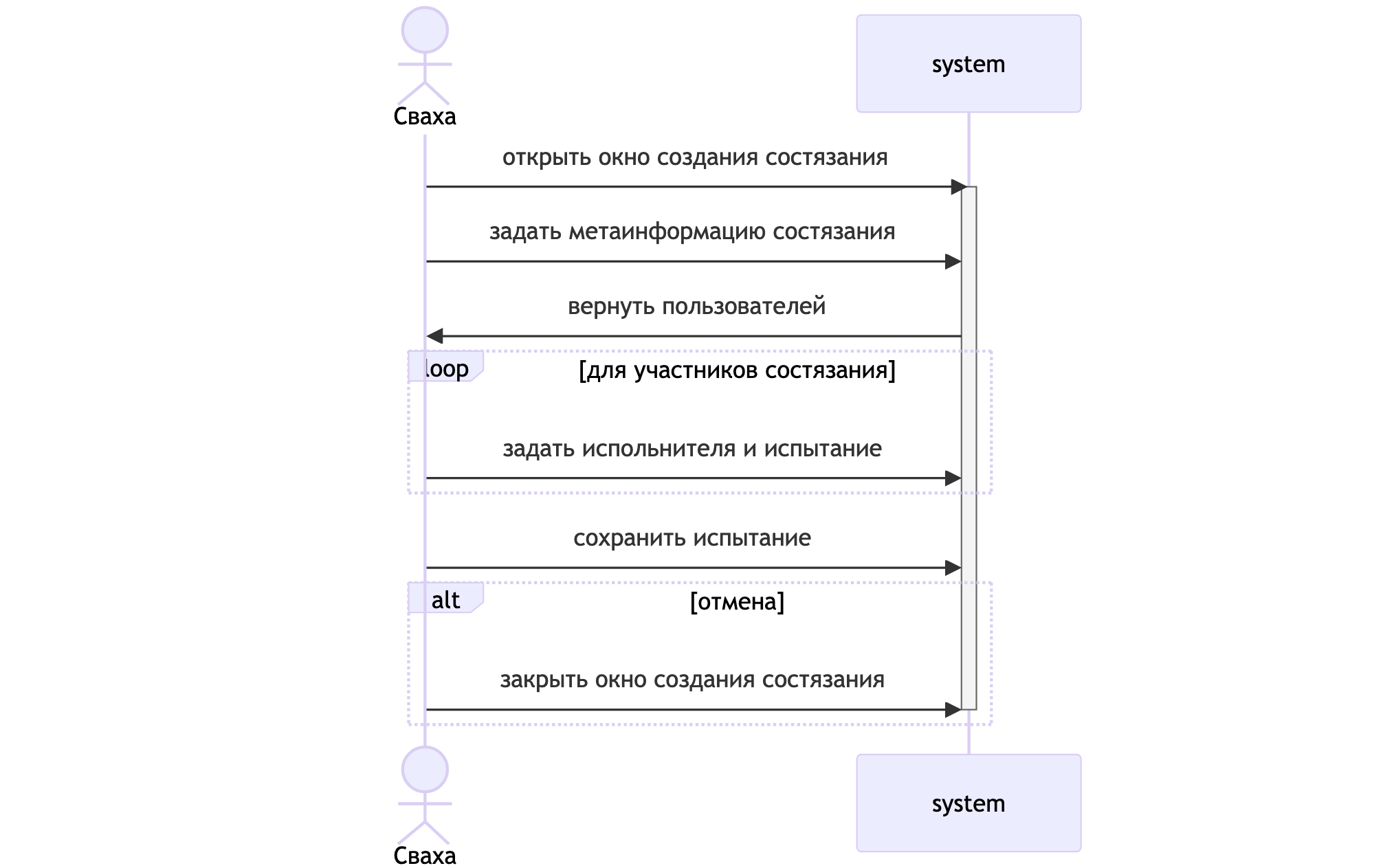
*Activity:*

**

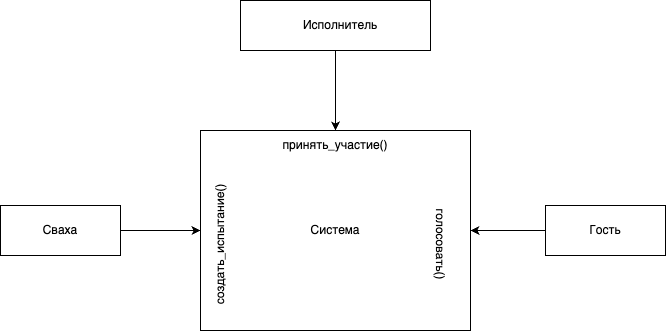
*State Machine:*

**

*Sequence:*

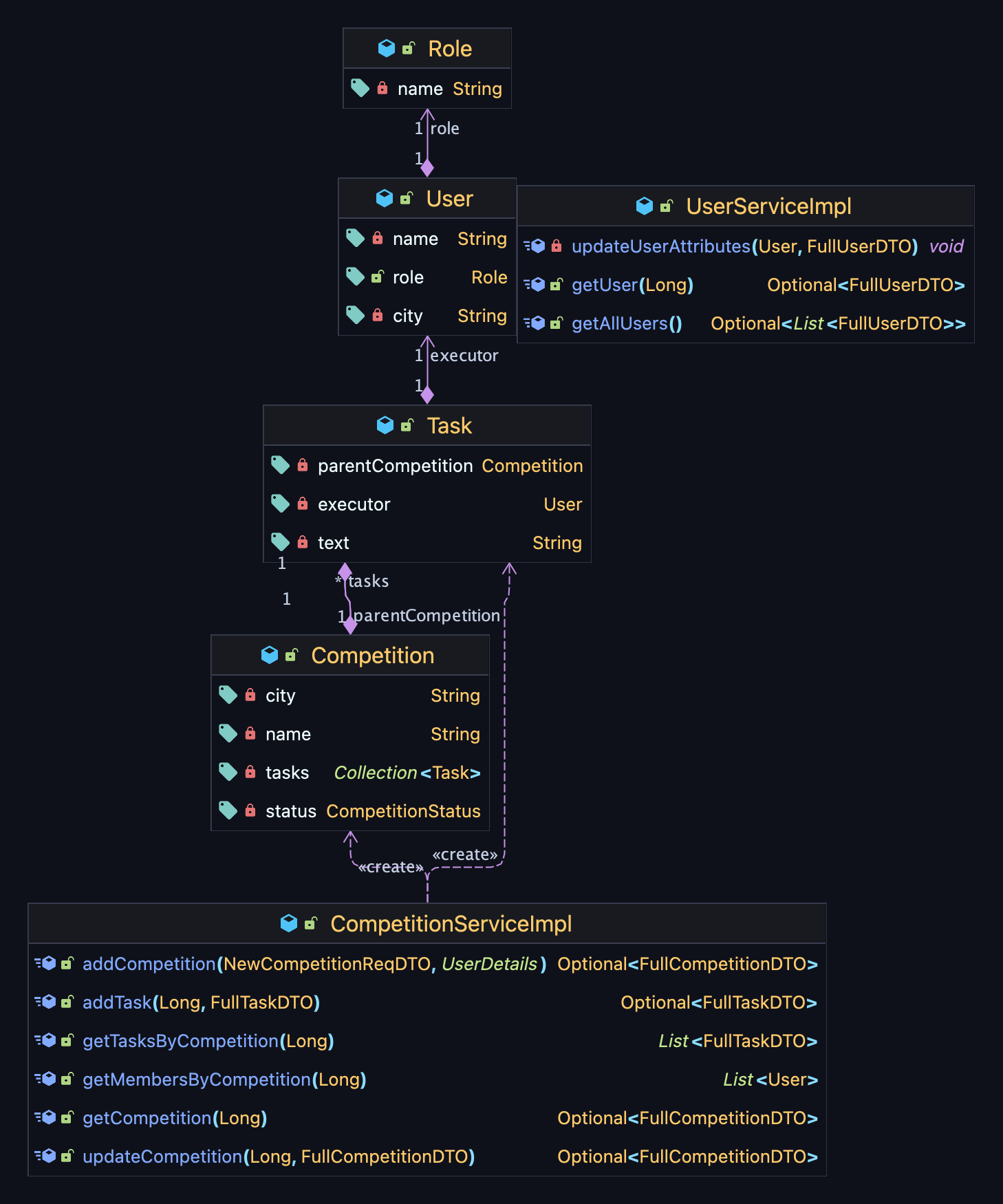
**

*Cooperative:*

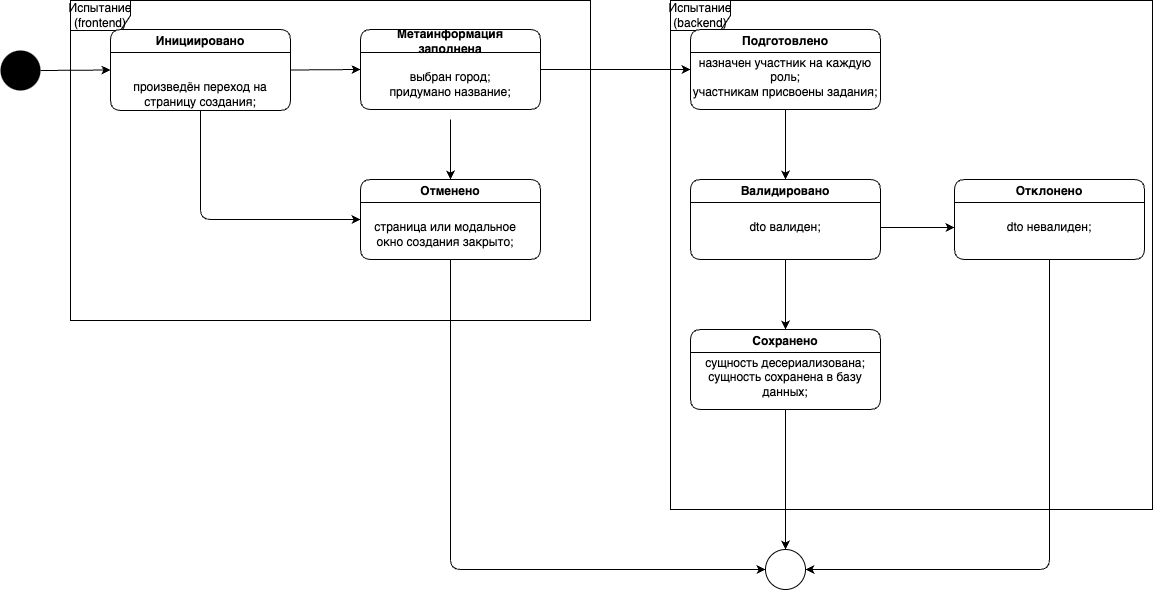
**

**5.** **Logical View**

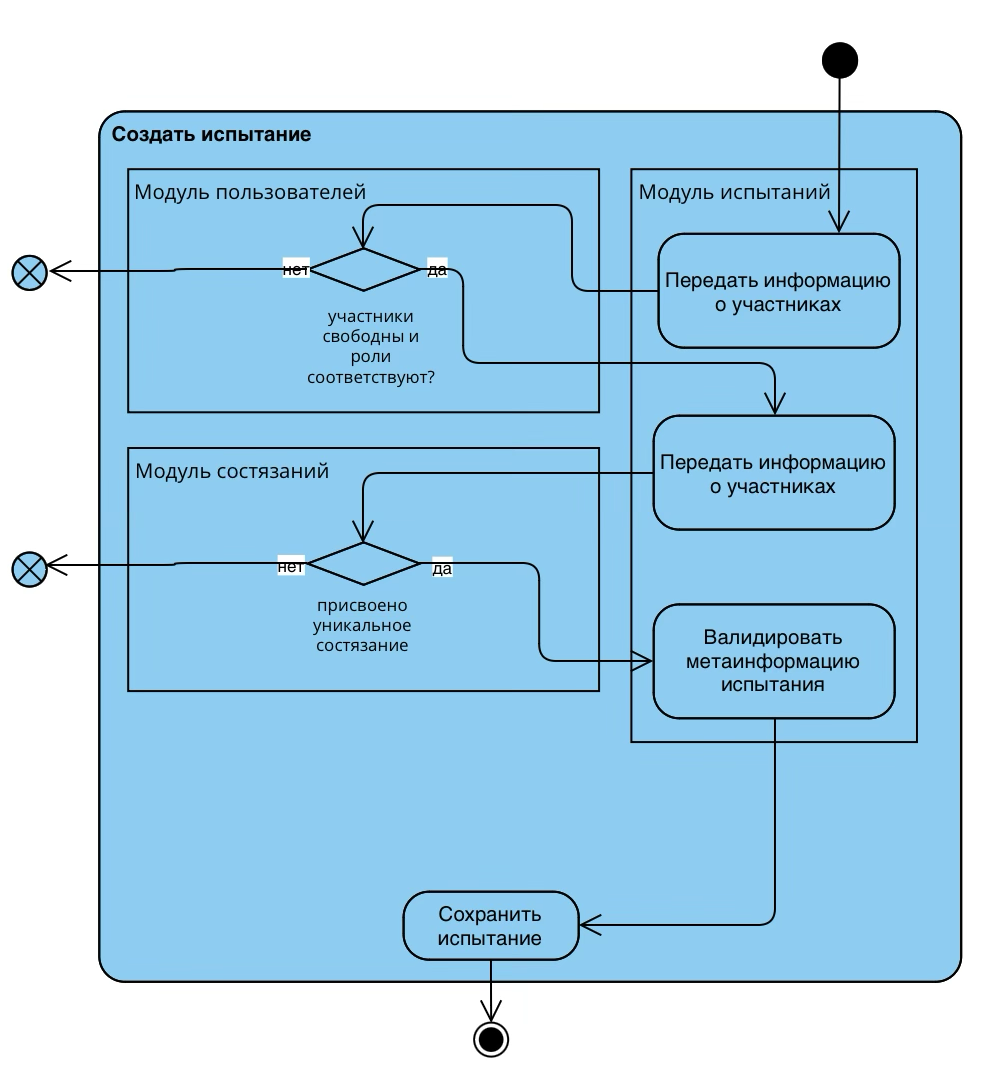
*Class:*

**

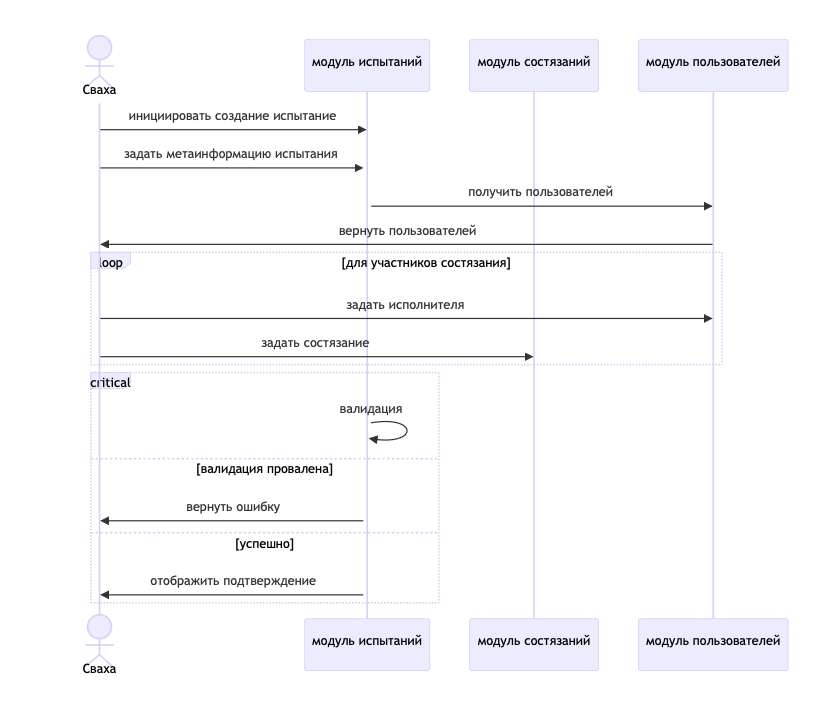
*State Machine:*

**

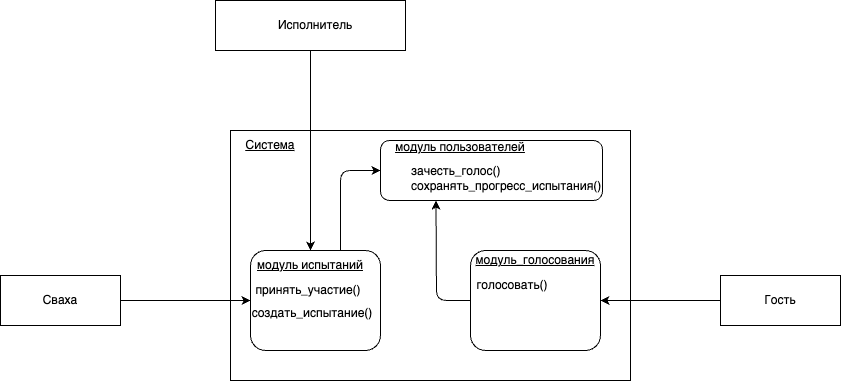
*Activity:*

**

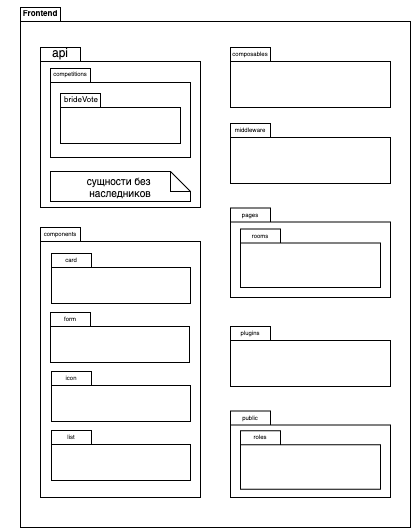
*Sequence:*

**

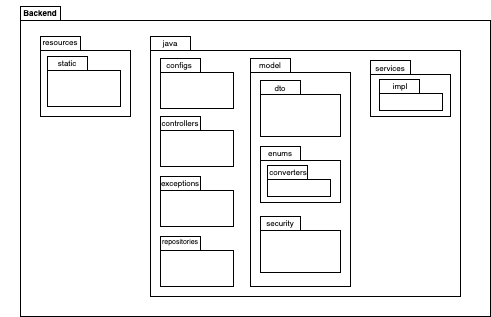
*Cooperative:*

**

*Package (frontend):*

**

*Package (backend):*

**

**6.** **Process View**

В наших usecase нет привязки к определенным моментам времени. Usecase-ы одним пользователей являются “callback-ом” usecase других, но задачи, являющиеся темпорально-зависимыми отсутствуют.

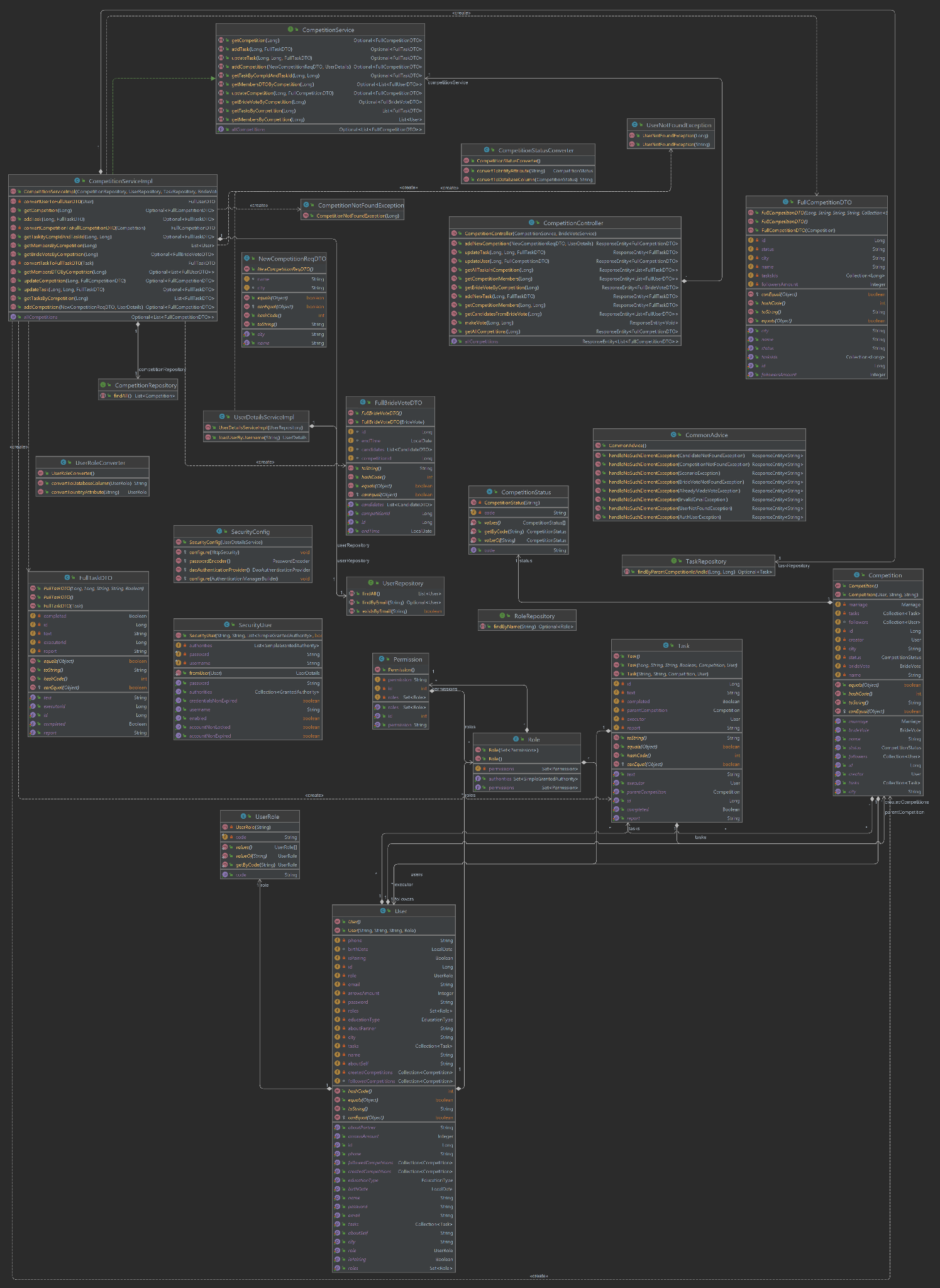
**7.** **Deployment View**

*[Данный раздел содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]*

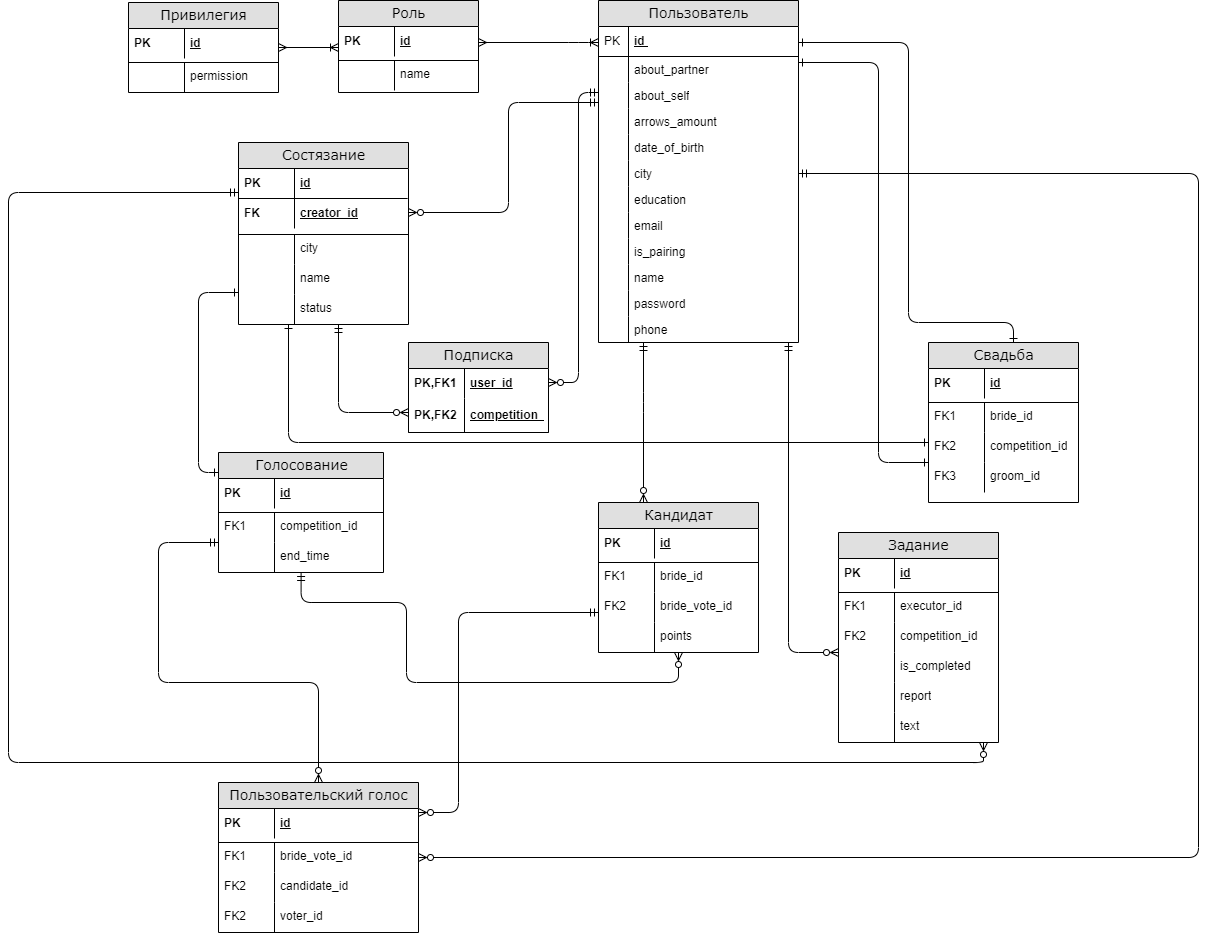
**8.** **Implementation View**

*[Данный раздел содержит описание системы в уже реализованном виде. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]*

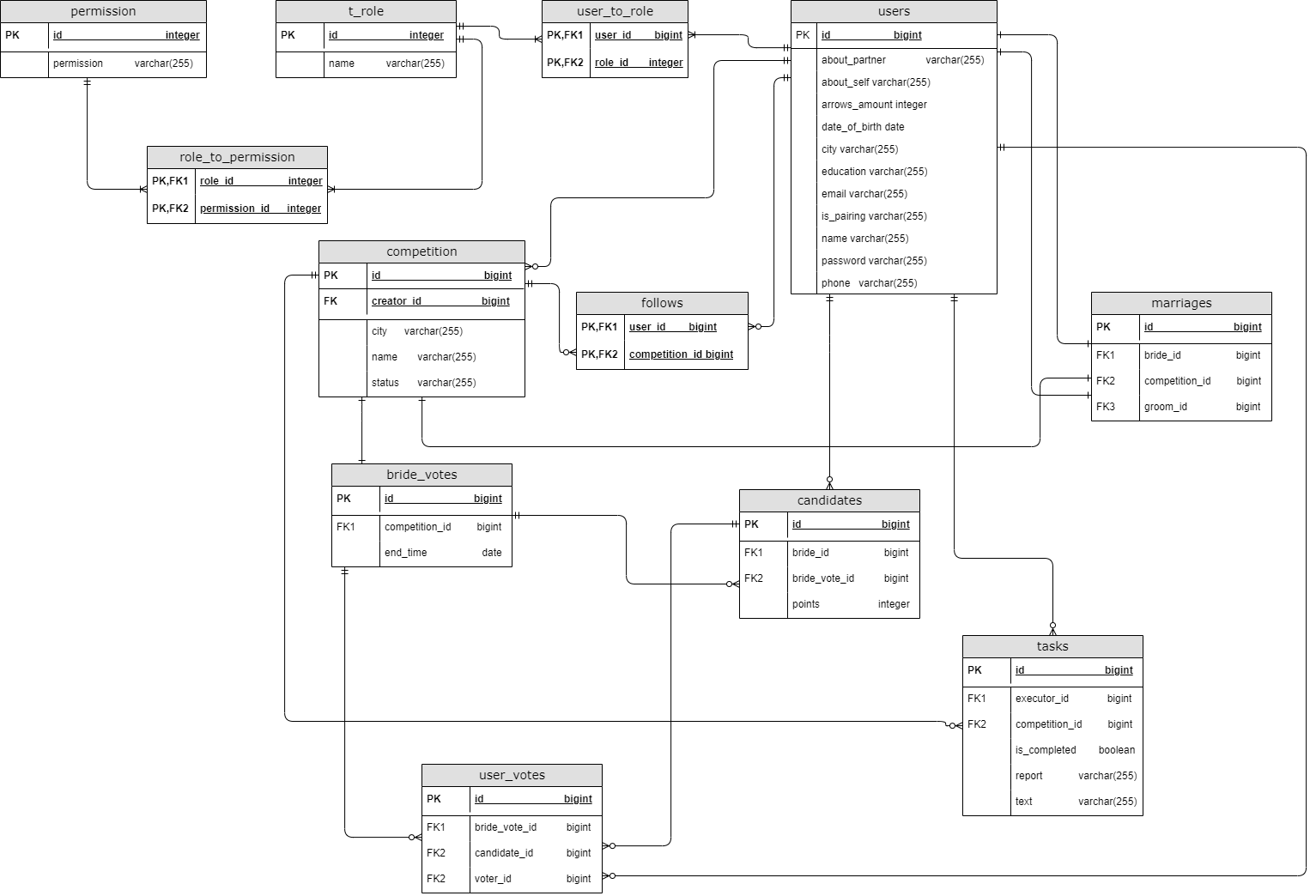
***Class Diagram***

******

**Инфологическая модель базы данных (ER)**

****

**Даталогическая модель базы данных**



**9.** **Size and Performance (Производительность)**

*Данный раздел описывает основные характеристики измерения производительности системы и их границы, которые могут оказать влияние на архитектуру,]*

**10.** **Quality (Качество)**

*Данный раздел описывает, каким образом архитектура системы удовлетворяет её показателям качества - масштабируемости, надежности, мобильности, безопасности и т.д.]*