

Software Architecture Document

(Описание Архитектуры)

1. Introduction (Введение)

[Введение представляет собой обзор на весь документ в целом и включает в себя следующие разделы - назначение, область применения, определения и аббревиатуры, ссылки и обзор.]

1.1 Purpose

[Данный документ описывает архитектуру приложения как набор точек зрения на неё - use case view, logical view, process view, deployment view, implementation view. Взаимодействие элементов разных точек зрения представлено в виде UML-диаграмм.]

1.2 Scope (Область применения)

[Приведите краткое описание области применения данного документа, к какому(им) проекту(ам) он относится, кем будет использоваться и т.д.]

Этот документ описывает систему “Услуги волшебника”, позволяющую пользователям оставлять заявку на обслуживание волшебником, оплату и просмотр статуса выполнения данной заявки. Документ предназначен для команды разработки системы “Услуги волшебника”

1.3 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)

[Укажите значение терминов и аббревиатур, которые употребляются в данном документе. Возможно указание ссылки на Глоссарий проекта.]

См. глоссарий

1.4 References (Ссылки)

[Перечислите список названия документов, на которые ссылаетесь в данном, укажите их источники.]

Амулет Самарканда, Д. Страуд, 2003, пер. А. Хромова, изд. Эксмо

1.5 Overview (Обзор документа)

[Приведите краткое описание остальных разделов документа.]

Документ содержит:

В пункте 1 — введение.

В пункте 2 — описание представления архитектуры.

В пункте 3 — описание целей и ограничения архитектуры.

В пункте 4 — описание основных сценариев использования системы.

В пункте 5 — описание логического представления.

В пункте 6 — описание системы как совокупность взаимодействующих процессов.

В пункте 7 — описание конфигурации файлов.

В пункте 8 — описание системы в уже реализованном виде.

В пункте 9 — описание производительности.

В пункте 10 — описание качества

2. Architectural Representation (Представление архитектуры)

[Данный раздел описывает в общем архитектуру системы. Укажите, какие типы диаграмм необходимы для описания разных точек зрения. Рекомендуется воспользоваться следующей таблицей:]

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	<p>+</p> <p>(Абстрактное описание)</p> <p>Log in</p>	<p>+</p> <p>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов)</p> <p>Add new wish</p>	<p>+</p> <p>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов).</p>		-
State Machine Diagram	<p>+</p> <p>(Абстрактное описание)</p>	<p>+</p> <p>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов)</p>	<p>+</p> <p>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов).</p>		-
Sequence Diagram	<p>+</p> <p>(Абстрактное описание)</p>	<p>+</p> <p>(Более подробное описание, уровни взаимодействия должны совпадать с диаграммой пакетов)</p>	<p>+</p> <p>(Полное описание прецедента с указанием вызываемых методов, используемых классов и объектов).</p>		-
Cooperative Diagram	<p>+</p> <p>(Абстрактное описание)</p>	<p>+</p> <p>(Более подробное описание, уровни взаимодействия должны совпадать с</p>	<p>+</p> <p>(Полное описание прецедента с указанием вызываемых методов, используем</p>		-

		диаграммой пакетов)	ых 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 diagramm				+	

**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)

[Перечислите здесь все архитектурно-значимые факторы - важные прецеденты, специфичные требования к работе системы и т.д.]

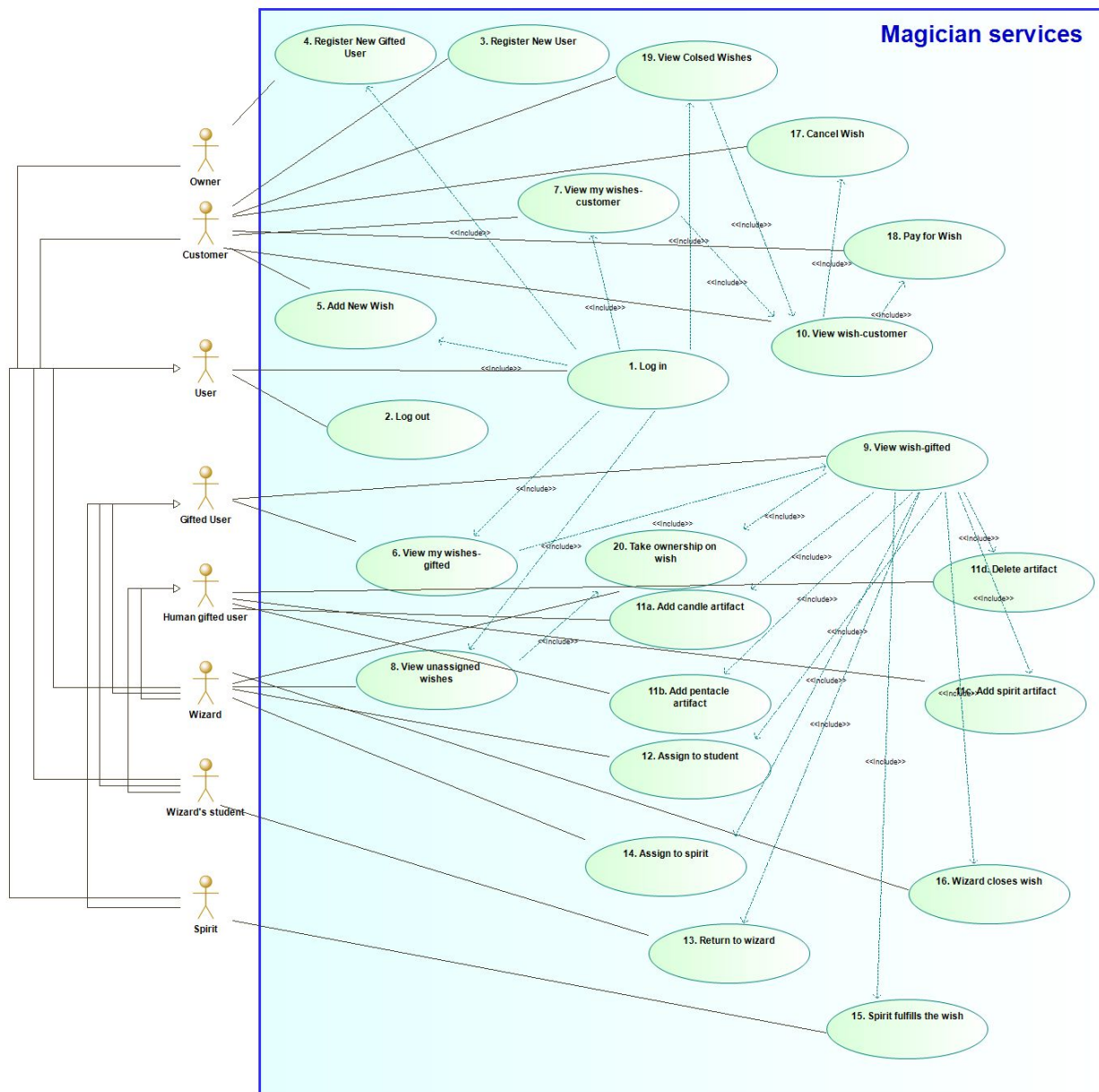
Важные прецеденты:

- Вход пользователя
- Создание желания
- Оплата желания
- Назначение желания
- Выполнение желания

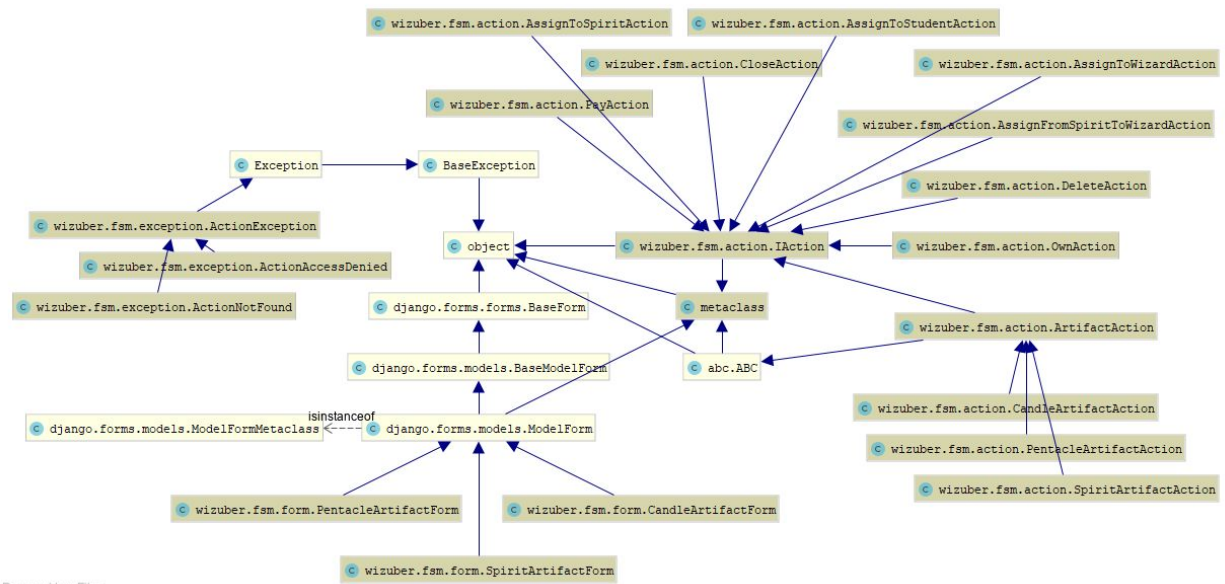
Система поддерживает работу с БД желаний и позволяет ее пользователям отслеживать состояние желания, оплачивать (для заказчика) и назначать желания другим пользователям (для пользователей, владеющих магией).

4. Use-Case View

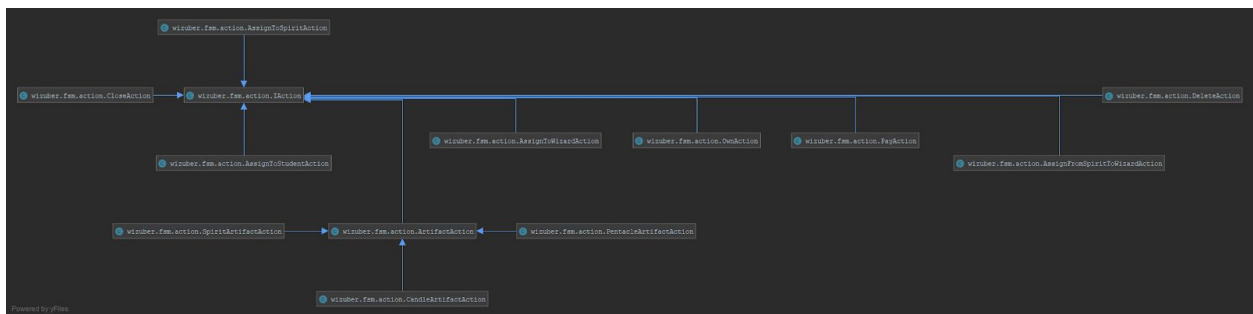
[Данный раздел содержит описание основных сценариев использования системы разными типами пользователей. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]



4.1 Use case diagram

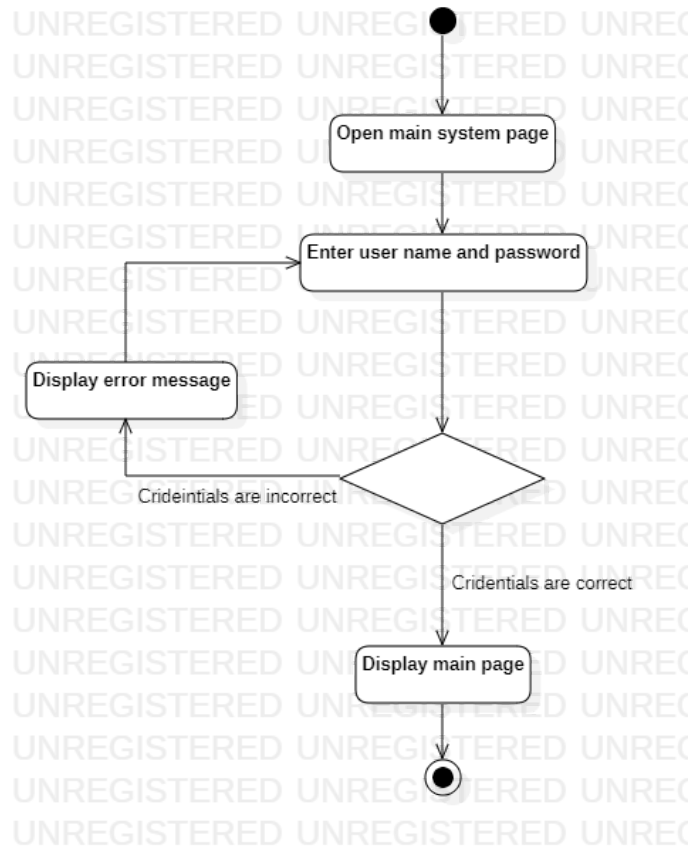


Powered by yFiles

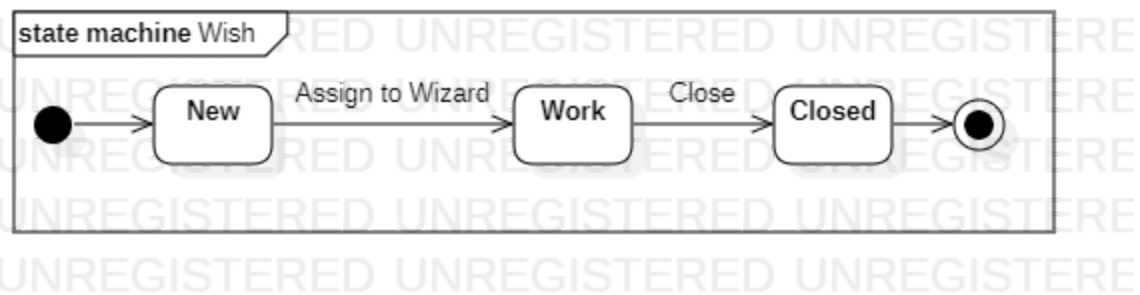


Powered by yFiles

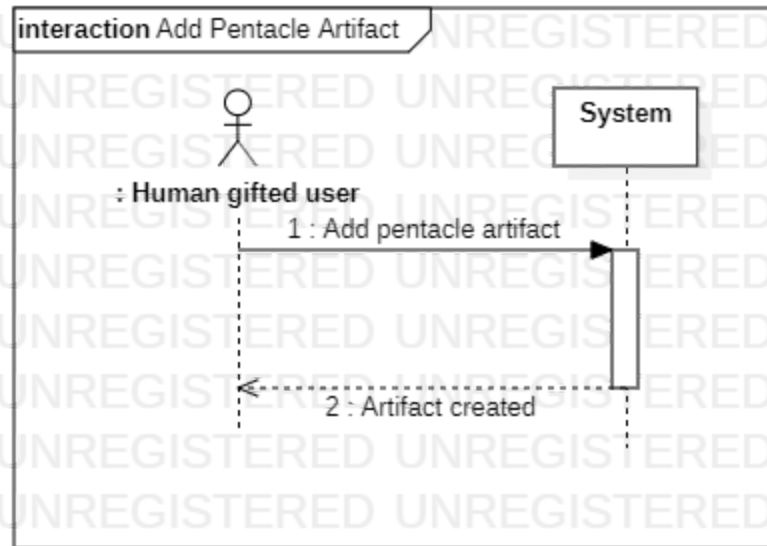
4.2 Class diagram



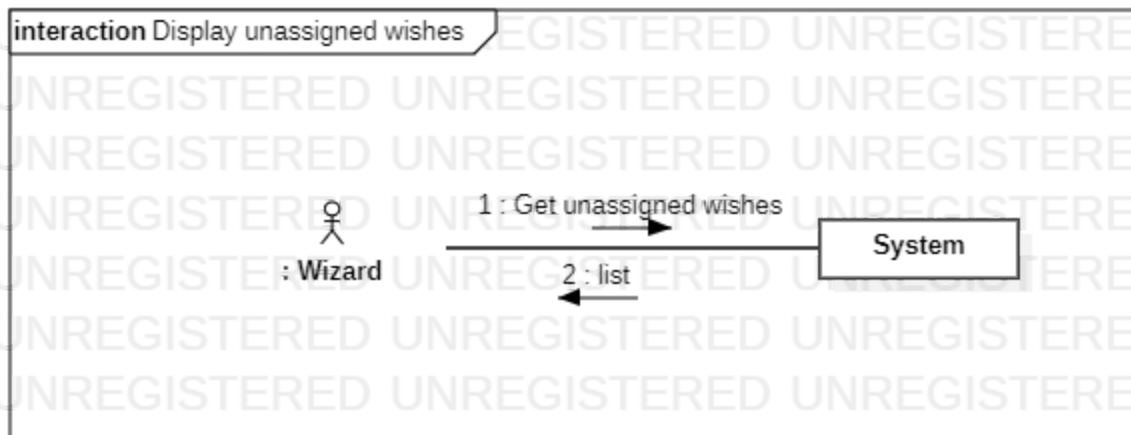
4.3 Activity diagram - Log in



4.4. State machine diagram - wish processing



4.5 Sequence Diagram - Add Pentacle Artifact

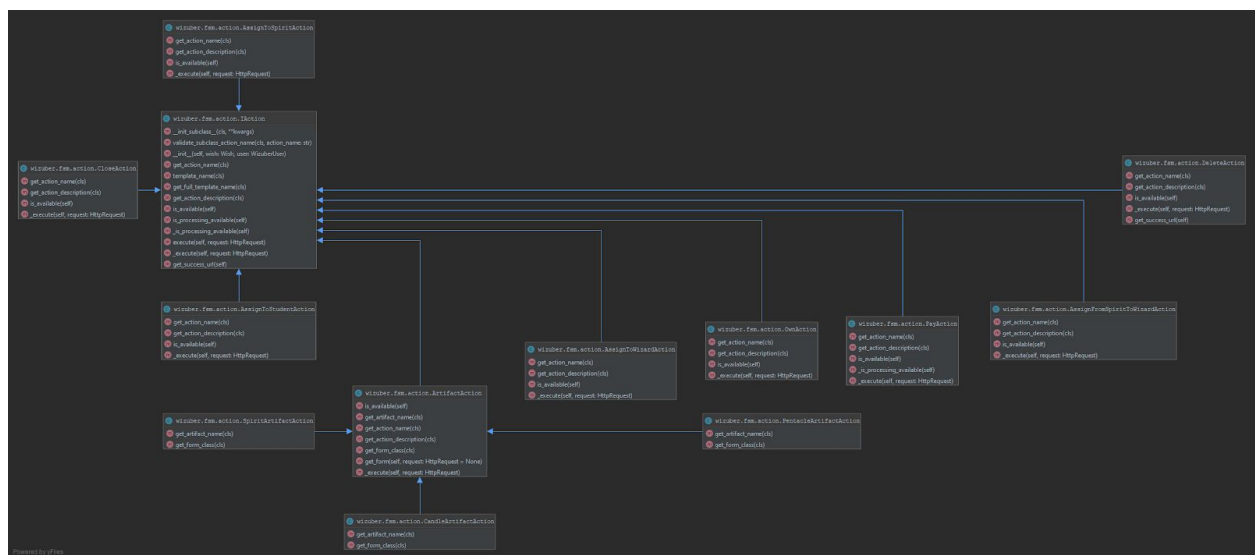
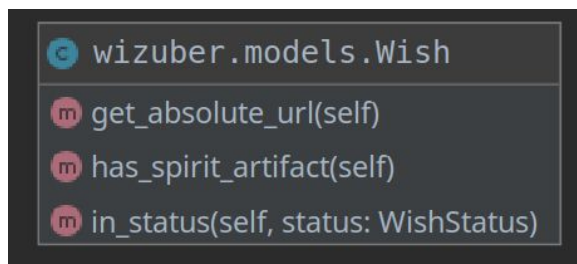


4.6. Communication diagram - get unassigned wishes

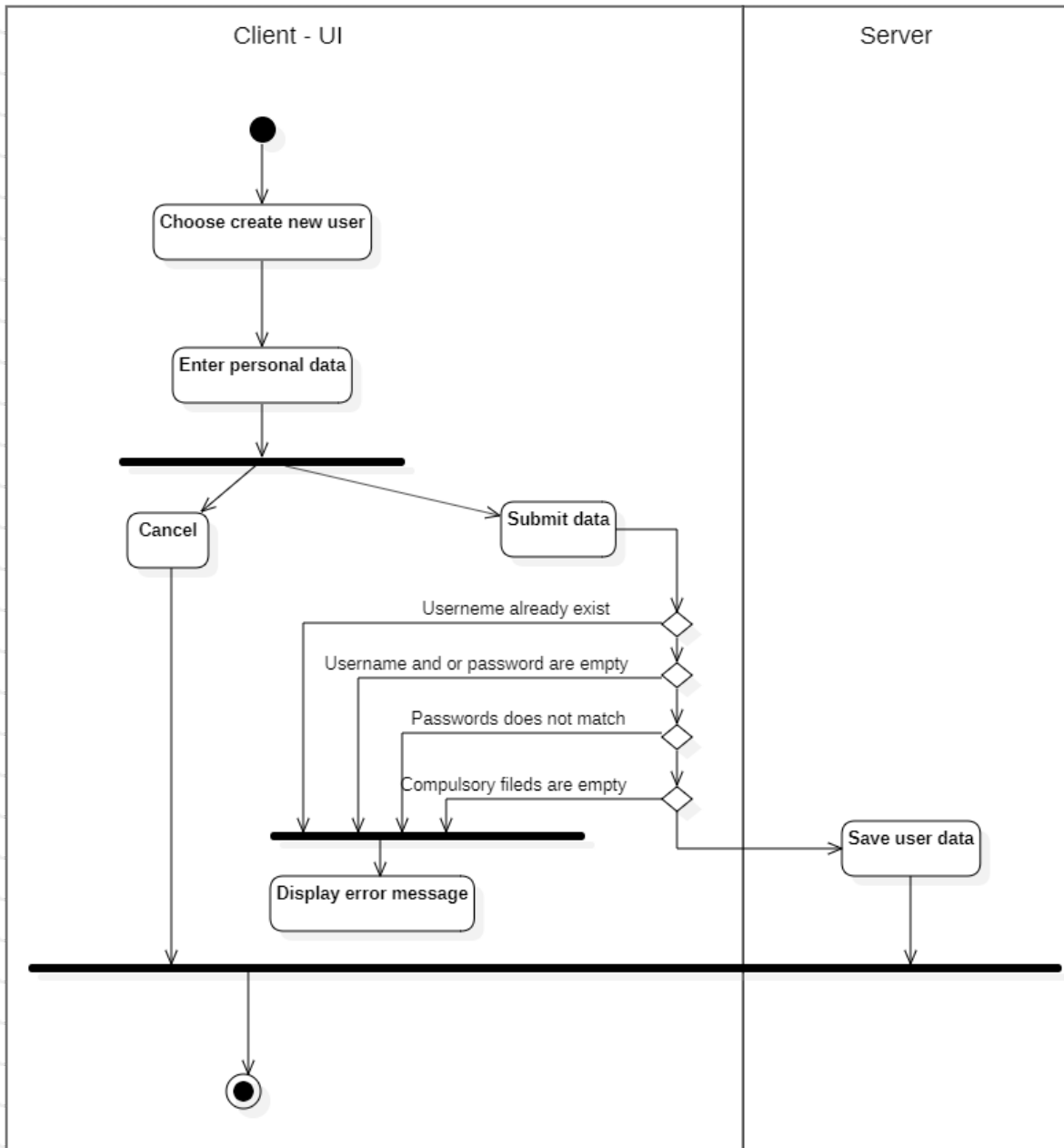
5. Logical View

[Данный раздел содержит описание слоев, на которые делится приложение, а также интерфейсов их взаимодействия. Приведите описание каждого из слоев, как они связаны между собой, их назначение. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

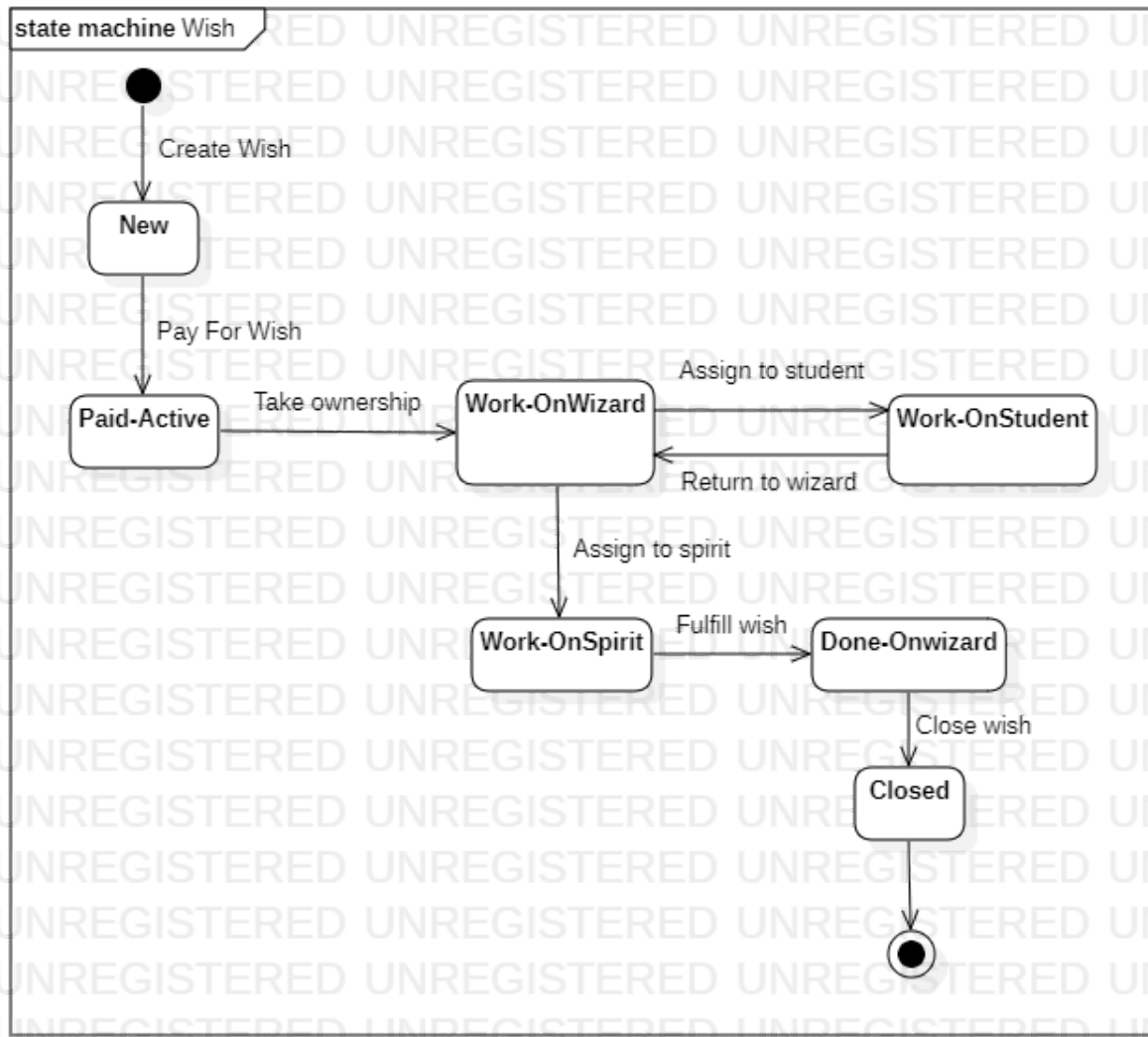
```
classDiagram
    class WizuberUser {
        +get_queryset_for_wish_list(self)
        +can_view_wish(self, wish: 'Wish')
    }
    class Wizard {
        +has_student(self)
        +get_absolute_url(self)
        +get_queryset_for_wish_list(self)
        +can_view_wish(self, wish: 'Wish')
    }
    class Student {
        +get_queryset_for_wish_list(self)
        +can_view_wish(self, wish: 'Wish')
    }
    class Spirit {
        +get_queryset_for_wish_list(self)
        +can_view_wish(self, wish: 'Wish')
    }
    class Customer {
        +get_queryset_for_wish_list(self)
        +can_view_wish(self, wish: 'Wish')
    }
    WizuberUser <|-- Wizard
    WizuberUser <|-- Student
    WizuberUser <|-- Spirit
    WizuberUser <|-- Customer
```



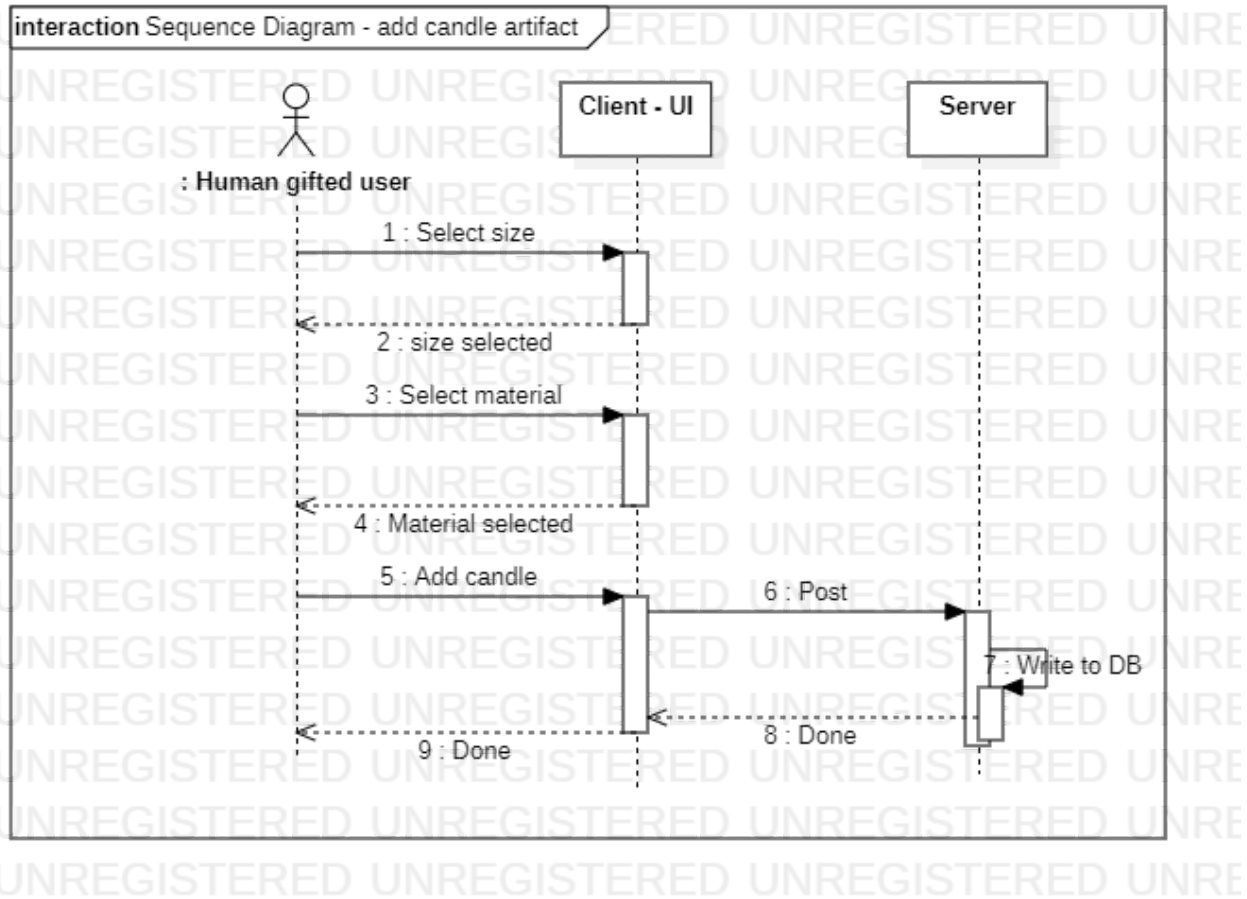
5.1. Class diagram



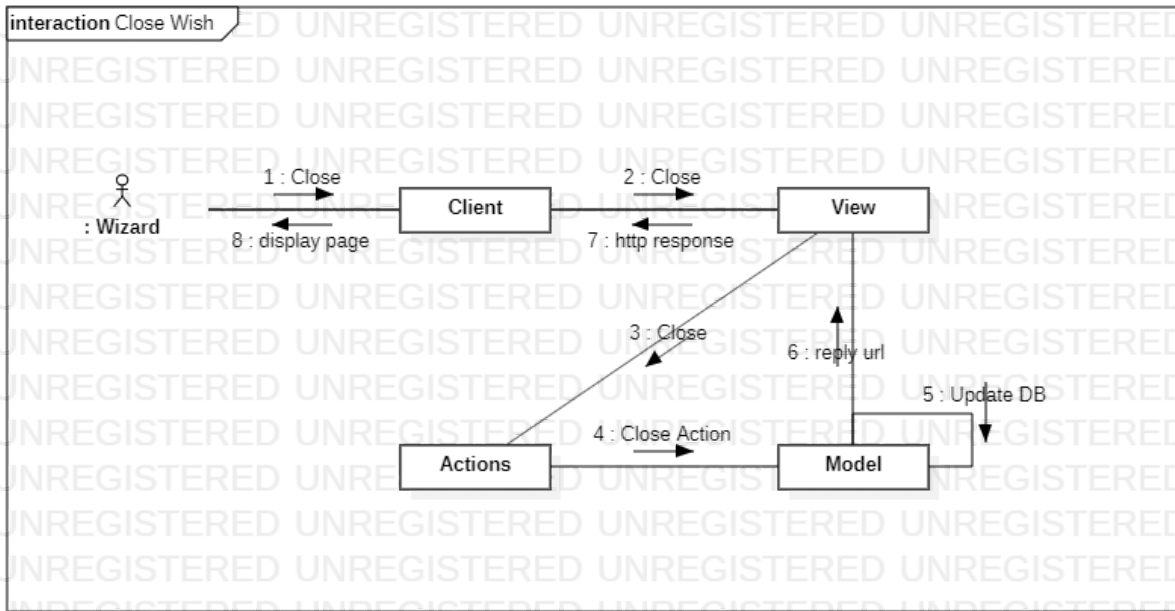
5.2. Activity diagram - register new user



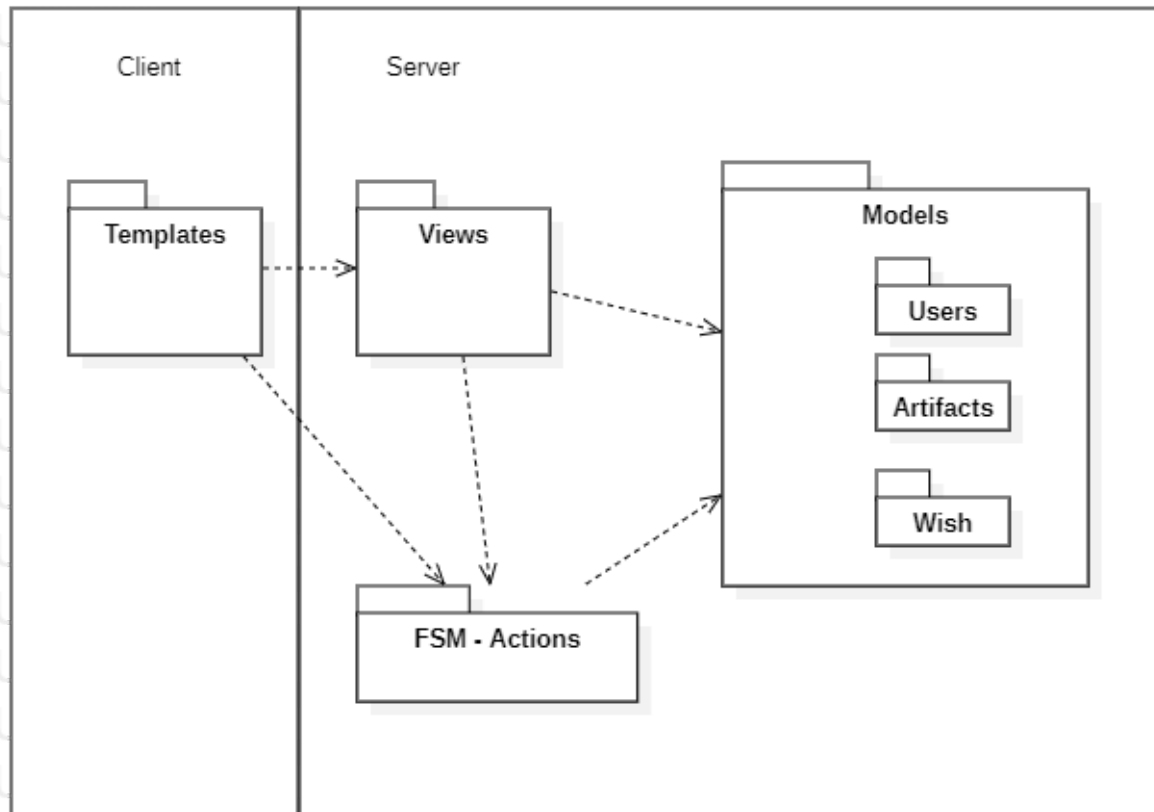
5.3. State machine diagram - Wish



5.4. Sequence Diagram - add candle



5.5. Communication diagram - close wish



5.6. Package diagram

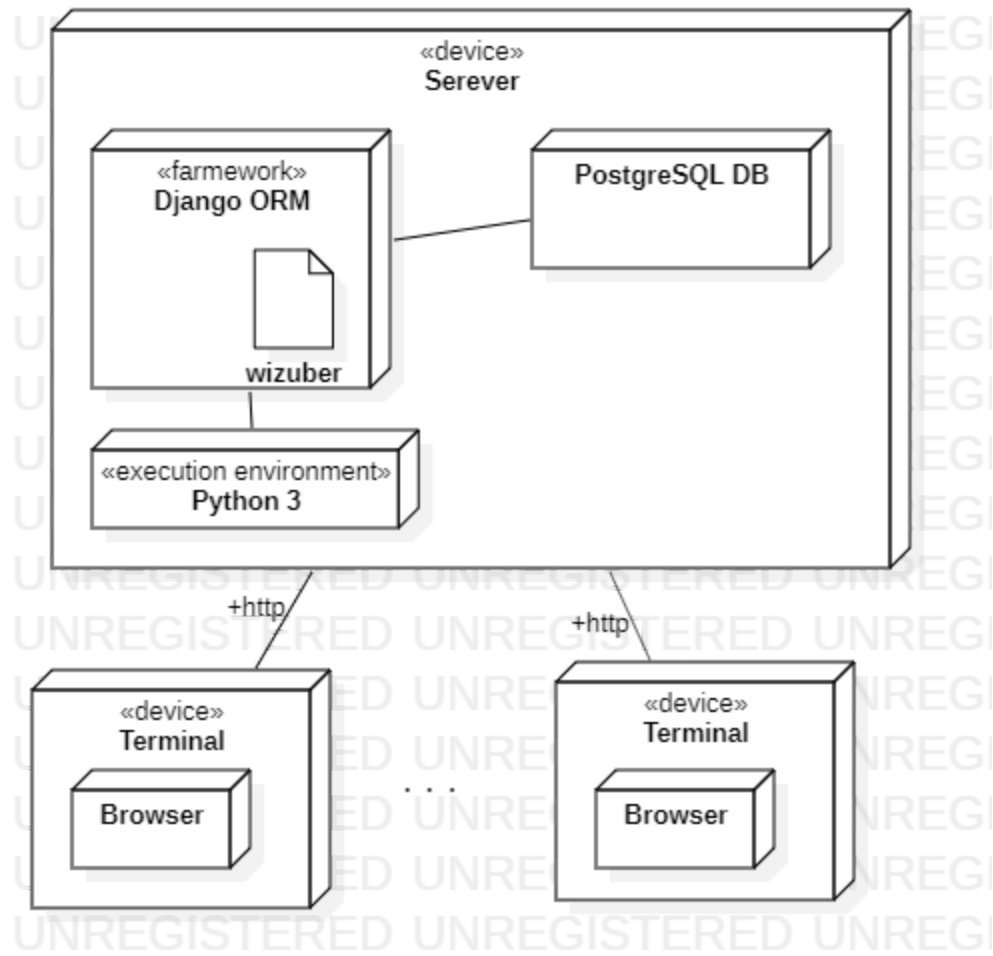
6. Process View

[Данный раздел описывает систему как совокупность взаимодействующих процессов, привязанных к определенным моментам времени. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

Не участвует в данной системе. Время ответа сервера клиенту должно соответствовать требованиям (см. SRS).

7. Deployment View

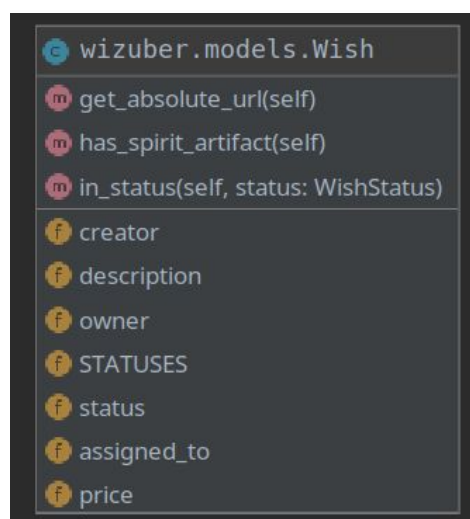
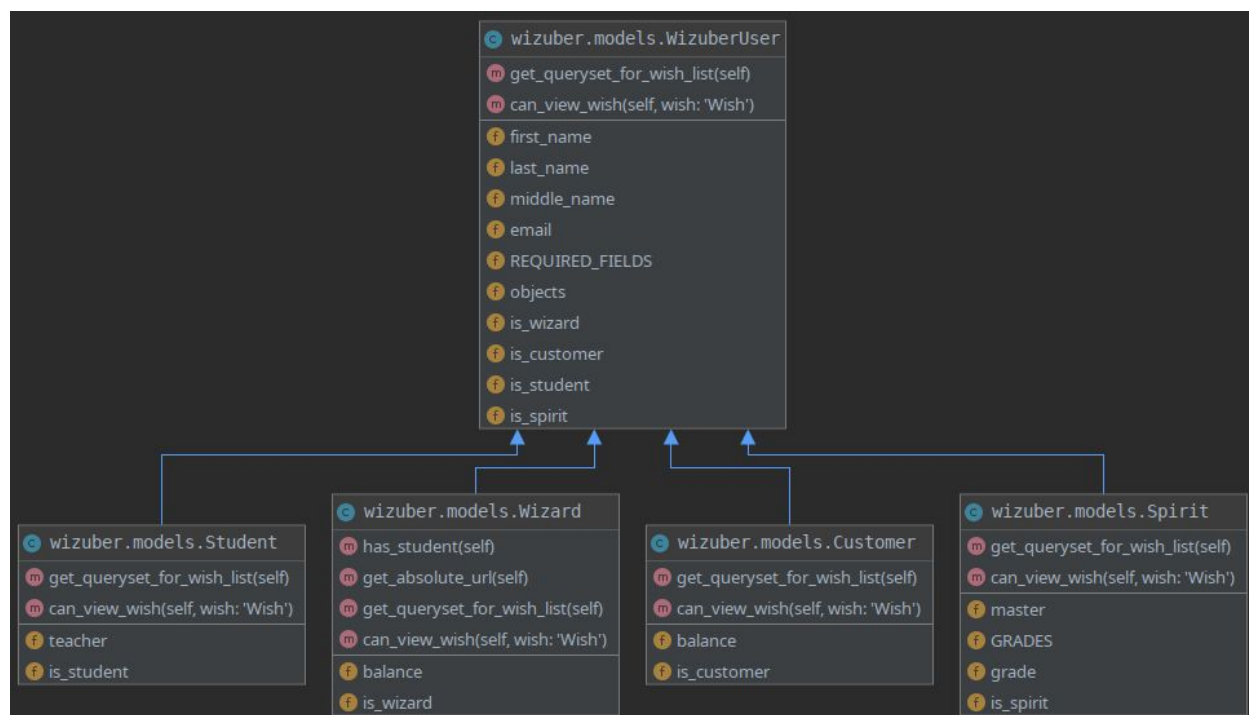
[Данный раздел содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом. Включите сюда необходимые диаграммы, указанные в п.2, приведите краткое описание каждой диаграммы.]

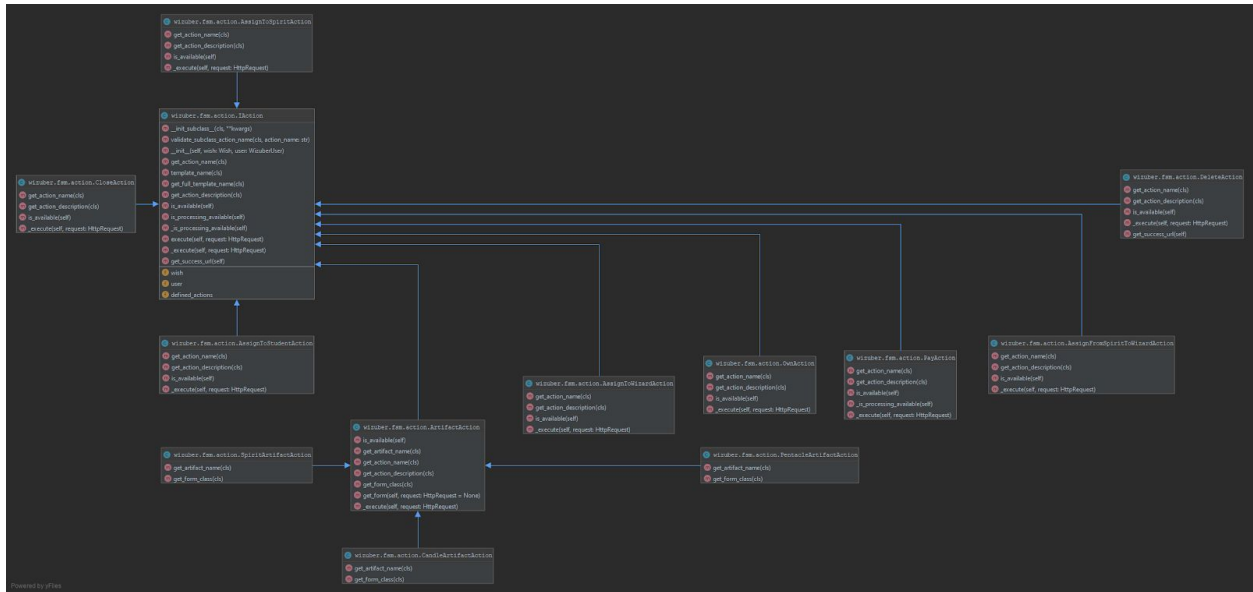


7.1. Deployment diagram

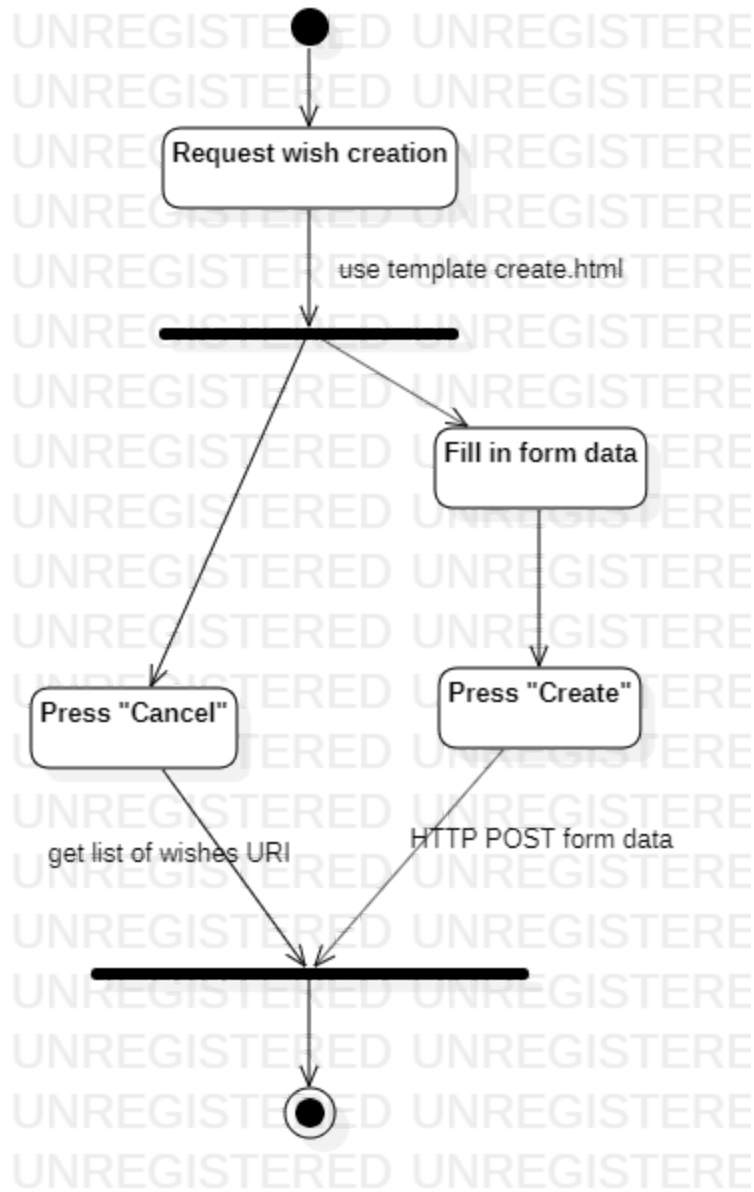
8. Implementation View

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

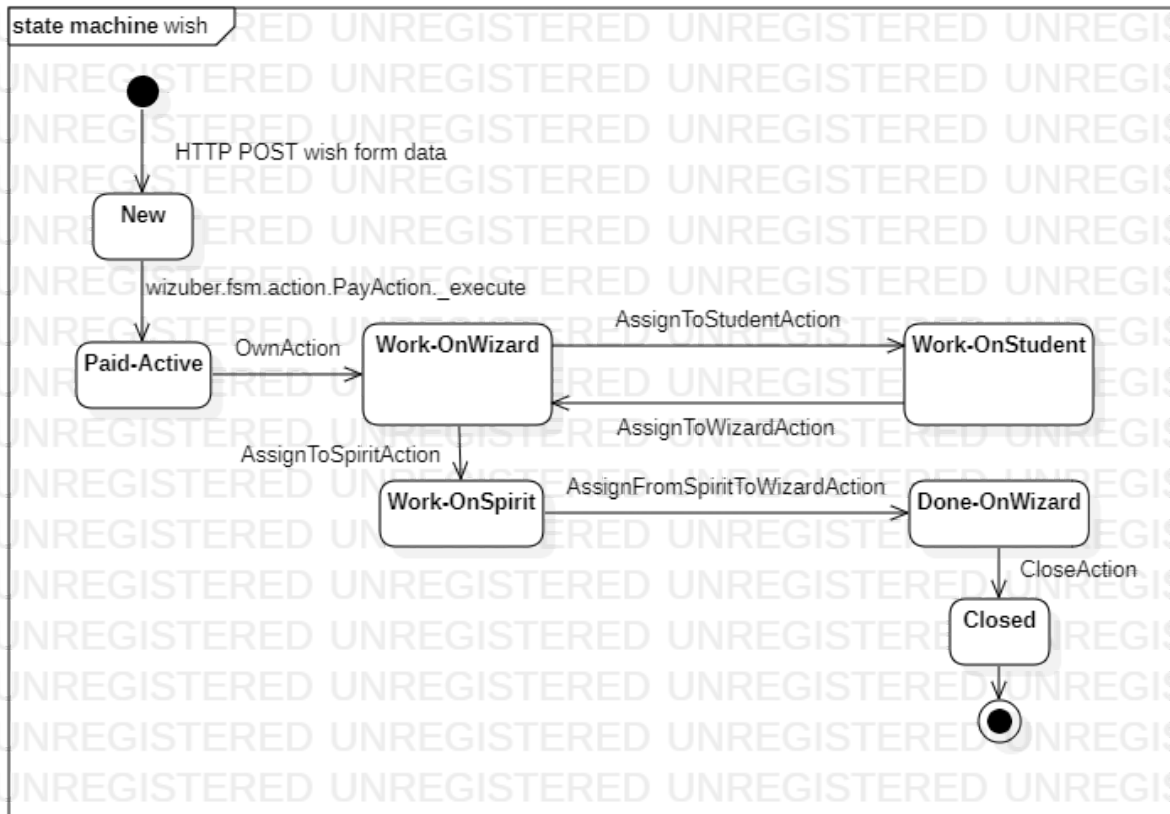




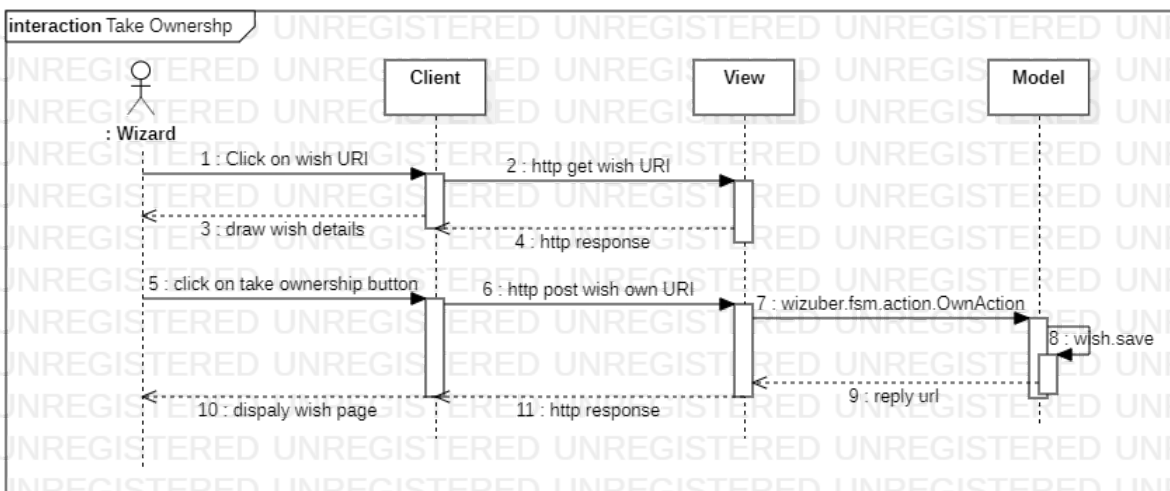
8.1 Class diagram



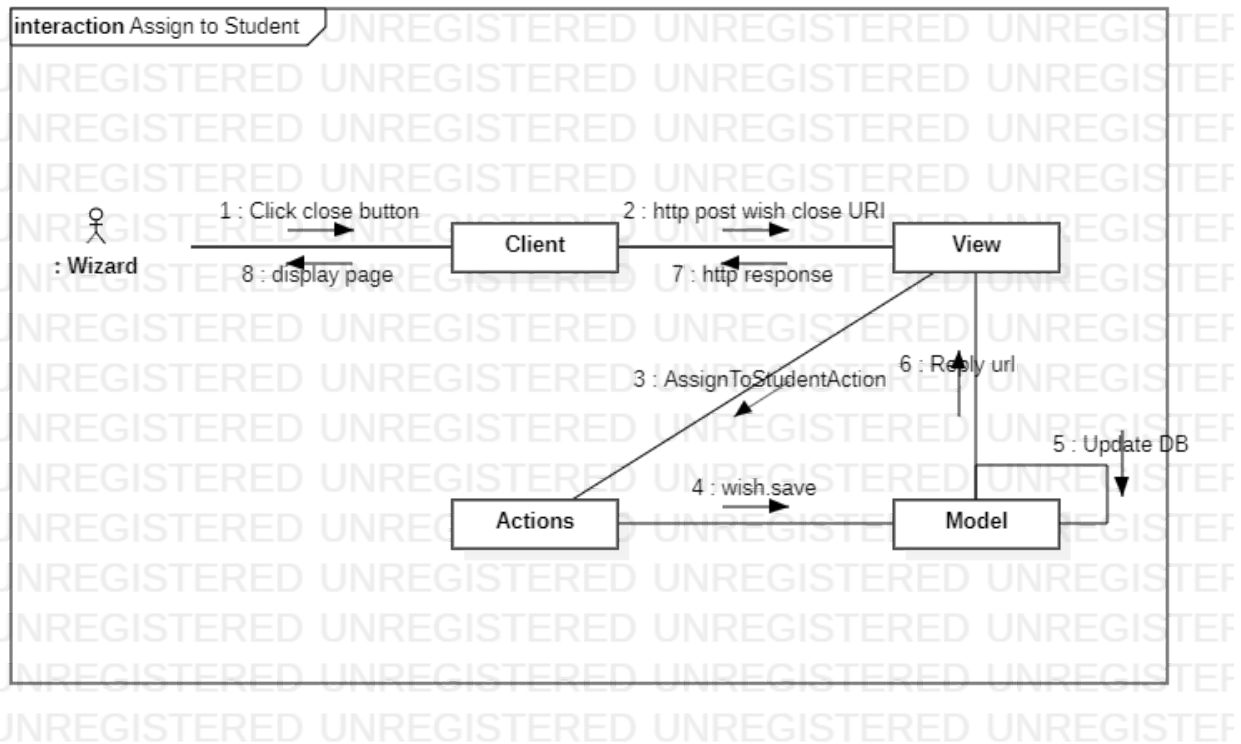
8.2. Activity diagram - Add new wish



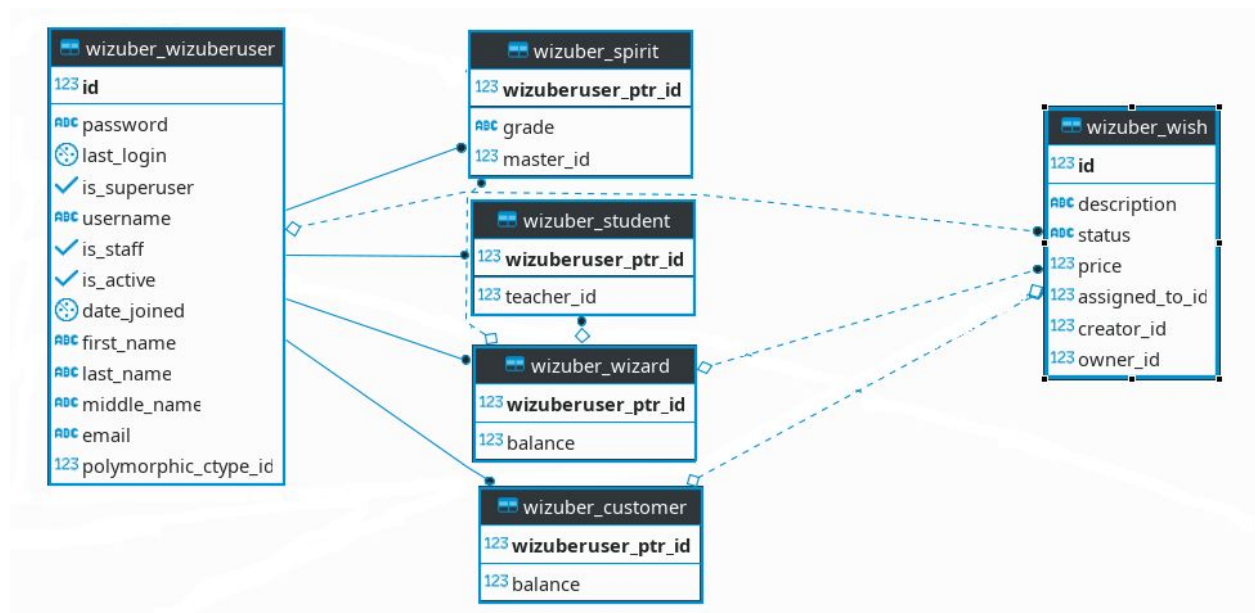
8.3. State machine diagram - wish



8.4. Sequence diagram - take wish ownership



8.5. Communication diagram - assign to student



8.6. Database diagram

9. Size and Performance (Производительность)

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

Все характеристики по производительности описаны в System Requirement Specification

10. Quality (Качество)

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

Требования к качеству указаны в SRS