< **Межконтинентальный конвейер средств индивидуального перевоплощения «Иллюзия»**>

Software Architecture Document

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <20/03/14> | <0.1> | Создание документа | Кобцев С.В. |
| <27/03/14> | <0.2> | Добавление таблицы соотношения типов диаграмм и слоев | Грудина А.М. |
| <04/06/14> | <0.3> | Добавление структуры Use Case View и описаний для добавленных блоков и блоков «Введение» | Грудина А.М. |
| <05/06/14> | <0.4> | Добавление логических уровней Добавление диаграммы прецедентов и описания диаграмм Добавление диаграмм базы данных для Use Case View и Implementation View | Гапонов А.И. Грудина А.М. Кобцев С.В. |
| <06/06/14> | <0.5> | Добавление диаграммы состояний для ключевой сущности «Заказ» («Order») в Use Case View Изменение ссылок на другие источники в References Добавление информации о реализации прецедентов в Use Case Realization Добавлено описание в Logical View Изменение Use Case Diagram | Грудина А.М. |
| <11/06/14> | <0.6> | Изменена диаграмма слоев | Гапонов А.И. |
| <11/06/14> | <0.7> | Добавлена крупноблочная диаграмма слоев и общее описание Logical View | Кобцев С.В. |
| <11/06/14> | <0.8> | Изменение диаграммы состояний для статусов сущности «Заказ»  Добавлено соотношение состояний и статусов «Заказов». | Грудина А.М. |
| <13/06/14> | <0.9> | Изменена диаграмма прецедентов | Грудина А.М. |
| <13/06/14> | <1.0> | Маркировка несделанных частей  Добавление подразделов для соответствующих диаграмм в представлениях  Добавлена диаграмма состояний для запроса на изменение состояния заказа  Добавление диаграммы деятельности для прецедента «Создание заказа» | Грудина А.М. |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 6

1.4 References 6

1.5 Overview 6

2. Architectural Representation 6

3. Architectural Goals and Constraints 7

4. Use-Case View 7

4.1 Use Case Diagrams 7

4.2 Activity Diagram for creating order by townsman 9

4.3 Sequence Diagrams 10

4.4 Cooperation and Interaction Diagrams 10

4.5 State Machine diagram for key entity “Order” 10

4.6 Preliminary Class Diagram 13

4.7 Data Base Diagram 13

4.8 Use-Case Realizations 13

5. Logical View 14

5.1 Overview 14

5.2 Architecturally Significant Design Packages 14

5.3 Detailed Class Diagram for key-value entities 15

5.4 Activity Diagram 15

5.5 State Machine Diagram for work states (role work state) of each order state 15

5.6 Sequence Diagram 17

5.7 Cooperation Diagram 17

5.8 Package Diagram 17

6. Process View 18

6.1 Timeline Diagram 18

7. Deployment View 19

7.1 Deployment Diagram 19

7.2 Package Diagram 19

8. Implementation View 20

8.1 Overview 20

8.2 Layers 20

8.3 Activity Diagram 20

8.4 State Machine Diagram 20

8.5 Sequence Diagram 20

8.6 Cooperation and Interaction Diagrams 20

8.7 Class Diagram (full version) 20

8.8 Data Base Diagram 20

8.9 Deployment Diagram 23

8.10 Package Diagram 23

9. Data View (optional) 24

10. Size and Performance 24

11. Quality 24

Software Architecture Document

# Introduction

Введение описывает назначение данного документа с точки зрения цели, масштаба работы, определений, акронимов, аббревиатур, ссылок.

## Purpose

Этот документ обеспечивает всесторонний архитектурный обзор системы МКСИД «Иллюзия» для отображения различных аспектов системы. Данный документ предназначен для получения и передачи архитектурных решений, которые были приняты в системе.

Чтобы изобразить программное обеспечение максимально точно, структура этого документа основывается на модели “4+1 представление” [KRU41].



Рисунок1 - "4+1 Views" Model

## Scope

Данный документ должен отобразить архитектуру программного обеспечения системы «Иллюзия», разрабатываемой компанией «Крафт-Девелопмент».

Документ описывает те аспекты системы «Иллюзия», которые являются архитектурно значимыми; т.е. те элементы и их поведения, которые являются основной частью системы и необходимы для понимания системы в целом.

## Definitions, Acronyms, and Abbreviations

**RUP**: Rational Unified Process

**UML:** Unified Modeling Language

**SAD:** Software Architecture Document

**WWW**: World Wide Web

## References

[SRS]: Software Requirements Specification

[MedBiquitous]: Sample SAD, <http://medbiq.org/std_specs/techguidelines/softwarearchitecture.pdf>

[KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

[USECASES]: Use Case документ, описывающий поведений всей системы в целом, и набор документов, описывающий детально каждый из use cases.

## Overview

Данный документ состоит из описания 4+1 представлений, которые в совокупности описывают все значимые архитектурные аспекты разрабатываемой системы. Структуру документа смотрите вначале документа в оглавлении.

# Architectural Representation

Данный документ описывает архитектуру системы с помощью 5 представлений: Use-Case View, Logical View, Process View, Deployment View и Implementation View, которые выполнены в форме соответствующих UML-диаграмм. Соотношение представлений и диаграмм представлено в таблице ниже.

Таблица 1 - Соотношение представлений и диаграмм

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Diagram\View | Use Case View | Logical View | Implementation view | Process View | Deployment View |
| Use Case Diagram | 100% | - | - | - | - |
| Class Diagram | 30% (class analyze) | 60% (detailed class analyze) | 100% | - | - |
| Activity Diagram | 20% | 50% | 100% | - | - |
| Timeline Diagram | - | - | - | 100% | - |
| State Machine Diagram | + | + | + | - | - |
| Deployment Diagram | - | - | 40% | - | 100% |
| Sequence Diagram | + | + | + | - | - |
| Cooperation Diagram (+Interaction Diagram) | + | + | + | - | - |
| Package Diagram | - | 100% | + | - | + |
| Data Base Diagram | 100% | - | 100% (detailed) | - | - |

# Architectural Goals and Constraints

В этом разделе описываются требования к программному обеспечению и задачи, которые имеют существенное влияние на архитектуру разрабатываемой системы.

Требования, которые являются ключевыми для архитектуры системы:

1. Система должна предоставлять различные функциональные возможности для различных групп пользователей.
2. Система должна обеспечивать, чтобы все запросы, созданные пользователями, немедленно передавались следующим участникам процесса и обновлялись связанные аналитические таблицы. [SRS]
3. Система должна быть доступна для рабочего персонала для использования и обслуживания - семь дней в неделю по 8 часов каждый день, исключения составляют профилактические дни (1 раз в месяц).[SRS]
4. Система должна предоставлять валидные на текущий момент времени данные о статусе заказа без задержек. [SRS]
5. Система должна иметь модуль для печати, необходимый для обеспечения возможности пользователю печатать описание своего заказа вне зависимости от его состояния и возможности аналитикам печатать аналитические таблицы.
6. Система должна представлять из себя веб-портал, поддерживающий работу во всех популярных браузерах (Chromeи Safari).

# Use-Case View

В данном разделе будут представлены диаграммы прецедентов, диаграммы деятельности, диаграммы последовательностей, диаграммы коопераций и взаимодействий, предварительная диаграмма классов и диаграмма состояний ключевой сущности системы – «Заказ», которые предназначены для описания основных сценариев использования системы различными пользователями и базового описания сущностей системы.

## Use Case Diagrams

Этот тип диаграмм описывает возможные варианты работы с системой различными группами пользователей.

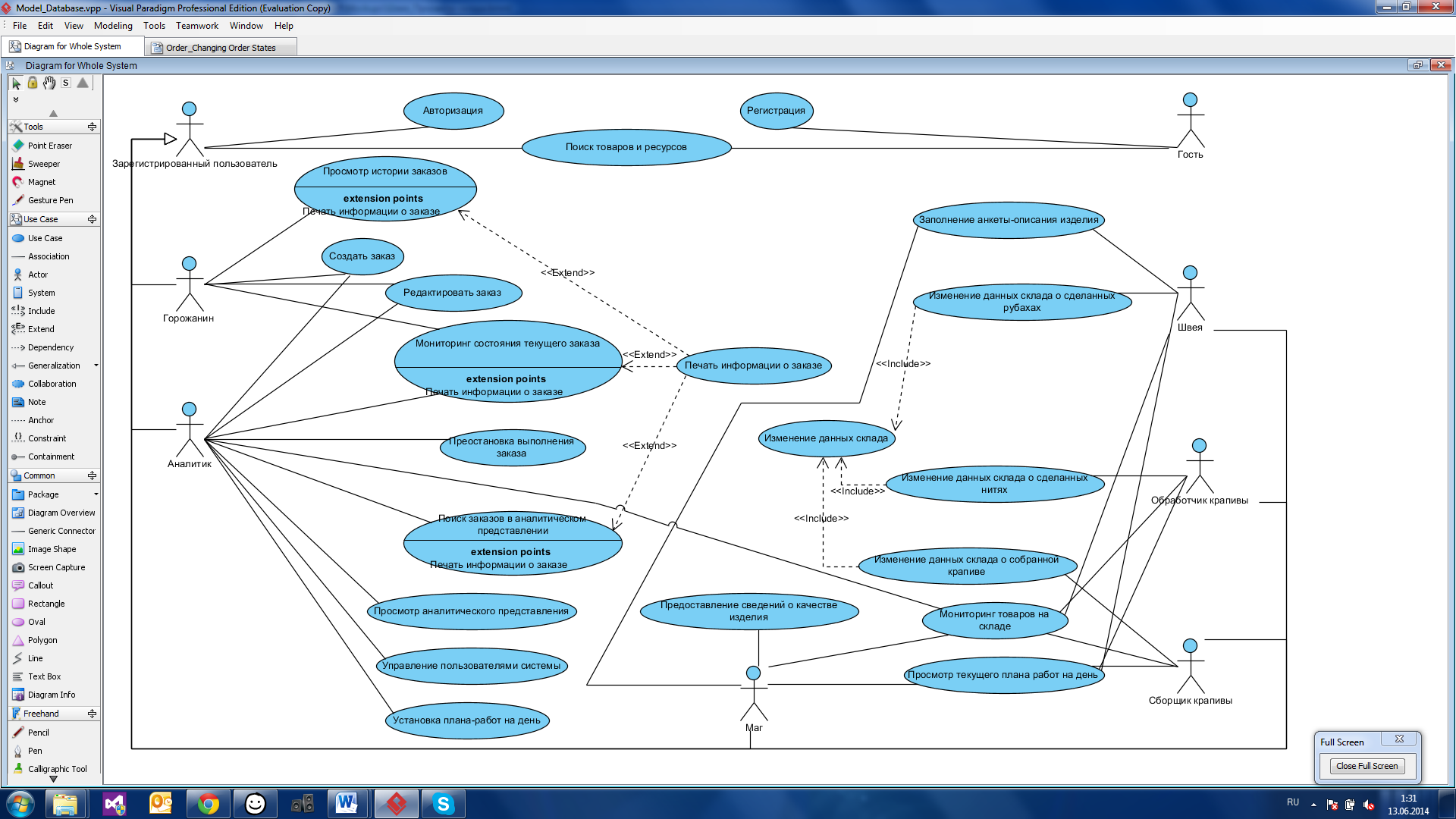


Рисунок2 - Use-Case Diagram

## Activity Diagram for creating order by townsman

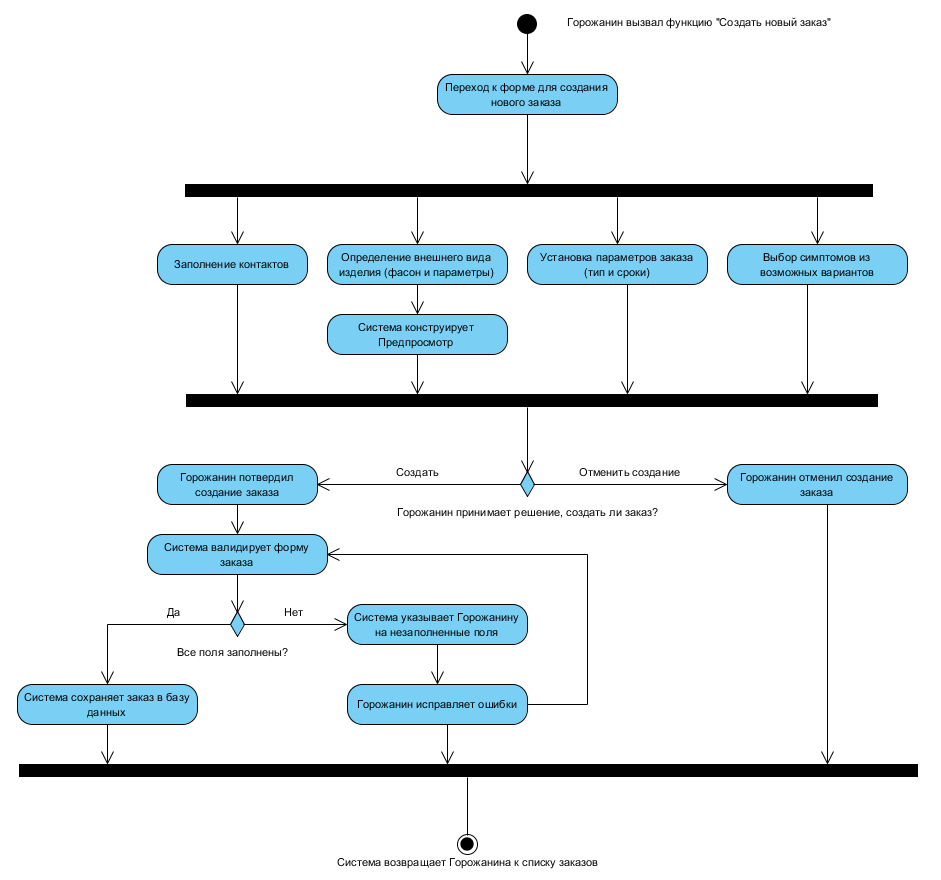


Рисунок 3 - Activity Diagram for creating order by townsman

Эта диаграмма описывает последовательность действий Горожанина и Системы для создания заказа. В описание входит также отмена создания и валидация формы.

## Sequence Diagrams

**Добавить готовую диаграмму по ее завершении!!!**

## Cooperation and Interaction Diagrams

**Добавить готовую диаграмму по ее завершении!!!**

## State Machine diagram for key entity “Order”

Ключевой сущностью системы «Иллюзия» является «Заказ» («Order»), так как вся система направлена на работу с ними, обеспечение их выполнения для конечных пользователей путем поочередной работы с этой сущностью всех участников процесса. В ходе этой работы состояния сущности «Заказ» изменяется, как показано на диаграмме ниже.

Остальные сущности в системе являются подчиненными этой сущности и, соответственно, каждое из состояний сущности «Заказ» определяется агрегацией нескольких других сущностей.

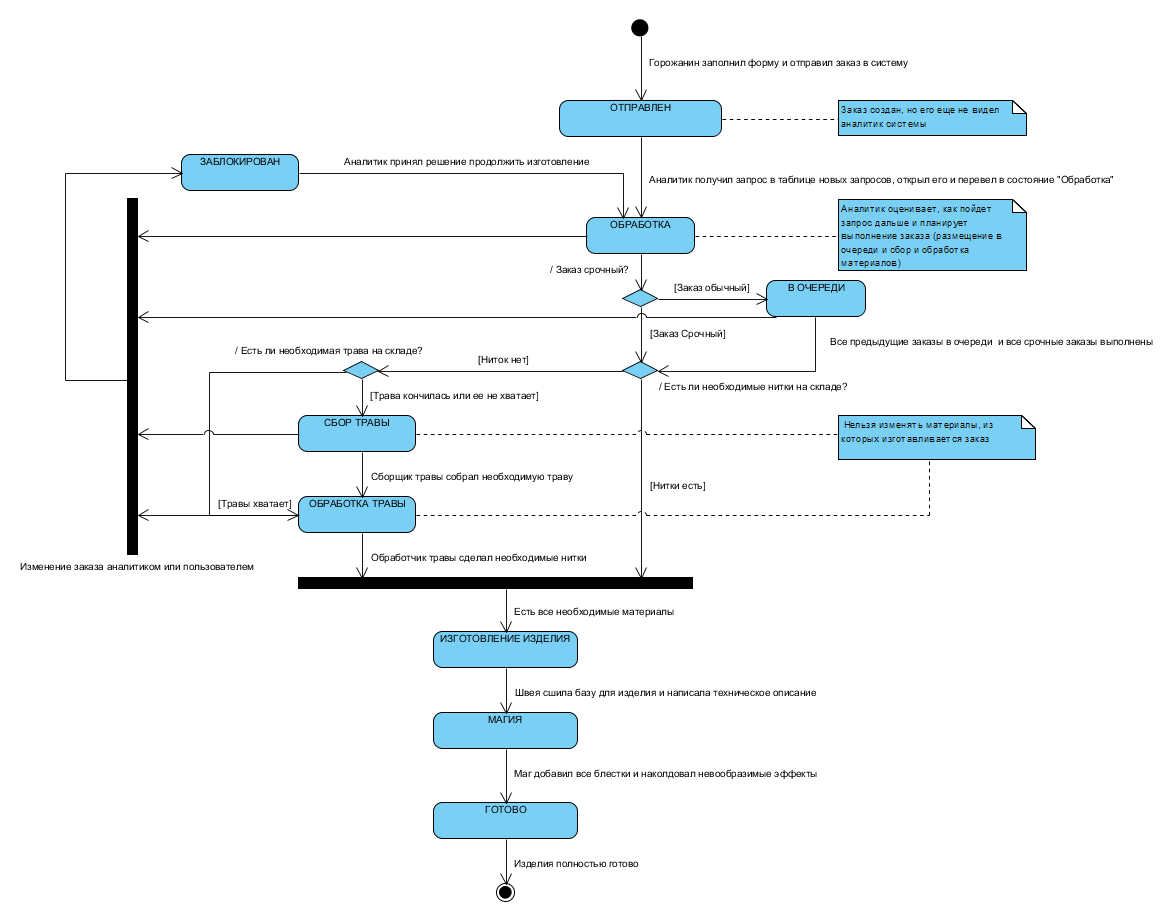


Рисунок4 - State Machine Diagram for states of the entity "Order"

Кроме состояний у заказов также есть статус, который может быть изменен пользователем или аналитиком. По умолчанию статус – «Обычный». Пользователь при создании заказа или во время стадии «Обработка» может изменить его на «Срочный». На дальнейших стадиях изменения запрещены для пользователя, а аналитик может только заблокировать «Заказ». Ниже представлена диаграмма изменений состояний статуса заказа.

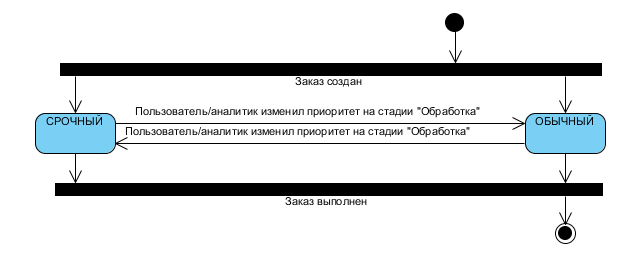


Рисунок5 - State Machine Diagram for statuses of the entity "Order"

Соответственно, статусы и состояния связаны следующим образом:

Таблица 2 - Соотношение крупноблочных состояний и статусов заказа

|  |  |  |
| --- | --- | --- |
| Состояния \ Статусы заказа | Обычный | Срочный |
| Новый | + | + |
| В процессе | + | + |
| Заблокированный | + | + |
| Сделанный | + | + |

В этой таблице состояния описаны крупными блоками:

* «Новый» - включает состояния «Отправлен», «Обработка»;
* «В процессе» - включает состояния «В очереди», «Сбор травы», «Обработка травы», «Изготовление изделия», «Магия»;
* «Заблокированный» - включает состояние «Заблокирован»;
* «Сделанный» - включает состояние «Готово».

## Preliminary Class Diagram

Данный тип диаграмм отображает предварительную структуру классов приложения и их взаимодействие друг с другом.

**Добавить готовую диаграмму по ее завершении!!!**

## Data Base Diagram

Данный тип диаграммы отображает предварительную структуру базы данных. Дальнейшее ее уточнение будет производиться во время реализации системы.

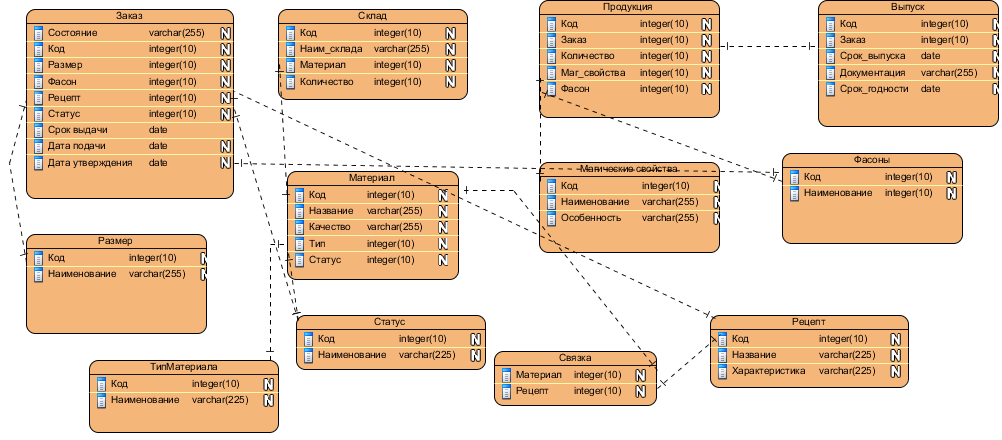


Рисунок6 - Data Base Diagram

## Use-Case Realizations

Описание реализации прецедентов описывается в [USECASES].

# Logical View

Этот раздел содержит архитектурно важные части конструкции модели такие, как ее декомпозиция в подсистемы и пакеты. И для каждого значимого пакета представлена его декомпозиция в классы и утилиты класса. Также раздел содержит подробное описание всех значимых классов системы с точки зрения их функциональных обязанностей. Показаны ключевые отношения между классами, операции класса и атрибуты.

## Overview



Рисунок 7 - Layers (base)

Модель иерархического представления приложения “Иллюзия” основывается на стратегии иерархического представления ответственности, которая связывает каждый уровень с определенной ответственностью.

Эта стратегия была выбрана, потому что она изолирует различные обязанности системы друг от друга так, чтобы были улучшены и дальнейшая системная разработка, и обслуживание.

## Architecturally Significant Design Packages

Данный тип диаграммы отображает взаимодействие логических уровней системы. В левой части диаграммы отображены названия уровней и то, как они взаимодействуют между собой. В правой части – уточнение и указание уровней и варианты реализации системы.

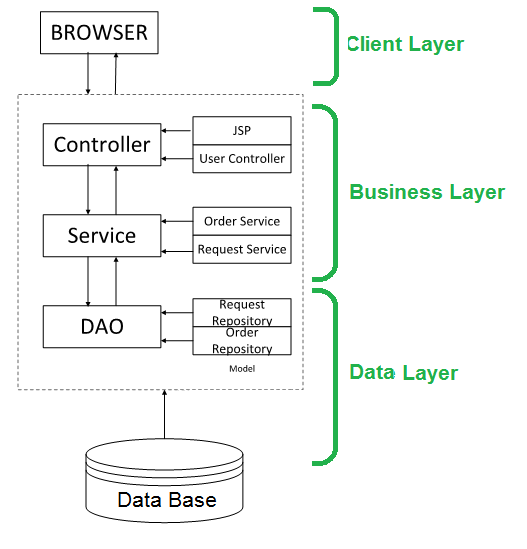


Рисунок 8 – Layers (extended)

## Detailed Class Diagram for key-value entities

**Добавить готовую диаграмму по ее завершении!!!**

## Activity Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## State Machine Diagram for work states (role work state) of each order state

Диаграмма состояний описывает запрос на изменение статуса работы над заказом для любого заказа, находящегося в группе состояний «В процессе», описанных в диаграмме состояний из Use Case View.

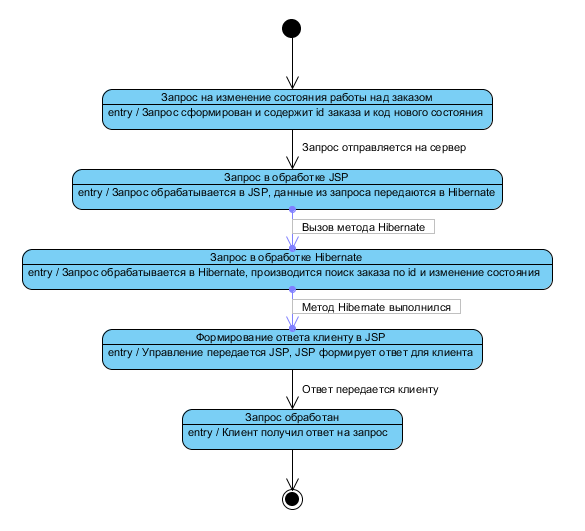


Рисунок 9 - State Machine Diagram for role work states

## Sequence Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Cooperation Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Package Diagram

**Добавить готовую диаграмму по ее завершении!!!**

# Process View

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

## Timeline Diagram

**Добавить готовую диаграмму по ее завершении!!!**

# Deployment View

[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. It is a view of the Deployment Model. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs) that execute the software and their interconnections (bus, LAN, point-to-point, and so on.) Also include a mapping of the processes of the **Process View** onto the physical nodes.]

## Deployment Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Package Diagram

**Добавить готовую диаграмму по ее завершении!!!**

# Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

## Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

Выделяется три слоя в данной системе, а именно Data layer, Business layer, Client layer. На уровне Data layer взаимодействие базы данных проходит через hibernate, что позволяет использовать средства для автоматического построения запросов и извлечения данных. Выделяется несколько сущностей для абстрагирования от таблиц в базе данных: fashion, material, order, recipe, size, status,typeofmaterials, user. Классы OrderRepository и RequestRepository позволяют производить различные операции при выборке данных из базы, так же как создание заказа, удаление заказа, получение необходимых данных с заданными параметрами. На уровне Business происходит обработка заказа, заполнение его данными, обновление уже существующего заказа, а так же его удаление. Для этого используется два класса OrderService и RequestService. UserController обеспечивает связь между пользователем и системой, контролирует ввод данных пользователем и использует модели описанные выше ля реализации определенных действий. Непосредственное взаимодействие с системой происходит через браузер, где пользователь вводит необходимые данные для создания заказа, они проходят валидность и проходят по следующим нижележащим уровням.

## Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

1. Data layer

В данный слой входит сама база данных, сущности описанные в виде VO, и взаимодействие по получению и отправке данных по заказу и/ или изменению через классы Request repository, Order repository.

1. Business layer

Данный слой описывает взаимодействие с получением, отправкой заказа, его изменением и/или обновлением. Service осуществляется через классы OrderService и RequetsService.Так же на этом уровне осуществляется обработка данных с формы с помощью класса UserController.

1. Client layer

В данном слое пользователь непосредственно может вносить данные для оформления заказа и выбирать из списка существующие характеристики изделия. После того, как заказ будет составлен, система добавляет заказ в базу, для дальнейшего выполнения.

## Activity Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## State Machine Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Sequence Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Cooperation and Interaction Diagrams

**Добавить готовую диаграмму по ее завершении!!!**

## Class Diagram (full version)

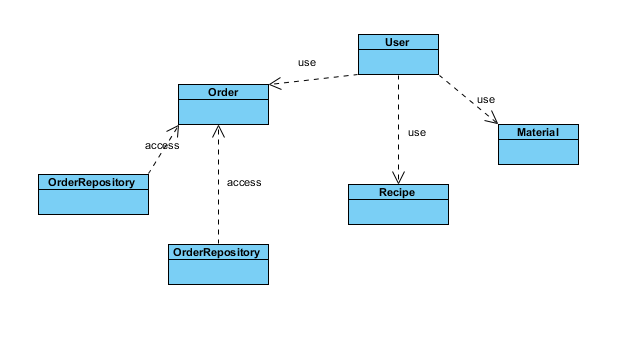


Рисунок 1. Class diagram (basic entity)

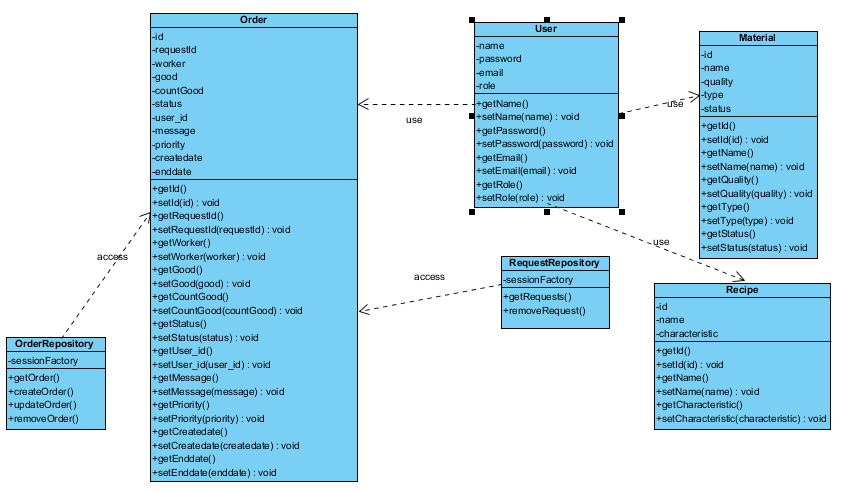


Рисунок 2. Class diagram (extended basic classes)

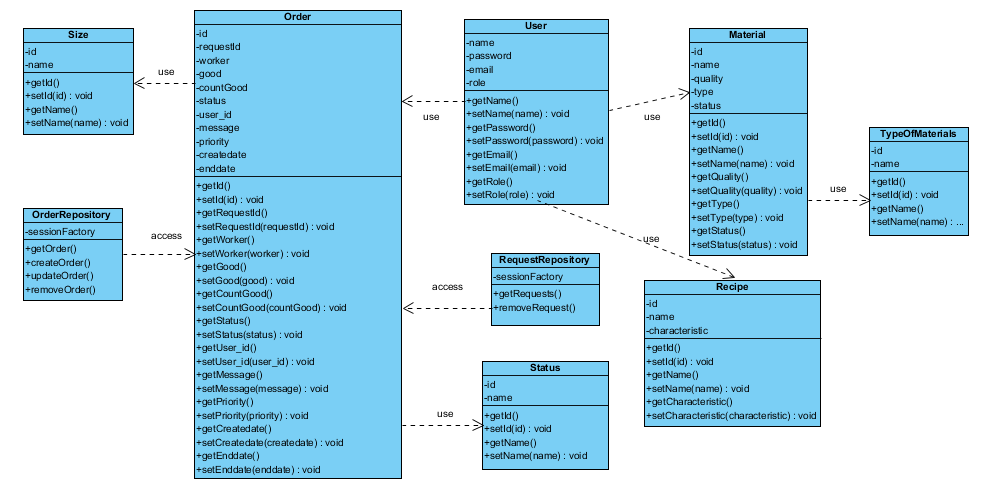


Рисунок 3. Class diagram (full version)

**Добавить готовую диаграмму по ее завершении!!!**

## Data Base Diagram

Данная диаграмма описывает архитектуру базы данных более детально по сравнению с Use Case View и является реальной структурой базы данных системы.

****

Рисунок7 – Data Base Diagram (detailed)

Таблицы Размер, Тип материалов, Статус, Фасон являются словарями, в которых хранятся уникальные значения.

Таблица Связка представляет реализацию many-to-many связи Рецептов и Материалов.

Поясним некоторые специфичные поля таблиц:

**ORDER –** таблица-заказ. В ней хранится заказ на изготовление текстильного изделия.

ORDER.ORDER\_STATUS – состояние заявки.

ORDER.ORDER\_SIZE – размер текстильного изделия

ORDER.ORDER\_RECIPE – рецепт, по которому будет выполняться изделие.

ORDER.ORDER\_FASON – тип изделия

**RECIPE\_GOOD** – таблица связка many to many таблиц RECIPE и GOOD

**SIZE** – таблица размеров текстильного изделия.

SIZE.SIZE\_NAME – размеры

STATE – таблица состояния заказа

STATE.STATE\_NAME - состояние

**FASHION** – таблица типов текстильных изделий

FASHION .FASHION\_NAME – типы изделий

**GOOD** – таблица материалов, которые используются в рецепте

GOOD.GOOD\_NAME – название материала

GOOD.GOOD\_QUALITY – качество материала

GOOD.GOOD\_GOODTYPE – тип материала.

**GOODTYPE** – таблица типов материала

GOODTYPE.GOODTYPE\_NAME– наименование типа материала

**WAREHOUSE**- таблица, описывающая наличие материалов и их кол-во на складе.

WAREHOUSE.WAREHOUSE\_NAME – имя склада

WAREHOUSE.WAREHOUSE\_GOOD – Материал

WAREHOUSE.WAREHOUSE\_COUNT – кол-во материала

**RECIPE** – таблица рецептов, по которым производится изготовление текстильного изделия.

RECIPE.RECIPE\_NAME– Название рецепта

RECIPE.RECIPE\_SPECIFICATION – Характеристики рецепта, т.е. его описание

**MAGIC-**  в данной таблице хранится вся информация о магических свойствах или особенностях.

MAGIC.MAGIC\_NAME – название свойства

MAGIC\_MAGIC\_SPECIFICATION - описание

**PRODUCT –** таблица хранит созданные товары, являющиеся результатом выполнения заказа.

PRODUCT.PRODUCT\_FINISHDATE– срок выпуска

PRODUCT.PRODUCT\_PERIODUSE– срок годности

PRODUCT.PRODUCT\_SPECIFICATION – документация

PRODUCT.MAGIC\_ID – магические свойства

PRODUCT.PRODUCT\_FASHION – фасон изделия

При заполнении нового Заказа необходимо выбрать(привязать) один из рецептов.

## Deployment Diagram

**Добавить готовую диаграмму по ее завершении!!!**

## Package Diagram

**Добавить готовую диаграмму по ее завершении!!!**

# Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

# Size and Performance

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

# Quality

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.]