Функциональные требования

Примеры использования

Версия 1.0.0

Задача ХХХХ

**ОГЛАВЛЕНИЕ**

[1. ОСНОВНЫЕ СВЕДЕНИЯ 4](#_Toc410314517)

[1.1 Термины и сокращения 4](#_Toc410314518)

[1.2 Заказчик: 4](#_Toc410314519)

[2. ОПИСАНИЕ ВЕРСИИ 1.0.0. 4](#_Toc410314520)

[2.1 Постановка требования заказчиком. 4](#_Toc410314521)

[2.1.1 Цель, причины изменений. 4](#_Toc410314522)

[2.1.2 Ссылки 4](#_Toc410314523)

[2.1.3 Конечные пользователи. 4](#_Toc410314524)

[2.2 Изменения требований к функционалу. 4](#_Toc410314525)

[3. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 4](#_Toc410314526)

[3.1.1 Описание бизнес-процесса. 4](#_Toc410314527)

[3.2 Требования к функциональности. 4](#_Toc410314528)

[3.2.1 Поставщик 4](#_Toc410314529)

[3.2.2 Товар 7](#_Toc410314530)

[3.2.3 Магазин (распределение товара по магазинам) 8](#_Toc410314531)

[3.2.4 Все товары магазина 8](#_Toc410314532)

[3.2.5 Запрос на закупку 9](#_Toc410314533)

[3.2.6 Обработка запроса 9](#_Toc410314534)

[3.2.7 Поиск товара на сайте 10](#_Toc410314535)

[3.2.8 Корзина отобранных товаров 10](#_Toc410314536)

[3.2.9 Справочники 11](#_Toc410314537)

[3.2.9.1 Тип товара 11](#_Toc410314538)

[3.2.9.2 Сегмент товара 11](#_Toc410314539)

[3.2.9.3 Типы мотивации 11](#_Toc410314540)

[3.2.9.4 Единицы измерения 12](#_Toc410314541)

[3.2.9.5 Каталог товаров 12](#_Toc410314542)

[3.2.9.6 Магазин 13](#_Toc410314543)

[3.2.9.7 Статус запроса на закупку 13](#_Toc410314544)

[3.3 Требования к производительности и результатам. 13](#_Toc410314545)

[3.4 Тестирование (интеграционное) 13](#_Toc410314546)

[3.5 Развитие 13](#_Toc410314547)

[3.6 Приложение 1. Иерархия пользовательских форм 14](#_Toc410314548)

Л**ист изменений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Дата** | **Версия** | **Статус** | **Описание** | **Автор** |
| 12.05.2014 | 1.0.0 | Создание | Создание документа | Талышев В. |
|  |  |  |  |  |

Примеры использования

# ОСНОВНЫЕ СВЕДЕНИЯ

## Термины и сокращения

## Заказчик:

|  |  |
| --- | --- |
| **Название структуры** | *ИТ* |
| **Бизнес-направление** | *Все* |
| **Заказчик (ФИО, должность)** | *Талышев В.* |
| **Ответственный за тестирование (ФИО, должность)** |  |

# ОПИСАНИЕ ВЕРСИИ 1.0.0.

Создание основы для функционала, который будет служить примером использования компонентов, библиотек и подходов.

## Постановка требования заказчиком.

### Цель, причины изменений.

Создание модуля, который будет использоваться для тестирования системных (общих) библиотек, отработки и демонстрации общих подходов к реализации функционала.

### Ссылки

JEP/Module/JepRiaShowcase – полное имя модуля (ссылка в репозитарии SVN);

JEP - репозитарий с общими модулями уровня приложения (системные библиотеки, библиотеки компонентов и т.п.);

Oracle – репозитарий с общими модулями уровня базы данных (системные модули, утилитарные модули и т.п.).

### Конечные пользователи.

Разработчики, Системные Аналитики.

## Изменения требований к функционалу.

Создание версии 1.0.0.

# ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

### Описание бизнес-процесса.

Для достижения поставленных выше целей, решим вымышленную задачу автоматизации процессов сети магазинов с интернет представительством.

## Требования к функциональности.

### Поставщик

Необходимо создать форму для внесения и редактирования информации о поставщике.

Работа с данными о поставщике возможна только при наличии у пользователя соответствующей роли.

Атрибуты поставщика:

- идентификатор поставщика – автоинкрементное целочисленное поле;

- наименование поставщика – уникальное символьное поле длинной 255 символов;

- дата действия договора (JepDateField);

- описание – необязательное для заполнения безразмерное текстовое поле (JepTextAreaField);

- безальтернативный поставщик – логическое поле 0/1 (JepCheckBoxField), значение по умолчанию – 0.

В таблице с результатами поиска поставщиков в том случае, если значение Даты, до который действует договор меньше текущей системной даты, выделять строку красным цветом.

**Добавить параметры:**

Телефон

Факс

Банк

Получатель

Расчетный счет

**Изменения в DB:**

В таблицу jrs\_supplier добавить поля:

Phone\_number (nullable, varchar2(20))

Fax\_number(nullable, varchar2(20))

bank\_bic (nullable, varchar2(9))

recipient (nullable, varchar2(150))

settlement\_account(nullable, varchar2(20)

В функцию createSupplier и процедуру updateSupplier добавить параметры для заполнения/изменения новых параметров.

В функцию findSupplier добавить в возвращаемый курсор добавляемые поля, а также через внешнее объединение с v\_bic\_bank выбрать bankname, ks.

**Реализовать функцию getBank (bankBic varchar2).**

Функция выбирает записи из v\_bic\_bank по условию bic like bankBic

Возвращает курсор, состоящий из следующих полей bic, bankname, ks

**Изменения в APP**

В GUI данные поля должны представлять поля с маской. Допустим только ввод цифр от 0 до 9

Формат полей:

(\*\*\*) \*\*\*-\*\*-\*\*

Добавить блок **Реквизиты**.

Должны быть определены следующие параметры:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Название | *Тип* | Параметр | *Состояние* | *Обязательность* | Примечание |
| 1. 7 | Банк | *Combobox c autocomplete* | bankBic в режиме редактирования, bankname в режиме просмотра |  | *нет* | Выводить подсказку:  Введите БИК банка. Допустимые символы для ввода – только цифры.  По вводу вызывается getBank(bankBik) |
| 1. 8 | Корреспондентский счет | *Text* | Ks | *просмотр* | *-* | Обновляется по выбору Банка выводится поле из курсора |
| 1. 11 | Получатель (отделение банка) | *Text* | recipient | *редактирование* | *Да* |  |
| 4 | Расчетный счет | *Text* | settlementAccount | *редактирование* | *Да* | Длина строго 20 символов, только цифры  Маска:  40\*\*\*810\*\*\*\*\*\*\*\*\*\*\*\* |

Необходимо обеспечить поиска поставщика по следующим атрибутам:

- идентификатор поставщика – по точному соответствию;

- наименование поставщика – по вхождению символов без учета регистра;

- дата действия договора – по диапазону дат;

- безальтернативный поставщик.

С помощью утилит тестирования на уровне базы данных создать тесты:

- создания поставщика;

- редактирования поставщика;

- поиска поставщика по двум комбинациям атрибутов поиска;

- удаления поставщика.

На уровне базы данных создать скрипт первоначального заполнения данных о поставщиках с различными атрибутами – 10 записей.

### Товар

Необходимо создать форму для внесения и редактирования информации о товарах конкретного поставщика (дочерняя к поставщикам форма).

Работа с данными о товарах возможна только при наличии у пользователя соответствующей роли.

Атрибуты товаров:

- наименование товара – уникальное в рамках поставщика символьное поле длинной 255 символов;

- goods\_photo – необязательное поле фото товара – blob поле (JepImageField);

- goods\_photo\_mime\_type – служебное необязательное поле - mime type фото товара, символьное длинной 70 символов;

- goods\_photo\_extension – служебное необязательное поле - окончание файла фото товара, символьное длинной 25 символов;

- goods\_portfolio – необязательное поле спецификация товара – blob поле (JepFileField);

- goods\_portfolio\_mime\_type – служебное необязательное поле - mime type спецификации товара, символьное длинной 70 символов;

- goods\_portfolio\_extension – служебное необязательное поле - окончание файла спецификации товара, символьное длинной 25 символов;

- тип товара (JepComboBoxField) – код типа товара из справочника «Тип товара»;

- сегмент товара – поле множественного выбора (JepListField) – организуется в виде подчиненной к товарам сущности, значения в которую набираются из справочника «Сегмент товара»;

- тип мотивации (JepRadioField) – код типа мотивации из справочника «Тип мотивации», значение по умолчанию – «Обычная мотивация»;

- единицы измерения (JepComboBoxField) – код единицы измерения из справочника «Единицы измерения»;

- закупочная цена (JepMoneyField) – поле типа number(15,2);

- принадлежность к каталогу – поле множественного выбора (JepTreeField) – организуется в виде подчиненной к товарам сущности, значения в которую набираются из древовидного справочника «Каталог товаров».

Необходимо обеспечить поиск товара по следующим атрибутам:

- идентификатор поставщика – по точному соответствию;

- наименование товара – по вхождению символов без учета регистра;

- тип товара;

- сегмент товара – идентификаторы сегментов товара через запятую;

- принадлежность к каталогу - идентификаторы разделов каталога товаров через запятую, выводятся также все товары, принадлежащие всем дочерним узлам из указанных.

С помощью утилит тестирования на уровне базы данных создать тесты:

- создания товара;

- редактирования товара;

- поиска товара заданного поставщика по двум комбинациям атрибутов поиска;

- удаления товара.

На уровне базы данных создать скрипт первоначального заполнения данных о товарах поставщиков с различными атрибутами по 1-3 записи на поставщика.

### Магазин (распределение товара по магазинам)

Необходимо создать форму для внесения и редактирования информации о распределении конкретного товара по магазинам (дочерняя к товарам форма).

Работа с данными о распределении товара по магазинам возможна только при наличии у пользователя соответствующей роли.

Атрибуты распределения товаров по магазинам:

- идентификатор магазина (JepComboBoxField с typing-timeout функционалом);

- количество - поле типа number(15,4);

- отпускная цена - поле типа number(15,2).

Необходимо обеспечить поиск распределения товара по следующим атрибутам:

- идентификатор товара – по точному соответствию;

- идентификатор магазина – по точному соответствию.

На уровне базы данных создать скрипт первоначального заполнения данных о распределении товаров по магазинам с различными атрибутами по 1-3 записи на товар.

### Все товары магазина

Необходимо создать форму для внесения и редактирования информации о наличии товаров в конкретном магазине (дочерняя к распределению товаров по магазинам форма).

Работа с данными о товарах магазина возможна только при наличии у пользователя соответствующей роли.

Атрибуты товаров магазина:

- идентификатор товара (JepComboBoxField с typing-timeout функционалом);

- количество - поле типа number(15,4);

- отпускная цена - поле типа number(15,2).

### Запрос на закупку

Необходимо создать форму для внесения и редактирования информации о запросах конкретного магазина на закупку товара (дочерняя ко всем товарам магазина форма).

Работа с данными о запросах на закупку возможна только при наличии у пользователя соответствующей роли.

Атрибуты запросов:

- идентификатор запроса – автоинкрементное поле;

- дата создания запроса;

- статус (JepComboBoxField) – код статуса из справочника «Статус запроса на закупку»

- идентификатор товара (JepComboBoxField с typing-timeout функционалом);

- количество - поле типа number(15,4).

Необходимо обеспечить поиск запроса по следующим атрибутам:

- идентификатор магазина – по точному соответствию;

- дата создания – по диапазону дат;

- статус;

- идентификатор товара – по точному соответствию.

На уровне базы данных создать скрипт первоначального заполнения данных о заказах с различными атрибутами по 1-3 записи на магазин.

### Обработка запроса

Необходимо создать форму для внесения и редактирования информации об обработке конкретного запроса (дочерняя к запросу на закупку форма).

Работа с данными об обработке запроса возможна только при наличии у пользователя соответствующей роли.

Атрибуты обработки запроса:

- дата записи;

- комментарий - безразмерное текстовое поле (JepTextAreaField);

- оператор, создавший запись.

Необходимо обеспечить поиск информации об обработке по следующим атрибутам:

- идентификатор запроса – по точному соответствию;

- дата записи – по диапазону дат;

- оператор, создавший запись.

На уровне базы данных создать скрипт первоначального заполнения данных об обработке заказов с различными атрибутами по 1-3 записи на заказ.

### Поиск товара на сайте

Необходимо создать форму поиска и отображения списка найденных товаров на сайте. Форма представляет собой Gwt-модуль со встроенными в произвольную html-разметку Gwt-компонентами.

Атрибуты поиска:

- наименование товара – поиск по вхождению.

Атрибуты результатов поиска:

- фото товара;

- наименование товара;

- кнопка купить, по которой товар помещается в Корзину отобранных товаров.

Необходимо отобразить элементы инструментальной панели настройки списка найденных товаров с возможностями стандартной инструментальной панели управления списком (листание, количество записей на странице, первая/последняя и т.п.).

Пример можно посмотреть на сайте http://www.ulmart.ru.

### Корзина отобранных товаров

Необходимо создать форму отображения списка отобранных товаров на сайте. Информация об отобранных товарах сохраняется на стороне браузера в cookies. Форма представляет собой Gwt-модуль со встроенными в произвольную html-разметку Gwt-компонентами.

Атрибуты отобранных товаров:

- фото товара;

- наименование товара;

- кнопка удалить, по которой товар удаляется из Корзины отобранных товаров.

### Справочники

Для редактирования всех справочников создается одна роль.

#### Тип товара

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты типа товара:

- код типа – символьное уникальное поле;

- наименование типа – символьное уникальное поле длинной 100 символов.

Необходимо обеспечить получение значений всего справочника get-методом без проверки ролей для заполнения соответствующих выпадающих списков пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: Продукты питания, Промышленные товары, Печатная продукция.

#### Сегмент товара

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты сегмента товара:

- код сегмента – символьное уникальное поле;

- наименование сегмента – символьное уникальное поле длинной 150 символов.

Необходимо обеспечить получение значений всего справочника get-методом без проверки ролей для заполнения соответствующих списков пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: Повседневного спроса, Туризм и отдых, Для дома и дачи, Автозапчасти, Игрушки.

#### Типы мотивации

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты типов мотивации:

- код типа мотивации – символьное уникальное поле;

- наименование типа мотивации – символьное уникальное поле длинной 50 символов;

- комментарий – символьное поле длинной 4000 символов.

Необходимо обеспечить получение значений всего справочника get-методом без проверки ролей для заполнения соответствующих списков пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: Обычная мотивация (Специальная мотивационная схема отсутствует), Квартальная мотивация (Бонусы по результатам продажи товара за квартал), Уровень продаж в месяц (Бонус за превышение некого уровня продаж в месяц), Процент с дохода (Выплата процента с прибыли за товар).

#### Единицы измерения

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты единиц измерения:

- код измерения – символьное уникальное поле;

- наименование единицы измерения – символьное уникальное поле длинной 50 символов;

- краткое наименование единицы измерения – символьное уникальное поле длинной 50 символов (возможна html-последовательность).

Необходимо обеспечить получение значений всего справочника get-методом без проверки ролей для заполнения соответствующих выпадающих списков пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: Штуки (Шт.), Килограммы (Kg), Литры (L), Метры (M), Квадратные метры (Кв.M).

#### Каталог товаров

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты каталога товаров:

- идентификатор раздела каталога – автоинкрементное поле;

- наименование раздела каталога – символьное поле длинной 100 символов;

- ссылка на идентификатор родительского раздела каталога – необязательное поле.

Необходимо обеспечить получение значений дочерних элементов по идентификатору родительского get-методом без проверки ролей для заполнения соответствующих элементов пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: взять с сайта ozon.ru первые три уровня иерархии каталога товаров (блок слева на главной странице).

#### Магазин

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты магазина:

- идентификатор магазина – автоинкрементное поле;

- наименование магазина – символьное уникальное поле длинной 255 символов.

Необходимо обеспечить поиска магазина по следующим атрибутам:

- наименование магазина – по вхождению символов без учета регистра.

На уровне базы данных создать скрипт первоначального заполнения данных о магазинах с различными атрибутами – 10 записей.

#### Статус запроса на закупку

Пользовательский интерфейс, в текущей версии, отсутствует.

Атрибуты статуса запроса на закупку:

- код статуса – символьное уникальное поле;

- наименование статуса – символьное уникальное поле длинной 100 символов.

Необходимо обеспечить получение значений всего справочника get-методом без проверки ролей для заполнения соответствующих выпадающих списков пользовательского интерфейса.

На уровне базы данных создать скрипт первоначального заполнения справочника. Возможные значения: Взят в обработку, Выбор поставщика, Заказ у поставщика, Поставка, Запрос выполнен, Запрос отклонен.

## Требования к производительности и результатам.

Реакция системы на действия пользователя должна быть не более 3 секунд.

## Тестирование (интеграционное)

TBD.

## Развитие

Для дальнейшего использования функционал для отладки компонентов, подходов, разработки автоматизированных тестов возможны следующие направления развития (добавления функционала):

- чат, обсудить товар в чате;

- аналитика продаж в различных разрезах.

## Приложение 1. Иерархия пользовательских форм

Поставщик->Товар

Товар->Магазин

Магазин->Все товары магазина

Все товары магазина->Запрос на закупку

Запрос на закупку->Обработка запроса