Оглавление

[Глава 3. Первоначальная схема базы данных 3](#_Toc120644927)

[Глава 4. Переработанная схема базы данных 4](#_Toc120644928)

[Глава 6. Диаграмма переходов между экранными формами 6](#_Toc120644929)

[Глава 7. Список ролей 7](#_Toc120644930)

[Глава 8. Экранные формы 8](#_Toc120644931)

[Домашняя страница 8](#_Toc120644932)

[Просмотр очереди задач – faceted view 8](#_Toc120644933)

[Просмотр очереди задач – classic view 10](#_Toc120644934)

[Просмотр очереди задач – calendar view 10](#_Toc120644935)

[Подробная информация о задаче 11](#_Toc120644936)

[Заведение\Редактирование задачи 12](#_Toc120644937)

[Справочник сотрудников 13](#_Toc120644938)

[Информация о сотруднике 14](#_Toc120644939)

[Персональная информация 15](#_Toc120644940)

[Статистика 15](#_Toc120644941)

[Администрирование 16](#_Toc120644942)

[Управление пользователями 17](#_Toc120644943)

[Управление ролями 17](#_Toc120644944)

[Управление типами заявок 18](#_Toc120644945)

[Управление приоритетами заявок 18](#_Toc120644946)

[Управление статусами заявок 19](#_Toc120644947)

[Управление департаментами 19](#_Toc120644948)

[Управление фотографиями сотрудников 20](#_Toc120644949)

[Глава 9. Внутренние настройки 21](#_Toc120644950)

[ Buttons 21](#_Toc120644951)

[ Calendar 21](#_Toc120644952)

[ Check Box 22](#_Toc120644953)

[ Classic Report 22](#_Toc120644954)

[ Date Picker 23](#_Toc120644955)

[ Dynamic Action 23](#_Toc120644956)

[ Form 24](#_Toc120644957)

[ Interactive Report 24](#_Toc120644958)

[ Link 25](#_Toc120644959)

[ List of Values 25](#_Toc120644960)

[ List 26](#_Toc120644961)

[ Page Item 26](#_Toc120644962)

[ Security 27](#_Toc120644963)

[ Select List 27](#_Toc120644964)

[ Static Content 28](#_Toc120644965)

[ Charts 29](#_Toc120644966)

[ Authorization scheme 29](#_Toc120644967)

[ Login 30](#_Toc120644968)

[ Processes 32](#_Toc120644969)

[ Server-side Condition 32](#_Toc120644970)

[ Application items 33](#_Toc120644971)

# Глава 1. Первоначальная схема базы данных

Рисунок 1. Домашняя страница

Рисунок 3.2. Предыдущая схема

# Глава 2. Переработанная схема базы данных

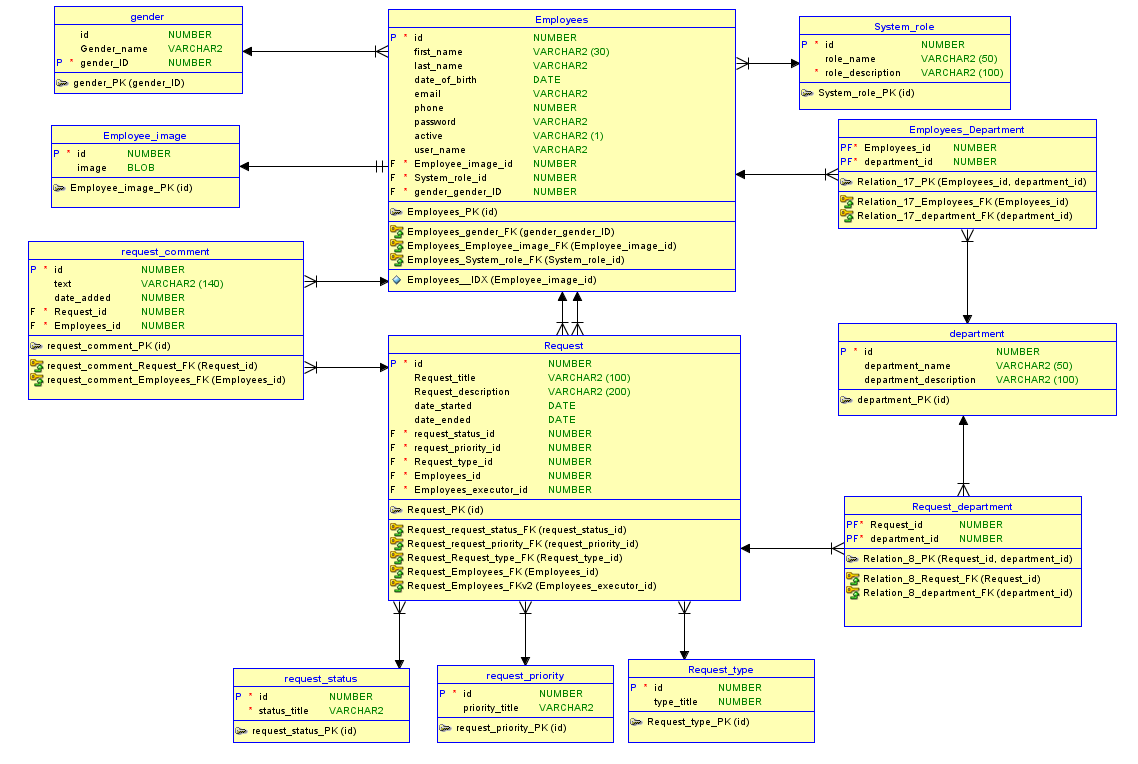


Рисунок 3.3. Актуальная схема

Глава 3. Обоснования изменения схемы базы данных

Request:

Отказ от связей «Многие ко многим» к таблицам «Request\_priority», «Request\_status», «Request», от чего в таблицы появились дополнительные столбцы «Request\_priority\_ID», «Request\_status\_ID», «Executor\_employee\_id».

Такая связь задумывалась для истории изменений, но ее реализация в рамках данной задачи мне показалось избыточна, поэтому я ее упростил.

Employees:

Отказ от связей «Многие ко многим» к таблицам «Request», «System\_role», от чего в таблицы появились дополнительные столбцы «System\_role \_ID». Также появился столбец «Employee\_image\_id» из-за появления таблицы с фотографией сотрудника.

Employee\_image:

Пришла на смену Request\_image, так как фотографии сотрудников показались мне более логичным способ использования этого типа данных.

# Глава 4. Диаграмма переходов между экранными формами

Легенда диаграммы:

Односторонней стрелкой отмечены переходы только в одну сторону

Двусторонней стрелкой отмечены переходы в обе стороны

Обычный текст – обычные страницы приложения

Курсивный текст – модальные страницы



Рисунок 3.4. Карта сайта

# Глава 5. Список ролей

* Администратор  
  Имеет доступ к приложению и всем его функциям, к базе данных, а также к среде разработки приложения.
* Исполнитель

Может создавать, редактировать и закрывать заявки. Доступ есть на все страницы, кроме «Администрирование» и дочерних.

* Пользователь.

Может создавать и смотреть заявки. Доступ есть на все страницы, кроме «Статистика», «Администрирование» и дочерних.

# Глава 6. Экранные формы

## Домашняя страница

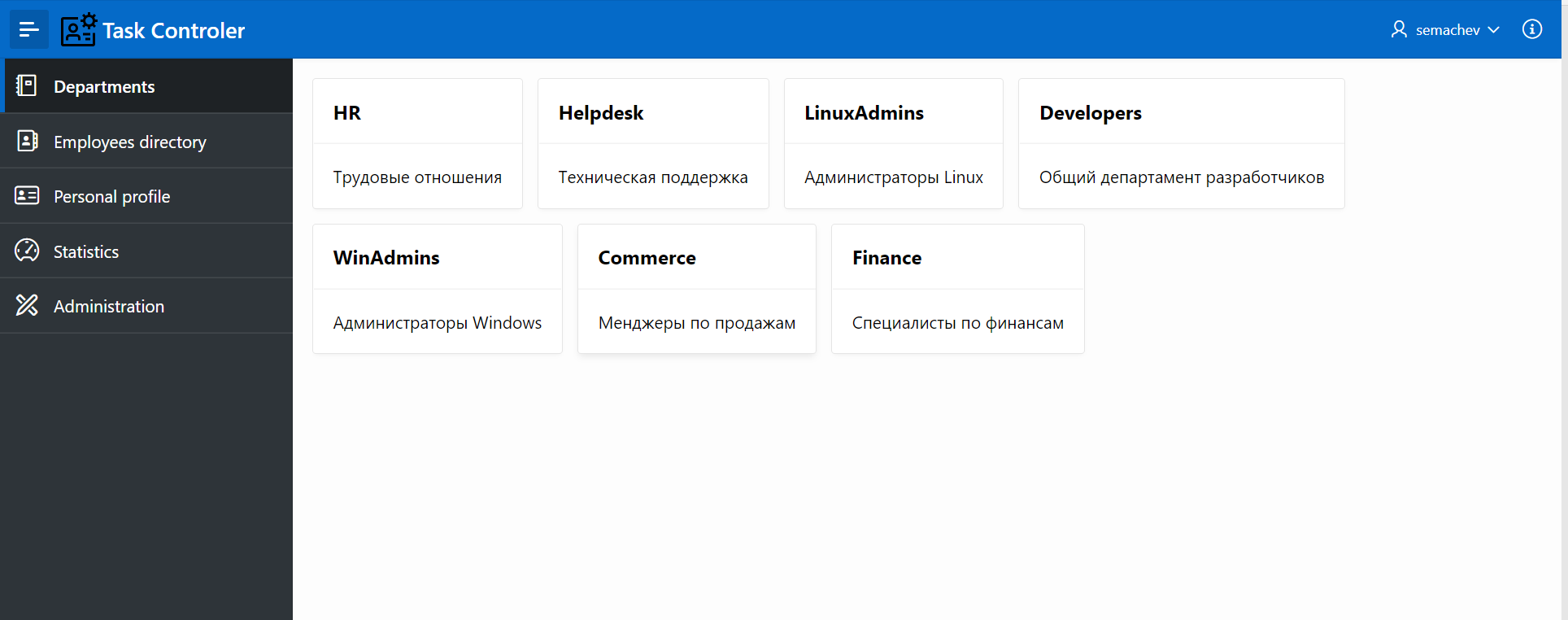


Рисунок 5. Домашняя страница

Она же страница списка департаментов. Выполнено в виде карточек, которые являются гиперссылками. Карточки ведут на страницы с отчетами по задачам каждого департамента. На следующую страницу передается ID департамента, и на его основе формируется отчет.

## Просмотр очереди задач – faceted view

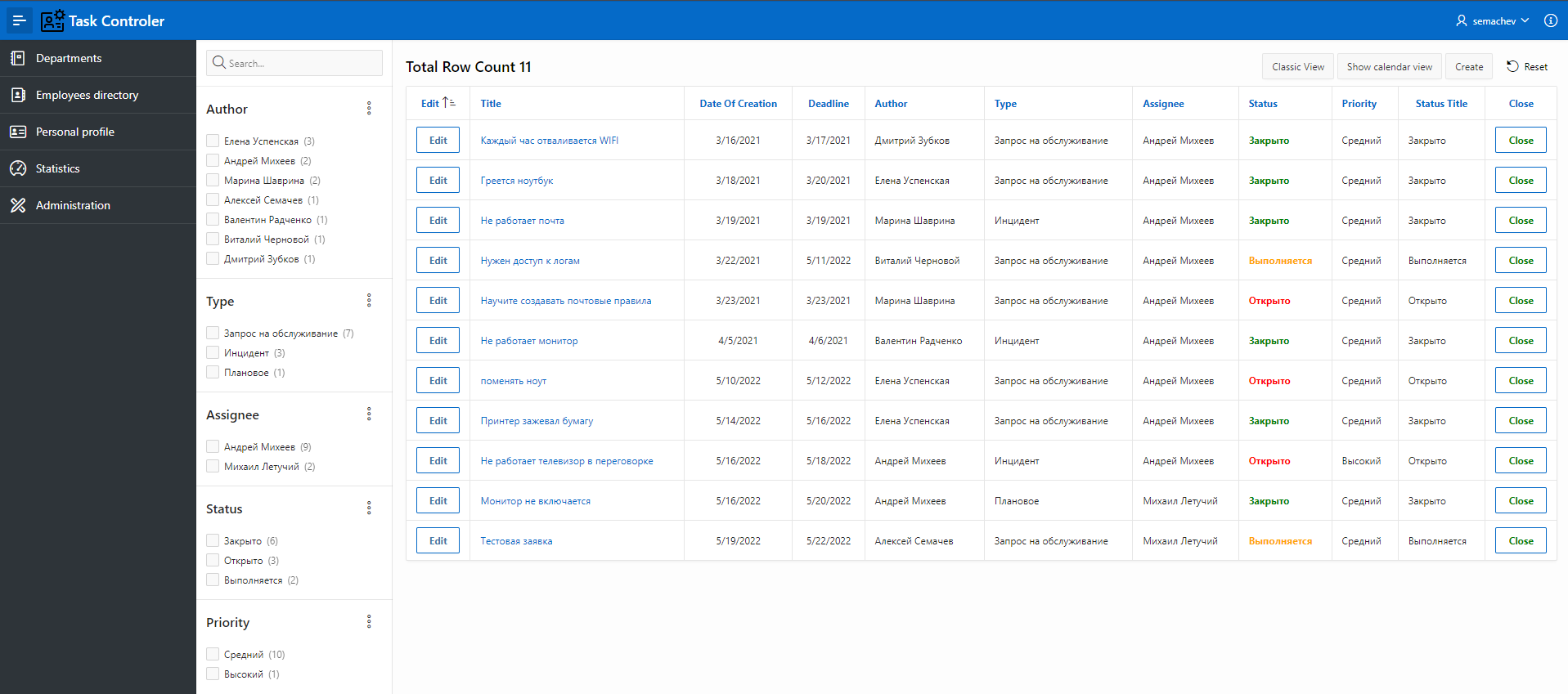


Рисунок 6. Очередь задач департамента

На этой странице отображаются все задачи определенного департамента. В левой части находятся фильтры, по которым задачи можно сортировать. Например, статус задачи – открыто\закрыто\выполняется.

Отчет выполнен с помощью региона «Faceted search». Первый и последний столбец – добавленные статические столбцы с типом «Link», добавляющие некоторую интерактивность. Кнопка Edit позволяет перейти в форму редактирования задачи, а кнопка Close позволяет быстро закрыть задачу.

В верхней части экрана есть также несколько кнопок – Classic view и calendar view переадресовывают на страницы, которые выводят ту же очередь заявок, но в другом виде. Кнопка Create открывается модальное окно с формой для создания новой задачи.

Столбец Title является ссылкой на модальное окно с подробной информацией о заявке. В этом окне также можно добавить комментарий к заявке.

Также здесь динамические раскрашивается столбец Status в зависимости от того, какое значение он имеет. Сделано это с помощью дополнительного столбца Status, у которого есть особое HTML отображение, раскрашивающие и выводящие значение столбца Status\_Title:

<span style="color:#STATUS#;font-weight:bold;">#STATUS\_TITLE#</span>

Запрос SQL выглядит так:

**select** 'Edit' **as** Edit**,**

r**.**ID**,**

r**.**REQUEST\_TITLE **as** Title**,**

r**.**REQUEST\_DESCRIPTION **as** "Description"**,**

r**.**DATE\_STARTED **as** "Date of creation"**,**

r**.**DATE\_ENDED **as** Deadline**,**

**concat(concat(**e**.**first\_name**,** ' '**),** e**.**last\_name**)** **as** Author**,**

rt**.**type\_title **as** **Type,**

**concat(concat(**ex**.**first\_name**,** ' '**),** ex**.**last\_name**)** **as** Assignee**,**

rs**.**status\_title**,**

rp**.**priority\_title **as** Priority**,**

'Close' **as** **Close,**

**case** **when** rs**.**status\_title **=** 'Открыто' **then** 'red'

**when** rs**.**status\_title **=** 'Выполняется' **then** 'orange'

**when** rs**.**status\_title **=** 'Закрыто' **then** 'green'

**end** Status

**from** REQUEST r **join** REQUEST\_DEPARTMENT rd **ON** **(**r**.**ID **=** rd**.**REQUEST\_ID**)**

**join** EMPLOYEES e **on** **(**e**.**ID **=** r**.**EMPLOYEES\_ID**)**

**left** **join** EMPLOYEES ex **on** **(**ex**.**id **=** r**.**executor\_employee\_id**)**

**join** REQUEST\_TYPE rt **on** **(**r**.**REQUEST\_TYPE\_ID **=** rt**.**ID**)**

**join** request\_status rs **ON** **(**r**.**Request\_status\_id **=** rs**.**id**)**

**join** Request\_Priority rp **ON** **(**r**.**Request\_Priority\_id **=** rp**.**id**)**

**where** rd**.**DEPARTMENT\_ID **=** **:**P30\_DEP\_ID

## Просмотр очереди задач – classic view

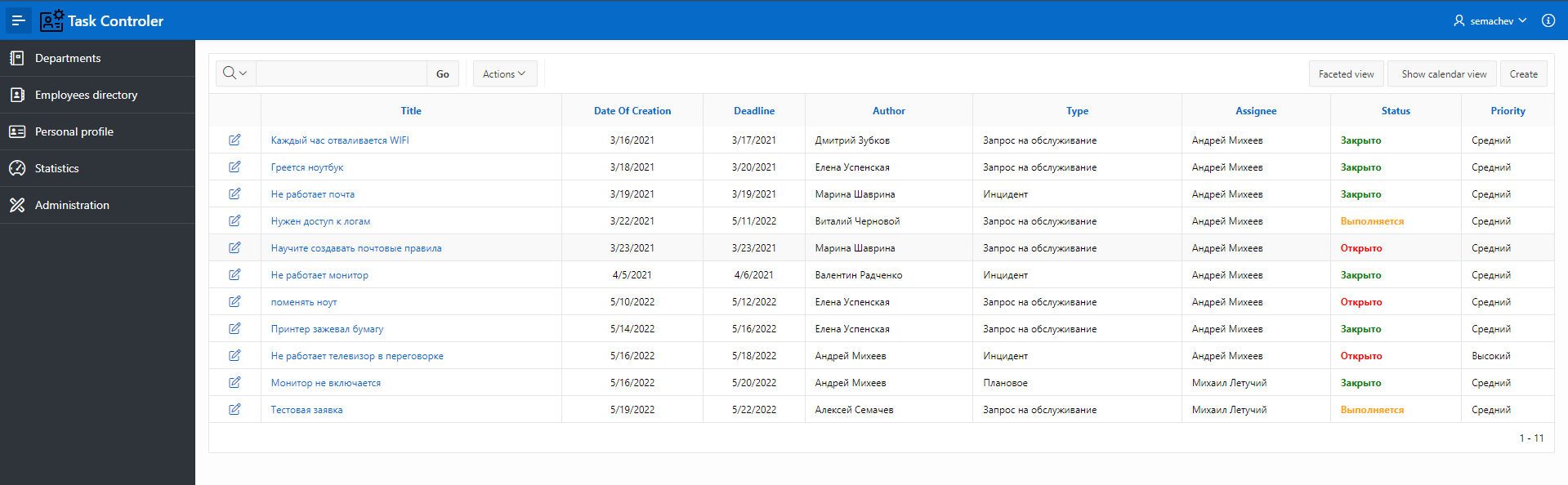
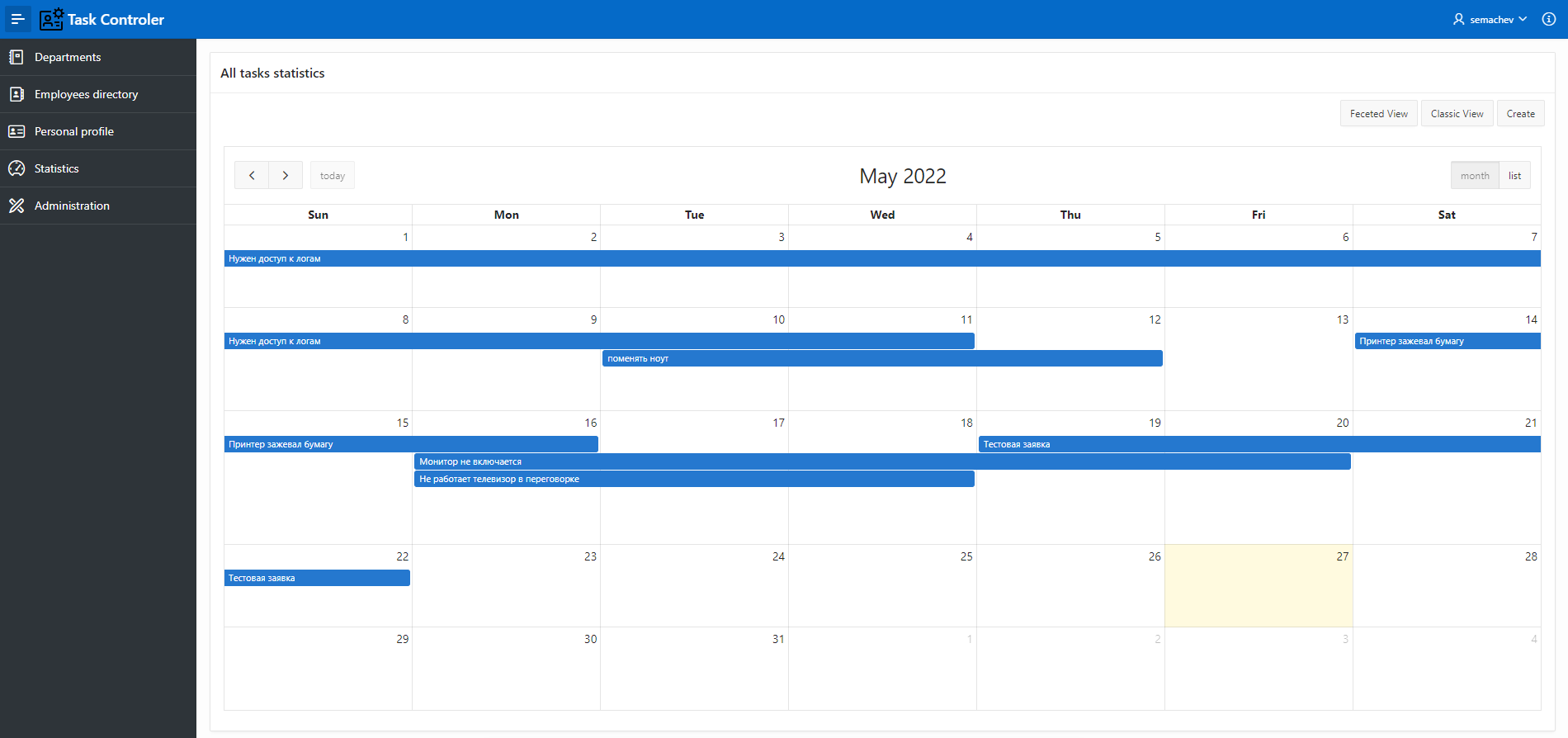


Рисунок 7. Просмотр очереди задач – classic view

Все то же самое, только другой способ представления. На этот раз, с помощь Interactive report. Он появился первый, и мне жалко было его удалять. Поэтому я решил сделать возможность переключение между режимами отображение – такое есть во многих таск трекерах. SQL запрос точно такой же. По кнопкам переключения вида между страницами передается ID департамента.

## Просмотр очереди задач – calendar view

*Рисунок 8. Просмотр очереди задач – calendar view*

Здесь тот SQL запрос, но для его отображения используется регион Calendar. Срок задачи определяется двумя датами – датой создания и дедлайном. Каждая задача кликабельна – при нажатии откроется модальное окно с подробной информацией о задаче.

## Подробная информация о задаче

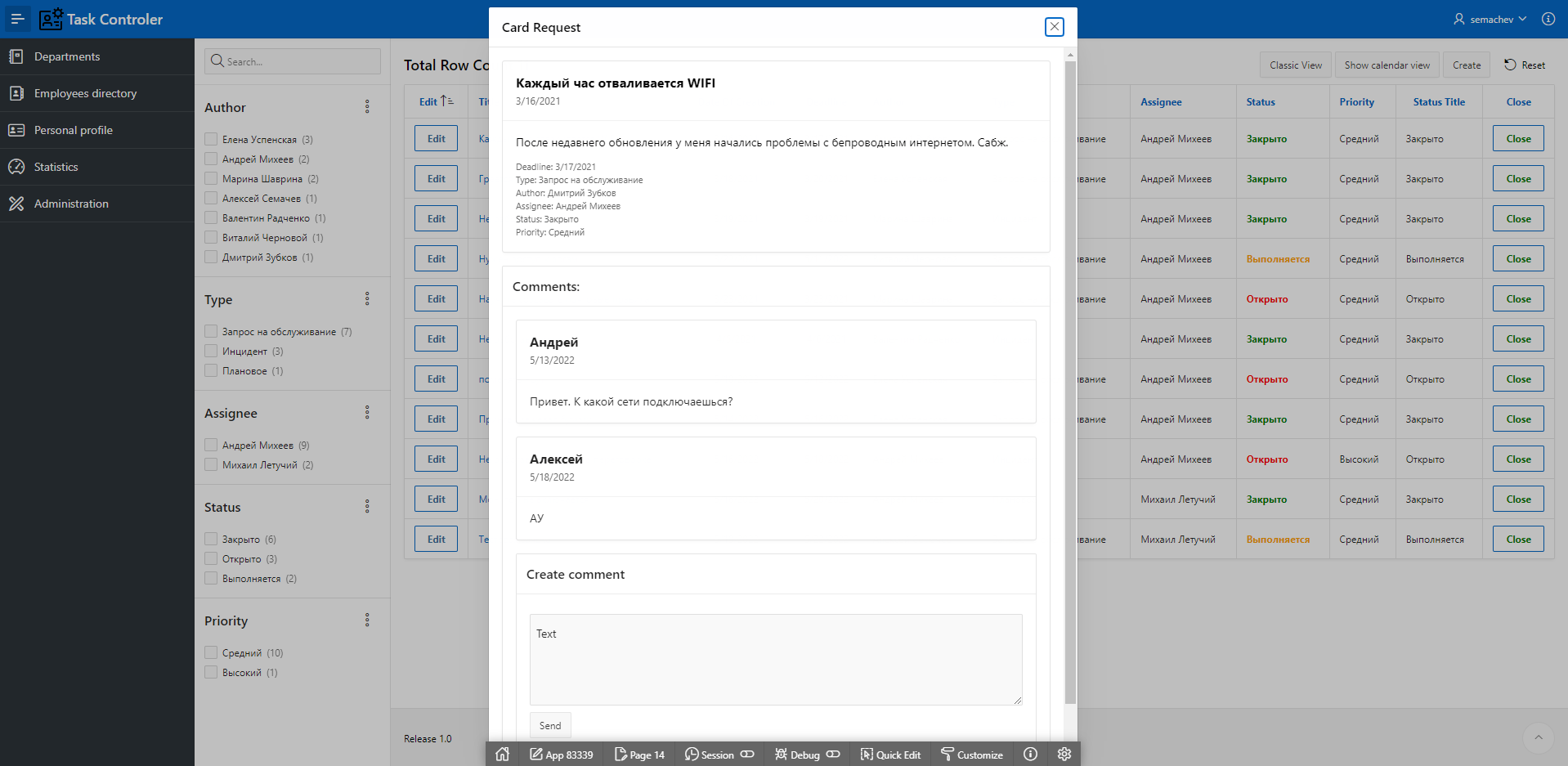


Рисунок 9. Подробная информация о задаче

Данное модальное окно является отображением подробной информации о задаче, а также позволяет просмотреть и оставить комментарии к задаче.

Информация о задаче выводится с помощью региона Cards, на основе переданного с предыдущей страницы ID задачи. SQL запрос выглядит так:

select r.ID,

r.REQUEST\_TITLE as Title,

r.REQUEST\_DESCRIPTION as "Description",

r.DATE\_STARTED as "Date of creation",

r.DATE\_ENDED as Deadline,

concat(concat(e.first\_name, ' '), e.last\_name) as Author,

rt.type\_title as Type,

concat(concat(ex.first\_name, ' '), ex.last\_name) as Assignee,

rs.status\_title as Status,

rp.priority\_title as Priority

from REQUEST r join REQUEST\_DEPARTMENT rd ON (r.ID = rd.REQUEST\_ID)

join EMPLOYEES e on (e.ID = r.EMPLOYEES\_ID)

join REQUEST\_TYPE rt on (r.REQUEST\_TYPE\_ID = rt.ID)

join request\_status rs ON (r.Request\_status\_id = rs.id)

join Request\_Priority rp ON (r.Request\_Priority\_id = rp.id)

join EMPLOYEES ex ON (r.executor\_employee\_id = ex.ID) where r.ID = :P14\_REQ\_ID

Отображение комментариев также сделано с помощью Cards с некоторыми настройками. SQL запрос такой:

select rc.TEXT,

rc.DATE\_ADDED,

e.first\_name

from REQUEST\_COMMENT rc join EMPLOYEES e ON (rc.EMPLOYEES\_ID = e.ID) where REQUEST\_ID = :P14\_REQ\_ID

Создание комментариев выполнено с помощью формы. Полей в форме чуть-чуть больше, чем отображается для заполнения. ID задачи берется из прошлой страницы. Для даты комментария берется текущая системная дата с мощью PL\SQL блока:

to\_char(sysdate,'MM.DD.YYYY')

А для автора комментария используется ID текущего пользователя сессии. SQL запрос:

SELECT id FROM employees WHERE user\_name = :SESSION\_USER\_NAME;

## Заведение\Редактирование задачи

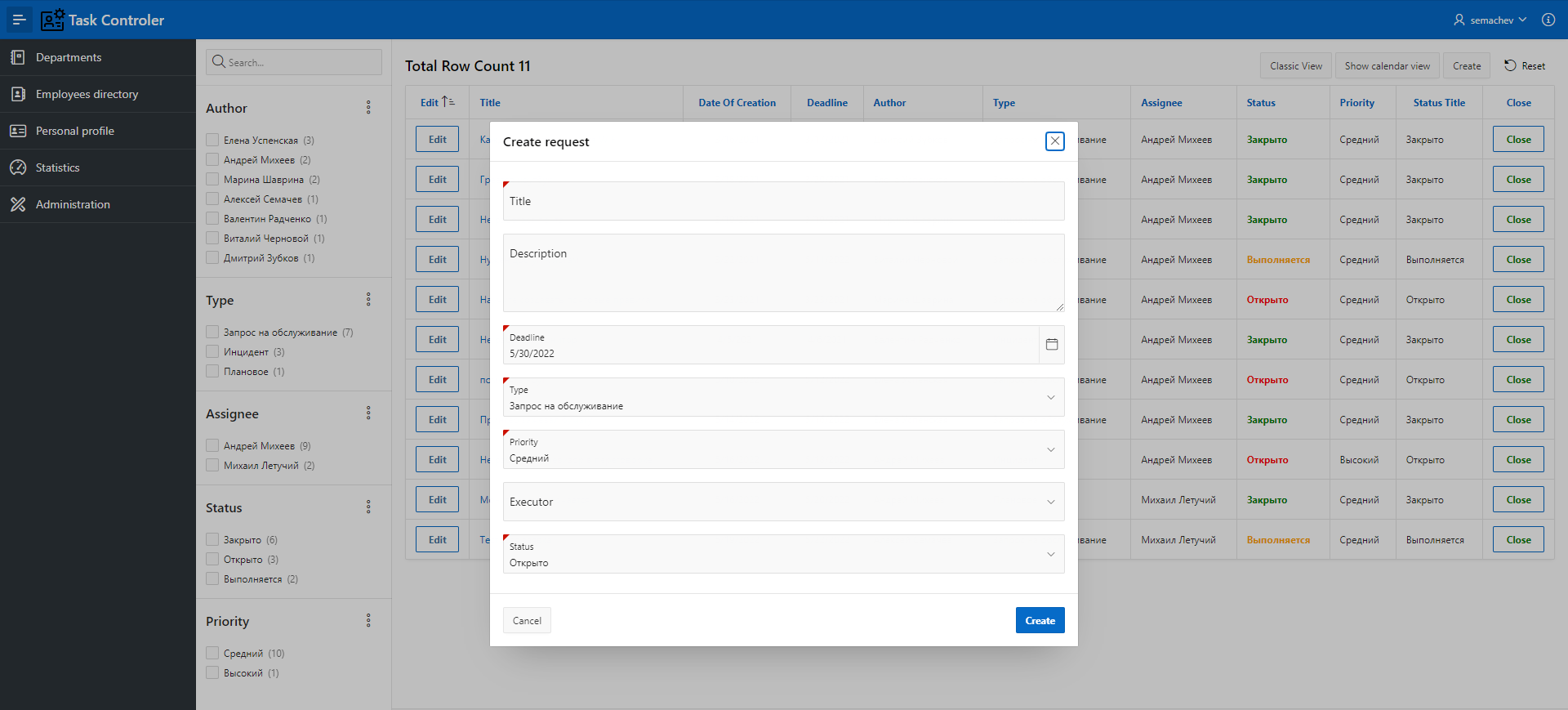


Рисунок 10. Заведение\Редактирование задачи

Данное модальное окно вызывается по кнопке «Create» или «Edit». При вызове этого модального окна через кнопку Edit в форме отображают кнопки Save и Delete. Так вот, эти кнопки может нажать только пользователь с ролью «Администратор» или «Исполнитель». Простой пользователь удалить или отредактировать задачу не сможет.

Некоторое поля заполняются по умолчанию.

Для даты комментария берется текущая системная дата с мощью PL\SQL блока:

to\_char(sysdate,'MM.DD.YYYY')

А для автора комментария используется ID текущего пользователя сессии. SQL запрос:

SELECT id FROM employees WHERE user\_name = :SESSION\_USER\_NAME;

В поле Deadline добавляется текущая дата + 3 дня, но ее можно поменять на свою.

А в поле Executor можно выбрать только тех пользователей, кто числится в этом департаменте и имеет роль «Исполнитель» или «Администратор». SQL запрос:

**SELECT** **concat(concat(**e**.**first\_name**,** ' '**),** e**.**last\_name**)** **as** display\_val

**,**e**.**id **as** read\_val

**FROM** employees e **join** employee\_department ed **ON** **(**e**.**id **=** ed**.**employees\_id**)**

**where** ed**.**department\_id **=** **:**P24\_department\_id **and** **(**e**.**role **=** 1 **or** e**.**role **=** 2**);**

## Справочник сотрудников

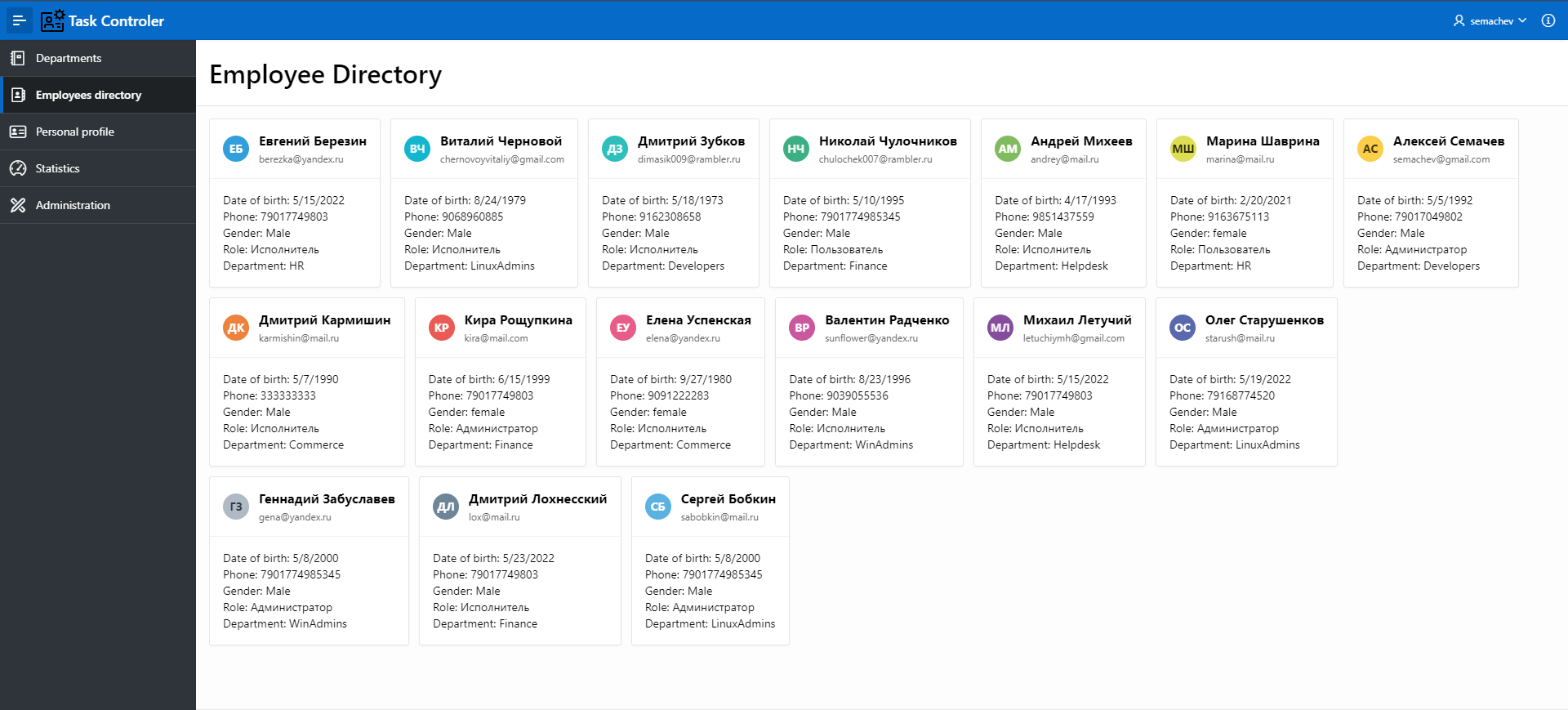


Рисунок 11. Справочник сотрудников

Эта страница выполнена с помощью региона Cards, которая выводит список всех сотрудников. В основном блоке выводится ФИО и электронная почта, во второстепенном блоке выводится остальная полезная информация. Каждая карточка сотрудника кликабельна и при нажатии открывается окно с подробной информацией о сотруднике.

SQL запрос для карточек:

select e.ID,

concat(concat(e.first\_name, ' '), e.last\_name) as Name,

e.DATE\_OF\_BRITH,

e.EMAIL,

e.PHONE,

e.PASSWORD,

g.gender\_name,

r.role\_name,

e.ACTIVE,

e.USER\_NAME,

d.department\_NAME

from EMPLOYEES e join GENDER g on (g.id = e.GENDER\_ID)

join system\_role r on (e.ROLE = r.id)

join employee\_department ed on (e.id = ed.employees\_id)

join department d on (ed.department\_id = d.id)

## Информация о сотруднике

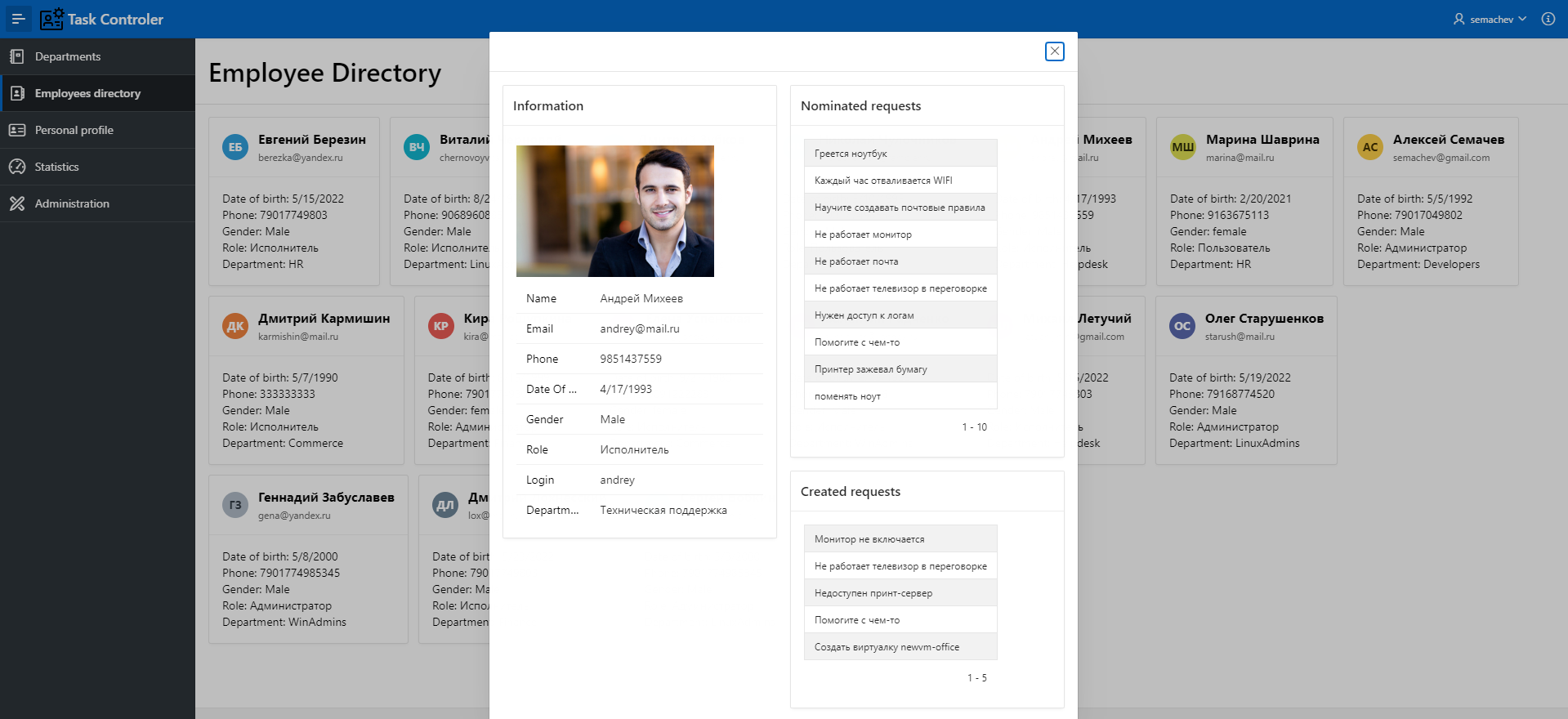


Рисунок 12. Информация о сотруднике

Данная модальная страница отображается дополнительную информацию о сотруднике. Здесь 3 региона, и все – classic report. Ну, кроме фотографии, она отдельно. В двух отчетах справа выводится информации о назначенных задачах на сотруднике, и о созданных задачах сотрудником.

SQL запрос для информации о сотруднике:

select

concat(concat(e.first\_name, ' '), e.last\_name) as Name,

e.DATE\_OF\_BRITH as "Date of birth",

e.EMAIL as Email,

e.PHONE as Phone,

g.gender\_name as Gender,

r.role\_name as Role,

e.USER\_NAME as Login,

d.department\_description as Department

from EMPLOYEES e join GENDER g on (g.id = e.GENDER\_ID)

join system\_role r on (e.ROLE = r.id)

join employee\_department ed on (e.id = ed.employees\_id)

join department d on (ed.department\_id = d.id)

where e.id = :P5\_ID

SQL запрос для назначенных задач:

select

REQUEST\_TITLE

from REQUEST r where EXECUTOR\_EMPLOYEE\_ID = :P5\_ID

SQL запрос для созданных задач:

select

REQUEST\_TITLE

from REQUEST r where EMPLOYEEs\_ID = :P5\_ID

## Персональная информация

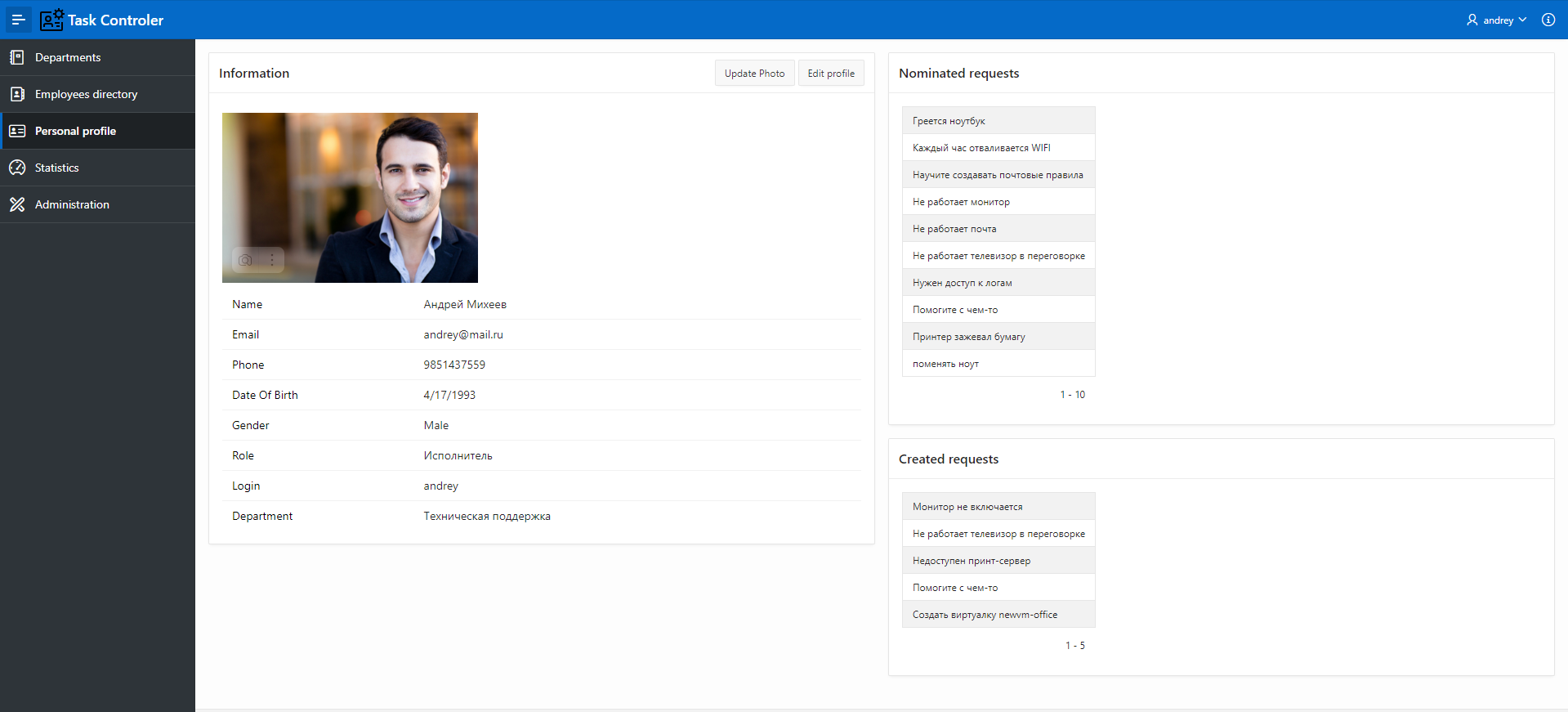


Рисунок 13. Персональная информация

Здесь все примерно то же самое, что и на странице выше. Запрос делается по ID текущего пользователя сессии. Из отличий две кнопки – обновить фотографию и редактировать профиль. Обе кнопки открываются соответствующие формы в виде модального окна.

SQL запрос для получения ID пользователя:

SELECT id FROM employees WHERE user\_name = :SESSION\_USER\_NAME;

Остальные запросы такие же, как на странице из справочника.

## Статистика

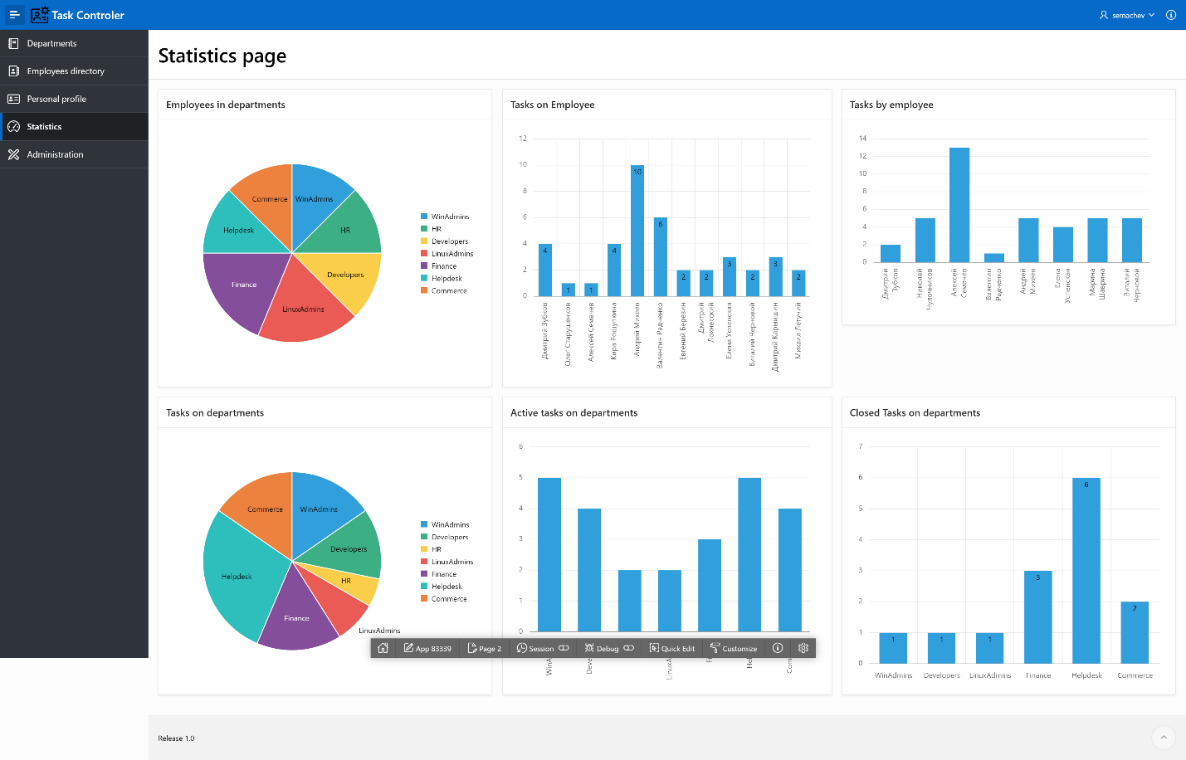


Рисунок 14. Статистика

Данная страница отображается статистику по сотрудникам и департаментам. Здесь используется регионы Charts.

SQL запрос для количества сотрудников по департаментам:

Select d.department\_name Department, count(ed.employees\_id) Employees

from Department d JOIN Employee\_department ed ON (d.id = ed.department\_id)

GROUP BY d.department\_name

SQL запрос для количества заявок по сотрудникам:

Select concat(concat(e.first\_name, ' '), e.last\_name) as Assignee, count(r.id) as Tasks

FROM Employees e JOIN Request r on (e.id = r.EXECUTOR\_EMPLOYEE\_ID)

GROUP BY concat(concat(e.first\_name, ' '), e.last\_name)

SQL запрос для количества заявок по авторству:

Select concat(concat(e.first\_name, ' '), e.last\_name) as Author, count(r.id) as Tasks

FROM Employees e JOIN Request r on (e.id = r.EMPLOYEES\_ID)

GROUP BY concat(concat(e.first\_name, ' '), e.last\_name)

SQL запрос для количества заявок по департаментам:

SELECT d.department\_name as Department, count(rd.request\_id) as Tasks

FROM department d join request\_department rd ON (d.id = rd.department\_id)

GROUP BY d.department\_name

SQL запрос для количества активных заявок по департаментам:

SELECT d.department\_name as Department, count(rd.request\_id) as Tasks

FROM department d join request\_department rd ON (d.id = rd.department\_id)

join request r ON (r.id = rd.request\_id)

WHERE r.REQUEST\_STATUS\_ID = 1 OR r.REQUEST\_STATUS\_ID = 2

GROUP BY d.department\_name

SQL запрос для количества закрытых заявок по департаментам:

SELECT d.department\_name as Department, count(rd.request\_id) as Tasks

FROM department d join request\_department rd ON (d.id = rd.department\_id)

join request r ON (r.id = rd.request\_id)

WHERE r.REQUEST\_STATUS\_ID = 3

GROUP BY d.department\_name

## Администрирование



Рисунок 15. Администрирование

Данная страница является просто местом доступа к страницам для управления пользователями\департаментами\фотографиями и прочим. Сделана с помощью региона List, для которого создан список со страницами, ссылку к которым нужно отображать.

Для этой страницы настроен доступ только для пользователей с ролью «Администратор».

## Управление пользователями

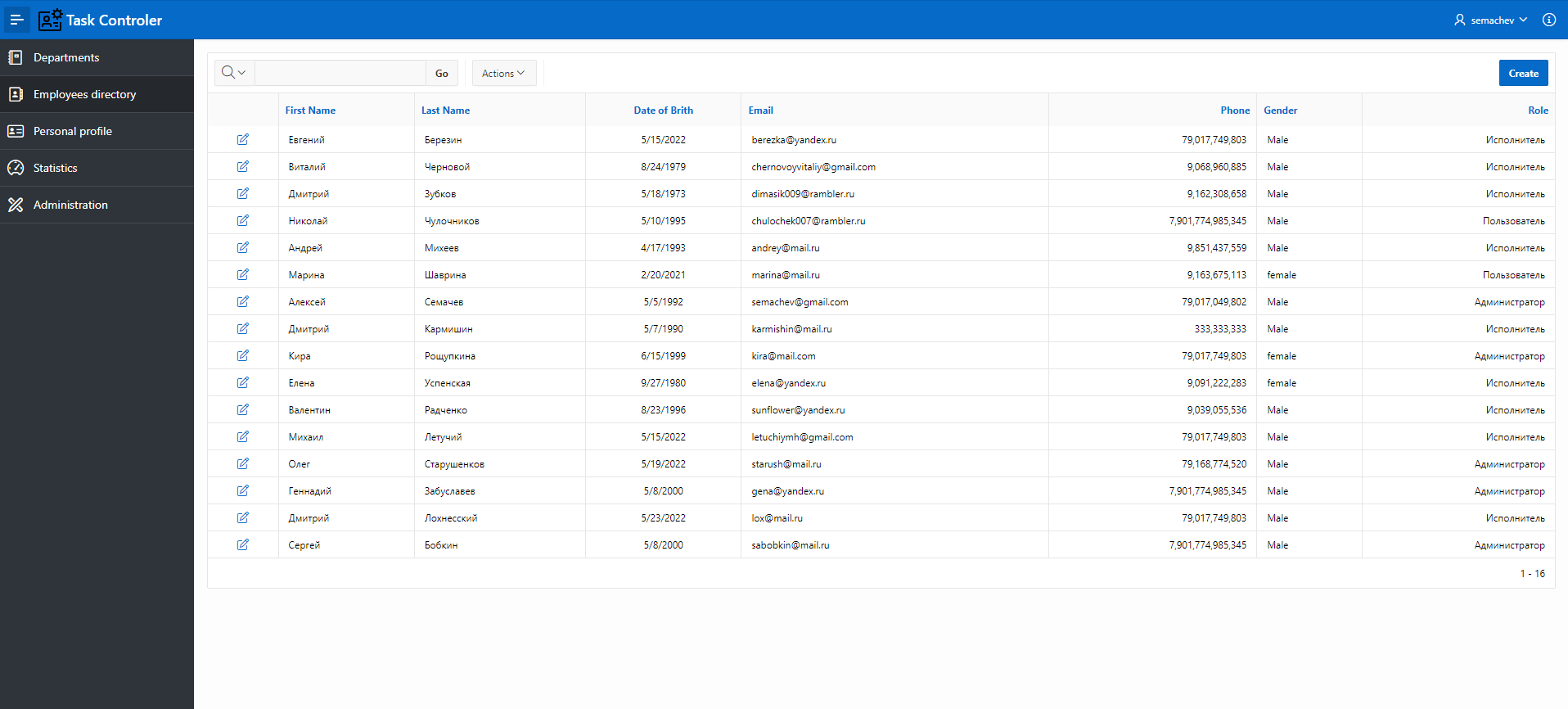


Рисунок 16. Управление пользователями

Данная страница отображает список всех сотрудников, зарегистрированных в информационной системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущего пользователя.

Страница не содержит SQL запроса, а столбцы трансформированы в удобочитаемый вид с помощью List of values.

## Управление ролями

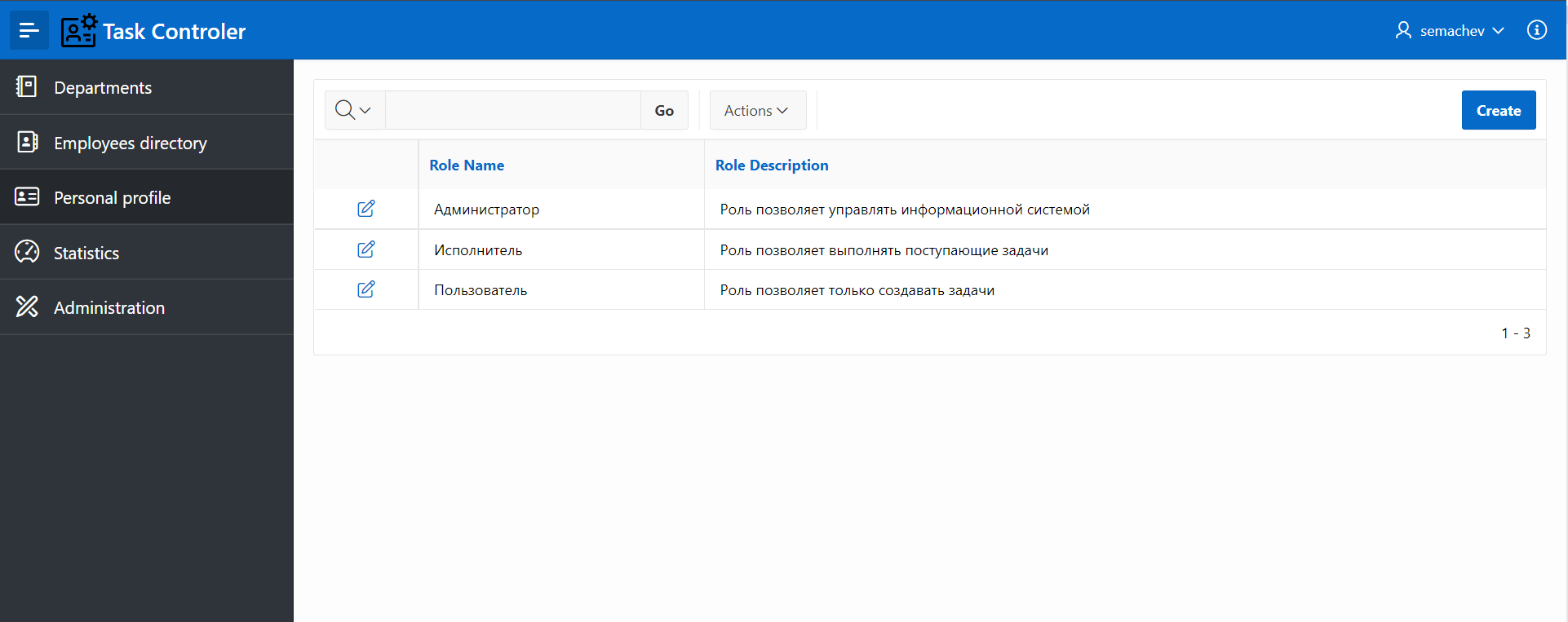
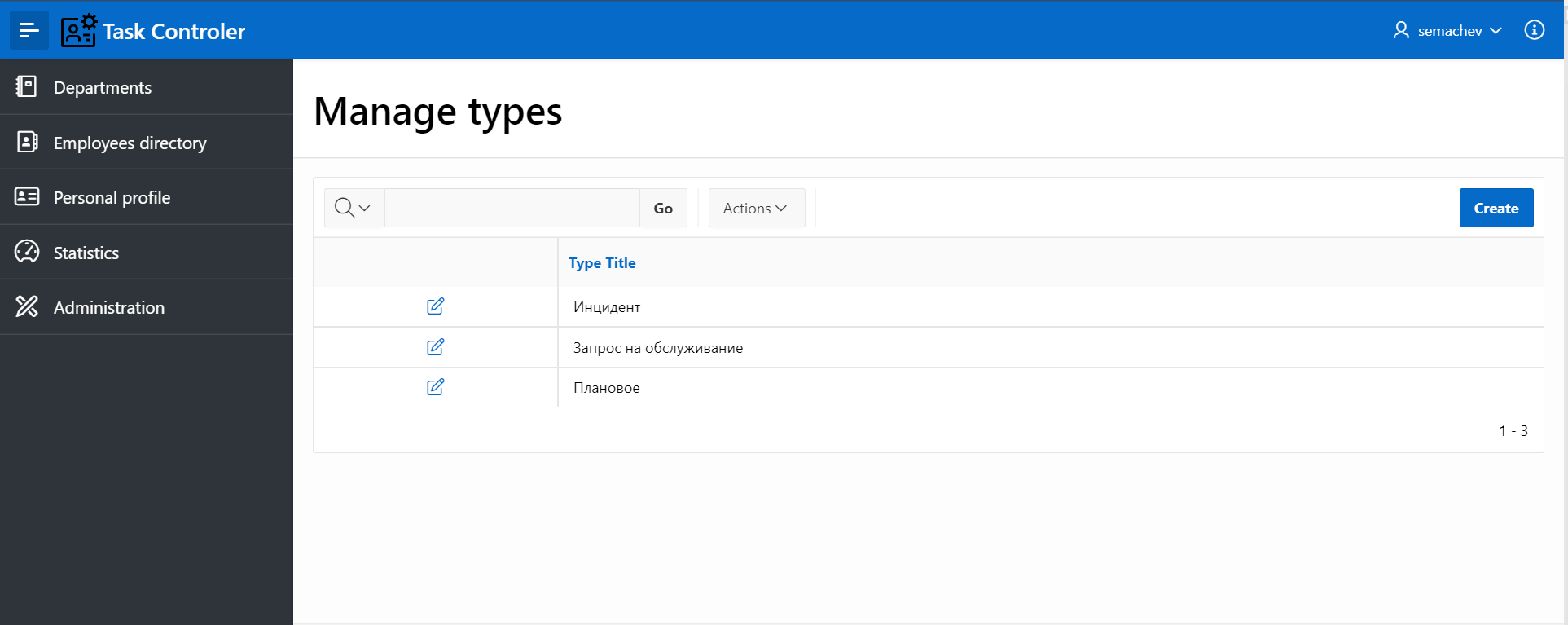


Рисунок 17.Управление ролями

Данная страница отображает список всех ролей в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

## Управление типами заявок

*Рисунок 18. Управление типами заявок*

Данная страница отображает список всех ролей в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

## Управление приоритетами заявок

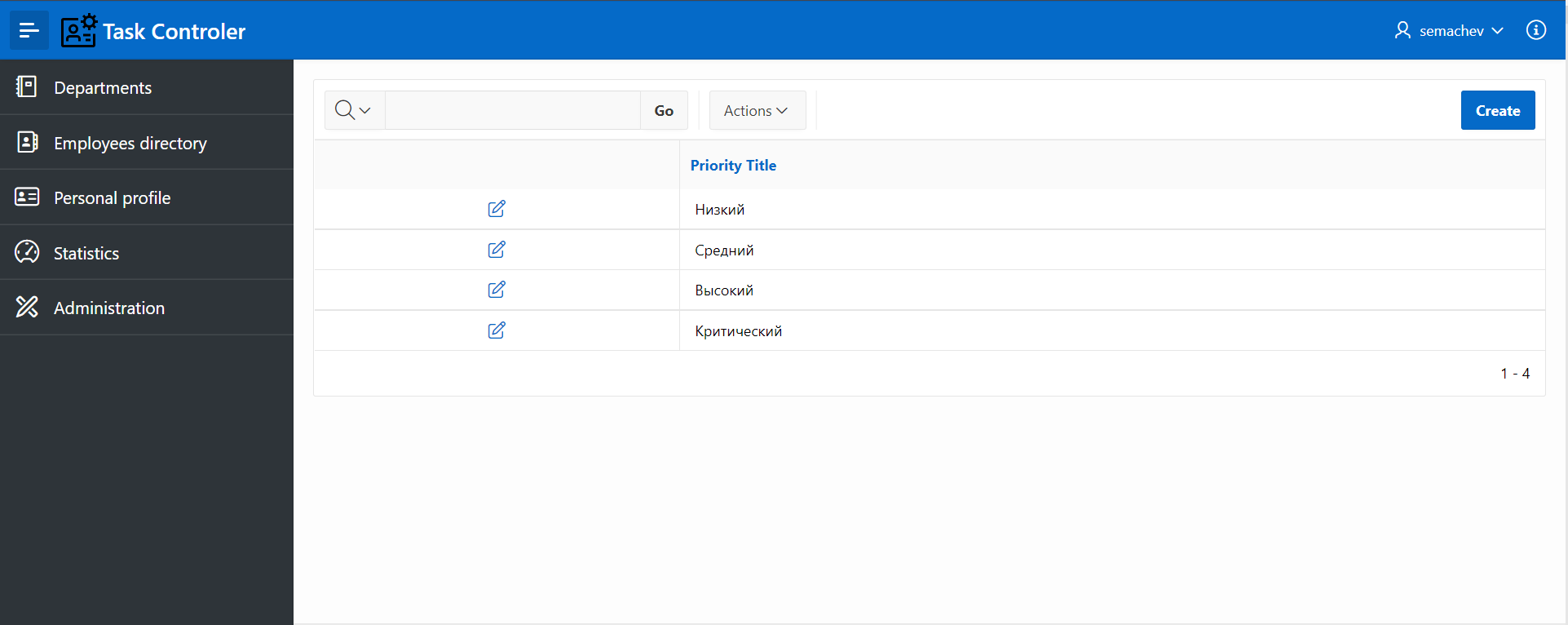


Рисунок 19. Управление приоритетами заявок

Данная страница отображает список всех приоритетов в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

## Управление статусами заявок

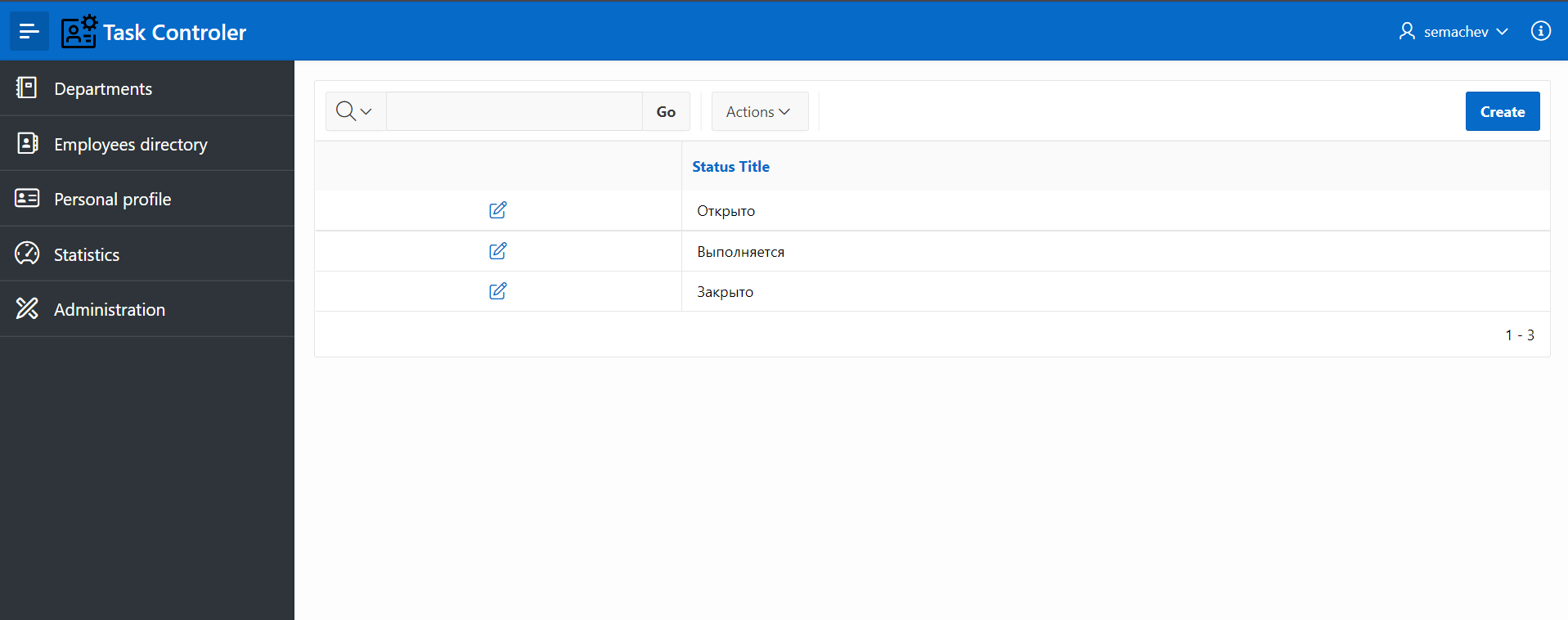


Рисунок 20. Управление статусами заявок

Данная страница отображает список всех статусов в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

## Управление департаментами

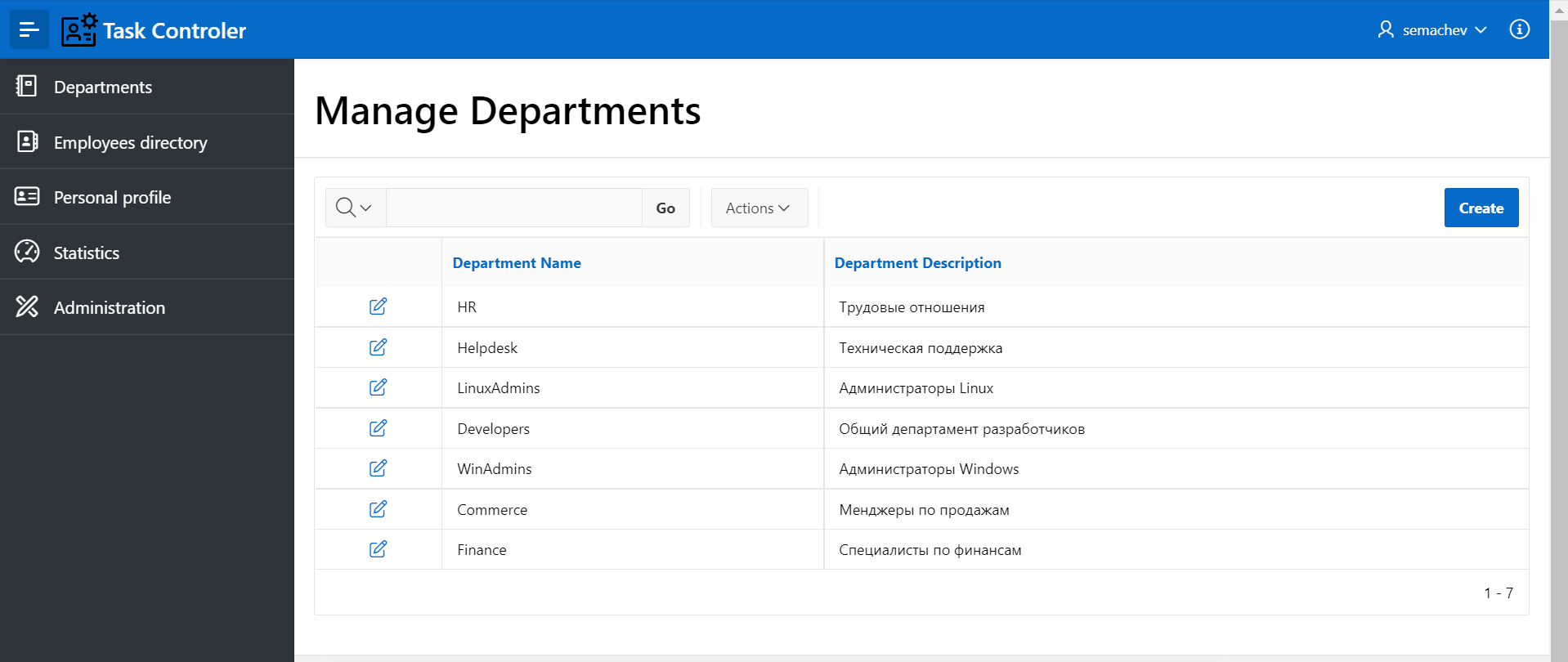


Рисунок 21. Управление департаментами

Данная страница отображает список всех департаментов в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

## Управление фотографиями сотрудников

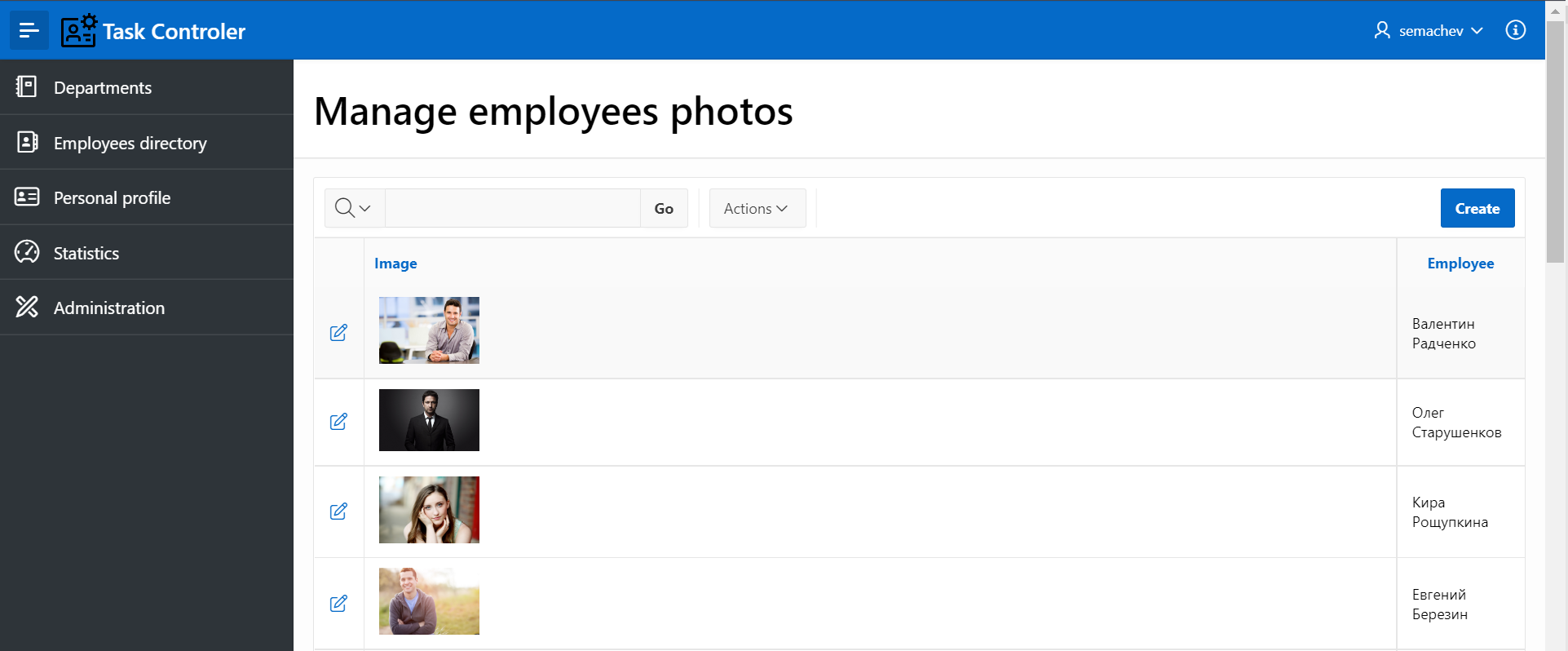


Рисунок 22. Управление фотографиями сотрудников

Данная страница отображает фотографии всех сотрудников, зарегистрированных в системе. Страница сделана с помощью Interactive Report. Сверху есть кнопка создания нового пользователя. При нажатии на значок карандаша можно отредактировать текущие роли.

# Глава 7. Внутренние настройки

В ходе разработки приложения были использованы следующие инструменты и приемы:

## Buttons

Используется для перехода на другие страницы приложения, а также для подтверждения действий на странице.

Примером перехода на другую страницу является кнопка «Classic view» на странице «Задачи департамента \ faceted view», которая осуществляет переход на экранную форму «Задачи департамента \ classic view». [Описание страницы тут.](#_Просмотр_очереди_задач)

Примером подтверждения действия является кнопка «Создать» на экранной форме «Создать задачу». [Описание страницы тут.](#_Запись_на_Тест-Драйв)

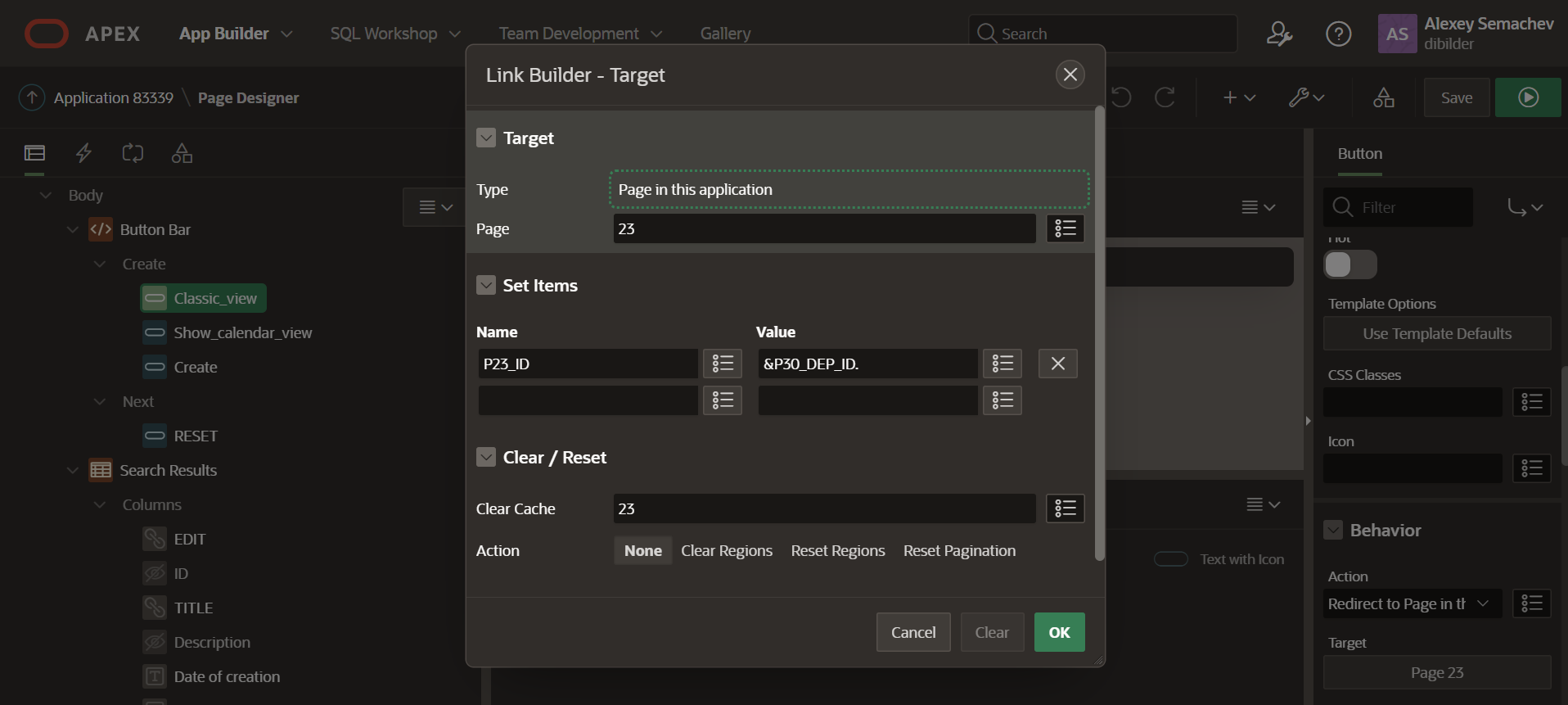


Рисунок 19. Buttons

## Calendar

Используется на странице «Задачи департамента \ Calendar view», создает большой функциональный календарь, с которым можно взаимодействовать. Примером является страница, приведенная выше.

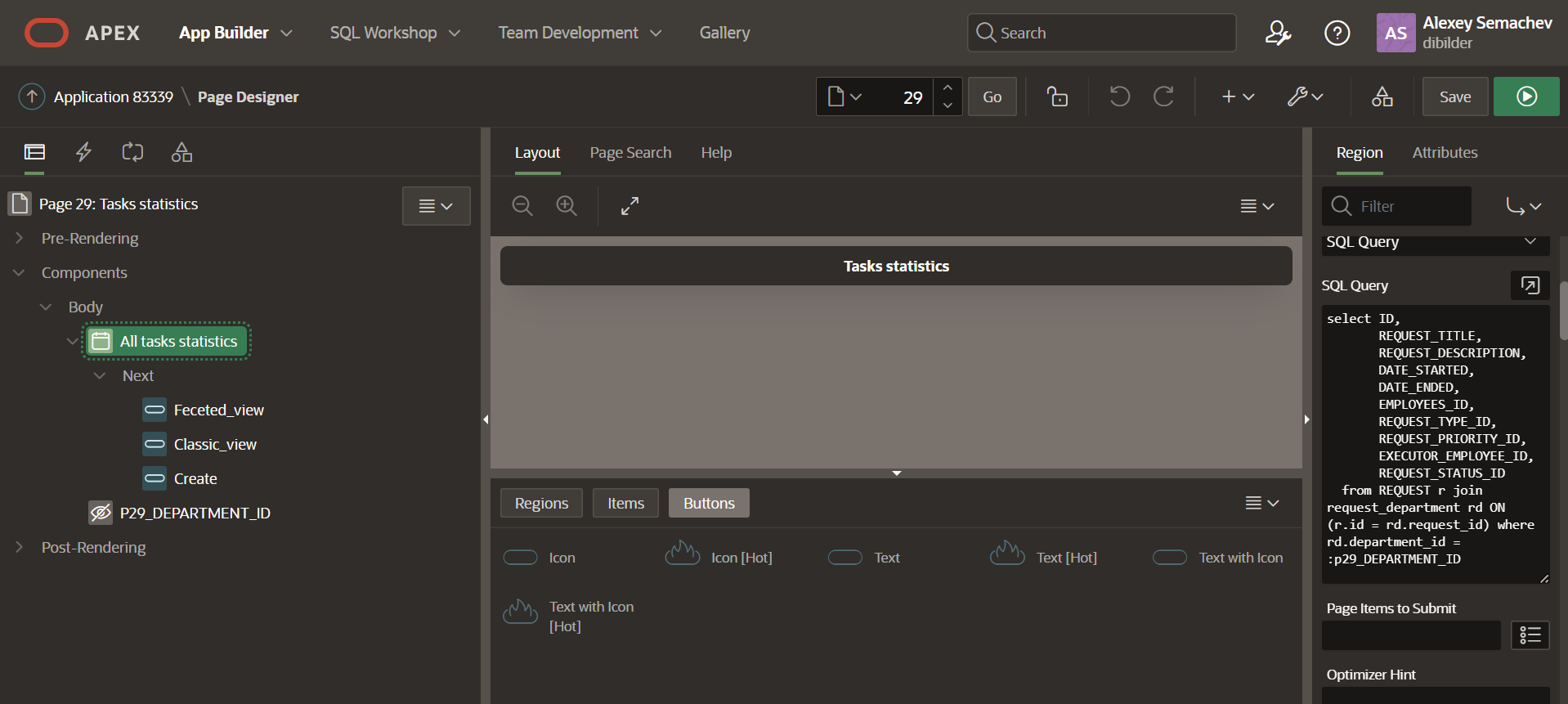
****

Рисунок 20. Calendar

## Check Box

Применяется в фильтрах на страницах с отчетами. Позволяет создавать удобные фильтры для быстрого взаимодействия с отчетом. Примером использования данного инструмента является фильтр «Автор» на странице «Задачи департамента \ classic view». [Описание страницы тут.](#_Просмотр_очереди_задач)

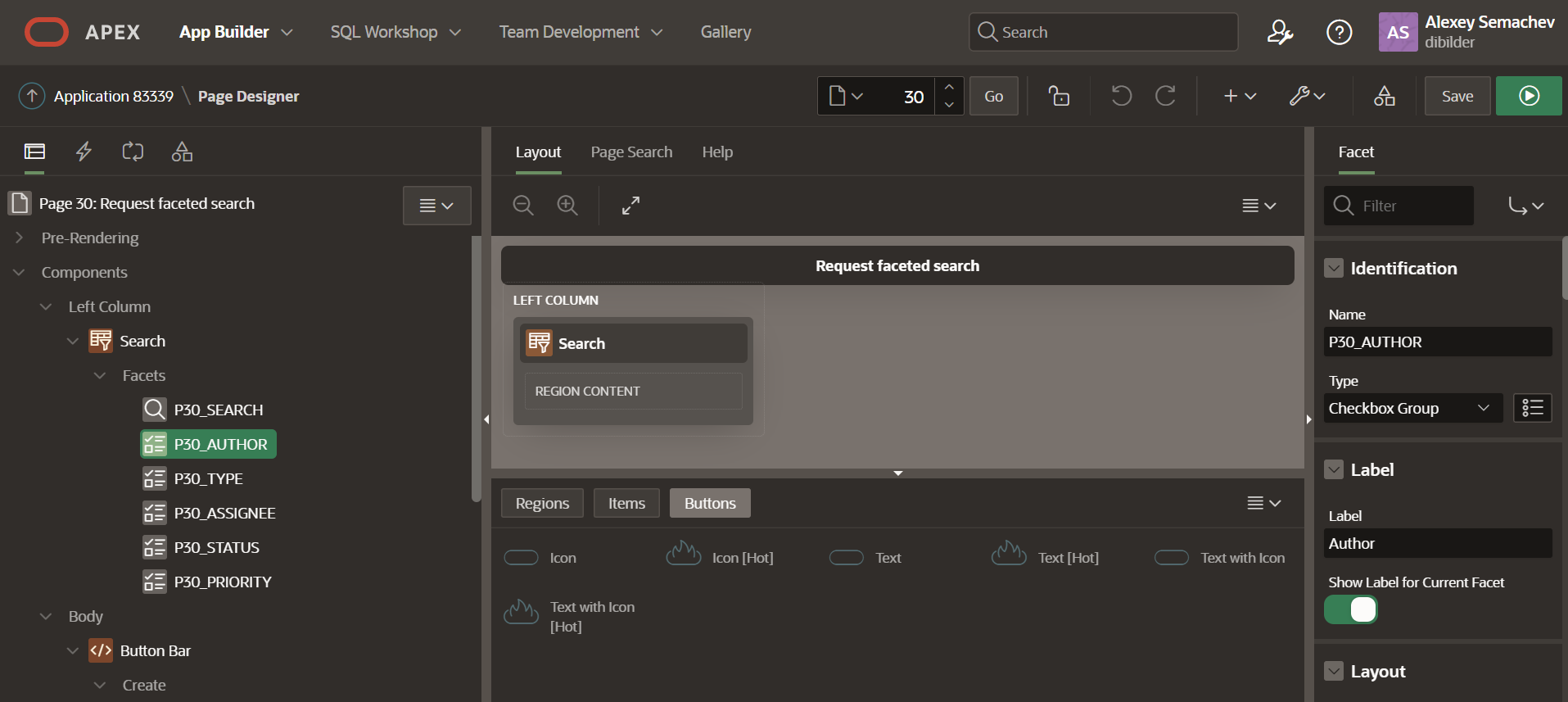


Рисунок 21. Check Box

## Classic Report

Применяется в странице «Информация о сотруднике». Выставлены особые настройки внешнего вида. [Описание страницы тут.](#_Информация_о_сотруднике)

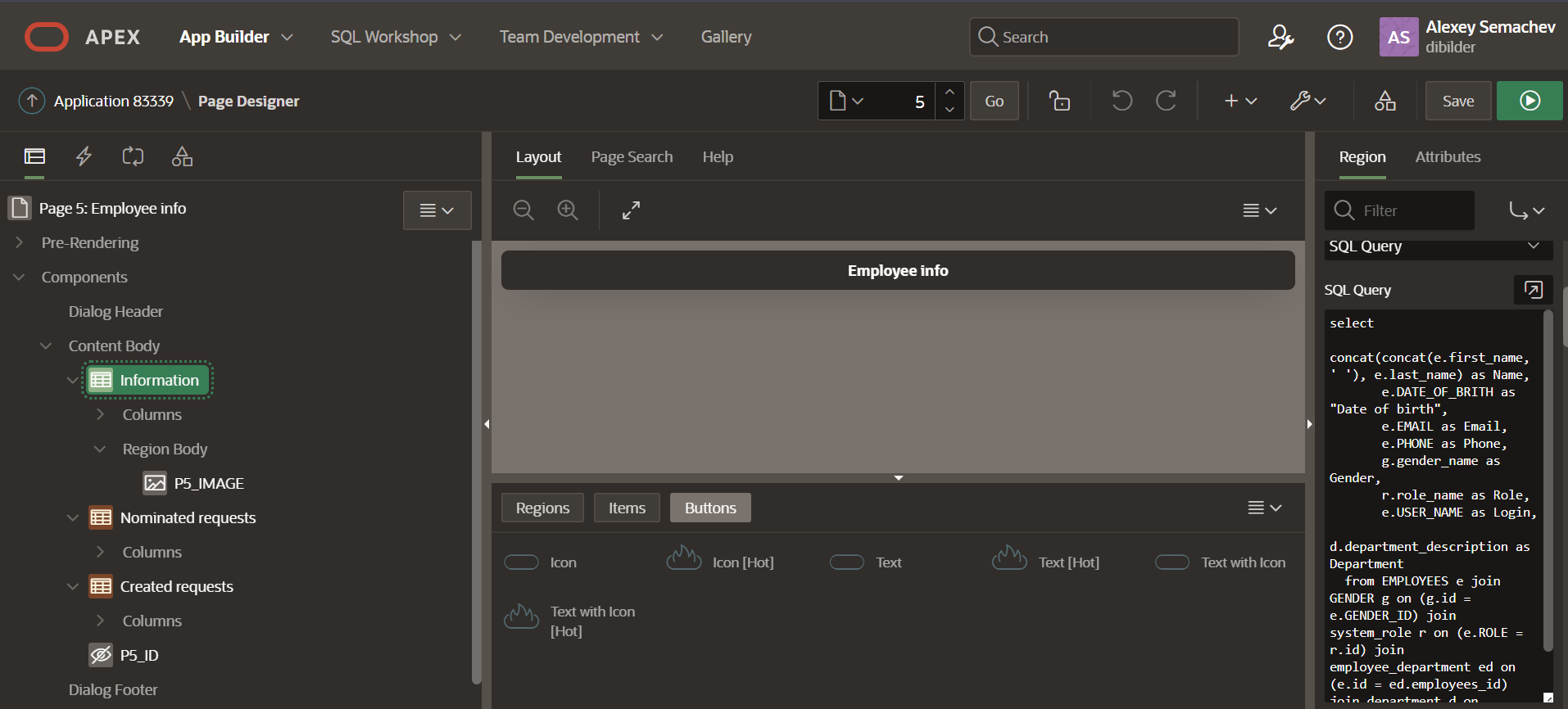


Рисунок 22. Classic Report

## Date Picker

Применяется для удобной работы с датами в отчетах. Примером использования данного инструмента является экранная форма «Создание задачи». На данной экранной форме присутствуют пример использования данного инструмента, который обозначает срок выполнения задачи. [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

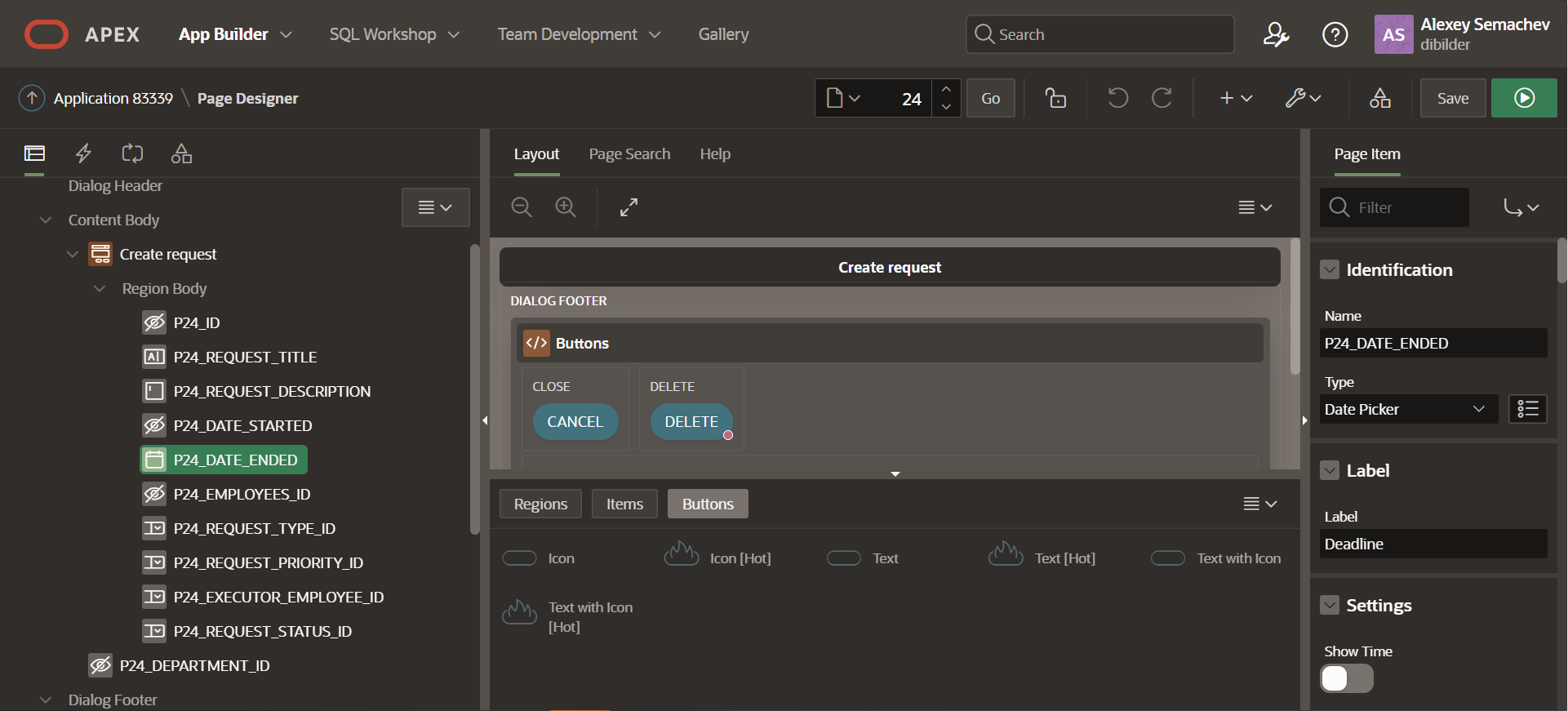


Рисунок 23. Date Picker

## Dynamic Action

Применяется для множества целей: обновление региона, изменения значения элемента и т.д. Примером использования является страница «Задачи департамента \ faceted view». [Описание страницы тут.](#_Просмотр_очереди_задач)

При закрытии какого-либо модального окна, отчет обновляется.

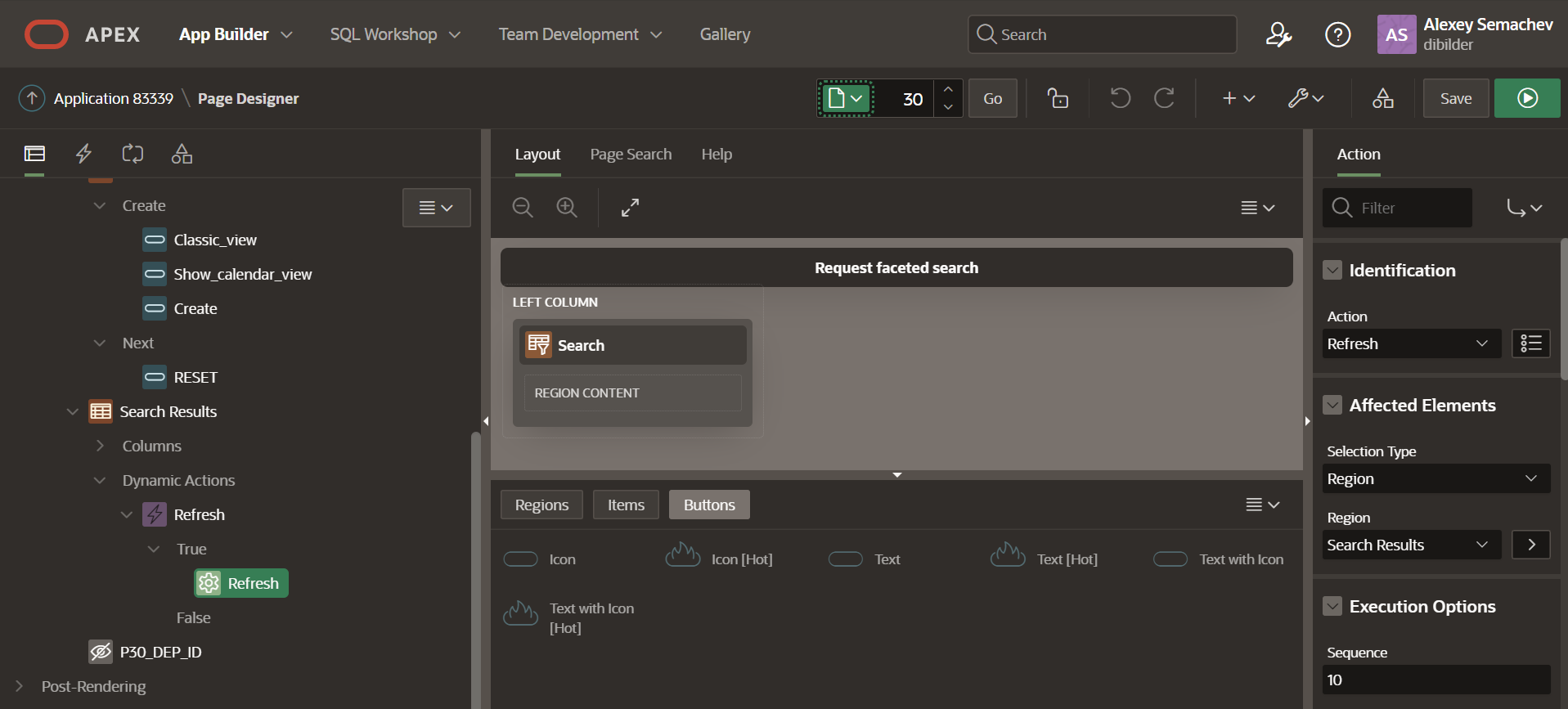


Рисунок 24. Dynamic Action

## Form

Применяется на модальных страницах для добавления (редактирования и удаления в некоторых случаях) записей в информационную систему. Примером использования служит форма создания задачи или создания комментария. [Подробная информация о задаче](#_Подробная_информация_о).

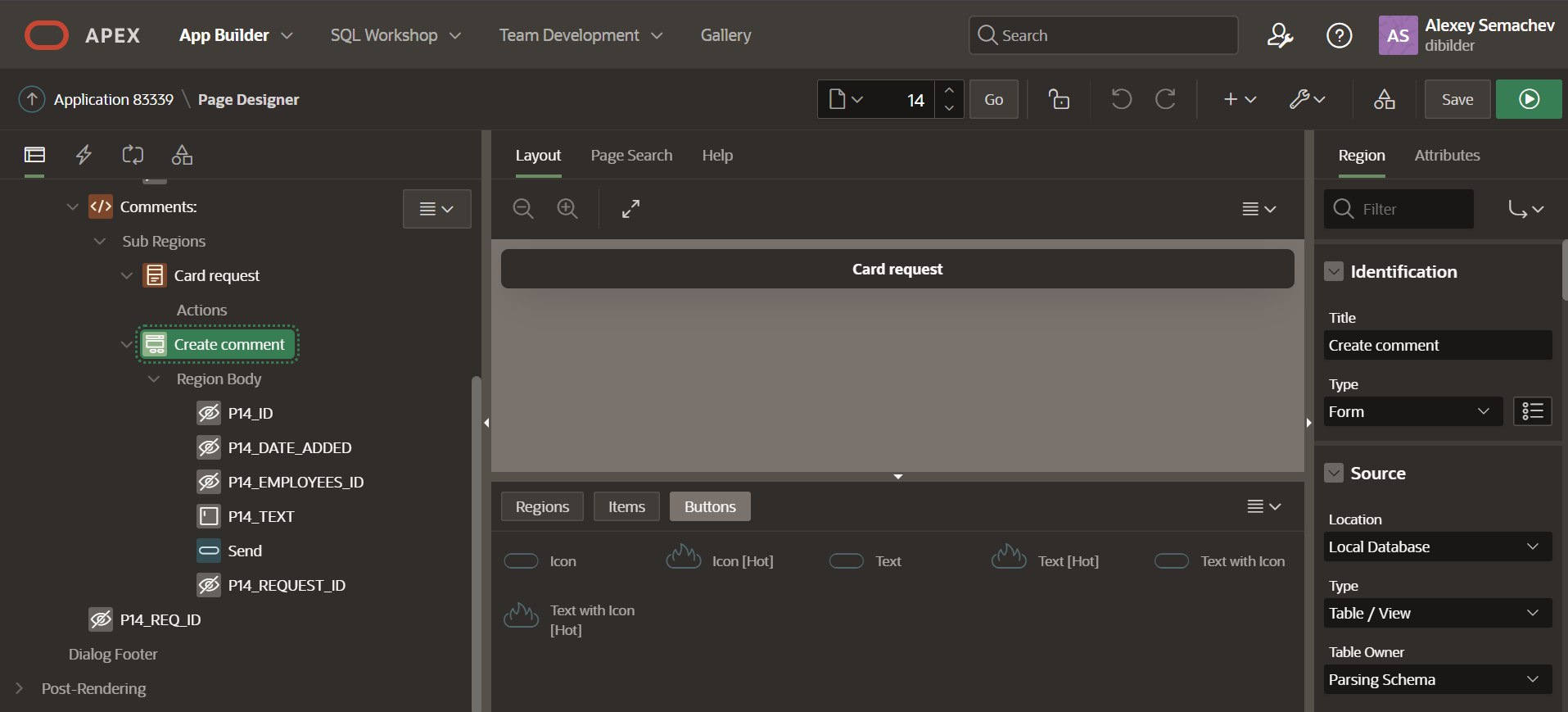


Рисунок 25. Form

## Interactive Report

Применяется для многих экранных форм для создания отчетов. Применяется в странице «Задачи департамента \ classic view». Выставлены особые настройки внешнего вида. [Описание страницы тут.](#_Просмотр_очереди_задач_1)

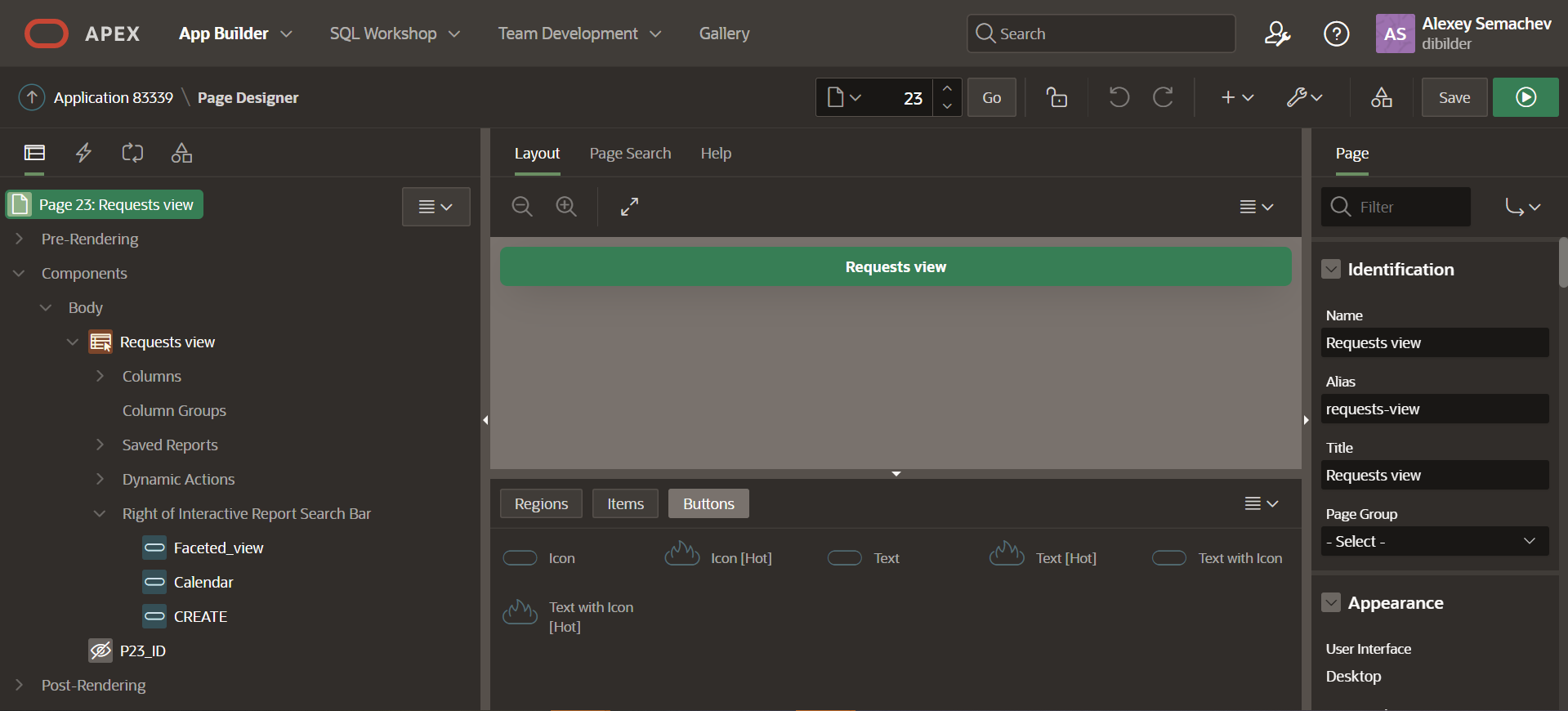


Рисунок 26. Interactive Report

## Link

Применяется в интерактивных отчетах для создания различных ссылок.

Примером использования является поле «Title» на экранной форме «Задачи департамента \ classic view». В данном случае «Заголовок» является гиперссылкой на карточку задачи. [Описание страницы тут.](#_Просмотр_очереди_задач_1)

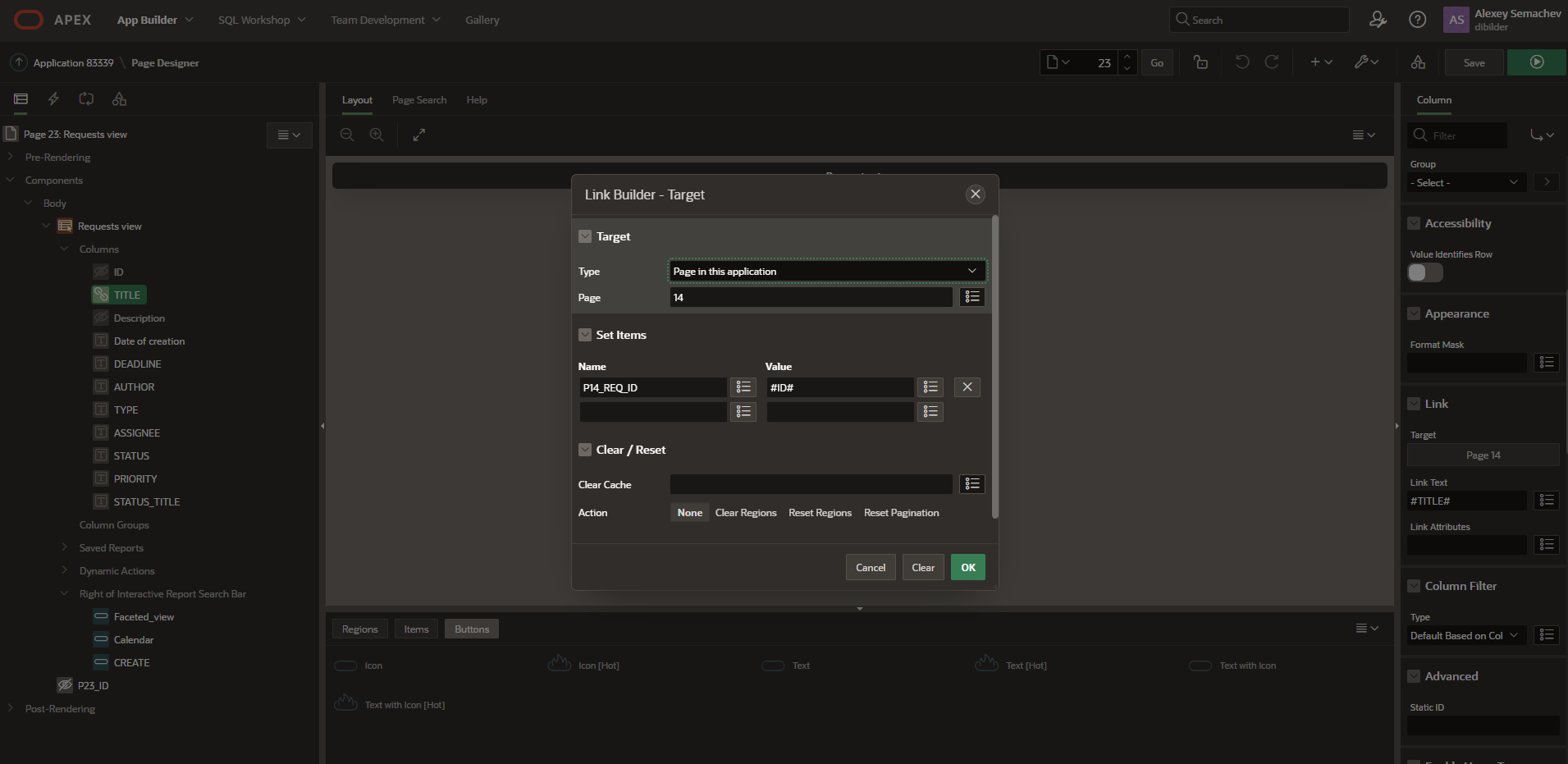


Рисунок 27. Link

## List of Values

Применяется для полей типа «Select List» для отображения значений. Примером использования данного инструмента является тип задачи в экранной форме «Создание задачи». [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

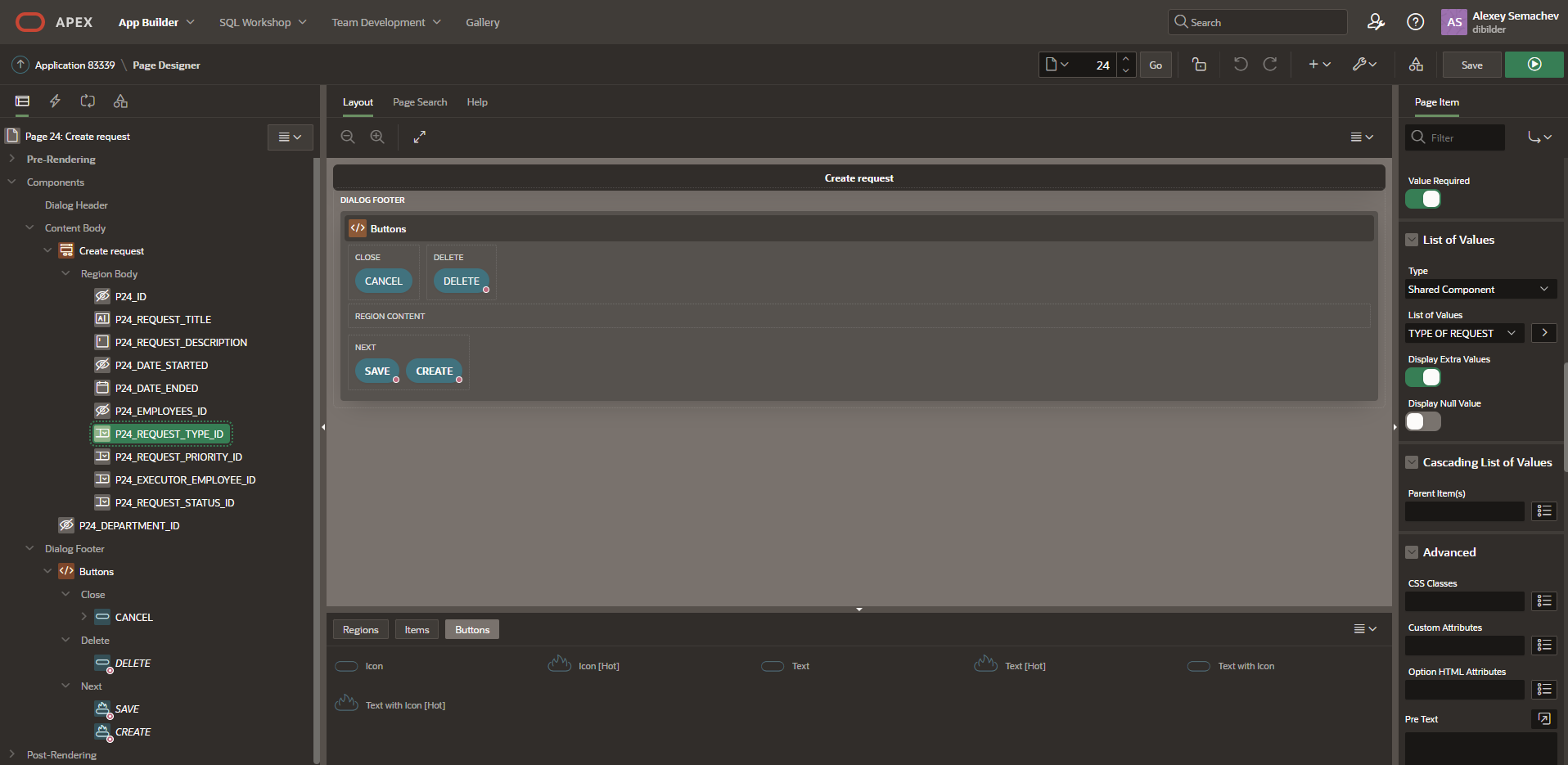


Рисунок 28. List of Values

## List

Применяется для страниц, где необходимо сделать множество гиперссылок на другие страницы. Примером использования является «Администрирования», меню на которой ведет на страницы для управления базой данных приложения. [Описание страницы тут.](#_Администрирование)

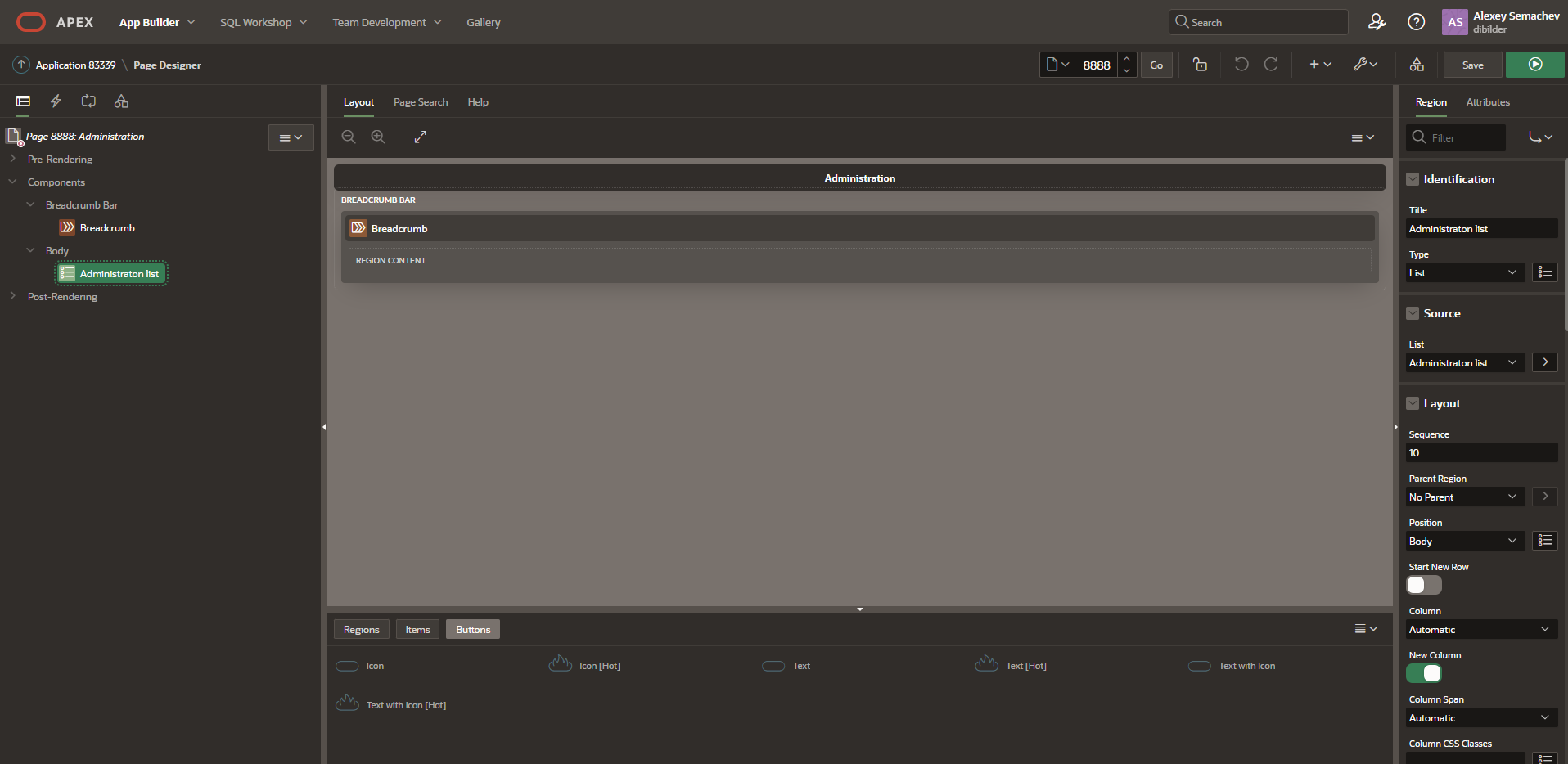


Рисунок 29. List

## Page Item

Применяется для множества целей: отображение значения, скрытая переменная для подсчета, фильтрации и т.д. Примером использования являются скрытый элемент для хранения ID департамента на странице «Создание задачи». [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

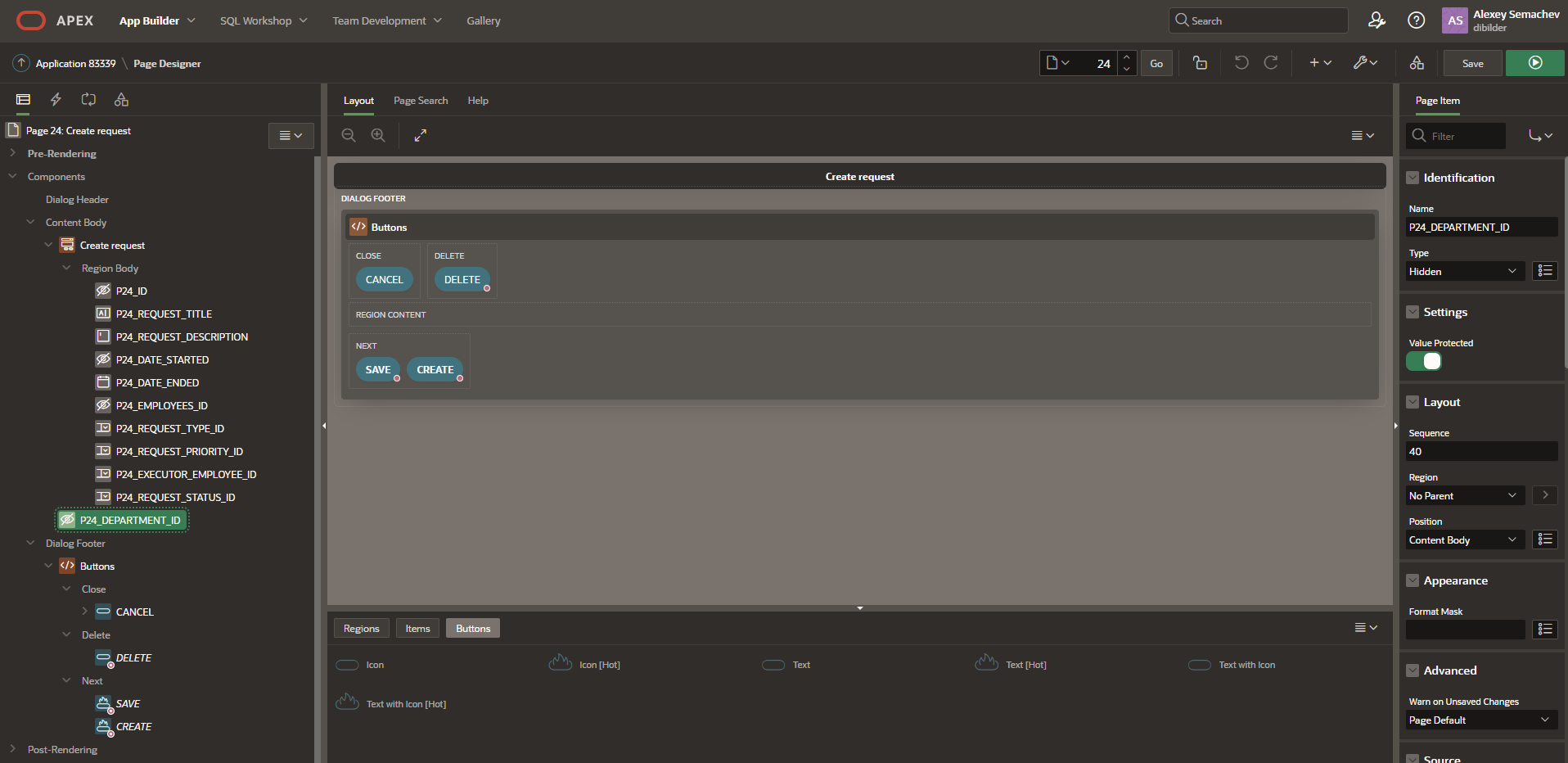


Рисунок 30. Page Item

## Security

Применяется для разграничения доступа некоторых функций с помощью прав. Одним из примеров использования является кнопка «Сохранить» на экранной форме «Создание\Редактирование задачи». Данная кнопка доступна только исполнителям и администраторам, так как редактировать заявки. [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

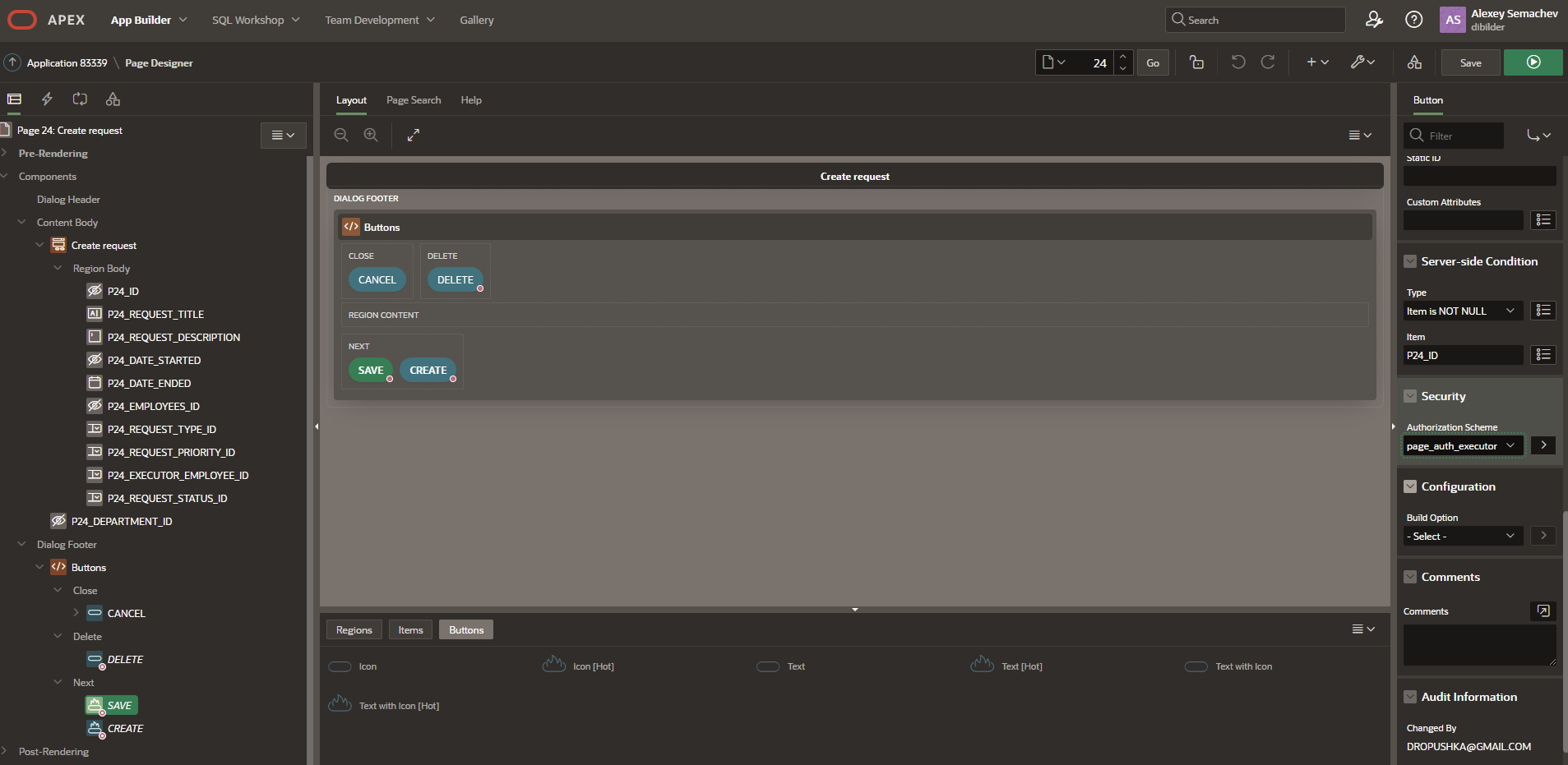


Рисунок 31. Security

## Select List

Применяется для отображения списка доступных значений. Примером использования служит выбор исполнителя на странице «Создание\Редактирование задачи». С помощью SQL запроса составлен список значений. [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

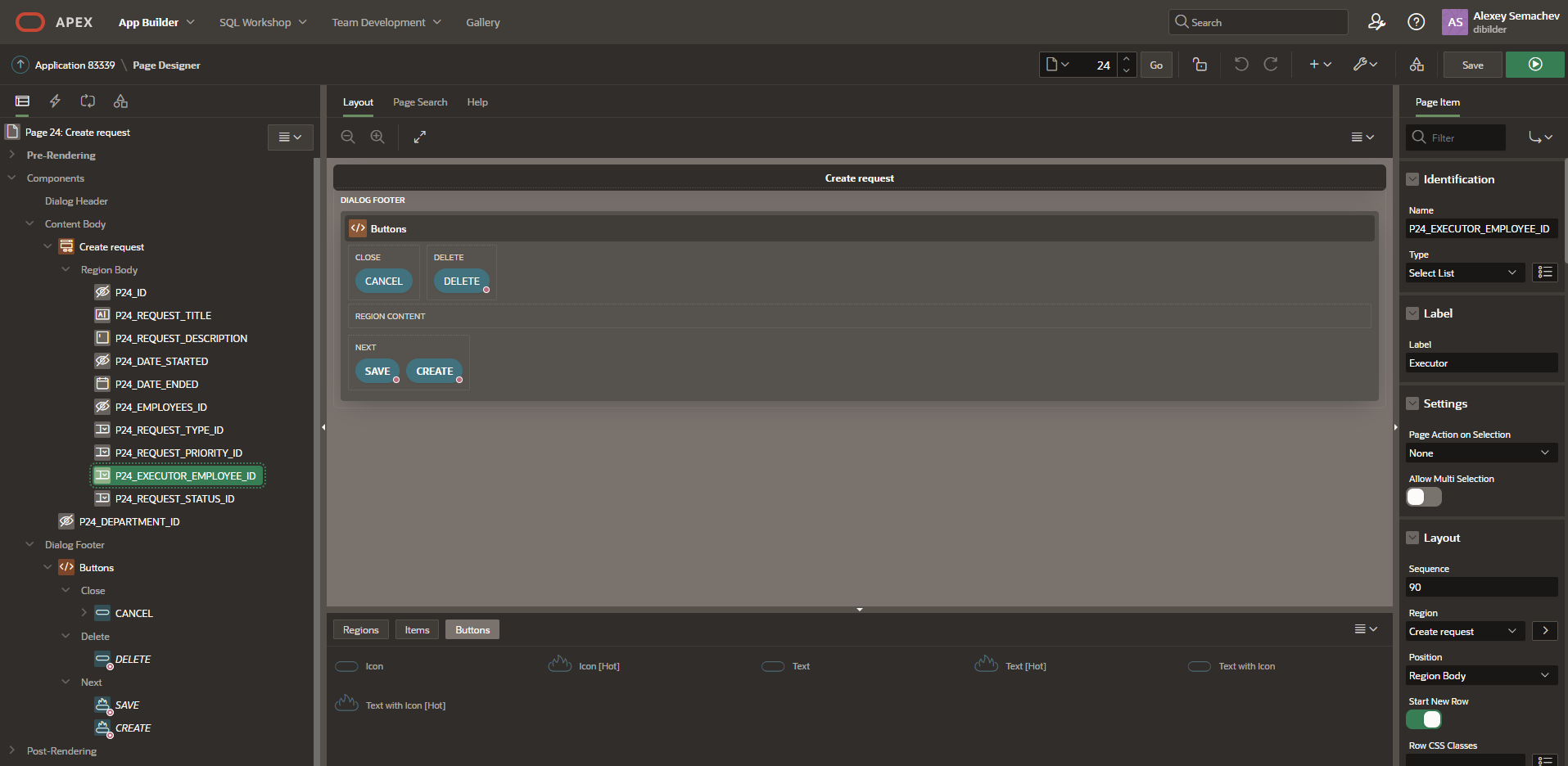


Рисунок 32. Select List

## Static Content

Применяется во множестве экранных форм для разных целей: хранение справочной информации, хранение page items и другие. Примером использования данного инструмента является регион «Comments» на экранной форме «Информация о задаче». [Ссылка на страницу тут.](#_Тест-Драйв)

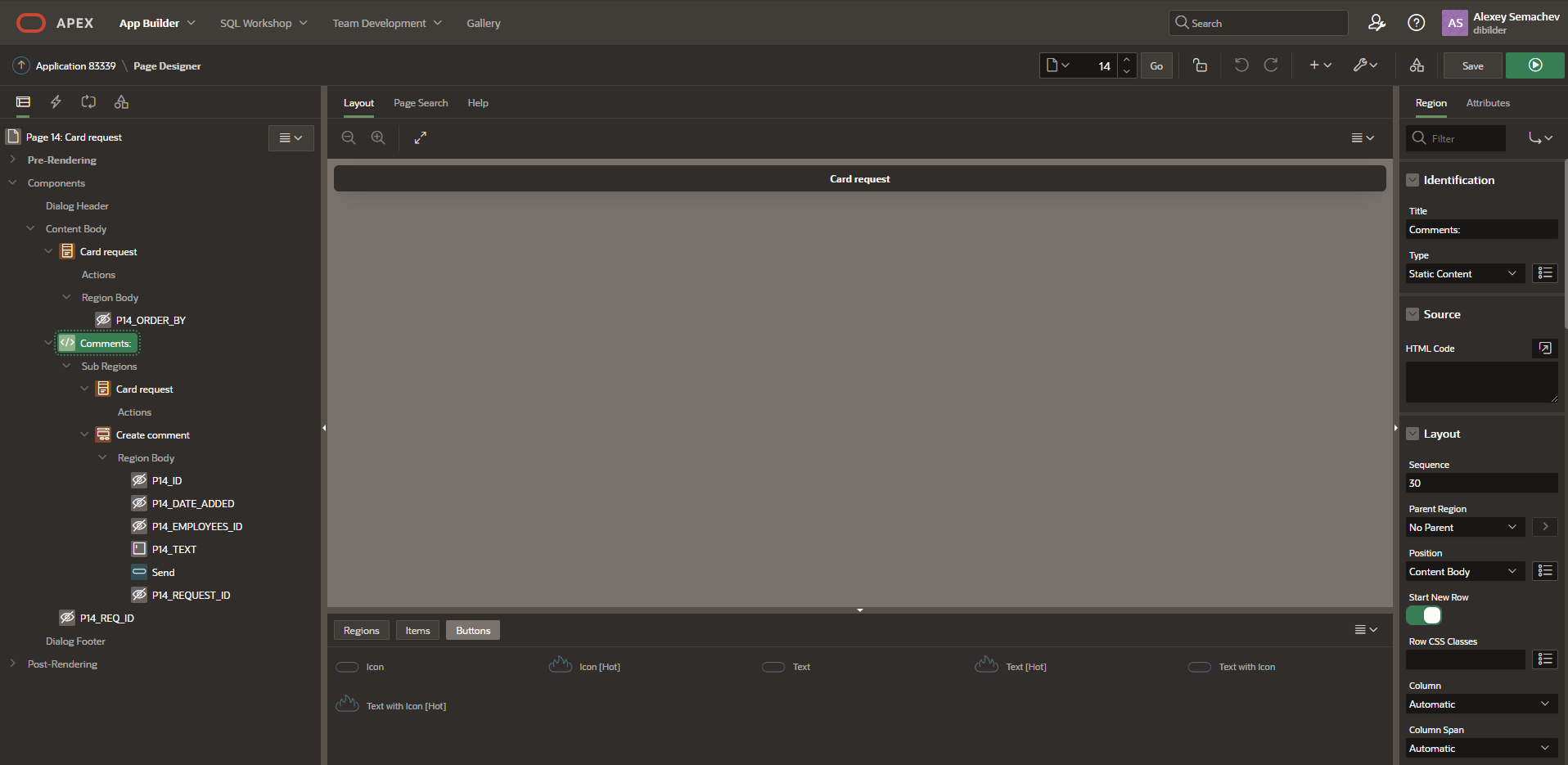


Рисунок 33. Static Content

## Charts

Визуализированное представление какого-либо запроса. На выбор есть гистограммы, диаграммы и т.д. Используется для красивого отображения каких-либо показателей. Примером является страница статистики, состоящая из 6 Charts. [Ссылка на страницу тут.](#_Статистика)

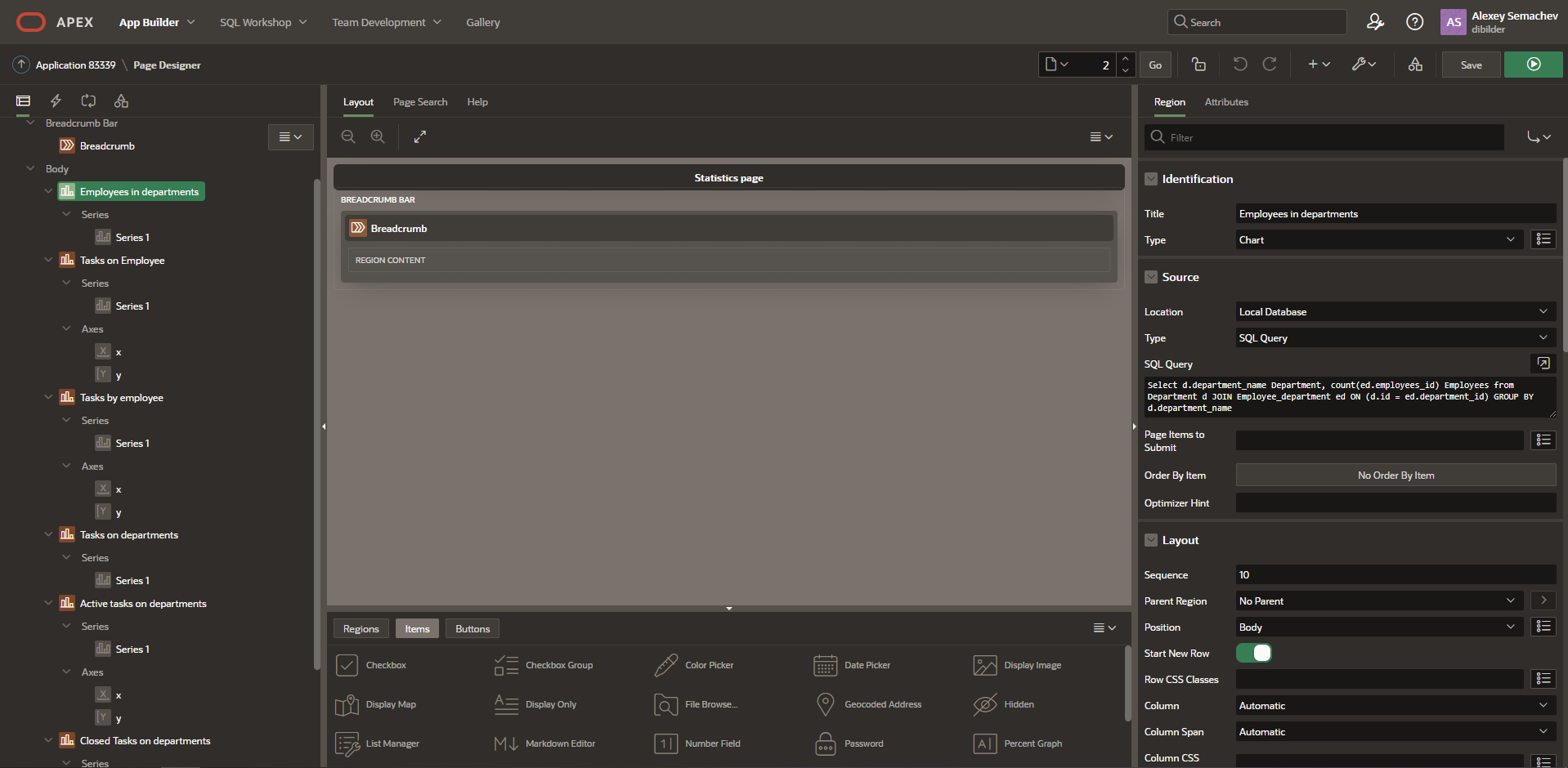


Рисунок 34. Charts

## Authorization scheme

Схема авторизации. Используется для разграничений прав доступа к страницам по каким-то заданным условием. Схема создается через Shared Components. Для ее работы нужно создать PL\SQL функцию, на выходе возвращающее булевое значение. В данном случае, функция получает логин текущего пользователя, и возвращает TRUE, если пользователь имеет роль «Администратор». Аналогичная функция есть и для роли «Исполнитель».

Код PL\SQL функции:

create or replace function PAGE\_AUTH (p\_user in varchar2)

return boolean

as

l\_cnt number;

begin

SELECT role into l\_cnt from employees where user\_name = p\_user;

if l\_cnt = 1 then

return TRUE;

else return FALSE;

end if;

end PAGE\_AUTH;

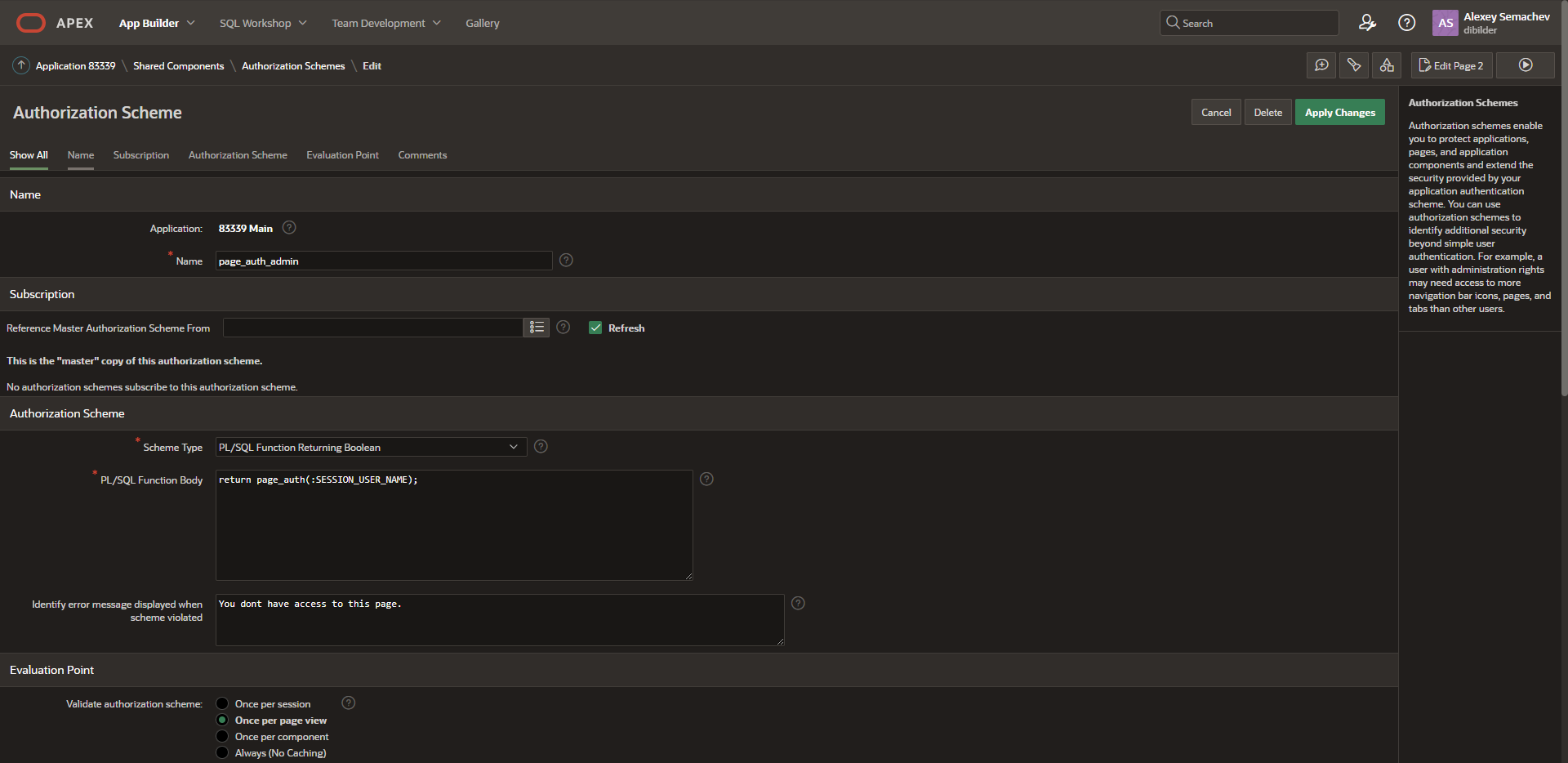


Рисунок 35. Authorization scheme

## Login

Метод аутентификации на странице Login. Используется для собственного способа аутентификации. Например, не по учетным записям Apex, а по учетным записям из внутренней таблицы. Вариантов реализации много, но в данном случае создается собственный Package и редактируется процесс Login для использования этого Package.

Код PL\SQL процесса Login:  
Pkg\_Security.Process\_Login(:P9999\_USERNAME,:P9999\_PASSWORD,:APP\_ID);

Код PL\SQL создания Package:

create or replace Package Body Pkg\_Security Is

Function Authenticate\_User(p\_User\_Name Varchar2

,p\_Password Varchar2) Return Boolean As

v\_Password employees.Password%Type;

v\_Active employees.Active%Type;

v\_Email employees.Email%Type;

Begin

If p\_User\_Name Is Null Or p\_Password Is Null Then

-- Write to Session, Notification must enter a username and password

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Please enter Username and password.');

Return False;

End If;

----

Begin

Select u.Active

,u.Password

,u.Email

Into v\_Active

,v\_Password

,v\_Email

From employees u

Where u.User\_Name = p\_User\_Name;

Exception

When No\_Data\_Found Then

-- Write to Session, User not found.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User not found');

Return False;

End;

If v\_Password <> p\_Password Then

-- Write to Session, Password incorrect.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Password incorrect');

Return False;

End If;

If v\_Active <> 'Y' Then

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User locked, please contact admin');

Return False;

End If;

---

-- Write user information to Session.

--

Apex\_Util.Set\_Session\_State('SESSION\_USER\_NAME'

,p\_User\_Name);

Apex\_Util.Set\_Session\_State('SESSION\_EMAIL'

,v\_Email);

---

---

Return True;

End;

--------------------------------------

Procedure Process\_Login(p\_User\_Name Varchar2

,p\_Password Varchar2

,p\_App\_Id Number) As

v\_Result Boolean := False;

Begin

v\_Result := Authenticate\_User(p\_User\_Name

,p\_Password);

If v\_Result = True Then

-- Redirect to Page 1 (Home Page).

Wwv\_Flow\_Custom\_Auth\_Std.Post\_Login(p\_User\_Name -- p\_User\_Name

,p\_Password -- p\_Password

,v('APP\_SESSION') -- p\_Session\_Id

,p\_App\_Id || ':1' -- p\_Flow\_page

);

Else

-- Login Failure, redirect to page 101 (Login Page).

Owa\_Util.Redirect\_Url('f?p=&APP\_ID.:101:&SESSION.');

End If;

End;

End Pkg\_Security;

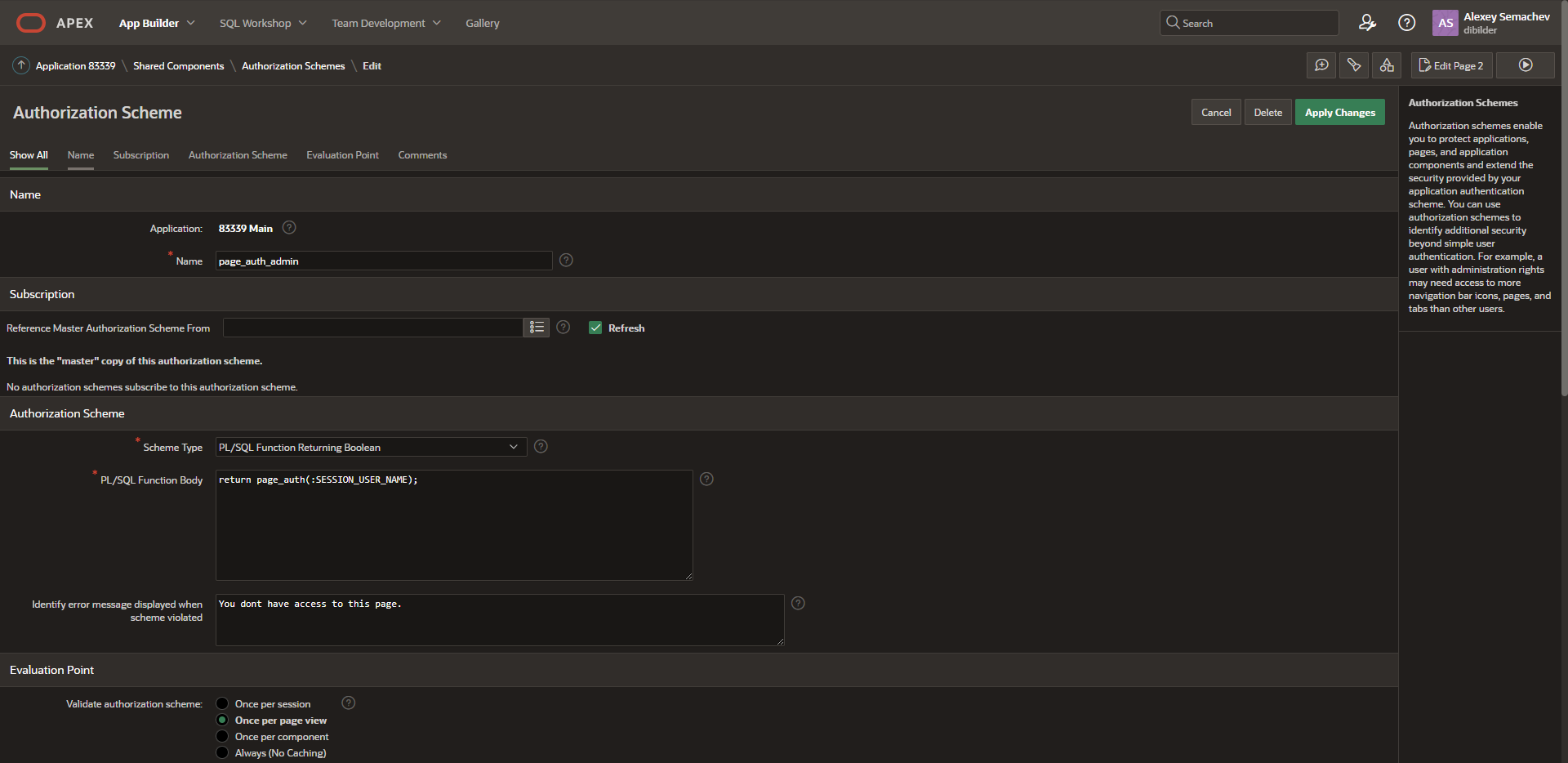


Рисунок 36. Login

## Processes

Процессы используются для выполнения каких-либо действий, как правило, во время события Submit. С помощью процессов чаще всего производят Insert\Update в базу данных. Примером может быть дополнительный процесс «INSERT INTO Request\_Department» на странице «Создание\Редактирование задачи». [Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

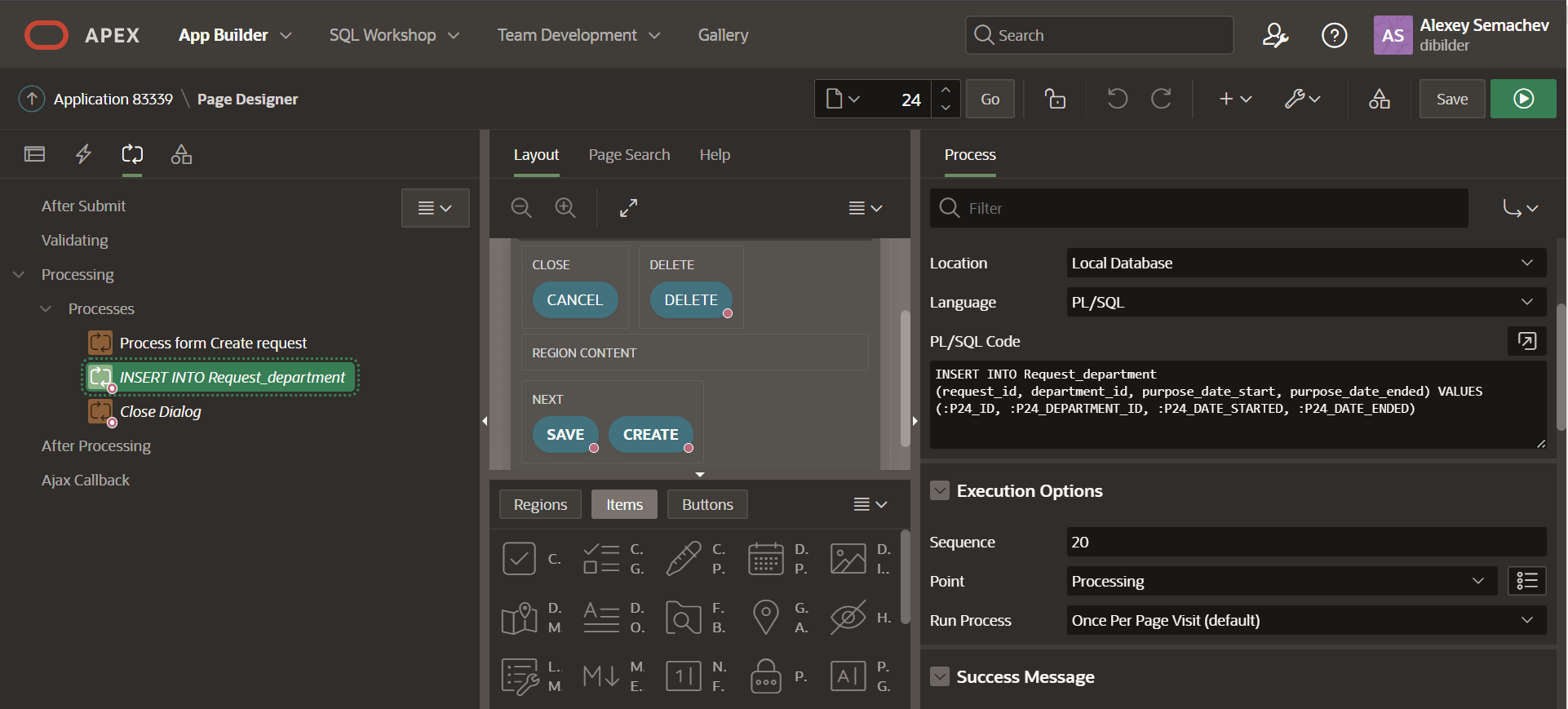


Рисунок 37. Processes

## Server-side Condition

Атрибут предназначен для назначения конкретного действие на выполнение процесса. Например, процесс, описанный выше, на странице «Создание\Редактирование задачи», выполняется не всегда, а только при нажатии кнопки CREATE. При редактировании существующей заявки никаких дополнительных INSERT-ов не нужно.  
[Ссылка на страницу тут.](#_Запись_на_Тест-Драйв)

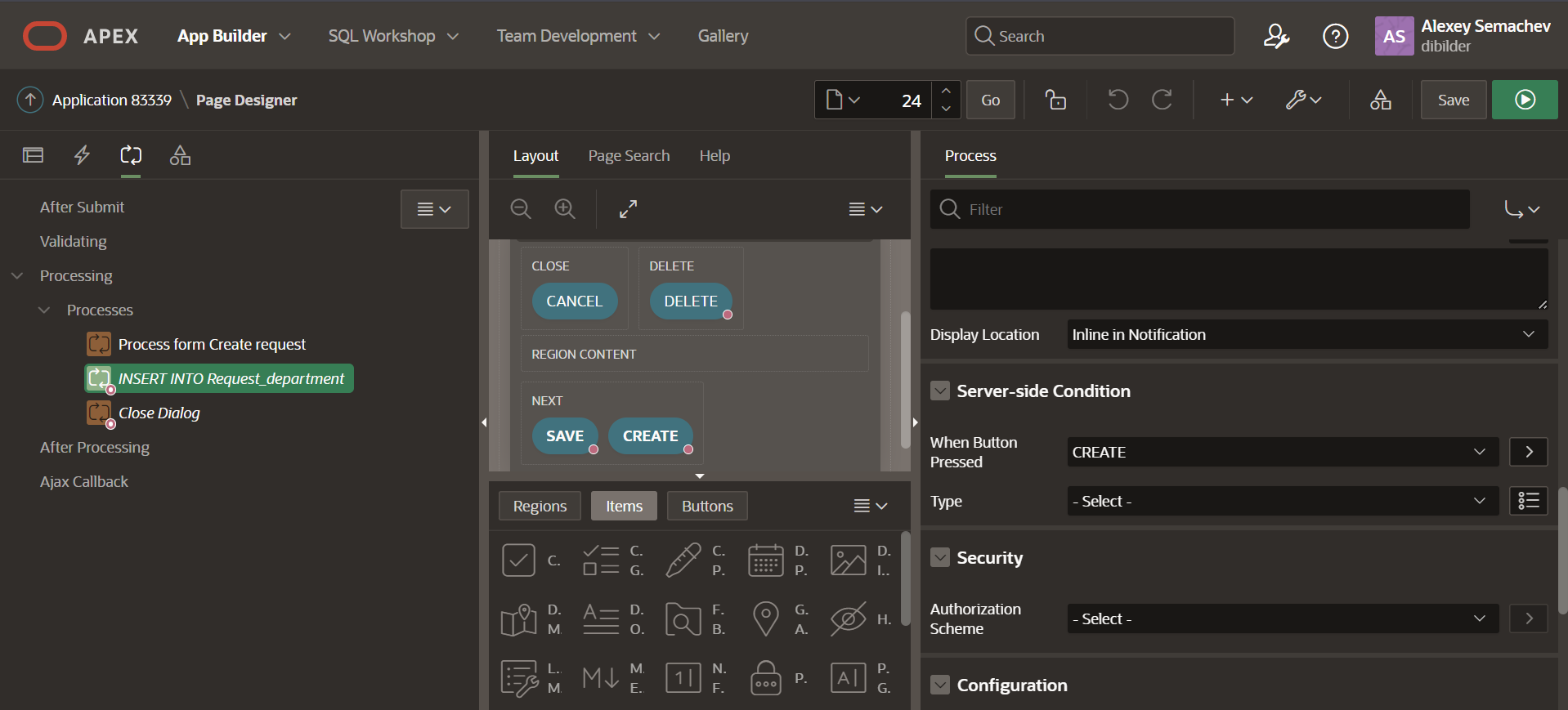


Рисунок 38. Server-side Condition

## Application items

Некоторые «глобальные переменные» приложения, доступные из любой страницы. Добавляются через Shared Components. В качестве примера – используются для работы собственного метода аутентификации на странице Login. Эти же Items позволяют в дальнейшем получать логин текущего пользователя в любом необходимом месте.

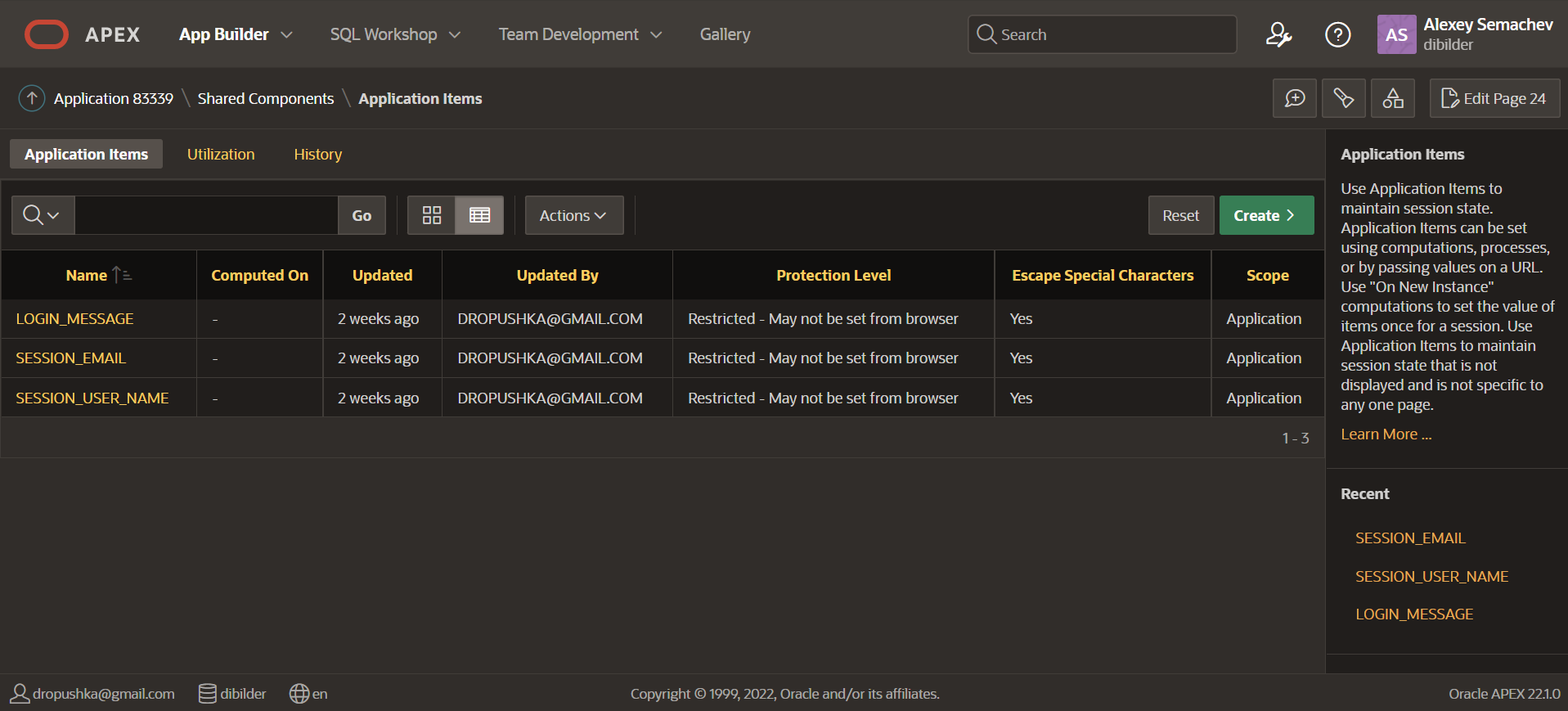


Рисунок 39. Application items