**451F**

**Software Architecture Document**

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 1/05/14 | 1.0 | Added introduction and architectural goals | Sergey Koyushev |
| 3/05/14 | 1.0 | Added class diagram | Sergey Koyushev |
| 9/06/14 | 1.0 | Added activity and Sequence diagram | Sergey Koyushev |
| 11/06/14 | 1.0 | State diagrams | Sergey Koyushev |
| 13/06/14 | 2.0 | Edited logic view | Sergey Koyushev |
| 14/06/14 | 2.1 | Added impl view | Sergey Koyushev |
| 15/06/14 | 2.2 | Fixed db diagram | Sergey Koyushev |

**Software Architecture Document**

#### **1.** **Introduction**

##### **1.1** **Purpose**

Данный документ предназначен для разработчиков системы. В этом документе описывается архитектура системы, рассмотренная с различных точек зрения. Документ содержит архитектурные решения, которые были сделаны в системе.

**1.2** **Scope**

Данный документ является частью документации проекта “451F”

##### **1.3** **Definitions, Acronyms and Abbreviations**

См. Glossary[1].

##### **1.4** **References**

1) Glossary;

2) SRS;

##### **1.5** **Overview**

В этом документе описывается архитектура системы, рассмотренная с различных точек зрения. Архитектура рассматривается на разных уровнях детализации, таких как usu-case, logic, implementation. Документ содержит архитектурные решения, которые были сделаны в системе.

Так же в документе содержится инфологическая модель базы данных, используемая системой, а так же подробное описание все таблиц и полей в них.

#### **2.** **Architectural Representation**

Данный документ описывает архитектуру системы с различных точек зрения - use case view, logical view, process view, deployment view, implementation view. Взаимодействие элементов разных точек зрения представлено в виде UML-диаграмм.

#### **3.** **Architectural Goals and Constraints**

**Основные требования к системе:**

### 3.1 Гражданин

Гражданин должен зарегистрироваться в системе.

Гражданин должен авторизоваться в системе.

Гражданин должен составить «Донос» на «Гражданина» города.

Гражданин может просматривать новостные сводки города.

### 3.2 Офицер полиции

Офицер полиции должен просматривать список «Доносов» от Граждан города и выбирать один из них.

Офицер полиции должен оправлять «Электрического Пса» на проверку потенциального «Книголюба».

Офицер полиции должен составлять рапорт о результате проверки «Электрическим Псом» потенциального «Книголюба».

Офицер полиции должен выписать «Ордера на сожжение» книг.

### 3.3 Начальник пожарного расчета

Начальник пожарного расчета должен выбрать из «Ордеров на сожжение» «Книголюбов» того, который наиболее опасен для города.

Начальник пожарного расчета должен отчитаться о потраченном горючем, на сожжение запрещенных книг.

### 3.4 Чиновник

Чиновник должен финансировать проведение проверки «Электрическим Псом» потенциального «Книголюба» и расходы на горючее, потраченное на сожжение запрещенных книг.

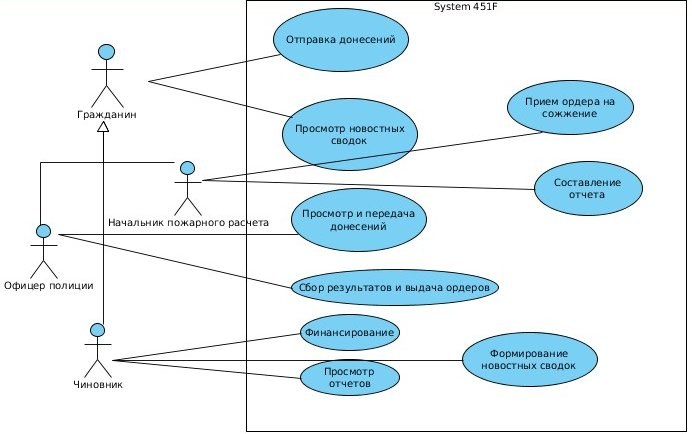
Чиновник может составлять новостные сводки города.

Чиновник может просматривать обстановку в городе.

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

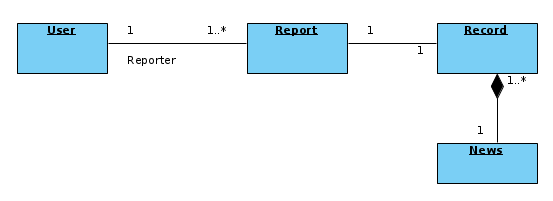
Данный раздел содержит описание основных сценариев использования системы разными типами пользователей. Документ рассматривает следующие сценарии:

**4.1 Use-case diagrama**



**4.2 Class Diagram**

Отображает взаимодействие ключевых сущностей



**4.3 Activity Diagram**

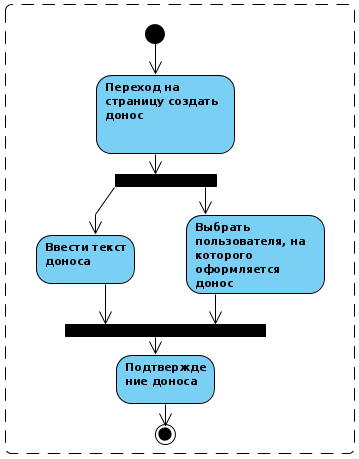
Прецедент: Донос от гражданина

Предусловие:

1. В системе существуют пользователь с ролью «Гражданин».

2. Пользователь с ролью «Гражданин» успешно прошел аутентификацию и авторизацию в системе.

Постусловие: Донос поступает в систему для дальнейшего рассмотрения.



Прецедент: Выбор и исполнение действующих подтвержденных доносов.

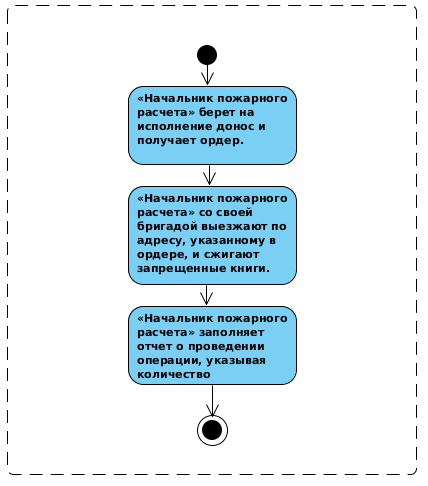
Предусловие:

1. В системе существует пользователь с ролью «Начальник пожарного расчета».

2. Пользователь с ролью «Начальник пожарного расчета», участвующий в прецеденте, успешно прошел аутентификацию и авторизацию в системе.

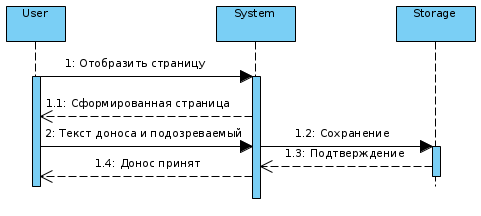
3. В системе имеются один или более действующих ордеров на сожжение книг.

Постусловие: Запрещенные книги сожжены.



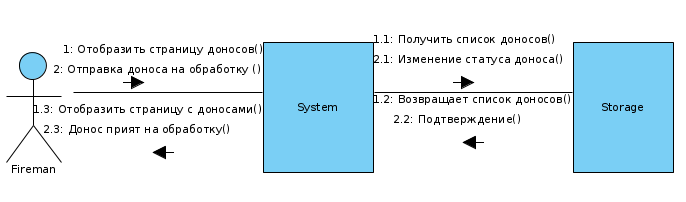
**4.4 Sequence diagram**

Прецедент: Создание доноса



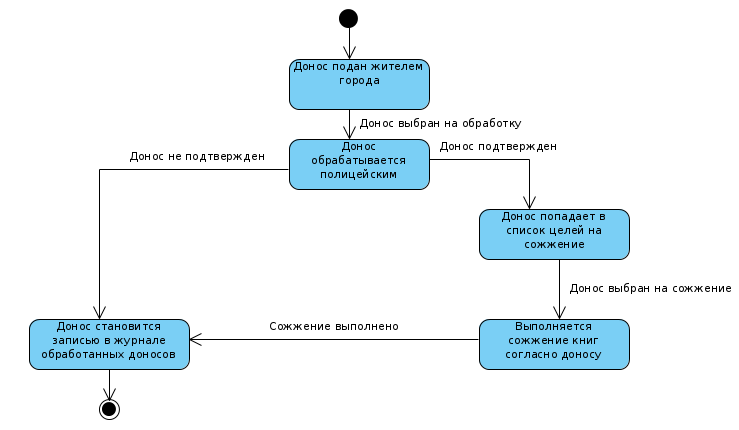
**4.5 *Cooperative diagram***

*Коммуникационная диаграмма обработки пожарным проверенных доносов.*



**4.6 *State diagram***

*Диаграмма состояний доноса в системе*



#### 

#### 

#### 

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

Данный раздел содержит описание слоев, на которые делится приложение, а также интерфейсов их взаимодействия.

Диаграмма пакетов состоит из следующих логических структур:

View - содержит все jsp страницы, которые отображаются пользователю, так же содержат логику валидацию вводимых данных.

Controller - выполняет обработку введеных данных пользователем и вызывает соответствующую бизнес логику, а так же выполняет вывод данных на view

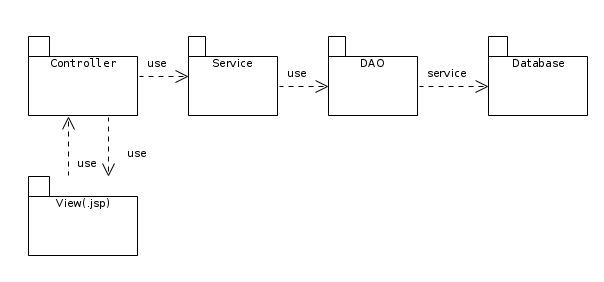
Service - содержит бизнес логику работы системы

DAO - содержит простой интерфейс доступа к данным хранящихся в базе

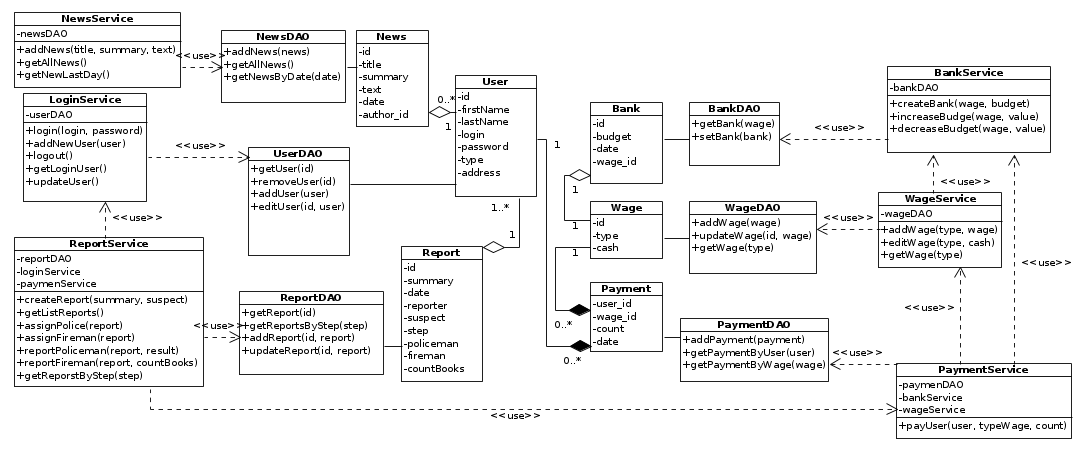
Database - база данных

**5.1 Package Diagram**

Диаграмма основных пакетов в системе

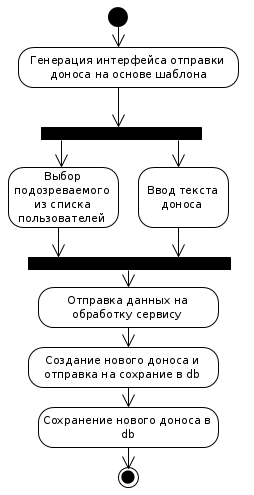


**5.2 Class Diagram**



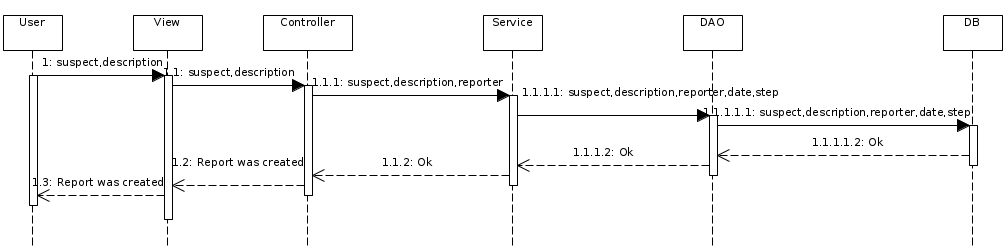
**5.2 Activity Diagram**

Диаграмма деятельности создания доноса пользователем



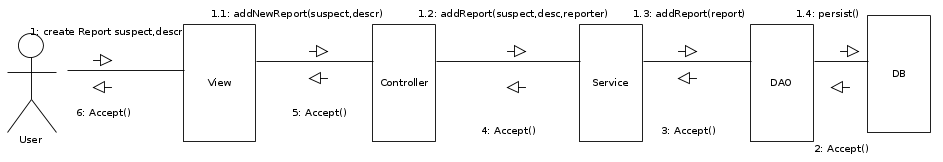
5.3 Secuence Diagram

Отправка доноса пользователем системы



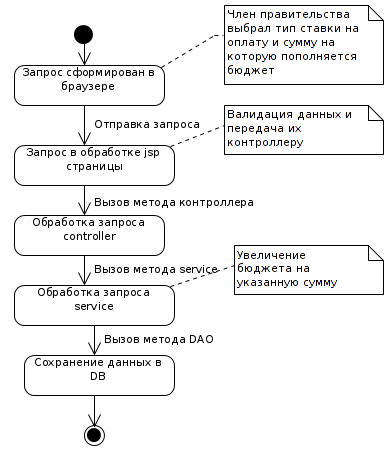
5.3 Communication Diagram

Отправка доноса пользователем системы



5.4 State Diagram

Диаграмма состояний увеличения бюджета членом правительства



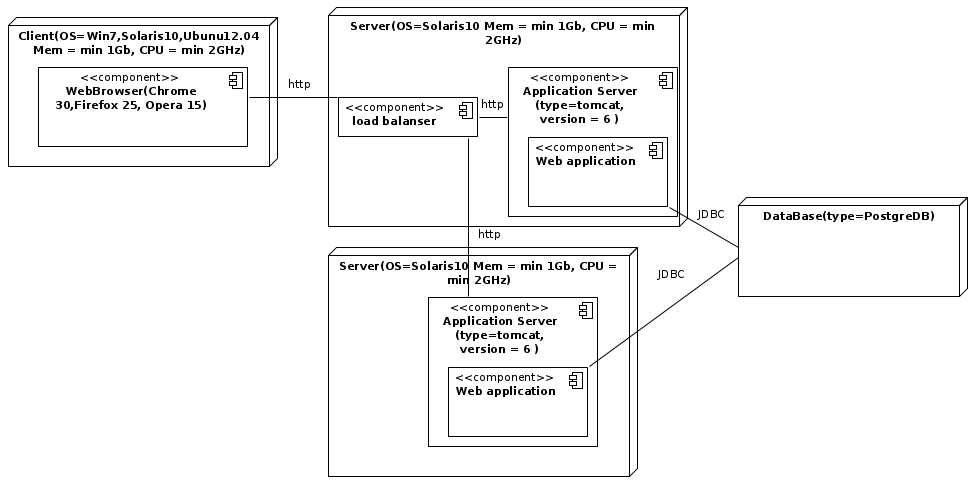
#### 

#### 6**.** **Deployment View**

Данный раздел содержит описание конфигурации файлов, из которых состоит система, мест их расположения и описание взаимодействия их друг с другом.

6.1. Диаграмма развертывания

Диаграмма развертывания определяет физическое оборудование, на котором будет выполняться программная система, а также описывает, как программное обеспечение развертывается на это оборудование и интерфейсы взаимодействия компонентов системы.

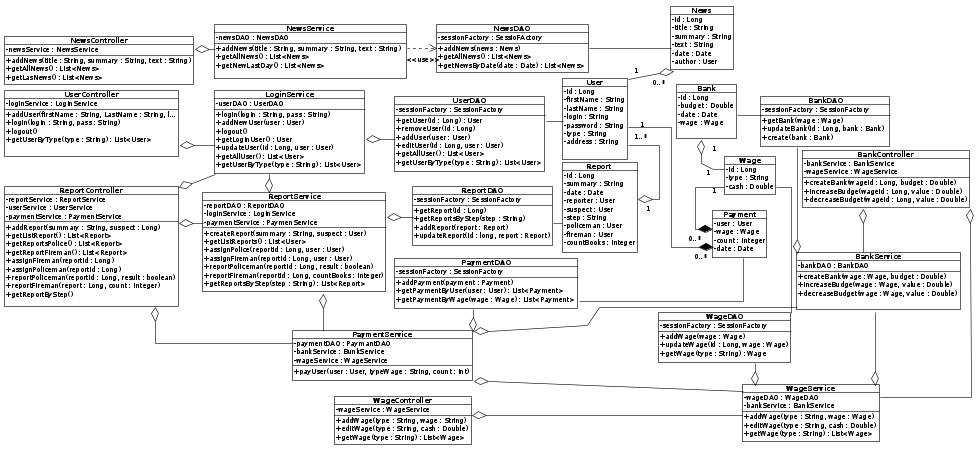


#### 7**.** **Implementation View**

##### 7**.1** **Overview**

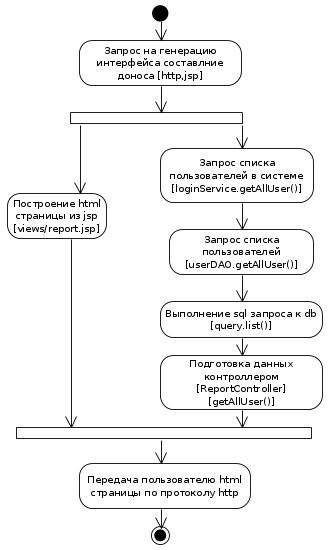
*Данный раздел содержит описание Системы в уже реализованном виде.*

##### 7**.2** Class diagram



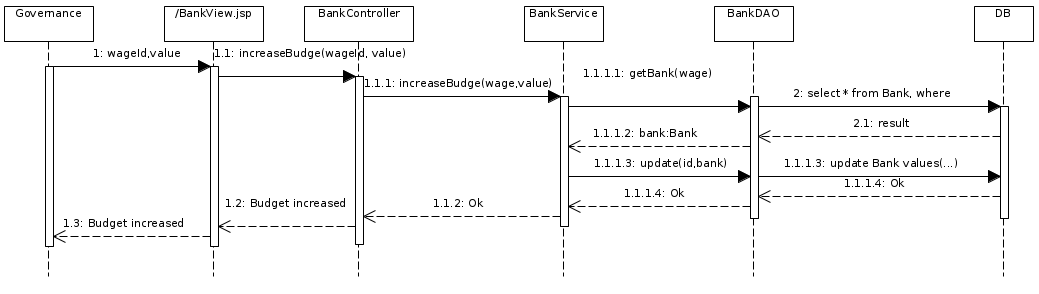
7.3 Activity diagram

Диаграмма деятельности получения списка доносов на жителей полицейским



7.4 Sequence diagram

Диаграмма пополнения бюджета чиновником



#### 

#### 7.5 Cooperative diagram

Диаграмма получения полицейским списка подозреваемых

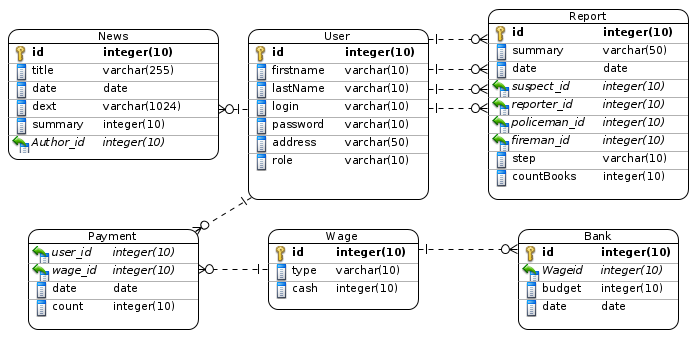
#### 

#### 7.6 State diagram

Диаграмма добавления новостной сводки

#### 

#### 8**.** **Data View**



Описание инфологической модели базы данных:

*User* - данные о пользователях, зарегистрированных в системе.

Поля таблицы:

* id - первичный ключ
* firstName - поле содержит имя пользователя
* lastName - поле содержит фамилию пользователя
* login - поле содержит логин пользователя, через которое пользователь осуществляет вход в систему
* password - поле содержит пароль для аутентификации пользователя
* address - поле содержит домашний адрес пользователя
* role - роль пользователя в системе (sittizen, policeman,fireman,official)

*Report -* данные о доносах, содержащихся в системе.

Поля таблицы:

* id - первичный ключ
* summary - поле содержит описание доноса
* date - поле содержит дату создания доноса
* reporter\_id - поле содержит id пользователя, который создал донос
* suspect\_id - поле содержит id пользователя, на которого был составлен донос
* policeman\_id - поле содержит id полицейского, который обрабатывает донос
* fireman\_id - поле содержит id начальника пожарного расчета, который сжигает книги
* step - поле содержит название стадии, в которой находится донос
* countBooks -поле содержит число сожженных книг

*Payment -* данные о выплатах денег служащим

Поля таблицы:

* *user\_id* - поле содержит id пользователя, которому произведена выплата
* wage\_id - поле содержит id ставки заработной платы
* *date* - поле содержит дату выплаты заработной платы
* *count* - поле содержит коэффициент выплат

*Wage -* данные о заработных платах служащим

Поля таблицы:

* *id* - первичный ключ
* *type -* поле содержит тип должности
* *cash* - пол содержит денежную ставку

*Bank -* данные о текущем балансе бюджета города

Поля таблицы:

* *id* - первичный ключ
* *wage\_id -* поле содержит id ставки заработной платы
* budget - поле содержит текущее количество денег выплаты соответствующих з/п.
* date - поле содержит время последнего изменения бюджета

News - данные о новостных сводках

* *id* - первичный ключ
* *title - поле содержит заголовок новости*
* *summary - поле содержит краткое описание новости*
* *text - поле содержит текст новостной сводки*