Московский государственный технический университет

им. «Н.Э.Баумана»

**Курсовая работа**

по дисциплине:

**«Технологии разработки программного обеспечения»**

на тему:

**«Разработка журнала сообщений»**

исполнители:

студенты группы ИУ5-27 (м)

Тихонов И.В. Лахвич Д. С.

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

преподаватель:

Виноградова М.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва 2014**

Оглавление

[Варианты и распределение студентов. 3](#_Toc388718186)

[1. Этап анализа и планирования требований 4](#_Toc388718187)

[1.1 Постановка задачи. 4](#_Toc388718188)

[1.2 Спецификация основных проектных требований 4](#_Toc388718189)

[1.3 Описание бизнес процессов. 5](#_Toc388718190)

[1.4 модель предметной области 5](#_Toc388718191)

[1.5 Выявленные актеры. 7](#_Toc388718192)

[1.6 Выявленные прецеденты 7](#_Toc388718193)

[1.7 диаграмма прецедентов 7](#_Toc388718194)

[1.8 описание прецедентов 8](#_Toc388718195)

[1.9 Перечень критических рисков 9](#_Toc388718196)

[1.10 Список приоритетов прецедентов. 9](#_Toc388718197)

[1.11 Описание возможной архитектуры 9](#_Toc388718198)

[1.12 Начальная оценка стоимости, затрат и длительности проекта 10](#_Toc388718199)

[1.12 Начальный план выпусков версий. 12](#_Toc388718200)

[2 Этап проектирования (Развитие) 13](#_Toc388718201)

[3 Этап построения 13](#_Toc388718202)

[1 Студент 13](#_Toc388718203)

[1) Полностью определить и реализовать прецеденты (по варианту). 13](#_Toc388718204)

[2 Студент 14](#_Toc388718205)

# Варианты и распределение студентов.

Работу выполняли:

Студент 1 – Тихонов И. В. вариант 92

Паттерны: модуль таблицы (бизнес-логика) и преобразователь данных

(обращение к БД), мост (gof)

Студент 2 – Лахвич Д. С. вариант 90

Паттерны: модуль таблицы (бизнес-логика) и шлюз записи данных (обращение к

БД), компоновщик (gof).

# 1. Этап анализа и планирования требований

## 1.1 Постановка задачи.

Журнал операций: подсистемы

1) фиксация сообщений от внешних систем (сервис и утилита)

2) просмотр и поиск по журналу сообщений

3) экспорт журнала в другие форматы

Необходимо разработать систему для работы с журналом сообщений. Система должна фиксировать сообщения от других систем и выполнять поиск по журналу сообщений. Программа может принимать сообщения например об ошибках других программ. Областью применения данной программы будут компании которые занимаются тестированием приложений. Курсовая работа выполняется двумя людьми. В рамках этого проекта будут рассмотрены 2 подсистемы

1) фиксация сообщений от внешних систем (сервис и утилита)

2) просмотр и поиск по журналу сообщений

## 1.2 Спецификация основных проектных требований

Система «Журнал операций» представляет собой систему с интерфейсом для работы с сообщениями. Интерфейс пользователя позволяет пользователю:

1. Вести поиск в БД сообщений в ручном режиме
2. Вести поиск сообщения по ключевым словам
3. Вносить изменения и удалять сообщения
4. Экспортировать журнал в другие форматы (выполняется 3 учеником которого нет)
5. комментировать сообщения
6. ввод в БД новых сообщений вручную.

Интерфейс пользователя должен отображать следующие поля:

1. таблицу БД журнала сообщений с возможностью ручного корректирования
2. Поле для поиска по журналу сообщений, и кнопка для подтверждения поиска
3. Возможность для подключению к другому источнику (программе) посылающему сообщения.

## 1.3 Описание бизнес процессов.

На рисунке 1 показана бизнес модель журнала сообщений на примере превентивного (предварительного) обслуживания оборудования при помощи SAP HANA и Hadoop. В данной модели журнал сообщений принимает сообщения от серверов и экспортирует анализатору, имея возможность просмотра в ручном режиме.

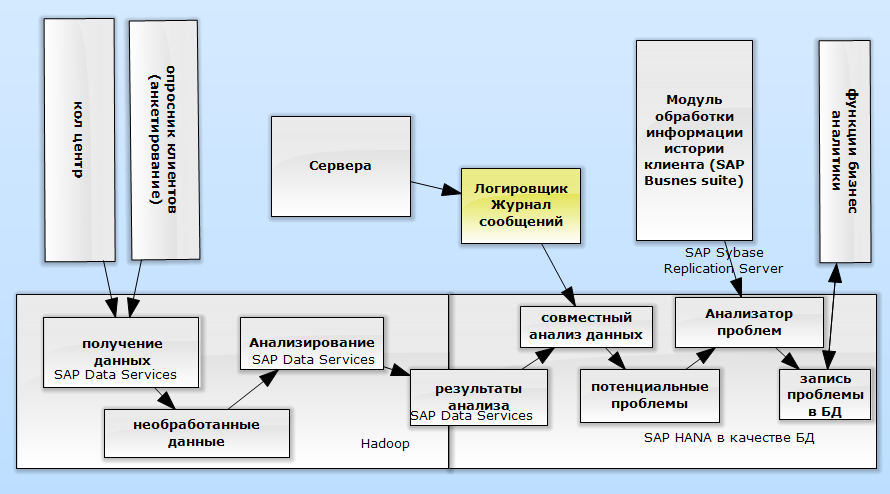


Рисунок 1. Бизнес модель журнала сообщений

## 1.4 модель предметной области

Модель предметной области описан на рисунке 1. //скорее всего будет упрощена

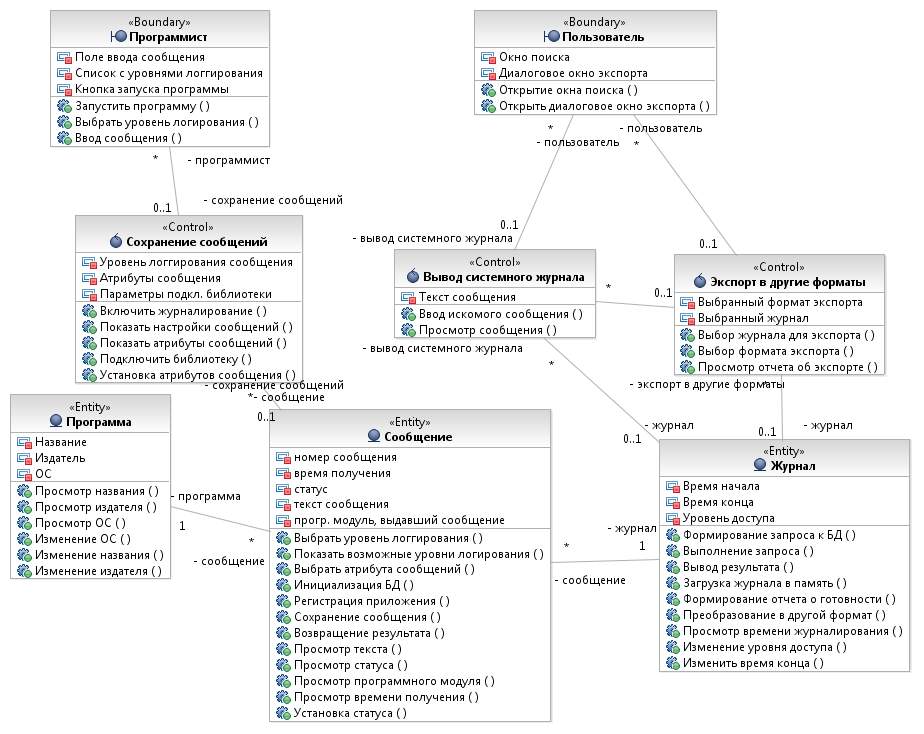


Рисунок 2. Модель предметной области.

На рисунке 1 описана модель предметной области. Она состоит из двух интерфейсов, трех сущностей и трех управляющих классов.

Интерфейсы:

* Интерфейс программиста (программист или внешняя система). Имеет поля ввода сообщения, список с уровнями логгирования и кнопку запуска программы. В случае «сторонней системы» она представляется неким «черным ящиком» отправляющим готовые сообщения, которые перехватывает система.
* Интерфейс пользователя - основной интерфейс пользователей программного продукта. Содержит: окно поиска по БД и диалоговое окно экспорта.

Сущности:

* Сущность программа содержит данные о программе, такие как: Название, Издатель, Операционная система.
* Сущность сообщение хранит все данные о полученном сообщении, такие как: Номер сообщения, Время получения сообщения, статус сообщения, текст сообщения, программный модуль выдавший сообщение.

Управляющие классы:

* Сохранение сообщений: Позволяет системе сохранять сообщения от внешних систем или программиста. Содержит Уровень логгирования сообщения, Атрибуты сообщения, Параметры подключаемой библиотеки.
* Управляющий класс вывод системного журнала. Содержит Текст сообщения.
* Управляющий класс Экспорт в другие форматы. Содержит выбранный формат экспорта, Выбранный журнал для экспорта.

## 1.5 Выявленные актеры.

В нашей системе будет три актера.

1. Пользователь. Он будет работать с интерфейсом системы, читать изменять или корректировать сообщения.
2. Сторонняя система, программа (или программист вручную) которая будет передавать сообщения нашей системе.
3. Система которая позволяет производить изменения в бд и ловит сообщения

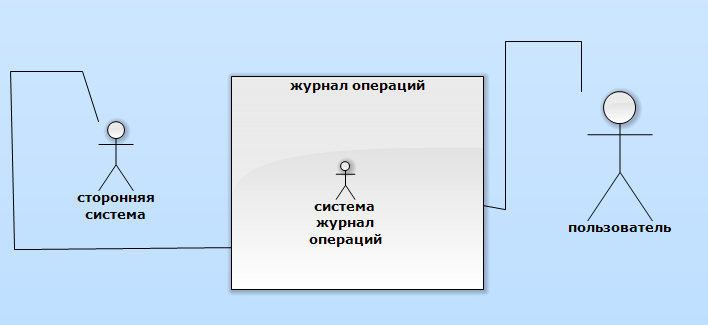


Рисунок 2. выявленные актеры журнала сообщений

## 1.6 Выявленные прецеденты

* Отправка сообщения
* Получение сообщения
* Запись сообщения в БД
* Вывод сообщений
* Поиск по журналу сообщений
* Экспорт в другие форматы

## 1.7 диаграмма прецедентов

Диограмма основных прециндентов Use Case, показана на рисунке 4.

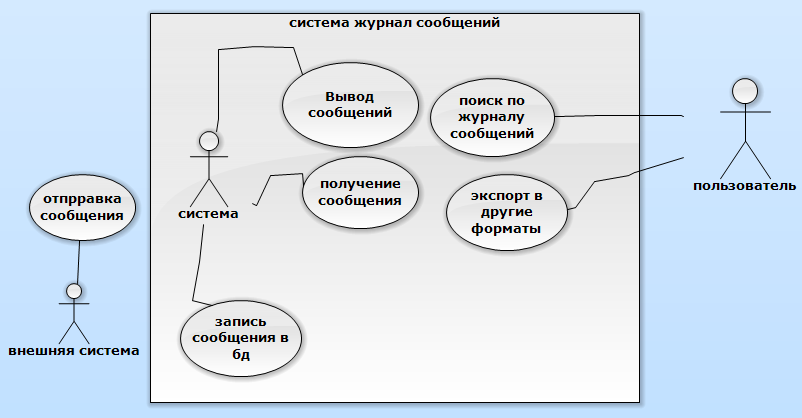


Рисунок 4. диаграмма прецедентов для системы «журнал сообщений»

## 1.8 описание прецедентов

* Отправка сообщения: сторонняя система или программист отправляет сообщения системе «журнал сообщений»
* Получение сообщения: система «журнал сообщений» принимает сообщение от сторонней системы.
* Запись сообщения в БД: принятое сообщение записывается в базу данных.
* Вывод сообщений: по запросу пользователя из БД система выводит на экран журнал сообщений.
* Поиск по журналу сообщений: позволяет пользователю вести поиск по журналу сообщений, и просматривать оставленные сообщения
* Экспорт в другие форматы: позволяет пользователю сохранять журнал сообщений в других форматах.

## 1.9 Перечень критических рисков

Выделяют три категории источников риска: проектный риск, технический риск, коммерческий риск.

Для данного проекта наиболее значимый коммерческий риск, и иметься проектный риск. Технический риск незначителен. Поскольку на рынке уже имеются такие системы, коммерческие риски наиболее значительны. Коммерческие риски значимые для этого проекта:

1. создание продукта, не требующегося на рынке;
2. создание продукта, отстающего от требований рынка;

Значимые проектные риски: не уложиться в бюджет, нехватка ресурсов и персонала.

## 1.10 Список приоритетов прецедентов.

* Приоритет 1. Наиболее приоритетные прецеденты: Получение сообщения, Запись сообщения в БД.
* Приоритет 2. Средние по значимости прецеденты: Получение сообщения, Поиск по журналу сообщений, Вывод сообщений.
* Приоритет 3. Низкие по значимости прецеденты: экспорт в другие форматы, отправка сообщения. ( отправкой сообщений занимается другая система).

## 1.11 Описание возможной архитектуры

Изначальный выбор архитектуры, учитывает основные технические и функциональные структурные компоненты и связи между ними. На рисунке 5 показана базовая архитектура журнала сообщений.

перечень архитектурно-значимых прецедентов?

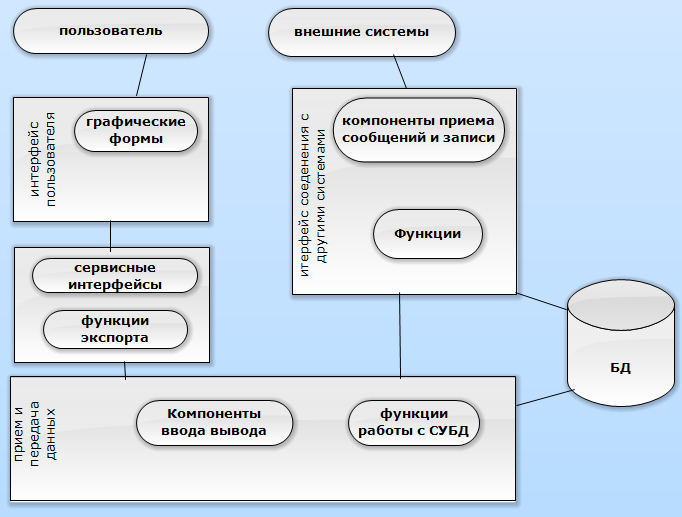


Рисунок 5. Базовая архитектура журнала сообщений.

Обобщенные механизмы проектирования:

Буем использовать каскадную модель проектирования (классический жизненный цикл), поскольку он дает план и временной график по всем этапам проекта, упорядочивает ход конструирования. Проект нам надо сдать уже готовый.

Системное ПО:

Кодирование будм производить в среде visual studio 2013, диаграммы на Software IdeasModeller, MS Project, БД Microsoft sql server, на Windows 8, Поскольку эти продукты быстрые, надежные и имеются в наличии.

Планируется использовать языки C# , SQL, приложение планируеться типа Forms Application. // добавим реально используемые компоненты.

// необходимо привести варианты, их достоинства и недостатки и обоснование выбора. ПОЗЖЕ СДЕЛАЕМ

## 1.12 Начальная оценка стоимости, затрат и длительности проекта

Используется конструктивная модель стоимости (COCOMO II), этапа композиции приложения на основе объектных указателей.

Система использует 1 экран и 1 отчет. Экран используемый системой, это экран интерфейса пользователя, определяется как «простой» (количество представлений <7 и <4 таблиц данных). Отчет так же является простым так как содержит одну таблицу данных.

Процент повторного использования программных компонентов %REUSE равен нулю (новый проект). Таким образом, количество новых объектных указателей (NOP) = (Объектные указатели) х [(100 - %REUSE) /100] = 3 х (100 – 0)/100 = 3.

Скорость разработки (PROD), рассчитывается как номинальная (учитывая высокую зрелость среды разработки и низкий опыт разработчиков). PROD = 13.

Затраты, вычисляются формулой:

ЗАТРАТЫ = NOP /PROD [чел.-мес] = 3/13 = 0,23 [чел.-мес]

Далее, вычисляется стоимость проекта, учитывая среднее значение рабочего коэффициента равен $1 000 за чел.-месяц (рекомендуется брать 15000 но студенты мало получают):

СТОИМОСТЬ = ЗАТРАТЫ х РАБ\_ КОЭФ = 0,23 х $1000 = $230

Для вычисления оценки длительности разработки используется расчет характеристики масштабных факторов, показанный на таблице 2.

*Таблица 2.*

|  |  |  |
| --- | --- | --- |
| **Масштабный фактор (*Wi*)** | **Значение** | **Описание** |
| Предсказуемость PREC | 4 | низкий опыт в разработке данной программы |
| Гибкость разработки FLEX | 4 | клиент установил общие цели, и дал идеи осуществляемого процесса |
| Разрешение архитектуры /риска RESL | 3 | средний анализ рисков |
| Связность группы TEAM | 2 | уровень взаимодействия выше среднего |
| Зрелость процесса РМАТ | 3 | для оценки зрелости процесса в организации, берется номинальное значение |

**

*В =* 1,01 + 0,01 (4 + 4 + 3 + 2 + 3) = 1,17

После определения затрат и стоимости отцениваем длительность разработки:

*TDEV* = [3,0х(ЗАТРАТЫ)(0,33+0,2(B-1,01))] х SCEDPercentage/100 [мес]

SCEDPercentage — процент увеличения (уменьшения) номинального графика. Учитывая рекомендацию, полученную при анализе проектного риска, имеем SCEDPercentage = 40. Таким образом, получаем длительность:

TDEV = [3,0х(0,23)(0,33+0,2(1,17-1,01))]х 40/100 = 0.7 [мес]

## 1.12 Начальный план выпусков версий.

На рисунках 6,7,8 показан начальный план выпусков версий, выполненный на MS Project.

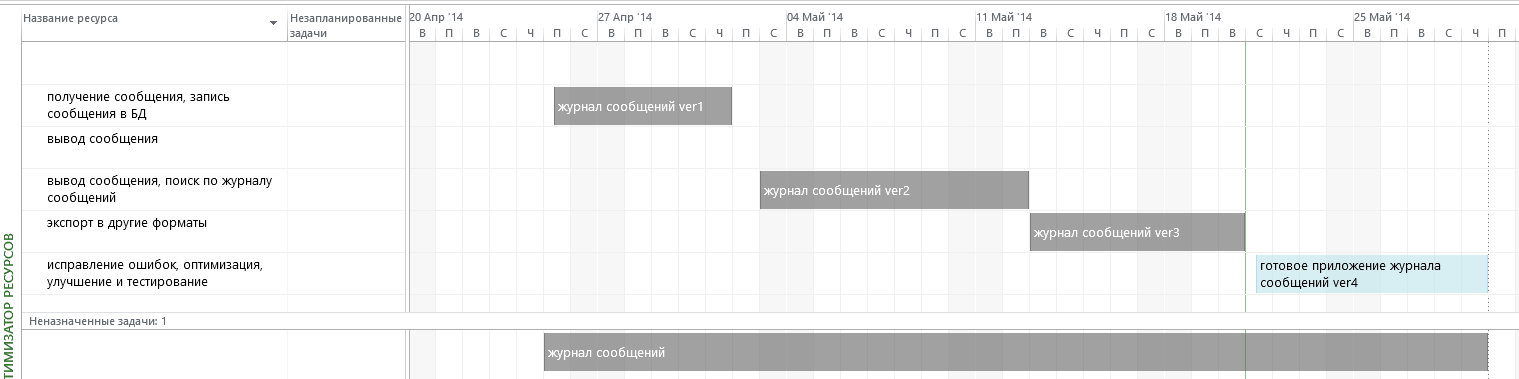


Рисунок 6. Начальный план выпусков версий.

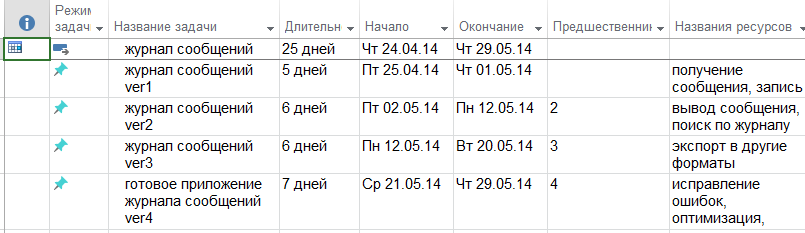


Рисунок7. Начальный план выпусков версий.

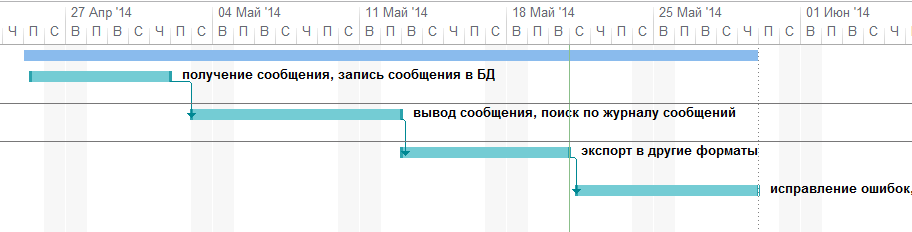


Рисунок 8. Начальный план выпусков версий.

# 2 Этап проектирования (Развитие)

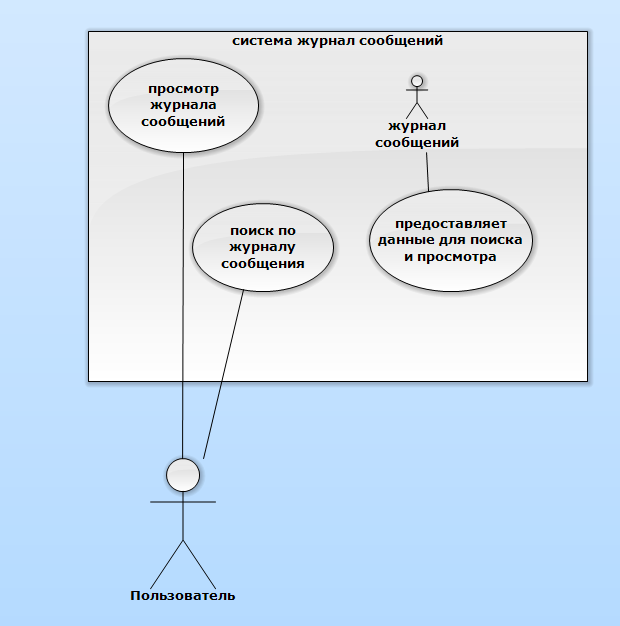
2.1

# 3 Этап построения

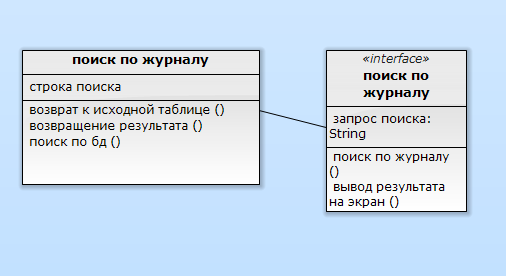
## 1 Студент

### Полностью определить и реализовать прецеденты (по варианту).

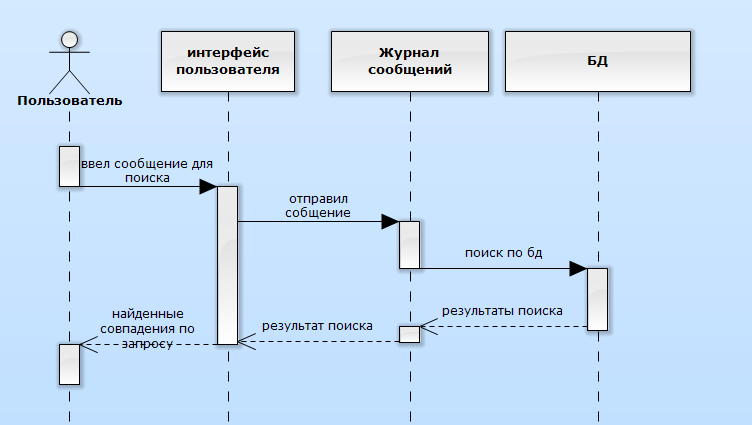
Диаграмма прецедентов для 1 студента.



1. Диаграмма(ы) классов подсистемы



1. Полностью разработать компоненты с использованием паттернов (по вариантам).
2. Диаграммы последовательностей для основных функций программы,

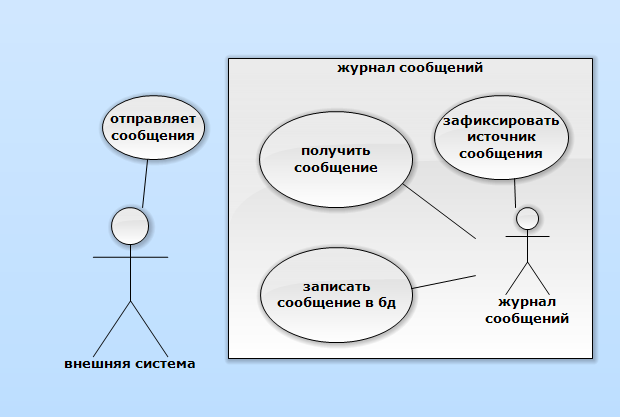


1. Диаграммы классов и последовательностей для иллюстрации примененных паттернов.
2. Провести тестирование и отладку.

## 2 Студент

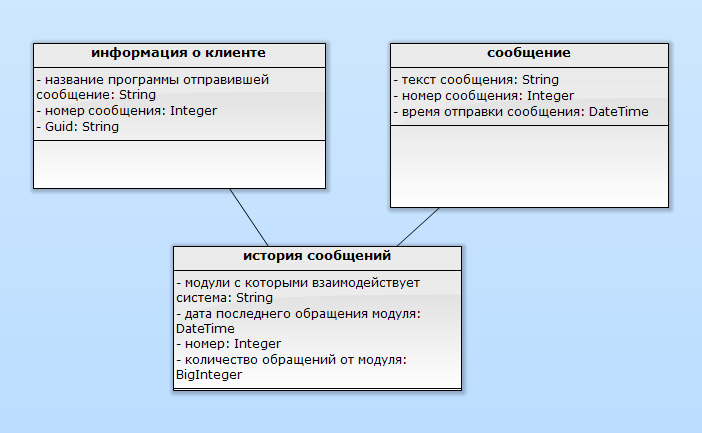
### Полностью определить и реализовать прецеденты (по варианту).

Диаграмма прецедентов для 2 студента.

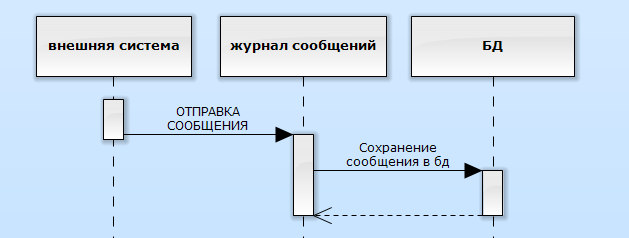


1. Диаграмма(ы) классов подсистемы,

Описание БД системы журнала сообщений



1. Полностью разработать компоненты с использованием паттернов (по вариантам).
2. Диаграммы последовательностей для основных функций программы



1. Диаграммы классов и последовательностей для иллюстрации примененных паттернов.
2. Провести тестирование и отладку.