**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра информационных систем управления**

**ОТЧЕТ ПО ППС**

**Генерация кода проектируемого программного обеспечения и разработка приложения Тестирование разработанного приложения и финальный отчет**

Выполнил студент группы № 12

*Шишлянников Иван Викторович*

**Минск 2020**

Оглавление

[Постановка задачи 3](#_Toc41439483)

[Индивидуальный план преподавателя ВУЗа. 3](#_Toc41439484)

[Определение требований 4](#_Toc41439485)

[Функциональная модель системы в виде диаграмм вариантов использования 4](#_Toc41439486)

[Глоссарий 5](#_Toc41439487)

[Сценарии 6](#_Toc41439488)

[Анализ вариантов использования 11](#_Toc41439489)

[Проектирование архитектуры системы 18](#_Toc41439490)

[Проектирование элементов системы 20](#_Toc41439491)

[Разработка приложения 24](#_Toc41439492)

[Структура приложения 24](#_Toc41439493)

[База данных 24](#_Toc41439494)

[Скриншоты приложения 25](#_Toc41439495)

[Результаты тестирования 28](#_Toc41439496)

[Integration testing 29](#_Toc41439497)

[System testing 29](#_Toc41439498)

[Acceptance testing 30](#_Toc41439499)

[Scenario testing 31](#_Toc41439500)

# Постановка задачи

## Индивидуальный план преподавателя ВУЗа.

**Общие сведенья**

Для каждого преподавателя (ФИО, Год рождения, Домашний адрес, Контактные телефоны) конкретного высшего учебного заведения (Код, Название, Краткое название) на каждый учебный год (Год начала учебного года, Год окончания учебного года) формируется индивидуальный план. В индивидуальном плане отражается общий объем работ преподавателя, который он должен выполнить в течение учебного года.

**Назначение и цели создания (развития) системы**

Необходимо осуществлять следующую обработку данных:

• формирование для каждого преподавателя итоговой суммы (в часах) запланированных и выполненных объемов работ по семестрам;

• список преподавателей, у которых фактическое значение выполненных работ превышает плановое (факультет, кафедра, ФИО, уч. степень, уч. звание, должность, семестр, кол-во перевыполненных объемов работ);

• список преподавателей заданной кафедры, имеющих заданную ученую степень на заданную дату.

**Характеристика объекта автоматизации**

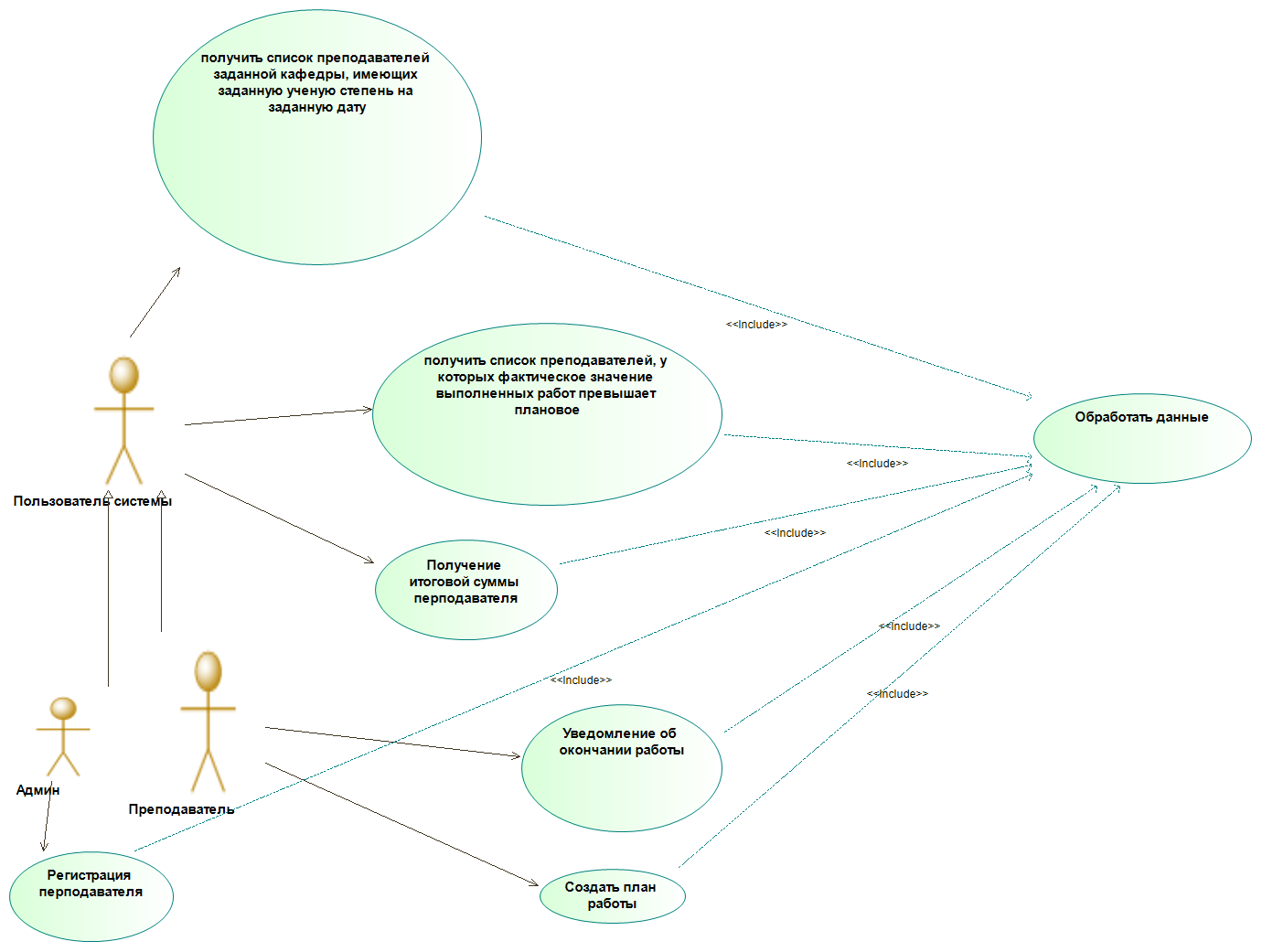
Объектом автоматизации является сервер. На нем хранятся все данные.

**Требования к системе**

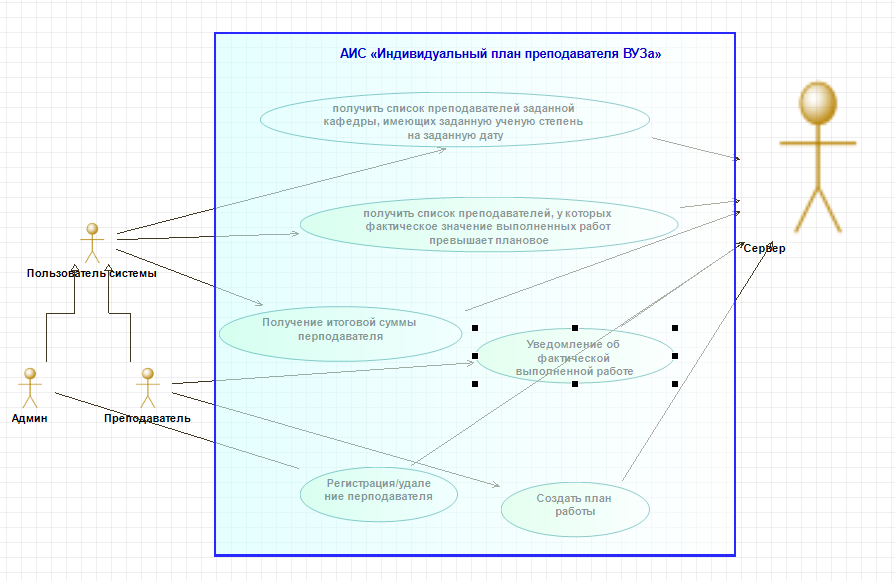
Система отвечает за хранение и обработку всех данных полученных от любого ее пользователя.

# Определение требований

## Функциональная модель системы в виде диаграмм вариантов использования



Бизнес – диаграмма



Main-диаграмма

## Глоссарий

|  |  |
| --- | --- |
| Админ(admin) | Системный администратор, отвечающий за добавление новых преподавателей в систему. |
| Преподаватель(teacher) | Пользователь сети. Имеет доступ к обновлению своих данных. Составляет свой учебный план. |
| Сервер(server) | Запоминающее устройство всей сети. Занимается обработкой и хранением данных о всех преподавателях в сети. |
| Регистрация пользователя (signing up new user) | Добавление нового пользователя в сеть |
| Уведомление об окончании работы (end of working plan report) | Преподаватель по итогу выполнения определенной деятельности, сообщает об этом системе. |
| Получение итоговой суммы преподавателя (getting checksum) | формирование для каждого преподавателя итоговой суммы (в часах) запланированных и выполненных объемов работ по семестрам, так же с комментариями при несоответствии отработанных фактически и запланированных часов; |

## Сценарии

Сценарии для вариантов использования будем описывать по следующему шаблону:

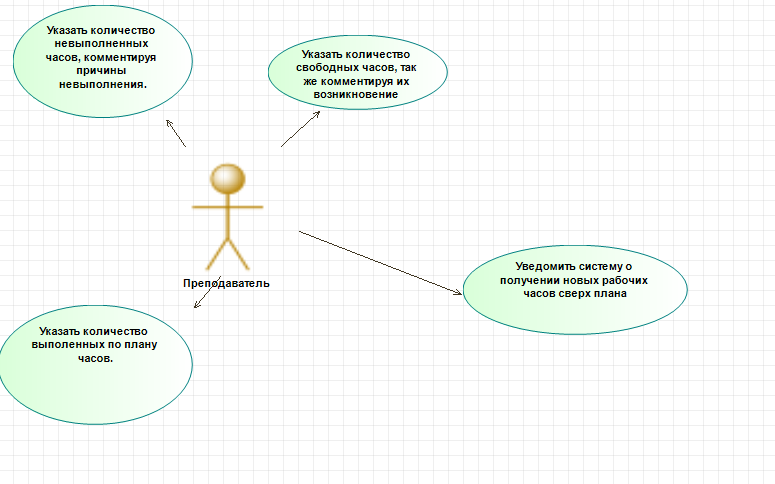
• Х.1 предусловия;

• Х.2 главный сценарий;

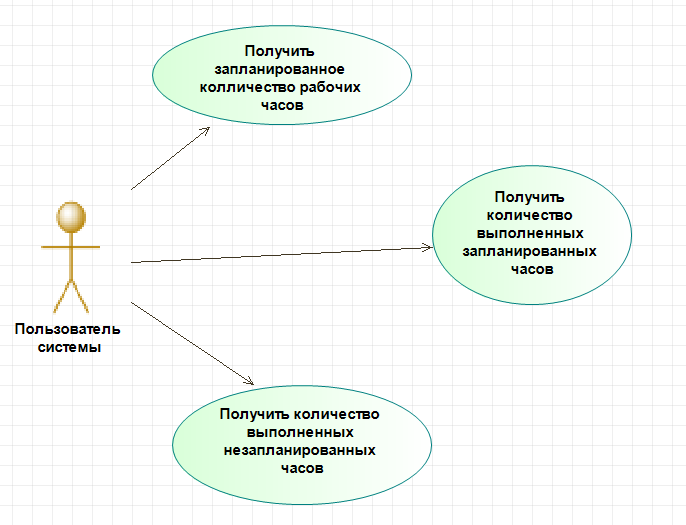
• Х.3 альтернативный сценарий;

1. Сервер
   1. При первом запуске системы создается пустой аккаунт администратора со стандартным пароль и логином (admin, admin);
   2. Сервер находится в состоянии ожидания запроса от пользователя системы, после обработки полученного запроса снова переходит в состояние ожидания.
   3. При различных сбоях системы сервер может “перезагрузиться”, или “откатиться” до своего предыдущего состояния.
2. Преподаватель
   1. При первом входе в систему может изменить пароль
   2. Преподаватель составляет свой план на текущий семестр, или же отмечает уже выполненный.
   3. При различных как системных, так и пользовательских ошибках, преподаватель редактирует необходимые данные вручную либо же просит об этом администратора.
3. Администратор
   1. При первом входе в систему может изменить пароль и логин.
   2. “Создает” или “Удаляет” преподавателей в системе.
   3. При различных как системных, так и пользовательских ошибках Администратор может напрямую работать с сервером, “перезагрузить”, “отключить” сервер.
4. Регистрация преподавателя
   1. Администратор, вводит во всплывшем окне данные нового преподавателя. В результате создается новый пользовать с логином и паролем admin, admin. Каждому преподавателю задается определенное количество часов для отработки.
   2. При невалидных данных , система потребует их повторного ввода.
5. Создать план работы
   1. .
   2. “Создает” новый план работы. Во всплывающем окне преподаватель вводит данные о своем плане. Часы отведенные на план не могут превышать часы отведенные на преподавателя.
   3. При невалидных данных , система потребует их повторного ввода.
6. Уведомление об окончании работы
   1. .
   2. Преподаватель, выбрав в списке планов необходимый план, подтверждает свой выбор, после чего данный план помечается как “выполненный” и попадает в список “выполненные планы”.
   3. .
7. Получить итоговую сумму преподавателя
   1. .
   2. Пользователь, выбрав в списке преподавателей необходимый план, выбирает опцию “получить итоговую сумму” и подтверждает свой выбор, после чего во всплывающем окне выводится итоговая сумма преподавателей, посчитанная системой.
   3. .
8. Получить список преподавателей (по критерию X)
   1. .
   2. Пользователь, выбрав в главном меню опцию “получить список”, выбирает опцию “по критерию X” и подтверждает свой выбор, после чего во всплывающем окне выводится список преподавателей по критерию X, полученный системой.
   3. .

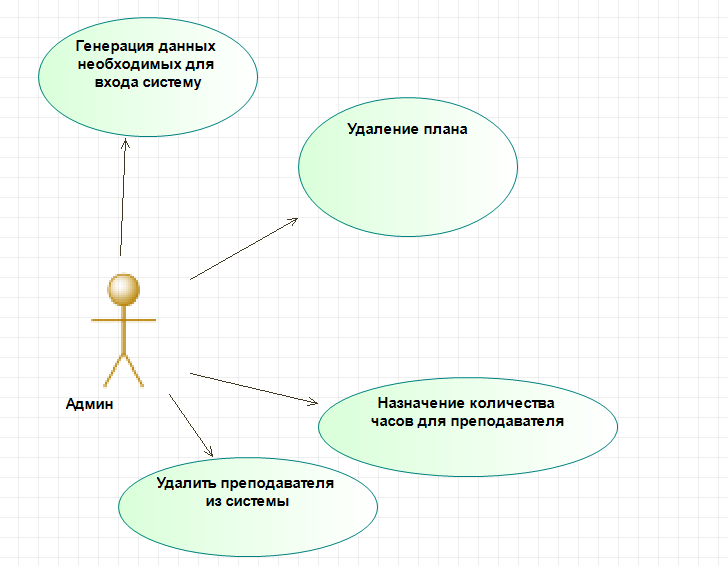
Доп диаграма : уведомление о фактически выполненных часах



Получение контрольной суммы

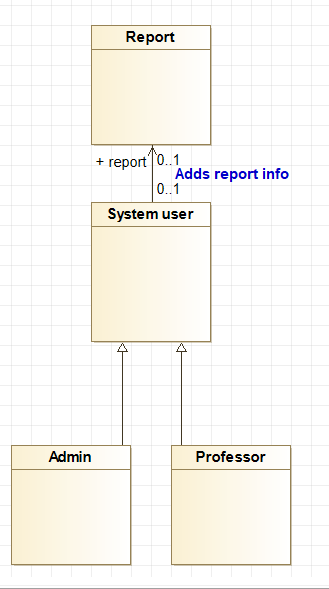


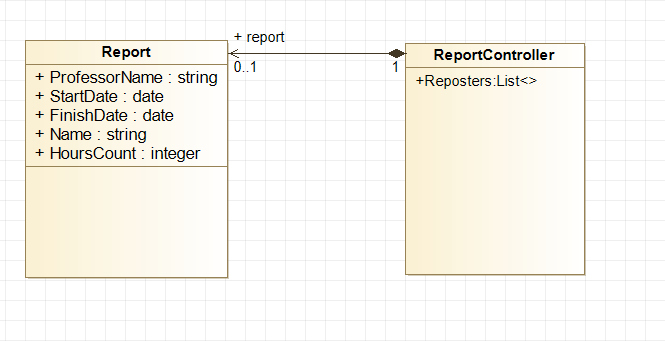
Регистрация/удаление преподавателя



# Анализ вариантов использования

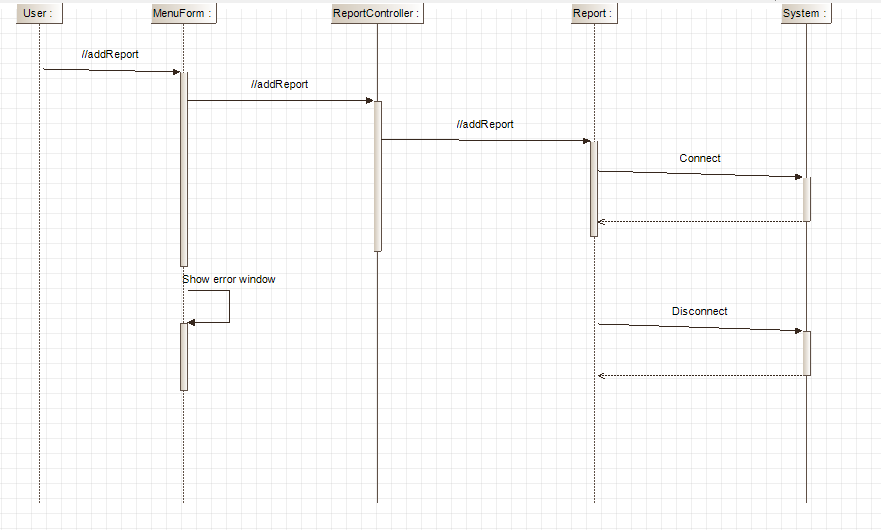
Диаграммы классов

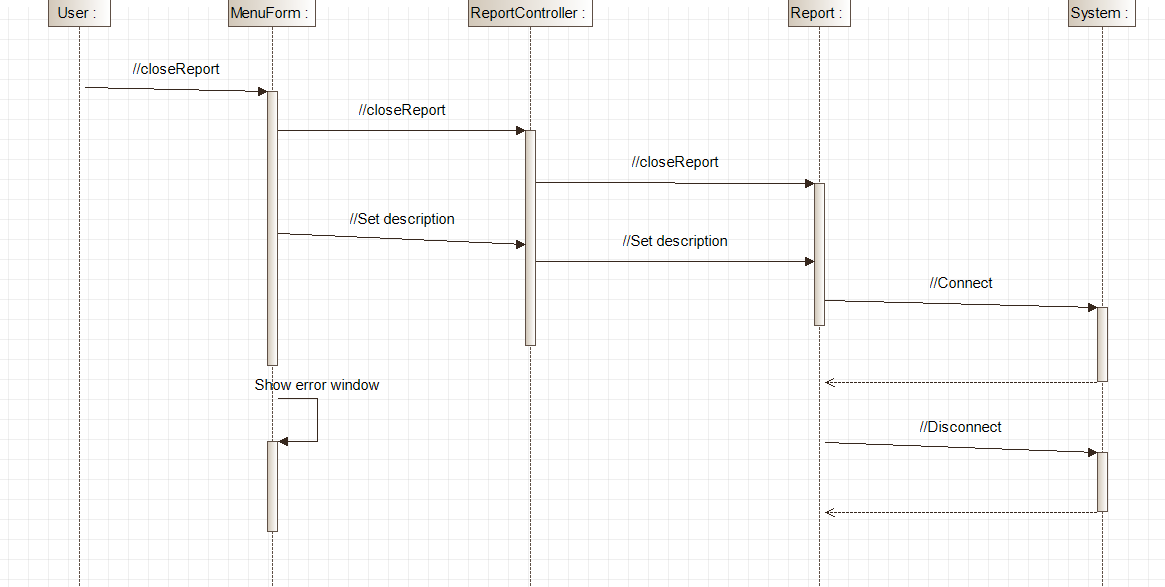




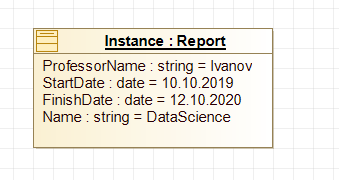


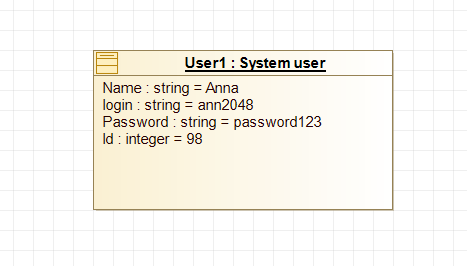
Диаграммы последовательности



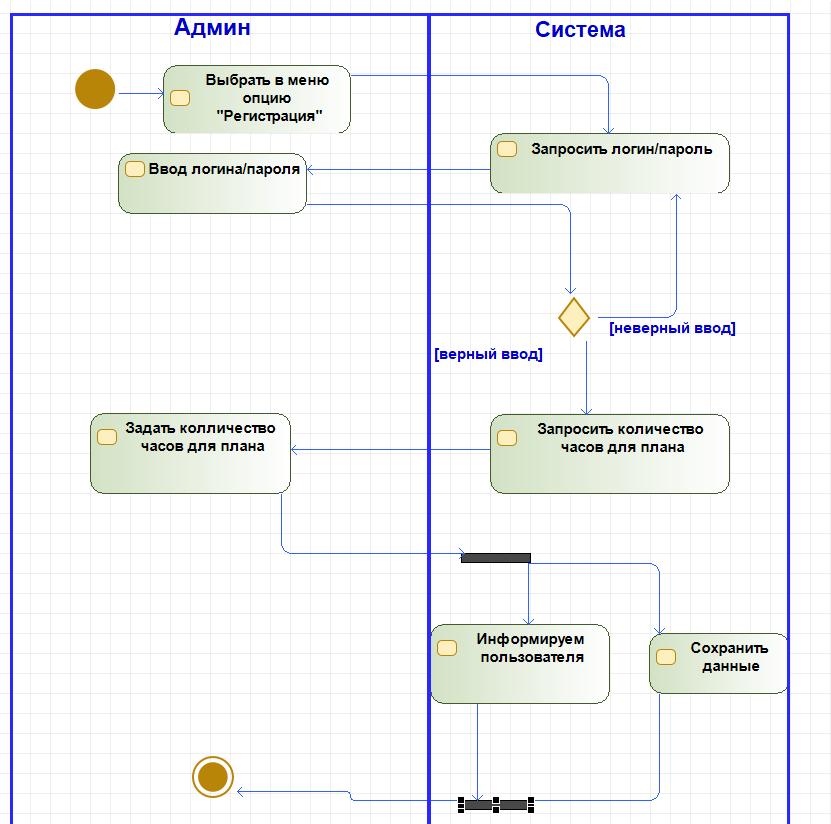


Диаграммы объектов

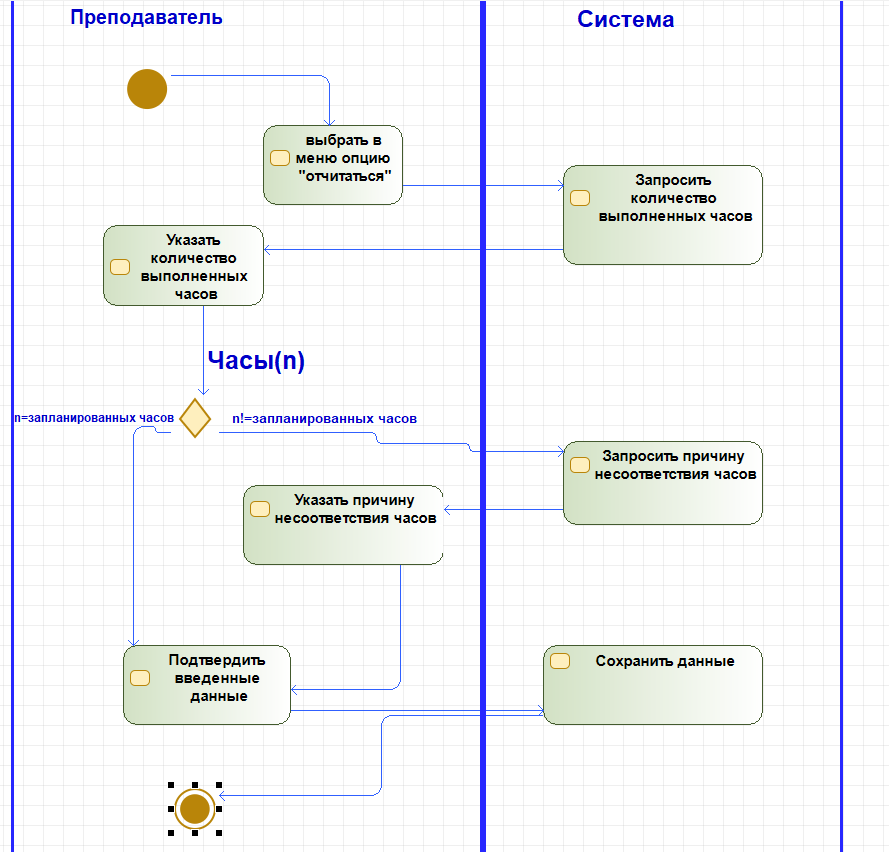




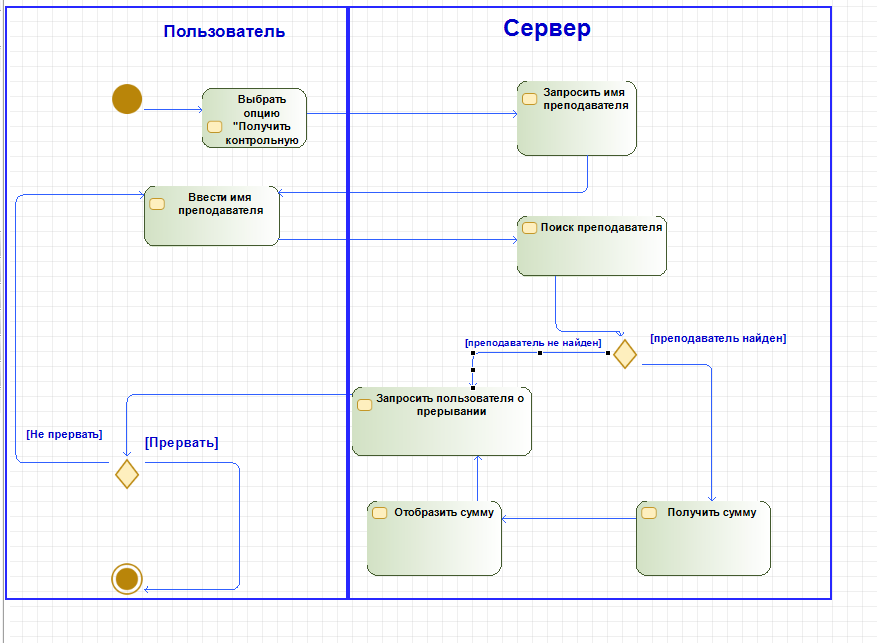
Диаграма действия “Регистрация нового преподавателя”



Диаграма действия “Уведомление о фактически выполненной работе”

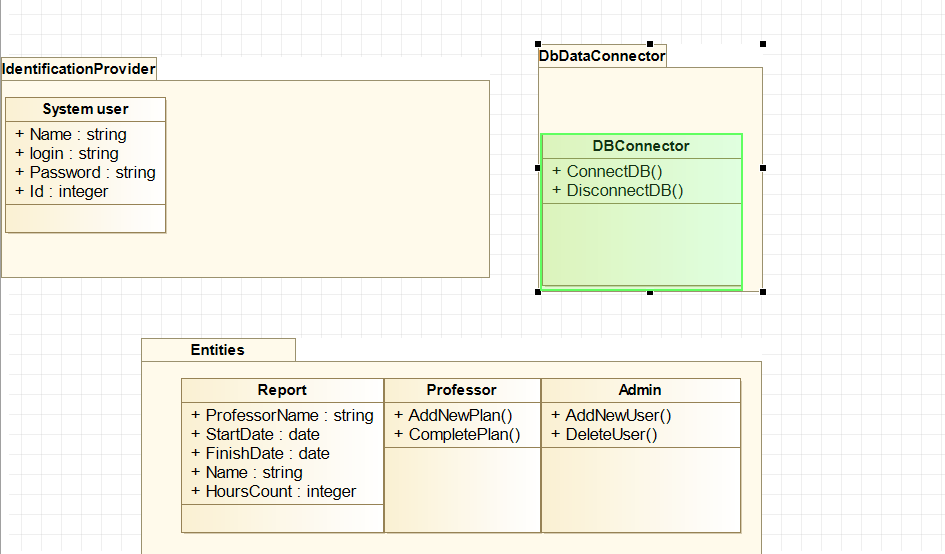


Диаграма действия “Получить итоговую сумму преподавателя”



# Проектирование архитектуры системы

Диаграмма пакетов



Диаграммы развертывания

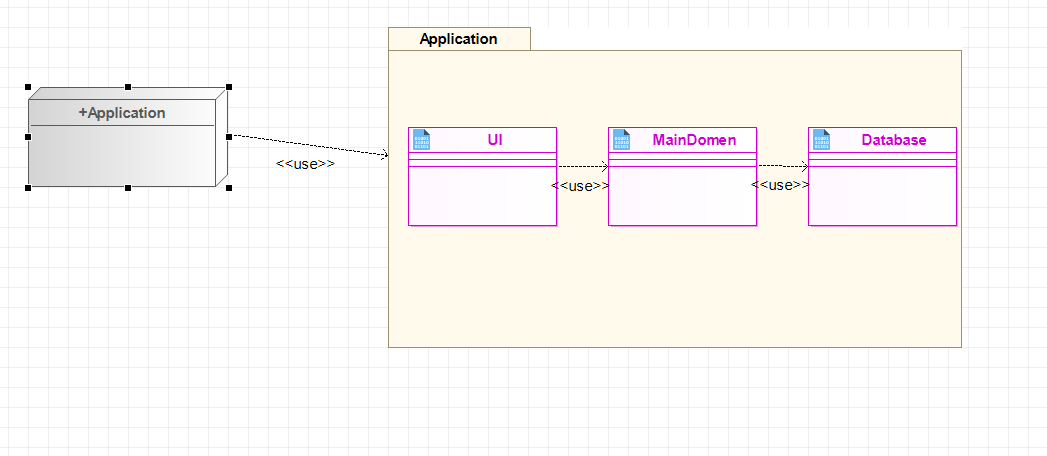
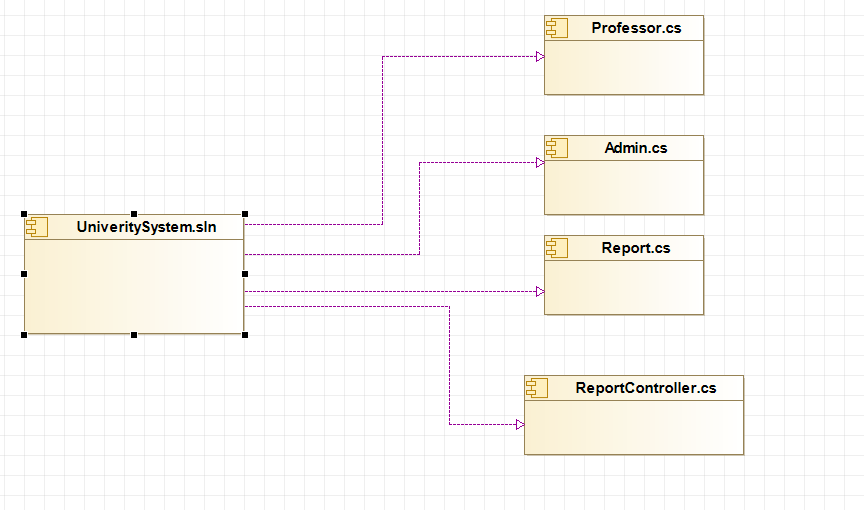
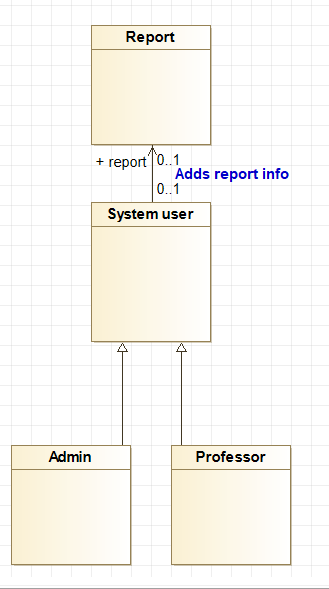


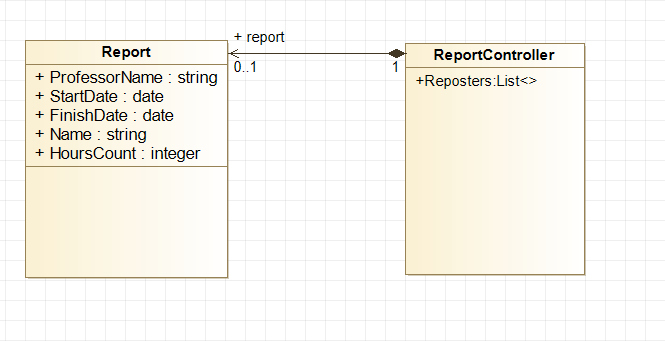
Диаграмма компонентов



# Проектирование элементов системы

Диаграммы классов







Диаграма состояния “Получение списка преподавателей по критерию”



Диаграма состояния “Создание плана”

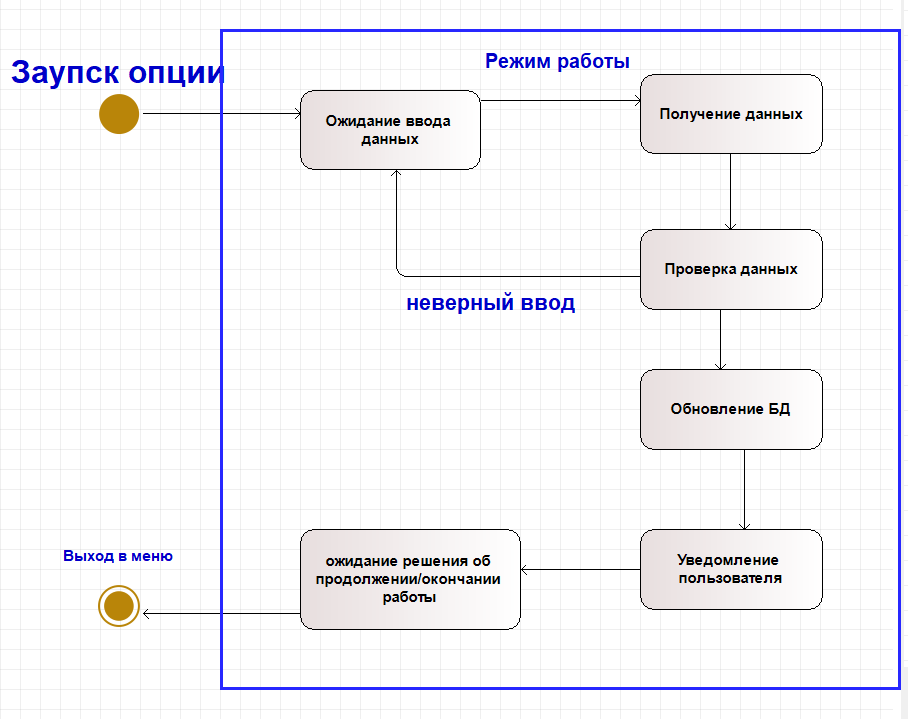


Диаграмма базы данных.

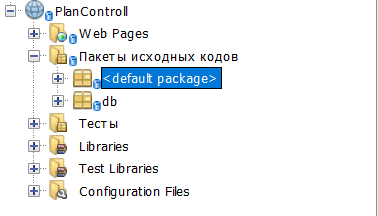


# Разработка приложения

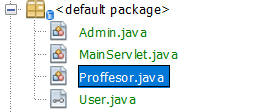
## Структура приложения

Разработка приложения велась в среде NetBeans, так как данная среда отлично подходит для создания web приложения.

Структура проекта:



Классы осуществляющие работы программы:

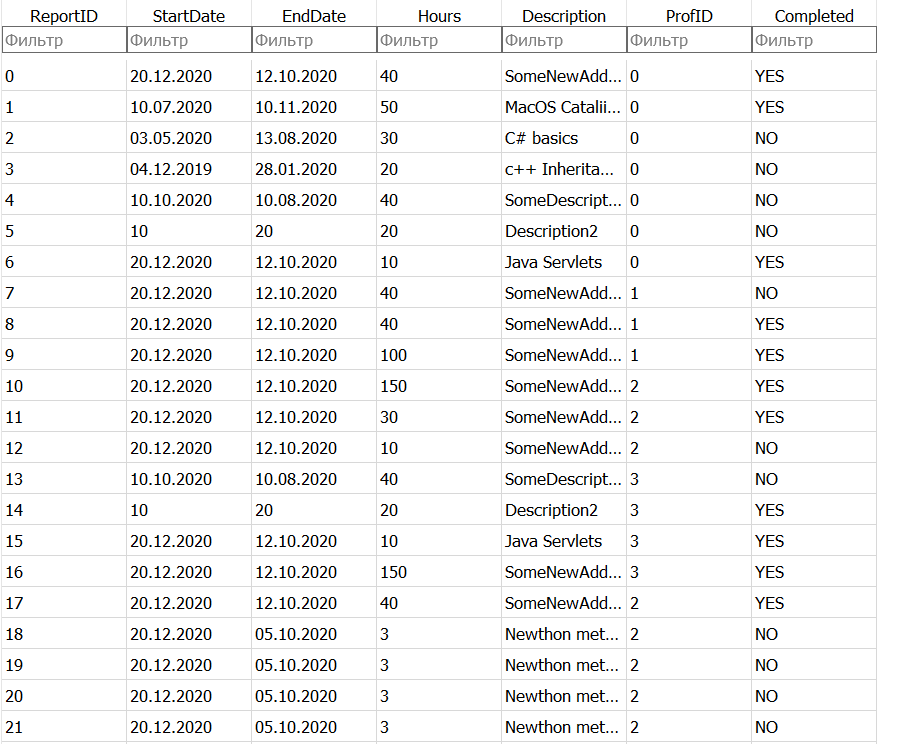


## База данных

Разработка и БД велась при помощи программы SQLite DB Browser.

Фрагменты базы данных:

Таблица записей о задачах преподавателей:

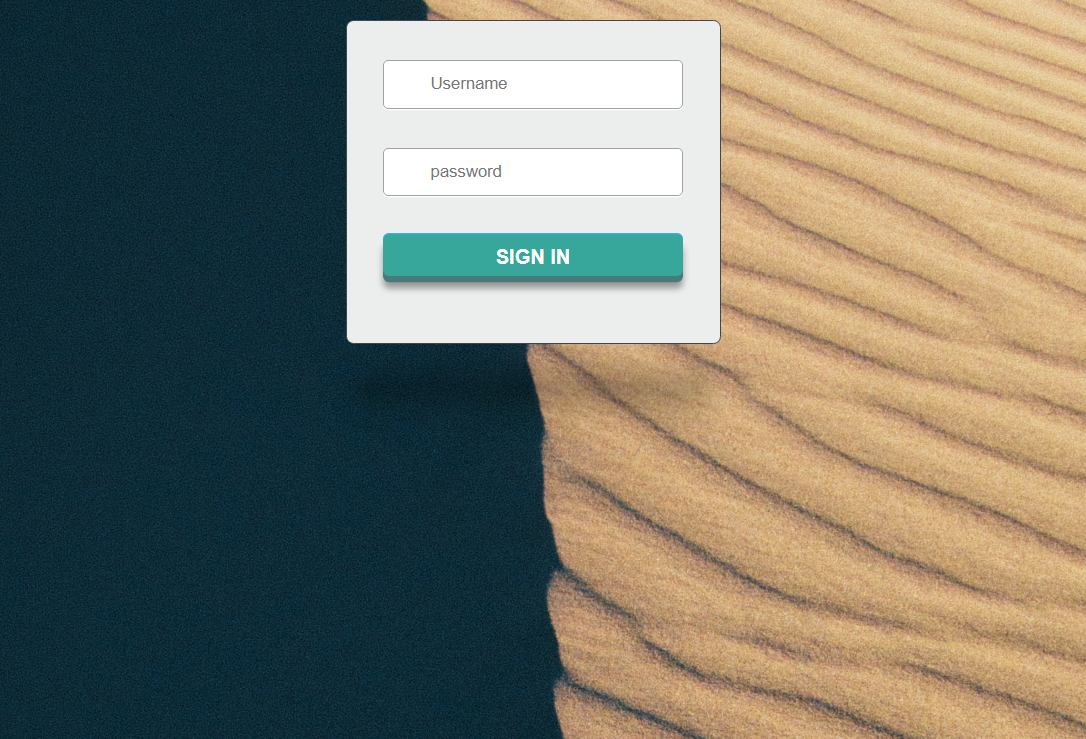


Преподаватели:

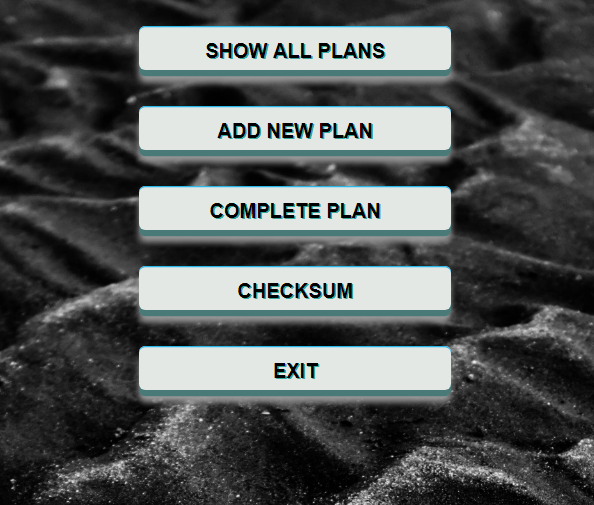


## Скриншоты приложения

Авторизация:



Главное меню преподавателя:



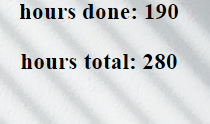
Завершения выполнения задачи



Вывод информации задач запросившего преподавателя:

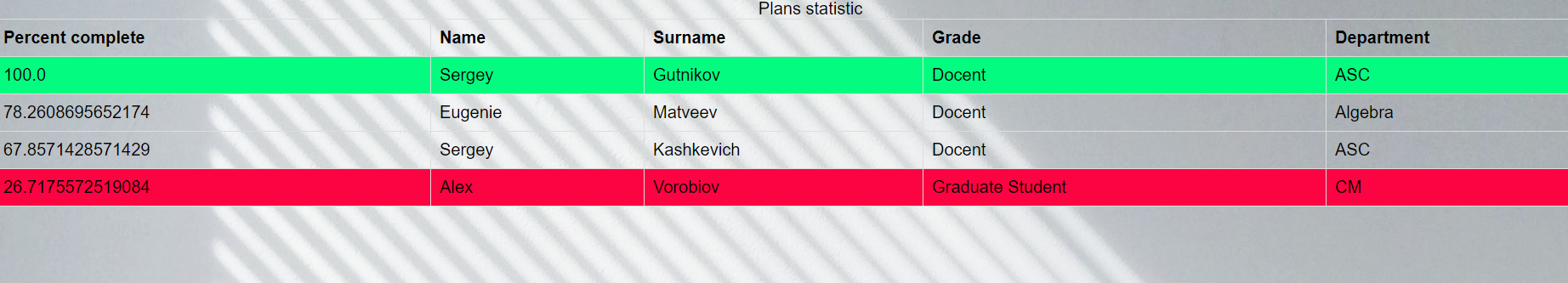


Информация о чек-суммах преподавателя:

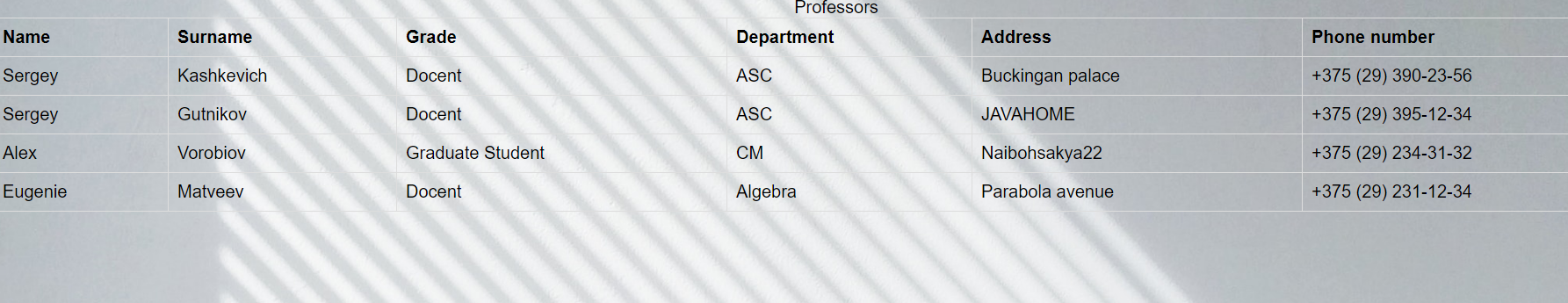


Информация о всех преподавателях доступная только администратору:

Статистика по выполнению плана



Вся информация о преподавателях:

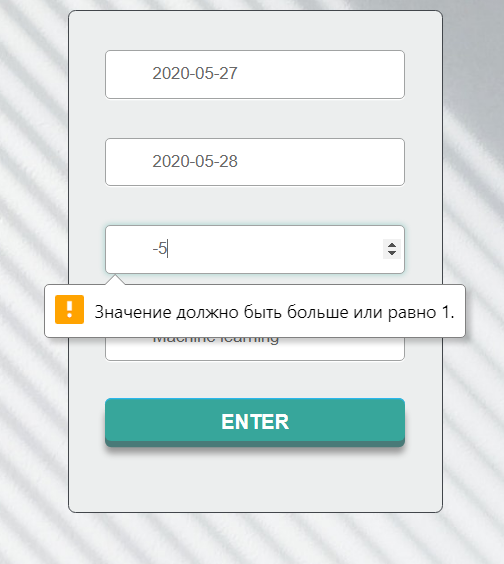


# Результаты тестирования

1. Тест на правильность ввода количества часов, уделенное на план: *void SetHoursCount (int count)* должна принимать числовое значение и быть больше нуля

|  |  |  |
| --- | --- | --- |
| № | Вход | Результат тестирования |
| 1 | 80 | True |
| 2 | 0 | False |
| 3 | -123 | False |

Фрагмент тестирования:



2. Тест на правильность поиска преподавателя: *Professor GetProfessor (string name)*

|  |  |  |
| --- | --- | --- |
| № | Вход | Результат тестирования |
| 1 | Kash | Valid user |
| 2 | Gutos | Valid user |
| 3 | Gena | null |

## Integration testing

Проверяем подключение к БД. Работа библиотеки sqlite.

(наличие библиотеки sqlite)

Авторизация пользователя произошла успешно, что следует об успешном подключении БД ( sqlite)

## System testing

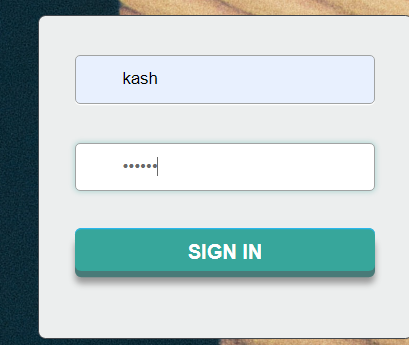
Пользователь может ввести логин и пароль.

Система ищет соответствие в БД. Если авторизация успешна, пользователю становится доступен весь функционал приложения.

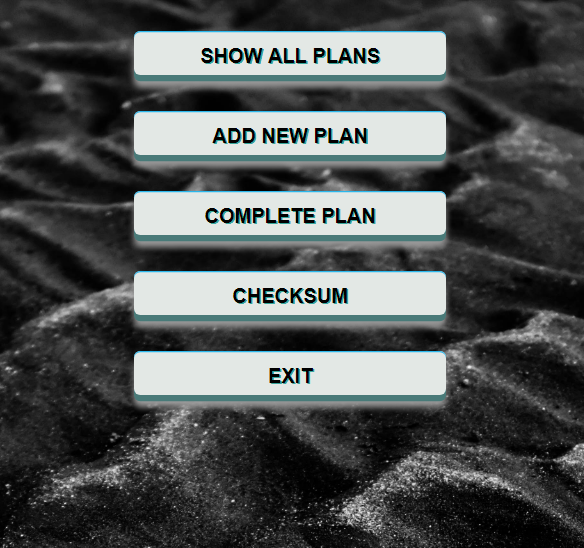
Соответственно, необходимо протестровать:

1. Подключение к базе данных
2. Корректность поиска

Попытка авторизации с валидного аккаунта kash 123456:



Авторизация успешна:



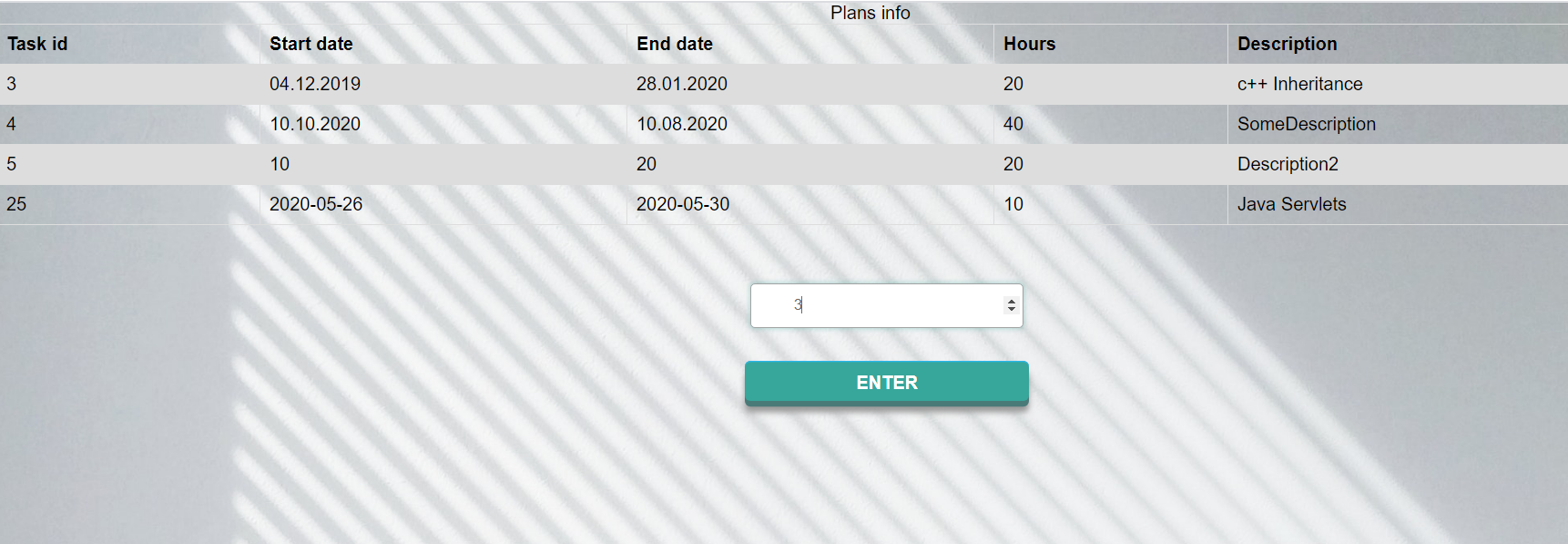
## Acceptance testing

Проверяем, обеспечивает ли система:

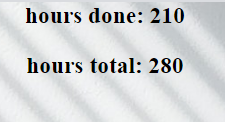
1. Получение доступа к БД планов преподавателей



1. Добавление и изменение информации о планах преподавателей



1. Возможность расчета итоговой суммы преподавателя



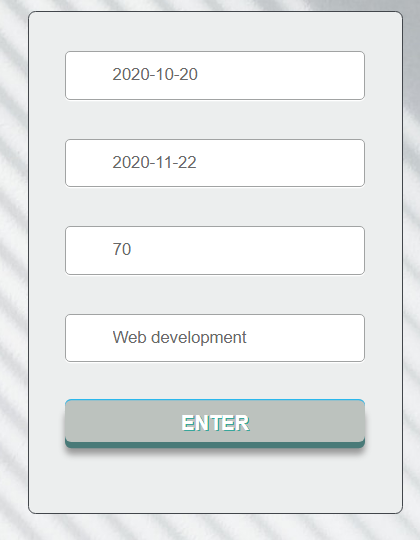
## Scenario testing

Сценарий для претендента “Создать план работы”

* 1. .
  2. “Создает” новый план работы. Во всплывающем окне преподаватель вводит данные о своем плане. Часы отведенные на план не могут превышать часы отведенные на преподавателя.
  3. Идет проверка ввода на валидность.
  4. При невалидных данных , система потребует их повторного ввода.

Тестирование:

Преподаватель вводит имя, даты, описание, количество часов отведенные на план (20.10.2020, 22.11.2020, 70, “Web development”).  
Нажимает на кнопку “Enter”   
После в окне всех задач можно посмотреть наличие новой задачи



Проверяем наличие новой задачи в общем списке:



Все тесты пройдены успешно.