МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-40 05 01 Информационные системы и технологии

Направление специальности 1-40 01 02 03 Информационные системы

и технологии (издательско-полиграфический комплекс)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Применение технологии Oracle Database Vault для разграничения доступа в базе данных расписания занятий в ВУЗе»

Выполнил студент Полынская Ю. В.

(Ф.И.О.)

Руководитель проекта ассистент Колмаков М.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: ассистент Колмаков М.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: ассистент Колмаков М.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2020

Содержание

[Введение 3](#_Toc41584453)

[1. Аналитический обзор аналогов 4](#_Toc41584454)

[2.Анализ требований к программному средству и разработка функциональных требований 10](#_Toc41584455)

[3.Разработка архитектуры проекта 11](#_Toc41584456)

[3.1 Обобщенная структура 11](#_Toc41584457)

[3.2 Диаграммы UML, взаимосвязь всех компонентов 13](#_Toc41584458)

[4. Разработка модели базы данных 15](#_Toc41584459)

[5*.* Установка, настройка и использование Oracle Database Vault 16](#_Toc41584460)

[5.1 Установка ODV 16](#_Toc41584461)

[5.2 Создание таблиц 19](#_Toc41584462)

[5.3 Создание Realms для запрета доступа 20](#_Toc41584463)

[5.4 Проверка работоспособности созданных Realms (областей) 23](#_Toc41584464)

[6. Описание технологии 24](#_Toc41584465)

[7. Тестирование, проверка работоспособности и анализ полученных результатов 25](#_Toc41584466)

[8. Руководство по использованию программного средства 27](#_Toc41584467)

[Заключение 32](#_Toc41584468)

[Cписок источников 33](#_Toc41584469)

## **Введение**

Век высоких технологий диктует нам все больше новых тенденций и максимально оптимизирует процессы, которые раннее занимали приличную часть времени. Общество все больше использует технологии в повседневной жизни и вряд ли сможет отказаться от них. Нас же интересует сфера образования.

Важным направлением в развитии образования является разработка и внедрение единой системы электронного документооборота. Сервисы «Электронный журнал/дневник» позволяют организовать индивидуальный учет результатов освоения обучающимися образовательных программ в электронном виде, а также хранение в архивах данных об этих результатах. Такие журналы позволяют в кратчайшие сроки делиться изменениями и обновлениями данных, что способствует повышению продуктивности.

Уже многие образовательные учреждения перешли на электронные системы хранения информации. Как пример можно привести такие университеты как: БГУ, ГрГУ, ГГМУ и другие.

Так же во многих статьях, посвященных внедрению информационных технологий в учебный процесс, говориться о том, что «…Для студента основным инструментом формирования его информационной культуры, получения им необходимых общекультурных и профессиональных компетенций стали информационные, облачные и интернет-технологии. Для преподавателя, практически решающего задачу объективной оценки учебных достижений студента, важными элементами современного учебного процесса стали методы компьютерного тестирования, балльно-рейтинговая система оценивания и система зачетных единиц. Эффективное использование этих систем возможно с помощью нового инструмента – электронного журнала.»

Актуальность разработки и применения в вузе электронного журнала определяется следующими факторами:

* балльно-рейтинговая система;
* требование к уровню подготовки студентов и процедуры их оценки должны быть открытыми и понятными для студентов, их родителей и будущих работодателей;
* оценка учебных достижений студента на всех промежуточных этапах обучения по курсу.

В современных системах электронных журналов обычно реализованы технологии личных кабинетов с различным набором функций, зависящих от прав доступа и той роли, которую выполняет пользователь (преподаватель, студент, администратор.

Цели проекта: формирование навыков проектирования баз данных, проектирования и разработки пользовательского интерфейса.

Задача проекта: разработать архитектуру приложения, создать интерфейс, взаимодействие с которым будет понятно пользователю любого возраста. Построить базу данный и интерфейс основываясь на обзоре аналогов и выполнить тестирование готового продукта.

# **1. Аналитический обзор аналогов**

**1.1. Журнал преподавателя**

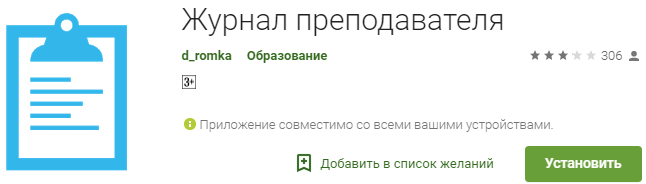


Рисунок.1.1 – Иконка аналога приложения

Программа предназначена для преподавателей университетов. Предоставляет возможность вести журналы посещаемости и успеваемости учащихся.

Дополнительные возможности:

* настраиваемая сводная ведомость;
* студенты могут быть зачислены в несколько групп/подгрупп одновременно;
* настройка графического представления журнала (таблицы);
* импорт записей о студентах/группах из файлов;
* сохранение/восстановление базы данных (SD-card).
* графа "Среднее значение";
* синхронизация с календарем (событие календаря должно содержать название группы, а также аббревиатуры дисциплины и вида занятия).

Интерфейс:

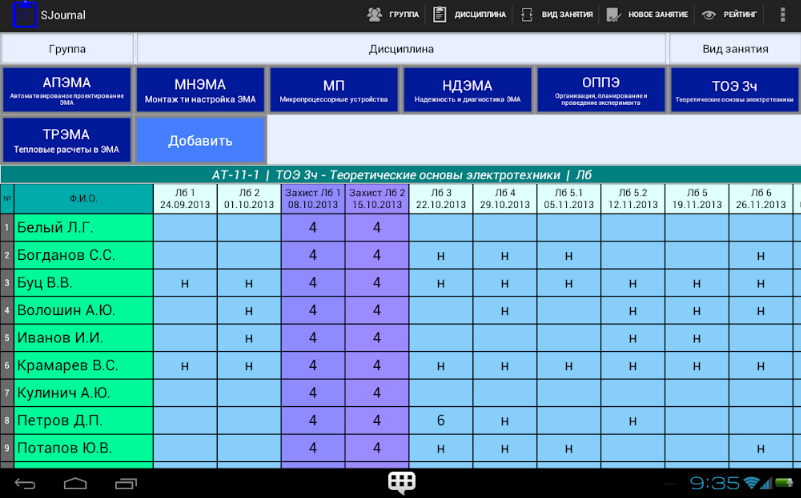


Рисунок.1.2 – Страница со списком группы и выводом дисциплин

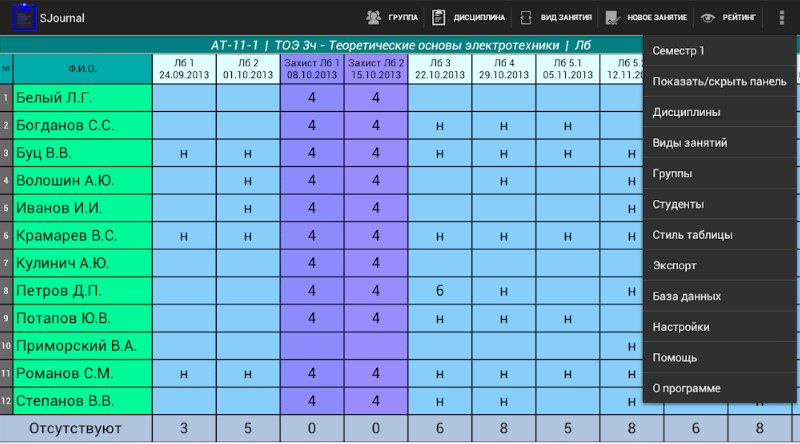


Рисунок.1.3 – Страница со списком группы и выводом возможных действий пользователя

Вывод: Визуально приложение не привлекает. По функционалу тоже есть вопросы. Учитывая отзывы пользователей, могу сказать, что приложение не примитивное и не совсем адаптированное для комфортной работы.

**1.2. iTeacherBook-журнал преподавателя**

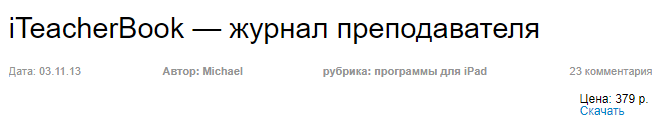


Рисунок.1.4 – Иконка аналога приложения

Красивое и простое приложение поможет сохранить расписание и задания на неделю в удобной форме.

Особенности приложения:

* простое и удобное расписание;
* множество настроек, включая дни недели и количество занятий;
* возможность сохранять задания и заметки;
* создание заданий и прописывание их атрибутов;
* выставление баллов за задания;
* отслеживайте процент посещаемости;
* вы будите видеть свое расписание на текущую неделю, причем можно сделать так, чтобы в нем отображались события из системного календаря;
* можно задавать праздники и правила переноса занятий;
* в режиме текущего дня приложение автоматически показывает идущее сейчас занятие.

Интерфейс:

Осталось создать расписание, то есть для каждого курса указать время и место проведения занятий. Это очень похоже на обычную работу с календарными событиями. Занятия могут быть разных типов, периодичность настраивается без проблем (раз в неделю, раз в две недели — как хотите), допускаются некоторые оформительские элементы. В настройках приложения можно указать продолжительность занятия, а заданный для семестра временной интервал укажет, сколько именно повторений будет создано.



Рисунок.1.5 – Процесс создания расписания

Здесь можно отметить посещаемость. Предусмотрены статусы «присутствовал» и «отсутствовал», а также два расширенных: «опоздал» и «отсутствовал по уважительной причине». Возможно, некоторым преподавателям не хватит статуса «присутствовал, был у доски»

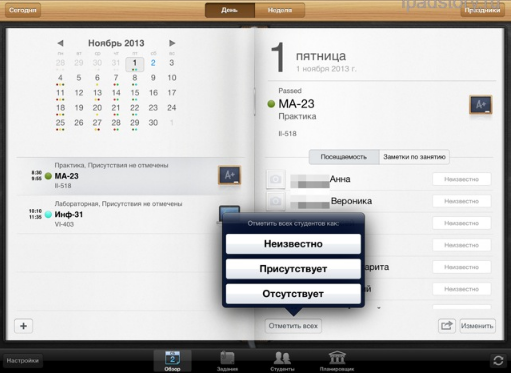


Рисунок.1.6 – Отметка посещаемости

Ну и самая главная задача преподавательского журнала — учёт студенческих успехов. Это делается посредством заданий, под каковое понятие подпадает всё, что подлежит оцениванию. Контрольные работы, домашние задания, индивидуальные задания, лабораторные работы, коллоквиумы, зачёты, экзамены.

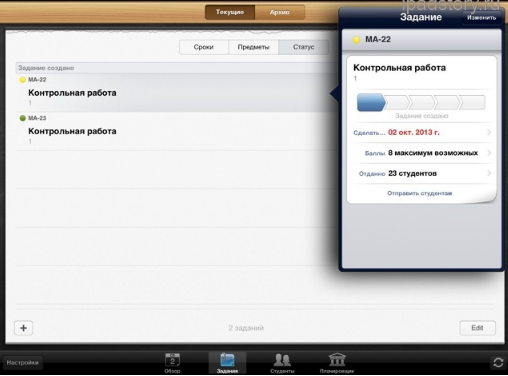


Рисунок.1.7 – Создание заданий

Вывод: хорошое приложение для контроля посещаемости и успеваемости. Комфортное и примитивное в использовании.

**1.3. Magellan – система управления учебным процессом**



Рисунок.1.8 – Иконка аналога приложения

Особенности приложения:

* понятный и удобный пользовательский интерфейс;
* наличие встроенного инструмента по созданию и редактированию печатных форм;
* возможность формирования всех журналов регистрации справок, документов, приказов, гибкая система документооборота;
* **вы сможете гибко настраивать права доступа** для преподавателей, методистов и администраторов;
* **с помощью большого количества встроенных отчётов вы легко получите картину по обучающимся;**
* отличная работоспособность с огромным массивом данных.

Интерфейс:

**Электронный журнал успеваемости** позволит вам хранить и редактировать данные по текущей успеваемости и посещаемости обучающихся, формировать на их основе фактически выполненную аудиторную нагрузку преподавателей и рейтинги обучающихся, выявлять хронических прогульщиков.

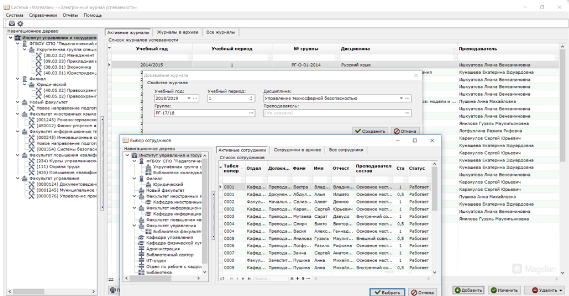


Рисунок.1.9 – Создание электронного журнала – закрепление преподавателя за журналом

**С помощью модуля вы сможете создавать и выбирать типы занятий (лекции, практические, лабораторные, зачеты, экзамены).**

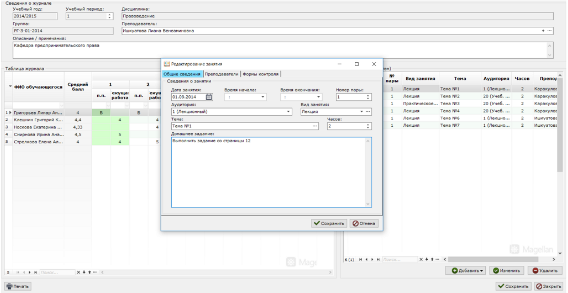


Рисунок.1.10 – Окно редактирования занятий в электронном журнале

Кроме того, у вас появятся [личный кабинет обучающегося](https://magellanius.ru/studentboard/), в котором обучающийся может посмотреть свои оценки, и [личный кабинет сотрудника](http://magellanius.ru/teacherboard/), где преподаватель может редактировать доступные ему журналы успеваемости, создавая новые занятия, указывая успеваемость и посещаемость.



Рисунок.1.11 – Окно редактирования занятий в веб-версии модуля

Вывод: Приложение простое в использование. Функционал минимальный, но оптимальный для студентов и преподавателей. Ничего лишнего.

Обобщение: данные приложения являются аналогами того, что мы будем создавать. Но просмотрев функционал и отзывы пользователей можно взять на заметку не перегруженный, аккуратный и понятный интерфейс, адаптивный функционал и идеи по оптимизации данных.

Требования к электронному журналу:

* возможность ведения учета успеваемости каждого студента вуза;
* разграничение прав доступа к базе данных для различных пользователей;
* возможность анализа успеваемости студента, группы;
* интерфейс приложения должен обеспечивать простой и понятный способ заполнения электронного журнала.

# **2. Анализ требований к программному средству и разработка функциональных требований**

Разрабатываемое приложение должно содержать форму авторизации. Следует разработать два уровня доступа: уровень доступа для преподавателей и уровень доступа для студентов. Преподавателю дополнительно предоставляется возможность добавления студентов, дисциплин, оценок, пропусков и изменения расписания.

Форма реализации продукта – десктоп приложение.

Десктоп приложение должно быть реализовано с использованием языка С#. Технология для реализации – WPF. База данных приложения должно взаимодействовать с Oracle Database 11g.

В приложении требуется организовать систему для взаимодействия преподавателя и студентов, с возможностью контроля своей успеваемости. Необходимо построить удобную систему, содержащую личные данные преподавателя/студента, а так же информацию о посещении дисциплин и успеваемости по дисциплинам, так же возможность вносить изменения в расписание (переносы, отмены занятий).

Сделать возможность редактирования личных данных.

При нажатии на знак меню (в верхнем правом углу) должна открываться панель с расшифровкой обозначений меню.

# **3. Разработка архитектуры проекта**

## **3.1 Обобщенная структура**

Решение проекта представлено одним проектом Курсовая, имеющим структуру, представленную в таблице 3.1.

Таблица 3.1

|  |  |
| --- | --- |
| Составляющие проекта | Описание |
| Properties | Свойства проекта, содержит информацию о сборке, используемых ресурсах и настройках |
| Ссылки | Перечень сборок, используемых в проекте |
| Properties | Папка со свойствами проекта |
| App.config | Файл с параметрами проекта |
| App.xaml | Класс Application |
| GroopAdd.xaml | Окно добавления группы |
| StudentAdd.xaml | Окно добавления студента |
| SubjAdd.xaml | Окно добавления дисциплины |
| Папка Database | Все классы таблиц, репозитории и их интерфейсы |
| BaseViewModel.cs | Базовый класс для всех viewmodel |
| EnumHelper.cs | Класс перечислений |
| RelayCommand.cs | Класс для обработки событий или команд |
| Services | Папка с сервисами |
| Res | Папка с ресурсами |
| StudentControl | Папка с контролами для взаимодействия студента с приложением и преподавателем |
| GroupWindowViewModel.cs | Класс viewmodel для контрола заполнения групп |
| UserControlPeople.xaml | Контрол, содержащий информацию о группах с кнопками для изменения данных |
| LoginWindow.xaml | Окно входа/регистрации |
| LoginWindowViewModel | Класс viewmodel для авторизации |
| RegistrationWindow.xaml | Окно регистрации |
| WindowRegistrViewModel.cs | Класс viewmodel для регистрации |
| MainWindow.xaml | Начальная страница приложения |
| MainWindowViewModel.cs | Класс viewmodel для начальной страницы |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| Составляющие проекта | Описание |
| Папка UiCommon | Класс который содержит методы конвертирования объектов базы данных в объекты представления |
| UserControlUser.xaml | Страница профиля |
| UserViewModel.cs | Класс viewmodel для страницы профиля |
| NewFacultyWindow.xaml | Окно добавления факультета |
| NewFacultyViewMode.cs | Класс viewmodel для окна добавления факультета |
| NewDepartmentWindow.xaml | Окно добавления специальности |
| NewDepartmentViewModel.cs | Класс viewmodel для окна добавления специальности |
| UserControlBook.xaml | Страница добавления дисциплин |
| UserControlCalendars.xaml | Страница добавления расписания |
| UserControlCheckboxesMarked.xaml | Страница пропусков и оценок |
| BaseUser.cs | Класс для добавления пользователей |
| ClassDiagram1.cd | Класс диаграмм |

## **3.2 Диаграммы UML, взаимосвязь всех компонентов**

Диаграмма UML – это графическое представление выбора элементов, изображаемое чаще всего в виде графа с вершинами (сущностями) и рёбрами (отношениями). Диаграммы рисуют для визуализации. Основная цель диаграмм – визуализация разрабатываемой системы с разных точек зрения.

На диаграмме 3.2.1 представлены принципы работы приложения с точки зрения студента и преподавателя соответственно:

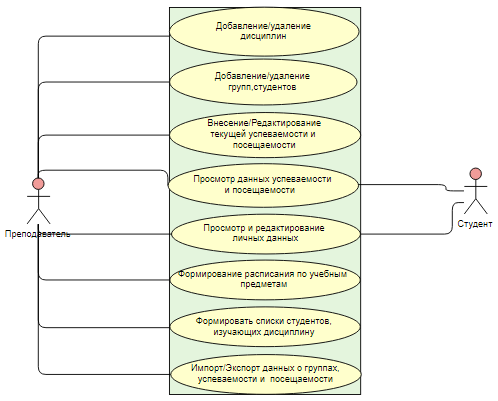


Рисунок 3.2 – Принцип работы студента и преподавателя

Данное приложение предоставляет минимальные требования к выполнению функций приложения и выполняет следующие функции:

* регистрация с последующей авторизацией пользователя;
* возможность изменения личных данных пользователя;
* добавление и удаление студентов в группах;
* импорт записей о студентах/группах из файлов;
* формировать расписание по учебным предметам;
* сохранять, изменять и удалять данные;
* формировать списки студентов, изучающих дисциплину;
* вести учет посещения занятий;
* вести учет cдачи работ;

# **4. Разработка модели базы данных**

Первым этапом курсовой работы будет создание логически взаимосвязанных таблиц. Чтобы составить визуальную взаимосвязанную структуры нашей базы данных, нам необходимо продумать, какая информация будет храниться в этих таблицах, после этого создать связи с помощью первичных и внешних ключей.

База данных состоит из 13 таблиц, перечисленных ниже (рисунок 4.1):

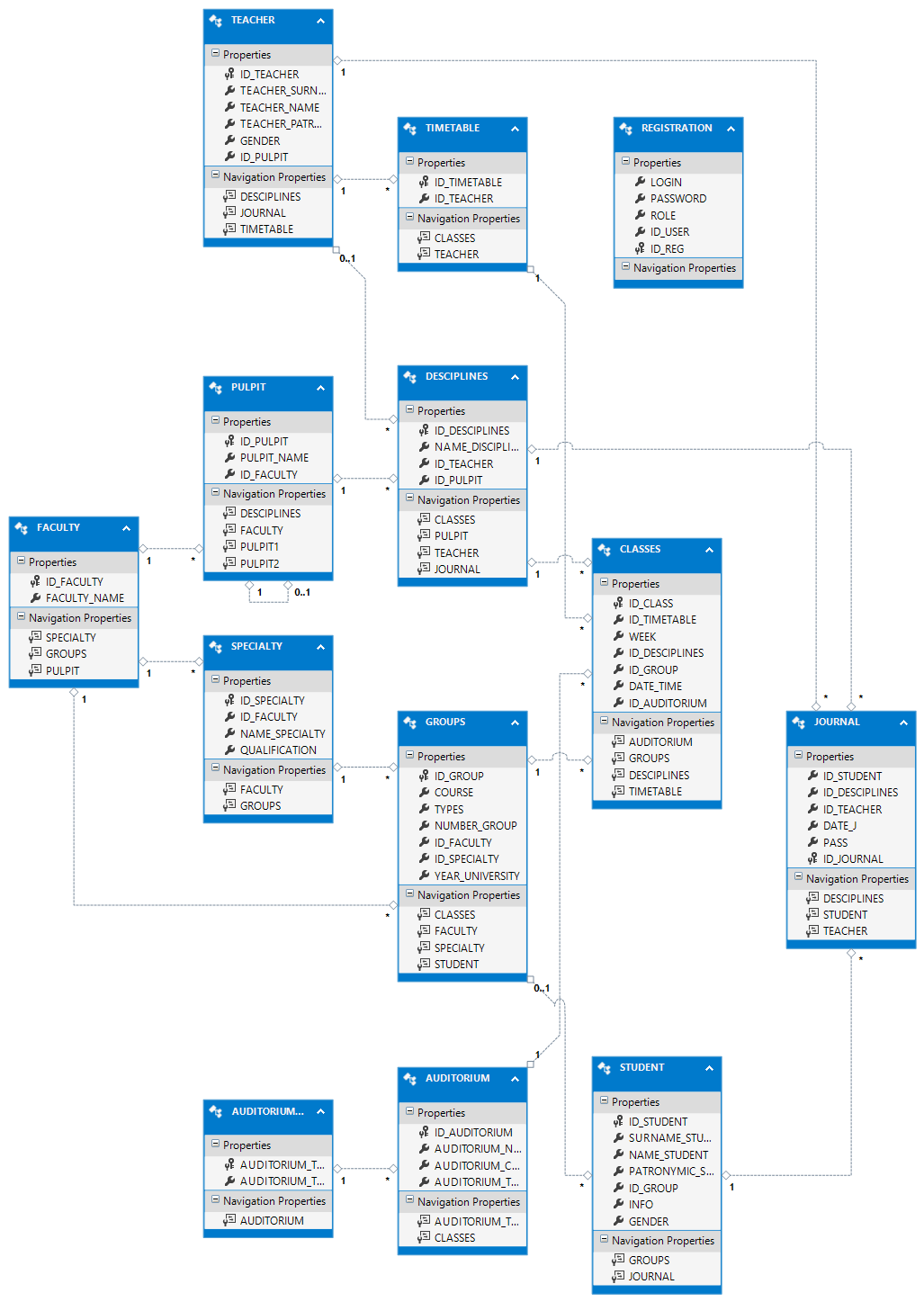


Рисунок 4.1 – Взаимосвязь таблиц данных

# **5*.*** **Установка, настройка и использование Oracle Database Vault**

## **Установка ODV**

Чтобы установить Oracle Database Vault (в дальнейшем ODV) сперва необходимо установить сам Oracle Database на виртуальную машину, которая будет являться сервером. Вся установка осуществляется со стандартными конфигурациями, но на этапе выбора типа установки, необходимо выбрать тип advanced install, это позволит добавить в пакет устанавливаемых компонентов нашу технологию ODV.



Рисунок. 5.1 – Выбор компонентов Oracle Database

Следующие этапы установки включают в себя настройку CDB и пользователей SYS и SYSTEM (задание паролей для авторизации).

После завершения установки необходимо перезагрузить ОС. После перезапуска необходимо запустить OCA (Oracle Configuration Assistant), именно тут мы будем задавать параметры установки ODV

На первом шаге необходимо выбрать пункт Configure Database Option.

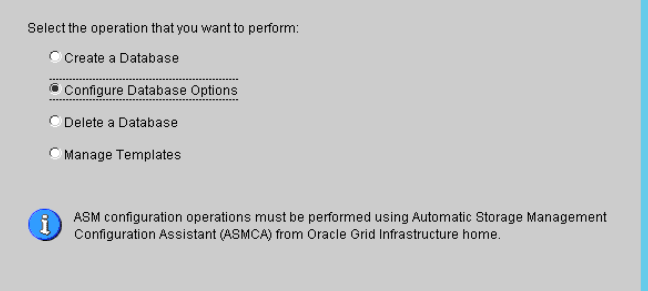


Рисунок. 5.2 – Выбор операции

На шаге номер 2 мы выбираем БД на которой будем осуществлять настройку ODV. Также необходимо ввести данные пользователя, которому назначена роль DBA.

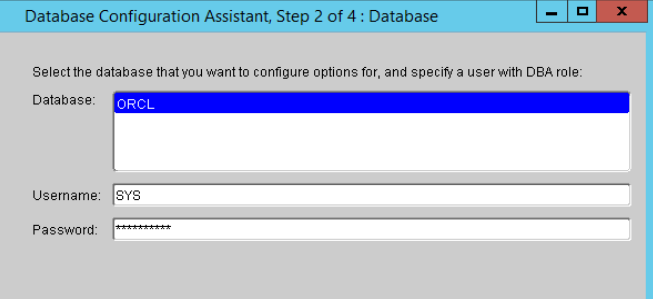


Рисунок. 5.3 – Выбор БД

После выбора БД мы должны выбрать компоненты, которые желаем установить, а именно ODV и Oracle Label Security.

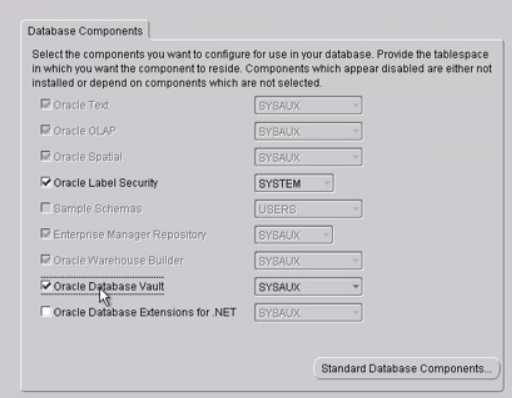


Рисунок. 5.4 – Выбор компонентов

На следующем этапе необходимо создать двух пользователей, которые будут управлять данной технологией: OWNER – создает запреты и правила доступа, MANAGER – заменяет OWNER при его бездействии и управляет созданием пользователей.

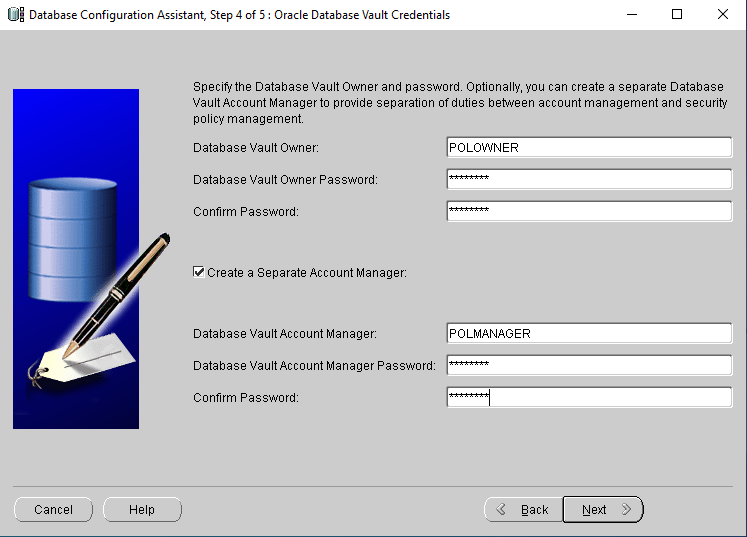


Рисунок. 5.5 – Регистрация пользователей

После этого этапа нажимаем кнопку Finish и ожидаем окончания установки. По завершению операции необходимо перезапустить ОС для нормальной работы.

После перезапуска ОС нам необходимо открыть любой из доступных браузеров и в адресной строке ввести следующее: <http://hostname:port/dva>, где:

* hostname – имя сервера DB
* port – порт, который можно узнать, установив DB

После этих манипуляций у нас откроется стартовая страница для работы с Oracle Database Vault.

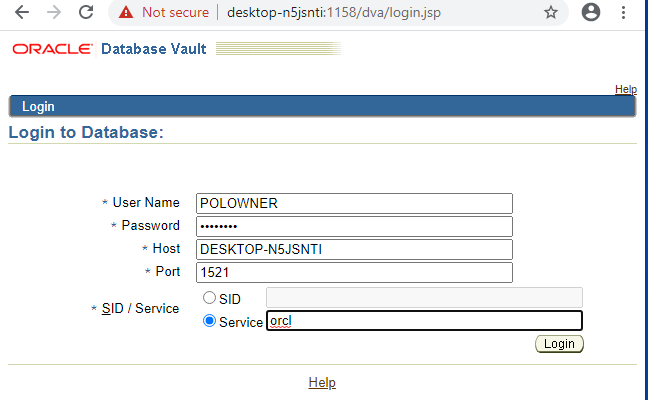


Рисунок.5.6 – Авторизация в ODV

На этой странице необходимо ввести данные для авторизации под одним из выше созданных пользователей (OWNER или MANAGER). Авторизация позволит получить доступ к настройке ресурса.



Рисунок.5.7 – ODV

Как видим компонент ODV успешно работает, но для того, чтобы начать его применять, необходимо в нашей базе данных создать различные объекты.

## **Создание таблиц**

Перед работой с ODV мы должны создать таблицы и организовать между ними связи, которые будут представлять логическую структуру БД журнала. В нашу базу будут включены такие таблицы:

1. регистрация,
2. группы,
3. дисциплины,
4. преподаватели,
5. журнал,
6. занятия,
7. кафедры,
8. расписание,
9. специальности,
10. студенты,
11. аудитории,
12. типы аудиторий,
13. факультеты.

Создание этих таблиц проходит через консольное приложение SQL Plus. Связи между таблицами будут прописываться тут же.

## **Создание Realms для запрета доступа**

После установка и настройка ODV и созданных нами таблиц, мы можем приступить к созданию запретов в нашей технологии. Для этого на сайте необходимо выбрать пункт меню REALMS (области)



Рисунок.5.8 – Realms

После выбора мы перейдем на страницу с созданными областями. Как видим, при установке технологии тут уже присутствуют некоторые области с запретами. Эти области можно редактировать (кнопка edit), можно удалить (remove), т.к. мы имеем права пользователя OWNER.

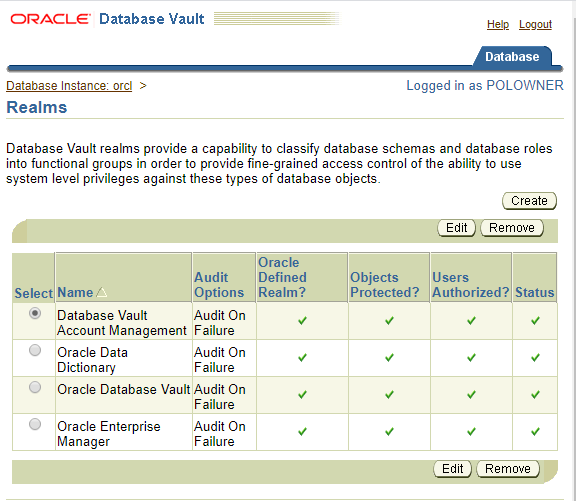


Рисунок.5.9 – Созданные области

Чтобы создать собственный realm, необходимо нажать на кнопку CREATE, это позволит перейти на страницу создания и настройки областей.

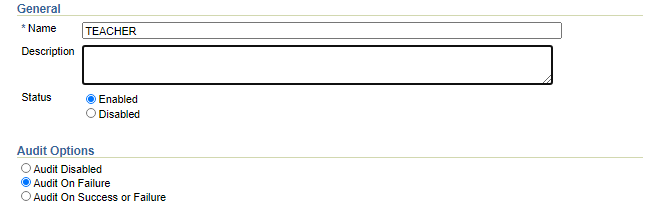


Рисунок.5.10 – Базовые настройки

После указания имени области нажимаем кнопку OK и переходим к ее редактированию. В редактировании области у нас доступно две таблицы для создания охраняемых объектов и для пользователей, которые имеют доступ к охраняемым объектам.



Рисунок.5.11 – Редактирование области

Указываем имя объекта для защиты и тип объекта. В последнее поле вводим имя выбранного объекта и нажимаем OK.

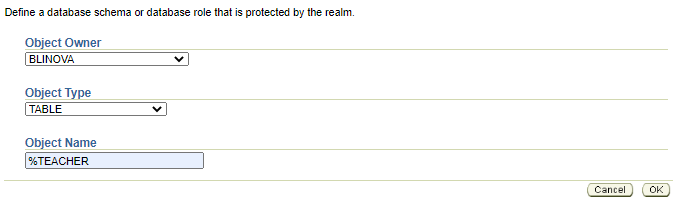


Рисунок.5.12 – Охраняемый объект

После создания объекта выбираем пользователя, который будет иметь доступ к этому объекту БД.

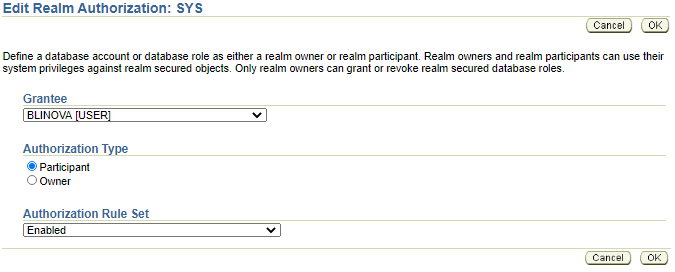


Рисунок.5.13 – Владелец

После нажатия клавиши OK последней манипуляции мы увидим, что созданный realm создан и подключен (имеет статус положительный)

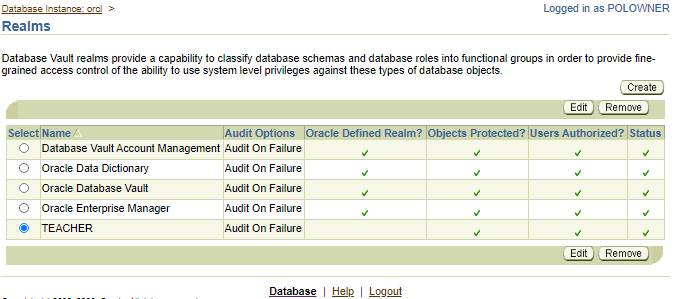


Рисунок.5.14 – Созданный realm

Чтобы удалить созданные запреты, необходимо авторизоваться в ODV под одним из пользователей OWNER или MANAGER и перейти в раздел Realms. Удаление будет выполнено если нажать кнопку remove, под таблице realms.

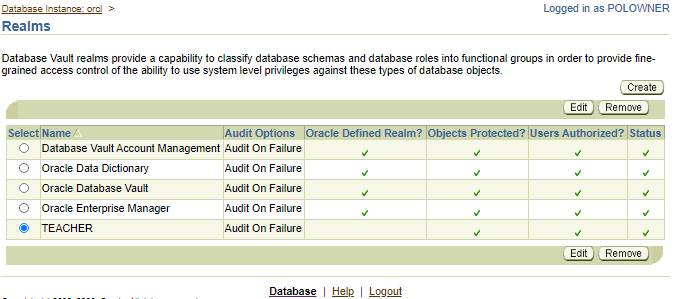


Рисунок.5.15 – Удаление области

## **Проверка работоспособности созданных Realms (областей)**

После завершения установки технологии ODV (Oracle Database Vault) необходимо проверить работоспособность запретов, которые были настроены. Чтобы выполнить проверку, мы переходим в программу Database Developer и выполнить любой запрос к таблице, на которую был настроен realm. Но перед этим нужно авторизоваться под обычным пользователем. Защищенной таблицей данном примере будет являться таблица с именем TEACHER.

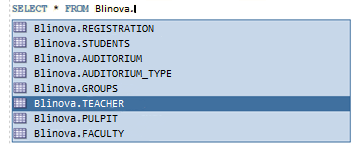


Рисунок.5.16 – Созданный realm

После некоторого промежутка времени у нас появляется исключение, которое показывает, что у данного пользователя недостаточно привилегий для просмотра этой таблицы.

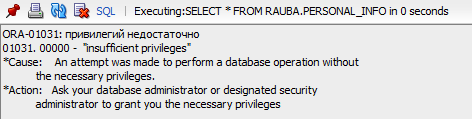


Рисунок.5.17 – Недостаточно привилегий на Select

Realms можно создавать практически на любые объекты базы данных (таблицы, процедуры, функции, представления, правила, роли и т.д.). Это позволит обезопасить любую БД, на которой установлен ODV, от несанкционированного доступа вне зависимости от назначенных привилегий.

# **6. Описание технологии**

Что такое Oracle Database Vault?

Oracle Database Vault – настройка безопасности БД Oracle, которая позволяет контролировать или исключать доступ администратора СУБД к данным приложений, усиливать и контролировать защиту структур СУБД от несанкционированных действий персонала и реализовать динамическую настройку политик безопасности. Данные средства имеют основу на усовершенствованном механизме ролевого доступа и позволяют обеспечивать выполнение нормативных требований и стандартов внутреннего аудита.

Привилегированные учетные записи являются одним из наиболее часто используемых путей для получения доступа к конфиденциальным данным приложений в БД. Хотя их широкий и неограниченный доступ облегчает обслуживание базы данных, такой доступ может создавать точку атаки и получения доступа к большим объемам данных.

Oracle Database Vault ограничивает доступ к определенным областям в базе данных Oracle от любого пользователя, включая пользователей, имеющих административный доступ. Например, вы можете ограничить административный доступ к заработной плате сотрудников, банковской информации клиентов или другой конфиденциальной информации.

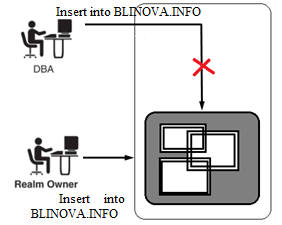


Рисунок 6.1 – Схема работы ODV

Это позволяет вам применять точечный контроль доступа к вашим конфиденциальным данным различными способами. Самое главное, он защищает ваши данные от супер-привилегированных пользователей, но все же позволяет им поддерживать БД. Oracle Database Vault является неотъемлемой частью вашего предприятия.

С Oracle Database Vault мы можем решить самые сложные проблемы безопасности, которые являются на сегодняшний день основными: защита от внутренних угроз, соблюдение нормативных требований и обеспечение разделения обязанностей.

# **7. Тестирование, проверка работоспособности и анализ полученных результатов**

Для оценки работоспособности разработанного приложения были проведены тесты использование памяти и использование ЦП.

Тестирование использование памяти проводится с целью обнаружить проблемы, как утечка памяти.

****

Рисунок 7.1 – Тестирование памяти

Как показано на Рисунке 7.1 тестирование приложения на проблему утечку памяти пройдено успешно. Приложения в процессе своей работы использует 150-190 мб памяти.

Тестирование использование ЦП проводится с целью нахождения степени загруженности центрального процессора при работе программы.

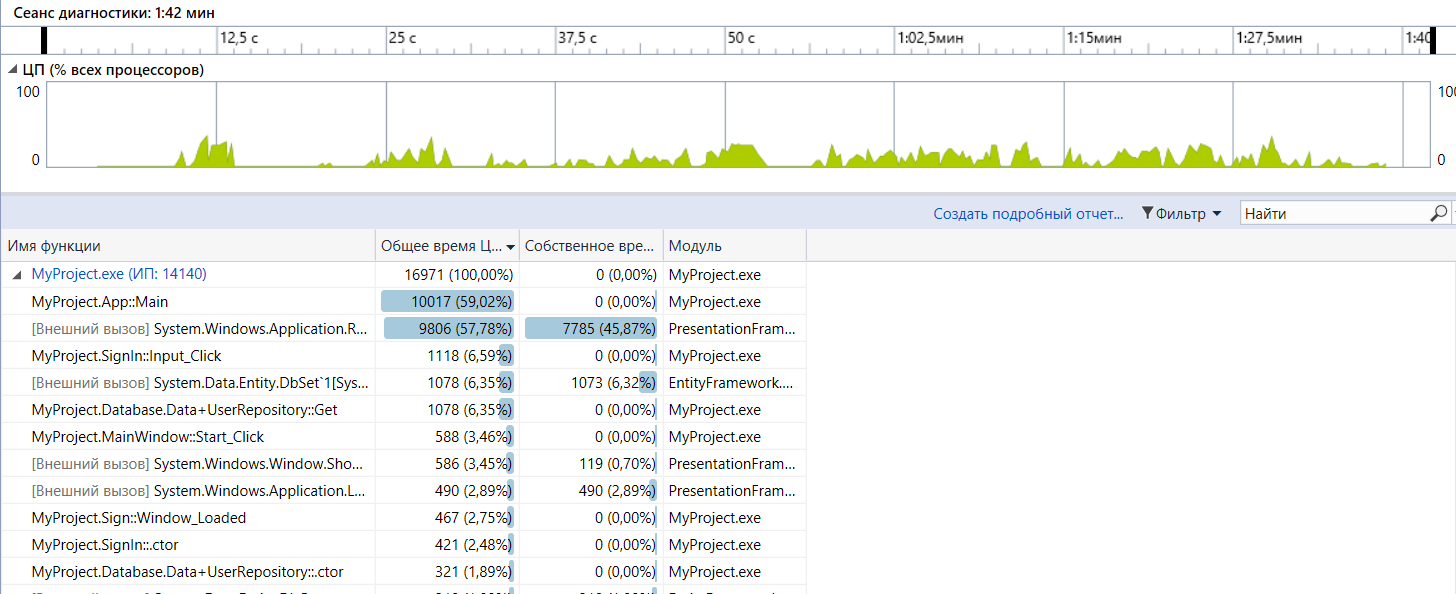


Рисунок 7.2 – Тестирование ЦП

Как показано на Рисунке 7.2 тестирование приложения на загруженность процессора основная нагрузка на приложение (59%) при запуске приложения. Стоит заметить, что нагрузка на процессор не постоянная, а в среднем 0.5-0.9 секунды при том, что все страницы генерируются из базы данных данный результат можно считать хорошим.

На рисунке 7.3 показана временная шкала выполнения процессов в приложении

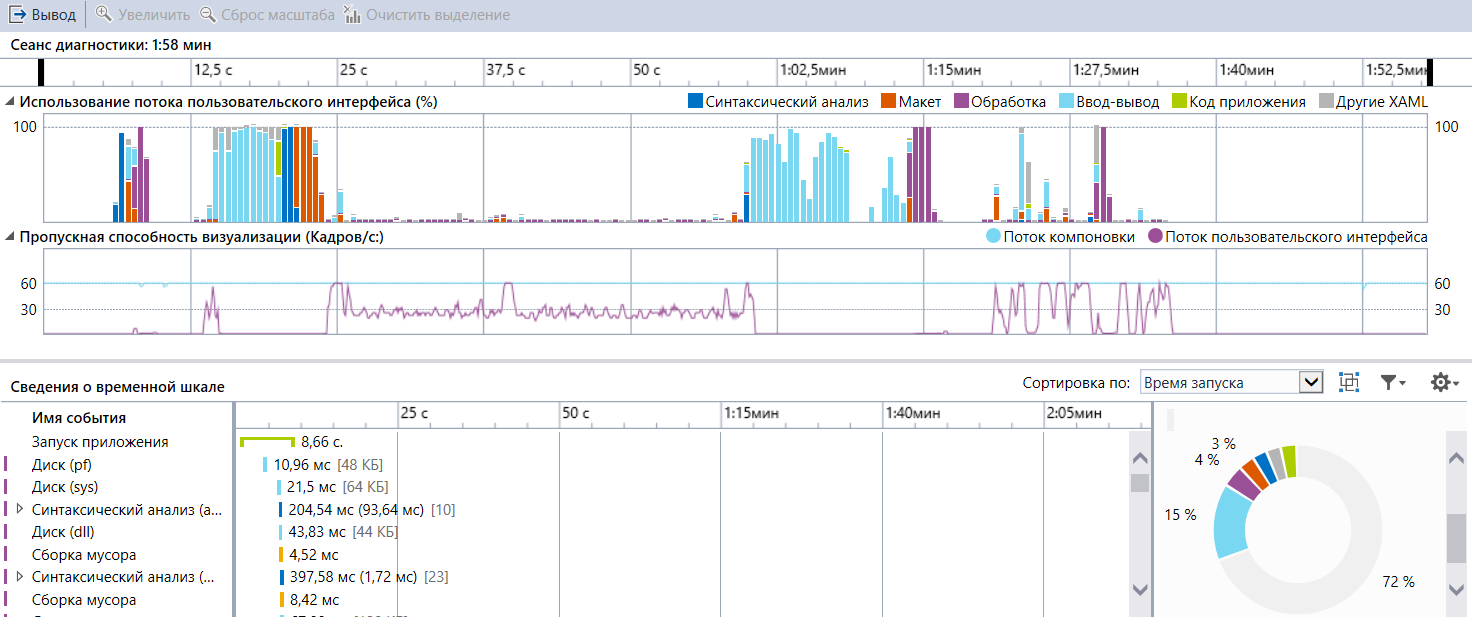


Рисунок 7.3 – Временная шкала

По результату тестирование и получение временной шкалы программного средства можно сделать вывод что из во время работы нашего приложения 72% времени свободно от процессов, что говорит о хорошей оптимизации приложения.

Протестировав приложение можно сделать вывод, что программное средство «Электронный журнал для ВУЗа» является нетребовательным к системным характеристикам приложением и может запуститься практически на любом устройстве с оперативной памятью не ниже 512 мб.

# **8. Руководство по использованию программного средства**

При запуске приложения открывается окно входа/регистрации (рисунок.8.1 и рисунок 8.2):

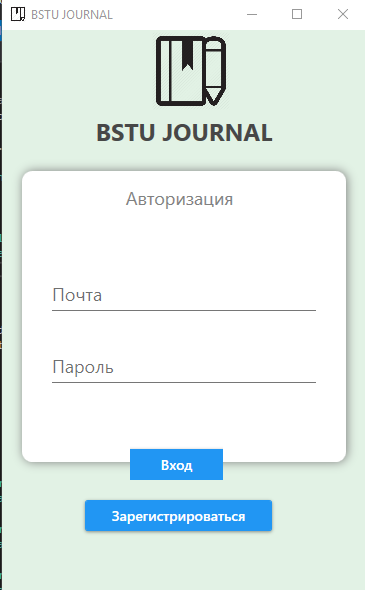
 

Рисунок 8.1 – окно входа Рисунок 8.2 – окно регистрации

После входа/регистрации откроется окно персональной страницы (рис. 8.3):

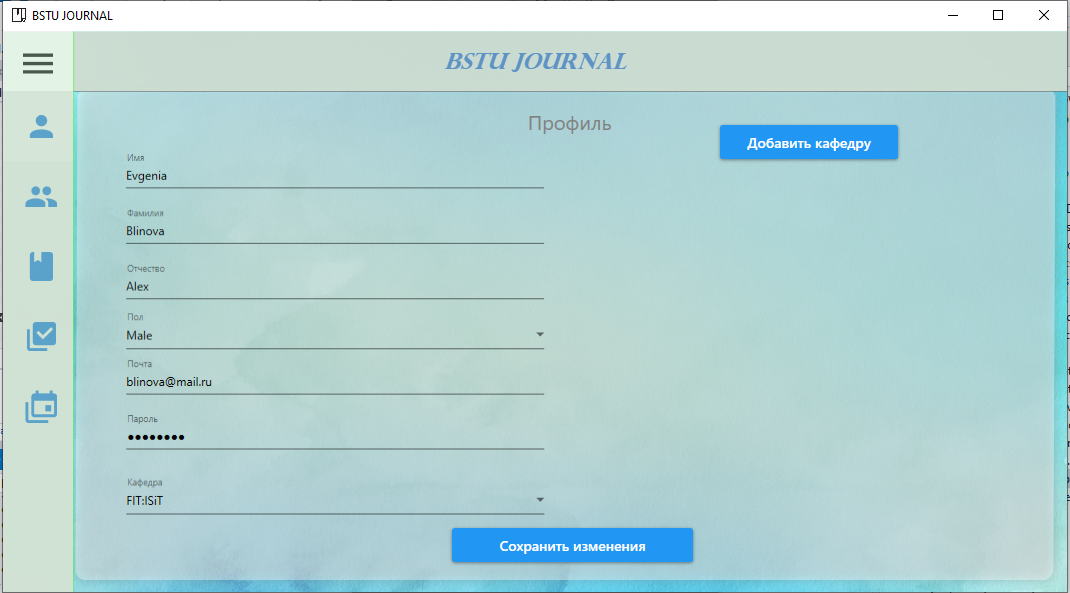


Рисунок 8.3 – профиль преподавателя

Форму можно считать универсальной для университета, так как кафедра и факультет добавляется вручную, далее загружается в бд, а после выбирается в выпадающем меню (рисунок.8.4 и рисунок.8.5):

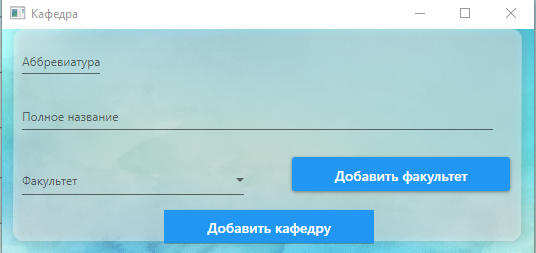


Рисунок 8.4 – окно добавления кафедры

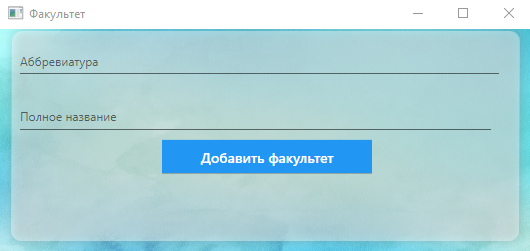


Рисунок 8.5 – окно добавления факультета

Следующий раздел – Группы. В данном разделе мы можем создать группы и добавить/удалить студентов из данной таблицы (рисунок.8.6, рисунок.8.7, рисунок.8.8):

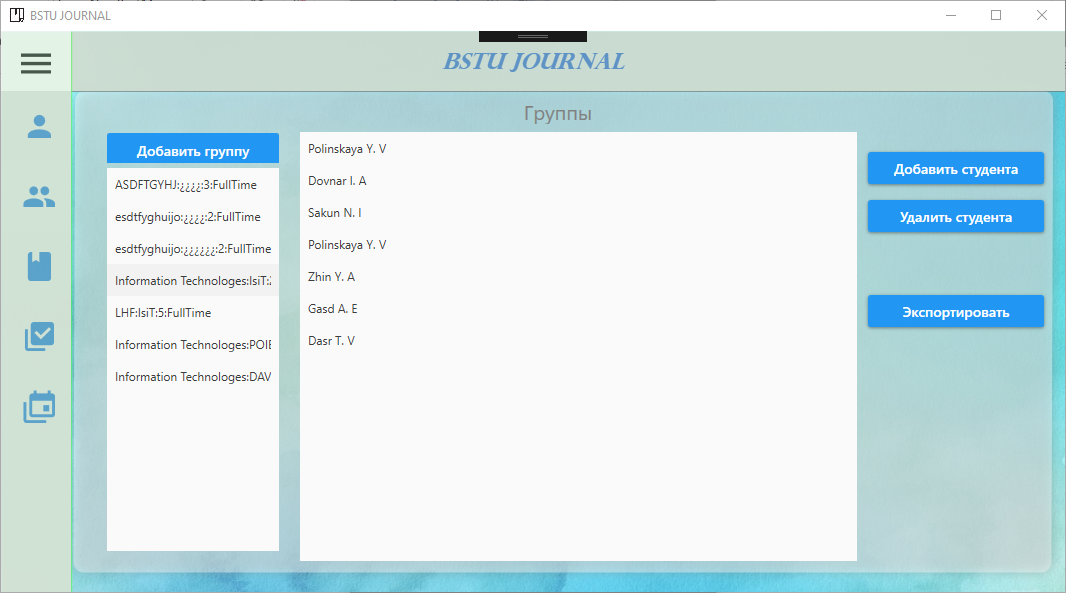


Рисунок 8.6 – Списки группы



Рисунок 8.7 – окно добавления группы

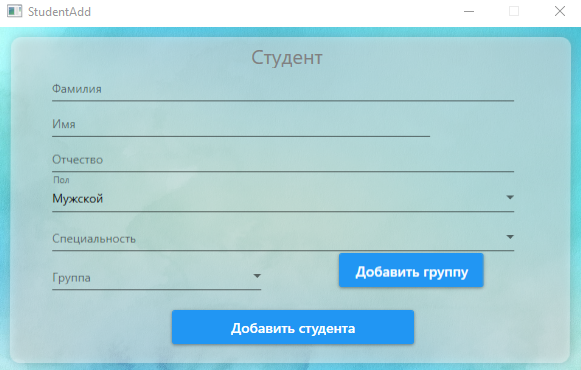


Рисунок 8.8 – окно добавления студентов

Так же для удобства сделан экспорт введенных данных, то есть заполненных раннее форм (рисунок.8.7, рисунок.8.8), которые выводятся в Excel таблицы (рисунок.8.9):



Рисунок 8.9 – Экспорт в Excel

Раздел Дисциплины позволяет преподавателю создать нужные Дисциплины, которые далее тот сможет использовать при создании расписания, а также просмотривать то,что вел (рисунок.8.10):

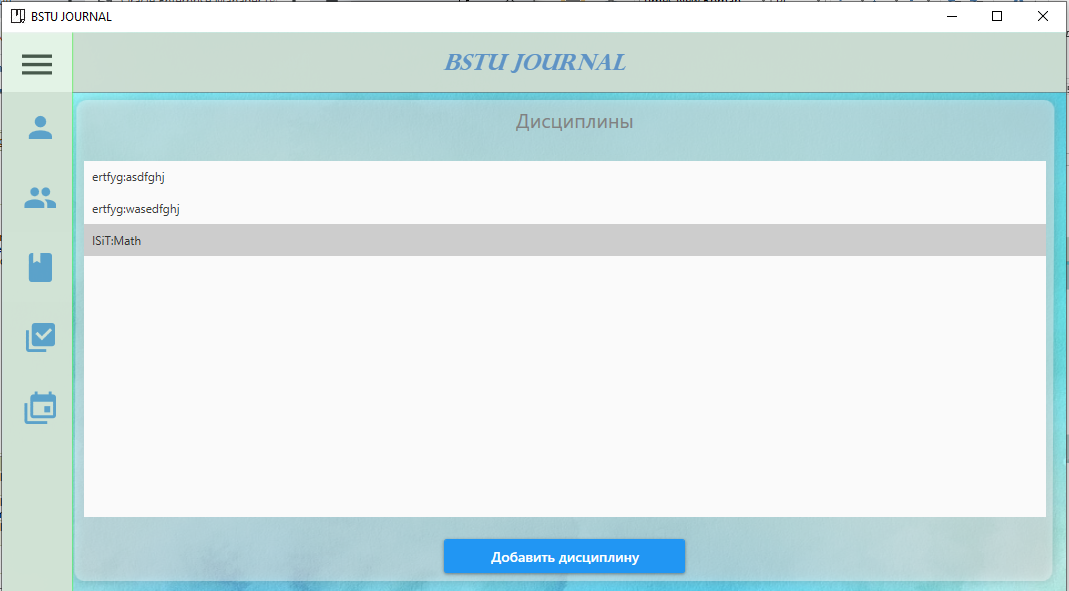


Рисунок 8.10 – Дисциплины

Добавление дисциплин в список (рисунок 8.11):

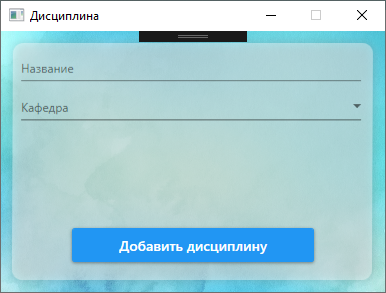


Рисунок 8.11 – Добавление дисциплин

Раздел Успеваемость нужен для того, чтобы преподаватель мог отслеживать посещаемость и успеваемость студентов:

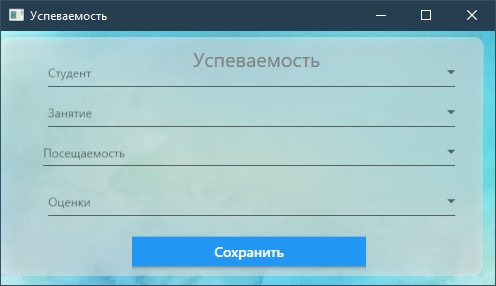


Рисунок 8.12 – Успеваемость и посещаемость

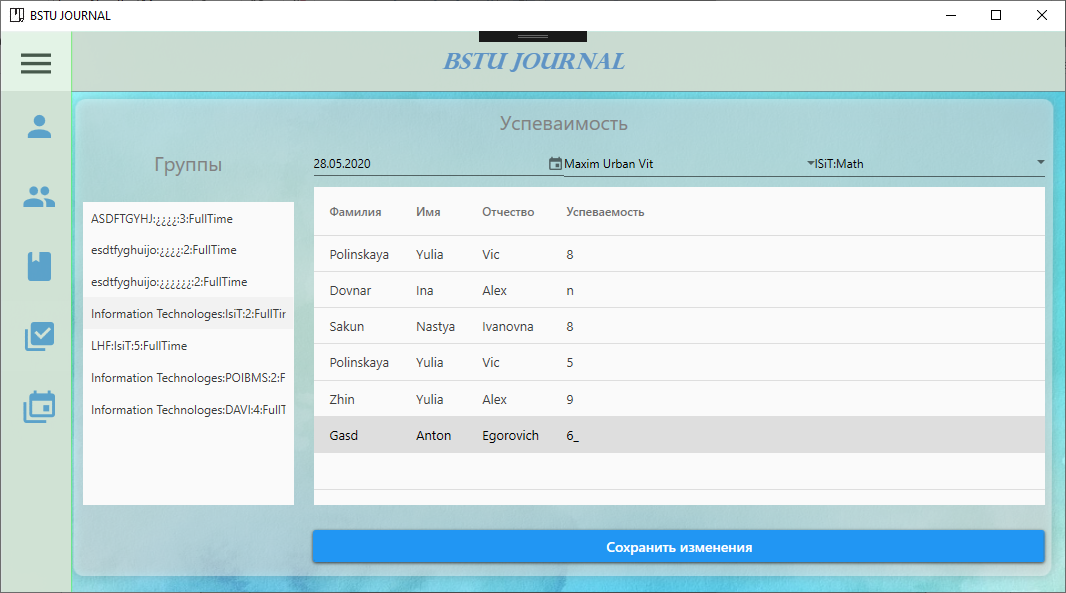


Рисунок 8.13 – Добавление оценки и пропуска

Расписание создается преподавателем для личного контроля, отметки возможных переносов или замен:

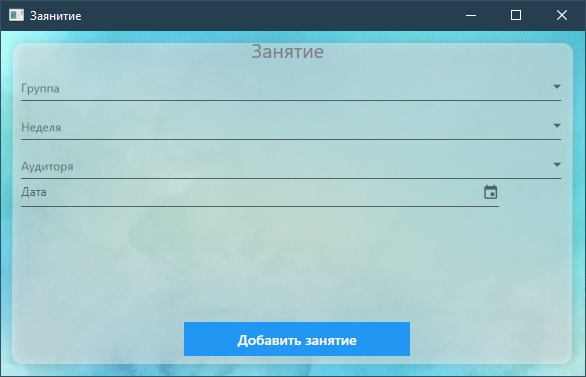


Рисунок 8.14 – Создание расписания

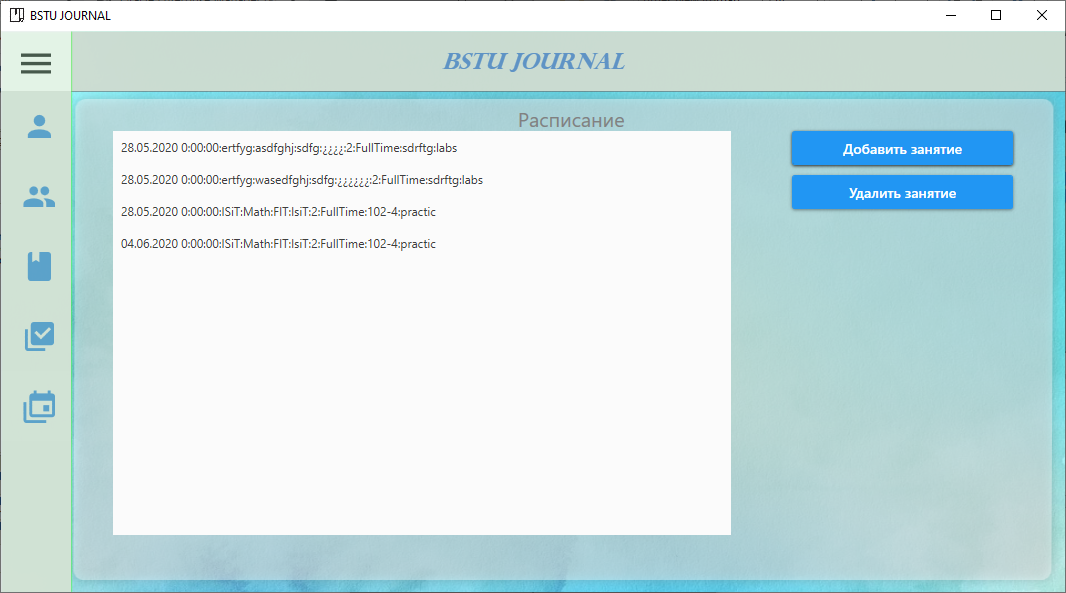


Рисунок 8.15 – Расписание

# **Заключение**

Oracle Database Vault – технология которая усиливает и усложняет защиту данных в любой базе, ограничивает доступ к определенным областям в базе данных Oracle от любого пользователя, включая пользователей, имеющих административный доступ.

С Oracle Database Vault мы можем решить самые сложные проблемы безопасности, которые являются на сегодняшний день наиболее актуальными: защита от внутренних угроз, соблюдение нормативных требований и обеспечение разделения обязанностей.

Сама же технология не мешает работе сервера, однако работа пользователей может быть затруднена некоторыми проблемами с доступом и привилегиями, но это все исправляется, назначив необходимые привилегии. При утере жесткого диска клиента с помощью ODV мы можем заблокировать полный доступ к базе данных. При утере жесткого диска сервера, злоумышленник не сможет получить доступ к технологии, т.к. данные пользователей OWNER и MANAGER защищены усложненным паролем.

В ходе выполнения работы были достигнуты следующие результаты:

* исследованы обучающие приложения-аналоги;
* сформулирована модель данных и приведено её графическое представление;
* разработано руководство по использованию приложения;
* подготовлена база данных Oracle с использованием технологии ODV.

# **Cписок источников**

1. Язык C# и .NET Framework [Электронный ресурс] – http://professorweb.ru/my/csharp/charp\_theory/level1/infonet.php.
2. Руководство по ADO.NET и работе с базами данных [Электронный ресурс] – https://metanit.com/sharp/adonet/.
3. Документация Oracle [Электронный ресурс] / Foundation, Inc. https://docs.oracle.com/cd/B28359\_01/server.111/b31222/toc.htm
4. Официальный сайт Oracle [Электронный ресурс] / Foundation, Inc. https://www.oracle.com/database/database-vault/index.html
5. Продукты Oracle [Электронный ресурс] [Электронный ресурс] / Foundation, Inc. http://www.interface.ru/home.asp?artId=24678
6. Документы Oracle [Электронный ресурс] / Foundation, Inc. http://www.oracle.com/technetwork/database/security/database-vault-ds-12c-1898877.pdf