МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Чистяков Г. А./

(подпись) (Ф.И.О)

«\_\_»\_\_\_\_\_\_\_\_\_\_2016г.

РАЗРАБОТКа Серверной чаСТИ приложенИЯ «МобильНЫй ВЯТГУ»

Пояснительная записка

Курсовая работа по дисциплине

«Разработка программных систем»

ТПЖА 09.03.01.024 ПЗ

Разработал студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д.С./

Руководитель доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

Проект защищен с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Киров 2016

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

1

ТПЖА 09.03.01.024 ПЗ

Разраб.

Ф.И.О.

Провер.

Ф.И.О.

Т. Контр.

Ф.И.О.

Н. Контр.

Ф.И.О.

Утверд.

Ф.И.О.

Разработка серверной части приложения «Мобильный ВЯТгу»

Лит.

Листов

1

ВятГУ

Реценз.

Ф.И.О.

Масса

Масштаб

1

1 : 1

[Введение 3](#_Toc470524391)

[1 Постановка расширенного технического задания 4](#_Toc470524392)

[1.1 Анализ предметной области 4](#_Toc470524393)

[1.2 Анализ аналогов 5](#_Toc470524394)

[1.3 Требования к приложению 9](#_Toc470524395)

[1.4 Требования к программной документации 11](#_Toc470524396)

[1.5 Технико-экономические показатели 11](#_Toc470524397)

[1.6 Стадии и этапы разработки 12](#_Toc470524398)

[1.7 Порядок контроля и приемки 12](#_Toc470524399)

[1.8 Итоги 12](#_Toc470524400)

[2 Проектирование структуры сервера 13](#_Toc470524401)

[2.1 Словарь терминов 13](#_Toc470524402)

[2.2 Разработка базы данных 14](#_Toc470524403)

[2.3 Диаграммы потоков данных 16](#_Toc470524404)

[2.4 Диаграмма модульной структуры 16](#_Toc470524405)

[2.5 Итоги 16](#_Toc470524406)

[3 Реализация программного обеспечения 17](#_Toc470524407)

[3.1 Выбор необходимых библиотек 17](#_Toc470524408)

[3.2 Разработка лексического анализатора для расписания 17](#_Toc470524409)

[3.3 Разработка сервера 18](#_Toc470524410)

[3.4 Разработка пользовательской документации 18](#_Toc470524411)

[3.5 Разработка программного кода 18](#_Toc470524412)

[3.6 Итоги 19](#_Toc470524413)

[4 Тестирование программного обеспечения 20](#_Toc470524414)

[4.1 Функциональное тестирование 20](#_Toc470524415)

[Заключение 21](#_Toc470524416)

# **Введение**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В век мобильных устройств человечество стремится, чтобы вся информация была под рукой и в удобном формате. Иметь возможность узнать последние новости, посмотреть местоположение чего-либо, иметь возможность организовать свой день в соответствий с расписанием. Все это можно быстро делать имея под рукой такое устройство как смартфон и соответствующее приложение. Но что делать студенту, если такого приложения нет под рукой? Если каждый раз приходится загружать страницу с расписанием, чтобы узнать где пара. Если каждый раз нужно узнавать у других где будет тот или иной преподаватель. Если каждый раз необходимо выполнять массу операций, чтобы найти нужную информацию по поводу расположения корпусов, посещения пар, изменения в расписании и т.д. На все эти вопросы приходит очевидный ответ – необходимо создать удобное приложение для мобильных устройств, которое выполняло бы все эти действия.

Данное приложение получило название «Мобильный ВятГУ». Оно направлено на упрощение доступа студентов к различным часто используемым вещам, таким как: расписание как студентов так и преподавателей, местоположение корпусов, справочная информация

# **Постановка расширенного технического задания**

В данной главе описываются требования, которые предъявляются к конечному продукту и на основе заданных требований составляется расширенное техническое задание.

## **Анализ предметной области**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

## **Анализ аналогов**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В данном разделе производится анализ аналогичных продуктов с целью выявления их недостатков и преимуществ.

**Мобильное приложение для просмотра расписания студентов ВятГУ**

Основная цель данного приложения – удобное получение информации студентам с помощью мобильных устройств.

Основные возможности:

* Демонстрация текущей и следующей пары для быстрого получения информации.
* Вывод расписания для текущего и завтрашнего дня.
* Представление информации о звонках — начало и конец каждой пары.
* Таймер отсчета времени до конца текущей пары.
* Публикация адреса для каждого корпуса.

Основной интерфейс:

* Блок «Общая информация», состоящий из 4 ключевых элементов: текущая, следующая, сегодняшние и завтрашние пары, адреса корпусов.
* Блок «Расписание» - вывод текущей недели, демонстрация занятий каждого дня недели, расписание звонков.
* Блок «Мероприятия» - вывод информации из раздела официального веб-сайта с расписанием грядущих мероприятий.

Недостатки:

* Нет расписаний преподавателей
* Долгая работа сервиса, так как он обращается к серверу университета для получения расписания

Пример графического интерфейса приложения представлен на рисунке 1



Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

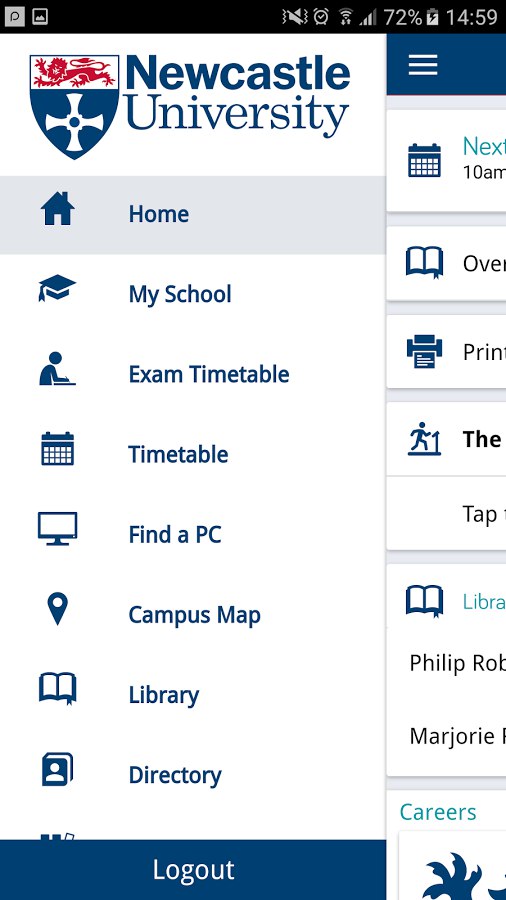
Рисунок 1 – графический интерфейс мобильного приложения ВятГУ

**Newcastle University**

Данное приложение разработано для одноименного университета. Оно нацелено в основном на студентов и имеет следующие возможности:

* Доступ к библиотеке университета
* Удобный вывод расписания учебных пар
* Вывод расписания экзаменационной сессии
* Отображения количества свободных компьютеров
* Вывод новостей университета
* Наличие навигации по корпусам
* Встроенная необходимая справочная информация для студентов

Пример графического интерфейса представлен на рисунке 2



Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

Рисунок 2 – Графический интерфейс приложения Newcastle University

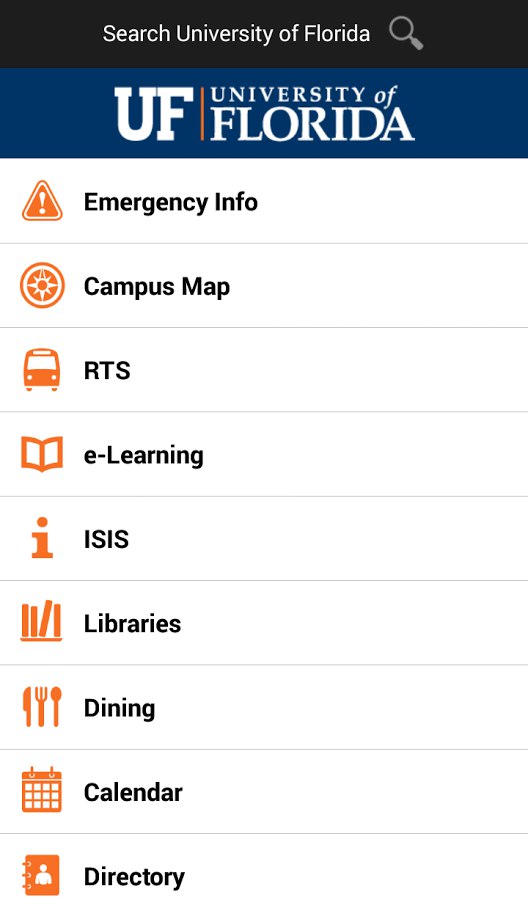
**University of Florida**

Официальное мобильное приложение разработанное для студентов, преподавателей и другого персонала университета. Данное приложение предоставляет навигацию по учебным корпусам, доступ к свежим новостям и событиям университета, доступ к различным ресурсам университета

Основные возможности:

* Лента новостей, которая рассказывает о последних событиях университета
* Календарь последних университетских событий
* Карта кампуса для удобного поиска учебных аудиторий
* Доступ к библиотечным ресурсам
* Расписание учебных пар студентов

Пример графического интерфейса представлен на рисунке 3



Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

Рисунок 3 – Графический интерфейс приложения University of Florida

Сравнительная таблица аналогов и их критериев представлена в таблице 1

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Мобильное приложение для расписания ВятГУ | Newcastle University | University of Florida |
| Лента новостей | - | + | + |
| Расписание для студентов | + | + | + |
| Расписание для преподавателей | - | - | + |
| Справочная информация | + | + | + |
| Библиотечные ресурсы | - | + | + |
| Календарь событий | + | + | + |

Как видно из представленной выше таблице, приложение для университета ВятГУ как минимум должен обладать возможностями приложений для других университетов, а именно:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

* Лента новостей о происходящих событиях в ВятГУ
* Расписание как для студентов, так и для преподавателей
* Доступ к библиотечным ресурсам
* Наличие справочное информации

На основании исследования аналогов можно выдвинуть требования, которым должно удовлетворять приложение

## Требования к приложению

В данном разделе, на основе анализа аналогов выдвигаются требования, которым должно удовлетворять приложение

**Функциональные характеристики**

Основные функции приложения:

* Вход в приложение по логину и паролю
* Отображение расписания на смартфоне в удобном виде. Уведомления о следующей паре
* Раздел новостей университета
* Возможность преподавателям вносить поправки в расписания
* Модуль объявлений, который необходим для своевременного оповещения студентов.
* Предоставление справочной информации. В их числе – адреса корпусов, кафедр, деканатов

**Требования к надежности**

Надежное функционирование приложения должно быть обеспечено выполнением конечным пользователем ряда требований по эксплуатации:

* Организация стабильной работы вычислительного устройства, в данном случае – смартфона
* Организация доступа в сеть для приложения, для имения наиболее актуальных данных внутри приложения

Со стороны разработчика приложение гарантирует корректную обработку всех входных данных, и если они не соответствуют желаемым, то оповещение об этом пользователя

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

**Условия эксплуатации**

В состав технических средств должен входить смартфон, включающий в себя:

* 2-х ядерный процессор с частотой 1.2 ГГц или больше
* 1 Гб оперативной памяти
* 20 мб свободного места в памяти смартфона

**Требования к информационной и программной совместимости**

Изменение формата вывода расписания на сайте может привести к некорректному разбору расписания сервером. Будет необходимо изменение парсера

**Требования к информационным структурам и методам решения**

Программное обеспечение представляет из себя самостоятельное исполняемое приложение для выполнения в операционной системе Android

**Требования к исходным кодам и языкам программирования**

Программное обеспечение должно быть разработано на языке Java 1.8 (клиентская и серверная часть), в среде программирования Android Studio. Исходный код оформляется с необходимыми комментариями.

**Требования к программным средствам, используемым программой**

Системные программные средства, используемые программой, должны быть представлены операционной системой Android версии 4.0 или выше.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

## Требования к программной документации

Разрабатываемое приложение будет поставляться со следующей программной документацией:

* Руководство пользователя, которое включает в себя описание всех возможностей приложения и исчерпывающую информацию по назначению каждого элемента.
* Исходный код с документацией.

## Технико-экономические показатели

Приложение будет распространятся по свободной модели, поэтому технико-экономические показатели рассчитывать не требуется

## Стадии и этапы разработки

Стадии этапов разработки представлены в таблице 2

Таблица 2

|  |  |
| --- | --- |
| **Месяц** | **Этап разработки** |
| Октябрь | Проектирование |
| Ноябрь | Реализация |
| Декабрь | Тестирование |

## Порядок контроля и приемки

Тестирование проводится путем:

* Ручного тестирования
* Автоматического тестирования

## Итоги

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В результате исследования аналогичных приложений было разработано расширенное техническое задание, которое в подробностях описывает требования, которые предъявляются к конечному продукту. Были определены технические требования к приложению, требования к исходным кодам, требования к документации, технико-экономические показатели, а так же стадии и этапы разработки.

# Проектирование структуры сервера

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В данной главе осуществляется этап проектирования конечного продукта. Основная цель данной главы - создать формальные модели и на их основе определить спецификации разрабатываемого программного обеспечения.

## Словарь терминов

Словарь терминов представляет собой краткое описание основных понятий, используемых при составлении спецификации. Он предназначен для повышения степени понимания предметной области, позволяет исключить разногласия при определении моделей между заказчиками и разработчиками.

Основные термины представлена в таблицах 3-4

Таблица 3

|  |  |
| --- | --- |
| Термин | API |
| Категория | программирование |
| Описание | Application Programming Interface - посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать |

Таблица 4

|  |  |
| --- | --- |
| Термин | JSON |
| Категория | Интернет-программирование |
| Описание | Java Script Object Notation - Тестовый формат обмена данными, основанный на JavaScript |

## Разработка базы данных

Изм.

Лист

№ докум.

Подпись

Дата

Лист

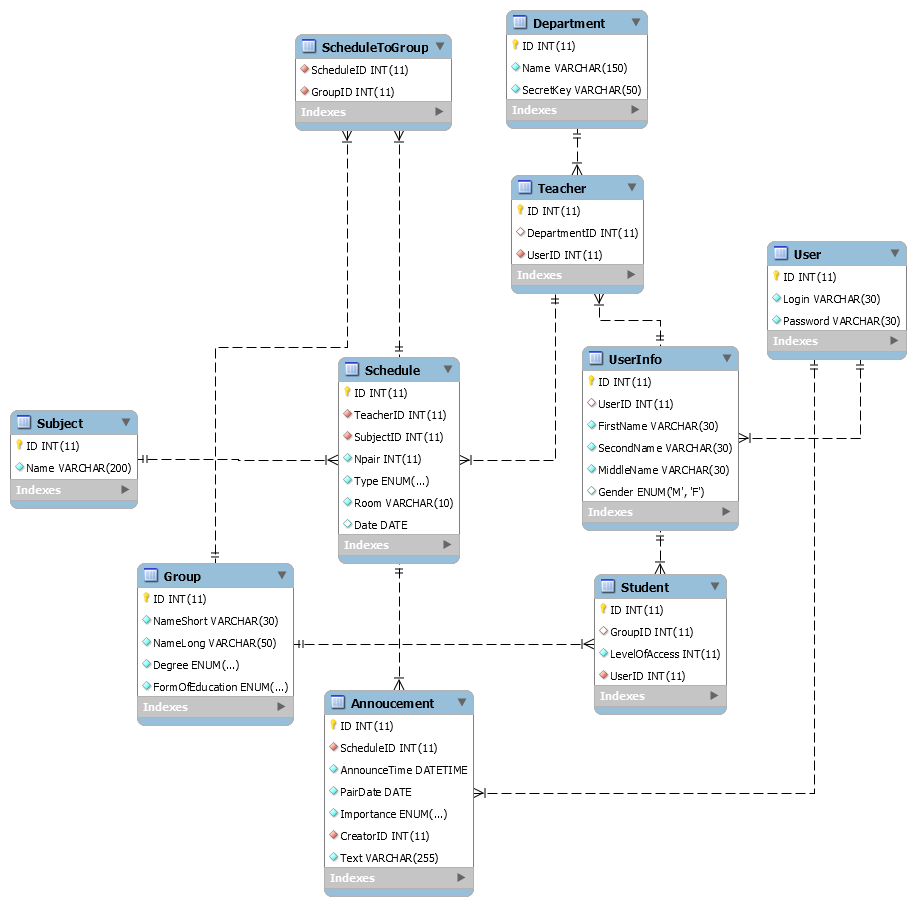
ТПЖА 09.03.01.024 ПЗ

Основная цель базы данных – хранить ученые записи студентов и преподавателей, иметь актуальную версию расписания. В ходе определения данных, которые должна хранить база данных, были разработаны таблицы. Таблица таблиц в базе данных представлена в таблице 5.

Таблица 5

|  |  |
| --- | --- |
| Название таблицы | Назначение |
| User | Хранит логин и пароль пользователя |
| UserInfo | Хранит информацию о ФИО и поле пользователя |
| Student | Хранит информацию о студенте: его пользовательский индентификатор, группу. |
| Group | Хранит название группы, степень, форму образования |
| Teacher | Хранит информацию о идентификаторе кафедры, и идентификаторе пользовательской информации |
| Department | Хранит название кафедр и их секретные ключи |
| Schedule | Хранит номер учебной пары, идентфикаторы преподавателей, предметов, даты пар |
| ScheduleToGroup | Промежуточная таблица, которая связывает таблицы Schedule и Group |
| Subject | Хранит учебные предметы |
| Annotation | Хранит сообщения, прикрепленные к парам |

Диаграмма базы данных представлена на рисунке 4



Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

Рисунок 4 – Диаграмма базы данных

## Диаграммы потоков данных

Процесс взаимодействия пользователя и функций сервера представлена в виде диаграммы потоков данных в приложении Б.

## Диаграмма модульной структуры

Диаграмма модульной структуры представлена в приложении В

## Итоги

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В результате процесса проектирования были разработаны и разобраны основные элементы из которых будет состоять приложение:

* База данных
* Серверная часть
* Клиентская часть

В ходе проектирования было выяснено, что необходима реляционная база данных, которая будет хранить списки студентов, групп, преподавателей, учебных пар и тд.

В ходе проектирования серверной части было разработано 24 метода, которые будет реализовывать сервер. Методы были разделены на 7 групп:

1. Методы пользователя
2. Методы студентов
3. Методы групп
4. Методы преподавателей
5. Методы кафедры
6. Методы расписания
7. Методы оповещений

Каждая из групп методов реализует методы, относящиеся к одной из таблиц в базе данных.

# Реализация программного обеспечения

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В данной главе описывается процесс реализации программного продукта и обосновываются выборы различных решений в ходе разработки.

## Выбор необходимых библиотек

Для разработки сервера было выбрано несколько библиотек, которые упрощают разработку:

* Jetty. Данная библиотека необходима для реализации Http сервера, она позволяет не задумываться о многопоточности, синхронизации потоков и тд. Основные достоинства:
  + Реализация многопоточного приложения используя пул свободных потоков
  + Распознавание аргументов переданных в запросе
  + Обработка заголовков запроса
  + Поддержка POST, GET и других видов запросов
* SpringJDBC. Данная библиотека реализует взаимодействие с базой данных, позволяя программисту не задумываться о тонкостях. Основные достоинства:
  + Простой и удобные способ реализации запросов к базе данных
  + Защита от SQL инъекций
* Gson. Данная библиотека позволяет «конвертировать» объекты в JSON файлы.

## Разработка лексического анализатора для расписания

Для поддержки актуальной версии расписания, учебных группы и преподавателей был разработан лексический анализатор, который распознает расписание и создает соответствующие записи в базе данных. В качестве языка программирования для написания лексического анализатора был выбран Python, так как он позволяет в очень короткие сроки написать программное обеспечение, а наличие библиотек позволяет упростить процесс загрузки и распознавания html страниц.

## Разработка сервера

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

Сервер для приложения был разработан на языке программирования Java.

Сервер реализует API (Application Programming interface) – это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными. Это интерфейс, который позволяет получать информацию из базы данных с помощью http-запросов к специальному серверу. Не требуется знание в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

В качестве ответа сервер возвращает документ в формате JSON, который является широко распространенным форматом передачи данных за счет простого описания данных, который позволяет создавать документы, которые не содержат множество служебной информации.

## Разработка пользовательской документации

Данная документация включает в себя описание всех возможностей сервера. В ней описаны:

* Основные структуры данных
* Ошибки, которые могут возникнуть в процессе выполнения запросов к методам
* Описание методов с их входными параметрами и примерами

Пользовательская документация представлена в приложении В.

## Разработка программного кода

Программный код продукта представлен в репозитории gitHub, который расположен по адресу <https://github.com/Nellrun/JavaServer>

## Итоги

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

На этапе реализации был разработан сервер приложения на языке программирования Java, который реализует интерфейс API. Взаимодействие с сервером осуществляется при помощи Get запросов по Http протоколу. В качестве ответа сервер возвращает документ в формате JSON.

Для поддержания актуальности информации в базе данных был разработан лексический анализатор на языке программирования Python. Данный лексический анализатор распознает расписание преподавателей и обновляет списки пар, учебных групп и преподавателей в базе данных.

# Тестирование программного обеспечения

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

В данной главе доказывается работоспособность разработанного приложения при помощи проведения различных тестов:

* Функциональных
* Тестированием белого ящика

## Функциональное тестирование

Данный этап тестирования предполагает тестирование «черного ящика». Это тестирование подразделяется на 2 этапа:

* Выделение классов эквивалентности
* Составление тестов и проведение тестирования

На данном этапе проходить тестирование будут не все методы сервера, а только самые основные:

* Student.get
* Student.search
* Group.changeAccess

# Заключение

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

Библиографический список

Изм.

Лист

№ докум.

Подпись

Дата

Лист

ТПЖА 09.03.01.024 ПЗ

1. Ростовцев В.С. Оформление курсовых и дипломных проектов для студентов специальности 230101 [Текст] / В.С. Ростовцев, С.Д. Блинова. – Киров: Изд-во ВятГТУ, 2006. – 39 с.
2. Браудэ Э. Технология разработки программного обеспечения . – СПБ.: Питер, 2004.- 655 с.: ил.
3. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОСМОТРА РАСПИСАНИЯ СТУДЕНТОВ // Молодежный научный форум: Технические и математические науки: электр. сб. ст. по материалам XXXI студ. междунар. заочной науч.-практ. конф. — М.: «МЦНО». — 2016 —№ 2(31) / [Электронный ресурс] — Режим доступа. — URL: http://nauchforum.ru/archive/MNF\_tech/2(31).pdf

Приложение А





Приложение Б



