

КУРСОВЫЕ

№ п/п	Наименование дисциплины (модуля)	Тема курсовой работы (проекта)
1	Информатика	Прикладные основы АВА
2	Комп. граф.	Разработка компьютерной игры "Подсудимые"
3	Теория автоматов	Синтез микропрограммного управляющего автомата
4	РПС	Задача на применение "Модельный Аппарат"
5	Разработка модулей СПД	Разработка программы поискового выбора методов решения задач

31

РАБОТЫ (ПРОЕКТЫ)

Чусовая Д.С.
(Фамилия, И.О. студента)

Семестр	Оценка	Дата сдачи	Подпись преподавателя	Фамилия преподавателя
1	отлично	05.06.15	<i>[Подпись]</i>	Ширяев
3	отлично	17.12.15	<i>[Подпись]</i>	Киселева
4	хорошо	20.06.16	<i>[Подпись]</i>	Мельцов
5	хорошо	20.09.17	<i>[Подпись]</i>	Чистяков
6	хорошо	14.06.17	<i>[Подпись]</i>	Ширяев

Декан

32

(подпись)

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)
«__»_____ 2017г.

РАЗРАБОТКА 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 2017

Содержание

Введение	4
1 Постановка расширенного технического задания	5
1.1 Анализ предметной области.....	5
1.2 Анализ аналогов	6
1.3 Требования к приложению.....	10
1.4 Требования к программной документации.....	12
1.5 Техничко-экономические показатели	12
1.6 Стадии и этапы разработки	13
1.7 Порядок контроля и приемки.....	13
1.8 Выводы.....	13
2 Проектирование структуры сервера.....	14
2.1 Словарь терминов	14
2.2 Разработка структур данных	14
2.3 Разработка базы данных	17
2.4 Диаграммы потоков данных.....	19
2.5 Диаграмма модульной структуры	19
2.6 Выводы.....	19
3 Реализация серверной части.....	20
3.1 Выбор необходимых библиотек	20
3.2 Разработка лексического анализатора для расписания	20
3.3 Разработка сервера	23
3.4 Разработка пользовательской документации	24
3.5 Разработка диаграммы классов.....	24
3.6 Разработка программного кода.....	26
3.7 Выводы.....	26
Заключение.....	27

					ТПЖА 09.03.01.024 ПЗ						
					РАЗРАБОТКА ПРИЛОЖЕНИЯ «МОБИЛЬНЫЙ ВЯТГУ»			Лит.	Масса	Масштаб	
Изм.	Лист	№ докум.	Подпись	Дата							
Разраб.											
Провер.											
Т. Контр.											
								Лис	Листов		
Реценз.								ВятГУ			
Н. Контр.											
Утверд.											
										3	

Введение

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 соответствии с расписанием. Все это можно быстро делать имея под рукой такое устройство как смартфон и соответствующее приложение.

Основной проблемой является отсутствие такого приложения у ВятГУ. Студенты не могут быстро найти необходимую им информацию на сайте из-за ее разрозненности. Преподаватели не всегда имеют возможности вовремя оповестить студентов об изменении расписаний. Проверить новости университета не всегда возможно из-за большого размера страниц и долгого времени их загрузки. Именно эти проблемы должно решать приложение для мобильных устройств.

Таким образом, задача разработки приложения, ориентированного на упрощение рутинных задач студентов и преподавателей, является актуальной.

					ТПЖА 09.03.01.024 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

1 Постановка расширенного технического задания

В данной главе производится анализ предметной области, исследуются аналоги, описываются требования, которые предъявляются к конечному продукту. Результатом данной главы является постановка расширенного технического задания.

1.1 Анализ предметной области

Данный учебный проект нацелен на две основные группы: студенты и преподаватели. В рамках университета, каждый студент закрепляется за группой, а преподаватель – за кафедрой. Каждая группа или преподаватель имеет в расписаний свой список учебных пар. Каждую пару ведет только один преподаватель, но пару может посещать несколько учебных групп.

Основным способом получения информации об учебных парах как для студентов, так и для преподавателей является расписание. Бывают случаи, при которых преподавателю необходимо поставить дополнительную учебную пару или консультацию по учебной дисциплине. Для этого преподаватель должен своевременно оповестить старосту, либо студентов группы, но это не всегда возможно. Иногда, из-за какого-либо стечения обстоятельств необходимо заменить учебную пару, тогда преподаватель так же должен заранее предупредить студентов. Это одна из основных проблем, которые необходимо решить в данном проекте.

События могут происходить как внутри учебного расписания, так и в рамках самого университета. Данные события публикуются на официальном сайте университета. Для наиболее своевременного и удобного получения информации о предстоящих или уже случившихся событиях студент или преподаватель должен иметь приложение, которое смогло бы его оповещать при появлении новых новостей на официальном сайте.

Наиболее часто необходимой информацией является справочная информация: расположение учебных корпусов, деканатов, кафедр, библиотеки и т.д. Но все эти данные разрозненные на официальном сайте университета. Необходимо так же решить эту проблему во время разработки учебного проекта.

					ТПЖА 09.03.01.024 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Анализ аналогов

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.



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.

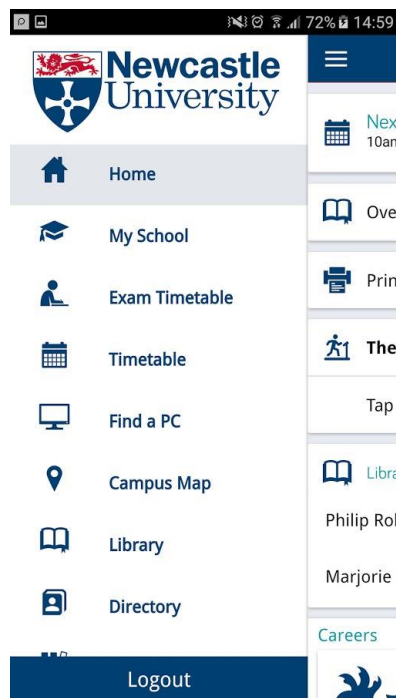


Рисунок 2 – Графический интерфейс приложения Newcastle University

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

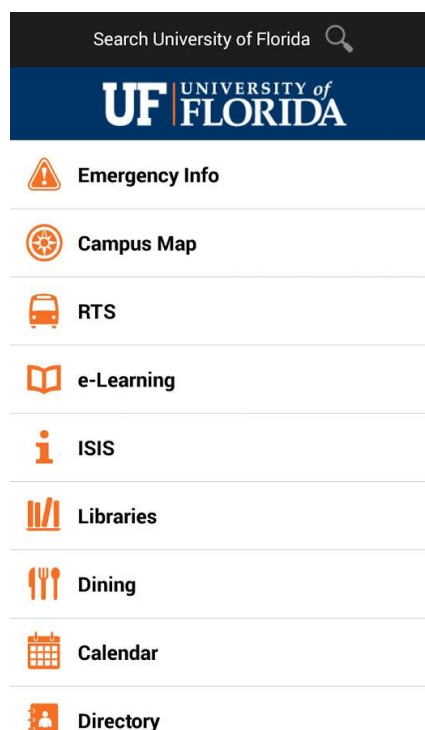


Рисунок 3 – Графический интерфейс приложения University of Florida

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	+	+	+

Представленная выше таблица наглядно демонстрирует функций, которые являются наиболее часто реализуемыми и полезными для приложений такого типа. Приложение для университета ВятГУ как минимум должен обладать возможностями приложений для других университетов, а именно:

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

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

1.3 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 имени 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.

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 лексического анализатора.

Требования к информационным структурам и методам решения

Программное обеспечение представляет из себя самостоятельное исполняемое приложение для выполнения под управлением операционной системы Android.

					ТПЖА 09.03.01.024 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

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.

1.4 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.

1.5 Техничко-экономические показатели

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

					ТПЖА 09.03.01.024 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

1.6 Стадии и этапы разработки

Стадии этапов разработки представлены в таблице 2.

Таблица 2

Месяц	Этап разработки
Октябрь	Проектирование
Ноябрь	Реализация
Декабрь	Тестирование

1.7 Порядок контроля и приемки

Тестирование проводится путем:

- Ручного тестирования.
- Автоматического тестирования.

1.8 Выводы

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, исходным кодам, документации, технико-экономическим показателям.

2 Проектирование структуры сервера

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

2.1 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 таблице 3

Таблица 3 – описание терминов

Термин	Категория	Описание
API	Программирование	Application Programming Interface - посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать
JSON	Интернет-программирование	Java Script Object Notation - Тестовый формат обмена данными, основанный на JavaScript

2.2 Разработка структур данных

Разработка структур данных позволяет выделить сущности, с которыми будет взаимодействовать пользователь при обращении к веб-серверу.

Структура объекта студента

Данная структура будет описывать пользователя-студента. Данная сущность может состоять в учебной группе и иметь определенные права доступа.

Структура:

- Фамилия, имя, отчество.
- Учебная группа.
- Права доступа. Определяют уровень возможностей, которые доступны студенту внутри учебной группы:
 - Не авторизован. Студент не является авторизованным в данной учебной группе.
 - Только чтение. Разрешено только чтение сообщений, адресованных данной учебной группе
 - Чтение и запись. Разрешено читать и писать сообщения.
 - Староста группы. Разрешено редактировать права доступа студентов, которые состоят в данной группе.

Возможности студента:

- Получать расписание.
- Читать и писать сообщения, прикрепленные к учебным парам
- Смена учебной группы

Структура объекта преподавателя

Данная структура будет описывать пользователя-преподавателя. Регистрация преподавателя в данном приложении будет проходить при помощи проверки секретного ключа, который закрепляется за кафедрой, внутри которой состоит преподаватель.

Основные возможности преподавателя:

- Прикрепление сообщений к учебным парам.
- Изменение прав доступа внутри учебной группы.
- Добавление и удаление учебных пар из расписания групп.
- Смена секретного ключа кафедры.

Структура преподавателя:

- ФИО.
- Идентификатор кафедры.

Структура объекта группы

Данная структура описывает учебную группу. В учебной группе могут состоять только студенты.

Структура:

- Имя группы
- Форма обучения.
 - Очная.
 - Заочная.
 - Дистанционная.

Структура объекта расписания

Данная структура описывает одну учебную пару расписания. У учебного занятия может быть только один преподаватель, но несколько групп.

Структура:

- Преподаватель.
- Название предмета.
- Номер учебной пары. Номер учебной пары рассчитывается по формуле: $(\text{номер недели} - 1) * 42 + (\text{день недели} - 1) * 7 + \text{номер пары}$.
- Тип пары.
 - Лекция.
 - Практика.
 - Лабораторная работа.
- Дата проведения.
- Список учебных групп.

Структура объекта сообщения

Данная структура описывает сообщения, которые прикрепляются к учебным парам. Сообщения могут писать только преподаватели и студенты с определенным уровнем доступа. Студенты могут удалять только свои сообщения, если они имеют уровень доступа ниже старосты. Удалять чужие сообщения могут только преподаватели и студенты с уровнем старосты.

Структура:

- Учебная пара, к которой прикрепляется сообщение.
- Указатель на пользователя, который создал сообщение.
- Текст сообщения.
- Дата проведения пары.
- Дата написания сообщения.

2.3 Разработка базы данных

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 следующие таблицы:

- User. Хранит логин и пароль пользователя.
- UserInfo. Хранит информацию о ФИО и поле пользователя.
- Student. Хранит информацию о студенте: его пользовательский идентификатор, 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, идентификаторы преподавателей, предметов, даты пар.
- ScheduleToGroup. Промежуточная таблица, которая связывает таблицы Schedule и Group.
- Subject. Хранит учебные предметы.
- Announcement. Хранит сообщения, прикрепленные к учебным парам.

Диаграмма сущность-связь представлена на рисунке 4.

По причине того, что множество объектов различных таблиц связаны между собой, например, таблица, которая хранит расписание, связана связью один ко многим с промежуточной таблицей, которая связывает группы, то наиболее логичным вариантом является использование реляционной модели базы данных.

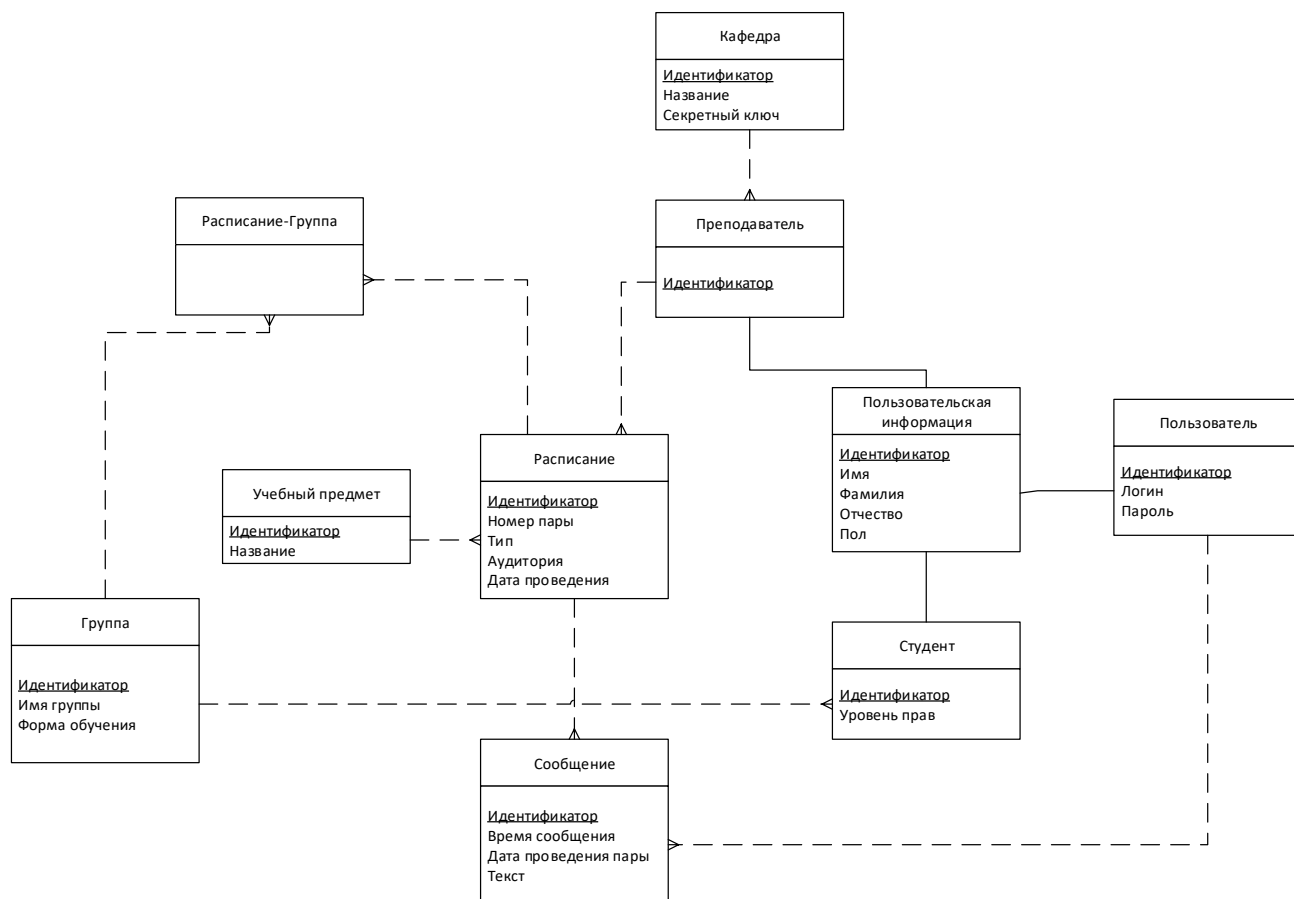


Рисунок 4 – Диаграмма сущность-связь

2.4 Диаграммы потоков данных

Диаграммы потоков данных описывают взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе. Процесс взаимодействия пользователя и сервера представлена в виде диаграммы потоков данных в приложении Б.

2.5 Диаграмма модульной структуры

Диаграмма модульной структуры представлена в приложении В.

2.6 Выводы

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

- База данных.
- Серверная часть.
- Клиентская часть.

Были определены основные структуры данных, которые будут храниться на сервере.

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

3 Реализация серверной части

В данной главе описывается процесс реализации серверной части приложения и лексического анализатора. Так же обосновываются выборы различных решений в ходе разработки.

3.1 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 удобный способ реализации запросов к базе данных
 - Защита от SQL инъекций
- Gson. Данная библиотека позволяет «конвертировать» объекты в JSON файлы.

3.2 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.

Выбор расписания для лексического анализа

Внутри ВятГУ существует несколько видов расписания: расписание преподавателей и расписание учебных групп. Каждое из этих расписаний имеет свой формат. Выбор расписания, которое необходимо проанализировать является ключевым моментом для разработки лексического анализатора. Вариант расписания преподавателя представлен на рисунке 5, расписания учебных групп – на рисунке 6.

День	Интервал	Архангельский В.В.	Вожегов Д.В.
Понедельник (1 неделя)	1 пара	2-308 Исследование операций Проведение лабораторных занятий ИВТ6-3301-01-00 ИВТ6-3301-01-00, 01 подгруппа 2-308 Исследование операций Проведение лабораторных занятий ИВТ6-3301-01-00 ИВТ6-3301-01-00, 02 подгруппа	1-116 Компьютерная графика Проведение лабораторных занятий ИВТ6-2301-01-00 ИВТ6-2301-01-00, 01 подгруппа
	2 пара	2-308 Исследование операций Проведение лабораторных занятий ИВТ6-3301-01-00 ИВТ6-3301-01-00, 01 подгруппа 2-308 Исследование операций Проведение лабораторных занятий ИВТ6-3301-01-00 ИВТ6-3301-01-00, 02 подгруппа	1-116 Компьютерная графика Проведение лабораторных занятий ИВТ6-2301-01-00 ИВТ6-2301-01-00, 01 подгруппа
	3 пара	2-300 Исследование операций Чтение лекций ИВТ6-3301-01-00 ИВТ6-3301-01-00, 01 подгруппа ИВТ6-3302-02-00, 02 подгруппа ИВТ6-3302-02-00, 01 подгруппа	1-313 Компьютерная графика Чтение лекций ИВТ6-2301-01-00 ИВТ6-2301-01-00, 01 подгруппа ИВТ6-2301-01-00, 02 подгруппа ИВТ6-2302-01-00 ИВТ6-2302-01-00, 01 подгруппа ИВТ6-2302-01-00, 02 подгруппа ИВТ6-2303-01-00
	4 пара		
	5 пара		

Рисунок 5 – Пример расписания преподавателей

Пятница (1 нед)	1 пара	
	2 пара	
	3 пара	
	4 пара	Метрология, стандартизация и сертификация Чтение лекций Скворцов А.А. 1-313
	5 пара	ИВТ6-3302-02-00, 01 подгруппа Лб. Разработка модулей системного программного обеспечения Шихов М.М. 1-114 ИВТ6-3302-02-00, 02 подгруппа Лб. Разработка модулей системного программного обеспечения Караваева О.В. 2-308
	6 пара	ИВТ6-3302-02-00, 01 подгруппа Лб. Разработка модулей системного программного обеспечения Шихов М.М. 1-114 ИВТ6-3302-02-00, 02 подгруппа Лб. Разработка модулей системного программного обеспечения Караваева О.В. 2-308
	7 пара	

Рисунок 6 – Пример расписания учебной группы

В результате анализа было выявлено, что лексический разбор расписания преподавателей более эффективный чем разбор расписания учебных групп по нескольким причинам:

- Простой формат. Расписание преподавателей имеет единый формат, а расписание учебных групп различный из-за деления групп на подгруппы, в результате чего одна группа может быть в одно время на различных предметах. Данного недостатка лишено расписание преподавателей по вполне очевидным причинам.

- Меньшее количество обращений к веб ресурсам. Расписание групп разбито таким образом, что на одной странице находится только расписание одной группы, из этого следует, что количество страниц равно количеству учебных групп. Расписание преподавателей разбито на учебные кафедры.

Определение шаблона в расписании

После определения типа расписания, необходимо найти шаблон, которому соответствует выбранное ранее расписание. Структура расписания преподавателей выглядит в виде таблицы, где столбцы — это имена преподавателей, а строки – номера учебных пар. Каждая ячейка данной таблицы содержит: номер учебного кабинета, название дисциплины, список групп. Номер кабинета, название дисциплины и списки групп имеют строго заданную последовательность, которую можно формализовать. Формализованный вид представлен на рисунке 7

<Номер учебного корпуса>-<Номер кабинета> <Название дисциплины> (Чтение лекций Проведение практических занятий, семинаров Проведение лабораторных работ) <Наименование учебной группы>
--

Рисунок 7 – Формализованный вид ячейки таблицы расписания

Реализация лексического анализатора

В качестве языка для реализации лексического анализатора был выбран язык Python, так как он является очень удобным языком для разработки, имеет множество стандартных библиотек, в число которых входит библиотека для поддержки регулярных выражений.

В качестве основного инструмента для анализа строк были выбраны регулярные выражения, в виду того, что они позволяют по заданному шаблону, получить из строки данные.

Реализация функции анализа строки представлена на рисунке 8. Результат лексического анализа одной ячейки расписания представлен на рисунке 9.

```

import re

def parse(s):

    exp = "^(\d{1,2}-\d{3})\s(.+)\s(Чтение лекций|Проведение лабораторных
занятий|Проведение практических занятий, семинаров)"

    groupsExp = "([А-Я]+[а-я]-\d+-\d+-\d+)"

    return re.findall(exp, s) + [set(re.findall(groupsExp, s))]

```

Рисунок 8 – программная реализация лексического анализатора

```

[('2-308', 'Исследование операций', 'Проведение лабораторных занятий'),
{'ИВТб-3301-01-00'}]

```

Рисунок 9 – Результат работы лексического анализатора

3.3 Разработка сервера

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.

3.4 Разработка пользовательской документации

Важным этапом разработки программного продукта является составление программной документации. Программная документация должна содержать все необходимые сведения по эксплуатации программного обеспечения. 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

3.5 Разработка диаграммы классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Диаграмма классов представлена в приложении В. Краткое описание и назначение классов:

- AnnouncementGetPage. Данная страница отвечает за выдачу последних сообщений, прикрепленных к учебной паре.
- AnnouncementAddPage. Позволяет добавить авторизованному пользователю сообщение, прикрепив его к учебной паре.
- AnnouncementDelPage. Позволяет удалять сообщения пользователей.
- ChangePasswordPage. Позволяет сменить пароль.
- DepartmentChangeSecretKeyPage. Позволяет сменить пароль кафедры.
- ScheduleGetPage. Позволяет получить расписание учебной группы или преподавателя.
- ScheduleAddPage. Позволяет преподавателю добавить учебную пару в расписание.
- ScheduleAddGroupPage. Позволяет преподавателю добавить учебную группу в пару.
- ScheduleDelPage. Позволяет удалить преподавателю учебную пару.

- ScheduleDelGroupPage. Позволяет убрать у группы учебную пару, которая была ранее добавлена.
- GroupGetPage. Позволяет получить информацию о учебной группе по ее идентификатору.
- GroupSearchPage. Позволяет найти учебную группу по ее короткому названию.
- GroupGetMembersPage. Позволяет получить список студентов, состоящих в данной группе.
- LogoutPage. Позволяет выйти пользователю из системы.
- SignInPage. Позволяет войти пользователю в систему.
- SignUpPage. Позволяет зарегистрироваться пользователю.
- StudentChangeGroupPage. Позволяет сменить учебную группу студента.
- StudentGetPage. Позволяет получить информацию о студенте.
- StudentSearchPage. Позволяет найти студента по его ФИО.
- TeacherGetPage. Позволяет получить информацию по преподавателе.
- TeacherSearchPage. Позволяет найти преподавателя по его ФИО.
- TeacherGetSubjectsPage. Позволяет получить учебные предметы, которые ведет преподаватель.
- UserChangeNamePage. Позволяет сменить ФИО пользователя.

3.6 Разработка программного кода

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

3.7 Выводы

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.

В ходе разработки серверной части было разработано 24 метода, которые реализует 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.

Заключение

В результате разработки приложения для мобильных устройств под операционной системой Android было получено приложение, удовлетворяющее следующим требованиям:

- Быстрый и удобный доступ к расписанию
- Возможность оповещения об изменении в расписании
- Возможность добавления новых учебных пар
- Оповещение о новых новостях университета
- Удобный доступ к справочной информации

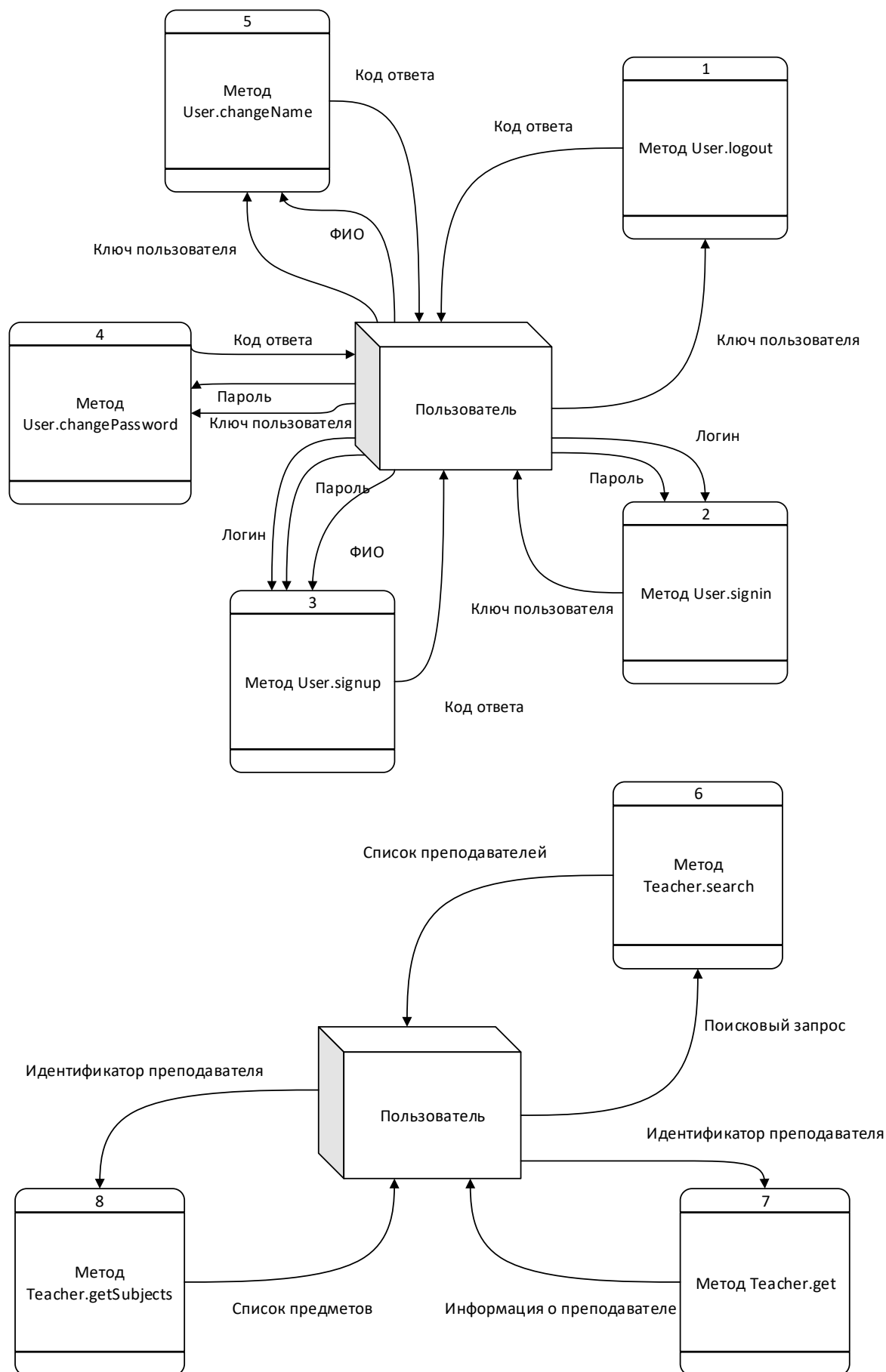
Данное приложение было разработано на языке программирования Java под Android 4.0 и выше. Приложение имеет серверную часть, которая разрабатывалась на языке программирования Java с использованием библиотек Jetty, Spring, Gson.

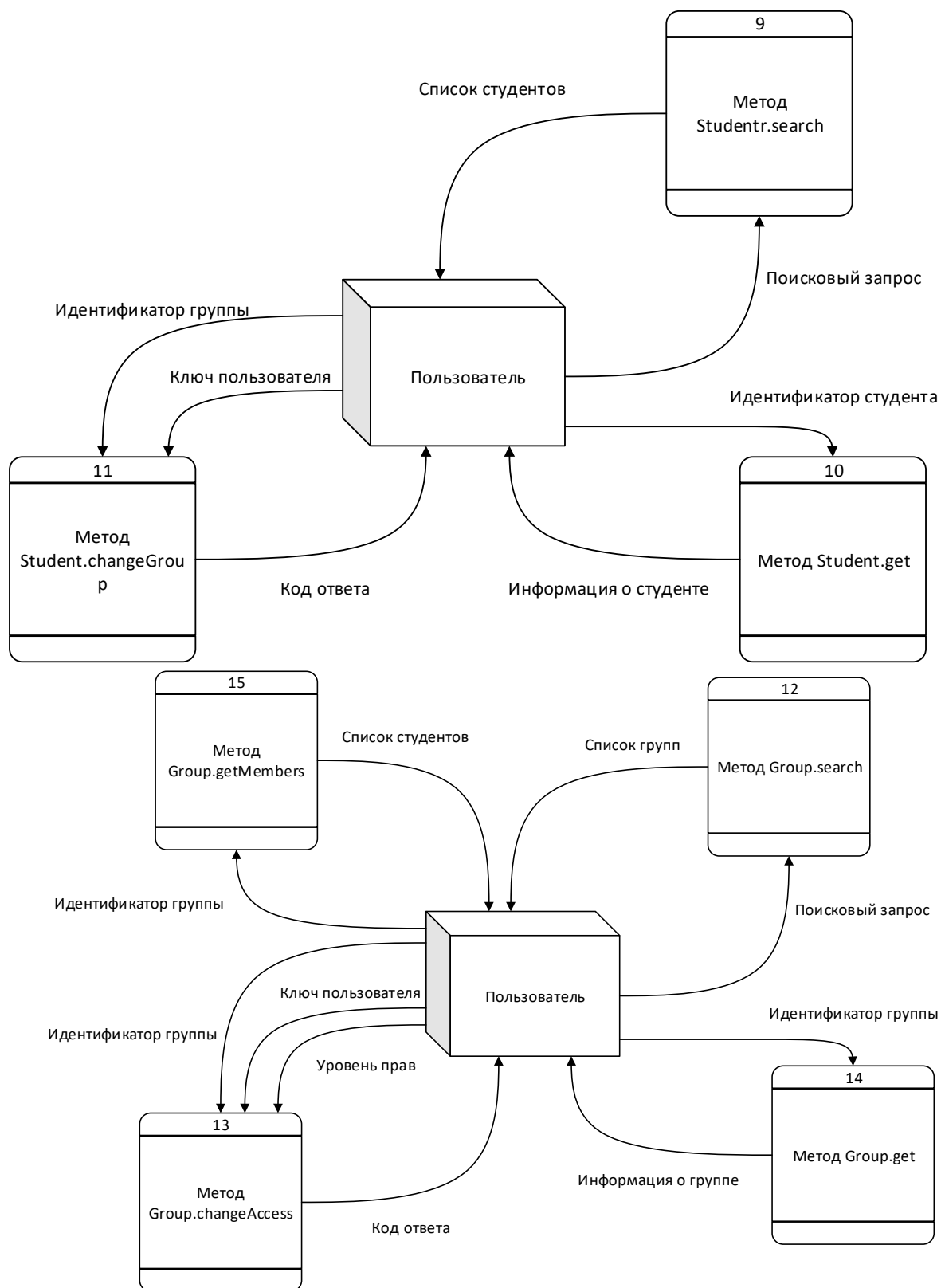
					ТПЖА 09.03.01.024 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

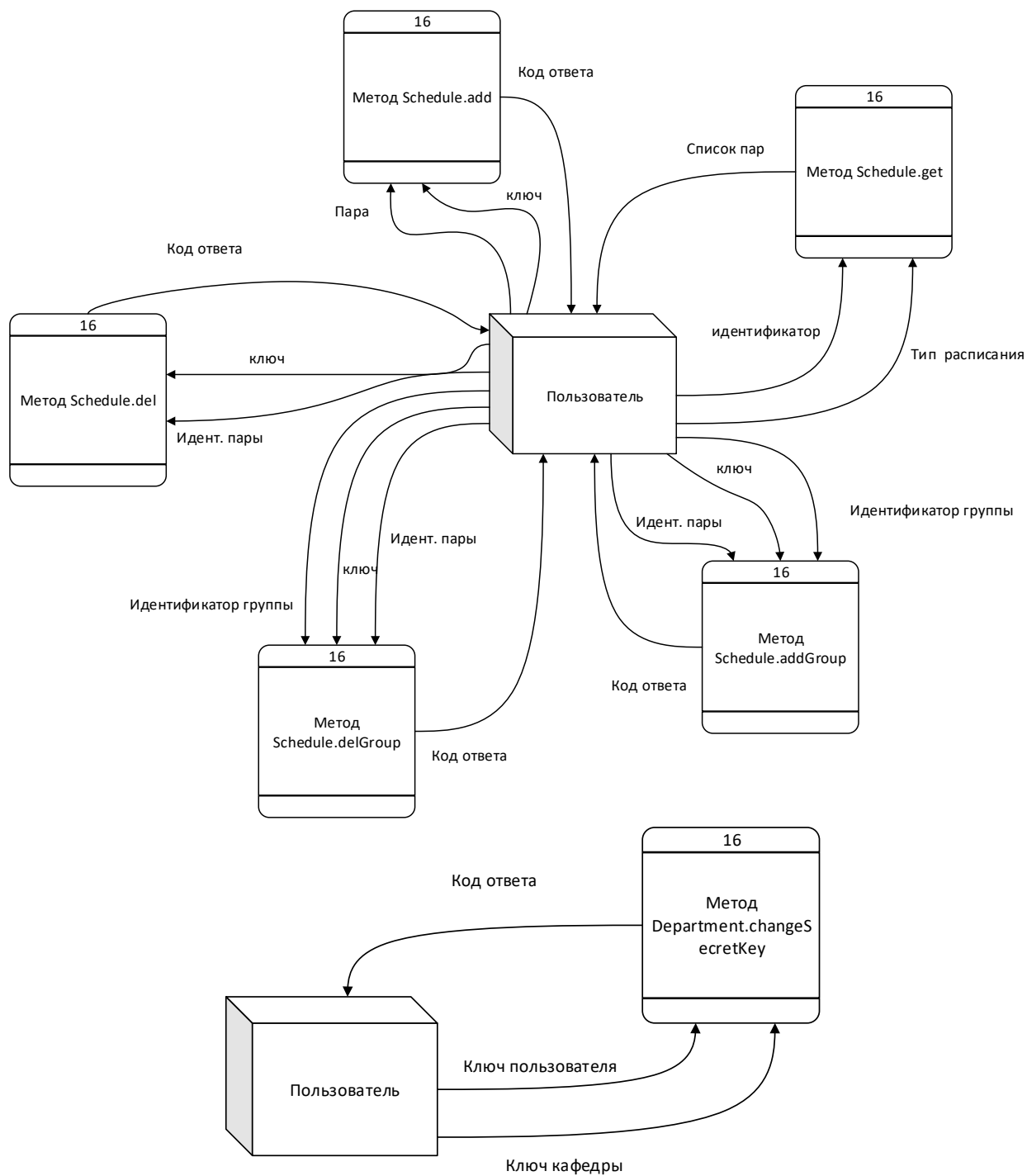
Библиографический список

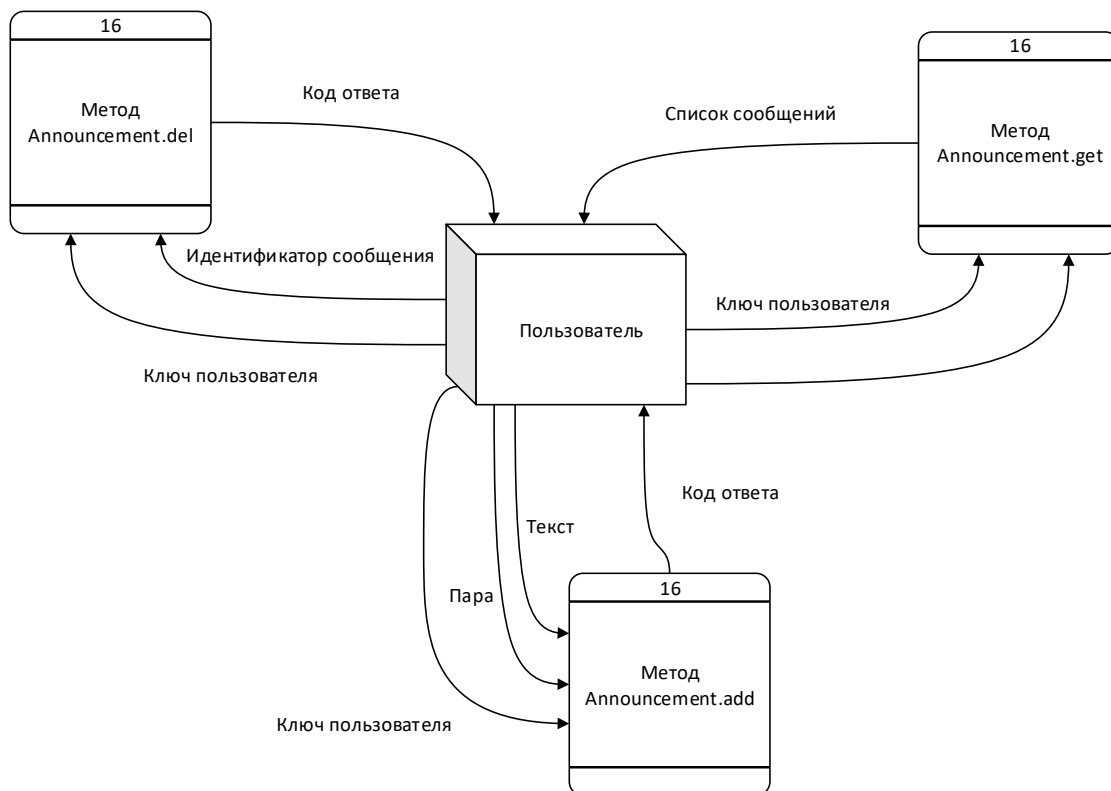
1. Ростовцев В.С. Оформление курсовых и дипломных проектов для студентов специальности 230101 [Текст] / В.С. Ростовцев, С.Д. Блинова. – Киров: Изд-во ВятГТУ, 2006. – 39 с.
2. Браудэ Э. Технология разработки программного обеспечения . – СПб.: Питер, 2004.- 655 с.: ил.
3. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОСМОТРА 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](http://nauchforum.ru/archive/MNF_tech/2(31).pdf)

Приложение А

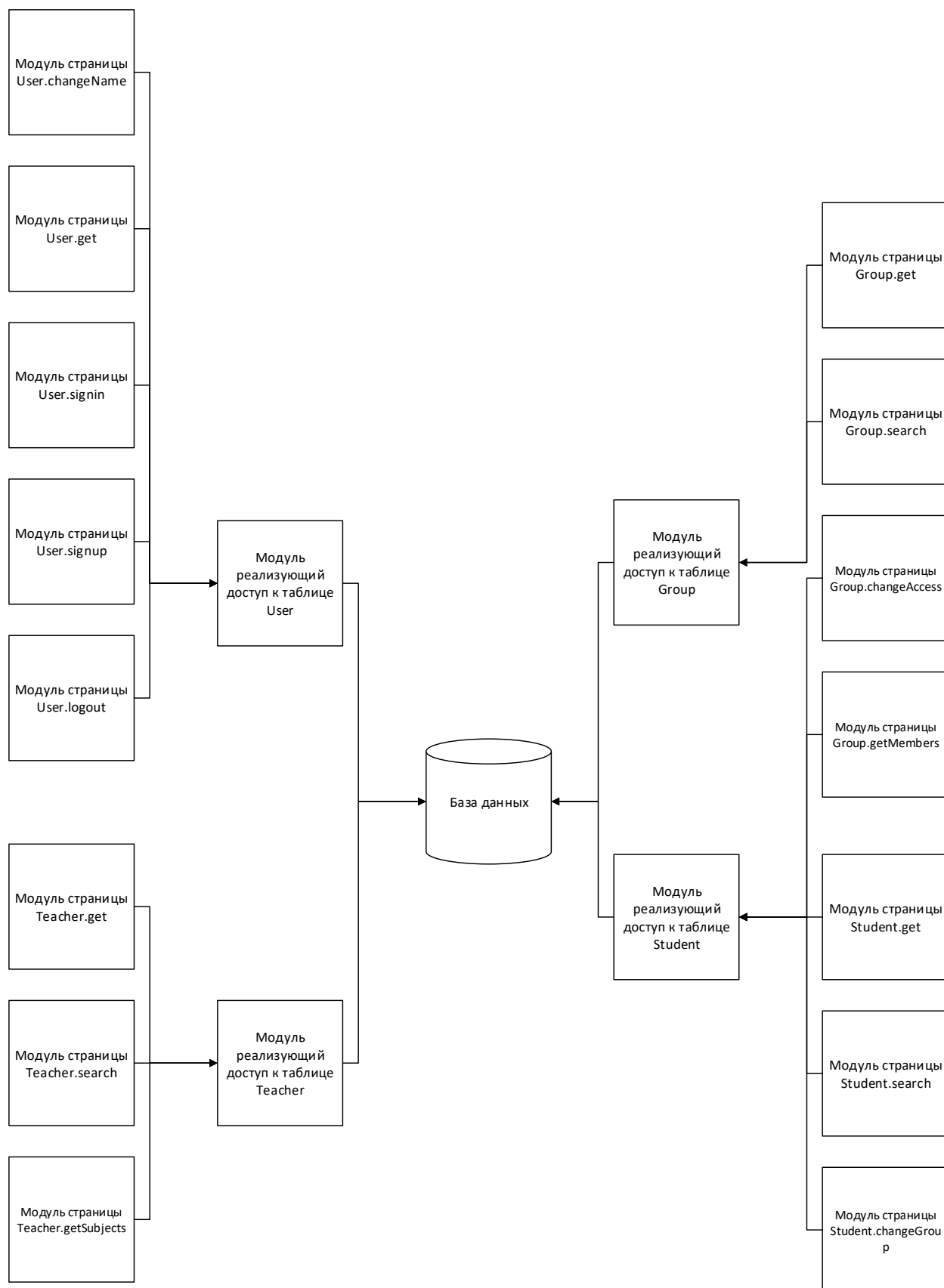








Приложение Б

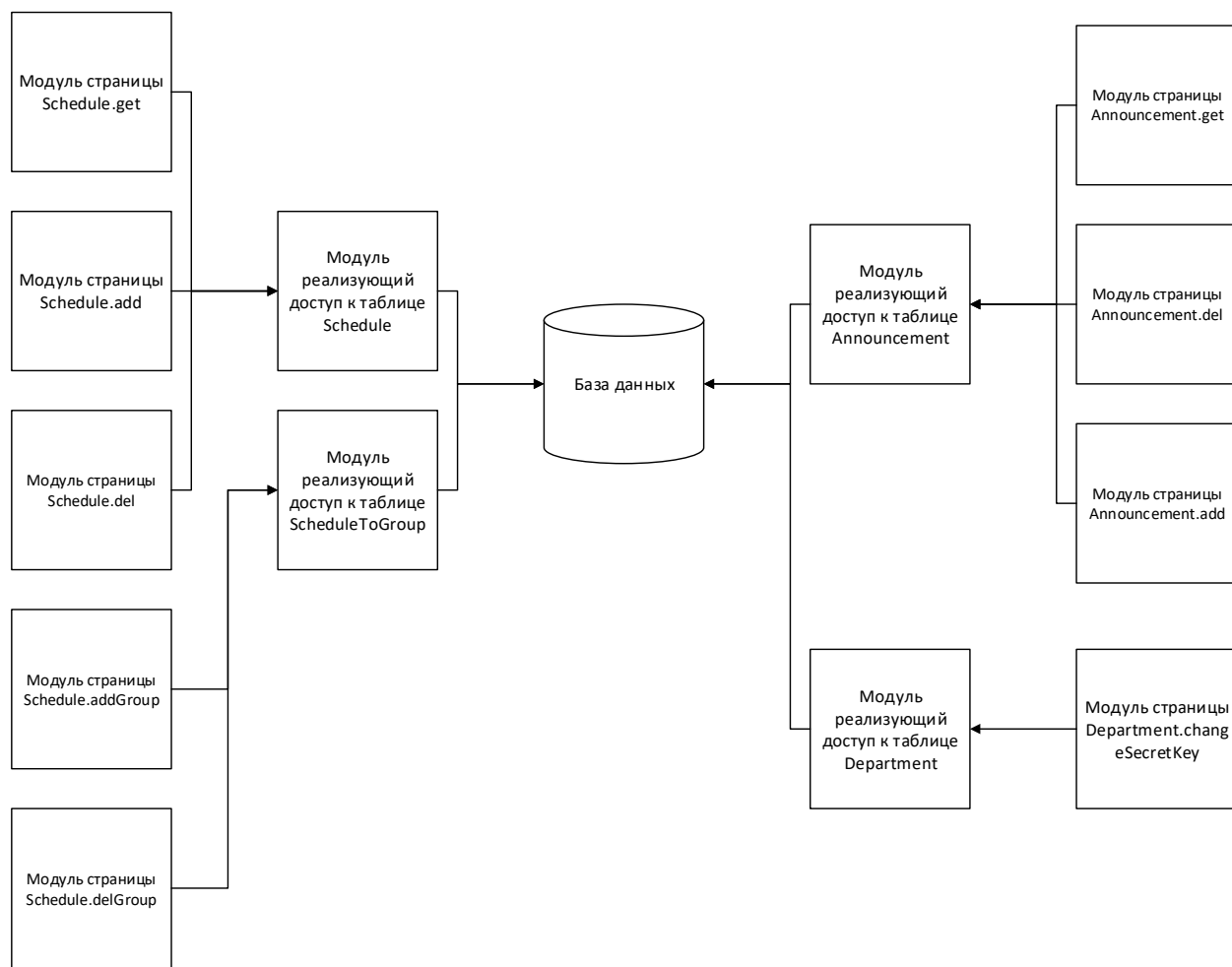


Изм.	Лист	№ докум.	Подпись	Дата

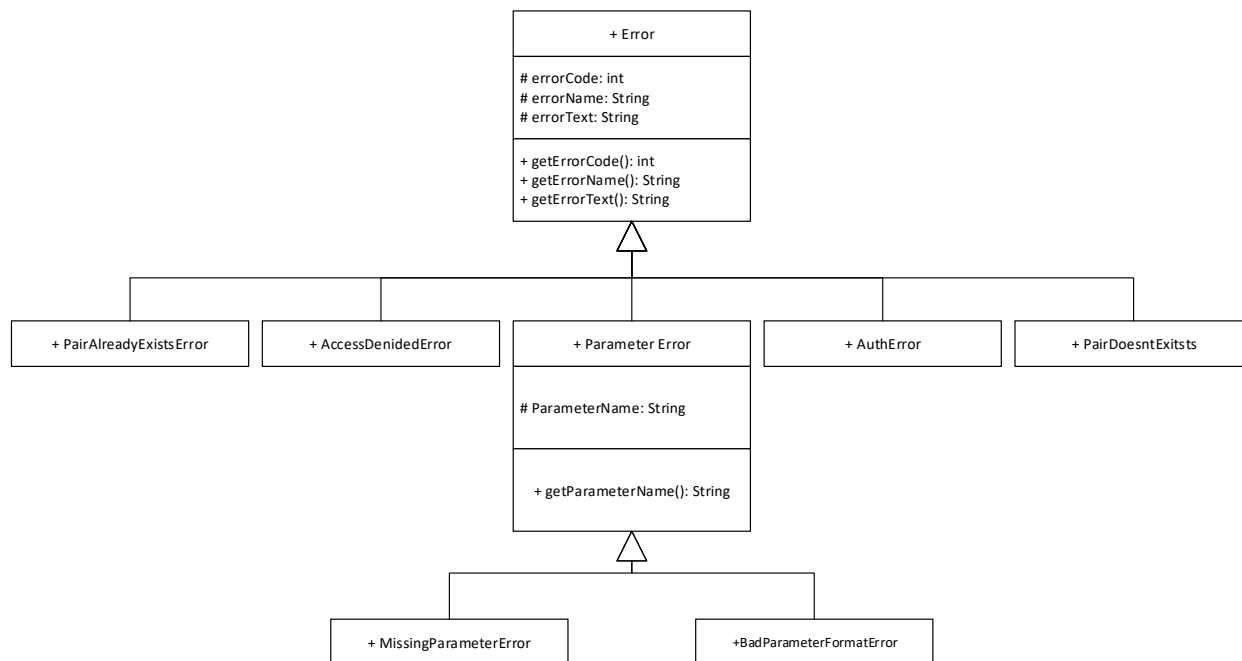
ТПЖА 09.03.01.024 ПЗ

Лист

33



Приложение В



Изм.	Лист	№ докум.	Подпись	Дата

ТПЖА 09.03.01.024 ПЗ

Лист

35



