

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
РУБЦОВСКИЙ ИНСТИТУТ (ФИЛИАЛ)

Кафедра математики и прикладной информатики

**Выпускная квалификационная работа
(дипломный проект)**

Тема: Разработка мобильного приложения для организации доступа к ресурсам информационно-образовательного портала Рубцовского Института (филиала) Алтайского Государственного Университета (на примере «РИ АлтГУ – Расписание» для платформы Android)

Выпускную квалификационную
работу (дипломный проект)
выполнил студент 5 курса,
группы 1205
Асадов А.А.

(подпись)

Научный руководитель:
старший преподаватель
Рязанова О.В.

(подпись)

Выпускная квалификационная работа
(дипломный проект) защищена

«__» _____ 2015 г.

Оценка _____

Председатель ГЭК

д.т.н., профессор

Пятковский О.И.

(подпись)

Допустить к защите

Зав. кафедрой

к.т.н., доцент

Жданова Е.А.

(подпись)

«__» _____ 2015 г.

Реферат

Отчет 92 страницы, 20 рисунков, 3 таблицы, 40 источников, 1 приложение.

СИСТЕМНЫЙ АНАЛИЗ, ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНЫЙ ПОРТАЛ, МОБИЛЬНЫЙ ИНТЕРНЕТ, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, РАСПИСАНИЕ.

Цель исследования – разработка мобильного приложения для организации доступа к ресурсам информационно-образовательного портала Рубцовского Института (филиала) Алтайского Государственного Университета (на примере «РИ АлтГУ – Расписание» для платформы Android), которое позволит обеспечить студентов, преподавателей, сотрудников института, а также слушателей курсов дополнительного профессионального образования постоянным актуальным расписанием, необходимым для непрерывного учебного процесса.

Объект исследования – информационно-образовательный портал Рубцовского Института (филиала) Алтайского Государственного Университета.

Предмет исследования – организация доступа к ресурсам информационно-образовательного портала.

Методы решения поставленных задач: системный анализ, моделирование предметной области, функционально-ориентированная методология.

Результаты проекта: проанализированы данные о предметной области, проведен системный анализ предметной области с использованием CASE-средства BPWin, на основе функциональной модели выявлены цели создания проекта, требования к нему, пользователи и разработано мобильное приложение.

Содержание

Введение.....	5
1 Аналитическая часть.....	8
1.1 Техничко-экономическая характеристика РИ (филиала) Алтайского Государственного Университета.....	8
1.2 Анализ функционирования объекта исследования.....	17
1.3 Определение цели и задач проектирования мобильного приложения	24
1.4 Обзор и анализ существующих разработок, выбор технологии проектирования.....	25
1.5 Выбор и обоснование проектных решений.....	30
2 Проектная часть.....	36
2.1 Разработка функционального обеспечения.....	36
2.2 Разработка информационного обеспечения.....	38
2.2.1 Используемые классификаторы и системы кодирования ...	38
2.2.2 Характеристика нормативно-справочной и входной оперативной информации.....	38
2.2.3 Характеристика результатной информации.....	39
2.2.4 Структура результатной информации.....	41
2.2.5 Информационная модель и ее описание.....	46
2.3 Разработка программного обеспечения.....	56
2.3.1 Описание серверных программных модулей.....	57
2.3.2 Описание клиентских программных модулей.....	58
2.3.3 Компоненты пользовательского интерфейса.....	60
2.4 Компьютерно-сетевое обеспечение.....	66

2.5	Обеспечение информационной безопасности	67
3	Оценка эффективности от внедрения мобильного приложения	72
3.1	Общие положения.....	72
3.2	Показатели эффективности.....	74
3.3	Расчет экономической эффективности.....	76
3.3.1	Расчет трудоемкости обработки информации.....	78
3.3.2	Расчет трудоемкости разработки программного обеспечения.....	79
3.3.3	Смета затрат на разработку программного обеспечения	83
	Заключение	87
	Список использованных источников	89
	Приложение А	93

Введение

Существенная доля всех пользователей сети Интернет для выхода во всемирную паутину используют мобильные устройства, такие как смартфоны, планшеты, коммуникаторы. Причем данный сегмент пользователей неуклонно растет. На мобильных устройствах установлена своя операционная система, и, можно сказать что по функционалу они не намного уступают персональным компьютерам. Самыми распространёнными операционными системами на мобильных устройствах являются iOS и Android.

Развитие данных областей меняет подход к получению необходимой информации. В последнее время, одним из самых распространенных подходов является получение информации при помощи мобильных приложений. Основным отличием является возможность его настройки под нужды пользователей.

По данным исследовательской компании Nielsen – пользователи все чаще пользуются сервисами при помощи мобильных приложений, а не на сайтах.

В настоящее время у многих учебных заведений существует мобильное приложение, которое позволяет обеспечить удобный доступ к необходимой информации.

На данный момент в Рубцовском Институте (филиале) Алтайского Государственного Университета функционирует информационно-образовательный портал, который обеспечивает доступ к информационным ресурсам в ключевых областях, таких как образование, наука, внеучебная жизнь, законодательные акты, библиотека и др.

Основной информацией, необходимой для студентов и сотрудников в учебном заведении, является расписание занятий, консультаций и предметных комиссий. Доступ к расписанию можно получить при помощи десктопной и мобильной версий сайта. Использование мобильного приложения для получения данной информации позволит обеспечить пользователей постоянным и актуальным расписанием, что сократит случаи, связанные с

отсутствием расписания и опозданиями.

Основными достоинствами мобильных приложений, по сравнению с мобильными сайтами, являются:

- интерактивные возможности;
- дружелюбный и понятный пользовательский интерфейс;
- работа без наличия интернета – в автономном режиме;
- сокращение мобильного трафика;
- автоматизация необходимых действий.

Отсутствие мобильного приложения для РИ (филиале) АлтГУ делает выбранную тему актуальной.

В связи с этим, объектом исследования данного проекта является информационно-образовательный портал Рубцовского Института (филиала) Алтайского Государственного Университета.

Предметом исследования является организация доступа к ресурсам информационно-образовательного портала.

Целью исследования является разработка мобильного приложения для организации доступа к ресурсам информационно-образовательного портала Рубцовского Института (филиала) Алтайского Государственного Университета (на примере «РИ АлтГУ – Расписание» для платформы Android), которое позволит обеспечить студентов, преподавателей, сотрудников института, а также слушателей курсов дополнительного профессионального образования постоянным актуальным расписанием, необходимым для непрерывного учебного процесса.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- дать характеристику исследуемой предметной области;
- провести анализ предметной области;
- построить функциональную модель существующей организации доступа к ресурсам информационно-образовательного портала;
- выявить недостатки существующей системы, на основании

выявленных недостатков предложить функциональную модель «Как должно быть»;

- разработать мобильное приложение;
- оценить эффективность от внедрения мобильного приложения.

Методы, используемые при написании проекта: системный анализ, моделирование предметной области.

1 Аналитическая часть

1.1 Техничко-экономическая характеристика РИ (филиала) Алтайского Государственного Университета

РИ (филиал) АлтГУ – обособленное структурное подразделение Алтайского Государственного университета, расположенное в городе Рубцовске Алтайского края и осуществляющее постоянно его функции в рамках выданной Институту лицензии на образовательную деятельность.

Предприятие является государственной, некоммерческой организацией, имеет право на ведение образовательной деятельности и на льготы, установленные законодательством РФ, в соответствии с лицензией, выданной Министерством образования РФ.

Институт реализует образовательные программы полного и неполного высшего профессионального образования.

Институт обладает правом реализации образовательных программ по хозрасчетным договорам в рамках установленного перечня специальностей.

Основными функциями Института является:

– реализация образовательных программ в сфере высшего профессионального образования по очной, заочной и очно-заочной (вечерней) формам обучения по направлениям подготовки (специальностям) Алтайского Государственного Университета в соответствии с лицензией, выданной Институту; организация учебного процесса в Институте по образовательным программам высшего профессионального образования регламентируется учебным планом по направлению подготовки специальности и расписанием учебных занятий для каждой формы обучения, которые разрабатываются Институтом самостоятельно и утверждаются на основе государственного образовательного стандарта высшего профессионального образования, примерных образовательных программ, учебных планов по направлению подготовки специальности и программам дисциплин, утверждаемых

федеральным органом управления образованием. При этом примерный учебный план и программы дисциплин имеют рекомендательный характер;

- реализация программ дополнительного образования (по полной и ускоренной программам) на базе очного или заочного обучения;

- осуществление гибких программ для курсов переподготовки и повышения квалификации работников предприятий, организаций, учреждений и органов административного управления городов и поселков юга Алтайского края;

- разработка и внедрение инновационных технологий обучения, исследования в области предпринимательской деятельности, экономики, региональных и муниципальных реформ.

Содержание образования в Институте ориентировано:

- на обеспечение самоопределения личности, создание условий для ее саморазвития;

- на развитие гражданского общества;

- на укрепление и совершенствование правового государства.

Профессиональное образование в Институте обеспечивает получение обучающимися профессий соответствующей квалификации. При этом гарантируется государственный образовательный стандарт.

Предприятие было создано для удовлетворения потребности в высококвалифицированных специалистах и для повышения уровня образования в регионе. В соответствии с заданными целями, предприятие вправе осуществлять следующие виды деятельности:

- предоставление услуг высшего образования;

- проведение курсов повышения квалификации;

- оказание платных библиотечных услуг;

- услуги передачи данных, информационные услуги, в том числе на основе организуемых компьютерных классов открытого доступа;

- услуги по разработке, производству и поставке программных продуктов, информационных систем, программных и методических средств;

- сервисных работ и услуг по техническому обслуживанию средств вычислительной техники, оргтехники;
- организация и проведение физкультурно-оздоровительных и спортивных мероприятий, создание спортивных секций;
- организация и предоставление культурно-бытовых услуг.

Финансирование деятельности Института осуществляется в установленном порядке за счет:

- средств федерального бюджета;
- средств, полученных от выполнения хозяйственных договоров;
- средств, полученных от подготовки студентов по хозрасчетным договорам;
- поступление от спонсоров, добровольных взносов, пожертвований граждан, организаций;
- других законных источников.

Организационная структура – совокупность звеньев, расположенных в строгой соподчиненности и обеспечивающих взаимосвязь между отдельными подразделениями организации, а также распределение между ними прав и ответственности. Она проявляется через разделение труда, создание специализированных подразделений, иерархию должностей и является необходимым элементом эффективной организации, т.к. придает ей внутреннюю стабильность и позволяет добиться определенного порядка в использовании ресурсов [18].

Организационная структура предприятия представляет собой строгую иерархию и представлена на рисунке 1.1.

На данный момент филиал располагается в двух учебных корпусах. Численный состав сотрудников филиала составляет 138 человек. Руководит филиалом правление, решения правления осуществляет директор. Директору непосредственно подчиняются заместители: по учебной работе, по научной работе и информатизации, по профориентационной работе, ведущий юрисконсульт, отдел по работе с персоналом, финансово-экономический отдел,

ведущий документовед, магазин, ведущий инженер, комендант.



Рисунок 1.1 – Структура управления РИ (филиала) АлтГУ

При рассмотрении, очевидно, что полномочия для управления конкретными сферами деятельности рассредоточиваются между заместителями директора. В целом рационального отношения организационной структуры и информационного аспекта управления трудно достичь без оптимального распределения системы управленческого аппарата, которое наглядно представлено в АлтГУ.

Параметры основных структурных элементов объекта.

Каждый отдел и структурное подразделение в филиале имеет свою функциональную особенность.

Отдел по работе с персоналом – проводит работу с персоналом, заключает договора, поддерживает социальную обеспеченность работника со стороны работодателя, численность отдела составляет 2 человека.

Отдел программного и технического обеспечения – обеспечение работоспособности ЛВС, ремонт и наладка оборудования в филиале, создание и освоение современного программного обеспечения. Модификация и поддержка

информационно-образовательного портала. Численность отдела составляет 10 человек.

Отдел по работе со студентами – работа со студентами, формирование групп, составление приказов, составление отчётов успеваемости, 6 человек.

Приемная комиссия – приём абитуриентов и проведение вступительных экзаменов. Не имеет постоянного состава, собирается по мере необходимости (1-4).

Бюро расписаний – составление план-графиков учебных процессов, распределение нагрузок среди преподавателей, расписание. 2 человека.

Библиотека – выдача учебной литературы студентам и сотрудникам, поддержка библиотечного фонда. 3 человека.

Финансово-экономический отдел – ведение бухгалтерского учёта и анализа предприятия, начисление заработной платы сотрудникам. 6 человек.

Кафедры – организация и проведение образовательного процесса согласно общегосударственным стандартам, 60 человек.

Центр дополнительных образовательных услуг организует дополнительные образовательные курсы, производит набор на курсы. 2 человека.

Дирекция – руководящий орган предприятия. 11 человек.

Служба коменданта и служба ведущего инженера. 34 человека

В настоящее время РИ (филиал) АлтГУ в г. Рубцовске предоставляет образовательные услуги высшего и средне-профессионального образования по следующим отделениям и специальностям:

Дневное и заочное отделения: «Экономика», «Государственное и муниципальное управление», «Прикладная информатика (в экономике)», «Юриспруденция», «Психология», «Менеджмент».

Отделение среднего профессионального образования: «Информационные системы (по отраслям)», «Экономика, бухгалтерский учет», «Правоохранительная деятельность», «Земельно-имущественные отношения».

Парк средств вычислительной техники (СВТ) РИ (филиала) АлтГУ

составляет 275 ед. персональных компьютеров (ПК). Из них 68 ПК на мобильных платформах и 38 ПК с процессорами Pentium III, остальные Pentium4. Совместно с данным оборудованием, в учебном и производственном процессах, используется 10 мультимедийных проекторов (5 стационарных), 45 ед. лазерных принтеров (1 цветной), 19 ед. планшетных сканеров, 16 ед. копировально-множительной техники.

Количество ПК, используемых в учебном процессе, составляет 275 ед. Аудиторный фонд Института, оснащенный СВТ, включает 10 компьютерных классов: 6 классов по 15 ПК в каждом, 2 – по 17 ПК, 2 – по 18 ПК. Из них 2 класса по 15 ПК используются в режиме свободного доступа студентов. Кроме того, 1 класс на 8 ПК используется для проведения лабораторных занятий по дисциплинам специальностей ВМ. Мобильный класс на ноутбуках (14 ед.), также активно используется в учебно-образовательной деятельности, как для тематических занятий, так и для организации доступа к ресурсам корпоративной сети и Internet на территории РИ (филиала) АлтГУ.

Все административные подразделения, кафедры, компьютерные классы и лаборатории объединены в единую локальную вычислительную сеть, построенную на основе традиционных структурированных кабельных системах (СКС), суммарной протяженностью около 12 км, со скоростями передачи данных: в сетях подразделений 1Гбит/с, конечных сегментах 100Мбит/с. Кроме этого, имеется 2 сегмента магистральных волоконно-оптических каналов связи, объединяющих между собой оба корпуса РИ (филиала) АлтГУ в пределах кампуса. Общая протяженность данной линии 1,58 км с пропускной способностью кабельной системы 1Гбит/с и возможностью расширения до 10 Гбит/с за счет модернизации активного сетевого оборудования. Один оптический сегмент используется как опорный канал для доступа пользователей к ресурсам локальной сети Института и внешним сетям; другой – для организации внутрикампусной цифровой IP-телефонии на основе голосовых шлюзов VoIP.

Необходимо отметить, что доступ к ресурсам ЛВС на территории РИ

(филиала) АлтГУ может осуществляться как по традиционным проводным технологиям, так и через беспроводные системы Wireless WiFi (до 100Мбит/с). В настоящее время существует следующая конфигурация основных серверов Института:

1. Файловый сервер RFAGU. Выполняет функции главного контроллера домена RBDOM, обеспечивающего централизацию управления ЛВС Института; файлового сервера личных и общедоступных ресурсов; СПС «Консультант+», «Гарант», «Электронная библиотека»). Технические характеристики: 2xP4Xeon 2,8GHz, 4x1Gb, IDE-SATA 40Gb,1Tb, RAID5 SATA 3x500Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux 2008 Server Edition.

2. Сервер WWW. Выполняет функции базового WEB-сервера со следующим набором сервисов: HTTP, FTP, DNS, MAIL, PROXY, APACHE, MySQL, DIALUP. Технические характеристики: 2xP4Xeon, 3,0GHz, 2x1Gb, RAID1 SATA 2x80Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux 2008 Server Edition.

3. Сервер GTW. Выполняет функции шлюза, обеспечивающего интеграцию сети Института в глобальные сети Internet. Здесь же установлен корпоративный межсетевой экран. Технические характеристики: PIII-Celeron 700MHz, 256Mb, IDE-ATA 40Gb, Fast Ethernet. Установленное ПО – ASP Linux 7.3.

4. Сервер RS-BANK. Выполняет функции сервера БД для используемых в учебном процессе банковских информационных систем «RS-Bank 5.1». Здесь же установлен Web-сервер Apache с модулем PHP 4.0, обеспечивающий on-line доступ к расписанию занятий, модуль FastReport, обеспечивающий формирование отчетов печати для приложений. Технические характеристики: P-IV Celeron 2,00GHz, 2Gb, IDE-SATA500Gb, Gigabit Ethernet. Установленное ПО – MS Windows 2003 AS.

5. Сервер ORACLE. Выполняет функции сервера БД для хранения информации для портала и информации необходимой для учебного процесса.

Технические характеристики: 2xP4Xeon, 4,0GHz,, 12Gb, 324Gb, Gigabit Ethernet. Установленное ПО – Gentoo Linux 2008 Server Edition.

Характеристика информационных потоков Института

Информация об абитуриенте попадает в базу данных по абитуриентам, это в первую очередь входная анкетная информация (фамилия, имя, отчество, адрес, условия поступления, специальность и т.д.), после успешной сдачи вступительных экзаменов (или наличия различных льгот), информация о поступивших студентах передаётся в базу данных по студентам. В ней отражается состояние успеваемости студента, академические долги. Информация о количестве студентов в группах, количестве групп, попадает в учебно-методический сервер, для составления учебного расписания. Готовая информация попадает на кафедры в виде составленных расписаний, как для студентов, так и для преподавателей.

База данных по студентам. Это программный продукт, разработанный сотрудниками Института, выполненный на основе СУБД Oracle. Входящей информацией для неё, является информация о поступлении студентов, передаваемая из базы данных по абитуриентам. В базе по студентам ведётся медицинский учёт, осуществляемый медиком; воинский учёт, осуществляемый отделом по работе с персоналом; отдел по работе со студентами вносит данные о студентах, группах, дисциплинах, успеваемости, академических задолженностях. Выходной информацией считается: списки групп, задолжников, приказы об отчислении, переводе на другой курс, специальность.

База по сотрудникам, также собственная разработка информационной системы ведения учёта работающего персонала. Входной информацией являются анкетные данные всех служащих. Ведение бухгалтерского учёта, начисления зарплаты, производится в системе «1С: Предприятие», используя данные из базы по сотрудникам, что позволяет отказаться от двойного ввода информации. Так же, «1С: Предприятие», используя данные из базы по студентам, производит учёт оплаты за обучение студентов.

Учебно-методический сервер, используя информацию из базы по

студентам, используя данные об учебных планах, рабочих программах, составляет учебное расписание, графики учебных процессов.

В структурном подразделении финансово-экономический отдел, существуют информационные потоки, такие как очередь информация о поступивших товарах, постановка их на учёт, счёт-фактуры. Выходной информацией являются акты о списании расходных материалов, накладные, отчёты о задолженностях студентов по оплате. Отчёты в инспекцию по налогам и сборам, в пенсионный фонд, так же считаются выходной информацией.

Все информационные ресурсы, доступные для общего пользования, можно получить практически с любого компьютера в Институте, благодаря использованию высокоскоростной локально-вычислительной сети. Также в двух корпусах филиала, установлены для общего пользования, отдельные компьютеры с сенсорным экраном, с помощью которых любой желающий может найти информацию о группах, студентах, их задолженностях, успеваемости, о преподавателях, а также узнать расписание.

Использование электронного библиотечного каталога в большой мере облегчает работу студентам с литературой. Электронный библиотечный каталог, позволяет найти нужную информацию о каком-либо издании, находящимся в библиотеке Института. Библиографическая информация о имеющихся книгах и периодических изданиях заносится в базу данных работниками библиотеки.

Официальный информационно-образовательный портал Рубцовского Института Алтайского государственного университета призван решать следующие задачи:

- обеспечивать построение единого информационного пространства университета на основе интеграции информационных ресурсов и сервисов структурных подразделений РИ (филиала) АлтГУ;

- обеспечивать повышение эффективности образовательной деятельности университета за счет использования дистанционных форм обучения;

- способствовать формированию целостного позитивного образа РИ (филиала) АлтГУ в стране и мире, представление информации о деятельности университета, его учебном и научном потенциале;
- объективно и оперативно информировать российское и мировое сообщества о наиболее значимых событиях, происходящих в РИ (филиале) АлтГУ;
- способствовать развитию научных и учебных связей с ВУЗами России, ближнего и дальнего зарубежья;
- обеспечивать доступ студентов, сотрудников и преподавателей к различным по представлению информационным ресурсам, нормативно-методическим материалам, системам промежуточного и выходного контроля знаний, а также системам управления качества образования, для повышения эффективности образовательной деятельности ВУЗа;
- оперативно предоставлять достоверную информацию абитуриентам университета.

Интернет сервер Института позволяет получить информацию о текущих новостях Института, информацию о расписании учебного процесса, провести голосование, оставить сообщение на форуме. Система тестирования позволяет провести дистанционное тестирование студентов по различным дисциплинам.

1.2 Анализ функционирования объекта исследования

Системный анализ – научный метод познания, представляющий собой последовательность действий по установлению структурных связей между переменными или элементами исследуемой системы [2].

Моделирование бизнес-процессов необходимо для выявления текущих проблем на предприятии и предвидения будущих. Моделирование деловых процессов, как правило, выполняется с помощью CASE-средств. К таким средствам относятся AllFusion ERwin Data Modeler (CA Technologies), Oracle Designer (Oracle), Rational Rose (IBM) и т.д.

Моделирование предметной области будет выполнено при помощи CASE-средства AllFusion ERwin Data Modeler. Данная программа поддерживает три методологии моделирования:

- функциональное моделирование (IDEF0);
- описание бизнес-процессов (IDEF3);
- диаграммы потоков данных (DFD).

IDEF0 (Function Modeling) – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Одним из основных понятий стандарта IDEF0 является декомпозиция (Decomposition). Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции.

Модель IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, уходящими за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется контекстной диаграммой.

В пояснительном тексте к контекстной диаграмме должна быть указана цель (Purpose) построения диаграммы в виде краткого описания и зафиксирована точка зрения (Viewpoint). Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь. Точка зрения определяет основное направление развития модели и уровень необходимой детализации.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему, а каждый из функциональных блоков, принадлежащих

дочерней диаграмме, соответственно называется дочерним блоком (Child Box).

В основе функционального моделирования лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий.

Функциональный блок (Activity Box) представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении. На диаграмме функциональный блок изображается прямоугольником. Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), при этом:

- верхняя сторона имеет значение «Управление» (Control);
- левая сторона имеет значение «Вход» (Input);
- правая сторона имеет значение «Выход» (Output);
- нижняя сторона имеет значение «Механизм» (Mechanism).

Интерфейсная дуга (Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира или потоки данных и информации.

В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей».

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram) [25].

DFD (Data Flow Diagramming) – диаграммы потоков данных представляют собой иерархию процессов, которые связаны между собой потоками данных. Диаграммы показывают, как обрабатывает информацию

каждый процесс, как процессы связаны друг с другом, а также как работает сама система, каким образом она обрабатывает поступающие данные.

Основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- накопители данных или хранилища;
- процессы;
- потоки данных;
- системы/подсистемы.

Внешняя сущность представляет собой материальный объект или физическое лицо, которые могут выступать в качестве источника или приемника информации. Обозначается прямоугольником с тенью, внутри которого указывается ее имя. При этом в качестве имени рекомендуется использовать существительное в именительном падеже. Иногда внешнюю сущность называют также терминатором.

Процесс представляет собой совокупность операций по преобразованию входных потоков данных в выходные в соответствии с определенным алгоритмом или правилом. Хотя физически процесс может быть реализован различными способами, наиболее часто подразумевается программная реализация процесса. Обозначается прямоугольником с закругленными вершинами, разделенным на три секции или поля горизонтальными линиями. Поле номера процесса служит для идентификации последнего. В среднем поле указывается имя процесса. В качестве имени рекомендовано использовать глагол в неопределенной форме с необходимыми дополнениями. Нижнее поле содержит указание на способ физической реализации процесса.

Информационная модель системы строится как некоторая иерархическая схема в виде так называемой контекстной диаграммы, на которой исходная модель последовательно представляется в виде модели подсистем соответствующих процессов преобразования данных. При этом подсистема или система на контекстной диаграмме DFD изображается так же, как и процесс – прямоугольником с закругленными вершинами [23].

IDEF3 (Integrated DEFinition for Process Description Capture Method) – предлагает структурный метод описания процессов. Модель описывается как упорядоченная последовательность событий. Методика IDEF3 хорошо приспособлена для сбора данных. Данная методика не имеет жёстких синтаксических и семантических ограничений. Очень часто IDEF3 используют как метод, дополняющий IDEF0. Каждый функциональный блок IDEF0 может быть представлен в виде отдельного процесса.

Действие в IDEF3 называется единицей работы (Unit of Work, UOW) и обозначается прямоугольником. Действия именуются глаголами или отглагольными существительными. Каждому действию назначается уникальный номер.

С помощью связей выделяются существенные взаимоотношения между действиями, задавая их последовательность. Все связи однонаправленные. Обычно стрелки рисуют слева направо, выходящими из правой и входящими в левую сторону блоков, либо сверху вниз (что является лишь соглашением, а не нормируется).

Выделяют три вида связей:

- временное предшествование показывает, что исходное действие должно завершиться прежде, чем начнётся выполнение конечного действия, а также, что исходное действие может инициировать выполнение конечного действия;
- объектный поток применяется в случае, когда объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия;
- нечёткое отношение используется, если невозможно применить предыдущие типы связей.

При помощи связи типа Нечёткое отношение можно, например, описывать параллельные процессы.

Одно действие может порождать несколько. Или для выполнения действия требуется завершение нескольких действий. Для описания ветвлений

процессов используют соединения:

- асинхронное «И» (Asynchronous AND);
- синхронное «И» (Synchronous AND);
- асинхронное «ИЛИ» (Asynchronous OR);
- синхронное «ИЛИ» (Synchronous OR);
- исключающее «ИЛИ» (XOR – Exclusive OR) [15].

В Рубцовском Институте (филиале) АлтГУ функционирует информационно-образовательный портал, который предоставляет студентам и преподавателям необходимую информацию. На портале существует множество сервисов, которые обычным пользователем используются крайне редко. Для доступа к portalу существуют 2 версии сайта – десктопная и мобильная.

На рисунке 1.2 представлена контекстная диаграмма процесса «Организация доступа к ресурсам информационно-образовательного портала РИ (филиала) АлтГУ» AS-IS.

Главная функция делится на 7 основных подфункций:

- доступ к библиотеке;
- доступ к информации для абитуриентов;
- доступ к расписаниям;
- доступ к информации об образовании;
- доступ к новостной ленте;
- доступ к персональной странице;
- доступ к формируемым отчетам.

Подфункции портала представлены на рисунке 1.3.

После того, как модель системы построена ее необходимо верифицировать. Все ее объекты должны быть тщательно описаны и детализированы.

Способ распространения информационных ресурсов посредством веб-портала имеет ряд недостатков:

- отсутствие способа для хранения полученной информации в памяти устройства;

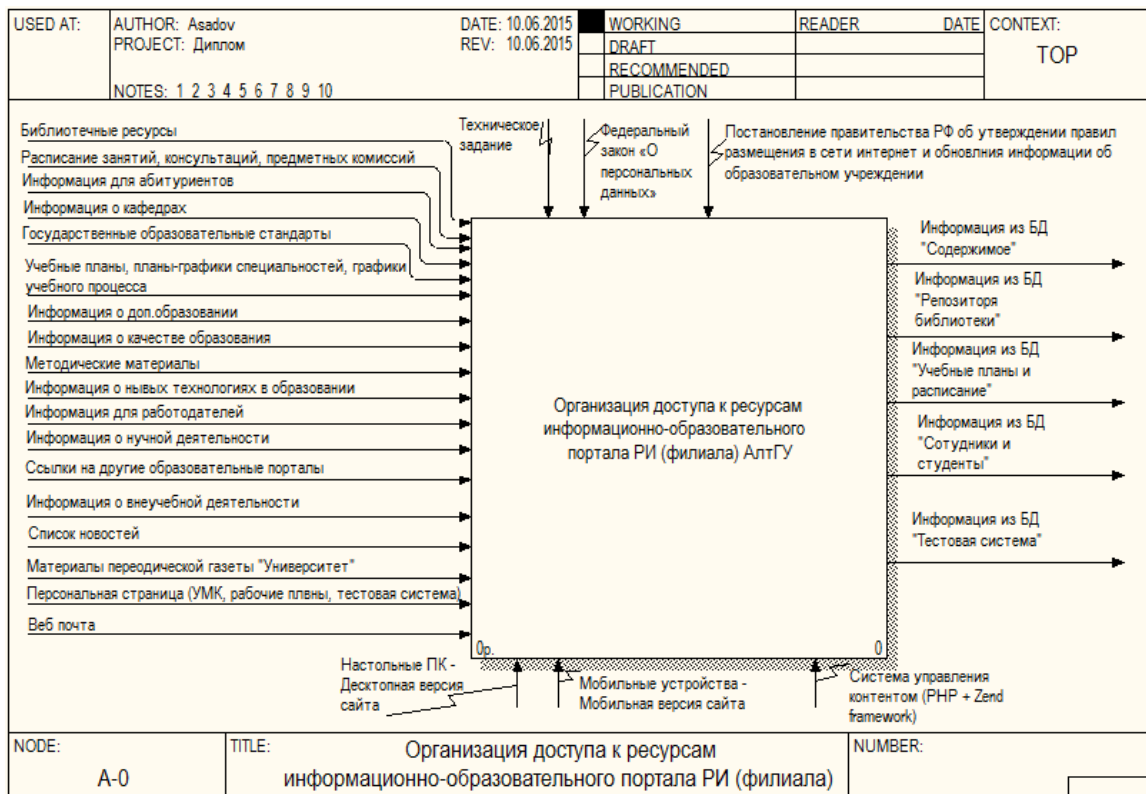


Рисунок 1.2 – «Контекстная диаграмма процесса «Организация доступа к ресурсам информационно-образовательного портала РИ (филиала) АлтГУ»
(Как есть)

- отсутствие возможности работать в автономном режиме – без интернета;
- неудобный пользовательский интерфейс;
- отсутствие интерактивного взаимодействия с пользователем;
- отсутствие возможности кастомизации.

Еще одним способом доступа к ресурсам информационно-образовательного портала является использование мобильного приложения, данный способ немаловажен для низкоскоростных соединений и дорогого мобильного трафика. Мобильные приложения могут работать в режиме офлайн – без наличия интернета.

Исходя из этого, было принято разработать мобильное приложение.

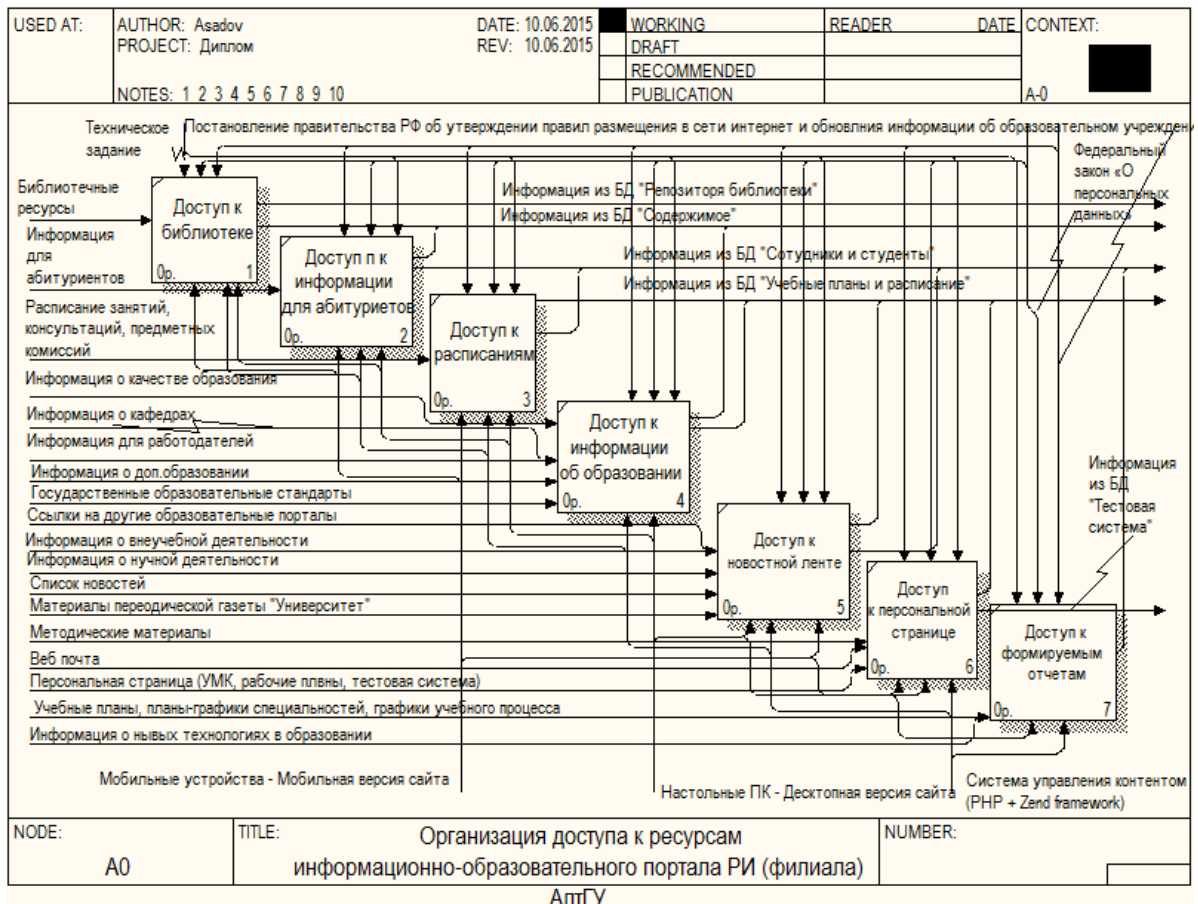


Рисунок 1.3 – Диаграмма декомпозиции процесса «Контекстная диаграмма процесса «Организация доступа к ресурсам информационно-образовательного портала ПИ (филиала) АлтГУ» (Как есть)

1.3 Определение цели и задач проектирования мобильного приложения

Мобильное приложение – последовательность инструкций, предназначенных для исполнения операционной системой мобильного устройства. Способ получения информации посредством мобильного приложения позволяет использовать настраиваемые параметры. Такими параметрами могут быть: автоматическое скачивание, получение интерактивных уведомлений, задание пользовательских настроек и т.д. Данные факторы обеспечивают удобство от использования мобильного приложения.

В связи с этим можно выделить цель: разработать мобильное

приложение, которое бы обеспечивало пользователей удобным и постоянным доступом к необходимой им информации, а именно к своему расписанию, расписанию консультаций и комиссий.

Исходя из цели, можно выделить задачи, решаемые мобильным приложением:

- получение расписания занятий, консультаций и комиссий;
- корректное отображение информации на любых экранах мобильных устройств;
- обновление списков групп, преподавателей и аудиторий;
- возможность сохранения пользовательских настроек;
- автоматическая фоновая загрузка выбранного расписания;
- наличие виджета;
- оповещение об изменившемся расписании;
- экономия мобильного трафика.

Приложение «РИ АлтГУ – Расписание» будет интегрировано с уже действующим информационно-образовательным порталом и являться дополнительным, более удобным, способом доступа к ресурсам информационно-образовательного портала.

1.4 Обзор и анализ существующих разработок, выбор технологии проектирования

На данный момент существуют разнообразные технологии для разработки мобильных приложений – от создания HTML-приложения до нативного Java-приложения. Самый простой способ создания мобильных приложений – создание HTML-приложений.

HTML-приложения, иными словами – сайт, упакованный специальным образом. Данная технология использует системный веб-браузер для работы приложения. Для создания приложения разработчику необходимо знать язык

гипертекстовой разметки – HTML, JavaScript и умение работать с фреймворками.

Фреймворки – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта [33].

Основными фреймворками для создания мобильных HTML-приложений являются *Sencha Touch* и *Jquerymobile*.

Основная задача *Sencha Touch* (ST) по мере возможности мимикрировать стандартный интерфейс и поведение элементов интерфейса мобильной платформы, предоставив программистам гибкое API. Соответственно, ST-приложение – это обычная клиент-серверная разработка, в качестве клиента выступает браузер Safari или Chrome, или какой-либо другой webView, а в качестве сервера – любой сервис, способный «отвечать» в форматах JSON, XML.

ST фреймворк представляет связку HTML5 + JS + CSS. Фреймворк предоставляет гибкие средства для разработки собственных UI компонентов, как на основе уже существующих, так и с нуля. Любой компонент библиотеки по факту является ST-объектом с набором методов и свойств.

К достоинствам *Sencha* можно отнести:

- гибкость создания собственного интерфейса, посредством создания собственных компонент (более строгие требования к компонентам);
- документация для быстрого старта;
- графическая среда разработки интерфейсов: *Sencha Architect*;
- низкий порог вхождения [34].

Стоимость фреймворка – 108044,75 р.

Jquerymobile – JavaScript фреймворк, который позволяет быстро отстраивать веб-сайт для мобильных устройств. Это сенсорно-оптимизированный веб-фреймворк разработанный для смартфонов и планшетов. jQuery Mobile работает на подавляющем большинстве современных ПК, смартфонах, планшетах и e-ридер платформах. Фреймворк jQuery Mobile

прост в использовании и включает веб-средства управления, такие как кнопки, прокрутку, список элементов и многое другое. Каждый макет и элемент интерфейса, включенный в jQuery Mobile, строится вокруг нового объектно-ориентированного CSS фреймворка. Фреймворк позволяет применять законченный единый визуальный дизайн темы для мобильных сайтов и приложений.

Преимуществами создания приложений по *HTML* технологии является кроссплатформенность.

Недостатки HTML-приложений:

- основные возможности – возможности веб-браузера;
- слабая производительность [35].

Такое сочетание достоинств и недостатков не подходит для создания удобного и производительного приложения.

Цель же данного проекта – разработка программного продукта, который удовлетворяет запросам и возможностям пользователя. Гораздо выгоднее разработать приложение, которое будет поддерживать все заявленные функциональные возможности мобильной операционной системы, а не использовать лишь определенные, уже существующие, возможности обычных веб-браузеров.

Поэтому, в качестве технологии разработки мобильного приложения была выбрана разработка на языке программирования – Java, который является нативным языком для разработки приложений на платформе Android.

Преимуществами такого выбора являются:

- скорость работы приложения;
- использование самых новых технологий в области разработки приложений.

Предварительно было произведено сравнение интегрированных сред для разработки мобильного приложения по стоимости, гибкости и возможностям.

На данный момент для разработки Android приложений существует 3 основных программных продукта – Android Studio, Eclipse (с плагином ADT) и

Embarcadero RAD Studio XE 6+.

Android Studio – интегрированная среда разработки для работы с платформой Android. Создана совместно с компанией JetBrains на продукте IntelliJ IDEA. Данная система была спроектирована специально для Android разработки и доступна на платформах: Windows, Mac OS X и Linux. Данная среда разработки создана для замены Eclipse Android Development Tools (ADT).

Преимущества:

- визуальный редактор форм, редактирование интерфейса в реальном времени;
- система автоматической сборки Gradle;
- Android-ориентированные инструменты рефакторинга;
- встроенные инструменты проверки совместимости с прошлыми версиями;
- инструменты для выявления проблем с производительностью
- система сокращения и оптимизации кода ProGuard;
- поддержка Google Cloud [36].

Android Studio распространяется бесплатно.

Eclipse (с плагином ADT) – свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Eclipse служит в первую очередь платформой для расширений, чем он и завоевал популярность: любой разработчик может расширить Eclipse своими модулями. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT), разрабатываемые инженерами QNX совместно с IBM, и средства для языков Ada (GNATbench, Hibachi), COBOL, FORTRAN, PHP, X10 (X10DT) и пр. от различных разработчиков. Множество расширений дополняет среду Eclipse диспетчерами для работы с базами данных, серверами приложений и др.

Eclipse написана на Java, потому является платформо-независимым продуктом, за исключением библиотеки SWT, которая разрабатывается для всех распространённых платформ. Библиотека SWT используется вместо

стандартной для Java библиотеки Swing. Она полностью опирается на нижележащую платформу (операционную систему), что обеспечивает быстроту и натуральный внешний вид пользовательского интерфейса, но иногда вызывает на разных платформах проблемы совместимости и устойчивости приложений.

Android Development Tools (ADT) – плагин для интегрированной среды разработки Eclipse, который расширяет возможности Eclipse и позволяет быстро создать новый Android проект, спроектировать интерфейс приложения, добавлять необходимые библиотеки, отладить приложение используя инструменты программной разработки, создать и подписать .apk-файл приложения для дальнейшего распространения [37].

Eclipse с плагином ADT распространяется бесплатно.

Embarcadero RAD Studio – среда быстрой разработки приложений (RAD). Это комплексное решение для разработки программного обеспечения, ускоряющее создание нативных приложений для Windows, Mac и Android на базе единого исходного кода. Данная платформа позволяет разрабатывать высокопроизводительные, компилированные нативные приложения для простого подключения к корпоративным данным, облачным службам, устройствам, датчикам и гаджетам.

Возможности:

- наличие компилятора Delphi Android ARM для устройств и эмулятора;
- платформа FM platform для создания «нативных» приложений под Android для Gingerbread (2.3.3 – 2.3.7), Ice Cream Sandwich (4.0.3, 4.0.4) и Jelly Bean (4.1.x, 4.2.x, 4.3.x*);
- элемент управления Time Picker для Android, iOS, Windows и OS X;
- компонент Notification Center для Android, iOS и OS X;
- встроенная поддержка поисковой фильтрации для TListView на Android, iOS, Windows и OS X;
- жест «смахивания» для удаления на Android и iOS;
- поддержка «Share sheet» на Android и iOS;

- оптимизация производительности платформы FMDelphi RTL для Android;
- Deployment Manager для Android;
- поддержка запуска на эмуляторе для Ice Cream Sandwich и Jelly Bean;
- запуск приложений на устройствах под Android (Debug / App Store);
- удалённая отладка для Android [38].

Стоимость RAD STUDIO XE8 PROFESSIONAL – 101514,83 р.

Сравнение стоимости программных средств представлено в таблице 1.1.

Таблица 1.1 – Стоимость программных средств

	HTML	Android Studio	Eclipse + ADT	RAD STUDIO
Стоимость	108044,75 р.	Бесплатно	Бесплатно	101514,83 р.

Android Studio оказалась наиболее оптимальным выбором, т.к. она более гибкая для расширения функциональных возможностей приложения, по сравнению с остальными рассмотренными выше IDE, а также она является официальной IDE, предоставляемой, компанией, являющейся разработчиком системы Android – Google.

1.5 Выбор и обоснование проектных решений

Проектируемое приложение «РИ АлтГУ – Расписание» будет предоставлять оперативную информацию о расписании Рубцовского Института (филиала) Алтайского Государственного Университета.

Цель создания данного приложения – повышение эффективности и удобства работы с расписанием, что повлечет к сокращению случаев, связанных с отсутствием расписания и опозданиями.

Проектируемое приложение будет иметь клиент-серверную архитектуру:

- Серверная часть:

- база данных Oracle;
- язык программирования PHP, с помощью текстового редактора Sublime Text.

- Клиентская часть:
 - Java, с помощью интегрированной среды разработки Android Studio.

Для реализации приложения необходимо указать все ограничивающие факторы, присутствовавшие во время проектирования. Все факторы можно разделить на три категории:

- информационные (оперативное отображение всех изменений объектов управления);
- организационные (используемое программно-аппаратное обеспечение);
- экономические (экономия средств реализации проекта) [20].

Информационное обеспечение (ИО) включает в себя: систему классификации и кодирования; систему унифицированной документации, используемой в ИО; информационную базу, т.е. информацию из всех информационных массивов. Центральным компонентом являются БД, через которые осуществляется обмен данными различных задач. БД обеспечивают интегрированное использование различных информационных объектов в функциональных подсистемах [19].

Использование вычислительной техники базируется на обеспечении эффективного доступа к ее ресурсам. Это правило формируется на основе следующих принципов:

- принцип однократной фиксации данных (реализуется через запрет дублирования данных);
- использование для фиксирования информации исключительно первичной документации (именно в такой документации содержатся самые точные для обработки данные);
- полноты информации, то есть информационный фонд должен

содержать всю информацию для решения задач, для ответов на нерегламентированные запросы и для принятия управленческих решений;

- целостности информации, то есть в информационном фонде должны быть разработаны средства обеспечения достоверности хранимой информации, средства поддержки непротиворечивости данных, обеспечения своевременности актуализации данных, обеспечения защиты данных от технических поломок машины;

- гибкости хранения данных, то есть в информационном фонде должны быть средства увеличения и уменьшения хранимого количества данных;

- доступности информации за счет развитых средств диалога пользователя с базой, оперативности выдачи ответа на запрос [1].

Существует три способа организации информационной базы (ИБ): файловая организация ИБ; интегрированная ИБ, смешанная организация ИБ.

Наиболее целесообразной организацией ИБ является интегрированная база данных, которая представляет собой совокупность взаимосвязанных и хранящихся вместе данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для любых приложений и при этом обеспечивается независимость данных от программы, а для актуализации данных используется общий способ управления [3].

Существует три классических модели логической структуры базы данных (по способу установления связей между данными): иерархическая, сетевая и реляционная.

Реляционная модель основана на математической логике и является простейшей и наиболее привычной формой представления данных в виде таблицы, поэтому в проекте реализуется именно этот способ организации хранения данных.

Преимущества использования реляционных баз данных состоит в следующем: в реляционной модели данных существует всего одна информационная конструкция, которая формализует табличное представление данных, привычное для пользователей. Теоретическое обоснование – наличие

теоретически обоснованных методов нормализации отношений позволяет получать базы данных с заранее заданными свойствами (в основном, с гарантией минимальной избыточности представления данных).

Независимость данных – когда необходимо изменить структуру реляционной базы данных, то это приводит к минимальным изменениям в программном продукте.

Программное обеспечение (ПО) – совокупность программ системы обработки данных и программных документов, необходимых для эксплуатации этих программ. ПО предназначено для придания вычислительной системе определенных свойств, связанных с увеличением производительности, повышением достоверности получаемых результатов, повышением надежности функционирования системы, улучшения работы пользователя.

Классификация методов проектирования ПО:

- метод «Сверху-Вниз»;
- метод «Снизу-Вверх»;
- метод структурного проектирования;
- метод структурного программирования;
- метод группы главного программиста;
- метод модульного проектирования [16].

В работе будет использован метод проектирования «сверху-вниз». Метод вначале предусматривает определение задачи в общих чертах, а затем постепенное уточнение структуры путем внесения более мелких деталей. Построение программы представляет собой последовательность шагов такого уточнения.

Первоначально рекомендуется описать задачу на естественном языке. На каждом последующем шаге необходимо выявить основные функции, которые нужно выполнить, то есть данная задача разбивается на ряд подзадач до тех пор, пока они не станут настолько простыми, что каждой из них будет соответствовать один модуль, который можно записать на алгоритмическом языке, фразой.

Далее должны будут описаны данные, отмечены основные структуры и вид обработки данных. Каждый модуль должен будет иметь тестовые данные, описываемые вместе с модулем, которые впоследствии будут использованы на этапе тестирования.

Общее (базовое, системное) ПО включает в себя операционные системы, системы программирования и сервисные программы, которые предназначены для управления работой процессора, организации доступа к памяти, периферийным устройствам и сети, для управления файлами, запуска прикладных программ и управления процессом их выполнения, а также для обеспечения выполнения программ на языках высокого уровня.

Системным ПО в работе будет использоваться ОС Windows. В комплекс программ функционального (прикладного, специального) ПО будут входить уникальные программы и функциональные пакеты прикладных программ (ППП). Функциональное ПО предназначено для автоматизации решения функциональных задач отдела, а также связанных с ними некоторых оригинальных задач.

В работе будут использованы пакеты прикладных программ, предназначенные для проектирования и разработки программных средств: СУБД Oracle, текстовый редактор Sublime Text и свободная интегрированная среда разработки приложений Android Studio.

Операции сбора и регистрации данных будут осуществляться с помощью различных средств, отображающих реальное состояние объекта на носители информации.

Заключительным этапом обработки информации является выдача результатной информации пользователю. Она будет осуществляться непосредственно на экран мобильного устройства.

Обоснование выбора технического обеспечения (ТО).

Техническое обеспечение на реализацию проекта предполагает наличие сервера с установленной системой управления базой данных (СУБД) Oracle.

В качестве серверов будут использоваться существующие серверы в

парке вычислительной техники университета.

В дальнейшем программное обеспечение мобильного решения может быть доработано требующейся функциональностью.

В качестве операционной системы для разработки приложения была выбрана система Android, т.к. она является самой распространенной операционной системой на мобильных устройствах.

2 Проектная часть

2.1 Разработка функционального обеспечения

Выявленные для пользователей мобильных устройств недостатки в работе действующей мобильной версии информационно-образовательного портала приводят к необходимости построения модели «Как должно быть» (модель ТО-ВЕ).

Задачей описания модели ТО-ВЕ является нахождение мер блокирования отрицательного влияния неудовлетворительных бизнес-факторов, найденных при анализе. Полученная модель представлена на рисунке 2.1.

На данной модели видно, что мобильное приложение будет содержать функционал, предоставляющий доступ к расписанию предметов, консультаций и комиссий.

Это уменьшит избыточность информации для данного решения и ускорит удобство работы с необходимой информацией.

На рисунке 2.2 представлена декомпозиция функций «Как должно быть».

На диаграмме изображено как поменялась организация доступа к информации. Изменения коснулись входных, выходных данных, механизма и управления. Из полученной диаграмм были удалены, лишние функциональные блоки, такие как:

- доступ к информации об образовании;
- доступ к формируемым отчетам;
- доступ к библиотеке;
- доступ к новостной ленте;
- доступ к информации для абитуриентов;
- доступ к персональной странице.

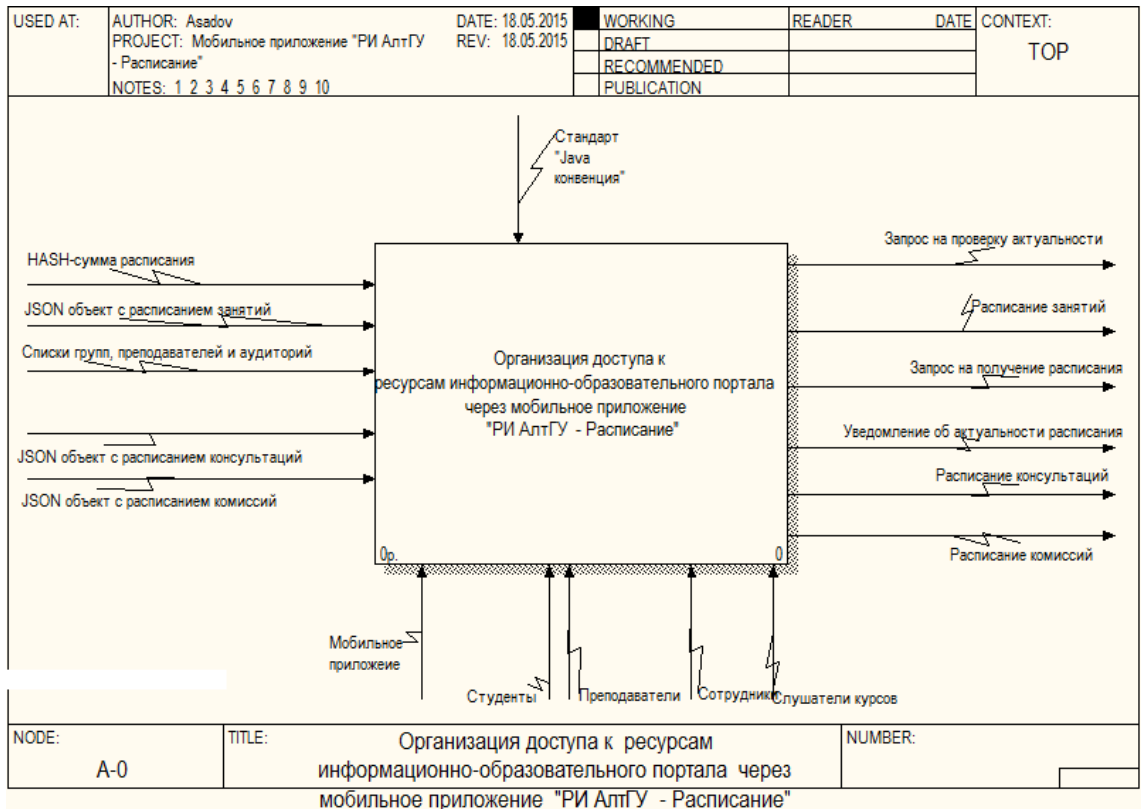


Рисунок 2.1 – Контекстная диаграмма «Как должно быть»

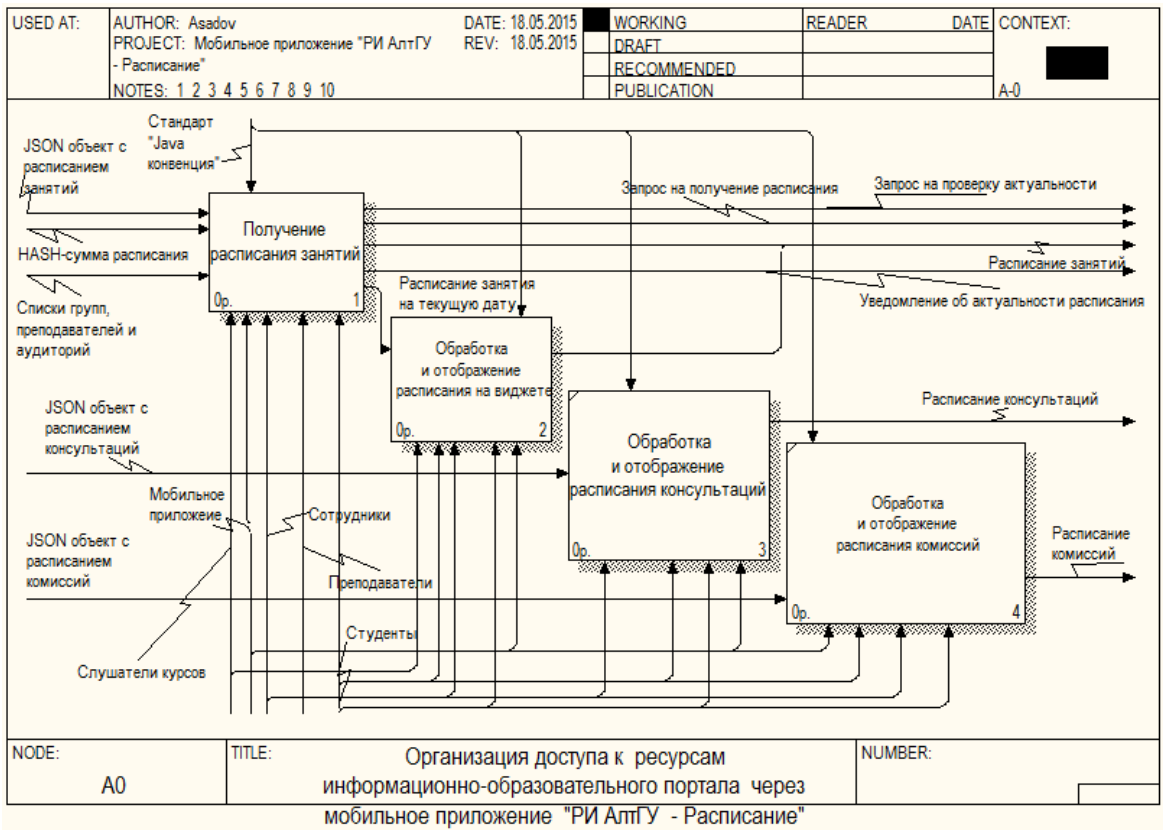


Рисунок 2.2 – Диаграмма декомпозиции «Как должно быть»

2.2 Разработка информационного обеспечения

2.2.1 Используемые классификаторы и системы кодирования

При реализации проекта для обеспечения уникальности объектов в пределах класса применяется порядковая система кодирования объектов. При этом все множество объектов предварительно не упорядочивается. Такой подход обусловлен использованием объектного подхода на базе СУБД Oracle. В результате чего всю работу по обеспечению максимальной скорости поиска объекта и определения его принадлежности во время сложной логической обработки с использованием имеющихся данных берет на себя СУБД [14].

Таким образом, некоторые таблицы имеют в своем составе автоинкрементное поле – идентификатор. Это позволяет не контролировать процессы генерации и присвоения уникальных кодов для записей программно, а полностью предоставить этот процесс под контроль СУБД.

Все даты представляются в стандартном для России формате «ДД.ММ.ГГГГ», где ДД – день месяца от 0 до 31, ММ – месяц от 0 до 12, ГГГГ – год в четырехзначном представлении.

Вся информация в базе данных представлена в кодировке UTF-8 – распространённой кодировке, реализующей представление Юникода, совместимое с 8-битным кодированием текста, позволяющая представить знаки практически всех письменных языков.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Вся справочная информация единого информационного пространства может быть представлена в виде объектов, порожденных на основе классов, имеющих в своей структуре следующие свойства: идентификатор объекта, собственно справочную информацию и ряд специфических, дополнительных

свойств. Методами у таких объектов могут быть добавление новой справочной информации, изменение существующей и удаление.

Для обеспечения функционирования информационных систем и портала в Институте реализована структура справочников. Все методы объектов реализованы с помощью программных средств организации, а так же с использованием существующей системы управления содержимым сайта.

Для исследуемой предметной области можно выделить следующие основные справочники:

- студенты;
- преподаватели;
- учебные планы;
- предметы.

2.2.3 Характеристика результатной информации

Так как приложение будет иметь клиент-серверную архитектуру, в качестве результатной информации работы информационно-образовательного портала (сервера) для мобильного приложения (клиента) будут выступать:

- списки групп, аудиторий и преподавателей;
- расписание занятий (представляет собой интерактивную форму выбора вывода расписания по различным параметрам; предназначено в основном для преподавателей и студентов);
- расписание консультаций (форма вывода информации о проведении консультационных занятий преподавателей по времени и аудитории, предназначено соответственно для студентов и преподавателей);
- список предметных комиссий со временем их проведения и составе, сгруппированный по кафедрам; предназначен для студентов и преподавателей.

Все перечисленные результатные сведения будут передаваться мобильному устройству для дальнейшей обработки и представления в удобном для пользователя виде.

В качестве формата для передачи данных мобильному устройству могут быть такие форматы, как: JSON, XML, YAML и INI.

Основными требованиями к формату стали: удобство редактирования, скорость парсинга, скорость сериализации, размер в сериализованном виде, распространённость и поддержка редакторами.

Результаты сравнения форматов представлены в таблице 2.1.

Таблица 2.1 – Сравнение форматов данных

Свойство	JSON	XML	YAML	INI
Человекопонятность	3	1	4	5
Удобство редактирования	3	1	4	5
Произвольная иерархия	3	3	3	1
Простота реализации	3	2	1	5
Скорость парсинга/сериализации	3	1	1	5
Размер в сериализованном виде	3	1	4	5
Поддержка поточной обработки	0	0	5	5
Бинарная безопасность	3	0	0	0
Распространённость	5	5	3	3
Поддержка редакторами	5	5	3	5
Поддержка языками программирования	5	5	3	5

В итоге, исходя из показателей, в качестве формата для передачи мобильному устройству выбран – JSON.

Мобильное приложение, посредством POST-запроса, будет передавать необходимые параметры серверу, а в качестве результатной информации будет получать ответ в формате JSON.

На рисунке 2.3 показаны два способа доступа к ресурсам информационно-образовательного портала: как на данный момент осуществлено взаимодействие между информационно-образовательным порталом и цифровыми устройствами, посредством веб-сайта, и как будет осуществлено взаимодействие между информационно-образовательным

порталом и мобильным приложением.

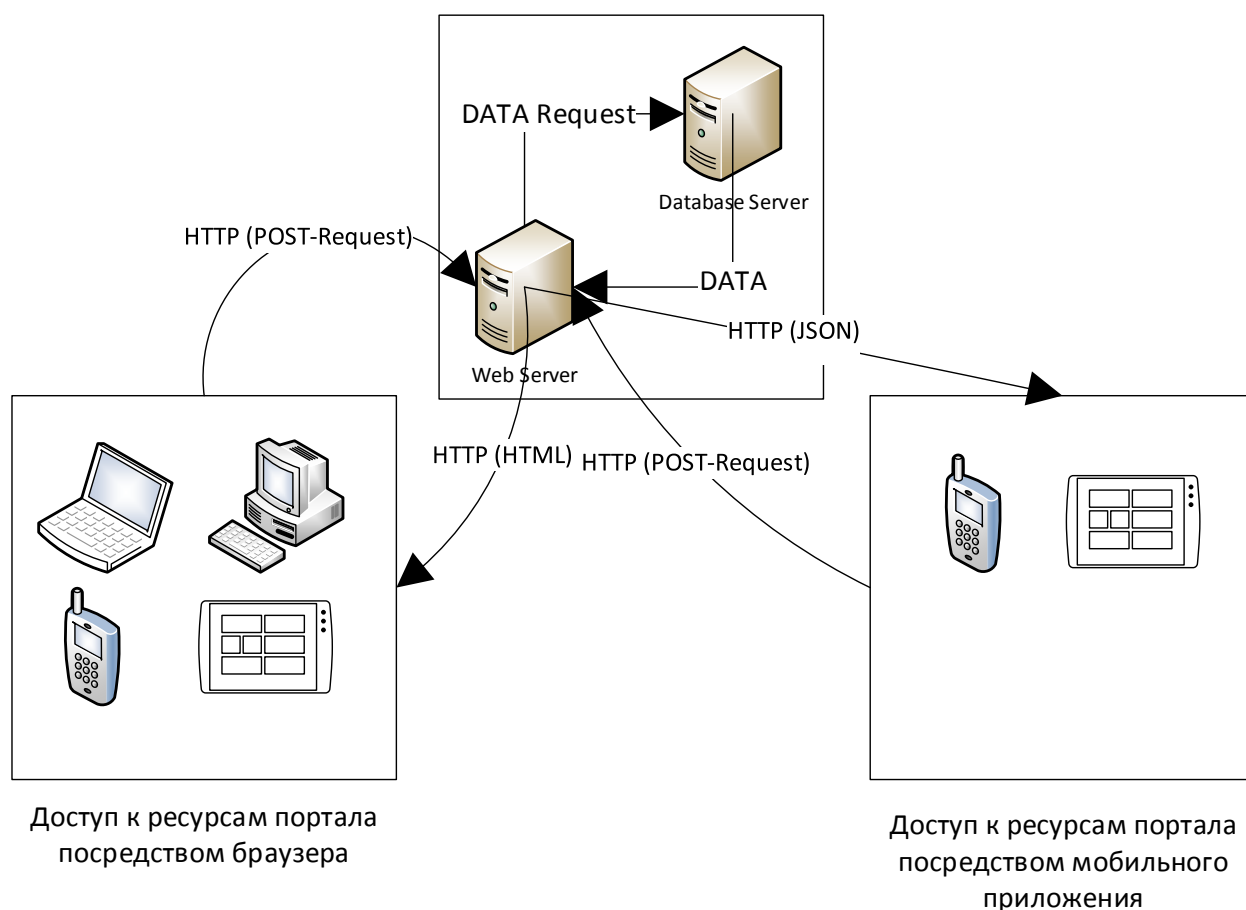


Рисунок 2.3 – Способы доступа к ресурсам информационно-образовательного портала

Использование формата данных JSON позволит сократить мобильный трафик и упростить обработку резульатной информации за счет того, что передаваемый JSON-объект содержит только данные о расписании, в отличие от использования веб-сайта, когда клиенту передается весь исходный код страницы, включая данные разметки, картинки, компоненты и т.д.

2.2.4 Структура резульатной информации

В качестве резульатной информации на мобильное устройство будет отправляться JSON-объект.

JSON – текстовый формат обмена данными, основанный на JavaScript.

Несмотря на происхождение от JavaScript (точнее, от подмножества языка

стандарта ECMA-262 1999 года), формат считается языконезависимым и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, в таком ключе он уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-интерфейсы).

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

- набор пар ключ: значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив. Ключом может быть только строка, значением – любая форма;
- упорядоченный набор значений. Во многих языках это реализовано как массив, вектор, список или последовательность.

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования.

В качестве значений в JSON используются структуры:

- объект – это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми.
- массив (одномерный) – это упорядоченное множество значений. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми.
- значение может быть строкой в двойных кавычках, числом, объектом,

массивом, одним из литералов: true, false или null. Т.о. структуры могут быть вложены друг в друга.

– строка – это упорядоченное множество из нуля или более символов юникода, заключенное в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» [39].

Перед передачей JSON необходимо определиться с его структурой. Структура JSON должна удовлетворять заданным требованиям: человекопонятность, удобство редактирования, удобство для парсинга.

Передаваемые данные должны полностью описывать передаваемое расписание. Поэтому необходимо продумать из каких данных будет формироваться JSON-файл.

Для расписания занятий необходимо передавать следующие данные:

- дата занятия;
- порядковый номер занятия;
- ID аудитории;
- номер аудитории;
- название групп;
- ID групп;
- тип занятия;
- название дисциплины;
- ID преподавателя;
- ученая степень преподавателя;
- должность преподавателя;
- фамилия имя отчество преподавателя;
- начало занятия;
- окончание занятия.

Структура файла для передачи расписания занятий представлена на рисунке 2.4.

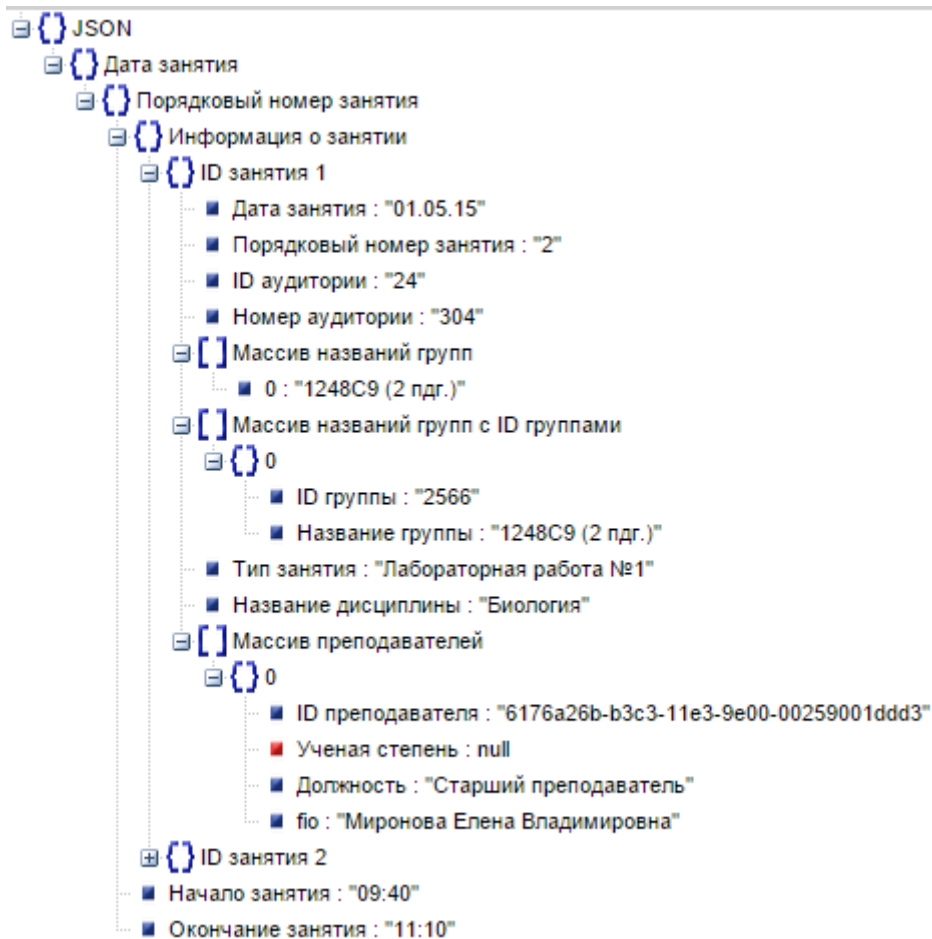


Рисунок 2.4 – Структура JSON передаваемого расписания занятий

Для расписания консультаций необходимо передавать следующие данные:

- ID консультации;
- дата консультации;
- номер консультации;
- ID аудитории;
- название аудитории;
- корпус;
- тип;
- ID преподавателя;
- фамилия имя отчество преподавателя;
- ученая степень;
- должность;

- начало консультации;
- окончание консультации.

Структура файла для передачи расписания консультации представлена на рисунке 2.5.

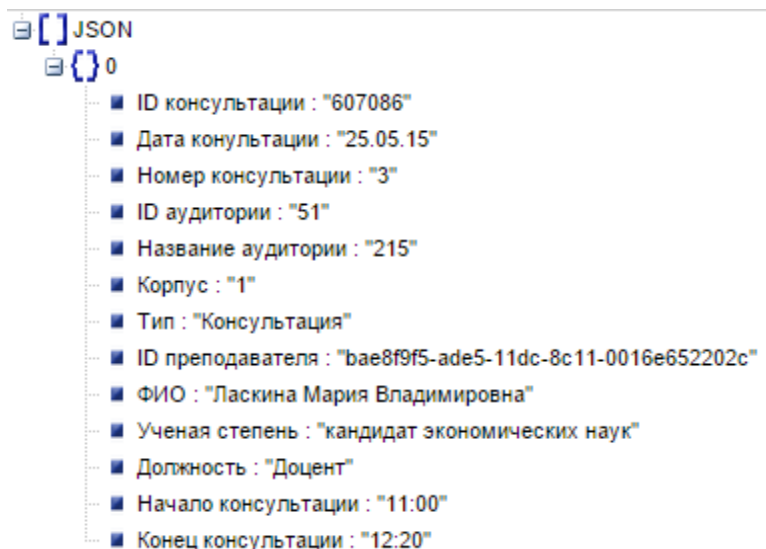


Рисунок 2.5 – Структура JSON передаваемого расписания консультаций

Для расписания комиссий было принято передавать 2 JSON-объекта, один из которых будет содержать данные о преподавателях и ID комиссии, а другой все данные, относящиеся к описанию комиссии.

JSON с информацией о преподавателе будет состоять из:

- Фамилия Имя Отчество преподавателя;
- ID комиссии.

JSON, описывающий комиссию будет содержать следующие данные:

- ID комиссии;
- кафедра;
- дата комиссии;
- время;
- аудитория;
- название дисциплины.

Структура файла для передачи преподавателей комиссий представлена на рисунке 2.6.

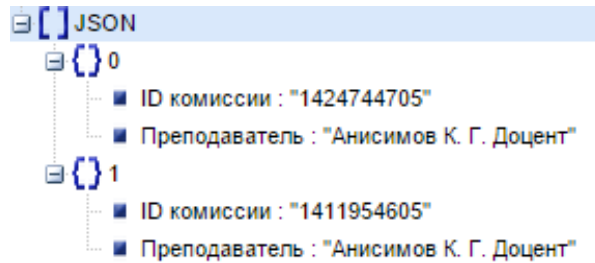


Рисунок 2.6 – Структура JSON передаваемых преподавателей комиссий

Структура файла для передачи информации о комиссиях представлена на рисунке 2.7.

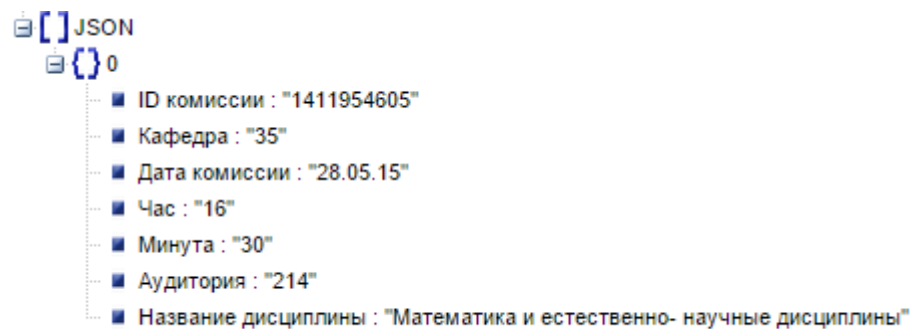


Рисунок 2.7 – Структура JSON передаваемой информации о комиссиях

2.2.5 Информационная модель и ее описание

Проблемная область – взаимосвязанная совокупность управляемых объектов предприятия (предметная область), субъектов управления, автоматизируемых функций управления и программно-технических средств их реализации.

Модель – некоторая система, имитирующая структуру или функционирование исследуемой проблемной области, отвечающей основному требованию – адекватности этой области.

С точки зрения объектов моделирования необходимо различать модели предметной области и модели базы данных. Эти модели взаимосвязаны, поскольку представляют собой образы одного и того же оригинала –

некоторого множества предметов реального мира, информацию о которых предполагается хранить и обрабатывать с помощью проектируемой БД.

Модель предметной области скорее ассоциируется с неформальным уровнем семантического моделирования, а модель базы данных – с формализованным уровнем системы (и в частности, СУБД). В идеале целью семантического моделирования является формирование систематического основания для хорошо формализованного процесса проектирования базы данных.

Требования, предъявляемые к базам данных, и, в частности, к способам описания данных:

- описания должны быть понятны пользователю, не проектировавшему базу;

- однажды принятые способы представления данных должны допускать присоединение новых элементов данных без изменения существующих схем данных и прикладных программ;

- СУБД должны позволять эффективно обрабатывать произвольные запросы к базе данных.

Эти требования отражают, с одной стороны, точку зрения пользователя, для которой характерны требования высокой степени общности и широты представления, позволяющих ему получить достаточно сведений без затраты значительных временных или интеллектуальных ресурсов. С другой стороны – точку зрения администратора, выполняющего проектирование и оптимизацию системы баз данных, что предполагает высокую степень детализации и формализации, обеспечивающих обоснованность технических решений, а также возможность автоматизации проектирования [4].

Начальной стадией проектирования системы баз данных является построение семантической модели предметной области, которая базируется на анализе свойств и природы объектов предметной области и информационных потребностей будущих пользователей разрабатываемой системы. Эту стадию принято называть концептуальным проектированием системы, а ее результат –

концептуальной моделью предметной области. Объектом моделирования здесь является предметная область будущей системы. Этой стадии соответствуют также термины «инфологическое проектирование» и «инфологическая модель» [5].

Такие модели обобщенно представляют информационные потребности пользователей создаваемой системы в части использования хранимых данных и по существу являются средством коммуникации как разработчиков, так и пользователей на разных стадиях жизненного цикла базы данных.

К инфологическим моделям относятся различные компоненты, по-разному и разными средствами отражающие предметную область. Помимо наиболее известного описания объектов и связей между ними (модель «сущность-связь») к инфологическому уровню описания предметной области можно отнести следующие компоненты:

- систему атрибутов и средств описания предметной области. Например, логические (алгоритмические) связи между показателями или лингвистические свойства языка (синонимии, синтаксис и т. д.), используемого для вербального представления объектов;

- ограничения целостности, определяющие допустимость значения отдельных полей и взаимосвязей как на уровне семантики содержимого БД, так и ее физической структуры (отдельных файлов данных и взаимосвязей между ними);

- описание информационных потребностей пользователей, например, в виде типовых запросов, отражающих процедурные особенности обращения к данным.

Моделирование предметной области на основе модели «Сущность-связь» также называется ER-моделирование. Оно базируется на использовании графических диаграмм, как простого (привычного), наглядного и в то же время информативного и многоаспектного способа отображения компонентов проекта.

Сущность, с помощью которой моделируется класс однотипных

объектов, определяется как «предмет, который может быть четко идентифицирован». Так же как каждый объект уникально характеризуется набором значений свойств, сущность должна определяться таким набором атрибутов, который позволял бы различать отдельные экземпляры сущности.

Каждый экземпляр сущности должен быть отличим от любого другого экземпляра той же сущности. Уникальным идентификатором сущности может являться атрибут, комбинация атрибутов, комбинация связей или комбинация связей и атрибутов, однозначно отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Сущность имеет имя, уникальное в пределах модели. При этом имя сущности – это имя типа, а не некоторого конкретного экземпляра.

Сущности подразделяются на сильные и слабые. Сущность является слабой, если ее существование зависит от другой сущности – сильной по отношению к ней.

Семантическую основу ER-модели составляют следующие предположения:

- часть реального мира (совокупность взаимосвязанных объектов), сведения о которых должны быть помещены в базу данных, может быть представлена как совокупность сущностей;

- каждая сущность обладает характеристическими свойствами (атрибутами), отличающими ее от других сущностей и позволяющими ее идентифицировать;

- сущности можно классифицировать по типам сущностей: каждый экземпляр сущности (представляющий некоторый объект) может быть отнесен к классу – типу сущностей, каждый экземпляр которого обладает общими для них и отличающими их от сущностей других классов свойствами;

- систематизация представления, основанная на классах, в общем случае предполагает иерархическую зависимость типов сущность типа А является подтипом сущности В, если каждый экземпляр типа А является экземпляром сущности типа В;

– взаимосвязи объектов могут быть представлены как связи – сущности, которые служат для фиксирования (представления) взаимозависимости двух или нескольких сущностей.

Одна из основных целей семантического моделирования состоит в том, чтобы результаты анализа предметной области были отражены в достаточно простом, наглядном, но в то же время формализованном и достаточно информативном виде. В этом смысле ER-диаграмма является очень удачным решением. В ней сочетаются функциональный и информационный подходы, что позволяет представлять как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных. При этом графическая форма позволяет отобразить в компактном виде типологию и свойства сущностей и связей за счет наглядных условных обозначений, а формализмы, положенные в основу ER-диаграмм, позволяют использовать на следующем шаге проектирования логической структуры базы данных строгий аппарат нормализации.

Сущности. Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности. В качестве имени обычно используются существительные (или обороты существительного) в единственном числе.

Связь – графически изображаемая ассоциация, устанавливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри.

Как и в реляционных схемах баз данных, в ER-диаграммах вводится понятие нормальных форм, причем их смысл очень близок смыслу реляционных нормальных форм. Приведем краткие и неформальные определения трех первых нормальных форм.

В первой нормальной форме ER-диаграммы устраняются повторяющиеся атрибуты или группы атрибутов, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

Во второй нормальной форме устраняются атрибуты, зависящие только

от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности [12].

Задачей следующей стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Другими словами, модель предметной области разрабатываемой системы должна быть представлена в терминах модели данных концептуального уровня выбранной конкретной СУБД. Эту стадию называют логическим (или даталогическим) проектированием базы данных, а ее результатом является концептуальная схема базы данных, включающая определение всех информационных элементов (единиц) и связей, в том числе задание типов, характеристик и имен.

Существует много вариантов отображения инфологической модели предметной области в даталогическую модель базы. Здесь следует учитывать влияние двух следующих значимых факторов, связанных с практикой разработки базы данных.

Во-первых, связи предметной области могут отображаться двумя путями: как декларативным – в логической схеме, так и процедурным – отработкой связей через программные модули, обрабатывающие (связывающие) соответствующие хранимые данные.

Во-вторых, существенным фактором может оказаться характер обработки информации. Например, частые обращения к совместно обрабатываемым данным, очевидно, предполагают их совместное хранение, а данные (особенно большого объема), к которым обращаются редко, целесообразно хранить отдельно от часто используемых.

Стадия физического проектирования базы данных в общем случае включает:

- выбор способа организации базы данных;

- разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;
- описание отображения концептуальной схемы во внутреннюю.

Важно заметить, что в отличие от ранних СУБД, многие современные системы не предоставляют разработчику какого-либо выбора на этой стадии.

Способ хранения базы данных определяется механизмами СУБД автоматически «по умолчанию» на основе спецификаций концептуальной схемы базы данных, и внутренняя схема в явном виде в таких системах не используется. Следует также отметить, что внешние схемы базы данных обычно конструируются на стадии разработки приложений [11].

На основе проведенного анализа, изученной теории создания информационных систем и поставленных целей и задач можно перейти непосредственно к проектированию информационной системы.

Для составления логической модели данных с помощью обратного проектирования будет использоваться CASE-средство ERwin.

ERwin – средство концептуального моделирования БД, позволяет проектировать, документировать и сопровождать базы данных. Создав наглядную модель базы данных, можно оптимизировать структуру БД и добиться её полного соответствия требованиям и задачам организации. Визуальное моделирование повышает качество создаваемой базы данных, продуктивность и скорость её разработки. Благодаря удобной в использовании графической среде, которая упрощает проектирование баз данных и автоматизирует многие трудоемкие задачи.

Разработчики с помощью ERwin Data Modeler могут сначала, используя визуальные средства, описать схему БД, а затем автоматически сгенерировать файлы данных для выбранной реляционной СУБД (прямое проектирование). Пользователь описывает структуру данных визуально. Он задает служащие прообразами реляционных таблиц сущности с их атрибутами и при помощи мыши выстраивает между ними связи, которые являются прототипами реляционных отношений.

ERwin поддерживает следующие типы баз данных: Oracle, InterBase, Ingres, Microsoft SQL Server, Clipper, ODBC, DB2, dBASE, Paradox, FoxPro, Rdb, HiRDB, Red Brick Warehouse, Informix, SAS, SQL Anywhere, Microsoft Access, SQL Base, Teradata, Sybase [12, 26].

Когда логическая модель недоступна, возникает необходимость воссоздания модели из существующей базы данных. В ERwin этот процесс называется обратным проектированием. Обратное проектирование может производиться несколькими способами. Разработчик модели может исследовать структуры данных в базе данных и воссоздать таблицы в визуальной среде моделирования.

Также, есть возможность импортировать язык описания данных (DDL – data definitions language) в инструмент, который поддерживает проведение обратного проектирования (например, ERwin). Развитые средства, такие как ERwin, включают функции, обеспечивающие связь через ODBC с существующей базой данных, для создания модели путем прямого чтения структур данных.

Обратное проектирование (Reverse engineering), то есть восстановление информационной модели по существующей базе данных, применяется после решения создать новую СУБД или расширить (изменить) существующую структуру [30].

На уровне физической модели сущности соответствует таблица в реальной СУБД, атрибуту – колонка таблицы, связи – внешний ключ (если для связи задавалось имя роли, то оно соответствует имени колонки внешнего ключа в дочерней таблице), первичным и альтернативным ключам – уникальные индексы [29].

На рисунке 2.8 представлена физическая модель БД модуля расписания.

Данная модель содержит сущности непосредственно для формирования расписания занятий. Основная сущность schedule – само расписание. Остальные сущности: lessons – занятия, auditorium – аудитории, t_students – список студентов, t_plans – учебные планы, students_groups – группы студентов,

specialityeducations – специальности, group_for_lesson – группы для расписания, gfl_teachers, физические лица – список преподавателей – вспомогательные и служат для формирования данных сущности schedule, которую использует портал для вывода расписания.

В результате проведения обратного проектирования были выделены сущности предметной области, определены взаимосвязи между ними и составлена логическая модель базы данных.

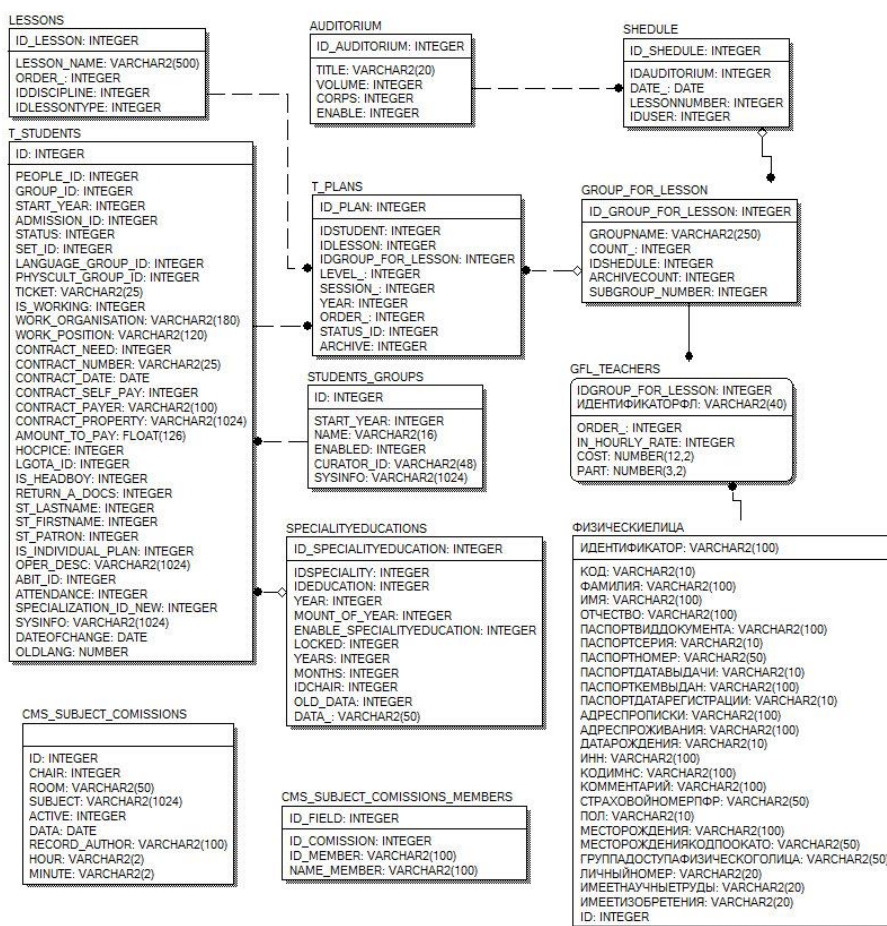


Рисунок 2.8 – Физическая модель базы данных модуля расписания

На рисунке 2.9 представлена логическая модель базы данных.

Данная диаграмма представляет собой логическую модель БД, т.е. модель без привязки к конкретной СУБД.

В качестве способа хранения данных в мобильном приложении была выбрана сериализация.

Сериализация – процесс перевода структур данных в последовательность битов. Обратной к операции сериализации является операция десериализации (структуризации) – восстановление начального состояния структуры данных из битовой последовательности. Сериализация используется для передачи объектов по сети и для сохранения их в файлы [40].

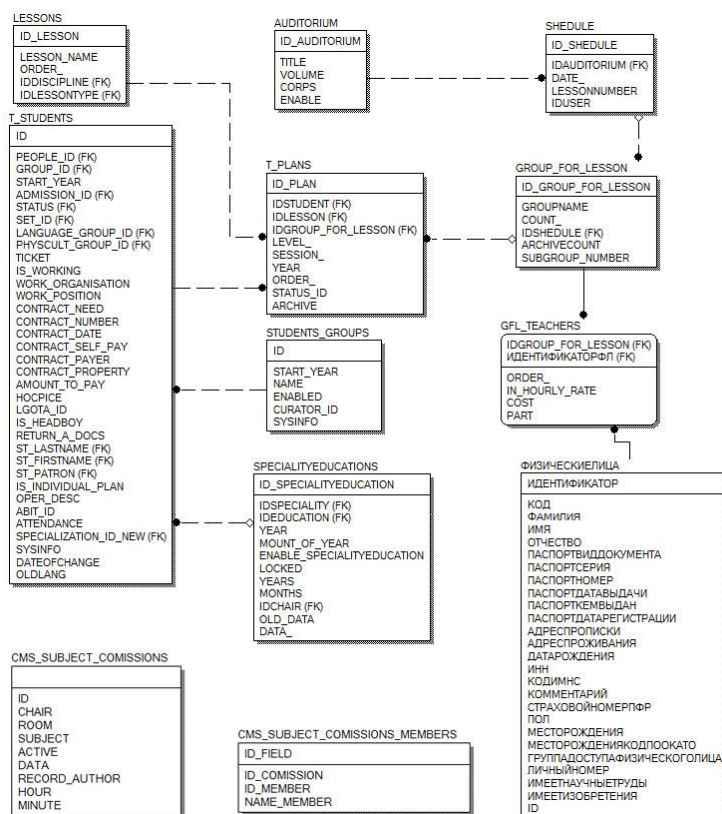


Рисунок 2.9 – Логическая модель базы данных модуля расписания

Использование данного способа хранения данных обусловлено следующими факторами:

- небольшое количество данных;
- отсутствие сложных выборок;
- скорость работы;
- простота реализации.

Данные будут сохраняться в памяти телефона файлом с расширением .dat.

2.3 Разработка программного обеспечения

Проектируемое мобильное приложение «РИ АлтГУ – Расписание», позволяет:

- получать расписания занятий, консультаций и комиссий;
- корректно отображать информация на любых экранах мобильных устройств;
- обновлять списки групп, преподавателей и аудиторий;
- сохранять пользовательские настройки;
- автоматически загружать в фоне выбранное расписание;
- использовать виджет для отображения расписания на главном экране и экране блокировки смартфона;
- оповещать об изменившемся расписании;
- сэкономить мобильный трафик.

Разработка мобильного приложения произведена с помощью языка Java, разработка программного модуля серверной части, занимающегося выборкой и передачей информации на мобильное устройство было произведено с помощью языка PHP на основе Zend-Framework.

Java – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной

машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание [31].

PHP – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов [13].

Популярность в области построения веб-сайтов, определяется наличием большого набора встроенных средств для разработки веб-приложений [28].

2.3.1 Описание серверных программных модулей

Для обеспечения функционирования приложения на сервере необходимы следующие программные модули:

- модуль консультаций (consultation) обеспечивает формирование списка консультаций;
- модуль предметных комиссий (subjectcomissions) обеспечивает формирование расписания предметных комиссий по различным параметрам;
- модуль расписания (timetable) обеспечивает формирование расписания по различным параметрам;

Данные модули будут созданы с помощью языка PHP, на фреймворке Zend Framework, который будет обеспечивать работу данных модулей.

Определенная соглашением модульная структура директорий позволяет разделять различные приложения MVC в автономные единицы и повторно использовать их с различными фронт-контроллерами.

Как было сказано ранее, Zend Framework для построения приложений использует схему «Модель-представление-поведение». Исходя из этого, была сформирована структура каталогов информационно-образовательного портала, которая представлена на рисунке 2.10.

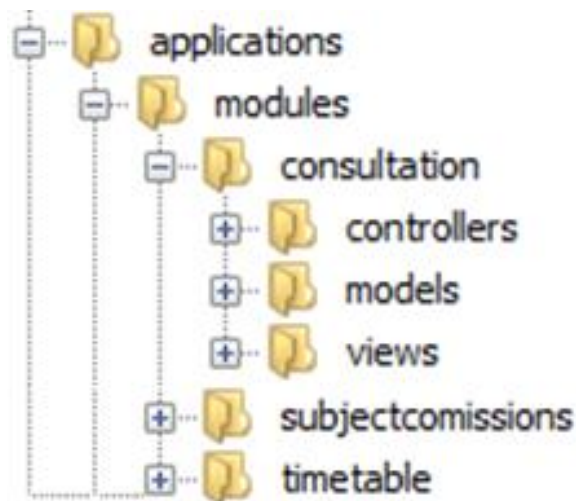


Рисунок 2.10 – Модульная структура директорий проекта

2.3.2 Описание клиентских программных модулей

Для полноценного функционирования приложения необходимо наличие следующих модулей:

- основной модуль программы;
- модуль получения расписания занятий (день, неделя, месяц);
- модуль получения расписания консультаций;
- модуль получения расписания комиссий;
- модуль настроек;
- модуль вывода расписания на экран;
- модуль автоматической загрузки расписания;
- модуль уведомления об изменении расписания на текущую дату;
- модуль сбора информации для виджета;
- модуль отображения информации на виджете;
- модуль запуска и обновления виджета;
- модуль отслеживания состояния мобильного телефона.

Данные модули представляют собой отдельный класс в Java. Их можно увидеть на рисунке 2.11.

Функционал основного модуля служит «каркасом» для приложения и

отвечает за внешний интерфейс и жизненный цикл приложения.

Функционал модуля расписания занятий отвечает за получение расписаний занятий за день, неделю или месяц, в зависимости от выбранного параметра. Данный модуль занимается получением и обработкой JSON-файла расписания занятий, в зависимости от выбранных пользователем параметров.

Модуль получения расписаний консультаций занимается получением и выводом расписания консультаций на экран.

Модуль получения расписаний комиссий занимается обработкой передаваемых параметров, получением расписания комиссий и обработкой полученной информации для предоставления пользователю.

Модуль настроек используется для создания и вывода пользовательских настроек.

Модуль сбора информации для виджета получает информацию из памяти телефона, выбирает необходимую информацию, обрабатывает её и передает в модуль отображения на виджет, который отображает её на экране смартфона.

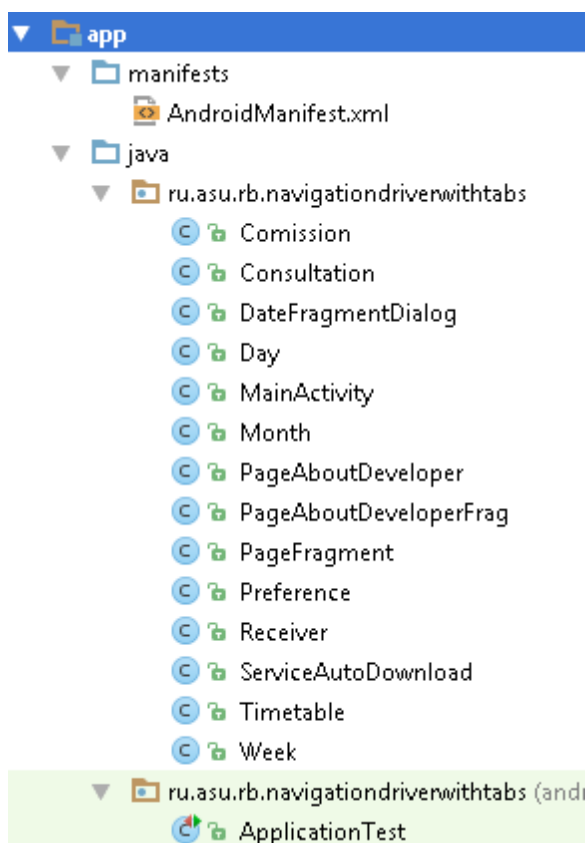


Рисунок 2.11 – Модули Android приложения

Модуль запуска и обновления виджета обновляет его через заданные промежутки времени.

Модуль отслеживания состояния мобильного телефона отслеживает изменения состояния сети, включения/выключения смартфона и вызывает модуль автоматической загрузки расписания.

2.3.3 Компоненты пользовательского интерфейса

Интерфейс пользователя – коммуникационный канал, реализующий взаимодействие пользователя с программным обеспечением с помощью элементов управления.

Графический интерфейс пользователя – среда организации взаимодействия пользователя с вычислительной системой.

Стандартный графический интерфейс пользователя должен отвечать ряду требований:

- поддерживать информационную технологию работы пользователя с программным продуктом – содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки, расположенные в естественной последовательности использования;
- ориентироваться на конечного пользователя, который общается с программой на внешнем уровне взаимодействия;
- удовлетворять правилу «шести» – в одну линейку меню включать не более шести понятий, каждое из которых содержит не более шести операций;
- графические объекты сохраняют свое стандартизованное назначение и, по возможности, местоположение на экране [6, 7].

Разработка пользовательского интерфейса – одна из самых сложных и ответственных задач проектирования. Это объясняется тем, что пользователя интересует в первую очередь удобство, эргономичность и наглядность [8].

Графический интерфейс Android приложений строится на иерархии из

элементов двух типов: View и ViewGroup. Элементами типа View являются любые дочерние элементы пользовательского интерфейса (UI widgets), такие как кнопка, поле для ввода и т.д. ViewGroup – это невидимый контейнер графических элементов, определяющий их размещение на экране. Это может быть вертикальный, горизонтальный список или таблицы элементов. Один ViewGroup может содержать множество view и другие ViewGroup. Оперируя различными графическими элементами и контейнерами элементов можно создавать интерфейсы. Иерархия View и ViewGroup представлена на рисунке 2.12.

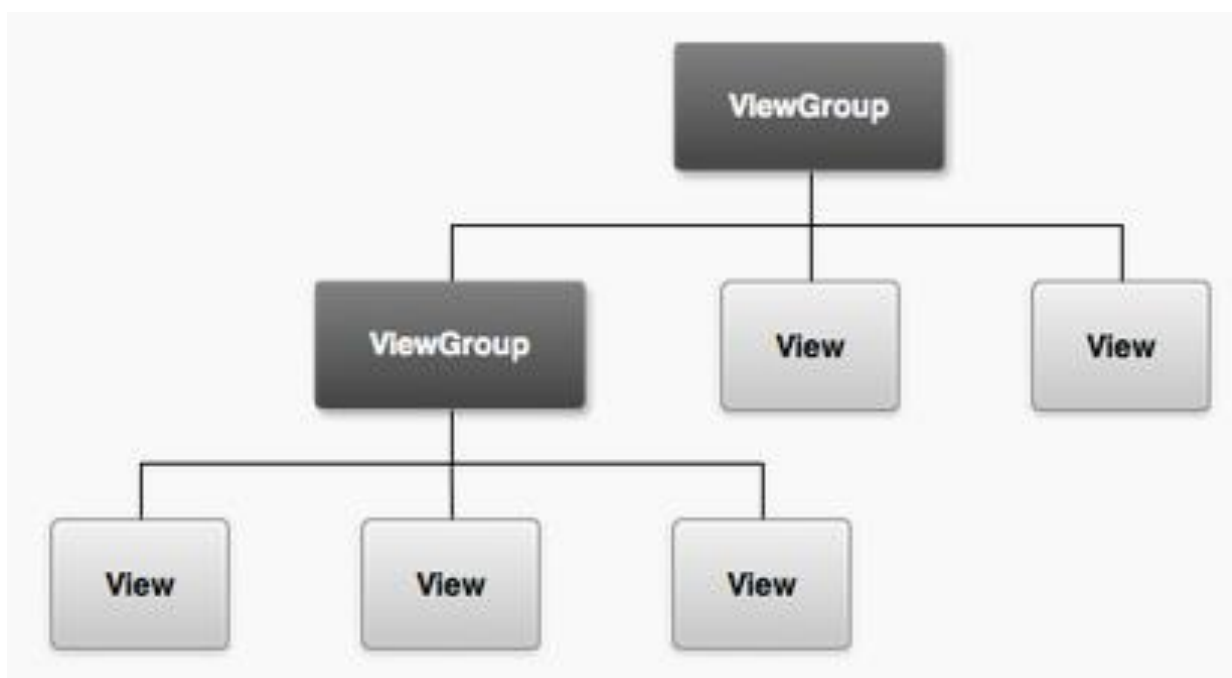


Рисунок 2.12 – Иерархия View и ViewGroup

Доступны следующие виды ViewGroup:

- `FrameLayout` – самая простая разметка, прикрепляет каждое новое дочернее представление к левому верхнему углу экрана, накладывая новый элемент на предыдущий, заслоняя его;
- `LinearLayout` – помещает дочерние представления в горизонтальный или вертикальный ряд. Вертикальная разметка представляет собой колонку, а горизонтальная – строку с элементами. Данная разметка позволяет задавать не только размеры, но и «относительный вес» дочерних элементов, благодаря

чему можно гибко контролировать их размещение на экране;

- `RelativeLayout` – наиболее гибкий среди стандартных видов разметки. Позволяет указывать позиции дочерних Представлений относительно границ свободного пространства и других представлений;

- `TableLayout` – позволяет размещать дочерние представления внутри ячеек «сетки», состоящей из строк и столбцов. Размеры ячеек могут оставаться постоянными или автоматически растягиваться при необходимости;

- `GridLayout` – отображает прокручиваемую сетку со строками и столбцами.

Для создания пользовательского интерфейса в Android приложениях используется расширяемый язык разметки XML. Для создания окна или группы элементов необходимо создать отдельный файл с расширением `.xml`.

Каждый созданный `.xml` файл должен иметь хотя бы одну `ViewGroup`, которая будет описывать то, каким образом будут располагаться элементы на форме.

На рисунке 2.13 представлена главная форма приложения. Она представляет собой окно получения расписания занятий.

На данной форме можно выбрать тип получаемого расписания: группа, аудитория, преподаватель. Кроме этого на этой форме можно выбрать дату, на которую необходимо получить расписание.

В приложении имеется меню выбора необходимого расписания. Оно реализовано списком, который находится слева от основной формы. Меню можно открыть, потянув от левого края вправо, либо нажав на кнопку меню в верхней левой части приложения. Пункты меню показаны на рисунке 2.14.

Форма получения консультаций представлена в виде окна с кнопкой, для получения списка. Данное окно изображено на рисунке 2.15.

Форма получения списка комиссий представляет собой окно с возможностью выбора необходимой кафедры. Данное окно изображено на рисунке 2.16.



Рисунок 2.13 – Стартовая форма приложения

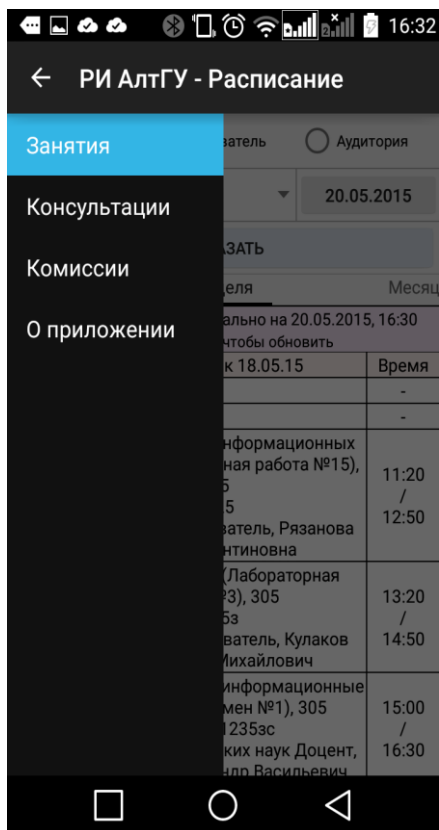


Рисунок 2.14 – Меню приложения

Среда 20.05.15	Аудитория	Время
Быстрова Валентина Михайловна Старший преподаватель	204 (1 корпус)	16:40 / 18:10 (6 пара)
Четверг 21.05.15	Аудитория	Время
Бирюкова Тамара Дмитриевна Доцент	211 (1 корпус)	11:20 / 12:50 (3 пара)
Быстрова Валентина Михайловна Старший преподаватель	204 (1 корпус)	16:40 / 18:10 (6 пара)
Ворожбит Елена Геннадьевна Доцент, кандидат экономических наук	213 (1 корпус)	15:00 / 16:30 (5 пара)
Матюшкина Елена Александровна Преподаватель	204Б (2 корпус)	09:40 / 11:10 (2 пара)
Шейкина Ольга Владимировна Старший преподаватель	211 (1 корпус)	13:20 / 14:50 (4 пара)

Рисунок 2.15 – Форма получения списка консультаций

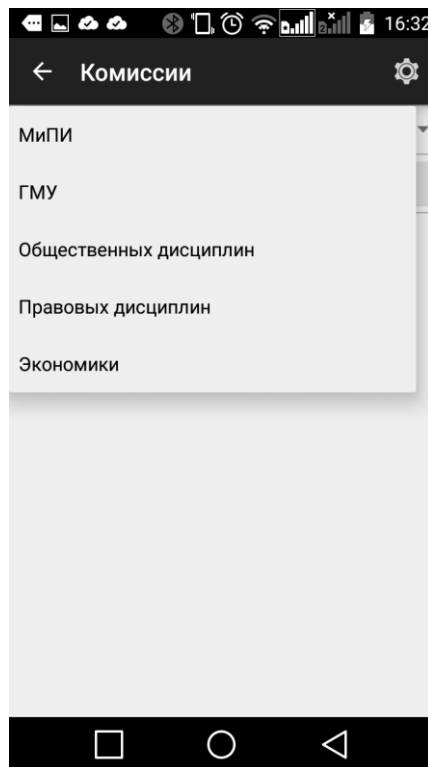


Рисунок 2.16 – Форма получения списка комиссий

Настройки приложения можно запустить, нажав на кнопку сверху справа формы.

Пункты настроек:

- автоматическая загрузка;
- автоматическая загрузка только по Wi-Fi;
- тип расписания (группа, аудитория, преподаватель);
- элемент (название группы, номер аудитории, имя преподавателя);
- настройка виджета;
 - настройка виджета – размер шрифта;
 - настройка виджета – цвет шрифта;
- настройка списков;
 - загрузить списки;
- версия.

«Автоматическая загрузка» позволяет пользователю автоматизировать загрузку расписания. При выборе данного пункта в приложении запускается сервис, который следит за актуальностью загруженного в память расписания и, при обнаружении изменившегося расписания – сразу загружает его в память телефона. Тем самым у пользователя всегда будет актуальное расписание занятий. Данный сервис запускается и перезапускается автоматически, тем самым пользователю не нужно думать о том, запущен ли сервис или нет.

При активированном пункте меню «Автоматическая загрузка расписания через Wi-Fi» расписание будет загружаться в телефон при наличии доступной по близости Wi-Fi точки. Активация данного пункта позволит пользователям сократить мобильный трафик, который расходуется на автоматическую загрузку расписания.

В пункте «Тип расписания» можно выбрать тип загружаемого расписания – группа, аудитория, преподаватель.

В пункте «Элемент» можно выбрать название группы, номер аудитории или имя преподавателя для которой(ого) нужно получать расписание.

Пункт «Настройки виджета» отвечает за его настройку, а именно настройку: цвета и размера шрифта.

Пункт «Настройка списков» отвечает за загрузку и обновление списков

аудиторий, групп и преподавателей. Загрузку необходимо делать раз в год либо при отсутствии группы, аудитории или преподавателя в списке.

Окно настроек представлено на рисунке 2.17.

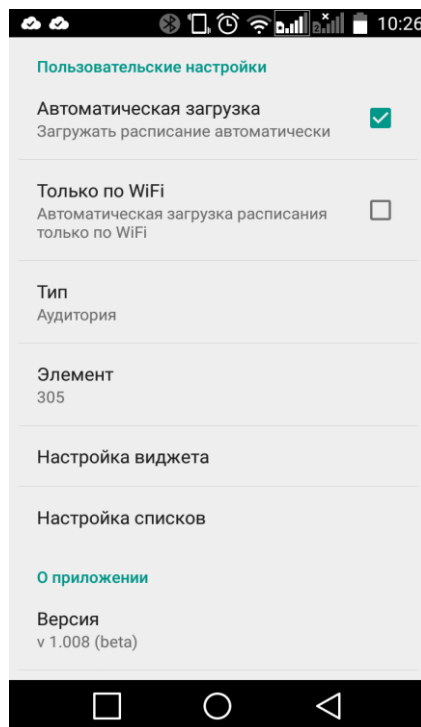


Рисунок 2.17 – Окно настроек приложения

2.4 Компьютерно-сетевое обеспечение

Мобильное приложение имеет архитектуру клиент-сервер.

Серверная часть приложения использует существующее сетевое оборудование РИ (филиала) АлтГУ без каких-либо изменений. Структура сети также не подвергается изменениям, т.к. ее характеристики полностью удовлетворяют предъявленным требованиям. Используется следующее оборудование:

1. Сервер WWW. Выполняет функции базового WEB-сервера со следующим набором сервисов: HTTP, FTP, DNS, MAIL, PROXY, APACHE, MySQL, DIALUP. Технические характеристики: 2xP4Xeon, 3,0GHz, 4x1Gb, RAID1 SATA 2x80Gb, 2xGigabit Ethernet. Установленное ПО – Gentoo Linux 2008 Server Edition.

2. Сервер ORACLE. Выполняет функции сервера БД для хранения информации для портала и информации необходимой для учебного процесса. Технические характеристики: 2xP4Xeon, 4,0GHz,, 12Gb, 324Gb, Gigabit Ethernet. Установленное ПО – Gentoo Linux 2008 Server Edition.

2.5 Обеспечение информационной безопасности

Информационная безопасность (ИБ) – состояние защищённости информационной среды общества, обеспечивающее её формирование, использование и развитие в интересах отдельных граждан, организаций и государства [18].

Персональные данные – любая информация, относящаяся к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

Информационная система персональных данных – информационная система, представляющая собой совокупность персональных данных, содержащихся в базе данных, а также информационных технологий и технических средств, позволяющих осуществлять обработку таких персональных данных с использованием средств автоматизации или без использования таких средств [24].

Классификация информационной системы персональных данных.

Определение категории обрабатываемых персональных данных:

– первая категория – персональные данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных и философских убеждений, состояния здоровья, интимной жизни;

– вторая категория – персональные данные, позволяющие идентифицировать субъекта персональных данных и получить о нём

дополнительную информацию, за исключением персональных данных, относящихся к категории 1;

- третья категория – персональные данные, позволяющие идентифицировать субъекта персональных данных;

- четвертая категория – обезличенные и (или) общедоступные персональные данные.

По результатам анализа исходных данных ИС присваивается один из следующих классов:

- класс 1 (К1) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к значительным негативным последствиям для субъектов персональных данных;

- класс 2 (К2) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к негативным последствиям для субъектов персональных данных;

- класс 3 (К3) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к незначительным негативным последствиям для субъектов персональных данных;

- класс 4 (К4) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, не приводит к негативным последствиям для субъектов персональных данных [22, 21].

Для защиты от постороннего вторжения предусматриваются определенные меры безопасности. В ИС это осуществляется программными средствами, которые выполняют следующие функции:

- идентификация субъектов и объектов;
- разграничение доступа к ресурсам и информации;
- контроль и регистрация действий.

Процедура идентификации и подтверждения подлинности предполагает проверку является ли субъект, осуществляющий доступ, или объект, к которому осуществляется доступ, тем за кого себя выдает. Здесь используются следующие методы:

- простые, сложные или одноразовые пароли;
- средства анализа индивидуальных характеристик субъекта (геометрических данных);
- ключи, жетоны, магнитные карты;
- обмен вопросами и ответами с администратором системы;
- специальные идентификаторы и контрольные суммы для программ и данных.

После процедуры идентификации пользователь получает доступ к системе, где защита от несанкционированного доступа реализуется на трех уровнях:

- на уровне аппаратуры;
- на уровне программного обеспечения;
- на уровне данных.

Защита информации на 1 и 2 уровнях предусматривает управление доступом к различным вычислительным ресурсам (отдельным устройствам, операционной системы, служебным или личным программам пользователя).

Защита информации на уровне данных направлена на защиту информации в процессе обращения к ней, в процессе работы с файловой структурой ЭВМ, в процессе передачи информации по каналам связи.

В общем, комплекс программно-технических средств и организованных решений по защите информации от несанкционированного доступа характеризуется следующими действиями:

- логическое управление доступом;
- регистрацией, контролем и учетом работы;
- применением криптографических средств;
- обеспечением целостности информации.

Выделяют следующие формы контроля и управления доступом:

- предотвращение доступа к жесткому диску, каталогам и файлам;
- установка привилегий к группам файлов;
- защита от модификаций и изменений;
- защита от уничтожения;
- предотвращение копирования.

Под средствами защиты от копирования понимается такие средства, которые обеспечивают выполнение программой своих функций только при опознании некоторого уникального не копированного элемента. Таким элементом может быть ключевой сменный носитель, часть оборудования или специальное устройство, подключаемое к компьютеру.

Защита от копирования реализуется выполнением ряда функций:

- идентификация среды, из которой запускается система;
- проверка подлинности среды, из которой произошел запуск;
- немедленная реакция на запуск из несанкционированной среды;
- обязательная регистрация санкционированного копирования;
- противодействие изучению алгоритмов работы системы [27].

Область информационной безопасности является компетенцией отдела программного и технического обеспечения. Данным отделом регулярно производятся плановые профилактические работы всех программно-аппаратных систем Института.

В Рубцовском институте (филиале) АлтГУ доступ к информационным системам управления вузом осуществляется на основании приказа директора «Об оптимизации доступа к БД, в целях защиты информации», в котором указаны лица, имеющие полный доступ на внесение и изменение информации и лица, имеющие доступ для чтения информации, в случае ее использования для принятия решения.

Полный доступ на изменение информации имеют только сотрудники отдела по работе со студентами, у методистов кафедр, заведующих кафедрами, директора и его заместителей имеется доступ на чтение информации.

Поскольку авторизация и доступ к конфиденциальной информации по разграниченным правам в мобильном приложении не реализовано, это облегчает защиту информации от несанкционированного доступа, так как любой пользователь имеет доступ только к общедоступной информации.

Защита информации БД на программном уровне информационно-образовательного портала осуществляется стандартными методами, такими как: проверка введенной пользователем информации и проверка корректности принимаемых от клиента данных.

3 Оценка эффективности от внедрения мобильного приложения

3.1 Общие положения

Эффективность процессов характеризуется системой показателей, отражающих соотношение их затрат и результатов. Эффективность процесса тем выше, чем выше результаты и ниже приложенные усилия.

Эффективность ИС – это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством. Эта характеристика отражает:

- действенность системы, т.е. степень соответствия ИС своему назначению (прагматическая эффективность);
- техническое совершенство ИС (техническая эффективность);
- простоту, технологичность разработки и создание системы (технологическая эффективность);
- удобство использования и обслуживания системы (эксплуатационная эффективность);
- улучшение и облегчение условий труда, изменение его содержания, развитие творческих функций, способностей и потребностей людей, преодоление существенных различий в труде и др. (социальная эффективность);
- экономическую целесообразность внедрения ИС, т.е. целесообразность произведенных на создание и функционирование системы затрат (экономическая эффективность).

Понятие эффективности связано с получением некоторого полезного результата – эффекта использования.

В соответствии с ГОСТ Р ИСО 9000-2001, эффективность функционирования ИС определяется соотношением результата (эффекта) и

затраченными ресурсами. Приведенной оценкой затрат ресурсов выступает их стоимость. Затраты на функционирование ИС состоят из:

- стоимости приобретения программной платформы;
- стоимости доработки;
- стоимости внедрения;
- стоимости системного и вспомогательного программного обеспечения;
- стоимости аппаратного и сетевого обеспечения ИС;
- количества циклов (лет) эксплуатации;
- стоимости эксплуатации.

Основные задачи, стоящие при создании ИС – минимизация стоимости и обеспечение требуемого качества ИС.

Качество – это совокупность свойств системы, обуславливающих возможность ее использования для удовлетворения определенных потребностей пользователей в соответствии с ее назначением.

Основными показателями качества ИС являются:

- надежность;
- достоверность;
- безопасность.

Надежность – свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения.

Надежность ИС является средством обеспечения актуальной и достоверной информации на выходе системы.

Достоверность функционирования – свойство системы, обуславливающее безошибочность производимых ею преобразований информации. Достоверность функционирования ИС полностью определяется и измеряется достоверностью ее резульатной информации.

Безопасность – свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, т.е. защиту информации от несанкционированного доступа.

В любой сфере человеческой деятельности оценка эффективности внедрения любой новой техники, технологий и ИС осуществляется с помощью множества показателей. К ним относятся показатели прагматической, технической, эксплуатационной, социальной и экономической эффективности [1].

3.2 Показатели эффективности

В качестве показателей прагматической эффективности для мобильного приложения могут выступать:

- показатели безопасности информационной системы (защита сервера от различных атак, уменьшение уязвимости портала за счет контроля вводимой пользователем информации);

- показатели оперативности (использование созданного приложения позволяет снизить затраты ресурсов на доступ к информации, значительно снижены временные затраты на поиск необходимой информации).

Показатели технической эффективности должны оценивать техническое совершенство ИС, научно-технический уровень организации и функционирования этой системы.

Разработанное приложение:

- работает по архитектуре клиент-сервер;
- серверная часть базируется на отработанном и проверенном годами реляционном принципе построения баз данных.

Показатели эксплуатационной эффективности:

- показатели надежности – обрабатываемые данные зависят от надежности компьютера и спроектированной системы;

- функциональные возможности – появилась возможность использования портала с помощью мобильных устройств;

- технология обслуживания – система не требует вмешательства программиста во время работы.

Среди показателей социальной эффективности можно выделить предоставление постоянного и удобного доступа к необходимому расписанию, а также повышение престижности ВУЗа, конкурентоспособности.

Обобщающими показателями эффективности любой ИС являются показатели экономической эффективности. Часто прибыль определяется путем экспертной оценки или по аналогии с другими подобными системами.

Для оценки эффективности могут использоваться две группы показателей: интегральные традиционные показатели и частные показатели.

Обычно в качестве экономических показателей используются:

- годовой экономический эффект;
- коэффициент экономической эффективности капитальных вложений;
- срок окупаемости капитальных вложений;
- трудоемкость обработки информации;
- эксплуатационная стоимость затрат;
- расчет текущих затрат пользователя;
- экономия текущих затрат при автоматизации;
- годовая экономия затрат на материалы.

Экономический эффект – это результат внедрения какого-либо мероприятия, выраженный в стоимостной форме в виде экономии от его осуществления. Основными источниками экономии являются:

- улучшение показателей их основной деятельности, происходящее в результате использования программного изделия;
- повышение технического уровня, качества и объёмов вычислительных работ;
- увеличение объёмов и сокращение сроков переработки информации;
- повышение коэффициента использования вычислительных ресурсов, средств подготовки и передачи информации;
- уменьшение численности персонала, занятого обработкой исходных данных, переработкой и получением необходимой информации;
- снижение затрат на эксплуатационные материалы.

Предварительный экономический эффект рассчитывается до выполнения разработки на основе данных технических предложений и прогноза использования. Предварительный эффект является элементом технико-экономического обоснования (ТЭО) разработки проекта.

Потенциальный экономический эффект рассчитывается по окончании разработки на основе достигнутых технико-экономических характеристик и прогнозных данных о максимальных объёмах использования программного изделия.

Коэффициент экономической эффективности капитальных вложений показывает величину годового прироста прибыли, образующуюся в результате производства или эксплуатации программного изделия на один рубль капитальных единовременных вложений.

Срок окупаемости (величина, обратная коэффициенту эффективности) – показатель эффективности использования капиталовложений, представляет собой период времени, в течение которого произведённые затраты на программные изделия окупаются полученным эффектом.

Для оценки экономической эффективности внедрения ИС можно использовать систему частных показателей.

Частные показатели необходимы для оценки частного экономического эффекта, получаемого по отдельным источникам экономии.

Например, частные показатели в сфере оказания образовательных услуг:

- сокращение доли неквалифицированного и ручного труда;
- сокращение материальных затрат;
- сокращение инвестиций в рекламный бюджет;
- снижение уровня «бумажного» документооборота;
- сокращение времени на принятие решения звене управления [1].

3.3 Расчет экономической эффективности

Экономическая эффективность учитывает затраты и результаты

реализации проекта, выходящие за пределы прямых финансовых интересов его участников.

Экономическая эффективность позволяет судить о необходимости внедрения программного продукта. В основе исчисления экономической эффективности лежит сопоставление существующего реально метода обработки данных (базовый вариант) и внедряемого метода обработки (проектный вариант). При этом обязательно проводится анализ затрат, необходимых для выполнения всех операций, сопутствующих внедрению нового метода обработки данных. К таким затратам относятся затраты на разработку, реализацию, внедрение и эксплуатацию программного продукта.

Выбор базы для сравнения зависит от цели расчета эффективности, от того, что требуется определить: ожидаемую, а также фактическую эффективность в конкретных условиях применения вычислительной техники или наиболее выгодный способ обработки данных. В первом случае за базу для сравнения следует принять способ выполнения работ, существующий в конкретных условиях до применения данной вычислительной техники, во втором случае – предлагаемый лучший способ обработки данных.

Особенностью расчетов сравнительной эффективности автоматизированной обработки данных является то, что в отдельных случаях базовый вариант может отсутствовать. Весь эффект определяется сопоставлением экономии от использования информации с затратами на ее получение.

В данном дипломном проекте в качестве базового варианта выступает действующая мобильная версия информационно-образовательного портала. В качестве предлагаемого варианта используется созданное в дипломном проекте мобильное приложение для получения расписания.

Сопоставление базового и проектного вариантов производится на основании расчета экономических показателей. Основными из них являются:

- показатель трудоемкости обработки информации;
- показатель эксплуатационных стоимостных затрат;

- экономический эффект;
- текущие затраты пользователя;
- экономия текущих затрат при автоматизации;
- относительная годовая экономия затрат на материалы.

3.3.1 Расчет трудоемкости обработки информации

Пусть T_0 – трудозатраты по базовому варианту (чел/час), T_j – трудозатраты по предлагаемому варианту (чел/час).

Базовый вариант:

Действующая мобильная версия информационно-образовательного портала приспособлена для работы с мобильными устройствами, однако имеет ряд недостатков. Большое количество функциональных возможностей довольно сильно затрудняют процесс представления информации.

В общей сложности для загрузки страницы через мобильный интернет приходится в среднем около 2 секунд. Для каждого пользователя регистрируется около 40 обращений к portalу в год.

$$T_0 = 40 * (2/3600) = 0,0222 \text{ чел/час.}$$

Проектный вариант:

Мобильное приложение, в отличие от мобильной версии сайта, в качестве данных получает информацию, содержащую только расписание, а не полностью исходный код страницы со всеми стилями и картинками. Данное решение позволит снизить нагрузку на сервер, мобильный трафик и время загрузки расписания. Время загрузки расписания составляет ~ 1 секунду, при этом, в отличие от мобильной версии сайта, объем передаваемых мобильных данных уменьшился на 25%. При наличии расписания в памяти телефона прорисовка происходит за 0,3 секунды.

$$T_{j \text{ интернет}} = 40 * (1/3600) = 0,0111 \text{ чел/час.}$$

$$T_{j \text{ память}} = 40 * (0,3/3600) = 0,0033 \text{ чел/час.}$$

Показатель снижения трудовых затрат (ΔT) рассчитывается по формуле:

$$\Delta T = T_0 - T_j \quad (3.1)$$

$$\Delta T_{\text{интернет}} = 0,0222 - 0,0111 = 0,0111 \text{ чел/час.}$$

$$\Delta T_{\text{память}} = 0,0222 - 0,0033 = 0,0189 \text{ чел/час}$$

Коэффициент снижения трудовых затрат (K_m) вычисляется по формуле:

$$K_m = \Delta T / T_0 \quad (3.2)$$

$$K_m_{\text{интернет}} = 0,0111 / 0,0222 = 0,5.$$

$$K_m_{\text{память}} = 0,0189 / 0,0222 = 0,85.$$

На 50% и 85% процентов снижаются временные затраты предлагаемого варианта, в зависимости от способа получения расписания, по сравнению с базовым.

3.3.2 Расчет трудоемкости разработки программного обеспечения

Расчет затрат времени на разработку программного обеспечения охватывает работы, выполняемые специалистами на следующих стадиях, каждая из которых имеет следующую трудоемкость:

- техническое задание (3 дня);
- эскизный проект (7 дней);
- технический проект (15 дней);
- рабочий проект (35 дней);
- внедрение (2 дня).

Нормы времени рассчитаны на комплексы задач (задачи) и указаны в человеко-днях.

При расчете фактических затрат времени на программирование необходимо учесть влияние таких факторов, как:

- степень новизны комплекса задач;
- сложность алгоритма;
- виды используемой информации;
- сложность контроля входной и выходной информации.

Предусматриваются четыре степени новизны разрабатываемых задач:

А – разработка комплекса задач, предусматривающая применение принципиально новых методов разработки, проведение научно-исследовательских работ;

Б – разработка типовых проектных решений, оригинальных задач и систем, не имеющих аналогов;

В – разработка проекта с использованием типовых проектных решений, при условии их изменения; разработка проектов, имеющих аналогичные решения;

Г – привязка типовых проектных решений.

Так как, данное приложение уже имеет аналоги на рынке, то степень новизны можно определить, как – разработка проекта с использованием типовых проектных решений, при условии их изменения, разработка проектов, имеющих аналогичные решения (В).

Сложность алгоритма

Сложность алгоритма представлена тремя группами:

- 1) алгоритмы оптимизации и моделирования систем и объектов;
- 2) алгоритмы учета и отчетности, статистики, поиска;
- 3) алгоритмы, реализующие стандартные методы решения, а также не предусматривающие применения сложных численных и логических методов.

Сложность данного приложения при разработке была определена как «Алгоритмы оптимизации и моделирования систем и объектов» (1).

Трудоемкость разработки проекта зависит также от вида:

- 1) используемой информации:

- ПИ – переменной информации (4);
- НСИ – нормативно-справочной информации (1);
- БД – банк данных (4);

2) разработки и режима работы:

- РВ – режим работы в реальном времени;
- ТОУ – телекоммуникационная обработка данных и управление удаленными объектами; от объема входной информации.

Данное приложение относится к системе реального времени (РВ).

Сложность организации контроля.

Сложность организации контроля входной и выходной информации представлена следующими группами:

11 – входные данные и документы разнообразного формата и структуры. Контроль осуществляется перекрестно, т.е. учитывается связь между показателями различных документов;

12 – входные данные и документы однообразной формы и содержания, осуществляется формальный контроль;

21 – печать документов сложной многоуровневой структуры разнообразной формы и содержания;

22 – печать документов однообразной формы и содержания, вывод массивов данных на машинные носители.

Данное приложение предполагает следующий уровень сложности:

- входная информация – 12;
- выходная информация – 22.

Все необходимые коэффициенты выбираются из таблиц, представленных в приложении А.

При использовании информации различных видов, поправочный коэффициент на стадиях «Технический проект» и «Рабочий проект» рассчитывается по формуле:

$$K_{II} = \frac{K_1 \cdot m + K_2 \cdot n + K_3 \cdot p}{m + n + p}, \quad (3.3)$$

где K_{II} – поправочный коэффициент; K_1, K_2, K_3 – поправочные коэффициенты согласно таблицам А.3 и А.4; m, n, p – количество наборов данных переменной информации (ПИ), нормативно-справочной информации (НСИ), информации при использовании банка данных (БД) соответственно.

$m = 4, n = 1, p = 4;$

$K_1 = 1, K_2 = 0,72, K_3 = 2,08;$

Технический проект: $K_{II} = (1 \cdot 4 + 0,72 \cdot 1 + 2,08 \cdot 4) / (4 + 1 + 4) = 1,45.$

Рабочий проект: $K_{II} = (1,20 \cdot 4 + 0,65 \cdot 1 + 0,54 \cdot 4) / (4 + 1 + 4) = 0,85.$

Расчет общей трудоемкости

Общий поправочный коэффициент $K_{об}$ определяется как произведение всех применяемых коэффициентов по следующей формуле:

$$K_{об} = K_1 \cdot K_2 \cdot \dots \cdot K_n, \quad (3.4)$$

где K_1, K_2, \dots, K_n – поправочные коэффициенты, учитывающие влияние факторов на изменение затрат времени при выполнении конкретной стадии проектирования, $K_{об}$ – общий поправочный коэффициент (i-го вида работы).

Стадии разработки:

- техническое задание – 3 дня;
- эскизный проект – 7 дней;
- технический проект – 15 ($K_{об} = 1,45 \cdot 1,00 \cdot 1,26 = 1,83$);
- рабочий проект – 35 ($K_{об} = 0,85 \cdot 1,00 \cdot 1,36 = 1,15$);
- внедрение – 2 ($K_{об} = 1,21 \cdot 1,00 = 1,21$).

Трудоемкость по этапам с учетом коэффициентов:

- техническое задание – 3 дня;
- эскизный проект – 7 дней;

- технический проект – 27 дней;
- рабочий проект – 40 дней;
- внедрение – 3 дня.

Расчет общей трудоемкости разработки проекта $T_{об}$ производится по формуле:

$$T_{об} = \sum_1^n t, \quad (3.5)$$

где t – трудоемкость работ по стадиям проектирования (от 1 до n), дней.

$$T_{об} = 3 + 7 + 27 + 40 + 2 = 79 \text{ дней.}$$

Численность исполнителей, необходимая для выполнения работ по стадиям проектирования и по комплексам задач (задаче) в целом вычисляется по формуле:

$$Ч = \frac{T_{об}}{\Phi_{пл}}, \quad (3.6)$$

где $Ч$ – численность специалистов чел, $T_{об}$ – общая трудоемкость разработки проекта, $\Phi_{пл}$ – плановый фонд рабочего времени одного специалиста.

$$Ч = 79 / 150 = 0,53.$$

Поскольку коэффициент численности исполнителей не превышает единицы, значит для разработки приложения достаточно одного специалиста.

3.3.3 Смета затрат на разработку программного обеспечения

В смету затрат на разработку программного обеспечения включаются:

- основная и дополнительная зарплата разработчика;
- отчисления на социальные нужды;

- стоимость машинного времени на подготовку и отладку программ;
- материальные затраты.

Основная заработная плата разработчика ($Z_{осн}$) рассчитывается по следующим формулам:

$$Z_{осн} = T_{об} \cdot Z_{ср.дн}, \quad (3.7)$$

$$Z_{ср.дн} = \frac{Z_{ср.мес}}{K_{р.д}}, \quad (3.8)$$

где $Z_{ср.дн}$ – среднедневная зарплата персонала в рублях, $T_{об}$ – общая трудоемкость проекта в днях, $Z_{ср.мес.}$ – среднемесячная заработная плата, $K_{р.д.}$ – среднее количество рабочих дней в месяце.

Среднемесячная заработная плата рассчитана и составляет 9000 рублей в месяц.

$$K_{р.д.} = 22 \text{ дня.}$$

$$Z_{ср.дн} = 12\,000 / 22 = 545,45 \text{ р.}$$

$$Z_{осн} = 545,45 * 80 = 43636,36 \text{ р.}$$

Дополнительная заработная плата рассчитывается по формуле:

$$Z_{доп} = Z_{осн} \cdot 10\% \quad (3.9)$$

$$Z_{доп} = 43636,36 * 0,1 = 4363,636 \text{ р.}$$

Отчисления на социальные нужды составляют 26% от основной и дополнительной заработной платы или 11345,45 рублей.

Стоимость машинного времени зависит от себестоимости машино-часа работы ЭВМ и времени работы ЭВМ и включает амортизацию ЭВМ и оборудования (A_m) и затраты на электроэнергию ($Z_{эл}$).

$$A_m = \frac{O_\phi \cdot H_{ам}}{365 \cdot 100} \cdot T_m, \quad (3.10)$$

где O_ϕ – стоимость ЭВМ и оборудования в рублях, $H_{ам}$ – норма амортизации, принята равной 25%, A_m – амортизационные отчисления, денежные единицы, T_m – время использования оборудования в днях, равное:

$$T_m = 0,3 \cdot T_{тех.пр} + 0,8 \cdot T_{раб.пр} + 0,6 \cdot T_{вн}, \quad (3.11)$$

где $T_{тех.пр}$, $T_{раб.пр}$, $T_{вн}$ – затраты времени на разработку технического проекта, рабочего проекта и внедрения соответственно.

$$T_m = 0,3 \cdot 30 + 0,8 \cdot 35 + 0,6 \cdot 3 = 39 \text{ дней.}$$

Средняя стоимость компьютера, системные параметры которого будут достаточны для реализации проекта, составляет 25000 рублей, норма амортизации, принята равной 20%.

$$A_m = (25000 \cdot 20 / 365 \cdot 100) \cdot 39 = 534 \text{ р.}$$

Затраты на электроэнергию находятся по формуле:

$$Z_{эл} = C_{эл} \cdot M_{эвм} \cdot T_m \cdot T_{сут}, \quad (3.12)$$

где $C_{эл}$ – стоимость 1 кВт/ч электроэнергии в рублях (2,3 рубля), $M_{эвм}$ – мощность ЭВМ (1,7 кВт/ч), $T_{сут}$ – суточное время работы ЭВМ в часах (8 ч.).

$$Z_{эл} = 2,3 \cdot 1,7 \cdot 35 \cdot 8 = 1095 \text{ р.}$$

Материальные затраты можно определить в размере 100 рублей. Это затраты на различного рода расходные материалы.

Полный перечень затрат на разработку программного обеспечения представлен в таблице 3.1.

Таблица 3.1 – Таблица затрат на разработку программного обеспечения

Элемент затрат	Стоимость (в рублях)
Основная заработная плата	43636,3636

Дополнительная заработная плата	4363,63636
Отчисления на социальные нужды	11345,4545
Амортизация ЭВМ и оборудования	534
Затраты на электроэнергию	1095
Материальные затраты	100
ИТОГО:	61074,4545

Таким образом невысокие затраты на создание мобильной версии портала, экономия трафика и времени пользователей (около 85%), повышение удобства пользования порталом принесут значительную социальную эффективность.

Можно выделить следующие положительные стороны внедрения мобильной версии портала:

- увеличится количество посетителей за счет пользователей мобильного интернета, а, следовательно, и потенциальных клиентов Института, возрастет престижность ВУЗа;

- сократится время доступа к информации, повысится удобство доступа;
- сократятся случаи, связанные с опозданиями;
- сократятся затраты пользователей в связи со снижением трафика.

В итоге Институт не только не теряет потенциальных клиентов, но и наоборот приобретает совершенно новую аудиторию пользователей.

Заключение

В данном проекте объектом исследования являлся информационно-образовательный портал Рубцовского Института (филиала) Алтайского Государственного Университета, а предметом исследования – организация доступа к ресурсам информационно-образовательного портала.

Целью исследования была разработка мобильного приложения для организации доступа к ресурсам информационно-образовательного портала Рубцовского Института (филиала) Алтайского Государственного Университета (на примере «РИ АлтГУ – Расписание» для платформы Android).

Цель достигнута полностью за счет решения следующих задач:

- дана характеристика исследуемой предметной области;
- проведен анализ предметной области;
- построена функциональная модель существующей организации доступа к ресурсам информационно-образовательного портала;
- выявлены недостатки существующей системы, на основании выявленных недостатков предложена функциональная модель «Как должно быть»;
- разработано мобильное приложение «РИ АлтГУ – Расписание»;
- оценена эффективность от внедрения мобильного приложения.

Результатом проекта является мобильное приложение «РИ АлтГУ – Расписание», которое позволяет:

- получать расписание занятий, консультаций и комиссий;
- корректно отображать информации на любых экранах мобильных устройств;
- обновлять списки групп, преподавателей и аудиторий;
- сохранять пользовательские настройки;
- автоматически загружать в фоне выбранное расписание;
- использовать виджет для отображения расписания на главном экране

или экране блокировки;

- оповещать об изменившемся расписании;
- сэкономить мобильный трафик.

Описанный способ доступа к ресурсам информационно-образовательного портала является универсальным – может использовать информацию любого вида, а также является платформонезависимым.

В качестве дальнейшего развития планируется реализация доступа к новостям портала, а также к списку задолженностей студента.

Приложение «РИ АлтГУ – Расписание» доступно для загрузки всем пользователям платформы Android в магазинах приложений: Google Play и Yandex Store.

Список использованных источников

1. Антонов, А.В. Системный анализ / А.В. Антонов. – М.: Высш. Шк., 2004.- 454 с.
2. Башмаков, А.И. Теория систем и системный анализ: методические указания по выполнению курсовой работы по курсу «Теория систем и системный анализ» по направлениям «Прикладная математика и информатика» и «Экономика» / А.И. Башмаков. – М.: Изд. дом МЭИ, 2010. – 52 с.
3. Волкова, В.Н. Теория систем и системный анализ в управлении организациями: Справочник: Учебное пособие / В.Н. Волкова, А.А. Емельянова. – М.: Финансы и статистика, 2006. – 848 с.
4. Глумаков, С.В. Базы данных: Учебный курс / С.В. Глумаков, Д.В. Ломотько. – М.: АСТ, 2002. – 504 с.
5. Голицына, О. Л. Базы данных: Учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – Форум, Москва – 2009. – 400 с.
6. Головач, В.В. Дизайн пользовательского интерфейса: учебное пособие / В.В. Головач. – М.: Наука, – 132 с.
7. Гультияев, А.К. Проектирование и дизайн пользовательского интерфейса / А.К. Гультияев, В.А. Машин. – СПб: КОРОНА принт, 2010. – 352 с.
8. Дженифер Тидвелл. Разработка пользовательских интерфейсов / книга / Переводчик Е. Шикарева. – Питер 2008. – 416 с.
9. Диго, С.М. Базы данных. Проектирование и создание: учебно-методический комплекс / С.М. Диго. – М.: Изд. центр ЕАОИ, 2008. – 172 с.
10. Дубейковский, В. И. Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM.: Книга / В.И. Дубейковский. – Диалог-МИФИ, 2007. – 464 с.
11. Калянов, Г.Н. Case-технологии: Консалтинг в автоматизации бизнес-процессов: книга / Г. Н. Калянов. – Горячая Линия, Телеком, 2002. – 320 с.

12. Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб: БХВ – Петербург, 2009. – 464 с.
13. Концепция информатизации Рубцовского Института (филиала) Алтайского Государственного Университета.
14. Корнеев, В.В. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх. – М.: Нолидж, 2001. – 496 с.
15. Кулябов, Д.С. Введение в формальные методы описания бизнес-процессов: Учебное пособие / Д.С. Кулябов, А.В. Королькова, 2008. – 173 с.
16. Маклаков, С.В. Создание информационных систем с AllFusionModelingSuite / С.В. Маклаков. – М.: Диалог-МИФИ, 2004. – 432 с.
17. Маховикова, Г.А. МЕНЕДЖМЕНТ: УЧЕБНЫЙ КУРС / Г.А. Маховикова, В.Е. Кантор. – Эксмо, 2013. – 312 с.
18. Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников, А.М. Петраков, С.А. Клейменов. – М.: АСАДЕМА, 2009. – 336 с.
19. Мишенин, А.И. Теория экономических информационных систем / А.И. Мишенин. – М.: Финансы и статистика, 2005. – 240 с.
20. Гвоздева, Т.В. Проектирование информационных систем: учебное пособие / Т.В. Гвоздева, Б.А. Баллод. – Ростов-н/Д: Феникс, 2009. – 508 с.
21. Приказ Федеральной службы по техническому и экспортному контролю (ФСТЭК России) Федеральной службы безопасности Российской Федерации (ФСБ России) Министерства информационных технологий и связи Российской Федерации (Мининформсвязи России) от 13 февраля 2008 г. N 55/86/20 г. Москва.
22. Родичев, Ю.А. Информационная безопасность: Нормативно-правовые аспекты : Учебное пособие / Ю.А. Родичев. – СПб: Питер, 2008. – 272 с.
23. Титоренко, Г.А. Информационные системы в экономике: учебник / Г.А. Титоренко. – Юнити-Дана, 2012. – 463 с.
24. Федеральный закон «О персональных данных» от 27 июля 2006 года

№ 152-ФЗ. (08.062015)

25. Функциональное моделирование на базе стандарта IDEF0. Учебный курс – Минск, 2002. – 35 с.

26. Хомоненко, А.Д. Базы данных: Учебник для высших учебных заведений / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев. – СПб: КОРОНА принт, 2011. – 672 с.

27. Ярочкин, В.И. Информационная безопасность: Учебник для вузов / В.И. Ярочкин. – М.: Мир, 2008. – 640 с.

28. Сообщество PHP-Программистов, документация PHP, форум php-программистов / <http://www.php.ru/> (08.06.2015)

29. Элементы модели «сущность-связь» / www.citforum.ru/database/ (08.062015)

30. XServer.ru – бесплатная виртуальная электронная Интернет библиотека онлайн. Прямое и Обратное Проектирование / <http://www.xserver.ru/computer/database/erwin/2/19.shtml> (08.06.2015)

31. Википедия свободная энциклопедия Java. / <https://ru.wikipedia.org/wiki/Java> (08.06.2015)

32. Википедия свободная энциклопедия. PHP / <https://ru.wikipedia.org/wiki/PHP> (08.06.2015)

33. Википедия свободная энциклопедия. Фреймворк / <https://ru.wikipedia.org/wiki/%D4%F0%E5%E9%EC%E2%EE%F0%EA> (08.06.2015)

34. Обзор мультиплатформенных фреймворков – Sencha Touch / http://peekaboo-games.com/framework_sencha/ (08.06.2015)

35. Nabrahabr jQuery Mobile / <http://habrahabr.ru/post/194030/> (08.06.2015)

36. Википедия свободная энциклопедия Android Studio / https://ru.wikipedia.org/wiki/Android_Studio (08.06.2015)

37. Установка Eclipse и подключение плагина Android Development Tools (ADT) / <http://www.fandroid.info/ustanovka-eclipse-i-podklyuchenie-plagina-android-development-tools-adt-dlya-razrabotki-android-prilozhenij/> (08.06.2015)

38. Википедия свободная энциклопедия Embarcadero RAD Studio /
https://ru.wikipedia.org/wiki/Embarcadero_RAD_Studio (08.06.2015)

39. Википедия свободная энциклопедия JSON /
<https://ru.wikipedia.org/wiki/JSON> (08.05.2015)

40. Википедия свободная энциклопедия Сериализация /
<https://ru.wikipedia.org/wiki/%D1%E5%F0%E8%E0%EB%E8%E7%E0%F6%E8%FF> (08.05.2015)

Приложение А

Вспомогательные таблицы для расчета экономической эффективности

Таблица А.1 – Затраты времени при выполнении работ на стадии «Техническое задание»

Комплекс задач, задачи подсистем	Степень новизны			
	А	Б	В	Г
Перспективное планирование развития и размещения отрасли, управление проектированием и капитальным строительством, технико-экономическое планирование, оперативное управление, управление ценообразованием	79	57	37	34
Управление материально-техническим снабжением, управление сбытом продукции, управление комплектацией, управление экспортными и импортными поставками	105	76	42	30
Бухгалтерский учет, управление финансовой деятельностью	103	72	30	35
Управление организацией труда и заработной платой, управление кадрами, нормы и нормативы, управление охраной труда	63	46	30	19
Управление качеством продукции, управление технологическими процессами, управление стандартизацией, управление технической подготовкой производства	64	47	31	22
Управление транспортными перевозками, управление техническим обслуживанием производства, управление вспомогательными службами и энергоснабжением	91	66	43	26
Управление НИР и ОКР	50	36	24	15
Управление научно-технической информацией	50	36	24	15
Совершенствование документооборота и контроль исполнения документов	50	36	24	15
Управление охраной природы и окружающей средой	50	36	24	15
Учет пенсий, пособий и страховых операций	79	55	36	26
Статистические задачи	12	111	61	38
Задачи расчетного характера	92	69	47	29

Таблица А.2 – Затраты времени при выполнении работ на стадии «Эскизный проект»

Комплекс задач, задачи подсистем	Степень новизны			
	А	Б	В	Г
Перспективное планирование развития и размещения отрасли, управление проектированием и капитальным строительством, технико-экономическое планирование, оперативное управление, управление ценообразованием	175	117	77	53
Управление материально-техническим снабжением, управление сбытом продукции, управление комплектацией, управление экспортными и импортными поставками	115	79	53	35
Бухгалтерский учет, управление финансовой деятельностью	166	112	67	57
Управление организацией труда и заработной платой, управление кадрами, нормы и нормативы, управление охраной труда	151	101	67	46
Управление качеством продукции, управление технологическими процессами, управление стандартизацией, управление технической подготовкой производства	157	99	67	44
Управление транспортными перевозками, управление техническим обслуживанием производства, управление вспомогательными службами и энергоснабжением	170	100	70	45
Управление НИР и ОКР	151	101	67	46
Управление научно-технической информацией	151	101	67	46
Совершенствование документооборота и контроль исполнения документов	151	101	67	46
Управление охраной природы и окружающей средой	151	101	67	46
Учет пенсий, пособий и страховых операций	103	70	45	36
Статистические задачи расчетного характера				49
				41

Таблица А.3 – Поправочные коэффициенты для определения трудоемкости работ стадии «Технический проект» ($K_1 K_2 K_3$)

Вид используемой информации	Степень новизны			
	А	Б	В	Г
ПИ	1,70	1,20	1,00	0,50
НСИ	1,45	1,08	0,72	0,43
БД	4,37	3,12	2,08	1,25

Таблица А.4 – Поправочные коэффициенты для определения трудоемкости работ стадии «Рабочий проект» ($K_1 K_2 K_3$)

Вид используемой информации	Группа сложности алгоритма	Степень новизны			
		А	Б	В	Г
ПИ	1	2,27	1,62	1,20	0,65
	2	2,02	1,44	1,10	0,58
	3	1,68	1,20	1,00	0,48
НСИ	1	1,36	0,97	0,65	0,40
	2	1,21	0,86	0,58	0,34
	3	1,01	0,72	0,48	0,29
БД	1	1,14	0,81	0,54	0,32
	2	1,05	0,72	0,48	0,29
	3	0,85	0,60	0,40	0,24

Таблица А.5 – Поправочные коэффициенты, учитывающие сложность контроля входной и выходной информации на стадиях “Рабочий проект” и “Внедрение”

Сложность контроля входной информации	Сложность контроля выходной информации	
	21	22
11	1,16	1,07
12	1,08	1,00

Таблица А.6 – Поправочные коэффициенты, учитывающие вид информации на стадиях «Рабочий проект», «Внедрение» и «Технический проект»

Стадия разработки проекта	Вид обработки	Степень новизны			
		А	Б	В	Г
Технический проект	РВ	1,67	1,45	1,26	1,10
	ТОУ	1,75	1,52	1,36	1,15
Рабочий проект	РВ	1,75	1,52	1,36	1,15
	ТОУ	1,92	1,67	1,44	1,25
Внедрение	РВ	1,60	1,39	1,21	1,05
	ТОУ	1,67	1,45	1,26	1,10