СОДЕРЖАНИЕ

ВВЕДЕНИЕ 6

1. АНАЛИЗ ТРЕБОВАНИЙ 7

1.1. Мониторинг успеваемости и посещаемости студентов 7

1.1.1. Успеваемость и посещаемость 7

1.1.2. Традиционный способ мониторинга успеваемости и посещаемости 8

1.1.3. Разновидности журналов 9

1.1.4. Пример мониторинга успеваемости и посещаемости 10

1.1.5. Мониторинг посещаемости и успеваемости на кафедре ИиПО в Google Docs 14

1.2. Описание программ-аналогов 15

1.2.1. Мобильное приложение Классный журнал. Руководство учителя 15

1.2.2. Мобильное приложение Дневник в кармане 16

1.2.3. LMS Школа. Классный журнал 17

1.3. Модель потоков данных разрабатываемой системы 19

1.4. Функциональные требования 19

1.4.1. Авторизация и роли 22

1.4.2. Учетные единицы 22

1.4.3. Возможности роли «Преподаватель» 22

1.4.4. Возможности роли «Староста» 23

1.4.5. Возможности роли «Студент» 23

1.5. Выводы 23

2. ЭКОНОМИЧЕСКИЙ АНАЛИЗ 25

2.1. Организационная структура проекта 25

2.2. Календарный план проекта 25
2.3. Расчёт затрат на разработку продукта 27
2.3.1. Расчёт заработной платы исполнителей работ по созданию программного продукта 28
2.3.2. Расчёт отчислений на социальные нужды 29
2.3.3. Арендные платежи за производственные помещения 29
2.3.4. Амортизация используемых основных средств и нематериальных активов 30
2.3.5. Расходы на модернизацию и приобретение основных средств ... 31
2.3.6. Расходы на приобретение необходимого ПО 31
2.3.7. Расходы на интернет и связь 31
2.3.8. Расходы на канцелярские товары и расходные материалы 31
2.3.9. Прочие расходы 31
2.3.10. Расчёт себестоимости программного продукта 31
3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 33
3.1. Архитектура программного продукта 33
3.2. Модель данных 36
3.3. Низкоуровневое проектирование 40
3.4. Проектирование интерфейсов 42
3.4.1. Проектирование интерфейса преподавателя 42
3.4.2. Проектирование интерфейса старосты 44
3.4.3. Проектирование интерфейса студента 45
3.5. Реализация модулей приложения 46
3.5.1. Модуль авторизации 46
3.5.2. Модуль вывода параметров фильтрации 46

3.5.3. Модуль добавления ведомости ... 48
3.5.4. Модуль редактирования ведомости ... 48
3.5.5. Модуль просмотра ведомости ... 49
3.5.6. Серверная часть ... 49
3.6. Руководство пользователя ... 51
3.6.1. Запуск приложения ... 51
3.6.2. Авторизация ... 52
3.6.3. Преподаватель ... 52
3.6.4. Староста ... 56
3.6.5. Студент ... 59
3.6.1. Завершение работы приложения ... 61
4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ ... 62
4.1. План испытаний ... 62
4.2. Проверка выполнения функциональных требований ... 63
4.2.1. Проверка авторизации ... 63
4.2.2. Проверка в случае обрыва соединения с интернетом ... 63
4.2.3. Проверка добавления ведомости ... 63
4.2.4. Проверка редактирования ведомости преподавателем ... 63
4.2.5. Проверка редактирования ведомости старостой ... 64
4.2.6. Проверка экрана просмотра ведомости ... 64
4.3. Нагрузочное тестирование ... 65
5. ОРГАНИЗАЦИОННАЯ ЧАСТЬ ... 66
5.1. Общая характеристика радиочастотных излучений ... 66
5.2. Медицинские исследования влияния мобильных телефонов на здоровье человека ... 67

5.3. Вероятные клинические последствия влияния мобильных телефонов68

ЗАКЛЮЧЕНИЕ 72

СПИСОК ЛИТЕРАТУРЫ 73

ВВЕДЕНИЕ

В марте 2012 года Дневник.ру предложил нескольким регионам страны заменить бумажные журналы на планшетные компьютеры. Первой в эксперимент вступила Нижегородская область, позже к ней присоединились ещё пять регионов: Алтай, Новосибирская, Тамбовская и Астраханская области, Хабаровский край. Четырнадцати наиболее активным образовательным учреждениям проект предоставил в пользование планшетные компьютеры с предустановленной системой электронных классных журналов.

Почти 90% российских учителей, участвующих в эксперименте по введению электронного классного журнала, положительно оценили нововведение, сообщили во вторник РИА Новости в пресс-службе образовательного интернет-проекта Дневник.ру [4].

В настоящее время существует приличное количество электронных журналов для школы, со своими достоинствами и недостатками, а журналов для университетов с мониторингов процесса обучения почти нет.

Целью данной работы является разработка мобильного приложения для автоматизации ведения журнала успеваемости и посещаемости занятий и повышения удобства работы преподавателей.

Для достижения этой цели были поставлены следующие **задачи**:

1. Проанализировать процесс ведения журнала.
2. Рассмотреть программы аналоги, выявить их сильные и слабые стороны.
3. Спроектировать будущее приложение согласно полученным данным.
4. Реализовать и отладить мобильное приложение.

Объектом является мониторинг успеваемости и посещаемости студентов. **Предметом** является автоматизация мониторинга успеваемости и посещаемости студентов.

1. АНАЛИЗ ТРЕБОВАНИЙ

1.1. Мониторинг успеваемости и посещаемости студентов

Для принятия управленческих решений, куратору группы и администрации университета необходима информация о текущей успеваемости студентов.

1.1.1. Успеваемость и посещаемость

Успеваемость – степень усвоения объема знаний, навыков, умений, установленных учебной программой, с точки зрения их осмысленности, полноты, глубины, прочности.

Следует отметить, что успешность обучения является одной из составляющих психологического здоровья студентов, сохранением которого обеспокоено всё психолого-педагогическое сообщество.

Проблема успешности обучения волнует умы ученых уже не одно столетие. Со времен Я.А. Коменского и по настоящее время продолжается изучение сущности, причин неуспехов учащихся в учебе, разрабатываются практические рекомендации.

Несмотря на длительность изучения, данная проблема остается еще недостаточно хорошо исследованной.

Учебная посещаемость как явление социально-педагогическое детерминирует академическую успеваемость и воспитание учащихся.

Посещаемость тесно связана с решением ряда организационных задач учебного процесса, с решением нравственных и социальных проблем семьи и образовательного учреждения, оказывает существенное влияние на их связь между собой, а также на создание имиджа образовательного учреждения и признание высокой квалификации педагогических работников. Посещение учебных занятий – одна из обязанностей учащихся.

1.1.2. Традиционный способ мониторинга успеваемости и посещаемости

Традиционный способ контроля процесса обучения в образовательном учреждении заключается в использовании журнала успеваемости, находящегося у преподавателя [7].

Журнал обычно бывает в виде бумажной книги. В нем преподаватель отмечает посещение занятий обучающимися, их успеваемость и темы занятий (задания). Обучающийся (студент, школьник, гимназист) периодически нуждается в информации, размещенной в этом журнале. Не всегда обучающийся точно представляет, как в журнале отражён процесс его личного обучения. Ему также не ясно, какие темы занятий были в то время, когда он отсутствовал или какие темы и задания запланированы на будущих занятиях. Временами это приводит к потери связи обучающегося с процессом обучения, отсутствием возможности самоорганизации и планирования процесса собственного обучения.

Преподавателю в этой ситуации приходится тратить время на напоминание обучающимся о задолженностях (о несданных учебных работах), да и сами обучающиеся постоянно интересуются ходом их обучения, прося преподавателя показать журнал, чтобы сверить свою информацию о том, как они представляют свой процесс освоения учебной дисциплины с фактически отраженным в журнале успеваемости. Нередки случаи рассогласованности двух информаций: имеющихся у обучающегося и у преподавателя.

На синхронизацию информации, имеющейся у обучающегося и у преподавателя затрачиваются время и силы обоих сторон. Кроме того, традиционная система не позволяет обучаемому выполнять самоорганизацию собственного процесса обучения, согласуя это с выполнением учебных работ (прохождением будущих учебных тем), будущих учебных занятий и сроками сдачи аттестационных работ. Проблема высвобождения времени

преподавателя от непроизводительных разговоров с обучающимися (многократные напоминания о графике учебного процесса, очередности выполнения аттестационных работ, наименований зачтенных и не зачтенных работ) тоже насущна.

Помимо указанного, реализация оперативного информирования обучающихся о процессе обучения могла бы стать действенным инструментом для формирования и укрепления у обучающихся навыков самообучения, самоконтроля и самоорганизации.

Поэтому решение проблемы оперативного отображения успеваемости в образовательном учреждении важна как для обучающихся, так и для преподавателей.

Для решения проблемы оперативного отображения успеваемости (оперативного информирования обучающихся) в образовательном учреждении была сделана попытка создания модели электронного журнала.

Часто при ведении журнала ставятся следующие задачи:

- в журнале должна быть полная информация обо всех аттестациях и всех темах учебных занятий за весь учебный период (семестр - для вуза и четверть или пол года для школы);
- дополнительно к журналу должны быть все тексты к заданиям и тексты лекционных (теоретических занятий);
- доступность к размещенной в нем информации в любое время (на занятиях и вне занятий);
- запрет редактирования журнала обучающимися;
- автоматическое отображение статистики процесса обучения в таблицах и в диаграммах;
- доступ к информации через Интернет.

1.1.3. Разновидности журналов

Электронный дневник предполагает следующие возможности:

- информация о домашних заданиях и оценках;
- оперативное взаимодействие преподавателя и обучающегося;
- портфолио обучающегося, включающее его активность в колледже и за его пределами;
- доступ через личный кабинет, использование личных настроек приватности;
- оперативное получение родителями достоверной информации об успеваемости и заданиях для обучающихся.

Электронный журнал:

- информация о домашних заданиях и оценках;
- оперативный контроль выполнения учебной нагрузки;
- контроль учета часов преподавателей;
- повышение ИКТ-компетентности сотрудников колледжа, обеспечение нового уровня культуры проведения учебных занятий;
- доступность электронного журнала в любое время и с любого компьютера;
- возможность ведения журнала практики (учебной, производственной, преддипломной);
- инструмент работы с курсовыми работами;
- накопление и систематизация учебно-методических наработок, формирование портфолио преподавателей;
- анализ успеваемости и посещаемости обучающихся, корреляция качества усвоения учебного материала по темам с учетом используемых методик и технологий обучения.

1.1.4. Пример мониторинга успеваемости и посещаемости

По аналогии с обычным бумажным журналом успеваемости был сделан электронный журнал в программе MS Excel [2]. Преподаватель имел к

электронному журналу полный доступ, обучающиеся имели доступ только для чтения. Обычный журнал, в случае необходимости, мог быть получен распечаткой электронного журнала на бумажных листах.

Испытания модели проходили в течение одного семестра в нескольких учебных группах студентов ХГАЭП.

По результатам испытаний проводилось анкетирование учащихся и преподавателей о полезности новой модели.

Большинству студентов модель понравилась. Они отмечали ее удобность, наглядность, возрастание уровня собственного контроля (самоконтроля) за успеваемостью. Возможность заранее выполнять лабораторные работы и знакомиться с предстоящими лекциями. Были и безразличные к этой новинке, но противников не было.

Преподавателями было отмечено возрастание лидерских и соревновательных настроений в учебном коллективе, повышение уровня самостоятельности и самооценки студентов за счет непрерывного наблюдения уровня роста собственной успеваемости.

Наряду с этим преподаватели отметили возрастание собственной нагрузки и напряжения за счет смены традиционных навыков ведения бумажного журнала (например, необходимости постоянного внесения изменений в электронный журнал, который надо было постоянно иметь не только на практических занятиях, но и на лекциях), а также большими затратами времени на работу с электронной почтой.

Поскольку предложенный метод взаимодействия с обучаемыми имел важные положительные итоги, было решено продолжить эксперимент и создать новую модель электронного журнала в образовательном учреждении. Было решено изменить (реконструировать, откорректировать, модифицировать, переделать) модель электронного журнала в образовательном учреждении.

Поскольку то, что было предложено для обучаемых имело только положительные результаты, а для преподавателей – как положительные, так и

отрицательные, было решено отказаться от всего, что давало отрицательный результат.

Задача реконструкции модели была поставлена так, что если обучающимся нужен электронный журнал, то пусть они его и ведут, а если преподавателю он не нужен, то пусть он его не ведет, но пользуется его результатами.

Проблема высвобождения времени преподавателя в новой модели была решена за счет прекращения заполнения электронного журнала и возврата к более мобильному, удобному и традиционному – бумажному журналу.

Файл электронного журнала был создан по аналогии с прежним, но можно также выделить следующие отличия:

- обучаемые имели полный доступ только к таблице данных о выполнении ими графика учебных работ. Остальные разделы были им доступны в режиме чтения;
- преподаватель имел полный доступ ко всем страницам электронного журнала, но не заполнял его в течение семестра, подготовив его для обучающихся только в начале семестра;
- прием выполненных работ и консультации по электронной почте были отменены.

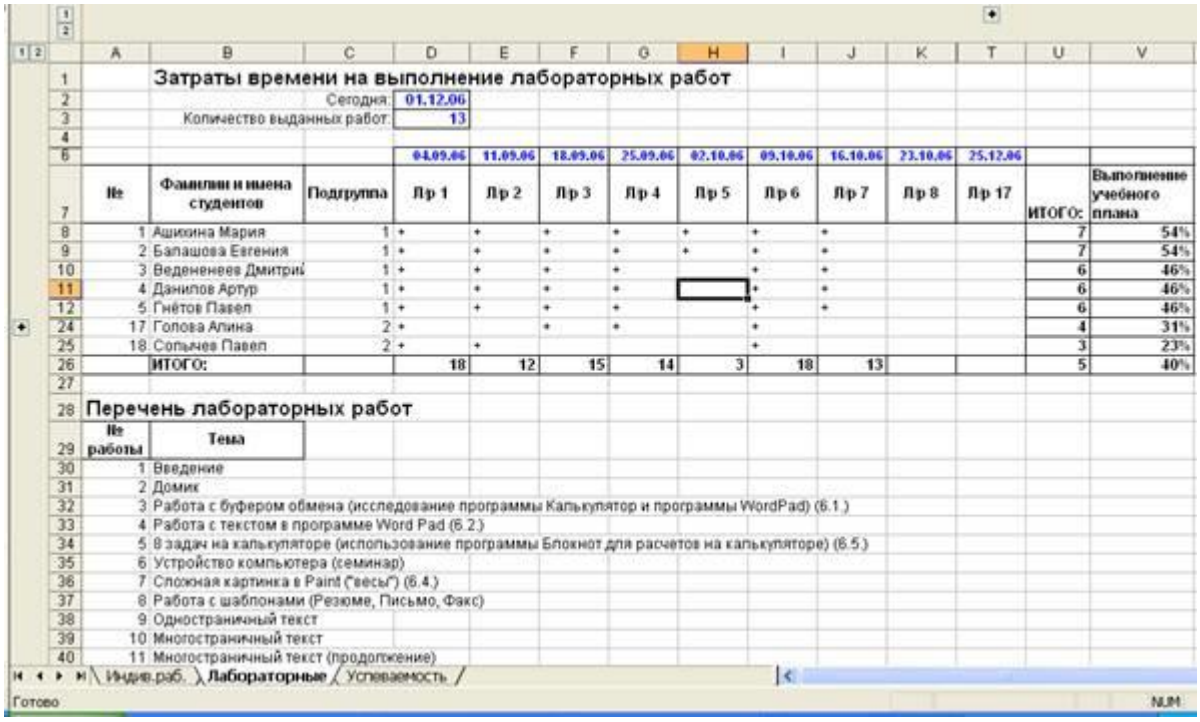
Новая модель электронного журнала в образовательном учреждении была названа Вариант 2 «Самоконтроль» рис. 1.1.

Модель содержала:

- список группы;
- даты отчетности по лабораторным работам;
- темы лабораторных работ;
- таблицу выполнения лабораторных работ (заполняемую самими обучаемыми);
- статистические данные успеваемости;
- диаграммы текущей успеваемости;

- таблицу выполнения индивидуальных работ.

Технология работы с электронным журналом описана ниже.



Затраты времени на выполнение лабораторных работ

Сегодня: 01.12.06

Количество выданных работ: 13

	№	Фамилии и имена студентов	Подгруппа	04.09.06 Лр 1	11.09.06 Лр 2	18.09.06 Лр 3	25.09.06 Лр 4	02.10.06 Лр 5	09.10.06 Лр 6	16.10.06 Лр 7	23.10.06 Лр 8	25.12.06 Лр 17	ИТОГО:	Выполнение учебного плана
8	1	Ашихина Мария	1	+	+	+	+	+	+	+			7	54%
9	2	Балашова Евгения	1	+	+	+	+	+	+	+			7	54%
10	3	Веденеев Дмитрий	1	+	+	+	+		+	+			6	46%
11	4	Данилов Артур	1	+	+	+	+		+	+			6	46%
12	5	Гнётов Павел	1	+	+	+	+		+	+			6	46%
24	17	Голова Алина	2	+		+	+		+				4	31%
25	18	Сопычев Павел	2	+	+				+				3	23%
26		ИТОГО:		18	12	15	14	3	18	13			5	40%

Перечень лабораторных работ

№ работы	Тема
1	Введение
2	Домик
3	Работа с буфером обмена (исследование программы Калькулятор и программы WordPad) (6.1.)
4	Работа с текстом в программе Word Pad (6.2.)
5	8 задач на калькуляторе (использование программы Блокнот для расчетов на калькуляторе) (6.5.)
6	Устройство компьютера (семинар)
7	Сложная картинка в Paint ("весы") (6.4.)
8	Работа с шаблонами (Резюме, Письмо, Факс)
9	Одностраничный текст
10	Многостраничный текст
11	Многостраничный текст (продолжение)

Индив.раб. / Лабораторные / Успеваемость

Готово

Рис. 1.1. Пример работы программы «Самоконтроль»

Преподаватель заполнял журнал вначале семестра (расставлял даты занятий, корректировал названия занятий) и открывал к нему доступ студентов. Текстовые материалы, необходимые для учебной дисциплины, как правило, уже подготовлены и изредка корректируются и модифицируются.

Студенты могли видеть даты занятий, темы и перечень аттестационных работ. Остальные учебные материалы – самые новые они могли брать в электроном виде из специального раздела локальной сети, а старые – из вузовской библиотеки.

Преподаватель вел традиционный бумажный журнал. При защите очередной аттестационной работы говорил студенту об этом, и студент самостоятельно отмечал это плюсом в электронном журнале. Статистика формировалась автоматически (рис. 1.2).

Доступ к редактированию данных был открыт только студентам одной группы, но результаты могли видеть студенты всех групп.

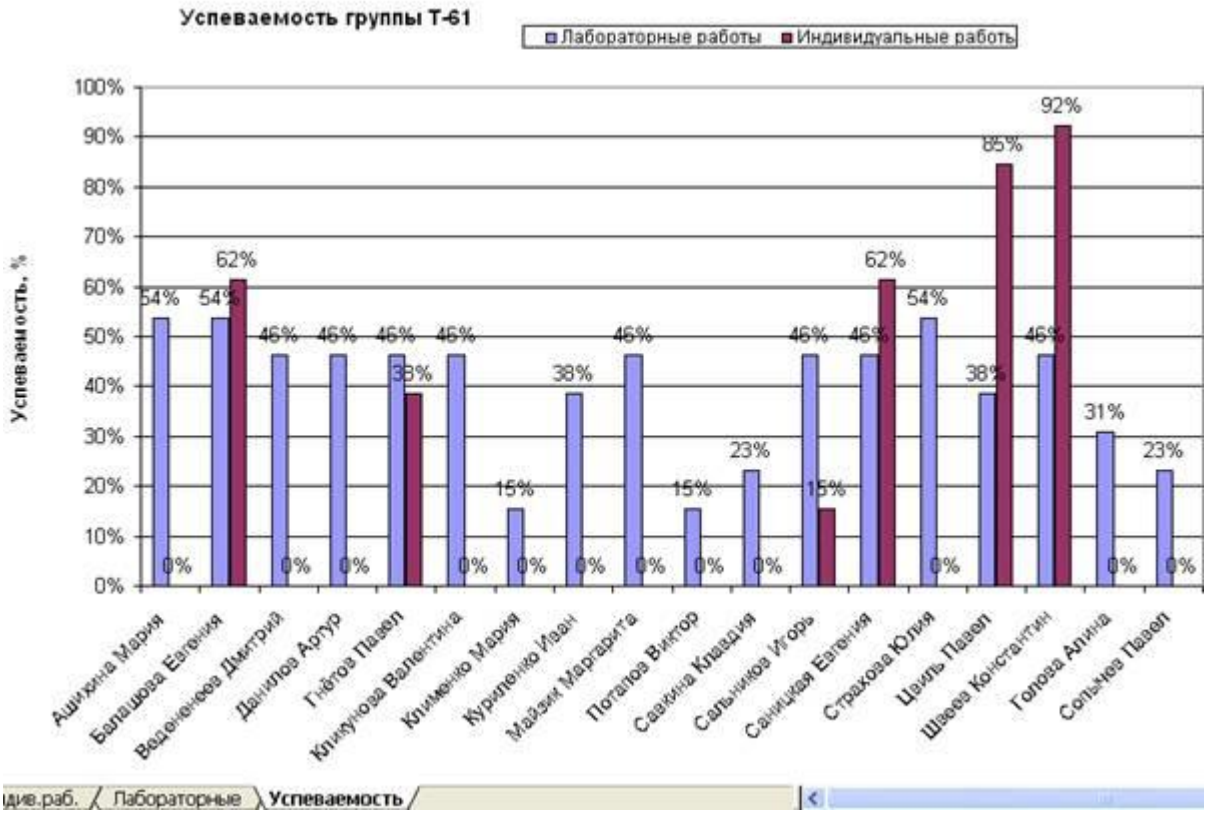


Рис. 1.2. Вывод автоматической статистики

Рассмотренная модель электронного журнала не застрахована от вандализма некоторых студентов и искушения у них проставить себе дополнительные плюсы. Для исключения этого преподаватель всё равно должен иметь свой традиционный журнал, данные которого не зависят от электронного журнала, поэтому искажение реальных данных в электроном журнале студенту не выгодно.

1.1.5. Мониторинг посещаемости и успеваемости на кафедре ИиПО в Google Docs

В качестве эксперимента на кафедре была создана ведомости в Google Docs, были настроены права доступа преподавателей к ведомости для редактирования своих ведомостей, а также построен ряд отчетов.

По результатам использования были сделаны выводы:

- основная проблема, перевод ведомости в состояние «подписано» для запрета дальнейшего редактирования старостой;

- неудобство редактирования в аудитории на компьютере;
- неудобство работы с ведомостью с мобильного устройства.

1.2. Описание программ-аналогов

Рассмотрим наиболее распространенные мобильные приложения, а также выделим их наиболее сильные стороны которые помогут в разработке.

1.2.1. Мобильное приложение Классный журнал. Руководство учителя

Мобильное приложение «Классный журнал» разработано ООО «Дневник.ру» для использования на мобильных устройствах с операционной системой Android и предназначено для осуществления оперативной работы по заполнению журналов успеваемости учителями образовательных учреждений [3].

Информация, вносимая через мобильное приложение «Классный журнал», хранится в базе данных Единой образовательной сети «Дневник.ру» и доступна при авторизации в ней. И наоборот – все изменения, внесенные в журнал класса в информационной системе «Дневник.ру», доступны в мобильном приложении (рис. 1.3).

Мобильное приложение «Классный журнал», имеет два представления:

- журнал класса позволяет выставлять, изменять и просматривать отметки присутствия и текущие оценки учащихся за каждый вид работы на уроке;
- страница урока позволяет изменять тему урока, добавлять новые типы работ на уроке и выставлять оценки и отметки присутствия, оставлять комментарий к уроку для каждого ученика, создавать и выдавать классу домашние задания.

Вывод. Наиболее интересным в рассмотренной программе является удачно спроектированный интерфейс, который может быть взят за основу при построении собственного приложения.

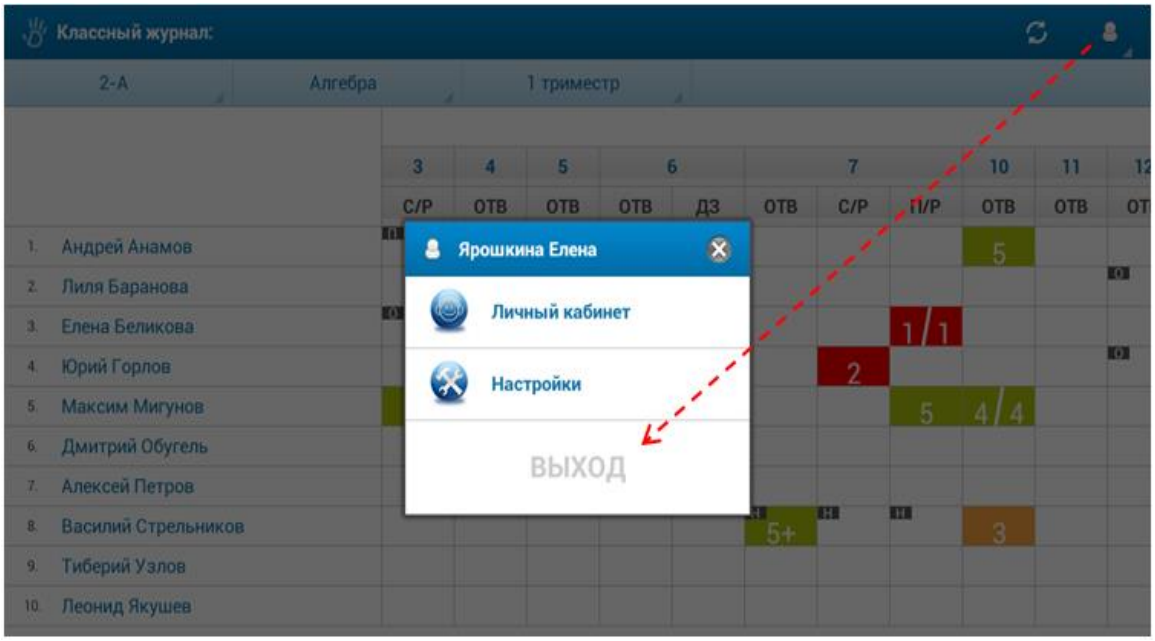


Рис. 1.3. Мобильное приложение «Классный журнал»

1.2.2. Мобильное приложение Дневник в кармане

Компания «БАРС Груп» разработала бесплатное приложение «Дневник в кармане» для платформы Windows 8, уже сегодня доступное в Магазине Windows. Благодаря ему любой родитель может получить необходимые сведения об успеваемости своего ребёнка с экрана своего мобильного устройства [6].

С помощью этого приложения каждый учащийся, а также его родители получают доступ ко всей информации об учебном процессе: от расписания занятий до домашних заданий и оценок ребёнка. Есть возможность просмотра комментариев, оставленных учителем за работу ученика на уроке. Кроме того, родитель всегда будет знать об отсутствии своего ребенка в школе. Полная функциональность и обновление сведений в режиме онлайн

обеспечивается за счёт информации, собранной в системе «БАРС.Образование-Электронная Школа» (рис. 1.4).

Вывод. Во многих приложениях данной компании можно выделить большое количество интересных идей, интуитивно понятный интерфейс и не раздражающий логотип приложения.

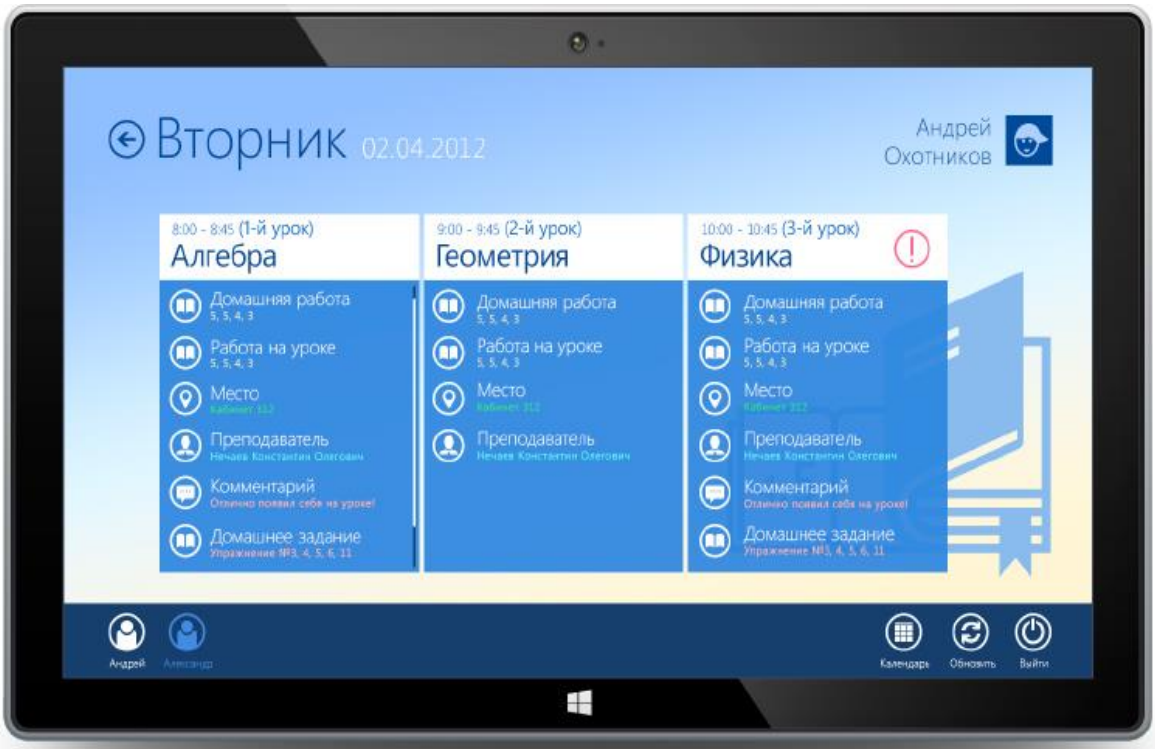


Рис. 1.4. Мобильное приложение «Дневник в кармане»

1.2.3. LMS Школа. Классный журнал

Мобильное рабочее место педагога – это легкий, переносной планшетный компьютер («планшетник»), на котором установлено приложение «Электронный журнал класса» LMS «Школа» [8].

Данная система позволяет автоматизировать работу педагога. С ее помощью преподаватели имеют возможность заносить информацию в школьный электронный журнал, просматривать личное расписание занятий, пользоваться библиотекой цифровых образовательных материалов,

контролировать посещаемость и совершать многие другие полезные операции.

Приложение интегрировано с LMS «Школа». Программа позволяет организовать взаимодействие между школьным электронным журналом и такими же дневниками учеников, а также с преподавателями, администрацией, учащимися и их родителями.

Для успешного функционирования не нужно дополнительных устройств. Все операции с электронным журналом в школе выполняются нажатием пальца по сенсорному экрану планшетного компьютера (рис. 1.5).

Рис. 1.5. Мобильное приложение «LMS Школа»

Вывод. Из данного приложения можно взять идею хранения данных при разрыве соединения с интернетом.

1.3. Модель потоков данных разрабатываемой системы

В мобильном приложении, входными данными от пользователя являются (рис. 1.6):

- данные учетной записи (логин, пароль);
- информация о ведомости;
- значения атрибутов фильтрации;
- команда утвердить ведомость.

Выходными данными к пользователю является:

- список атрибутов фильтрации;
- сведения о ведомости.

При более детальном рассмотрении модели, можно выделить следующие этапы формирования модели: авторизация, модуль выбора, модуль добавления занятия, модуль редактирования занятия (рис. 1.7).

На этапе авторизации, входными данными являются данные учетной записи (логин, пароль), выходными – запрос данных учетной записи.

На этапе выбор параметров фильтрации, входными данными являются значения атрибутов фильтрации, выходными данными являются отправка запроса проверки заданного фильтра, список атрибутов фильтрации.

На этапе добавления ведомости, входными данными являются информация о новой ведомости, команда утвердить ведомость, результат операции. К выходным относится запрос на добавление данных.

На этапе редактирования ведомости, входными данными являются команда утвердить ведомость, результат операции. Выходными данными являются запрос обновления данных.

1.4. Функциональные требования

Ниже описано техническое задание для мобильного приложения, описаны основные требования к программной части, учтены потенциальные роли приложения.

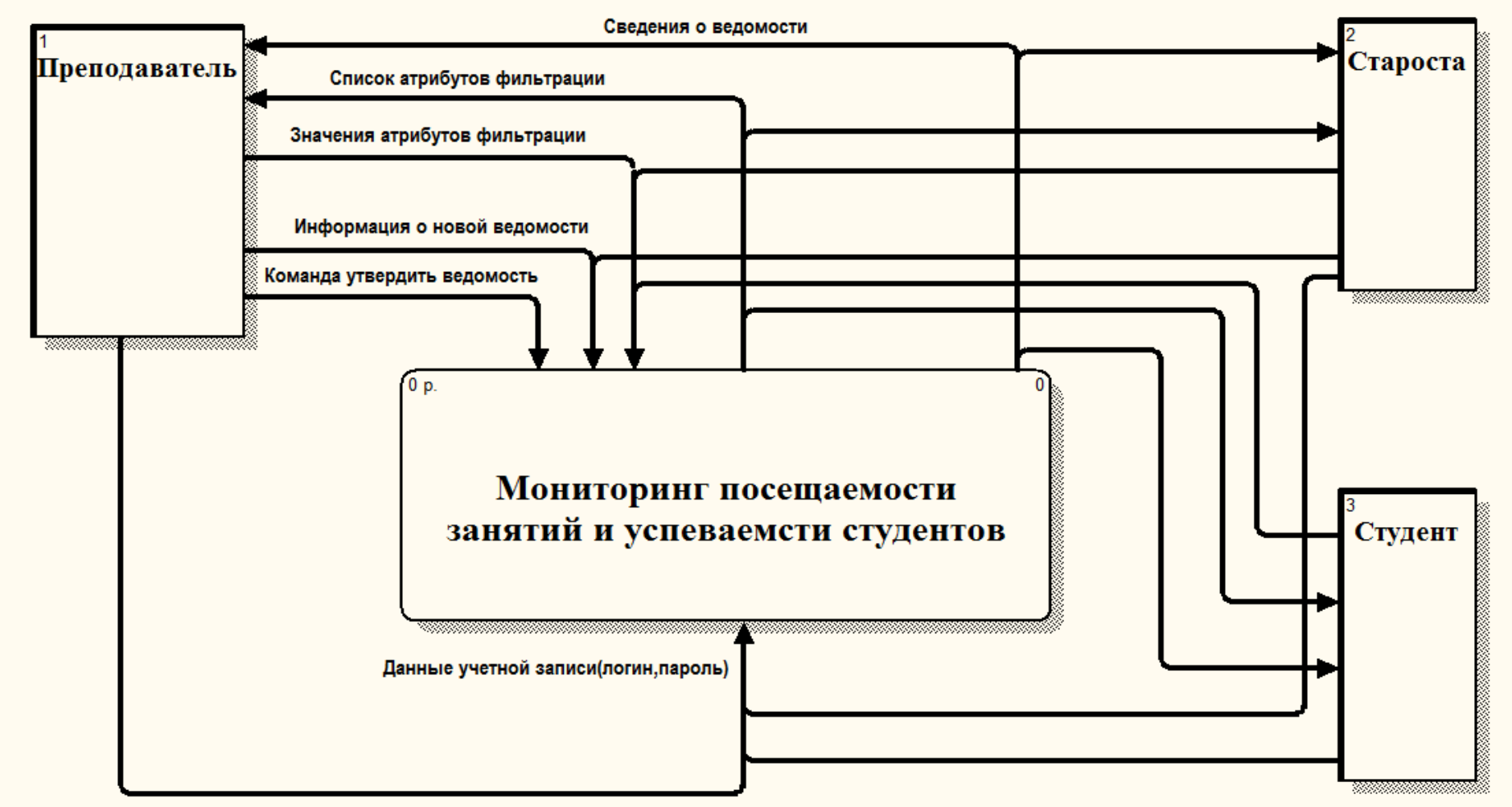


Рис. 1.6. Контекстная диаграмма модели потоков данных

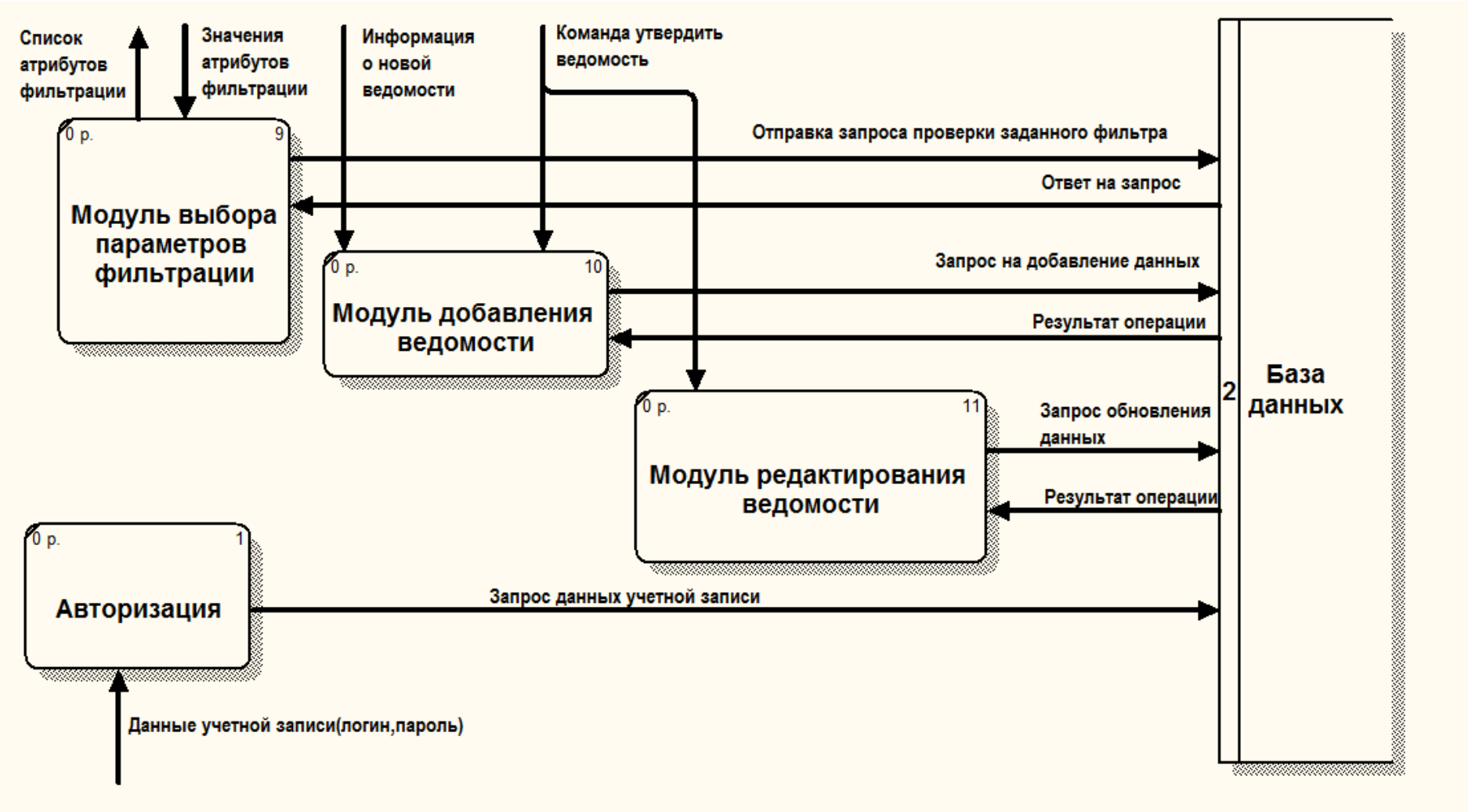


Рис. 1.7. Диаграмма второго уровня

1.4.1. Авторизация и роли

Работать в приложении могут только авторизованные пользователи. Приложение должно поддерживать следующие роли пользователей:

- преподаватель;
- староста;
- студент.

Роль пользователя должна быть определена приложением автоматически по учетным данным пользователя.

1.4.2. Учетные единицы

Единицы учета успеваемости и посещаемости является ведомость. Ведомость имеет следующий атрибуты:

- дата;
- тема;
- группа;
- предмет;
- является утвержденной;
- табличная часть включает следующие поля:
  a) номер по порядку;
  b) ФИО студента;
  c) количество пропусков на данном занятии;
  d) оценка на данном занятии.

1.4.3. Возможности роли «Преподаватель»

Данные параметров фильтрации выводятся в соответствии с нагрузкой преподавателя по учебному плану:

- группы;
- предметы;

- виды занятий.

Преподаватель имеет возможность:

- добавлять ведомость;
- редактировать ведомость;
- утверждать ведомость;
- просматривать ведомость.

1.4.4. Возможности роли «Староста»

Группа определяется автоматически по учетным данным, остальные параметры фильтрации выводятся в соответствии с учебным планом группы:

- предметы;
- виды занятий.

Староста имеет возможность:

- добавлять ведомость;
- редактировать ведомость;
- просматривать ведомость.

1.4.5. Возможности роли «Студент»

Группа определяется автоматически по учетным данным, остальные параметры фильтрации выводятся в соответствии с учебным планом группы:

- предметы;
- виды занятий.

Студент имеет возможность – просматривать ведомость.

1.5. Выводы

На основании приведенного исследования, были сделаны выводы о необходимости написания мобильного приложения, т.к. программы аналоги адаптированы в основном для школы а не для университета.

Целью данной работы является разработка мобильного приложения для автоматизации ведения журнала успеваемости и посещаемости занятий и повышения удобства работы преподавателей.

Следовательно, написано приложение, в котором будут выполняться следующие **задачи:**

1. Проанализировать процесс ведения журнала.
2. Рассмотреть программы аналоги, выявить их сильные и слабые стороны.
3. Спроектировать будущее приложение согласно полученным данным.
4. Реализовать и отладить мобильное приложение.

Объектом является журнал успеваемости и посещаемости. **Предметом** является автоматизация журнала успеваемости и посещаемости.

2. ЭКОНОМИЧЕСКИЙ АНАЛИЗ

2.1. Организационная структура проекта

Организационная структура проекта (OBS) приведена на рис. 2.1.

Рис. 2.1. Организационная структура проекта

2.2. Календарный план проекта

Для оценки расходов на реализацию проекта в числе прочих необходимо определить временные затраты на его реализацию. Для определения временных затрат проекта необходимо разработать календарный план проекта. Реализуемый проект является типовым для Компании, исходя из этого, был сформирован состав работ проекта, определена их длительность, а также распределение ресурсов по ним. При разработке календарного плана были учтены ограничения и допущения, накладываемые на проект Заказчиком.

Для реализации проекта необходимо выполнить следующие работы:

- сбор требований Заказчика к разрабатываемому ПО;
- разработка и согласование технического задания;
- разработка ПО;
- разработка интерфейса ПО;

- тестирование ПО.

Распределение человеческих ресурсов по работам проекта и степень их загрузки приведены в табл. 2.1.

Таблица 2.1

Структура общего времени на создание программного продукта

№ этапа	Этап работ	Ответственные исполнители (занятость на этапе)	Длительность, дней
1	Сбор требований Заказчика к разрабатываемому ПО	Руководитель проекта 100%	5
2	Разработка и согласование технического задания	Руководитель проекта 100% Архитектор 50%	4
3	Разработка ПО	Архитектор 100% Разработчик 100% Руководитель проекта 40%	20
4	Разработка интерфейса ПО	Дизайнер интерфейса 100%	10
5	Тестирование ПО	Специалист по тестированию ПО 100% Архитектор 20% Руководитель проекта 40%	10

При реализации данного проекта работы выполняются последовательно. Диаграмма Ганта [11] приведена на рис. 2.2-2.3.

		Режим задачи	Название задачи	Длительнс	Начало	Окончани	Названия ресурсов
1			Сбор требований Заказчика к разрабатываемому ПО	5 дней	Вт 15.04.14	Пн 21.04.14	Руководитель проекта
2			Разработка и согласование технического задания	4 дней	Вт 22.04.14	Пт 25.04.14	Архитектор[50%]; Руководитель проекта
3			Разработка ПО	20 дней	Сб 26.04.14	Чт 22.05.14	Разработчик;Руководитель проекта[40%]
4			Разработка интерфейса ПО	10 дней	Пн 19.05.14	Пт 30.05.14	Дизайнер
5			Тестирование ПО	10 дней	Вт 20.05.14	Пн 02.06.14	Специалист по тестированию ПО;Арх

Рис. 2.2. Табличное представление Диаграммы Ганта

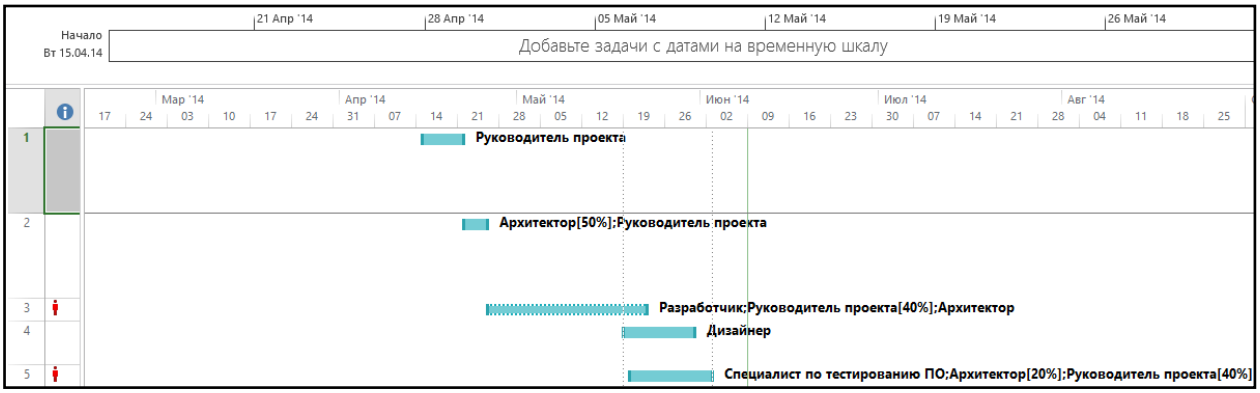


Рис. 2.3. Графическое представление Диаграммы Ганта

Исходя из длительности работ и коэффициента загрузки членов проектной команды, определим их трудозатраты в табл. 2.2.

Таблица 2.2

Трудозатраты членов проектной команды

№	Исполнитель	Трудозатраты, человеко-часов
1	Руководитель проекта	136
2	Архитектор	192
3	Разработчик	160
4	Дизайнер интерфейсов	80
5	Специалист по тестированию ПО	80

2.3. Расчёт затрат на разработку продукта

Расчет затрат на создание и внедрение ПО включает следующие составляющие с последующим их графическим представлением в виде круговой диаграммы:

- заработная плата исполнителей работ по проекту – $ЗП_{осн}$;
- отчисления на социальные нужды (страховые взносы) – $Н_{зп}$;
- арендные платежи за производственные (офисные) помещения – $А_{пм}$;
- амортизация используемых основных средств и нематериальных активов – $А$;
- расходы на модернизацию и приобретение основных средств – $Р_{мод}$;
- расходы на приобретение необходимого ПО – $Р_{ПО}$;

- расходы на интернет, связь – $Р_{тел}$;
- расходы на канцелярские товары и расходные материалы – $Р_{р.м.}$;
- прочие расходы – $П_{р.р.}$.

2.3.1. *Расчёт заработной платы исполнителей работ по созданию программного продукта*

Основная ЗП определяется по формуле:

$$ЗП_{осн} = \frac{М \cdot Т}{Ч_р \cdot t_{р.д.}}\left(1+\frac{П}{100}\right), руб, \qquad (2.1)$$

где $М$ – месячная зарплата (*руб.*), $Т$ – общие трудозатраты (*чел.-ч*), $Ч_р$ – число рабочих дней в месяц, $t_{р.д.}$ – продолжительность рабочего дня в часах, $П$ – процент премии. В данной работе $Ч_р = 21$ *день*, $t_{р.д.}=8ч$, $П=0$.

Значение месячной заработной платы (М), суммарные трудозатраты членов, а также рассчитанная по формуле 2.1 основная заработная плата проектной команды приведены в табл. 2.3.

Таблица 2.3

Основная заработная плата членов проектной команды

№	Исполнитель	Месячная заработная плата (М), руб.	Трудозатраты, человеко-часов	$ЗП_{осн}$, руб.
1	Руководитель проекта	40 000	136	32 380
2	Архитектор	30 000	192	34 285
3	Разработчик	25 000	160	23 809
4	Дизайнер интерфейса	20 000	80	9 523
5	Специалист по тестированию ПО	20 000	80	9 523

Общая заработная плата составит 109 523 (руб.).

Проект реализуется в небольшой ИТ-компании, где доля вспомогательного и административного персонала по отношению к основному персоналу не велика. Большая часть административного персонала задействована в проектной деятельности в качестве руководителей

проекта. Кадровый учет, бухгалтерский и налоговый учет в Компании отдан на аутсорсинг. Затраты на аутсорсинг войдут в прочие расходы. В связи с этим примем заработную плату обслуживающего персонала равной 0 руб.

2.3.2. Расчёт отчислений на социальные нужды

Теперь можно рассчитать величину отчислений на социальные нужды (страховые взносы) [17], которые начисляются на заработную плату и в 2014 г. для организаций, осуществляющих деятельность в области информационных технологий, составляют 14% по выплатам в пределах 568 тыс. руб. Структура отчислений на социальные нужды приведена в табл. 2.4.

Таблица 2.4

Структура отчислений на социальные нужды (страховые взносы)

Пенсионный фонд Российской Федерации	8,0%
для лиц 1966 года рождения и старше	
страховые взносы на страховую часть трудовой пенсии	8,0%
для лиц 1967 года рождения и моложе	
страховые взносы на страховую часть трудовой пенсии	2,0%
страховые взносы на накопительную часть трудовой пенсии	6,0%
Фонд социального страхования Российской Федерации	2,0%
Федеральный фонд обязательного медицинского страхования	4,0%

Таким образом, $H_{зп}$= 24 095 (*руб.*).

2.3.3. Арендные платежи за производственные помещения

Компания, реализующая проект по разработке и внедрению ПО для автоматизации внутри складской логистики, арендует офисные помещения в г. Брянск.

Стоимость аренды составляет 630 руб/м^2 в месяц. Арендная плата включает в себя оплату как площади занимаемых Компанией помещений, так и электроэнергии, отопления, водоснабжения, кондиционирования и уборки помещений, вывоза и утилизации технико-бытовых отходов, парковочных мест на автостоянке.

На каждого члена проектной команды приходится 4,5 $м^2$ арендуемого офисного помещения. На период данного проекта члены проектной команды в других проектах не задействованы.

Исходя из изложенного выше, затраты на аренду помещений, отнесенные на проект составят $A_{пм}$ = 21 262 (руб.).

2.3.4. Амортизация используемых основных средств и нематериальных активов

При реализации проекта по разработке и внедрению ПО для автоматизации внутри складской логистики задействованы следующие основные средства:

- 4 персональных компьютеров в сборе первоначальной стоимостью 20 000 (руб.) каждый;
- 1 ноутбук первоначальной стоимостью 25 000 (руб.).

Срок полезного использования для задействованных в проекте основных средств определен в 3 года. Метод начисления амортизации – линейный.

Амортизационные отчисления для персонального компьютера на 1 месяц составят 20 000 / 36 = 555,56 (руб.). Амортизационные отчисления для ноутбука на 1 месяц составят 25 000 / 36 = 694,44 (руб.).

Амортизационные отчисления по ОС, относящиеся на проект составят:

$$A_{ос} = 4 \cdot 1{,}5 \cdot 555{,}56 + 694{,}44 \cdot 1{,}5 = 6041{,}7 \text{ (руб.)}.$$

В качестве ОС используется свободно распространяемое ПО Linux. В качестве сервера БД используется свободно распространяемое ПО – MySQL. В качестве среды для разработки используется свободно распространяемое ПО – IntelliJ IDEA Community Edition. В качестве проектирования интерфейса используется свободное ПО – Gimp.

Суммарные амортизационные отчисления составят: A=6041,7руб.

2.3.5. Расходы на модернизацию и приобретение основных средств

При реализации проекта по разработке и внедрению ПО для автоматизации внутри складской логистики не планируется приобретение новых и модернизация существующих основных средств.

2.3.6. Расходы на приобретение необходимого ПО

При реализации проекта не планируется приобретение ПО.

2.3.7. Расходы на интернет и связь

Так как в Компании, реализующей проект не производится биллинг и тарификация телекоммуникационных услуг в разрезе сотрудников, затраты на интернет и связь войдут в прочие затраты, рассчитываемые как процент от прямых затрат.

2.3.8. Расходы на канцелярские товары и расходные материалы

Затраты на расходные материалы берутся по факту и составляют $Р_{р.м.}$= 3 500 (*руб.*). К данным затратам относятся затраты на канцтовары, тонер и бумагу для принтера и т.д.

2.3.9. Прочие расходы

Прочие расходы составляют 20% от суммы следующих элементов структуры затрат: $ЗП_{осн}$, $ЗП_{доп}$, $Н_{зп}$, $А_{пм}$, $А$, $Р_{мод}$, $Р_{ПО}$, $Р_{тел}$ и $Р_{р.м.}$.

$$П_{р.р.} = 0.2(ЗП_{осн} + ЗП_{доп} + Н_{зп} + А_{пм} + А + Р_{мод} + Р_{ПО} + Р_{тел} + Р_{р.м.}) \quad (2.2)$$

Таким образом, $П_{р.р.}$= 32 884(*руб.*).

2.3.10. Расчёт себестоимости программного продукта

В себестоимость программного продукта входят следующие элементы: $ЗП_{осн}$, $ЗП_{доп}$, $Н_{зп}$, $А_{пм}$, $А$, $Р_{мод}$, $Р_{ПО}$, $Р_{тел}$, $Р_{р.м.}$ и $П_{р.р.}$.

Сложив все элементы, можно определить себестоимость программного продукта и услуг по его внедрению: $C_{п.п.}$= 197 306 (*руб.*).

Структура себестоимости программного продукта отражена в табл. 2.5 и представлена на рис. 2.4.

Таблица 2.5

Структура себестоимости программного продукта

№	Элементы себестоимости	Сумма (руб.)	% в общ. сумме себестоимости
1	Основная заработная плата исполнителя	109523	55
2	Отчисления на социальные нужды (страховые взносы)	24095	12
3	Арендные платежи за производственные (офисные) помещения	21262	11
4	Амортизация используемых основных средств и нематериальных активов	6041,7	3
5	Расходы на модернизацию и приобретение основных средств	–	–
6	Расходы на приобретение необходимого ПО	–	–
7	Расходы на интернет, связь	–	–
8	Расходы на канцелярские товары и расходные материалы	3 500	2
9	Прочие расходы	32884,34	17
	Итого:	**197 306**	**100**

Рис. 2.4. Структура себестоимости программного продукта

3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

3.1. Архитектура программного продукта

Фаза архитектурного проектирования – фаза «принятия решения». Цель этой фазы – определить совокупность компонент программного изделия и их интерфейсы, чтобы дать каркас для последующей разработки программного изделия. Архитектурный проект должен охватывать все требования, сформулированные на предыдущей фазе – системный анализ.

На рис. 3.1 приведена архитектурная схема приложения «Мобильное приложение мониторинга посещаемости занятий и успеваемости студентов». Для работы системы реализовано мобильное приложение и серверная часть. Серверная часть предоставляет программный интерфейс и отвечает за сохранение данных и предоставление их по запросам мобильного приложения.

Мобильное приложение обеспечивает интерфейс пользователя и включает в себя модуль авторизации, модуль связи с серверной частью, а также три интерфейса в которые в зависимости от полномочий роли входят:

- модуль выбора параметров фильтра;
- модуль добавления ведомости;
- модуль редактирования ведомости;
- модуль просмотра ведомости.

В качестве среды для разработки мобильного приложения была выбрана **IntelliJ IDEA Community Edition 13.0.1**. Версия среды разработки Community Edition 13.0.1 распространяется бесплатно, имеет хорошую техническую поддержку. IntelliJ IDEA [9] позволяет быстрее и качественнее писать и менять код, подсказывает подходящие имена, находит подходящие методы. Не требует точно выделить выражение, а по тому, где вы находитесь, угадывает, что вы хотели сделать и как хотели это назвать. Платформа IntelliJ,

разработанная JetBrains, стала основой для нового продукта Google – Android Studio [5].

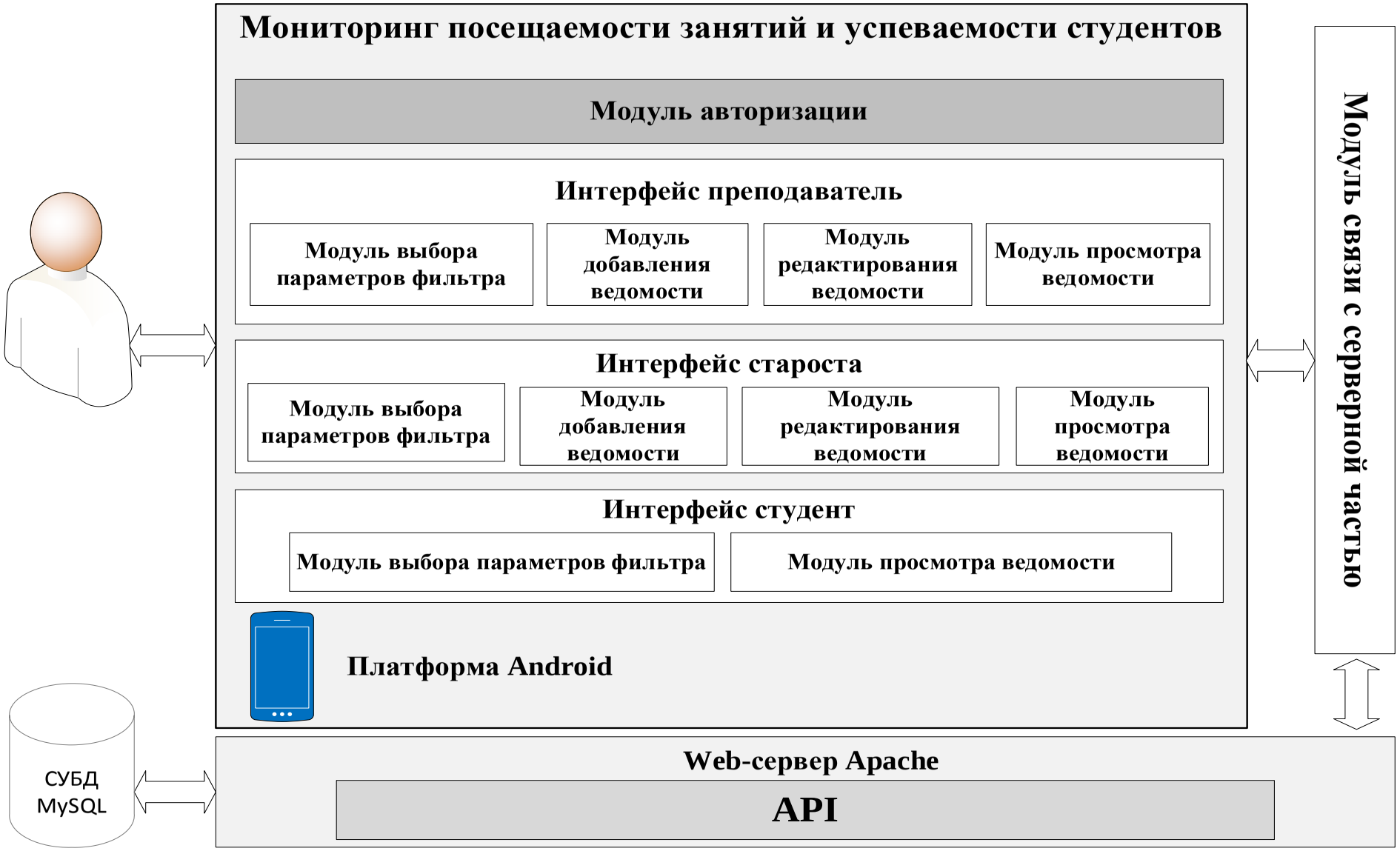


Рис. 3.1. Архитектурная схема приложения

3.2. Модель данных

В разделе рассматривается созданная база данных [15] для мобильного приложения «Мониторинг посещаемости занятий и успеваемости студентов».

Для мобильного приложения используется база данных, импортированная в MySQL [20]. Для управления базой данных используется приложение phpMyAdmin [12], включенное в состав хостинга, на котором размещена база данных. База данных мобильного приложения приведена к 4 нормальной форме и состоит из 8 связанных между собой отношений.

- Отношение «student» служит для хранения информации о студенте.
- Отношение «group» хранит название существующих групп.
- Отношение «prepodavatel» служит для хранения информации о преподавателе.
- Отношение «ucheb_plan» создана для хранения учебного плана.
- Отношение «vid_zanytiya» создана для хранения привязки вида занятия.
- Отношение «predmet» создана для хранения данных о предмете.
- Отношение «posetchemost» создана для хранения данных о посещаемости занятий студентами.
- Отношение «uspevaemost» создана для хранения успеваемости студентов.

База данных для приложения была спроектирована в ERwin Data [21]. На рис. 3.2 приведен логический уровень базы данных, который позволяет лучше понять, какие данные будут храниться.

Далее описан физический уровень модели рис. 3.3. Физический уровень показывает то состояние базы данных в котором она будет размещена. На рис. 3.4 изображена модель базы данных выгруженная в СУБД MySQL. Эта модель базы данных с которой работает приложение.

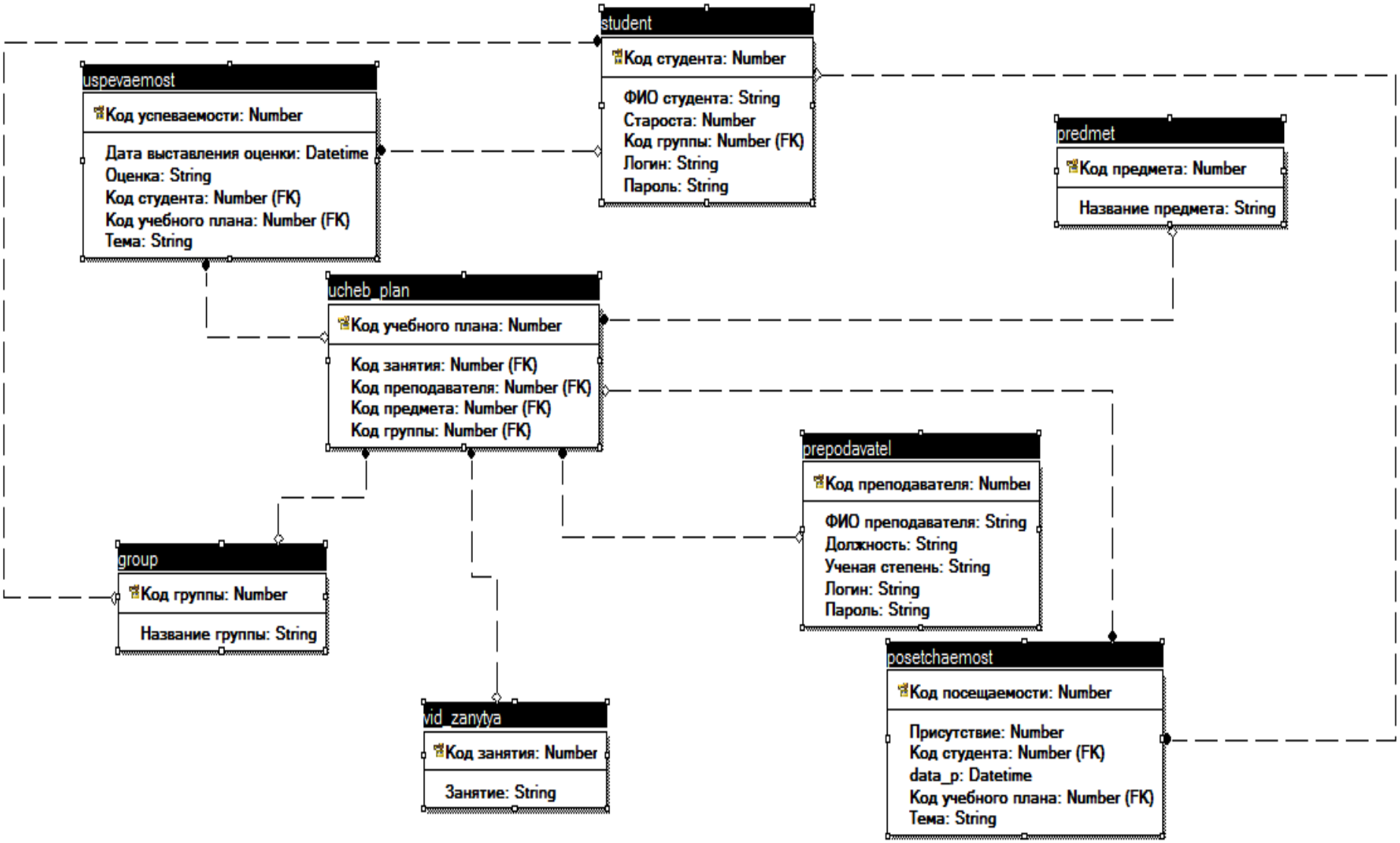


Рис. 3.2. Логический уровень базы данных

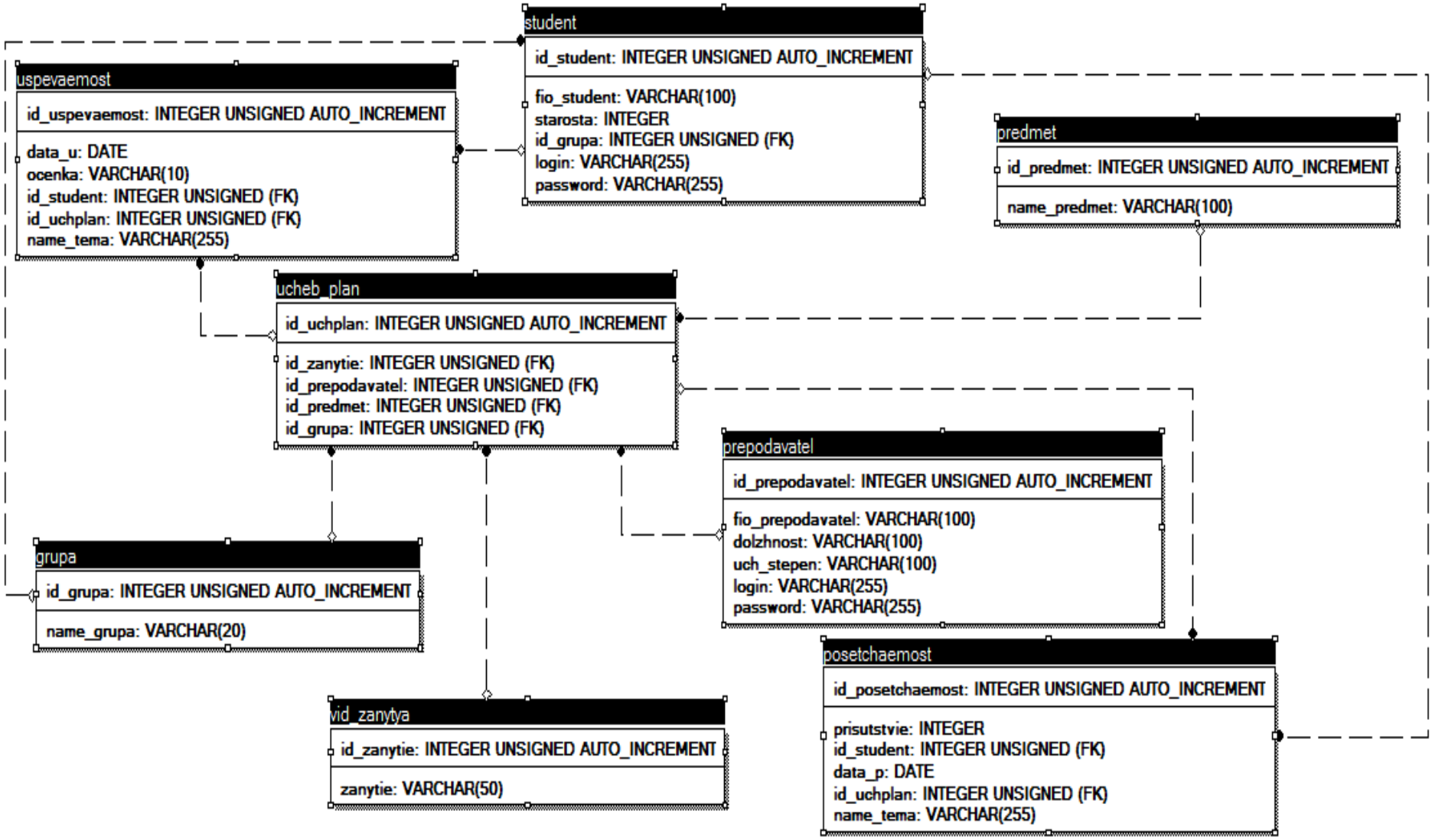


Рис. 3.3. Физический уровень модели базы данных

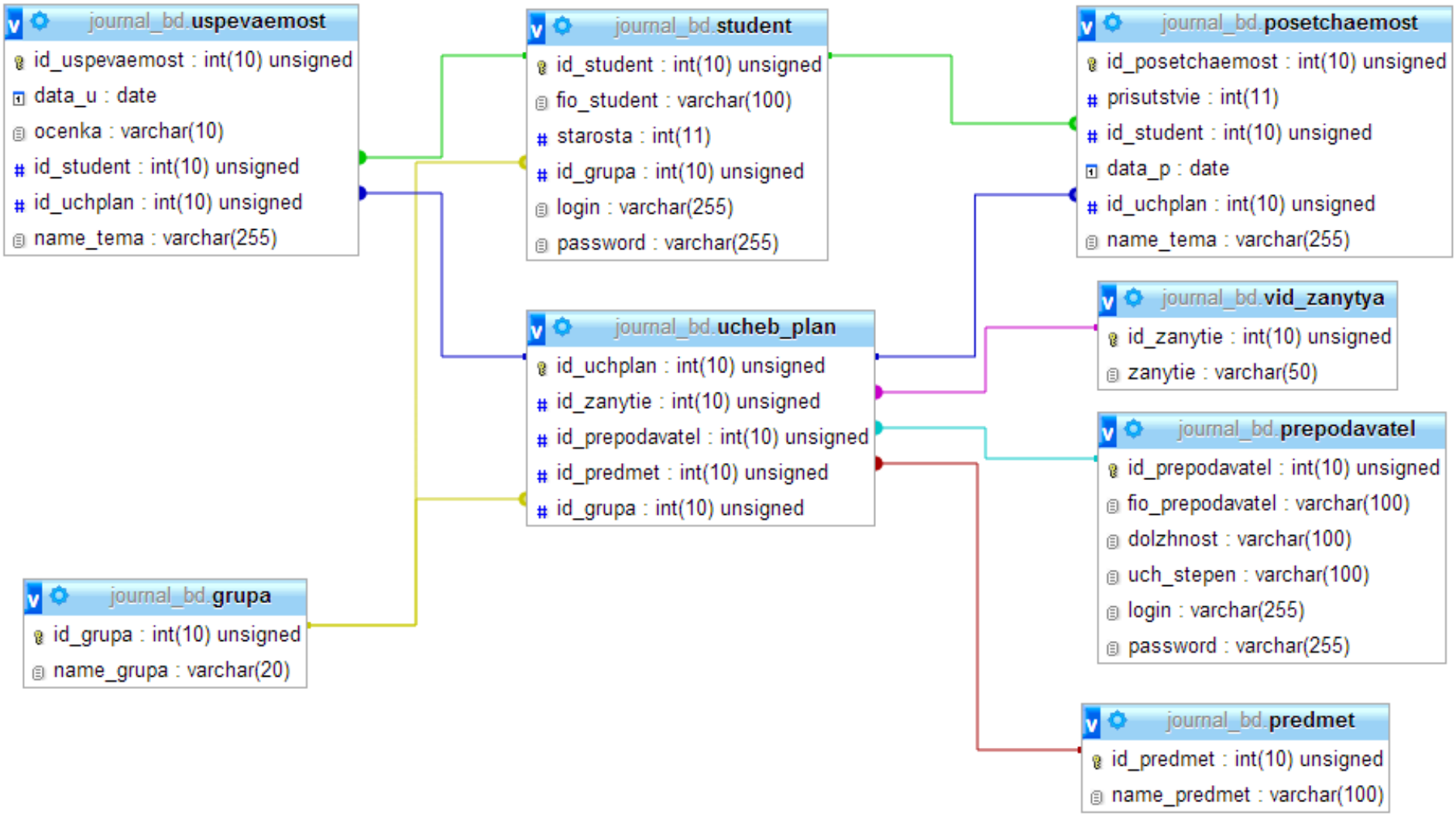


Рис. 3.4. Модель базы данных, выгруженная в СУБД MySQL

3.3. Низкоуровневое проектирование

На приведенной ниже диаграмме [13] описываются функции, доступные каждому из ролей пользователей мобильного приложения рис. 3.5.

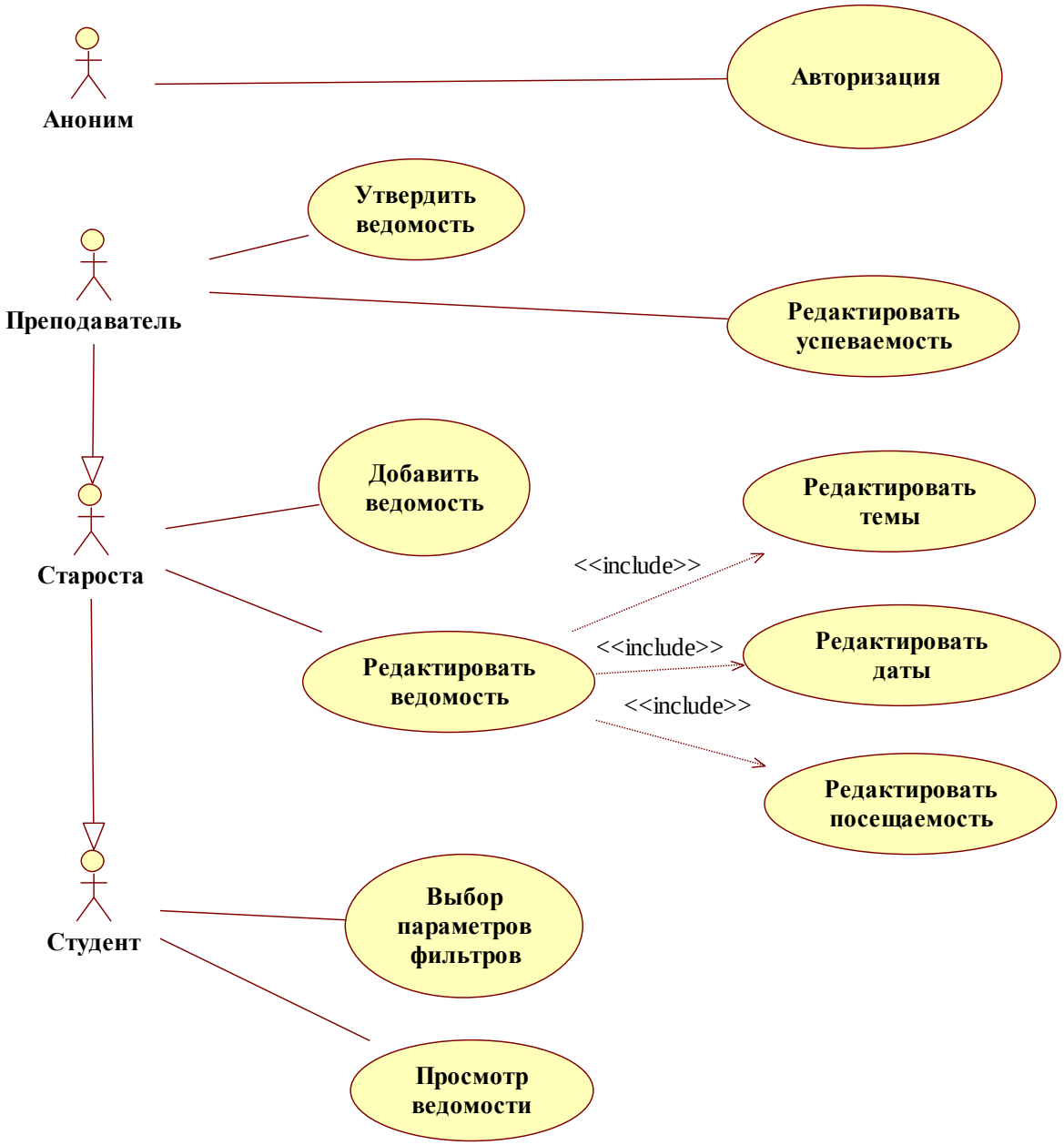


Рис. 3.5. Диаграмма вариантов использования

На диаграмме классов изображены классы и функции, доступные каждому из видов пользователей рис. 3.6.

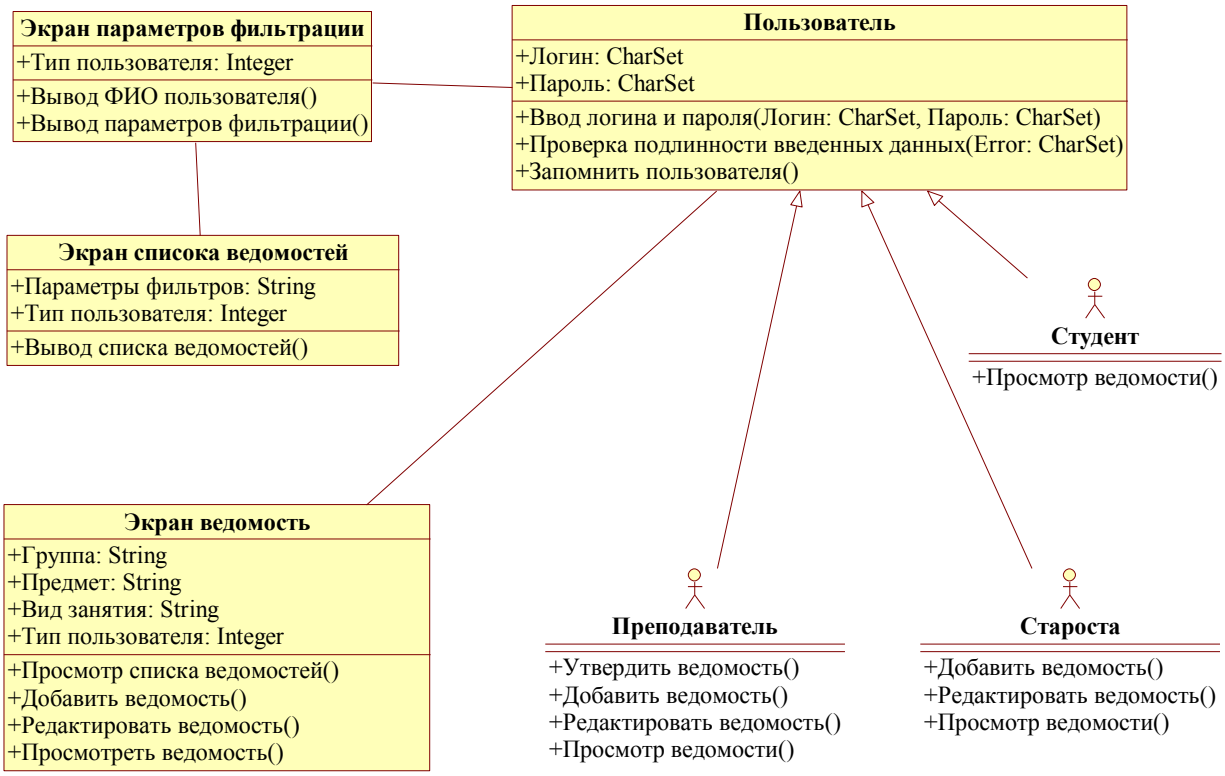


Рис. 3.6. Диаграмма классов

На приведенном ниже рис. 3.7 рассматривается процесс добавления ведомости преподавателем.

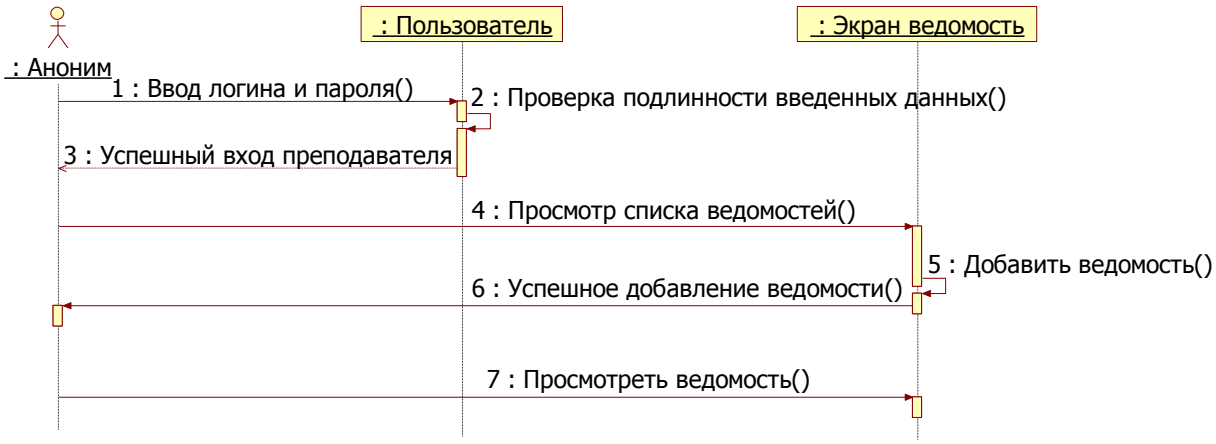


Рис. 3.7. Диаграмма последовательности

При успешном входе в программу преподавателем, пользователь может добавить ведомость. Для этого необходимо перейти к списку ведомостей. И добавить ведомость, в случае успеха, приложение уведомляет о успешном

добавлении. После добавления, ведомость появляется в списке ведомостей и ее можно просмотреть.

3.4. Проектирование интерфейсов

В разделе рассматриваются предварительные наброски пользовательского интерфейса мобильного приложения [14].

На рис. 3.8 изображен интерфейс экрана авторизации пользователя.

Рис. 3.8. Интерфейса экрана авторизации

Основой служит LinearLayout, который задает расположение элементов на экране мобильного устройства, компонент ScrollView изначально не виден, но он необходим при малом разрешении экрана. Компонент TextView служит для вывода статического текста, компоненты EditText и Button служат ввода и отправки введенных данных [1].

3.4.1. Проектирование интерфейса преподавателя

На рис. 3.9 приведены интерфейсы преподавателя. На первом экране слева на право в качестве разметки используется RelativeLayout.

Присутствует также компонент ScrollView. В TextView отображает ФИО, компоненты Spinner выводит список доступных для выбора данных.

На втором экране изображен интерфейс, который отображает список ведомостей. Каждый элемент списка содержит тему, дату, вид занятия и индикатор, который отображает утверждена ведомость или нет. В качестве разметки используется RelativeLayout. Для отображения списка используется компонент ListView. Чтобы добавить ведомость, используется компонент Button.

Рис. 3.9. Интерфейсы для преподавателя

На третьем экране приведен интерфейс добавления ведомости. В качестве разметки используется TableLayout, для вывода строк таблицы используется компонент TableRow. Компоненты TextView используются для вывода группы и предмета. Компонент EditText служит для ввода темы занятия, Компонент CheckBox используются для утверждения ведомости. Компонент Button используется для добавления колонки с датой и ячейками.

На четвертом экране приведен интерфейс редактирования ведомости. В качестве разметки используется TableLayout, для вывода строк таблицы используется компонент TableRow. Компоненты TextView используются для

вывода группы и предмета. Компонент EditText служит для изменения темы занятия. Компонент CheckBox используются для утверждения ведомости.

Интерфейс экрана просмотра ведомости не приведён, так как он идентичен экрану редактирования ведомости.

3.4.2. Проектирование интерфейса старосты

На рис. 3.10 приведены интерфейсы старосты. На первом экране слева на право в качестве разметки используется RelativeLayout.

Рис. 3.10. Интерфейсы для старосты

Присутствует также компонент ScrollView. В TextView отображает ФИО, компоненты Spinner выводит список доступных для выбора данных.

На втором экране изображен интерфейс, который отображает список ведомостей. Каждый элемент списка содержит тему, дату, вид занятия и индикатор, который отображает утверждена ведомость или нет. В качестве разметки используется RelativeLayout. Для отображения списка используется компонент ListView. Чтобы добавить ведомость, используется компонент Button.

На третьем экране приведен интерфейс добавления ведомости. В качестве разметки используется TableLayout, для вывода строк таблицы используется компонент TableRow. Компоненты TextView используются для вывода группы и предмета. Компонент EditText служит для ввода темы занятия. Компонент Button используется для добавления колонки с датой и ячейками.

На четвертом экране приведен интерфейс редактирования ведомости. В качестве разметки используется TableLayout, для вывода строк таблицы используется компонент TableRow. Компоненты TextView используются для вывода группы и предмета. Компонент EditText служит для изменения темы занятия.

Интерфейс экрана просмотра ведомости не приведён, так как он идентичен экрану редактирования ведомости.

3.4.3. Проектирование интерфейса студента

На рис. 3.11 приведены интерфейсы старосты. На первом экране слева на право в качестве разметки используется RelativeLayout. Присутствует также компонент ScrollView. В TextView отображает ФИО, компоненты Spinner выводит список доступных для выбора данных.

На втором экране изображен интерфейс, который отображает список ведомостей. Каждый элемент списка содержит тему, дату, вид занятия и индикатор, который отображает утверждена ведомость или нет. В качестве разметки используется RelativeLayout. Для отображения списка используется компонент ListView. Чтобы добавить ведомость, используется компонент Button.

На третьем экране приведен интерфейс просмотра ведомости. В качестве разметки используется TableLayout, для вывода строк таблицы используется компонент TableRow.

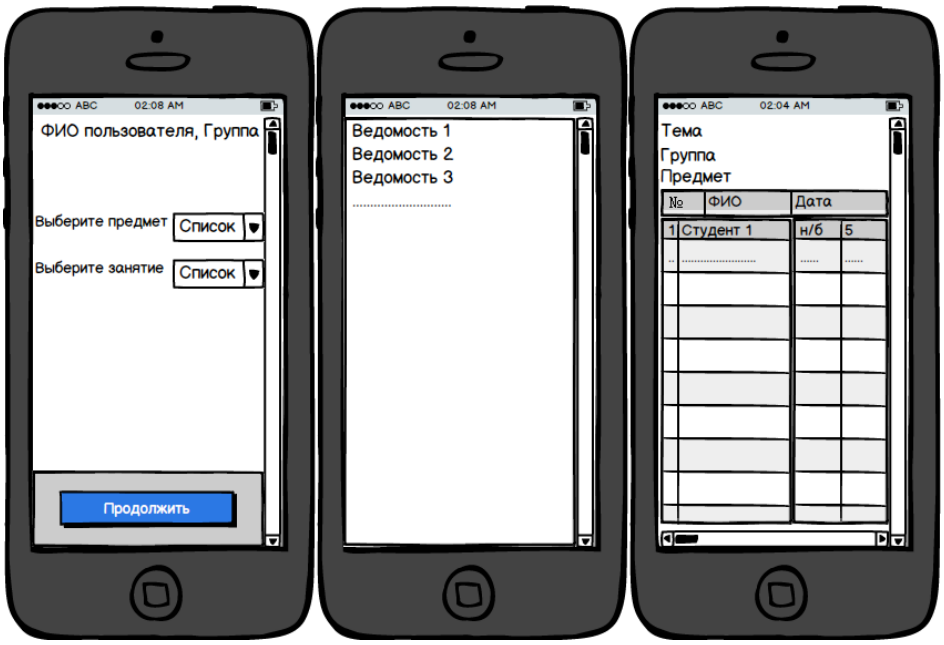


Рис. 3.11. Интерфейсы для студента

Компоненты TextView используются для вывода темы, группы и предмета.

3.5. Реализация модулей приложения

3.5.1. Модуль авторизации

В листинге 3.1 приведено описание основной части работы модуля авторизации. Полученный ответ с сервера в виде JSON объекта нужно преобразовать, затем извлечь из него данные. Если пользователь с введенными данными найден в базе данных, то JSON объект хранит в себе флаги, и ФИО пользователя. Если нет, то хранит сообщение об ошибке.

3.5.2. Модуль вывода параметров фильтрации

В листинге 3.2 описана работа с выводом полученных данных с сервера в список. Для каждого списка создается адаптер, в который помещается массив с данными и xml разметкой. После выводим данные на экран. Каждый список содержит обработчик нажатия на элемент списка.

Листинг. 3.1

Авторизация

```
//полученные данные преобразуем в объект
JSONObject json = new JSONObject(response);
flag = json.getInt(TAG_RESULT);
JSONArray urls = json.getJSONArray("prepod");
for (int i = 0; i < urls.length(); i++) {
    prepod = urls.getJSONObject(i).getString("name").toString();
    star = urls.getJSONObject(i).getInt("star");
}
if(flag == 1){
    Intent intent = new Intent(MainActivity.this, PrepodActivity.class);
    intent.putExtra("login", prepodavatel);
    startActivity(intent);
    finish();
}
if(flag == 2 && star == 1){
    Intent intent = new Intent(MainActivity.this,StarostaActivity.class);
    intent.putExtra("login", starosta);
    startActivity(intent);
    finish();
}
if(flag == 2 && star == 0){
    Intent intent = new Intent(MainActivity.this, StudentActivity.class);
    intent.putExtra("login", student);
    startActivity(intent);
    finish();
}
```

Листинг. 3.2

Вывод списков параметров фильтрации

```
ArrayAdapter<String> adapter_groups = new
ArrayAdapter<String>(ChangePrepodavatelActivity.this,
android.R.layout.simple_spinner_item, groupArray);
adapter_groups.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item);

spinner_group = (Spinner)findViewById(R.id.spinner_groups);
spinner_group.setAdapter(adapter_groups);
// заголовок
spinner_group.setPrompt("Группы");
spinner_group.setOnItemSelectedListener(new OnItemSelectedListener() {
@Override
public void onItemSelected(AdapterView<?> parent, View view, int position,
long id) {
// показываем позиция нажатого элемента
((TextView)parent.getChildAt(0)).setTextColor(getResources().getColor(R.color
.font_style));
//Toast.makeText(getBaseContext(), "Position = " +
spinner_group.getSelectedItem().toString(), Toast.LENGTH_SHORT).show();
name_group = spinner_group.getSelectedItem().toString();
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
}
});
```

3.5.3. Модуль добавления ведомости

В листинге 3.3 описывается добавление ведомости путем формирования JSON объекта и отправки его на сервер, для дальнейшей обработки. Здесь же получаем ответ от сервера о успешном или неуспешном добавлении.

Листинг. 3.3

Отправка заполненных данных на сервер

```
//создаем запрос на сервер
DefaultHttpClient hc = new DefaultHttpClient();
ResponseHandler<String> res = new BasicResponseHandler();
HttpPost postMethod = new HttpPost(params[0]);
//будем передавать 1 параметр
List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(1);
String sendJson = "";
JSONObject jsonObject = new JSONObject();
JSONArray jsonID_STUDENT = new JSONArray();
JSONArray jsonPRISUTSTVIE = new JSONArray();
JSONArray jsonUSPEVAEMOST = new JSONArray();
for(int i = 0; i <studentArray.size(); i++)
{
    jsonID_STUDENT.put(id_studentArray.get(i));
    jsonPRISUTSTVIE.put(GetColumPrisutstvieTable(i));
    jsonUSPEVAEMOST.put(GetColumUspevaemostTable(i));
}
jsonObject.put("jsonID_STUDENT",jsonID_STUDENT);
jsonObject.put("jsonPRISUTSTVIE",jsonPRISUTSTVIE);
jsonObject.put("jsonUSPEVAEMOST",jsonUSPEVAEMOST);
jsonObject.put("sendDate",sendDate);
jsonObject.put("id_uchplan", id_ucheb_plan);
jsonObject.put("sendThema",TextThema.getText().toString());
jsonObject.put("utverdit", utverdit.toString());
sendJson = jsonObject.toString();
nameValuePairs.add(new BasicNameValuePair("sendJson", sendJson));
//соберaем данные вместе и посылаем на сервер
postMethod.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));
//получаем ответ от сервера
String response = hc.execute(postMethod,res);
JSONObject json = new JSONObject(response);
```

3.5.4. Модуль редактирования ведомости

В листинге 3.4 описана отправка отредактированных данных на сервер. Данные собираются из ячеек таблицы и записываются в JSON объект, который передается на сервер для дальнейшей обработки. Отправка данных похожа с отправкой данных при добавлении, но здесь еще необходимо передать те данные, которые были выведены на момент редактирования.

3.5.5. Модуль просмотра ведомости

В листинге 3.5 приведен пример функции создания таблицы, в которую буду выведены загруженные данные.

Листинг. 3.4

Отправка отредактированных данных на сервер

```
//создаем запрос на сервер
DefaultHttpClient hc = new DefaultHttpClient();
ResponseHandler<String> res = new BasicResponseHandler();
HttpPost postMethod = new HttpPost(params[0]);
//будем передавать два параметра
List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(1);
String sendJson = "";
JSONObject jsonObject = new JSONObject();
JSONArray jsonID_STUDENT = new JSONArray();
JSONArray jsonPRISUTSTVIE = new JSONArray();
JSONArray jsonUSPEVAEMOST = new JSONArray();
for(int i = 0; i <studentArray.size(); i++)
{
    jsonID_STUDENT.put(id_studentArray.get(i));
    jsonPRISUTSTVIE.put(GetColumPrisutstvieTable(i));
    jsonUSPEVAEMOST.put(GetColumUspevaemostTable(i));
}
jsonObject.put("jsonID_STUDENT",jsonID_STUDENT);
jsonObject.put("jsonPRISUTSTVIE",jsonPRISUTSTVIE);
jsonObject.put("jsonUSPEVAEMOST",jsonUSPEVAEMOST);
jsonObject.put("sendDate",sendDate);
jsonObject.put("sendStartDate", view_data);
jsonObject.put("id_uchplan", id_ucheb_plan);
jsonObject.put("sendThema",TextThema.getText().toString());
jsonObject.put("sendStartThema", view_thema);
jsonObject.put("utverdit", utverdit.toString());
jsonObject.put("start_utverdit", getIntent().getStringExtra("utverzhdeno"));
sendJson = jsonObject.toString();
nameValuePairs.add(new BasicNameValuePair("sendJson", sendJson));
//собераем их вместе и посылаем на сервер
postMethod.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));
```

3.5.6. Серверная часть

Серверная часть обеспечивает связь мобильного приложения и базы данных, обеспечивая интерфейс прикладного программиста (Application programmer interface, API), в который включены следующие основные методы.

1. Методы чтения данных:

- получить список групп;
- получить список предметов;
- получить список занятий;
- получить список ведомостей;
- получить табличные данные.

2. Методы модификации данных – обновить сведения о ведомости.

Листинг. 3.5

Вывод данных в таблицу

```
public void CreateTable(final ArrayList studentArray)
{
   tableLayout = (TableLayout) findViewById(R.id.table);
   tableLayout.setStretchAllColumns(true);
   TextThema = (TextView) findViewById(R.id.TextThema);
   TextThema.setText(view_thema);
   TextView TextPredmet = (TextView) findViewById(R.id.TextPredmet);
   TextPredmet.setText("Предмет: "+name_predmet);
   TextView TextGrupa = (TextView) findViewById(R.id.TextGrupa);
   TextGrupa.setText("Группа: "+name_group);
   TextView FIO = new TextView(this);
   FIO.setText("ФИО");
   FIO.setWidth(500);
   FIO.setBackgroundResource(android.R.drawable.editbox_background_normal);
   FIO.setPadding(10,10,10,10);
   current_date = new TextView(this);
   current_date.setText(getIntent().getStringExtra("view_data"));
   current_date.setGravity(Gravity.CENTER_HORIZONTAL);
   current_date.setClickable(true);
   current_date.setBackgroundResource(R.drawable.editbox_background);
   current_date.setPadding(10,10,10,10);
   current_date.setOnClickListener(add_current_data);
   fio_data = new TableRow(this);
   fio_data.setBackgroundColor(0xffcccccc);
   fio_data.addView(FIO);
   fio_data.addView(current_date);
   TableRow.LayoutParams params = (TableRow.LayoutParams)
   current_date.getLayoutParams();
   params.span = 2;
   current_date.setLayoutParams(params);
   tableLayout.addView(fio_data);
   for(int i=0;i<studentArray.size();i++)
   {
       TextView txt = new TextView(this);
       txt.setText(studentArray.get(i).toString());
       txt.setBackgroundResource(android.R.drawable.editbox_background);
       txt.setPadding(10,10,10,10);
       rowS = new TableRow(this);
       rowS.setId(100+i);
       rowS.addView(txt);
       tableLayout.addView(rowS);
   }
   for(int i=0;i<studentArray.size();i++)
   {
       AddColumPrisutstvieTable(i);
   }
```

```
for(int i=0;i<studentArray.size();i++){
    AddColumUspevaemostTable(i);
}
```

Серверная часть состоит и PHP скриптов. В листинге 3.6 показан PHP скрипт, который вернет приложению JSON объект в который будут записаны группы, предметы, виды занятий в соответствии с учебным планом пользователя. Работа скрипта заключается в том, что он принимает данные оправленные POST методом и формирует запрос в базу данных.

Листинг. 3.6

Обработка и отправка данных в соответствии с ролью пользователя

```
<?php
include ("bd.php");
$response = array();
$prepod = $_POST['prepod']; $predmet = $_POST['predmet'];
$grupa = $_POST['grupa']; $zanytie = $_POST['zanytie'];
$lekciya = "Лекция";
$practika = "Практическое занятие";
$lab = "Лабораторная работа";
$flag_zanytiya = 0;
$result = mysql_query
("SELECT `id_uchplan`, `zanytie`FROM  `ucheb_plan`
  INNER JOIN  `vid_zanytya` ON `vid_zanytya`.`id_zanytie` =
              ucheb_plan`.`id_zanytie`
  INNER JOIN  `prepodavatel` ON `prepodavatel`.`id_prepodavatel` =
              `ucheb_plan`.`id_prepodavatel`
  INNER JOIN  `predmet` ON  `predmet`.`id_predmet` =
              `ucheb_plan`.`id_predmet`
  INNER JOIN  `grupa` ON  `grupa`.`id_grupa` =  `ucheb_plan`.`id_grupa`
  WHERE `vid_zanytya`.`zanytie` = '$zanytie' and
        `prepodavatel`.`fio_prepodavatel` = '$prepod' and
        `predmet`.`name_predmet` = '$predmet' and `grupa`.`name_grupa` =
        '$grupa'
 ") or die(mysql_error());
if (mysql_num_rows ($result) == 0){
      $response["Eror"] = 1;
      echo json_encode($response);
}
else{
        $response["Eror"] = 0;
}
?>
```

3.6. Руководство пользователя

3.6.1. Запуск приложения

Запуск приложения осуществляется нажатием по иконке приложения на экране мобильного устройства.

Для функционирования мобильного приложения требуется наличие доступа в интернет.

3.6.2. Авторизация

Для начала работы с приложением пользователю необходимо ввести свои учетные данные, как на рис. 3.12 и нажать кнопку «Вход». Пункт меню в верхней части экрана содержит краткую информацию данном экране. Если установить флажок «запомнить», приложение сохранит учетные данные пользователя и при следующем запуске, авторизация не потребуется.

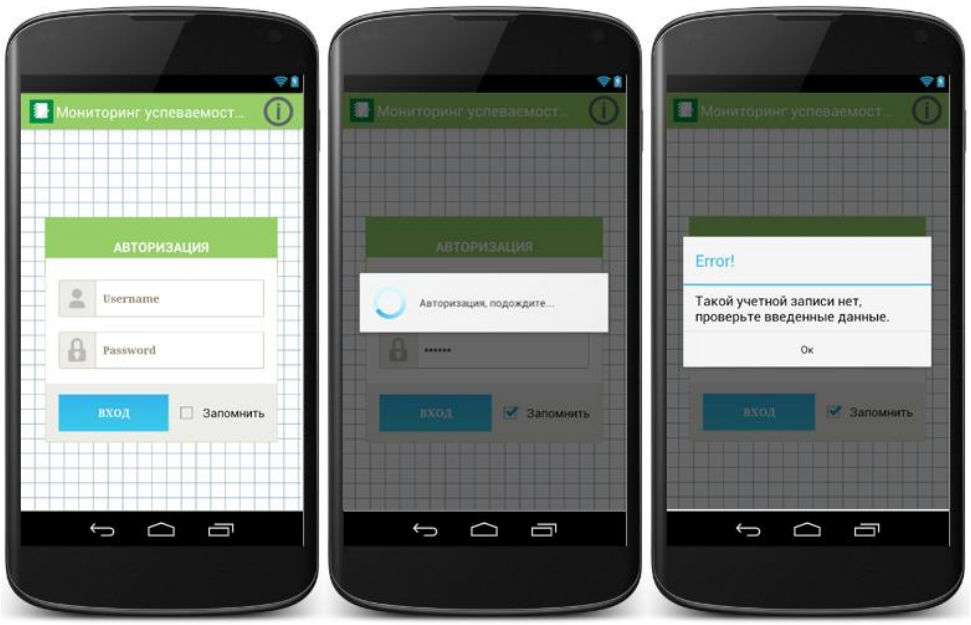


Рис. 3.12. Авторизация пользователя

После успешной авторизации происходит переход на экран параметров фильтрации в зависимости от роли авторизованного пользователя.

3.6.3. Преподаватель

3.6.3.1. Параметры фильтрации

Экран выводит три списка:

- список групп;

- список предметов;
- список видов занятий.

В верхней части экрана выводится ФИО преподавателя. Чтобы изменить параметры фильтрации, необходимо нажать на соответствующий пункт и в появившемся диалоге выбрать то что требуется. В пункте меню присутствует кнопка «Обновить», которая повторно перезагрузит данные с сервера в случае сбоя (рис. 3.13), кнопка «Помощь», по нажатию которой можно получить информацию по работе с данным экраном, а также кнопка «Выход из учетной записи», после нажатия которой приложение отправит пользователя на экран авторизации. По нажатию кнопки «Продолжить» происходит переход на экран, который отображает все имеющиеся ведомости в зависимости от выбранных параметров фильтрации.

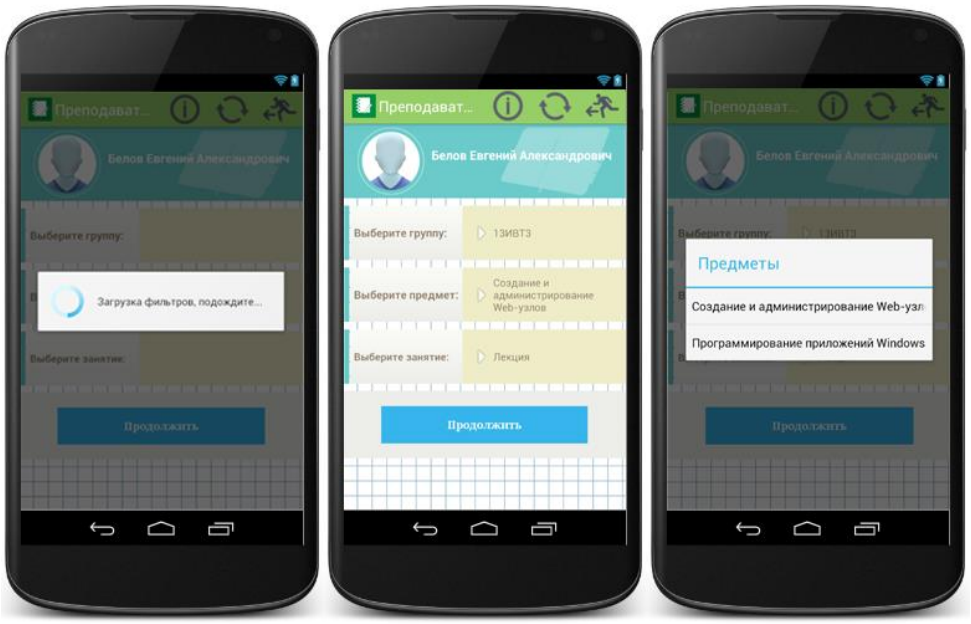


Рис. 3.13. Параметры фильтрации для преподавателя

Экран оборудован списком, каждый элемент которого содержит картинку, которая отображает утверждена ведомость преподавателем или нет, тему занятия, вид занятия и дату. В нижней части экрана закреплена панель с кнопкой для добавления ведомости. Пункт меню дублирует кнопку в нижней части экрана.

3.6.3.2. Добавление ведомости

По нажатию на кнопку «Добавить ведомость» происходит переход на экран добавления ведомости. Экран содержит таблицу в которую выводятся, группа, предмет, студенты для выбранной группы. Можно записать тему занятия. Чтобы добавить дату и колонки для записи количества пропущенных часов и оценки нужно нажать на кнопку «+». Выставление количества пропущенных часов и оценки осуществляется нажатием на ячейку, после чего выводится диалоговое окно в котором можно выбрать пункт с соответствующим количеством пропущенных часов или оценкой. Можно утвердить ведомость и запретить этим ее редактирование старосте (рис. 3.14).

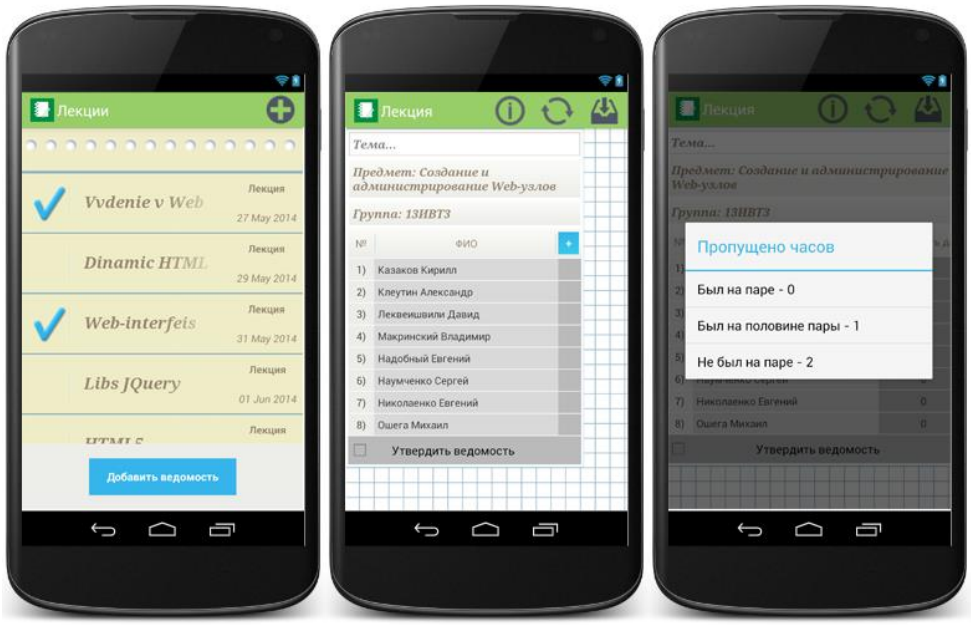


Рис. 3.14. Добавление ведомости преподавателем

По завершению заполнения можно сохранить данные путем нажатия кнопки пункта меню.

3.6.3.3. Редактирование ведомости

Чтобы редактировать ведомость, необходимо задать параметры фильтрации и перейти к списку ведомостей. Редактирование ведомости

происходит по нажатию на элемент списка в котором отображаются все доступные для редактирования ведомости, сформированные в зависимости от параметров фильтра.

После нажатия на элемент списка осуществляется переход на экран редактирования, в который выводятся данные выбранной ведомости.

Можно редактировать тему, дату, количество пропущенных часов и оценку, если это практическое или лабораторное занятие, а также можно утвердить ведомость. По завершению редактирования, можно сохранить данные путем нажатия кнопки пункта меню (рис. 3.15).

Рис. 3.15. Редактирование ведомости преподавателем

3.6.3.4. Просмотр ведомости

Для преподавателя просмотр ведомости входит в редактирование, экран идентичен экрану просмотра ведомости, но экран редактирования позволяет выполнять действия.

3.6.4. Староста

3.6.4.1. Параметры фильтрации

Экран выводит два списка:

- список предметов;
- список видов занятий.

В верхней части экрана на рис. 3.16 выводится ФИО старосты и его группа. Чтобы изменить параметры фильтрации, необходимо нажать на соответствующий пункт и в появившемся диалоге выбрать то что требуется.

В пункте меню присутствует кнопка «Обновить», которая повторно перезагрузит данные с сервера в случае сбоя, кнопка «Помощь», по нажатию которой можно получить информацию по работе с данным экраном, а также кнопка «Выход из учетной записи», после нажатия которой приложение отправит пользователя на экран авторизации.

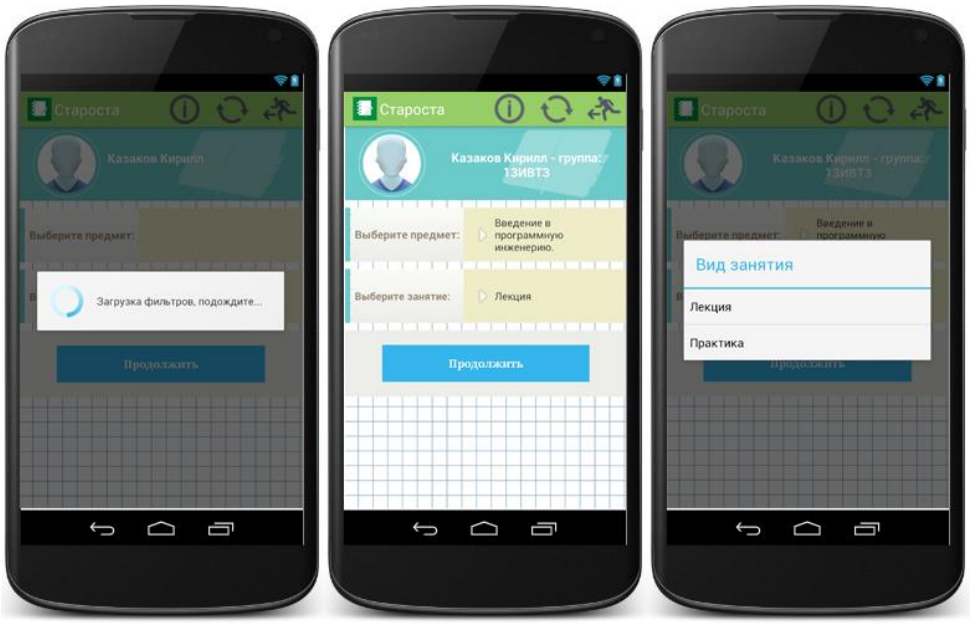


Рис. 3.16. Параметры фильтрации для старосты и студента

По нажатию кнопки «Продолжить» происходит переход на экран, который отображает все имеющиеся ведомости в зависимости от выбранных параметров фильтрации. Экран оборудован списком, каждый элемент

которого содержит картинку, а также отображает утверждена ведомость преподавателем или нет, тему занятия, вид занятия и дату. В нижней части экрана закреплена панель с кнопкой для добавления ведомости. Пункт меню дублирует кнопку в нижней части экрана.

3.6.4.2. Добавление ведомости

По нажатию на кнопку добавляющую ведомость происходит переход на экран добавления ведомости. Экран содержит таблицу в которую выводятся, группа, предмет, студенты для выбранной группы.

Можно записать тему занятия. Чтобы добавить дату и колонки для записи количества пропущенных часов нужно нажать на кнопку «+». Выставление количества пропущенных часов осуществляется нажатием на ячейку, после чего выводится диалоговое окно в котором можно выбрать пункт с соответствующим количеством пропущенных часов. Если это практическое или лабораторное занятие, колонка оценок создается, но блокируется.

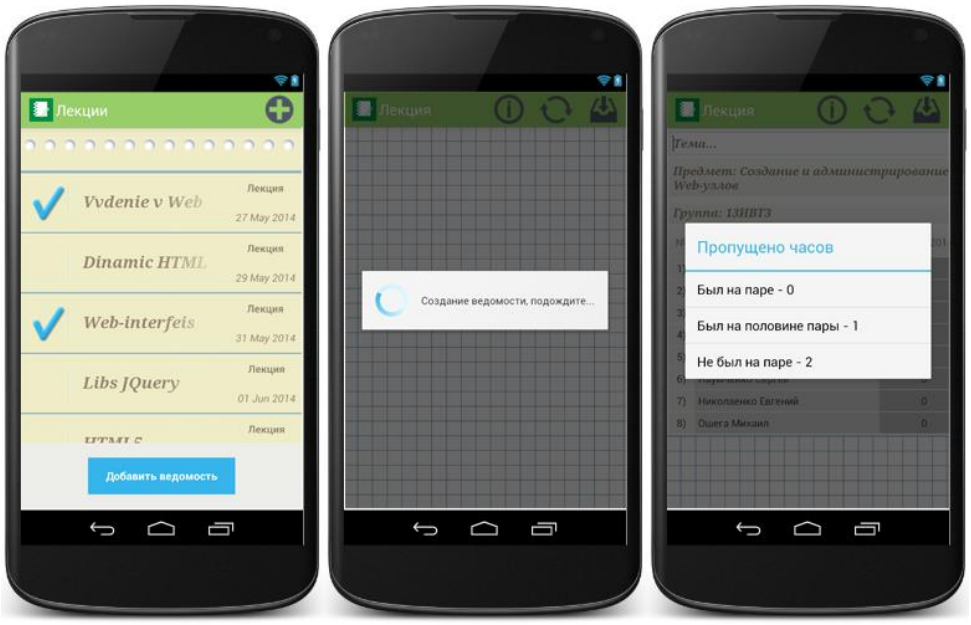


Рис. 3.17. Добавление ведомости старостой

По завершению заполнения можно сохранить данные путем нажатия кнопки пункта меню (рис. 3.17).

3.6.4.3. Редактирование ведомости

Чтобы редактировать ведомость, необходимо задать параметры фильтрации и перейти к списку ведомостей. Редактирование ведомости происходит по нажатию на элемент списка в котором отображаются все доступные для редактирования ведомости, сформированные в зависимости от параметров фильтра.

После нажатия на элемент списка осуществляется переход на экран редактирования, в который выводятся данные выбранной ведомости. Можно редактировать тему, дату, количество пропущенных часов, если это практическое или лабораторное занятие то колонка с оценкой будет заблокирована (рис. 3.18).

По завершению редактирования, можно сохранить данные путем нажатия кнопки пункта меню.

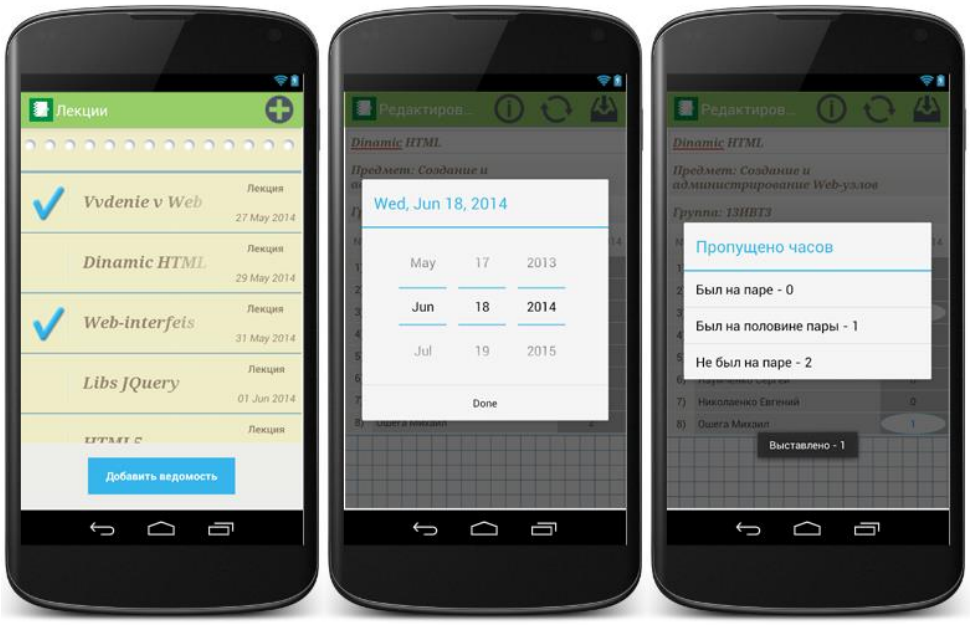


Рис. 3.18. Редактирование ведомости старостой

3.6.4.4. Просмотр ведомости

Экран просмотра ведомости срабатывает в случае если ведомость утверждена преподавателем и ее нельзя редактировать. Чтобы просмотреть ведомость, необходимо задать параметры фильтрации и перейти к экрану списка ведомостей и нажать на элемент списка (рис. 3.20).

3.6.5. Студент

3.6.5.1. Параметры фильтрации

Экран выводит два списка:

- список предметов;
- список видов занятий.

В верхней части экрана на рис. 3.19 выводится ФИО студента и группа. В пункте меню присутствует кнопка «Обновить», которая повторно перезагрузит данные с сервера в случае сбоя, кнопка «Помощь», по нажатию которой можно получить информацию по работе с данным экраном, а также кнопка «Выход из учетной записи», после нажатия которой приложение отправит пользователя на экран авторизации.

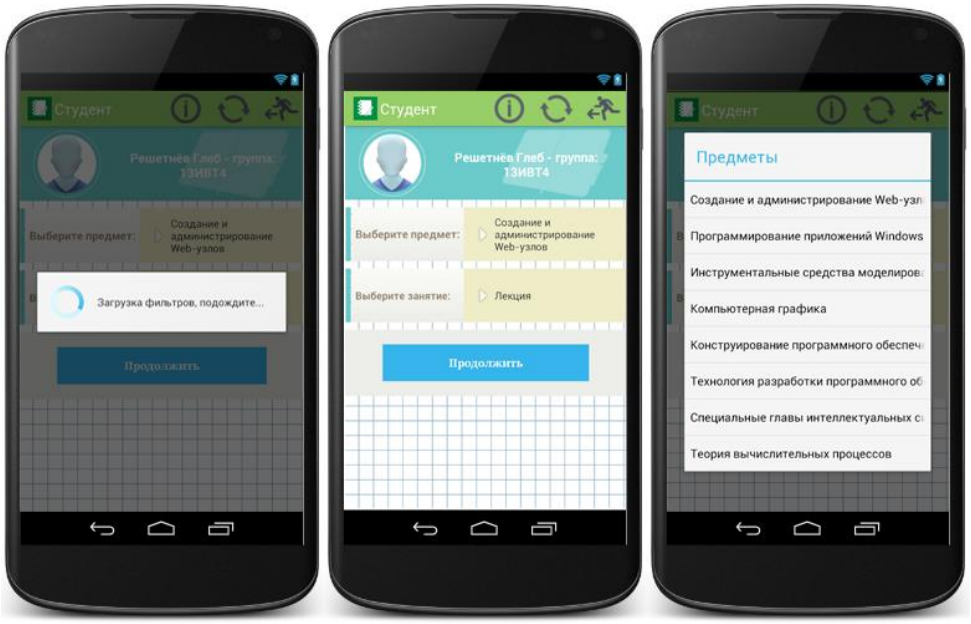


Рис. 3.19. Параметры фильтрации для старосты и студента

Чтобы изменить параметры фильтрации, необходимо нажать на соответствующий пункт и в появившемся диалоге выбрать то что требуется.

По нажатию кнопки «Продолжить» происходит переход на экран, который отображает все имеющиеся ведомости в зависимости от выбранных параметров фильтрации. Экран оборудован списком, каждый элемент которого содержит картинку, которая отображает утверждено занятие преподавателем или нет, тему занятия, вид занятия и дату.

3.6.5.2. Просмотр ведомости

Для студента доступен только экран просмотра ведомости (рис. 3.20). Чтобы просмотреть ведомость, необходимо задать параметры фильтрации и перейти к экрану списка ведомостей и нажать на элемент списка.

Рис. 3.20. Просмотр ведомости студентом и старостой

3.6.1. Завершение работы приложения

В мобильном приложении следует перейти к экрану параметров фильтрации и нажать кнопку «Back» на мобильном устройстве, после чего подтвердить свой выбор.

4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Тестирование – неотъемлемая составляющая процесса программной инженерии, один из методов улучшения качества разработанного программного обеспечения системы посредством выявления дефектов, не обнаруженных ранними видами проверок. В данном разделе описываются методы и результаты тестирования.

При тестировании использовался метод «черного ящика». Система, которую представляют, как «черный ящик», рассматривается как имеющая некий «вход» для ввода информации и «выход» для отображения результатов работы, при этом происходящие в ходе работы системы процессы наблюдателю неизвестны. Предполагается, что состояние выходов функционально зависит от состояния входов [10].

Изучение системы по методу чёрного ящика сводится к наблюдениям за ней и проведению экспериментов по изменению входных данных, при этом в ходе наблюдения над реакциями системы на внешние воздействия достигается определённый уровень знаний об исследуемом объекте, позволяющий осуществлять прогнозирование поведения «чёрного ящика» при любых заданных условиях.

4.1. План испытаний

Для того, чтобы провести испытание соответствия мобильного приложения поставленным целям, составлен план испытаний:

- проверка авторизации;
- проверка в случае обрыва соединения с интернетом;
- проверка добавления ведомости;
- проверка редактирования ведомости преподавателем;
- проверка редактирования ведомости старостой;
- проверка экрана просмотра ведомости.

4.2. Проверка выполнения функциональных требований

4.2.1. Проверка авторизации

Для проверки авторизации был выполнен вход случайно выбранных пользователей. Результаты не выявили ошибок, каждый пользователь авторизовался по соответствующей ему ролью. При вводе неправильных учетных данных выдавалось соответствующее сообщение.

4.2.2. Проверка в случае обрыва соединения с интернетом

Для проверки ситуации, когда происходит обрыв соединения с интернетом, случайным образом в момент загрузки каких-либо данных или их сохранения отключался интернет. В случае с загрузкой данных, данные загружаются не полностью, но есть возможность повторного соединения, что решает данную проблему. Целостность сохраняемых данных обеспечивает транзакция базы данных. При проверки обрыв соединения при сохранении привел к невозможности сохранения всей порции данных.

После восстановления соединения приложение продолжает работать штатно.

4.2.3. Проверка добавления ведомости

Для проверки добавления ведомости было выполнено 30 добавлений для каждого вида занятия под несколькими пользователями. Результат показал, что все ведомости добавляются корректно, данные записываются в базу без потери.

4.2.4. Проверка редактирования ведомости преподавателем

Чтобы проверить возможность редактирования преподавателем ведомости, было запущено приложение и осуществлен вход под преподавателем. Были заданы параметры фильтрации после чего перешли на

экран списка ведомостей. Выбрали произвольную ведомость и открыли её, данные корректно загрузились с сервера. Была изменена тема, и оценка одного студента, после чего данные сохранили. Ведомость без ошибок обновилось с новыми заданными данными. После провели еще пару экспериментов, с другими параметрами фильтра. Проверили возможность редактировать утверждённые занятия. Результаты не выявили ошибок.

4.2.5. Проверка редактирования ведомости старостой

Чтобы проверить возможность редактирования старостой ведомости, было запущено приложение и осуществлен вход под старостой. Были заданы параметры фильтрации после чего перешли на экран списка занятий. Выбрали произвольную ведомость и открыли её, данные корректно загрузились с сервера. Была изменена дата, и количество пропущенных часов пары студентов, после чего сохранили. Ведомость без ошибок обновилось с новыми заданными данными. Проверили что кнопка утвердить для старосты отсутствует. После провели еще пару экспериментов, с другими параметрами фильтра. Проверили невозможность редактировать утверждённые занятия. Результаты не выявили ошибок.

4.2.6. Проверка экрана просмотра ведомости

Экран просмотра ведомости активируется в случае, если редактирование запрещено. Для проверки зашли с двух мобильных устройств, на одном авторизовались под старостой, на другом по студентом. Задали параметры фильтрации и перешли к списку ведомостей. В случае со старостой, как и должно быть осуществлялся переход на экран просмотра ведомости, если занятие не утверждено. В случаем со студентом, всегда был переход на экран просмотра ведомости, как и должно быть. Результаты не выявили ошибок, каждый пользователь переходит на экран просмотра в соответствии с его правами доступа.

4.3. Нагрузочное тестирование

Для нагрузочного тестирования используется одновременное обращение 40 устройств к серверу. Для создания такой нагрузки, использовались 10 реальных устройств, 10 эмуляторов Android (рис. 4.2), а также программа SoapUI [16] с помощью которой были сформированы запросы и оправлены на сервер, результаты выведены на рис. 4.1.

Simulate 20 clients

Limit: 60 Seconds 100%

Threads: 500 Strategy Simple Test Delay 10000 Random 1

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
initial wait [1000]	999	1018	999.33	999	501	5.77	0	0	0	0
ApproveSession	106	792	227.05	214	501	5.77	198396	2285	0	0
wait 1 [10000]	9999	10737	10,033.13	9999	501	5.77	0	0	0	0
ApproveSession 2	105	3593	263.93	200	501	5.77	198396	2285	0	0
wait 2 [10000]	9999	10016	9,999.49	9999	501	5.77	0	0	0	0
ApproveSession 3	105	1019	255.89	267	501	5.77	198396	2285	0	0
wait 3 [10000]	9999	10177	10,002.55	10177	501	5.77	0	0	0	0
ApproveSession 4	106	1144	245.57	284	501	5.77	198396	2285	0	0
wait 4 [10000]	9999	10081	9,999.66	9999	501	5.77	0	0	0	0
TerminateSession	116	931	235.18	292	501	5.77	151302	1742	0	0
TestCase:	41533	49508	42,261.82	42430	501	5.77	944886	10883	0	0

Рис. 4.1. Нагрузочное тестирование с помощью SoapUI

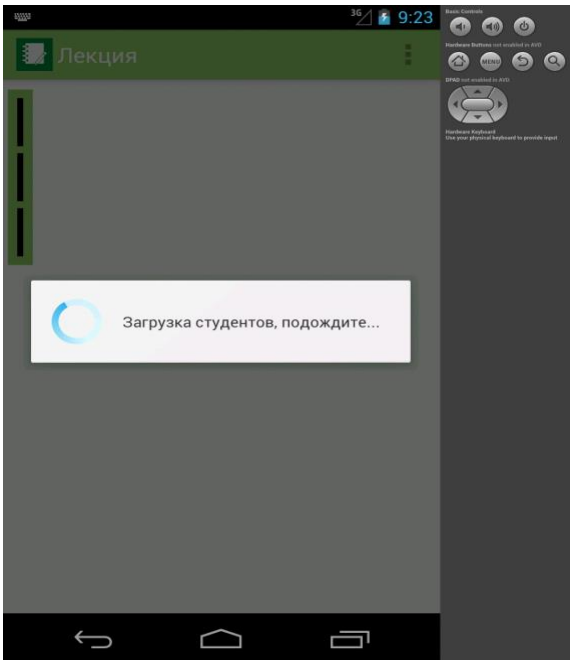


Рис. 4.2. Эмулятор Android

5. ОРГАНИЗАЦИОННАЯ ЧАСТЬ

Бурная мобильная «телефонизация», захлестнувшая человечество каких-то десять лет назад, продолжается и по сей день. Неудивительно, что людей стал беспокоить вопрос о влиянии электромагнитных волн на здоровье человека. Эта проблема исследуется уже давно, еще со второй половины XX века, но она затрагивала довольно узкий круг лиц, преимущественно персонал вещательных и специализированных радиостанций. Уже тогда принимались меры к защите человека от излучения, существовали определенные правила при работе вблизи источников мощных излучений. И, несмотря на революционные перемены в области телекоммуникаций, а также на множество открытий, влияние электромагнитных волн различных частот горячо обсуждается до сих пор.

5.1. Общая характеристика радиочастотных излучений

При обсуждении возможных неблагоприятных эффектов для здоровья при воздействии на человека радиочастотных (РЧ) полей важно не путать РЧ-поля с ионизирующим излучением – таким как, рентгеновские лучи, гамма-лучи или коротковолновое ультрафиолетовое излучение. В отличие от ионизирующего излучения даже мощные РЧ-поля не могут вызвать ионизацию или радиоактивность в организме. По этой причине РЧ-поля называют неионизирующей радиостанцией. Современные системы мобильной телефонной связи работают на частотах от 800 до 1900 МГц. Системы, использующие частоты 2100 МГц и выше, могут появиться вскоре. Эти частоты попадают в диапазон от 1 МГц до 10 ГГц (1 ГГц = 1 тысячи МГц). Неблагоприятные РЧ-воздействия для здоровья в этом диапазоне излучаются на протяжении 45 лет [18].

Исследования влияния электромагнитных полей на здоровье человека касались в основном работников промышленных предприятий, имеющие контакт с генераторами электромагнитного излучения. Было выделено

заболевание – радиоволновая болезнь. Она проявляется в функциональных нарушениях нервной системы, неврастеническом и астеническом синдромах. Люди, долгое время находящиеся в электромагнитном поле, жалуются на слабость, раздражительность, быструю утомляемость, ослабление памяти, нарушение сна.

Воздействие на здоровье РЧ-полей от 1 МГц до 10 ГГц. Как известно, РЧ-поля в этом диапазоне проникают сквозь экспонированные ткани и создают нагрев вследствие поглощения энергии. Глубина поглощения РЧ-поля в ткань зависит от частоты поля, и она выше при более низких частотах. Однако международные технические стандарты, согласно которым производятся мобильные телефоны и их базовые станции, не позволяют им вызвать сколь-либо значительного нагрева (больше 1 градуса Цельсия).

На всех расстояниях уровни РЧ-поля на поверхности земли от базовых станций вполне вписываются в международные нормативы по воздействию РЧ-излучений на население. Некоторые антенны, установленные на крышах зданий, снабжены ограждениями, предотвращающим доступ людей в те места, где РЧ-поля превышают эти пределы. Поскольку антенны, установленные по периметру зданий, посылают свою мощность наружу, люди, находящиеся внутри, не подвергаются воздействию высоких уровней электромагнитных полей (ЭМП).

5.2. Медицинские исследования влияния мобильных телефонов на здоровье человека

По данным исследований, проведенным в Норвегии и Швеции, была выявлена подверженность молодых людей (11 тысяч опрошенных пользователей сотовой связи) влиянию излучения. Исследование показало, что даже люди, которые использовали телефон менее двух минут в день, жаловались на дискомфорт и сторонние эффекты [19].

Проблемы со здоровьем возникают, если пользоваться телефоном дольше. У тех, кто им пользуется около 30 минут ежедневно, повышается вероятность потери памяти почти в два раза по сравнению с теми, кто ограничивается двумя минутами в день. Половина опрошенных абонентов сообщили, что при использовании мобильных телефонов испытывают неприятный разогрев в области головы вокруг уха. Наибольшему риску подвергаются люди. Те, кому нет еще 30, в 3-4 раза чаще подвержены посторонним эффектам.

5.3. Вероятные клинические последствия влияния мобильных телефонов

Можно выделить четыре системы, наиболее подверженные вредному влиянию:

1. **Центральная нервная система**. Она наиболее чувствительна к электромагнитным полям. Наблюдаются изменения высшей нервной деятельности, ухудшение памяти, внимания, воли, нарушение сна.

Особенно интересен разбор влияния мобильного телефона на человека при разговоре разной длительности. Первые 15 секунд не отмечается никаких изменений. С 15-й секунды – наступает угнетение альфа-волн мозговой деятельности, что проявляется ослаблением волевых функций, памяти, внимания, концентрации. При более длительном и регулярном использовании телефона в течение нескольких лет могут наблюдаться и другие изменения – нарушение сна, угнетение иммунитета, анемия.

Поскольку телефон расположен около головы, значительно повышается риск развития опухоли мозга, нарушение нормального режима сна, нарушение смены фаз дня.

2. **Иммунная система**. Происходит угнетение иммуногенеза, что приводит к ухудшению сопротивляемости организма к различным

инфекциям. Мобильные телефоны в свою очередь способствуют развитию астмы, экземы.

3. **Эндокринная система**. В результате воздействия на неё увеличивается содержание адреналина в крови.
4. **Половая система**. Этот вопрос наиболее важен, так как именно молодые люди репродуктивного возраста особенно часто пользуются источниками электромагнитных волн.

Вернемся к тому, что на физическое здоровье влияет электромагнитное излучение сотового телефона. В действии любого электромагнитного излучения принято выделять два эффекта: **термический** и **нетермический**.

Термический эффект можно наблюдать, если поместите курицу в микроволновку и через некоторое время получите готовое блюдо. Примерно такое же действие оказывает и излучение от сотового телефона. А антенна, основной излучатель телефона, находится в 3 см от головного мозга. Естественно температура отдельных участков мозга повышается. При длительном разговоре этот эффект можно ощущать по повышению температуры ушной раковины. Другой орган, подверженный влиянию излучения от сотового телефона – это хрусталик глаза. Он плохо кровоснабжается и поэтому особенно подвержен действию электромагнитного излучения. В результате падает острота зрения.

Нетермический или информационный эффект проявляется в том, что частоты сотовых аппаратов совпадают с частотами естественной биоэлектрической активности головного мозга человека (с так называемым гамма-ритмом мозга, с альфа-ритмом и с дельта-ритмом). Путем резонанса внешних (от телефона) и внутренних (собственных мозга) частот нарушается нормальное течение функций головного мозга. Такие изменения заметны на электроэнцефалограмме и не исчезают длительное время после окончания разговора. Когда мобильник теряет сеть, он становится по-настоящему вредным, он включает специальный режим «Поиск сети» – начинает

испускать сильные направленные потоки электромагнитных импульсов, как бы пытаясь «докричаться» до сети. При этом мощность и частотность поискового сигнала значительно выше, чем при работе в обычном режиме. Эти импульсы попадают не только в воздух, но и в хозяина телефона. Если телефон лежит в верхнем кармане пиджака или висит на шейном ремешке, то страдают легкие и сердце. У девушек может развиться мастопатия и даже перерождение ткани молочной железы. Если телефон висит на поясе или лежит в нижнем кармане, страдают печень и половые органы.

Теоретические расчеты показывают, что если режим «Поиск сети» включается раз в день минут на 20 – это абсолютно безопасно. Но если телефон работает в этом режиме по 4-5 часов ежедневно (например, при поездках в метро, если сеть под землей не поддерживается), то через 3-4 года могут появиться различные осложнения. Основными симптомами неблагоприятного воздействия сотового телефона на состояние здоровья являются: головные боли; нарушения памяти и концентрации внимания; непреходящая усталость; депрессивные заболевания; боль и резь в глазах, сухость их слизистой; прогрессивное ухудшение зрения; снижение потенции; повышение артериального давления и пульса.

На сегодняшний день все-таки нельзя точно сказать, что использование сотового телефона вредно или безопасно. Но следует придерживаться следующих правил:

- не следует разговаривать непрерывно более 3-4 минут;
- дома и в офисе следует пользоваться обычными проводными телефонами;
- пользуйтесь чаще услугами SMS;
- не держите телефон у изголовья кровати ночью;
- носите мобильный телефон в сумочке;
- не следует пользоваться сотовыми телефонами детям;
- не следует разговаривать в автомашине по сотовому телефону;

- отключайте телефоны или перекладывать их из карманов в сумки при попадании в зоны неустойчивой сети.

ЗАКЛЮЧЕНИЕ

В ходе дипломной работы была разобрана предметная область, рассмотрены программы аналоги, вывялены их слабые и сильные стороны, которые помогли в построении модели разрабатываемого мобильного приложения Определена среда разработки будущего приложения на основании сравнения достоинств и недостатков под конкретную задачу работы. Так же были поставлены цели, которые необходимо выполнить для успешного выполнения работы.

Результатом работы стало в полной мере функционирующее мобильное приложение мониторинга посещаемости занятий и успеваемости студентов, которое позволяет редактировать и просматривать ведомости занятий пользователям в зависимости от роли.

В дальнейшие планы развития проекта входит доработка серверной части системы, предусматривающая создание web-интерфейса для построения различных отчетов, а также механизмов интеграции с другими системами учебного заведения.

СПИСОК ЛИТЕРАТУРЫ

1. Знакомьтесь, Google Studio – Режим доступа: http://developer.alexanderklimov.ru/android/theory/androidstudio.php
2. Электронный журнал в образовательном учреждении для самоконтроля студентов – Режим доступа: http://instrument-p.narod.ru/_zanat/zhurnal/zhurn_e.htm
3. Дневник.ру – Режим доступа: http://dnevnik.ru/
4. Зачем нужны электронные журналы – Режим доступа: http://eljur.ru/elektronnye-zhurnaly-i-dnevniki
5. Android Studio 0.4.0 Released – Режим доступа: http://tools.android.com/recent/androidstudio040released
6. Дневник в кармане. Установка приложения из Магазина Windows – Режим доступа: http://apps.microsoft.com/windows/ru-ru/app/5675013e-2c4e-4eb3-a64b-13cdfdf69f50
7. Преимущества электронных классных журналов – Режим доступа: http://www.edu.ru/index.php?page_id=5&topic_id=3&sid=27221
8. LMS Школа. Классный журнал – Режим доступа: http://www.lms-school.ru/index.php/products2/lms-shkola-klassnyj-zhurnal
9. IntelliJ IDEA. Среда разработки под Android – Режим доступа: http://www.jetbrains.com/idea/
10. Бейзер, Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем /Б. Бейзер. – М.:Питер, 2010. – 230 с.
11. Рассел, Д. Диаграмма Ганта: учеб. пособие /Д. Рассел. – М.: Книга по Требованию, 2012. – 42 с.
12. Делисль, М. Mastering phpMyAdmin / М. Делисль. – М.: РАСКТ, 2012. – 394с.
13. Фаулер, М. UML. Основы объектно-ориентированного моделирования: учеб. пособие /Д. Фаулер. – СПб.: Питер, 2010. – 150с.

14. Проектирование интерфейсов – Режим доступа: http://balsamiq.com/products/mockups/
15. Харрингтон, Д. Проектирование реляционных баз данных: учеб. пособие / Д. Харрингтон. – М.: МК-Пресс, 2008. – 403с.
16. Автоматизация тестирования веб-сервисов – Режим доступа: http://www.soapui.org/
17. О страховых взносах в сфере IT-технологий: Федеральный закон о страховых взносах от 24.07.09 № 212-ФЗ // Собрание законодательства РФ. – 2014. – № 5. – 35 с.
18. Прохорова, О. Безопасность жизнедеятельности. Учебник для бакалавров: учеб. пособие / О. Прохорова. – М.: Дашков и К, 2013. 457 с.
19. Стожаров, А. Медицинская экология: учеб. Пособие /А. Стожаров. – М.:Минск: Высшая школа, 2007. 254 с.
20. Шварц, Б. MySQL. Оптимизация производительности / Б. Шварц, П. Зайцев, В. Ткаченко; пер. с англ. под ред. А. Слинкина. – СПб.: Символ-Плюс, 2012. – 832с.
21. CA ERwin Data Modeler – Режим доступа: http://erwin.com/worldwide/russian-russia