

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт ИнЭО

Направление подготовки Информатика и вычислительная техника

Кафедра Автоматики и компьютерных систем

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка компонентов информационной поддержки расписания медицинского учреждения УДК <u>004.4'22</u>

Студент

Группа	ФИО	Подпись	Дата
3-8В12	Николаев Андрей Вячеславович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Пономарев Алексей Анатольевич	К.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Николаенко Валентин Сергеевич	Аспирант		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Назаренко Ольга Брониславовна	Д.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Фадеев Александр Сергеевич	К.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЕ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
ПК1	Разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием.
ПК2	Осваивать методики использования программных средств для решения практических задач
ПК3	Разрабатывать интерфейсы "человек - электронно-вычислительная машина".
ПК4	Разрабатывать модели компонентов информационных систем, включая модели баз данных.
ПК5	Разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования.
ПК6	Обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности.
ПК7	Готовить презентации, научно-технические отчеты по результатам выполненной работы, оформлять результаты исследований в виде статей и докладов на научно-технических конференциях.
ПК8	Готовить конспекты и проводить занятия по обучению сотрудников применению программно-методических комплексов, используемых на предприятии.
ПК9	Участвовать в настройке и наладке программно-аппаратных комплексов.
ПК10	Сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.
ПК11	Инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт ИнЭО

Направление подготовки Информатика и вычислительная техника

Кафедра Автоматики и компьютерных систем

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
3-8В12	Николаев Андрей Вячеславович

Тема работы:

Разработка кроссплатформенного клиент-серверного игрового приложения «Найди пару»	
-----------------------------------------------------------------------------------	--

Утверждена приказом директора (дата, номер)	От 15.04.2016, №2917/с
---------------------------------------------	------------------------

Срок сдачи студентом выполненной работы:	1.06.2016
------------------------------------------	-----------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	<p>1) Язык разработки C#</p> <p>2) Среда разработки MS Visual Studio 2015 с использованием технологии XAMARIN</p> <p>3) .NET Framework 4.5.2</p> <p>4) Android SDK API level 20-23</p>
Перечень подлежащих исследованию, проектированию и разработке вопросов	<p>1) Обзор медицинских информационных систем и существующих решений</p> <p>2) Обзор и выбор платформы и среды разработки</p> <p>3) Создание шаблона мобильного приложения</p> <p>4) Проектирование мобильного приложения</p> <p>6) Финансовый менеджмент проекта</p> <p>7) Социальная ответственность проекта</p>
Перечень графического материала	---
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент	Николаенко Валентин Сергеевич, ассистент
Социальная ответственность	Назаренко Ольга Брониславовна, профессор
Названия разделов, которые должны быть написаны на русском и иностранном языках:	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	30.01.2016
-------------------------------------------------------------------------------------------------	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Пономарев Алексей Анатольевич	К.т.н.		30.01.2016

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8B12	Николаев Андрей Вячеславович		30.01.2016

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт ИнЭО

Направление подготовки (специальность) Информатика и вычислительная техника

Уровень образования Бакалавр

Кафедра АиКС

Период выполнения весенний семестр 2015/2016 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	1.06.2016
------------------------------------------	-----------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	<i>Введение</i>	
	<i>Обзор литературы</i>	
	<i>Объект и методы исследования</i>	
	<i>Описание предметной области</i>	
	<i>Техническое задание</i>	
	<i>Проектирование</i>	
	<i>Руководство пользователя</i>	
	<i>Финансовый менеджмент</i>	
	<i>Социальная ответственность</i>	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Пономарев Алексей Анатольевич	К.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Фадеев Александр Сергеевич	К.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
3-8В12	Николаев Андрей Вячеславович

Институт	ИнЭО	Кафедра	АиКС
Уровень образования	Бакалавр	Направление/специальность	Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Человеческие ресурсы: 1 чел.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	1) Анализ конкурентов 2) SWOT - анализ
2. <i>Планирование и формирование бюджета научных исследований</i>	
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	

Перечень графического материала:

Дата выдачи задания для раздела по линейному графику	1.03.2016
-------------------------------------------------------------	-----------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Николаенко Валентин Сергеевич	Аспирант		1.03.2016

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8В12	Николаев Андрей Вячеславович		1.03.2016

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
3-8В12	Николаев Андрей Вячеславович

Институт	ИнЭО	Кафедра	АиКС
Уровень образования	Бакалавр	Направление/специальность	Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:

<i>Описание рабочего места (рабочей зоны, технологического процесса, механического оборудования)</i>	<i>Рабочее место программиста</i>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Анализ выявленных вредных факторов проектируемой производственной среды в следующей последовательности:</p> <ul style="list-style-type: none"> – физико-химическая природа вредности, её связь с разрабатываемой темой; – действие фактора на организм человека; – приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ); – предлагаемые средства защиты (сначала коллективной защиты, затем – индивидуальные защитные средства) 	<p>1) Требования к помещениям при работе за компьютером</p> <p>2) Требования к микроклимату, ионному составу и концентрации вредных химических веществ в воздухе помещений</p> <p>3) Требования к освещению помещений и рабочих мест</p> <p>4) Требования к шуму и вибрации в помещениях</p> <p>5) Требования к организации и оборудованию рабочих мест</p> <p>6) Режим труда и отдыха при работе с компьютером</p>
<p>2. Анализ выявленных опасных факторов проектируемой произведённой среды в следующей последовательности</p> <ul style="list-style-type: none"> – механические опасности (источники, средства защиты); – электробезопасность; – пожаровзрывобезопасность 	<p>Обеспечение электробезопасности и пожарной безопасности на рабочем месте</p>
Перечень графического материала:	

Дата выдачи задания для раздела по линейному графику	
-------------------------------------------------------------	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Назаренко Ольга Брониславовна	Д.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8В12	Николаев Андрей Вячеславович		

РЕФЕРАТ

Выпускная квалификационная работа _____ 77 _____ с., _____ 16 _____ рис., _____ 3 _____ табл.,
_____ 17 _____ источников, _____ 0 _____ прил.

Ключевые слова: запись на прием, хатагин, мобильное приложение, личный кабинет пациента

Объектом исследования являются: мобильные технологии в медицинских информационных системах

Цель работы – разработка мобильного приложения для задач работы с расписанием медицинского учреждения

В процессе исследования проводились анализ предметной области, проектирование прототипа приложения, проектирование и разработка мобильного приложения

В результате исследования: разработано мобильное приложение для работы с медицинским расписанием для платформы Android

Основные конструктивные, технологические и технико-эксплуатационные характеристики: поддержка платформы Android, мобильный клиент

Степень внедрения: планируется вывести на рынок в 3 квартале текущего года

Область применения: здравоохранение

Экономическая эффективность/значимость работы: повышение доходов компании за счет постепенного проникновения на рынок

В будущем планируется: усовершенствование интерфейса, расширение функционала, перенос на платформы iOS, Windows Mobile

Оглавление

ВВЕДЕНИЕ	11
1 МЕДИЦИНСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ.....	13
1.1. Медицинские информационные системы	13
1.2 Мобильные технологии в медицинских информационных системах	14
1.3 Описание предметной области	15
1.4 Обзор существующих решений.....	16
1.5 Анализ популярных мобильных операционных систем	20
1.6 Анализ существующих сред разработки	22
1.7 Обоснование выбора мобильной платформы	24
1.8 Обоснование выбора среды разработки	24
1.9 Техническое задание	25
2 ПРОЕКТИРОВАНИЕ МЕДИЦИНСКОГО ПРИЛОЖЕНИЯ.....	29
2.1 Создание прототипа.....	41
2.2 Проектирование приложения на платформе Android.....	46
2.3.2 Работа со списками.....	49
2.3.3 Работа с уведомлениями	50
2.3.4 Работа с сетью.....	51
2.3.5 Модели данных	52
2.4 Результат разработки.....	58
3 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ	60
3.1 Введение	60
3.2 SWOT анализ.....	64
3.3 Заключение.....	65
4 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	66
4.1 Требования к помещениям при работе за компьютером	66
4.2 Требования к освещению	66
4.3 Требования к помещениям при работе за компьютером	68
4.4 Требования к микроклимату помещений	69
4.5 Требования к организации и оборудованию рабочих мест.....	69
4.6 Обеспечение электро-пожаробезопасности на рабочем месте.....	71
ЗАКЛЮЧЕНИЕ.....	74

ВВЕДЕНИЕ

В настоящее время мобильные устройства занимают все большее место в повседневной жизни людей. Современные мобильные устройства начали свой путь в качестве простых мобильных телефонов с весьма ограниченным набором функций, со временем приобретая все больше и больше всевозможных функций, превратившись, впоследствии, в незаменимых помощников во многих сферах человеческой жизни. Кроме того, с развитием сети интернет мобильные устройства получили возможность доступа к любым информационным ресурсам из любой точки земли, что еще расширило возможности современных гаджетов.

Современные мобильные устройства позволяют не только просматривать интернет страницы, но и работать с так называемыми облачными сервисами, которые в свою очередь делают возможным доступ к конфигурированным вычислительными ресурсам, например, серверам, устройствам хранения данных, приложениям и так далее.

Развитие мобильных и интернет технологий привело к появлению разнообразного программного обеспечения для работы с ресурсами глобальной сети. Оно позволяет решать различные задачи обучения, бизнес-задачи, обмена сообщениями и медиа данными, работу с различными расписаниями, документами, игровыми приложениями и многим другим, в чем непосредственную роль играет работа с каким-либо удалённым сетевым ресурсом или множеством ресурсов.

Развитие программного обеспечения современных мобильных устройств играет немаловажную роль и в медицине. Так как в настоящее время в системах здравоохранения появился ряд проблем, решение которых находится в плоскости информационных технологий. Огромные массивы различных данных, которые хранятся в большем своем объеме в бумажном виде, тяжело поддающиеся обработке и анализу, работа с бумагами, что занимает значительное время медработника и отвлекает его от выполнения своих

непосредственных обязанностей, тем самым снижая эффективность медицинской помощи. Помимо этого, отсутствие информационной связи между пациентами и лечебными учреждениями требует от пациента заводить медкарту в каждой больнице, диагностическом центре, куда он обращается и простаивать огромные очереди теряя силы и время для того что бы записаться к необходимому специалисту. Все это, а также множество других проблем привело к необходимости создания мобильных приложений, которые должны значительно оптимизировать работу медперсонала и облегчить процедуры записи на прием самого пациента.

Поэтому для решения выше поставленных проблем было принято решение разработать компоненты информационной поддержки расписания медицинского учреждения, используемых непосредственно на мобильных устройствах.

Задачами данной выпускной квалификационной работы является:

- 1) анализ существующих решений;
- 2) анализ популярных мобильных операционных систем, с целью выявления наиболее подходящей для разработки прототипа;
- 3) разработка требований к пользовательскому интерфейсу приложения для работы с расписанием медицинского учреждения;
- 4) разработка прототипа пользовательского интерфейса;
- 5) анализ конкурентов и проведение SWOT-анализа;
- 6) анализ социальной ответственности;
- 7) выбор среды разработки мобильного приложения;
- 8) разработка внутренних алгоритмов работы приложения и алгоритмов взаимодействия с удаленным сервером.

1 МЕДИЦИНСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

1.1. Медицинские информационные системы

Организация работы медицинских учреждений всегда сопровождается огромным оборотом данных, которые в большем объеме хранятся на бумажных носителях. Работа с бумагами значительно затрудняет процесс организации и анализа медицинской информации, что в свою очередь занимает большее количество времени врача, препятствуя этим повышению качества оказываемых им услуг. Поэтому необходимость использования больших и при этом постоянно растущих объемов информации при решении разного рода медицинских задач, обуславливает сегодня использование современных информационных технологий в работе медицинских учреждений.

Информационная технологии представляют собой упорядоченную совокупность способов и методов сбора, обработки, накопления, хранения, поиска распространения, защиты и потребления информации, осуществляемых в процессе управленческой деятельности.

Современные информационные технологии широко используют компьютеры, вычислительные сети и всевозможные виды программного обеспечения в процессе управления. Целью внедрения информационных технологий является создание информационных систем для анализа и принятия на их основе управленческих решений.

Медицинская информационная система – это средства автоматизации медицинского документооборота для лечебно-профилактических учреждений, который включает в себя ведения медицинской электронной карты о пациентах, данные о медицинских исследованиях, данные мониторинга пациентов с медицинских приборов, возможность электронной записи пациентов к врачу.[1]

Использование медицинской информационной системы предоставляет различные возможности разным группам пользователей:

- 1) медицинский персонал — для этих пользователей информационная система предоставляет доступ к сведениям о пациентах;
- 2) медицинская администрация — для этой группы на первый план выходит возможность оперативного контроля и управления лечебным процессом, оперативной получение достоверных статистических требований;
- 3) научным сотрудникам предоставляется возможность систематического сбора и анализа данных для научных исследований;
- 4) сотрудникам планово-экономических служб информационная система позволяет отслеживать материальные и финансовые потоки, связанные с лечебно-диагностическим процессом;
- 5) пациентам предоставляется возможность хранения своей истории болезни, назначенного курса лечения, врачебные рекомендации и возможность использования электронной регистратуры для упрощения и экономии времени записи к врачу.[1]

Все это в купе позволяет значительно повысить общую эффективность медицинских работников и облегчить взаимодействие пациентов с медицинскими учреждениями.

1.2 Мобильные технологии в медицинских информационных системах

В настоящее время мобильные технологии все больше входят в повседневную жизнь человека. Сейчас практически каждый человек имеет мобильный смартфон, который предоставляет мгновенный доступ к любым информационным ресурсам. Развитие информационных технологий значительно расширило возможности различных информационных систем, в том числе и медицинских. Современные мобильные приложения позволяют организовывать взаимодействие пациентов и учреждений посредством персонального кабинета пациента. В личном кабинете пользователь может

найти необходимое ему лечебно-профилактическое учреждение, специалиста, просмотреть информацию об услугах, удаленно записаться на прием, выбрав удобное для себя время, так же с помощью таких мобильных приложений пользователь может просмотреть личную историю болезни и назначенное ему лечение.

1.3 Описание предметной области

В данной выпускной квалификационной работе будет разработано мобильное приложение для задач работы с расписанием медицинских учреждений.

Данное приложение предназначено для организации личного кабинета с возможностью поиска требуемых медицинских услуг на медицинском портале и возможностью записи на прием к необходимому врачу.

Пользователю приложения предлагается зарегистрироваться в информационной системе медицинского портала через мобильное приложение. Зарегистрированный пользователь может войти в систему, которая представлена как личный кабинет. В личном кабинете пользователь может записаться на прием к необходимому врачу пройдя цепочку выбора от необходимого направления медицины до выбора даты и времени прием конкретного специалиста. Записавшись на прием, пользователь может настроить время уведомления и способ уведомлений в меню настроек, а также удалить свою запись. Все уведомления отображаются на главном экране приложения. Помимо просмотра текущих записей пользователя возможен просмотреть прошедшие записи в архиве. Так же личный кабинет позволяет найти необходимые лечебные поликлинические учреждения, посмотреть специалистов и ознакомиться с прайсами медицинских услуг. В настройках профиля возможно задать ФИО, телефон, адрес электронной почты и номер медицинского полиса.

1.4 Обзор существующих решений

Из существующих решений мобильных приложений по работе с расписанием медицинских учреждений можно выделить несколько наиболее функциональных. Непосредственно между собой приложения данной категории сложно сравнивать, так как в основном такие приложения разрабатываются индивидуально для каждого медицинского учреждения или медицинской информационной системы крупных городов, к которой прикрепляются несколько поликлиник. Но несмотря на это возможно рассмотреть функциональность, юзабилити интерфейса, особенности и недостатки подобных мобильных приложений со стороны поставленной для них задачи.

1.4.1 «ЕМИАС»

Приложение ЕМИАС, разработанное компанией ОАО «Электронная Москва» предназначено для электронной записи на прием в медицинские учреждения города Москвы, позволяет использовать поиск по поликлиникам и аптекам, может хранить информацию о пациенте, список рецептов, а также имеет функцию автоматического напоминания о приеме.[2]

Интерфейс приложения «ЕМИАС» представлен на рисунке 1.

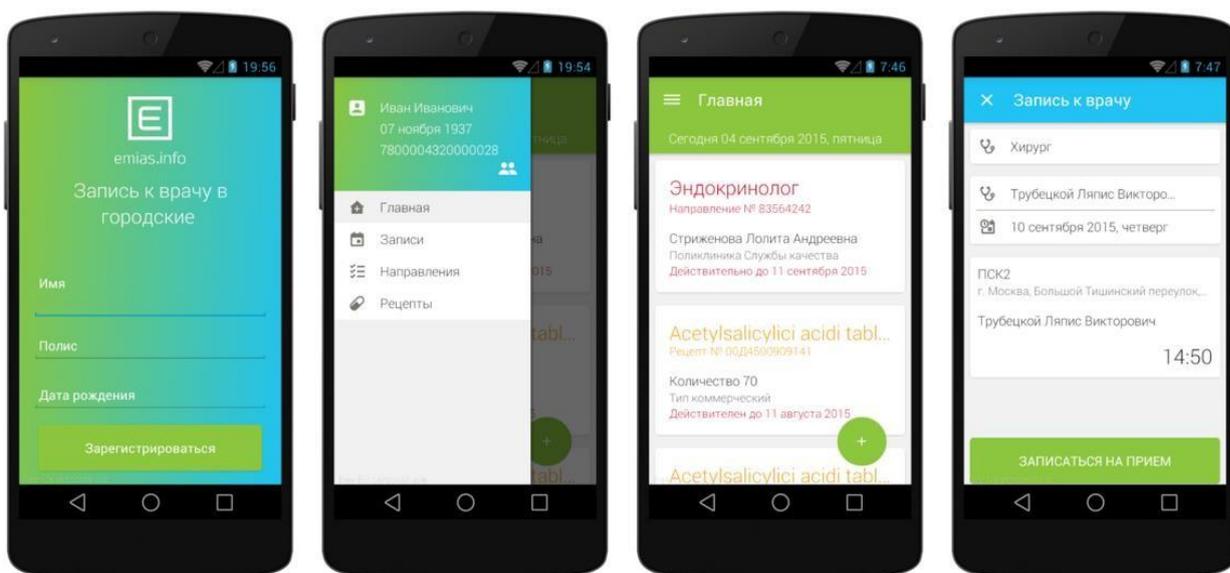


Рисунок 1 – Интерфейс приложения «ЕМИАС»

Главный экран приложения представляет собой список записей на прием пациента. У каждой записи есть время, дата, а также название медицинского учреждения. Можно посмотреть все записи. Также на главном экране расположены кнопки «Записаться к врачу», «Наблюдения» позволяющая пациенту записывать свои наблюдения, и кнопка «Карта», где пользователь может посмотреть на карте ближайшие поликлиники и аптеки. Вверху главной страницы располагается блок «Совет дня», в котором отображаются советы медицинской тематики. При нажатии на запись из блока

«Мои записи» появляется окно с полной информацией по записи. Так же в приложении можно настроить уведомления о записях на прием, посмотреть историю посещений.

1.4.2 «К врачу»

Приложение «К Врачу», позволяет самостоятельно записать к любому врачу на территории Вашего региона как взрослого, так и ребенка, просмотреть историю посещений, подписаться на уведомления. Приложение позволяет сэкономить время и исключить необходимость посещения поликлиник для записи на прием. Кроме этого, есть возможность получить справочную информацию обо всех поликлиниках региона, контакты и адреса

клиник. Приложение доступно на территории Пермского края и является клиентом к порталу k-vrachu.ru ООО «СВАН».[2]

Интерфейс приложения «К Врачу» представлен на рисунке 2.

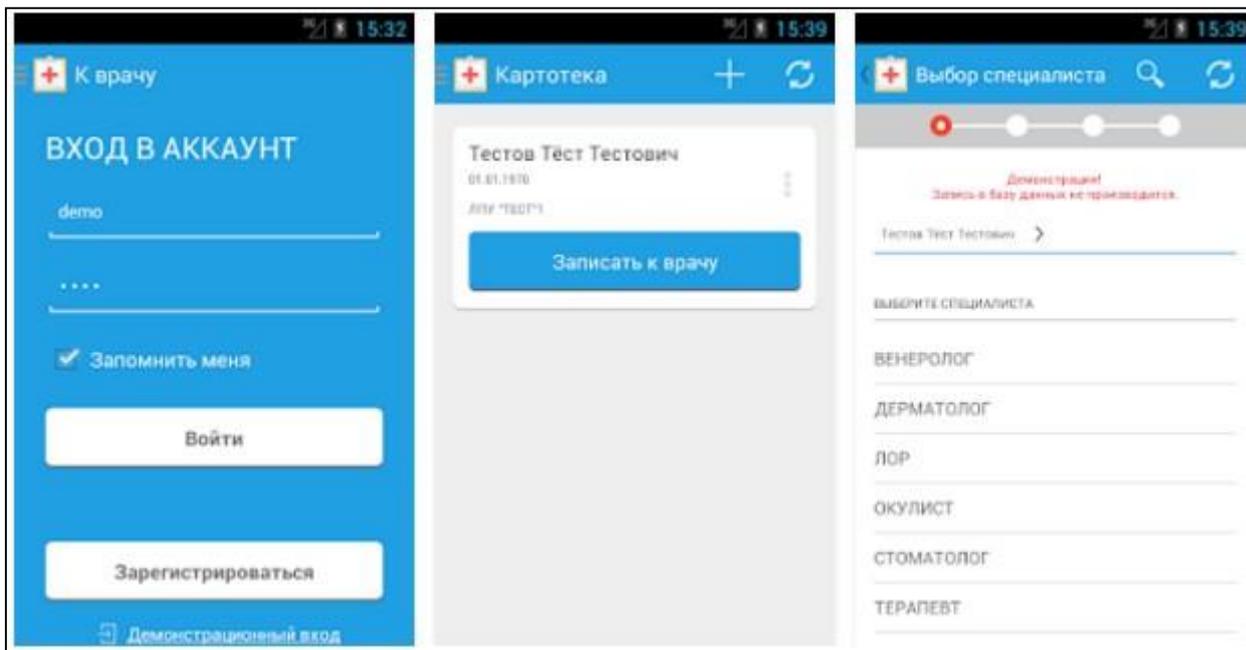


Рисунок 2 – Интерфейс приложения «К врачу»

Главный экран приложения представляет собой так называемую картотеку, где можно добавлять данные для разных людей, что позволяет пользователю записать на прием не только себя, но и своих родственников. Приложение позволяет записаться на прием пройдя цепочку выбора специальности, поликлиники, специалиста и закончить запись выбором даты и времени приема. При записи на прием возможен выбор даты только в диапазоне ближайших трех недель. В настройках профиля можно указать тип уведомления для всех записей по СМС, либо по Email. Также возможно просмотреть список доступных медицинских учреждений Пермского края

1.4.3 «Медицина 72»

Официальное приложение Правительства Тюменской области для записи на прием к врачу в медицинские учреждения города Тюмени и юга Тюменской области. Приложение Медицина 72 позволяет:

- записаться на прием в несколько кликов;
- записать на прием родных и близких;
- хранить личные данные;
- просматривать информацию о медицинских учреждениях;
- установить напоминание о приеме и добавить событие о приеме в календарь мобильного устройства.[2]

Интерфейс приложения «Медицина 72» представлен на рисунке 3.

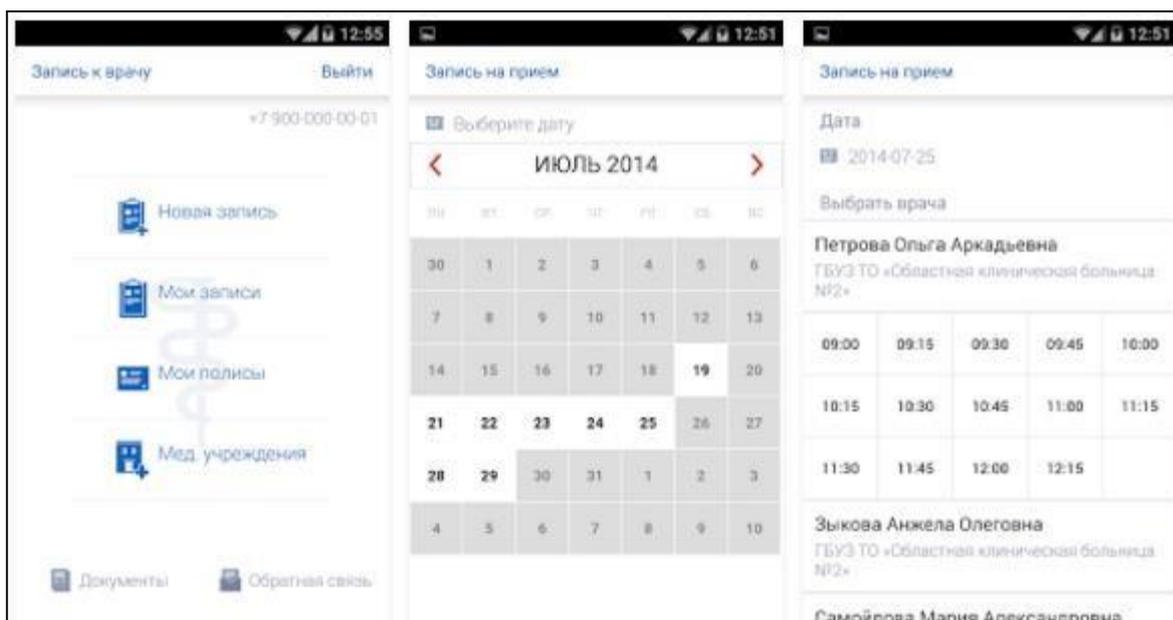


Рисунок 3 – Интерфейс приложения «Медицина 72».

Главный экран приложения представляет собой меню, где можно выбрать один из вариантов «Новая запись» которая запускает экран с календарем, где можно выбрать дату приема, а далее можно выбрать поликлинику, конкретного специалиста и записаться на прием.

В окне «Мои записи» можно установить СМС-уведомление о посещении и удалить записи.

В окне «Мои полисы» можно сохранить полис для любого человека, что позволяет записать на прием родственника или близкого, у которого нет приложения на телефоне, хотя уведомление о посещении приходит только на телефон зарегистрированного в приложении пользователя.

В окне «Послы жизни» можно вести историю измерений давления, пульса, глюкозы и холестерина, а также привязать приложение к приложению другого пользователя, что бы тот выступал в роли попечителя и получал уведомления о измерениях.

1.5 Анализ популярных мобильных операционных систем

На данный момент по версии американской компании International Data Corporation (IDC), проводящей маркетинговые исследования по всему миру, на рынке мобильных операционных систем ведущими являются следующие платформы: iOS, Android, Windows Phone.[3]

IOS — это закрытая операционная система от компании Apple. Используется исключительно в устройствах, разрабатываемой компанией Apple с 2007 года. В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple и с самой первой версии используется на мобильных процессорах архитектуры ARM. В связи с тем, что данная компания ориентирована на поддержку только своих продуктов, то разработка под IOS имеет свои минусы, наиболее существенным из которых является невозможность скомпилировать и отладить разрабатываемое приложения на таких наиболее распространенных операционных системах как Windows и Linux, необходимо наличие либо непосредственно техники Apple с предустановленной операционной системой OS X, либо использование виртуальной машины или же необходимо использовать сторонние сервисы для удаленной компиляции создаваемых

приложений.

Так же существует возможность разработки на Flash CS5.5 или других средах флеш разработки с помощью компилятора Flex, но при этом необходимо учитывать, что производительность будет невысока. Публикация приложений так же не возможна без сертификатов от виртуального магазина App Store, так же App Store, является единственным полностью легальным средством дистрибуции приложений. Стоит отметить, что так же возможно распространять приложение через неофициальные площадки приложений, но это требует разблокировки устройств, которая нарушает гарантию, вследствие чего, является непопулярной среди пользователей данной платформы. К плюсам этой системы можно отнести стабильность и гарантированную работу приложений на современных устройствах. Неподобающие по качеству или контенту приложения не допускаются к распространению в App Store. По версии IDC процент на рынке платформы iOS на 2015 год составляет 13,9%. [3,4]

Windows Phone – молодая операционная система, разработанная компанией Microsoft, которая вышла 11 октября 2010 года. Данная операционная система является преемником Windows Mobile, с полностью переработанным интерфейсом и не совместимая со своим предшественником. Платформа Windows Phone предлагает на выбор два фреймворка для разработки приложений:

- Silverlight, который поддерживает разработку приложений, управляемых событиями, на основе языка XAML;

- XNA Framework, который поддерживает разработку циклических игр.

Для разработки приложений для Windows Phone необходим Visual Studio 2010 с Service Pack 1 редакции Professional или выше. [5]

По версии IDC доля на рынке платформы Windows Phone составляет 2,6%. [3]

Android OS — открытая операционная система от Google. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать (но не отлаживать) библиотеки и компоненты приложений, написанные на C и других языках. Для дистрибуции приложений Google запустила онлайн-магазин Google Play. Для распространения приложения через онлайн-магазин требуется учетная запись, но также возможно распространять приложения любым удобным для разработчика способом как просто передавать установочный .apk файл так и через сторонние онлайн-магазины. По версии IDC доля на рынке Android составляет 82,8% на начало 2015 года, что делает ее безоговорочным лидером.[3,6]

1.6 Анализ существующих сред разработки

Для разработки приложений под операционную систему Android можно использовать множество сред разработки, рассмотрим на сегодняшний день самые известные из них: Eclipse, IntelliJ IDEA, Android Studio, Xamarin Studio.

Eclipse — открытая интегрированная среда разработки приложений, изначально разрабатываемая фирмой IBM. Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation. Особенности платформы Eclipse:

- 1) кроссплатформенность - работает под операционными системами Windows, Linux, Solaris и Mac OS X;
- 2) используя Eclipse можно программировать на множестве языков, таких как Java, C и C++, PHP, Perl, Python, Cobol и других;
- 3) является фреймворком для разработки других инструментов и предлагает обширный набор API для создания модулей;
- 4) используя подход RCP (Rich Client Platform) Eclipse является инструментом для создания практически любого клиентского программного обеспечения.[7]

IntelliJ IDEA — коммерческая интегрированная среда разработки приложений. Разработана фирмой JetBrains и позволяет разрабатывать приложения на многих языках программирования в частности Java, JavaScript, Python. Основным преимуществом данной среды разработки перед Eclipse является понимание контекста. Среда индексирует весь проект, анализирует все, что в нем есть и строит синтаксическое дерево. Рассмотрим аспекты, в которых выражается умение «понимать контекст»:

1) при автозаполнении благодаря «пониманию контекста» данная среда анализирует положение курсора и предлагает актуальный вариант для заполнения;[8]

2) в плане рефакторинга IDEA также имеет преимущество над Eclipse, так как использует более продвинутые методы рефакторинга.[8]

Самым заметным недостатком IntelliJ IDEA служит ее коммерческая направленность и урезание функций в бесплатной версии.

Android Studio — среда разработки, созданная компанией Google. Основой для Android Studio стала платформа IntelliJ от компании JetBrains. Отличием от IDEA является изначальная направленность на разработку приложений для операционной системы Android.[9]

Xamarin Studio – это кроссплатформенная интегрированная среда разработки, которая работает как на Mac OS X, так и на Windows. Позволяет разрабатывать приложения для трех основных платформ, включая Android с использованием языка C#. Это позволяет использовать полный набор языковых конструкций и возможностей свойственных C# и иметь полный доступ к SDK платформы и механизму создания UI, получая в результате приложение, которое ничем не отличается от нативных приложений и не уступает им в производительности.[10]

Так же Xamarin выпустила расширение, позволяющее использовать свои уникальные функции по разработке в других средах. Например, установив такое расширение в Visual Studio от Microsoft, среда приобретает все возможности разработки в Xamarin включая все особенности Visual

Studio.[10,11]

Основные достоинства разработки с использованием Xamarin:

1) позволяет разрабатывать приложения для разных мобильных платформ с использованием C# и .Net, что исключает необходимость изучать различные фреймворки и языки для разработки для разных мобильных платформ;[10]

2) позволяет использовать ранее написанный код для разных платформ практически его не изменяя;[10]

3) так как UI описывается только в одном месте, то приложения под разными системами будут выглядеть очень похоже. Что может быть важно, например, в корпоративных разработках.[10]

1.7 Обоснование выбора мобильной платформы

При выборе мобильной платформы основными факторами служат:

- доля на рынке мобильных устройств;
- удобство и возможности при разработке;
- сильные и слабые стороны самой платформы.

В результате анализа мобильных платформ выбор был сделан в пользу Android так как он занимает 82,8% доли мирового рынка; комплект средств разработки Android SDK имеет широкий набор возможностей при создании приложений; платформа позволяет портировать компоненты, написанные на C-подобных и других языках; для создания приложений для платформы есть возможность выбрать среду разработки наиболее удобной для разработчика.

1.8 Обоснование выбора среды разработки

Основным критерием выбора интегрированной среды разработки является удобство разработки и отладки приложения. Учитывая преимущества, указанные выше, и тот факт, что разрабатываемое приложение рассчитано на

мобильные устройства, была выбрана Visual Studio с использованием технологии Xamarin.

1.9 Техническое задание

1.9.1 Введение

1.9.1.1 Наименование программы

Наименование программы – «Личный кабинет пациента»

1.9.1.2 Краткая характеристика области применения

Система предназначена для организации личного кабинета с возможностью поиска требуемых медицинских услуг на медицинском портале.

1.9.1.3 Цели разработки

Целью разработки является создание мобильного приложения для решения задач работы с расписанием медицинского учреждения для ООО «Медицинские облачные технологии».

1.9.2 Основание разработки

1.9.2.1 Основание для проведения разработки

Основанием для проведения разработки является задание на разработку мобильного приложения в рамках задания на выпускную квалификационную работу на получение квалификации «бакалавр», которая входит в состав учебного курса для студентов Томского политехнического университета кафедры АИКС по направлению Информатика и вычислительная техника по профилю Программное обеспечение средств вычислительной техники и автоматизированных систем.

1.9.3 Назначение разработки

1.9.3.1 Функциональное назначение

Функциональным назначением данного мобильного приложения является организация электронного личного кабинета с возможностью поиска требуемых медицинских услуг на медицинском портале.

1.9.3.2 Эксплуатационное назначение

Использовать данное мобильное приложение могут пользователи на любых мобильных устройствах с предустановленной операционной системой Android. Для работы с мобильным приложением нет необходимости в каких-либо специфических знаниях, поэтому данным приложением может пользоваться любое заинтересованное лицо.

1.9.4 Требования к программе и программному изделию

1.9.4.1 Требования к эргономике и технической эстетике

Взаимодействие пользователей с программным приложением должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс программы должен быть интуитивно понятным и удобным для любых пользователей и не должен быть перегружен текстовыми и графическими элементами, а также должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной форме. Прием команд программой и отображение результатов должны выполняться в интерактивном режиме. Интерфейс должен быть рассчитан для работы на мобильных устройствах с сенсорным управлением, то есть иметь элементы управления достаточного размера для удобного управления с помощью пальцев. Все надписи экранных форм, а также сообщения, выдаваемые пользователю должны быть на русском языке.

Все экранные формы должны разрабатываться с учетом унификации:

- все элементы управления, имеющие единое функциональное назначение, используемые на различных экранных формах должны иметь одинаковый интерфейс;

- внешнее поведение однотипных элементов интерфейса должно реализовываться одинаково.

При разработке интерфейса приложения необходимо учитывать особенности по организации дизайна Android приложений рекомендованные компанией Google.

1.9.4.2 Требования к защите информации от несанкционированного доступа

Система должна обеспечивать защиту от несанкционированного доступа с помощью следующих средств:

- идентификация пользователя посредством использования идентификатора учетной записи пользователя и пароля;

- проверка полномочий доступа пользователя при работе с системой;

- доступ пользователь должен быть разграничен на уровне информационных массивов.

Вход в систему должен осуществляться с использованием «слепых» паролей, то есть паролей, графически отображающихся как набор однотипных символов, а также при авторизации пользователя должно производиться хэширование пароля для передачи его серверу.

1.9.4.3 Требования к составу выполняемых функций

Разрабатываемое мобильное приложение должно обеспечивать выполнение следующих функций:

- 1) регистрация пользователя на медицинском портале;
- 2) настройка адреса сервера при входе с систему;
- 3) добавление записи к врачу;

- 4) хранение личных записей пользователя;
- 5) настройка времени и типа напоминания о приеме индивидуально для каждой записи;
- 6) автоматическое напоминание о приеме;
- 7) удаление записи на прием;
- 8) при записи на прием обеспечить выбор: специальности медицинских работников, медицинского учреждения, специалиста, даты и времени приема;
- 9) при просмотре сетки расписания специалиста обеспечить выбор произвольной даты и возможность быстрого перехода на следующий день;
- 10) просмотр архивных записей пользователя;
- 11) редактирование профиля пользователя;
- 12) отправка сообщений разработчикам;
- 13) отображения информации о клиниках.

1.9.4.4 Требования к платформе развертывания

Разрабатываемое приложение должно быть совместимо с платформой Android не ниже версии 4.4.

1.9.4.5 Требования к формату и протоколу передачи данных

Приложение должно обмениваться данными с сервером через его API посредством протокола HTTP и использовать формат для передачи данных JSON.

2 ПРОЕКТИРОВАНИЕ МЕДИЦИНСКОГО ПРИЛОЖЕНИЯ

Согласно разработанному техническому заданию было спроектировано мобильное приложение, удовлетворяющее все указанные в техническом задании требования.

Исходя из разработанного технического задания составлена диаграмма вариантов использования, которая позволит наглядно выделить все возможности программы и взаимодействие с ними пользователя, а также позволит описать проектируемое приложение на концептуальном уровне. Диаграмма вариантов использования представлена на рисунке 1.

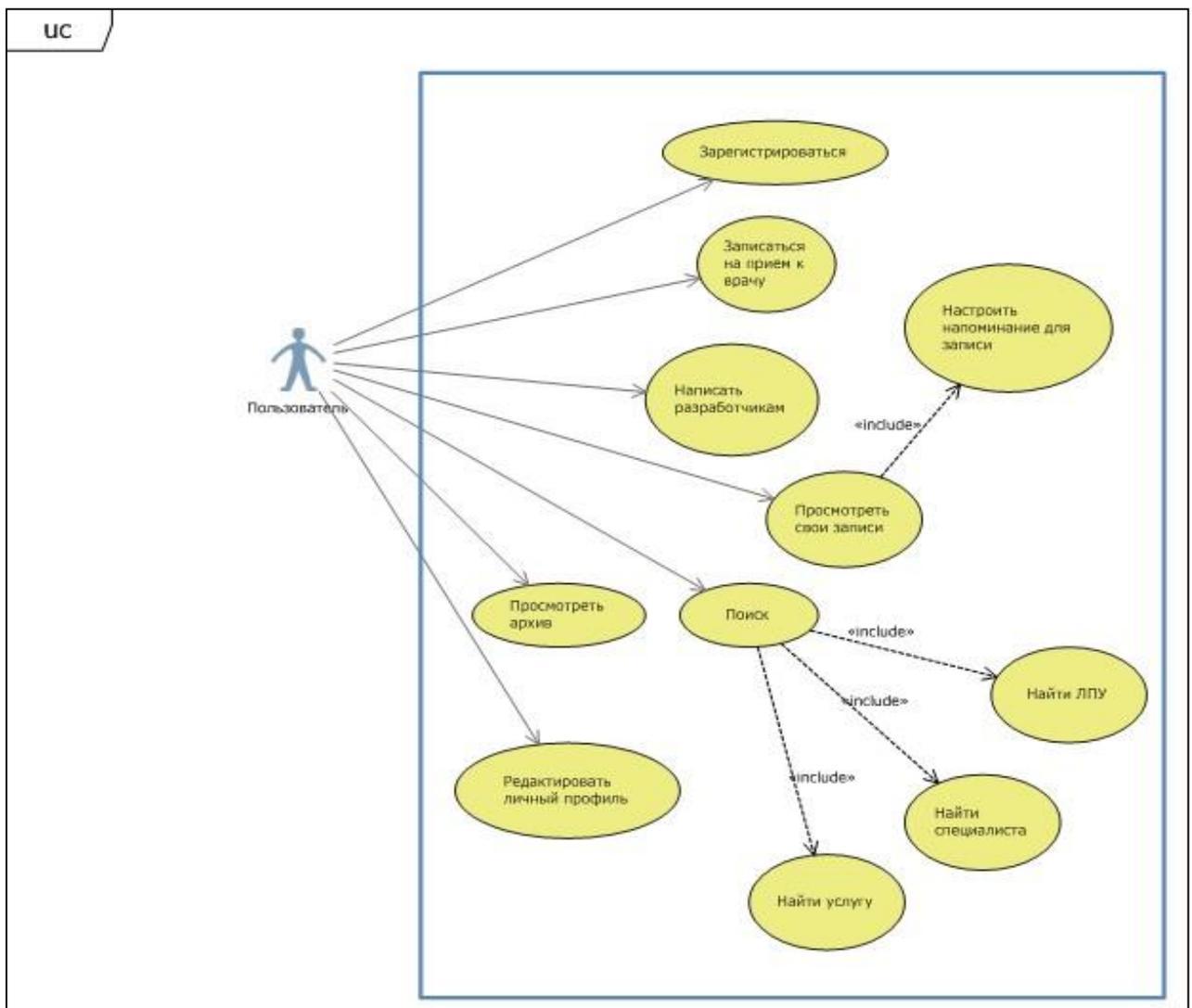


Рисунок 4 – Основная диаграмма вариантов использования.

Если рассматривать диаграмму вариантов использования, представленную на рисунке 1, более подробно, то получим соответствующие диаграммы вариантов использования, представленные на рисунках 2 и 3.

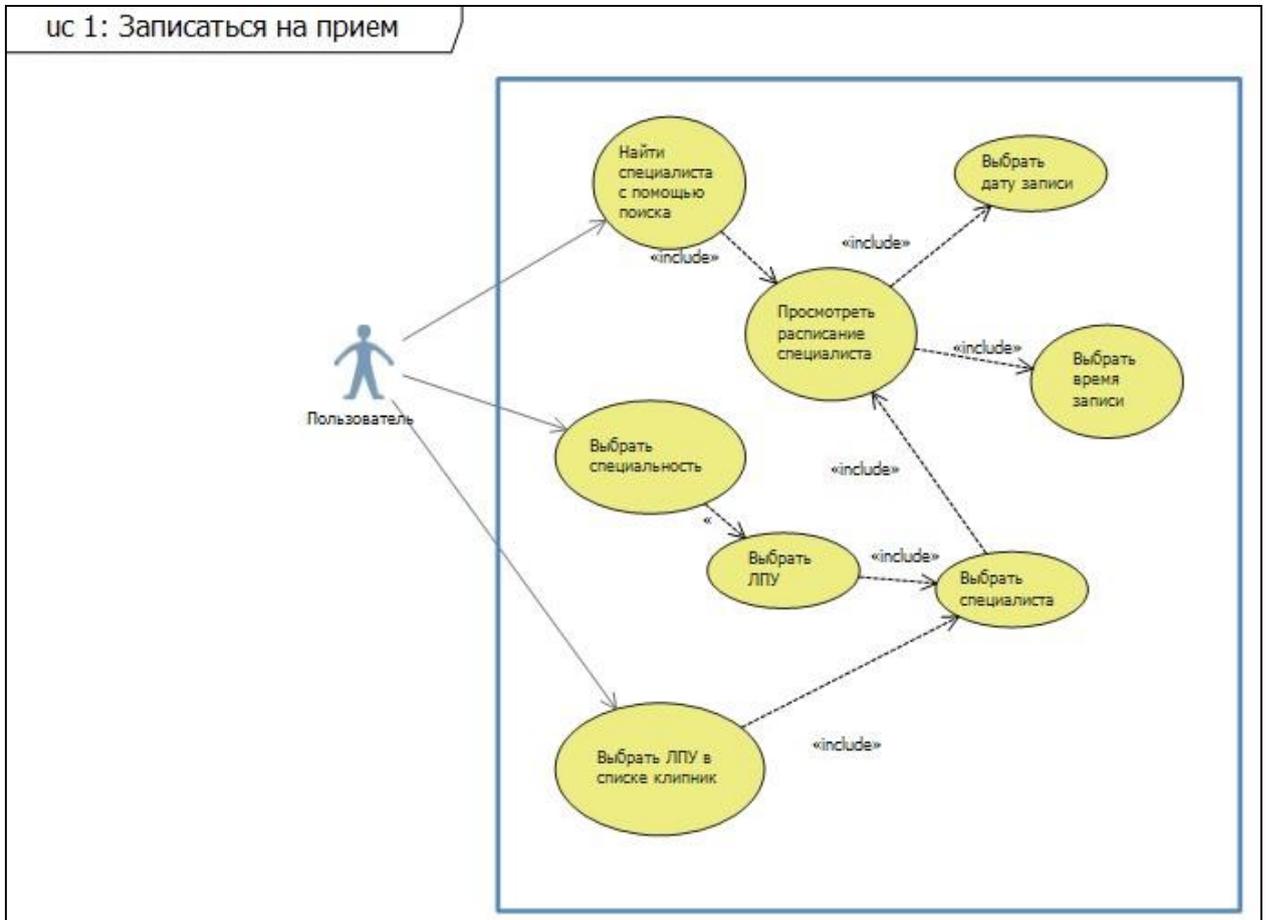


Рисунок 5 – Подробная диаграмма для варианта использования «Записаться на прием»

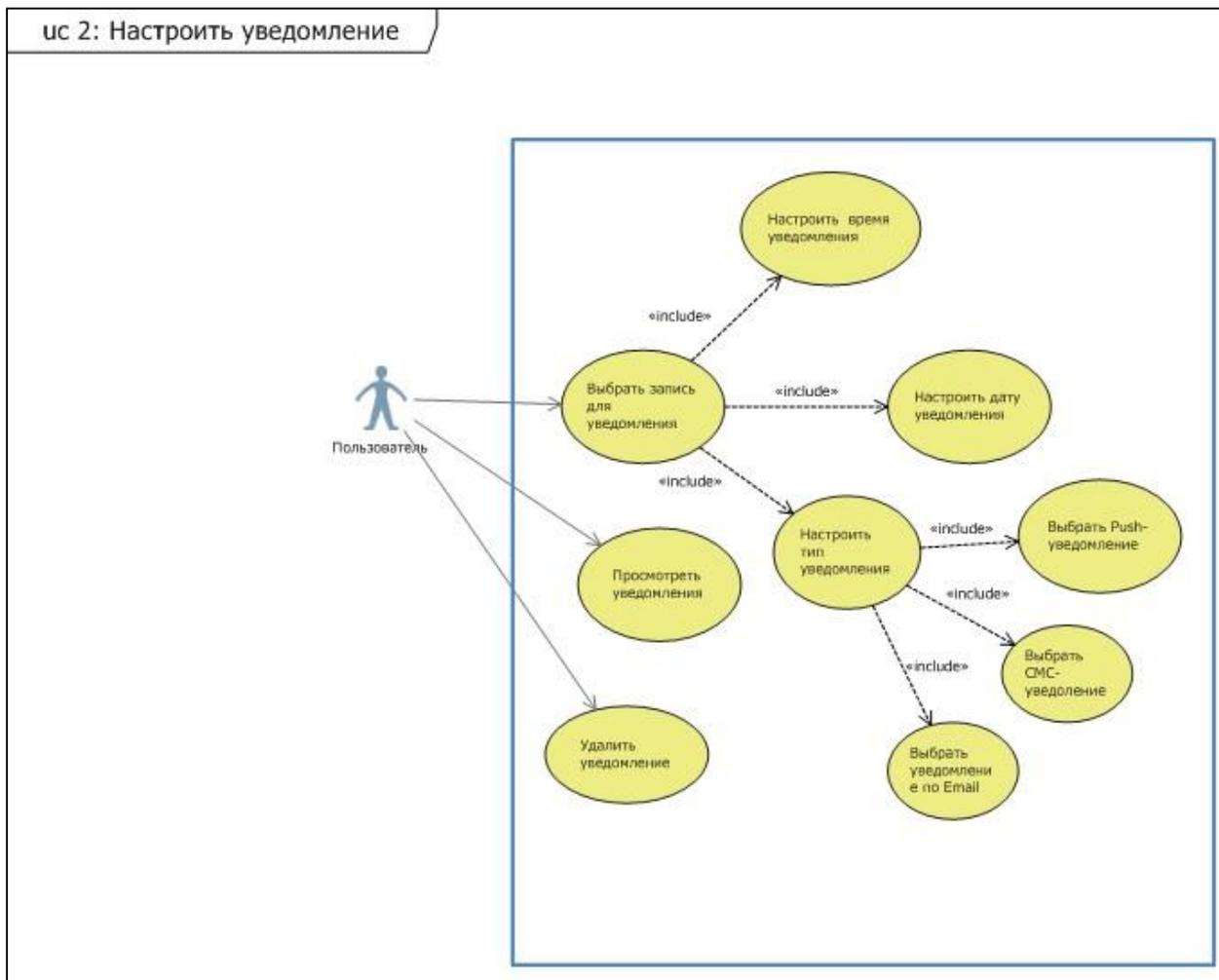


Рисунок 6 – подробная диаграмма для варианта использования «Настроить уведомление»

В результате разработанных диаграмм вариантов использования система была четко отделена от ее окружения, были определены варианты взаимодействия пользователя с системой и ожидаемый функционал системы.

2.1 Создание прототипа

Исходя из разработанных диаграмм вариантов использования и требований, представленных в техническом задании, был создан прототип будущего мобильного приложения. Прототип позволяет определить

правильность структуры приложения, его функциональность и, в целом, концепцию приложения. Прототип создавался в программном комплексе Axure RP, который предназначен для создания шаблонов и спецификаций веб-сайтов и приложений. Axure RP позволяет запрограммировать поведение кнопок, текстовых полей, панелей и других виджетов, что в свою очередь делает макет максимально приближенным к окончательному результату. Также данный программный комплекс позволяет разместить полученный прототип на сервере Axure, что дает возможность показывать результат работы, просто обмениваясь ссылкой на html страницу с макетом.

Прототипы экранов авторизации и регистрации приведены на рисунке 7.

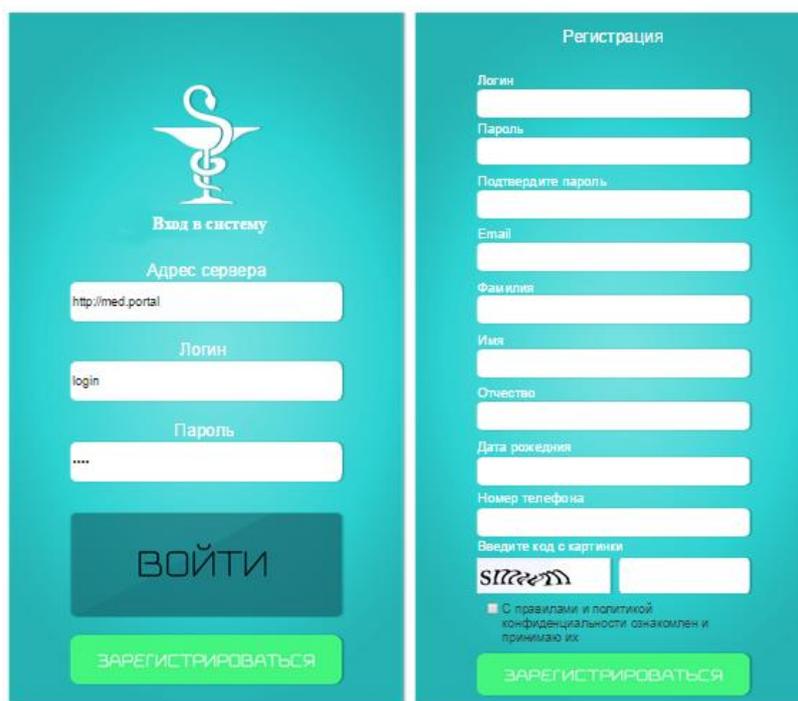


Рисунок 7 – Прототипы экранов авторизации и регистрации

Для просмотра записей к специалистам медицинских учреждений и настройки уведомлений для каждой записи разработаны прототипы экранов «Мои записи» и «Уведомления», которые представлены на рисунке 8.

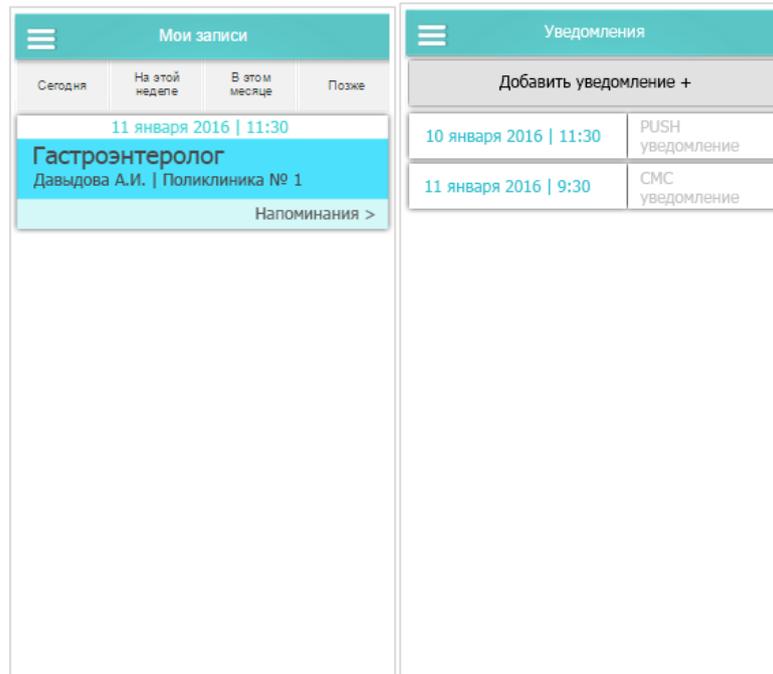


Рисунок 8 – Прототипы экранов «Мои записи» и «Уведомления»

Для записи к специалисту используется цепочка экранов «Выбор ЛПУ», «Выбор специальности» и «Выбор специалистов», прототипы которых приведены на рисунке 9.



Рисунок 9 – Прототип цепочки экранов выбора специалиста

Для просмотра общей информации о медицинском учреждении и информации о специалисте, с последующей записью на прием используются экраны «Страница медучреждения» и «Страница специалиста», прототипы которых приведены на рисунке 10.

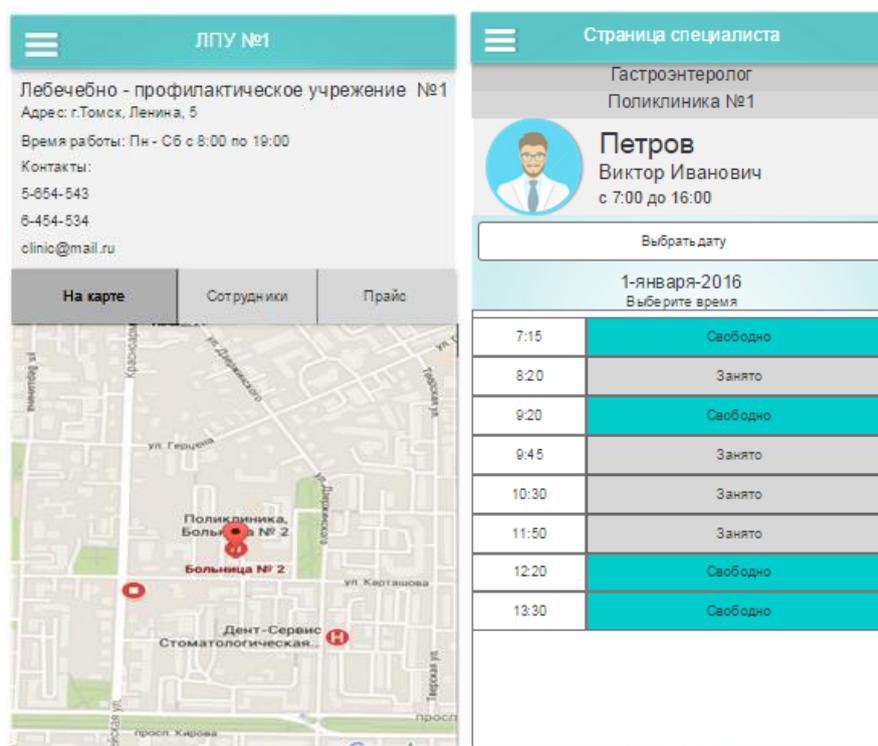


Рисунок 10 – Прототипы экранов «Страница медучреждения» и «Страница специалиста»

Для настройки личного профиля пользователя и поиска специалистов, медучреждений и услуг разработаны прототипы экранов «Мой профиль» и «Поиск», которые приведены на рисунке 11.

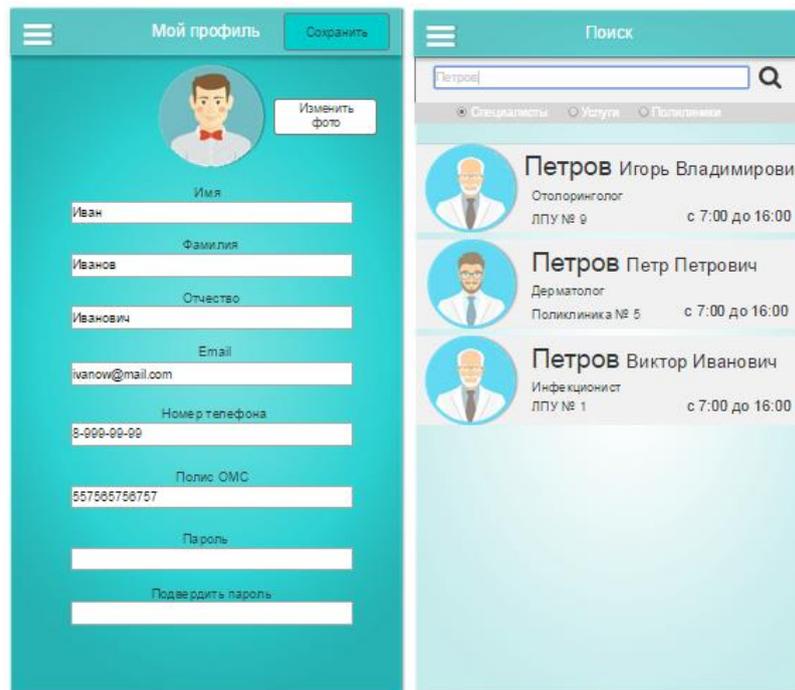


Рисунок 11 – Прототипы экранов поиска и настройки профиля

Для навигации в приложении был спроектирован прототип бокового меню навигации, который представлен на рисунке 12.

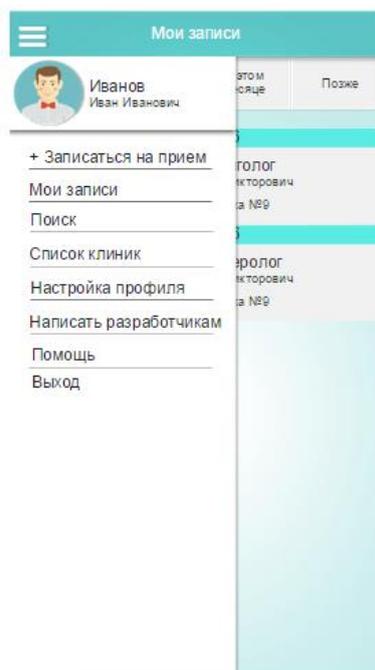


Рисунок 12 – Прототип меню навигации

2.2 Проектирование приложения на платформе Android

При проектировании мобильного приложения для платформы Android необходимо учитывать некоторые свойства, особенности платформы и жизненный цикл программы. Разработка приложения на Xamarin подразумевает, что весь основной код программы будет строиться на языке C#, а визуальный интерфейс будет разработан на расширенном языке разметки XAML.

Базовые компоненты, из которых состоит приложение для Android представляются классами `android.content.Context` и `android.app.Activity`, или активность. Класс `Context` используется для управления специфичными для приложения ресурсами и конфигурацией. С помощью класса `Context` мы можем получить различные строковые, графические ресурсы, получить доступ к системным сервисам, управлять базами данных SQLite, управлять файлами и каталогами приложения.

Основной же класс, на котором будет строиться разрабатываемое приложение это `Activity`. `Activity` (далее активность) представляет собой класс приложения, который предназначен для управления логикой слоя экрана, разметка которого написана на языке XAML, и с которым пользователи могут взаимодействовать. Каждая активность может использоваться как для одного слоя, так и для нескольких, но второй подход не всегда целесообразен, так как это значительно запутывает код и загружает программу.

В результате создания прототипа приложения было определено, что проектируемое приложение будет иметь шестнадцать экранов, это означает, что для каждого экрана должен быть разработан свой слой с XAML разметкой и к каждому слою прикрепить свой управляющий класс.

Так как в проектируемом мобильном приложении присутствуют элементы управления, которые повторяются практически на всех экранах, это приводит к тому, что код для этих элементов необходимо дублировать на каждый экран, что весьма нецелесообразно и нарушает принцип модульности при-

ложения. Для решения этой проблемы было принято решение использовать класс `android.app.fragment.Fragment` (далее фрагмент) – это класс, аналогичный классу `Activity`, но позволяет использовать его не как отдельный экран, а как часть пользовательского интерфейса внутри какой либо `Activity` со своей логикой и разметкой. Смысл использования `Fragment` в данном приложении состоит в том, чтобы на главную `Activity` поместить все основные элементы управления и логика для управления фрагментами, а в разметку добавить контейнер `FragmentManager` – это слой в котором будут отображаться фрагменты каждого экрана.

3.2.1 Основные элементы управления

Основной экран содержит панель действий, на которой располагаются кнопки вызова бокового меню и обновления страницы и непосредственно меню навигации. За логику основного экрана отвечает класс `MainActivity`, который унаследован от класса `ActionBarActivity` и работает со слоем `Main`. Данный экран является корневым, так как агрегирует в себе фрагменты всех остальных экранов, кроме экрана авторизации и регистрации.

Для работы с меню навигации и панелью действий используются элементы `android.support.v7.widget.toolbar` и `android.support.v4.widget.DrawerLayout`, которые располагаются на слое `Main`.

Для отображения элементов меню на XAML разметке используется элемент `ListView` внутри элемента `android.support.v4.widget.DrawerLayout`. Непосредственно на активности вызов панели действий происходит с помощью метода `setSupportActionBar`.

Пункты навигационного меню хранятся в коллекции `List` и взаимодействуют с элементом `ListView` на слое с помощью стандартного адаптера, который отвечает за создание `View`-компонента для каждой единицы данных в коллекции. При событии нажатия на пункт навигационного меню вызывается ме-

тод `LeftDrawer_itemClick`, который в соответствии с позицией нажатого пункта меню выполняет соответствующее пункту действие.

Для работы с фрагментами в классе `MainActivity` используются следующие основные экземпляры классов:

- 1) `SupportFragmentManager` с помощью которого происходит взаимодействие между объектами;
- 2) `FragmentTransaction` с помощью которого выполняются транзакции фрагментов, такие как добавление, удаление, замена на `Activity`.

Далее, при помощи методов класса `SupportFragmentManager` добавляются все существующие фрагменты. С помощью определенного в `MainActivity` метода `ShowFragment` производится отображение переданного в параметре метода фрагмента. В методе `OnBackPressed` производится выборка из стека предыдущих фрагментов и отображение их на экране при нажатии на системную кнопку «Назад». Для выбора отображающихся фрагментов из самих фрагментов используется метод `selectFragment`, которому в параметре передается строка с тэгом необходимого фрагмента и после чего с помощью метода `ShowFragment` фрагмент отображается на экране. Для отображения фрагментов непосредственно на слое используется элемент `FrameLayout`.

Данный экран объединяет в себе следующие фрагменты:

- 1) `PoliclinicFragment` – фрагмент выбора медицинского учреждения, который представляет собой список, содержащий в себе наименование поликлиник их адреса;
- 2) `SelectSpecialtyFragment` - фрагмент выбора специальности представляет собой список, который содержит наименование специальности и количество специалистов этой специальности;
- 3) `SelectEmployeeFragment` - фрагмент экрана выбора специалиста представляет собой список, который содержит ФИО специалиста и его фотографию.
- 4) `EmployeePageFragment` – фрагмент страницы специалиста, где содер-

жится основная информация о специалисте, его фото и возможность выбора даты и времени приема;

5) PolyclinicPageFragment – фрагмент страницы медицинского учреждения, содержащая общую информацию о медицинском учреждении, контактные данные, список сотрудников;

6) MyNotesFragment – фрагмент страницы записей пользователя;

7) MedicalRecordFragment – фрагмент экрана персональной электронной медицинской карты;

8) NotificationFragment – фрагмент страницы уведомлений, содержащий список всех уведомлений на события, добавленных пользователем;

9) SearchFragment – фрагмент страницы поиска;

10) ProfilPageFragment – фрагмент страницы редактирования профиля пользователя;

11) FeedbackFrarment – фрагмент экрана обратной связи с разработчиками;

12) HelpFragment – фрагмент экрана помощи.

2.3.2 Работа со списками

Исходя из того, что разрабатываемое приложение должно работать в основном со списочной информацией, то необходимо организовать правильное отображение этих списков на экране. Для этой цели используются экземпляры класса ArrayAdapter, которые отвечают за создание View- компонента для каждой единицы данных в коллекции и являются связующим звеном между информацией и ее отображением. В случае, когда для элементов списка на экране необходимо задать нестандартную структуру и добавить дополнительные элементы управления, то используются классы, реализующие базовый класс BaseAdapter, позволяющие применять структуру отдельного слоя экрана как элемент списка. Так как фотографии, отображаемые в списке сотрудников

приходят с сервера и хранятся в строках формата base64, то адаптер списка сотрудников с помощью метода `ConvertImage` преобразует получаемую строку в объект `Bitmap` пригодный для отображения на экране.

2.3.3 Работа с уведомлениями

Уведомление — это сообщение, которое может быть выведено на экран за пределами обычного пользовательского интерфейса приложения. Уведомления имеют следующую структуру:

- 1) небольшой значек, который задается методом `setSmallIcon()`;
- 2) заголовок, который задается методом `setContentTitle()`;
- 3) подробный текст, который задается методом `setContentText()`.

Так как одной из функций приложения является уведомление пользователя о последующих приемах в медицинских учреждениях, то необходимо организовать правильную работу уведомлений. Для решения этой задачи используется класс `NotificManager`, который содержит в себе следующие методы:

- 1) `ConvertTime` – метод предназначенный для получения разницы от времени приема до установленного времени и конвертации времени из типа `DateTime` в `long`, для использования его в методе `NotificationCreate`;

- 2) `NotificationCreate` – метод, предназначенный для создания уведомлений. Для работы с уведомлениями использует объекты:

- `Notification.Builder`, в котором задаются параметры уведомления;
- `Notification.Builder.Build()` – метод, возвращающий объект `Notification`, содержащий заданные пользователем спецификации;
- `NotificationManager.notify()` – метод, передающий в систему объект `Notification`.

- 3) `NotificationDelete` – метод, предназначенный для удаления

уведомлений.

2.3.4 Работа с сетью

Так как для приложение работает с данными, расположенными на сервере, необходимо обеспечить возможность работы с API сервера. API – программный интерфейс, который обеспечивает доступ к данным и методам на сервере через специальные функции. Взаимодействие с API сервера осуществляется при помощи экземпляра класса `restConnect`, методы которого отправляют HTTP запросы сервисам и получают ответ в виде пакета, содержащий следующие поля:

- `ApiVersion` – версия интерфейса веб сервиса;
- `Status` – статус ответа;
- `ApiDescription` – описание интерфейса веб сервиса;
- `Data` – непосредственно запрашиваемые данные;
- `Message` – дополнительные сообщения от сервера

Класс `restConnect` содержит следующие методы:

1) `PostLogin` – предназначен для авторизацией пользователя. Принимает в качестве параметров логин и пароль пользователя, введенные в окне авторизации. Метод кэширует полученный пароль, из логина и кэша пароля формирует строку формата `json`, которую отправляет на сервер с помощью запроса `POST`. При удачной авторизации в ответе от сервера будет получен токен и индивидуальный номер пользователя `GuID`, которые используются для отправки других запросов;

2) `GetPoliclinic` – получает список поликлиник, содержащий название медицинского учреждения и его адрес;

3) `GetSpecialty` – получает список имеющихся в выбранной клинике медицинских специальностей;

4) `GetEmployee` – получает список специалистов в соответствии с

выбранными ранее клиникой и специальностью, в виде полного имени и фотографии;

5) `GetSchedule` – получает расписание специалиста с указанием свободных мест на прием;

6) `GetMyNostes` – получает записи пользователя ко врачу;

7) `PostRecord` – отправляет информацию о записи на прием;

8) `PostNotification` – отправляет информацию о напоминании;

9) `GetEmployeePage` – получает данные о странице специалиста;

10) `GetAllPoliclinic` – получает список всех поликлиник;

11) `GetMedicalRecords` – получает персональную электронную медицинскую карту пользователя;

12) `GetProfile` – получает профиль пользователя;

13) `PostProfile` – отправляет измененные данные пользователя;

14) `GetService` – получает список услуг;

15) `GetPoliclinicPage` – получает подробные данные о поликлинике.

Для работы с сетью используется экземпляр класса `HttpClient`.

2.3.5 Модели данных

При выполнении запроса к серверу приложение в ответе получает пакет с данными. Модель этого пакета реализована в виде класса `ReturnPackage`, который является универсальным и может содержать в себе любые типы данных. UML модель класса представлена на рисунке 7.

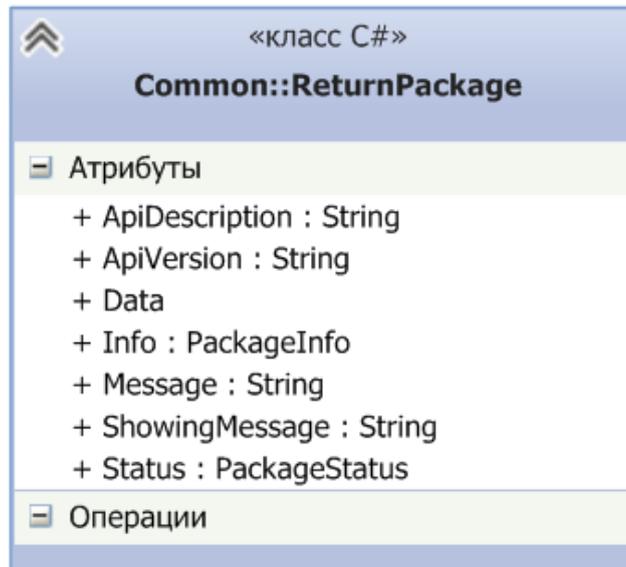


Рисунок 7 – UML модель класса ReturnPackage

Пакет данных содержит следующие поля:

- ApiDescription – содержит описание API, который вернул данный пакет;
- ApiVersion – содержит версию API, который вернул данный пакет;
- Data – поле, содержащее различные типы данных в зависимости от запрашиваемых данных;
- Info – содержит дополнительную информацию о пакете;
- Message – содержит строковое сообщение от сервера в зависимости от статуса полученного пакета;
- ShowingMessage – содержит сообщение для отображения в пользовательском интерфейсе в зависимости от статуса полученного пакета;
- Status – содержит статус ответа от сервера.

Для авторизации пользователя используется модель данных, которая реализована в виде класса AuthorisationResult, UML диаграмма которого представлена на рисунке 8.

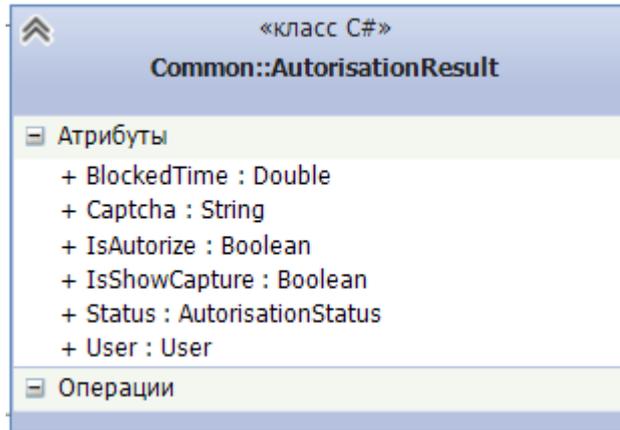


Рисунок 8 – UML диаграмма класса `AutorisationResult`

Класс `AutorisationResult` содержит поля о статусе авторизации в системе, если авторизация прошла успешно, то пакет будет содержать статус о удачной авторизации и в поле `User` будет храниться вся необходимая информация о пользователе, включая его `token` и индивидуальный номер типа `Guid`. При неудачной авторизации в поле `Captcha` будет содержаться строка с кодом, который пользователю необходимо ввести для повторной попытки авторизации.

Результат выполнения запроса на получение списка медицинских учреждений представляет модель данных, которую реализует класс `GetOrganisationResult`, UML диаграмма которого представлена на рисунке 9.

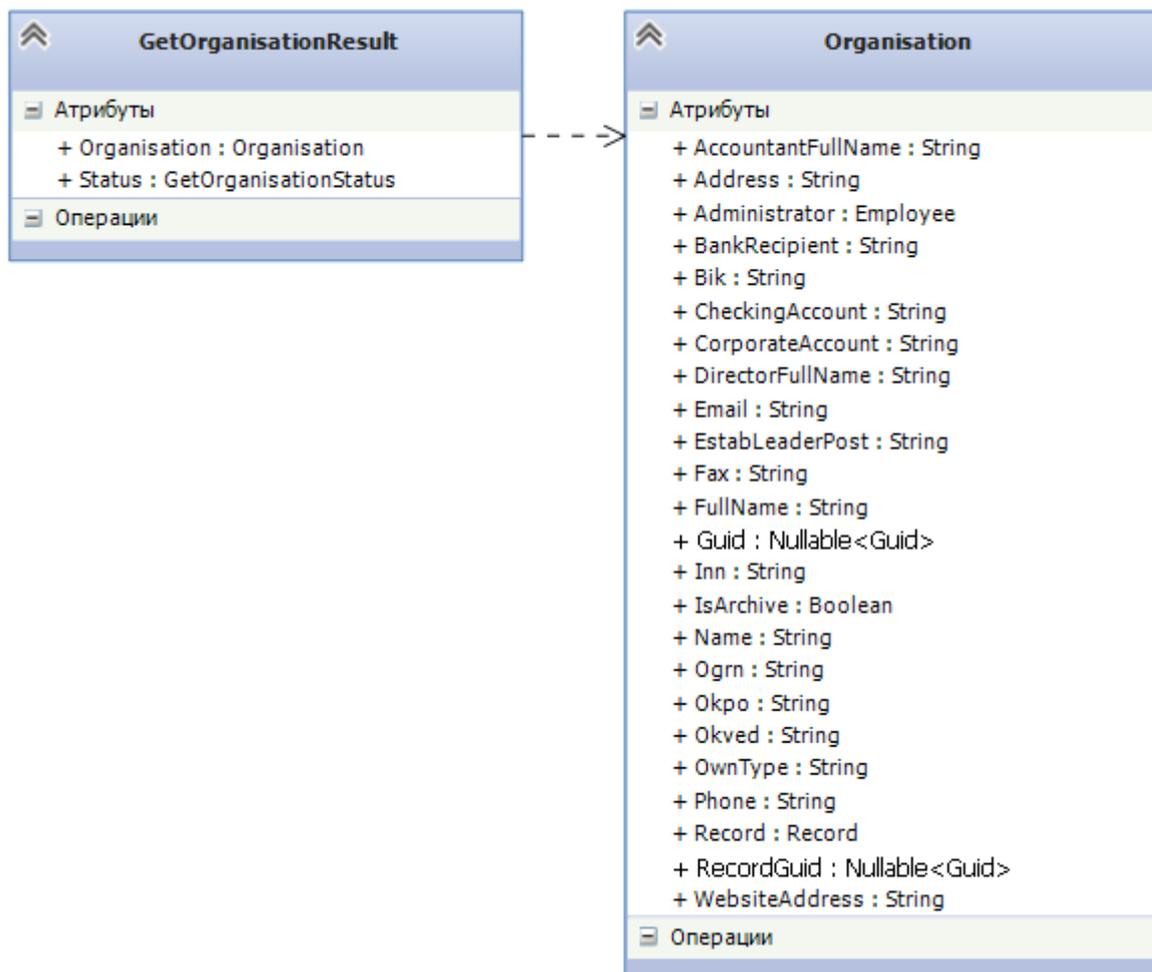


Рисунок 10 – UML диаграмма классов GetOrganisationResult и Organisation

Класс GetOrganisationResult содержит поля, содержащие статус ответа на запрос о получении списка организаций и поле типа Organisation в котором при удачном получении данных будет храниться подробная информация о организации, включая контактные данные, индивидуальный номер типа Guid и администратора организации из экземпляра класса Employee

Модель данных представляющую получаемые в ответ на запрос информацию о медицинских сотрудниках и их специальностях реализует класс GetEmployeeResult, который зависит от классов EmployeePost и Employee. Класс GetEmployeeResult представляет модель данных о сотрудниках, а EmployeePost представляет модель данных о специальностях. UML диаграмма классов представлена на рисунке 11

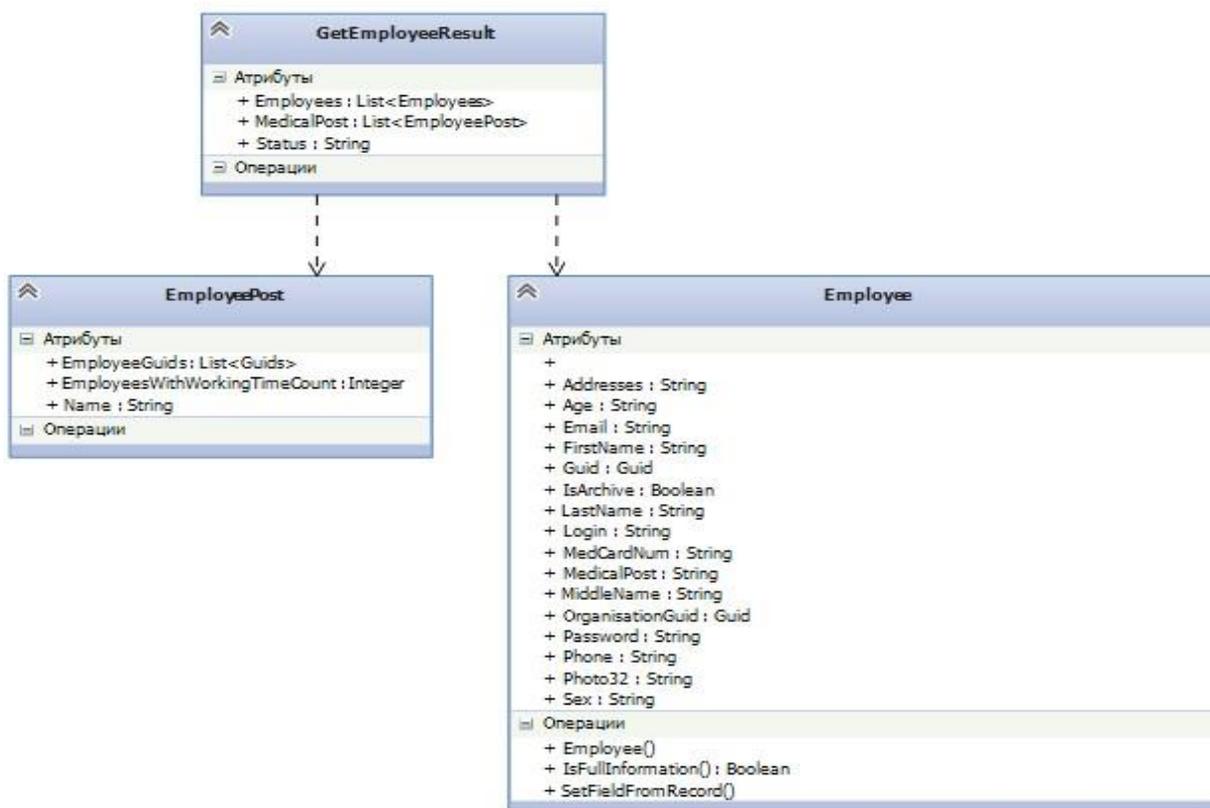


Рисунок 11 – UML диаграмма классов, реализующих модель данных сотрудников

Класс `GetEmployeeResult` содержит поля для хранения списков сотрудников и занимаемых должностей в типах, которые реализуют классы `Employee` и `EmployeePost` соответственно.

Класс `Employee` содержит поля для хранения индивидуального номера сотрудника типа `Guid`, его персональные данные, занимаемую должность, номер организации типа `Guid` и фото в виде строки, закодированной в формате `Base64`.

Класс `EmployeePost` содержит строку для хранения названия должности и поле для хранения номеров типа `Guid` тех сотрудников, которые относятся к данной должности.

Модель данных, отправляемая при запросе на регистрацию пользователя реализует класс `PostRegistration`, UML диаграмма которого представлена на рисунке 12.



Рисунок 12 – UML диаграмма класса PostRegistration.

Класс PostRegistration включает поля для регистрации пользователя, содержащие основную информацию как логин, имя, пароль и электронный адрес для подтверждения регистрации. Строки подтверждения пароля не требуется, так как при вводе пароля система автоматически сравнивает вводимые пароли.

При работе с личными записями пациента используется модель данных, реализуемая классом SheduleEvent, UML диаграмма которой представлена на рисунке 13.

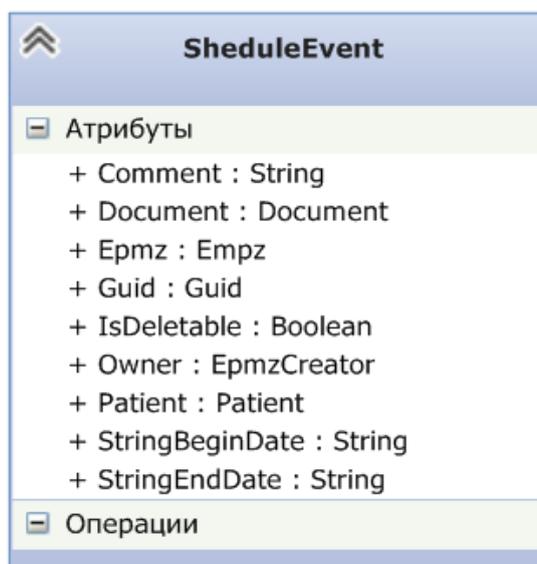


Рисунок 13 – UML диаграмма класса SheduleEvent

2.4 Результат разработки

В результате проектирования было разработано мобильное приложение, позволяющее используя API сервера медицинской информационной системы организовать личный кабинет пользователя, с помощью которого возможно произвести запись на прием к необходимому специалисту, просматривать страницы медицинских учреждений, производить поиск услуг и просматривать персональную медицинскую карту пациента.

В ходе проектирования составлены UML диаграммы вариантов использования, с помощью которых был построен прототип разрабатываемого приложения. Используя прототип для каждого экрана пользовательского интерфейса спроектированы элементы управления, выделены и спроектированы функциональные части разрабатываемого приложения, для которых описаны классы и методы, также построены UML диаграммы классов, которые реализуют модели используемых данных.

Снимки основных экранов приложения приведены на рисунках 14, 15, 16.

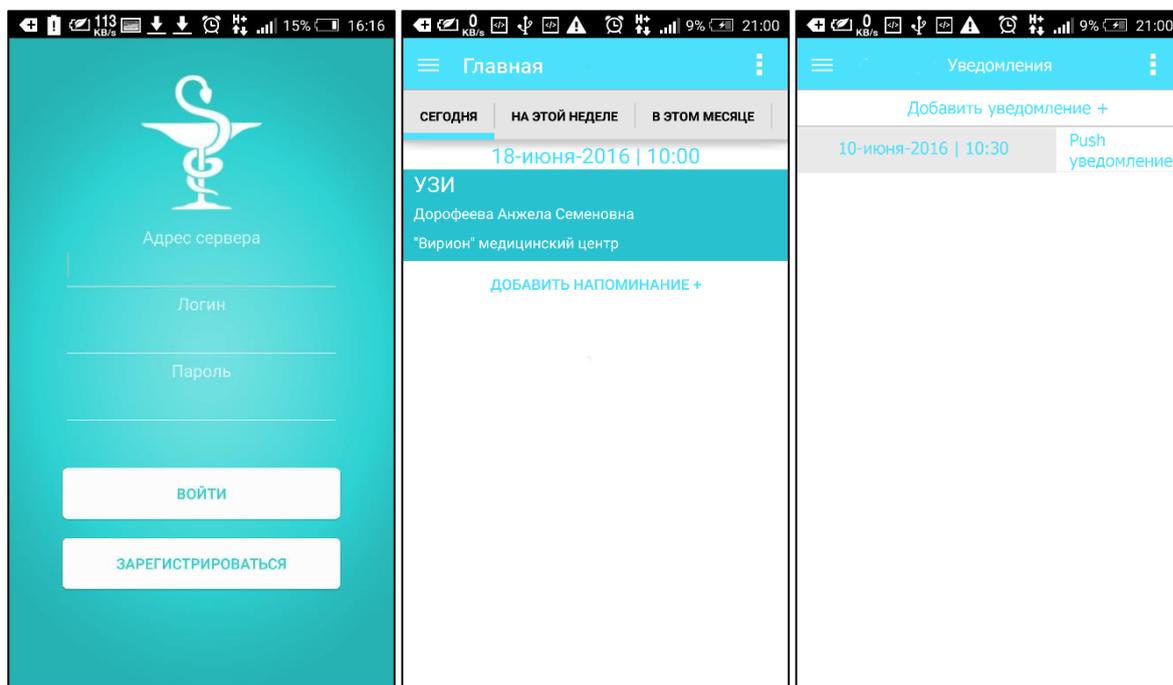


Рисунок 14 – Снимки основных экранов приложения

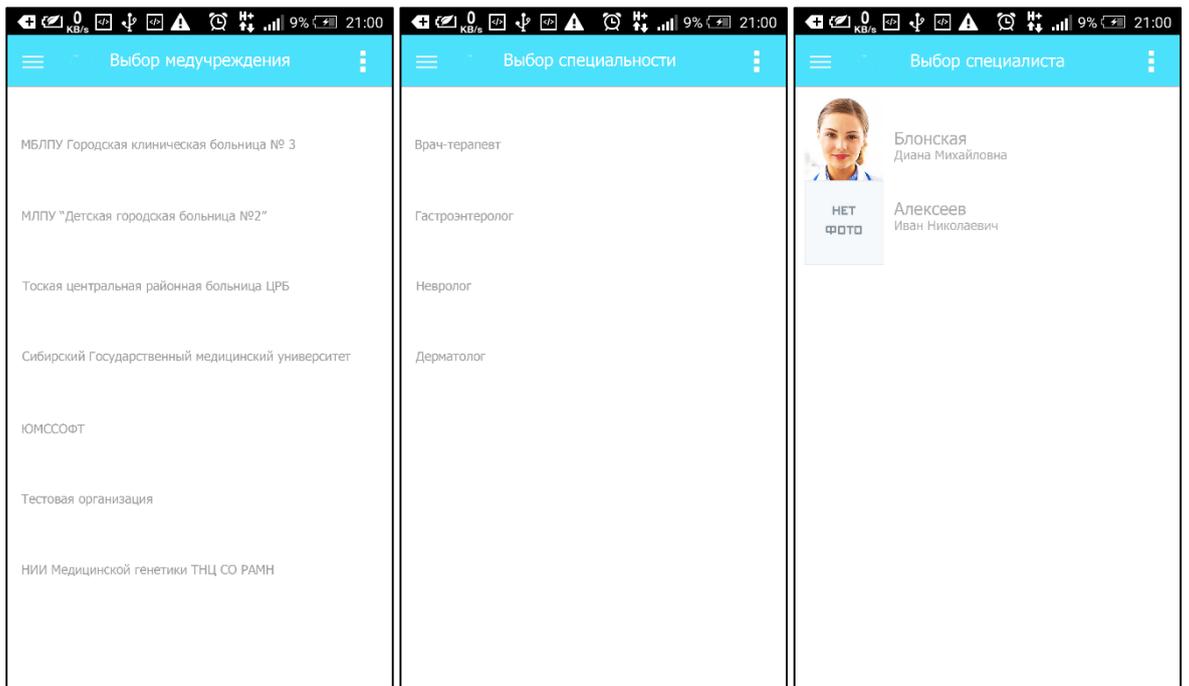


Рисунок 15 – Снимки основных экранов приложения

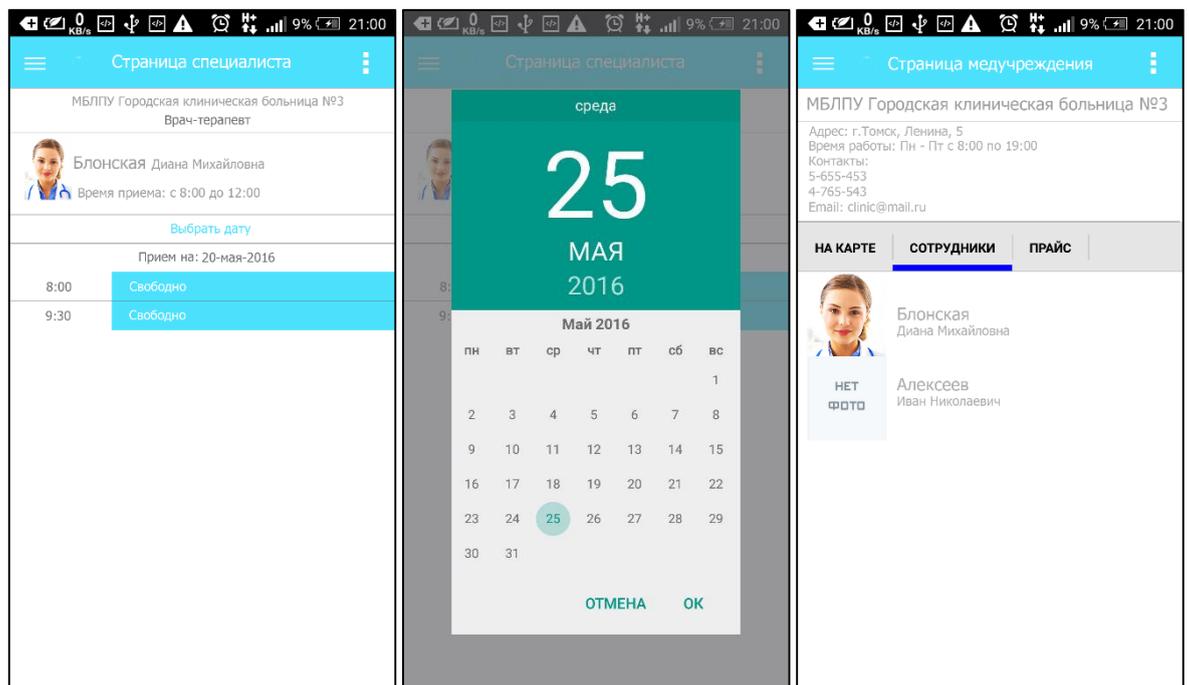


Рисунок 16 – Снимки основных экранов приложения

3 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ

3.1 Введение

В настоящее время научные исследования определяют свою перспективность не только величиной открытия, его пользой обществу, но и коммерческой ценностью исследования. Анализ коммерческой ценности является важной основой для дальнейшей монетизации проекта, а также для поиска финансирования и привлечения сотрудников для дальнейшего развития проекта. Так же очень важно анализировать рынок и конкурентоспособность разрабатываемого продукта. Все это позволяет определить насколько разрабатываемый продукт будет востребован на рынке, по какому пути стоит развивать проект, определить бюджет научного проекта и какова будет конечная цена результатов проекта.[15]

Так как целью данной выпускной квалификационной работы является разработка компонентов информационной поддержки расписания медицинского учреждения, то анализ проекта будет производиться как анализ программного продукта. Разрабатываемые компоненты устанавливаются на мобильные устройства и предназначены для решения задач работы с расписанием медицинского учреждения, с помощью которых возможен запись на прием к врачу, получение уведомлений о приеме, поиск специалистов и ближайших поликлиник. В связи с тем, что в настоящее время в системах здравоохранения имеется ряд проблем, решение которых находится в плоскости информационных технологий и в частности проблем, связанных непосредственно со взаимодействием пациентов с медицинскими учреждениями, которые должно помочь решить данное мобильное приложение, то данная разработка будет весьма востребована на рынке именно медицинских информационных технологий.

Целью данного раздела является проектирование конкурентоспособного мобильного приложения для работы с расписанием медицинских учреждений.

Задачами данного раздела являются оценка коммерческого потенциала и перспективности проведения разработки, а также определение сильных и слабых сторон для формирования общей экономической картины разрабатываемого приложения и корректировки курса его развития.

Процесс решения этих задач будет состоять из анализа существующих конкурентных технических решений и SWOT анализ приложения.

2.1 Анализ конкурентных технических решений

Так как рынки находятся в непрерывном движении, то детальный анализ разработок, которые являются конкурирующими, необходимо проводить систематически. Анализ конкурентов позволяет корректировать курс разработки чтобы выдерживать конкуренцию с подобными разработками и превосходить их. Чем реалистичнее и подробнее будут рассмотрены конкурирующие разработки, тем четче будет сформирован путь развития собственной разработки.

Для анализа конкурентов может быть использована следующая информация:

1. уровень проникновения на рынок;
2. бюджет разработки;
3. уровень завершенности научного исследования;
4. технические характеристики разработки.

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения.

Для оценки сравнительной эффективности научной разработки необходимо провести анализ конкурентных технических решений со стороны

ресурсосбережения и ресурсоэффективности. Это позволит определить направление будущего развития данных сторон.

Анализ конкурентов проводится с помощью оценочной карты конкурентов.

На сегодняшний день у данной разработки существует несколько конкурентов: приложение «ЕМИАС», мобильное приложение «К врачу», и мобильное приложение «Медицина 72». Так как разработка производится для платформы Android, то конкуренты были выбраны для данной платформы. Анализ конкурентных разработок приведен в таблице 1.

Таблица 1 – Оценочная карта конкурентов

Критерий оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
1. Удобство в эксплуатации	0,15	5	5	3	4	0,75	0,75	0,45	0,6
2. Энергоэкономичность	0,08	5	4	4	3	0,4	0,32	0,32	0,24
3. Надежность	0,07	5	3	4	3	0,35	0,21	0,28	0,21
4. Безопасность	0,01	5	5	5	3	0,05	0,05	0,05	0,03
5. Потребность в ресурсах памяти	0,03	5	4	4	3	0,15	0,12	0,12	0,09
6. Функциональная мощность	0,1	5	4	2	3	0,5	0,4	0,2	0,3
7. Простота эксплуатации	0,07	5	4	3	4	0,35	0,28	0,21	0,28
8. Качество интерфейса	0,17	4	5	3	3	0,68	0,85	0,51	0,51
Экономические критерии оценки эффективности									
1. Конкурентоспособность продукта	0,1	4	3	2	1	0,4	0,3	0,2	0,1
2. Уровень проникновения на рынок	0,1	1	2	1	4	0,1	0,2	0,1	0,4
3. Цена	0,05	5	5	5	5	0,25	0,25	0,05	0,25
4. Предполагаемый срок эксплуатации	0,01	3	3	2	3	0,03	0,03	0,02	0,03
5. Послепродажное обслуживание	0,01	3	1	1	4	0,03	0,01	0,01	0,04

Продолжение таблицы 1.

Критерий оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
6. Финансирование научной разработки	1	1	1	1	2	0,02	0,02	0,04	1
7. Срок выхода на рынок	3	3	5	2	4	0,1	0,04	0,08	3
8. Наличие сертификационной разработки	1	1	1	1	1	0,01	0,01	0,01	1
Итого	1	60	55	43	50	4,13	3,9	2,59	3,21
Примечание: 1) Ф – данная разработка; 2) к1 – мобильное приложение для Android «ЕМИАС»; 3) к2 – мобильное приложение для Android «К врачу»; 4) к3 – мобильное приложение для Android «Медицина 72».									

Из составленной оценочной карты видно, что разрабатываемое приложение превосходит своих конкурентов по большинству сравниваемых параметров и из этого можно сделать вывод, что в совокупности оно лучшее среди конкурентов.

Уязвимость позиции конкурентов обусловлена несколькими критериями.

Приложение «ЕМИАС» имеет большую функциональную мощность, удобство в эксплуатации за счет продуманного интерфейса, но имеет слабую надежность работы с сетью, может использоваться только с одним сервером, низкий уровень проникновения на рынок, практически не имеет послепродажного обслуживания, приложение обновляется редко.

Приложение «К врачу» имеет большую надежность по сравнению с приложением «ЕМИАС», но в свою очередь не обеспечивает требуемого от него удобства в эксплуатации и имеет малую функциональную мощность вследствие ограниченности функционала только просмотром расписания и записью на прием. Имеет низкий уровень проникновения на рынок, ограниченное финансирование разработки и почти не поддерживается после

выпуска, приложение так же обновляется редко и обновления происходят незначительно.

Приложение «Медицина 72» имеет низкий уровень эргономичности, интерфейс однообразен, но в то же время приложение просто в эксплуатации за счет малого количества поддерживаемых функций которые удобно для пользователя структурированы всего на нескольких экранах приложения.

В разрабатываемом приложении были учтены все недостатки своих конкурентов для достижения максимальной конкурентоспособности. Помимо недостатков были также учтены и достоинства конкурирующих программ для того что бы и в этом плане им не уступать. Но несмотря на то, что были выделены критерии сильных и слабых сторон относительно конкурентов для того что бы построить наиболее четкий путь развития конкурентоспособного приложения необходимо рассмотреть факторы внешней и внутренней среды разработки более детально, что позволяет SWOT-анализ.

3.2 SWOT анализ

SWOT – это метод стратегического планирования, который заключается в выявлении факторов внутренней и внешней среды относительно следующих категорий:

1. Strengths – сильные стороны;
2. Weaknesses – слабые стороны;
3. Opportunities – возможности;
4. Threats – угрозы.

Матрица SWOT анализа представлена в таблице 2.

Таблица 2 - SWOT анализ

	Сильные стороны	Слабые стороны
	1) Отказоустойчивость приложения 2) Постоянная техническая поддержка 3) Высокая производительность 4) Низкие системные требования 5) Может работать с разными серверами 6) Хранение записей на сервере, что исключает потерю данных при сбоях клиента	1) Нет возможности связаться со медицинским работником через приложение 2) Нет возможности работать локально 3) Отсутствует возможность записи на прием другого родственника
Возможности	1) Расширить количество поддерживаемых платформ 2) Оптимизация работы с данными 3) Доработка интерфейса	1) Активная рекламная кампания среди потенциальных пользователей 2) Внедрение в большое количество медицинских порталов 3) Увеличение количества использующих систему ЛПУ
Угрозы	1) Увеличение функционала и удобства использования 2) Увеличение темпов реализации планов на рост и развитие приложения	1) Увеличение стабильности работы с сетью 2) Анализ причин развития конкурентов и перенятие их опыта

3.3 Заключение

В данном разделе произведен анализ конкурентов и SWOT анализ, поставленная цель достигнута, задачи решены. Рассмотрены сильные и слабые стороны конкурентов, также выявлены свои сильные и слабые стороны. Определён путь развития в будущем. В результате можно сделать вывод, что было спроектировано конкурентоспособное приложение на базе новой современной технологии, которое готово к выходу на рынок.

4 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Поскольку, приложение в данной работе разрабатывается с помощью компьютера, то необходимо выделить особенности организации рабочего места за компьютером.

4.1 Требования к помещениям при работе за компьютером

Помещения должны иметь естественное и искусственное освещение. Расположение рабочих мест за мониторами для взрослых пользователей в подвальных помещениях не допускается. [16,17]

Площадь на одно рабочее место с компьютером для взрослых пользователей должна составлять не менее 4,5 м², а объем не менее – 15 м³. [16,17]

Помещения с компьютерами должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией. [16,17]

Для внутренней отделки интерьера помещений с компьютерами должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка — 0,7-0,8; для стен — 0,5-0,6; для пола — 0,3-0,5.

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами. [16,17]

В помещении должны находиться аптечка первой медицинской помощи, углекислотный огнетушитель для тушения пожара.[16,17]

4.2 Требования к освещению

В компьютерных залах должно быть естественное и искусственное освещение. Естественное освещение обеспечивается через оконные проемы с коэффициентом естественного освещения КЕО не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории.

Световой поток из оконного проема должен падать на рабочее место пользователя с левой стороны.[16,17]

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.[16,17]

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м². [16,17]

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не должна превышать 40 кд/м². Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 — 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.[16,17]

Для искусственного освещения помещений с персональными компьютерами следует применять светильники с зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5, ЛСО4, ЛПО34, ЛПО31 с люминисцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении — линии светильников должны

располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь непросвечивающийся отражатель с защитным углом не менее 40 градусов. [16,17]

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп. [17]

4.3 Требования к помещениям при работе за компьютером

Помещения должны иметь естественное и искусственное освещение. Расположение рабочих мест за мониторами для взрослых пользователей в подвальных помещениях не допускается. [16,17]

Площадь на одно рабочее место с компьютером для взрослых пользователей должна составлять не менее 4,5 м², а объем не менее - 15 м³. [16,17]

Помещения с компьютерами должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией. [16,17]

Для внутренней отделки интерьера помещений с компьютерами должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка — 0,7-0,8; для стен — 0,5-0,6; для пола — 0,3-0,5. [10,9]

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами. [16,17]

В помещении должны находиться аптечка первой медицинской помощи, углекислотный огнетушитель для тушения пожара. [16, 17]

4.4 Требования к микроклимату помещений

На рабочих местах пользователей персональных компьютеров должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПиН 2.2.2/2.4.1340-03, требования которого указаны в таблице 3.

Таблица 3 – Требования к микроклимату помещений

Параметр микроклимата	Период года	Величина
Температура воздуха в помещении	Холодный	22 - 24С
	Теплый	23 - 25С
Относительная влажность	Холодный	40-60%
	Теплый	
Скорость движения воздуха	Холодный	до 0,1 м/с
	Теплый	0,1 – 0,2 м/с

4.5 Требования к организации и оборудованию рабочих мест

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.[17]

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.[16,17]

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина — соответственно 1 600 мм и 1 200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.[16,17]

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног — не менее 650 мм.[16,17]

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали).[16,17]

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.[16,17]

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.[17]

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно-поворотные рабочие стулья с сиденьем и спинкой, регулируемые по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов.;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину — не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;

- регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухонепроницаемым покрытием, легко очищаемым от загрязнения. [16, 17]

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм. [16, 17]

4.6 Обеспечение электро-пожаробезопасности на рабочем месте

4.6.1 Электробезопасность. На рабочем месте пользователя размещены: дисплей, клавиатура и системный блок. При работе компьютера запрещается вытирать пыль с дисплея и системного блока, работать на компьютере во влажной одежде и влажными руками, поскольку существует риск удара.[17]

Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.[17]

Токи статического электричества, наведенные в процессе работы компьютера на корпусах монитора, системного блока и клавиатуры, могут при-

водить к разрядам при прикосновении к этим элементам. Такие разряды опасности для человека не представляют, но могут привести к выходу из строя компьютера. Для снижения величин токов статического электричества используются нейтрализаторы, местное и общее увлажнение воздуха, использование покрытия полов с антистатической пропиткой.[17]

4.6.2 Пожарная безопасность. Пожарная безопасность — состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных его факторов и обеспечивается защита материальных ценностей.[17]

Противопожарная защита — это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара.[17]

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты. Во всех служебных помещениях обязательно должен быть «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.[17]

Пожары в вычислительном центре представляют особую опасность, так как сопряжены с большими материальными потерями. Характерная особенность вычислительных центров — небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окислителя и источников зажигания. В помещениях вычислительных центров присутствуют все три основных фактора, необходимых для возникновения пожара.[17]

Горючими компонентами на вычислительных центрах являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, изоляция кабелей и др.[17]

Источниками зажигания в вычислительных центрах могут быть электрические схемы компьютера, приборы, применяемые для технического об-

служивания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.[17]

В современных компьютерах очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты от оборудования служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность. [17]

Для большинства помещений вычислительных центров установлена категория пожарной опасности В.[17]

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были рассмотрены существующие решения, произведен анализ популярных мобильных операционных систем и интегрированных сред разработки, выявлены требования к пользовательскому интерфейсу приложения и согласно заданной предметной области составлено техническое задание для разрабатываемого приложения.

Для реализации мобильного приложения был выбрана и обоснована мобильная платформа и интегрированная среда разработки.

Во время проектирования мобильного приложения были разработаны UML диаграммы вариантов использования, с помощью которых в программном обеспечении Axure RP был построен прототип разрабатываемого приложения.

Разработка приложения происходила в интегрированной среде разработки Visual Studio 2015 Enterprise с использованием технологии Xamarin с применением средств разработки Android SDK на платформе .NET Framework 4.5.2.

В разделе Финансовый менеджмент были выделены основные достоинства и недостатки конкурентных приложений, была определена конкурентоспособность приложения, его слабые и сильные стороны, возможности и угрозы, что позволило определить дальнейший путь развития приложения.

В разделе социальная ответственность были определены основные требования к организации рабочего места разработчика, требования к обеспечению пожарной безопасности, микроклимату и освещенности помещения.

В результате выполнения выпускной квалификационной работы были достигнуты все поставленные цели, решены все задачи.

CONCLUSION

During execution of the qualification work were considered existing solutions, analysis was made of the popular mobile platforms and integrated development environments, identified requirements for user interface applications and according to the given subject area has been compiled the technical specification for develop application.

To implement mobile applications has been selected and justified on a mobile platform and integrated development environment.

During the design of mobile apps have been designed UML diagrams of use cases, which in the software Axure RP was built a prototype develop application. According to the obtained prototype was built the application for the chosen platform.

The application was carried out in the integrated development environment Visual Studio 2015 Enterprise using Xamarin technology with application development tools Android SDK platform .NET Framework 4.5.2.

In the section Financial management were highlighted the main advantages and disadvantages of competitive applications, determined the competitiveness of the develop application, his strengths and weaknesses, opportunities and threats, which allowed to determine the future path of development of the application.

In the social responsibility, section sets out the main requirements for the organization of a workplace of the developer, requirements electrical and fire safety, microclimate and ambient light.

As a result of execution of final qualifying works were all achieved goals, solved all the tasks.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Информационные технологии в медицине. [Электронный ресурс] URI: <http://itm.consef.ru>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 10.02.2016

2 Google Play. [Электронный ресурс] URI: <https://play.google.com/store>, свободный. – Загл. с экрана. – Яз. рус., англ. Дата обращения: 15.02.2016

3 International Data Corporation. [Электронный ресурс] URI: <https://www.idc.com>, свободный. – Загл. с экрана. – Яз. англ. Дата обращения: 20.02.2016

4 Дэйв Марк. Разработка приложений для iPhone, iPad и iPod touch с использованием iOS SDK [Текст] / Дэйв Марк. – Вильямс, 2011. – 624с.

5 Пугачев С. Разработка приложений для Windows Phone 7.5 [Текст]: учебное пособие / С. Пугачев, С. Павлов, Д. Сошников. – БХВ – Петербург, 2012. – 384с.

6 Голощапов А. Google Android: Программирование для мобильных устройств [Текст] / А. Голощапов. – БХВ-Петербург, 2010. – 448с.

7 Введение в интегрированную среду разработки Eclipse. [Электронный ресурс] URI: http://www.javaportal.ru/java/ide/intro_eclipse.html, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 25.02.2016

8 Давыдов С. IntelliJ IDEA. Профессиональное программирование на Java. Наиболее полное руководство [Текст] / С. Давыдов, А. Ефимов. – БХВ – Петербург, 2005. – 800с.

9 Центр разработчиков ANDROID. [Электронный ресурс] URI: <http://developer.android.com>, свободный. – Загл. с экрана. – Яз. рус., англ. Дата обращения: 10.03.2016

10 Центр разработчиков XAMARIN. [Электронный ресурс] URI: <http://developer.xamarin.com>, свободный. – Загл. с экрана. – Яз. англ. Дата обращения: 10.03.2016

11 Visual Studio Enterprise. [Электронный ресурс] URI:

<https://www.visualstudio.com/products/visual-studio-enterprise-vs.> – Загл. с экрана. – Яз. рус. Дата обращения: 20.03.2016

12 Введение в интегрированную среду разработки Eclipse. [Электронный ресурс] URI: http://www.javaportal.ru/java/ide/intro_eclipse.html, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 13.01.2016

13 Центр разработчиков MSDN. [Электронный ресурс] URI: <http://msdn.microsoft.com>, свободный. – Загл. с экрана. – Яз. рус., англ. Дата обращения: 25.03.2016

14 Шилдт Герберт. Полный справочник по С# [Текст]: справочник / Шилдт Герберт. - Вильямс, 2005. – 752с.

15 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова, Н.В. Шаповалова, Л.Р. Тухватулина З.В. Криницына; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 36 с.

16 Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 (с изменениями от 25 апреля 2007 г.). [Электронный ресурс] URI: http://www.infosait.ru/norma_doc/39/39082/index.htm#i222500, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 10.04.2016

17 Пособие по БЖД. [Электронный ресурс] URI: <http://www.studfiles.ru/preview/434015>, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 10.04.2016