**Министерство образования и науки РФ**

**Федеральное государственное автономное образовательное учреждение**

**Высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННЫХ СИСТЕМ**

Специальность: 09.03.03. – Прикладная информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Разработка комплекса программ для автоматизации процесса сбора и анализа данных в мед. учреждениях.**

**Работа завершена:**

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Студент группы 11-302 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Р. Назмутдинова

**Работа допущена к защите:**

Научный руководитель

Ст. преп. ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_201 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Кугуракова

Директор Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_201 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Ф. Хасьянов

Казань – 2017 г.

Содержание

**[Введение](#_Toc484118408)** [4](#_Toc484118408)

[**1.** **Анализ области** 5](#_Toc484118409)

[**1.1** **Предметный анализ** 5](#_Toc484118410)

[**1.2** **Анализ существующих методов, методик и технологий автоматизации работы с тестами/опросами.** 6](#_Toc484118411)

[1.3 Анализ и изучение документов/тестов для выбора оптимального решения автоматизации процесса. 7](#_Toc484118412)

[1.3.1. Тесты, используемые для оценки. 7](#_Toc484118413)

[1.3.2. Скрининговые тесты. 12](#_Toc484118414)

[1.3.3. Короткие скрининг тесты. 14](#_Toc484118415)

[1.3.4 Анализ сайтов схожей тематики 16](#_Toc484118416)

[**2.** **Постановка задачи** 17](#_Toc484118417)

[**3.** **Техническое задание** 18](#_Toc484118418)

[**3.1** **Основная информация** 18](#_Toc484118419)

[3.2 Системные и нефункциональные требования 18](#_Toc484118420)

[3.2.1 Для сайта 18](#_Toc484118421)

[3.2.2 Для приложения 18](#_Toc484118422)

[3.3 Описание 18](#_Toc484118423)

[3.3.1 Сайт 18](#_Toc484118424)

[3.3.2 Приложение 22](#_Toc484118425)

[**4.** **Разработка** 28](#_Toc484118426)

[**4.1** **Сайт** 28](#_Toc484118427)

[**4.1.1** **Выбор инструментария** 28](#_Toc484118428)

[**4.1.2** **Разработка** 28](#_Toc484118429)

[**4.1.3** **Проблемы при разработке** 28](#_Toc484118430)

[**4.1.4** **Итоговый результат** 29](#_Toc484118431)

[**4.2** **Android приложение** 30](#_Toc484118432)

[**4.2.1** **Выбор инструментария** 30](#_Toc484118433)

[**4.2.2** **Разработка** 30](#_Toc484118434)

[**4.2.3** **Проблемы при разработке** 32](#_Toc484118435)

[**4.2.4** **Итоговый результат** 32](#_Toc484118436)

[**5.** **Заключение** 33](#_Toc484118437)

[**6.** **Список используемых источников** 34](#_Toc484118438)

[**7.** **Приложение** 35](#_Toc484118439)

# **Введение**

На сегодняшний день происходит стремительное развитие в области информационных технологий. IT сейчас затрагивает почти все области жизни человека, в том числе и медицину. Как ни странно, какие-то аспекты прорывными шагами идут вперед, благодаря информационным технологиям, а другие вовсе забыты. Например, в наших поликлиниках и больницах проводят тесты на бумажных носителях, врачи сами проверяют и анализируют результаты сотнями, тратя свое время.

Была поставлена цель: упростить работу врачей путем перевода тестов в электронную версию. Эти тесты направлены на обнаружение​ ​отклонений​ ​в​ ​области​ ​неврологии. Так же с помощью данного приложения возможен анализ большого количества данных и отслеживание динамики пациента.

Объектом исследования является методологии сбора и анализа данных в мед. учреждениях.

Задачи:

1. Рассмотреть возможность перехода с письменного заполнения документов/тестов для работы с пациентами в электронный режим, автоматизация процесса.

2. Проанализировать существующие методы, методики и технологии сбора данных.

3. Выявить потребности

4. Разработать комплекс программ для вышеупомянутых задач

5. Контроль и помощь во внедрении разработанной продукции

Методы исследования:

– анализ научной литературы,

– анализ документов,

– анкетирование,

– опрос.

# **1.** **Анализ области**

* 1. **Предметный анализ**

Выявление и анализ клинических особенностей нарушений когнитивных функций имеет большое значение для диагностики и дифференциальной диагностики неврологических заболеваний. Многие неврологические заболевания, особенно в детском и пожилом возрасте, проявляются почти исключительно когнитивными нарушениями (КН). Наличие и выраженность КН во многом определяют прогноз и тактику ведения пациента при целом ряде распространенных нервных болезней. Диагностика подобных нарушений, определение степени поражения и оценка эффективности лечения является важным направлением в неврологии.

В настоящее время существует большое количество нейропсихологических тестов, которые направлены на определение степени и вида поражения. Наиболее популярные тесты мини-ког, MMSE, МоСА тест, [6]которые чаще всего проводятся неврологами и нейропсихологами. Эти тесты охватывают основные когнитивные функцию, дают возможность провести скрининг когнитивного статуса среди населения.

В течение длительного времени исследователей в области нейропсихологии интересует вопрос о возможности автоматизированного тестирование с использование современных компьютерных технологий и многие традиционно используемые тесты были компьютеризированы, которые используются в течение, примерно, 50 лет. [[1]](#footnote-1)

* 1. **Анализ существующих методов, методик и технологий автоматизации работы с тестами/опросами.**

Шкала Векслера была первой переведена в электронный вид уже в 1969 году, после чего автоматизированы были его различные вариации. Так как автоматизированные формы имели сложности в оценке и их проведении, в период 1980 по 1990 года тесты были перепрограммированы и адаптированы.[[2]](#footnote-2) В 1975 году был компьютеризован тест категорий Холстеда[[3]](#footnote-3), который использовался вплоть до 1980 года, хотя и не смог заменить стандартный и валидизированный его вариант. Рост и развитие компьютерных технологий дали возможность усовершенствовать эти тесты и максимально приблизили их к оригиналу.[[4]](#footnote-4) Широко используемый Висконсинский тест сортировки карт так же был компьютеризован и широко используется в нейропсихологии, хотя ряд исследователей указывают на отсутствие психометрической схожести с оригинальным тестом. Использование этих тестов имеет ряд преимуществ[[5]](#footnote-5):

* Может захватить и заинтересовать пациента.
* Правильно сделанный и предоставленный компьютерный тест поможет избежать разочарования пациента и потери достоинства тестов, которые выполняются с легкостью.
* Обучение пользованию компьютером может предоставить пациенту опыт мастерства и ощущение контроля своего состояния.
* Автоматизированный сбор и хранение данных может помощь врачу не отвлекать и фокусировать внимание на лечении пациента.
* Компьютер может дать возможность множественного измерения производительности (скорость, сила и скорость ответа), что вызывает сложности у врача.
* Компьютер эффективно выполняет задачи, которые требует большего времени у человека (к примеру, смена текста, быстрая модификация графического материала)

В то же врем существуют такие ограничения, как низкая нормативная база и низкие психометрические стандарты. Кроме этого, ограничением является сложности в понимании тестов и в использовании новых технологий среди пожилого населения.[[6]](#footnote-6)

На основании обзора баз данных Scopus, Medline с использованием слов: компьютерные тесты, пожилой возраст, деменция, болезнь Альцгеймера, нейропсихологичесие тесты, нейропсихиатрия было найдено несколько тестов, которые чаще всего используется, и были разделены в группы. Кроме этого, обзор Российской литературы показал, что несмотря на большую популярность компьютеризованных тестов среди исследователей всего Мира, в Росси они не имеют широкой популярности среди ученых.

## 1.3 Анализ и изучение документов/тестов для выбора оптимального решения автоматизации процесса.

Цель этих тестов - оценка когнитивного статуса пациентов, которые оценивают не только общий когнитивный статус, но и отдельные его функции.

### 1.3.1. Тесты, используемые для оценки.

Automatic Neuropsychological Assement Metrics[7] (ANAM)

Тест был разработан Министерством обороны США, который изначально использовался для тестирования когнитивного статуса здоровых индивидов, который может меняться под воздействием окружающей среды.[[7]](#footnote-7) Тест на протяжении 30 лет использовался учеными в различных исследованиях, после чего был переведен в компьютеризированный вид. Тест проводится обученным специалистом и занимает от 20 до 60 минут и оценивает такие параметры, как время отклика и количество правильных ответов, а также пропускную способность, который показывает количество предъявленных правильных ответов в минуту и эффективность тестирования.[[8]](#footnote-8) Одним из вариантов этих тестов является тест Dementia Automatic Neuropsychological Assement Metrics (DANAM), который в исследовании Levinson и др. показал себя чувствительным для определения снижения когнитивных функций среди пациентов с болезнью Альцгеймера (БА) за счет оценки времени пропускной способности, которая оказалась более точной, чем просто определение когнитивного снижения. Это означает, что тест может быть полезен для определения ранних стадий БА. ANAM дает возможность оценить: скорость выполнения задания/эффективность, удержание/память и рабочая память.[[9]](#footnote-9) Кроме этого тест показал свою высокую эффективность в оценке эмоционального состояния испытуемых.[[10]](#footnote-10)

Cambridge Neuropsychological Test Automated Battery[8] (CANTAB).

Опросник включает в себя различные тесты, которые оценивают зрительную память, исполнительную функцию, внимание, семантическую/вербальную память, принятие решений, контроль над реакцией и социальную активность. Тест может быть составлен с учетом необходимости определения БА или же умеренных когнитивных нарушений (УКН). Проводится обученным специалистом, при этом требует интерпретации результатов. Очень важным моментом является то, что тест прошел валидизацию на 3000 здоровых респондентах.[[11]](#footnote-11) и показал себя эффективным в дифференциальной диагностике здоровых пациентов, БА и болезни Паркинсона (БП).[[12]](#footnote-12) Практическая значимость тестов получена из отдельных субтестов и эта практическая ценность зависит от степени возможностей респондента[[13]](#footnote-13). Исследование продемонстрировало, что пациенты с БА и УКН показывают разный результат при выполнении теста парных ассоциаций (Paired Associate Learning task - PAL) и снижение выполнения теста может быть маркером преклинической стадии БА, а также признаком необходимости начала терапии.[[14]](#footnote-14)

Cognitive Assessment of Later Life Status [9](CALLS).

Тест разработан для оценки когнитивного статуса испытуемых через телефон. Выполнение тест стандартизировано точными репликами и сценариями для интервьюеров. Часть опросника это адаптированный вариант известных тестов, как MMSЕ, тогда как другие специально сделано для этого опросника и включает конфигурацию звука, дискриминацию шага, время реакции и время принятия решения. Участник опроса может выбрать степень слуха для лучшего выполнения теста. Кроме оценки когнитивного статуса, данный опросник так же оценивает уровень эмоционального состояния. Анализ отдельных компонентов показал, что тест состоит из 5 компонентов: вербального обучения и вербальной памяти, скорости выполнения задания, внимания и рабочей памяти, беглости речи и называния, а также формирования концепции. Тест имеет ряд преимуществ: возможность отдаленного проведения теста, а также использования его у пациентов со снижением зрения. В то же время недостатком является отсутствие непосредственного контакта с испытуемым и отсутствие оценки зрительно-пространственных нарушений, хотя часто первым признаком БА может быть подобное нарушение.[[15]](#footnote-15)

Computerized Neuropsychological Test Battery (CNTB)[10].

CNTB включает 11 субтестов, которые оценивает скорость движения, скорость обработки информации, вербальное и пространственное обучение, память, внимание, язык и пространственные возможности. Один из первых компьютеризованных батарей и несмотря на то, что тест был компьютеризован, нахождение исследователя во время тестирования необходимо для правильного его выполнения и непосредственной оценки результата. Время выполнения теста примерно 50 минут.[[16]](#footnote-16) CNTB является альтернативой ADAS и имеет высокую корреляцию с ADAS-Cog, MMSE, хотя и является более чувствительным тестом.[[17]](#footnote-17)

Mindstreams.

Тест включает 11 субтестов, которые оценивают вербальную и невербальную память, беглость речи, визуально пространственное мышление, скорость выполнение задания, исполнительная функция, внимание и моторную функции и позволяет оценить: память, исполнительную функцию, пространственное мышление и вербальную функцию. Важно, что тест сравним с бумажным вариантом нейропсихологичеких тестов.[[18]](#footnote-18) Многие субтесты являются цифровой адаптацией его письменного варианта и все субтесты доступны на разных языках. Тест устанавливается на компьютер, а результат направляется на сервер для расчета и оценки результата. Длительность проведения теста 45-60 минут. Тест адаптивный и очень удобный для минимизации эффекта обучения. При этом тест является удобным для пациента, простым для управления и практичным для использования у пожилого населения. Тест может проводить специалист с минимальным опытом его выполнения.[[19]](#footnote-19) Тест показал высокую чувствительность в дифференциальной диагностике разных форм УКН, а так же в выявлении тяжелой и средней степени деменции.[[20]](#footnote-20) Кроме этого даже короткая форма этого теста показала себя достаточно эффективным в дифференциальной диагностике УКН и выраженных когнитивных нарушений.[[21]](#footnote-21)

### 1.3.2. Скрининговые тесты.

Тесты направлены для выявления наличия когнитивного снижения на преклиническом уровне с целью направления пациента в специализированные центры для более глубокой оценки состояния. Представлено ряд тестов, которые направлены на выявление первых признаков когнитивного дефицита.

CNS vital signs (CNSVS).

Тест включает в себя 7 субтестов, которые оценивают вербальную, зрительную и память, исполнительную функцию, скорость выполнения задания, психомоторную скорость, время реакции, комплексное внимание и когнитивную гибкость. При выполнении теста необходимость присутствия интервьюера не обязательно. Время выполнения около 30 минут. CNSVS имеет внушительную нормативную базу, которая собрана на 1662 здоровых добровольцах в возрасте от 8 до 90 лет, которые разделены на 10 групп. Что важно и то отличает тест от других это возможность его использований среди пациентов с деменцией тяжелой степени. Тест эффективен для дифференциальной диагностики УКН, выраженных нарушений и нормы. Для дальнейшей диагностики важно, что результаты оценил обученный специалист, так как использования только этого теста не достаточно для установления окончательного диагноза.[[22]](#footnote-22)

CogState.

Тест включает 8 субтестов, которые оценивают скорость выполнения заданий, внимание, принятие решений, рабочую память, зрительное обучение и зрительное внимание, а также эпизодическую память. Время выполнения теста 15-25 минут в зависимости от вида теста (короткая или полная версия). CogState был разработан как тест для повторной оценки (repeat testing). Минимальные практические эффекты и надежность повторного тестирования делает возможность оценки внутрииндивидальные когнитивные нарушения как у здоровых пациентов, так и у пациентов с когнитивным дефицитом.[[23]](#footnote-23) Тест имеет высокую чувствительность, в частности зрительное обучение и рабочая память являются эффективными субтестами для выявления преклиническую и клиническую стадии деменции.[[24]](#footnote-24)

Computer Assesment of Mild Cognitive Impairment [11] (CAMCI).

Система была разработана с целью доклинического выявления когнитивного снижения и определение наличия деменции. Тест состоит из 8 субтестов оценивающих 4 когнитивные функции: внимание, исполнительную функцию, скорость выполнения задачи и памяти. Тест прост в выполнении, что упрощает использовании его среди пожилого населения, при этом чувствительность теста остается выше, чем у MMSE. Выполняет самостоятельно, врач получает почитанный результат, после чего проводит интерпретацию полученных результатов. Время заполнения опросника 25 минут.[[25]](#footnote-25)

### 1.3.3. Короткие скрининг тесты.

Тесты нового поколения, которые занимают короткое время для выполнения (10-15 минут), но остаются выскоэффективными для определения наличия когнитивного дефицита.

CANTAB Mobile[5].

Тест, который разработан для мобильных устройств. Занимает 10 минут, дает характеристику эпизодической памяти. Тест содержит тесты из Гериатрической шкалы депрессии и вопросы повседневной активности. Тест имеет большую доказательную базу, собранную на 4000 пожилых пациентах. Инструкции представлены в виде голосовых записей, которые доступны на 18 языках. Программа автоматически дает интерпретацию результата в виде: «Проблем нет», «Наблюдение», «Дальнейшее обследование».[[26]](#footnote-26)

Cognitive Function Test (CFT).

Короткий тест, разработанный для определения тонких когнитивных нарушений у пациентов в возрасте от 50 до 65 лет, которые могут признаком начала деменции. Основной целью теста является определение ранних признаков проблем с когнитивными функциями и раннего начала лечения патологии. Время выполнения теста 15 минут. Опросник адаптированы из бумажных вариантов тестов и содержит: называние, эпизодическую память, исполнительную функцию и скорость выполнения задания.

Тест был валидизирован среди здорового населения и имеет корреляцию с бумажными вариантами тестов.[[27]](#footnote-27)

MCI screen.

Тест разработан с целью телефонного опроса пациентов путем оценки памяти, языка и исполнительную функцию и занимает 10 минут. Определяет Индекс производительности памяти в пределах от 0 до 100 баллов. При относительной простоте выполнения тест оказался более чувствительным, чем MMSE и тест рисования часов. Так же как другие тесты дает возможность определить ранние признаки когнитивного дефицита.[[28]](#footnote-28)

Кроме тестов, которые представлены выше, в настоящее время ведется разработка новых тестов: CogniScreen, Experemental battery developed by Kluger et al., Experemental battery developed by Inoue et al., Experemental battery developed by Maki et al.[[29]](#footnote-29) Тесты находятся в стадии валидизации, но общим для них является возможность самоконтроля и выделение когнитивных нарушений и признаков депрессии, что является важным аспектом в диагностике когнитивного дефицита.

### 1.3.4 Анализ сайтов схожей тематики

Анализ сайтов схожей тематики [3,4] и др. показал большую загруженность информационными блоками сайта в целом, что отрицательно сказывается на восприятии. Так же появляется сложность при поиске нужной статьи либо рекомендации. Такие сайты содержат в себе максимум информации, однако отыскать нужную становится затруднительно. Все содержимое необходимо разбивать на небольшие блоки, объединяя все аспекты в строке меню.

1. **Постановка задачи**

На встрече с профессором кафедры неврологии и мануальной терапии университетской клиники Казань (Медико-санитарная часть ФГАОУ ВО «Казанский (Приволжский) федеральный университет») Есиным Р. Г. были выявлены следующие потребности:

1. Необходимость создания сайта "Клиническая неврология, клиническая лингвистика: диагностика и реабилитация для детей и взрослых" для размещения основной информации отделения и опросов различных направлений.

Мы пришли к выводу, что для второй части наиболее эффективным решением будет интеграция на сайт готового решения создания опросов – Google form и дальнейшая обработка выходных данных.

К разработке так же привлечены студенты кафедры дизайна и национальных искусств.

1. Создание offline приложения с опросами/анкетами для операционной системы android.

Данная операционная система была выбрана исходя из наличия некоторого количества устройств для тестирования. В дальнейшем, при необходимости, обговаривалась возможность закупки оборудования.

Главной задачей приложения является поддержание функционала (тестирование/опрос) в ненормированных условиях работы без доступа к интернету – командировки, выездные мероприятия и т.п.

1. **Техническое задание**
   1. **Основная информация**

Дипломный проект включает в себя разработку двух независимых компонентов (сайт, мобильное приложение) для упрощения обработки информации с бумажных носителей, анализа, сбора статистики.

Данные тесты должны быть максимально простыми, т.к. предназначены для пользования людьми всех возрастных групп.

## Системные и нефункциональные требования

## Для сайта

Требования к аппаратному обеспечению зависят от количества посетителей сайта и объема информации.

Для нормальной работы с сайтом требуется любая операционная система с установленным браузером и выходом в интернет.

Для просмотра выходных и статистических данных необходима установка пакета Microsoft Office не ранее 2003 года выпуска.

## Для приложения

Для данного приложения основными требованиями является наличие операционной системы android версии выше 4.0 и возможность создания точки доступа как минимум на одном из устройств

## Описание

## Сайт

Изначальной задумкой было написать страницу-опросник для конкретного теста. В ходе обсуждений было принято решение значительно увеличить функционал данного сайта, а именно возможность добавления нескольких тестов и создание учетных записей.

Данный сайт предназначен для тестирования/анкетирования пациентов в домашних условиях. Главным требованием было возможность врачом создания индивидуальной ссылки для каждого пациента с присвоением уникального идентификатора. Данная ссылка отправляется пациенту. Срок действия ссылки – 3 суток с момента создания. После ссылка становится неактивной.

Так же на сайте есть администраторы с 2 уровнями доступа. А именно:

* Главный администратор, который может контролировать работу остальных администраторов сайта,
* Остальные администраторы.

Механика работы:

Рассмотрим механику работы сайта со стороны администратора (врача) и обычного пользователя.

|  |  |
| --- | --- |
| **Действия администратора** | **Реакция системы** |
| Ввод в адресную строку браузера адрес сайта | Загрузка страницы авторизации |
| Ввод верной комбинации логина/пароля | Загрузка страницы личного кабинета, где отображается:   * Имя и должность администратора, * Фото профиля, * Плитка с количеством созданных опросов, * Плитка с количеством пользователей онлайн, * Плитка с суммой количества просмотров всех тестов.   Для главного администратора отображается следующее:   * Имя и должность администратора, * Фото профиля, * Плитка с общим количеством тестов, созданных на сайте, * Плитка с количеством зарегистрированных администраторов, * Плитка с общей суммой количества просмотров всех тестов. |
| Выбор плитки с опросами | Загрузка страницы тестов, где отображается:   * Все созданные тесты пользователя, * Форма для добавления теста * Форма для удаления теста   Для главного администратора отображается следующее:   * Тесты, созданные всеми пользователями, объединенные в папки с именами авторов тестов. * Форма для добавления теста * Форма для удаления теста любого пользователя |
| Выбор теста | Переход на страницу теста/опроса |
| Добавление теста:  Для добавления нового теста/опроса на сайт необходимо:   1. предварительно создать тест на docs.google.com/forms, отметив обязательные поля и создав там же документ для ответов. 2. скопировать ссылку в режиме предварительного просмотра теста 3. Нажать кнопку «добавить» | Добавление теста в список с присвоением порядкового номера, идентификатора(ID) и отображением количество просмотров теста. |
| Удаление теста:  Для удаления теста необходимо ввести в поле ввода идентификатор теста и нажать кнопку «удалить» | Удаление строки с данным идентификатором, обновление списка тестов. |
| Выбор плитки администратора | Нет действия  Для главного администратора отображается список пользователей со следующими данными:   * Логин * Пароль * Уровень доступа * ФИО * Должность   Так же отображается форма для регистрации нового пользователя, которая включает в себя:   * Логин * Пароль * Уровень доступа * ФИО * Должность |
| Добавление пользователя (только для главного администратора):  Для создания нового пользователя необходимо заполнить все поля формы и нажать кнопку добавить | Обновление списка пользователей, добавление строки нового пользователя |
| Выбор плитки просмотров | Нет действия |

|  |  |
| --- | --- |
| Действия пользователя | Реакция системы |
| Ввод в адресную строку браузера индивидуальной ссылки | Загрузка страницы с тестом.  Данная ссылка активна трое суток с момента создания. |
| Нажатие кнопки «завершить тест/отправить» | Сохранение всех введенных данных.  Вывод установленного текста. |

## Приложение

Данное приложение разрабатывается под конкретный тест – PARS (Педиатрическая шкала уровня тревоги)

PARS используется для оценки выраженности тревоги у детей и подростков в возрасте от 6 до 17 лет. В шкале PARS две части: перечень симптомов и оценка их выраженности. Перечень симптомов используется для определения набора симптомов в течение последней недели. Вторая часть состоит из 7 пунктов для определения выраженности симптомов и подсчета общего балла по шкале PARS.

Формат проведения обследования: целью обследования является максимальное извлечение информации об уровне тревоги ребенка. Для достижения данной цели необходимо получить информацию от ребенка и взрослого (как минимум). Клиницист также собирает всю иную информацию, которую только можно, для определения уровня тревоги.

Было принято решение проводить тестирование одновременно на трех устройствах (родитель, ребенок и врач). Таким образом устройства должны синхронизироваться и по завершению выводить наглядные результаты.

Одним из главных условий было возможность проведения теста вне стен больницы (например, командировки в другой город). Это значит, что возможность доступа к интернету может быть ограничена.

Механика работы:

На первый взгляд оптимальным средством синхронизации кажется Bluetooth, при его использовании мы не будем зависеть от интернета. Однако на практике моментальная синхронизация оказалась почти невозможной.

Так как на используемых устройствах имеется возможность создания точки доступа, было принято решение использовать сеть wi-fi. Это сильно облегчило синхронизацию.

Теперь главное условие правильной работы приложения – нахождение всех устройств в одной сети. Если даже у нас нет доступа к интернету мы можем создать точку доступа на одном из устройств, подключив остальные устройства к этой точке и продолжать работать без заминок.

При регистрации нового пользователя система запрашивает пароль для создания новых аккаунтов. Эта необходимость была введена с учетом того, что на устройствах будут проходить тесты дети разных возрастов, чтобы исключить лишних пользователей.

Рассмотрим механику сайта со стороны администратора (врача) и обычного пользователя.

|  |  |
| --- | --- |
| Действия администратора | Реакция системы |
| Запуск приложения | Загрузка страницы авторизации, включающая в себя:   * Поля для авторизации (логин, пароль) * Кнопка регистрации нового пользователя * Кнопка для создания связи с другими устройствами |
| Ввод верной комбинации логина/пароля | Загрузка страницы личного кабинета врача, где отображается:   * Название приложения * Имя и должность врача * Существующие опросы с возможностью: начать, редактировать или удалить опрос * Журнал с результатами пройденных тестов * Кнопка создания нового теста * Кнопка синхронизации тестов * Выход |
| * Ввод неверной комбинации * Ввод неверного пароля для регистрации новых пользователей * Некорректное заполнение одного или нескольких полей регистрации пользователя | Вывод сообщения с указанием места ошибки. |
| Нажатие кнопки регистрации нового пользователя | Запрос пароля для регистрации новых аккаунтов. |
| Ввод верного пароля для регистрации новых аккаунтов | Отображается форма для регистрации нового пользователя, которая включает в себя:   * Логин * Пароль * Повтор пароля * ФИО   Все вышеперечисленные поля являются обязательными   * Кнопка отмена * Кнопка создания пользователя |
|  | Вывод сообщения об ошибке |
| Заполнение корректными данными форму регистрации | Переход на страницу авторизации |
| Нажатие кнопки «отмена» | Выход на предыдущую страницу |
| Нажатие кнопки создания теста | * Поле для ввода названия и описания теста * Кнопка создания вопроса * Кнопка создания рекомендаций по количеству набранных баллов в тесте * Кнопка сохранения теста |
| * Ввод названия и описания теста * Ввод вопроса | Сохранение введенных данных |
| Нажатие кнопки создания вопроса | * Поле для ввода вопроса * Кнопка добавления ответа |
| Нажатие кнопки добавления ответа | * Поле для ввода ответа * Поле для присвоения ответу определенного балла * Кнопка добавления следующего ответа |
| Нажатии кнопки сохранения теста, все поля заполнены корректно | Переход на главную страницу личного кабинета, просмотр списка имеющихся тестов, просмотр ответов на пройденных тестах. |
| Нажатие кнопки удаления теста | Вывод формы для подтверждения удаления с копками «удалить» и «отмена» |
| Нажатие кнопки «удалить» | Удаление теста |
| Нажатие кнопки отмена | Переход на предыдущую страницу |
| Нажатие кнопки редактирования теста | Переход на страницу аналогичную странице создания теста с заполненными полями, возможность редактировать любое из полей. |
| Нажатие кнопки «начать тест» | Открывается тест на всех подключенных устройствах. Перед заполнением запрашиваются поля:  ФИО, возраст, роль (родитель или ребенок). Врач по умолчанию проходит тест на том устройстве, на котором произошла авторизация.  Кнопка завершения теста |
| Нажатие на кнопку завершения теста, корректный ответ на все вопросы | Переход на начальную страницу.  На остальных устройствах выводится сообщение о завершении теста на одном из устройств. |
| Нажатие кнопки поиска устройств | Поиск в подключенной сети wi-fi устройств. При нахождение автоматическое подключение к ним, вывод сообщения на всех устройствах о количестве устройств в сети. |

|  |  |
| --- | --- |
| Действия пользователя | Реакция системы |
| Запуск приложения | Загрузка страницы авторизации, включающая в себя:   * Поля для авторизации (логин, пароль) * Кнопка регистрации нового пользователя * Кнопка для создания связи с другими устройствами |
| Нажатие кнопки поиска устройств | Поиск в подключенной сети wi-fi устройств. При нахождение автоматическое подключение к ним, вывод сообщения на всех устройствах о количестве устройств в сети. |
| Прохождение начавшегося теста, заполнены корректные данные | Сохранение введенных данных  Кнопка завершения теста. |
| Нажатие кнопки завершения теста | Сохранение данных. Вывод рекомендаций при их наличии. Переход на главную страницу |

1. **Разработка**
   1. **Сайт** 
      1. **Выбор инструментария**

Для разработки сайта был выбран язык программирования HTML-5 и Java Script. В качестве СУБД был использован MySQL.

Вся работа велась в редакторе Sublime Text.

Для локального тестирования была выбрана программа XAMPP – среда разработки на PHP. Выбор пал именно на XAMPP, т.к. он бесплатный, простой в установке, с открытым исходным кодом, содержащая Apache, MySQL, интерпретатор скриптов PHP и множество библиотек, позволяющий запустить полноценный веб сервер.

* + 1. **Разработка**

Были реализован проект со следующим наполнением:

add\_admin.php - Функция добавления администратора

admins.php - Вывод всех админов на экран

allurl.php - Вывод всех ссылок с опросами на экран

bd.php - Файл подключения к базе данных

edit.php - Страница с редактированием данных пользователя

index.php - Главная страница сайта

list.css - Объект стиля

login.php - Функция авторизации на сайте

logout.php - Функция выхода на сайте

main.php - Главная страница пользователя

opros.php - Страница с опросом

upload.php - Страница с загрузками фотографий на сервер

* + 1. **Проблемы при разработке**

Основной проблемой при разработке было отсутствие знаний по PHP. Изучение литературы и ответы на вопросы опытным разработчиком почти искоренили эту проблему.

* + 1. **Итоговый результат**

Результатом является рабочий сайт, который прошел стадию тестирования. Сейчас ожидается разработанный дизайна сайта и покупка доменного имени для полноценной работы.

* 1. **Android приложение**
     1. **Выбор инструментария**

Для разработки данного приложение был выбран Android Studio на java.

Android Studio — это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux. 17 мая 2017 на ежегодной конференции Google I/O, Google анонсировал язык Kotlin используемый в Android Studio официальным языком программирования для платформы Android в добавление к Java и С++.

Для тестирования использовалось устройства на ОС android 2.3.3 и выше

* + 1. **Разработка**

Разработка велась в несколько этапов:

* Составление технического задания и последующие корректировки по требованию заказчика
* Разработка приложения
* Работа с дизайнерами, внедрение дизайна

На протяжении всей работы велось нагрузочное тестирование каждой из частей и работы приложения в целом.

Над данным проектом так же работали студенты кафедры дизайна и национальных искусств.

Пакет содержит в себе следующие классы:

EditQuestionActivity - активити редактирования опросов, фрагменты:

* QuestionInfoFragment - информация об опросе,
* QuestsFragment - редактирование вопросов,
* ResultsFragment - редактирование рекомендаций.

LoginActivity - авторизация и регистрация пользователя

MainActivity - главный экран приложения, фрагменты:

* QuestionsFragment - список опросов,
* AnswersFragment - результаты опросов.

PlayQuestionActivity - логика прохождения теста

UserActivity - редактирование профиля

ViewResultActivity - просмотр результата опроса

MainAdapter - главный адаптер для всех фрагментов

AnswersEditAdapter - адаптер ответов к вопросам

QuestionModel - общая модель опроса

QuestModel - модель одного вопроса

ResponseModel - модель результата опроса на одного участника

ResultModel - модель рекомендаций к опросу

UserModel - модель пользователя приложения

провайдеры контента для карточек:

ChartProvider - провайдер контента для графика и выводимой рекомендации

EditQuestProvider - провайдер редактирования вопроса

EditResultContentProvider - провайдер редактирования рекомендаций к опросу

QuestionProvider - провайдер опроса для списка опросов

QuestProvider - провайдер одного вопроса при прохождении теста

RegistrationClientProvider - провайдер регистрации пользователя перед опросом

ResponseProvider - провайдер результата теста для списка результатов

DBHelper - класс для создания и работы с базами данных

NetworkDiscovery - поиск и регистрация сервиса в локальной сети

Connection - соединение между клиентской и сервисной частью

QuestionsManager - менеджер опросов

ResponsesManager - менеджер результатов опросов

UsersManager - менеджер пользователей приложения

* + 1. **Проблемы при разработке**

Главной проблемой при разработке данного приложения стало сложность синхронизации устройств без сети интернет. На изучение [1,2] и реализацию данного аспекта было затрачено наибольшее время. Несколько раз изменялась логика данной части. В конечном счете была реализована синхронизация по сети wi-fi, создавая точку доступа на одном из устройств. Данный способ оказался наиболее эффективным и быстро действенным.

* + 1. **Итоговый результат**

Результатом разработки стало приложение для создания и прохождения тестов на нескольких устройствах одновременно. Простой и интуитивно понятный интерфейс позволяет пользоваться данным приложениям как взрослым, так и детям, не зависимо от возраста.

# **Заключение**

В процессе разработки комплекса программ была методологии сбора и анализа данных в мед. учреждениях, рассмотрены доступные инструменты для создания приложения на базе android. Выявлены слабые и сильные стороны данной технологии. Так же во время создания игр были реализованы скрипты, которые можно использовать в будущих разработках для данной технологии.

Все поставленные задачи были выполнены. Результатом стали два независимых продукта для автоматизации процесса сбора и анализа данных в мед. учреждениях.

Оба продукта уже внедрены и активно используются в клинике для изучения когнитивных отклонений пациентов. В дальнейшем планируется продолжить работу с врачами кафедры неврологии и мануальной терапии университетской клиники Казань (Медико-санитарная часть ФГАОУ ВО «Казанский (Приволжский) федеральный университет»).

1. **Список используемых источников**
2. <https://developer.android.com/index.html>
3. <http://fandroid.info>
4. [www.cambridgecognition.com](http://www.cambridgecognition.com)
5. <https://www.nia.nih.gov/alzheimers/features/taking-technology-home-researchers-test-new-methods-detect-cognitive-change>
6. <http://www.cambridgecognition.com/products/digital-healthcare-technology/cantab-mobile/>
7. <https://www.medideal.ru/zlneuro/neuro-0014.shtml>
8. <https://wikipedia.org/wiki/Automated_Neuropsychological_Assessment_Metrics>
9. <http://www.cambridgecognition.com/cantab/>
10. <https://bmcneurol.biomedcentral.com/articles/10.1186/1471-2377-7-10>
11. <https://www.ncbi.nlm.nih.gov/pubmed/1789909>
12. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2699993/>
13. **Приложение**

Приложение 1:

package consultation.dr.com.myapplication;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

import android.support.v4.app.Fragment;

import android.support.v4.view.ViewPager;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.Toolbar;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Toast;

import consultation.dr.com.myapplication.adapters.MainAdapter;

import consultation.dr.com.myapplication.fragments.QuestionInfoFragment;

import consultation.dr.com.myapplication.fragments.QuestsFragment;

import consultation.dr.com.myapplication.fragments.ResultsFragment;

import consultation.dr.com.myapplication.models.QuestModel;

import consultation.dr.com.myapplication.models.QuestionModel;

import consultation.dr.com.myapplication.models.ResultModel;

import consultation.dr.com.myapplication.models.UserModel;

import consultation.dr.com.myapplication.util.QuestionsManager;

public class EditQuestionActivity extends AppCompatActivity implements View.OnClickListener,

MainService.ConnectListener, ViewPager.OnPageChangeListener {

private FloatingActionButton mAddQuest;

private QuestionsManager mQuestionManager;

private QuestionModel mQuestion;

private ViewPager mPager;

private QuestionInfoFragment mInfo;

private QuestsFragment mQuests;

private ResultsFragment mResults;

private int mUser;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_question);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

mAddQuest = (FloatingActionButton) findViewById(R.id.fab);

mAddQuest.setOnClickListener(this);

mAddQuest.hide();

mPager = (ViewPager) findViewById(R.id.pager);

MainService.connect(this, this);

}

private void setupPager() {

MainAdapter adapter = new MainAdapter(getSupportFragmentManager());

mInfo = new QuestionInfoFragment();

adapter.addFragment(mInfo, "описание");

mQuests = new QuestsFragment();

adapter.addFragment(mQuests, "вопросы");

mResults = new ResultsFragment();

adapter.addFragment(mResults, "рекомендации");

mPager.setAdapter(adapter);

mPager.addOnPageChangeListener(this);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_edit, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == R.id.action\_save) {

try {

save();

finish();

} catch (Exception e) {

Toast.makeText(this, "Ошибка сохраенения опроса: " + e.getMessage(), Toast.LENGTH\_LONG).show();

}

return true;

}

return super.onOptionsItemSelected(item);

}

public void save() throws Exception {

mInfo.accept();

mQuests.accept();

mResults.accept();

mQuestionManager.update(mQuestion, mUser, true);

}

@Override

public void onBackPressed() {

confirmExit();

}

public void confirmExit() {

AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("Внимание");

alert.setMessage("Сохранить изменения?");

alert.setPositiveButton("Да", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

try {

save();

finish();

} catch (Exception e) {

Toast.makeText(EditQuestionActivity.this, "Ошибка сохраенения опроса: " + e.getMessage(), Toast.LENGTH\_LONG).show();

}

}

});

alert.setNeutralButton("Отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

}

});

alert.setNegativeButton("Нет", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

finish();

}

});

alert.show();

}

@Override

public void onClick(View v) {

if(v.equals(mAddQuest)) {

MainAdapter adapter = (MainAdapter) mPager.getAdapter();

Fragment current = adapter.getItem(mPager.getCurrentItem());

if(current.equals(mQuests)) {

QuestModel quest = new QuestModel();

mQuests.addQuest(quest);

} else if(current.equals(mResults)) {

ResultModel result = new ResultModel();

mResults.addResult(result);

}

}

}

public QuestionModel getQuestion() {

return mQuestion;

}

@Override

public void onConnect(MainService service) {

mQuestionManager = service.getQuestionsManager();

UserModel user = service.getUsersManager().getCurrentUser();

if(user == null) {

Toast.makeText(this, "требуется авторизация", Toast.LENGTH\_LONG).show();

finish();

return;

}

Intent intent;

if((intent = getIntent()) != null && intent.hasExtra("id")) {

String id = intent.getStringExtra("id");

if(id == null || (mQuestion = mQuestionManager.getById(id)) == null

|| (mQuestion = QuestionsManager.cloneQuestion(mQuestion)) == null) {

mQuestion = mQuestionManager.newQuestion();

}

} else mQuestion = mQuestionManager.newQuestion();

mQuestion.from = user.login;

mUser = user.id;

setupPager();

}

@Override

public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

}

@Override

public void onPageSelected(int position) {

Fragment selected = ((MainAdapter)mPager.getAdapter()).getItem(position);

if(selected instanceof QuestsFragment || selected instanceof ResultsFragment) {

mAddQuest.show();

} else mAddQuest.hide();

}

@Override

public void onPageScrollStateChanged(int state) {

}

}

package consultation.dr.com.myapplication;

import android.content.DialogInterface;

import android.content.Intent;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.Toolbar;

import android.view.KeyEvent;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.ViewGroup;

import android.view.inputmethod.EditorInfo;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import consultation.dr.com.myapplication.models.UserModel;

import consultation.dr.com.myapplication.util.UsersManager;

public class LoginActivity extends AppCompatActivity implements MainService.ConnectListener {

private static final String FIELD\_REQUIRED = "поле должно содержать не менее 4х сивловов";

public static final int REQUEST\_CODE = 1;

private EditText mLoginView;

private EditText mPasswordView;

private UserModel mAdmin;

private UsersManager mManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

mLoginView = (EditText) findViewById(R.id.login);

mPasswordView = (EditText) findViewById(R.id.password);

mPasswordView.setOnEditorActionListener(new TextView.OnEditorActionListener() {

@Override

public boolean onEditorAction(TextView textView, int id, KeyEvent keyEvent) {

if (id == R.id.login || id == EditorInfo.IME\_NULL) {

attemptLogin();

return true;

}

return false;

}

});

MainService.connect(this, this);

}

private void attemptLogin() {

mLoginView.setError(null);

mPasswordView.setError(null);

String login = mLoginView.getText().toString();

String password = mPasswordView.getText().toString();

View focusView = null;

boolean cancel = false;

UserModel user = null;

if (password.length() < 4) {

mPasswordView.setError(FIELD\_REQUIRED);

focusView = mPasswordView;

cancel = true;

}

if (login.length() < 4) {

mLoginView.setError(FIELD\_REQUIRED);

focusView = mLoginView;

cancel = true;

} else {

user = mManager.findByLogin(login);

if(user == null) {

mLoginView.setError("пользователь не найден");

focusView = mLoginView;

cancel = true;

} else if(!cancel) {

if(!user.password.equals(password)) {

mPasswordView.setError("неправильный пароль");

focusView = mPasswordView;

cancel = true;

}

}

}

if (cancel) {

focusView.requestFocus();

} else {

setResult(user);

}

}

public void setResult(UserModel user) {

Intent intent = new Intent();

if(user != null) {

Bundle bundle = new Bundle();

user.put(bundle);

intent.putExtras(bundle);

}

setResult(user == null ? RESULT\_CANCELED : RESULT\_OK, intent);

finish();

}

public void setupRegistrationPass() {

AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("Пароль регистраций");

alert.setMessage("Требуется создать пароль для регистраций новых аккаунтов");

final EditText editText = new EditText(this);

alert.setView(editText);

alert.setCancelable(false);

alert.setPositiveButton("задать", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

String password = editText.getText().toString();

mAdmin = new UserModel();

mAdmin.login = "admin";

mAdmin.password = password;

mManager.addUser(mAdmin);

showSignInForm(null);

}

});

alert.show();

}

public void showSignInForm(View view) {

AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("Новый аккаунт");

if(view == null) {

view = getLayoutInflater().inflate(R.layout.sign\_in, null);

} else {

ViewGroup parent = (ViewGroup) view.getParent();

if(parent != null) parent.removeView(view);

}

alert.setView(view);

alert.setCancelable(false);

final View finalView = view;

alert.setPositiveButton("создать", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

EditText nameView = (EditText) finalView.findViewById(R.id.name);

EditText loginView = (EditText) finalView.findViewById(R.id.login);

EditText emailView = (EditText) finalView.findViewById(R.id.email);

EditText passView = (EditText) finalView.findViewById(R.id.password);

EditText passVerifyView = (EditText) finalView.findViewById(R.id.password\_verify);

String name = nameView.getText().toString();

String login = loginView.getText().toString();

String email = emailView.getText().toString();

String password = passView.getText().toString();

String passwordVerify = passVerifyView.getText().toString();

boolean cancel = false;

if(name.length() < 4) {

cancel = true;

nameView.setError(FIELD\_REQUIRED);

}

if(login.length() < 4) {

cancel = true;

loginView.setError(FIELD\_REQUIRED);

}

if(email.length() < 4) {

cancel = true;

emailView.setError(FIELD\_REQUIRED);

}

if(password.length() < 4) {

cancel = true;

passView.setError(FIELD\_REQUIRED);

}

if(passwordVerify.length() < 4) {

cancel = true;

passVerifyView.setError(FIELD\_REQUIRED);

}

if(mManager.findByLogin(login) != null) {

cancel = true;

loginView.setError("пользователь существует");

}

if(!password.equals(passwordVerify)) {

cancel = true;

passVerifyView.setError("пароли не совпадают");

}

if(cancel) {

showSignInForm(finalView);

} else {

UserModel user = new UserModel();

user.name = name;

user.login = login;

user.password = password;

user.email = email;

mManager.addUser(user);

setResult(user);

}

}

});

alert.setNegativeButton("отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

}

});

alert.show();

}

public void signIn() {

mAdmin = mManager.findByLogin("admin");

if(mAdmin == null) {

setupRegistrationPass();

} else requestRegistrationPass(null);

}

public void requestRegistrationPass(String error) {

final AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("Пароль регистраций");

alert.setMessage("Пароль для регистраций новых аккаунтов: ");

final EditText editText = new EditText(this);

if(error != null) editText.setError(error);

alert.setView(editText);

alert.setPositiveButton("продолжить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

String password = editText.getText().toString();

if(password.equals(mAdmin.password)) showSignInForm(null);

else {

requestRegistrationPass("Неправильный пароль");

}

}

});

alert.setNegativeButton("отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

}

});

alert.show();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_login, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == R.id.action\_exit) {

Intent intent = new Intent(this, MainService.class);

stopService(intent);

setResult(null);

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onConnect(final MainService service) {

mManager = service.getUsersManager();

final Button logInButton = (Button) findViewById(R.id.log\_in\_button);

logInButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View view) {

attemptLogin();

}

});

final Button signInButton = (Button) findViewById(R.id.sign\_in\_button);

signInButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View view) {

signIn();

}

});

final Button connectButton = (Button) findViewById(R.id.connect\_button);

connectButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View view) {

if(service.isConnected()) {

service.disconnect();

} else {

if(service.isFinding()) {

service.stopFinding();

connectButton.setText("Подключиться к сервису");

} else {

service.findServer();

connectButton.setText("Остановить поиск");

}

}

}

});

logInButton.post(new Runnable() {

@Override

public void run() {

boolean enable = !service.isConnected();

logInButton.setEnabled(enable);

signInButton.setEnabled(enable);

}

});

service.setOnCommunicationListener(new MainService.CommunicationListener() {

@Override

public void onConnected() {

logInButton.post(new Runnable() {

@Override

public void run() {

logInButton.setEnabled(false);

signInButton.setEnabled(false);

connectButton.setText("Отключиться");

}

});

}

@Override

public void onDisconnected() {

logInButton.post(new Runnable() {

@Override

public void run() {

logInButton.setEnabled(true);

signInButton.setEnabled(true);

connectButton.setText("Подключиться к сервису");

}

});

}

});

}

}

package consultation.dr.com.myapplication;

import android.content.Intent;

import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

import android.support.v4.app.Fragment;

import android.support.v4.view.ViewPager;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.Toolbar;

import android.view.View;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.Toast;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import consultation.dr.com.myapplication.adapters.MainAdapter;

import consultation.dr.com.myapplication.fragments.AnswersFragment;

import consultation.dr.com.myapplication.fragments.QuestionsFragment;

import consultation.dr.com.myapplication.models.QuestionModel;

import consultation.dr.com.myapplication.models.UserModel;

public class MainActivity extends AppCompatActivity implements View.OnClickListener, ViewPager.OnPageChangeListener, MainService.ConnectListener {

private ViewPager mPager;

private FloatingActionButton mAddQuestion;

private MainService mService;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

mAddQuestion = (FloatingActionButton) findViewById(R.id.fab);

mAddQuestion.setOnClickListener(this);

mPager = (ViewPager) findViewById(R.id.pager);

MainService.connect(this, this);

}

private void setupPager() {

MainAdapter adapter = new MainAdapter(getSupportFragmentManager());

adapter.addFragment(new QuestionsFragment(), "Опросы");

adapter.addFragment(new AnswersFragment(), "Журнал");

mPager.setAdapter(adapter);

mPager.addOnPageChangeListener(this);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == R.id.action\_exit) {

mService.closeServer();

mService.getUsersManager().setCurrentUser(null);

Intent intent = new Intent(this, LoginActivity.class);

startActivityForResult(intent, LoginActivity.REQUEST\_CODE);

return true;

} else if (id == R.id.action\_syn) {

JSONObject json = new JSONObject();

int user = mService.getUsersManager().getCurrentUser().id;

try {

json.put("user", user);

json.put("type", "syn");

JSONArray questions = new JSONArray();

for(QuestionModel question : mService.getQuestionsManager().getQuestions()) {

questions.put(question.toJSON());

}

json.put("questions", questions);

mService.getConnection().sendMessage(json);

Toast.makeText(this, "синхронизировано", Toast.LENGTH\_LONG).show();

} catch (JSONException e) {

Toast.makeText(this, "ошибка синхронизации" + e.toString(), Toast.LENGTH\_LONG).show();

}

} else if (id == R.id.action\_account) {

Intent intent = new Intent(this, UserActivity.class);

startActivity(intent);

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onClick(View v) {

if(v.equals(mAddQuestion)) {

Intent intent = new Intent(this, EditQuestionActivity.class);

startActivity(intent);

}

}

@Override

public void onConnect(MainService service) {

mService = service;

if(service.getUsersManager().getCurrentUser() == null) {

Intent intent = new Intent(this, LoginActivity.class);

startActivityForResult(intent, LoginActivity.REQUEST\_CODE);

} else {

service.startServer();

setupPager();

}

}

public MainService getService() {

return mService;

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

if(requestCode == LoginActivity.REQUEST\_CODE && resultCode == LoginActivity.RESULT\_OK) {

UserModel user = UserModel.fromBundle(data.getExtras());

mService.getUsersManager().setCurrentUser(user);

mService.startServer();

setupPager();

} else finish();

super.onActivityResult(requestCode, resultCode, data);

}

@Override

public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

}

@Override

public void onPageSelected(int position) {

Fragment selected = ((MainAdapter)mPager.getAdapter()).getItem(position);

if(selected instanceof QuestionsFragment) mAddQuestion.show();

else mAddQuestion.hide();

}

@Override

public void onPageScrollStateChanged(int state) {

}

}

package consultation.dr.com.myapplication;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.os.Looper;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.Toolbar;

import android.view.View;

import android.widget.Toast;

import com.dexafree.materialList.card.Card;

import com.dexafree.materialList.card.CardProvider;

import com.dexafree.materialList.view.MaterialListAdapter;

import com.dexafree.materialList.view.MaterialListView;

import org.json.JSONException;

import org.json.JSONObject;

import consultation.dr.com.myapplication.models.QuestModel;

import consultation.dr.com.myapplication.models.QuestionModel;

import consultation.dr.com.myapplication.models.ResponseModel;

import consultation.dr.com.myapplication.models.UserModel;

import consultation.dr.com.myapplication.provider.QuestProvider;

import consultation.dr.com.myapplication.provider.RegistrationClientProvider;

import consultation.dr.com.myapplication.util.QuestionsManager;

public class PlayQuestionActivity extends AppCompatActivity implements MainService.ConnectListener, View.OnClickListener, QuestionsManager.StopQuestionListener {

private MainService mService;

private QuestionModel mQuest;

private MaterialListAdapter mAdapter;

private MaterialListView mList;

private UserModel mUser;

private String mQid;

private int mUserId;

private int mCurrentQuest;

private long mStartTime;

private long mQuestStartTime;

private Card mCurrentCard;

private ResponseModel mResponse;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_play\_question);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

findViewById(R.id.fab).setOnClickListener(this);

mList = (MaterialListView) findViewById(R.id.questions\_list);

mAdapter = mList.getAdapter();

onConnect(MainService.getInstance());

mStartTime = System.currentTimeMillis();

mCurrentQuest = 0;

mResponse = new ResponseModel();

mResponse.question = QuestionsManager.cloneQuestion(mQuest);

mResponse.qid = mQid;

}

public void addRegistrationForm() {

mCurrentQuest = -1;

Card card = new Card.Builder(this)

.withProvider(new RegistrationClientProvider())

.setTitle("Анкетные данные")

.endConfig()

.build();

mAdapter.add(card);

mCurrentCard = card;

}

public boolean nextQuest() {

if (mCurrentCard != null) {

CardProvider provider = mCurrentCard.getProvider();

if (provider instanceof RegistrationClientProvider) {

RegistrationClientProvider info = (RegistrationClientProvider) provider;

if (!info.isValid()) {

return true;

} else {

mResponse.name = info.getName();

mResponse.age = info.getAge();

mResponse.isParent = info.isParent();

mResponse.isChild = info.isChild();

mCurrentQuest = 0;

if(mQuest.quests.size() > 0) {

addQuest(mQuest.quests.get(mCurrentQuest));

return true;

} else return false;

}

} else if (provider instanceof QuestProvider) {

QuestProvider quest = (QuestProvider) provider;

if (quest.getSelected() == -1) {

return true;

} else {

mResponse.question.quests.get(mCurrentQuest).selected = quest.getSelected();

mResponse.question.quests.get(mCurrentQuest).time = System.currentTimeMillis() - mQuestStartTime;

mCurrentQuest ++;

if(mCurrentQuest < mQuest.quests.size()) {

mCurrentCard = addQuest(mQuest.quests.get(mCurrentQuest));

return true;

} else return false;

}

}

}

return false;

}

public void showQuestions() {

mStartTime = System.currentTimeMillis();

mUser = mService.getUsersManager().getCurrentUser();

if(mUser == null) addRegistrationForm();

else if(mQuest.quests.size() > 0) mCurrentCard = addQuest(mQuest.quests.get(0));

}

public Card addQuest(QuestModel model) {

Card card = new Card.Builder(this)

.withProvider(new QuestProvider())

.setModel(model)

.setTitle("Вопрос " + (mCurrentQuest + 1) + " из " + mQuest.quests.size())

.endConfig()

.build();

mAdapter.add(card);

mQuestStartTime = System.currentTimeMillis();

return card;

}

@Override

public void onConnect(MainService service) {

mService = service;

Intent intent = getIntent();

if(intent != null && intent.hasExtra("id") && intent.hasExtra("qid")) {

String id = intent.getStringExtra("id");

mQid = intent.getStringExtra("qid");

mUserId = intent.getIntExtra("user", 0);

mQuest = service.getQuestionsManager().getById(id);

service.getQuestionsManager().addStopQuestionListener(this);

if(mQuest != null) {

showQuestions();

setTitle(mQuest.name);

} else Toast.makeText(this, "Ошибка запуска теста, тест не найден", Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onClick(View v) {

if(v.getId() == R.id.fab) {

if(!nextQuest()) {

mResponse.time = System.currentTimeMillis() - mStartTime;

if(mUser != null){

mResponse.name = mUser.name;

}

try {

mService.getResponsesManager().add(mResponse, mUserId, true);

if(mUser != null) {

Intent intent = new Intent(this, ViewResultActivity.class);

intent.putExtra("qid", mResponse.qid);

startActivity(intent);

}

finish();

} catch (JSONException e) {

Toast.makeText(this, "ошибка сохранения результатов " + e.getMessage(), Toast.LENGTH\_LONG).show();

}

}

}

}

@Override

public void onDestroy() {

super.onDestroy();

mService.getQuestionsManager().removeStopQuestionListener(this);

}

@Override

public void onBackPressed() {

if(mUser != null) {

JSONObject args = new JSONObject();

try {

args.put("type", "stop\_question");

args.put("id", mQuest.id);

MainService.getInstance().getConnection().sendMessage(args);

finish();

} catch (JSONException e) {

e.printStackTrace();

}

}

}

@Override

public void onStopQuestion(String id) {

if(id.equals(mQuest.id)) finish();

}

}

package consultation.dr.com.myapplication.fragments;

import android.content.DialogInterface;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.support.v7.app.AlertDialog;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import com.dexafree.materialList.card.Card;

import com.dexafree.materialList.card.OnActionClickListener;

import com.dexafree.materialList.card.action.TextViewAction;

import com.dexafree.materialList.view.MaterialListAdapter;

import com.dexafree.materialList.view.MaterialListView;

import consultation.dr.com.myapplication.EditQuestionActivity;

import consultation.dr.com.myapplication.R;

import consultation.dr.com.myapplication.adapters.AnswersEditAdapter;

import consultation.dr.com.myapplication.models.QuestModel;

import consultation.dr.com.myapplication.models.QuestionModel;

import consultation.dr.com.myapplication.provider.EditQuestProvider;

public class QuestsFragment extends Fragment {

private MaterialListAdapter mAdapter;

private QuestionModel mQuestion;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.quests, container, false);

MaterialListView mQuestionsList = (MaterialListView) rootView.findViewById(R.id.quests\_list);

mAdapter = mQuestionsList.getAdapter();

if(getActivity() instanceof EditQuestionActivity) {

mQuestion = ((EditQuestionActivity)getActivity()).getQuestion();

showQuests();

}

return rootView;

}

public void confirmRemove(final Card card) {

final QuestModel quest = (QuestModel) card.getTag();

AlertDialog.Builder alert = new AlertDialog.Builder(getActivity());

alert.setTitle("Внимание");

alert.setMessage("Удалить " + card.getProvider().getTitle());

alert.setPositiveButton("Удалить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

mQuestion.quests.remove(quest);

card.setDismissible(true);

card.dismiss();

}

});

alert.setNegativeButton("Отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

}

});

alert.show();

}

public void addQuest(final QuestModel quest) {

if(!mQuestion.quests.contains(quest)) mQuestion.quests.add(quest);

accept();

final AnswersEditAdapter adapter = new AnswersEditAdapter(quest);

String title = "Вопрос #" + (mAdapter.getItemCount() + 1);

Card card = new Card.Builder(getContext())

.setTag(quest)

.withProvider(new EditQuestProvider().setQuest(quest.description))

.setLayout(R.layout.quest\_edit\_item)

.setTitle(title)

.addAction(R.id.action\_add\_answer, new TextViewAction(getContext())

.setText("добавить ответ")

.setTextResourceColor(R.color.colorPrimary)

.setListener(new OnActionClickListener() {

@Override

public void onActionClicked(View view, Card card) {

adapter.addAnswer(new QuestModel.Answer());

}

}))

.addAction(R.id.action\_remove\_quest, new TextViewAction(getContext())

.setText("")

.setTextResourceColor(R.color.colorAccent)

.setListener(new OnActionClickListener() {

@Override

public void onActionClicked(View view, Card card) {

confirmRemove(card);

}

}))

.setAdapter(adapter)

.endConfig()

.build();

mAdapter.add(card);

}

public void accept() {

if(mAdapter == null) return;

for (int i = 0; i < mAdapter.getItemCount(); i ++) {

Card card = mAdapter.getCard(i);

assert card != null;

EditQuestProvider provider = (EditQuestProvider)card.getProvider();

QuestModel quest = (QuestModel) card.getTag();

assert quest != null;

quest.description = provider.getQuest();

AnswersEditAdapter adapter = (AnswersEditAdapter) provider.getAdapter();

adapter.accept();

}

}

public void showQuests() {

for(QuestModel quest : mQuestion.quests) addQuest(quest);

}

}

package consultation.dr.com.myapplication.fragments;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.support.v7.app.AlertDialog;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import com.dexafree.materialList.card.Card;

import com.dexafree.materialList.card.OnActionClickListener;

import com.dexafree.materialList.card.action.TextViewAction;

import com.dexafree.materialList.view.MaterialListAdapter;

import com.dexafree.materialList.view.MaterialListView;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import consultation.dr.com.myapplication.MainActivity;

import consultation.dr.com.myapplication.MainService;

import consultation.dr.com.myapplication.R;

import consultation.dr.com.myapplication.ViewResultActivity;

import consultation.dr.com.myapplication.models.ResponseModel;

import consultation.dr.com.myapplication.provider.ResponseProvider;

import consultation.dr.com.myapplication.util.ResponsesManager;

public class AnswersFragment extends Fragment implements ResponsesManager.ResponseListener {

private MaterialListAdapter mAdapter;

private ResponsesManager mResponsesManager;

private MaterialListView mQuestionsList;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.questions, container, false);

mQuestionsList = (MaterialListView) rootView.findViewById(R.id.questions\_list);

mAdapter = mQuestionsList.getAdapter();

MainService service = ((MainActivity)getActivity()).getService();

mResponsesManager = service.getResponsesManager();

mResponsesManager.addResponseListener(this);

showResponses(mResponsesManager.getResponses());

return rootView;

}

public void showResponses(Map<String, List<ResponseModel>> responses) {

for(Map.Entry<String, List<ResponseModel>> response : responses.entrySet()) {

addResponse(response.getValue());

}

}

public void addResponse(List<ResponseModel> response) {

Card card = new Card.Builder(getContext())

.setTag(response)

.withProvider(new ResponseProvider())

.setModel(response)

.addAction(R.id.action\_remove\_question, new TextViewAction(getContext())

.setListener(new OnActionClickListener() {

@Override

public void onActionClicked(View view, Card card) {

confirmRemove(card);

}

}))

.addAction(R.id.action\_view\_result, new TextViewAction(getContext())

.setText("просмотреть")

.setTextResourceColor(R.color.colorPrimary)

.setListener(new OnActionClickListener() {

@Override

public void onActionClicked(View view, Card card) {

view((List<ResponseModel>)card.getTag());

}

}))

.endConfig()

.build();

mAdapter.add(card);

}

public void view(List<ResponseModel> models) {

Intent intent = new Intent(getContext(), ViewResultActivity.class);

intent.putExtra("qid", models.get(0).qid);

startActivity(intent);

}

@Override

public void onDestroy() {

super.onDestroy();

if(mResponsesManager != null) mResponsesManager.removeResponseListener(this);

}

public void confirmRemove(final Card card) {

final String id = ((ResponseProvider)card.getProvider()).getId();

String message = "Удалить ";

message += ((ResponseProvider)card.getProvider()).getName();

AlertDialog.Builder alert = new AlertDialog.Builder(getActivity());

alert.setTitle("Внимание");

alert.setMessage(message);

alert.setPositiveButton("Удалить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

mResponsesManager.removeResponse(id, true);

}

});

alert.setNegativeButton("Отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

}

});

alert.show();

}

@Override

public void onResponseRemove(List<ResponseModel> model) {

Card card = findCardByQid(model.get(0).qid);

if(card != null) {

card.setDismissible(true);

card.dismiss();

}

}

@Override

public void onResponseAdd(ResponseModel model) {

Card card = findCardByQid(model.qid);

if(card == null) {

List<ResponseModel> models = new ArrayList<>();

models.add(model);

addResponse(models);

} else {

ResponseProvider provider = (ResponseProvider) card.getProvider();

provider.addModel(model);

}

mQuestionsList.scrollToPosition(0);

}

public Card findCardByQid(String qid) {

for(int i = 0; i < mAdapter.getItemCount(); i ++) {

Card card = mAdapter.getCard(i);

Object cardTag;

if(card != null && (cardTag = card.getTag()) != null) {

List<ResponseModel> models = (List<ResponseModel>) cardTag;

if(models.get(0).qid.equals(qid)) return card;

}

}

return null;

}

}

Приложение 2:

Main.php:

<?php

session\_start();

include ("bd.php");

if(!isset($\_SESSION["login"])):

header("location:index.php");

else:

$all = mysql\_fetch\_row(mysql\_query('SELECT COUNT(`id`) FROM `admins`'));

$all = $all[0];

$opross = mysql\_fetch\_row(mysql\_query('SELECT COUNT(`id`) FROM `opros`'));

$opross = $opross[0];

$views = mysql\_fetch\_row(mysql\_query('SELECT SUM(`view`) FROM `opros`'));

$views = $views[0];

$names = $\_SESSION["login"];

$onlineadmin = mysql\_fetch\_row(mysql\_query('SELECT SUM(`online`) FROM `admins`'));

$onlineadmin = $onlineadmin[0];

if(isset($\_GET['del'])){}else{

$views = 0;

}

//SELECT SUM(`column\_name`) FROM `table\_name`;

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Adminка</title>

<!-- Tell the browser to be responsive to screen width -->

<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">

<!-- Bootstrap 3.3.6 -->

<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">

<!-- Font Awesome -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css">

<!-- Ionicons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">

<!-- Theme style -->

<link rel="stylesheet" href="dist/css/AdminLTE.min.css">

<!-- AdminLTE Skins. Choose a skin from the css/skins

folder instead of downloading all of them to reduce the load. -->

<link rel="stylesheet" href="dist/css/skins/\_all-skins.min.css">

<!-- iCheck -->

<link rel="stylesheet" href="plugins/iCheck/flat/blue.css">

<!-- Morris chart -->

<link rel="stylesheet" href="plugins/morris/morris.css">

<!-- jvectormap -->

<link rel="stylesheet" href="plugins/jvectormap/jquery-jvectormap-1.2.2.css">

<!-- Date Picker -->

<link rel="stylesheet" href="plugins/datepicker/datepicker3.css">

<!-- Daterange picker -->

<link rel="stylesheet" href="plugins/daterangepicker/daterangepicker.css">

<!-- bootstrap wysihtml5 - text editor -->

<link rel="stylesheet" href="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.min.css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body class="hold-transition skin-blue sidebar-mini">

<div class="wrapper">

<header class="main-header">

<!-- Logo -->

<a href="main.php" class="logo">

<!-- mini logo for sidebar mini 50x50 pixels -->

<span class="logo-mini"><b>A</b>ка</span>

<!-- logo for regular state and mobile devices -->

<span class="logo-lg"><b>Admin</b>ка</span>

</a>

<!-- Header Navbar: style can be found in header.less -->

<nav class="navbar navbar-static-top">

<!-- Sidebar toggle button-->

<a href="#" class="sidebar-toggle" data-toggle="offcanvas" role="button">

<span class="sr-only">Toggle navigation</span>

</a>

<div class="navbar-custom-menu">

<ul class="nav navbar-nav">

<!-- User Account: style can be found in dropdown.less -->

<li class="dropdown user user-menu">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<img src="<?= $\_SESSION['avatar'] ?>" class="user-image" alt="User Image">

<span class="hidden-xs"><?php print ($\_SESSION['login']); ?></span>

</a>

<ul class="dropdown-menu">

<!-- User image -->

<li class="user-header">

<img src="<?= $\_SESSION['avatar'] ?>" class="img-circle" alt="User Image">

<p>

<?php print ($\_SESSION['login']); ?> - <?php echo $\_SESSION['dolzhnost']; ?>

<small>neiroTest. 2017</small>

</p>

</li>

<!-- Menu Body -->

<li class="user-body">

<div class="row">

<div class="pull-right">

<a href="edit.php" class="btn btn-default btn-flat">Редактировать профиль</a>

<a href="logout.php" class="btn btn-default btn-flat">Sign out</a>

</div>

</li>

</ul>

</li>

</ul>

</div>

</nav>

</header>

<!-- Left side column. contains the logo and sidebar -->

<aside class="main-sidebar">

<!-- sidebar: style can be found in sidebar.less -->

<section class="sidebar">

<!-- Sidebar user panel -->

<div class="user-panel">

<div class="pull-left image">

<img src="<?= $\_SESSION['avatar'] ?>" class="img-circle" alt="User Image">

</div>

<div class="pull-left info">

<p><a href="edit.php"> <?php print ($\_SESSION['login']); ?> </a></p>

<a href="#"><i class="fa fa-circle text-success"></i> Online</a>

</div>

</div>

<!-- sidebar menu: : style can be found in sidebar.less -->

<ul class="sidebar-menu">

<li class="header">Главное меню</li>

<li class="active treeview">

<a href="#">

<i class="fa fa-dashboard"></i> <span>Опросы</span>

<span class="pull-right-container">

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<?php

$sql = mysql\_query("SELECT \* FROM opros where (author='$names')", $db);

while ($result = mysql\_fetch\_array($sql)) {

echo '<li class="active"><a href="opros.php/?hash='.$result['hash'].'"><i class="fa fa-circle-o"></i> '.$result['name'].'</a></li>';

}

?>

</ul>

</li>

</ul>

</section>

<!-- /.sidebar -->

</aside>

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Content Header (Page header) -->

<section class="content-header">

<h1>

ПУ

<small>Админ меню</small>

</h1>

</section>

<!-- Main content -->

<section class="content">

<!-- Small boxes (Stat box) -->

<div class="row">

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-aqua">

<div class="inner">

<h3><?php echo $opross; ?></h3>

<p>Опросы</p>

</div>

<div class="icon">

<i class="ion ion-bag"></i>

</div>

<a href="allurl.php" class="small-box-footer"> <i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<?php

if ($\_SESSION['level'] > 776){

echo '<div class="col-lg-3 col-xs-6">';

echo '<div class="small-box bg-yellow">';

echo '<div class="inner">';

echo '<h3>'.$all.'</h3>';

echo '<p>Админов зарегестрировано</p>';

echo '</div>';

echo '<div class="icon">';

echo ' <i class="ion ion-person-add"></i>';

echo '</div>';

echo '<a href="/admins.php" class="small-box-footer"><i class="fa fa-arrow-circle-right"></i></a>';

echo '</div>';

echo '</div>';

}else{

echo '<div class="col-lg-3 col-xs-6">';

echo '<div class="small-box bg-yellow">';

echo '<div class="inner">';

echo '<h3>'.$onlineadmin.'/'.$all.'</h3>';

echo '<p>Админов онлайн</p>';

echo '</div>';

echo '<div class="icon">';

echo ' <i class="ion ion-person-add"></i>';

echo '</div>';

echo '<a href="/admins.php" class="small-box-footer"><i class="fa fa-arrow-circle-right"></i></a>';

echo '</div>';

echo '</div>';

}

?>

<!-- ./col -->

<div class="col-lg-3 col-xs-6">

<!-- small box -->

<div class="small-box bg-red">

<div class="inner">

<h3><?php echo $views ?></h3>

<p>Просмотров</p>

</div>

<div class="icon">

<i class="ion ion-pie-graph"></i>

</div>

<a href="#" class="small-box-footer"><i class="fa fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

</div>

<!-- /.row -->

<!-- Main row -->

<div class="row">

</section>

<!-- right col -->

</div>

<!-- /.row (main row) -->

</section>

<!-- /.content -->

</div>

<!-- /.content-wrapper -->

<footer class="main-footer">

<div class="pull-right hidden-xs">

<b>Version</b> 0.0.1

</div>

<strong>Copyright &copy; 2017 Опросник.</strong> All rights

reserved.

</footer>

</div>

<!-- /.tab-pane -->

<!-- Stats tab content -->

<!-- jQuery 2.2.3 -->

<script src="plugins/jQuery/jquery-2.2.3.min.js"></script>

<!-- jQuery UI 1.11.4 -->

<script src="https://code.jquery.com/ui/1.11.4/jquery-ui.min.js"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->

<script>

$.widget.bridge('uibutton', $.ui.button);

</script>

<!-- Bootstrap 3.3.6 -->

<script src="bootstrap/js/bootstrap.min.js"></script>

<!-- Morris.js charts -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min.js"></script>

<script src="plugins/morris/morris.min.js"></script>

<!-- Sparkline -->

<script src="plugins/sparkline/jquery.sparkline.min.js"></script>

<!-- jvectormap -->

<script src="plugins/jvectormap/jquery-jvectormap-1.2.2.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-world-mill-en.js"></script>

<!-- jQuery Knob Chart -->

<script src="plugins/knob/jquery.knob.js"></script>

<!-- daterangepicker -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.11.2/moment.min.js"></script>

<script src="plugins/daterangepicker/daterangepicker.js"></script>

<!-- datepicker -->

<script src="plugins/datepicker/bootstrap-datepicker.js"></script>

<!-- Bootstrap WYSIHTML5 -->

<script src="plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.all.min.js"></script>

<!-- Slimscroll -->

<script src="plugins/slimScroll/jquery.slimscroll.min.js"></script>

<!-- FastClick -->

<script src="plugins/fastclick/fastclick.js"></script>

<!-- AdminLTE App -->

<script src="dist/js/app.min.js"></script>

<!-- AdminLTE dashboard demo (This is only for demo purposes) -->

<script src="dist/js/pages/dashboard.js"></script>

<!-- AdminLTE for demo purposes -->

<script src="dist/js/demo.js"></script>

</body>

</html>

<?php endif; ?>

Edit.php:

<?php

session\_start();

include ("bd.php");

if(!isset($\_SESSION["login"])):

header("location:index.php");

else:

if(!isset($\_POST['name'])){

}else{

$name = $\_POST['name'];

$names = $\_SESSION['login'];

$result = mysql\_query("UPDATE admins SET fio='$name' WHERE login='$names'",$db);

$\_SESSION['fio'] = $name;

header("location:main.php");

}

?>

<link rel="stylesheet" href="list.css" type="text/css"/>

<form class="form-wrapper cf" method="post" action="edit.php">

<input type="text" name="name" value="<?= $\_SESSION['fio']?>" required>

<button type="submit">Изменить</button>

</form>

<form class="form-wrapper cf" method="post" action="upload.php" enctype= "multipart/form-data">

<img src="<?= $\_SESSION['avatar']?>" alt="Аватар" width="150" height="150" />

<input type='file' name='filename'/>

<button type="submit">Изменить</button>

</form>

<?php endif; ?>

Allurl.php:

<link rel="stylesheet" href="list.css" type="text/css"/>

<?php

include ("bd.php");// файл bd.php должен быть в той же папке, что и все остальные, если это не так, то просто измените путь

session\_start();// вся процедура работает на сессиях. Именно в ней хранятся данные пользователя, пока он находится на сайте. Очень важно запустить их в самом начале странички!!!

$lvl = $\_SESSION['level'];

$names = $\_SESSION['login'];

if(isset($\_POST['add'])){

$namess = $\_POST['nameop'];

function generatePassword($length = 12){

$chars = 'abdefhiknrstyzABDEFGHKNQRSTYZ23456789';

$numChars = strlen($chars);

$string = '';

for ($i = 0; $i < $length; $i++) {

$string .= substr($chars, rand(1, $numChars) - 1, 1);

}

return $string;

}

$hash = generatePassword(12);

$addurl = $\_POST['add'];

$name = $\_SESSION['login'];

$result2 = mysql\_query ("INSERT INTO opros (url,name,hash,author) VALUES('$addurl','$namess','$hash','$name')");

// Проверяем, есть ли ошибки

if ($result2=='TRUE')

{

header("Location: http://".$\_SERVER['HTTP\_HOST']."/allurl.php");

}

else{

exit ("Error");

}

}

if(isset($\_POST['del'])){

$sql = mysql\_query("SELECT \* FROM `opros`", $db);

$result = mysql\_fetch\_array($sql);

$temp = $\_POST['del'];

$sqli = mysql\_query("DELETE FROM opros where (id='$temp')", $db);

//header("Location: localhost/addurl.php");

header("Location: http://".$\_SERVER['HTTP\_HOST']."/allurl.php");

exit;

}

if ($lvl > 776)

{

$sql = mysql\_query("SELECT \* FROM `opros`", $db);

echo '<ol class="rounded">';

while ($result = mysql\_fetch\_array($sql)) {

echo '<li><a href="opros.php/?hash='.$result['hash'].'">ID:'.$result['id'].' | Название: '.$result['name'].' | Ссылка на форму: '.$result['url'].' |Просмотров: '.$result['view'].' |Создал: '.$result['author'].'</a></li>';

}

}else{

$sql = mysql\_query("SELECT \* FROM opros where (author='$names')", $db);

echo '<ol class="rounded">';

while ($result = mysql\_fetch\_array($sql)) {

echo '<li><a href="opros.php/?hash='.$result['hash'].'">ID:'.$result['id'].' | Название: '.$result['name'].' |Ссылка на форму: '.$result['url'].' |Просмотров: '.$result['view'].'</a></li>';

}

}

?>

<form class="form-wrapper cf" method="post" action="allurl.php">

<input type="text" name="nameop" placeholder="Название опроса..." required>

<input type="text" name="add" placeholder="Url Google формы..." required>

<button type="submit">Добавить</button>

</form>

<?php

if ($\_SESSION['level'] > 776){

echo '<form class="form-wrapper cf" method="post" action="allurl.php">';

echo ' <input type="text" name="del" placeholder="ID удаления..." required>';

echo '<button type="submit">Удалить</button>';

echo '</form>';

}

?>

1. R. E. Schlegel and K. Gilliland, “Development and quality assurance of computer-based assessment batteries,” Archives of Clinical Neuropsychology, vol.22, no.1, pp.49–61, 2007. [↑](#footnote-ref-1)
2. (Bartram D, Bayliss R. Automated testing: past, present and future. J Occup Psychol. 1984; 57:221– 237. Vincent KR. Semi-automated full battery. J Clin Psychol. 1980;36:437–446) [↑](#footnote-ref-2)
3. (Beaumont JG. The validity of the Category Test administered by on-line computer. J Clin Psychol.1975; 31:458–462.) [↑](#footnote-ref-3)
4. (Mercer WN, Harrell EH, Miller DC, Childs HW, Rockers DM. Performance of brain-injured versus healthy adults on three versions of the category test. Clin Neuropsychol. 1997; 11:174–179.) [↑](#footnote-ref-4)
5. (Schatz P1, Browndyke J. Applications of computer-based neuropsychological assessment. J Head Trauma Rehabil. 2002 Oct;17(5):395-410.) [↑](#footnote-ref-5)
6. (Werner P, Korczyn AD. Willingness to use computerized systems for the diagnosis of dementia. Testing a theoretical model in an Israeli sample.Alzheimer Dis Assoc Disord. 2012;26(2):171-178) [↑](#footnote-ref-6)
7. (Kane RL, Roebuck-Spencer T, Short P, Kabat M, Wilken J. Identifying and monitoring cognitive deficits in clinical populations using automated neuropsychological assessment metrics (ANAM) tests. Arch Clin Neuropsychol. 2007;22 suppl 1: S115-S126) [↑](#footnote-ref-7)
8. (Levinson D, Reeves D, Watson J, Harrison M. Automated neuropsychological assessment metrics (ANAM) measures of cognitive effects of Alzheimer’s disease. Arch Clin Neuropsychol. 2005;20(3):403-408.) [↑](#footnote-ref-8)
9. (Kabat MH, Kane RL, Jefferson AL, DiPino RK. Construct validity of selected automated neuropsychological assessment metrics (ANAM) battery measures. Clin Neuropsychol. 2001; 15(4):498-507.) [↑](#footnote-ref-9)
10. (Johnson DR, Vincent AS, Johnson AE, Gilliland K, Schlegel RE. Reliability and construct validity of the automated neuropsychological assessment metrics (ANAM) mood scale. Arch Clin Neuropsychol. 2008;23(1):73-85.) [↑](#footnote-ref-10)
11. (Zygouris S, Tsolaki M omputerized cognitive testing for older adults: a review. Am J Alzheimers Dis Other Demen. 2015 Feb;30(1):13-28. doi: 10.1177/1533317514522852.) [↑](#footnote-ref-11)
12. (Sahakian BJ, Owen AM. Computerized assessment in neuropsychiatry using CANTAB: discussion paper. J Royal Soc Med. 1992;85(7):399-402) [↑](#footnote-ref-12)
13. (Lowe C, Rabbitt P. Test/re-test reliability of the CANTAB and ISPOCD neuropsychological batteries: theoretical and practical issues. Neuropsychologia. 1998;36(9):915-923) [↑](#footnote-ref-13)
14. (E´gerha´zi A, Berecz R, Barto´ k E, Degrell I. Automated neuropsychological test battery (CANTAB) in mild cognitive impairment and in Alzheimer’s disease.Prog Neuropsychopharmacol Biol Psychiatry. 2007;31(3):746-751) [↑](#footnote-ref-14)
15. (Crooks VC, Parsons TD, Buckwalter JG. Validation of the cognitive assessment of later life status (CALLS) instrument: a computerized telephonic measure. BMC Neurology. 2007; 7:10) [↑](#footnote-ref-15)
16. (Veroff AE, Cutler NR, Sramek JJ, Prior PL, Mickelson W, Hartman JK. A new assessment tool for neuropsychopharmacologic research: the computerized neuropsychological test battery. J Geriatr Psychiatry Neurol. 1991; 4(4):211-217; Cutler NR, Shrotriya RC, Sramek JJ, et al. The use of the computerized neuropsychological test battery (CNTB) in an efficacy and safety trial of BMY 21,502 in Alzheimer’s disease.Ann N Y Acad Sci. 1993; 695:332-336.) [↑](#footnote-ref-16)
17. (Veroff AE, Bodick NC, Offen WW, Sramek JJ, Cutler NR. Efficacy of xanomeline in Alzheimer disease: cognitive improvement measured using the computerized neuropsychological test battery (CNTB).Alzheimer Dis Assoc Disord. 1998;12(4):304-312) [↑](#footnote-ref-17)
18. (Dwolatzky T, Whitehead V, Doniger GM, et al. Validity of a novel computerized cognitive battery for mild cognitive impairment. BMC Geriatrics. 2003; 3:4.) [↑](#footnote-ref-18)
19. (Dwolatzky T, Dimant L, Simon ES, Doniger GM. Validity of a short computerized assessment battery for moderate cognitive impairment and dementia. Int Psychogeriatr. 2010; 22(5):795-803; Fillit HM, Simon ES, Doniger GM, Cummings JL. Practicality of a computerized system for cognitive assessment in the elderly. Alzheimers Dementia. 2008; 4(1):14-21; Neurotrax Corporation. Mindstreams Cognitive Health Assessment. Bellaire, TX: NeuroTrax Corporation; 2003) [↑](#footnote-ref-19)
20. (Dwolatzky T, Dimant L, Simon ES, Doniger GM. Validity of a short computerized assessment battery for moderate cognitive impairment and dementia. Int Psychogeriatr. 2010; 22(5):795-803.) [↑](#footnote-ref-20)
21. (Wild K, Howieson D, Webbe F, Seelye A, Kaye J. The status of computerized cognitive testing in aging: a systematic review. Alzheimers Dementia. 008;4(6):428-437) [↑](#footnote-ref-21)
22. (Gualtieri CT, Johnson LG. Reliability and validity of a computerized neurocognitive test battery, CNS vital signs. Arch Clin Neuropsychol. 2006;21(7):623-643.) [↑](#footnote-ref-22)
23. (Darby DG, Pietrzak RH, Fredrickson J, et al. Intraindividual cognitive decline using a brief computerized cognitive screening test. Alzheimers Dement. 2012;8(2):95-104; Lim YY, Ellis KA, Harrington K, et al. Use of the cogstate brief battery in the assessment of Alzheimer’s disease related cognitive impairment in the Australian imaging, biomarkers and lifestyle (AIBL) study. J Clin Exp Neuropsychol. 2012;34(4):345-358) [↑](#footnote-ref-23)
24. (Lim YY, Ellis KA, Harrington K, et al. Use of the cogstate brief battery in the assessment of Alzheimer’s disease related cognitive impairment in the Australian imaging, biomarkers and lifestyle (AIBL) study. J Clin Exp Neuropsychol. 2012;34(4):345-358) [↑](#footnote-ref-24)
25. (Saxton J, Morrow L, Eschman A, Archer G, Luther J, Zuccolotto A. Computer assessment of mild cognitive impairment. Postgrad Med. 2009;121(2):177-185) [↑](#footnote-ref-25)
26. (Cambridge Cognition. CANTABmobile: Government backs new high tech early dementia assessment service to reduce time to diagnosis from 18 months to 3 months CANTABmobile. 2012.) [↑](#footnote-ref-26)
27. (Trustram Eve C, de Jager CA. Piloting and validation of a novel self-administered online cognitive screening tool in normal older persons: the cognitive function test. Int J Geriatr Psychiatry. 2014;29(2):198-206) [↑](#footnote-ref-27)
28. (Rafii M, Taylor C, Coutinho A, Kim K, Galasko D. Comparison of the memory performance index with standard neuropsychological measures of cognition. Am J Alzheimers Dis Other Dement. 2011; 26(3):235-239). [↑](#footnote-ref-28)
29. (Kingsbury R, Pachana NA, Humphreys M, Tehan G, Byrne GJA. Utility of a computerised cognitive screen in MCI and depression in an older population. Aust J Rehabil Counsell. 2010; 16(1):14-26; Kluger BM, Saunders LV, Hou W, et al. A brief computerized self-screen for dementia. J Clin Exp Neuropsychol. 2009;31(2):234-244; Inoue M, Jinbo D, Nakamura Y, Taniguchi M, Urakami K. Development and evaluation of a computerized test battery for alzheimer’s disease screening in community-based settings. Am JAlzheimers Dis Other Dement. 2009;24(2):129-135) [↑](#footnote-ref-29)