

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерные технологии

«Допущен к защите»
Заведующий кафедрой Курманбаев З.К.
д. ф.-м. н., проф.
(Ф.И.О., ученая степень, звание)
« » 20 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка структуры базы данных
процесса формирования строк

Специальность 5В070400 ВТ.ИИ

Выполнил (а) Шайримова А.К. ПО-12-2
(Фамилия и инициалы) группа

Научный руководитель Дубжир С.Б. доцент, к.ф.-м.н.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Джансеген А.Ч., к.т.н., доцент
(Фамилия и инициалы, ученая степень, звание)
АЧ « 05 » 05 2016 г.
(подпись)

по безопасности жизнедеятельности:

Приходков Н.Г. д.х.н., профессор
(Фамилия и инициалы, ученая степень, звание)
НГ « 21 » 04 2016 г.
(подпись)

по применению вычислительной техники:

Дубжир С.Б., к.ф.-м.н., доцент
(Фамилия и инициалы, ученая степень, звание)
СБ « 31 » 05 2016 г.
(подпись)

Нормоконтролер: Дубжир С.Б., к.ф.-м.н., доцент
(Фамилия и инициалы, ученая степень, звание)
СБ « 31 » 05 2016 г.
(подпись)

Рецензент: Сейтаов К.А., к.п.н.
(Фамилия и инициалы, ученая степень, звание)
« » 20 г.
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерные технологии

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Нойтруша Аибек Кайырбекулы
(фамилия, имя, отчество)

Тема проекта Разработка структуры базы данных
приказа артиллерийских ствольных

утверждена приказом ректора № 21 от «10» марта 2016 г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка структуры базы данных
приказа артиллерийских ствольных

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Визуализация
2. Проектирование базы данных
3. Разработка структуры базы данных
4. Аналитические обоснования проекта
5. Ведомость принадлежности

Перечень графического материала (с точным указанием обязательных чертежей)

Рисунки структуры приложения
 Рисунки инструментов БД
 Рисунки, которые описывают требования к приложению
 Рисунки и таблицы БД
 Таблицы матрицы функционального обоснования
 Рисунки, описывающие безопасность взаимодействия

Рекомендуемая основная литература

Норман Т., Бен К. Базы данных. Проектирование, нормирование и сопровождение. Теория и практика. 3-е изд. Изд-во "Вильямс", 2003, 1436 с.
 Лемезнов А. В. Объектно-ориентированный дизайн и проектирование с использованием UML и IBM Rational Rose, 2006, 319 с.

Консультанты по проекту с указанием относящихся к ним разделов

| Раздел | Консультант | Сроки | Подпись |
|-------------------|----------------|------------------|-----------|
| БД | Приходько И.Г. | 01.04 - 21.04.16 | [Подпись] |
| Экранная часть | Безруков А.И. | 01.04 - 05.05.16 | [Подпись] |
| Базы данных | Иванов С.Б. | 24.02 - 16 | [Подпись] |
| Проектирование БД | Иванов С.Б. | 05.03.16 | [Подпись] |
| Разработка БД | Иванов С.Б. | 14.04.16 | [Подпись] |
| Программирование | Иванов С.Б. | 31.05.16 | [Подпись] |

ГРАФИК
подготовки дипломного проекта

| № п/п | Наименование разделов, перечень разрабатываемых вопросов | Сроки представления руководителю | Примечание |
|-------|--|----------------------------------|------------|
| 1 | Общие рисунки Постановка задачи дипломной работы Обзор предметной области | 24.02.16 | |
| 2 | Проектирование базы данных Предварительное проектирование Предварительные атрибуты в сс-схеме Составление рекурсивных отношений Технология работы Рассет места для хранения данных ИМД диаграммы Использование тематичеки программы информация и обеспечения работы СУБД | 05.03.16 | |
| 3 | Разработка базы данных Подготовка к работе Логические проектирование Панель управления интернет-базы данных | 14.04.16 | |
| 4 | Заключительное обоснование проекта | 05.05.16 | |
| 5 | Реферат статьи Иллюстрация работы | 21.04.16 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата выдачи задания «14» сентября 2015 г.

Заведующий кафедрой _____ Курманбаев З.К.
(подпись) (Фамилия и инициалы)

Руководитель _____ Зубшир С.Б.
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____ Шаймурзаев А.К.
(подпись) (Фамилия и инициалы)

Аннотация

В данной дипломной работе рассмотрена разработка базы данных «Прогноз футбольных событий» для мобильного приложения платформы *Android*.

Разработка базы данных осуществлялось при помощи встроенной СУБД SQLite. В ходе написания дипломной работы, была разработана общая структура базы данных.

Расчеты искусственного и естественного освещения, организация рабочего места показаны в части по безопасности жизнедеятельности.

В технико-экономической части были рассмотрены расчеты расходов по осуществлению проекта, а также прибыли от внедряемой системы.

Annotation

The mobile app "forecast football events" on android platform was considered in this thesis work.

Database development was carried out using the built-in database SQLite. In the course of writing the thesis, the overall structure of the database has been developed.

Calculations of natural and artificial lighting of workplaces are shown in part on life safety.

As part of the technical and economic calculations have been considered for the implementation of the project costs, and profits from the system being implemented.

Аңдатпа

Бұл дипломдық жұмыста *Android* платформасындағы мобильдік қосымшаға «Футбол оқиғалар болжамы» деректер қорын жобалау қарастырылған.

Деректер қорын жобалау кіріктірілген ДҚБЖ SQLite көмегімен жүзеге асырылды. Дипломдық жұмыс орындау барысында деректер қорының жалпы құрылымы әзірленді.

Жасанды және табиғи жарықтандыру есебі, жұмыс орнын ұйымдастыруы тіршілік қауіпсіздігі бөлімінде көрсетілген.

Техникалық-экономикалық бөлімінде шығыстардың есебі бойынша жобаны жүзеге асыру, сондай-ақ енгізілетін жүйенің пайдасы қарастырылған.

Содержание

| | |
|---|----|
| Введение..... | 12 |
| 1. Общее описание..... | 13 |
| 1.1 Постановка задачи дипломной работы..... | 13 |
| 1.2 Обзор предметной области..... | 13 |
| 1.2.1 Модели и методы прогнозирования..... | 13 |
| 1.2.2 Исследование букмекерских контор..... | 17 |
| 1.2.3 Анализ аналогичных приложений..... | 19 |
| 1.2.4 Анализ факторов прогнозирования..... | 24 |
| 2 Проектирование базы данных..... | 28 |
| 2.1 Предварительное проектирование..... | 28 |
| 2.2 Определение атрибутов всех сущностей..... | 29 |
| 2.3 Составление реляционных отношений..... | 30 |
| 2.4 Алгоритм работы..... | 31 |
| 2.5 Расчет места для хранения данных..... | 33 |
| 2.6 UML диаграммы..... | 34 |
| 2.7 Используемые технологии программирования и обоснования выбора СУБД..... | 38 |
| 3 Разработка базы данных..... | 43 |
| 3.1 Подготовка к работе..... | 43 |
| 3.2 Логическое проектирование..... | 45 |
| 3.3 Пользовательский интерфейс базы данных..... | 48 |
| 4 Экономическое обоснование проекта..... | 55 |
| 4.1 Руководство к работе..... | 55 |
| 4.2 Расчет вычисления расходов на разработку проекта..... | 55 |
| 4.3 Расчет цены программного продукта..... | 63 |
| 4.4 Вывод по технико–экономической части..... | 64 |
| 5 Безопасность жизнедеятельности..... | 65 |
| 5.1 Анализ условий труда..... | 65 |
| 5.2 Организация рабочего места..... | 70 |
| 5.3 Расчет искусственного освещения..... | 75 |
| Заключение..... | 79 |
| Список литературы..... | 80 |
| Приложение А..... | 81 |

Введение

В наши дни, неотъемлемой частью любого спортивного события считаются спортивные прогнозы. Среди людей, интересующимся различным видом спорта трудно отыскать человека, который не считался бы азартным.

Бурное развитие в мире спорта привело к появлению немалого количества контор, которые быстрыми темпами набирают популярность. Букмекер – это заведение, в котором принимаются ставки на спортивные события. Широко известными из букмекерских контор в Казахстане считаются: Olimp, Profit и Tennisi. Все эти букмекерские конторы имеют необходимые лицензии и могут принимать ставки на спортивные события. Много людей пользуются услугами букмекерских контор и рассчитывают получить прибыль. Они основываются на предположениях о некоем спортивном событии. В связи с заметным увеличением интереса многих людей получить прибыль в букмекерских конторах был разработан данный проект, который поможет людям достигнуть успеха.

Проделанный анализ позволил обнаружить потребность прогнозирования событий в данной сфере. В сфере ставок в букмекерских конторах отсутствует система прогноза ставок на футбольные события. Следуя вышесказанному, тема дипломной проекта является актуальной.

Целью данной дипломной работы является проектирование и разработка базы данных «Прогноза футбольных событий».

Для решения поставленной цели, были необходимы следующие задачи: исследовать предметную область в сфере ставок в букмекерских конторах, изучить структуру базы данных «Прогноз футбольных событий», произвести полный обзор аналогов на рынке, выбрать и обосновать информационные технологии, разработать структуру базы данных, сделать расчеты экономической части, проделать анализ вредных производственных факторов на рабочем месте.

Решение поставленных задач является необходимой базой для разработки мобильного приложения под платформу Android.

1. Общее описание

1.1 Постановка задачи дипломной работы

В данной дипломной работе требуется разработать структуру базы данных «Прогноз футбольных ставок», которая будет содержать данные о ставках, истории ставок зарегистрировавшихся пользователей. База данных должна предусматривать отражение предоставляемых ставок, истории ставок, пользователей, которые имеются в базе. Целесообразно иметь необходимые данные о ставках, такие как информации матча (т.е. какие команды будут играть и дата матча), «Win1», «X», «Win2». Далее в базе данных должно учитываться история ставок пользователей. Прежде чем иметь историю ставок, необходимо зарегистрироваться.

Для достижения поставленной задачи база данных «Прогноз футбольных ставок» должна предоставлять следующие возможности:

- хранение данных о пользователях (логин, пароль и email);
- хранение данных о ставках (матч, «Win1», «X», «Win2»);
- хранение данных об истории ставок (логин, матч, ставка).

При работе с базой данных любой пользователь имеет возможность решать нижеуказанные задачи:

- зарегистрироваться в системе;
- хранить данные о ставках в истории ставок;
- посмотреть данные о матче (коэффициенты на победу, поражение, ничью).

1.2 Обзор предметной области

1.2.1 Модели и методы прогнозирования

Методом прогнозирования называется сложный способ, совокупность простых способов, которые направлены на осуществления прогноза, в общем. А методикой называется упорядоченная совокупность способов, операций, процедур, а также правил изучения на основе конкретного сочетания некоторого количества методов.

Моделью прогнозирования является простой образ объекта, отражающий его наиболее основные характеристики, исходя из точки зрения анализа.

Методы прогнозирования делятся на два вида: интуитивные и формализованные методы. На рисунке 1.1 изображена классификация методов прогнозирования.

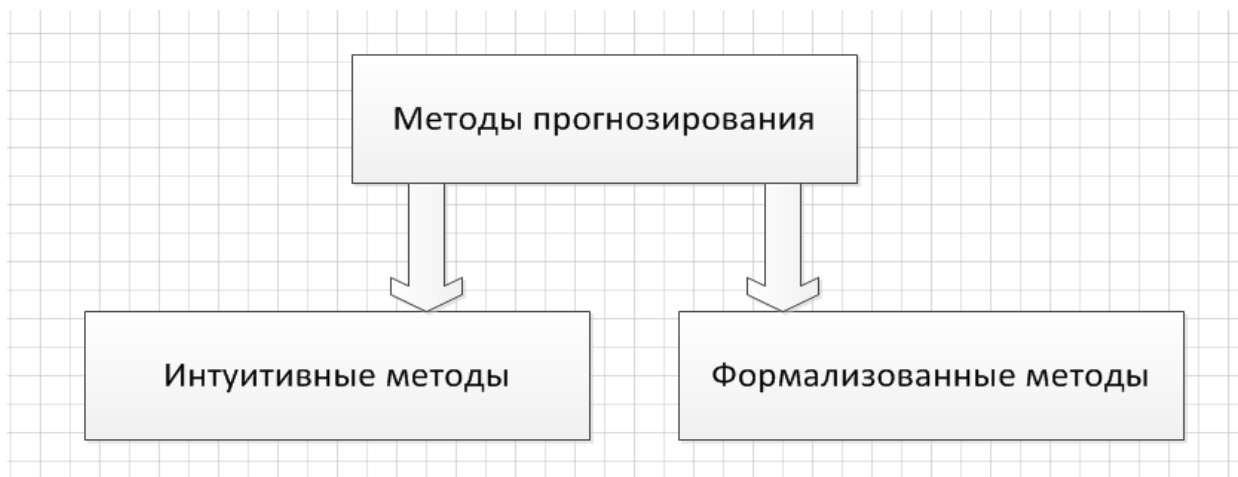


Рисунок 1.1 - Классификация методов прогнозирования

Интуитивные методы – это методы, прогнозирования, использующиеся для процессов, которых нельзя описать математическими способами, формулами [1].

Использование интуитивного метода дает шанс получить предсказываемую оценку положения развития объекта в момент будущего времени независимо от ИО (информационной обеспеченности). Построение процедуры интуитивного и логического мышления индивида в связи с методами оценки, а также обработки полученных значений является сущностью интуитивных методов прогнозирования. В основном, решение данной проблемы базируется на мнении экспертов.

Когда информация об определенном объекте прогнозирования имеет количественный характер, а вредное воздействие разных факторов можно предоставить с помощью формул математики, то используются формализованные методы.

Общая квалификация моделей прогнозирования показана на рисунке 1.2.

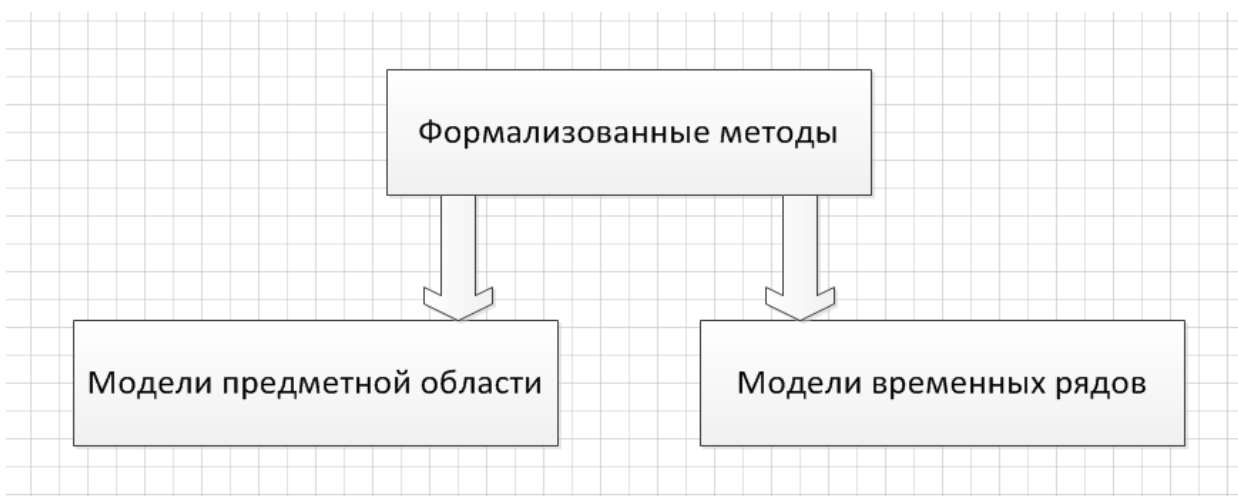


Рисунок 1.2 - Классификация моделей

Визуальным представлением концептуальных объектов или классов реального мира в предметной области является модель предметной области. Одним из начальных стадий проектирования системы, которое необходимо для классификации, выявления и формализации информационных сведений об аспектах предметной области, характеризующих свойства осуществляемой системы, является моделирование предметной области.

Модель предметной области показывает основные понятия контекста в виде объектов предметной области, а также осуществляет связь между этими объектами.

Одним из математической модели прогнозирования является модели временных рядов, стремящихся найти зависимость прошлого значения от будущего во внутренней части самого процесса, а также вычислить прогноз в данной зависимости [2]. Данные модели прогнозирования универсальны для разных предметных областей, т.е. общий вид предметной области не изменяется в зависимости от временного ряда природы. Нейронные сети можно использовать для прогноза температуры воздуха определенной местности, а затем для прогноза биржевых индексов применить аналогичную модель, осуществляющих на нейронных сетях. Это и есть обобщённая модель, т.е. если в кипяток бросить продукт, то он через некоторое время сварится вне зависимости от природы данного продукта.

Классификация моделей временных рядов изображена на рисунке 1.3.

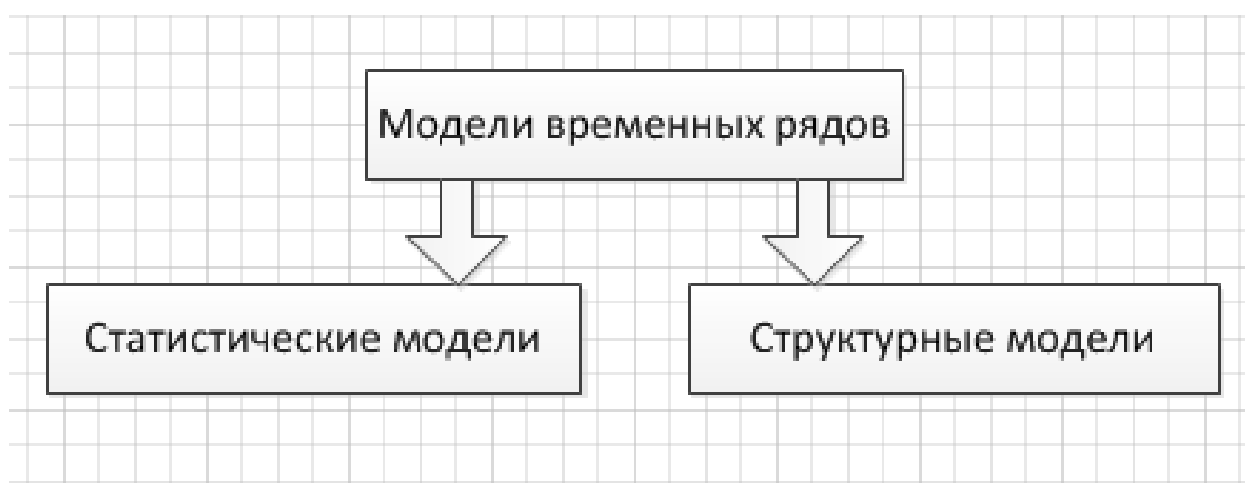


Рисунок 1.3 - Классификация моделей

Модели временных рядов можно разделить на две модели: статистические и структурные.

Отличие статистических моделей от структурных тем, что функциональная зависимость в статистических моделях между фактическими и будущими значениями временного ряда и внешними факторами указана аналитически, а в структурных моделях зависимость задана структурно.

Группы статистических моделей:

- авторегрессионные модели;

- регрессионные модели;
- модели экспоненциального сглаживания.

Группы структурных моделей:

- модели на базе цепей Маркова;
- нейросетевые модели;
- модели на базе классификационно-регрессионных деревьев.

На рисунке 1.4 изображается общая классификация методов прогнозирования.



Рисунок 1.4 - Общая классификация методов прогнозирования

Сравнение некоторых моделей прогнозирования приведено в таблице 1.2.1.

Таблица 1.2.1 - Сравнение моделей прогнозирования

| Модель и метод | Достоинства | Недостатки |
|-------------------------------|---|--|
| Регрессионные модели и методы | простота, гибкость, прозрачность моделирования; единообразие анализа и проектирования | сложность определения функциональной зависимости; трудоемкость нахождения коэффициентов зависимости; |

| | | |
|---|--|--|
| Авторегрессионные модели и методы | простота, прозрачность моделирования; единообразие анализа и проектирования; множество примеров применения | трудоемкость и ресурсоемкость идентификации моделей; невозможность моделирования нелинейностей; низкая адаптивность |
| Модели и методы экспоненциального сглаживания | простота моделирования; единообразие анализа и проектирования | недостаточная гибкость; узкая применимость моделей |
| Нейросетевые модели и методы | нелинейность моделей; масштабируемость, высокая адаптивность; единообразие анализа и проектирования; множество примеров применения | отсутствие прозрачности; сложность выбора архитектуры; жесткие требования к обучающей выборке; сложность выбора алгоритма обучения; ресурсоемкость процесса обучения |
| Модели и методы на базе цепей Маркова | простота моделирования; единообразие анализа и проектирования | невозможность моделирования процессов с длинной памятью; узкая применимость моделей |
| Модели и методы на базе классификационно-регрессионных деревьев | масштабируемость; быстрота и простота процесса обучения; возможность учитывать категориальные переменные | неоднозначность алгоритма построения дерева; сложность вопроса останова |

1.2.2 Исследование букмекерских контор

Букмекерская контора (букмекер) – юридическая фирма или же компания, которая занимается предоставлением возможности заключения денежного выигрыша на исходы результатов определенных спортивных событий.

Ставки принимаются не только на спортивные события, но и на культурные, политические и т.п. По-другому букмекерскую контору можно назвать тотализатором. Раньше механическое оборудование в виде счетчика, показывающее количество ставок, сделанную на каждую лошадь в заезде и общее количество денег называли тотализатором.

Самые распространенные букмекерские конторы в Казахстане:

- БК Олимп;
- Tennisi;
- PROFIT.

БК Олимп. БК Олимп в Казахстане берет начало с 2004 года. В настоящее время данная компания разрослась по территории СНГ и стала доступна для любителей ставок дальнего зарубежья. Олимп имеет широкую сеть представителей в Казахстане.

Данный букмекер предоставляет широкую линию ставок, т.е. игрок может делать ставки на события 17 разных видов спорта. Конечно, большое внимание уделено на футбол. Коэффициенты не являются заниженными. Минимальная ставка начинается с 50 тг. Также БК Олимп предоставляет бонусы или специальные предложения. Букмекерская контора дает повышение коэффициента на 10% в ставках экспресс с коэффициентом 2,5 и больше. На рисунке 1.5 изображен логотип данной компании.



Рисунок 1.5 – БК Олимп

Tennisi. Букмекерская контора Tennisi до 2009 года была в пятерке самых оффлайн-масштабных компаний России. Но несмотря на это, Tennisi, являющаяся членом НАБ (Национальной Ассоциации Букмекеров), имеет авторитет и отлично продолжает качественно работать. Данный букмекер также позволяет делать ставки на: футбол, хоккей, теннис, баскетбол и т.д.

Футбольная линия БК Tennisi разнообразна. Уровень коэффициентов не такой уж высокий, составляет 1.85%. Размеры спортивных ставок зависят от важности турниров, от видов спорта, а также от каждого определенного матча. На рисунке 1.6 изображен логотип данной компании.



Рисунок 1.6 – БК Tennis

Profit. Данная контора была зарегистрирована в Алматы в 2011 году, именно с этого года букмекерская контора начала прием ставок.

Profit Betting была базирована на местности Казахстан, в которой собственные активы охватывают умиряющую большую часть гэмблинговых текстур из разных ареалов СНГ. Лицензия на воплощение деловитости выдана районными контрольными органами, а данные в крайнее время бумаги набирают все огромную злободневность.

Фирма отличается открытостью, постоянно предоставляя всю нужную информацию для собственных покупателей и партнеров. Такая политическая деятельность постоянно оставляет приятное ощущение и располагает к долговременной совместной работе. Не считая это, благотворительные договоры предугадывают узкую ассоциацию с 15 неповторимыми планами. Из данного надлежит, будто не внушающую доверие текстуру вряд ли встанут избирать в качестве напарника по бизнесу.



Рисунок 1.7 – БК Profit

1.2.3 Анализ аналогичных приложений

В данном разделе проводится анализ аналогичных приложений. Рассматриваем аналоги в России, т.к. в Казахстане нет аналогов нашего программного продукта.

1 Мобильное приложение «Робобет» (рисунок 1.8 и 1.9)

Плюсы:

- Быстрая скорость работы приложения;
- Простой интерфейс.

Минусы:

- Нет экспертной системы;
- Нет полной информации ставки на определенное событие;
- Отсутствует возможность сохранять данные в базе (История ставок).



Рисунок 1.8- Мобильное приложение «Роботет»

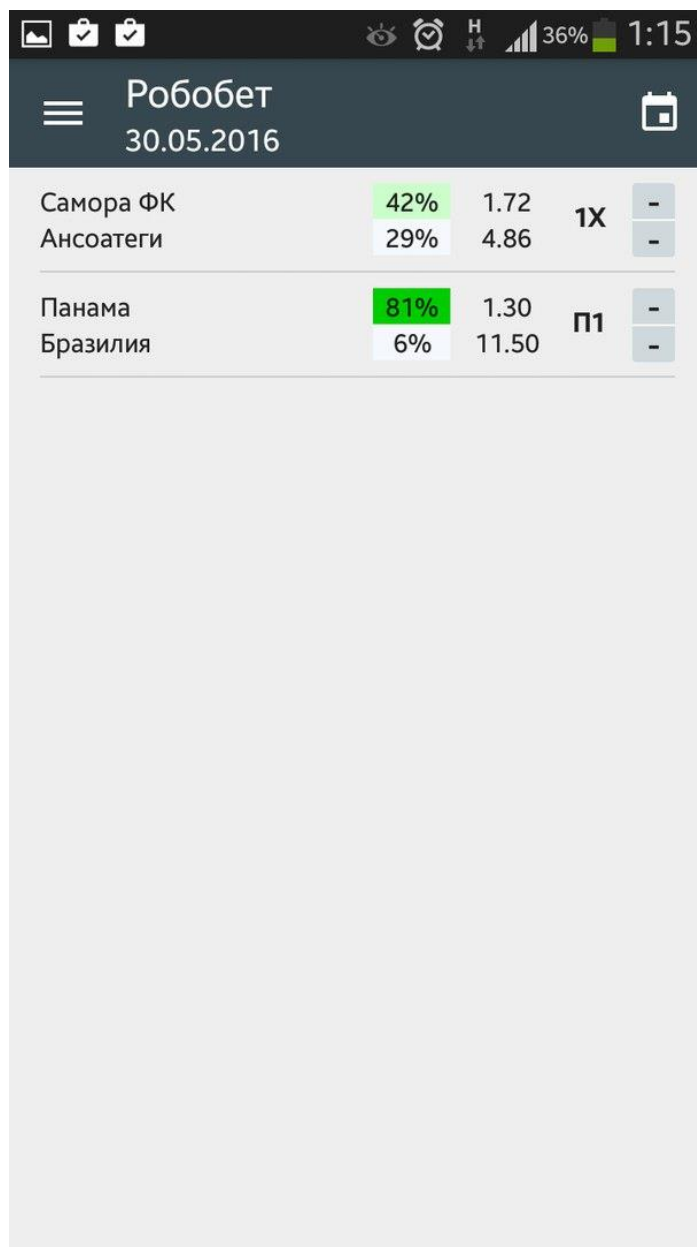


Рисунок 1.9- Мобильное приложение «Робобет»

2 Мобильное приложение «Спорт Прогноз» (рисунок 1.10 и 1.11)

Плюсы:

- Высокая скорость работы приложения;
- Полезные статьи «Как правильно делать ставку», «Тест на стратегию выигрыша».

Минусы:

- Нет экспертной системы;
- Нет полной информации ставки на определенное событие;
- Отсутствует возможность сохранять данные в базе (История ставок);
- Неполная версия программы.

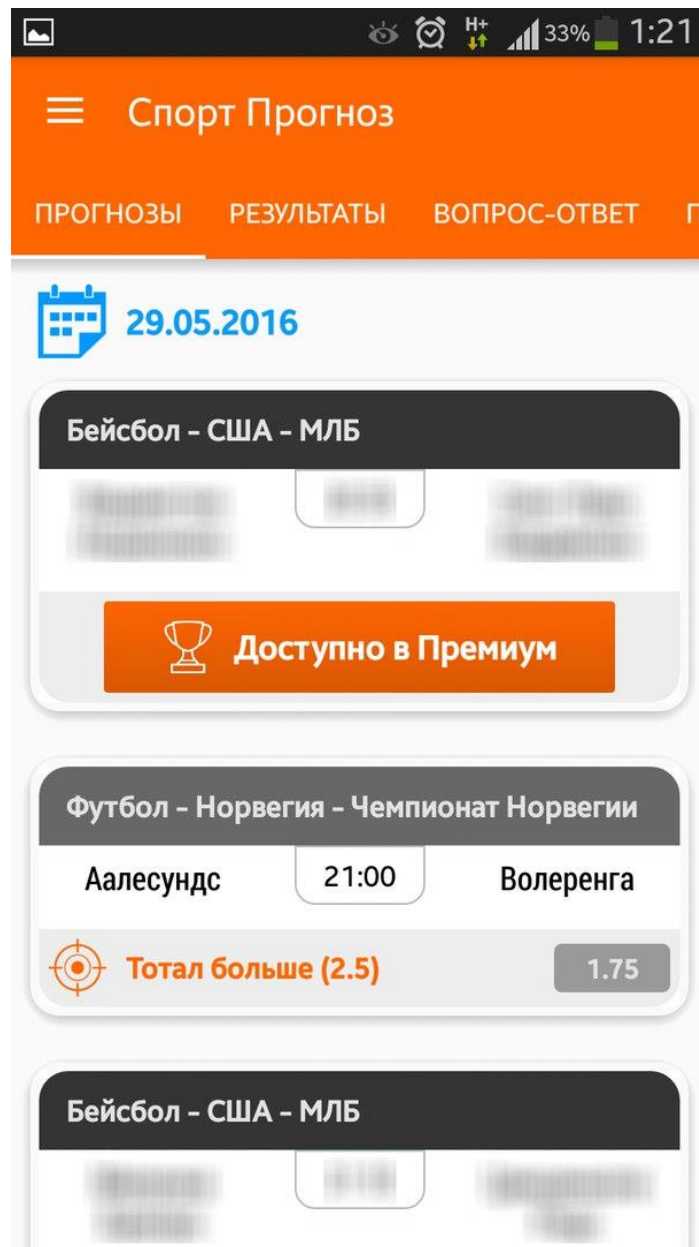


Рисунок 1.10 - Мобильное приложение «Спорт Прогноз»

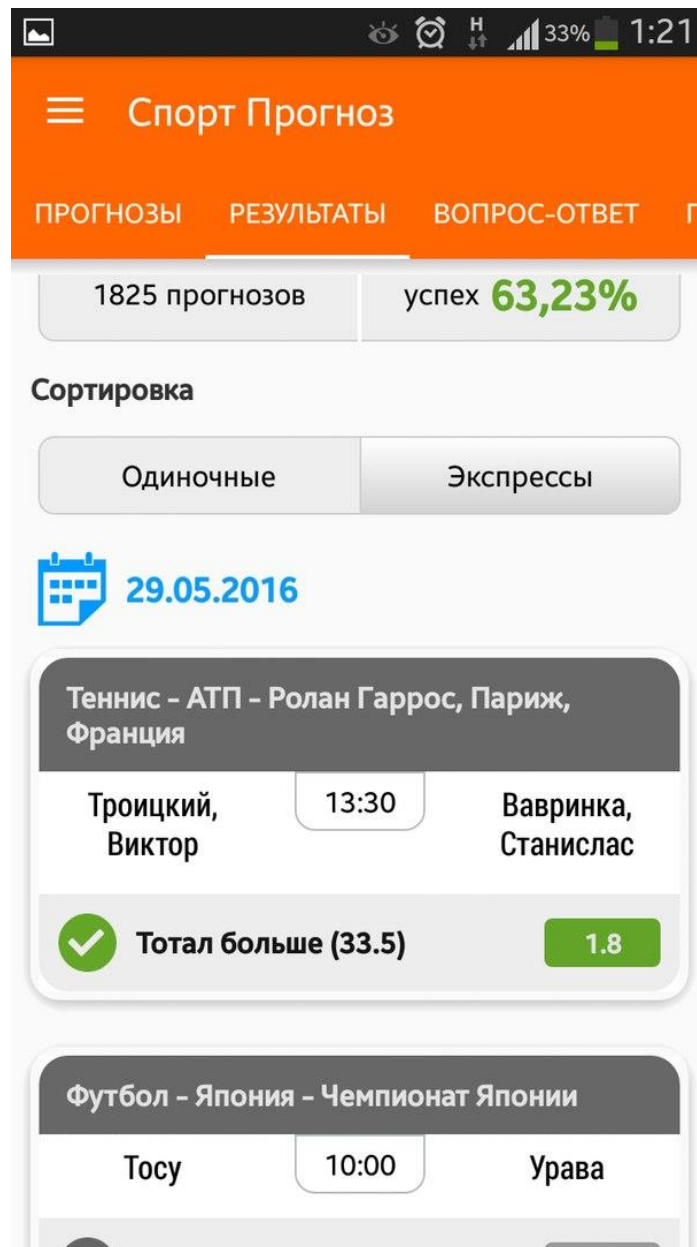


Рисунок 1.11 - Мобильное приложение «Спорт Прогноз»

3 Мобильное приложение «ForeBet» (рисунок 1.12)

Плюсы:

- Понятный интерфейс;
- Функциональность;
- Информации о матче.

Минусы:

- Нет экспертной системы;
- Отсутствует возможность сохранять данные в базе (История ставок);
- Низкая скорость работы.

Forebet
Завтра | Прогнозы 1X2

| Хозяева - Гости | Вероятность в % | | | Пр. | Коэфф. | Рез. |
|---|-----------------|----|----|-----|--------|------|
| | 1 | X | 2 | | | |
| Предстоящий Закончилось | | | | | | |
| Persiba Balikpapan Bali United Pusam Id1 5/30/2016 18:00 | 30 | 50 | 20 | X | 0.00 | |
| Отвидаберг Вернаму Se2 5/30/2016 23:00 | 37 | 36 | 27 | 1 | 2.15 | |
| Örgryte IS Хальмстад Se2 5/30/2016 23:30 | 26 | 24 | 50 | 2 | 3.30 | |
| FH Hafnarfjordur Vikingur Olafsvik Is1 6/1/2016 1:15 | 44 | 38 | 18 | 1 | 1.36 | |
| Fylkir FC Fjölirnir Is1 6/1/2016 1:15 | 23 | 21 | 56 | 2 | 0.00 | |
| Stjarnan FC Breiðablik Is1 6/1/2016 2:00 | 44 | 29 | 27 | 1 | 2.10 | |
| Предстоящий Закончилось | | | | | | |

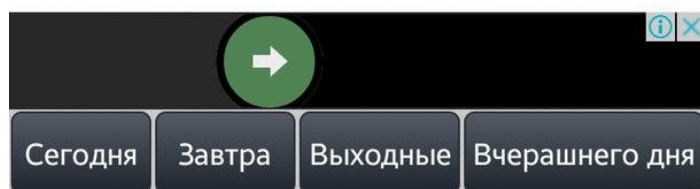


Рисунок 1.12 - Мобильное приложение «ForeBet»

Рассмотрев аналоги приложения, видим, что наше приложение полностью покрывает их минусы.

1.2.4 Анализ факторов прогнозирования

В данном пункте рассматривается правильное составление прогноза для футбольного события. Эти принципы также можно применить для прогноза других видов спорта.

Для начала вам необходимо – точка отсчета. Точка отсчета – это один из быстрых способов, которая позволяет определить предварительный счет матча.

Логическим шагом для начала составления своего мнения о результате футбольной игры является стадия составления предварительного прогнозирования; точка отсчета – «начальный прогноз». Пользуясь данным «первый счет» как точку отсчета, у вас появится возможность сделать свое суждение, прибавляя иные факторы, оказывающие большое влияние на игру.

Ваш составленный подготовительный прогноз о некоем событии станет начальной точкой, с которого будет выводиться прогноз, пользуясь абстрактными и субъективными факторами, которое послужит частью вашего окончательного прогноза.

Данный подготовительный прогноз лучше осуществлять при помощи статистики прошлых игр. Статистика является фундаментом и основой любой нормальной гандикапперской системы, т.к. эти системы предлагают ключи к навыкам [3]. Но навык, который позволяет работать со статистикой, не является, так сказать, самым определяющим. Победитель в сфере прогнозирования в футболе для составления прогноза определенно пользуется следующие факторы:

- Абстрактные;
- Объективные;
- Субъективные.

Абстрактные факторы – это факторы, которые не относятся к уровням талантливых команд. К ним относятся: неожиданные травмы игроков, ослабление команды в результате травм, непредсказуемые повороты в матчах из-за шума болельщиков, погоды и т.п. Главным абстрактных факторов является удача и неудача.

Субъективные факторы - это факторы, включающий в себя мотивацию команды, который играет также важную роль. Субъективные факторы включают в себя то, что способствует влиять на желание игроков команды выиграть. Также включают: переговоры по контрактам, внутрикомандные интриги, травмы, ревность, увольнение тренеров, семейные проблемы.

Объективные факторы – это факторы, которые очевидны каждому и понятны: способность команды, способность, талант каждого из игроков [4].

Первый фактор является самым легким, если пройтись по всем остальным факторам. Т.к. трудно обращаться с данным фактором – практически всегда ее можно игнорировать.

Когда вы сталкиваетесь с профессиональными футбольными командами, то точно можете считать, что более существенными являются субъективные факторы, часто в связи с линиями. Знание влияния факторов мотивации будет определять ваш успех в ставках и способность в правильной оценке шансов определенных команд.

Для составления прогноза, исходя из субъективных факторов, несомненно, вам потребуется некая мера или же точка отсчета. В следствии этого, главной целью математических формул, которые основываются на статистике считается вывод данной точки отсчета, которые основан на

прошлом перформансе. Естественно статистика позволит вам много рассказать, и конечно необходимо знать предыдущую силу футбольных команды.

Не стоит слишком часто полагаться на математические формулы. Чтобы составить точный прогноз, который по возможности будет учитывать абстрактные и субъективные факторы, лучше всего использовать прогноз, который основан на статистике.

Использование математических формул. Во-первых находим футбольное событие, где по вашему мнению букмекер недооценил исход матча. Во-вторых проверяем ее на прибыль. Для данного случая используем формулу

$$K \times P > 1 \quad (1.1)$$

Где K – коэффициент букмекера,

P – вероятность положительного результата события (0 и 1) [5].

На примере рассмотрим матч Алания – Zenit. Ясно, что Zenit сильнее Алании. Из-за этого на Zenit будут ставить ставку много людей. Ясное дело, что коэффициент на победу Zenita будет занижен. Допустим на Аланию выставляют коэффициент 4,6. Судя по статистике этих команд, Алания побеждает в одном из четырех матчей. Т.е. вероятность победы Алании будет равняться $\frac{1}{4} = 0,25 \%$.

Данные вставляем в вышеприведенную формулу:

$$4,6 \times 0,25 = 1,15 > 1$$

Результат вычисления больше единицы, т.е. ставка на Аланию будет выгодна для вас.

Результат получается больше единицы. То есть ставка на команду Алания для вас будет выгодна.

Заметьте, коэффициент на победу не обязательно быть высоким. Самое главное, что результат расчетов по определенной формуле должен $= > 1$.

Погрешность в вероятности победы команды становится больше при уменьшении коэффициента на его победу. Например, если коэффициент равен 5, то погрешность будет равняться 10 %.

Установка рейтинга силы команд. Для установки рейтинга сил команд вы можете воспользоваться результатами последних четырех игр этих команд, участвующих, в каком либо турнире. Рейтинг сил команд позволяет определить вашу оценку на определенную команду – рейтинг защиты и силы нападения этой команды. Установка рейтинг хороша тем, что вы можете использовать ее для определения итогового счета между командами. В начале сезона вы можете использовать статистику матчей определенной команды предыдущего года, которая также может повлиять на ставку.

Чтобы примерно определить рейтинг линий нападения и защиты команды необходимо игнорировать самую низкую и самую высокую результативность по пропущенным и забитым мячам в прошлых четырех играх, а нужно воспользоваться оставшимся двумя счетами. Например, если данная команда забила 5, 10, 15, 20 голов в последних четырех играх, то необходимо убрать 5 и 20, а сложить 10 и 15 и полученный результат разделить на два.

$$(10 + 15) \div 2 \cong 12$$

Окончательный результат дает нам среднюю статистику линии нападения. А для расчета рейтинга защиты почти также, но наоборот. Например, команда за последние четыре матча пропустила 7, 7, 9 и 13 голов, то 7 и 9 – берем среднее значение 8, $8 \div 2 = 4$.

Самый маленький и самый большой показатель пропущенных и забитых голов не рассматривается с целью исключения «шума», который обычно создается при чередовании «провальных» и «в ударе» игр. Данные матчи показывают счет, а не реальную талант и силу команд. В футболе во многих случаях исходы встреч получаются такими, что не показывают таланты футбольных команд.

Одним из образцов данного является то, что при отставании одной команды от другой в счете оппонент находится в ситуации превосходства эмоции – и просто с силой волей заставляет себя забивать снова проигрывающей команде, которая фактически сдалась. В любом случае, при прогнозе футбольного события стоит полагаться на общую статистику двух команд.

2 Проектирование базы данных

2.1 Предварительное проектирование

По вышеприведенным характеристикам системы данной работы, и выделяя главные сущности и связи, друг с другом, показал их структуру на предварительной ER-диаграмме. ER-диаграмма приведена на рисунке 2.1.

Программное обеспечение MS Visio от компании Microsoft было использовано для построения предварительной ER-диаграммы.

Visio — это многостороннее чертежное средство, которое может пригодиться в построении диаграммы для формализации и передачи коммуникационных данных о процессах, инфраструктуре и приложениях.

Visio дает в пользование мощнейшие необходимые средства для создания графических диаграмм и осуществления работы с данными без художественных, либо технологических знаний и умений. При создании организационной, либо сетевой диаграммы или диаграммы процессов, вам предоставляется возможность получить нужную схему с помощью готовых фигур.

Данное программное обеспечение дает возможность не только построить диаграмму сущность-связь, но и в предстоящем будущем экспортировать данную диаграмму в любую из большого количества поддерживаемых ей СУБД, в моем случае это станет экспортирование схемы в SQLite .

Бизнес-правила это короткое, но информативное и четкое повествовательное описание политики, процедур и норм в условиях предприятия. Если бизнес-правила в данной работе заданы правильно, то в них определяются сущности, атрибуты и связи, мощности и ограничения. Некоторые бизнес-правила выполняются при поддержке процедур и триггеров.

Далее поставим связи между сущностями, делая упор на описание операции на ER-диаграмме. То есть связи наших сущностей можно квалифицировать на базе бизнес-правил, которые, в лично свою очередь, построены на основе баз досконального описания операций [6].

Т.к. в данном дипломном проекте было создано мобильное приложение «Прогноз футбольных ставок», которое даст пользователям необходимую информацию для ставки на спортивное событие (т.е. приложение парсит нужные данные из сайтов), были рассмотрены необходимые основные таблицы для базы данных.

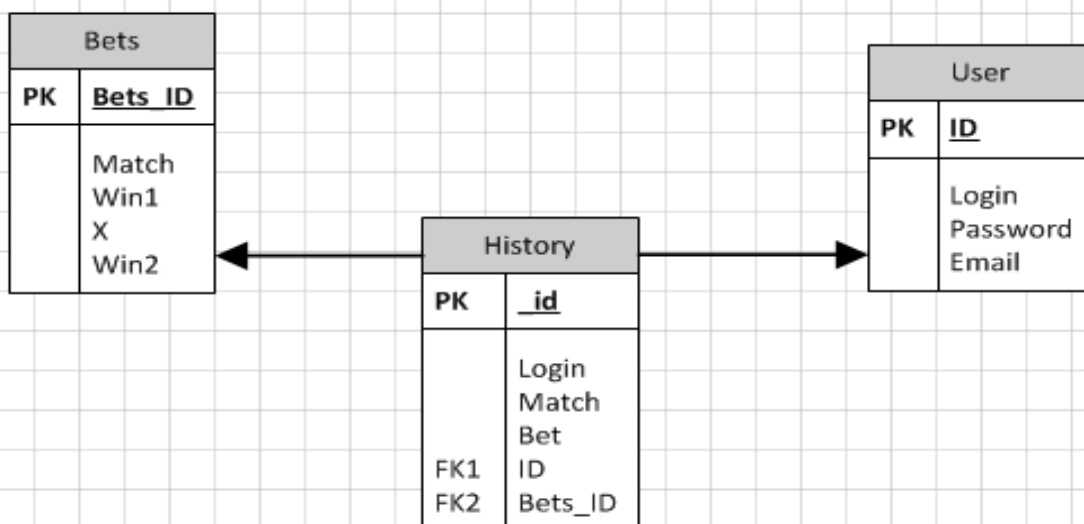


Рисунок 2.1 – ER-диаграмма основных таблиц

2.2 Определение атрибутов всех сущностей

Атрибуты каждой сущности разработанной базы данных показано в таблице 2.2.1. Данная таблица необходима для того, чтобы продолжить последующее проектирование базы данных. В этой таблице определяются все атрибуты сущностей, а также описано их краткое описание.

Таблица 2.2.1 – Атрибуты и сущности

| Область модели | ER | Содержание | | | | | | | | | | |
|--|----------------------------|------------|----|-----------|--|----------------------------|-----|--|-----|---------|--|--|
| <table border="1"> <thead> <tr> <th colspan="2">History</th> </tr> <tr> <th>PK</th> <th><u>id</u></th> </tr> </thead> <tbody> <tr> <td></td> <td>Login Match Bet</td> </tr> <tr> <td>FK1</td> <td>ID</td> </tr> <tr> <td>FK2</td> <td>Bets_ID</td> </tr> </tbody> </table> | History | | PK | <u>id</u> | | Login Match Bet | FK1 | ID | FK2 | Bets_ID | | Таблица History содержит следующие данные: <u>id</u> – идентификатор истории; Login – логин пользователя; Match - название и дата матча; Bet – ставка на событие; ID – идентификатор пользователя; Bets_ID – идентификатор ставки. |
| History | | | | | | | | | | | | |
| PK | <u>id</u> | | | | | | | | | | | |
| | Login Match Bet | | | | | | | | | | | |
| FK1 | ID | | | | | | | | | | | |
| FK2 | Bets_ID | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="2">User</th> </tr> <tr> <th>PK</th> <th><u>ID</u></th> </tr> </thead> <tbody> <tr> <td></td> <td>Login Password Email</td> </tr> </tbody> </table> | User | | PK | <u>ID</u> | | Login Password Email | | Таблица User содержит следующие данные: ID – идентификатор пользователя; Login – логин пользователя; Password – пароль пользователя; Email – электронная почта пользователя. | | | | |
| User | | | | | | | | | | | | |
| PK | <u>ID</u> | | | | | | | | | | | |
| | Login Password Email | | | | | | | | | | | |

| | | |
|-------------|-----------------------|---|
| Bets | | <p>Таблица Bets содержит следующие данные:</p> <p>Bets_ID – идентификатор ставки;</p> <p>Match – название и дата матча;</p> <p>Win1- победа первой команды;</p> <p>X - ничья</p> <p>Win2 – победа второй команды.</p> |
| PK | <u>Bets_ID</u> | |
| | Match | |
| | Win1 X Win2 | |

2.3 Составление реляционных отношений

Между несколькими таблицами базы данных может существовать отношения подчиненности. Суть отношения подчиненности заключается в том, что для каждой записи главной таблицы могут существовать одна или несколько записей таблицы, которая ранее была подчинена.

Некоторое реляционное отношение подходит единой сущности и внутри них задаются некоторые атрибуты сущности.

Для всякого реляционного отношения ориентированы атрибуты, которые имеют название, тип и длину. Обязательные поля обозначено not null, необязательные поля – null. Тип данных указано как: int –числовой, text – символный. Схемы отношений указаны в таблицах 2.3.1 – 2.3.3.

Таблица 2.3.1 – Схема отношения Ставки (Bets)

| Содержимое | Название поля | Тип | Замечания |
|-----------------------|---------------|------|----------------|
| Идентификатор ставки | Bets_ID | Int | Первичный ключ |
| Название и дата матча | Match | Text | Not null |
| Победа первой команды | Win1 | Text | Not null |
| Ничья | X | Text | Not null |
| Победа второй команды | Win1 | Text | Not null |

Таблица 2.3.2 – Схема отношения Истории (History)

| Содержимое | Название поля | Тип | Замечания |
|----------------------------|---------------|------|----------------|
| Идентификатор истории | _id | Int | Первичный ключ |
| Логин пользователя | Login | Text | Not null |
| Название и дата матча | Match | Text | Not null |
| Ставка на событие | Bet | Text | Not null |
| Идентификатор пользователя | ID | Int | Внешний ключ |
| Идентификатор ставки | Bets_ID | Int | Внешний ключ |

Таблица 2.3.3 – Схема отношения Пользователи (User)

| Содержимое | Название поля | Тип | Замечания |
|--------------------------------|---------------|------|----------------|
| Идентификатор пользователя | ID | Int | Первичный ключ |
| Логин пользователя | Login | Text | Not null |
| Пароль пользователя | Password | Text | Not null |
| Электронная почта пользователя | Email | Text | Not null |

2.4 Алгоритм работы

Структура работы приложения показана на рисунке 2.2. В данной схеме показано как работает внутренняя часть программы, когда пользователь использует мобильное приложение.

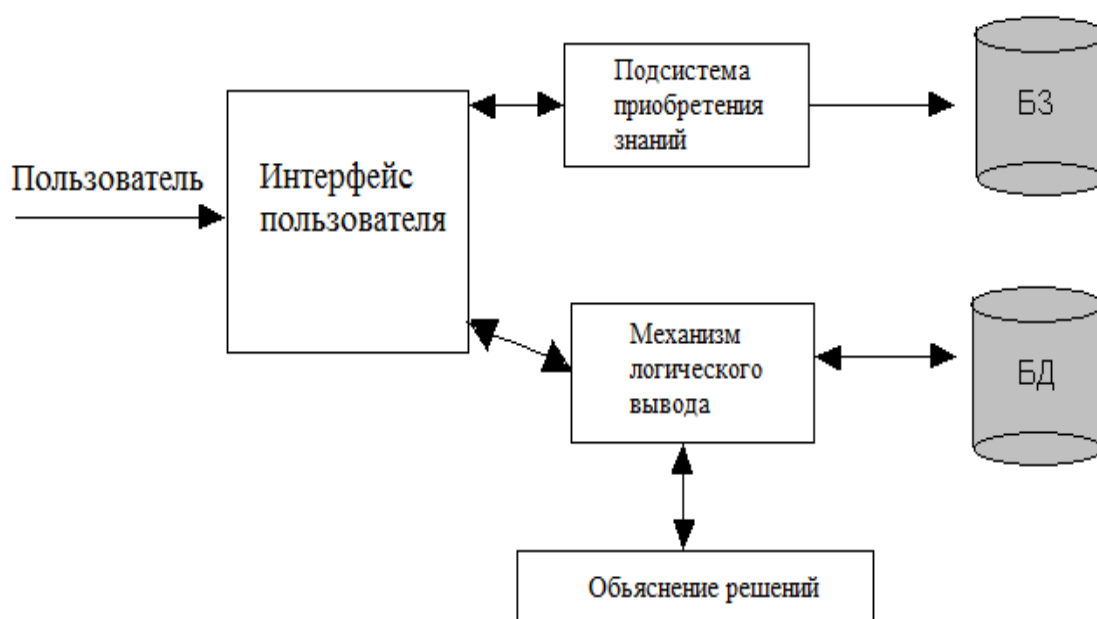


Рисунок 2.2 – Структура работы приложения

База данных предназначена для хранения данных решаемой в текущий момент задачи.

Важным компонентом ЭС является база знаний. База знаний предназначена для хранения долгосрочных информации, описывающих определенную предметную область, а также правил, описывающих целесообразную генерацию данных данной области.

Обычный юзер (либо эксперт) – пользователь, которому необходима консультация экспертной системы, т.е. диалоговый сеанс работы с ней, в

результате которого он решает некоторую экспертную задачу. Диалог пользователя с ЭС осуществляется через интерфейс [7].

Интерфейс пользователя – это конструктивный порядок программных и аппаратных средств, обеспечивающих для конечного юзера употребление устройства для выполнения задач. Данные задачи возникают в среде его профессиональной деятельности.

Для добавления в базу знаний свежих правил и модификации предназначена подсистема приобретения знаний. Ее задача: привести правила к виду, чтобы позволить подсистеме вывода применять это правило при выполнении работы.

Подсистема логического вывода предназначена в основном для работы с БД, используя информацию из БЗ.

Объяснение решений ЭС объясняет, как система получила или не получила решение задачи и какие знания она при этом использовала, что облегчает эксперту тестирование и повышает доверие пользователя к полученному результату.

На рисунке 2.3 показано работа внутренней части программы при работе с базой данных. Например, при входе юзера к системе под своим логином и паролем.



Рисунок 2.3 – Структура работы приложения с БД

2.5 Расчет места для хранения данных

В данном подразделе произведен расчет места для хранения данных, какой объем памяти будет занимать создаваемая база данных.

Формирование объема наружной памяти, который нужен для функционирования системы, состоит из двух составляющих: память, которая занимается модулями СУБД и память, отводимая под данные (МД). Размер самих данных по сравнению размером памяти, занимаемый программными модулями пользователя традиционно слишком велик, из-за этого размер памяти имеет возможность не учитываться.

В данной работе производится примерные расчеты наибольшего объема памяти занимаемой БД. Расчет физической памяти данных каждой таблицы БД указано в нижеприведенных таблицах 2.5.1-2.5.3.

Таблица 2.5.1 – Подсчет физической памяти для Bets

| Содержимое | Название поля | Тип | Длина (байт) |
|-----------------------|---------------|------|--------------|
| Идентификатор ставки | Bets_ID | Int | 4 |
| Название и дата матча | Match | Text | 8 |
| Победа первой команды | Win1 | Text | 8 |
| Ничья | X | Text | 8 |
| Победа второй команды | Win1 | Text | 8 |

Общий объем поля: 36 байт.

Количество: ~100.

Требуемая память: ~ 3600 байт.

Таблица 2.5.2 – Подсчет физической памяти для History

| Содержимое | Название поля | Тип | Длина (байт) |
|----------------------------|---------------|------|--------------|
| Идентификатор истории | _id | Int | 4 |
| Логин пользователя | Login | Text | 8 |
| Название и дата матча | Match | Text | 8 |
| Ставка на событие | Bet | Text | 8 |
| Идентификатор пользователя | ID | Int | 4 |
| Идентификатор ставки | Bets_ID | Int | 4 |

Общий объем поля: 36 байт.

Количество: ~20.

Требуемая память: ~ 720 байт.

Таблица 2.5.3 – Подсчет физической памяти для User

| Содержимое | Название поля | Тип | Длина (байт) |
|--------------------------------|---------------|------|--------------|
| Идентификатор юзера | ID | Int | 4 |
| Логин пользователя | Login | Text | 8 |
| Пароль пользователя | Password | Text | 8 |
| Электронная почта пользователя | Email | Text | 8 |

Общий объем поля: 28 байт.

Количество: ~1.

Требуемая память: ~ 28 байт.

Итак, из проведенных расчетов данных следует рассчитать ориентировочный наивысший объем базы данных

Таким образом, из рассчитанных данных следует рассчитать приблизительно наивысший объем БД:

$$M_c = 2 \cdot (3600 + 720 + 28) = 8696 \text{ (байт)}$$

Объем памяти будет равен сумме объемов всех таблиц базы данных, учитывая, что данные в таблицах будут увеличиваться.

2.6 UML диаграммы

Графическим представлением набора элементов, который чаще всего изображается в виде связанного графа с ребрами и вершинами. Вершинами графа являются сущности, а ребрами – отношения. В основном диаграммы изображаются для визуализации. Главной задачей диаграмм является визуализация системы, которая разрабатывается с разных точек зрения.

Для того, чтобы нарисовать статические объекты в UML есть двенадцать видов диаграмм [8]. В данной работе используются четыре основных диаграмм.

- диаграмма развертывания;
- диаграмма прецедентов;
- диаграмма последовательности;
- диаграмма деятельности.

Диаграмма развертывания. Во время написания дипломной работы, мы делаем приложение для того, чтобы пользователь мог ее запускать на персональном компьютере, имеющий аппаратную конфигурацию и работающий под управлением определенной операционной системы

Когда мы пишем программу, мы пишем ее для того, чтобы запускать на мобильном устройстве, который имеет некоторую аппаратную конфигурацию и работает под управлением некоторой операционной системы.

Корпоративные приложения нередко просят для своей работы ИТ-инфраструктуру, вызывают веб сервисы, хранят данные в базе, которая расположена на серверах компании, используют общие ресурсы. В этих случаях необходимо иметь графическое изображение инфраструктуры, на которую станет развернуто мобильное приложение.

Данная диаграмма изображает саму топологию определенной системы, соединения – пути отправки данных между аппаратными узлами, а также распределение всех компонентов системы по узлам. Диаграмма развертывания в UML является единственной диаграммой, на которой используются трехмерные обозначения. Узлы системы указаны кубиками. А остальные обозначения – плоские фигуры. Для данной дипломной работы диаграмма развертывания представлена на рисунке 2.4.



Рисунок 2.4 – Диаграмма развертывания

Диаграмма прецедентов. Любые системы (включая программные) разрабатываются с учетом того, что в ходе своей работы эти системы будут управляться людьми, а также могут взаимодействовать с определенными другими системами. Экторами являются сущности, взаимодействующие с системой в ходе своей работы. Причем любой эктор ожидает того, что система поведет себя предсказуемым, определенным образом. Эктором могут являться человек или система, класс или подсистема, представляющие нечто вне

сущности. Эктор графически изображается либо человеком, либо символом класса. Эти формы изображения эктора содержат один и тот же смысл и они могут употребляться в диаграммах.

На рисунке 2.5 изображена диаграмма прецедентов для данной дипломной работы.

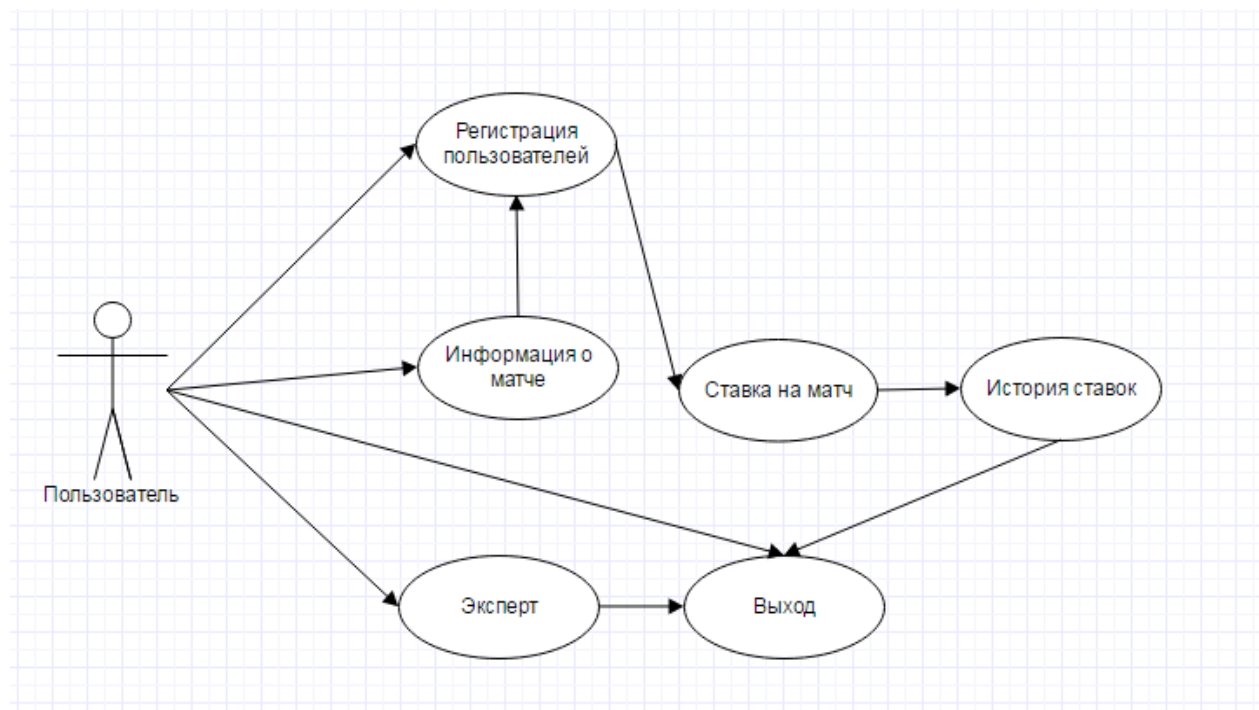


Рисунок 2.5 – Диаграмма прецедентов

Диаграмма последовательностей. Взаимодействие объектов в UML понимается – обмен информацией между объектами. В ходе обмена, информация передается как сообщение. Сообщение, которое несет определенную информацию, также неким образом влияет на получателя.

К диаграммам взаимодействия, которые описывают поведенческие аспекты данной системы, и рассматривает взаимосвязь объектов в реальном времени, относится диаграмма последовательностей. Для нашего случая на рисунке 2.6 изображена диаграмма последовательностей.

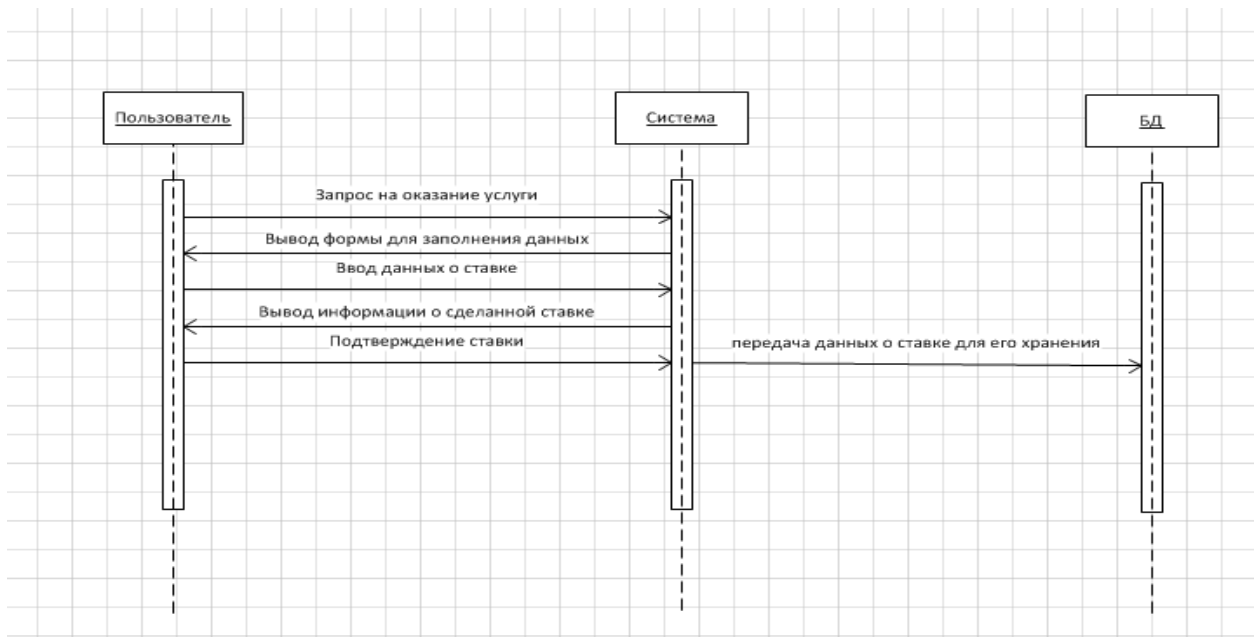


Рисунок 2.6 – Диаграмма последовательностей

Диаграмма деятельности. Одним из основных UML-диаграмм является диаграмма деятельности, показывающая разложение некой деятельности на свои составные части. Также под деятельностью понимается спецификация выполняемого поведения в виде параллельного и последовательного исполнения подчинённых элементов: отдельных действий, которые соединены между собой потоками; вложенных видов деятельности. Данная диаграмма используется при проектировании бизнес процессов, параллельных и последовательных вычислений, а также при моделировании технологических процессов. На рисунке 2.7 изображается диаграмма деятельности мобильного приложения «Прогноз футбольных ставок»

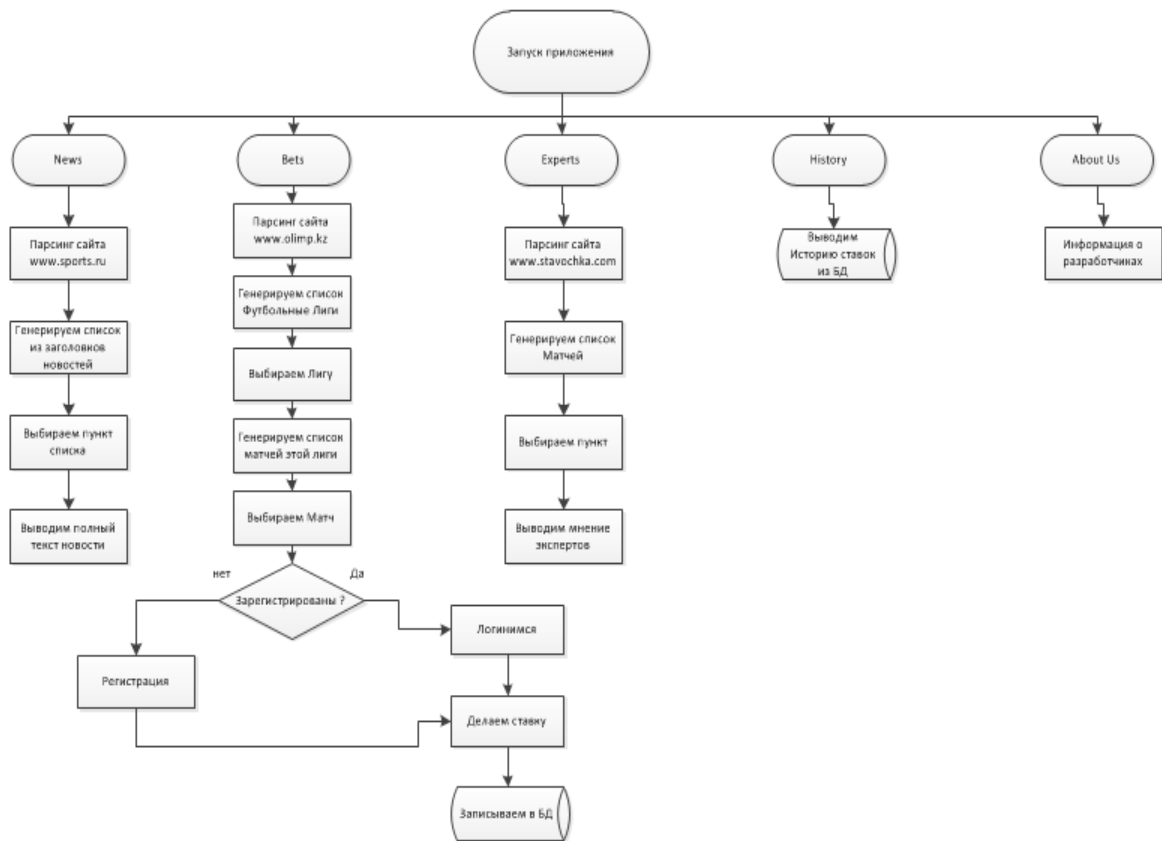


Рисунок 2.7 – Диаграмма деятельности

2.7 Используемые технологии программирования и обоснования выбора СУБД.

СУБД SQLite. В качестве СУБД данного программного продукта была выбрана Android SQLite.

Android SQLite - база данных с открытыми исходниками, включаемая по дефолту в Android, которая имеет стандартные возможности реляционных баз данных:

- синтаксис;
- транзакция;
- prepared statements.

Для работы SQLite требует небольшое количество памяти (~ 250 кб.).

Чтобы пользоваться SQLite в Android нет никакой необходимости установки базы данных или администрирования. Достаточно указать SQL-запрос для работы с базой данных, а нужные операции администрирования происходят автоматически. В SQLite скорость работы с БД может быть медленной из-за операции ввода и вывода. Поэтому все нужные операции стоит выполнять с помощью класса AsyncTask (т.е. в фоне).

SQLite поддерживает следующие типы данных:

- TEXT (схожий с String в java);

- Integer (long);
- Real (double).

Расположение базы данных в мобильном приложении: «DATA/data/APP_NAME/databases/FILENAME». DATA — это путь, возвращаемый методом Environment.getDataDirectory(), а APP_NAME — имя вашего приложения и FILENAME — это имя, которое вы даете базе данных при создании.

Еще одно свойство SQLite это когда база данных доступна только приложению, которая создает ее.

MS Visio. Для создания схем и диаграмм существует приложение MS Visio, которое позволяет помогать визуализировать, исследовать, а также распространять сложные данные. Сложные таблицы и текст можно преобразовать в наглядные для понимания доступные схемы. Для создания самых различных схем в определенных областях, как управление ИТ-средой, строительство и архитектурное проектирование, управление кадрами, разработка пользовательского интерфейса и т.д. данное приложение содержит готовые современные фигуры и шаблоны.

Вместо статических изображений есть возможность создавать связанные с данными схемы MS Visio профессиональной версии 2010 года и MS Visio версии премиум 2010 года, отображающие сведения, которые легко и быстро обновляются, а также заметно повышают производительность работы.

В данном приложении пользователь может использовать различные готовые шаблоны схем и набор элементов, чтобы предоставить, обработать и распространить сведения о системах, ресурсах и процессах в организационном масштабе.

Данные в MS Visio в режиме реального времени можно интегрировать в фигуры из большого количества разных источников, включая SQL, Access, Excel, списки SharePoint и любые источники данных ODBC и OLEDB, путем нескольких щелчков мышью в одном из мастеров работы с данными.

Несмотря на стандартные методы обмена данными по электронной почте и с помощью статических веб страниц, новые функции Visio2010 дают возможность просмотра ярких, наглядных, визуальных элементов, фигур и процессов непосредственно через любой браузер. Веб-схемы Visio профессиональная версия 2010 г. и Visio премиум 2010 г. можно публиковать с помощью служб Visio — функции сервера SharePoint Server 2010, которая визуализирует интерактивные схемы и схемы, связанные с источниками данных. На веб-схемах Visio могут отображаться данные из различных источников, включая Excel, SQL, списки SharePoint и любые источники данных OLEDB или ODBC. На рисунке 2.8 изображено главная страница MS Visio.

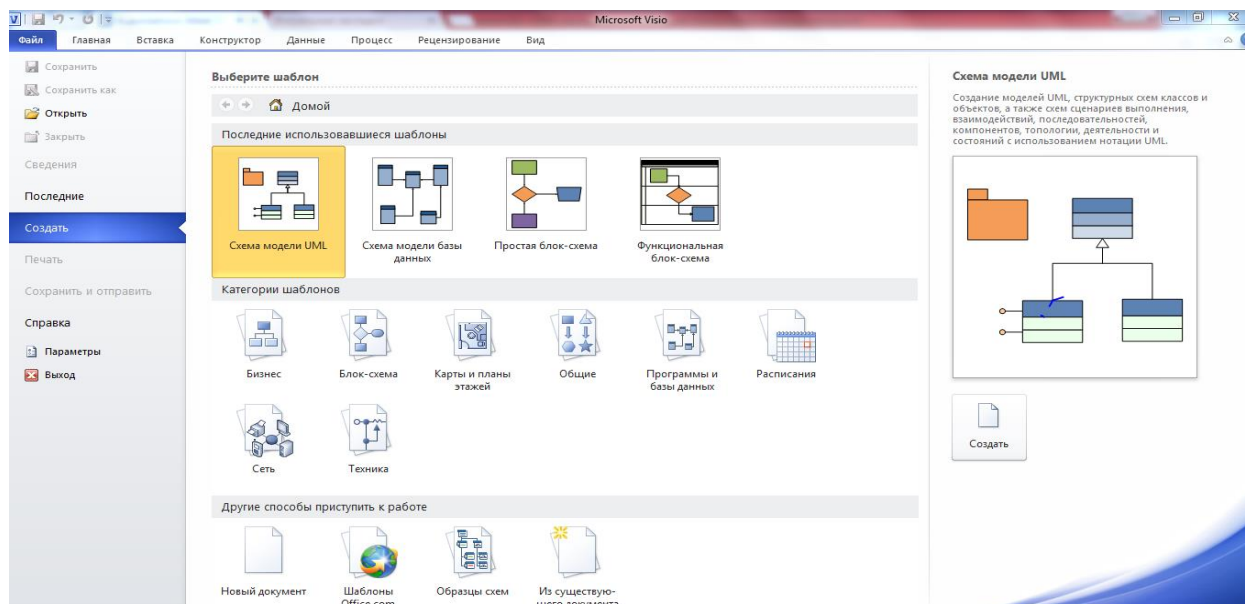


Рисунок 2.8 – MS Visio

Для изображения диаграмм для данной работы было использовано сервис Gliffy. Gliffy – сервис, который позволяет построить не только графики и диаграммы, но и схемы, алгоритмы и планы. Gliffy поддерживает «русский язык». Посередине окна данного сервиса расположено рабочее пространство, а слева – панель элементов. В Gliffy доступны элементы блок-схем, базовые фигуры, UML-диаграммы, схемы сайта, схемы локальных сетей и др. А также есть возможность загружать собственные элементы. Панель инструментов расположена в верхней части, которая позволяет настроить масштаб объектов.

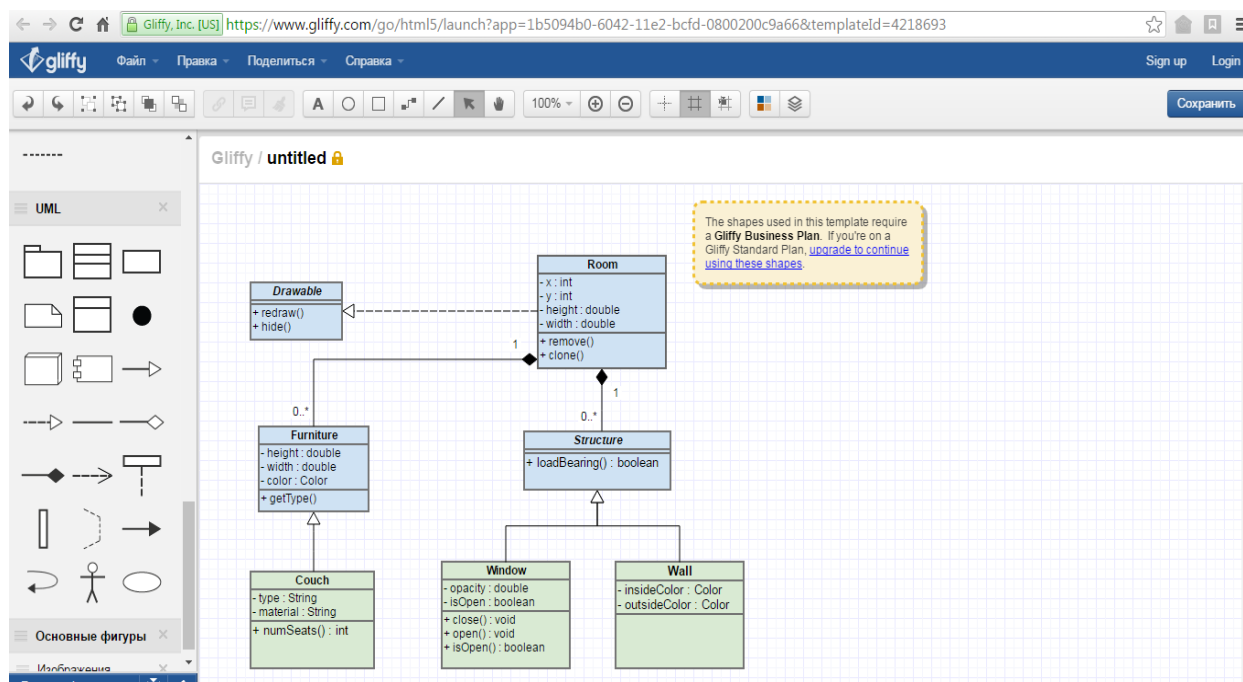


Рисунок 2.9 – Gliffy

Обоснование выбора СУБД. Для разработки базы данных для программного продукта данного проекта была использована СУБД SQLite Android. SQLite дает возможность значительно понизить издержки на поддержку базы данных, а также улучшить качество работы с информацией с помощью мощной интерпретацией языка SQL, который называется TRANSACT SQL [9]. Данная система является надежной, верной, а еще удобной в управлении. Полностью соответствуют для разработки базы данных данной дипломной работы.

СУБД SQLite Android отлично зарекомендовала себя благодаря следующим характеристикам:

- свобода;
- соответствие стандартам;
- легковесность;
- одноуровневость.

Несмотря на скорость и эффективность работы SQLite существуют ряд других преимуществ, с помощью которых данное СУБД является уникальным решением для многих задач, включая мою задачу. Т.к. обычные файлы являются базами данных SQLite, нет необходимости в дополнительных средствах администрирования, которые требуют большое количество времени на проектирование сложной структуры прав доступа, чтобы защитить пользовательские базы данных.

Одним из важных преимуществ SQLite для пользователей считается то, что можно создать такое количество БД, какие они захотят, а также большой плюс заключается в абсолютном контроле над всеми этими БД.

В таблице 2.7.1. приводится сравнение характеристик выбранного мною СУБД с другими. Оценка характеристик СУБД было указано после их анализов. На рисунке 2.10 изображено главная страница Android SQLite.

Таблица 2.7.1 – Сравнение SQLite с другими СУБД (оценка по пятибалльной шкале)

| Возможность СУБД | SQLite | My SQL | Postgresql |
|--|--------|--------|------------|
| Скорость в многопользовательском окружении | 5 | 4 | 3 |
| Надежность | 4 | 4 | 5 |
| Простота построения таблиц | 5 | 5 | 3 |
| Удобство в работе | 5 | 4 | 3 |
| Безопасность | 4 | 3 | 4 |

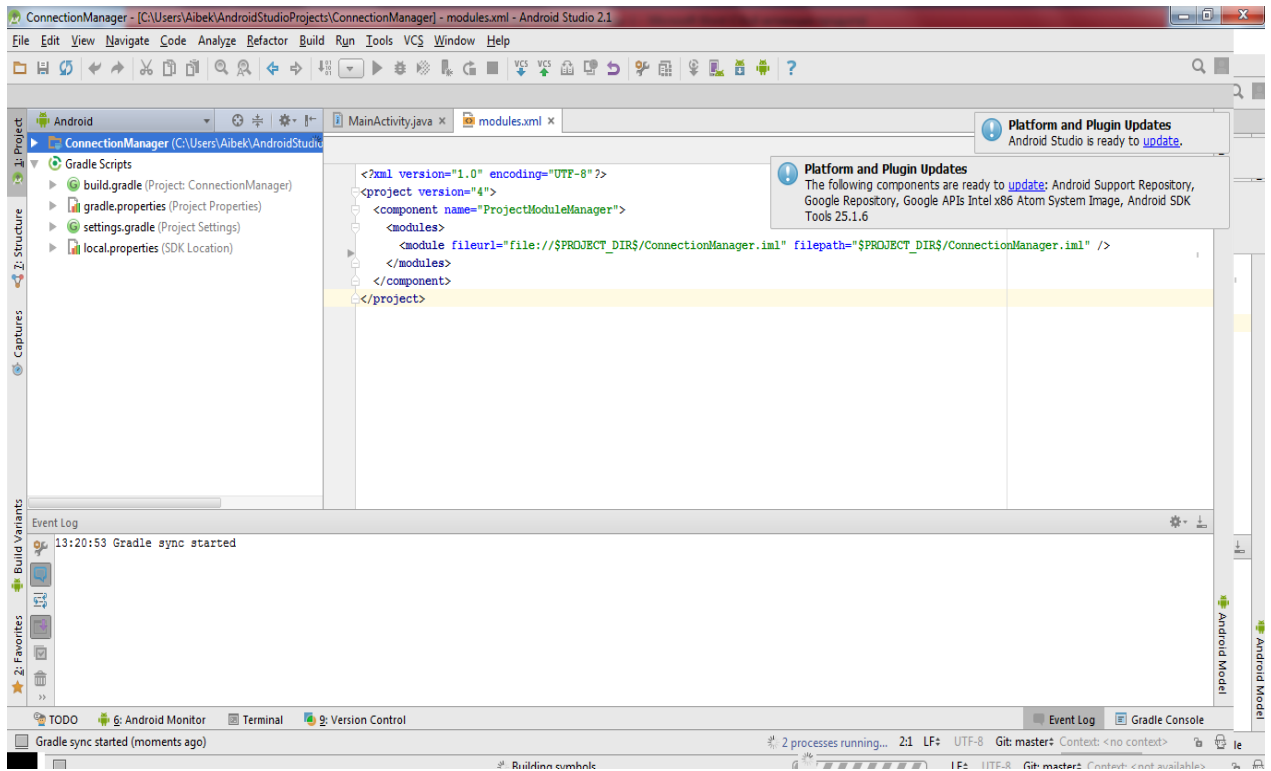


Рисунок 2.10 – Android SQLite.

3 Разработка базы данных

3.1 Подготовка к работе

Разработка базы данных для данного приложения является актуальным, т.к. при ее разработке мобильное приложение получает возможность хранить необходимые данные в базе, а также делать запросы на нее. База данных является важным компонентом мобильного приложения «Прогноз футбольных ставок» в ряду с базой знаний и пользовательского интерфейса. Разработка базы данных осуществлялось в Android Studio.

Установка и настройка Android Studio. Скачиваем и устанавливаем Android Studio, а также необходимые нам компоненты (рисунок 3.1).



Рисунок 3.1 – Установка Android Studio

Для компиляции приложения используем Java 8 Developer Kit (JDK 8), а для использования некоторых инструментов с Android Studio 2.1 устанавливаем Java 8 Runtime Environment (JRE 8). Для тестирования приложения устанавливаем Genymotion (рисунок 3.2).



Рисунок 3.2 – Установка Genymotion

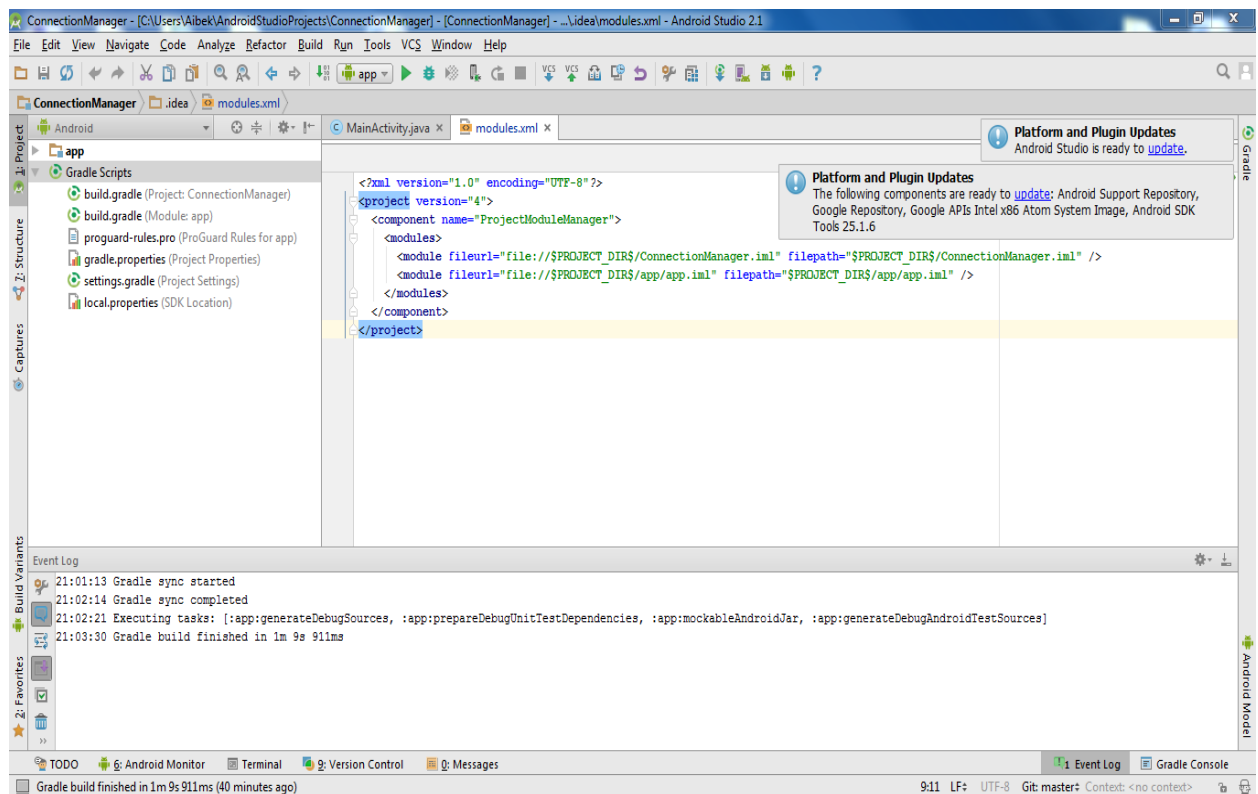


Рисунок 3.3 – Главная страница Android Studio

3.2 Логическое проектирование

Целью логического проектирования является разработка реляционной табличной структуры на языке DDL.

Создание базы данных «Прогноз футбольных ставок» с таблицами на встроенном СУБД Android - SQLite. (рисунки 3.4 -3.6)

```
package com.example.batman.toolbarnavigationversion100500;

import ...

public class DbHelper extends SQLiteOpenHelper {

    private static final String DataBase_Name = "myDataBase.db";
    private static final int DataBase_Version = 1;

    private static final String DataBase_Table_BETS = "Bets";

    public static final String Bets_match = "Match";
    public static final String Bets_P1 = "Win1";
    public static final String Bets_X = "X";
    public static final String Bets_P2 = "Win2";

    private static final String DataBase_CreateTable_Bets= "create table "
        + DataBase_Table_BETS + " ( Bets_ID integer primary key autoincrement, " + Bets_match
        + " text, " + Bets_P1 + " text, " + Bets_X
        + " text, " + Bets_P2 + " text);";

    private static final String DataBase_Table_User = "User";

    public static final String User_Login = "Login";
    public static final String User_Password = "Password";
    public static final String User_Email = "Email";

    private static final String DataBase_CreateTable_User= "create table "
```

Рисунок 3.4 – Создание базы данных 1

```

+ DataBase_Table_User + "( _id integer primary key autoincrement, " + User_Login
+ " text, " + User_Password + " text, " + User_Email
+ " text);";

private static final String DataBase_Table_History = "History";

public static final String History_Login = "Login";
public static final String History_Match = "Match";
public static final String History_Bet = "Bet";

private static final String DataBase_CreateTable_History= "create table "
+ DataBase_Table_History + "( _id integer primary key autoincrement, " + History_Login
+ " text, " + History_Match + " text, " + History_Bet
+ " text);";

public DBHelper(Context context) { super(context, DataBase_Name, null, DataBase_Version); }
public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}
public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler) {
    super(context, name, factory, version, errorHandler);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(DataBase_CreateTable_Bets);
    db.execSQL(DataBase_CreateTable_User);
    db.execSQL(DataBase_CreateTable_History);
}

```

Рисунок 3.5 – Создание базы данных 2

```

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(DataBase_CreateTable_Bets);
    db.execSQL(DataBase_CreateTable_User);
    db.execSQL(DataBase_CreateTable_History);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.v("SQLite", "Обновляемся с версии " + oldVersion + " на версию " + newVersion);
    db.execSQL("DROP TABLE IF IT EXISTS " + DataBase_Table_BETS);
    onCreate(db);
}

```

Рисунок 3.6 – Создание базы данных 3

Далее рассматриваем связь базы данных с приложение платформы Android. Для работы данного приложения с БД использовалась связь SQLite+Java. На рисунке 3.7 изображено запись в базу данных. В данном

случае, после парсинга нужной информации из источников осуществляется запись данных в таблицу Bets.

```
DbHelper mDbHelper = new DbHelper(getActivity(), "myDataBase.db", null, 1);
SQLiteDatabase sdb = mDbHelper.getWritableDatabase();
sdb.delete("Bets", null, null);
ContentValues cv = new ContentValues();
for(String key :hashMap.keySet()){
    cv.put(DbHelper.Bets_match, key);
    ArrayList<String> values = hashMap.get(key);
    for(String val:values){Log.d("#####",val);}
    cv.put(DbHelper.Bets_P1, values.get(0));
    cv.put(DbHelper.Bets_X, values.get(1));
    if(values.size() == 3) {
        cv.put(DbHelper.Bets_P2, values.get(2));
    }
    sdb.insert("Bets", null, cv);
}
sdb.close();
```

Рисунок 3.7 – Запись данных в базу

На рисунке 3.8 осуществляется вывод данных из таблицы History.

```
@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    if(MainActivity.USER_LOGIN == "") {
        DialogFragment frg_signin = new Fragment_Sign();
        frg_signin.show(getFragmentManager(), "frg_signin");
    }
    mDbHelper = new DbHelper(getActivity(), "myDataBase.db", null, 1);
    smpl_cursor_adapter = new SimpleCursorAdapter(getActivity(), R.layout.for_history, null, from, to, 0);
    //lv_history = (ListView) v.findViewById(R.id.lv_history);
    setListAdapter(smpl_cursor_adapter);
    // MainActivity.USER_LOGIN = "Eldar";
    getLoaderManager().initLoader(0, null, this);
}
```

Рисунок 3.8 – Вывод информации из базы данных

3.3 Пользовательский интерфейс базы данных

В результате выполнения работы получена часть нашего мобильного приложения платформы Android, которая отвечает за предоставление записи, вывода, удаления информации из базы данных. На рисунках 3.9-3.15 показано разработанная мною база данных, которая доступна в совместном готовом проекте.

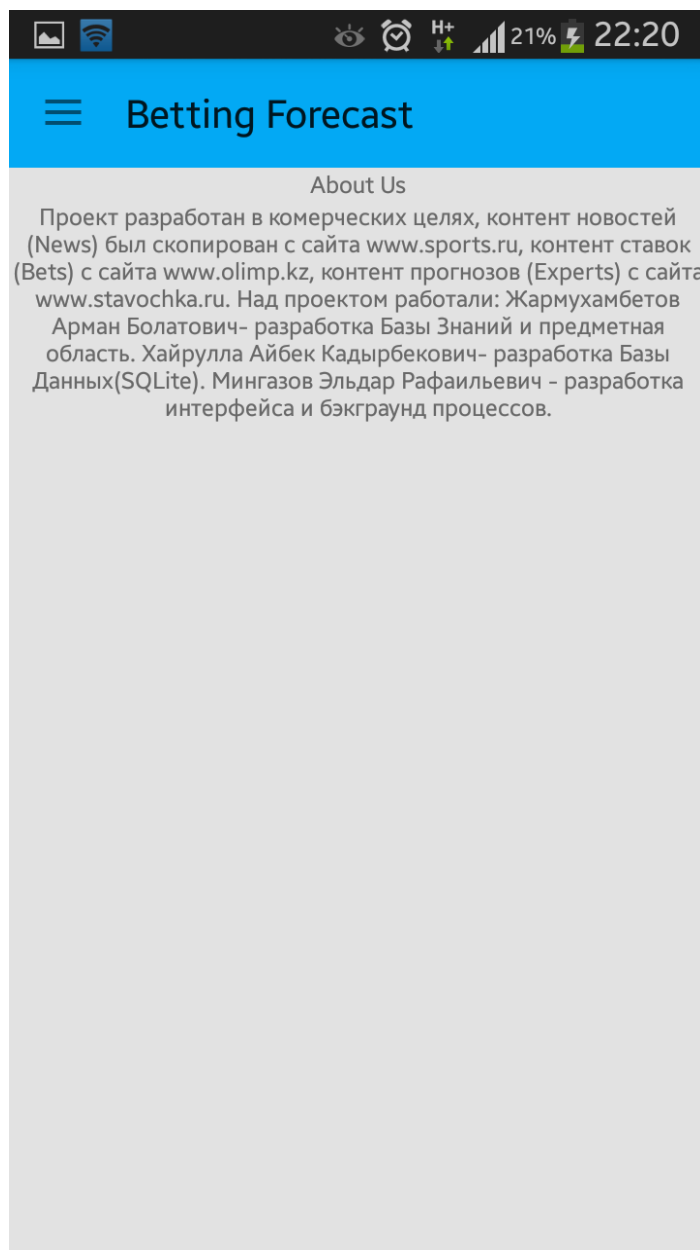


Рисунок 3.9 – Информация о нас

На рисунке 3.9 изображена информация о разработчиках, а также, какие части мобильного приложения они разработали

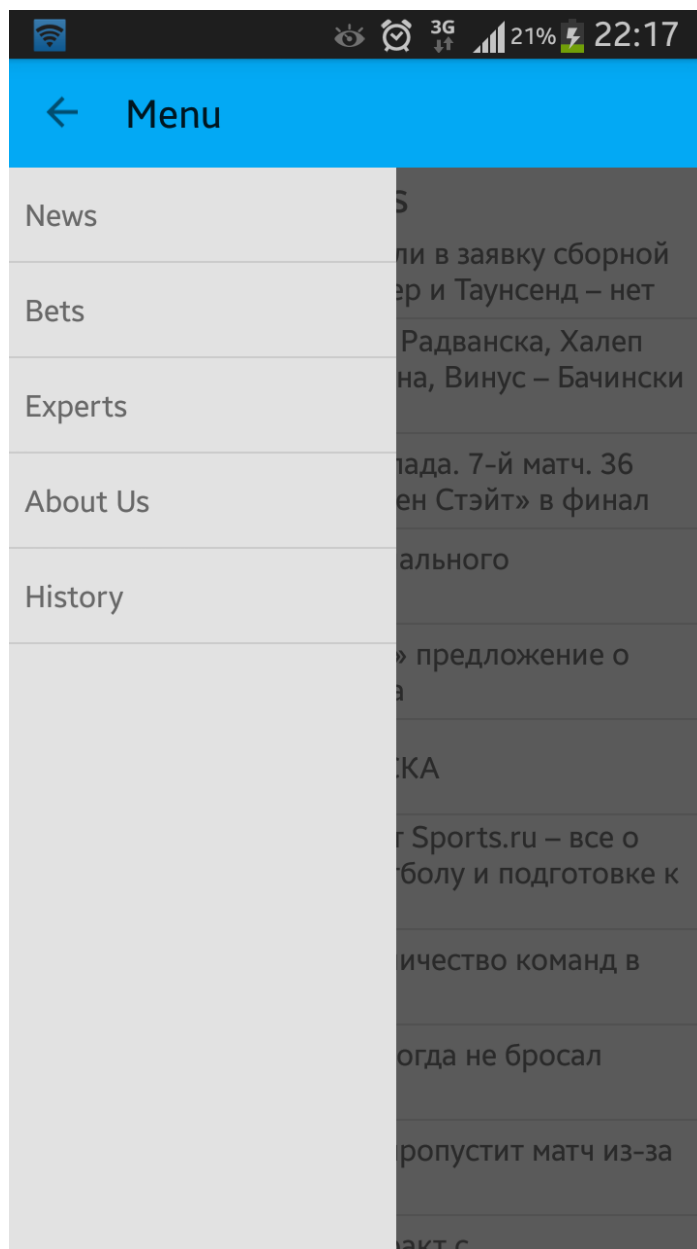


Рисунок 3.10 – Главная страница

Главная страница нашего приложения состоит из пяти разделов. Запись данных в базу осуществляется во вкладке Bets – Выбрать матч. – Войти под логином – Сделать ставку. При пункте «Сделать ставку» сохраняется ставка в History.



Рисунок 3.11 – Вкладка Bets

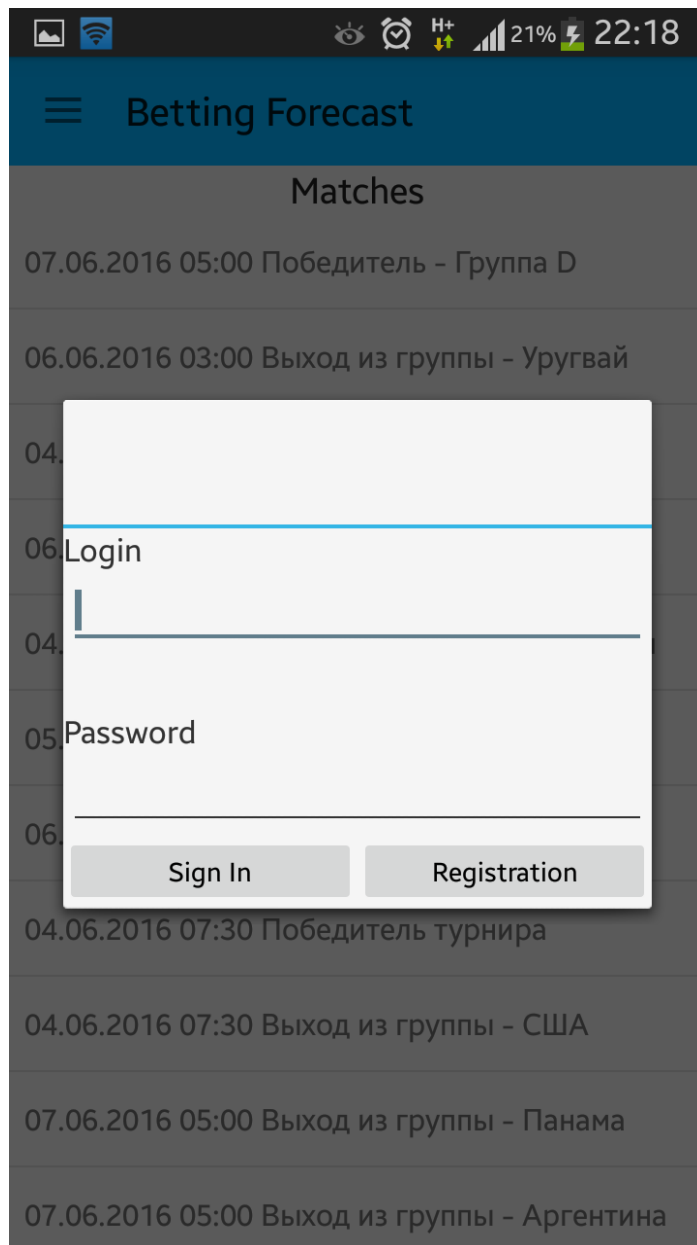


Рисунок 3.12 – Вход в систему

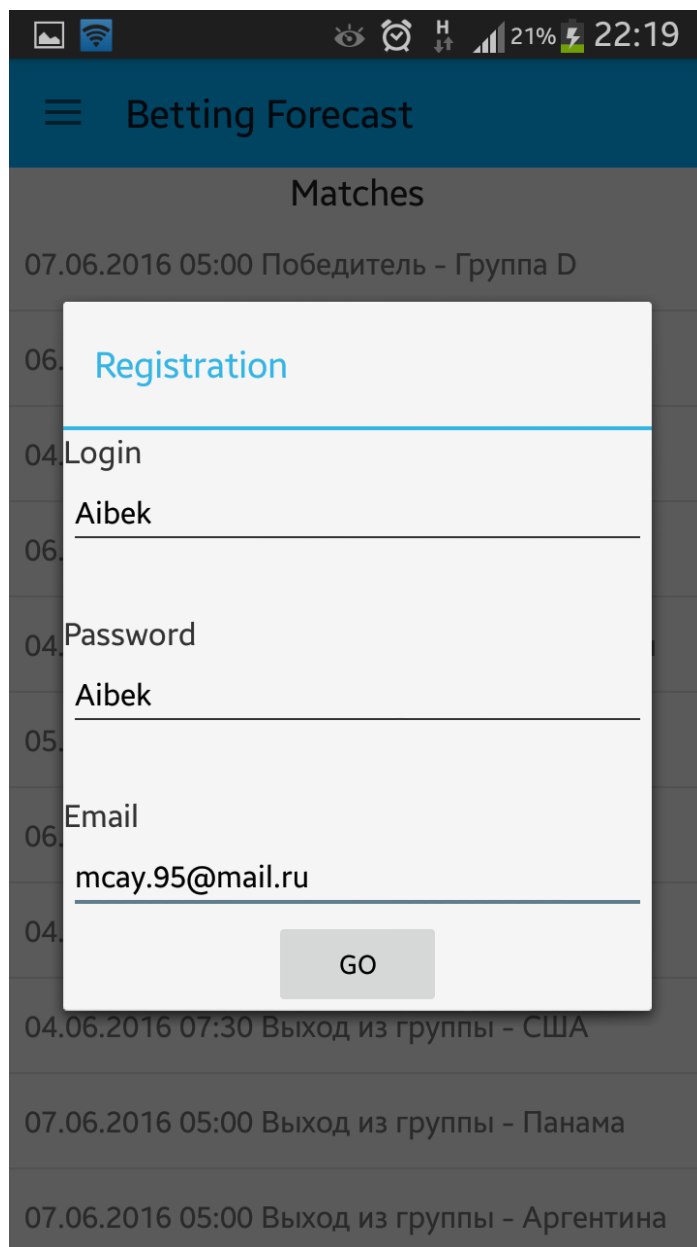


Рисунок 3.13 – Регистрация пользователя

Чтобы пользователь смог сохранить информацию (т.е. записать в базу данных) о своих ставках, необходимо зарегистрироваться.

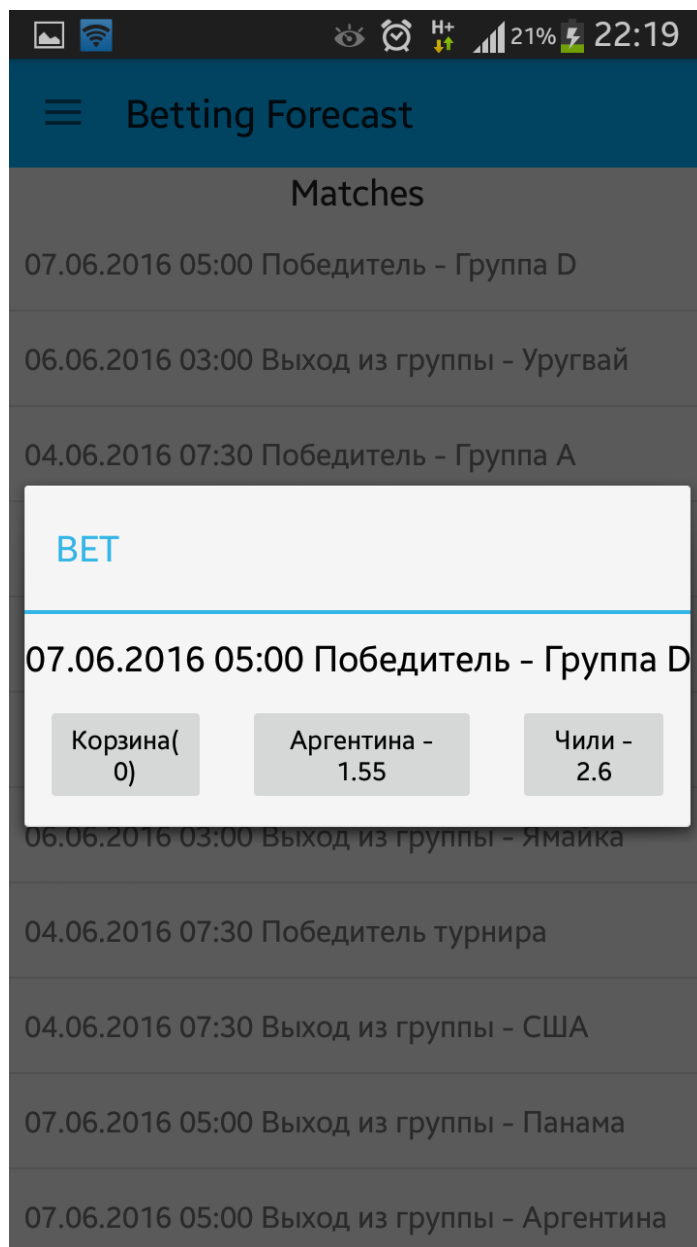


Рисунок 3.14 – Ставка на футбольное событие

На данном рисунке пользователь выбирает одно значение из трех предоставленных, чтобы записать в базу данных.

| Login | Match | Bet |
|-------|---|------------------|
| aibek | 22.05.2016 22:00 Сондаль - Старт | П1 - 1.68 |
| aibek | 23.05.2016 00:00 Лиллестрём - ФК Викинг | X - 3.37 |
| aibek | 24.05.2016 00:30 Аданадемирспор - Элазигспор Первый матч 2:3 | П2 - 3.41 |
| Aibek | 07.06.2016 05:00 Победитель - Группа D | Корзина(0) |
| Aibek | 07.06.2016 05:00 Победитель - Группа D | Аргентина - 1.55 |

Рисунок 3.15 – Вывод данных из базы

4 Экономическое обоснование проекта

4.1 Руководство к работе

В данном дипломном проекте рассматривается разработка БД для мобильного приложения “Прогноз футбольных ставок”.

В современном мире успешно развиваются букмекерские конторы. Данный программный продукт предоставляет информацию и статистику пользователю в сфере спорта.

База данных мобильного приложения позволяет пользователю получать информацию в понятном и удобном виде.

В разделе экономической части приводится рассмотрение составляющих реализации данного проекта, отражающих временные, трудовые и финансовые затраты на проект.

4.2 Расчет вычисления расходов на разработку проекта

Расчет полного вычисления расходов на создание приложения данного проекта C_{ni} выполняется по формуле:

$$C_{ni} = Z_{фот} + Z_{csi} + M_i + P_{ci} + P_{mi} + П_{zi} + P_{ni} \quad (4.1)$$

где $Z_{фот}$ – общий фонд оплаты труда работников (тенге);
 Z_{csi} – отчисления по социальному налогу (тенге);
 M_i – расходы на материалы (тенге);
 P_{ci} – расходы на специальные программные средства, которые необходимы для разработки проектного решения (тенге);
 P_{mi} – расходы, связанные с эксплуатацией техники (тенге);
 $П_{zi}$ – прочие расходы (тенге);
 P_{ni} – накладные расходы (тенге).

Размер фонда оплаты труда работников $Z_{фот}$ считается по формуле:

$$Z_{фот} = Z_{oi} + Z_{di}, \quad (4.2)$$

где Z_{oi} – основной заработок работника (тенге);
 Z_{di} – дополнительный заработок работника (тенге).

Расходы на оплату труда находятся в зависимости от объема и трудоемкости создание программного продукта [10]. Общий объем V_0 данного приложения находится исходя от численности и объема функции, реализуемых в ней.

$$V_0 = \sum_{i=1}^n V_i \quad (4.3)$$

где V_i - объем функции программного обеспечения;
 n – общее количество используемых функций.

Из приложения определяем объем ПО (строки исходного кода приложения, LOC) составляют 6936.

Таким образом

$$V_0 = 6936 \text{ строк кода}$$

Общий трудовой объем проекта рассчитывается по формуле:

$$T_0 = T_n \cdot K_c \cdot K_T \cdot K_n \quad (4.4)$$

где K_c – коэффициент, который учитывает сложность приложения;
 K_T – поправочный коэффициент, который учитывает степень использования при разработке стандартных модулей;
 K_n – коэффициент, который учитывает степень новизны приложения;
 T_n – нормативный трудовой объем.

Коэффициент сложности находится на основе данных, которые предоставлены в таблице 4.1 и значение K_c составляет 0,18, т.к. в данной работе существует более трех характеристик – обеспечение хранения, функционирование системы, интерактивный доступ, ведения и поиска информации в сложных структурах данного проекта.

Таблица 4.1 – Дополнительные коэффициенты сложности ПО

| Характеристические данные программы | Значения K_c |
|-------------------------------------|----------------|
| две характеристики | 0,12 |
| три характеристики | 0,18 |
| свыше трех характеристик | 0,26 |

Поправочный коэффициент, который учитывает степень использования при создании проекта стандартных модулей (K_T), которые определяются на основе данных показанных в таблице 4.2 и $K_T = 0,7$.

Таблица 4.2 – Значения поправочного коэффициента, учитывающего использование стандартных модулей типовых программ и ПО (K_T)

| Степень охвата реализуемых функций разрабатываемого приложения обычными модулями и стандартными программами | Значения |
|---|----------|
| От 40 % до 60 | 0,7 |

Поправочный коэффициент, который определяет новинку данного создаваемого проекта K_n находится на основе данных, которые показаны в таблице 4.3 и составляет $K_n = 0,9$.

Таблица 4.3 – Поправочные коэффициенты (K_n)

| Категория новизны | Степень новизны | Использование | | Значение K_n |
|-------------------|---|--------------------------|------------------|----------------|
| | | На основе нового типа ПК | В среде новой ОС | |
| А | Принципиально новые приложения, которые не имеют аналогов | + | + | 1,75 |
| | | - | + | 1,6 |
| | | + | - | 1,2 |
| | | - | - | 1,0 |
| Б | Приложения, которые являются развитием определенного параметрического ряда ПО | + | + | 1,0 |
| | | - | - | 0,9 |
| | | + | - | 0,8 |
| В | ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС | - | - | 0,7 |

Оптимальные нормы времени на разработку данного дипломного проекта, которая зависит от группы сложности и уточненного объема программного обеспечения являются базой для нахождения нормативной трудоемкости.

На основе объема программного продукта и категории сложности, уточняющая с учетом новизны и сложности данного проекта, и степени пользования стандартных модулей при ее разработке, определяется нормативный трудовой объем проекта T_n [11].

Для 1й категории сложности приложения:

$$T_n = 218$$

Общий трудовой объем осуществляется по формуле (4.4):

$$T_o = 218 \cdot 0,18 \cdot 0,7 \cdot 0,9 = 24,72 \text{ (чел./час.)}$$

Плановое количество работников $Ч_p$ и плановые сроки, которые необходимы для разработки данного проекта в целом T_p определяются на основе трудоемкости [12]. Причем могут выполняться следующие задачи:

– расчет количества работников при указанных сроках реализации проекта;

– расчет срока реализации проекта при заданном количестве работников.

Численность работников проекта $Ч_p$ определяется по формуле:

$$Ч = T_o / (T_p \cdot \Phi_{эф}) \quad (4.5)$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

T_o – общая трудоемкость реализации программного продукта (чел./дн.);

T_p – срок создания проекта (лет).

Срок создания проекта (T_p) вычисляется по формуле:

$$T_p = T_o / (Ч_p \cdot \Phi_{эф}) \quad (4.6)$$

где $Ч_p$ – плановое число работников.

Эффективный фонд времени работы одного работника ($\Phi_{эф}$) вычисляется по формуле:

$$\Phi_{эф} = D_r - D_b - D_o - D_n \quad (4.7)$$

где D_r – общее число дней работника в году;

D_n – общее число праздничных дней работника в году;

D_b – общее число выходных дней работника в году;

D_o – общее число дней отпуска работника.

Т.к., в соответствии с производственным календарем на 2016 год:

$D_o = 14$, $D_n = 15$, $D_r = 366$, $D_b = 112$.

Эффективный фонд времени одного работника составит:

$$\Phi_{эф} = 366 - 112 - 14 - 15 = 225 \text{ (дней)}$$

Плановое общее число работников $Ч_p = 1$, следовательно, по формуле (4.6)

$$T_p = 24,72 / (1 \cdot 225) = 0,11 \text{ лет} = (40 \text{ дней})$$

Тогда плановое количество работников

$$Ч = 24,72 / (0,11 * 225) = 1 \text{ (чел.)}$$

Основной заработок на конкретный программный продукт осуществляется по формуле:

$$З_{oi} = \sum_{i=1}^n T_{чi} \cdot K \cdot T_{ч} \quad (4.8)$$

где n – количество работников, занятых реализацией определенного приложения;

$T_{чi}$ – тарифная ставка за час i-го работника (тыс. тенге);

$\Phi_{п}$ – плановый фонд рабочего времени i-го работника (дней), общее количество рабочих дней в одном месяце;

$T_{ч}$ – общее число часов работы работника в день (час), 8 часов; K – коэффициент премии, которая составляет 1,38.

Так как на создание данного проекта уйдет \approx 1 месяц и 10 дней, а число рабочих дней в каждом месяце разное, то берем среднее значение $\Phi_{п} = 22$ рабочих дней.

Штатное рабочее расписание группы работников, которые участвуют в реализации проекта, с указанием образования, квалификации, специальности и должности составляется исходя из данных о специфике.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов–исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности (таблица 4.4).

Таблица 4.4 - Сведения по работникам, участвующих в данном проекте

| Разработчик | Количество, чел. | Заработок в месяц, тенге |
|----------------|------------------|--------------------------|
| IT-программист | 1 | 70000 |
| Итого | 1 | 70000 |

Часовая тарифная ставка определяется делением месячной тарифной ставки, которая установлена при 40 – часовой рабочего времени недельной нормы, а также общего фонда времени (Φ_p):

$$T_{ч} = \frac{T_{м}}{\Phi_p} \quad (4.9)$$

где $T_{м}$ – месячная тарифная ставка (тыс.тенге).

$T_{ч}$ – часовая тарифная ставка (тыс.тенге);

Формула вычисления общего фонда времени

$$\Phi_p = T_{\text{ч}} \cdot \Phi_{\text{п}} \quad (4.10)$$

Таким образом, получается

$$\Phi_p = 8 \cdot 22 = 176 \text{ (часов)}$$

Тарифная ставка IT-программиста проекта:

$$T_{\text{ч}} = 70000 / 176 = 398 \text{ (тенге в час)}$$

Основная заработная плата программиста составит:

$$З_{oi} = 398 \cdot 1,38 \cdot 235 = 129071 \text{ (тенге)}$$

Итоги расчета основного заработка записаны в таблицу 4.5.

Таблица 4.5 – Итоги расчета расхода основной заработной платы

| Наименование содержания работ | Разработчик | Трудовой объем | Заработная плата за час работы, т/час | Сумма заработной платы, тенге |
|-------------------------------|----------------|----------------|---------------------------------------|-------------------------------|
| Тех. задание | IT-программист | 65 | 398 | 23880 |
| Моделирование | IT-программист | 55 | 398 | 21890 |
| Программирование | IT-программист | 75 | 398 | 29850 |
| Тестирование | IT-программист | 20 | 398 | 7960 |
| Внедрение | IT-программист | 25 | 398 | 9950 |
| Итого | | 235 | 398 | 93530 |

Дополнительная заработная плата составляет 23% от основной заработной платы и вычисляется по формуле:

$$З_{di} = З_{oi} \cdot H_d / 100 \quad (4.11)$$

где H_d – коэффициент дополнительного заработка работника

$$З_{di} = 129071 \cdot 23 / 100 = 29686 \text{ (тенге)}$$

Общий фонд оплаты труда работника равен

$$З_{\text{фот}} = 129071 + 29686 = 158757 \text{ (тенге)}$$

По ст. 358 п. 1 НК РК социальный налог (Z_{csi}) составляет 11 % от дохода работника, и вычисляется по формуле:

$$Z_{csi} = (\text{ФОТ} - \text{ПО}) \cdot 11\% \quad (4.12)$$

где ПО – пенсионные отчисления, составляющий 10% от ФОТ

$$\text{ПО} = \text{ФОТ} \cdot 10\% \quad (4.13)$$

Вычисление пенсионного отчисления:

$$\text{ПО} = 158757 \cdot 0,1 = 15875,7 \text{ (тенге)}$$

$$Z_{csi} = (158757 - 15875,7) \cdot 0,11 = 15717 \text{ (тенге)}$$

Расходы на материалы находятся по формуле

$$M_i = (Z_{och} \cdot H_{mz}) / 100\% \quad (4.14)$$

где H_{mz} – норма расхода материалов от основного заработка (3–5%)

$$M_i = 129071 \cdot 0,03 = 3872 \text{ (тенге)}$$

Расходы по части «Машинное время» P_{mi} включают оплату машинного времени, который необходимо для реализации и отладки программы, определяющийся по нормативам (в машино – часах) на 100 строк исходного кода (H_{mb}) машинного времени в зависимости от характера решаемых задач и типа персонального компьютера:

$$P_{mi} = C_{mi} \cdot (V_{oi} / 100) \cdot H_{mb} \quad (4.15)$$

где C_{mi} – цена одного машинного времени (тыс.тенге);

H_{mb} – норматив расхода машинного времени на отладку 100 строк исходного кода (машинно–часов);

V_{oi} – общий объем ПО (строк исходного кода).

Норматив расхода машинного времени на отладку 100 строк исходника определяется на основе таблицы Д.1 (Приложения Д) и составляет 12 ч/100 строк кода

$$P_{mi} = 398 \cdot (6936 / 100) \cdot 12 = 331263 \text{ (тенге)}$$

Расходы по части «Прочие затраты» P_{zi} на определенное приложение включают себе затраты на приобретение, а также подготовку специальной

научно–технической информации и специальной литературы. Они определяются по нормативу, реализуемому в целом по организации, в процентах к основной заработной плате

$$П_{zi} = З_{oi} \cdot Н_{пз} / 100 \quad (4.17)$$

где $Н_{пз}$ – норматив прочих расходов в общем по организации в (20%)

Таким образом:

$$П_{zi} = 129071 \cdot 0,2 = 25814 \text{ (тенге)}$$

Затраты части «Накладные расходы» P_{ni} вычисляется по нормативу $Н_{рн}$ в процентном отношении к основной заработной плате работника. Норматив устанавливается в целом по организации

$$P_{ni} = З_{oi} \cdot Н_{рн} / 100\% \quad (4.18)$$

где P_{ni} – накладные расходы на определенное ПО (тыс.тенге);
 $Н_{рн}$ – норматив накладных расходов в целом по организации 70%.

$$P_{ni} = 129071 \cdot 0,7 = 90349,7 \text{ (тенге)}$$

Полные затраты на создание базы данных, формула (4.1)

$$C_{ni} = 158757 + 15717 + 4762,7 + 331263 + 31751 + 111130 = 6922,56 \text{ (тенге)}$$

Результаты расчета расходов на создание БД и их структура представлены в таблице 4.6 и на рисунке 4.1.

Таблица 4.6 – Итоги расчета расходов на разработку базы данных

| Затраты на разработку | Условное обозначение | Значение, тенге | В процентах от общей суммы |
|-----------------------|----------------------|-----------------|----------------------------|
| Фонд оплаты труда | $З_{фот}$ | 158757 | 25 |
| Социальный налог | $З_{сзи}$ | 15717 | 3 |
| Материалы | M_i | 3872 | 1 |
| Машинное время | P_{mi} | 331263 | 53 |
| Прочие затраты | $П_{zi}$ | 25814 | 4 |
| Накладные расходы | P_{ni} | 90349,7 | 14 |
| Итого | | 625772,7 | 100 |

Структура затрат

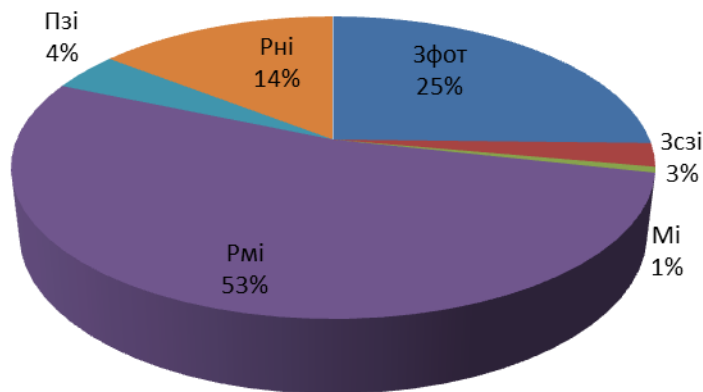


Рисунок 4.1 – Структура расходов на создание базы данных для мобильного приложения

4.3 Расчет цены программного продукта

Расчет цены программного продукта, который был реализован одной организацией по заказу другой организацией, а также не предназначенный для тиража вычисляется по формуле:

$$C_{\text{пп}} = Z_{\text{рпр}} + П_{\text{п}} + \text{НДС} \quad (4.19)$$

где $C_{\text{пп}}$ – цена программного продукта, тенге;
 $П_{\text{п}}$ – планируемая прибыль, тенге;
 $Z_{\text{рпр}}$ – затраты на реализацию проекта, в данном случае программного продукта, тенге;
НДС – налог на добавленную стоимость, тенге.

Планируемая прибыль составляет 20% от себестоимости разработки

$$П_{\text{п}} = 625772,7 \cdot 0,2 = 125154,5 \text{ тенге}$$

НДС, начисленный на программу, осуществляется следующим образом

$$\text{НДС} = (Z_{\text{рпр}} + П_{\text{п}}) \cdot k_{\text{НДС}} \quad (4.20)$$

где $k_{\text{НДС}}$ – ставка налога на добавленную стоимость.

Подставив данные в формулу 4.20, получаем

$$\text{НДС} = (625772,7 + 125154,5) \cdot 0,12 = 90111 \text{ тенге}$$

Подставив данные в формулу 4.19, получаем

$$C_{\text{итп}} = 625772,7 + 125154,5 + 90111 = 841038 \text{ тенге}$$

4.4 Вывод по технико–экономической части

Проектирование базы данных для мобильного приложения «Прогноз футбольных ставок» является дорогим проектом, задействующим существенных затрат, а еще обязательного применения компьютерной техники и программного снабжения. В экономической части дипломного проекта был проведен расчет издержек на проектирование базы данных «Прогноз футбольных ставок» и расчет стоимости программного продукта. Стоимость разработки базы данных «Прогноз футбольных ставок» для мобильного приложения на платформе Android составляет 625772,7 тенге.

Программный продукт сконструирован для использования на коммерческой основе. На сегодняшний день множество азартных людей стремятся получить прибыль в букмекерских конторах. В таком случае данный проект является целесообразным, потому что он поможет людям увеличить доход на ставках. Цена реализации базы данных «Прогноз футбольных ставок» составит 936663,872 тенге. Создание данного программного продукта является экономически выгодной и эффективной.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

В настоящее время процесс развития техники и науки становится быстрым и непрерывным. Главную роль у людей играет возможность безопасного выполнения своих должностных обязанностей на работе.

Работа разработчика проекта в данной сфере связана с использованием компьютера, а соответственно с вредным воздействием некоторых факторов, которые существенно снижают продуктивность их работы. Компьютерная техника наносит значительный большой урон здоровью человека. Следовательно, работодателю нужно учесть все факторы для безопасной работы сотрудников, которые могут навредить их здоровью при работе с техникой.

Описание рабочего помещения:

- рабочее место расположено на 7 этаже;
- помещение состоит из двух соединенных комнат, размеры которых 5х3х3 и 5х6х3 (ширина, длина, высота);
- источники света являются светильники – 3 шт., в одном кабинете и 4 шт. во втором кабинете. В каждой комнате по 4 люминесцентные лампы;
- в помещении три окна (1+2), которые создают благоприятные условия для зрительного восприятия;
- жалюзи, для защиты от избыточной яркости с окон;
- в рабочем помещении работают 4 человека. Смена работы: дневная.

Режим работы с 8:20 до 18:00.

Схема рабочего места приведена на рисунке 5.1.

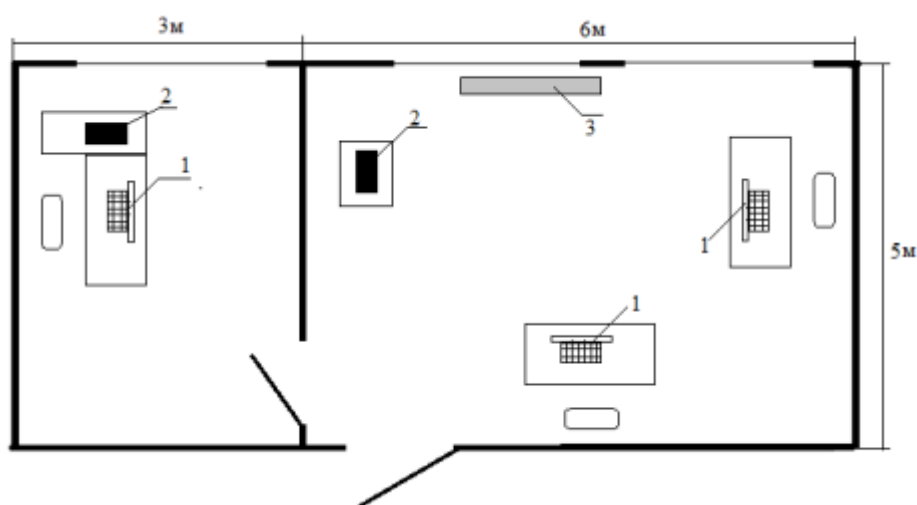


Рисунок 5.1 – План рабочего места

Условия труда сотрудника, работающего с персональным компьютером, определяются:

- особенностями организации рабочего помещения;
- условиями производственной среды (микроклиматом, освещением, шумом, электростатическими и электромагнитными полями и т. д.);
- описанием информационного взаимодействия персональных электронно-вычислительных машин и человека.

При работе с компьютером согласно ГОСТу 12.0.003-74 “ССБТ. Опасные и вредные производственные факторы. Классификация” могут иметь место следующие факторы:

- наиболее высокая температура поверхностей ПК;
- наиболее высокая или наиболее низкая температура воздуха рабочей зоны;
- выделение ряда химических веществ в воздух рабочей зоны;
- пониженная или повышенная влажность воздуха;
- повышенный или пониженный уровень отрицательных и положительных аэроионов;
- повышенное значение напряжения в электрической цепи, замыкание;
- повышенный уровень статического электричества;
- повышенный уровень электромагнитных излучений;
- повышенная напряженность электрического поля;
- отсутствие или недостаток естественного света;
- недостаточная искусственная освещенность рабочей зоны;
- повышенная яркость света;
- повышенная контрастность;
- зрительное напряжение;
- монотонность трудового процесса;
- нервно-эмоциональные перегрузки.

Мебель, которая часто используется разработчиком: это рабочий стол и кресло. Основное время работы разработчику приходится проводить в сидячем положении, которое практически не вызывает утомления.

Одним из главных факторов, оказывающих большое влияние на работоспособность, избежание болезней и утомлений сотрудника является освещенность. Утомляемость органов зрения человека зависит от следующих причин:

- недостаточность освещенности;
- излишняя освещенность;
- неправильное направление света.

Причиной напряжения зрения считается недостаточность освещения, ослабляется внимание, и появляется утомленность и усталость.

Излишняя освещенность рабочего помещения ослепляет и раздражает глаза.

Микроклимат рабочего места может оказывать пагубное влияние на состояние организма [13]. К примеру, повышенные температуры вызывают у человека перегрев организма, быструю утомляемость, и конечно же повышенное потоотделение, в то время как пониженные температуры способствуют общему или местному переохлаждению организма, к тому же могут привести к различным заболеваниям.

В кабинетах, где расположены компьютеры, особенно важен некие параметры микроклимата. При санитарных нормах «СН-245-71» при котором данные параметров микроклимата являются благоприятным для создания комфортного рабочего места работника. Эти санитарные нормы устанавливаются в зависимости от сезона. Характер производственных объектов и характера трудового процесса приведены в таблице 5.1.

Объем кабинетов, в которых размещены работники, составляют не меньше, чем 19.5 м³/человека с максимальным количеством одновременных работников в смену. Подача норм свежего воздуха в кабинеты, где есть компьютеры, приведены в таблице 5.2.

Таблица 5.1 - Параметры микроклимата

| Сезон года | Параметр микроклимата | Величина |
|------------|--------------------------------|---------------|
| Холодный | Температура воздуха в кабинете | 22...24 °С |
| Холодный | Относительная влажность | 40...60% |
| Холодный | Скорость движения воздуха | до 0,1 м/с |
| Теплый | Температура воздуха в кабинете | 23...25°С |
| Теплый | Относительная влажность | 40...60% |
| Теплый | Скорость движения воздуха | 0,1...0,2 м/с |

Таблица 5.2 - Нормы подачи свежего воздуха в помещения, где расположены компьютеры

| Характеристика кабинета | Объемный расход подаваемого в кабинет свежего воздуха, м ³ /на одного человека в час |
|---------------------------------------|---|
| Объем до 20м ³ на человека | Не менее 30 |
| 20...40м ³ на человека | Не менее 20 |
| Более 40м ³ на человека | Естественная вентиляция |

Для снабжения удобных условий организации работы в зависимости от сезона используют организационные методы (рациональной, чередование труда и отдыха) и оборудование (вентиляция, кондиционирование, системы отопления).

Правильно произведенное и выполненное промышленное освещение способствует к улучшению условия зрительной работы, позволяет снизить утомляемость, стремится к повышению производительности труда, а также

благоприятно влияют на производственную среду, оказывая положительное психологическое воздействие на сотрудника, позволяет повысить безопасность и снизить риск травм.

Недостаточное освещение рабочего помещения вызывает зрительное напряжение человека, понижает интерес, а также приводит к наступлению усталости. Чрезвычайно яркое освещение рабочего места вызывает бликов, следовательно, оказывая раздражение в глазах. Искажение направления света в рабочем помещении создают резкие тени, отражения, которые могут дезориентировать работника. Все вышесказанные факторы могут привести к профилактическому заболеванию или аварии, поэтому необходимо сделать верный расчет параметров освещения в помещении.

Освещение состоит из трех типов:

- естественное;
- искусственное;
- смешанное.

Негативными факторами для профессиональной деятельности разработчика, плохо влияющими на зрение человека, считаются:

- низкий уровень освещенности помещения, приводящий к перенапряжению глаз и, вследствие чего к быстрому утомлению.
- чрезвычайно высокая освещенность рабочего места также считается причиной быстрой утомляемости, которая приводит к раздражению и рези в глазах;
- неправильная направленность света приводит к появлению резких теней или сильных бликов, а также заметно утомляющих глаза.

Для освещения рабочего помещения с персональным компьютером должно использоваться искусственное, естественное и совмещенное освещение.

Рабочие места с персональным компьютером по отношению к световым проемам необходимо разместить так, чтобы естественный свет падал сбоку, в большой степени слева.

Согласно СНиП II-4-79 в вычислительных центрах рабочего помещения должны использоваться комбинированные системы освещения.

В результате проведения визуальной работы рабочего помещения категории высокой точности (значение минимального размера объекта дискриминации равно 0,3 ... 0,5 мм). Коэффициент естественного освещения (КЕО) не должна быть ниже 1,5%, в то время как средняя точность визуальных работ коэффициента естественного освещения должен быть не ниже 1.0%. Обычно, вместо источников искусственного освещения используются люминесцентные лампы типа ДРЛ или ЛБ, которые объединившись в светильники должны быть, размещены над рабочей поверхностью равномерно.

Необходимые требования освещения света в рабочих помещениях, где компьютеры устанавливаются следующим образом: зрительная работа

составляет 300 лк освещения, при комбинированной 750 лк; соответственно при работе средней класса точности требуется от 200 до 300 лк.

Одним из главных гигиенических требований является умеренное освещение всего поле зрения работающего сотрудника. Иными словами, степень освещенности рабочего помещения и насыщенность экрана персонального компьютера должны быть приблизительно одинаковыми, поэтому яркий свет в периферийной зоне рабочего помещения увеличивает напряженность глаз, вследствие чего приводит к их усталости. В таблице 5.3 описываются необходимые требования к освещенности помещения

Таблица 5.3 - Требования к освещенности помещения

| | | |
|---|--|------------|
| Освещение должно быть равномерным во всей комнате, особенно в рабочей зоне. Одинаково плохи и темный угол за шкафом, и непрерывно освещаемое солнцем пространство возле окна. | Нормы освещенности | |
| | Экрана | 100-250 лк |
| | Стола | 300-500 лк |
| | Яркость экрана должна быть не менее 35 кд/м ² ; | |

Шум, воздействуя в рабочее время на организм человека, понижает производительность труда его работы [14]. Работая длительное время при воздействии шума в рабочем помещении, разработчик чувствует на себе головные боли, раздражительность, головокружение, частичную потерю памяти, потерю аппетита, утомляемость, и боли в ушах. Целый список таковых нарушений систем человеческого организма приводит к неблагоприятному эмоциональному состоянию человека до стрессового состояния. Сосредоточивание интереса под воздействием сильного шума у сотрудника наблюдается снижение, повреждение физиологических функций, возникает утомляемость из-за затрачивания умственных нагрузок и энергии, и изменения в худшую сторону голоса переключения. Все это воздействует на работоспособности работника и на его продуктивность работы, а также на качество и безопасность работы.

В таблице 5.4 указаны предельные уровни звука в рабочем помещении.

Таблица 5.4 - Предельные уровни звука, дБ, на рабочих местах

| Категория напряженности труда | I. Низкая | II. Средняя | III. Высокая | IV. Очень высокая |
|-------------------------------|-----------|-------------|--------------|-------------------|
| Мало напряженный | 80 | 80 | 75 | 75 |
| Умеренно напряженный | 70 | 70 | 65 | 65 |
| Очень напряженный | 50 | 50 | - | - |

Уровень шума видеоматериалов на рабочем месте должно быть ниже 50 дБА, а в вычислительных центрах на ПК отделки информации не должно быть выше 65 дБА. Для значительного понижения уровни шума потолка и стены в помещении необходимо, чтобы компьютерная техника должна быть установлена противозумным материалом.

Современные эксперты считают, что воздействие всех видов излучений от монитора ПК не являются вредным для человека, который работает с ней. Необходимо подчеркнуть, что взаимоисключающие информации об опасности угрозы, которые связаны с излучением и изучение в этой сфере находится в состоянии решения.

В нижеприведенной таблице 5.5 показаны возможные значения параметров неионизирующих электромагнитных излучений при воздействии излучений от экрана компьютера.

На рабочей поверхности ПК, как правило, интенсивность инфракрасного и ультрафиолетового излучений должно находиться в диапазоне 10 – 100 м Вт/м², а наиболее высокий уровень рентгеновского излучения должно не превышать 10 мк Бар/ч.

Таблица 5.5 - Возможные значения параметров неионизирующих электромагнитных излучений

| Наименование параметра | Возможные значения |
|--|--------------------|
| На расстоянии 50см от грани монитора | 10В/м |
| На расстоянии 50см от грани монитора | 0,3А/м |
| Для учащихся в средних специальных и высших учебных заведениях, пользователей, а также для детей дошкольных учреждений | 20кВ/м 15кВ/м |

Для уменьшения нанесения вредности данных видов излучения необходимо работать с мониторами, которые имеют ограниченное излучение (ТСО-99, МРР-II), а также нужно установить щиты и сохранять регламентированные режимы отдыха во время работы.

Конструкция рабочих мест, которые оснащены видеотерминалами, является одним из главных вопросов дизайна в сфере компьютерных наук.

Одним из основных вопросов дизайнера в области компьютерных наук является проектирование рабочего места, которое оснащено видеотерминалами.

5.2 Организация рабочего места

Немаловажными элементами рабочего места сотрудника являются стол и кресло. Основное положение работы человека - сидя.

Во время рабочих заседаний у работника вызывает наименьшую степень усталости, в котором обычно он находится в положении сидя.

Рациональная планировка рабочего места гарантирует точность и очередность порядка расположения объектов, ручные комплектные наборы инструментов и документацию. То, что необходимо для работы, обычно расположено в зоне досягаемости рабочего места.

Двигатель поле - это место на поверхности рабочего места, при котором происходит движение работника.

Максимальное положение позиций руки - это часть рабочего места, который ограничен дугами, описано максимально возможное вытянутое руками двигательное действие плечевого размаха.

Оптимальная зона является частью моторного поля на поверхности рабочего стола, который ограничен дугами, описанные условно сравнительно фиксированными движениями предплечья в локтевых суставах опираясь на локоть и плечо.

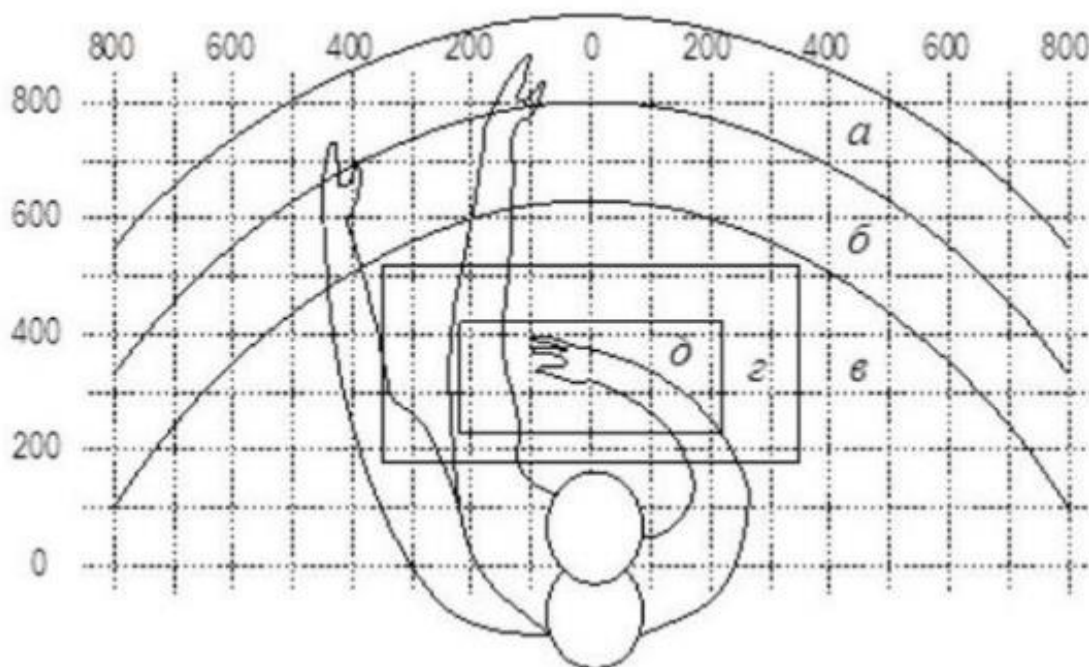


Рисунок 5.2 – Зона досягаемости рук в горизонтальной поверхности стола

Наиболее лучший вариант расположение предметов труда и документации работника в зонах досягаемости:

- максимальной досягаемости показано на зоне «а»;
- досягаемость пальцев при вытянутой руке указано в зоне «б»;
- легкой досягаемости ладони в зоне «в»;
- наиболее лучший вариант пространства для грубой ручной работы в зоне «г»;
- наиболее лучший вариант пространство для тонкой ручной работы в зоне «д».

Расположение предметов. В зоне «а» расположен монитор (в центре рабочей поверхности), системный блок помещен в указанное место по таблице ниже стола, клавиатура поставлена в зоне «г» и «д». Мышь находится в зоне «в» справа и сканер в области «а» и «б» слева. Принтер размещается в зоне «а» справа.

Документация на работе располагается так, что сотрудник мог достигать рукою до зоны «в», а в ящиках стола будет расположено литературная письменность. На рисунке 5.3 можно посмотреть, как верно расположены главные составляющие и периферийные устройства разработчика на рабочей поверхности персонального компьютера.

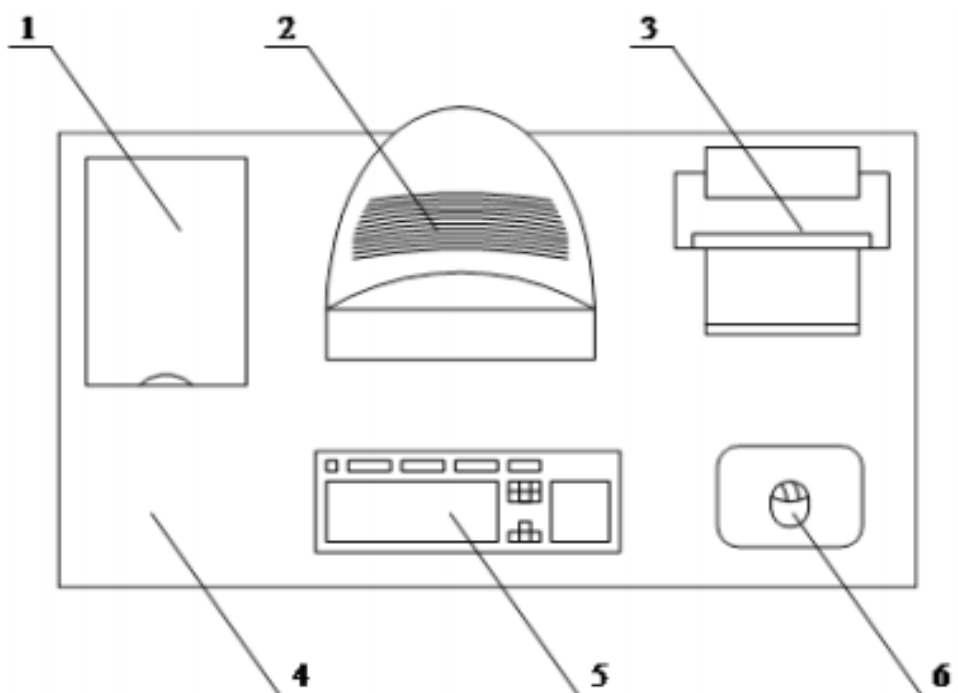


Рисунок 5.3 – Расположение основных и периферийных составляющих персонального компьютера

Расположение основных и периферийных составляющих персонального компьютера на рабочей поверхности разработчика:

- сканер;
- монитор;
- принтер;
- граница рабочей поверхности;
- клавиатура;
- мышь.

Требования рабочей поверхности разработчика для комфортной работы:

– высота стола должно представляться таким образом, для того чтобы была возможность сидеть свободно в комфортном расположении и не было необходимой надобностью основываться на подлокотники;

– в нижней части стола должны быть изготовлены места для того, чтобы разработчику комфортно было расположиться, и не было никакой надобности поджимать ноги;

– в поле зрения работающего разработчика граница, на которой он работает, должна владеть таким характерным свойством, чтобы было допустимым предотвратить возникновение бликов;

– стол обязан быть сконструирован так, для того чтобы подключать в себя некоторое количество выдвижных ящиков (для хранения деловых бумаг, списков и канцтоваров);

– оптимальные пределы для значения высоты грани рабочей поверхности от 680 до 760 мм.

Оптимальная высота расположения грани, для клавиатуры не должна превышать 650 мм. Особое внимание уделяется для свойства стола и стула разработчика. Исходя из этого, нормальная высота сиденья над полом должна находиться в диапазоне 430-560 мм. Желательно, чтобы сиденья были мягкими с закругленными передними краями, а также, чтобы угол наклона спинки регулировался.

При организации рабочего места необходимо снабдить расположение различных документов на рабочей поверхности со стороны видеотерминала и со стороны между монитором и клавиатуры.

Необходимо обеспечить при организации расположение различных документы на поверхности стола со стороны видео терминала, а также со стороны между монитора и клавиатуры. Нормальное расстояние от глаз до экрана монитора должно составлять около 700 мм, тогда расстояние от глаз до документа будет составлять примерно 300-450 мм.

Оптимальное расположение экрана монитора определяется:

– расстояние от глаз до монитора подсчитывается от 0,6 до 0,7 м;

– угол чтения просмотра зрения направленности в 20 градусов ниже горизонтальной, а также на дисплее монитора перпендикулярно к этому направлению.

Предусмотрена возможность управление экраном, т.е. передвигать в удобное положение:

– 3,5 см в высоту;

– по склону от -10 до +20 градусов относительно вертикали;

– по левой и правой стороне.

Огромный интерес оказывает верная рабочая поза юзера. Когда некомфортно сидеть на работе, то есть вероятность появления боли в мышцах, сухожилиях и суставах. Для комфортного условия к работе были предусмотрены следующие требования:

– при наклоне головы более чем на 20 градусов;

- расслабленные плечи;
- локти размещены под углом 80 - 100 градусов;
- руки и предплечья находятся в горизонтальном положении.

Следующие факторы являются причинами неправильной осанки пользователя: высокое размещение клавиатуры, документы находятся ниже, чем надо, отсутствие удобной подставки для документов.

Для преодоления этих недостатков предоставляются следующие рекомендации: использование беспроводных клавиатур; предоставление специальных инструментов для управления высоты стола, монитора, клавиатуры и доставки для рук.

Для продуктивной и качественной работы на персональном компьютере размеры символов, размещение их плотностей, контраст, а также яркость отношение символов являются необходимым.

Во время работы за компьютером рекомендуется установить монитор на расстоянии 50-60 см от глаз. По данным экспертов, в верхней части дисплей должен быть ниже или на уровне глаз.

Эксперты считают, что в верхней части видео должна быть на уровне глаз или ниже. Когда человек смотрит прямо, то его глаза открыты намного шире, чем когда он смотрит вниз.

Предложение благоприятных условий труда для работника, правильный внешний вид рабочего места в промышленности имеет гигантское значение, поскольку это упрощает производительность работы. А помимо прочего это повышает ее вкладывательную притягательность, собственно также положительно сказывается на производительности.

Часто отмечалось, собственно при работе с вычислительной техникой, играет очень главную роль надлежащее соблюдение работы и отдыха. В противном случае у сотрудников отмечали претензии: вескую напряженность в зрительном агрегате, боли в голове, расстройство и нарушение сна, боли шей, рук и в позвоночной области, что вызывало недовольство работой.

Информация о регламентированных перерывах представлены на таблице 5.6.

Таблица 5.6 – Время регламентированных перерывов

| Категория работы с ВДТ или ПК | Уровень нагрузки за рабочую смену при видах работы с ВДТ | | | Общее регламентированных перерывов | |
|-------------------------------|--|-----------------------------|----------------|------------------------------------|----------------------|
| | Группа А, количество знаков | Группа Б, количество знаков | Группа В часов | При 8- часовой смене | При 12-часовой смене |
| I | до 20000 | до 15000 | до 2,0 | 30 | 70 |
| II | до 40000 | до 30000 | до 4,0 | 50 | 90 |
| III | до 60000 | до 40000 | до 6,0 | 70 | 120 |

Перерывы введены в соответствии с конкретными медико-санитарными правилами и нормами. Время перерывов увеличивается на 30% при несоответствии практических критерий труда санитарным правилам и нормам. Все виды работ, относящиеся к работам с применением вычислительной машиной, разделены на три категории (в соответствии с СанПиН 2.2.2 546-96). Категория «А» имеет в своем составе работу на чтение инфы с экрана ВДТ либо ПК с ранним запросом. Ввод информации включает в себя категория «Б». Творческая работа режима диалога с компом относится к группе «В».

Комфортная нежная мебель, месторасположение аквариумов, живых растений, или умышленно выделенного помещения с газонем хорошо отразятся при работе во время перерывов. И еще можно заняться нетяжелой гимнастикой, что эффективно отразится на работе разработчика программного обеспечения.

5.3 Расчет искусственного освещения

Освещенность рабочей поверхности, которая создается светильниками общего освещения в системе, обязана оформлять минимум 10% нормируемой для сочетанного освещения при тех источниках света, которые используются для местного освещения. При этом освещенность обязана быть не менее 400 лк при люминесцентных лампах. В таблице 5.7 приведены общепризнанных мерок освещенности при происхождения искусственного освещения.

Таблица 5.7 - Нормы освещенности при искусственном освещении

| Характеристика зрительной работы | Наименьший или эквивалентный размер объекта различения, мм. | Разряд зрительной работы | Искусственном освещении |
|----------------------------------|---|--------------------------|---|
| | | | Освещенность, лк при комбинированном освещении. |
| Средней точности | Свыше 0.5 до 1.0 | IV | 400 |

Расчет освещенности рабочего помещения приводится к выбору системы освещения, определению нужного числа светильников, их типа и размещения. Ход работы разработчика в определенных условиях, как скоро природное освещение недостаточно либо отсутствует. Исходя из этого, рассчитаем параметры искусственного освещения.

Искусственное освещение создается при помощи электрических источников света двух видов: ламп накаливания и люминесцентных ламп.

Искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп.

Станем использовать люминесцентные лампы, которые, по сопоставлению с лампами накаливания, имеют значительные плюсы:

- по спектральному составу света они близки к дневному;
- натуральному свету;
- владеют более высоким КПД (в 1,5-2 раза выше, чем КПД лампочек);
- владеют завышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
- наиболее долгие срок эксплуатации.

Расчет освещения производится для комнаты, длина равняется 9 м, ширина - 6 м. Найдем значение освещенности этого здания и сопоставим его с нормированным значением, чтобы определить достаточно ли текущего искусственного освещения для работы в помещении.

Вычислим высоту подвеса светильника над рабочей поверхностью

$$H = h - h_p - h_c, \quad (5.1)$$

где h_c – расстояние от светильника до перекрытия $h_c = 0,05$ м;
 h_p – высота рабочей поверхности над полом, $h_p = 0,8$ м;
 h – высота помещения, $h = 3$ м;
 $H = 3 - 0,8 - 0,05 = 2,15$ м.

Наиболее выгодное расстояние между светильниками определяется как

$$L = \lambda \cdot H, \quad (5.2)$$

где $\lambda = 1,2 \div 1,4$

$$L = 1,3 \cdot 2,15 = 2,795 \text{ м.}$$

Определяем индекс помещения

$$i = \frac{S}{H \cdot (A+B)}, \quad (5.3)$$

где S – площадь помещения, $S = 45 \text{ м}^2$;
 H – расчетная высота подвеса, $H = 2,15$ м;
 A – ширина помещения, $A = 6$ м;
 B – длина помещения, $B = 9$ м.

Подставив значения получим:

$$i = \frac{45}{2,15 \cdot (6+9)} = 1,495,$$

Коэффициенты отражения от потолка, стен и пола равны:

$$P_{пот} = 50\%$$

$$P_{ст} = 10\%$$

$$P_{пол} = 30\%$$

Зная индекс помещения i находим коэффициент использования $\eta = 40\%$;

K_z – коэффициент запаса: $K_z = 1,2$.

Формула для расчета освещенности

$$E = \frac{N \cdot \Phi_{л} \cdot \eta \cdot \tau}{S \cdot K_z \cdot z}, \quad (5.4)$$

Для освещения выбираем люминесцентные лампы типа ЛД-65, световой поток которых $\Phi_{л} = 3750$ Лк.

Подставим все значения в формулу (5.4) и получим

$$E = \frac{7 \cdot 3750 \cdot 0,4}{45 \cdot 1,2 \cdot 1,1} = 388,89 \text{ Лк.}$$

Нормируемая освещенность составляет 400 Лк, а освещенность в помещении 388,89 Лк. Значит искусственное освещение в помещении в пределах нормы. Значит искусственное освещение в пределах нормы.

Так как L = наиболее выгодное расстояние между светильниками, l - расстояние от крайних светильников или рядов светильников до стены

$$l = 0,3 \div 0,5 L, \quad (5.5)$$

$$l = 0,3 \cdot 2,795 = 0,8385 \text{ м.}$$

Предлагается установить в первой комнате 3 светильника, 2 в ряд, так как длина комнаты 5 метров, а ширина 3 метра. А во второй комнате 4 светильника (рисунок 5.4)

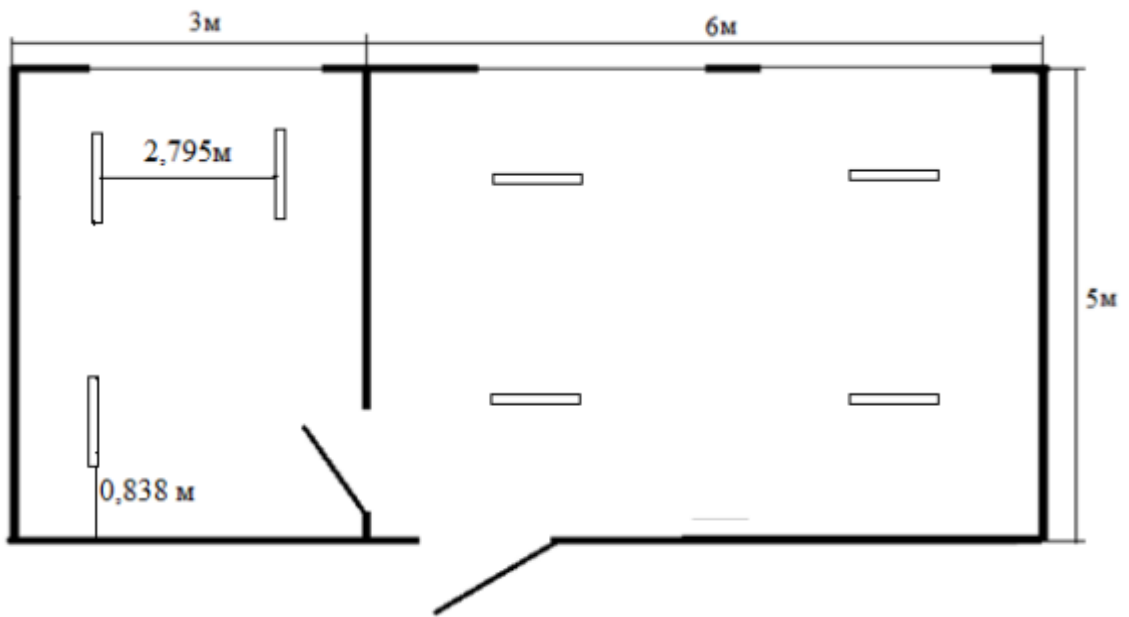


Рисунок 5.4 – Схема расположения светильников

Заключение

В ходе написания дипломной работы мною было разработано база данных «Прогноз спортивных событий» для мобильного приложения, которое позволит хранить необходимые информации в базе. Для проектирования базы данных было использовано СУБД SQLite, т.к. SQLite считается наиболее удобной по функциональным возможностям при разработке базы данных для мобильного приложения платформы Android.

Для достижения вышеуказанных целей были поставлены и решены следующие задачи: исследовать предметную область в сфере ставок в букмекерских конторах, изучить структуру базы данных, произвести полный обзор аналогов на рынке, выбрать и обосновать информационные технологии, разработать структуру базу данных.

При исследовании предметной области «Прогноз спортивных событий», изучены данные и получены навыки в области разработки баз данных в SQLite.

Программный продукт работает на платформе Android, которая имеет огромную популярность в сфере операционных систем для мобильных устройств. Данный совместный проект является целесообразным, в наши дни сфера азартных игр и ставок считается очень популярной. Люди хотят получить выгоду на ставках, следовательно, данный программный продукт даст им больше возможностей для удачного прогноза спортивного события. В ближайшее время планируется разместить данный проект в интернет магазине приложений «Play Market» и получать прибыль от продажи данной интеллектуальной собственности. Область мобильного приложения в настоящее время является распространенной и актуальной в нашей стране, он также социально значимый и наверняка будет пользоваться популярностью.

В технико-экономической части дипломной работы была рассчитана общая стоимость программного продукта и расчет затрат при ее разработке. Исходя из полученных данных, можно сделать вывод, что данный программный продукт является экономически эффективным и выгодным.

В части безопасности жизнедеятельности был произведен анализ организации рабочего места, расчет искусственного и естественного освещения помещения.

Список литературы

- 1 Бутакова М.М.. Экономическое прогнозирование: методы и приемы практических расчетов : учебное пособие / М.М. Бутакова. — 2-е изд., испр. — М.: КНОРУС. - 168 с., 2010
- 2 Арефьева Н. Т. Прогнозирование и его социокультурные цели // Электронный журнал «Знание. Понимание. Умение». — 2010. — № 4 — Культурология.
- 3 Бехтер Л. В., группа ЭК-401-д. Прогнозирование исхода футбольных матчей / Л. В. Бехтер, Н. И. Клевец // Социально-экономическое развитие АР Крым: проблемы и перспективы. Сб. науч. трудов. Вып. 6. — Симферополь : НАПКС, 2012. — С. 268—270.
- 4 Бестужев-Лада И.В. Рабочая книга по прогнозированию, М., Мысль, 1982. – 125 с.
- 5 Брусиловский Б.Я. Математические модели в прогнозировании и организации науки [Текст] / Б.Я. Брусиловский. – Киев: Наук, думка, 2005. – 232 с.
- 6 Конноли Т., Бегг К Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е изд Изд-во «Вильямс», 2003, 1436 с
- 7 Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. - М.: Наука. Гл. ред. физ.-мат. Лит., 1987 г. – 221 с.
- 8 Леоненков А.В Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006, 319 с
- 9 Роджерс Р., Ломбардо Д. Android. Разработка приложений. — М.: ЭКОМ Паблшерз, 2010. — 400 с. — ISBN 978-5-9790-0113-5.
- 10 Любушин Н.П. Экономический анализ. 3-е изд., перераб. и доп./ Н.П. Любушин — Издательство: Юнити, 2010. – 575 с.;
- 11 Гейц И.В. Заработная плата и другие расчёты с физическими лицами: Практическое пособие для бухгалтера. – М.: Издательство «Дело и сервис», 1999. 640 с.
- 12 Кокин Ю.П., Шлендер П.Э. Экономика труда. Учебник. - 2-е изд., перераб. И доп. – М.: Магистр, 2010 – 686 с.
- 13 ГОСТ 12.1.005-88 "Общие санитарно-гигиенические требования к воздуху рабочей зоны". — С. 4.
- 14 Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274

Приложение А

Листинг программы

```
package com.example.batman.toolbarnavigationversion100500;

import android.content.Context;
import android.database.DatabaseErrorHandler;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.provider.BaseColumns;
import android.util.Log;

public class DbHelper extends SQLiteOpenHelper {

    private static final String DataBase_Name = "myDataBase.db";
    private static final int DataBase_Version = 1;
    private static final String DataBase_Table_BETS = "Bets";
    public static final String Bets_match = "Match";
    public static final String Bets_P1 = "Win1";
    public static final String Bets_X = "X";
    public static final String Bets_P2 = "Win2";

    private static final String DataBase_CreateTable_Bets= "create table "
        + DataBase_Table_BETS + "( Bets_ID integer primary key autoincrement, " +
    Bets_match
        + " text, " + Bets_P1 + " text, " + Bets_X
        + " text, " + Bets_P2 + " text);";

    private static final String DataBase_Table_User = "User";

    public static final String User_Login = "Login";
    public static final String User_Password = "Password";
    public static final String User_Email = "Email";

    private static final String DataBase_CreateTable_User= "create table "
        + DataBase_Table_User + "( _id integer primary key autoincrement, " + User_Login
        + " text, " + User_Password + " text, " + User_Email
        + " text);";

    private static final String DataBase_Table_History = "History";

    public static final String History_Login = "Login";
    public static final String History_Match = "Match";
    public static final String History_Bet = "Bet";

    private static final String DataBase_CreateTable_History= "create table "
        + DataBase_Table_History + "( _id integer primary key autoincrement, " + History_Login
        + " text, " + History_Match + " text, " + History_Bet
        + " text);";
```

```

public DBHelper(Context context){
    super(context, DataBase_Name, null, DataBase_Version);
}
public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int
version) {
    super(context, name, factory, version);
}
public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int
version, DatabaseErrorHandler errorHandler) {
    super(context, name, factory, version, errorHandler);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(DataBase_CreateTable_Bets);
    db.execSQL(DataBase_CreateTable_User);
    db.execSQL(DataBase_CreateTable_History);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.w("SQLite", "Обновляемся с версии " + oldVersion + " на версию " + newVersion);
    db.execSQL("DROP TABLE IF IT EXISTS " + DataBase_Table_BETS);
    onCreate(db);
}
}
package com.example.batman.toolbarnavigationversion100500;

import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

import java.util.ArrayList;

public class Dialog_ListOfMatches extends DialogFragment implements OnClickListener{
    String stavka;
    String mat4;
    String login;

    Bundle bundle = new Bundle();

```

```

ArrayList<String> grev = new ArrayList<String>();
Button button_p1;
Button button_x;
Button button_p2;
TextView txt;

public Dialog_ListOfMatches() {
    // Required empty public constructor
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    grev = this.getArguments().getStringArrayList("asd");
    for(String s:grev){
        Log.d("LOOOOOGTAAAG",s);
    }

    getDialog().setTitle("BET");

    View v =inflater.inflate(R.layout.dialog__list_of_matches,container,false);

    v.findViewById(R.id.btn_p1).setOnClickListener(this);
    v.findViewById(R.id.btn_x).setOnClickListener(this);
    v.findViewById(R.id.btn_p2).setOnClickListener(this);

    button_p1 = (Button) v.findViewById(R.id.btn_p1);
    button_x = (Button) v.findViewById(R.id.btn_x);
    button_p2 = (Button) v.findViewById(R.id.btn_p2);
    txt = (TextView) v.findViewById(R.id.txt);

    txt.setText(grev.get(0));
    button_p1.setText(grev.get(1));
    button_x.setText(grev.get(2));
    button_p2.setText(grev.get(3));
    return v;
}

@Override
public void onClick(View v) {
    DBHelper mDbHelper = new DBHelper(getActivity(),"myDataBase.db",null,1);
    SQLiteDatabase sdb = mDbHelper.getWritableDatabase();
    ContentValues cv = new ContentValues();

    mat4 = txt.getText().toString();
    stavka = ((Button)v).getText().toString();

    cv.put(DBHelper.History_Login,MainActivity.USER_LOGIN);

```

```

        cv.put(DbHelper.History_Match,mat4);
        cv.put(DbHelper.History_Bet,stavka);

        sdb.insert("History",null,cv);
        Log.d(")))))))))", "Match = "+ mat4 + ", stavka = "+stavka);
        Log.d("LOG_TAG", "Dialog : " + ((Button) v).getText());
        dismiss();
    }
}
package com.example.batman.toolbarnavigationversion100500;

import android.content.Context;
import android.database.Cursor;
import android.os.Bundle;

import android.support.v4.app.DialogFragment;
import android.support.v4.app.ListFragment;
import android.support.v4.app.LoaderManager;
import android.support.v4.content.CursorLoader;
import android.support.v4.content.Loader;
import android.support.v4.widget.SimpleCursorAdapter;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;

public class Fragment_History extends ListFragment implements
LoaderManager.LoaderCallbacks<Cursor> {

    String[] from = new String[] {DbHelper.History_Login,DbHelper.History_Match,DbHelper.History_Bet};
    String[] to = new String[] {R.id.login_history,R.id.match_history,R.id.bet_history};
    int[] to = new int[] {R.id.login_history,R.id.match_history,R.id.bet_history};
    ListView lv_history;
    Cursor cursor;
    SimpleCursorAdapter smpl_cursor_adapter;
    DbHelper mDbHelper;
    //SQLiteDatabase sdb = mDbHelper.getReadableDatabase();

    public Fragment_History() {
        // Required empty public constructor
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        if(MainActivity.USER_LOGIN == "") {
            DialogFragment frg_signin = new Fragment_SIgn();
            frg_signin.show(getFragmentManager(), "frg_signin");
        }
    }
}

```

```

        mDbHelper = new DbHelper(getActivity(),"myDataBase.db",null,1);
        smpl_cursor_adapter =
SimpleCursorAdapter(getActivity(),R.layout.for_history,null,from,to,0);
        //lv_history = (ListView) v.findViewById(R.id.lv_history);
        setListAdapter(smpl_cursor_adapter);
        // MainActivity.USER_LOGIN = "Eldar";
        getLoaderManager().initLoader(0, null,this);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        View v =inflater.inflate(R.layout.fragment__history, container, false);

        return v;
    }

    @Override
    public Loader<Cursor> onCreateLoader(int id, Bundle args) {
        Log.d("ebalashkaEldar", "onCreate");
        return new MyCursorLoader(getActivity(),mDbHelper);
    }

    @Override
    public void onLoadFinished(Loader<Cursor> loader, Cursor data) {
        smpl_cursor_adapter.swapCursor(data);
        Log.d("ebalashkaEldar", "onLoadFinished");
    }

    @Override
    public void onLoaderReset(Loader<Cursor> loader) {
        Log.d("ebalashkaEldar", "onLoadReset");
        smpl_cursor_adapter.swapCursor(null);
    }

    public static class MyCursorLoader extends CursorLoader {

        DbHelper db;

        public MyCursorLoader(Context context,DbHelper db) {
            super(context);
            this.db =db;
        }

        @Override
        public Cursor loadInBackground() {
            String selection ="Login LIKE ?";

```

```

        String[] selectionArgs = new String[]{MainActivity.USER_LOGIN};
        Cursor cursor = db.getWritableDatabase().query("History", null, selection, selectionArgs,
null, null, null);
        if(cursor != null && cursor.getCount(>0) {
            while (cursor.moveToNext()) {
                Log.d("#####",
cursor.getString(cursor.getColumnIndex(DbHelper.History_Login)));
                Log.d("#####",
cursor.getString(cursor.getColumnIndex(DbHelper.History_Match)));
                Log.d("#####",
cursor.getString(cursor.getColumnIndex(DbHelper.History_Bet)));

            }
        }
        return cursor;
    }
}

@Override
public void onDestroyView() {
    super.onDestroyView();

    mDbHelper.close();
}
}
package com.example.batman.toolbarnavigationversion100500;

```

```

import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;

```

```

import android.support.v4.app.DialogFragment;

```

```

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;

```

```

public class Fragment_Registration extends DialogFragment implements View.OnClickListener {
    EditText Login;
    EditText Password;
    EditText Email;

    public Fragment_Registration() {
        // Required empty public constructor
    }
}

```

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View v =inflater.inflate(R.layout.fragment__registration, container, false);

    getDialog().setTitle("Registration");
    v.findViewById(R.id.go).setOnClickListener(this);
    Login = (EditText) v.findViewById(R.id.login_reg);
    Password = (EditText) v.findViewById(R.id.password_reg);
    Email = (EditText) v.findViewById(R.id.email);
    return v;
}

@Override
public void onClick(View v) {

    Log.d("555555555", "Login = "+ Login.getText().toString() + "Password = "
+Password.getText().toString() +"Email = "+Email.getText().toString());

    MainActivity.USER_LOGIN = Login.getText().toString();
    DBHelper dbHelper = new DBHelper(getActivity(), "myDataBase.db", null, 1);
    SQLiteDatabase sdb = dbHelper.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(DBHelper.User_Login, Login.getText().toString());
    cv.put(DBHelper.User_Password, Password.getText().toString());
    cv.put(DBHelper.User_Email, Email.getText().toString());
    sdb.insert("User", null, cv);
    sdb.close();
    dismiss();
}
}

```