СОДЕРЖАНИЕ

[Введение 4](#_Toc531957641)

[1 Теоретическая часть 8](#_Toc531957642)

[1.1 Выбор языка и среды программирования 8](#_Toc531957643)

[1.2 Разработка системы требований 11](#_Toc531957644)

[1.3 Разработка функциональной спецификации 12](#_Toc531957645)

[2 Практическая часть 19](#_Toc531957646)

[2.1 Разработка CRC – карточек 19](#_Toc531957647)

[2.2 Структура и функциональная схема 21](#_Toc531957648)

[2.3 Диаграмма деятельности 23](#_Toc531957649)

[2.4 ER-диаграмма 24](#_Toc531957650)

[2.5 Диаграмма классов 28](#_Toc531957651)

[2.6 Тестирование программного обеспечения 29](#_Toc531957652)

[Заключение 39](#_Toc531957653)

[Список литературы 43](#_Toc531957654)

[Приложение А 44](#_Toc531957655)

[Приложение Б 50](#_Toc531957656)

[Приложение В 51](#_Toc531957657)

[Приложение Г 59](#_Toc531957658)

Введение

На сегодняшний день профессиональный спорт настоятельно требует качественного управления на основе последних достижений и актуальных тенденций международного спортивного менеджмента. Так, для поддержания экономической эффективности клубы должны активно работать со своими болельщиками, рекламодателями и спонсорами, министерства и ведомства – контролировать целевое использование средств на подготовку спортсменов и проведение спортивных мероприятий.

Большой спорт и большой бизнес – эти понятия тесно связаны между собой. А связь высоких достижений и высокой доходности уже давно стала закономерностью. И правильное, целесообразное управление этими доходами наряду с грамотно организованным тренировочным процессом сегодня является одним из самых важных факторов успеха в спорте.

Современные спортивные организации: клубы, ассоциации, федерации – решают целый спектр традиционных и специфических задач управления. Среди первых — управление финансами, складом, закупками и продажами, персоналом и контрактами игроков. Среди вторых – управление подготовкой спортсменов и работой с болельщиками, учет членских взносов, управление спортивной инфраструктурой: стадионами, тренировочными базами, – ведение спортивной статистики. Разнообразие форматов спортивных событий проявляется и в разнообразии объектов автоматизации в мире спорта.

Технологической особенностью автоматизации спортивных организаций является необходимость стыковки единого информационного пространства клуба, федерации и т.д. с большим количеством сторонних информационных систем, среди которых – внешний портал или Web-сайт клуба, интернет-магазин, системы бронирования или продажи билетов, системы контроля доступа и т.д. И обойтись без такой полной интеграции информационного ландшафта нельзя.

Другая грань спорта – специализированные комплексы и объекты недвижимости: стадионы и дворцы спорта, базы подготовки спортсменов, прочие спортивные объекты, – автоматизация управления которыми решает свои задачи. В числе них – управление арендными отношениями, учет заявок на проведение мероприятий, планирование и контроль загрузки элементов инфраструктуры: полей, кортов, залов, катков, – управление работой с рекламодателями и учет рекламных поверхностей.

Безусловно, решение названных выше задач требует мощной технологической поддержки, предполагающей полномасштабное внедрение CRM-систем и интеграционных продуктов.

Конечно, отечественный рынок автоматизации спортивных организаций пока сравнительно невелик, однако, первые результаты, достигнутые российскими интеграторами и профессиональными спортивными клубами, позволяют уверенно говорить о значимости внедрения современных информационных систем в мире спорта.

Проблема: как составить модель программного продукта, позволяющего автоматизировать работу персонала спортивных мероприятий.

Цель работы: спроектировать информационную систему спортивных мероприятий, разработать часть функционала и протестировать его.

Задачи:

1. Выбрать среду разработки и обосновать этот выбор.
2. Составить системные требования к разрабатываемой информационной системе.
3. Выявить основных лиц, взаимодействующих с информационной системой.
4. Составить диаграмму прецедентов.
5. Изучить структуру проектируемой информационной системы с помощью составления CRC-карточек.
6. Составить структурную схему.
7. Выявить основной функционал.
8. Разработать алгоритм работы по средствам диаграммы деятельности.
9. Спроектировать базу данных с помощью диаграммы «Сущность - связь».
10. Составить диаграмму классов.
11. Разработать тестовые сценарии для проверки работоспособности информационной системы.
12. Составить сопутствующую техническую документацию.

В ходе выполнения курсовой работы необходимо изучить предметную область.

Разработать алгоритм работы таких функций программного обеспечения как:

1. Просмотр информации.
2. Авторизация/регистрация пользователя в системе.
3. Регистрация бегуна на марафон.
4. Редактирование профиля пользователя.
5. Импорт данных.

Проанализировать различные средства автоматизации по таким параметрам, как знание языка разработки, удобство среды, рационального выполнения. Структурировать систему требований, выявить пред причины разработки информационной системы. Составить систему ограничений программного продукта с разных сторон. Проанализировать ситуации, при которых в работе программного продукта могут возникнуть исключительные ситуации. Выявить основных действующих лиц и описать их поведение в системе с помощью диаграммы вариантов использования.

Для полноценной работы составляются CRC-карточки, которые показывают взаимодействие всех классов систем, после чего они проверяются с помощью диаграммы взаимодействия. Структурная схема отражает взаимодействие составных систем информационной системы, а также функциональная схема, которая показывает работу информационной системы. Процесс работы информационной системы отображается на диаграмме деятельности, для построения которой необходимо учитывать процесс работы каждой функции. Для хранения необходимой информации проектируется база данных. Для этого выделяются необходимые сущности, указываются их атрибуты и связи между ними. Взаимодействие всех классов системы, а также их свойств и методов отражается на диаграмме классов. Разработка интуитивно понятного интерфейса является не менее важным направление работы, поскольку интерфейс пользователя - это элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением.

После разработки части функционала, его необходимо протестировать. Для этого выбираются методы «Черного ящика» и «Белого ящика».

Метод черного ящика заключается в тестировании на стороне пользователя, а метод белого ящика позволяет тестировать на стороне разработчика с использованием программного кода.

Для работы пользователей разрабатывается несколько программных документов.

Например, руководство пользователя, содержащее алгоритм работы с информационной системой, технические характеристики и всевозможные ошибки.

Техническое задание регламентирует назначение разработки, требования к программному продукту и сопроводительной документацией, а также учитываются все стадии разработки, и порядок контроля и приемки информационной системы. Техническое задание является юридическим документом — как приложение включается в договор между заказчиком и исполнителем на проведение проектных работ и является его основой: определяет порядок и условия работ, в том числе цель, задачи, принципы, ожидаемые результаты и сроки выполнения. Руководство по техническому обслуживанию помогает техникам в сопровождении разработанного программного продукта.

1 Теоретическая часть

1.1 Выбор языка и среды программирования

Для разработки информационной системы была выбрана среда разработки - 1С предприятие.

1С: Предприятие - это система программ, предназначенная для автоматизации различных областей экономической деятельности предприятий, организаций и учреждений, независимо от их вида деятельности и формы собственности, с различным уровнем сложности учета. За счет своей универсальности она используется для автоматизации самых разных участков экономической деятельности предприятия: бухгалтерский, кадровый, оперативный торговый, складской и производственный учет, а также расчет заработной платы, учет товарных и материальных средств, взаиморасчетов с контрагентами.

Встроенный язык программирования 1С - язык программирования, который используется в семействе программ «1С:Предприятие». Язык является предварительно компилируемым предметно-ориентированным языком высокого уровня. Средой исполнения языка является программная платформа «1С:Предприятие». Визуальная среда разработки («Конфигуратор») является неотъемлемой частью пакета программ «1С:Предприятие». Платформой предоставляется фиксированный набор базовых классов, ориентированных на решение типовых задач прикладной области: справочник, документ, журнал документов, перечисление, отчет, регистр и др.

В режимном управлении приложения интерфейс не «изображается», а «описывается». Разработчики фиксируют всю схему командного интерфейса и в тоже время одновременно и схему форм. Такое описание платформа применяет для построения интерфейса направленно для определенного о пользователя, учитывая все факторы и сохраняя при этом все настройки, сделанные непосредственно самим пользователем.

Координируемое управляемое приложение в большей мере облегчает введение теоретических решений. Теперь можно не заниматься доработкой, а просто настраивать приложения для специфики определенной компании, оставляя при этом неизменным само практическое решение. Управляемое приложение дает возможность включать и выключать дополнительные возможности и более сложные функциональные решения, которые могут понадобиться в практических разработках.

В последней версии платформы «1С: Предприятие 8» включена возможность использования кластеров сервера для защиты от непредвиденных отказов системы или как её еще называют отказоустойчивость кластер серверов – эта возможность позволяет обеспечивать взаимодействие между пользователем и СУБД (Система управления базами данных)

Теперь вся информация сохраняется в базе кластер серверов и сбои в работе пользовательских компьютеров или внезапное отключение по непредвиденным обстоятельствам не приведет к потере информации. Все данные будут зарезервированы вместе с резервированием рабочих процессов внутри действующего кластера.

Теперь существенно повышена и устойчивость системы к внезапному обрыву связи между пользователем и работающим кластером серверов. В такой ситуации после восстановления прерванного соединения пользователь сможет, как обычно продолжить работу именно с того «места», где произошел обрыв связи. Происходит то, что кластер запоминает всех подключенных пользователей и те действия которые они выполняли. Вся информация восстановиться при подключении соединения именно с того места где произошел сбой.

Механизм агрегатов для оптимизации построения аналитических отчетов. Применение агрегатного механизма существенно сокращает отрезок времени на форматирование всех отчетов. Что качественно отражается при работе с многомиллионными информационными базами данных и реестровых записей.

Агрегатные механизмы применяются для оборотных регистров накапливания. Для определенного регистра могут создаваться несколько специальных хранилищ, содержащих объединенные шифрованные данные, которые удобны для последующего формирования данных информационной базы. Система автоматически дает оценку напряженности работы пользователей с любым разделом информации и на основе выверенной статистики определяет лучший состав работающих агрегатов.

Применение агрегатных механизмов позволит анализировать специалистам существующую информацию, переходя между разными разделами просмотра с незначительным временем отклика системы. Происходит это благодаря накопленным агрегированным данным систем с одной стороны и актуальным обеспечением принимаемых отчетов, с другой стороны.

Для любого отчета конфигурации консультант или же разработчик может сформировать несколько различных вариантов одного и того же отчета. И любой из вариантов будет выдавать одни и те же данные отчета в разном виде. Пользователь выбирает из готовых вариантов тот, который ему больше подходит и выполняет отчет.

Все настройки, которые доступны любому пользователю описываются разработчиками как пользовательские настройки, а те, которые непосредственно присутствуют в форме отчета, и описываются как быстрые пользовательские настройки. Такие настройки помогут составить форму отчета непосредственно для себя, сохранить и использовать уже готовый вариант отчета в дальнейшем. Разработчик может описать, какие настройки отчета будут доступны пользователю (пользовательские настройки) и какие из этих настроек будут присутствовать прямо в форме отчета (быстрые пользовательские настройки). С их помощью пользователь может настроить вариант отчета «под себя» и сохранить его для дальнейшего использования.

Повышена скорость для работы с материалами базы данных. Повышена скорость выполнения запросов к базе данных. Усовершенствовано использование оперативной памяти и электронно-вычислительных ресурсов компьютера. Снижено время начального открытия файлов конфигурации. Другие изменения для быстрого реагирования системы

1.2 Разработка системы требований

Для разработки информационной системы спортивных мероприятий необходимо составить систему требований к программному продукту и выделить основные этапы.

Этап 1. В первую очередь нужно правильно определить проблему.

1. Структурирование проблемы

|  |  |
| --- | --- |
| Элемент | Описание |
| Проблема | Как составить информационную систему позволяющую вести учет на спортивных мероприятий |
| Воздействует на что (кого) и результатом чего является | Пользователей, Координаторов, Бегунов, Администратор |
| Выигрыш от решения может состоять в следующем | Изучение предметной области и проектирование модели информационной системы. Результатом станет создания информационной модели с последующим ее созданием |

Этап 2. Причины отсутствия информационной модели:

* Замедленная работа персонала;
* Недостаточная качественность обработки результатов;
* Допущение ошибок работников при исполнении обязанностей;
* Отсутствие хранения информации на сервере.

Этап 3. Системой будут пользоваться следующие пользователи:

1 Администратор – сопровождение СУБД.

2 Координатор – управляет бегунами.

3 Бегун – участник соревнований.

Появление новой системы положительно скажется на работе компании. Создание системы нужно для повышения качества работы сотрудников фирмы. Заказчиком системы является организатор спортивных мероприятий. Система окажет положительное влияние на работу. В первую очередь повыситься качество работы сотрудников, а вторым фактором станет повышение прибыли компании.

Этап 4. При проектировании информационной системы будут вводится ограничения. Это сделано для реализации всех запланированных функции, которые должна будет выполнять проектируемая информационная система.

Управлением информационной системой будет заниматься администратор. Удаленно системой будет пользоваться специалист налаживающий работу системы и исправляющий возможные ошибки в ней. Вся информация будет хранится на сервере. При каждом обращении пользователь получает информацию из сервера за счет некоторых действий и сервер присылает данные, выводя их на дисплей.

Этап 5. Основными ограничениями информационной системы являются:

* Среда проектирования.
* Защита персональных данных.
* Шифрование данных.
* Среда разработки.
* Соединение с БД.

1.3 Разработка функциональной спецификации

Для разработки функциональной спецификации информационной системы, необходимо проанализировать требования к его функциональности, основой которого являются варианты его использования.

Основными действующими лицами в системе будут:

* Администратор.
* Координатор.
* Бегун.

Основные прецеденты бегуна:

* Редактирование профиля.
* Регистрация/Авторизация.
* Просмотр результатов бегуна.
* Регистрация бегуна на марафон.

Основные прецеденты координатора:

* Редактирование профиля бегуна.
* Авторизация.
* Просмотр бегунов.
* Печать сертификата об участии.

Основные прецеденты администратора:

* Авторизация.
* Импорт данных.
* Просмотр пользователей.
* Просмотр благотворительных организации.
* Добавление/изменение благотворительных организаций.
* Добавление нового пользователя.
* Редактирование пользователей.
* Просмотр волонтеров.

Краткое описание вариантов использования системы представлено в следующих таблицах.

1. Вариант использования «Авторизация пользователя»

|  |  |
| --- | --- |
| Название варианта | Авторизация пользователя |
| Цель | Авторизация пользователя в системе |
| Действующие лица (актеры) | Бегун, координатор, администратор |
| Краткое описание | Пользователь регистрируется, вводит свои данные, входит под своей учетной записью, для дальнейших действий. |
| Тип варианта | Основной |

1. Вариант использования «Регистрация бегуна»

|  |  |
| --- | --- |
| Название варианта | Регистрация бегуна |
| Цель | Зарегистрироваться в системе |
| Действующие лица (актеры) | Бегун |
| Краткое описание | Пользователь регистрируется в качестве бегуна |
| Тип варианта | Основной |

1. Вариант использования «Редактирование профиля».

|  |  |
| --- | --- |
| Название варианта | Редактирование профиля |
| Цель | Возможность редактирование профиля бегуна |
| Действующие лица (актеры) | Бегун, Координатор |
| Краткое описание | Бегун, или координатор может изменить данные о конкретном пользователе |
| Тип варианта | Вспомогательный |

1. Вариант использования «Регистрация бегуна на марафон»

|  |  |
| --- | --- |
| Название варианта | Регистрация бегуна на марафон |
| Цель | Зарегистрироваться на марафон |
| Действующие лица (актеры) | Бегун |
| Краткое описание | Бегун регистрируется на марафон |
| Тип варианта | Вспомогательный |

1. Вариант использования «Просмотр результатов бегуна»

|  |  |
| --- | --- |
| Название варианта | Просмотр результатов бегуна |
| Цель | Просмотреть результат |
| Действующие лица (актеры) | Бегун |
| Краткое описание | Бегун просматривает результаты своих гонок |
| Тип варианта | Вспомогательный |

1. Вариант использования «Просмотр бегунов»

|  |  |
| --- | --- |
| Название варианта | Просмотр бегунов |
| Цель | Просмотреть бегунов |
| Действующие лица (актеры) | Координатор |
| Краткое описание | Координатор просматривает бегунов |
| Тип варианта | Вспомогательный |

1. Вариант использования «Печать сертификата об участии»

|  |  |
| --- | --- |
| Название варианта | Печать сертификата об участии |
| Цель | Возможность печати сертификата об участии |
| Действующие лица (актеры) | Координатор |
| Краткое описание | Позволяет координатору распечатать бегуну сертификат участия в предыдущих соревнованиях |
| Тип варианта | Вспомогательный |

1. Вариант использования «Импорт данных»

|  |  |
| --- | --- |
| Название варианта | Импорт данных |
| Цель | Возможность импортирования данных |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору импортировать информацию в справочники |
| Тип варианта | Вспомогательный |

1. Вариант использования «Просмотр благотворительных организации»

|  |  |
| --- | --- |
| Название варианта | Просмотр благотворительных организации |
| Цель | Возможность просматривания списка благотворительных организаций |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору просматривать список благотворительных организаций |
| Тип варианта | Вспомогательный |

1. Вариант использования «Добавление/изменение благотворительных организации»

|  |  |
| --- | --- |
| Название варианта | Добавление/изменение благотворительных организации |
| Цель | Возможность добавлять/изменять информацию о благотворительных организациях |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору добавлять или изменять имеющуюся информацию |
| Тип варианта | Вспомогательный |

1. Вариант использования «Просмотр пользователей»

|  |  |
| --- | --- |
| Название варианта | Просмотр пользователей |
| Цель | Возможность просматривать список всех пользователей системы |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору просматривать список всех пользователей системы |
| Тип варианта | Вспомогательный |

1. Вариант использования «Редактирование пользователя»

|  |  |
| --- | --- |
| Название варианта | Редактирование пользователя |
| Цель | Возможность редактирования пользователя |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору изменять имеющуюся информацию о пользователе |
| Тип варианта | Вспомогательный |

1. Вариант использования «Добавление нового пользователя»

|  |  |
| --- | --- |
| Название варианта | Добавление нового пользователя |
| Цель | Возможность добавление нового пользователя |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору добавлять нового пользователя |
| Тип варианта | Вспомогательный |

1. Вариант использования «Просмотр волонтеров»

|  |  |
| --- | --- |
| Название варианта | Просмотр волонтеров |
| Цель | Возможность просматривать список волонтеров |
| Действующие лица (актеры) | Администратор |
| Краткое описание | Позволяет администратору просматривать список волонтеров |
| Тип варианта | Вспомогательный |

Диаграмма прецедентов для информационной системы приведен на рисунке 1

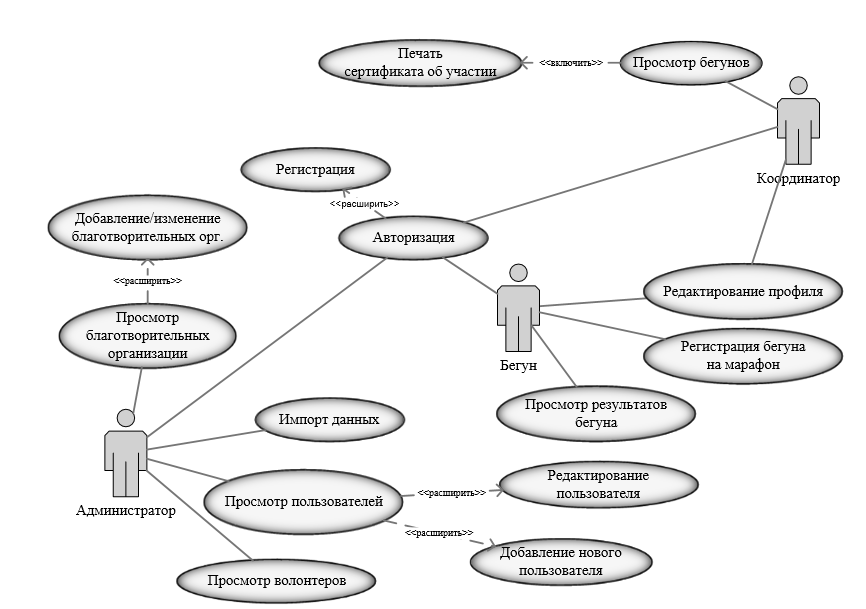


Рисунок 1 – Диаграмма прецедентов

Диаграмма прецедентов показывает отношения между актёрами и прецедентами на информационной системе.

Варианты использования позволяют избегать создания предположений о том, как функциональные возможности будут реализованы.

При выполнении определенных функций приложения могут возникнуть исключительные ситуации.

1. Исключительные ситуации

|  |  |  |
| --- | --- | --- |
| Функция | Случай нарушения работы функции | Реакция |
| Регистрация | При создании профиля допущены недостающие данные. | Выскакивает модальное окно с ошибкой о недостающих данных |
| Авторизация | Данные введены неверно или отсутствуют в системе | Выскакивает модальное окно с ошибкой о неверно введенных данных |

2 Практическая часть

2.1 Разработка CRC – карточек

CRC-карты используются для представления: объектов, их ответственности и взаимодействия с другими объектами. Они создаются в процессе исследования сценария работы программы, основанной на моделировании поведения системы в целом.

Для работы информационной системы необходимо выделить следующие классы:

1. Бегун.
2. Координатор.
3. Администратор.
4. CRC-карточка бегуна

|  |  |
| --- | --- |
| Бегун | |
| Регистрируется/авторизуется | Система |
| Редактирует профиль |
| Регистрируется на марафон |
| Просматривает свои результаты |

1. CRC-карточка координатора

|  |  |
| --- | --- |
| Координатор | |
| Авторизуется в системе | Система  Бегун |
| Просматривает бегунов |
| Редактирует профиль |
| Печатает сертификаты бегунам |

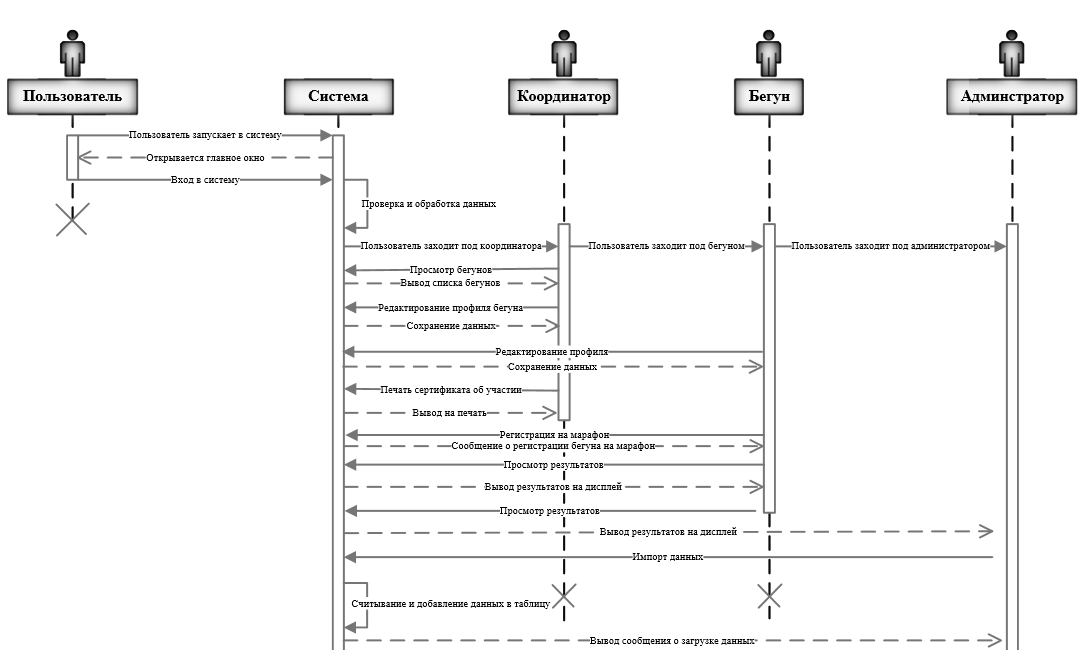
1. CRC-карточка администратора

|  |  |
| --- | --- |
| Администратор | |
| Авторизуется в системе | Система  Бегун  Координатор |
| Импортирует данные в бд |

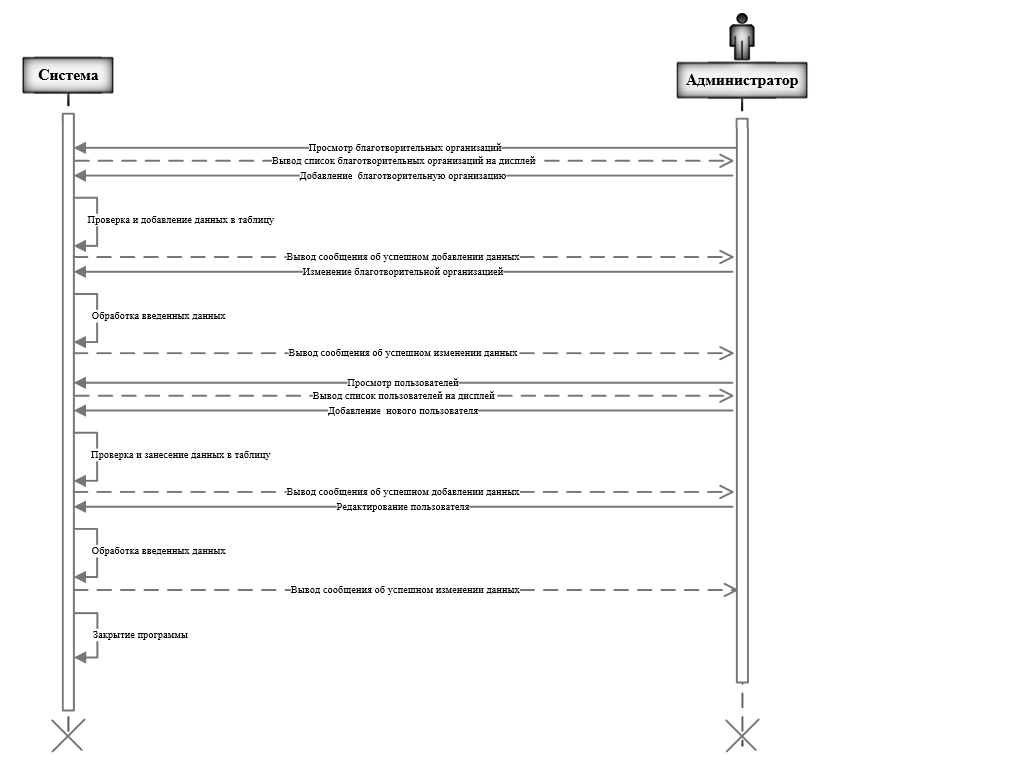
Продолжение таблицы 19

|  |
| --- |
| Администратор |

|  |  |
| --- | --- |
| Просматривает список благотворительных организации |  |
| Добавляет новую благотворительную организацию |
| Изменяет существующую благотворительную организацию |
| Просматривает список пользователей |
| Добавляет нового пользователя |
| Редактирует профиль пользователя |
| Просматривает список волонтеров |



Продолжение рисунка 2



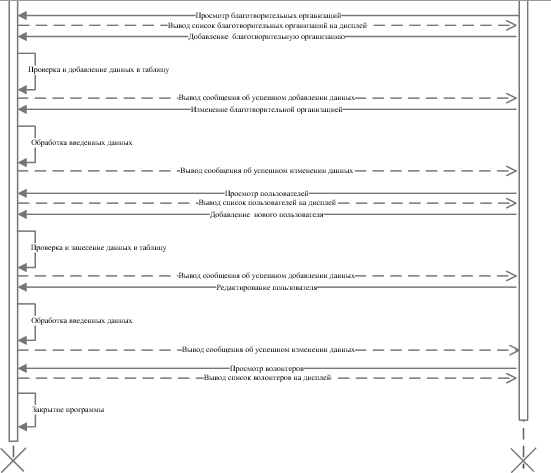


Рисунок 2 – Диаграмма последовательности

На диаграмме последовательности изображен жизненный цикл информационной системы. Она показывает проверку достаточности или избыточности придуманных классов информационной системы, а также корректности их взаимодействия.

2.2 Структура и функциональная схема

Для проектирования информационной системы была разработана структурная схема, которая служит для отображения элементарных частей объекта, реализующие элементарную функцию системы и связей между ними.

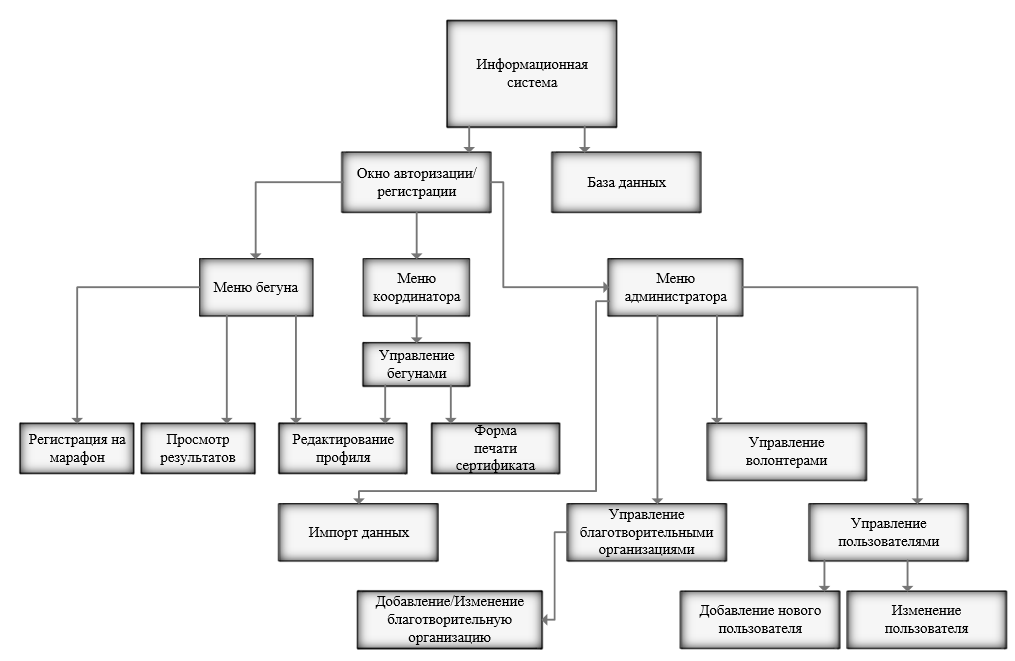


Рисунок 3 – Структурная диаграмма

После построении структурной схемы, необходимо построить функциональную схему. Она используется, чтобы разбить всю программу на основные функции выполняемые ей. Для каждой функции выделяются методы обработки и получаемый результат, что помогает при дальнейшем проектировании системы.

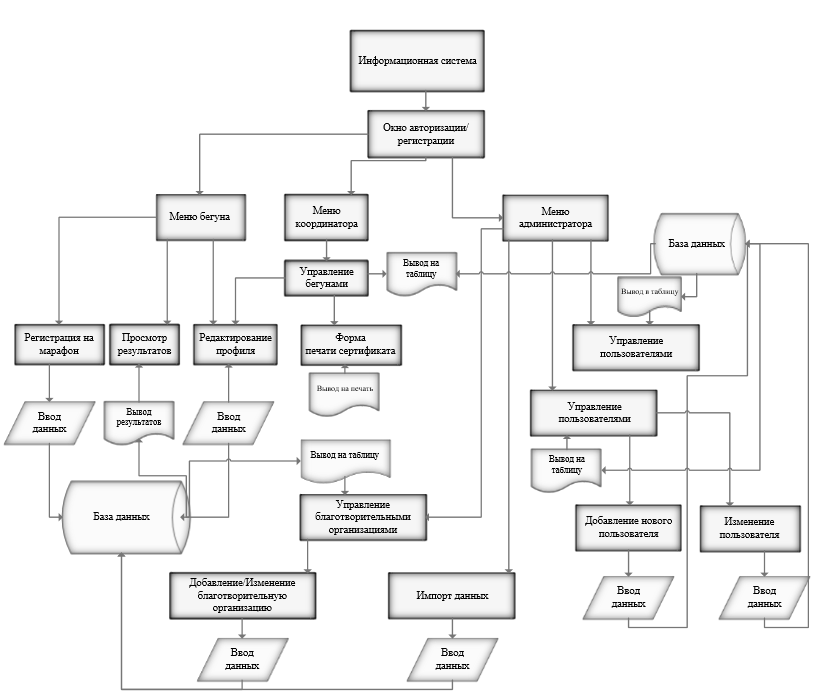


Рисунок 4 – Функциональная диаграмма

На функциональной схеме приведены основные функции. Схема наглядно отражает устройство функциональных изменений системы.

2.3 Диаграмма деятельности

Диаграмма деятельности, описывает часть системы, представляя ее различные состояния и действия, которые приводят к их изменению. Диаграмма хорошо подходит для визуального описания сложного поведения.

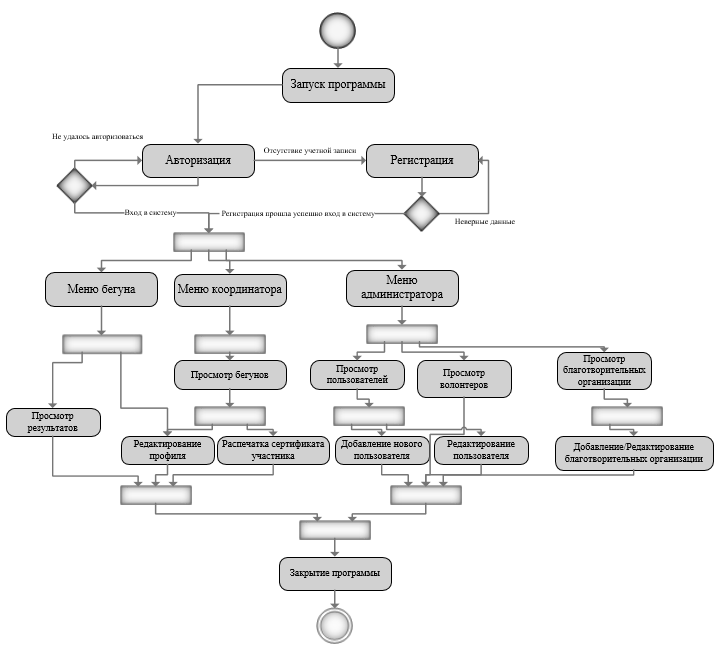


Рисунок 5 – Диаграмма деятельности

На диаграмме показано, что при входе в систему пользователь должен совершить авторизацию, если у пользователя отсутствует учетная запись, то он сначала регистрируется, а после этого он входит в систему. После этого, пользователю предоставляется доступ к функционалу информационной системы.

2.4 ER-диаграмма

Диаграмма сущность-связь (ER) используются для разработки данных и представляют собой стандартный способ определения данных и отношений между ними. Таким образом, осуществляется детализация хранилищ данных. ER-диаграмма содержит информацию о сущностях системы и способах их взаимодействия, включает идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их отношений с другими объектами (связей). Во многих случаях информационная модель очень сложна и содержит множество объектов.

Информационная система имеет следующие сущности:

1. Роль.
2. Пользователи.
3. Бегуны.
4. Страны.
5. РезультатыСоревнований.
6. Марафон.
7. СоревнованиеМарафона.
8. ГоночныйКомплект.
9. РегистрацияНаМарафон.
10. БлаготворительнаяОрганизация.
11. Спонсоры.
12. Волонтеры.
13. Сущность «Роль»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Наименование | Строка |

1. Сущность «Пользователи»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Email | Строка |
|  | Пароль | Строка |
|  | Имя | Строка |
|  | Фамилия | Строка |
| FK | Роль | Строка |

1. Сущность «Бегуны»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
| FK | Email | Пользователи |
|  | Пол | Перечисление |
|  | Дата рождения | Дата |
| FK | Страна | Строка |

1. Сущность «Страны»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Название | Строка |
|  | КодСтраны | Строка |

1. Сущность «Регистрация на марафон»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
| FK | Бегун | Строка |
| FK | Гоночный комплект | Строка |
|  | Регистрационный статус | Перечисление |
|  | Стоимость | Числовой |
| FK | Благотворительная организация | Строка |

1. Сущность «Гоночный комплект»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Комплект | Строка |
|  | Стоимость | Числовой |

1. Сущность «Результаты соревнований»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
| FK | Соревнование марафона | Строка |
| FK | Бегун | Бегун |
|  | Время прохождения | Дата время |

1. Сущность «Соревнование марафона»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Наименование | Строка |
|  | Дистанция | Перечисление |
|  | Дата начала | Дата время |
| FK | Марафон | Строка |
|  | Стоимость участия | Числовой |
|  | Максимум участников | Числовой |

1. Сущность «Марафон»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Наименование | Строка |
|  | Город | Строка |
| FK | Страна | Страны |
|  | Год проведения | Дата |

1. Сущность «Спонсоры»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Имя | Строка |
| FK | РегистрацияНаМарафон | Строка |
|  | Количество | Страны |

1. Сущность «Волонтеры»

|  |  |  |
| --- | --- | --- |
| Назначение поля | Название поля | Тип данных |
| PK | Код | Счетчик |
|  | Имя | Строка |
|  | Фамилия | Строка |
|  | Пол | Строка |
| FK | Страна | Строка |

ER- диаграмма представлена на рисунке 6

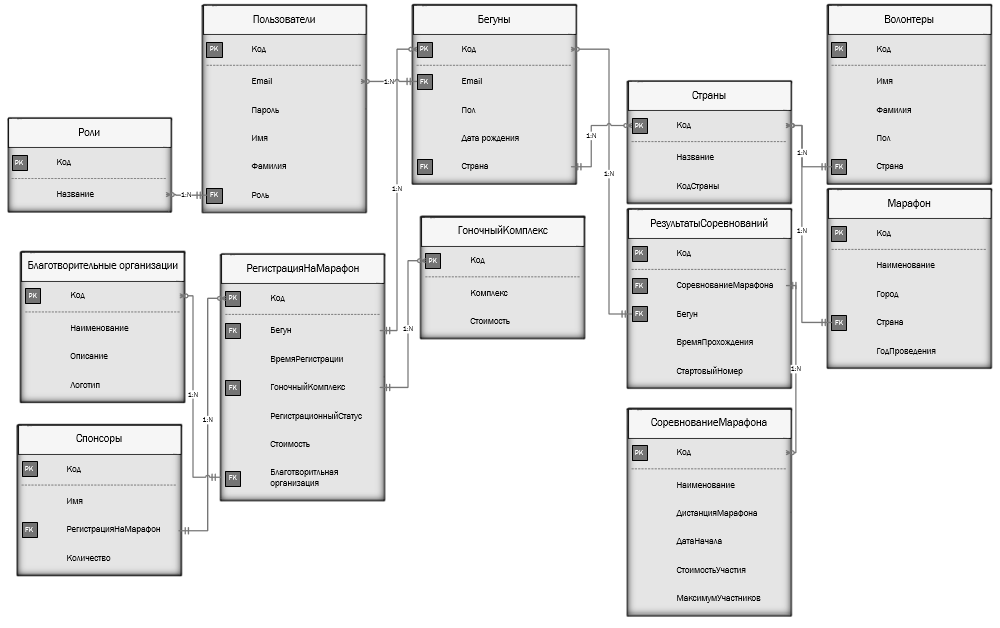


Рисунок 6 – Диаграмма «Сущность-связь»

На диаграмме изображены сущности, в которых хранятся данные в виде таблицы в базе данных. Таблицы взаимосвязаны между собой и при этом не нарушается целостность всей информационной системы.

2.5 Диаграмма классов

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Для построения диаграммы необходимо выделить следующие классы:

1 Бегун.

2 Координатор.

3 Администратор.

4 Система.

Диаграмма классов представлена на рисунке 7

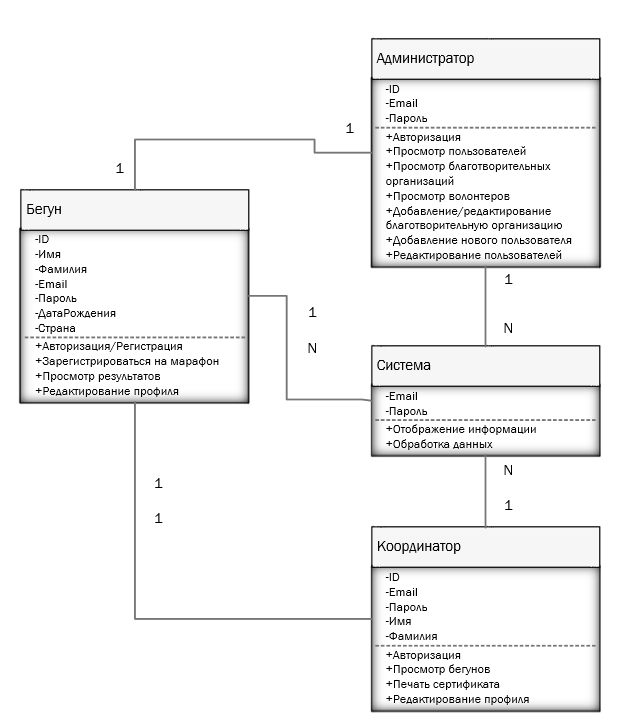


Рисунок 7 – Диаграмма классов

Диаграмма показывает классы информационной системы, их атрибуты, методы и взаимосвязи между ними.

2.6 Тестирование программного обеспечения

Тестирование программного обеспечения заключается в выполнении приложения на некотором множестве исходных данных и сверке получаемых результатов с заранее известными, с целью установить соответствие различных свойств и характеристик приложения заказанным свойствам.

Тестирование методом черного ящика не требует каких-либо знаний о внутренней работе приложения. Рассматриваются только основные аспекты системы, не связанные или мало связанные с ее внутренней логической структурой.

Регистрация представляет собой процесс сообщения к системе своих данных для получения доступа к функционалу, которые не могут просматривать или загружать неавторизированные пользователи. Тестирование функции регистрации приведено в таблице 31

1. Тестирование функции регистрации

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Email | Пароль | Подтверждение пароля | Имя | Фамилия | Пол | Дата рождения | Ожидаемый результат | Что проверяется |
| 1 |  |  | - |  |  |  |  | Вывод сообщение о не ведённых данных | Не заполненные поля |
| 2 | vlad | vlad | vlad | Влад | Маркелов | М | 18.07.1999 | Вывод сообщение о неверном вводе email | Правильность ввода email |
| 3 | vlad@mail.ru | vlad | vlad1 | Влад | Маркелов | М | 18.07.1999 | Вывод сообщения «Пароли не совпадают» | Правильность ввода паролей |
| 4 | vlad@mail.ru | vlad | vlad | Влад | Маркелов | М | 18.07.1999 | Вывод сообщения об успешной регистрации | Успешная регистрация |

Тест № 1. Не полный ввод данных приведен на рисунке 8

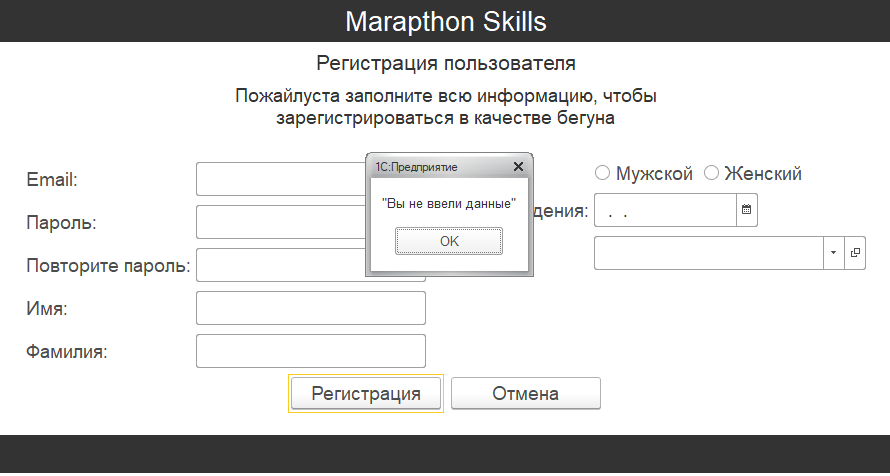


Рисунок 8 – Незаполненные поля

Тест № 2. Правильность ввода email приведен на рисунке 9

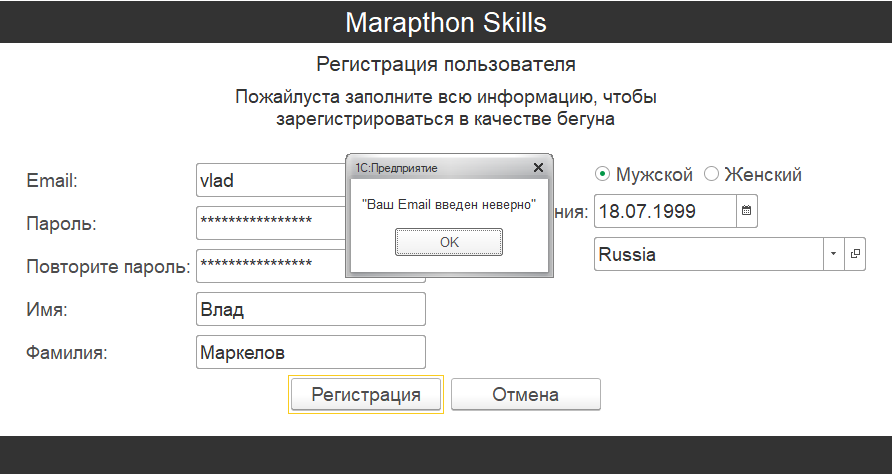


Рисунок 9 – Проверка почты на валидность

Тест № 3. Подтверждение паролей приведен на рисунке 10

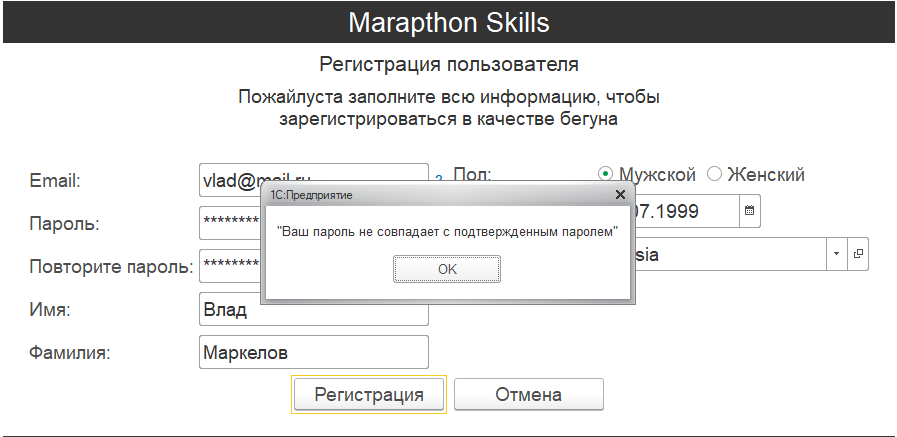


Рисунок 10 – Не совпадение паролей

Тест № 4. Успешный ввод данных приведен на рисунке 11

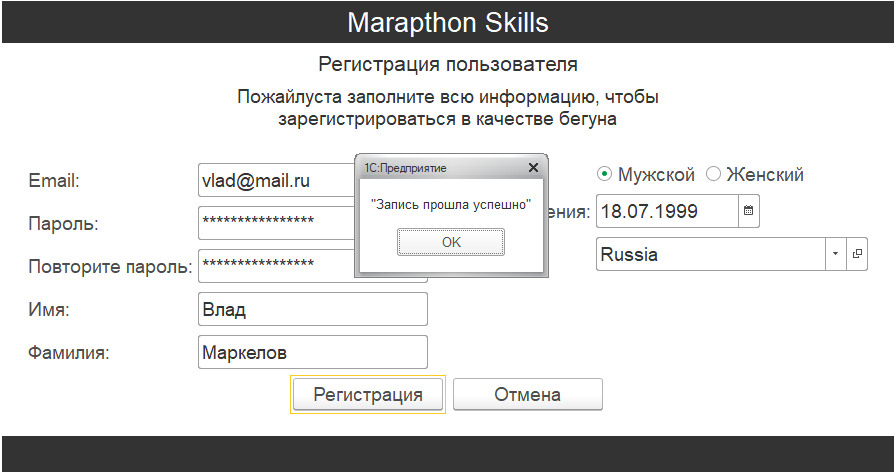


Рисунок 11 – Успешная регистрация

Авторизация - это функция, которая предусматривает ввод своего уникального логина и пароля, которые были указаны при регистрации, авторизация открывает доступ к полному функционалу сайта. Тестирование функции авторизации приведено в таблице 32

1. Тестирование функции авторизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Email | Пароль | Ожидаемый результат | Что проверяется |
| 1 |  |  | Вывод сообщения о незаполненных полей | Не полный ввод данных |
| 2 | vlad@mail.ru | vlad1 | Вывод сообщения о неверном введённом пароле или email | Неверный ввод пароля |
| 3 | vlad1@mail.ru | vlad | Вывод сообщения о неверном введённом пароле или email | Неверный ввод email |
| 4 | vlad@mail.ru | vlad | Переход на форму «Меню бегуна» | Вход систему под ролью бегуна |
| 5 | internet@mail.ru | internet | Переход на форму «Меню координатора» | Вход систему под ролью координатора |
| 6 | admin@mail.ru | admin | Переход на форму «Меню администратора» | Вход систему под ролью администратора |

Тест № 1. Не полный ввод данных приведен на рисунке 12

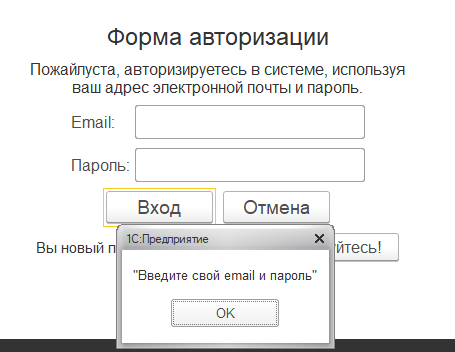


Рисунок 12 – Сообщение об отсутствии данных

Тест № 2. Авторизация в неверном вводе пароля приведен на рисунке 13

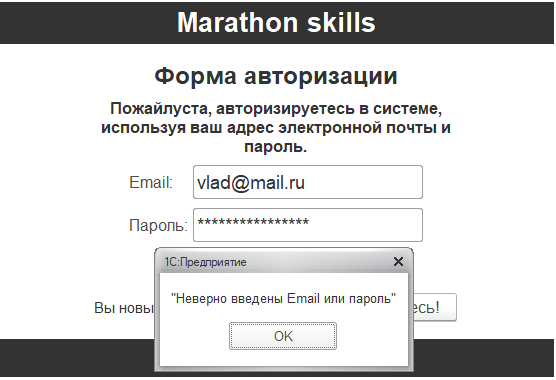


Рисунок 13 – Неверный ввод пароля

Тест № 3. Авторизация в неверном вводе email приведен на рисунке 14

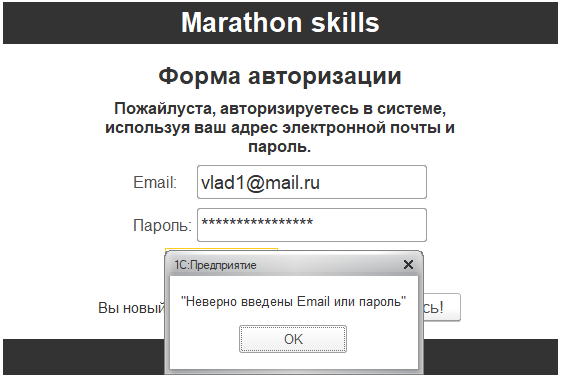
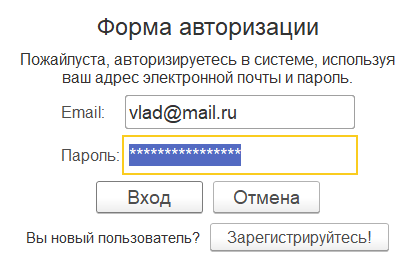


Рисунок 14 – Неверный ввод email

Тест № 4. Авторизация в системе под бегуном приведен на рисунке 15



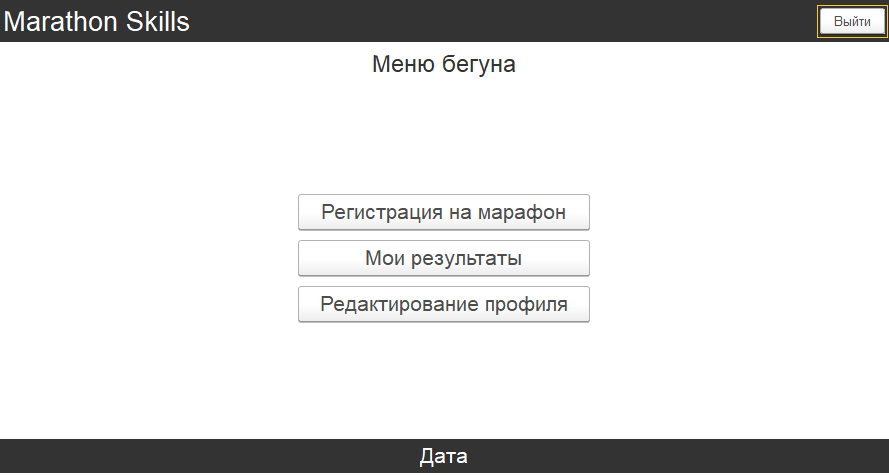
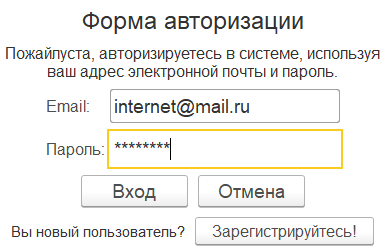


Рисунок 15 – Вход в систему под бегуном

Тест № 5. Авторизация в системе под координатором приведен на рисунке 16



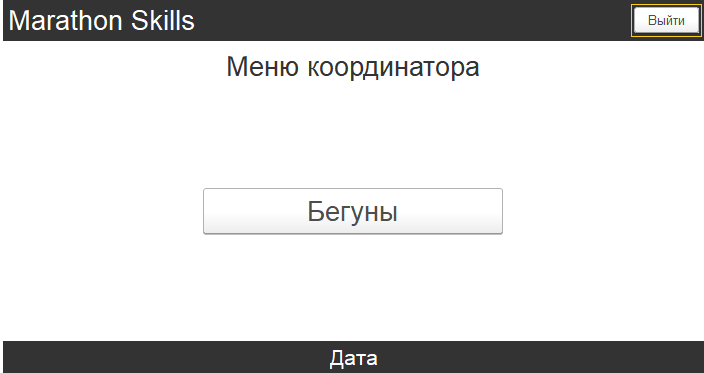
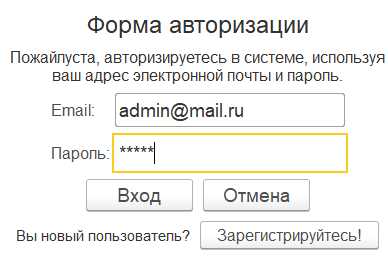


Рисунок 16 – Вход в систему под координатором

Тест № 6. Авторизация в системе под администратором приведен на рисунке 17



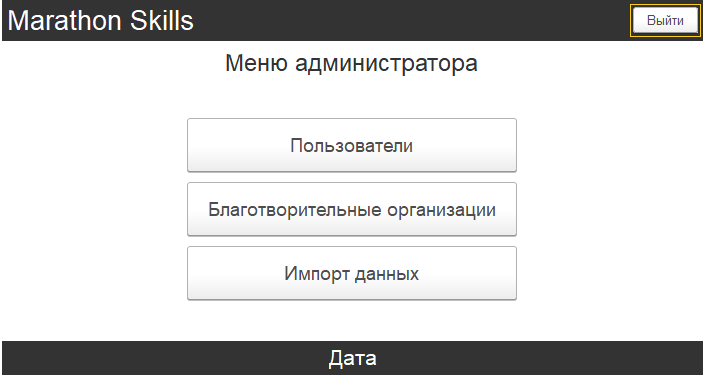


Рисунок 17 – Вход в систему под администратором

Редактирование профиля – возможность внесения изменений в свой аккаунт.

Тестирование функции редактирование профиля приведено в таблице 33

1. Тестирование функции редактирование профиля

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Имя | Фамилия | Пол | Дата рождения | Пароль | Подтверждение пароля | Ожидаемый результат | Что проверяется |
| 1 |  |  |  |  |  |  | Вывод сообщение что данные не изменены | Не заполненные поля |
| 2 | Владислав |  |  |  |  |  | Вывод сообщение что данные изменены | Сохранения имени |
| 3 |  | Новиков |  |  |  |  | Вывод сообщение что данные изменены | Сохранения фамилии |
| 4 |  |  |  |  | vladislav | vlad | Вывод сообщения что пароли не совпадают и данные не удалось обновить | Совпадение паролей |
| 5 |  |  |  |  | vladislav | vladislav | Вывод сообщение что данные изменены | Сохранение пароля |

Тест № 1. Не полный ввод данных приведен на рисунке 18

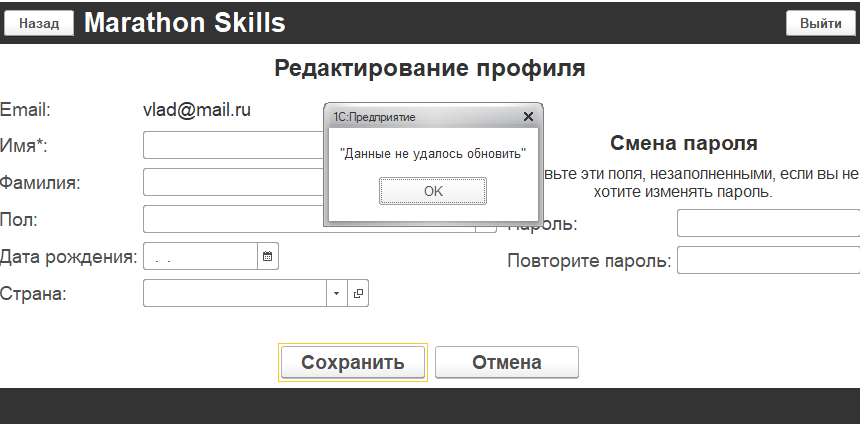


Рисунок 18 – Вывод сообщения «Данные не удалось обновить»

Тест № 2. Изменение имени приведен на рисунке 19

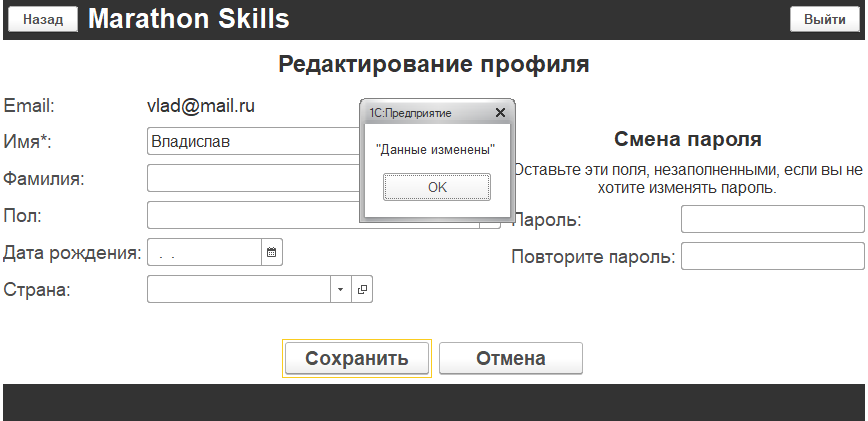


Рисунок 19 – Изменения имени

Тест № 3. Изменение фамилии приведен на рисунке 20

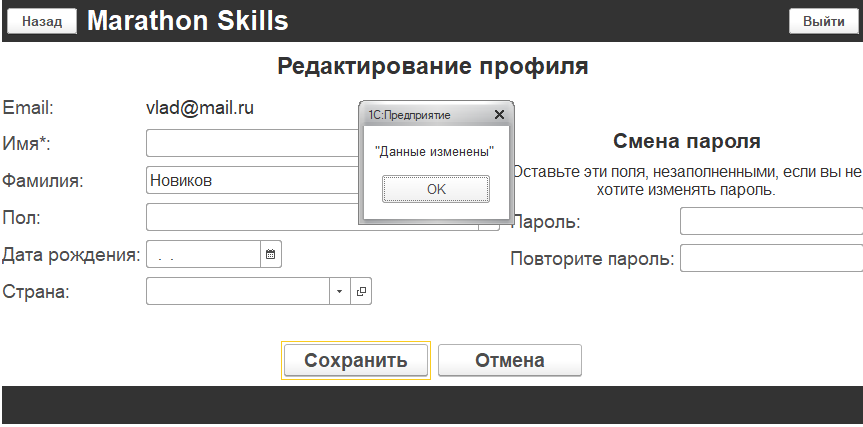
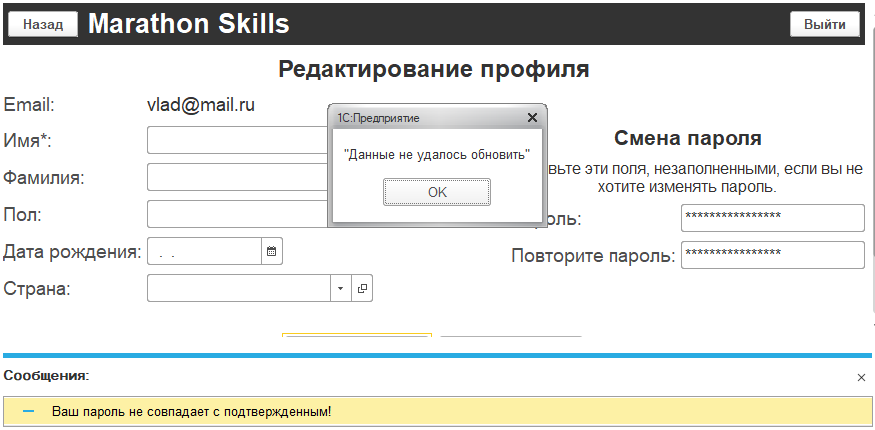


Рисунок 20 – Изменения фамилии

Тест № 4. Совпадение паролей приведен на рисунке 21



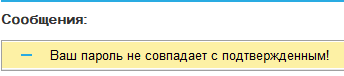


Рисунок 21 – Сообщение о несовпадении паролей

Тест № 5. Сохранение пароля приведен на рисунке 22

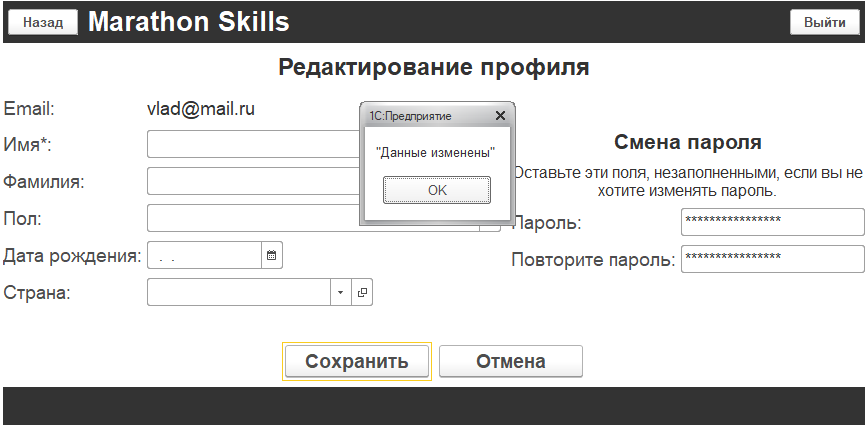


Рисунок 22 – Сохранения пароля

Тестированием методом белого ящика направлены на локализацию ошибок, которые сложнее выявить, найти и зафиксировать. С их помощью можно обнаружить логические ошибки и проверить степень покрытия тестами. Тестирование методом белого ящика приведено в таблице 34

1. Тестирование методом «белого ящика»

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| Замена конструкции:    На: | При незаполненных полей выскакивает сообщение об ошибке | Вывод сообщения | При незаполненных полей пользователь нажимает кнопку «Войти»,  выводится сообщение |
| Замена конструкции:    На: | При вводе пользовательских данных, входит в систему | Переход на форму | При вводе email и пароль, пользователь входит в систему |

Тест №1 изменения логики авторизации приведен на рисунке 23

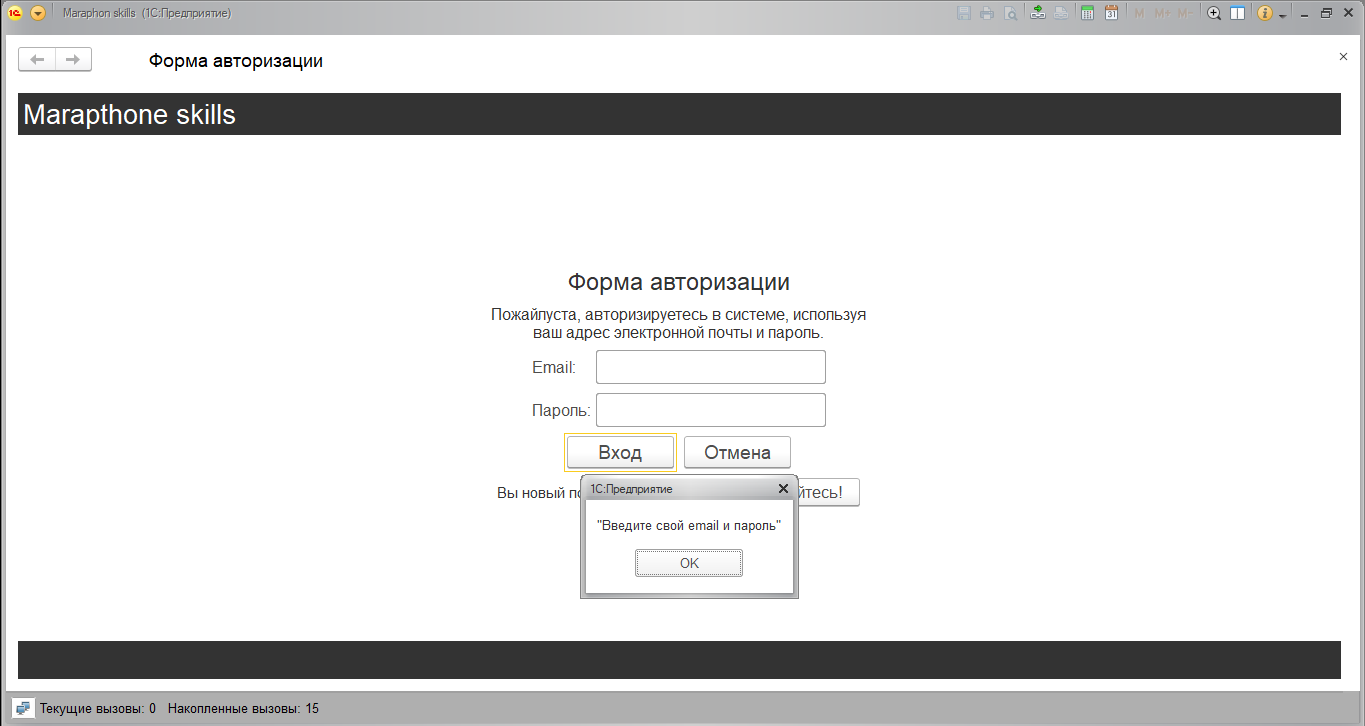


Рисунок 23 – Вывод сообщения о незаполненных полях

Тест №2 изменения логики авторизации приведен на рисунке 24

|  |  |
| --- | --- |
|  |  |

Рисунок 24 – При успешной авторизации осуществляется вход в систему

Заключение

Целью работы являлось спроектировать информационную систему спортивных мероприятий, разработать часть функционала и протестировать его.

В курсовой работе были успешно реализованы все поставленные задачи, и в заключении будут разобраны все решения по задачам курсовой работы, а также порядок их выполнения.

Изучена предметная область в сфере спортивных мероприятий, для которой разрабатывается информационная система, что позволило определить основные функции программы, действующих ролей, а также определить наилучшие варианты для разработки интерфейса.

Для разработки информационной системы была выбрана среда разработки - 1С Предприятие. 1С Предприятие - это система программ, предназначенная для автоматизации различных областей экономической деятельности предприятий, организаций и учреждений, независимо от их вида деятельности и формы собственности, с различным уровнем сложности учета. У среды разработки имеется встроенный язык 1С – язык программирования, который используется в семействе программ «1С Предприятие». Язык является предварительно компилируемым предметно-ориентированным языком высокого уровня.

Разработана система требований, которая позволила эффективно разрабатывать информационную систему спортивных мероприятий.

Проанализированы требования к функциональной спецификации, а также были описаны варианты использования системы:

* Регистрация/авторизация пользователя.
* Редактирование профиля.
* Просмотр пользователей.

Разработана диаграмма прецедентов, показывающая отношения между актёрами и прецедентами на информационной системе. Бегун регистрируется и участвует в соревновании, координатор управляет с бегунами, администратор работает с базой данных.

Определены исключительные ситуации и реакции на эти ситуации и все существенные случаи, когда приложение не сможет нормально выполнить ту или иную свою функцию. При проверке введенных данных при регистрации, появляется сообщение об ошибке. При вводе неверных данных в авторизации, появляется о неправильном вводе логина или пароля.

Разработаны CRC карточки с целью исследования взаимодействия объектов системы, выявлены классы, которые необходимо создать в программе, так же выделены следующие классы:

* Администратор.
* Бегун.
* Координатор.

После выделения основных классов, спроектирована диаграмма последовательности, показывающая проверку достаточности или избыточности придуманных классов информационной системы, а также корректности их взаимодействия. На диаграмме показано как пользователь запускает приложение, авторизуется под своей учетной записью и входит под своей ролью. Тем самым система, проверяя пользовательские данные, отправляет запрос на переход на соответствующее меню.

Реализована структурная схема, отображающая элементарные части объекта, системы управления, реализующие элементарную функцию системы и связей между ними. Информационная состоит из следующих частей таких как:

1. Интерфейсная часть.
2. База данных.

Также разработана функциональная схема, наглядно отражающая устройство функциональных изменений системы.

Разработанная диаграмма деятельности, описывает часть системы, представляя ее различные состояния и действия, которые приводят к их изменению. Диаграмма хорошо подходит для визуального описания сложного поведения.

Реализована ER-диаграмма, в которой изображены сущности, в которых хранятся данные в виде таблицы в базе данных. Таблицы взаимосвязаны между собой и при этом не нарушается целостность всей информационной системы.

Для разработанной ER – диаграммы выделены следующие сущности:

1. Роль.
2. Пользователи.
3. Бегуны.
4. Страны.
5. РезультатыСоревнований.
6. Марафон.
7. СоревнованиеМарафона.
8. ГоночныйКомплект.
9. РегистрацияНаМарафон.
10. БлаготворительнаяОрганизация.
11. Спонсоры.
12. Волонтеры.

На диаграмме показано, что у администратора имеется полный доступ к базе данных; бегуны участвую и просматривают свои результаты; координатор может просматривать и сопровождать бегунов.

Диаграмма классов, показывающая классы информационной системы, их атрибуты, методы и взаимосвязи между ними. Для разработанной диаграммы классов выделяются следующие классы:

1. Бегун.
2. Координатор.
3. Администратор.
4. Система.

Тестирование программного обеспечения заключается в выполнении приложения на некотором множестве исходных данных и сверке получаемых результатов с заранее известными, с целью установить соответствие различных свойств и характеристик приложения заказанным свойствам.

Проведенное тестирование методом «черного ящика» рассматриваются только основные аспекты системы, не связанные или мало связанные с ее внутренней логической структурой.

Для тестирования были выделены две функции:

* Регистрация/авторизация.
* Редактирование профиля.

Тестирование методом «белого ящика» позволяет обнаружить логические ошибки и проверить степень покрытия тестами.

В тестировании использовалось:

1. При заполнении пустых полей выскакивало сообщение об ошибке.
2. Проверка введенных данных для правильности входа в систему.

Разработано руководство пользователя содержащее алгоритм работы с информационной системой, технические характеристики и всевозможные ошибки.

Разработана техническая спецификация, которая определяет набор требований, которым должна соответствовать обучающая информационная система.

Разработано техническое задание, которое регламентирует назначение разработки, требования к программному продукту и сопроводительную документацию, а также учитываются все стадии разработки, и порядок контроля и приемки информационной системы.

Разработано руководство по техническому обслуживанию, объясняющее персоналу о правильном применении и техническом обслуживании программы. Руководство по техническому обслуживанию разрабатывается для пользователей программы, а также специалистов, которые занимаются обслуживанием программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. Ананьева, Т.Н. Стандартизация, сертификация и управление качеством программного обеспечения [Электронный ресурс] / Ананьева Т.Н. – М.: ИНФРА-М, 2017.- (ЭБС Знаниум)
2. Ажеронок В.А. Разработка управляемого интерфейса [Электронный ресурс] / Ажеронок В.А. - М.: НИЦ Инфра-М, 2017.- (ЭБС Знаниум)
3. Байдаков, В.А. 1С:Предприятие Руководство разработчика прикладных решений[Электронный ресурс] /Байдаков В.А. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2017. - (ЭБС Знаниум)
4. Буч, Г.К. Объектно-ориентированное программирование [Текст] / Буч Г.К. - СПб.: Питер, 1992. – 517 с.
5. Гагарина, Л.Г. Разработка и эксплуатация автоматизированных информационных систем [Электронный ресурс] / Гагарина Л.Г.-М.: ФОРУМ:ИНФРА-М,2017- (ЭБС Знаниум)
6. Гагарина, Л.Г. Технология разработки программного обеспечения [Электронный ресурс] /Гагарина Л.Г. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2017. - (ЭБС Знаниум)
7. Кашаев, С.М. Программирование в 1С:Предприятие [Электронный ресурс] / Кашаев С.М. - М.: НИЦ Инфра-М, 2017.- (ЭБС Знаниум)
8. Радченко, М.Г. 1С:Предприятие Практическое пособие разработчика [Электронный ресурс] / Радченко М.Г.-М.: ФОРУМ:ИНФРА-М,2017- (ЭБС Знаниум)
9. Маклафлин, Б.Н. Объектно-ориентированный анализ и проектирование [Текст] / Маклафлин Б.Н. - СПб.: Питер, 2017. – 680 с.
10. Канакова, С.Г. Методические указания по выполнению курсовой работы по ПМ 03 Участие в интеграции программных модулей МДК 03.01 Технология разработки программного обеспечения [Текст]: Методические указания / С.Г. Канакова. - Прокопьевск, 2018. – 21с.

Приложение А

ГКПОУ ПГТТ им. В.П. Романова

ИНФОРМАЦИОННАЯСИСТЕМА

**«**MARATHON SKILLS**»**

Руководство пользователя

Версия 0.5

Маркелов В. А.

Прокопьевск, 2018

1 Общие положения

Информационная система «Marathon skills» является средством автоматизации работников на спортивном мероприятии.

Данный программный продукт выполняет следующие функции:

1. Просмотр информации;
2. Авторизация/регистрация пользователя в системе;
3. Регистрация бегуна на марафон;
4. Редактирование профиля пользователя;
5. Импорт данных.

2 Описание установки

Первый этап к началу работы с информационной системы, это ее установка на персональный компьютер пользователя.

Для установки информационной системы нужно сделать следующие шаги:

1. Запуск файла «Setup.exe».
2. Выбор расположения ИС.
3. Ожидание установки.
4. Нажать кнопку «Готово».

В таблице 1 представлены системные требования к программному продукту.

1. Системные требования к программному продукту

|  |  |
| --- | --- |
| Минимальные: | Рекомендованные: |
| ОС: Windows XP, Vista, 7, 8.1, или 10 | ОС: Windows 7, 8.1, 10 |
| Процессор: 1,3 ГГЦ | Процессор: 2 ГГЦ |
| Оперативная память: 1 ГБ | Оперативная память: 2 ГБ |
| Видеокарта: 128 МБ | Видеокарта: 512 МБ |
| Место на жестком диске: 200 МБ | Место на жестком диске: 200 МБ |

3 Описание запуска

Перед запуском программы необходимо удостовериться в наличии интернет соединения.

После выполнения всех перечисленных условий можно приступать к запуску приложения.

Открыть на рабочем столе ярлык «MarathonSkills.exe» и подождать несколько минут, когда программа выполнит запуск.

При запуске программы пользователю открывается окно авторизации, в котором можно ввести свои пользовательские данные для входа в систему.

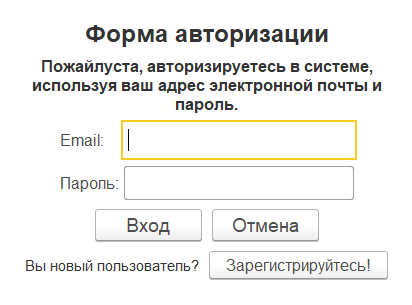


Рисунок 1 – Форма авторизации

Если пользователь не имеет учетной записи, есть возможность зарегистрироваться, после ввода всех имеющихся данных, пользователь входит в систему под ролью бегуна.

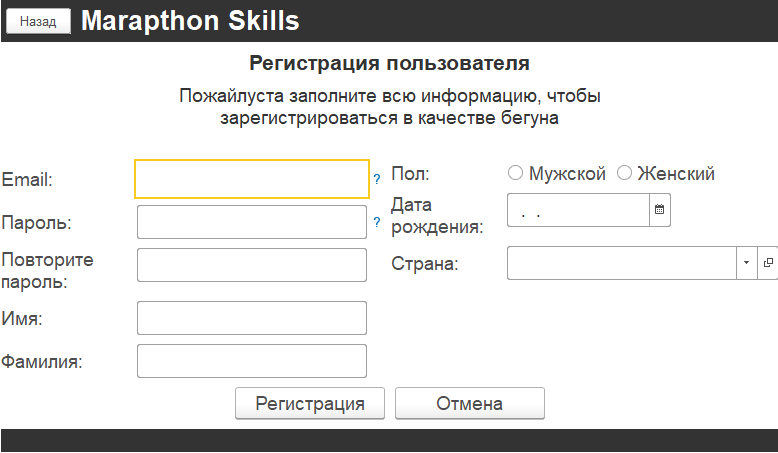


Рисунок 2 – Форма регистрации

В меню бегуна, пользователь имеет возможность:

1. Регистрироваться на марафон.
2. Редактировать профиль.
3. Просматривать свои результаты.
4. Выйти из профиля.

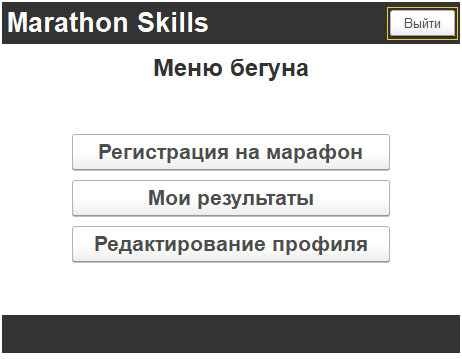


Рисунок 3 – Меню бегуна

При авторизации под ролью координатора открывается меню. При переходе на форму «Управление бегунами» можно просматривать, редактировать данные бегуна.

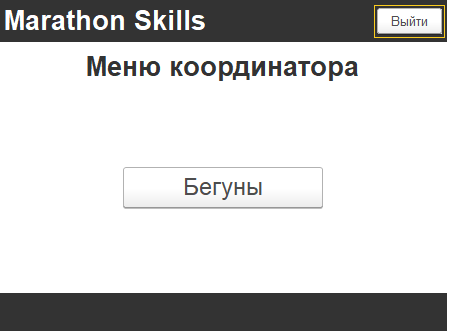


Рисунок 4 – Меню координатора

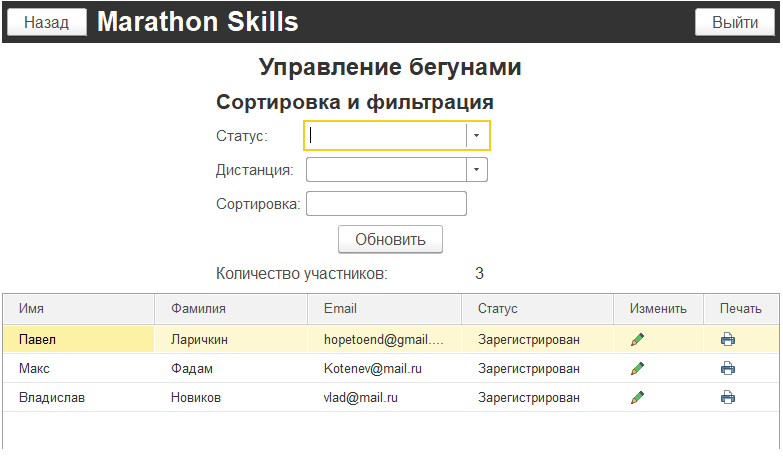


Рисунок 5 – Форма управление бегунами

При входе под учетной записью администратора, пользователю предоставляет следующие возможности:

1. Просматривать пользователей, а также добавлять и изменять его профиль;
2. Просматривать список благотворительных организаций, включающее добавление/изменение информации.
3. Просматривать волонтеров.
4. Импорт данных.

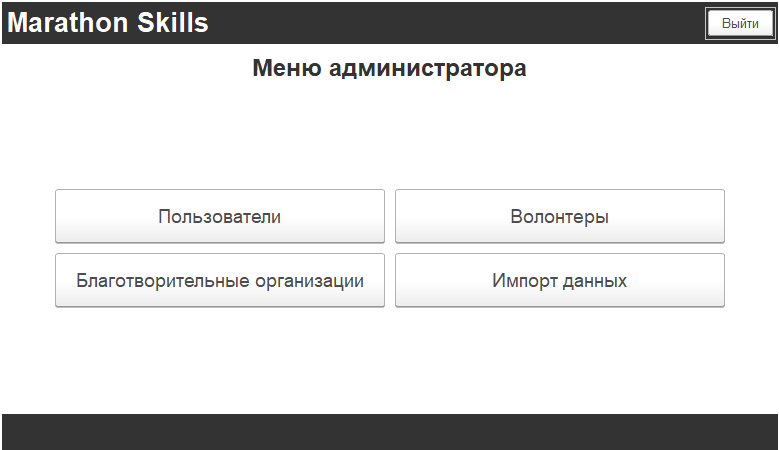


Рисунок 6 – Меню администратора

4 Сообщения пользователю

На таблице 2 приведены системные сообщения пользователю при совершении различных действий.

1. Сообщение пользователю

|  |  |  |
| --- | --- | --- |
| Текст сообщения | Описание сообщения | Действия |
|  | Сообщение появляется при неправильном набранном логине или пароле | Ввести правильные данные своей учетной записи |
|  | Сообщение появляется при отсутствии данных в полях ввода | Ввести данные учетной записи |
|  | Сообщение появляется при незаполненных необходимых полей | Ввести недостающие данные |
|  | Сообщение появляется когда пользователь при редактировании профиля оставил поля пустыми | Ввести необходимые данные для изменения |
|  | Сообщение появляется когда пользователь изменил свои данные | Продолжить работу в системе |

Приложение Б – Спецификация

1. Спецификация

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
| 01.02.01.ПЗ | Пояснительная записка |  |
| 01.02.02.РП | Руководство пользователя |  |
| 01.02.03.ТЗ | Техническое задание |  |
| 01.02.04.РТО | Руководство по техническому обслуживанию |  |

Приложение В

ГКПОУ ПГТТ им. В.П. Романова

ИНФОРМАЦИОННАЯСИСТЕМА

**«**MARATHON SKILLS**»**

Техническое задание

Версия 0.5

Маркелов В. А.

Прокопьевск, 2018

Информационная система «Marathon skills» является средством автоматизации работников на спортивном мероприятии.

Автоматизированная информационная система может принести огромную пользу для организации, за счет автоматизации задач» которые раньше решались вручную. Преимущества ИС сводятся к следующим ключевым понятиям: быстрее, лучше и больше. Тем не менее для того, чтобы осознать пользу информационных систем, мы должны иметь возможность разрабатывать их вовремя и с минимальными затратами.

1. Основания для разработки

Документ, на основании которого ведётся разработка – задание по курсовой работе «Технология разработки программного обеспечения».

Заказчиком информационной системы является организатор спортивных мероприятий.

1. Назначения разработки

Программа предназначена для ведения учета на спортивных мероприятиях.

Функциональные требования:

1. Простой функционал.
2. Выведение точного результата.
3. Корректность выполнения работы.
4. Постоянное обновление программы.

Эксплуатационные требования:

* Совместимость – поддерживается на устройствах, под управлением операционной системы Windows.
* Простота – упрощенное обращение, ориентированное на обычных пользователей.
* Оптимизация – использование минимального количества ресурсов.
* Результативность – выявления правильного и точного результата.

3 Требования к представлению

Программный продукт должен быть разработан с использованием системы 1С Предприятия. По истечению срока, заказчику необходимо предоставить весь функционал информационной системы, интуитивно понятный интерфейс, и доступ к системе.

Все интерфейсы, документы и пр. визуально аккуратны, единообразны, внешне привлекательны и не содержат ошибок. Выбирать правильное расположение элементов и оставлять между ними необходимое пространство. Логично группируя элементы пользовательского интерфейса по мере необходимости, чтобы обеспечить удобство работы с системой.

Главная форма приложения должна содержать авторизацию, для входа в систему для координатора, администратора и бегуна, а также регистрацию, чтобы незарегистрированный пользователь мог войти в систему в качестве участника.

Меню бегуна должна содержать следующие разделы:

1 Регистрация на марафон – возможность бегуну зарегистрироваться на марафон.

2 Редактирование профиля – возможность редактировать свои персональные данные.

3 Результаты бегуна – просмотр результатов своих гонок.

Меню координатора должна содержать следующие разделы:

1 Управление бегунами – возможность просмотра информации о бегунах.

2 Редактирование профиля бегуна – возможность редактировать профиль бегуна.

3 Печать сертификата об участии – форма предоставляемая возможность записать нормативы участника и предоставить в виде документа.

Меню администратора должна содержать следующие разделы:

1 Управление благотворительными организациями – возможность просмотра информации о благотворительных организации.

2 Добавление/Редактирование благотворительных организаций – возможность редактировать или добавлять благотворительную организацию.

3 Импорт данных – форма предоставляемая возможность импортировать в справочники, данные из excel.

4 Управление пользователями – возможность просмотра информации о пользователей.

5 Добавление пользователя – возможность добавлять пользователя.

6 Редактирование профиля – возможность редактировать пользовательские данные.

Для работы программы необходимы соответствующие параметры:

* Процессор частотой не менее 1,3 Гц, разрядностью 32 бит и выше.
* Оперативная память объемом не менее 1 Гбайт.
* Свободного места на жестком диске не менее 200 Мбайт.

В качестве среды разработки является среда разработки 1С предприятие, т.к. она позволяет разработать хорошую информационную систему. Также позволит разработать удобный интерфейс. Хранится информация будет на специальных серверах.

4 Требование к программной документации

Согласно ГОСТ 19.101-77 Виды программой документации и программ, для разработанного программного продукта необходим следующий состав программных документов:

Спецификация, содержащая состав программы и документации на нее.

Ведомость держателей подлинников, содержащая юридические данные разработчика ПП.

Текст программы, в котором прокомментированы элементы взаимодействия систем.

Описание программы, содержащее сведения о логической структуре и функционировании программы.

Программа и методика испытании, в котором отражена стратегия проверки работоспособности информационной системы.

Пояснительная записка, в котором представляет собой алгоритм различного рода функционала.

Эксплуатационные документы, в котором предъявлены технические программные требования к программному продукту.

Формуляр, в котором отражены основные характеристики программы и сведения об эксплуатации программы, комплексность, сведения об упаковке носителя информации, гарантия поставщика, условия и шаблон подачи жалоб на программный продукт, сведения о хранении, сведения об изменении.

Описание применения, содержащее сведения об информационной системы, которая предназначена для работников турфирмы. Также указаны оптимальные системные и программные характеристики.

Руководство программиста, в котором указаны сведения для эксплуатации программы.

Руководство по техническому обслуживанию для применения тестовых и диагностических программ при обслуживании технических средств.

5 Технико-экономические показатели

Информационная система может получить высокий спрос. Продукт имеет высокую эффективность работы и может конкурировать с другими системами.

В отличие от аналогичных программ наш продукт имеет более удобный интерфейс для пользователя. Так как информационная система, имеющий удобность и функционал, и привлекает внимания клиентов.

6 Стадии и этапы разработки

Как и любой программный продукт проходит несколько стадий и этапов разработки.

1. Стадии и этапы разработки

|  |  |  |
| --- | --- | --- |
| Стадии разработки | Этапы | Содержание работ |
| 1 Техническое задание | Обоснование необходимости разработки программы | Программный продукт предназначен для работников турфирмы. |
|  | Научно исследовательские работы | Для того что бы приступить к работе с программой необходимо изучить инструкцию применения который находится на официальном сайте. |
| 2 Эскизный проект | Разработка эскизного проекта | Входными данными программы является данные  Выходными данными будет сам результат программы |
| 3 Технический проект | Разработка технического проекта | В качестве среды разработки выбран 1С предприятие. Эта среда ориентируется на удобный функционал, защиту системы, и удобный интерфейс. |
| 4 Рабочий проект | Разработка программы | Разработка программного кода велась на языке 1С. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Стадии разработки | Этапы | Содержание работ |
|  | Разработка программной документации | К разработанному продукту необходимо разработать необходимую документацию.   * Спецификация. * Ведомость держателей подлинников. * Текст программы. * Описание программы. * Программа и методика испытаний. * Техническое задание. * Пояснительная записка. * Эксплуатационных документов. * Формуляр. * Описание применения. * Руководство системного программиста. * Руководство программиста. * Руководство оператора. * Описание языка. * Руководство по техническому обслуживанию. |
| 5 Внедрение | Подготовка и передача программы. | Для внедрения необходимо провести тестирование. После тестирования необходимо внести изменения в программу. |

7 Порядок контроля и приемки

Как и было написано для эффективного внедрение продукта нужно провести тестирование. Для этого его необходимо собрать команду тестировщиков. После проведения теста все необходимые ошибки указанными тестерами будут исправлены.

Тестирование будет проводиться в режиме черного и белого ящика.

Тестирование в черном ящике позволит проводить тестирование только через общедоступный пользовательский интерфейс программы.

Тестирование методом белого ящика представляет собой тестирование программного обеспечения, анализируя логику работы программы для получения тестовых данных.

Испытания и тестирование программы должно проводится в процессе создания продукта и контролироваться разработчиками.

1. С использованием тестов эта проверка позволяет добиться правильности работы программного продукта.

2. Путем пошаговых исполнений в программе или модуля в соответствии с набором тестовых примеров и сравнение полученных в процессе тестирования с контрольными значениями тестируемой программы.

3. Так же благодаря активным действиям бета тестеров, которые участвуют в процессе активно и сообщают о любых нарушениях и ошибках так же помогаю разработчикам ускорить процесс выхода полной версии.

Испытания и тестирование программного продукта должно проводится после установки программы и ознакомление с правилами, которые находятся на официальном сайте программного продукта.

Приемка программы должна осуществляться заказчиком. Программа должна являться надежной и внушающее доверие обычным пользователям, если она удовлетворяет всем этим требованиям данного технического задания.

Приложение Г

ГКПОУ ПГТТ им. В.П. Романова

ИНФОРМАЦИОННАЯСИСТЕМА

**«**MARATHON SKILLS**»**

Руководство по техническому обслуживанию

Версия 0.5

Маркелов В. А.

Прокопьевск, 2018

Документ представляет собой руководство по техническому обслуживанию при осуществлении работ в рамках сопровождения информационной системы «Marathon skills».

Руководство по техническому обслуживанию предназначено:

* Администратору.
* Специалисту по техническому обслуживанию.

1 Общие указания

В рамках технического обслуживания системы рассматривается выполнение следующих основных функций:

1. Обновление программного обеспечения.
2. Осуществление резервного копирования информационной базы.

Состав персонала, необходимый для обслуживания Системы, должен обеспечивать выполнение следующих видов технической поддержки:

* обеспечение единой службы поддержки пользователей в случае утери персональных данных;
* проведение профилактических работ;
* обеспечение бесперебойного доступа специалистов к системе.

Для обеспечения квалифицированной поддержки системы, персонал должен обладать соответствующими профессиональными навыками, подтверждаемыми сертификатами производителей установленного оборудования и ПО.

2 Требование к техническим средствам

Для обеспечения корректной работы программного продукта, минимальный набор комплекса технических средств должен состоять из следующих основных элементов:

* Сервер СУБД.
* Персональный компьютер.

Для запуска программного продукта компьютер должен обладать следующими системными требованиями:

1. Системные требования к программному продукту

|  |  |
| --- | --- |
| Минимальные: | Рекомендованные: |
| ОС: Windows XP, 7, 8.1, или 10 | ОС: Windows 7, 8.1, 10 |
| Процессор: 1,3 ГГЦ | Процессор: 2 ГГЦ |
| Оперативная память: 1 ГБ | Оперативная память: 2 ГБ |
| Видеокарта: 128 МБ | Видеокарта: 512 МБ |
| Место на жестком диске: 200 МБ | Место на жестком диске: 200 МБ |

3 Описание функций

В данном разделе приводится порядок выполнения следующих основных функций:

* обновление программного обеспечения;
* осуществление резервного копирования.

Обновление программного обеспечения системы должно производиться не реже одного раза в год. Обязательным условием для обновления является предварительно сделанная резервная копия данных. После установки новой версии текущая установленная информационная база не будет работать со старыми версиями начиная с версии 8.3 или более. Для отката к предыдущей версии, необходимо использовать старую базу. Ответственным за проведение данной процедуры назначается администратор системы.

Резервное копирование информационной системы должно производиться в соответствии с документом «Регламент технического обслуживания». Архивированию подлежат данные в базе, возрастом старше года, предшествующего году архивирования. Архивирование производится ежегодно, период хранения архивных данных пять лет. При повреждение информационной базы необходимо запустить chdbfl.exe. Программа позволяет проверить целостность информационной базы и исправить всевозможные ошибки. За создание резервных копий назначается администратор системы.