**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное   
учреждение высшего образования   
«Санкт-Петербургский политехнический университет Петра Великого»**

**ИНСТИТУТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

|  |  |
| --- | --- |
|  | **Допустить к защите**  Заместитель директора по  учебно-методической работе  \_\_\_\_\_\_\_\_\_\_\_ \_\_Е. Г. Конакина\_\_\_\_\_  (Подпись) (И.О.Ф.)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема Разработка базы знаний для новых сотрудников компании

специальность 09.02.07 группа 42919/4

Студент (ка) Марченкова К.И

(подпись) (ФИО)

Руководитель Рябова В.В.

(подпись) (ФИО)

Санкт-Петербург

2023

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc137236844)

[1 Общая часть 5](#_Toc137236845)

[1.1 Анализ предприятия 5](#_Toc137236846)

[1.2 Анализ работы отделов предприятия 6](#_Toc137236847)

[1.3 Анализ задач сотрудников предприятия 6](#_Toc137236848)

[1.4 Описание предметной области 7](#_Toc137236849)

[1.5 Требования к разрабатываемой базе знаний 10](#_Toc137236850)

[1.6 Описание задач проекта 11](#_Toc137236851)

[1.7 Анализ рынка существующих решений 11](#_Toc137236852)

[1.8 Анализ целесообразности выполнения проекта 14](#_Toc137236853)

[1.9 Анализ средств разработки 14](#_Toc137236854)

[2 Специальная часть 22](#_Toc137236855)

[2.1 Описание структуры приложения 22](#_Toc137236856)

[2.2 Проектирование Базы Знаний 25](#_Toc137236857)

[2.3 Описание таблиц и примеры заполнения 28](#_Toc137236858)

[2.4 Описание программы 30](#_Toc137236859)

[2.5 Тестирование Базы 30](#_Toc137236860)

[3 Экономическая часть 35](#_Toc137236861)

[3.1 Технический и социальный эффект 35](#_Toc137236862)

[3.2 Организационная часть 35](#_Toc137236863)

[3.3 Экономическая часть 38](#_Toc137236864)

[3.4 Определение цены проекта для Базы Знаний для новых сотрудников компании для предприятия ГУП СПб «СПб ИАЦ» 39](#_Toc137236865)

[4 Техника безопасности и охрана труда 42](#_Toc137236866)

[4.1 Анализ условий труда техника-программиста, работающего в «СПб ИАЦ» 42](#_Toc137236867)

[4.2 Расчёт искусственного освещения в Расчёт искусственного освещения в помещении «СПб ИАЦ», где находится рабочее место программиста 44](#_Toc137236868)

[4.3 Электробезопасность в «СПб ИАЦ» 46](#_Toc137236869)

[4.4 Пожарная безопасность в «СПб ИАЦ» 48](#_Toc137236870)

[Заключение 51](#_Toc137236871)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc137236872)

[Приложение А 54](#_Toc137236873)

[Приложение Б 58](#_Toc137236874)

# ВВЕДЕНИЕ

Знание – сила. Всем известное утверждение как никогда становится актуальным в наше время, так как прогресс не стоит на месте и технологии постоянно развиваются. Вследствие чего становится затруднительно сделать логический вывод на основании тех или иных фактов, для облегчения процесса установления вывода необходимо формализировать знания, упорядочить и представить в виде некой структуры, удобной для хранения и отображения информации.

В настоящее время, интернет стал не только наиболее популярным способом общения и обмена информацией, но также и торговой площадкой с множеством уникальных функций и преимуществ. Безусловно, большинство кампаний осваивают, а значительная часть уже освоили, механизмы и методики продажи товаров и услуг через сеть интернет.

Дипломная работа является важным этапом подготовки квалифицированных специалистов. Она является видом учебно-вспомогательного процесса, в ходе которого закрепляется теоретические знания.

В соответствие с целью определим следующие задачи дипломной работы:

* выполнить структурный анализ предметной области;
* проведено объектно-ориентированное проектирование системы, построены UML диаграммы этапа структурирования;
* выполнить функциональное моделирование потоков данных;
* изучение особенностей строения, состояния и функционирования конкретных информационных процессов;
* выполнить моделирование даталогической модели базы данных;
* спроектировать интерфейс приложения.

## **1 Общая часть**

## **1.1 Анализ предприятия**

Санкт-Петербургский информационно-аналитический центр – это государственное унитарное предприятие. Это означает, что СПБ ИАЦ находится в ведении Комитета по информатизации и связи Санкт-Петербурга и напрямую деятельность данного предприятия направленна на улучшение и автоматизацию работы государственных систем и служащих.

В компетенции СПб ИАЦ – разработка и реализация информационных и информационно-аналитических проектов и систем в различных предметных областях.

Экспертиза СПб ИАЦ базируется на многолетнем опыте успешной работы в сфере информатизации органов государственной власти и ряде проектов для государственных и коммерческих компаний.

Основными задачами СПб ИАЦ являются создание, сопровождение и системная интеграция информационных и информационно-аналитических систем Администрации Санкт-Петербурга и подведомственных ей организаций на основе современных достижений информационных и телекоммуникационных технологий.

Основы политики СПБ ИАЦ:

* личное участие руководства предприятия, стимулирование и поддержка предложений и инициатив сотрудников в определении и реализации политики, целей и текущих задач в области качества, разумной интеграции требований стандартов качества в бизнес-процессы предприятия;
* комплексное и сбалансированное управление процессами достижения качества на основе базовых принципов корпоративной культуры, с использованием процессного подхода, оценки рисков и возможностей;
* применение системы менеджмента качества для выявления и практического использования прогрессивных методологий и лучших практик разработки, эксплуатации и защиты информации информационно-аналитических и информационных систем в разных предметных областях;
* выявление и стимулирование развития основных и вспомогательных бизнес-процессов, способствующих реализации маркетинговой стратегии предприятия.

## **1.2 Анализ работы отделов предприятия**

Деятельность предприятия состоит из работы нескольких разделов.

Первым делом отдел аналитики собирает статистику, анализирует потребности администрации города и государственных служащих, общается с клиентами и составляет план требования к системе.

Далее в ход вступает отдел руководителей проектов. Данный отдел занимается постановкой задачи для программистов, составлением технического задания и другой программной документации.

Отдел программистов непосредственно занимается разработкой новых систем, и что не мало важно, поддержкой работоспособности старых систем.

Также в штате предприятия присутствуют системные администраторы, начальники отделов и люди, с разными специальностями, необходимые для обеспечивайся качественной и продуктивной работоспособности предприятия.

## **1.3 Анализ задач сотрудников предприятия**

Работа предприятия ГУП СПБ ИАЦ строится на взаимодействии множества сотрудников, ведь только грамотное управление персоналом, которым занимаются руководители проектов, позволяет эффективно выполнять поставленные перед предприятием цели.

Работа руководите проекта сложна, ведь необходимо взаимодействовать с людьми разных разделов, то есть руководитель проекта должен получать информации у так называемых «аналитиков», то есть людей, анализирующих потребности администрации города и государственных служащих. Аналитики собирают статистику, общаются с потенциальными клиентами и описывают системы, необходимые для реализации.

Тут руководители проектов разбирают проекты, составляют техническое задание, обсуждают с начальством необходимые тонкости и дают указания программистам.

Тут в дело вступают программисты, они, основываясь на информации, предоставленной аналитиками, проектируют и реализуют систему, это трудоёмкий и долгий процесс, очень часто сопровождаемый внесением дополнительных правок в течении реализации проекта. По завершению создания системы, проведения тестирования и исправления ошибок, программист просит системного администратора выгрузки проекта на производственный сервер, то есть сервер, доступный для использования клиентами.

На этом работа программиста редко когда заканчивается над проектом, программист и дальше продолжает улучшать работу приложения, вносит новый функционал, а также исправляет ошибки.

## **1.4 Описание предметной области**

На время прохождения производственной практики в ГУП СПБ ИАЦ была поставлена задача разработать базу знаний, в которой можно сохранять важную информацию и полезные материалы, собрать воедино все активы компании: важные документы, знания и опыт сотрудников.

База знаний представляет собой единое хранилище корпоративной информации, доступ к которому есть только у членов определенной организации. Проще говоря, это личная Википедия для персонала, где в любой момент можно получить ответ на любой вопрос, связанный с компанией и рабочим процессом.

База знаний бывает двух видов – внутренняя и внешняя.

**Внутренняя база знаний**. Обслуживает только сотрудников компании и недоступна сторонним пользователям. Может содержать такую информацию:

* материалы для адаптации новых сотрудников;
* данные о трудоустройстве, отпусках, больничных и зарплатах;
* должностные инструкции и правила поведения в офисе.
* обучающие материалы для разных отделов и специалистов: как нанимать сотрудников, как общаться с клиентами, как оформлять договоры;
* советы по организации удаленной работы;
* нововведения в компании.

**Внешняя база знаний**. Обслуживает клиентов компании. Находится в открытом доступе, ее легко найти в интернете. Может содержать такую информацию:

* ответы на часто задаваемые вопросы;
* обучающие материалы: как пользоваться сервисом или программой;
* информация о продуктах и услугах компании;
* тенденции и изменения в отрасли.

**Для чего нужно создавать базу знаний?**

Основная задача любой базы знаний – это сокращение времени на поиск нужной информации и возможность быстрого получения ответов на типовые вопросы.

Базу знаний можно использовать как:

* инструмент продвижения бизнеса в поисковой выдаче;
* портал вопросов и ответов для сотрудников или клиентов;
* сервис для обмена знаниями между отделами или сотрудниками компании.

С необходимостью аккумуляции знаний для сотрудников сталкивается почти каждая компания на определенном этапе. Например, нужно быстро обучить новичков, поддержать уровень знаний сотрудников на должном уровне или показать примеры работ всему коллективу.

База знаний помогает:

* адаптировать новых сотрудников, разъясняя им правила поведения в компании и должностные обязанности;
* демонстрировать новичкам примеры документов и решений, не отвлекая других специалистов;
* сохранять всю важную информацию, распределяя ее по тематическим разделам;
* объединить знания отдельных сотрудников в корпоративные знания и сохранить их в едином поле.

Самое главное — все знают, что и где найти, и никто не задает повторяющихся вопросов. Не имеет значения, к какой сфере относится ваш бизнес. База знаний актуальна при любом виде деятельности.

Нет каких-то стандартов, какой должна быть база знаний. Но делать её унылым хранилищем только должностных инструкций точно не стоит. В базу знаний можно добавить следующие материалы:

1. Сведения о компании: здесь можно описать деятельность компании и рассказать о правилах поведения в офисе. Также можно указать контактные данные всех сотрудников, состав отделов, список контрагентов. В этом же разделе хранят все должностные инструкции, коллективные договоры, приказы и иные официальные документы, касающиеся трудовых обязанностей.
2. Полезные статьи: в создании базы знаний могут принимать участие все сотрудники. К примеру, кто-то решит поделиться своим опытом или рассказать об интересном решении задачи. Можно создать «песочницу», в которую каждый сотрудник может писать то, что сочтёт полезным. После обсуждения и проверки можно перемещать статью непосредственно в базу знаний.
3. Вопросы-ответы: наверняка в любой компании есть вопросы, которые сотрудники задают со стабильной регулярностью. Таким образом, можно собрать все самые частые вопросы и дать на них ответы.

## **1.5 Требования к разрабатываемой базе знаний**

Чтобы наладить систему хранения и использования данных в компании, внутрикорпоративная платформа должна обладать следующими качествами:

1. **Стабильность работы.** Большие корпорации могут использовать базы для поддержки клиентов. А это значит, что отказ в работе платформы приведет к убыткам. Поэтому система должна работать безотказно.
2. **Безопасность внутренней информации.** Иногда системы могут быть открыты для всех пользователей: такие платформы не требуют строгих настроек доступа. Но если речь идет о конфиденциальной информации в крупной компании, то безопасность должна быть на первом месте.
3. **Структуризация**. Продуктивность сотрудников, работающих в больших корпорациях, падает, если не удается найти нужный файл или запись. Структуризация информации повышает продуктивность сотрудников.

## **1.6 Описание задач проекта**

В ходе анализа требований к разработке системы можно поставить следующие задачи:

* разработать базу данных;
* разработать удобный пользовательский интерфейс;
* разработать приложение ASP.NET Core на C#;
* провести испытания программы;
* ввод в эксплуатацию.

Задача «Разработка базы данных» включает в себя разработку ER-диаграммы и диаграммы связей, а также реализация ее, с использованием СУБД MYSQL Server.

Задача «Разработка удобного пользовательского интерфейса» включает в себя проектирование интерфейса, с учётом пожеланий пользователей системы и установленных стандартов.

Задача «Разработка приложения ASP.NET Core на C#» включает в себя реализацию программы с использованием языка программирования C# и фреймворков для пользовательского интерфейса.

Задача «Проведение испытаний» включает в себя составления test-case для проверки работоспособности всех модулей программы.

## **1.7 Анализ рынка существующих решений**

Проанализировав рынок существующих решений, сделав некоторые выводы, мы понимаем, что на данный момент многие системы предоставления такого функционала далеко не идеальны. Среди отечественных разработок найдено несколько систем таких, как платформа знаний «**KNOWLEDGE SPACE**», платформа развития знаний сотрудников «TEAMLY» и «L2U InKnowledge».

**Платформа KNOWLEDGE SPACE**

Это социально-технологическая платформа знаний, ориентированная на создание цифрового слоя управленческих процессов организации и поддерживающих данный слой ИТ-решений.

KNOWLEDGE SPACE интегрирует основные компоненты управленческой ИТ-системы: систему планирования, бизнес-архитектуру и ИТ-архитектуру предприятия (рисунок 1).

KNOWLEDGE SPACE объединяет в себе следующий функционал:

* конструктор моделей процессов, объектов, показателей, формул;
* инструменты планирования, мониторинга, анализа, поддержки решений;
* средства коммуникаций и совместной работы – социальная сеть;
* средства администрирования ролей модели.

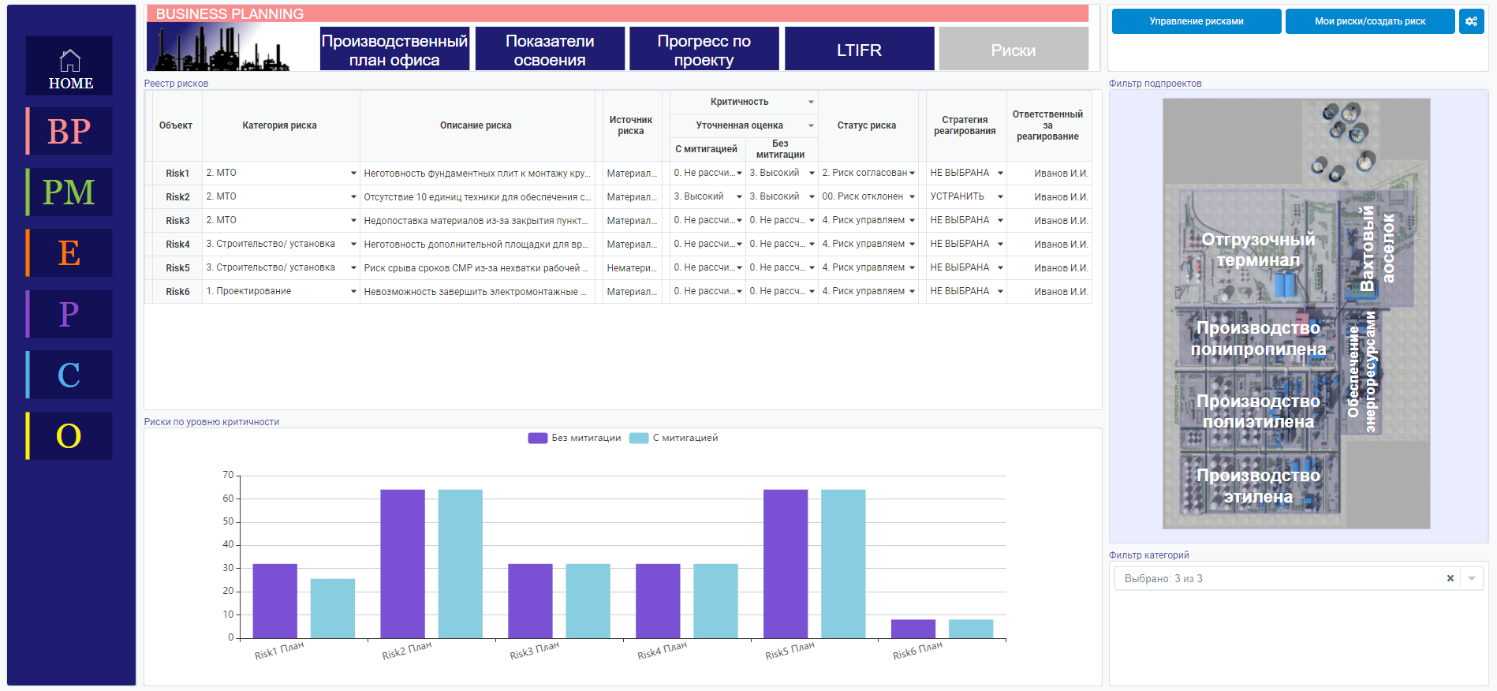


Рисунок 1 – KNOWLEDGE SPACE

**Платформа TEAMLY**

На платформе есть полноценная рабочая среда, где можно создавать документы или статьи, редактировать их вместе с коллегами и сохранять в базе знаний. История версий сохраняется. Для каждого отдела свое рабочее пространство и хранилище файлов. Есть умная лента и уведомления об изменениях. Права доступа настраиваются (рисунок 2.).

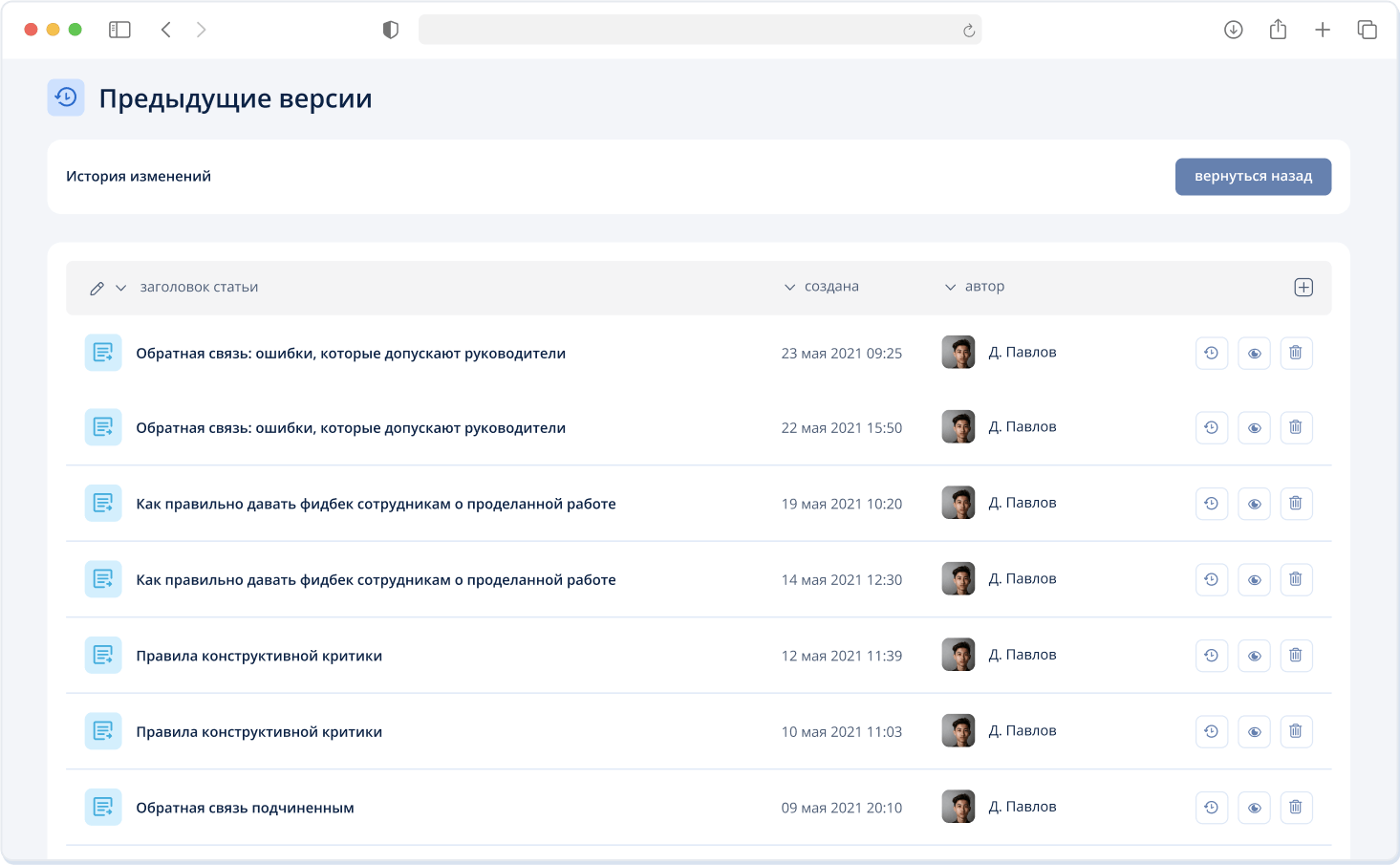


Рисунок 2 - TEAMLY

Базу знаний можно использовать для обучения новичков или повышения квалификации. Для этого есть встроенный конструктор курсов. Модуль тестов и опросов для закрепления пройденного. Можно просматривать статистику.

**Платформа L2U InKnowledge**

L2U InKnowledge - платформа для организации единого информационного центра компании. Она позволяет создавать и поддерживать в актуальном состоянии информацию, как необходимую для обслуживания клиентов во всех каналах коммуникации, так и информацию, используемую для операционной деятельности сотрудников back-офиса (Бэк-офис — сотрудники внутренних отделов компании, которые не взаимодействуют с клиентами.) (рисунок 3.).

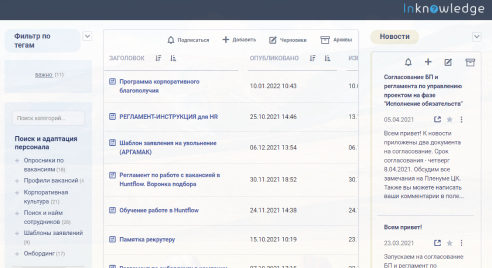


Рисунок 3 - L2U InKnowledge

Вывод: на таких база присутствует неудобный интерфейс, который сбивает пользователя с поиска нужной информации. Именно поэтому мы ставим задачи на разработку базы знаний, которая имеет уникальный дизайн и не будет вводить пользователей в заблуждение.

## **1.8 Анализ целесообразности выполнения проекта**

Так как по результатам анализа рынка существующих решений не было выявлено программ, обеспечивающих необходимый функционал, а также обеспечивающих простой и удобный пользовательский интерфейс было решено, что следует разработать свою базу, которая будет удовлетворять все необходимые требования.

## **1.9 Анализ средств разработки**

Для начала необходимо определится чем будет являться приложение. Есть несколько вариантов для реализации разрабатываемого продукта:

1. Мобильный сайт: мобильный сайт – это адаптивная версия сайта под мобильные устройства, т.е. это тот же самый сайт, но который хорошо выглядит и удобен в использовании на планшетах и смартфонах.

Преимущества мобильного сайта:

* + Простое использование – пользоваться сайтом будет гораздо большее количество пользователей из-за простоты перехода, не нужно делать лишние действия. Чтобы использовать приложение, его необходимо будет установить. Если верить статистике, то сравнительно большой процент пользователей так и не установит приложение, даже зайдя в магазин или маркетплейс.
  + Мобильный сайт выглядит корректно на всех устройствах не зависимо от платформы (Windows Phone, Apple, Android).
  + Сайт легко обновлять и все изменения будут тут же происходить у пользователя.
  + Стоимость мобильной версии всегда в разы дешевле, чем приложения.

1. Мобильное приложение: мобильное приложение – программный продукт, который устанавливается пользователями через магазины, маркетплейсы. Его главная и отличительная черта в том, что разработка ведется под каждую платформу отдельно. Это связано с различием функционала операционных систем и нужно для корректной работы приложений. Бывают случаи, когда приложение, которое на одной платформе работает замечательно, а на другой, ужасно. Также нередки случаи, что на одной платформе полный функционал, а на другой – его обрезанная версия.

Преимущества приложений:

* + Можно отправлять push-уведомления. А также напоминать, о том, что пользователь давно не заходил в приложение, либо о каком-то важном событии. На сайте такая функция не доступна.
  + Зачастую сайты уступают приложениям по интерфейсу, т.к. они значительно более громоздкие. Следовательно, уместить их в маленький экран достаточно сложный процесс. С приложением все проще. На экране будет появляться только нужная информация, кнопки будут достаточно крупными и не нужно использовать точилку для пальцев, чтобы попасть в нее, как в случае с некоторыми сайтами.
  + Пользователям достаточно просто и легко разобраться в работе приложения, чем в работе сайта, особенно со сложной структурой. Ведь у каждой платформы есть свои стандартные действия (например, перемещение между экранами приложения, вызова меню и т.д.).
  + В любом приложении есть возможность запоминать действия пользователя и адаптировать приложение под него. Например, пользователь вводит какие-то данные. В дальнейшем использовании приложения данные больше вводить не придется.
  + Работа с приложением без использования интернета, вы всегда сможете воспользоваться приложением даже в оффлайн режиме. И это достаточно весомый аргумент, которого нет у сайтов.
  + Подстройка приложения под пользователя, популярные платформы умеют следить за своим владельцем и точно знают, когда и что он делает (ставит на зарядку, использует определенное приложение, не пользуется смартфоном и т.д.). Благодаря этому вы всегда сможете вносить изменения и дополнения в приложение к моменту его использования.

Вывод: Мной был выбран сайт. Так как он позволяет использовать Базу Знаний с персонального компьютера, так же сайт проще и быстрее разрабатывать.

Далее необходимо определится с интегрированными средствами разработки. Для C# существует огромное количество платных и бесплатных IDE.

Примеры доступных IDE для C#:

1. **Visual Studio**: с Visual Studio многие начинают знакомиться с языком и не расстаются с ней на протяжении всей карьеры программиста.

**Плюсы**:

* Официальная. Так как и язык, и среда разработки созданы в Microsoft, логично предположить, что ничего более функционального вы не найдете во всем Интернете. В некоторых случаях без Visual Studio не обойтись – например, при использовании технологий UWP и WPF.
* Бесплатная. Версии «Community edition» для рядового пользователя будет достаточно. Тем более, теперь можно подключать плагины (в отличие от старой версии Express).
* Функциональная. В Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки.
* Поддерживает платформы .NET. Visual Studio имеет широкие возможности по разработке приложений под Windows, в том числе в .NET-сегменте.
* Облачные хранилища. Зарегистрируйтесь в сообществе Visual Studio – и получите доступ к облачному хранилищу, где сможете располагать файлы проектов.
* Корпоративность. Технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

**Минусы**:

* Баги при переходах с триал-версии. При переходе на платную версию могут теряться настройки и нарушаться работа корпоративного сервера.
* Сложность. Самостоятельно освоить Visual Studio новичку будет непросто — слишком много доступных функций, спрятанных в подразделах меню.

1. **Project Rider:** среда от JetBrains для работы с платформой .NET. Выпущена в прошлом году, но уже приобрела много поклонников.

**Плюсы**:

* ReSharper. Это плагин, изначально разработанный для повышения производительности Visual Studio. Теперь на его основе выпущена IDE.
* Поддержка полного цикла. Фирменная черта продуктов JetBrains, воплощенная и в Project Rider. С ним вы сможете организовать весь цикл создания ПО: от идеи до поддержки.
* Функциональность. Project Rider позволяет подключить MSBuild и XBuild, работать с CLI-проектами и организовать отладку приложений .NET and Mono. Множество опций для быстрого создания кода улучшает производительность.
* Multiple runtime. Поддержка нескольких запущенных программ.
* Кроссплатформенность. Project Rider работает с Windows, Linux и MacOS.
* Контроль версий. Встроенный инструмент позволяет напрямую организовать работу с Git, Mercurial и TFS.

**Минусы**:

* Молодость. Часть функциональности еще в разработке, не все стартовые баги исправлены.
* Стоимость. Самая дешевая версия Project Rider обойдется в 139 долларов за первый год использования. Но есть триал-версия и специальные предложения для студентов и непрофильных организаций.

1. **Eclipse:** одна из самых популярных мультиязычных сред. Ориентирована преимущественно на разработку Java-приложений, но полезна и для кодов на C#.

**Плюсы**:

* Множество плагинов. У Eclipse едва ли не самое большое число надстроек.
* Активное сообщество. Помогает быстрее освоить среду разработки, выпускает новые плагины.
* Отличные компилятор и отладчик. Первый работает на порядок быстрее, чем у конкурентов, второй — показывает потоки, пересечения, позволяет гибко управлять ходом отладки.
* Кастомизация. Благодаря плагинам и настройкам можно полностью персонализировать Eclipse.
* Бесплатность. Это open-source проект, абсолютно бесплатный.
* Высокая функциональность. Благодаря разработчикам-официалам и членам сообщества с помощью Eclipse можно провести любой C#-продукт по полному циклу разработки.

**Минусы**:

* Сложность. Как и любой функциональный продукт, Eclipse может показаться новичку слишком сложным.
* Нет гарантий надежности. Так как плагины создаются сообществом, за их качество отвечает только разработчик. Кроме того, сами создатели Eclipse с каждой новой версией плодят баги, не успевая порой исправлять старые.

1. **Visual Studio Code:** кроссплатформенный редактор кода, который при помощи плагинов можно «подтянуть» к статусу IDE.

**Плюсы**:

* Кроссплатформенность. Работает на MacOS, Ubuntu и Windows. Пока недоступен на Android и iOS.
* Бесплатность. Простой open-source редактор и плагины — платить не надо.
* Легковесность. Потребуется совсем мало ресурсов, чтобы приступить к работе с минималистичным VSC.

**Минусы**:

* Низкая функциональность. Несмотря на поддержку .NET-платформы, VCS неудобен для сложных проектов.
* Сомнительная надежность. Многие надстройки имеют низкое качество сборки и не всегда выполняют даже основные функции.

1. **MonoDevelop:** свободная среда разработки от Xamarin для создания приложений на множестве языков, в том числе на C#.

**Плюсы**:

* Мультиплатформенность. Поддерживает Linux, Windows и Mac OS.
* Кастомизация. На рабочем столе можно расположить функции и окна по своему усмотрению.
* Unity 3D. Полноценная поддержка популярной платформы для разработки игр.
* Бесплатность.

**Минусы**:

* Ограниченная функциональность. У MonoDevelop есть собственный отладчик и инструменты для работы с кодом. Но в части поддержки разных платформ и проектов — это пока сырая IDE.

Вывод: Мной было выбрано Visual Studio, т.к это лучшая интегрированная среда разработки для создания многофункциональных, кроссплатформенных приложений. Предоставляет разработчикам полнофункциональную среду разработки для эффективной и совместной разработки высококачественного кода.

При разработке также необходимо было приложение для создания макета и дизайна сайта.

Выбор состоял из таких вариантов как:

а) Axure.

Одним из главных достоинств программы является возможность программирования поведения кнопок, текстовых полей, панелей и прочих виджетов, вследствие чего получившиеся макеты или прототипы приближены к окончательному результату и доступны для тестирования. Взаимодействие со многими виджетами, поставляемыми вместе с Axure RP, может быть настроено с помощью создания тестовых случаев, условий, обрабатываемых событий и выполняемых действий. Например, для выпадающего списка можно обрабатывать следующие события: нажатие, смена элемента, фокус и потеря фокуса.

С программой поставляется лишь небольшое количество виджетов. Тем не менее, на сайте производителя можно скачать дополнительные библиотеки виджетов, созданные как разработчиками, так и пользователями программы. Полезные библиотеки виджетов можно скачать и на других сайтах. Другая возможность получения сложного виджета – это его создание с помощью виджета Panel, используя который можно создавать очень сложные интерактивные компоненты.

б) FlairBuilder.

Это приложение позволяет начинать работу с самого приблизительного эскиза и заканчивать макетом страницы (wireframe) большой точности.

Группа или конечные пользователи могут ознакомиться в работе с ранними опытными образцами, это дает обратную связь. При этом пользователи могут быстро учесть указания, проверить различные подходы и выбрать лучший. Более 60 встроенных компонентов прототипа (виджетов) - текстовые вставки (заголовки, текстовые области, параграфы, маркированные списки, ссылки и пр.), навигационные панели (вертикальные, горизонтальные, раскрывающие списки, аккордеон, раскрывающиеся вкладки и пр.), компоненты веб-форм, компоненты для мобильных телефонов и другое. Компоненты полностью интерактивны и готовы к использованию.

в) NinjaMock.

С помощью NinjaMock очень просто можно создать структуру сайта под множество платформ. Пользовательский интерфейс интуитивно понятный, а в библиотеку шаблонов каждый пользователь может добавить свой макет.

г) Moqups.

Множество шрифтов, разнообразные шаблоны, а также встроенная библиотека значков делают Moqups максимально удобной для создания структуры сайтов онлайн. Можно выбрать подходящий для ваших целей тарифный план, либо воспользоваться бесплатной версией для создания одного простого проекта.

Вывод: мной было выбрано Moqups, так как у него большой выбор компонентов во встроенной библиотеки. Так же там есть возможность создания интерактивных проектов.

## **2 Специальная часть**

## **2.1 Описание структуры приложения**

База знаний состоит из правил анализа информации от пользователя по конкретной проблеме. База пользователя включает в себя базу данных, содержащую правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области. Здесь же содержится подробная информация о базе знаний на главной странице, модуль авторизации пользователей. Также в разделе О компании находится информация об организации, в разделе Часто задаваемые вопросы находится доска с вопросами.

Карта сайта содержит всего несколько страниц, но в виду специфики веб-приложения они являются динамическими и пользователю нет необходимости перемещаться между страницами для работы с сайтом. Карта сайта представлена на рисунке 1.

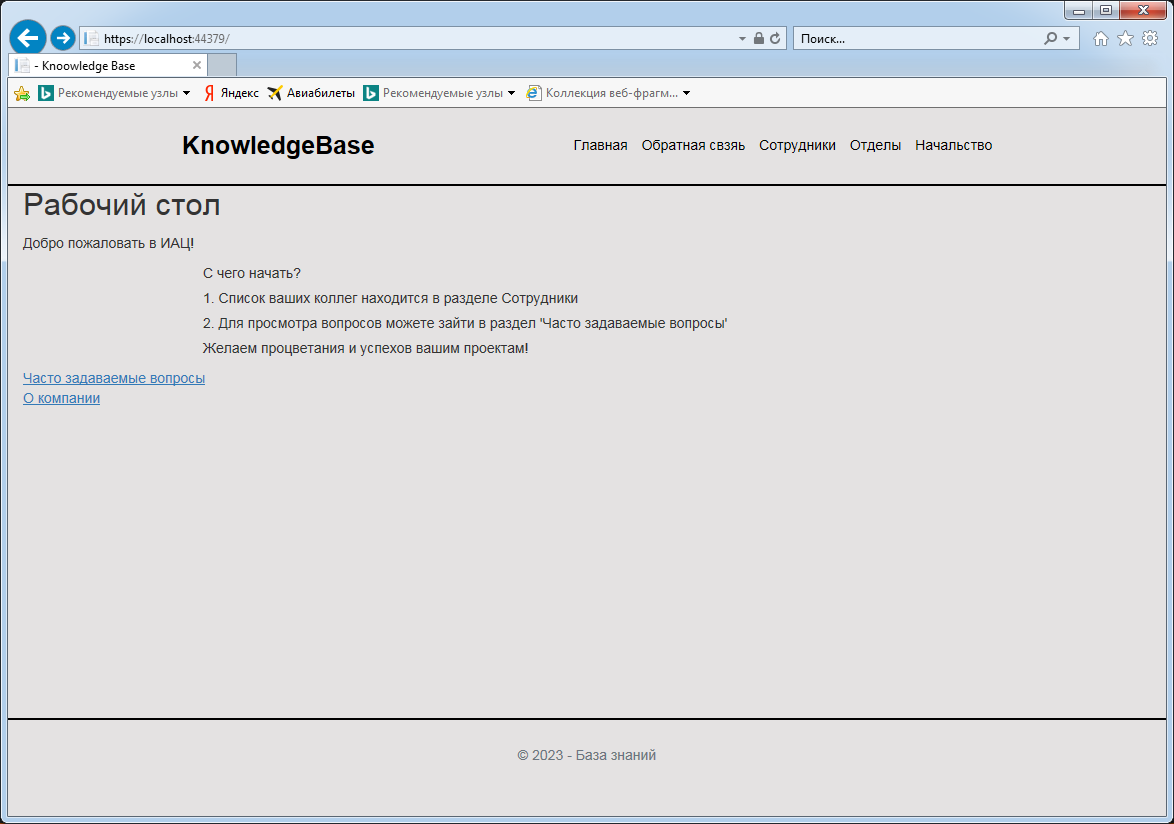


Рисунок 1 – Карта сайта

На главной странице располагается рабочий стол, где пользователь может получить доступ ко всем функциям веб-приложения. Для того, чтобы войти на главную страницу необходимо авторизоваться в системе. Главная страница сайта представлена на рисунке 2.

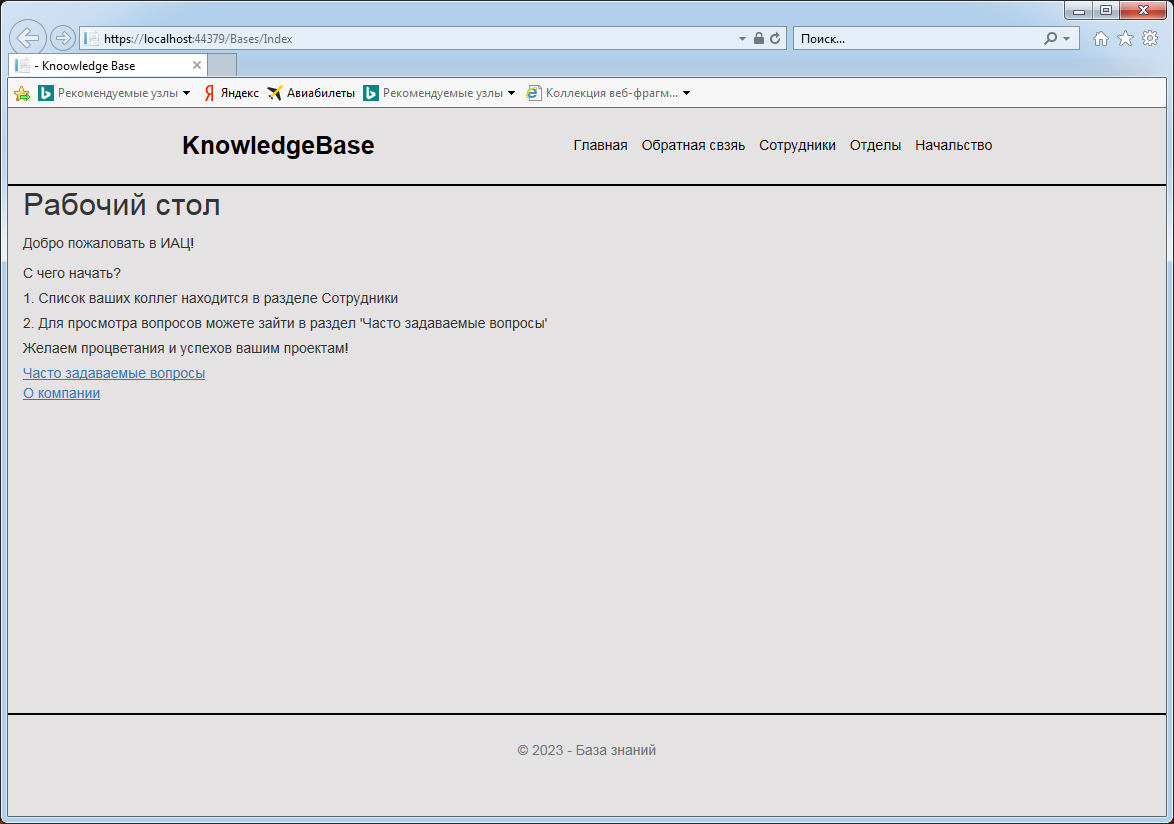


Рисунок 2 – Главная страница

В разделе О компании находится информация о компании, названия отделов, количество сотрудников в этих отделах и каждый сотрудник указан в каком отделе он работает и кем, чтобы его можно было найти или связаться с ним. Внешний вид раздела О компании представлен на рисунке 3.

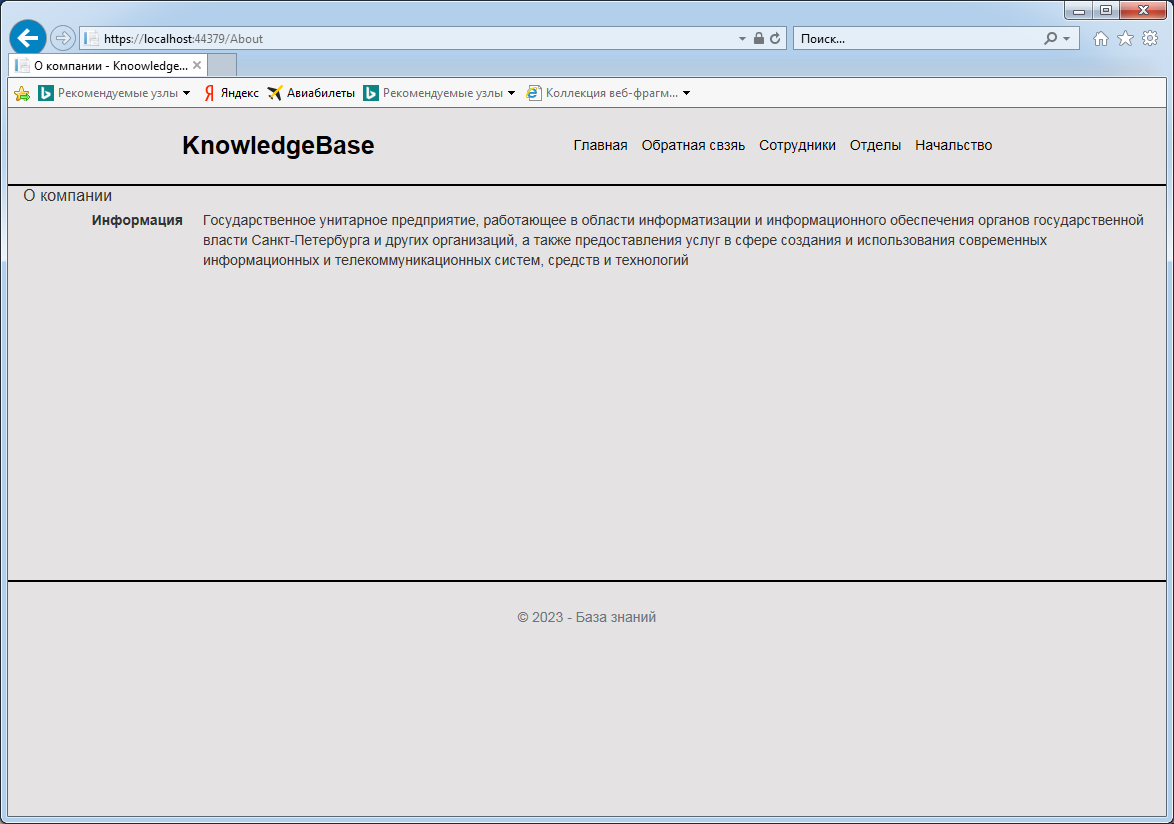


Рисунок 3 –Страница О компании

На странице Часто задаваемые вопросы находится доска с вопросами. Там можно найти самые востребованные вопросы сотрудников и ответы на них. Также там находится чат для вопросов, которых нет на данной странице. Внешний вид страницы Spaces представлен на рисунке 4.

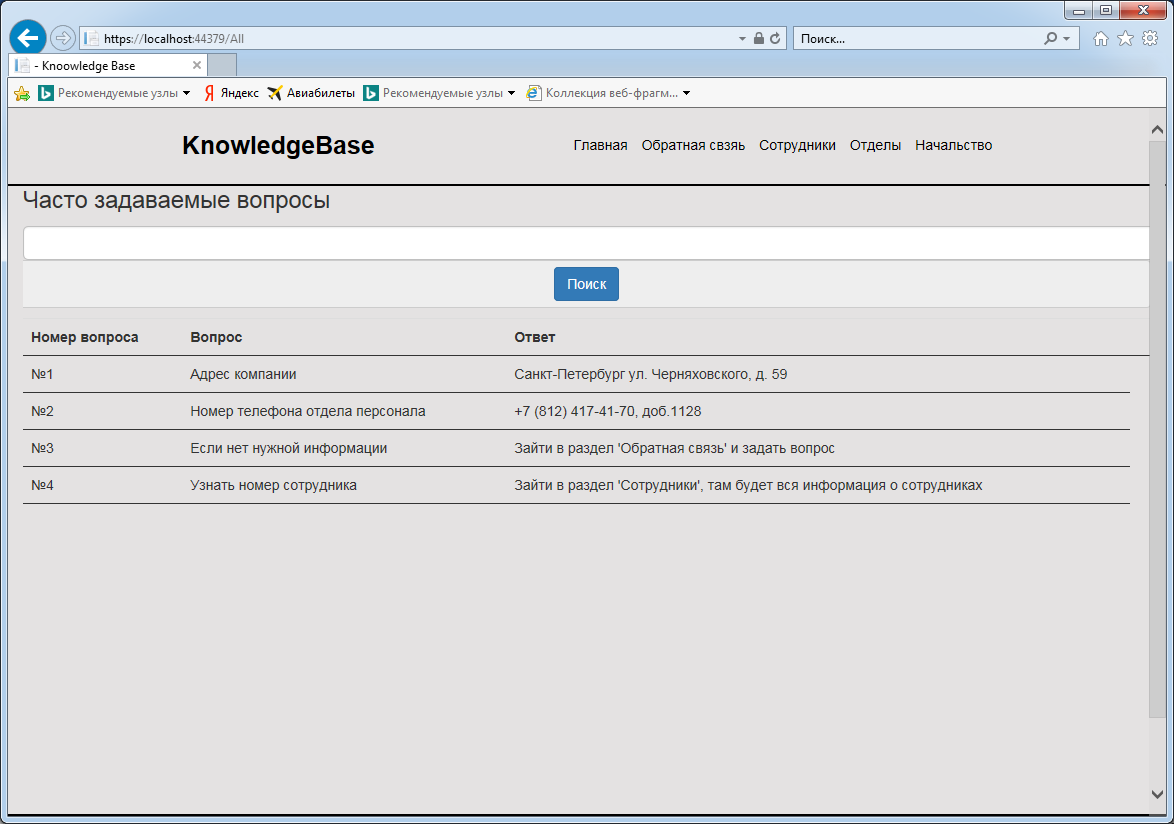


Рисунок 4 – Страница с вопросами

На странице Обратная связь находятся форма. На этой странице можно заполнить форму для неизвестного пользователя вопроса. Внешний вид страницы Обратная связь представлен на рисунке 5.

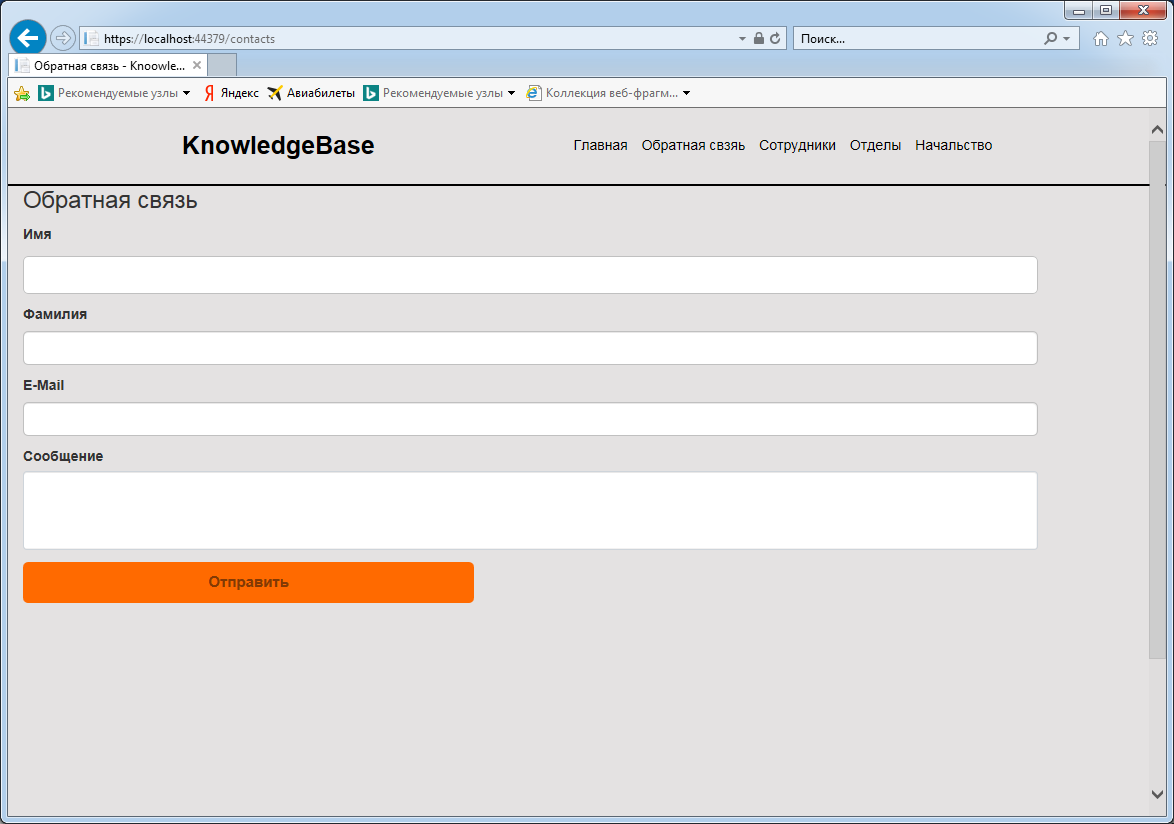


Рисунок 5 –Страница Обратная связь

## **2.2 Проектирование Базы Знаний**

**2.2.1 Объектно-ориентированное проектирование**

С помощью UML разработчики программного обеспечения могут обеспечить полное соглашение в используемых графических обозначениях, чтобы представить общие понятия, такие как: компонент, обобщение, класс, поведение и агрегация. За счет этого достигается большая степень концентрации на архитектуре и проектировании.

Также стоит отметить, что есть несколько видов таких диаграмм. Для данной информационной системы были разработаны такие как диаграмма последовательности, диаграмма прецедентов, диаграмма классов, диаграмма состояний.

Диаграмма последовательности – такая диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Основные элементы диаграммы последовательности это: обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

Диаграмма последовательности представлена на рисунке 6.

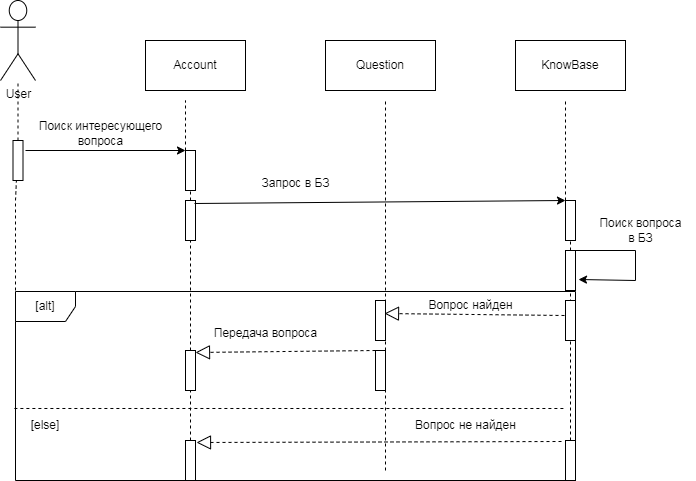


Рисунок 6 – Диаграмма последовательности

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 7.

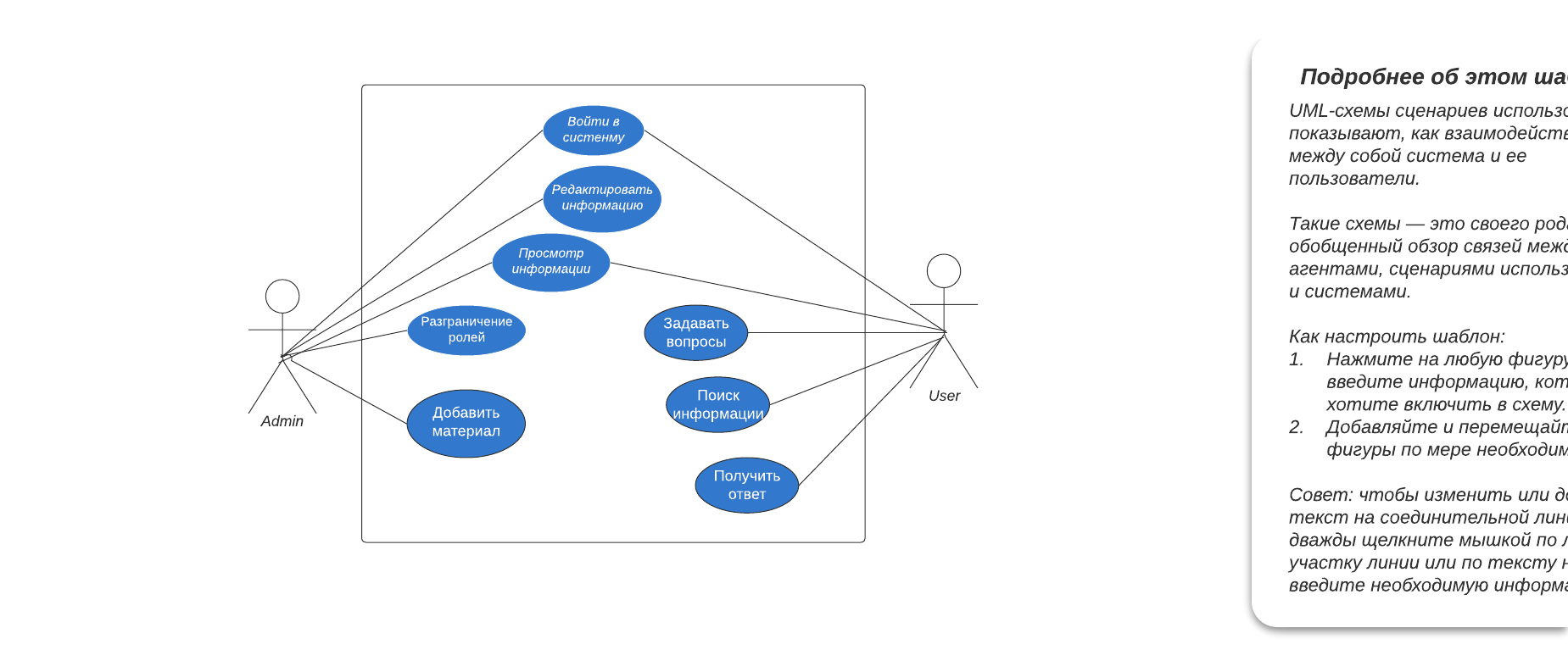


Рисунок 7 – Диаграмма прецедентов

Диаграмма состояний – это тип диаграммы, используемый в UML для описания поведения систем, который основан на концепции диаграмм состояний Дэвида Харела.

Диаграмма состояний представлена на рисунке 8.

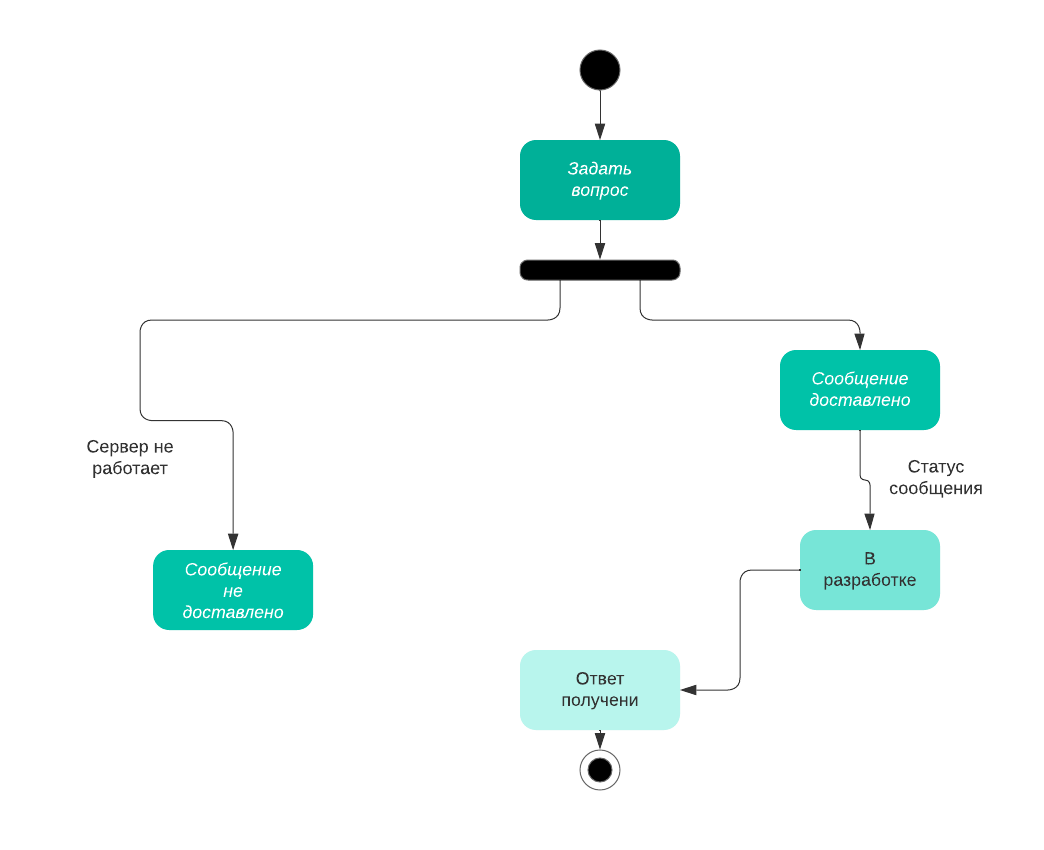


Рисунок 8 – Диаграмма состояний

Так же была разработана схема вариантов взаимодействия пользователя с веб-приложением (рисунок 9).



Рисунок 9 – Схема вариантов использования

**2.2.2 Построение логической модели Базы данных**

Проанализировав все сущности и связи между ними была составлена ER-диаграмма, которая представлена на рисунке 10.

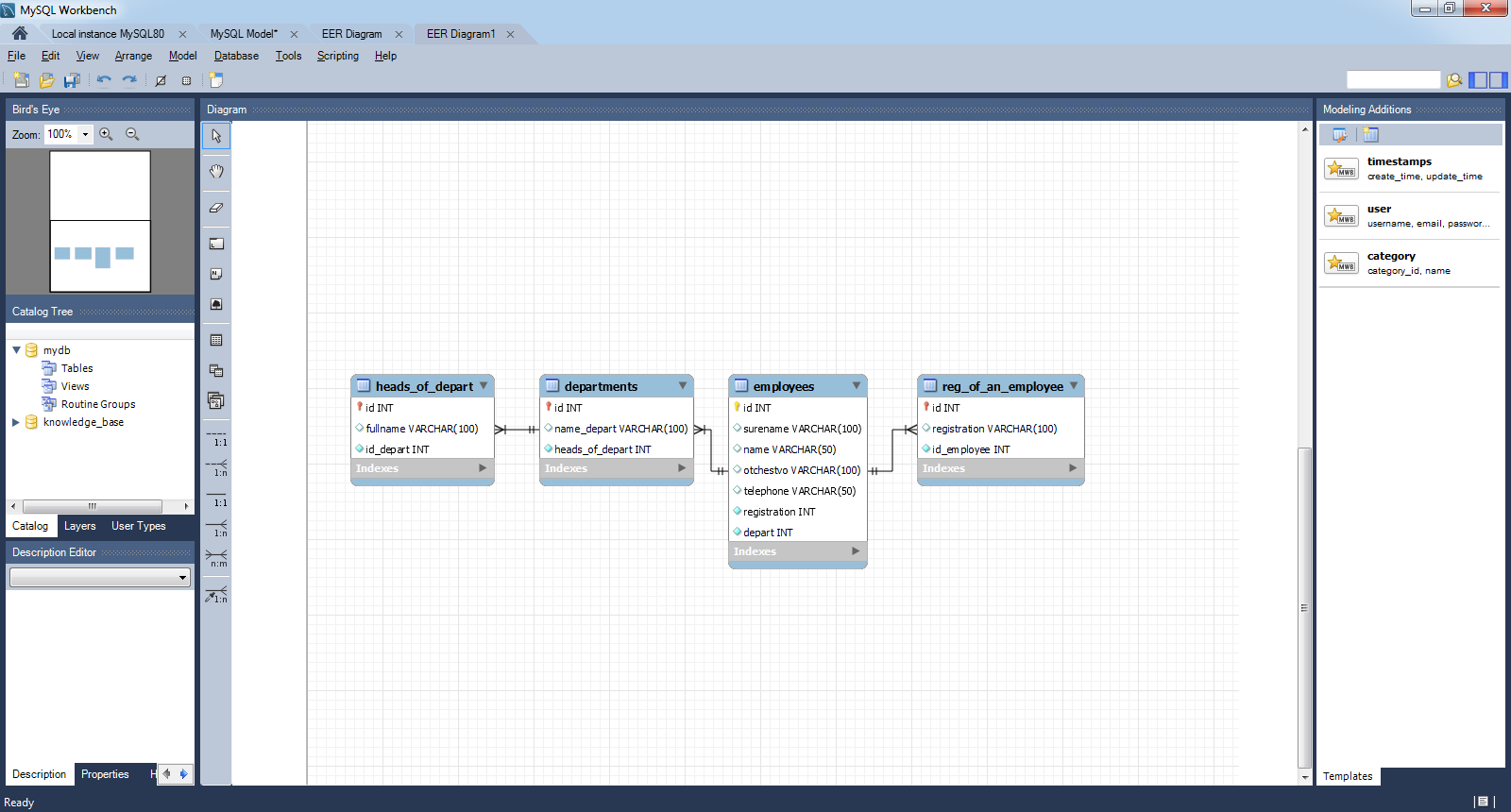


Рисунок 10 – Диаграмма сущнось-связь

БД состоит из 4 таблиц:

* начальники отделов – хранит информацию о тех кто возглавляет отделы;
* отделы – хранит информацию о наиминовании отделов;
* сотрудники – хранит информацию о данных сотрудников и их контактную информацию;
* оформление сотрудников – хранит информацию об оформлении сотрудников будь он официально оформлен, либо стажер.

## **2.3 Описание таблиц и примеры заполнения**

Для проверки составления схемы базы данных, а также лучшего понимания отношений таблиц и данных в базе, необходимо описать каждое поле каждой таблицы.

Таблица 1 – Описание таблицы «Employees»

| **Employees** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Имя поля** | **Id** | **Ключ** | **Тип данных** | **Размер** | **Допустимо ли NULL** | **Описание** |
| ID | **PK** |  | Int |  | NOT NULL | Первичный ключ |
| Surname |  |  | Varchar | 100 |  | Фамилия сотрудника |
| Name |  |  | Varchar | 50 |  | Имя сотрудника |
| Otchestvo |  |  | Varchar | 100 |  | Отчество сотрудника |
| telephon |  |  | Varchar | 50 |  | Телефон сотрудника |
| Registr |  | FK | Int |  | NOT NULL | Внешний ключ |
| Depart |  | FK | Int |  | NOT NULL | Внешний ключ |

Таблица 2 – Описание таблицы «Registration\_of\_an\_employee»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Registration\_of\_an\_employee** | | | | | | |
| **Имя поля** | **Id** | **Ключи** | **Тип данных** | **Размер** | **Допустимо ли NULL** | **Описание** |
| ID | **PK** |  | Int |  | NOT NULL | Первичный ключ |
| Registration |  |  | Varchar | 100 |  | Оформление |
| ID\_employee |  | **FK** | Int |  | NOT NULL | Внешний ключ |

Таблица 3 – Описание таблицы «Departments»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Departments** | | | | | | |
| **Имя поля** | **Id** | **Ключи** | **Тип данных** | **Размер** | **Допустимо ли NULL** | **Описание** |
| ID | **PK** |  | Int |  | NOT NULL | Первичный ключ |
| Name\_depart |  |  | Varchar | 100 |  | Название отделов |
| Heads\_of\_depart |  | **FK**  **(**Heads\_of\_departments.ID**)** | int |  | NOT NULL | Внешний ключ |

Таблица 4 – Описание таблицы «Heads\_of\_departments»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Heads\_of\_departments** | | | | | | |
| **Имя поля** | **Id** | **Ключи** | **Тип данных** | **Размер** | **Допустимо ли NULL** | **Описание** |
| ID | **PK** |  | Int |  | NOT NULL | Первичный ключ |
| fullname |  |  | Varchar | 100 | NOT NULL | Внешний ключ |

## **2.4 Описание программы**

**2.4.1 Наименование программы**

Полное наименование программной разработки – «База знаний для новых сотрудников компании»

**2.4.2 Описание логической структуры**

Структура проекта определена шаблоном MVC, следовательно, проект разделен на контроллер, модели и представления.

**2.4.3 Используемые технические средства**

В состав технических средств должен входить IВМ–совместимый компьютер, содержащий набор следующих компонентов:

− объем оперативной памяти 1Гигабайт, не менее;

− доступ к сети Интернет;

− процессор Pentium–2.0 или выше;

− свободного места на жестком диске не менее 100 Мб.

**2.4.4 Вызов и загрузка**

Вызовов программы осуществляете путем запуска локального сервера на своем компьютере и переход на страницу https:// localhost:8080/ в браузере.

**2.4.5 Результат работы Базы Знаний**

Результатом работы приложения представлены в Приложении А.

## **2.5 Тестирование Базы**

Были подготовлены 5 Test-Case, проверяющие поиск вопросов необходимые сотрудникам компании, а также вход в базу (Таблицы 5-10).

Таблица 5 – Аннотация теста

|  |  |
| --- | --- |
| **Название проекта** | База знаний |
| Рабочая версия | 1.0 |
| Имя тестирующего | Марченкова Ксения Игоревна |
| Дата(ы) теста | 14.05.2023 |

Таблица 6 – Вход в систему

|  |  |
| --- | --- |
| **Тестовый пример #** | TU01 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка входа пользователя с логином и паролем |
| Краткое изложение теста | Проверить войдет ли работник с логином и паролем на свою страницу |
| Этапы теста | Откройте сайт – Введите логин – Введите пароль – Нажмите кнопку «Войти» |
| Тестовые данные | Логин пользователя и пароль |
| Ожидаемый результат | Пользователь должен попасть на главную страницу |
| Фактический результат | Ожидаемо |
| Статус | Пройден успешно |
| Предварительное условие | Логин = 111 Пароль = 111 |
| Постусловие |  |
| Примечания/комментарии |  |

Таблица 7 – Проверка видимости готовых ответов

|  |  |
| --- | --- |
| **Тестовый пример #** | TU02 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка на видимость готовых ответов для пользователей |
| Краткое изложение теста | Проверить видимы ли готовые ответы для работников компании |
| Этапы теста | Откройте страницу Часто задаваемые вопросы – Найдите интересующий стандартный вопрос |
| Тестовые данные | Стандартный вопрос работника |
| Ожидаемый результат | Пользователь должен увидеть ответ на интересующий вопрос |
| Фактический результат | Ожидаемо |
| Статус | Пройден успешно |
| Предварительное условие | Просмотр стандартного вопроса для работника компании |
| Постусловие |  |
| Примечания/комментарии |  |

Таблица 8 – Поиск сотрудников

|  |  |
| --- | --- |
| **Тестовый пример #** | TU03 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Поиск Сотрудников компании |
| Краткое изложение теста | Видны ли данные о сотрудниках |
| Этапы теста | Поиск сотрудников |
| Тестовые данные | Данные сотрудников |
| **Тестовый пример #** | TU03 |
| Ожидаемый результат | Поиск выполнен |
| Фактический результат | Ожидаемый |
| Статус | Пройдено успешно |
| Предварительное условие | ФИО сотрудников |
| Постусловие |  |
| Примечания/комментарии |  |
|  |  |

Таблица 9 – Задать вопрос администратору

|  |  |
| --- | --- |
| **Тестовый пример #** | TU04 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Задать вопрос |
| Краткое изложение теста | Задать вопрос которого нету в базе знаний |
| Этапы теста | Зайти на страницу Обратная связь Нажать на кнопку «Задать вопрос» Написать, что вас интересует |
| Тестовые данные | Сколько работников в отделе разработок |
| Ожидаемый результат | Ответ от администратора |
| Фактический результат | Ожидаемый |
| Статус | Пройдено успешно |
| Предварительное условие | Знать, как задать вопрос |
| Постусловие |  |
| Примечания/комментарии |  |

Таблица 10 – Задать вопрос

|  |  |
| --- | --- |
| **Test Case #** | TU05 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Задать вопрос |
| Краткое изложение теста | Кнопка задать вопрос не работает |
| Этапы теста | Зайти на страницу Обратная связь Нажать на кнопку «Задать вопрос»  Выводит ошибку |
| Тестовые данные | Сколько работников в отделе разработок |
| Ожидаемый результат | Вывод ошибки после нажатия кнопки |
| Фактический результат | Ожидаемый |
| Статус | Пройдено успешно |
| Предварительное условие | Ошибка при нажатии |
| Постусловие |  |
| Примечания/комментарии |  |

## **3 Экономическая часть**

В экономической части определяются затраты на разработку и в целом капитальные затраты на разработку и внедрение Базы Знаний для сотрудников компании.

## **3.1 Технический и социальный эффект**

Целью данного дипломного проекта является разработка «Базы знаний для сотрудников компании».

Конечный продукт должен иметь следующее:

1. удобный интерфейс для работы пользователя;
2. удобное изучение вопросов;
3. просмотр неизвестных вопросов;
4. примеры заполнения документов.

## **3.2 Организационная часть**

Таблица 11 − Этапы и трудоёмкость разработки

| **Этапы работ** | **Содержание работ** | **Исполнители** | | | **Время выполнения , дни** |
| --- | --- | --- | --- | --- | --- |
| **Кол-во, чел.** | **Должность** | **Оклад, руб.** |
| Подготовительный | Ознакомление с заданием и выбор группы исполнителей | 1 | Техник-программист | 50000 | 2 |
| Подготовительный | Разработка технического задания | 1 | Техник-программист | 50000 | 2 |
| Подготовительный | Анализ методов решений | 1 | Техник-программист | 50000 | 2 |
| Разработка | Установка требуемого ПО | 1 | Техник-программист | 50000 | 6 |
| Разработка | Создание БД | 1 | Техник-программист | 50000 | 3 |
| Разработка | Создание интерфейса | 1 | Техник-программист | 50000 | 5 |
| Завершающий | Тестирование | 1 | Техник-программист | 50000 | 3 |
| Завершающий | Оформление документации | 1 | Техник-программист | 50000 | 3 |
| Всего (общая трудоёмкость выполнения разработки): | | | | | 26 |

В смету затрат включаются расходы:

* затраты на материалы, расходуемые в процессе проектирования;
* заработная плата работников, занятых разработкой;
* страховые взносы;
* прочие расходы (45% от заработной платы или по данным предприятия).

Затраты на материалы определяются в таблице 12.

Таблица 12 − Затраты на материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование материалов** | **Ед. изм.** | **Количество** | **Цена, руб.** | **Стоимость (без НДС), руб.** |
| Бумага формата А4 | пачка | 1 | 150 | 150 |
| Диск | шт. | 2 | 50 | 100 |
| Ручка | шт. | 3 | 110 | 110 |
| Флешка | шт. | 2 | 560 | 1120 |
| Итого затрат на материалы | | | | 1480 |
| Транспортные расходы по доставке материалов (15% от итого затрат на материалы). | | | | 222 |
| Всего затрат на материалы с учётом транспортных расходов по доставке | | | | 1702 |

Заработную плату можно определить по формулам 1, 2 (таблица 13).

Таблица 3 − Заработная плата работников

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Должность** | **Оклад, руб./мес.** | **Оплата в день, руб.** | **Трудоёмкость, дни** | **Основная заработная плата, руб.** |
| Техник-программист | 50000 | 1923,07 | 26 | 74999,73 |
| Итого основная заработная плата | | | | 74999,73 |
| Дополнительная заработная плата (12% от основной заработной платы) | | | | 8999,96 |
| Всего заработная плата (сумма основной и дополнительной заработной платы) | | | | 83999,7 |
| Страховые взносы (30% от заработной платы) | | | | 25199,91 |

Если разработкой занимается один разработчик, заработная плата разработчика (Зот) определяется по формуле:

Зот = Ор Тер (1+Кд), руб. (1)

где Ор – оклад разработчика, руб. в месяц;

ТЕр – трудоемкость проекта (т.е. время необходимое для разработки проекта), месяц;

Кд – коэффициент дополнительно заработной платы (принимается по данным предприятия).

Если трудоемкость задается в днях, то затраты по оплате труда определяются по формуле:

Зот = Ор Тер (1+Кд), руб. (2)

Зот = 1923,07 26 (1+0,5)=74999,73

Все полученные расчеты по разработке проекта сводятся в таблицу 4.

Таблица 14 − Смета затрат на разработку Базы знаний для предприятия «СПб ИАЦ»

|  |  |
| --- | --- |
| **Наименование затрат** | **Сумма, руб.** |
| Затраты на материалы | 1702 |
| Заработная плата | 83999,7 |
| Страховые взносы | 25199,91 |
| Прочие расходы | 26590,94 |
| Итого затрат | 137492,55 |

## **3.3 Экономическая часть**

В экономической части определяется себестоимость внедрения «Базы знаний для сотрудников компании».

В себестоимость внедрения проекта входят:

* затраты на техническое и программное обеспечение;
* заработная плата работников, занятых внедрением;
* страховые взносы.

Затраты на техническое и программное обеспечение определяются аналогично затратам на материалы в таблице 15.

Таблица 15 − Затраты на техническое и программное обеспечение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Ед. изм.** | **Расход** | **Цена, руб.** | **Стоимость, руб.** |
| Лицензия MS Office 2019 | Шт. | 1 | 5490 | 5490 |
| Виртуальный выделенный сервер | Шт. | 1 | 900 | 900 |
| Итого (общая стоимость) | | | | 6390 |
| Затраты по доставке (10% от общей стоимости) | | | | 639 |
| Итого затрат с учётом доставки | | | | 7029 |

Все полученные расчеты по внедрению проекта представлены в таблице 16.

Таблица 16 – Смета затрат на внедрение

|  |  |
| --- | --- |
| **Наименование затрат** | **Сумма, руб.** |
| Затраты на техническое и программное обеспечение | 7029 |
| Заработная плата | 1923,07 |
| Страховые взносы | 576,92 |
| Прочие расходы | 8051,73 |
| Итого затрат | 17580,73 |

## **3.4 Определение цены проекта для Базы Знаний для новых сотрудников компании для предприятия ГУП СПб «СПб ИАЦ»**

Цена разработки программного продукта с учётом затрат на внедрение проекта определяется по формуле:

Цпп = Зобщ + Пп + НДС, (3)

где Зобщ – сумма общих затрат по разработке и внедрению проекта;

Пп – планируемая прибыль (принимается≥25% от суммы затрат на разработку и внедрение проекта руб.);

НДС – налог на добавленную стоимость (ставка НДС = 20%), руб.

ЦПП = (137492,55+ 17580,73) + 37011,07 + 31014,6 = 223098,95 руб.

Планируемая прибыль может быть рассчитана исходя из нормативной рентабельности программного продукта, определяемой организацией и рассчитывается по формуле:

Пп = Зобщ Rп, руб. (4)

где Rп – коэффициент рентабельности, определенный предприятием.

Пп = 137492,55 0,15 = 20623,9 руб.

Налог на добавленную стоимость определяется по ставке, установленной государством от суммы затрат на разработку программного продукта и величины плановой прибыли по формуле:

НДС = (Зобщ + Пп)Кндс, руб. (5)

где Зобщ – ­сумма общих затрат по разработке и внедрению проекта.;

Пп – планируемая прибыль руб.;

Кндс – коэффициент НДС (Кндс = 0,2 при ставке НДС = 20%).

НДС = (137492,55 + 37011,07) 0,2 = 34900,7 руб.

Цену программного продукта можно определить также и по формуле:

Цпп = Зощ ,руб.

В результате полученного положительного финансового результата (прибыли) предприятие должно начислить налог на прибыль (ставка налога-20% от прибыли) по формуле:

НП = Пп, (6)

где Пп – планируемая прибыль руб.

НП = 37011,07 0,2 = 7402,2 руб.

Прибыль, оставшаяся на предприятии после уплаты налога на прибыль, определяется по формуле:

Пч = Пп НП, руб. (7)

где Пп – планируемая прибыль руб.;

НП − налог на прибыль.

Пч = 37011,07 – 7402,2 = 29608,9 руб.

В целом поступления в бюджет от реализации программного продукта составят:

БП = НП + НДС, руб. (8)

где НП − налог на прибыль;

НДС – налог на добавленную стоимость (ставка НДС = 20%), руб.

Бп = 7402,2 + 31014,6 = 38416,8 руб.

Вывод: затраты на разработку программного продукта составили 137492,55 руб. при использовании на данном предприятии. Для предприятия СПб ГУП «СПб ИАЦ» это приемлемая сумма затрат, которая в перспективе должна полностью себя окупить благодаря разработанному продукту. Если появится возможность реализации продукта, его цена будет составлять 223098,95 руб.

## **4 Техника безопасности и охрана труда**

**4.1 Анализ условий труда техника-программиста, работающего в «СПб ИАЦ»**

Рабочее место техника – программиста – это часть помещения, где он проводит большую часть рабочего времени. Совокупность показателей технических и программных средств определяет качество и производительность работы, наряду с габаритными показателями мебели и её удобством.

Программист имеет собственное рабочее место, оборудованное ПЭВМ. Рабочий стул (кресло) должен быть подъёмно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также – расстоянию спинки до переднего края сиденья.

Помещение для работы представляет собой кабинет длиной 3 м, шириной 4 м и высотой потолков 2.7 м. Рабочая поверхность стола сделана из дерева. На столе размещены ЭВМ, монитор и принтер. Площадь поверхности стола предполагает размещение монитора, устройств ввода/вывода и рабочей зоны с местом расположения оперативной документации.

Кабинет, в котором расположено рабочее место, представляет собой помещение площадью 18,02 м2, длиной 5,3 м, шириной 3,4 м и высотой 2,3 м.

Для создания благоприятных условий для зрительного восприятия кабинет оформлен в светло-бежевом цвете.

Для создания и поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты воздуха, в холодное время используется водяное отопление, в тёплое время года применяется кондиционирование воздуха.

Режим работы программиста - односменный с пятидневной рабочей неделей и ненормированным рабочим днём. Продолжительность ежегодного отпуска - 28 календарных дней.

Для снижения нагрузки на технические средства, влекущей за собой перегрев компьютера и ноутбука, используются программные средства, позволяющие снизить нагрузку на технические средства за счёт отказа в использование ненужных служб.

При идентификации вредных производственных факторов было установлено, что опасными могут являться:

* электромагнитные поля и излучения от технических средств;
* статическое электричество;
* статические перегрузки, вызываемые неподвижной работой за компьютером;
* перенапряжение органов зрения.

В целях выявления вредных и (или) опасных факторов производственной среды и трудового процесса, оценке уровня их воздействия на работника с учетом отклонения их фактических значений от установленных уполномоченным Правительством Российской Федерации федеральным органом исполнительной власти гигиенических нормативов условий труда и применения средств индивидуальной и коллективной защиты работниковпроводится специальная оценка условийна рабочих местах. Оценке подлежат все имеющиеся в организации рабочие места.

Работа по специальной оценке условий труда заключается в исследовании следующих факторов на рабочих местах организации:

* химические факторы;
* физические факторы;
* биологические факторы;
* тяжесть и напряженность трудового процесса.

Для уменьшения нагрузки на органы зрения, а также снятия мышечного напряжения во время рабочего дня проводится проветривание в течении 10-15 минут каждые 3 часа.

Согласно санитарным правилам СП 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда", дисплей должен располагаться на расстоянии 60-70 см, но не менее 50 см от глаз. Между боковыми поверхностями мониторов должно быть не менее 1,2 м. При использовании жидкокристаллических дисплеев на каждый компьютер должно приходиться не менее 4,5 м2 площади. На дисплей ПК не должен попадать прямой солнечный свет во избежание бликов и повышенной нагрузки на зрение.

Поэтому в кабинете, где работает программист, стоит только один компьютер, а комплектующая оргтехника расположена достаточно далеко непосредственно от самого специалиста. Это обусловлено тем, что все копировальные аппараты, принтеры и факс в периоды своей работы выделяют не только вредные токсичные газы, но и шумы и вибрацию. Данные факторы оказывают гораздо меньшее негативное влияние, нежели излучение от компьютера, но также являются вредными и могут нанести вред здоровью сотрудника.

Для снижения напряжения с глаз, а также для снятия мышечного напряжения во время рабочего дня проводится проветривание в течении 10 – 15 минут каждые 1,5 часа.

**4.2 Расчёт искусственного освещения в** **Расчёт искусственного освещения в помещении «СПб ИАЦ», где находится рабочее место программиста**

Через зрительный анализатор человек получает около 80% из общего объема информации. Качество поступающей, в основном, зависит от освещения в помещении. Если оно неудовлетворительное, это приводит к утомлению организма в целом. Кроме того, нерациональное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции. Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места техник - программиста являются:

* помещение размером 18,02 квадратных метров;
* высота помещения 2,3 метров, длина 5,3 метра, ширина 3,4 метра;
* лампы люминесцентные (дневного света) в количестве 5 штук, новые;
* крупногабаритная мебель отсутствует.

Основным критерием, по которому определяется необходимое количество осветительных приборов, является нормируемый уровень освещенности.

Базовым методом проектирования осветительной установки является метод коэффициентов использования, позволяющий вручную проводить все вычислительные процедуры при решении относительно простых светотехнических задач.

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента использования светового потока. Световой поток (лм) одной лампы рассчитывается по следующей формуле:

 (9)

где Ен – нормируемая минимальная допустимая освещённость по СНиП 23-05-95, для помещения 2000 лк;

S – площадь освещаемого помещения, м2;

z – коэффициент неравномерности освещения, z = 1,1;

K – коэффициент запаса, учитывающий снижение освещённости в процессе эксплуатации вследствие загрязнения и старения ламп и светильников, а также снижения отражающих свойств поверхностей помещения для кабинета будет равен 1,3;

Nc – число светильников в помещении;

– коэффициент затенения, обычно 1;

=i – коэффициент использования светового потока.

Коэффициент использования светового потока, давший название методу расчёта, определяется по индексу помещения i в зависимости от типа светильника и коэффициентов отражения света от потолка, стен и пола:

Индекс помещения:

 (10)

где А, В – длина и ширина помещения в плане, м.;

Нс – высота подвеса светильников над рабочей поверхностью, м.

Полученное количество светильников совпадает с числом светильников, установленных в помещении, что соответствует нормативным показателям искусственного освещения согласно СП 52.13330.2016 «Естественное и искусственное освещение».

Согласно СНиП 23-05-95 величина полученного светового потока достаточно для данного помещения.

**4.3** **Электробезопасность в «СПб ИАЦ»**

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности и электробезопасности. Работника знакомят с основными правилами по технике безопасности, предлагают внимательно прочитать действующие на предприятии инструкции, поясняя при этом отдельные правила и требования.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 1 раза в 6 месяцев по программе инструктажа на рабочем месте.

При нарушении работающими правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Все инструктажи оформляются записями в специальном журнале с указанием номеров или шифров инструкций. Журнал о проведении инструктажа хранится у руководителя подразделения.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

* прикасаться к задней панели системного блока при включенном питании;
* допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
* производить самостоятельно вскрытие и ремонт оборудования;
* пользоваться неисправными электроприборами и электропроводкой;
* ремонтировать электроприборы самостоятельно;
* подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки;
* прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т.п.);
* пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью;
* наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения):

* Автоматические выключатели и электрические предохранители должны быть всегда исправны;
* Изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии;
* Электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений;
* Для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.

**4.4** **Пожарная безопасность в «СПб ИАЦ»**

Инструктаж по пожарной безопасности проводится по программе, разработанной инженером по охране труда «СПб ИАЦ», с учётом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утверждённой программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учёта проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 ч. Инструктируемые должны ознакомиться:

* с действующими на объекте правилами пожарной безопасности и инструкциями;
* с производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
* с возможными причинами возникновения пожара и мерами его предупреждения;
* с практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т.д.).

## **Заключение**

В данной работе был разработан и реализован проект веб-приложения для компании, а также рассмотрены методы автоматизации и оптимизации процесса создания информационного приложения в сети интернет. Был разработан удобный и понятный, при эксплуатации пользователем, интерфейс.

При разработке были исследованы потенциальные проблемы и произведен анализ аналогичных существующих решений, было приведено обоснование предлагаемого решения.

В результате получены следующие результаты:

проведено объектно-ориентированное проектирование системы, построены UML диаграммы этапа структурирования;

повторены и обобщены знания при работе с реляционными базами данных MS SQL;

разработана база данных MS SQL для заданной предметной области системы;

изучение языка программирования C# в частности;

определен интерфейс веб-приложения для работы с созданной базой данных.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. – СПб. : Питер, 2019. – 896 с
2. Советы по формированию базы знаний для сотрудников компании - [Электронный ресурс] - <https://yagla.ru/blog/samorazvitie/sovety-po-formirovaniyu-bazy-znaniy-dlya-sotrudnikov-kompanii--2212u99870/>
3. Зачем нужна корпоративная база знаний и как ее создать? [Электронный ресурс] - https://personik.ai/blog/knowledge\_base.
4. IT База Знаний в вашей компании [Электронный ресурс] - <https://habr.com/ru/articles/153575/>
5. Как создать внутреннюю базу знаний для IT-компании? [Электронный ресурс] - <https://vc.ru/life/603886-kak-sozdat-vnutrennyuyu-bazu-znaniy-dlya-it-kompanii-eksperiment-chast-1>
6. Базы знаний [Электронный ресурс] - <https://catalog.arppsoft.ru/section/6101225>.
7. UML – [Электронный ресурс] / <https://www.bestreferat.ru/referat-142715.html>
8. Магнадуров, Г. И. ASP.NET MVC Framework / Г. И. Магнадуров, В. А. Юнев. – СПб. : БХВ-Петербург, 2019. – 300 с
9. Фримен, А. Entity Framework Core 2 для ASP.NET Core MVC для профессионалов / А. Фримен. – М. : Диалектика-Вильямс, 2019. – 624 c
10. Как создать базу знаний, которая поможет компании развиваться [Электронный ресурс] - <https://infoservice.ru/blog/knowledge-base/kak-sozdat-bazu-znaniy-kotoraya-pomozhet-kompanii-razvivatsya/>
11. Knowledge Space [Электронный ресурс] - <https://im.systems/>
12. Документация для построения диаграммы последовательности [Электронный ресурс] / https://intuit.ru/studies/courses/32/32/lecture/1014
13. Общая информация о диаграмме состояния – [Электронный ресурс] / <https://studopedia.ru/11_96547_ponyatie-o-diagrammah-sostoyaniya.html>
14. Байер, Д. Microsoft ASP .NET. Обеспечение безопасности / Д. Байер. – М. : Русская Редакция, 2018. – 430 с.
15. Роббинс, Д. Н. HTML5. Карманный справочник / Д. Н. Роббинс. – М. : Вильямс, 2019. – 192 с.

## Приложение А

(справочное)

**Результаты работы**

При открытии приложения запускается главная страница, изображенная на рисунке А.1.

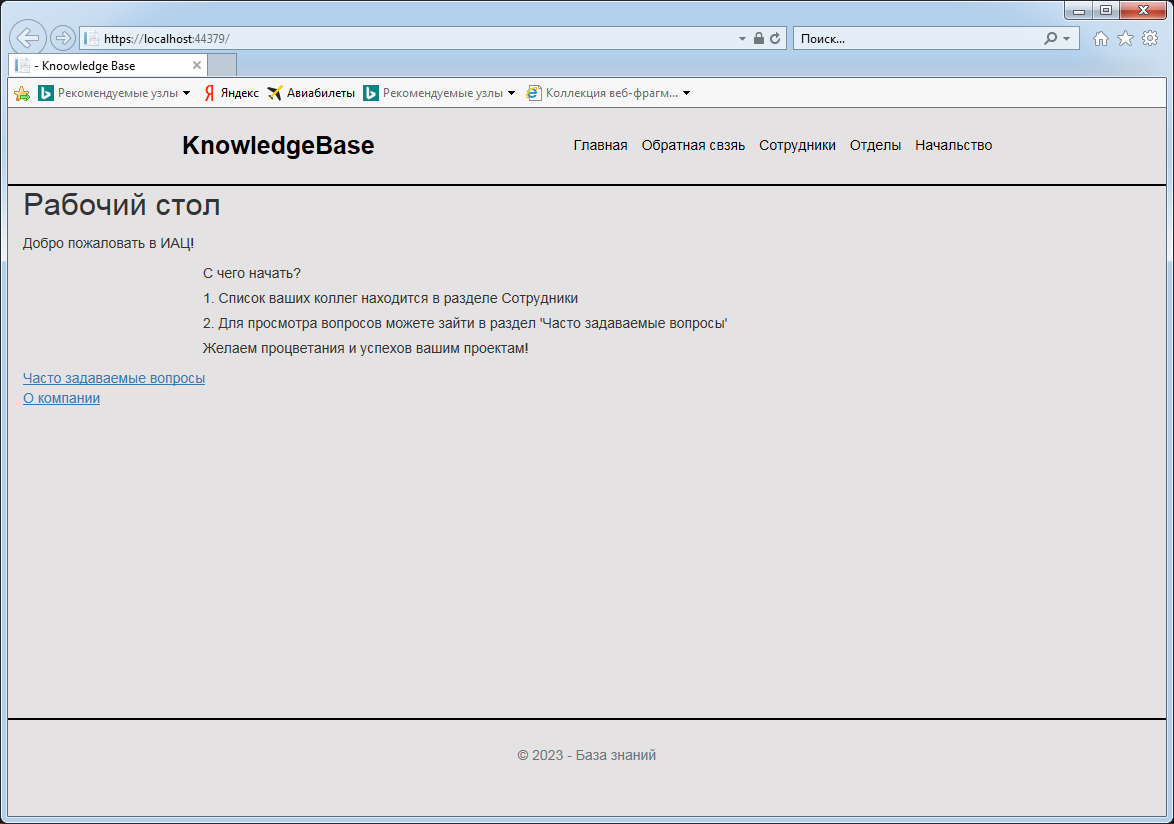


Рисунок А.1 – Главная страница

При переходе в Базу знаний открывается страница, где находятся все вопросы, интересующие сотрудников, как изображено на рисунке А.2, а также поиск нужной информации, изображено на рисунке А.3.

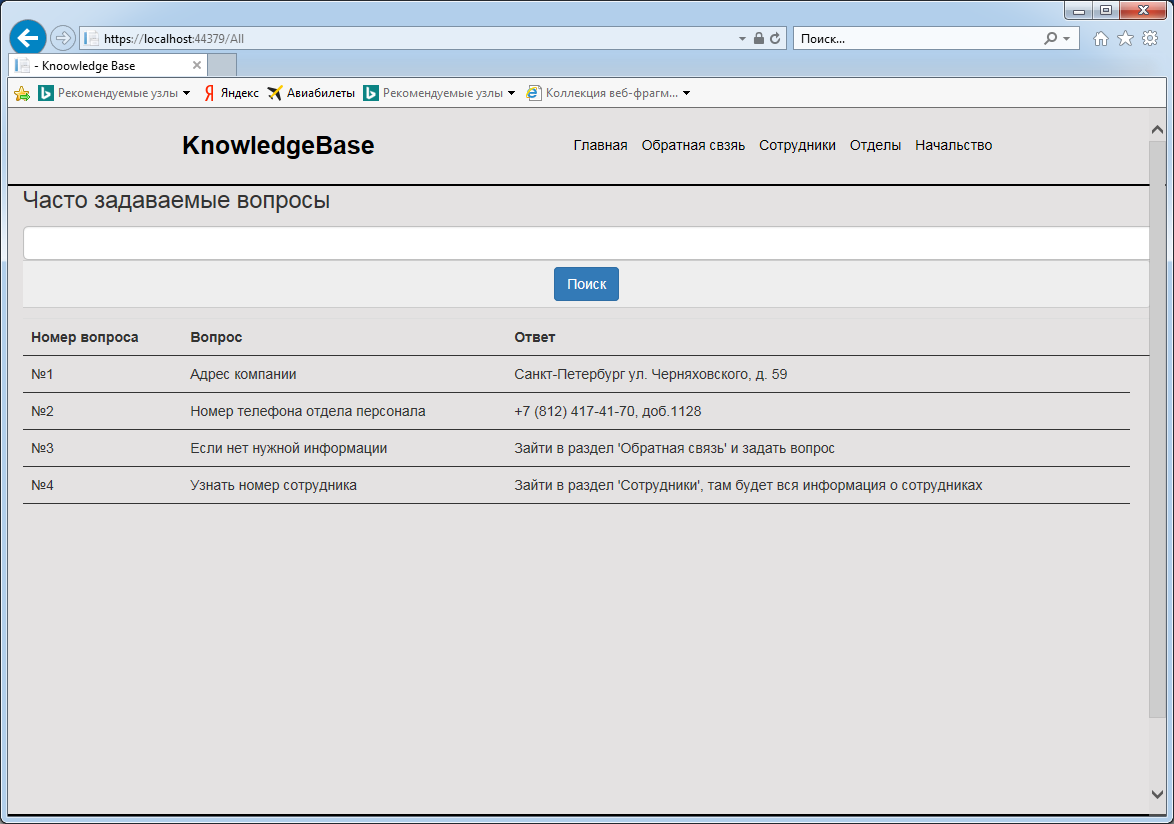


Рисунок А.2 – База знаний

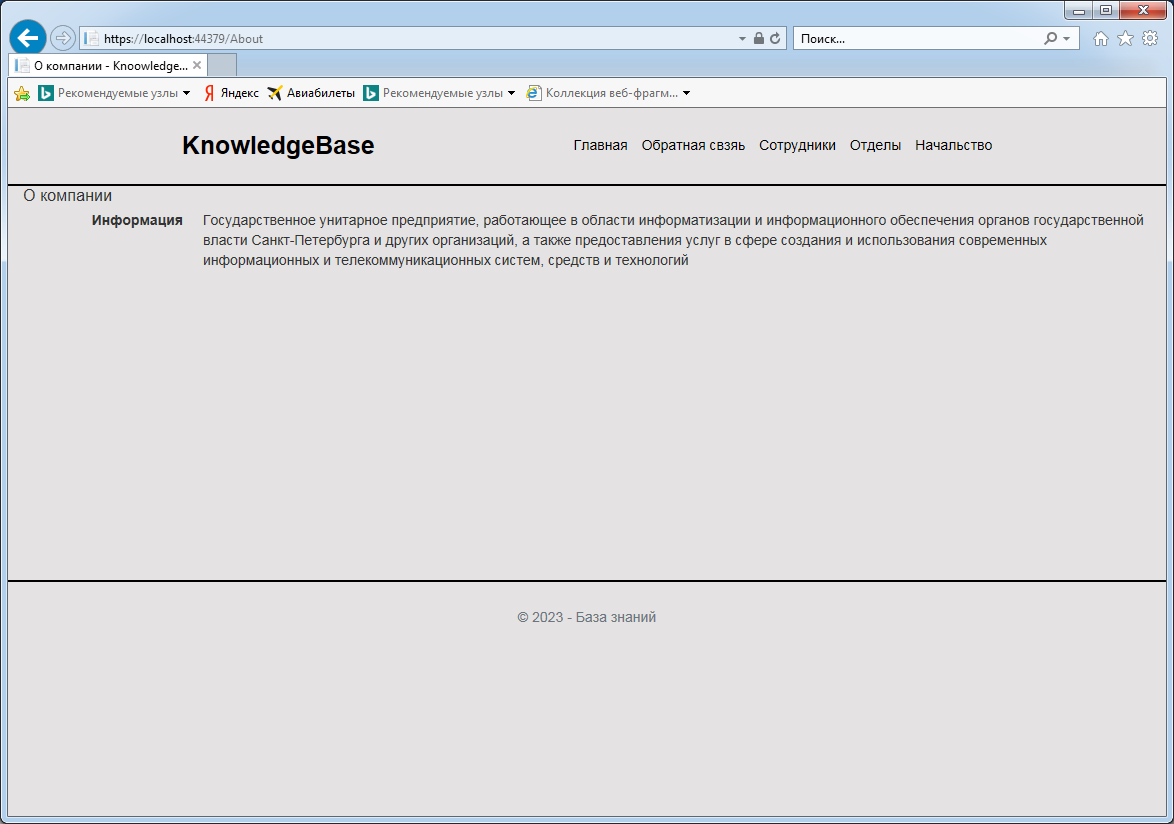


Рисунок А.3 – Информация О компании

Также существует обратная связь, куда может обратиться сотрудник компании, если не нашел нужной информации, показано на рисунке А.4.

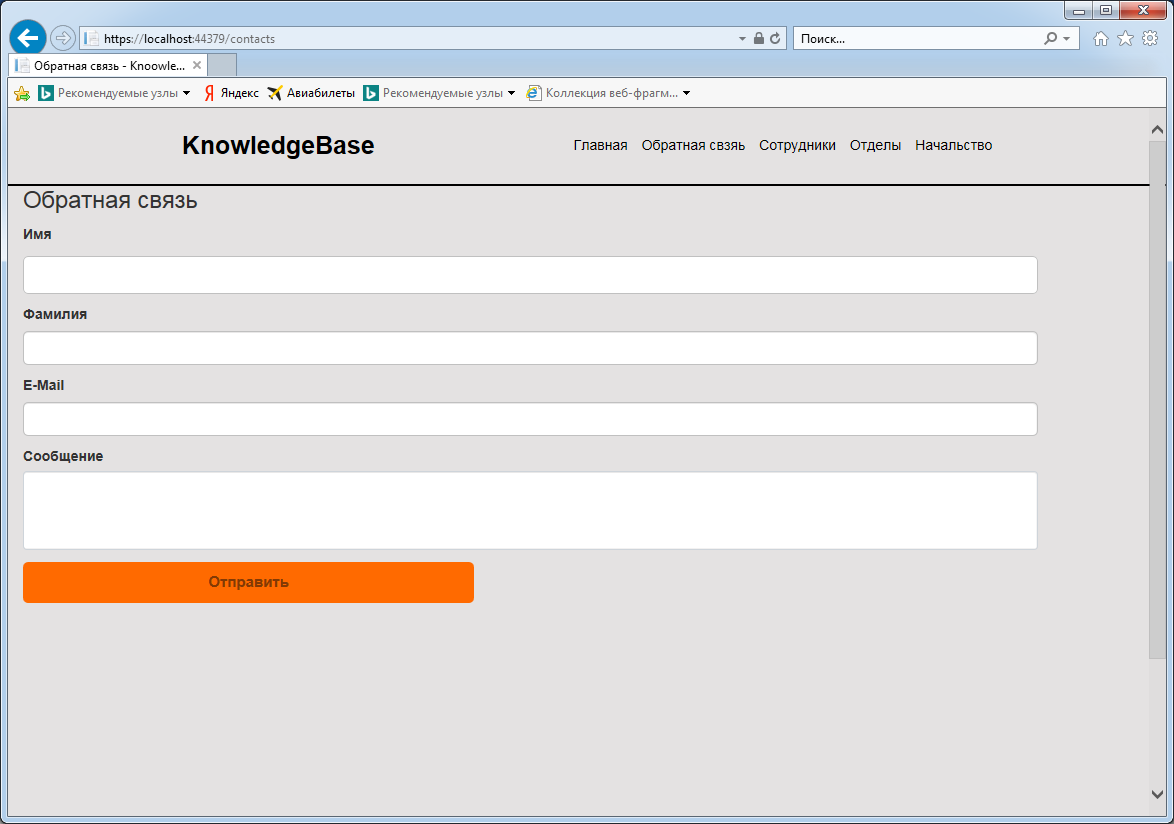


Рисунок А.4 – Обратная связь

При переходе на страницу «Сотрудники» можно увидеть данные о сотрудниках, работающих в компании, изображено на рисунке А.5.

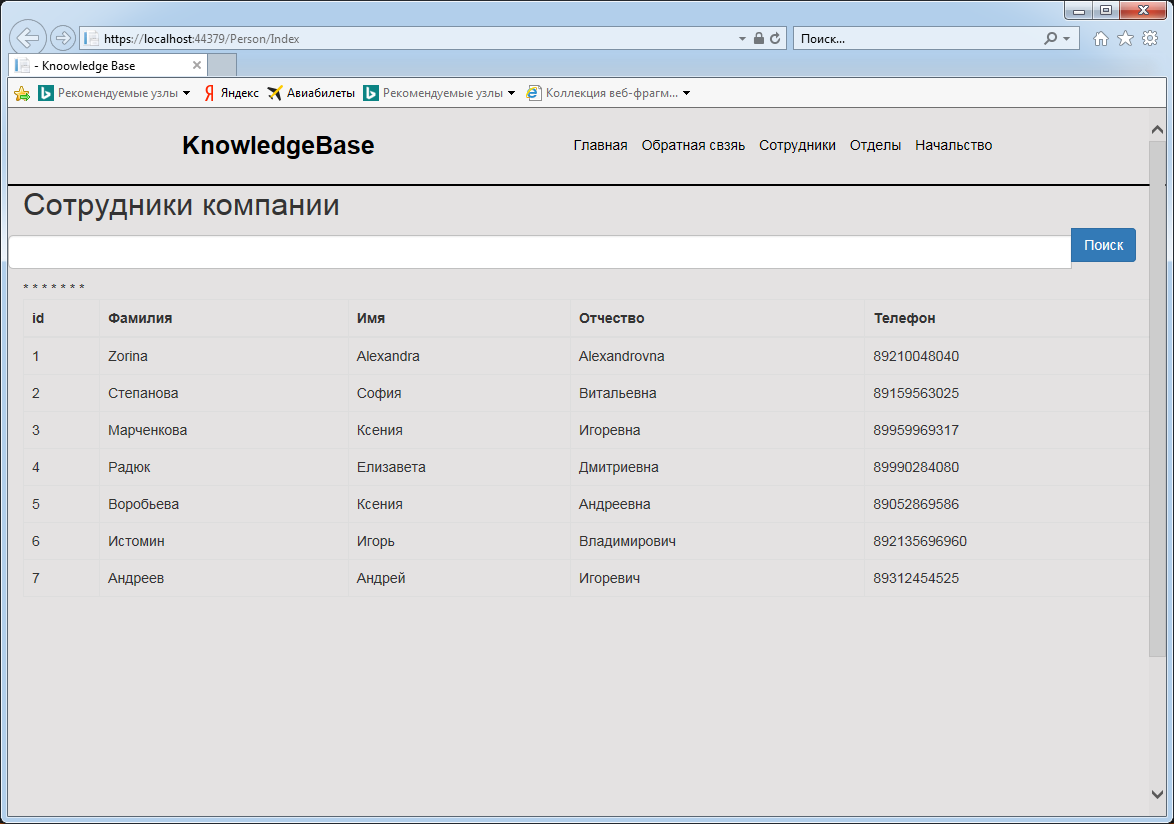
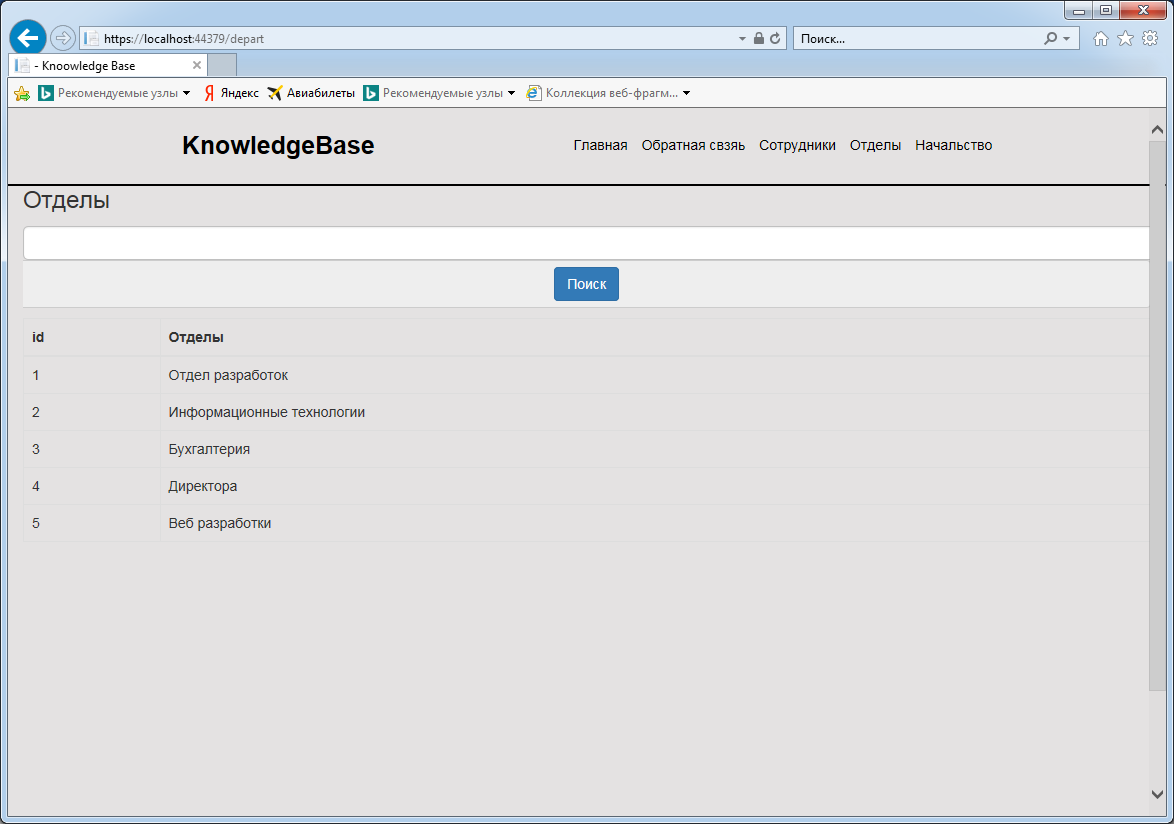
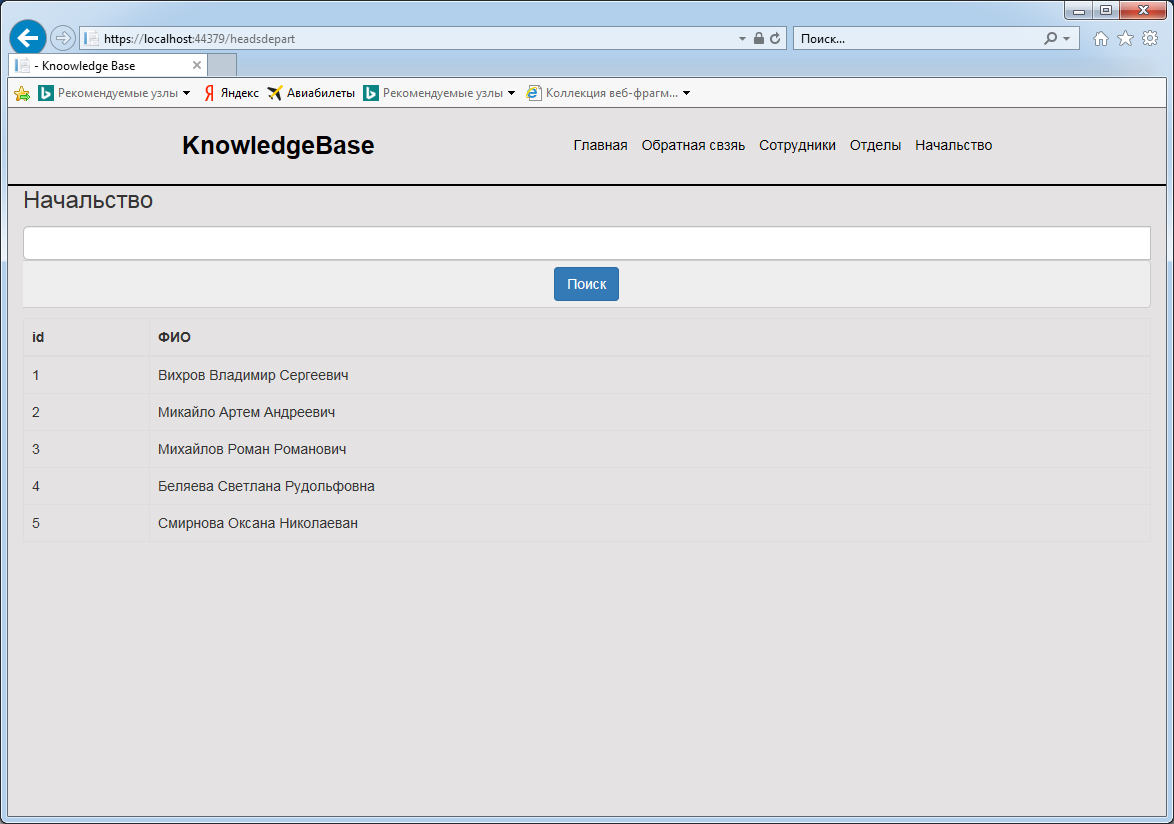


Рисунок А.5 – Сотрудники

При переходе на страницу «Отделы» можно увидеть данные об отделах, которые присутствуют в компании, изображено на рисунке А.6.

Рисунок А.6 – Отделы

При переходе на страницу «Начальство» можно увидеть данные об отделах, которые присутствуют в компании, изображено на рисунке А.7.

Рисунок А.7 – Начальство

## Приложение Б

(справочное)

**Исходный код**

**RequireBasicAuthentication:**

using System;

using System.Web;

using MySql.Data.MySqlClient;

using System.Web.Mvc;

using Diplom1.Components;

namespace Diplom1.Controllers.Filter

{

public class RequireBasicAuthentication : ActionFilterAttribute

{

private readonly MySqlConnection \_connection = ConnectionManager.GetConnection();

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

var request = filterContext.HttpContext.Request;

if (String.IsNullOrEmpty(request.Headers["Authorization"]))

{

filterContext.HttpContext.Response.AddHeader("WWW-Authenticate", "Basic realm=\"Knowledge base\"");

filterContext.Result = new HttpUnauthorizedResult();

}

else

{

var authorization = request.Headers["Authorization"];

var credentials = System.Text.Encoding.UTF8.GetString(

Convert.FromBase64String(authorization.Substring(6)))

.Split(':');

var user = new { Username = credentials[0], Password = credentials[1] };

var command = new MySqlCommand("select \* from vxod v, staff s where v.password = '" + user.Password + "' AND name = '" + user.Username + "'", \_connection);

using (var reader = command.ExecuteReader())

{

if (reader.HasRows)

{

// Get current user - HttpContext.Session["Usernaem"]

filterContext.HttpContext.Session["Username"] = user.Username;

// Get current user - HttpContext.Session["Role"]

filterContext.HttpContext.Session["Role"] = user.Username;

}

else

{

filterContext.HttpContext.Response.AddHeader("WWW-Authenticate", "Basic realm=\"Knowledge base\"");

filterContext.Result = new HttpUnauthorizedResult();

}

}

}

}

}

}

**Models**

**Base:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace Diplom1.Models

{

public class Base

{

[Display(Name = "Номер вопроса")]

public string Question { get; set; }

[Display(Name = "Вопрос")]

public string Question1 { get; set; }

[Display(Name = "Ответ")]

public string Answer { get; set; }

public string About { get; set; }

public string Info { get; set; }

public string greeting { get; set; }

public string Answer2 { get; set; }

public string Answer3 { get; set; }

public string Answer4 { get; set; }

}

}

**Contact:**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace Diplom1.Models

{

public class Contact

{

[Display(Name = "Введите имя")]

[Required(ErrorMessage = "Вам нужно ввести имя")]

public string Name { get; set; }

[Display(Name = "Введите фамилию")]

[Required(ErrorMessage = "Вам нужно ввести фамилию")]

public string Surname { get; set; }

[Display(Name = "Введите почту")]

[Required(ErrorMessage = "Вам нужно ввести почту")]

public string Email { get; set; }

[Display(Name = "Введите сообщение")]

[Required(ErrorMessage = "Вам нужно ввести сообщение")]

[StringLength(120, ErrorMessage = "Текст не менее 120 символов")]

public string Message { get; set; } } }

**Depart:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Diplom1.Models

{

public class Depart

{

public int Id { get; set; }

public string Name\_Depart { get; set; }

}

}

**HeadsDepart:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Diplom1.Models

{

public class HeadsDepart

{

public int Id { get; set; }

public string FullName { get; set; }

}

}

**Person:**

using System.Collections.Generic;

namespace Diplom1.Models

{

public class Person

{

public string SerchTerm { get; set; }

public int Id { get; set; }

public string Surname { get; set; }

public string Name { get; set; }

public string Otchestvo { get; set; }

public string Telephone { get; set; }

public string Peronsearch { get; set; }

}

}

**Controller**

**BasesController:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Diplom1.Components;

using Diplom1.Models;

using MySql.Data.MySqlClient;

namespace Diplom1.Controllers

{

public class BasesController : Controller

{

// GET: Bases

public ActionResult About()

{

var bases = new Base

{

About = "О компании",

Info = "Государственное унитарное предприятие, работающее в области информатизации и информационного обеспечения органов государственной власти Санкт-Петербурга и других организаций, а также предоставления услуг в сфере создания и использования современных информационных и телекоммуникационных систем, средств и технологий"

};

return View("About", bases);

//return new ViewResult { };

}

public ActionResult Index()

{

var bases = new Base

{

greeting = "Добро пожаловать в ИАЦ!",

Question = "Вопрос-ответ",

Question1 = "С чего начать?",

Answer = "1. Список ваших коллег находится в разделе Сотрудники" ,

Answer2 = "2. Для просмотра вопросов можете зайти в раздел 'Часто задаваемые вопросы'",

Answer3 = "Желаем процветания и успехов вашим проектам!"

};

return View("Index", bases);

//return new ViewResult { };

}

public ActionResult All()

{

var basess = new List<Base>

{

new Base

{

Question = "№1",

Question1 = "Адрес компании",

Answer = "Санкт-Петербург ул. Черняховского, д. 59"

},

new Base

{

Question = "№2",

Question1 = "Номер телефона отдела персонала",

Answer = "+7 (812) 417-41-70, доб.1128"

},

new Base

{

Question = "№3",

Question1 = "Если нет нужной информации",

Answer = "Зайти в раздел 'Обратная связь' и задать вопрос "

},

new Base

{

Question = "№4",

Question1 = "Узнать номер сотрудника",

Answer = "Зайти в раздел 'Сотрудники', там будет вся информация о сотрудниках"

},

};

return View("All", basess);

}

}

}

**ContactsController:**

using Diplom1.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Diplom1.Controllers

{

public class ContactsController : Controller

{

// GET: Contacts

public ActionResult Index() //contacts

{

return View();

}

[HttpPost]

public ActionResult Check(Contact contact)

{

if (ModelState.IsValid)

{

return Redirect("/");

}

return View("Index");

}

}

}

**DepartController:**

using Diplom1.Components;

using Diplom1.Models;

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

using Diplom1.Controllers.Filter;

namespace Diplom1.Controllers

{

public class DepartController : Controller

{

// GET: Person

private readonly MySqlConnection \_connection = ConnectionManager.GetConnection();

// GET: Depart

[RequireBasicAuthentication]

public ActionResult Index()

{

// Example of getting current username

var user = HttpContext.Session["Username"];

//-----------------------------------

var depart = new List<Depart>();

var command = new MySqlCommand("select \* from departments", \_connection);

using (var reader = command.ExecuteReader())

{

while (reader.Read())

{

var departt = new Depart

{

Id = Convert.ToInt32(reader["id"]),

Name\_Depart = reader["name\_depart"].ToString(),

};

depart.Add(departt);

}

}

return View(depart);

}

}

}

**HeadsDepartController:**

using Diplom1.Components;

using Diplom1.Controllers.Filter;

using Diplom1.Models;

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Diplom1.Controllers

{

public class HeadsDepartController : Controller

{

// GET: Person

private readonly MySqlConnection \_connection = ConnectionManager.GetConnection();

// GET: Depart

[RequireBasicAuthentication]

public ActionResult Index()

{

// Example of getting current username

var user = HttpContext.Session["Username"];

//-----------------------------------

var headsdepart = new List<HeadsDepart>();

var command = new MySqlCommand("select \* from heads\_of\_depart ", \_connection);

using (var reader = command.ExecuteReader())

{

while (reader.Read())

{

var headsdepartt = new HeadsDepart

{

Id = Convert.ToInt32(reader["id"]),

FullName = reader["fullname"].ToString(),

};

headsdepart.Add(headsdepartt);

}

}

return View(headsdepart);

}

}

}

**PersonController:**

using Diplom1.Components;

using Diplom1.Controllers.Filter;

using Diplom1.Models;

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

namespace Diplom1.Controllers

{

public class PersonController : Controller

{

private readonly PersonService \_personService;

public PersonController()

{

\_personService = new PersonService();

}

// GET: Person

private readonly MySqlConnection \_connection = ConnectionManager.GetConnection();

[RequireBasicAuthentication]

public async Task<ActionResult> Index()

{

// Example of getting current username

var user = HttpContext.Session["Username"];

//-----------------------------------

var person = new List<Person>();

var command = new MySqlCommand("select \* from employees", \_connection);

using (var reader = command.ExecuteReader())

{

while (reader.Read())

{

var employees = new Person

{

Id = Convert.ToInt32(reader["id"]),

Surname = reader["surename"].ToString(),

Name = reader["name"].ToString(),

Otchestvo = reader["otchestvo"].ToString(),

Telephone = reader["telephone"].ToString()

};

person.Add(employees);

}

}

return View(person);

}

public ActionResult AddPerson()

{

return View();

}

public ActionResult EditPerson()

{

return View();

}

public async Task<ActionResult> DeletePerson(int? id)

{

using (MySqlConnection connect = new MySqlConnection(ConnectionManager.strConnect))

if (id != null)

{

string sql = "Delete from 'employeeys' Where'Id'= @Id";

using (MySqlCommand cmd = new MySqlCommand(sql, connect))

{

cmd.Parameters.Add("Id", MySqlDbType.Int32).Value = id;

connect.Open();

return View();

}

}

return RedirectToAction("Index");

}

}

}

**View**

**Bases(Index):**

@model Diplom1.Models.Base

<h1>Рабочий стол</h1>

<p style="font-family:'Times New Roman', sans-serif">

<dl class=" dl-horizontal">

<dd>@Model.greeting</dd>

</dl>

</p>

<dl class=" dl-horizontal">

<dd>@Model.Question1</dd>

<dd>@Model.Answer</dd>

<dd>@Model.Answer2</dd>

<dd>@Model.Answer3</dd>

</dl>

@\*<a href="/Bases/All">Часто задаваемые вопросы<</a>\*@

@Html.ActionLink("Часто задаваемые вопросы", "All")

<dl class=" dl-horizontal">

@Html.ActionLink("О компании", "About")

</dl>

**Bases(About):**

@model Diplom1.Models.Base

@{

ViewBag.Title = "О компании";

}

<h4>@Model.About</h4>

<dl class="dl-horizontal">

<dt>Информация</dt>

<dd>@Model.Info</dd>

</dl>

**Bases(All):**

@model IEnumerable<Diplom1.Models.Base>

<h2>Часто задаваемые вопросы</h2>

<form method="get" class="mb-3">

<div class=" input-group">

<input class="form-control" asp-for="SerchTerm" />

<div class="input-group-addon">

<button type="submit" class="btn btn-primary">Поиск</button>

</div>

</div>

</form>

<table class="table">

<tr>

<th>Номер вопроса

@\*@Html.DisplayNameFor(model => model.Question)\*@

</th>

<th>Вопрос

@\*@Html.DisplayNameFor(model => model.Question1)\*@

</th>

<th>Ответ

@\*@Html.DisplayNameFor(model => model.Answer)\*@

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Question)

</td>

<td>

@Html.DisplayFor(modelItem => item.Question1)

</td>

<td>

@Html.DisplayFor(modelItem => item.Answer)

</td>

</tr>

}

</table>

**Contacts(Index):**

@model Diplom1.Models.Contact

@{

ViewBag.Title = "Обратная связь";

}

<h2>Обратная связь</h2>

<form method="post" asp-cotroller="Contacts" asp-action="Check">

<label asp-for="Name" for="Name">Имя</label>

<input asp-for="Name" />

<span asp-validation-for="Name"></span>

<label asp-for="Surame" for="surname">Фамилия</label>

<input asp-for="Surame" id="surname" class="form-control"/>

<span asp-validation-for="Surame"></span>

<label asp-for="Email" for="email">E-Mail</label>

<input asp-for="Email" id="email" class="form-control"/>

<span asp-validation-for="Email"></span>

<label asp-for="Messege" for="message">Сообщение</label>

<textarea asp-for="Messege" id="message" class="form-control" style="width: 90%" rows="6"></textarea>

<span asp-validation-for="Messege"></span>

<input type="submit" value="Отправить" />

</form>