**ТИТУЛЬНЫЙ ЛИСТ**

**РЕФЕРАТ**

**Путеева, А.С.** Программное средство поддержки процесса организации вебинаров для взаимодействия корпоративных сотрудников: пояснительная записка к дипломному проекту / А.С. Путеева. – Минск: БГУИР, 2020. – 94с.

Пояснительная записка 94 с., 50 рис., 8 табл., 50 источников, 3 приложения

ПРОГРАММНОЕ СРЕДСТВО, ОРГАНИЗАЦИЯ ВЕБИНАРОВ, ПРЕДПРИЯТИЕ, автоматизация, бизнес-ПРОЦЕССами, ПРОЕКТНАЯ ЧАСТЬ, РАСЧЕТ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ.

*Цель* *проектирования*: разработка программного средства поддержки процесса организации вебинаров для взаимодействия корпоративных сотрудников, которое позволит отделу повышения квалификации, отделу кадров и руководителям подразделений уменьшить временные и денежные затраты на управление повышения подготовки персонала.

*Методология проведения работы*: в процессе решения поставленных задач использованы принципы системного подхода, нотация *IDEF*0 для функционального моделирования, графический язык *UML* для проектирования программных модулей.

*Результаты работы*: разработано программное средство поддержки процесса организации вебинаров для взаимодействия корпоративных сотрудников, которое позволяет отделу кадров и руководителям подразделений эффективно управлять организовывать вебинары, управлять уже созданными, а также просматривать статистику посещений пользователями по каждому из них. Пользователям будет легче определиться с выбором, используя критерии отображения и информацию о каждом вебинаре; произведено моделирование процессов проектируемой системы; уделено внимание вопросам технико-экономического обоснования; разработана графическая часть проекта.

Разработанная система позволяет администратору системы просматривать заявки на создание вебинаров от пользователей, создавать и одобрять вебинары, изменять их статус. Имеется возможность просматривать статистику по каждому посещенному вебинару, общую статистику категорий, а также списки посетивших.

*Область применения результатов*: могут быть использованы в РДТУП «Медтехника» или на других предприятиях, нуждающихся в автоматизации процесса организации вебинаров.

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ. 7](#_Toc60238352)

[ВВЕДЕНИЕ 8](#_Toc60238353)

[1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИССЛЕДОВАНИЙ И ПРОГРАММНЫХ РЕШЕНИЙ 9](#_Toc60238354)

[1.1 Особенности и проблемы организации вебинаров для взаимодействия корпоративных сотрудников 9](#_Toc60238355)

[1.2 Сравнительный обзор аналогов программного средства 15](#_Toc60238356)

[2. МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ 20](#_Toc60238357)

[2.1. Анализ и формализация бизнес-процессов предметной области 20](#_Toc60238358)

[2.2. Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований 34](#_Toc60238359)

[2.3. Информационная модель и ее описание 36](#_Toc60238360)

[2.4. Модели представления программного средства 40](#_Toc60238361)

[3. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА ЦЕЛЕВОГО УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ ПРЕДПРИЯТИЯ СФЕРЫ УСЛУГ. 43](#_Toc60238362)

[3.1. Архитектурные решения. 43](#_Toc60238363)

[3.2. Описание алгоритмов, реализующих бизнес-логику разрабатываемого программного средства. 47](#_Toc60238364)

[3.3. Проектирование пользовательского интерфейса. 49](#_Toc60238365)

[4. РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ПОДДЕРЖКИ ПРОЦЕССА ОРГАНИЗАЦИИ ВЕБИНАРОВ ДЛЯ ВЗАИМОДЕЙСТВИЯ КОРПОРАТИВНЫХ СОТРУДНИКОВ. 58](#_Toc60238366)

[4.1. Обоснование выбора компонентов и технологий для реализации дипломного проекта. 58](#_Toc60238367)

[4.2. Программная реализация алгоритмов, реализующих бизнес-логику программного средства. 61](#_Toc60238368)

[4.4. Руководство по развертыванию и использованию программного средства. 67](#_Toc60238369)

[5. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО СРЕДСТВА. 76](#_Toc60238370)

[5.1. Характеристика программного средства. 76](#_Toc60238371)

[5.2. Расчет сметы затрат на разработку программного средства. 76](#_Toc60238372)

[5.3. Расчет стоимостной оценки результата. 79](#_Toc60238373)

[5.4. Расчет показателей эффективности разработки и использования программного средства. 80](#_Toc60238374)

[ЗАКЛЮЧЕНИЕ 82](#_Toc60238376)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 83](#_Toc60238377)

[ПРИЛОЖЕНИЕ А 87](#_Toc60238378)

[ПРИЛОЖЕНИЕ Б 88](#_Toc60238379)

[ПРИЛОЖЕНИЕ В 89](#_Toc60238380)

## ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ.

|  |  |  |
| --- | --- | --- |
| БД | – | База данных |
| НДС | – | Налог на добавленную стоимость |
| ПО  ПП  ПС | –  –  – | Программное обеспечение  Программный продукт  Программное средство |
| СУБД | – | Система управления базами данных |
| *CSS*  *DOM*  *GUI*  *HMVC* | –  –  –  – | *Cascading Style Sheets*, каскадные таблицы стилей  *Document Object Model*, объектная модель документа  *Graphical User Interface*, графический интерфейс пользователя  *Hierarchical Model-View-Controller*, Иерархические Модель-Вид-Контроллер |
| *HTML*  *HTTP* | –  – | *HyperText Markup Language*, язык гипертекстовой разметки  *HyperText Transfer Protocol*, протокол передачи гипертекста |
| *IDEF* | – | *Integrated DEFinition*, методологии для решения задач моделирования сложных систем |
| *IT* | – | *Information Technology*, информационные технологии |
| *MVC* | – | *Model-View-Controller*, Модель-Представление-Контроллер |
| *UML* | – | *Unified Modeling Language*, унифицированный язык моделирования |

## ВВЕДЕНИЕ

Общение в формате вебинара получило широкую популярность благодаря возможности обмениваться новостями, мыслями, эмоциями и знаниями. Рост и широкое распространение обучающих онлайн-трансляций, видеолекций и вебинаров обусловлено повышением интенсивности получения, обработки, анализа и обмена информации. В деловом секторе, а здесь вышеперечисленные методы используют чаще всего, объем новых знаний, техник, новаций, полученных от иностранных и отечественных предпринимателей, компаний и авторов бизнес-литературы, растёт постоянно, и следует своевременно обращать внимание на новые тренды. Поэтому недостаточно только читать бизнес-литературу, мониторить блоги бизнес-гуру или посещать встречи с ними; новые тенденции появляются ежедневно и для успешного ведения бизнеса необходимо быть на передовой позиции.

Для компаний, предприятий, индивидуальных потребителей обучающих продуктов также встает вопрос о стоимости посещения встречи или приглашения тренера на свою площадку – в этом случае онлайн-трансляции выглядят привлекательнее. Ширина аудитории не ограничена количеством мест в зале.

Однако организовать вебинар, или даже организовать посещение уже организованного вебинара сотрудниками предприятия не так и просто

На сегодняшний день роль внутренних коммуникаций для организации чрезвычайно высока и состоит в обмене своевременной и достоверной информацией, а вебинар как раз является одной из форм этого общения.

Интегрирование информационных потоков в единую среду является одной из основных задач, которая стоит перед организацией. Любой группе сотрудников, чтобы быть эффективной в своем деле, нужно иметь инструменты для быстрого обмена данными и информацией.

Анализ рынка таких систем показал, что, во-первых, большинство разработанных решений имеют избыточный функционал. Это приводит к тому, что пользоваться такими системами сложно, а их внедрение для средних и малых компаний являются затратными в финансовом плане. Во-вторых, в компаниях используются, преимущественно, собственные разработки, которые не доступны за их пределами.

Актуальность создания системы организации вебинаров обуславливается необходимостью обеспечить сотрудников компании инструментом, который позволяет быстро организовать или осуществить информирование по уже существующему вебинару.

## 1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИССЛЕДОВАНИЙ И ПРОГРАММНЫХ РЕШЕНИЙ

1.1 Особенности и проблемы организации вебинаров для взаимодействия корпоративных сотрудников

В современной экономике процесс ускорения технологического прогресса, глобализации, чрезмерной конкуренции и коммодификации делает интеллектуальную составляющую человеческого капитала основой конкурентного преимущества компании. От 70 до 80% добавленной стоимости формируется человеческим интеллектом, основным средством производства которого является человеческий мозг. Знания становятся новой сферой конкуренции между странами, компаниями и частными лицами.

Для компаний становится все более важным не только повышать квалификацию своих сотрудников, но и обучать их конкретным знаниям, навыкам и умениям, необходимым им для осуществления профессиональной деятельности на конкретном рабочем месте. Для непрерывного развития компании по выращиванию новых талантов необходимы новые формы обучения, одной из которых является обучение персонала предприятия [1].

Под корпоративным обучением понимается совершенствование образования и приобретение новых навыков и компетенций для сотрудников одной и той же компании. Целью корпоративного обучения является повышение эффективности работы каждого сотрудника в отдельности и компании в целом. Руководство компании устанавливает цели и задачи, которые необходимо решить, участников процесса обучения, его вид и метод [2].

Корпоративное обучение может быть системным или разовым, ориентированным на решение конкретных задач. Например, для сотрудников отдела продаж, когда в продуктовой линейке появляется новый продукт, может возникнуть необходимость в обучении для решения конкретной проблемы. Иногда будут проводиться разовые тренинги, такие как тимбилдинг, оптимизация рабочего времени. Систематическое обучение используется в ситуациях, когда сотрудники должны регулярно адаптироваться к меняющимся условиям труда, не теряя производительности. Бухгалтеры, юристы и операторы программного обеспечения нуждаются в систематическом обучении.

Корпоративное обучение может осуществляться как собственными, так и сторонними силами. Для проведения единоразового корпоративного обучения обычно приглашают внешнего тренера или тренинговую компанию, которые получают конкретное задание и выстраивают процесс обучения в соответствии с поставленной задачей. Для систематического обучения чаще всего используется штатный тренер [3]. Это позволяет снизить затраты на обучение, но есть и отрицательные стороны внутреннего обучения: ежедневная вовлеченность штатного тренера в проблемы предприятия не позволяет ему взглянуть на поставленную задачу «сверху» и увидеть новые пути её решения. Также отношение персонала к штатному тренеру как к «своему» человеку может помешать полноценному восприятию материала.

Существует множество различных форм корпоративного обучения: семинары, лекции, тренинги и деловые игры. Для каждой задачи тренер должен найти оптимальный способ тренировки или их комбинацию. Учебные мероприятия могут проводиться на месте или в офисе компании. Есть также преимущества и недостатки. Изменение окружающей среды помогает людям взглянуть на старые проблемы по-новому. Живые мероприятия запоминаются обучающимися как мероприятия для большой компании, что улучшает их отношение к компании, менеджменту и работе. Обучение на месте снижает затраты компании, но оно может отвлечь сотрудников от повседневных рабочих проблем и решить задачи, поставленные в ходе обучения.

Развитие мультимедийных технологий и расширение каналов передачи данных через интернет открыли широкий спектр возможностей для использования дистанционных методов обучения на предприятиях. Дистанционное обучение может осуществляться как в автономном режиме, так и в режиме онлайн. В офлайн-форме учебные материалы размещаются на сайте, а участники учебного процесса могут получить ключи доступа к своим ресурсам. Обучающиеся самостоятельно знакомятся с учебными материалами и при необходимости консультируются с тренером (наставником), сопровождающим учебный процесс. Во время обучения преподаватель осуществляет промежуточный контроль знаний обучающегося с помощью тестов или индивидуальных заданий, а обучение заканчивается выпускным экзаменом.

Возможно повысить интерактивность дистанционного обучения, запустив учебный процесс на веб-сайте вебинара. За последние несколько лет вебинары стали одним из самых популярных инструментов дистанционного обучения. Сегодня трудно найти организацию, которая широко использует дистанционные методы обучения и не использует вебинары. Популярность вебинаров объясняется их сочетанием низкой стоимости и высокой эффективности.

Вебинар – это презентация, лекция, семинар или курс, организованные при помощи web-технологий в режиме реального времени [4]. Во время вебинара его участники находятся у своих компьютеров, а связь между ними поддерживается через Интернет или Интранет посредством загружаемого приложения. Слушатели могут задавать вопросы ведущему в ходе вебинара:

* в текстовом чате или в микрофон – и получать на них ответы. Ведущий может проводить опросы обучающихся, контролируя усвоение материала, демонстрировать им слайды, фотографии, видеоролики и т.д. В конце образовательного курса также проводится тестирование или экзаменация [5].

Технология вебинаров имеет широкие границы применения и может использоваться для:

* проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения;
* проведения единоразовых семинаров и тренингов;
* проведения опросов;
* выступления с докладами и защиты выполненных работ;
* презентации продуктов и услуг;
* организации коллективной работы.

Многие бизнес-компании проводят вебинары для демонстрации своих продуктов и услуг. Вебинары широко используются как средство проведения встреч для компаний с разветвленной филиальной сетью. Но наиболее распространенным применением технологии вебинаров является онлайн-обучение,как форма дистанционного обучения.

Вебинары появились совсем недавно, но их использование настолько интенсивно, что на сегодняшний день накопилось большое количество инструментов и инструментов, которые можно использовать во время обучения:

Аудио.Во время вебинара основная коммуникация между ведущим и аудиторией осуществляется посредством аудиосвязи в режиме реального времени. Участники вебинара могут публиковать записи, чтобы другие участники могли получить к ним доступ.

Видео. Во время учебного процесса обучающимся могут отображать видео. Он также может транслировать видео в режиме реального времени как от ведущего к аудитории, так и в обратном направлении; презентация. Во время проведения вебинара ведущие могут представлять слайды презентации аудитории на своих персональных компьютерах, чтобы управлять презентацией в режиме реального времени.

Файл презентации. Во время вебинара ведущие могут представить аудитории различные файлы на своих персональных компьютерах, выделяя области, требующие особого внимания.

Обмен файлами. Во время обучения ведущие и слушатели могут обмениваться файлами или предоставлять к ним доступ.

Электронная доска. Так же, как и в реальном классе для обучения сотрудников, возможно рисовать, стирать и так далее на доске. Все участники могут оставлять заметки на доске в соответствии со своими существующими правами.

Демонстрационный рабочий стол. Во время вебинара можно показать всем участникам свой рабочий стол Windows и показать, что с ним происходит.

Эффективным средством взаимодействия участников вебинара является чат, с помощью которого они могут обмениваться мгновенными сообщениями в режиме реального времени. Граница видимости сообщения определяется участниками вебинара.

Голосование и опросы. Голосование и опросы являются очень эффективным средством обучения через вебинары, которые позволяют собирать информацию от вашей аудитории по конкретным вопросам в режиме реального времени.

Удаленный рабочий стол. Многие программные продукты, используемые во время вебинара, предоставляют участникам возможность манипулировать объектами на рабочих столах других пользователей. Это может быть полезно, когда вам нужно что-то показать аудитории вебинара. Такая возможность доступна всем участникам вебинара;

Поддержка мобильных устройств. Многие программные продукты и сервисы, предназначенные для проведения вебинаров, поддерживают большинство существующих мобильных устройств (iOS, Android, BlackBerry и т. д.).).

Запись вебинара. Пользователи вебинаров имеют возможность записывать вебинары, в которых они участвуют, чтобы позже просмотреть их снова.

Интеграция с другими информационными системами. Программное обеспечение и сервисы вебинаров обычно предоставляют пользователям возможность интегрировать вебинары в веб-сайты или интранеты.

Для любого вида корпоративного обучения отслеживается его эффективность. Процедура оценки эффективности вебинара зависит от целей его проведения. Лучше всего проводить вебинар после завершения тренинга, что позволит выявить опрос среди слушателей: достигли ли ожидаемых результатов обучения; есть ли у вебинара модератор, стоит ли привлекать аудиторию к теме вебинара и т.д. Полезно собирать статистику активности аудитории во время вебинаров (как они участвуют в них, как часто задают или отвечают на вопросы, сколько слушателей изъявили желание посетить будущие вебинары). Информация о пройденных курсах, оценках и сертификациях может быть записана в виртуальный профиль сотрудника, что позволит HR-экспертам учитывать потенциал и деятельность сотрудников при разработке кадровой политики, систем стимулирования сотрудников и корпоративных программ обучения.

Как и любой метод обучения, вебинары имеют возможность и ограничения для определения сферы их применения. Абсолютным преимуществом вебинаров перед очными методами обучения является возможность собрать большое количество людей (более 100 человек) одновременно в информационном пространстве, независимо от их местонахождения, без необходимости организовывать их к месту обучения по сравнению с офлайн-формой дистанционного обучения, вебинары обеспечивают участникам высокую степень интерактивности [6]. Кроме того, широкий спектр обучающих средств и инструментов способствует процессу восприятия слушателями вебинара информации. Все это оказывает положительное влияние на результаты обучения.

Конечно, у вебинаров есть свои ограничения. Обучение посредством вебинаров имеет ряд особенностей, которые следует учитывать при проведении корпоративного обучения. К ним относятся:

* + отсутствие непосредственного визуального контакта между ведущим и участниками вебинара, что затрудняет контроль поведения слушателей;
  + высокие требования к уровню подготовки всех участников вебинара в области использования современных информационных технологий;
  + высокие требования к технической инфраструктуре проведения вебинаров;
  + высокая стоимость разработки учебного контента вебинаров.

В заключение следует отметить, что на современном этапе лучше всего сочетать вебинары с другими инструментами обучения компании, в основном дистанционными курсами. Включение вебинаров в существующие системы дистанционного обучения значительно повысит эффективность обучения в компании. Принимая во внимание возможности и ограничения данного метода обучения, можно определить, что использование вебинаров будет целесообразным и наиболее эффективным в данном случае.

Вебинары могут использоваться в качестве до- и пост- тренинговой поддержки очного обучения. Предварительное ознакомление обучающихся с теорией, алгоритмами и инструментами тренинга позволит сократить его длительность и сконцентрировать внимание участников на разборе и отработке практических ситуаций по теме тренинга. Получение первоначальной информации посредством вебинара, позволит участникам задать вопросы, уточнить непонятные моменты и лучше запомнить принципиально важные блоки. После основного обучения можно повторить основные положения тренинга, дать дополнительную информацию по ключевым вопросам, разобрать и обсудить вопросы и сложные ситуации участников тренинга.

1. Вебинары могут использоваться как самостоятельный метод обучения, когда содержание обучения не требует интенсивного обсуждения, но у студентов могут возникнуть проблемы. Это может быть обучение стандартам работы компании,передача опыта и различных рабочих инструментов. Основная задача этого тренинга-своевременно и регулярно информировать сотрудников об изменениях, а затем каждый сотрудник выбирает, что и как лучше использовать.
2. Для распространения опыта можно использовать вебинары. Предположим, что в одном из филиалов успешно завершен новый проект. Очень важно, чтобы он не потерялся и не нашел его в других отраслях. Вместо того чтобы попросить руководителя проекта описать этот опыт и опубликовать его на сайте компании, был организован вебинар. А позже это может стать традицией, когда раз в месяц кто-то делится своим опытом с другими в режиме вебинара.
3. Вебинары могут использоваться для передачи информации о самой системе обучения, конкретных программах обучения и т. д. Рекламные вебинары помогут сотрудникам лучше понять цели тренинга, ознакомиться с инструкторами и участниками, правилами и процедурами предстоящего тренинга, то есть психологически подготовиться к будущему обучению.

Таким образом, включение вебинаров в систему корпоративного обучения позволит компании расширить аудиторию обучающихся без потери качества обучения и при сохранении текущих бюджетов на образовательные цели.

И все же стоит помнить: часто бывает так, что открытые вебинары проводят другие компании, или же индивидуалисты, разбирающихся в своей сфере. И о проведении таких вебинаров уведомляют руководителей других компаний, которые по цепочке должны передавать информацию о проведении остальным людям среди корпоративных сотрудников. Такие вещи легко забыть, если никто не будет напоминать время от времени.

1.2 Сравнительный обзор аналогов программного средства

Вебинары, как современное средство коммуникации и информационного общения, используются самыми разными специалистами, от преподавателей до бренд-менеджеров и основателей компаний.

Вебинары – это популярный и удобный способ подключения пользователей к вашей аудитории, будь то студенты или клиенты. Популярность вебинаров достигается сочетанием гибкости взаимодействия с удобством передовых технологий: мгновенной адаптацией информации к аудитории, персонализированной подачей информации путем ответов на вопросы после вебинара и так далее.

На самом базовом уровне программное обеспечение для вебинаров включает в себя возможности видео-и аудиоконференций. Некоторые из них также предоставляют такие функции, как общий доступ к экрану, запись видео, общий доступ к файлам и интеграция с другими программами, такими как Microsoft PowerPoint.

Большинство программ вебинаров платные, и многие пользователи обращаются к традиционным программам видеоконференцсвязи. Программное обеспечение для видеоконференций и вебинаров часто очень похожи, но количество и функциональность участников различны. Однако многие люди считают, что им нужна функциональность для проведения вебинаров в бесплатном программном обеспечении для видеоконференций.

При проведении вебинара, необходимо учитывать следующее:

* количество участников;
* длительность вашего мероприятия;
* желаемые функции (запись, брендинг и т. д.);
* потребность в масштабируемости.

Существует 7 современных аналогов программных средств для проведения вебинаров:

1. FreeConferenceCall.com;
2. Zoom;
3. UberConference;
4. Skype;
5. Join.me;
6. Apache OpenMeetings;
7. InstaWebinar.

Сравнительная характеристика программных аналогов для проведения вебинаров представлена в таблице 1.1.

Таблица 1.1 – Сравнительная характеристика программных аналогов для проведения вебинаров

|  |  |  |  |
| --- | --- | --- | --- |
| Программные аналоги | Рейтинг по 5-балльной шкале | Максимальное число участников | Наличие платной версии |
| 1 FreeConferenceCall.com | 4,6 | 1 000 | Нет |
| 2 Zoom | 4,5 | 100 | Да |
| 3 UberConference | 4,3 | 100 | Да |
| 4 Skype | 4,2 | 25 | Нет |
| 5 Join.me | 4,1 | 3 | Да |
| 6 Apache OpenMeetings | 4,1 | Неограниченно | Нет |
| 7 InstaWebinar | 4,1 | 100 | Нет |

Примечание – Источник: собственная разработка на основе.

1 FreeConferenceCall.com. Вебинары с поддержкой 1000 участников [7]. Продукт также имеет множество функций, которые обычно не предлагаются бесплатными продуктами, включая брендинг и круглосуточное обслуживание клиентов.

Бесплатный вариант: Доступно для Windows, Mac, iOS и Android

Платный вариант: Нет.

Характеристики:

1. Совместное использование экрана;
2. Международные набор номера;
3. Запись;
4. Круглосуточная поддержка;
5. Пользовательский брендинг;
6. Настраиваемый URL.

Преимущества: запись звонков и возможность совместного использования экрана.

Недостатки: Отсутствие возможности отправления файлов. Иногда происходит задержка звука.

2 Zoom. Zoom стал синонимом бизнес-видеоконференций с интуитивно понятным дизайном и высокой простотой использования [8]. С помощью бесплатного плана пользователи могут разместить до 100 участников одновременно в течение 40 минут неограниченных встреч один на один.

Бесплатный вариант: Доступно для Windows, Mac, iOS и Android

Платный вариант: Zoom Вебинар: 54,99 долл. США в месяц

Характеристики:

* одновременная трансляция экрана;
* показ активного спикера;
* возможность присоединения по телефону;
* совместная аннотация на общем экране.

Преимущества: возможность участников задавать вопросы и добавлять комментарии. Возможность использовать на телефоне.

Недостатки: функций вебинара достаточно для проведения полноценного вебинара, в том числе с использованием опросов, инструментов разметки и т. д. В то же время платформе по-прежнему не хватает интеграции с другими инструментами, такими как автоответчики (например, ConvertKit, Active Campaign и т. д.).

3 UberConference. UberConference предлагает простые функции, может вместить до 10 человек и предлагает бесплатный уровень [9]. Это отличный вариант, если есть возможность провести вебинар без видео, только с медиа-материалами, такими как PowerPoint или Google Slides.

Бесплатный вариант: доступно для Windows, Mac, iOS и Android

Платный вариант: 10 долларов США за линию: до 100 участников, без аудиорекламы, международный доступ.

Характеристики:

* неограниченное количество конференций;
* cовместное использование экрана
* обмен документами;
* запись.

Преимущества. В Uberconference легко разобраться. Легко настроить встречи. Легко использовать. Отличное программное обеспечение.

Недостатки. Единственный минус в том, что параметры совместного доступа к экрану доступны только пользователям, использующим Google Chrome в качестве основного браузера. Ведущий использует конференцию Uber Conference Google Extension.

4 Skype. Функция группового голосового и видеозвонка Skype позволяет одновременно разместить до 25 человек для участия в вебинаре [10]. Учитывая популярность Skype, ваша аудитория,возможно, установила приложение, что является дополнительным минусом шага между вами и участниками вашего мероприятия.

Бесплатный вариант: Доступно для Windows, Mac, iOS и Android

Платный вариант: Нет.

Характеристики:

* обмен файлами;
* групповой чат;
* совместное использование экрана.

Преимущества: Skype позволяет мне общаться с несколькими людьми одновременно, без дополнительной платы. Компания признана во всем мире, и она обеспечивает меньшее количество хлопот при общении с людьми из-за ее легкости и доступности.

Недостатки: Иногда, когда подключение к сети слабое, очень сложно вести коммуникацию. Проблемы со звуком в системе также являются распространенной проблемой, с которыми сталкиваются пользователи.

5. join.me. Он служит отличным, без излишеств решением для проведения конференций и вебинаров [11]. Он имеет надежное мобильное приложение и чистый пользовательский интерфейс. Бесплатная версия предлагает хостинг для трех участников, что делает ее идеальной для небольших командных встреч.

Бесплатный вариант: Доступно для Windows, Mac, iOS и Android.

Платный вариант: 10 долларов США за пользователя в месяц: до пяти участников, персональный URL-адрес, автономный звук, неограниченное количество звонков и интернет-звонков

20 долларов США за пользователя в месяц: до 50 участников, 10 видеопотоков, дополнительные бесплатные номера, управление пользователями, неограниченная аудиоконференция

30 долларов США за пользователя в месяц: до 250 участников собрания, единого входа, интеграции Salesforce

Характеристики:

* совместное использование экрана;
* бесплатный звук через VoIP.

Преимущества: Простой в использовании пользовательский интерфейс. Все интуитивно, редко возникают проблемы с настройкой. Кроме того, функция аудиоконференции всегда очень хорошего качества. Возможность настройки фона и значков также является большим плюсом.

Недостатки: невозможность использовать приложение для десктопа, но приложение для мобильных приложений и планшетов компенсирует это.

6. Apache OpenMeetings. Apache OpenMeetings предлагает множество уникальных функций для бесплатных продуктов, включая модерацию и систему управления пользователями [12]. Программное обеспечение является открытым исходным кодом, и его сообщество стремится поддерживать новых пользователей через активный форум.

Бесплатный вариант: Доступно для Windows и Mac

Платный вариант: Нет

Характеристики:

* запись;
* проводник файлов;
* система модерации;
* чат;
* управление пользователями и комнатами;
* частные сообщения;
* интегрированный календарь
* опросы и голосования.

Преимущества: Whiteboard и потоковый сервер были функциями, которые получили самые положительные отзывы. Удобная связь и время безотказной работы.

Недостатки: Apache OpenMeetings выглядит устаревшим и не имеет современного дизайна.

7. InstaWebinar. InstaWebinar – это совершенно бесплатный сервис с уникальной функциональностью [13]. Он позволяет до 100 участников без ограничений по длительности.

Бесплатный вариант: Доступно для Windows, Mac, iOS и Android

Платный вариант: Нет.

Характеристики:

* совместное использование PowerPoint и веб-камеры;
* создает списки имен и адресов электронной почты участников.

Таким образом, анализ аналогов приложения для проведения вебинаров, позволил установить, что при выборе, какое программное обеспечение для вебинаров лучше всего подходит определенной компании, необходимо подумать о том, сколько людей ожидается обучать. Также необходимо не забывать о дополнительных функциях, которые понадобятся, чтобы сделать мероприятие наиболее эффективным для достижения целей.

## 2. МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ

2.1. Анализ и формализация бизнес-процессов предметной области

В данном подразделе будут рассмотрены бизнес-процессы текущей прикладной области. При этом будет произведено описание процессов, как в текстовом формате, так и при помощи нотации IDEF0 в графическом виде в системе Business Studio 4.2.

Методология IDEF0 – это одна из самых известных методологий функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов [14].

Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Стандарт IDEF0 представляет организацию как набор модулей. Здесь существует правило – наиболее важная функция находится в верхнем левом углу, кроме того, есть правило стороны:

– стрелка входа всегда приходит в левую кромку активности;

– стрелка управления – в верхнюю кромку;

– стрелка механизма – нижняя кромка;

– стрелка выхода – правая кромка.

Описание выглядит как «черный ящик» с входами, выходами, управлением и механизмом, который постепенно детализируется до необходимого уровня.

Также для того чтобы быть правильно понятым, существуют словари описания активностей и стрелок. В этих словарях дается описание того, какой смысл вкладывается в данную активность либо стрелку.

Как можно понять из определения, методология описывает бизнес-процессы, их иерархичность и взаимодействие в рамках определенной системы.

В рамках текущей прикладной области будет описана деятельность в двух вариантах:

1 Модель «AS-IS». Представляет собой описание системы в нотации IDEF0 до того, как было принято решение об автоматизации и оптимизации системы, то есть «как есть».

2 Модель «TO-BE». Представляет собой описание системы в нотации IDEF0 после того, как были проведены автоматизация и оптимизация системы, то есть «как должно быть».

Для начала будет рассмотрено описание предметной области в нотации IDEF0 моделью «AS-IS».

Первый блок модели, как и предписывает методология *IDEF*0, представляет собой одну контекстную диаграмму верхнего уровня. Представляет собой процесс «Процесс организации вебинаров». Диаграмма представлена на рисунке 2.1.

На вход подаются параметры: информация о проведении вебинара, желание сотрудника провести вебинар. Данные параметры будут преобразованы благодаря процессу в выходные параметры.

Выходные параметры из данного блока: сотрудники предприятия с повышенным уровнем знаний и закрытая отзывы о проведении вебинара.

Управление – приказ о повышении квалификации сотрудника, правила проведения вебинара, штатное расписание.

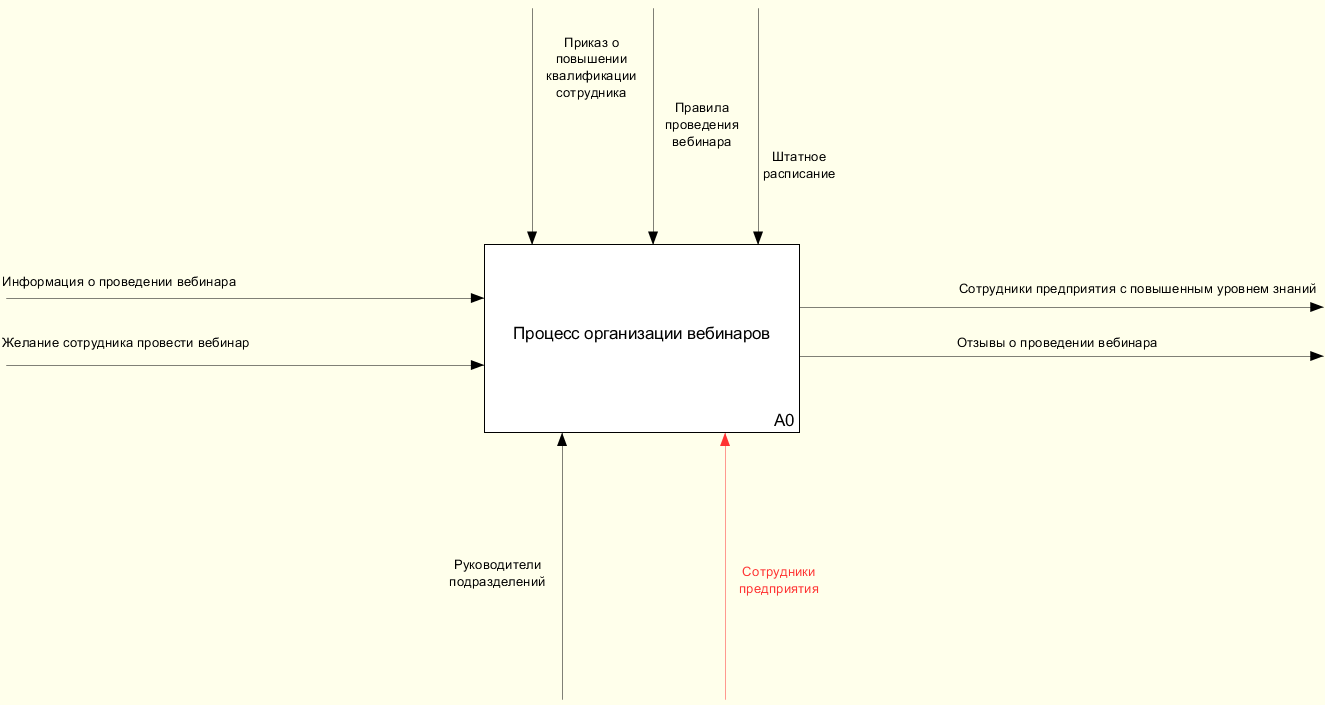


Рисунок 2.1 – Контекстная диаграмма функциональной модели системы в нотации *IDEF*0 модели «*AS*-*IS*»

Механизмы – те или то, кто осуществляет сам процесс: руководители подразделений и сотрудники предприятия.

На диаграмме отражен процесс организации вебинаров на предприятии.

Контекстная диаграмма верхнего уровня модели «AS-IS» разбивается на четыре процесса, и образуется первый уровень модели «AS-IS», представленный на рисунке 2.2.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Получение приказа о необходимости проведения вебинара
* Уведомление сотрудников о проведении вебинара
* Проведение вебинара
* Получение отзыва об вебинаре

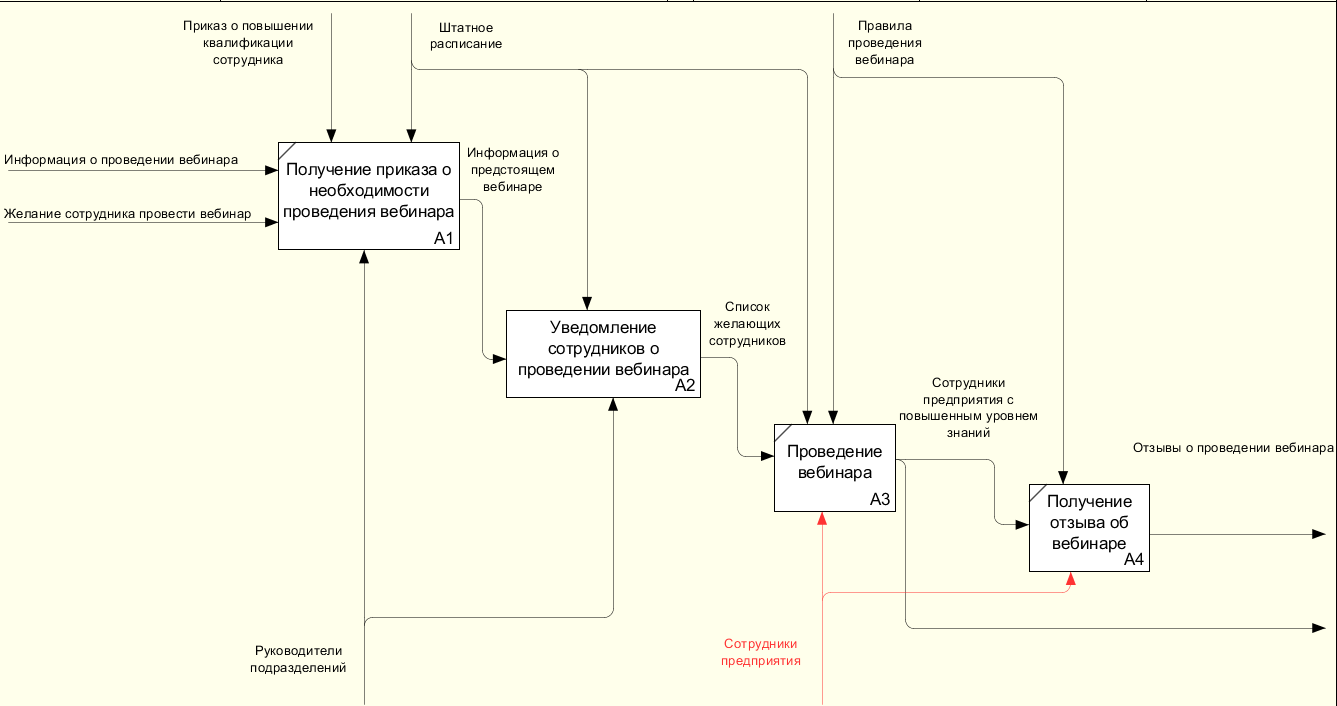


Рисунок 2.2 – Первый уровень функциональной модели системы в нотации *IDEF*0 модели «*AS*-*IS*»

Бизнес-процесс «Получение приказа о необходимости проведения вебинара» предполагает получение всей необходимой первичной информации, необходимой для организации данного вебинара

В ходе выполнения данного процесса происходит получение информации, которая отвечает на следующие вопросы:

* Каковы цели проведения вебинара
* Кто будет проводить вебинар
* Для какой аудитории рассчитан вебинар
* Какая его тема и название
* В какое время он будет проводиться

Выполняется процесс таким механизмом, как руководители подразделений.

Бизнес-процесс «Уведомление сотрудников о проведении вебинара» предполагает последовательность действий, направленных передачу информации от руководителей подразделений их непосредственным сотрудникам.

Результатом является преобразование информации о предстоящем вебинаре в список сотрудников, желающих посетить вебинар.

Выполняется такими механизмами, как руководители подразделений

Бизнес-процесс «Проведение вебинара» предполагает последовательность действий, направленных на повышение уровня знаний сотрудников посредством проведения онлайн-конференции.

Результатом является преобразование списка желающих сотрудников в список сотрудников предприятия с повышенным уровнем знаний.

Выполняется такими механизмами, как сотрудники предприятия и управляется правилами проведения вебинаров.

Бизнес-процесс «Получение отзыва об вебинаре» предполагает последовательность действий, направленных получение информации о качестве вебинара, который был проведен

Результатом является преобразование сотрудников с повышенным уровнем знаний, которые посетили вебинар, в отзывы о проведении вебинара.

Выполняется такими механизмами, как сотрудники предприятия и управляется правилами проведения вебинара.

Второй уровень диаграммы – это разбиение бизнес-процесса «Уведомление сотрудников о проведении вебинара» на четыре подпроцесса. Этот уровень представлен на рисунке 2.3.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Подтверждение проведения вебинара
* Получение информации о проведении вебинара
* Ручная рассылка информации сотрудникам в группы Skype
* Составление списка желающих посетить вебинар

Все эти бизнес-процессы управляются штатным расписанием, а механизмом выступают руководители подразделений.

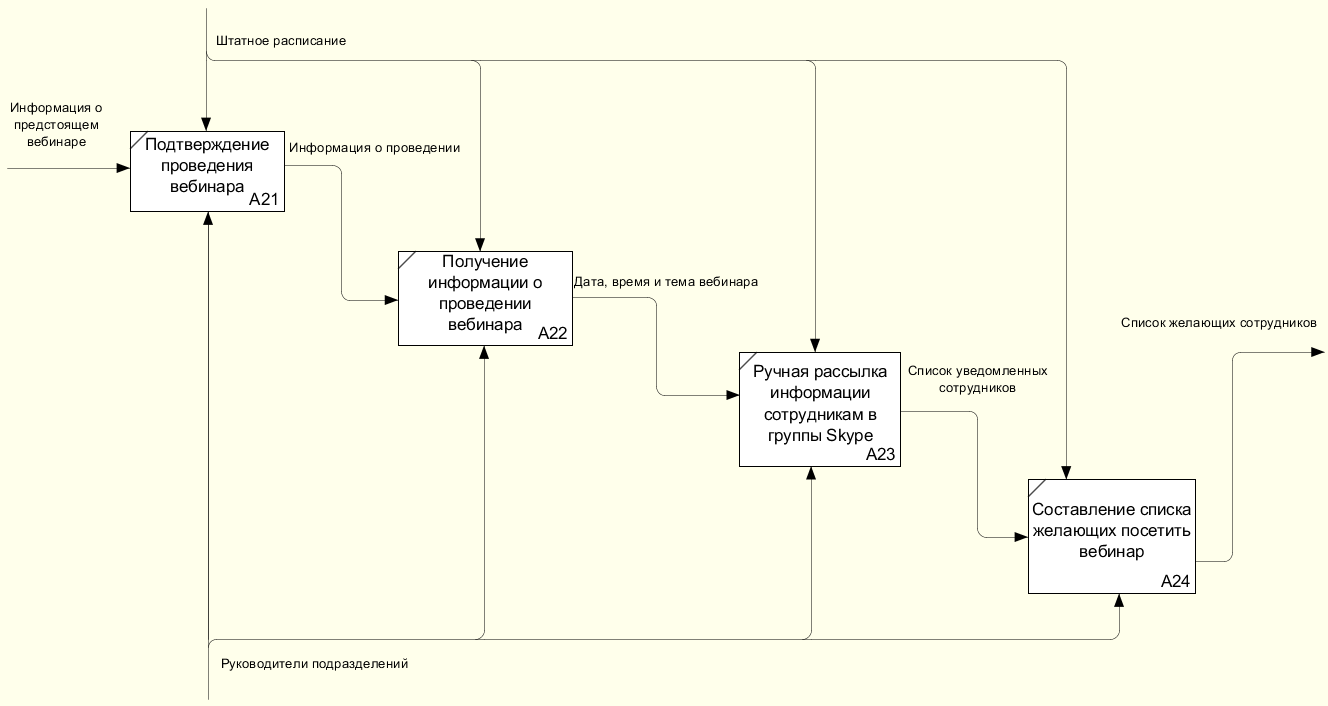


Рисунок 2.3 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*AS*-*IS*» (разбиение бизнес-процесса «Управлять персоналом предприятия»)

Бизнес-процесс «Подтверждение проведения вебинара» предполагает получение точной информации о состоянии вебинара, который планируется провести.

В ходе выполнения данного процесса формируется информация о проведении вебинара.

Бизнес-процесс «Получение информации о проведении вебинара» предполагает получение необходимых данных, которые будут использоваться в рассылке сотрудникам предприятия.

В ходе выполнения данного процесса формируется информация о дате и времени проведения вебинара, а также самой главной ее веще – теме вебинара.

Бизнес-процесс «Ручная рассылка информации сотрудникам в группы Skype» предполагает отправку информации о предстоящем вебинаре руководителями подразделений их непосредственным сотрудникам с целью их уведомления и получения списка откликнувшихся на данную рассылку сотрудников.

В ходе выполнения данного процесса формируется список уведомленных о вебинаре сотрудников.

Бизнес-процесс «Составление списка желающих посетить вебинар» предполагает составление руководителями предприятия списка желающих посетить вебинар.

В ходе выполнения данного процесса, список уведомленных сотрудников преображается в список желающих на вебинар сотрудников.

Так как некоторые моменты в системе управления персоналом предприятия можно автоматизировать, следующим этапом будет рассмотрение описания предметной области в нотации IDEF0 модели «TO-BE».

Первый блок модели «TO-BE» представляет собой контекстную диаграмму верхнего уровня. Представляет собой процесс «Организовать работу средства поддержки процесса организации вебинаров». Диаграмма представлена на рисунке 2.4.

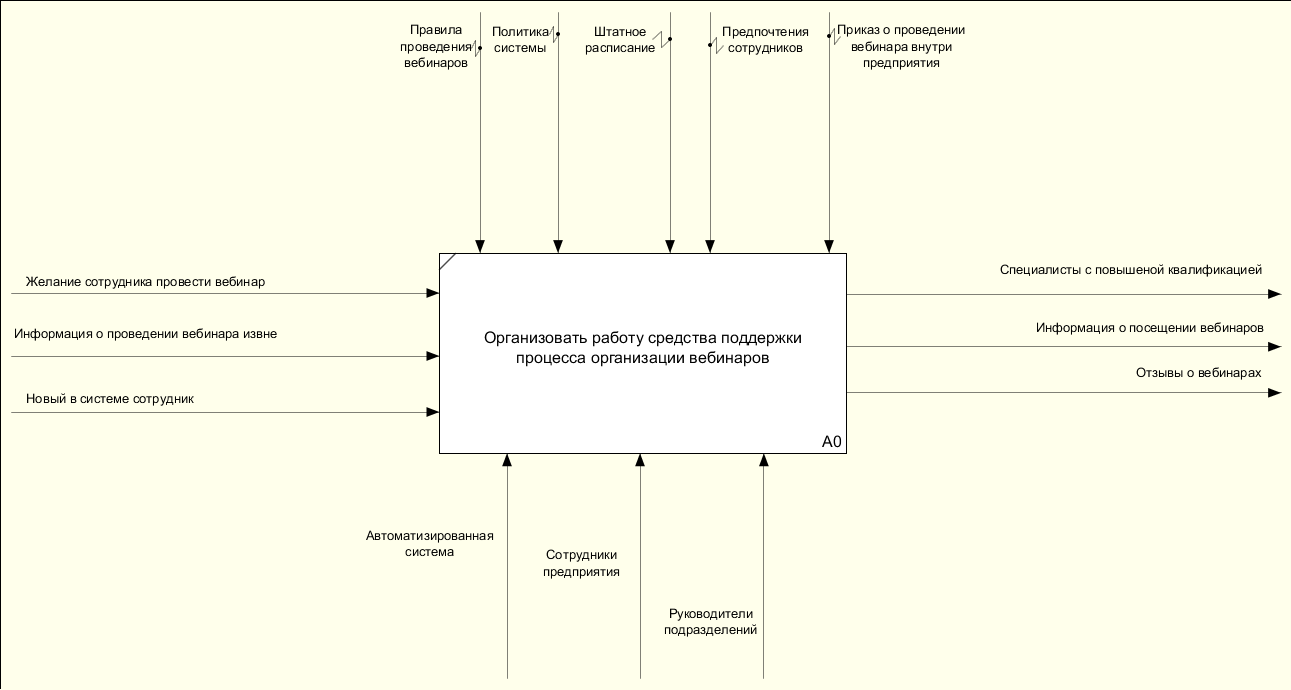


Рисунок 2.4 – Контекстная диаграмма функциональной модели системы в нотации *IDEF*0 модели «*TO*-*BE*»

Входом в процесс является:

* Информация о проведении вебинара
* Новый в системе сотрудник
* Желание сотрудника провести вебинар

В результате проведения процедур организации работы средства поддержки процесса организации вебинаров предприятие получает:

* Специалисты с повышеной квалификацией
* Информация о посещении вебинаров
* Отзывы о вебинарах

Для управления средством поддержки организации вебинаров как процессом необходимы:

* Правила проведения вебинаров
* Политика системы
* Штатное расписание
* Предпочтения сотрудников
* Приказ о проведении своего вебинара

Механизмами реализации функциональной модели системы в нотации *IDEF*0 модели «*TO*-*BE*» являются:

* Автоматизированная система
* Сотрудники предприятия
* Руководители подразделений

На диаграмме отражена работа системы поддержки процесса организации вебинаров предприятия.

Контекстная диаграмма верхнего уровня модели «*TO*-*BE*» разбивается на шесть процессов, и образуется первый уровень модели «*TO*-*BE*», представленный на рисунке 2.5.

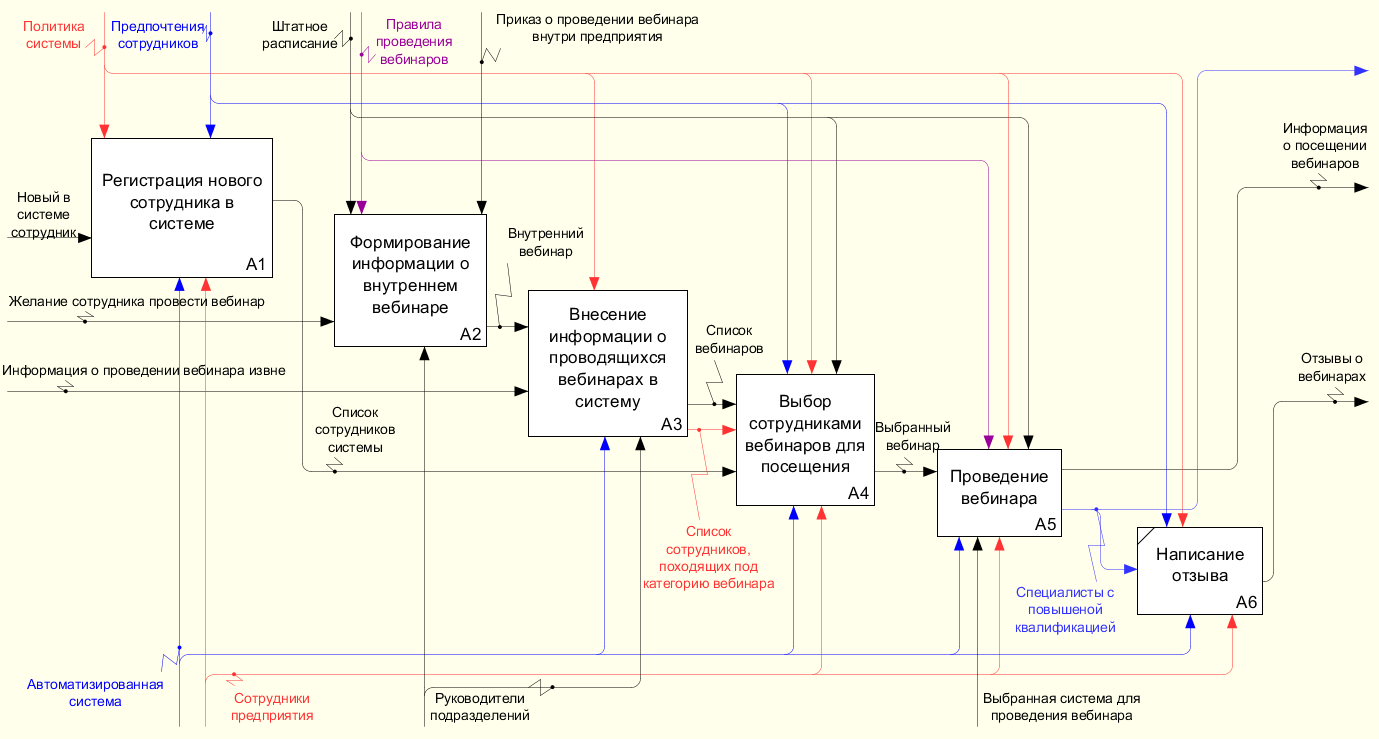


Рисунок 2.5 – Первый уровень функциональной модели системы в нотации *IDEF*0 модели «*TO*-*BE*»

Бизнес-процессы, представленные в рамках данной диаграммы:

* Регистрация нового сотрудника в системе
* Формирование информации о внутреннем вебинаре
* Внесение информации о проводящихся вебинарах в систему
* Выбор сотрудниками вебинаров для посещения
* Проведение вебинара
* Написание отзыва

Бизнес-процесс «Регистрация нового сотрудника в системе» представляет собой модуль регистрации и настройки профиля новых для системы пользователей

В результате новый в системе сотрудник преобразуется в список сотрудников системы

Выполняется таким механизмом, как автоматизированная система и сотрудники предприятия.

Управляется политикой системы и предпочтениями сотрудников

Бизнес-процесс «Формирование информации о внутреннем вебинаре» представляет собой модуль получения информации о вебинаре, который собрались проводить внутри корпорации.

В результате желание сотрудника провести вебинар преобразуется в внутренний вебинар.

Выполняется таким механизмом, как руководители системы.

Управляется штатным расписанием, правилами проведения вебинаров и приказом о проведении вебинара внутри предприятия

Бизнес-процесс «Внесение информации о проводящихся вебинарах в систему» представляет собой модуль создания записи о вебинарах внутри системы, для последующего отображения всех вебинаров сотрудникам

В результате внутренний вебинар и информация о проведении вебинаров извне преобразуется в список всех вебинаров системы.

Выполняется таким механизмом, как автоматизированная система и руководители подразделений.

Бизнес-процесс «Выбор сотрудниками вебинаров для посещения» представляет собой возможность сотрудникам самим выбирать, какие вебинары из списка предлагаемых они будут посещать.

В результате список вебинаров и список сотрудников, подходящих под категорию вебинара преобразуется в выбранный сотрудником вебинар.

Выполняется таким механизмом, как автоматизированная система и сотрудники предприятия.

Бизнес-процесс «Проведение вебинара» представляет собой непосредственное мероприятие по проведению онлайн-конференции

В результате выбранный вебинар преобразуется в информацию о посещенном вебинаре и специалистов с повышенной квалификацией.

Выполняется таким механизмом, как автоматизированная система, сотрудники предприятия, а также выбранной системой для проведения вебинара.

Бизнес-процесс «Написание отзыва» представляет собой отправку сотрудниками текстовой реакции на посещенный ими вебинар.

В результате специалисты с повышенной квалификацией преобразуются в отзывы о вебинарах.

Выполняется таким механизмом, как автоматизированная система и сотрудники предприятия.

Второй уровень диаграммы – это разбиение бизнес-процесса «Регистрация нового сотрудника в системе» на 3 подпроцесса. Этот уровень представлен на рисунке 2.6.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Краткое знакомство с системой
* Получение и ввод персональных данных в систему
* Настройка типов вебинаров для отображения

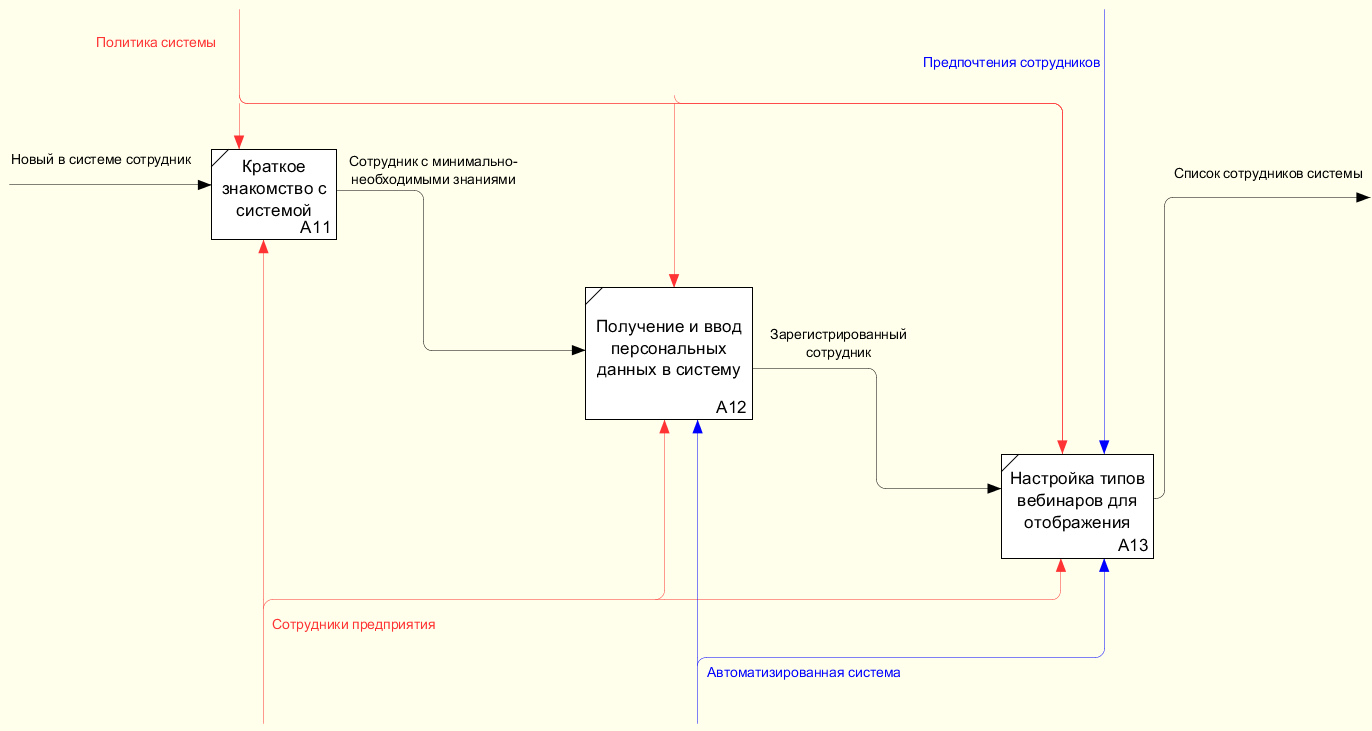


Рисунок 2.6 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*TO-BE*» (разбиение бизнес-процесса «Регистрация нового сотрудника в системе»)

Бизнес-процесс «Краткое знакомство с системой» подразумевает введение сотрудников предприятия в работу с автоматизированную систему, обучение основам работы с ней.

В результате Новый в системе сотрудник преобразуется в сотрудника с минимально-необходимыми знаниями.

Механизмом является сотрудник предприятия

Управляется политикой системы

Бизнес-процесс «Получение и ввод персональных данных в систему» подразумевает регистрацию сотрудника в системе, для дальнейшей работы с ней.

В результате сотрудник с минимально-необходимыми знаниями преобразуется в зарегистрированного сотрудника.

Механизмом является сотрудник предприятия и автоматизированная система.

Управляется политикой системы.

Бизнес-процесс «Настройка типов вебинаров для отображения» подразумевает выбор зарегистрированным сотрудником тех типов/категорий вебинаров, которые ему нравятся и о которых нужно уведомлять.

В результате зарегистрированный сотрудник преобразуется в законченный список сотрудников системы.

Механизмом является сотрудник предприятия и автоматизированная система.

Управляется процесс предпочтениями сотрудника.

Второй уровень диаграммы – это разбиение бизнес-процесса «Формирование информации о внутреннем вебинаре» на 3 подпроцесса. Этот уровень представлен на рисунке 2.7.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Выяснение цели и задач вебинара
* Определение спикера и темы вебинара
* Утверждение даты и времени проведения вебинара

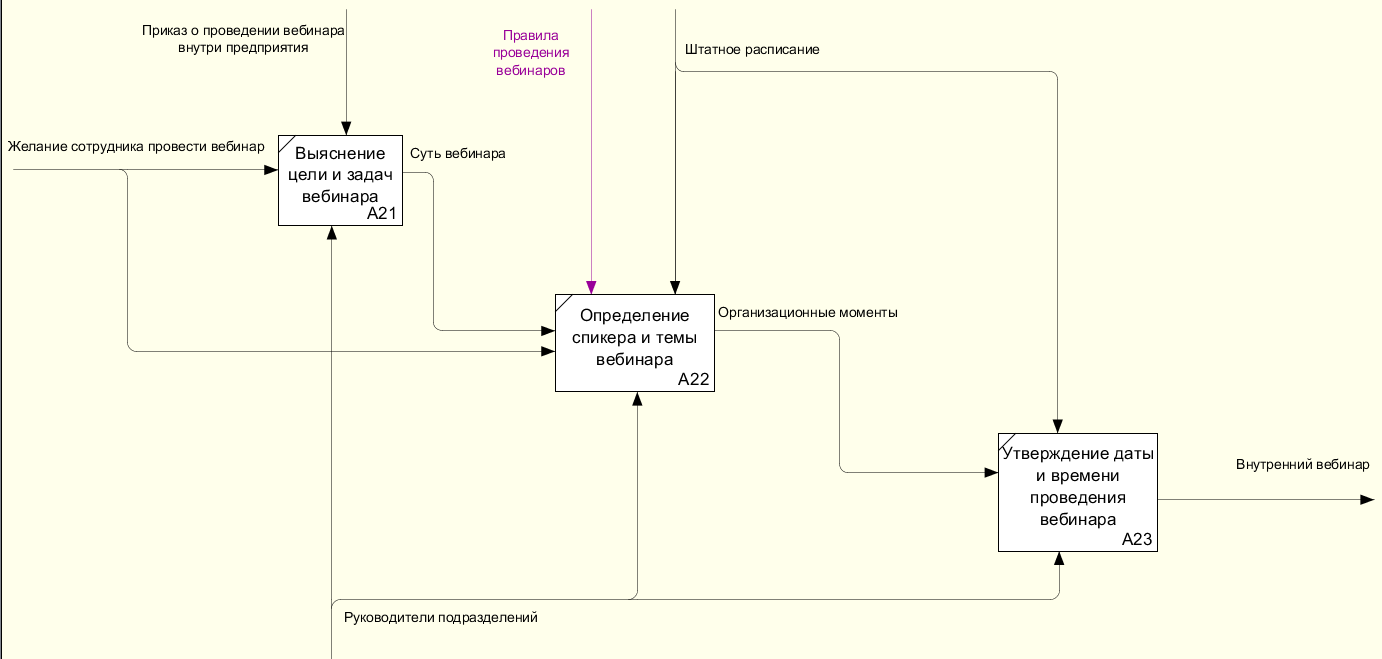


Рисунок 2.7 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*TO-BE*» (разбиение бизнес-процесса «Формирование информации о внутреннем вебинаре»)

Бизнес-процесс «Выяснение цели и задач вебинара» подразумевает определение, чего хотят добиться приказом сверху, на что должен влиять вебинар и о чем быть.

В результате желание сотрудника провести вебинар преобразуется в суть вебинара.

Механизмом является руководители подразделений.

Управляется приказом о проведении вебинара внутри предприятия.

Бизнес-процесс «Определение спикера и темы вебинара» подразумевает решение руководства о назначении спикера для проведения вебинара, а также окончательно е утверждение темы вебинара.

В результате суть вебинара преобразуется в организационные моменты.

Механизмом является руководитель подразделения

Управляется правилами проведения вебинаров и штатным расписанием.

Бизнес-процесс «Утверждение даты и времени проведения вебинара» подразумевает решение оставшихся организационных моментов перед окончательным утверждением о проведении вебинара.

В результате организационные моменты преобразуется во внутренний вебинар.

Механизмом является руководитель подразделений.

Управляется штатным расписанием.

Второй уровень диаграммы – это разбиение бизнес-процесса «Внесение информации о проводящихся вебинарах в систему» на 3 подпроцесса. Этот уровень представлен на рисунке 2.8.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Разбиение всей информации на информационные куски
* Сохранение информационных частей в системе
* Уведомление сотрудников о новом вебинаре

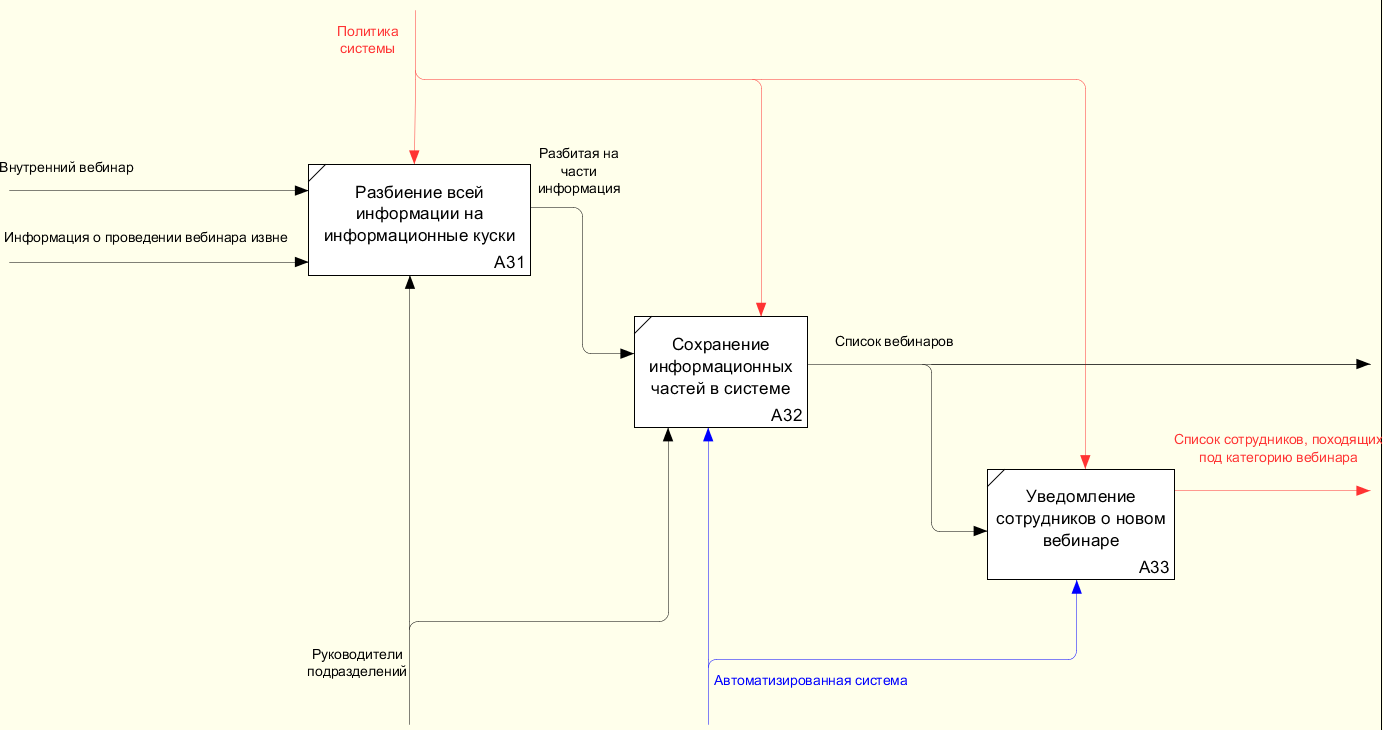


Рисунок 2.8 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*TO-BE*» (разбиение бизнес-процесса «Внесение информации о проводящихся вебинарах в систему»)

Бизнес-процесс «Разбиение всей информации на информационные куски» подразумевает преобразование утвержденного внутреннего вебинара или информации о проведении внешнего вебинара на малые информационные части, которые можно записать в систему.

В результате внутренний вебинар и информация о проведении вебинара извне преобразуется в разбитую на части информацию

Механизмом является руководители подразделений

Управляется политикой системы.

Бизнес-процесс «Сохранение информационных частей в системе» подразумевает непосредственное сохранение информации в базе данных, для последующего отображения ее сотрудникам предприятия

В результате разбитая на части информация преобразуется в список вебинаров.

Механизмом является автоматизированная система и руководители подразделений.

Управляется политикой системы.

Бизнес-процесс «Уведомление сотрудников о новом вебинаре» подразумевает передачу сотрудникам компании информации о появлении нового вебинара по темам, которые они указали в своих настройках.

В результате список вебинаров преобразуется в список сотрудников, подходящих под категорию вебинара.

Механизмом является автоматизированная система.

Управляется политикой системы.

Второй уровень диаграммы – это разбиение бизнес-процесса «Выбор сотрудниками вебинаров для посещения» на 4 подпроцесса.

Этот уровень представлен на рисунке 2.9.

Бизнес-процессы, представленные в рамках данной диаграммы:

* Получение списка вебинаров выбранного в настройках типа
* Выбор понравившегося вебинара
* Чтение информации о проведении
* Запись на участие в вебинаре

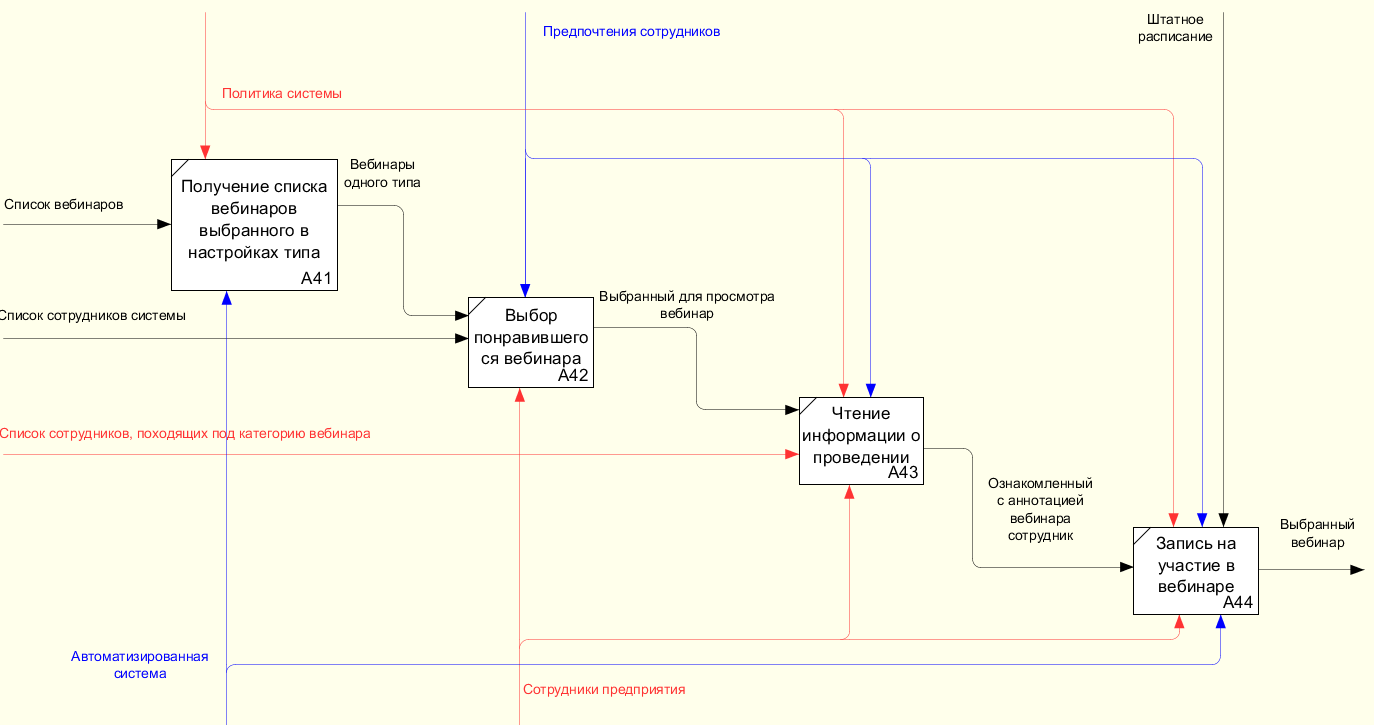


Рисунок 2.9 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*TO-BE*» (разбиение бизнес-процесса «Выбор сотрудниками вебинаров для посещения»)

Бизнес-процесс «Получение списка вебинаров выбранного в настройках типа» подразумевает отображение все вебинаров, темы которых попадают под категории, которые пользователь указал в настройках своего профиля.

В результате список вебинаров преобразуется в вебинары одного типа.

Механизмом является автоматизированная система.

Управляется политикой системы.

Бизнес-процесс «Выбор понравившегося вебинара» подразумевает выбор пользователя из списка представленных вариантов того, который он был бы не против посетить.

В результате вебинары одного типа и список сотрудников системы преобразуются в выбранный для просмотра вебинар.

Механизмом является сотрудник предприятия.

Управляется предпочтениями сотрудника.

Бизнес-процесс «Чтение информации о проведении» подразумевает ознакомление сотрудника с информацией о вебинаре, который тот выбрал для посещения

В результате выбранный для просмотра вебинар и список сотрудников, подходящих под категорию вебинара преобразуется в ознакомленного с аннотацией вебинара сотрудника.

Механизмом является сотрудник предприятия.

Управляется политикой системы и предпочтениями сотрудников.

Бизнес-процесс «Запись на участие в вебинаре» подразумевает согласие пользователя с условиями вебинара и подтверждение оного о будущем посещении.

В результате ознакомленный с аннотацией вебинара сотрудник преобразуется в выбранный вебинар.

Механизмом является сотрудник предприятия и автоматизированная система.

Управляется политикой системы, предпочтениями сотрудников, а также штатным расписанием.

Также второй уровень диаграммы – это разбиение бизнес-процесса «Проведение вебинара» на 4 подпроцесса. Этот уровень представлен на рисунке 2.10.

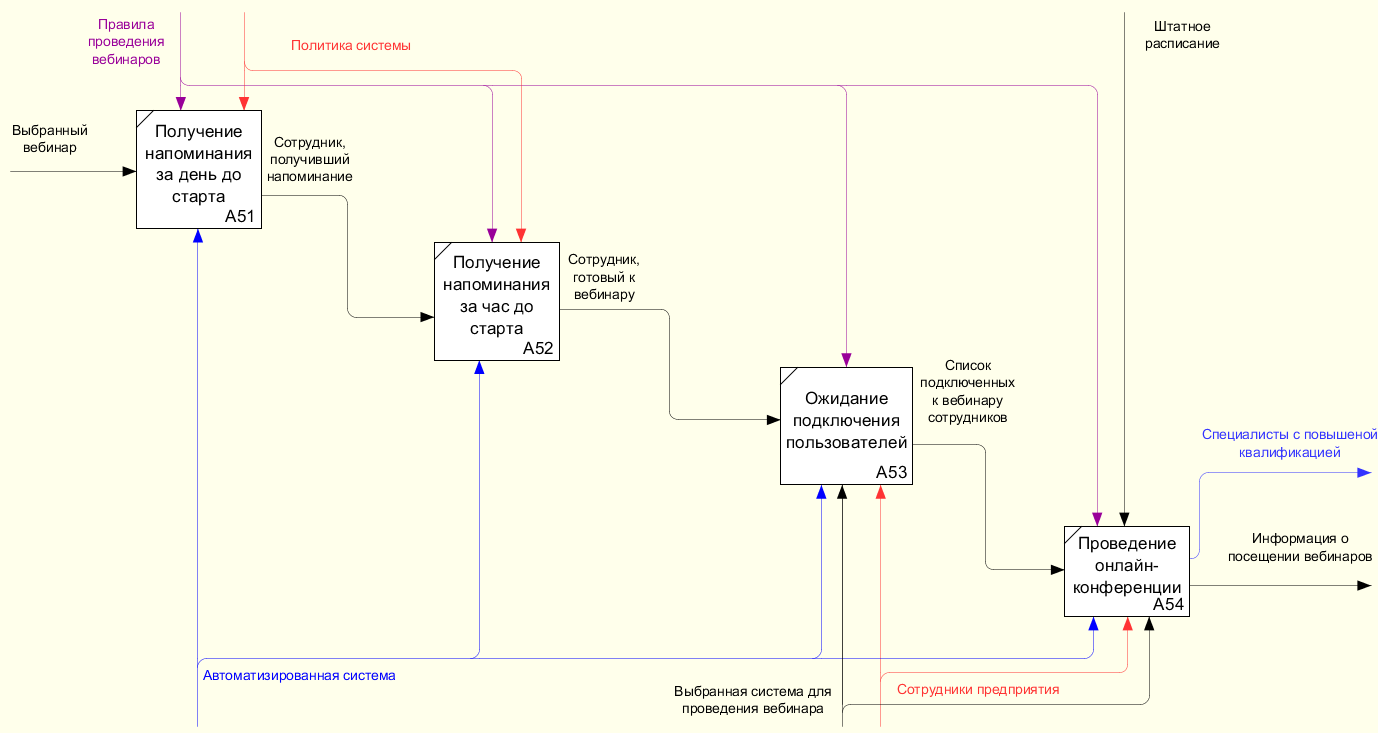


Рисунок 2.10 – Второй уровень функциональной модели системы в нотации *IDEF*0 модели «*AS*-*IS*» (разбиение бизнес-процесса «Проведение вебинара»)

Бизнес-процессы, представленные в рамках данной диаграммы:

* Получение напоминания за день до старта
* Получение напоминания за час до старта
* Ожидание подключения пользователей
* Проведение онлайн-конференции

Бизнес-процесс «Получение напоминания за день до старта» подразумевает уведомление пользователя о предстоящем вебинаре

В результате выбранный вебинар преобразуется в сотрудника, получившего напоминания

Механизмом является автоматизированная система.

Управляется правилами проведения вебинаров и политикой системы.

Бизнес-процесс «Получение напоминания за час до старта» подразумевает очередное уведомление пользователя о скором старте вебинара

В результате сотрудник получивший напоминание преобразуется в сотрудника, готового к вебинару.

Механизмом является автоматизированная система.

Управляется правилами проведения вебинаров и политикой системы.

Бизнес-процесс «Ожидание подключения пользователей» подразумевает уже старт вебинара, в котором перед выступлением спикера ожидается подключение большинства согласившихся сотрудников.

В результате сотрудник, готовый к вебинару, преобразуется в список подключенных к вебинару сотрудников.

Механизмом является автоматизированная система, сотрудники предприятия а также выбранная система для проведения вебинара.

Управляется правилами проведения вебинаров.

Бизнес-процесс «Проведение онлайн-конференции» подразумевает непосредственно саму онлайн-конференцию в выбранной системе проведения вебинаров, где сотрудников компании обучают новым знаниям.

В результате список подключенных к вебинару сотрудников преобразуется в специалистов с повышенной квалификацией и информацию о посещении вебинара

Механизмом является сотрудники предприятия и выбранная система для проведения вебинара.

Управляется штатным расписанием.

Данная функциональная модель представляет собой работу системы поддержки процесса организации вебинаров после внедрения программного средства.

На данном этапе были описаны две функциональные модели системы в нотации *IDEF*0 «*AS*-*IS*» и «*TO-BE*».

2.2. Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований

Перед тем как начать разрабатывать программное средство необходимо проанализировать требования к программному средству и поставить задачи.

Цель процесса анализа требований к программному средству заключается в установлении и документировании требований к программному обеспечению [15].

Разработка программного средства будет вестись на операционной системе Windows 10. Реализоввывать программное средство с использованием объектно-ориентированного языка программирования Java. Структура программного средства должна иметь классическую трехслойную архитектуру, включающую презентационную логику, бизнес-логику, слой доступа к данным. Реализацию абстрактного уровня доступа к данным осуществить на основе фреймворка Spring Boot. Для прозрачной передачи данных клиентам использовать фреймворк Spring Framework. При разработке пользовательского интерфейса использовать библиотеку Thymeleaf, фреймворк Bootstrap, а также JavaScript библиотеку JQuery. Взаимодействие между серверной и клиентскими частями должно осуществляться с использованием протокола HTTP.

Исполняемые файлы должны работать в среде OC Windows 7 и выше. Используемая система управления базой данных – MySQL 8. Язык интерфейса и элементов управления – русский. Программное средство должно запускаться без использования интегрированных средств разработки (IDE Eclipse).При разработке программного средства будут использованы следующие технологии: зык программирования Java, сервер базы данных MySQL. В качестве фреймворка будет выбран фреймворк Spring Framework.

Существует огромное количество инструментария для программирования как на стороне сервера, так и на стороне клиента, как платных, так и бесплатных.

Основные программные средства, которые будут использованы при разработке программного средства:

– интегрированная среда разработки Eclipse для разработки на языках программирования Java и JavaScript, языке разметки HTML и каскадных таблиц стилей CSS с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для Java;

– браузеры Opera и Google Chrome для проверки на кроссбраузерность программного средства, а также проверки результатов работы программного средства при разработке.

Все перечисленные выше серверные и клиентские технологии, а также соответствующий инструментарий идеально подходят для разработки веб-приложений.

Любой цикл разработки программного средства начинается с анализа требований. Цель этой стадии – определение детальных требований к программному средству.

Основные требования к программному средству системы управления персоналом предприятия представлены в таблице 1.

Таблица 2.1 – Основные требования к программному средству

|  |  |
| --- | --- |
| Вид требования | Содержание требования |
| Бизнес-требование (*B*1) | Все формы программного средства должны быть удобными для пользователей |
| Бизнес-требование (*B*2) | Для эксплуатации интерфейса программного средства от пользователей не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером |
| Нефункциональное требование (*NF*1) | Программное средство должно быть адаптивно под любое устройство |
| Нефункциональное требование (*NF*2) | Интерфейс программного средства должен быть на русском языке |
| Нефункциональное требование (*NF*3) | Цветовая гамма программного средства должна иметь четыре основных цвета: белый, серый, синий, голубой. |
| Функциональное требование (*F*1) | В программном средстве должно быть предусмотрено разделение ролей на гостя, пользователя, администратора |
| Функциональное требование (*F*2) | Не авторизовавшись в системе, пользователь должен иметь доступ к навигационной панели, которая обеспечивает переход к основным пунктам меню («Вход», «Регистрация» и главной странице) |
| Функциональное требование (*F*3) | Графическая оболочка страниц должна делиться на следующие разделы:  – навигационное меню;  – ссылка «Главная»;  – поле для отображения контента выбранной страницы |
| Функциональное требование (*F*4) | Авторизованный пользователь, при нажатии на «Профиль», должен переходить в личный кабинет |

Требования к программному средству определены

2.3. Информационная модель и ее описание

В данном подразделе будет представлено описание информационной модели данных и ее структура, которая обеспечивает работу системы. При этом данное описание будет, как в текстовом формате, так и при помощи диаграммы в нотации IDEF1X в MYSQL Workbench.

IDEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы [16]. Концептуальной схемой называется универсальное представление структуры данных в рамках коммерческого предприятия, независимое от конечной реализации базы данных и аппаратной платформы.

Основными компонентами информационной модели в нотации IDEF1X являются сущности, отношения и атрибуты. Для этих компонентов приняты специальные средства графического изображения.

Сущность определяют, как множество объектов, обладающих общими свойствами.

Конкретные элементы этого множества называют экземплярами сущности. Если сущность A может быть определена только с помощью ссылки на свойства некоторой другой сущности B, то A называют дочерней сущностью, а B выступает в роли родительской сущности.

Сущности в IDEF1X-диаграммах изображают в виде прямоугольников, причем рекомендуется у зависимых сущностей углы прямоугольников изображать скругленными.

Отношения между сущностями в IDEF1X являются бинарными отношениями. Выделяют идентифицирующие отношения – связи типа родитель-потомок, в которых потомок однозначно определяется своей связью с родителем, и неидентифицирующие отношения, означающие, что у связанного этим отношением экземпляра одной сущности может быть, а может и не быть соответствующего экземпляра второй сущности.

То есть это нотация, которая отображает логическую модель некоторой модели данных.

Логическая модель данных является визуальным представлением структур данных, их атрибутов и бизнес-правил. Логическая модель представляет данные таким образом, чтобы они легко воспринимались бизнес-пользователями.

Проектирование логической модели должно быть свободно от требований платформы и языка реализации или способа дальнейшего использования данных.

Разработчик модели использует требования к данным и результаты анализа для формирования логической модели данных. Разработчик приводит логическую модель к третьей нормальной форме и проверяет ее на соответствие корпоративной модели данных, если она существует.

Информационная модель в нотации IDEF1X представлена на рисунке 2.11.

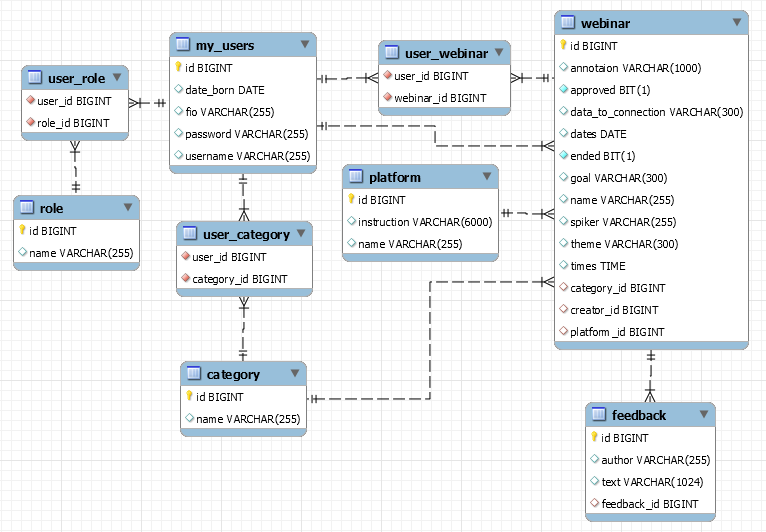


Рисунок 2.11 – Информационная модель

В данной информационной модели присутствует шесть сущностей и вспомогательные таблицы.

Перечень таблиц, используемых в текущем проекте:

Категории вебинаров. Хранит список категорий вебинаров.

Платформы вебинаров. Хранит список платформ вебинаров, а также краткую инструкцию по подключению к ним.

Вебинар. Хранит основную информацию о вебинаре.

Роль. Хранит роль пользователя.

Пользователь. Таблица пользователя с его личными данными.

Отзывы. Таблица отзывов об вебинарах.

Список желаний. Таблица, хранящая значения, какие пользователи на какие вебинары записались.

Список посещенных. Таблица, хранящая значения о посещенных пользователями вебинарах.

Ниже будут приведены описания полей всех таблиц в базе данных.

Категории вебинаров

* ИД
* Название

Платформы вебинаров

* ИД
* Название
* Инструкция по подключению

Вебинар

* ИД
* Тема
* Цели и задачи
* Спикер
* Категория
* Дата и время проведения
* Аннотация
* Данные для подключения

Роль

* ИД
* Название

Пользователь

* ИД
* Логин
* Пароль
* ФИО
* Дата рождения
* Роль
* Темы вебинаров для показа

Отзывы

* ИД
* ИД вебинара
* ИД пользователя
* Отзыв

Список желаний

* ИД
* ИД вебинара
* ИД пользователя

Список посещенных

* ИД
* ИД вебинара
* ИД пользователя

На данном этапе была построена информационная модель в нотации IDEF1X. Описаны сущности и их атрибуты. Для информационной системы была разработана база данных, которая будет в качестве СУБД использовать MYSQL. Данная база данных позволяет эффективно хранить данные и получать к ним доступ, обеспечивает надежность и гибкость использования.

2.4. Модели представления программного средства

Спецификация программного средства содержит описание или ссылки на описания исполняемого программного обеспечения, исходных файлов и информацию о программной реализации, включая информацию проекта построения, компиляции, построения и процедуры модификации.

Описание спецификации вариантов использования будет произведено двумя способами:

– текстовый вариант, описывающий в текстовом виде возможности системы;

– графический вариант при помощи диаграммы UML 2.0 – диаграммы вариантов использования [17].

Каждый вариант использования – это некоторое действие, которое доступно некоторому человеку, или актёру. То есть, это некоторый логически завершённый вариант работы системы.

Актёр – это некоторая обобщённая роль человека в системе, характеризующаяся конкретными привилегиями и возможностями в данной системе.

Актёры и варианты использования – основы диаграммы вариантов использования.

Диаграмма вариантов использования (Use Case) в UML – это диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [18].

Прецедент – это возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе [19].

Для данного проекта диаграмма вариантов использования представлена на рисунке 2.12.

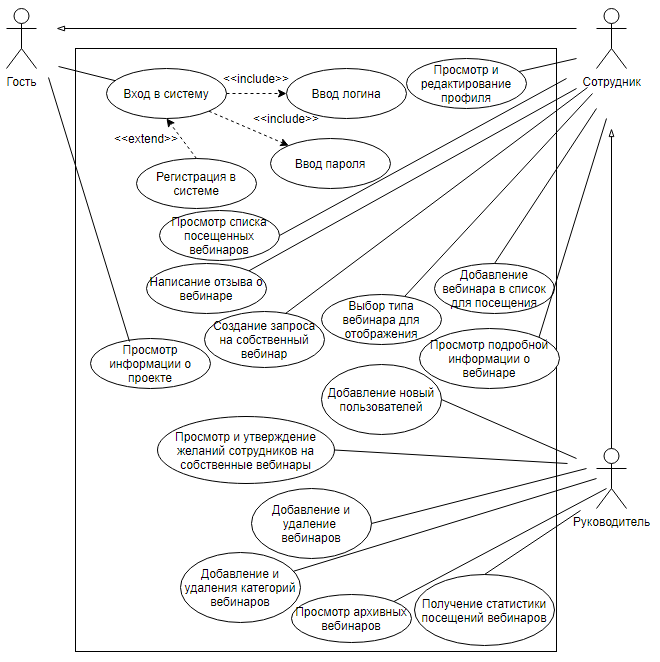


Рисунок 2.12 – Диаграмма вариантов использования

Данная диаграмма демонстрирует все варианты использования системы всеми основными актёрами.

При этом применяются четыре основных типа отношений [20]:

– отношение включения (include), которое показывает зависимость между базовым вариантом использования и его специальным случаем;

– отношение расширения (extend), которое определяет взаимосвязь базового варианта использования с другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении дополнительных условий;

– отношение ассоциации, которое служит для обозначения специфической роли актера при его взаимодействии с отдельным вариантом использования;

– отношение обобщения, которое показывает, что два и более актера могут иметь общие свойства, то есть взаимодействовать с одним и тем же множеством вариантов использования одинаковым образом.

Диаграмма вариантов использования включает три актера. Перечень актеров:

* гость
* сотрудник
* руководитель

Гостю доступны следующие возможности:

* вход в систему
* регистрация в системе
* просмотр информации о проекте

У всех актеров кроме гостя есть возможность входа в личный кабинет после авторизации в системе.

Сотрудник имеет все возможности гостя, а также, после входа в систему, ему доступны следующие возможности:

* Просмотр и редактирование профиля
* Просмотр списка посещенных вебинаров
* Написание отзыва о вебинаре
* Создание запроса на собственный вебинар
* Выбор типа вебинаров для отображения
* Добавление вебинара в список для посещения
* Просмотр подробной информации о вебинаре

Руководитель имеет все возможности сотрудника, а также ему доступны следующие:

* Добавление новых пользователей
* Добавление и удаление вебинаров
* Добавление и удаление категорий вебинаров
* Просмотр архивных вебинаров
* Получение статистики посещений вебинаров

Таким образом, описание всех вариантов использования системы и актёров завершено. Были описаны все варианты использования системы для каждой из ролей. На этом описание подраздела завершено.

## 3. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА ЦЕЛЕВОГО УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ ПРЕДПРИЯТИЯ СФЕРЫ УСЛУГ.

3.1. Архитектурные решения.

Среда Spring Web MVC предоставляет архитектуру модель-представление-контроллер и готовые компоненты, которые можно использовать для разработки гибких и слабо связанных веб-приложений. Шаблон MVC приводит к разделению различных аспектов приложения (логика ввода, бизнес-логика и логика пользовательского интерфейса), обеспечивая при этом слабую связь между этими элементами [2].

* Модель инкапсулирует данные приложения и, как правило, они состоят из POJO .
* Представление отвечает за визуализацию данных модели и, в общем, генерирует вывод HTML, который может интерпретировать браузер клиента.
* Контроллер отвечает за обработку пользовательских запросов и построение соответствующей модели и передает ее в представление для визуализации.

Модель инкапсулирует данные приложения и, как правило, они состоят из POJO .

Представление отвечает за визуализацию данных модели и, в общем, генерирует вывод HTML, который может интерпретировать браузер клиента.

Контроллер отвечает за обработку пользовательских запросов и построение соответствующей модели и передает ее в представление для визуализации.

Среда Spring Web Model-View-Controller (MVC) разработана на основе DispatcherServlet, который обрабатывает все HTTP-запросы и ответы. Рабочий процесс обработки запросов Spring Web MVC DispatcherServlet показан на следующем рисунке.

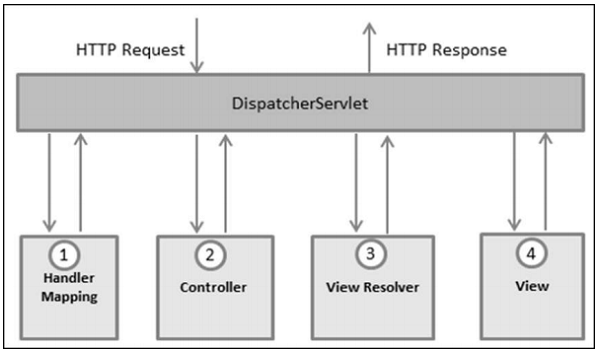


Рисунок 3.1 – DispatcherServlet

Ниже приведена последовательность событий, соответствующая входящему HTTP-запросу в DispatcherServlet.

* После получения HTTP-запроса DispatcherServlet обращается к HandlerMapping для вызова соответствующего контроллера.
* Контроллер принимает запрос и вызывает соответствующие методы обслуживания на основе используемого метода GET или POST . Сервисный метод устанавливает данные модели на основе определенной бизнес-логики и возвращает имя представления в DispatcherServlet.
* DispatcherServlet примет помощь от ViewResolver, чтобы подобрать определенное представление для запроса.
* После того, как представление завершено, DispatcherServlet передает данные модели представлению, которое, наконец, отображается, в браузерах.

После получения HTTP-запроса DispatcherServlet обращается к HandlerMapping для вызова соответствующего контроллера.

Контроллер принимает запрос и вызывает соответствующие методы обслуживания на основе используемого метода GET или POST . Сервисный метод устанавливает данные модели на основе определенной бизнес-логики и возвращает имя представления в DispatcherServlet.

DispatcherServlet примет помощь от ViewResolver, чтобы подобрать определенное представление для запроса.

После того, как представление завершено, DispatcherServlet передает данные модели представлению, которое, наконец, отображается, в браузерах.

Все вышеупомянутые компоненты, то есть HandlerMapping, Controller и ViewResolver, являются частями WebApplicationContext , который является расширением простого ApplicationContext с некоторыми дополнительными функциями, необходимыми для веб-приложений.

DispatcherServlet делегирует запрос контроллерам для выполнения специфической для него функциональности.

Аннотация **@Controller** указывает, что определенный класс выполняет роль контроллера. Аннотация **@RequestMapping** используется для сопоставления URL либо с целым классом, либо с конкретным методом-обработчиком. Аннотация **@Controller** определяет класс как контроллер Spring MVC [22].

Следующая аннотация **@RequestMapping (method = RequestMethod.GET)** используется для объявления **метода printHello () в** качестве метода службы контроллера по умолчанию для обработки HTTP-запроса GET. Мы можем определить другой метод для обработки любого запроса POST по тому же URL.

Атрибут **value** указывает URL-адрес, с которым сопоставляется метод обработчика, а атрибут **метода** определяет метод службы для обработки запроса HTTP GET.

Ниже приведены некоторые важные моменты, касающиеся контроллера, определенного выше:

* Вы определите необходимую бизнес-логику внутри метода службы. Вы можете вызвать другой метод внутри этого метода согласно требованию.
* На основе определенной бизнес-логики вы создадите модель в этом методе. Вы можете установить различные атрибуты модели, и эти атрибуты будут доступны представлению для представления результата. В этом примере создается модель с атрибутом «сообщение».
* Определенный метод службы может возвращать строку, которая содержит имя **представления,** которое будет использоваться для визуализации модели. В этом примере возвращается «привет» в качестве имени логического представления.

Вы определите необходимую бизнес-логику внутри метода службы. Вы можете вызвать другой метод внутри этого метода согласно требованию.

На основе определенной бизнес-логики вы создадите модель в этом методе. Вы можете установить различные атрибуты модели, и эти атрибуты будут доступны представлению для представления результата. В этом примере создается модель с атрибутом «сообщение».

Определенный метод службы может возвращать строку, которая содержит имя **представления,** которое будет использоваться для визуализации модели.

На рисунке 3.2 представлена диаграмма последовательности, которая представляет собой один конкретный экземпляр работы программы под управлением пользователя.

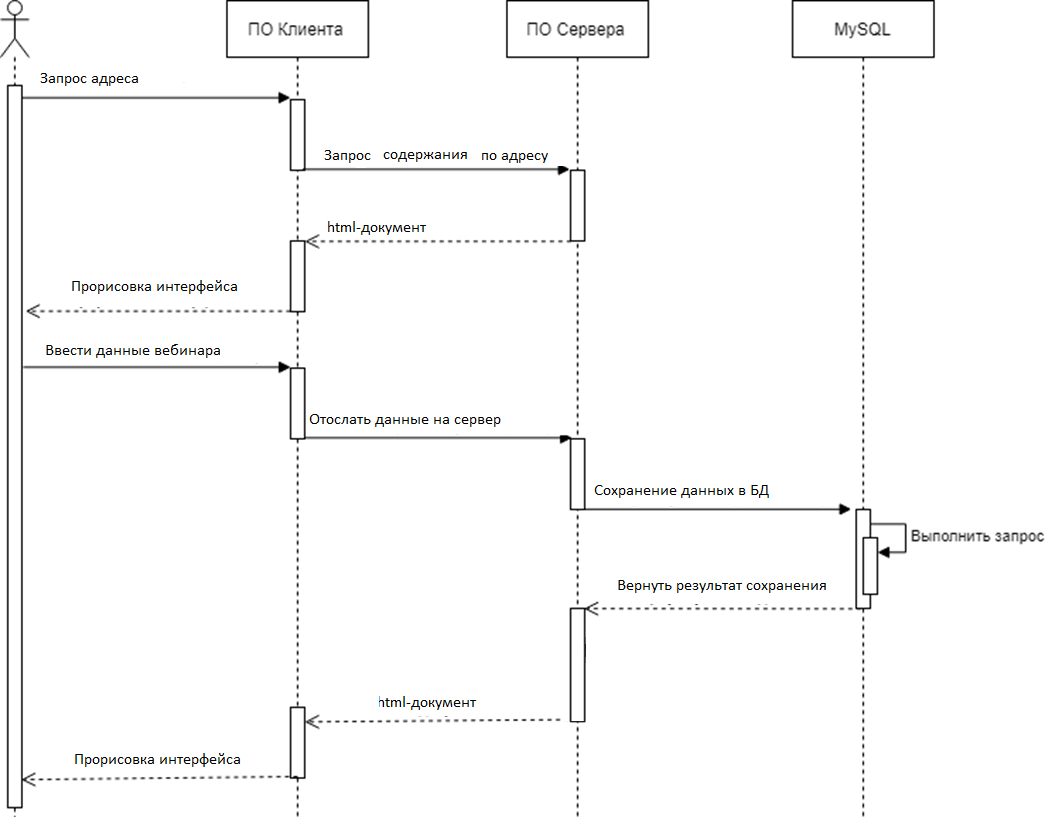
****

Рисунок 3.2 – Диаграмма последовательности

Диаграмма последовательности – это диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание – деятельность – уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках какого-либо определённого прецедента (отправка запросов и получение ответов) [23-24].

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни», отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

На диаграмме последовательности изображен один актёр – пользователь системы, а также следующие объекты:

– ПО Клиента;

– ПО Сервера;

– MySQL.

Представлена последовательность действия в три этапа:

1. Пользователь вводит адрес сайта в браузере, а сервер отправляет клиенту html-документ.

2. Клиент вводит данные вебинара для сохранения его в системе. Браузер посылает данные серверу, а сервер посылает запрос на сервер MySQL на сохранение данных, который полученные данные пытается сохранить в системе. Сервер результат возвращает клиентскому браузеру. Браузер отображает информацию для клиента.

3.2. Описание алгоритмов, реализующих бизнес-логику разрабатываемого программного средства.

Модели представления системы представляют собой некоторое формальное описание системы посредствам некоторых блок-схем, диаграмм или прочих способов иллюстрирования. Соответственно, данный подраздел и посвящён описанию этих моделей представления информации.

Система будет описана тремя графическими представлениями, которые приведены далее:

* диаграмма состояний;
* диаграмма классов;
* схема алгоритма.

Первая диаграмма – это диаграмма состояний. Диаграмма состояний – это диаграмма, которая определяет множество систем: от компьютерных программ до бизнес-процессов [25].

Условные обозначения:

* Круг – обозначает начальное состояние.
* Окружность с маленьким кругом внутри – обозначает конечное состояние (если есть).
* Скруглённый прямоугольник – обозначает состояние. Верхушка прямоугольника содержит название состояния. В середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии.
* Стрелка – обозначает переход. Название события (если есть), вызывающего переход, отмечается рядом со стрелкой.
* Толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих – обозначает объединение и разветвление соответственно.

Для данного дипломного проекта диаграмма состояний представлена на рисунке 3.3.

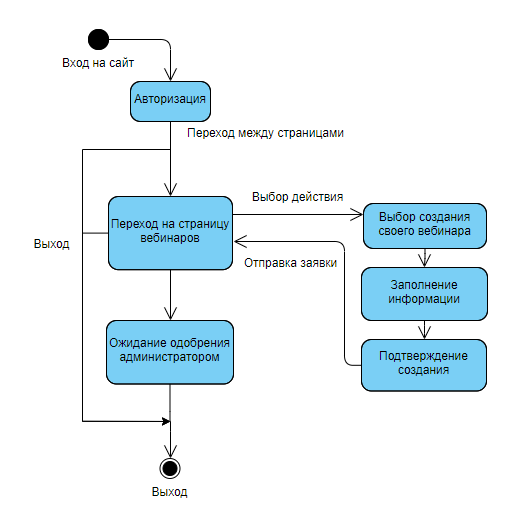


Рисунок 3.3 − Диаграмма состояний

На данной диаграмме представлен один из основных процессов – создание вебинара для проведения. Первое состояние – авторизация. После этого события пользователь переходит на страницу вебинаров и может воспользоваться функцией создать запрос на собственный вебинар или покинуть страницу. После выбора создания своего вебинара идет переход на соответствующую страницу, на которой нужно заполнить информацию и подтвердить создание. Затем пользователю остается лишь ожидать одобрения вебинара администратором, иначе говоря состояние подтверждения заказа переходит в состояние ожидания одобрения и цикл начинается сначала.

Далее следует рассмотреть диаграмму классов.

Диаграмма классов – структурная диаграмма языка моделирования *UML*, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования [26-27].

Диаграмма классов представлена в приложении Б.

Последняя схема данного подраздела – это схема алгоритма выбора пользователем понравившегося вебинара для участия. Данная схема алгоритма представлена на рисунке 3.4.

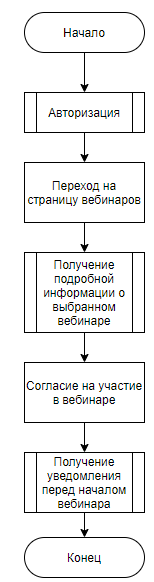


Рисунок 3.4 – Схема алгоритма выбора пользователем понравившегося вебинара для участия

Для начала пользователь заходит на сайт и авторизируется со своими личными данными. После чего, он переходит на страницу со списком всех вебинаров, которые будут проводиться. Для выбора вебинара, пользователь может получить подробную информацию о каждом из них, после чего он соглашается на участие в понравившихся вебинарах. После согласия на участия, за определенное время до старта вебинара, пользователь получает соответствующие уведомления о скором старте, на чем алгоритм завершается.

3.3. Проектирование пользовательского интерфейса.

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя и вычислительной системы [28].

ГОСТ «Эргономика взаимодействия человек-система», введенный в 2012 г., определяет пользовательский интерфейс (ПИ) как «компоненты интерактивной системы, предоставляющие пользователю информацию и являющиеся инструментами управления для выполнения определенных задач» [29].

Проектирование пользовательского интерфейса – это создание тестовой версии приложения. Это начальный этап разработки пользовательского интерфейса, когда распределяются функции приложения по экранам, определяются макеты экранов, содержимое, элементы управления и их поведение [30].

Пользователь при обращении с интерфейсом должен представить себе, какая информация о выполняемой задаче у него существует, и в каком состоянии находятся средства, с помощью которых он будет решать данную задачу. Эффективность работы пользователя и его интерес обеспечивает правильно сформулированная методика разработки и проектирования пользовательского интерфейса.

Именно поэтому необходимо большое внимание уделять процессу построения пользовательских интерфейсов (UI) и выстраиванию пользовательского опыта в целом (UX). Проектирование UI – это не разовая фаза проекта, это непрерывный итерационный процесс, в который вовлечены бизнес-пользователи, UX-инженеры, дизайнеры и программисты [31].

Первым делом спроектируем, как будет выглядеть страница для незарегистрированного пользователя

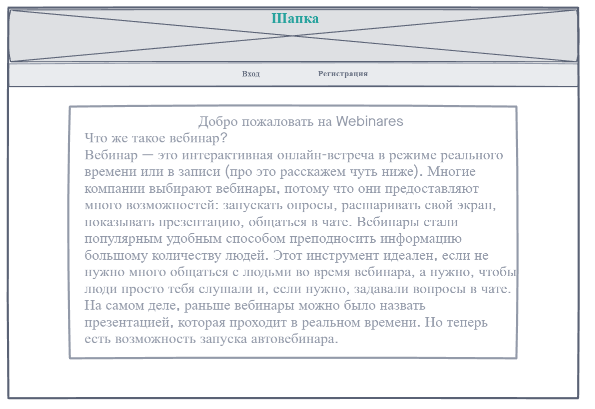


Рисунок 3.5 – Главная страница для незарегистрированного пользователя

Опираясь на требования, страница будет выглядеть следующим образом – в самом верху страницы будет находиться шапка сайта, которая будет одинакова для всех ролей. Ниже шапки будет расположено навигационное меню, пункты которого будут изменяться в зависимости от роли пользователя.

Сразу под навигационным меню будет расположено окно информации о проекте, что такое вебинары и для чего они. Видеть страницу сможет не только незарегистрированный пользователь, а также и все остальные роли, так как на нее можно перейти при нажатии на логотип сайта в меню навигации.

Для входа в системе, необходимо продумать страницу входа. Она будет выглядеть следующим образом.

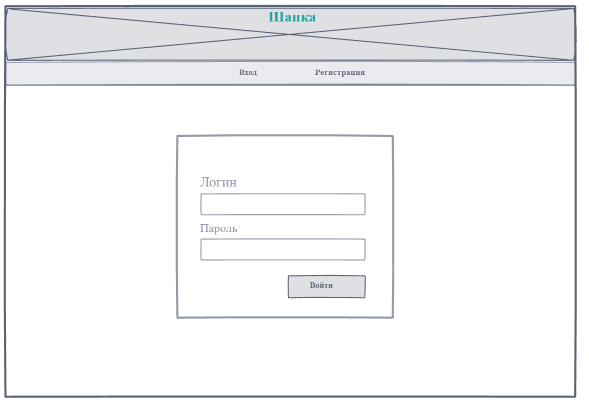


Рисунок 3.6 – Страница входа в систему

На ней пользователь, не важно с какой ролью зарегистрированный, будет вводить логин и пароль от своего аккаунта. Как только он его введет, либо произойдет переадресация на главную страницу с успешным входом, либо пользователь останется на этой странице и ему покажут уведомление о неправильном вводе.

Веб-приложению нужна не только авторизация, но еще и регистрация клиентов. Для этого предусмотрена страница регистрации пользователей. На ней они вводят необходимую информацию, которая будет использована в дальнейшем для работы с этим пользователем.

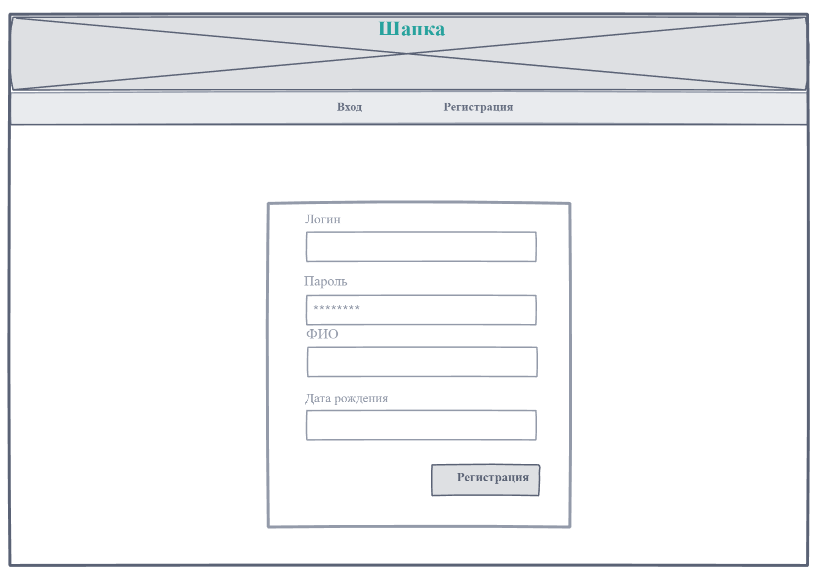


Рисунок 3.7 – Страница регистрации

На этой странице от пользователя будут получены его ФИО и дата рождения, не учитывая его логин и пароль для последующей авторизации. При успешной регистрации, пользователь получит соответствующее уведомление. Если по какой-либо причине регистрация не удалась, пользователь получит сообщение об ошибке и ему будет предложено попробовать зарегистрироваться еще раз.

После успешного входа в систему, если ролью пользователя является клиент, пункты навигационного меню показывают только те варианты, к которым у клиента есть доступ.

Так, клиенту будет иметь возможность перейти на следующие страницы: главная, вебинары, выбранные для посещения, моя история, создать свой. Пункт выхода из аккаунта подставляется в меню авторизованного пользователя независимо от его роли.

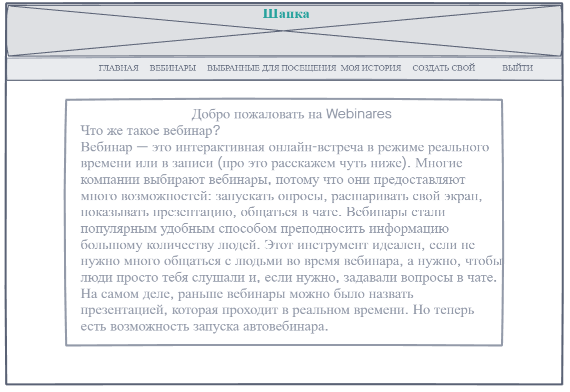


Рисунок 3.8 – Главная страница от имени пользователя

Страницы пользователя похожи между собой. Так, на странице вебинаров, он сможет видеть все доступные вебинары и либо перейти на страницу подробностей, либо сразу согласиться на участие в вебинаре.

Страница выбранных для посещения похожа на страницу вебинаров за исключением того, что вместо участвую у пользователя есть возможность отказаться от участия.

Страница истории выглядит практически так же, как и две предыдущих за исключением того, что пользователь больше не имеет возможности выбирать участвует он в вебинаре или нет, так как тот уже закончился, он имеет только один вариант взаимодействия – а именно посмотреть подробности вебинара.

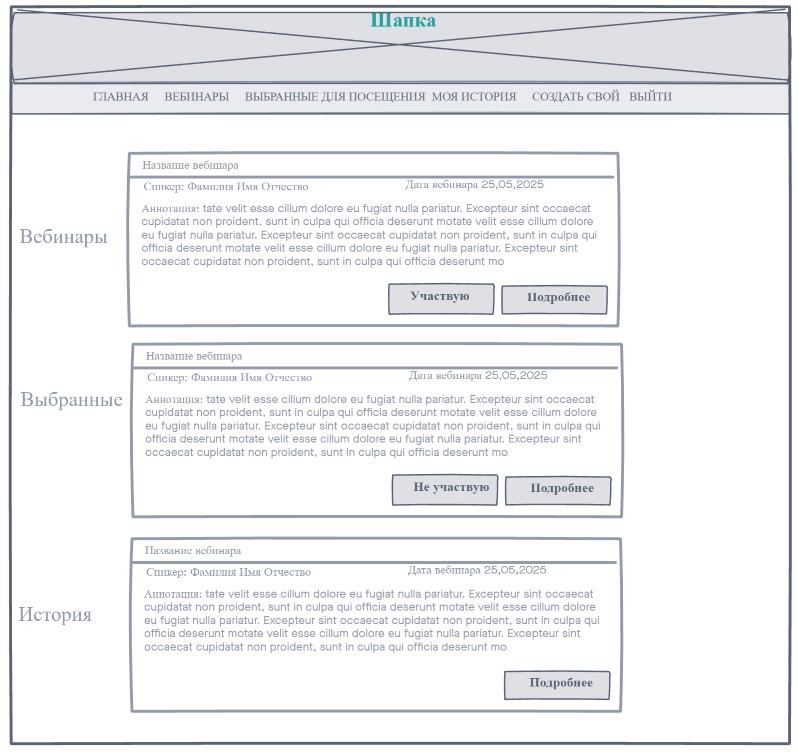


Рисунок 3.9 – Страницы отображения информации о вебинарах

Проектирование страниц для пользователя на этом закончено. Осталось проектирование видов страниц для администраторов. Вид навигационного меню для администратора будет выглядеть следующим образом:



Рисунок 3.10 – Вид меню от имени администратора

Возможностями администратора будет как работа с пользователями, так и непосредственно с заявками пользователей на вебинары.

При переходе на страницу пользователей, администратору будут доступна таблица зарегистрированных в системе пользователей с возможностью посмотреть из историю или удалить пользователя с системы



Рисунок 3.11 – Работа с пользователями

Для работы с запросами пользователей на личные вебинары предусмотрена отдельная страница, которая похожа на те, которые отображаются пользователям при выборе вебинара.

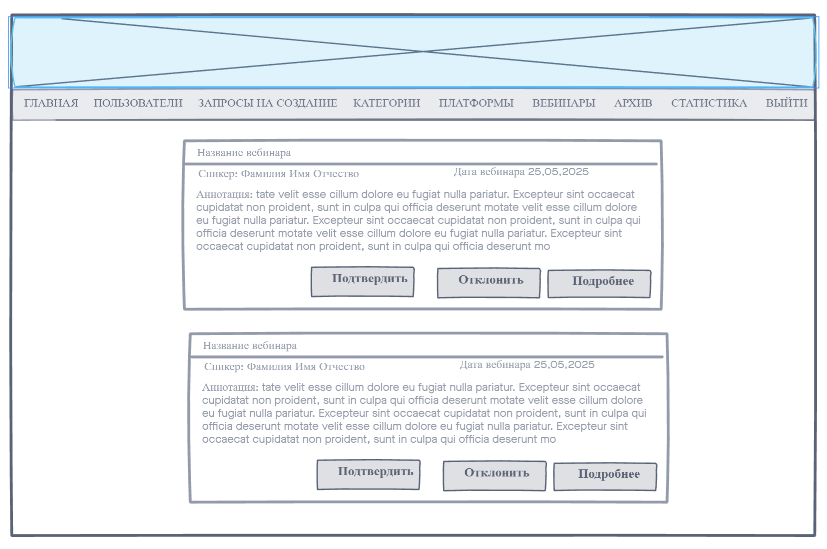


Рисунок 3.12 – Заявки пользователей

На этой странице администратор выбирает, будет ли запрошенный пользователем вебинар проводиться или нет.

Администратор также вправе создавать и удалять категории вебинаров, ровно так же, как и платформы их проведения.

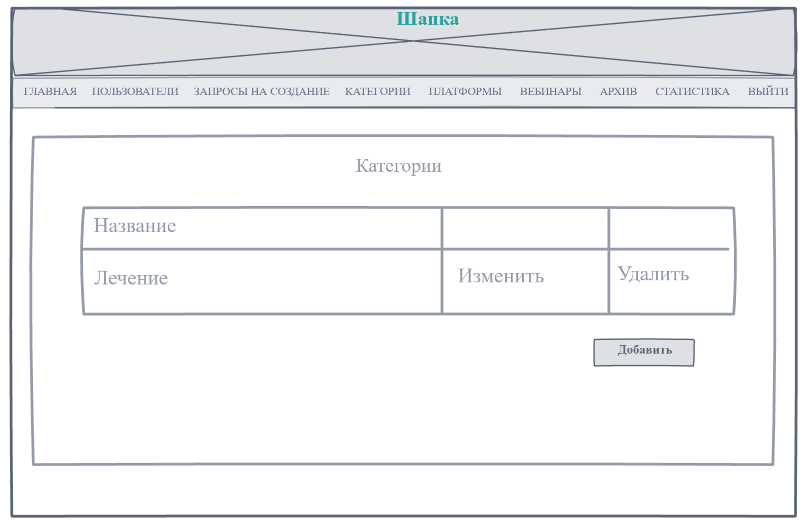


Рисунок 3.13 – Категории вебинаров

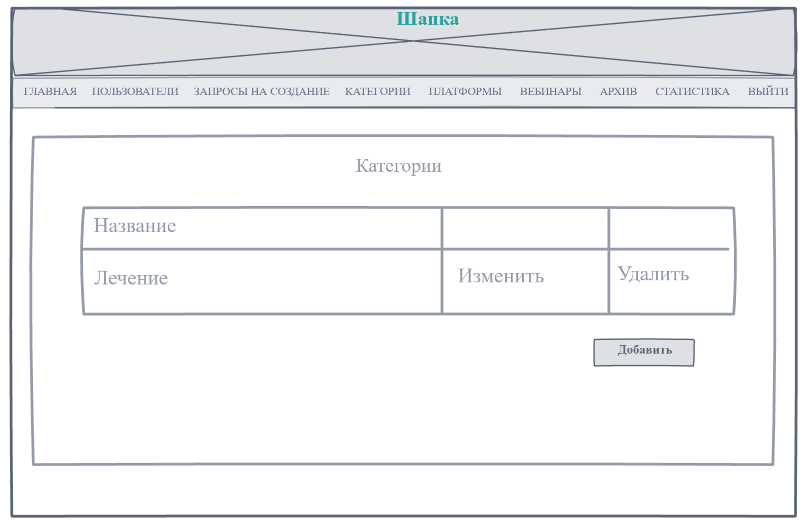


Рисунок 3.14 – Платформы вебинаров

Самым важным окном является окно создания (или редактирования) вебинара. На этой странице вводится вся нужная информация о предстоящем вебинаре, а также из списков платформ и категорий выбираются те данные, с помощью которых будет проводить вебинар.

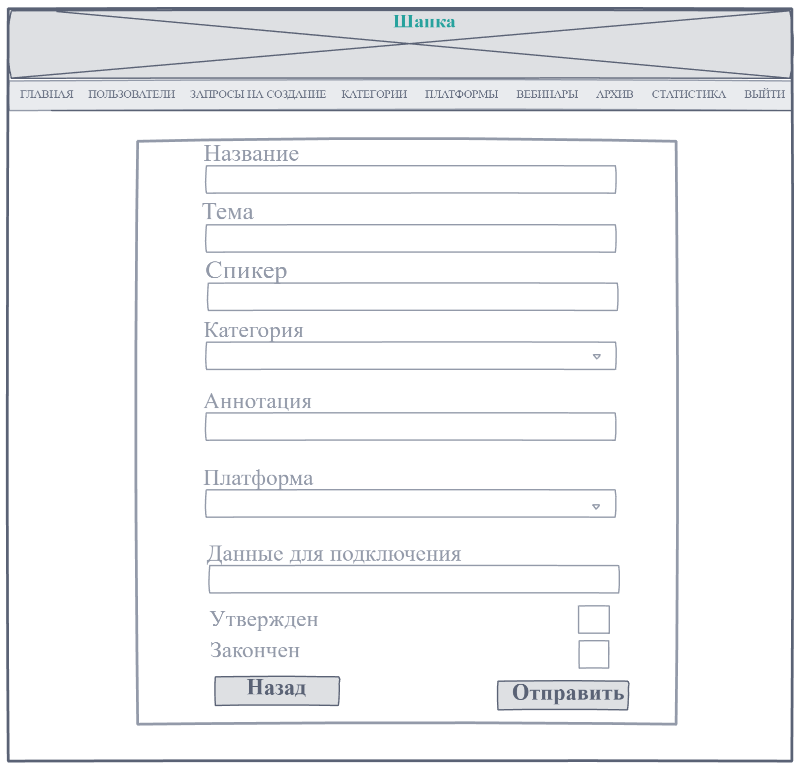


Рисунок 3.15 – Страница создания вебинаров

## 4. РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ПОДДЕРЖКИ ПРОЦЕССА ОРГАНИЗАЦИИ ВЕБИНАРОВ ДЛЯ ВЗАИМОДЕЙСТВИЯ КОРПОРАТИВНЫХ СОТРУДНИКОВ.

4.1. Обоснование выбора компонентов и технологий для реализации дипломного проекта.

При разработке программного средства необходимо определиться с компонентами и технологиями, которые необходимы для реализации дипломного проекта.

Архитектура данного проекта – это веб-приложение. Архитектура приложения во многом предопределило и технологии, используемые для построения проекта.

Основные технологии, применяемые при создании приложения:

– серверный язык *JAVA*;

– фреймворк *Spring*;

– таблица стилей *CSS*;

– клиентский язык *JavaScript*.

– язык запросов *SQL* (*MySQL*);

– язык разметки *HTML*;

Spring Framework (или Spring для краткости) - это универсальный фреймворк с открытым исходным кодом для платформы Java..NET каркас,Spring.NET [32].

Первая версия была написана родом Джонсоном, который впервые опубликовал свою книгу"Expert One-on-One Java Ee Design And Development"(Wrox Press, октябрь 2002).

Фреймворк был впервые выпущен в июне 2003 года под лицензией Apache2.0. Первая стабильная версия 1.0 была выпущена в марте 2004 года. Spring2. 0 был выпущен в октябре 2006 года, Spring2.5 был выпущен в ноябре 2007 года, Spring3.0 был выпущен в декабре 2009 года, а Spring3. 1 был выпущен в декабре 2011 года. Текущая версия - 5.2. икс

Хотя Spring не предоставляет какой-либо конкретной модели программирования, она стала распространенной в сообществе Java, главным образом как альтернатива и альтернатива корпоративной модели JavaBeans. Spring предоставляет разработчикам Java большую свободу проектирования; кроме того, он предоставляет хорошо документированные и простые в использовании инструменты для решения проблем при создании приложений корпоративного класса.

В то же время функциональность Spring Core подходит для любого Java-приложения, и существует множество расширений и улучшений для создания веб-приложений на платформе Java Enterprise. По этим причинам Spring приобрел большую популярность и считается разработчиками стратегически важным фреймворком [33].

Каждая выбранная технология отвечает за различные аспекты программы.

Появление языка Java в 1995-90-х-это, как правило, плодотворное для новых языков и концепций программирования. В Эдемском саду такого языка важно не заблудиться и не принять Святой Грааль за технику, которая не может пройти испытание временем. Java прошла тест, хотя и довольно давно. Не рекомендуется путать этот язык с JavaScript-они выглядят похожими, но это совершенно разные языки [34].

Вероятно, в Java впервые реализовали концепцию того, что язык должен быть максимально изолирован от платформы разработки, чтобы применять его без изменений везде: в компьютерах, часах, сотовых телефонах, бытовой технике. С «железной частью» должна была справляться виртуальная машина (JVM), которая, собственно, и создавалась индивидуально под каждое устройство. Сам же язык был неизменен и в качестве результата выдавал байт-код. С самого начала было известно, что код не может исполняться очень быстро, но многие устройства не требовали высокой скорости исполнения. Кроме того, со временем появились оптимизирующие компиляторы, так что, в среднем, программа на Java работает раза в 2-3 медленнее, чем на C++. Постоянное сравнение с C/C++ здесь не случайно: многие современные языки взяли за основу его конструкции и синтаксис, так что, бывает, узнать сходу язык очень трудно. Вместе с тем, Java с тех пор сильно «размножилась», и даже J#, J и прочие аналоги являются не родными братьями, а лишь подобием.

Сама идея языка, вполне, кстати, достаточного для создания софта любой сложности, была сначала не понята: был ли, мол, смысл создавать между аппаратурой и кодом промежуточные слои исполняющих машин? Со временем сомнения рассеялись: появилась мультиязычная платфома .

NET, и даже в Windows появились слои – аппаратно-зависимые, платформо-независимые. Самое же простое объяснение – софт стал очень сложным, а программисты очень ленивыми, чтобы переписывать программы под каждый отдельный аппарат.

Но вернемся к языку. Как уже говорилось, чем-то он похож на C++, чем-то на старый добрый Бейсик. Нет сейчас ни одного языка, который бы не хвалился своими возможностями ООП, и Java здесь не отличается от канонов: классы и объекты здесь используются везде, даже в самых примитивных задачах вроде вывода строки на экран. Из особенностей можно отметить, что все объекты в языке создаются только динамически, а все функции являются методами классов. Множественное наследование не поддерживается, как в C++, как и «опасные» указатели. ООП дает много преимуществ, но и требует слишком многого – в случае Java памяти устройства никогда не будет слишком много. В остальном же, имеются библиотеки классов для практически всех задач; преимущественно – под написание клиентских и серверных приложений. Хозяин Java – Oracle – успешно использует язык для использования в разработках своей одноименной СУБД. На сегодняшний день язык считается наиболее востребованным на рынке.

Следующие две технологии – это *HTML* и *CSS*, которые предопределены архитектурой проекта. Имеется единственное обоснование их выбора – это веб-архитектура разрабатываемого программного средства.

*HTML* – язык гипертекстовой разметки, который используется для структурирования и отображения веб-страницы и ее контента [35]. Для разработки данного программного средства будет применяться версия этого языка *HTML*5.

*CSS* – это формальный язык описания внешнего вида документа, написанного с использованием языка разметки [36]. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки *HTML.* Применяемая версия данной технологии – *CSS*3.

Последняя из основных применяемый технологий – это *JavaScript*. С помощью данного языка программирования, можно создавать динамически обновляемый контент, управлять мультимедиа, анимировать изображения. Что касается обоснований выбора в качестве клиентского языка *JavaScript*, то здесь выбор достаточно очевиден, ведь конкурентов у данного языка в данной сфере практически нет [37]. Лишь некоторые достоинства данного языка: возможность навигации и управления по *DOM HTML*-страницы, возможность управления браузером, гибкость подхода объектно-ориентированного программирования, поддержка асинхронности.

Для проектирования программного средства используется язык графического описания *UML [38]*.

*UML* – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

*UML* является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой *UML*-моделью. *UML* был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. *UML* не является языком программирования, на основании *UML*-моделей возможна генерация кода.

Следующим этапом данного пункта будет рассмотрение структуры программы посредствам следующих *UML*-диаграмм:

– диаграмма компонентов;

– диаграмма развёртывания;

– диаграмма последовательности.

Диаграмма компонентов – это элемент языка моделирования *UML*, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами [39].

В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и другое.

Диаграмма компонентов для текущего проекта представлена на рисунке 4.1. Как видно, система состоит из трёх компонентов:

– СУБД *MySQL*;

– серверное приложение;

– браузер клиента.

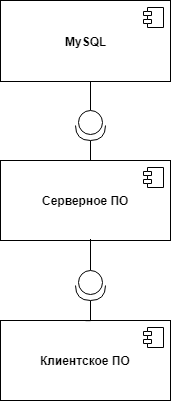


Рисунок 4.1 – Диаграмма компонентов

Каждый аспект отвечает за свой функционал. СУБД *MySQL* – за хранение и выборку данных, серверное приложение – за логику системы, а браузер клиента – за отображение информации.

4.2. Программная реализация алгоритмов, реализующих бизнес-логику программного средства.

В этом подразделе опишем, каким способом были реализованы алгоритмы работы с приложением.

Будем описывать реализацию алгоритма выбора пользователем понравившегося вебинара для участия. Заботу по авторизации берет на себя фреймворк спринг, так что мы сразу получаем авторизированного пользователя.

После авторизации, если пользователь только что был создан, пользователю не будет показано ни одного вебинара, поскольку категории вебинаров для показа настраиваются в личном кабинете. Первым делом пользователь переходит туда и выбирает из списка с многократным вариантом все категории вебинаров, которые он хочет видеть.

Как только он это сделает и сохранит, все данные отправляются в контролер на сервер.

Листинг 1. Обработка запроса сохранения категорий вебинаров для показа.

@RequestMapping("/edit")

**public** String edit(Model model,

@RequestParam(value = "categ", required = **true**) **int**[] categ,

Principal principal) **throws** Exception {

User user = userService.findByUsername(principal.getName());

**if**(categ!=**null**) {

user.getCategoriesToShow().clear();

**for**(**int** i=0; i<categ.length;i++)

{

user.getCategoriesToShow().add(categoryService.read(categ[i]));

}

}

userService.update(user);

model.addAttribute("success", "Выбранные категории сохранены");

**return** "redirect:/profile";

}

Идет получение массива ид категорий, которые записываются пользователю.

После настройки категорий для отображения, идет уже непосредственно само отображение списка вебинаров. Для того, чтобы категории совпадали с теми, что выбрал пользователь, они проходят через следующий метод контроллера

// список вебинаров кроме тех, на которые уже иду

@GetMapping

String getWebinaries(Model model, Principal principal) {

User user = serviceUser.findByUsername(principal.getName());

List<Webinar> list = service.repository.findAll();

**if** (list == **null**)

model.addAttribute("list", **null**);

**else** {

// цикл вебинаров с бд

**for** (**int** i = list.size() - 1; i >= 0; i--) {

**if**(list.size()<=i)**continue**;

// удаление не подтвержденных и тех, что уже прошли

**if** (!list.get(i).isApproved() || list.get(i).isEnded()) {

list.remove(i);

**continue**;

}

// цикл вебинаров для посещения пользователей, удаляем те, на которые уже идем

**for** (**int** j = 0; j < user.getWebinariesToGo().size(); j++) {

**if**(list.size()<=i)**continue**;

**if** (user.getWebinariesToGo().get(j).getId() == list.get(i).getId()) {

list.remove(i);

**continue**;

}

}

// проверяем, совпадает ли категория вебинаров с теми, что уже есть

**boolean** isWebinarInRightCategory = **false**;

**for** (Category cat : user.getCategoriesToShow()) {

**if**(list.size()<=i)**continue**;

**if** (cat.getId() == list.get(i).getId())

isWebinarInRightCategory = **true**;

}

**if**(list.size()<=i)**continue**;

**if** (!isWebinarInRightCategory)

list.remove(i);

}

model.addAttribute("isContains", **false**);

model.addAttribute("list", list);

}

**return** "webinariesToUser";

}

**4.3. Тестирование и проверка работоспособности программного средства.**

Для проверки работы спроектированного ПС в соответствии с требованиями ТЗ, разработаны тесты, с помощью которых можно оценить корректную работу веб-сервиса [40].

В таблице 4.1 приведены результаты тестирования для формы авторизации пользователя, которая отображается на главной странице веб-сервиса, так же тестировалась страница и форма регистрации, были рассмотрены более вероятные сценарии поведение пользователей.

Таблица 4.1 – Набор тестов авторизации и регистрации в веб-сервисе

|  |  |  |  |
| --- | --- | --- | --- |
| Место проведения теста | Содержание теста | Ожидаемый результат | Отметка о прохождении теста |
| Страница авторизации | Введены некорректные данные в любое из полей, нажата кнопка «Войти». | Перехода на другую страницу не происходит. В адресной строке появляется сообщение о ошибке. | Да |
| Страница авторизации | Введен корректный логин и пароль, нажата кнопка «Войти». | Отображение главной страницы сайта, отображение соответствующих функциональных возможностей. | Да |
| Страница с формой «Регистрация» | Введен существующий логин либо e-mail, нажата кнопка «Регистрация». | Переход на главную страницу, отображение сообщения об ошибке | Да |
| Страница с формой «Регистрация» | Введены некорректные данные в любое из полей. | Отображение информации об ошибке | Да |
| Страница с формой «Регистрация» | Введены корректные данные во все поля, нажата кнопка «Регистрация». | Перенаправление со страницы регистрации на главную страницу сайта, уведомление об успешной регистрации | Да |

Таблица 4.2 – Набор тестов для проверки возможностей администратора

|  |  |  |  |
| --- | --- | --- | --- |
| Место проведения теста | Содержание теста | Ожидаемый результат | Отметка о прохождении теста |
| Страница пользователей | Нажатие кнопки подробнее у каждого пользователя | Отображение под пользователем списка вебинаров, которые тот посетил | Да |
| Страница запросов на создание (запрос должен существовать) | Нажатие редактирования | Переход на страницу редактирования заказа на вебинар | Да |
| Страница категорий | Нажатие кнопки удалить | Выбранная категория удаляется | Да |
| Страница статистики | Вывод информации о посещениях вебинаров | Отображение информации о вебинарах. | Да |

В таблице 4.2 приведены результаты тестирования возможностей администратора веб-сервиса. Проверен функционал редактирования заказов не вебинар, работа с категориями, пользователями. Рассмотрены основные сценарии использования администратором.

Таблица 4.3 – Набор тестов для проверки возможностей пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| Место проведения теста | Содержание теста | Ожидаемый результат | Отметка о прохождении теста |
| Страница профиль | Выбор вариантов категорий из списка и отправка на сервер | После перезагрузки страницы выбранные категории сохраняются | Да |
| Страница вебинары | Нажатие кнопки подробнее у вебинара | Отображение вебинаров строго по критериями, переход на страницу подробностей | Да |
| Страница «Создать свой» | Заполнение информации и отправка заказа | Администратору в список запросов на вебинар добавляется только что созданный запрос | Да |

В таблице 4.3 приведены результаты тестирования возможностей пользователя веб-сервиса. Проверен функционал оформление заказа на бронирование столика. Рассмотрены основные сценарии использования возможностей веб-сервиса пользователем.

Таблица 4.4 – Набор тестов отображения информации

|  |  |  |  |
| --- | --- | --- | --- |
| Место проведения теста | Содержание теста | Ожидаемый результат | Отметка о прохождении теста |
| Все страницы | Перейти на любую из вкладок в произвольном порядке. | Корректное отображение информации, отображение того, чего ожидает пользователь. | Да |
| Все вкладки | Увеличение масштаба отображения средствами браузера. | Появление скроллов справа и снизу, корректное отображение информации. | Да |
| Все вкладки | Обновление страницы используя средства браузера. | Состояние веб-сервиса до обновления и после не изменилось. | Да |
| Меню браузера | Нажатие функциональных кнопок браузера «Назад» и «Вперед» в произвольный момент работы веб-приложения. | Корректное поведение веб-сервиса. Отображение информации. | Да |

Были проведены экспериментальные проверки на реальных данных. Разработанный веб-сервис полностью реализует все поставленные задачи. Весь реализованный функционал работает корректно.

4.4. Руководство по развертыванию и использованию программного средства.

Первое, что необходимо сделать, это привести описание запуска данного программного средства на хостинге без использования сторонних сред разработки.

Для установки веб-сервиса необходимым требованием является наличие Java Virtual Machine, Apache Tomcat 8, а также MySQL Server 8.0.1 в качестве целевой СУБД, maven 3.6 [41-43].

База данных для веб-сервиса генерируется сама после первого запуска приложения.

Для запуска приложения на клиентском компьютере, достаточно в консоли перейти в папку проекта и написать код

mvn clean package spring-boot:run [44]

Если запуск решено производить посредством war-архива, то его нужно переместить или скопировать в папку, где установлен Apache Tomcat 8. Далее необходимо запустить Apache Tomcat 8.

Для запуска веб-сервиса необходимо открыть браузер на вашем компьютере и написать в адресную строку следующий URL-адрес: <http://localhost:8080/> .

Далее будет рассмотрен интерфейс программного средства.

При запуске программного средства отображается главная страница. Главная страница состоит из навигационного меню, с помощью которого можно просмотреть информацию о сервисе, а также войти в систему или зарегистрироваться в ней. На рисунке 4.7 показана главная страница программного средства.

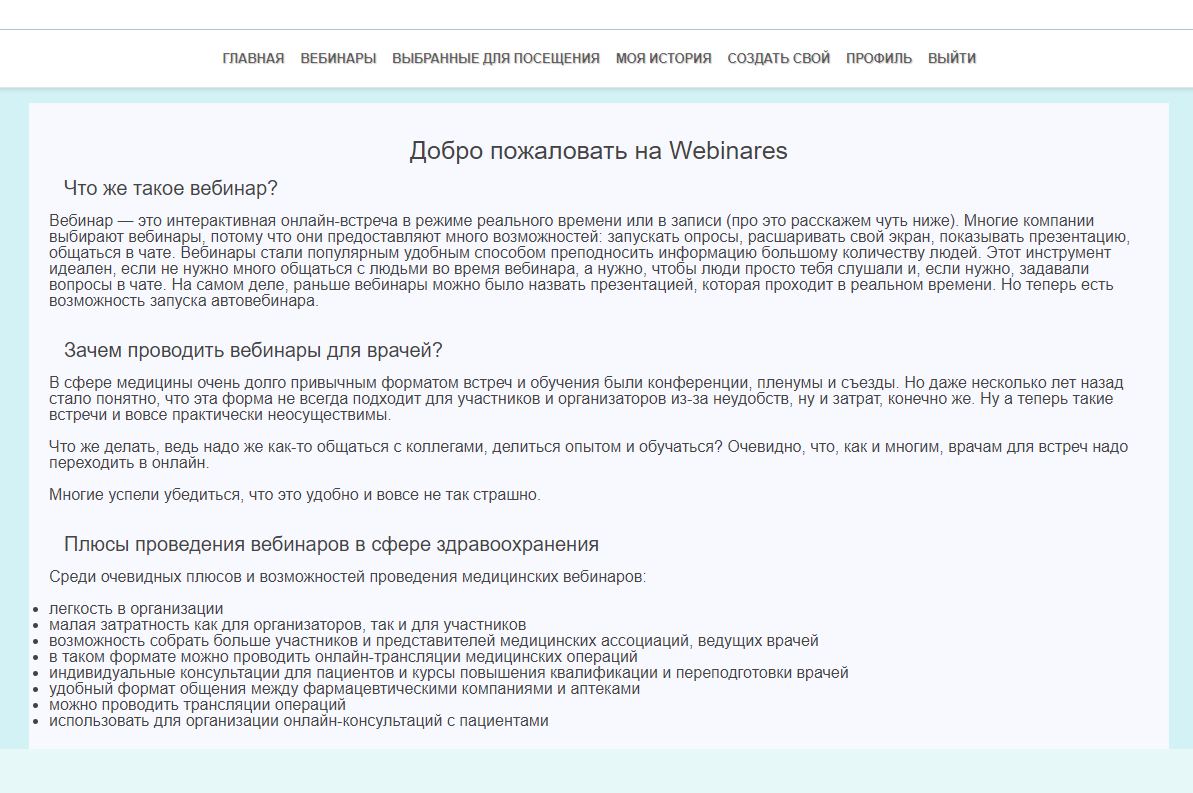


Рисунок 4.7 – Главная страница

Для осуществления любых операций, необходима авторизация в системе. Представим, что мы являемся новым пользователем, так что нам нужно также зарегистрироваться. Для этого переходим на страницу регистрации, показанную на рисунке 4.8 и вносим свои данные.

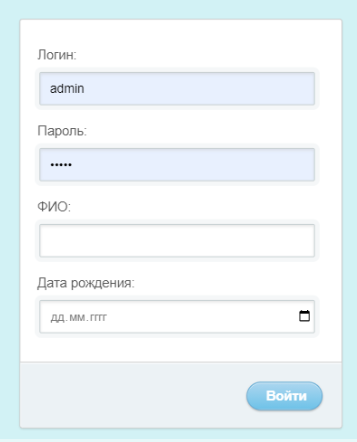


Рисунок 4.8 -Страница регистрации

После нажатия кнопки, нас переместит на главную страницу с уведомлением об успешной регистрации, и мы сможем войти в систему.

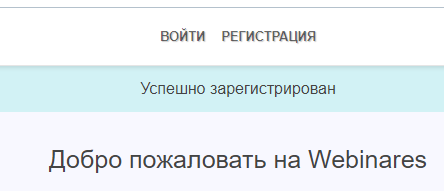


Рисунок 4.9 -Уведомление об успешной регистрации

Далее переходим на страницу входа и вводим свои данные.

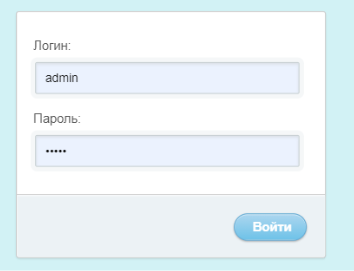


Рисунок 4.10 -Страница входа

После успешной авторизации, в зависимости от роли, мы переходим на главную страницу с меню обычного пользователя или администратора. Пока будем работать от лица администратора.

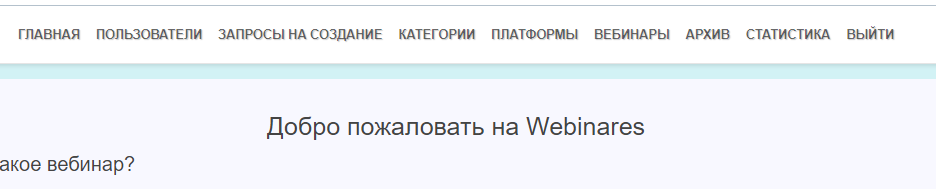


Рисунок 4.11 -Меню администратора

Пройдем по все меню по порядку. Администратору доступен просмотр списка всех пользователей системы с возможностью удалять пользователей.

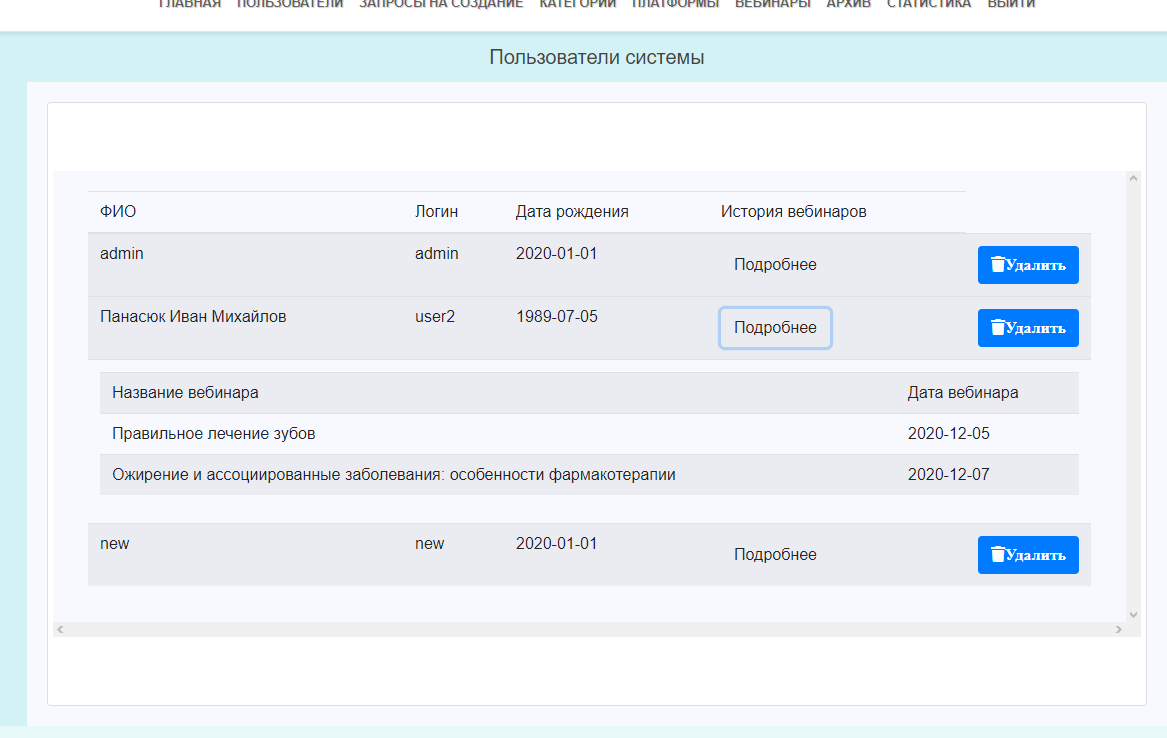


Рисунок 4.12 – Страница пользователей

Стоит упомянуть, что на этой странице доступен список каждого пользователя о посещенных им вебинарах. При необходимости, администратор может пользователя удалить.

Следующая страница – запросы на создание

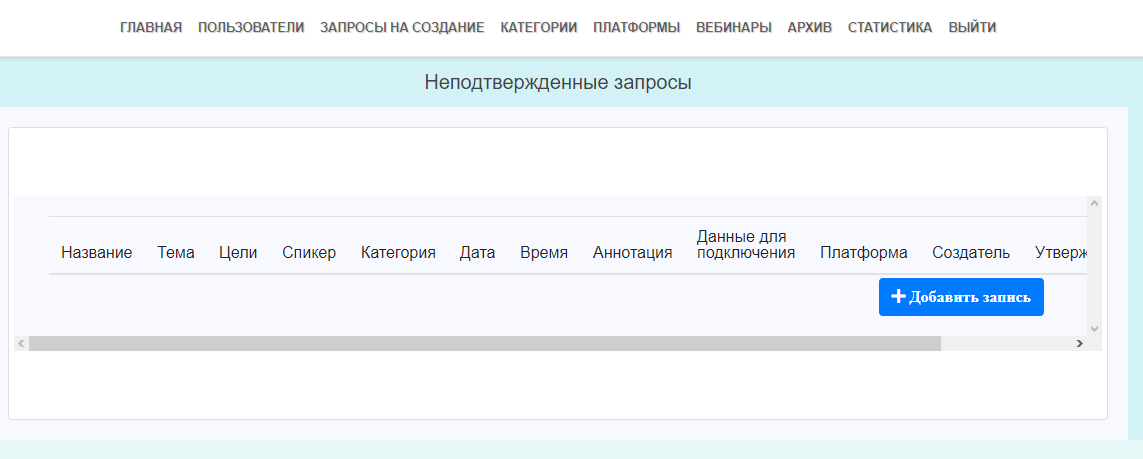


Рисунок 4.13 – Запросы на создание

Тут указаны те вебинары, которые хотят провести или о которых хотят рассказать пользователи. Вид страницы с вебинарами в большинстве случаев один и тот же. Такой же вид и у списка вебинаров.

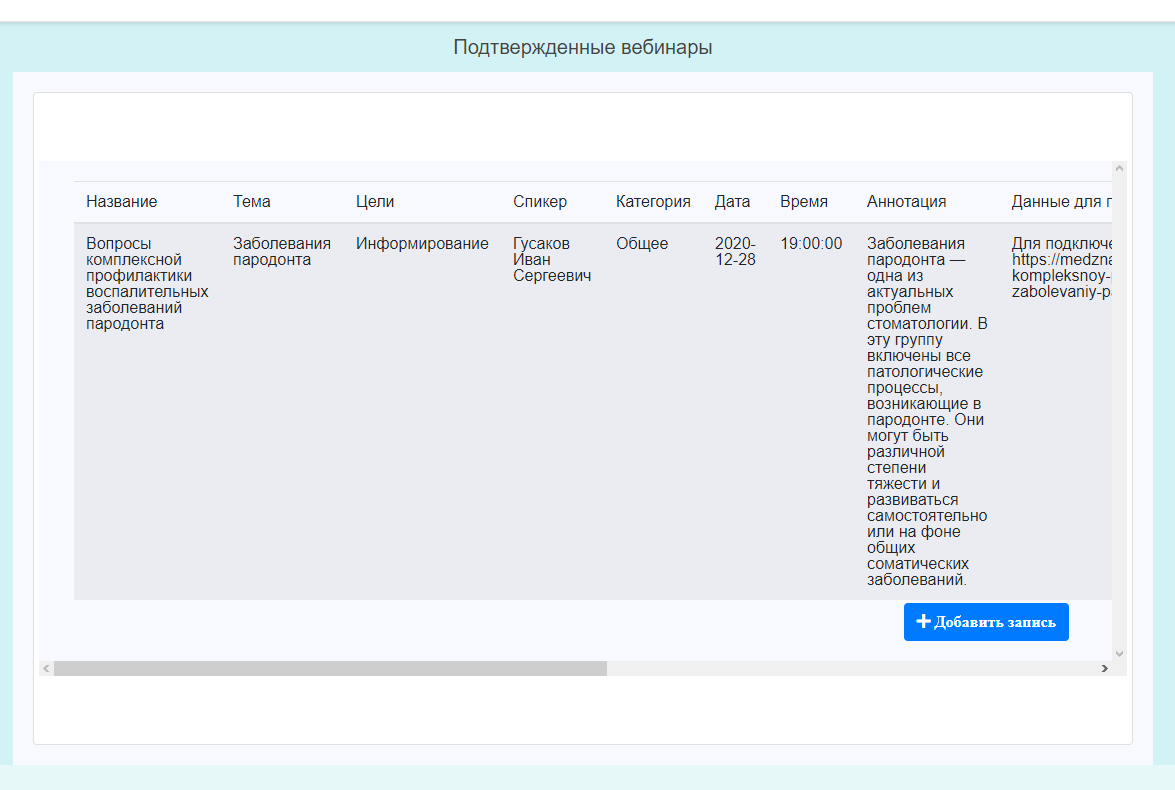


Рисунок 4.14 – Список вебинаров

Вся информация за раз не влазит, поэтому предусмотрен горизонтальный скролл внутри таблицы. При перемещении можно увидеть, что кнопки взаимодействия показаны именно там.

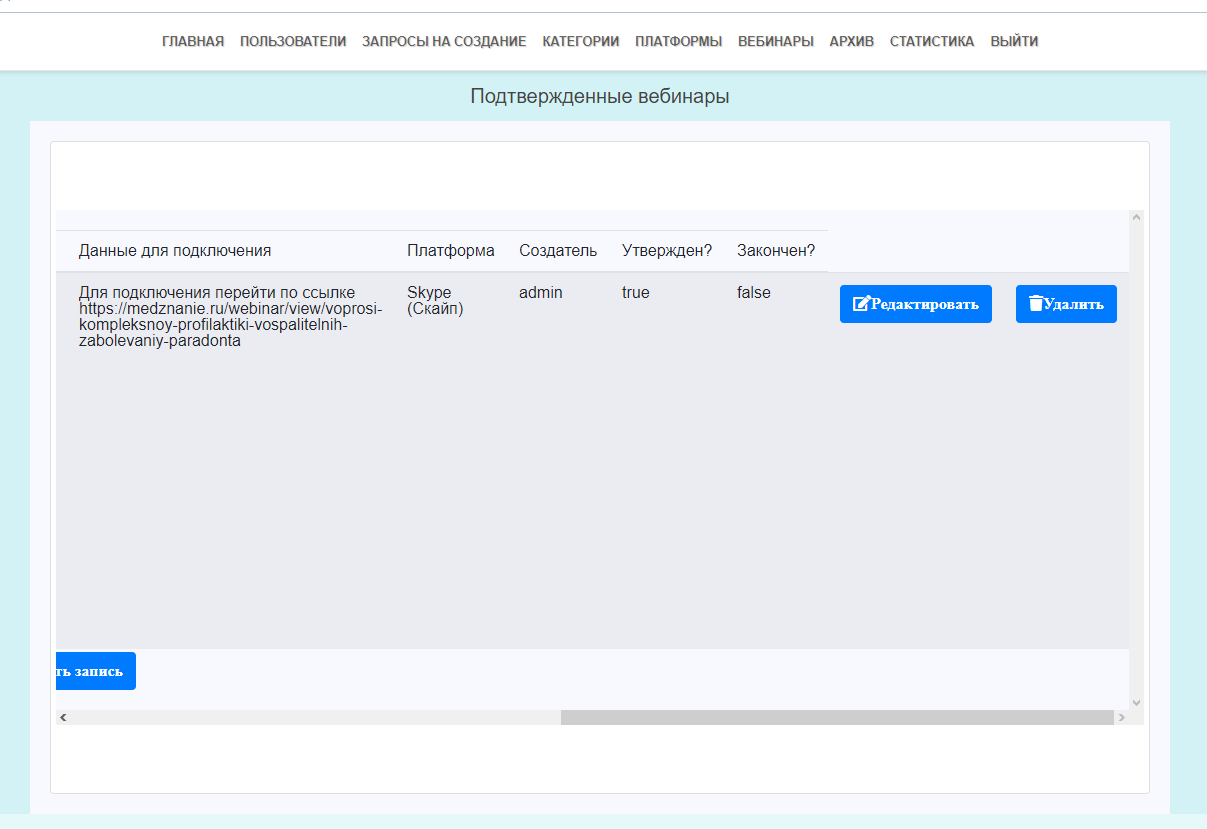


Рисунок 4.15 – Кнопки взаимодействия с вебинаром

Страница создания его или редактирования будет выглядеть следующим образом:

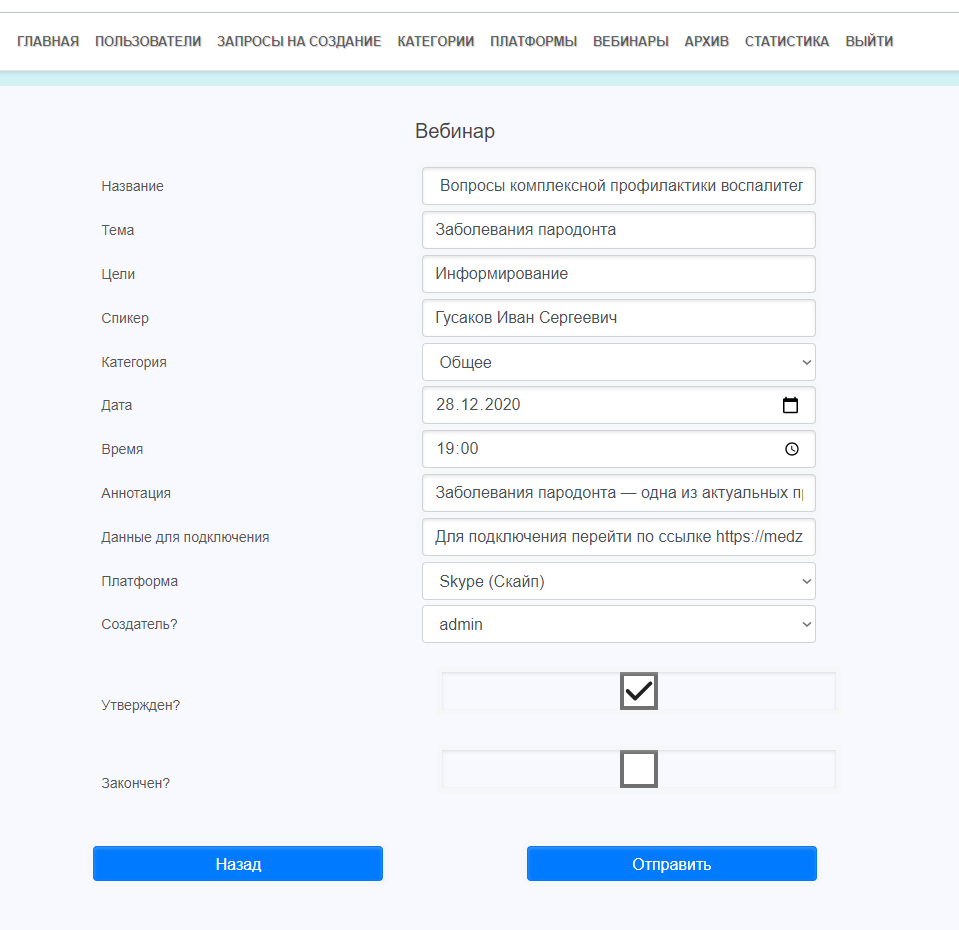


Рисунок 4.16 – Страница создания вебинара

Для выбора платформ используются ранее записанные данные

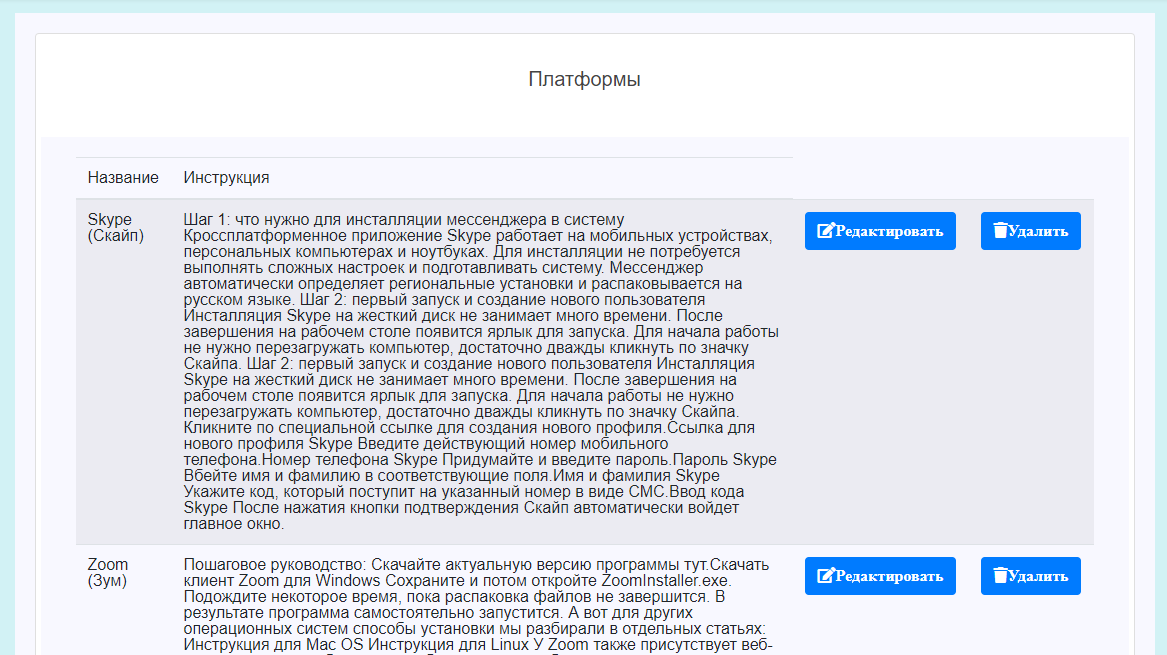


Рисунок 4.17 – Страница платформы

Последнее на страницах администратора, это статистика. Она разделена на 2 части. В первой показано, сколько всего вебинаров определенной категории, сколько из них активных и сколько людей всего посетило данную категорию.

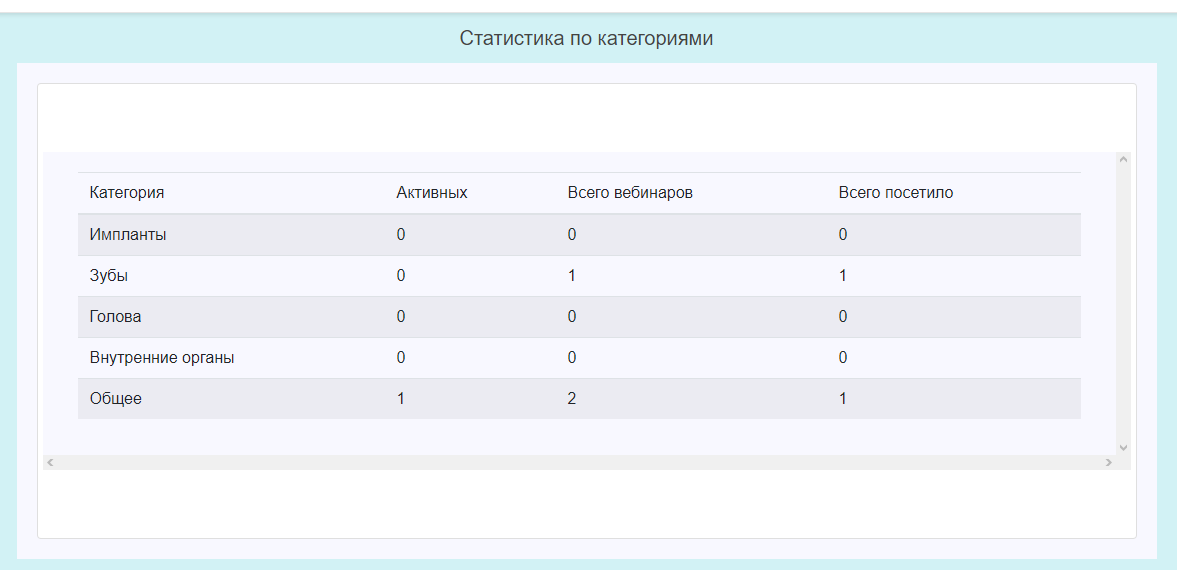


Рисунок 4.18 – Статистика по категориям

Вторая часть показывает список посетителей каждого щаконченного вебинара.

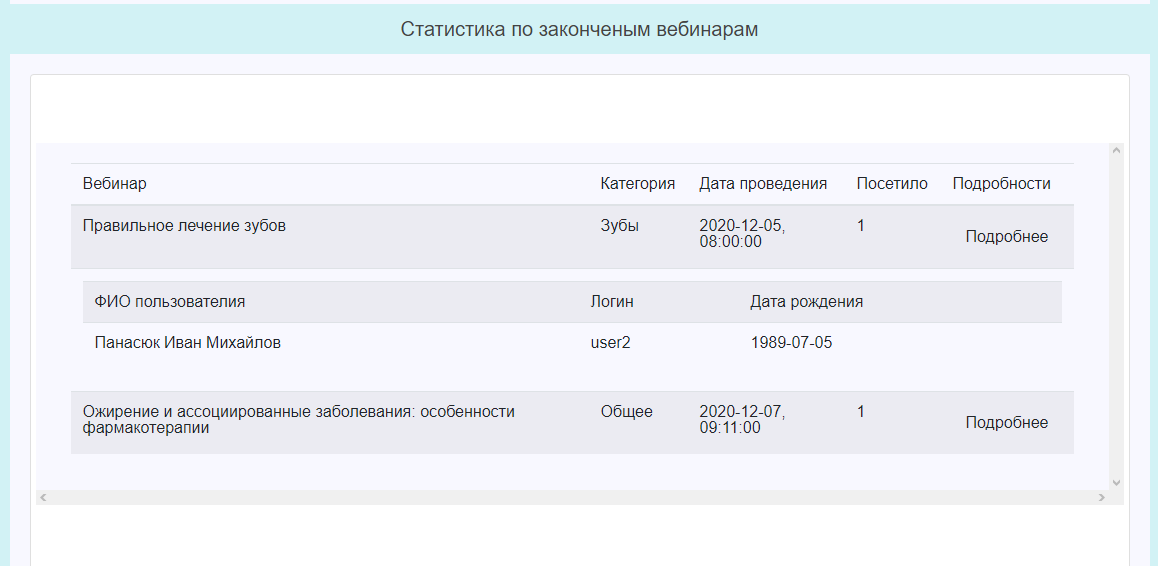


Рисунок 4.19 – Статистика по законченным вебинарам

С администратором закончили, осталось рассмотреть пару страниц в работе пользователя.

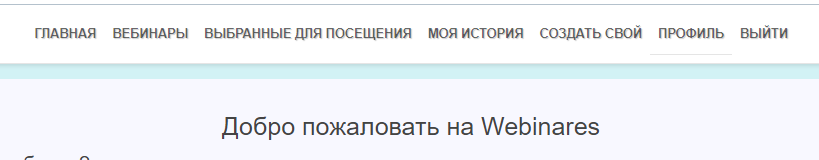


Рисунок 4.20 – Меню при работе как пользователь

На странице вебинаров отражен список тех вебинаров, которые не закончились и попадают под выбранный в профиле категории.

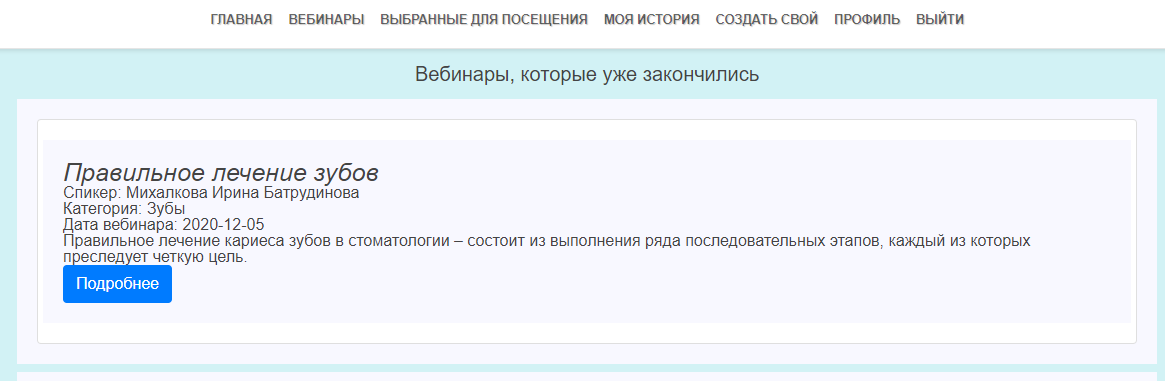
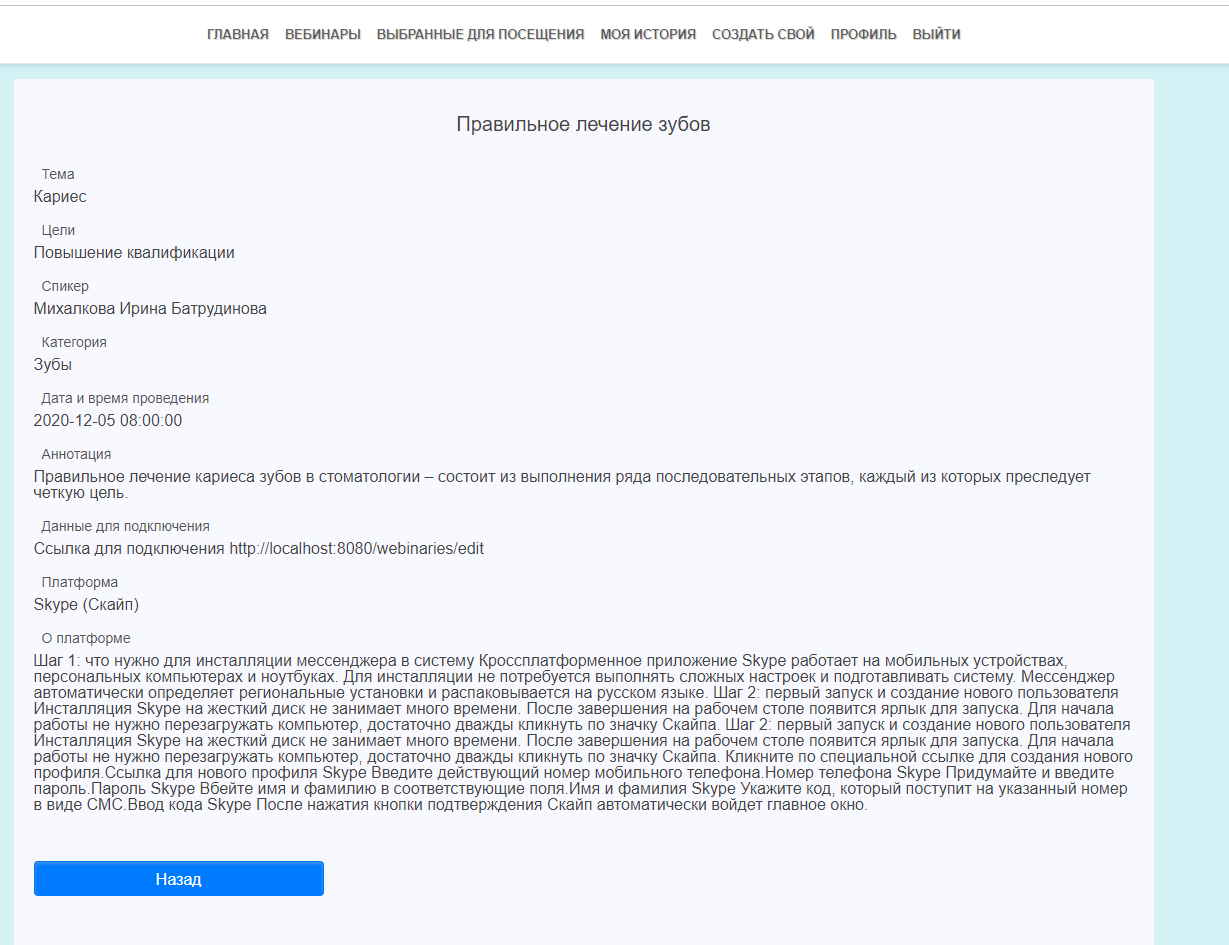


Рисунок 4.21 – Законченный вебинары.

Отличие отображения законченных и незаконченных вебинаров в том, что у законченный нет кнопки «участвовать».

Страница с подробной информацией выглядит как на рисунке 4.22

Рисунок 4.22 – Страница подробностей

И последняя страница пользователя – профиль

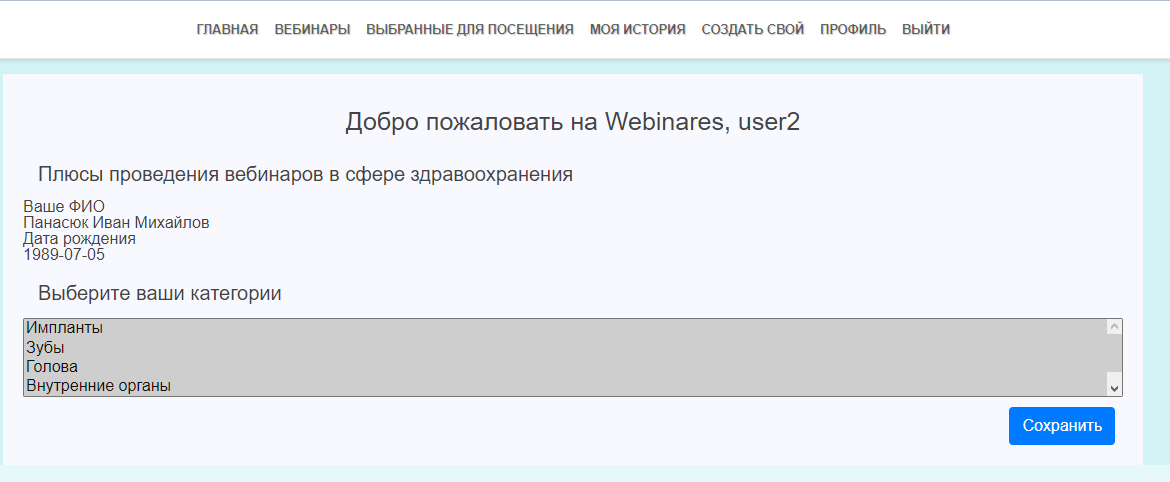


Рисунок 4.23 – Страница профиля

На данной странице можно увидеть информацию о вошедшем пользователей и выбрать категории для отображения вебинаров.

На этом описание четвертой главы завершено.

## 5. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО СРЕДСТВА.

5.1. Характеристика программного средства.

Целью дипломного проекта является разработка программного средства для автоматизации процесса взаимодействия корпоративных сотрудников. Данное программное средство разработано в рамках компании УП «Медтехника» г. Витебска и предназначено для собственного использования.

Это программное средство представляет собой веб-приложение, в котором реализованы следующие функции:

* просмотр списка категорий вебинаров;
* просмотр списка платформ для проведения вебинаров;
* просмотр подробной информации о вебинаре;
* отправка пользователем запроса на создание собственного вебинара;
* отображение архива посещенных вебинаров.

Экономическая оценка целесообразности инвестиций в разработку и использование программного продукта осуществляется на основе расчета и оценки следующих экономических показателей:

- чистой дисконтированной стоимости (ЧДД);

- срока окупаемости инвестиций (ТОК);

- рентабельности инвестиций (Ри).

5.2. Расчет сметы затрат на разработку программного средства.

Основная заработная плата исполнителей разработки программного продукта определяется по формуле (5.1) [45].

|  |  |
| --- | --- |
|  | (5.1) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | *n* | – | количество исполнителей, занятых разработкой ПП; |
|  | *T*Ч*i* | – | часовая тарифная ставка *i*-го исполнителя, руб.; |
|  | ФЭ*i* | – | эффективный фонд рабочего времени *i*-исполнителя, дни; |
|  | *T*Ч | – | количество часов работы в день, 8 часов; |
|  | *K* | – | коэффициент премирования (10%). |

В настоящий момент базовая ставка (на 13.12.2020) в бюджетной организации составляет 185 рублей [46].

Расчет основной заработной платы предоставлен в таблице 5.1.

Таблица 5.1 – Расчет основной заработной платы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Должность | Месячная тарифная ставка, руб. | Дневная тарифная ставка, руб. | Плановый фонд рабочего времени, дни | Премия, руб. | Заработная плата исполнителей, руб. |
| Руководитель проекта | 600 | 28,57 | 42 | 119 | 1318,94 |
| Инженер-программист | 520 | 24,76 | 42 | 103 | 1142,92 |
| Итого с учётом премии **()** |  |  |  |  | 2461,86 |

Дополнительная заработная плата определяется по формуле (5.2)

|  |  |
| --- | --- |
|  | (5.2) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | НД | – | норматив дополнительной заработной платы (10%). |

Дополнительная заработная плата составит:

246,19

Отчисления в фонд социальной защиты населения и на обязательное страхование () определяются в соответствии с действующими законодательными актами по формуле (5.3) [47]

|  |  |
| --- | --- |
|  | (5.3) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | НСЗ | – | норматив отчислений в фонд социальной защиты населения и на обязательное страхование (34% + 0,6%). |

Отчисления в фонд социальной защиты населения и на обязательное страхование составят:

936,98

Затраты машинного времени на разработку программного средства. Расходы по статье «Машинное время» () включает оплату машинного времени, необходимого для разработки и отладки ПП, и определяются по формуле (5.4)

|  |  |
| --- | --- |
|  | (5.4) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | Ц | – | цена одного машино-часа, руб.; |
|  | Т | – | количество часов в день, 8ч.; |
|  | СР | – | длительность проекта, дни. |

Стоимость машино-часа в организации составляет 0,40 руб. Разработка проекта занимает 42 дня для руководителя проекта и 42 дня для инженера-программиста.

Затраты по статье «Машинное время» составят:

176,40

Затраты по статье «Накладные расходы» (), связанные с необходимостью содержания аппарата управления, определяются по формуле (5.5)

|  |  |
| --- | --- |
|  | (5.5) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | НРН | – | норматив накладных расходов (30%). |

Затраты по статье «Накладные расходы» составят:

Общая сумма расхода по всем статьям сметы () на разработку ПС рассчитывается по формуле (5.6)

|  |  |
| --- | --- |
|  | (5.6) |

Общая сумма расхода по всем статьям сметы составят:

Затраты на сопровождение и адаптацию определяются по формуле (5.7)

|  |  |
| --- | --- |
|  | (5.7) |

|  |  |  |  |
| --- | --- | --- | --- |
| где | Н*pca*  CР | –  – | норматив расходов на сопровождение и адаптацию (10%);  смета расходов в целом по организации без расходов на сопровождение и адаптацию, руб. |

Затраты на сопровождение и адаптацию составят:

руб.

Общая сумма расходов на разработку (с затратами на сопровождение и адаптацию) как полная себестоимость ПП (С) определяется по формуле (5.8)

|  |  |
| --- | --- |
|  | (5.8) |

Так как программное средство предназначено для собственных нужд организации, мы ограничиваемся расчетом полной себестоимости.

Полная себестоимость ПП составит:

5.3. Расчет стоимостной оценки результата.

Результатом в использовании программного средства выступает прирост чистой прибыли за счет экономии затрат на заработную плату и амортизационные отчисления.

Экономия затрат на заработную плату. Прирост прибыли за счет экономии расходов на заработную плату и отчислений на социальное страхование в результате снижения трудоемкости определяется по формуле:

(5.9)

Где - плановый объем работ;

, - трудоемкость выполнения работы до (1) и после внедрения программного продукта (0,25), нормо-час;

, - часовая тарифная ставка до и после внедрения ПС (7,14) руб./ч;

- коэффициент премий (1,1);

- норматив дополнительной заработной платы (20%);

- ставка отчислений от заработной платы, включаемых в себестоимость (34,6%).

Экономия затрат на заработную плату составит:

Прирост чистой прибыли определяется по формуле:

(5.10)

Где – ставка налога на прибыль (18%).

Амортизационные отчисления являются источником погашения инвестиций в разработку программного продукта.

Расчет амортизационных отчислений осуществляется по следующей формуле:

(5.11)

Где З – затраты на разработку программного продукта, руб.

– норма амортизации программного продукта (20%).

5.4. Расчет показателей эффективности разработки и использования программного средства.

Для расчета показателей экономической эффективности использования программного продукта необходимо полученные суммы результата (прироста чистой прибыли) и затрат (капитальных вложений) по годам приводят к единому времени – расчетному году путем умножения результатов и затрат за каждый год на коэффициент привидения (), который рассчитывается по формуле [48]:

Где *Ен* – норматив привидения разновременных затрат и результатов (10%) [49];

*tp* – расчетный год, *tp* = 1;

*t* – номер года, результаты и затраты которого приводятся к расчетному.

2021 г. –

2022 г. –

2023 г. –

2024 г. –

Результаты расчета показателей эффективности приведены в таблице 5.2.

Таблица 5.2 − Расчет экономического эффекта от использования нового ПС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование показателей | По годам производства | | | |
| 2021 | 2022 | 2023 | 2024 |
| 1. Прирост прибыли | 1560,35 | 1560,35 | 1560,35 | 1560,35 |
| 2. Амортизационные отчисления | 1094,39 | 1094,39 | 1094,39 | 1094,39 |
| 3. Прирост результата | 2654,74 | 2654,74 | 2654,74 | 2654,74 |
| 4. Коэффициент дисконтирования | 1 | 0,909 | 0,82 | 0,751 |
| 5. Дисконтированный результат | 2654,74 | 2413,15 | 2176,88 | 1993,71 |
| 6. Инвестиции в разработку программного средства | 5471,98 | - | - | - |
| 7. Дисконтированные инвестиции | 5471,98 | - | - | - |
| 8. Чистый дисконтированный доход по годам | -3911,63 | 2413,15 | 2176,88 | 1993,71 |
| 9. ЧДД с нарастающим итогом | -3911,63 | 5847,44 | 8024,32 | 10018,03 |

Рассчитаем рентабельность инвестиций в разработку и внедрение программного продукта (РИ) по формуле:

, (5.13)

где – результат с учетом фактора времени, руб.;

– инвестиции с учетом фактора, руб.;

*n* – количество лет в расчетном периоде, равное 4.

Тогда рентабельность составит:

Ри=((-3911,63+5847,44+8024,32+10018,03)/ 5471,98)\*100% = 365%

В результате технико-экономического обоснования разработки и применения программного средства были получены следующие значения показателей эффективности [50]:

- чистый дисконтированный доход составит 10018,03;

- инвестиции окупаются на второй год использования;

- рентабельность программного продукта составит: 365%

Таким образом, применение программного продукта является эффективным и инвестиции в его разработку целесообразно осуществлять.

## ЗАКЛЮЧЕНИЕ

В данной работе представлена разработка программного средства поддержки процесса организации вебинаров для взаимодействия корпоративных сотрудников, которое позволит отделу повышения квалификации, отделу кадров и руководителям подразделений уменьшить временные и денежные затраты на управление повышения подготовки персонала:

* + имеет интуитивно понятный интерфейс для осуществления навигации по приложению;
  + осуществляет вход в приложение с использованием собственных данных для аутентификации;
  + создает информационные потоки для каждой группы пользователей;
  + имеет возможность редактирования личных данных
  + добавляет новые категории вебинаров
  + добавлять новые платформы проведения вебинаров

Для разработки справочно-поисковой системы был выбран следующий комплекс средств:

* + комплекс инструментальных средств Eclipse IDE для разработки серверной и клиентской частей приложения;
  + фреймворк Spring Framework как средство разработки web-приложений;
  + СУБД Microsoft SQL Server как средство хранение данных.

К преимуществам разработанной системы управления бизнес-процессами следует отнести:

* + адаптированный под нужды сотрудников интерфейс;
  + единую систему авторизации;
  + доступ на основе ролей;
  + гибкую адаптацию под новые требования, возможность добавления новых сервисов и функций;
  + высокую доступность приложения;
  + осуществление обмена информацией без поддержки IT-персонала.

Таким образом, система, обладая перечисленными преимуществами, в полной мере сможет восполнить потребность сотрудников компании в эффективном инструменте для информационного обмена и внутреннего взаимодействия.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Алгоритм направления работника на повышение квалификации [Электронный ресурс]. – Режим доступа: https://www.spok.by/izdaniya/ya-spok/algoritm-napravleniya-rabotnika-na-povys\_0000000. – Дата доступа: 31.12.2020.

2 Развитие навыков с целью повышения эффективности [Электронный ресурс]. – Режим доступа: http://vestnikmckinsey.ru/organizational-models-and-management-systems/Razvitie-navykov-s-tselyu-povysheniya-effectivnosti. – Дата доступа: 31.12.2020.

3 Корпоративноеобучение [Электронный ресурс]. – Режим доступа: cyclowiki.org/wiki/Корпоративное\_обучение. – Дата доступа: 31.12.2020.

4 Вебинар. Что такое вебинар ? [Электронный ресурс]. – Режим доступа: http://www.myshared.ru/slide/225926/. – Дата доступа: 31.12.2020.

5 Вебинары как инновационная форма корпоративного обучения [Электронный ресурс]. – Режим доступа: http://sbricur.com/wp-content/uploads/2014/03/13\_Tretyakova.pdf. – Дата доступа: 31.12.2020.

6 Вебинар как средство повышения квалификации сотрудников [Электронный ресурс]. – Режим доступа: http://www.rlib.yar.ru/\_metod\_mater/v\_7/01/sokolova\_doklad.htm. – Дата доступа: 31.12.2020.

7 Лучший инструмент для проведения бесплатных конференций и эффективного сотрудничества // freeconferencecall.com [Электронный ресурс]. – Режим доступа: https://www.freeconferencecall.com/ru/by. – Дата доступа: 23.11.2020.

8 Zoom: лидер в сфере конференц-решений согласно отчетам Gartner Magic Quadrant, 2019 г // zoom.us [Электронный ресурс]. – Режим доступа: https://zoom.us/. – Дата доступа: 23.11.2020.

9 Free Video Conferencing // uberconference.com [Электронный ресурс]. – Режим доступа: https://www.uberconference.com/. – Дата доступа: 23.11.2020.

10 Skype // skype.com [Электронный ресурс]. – Режим доступа: https://www.skype.com/ru/. – Дата доступа: 23.11.2020.

11 Just join me to train a team // join.me [Электронный ресурс]. – Режим доступа: https://www.join.me/. – Дата доступа: 23.11.2020.

12 Apache OpenMeetings - Features and overview // openmeetings.apache.org [Электронный ресурс]. – Режим доступа: https://openmeetings.apache.org/. – Дата доступа: 23.11.2020.

13 Проводите вебинары — для обучения, презентаций или совещаний онлайн // webinar.ru [Электронный ресурс]. – Режим доступа: https://webinar.ru/. – Дата доступа: 23.11.2020.

14 IDEF0 // ru.wikipedia.org [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/IDEF0. – Дата доступа: 23.11.2020.

15 Анализ требований // ru.wikipedia.org [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Анализ\_требований. – Дата доступа: 23.11.2020.

16 Основы методологии IDEF1X // cfin.ru [Электронный ресурс]. – Режим доступа: https://www.cfin.ru/vernikov/idef/idef1x.shtml. – Дата доступа: 23.11.2020.

17 UML // ru.wikipedia.org [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/UML. – Дата доступа: 23.11.2020.

18 Введение в язык UML [Электронный ресурс]. – Режим доступа: https://www.bsuir.by/m/12\_100229\_1\_65759.pdf. – Дата доступа: 31.12.2020.

19 Диаграммапрецедентов [Электронный ресурс]. – Режим доступа: ru.wikipedia.org/wiki/Диаграмма\_прецедентов. – Дата доступа: 31.12.2020.

20 Элементы графической нотации диаграммы вариантов использования [Электронный ресурс]. – Режим доступа: https://intuit.ru/studies/courses/32/32/lecture/1004?page=2. – Дата доступа: 31.12.2020.

21 Spring — Обзор MVC Framework [Электронный ресурс]. – Режим доступа: https://coderlessons.com/tutorials/java-tekhnologii/uchis-spring-mvc/spring-obzor-mvc-framework. – Дата доступа: 31.12.2020.

22 Spring — MVC Framework [Электронный ресурс]. – Режим доступа: https://coderlessons.com/tutorials/java-tekhnologii/uchis-vesne/spring-mvc-framework. – Дата доступа: 31.12.2020.

23 Диаграмма последовательности [Электронный ресурс]. – Режим доступа: ru.wikipedia.org/wiki/Диаграмма\_последовательности. – Дата доступа: 31.12.2020.

24 Разновидности UML диаграмм [Электронный ресурс]. – Режим доступа: https://forum.itvdn.com/t/urok-2-raznovidnosti-uml-diagramm/3315. – Дата доступа: 31.12.2020.

25 Диаграмма состояний [Электронный ресурс]. – Режим доступа: ru.wikipedia.org/wiki/Диаграмма\_состояний\_(UML). – Дата доступа: 31.12.2020.

26 Диаграмма классов [Электронный ресурс]. – Режим доступа: ru.wikipedia.org/wiki/Диаграмма\_классов. – Дата доступа: 31.12.2020.

27 Диаграммы классов [Электронный ресурс]. – Режим доступа: https://logrocon.ru/news/plantuml\_diagrams. – Дата доступа: 31.12.2020.

28 Интерфейс пользователя [Электронный ресурс]. – Режим доступа: ru.bmstu.wiki/Интерфейс\_пользователя. – Дата доступа: 31.12.2020.

29 ГОСТ «Эргономика взаимодействия человек-система» [Электронный ресурс]. – Режим доступа: http://docs.cntd.ru/document/1200097755. – Дата доступа: 31.12.2020.

30 Проектирование интерфейса пользователя [Электронный ресурс]. – Режим доступа: https://elib.belstu.by/bitstream/123456789/32078/1/Brusencova\_proektirovanie\_2019.pdf. – Дата доступа: 31.12.2020.

31 Что такое UX дизайн? [Электронный ресурс]. – Режим доступа: https://tilda.education/articles-what-is-ux-design. – Дата доступа: 31.12.2020.

32 Spring Framework [Электронный ресурс]. – Режим доступа: ru.wikipedia.org/wiki/Spring\_Framework. – Дата доступа: 31.12.2020.

33 Spring tutorial [Электронный ресурс]. – Режим доступа: https://javastudy.ru/frameworks/spring/. – Дата доступа: 31.12.2020.

34 История языка программирования Java [Электронный ресурс]. – Режим доступа: https://www.interestprograms.ru/article-istoriya-sozdaniya-yazyka-programmirovaniya-java. – Дата доступа: 31.12.2020.

35 Что такое HTML на самом деле? [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics. – Дата доступа: 31.12.2020.

36 Каскадные таблицы стилей [Электронный ресурс]. – Режим доступа: https://iit-web-lectures.readthedocs.io/ru/latest/www/css.htm. – Дата доступа: 31.12.2020.

37 Что такое JavaScript на самом деле? [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/JavaScript\_basics. – Дата доступа: 31.12.2020.

38 UML [Электронный ресурс]. – Режим доступа: https://myalm.ru/glossary/UML. – Дата доступа: 31.12.2020.

39 Uml диаграмма компонентов описание. UML-диаграмма. Виды диаграмм UML. Отношения в UML [Электронный ресурс]. – Режим доступа: https://montazhtv.ru/uml-diagramma-komponentov-opisanie-uml-diagramma-vidy-diagramm-uml-otnosheniya-v/. – Дата доступа: 31.12.2020.

40 Набор тест-кейсом [Электронный ресурс]. – Режим доступа: https://www.qastart.by/mainterms/34-testcase-testsuite. – Дата доступа: 31.12.2020.

41 Установка Tomcat [Электронный ресурс]. – Режим доступа: https://metanit.com/java/javaee/2.1.php. – Дата доступа: 31.12.2020.

42 Установка MySQL 8 на Windows 10 [Электронный ресурс]. – Режим доступа: https://info-comp.ru/install-mysql-on-windows-10. – Дата доступа: 31.12.2020.

43 Maven. Часть 1 – Знакомство и настройка [Электронный ресурс]. – Режим доступа: devcolibri.com/maven-часть-1-знакомство-и-настройка/. – Дата доступа: 31.12.2020.

44 Run Spring Boot App from a Command Line [Электронный ресурс]. – Режим доступа: https://www.appsdeveloperblog.com/run-spring-boot-app-from-a-command-line/. – Дата доступа: 31.12.2020.

45 Технико-экономическое обоснование дипломных проектов для студентов [Электронный ресурс]. – Режим доступа: https://www.bstu.by/uploads/attachments/metodichki/kafedri/UEiF\_Elektr.pdf. – Дата доступа: 31.12.2020.

46 Повышение базовой ставки с 1 января Подробнее на сайте Экономической газеты: https://neg.by/novosti/otkrytj/povyshenie-bazovoj-stavki-s-1-yanvarya [Электронный ресурс]. – Режим доступа: https://neg.by/novosti/otkrytj/povyshenie-bazovoj-stavki-s-1-yanvarya. – Дата доступа: 31.12.2020.

47 Отчисления в фонд социальной защиты населения [Электронный ресурс]. – Режим доступа: https://www.bsuir.by/m/12\_100229\_1\_109181.doc. – Дата доступа: 31.12.2020.

48 Расчет стоимостной оценки результата [Электронный ресурс]. – Режим доступа: https://studbooks.net/2034450/informatika/raschet\_stoimostnoy\_otsenki\_rezultata. – Дата доступа: 31.12.2020.

49 Норматив для приведения разновременных затрат [Электронный ресурс]. – Режим доступа: https://economy-ru.info/info/2026/. – Дата доступа: 31.12.2020.

50 Оценка финансово-экономической эффективности внедрения современных компьютерных информационных технологий в организациях [Электронный ресурс]. – Режим доступа: https://libeldoc.bsuir.by/bitstream/123456789/38249/1/Nasonova\_Ocenka2.pdf. – Дата доступа: 31.12.2020.

## ПРИЛОЖЕНИЕ А

**(обязательное)**

**Результат проверки пояснительной записки в системе «Антиплагиат»**

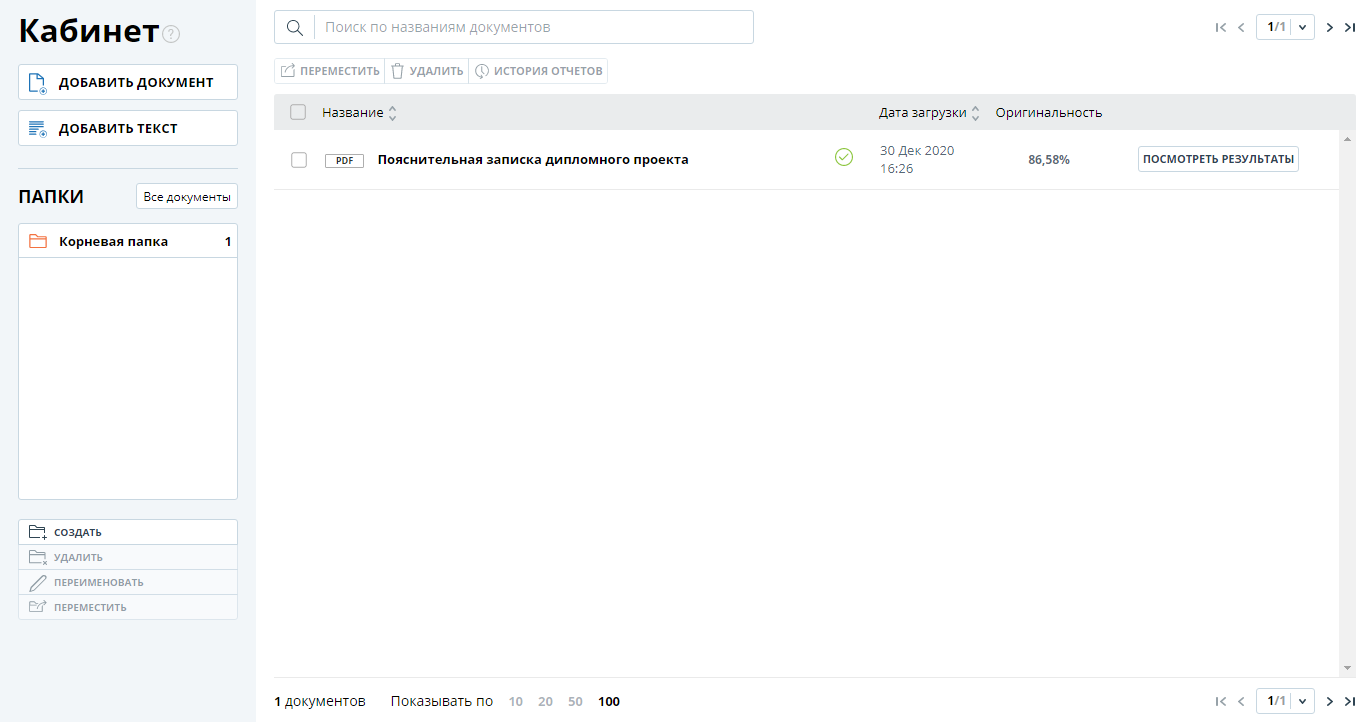
****

Рисунок А.1 – Результат проверки в системе «Антиплагиат»

## ПРИЛОЖЕНИЕ Б

**(обязательное)**

**Диаграмма классов**

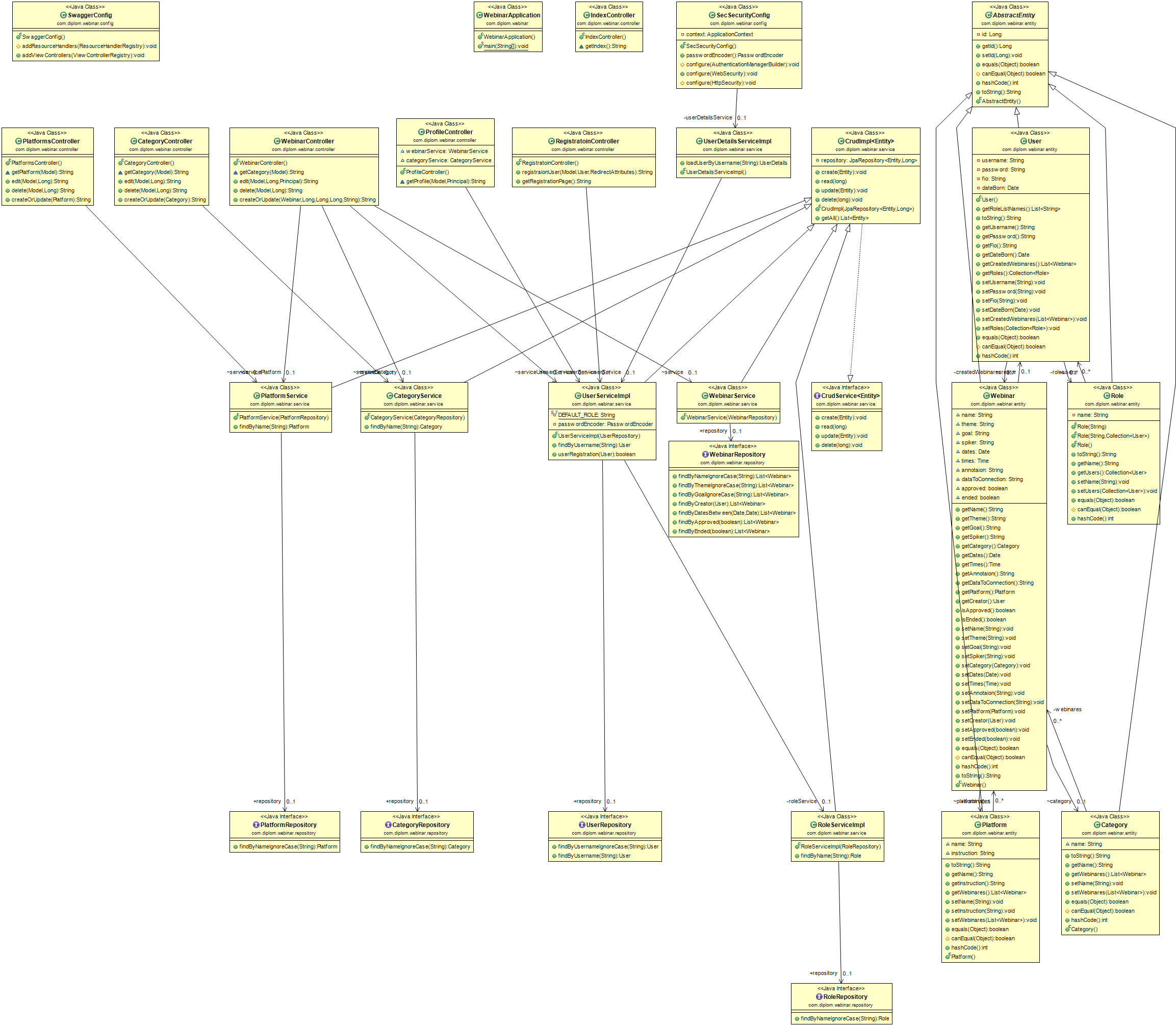


Рисунок Б.1 – Диаграмма классов

## ПРИЛОЖЕНИЕ В

**(обязательное)**

**Листинг алгоритмов, реализующих бизнес-логику**

package com.diplom.webinar.controller;

import java.security.Principal;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import com.diplom.webinar.entity.Category;

import com.diplom.webinar.entity.Feedback;

import com.diplom.webinar.entity.User;

import com.diplom.webinar.entity.Webinar;

import com.diplom.webinar.exception.RecordNotFoundException;

import com.diplom.webinar.service.CategoryService;

import com.diplom.webinar.service.FeedbackService;

import com.diplom.webinar.service.PlatformService;

import com.diplom.webinar.service.UserServiceImpl;

import com.diplom.webinar.service.WebinarService;

@Controller

@RequestMapping("/webinariesToUser")

public class WebinarToUserController {

@Autowired

WebinarService service;

@Autowired

UserServiceImpl serviceUser;

@Autowired

CategoryService serviceCategory;

@Autowired

PlatformService servicePlatform;

@Autowired

FeedbackService serviceFeedback;

// список вебинаров кроме тех, на которые уже иду

@GetMapping

String getWebinaries(Model model, Principal principal) {

User user = serviceUser.findByUsername(principal.getName());

List<Webinar> list = service.repository.findAll();

if (list == null)

model.addAttribute("list", null);

else {

// цикл вебинаров с бд

for (int i = list.size() - 1; i >= 0; i--) {

if(list.size()<=i)continue;

// удаление не подтвержденных и тех, что уже прошли

if (!list.get(i).isApproved() || list.get(i).isEnded()) {

list.remove(i);

continue;

}

// if(list.size()<=i)continue;

// цикл вебинаров для посещения пользователей, удаляем те, на которые уже идем

for (int j = 0; j < user.getWebinariesToGo().size(); j++) {

if(list.size()<=i)continue;

if (user.getWebinariesToGo().get(j).getId() == list.get(i).getId()) {

list.remove(i);

continue;

}

}

// проверяем, совпадает ли категория вебинаров с теми, что уже есть

boolean isWebinarInRightCategory = false;

for (Category cat : user.getCategoriesToShow()) {

if(list.size()<=i)continue;

if (cat.getId() == list.get(i).getId())

isWebinarInRightCategory = true;

}

if(list.size()<=i)continue;

if (!isWebinarInRightCategory)

list.remove(i);

}

model.addAttribute("isContains", false);

model.addAttribute("list", list);

}

return "webinariesToUser";

}

// когда согласился пойти на выбранный

@GetMapping

@RequestMapping(path = { "/igo/{id}" })

public String addWebinarInList(Model model, @PathVariable("id") Long id, Principal principal) throws Exception {

User user = serviceUser.findByUsername(principal.getName());

user.getWebinariesToGo().add(service.read(id));

serviceUser.update(user);

return "redirect:/webinariesToUser/toGo";

}

//список всех выбранных для похода

@GetMapping

@RequestMapping(path = { "/toGo" })

public String getMyWebinatiesToGo(Model model, Principal principal) throws RecordNotFoundException {

System.out.println("Webinar toGo");

User user = serviceUser.findByUsername(principal.getName());

List<Webinar> list = user.getWebinariesToGo();

for (int i = list.size() - 1; i >= 0; i--) {

if (!list.get(i).isApproved() || list.get(i).isEnded())

list.remove(i);

}

model.addAttribute("isContains", true);

model.addAttribute("list", list);

return "webinariesToUser";

}

// список всех выбранных для похода

@GetMapping

@RequestMapping(path = { "/history" })

public String getMyHistory(Model model, Principal principal) throws RecordNotFoundException {

User user = serviceUser.findByUsername(principal.getName());

List<Webinar> list = user.getWebinariesToGo();

for (int i = list.size() - 1; i >= 0; i--) {

if (!list.get(i).isEnded())

list.remove(i);

}

model.addAttribute("isContains", true);

model.addAttribute("isEnded", true);

model.addAttribute("list", list);

return "webinariesToUser";

}

// удаление выбранного из своего списка для похода

@GetMapping

@RequestMapping(path = "/delete/{id}")

public String delete(Model model, @PathVariable("id") Long id, Principal principal) throws Exception {

User user = serviceUser.findByUsername(principal.getName());

for (int i = 0; i < user.getWebinariesToGo().size(); i++)

if (user.getWebinariesToGo().get(i).getId() == id)

user.getWebinariesToGo().remove(i);

serviceUser.update(user);

return "redirect:/webinariesToUser/toGo";

}

// получение информации о выбранном

@GetMapping

@RequestMapping(path = "/info/{id}")

public String getInfo(Model model, @PathVariable("id") Long id, Principal principal) throws Exception {

User user = serviceUser.findByUsername(principal.getName());

Webinar read = service.read(id);

boolean isContains = false;

if (user == null)

isContains = true;

else

for (int i = 0; i < user.getWebinariesToGo().size(); i++)

if (user.getWebinariesToGo().get(i).getId() == id)

isContains = true;

model.addAttribute("isContains", isContains);

if (read.isEnded())

model.addAttribute("isEnded", true);

model.addAttribute("entity", read);

if(user!=null)

model.addAttribute("username", user.getFio());

else

model.addAttribute("username", "admin");

return "webinariesInfo";

}

// оставить отзыв

@PostMapping

@RequestMapping(path = "/addFeedback/{id}")

public String addFeedback(Model model, @PathVariable("id") Long id,

@RequestParam(value = "text") String text,

Principal principal) throws Exception {

Webinar read = service.read(id);

Feedback fb = new Feedback();

fb.setAuthor(principal.getName());

fb.setText(text);

serviceFeedback.create(fb);

Feedback feedback = serviceFeedback.repository.findByAuthorAndText(fb.getAuthor(), text);

read.getFeedbacks().add(feedback);

service.update(read);

feedback.setWebinar(read);

serviceFeedback.update(feedback);

return "redirect:/webinariesToUser/info/"+id;

}

}

package com.diplom.webinar.controller;

import java.security.Principal;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import com.diplom.webinar.entity.Category;

import com.diplom.webinar.entity.User;

import com.diplom.webinar.exception.RecordNotFoundException;

import com.diplom.webinar.service.CategoryService;

import com.diplom.webinar.service.UserServiceImpl;

import com.diplom.webinar.service.WebinarService;

@Controller

@RequestMapping("/profile")

public class ProfileController {

@Autowired

UserServiceImpl userService;

@Autowired

WebinarService webinarService;

@Autowired

CategoryService categoryService;

@GetMapping

String getProfile(Model model,Principal principal) {

User user = userService.findByUsername(principal.getName());

model.addAttribute("entity", user);

model.addAttribute("list", categoryService.repository.findAll());

return "profile";

}

@RequestMapping("/edit")

public String edit(Model model,

@RequestParam(value = "categ", required = true) int[] categ,

Principal principal) throws Exception {

User user = userService.findByUsername(principal.getName());

if(categ!=null) {

user.getCategoriesToShow().clear();

for(int i=0; i<categ.length;i++)

{

user.getCategoriesToShow().add(categoryService.read(categ[i]));

}

}

userService.update(user);

model.addAttribute("success", "Выбранные категории сохранены");

return "redirect:/profile";

}

}package com.diplom.webinar.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import com.diplom.webinar.entity.Category;

import com.diplom.webinar.exception.RecordNotFoundException;

import com.diplom.webinar.service.CategoryService;

@Controller

@RequestMapping("/categories")

public class CategoryController {

@Autowired

CategoryService service;

@GetMapping

String getCategory(Model model) {

List<Category> list = service.repository.findAll();

if (list == null)

model.addAttribute("list", null);

else

model.addAttribute("list", list);

return "categories-list";

}

@RequestMapping(path = { "/edit", "/edit/{id}" })

public String edit(Model model, @PathVariable(name="id",required = false) Long id) throws RecordNotFoundException {

if (id!=null) {

Category entity = service.read(id);

model.addAttribute("entity", entity);

} else {

model.addAttribute("entity", new Category());

}

return "categories-add-edit";

}

@RequestMapping(path = "/delete/{id}")

public String delete(Model model, @PathVariable("id") Long id) throws Exception {

service.delete(id);

return "redirect:/categories";

}

@RequestMapping(path = "/create", method = RequestMethod.POST)

public String createOrUpdate(Category entity) throws Exception {

service.create(entity);

return "redirect:/categories";

}

}