

Разработка информационной системы автоматизации приема заявок на техническое обслуживание

Цой Андрей Дэгиевич

Сахалинский государственный университет

Студент

Козлов Евгений Николаевич

Сахалинский государственный университет

старший преподаватель кафедры информатики

Аннотация

В данной статье говорится о разработке информационной системы автоматизации приема заявок на техническое обслуживание. Статья содержит в себе краткое описание этапов проектирования и разработки информационной системы, сопровождаемое схемой базы данных и примерами форм программы.

Ключевые слова: Информационная система, база данных, техническое обслуживание, автоматизация.

Development of information system of automation accepting applications for maintenance

Coi Andrei Degievich

Sakhalin State University

Student

Kozlov Evgenii Nikolaevich

Sakhalin State University

IT Department Lecturer

Abstract

This article describes the development of the information systems of automation accepting applications for maintenance. This article contains a brief description of the stages of design and development of information systems, followed by database schema and examples of program forms.

Keywords: Information system, database, maintenance, automation.

Задача автоматизации тех или иных процессов возникает в современном бизнесе довольно часто. В настоящее время без использования компьютерной техники не мыслим практически ни один бизнес, информационные технологии проникли во все сферы нашей жизни. Естественно, различные фирмы, компании стараются найти им применение в

своём бизнесе независимо от направления деятельности, которое может является довольно обширным.

На сегодняшний день для большинства крупных организаций, существуют проблемы связанные с отсутствием средств автоматизации учета и исполнения технических заявок, таких как устранение неисправностей, установка нового ПО оборудования. Основные из них:

- Неудобность работы с бумажными экземплярами;
- Сложность хранения бумажных экземпляров;
- Сложность в поиске необходимой информации о заявках и отчетах;
- Невозможность ведения анализа выполнения заявок;
- Невозможность оценить работу сотрудников технического отдела;
- Загруженность телефонной связи в пределах технического отдела;
- Нецелесообразная трата рабочего времени.

Эти проблемы актуальны для всех крупных организаций, имеющие в своем штате большое количество компьютерной техники. Решением этих проблем является установка в организацию информационной системы, которая выполняет автоматизацию учета, подачи, сбора и обработки заявок на техническую поддержку и установку программного обеспечения. Данная информационная система позволяет значительно упростить процесс учета заявок на обслуживание, тем самым позволит рационально расходовать время работы специалиста технического отдела.

На российском рынке существует множество программных средств типа ServiceDesk, решающих задачу автоматизации подачи заявок на техническое обслуживание и их обработку. Но из-за многообразия средств реализации и способов решения данной задачи были определены критерии и в соответствии с ними проведен анализ систем с целью определения подходящих программных средств. Выявление их преимуществ и недостатков перед реализуемой системой.

При анализе программных средств, предложенных на российском рынке ПО, были отобраны те, которые удовлетворяли двум основным требованиям:

- Автоматизация процесса управления заявками;
- Клиент-серверная архитектура.

Этим требованиям соответствуют следующие системы:

- Система обработки заявок HelpDesk;
- 1С: Предприятие 8 управление ремонтами и обслуживанием оборудования;
- SmurtNut;
- IntraService;

Для сравнения вышеупомянутых программных средств были определены следующие требования к каждому из них:

- Возможность подачи заявки как на техническую поддержку, установку ПО;

- Динамическое изменение метаданных и их количество у объектов системы;
 - Составление отчетов по заявкам;
 - Оповещение по электронной почте пользователей системы при получении новой заявки;
 - Возможность редактирования заявки и ее повторной отправки;
 - Отображение статуса заявки;
 - Несколько ролей пользователя;
 - Платность программного средства;
- По этим требованиям была построена таблица 1.

Таблица 1 Сравнение систем типа ServiceDesk

	Несколько типов заявок	Отчеты по заявкам	Статус заявки	Оповещение по e-mail	Несколько ролей пользователей	Расширенный поиск	Редактирование заявок	Платность
HelpDesk	+	-	+	-	+	+	-	+
1С:Предприятие 8	+	+	+	+	+	-	+	+
SmartNut	-	+	+	+	+	-	-	+
IntraService	-	+	+	+	+	+	-	+

Исходя из результатов сравнения было принято решение разработать свою автоматизированную ИС для работы с заявками, распространяющейся на бесплатной основе, простой в использовании, требующая минимальное количество затрачиваемых ресурсов и отвечающая всем требованиям.

Для успешной разработки информационной системы была хорошо изучена предметная область проекта после чего была разработана информационная модель проекта и выявлено на основе каких входных документов и какой нормативно-справочной информации происходит выполнение функций по обработке данных и формирование конкретных выходных документов. А также весь цикл обработки информации был разбит на два этапа:

1. Прием, обработка и ввод первичной входящей информации (паспортные данные, реквизиты организаций и т.д.).
2. Формирование отчетности (списков клиентов, списков заявок).

После построения информационной модели можно приступить к построению логической модели базы данных, которая дает возможность увидеть для чего нужна каждая из сущностей базы данных. Основываясь на логической модели базы данных проектируется готовая физическая модель

(проектирование базы данных). Физическая модель данных создана при помощи веб приложения phpMyAdmin. Данная модель данных продемонстрирована на рисунке 1.

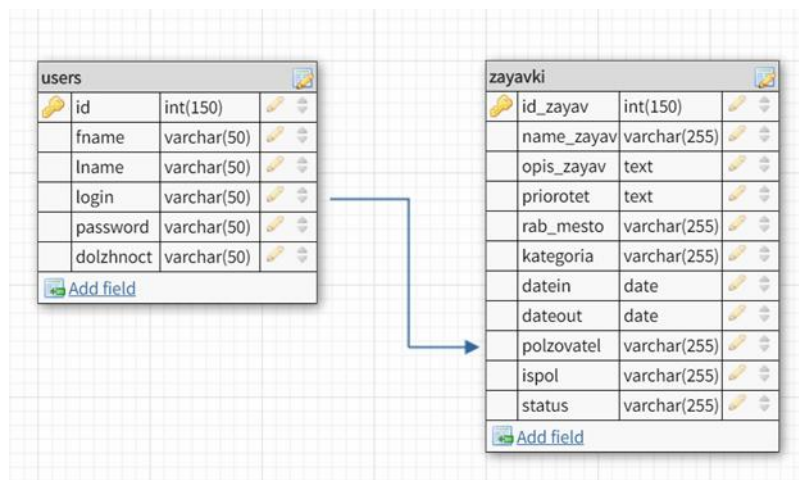


Рисунок 1. Физическая модель базы данных

В БД содержится 2 взаимосвязанные таблицы, приведенных к 3 нормальной форме, то есть: в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов, при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого потенциального ключа и отсутствуют транзитивные зависимости не ключевых атрибутов от ключевых.

После анализа различных СУБД было решено остановиться на выборе одной из двух MySQL и PostgreSQL. СУБД MySQL и PostgreSQL довольно схожи, но MySQL превосходит PostgreSQL по скорости работы, это также реализуется благодаря использованию кэша в запросах. Поскольку зачастую запросы на чтение повторяются, например, к страницам веб-сайта, и кэш играет свою роль. Также у MySQL больше пользователей, чем у PostgreSQL, например, MySQL очень часто выбирают для производственных предприятий. От большого числа пользователей увеличивается и ответственность у разработчиков и тестировщиков, ведь MySQL предоставляет пользователям поддержку на коммерческой основе. А у PostgreSQL поддержка появилась только в последнее время и не настолько развита. Также MySQL имеет превосходство над PostgreSQL в Windows-окружении, MySQL работает как нативное Windows-приложение, а PostgreSQL запускается из эмуляции Unix-среды – окружения Cygwin.

Учитывая требования к системе, в качестве СУБД была выбрана среда MySQL – это многопоточная, многопользовательская СУБД, основными достоинствами которой является быстрота, надежность и простота использования. Несмотря на то, что MySQL не представляет такой широкий набор возможностей, как например Oracle, использование MySQL оправдывается из-за значительных меньших требований к мощности

оборудования и большей скоростью работы при работе со средним объемом данных БД.

Система состоит из двух модулей – базы данных MySQL и приложения для взаимодействия с базой данных, реализованного на языке программирования PHP и с использованием HTML.

Работа с системой осуществляется через браузер. Для работы необходимо установить локальный сервер в локальной сети предприятия, где будет также расположена база данных. Доступ к базе осуществляется с помощью набора адреса в адресной строке браузера.

При запуске программы открывается окно авторизации, где предлагается ввести логин и пароль. Для более удобной работы с системой реализована функция, позволяющая оставаться авторизованным в ней сроком на 31 день. Данный механизм реализован с помощью механизма Cookies.

Разработанная ИС является многопользовательской. Данная система имеет четыре типа пользователей:

- Администратор;
- Диспетчер;
- Исполнитель;
- Клиент

Администратор имеет право просмотреть все заявки и всю информацию о них («Заявки»), данная функция реализована для ведения статистики и оценки загруженности специалистов ИТ-отдела. Здесь же располагаются ссылки на расширенный поиск (Рисунок 2), поиск по номеру заявки, редактирование и удаление заявок.

Вы вошли как АДМИНИСТРАТОР
Здравствуйте, admin!

Подать заявку Заявки Рабочие места Регистрация

Расширенный поиск:

Поиск по категориям:

Приоритет: Рабочее место: Категория: Статус:

Поиск по временному промежутку:

От по

Поиск по заявителю:

Поиск по исполнителю:

Контактная информация технического отдела:
Тел: 728-936-354
Эл. почта: Example@mail.com

Рисунок 2 – Дизайн страницы «Расширенный поиск заявок»

Для управления уже существующими пользователями реализован архив пользователей («Рабочие места»). В нем присутствует вся необходимая информация о зарегистрированных пользователях, а так же функция удаления пользователей.

Регистрация пользователя представляет собой функцию, которая доступна только администратору. В целях того, чтобы предотвратить создание нескольких учетных записей для одного пользователя по причине забытого пароля.

Функция оформления новой заявки реализована в первую очередь для тестирования системы администратором, выявления ее неполадок. И при случаях, когда клиент по техническим причинам сам не может создать заявку.

Для диспетчера доступны следующие функции:

- Просмотр информации о заявках, имеющих статус «невыполненная» (Рисунок 3);
- Расширенный поиск заявок;
- Назначение исполнителя определенной заявке.

The screenshot shows a web interface for a dispatcher. At the top, a brown header contains the text "Вы вошли как ДИСПЕЧЕР" and "Здравствуйте, Ким!". Below the header is a white bar with "Архив заявок". The main content area has a yellow background and is titled "Информация о заявке:". It displays the following details: "Номер заявки: 2", "Тема заявки: Проблемы с монитором", and "Описание заявки: Погас монитор, перезагрузка не помогает." Below this is a form with several fields: "Приоритет: Средний", "Рабочее место: Кабинет 214", "Категория: Аппаратная ошибка", and "Статус: Не выполнена". There are also sections for "Заявитель: Стой" and "Назначить исполнителя заявки:" with a dropdown menu set to "Без исполнителя" and an "Изменить" button. At the bottom, there is contact information for the technical department: "Контактная информация технического отдела", "Тел: 728-916-384", and "Эл. почта: Ekimangle@mail.com".

Рисунок 3 – Дизайн страницы «Информация о заявке»

Исполнителю доступен только просмотр заявок отнесенных к его компетенции, а также изменение статуса заявок в системе.

На рисунке 4 изображен список заявок имеющих статус либо «не выполнена», либо «в процессе», а так же присутствует ссылка на форму просмотра полной информации о заявке.

ВЫ ВОШЛИ КАК РАБОТНИК
Здравствуйте, Титов!

Архив заявок				Незавершенные заявки					
Номер заявки	Тема заявки	Описание заявки	Приоритет	Рабочее место	Категория	Заявитель	Исполнитель	Статус	
3	Тема 2	Описание 6	Низкий	Кабинет 216	Аппаратная ошибка	Choi_A_D	Titov	Не выполнена	
7	Тема 6	Описание заявки 6	Низкий	Кабинет 215	Программная ошибка	Choi_A_D	Titov	Не выполнена	
8	Тема 7	Описание заявки 7	Низкий	Кабинет 214	Проблема с докумен	Petr	Titov	Не выполнена	

Контактная информация технического отдела:
 Тел: 728-936-384.
 Эл. почта: Example@mail.com

Рисунок 4 - Дизайн страницы «Незавершенные заявки»

В возможности клиента входит возможность оставить заявку, а также полное управление своими заявками от их редактирования до назначения исполнителя, которому данная заявка будет приписана.

Разработанная система имеет удобный пользовательский интерфейс, позволяющий легко освоить работу с ней. Доступность используемых технологий, а также гибкость программного кода позволяют расширять функциональность системы по мере необходимости. Система может быть легко адаптирована под нужды любой организации, имеющей сервер под управлением любой из известных Windows- или Unix-подобных операционных систем.

Библиографический список

1. Гоше Х. HTML5. Для профессионалов. СПб.: Питер 2013. 496 с.
2. Тименко Д.А. Кучер Л.В. Разработка web-сайта «Электронная торговая площадка» для автокомпании «Магнит» // Постулат. 2016. №7. С.5.
3. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. СПб.: Питер, 2014. 544 с
4. Фрэйз Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. СПб.: Питер, 2013. 304 с.