

Информационные системы и технологии

УДК 004.415.25

DOI:10.25729/ESI.2024.34.2.014

Система учета заявок для ИТ-подразделений КНЦ СО РАН

Малимонов Максим Игоревич

Федеральный исследовательский центр «Красноярский научный центр СО РАН»,
Россия, Красноярск, malimonov.mi@ksc.krasn.ru

Аннотация. Статья описывает процесс создания системы заявок для КНЦ СО РАН с применением следующих технологий и инструментов: vue, express, node, sequelize, redis и primevue. В статье подробно рассматривается архитектура приложения «Система учета заявок для ИТ-подразделений КНЦ СО РАН», включая клиентскую и серверную части, систему кэширования данных и пользовательский интерфейс. Создание такой системы обеспечивает высокую производительность и удобство применения для пользователей.

Ключевые слова: заявки, система заявок, servicedesk, helpdesk, vue

Цитирование: Малимонов М.И. Система учета заявок для ИТ-подразделений КНЦ СО РАН / М.И. Малимонов // Информационные и математические технологии в науке и управлении. – 2024. – № 2(34). – С. 144-154. – DOI:10.25729/ESI.2024.34.2.014.

Введение. Существует множество различных систем для взаимодействия поставщика ИТ-сервисов и клиентов [1-7]. Существуют как небольшие системы для учета и обработки заявок, называемые «HelpDesk», так и огромные системы, которые позволяют не только обрабатывать заявки ИТ-сервисов, но обрабатывать заявки других отделов, вести учет оборудования, сотрудников и бизнес-процессов компании. Такие большие системы называются «ServiceDesk». Системы для учета заявок используются не только в ИТ-сегменте, но и в различных других областях, например, в телекоммуникации [8-9]. В статье все системы подобного рода будут именоваться как «система заявок». Для организации понятной работы ИТ-отдела используются документы, называемые SLA и OLA.

SLA (Service Level Agreement) – это документ, описывающий условия и требования качества обслуживания, которые должны быть выполнены поставщиком услуг для клиента. SLA определяет уровень сервиса, который должен быть предоставлен клиенту, а также сроки и способы реагирования на запросы клиента [10]. OLA (Operational Level Agreement) – это документ, который определяет условия и требования между различными подразделениями или командами внутри организации для обеспечения достижения целей SLA. OLA определяет взаимодействие между службами поддержки, чтобы обеспечить выполнение SLA [11].

Использование этих документов необходимо для прозрачности работы ИТ-структур, чтобы пользователи понимали, кто отвечает за решение их проблем, куда и к кому необходимо обращаться, в какие сроки их проблема будет решена и каковы способы получения обратной связи. Системы заявок в ИТ-области строятся на принципах SLA и OLA, но учесть всю структуру компании – весьма сложная работа, которая требует огромных временных затрат. Существующие системы не позволяют точно передать структуру организации и их внутреннюю работу, поэтому большие компании занимаются разработками своих собственных систем заявок, в зависимости от ресурсов и потребностей, в которые можно заложить необходимую функциональность и структуру организации.

Здесь очень показательным примером является функционирование подобной системы в Федеральном исследовательском центре Красноярского научного центра Сибирского отделения Российской академии наук (далее ФИЦ КНЦ СО РАН). ФИЦ КНЦ СО РАН включает 18 обособленных подразделений, часть из которых имеют свои собственные ИТ-отделы. Многие

ИТ-отделы не имеют доступа в системы другого ИТ-отдела и вынуждены, как и обычные пользователи, для решения проблем, связанных, например, с работами в 1С, использовать систему заявок. Данная система заявок должна быть построена таким образом, чтобы при создании заявки, например, «подключить рабочее место сотрудника», эта заявка попадала в нужный отдел и нужному исполнителю, в зависимости от места работы сотрудника, подающего заявку. При этом сам пользователь не должен задумываться над тем, как и куда заявка попадет, система должна сама назначить заявку, пользователю нужно лишь выбрать категорию и заполнить необходимые поля. При этом необходимо учесть то, что некоторые пользователи не должны иметь доступ к некоторым категориям. Например, только начальники отдела или руководители могут подать заявку на подключение рабочего места. С учетом всего этого поставлена цель: создать систему заявок для ФИЦ КНЦ СО РАН.

1. Существующие системы. Существует множество систем подобного рода, рассмотрим некоторые из них.

Amelia 2.0. Отечественный продукт. Эффективное управление коммерческой недвижимостью. Предоставляет модули по управлению заявками, паспортизации объектов, систематизации данных по потреблению ресурсов, отслеживанию услуг клининга, учету оборудования и т.д. Это огромная система, которая покрывает большую часть потребностей. Стоимость такой системы от пятнадцати тысяч рублей в месяц, до двух миллионов, зависит от требований заказчика. Имеет маршрутизацию заявок только по категориям работ и отделам [12].

OkDesk. Отечественный продукт. Называют свою систему «HelpDesk», хотя по размерам системы – это не так. Поддерживает множество модулей. Существует гибкий SLA, который зависит не только от категории, но и от приоритета, категории клиента, исполнителя и типа заявки, что является огромным плюсом. Есть много правил для маршрутизации. В системе есть модули для учета техники и программного обеспечения, бухгалтерии и т.д. Стоимость аренды варьируется от восьми тысяч до шестидесяти тысяч рублей в месяц [13].

ITSM365. «ServiceDesk» система. Система позволяет управлять обращениями, имеет отдельную функцию для разграничения обращений на инциденты, запросы на изменения и проблемы, через создание типов услуг, а не через построение множественного дерева категорий. Является отечественным продуктом. Стоимость от пятнадцати до шестидесяти тысяч рублей за 10 лицензий [14].

OTRS. Имеет две версии продукта: платную и бесплатную. Бесплатная версия является открытой, что позволяет доработать данную систему под себя, но не имеет множества модулей (не умеет работать с почтой), на данный момент бесплатная версия не обновляется и поддерживается только сообществом. Платная версия работает только через облако самой компании, имеет множество отдельных модулей и постоянно обновляется [15].

OsTicket. Зарубежный продукт. Существует несколько версий продукта, в том числе и бесплатная версия. Функциональность направлена на работу с заявками и базами знаний. Система проста в обращении. Поддерживает SLA, но заявка попадает к исполнителю в зависимости от категории. На данный момент используется в ФИЦ КНЦ СО РАН. Поддерживает установку на собственные серверы. Системы, описанные выше, работают в основном через облако [16].

HubEx. Данная система представляет собой «ServiceDesk» и FSM-систему. FSM – система управления мобильными сотрудниками. Включает такие функции, как: учет и управление заявками и объектами, планирование расписаний, учет выездов и работ, автоматизация диспетчерской, электронный паспорт оборудования, GPS-контроль, модуль аналитики и отчетности. Отечественный продукт. Стоимость от тринадцати тысяч рублей [17].

2. Материалы и методы. Для создания системы заявок для организации ФИЦ КНЦ СО РАН были использованы следующие инструменты:

1. Vue – фреймворк для разработки пользовательских интерфейсов [18-21].
2. Vue-router – официальная библиотека для маршрутизации в SPA-приложении.
3. Vuex – библиотека управления состоянием приложения.
4. Express.js – веб-фреймворк для Node.js, который обеспечивает функциональность сервера.
5. Node.js – среда выполнения JavaScript, используемая для создания серверной части приложения [22].
6. Sequelize – ORM (Object-Relational Mapping) для работы с базами данных, позволяет создавать модели и обращаться к базе данных без знания языка SQL.
7. Redis – высокопроизводительная система управления базами данных, используемая для кэширования данных, чтобы ускорить их выдачу, без обращения к основной базе данных.
8. Primevue – библиотека UI элементов для создания пользовательских интерфейсов. Поддерживается фреймворком Vue.
9. PostgreSQL – свободная объектно-реляционная база данных [23].

В системе существуют следующие сущности:

1. Область работ – сущность для группировки категорий. Представляет собой дерево, где на узлах могут быть установлены категории. Данная сущность хранит структуру дерева в формате JSON.
2. Категории – это вид работы или проблемы, для которых требуется ответ или выполнение задачи.
3. Папки – фильтры заявок, созданные пользователями. Данная сущность хранит правила для отбора заявок в формате JSON и редактируется через меню «Папки».
4. Чат – массив сообщений от пользователей и исполнителей, которые прикреплены к конкретной заявке. Также здесь размещены основные элементы управления заявкой: передать другому сотруднику, сменить статус заявки, заявка закрыта или отклонить заявку.
5. Организации – обособленные подразделения и головная организация. Необходимы для составления маршрутизации заявок в зависимости от подразделения.
6. Приоритеты – первенство по времени выполнения заявки. В зависимости от выставленного значения, заявка будет помечена определенным цветом в системе. Также доступна сортировка заявок по приоритетам.
7. Роли – набор полномочий, который необходим пользователю для выполнения рабочих задач и ограничивает доступ к частям системы.
8. Статусы – состояние заявки.
9. Шаги выполнения – составная часть «Пути выполнения». Содержит перечисление сотрудников с описанием выполняемых работ.
10. Системные настройки – набор настроек для старта системы. Содержит поля для настройки статусов при открытии и закрытии заявки. Также содержит допустимые почтовые сервисы, с которых можно зарегистрироваться в системе.
11. Заявки – обращения пользователей.
12. Пользователи – учетные записи в системе, которые имеют свои настройки уровня доступа для выполнения какой-либо работы.
13. Пути выполнения – маршрут выполнения заявки.

На рисунке 1 представлена структура клиентской части приложения.

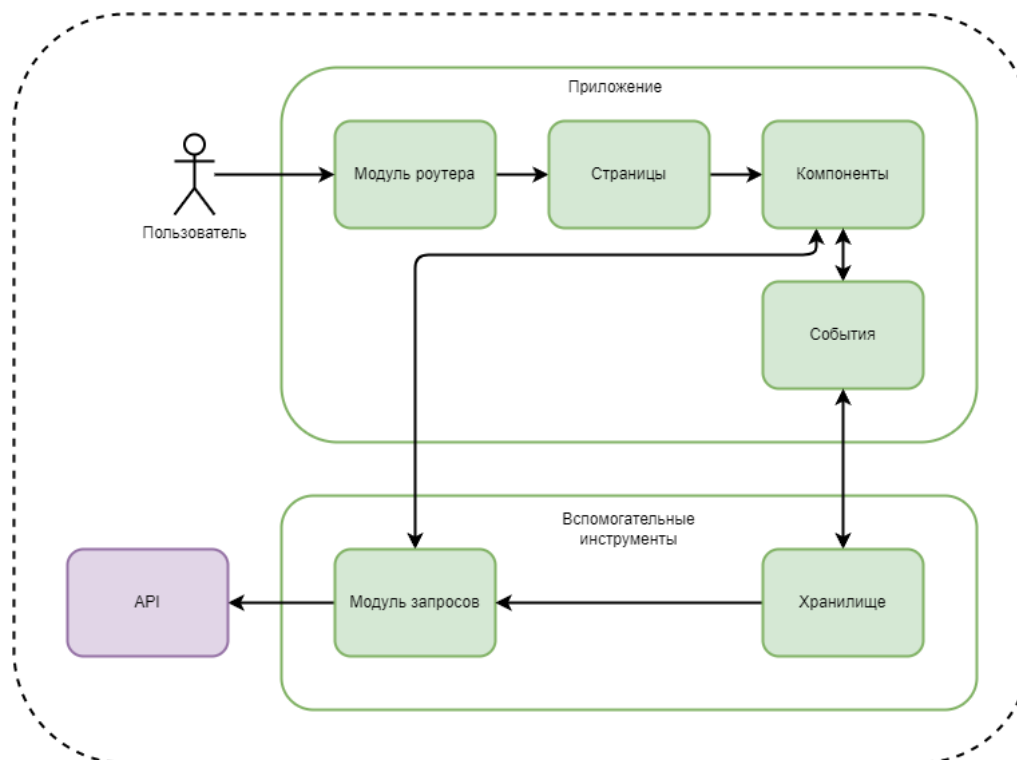


Рис. 1. Структура клиентской части приложения

Данные о пользователе хранятся в JWT. JWT – это JSON объект, используется для передачи данных для авторизации в приложении. Состоит из трех частей – заголовок, тело или полезная нагрузка, и подпись. В теле JWT как раз и хранится вся информация о пользователе. Данная информация передается между клиентом и сервером, когда пользователь заходит в систему, при этом пользователь получает ключ от сервера и ключ хранится в клиентском приложении. При выполнении запросов этот ключ вставляется в заголовок запроса, сервер проверяет заголовок, если заголовок пустой, то значит, пользователь не авторизован, если заголовок не пустой, то сервер выявляет роль пользователя и проверяет, имеет ли пользователь доступ к маршруту для выполнения запроса. Подпись устанавливается в конфигурации сервера, чтобы ключ не могли подделать. Ключ выдается на 8 часов, время можно изменить; при истечении срока система “выбрасывает” пользователя и необходимо заново выполнить вход в систему.

Пользователь при запуске приложения взаимодействует с модулем маршрутизации. Если пользователь не авторизован, то приложение автоматически перенаправляет пользователя на страницу входа. Если авторизован, то загружается та страница, на которую пытается зайти пользователь. Также на некоторых страницах существует уровень доступа, и если пользователь не имеет необходимых прав, то приложение перенаправит пользователя на главную страницу и выдаст ошибку об этом. Каждая страница имеет метаданные, где указывается шаблон страницы, уровень доступа, тип страницы и наименование. Шаблон отвечает за то, какие части страницы будут статистическими, а какие – динамическими. Например, шапка сайта – статическая часть страницы, она должна отображаться везде, а форма для заполнения пользователя или таблица всех пользователей – динамические части страницы и зависят от того, на какой мы странице находимся. Уровень доступа отвечает за то, кто имеет доступ к странице. Тип страницы имеет три значения: создание, изменение и отображение. От этих настроек зависят тип HTTP-запроса и некоторые отображаемые элементы. Например, кнопка «Создать» появится только на странице с типом «Создание», а кнопки удаления и сохранения – при типе

страницы «Изменение». Тип «Отображение» стоит на тех страницах, на которых информация служит только для отображения и ее нельзя изменить.

На рисунке 2 изображена структура серверной части приложения.

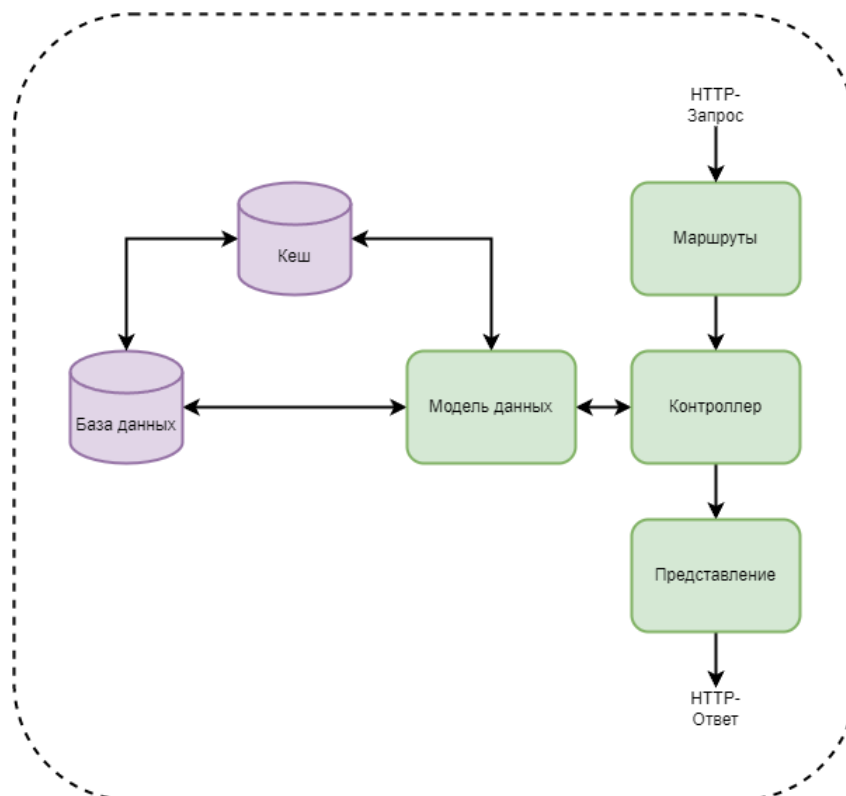


Рис. 2. Структура серверной части приложения

Далее на вход поступает HTTP-запрос. В зависимости от типа запроса и маршрута контроллер выполнит определенное действие. Некоторые данные кешируются, что ускоряет ответ сервера. Кеширование данных необходимо, потому что на данный момент в организации ФИЦ КНЦ СО РАН работают две с половиной тысячи сотрудников. Основным действием пользователя является просмотр данных, что вызывает обращения к базе данных. При редактировании и создании различных сущностей в системе необходима дополнительная информация, что также требует обращения к базе данных. Кеширование ускоряет процесс выдачи данных для просмотра и редактирования элементов системы. У сервера предусмотрен механизм обновления кеша при обновлении сущностей системы.

Модели данных составлены с помощью библиотеки *sequelize*, которая позволяет создавать модели данных, указывать, какой тип у поля будет в базе данных и задавать валидацию данных. Также у маршрутов присутствует связующее программное обеспечение. Перед передачей запроса в контроллер вызывается модуль проверки пользователя, где проверяется его роль и авторизован ли пользователь. Если пользователь не авторизован, то выдается сообщение об ошибке. Если авторизован, то сервер проверяет роль пользователя. Некоторые маршруты имеют уровень доступа. Если уровень доступа не соответствует, то система выводит сообщение об ошибке.

3. Структура базы данных. Для отображения структуры базы данных она была разбита на три части: заявки, пользователи и пути. На рисунках 3-5 показана вся структура базы данных. Напротив поля указан тип данных: t – текст; b – булево значение; d – дата и время; # – целочисленный тип; [] – массив.

Если тип не указан, то это тип данных json или jsonb. На рисунке 3 отображен фрагмент базы данных (БД) «Заявки».

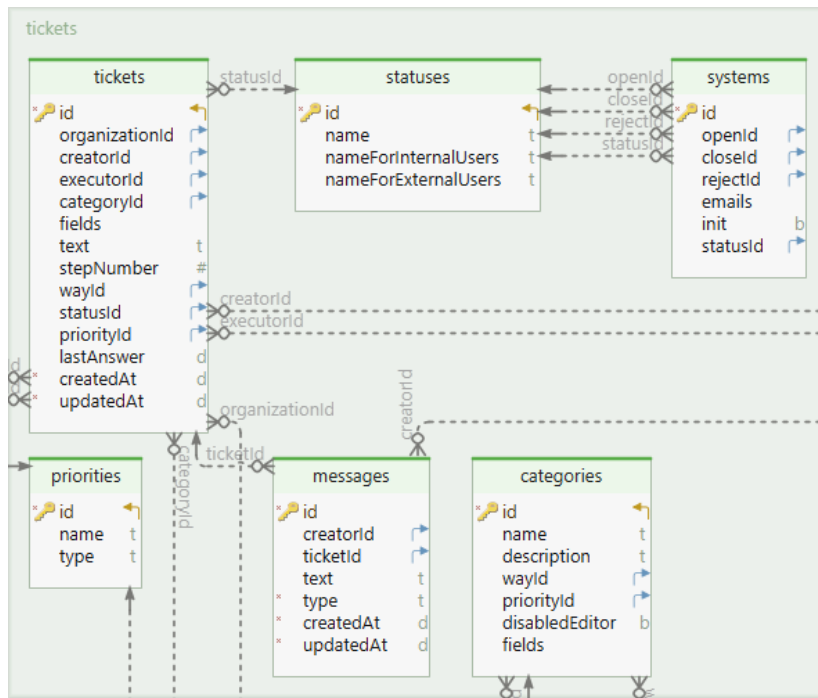


Рис. 3. Фрагмент базы данных «Заявки»

Этот фрагмент включает такие таблицы, как: заявки, статусы, система, приоритеты, сообщения и категории, т.е. содержит основные элементы, связанные с сущностью «Заявка»

На рисунке 4 представлен фрагмент БД «Пользователи».

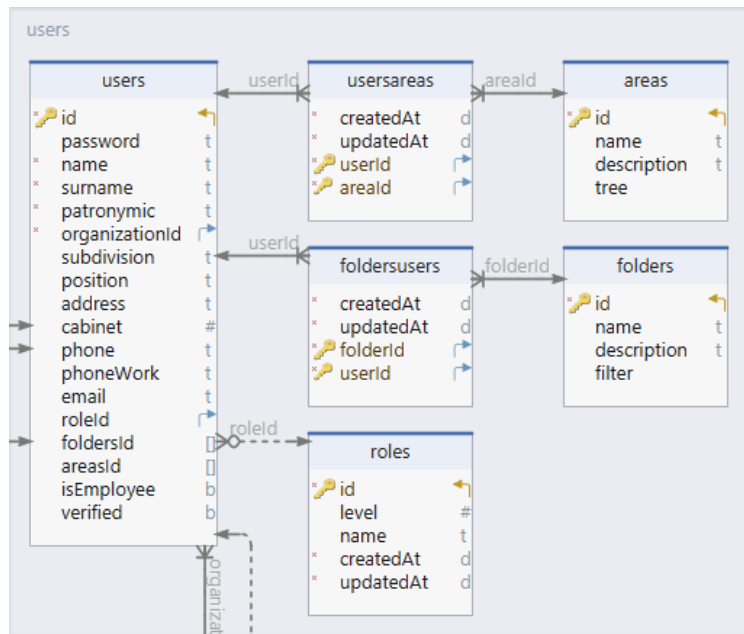


Рис. 4. Фрагмент базы данных «Пользователи»

Этот фрагмент включает таблицы: пользователи, области работ, папки, и роли, т.е. содержит основные элементы, относящиеся к пользователям.

На рисунке 5 представлен фрагмент БД «Пути».

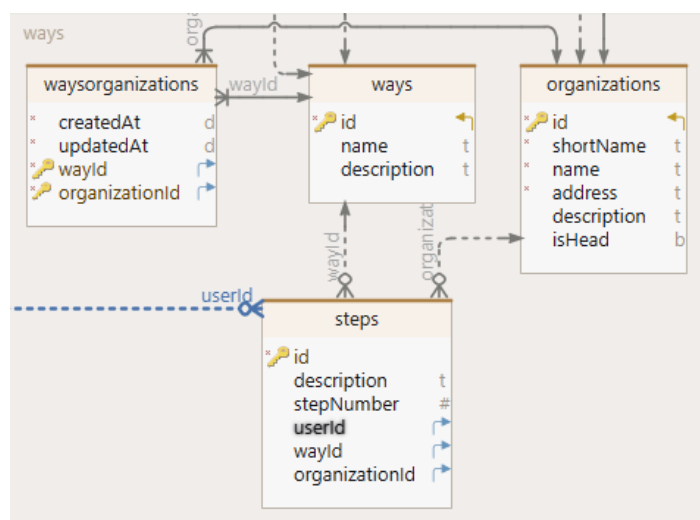


Рис. 5. Фрагмент базы данных «Пути»

Этот фрагмент включает таблицы: пути, организации и шаги, т.е. содержит основные элементы, связанные с сущностью «Путь».

4. Работа системы. Основной задачей системы является маршрутизация заявок в зависимости от организации и категории. Все маршруты заполняются в объекте «Путь», указывается организация, для этой организации перечисляются сотрудники в порядке выполнения работ, затем созданный путь прикрепляется к категории.

На рисунке 6 представлена модель маршрутизации заявки.

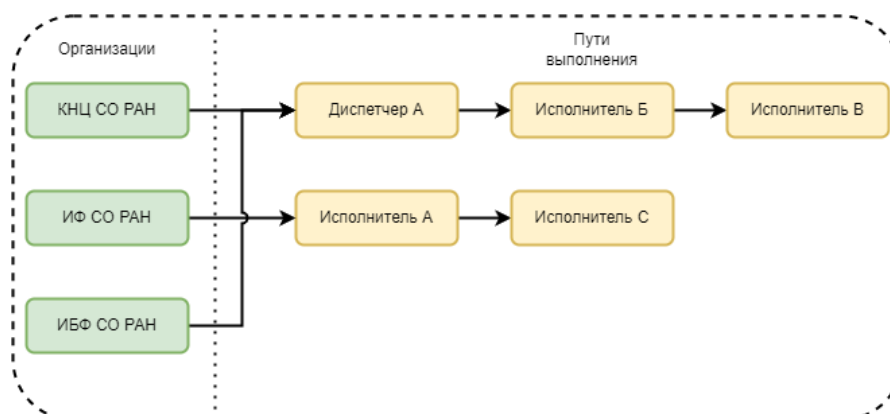


Рис. 6. Маршрутизация заявки с учетом структуры организации

В левой части рисунка перечислены некоторые обособленные подразделения ФИЦ КНЦ СО РАН. Например, необходимо подключить рабочее место работника. Выбирается категория «Подключить рабочее место». В зависимости от того, к какой организации принадлежит заявитель, заявка попадет разным исполнителям:

- если заявитель находится в КНЦ СО РАН, то заявка попадет диспетчеру «А», затем исполнителю «Б», затем исполнителю «В»;
- если заявитель находится в ИФ СО РАН, то заявка попадет исполнителю «А», затем исполнителю «С»;
- если заявитель из ИБФ СО РАН и для этой организации не настроен маршрут выполнения заявки, то заявка пойдет по маршруту головной организации – КНЦ СО РАН. Естественно, головную организацию можно указать в системе.

На рисунке 7 представлен пример, когда исполнитель «Б» не может выполнить свою часть работы по каким-либо причинам.

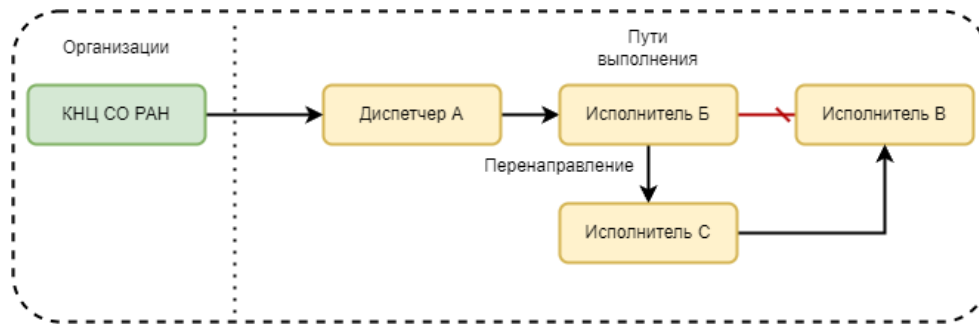


Рис. 7. Маршрутизация заявки с перенаправлением на другого исполнителя

В таком случае исполнитель «Б» перенаправляет свою заявку на исполнителя «С», который выполнит работу и направит следующему исполнителю «В» по маршруту заявки; это перенаправление будет выполнено автоматически, в зависимости от настроенного маршрута заявки. В системе заявка всегда движется по составленному маршруту. При составлении маршрута указываются необходимые шаги для выполнения этой заявки. К шагу прикрепляется описание работ и пользователь (userId), который является исполнителем (isEmployee). Данный исполнитель меняется автоматически, следуя составленному маршруту, либо вручную. Если требуется переназначить заявку другому сотруднику, то для этого есть инструменты внутри самой заявки, можно выбрать абсолютно любого исполнителя через меню чата. Можно указать исполнителя из любой организации или подразделения, независимо от категории, составленного маршрута заявки и организации заявителя.

На рисунке 8 представлен пример созданной заявки. Слева расположена боковая панель с навигацией и папки. В нижней части меню управления заявкой: открыть чат, передать предыдущему исполнителю, передать следующему, отклонить заявку и выполнить заявку.

Пользователь		Заявка	
ФИО	Малимонов Максим Игоревич	Дата создания	30.02.2023, 18:51:01
Организация	КНЦ СО РАН	Срок сдачи	30.02.2023, 18:51:01
Подразделение	ИТ	Последний ответ	30.02.2023, 18:51:01
Должность	Младший научный сотрудник	Статус	Открыта
Кабинет	101	Приоритет	Средний
Адрес	КНЦ СО РАН	Назначена	Исполнитель не назначен
Телефон		Категория	Тестовая Категория
Телефон рабочий		Путь	Тестовый
Почта	thelemoh@yandex.ru		

Дополнительные поля

Heading Sans Serif B I U A [Icons]

Решите мою проблему

[Chat] [Previous] [Next] [Cancel] [Check]

Рис. 8. Визуализация формы заявки

Заявитель может отслеживать выполнение заявки и видеть, к кому попадет заявка на всем пути следования. Также для каждого шага на маршруте описывается работа исполнителя и заявитель знает, что должен сделать исполнитель.

Таким образом, можно обеспечить прозрачную и понятную работу ИТ-отделов, а не представлять ее в виде черного ящика.

Заключение. В статье описана разработка системы заявок для организации ФИЦ КНЦ СО РАН. Следует отметить, что внедрение системы учета заявок с маршрутизацией по организациям для ФИЦ КНЦ СО РАН является значимым событием. Благодаря новой системе процесс обработки заявок станет более прозрачным, быстрым и удобным для всех участников. Кроме того, система позволит более точно контролировать выполнение задач и улучшить взаимодействие между различными подразделениями ФИЦ КНЦ СО РАН. В целом, внедрение системы учета заявок с маршрутизацией по организациям является важным шагом в совершенствовании работы организации и повышении ее эффективности.

Список источников

1. Al-Hawari F., Barham H. A machine learning based help desk system for IT service management. *Journal of King Saud university – Computer and information sciences*, 2021, no. 33, pp. 702-718.
2. Serdar Serbest, Yilmaz Goksen, Onur Doğan, Anıl Tokdemir Design and implementation of Help Desk system on the effective focus of information system. *Procedia economics and finance*, 2015, no. 33, pp. 461-467.
3. Akinnuwesi B.A., Enikuomehin O.A., Uzoka, F.E., et al. Electronic helpdesk support system in tertiary institutions in developing countries. *Int. J. Comput. Inf. Technol.*, 2014, 3 (6), pp. 1-12.
4. Wongsakthawom R., Limpiyakorn Y. Development of IT helpdesk with microservices. In: *IEEE 8th International conference on electronics information and emergency communication (ICEIEC)*, 2018, pp. 31-34.
5. Zhou Y., Tong Y., Gu R., Gall H. Combining text mining and data mining for bug report classification. *J. Software, Evol. Process*, 2016, no. 28 (3), pp. 150-176.
6. Zhu X., Zhou W., Li T. A visual tool for ticket monitoring and management. In: *2017 IEEE 12th International conference on intelligent systems and knowledge engineering (ISKE)*, 2017, 2016, 107 p.
7. Serbest S., Goksen Y., Dogan O., Tokdemir A. Design and implementation of help desk system on the effective focus of information system. *Procedia economics and finance*, 2015, 33, pp. 461-467.
8. Tanovic A., Mastorakis N.E. Advantage of using service desk management systems in real organizations. *Int. J. Econ. Manage. Syst.*, 2016, 1, pp. 81-86.
9. Tanovic A., Orucevic F., Butkovic A. Advantages of the implementation of service desk based on ITIL framework in telecommunication industry. *Math. Comput. Sci. Ind.*, 2016. 1, pp. 244–258.
10. Adrian Paschke, Martin Bichler Knowledge representation concepts for automated SLA management. *Decision support systems*, 2008. no. 46, pp. 187-205
11. Maria-Cruz Valiente, Elena Garcia-Barriocanal, Miguel-Angel Sicilia. Applying an ontology approach to IT service management for business-IT integration. *Knowledge-Based systems*, 2012, no. 28, pp. 76-87.
12. Amelia 2.0. Цифровая эксплуатация недвижимости. – URL: <https://info.amelia-st.ru/> (дата обращения: 10.06.2023).
13. Okdesk. Help desk система учета заявок. – URL: <https://okdesk.ru/> (дата обращения: 10.06.2023).
14. ITSM365. Help desk система учета заявок. – URL: <https://itsm365.com/> (дата обращения: 12.02.2024).
15. OTRS. Software solutions for customer service, ITSM, ISMS and cyber defense, available at: <https://otrs.com/>. (accessed:12.02.2024).
16. osTicket. Support ticketing system. Available at: <https://osticket.com/> (accessed:10.06.2023).
17. Hubex. Управление мобильными сотрудниками – FSM HubEx. – URL: <https://hubex.ru/> (дата обращения: 12.02.2024).
18. Malimonov M.I. Yakubailik O.E. Cloud-based software tools for the rapid development of web gis, *CEUR workshop proceedings*, 2021, no. 3006, pp. 172-179.
19. Shiyong Xiong, Xue Wang, Zhengpeng Lan. Model research of visual report components. *Procedia computer science*, 2022, no. 208, pp. 478-485.
20. Nian L, Bo Z. The research on single page application front-end development based on vue. *Journal of physics: conference series*, 2021, no. 1883(1).
21. Qiu Bin Liu, Sheng Gao, Bi Feng Guo, Hui Liu, Yong Zhao Feng, Hao Xia. Research and development of movie social system. *Procedia computer science*, 2020, no. 166, pp. 154-159.

22. Fabian Kaimer, Philipp Brune Return of the JS: Towards a Node.js-Based software architecture for combined CMS/CRM applications. *Procedia computer science*, 2018, no. 141, pp. 454-459.
23. Massimiliano Pirani, Alessandro Cucchiarelli, Luca Spalazzi Paradigms for database-centric application interfaces. *Procedia computer science*. 2023, no. 217, pp. 835-845.

Малимонов Максим Игоревич. Младший научный сотрудник Красноярского научного центра СО РАН. Основное направление – программирование. SPIN: 7645-6804, malimonov.mi@ksc.krasn.ru, 660036, г. Красноярск, ул. Академгородок, 50.

UDC 004.415.25

DOI:10.25729/ESI.2024.34.2.014

Ticketing system for IT-department of KSC SB RAS

Maksim I. Malimonov

Federal researcher center “Krasnoyarsk scientific center SB RAS”,
Russia, Krasnoyarsk, malimonov.mi@ksc.krasn.ru

Abstract. This article describes the process of creating an application system for the KSC SB RAS using the following technologies and tools: vue, express, node, sequelize, redis and primevue. The article discusses in detail the architecture of the application “Application accounting system for IT departments of the KSC SB RAS”, including the client and server parts, data caching system and user interface. The creation of such a system ensures high performance and ease of use for users.

Keywords: ticket, ticket system, servicedesk, helpdesk, vue

References

1. Al-Hawari F., Barham H. A machine learning based help desk system for IT service management. *Journal of King Saud university – Computer and information sciences*, 2021, no. 33, pp. 702-718.
2. Serdar Serbest, Yilmaz Goksen, Onur Doğan, Anıl Tokdemir Design and implementation of Help Desk system on the effective focus of information system. *Procedia economics and finance*, 2015, no. 33, pp. 461-467.
3. Akinnuwesi B.A., Enikuomelin O.A., Uzoka, F.E., et al. Electronic helpdesk support system in tertiary institutions in developing countries. *Int. J. Comput. Inf. Technol.*, 2014, 3 (6), pp. 1-12.
4. Wongsakthawom R., Limpiyakorn Y. Development of IT helpdesk with microservices. In: *IEEE 8th International conference on electronics information and emergency communication (ICEIEC)*, 2018, pp. 31-34.
5. Zhou Y., Tong Y., Gu R., Gall H. Combining text mining and data mining for bug report classification. *J. Software, Evol. Process*, 2016, no. 28 (3), pp. 150-176.
6. Zhu X., Zhou W., Li T. A visual tool for ticket monitoring and management. In: *2017 IEEE 12th International conference on intelligent systems and knowledge engineering (ISKE)*, 2017, 2016, 107 p.
7. Serbest S., Goksen Y., Dogan O., Tokdemir A. Design and implementation of help desk system on the effective focus of information system. *Procedia economics and finance*, 2015, 33, pp. 461-467.
8. Tanovic A., Mastorakis N.E. Advantage of using service desk management systems in real organizations. *Int. J. Econ. Manage. Syst.*, 2016, 1, pp. 81-86.
9. Tanovic A., Orucevic F., Butkovic A. Advantages of the implementation of service desk based on ITIL framework in telecommunication industry. *Math. Comput. Sci. Ind.*, 2016. 1, pp. 244–258.
10. Adrian Paschke, Martin Bichler Knowledge representation concepts for automated SLA management. *Decision support systems*, 2008. no. 46, pp. 187-205
11. Maria-Cruz Valiente, Elena Garcia-Barriocanal, Miguel-Angel Sicilia. Applying an ontology approach to IT service management for business-IT integration. *Knowledge-Based systems*, 2012, no. 28, pp. 76-87.
12. Amelia 2.0. Tsigrovaya ekspluatatsiya nedvizhimosti [Amelia 2.0. Digital operation of buildings]. Available at: <https://info.amelia-st.ru/> (accessed: 06/10/2023).
13. Okdesk. Help desk sistema ucheta zayavok [Okdesk. Help desk ticket system]. Available at: <https://okdesk.ru/> (accessed: 06/10/2023).
14. ITSM365. Help desk sistema ucheta zayavo [ITSM365. Help desk ticket system]. Available at: <https://itsm365.com/> (accessed: 02/12/2024).

15. OTRS. Software solutions for customer service, ITSM, ISMS and cyber defense, available at: <https://otrs.com/>. (accessed:12.02.2024).
16. osTicket. Support ticketing system. Available at: <https://osticket.com/> (accessed:10.06.2023).
17. Hubex. Upravleniye mobil'nymi sotrudnikami – FSM HubEx [Hubex. Mobile employee management – FSM HubEx]. Available at: <https://hubex.ru/> (accessed: 02/12/2024)
18. Malimonov M.I. Yakubailik O.E. Cloud-based software tools for the rapid development of web gis, CEUR workshop proceedings, 2021, no. 3006, pp. 172-179.
19. Shiyong Xiong, Xue Wang, Zhengpeng Lan. Model research of visual report components. Procedia computer science, 2022, no. 208, pp. 478-485.
20. Nian L, Bo Z. The research on single page application front-end development based on vue. Journal of physics: conference series, 2021, no. 1883(1).
21. Qiu Bin Liu, Sheng Gao, Bi Feng Guo, Hui Liu, Yong Zhao Feng, Hao Xia. Research and development of movie social system. Procedia computer science, 2020, no. 166, pp. 154-159.
22. Fabian Kaimer, Philipp Brune Return of the JS: Towards a Node.js-Based software architecture for combined CMS/CRM applications. Procedia computer science, 2018, no. 141, pp. 454-459.
23. Massimiliano Pirani, Alessandro Cucchiarelli, Luca Spalazzi Paradigms for database-centric application interfaces. Procedia computer science. 2023, no. 217, pp. 835-845.

Malimonov Maxim Igorevich. Junior researcher, Krasnoyarsk scientific center SB RAS. The main direction is programming. SPIN: 7645-6804, malimonov.mi@ksc.krasn.ru, 660036, Krasnoyarsk, st. Akademgorodok, 50.

Статья поступила в редакцию 21.06.2023; одобрена после рецензирования 02.04.2024; принята к публикации 30.05.2024.

The article was submitted 06/21/2023; approved after reviewing 04/02/2024; accepted for publication 05/30/2024.