

Описание технической архитектуры программного обеспечения «Шаникс»

1. Общие сведения

Настоящий документ содержит описание технической архитектуры программного обеспечения «Шаникс».

Программное обеспечение «Шаникс» предназначено для автоматизации работы с сервисными заявками.

Правообладатель программного обеспечения: ООО «ШАНИКС».

Документ подготовлен для предоставления в составе комплекта документов при регистрации программного обеспечения в Едином реестре российских программ для электронных вычислительных машин и баз данных.

В рамках настоящего документа описывается программное обеспечение «Шаникс», включающее web-интерфейс, мобильное приложение, серверную часть, API, базу данных, файловое хранилище, механизмы уведомлений и отчетности.

2. Назначение программного обеспечения

Программное обеспечение «Шаникс» представляет собой сервис-деск систему, предназначенную для автоматизации процессов приема, обработки и контроля выполнения сервисных заявок.

Система позволяет организовать работу с заявками в едином интерфейсе, назначать исполнителей, контролировать сроки реакции и решения, учитывать SLA, хранить историю обработки заявок, работать с вложениями, направлять уведомления пользователям и формировать отчетность.

Основные задачи программного обеспечения:

1. прием и регистрация сервисных заявок;
2. назначение исполнителей по заявкам;
3. контроль статусов заявок;
4. отслеживание времени реакции;
5. отслеживание времени решения;
6. контроль соблюдения SLA;
7. хранение истории работы по заявкам;
8. хранение файлов и вложений, связанных с заявками;
9. отправка почтовых уведомлений пользователям;
10. отправка push-уведомлений пользователям мобильного приложения;
11. формирование отчетов по заявкам, исполнителям, срокам и SLA.

3. Пользователи программного обеспечения

В программном обеспечении предусмотрена работа пользователей с разными ролями и уровнями доступа.

| Роль пользователя | Основные действия |
|--------------------------|--|
| Пользователь / заявитель | Создание сервисных заявок, просмотр статуса своих заявок, добавление комментариев и вложений |
| Исполнитель | Получение назначенных заявок, изменение статусов, фиксация выполненных работ, добавление комментариев и вложений |
| Руководитель / контролер | Контроль сроков реакции и решения, просмотр загрузки исполнителей, анализ SLA и отчетности |
| Администратор | Настройка пользователей, ролей, справочников, параметров системы и прав доступа |

Пользователи могут работать с программным обеспечением через web-интерфейс или мобильное приложение, в зависимости от доступного способа использования.

Перечень ролей может быть расширен или изменен в зависимости от настроек конкретной инсталляции программного обеспечения.

4. Общая архитектура программного обеспечения

Программное обеспечение «Шаникс» построено по клиент-серверной архитектуре.

Клиентская часть программного обеспечения реализована с использованием **Flutter**. На базе Flutter формируется web-интерфейс программного обеспечения, а также мобильные приложения для поддерживаемых мобильных платформ.

Web-интерфейс обеспечивает пользователям доступ к функциям работы с сервисными заявками через браузер.

Мобильное приложение обеспечивает пользователям доступ к функциям работы с сервисными заявками с мобильных устройств.

Серверная часть программного обеспечения реализована на базе **Laravel**. Она выполняет обработку бизнес-логики, управление заявками, пользователями, ролями, статусами, SLA, вложениями, почтовыми уведомлениями, push-уведомлениями и отчетностью.

Для хранения прикладных данных используется СУБД **PostgreSQL**.

Взаимодействие web-интерфейса и мобильного приложения с серверной частью выполняется через программный интерфейс API.

5. Состав основных компонентов

В состав программного обеспечения «Шаникс» входят следующие основные компоненты:

| Компонент | Технология / средство | Назначение |
|-----------------------------|--|---|
| Web-интерфейс | Flutter Web | Пользовательский интерфейс для работы с заявками, комментариями, вложениями, статусами, отчетами и настройками через браузер |
| Мобильное приложение | Flutter | Пользовательский интерфейс для работы с заявками, комментариями, вложениями, статусами и уведомлениями на мобильных устройствах |
| Серверная часть / backend | Laravel | Обработка бизнес-логики, управление заявками, пользователями, SLA, уведомлениями и отчетностью |
| API | Laravel API | Обмен данными между клиентскими приложениями и серверной частью |
| База данных | PostgreSQL | Хранение заявок, пользователей, ролей, статусов, SLA, истории обработки и отчетных данных |
| Файловое хранилище | Серверное файловое хранилище | Хранение файлов и вложений, связанных с заявками |
| Модуль почтовых уведомлений | Laravel / почтовый сервис | Формирование и отправка уведомлений пользователям по электронной почте |
| Модуль push-уведомлений | Laravel / сервис доставки push-уведомлений | Формирование и отправка push-уведомлений пользователям мобильного приложения |
| Модуль отчетности | Laravel / серверная часть | Формирование отчетов по заявкам, срокам реакции, срокам решения и SLA |
| Модуль контроля SLA | Laravel / серверная часть | Расчет и контроль сроков реакции и решения по заявкам |

6. Логическая схема архитектуры

Общая схема взаимодействия компонентов программного обеспечения:

Пользователь через браузер



Web-интерфейс (Flutter Web)



API



Серверная часть (Laravel)



База данных (PostgreSQL)

Пользователь через мобильное устройство



Мобильное приложение (Flutter)



API



Серверная часть (Laravel)



База данных (PostgreSQL)

Дополнительно серверная часть взаимодействует со следующими компонентами:

- файловое хранилище — для хранения вложений и прикрепленных файлов;
 - модуль почтовых уведомлений — для отправки уведомлений по электронной почте;
 - модуль push-уведомлений — для отправки уведомлений пользователям мобильного приложения.
- Web-интерфейс и мобильное приложение не обращаются к базе данных напрямую. Все операции с данными выполняются через серверную часть и API.

7. Описание взаимодействия компонентов

Пользователь работает с программным обеспечением «Шаникс» через web-интерфейс или мобильное приложение.

Web-интерфейс реализован с использованием Flutter Web и предназначен для работы пользователей через браузер.

Мобильное приложение реализовано с использованием Flutter и предназначено для работы пользователей на мобильных устройствах.

Клиентские приложения направляют запросы к серверной части через API. Через API выполняются основные операции: создание заявок, просмотр списка заявок, изменение

статусов, назначение исполнителей, добавление комментариев, загрузка и просмотр вложений, получение отчетов и данных по SLA.

Серверная часть на Laravel принимает API-запросы, выполняет проверку прав доступа, обрабатывает бизнес-логику и обращается к базе данных PostgreSQL.

База данных PostgreSQL используется для хранения основной информации системы, включая заявки, пользователей, роли, статусы, историю обработки заявок, параметры SLA и данные, необходимые для формирования отчетности.

Файловое хранилище используется для хранения файлов и вложений, прикрепленных к заявкам.

Модуль почтовых уведомлений формирует и отправляет пользователям уведомления по электронной почте о событиях, связанных с обработкой заявок.

Модуль push-уведомлений формирует уведомления для пользователей мобильного приложения о событиях, связанных с обработкой заявок.

Ответы серверной части возвращаются клиентским приложениям в структурированном виде через API.

8. Хранение и обработка данных

Основные данные программного обеспечения хранятся в базе данных PostgreSQL.

В базе данных хранятся следующие группы данных:

| Группа данных | Описание |
|----------------------|--|
| Пользователи | Учетные записи пользователей системы |
| Роли и права доступа | Роли пользователей и настройки доступа к функциям системы |
| Заявки | Основные данные сервисных заявок |
| Исполнители | Данные о назначении заявок ответственным исполнителям |
| Статусы заявок | Текущие и исторические статусы обработки заявок |
| Комментарии | Комментарии пользователей и исполнителей по заявкам |
| История обработки | Сведения об изменениях, комментариях и действиях пользователей по заявкам |
| SLA | Параметры сроков реакции и сроков решения |
| Почтовые уведомления | Данные, необходимые для формирования и отправки уведомлений по электронной почте |
| Push-уведомления | Данные, необходимые для отправки уведомлений пользователям мобильного приложения |
| Отчетные данные | Данные, используемые для формирования отчетности |

Файлы и вложения, прикрепляемые к заявкам, хранятся в файловом хранилище на серверной инфраструктуре правообладателя.

Обработка данных выполняется серверной частью программного обеспечения. Web-интерфейс и мобильное приложение используются для отображения данных и передачи действий пользователя на сервер через API.

9. Развертывание программного обеспечения

Серверная часть программного обеспечения «Шаникс» размещается на серверной инфраструктуре правообладателя.

Развертывание серверной части программного обеспечения на серверной инфраструктуре заказчика не требуется. Пользователи получают доступ к системе через web-интерфейс или мобильное приложение, которые взаимодействуют с серверной частью, размещенной на сервере правообладателя.

В состав серверной инфраструктуры входят:

| Узел / компонент | Назначение |
|----------------------------------|---|
| Сервер приложения | Размещение серверной части программного обеспечения на Laravel |
| Сервер базы данных | Размещение СУБД PostgreSQL и прикладных данных |
| API | Обработка запросов web-интерфейса и мобильного приложения |
| Файловое хранилище | Хранение файлов и вложений, связанных с заявками |
| Почтовый сервис | Отправка уведомлений пользователям по электронной почте |
| Сервис доставки push-уведомлений | Доставка push-уведомлений пользователям мобильного приложения |
| Средства резервного копирования | Создание резервных копий базы данных, файлового хранилища и конфигурационных данных |

Допускается размещение серверной части, базы данных и файлового хранилища на одном сервере при условии соответствия требованиям производительности, безопасности и надежности.

Мобильное приложение устанавливается на мобильные устройства пользователей и взаимодействует с серверной частью программного обеспечения через API.

10. Безопасность и разграничение доступа

В программном обеспечении применяются механизмы аутентификации пользователей и разграничения прав доступа.

Доступ к функциям системы определяется ролью пользователя. Пользователь получает доступ только к тем операциям и данным, которые соответствуют его полномочиям.

В системе применяются следующие механизмы безопасности:

1. аутентификация пользователей;
2. разграничение прав доступа по ролям;
3. ограничение доступа к административным функциям;
4. контроль действий пользователей в рамках обработки заявок;
5. хранение истории изменений по заявкам;
6. использование серверной части как единой точки доступа к данным;
7. отсутствие прямого доступа web-интерфейса и мобильного приложения к базе данных;
8. ограничение доступа к файлам и вложениям в соответствии с правами пользователя;
9. передача данных между клиентскими приложениями и серверной частью через API.

Все операции с данными выполняются через серверную часть программного обеспечения.

11. Масштабирование

Масштабирование программного обеспечения может выполняться за счет увеличения вычислительных ресурсов сервера, на котором размещены серверная часть программного обеспечения, база данных и файловое хранилище.

При росте нагрузки возможно увеличение ресурсов сервера: процессорной мощности, объема оперативной памяти, дискового пространства и производительности подсистемы хранения данных.

12. Отчетность

Программное обеспечение содержит функции формирования отчетности по сервисным заявкам.

Отчетность может использоваться для анализа:

1. количества поступивших заявок;
2. распределения заявок по исполнителям;
3. статусов заявок;
4. сроков реакции;
5. сроков решения;
6. соблюдения SLA;
7. общей нагрузки на исполнителей и подразделения.

Формирование отчетности выполняется серверной частью программного обеспечения на основании данных, хранящихся в базе данных PostgreSQL.

13. Уведомления

Программное обеспечение содержит механизмы почтовых уведомлений и push-уведомлений.

Почтовые уведомления используются для информирования пользователей по электронной почте о событиях, связанных с обработкой сервисных заявок.

Push-уведомления используются для информирования пользователей мобильного приложения о событиях, связанных с обработкой сервисных заявок.

Уведомления могут направляться в следующих случаях:

1. создание новой заявки;
2. назначение исполнителя;
3. изменение статуса заявки;
4. добавление комментария;
5. приближение или нарушение срока реакции;
6. приближение или нарушение срока решения;
7. иные события, связанные с обработкой заявки.

Формирование уведомлений выполняется серверной частью программного обеспечения.

14. Внешние интеграции

В базовой архитектуре программного обеспечения web-интерфейс и мобильное приложение взаимодействуют с серверной частью через API.

Для отправки почтовых уведомлений используется почтовый сервис.

Для доставки push-уведомлений используется сервис доставки push-уведомлений мобильных платформ.

Сведения о дополнительных внешних интеграциях указываются при их наличии.

| Внешняя система | Способ взаимодействия | Назначение |
|----------------------------------|---------------------------------------|---|
| Почтовый сервис | SMTP / иной почтовый протокол | Отправка уведомлений пользователям по электронной почте |
| Сервис доставки push-уведомлений | API сервиса доставки push-уведомлений | Доставка push-уведомлений пользователям мобильного приложения |

Дополнительные внешние интеграции в настоящем описании не рассматриваются.

15. Состав поставки

В состав программного обеспечения «Шаникс» входят:

1. web-интерфейс, реализованный с использованием Flutter Web;
2. мобильное приложение, реализованное с использованием Flutter;
3. серверная часть, реализованная на Laravel;
4. API для взаимодействия web-интерфейса и мобильного приложения с серверной частью;
5. структура базы данных PostgreSQL;
6. программные модули обработки сервисных заявок;
7. программные модули назначения исполнителей;
8. программные модули контроля сроков реакции и решения;
9. программные модули контроля SLA;
10. программные модули формирования отчетности;
11. программные модули работы с файлами и вложениями;

12. программные модули отправки почтовых уведомлений;
13. программные модули формирования push-уведомлений;
14. конфигурационные файлы и служебные компоненты, необходимые для работы программного обеспечения.

16. Заключение

Техническая архитектура программного обеспечения «Шаникс» обеспечивает автоматизацию процессов работы с сервисными заявками, включая прием заявок, назначение исполнителей, контроль сроков реакции и решения, учет SLA, работу с вложениями, отправку уведомлений и формирование отчетности.

Клиентская часть программного обеспечения реализована с использованием Flutter и обеспечивает формирование web-интерфейса, а также мобильного приложения для поддерживаемых мобильных платформ.

Серверная часть программного обеспечения реализована на Laravel. Хранение прикладных данных выполняется в PostgreSQL. Взаимодействие web-интерфейса и мобильного приложения с серверной частью выполняется через API.

Размещение серверной части программного обеспечения выполняется на серверной инфраструктуре правообладателя.