

Ю.А. Скудняков, Д.А. Калачев, И.И. Шпак
АВТОМАТИЗИРОВАННАЯ СИСТЕМА СБОРА И АНАЛИЗА
ЛОГИРУЕМОЙ ИНФОРМАЦИИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА
ОСНОВЕ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID

Белорусский государственный университет информатики и
радиоэлектроники, г.Минск

Разработана автоматизированная система (АС) сбора и анализа логируемой информации разрабатываемых и используемых мобильных приложений (МП) на основе операционной системы Android. Автоматизированный процесс сбора и анализа логируемой информации осуществляется с помощью разработанного программного средства (ПС). Использование АС позволяет упростить и ускорить этапы разработки, тестирования и поддержки мобильного приложения, что значительно повышает эффективность работы разработчиков и тестировщиков мобильных приложений, а также предоставляет более широкие возможности для поддержки существующих приложений.

Ключевые слова: автоматизированная система; операционная система Android; логируемая информация; мобильное приложение; программное средство; клиент; сервер; система управления базой данных.

Введение

Целью данной работы являлась разработка АС сбора и анализа логируемой информации, использование которой позволяет повысить эффективность данного процесса с точки зрения упрощения и сокращения разработки, тестирования и поддержки МП.

Объектом исследования являлись процессы сбора и анализа логируемой информации разрабатываемых и используемых МП на основе операционной системы Android.

Сбор и анализ логируемой информации на мобильных устройствах позволяет получать широкий диапазон информации о произошедших у пользователей МП ошибках и другие сообщения, способствующие производить анализ действий пользователя и улучшать качество МП.

Логируемая информация может быть следующих типов: сообщения об ошибках и аварийном закрытии приложения; информация об устройстве, на котором запущено приложение; параметры сессии пользователя мобильного приложения; специальные сообщения, логируемые разработчиками для отправки на сервер и используемые для анализа действий пользователей приложения.

Дополнительно к основным возможностям средства сбора и анализа логируемой информации часто имеют реализованную интеграцию со сторонними ПС, а также используемыми при разработке мобильного приложения.

Использование интеграции позволяет легко передать интересующую разработчиков информацию в другие ПС, а также для применения ПС управления задачами или ПС коммуникации. В случае с ПС управления задачами использование интеграции позволяет переносить информацию о новых ошибках в систему и создаёт новые задачи для разработчиков. Примерами таких ПС являются: Pivotal, Trello, Jira. Интеграция с ПС коммуникации позволяет оповестить всех заинтересованных в этом разработчиков о новых ошибках, произошедших в системе, наличие которых может потребовать их срочного устранения. Примерами таких ПС являются: Slack, HipChat.

Также такие ПС имеют возможность расчёта аналитики о количестве пользователей и их сессиях использования приложения на основании логируемой информации.

Обзор существующих АС сбора и анализа логируемой информации показал, что большинство из них имеют ограниченные или избыточные функциональные возможности, что в первом случае решаются несложные задачи, а во втором повышается сложность использования систем [1].

Для проведения обзора существующих аналогов были выбраны АС, в основе функционирования которых используются следующие ПС: «Aptelligent»; «SplunkMint»; «Crashlytics».

На основе результатов проведенного анализа данных ПС выявлены их достоинства и недостатки: возможность подключения модуля сбора метрик скорости выполнения http-запросов; большой выбор возможных интеграций (достоинства ПС «Aptelligent»); сложный интерфейс; ограниченный бесплатный набор функционала; нестабильная работа; нет поддержки русского языка (недостатки ПС «Aptelligent»); наличие очень большого набора возможностей; постоянное развитие и поддержка ПС (достоинства ПС «SplunkMint»); сложный интерфейс; высокая стоимость; нет поддержки русского языка (недостатки ПС «SplunkMint»); постоянное развитие и поддержка проекта; простой, удобный и понятный интерфейс; большой набор возможных интеграций; возможность получить состояние различных сенсоров устройства (достоинства ПС «Crashlytics»); нет возможности программно отправлять сообщения на сервер; нет поддержки русского языка (недостатки ПС «Crashlytics»).

Архитектура АС

В работе было осуществлено обоснование выбора архитектуры АС, а для разработки – ПС–платформ и языков программирования, системы управления базами данных, стека технологий для клиентской и серверной частей.

Архитектура АС состоит из клиентской, серверной частей и android-библиотеки с подключением системы управления базой данных (СУБД) (рис.1).

Клиентская часть АС отвечает за взаимодействие с пользователем, графический интерфейс пользователя и взаимодействие с программным интерфейсом серверного приложения.

Серверное приложение взаимодействует с базой данных, предоставляет программный интерфейс для клиентского веб-приложения и библиотеки, обеспечивает целостность данных и отвечает за безопасность всей системы.

Android-библиотека подключается к android-приложениям на стадии их разработки и отвечает за сбор и отправку логируемой информации на серверное приложение.

В качестве СУБД выбрана объектно-реляционная система управления базами данных PostgreSQL [2], обладающая следующими достоинствами: наиболее развитая СУБД с открытым кодом; надежность и устойчивость на очень больших нагрузках; кросс-платформенность: работает в широком диапазоне диалектов UNIX (Linux, FreeBSD, Solaris и т.д.), а также на платформе Microsoft Windows; высокий уровень соответствия стандартам; существует множество интерфейсов и библиотек взаимодействия для других языков: Java (JDBC), ODBC, Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme и Qt; поддержка баз данных практически неограниченного размера; наследование; быстрое действие; расширяемость. СУБД может находиться как на одном сервере с серверной частью АС, так и на отдельном сервере.

Взаимодействие между компонентами АС происходит по протоколу HTTP, используя сообщения в текстовом формате обмена данными JSON, который за счёт своей лаконичности, по сравнению с XML, может быть более подходящим для сериализации сложных структур, так как сериализованная структура имеет меньший размер.

Основными достоинствами данной архитектуры являются: использование REST подхода к построению приложения; возможность реализовать своих клиентов или библиотек под каждую платформу за счёт использования одного REST-интерфейса; снижена нагрузка на сервер по сравнению с обычным веб-приложением за счёт того, что сервер будет отвечать только за предоставление данных; вся информация передается в формате JSON, что сокращает объём передаваемой информации и нагрузку на сервер, также есть возможность включить сжатие передаваемой информации; используя браузер, можно сделать универсальный клиент под все платформы и операционные системы.

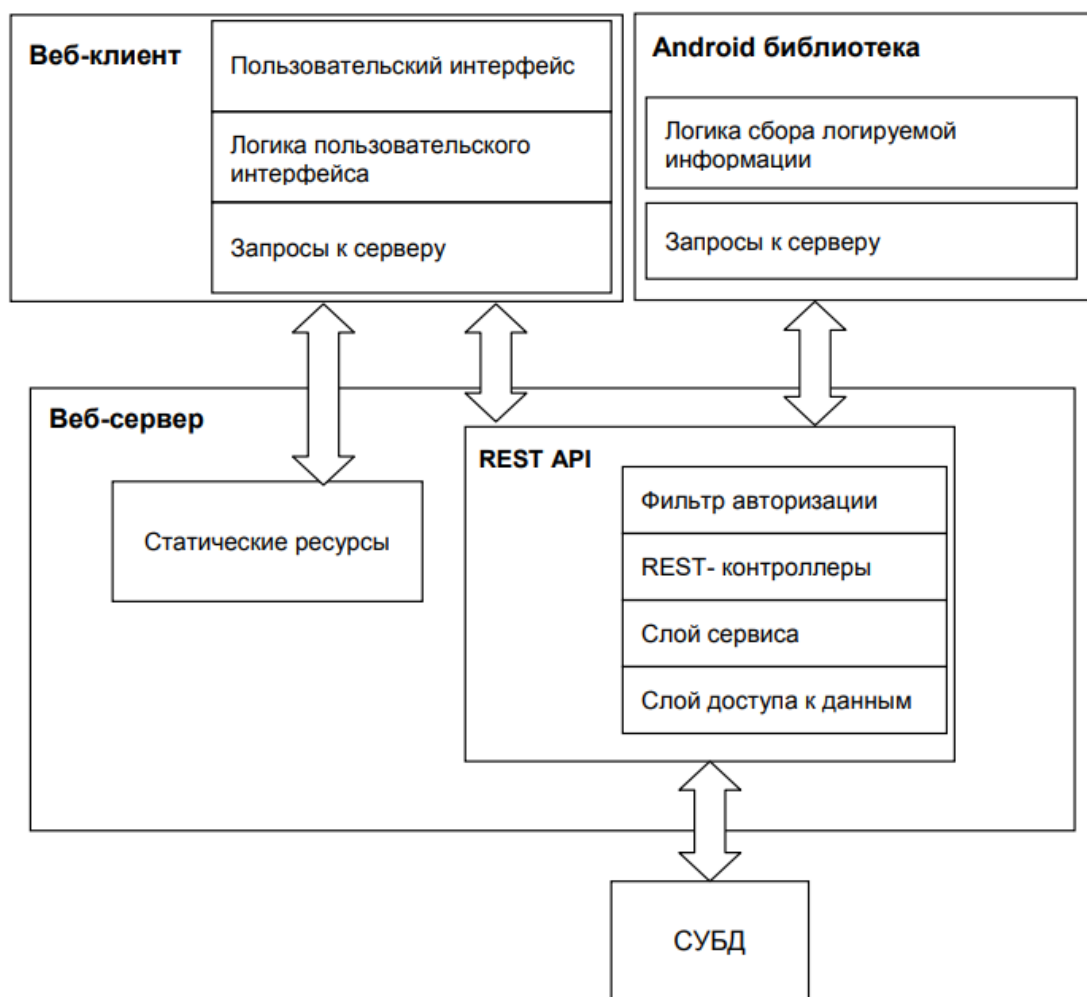


Рис.1. Архитектура АС

Предложенная архитектура АС позволяет написать клиента под любую платформу (мобильную или стационарную) и под любую операционную систему. Для обеспечения гибкой архитектуры АС в качестве клиента был выбран браузер, обладающий следующими достоинствами: есть реализации под все платформы и операционные системы; в настоящее время браузер установлен на всех устройствах с выходом в сеть; все пользователи сети Интернет постоянно пользуются браузером (не нужно тратить время на обучение); при каждом обновлении страницы пользователь получает последнюю версию приложения и не может использовать устаревшую версию; универсальность отображения содержимого для различных устройств при использовании адаптивного дизайна. В качестве языка приложения для android библиотеки используется статически типизированный язык программирования Kotlin, работающий под платформу JVM и полностью совместимый с языком Java, что позволяет разработчикам постепенно перейти с Java на Kotlin.

Разработаны алгоритмы регистрации новых пользователей, создания, редактирования и удаления новых проектов МП, сбора и просмотра списка ошибок в МП с информацией о пользователях, их устройствах и версиях МП,

создания и использования групп логируемых сообщений, предоставления широкого доступа к проекту МП. Предложенные алгоритмы реализованы с помощью разработанного ПС на языках: TypeScript (клиентская часть) и Java (серверная часть). Проведено тестирование ПС.

Заключение

В процессе выполнения данной работы разработана АС сбора и анализа логируемой информации, использование ПС которой позволяет обеспечить: применение современных информационных технологий и архитектурных решений; удобный и безопасный доступ к необходимой информации в зависимости от прав доступа; удобство использования мобильных устройств; возможность в дальнейшем разработки библиотек для других мобильных платформ; интуитивно понятный и удобный графический пользовательский интерфейс, не требующий специальной подготовки; высокая скорость работы; невысокая стоимость; интерфейс на русском языке.

Библиографический список

1. Wikipedia [Электронный ресурс]: – Режим доступа: <http://wikipedia.org>- Дата доступа 19.02.2018.
2. База данных PostgreSQL [Электронный ресурс]: – Режим доступа: <http://www.postgresql.org> - Дата доступа 19.02.2018.