

Г. Л. МУРАВЬЕВ, А. П. ЛИПОВЦЕВ, С. В. МУХОВ
БрГТУ (г. Брест, Беларусь)

ПРОЕКТНЫЕ РЕШЕНИЯ ДЛЯ СИСТЕМ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ С МОБИЛЬНЫМ ДОСТУПОМ

В работе анализируются особенности разработки программно-информационного обеспечения для организации коллективного использования ресурсов пользователями различных категорий в процессе решения общих задач. Это системы оперативного доступа к общим данным, справочные системы, системы экстренных служб. Системы для оперативной координации действий персонала учреждения, например, торгового предприятия. Система для оперативного информирования пользователей о состоянии учебного процесса, текущей, итоговой успеваемости студентов. Система, обеспечивающая взаимодействие владельцев транспортных средств, персонала технического обслуживания и т. д.

Их отличительные черты: необходимость поддержки эффективного взаимодействия пользователей с разными правами доступа; обеспечение доступа через типовые мобильные устройства, традиционные средства интернета; относительно невысокая квалификация пользователей; стохастичность запросов и характера выполняемых функций; относительная алгоритмическая простота, наличие типовых решений, ограниченность типов запросов; использование интернета, «облачных» ресурсов; функционирование в рамках ограниченных ресурсов используемых средств доступа; использование специфических средств (навигация, карты и т. д.); функционирование в реальном масштабе времени для работы с оперативной информацией, в офф-лайн режиме с медленно обновляющимися данными, учет приоритетов запросов и др.

При построении таких систем целесообразно максимально использовать типовые технологии, решения, библиотеки. Соответственно задачи разработки сводятся к выбору: структуры системы, модели логического представления, отвечающих характеру использования системы и требованиям к ней; форматов, механизмов передачи данных; способов диспетчеризации вычислительной нагрузки; механизмов обеспечения безопасности, надежности функционирования.

Особенности разработки показаны на примере системы поддержки процессов по эксплуатации индивидуальных транспортных средств, прикрепленных к сервисным центрам. Категории пользователей – владельцы транспортного средства, сотрудники сервисных центров.

Система обеспечивает: ведение базы данных о транспортном средстве, его текущем состоянии, о сервисных центрах, расходах на эксплуатацию и т. д.; поддержку комплекса задач по обслуживанию; задачи эксплуатации (навигация, работа с картами, маршрутами, сервисами, диагностическими сообщениями); задачи по планированию и учету расходов; генерацию отчетов.

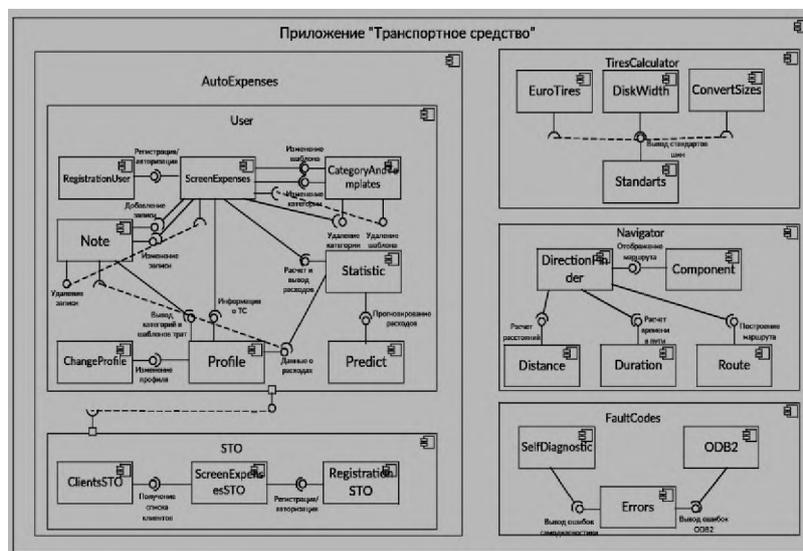


Рисунок 1. – Диаграмма компонентов

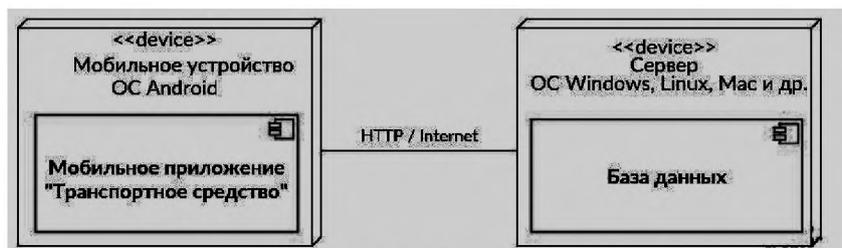


Рисунок 2. – Диаграмма развертывания

Система спроектирована как веб-сервис на основе модели MVC [1] с естественным представлением в виде относительно «автономных» компонентов – данных, отображений данных, процессов их изменения в результате взаимодействия пользователей. Для поддержки качества коллективного доступа, производительности, а также для обеспечения масштабируемости системы в серверной части использован интерфейс RESTful API (фреймворк Yii). Клиентское ПО размещается на мобильных устройствах. Хранение общих данных организовано с помощью хостинга Firebase Hosting [2]. Аутентификация пользователей реализована с помощью сервиса Firebase Simple Login.

Для поддержки интерактивности пользовательских интерфейсов мобильных устройств организованы потоки, выполняемые в фоновом режиме (AsyncTask). Данные принимаются посредством «загрузки» в несколько потоков в фоновом режиме, «отрисовка» интерфейса на время получения данных приложением не блокируется. При отсутствии подключения к сети данные кэшируются, а при подключении – синхронизируются с сервером в фоновом режиме, что повышает «реактивность» системы. Для работы с информацией применен формат JavaScript Object Notation, базирующийся на использовании структур данных, поддерживаемых большинством языков программирования. Это обеспечивает безизбыточность описаний, простоту использования. Для реализации клиентской составляющей использована среда разработки Android Studio [3, 4], в качестве языка программирования – Java, в качестве средства разметки – XML. Решения документированы диаграммами UML (на рисунках 1, 2 приведены упрощенные диаграммы компонентов и развертывания).

Клиентское приложение включает подсистемы ведения расходов AutoExpenses (модули авторизации RegistrationUser, визуализации ScreenExpenses, редактирования записей и категорий трат Note, CategoryAndTemplates, статистики расходов Statistics, включая прогнозируемые в Predict), данные автосервисов STO, кодов неисправностей FaultCodes, подсистему навигацию Navigator (построение и отображение маршрутов Route, DirectionFinder с расчетом параметров Duration, Distance [5, 6]), расчет и отображение параметров шин TiresCalculator и др.

Приложение устанавливается на мобильном устройстве с ОС Android 4.1 и выше, требует около 8 Мб памяти, что является не критичным даже для высоконагруженных интерактивных пользовательских интерфейсов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архитектура MVC [Электронный ресурс]. – 2011. – Режим доступа: <http://en.wikipedia.org/wiki/Model-view-controller>.
2. Документация Firebase [Электронный ресурс]. – 2017. – Режим доступа: <https://firebase.google.com>.
3. Основы Android [Электронный ресурс]. – 2015. – Режим доступа: <http://developer.android.com>.
4. Android Arsenal. Free libs and tools [Электронный ресурс]. – 2017. – Режим доступа: <https://android-arsenal.com/free>.
5. Руководство Google Maps API [Электронный ресурс]. – 2017. – Режим доступа: <https://developer.google.com/maps/documentation>.
6. Справочник Google Maps API [Электронный ресурс]. – 2017. – Режим доступа: <https://developers.google.com/maps/documentation/android-api>.