

Для повышения эффективности создания мобильных приложений разработано программное средство, использование которого позволяет создавать неограниченное количество мобильных приложений, настраивать гибкую систему прав доступа к получению определённых данных из мобильного приложения, получать информацию об ошибках, действиях пользователей и конфигураций мобильных устройств.

Разработанное программное средство состоит из нескольких компонентов: веб-сервер, веб-клиент и Android-библиотека, подключаемая к проекту мобильного приложения.

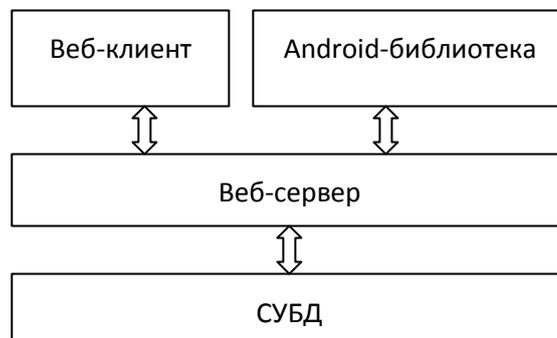


Рисунок 1 – Архитектура программного средства

Взаимодействие между компонентами программного средства осуществляется по протоколу HTTPS с использованием сообщений в формате JSON.

Веб-сервер программного средства реализован с использованием архитектурного стиля REST, предоставляющего ряд согласованных ограничений, учитываемых при проектировании распределённого программного средства [1] и реализован под платформу JVM с использованием языка программирования Java. Данный компонент программного средства отвечает за обработку информации, полученной из Android-библиотеки, и предоставление данных для веб-клиента. Сервер хранит всю информацию в базе данных.

В качестве СУБД для данного программного средства была выбрана PostgreSQL. Использование данной СУБД обусловлено её свободной доступностью и наличием возможности по хранению данных в объектно-реляционном представлении и в формате JSON с использованием встроенного типа данных. Встроенный тип данных JSON хорошо подходит для хранения сложной структуры ошибок мобильных приложений, а также обладает рядом возможностей по оперированию с данными структурами на уровне базы данных [2].

Android-библиотека может быть подключена к любому разрабатываемому Android-приложению. Данный компонент имеет несколько публичных классов, которые необходимы для настройки передаваемой информации и инициализации библиотеки. Основная функциональная задача Android-библиотеки состоит в обнаружении ошибок, ведущих к аварийному выходу приложения, сбору действий пользователей и данных по конфигурации используемого устройства, а также отправке всей полученной информации на веб-сервер.

Веб-клиент состоит из логики навигации и отправки запросов к серверному приложению. Веб-клиент предоставляет следующие возможности:

- регистрация и аутентификация пользователя;
- управление мобильными приложениями, доступными пользователю;
- управление правами доступа к мобильному приложению;
- получение информации об ошибках в мобильном приложении и управление ими;
- получение информации о пользователях мобильного приложения и их устройствах;
- получение информации о поведении пользователей мобильного приложения.

Список использованных источников:

1. Wikipedia [Электронный ресурс]. – Режим доступа : <http://wikipedia.org>. Дата доступа: 27.04.2017.
2. База данных PostgreSQL [Электронный ресурс]. – Режим доступа: <http://www.postgresql.org>. Дата доступа: 27.04.2017.

## ПРОГРАММНОЕ СРЕДСТВО ДЛЯ БЫСТРОГО СОЗДАНИЯ CMS СИСТЕМ

*Институт информационных технологий БГУИР, г. Минск, Республика Беларусь*

*Карнющенко В.В.*

*Шелягович А.С. – магистр техн. наук, ассистент*

Современное создание сайтов – это широкое применение веб-технологий, призванных значительно повысить их привлекательность в глазах потенциальных клиентов. Одним из компонентов веб-технологий и является фреймворк(программное средство).

Фреймворк – это программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Эта платформа подходит для создания сайтов, бизнес-приложений и веб-сервисов. Фреймворк диктует правила построения архитектуры приложения, задавая на начальном этапе разработки поведение по умолчанию, формируя каркас, который нужно будет расширять и изменять согласно указанным требованиям.

Большинство современных фреймворков построено на концепции MVC (Model–View–Controller, «Модель–Представление–Контроллер») — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо. Принцип концепции MVC представлен на рисунке 1:



Рисунок 1 – Основная концепция MVC фреймворков

На схеме видна основная цель применения этой концепции: в отделении бизнес-логики (*модель*) от её визуализации (*представление, вид*), что упрощает повторное использование кода и особенно полезно, если пользователь должен видеть одни и те же данные в различных контекстах и (или) с различных точек зрения одновременно. Выполняется следующее:

Основные преимущества фреймворка:

- Разработка на фреймворке (в отличие от самописных решений) позволяет добиться простоты сопровождения проекта.

- Решения на фреймворках, как правило, работают значительно быстрее и выдерживают большую нагрузку, чем CMS и самописные системы. Именно поэтому много популярных интернет-магазинов работают не на коробочных CMS, а на фреймворках. По уровню безопасности решения на фреймворках значительно превосходят самописные системы и сравнимы с CMS (как правило, сайты на фреймворках даже безопаснее).

Основной недостаток фреймворков - сроки разработки типового функционала на фреймворках больше, чем при использовании CMS. Фреймворки содержат только базовые компоненты уровня приложения, поэтому многие функции реализовываются индивидуально. Кроме того, для разработки на фреймворке требуется понимание бизнес-процессов, которые требуется реализовать.

Список использованных источников:

1. Пьюривал, С. Основы разработки веб-приложений/ С.Пьюривал. - СПб: Питер, 2015 – 272 с.

## УСТРОЙСТВО КОНТРОЛЯ ЧАСТОТЫ

Институт информационных технологий БГУИР, г.Минск, Республика Беларусь

Ковалевский В.С.

Скудняков Ю.А. – канд. техн. наук, доцент

В некоторых случаях для контроля частоты сигналов может оказаться недостаточно возможностей цифрового частотомера, например, когда частота сигнала тем или иным образом быстро изменяется. В таком случае будет удобнее воспользоваться звуковым индикатором частоты сигнала.

Для решения вышесформулированной задачи используется устройство, представляющее собой совмещенный с цифровым делителем частоты активный щуп с высоким входным сопротивлением и малой входной емкостью. Его особенно удобно использовать, например, когда частота контролируемых сигналов изменяется хаотичным образом или по заданному алгоритму, а также в случае сильной зашумленности или при пульсирующем характере контролируемого сигнала. При таких входных сигналах показания на дисплее цифрового частотомера «скачут», их трудно интерпретировать, а показываемые значения могут даже приблизительно не отражать реальной ситуации. Узел входного щупа собран на полевом транзисторе.

На рисунке 1 представлена принципиальная электрическая устройства, на которой узел входного щупа обведен пунктирной линией. Входной контролируемый сигнал через защитные резисторы R1, R2, R3 и конденсаторы C1, C2, C3 поступает на затвор маломощного полевого транзистора VT1. Встречно-параллельно включенные кремниевые диоды VD1-V D4 защищают полевой транзистор от пробоя высоким