**Тема 4. Разработка мобильных web-приложений**

**Содержание лекции** – Разработка мобильных web-приложений: особенности мобильных устройств, адаптивный дизайн, инструменты для разработки мобильных приложений и техники оптимизации для мобильных платформ.

**Рекомендуемая литература**

1. Соколова В. В. Разработка мобильных приложений : учебное пособие / В. В. Соколова. - Москва: Издательство Юрайт, 2024. - 160 с. ISBN 978-5-534-16868-6. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/542342> (дата обращения: 03.04.2024).
2. Д. Гриффитс. Head First. Программирование для Android на Kotlin. – Спб: Питер, 2023. – 912 с.

**Основные понятия и термины**

В настоящее время, всё больше и больше людей используют электронные устройства для упрощения своей жизни. Использование громоздких ПК плавно перетекло в ноутбуки, а теперь и в смартфоны. Крупные компании, как Apple, Dell и Microsoft хотят получить часть рынка смартфонов: ожидается, что к концу 2011 года, мировые продажи смартфонов побьют мировые продажи ПК.



Именно поэтому мобильная веб разработка очень актуальна сейчас. Продажи смартфонов растут каждый день и это говорит о необходимости повышенного внимания к интерфейсам для мобильных устройств. Как удовлетворить все запросы от пользователей, которые становятся все более и более требовательными? Как сделать так, чтобы люди могли смотреть сайт, как на Full HD монитор, так и на мобильном телефоне?

У веб-разработчиков всегда были проблемы с кросс-браузерностью. Теперь у них есть еще одна головная боль: кросс-платформенность, потому что Android приложение никогда не будет работать на iPhone и наоборот.
Основная проблема заключается в том, что все больше и больше людей используют и хотят использовать интернет на своих мобильных устройствах, но не все сайты готовы предоставить эту услугу.

**Обзор мобильного веба**

Прежде всего, важно понять, почему людям на самом деле нужна мобильная версия сайта. Может быть, кто-то скажет, что слишком маленький процент пользователей, посещают веб-сайт с помощью мобильного телефона, чтобы думать над этим. На самом деле, он не прав. Согласно статистике Facebook «более чем 250 миллионов активных пользователей в настоящее время заходят на Facebook через мобильные устройства. » [1] Facebook насчитывает более 500 миллионов активных пользователей, так что 50% из них являются мобильными пользователями. 75% пользователей YouTube сообщают, что мобильный является их основным способом доступа к контенту.

 Почему так? Потому, что люди не всегда могут носить с собой компьютер или ноутбук, а мобильный телефон всегда рядом.

Самый простой способ узнать процент посетителей, использующих мобильные девайсы, это запустить Google Anlytics. Не забудьте обратить внимание на типы устройств и операционных систем, которые используются вашими пользователями. Попытайтесь понять, почему пользователи посещают сайт, используя мобильное устройство и какие функции необходимы им.



В настоящее время многие разработчики после создания веб-сайта тестируют его в различных браузерах (IE, Firefox, Chrome, Opera и Safari) и не думают о том, как их создание будет смотреться на мобильном устройстве. Компании должны правильно оценивать, то как быстро смартфоны и другие мобильные устройства меняют способ доступа клиентов в Интернет, как делает BBC.

**Варианты создания**

Есть несколько возможностей создать приложение для сматрфонов. Первый способ: нанять Objective-C программиста, который пишет код для iPhone и iPad, Java-программиста, который пишет для Android, BlackBerry и. NET программиста, который пишет для Windows Phone 7.

Как видите это не самое лёгкое решение, поэтому я предлагаю ещё одно: веб-приложение.

Вот основные преимуществ для создания веб-приложения, а не native:
• Кросс-платформенность.
• Гораздо проще найти HTML программиста, чем эксперта в Objective-C или. NET.
• Гораздо проще переделать один веб-сайт, чем изменить дизайн для каждого native приложения.
• Вы можете обновлять свой сайт в любое время.
• Пользователи предпочитают браузеры, а не собственные приложения


Часто компании начинают задумывать о мобильной версии с того, что «нужно iPhone или Android приложение» вместо того, чтобы разобраться, что их пользователи используют не только эти платформы.
Есть много приложений для сматрфонов с операционной системой Android или iOS, но а как насчёт [Windows Phone 7, Blackberry и Symbian](http://www.gartner.com/it/page.jsp?id=1622614)? Кто хочет учить ещё одну SDK, кто хочет учить ещё один язык? Кто будет обновлять код на всех платформах при малейших изменениях?

Стоит разрабатывать native приложение только в том случае, если вам нужен доступ к камере, акселометру и т.д. Да и то, это [до поры до времени](http://my.opera.com/core/blog/2011/03/23/webcam-orientation-preview).

Существует только один способ поддержки всех платформ: использование веб-стандартов, и сейчас этот стандарт HTML5.

**Проблемы мобильной разработки**

В дополнении к проблеме кросс-платформенности о которой я писал выше, на любом девайсе может быть разные веб-браузеры. Например, пользователь с Android телефоном может зайти ваш сайт используя, как браузер по умолчанию, так и Opera Mini или Firefox Mobile.

Общие проблемы для разработчиков под мобильные девайсы:
• Размер экрана. Девасов много и у всех разный размер, начиная с 240 x 320 и заканчивая 800 x 480. Это значит, что разработчик не может делать дизайн фиксированой ширины, он должен меняться в зависмости от устройства.
• Ориентация. Мало того эти 240 x 320 легко могут превратиться в 320 x 240.
• Touchable. Должно быть достаточно места между разными кликабельными элементами.

**Рекомендации по созданию мобильных приложений**

Основная цель разработки для мобильных устройств заключается в изменении макета для меньшего дисплея.
Мобильный веб-страницы можно просматривать в двух режимах: портрет и пейзаж. Используя JavaScript вы можете получить доступ к *window.orientation*:
• 0 — портретная ориентацию
• -90 И 90 — альбомная ориентация.

Вы можете также использовать событие *orientationchange* для выполнения какие-либо действия, когда пользователь переворачивает устройство. Этот пример просто устанавливает атрибут класса при изменении текущей ориентации:

window.onorientationchange = function() {

 var orientation = window.orientation;

 switch(orientation) {

 case 0:

 document.body.setAttribute("class","portrait");

 break;

 case 90:

 document.body.setAttribute("class","landscape");

 break;

 case -90:

 document.body.setAttribute("class","landscape");

 break;

 }

 }

После загрузки страницы на мобильном устройстве URL строка ввода исчезает только при прокрутке.

Но можно это сделать и программно:

window.addEventListener ('load', функция () {

 setTimeout (ScrollTo, 0, 0, 1);

}, False);

Оптимизация для мобильного сайта означает, что вы должны уменьшить количество запросов к серверу:
• Используйте один CSS.
• Используйте CSS Sprites.
• Используйте HTML5 local storage для кэширования данных.

**Советы по дизайну**

• Не пытайтесь заполнить все пустое пространство на экране. Все пользователи имеют разные скорости соединения, поэтому сайт не должен «весить» слишком много. Чрезмерное количество изображений, текста, кода и т.д. не только ухудшает восприятие пользователями сайта, но и значительно увеличивает время загрузки.

• Даже на маленьком экране, пользователь не должен иметь проблем с чтением текста.

• Около интерактивных элементов сайта должно быть достаточно пространства, чтобы нажать на них. Всё, чего пользователь касается должна быть не менее размера среднего пальца.

• Показывайте прогресс загрузки с помощью анимации.

**Тестирование мобильных приложений**

Существует очень широкий спектр мобильных устройств, но, к сожалению, веб-разработчики не могут так легко проверить свои веб-приложения, как они делают это на PC. С одной стороны, тестирование должно проходить на реальных устройствах, но покупка устройства для каждой платформы глупо. Для большинства платформ есть эмуляторы устройств, некоторые из них даже онлайн.
 **Создание собственных мобильных приложений**

Фреймворков для мобильный приложений достаточно: jQTouch, iWebKit, Wink Toolkit, jQuery Mobile и Sencha Touch. Три первых не очень хороши — они поддерживают только одну или две платформы. Два оставшихся — это то из чего стоит выбирать.