

объединение нескольких таблиц в одну, что позволяет сократить количество операций объединения [1];

- в запросе происходят сложные вычисления, возникающие, например, при использовании групповых и агрегатных функций. Чтобы избежать выполнения таких вычислений каждый раз при извлечении данных, применяется предварительное вычисление необходимых значений во время операции вставки новой записи в таблицу с последующим сохранением полученного значения в отдельной колонке, принадлежащей этой же таблице. Такой подход демонстрирует высокую производительность для данных, которые необходимо часто извлекать, но которые редко изменяются [2].

Основными недостатками денормализации является появление избыточной информации. Также ее применение негативно сказывается для данных, которые часто обновляются. Кроме того, нельзя недооценивать тот факт, что денормализация ухудшает понимание модели данных и ее дальнейшее сопровождение.

Денормализация является очень действенным способом для улучшения производительности обработки данных. Она часто находит свое практическое применение в различных моделях данных. Но у нее есть свои недостатки. Прежде чем применять денормализацию, для начала нужно попробовать другие способы оптимизации производительности. И в том случае, когда они не приносят должного результата, стоит обратиться к денормализации.

Список использованных источников:

1. Grant Fritchey, *Sql Server 2012 Query Performance Tuning* – 470с.
2. <https://habrahabr.ru/post/64524/> - электронный ресурс

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Хильчук А.С.

Куликов С.С. – к.т.н., доцент

На данный момент во многих сферах общества используется множество программных продуктов. Также наблюдается тенденция к использованию мобильных приложений и отход от веб-вариантов приложений. Поэтому требования к стабильности и удобству использования этих приложений становятся выше.

Тестирование ПО – это процесс исследования, испытания программного продукта, по результатам которого можно выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации. [1]

Для снижения затрат на проведение ручного тестирования внедряется её автоматизация. Автоматизированное тестирование ПО – это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически, что помогает сократить время тестирования и упростить его процесс. [2]

Одной из главных задач при внедрении автоматизации тестирования на проекте по разработке мобильных приложений является выбор фреймворка для разработки тестов. Самым популярным из них – Appium [3], который является платформой для кроссплатформенной автоматизации тестирования с открытым исходным кодом. С помощью данного программного решения можно выполнять тестирование нативных, гибридных и мобильных веб-приложений на платформе iOS и Android. Управление элементами приложения и мобильным устройством осуществляется с использованием WebDriverJSONWireProtocol. Данный фреймворк является инструментом для тестирования по методу чёрного ящика, следовательно, нет необходимости в рекомпиляции приложения, а также в доступе к исходному коду. Данный фреймворк доступен для языков программирования Java, .Net (C#), Python, Ruby, JavaScript и PHP. К минусам можно отнести невозможность параллельного запуска тестов для платформы iOS, ограниченную поддержку жестов, сложность в настройке, поддержку версии Android только 4.1 и выше и достаточно низкую скорость работы.

К популярным фреймворкам по разработке тестов для мобильных приложений можно отнести Robotium [4]. Данная платформа с открытым исходным кодом позволяет тестировать нативные и гибридные Android-приложения по методу чёрного ящика, а значит для запуска теста, как и с Appium, от разработчика потребуется только исполняемый арк-файл. Но стоит учесть, что данная платформа подойдёт только для приложений, где переход к следующей итерации разработки не влечёт за собой сильного изменения пользовательского интерфейса, в противном случае разработчику тестов будет необходимо переписать вплоть до 90% кода тестов. Так же к минусам можно отнести поддержку только языка Java в качестве языка программирования тестов, а также поддержку только Android-приложений.

К группе часто используемых платформ для разработки автоматизированных тестов под мобильные приложения также относится Espresso от компании Google [5]. Основной API данного решения невелик, но так как это проект с открытым исходным кодом, код может быть расширен разработчиком тестов. Отличительной особенностью данной платформы является синхронизация с главным UI-потокот Android-приложения. Большинство фреймворков запускают тесты в отдельном потоке, вследствие чего возникают проблемы со скоростью и могут возникнуть критические ошибки в самих тестах вследствие обновления пользовательского интерфейса приложения. Большинство разработчиков игнорируют этот факт и ставят задержки в потоке

(sleep), что является достаточно грубым нарушением принципов построения логики работы тестов. С Espresso разработчику тестов не придётся осуществлять методы синхронизации, так как фреймворк осуществляет её сам посредством интеграции с `MainUIThread`.

Для написания автоматизированных тестов для iOS-приложений также используют платформу XCUITest [6], которая разрабатывается компанией Apple и поставляется вместе со средой разработки XCode. Также её можно использовать для проведения юнит-тестов и тестов производительности iOS-приложений. Как и Espresso, данная платформа потребует доступ к исходному коду тестируемого приложения. По этой причине, разработчик тестов имеет возможность написания тестов только на языках программирования Swift или Objective-C и поэтому больше используется разработчиками приложения, нежели сотрудниками отдела тестирования. Так как это не кроссплатформенное решение, скорость выполнения тестов в значительной степени выше по сравнению с Appium. К дополнительным плюсам можно отнести низкую степень появления ложного результата теста. Недостатком является сложность связки с системами непрерывной интеграции для автоматического запуска.

Таким образом, для мобильных приложений, будь это приложение интернет-банкинга, интернет-магазина или новостного портала, очень важна правильность работы, устойчивость к некорректным действиям пользователя и другим негативным факторам. Автоматизация тестирования в значительной степени ускоряет процесс проведения тестов, которые позволяют удостовериться в том, что приложение соответствует установленному техническому заданию, за достаточно короткий промежуток времени, выполняя проверки на различных мобильных окружениях и их конфигурациях.

Список использованных источников:

1. Гленфорд Майерс, Том Баджетт, Кори Сандлер. Искусство тестирования программ, 3-е издание (The Art of Software Testing, 3rd Edition.) — М.: «Диалектика», 2012. — 272 с.
2. Про Тестинг [Электронный ресурс] // protesting.ru: Тестирование Программного Обеспечения. URL: <http://www.protesting.ru/automation> (дата обращения: 12.03.2018).
3. Appium [Электронный ресурс] // appium.io: Appium – AutomationforApps. URL: <http://www.appium.io> (дата обращения: 12.03.2018).
4. Robotium [Электронный ресурс] // [robotium.com](https://github.com/RobotiumTech/robotium): UserscenariotestingforAndroid. URL: <https://github.com/RobotiumTech/robotium> (дата обращения: 13.03.2018).
5. Espresso [Электронный ресурс] // [developer.android.com](https://developer.android.com/training/testing/espresso/index.html): Espresso. URL: <https://developer.android.com/training/testing/espresso/index.html> (дата обращения: 13.03.2018).
6. XCUITest [Электронный ресурс] // [developer.apple.com](https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/09-ui_testing.html): Testing with Xcode. URL: https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/09-ui_testing.html (дата обращения: 13.03.2018).

ВЕБ-ПРИЛОЖЕНИЕ ОКАЗАНИЯ УСЛУГ СТРАХОВОЙ КОМПАНИИ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Чернышев Н.С.

Фадеева Е.Е. – ассистент

В последнее время веб-технологии активно развиваются и набирают популярность. На сегодняшний день сложно представить нашу жизнь без компьютера, телефона или сети Интернет.

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, а обмен информацией происходит по сети.

Одним из преимуществ подхода с использованием веб-приложения является тот факт, что клиенты не зависят от конкретной операционной системы, поэтому веб-приложения являются кроссплатформенными средствами. Вместо того, чтобы писать различные версии веб-приложения для Microsoft Windows, MacOS, Unix и других систем, приложение создается один раз и может быть развернуто на любой системе. В связи с этим, отсутствует необходимость установки специализированных программных средств и разработки множества вариантов одного приложения для каждой платформы на мобильные устройства и персональные компьютеры, так как любой современный браузер может быть использован для работы с веб-приложением, что уменьшает затраты времени и средств на разработку этого приложения.

Также веб-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере.

Страхование — одна из важнейших категорий общественных отношений. В широком смысле оно включает различные виды страховой деятельности, которые в комплексе обеспечивают страховую защиту.

С древних времён объективная потребность в страховании обуславливается тем, что убытки подчас возникают вследствие разрушительных факторов, не подконтрольных человеку, таких как стихийные бедствия или различного рода чрезвычайные ситуации. В подобной ситуации невозможно взыскивать убытки с кого-либо и заранее созданный страховой фонд может быть источником возмещения полученного ущерба.

В общем и целом, страхование — система страховой защиты от возможного наступления различного рода рисков. Оно представляет собой способ возмещения страховщиком убытков пострадавшим путем их