Interparts.by.Сайт по своей структуре похож на Exist, схема поиска, выбора и покупки запчастей аналогичная. Цена на некоторые позиции заметно ниже, чем у Exist.

Таким образом, данная область интернет-торговли является актуальной и весьма прибыльной на сегодняшний день. Анализ позволяет сделать вывод, что лучшим интернет-магазином в г.Гомеле является магазин Exist.by: здесь хорошо налажена поставка запчастей в кратчайшие сроки, имеется квалифицированный персонал, который поможет с выбором автозапчасти за приемлемые деньги. В офисе компании, куда поставляются заказы, есть клиентский терминал (клиентский ПК) для автономной работы покупателя на сайте.

Представленная мною модель отражает налаженную структуру интернет сайта, готовую заниматься интернет торговлей (рисунок 1).

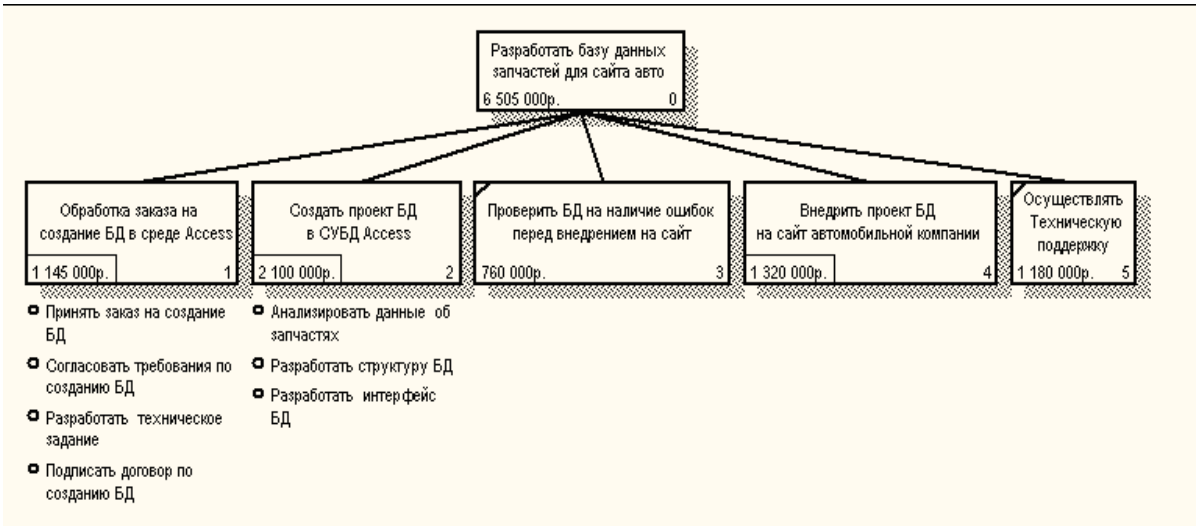

Рисунок 1– Дерево сайта интернет магазина

В качестве основы для разработанного мной интернет сайта взята структура сайта Existнаписанный HTML-кодом, размещенном на сервере Server: Microsoft-IIS/6.0. Такой сервер может обслуживать сразу несколько сайтов.

А.В. Сапанович (УО «ГГТУ им. П.О. Сухого», Гомель)
Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПЛАТЕЖНЫХ ПРИЛОЖЕНИЙ

На текущий момент электронная коммерция стала неотъемлемой частью мировой финансовой системы. Среди наиболее трудоемких операций при тестировании данных систем выделяется тестирование

мобильных приложений. Если учитывать, что одновременно сопровождается несколько мобильных платформ, то в этом направлении в первую очередь необходима автоматизация. На текущий момент среди уже готовых решений есть два направления: утилиты, интегрированные в среды разработки, и удаленные сервисы тестирования, например, TestCloud, производства компании Xamarin [1]. Вариант с встроенным ПО является самым худшим, потому что разные компании используют различные среды разработки, а значит, ограничивается возможность распространения системы тестирования. Использование удаленных сервисов тоже вызывает определенные неудобства: тестовый сценарий сервис генерирует сам, либо его нужно заранее описать на понятном сервису языке, что влечет за собой дополнительные затраты по времени [2]. Также подобные сервисы не проводят все тесты на реальных устройствах, а часть ошибок проявляется именно на них. Самостоятельная разработка системы тестирования позволяет избежать описанных недостатков: написание сценария можно вести на привычном языке программирования, нет необходимости в жесткой связи со средой разработки, а также можно гарантированно использовать набор реальных устройств.

Для пробного проекта наиболее удобной является платформа Android, поскольку она является достаточно открытой, и средства разработки для нее в основном являются бесплатными.

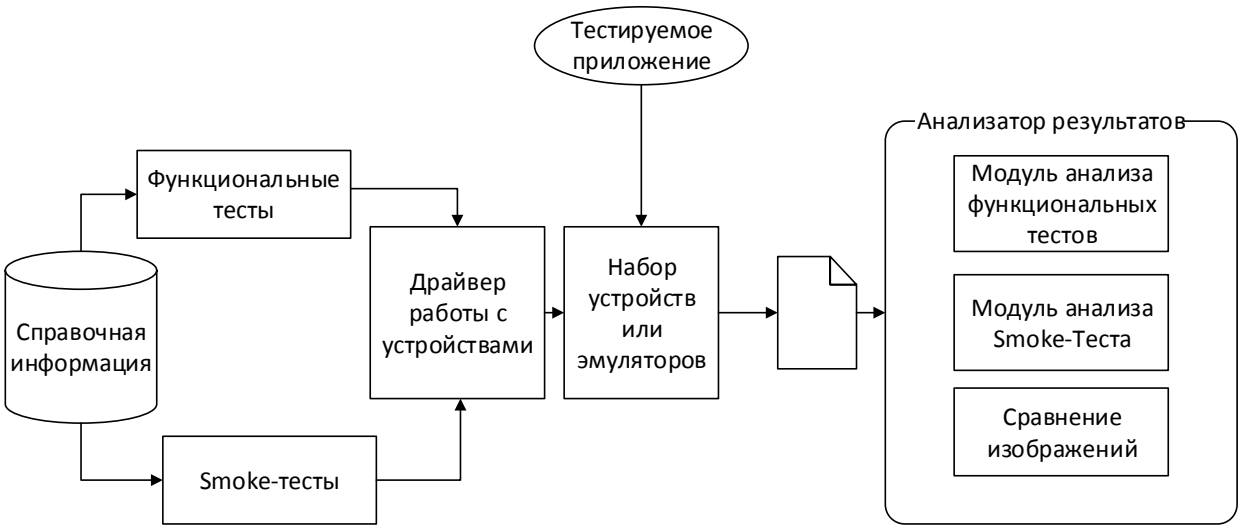

Рисунок 1 – Структурная схема системы тестирования

Тестовая структура была построена на основе фреймворка Robotium. Средство тестирования представляет собой структуру, представленную на рисунке 1. Вся справочная информация хранится в базе данных и конфигурационных файлах, поскольку компоненты могут быть

разнесены по различным компьютерам и обмениваться данными по сети. Функциональные и smoke-тесты основаны на различной логике и разнесены в два приложения. При запуске каждого из сценариев приложение собирается и при помощи драйверов загружается в тестовое устройство или эмулятор – на схеме они обозначены одним условным блоком, потому что в различных ситуациях используются различные наборы драйверов: androiddebugbridge, драйвера конкретных устройств. Также возможно использование промежуточного интерфейса Spoon, который позволяет запускать один и тот же сценарий одновременно на нескольких устройствах [3].

Функциональные тесты основаны на поиске текстов и элементов, содержащих эти тексты: например, ищутся все поля на итоговом чеке, содержащие данные о суммах. Сами тесты реализованы с учетом особенностей тестируемых приложений. Так, например, тесты «умеют» генерировать номера банковских карт, запрашивать у компонент внешних систем контрольные суммы по необходимым протоколам. Существующая реализация тестовых сценариев позволяет учитывать даже мелкие особенности такие, как частота появления определенных сообщений: например, всевозможных напоминаний. Использование удаленных сервисов для тестирования не дает такого полного охвата. Также учитывается и другая особенность проверяемых приложений: динамическая генерация идентификаторов меню. Автотесты полностью охватывают элементы меню, при этом отсутствует привязка к какому-то конкретному устройству. В процессе выполнения после каждого пункта сценария система перезапускает тестируемое приложение, с целью сброса кэша и т.д. Это делается для того, чтобы обеспечить возможность проведения тестирования на старых устройствах с малым объемом памяти.

Каждая ветка генерирует свой лог-файл, который перезаписывается при каждом запуске тестов. При формировании файлов используются различные подходы, основанные на логике тестов. Smoke-тест построен так, что если какая-либо операция не выполнилась, то зависящие от нее операции утилита даже не будет пытаться выполнить. В итоге в отчет будет добавлен перечень всех операций, которые не выполнились с указанием причины. В основных функциональных тестах такая логика изначально не была заложена, поэтому в отчет пишутся служебные сообщения, в которых указывается, какой этап начат и закончен при помощи специальных тегов, а также сообщения с ошибкой при обработке исключений. Утилита-анализатор отчетов ищет для каждого этапа начало и конец и, если тег конца не найден, выводит сообщение об ошибке и отмечает этой ошибкой данный этап.

Литература

1. Xamarintestcloud [Электронный ресурс] / Xamarin. – 2013. – Режим доступа: http://xamarin.com/test-cloud. – Дата доступа: 22.12.2013.
2. Тестирование мобильных приложений в облаке [Электронный ресурс] / Открытые системы. – 2013. – Режим доступа: http://www.osp.ru/os/2013/04/13035541/. – Дата доступа: 02.01.2014.
3. Distributing instrumentation tests to all your Androids [Электронный-ресурс] / Spoon. – 2013. – Режимдоступа: http://square.github.io/spoon/. – Датадоступа: 22.12.2013.

Д.Л. Семенякин (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.И. Кучеров,** ст. преподаватель

ВЫБОР ВИДЕОКАМЕР ДЛЯ ВИДЕОНАБЛЮДЕНИЯ

Правильный подбор и размещение видеокамер – одна из самых важных задач при проектировании системы видеонаблюдения на любом объекте. И успех в решении этой задачи напрямую зависит от понимания характеристик, возможностей и вариантов применения существующих на данный момент видеокамер. Именно этому и посвящена данная статья.

На рисунке 1 ниже отображена обобщенная функциональная схема видеокамеры:

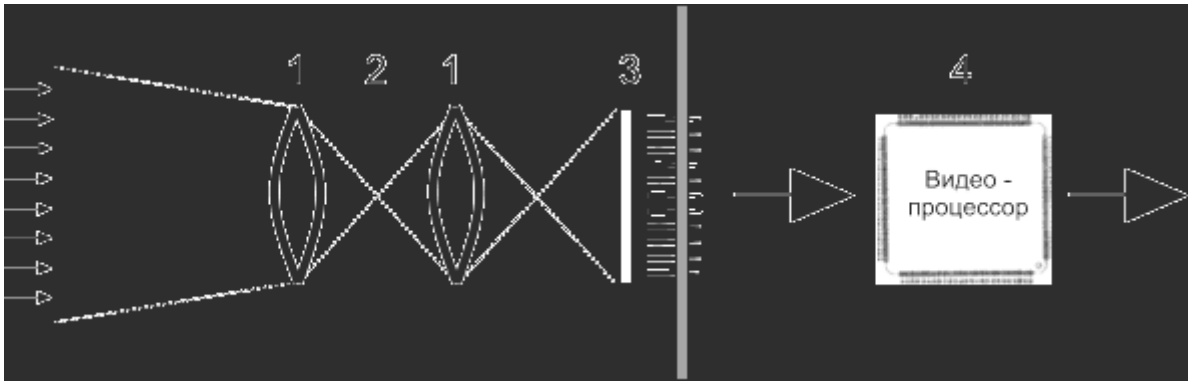

Рисунок 1 – Устройство видеокамеры

Как видим, оптическая часть видеокамеры функционально очень похожа на человеческий глаз: изображение проходит через систему линз (1), диафрагму (2) и проецируется на фотоприемную ПЗС-матрицу (3). Далее, в ПЗС-матрице (ПЗС – Прибор с Зарядовой Связью) изображение преобразуется в электрический сигнал, который поступает уже в электронную часть (4) видеокамеры и там подвергается обработке. Электроника видеокамеры может, в свою очередь, управлять