

7 Бизнес-план Акционерного Общества «Узбекистон Темир Йуллари» на 2019 год. – Ташкент, 2018.

8 Theeg, G. Railway Signalling & Interlocking / G. Theeg, S.Vlasenko ; DW Media Group GmbH | Eurailpress, Hamburg, 1st Edition, 2009. – 448 p.

9 Акты выполненных работ по х/д теме № 61. «Разработка технического и программного обеспечения системы контроля состояния станционных устройств железнодорожной автоматики и телемеханики», Ташкент. – 2016–2017.

УДК 625.8

## **РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОБИЛЬНОГО ТЕРМИНАЛА КОНТРОЛЯ ДОКУМЕНТОВ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ IOS**

*А. С. БАБАРИКИН*

*Белорусский государственный университет транспорта, г. Гомель*

Во всем мире используется 4 миллиарда смартфонов. К 2020 году этот показатель может повыситься до 6 миллиардов. Гаджеты прочно вошли в повседневность современного человека и не собираются сдавать своих прежних высот. Смартфоны стали неотъемлемой частью в автоматизации процессов. От регистрации на рейс самолета до сканирования электронного билета по штрих-коду смартфоны помогают нам оптимизировать многие процессы, на которые уходило достаточно много средств и усилий. Так, в свое время Белорусская железная дорога решилась на эксперимент в создании мобильного терминала для контроля документов (далее – МТКД) на платформе Android.

Целью создания ПО МТКД является автоматизация процесса получения информации о проездных документах, оформленных с использованием сети Интернет и билетной кассы. В ПО МТКД разработаны инструменты для проверки проездных документов, выполнения процедуры принудительной электронной регистрации (далее – ЭР) на борту поезда, сбора информации о посадке и высадке пассажиров и т. д.

Разработка ПО МТКД направлена:

- на совершенствование технологии предоставления услуг по перевозке пассажиров за счет снятия ограничения на приобретение проездных документов с ЭР по ходу следования поезда;
- автоматизацию процедуры выполнения электронной регистрации на борту поезда;
- сокращение трудозатрат по организации возврата денежных средств за неиспользованные электронные проездные документы (билеты), сданные в претензионном порядке;
- оперативное получение информации поездными бригадами о предстоящей посадке (высадке) пассажиров по ходу следования поезда.

ПО МТКД обеспечивает выполнение следующих функций:

- получение информации по «Рейсу»:
  - ведомость проездных документов (состоит из электронных билетов, включая черный список, оформленных в билетной кассе);
  - маршрут следования поезда;
  - схема состава поезда;
  - статус проездного документа;
  - уточнение названия станции;
- выполнение отметок для проездных документов;
- выполнение принудительной ЭР (за пассажира);
- автоматическое обновление ведомости проездных документов по ходу следования поезда;
- проверка наличия проездных документов в ведомости;
- ручной ввод информации о пассажире при посадке на место начальника поезда (далее – ЛНП);
- передача на сервер автоматизированной системы контроля посадки пассажиров (далее – АСКПП) информации по рейсу;
  - получение отчета по расходу постельных принадлежностей;
  - автоматическое и ручное обновление ПО МТКД, профиля настроек ПО МТКД;
  - срабатывание первого будильника для выдачи билетов пассажирам, срабатывание второго будильника перед прибытием на станцию;
  - просмотр информации о событиях работы ПО МТКД в журналах;
  - просмотр информации о расходе трафика, проверка баланса лицевого счета, проверка номера телефона SIM-карты;
  - сбор информации о посадке (высадке) пассажиров поезда;
  - закрытие рейса (удаление информации по рейсу).

На текущий момент поставлена задача разработать альтернативный графический интерфейс и реализацию основных функций мобильного терминала контроля документов на мобильных устройствах платформы iOS. Опираясь на статистику 2019 года, количество пользователей iOS достигло отметки в 30 процентов и является второй самой популярной мобильной платформой после Android. Следовательно, повышается актуальность реализации системы МТКД на мобильной платформе iOS. Одной из важных задач, которую требуется выполнить, является создание удобного и интуитивно понятного интерфейса, которого лишена текущая реализация на Android.

iOS (до 24 июня 2010 года – iPhone OS) – мобильная операционная система для смартфонов, электронных планшетов, носимых проигрывателей и некоторых других устройств, разрабатываемая и выпускаемая американской компанией Apple. Была выпущена в 2007 году; первоначально для iPhone и iPod touch, позже – для таких устройств, как iPad. В 2014 году появилась поддержка автомобильных мультимедийных систем Apple CarPlay. В отличие от Android (Google), выпускается только для устройств, производимых фирмой Apple.

В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. Ядро iOS почти идентично ядру настольной операционной системы Apple macOS (ранее называвшейся OS X). Начиная с самой первой версии, iOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM. iOS разработана на основе операционной системы OS X (с тех пор переименованной в macOS) и использует тот же набор основных компонентов Darwin, совместимый со стандартом POSIX.

Пользовательский интерфейс iOS основан на концепции прямого взаимодействия с использованием жестов «мультикас». Элементы управления интерфейсом состоят из ползунков, переключателей и кнопок.

Слои абстракции iOS:

- Core OS;
- Core Services;
- Media Layer;
- Cocoa Touch.

Ключевым моментом при старте разработки является выработка требований к архитектуре. Основными критериями при выборе архитектуры являются: распределение, тестируемость, простота использования. В качестве архитектуры приложения, которая удовлетворяет данным критериям, было решено выбрать Model-View-ViewMode (далее MVVM) (рисунок 1).

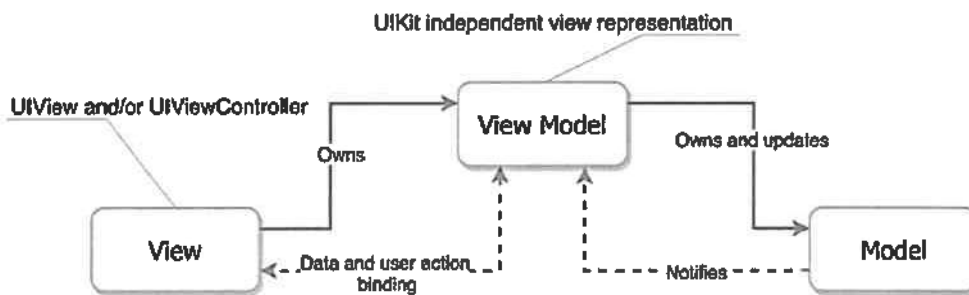


Рисунок 1 – Схема взаимосвязей модулей в MVVM

Model-View-ViewModel (MVVM) – шаблон проектирования архитектуры приложения. Представлен в 2005 году Джоном Госсманом (John Gossman) как модификация шаблона Presentation Model. Используется для разделения модели и ее представления, что необходимо для их изменения отдельно друг от друга. Например, разработчик задает логику работы с данными, а дизайнер работает с пользовательским интерфейсом. MVVM удобно использовать вместо классического MVC и ему подобным в тех случаях, когда в платформе, на которой ведется разработка, есть «связывание данных». В шаблонах проектирования MVC/MVP изменения в пользовательском интерфейсе не влияют непосредственно на Модель, а предварительно идут через Контроллер (англ. controller) или Presenter. В таких технологиях, как WPF и Silverlight, есть концепция «связывания данных», позволяющая связывать данные с визуальными элементами в обе стороны. Следовательно, при использовании этого приема применение модели MVC становится крайне неудобным из-за того, что привязка данных к представлению напрямую не укладывается в концепцию MVC/MVP.

Шаблон MVVM делится на три части:

Модель (англ. model) (так же, как в классической MVC) представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения.

Представление (англ. view) – графический интерфейс (окна, списки, кнопки и т. п.). Выступает подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью Представления. В случае, если в Модели Представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление в свою очередь запрашивает обновленное значение свойства из Модели Представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью Представления.

Модель Представления (англ. ViewModel) – с одной стороны, абстракция Представления, а с другой – обертка данных из Модели, подлежащие связыванию. То есть она содержит Модель, преобразованную к Представлению, а также команды, которыми может пользоваться Представление, чтобы влиять на Модель.

На данный момент производится выработка технического задания и реализация мобильного приложения в среде разработки Xcode. Также ведется интеграция облачной базы данных Firebase в приложение для имитации серверной части МТКД ввиду закрытого доступа к оригинальным серверам. Таким образом, разработка МТКД под операционную систему iOS расширит круг мобильных устройств, которые могут обеспечить функционирование МТКД, а также предложит новые решения, которые в последующем можно применить в версии под Android.

#### Список литературы

- 1 Neuburg, M. iOS 12 Programming Fundamentals with Swift / M. Neuburg. – O'REILLY, 2019.
- 2 Apple Developer Documentation [Электронный ресурс]. – Режим доступа : <https://developer.apple.com/documentation>. – Дата доступа : 17.09.2019.

УДК 621.38

## ВЛИЯНИЕ ФУНКЦИОНАЛЬНОЙ, ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И ПРЕДНАМЕРЕННОГО ЭЛЕКТРОМАГНИТНОГО ВОЗДЕЙСТВИЯ НА МИКРОЭЛЕКТРОННЫЕ СИСТЕМЫ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

*К. А. БОЧКОВ, Н. В. РЯЗАНЦЕВА, Д. В. КОМНАТНЫЙ*

*Белорусский государственный университет транспорта, г. Гомель*

На железнодорожном транспорте системы железнодорожной автоматики и телемеханики (СЖАТ) призваны в первую очередь обеспечить безопасность движения поездов. Повышенные требования по обеспечению безопасности движения поездов налагали и особые методы построения СЖАТ. Ранее СЖАТ строились на основе аппаратной реализации с использованием специальных реле первого класса надежности с несимметричными отказами. При этом не существовало проблем обеспечения информационной безопасности и доказательства функциональной безопасности и угроз преднамеренного электромагнитного воздействия на СЖАТ.

Современные СЖАТ строятся на основе аппаратно-программных комплексов (АПК) с использованием микроэлектронной элементной базы с симметричными отказами. Для СЖАТ принято различать согласно ГОСТ Р 53431–2009 два вида неработоспособного состояния: защитное и опасное. При этом в защитном состоянии все функции по обеспечению безопасности движения поездов соответствуют требованиям нормативно-технической документации (НТД). В опасном состоянии значение хотя бы одного параметра по обеспечению функций безопасности движения поездов не соответствуют требованиям НТД. В опасное состояние система переходит при наличии опасного отказа. Для возможности оценки наличия опасных отказов для каждой из СЖАТ или ее компонентов формулируются критерии опасных отказов в соответствующих НТД.

Следует отметить, что наличие опасного отказа еще не означает нарушения условий безопасности движения поездов, поскольку оно может возникнуть только при условии

$$Q_{\text{дп}} = Q_{\text{оо}} Q_{\text{пс}} Q_{\text{чс}},$$

где  $Q_{\text{дп}}$  – вероятность нарушения условий безопасности движения поездов (авария, крушение, гибель людей, огромные материальные потери и т. д.);  $Q_{\text{оо}}$  – вероятность опасного отказа;  $Q_{\text{пс}}$  – веро-