

АРХИТЕКТУРА МОБИЛЬНОГО КЛИЕНТА ПОД IOS ДЛЯ ДОСТУПА К ВЕБ-СЛОВАРЮ НАРОДНОЙ РЕЧИ СРЕДНЕГО ПРИИРТЫШЬЯ

И.А. Балезин

студент, e-mail: iabalezin@gmail.com

Д.Н. Лавров

к.т.н., доцент, e-mail: dmitry.lavrov72@gmail.com

М.А. Харламова

к.фил.н., доцент, e-mail: khr-spb@mail.ru

Омский государственный университет им. Ф.М. Достоевского

Аннотация. Представлена архитектура и разработано мобильное клиент-серверное приложение для доступа с iPhone к словарю констант народной речи Среднего Прииртышья с необходимым набором возможностей.

Ключевые слова: архитектура приложения, iOS, мобильное приложение, словарь народной речи.

1. Введение

Изучение национального мировидения и отражения его в различных формах народной речи — одна из важнейших задач и неотъемлемая часть лингвистической науки, поэтому создание региональных словарей, манифестирующих традиционную народную культуру, представляется безусловно актуальным.

Объектом описания нового словаря, над которым трудятся диалектологи и программисты ОмГУ им. Ф.М. Достоевского, является речь диалектоносителей полиэтнического региона Среднего Прииртышья (без дифференциации по этническим, социальным, образовательным и др. признакам). Предметом же становится описание культурно значимых констант (ключевых слов), объективирующих мировидение сельского жителя Омского Прииртышья и сохраняющих устойчивые элементы смысла на протяжении истории. Словарь имеет полевою структуру, что позволяет объёмно и полно реконструировать тот или иной фрагмент диалектной картины мира, см. подр. [1].

Электронный словарь выполняет, прежде всего, важную практическую функцию — позволяет быстро найти слово и его контекстуальное употребление. Эвристическая функция электронного словаря обеспечивает решение задач исследовательского характера, касающихся как семантики отдельного слова-репрезентанта константы, его синтагматических и парадигматических связей, так и способности слова к участию в устойчивых сочетаниях и в прецедентных высказываниях. История функционирования константы, представленная в словаре, репрезентирует своеобразие концептуализации мира носителями традици-

онной крестьянской культуры. Электронный формат словаря значительно расширяет пользовательскую аудиторию: доступен как специалистам-филологам, культурологам, историкам, этнографам, так и всем ценителям народного слова.

Мобильный клиент, представленный в данной работе, разработан для электронного веб-словаря народной речи Среднего Прииртышья. С рабочей версией веб-словаря можно ознакомиться по адресу: <http://dict.univer.omsk.su>. Представляемая статья является продолжением работ [2, 3].

2. Требования к клиенту «MiddleIrtyshDictionary»

Мобильное приложение «MiddleIrtyshDictionary» представляет собой клиент информационной платформы MediaWiki, основной задачей которого является предоставление пользователям возможности читать статьи с экрана мобильного устройства, работающего на ОС iOS, в частности речь идёт о смартфоне iPhone на iOS v 9.0.2 и выше. Приложение представляет карманный словарь констант народной речи Среднего Прииртышья. Функционал приложения дополняется возможностью работы с учётной записью. Статьи должны быть отсортированы по алфавиту, а также должен быть поиск, история открытых статей, закладки в приложении и возможность делиться статьями в соц. сетях или по e-mail.

Представлена диаграмма прецедентов (см. рисунок 1), которая отражает варианты использования разработанного клиент-приложения. Пользователь — основной исполнитель, который является пользователем приложения «MiddleIrtyshDictionary».

Перечислим основные варианты использования разработанного приложения:

1. Войти в приложение: пользователь может войти в приложение. Вход в приложение возможен посредством авторизации, если пользователь уже имеет аккаунт, или регистрации, если это новый пользователь.
2. Авторизация: пользователь вводит в форму авторизации свой логин и пароль. Если данные указаны верно, то выполняется вход в приложение.
3. Регистрация: пользователь указывает в форме регистрации желаемый логин и пароль, подтверждает ещё раз пароль и e-mail. Если пользователя с таким логином не существует, то выполняется регистрация нового пользователя и вход в приложение, в противном случае пользователю необходимо выбрать другой логин.
4. Выйти из приложения: пользователь может выйти из приложения.
5. Открыть список статей: после входа в приложение пользователю становится доступен экран со списком статей.
6. Открыть статью: на экране со списком опубликованных словарных статей пользователь выбирает любую по заголовку, открывает статью для чтения.
7. Добавить статью в закладки: если статья заинтересовала, у пользователя есть возможность добавить статью в избранные.
8. Открыть меню приложения: доступ к остальным функциям приложения пользователю предоставляется по действию свайпа вправо с левой сторо-

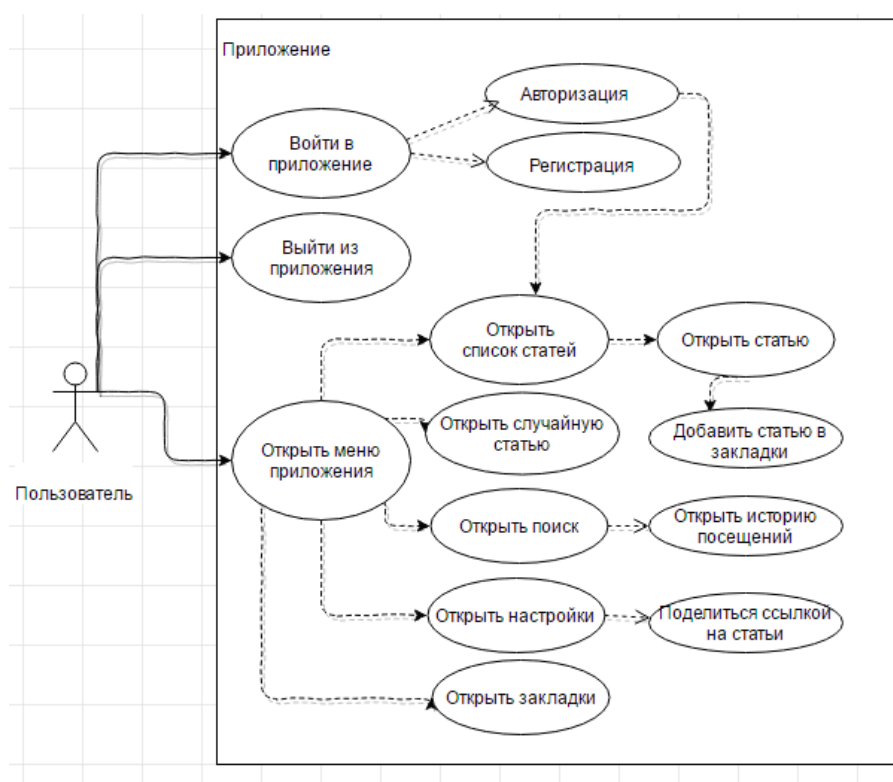


Рис. 1. Диаграмма прецедентов

ны экрана, или по нажатию на иконку меню с вызовом бокового меню.

9. Открыть случайную статью: одной из функций MediaWiki является открытие любой случайной статьи.
10. Открыть поиск: пользователь может начать поиск среди опубликованных статей по заголовку.
11. Открыть историю посещения: не возвращаясь к экрану со списком опубликованных статей, пользователь может вернуться к прочитанной статье через историю.
12. Открыть настройки: экран приложения предоставляет возможность пользователю увидеть имя учётной записи, из-под которой совершается сессия, и завершить сессию.
13. Поделиться ссылкой на статьи: пользователь может отправить сообщение кому-либо со ссылкой на сайт словаря из приложения.
14. Открыть закладки: экран приложения, содержащий перечень ссылок на статьи, которые пользователь добавляет сам при чтении статей.

3. Архитектура приложения «MiddleIrtyshDictionary»

Архитектура рассматриваемого приложения использует паттерн проектирования (см. рисунок 2) модель-представление-контроллер (MVC, model-viewcontroller) [4]. Основная идея данного паттерна заключается в том, чтобы



Рис. 2. Модель-контроллер-представление

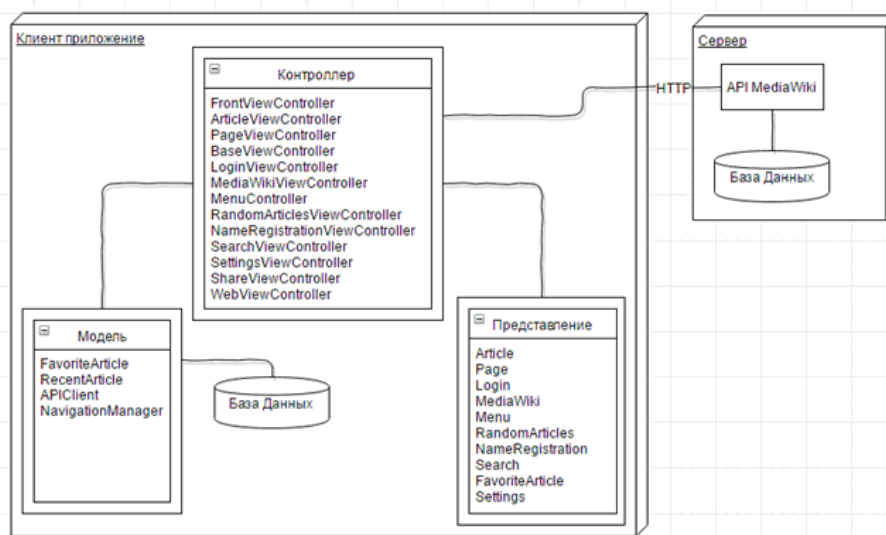


Рис. 3. Архитектура приложения

разделить модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Представлена архитектура разработанного приложения (см. рисунок 3).

Компонент «представление» отвечает за отображение информации. В разработанном приложении данный компонент включает следующие экраны:

- Article — экран со списком статей;
- Page — экран текущей статьи;
- Login — экран авторизации;
- MediaWiki — экран полной версии сайта;
- Menu — экран бокового меню;
- RandomArticles — экран случайной статьи;
- NameRegistration — экран регистрации;
- Search — экран поиска;
- FavoriteArticle — экран с закладками.
- Settings — экран с настройками.

Компонент «контроллер» обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представ-

ление для реализации необходимой реакции. В рассматриваемом приложении модуль контроллеров состоит из:

- `FrontViewController` — контроллер, отвечающий за анимацию бокового меню и кнопки.
- `ArticleViewController` — контроллер, который отвечает за взаимодействие с представлением `Article`.
- `PageViewController` — контроллер, который отвечает за взаимодействие с представлением `Page`.
- `BaseViewController` — контроллер, отвечающий за навигацию всех контроллеров.
- `LoginViewController` — контроллер, который отвечает за взаимодействие с представлением `Login`.
- `MediaWikiViewController` — контроллер, который отвечает за взаимодействие с представлением `MediaWiki`.
- `MenuController` — контроллер, который отвечает за взаимодействие с представлением `Menu`.
- `RandomArticlesViewController` — контроллер, который отвечает за взаимодействие с представлением `RandomArticles`.
- `NameRegistrationViewController` — контроллер, который отвечает за взаимодействие с представлением `NameRegistration`.
- `SearchViewController` — контроллер, который отвечает за взаимодействие с представлением `Search`.
- `SettingsViewController` — контроллер, который отвечает за взаимодействие с представлением `Settings`.
- `ShareViewController` — контроллер, который отвечает за взаимодействие с представлением `FavoriteArticle`.
- `WebViewController` — представляет собой область, которая может отображать HTML-контент.

Компонент «модель» предоставляет данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние.

В разработанном приложении модель состоит из:

- `APIClient` — модель, в которой устанавливается `content-type` (`application/json`, `text/html`), и методы запросов (`GET`, `POST`), а также URL, на который они будут отправляться.
- `NavigationManager` — модель, содержащая кнопки для навигации в отдельном баре.
- `NSError+APIClient` — модель, которая обрабатывает ошибки с сервера.
- `NSError + NSresponseData` — модель, которая показывает ответ с сервера.
- `UIViewController+AlertHelpersViewController` — модель, которая показывает уведомление об ошибке, а также классов, взаимодействующих с базой данных (`core Data`):
- `FavoriteArticle` — модель, работающая с добавлением элементов в избранные.
- `RecentArticle` — модель, работающая с добавлением элементов в историю посещений.

Таблица 1. Таблица FavoriteArticle

№	Атрибут	Семантика	Тип
1	Date	Дата добавления	DATE
2	PageID	id страницы	STRING
3	Title	Заголовок статьи	STRING
4	Url	Ссылка на статью	STRING

Для хранения закладок и истории посещений используется база данных, состоящая из двух таблиц: FavoriteArticle и RecentArticle. Описание таблицы, в которой хранятся избранные статьи, приведено в табл. 1. В аналогичной таблице хранится история посещений — Таблица RecentArticle.

4. Графический интерфейс приложения

Компонент «представление» состоит из набора экранов, созданных с помощью стандартного средства платформы OS X — Interface Builder. Interface Builder предоставляет палитры или коллекции объектов пользовательского интерфейса для разработчиков. Эти объекты содержат такие элементы, как текстовые поля, таблицы данных, слайдеры и всплывающие меню. Палитры Interface Builder являются полностью расширяемыми, то есть любой разработчик может создавать новые объекты и добавлять их к уже существующим. Для создания интерфейса разработчик просто перетаскивает элементы интерфейса с палитры на окно или меню. Конкретные объекты, которые получают сообщения, указываются в коде приложения. Таким образом, все инициализации происходят до выполнения, что ведёт к повышению производительности и делает процесс разработки более упорядоченным. Разработанное приложение включает в себя 10 экранов. При входе в приложение игрок может авторизоваться (представление LoginViewController) или создать аккаунт, т.е. зарегистрироваться (представление NameRegistrationViewController), см. рисунок 4.

После того как пользователь авторизуется, осуществляется переход на экран со списком статей, представление ArticleViewController (см. рисунок 5), на скриншоте экрана также присутствует библиотека MJNIndexPathView. Находясь в представлении ArticleViewController, пользователь может открыть статью (нажатием по заголовку) для чтения. После выбора элемента из списка (статьи) открывается представление PageViewController (см. рисунок 5), в представлении PageViewController имеется возможность добавить текущую статью в избранные, список добавленных статей в закладки отображается в представлении FavoriteArticle (см. рисунок 6).

Представление FavoriteArticle содержит метод segmentedControlChanged типа IBAction, что в результате является кнопкой-переключателем с «Закладок» на «Последние посещения», последнее — это список с историей открытия статей. Модели FavoriteArticle (закладки) и RecentArticle (послед-

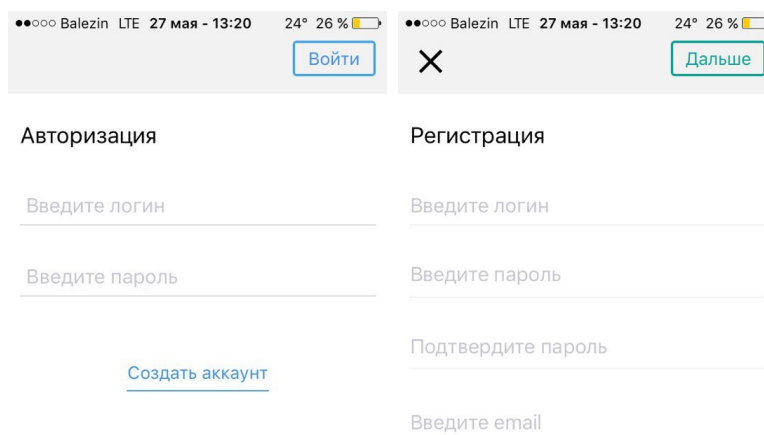


Рис. 4. Экраны LoginViewController и NameRegistrationViewController

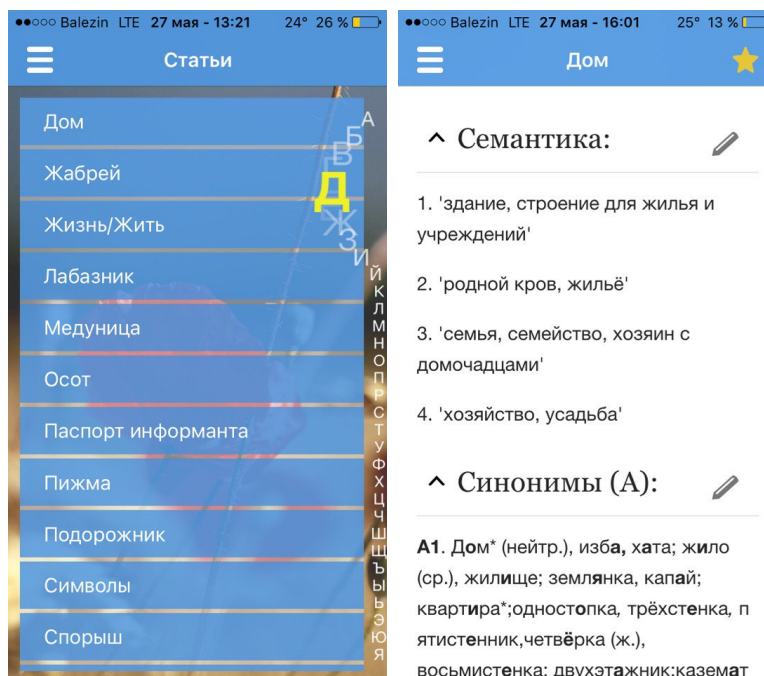


Рис. 5. Экран ArticleViewController и Экран PageViewController

ние посещения) доступны в представлении FavoriteArticle через контролер ShareViewController (см. рисунок 6). Открыть меню приложения для доступа к остальным функциям приложения пользователю предлагается свайпом вправо с левой стороны экрана или по нажатию на иконку меню. Меню является представлением MenuController, боковое меню разработано на основе библиотеки SWRevealViewController.

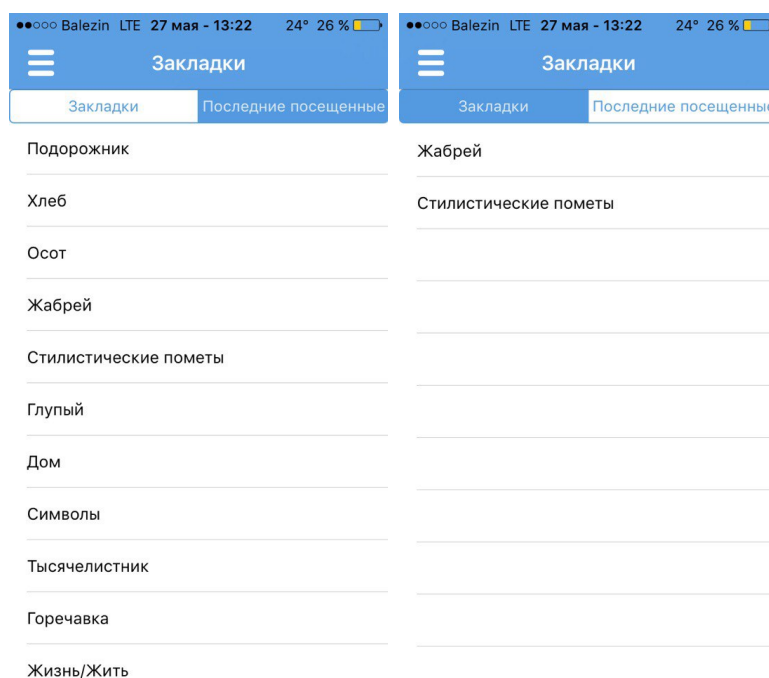


Рис. 6. Экран FavoriteArticle и Экран FavoriteArticle

Представление MenuController, меню приложения, содержит перечень ссылок в виде кнопок на функции приложения, представляет навигацию внутри приложения. Из представления MenuController уже доступны и другие экраны, например, случайная статья, поиск, полная версия, настройки. Экран со случайной статьёй, представление RandomArticlesViewController, предоставляет возможность пользователю открыть любую случайную статью из опубликованных, реализуется функция информационной системы MediaWiki. Экран поиска, представление SearchViewController, представляет возможность пользователю, начать поиск среди опубликованных статей по заголовку. Экран полной версии, представление MediaWikiViewController, открывает веб-сайт, сайт-словарь констант народной речи Среднего Прииртышья. Экран настройки, представление SettingsViewController (см. рисунок 7), предоставляет возможность пользователю увидеть имя учётной записи, из-под которой совершается сессия, и завершить сессию. Поделиться ссылкой на статьи: пользователь может отправить сообщение кому-либо со ссылкой на сайт словаря из приложения.

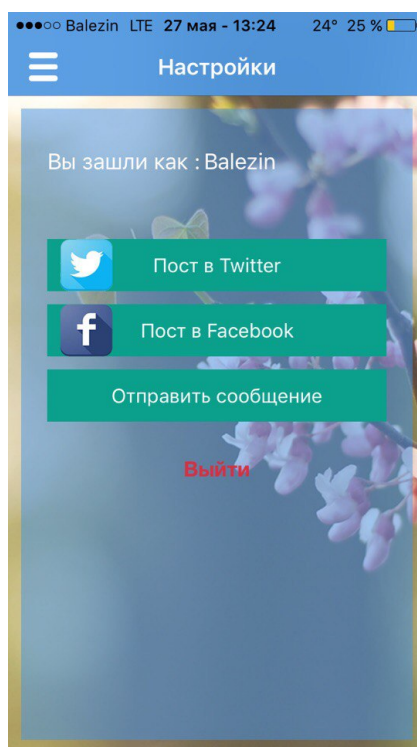


Рис. 7. Экран SettingsViewController

Заключение

Основной результат работы — это разработанная и реализованная архитектура мобильного клиента под iOS с возможностью чтения статей с сайта на небольшом экране. Проработан и реализован эргономичный, интуитивно понятный графический интерфейс мобильного приложения.

ЛИТЕРАТУРА

1. Харламова М.А. Константы народной речемысли и их лексикографическая интерпретация. Омск : Изд-во Ом.гос.ун-та, 2014. 290 с.
2. Лавров Д.Н., Харламова М.А. Словарь констант народной речи: выбор платформы представления // Вестник Омского университета. Омск : Ом. гос. ун-т, 2015. № 1(75). С. 213–215.
3. Балезин И.А., Лавров Д.Н. Разработка формы для заполнения словарных статей и мобильного клиента под iOS для словаря констант народной речи Среднего Прииртышья // XL региональная студенческая научно-практическая конференция «Молодёжь третьего тысячелетия». 8–30 апреля 2016 г. 8 с. В печати.
4. Piper I. Learn XCode Tools for Mac OS X and iPhone Development. USA: Appress, 2009. 450 p.

**THE ARCHITECTURE OF THE MOBILE CLIENT FOR IOS TO ACCESS
WEB-DICTIONARY OF FOLK SPEECH OF THE MIDDLE IRTYSH**

I.A. Balezin

Student, e-mail: iabalezin@gmail.com

D.N. Lavrov

Ph.D. (Eng.), Associate Professor, e-mail: dmitry.lavrov72@gmail.com

M.A. Harlamova

Ph.D. (Philological), Associate Professor, e-mail: khr-spb@mail.ru

Dostoevsky Omsk State University

Abstract. The architecture is presented and the mobile client-server application is developed for access from iPhone to the dictionary of the constants of folk speech of the Middle Irtysh area with the necessary set of options.

Keywords: application architecture, of iOS, mobile app, dictionary of folk speech.

Дата поступления в редакцию: 31.10.2016