

При нажатии на строку списка пользователь увидит экран с подробной информацией о контакте (рисунок 3). Вверху экрана расположена панель навигации. Возврат на предыдущую страницу осуществляется по нажатию на значок «←».

Реализовано приложение, выполняющее основные функции телефонного справочника. Гибридное мобильное приложение адаптировано для работы на мобильных устройствах под управлением Android и iOS. Серверная часть приложения разработана на платформе IBM MobileFirst, отвечает за проверку авторизации пользователей и передачу информации по защищённому каналу.

При разработке использовались языки HTML5, CSS3, Java Script, TypeScript, а также фреймворки Angular 2, Cordova, Ionic2, IBM MobileFirst.

## Литература

1 IBM MobileFirst Strategy Software Approach / E. Duong [et al.]. – Springville. UT : Vervante, 2014. – 272 с.

2 The top open source framework for building amazing mobile apps [Electronic resource]. – Mode of access : ionicframework.com. – Date of access : 18.02.2020.

3 One framework. Mobile & desktop [Electronic resource]. – Mode of access : angular.io. – Date of access : 28.01.2020.

4 Фрейн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – Санкт-Петербург : Питер, 2017. – 272 с.

5 IBM Mobile Foundation. [Electronic resource]. – Mode of access : mobilefirstplatform.ibmcloud.com/tutorials/en/. – Date of access : 15.01.2020.

УДК 004.41:378.146:004.43

*А. Ю. Пищик*

### РАЗРАБОТКА И СОЗДАНИЕ СИСТЕМЫ ТЕСТИРОВАНИЯ ПО КУРСУ «ЯЗЫКИ ПРОГРАММИРОВАНИЯ»

*Статья посвящена разработанной системе закрепления и проверки знаний, полученных в процессе обучения. Система включает в себя два приложения: одно предназначено для работы с базой данных и просмотра результатов, второе – непосредственно для прохождения тестовых заданий. В первом приложении администратор может вводить задания тестов, просматривать результаты прохождения тестов, получать различные графики и отчёты. Во втором приложении у пользователя есть возможность выбрать тест для прохождения из списка. Систему проверки знаний можно использовать, как и для самопроверки так для централизованной проверки знаний учащихся и студентов учебных заведений.*

В современном мире образование считается одним из самых важных аспектов человеческой жизни. Образование включает в себя получение знаний и проверку полученных знаний. Благодаря проверке можно понять, какой материал усвоен, а над каким ещё следует поработать. Опыт использования компьютерных программ в процессе обучения показывает, что наибольшая эффективность достигается при использовании программ, ориентированных на обучение решению проблем (задач) в различных предметных средах при организации диалога между компьютером и обучаемым. Чем выше интенсивность диалога, тем более эффективно проходит процесс обучения.

В развитых странах компьютеризации образования уделяется значительное внимание, существуют национальные программы в этой области. Однако реальный эффект улучшения образования чаще всего намного меньше, чем ожидаемый. В нашей стране компьютеризации образования придается большое значение. В связи с этим необходимо проанализировать обоснованность требований к ожидаемым результатам с учетом уже имеющегося опыта, выбрать наиболее перспективное направление дальнейшего развития и в соответствии с ним реализовать обучающие системы.

Традиционное преподавание накопило достаточно большой арсенал разнообразных форм обучения и контроля знаний. Растущие потребности в интенсификации процесса обучения на протяжении многих лет стимулировали исследование применений все новых и новых технических средств, идей и методов обучения.

Существует огромное количество способов получения знаний: университет, книги, электронные книги, платные и бесплатные онлайн-курсы. Для проверки знаний чаще всего используется тестирование, так как именно его проще всего автоматизировать и результаты можно получить сразу же.

Как раз для проверки знаний мною была разработана система тестирования. Она включает в себя два приложения: для работы с тестами и для прохождения тестов. В качестве темы тестов были выбраны языки программирования, однако систему можно использовать для любой тематики [1, с. 164].

Оба приложения работают с одной базой данных, которая включает в себя ряд связанных таблиц, необходимых для полноценной работы с тестами: таблица тестов, вопросов, ответов, результатов, студентов, тьюторов, группы студентов.

Кроме таблиц были разработаны триггеры, которые поддерживают целостность данных, заполняют вычисляемые столбцы, реализуют дополнительную логику. Например, нельзя добавить два правильных ответа на вопрос. Также были разработаны различные функции и процедуры, которые позволяют подсчитывать статистику по прохождению тестов, фильтровать таблицы по указанным параметрам, просматривать пользователей и их роли. В базе данных предусмотрено разделение пользователей по ролям, и каждая роль имеет определенные права на запись, просмотр, удаление и выполнение. Все это реализовано в sql-скриптах. В качестве базы данных использовался MS SQL Server. Для того, чтобы пользователи могли работать с базой, была проведена настройка как самой базы (разрешение удаленного подключения по TCP/IP), так и настройка VPN.

Приложение для работы с тестами написано на языке C# с использованием Windows Forms фреймворка. Оно представляет собой оконное приложение для Windows.

В нем реализована авторизация для входа в базу данных, далее пользователь, в зависимости от его роли и уровня доступа, выбирает доступные ему функции. Реализовано три роли: administrators, testCreators и students. Администраторы могут регистрировать новых пользователей, просматривать уже существующих, просматривать журналы изменения таблиц, выполнять все функции, изменять все таблицы. Создатели тестов могут редактировать созданные ими тесты, вопросы, ответы, выполнять определенные функции, редактировать таблицы студентов и групп. Студенты могут выполнять некоторые функции. Никто не может изменять таблицу результатов тестов.

В приложении существует три основных функции: заполнение таблиц, редактирование и удаление данных в таблицах, просмотр статистики.

Для заполнения сделана специальная форма, в которой пользователь может ввести данные для новой записи в выбранной им таблице. Для добавления данных реализовано следующее: проверки на пустоту для текстовых и числовых полей, запрет на ввод не цифр для числовых полей, для внешних ключей для целостности данных реализован выбор из списка, например, заполнение поля в вопросе, ссылающееся на код теста, выглядит как список названий тестов. По форме можно перемещаться, используя клавишу Tab, добавлять запись с помощью клавиши Enter.

Также для добавления тестов целиком предусмотрено добавление теста из файла. На форме расположен шаблон, которому содержимое файла должно соответствовать, далее пользователь с помощью проводника выбирает файл. Если он соответствует шаблону и все данные введены верно, автоматически заполняются таблицы тест, вопрос, ответ. Если же что-то пошло не так, транзакция отменяется и ничего не будет добавлено.

Для редактирования сделана форма, в которой пользователь в табличном виде видит данные выбранной им таблицы. Удаление данных происходит выделением записи с последующим нажатием клавиши Del. Обычные поля можно редактировать прямо в таблице, изменения будут сохранены, как только пользователь перейдет в другое поле. Для редактирования внешних ключей разработана такая же логика, как и в добавлении. Пользователю доступны следующие функции: фильтрация по трем-четырем критериям, выбор столбцов для отображения, изначальная фильтрация по создателю для членов роли testCreators. Для фильтрации разработана специальная форма, где пользователь может выбрать критерий и параметры.

Просмотр статистики включает в себя ряд различных графиков, таблиц, построенных на основе хранимых функций и процедур. Пользователь может выбрать тот вид статистики, который он хочет, отфильтровать полученные данные.

Администратор может зарегистрировать пользователей в базу данных. Работает это так: администратор выбирает из таблицы студентов и тьюторов пользователя, присваивает ему логин, пароль и роль, далее срабатывает триггер, который устанавливает поле логин для данного студента/тьютора.

Кроме этого, с помощью триггеров реализован журнал действий для некоторых таблиц, например, если кто-то удалит тест, будет внесена запись, кто это сделал, когда и какой тест. В приложении администратор может просматривать этот журнал и отменять какие-то действия.

Приложение для прохождения тестов написано на языке C# с использованием фреймворка Unity. Оно представляет собой мобильную игру для Android.

В приложении предусмотрена авторизация по логину и паролю, входить имеют право только студенты. Если что-то введено не верно или у пользователя недостаточно прав, он увидит окно, в котором будет написано, что пошло не так.

Далее студент может выбрать тест (выбор идет из тестов, в которых 40 вопросов и 4 ответа). После этого при нажатии на клавишу играть, выполняется запрос в базу данных, который получает вопросы и ответы на них, выбирает из этих вопросов то количество, которое указал создатель тестов. Также в зависимости от соответствующих полей в таблице теста устанавливается время на ответ, возможность использования подсказок.

Студенту нужно выбрать ответ на вопрос. Ответы идут в рандомном порядке. При каждом его ответе отправляется вставка в таблицу результатов, где указывается логин студента, тест, вопрос, ответ, правильность, количество потраченного времени. Если создателем теста было предусмотрено использование подсказок, студент может один раз за игру использовать заморозку времени, убиение одного неправильного ответа.

Кроме этого, студент может посмотреть свой рекорд по каждому тесту, выключить/включить музыку, сменить пользователя, вернуться в меню, поделиться скриншотом.

Так как сервер с базой данных располагается на одном компьютере, а доступ к нему нужно иметь с любого устройства, была настроена виртуальная частная сеть. Она работает поверх интернета, а потому подключиться к ней можно откуда угодно. В качестве логина и пароля для пользователей используются их логин и пароль для входа в базу данных.

Система разработана таким образом, что не составит труда переделать её под другую тематику. Систему проверки знаний можно использовать для централизованной проверки знаний учащихся и студентов.

Проблемы компьютерного обучения, о чем говорилось выше, не сводятся к массовому производству компьютеров и встраиванию их в существующий учебный процесс. Изменение средства обучения, как, впрочем, и изменения в любом звене дидактической системы, неизбежно приводят к перестройке всей этой системы. Использование вычислительной техники расширяет возможности человека, однако оно является лишь инструментом, орудием решения задач, и его применение не должно превращаться в самоцель, моду или формальное мероприятие.

Судьба компьютеризации в конечном счете будет зависеть от педагогически и психологически обоснованной перестройки всего учебно-воспитательного процесса.

### Литература

1 Пищик, А. Ю. Разработка системы проверки знаний по курсу «Язык программирования» / А. Ю. Пищик, Г. Л. Карасёва // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: материалы XXIII Республиканской научной конференции студентов и аспирантов, Гомель, 23–25 марта 2020 г. / ГГУ им. Ф. Скорины; редкол.: С. П. Жогаль (гл. ред.) [и др.]. – Гомель, 2020. – С.164–165.

УДК 519.2

*А. В. Потехин*

### ИССЛЕДОВАНИЕ СЕТИ МАССОВОГО ОБСЛУЖИВАНИЯ С ДЛИННОЙ ОБРАТНОЙ СВЯЗЬЮ

*Статья посвящена исследованию стационарного функционирования трёхузловой марковской сети массового обслуживания с длинной обратной связью. Составлены уравнения глобального и локального равновесия, установлено условие эргодичности, найдено стационарное распределение вероятностей состояний, применён метод обращения времени и исследован выходящий поток.*

Теория массового обслуживания является разделом теории случайных процессов, которая изучает класс задач, возникающих на практике. Моделируются ситуации, связанные с обработкой поступающих требований, заявок к обслуживающему устройству. В настоящее время модели теории массового обслуживания находят широкое применение в компьютерных сетях, сетях передачи данных и т. д. [1, 2].

Сетью массового обслуживания называется совокупность одновременно функционирующих систем массового обслуживания, в которой заявки перемещаются в соответствии с матрицей маршрутов.

Исследована открытая марковская сеть с тремя узлами, в которую поступает пуассоновский (простейший) поток заявок с параметром  $\lambda$ . Все заявки входного потока направляются в первый узел. Времена обслуживания заявок в различных узлах независимы, не зависят от процесса поступления заявок и имеют показательное распределение с параметрами  $\mu_i, i=1,2,3$ . После окончания обслуживания заявки в первом узле она переходит на обслуживание во второй узел, а из второго – в третий. После окончания обслуживания заявки в третьем узле она с вероятностью 0,25 переходит в очередь первого узла, а с вероятностью 0,75 покидает сеть. Функционирование сети описывается марковским процессом  $n(t)$ . Матрица маршрутов имеет вид