

УДК 004.42

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ОРГАНИЗАЦИОННОГО ОБЕСПЕЧЕНИЯ СОРЕВНОВАНИЙ ПО ПРОГРАММИРОВАНИЮ

Д. О. Трошин, М. С. Русакова

В статье рассматривается разработка кроссплатформенного мобильного приложения организационного обеспечения проводимого в университете командного чемпионата Поволжья по программированию. В работе спроектирована база данных для хранения организационных сведений о чемпионате, спроектировано приложение с помощью унифицированного языка моделирования UML. Для приложения используется клиент-серверная архитектура. Основными инструментами реализации выбраны: платформа Flutter, облачная база данных Firestore, локальная база данных SQLite, среда разработки Visual Studio Code. Реализованное мобильное приложение позволяет как просматривать справочную информацию Чемпионата, так и выполнять организационные действия для него.

Ключевые слова: клиент-серверная архитектура, олимпиадное программирование, базы данных и системы управления базами данных, фреймворк, визуализация, паттерн проектирования.

Неотъемлемой частью современного информационно-технологического сообщества является организация и проведение различных конференций, фестивалей и чемпионатов. Данные мероприятия позволяют людям, имеющим отношение к информационным технологиям, расширять свой кругозор в профессиональной деятельности, получать бесценный опыт или перенимать его у коллег из других информационных инфраструктур, а также применять полученные знания в событиях соревновательного типа.

При участии в чемпионатах по программированию или различных конференциях всю информацию, как правило, можно найти на информационных ресурсах организаторов мероприятий. В качестве таких ресурсов чаще всего выступает web-сайт [1], который можно просмотреть и на мобильном устройстве, но из-за значительно меньших габаритов по сравнению с современным компьютером или ноутбуком необходимая информация часто бывает представлена в неудобном виде.

Чтобы дать участникам соревнований оперативный доступ к организационной информации IT-мероприятия и представить ее в

наиболее удобном виде можно создать мобильное приложение организационного обеспечения [2, 3].

Постановка задачи и формирование требований к приложению

Каждый год весной в Самарском Университете проходит открытый командный чемпионат Поволжья по спортивному программированию [1]. Соревнования по спортивному программированию позволяют студентам продемонстрировать умения в разработке и тестировании программ, навыки командной работы, знания алгоритмов и структур данных. Чемпионат проводится в один тур в формате проведения международной студенческой олимпиады по программированию в корпусе Самарского Университета.

Мероприятие проходит в течение четырех дней. Первый день – заезд иногородних участников; день второй – открытие чемпионата, встреча с членами оргкомитета, жюри и технического комитета, пробный тур и разбор задач. Третий день – основной тур в компьютерных классах университета, разбор задач. Четвертый день – день отъезда для

© Трошин Д. О., Русакова М. С., 2020.

Трошин Дмитрий Олегович (deman.troshin@mail.ru),

студент кафедры информатики и вычислительной математики

Русакова Маргарита Сергеевна (ruta@ssau.ru),

доцент кафедры информатики и вычислительной математики

Самарского университета,

443086, Россия, г. Самара, Московское шоссе, 34.

иногородних участников [1]. К участию в чемпионате приглашаются команды, состоящие из трех студентов или аспирантов одного вуза.

Основываясь на информации со страницы чемпионата, можно сделать вывод, что мобильное приложение должно:

- позволять просматривать личный профиль каждого из участников и общий список команд;
- получать актуальную информацию о времени и месте проведения каждого из этапов соревнований
- обеспечивать своевременную доставку новой информации;
- обеспечивать информационную помощь в расселении участников и координации команд посредством карт и схем;
- предоставить возможность пройти итоговый опрос и ознакомиться с результатами соревнований и проводимых конкурсов.

Для волонтеров приложение должно также стать средством координирования действий согласно их роли и связи с организаторами, а для организаторов — помощником в проведении и организации чемпионата.

На основе анализа предметной области были выдвинуты следующие нефункциональные требования:

- надежность, устойчивость и безопасность;
- дружественный интерфейс;
- операционная система Android 6.0 и выше/iOS 10.0 и выше;
- разграничение прав доступа;
- наличие на мобильном устройстве интернет-соединения 2g и выше.

Архитектурное проектирование

Первым этапом проектирования мобильного приложения является выбор паттерна проектирования – повторяемой архитектурной конструкции, представляющей собой решение проблемы проектирования в рамках некоторого часто возникающего контекста [4]. В рамках данной работы был выбран паттерн MVC (Model-View-Controller, Модель-Представление-Контроллер), где

- Модель (model) – часть, которая содержит в себе функциональную бизнес-логику приложения, предоставляет данные и реагирует на команды контроллера, изменяя своё состояние;

- Представление (view) – отображает данные, полученные, в свою очередь, от модели. Не может влиять непосредственно на модель, а только обладает чтением данных;

- Контроллер (controller) – обработчик событий, инициируемых пользователем. Другими словами, контроллер перехватывает события извне и, реагируя на событие, изменяет модель [5].

В данном случае моделью является база данных для хранения организационной информации чемпионата, отображение – это интерфейс приложения, визуализированный при помощи контроллера, который в свою очередь обрабатывает полученные данные из базы.

В качестве архитектуры была выбрана трехуровневая архитектура клиент-сервер (тонкий клиент). В данной архитектуре алгоритмы, вычислительная и сетевая нагрузки распределены между серверами и клиентами.

Проектирование базы данных

Из семантического описания процесса проведения соревнований можно выделить следующие основные сущности. «Пользователь» – центральная сущность, содержит информацию о всех пользователях, которые принимают участие в мероприятии. «Команда» – сущность характеризует команды, которые принимают участие в чемпионате. «Аудитория» – сущность содержит в себе сведения об аудиториях, в которых на момент проведения чемпионата находятся команды. «Результаты чемпионата» – сущность, в которой содержится информация о местах и количестве решенных задач командами. «Чемпионат» – сущность, характеризующая проводимые ежегодно чемпионаты. «Спонсор» – содержит сведения о спонсорах мероприятия. «Конкурс» – сущность, содержащая информацию о проводимых конкурсах вне основного тура. «Новость» – сущность, отражающая актуальные новости до, во время и после проведения чемпионата. «Фото» – сущность, отражающая фотографии, сделанные во время проведения мероприятия.

Но семантических данных недостаточно для построения базы данных, необходима модель, которая строится на основе конкретной информационной структуры. Реляционная модель представляет собой прототип

будущей базы данных, является удобным представлением данной предметной области и строится на основе конкретной информационной структуры [6].

Исходя из особенностей и правил построения реляционных баз данных, а также с применением метода нормализации, была построена реляционная модель (рис. 1).

Проектирование приложения

Одним из требований к системе было разделение прав доступа. Для системы выделены роли «Неавторизованный пользователь», «Авторизованный пользователь», который является обобщением ролей «Волонтер», «Организатор», «Участник». Сами варианты использования являются спецификацией функций, предоставляемых актерам (рис. 2) [7].

На диаграмме вариантов использования видно, что такие действия, как просмотр расписания, списка команд, конкурсов, новостей, фотографий, информации о спонсорах и результатов соревнований, доступны как авторизованному, так и неавторизованному пользователю. В то же время доступ к различным организационным возможностям (просмотр памятки волонтеров, добавление новых участников и конкурсов, редактирование расписания, прохождение опроса) доступны только конкретным ролям авторизованного пользователя.

Для анализа взаимодействия объектов с системой необходимо построить диаграмму коопераций, которая определяет структуру поведения системы в терминах взаимодействия участников этой кооперации (рис. 3).

Для демонстрации общей структуры иерархии классов, их коопераций, полей, методов, интерфейсов и взаимосвязей между ними необходимо создать диаграмму классов [8]. Диаграмма классов представляет собой диаграмму языка моделирования UML, используемую для документирования, визуализации, а также для конструирования системы посредством проектирования (рис. 5).

Выбор средств реализации и примеры форм приложения

В качестве основного инструмента реализации выбрана платформа «Flutter» от компании Google и объектно-ориентированный язык «Dart». Одним из главных преимуществ языка Dart является «горячая перезагрузка», когда изменение исходного кода сразу отражается в работающем приложении без необходимости его перезапуска. Также «Flutter» был выбран благодаря тому, что он позволяет создавать кроссплатформенные приложения.

Средой разработки была выбрана Visual Studio Code. В качестве облачной СУБД использовалась NoSQL база данных «Firestore» от «Firebase», позволяющая синхронизировать данные между клиентскими приложениями через слушателей в реальном времени. В качестве локальной СУБД была выбрана встраиваемая кроссплатформенная SQLite. Наконец, в качестве сервера хранения файловых данных был выбран сервис «Storage» от «Firebase».

Из особенностей реализации стоит выделить осуществление автоматической рассылки уведомлений на мобильные устройства пользователей, что гарантирует своевременное получение важной и актуальной информации. Новая информация оперативно поступает в раздел «Новости», для этого был создан триггер на таблицу «News» в облачной базе данных, реагирующий на добавление данных. При добавлении новой информации, триггер обрабатывает поступившие данные и рассылает уведомления на экран мобильных устройств пользователей. Для реализации данной функции использовался сервис «Cloud Messaging» от «Firebase». Триггер написан на языке JavaScript с использованием платформы NodeJS, позволяющая преобразовать JavaScript из узкоспециализированного языка в язык общего назначения.

Примеры форм приложения представлены на рисунке 6.

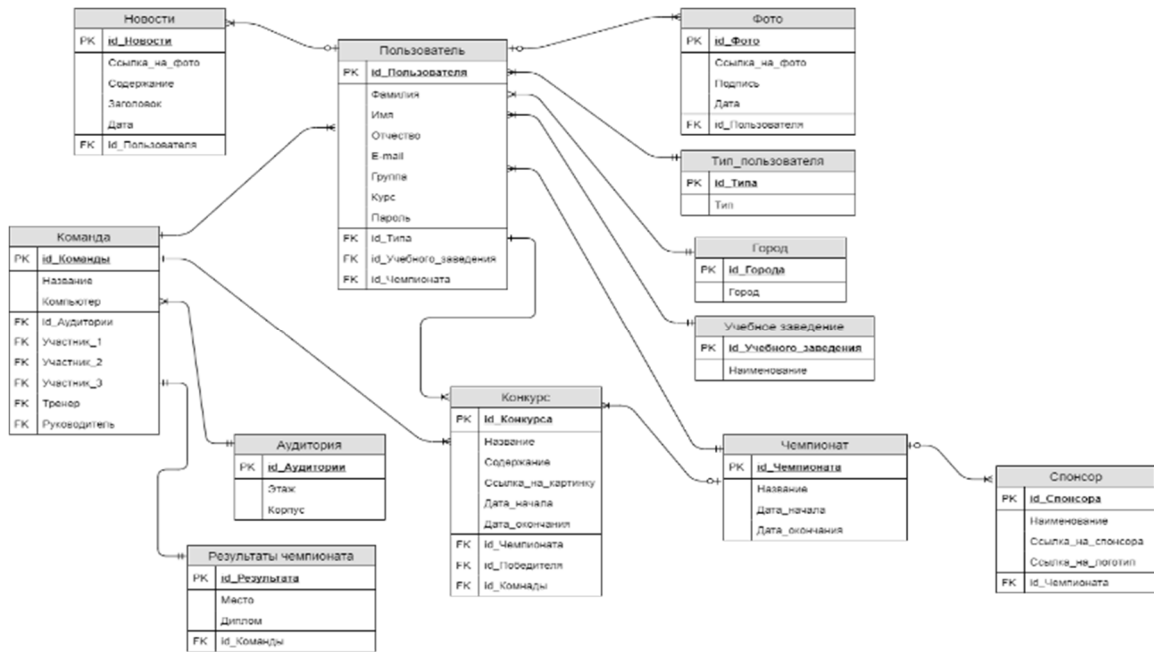


Рис. 1. Реляционная модель базы данных

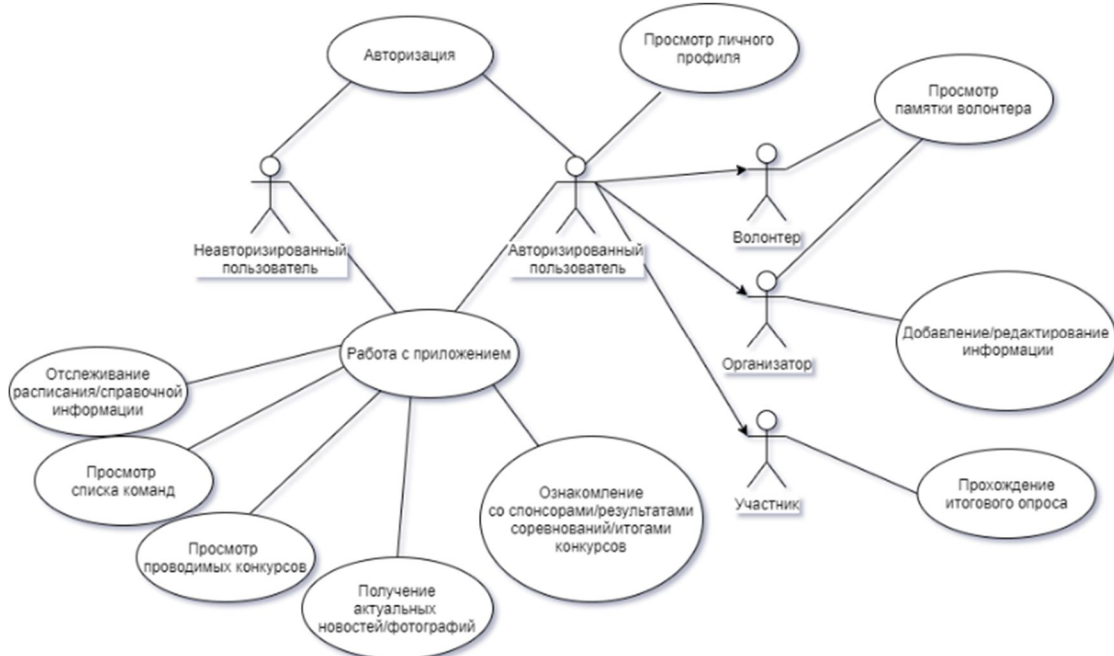


Рис. 2. Диаграмма вариантов использования

Литература

1. X (XXI) открытый командный студенческий чемпионат Поволжья по спортивному программированию. URL: http://contest.uni-smr.ac.ru/ru/championship_2020/ (дата обращения: 12.03.2020).
2. International Collegiate Programming Contest. URL: <https://icpc.baylor.edu/> (дата обращения: 06.02.2020).
3. Фестиваль деятелей интернета. URL: <https://2019.404fest.ru/> (дата обращения: 20.02.2020).
4. Паттерн. URL: <https://shwanoff.ru/tag/pattern/> (дата обращения: 14.04.2020).
5. Паттерны для новичков: MVC vs

MVP vs MVVM. URL: <https://habr.com/ru/post/215605/> (дата обращения: 14.04.2020).

6. Дейт К. Дж. Введение в системы баз данных. М.: Вильямс, 2005. 1328 с.

7. Диаграмма вариантов использования как концептуальное представление бизнес-системы в процессе ее разработки. URL: <https://devtype.blogspot.com/2015/06/Diagramma-variantov-ispolzovaniya-kak-kontseptualnoe-predstavlenie-biznes-sistemy-v-protssesse-ee-razrabotki.html> (дата обращения: 15.04.2020).

8. UML-диаграммы классов. URL: <https://prog-cpp.ru/uml-classes/> (дата обращения: 16.04.2020).



Рис. 3. Диаграмма коопераций

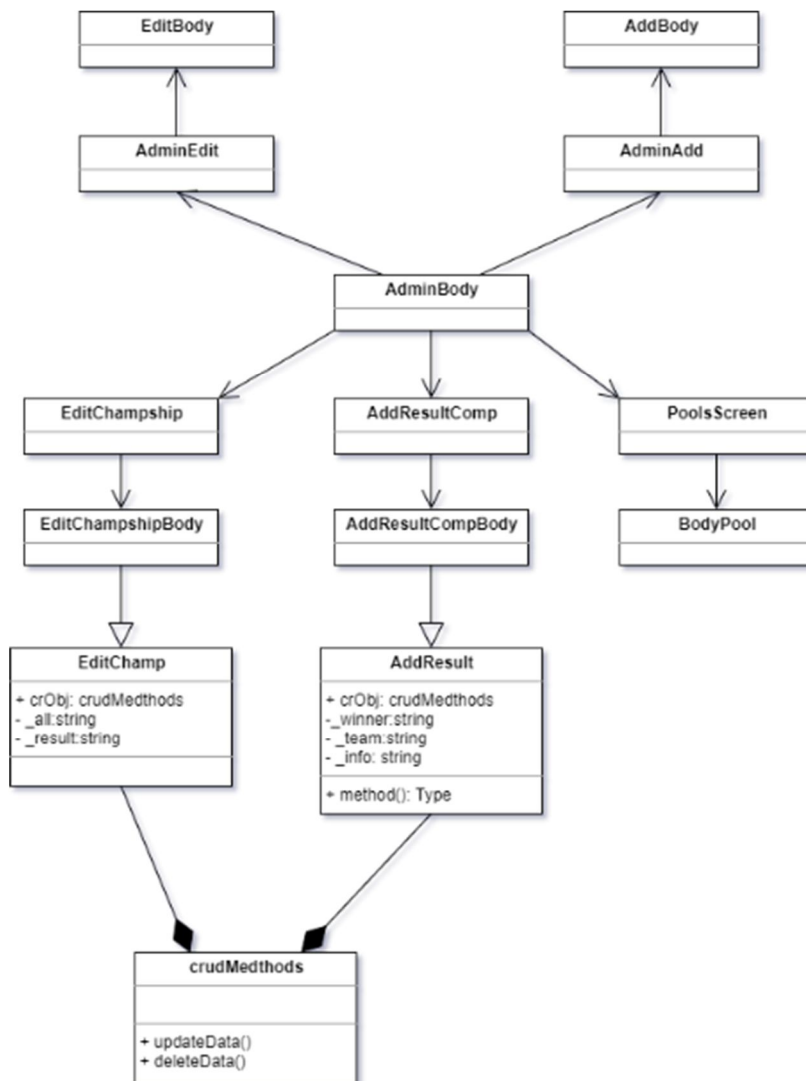


Рис. 4. Диаграмма классов раздела «Панель администрирования»

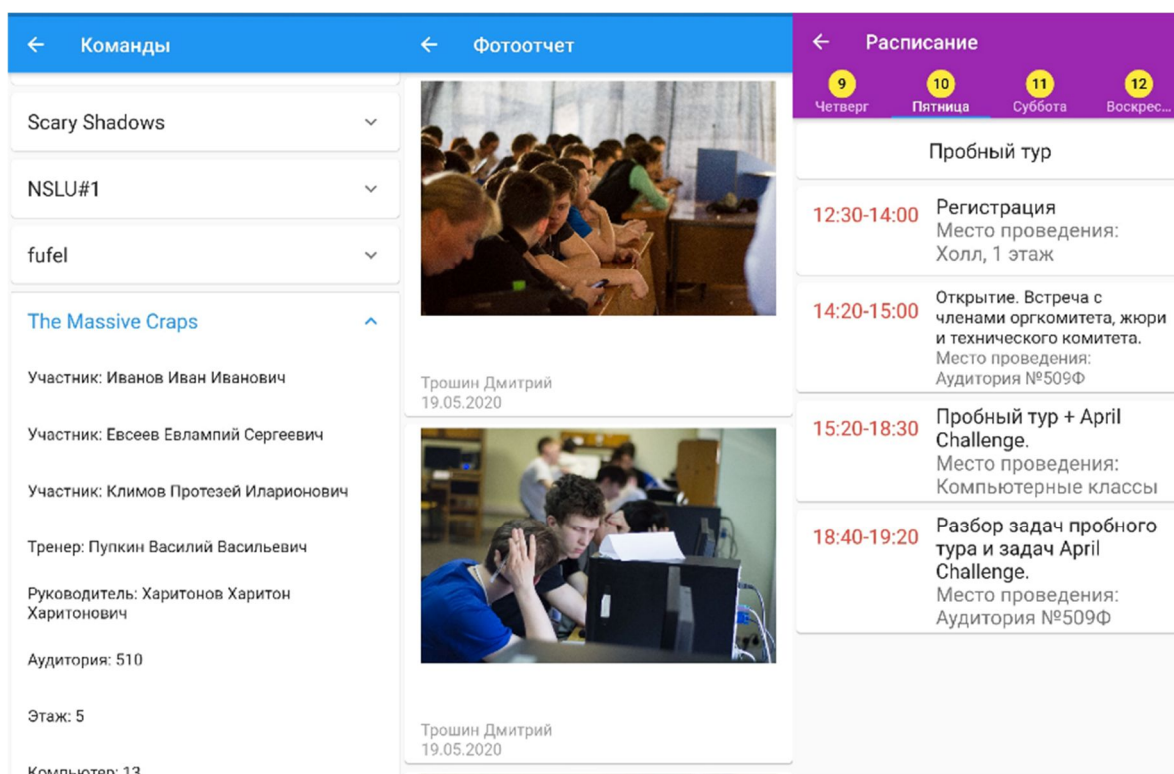


Рис. 5. Примеры форм приложения

MOBILE APPLICATION DEVELOPMENT TO SUPPORT PROGRAMMING CONTEST

D. O. Troshin, M. S. Rusakova

The article considers crossplatform mobile application development to support the Volga region Programming Team Contest, which is taken place at the University. Database was developed for storing of the organizational information about the contest. The application was designed with the help of the unified modeling language UML, the design pattern and client-server architecture were selected for the considering mobile application. The main implementation tools were: Flutter platform, Firestore cloud database, local SQLite database, Visual Studio Code development environment. Implemented application allows one both to see contest reference information and to act as an organizer.

Key words: client-server architecture, olympiad programming, databases and database management systems, framework, visualization, design pattern.

Статья поступила в редакцию 08.07.2020 г.

© Troshin D. O., Rusakova M. S., 2020.

Troshin Dmitriy Olegovich (deman.troshin@mail.ru), student

Rusakova Margarita Sergeevna (ruma@ssau.ru),

assistant professor of the department of Informatics and Computing Mathematics of the Samara University, 443086, Russia, Samara, Moskovskoye Shosse, 34.