

осуществляется сбор информации, которая впоследствии позволяет компании приобрести объективные и точные данные о состоянии предприятия. Обработанные данные доставляются во все отделы предприятия, что помогает наладить взаимодействие между сотрудниками разных подразделений и принимать обоснованные решения.

В. А. Логвинец, Е. В. Рафалова
(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

ВЕБ-СЕРВИС ПОТОКОВОГО ОБМЕНА СООБЩЕНИЯМИ

Информационные технологии уверенно вошли в наш век, нового уровня достигла и мировая культура под их влиянием. Всё большую популярность среди пользователей Интернета набирает видеостриминговые площадки, постепенно заменяющие телевидение.

Видеостриминг – это потоковое онлайн-вещание, при котором происходит непрерывная передача видео от провайдера (стримера) конечному пользователю. Каноничная видеостриминговая площадка условно состоит из двух частей: области видео, необходимой для вывода трансляции на экран пользователя, и области чата, необходимой для общения пользователей в реальном времени. Из них чат является наиболее абстрактным понятием и на большинстве видеостриминговых площадок имеет уникальную реализацию. Вопрос грамотной интерпретации чата в проекте является одним из основополагающих в сфере современного видеостримингового IT-бизнеса.

Большинство подобных площадок уже предлагает своё решение данного вопроса в качестве чата по умолчанию, но такие чаты не предоставляют возможность использовать потоковый обмен сообщениями, как отдельный модуль. Цель создания веб-сервиса – возможность интегрировать компонент обмена сообщениями в различные видеостриминговые площадки не привязывая функционал к платформе.

Для организации хранения сообщений пользователей выбрана облачная платформа Firebase. Это бесплатная платформа для разработки Unity, Android, iOS и Web приложений от компании Google. Firebase выступает в роли сервера, который запускается и работает на удалённой инфраструктуре Google. Одним из главных преимуществ этой платформы является наличие собственной СУБД под названием Firebase Realtime Database. Это облачная NoSQL база данных, которая

позволяет хранить и синхронизировать данные между клиентами в режиме реального времени.

При подключении Firebase Realtime Database к проекту между устройством клиента и серверами Firebase устанавливается связь по специальному протоколу WebSocket, что обеспечивает постоянное соединение в режиме реального времени для синхронизации сообщений и других необходимых данных. Так как в проекте предполагается наличие аккаунтов пользователей, то дополнительно добавлен авторизационный метод входа, используя сервисы Google в разделе Firebase Authentication.

Сервис поддерживает отправку исключительно текстовых сообщений длиной не более трёхсот символов. Для реализации бизнес логики была разработана UML диаграмма всех сущностей (классов) программного комплекса, которая включает 12 классов, каждый класс разработан для решения определенной задачи (рисунок 1).

Класс Auth выполняет функции авторизации в проекте. Класс хранит в себе информацию о текущем пользователе (UID или user ID, его email и ссылку на картинку профиля). Класс работает с провайдером авторизации Goggle, а именно Goggle Play Services Auth.

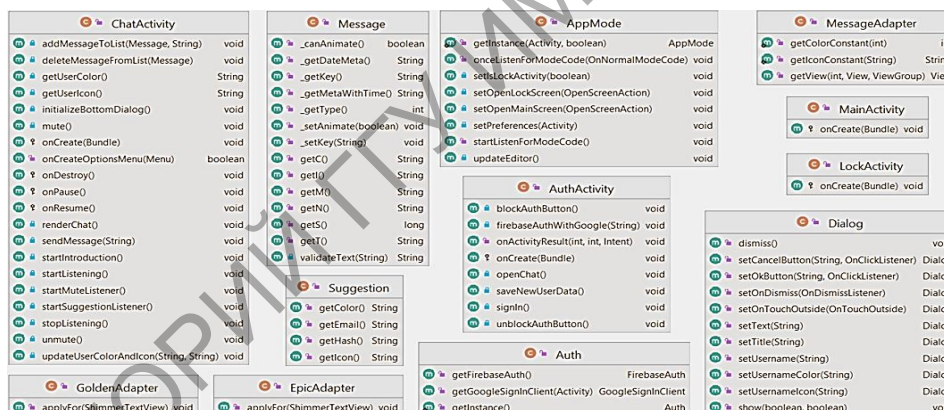


Рисунок 1 – UML диаграмма Java классов

Класс AppMode содержит информацию о текущем состоянии проекта. Находится приложение в штатном режиме, или в режиме технических работ. Как только сервер сообщает клиенту о необходимости активации или снятия состояния технических работ, происходит отображение или скрытие экрана блокировки.

Класс Dialog служит для отображения всплывающего диалогового окна, чтобы получить от пользователя обратную связь по необходимости.

Классы Message и Suggestion описывают сущность «Сообщение» в проекте, а также создают объект для хранения сообщения в базе данных.

Класс MainActivity описывает логику главного экрана приложения. Главный экран в данном приложении – это загрузочный экран, на котором изображён логотип приложения. Этот класс проверяет режим приложения и авторизацию пользователя, взаимодействуя с сервером. В зависимости от ответов сервера, класс открывает экран блокировки, экран авторизации (входа) или домашний экран с чатом.

Класс MessageAdapter выступает в роли соответствующего адаптера, так как адаптирует содержимое коллекции ArrayList<Message> для вывода данных в элемент ListView на экране мобильного устройства.

Классы GoldenAdapter и EpicAdapter, выступают в роли соответствующих адаптеров, которые преобразуют обычный текст, предназначенный для элемента TextView, для вывода в элемент ShimmerTextView с различными визуальными эффектами.

Класс AuthActivity описывает логику экрана авторизации (входа), предоставляя возможность входа с помощью провайдера авторизации Google Play Services Auth.

Класс LockActivity описывает логику экрана блокировки. В качестве блокировки выступает режим проведения технических работ на сервере. Реализацию этого класса обеспечивает класс AppMode.

Класс ChatActivity описывает логику домашнего экрана приложения с чатом. Он выполняет функции отображения сообщений, отправки новых сообщений, выдаёт пользователю предложения о новой настройке, контролирует ограничения отправки сообщений, предоставляет пользователю возможность выхода из аккаунта и контролирует текущее состояние приложения с помощью класса AppMode.

Таким образом, был разработан чат-сервис для потокового обмена сообщениями в режиме реального времени, который интегрируется в любое потоковое онлайн-вещание, например, на Twitch или YouTube, что, возможно, может послужить неплохой альтернативой стандартному чату любой из этих видеостриминговых площадок.

Д. Г. Логунов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. И. Сукач**, канд. техн. наук, доцент

АВТОМАТИЗАЦИЯ ОСНОВНЫХ ФУНКЦИЙ РАБОТЫ РАЙОННОГО ОТДЕЛЕНИЯ ГАИ

Информация является ключевым элементом в деятельности сотрудников районного отделения ГАИ при выполнении следующих