

Р. Р. Фиткулина^{1✉}, Н. С. Головачев¹

Разработка серверного модуля для поддержки сетевых функций игры «BlushBound»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ruslanafit04@gmail.com

Аннотация. В статье описывается процесс реализации серверной части для игрового продукта «BlushBound», в котором объединены элементы визуальной новеллы и системы накопления уникальных наград. Реализация синхронизации прогресса пользователя, долговременного хранения информации о пользователях и внедрения сетевых. Рассмотрены архитектура серверного блока, схема БД, а также алгоритмы API для входа в систему, синхронизации достижений, управления инвентарем и механики получения уникальных наград. Отдельно рассмотрены вопросы сохранения целостности данных через транзакции и валидации запросов. Проведены ручные, автоматические и нагрузочные тесты. Итоги подтверждают устойчивую работу модуля при последовательных вызовах и корректную реализацию игровых сценариев. Результаты могут быть востребованы при создании инди-игр с клиент-серверной моделью и небольшим бюджетом.

Ключевые слова: серверный модуль, клиент-серверная архитектура, видеоигра, PHP, MySQL, Unity

R. R. Fitkulina^{1✉}, N. S. Golovachev¹

Development of a server module for supporting network functions in the game «BlushBound»

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ruslanafit04@gmail.com

Abstract. The article describes the process of implementing the server-side component for the game project “BlushBound,” which combines elements of a visual novel and a unique reward collection system. The implementation includes user progress synchronization, long-term storage of user data, and the integration of network interaction features. The paper examines the architecture of the server module, the database schema, and the API algorithms for user authentication, achievement synchronization, inventory management, and unique reward mechanics. Particular attention is paid to maintaining data integrity through transactions and request validation. Manual, automated, and load testing were conducted. The results confirm the stable operation of the module under sequential requests and the correct implementation of gameplay scenarios. The obtained results may be useful in the development of indie games with a client-server architecture and a limited budget.

Keywords: server module, client-server architecture, video game, PHP, MySQL, Unity

Введение

В настоящее время игровые продукты все чаще отходят от модели исключительно локального хранения информации, переходя к использованию внешних серверных систем [1–3]. Даже те проекты, которые не задуманы как многополь-

зовательские, нуждаются в механизмах записи прогресса, отслеживания инвентаря и контроля внутриигровых ресурсов. Причина – как в необходимости защищать данные пользователей, так и в массовом распространении игр, построенных на коллекционировании предметов и персонажей [4, 5].

Для небольших команд разработчиков развертывание собственной серверной инфраструктуры – нетривиальная инженерная задача. Облачные сервисы «из коробки» ускоряют старт, но почти всегда сопряжены с регулярными платежами и мешают гибко менять внутреннюю логику [6, 7]. Поэтому при создании небольших игр сохраняет актуальность самостоятельная разработка серверных модулей, которые обеспечивают базовый сетевой функционал без привлечения платных внешних платформ [8, 9].

В данной работе речь идет о создании серверной составляющей для игры «BlushBound». Проект комбинирует признаки рогалика, визуального романа и системы случайного получения лута. Подобная структура подразумевает хранение больших объемов пользовательской информации: пройденные уровни, состояние аватаров, наполнение инвентаря, статистика сессий.

Ключевая цель исследования – реализовать такой серверный модуль, который через API-интерфейс соединяет клиентскую часть игры с базой данных [10]. Параллельно решались задачи по организации входа пользователей, обработке игровых вызовов и предотвращению разрушения целостности данных при параллельных операциях [5, 11].

Методы и материалы

На этапе предпроектного анализа перебирались разные варианты построения бэкенда: Cloud-решения Firebase, PlayFab и другие. Однако внедрение этих платформ ведет к росту эксплуатационных расходов и ограничивает возможность доработки игровой логики [7]. В итоге выбор остановлен на самостоятельной разработке серверного компонента с использованием PHP и MySQL [12, 13].

Бэкенд построен по классической схеме REST API [10]. Данные между клиентом и сервером передаются через HTTP-запросы, ответы формируются в JSON. Такое решение сильно упрощает сопряжение с клиентской частью Unity и отделяет логику клиента от деталей внутреннего устройства базы данных [14, 15].

Функционально серверная часть разделена на несколько крупных блоков: регистрация и аутентификация, загрузка и сохранение прогресса, управление инвентарем, проведение игровых транзакций, а также реализация механики случайной выдачи награды.

На рис. 1 представлена обобщенная архитектура взаимодействия пользователя, клиентского приложения на Unity и разработанного серверного модуля. Пользователь через клиентскую часть инициирует такие действия, как регистрация, авторизация, прохождение уровней, открытие предметов и управление инвентарем. Клиентское приложение, в свою очередь, формирует соответствующие запросы к серверному модулю: отправка боевых действий, запрос на открытие предметов, запрос инвентаря, экипировки, улучшений, профиля и заверше-

ния сессии. Серверный модуль обрабатывает каждый из этих запросов – от регистрации и авторизации до отдачи статических страниц, обработки боя, механики получения предметов, предоставления инвентаря, экипировки, улучшения, профиля и завершения сессии.

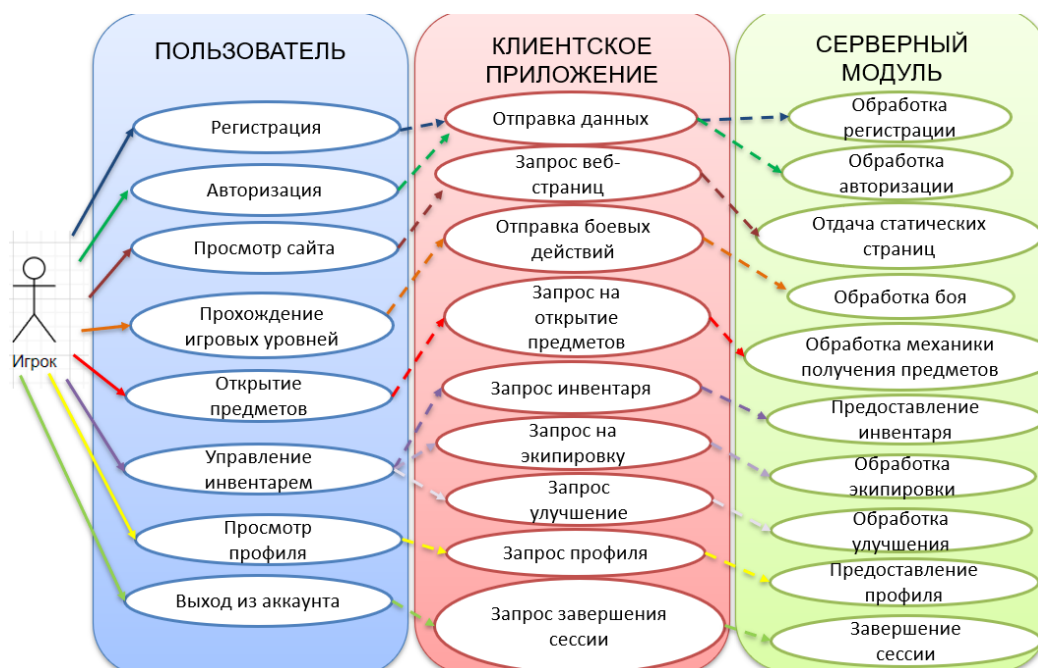


Рис. 1. Схема взаимодействия пользователя, клиентского приложения и серверного модуля

Каждый API-метод реализован в виде отдельного PHP-скрипта. На рис. 2 приведен сокращенный пример обработчика авторизации.

Данный фрагмент демонстрирует основные этапы аутентификации: получение учетных данных, подготовленный запрос к базе данных для предотвращения SQL-инъекций, верификацию пароля через `password_verify`, генерацию криптостойкого сессионного токена и его сохранение. Такой подход обеспечивает базовый уровень безопасности [5]. Пароли хранятся исключительно в хешированном виде, что исключает их компрометацию при возможной утечке базы данных [4].

Наибольшее внимание при кодировании было уделено операциям изменения игровой валюты и выдачи предметов. Такие действия обернуты в SQL-транзакции [11, 16]. Это гарантирует, что данные останутся непротиворечивыми даже при сбоях в ходе обработки запроса.

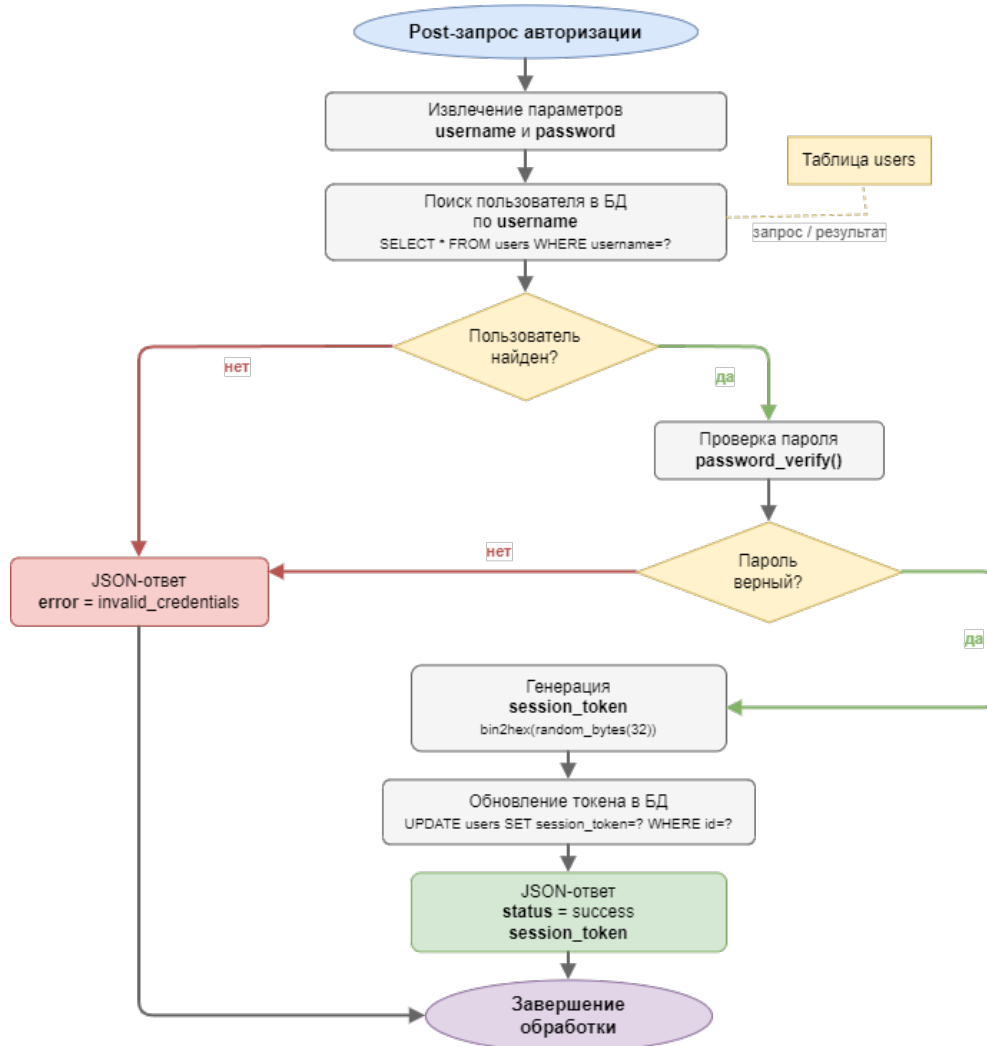


Рис. 2. Схема API-метода обработки авторизации

Результаты

Итогом работы стал полноценный серверный модуль, покрывающий все основные сетевые задачи игрового проекта «BlushBound». Серверная часть обеспечивает регистрацию и последующую авторизацию игроков, хранение и синхронизацию игрового прогресса, а также предоставление данных о содержимом инвентаря, и обработку событий, связанных с выдачей наград для развития персонажей пользователей.

В процессе тестирования серверный модуль корректно обработал последовательный набор запросов, поступавших от клиентской части, реализованной в среде разработки Unity [14, 17]. Проверке подверглись сценарии, отвечающие за создание учетных записей и повторный вход пользователей в систему, сохранение игрового процесса и получение информации из базы данных.

Особое внимание в процессе проверки было уделено механизму случайной выдачи предметов. Сервер самостоятельно производит вычисление результата, не привлекая к этой операции клиентскую сторону, после чего обновляет данные учетной записи игрока и направляет готовый ответ обратно клиенту [5].

Дополнительно оценивалась устойчивость API к некорректно сформированным вызовам. Сервер правильно обрабатывает ситуации, когда токен сессии передан неправильно, когда в запросе отсутствует обязательный параметр, а также когда значение игровой валюты указано в неверном формате.

Нагрузочное тестирование подтвердило, что при последовательной обработке серий запросов модуль функционирует стабильно, не допуская потери информации и не порождая критических сбоев в работе [18].

Обсуждение

Полученное решение демонстрирует потенциал простого стека технологий, для реализации серверной инфраструктуры, полностью покрывающую нужды инди-игры с элементами онлайн-взаимодействия [8, 12]. Для проектов с небольшим бюджетом это особенно критично, поскольку расходы на коммерческие облачные платформы растут вместе с увеличением числа игроков [7].

Плюс выбранного подхода – полный контроль над схемой БД и логикой обработки запросов [19]. В отличие от универсальных платформ, разработанный серверный модуль заточен именно под задачи «BlushBound» и содержит только необходимые функции.

При этом важно отметить, что текущая версия рассчитана на небольшую аудиторию. При многократном росте числа одновременных подключений потребуются доработки – добавление кэширования, балансировщиков нагрузки и асинхронной обработки [18, 20].

Заключение

В ходе выполнения работы создан серверный модуль для игрового приложения «BlushBound», который выполняет базовые сетевые функции, надежно хранит пользовательские данные, связывает клиент Unity с сервером через API, а также обеспечивает авторизацию пользователей, обработку запросов и синхронизацию игрового прогресса [10, 14].

Для хранения всей информации в разработанном решении используется реляционная база данных MySQL [16]. Все критически важные операции реализованы с применением механизма транзакций [11].

Проведенные тесты подтвердили корректность работы всех функций модуля и его устойчивость к ошибочным запросам [5, 18]. Полученные итоги работы демонстрируют, что полноценную серверную инфраструктуру для инди-проекта можно создать без использования сложных и дорогостоящих облачных платформ.

В числе ближайших направлений дальнейшего развития проекта можно выделить оптимизацию производительности, расширение аналитической подсистемы, а также внедрение инструментов масштабирования серверной части.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Newzoo. Global Games Market Report 2025. – Режим доступа: <https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2025-free-version> (дата обращения: 12.05.2026).

2. Смирнов А. В. Безопасность веб-приложений / А. В. Смирнов. – Санкт-Петербург: Питер, 2021. – 288 с. – ISBN 978-5-4461-1500-6.
3. Атлас А. PHP для профессионалов / А. Атлас. – Санкт-Петербург: Питер, 2023. – 480 с. – ISBN 978-5-4461-1567-3.
4. Кузнецов М. В., Симдянов И. В. PHP. Практика разработки Web-сайтов / М. В. Кузнецов, И. В. Симдянов. – 3-е изд. – Санкт-Петербург: БХВ-Петербург, 2022. – 560 с. – ISBN 978-5-9775-6839-9.
5. Кручинин А. В. REST API. Создание и использование / А. В. Кручинин. – Москва: ДМК Пресс, 2021. – 304 с. – ISBN 978-5-97060-907-1.
6. Unity Documentation: руководство по WebRequest API. – Режим доступа: <https://docs.unity3d.com/Manual/UnityWebRequest.html> (дата обращения: 12.05.2026).
7. Unity. Web request high-level API reference. – Режим доступа: <https://docs.unity3d.com/6000.4/Documentation/Manual/web-request-hlapi.html> (дата обращения: 12.05.2026).
8. Демидов С. В. Unity и C#. Геймдев от идеи до реализации / С. В. Демидов. – Санкт-Петербург: Питер, 2023. – 928 с. – ISBN 978-5-4461-0715-5.
9. Бенкен Е. С. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен. – 5-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2022. – 672 с. – ISBN 978-5-9775-6812-2.
10. Duckett J. PHP & MySQL: серверная веб-разработка / Дж. Дакет; пер. с англ. А. В. Логвинова. – Москва: Эксмо, 2025. – 720 с. – ISBN 978-5-04-201405-9.
11. Зандстра М. PHP: объекты, шаблоны и методики программирования / М. Зандстра. – Москва: Вильямс, 2022. – 560 с. – ISBN 978-5-8459-1564-7.
12. Мартин Р. С. Чистая архитектура. Искусство разработки программного обеспечения / Р. С. Мартин. – Санкт-Петербург: Питер, 2022. – 352 с. – ISBN 978-5-4461-1627-0.
13. Грофф Дж., Вайнберг П., Оппель Э. SQL. Полное руководство / Дж. Грофф, П. Вайнберг, Э. Оппель. – 3-е изд. – Москва: Вильямс, 2021. – 1024 с. – ISBN 978-5-8459-1920-1.
14. Орлов С. А. Технологии разработки программного обеспечения / С. А. Орлов. – 6-е изд. – Санкт-Петербург: Питер, 2024. – 624 с. – ISBN 978-5-4461-1129-9.
15. Docker Documentation. – Режим доступа: <https://docs.docker.com/> (дата обращения: 12.05.2026).
16. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Санкт-Петербург: Питер, 2021. – 368 с. – ISBN 978-5-4461-1203-6.
17. MySQL: официальная документация. – Режим доступа: <https://dev.mysql.com/doc/refman/8.0/en/> (дата обращения: 12.05.2026).
18. Петров А. Н. Разработка игр на Unity / А. Н. Петров. – Москва: Эксмо, 2023. – 416 с. – ISBN 978-5-04-170421-6.
19. Татро К., Макинтайр П. Создаем динамические веб-сайты на PHP / К. Татро, П. Макинтайр. – Санкт-Петербург: Питер, 2021. – 544 с. – ISBN 978-5-4461-1509-9.
20. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – Санкт-Петербург: Питер, 2020. – 352 с. – ISBN 978-5-4461-0816-9.

© Р. Р. Фиткулина, Н. С. Головачев, 2026