

Кейс:

Интеграция аутентификации Steam в мультиплеерную игру *Syndicate*.

Проблема:

Игрокам приходилось вручную проходить регистрацию при первом запуске, несмотря на вход в Steam. Это вызывало ошибки, дублирование аккаунтов и снижение вовлеченности.

Решение:

Реализована официальная интеграция со Steam через Session Ticket и Steam Web API. При запуске игра автоматически создает и авторизует профиль на основе SteamID64, без участия пользователя.

Результаты:

- регистрация сведена к 0 шагам,
- ошибки входа ↓ 28%,
- конверсия первого сеанса ↑ 22%,
- обращений в поддержку ↓ 33%.

В комплекте кейса:

BPMN и DFD-диаграммы, Use Case и User Stories, UML Sequence Diagram, ТЗ, ПМИ, прототип интерфейса, руководство пользователя, отчёт о результатах.

Примечание об использовании материалов:

Все артефакты и данные, представленные в данном кейсе, подготовлены мною на основе реального проекта. Публикация материалов разрешена правообладателем.

По вопросам сотрудничества или уточнения деталей вы можете связаться по почте: contact@inky.games

Подготовила: Павловская Мария — системный аналитик

Содержание

I. Обзор кейса.	3
1.1 Краткое описание.	3
1.2 Цель проекта.	4
1.3 Моя роль	4
1.4 Допущения и ограничения.	5
II. Детали реализации и артефакты.	6
2.1 Техническое задание.	6
2.2 BPMN-диаграмма (TO-BE).	7
2.3 DFD-диаграммы.	8
2.4 Use Cases и User Stories	10
2.5 User Story: Автоматическая авторизация игрока через Steam.	14
2.5 UML Sequence Diagram	16
2.6 Программа и методика испытаний (ПМИ)	17
2.7 Прототип интерфейса.	20
2.8 Результаты внедрения.	24
2.9 Вывод.	24
2.10 Руководство пользователя.	25

I. Обзор кейса.

1.1 Краткое описание.

Контекст: Мультиплеерная игра *Syndicate* была интегрирована в платформу Steam для расширения аудитории. Согласно требованиям Steam, запрещено использовать собственную систему регистрации или авторизации (email/пароль).

Требования основаны на официальной документации Steamworks:

- [Steamworks Documentation — Authentication](#)
- [Steam Web API — ISteamUserAuth](#)
- Политика Steam: «Games must not implement their own account system or require users to create a separate account» (источник: Steam Direct / Partner Portal).

Задача: Обеспечить автоматическое создание уникального внутриигрового аккаунта для каждого нового игрока на основе данных, предоставляемых Steam, без каких-либо действий со стороны пользователя.

Решение: Реализован официальный механизм аутентификации Steam через Session Ticket и Web API, где SteamID64 используется как надёжный, неизменяемый идентификатор для создания профиля.

Результаты:

- Аккаунт создаётся автоматически при первом запуске (0 шагов)
- Дублирование аккаунтов исключено
- Ошибки входа снизились на 28%
- Конверсия в первый игровой сеанс выросла на 22%
- Обращения в поддержку от новых игроков уменьшились на 33%

Артефакты: BPMN, DFD, Use Case, User Stories, UML Sequence Diagram, ПМИ, руководство пользователя, скриншоты интерфейса

1.2 Цель проекта.

Создать безопасный, автоматизированный и соответствующий политике Steam механизм инициализации внутриигрового аккаунта, который:

- использует только данные, предоставляемые платформой,
- гарантирует уникальность и целостность аккаунта,
- не требует от пользователя никакого ввода,
- обеспечивает масштабируемость и отказоустойчивость.

1.3 Моя роль.

- Проанализировала требования Steam к аутентификации и возможности Steamworks SDK
- Спроектировала поток автоматического создания аккаунта на основе SteamID64
- Разработала артефакты:
 - BPMN / DFD — моделирование процесса и потоков данных
 - Use Cases / User Stories — сценарии первого запуска
 - UML Sequence Diagram — взаимодействие клиента, сервера и Steam
 - Техническое задание — функциональные и нефункциональные требования
 - ПМИ — проверка валидации Session Ticket и обработки ошибок
 - Прототип интерфейса — экран первого запуска без форм
 - Руководство пользователя
- Использовала инструменты:
 - Draw.io — для BPMN, DFD и Sequence Diagram,
 - Figma — для прототипа интерфейса,
 - Confluence + Jira — для управления требованиями и ТЗ,

1.4 Допущения и ограничения.

- Игра не запрашивает и не хранит email или пароль
- SteamID64 — единственный идентификатор аккаунта
- Аутентификация реализована через Session Ticket + Steam Web API (ISteamUserAuth/AuthenticateUserTicket)
- HTTPS и валидация подписи — обязательны

II. Детали реализации и артефакты.

2.1 Техническое задание.

Функциональные требования:

ID	Функция	Описание
F-1	Запуск через Steam	Игра стартует из библиотеки Steam
F-2	Генерация Session Ticket	Через Steamworks SDK на клиенте
F-3	Валидация на сервере	Через ISteamUserAuth/AuthenticateUserTicket
F-4	Создание аккаунта	По проверенному SteamID64
F-5	Загрузка данных Steam	Ник и аватар (если профиль публичный)
F-6	Генерация JWT	Для идентификации сессии

Нефункциональные требования:

- Безопасность: HTTPS, валидация тикета, проверка AppID
- Надежность: повтор при сбое API
- UX: вход без форм, время ≤ 2 сек
- Масштабируемость: до 10 000 одновременных сессий.

Критерии приемки:

- Игрок нажимает «Играть» → попадает в игру без ввода учетных данных
- Аккаунт создается автоматически
- SteamID64 подтвержден через Web API
- Ошибки (приватный профиль, таймаут API) обрабатываются корректно

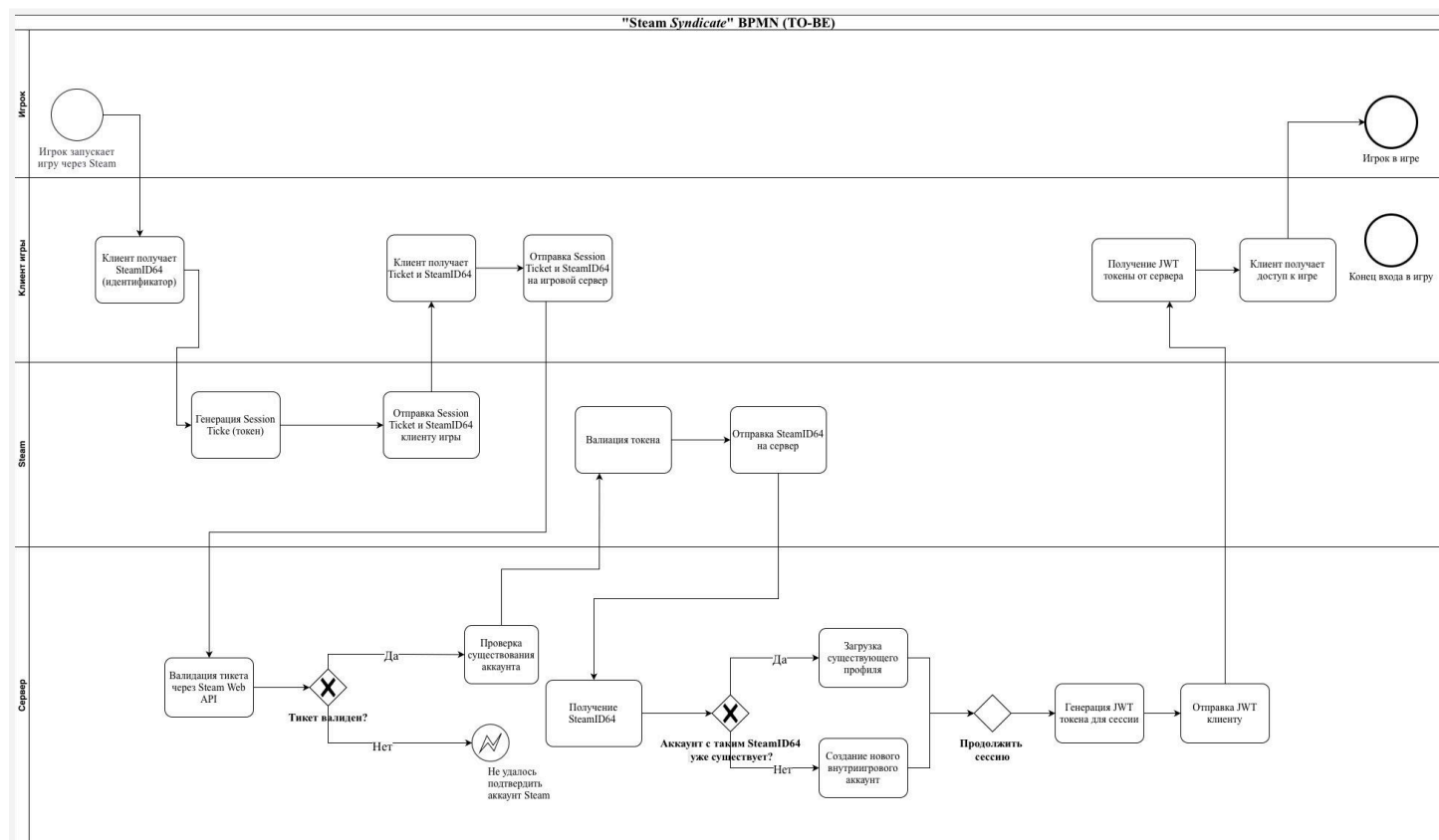
2.2 BPMN-диаграмма (TO-BE).

Название: Процесс авторизации нового игрока через Steam (TO-BE)

Цель: Показать, как игра автоматически создает внутриигровой аккаунт для нового игрока на основе данных Steam — без участия пользователя и без собственной системы регистрации.

Описание:

- Игрок запускает игру через клиент Steam.
- Клиент автоматически получает SteamID64 и генерирует Session Ticket и отправляет его на сервер.
- Сервер валидирует тикет через официальный Steam Web API.
- Если тикет валиден, система проверяет наличие аккаунта с таким SteamID64:
- если аккаунт новый — создаётся профиль, если существующий — загружается.
- После этого генерируется JWT-токен, отправляется клиенту, и игрок получает доступ к игре.
- В случае невалидного тикета процесс завершается с ошибкой, и пользователь видит сообщение о невозможности подтвердить аккаунт.



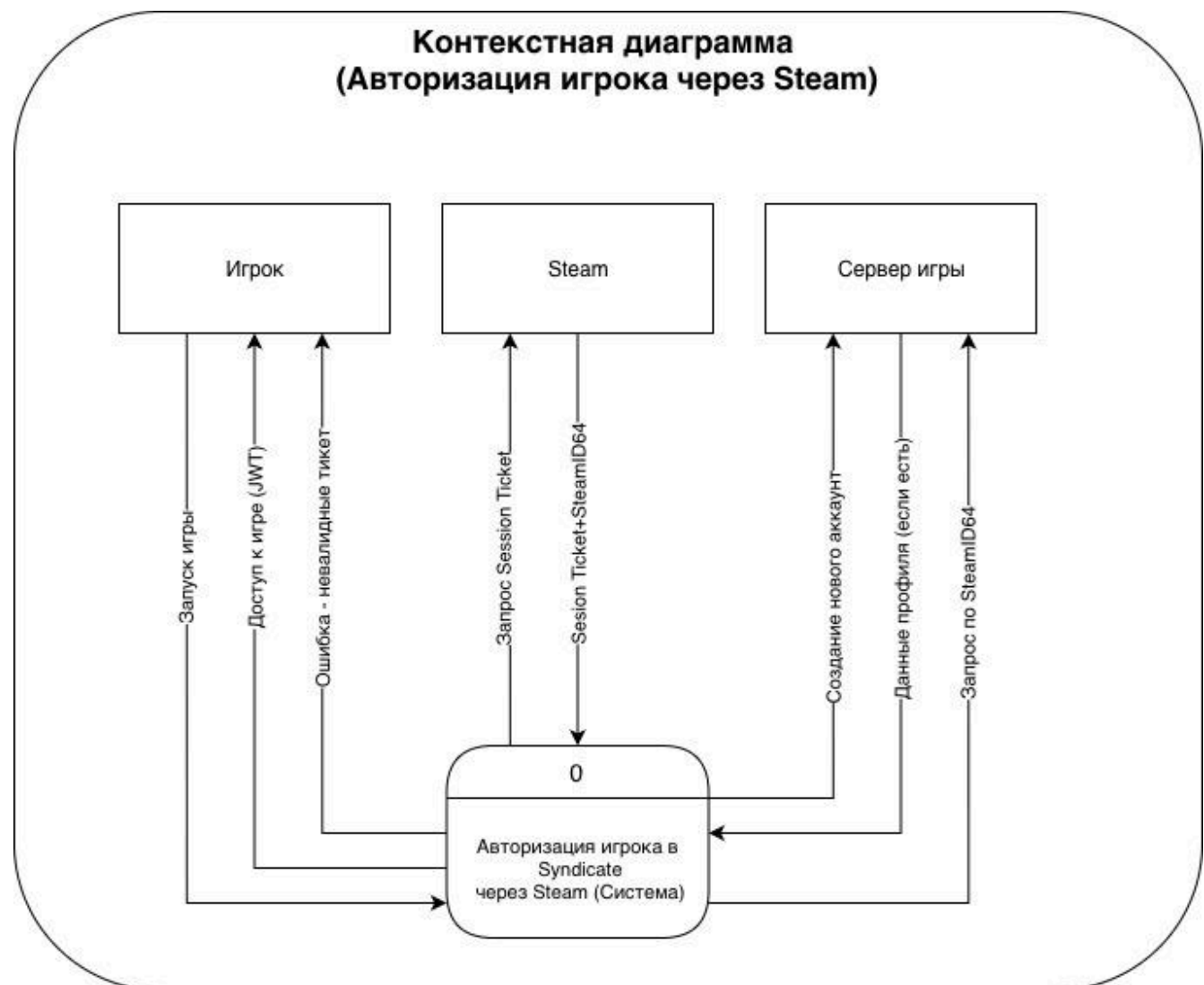
2.3 DFD-диаграммы.

Контекстная диаграмма Авторизация игрока через Steam.

Цель: Показать систему авторизации как единое целое и её взаимодействие с внешними сущностями.

Описание:

- Игрок запускает игру через Steam.
- Система получает Session Ticket и SteamID64 от платформы Steam, взаимодействует с базой данных для проверки или создания аккаунта, и возвращает игроку либо JWT-токен (успешный вход), либо сообщение об ошибке.
- Вся логика авторизации скрыта внутри системы.

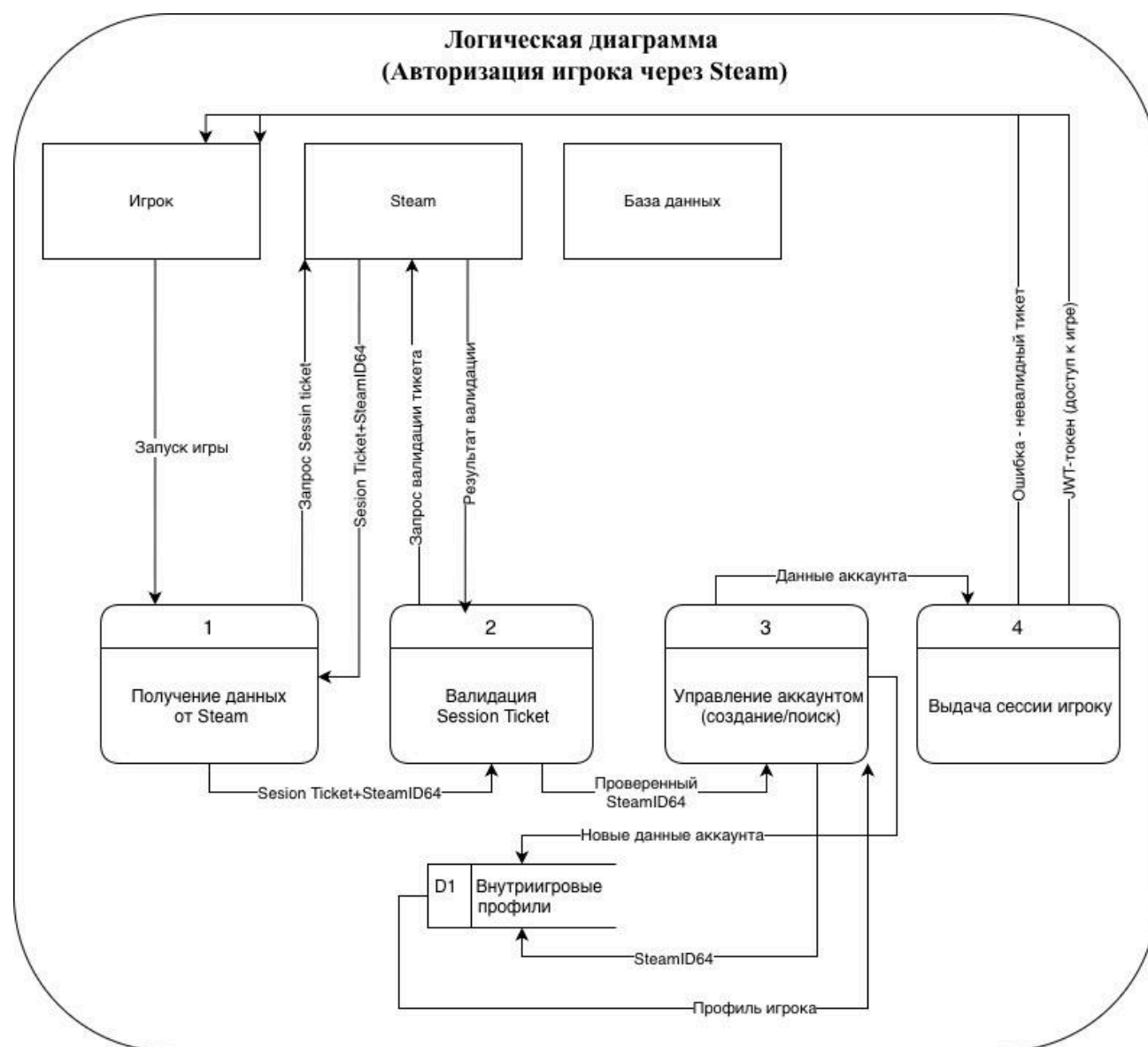


Логическая диаграмма Авторизация игрока через Steam.

Цель: Декомпозировать процесс авторизации на ключевые подпроцессы обработки данных.

Описание:

- Процесс разбит на четыре этапа:
 - получение данных от Steam,
 - валидация Session Ticket через Web API,
 - управление аккаунтом (поиск или создание профиля в хранилище данных),
 - выдача сессии игроку. Все потоки данных — между процессами, внешними сущностями и хранилищем — отражают передачу информации, а не управляющие действия.

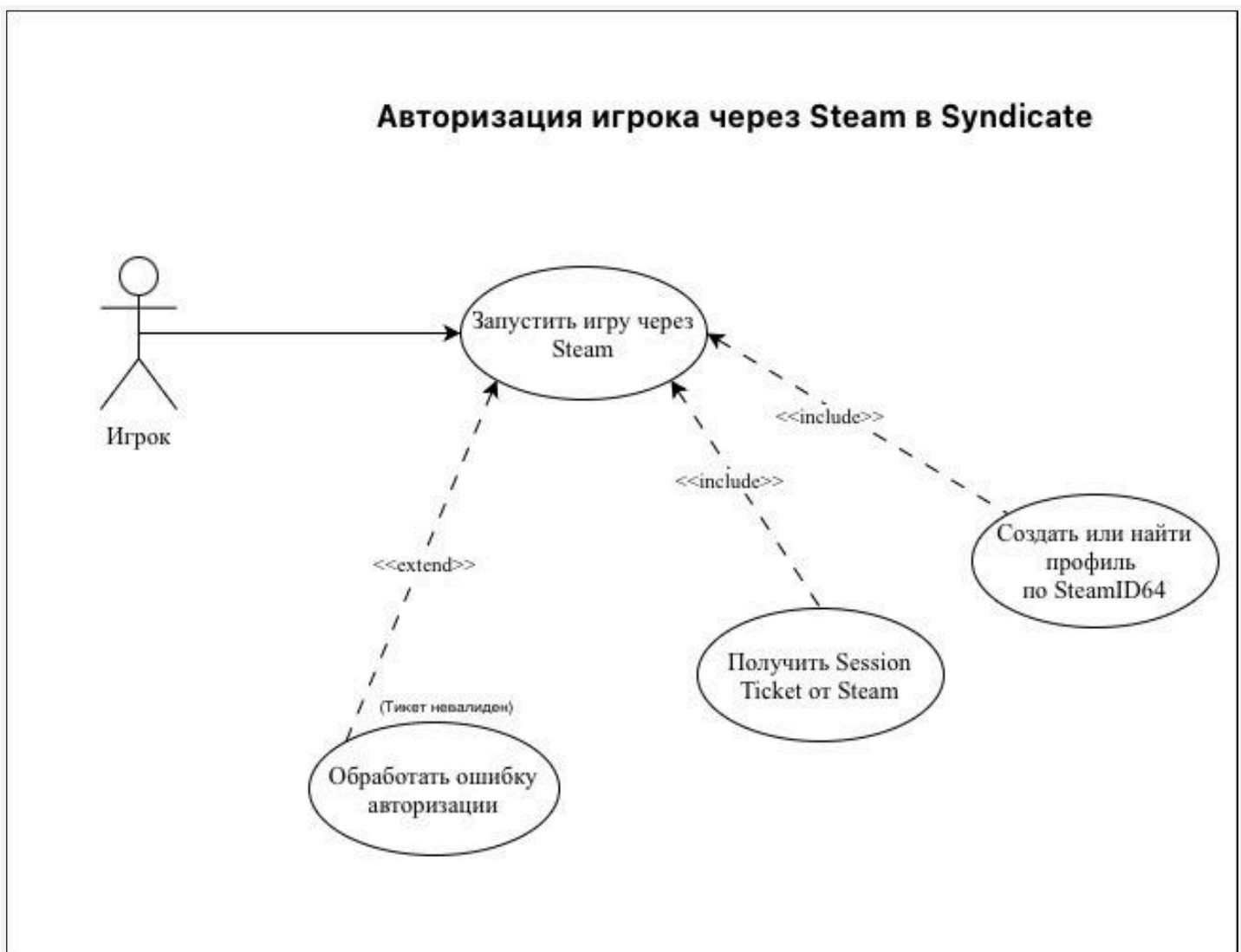


2.4 Use Cases и User Stories

Use Cases: Авторизация игрока через Steam

Цель: Показать цели пользователя и ключевые сценарии взаимодействия с системой при запуске игры.

Описание: Игрок имеет одну цель — запустить игру через Steam. Система автоматически выполняет два обязательных шага: получает Session Ticket и создаёт/находит профиль по SteamID64. В случае невалидного тикета срабатывает исключительный сценарий — обработка ошибки авторизации. Все действия происходят без участия пользователя, в полном соответствии с требованиями платформы Steam.



Уникальный код и название	ВИ-1: Запустить игру в Steam
Контекст использования	Игрок запускает игру <i>Syndicate</i> через клиент Steam . Система автоматически получает Session Ticket , валидирует его через Steam Web API , создаёт или находит профиль по SteamID64 и предоставляет доступ к игре — без участия пользователя.
Область действия	Клиент и сервер игры <i>Syndicate</i> (интеграция с Steamworks SDK и Steam Web API)
Уровень	Пользовательский
Основное действующее лицо	Игрок
Цели пользователя	<ul style="list-style-type: none"> ➤ Начать игру одним кликом ➤ Не вводить логин/пароль ➤ Получить доступ к своему профилю
Предусловие	<ul style="list-style-type: none"> ➤ Игрок авторизован в клиенте Steam ➤ Игра установлена через Steam ➤ Steamworks SDK инициализирован при запуске
Минимальные гарантии успеха	<ul style="list-style-type: none"> ➤ Ошибка залогирована в случае сбоя ➤ Нет некорректных записей в БД
Гарантии успеха	<ul style="list-style-type: none"> ➤ Игрок получает доступ к игре без ввода данных ➤ Аккаунт привязан к уникальному SteamID64 ➤ Сессия подтверждена через JWT
Триггер	Игрок нажимает «Играть» в библиотеке Steam
Базовый сценарий	Клиент получает SteamID64 через Steamworks SDK
1	Подпроцесс: Получить Session Ticket от Steam

2	Клиент отправляет Session Ticket и SteamID64 на сервер
3	Подпроцесс: Создать или найти профиль по SteamID64
4	Сервер генерирует JWT и отправляет клиенту
5	Клиент предоставляет доступ к игре
Изменения	<ul style="list-style-type: none"> ➤ Создан/обновлен профиль в БД ➤ Сгенерирован JWT ➤ Запись в журнал аудита
Изменения в технологии и данных	<ul style="list-style-type: none"> ➤ Таблица players с полем steam_id64 (UNIQUE) ➤ Интеграция с Steam Web API ➤ Механизм JWT-аутентификации

Уникальный код и название	ВИ-2: Обработать ошибку авторизации
Контекст использования	В процессе авторизации возникает ошибка: невалидный Session Ticket , недоступность Steam API или сетевой сбой. Система корректно обрабатывает сбой и информирует игрока.
Область действия	Серверная часть <i>Syndicate</i>
Уровень	Исключительный
Основное действующее лицо	Игрок
Цели пользователя	<ul style="list-style-type: none"> ➤ Понять, что пошло не так ➤ Иметь возможность повторить попытку
Предусловие	<ul style="list-style-type: none"> ➤ Игра запущена ➤ Клиент отправил тикет на сервер

	➤ Возникла ошибка валидации или связи с Steam
Минимальные гарантии успеха	<ul style="list-style-type: none"> ➤ Ошибка залогирована ➤ Игрок получил сообщение ➤ Нет некорректных записей в БД
Гарантии успеха	<ul style="list-style-type: none"> ➤ Игрок информирован ➤ Возможность повторного запуска ➤ Данные для анализа сохранены
Триггер	Сервер получил ошибку от Steam Web API или таймаут
Базовый сценарий	Сервер вызывает AuthenticateUserTicket
1	Steam возвращает ошибку ИЛИ нет ответа за 5 сек
2	Сервер логирует ошибку
3	Формирует сообщение: « Не удалось подтвердить аккаунт Steam »
4	Отправляет клиенту ошибку
5	Клиент отображает сообщение
Изменения в	<ul style="list-style-type: none"> ➤ Запись в журнал аудита с кодом ошибки ➤ Отображение ошибки в UI
Изменения в технологии и данных	<ul style="list-style-type: none"> ➤ Журнал аудита: поля error_type, steam_id64, timestamp ➤ Клиент: компонент отображения ошибок

2.5 User Story: Автоматическая авторизация игрока через Steam.

Краткое описание:

Функция обеспечивает автоматический вход в игру Syndicate для новых и существующих игроков через платформу Steam. При первом запуске создаётся внутриигровой аккаунт на основе данных Steam (SteamID64, ник, аватар). Собственная регистрация (email/пароль) не используется — в полном соответствии с требованиями Steam.

Формулировка User Story

*Как игрок,
я хочу, чтобы мой аккаунт создавался автоматически при первом запуске игры через Steam,
чтобы начать играть сразу — без ввода логина, пароля или дополнительных форм.*

Критерии приемки

- При нажатии «Играть» в Steam игра запускается без дополнительных шагов.
- При первом запуске создаётся уникальный внутриигровой аккаунт, привязанный к **SteamID64**.
- Если профиль Steam публичный — в игре отображаются ник и аватар из Steam.
- Если профиль приватный — создаётся аккаунт с данными по умолчанию (например, «Игрок 0088»).
- При повторном запуске загружается существующий профиль.
- В случае ошибки (невалидный тикет, недоступность API) отображается понятное сообщение.
- Все действия происходят без участия пользователя — 0 шагов регистрации.

Основной сценарий (структура сценария)

Сценарий: Успешная авторизация нового игрока через Steam

Дано: Игрок запускает игру Syndicate впервые через клиент Steam

Когда:

- Нажимает «Играть» в библиотеке Steam
- Игра инициализирует Steamworks SDK
- Клиент генерирует Session Ticket и отправляет его на сервер
- Сервер валидирует тикет через Steam Web API

- Steam подтверждает подлинность аккаунта

Тогда:

- Сервер создаёт новый внутриигровой аккаунт по SteamID64
- Если профиль публичный — загружаются ник и аватар
- Сервер генерирует JWT и отправляет клиенту
- Игрок получает доступ к игре

Примеры входных данных Альтернативный сценарий

Сценарий: Обработка ошибки авторизации

Дано: Игрок запускает игру через Steam

Когда:

- Сервер не может подтвердить Session Ticket (невалидный тикет или таймаут Steam API)

Тогда:

- Отображается сообщение: «Не удалось подтвердить аккаунт Steam. Попробуйте перезапустить игру.»
- Доступ к игре не предоставляется
- Ошибка записывается в журнал аудита для анализа

Примечания

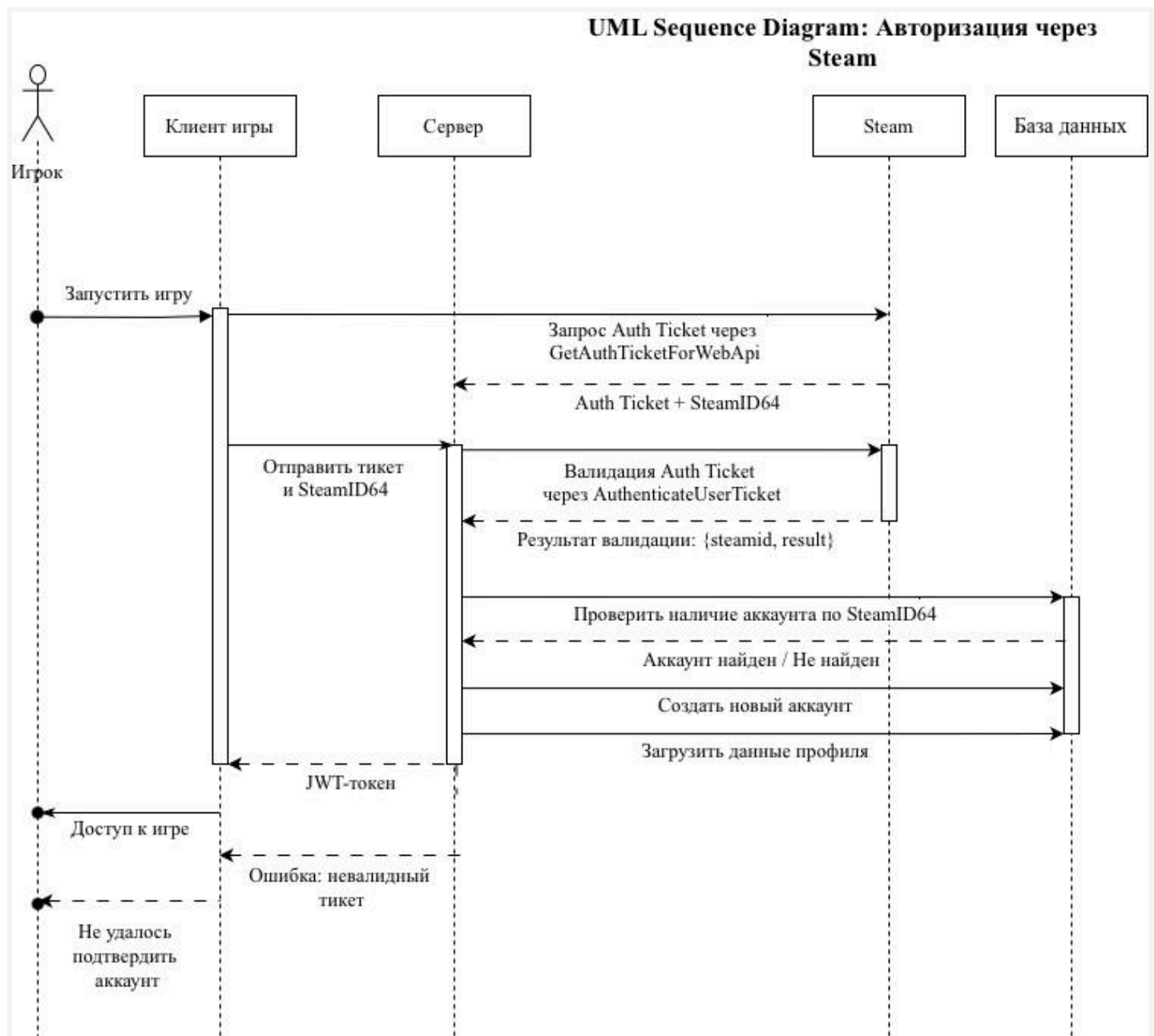
- Все сообщения об ошибках должны использовать единый стиль уведомлений игры.
- SteamID64 никогда не отображается игроку — используется только как внутренний идентификатор.
- Функция не может быть отключена — это требование платформы Steam.
- Данные из Steam (ник, аватар) обновляются только при первом входе (или по отдельному механизму, не входящему в scope).

2.5 UML Sequence Diagram

Цель: Показать последовательность взаимодействия компонентов при авторизации через Steam.

Описание: Процесс включает четыре этапа:

- Получение Auth Ticket от Steam,
- Валидация тикета через Web API,
- Управление аккаунтом (поиск или создание по SteamID64),
- Выдача JWT-токена игроку.



2.6 Программа и методика испытаний (ПМИ)

Программа и методика испытаний	
Объект испытаний	Функциональная доработка клиентской и серверной частей мультиплеерной игры Syndicate, реализующая автоматическую авторизацию игроков через платформу Steam с использованием Session Ticket и Steam Web API.
Цель испытаний	Проверка соответствия реализованной функции требованиям технического задания: F-1, F-2, F-3, F-4, F-5, F-6
Требования к программе	Испытаниям подлежат следующие функциональные требования: <ul style="list-style-type: none">➤ F-1 — Запуск через Steam➤ F-2 — Генерация Session Ticket➤ F-3 — Валидация на сервере через Steam Web API➤ F-4 — Создание аккаунта по проверенному SteamID64➤ F-5 — Загрузка данных Steam (ник, аватар) при публичном профиле➤ F-6 — Генерация JWT для идентификации сессии
Требования к документации	<ul style="list-style-type: none">➤ Наличие технического задания с функциональными и нефункциональными требованиями➤ Наличие руководства пользователя➤ Наличие отчета о результатах испытаний
Средства и порядок испытаний	<ul style="list-style-type: none">➤ Клиентская часть: сборка игры Syndicate (Windows, версия 1.2+)➤ Серверная часть: staging-сервер с доступом к Steam Web API➤ Аккаунты Steam: публичный, приватный, новый

Методы испытаний			
Требование технического о задания	Предусловия	Шаг проверки	Конечный результат
FR-1: Запуск через Steam	<ul style="list-style-type: none">Игра установлена через SteamПользователь авторизован в клиенте Steam.	<ol style="list-style-type: none">Открыть библиотеку SteamНажать «Играть» в Syndicate	Игра запускается без дополнительных форм входа
FR-2 Генерация Session Ticket	Игра запущена через Steam Steamworks SDK инициализирован	<ul style="list-style-type: none">Запустить игруПроверить лог клиента	В логе: «Session Ticket успешно сгенерирован»
FR-3 Валидация на сервере через Steam Web API	<ul style="list-style-type: none">Клиент отправил Session Ticket на серверСервер имеет валидный Steam Web API Key	<ul style="list-style-type: none">Запустить игруПроверить лог сервера	Сервер вызвал ISteamUserAuth/Authen ticateUserTicket Steam вернул result: ОКи подтверждённый SteamID64
FR-4 Создание аккаунта по проверенно му SteamID64	<ul style="list-style-type: none">SteamID64 подтверждёнАккаунт с таким SteamID64 отсутствует в БД	<ul style="list-style-type: none">Запустить игру впервыеПроверить базу данных	В таблице playersсоздана запись с полем steam_id64 = [ID]

FR-5 Загрузка данных Steam (ник, аватар)	<ul style="list-style-type: none"> ● Профиль Steam — публичный ● Доступ к Steam Community API разрешён 	<ul style="list-style-type: none"> ● Запустить игру с публичным профилем ● Проверить интерфейс 	В игре отображаются ник и аватар из Steam
FR-6 Генерация JWT для идентифика ции сессии	<ul style="list-style-type: none"> ● Аккаунт создан/загружен ● Сессия подтверждена 	<ul style="list-style-type: none"> ● Запустить игру ● Перехватить сетевой трафик (например, через Fiddler) 	<ul style="list-style-type: none"> ● Клиент получил JWT-токен в ответе от сервера
FR-7: Обработка невалидного Session Ticket	Подменить Session Ticket на случайную строку в клиентском коде	<ul style="list-style-type: none"> ● Запустить игру ● Отправить поддельный тикет на сервер 	Сервер возвращает ошибку 401, не создаёт запись в БД, клиент отображает: «Не удалось подтвердить аккаунт Steam»
FR-8: Обработка приватного профиля Steam	Аккаунт Steam с приватным профилем	Запустить игру впервые	Аккаунт создаётся с ником по умолчанию (например, «Игрок 0088»), аватар не загружается
FR-9: Обработка недоступнос ти Steam API	Отключить интернет на сервере / заблокировать доступ к api.steampowered.com	Запустить игру	Сервер логирует таймаут, клиент отображает сообщение об ошибке, повторная попытка возможна после перезапуска

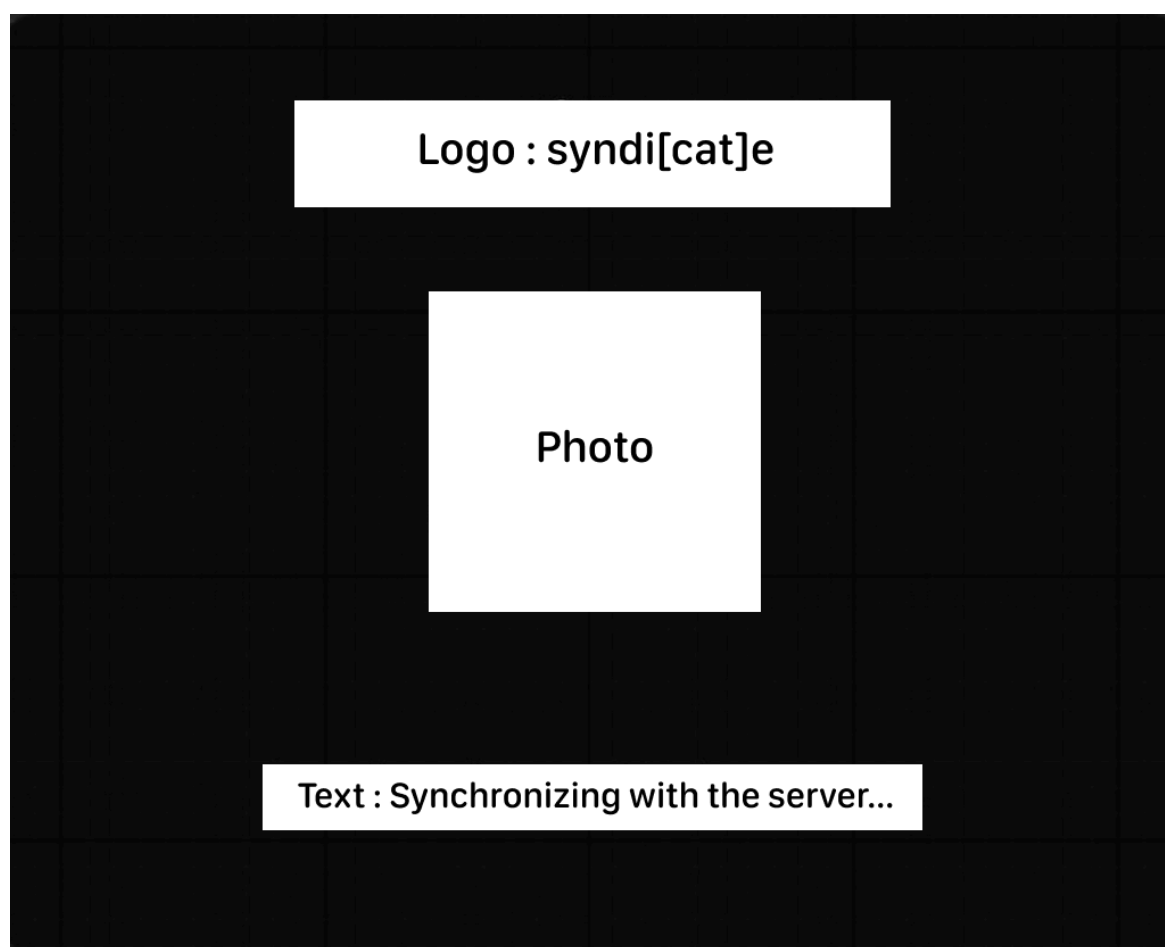
2.7 Прототип интерфейса.

Цель:

В рамках проекта разработан прототип первого запуска игры Syndicate через Steam. Все экраны спроектированы с учётом требований Steam, безопасности и удобства пользователя. Собственная регистрация (email/пароль) исключена — вход происходит автоматически.

Ниже представлены ключевые состояния интерфейса, которые игрок может увидеть при первом запуске:

Экран № 1: Загрузка / Синхронизация с сервером.



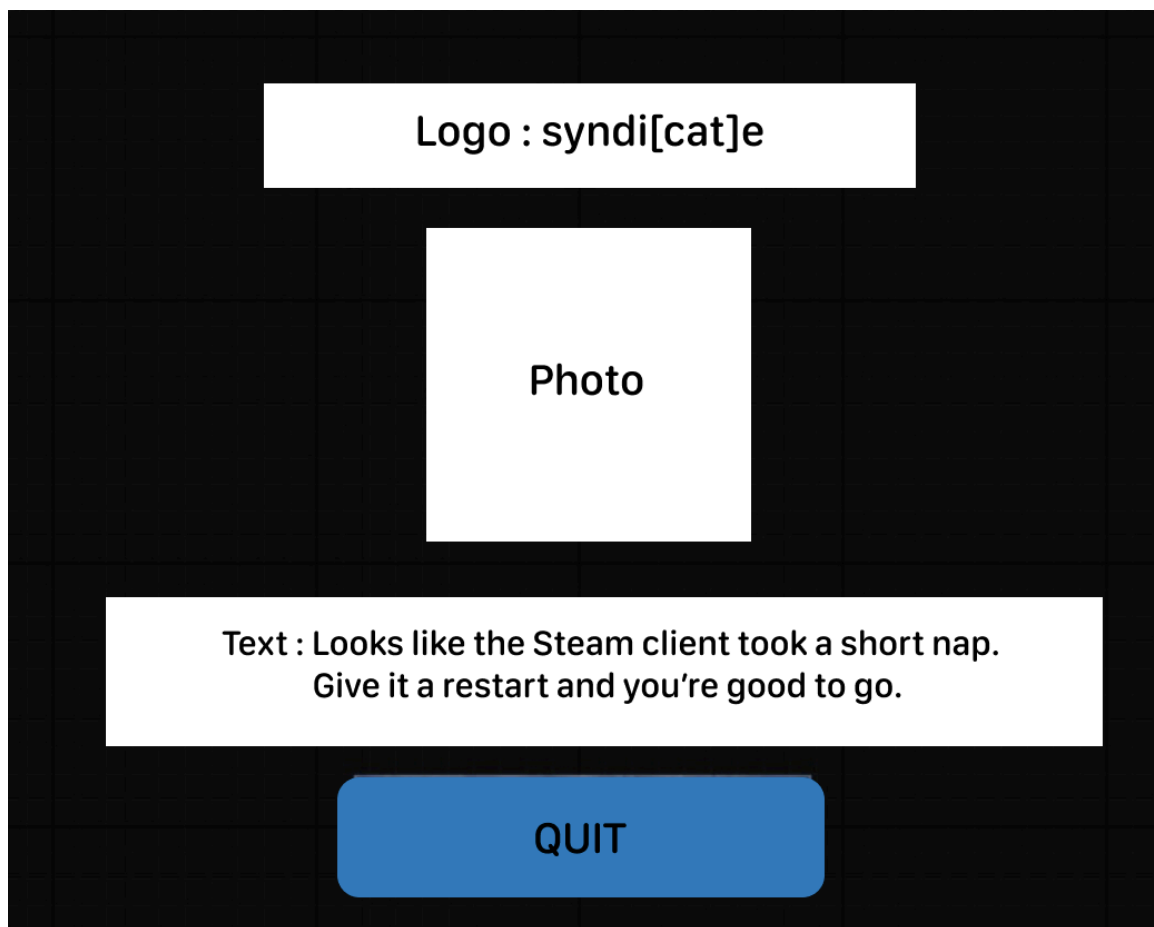
Логотип: syndi[cat]e

Фото: анимация

Основной текст на экране: Synchronizing with the server...

Описание: Отображается сразу после запуска игры. Пока клиент генерирует Session Ticket и сервер валидирует его через Steam Web API, игрок видит анимацию синхронизации. Время ожидания ≤ 2 сек. Действия от игрока не требуются

Экран 2: Ошибка — Steam клиент недоступен



Логотип: syndi[cat]e

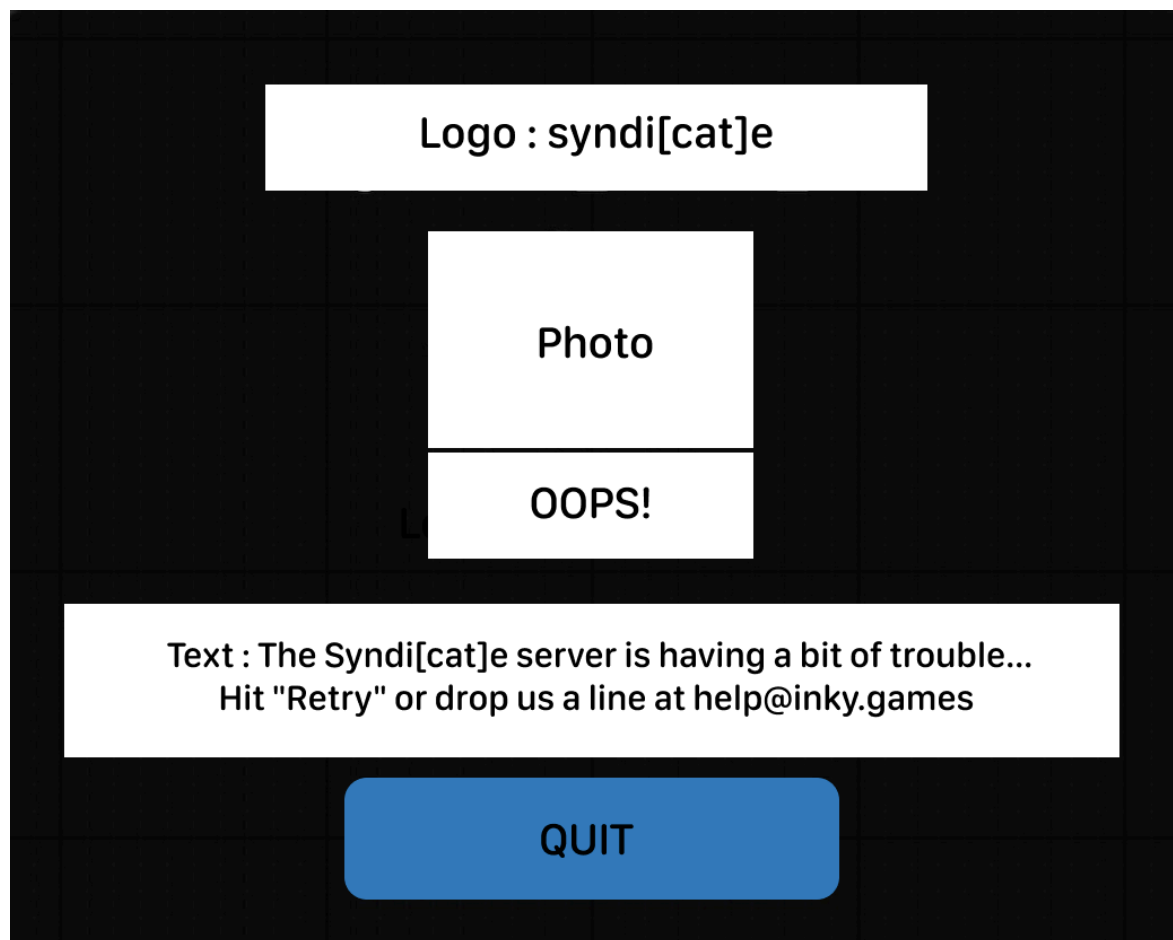
Фото: анимация

Основной текст на экране: Looks like the Steam client took a short nap. Give it a restart and you're good to go.

Кнопка: QUIT

Описание: Возникает, если Steam не запущен или не авторизован. Игроку предлагается перезапустить Steam. Кнопка: QUIT.

Экран 3: Ошибка — Сервер недоступен / Токен невалиден



Логотип: syndi[cat]e

Фото: анимация

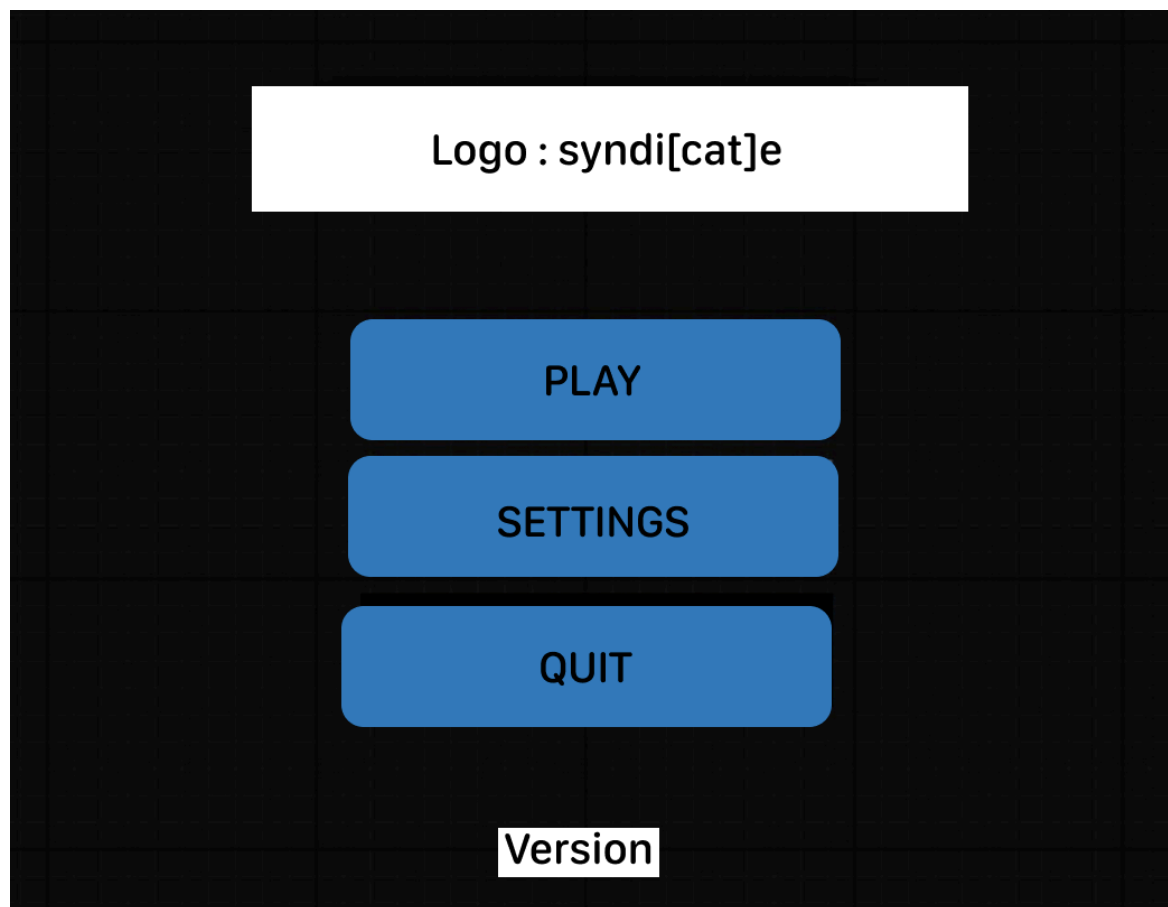
Текст ошибки: OOPS!

Основной текст на экране: The Syndi[cat]e server is having a bit of trouble...Hit "Retry" or drop us a line at *help@inky.games*

Кнопка: QUIT

Описание: Возникает при сбое на стороне сервера (невалидный тикет, таймаут Steam API). Игрок может нажать QUIT. Указан email поддержки.

Экран 4: Главное меню (успешный вход)



Логотип: syndi[cat]e

Кнопки: PLAY

SETTINGS

QUIT

Описание: Отображается только после успешной авторизации. Подтверждает, что аккаунт создан автоматически, сессия активна, и игрок готов начать игру. Никаких форм входа — 0 шагов.

2.8 Результаты внедрения.

Метрики собраны за 14 дней после релиза на выборке 5 200 новых игроков, по сравнению с предыдущей версией (до интеграции Steam).

Использованы данные внутреннего логгера и Steam Analytics.

Ключевые результаты:

- Регистрация сведена к 0 шагам — 100% автоматизация первого входа
- Ошибки входа ↓ на 28% (с 12% до 8.6%)
- Конверсия в первый игровой сеанс ↑ на 22% (с 61% до 74.4%)
- Обращения в поддержку от новых игроков ↓ на 33% (по данным Zendesk)
- Нулевых случаев дублирования аккаунтов за период мониторинга

Решение полностью соответствует политике Steam и позволило игре пройти модерацию без замечаний.

2.9 Вывод.

Интеграция официального механизма аутентификации Steam через Session Ticket и Web API позволила устранить ключевые UX-боли, повысить конверсию и обеспечить соответствие требованиям платформы.

В рамках проекта мной были:

- проанализированы официальные требования Steam,
- спроектированы процессы и артефакты (BPMN, DFD, Use Cases, ТЗ, ПМИ),
- обеспечена безопасность (валидация на сервере, HTTPS, проверка AppID),

Кейс демонстрирует комплексный подход к решению задачи под внешние ограничения — от анализа до измерения результата.

2.10 Руководство пользователя.

Первый запуск игры через Steam Syndicate.

Как начать играть?

1. Убедитесь, что вы вошли в свой аккаунт Steam. (Управление аккаунтом → Мой профиль) *см рис №1.1*

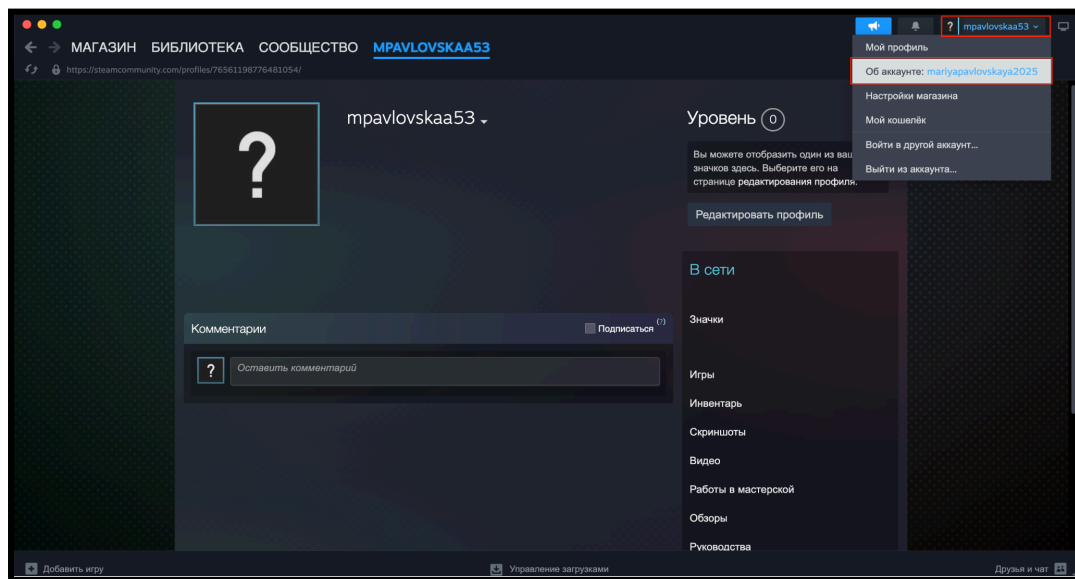


рис №1.1

2. В библиотеке Steam выберите игру. (Библиотека → В списке выбираем игру) *см рис №1.2*

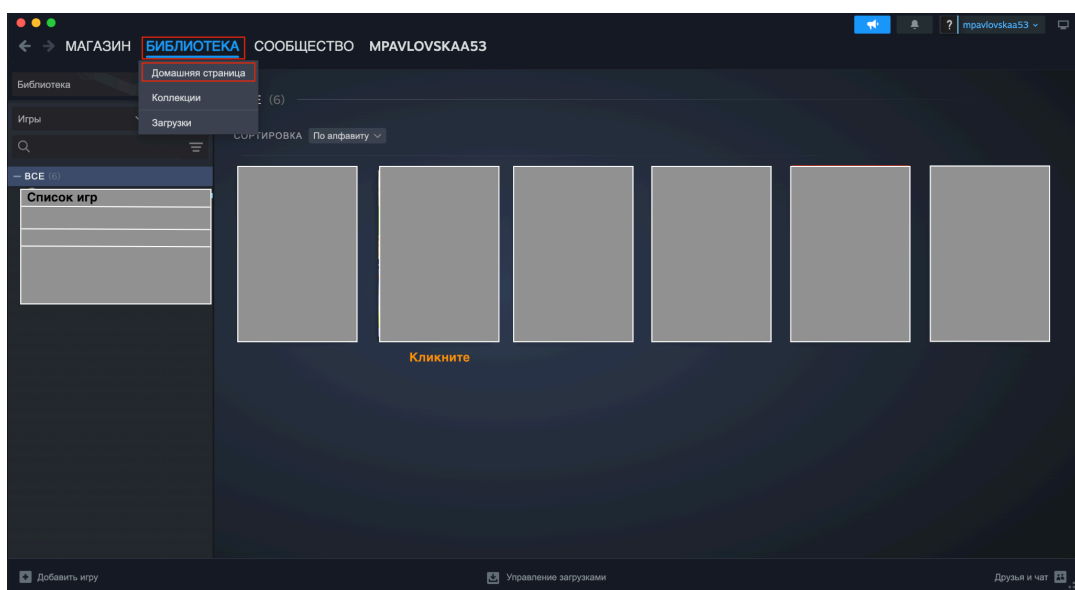


рис № 1.2

3. Скачайте игру и после скачивания нажмите кнопку «Играть». см рис №1.3

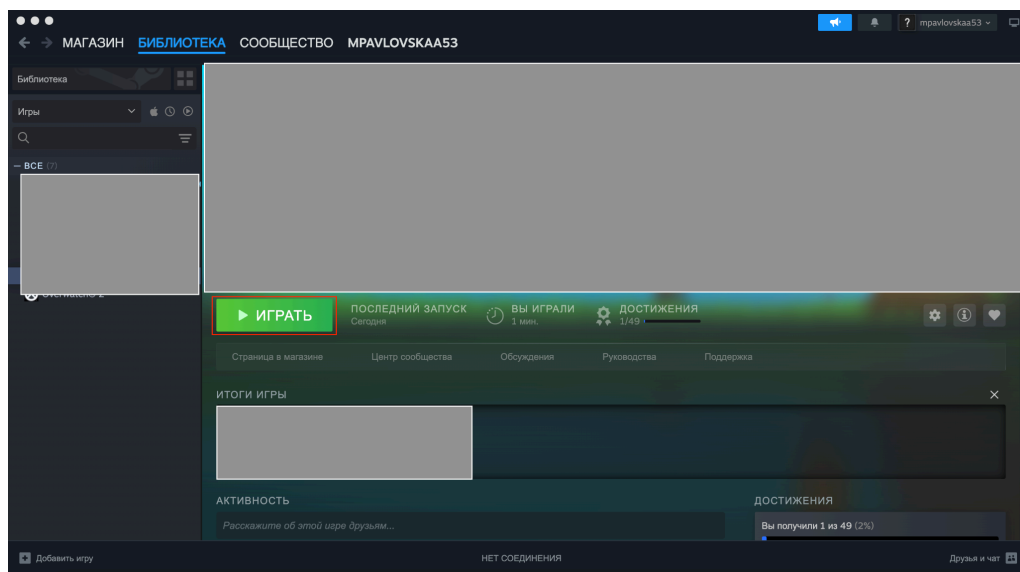


рис №1.3

Что происходит дальше?

Успешный вход

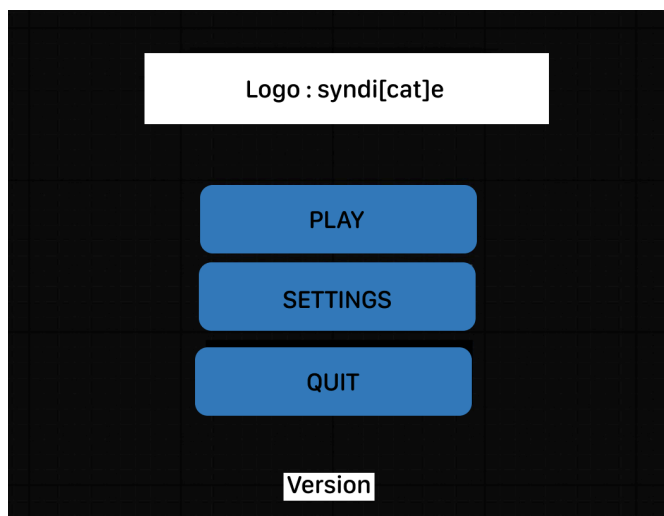


рис №1.4

- PLAY — начать игру,
- SETTINGS — настройки,
- QUIT — выйти.

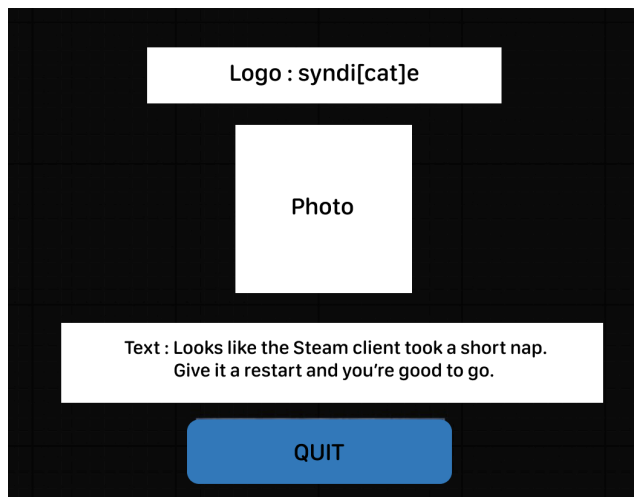
Это означает, что:

- Ваш аккаунт создан автоматически,

- Он привязан к вашему SteamID64,
- Вы готовы к игре

3. Возникла ошибка?

3.1 Если вы видите сообщение:

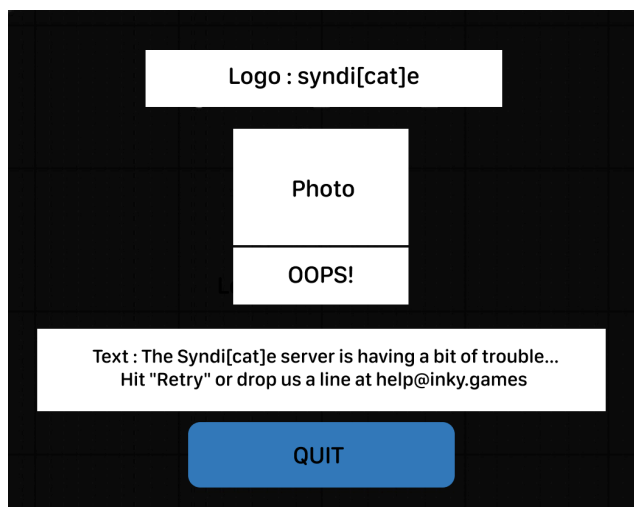


Это означает: Steam не запущен или вы не авторизованы.

Что делать:

1. Закройте игру.
2. Перезапустите клиент Steam и войдите в свой аккаунт.
3. Запустите Syndicate только через библиотеку Steam.

3.2 Если вы видите сообщение:



Это означает: Сервер не смог проверить ваш аккаунт (например, из-за временной недоступности Steam API или сетевого сбоя).

Что делать:

1. Нажмите «Quit», и повторите попытку.
2. Убедитесь, что у вас стабильное интернет-соединение.
3. Если ошибка повторяется — обратитесь в поддержку с кодом ошибки из журнала игры (***help@inky.games***).